FEMLAB®

User's Guide and Introduction

FOR USE WITH MATLAB

WINDOWS

UNIX

LINUX

MACINTOSH

VERSION 2.3

How to contact COMSOL:

COMSOL AB Tegnérgatan 23 SE-111 40 STOCKHOLM Sweden	COMSOL, Inc. 8 New England Executive Park Burlington, MA 01803 USA	Mail and visit
+46 (0)8-4129500	+1 781-273-3322	Phone
+46 (0)8-4129510	+1 781-273-6603	Fax
`http://www.femlab.com`		FEMLAB home page
`ftp://ftp.femlab.com`		FTP server
`http://www.comsol.se`	`http://www.comsol.com`	Company home page
`support@femlab.com`	`support@comsol.com`	Technical support
`suggest@femlab.com`		Product enhancement
`info@femlab.com`	`info@comsol.com`	Sales, pricing, and general information

Printing History:	October 2000	First printing	FEMLAB 2.0
	November 2001	Second printing	FEMLAB 2.2

C O N T E N T S

User's Guide and Introduction

1 User’s Guide and Introduction

Preface

FEMLAB 2.2

FEMLAB® was the first engineering tool for PDE-based multiphysics modeling in an interactive environment—MATLAB®. Our goals are ambitious. We are committed to lead the development in our field, and the new features in FEMLAB 2.2 prove this commitment. Despite its modest version number increment, version 2.2 is a major upgrade. We anticipate significant upgrades once per year for the foreseeable future.

Version 2.2 includes three major functionality *additions* and three major *improvements*. The three major additions are higher-order elements, the weak form, and extended multiphysics. The three major improvements are 3D CAD and image data import, powerful visualization tools, and automatic tuning of 3D solvers.

The *higher-order elements* significantly broaden the scope in the modeling process of physics phenomena. The higher-order elements are part of the new element library, which is designed to maintain the flexibility of equation-based modeling, while still allowing the addition of virtually any element type. The element library is open and extensible—you can add the element of your choice.

With the introduction of *extended multiphysics* and *the weak form*, we lay a profound foundation for a truly unique tool for modeling in engineering and science. We are confident that these new features will eventually prove even more useful than the original multiphysics concept. The Model Library shows how the weak constraint gives optimal accuracy in flux computations, and how extended multiphysics gives you the ability to post process on the full geometry while using symmetries to perform the computations on only a part of the geometry. The full capability of extended multiphysics with non-local couplings on multiple geometries is demonstrated in the thermal controller and packed bed reactor models. We are always impressed by the ingenuity of our users and by the way they utilize FEMLAB, and I rest assured that these inventions will be used in ways we never imagined.

CAD models in 3D can be imported in the IGES file format, the most common standard for product geometry data. The IGES translator includes a geometry repairing facility for increasing the quality of models. Utilities for the conversion of 2D images (JPG, GIF, TIF, BMP, etc.) and 3D MRI data to FEMLAB geometry objects have also been added.

You can now easily create cross-section plots in a number of ways. Choose between plotting the time evolution at a point or on a boundary or visualizing arbitrary expressions on cross-sections (lines in 2D and surfaces in 3D). An object can be sliced using the GUI and programming functions, and “probes” can be inserted anywhere in the geometry—a virtual test lab!

The new algebraic multigrid (AMG) preconditioner provides automatic tuning of the iterative solver parameters for large 3D models, reducing the need for manual interaction significantly.

The specialized modules provide comfortable working environments for particular fields of physics. They use standardized terminology, material libraries, specialized solvers and elements, and appropriate visualization tools. They all come with their own separate manuals and model libraries, which provide completed solutions to a variety of field-specific problems. The *Chemical Engineering Module*, the *Electromagnetics Module*, and the *Structural Mechanics Module* are available. In this release the Electromagnetics Module has been upgraded to include 3D support and vector elements. We planned to release the new Structural Mechanics Module with this version, but we had to postpone this release to version 2.3.

FEMLAB 2.3, due in the summer of 2002, will include improvements mainly to the add-on modules. Turbulence models for fluid dynamics will be included in the Chemical Engineering Module, as well as specialized solution methods for fluid dynamics. The new version of the Structural Mechanics Module will be fully integrated with FEMLAB multiphysics, and it will use the element library and other elements specific to structural mechanics: beam elements and shell elements—all in 3D. A pre-release version of the module is already available with version 2.2.

FEMLAB has been designed as a single platform for all physics-based modeling and analysis. It is already an industry-proven engineering tool, and with the future development that we envision, FEMLAB will grow to become the engineering standard for multiphysics modeling.

FEMLAB 2.3

FEMLAB 2.3 features new releases of the Structural Mechanics Module, the Chemical Engineering Module, and the Electromagnetics Module. FEMLAB 2.3 extends the versatility of the FEMLAB product line by introducing a set of new solvers for parametric analysis, modeling of large vector field models in electromagnetics and structural mechanics as well as transport processes in chemical engineering. It also introduces new eigenvalue solvers for large models in eigenfrequency and eigenmode analyses.

The new Structural Mechanics Module is now fully incorporated with FEMLAB and includes new application modes for shells, beams, and solids. In the Chemical Engineering Module, new application modes are introduced to study non-Newtonian fluids, compressible flow, multi-component transport, and turbulence. Furthermore, the Electromagnetics Module includes new application modes for optoelectronics and photonics, as well as new models in the fields of radio frequency and microwave engineering.

The new parametric solver offers an ideal way to examine a parameterized series of models —an important part of the multiphysics simulation procedure. The varied parameter typically represents a material property, frequency or reaction rate. The information obtained for one set of variables is then used for the following sets for fast and efficient parameter sweeping.

For highly nonlinear models, parameter sweeping may also be used for smooth convergence. The idea is to solve a series of increasingly difficult nonlinear problems. The solution of a slightly nonlinear problem, which is easy to solve, is used as the input to a more difficult nonlinear problem by increasing a parameter that represents the degree of nonlinearity. In this way, very difficult nonlinear problems can be solved without manual interaction.

The geometric multigrid (GMG) preconditioner significantly improves FEMLAB's ability to handle large models in structural mechanics. Moreover, in chemical engineering, this solver may be used to solve large diffusion-convection-reaction problems.

The new iterative eigenvalue solver allows one to solve large eigenfrequency problems in structural mechanics. It also extends FEMLAB's capabilities of handling analysis of cavities and other resonance structures in microwave engineering and photonics.

Enjoy!

May 5, 2002

[illegible]

Lars Langemyr, Vice President of Development

Introduction

Welcome to FEMLAB—your multiphysics solution in MATLAB®. This guide is intended to acquaint users with the full array of multiphysics capabilities within FEMLAB. Tutorials and examples step through FEMLAB's powerful functions and capabilities. You will learn how to set up a model in the FEMLAB graphical user interface (GUI), an environment that unleashes the power of the latest modeling techniques with the click of a mouse.

This *User's Guide and Introduction* is ideal for those who want to explore specific models as well as those who want a general overview of FEMLAB's cutting-edge multiphysics technology. We have structured this manual so that users can quickly and easily learn how to access FEMLAB's considerable power. It begins with a Quick Start Guide (page 1-26) that examines a typical problem in a step-by-step fashion to introduce you to FEMLAB's wide range of features and capabilities.

This manual contains a series of in-depth tutorials that explain how to:

- initialize new models and define their parameters through the Model Navigator as well as open prewritten models from the Model Library
- work with the FEMLAB GUI
- perform efficient modeling within the graphical interface
- enhance the capabilities of the graphical interface by working with FEMLAB functions in programming
- perform equation-based modeling using partial differential equations (PDE).

After reviewing the material in this manual, you will be ready to start creating your own models for your specific projects. Creating more advanced models generally requires that you become more familiar with FEMLAB's subtleties. For this reason, FEMLAB's documentation set includes several supplementary manuals. Specifically, the *Reference Manual* details the package's underlying concepts and shows how to take full advantage of its data structures and capabilities from the MATLAB prompt. The *Model Library* provides many examples from science and engineering that illustrate how to employ FEMLAB to solve real-world problems. For your convenience, the complete documentation can be found on your hard-drive after installation (PDF and HTML versions). Occasionally, the electronic version is more useful than the printed copy, such as when searching for key words.

What is FEMLAB?

FEMLAB is a powerful, interactive environment for modeling and solving scientific and engineering problems based on partial differential equations [1,2]. With it you can easily extend conventional models that address one branch of physics to state-of-the art multiphysics models that simultaneously involve multiple branches of science and engineering. Accessing this power, however, does not require that you have in-depth knowledge of mathematics or numerical analysis. Indeed, you can build many useful models simply by defining the relevant physical quantities rather than defining the equations directly. FEMLAB then internally compiles a set of PDEs representing the problem. FEMLAB also allows you to create equation-based models. Besides providing these multiple modeling approaches, FEMLAB offers multiple ways to harness this power: either through a flexible self-contained graphical user interface or from the MATLAB prompt.

The underlying mathematical structure with which FEMLAB operates is a system of *partial differential equations*. In FEMLAB you can represent PDEs in three ways: *coefficient form* (suitable for linear or nearly linear problems), *general form* (intended for nonlinear problems), and *weak form* (that works as a high-level finite element modeling language). Furthermore, it is possible to set up models as stationary or time-dependent, linear or nonlinear, scalar or multicomponent. The package also performs eigenfrequency or eigenmode analyses.

When solving the PDEs that describe a model, FEMLAB applies the *finite element method* (FEM). FEMLAB runs that method in conjunction with adaptive meshing and error control as well as with a variety of numerical solvers. A more detailed description of this mathematical and numerical foundation appears both in this manual and in the *Reference Guide*.

PDEs are the fundamental basis for the laws of science, hence they can and should be used to model scientific phenomena. FEMLAB has an extremely broad applicability, and it can model a large number of physical phenomena in many disciplines including:

- acoustics
- bioscience
- chemical reactions
- diffusion
- electromagnetics

- fluid dynamics
- fuel cells
- general physics
- geophysics
- heat transfer
- micro-electromechanical systems (MEMS)
- microwave engineering
- optics
- photonics
- porous media flow
- quantum mechanics
- radio-frequency components
- semiconductor devices
- structural mechanics
- transport phenomena
- wave propagation.

To show how FEMLAB solves familiar and interesting problems in many of these areas, this documentation set includes a *Model Library*. This separate volume contains an extensive selection of complete, ready-to-run models. Examining them is an excellent way to learn how to work with FEMLAB and to see how it applies to various application areas. Further, you can adapt, expand, or otherwise modify these models to suit your own requirements. They represent a handy starting point that can save considerable time in many instances.

Many real-world applications involve the simultaneous application of PDEs from several areas of science or engineering. Researchers now refer to this type of analysis as *multiphysics* modeling. For instance, the electrical resistance of a conductor often varies with temperature; thus a model of a conductor carrying current involves thermoelectric effects. This manual introduces you to FEMLAB's unique power in handling multiphysics (see "Forced and Free Convection Heat Transfer" on page 1-26 as well as "Thermo-Electric Heating in a Bus Bar" on page 1-109). Further, the *Model Library* devotes a separate chapter to the study of several interesting multiphysics examples.

Even in its base configuration FEMLAB offers enormous modeling and analytical power for many disciplines. However, the product has proven particularly useful in several of these areas. Thus we created *optional discipline-specific application modules* that make it easy to create and analyze models using terminology and solution methods appropriate for those disciplines. We presently offer the Chemical Engineering Module, the Electromagnetics Module, and the Structural Mechanics Module.

Despite all these aids, we recognize that it is impossible to anticipate every possible application area and every user's potential requirements. To give users the ultimate flexibility, the FEMLAB user interface integrates seamlessly with MATLAB, the package that provides the computational engine behind FEMLAB. Indeed, FEMLAB frequently uses MATLAB's syntax and data structures. An enormous benefit of this tight integration is that you can save and export FEMLAB models as MATLAB programs that run directly in that environment or incorporate them with still other products in the MATLAB family. Thus you gain the freedom to combine FEM-based modeling, simulation, and analysis with numerous other techniques in engineering and science. For instance, it is possible to create a finite-element model in FEMLAB and then export it to Simulink® or to the Control System Toolbox, where the model becomes an integral part of the simulation of a dynamic system. We refer to such applications of FEMLAB as *multidisciplinary*.

We are glad that you have chosen FEMLAB for your modeling needs, and we hope that FEMLAB will perform up to your expectations. If any difficulties arise in your FEMLAB modeling experience, please contact our support team at support@femlab.com. Thanks again for using FEMLAB!

REFERENCES

[1] S. Littmarck, *et. al.*, Solving differential equations, *Industrial Physicist*, American Institute of Physics, Feb/Mar 2001.
URL: http://www.aip.org/tip/pastiss.html

[2] S. Littmarck, *et. al.*, Math, models, motion and more, *PT Design Magazine*, Penton Media (Cleveland, OH), May 2000.

The FEMLAB Environment

To understand the discussions in this manual, as well as other elements of the documentation set, you need to be familiar with some basic terminology. At the top-most level, the FEMLAB environment consists of the following parts:

The Graphical User Interface (GUI) In this environment you create and manipulate models in two ways: by using one of several predefined *physics modes* where you work with familiar scientific laws and relationships, or in one of the predefined *PDE modes* where you work directly with a model's underlying partial differential equations. You can also perform multiphysics modeling by combining any of these physics modes and PDE modes.

The physics modes designed to run under FEMLAB's graphical interface address common application areas such as heat conduction, electromagnetics, fluid dynamics, and structural mechanics. Each mode, in turn, relies on a specific PDE model for which you define commonly used parameters, variables, and their values. Because of the predefined nature of these physics modes, you can set up a model without the need of explicitly stating the governing PDE.

The FEMLAB GUI contains a set of geometry tools (CAD) for modeling in 1D, 2D, and 3D, as well as tools for importing DXF and IGES files, with geometry repair. A built-in mesh generator automatically creates the mesh for any geometry. If your geometry is represented on an image file format or as magnetic resonance imaging (MRI) data, you can convert this data to a FEMLAB geometry object.

Recognizing the necessity of being able to visually examine results of model simulations, FEMLAB provides several tools that make it quick and easy to visualize virtually any interesting quantity or parameter. Among the choices available are surface plots, slice plots, isosurfaces, contour plots, cross-section plots, and animations.

Extensive export facilities in the graphical interface enable you to perform postprocessing from the MATLAB prompt using any FEMLAB or MATLAB function. Dialog boxes guide you through the process of exporting models to other tools in the MATLAB family, such as Simulink or the Control System Toolbox. You can save a GUI-built model as a Model MAT-file, handled by the Model Navigator, or a Model M-file. See below for details.

The Model Navigator This user interface tool is a multi-purpose dialog box in which you control the overall settings of a FEMLAB session. The New page initializes models in the various application modes, while the Multiphysics page lets you add additional application modes to create a multiphysics model. The Model Library page loads complete models supplied with FEMLAB. On the New page, you also choose to create a 1D, 2D, or 3D model. On the User Models page, you can browse your own set of saved models. If you have saved a model together with a model image and a model description, you can display the model with an image and a short description in the same way you display models on the Model Library page.

The Model Library This set of predefined models, which you can access from the Model Navigator dialog box, is important for several reasons. It offers a quick way to learn how to use FEMLAB's capabilities, and it lets you inspect the results of completed models. You can interact with a model by modifying its geometry and key parameters, such as the values of the underlying variables. Feel free to use a model as a starting point and save it under a different name (thus creating a copy).

Model M-files It is possible to export any model created in the graphical interface as a Model M-file—a MATLAB script version of the model. Representing a model in this form is convenient for documenting the work you perform in the graphical interface or for allowing further modeling using script programming.

Script Programming and the FEMLAB Functions Operating FEMLAB as a programming environment enables you to create and manipulate models within MATLAB. Here you work directly with the FEMLAB data structure (known as the FEM structure). Combining MATLAB and FEMLAB functions makes it possible to run a series of models by varying a parameter. Also, this is a powerful tool for highly nonstandard modeling.

The Application Program Interface (API) This library of MATLAB functions and methods allows you to create GUI components and thus customize FEMLAB's graphical interface for specific applications. By building a custom interface optimized for a specific task, you can create complete applets that shield occasional users from unnecessary complexity yet allow them to quickly obtain results from sophisticated analyses. You can also use the API for building parameterized models for the GUI.

Open Programming Environment The environment consisting of FEMLAB together with MATLAB allows for using C, C++, and FORTRAN programs for defining material properties, loads, and sources through the MATLAB API. For information on how to call C, C++, and FORTRAN programs from FEMLAB and MATLAB, see the MATLAB documentation on *External Interfaces and API*.

The FEMLAB Documentation Set

To help you take advantage of all this functionality, FEMLAB comes with an extensive set of documentation. There are also minicourses available for downloading from the web. Assistance starts with on-line help for FEMLAB functions as well as both on-line and printed versions of all FEMLAB manuals. The full set of printed documentation includes the following titles:

- The *User's Guide and Introduction* now in your hands explains how to get started with FEMLAB. It does so primarily by guiding you through several detailed model examples. It also contains brief descriptions of the operation of the graphical interface and FEMLAB functions.
- The *Model Library* consists of complete FEMLAB models from different areas of science and engineering such as heat transfer, chemical engineering, electromagnetics, structural mechanics, and fluid dynamics, as well as equation-based models. It serves as a basis for getting a deeper knowledge of FEMLAB modeling techniques.
- The *Reference Manual* is an extensive volume that contains several important sections:
  - The *Reference Guide* provides a detailed presentation of FEMLAB. It focuses on the data structures and functions available from the MATLAB prompt. The information is also necessary for advanced use of FEMLAB from the graphical interface.
  - *The Graphical User Interface* is a section that supplies a comprehensive description of the GUI including a review of all dialog boxes and their options.
  - The *Function Reference* explains the syntax and use of all FEMLAB functions for programming. This section is extremely valuable for users running FEMLAB from the MATLAB prompt as well as those working with Model M-files or script programming.
- The section titled *Installation Guide* contains information on how to install FEMLAB. The document also contains the FEMLAB license agreement.
- The *Automatic Control* manual contains detailed information about the automatic control applications in the FEMLAB Model Library. This document is available only in PDF format.
- The *FEMLAB Application Program Interface Guide* explains how to write MATLAB functions and methods that interact with FEMLAB. This document is available only in PDF and HTML formats.
- The *FEMLAB Release Notes* covers the latest news on the release. This document is available only in PDF and HTML formats.

All documents are available in two electronic formats, PDF and HTML. The electronic versions are available on your hard-drive after installation.

Optional Modules for Specific Disciplines

The Chemical Engineering Module, the Electromagnetics Module, and the Structural Mechanics Module are specialized modules, which provide comfortable working environments for particular fields of physics. They use standardized terminology, material libraries, specialized solvers and elements, and appropriate visualization tools. They all come with their own manuals and model libraries, unique to their field.

CHEMICAL ENGINEERING MODULE

The Chemical Engineering Module provides a powerful way of modeling equipment and processes in the field of chemical engineering. It lets you easily model mass, heat, and momentum transport coupled to chemical reactions in 1D, 2D, or 3D. The Chemical Engineering Module is designed for the chemical engineer in research, design, development, and education. It is used in many areas of chemical engineering and technology, including:

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

The Chemical Engineering Module provides tailored interfaces and formulations for problems involving momentum, mass and heat transport coupled with chemical reactions. You can use these formulations while still having the full flexibility of modeling with your own equations.

FEMLAB excels in solving systems of coupled nonlinear PDEs. These are especially widespread in chemical engineering, where they appear in problems involving

- Heat transfer
- Mass transfer through diffusion, convection and migration
- Fluid dynamics

- Chemical reaction kinetics
- Varying material properties.

You can choose to model fluid flow through porous media or to characterize the flow with the Navier-Stokes equations. Chemical reactions are easily represented by source or sink terms in the mass and heat balances and can be of any arbitrary order. All formulations exist for both Cartesian coordinates and axisymmetry and for stationary and time-dependent cases.

Available application modes are:

- Momentum equations
  - Navier-Stokes equations
  - Darcy's law
  - The Brinkman equation
  - Non-Newtonian flow
  - Turbulent flow, k - ε turbulence model
  - Compressible Euler flow
- Energy equations
  - Heat conduction
  - Heat convection and conduction
- Mass balances
  - Diffusion
  - Convection and diffusion
  - Maxwell-Stefan convection and diffusion
  - Nernst-Planck transport equations

In the field of chemical engineering, the multiphysics capabilities of FEMLAB enable you to fully couple and simultaneously model fluid flow, mass and heat transport, and chemical reactions.

ELECTROMAGNETICS MODULE

The Electromagnetics Module provides a unique environment for the simulation of wave and field propagation, as well as AC-DC electromagnetics, in 2D and 3D. It handles static, transient, and frequency-domain simulations, as well as mode analysis, in a user-friendly graphical user interface. The Electromagnetics Module offers the perfect foundation for modeling microwave components and photonic devices. For

AC-DC modeling it is a powerful tool for detailed analysis of coils, capacitors, and electromachinery.

The Electromagnetics Module is used for component design in virtually all areas where electromagnetic field simulations are needed:

- Power system components
- Micro-electromechanical systems (MEMS)
- Antennas
- Waveguides and cavity resonators in microwave engineering
- Optical fibers
- Photonic waveguides
- Photonic crystals
- Active devices in photonics
- Semiconductor devices

Available application modes are:

- Electrostatics
- Magnetostatics
- Low-frequency electromagnetics
- In-plane wave propagation
- Axisymmetric wave propagation
- Full 3D vector wave propagation
- Full vector mode analysis in 2D and 3D

You can use these application modes, while still having the full flexibility of modeling using your own equations. Any material properties imaginable are supported, including inhomogeneous and fully anisotropic materials and media with losses or gains. Complex-valued material properties can be used in all formulations for general time-harmonic field simulations. The multiphysics capabilities of FEMLAB enable coupling all simulations with heat transfer, structural mechanics, and fluid flow formulations, as well as any physical phenomena described by PDEs.

STRUCTURAL MECHANICS MODULE

The Structural Mechanics Module solves problems in the structural mechanics field of engineering, adding special structural mechanics elements: beams, plates, and shells, for different engineering simplifications. Like all discipline-specific modules, it provides a set of prewritten library models.

Available application modes are:

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

The analysis capabilities are static, eigenfrequency, transient, and frequency response. Both linear and nonlinear material models are supported.

The application modes are fully multiphysics enabled making it possible to couple with any other physics application mode in FEMLAB or the other modules. Coupling with temperature is one example of multiphysics easily implemented with the Structural Mechanics Module.

The underlying equations for structural mechanics are available in several of the application modes—a feature unique to FEMLAB. This makes non-standard modeling close at hand. You can change material models from isotropic to orthotropic. The Structural Mechanics Module also features an extended material library as well as a beam cross section library.

In addition, you can include accurate finite element models as blocks in a dynamic simulation set-up with Simulink. This combination reduces the need for approximations and ad hoc models in Simulink simulations. FEMLAB's tight integration into the MATLAB environment makes the Structural Mechanics Module very versatile. For instance, you can use any valid MATLAB expression to describe loads and constraints. The possibility to save the model as a Model M-file makes it very easy to perform complex parameter studies.

Installing FEMLAB

For information on how to install FEMLAB, see the separate booklet titled *Installation Guide*. In that manual you can also find information on system requirements and the license agreement. The release notes for FEMLAB 2.3 are available on the CD and on your hard drive after installation.

New Features in FEMLAB 2.2

The major new capabilities in FEMLAB 2.2 are:

HIGHER-ORDER ELEMENTS

The unique equation-based modeling in FEMLAB has taken another step: Higher-order elements give extreme flexibility in the process of modeling physics phenomena. You can easily combine elements of different type and order. User-defined elements can be added to the element library.

EXTENDED MULTIPHYSICS

FEMLAB now supports extended multiphysics modeling for simultaneous simulation of several physics models. You can now model the interaction between geometrical domains of different dimensions: 0D, 1D, 2D, 3D plus time, for example, by integrating a solution on one domain and using it as a material property in another. This enables true engineering solutions to complex problems, previously beyond reach for simulation. Automatic Jacobian computation for non-local couplings is supported. You can access dependent variables non-locally, and you can define scalar coupling variables by coordinate values or integration and coupling field variables by projection or extrusion.

3D CAD FILE IMPORT

CAD import for 3D models is now available, with geometry repair on industrial quality solid models. FEMLAB 2.2 supports the most common format, IGES, for CAD import.

IMAGE AND MRI IMPORT

The new release combines the power of image processing and visualization built into MATLAB with the multiphysics and FEM capabilities of FEMLAB in that images and MRI data can now be imported. Images and MRI data are converted to FEMLAB geometry objects and can then be used for analysis as any other geometry model.

CROSS-SECTION VISUALIZATION

Visualize postprocessing expressions of the solution and its gradient on cross sections by defining slicing planes in the graphical user interface. You can also visualize the dynamic behavior of a model by plotting the time evolution of the solution on a cross-section or at any location inside the model.

THE WEAK FORM

As the first simulation environment in the history of numerical computing, FEMLAB 2.2 introduces a high-level modeling language that makes it possible to model all types of finite element analysis problems by introducing “the weak form”. The weak form is at the heart of the finite element method, and it gives you complete control of the finite element model.

This functionality makes it possible to compute fluxes and other flux-related quantities with maximum accuracy. Some examples of flux-related quantities are reaction forces in structural mechanics and surface charges and surface currents in electromagnetics.

AUTOMATIC TUNING OF SOLVER PARAMETERS FOR LARGE MODELS

A new solver for large 3D models makes manual tuning of solver parameters unnecessary. The algebraic multigrid preconditioner (AMG) implements state-of-the-art numerics for higher performance and less memory usage.

MISCELLANEOUS

- Point and edge constraints and sources in 3D
- Point constraints and sources in 2D
- Define variables as symbolic expressions; variables can be differently defined on different domains
- Use space derivatives in boundary conditions
- Use time derivatives in Neumann boundary conditions
- Define a system of ODEs on a boundary, and couple this system to the PDE boundary conditions
- Fifth order Argyris element in 2D

New Features in FEMLAB 2.3

The major new capabilities in FEMLAB 2.3 are:

PARAMETRIC SOLVER

For examining the influence of different parameters that describe a model or simulation. the new parametric solver offers an ideal way to study a parameterized series of models. The varied parameter typically represents a material property, frequency or reaction rate. The information obtained for one set of variables is then used for the following sets for fast and efficient parameter sweeping.

For highly nonlinear models, parameter sweeping may also be used for smooth convergence. The idea is to solve a series of increasingly difficult nonlinear problems. The solution of a slightly nonlinear problem, which is easy to solve, is used as the input to a more difficult nonlinear problem by increasing a parameter that represents the degree of nonlinearity. In this way, very difficult nonlinear problems can be solved without manual interaction.

GEOMETRIC MULTIGRID (GMG) PRECONDITIONER

The new geometric multigrid (GMG) preconditioner is applicable to a range of problem types commonly known as elliptic PDE problems. This problem class contains structural mechanics problems, different kinds of diffusion problems, and problems including PDEs similar to Poisson's equation. The GMG preconditioner significantly improves FEMLAB's ability to handle large models in structural mechanics. Moreover, in chemical engineering, this solver may be used to solve large diffusion-convection-reaction problems.

ITERATIVE EIGENVALUE SOLVER

A new iterative eigenvalue solver is available that is based on the standard FEMLAB preconditioners and iterative solvers. It allows you to solve large eigenfrequency problems in structural mechanics. It also extends FEMLAB's capabilities of handling analysis of cavities and other resonance structures in microwave engineering and photonics.

AUTOMATIC SCALING OF VARIALBLES

If the dependent variables in your model has widely different magnitudes, a solver might have problem with the resulting ill-conditioning. For instance, in a structural mechanics problem, the displacements can be of the order of 0.0001 m, while the stresses are 1000000 Pa. To remedy this, FEMLAB 2.3 provides automatic internal rescaling of the variables so that a well-scaled system results. You can access the scaling properties from the graphical user interface and provide the automatic scaling mechanism with additional information on scaling known from the physical properties of the problem at hand.

MISCELLANEOUS

FEMLAB 2.3 additionally includes the following features:

Associative Geometry in 1D

FEMLAB features heuristic rules for keeping PDE and boundary conditions as changes are made to the geometry. This feature is called associative geometry and is supported in 2D and 3D since earlier versions of FEMLAB. In FEMLAB 2.3, the associative geometry feature is made available in all dimensions by the addition of associative geometry rules for 1D problems.

Line Plots in 3D

Line plots are now supported i 3D. This feature has been available in 2D since earlier versions.

Explicit Streamline Diffusion in the Navier-Stokes Application Modes

The Navier-Stokes application modes now features explicit streamline diffusion. Earlier versions of FEMLAB supported automatic streamline diffusion. This feature is still available and is now complemented by an explicit version. The explicit streamline diffusion feature makes the additional weak terms corresponding to the modified test functions available in the GUI. Thus, you can now manually inspect and modify the streamline diffusion contributions.

New Elements

Hermite elements are now available. The Hermite and Argyris element types are now available in the graphical user interface.

New Features in Chemical Engineering Module 2.1

NEW APPLICATION MODES

There is one new application mode: mass balance for multicomponent simulations.

NEW MODELS

- MEMS heat exchanger
- Packed bed reactor

MISCELLANEOUS

- The number of customized postprocessingprocessing variables has been increased significantly.

New Features in Chemical Engineering Module 2.3

NEW APPLICATION MODES

The new application modes are:

- Non-Newtonian fluids: Fluid flow studies can now be easily performed using the new application mode for non-Newtonian fluids, suitable for treating different types of shear-thickening and shear-thinning problems. This new application mode can be applied in the modeling of flow in polymer and fiber suspensions, pastes, etc.
- Compressible flow: Euler models are now available for the simulation of gas flow at high flow rates.
- Incompressible turbulent flow: A new k - ε turbulence model is now available as a ready-to-use application mode.
- Maxwell-Stefan diffusion: In studies regarding transport of diluted species in a carrier gas, Fickean diffusion accounts for the interaction between the dissolved species and the dominating solvent gas. When all species are present in a comparable amount, the interactions between all the involved species must be accounted for. This is addressed in the Chemical Engineering Module by the new Maxwell-Stefan application mode, which also include the combination of arbitrary reaction kinetics, momentum and heat transport. This feature allows for the modeling of gas phase reactors, separators, and filtration units.
- Nernst-Planck application modes: In applications regarding the mass transport of charged species, migration in an electrical field gives a considerable contribution to the flux. Hence, two new application modes in the Chemical Engineering Module ensure that the proper description of electro-kinetic fluid transport and transport in electrochemical cells are available in ready-to-use formulations. This considerably simplifies the task of modeling transport in MEMS devices, fuel cells, and batteries.

NEW MODELS

The Chemical Engineering Model Library features several new models:

- A model of the non-Newtonian flow of a polymer solution in a valve. The model shows on large variations in viscosity as a function of position and shear rate in the flow.
- A study of the flow in an open channel with porous walls, where the flow in the channels and in the porous media are fully coupled. The problem is typical for

monolithic reactors, filters, and other devices where the convective flow in the porous media cannot be neglected.

- New k-ε models including jets, a static mixer, and the classical backwards-facing step. Logarithmic wall functions are used as boundary conditions for the solid walls.
- Compressible flow at supersonic speed and in the presence of shock waves is modeled using the compressible Euler equations. The model treats the supersonic flow in a channel with a solid obstacle in the middle of the channel.
- The use of the new parametric solver is exemplified in a model of heat transport around a heated tube at different flow rates of the streaming fluid. The model shows the influence of detachment of the viscous layer on the temperature profile in the fluid.
- A new model of a fuel cell cathode presents the application of the multicomponent diffusion. The application mode is based on the Maxwell-Stefan diffusion and shows on the dependence of the binary diffusion coefficient on composition.
- The transport by diffusion and migration of ionic species in electrochemical cells is studied. The model treats the electrolysis of tumors where tumor destruction is achieved through acidification and chlorine production.
- A new model of electrokinetic flow in a DNA chip is presented. The electroosmotic flow is driven by the electric field and the deviations from electroneutrality at the walls of the channel. The model couples fluid flow and ionic current conduction in the chip.
- The exothermic reactions in a fixed-bed tubular reactor is investigated through the definition of coupled heat and mass balances. The production of phthalic anhydride is highly dependent on the temperature distribution in the reactor.

New Features in Electromagnetics Module 2.0

NEW APPLICATION MODES

There are four new application modes:

- 3D Electrostatics application mode for conductors, capacitors, and other types of voltage driven static problems
- 3D Magnetostatics application mode for permanent magnets, inductors, and other types of static magnetic problems

- 3D Quasi-Statics application mode for low frequency phenomena in electromagnetics, such as transformers and electromachinery
- 3D Electromagnetic Waves application mode for high frequency phenomena in electromagnetics, such as waveguides and optical fibers

ELEMENTS

- Element library with Lagrange elements of arbitrary order enabling higher accuracy
- Vector elements for 3D electromagnetic wave modeling: The vector elements ensure that the correct interface conditions are fulfilled at borders between different media.

NEW MODELS

- 3D MEMS capacitor
- 3D MEMS inductor
- Quasi-static 3D model of a crucible
- 3D waveguide including an S-parameter analysis
- Waveguide in a photonic crystal
- Step-index fiber
- Photonic micro prism model using absorbing layers
- T-junction waveguide model enhanced with an S parameter analysis

MISCELLANEOUS

- Multiphysics with non-local couplings for multiple geometry modeling
- Domain dependent variables
- Material library for user-defined materials
- Cross-section plots

New Features in Electromagnetics Module 2.3

NEW APPLICATION MODES

The new features related to application modes are:

- A new 2D application mode for mode computations of hybrid-mode waves is now available. The application mode has general support for anisotropic media and is applied to problems in photonics, optics, and microwave waveguides.

- Impedance boundary conditions have been added to the 3D quasi-statics application mode. This lets you model induced currents as surface currents along the boundaries, which makes it possible to work with higher frequencies.

NEW MODELS

The following models have been added to the Model Library:

- A model of a monoconical RF antenna with a study of the antenna impedance and the radiation pattern. Conical antennas are useful for many applications due to their broadband characteristics and relative simplicity. The rotational symmetry allows for using one of the axisymmetric electromagnetic wave propagation application modes. When modeling in 2D, a denser mesh can be used which gives excellent accuracy for a wide range of frequencies.
- A magnetostatic model using Maxwell's stress tensor to calculate the force on two parallel wires. This model illustrates the use of boundary integrals to compute the resultant force on current conductors. A comparison with the volume integral of the force density is made.
- A 3D model using showing how a scalar magnetic potential can be used to efficiently solve magnetostatic problems for permanent magnets. This model also uses Maxwell's stress tensor for force calculations.
- Two models calculating stress-induced birefringence in a photonic waveguide. The simulation is set up as a multiphysics problem coupling a plane strain equation with a optical mode analysis. One of the models uses a generalized form of the plane strain equation, which takes the strain if all three directions into account.
- The S-parameter analysis in the elliptical to rectangular waveguide transition model, as well as in the T-junction waveguide model is remade using the new parametric solver.

MISCELLANEOUS

- A "matched boundary" condition has been added to the wave application modes, which are exactly non-reflecting for eigenmodes of the boundary.
- A new chapter on force calculations has been added to the manual.

New Features in Structural Mechanics Module 2.3

NEW APPLICATION MODES

There are three new application modes.

- 3D solids
- 3D Euler beams
- Shells

ELEMENTS

- Element library with Lagrange elements of arbitrary order enabling higher accuracy
- Beam, bar, and shell elements in 3D

NEW MODELS

- 3D tower, Euler beam model
- Static analysis of a socket to a guyed mat
- 2D beam benchmark problem
- Axisymmetry piston model, transient coupling with temperature
- Steel aluminum component, coupling with temperature
- Pressure vessel modeled using shell elements
- Eigenfrequency analysis of a rotor in an electric motor
- Parameter analysis of a tube connection with prestressed bolts
- Vibration of a disk backed by an air-filled cylinder

MISCELLANEOUS

- Fully multiphysics enabled applications in 2D and 3D
- Domain dependent variables
- Cross-section plots
- Easy thermal coupling with arbitrary temperature field
- New design of the material library; simplifies and extends its use
- Possibility to use integrals in the problem definition
- General parametric solver

- Possibility to reuse inputs when specifying your problem
- Improved contact problem handling using non local coupling and nonlinear boundary conditions

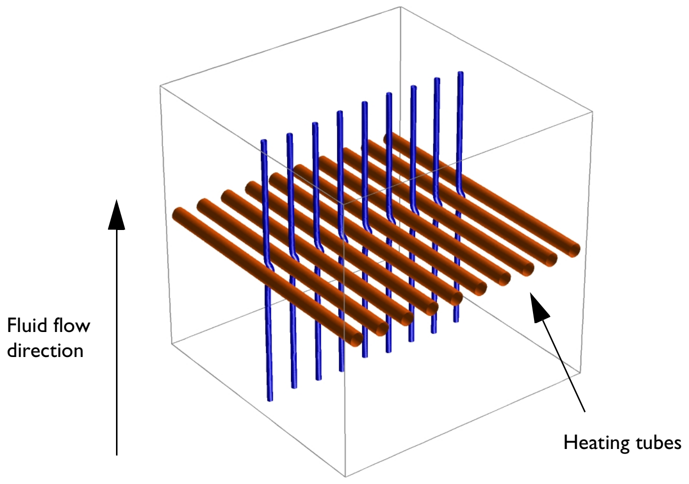
The FEMLAB Quick Start Guide

The goal of this section is to familiarize you with the FEMLAB environment, focusing primarily on how to use its graphical user interface. To facilitate a quick introduction, the following model will examine steps necessary to setup and evaluate a simple project, and review the major tasks of the modeling process.

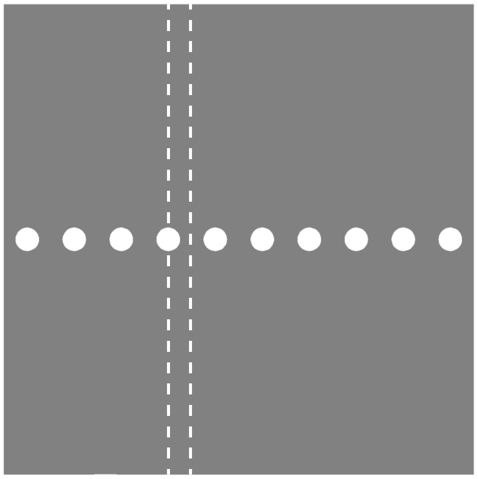
Forced and Free Convection Heat Transfer

MODEL BACKGROUND

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. The figure below depicts the setup.



A first consideration when modeling should always be the true dimension of the problem. Many problems do not show variations in three dimensions and can be extrapolated from the solution of a related 2D case. Assuming any end effects from the walls of the vessel can be neglected, the solution can be assumed constant in the direction of the heating tubes, and the model is therefore reduced to a 2D domain (below).



The next consideration is finding symmetries. In this case, inclusion of symmetry planes allows you to model only the thin domain indicated in the figure.

GOVERNING EQUATIONS

This is a multiphysics model, meaning that it involves more than one kind of physics. In this case, you have Incompressible Navier-Stokes equations from fluid dynamics together with a heat transfer equation, that is, essentially a convection-diffusion equation. There are four unknown field variables: the velocity field components *u* and *v*; the pressure, *p,* and the temperature, *T*. They are all interrelated through bidirectional multiphysics couplings.

The pure Incompressible Navier-Stokes equations consist of a momentum balance (a vector equation) and a mass conservation and incompressibility condition. The equations are

$$\begin{cases} \rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \eta\nabla^2\mathbf{u} + \mathbf{F} \\ \nabla\cdot\mathbf{u} = 0 \end{cases}$$

where **F** is a volume force, ρ the fluid density and η the dynamic viscosity. We denote the vector differential operator by ∇ (pronounced "del"). See further "Overview of PDE Models" on page 1-336.

The heat equation is an energy conservation equation, that says only that the change in energy is equal to the heat source minus the divergence of the diffusive heat flux.

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

where c_p is the heat capacity of the fluid, and ρ is fluid density as before. The expression within the brackets is the heat flux vector, and Q represents a source term. The heat flux vector contains a diffusive and a convective term, where the latter is proportional to the velocity field $\mathbf{u}$.

In this model, the above equations are coupled through the **F** and Q terms. First, add free convection to the momentum balance with the Boussinesq approximation. In this approximation, variations in density with temperature are ignored, except insofar as they give rise to a buoyancy force lifting the fluid. This force will be put in the **F**-term in the Navier-Stokes equations. See further "Marangoni Convection" on page 2-321 in the *Model Library.*

At the same time, the velocity field must be accounted for in the Heat equation. Instead of applying the heat equation as it stands, you can put the divergence of the convective heat flux into the Q coefficient, using the fact that the velocity field is divergence free. This puts the equation on a form that can be used in FEMLAB's Heat transfer application mode.

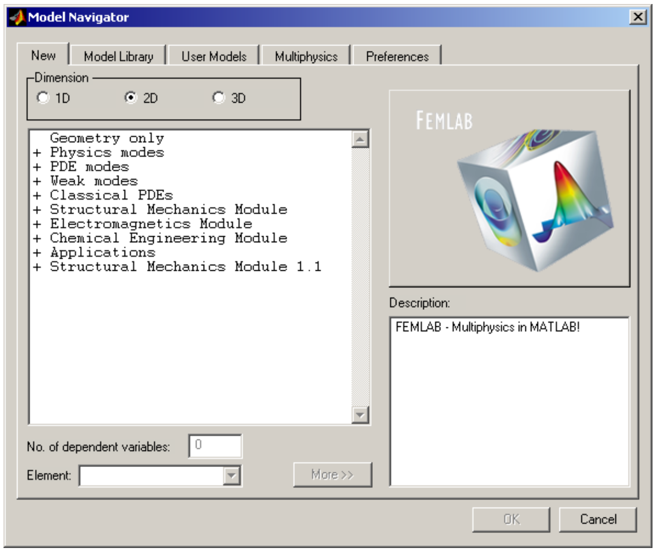
SETTING UP THE MODEL IN FEMLAB

Let's begin the modeling process.

- To do so on any platform, type the following entry into the MATLAB command window:

  ```
  femlab
  ```

This command starts FEMLAB and opens the Model Navigator. On a Windows PC you can also begin a FEMLAB session from the **Start** menu.



```
  Geometry only
+ Physics modes
+ PDE modes
+ Weak modes
+ Classical PDEs
+ Structural Mechanics Module
+ Electromagnetics Module
+ Chemical Engineering Module
+ Applications
+ Structural Mechanics Module 1.1
```

Dimension: 1D [✓ 2D] 3D

Description: FEMLAB - Multiphysics in MATLAB!

No. of dependent variables: 0

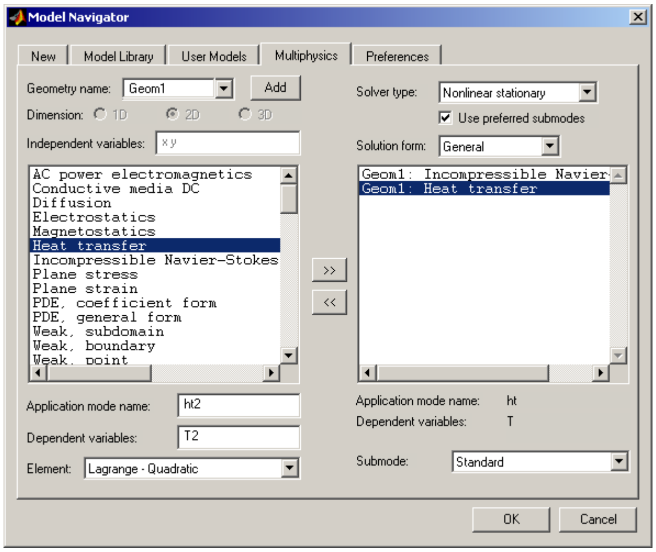
The **Model Navigator** is a multi-purpose dialog box in which you control the general settings of a FEMLAB session. You can find detailed information about the options in the section "The Model Navigator" on page 1-63 in this book. For now, let us focus on one of its five tabbed pages: the **Multiphysics** page lets you select application modes, dimension and other settings for your multiphysics model. The **Model Library** page loads complete models supplied with FEMLAB.

Note A complete version of this model can be found on the **Model Library** page under `FEMLAB/Multiphysics/free_convection`.

- Go to the **Multiphysics** page, and check that the **2D** radio button is selected. The space dimension must always be selected first, as the **Physics modes** available differ.
- Select the **Incompressible Navier-Stokes** application mode in the list on the left. As you can see, the default **Application mode name** is `ns`, and the default names of the **Dependent variables** are, `u`, `v` and `p`. These names can be changed, for example, if they conflict with variables in other application modes. The default finite elements

for the Incompressible Navier-Stokes application mode are mixed second and first order triangular *Lagrange elements*, so called **Lagrange - p2-p1** elements.

- Press the right arrow symbol (**>>**) in the center of the page. This accepts the settings in the **Application mode name, Dependent variables** and **Element** fields, and transfers the Incompressible Navier-Stokes mode to the list of active modes on the right.
- Then select the **Heat transfer** application mode from the list of available modes. Accept the default variable name, `T`, and elements—second order Lagrange elements—by again pressing the right arrow button.



Geometry name: Geom1
Dimension: 1D [✓ 2D] 3D
Independent variables: x y

```
AC power electromagnetics
Conductive media DC
Diffusion
Electrostatics
Magnetostatics
Heat transfer
Incompressible Navier-Stokes
Plane stress
Plane strain
PDE, coefficient form
PDE, general form
Weak, subdomain
Weak, boundary
Weak, point
```

Application mode name: ht2
Dependent variables: T2
Element: Lagrange - Quadratic

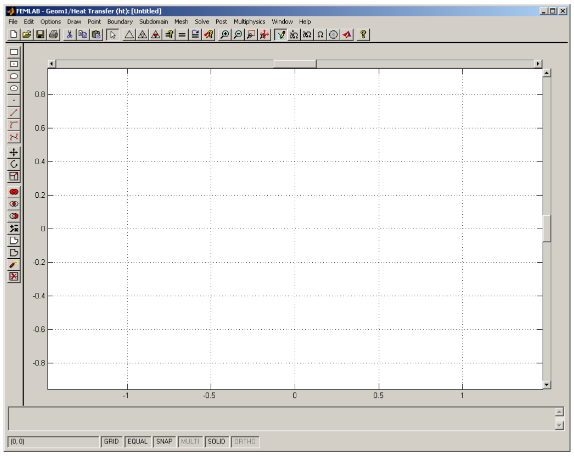
Solver type: Nonlinear stationary
[✓] Use preferred submodes
Solution form: General

```
Geom1: Incompressible Navier-
Geom1: Heat transfer
```

Application mode name: ht
Dependent variables: T
Submode: Standard

- Press **OK** to confirm your choices and close the **Model Navigator**.

At this point, the graphical interface opens up in the Heat transfer application mode. You can always find out in which application mode the package is presently running because its name appears on the bar at the very top of the FEMLAB window.


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1
(0, 0)
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

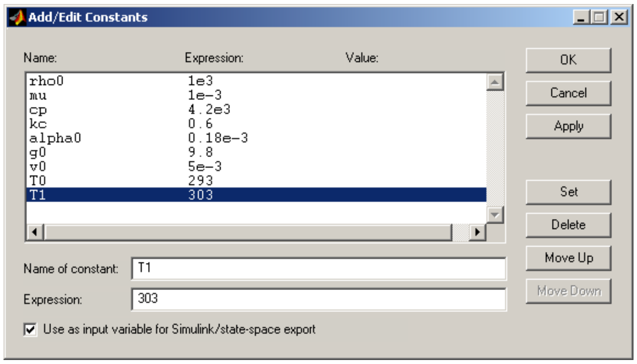
Note Because FEMLAB runs within the MATLAB environment, you always have access to the MATLAB Command Window except when certain modal dialog boxes are open. When referring to the prompt in MATLAB's Command Window, this documentation set uses the term "MATLAB prompt".

Options and Settings

Later in this model you'll need the physical properties of the fluid (water), the temperatures at the inlet and on the surface of the heating tubes, and the inlet velocity. This data can be entered as *constants* in the **Add/Edit Constants** dialog box. The values used in this model are all given in SI units.

- Select the **Options** menu at the top of the FEMLAB window and choose **Add/Edit Constants**.
- First add the density of the fluid: Enter `rho0` in the **Name of constant** field. Press page to move the cursor to the **Expression** field and enter `1e3`. The value is saved when you press `enter`, push the **Set** button, or otherwise leave the **Expression** field.
- Go on adding the dynamic viscosity, `mu=1e-3`; the heat capacity, `cp=4.2e3`; the thermal conductivity, `kc=0.6;` and the volume expansion coefficient,

alpha0=0.18e-3. Then enter the acceleration of gravity, g0=9.8, and the inlet velocity, v0=5e-3.



Name:	Expression:	Value:
rho0	1e3	
mu	1e-3	
cp	4.2e3	
kc	0.6	
alpha0	0.18e-3	
g0	9.8	
v0	5e-3	
T0	293	
T1	303	

Name of constant: T1

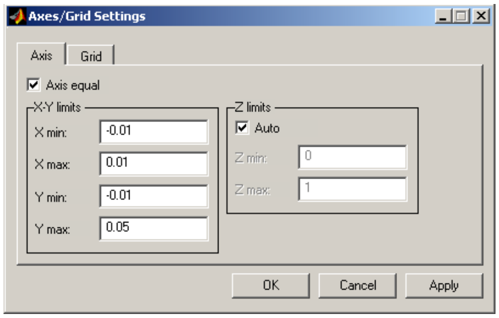
Expression: 303

☑ Use as input variable for Simulink/state-space export

- Finally add the temperature at the inlet and on the heaters, T0=293 and T1=303, respectively, and press **OK**.

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 you're modeling in SI units). Before you start drawing the geometry, you therefore have to change the size of the visible drawing area and the grid spacing.

- On the menu bar at the top of the screen select the **Options** menu, and from the drop-down choices choose **Axes/Grid Settings...**



☑ Axis equal

X-Y limits

X min: -0.01

X max: 0.01

Y min: -0.01

Y max: 0.05

Z limits

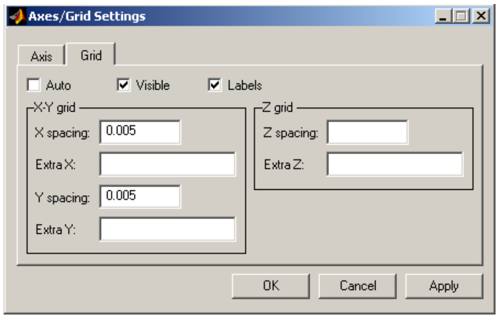
☑ Auto

Z min: 0

Z max: 1

- A dialog box opens up, and on the **Axis** page enter -0.01, 0.01, -0.01, and 0.05 as **X min**, **X max**, **Y min**, and **Y max**, respectively.

- Click the page for the **Grid** page and deselect the **Auto** check box. Enter `0.005` as both **X spacing** and **Y spacing**.


Axes/Grid Settings

Axis | Grid

[ ] Auto [✓] Visible [✓] Labels

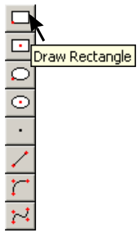
X-Y grid
X spacing: 0.005
Extra X:
Y spacing: 0.005
Extra Y:

Z grid
Z spacing:
Extra Z:

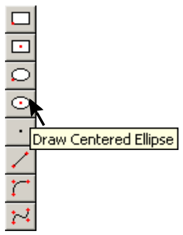
- Press **OK** to close the dialog box and to apply the settings.

Draw Mode

Your next task is to draw the model's geometry. This is easy, involving only the rectangle and circle primitives, and subtracting one from the other.

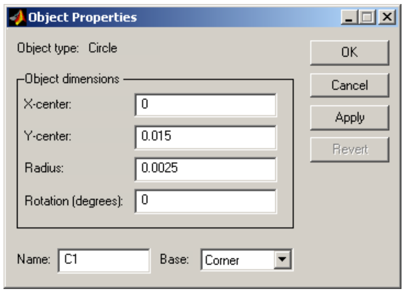


- First draw a rectangle of width 0.005 and height 0.04, with the lower left corner at the origin: Press the **Draw Rectangle** draw toolbar button. It is the first button on the draw toolbar, which is located on the left side of the drawing area. Then click at (0,0), using the left mouse button, and drag the mouse—keeping the button down—to (0.005,0.04). Release the button.



- Next, draw a circle with radius 0.005 centered at (0,0.015): Press the fourth button on the draw toolbar, **Draw Centered Ellipse**. 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 ensures that a circle, and not an ellipse, is drawn.

- The desired radius of the circle is, however, not 0.005, but 0.0025. To fix this, double click the circle to open the **Object Properties** dialog box. Enter `0.0025` as **Radius** and press **OK**.



Object Properties

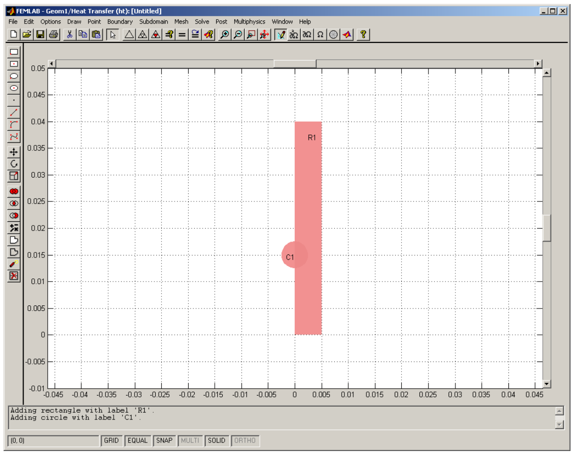
Object type: Circle

Object dimensions

X-center:	0
Y-center:	0.015
Radius:	0.0025
Rotation (degrees):	0

Name: C1 Base: Corner

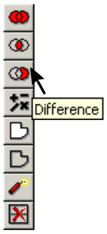
- Select both the circle and the rectangle. Either you can draw a rubber band box around both them, or you simply press **Ctrl-a** to quickly select all objects.

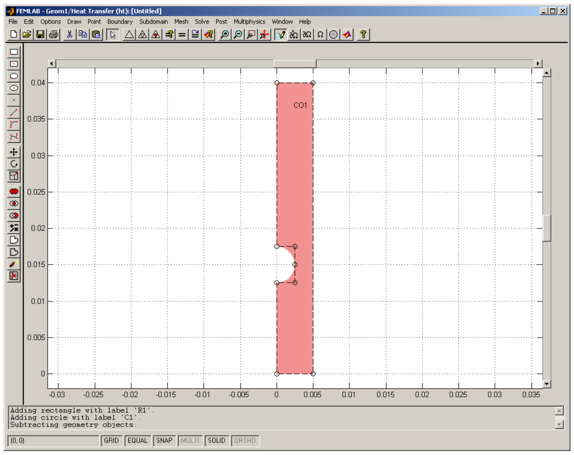


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]

```
Adding rectangle with label 'R1'.
Adding circle with label 'C1'.
```

- Press the **Difference** button on the draw toolbar and then the **Zoom Extents** button on the main toolbar. The last button modifies the axes settings so that everything fits nicely into the drawing area.




FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
(0, 0)
GRID EQUAL SNAP MULTI SOLID ORTHO

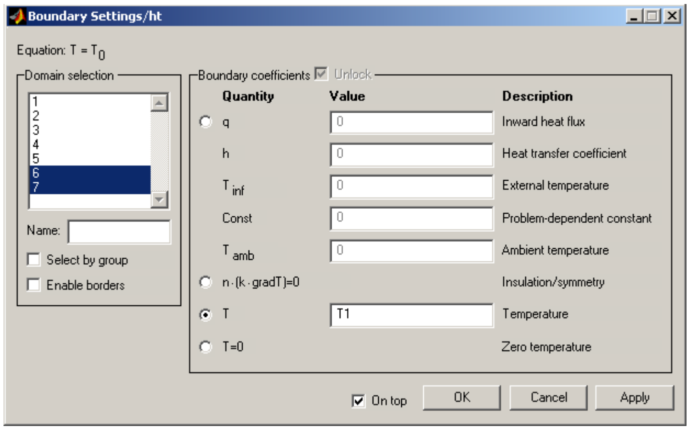
```
Adding rectangle with label 'R1'.
Adding circle with label 'C1'.
Subtracting geometry objects.
```

Boundary Mode

Now that the geometry is ready, you can start defining your physics. The boundary conditions and equation coefficients are set independently for the two application modes. First, set the boundary condition for the Heat transfer mode, and then for the Incompressible Navier-Stokes mode.

- Choose **Boundary Settings...** from the **Boundary** menu. This opens the **Boundary Setting** dialog box, and transfers FEMLAB to *boundary mode*. The dialog box has different options in different application modes.
- Start by insulating all boundaries. Select all boundaries in the **Domain selection** list by selecting number 1 in the list on the left side of the Boundary Settings dialog box and then number 7 holding the **Shift** key pressed. Now, insulate all boundaries by clicking the **Insulation/symmetry** radio button. For more information on how to select more than one object in the user interface, see “Object Selection Methods in 2D” on page 1-80 in this book.
- Select the inflow boundary by clicking at the corresponding edge in the geometry, or by selecting boundary number 2 in the list. Click the **Temperature** radio button and change the **Temperature** field to read `T0`.

- Select boundaries 6 and 7, that is, the heater, and set the **Temperature** field to `T1`.



Boundary Settings/ht

Equation: $T = T_0$

Domain selection: 1, 2, 3, 4, 5, [✓ 6], [✓ 7]

Name:

☐ Select by group

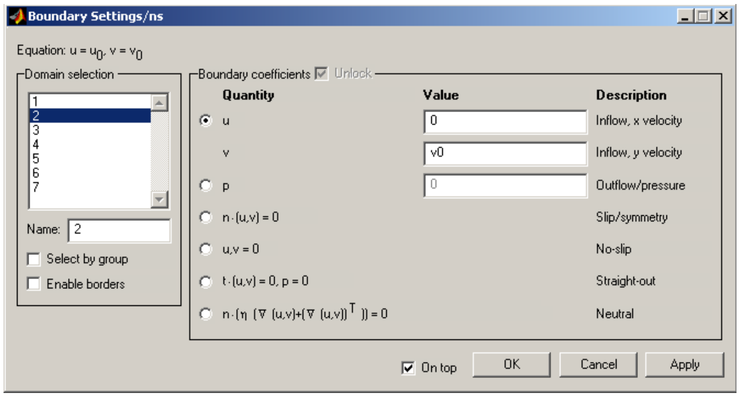
☐ Enable borders

Boundary coefficients ☑ Unlock

Quantity	Value	Description
○ q	0	Inward heat flux
h	0	Heat transfer coefficient
T_{inf}	0	External temperature
Const	0	Problem-dependent constant
T_{amb}	0	Ambient temperature
○ n·(k·gradT)=0		Insulation/symmetry
● T	T1	Temperature
○ T=0		Zero temperature

☑ On top · OK · Cancel · Apply

- Press **Apply**, to confirm the settings for the Heat transfer application mode. Then switch to Incompressible Navier-Stokes, by selecting this mode from the bottom of the **Multiphysics** menu. The appearance of the dialog box changes to reflect the new mode.
- Again, start by selecting all boundaries and click the **Slip/symmetry** condition.
- For the inflow boundary, select boundary 2, click the first radio button and set **Inflow, y velocity** to `v0`. Let the *x* velocity be `0`.



Boundary Settings/ns

Equation: $u = u_0, v = v_0$

Domain selection: 1, [✓ 2], 3, 4, 5, 6, 7

Name: 2

☐ Select by group

☐ Enable borders

Boundary coefficients ☑ Unlock

Quantity	Value	Description
● u	0	Inflow, x velocity
v	v0	Inflow, y velocity
○ p	0	Outflow/pressure
○ n·(u,v) = 0		Slip/symmetry
○ u,v = 0		No-slip
○ t·(u,v) = 0, p = 0		Straight-out
○ $n\cdot(\eta\ (\nabla\ (u,v)+(\nabla\ (u,v))^T\ )) = 0$		Neutral

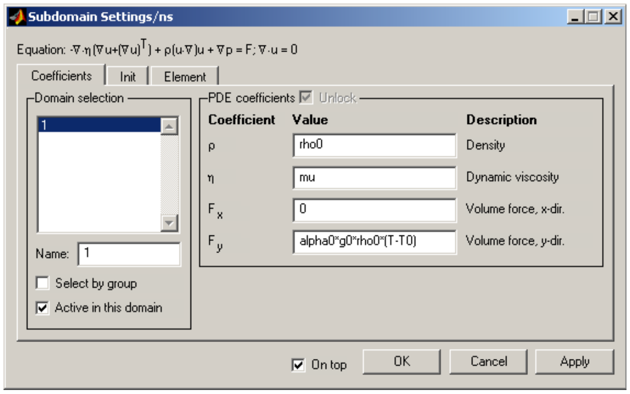
☑ On top · OK · Cancel · Apply

- Continue by selecting the outflow boundary, boundary number 4, and click the **Outflow/pressure** radio button. Leave the pressure, **p**, at `0`.
- Finally, select boundaries 6 and 7 and click the **No-slip** condition. Press **OK** to confirm your choises and close the dialog box.

Subdomain Mode

The coefficients in the governing equations can be interpreted, depending on the application mode, as, for example, material properties, forces and sources. You specify all of them in subdomain mode. This time, start with the Incompressible Navier-Stokes application mode, and continue on to the Heat transfer mode.

- Choose **Subdomain Settings** from the **Subdomain** menu to open the **Subdomain Settings** dialog box, and put the user interface in subdomain mode.
- Select the single subdomain, number 1, and use the already defined constants to set the density and the viscosity. Enter `rho0` and `mu` in the **Density** and **Dynamic viscosity** fields, respectively.



Subdomain Settings/ns

Equation: $-\nabla\cdot\eta(\nabla u+(\nabla u)^T)+\rho(u\cdot\nabla)u+\nabla p=F;\ \nabla\cdot u=0$

Domain selection: 1; Name: 1; Select by group: unchecked; Active in this domain: checked

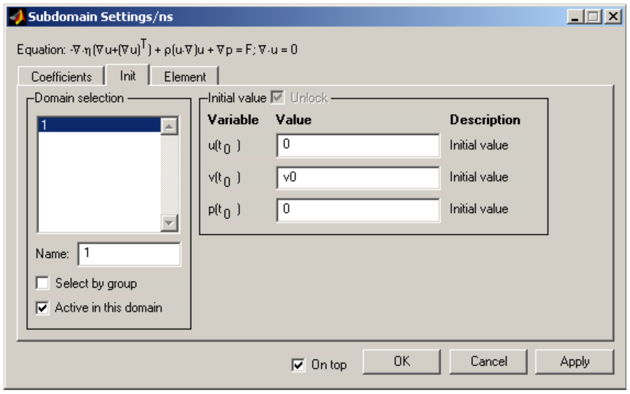
PDE coefficients

Coefficient	Value	Description
ρ	rho0	Density
η	mu	Dynamic viscosity
F_x	0	Volume force, x-dir.
F_y	alpha0*g0*rho0*(T-T0)	Volume force, y-dir.

- The effect of temperature on the density of the fluid is entered only as a buoyancy force in the y direction. Set the **Volume force, y-dir.** field to `alpha0*g0*rho0*(T-T0)`. Remember that *T* is the dependent variable from the Heat transfer application mode.

Initial values are also set in the **Subdomain Settings** dialog box. The Navier-Stokes equations are nonlinear, and therefore benefit from an educated guess as initial solution to the nonlinear solver. Sometimes, a good initial value is even necessary for convergence. The initial values are also used as initial condition for the time-dependent solver.

- Click the **Init** page in the **Subdomain Settings** dialog box to switch to the **Init** page. Set the initial value **V(t_0)** to `v0`.



Subdomain Settings/ns

Equation: $-\nabla\cdot\eta(\nabla u+(\nabla u)^T)+\rho(u\cdot\nabla)u+\nabla p=F;\ \nabla\cdot u=0$

Coefficients | Init | Element

Domain selection: 1

Name: 1

☐ Select by group

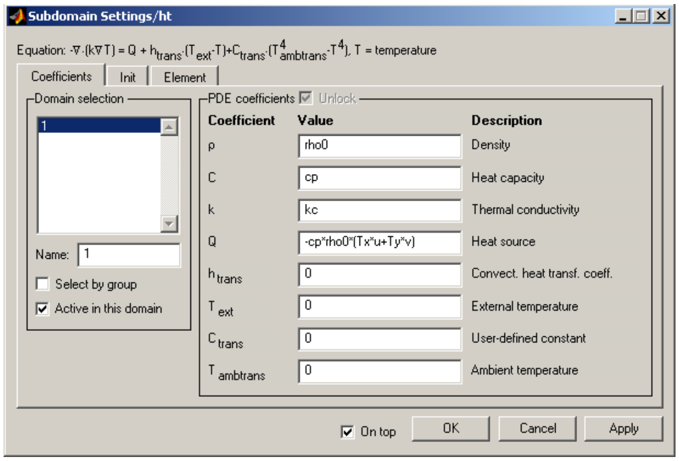
☑ Active in this domain

Initial value ☑ Unlock

Variable	Value	Description
$u(t_0)$	0	Initial value
$v(t_0)$	v0	Initial value
$p(t_0)$	0	Initial value

☑ On top | OK | Cancel | Apply

- Press **Apply** to confirm the settings and use the **Multiphysics** menu to switch to the Heat transfer application mode. The open dialog box returns automatically to the **Coefficients** page, but now for Heat transfer mode.



Subdomain Settings/ht

Equation: $-\nabla\cdot(k\nabla T)=Q+h_{trans}\cdot(T_{ext}-T)+C_{trans}\cdot(T^4_{ambtrans}-T^4)$, T = temperature

Coefficients | Init | Element

Domain selection: 1

Name: 1

☐ Select by group

☑ Active in this domain

PDE coefficients ☑ Unlock

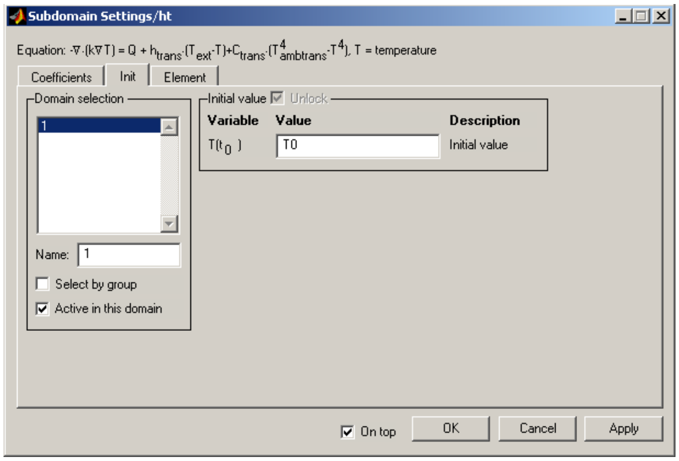
Coefficient	Value	Description
ρ	rho0	Density
C	cp	Heat capacity
k	kc	Thermal conductivity
Q	-cp*rho0*(Tx*u+Ty*v)	Heat source
h_{trans}	0	Convect. heat transf. coeff.
T_{ext}	0	External temperature
C_{trans}	0	User-defined constant
$T_{ambtrans}$	0	Ambient temperature

☑ On top | OK | Cancel | Apply

- Set the **Density**, **Heat capacity** and **Thermal conductivity** to `rho0`, `cp` and `kc`, respectively.
- There is no distributed heat source in this model, but the convective transport of heat acts as one. To account for the divergence of the convective heat flux, the term `-cp*rho0*(Tx*u+Ty*v)` should be entered into the **Heat source** field.

`Tx` and `Ty` in the above expression are the partial space derivatives of the temperature, *T*. As you can see, not only the dependent variables, but also their derivatives, are available when entering, for example, nonlinear material properties and source terms. In fact, you can use any valid MATLAB expression containing, for example, the dependent variables, any user defined constants and expressions, the space coordinates, *x*, *y* and *z*, and time, *t*.

- The temperature also needs an initial value: Go to the **Init** page and enter `T0` in the **T(t$_0$)** field. Then press **OK** to confirm all settings and close the dialog box.


Subdomain Settings/ht

Equation: $-\nabla\cdot(k\nabla T) = Q + h_{trans}\cdot(T_{ext}-T)+C_{trans}\cdot(T^4_{ambtrans}-T^4)$, T = temperature

Initial value: Unlock

Variable	Value	Description
$T(t_0)$	T0	Initial value

Name: 1

☐ Select by group

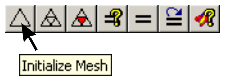
☑ Active in this domain

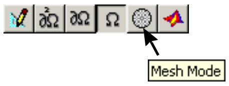
Mesh Mode

Because FEMLAB is based on the finite element method (FEM), it needs a subdivision of the geometry known as a mesh. A standard mesh is created automatically, as soon as you enter *mesh mode*. If you need a different mesh resolution, or require the mesh to be denser in some parts of the geometry than in others, you can work with the **Mesh Parameters** dialog box, accessible from the **Mesh** menu.

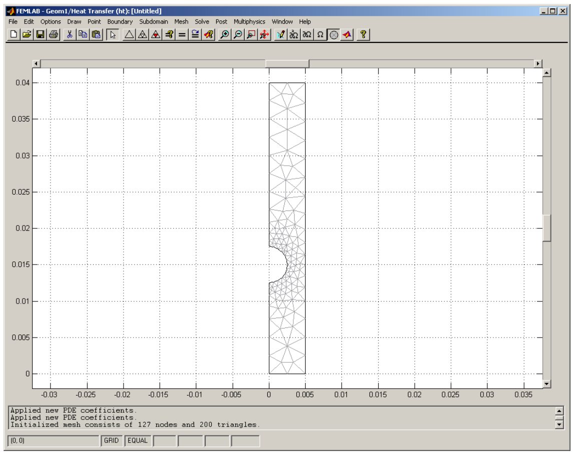
If you, on the other hand, trust the default settings, you can proceed directly to solving the model. A mesh is then created automatically when you press the **Solve**

Problem button. It is, however, often preferable to inspect the mesh before solving, as that can give you a hint as to how long time it will take to solve.





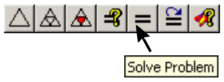
- Initialize the default mesh by pressing either the **Initialize Mesh** button or the **Mesh Mode** button, both on the main toolbar. Both in this case have the same effect.


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
-0.03 -0.025 -0.02 -0.015 -0.01 -0.005 0 0.005 0.01 0.015 0.02 0.025 0.03 0.035
(0, 0)
GRID
EQUAL

```
Applied new PDE coefficients.
Applied new PDE coefficients.
Initialized mesh consists of 127 nodes and 200 triangles.
```

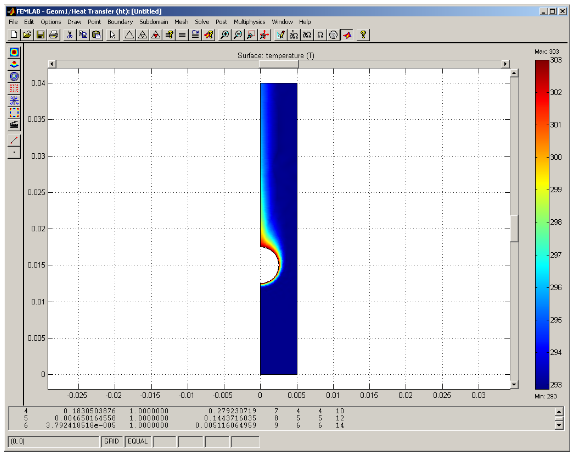
As you can see, a rather coarse mesh has been created. This means that you can obtain a first solution quickly, to check that your model has been entered correctly into the program. Later, you can return to mesh mode to refine the mesh. If solving again using a finer mesh gives the same result, you can be assured that the mesh resolution is sufficient, otherwise you'll have to refine the mesh again.

Solving



The required nonlinear solver is default, so all you have to do is press the **Solve Problem** button on the main toolbar.

- Press the **Solve Problem** button. Solving this problem takes 19 seconds on a 1.5-GHz Pentium 4 machine running Windows 2000.

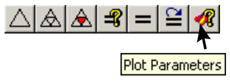

FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Surface: temperature (T)
Max: 303
Min: 293
GRID
EQUAL
(0, 0)

```
   4        0.1830503876   1.0000000          0.279230719    7    4    4   10
   5      0.004650164558   1.0000000         0.1443716035    8    5    5   12
   6    3.792418518e-005   1.0000000       0.005116064959    9    6    6   14
```

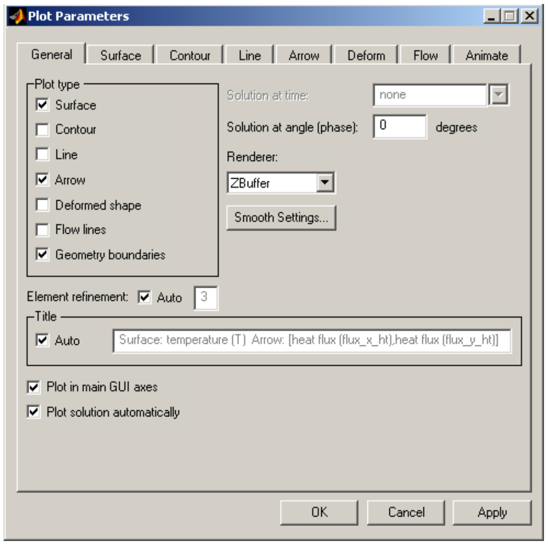
As soon as the solution is ready, a default plot is displayed. In this case you'll get a surface plot of the temperature. If you do not get a similar plot, go back and check the boundary conditions and subdomain settings.

Post Mode

In *post mode*, you can, for example, add additional plot types, and set parameters for the different plots. The postprocessing utilities can visualize any valid MATLAB expression, containing, for example, the solution variables, their derivatives and the space coordinates. Many frequently used expressions are predefined as *postprocessing variables*, directly available from drop-down lists in the **Plot Parameters** dialog box.



- Press the **Plot Parameters** button on the main toolbar. This opens the **Plot Parameters** dialog box on the general page.



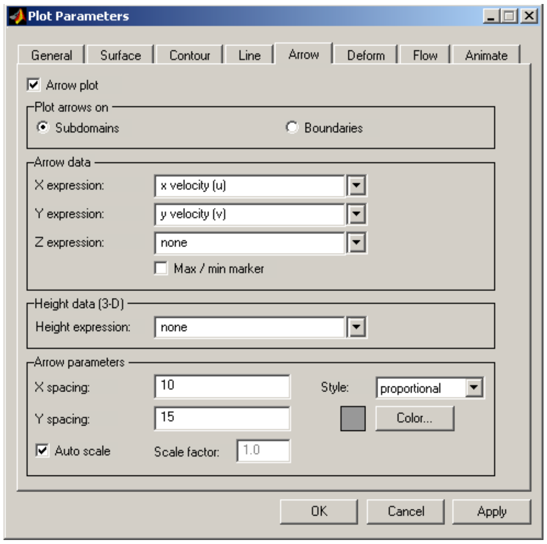
Plot type
- [✓] Surface
- [ ] Contour
- [ ] Line
- [✓] Arrow
- [ ] Deformed shape
- [ ] Flow lines
- [✓] Geometry boundaries

Solution at time: none
Solution at angle (phase): 0 degrees
Renderer: ZBuffer
Element refinement: [✓] Auto 3
Title
[✓] Auto Surface: temperature (T) Arrow: [heat flux (flux_x_ht),heat flux (flux_y_ht)]
[✓] Plot in main GUI axes
[✓] Plot solution automatically

On the **General** page, you can, for example, choose **Plot type** and **Element refinement** level. The latter property decides how many times the FEM mesh is refined for visualization purposes. A higher level of refinement gives a smoother plot when using higher-order elements, but you pay in increasing memory usage. In 2D, you can normally increase the refinement level, compared to the **Auto** setting, to get a better looking plot. In 3D, on the other hand, you might need to decrease the refinement in order not to run out of memory.

- Select plot types **Surface** and **Arrow** by clicking the corresponding check boxes.

- Click the **Arrow** page and select `x velocity (u)` and `y velocity (v)` in the **X expression** and **Y expression** drop-down lists, respectively.



[✓] Arrow plot

Plot arrows on
[● Subdomains] ○ Boundaries

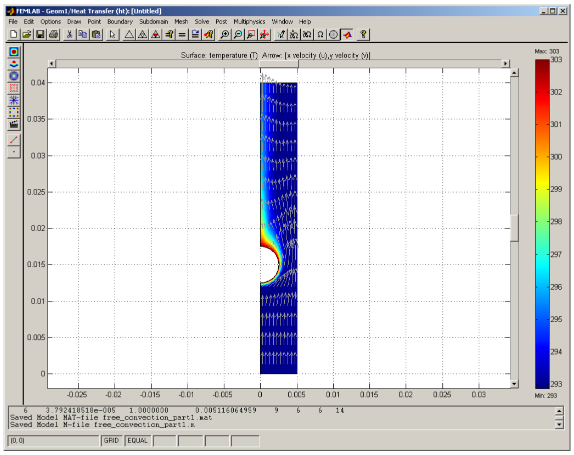
Arrow data
X expression: x velocity (u)
Y expression: y velocity (v)
Z expression: none
☐ Max / min marker

Height data (3-D)
Height expression: none

Arrow parameters
X spacing: 10
Style: proportional
Y spacing: 15
Color...
[✓] Auto scale
Scale factor: 1.0

- Finally, set **X spacing** and **Y spacing** to `10` and `15` respectively, and choose a suitable arrow color, preferably grey. Press **OK**.

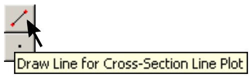
Without heating, one would expect an exit *y*-velocity that is slightly lower towards the left side, behind the heating tube (wake-effect). However, in this case you see that the *y*-velocity is higher on the left. This is because of the buoyancy effect of the free convection.


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Surface: temperature (T) Arrow: [x velocity (u),y velocity (v)]
Max: 303
Min: 293
(0, 0)
GRID
EQUAL

```
   6    3.792418518e-005   1.0000000      0.005116064959    9    6    6   14
Saved Model MAT-file free_convection_part1.mat
Saved Model M-file free_convection_part1.m
```

Advanced Postprocessing

In FEMLAB, it's very simple to obtain separate plots of the solution on various geometry cross sections. The easiest way is to use the **Draw Line for Cross-Section Line Plot** button, the second to last button at the bottom of the draw toolbar.

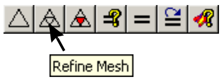


- To show the temperature profile just downstream of the heater, first press the **Draw Line for Cross-Section Line Plot** button.
- Point the mouse at (0,0.02); press the left mouse button; drag the mouse to (0.005,0.02) and release the button. The desired plot appears automatically in a separate MATLAB window.

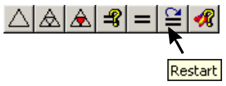
Leave the window open, as you'll need it further on. Separate figure windows opened by FEMLAB can be handled just like any other MATLAB figures. You can, for example, use the tools on the figure's toolbar to add text, and then print or save the result in any format supported by MATLAB's `print` command. Type `help print` at the prompt, or see the MATLAB documentation, for more information on printing.

Checking for Convergence

It is important to check that the solution obtained is indeed close to the best solution obtainable from the given mathematical model and discretization scheme, and thereby hopefully close to the “true” solution. This you can do by simply refining the mesh, solving again, and comparing the solutions.



- Press the **Refine Mesh** button in the main toolbar. This gives you a uniform refinement of the mesh, meaning that every triangle is split into four.



- To obtain the new solution, press the **Restart** button, also on the main toolbar. When you press the **Restart** button instead of the **Solve Problem** button, the last solution is used as initial data for the nonlinear solver. As the solution isn’t expected to change much, convergence is fast.

Visually, the solution hardly changes at all. To verify that this is indeed the case, you can plot the same cross section as above in the same window. But, first you have to tell MATLAB to draw on top of the existing plot, instead of deleting it.

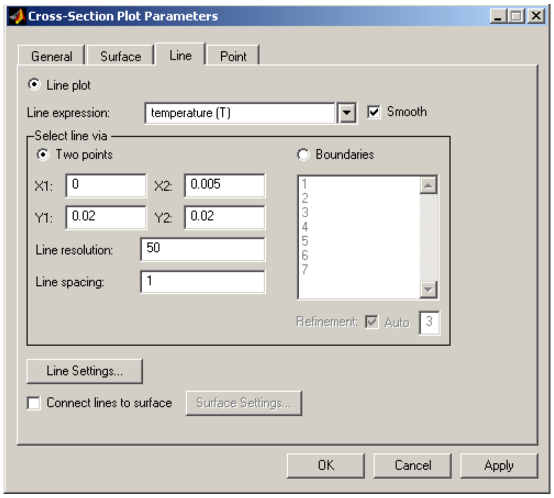
- Go to the MATLAB prompt and write

  ```
  figure(1)
  hold on
  ```

  The first line activates the figure containing the cross section, and the second tells MATLAB to overprint, rather than delete, any existing graphics. Note that if you have other figure windows opened the number of the figure window, here assumed to be 1, might need to be changed.

This time, you can use the **Cross-Section Plot Parameters** dialog box to change the line color of the cross-section plot. Using this dialog box also ensures that exactly the same cross section is plotted.

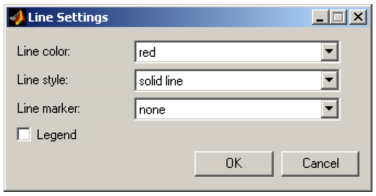
- Open the **Cross-Section Plot Parameters** dialog box by choosing **Cross-Section Plot Parameters...** from the **Post** menu.



Line plot: selected
Line expression: temperature (T) — Smooth: ✓
Select line via: Two points (selected) / Boundaries
X1: 0 X2: 0.005
Y1: 0.02 Y2: 0.02
Line resolution: 50
Line spacing: 1
Boundaries: 1 2 3 4 5 6 7
Refinement: Auto ✓ 3
Connect lines to surface: ☐

- Click the **Line** tab to reach the **Line** page, and then press the **Line Settings...** button.

- Change the **Line color** to `red`, selecting from the drop-down list in the **Line Settings** dialog box.


Line Settings

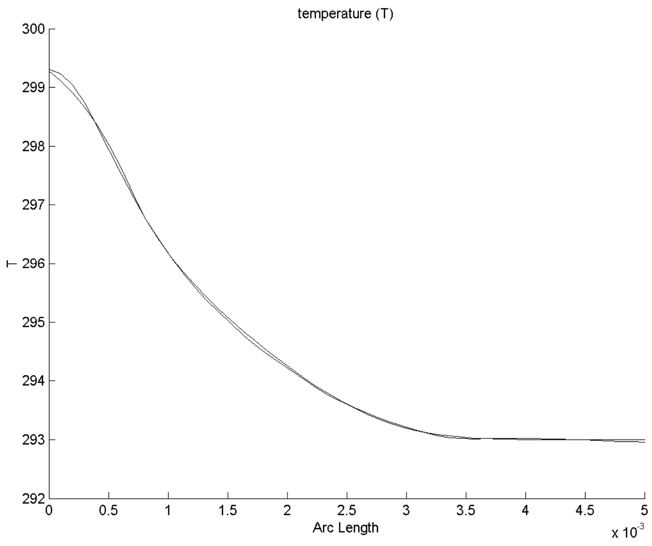
Line color: red

Line style: solid line

Line marker: none

☐ Legend

- Press **OK** twice to close both dialog boxes and display the new temperature profile as a red line.



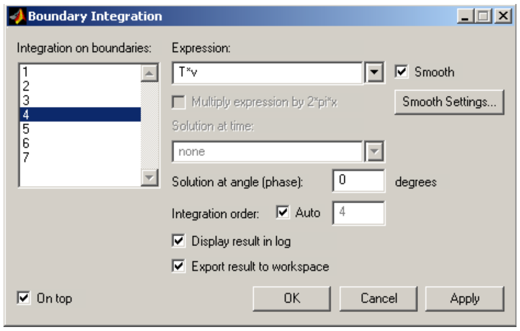
As you can see, the temperature profile from the solution on the refined mesh lies almost on top of the profile for the coarser mesh. Therefore, you can conclude that even the coarser mesh probably gives sufficient accuracy.

Integrating to Find the Mean Temperature

Since both the velocity and the temperature vary along the outlet at the top, advanced postprocessing must be employed to get the mean exit temperature. 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 properly mixed. For this 2D example, the cup mixing temperature is given by the following expression on boundary 4 (the outlet):

$$\langle T \rangle = \frac{\int T v \, dx}{\int v \, dx}$$

- To obtain the numerator, select **Boundary integration** from the **Post** menu, select boundary 4, and type `T*v` in the **Expression** field.



Boundary Integration

Integration on boundaries: 1, 2, 3, 4 (selected), 5, 6, 7

Expression: T*v

[✓] Smooth

[ ] Multiply expression by 2*pi*x

Solution at time: none

Solution at angle (phase): 0 degrees

Integration order: [✓] Auto 4

[✓] Display result in log

[✓] Export result to workspace

[✓] On top

- Click the **Export result to workspace** check box and press **Apply**. The numerator is now present in the MATLAB workspace as a variable called `ans`. The `ans` variable always contains the result of the last calculation.
- Save the numerator for later use by typing

  ```
  Itv = ans
  ```

  at the MATLAB prompt.
- To obtain the denominator, type `v` in the **Expression** field in the **Boundary Integration** dialog box and press **OK**.
- Finally, save also the denominator under a different name, and calculate the cup mixing temperature. Write

  ```
  Iv = ans
  Tmean = Itv/Iv
  ```

  at the prompt. The result is displayed immediately.

You should get a mean temperature of **293.8** K, that is, a temperature rise of approximately 0.8 K between inlet and outlet.

Saving

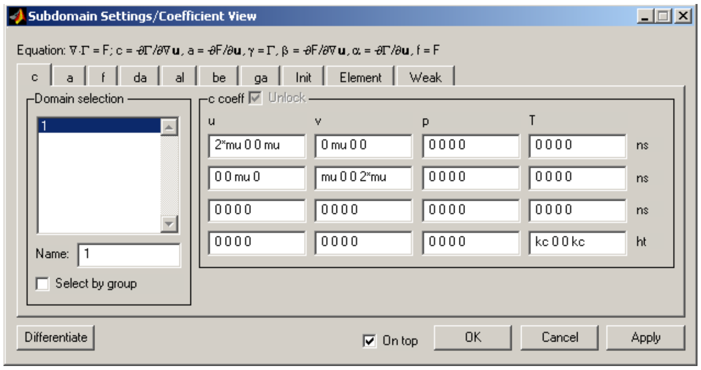
- To save the model to a file, go to the **File** menu on the menu bar at the top, then select **Save As** and **Model MAT-file**. This selection saves the model in a binary format with the extension `.mat`, a format you can load directly into the graphical interface.

It is also possible to save the model as a MATLAB script file for parameterized models and optimal design by selecting **Save As** and **Model M-file...**. A complete description of the Model M-file concept appears in this book in the section "Model M-files" on page 1-51. Use the *Model MAT-file* format to save entire FEMLAB sessions if you intend to resume the modeling later on.

Boundary and PDE Coefficient Views

As you might have noticed, this model doesn't require you to handle a PDE directly. When modeling in a physics application mode, such as the Heat transfer mode or the Incompressible Navier-Stokes mode in this example, you can reach the underlying PDE formulation by selecting **View as Boundary Coefficients** and **View as PDE Coefficients** from the **Boundary** and **Subdomain** menus, respectively. Say, for instance, you'd like to view the PDE coefficients corresponding to the material settings:

- Go to the **Subdomain** menu and choose **View as PDE Coefficients**. Then return to the same dropdown menu and open the **Subdomain Settings** dialog box.



Equation: $\nabla\cdot\Gamma = F$; $c = -\partial\Gamma/\partial\nabla \mathbf{u}$, $a = -\partial F/\partial \mathbf{u}$, $\gamma = \Gamma$, $\beta = -\partial F/\partial\nabla \mathbf{u}$, $\alpha = -\partial\Gamma/\partial \mathbf{u}$, $f = F$

Domain selection: 1

Name: 1

Select by group

u	v	p	T	
2*mu 0 0 mu	0 mu 0 0	0 0 0 0	0 0 0 0	ns
0 0 mu 0	mu 0 0 2*mu	0 0 0 0	0 0 0 0	ns
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	ns
0 0 0 0	0 0 0 0	0 0 0 0	kc 0 0 kc	ht

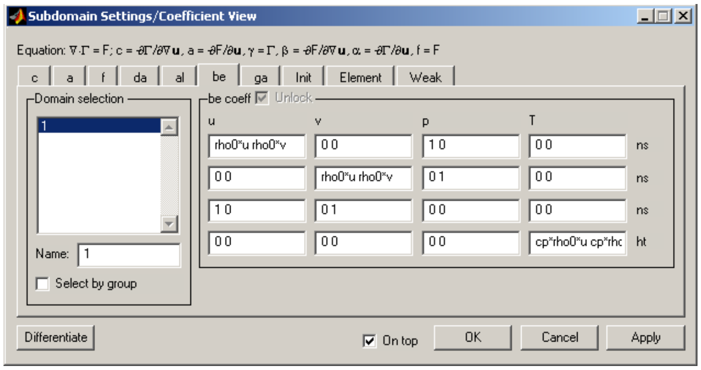
The underlying equation appears in the upper left corner, and the edit fields to the right now correspond to equation coefficients rather than material properties. The interpretation of the coefficients depend on whether FEMLAB currently uses the *coefficient form* or the *general form* to describe a generic PDE. See "Overview of PDE

Models" on page 1-336 in this manual. Multiphysics problems should normally be based on the general form, and it's this formulation that you can see at the top of the dialog box.

For a multiphysics model, the coefficient view shows the complete system of equations. It is important to understand that FEMLAB solves multiphysics problems directly as *one* PDE system. In general form, the *stiffness matrix* or *Jacobian* of the complete system is obtained automatically through symbolic differentiation. The result of this automatic manipulation can be seen on the **c**, **a**, **al** and **be** tabs.

Let us have a closer look at, for example, the β coefficient. In this case β is obtained through automatic differentiation of the *F* coefficient. β is usually identified as a convection coefficient and then contains a transport field.

- Click the **be** tab to show the convection coefficient. Each field contains a MATLAB vector of length two, making β in this case a four-by-four-by-two tensor.



Subdomain Settings/Coefficient View

Equation: $\nabla\cdot\Gamma = F$; $c = -\partial\Gamma/\partial\nabla\mathbf{u}$, $a = -\partial F/\partial\mathbf{u}$, $\gamma = \Gamma$, $\beta = -\partial F/\partial\nabla\mathbf{u}$, $\alpha = -\partial\Gamma/\partial\mathbf{u}$, $f = F$

Domain selection: 1

Name: 1

☐ Select by group

be coeff ☑ Unlock

u	v	p	T	
rho0*u rho0*v	0 0	1 0	0 0	ns
0 0	rho0*u rho0*v	0 1	0 0	ns
1 0	0 1	0 0	0 0	ns
0 0	0 0	0 0	cp*rho0*u cp*rhc	ht

☑ On top

Look at the coefficient in the lower right edit field, `cp*rho0*u cp*rho0*v`. This is the transport field for heat transport in the fourth equation, the heat equation. As you can see, the transport field is the fluid velocity field multiplied by the specific heat per unit volume.

In **Coefficient View**, you've also now got an extra tab, named **Weak**, where you can specify additional terms to your equations on weak form. Weak coefficients can be added on all levels—subdomain, boundary, edge and point—where they can be used to implement, for example, explicit streamline diffusion, weak constraints, line sources and point sources, respectively.

Model M-files

In FEMLAB you can save an entire session performed in the graphical interface as a MATLAB script. This script is called a *Model M-file*. You can modify a sequence of FEMLAB commands for whatever reason and reload them into the graphical interface or execute them from the MATLAB prompt.

The following code listing shows a Model M-file of the model you just developed; this code represents the model if saved directly after solving it the first time. The annotations give an overview of the main parts of a typical Model M-file. Note that information not necessary to grasp the Model M-file concept has been removed from the listing below. This means that the actual listing obtained when saving the model as a Model M-file will be slightly longer.

```
% FEMLAB Model M-file
% Generated 01-Mar-2002 13:14:31 by FEMLAB 2.3.

flclear fem
% FEMLAB Version
clear vrsn;
vrsn.name='FEMLAB 2.3';
vrsn.major=0;
vrsn.build=154;
fem.version=vrsn;

% Recorded command sequence

% New geometry 1
fem.sdim={'x','y'};

% Geometry
clear s c p
R1=rect2(0,0.005,0,0.04);
C1=circ2(0,0.015,0.0025);
CO1=R1-C1;

objs={CO1};
names={'CO1'};
s.objs=objs;
s.name=names;

c.objs={};
c.name={};

p.objs={};
p.name={};

drawstruct=struct('s',s,'c',c,'p',p);
fem.draw=drawstruct;
fem.geom=geomcsg(fem);
```

```
clear appl

% Application mode 1
appl{1}.mode=flpdens2d('dim',{'u','v','p'});
appl{1}.name='ns';
appl{1}.dim={'u','v','p'};
appl{1}.border='off';
appl{1}.form='general';
appl{1}.elemdefault='Lagp2p1';
appl{1}.assign={'rho','rho_ns'};
appl{1}.var={};
appl{1}.shape={'shlag(2,''u'')','shlag(2,''v'')','shlag(1,''p'')'};

% Application mode 2
appl{2}.mode=flpdeht2d('dim',{'T'});
appl{2}.name='ht';
appl{2}.dim={'T'};
appl{2}.border='off';
appl{2}.form='coefficient';
appl{2}.elemdefault='Lag2';
appl{2}.assign={'rho','rho_ht'};
appl{2}.var={};
appl{2}.shape={'shlag(2,''T'')'};
fem.appl=appl;

% Initialize mesh
fem.mesh=meshinit(fem);

% Boundary conditions
clear bnd
bnd.v={'0','v0','0','0'};
bnd.type={'slip','uv','out','noslip'};
bnd.ind=[1 2 1 3 1 4 4];
fem.appl{1}.bnd=bnd;

% Boundary conditions
clear bnd
bnd.T={'0','T0','T1'};
bnd.type={'q0','T','T'};
bnd.ind=[1 2 1 1 1 3 3];
fem.appl{2}.bnd=bnd;

% PDE coefficients
clear equ
equ.rho={'rho0'};
equ.eta={'mu'};
equ.Fy={'alpha0*g0*rho0*(T-T0)'};
equ.init={{{'0'};{'v0'};{'0'}}};
equ.usage={1};
equ.ind=1;
fem.appl{1}.equ=equ;

% PDE coefficients
clear equ
equ.rho={'rho0'};
equ.C={'cp'};
equ.k={{{'kc'}}};
equ.Q={'-cp*rho0*(Tx*u+Ty*v)'};
equ.init={{{'T0'}}};
equ.usage={1};
equ.ind=1;
fem.appl{2}.equ=equ;

% Define constants
fem.const={...
  'rho0',   1000,...
  'mu',     0.001,...
  'cp',     4200,...
  'kc',     0.6,...
  'alpha0', 0.18e-3,...
  'g0',     9.8,...
  'v0',     0.005,...
  'T0',     293,...
  'T1',     303};

% Multiphysics
fem=multiphysics(fem);

% Extend the mesh
fem.xmesh=meshextend(fem);

% Evaluate initial condition
init=asseminit(fem,...
  'init',   fem.xmesh.eleminit);

% Solve nonlinear problem
fem.sol=femnlin(fem,...
  'init',   init,...
  'report', 'on');

% Plot solution
postplot(fem,...
  'geom',   'on',...
  'tridata',{'T','cont','internal'},...
  'tribar','on',...
  'title',  'Surface: temperature (T)')
```

If you intend to load a modified Model M-file into FEMLAB's graphical interface, it's important that you follow the syntactic rules outlined in the section "Model M-file" on page 3-171 in the *Reference Manual*. If you intend to use the modified Model M-file only from the MATLAB prompt, format requirements are less strict.

A Model M-file is useful for many purposes:

- It documents the work you perform within the graphical interface in a human-readable manner.

- Studying the file FEMLAB generates can serve as a tool for learning how to work with FEMLAB functions.
- You can save a Model M-file and modify it for optimal design or for creating parameterized models.
- You can insert any MATLAB command anywhere in the Model M-file as long as they do not affect the FEM structure in any way inconsistent with the syntactic rules, and reload the model into the GUI.
- When working on a complicated model that requires you to write custom functions, you can start modeling in the graphical interface and then save the Model M-file to expand it. Note again, though, that if you work outside the syntactic rules, you can run the resulting model only from the MATLAB prompt.

For more detailed information on this topic see the section "Model M-file" on page 3-171 in the *Reference Manual*.

Time-dependent Simulation

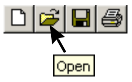
This example is an extension of the stationary free convection model above. In the first step of modeling the heat transfer by forced and free convection a steady state was found. See "Forced and Free Convection Heat Transfer" on page 1-26 on how to create and save the first part of the model.

Now, suppose that you first let the fluid flow through without any heating until a stable state is reached. Then you suddenly switch on the heaters: What happens? How long time does it take to reach a new stable state? These questions can be answered by a time-dependent simulation.

It is assumed that you have the final state of this model loaded in the graphical user interface

Loading a Previously Saved Model

If you've previously saved the results from the first part of the model as a Model MAT-file (.mat) or Model M-file (.m), you can now load it into the user interface.

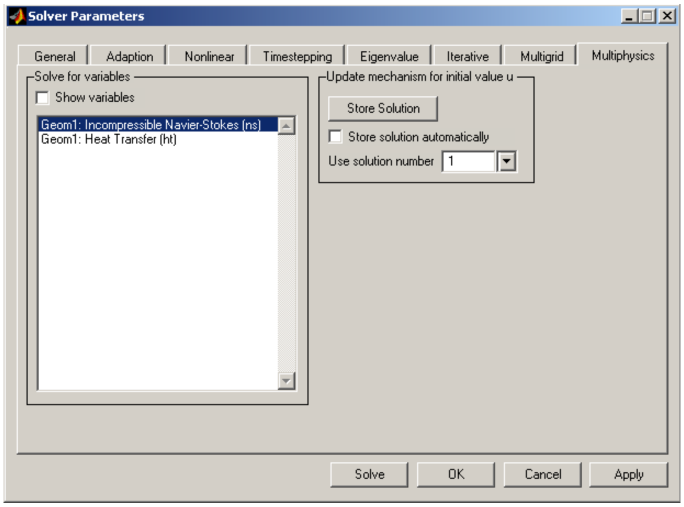


- To load the previously saved model, choose **Open** from the **File** menu, and then **Model MAT-file** or **Model M-file**, as appropriate, in the submenu. You can also press the **Open** button on the main toolbar to load a Model MAT-file or Model M-file.
- Locate your model file and press **OK**.

Solving

First you have to generate the initial state, without heating. This you can reach by simply solving *only* the Incompressible Navier-Stokes problem.

- Choose **Solve for Variables...** from the **Multiphysics** menu. This opens the **Solver Parameters** dialog box on the **Multiphysics** page.



Solve for variables

Show variables: ☐

Geom1: Incompressible Navier-Stokes (ns)
Geom1: Heat Transfer (ht)

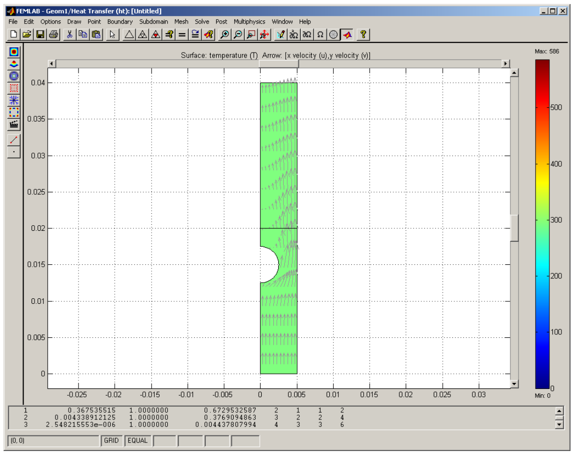
Update mechanism for initial value u

Store Solution

Store solution automatically: ☐

Use solution number: 1

- In the **Solve for variables** list, only select `Geom1: Incompressible Navier-Stokes (ns)`. Press the **Solve** button in the dialog box to solve only for the velocity components *u* and *v*, and pressure *p*.

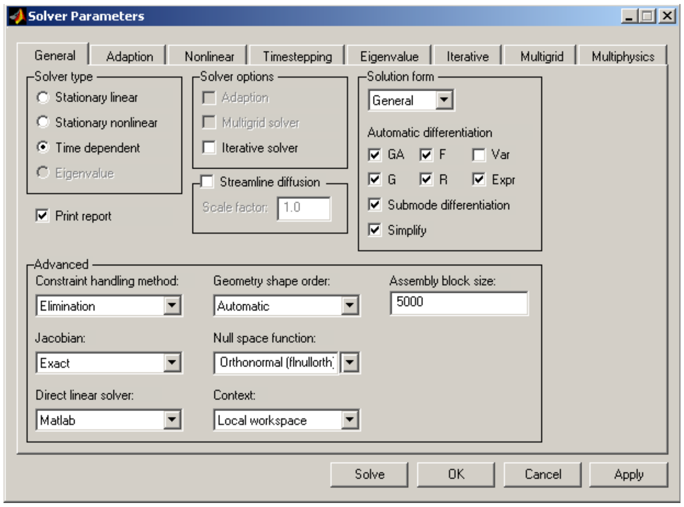


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Surface: temperature (T) Arrow: [x velocity (u),y velocity (v)]
Max: 586
500
400
300
200
100
0
Min: 0
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
-0.025
-0.02
-0.015
-0.01
-0.005
0
0.005
0.01
0.015
0.02
0.025
0.03
(0, 0)
GRID
EQUAL

```
  1        0.367535515   1.0000000        0.6729532587    2    1    1    2
  2     0.004338912125   1.0000000        0.3769094863    3    2    2    4
  3    2.548215553e-006  1.0000000      0.004437807994    4    3    3    6
```

The shown solution also verifies the assumption above, that there should be a *wake effect* visible behind the heating tube. Next, switch to the time-dependent solver:

- Select both application modes. To do so, open the **Solver Parameters** dialog box from the **Solve** menu. On the **Multiphysics** page, select both application modes, by selecting in the **Solve for variables** list. Click on the first application mode and drag down the list, or use **Ctrl**-click for multiple selection. The temperature used is the initial value of T set earlier in the **T(t$_0$)** field on the **Init** page of the **Subdomain Settings** dialog box.
- Click the **General** tab in the **Solver Parameters** dialog box. Under **Solver type** in the top left corner, select **Time dependent**.



Solver type
- ○ Stationary linear
- ○ Stationary nonlinear
- ● Time dependent
- ○ Eigenvalue

☑ Print report

Solver options
- ☐ Adaption
- ☐ Multigrid solver
- ☐ Iterative solver

☐ Streamline diffusion
Scale factor: 1.0

Solution form: General

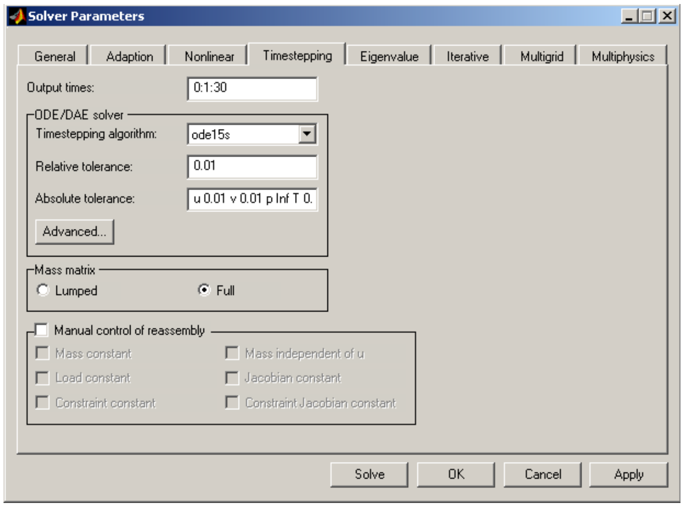
Automatic differentiation
☑ GA ☑ F ☐ Var
☑ G ☑ R ☑ Expr
☑ Submode differentiation
☑ Simplify

Advanced

Constraint handling method: Elimination	Geometry shape order: Automatic	Assembly block size: 5000
Jacobian: Exact	Null space function: Orthonormal (flnullorth)	
Direct linear solver: Matlab	Context: Local workspace	

- Then switch to the **Timestepping** page. The output times are given as a MATLAB vector, specifying for which times the solution should be obtained. Enter `0:1:30` in the **Output times** field, which means that the solution will be sampled every second during 30 seconds.

Note If you are running FEMLAB on MATLAB version 5.2, you also have to change the **Timestepping algorithm** to `fldaspk`. The MATLAB ODE suite solvers can handle differential-algebraic equations—such as the Navier-Stokes equations—only in MATLAB 5.3 and later.



Output times: 0:1:30

ODE/DAE solver

Timestepping algorithm: ode15s

Relative tolerance: 0.01

Absolute tolerance: u 0.01 v 0.01 p Inf T 0.

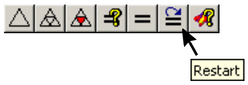
Mass matrix: Lumped [✓ Full]

☐ Manual control of reassembly

☐ Mass constant ☐ Mass independent of u

☐ Load constant ☐ Jacobian constant

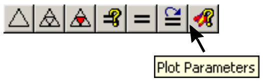
☐ Constraint constant ☐ Constraint Jacobian constant



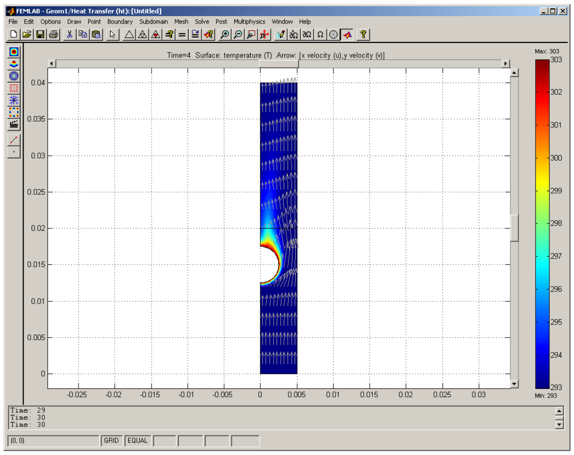
- Press **OK** to confirm the choices. Then press the **Restart** button in the main toolbar to start the simulation. The solution is ready after 168 seconds on a 1.5-GHz Pentium 4 machine.

Post Mode

The solution for the last time step is plotted, using the same plot settings as before. On the **General** page of the **Solver Parameters** dialog box, you can select which time step to display.



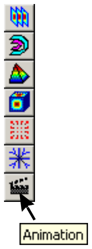
- Open the **Plot Parameters** dialog box, by pressing the **Plot Parameters** button. Go to the **Geneal** page and select the timestep you want to see from the **Solution at time** drop-down list. Press **Apply** to visualize the solution at that time


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Time=4 Surface: temperature (T) Arrow: [x velocity (u),y velocity (v)]
Max: 303
Min: 293
(0, 0)
GRID
EQUAL

```
Time: 29
Time: 30
Time: 30
```

Animating the Solution

The most appealing way to view the results from a time-dependent simulation is often as an animation. When watching an animation, you can easily see features of the solution, that you wouldn't find by examining the solutions for the separate time steps one by one.



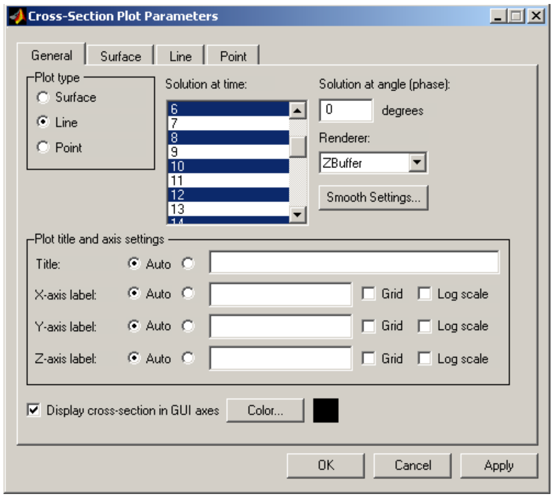
- Press the **Animation** button on the plot toolbar. A new MATLAB figure window is opened, where the animation is first recorded and then played five times. If you want to see the animaiton again, just press the same **Animation** button. The second time, you do not have to wait for the recording phase.

Plotting the Time Evolution

A different way to examine the time-dependent solution is to plot the same aspect of the solution evaluated at multiple times in the same figure. You can, for example, plot the time evolution of the temperature profile on the cross section drawn above.

- If you still have the previous cross-section window on screen, close it.

- Open the **Cross-Section Plot Parameters** dialog box by choosing **Cross-Section Plot Parameters...** from the **Post** menu.



Plot type: ( ) Surface [✓ Line] ( ) Point

Solution at time: 6, 7, 8, 9, 10, 11, 12, 13, 14 (selected: 6, 8, 10, 12)

Solution at angle (phase): 0 degrees

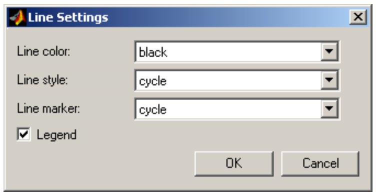
Renderer: ZBuffer

Plot title and axis settings

Title:	[✓ Auto] ( )				
X-axis label:	[✓ Auto] ( )		☐ Grid	☐ Log scale	
Y-axis label:	[✓ Auto] ( )		☐ Grid	☐ Log scale	
Z-axis label:	[✓ Auto] ( )		☐ Grid	☐ Log scale	

☑ Display cross-section in GUI axes ■

- Select all even time steps from 0 to 14 seconds in the **Solution at time** list.
- Switch to the **Line** page and press the **Line Settings** button. This opens up the **Line Settings** dialog box, in which the appearance of the line plot is defined. Set the **Line color** to black, the **Line style** to cycle, and the **Line marker** to cycle. Also check the **Legend** check box before pressing **OK** to close the dialog box.



Line color: black

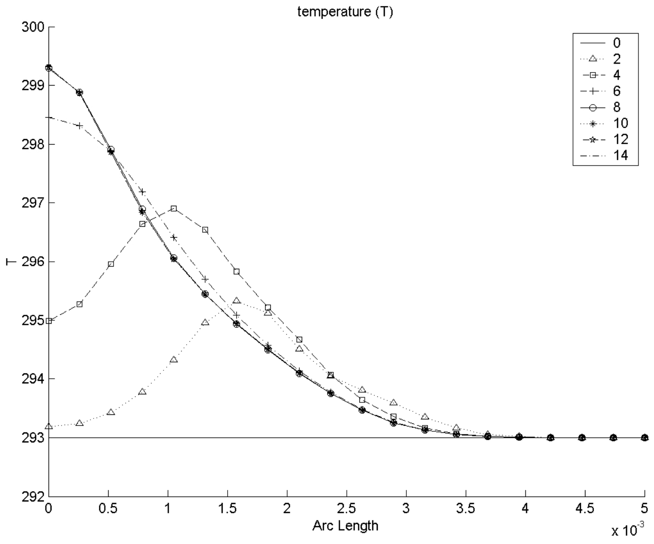
Line style: cycle

Line marker: cycle

☑ Legend

- Type 20 in the **Line resolution** edit box as a final step in setting up the plot parameters. Visualize the plot by pressing **Apply**.

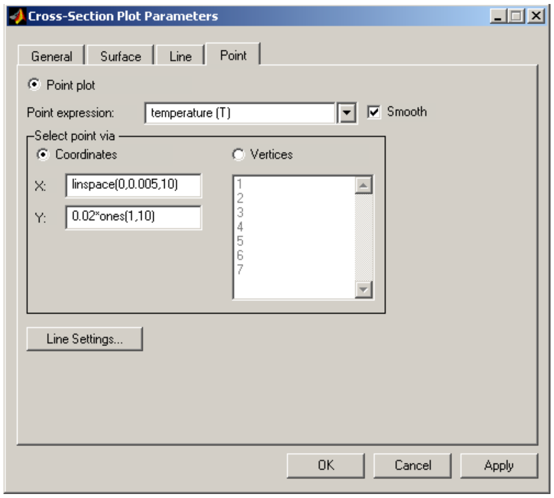
The figure shows line plots for each selected time step. For each curve the corresponding time can be seen in the legend box.



In the above plot, it is clear that the temperature does not change much between the last three time steps visualized. To get a clearer picture of when a stationary state is reached you can plot the time evolution of an expression at a geometry vertex, or at any location inside the geometry.

- Select all time steps in the **Solution time** list at the **General** page in the **Cross-Section Plot Parameters** dialog box.

- Click the **Point** radio button under **Plot type**, and go to the **Point** page of the dialog box. Choose `temperature (T)` from the **Point expression** drop-down list.



Point plot

Point expression: temperature (T)

Smooth: ✓

Select point via

Coordinates [✓]

Vertices

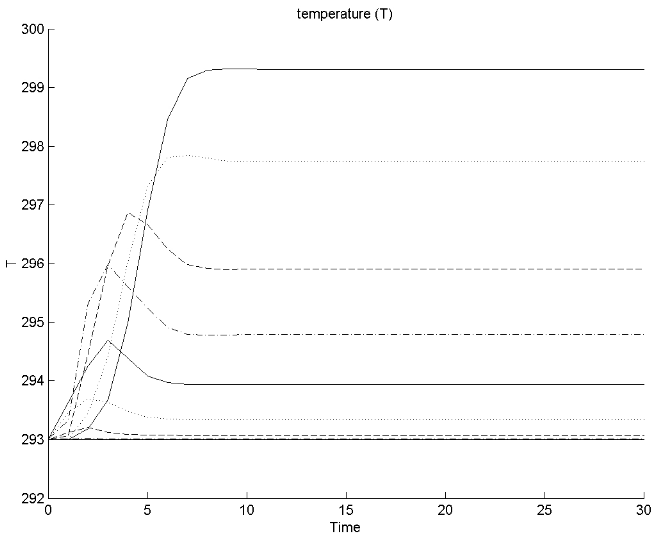
X: linspace(0,0.005,10)

Y: 0.02*ones(1,10)

1
2
3
4
5
6
7

- Press the **Line Settings** button and clear the **Legend** check box. Also set the **Line marker** to none. Press **OK** to use the new settings.
- The x and y coordinates of the points, for which you want to plot the time evolution, are entered as MATLAB vectors. To plot the solutions at 10 equally spaced points along the cross section used above, enter `linspace(0,0.005,10)` in the **X** field, and `0.02*ones(1,10)` in the **Y** field.
- Alternatively, press the **Draw Point for Cross-Section Point Plot** button at the bottom of the plot toolbar. Then you can place the points manually along the

cross-section line, using the left mouse button. Press **OK** in the **Cross-Section Plot Parameters** dialog box.



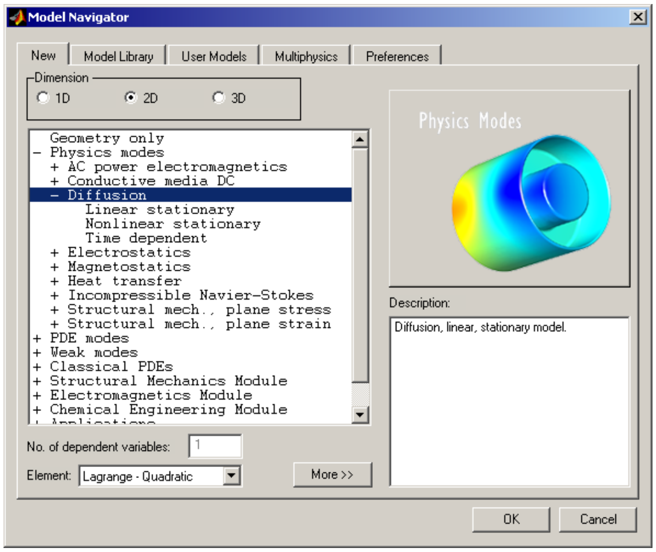
From this last plot, you can directly answer the second question: "How long time does it take to reach a new stable state?". After approximately 10 seconds, the solution has stabilized completely on the chosen cross section.

The Model Navigator and the Model Library

This section describes the **Model Navigator** dialog box. The **Model Navigator** is a multi-purpose tool used to control the general settings of a FEMLAB session.

The Model Navigator

When you start FEMLAB you invoke the **Model Navigator**.



Dimension: 1D [✓ 2D] 3D

```
  Geometry only
- Physics modes
  + AC power electromagnetics
  + Conductive media DC
  - Diffusion
      Linear stationary
      Nonlinear stationary
      Time dependent
  + Electrostatics
  + Magnetostatics
  + Heat transfer
  + Incompressible Navier-Stokes
  + Structural mech., plane stress
  + Structural mech., plane strain
+ PDE modes
+ Weak modes
+ Classical PDEs
+ Structural Mechanics Module
+ Electromagnetics Module
+ Chemical Engineering Module
```

No. of dependent variables: 1

Element: Lagrange - Quadratic

Description: Diffusion, linear, stationary model.

The **Model Navigator** is a tabbed dialog box with five pages:

- **New** page — for initializing application mode models
- **Model Library** page — for loading Model Library models
- **User Models** page — for loading your own models
- **Multiphysics** page — for initializing a multiphysics model
- **Preferences** page — for setting the Model Library paths and Preference File

You can see the five tabs at the top of the dialog box. To start creating a new model you click on the **New** tab and choose the application mode that you wish to work in. To load a model from the Model Library, click the **Model Library** tab and choose a model from one of the application areas. You can also build your own models and access them from the **User Models** page. The **Multiphysics** page is used for initializing multiphysics models. The **Preferences** page can be used to set the default session parameters for the Model Library, the User Models and the GUI.

New Page

There are four sets of modes available on the **New** page: *Physics modes*, *PDE modes*, *Weak modes*, and *Classical PDEs*. The optional *modules* are also listed here and in this case they are the Chemical Engineering Module, the Electromagnetics Module, and the Structural Mechanics Module. Different modes and modules have in turn a number of sub-modes where you can specify which type of model you want to run: linear or nonlinear, stationary, or time-dependent, etc. The application modes consist of:

- 1D Physics modes: Diffusion and Heat transfer
- 2D Physics modes: AC power electromagnetics, Conductive media DC, Diffusion, Electrostatics, Magnetostatics, Heat transfer, Incompressible Navier-Stokes, Structural mechanics Plane strain, Structural mechanics Plane stress
- 3D Physics modes: Conductive media DC, Diffusion, Electrostatics, Magnetostatics, Heat transfer, Incompressible Navier-Stokes, Structural mechanics
- PDE modes for modeling using partial differential equations, see "The FEMLAB PDE Language" on page 1-336
- Classical PDE modes: Laplace's equation, Poisson's equation, Helmholtz's equation, Heat equation, Wave equation, Schrödinger's equation, Convection diffusion equation
- Weak modes: Subdomain, Boundary, Edge (3D only), Point, Boundary constraint

For the physics modes, the notation for material properties and boundary conditions are adapted to the application at hand, for example, flux, heat capacity, conductivity, etc.

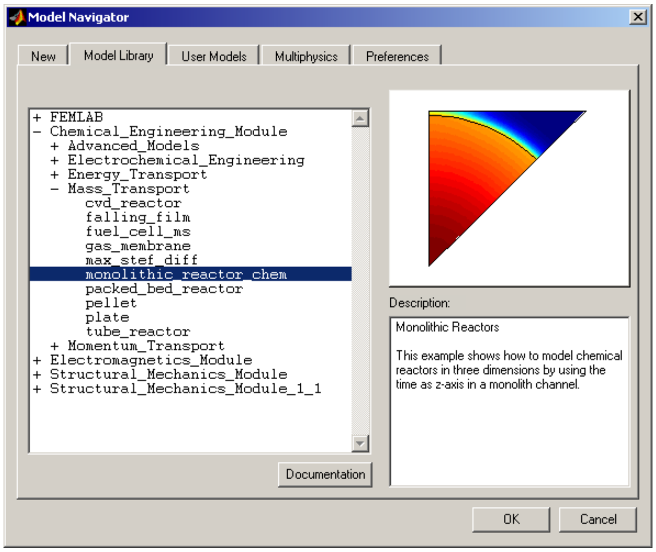
For models where there is no application mode in the **Model Navigator**, you can use the PDE modes to specify your PDE and boundary conditions. Depending on the special form of your PDE, there are several modes to choose from. A more detailed description of the types of PDEs available can be found in "The FEMLAB PDE Language" on page 1-336. A complete description can be found in the *Reference Guide*.

Many *Classical PDEs* can be given as instances of the coefficient form in FEMLAB. Some of them are available in the Model Navigator.

There is also an *Application* available of a wave guide model that was implemented using the FEMLAB API.

Model Library Page

On the **Model Library** page, there are a number of completed models from different areas of engineering and science.



```
+ FEMLAB
- Chemical_Engineering_Module
  + Advanced_Models
  + Electrochemical_Engineering
  + Energy_Transport
  - Mass_Transport
      cvd_reactor
      falling_film
      fuel_cell_ms
      gas_membrane
      max_stef_diff
      monolithic_reactor_chem
      packed_bed_reactor
      pellet
      plate
      tube_reactor
  + Momentum_Transport
+ Electromagnetics_Module
+ Structural_Mechanics_Module
+ Structural_Mechanics_Module_1_1
```

Description:

Monolithic Reactors

This example shows how to model chemical reactors in three dimensions by using the time as z-axis in a monolith channel.

The Model Library is important for several reasons: it offers a quick way to learn the facilities of FEMLAB and examine the results of completed models. You can also save time by using models from the Model Library as an initial platform for developing your own models. You can modify the geometry, the values of variables, boundary conditions, etc. The Model Library is also indispensable for demonstration purposes.

On the **Model Library** page you can find approximately 40 completed models. They are fully described in detail in the *Model Library*. In FEMLAB version 2.3, you will find the following models:

- Acoustic Models
  - Eigenmodes of a room
  - Acoustics of a heat exchanger
  - Humming from rotating electrical machinery
  - Humming from rotating electrical machinery with an obstacle
- Chemical Reaction Models
  - Monolithic reactor
  - Tubular reactor
  - Cylindrical tubular reactor with cooling
- Classical PDEs
  - Heat transfer problem with differing materials
  - Poisson's equation on the unit disk with a point source in the origin
  - Poisson's equation on the unit disk
  - Eigenfunctions of the Schrödinger equation for an electron
  - A transport problem with discontinuous solution
  - The wave equation for transverse vibrations of a membrane
- Electromagnetic Models
  - Safety Connector/Disconnector
  - Magnetic field in an electric motor
  - Electrostatic Precipitation Filter
  - A permanent magnet
  - Potential between two cylinders
  - Skin effect in circular wire
  - Protection of a steel tank immersed in seawater
  - A rectangular wave guide

- Equation-Based Models
  - The Black-Scholes equation in 1D
  - The Black-Scholes equation in 2D
  - The Burgers equation
  - Eigenvalues and eigenmodes of a square
  - The KdV equation and solitons
  - The Landau-Ginzburg equation
  - A minimal surface problem
- Fluid Dynamics Models
  - Stationary incompressible flow over backstep
  - Flow over backstep using stream function-vorticity formulation
  - Flow over backstep with Argyris element
  - Flow past a cylinder
  - Flow of paper pulp
  - The shallow water equations
  - Shock tube
- Geophysics Models
  - A rock fracture flow model
- Heat transfer Models
  - Heat generation in a disk brake
  - Heat distribution in a radioactive rod
- Multidisciplinary Models
  - Magnet brake
  - Thermal controller
  - Thermal controller using non-local coupling variables

- Multiphysics Models
  - Stresses in cracked heat exchanger tubes
  - Thermo-electric heating
  - Bus bar
  - Free convection
  - Marangoni convection
  - Micro robot
  - Vibrations in a milk containers
- Physics Models
  - The Schrödinger equation for the hydrogen atom
- Semiconductor Device Models
  - Semiconductor diode model
- Structural mechanics Models
  - The deformation of a feeder clamp
  - A Mindlin plate problem
- Wave Propagation Models
  - Diffraction patterns
  - Isospectral drums
  - Eigenvalues and eigenfunctions for the L-shaped membrane

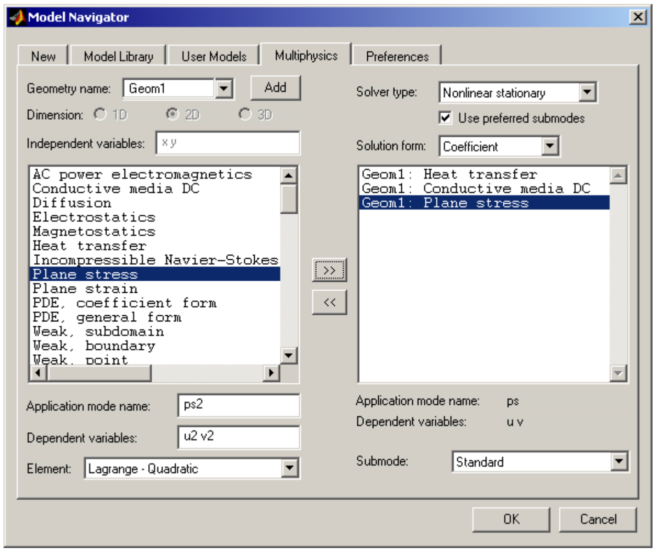
To obtain the on-line documentation for a specific model, press the **Documentation** button while the model is selected in the Model Library tab. The on-line documentation can be used in various ways, for example, as a starting point for working with scripts, by copying from the documents and pasting in the MATLAB Command Window.

User Models Page

On the **User Models** page, you can browse your own set of saved models. If you have saved a model together with a *model image* and a *model description*, you can display the model with an image and a short description, in the same way you display models on the **Model Library** page.

Multiphysics Page

On the **Multiphysics** page you initialize multiphysics models, by selecting from a list of application modes. In the GUI, you are then able to couple all the selected application modes. At any time during the modeling process, you can open the **Model Navigator**'s **Multiphysics** page again, to add or remove application modes.



Geometry name: Geom1

Dimension: 1D [✓ 2D] 3D

Independent variables: x y

```
AC power electromagnetics
Conductive media DC
Diffusion
Electrostatics
Magnetostatics
Heat transfer
Incompressible Navier-Stokes
Plane stress
Plane strain
PDE, coefficient form
PDE, general form
Weak, subdomain
Weak, boundary
Weak, point
```

Application mode name: ps2

Dependent variables: u2 v2

Element: Lagrange - Quadratic

Solver type: Nonlinear stationary

[✓] Use preferred submodes

Solution form: Coefficient

```
Geom1: Heat transfer
Geom1: Conductive media DC
Geom1: Plane stress
```

Application mode name: ps

Dependent variables: u v

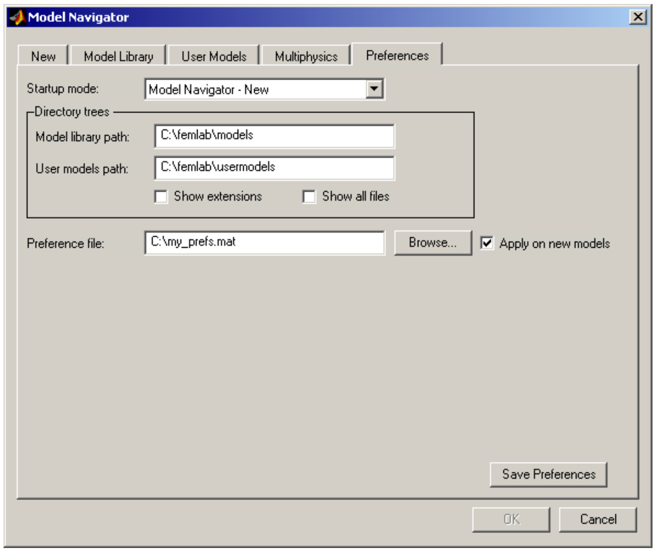
Submode: Standard

For more information on how to use the multiphysics feature, see the multiphysics model “Thermo-Electric Heating in a Bus Bar” on page 1-109, and also the multiphysics models of the *Model Library*.

Preferences Page

The **Preferences** page lets you specify the Model Library path, the User Models path and preference file. The preference file, with the extension `.mat`, is a FEMLAB

Model MAT-file that can be used for initializing the GUI settings, such as axis limits, grid spacing, etc. You can also specify the **Startup mode** for a FEMLAB session.


Model Navigator

Startup mode: Model Navigator - New

Directory trees

Model library path: C:\femlab\models

User models path: C:\femlab\usermodels

☐ Show extensions ☐ Show all files

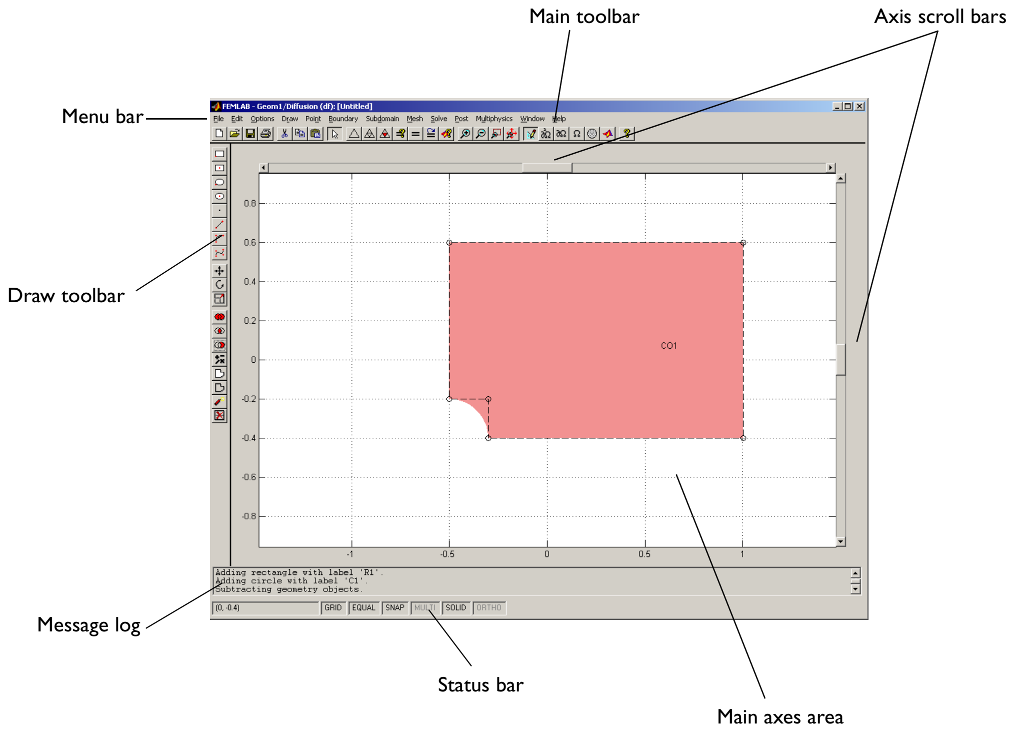
Preference file: C:\my_prefs.mat

☑ Apply on new models

The FEMLAB Graphical User Interface

FEMLAB features a graphical user interface (GUI) that handles all aspects of the modeling process. Indeed, the simplest way to model an object is with the graphical interface, which is automatically invoked when you specify an application mode in the **Model Navigator**. For some models, however, it can be more expeditious to use the programming language (see "The Programming Language" on page 1-301).

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



Main toolbar
Axis scroll bars
Menu bar
FEMLAB - Geom1/Diffusion (df): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Draw toolbar
CO1
Message log
Status bar
Main axes area
(0, -0.4)
GRID EQUAL SNAP MULTI SOLID ORTHO

```
Adding rectangle with label 'R1'.
Adding circle with label 'C1'.
Subtracting geometry objects.
```

The *menu bar* enables control of all aspects of the modeling process. It conforms to the common menu standards:

- Items followed by a right arrow lead to a submenu.

- Items followed by an ellipsis lead to a dialog box.
- Stand-alone items initiate a direct action.
- All menu items have corresponding keyboard interfaces (use ALT + the underlined character for drop-down menus). You can also execute some menu items with shortcut keys.

In the *main axes area* you draw the geometry, display the mesh and plot the solution, etc.

A *message log* in the lower part of the interface provides progress information from solvers as well other items of information.

Located just below the message log, the *status bar* provides details about the status of settings such as grid lines. Here you can also examine current cursor coordinates. You can toggle the state of a status bar item by double-clicking on it.

The *main toolbar* below the main menu contains icon buttons that provide quick access to some of FEMLAB's most important functions. For instance, additional toolbars sometimes appear: in draw mode, a draw toolbar appears; in post mode you gain access to a plot toolbar.

For a complete description of all menu items, dialog boxes, and other interface components, see the chapter *The Graphical User Interface* in the *Reference Manual*.

The Menus

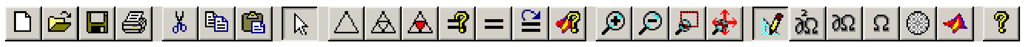
The graphical interface provides several selections from the menu bar.

- **File** menu—where you open and save files, as well as import and export data to the MATLAB main workspace or to a file. You can also print the current graphics and exit the graphical interface.
- **Edit** menu—where you cut, clear, copy, and paste geometry objects, boundary conditions, material properties and initial conditions. You can perform multiple undo steps of geometry and mesh actions. A **Select All** and a **Deselect All** option is also available.
- **Options** menu—where the options allow you to modify axis settings, zoom levels, rendering settings, and labels. You can also define constants and variables for use in boundary conditions, and when defining material properties, change assigned variable names, specify differentiation rules, etc.

- **Draw** menu—where you select and create primitive solid objects such as circles, spheres, blocks, and rectangles, which you can then modify with the mouse. You can move, scale, and rotate objects as well as do coercions of geometry objects. It is possible to also create composite objects.
- **Point** menu—where you can open a dialog box in which you define point settings. (Only in 2D and 3D.)
- **Edge** menu—where you can open a dialog box in which you define edge settings. (Only in 3D.)
- **Boundary** menu—where you can, among other things, open a dialog box in which you define boundary conditions.
- **Subdomain** menu—where you can, among other things, open a dialog box for specifying material properties or PDE coefficients.
- **Mesh** menu—where you create and modify the finite element mesh. You can initialize, refine, and smooth the mesh as well as display mesh quality.
- **Solve** menu—where you start and control model computations. From this menu you can open a dialog box where the solver parameters are set.
- **Post** menu—where you can post-process the solution. A wide range of plot types are available as well as integration tools.
- **Multiphysics** menu—where you add application modes to or delete application modes from a multiphysics model. It is also possible to switch among application modes that were added earlier. Also under this menu you can specify which solution components to solve for.
- **Window** menu—where you select any currently open MATLAB figure window; FEMLAB immediately brings the selected window to the front.
- **Help** menu—where you find help about FEMLAB's functions, read on-line documentation and reach additional help resources locally as well as over the Internet.

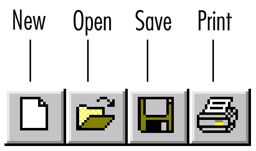
The Toolbars

The main toolbar underneath the main menu contains icon buttons that provide easy access to key functions. They do not, however, offer any additional functionality; all toolbar button functions are also available as menu items.



The toolbar consists of seven parts. From the left to right, there are: four buttons addressing file handling; three buttons handling editing; one button for object selections; seven buttons addressing meshing, solving, and plotting; four buttons implement zooming; six (in 2D) buttons handle mode navigation; and at the end comes a Help button.

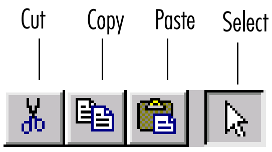
FILE HANDLING BUTTONS



The file handling buttons (from left to right) perform these actions:

- Open the **Model Navigator** to start a new model.
- Load a Model MAT-file/Model M-file from disk.
- Save the current model to a Model MAT-file.
- Print a hardcopy screen dump of the FEMLAB main axes.

EDIT BUTTONS AND THE SELECT BUTTON



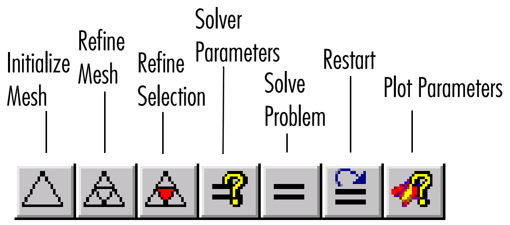
The edit buttons and the select button (from left to right) perform these actions:

- Move the selected geometry objects to the FEMLAB clipboard.
- Copy the selected geometry objects, boundary conditions, PDE coefficients, or initial conditions to the FEMLAB clipboard.

- Paste FEMLAB clipboard contents into the current model.
- Select items such as menus, buttons, or objects.

The select button (with an arrow icon) offers a way to select objects by clicking on them.

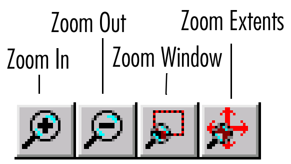
MESH, SOLVE, AND PLOT BUTTONS



The mesh, solve, and plot buttons (from left to right) perform these actions:

- Generate and display an initial mesh.
- Refine the entire current mesh.
- Uniformly refine only selected triangles (1D and 2D only).
- Open the **Solver Parameters** dialog box.
- Compute the solution for the current model and plot the result (you can disable the automatic solution plot).
- Restart the solution process after you previously halted it with the **Stop** button.
- Open the **Plot Parameters** dialog box.

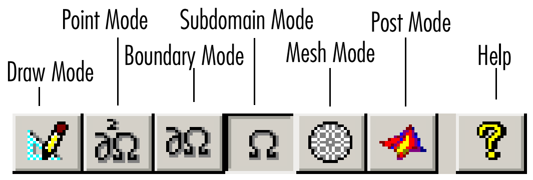
ZOOM BUTTONS



The zoom buttons (from left to right) perform these actions:

- Zoom in, centered.
- Zoom out, centered.
- Zoom in a user-defined zoom window.
- Zoom out as far as the extents of the geometry.

GUI MODE NAVIGATION BUTTONS AND THE HELP BUTTON



The mode navigation buttons provide a convenient mechanism for mode-selection accompanied with a visual indication of the current mode. The six buttons (from left to right) activate the following modes:

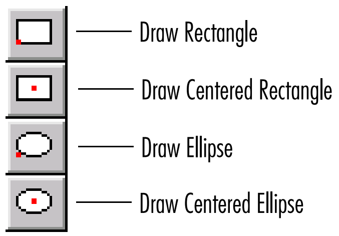
- Draw mode
- Point mode
- Boundary mode
- Subdomain mode
- Mesh mode
- Post mode

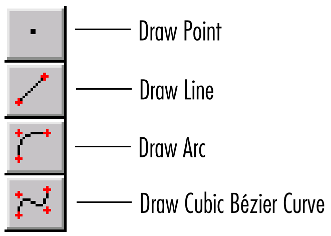
The last button on the right is the help button, which launches the FEMLAB Help Desk. In 3D, there is an additional edge mode and a corresponding button.

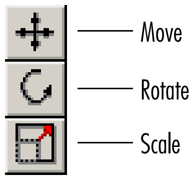
2D DRAW TOOLBAR BUTTONS

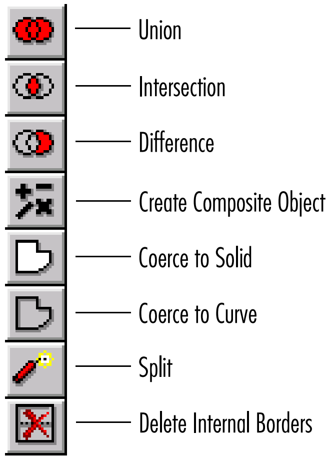
In its default configuration, the 2D draw mode provides an additional draw toolbar that contains buttons for creating and editing the geometry. The 2D draw toolbar is also available in the work planes for a 3D model. See the section “Work Plane Toolbar

Buttons" on page 1-83 for details. The 3D draw toolbar is described in the section"3D Draw Toolbar Buttons" on page 1-83.



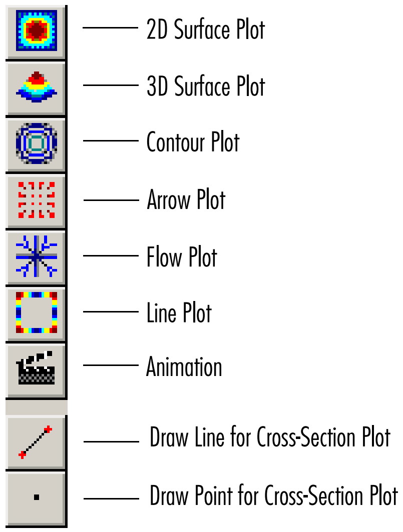






2D PLOT TOOLBAR BUTTONS

In its default configuration, post mode provides an additional plot toolbar that contains buttons for visualization and animation.



The 3D plot toolbar is described in the section “3D Plot Toolbar Buttons” on page 1-85.

The GUI Modes

The FEMLAB solution process typically involves the following steps:

1 Define the geometry.

2 Define boundary conditions.

3 Define material properties (PDE coefficients).

4 Create the mesh.

5 Solve the problem.

6 Plot the solution and other physical properties calculated from the solution (postprocessing).

The design of FEMLAB’s graphical interface mirrors this scheme. You work in different modes, each corresponding to a step in the solution process:

1 In *draw mode* you create the geometry. You create geometry models by using CAD tools and either solid-modeling or boundary-modeling techniques. For 2D and 3D modeling, this mode provides a set of primitive solid objects (rectangle, circle, ellipse, block, cone, cylinder, ellipsoid, and sphere) that can be combined into composite objects. You can also draw lines, arcs, and rational Bézier curves. Other tools can convert areas enclosed by curves into solids. It is also possible to import a geometry from an external file.

2 In *boundary mode*, *edge mode* (only in 3D) and *point mode* (in 2D and 3D), you specify boundary conditions and conditions on edges and points. You can specify different types of conditions on different domains.

   In these modes, you optionally set initial conditions for time-dependent problems and the nonlinear solver. It is possible to set varying initial values on different domains.

   Note that there is also support for periodic boundary conditions, see the *Model Library* for examples.

3 In *subdomain mode* you specify application-dependent material properties for problems you base on templates from FEMLAB's *Physics application modes* or supply PDE coefficients in the *PDE application modes*. You can specify values for each subdomain independently. For instance, you can easily specify multiple material properties within one FEMLAB model.

   In *subdomain mode* you also set initial conditions needed for time-dependent problems and the nonlinear solver. It is possible to set initial values on different subdomains.

4 In *mesh mode* you control automated mesh generation and mesh visualization.

5 You solve your problem by selecting **Solve Problem** on the **Solve** menu. You can also invoke the solver, and and set various solver properties, by calling the **Solver Parameters** dialog box.

6 *Post mode* provides a wide range of visualization and other postprocessing possibilities. You can visualize solutions either inside the FEMLAB graphical interface or outside it as a separate MATLAB plot. Several solution properties can be visualized at the same time using colored surfaces, contours, streamlines, and vector fields. It is also possible to plot expressions, including the solution, in cross-sections of the geometry and on boundaries and edges, and to make point plots or "time extrusion" plots. You can also animate any solution. In post mode, subdomain and boundary integration tools are also available.

Object Selection Methods in 2D

Throughout the 2D graphical interface, similar principles apply for selecting items such as geometry objects, subdomains, and boundaries.

- To select a single geometry object, click on it using the left mouse button.
- To select several objects or deselect some of them, **Shift**-click or **Ctrl**-click on the desired objects. On a 3-button mouse, clicking the middle or the right mouse buttons instead of the left button corresponds to **Shift**- and **Ctrl**-clicking, respectively.
- Clicking on the intersection of several objects selects all of them.
- To open an object's dialog box, double-click on the object.
- Clicking outside of the objects deselects all of them.
- To select all objects, use the **Select All** option from the **Edit** menu or the keyboard shortcut **Ctrl-a**.

You can also employ a *rubber band box* to select objects. Click and drag while holding the mouse button to create a rubber band box. Selection principles remain as above, and the selection applies only to objects that the rubber band box surrounds completely. **Shift**-clicking adds the rubber band box selection to the current selection, while **Ctrl**-clicking performs the exclusive-or operation with the current selection (in other words, presently selected objects becomes deselected when inside the rubber band box and vice versa).

Display Additional Information

In mesh mode, you can display the element number at the current position by pressing a mouse button. Clicking a triangle displays the triangle number in the message log.

In post mode in 2D, you can display the numerical value of the current surface or contour plot at the position where you click. Press a mouse button to display the value of the plotted property in the message log. In 1D, clicking in the main axes area, displays the value of the currently plotted expression.

If you want to get the values of the solution at several points, you can export the entire model to MATLAB as a so-called FEM structure. You do this by selecting **Export FEM Structure as 'fem'** from the **File** menu. Then you can use the function `postinterp` at MATLAB's prompt to compute numerical values for any expression. See the entry on `postinterp` in the *Function Reference*.

Entering MATLAB® Expressions in the GUI

In many of the edit fields that appear in dialog boxes within the graphical interface you can enter MATLAB expressions, for example, in the **Boundary Settings** and **Subdomain Settings** dialog boxes.

When entering parameter values as an expression—for example as a function of the space coordinates x, y and z, also known as independent variables—the string must be a valid MATLAB expression.

Each application mode works with a set of subdomain and boundary variables. You can enter these variables as part of a MATLAB expression in the corresponding edit fields. For instance, in the electrostatics application mode you have access to the electric potential `V` as a subdomain variable. Consider another example: in the plane stress application mode you have access to the surface traction variables `Fx` and `Fy` as boundary variables. For more information on valid subdomain and boundary variables, see the section "The FEMLAB PDE Language" on page 1-336 and "Application Mode Reference" on page 4-203 in the *Reference Manual*.

The FEMLAB Data Formats

FEMLAB can save anything you create in the graphical interface and reload it later. You can save and load models either as binary code (in a Model MAT-file) or as MATLAB M-code (in a Model M-file). You can also load a Model M-file into a text editor for modification or extension with additional MATLAB statements. A Model M-file executes in the MATLAB command window as does any MATLAB M-file. Find more details about these formats in the sections "Model M-files" on page 1-51 and "The Programming Language" on page 1-301.

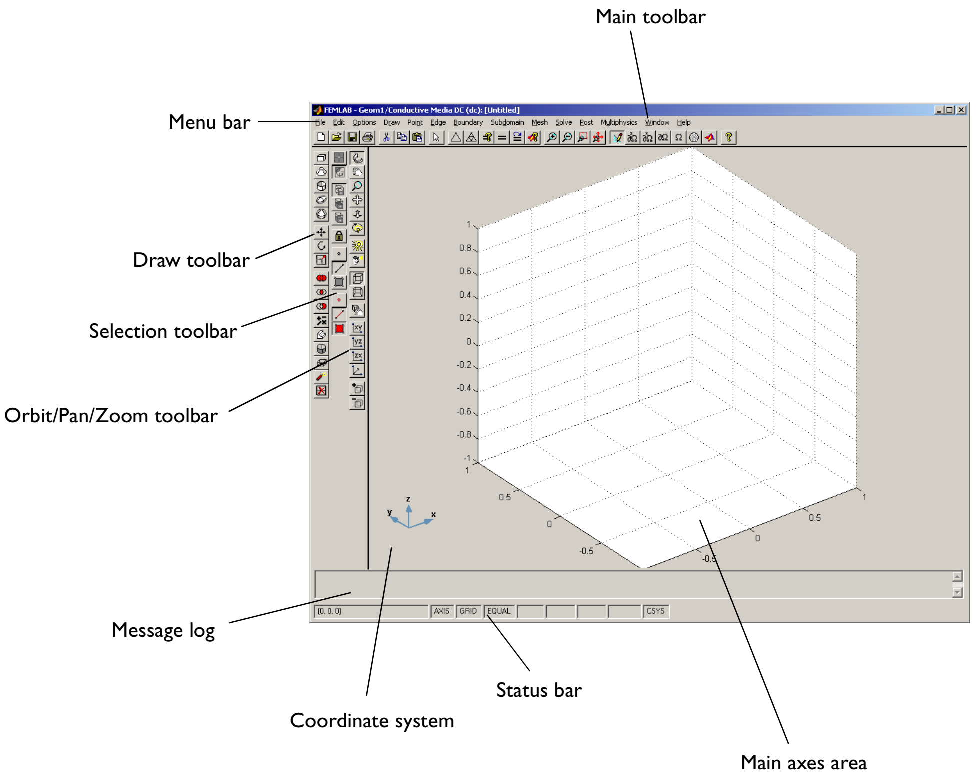
During a FEMLAB session you can export a version of the geometry description to the MATLAB workspace, operate on it from the MATLAB prompt, and import it back into FEMLAB for further development. It is also possible to export boundary conditions and PDE coefficients, or export all FEM data bundled into one MATLAB structure called the FEM structure (see "The FEM Structure" on page 1-301). You can import and export 2D geometry objects using a DXF file and import 3D geometry objects from an IGES file. See "DXF Import" on page 1-209 and "IGES

Import" on page 1-213, respectively. After importing a file, FEMLAB automatically converts its contents to FEMLAB geometry objects.

The 3D Graphical User Interface

This section addresses 3D aspects of modeling within FEMLAB. A significant part of what has been said about the 2D graphical interface is applicable in 3D. The main differences between the 3D and 2D graphical interfaces arise in geometry modeling, in visualization possibilities of a 3D geometry, and in methods for selecting objects.

The figure below shows the 3D graphical interface together with annotations pointing out its most important parts.

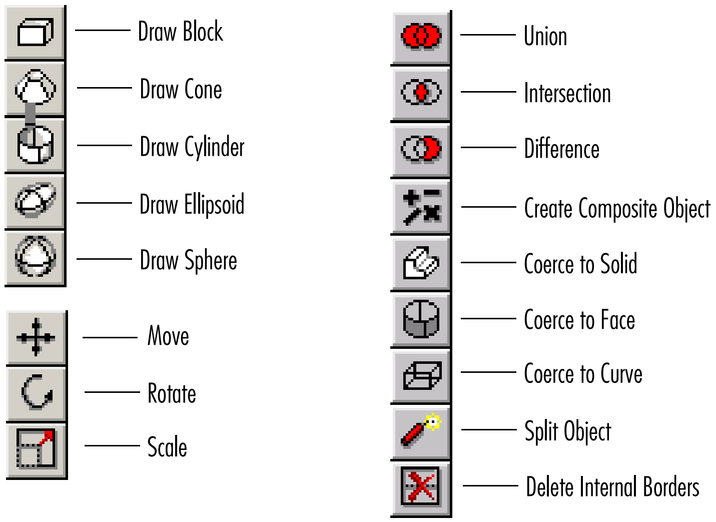


The 3D menu bar resembles the one in the 2D graphical interface, except for the **Edge** menu, which is not available in 2D. The contents of the menus also differ slightly. For example, the **Draw** menu differs considerably from the 2D case. You'll also find variations in the export/import submenus on the **File** menu as well as additional 3D visualization tools on the **Options** menu, the **Boundary** menu, the **Subdomain** menu, and the **Mesh** menu.

For a complete description of all menu items, dialog boxes, and other items, see the chapter "The 3D Graphical User Interface" in the *Reference Manual*.

3D DRAW TOOLBAR BUTTONS

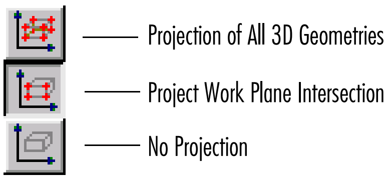
The draw toolbar in 3D has a different appearance than the 2D toolbar. However, many of the buttons work in a similar manner.



WORK PLANE TOOLBAR BUTTONS

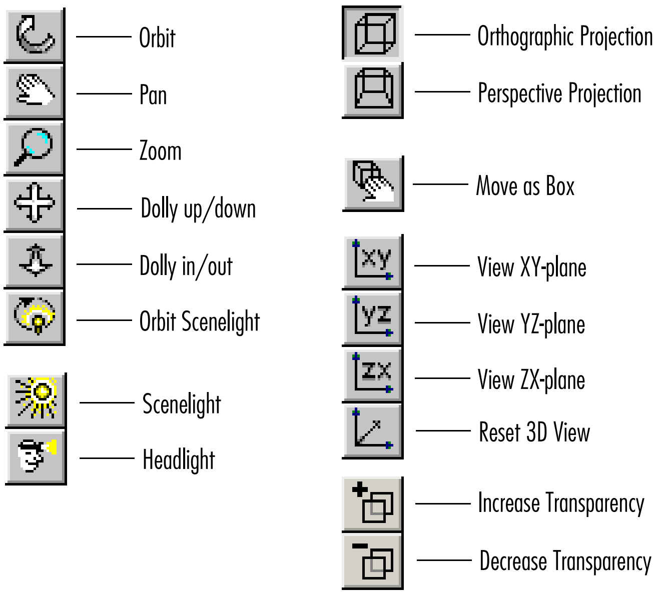
The work plane draw toolbar is visible only for 3D models when you are in a work plane view. This toolbar is essentially the same as the 2D draw toolbar with a few additions at the bottom. See "2D Draw Toolbar Buttons" on page 1-76 for details.

The additional buttons are the ones for the projection of the 3D geometry onto the work plane. You can snap to the projection when drawing.


Projection of All 3D Geometries
Project Work Plane Intersection
No Projection

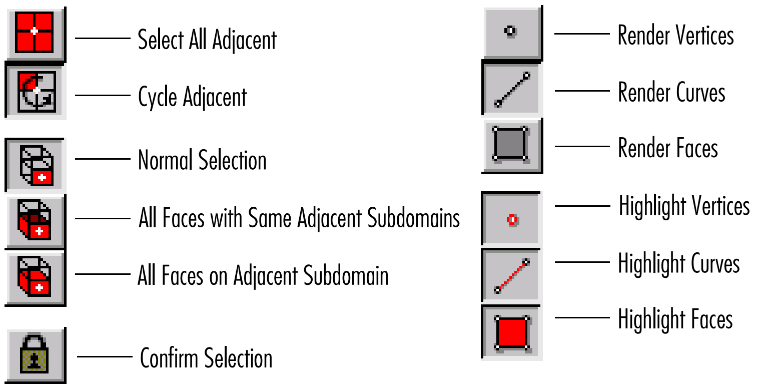
ORBIT/PAN/ZOOM TOOLBAR BUTTONS

The orbit/pan/zoom toolbar is available in all modes in 3D. It contains buttons to change the 3D view and to select lighting tools.



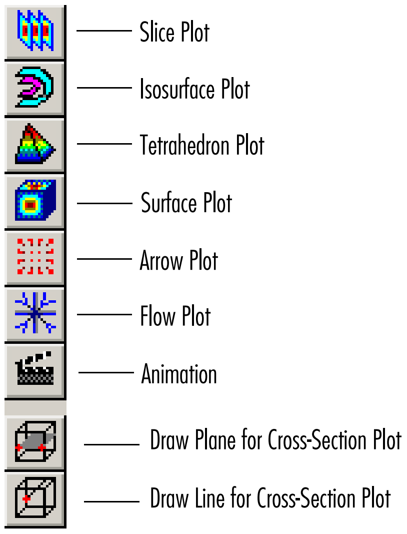
SELECTION TOOLBAR BUTTONS

The mechanism for making selections is by necessity more elaborate in 3D than in 2D. Therefore it is convenient to work with the select toolbar, which contains buttons for visualization and selection purposes. It is available in all modes.



3D PLOT TOOLBAR BUTTONS

In the default configuration, post mode provides an additional plot toolbar that contains buttons for visualization and animation.



Visualization in 3D

The visualization of a 3D geometry object is important not only for understanding what the geometry looks like, but also for selection purposes. The visualization depends on the object types being rendered and the ones being highlighted. For example, you can choose to render only a *wireframe plot* of deselected face objects or to render highlighted faces when they are selected. You control the visualization's quality through the choice of the geometry mesh detail, the graphics renderer, and lighting options.

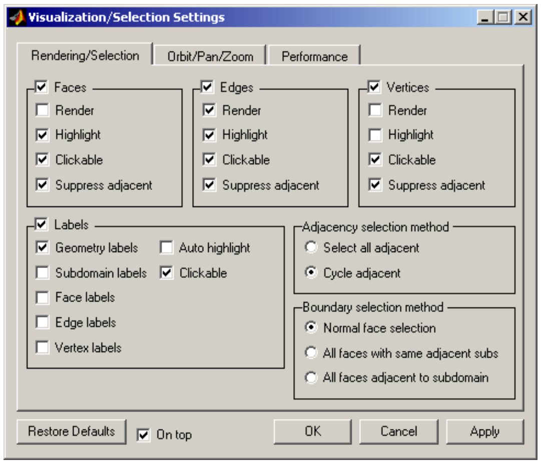
To fully understand the following sections it is important to understand how FEMLAB composes a geometry object. It does so using faces, edges and vertices. For example, a solid cylinder consists of one *subdomain* surrounded by six *faces*. The faces are bound by twelve *edges* that connect at eight *vertices*. Thus, FEMLAB can visualize a cylinder by displaying one or several of these four types. For instance, you can generate a wireframe plot by rendering only the cylinder's edges.

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

The **Visualization/Selection Settings** dialog box on the **Options** menu contains all the tools for controlling the visualization. The dialog box has three pages: **Rendering/ Selection**, **Orbit/Pan/Zoom** and **Performance**.

RENDERING AND HIGHLIGHTING

The first page in the **Visualization/Selection Settings** dialog box is the **Rendering/ Selection** page. Here you specify what to render and highlight. You also determine which objects are clickable and how to suppress objects. These settings apply to draw mode, boundary mode and subdomain mode. Settings on this page also determine what you actually select when clicking on adjacent objects or when clicking on faces in boundary mode. More details are available in the section “Object Selection Methods in 3D” on page 1-95.



Visualization/Selection Settings

Rendering/Selection | Orbit/Pan/Zoom | Performance

[✓ Faces]
Render
[✓ Highlight]
[✓ Clickable]
[✓ Suppress adjacent]

[✓ Edges]
[✓ Render]
[✓ Highlight]
[✓ Clickable]
[✓ Suppress adjacent]

[✓ Vertices]
Render
Highlight
[✓ Clickable]
[✓ Suppress adjacent]

[✓ Labels]
[✓ Geometry labels] Auto highlight
Subdomain labels [✓ Clickable]
Face labels
Edge labels
Vertex labels

Adjacency selection method
Select all adjacent [✓ Cycle adjacent]

Boundary selection method
[✓ Normal face selection] All faces with same adjacent subs All faces adjacent to subdomain

Restore Defaults [✓ On top] OK Cancel Apply

Each of the four object types (**Faces**, **Edges**, **Vertices** and **Labels**) displays its own frame on this page; further, a check box next to each frame title allows for the quick and complete disabling of a specific object type.

In each of the top three frames, the **Render** check box determines how an object should appear on the screen when it’s not selected. The **Highlight** check boxes determine which object types to highlight when you select an object. Note that highlighting and rendering properties are independent of each other.

In the labels frame you specify which labels the graphical interface should render. The labels also have an **Auto highlight** property that ensures the interface highlights the label of a selected object if you haven’t checked its highlight check box.

Rendering and highlighting properties for faces, edges and vertices have corresponding toolbar buttons in the select toolbar.

CLICKING

Each object-type frame on the **Rendering/Selection** page also contains a **Clickable** check box that controls whether that object type is clickable or not. This box must be checked if you want an object to recognize a mouse click. For example, if you have trouble selecting a small face object due to the presence of nearby edge, you can deactivate clickable for edges. For vertices, edges, and faces, what can be clicked on is independent of what is rendered. However, to be able to make labels clickable, you must also render them. The clickable-property has no relevance in mesh or post mode where you cannot normally make selections in the main graphical interface axes.

SUPPRESS ADJACENT PROPERTY

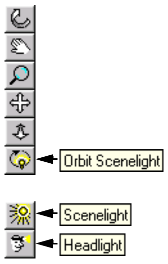
Another important aspect of the 3D visualization capabilities concerns the *suppression* feature. By suppressing objects, you can reveal hidden objects or clear the way to access objects that otherwise would be difficult to reach. In FEMLAB you can suppress geometry objects, subdomains, and faces. The **Suppress adjacent** property dictates whether or not to suppress adjacent objects of a suppressed object. For example, if you check the **Suppress adjacent** box in the **Edges** frame, the edges enclosing a suppressed face object are also suppressed if they're not adjacent to another non-suppressed face. Suppression affects all modes, but the suppress adjacent property applies only to draw mode, boundary mode, and subdomain mode. This manual devotes a separate section on page 1-106 to the suppression feature.

QUALITY OF VISUALIZATION

Several tools in FEMLAB can improve the quality of a visualization. You can control the resolution of geometry objects as well as select the graphics renderer and lighting method.

Lighting

The application of lighting can significantly improve the quality of visualization for a geometry object, a mesh plot, or a solution plot. Lighting properties appear on the **Orbit/Pan/Zoom** page in the **Visualization/Selection Settings** dialog box and also on the orbit/pan/zoom toolbar.



The FEMLAB graphical interface provides two light sources. The **Headlight** is positioned at the observer and directed towards the target. The **Scenelight**, on the other hand, radiates from a far distance, simulating the lighting of the sun. To set the scenelight's position use the **Orbit Scenelight** function.

You can also vary the lighting model and the reflection type for best result.

- *Flat* lighting produces uniform color across each triangular face of an object. Select this method to view faceted objects.
- *Gouraud* lighting calculates colors at the vertices and interpolates colors across faces. Select this method to view curved surfaces.
- *Phong* lighting interpolates the normals of all vertices across each face and calculates the reflectance at each pixel. Select this choice to view curved surfaces.

Phong lighting generally produces better results than Gouraud lighting but takes longer to render.

To change the reflection on face objects you can choose among four different materials in the **Reflection** drop-down list box: *Default*, *Dull*, *Metal*, and *Shiny*.

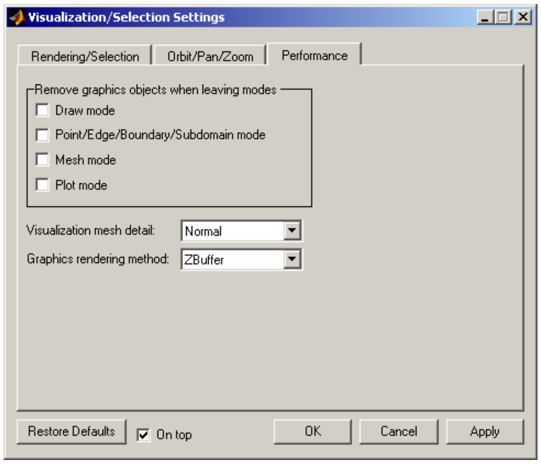
Renderer

On the **Performance** page you can choose between two graphics renderers: *Z-buffer* and *OpenGL*. OpenGL operates several times faster than Z-buffer, assuming the graphics card on your platform supports it.

The Z-buffer renderer generally generates higher resolution plots and the Phong lighting model isn't available for OpenGL rendering, which means that you can sometimes obtain higher quality plots using Z-buffer. Note also that using OpenGL requires more memory on your graphics card.

Geometry Mesh Detail

On the **Performance** page of the **Visualization/Selection Settings** dialog box you specify the geometry mesh detail. Options include **Fine**, **Normal**, **Coarse**, or **Extra Coarse**. This setting controls the number of polygons FEMLAB uses to render geometry surfaces. Note that the geometry mesh is separate from the mesh FEMLAB uses to compute problem solutions; you lose no solution accuracy by changing to a coarser geometry mesh.


Visualization/Selection Settings

Rendering/Selection | Orbit/Pan/Zoom | Performance

Remove graphics objects when leaving modes
- ☐ Draw mode
- ☐ Point/Edge/Boundary/Subdomain mode
- ☐ Mesh mode
- ☐ Plot mode

Visualization mesh detail: Normal

Graphics rendering method: ZBuffer

Restore Defaults | ☑ On top | OK | Cancel | Apply

RENDERING LARGE GEOMETRY OBJECTS

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

An obvious way to conserve memory is to render as little as possible of the geometry. Face objects are the most memory demanding objects. Open the **Visualization/ Selection Settings** dialog box on the **Rendering/Selection** page. Now remove all face rendering either by unchecking the **Render**, **Highlight**, and **Clickable** check boxes in the **Faces** frame, or by unchecking the check box in the frame title (the fastest way to completely disable all face rendering). To get a *wireframe plot* of the geometry, make sure the **Render** check box in the **Edges** frame is checked.

Remember that once FEMLAB has rendered a geometry using faces, it maintains the larger geometry mesh level needed for faces until it regenerates the geometry, even if you turn off all face rendering. See the `mlevel` property in the `meshinit` entry in the *Reference Manual* for more information about the mesh level. To be certain that you're working with the smallest possible mesh, turn off face rendering before creating the geometry or triggering a regeneration of the geometry. An easy way to do so is by changing **Geometry mesh detail** on the **Performance** page. This selection also affects the amount of memory consumed; a coarse geometry mesh requires less memory than a fine geometry mesh.

The same page holds a frame named **Remove graphics objects when leaving modes**. Here you can select to delete all graphics objects when they're no longer needed in order to free memory during the modeling process.

If you run out of memory on your graphics card when using the OpenGL renderer, you can either shrink the FEMLAB figure window until it's small enough for the graphics memory to render, or you can switch to Z-buffer rendering, which occupies less graphics memory than OpenGL does.

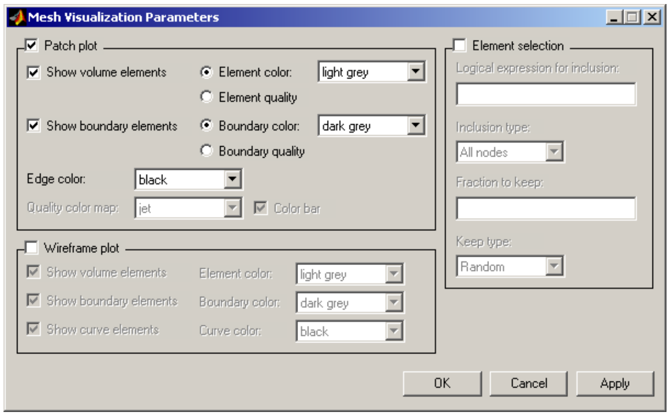
Another way to save computer memory is to reduce the number of undo steps for the geometry modeling and also the number of undo steps for mesh generation. This is done in the **Customize** dialog box found on the **Options** menu.

MESH VISUALIZATION

So far this chapter has dealt primarily with the visualization of a geometry whereby this section applies only to mesh mode and the finite element mesh.

Open the **Mesh Visualization Parameters** dialog box by selecting **Visualization Parameters...** from the **Mesh** menu. Here you find several tools for visualizing the finite element mesh. The **Patch plot** frame decides whether or not to plot the faces of the mesh's boundary elements and volume elements. You can turn off all patch plots by unchecking the check box in the frame title. Similarly, toggle the activation of boundary and volume elements independently of each other using the **Show boundary elements** and **Show volume elements** check boxes.

It is possible to set the faces to take on one of the predefined colors in the drop-down list boxes. Alternatively, you can have elements assume a different color depending on their quality. Choose the quality color map in the **Quality color map** drop-down list box in the lower part of the frame. To display a color bar select the **Color bar** check box to the right. The **Edge color** drop-down list box determines the color of the edges of the patches.


Mesh Visualization Parameters

[✓ Patch plot]
[✓ Show volume elements] — [✓ Element color:] light grey — Element quality
[✓ Show boundary elements] — [✓ Boundary color:] dark grey — Boundary quality
Edge color: black
Quality color map: jet — [✓ Color bar]

Wireframe plot
[✓ Show volume elements] — Element color: light grey
[✓ Show boundary elements] — Boundary color: dark grey
[✓ Show curve elements] — Curve color: black

Element selection
Logical expression for inclusion:
Inclusion type: All nodes
Fraction to keep:
Keep type: Random

OK Cancel Apply

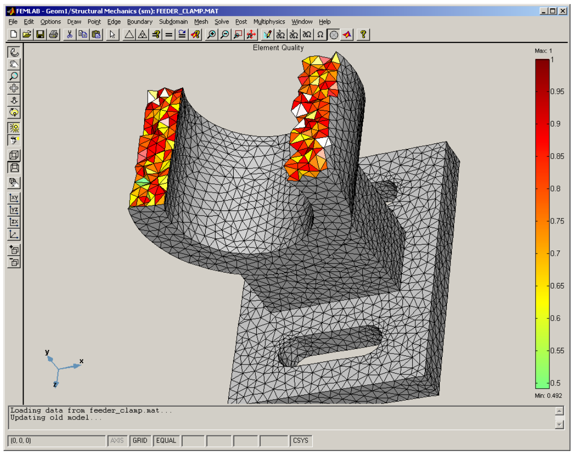
The **Wireframe plot** frame appears on the lower left. Here you can choose to plot a wireframe of the mesh elements. Next to the frame title you can turn all wireframe plots on or off with one click. Inside the frame, you plot volume elements, boundary elements, and curve elements by checking the **Show volume elements**, **Show boundary elements**, and the **Show curve elements** check boxes. Other options allow you to set the color of different wireframe plots individually.

To the right, in the **Element selection** frame, appears a set of tools that determine which parts of the mesh to visualize. In the **Logical expression for inclusion** edit field you can enter a MATLAB expression in terms of the space coordinates, `x`, `y` and `z` that specifies which parts of the mesh to include in the plot.

This edit field works in conjunction with the **Inclusion type** drop-down list box where you can choose between three options. The *All nodes* option covers any elements in which all of their nodes satisfy the inclusion expression. The *Any node* option includes elements in which at least one of their nodes satisfy the inclusion expression. The *Xor* alternative includes elements in which some but not all of their nodes satisfy the inclusion expression.

In the **Fraction to keep** edit field you specify a fraction of the total amount of volume elements to plot. Depending on settings in the **Keep type** drop-down list box, FEMLAB either randomly selects the fraction of elements or it works with the worst fraction in terms of element quality. You can disable these settings by unchecking the check box in the frame title for element selection. These settings apply on top of the suppression of boundaries and subdomains (see “Suppressing Boundaries and Subdomains” on page 1-107).

The following picture illustrates how to use the logical expression for inclusion feature on the `feeder_clamp` model from the *Model Library*. This example plots only those elements with all their nodes fulfilling the inclusion condition `y<40`. From it you can see the quality of the volume elements in the open cuts.


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Element Quality
Max: 1
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
Min: 0.492
y
x
z
(0, 0, 0)
AXIS GRID EQUAL CSYS

```
Loading data from feeder_clamp.mat...
Updating old model...
```

The 3D View

When working with 3D geometry objects, it’s important to be able to easily change the view so as to take close-up examination of a geometry part or select objects hidden behind other objects. This section applies to all modeling modes.

When you look at graphics objects displayed in a 3D axes, you’re viewing the scene from a particular location in space (in FEMLAB known as the camera position) and with a particular orientation (the camera up-vector). The center of the scene is the camera target, which appears to be near or far away depending on the view angle, which you set with the camera zoom lens.

CHANGING THE VIEW

You can completely define a 3D view in terms of camera position, camera up-vector, target position, viewing angle and projection type. You can select between two possible projection types on the **Orbit/Pan/Zoom** page of the **Visualization/Selection Settings** dialog box. The *orthographic* option projects graphics objects so as to maintain their actual size and the angles between them; object size does not depend on relative distance to the camera. The *perspective* option projects objects so as to make those further from the camera appear smaller. This projection type is useful for displaying realistic views of real objects.

Mouse Movement Actions

On the same page you can find a frame for **Mouse movement action**, which defines what happens when you click-and-drag in the 3D main axes. Selecting **Orbit** has the camera orbit around the target position, **Pan** orbits the target about the camera position, while **Zoom** changes the viewing angle. **Dolly up/down** moves the camera and target position up and down, while **Dolly in/out** moves the camera towards or away from the target position. This powerful tool can, for instance, make it possible to examine the inside of a geometry object. Selecting the **Orbit scenelight** radio button orbits the scenelight around the scene to get another lighting angle.

The frame also contains a **None** radio button that disables click-and-drag functionality and enables rubber band box selection. Further, the **Move as box** check box deactivates geometry updates during mouse dragging. Instead, the graphical interface represents the geometry by a bounding box and follows the mouse movement. The screen redraws the geometry only when a dragging session is finished, that is, when you release the mouse button. This functionality is useful when working with complex geometries that are heavy to render. All the controls in the mouse movement action frame have corresponding buttons in the orbit/pan/zoom toolbar.

The mouse movement action and selection functionality are both active concurrently in the 3D graphical interface (see the section “Click-and-Drag in 3D” on page 1-102).

Fine Tuning the View

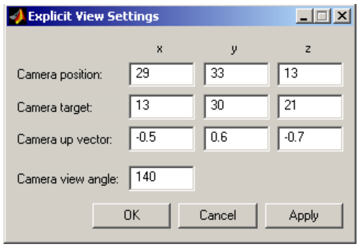
To fine tune the view using the tools described above, you can press the right mouse button instead of the left (or use **Ctrl**-clicking). This action improves precision by a factor of four compared to left-clicking.

Continuous Rotation

For **Orbit**, **Pan**, and **Orbit Scenelight**, a **Shift**-click (or click on the middle button in a 3-button mouse) starts a *continuous rotation* you interrupt by clicking anywhere in the graphical interface.

Explicit View Settings

If it becomes difficult to attain a specific view with the tools on the FEMLAB graphical interface, you can define a particular view by using the **Explicit View Settings** dialog box. Open it from within the **Orbit/Pan/Zoom** page in the **Visualization/Selection Settings** dialog box.


Explicit View Settings

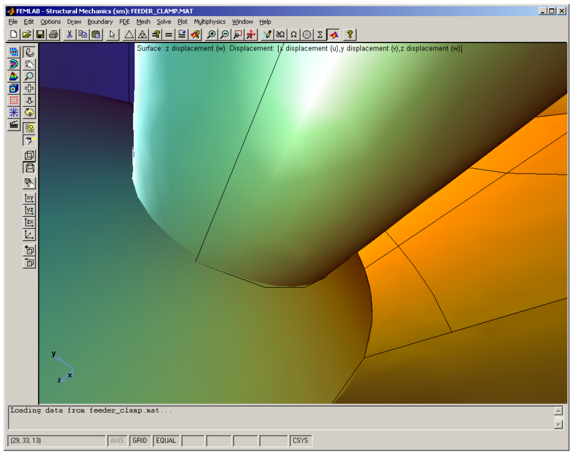
	x	y	z
Camera position:	29	33	13
Camera target:	13	30	21
Camera up vector:	-0.5	0.6	-0.7
Camera view angle:	140		

Example

Let us look at how to use explicit view settings and lighting settings to examine the solution plot of the example “Deformation of a Feeder Clamp” on page 2-372. in the *Model Library*, from the inside. Open the **Model Library** page in the **Model Navigator**. Select the `feeder_clamp` model from the `Structural_Mechanics` branch in the `FEMLAB` directory and press **OK**.

Go into post mode and open the **Plot Parameters** dialog box, deselect **Color bar** on the **Surface** page and click **OK**.

Open the **Visualization/Selection Settings** dialog box from the **Options** menu. On the **Orbit/Pan/Zoom** page in the **Lighting** frame, select **Scenelight**. On the **Performance** page select the OpenGL renderer. Select perspective projection from the **Perspective** drop-down list box on the **Orbit/Pan/Zoom** page and press the **Apply** button. Press the **Explicit View Settings...** button to change the 3D view. Enter the coordinates `(25,45,20)` in the fields for **Camera position** and the coordinates `(15,30,20)` for the **Camera target** position. Set **Camera up vector** to `(-1,1,-1)`. Finally, enter a view angle of 140 degrees and click **OK** to get the following fish-eye view from inside the middle cylindrical part of the clamp.


FEMLAB - Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Boundary PDE Mesh Solve Plot Multiphysics Window Help
y
z
x
(29, 33, 13)
AXIS
GRID
EQUAL
CSYS

Surface: z displacement (w) Displacement: [x displacement (u),y displacement (v),z displacement (w)]

```
Loading data from feeder_clamp.mat...
```

You can achieve the same effect using the **Dolly In/Out** and **Zoom** buttons on the orbit/pan/zoom toolbar.

Object Selection Methods in 3D

The selection mechanism in FEMLAB relies on clicking 3D objects in the main axes of the graphical interface. In other words, you do not have to call up any dialog boxes to make object selections. Throughout the 3D graphical interface, similar principles apply for selecting a variety of items including geometry objects, subdomains, boundaries, edges, and points.

SINGLE SELECTION

You perform a single selection using the left mouse button, and the interface highlights selected objects in red. You select objects by clicking on them or by clicking on adjacent objects. Sometimes this method can be faster and easier than clicking directly on the desired object. For example, you might select a face object by clicking on the face or its label, on its adjacent vertices or edges, or on the labels of its adjacent subdomains. See "Adjacency Selection Methods" on page 1-102 for details.

To be able to click on an object, you must set its object type to clickable. The rendering/highlighting settings also entail certain restrictions (see "Clicking" on page 1-87).

OVERLAPPING OBJECTS

When you click on a 3D geometry within the main axes, occasionally several objects might be overlapping at the cursor. Thus the intention of the click is ambiguous. How does FEMLAB handle this situation?

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

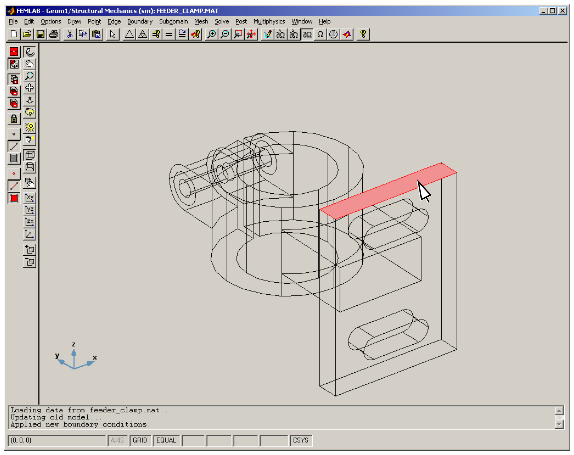
Ordering within an object type exists only for faces. Faces are initially sorted based on their area; the smallest face has highest priority, so the first click on overlapping faces selects the smallest one. A subsequent click selects the second smallest face under the cursor, and so on. In other words, multiple clicking on overlapping faces *cycles* among them.

Object precedence also means it's easy to accidentally click on labels, vertices or edges when you're aiming for a face object. While you're releasing the mouse button, look at the coordinate field on the **Status** bar to see which object you're actually selecting; the selection action doesn't trigger until you release the mouse button. See "Click-and-Drag in 3D" on page 1-102 for a description of how to cancel a pending selection.

Example

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

Open the **Model Library** page in the **Model Navigator**. Select the `feeder_clamp` model from the `Structural_Mechanics` branch in the `FEMLAB` directory and press **OK**. Enter boundary mode by pressing the **Boundary Mode** button in the main toolbar. Press the **Reset 3D View** button in the orbit/pan/zoom toolbar to get the default 3D view and turn off the lighting by deselecting the **Headlight** button. Select the **Select** button in the main toolbar to disable mouse movement action during this example. Move the mouse cursor to the position indicated by the arrow in the figure below and click once with the left button. As illustrated below, two faces are overlapping at the cursor position, and you've selected the smaller of the two.


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Loading data from feeder_clamp.mat...
Updating old model...
Applied new boundary conditions.
```

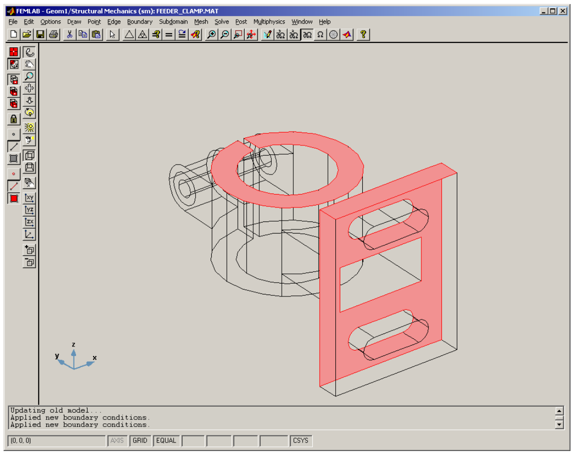
The face and its adjacent edges are highlighted in red. Click again using the left mouse button, without moving the mouse; doing so selects the larger face behind the previously selected face. Continuous clicking on the same spot cycles between the two faces.

MULTIPLE SELECTIONS

By holding down the **Shift** key during selection, you can add another object to the current selection. This feature allows for quick multiple selections. An alternate way to accomplish the same task is by using the middle button on a 3-button mouse because its assigned function is equivalent to holding the **Shift** key. You might find, though, that making multiple selections with **Shift**-clicking isn't always satisfactory when several objects are overlapping. To avoid such difficulties you can confirm selections as described in the next section below. To select all objects, use **Select All** from the **Edit** menu or use the shortcut **Ctrl-a**.

Example

Continuing from the previous example, hold down the **Shift**-key and click two times with the left mouse button on the same spot as before; both overlapping faces are now selected simultaneously. You can move the mouse to other faces and **Shift**-click on them to add them to the current selection. If you do not hold the **Shift** key down, you end up with just a single selection. The following figure shows a selection of four faces made in this fashion.



```
Updating old model...
Applied new boundary conditions.
Applied new boundary conditions.
```

You can also employ a *rubber band box* to select objects. Press the **Select** button in the main toolbar to enable rubber band box selection. Click and drag while holding the mouse button to create a rubber band box. The selection applies only to objects that the rubber band box surrounds completely. **Shift**-clicking adds the rubber band box selection to the current selection, while **Ctrl**-clicking performs the logical and operation with the current selection (only presently selected objects inside the rubber band box will be selected).

CONFIRMING SELECTIONS

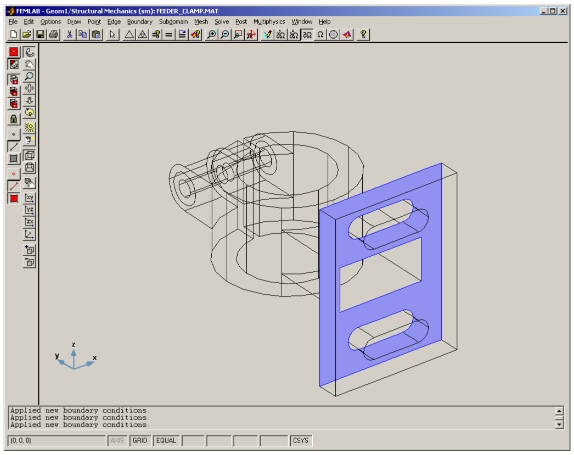
The confirmation feature is a way to *lock in* the current selection so that the interface keeps it active until you explicitly deselect it. Previously selected objects (highlighted in red according to the visualization settings) turn blue when confirmed. You confirm and thereby lock in a selection by clicking anywhere in the main axes, on the geometry, or outside the geometry either using the right mouse button or by holding the **Ctrl** key when clicking; an alternate is to press the **Confirm Selection** toolbar button in the **Selection** toolbar. Locking in a selection prevents it from being removed by accident, a feature that proves useful when you're making multiple selections among many overlapping objects.

You can reselect confirmed selections in order to mark them for removal. Their appearance in green indicates this action. When you confirm a green selection the objects become deselected. Hence, red and green selections can exist at the same time as blue selections. A confirm-click at this point removes green selections, and red selections turn blue.

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

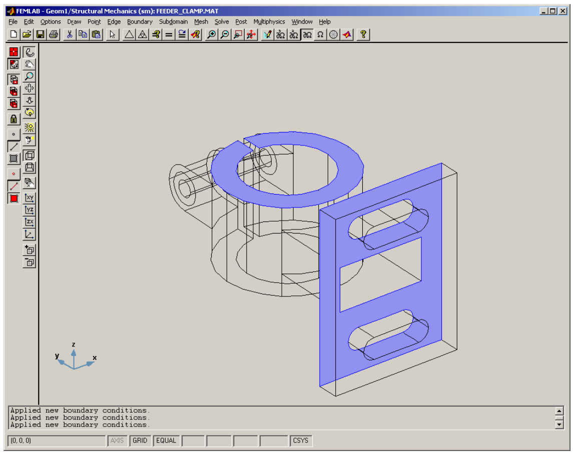
Example

Proceeding from the previous example, click on the same spot as in the single-selection example until you've selected the larger of the two faces. Confirm the current single selection by clicking with the right mouse button (or **Ctrl**-clicking) anywhere in the graphical interface main axes. The selected face now turns blue.


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Applied new boundary conditions.
Applied new boundary conditions.
Applied new boundary conditions.
```

The selection of this face segment is now locked, and it won't become deselected when you make additional single or multiple selections. Click on the top face of the middle cylindrical part of the clamp using the left mouse button and also confirm that selection by right-clicking. The selection now looks like this:


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

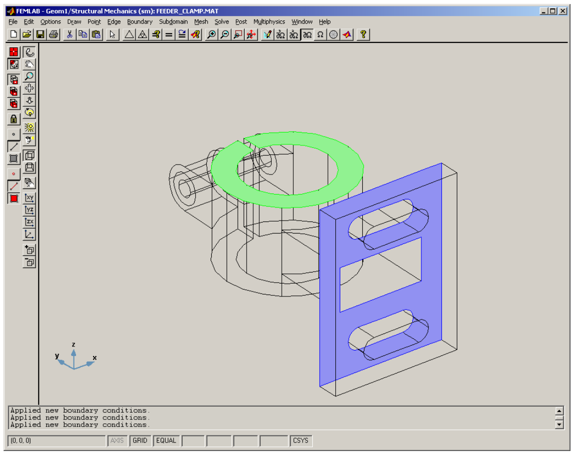
```
Applied new boundary conditions.
Applied new boundary conditions.
Applied new boundary conditions.
```

DESELECTION

Clicking outside the geometry with the left mouse button deselects all unconfirmed selections whereas confirmed selections remain intact. Deselection of a subset of confirmed (blue) selections is possible by selecting the desired objects with the left mouse button (the blue objects turn green) and removing them with a **Ctrl**-click or by clicking the right mouse button. To deselect everything, including confirmed selections, use **Deselect All** from the **Edit** menu or the shortcut **Ctrl-d**.

Example

Continuing from the previous example, left-click on the bottom face segment of the cylindrical subdomain in the middle of the clamp. You now have two blue faces and one red face selected. Click outside the geometry using the left mouse button and the red selection disappears. However, the confirmed selections remain. To remove the confirmed selection on the top of the cylindrical subdomain, click on the blue circular-shaped face until it turns green. You will probably need to click twice because the first click might result in selection of an underlying face object. The graphical interface now has the following appearance:


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Applied new boundary conditions.
Applied new boundary conditions.
Applied new boundary conditions.
```

At this point, clicking in the main axes using the right mouse button (or **Ctrl**-clicking) removes the green selection. Now only one selection remains. **Shift**-click anywhere on the geometry to add more selections. Press **Ctrl-d** or select **Deselect All** from the **Edit** menu to deselect all faces.

LIST BOX SELECTIONS

In many dialog boxes, such as in the one for specifying boundary conditions, you can select objects from within a list box. It is thus important to note that a direct correspondence exists between selections that are active in a list box and selections in the graphical interface main axes. Further, any list box selection overrides confirmed selections in the graphical interface. Shortcuts for **Select All** and **Deselect All** also work in list boxes.

DOUBLE-CLICKING

Double-clicking an object opens its associated dialog box. In 3D you do so with the right mouse button; recall that a double-click with the left button is reserved for rapid cycling between overlapping objects and adjacent objects when the target of selection is ambiguous. You can use double-click in draw mode to open the object properties dialog box, but only if a single geometry object is selected. In point, edge, boundary and subdomain mode, double-clicking opens the point settings, edge settings, boundary settings, and subdomain settings dialog boxes, respectively.

CLICK-AND-DRAG IN 3D

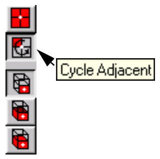
In FEMLAB you can make selections and rotate a geometry without pressing any graphical interface buttons to change between the two operations. It is very convenient to be able to work with both at the same time, such as using both rotation and selection. In fact, any mouse-movement action available in the orbit/pan/zoom toolbar can be active at the same time as selection. Selection is always available, but you can deactivate mouse movement actions by pressing the **Select** button in the main toolbar or by pressing the selected mouse movement action button once again. This enables rubber band box selection.

In addition, the combination of the two makes it possible to cancel a pending selection. When you click on an object in the graphical interface main axes while holding the left mouse button, the object's label appears in the coordinate field on the **Status** bar. The interface doesn't actually select the object until you release the mouse button. If you move the mouse before releasing the mouse button, the interface never selects the object and it instead executes the chosen mouse movement action. You can verify this new condition by looking at the **Status** bar, where the name of the mouse movement action now replaces the object label, or by looking at the mouse pointer, which now takes on the shape of the current mouse movement action icon.

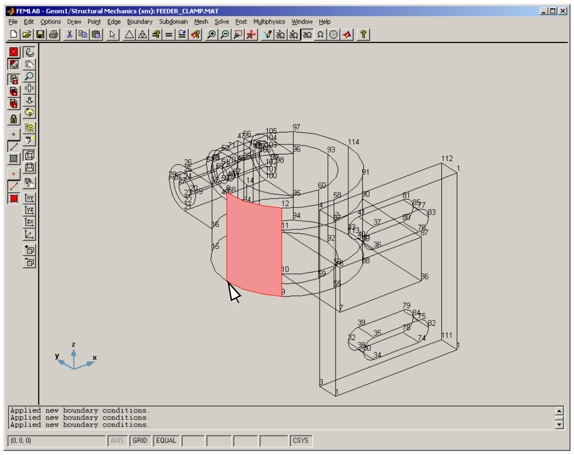
ADJACENCY SELECTION METHODS

In FEMLAB you can select 3D objects by clicking on them or on their adjacent objects. However, an adjacent object is often next to more than one object. In such a case you can either choose to select all adjacent objects or to cycle among them (multiple clicking selects adjacent objects one at a time). These two methods of selection are called adjacency selection methods, and controls for them appear on the **Rendering/Selection** page in the **Visualization/Selection Settings** dialog box and at the top of the select toolbar.

Example

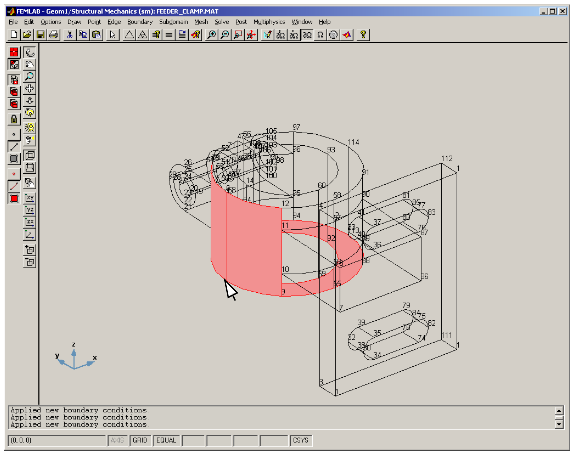


Continuing from the previous example, go to boundary mode and turn on vertex rendering by selecting the **Render Vertices** button in the select toolbar. At the top of the select toolbar, make sure that the **Cycle Adjacent** button is selected in the first group of buttons. Select **Show Vertex Labels** from the **Labels** submenu on the **Options** menu. With the left mouse button click once on vertex 5 (to the lower left of the middle cylindrical object) and you should see the following.


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Applied new boundary conditions.
Applied new boundary conditions.
Applied new boundary conditions.
```

One of the face segments adjacent to the selected vertex is now selected. Click on vertex 5 again. This action selects another of the adjacent faces. A third click on the same vertex selects the last of the three adjacent faces, and repeated clicking continues the cycle from the first selected face. Deselect the current selection by clicking outside the geometry. Select the **All Adjacent** toolbar button at the absolute top of the select toolbar. Click on vertex 5 again and all the adjacent faces are selected.


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Applied new boundary conditions.
Applied new boundary conditions.
Applied new boundary conditions.
```

BOUNDARY SELECTION METHODS

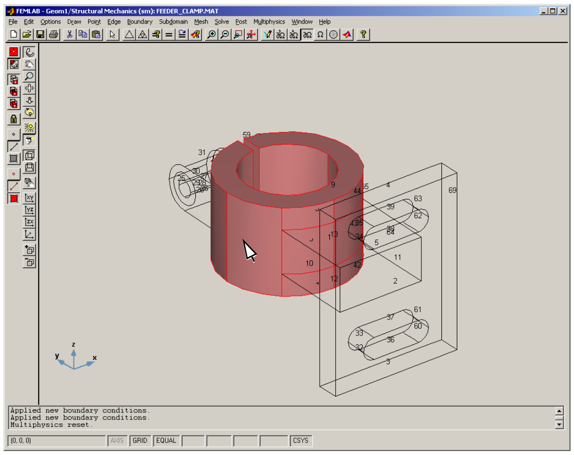
When you click a face object in boundary mode, such as when selecting boundaries, the clicked object is normally selected and highlighted according to your visualization settings. However, a few shortcuts can help get a desired multiple selection. The definitions of these shortcuts appear in the **Boundary selection methods** found on the **Rendering/Selection** page in the **Visualization Selection Settings** dialog box. The selection methods have corresponding toolbar buttons in the select toolbar.

- *Normal face selection* selects only the face segment being clicked. Multiple clicks cycle among overlapping faces.
- *All faces with same adjacent subdomains* selects all face segments in the geometry that have the exact same adjacent subdomains as the clicked face. Multiple clicks cycle among overlapping faces.
- *All faces adjacent to subdomain* selects all faces on the subdomain that are adjacent to the clicked face. If the face is adjacent to two subdomains, repeated clicking on it cycles between the subdomains. In addition, multiple clicking on overlapping faces cycles among the faces.

Example

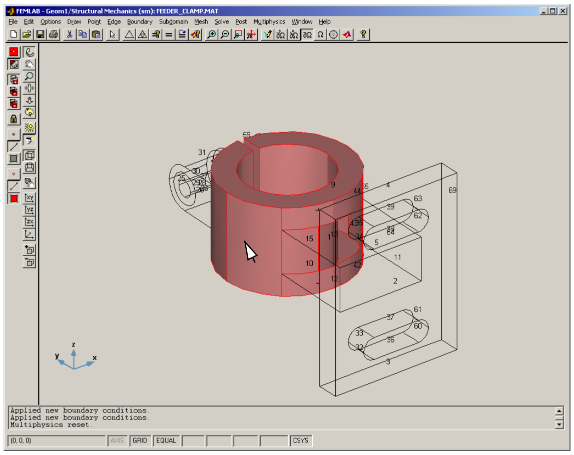
Continuing from the previous example, open the **Visualization/Selection Settings** dialog box from the **Options** menu. On the **Rendering/Selection** page, uncheck vertex labels in the **Labels** frame and check the **Face Labels** check box. Make sure that the **Normal selection** radio button is selected in the **Boundary selection method** frame before clicking **OK**.

Select **Enable Borders** from the **Boundary** menu. Click on face 6 using the left mouse button (you can also click on the label if desired). Doing so results in the single selection of face 6 or any underlying face object. Now select the **Cycle All Adjacent to Subdomain** button in the select toolbar. Click outside the geometry to remove the current selection. Click on the label for face 6 (just to make sure you won't select an underlying object by mistake) and all the faces on the cylindrical subdomain get selected. Click on the **Headlight** button in the orbit/pan/zoom toolbar to get a better 3D effect.


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Applied new boundary conditions.
Applied new boundary conditions.
Multiphysics reset.
```

Change the boundary selection method by clicking the **All with Same Adjacent Subdomains** button. Click outside the geometry to remove the current selection and click on the label for face 6 again. This time you should see a hole in the lateral area of the selection:


FEMLAB - Geom1/Structural Mechanics (sm): FEEDER_CLAMP.MAT
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Applied new boundary conditions.
Applied new boundary conditions.
Multiphysics reset.
```

The difference between this selection and the previous one arises from the fact that the differing faces are adjacent not only to the cylindrical subdomain but also to another adjoining subdomain. Note that the differing faces are so-called *internal* *borders*, in other words, they lie between two subdomains. You can select such faces only when you've selected **Enable Borders** in the **Boundary** menu.

Suppression of Objects in 3D

If you want to select objects buried deep inside a complex geometry it might be useful to suppress parts of the geometry. Another way to do so is with the tools on the orbit/pan/zoom toolbar. They allow you to change the 3D view and thereby make it easier to reach the desired objects. The suppression feature makes selected objects invisible according to the visualization settings.

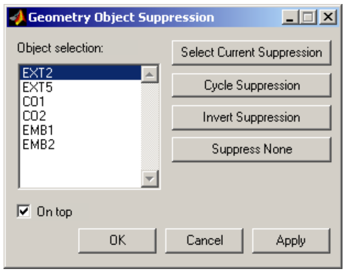
SUPPRESS ADJACENT OBJECTS

On the **Rendering/Selection** page of the **Visualization/Selection Settings** dialog box you can specify which parts of suppressed objects to display. Each of the **Vertices**, **Edges**, and **Faces** frames have a **Suppress Adjacent** check box. If you activate this selection for an object type, adjacent objects of that type disappear from any object you suppress—provided it's not adjacent to another object that isn't suppressed, as well.

The label of a suppressed object is always invisible. You can highlight a suppressed object if you haven't suppressed all of the highlighted object types. If a suppressed object is highlighted in any way, you can then select it by clicking on an adjacent object; otherwise you cannot select it by clicking. The only way to select a completely suppressed object is from a list box, apart from the **Select All** menu item which selects all objects including suppressed ones.

SUPPRESSING GEOMETRY OBJECTS

In 3D draw mode you can suppress geometry objects. Do so from the **Geometry Object Suppression** dialog box that opens when you choose **Suppress Object(s)...** from the **Draw** menu. The geometry object suppression affects only draw mode.



On the left side of the dialog box you select those objects you wish to suppress. You can also select objects by clicking in the graphical interface main axes. Pressing **Apply** or **OK** suppresses the selected objects. Now those objects are more or less invisible depending on the visualization settings. The **Select Current Suppression** button is a useful tool to quickly select all suppressed objects. The **Cycle Suppression** button cycles the current suppression by extending the selection in the list box downwards one step and removing a selection from the top. The **Invert Suppression** button suppresses objects not currently suppressed and vice versa. To remove all suppressions, press the **Suppress None** button or deselect all objects in the list box by **Ctrl**-clicking and pressing the **Apply** button.

SUPPRESSING BOUNDARIES AND SUBDOMAINS

By selecting **Suppress Boundaries...** from the **Boundary** menu you can open the **Boundary Suppression** dialog box. This dialog box looks like the geometry suppression dialog box, but its list box contains boundary labels instead of geometry object labels. The dialog box works in the same way as the geometry suppression dialog box. Boundary suppression is available in boundary mode, mesh mode and post mode. Similarly, on the **Subdomain** menu you can select **Suppress Subdomains...** to open the **Subdomain Suppression** dialog box. Subdomain suppression is available in subdomain mode, mesh mode, and post mode.

In mesh mode and post mode you cannot get visual feedback on the current selection in the graphical interface main axes. Nevertheless, you can still open the boundary and subdomain suppression dialog boxes to select and change the suppression of boundaries and subdomains. In mesh mode, element selections in the mesh visualization settings are added to the current suppression.

Modeling with FEMLAB

The objective of this section is to give an introduction to the FEMLAB modeling process by working through a number of models using the GUI. All models are set up from the start. You work your way through all the stages of modeling, from geometry creation to the postprocessing of solution data. Communication with MATLAB and Simulink is also described.

Style Conventions for the Model Descriptions

The basic flow of actions is indicated by the way the toolbar buttons and the menus are ordered from left to right. You work your way from left to right in the process of modeling, defining, solving, and postprocessing your problem using the FEMLAB GUI. Considering this, a certain style convention, or format, is used throughout the documentation for describing the models in the *User's Guide and Introduction*, as well as in the *Model Library*. The format includes a number of headlines corresponding to each major step in the modeling process. The headlines also roughly correspond to the different GUI modes.

MODEL NAVIGATOR

The **Model Navigator** starts on the **New** page when initializing a model, if you have not altered the preferences. Here you specify the space dimension (1D, 2D, or 3D), application mode, and number of dependent variables. On the **Multiphysics** page, you can also set up a combination of application modes. See the section "The Model Navigator" on page 4-4 for details on the Model Navigator dialog box.

OPTIONS AND SETTINGS

This section covers the basic settings, for example, the axis or grid spacing settings. These settings can usually be made with the **Options** menu or by double-clicking on the status bar. See the sections "Options Menu" on page 4-34 and "The Status Bar" on page 4-201 for details. You may also need to use the **Add/Edit Constants** dialog box to enter model parameters.

DRAW MODE

Here you set up the geometry of the model. You need to know how to use the draw menu and the draw toolbar. The details on this can be found in the sections "1D Draw Menu" on page 4-71 for 1D geometry modeling, "2D Draw Menu" on page

4-54 for 2D geometry modeling, and "3D Draw Menu" on page 4-59 for 3D modeling. It is advisable to get acquainted with 2D modeling before looking at the 3D modeling section. You can also refer to "The 2D Draw Toolbar" on page 4-187 and "The 3D Draw Toolbar" on page 4-189. For general information on the CAD modeling features and how to use them, see "Geometry Modeling" on page 1-170.

BOUNDARY MODE, EDGE MODE, POINT MODE

In these modes, a standard table format is used to describe the settings in the corresponding **Specify Boundary/Edge/Point Settings** dialog. For details on the corresponding menu items, see the sections "Boundary Menu" on page 4-75, "Edge Menu" on page 4-73, and "Point Menu" on page 4-72. The physics mode parameters in the dialog boxes are discussed in "Application Mode Reference" on page 4-203.

SUBDOMAIN MODE

In subdomain mode, a standard table format is used to describe the settings in the **Subdomain Settings** dialog. Details on the subdomain menu items can be found in the section "Subdomain Menu" on page 4-81. The physics mode parameters in the dialog box are discussed in "The Application Modes" on page 1-239 and "Application Mode Reference" on page 4-203.

MESH MODE

Here you mesh the geometry of the problem. Normally you just need to click the mesh buttons on the main toolbar. In some cases, you need to use the **Mesh** menu described in the section "Mesh Menu" on page 4-90.

SOLVING

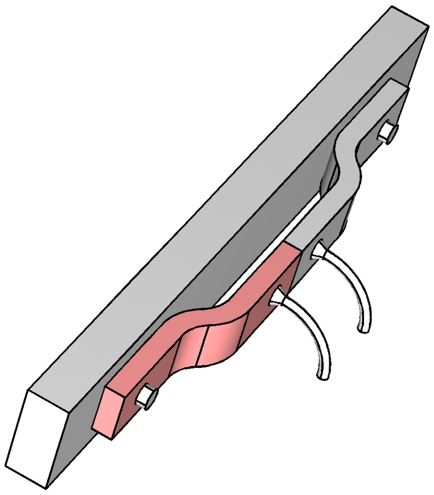
Often you only have to press the **=** button on the main toolbar. Sometimes you need to open the **Solver Parameters** dialog box. See the section "Solve Menu" on page 4-102 for details.

POST MODE

The visualization settings are made here. You need to be familiar with the **Plot Parameters** dialog box. See the sections "2D Post Menu" on page 4-127 and "3D Post Menu" on page 4-156 for details.

Thermo-Electric Heating in a Bus Bar

This model examines the relationship between current throughput and temperature inside a solid copper bar. Such bus bars are used as conductors in industrial environments requiring very high currents, for example, aluminum smelters and certain chemical plants.



The thermo-electric coupling is two-way: volume currents inside the bus bar, which are proportional to the conductivity, act as a distributed heat source, while at the same time the temperature affects the metal's conductivity. These kinds of dependencies makes it necessary to create a multiphysics model.

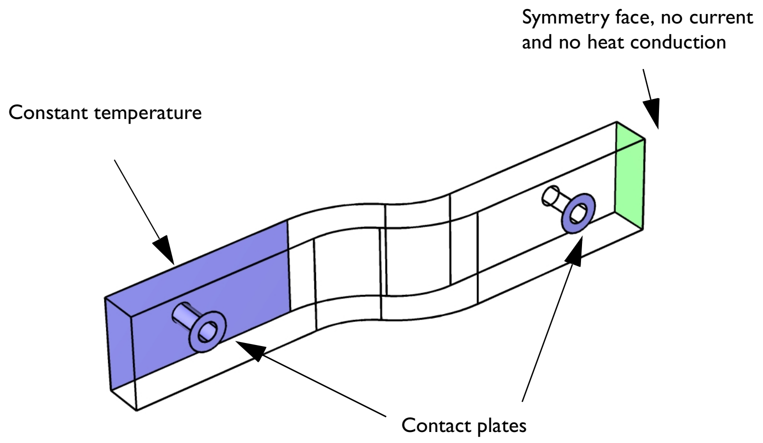
As a first step, the stationary current distribution at constant temperature will be modeled. It is highly recommended that you follow along with the description of the modeling process in this example even if you're not an electromagnetics expert. The discussion focuses on how to use FEMLAB's graphical interface rather than on the underlying physics.

The second part of the example adds a heat transfer equation, bidirectionally coupled to the stationary current model. This part is primarily about multiphysics modeling, showing how to connect phenomena from different fields of physics. It also shows how you define coupled variables, which can be, for example, an integral of the solution, evaluated on some part of the model geometry.

GEOMETRY AND BOUNDARY CONDITIONS

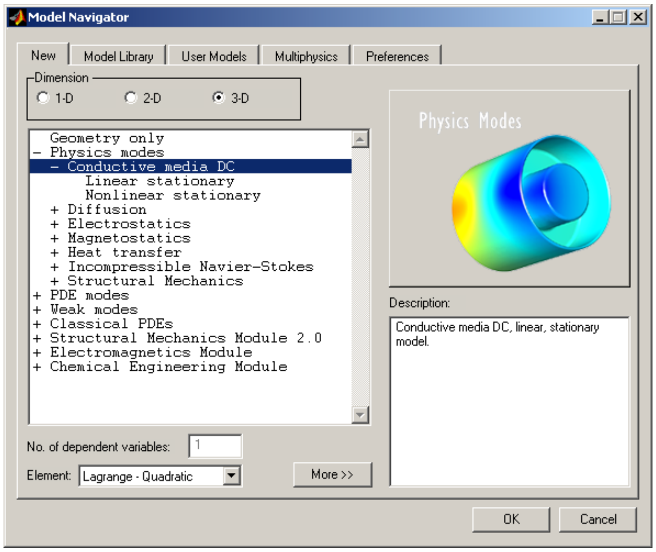
The bus bar is mounted as an intermediate step between two high voltage cables and a load. Because of the symmetry, it's only necessary to model one half of the true geometry.

The copper bus bar is electrically insulated everywhere, except for two contact plates around the mounting holes. The temperature is constant on the contact plates, and on the parts in contact with the load or mounting screw. All free faces have heat transfer coefficients corresponding to free air convection, but the symmetry face is isolated in all respects.



SETTING UP THE MODEL IN FEMLAB

Start FEMLAB either by typing `femlab` at the MATLAB prompt or by choosing FEMLAB from the start menu, if you have a Windows PC. Either option first invokes FEMLAB's Model Navigator.



```
   Geometry only
 - Physics modes
   - Conductive media DC
       Linear stationary
       Nonlinear stationary
   + Diffusion
   + Electrostatics
   + Magnetostatics
   + Heat transfer
   + Incompressible Navier-Stokes
   + Structural Mechanics
 + PDE modes
 + Weak modes
 + Classical PDEs
 + Structural Mechanics Module 2.0
 + Electromagnetics Module
 + Chemical Engineering Module
```

No. of dependent variables: 1

Element: Lagrange - Quadratic

Description: Conductive media DC, linear, stationary model.

- Go to the **New** page, and select the **3D** radio button. The space dimension should always be selected first, as the **Physics modes** available differ.
- Select **Physics modes**, and expose its submenu by double-clicking on the tree view control (the plus sign).
- Expand to the next submenu by double-clicking on **Conductive media DC** and select **Linear stationary**.
- The default finite elements are second order *Lagrange elements*, **Lagrange - Quadratic**. If you have limited memory and/or a very slow computer, you might want to change to linear elements. Do this by selecting **Lagrange - Linear** from the **Element** drop-down list.
- Press **OK** to confirm your choices and hide the **Model Navigator**.

Note This model can be found on the **Model Library** page under `FEMLAB/Multiphysics/bus_bar`.

At this point, the graphical interface opens up in the Conductive media DC application mode. You can always find out in which application mode the package is presently running because its name appears on the bar at the very top of the FEMLAB window.

Options and Settings

Entering the same numerical material data at different locations in the equations is tedious, and makes it difficult to change the model. Therefore, FEMLAB lets you introduce short-hand names for numerical constants and symbolic expressions. These are defined using dialog boxes accessed from the **Options** menu in the main menu bar at the top of the FEMLAB window.

Constants are global, that is, the same for all geometries and subdomains, and are always evaluated once—when you press the **Apply** or **OK** button in the **Add/Edit Constants** dialog box. They may depend on other user-defined constants, and contain any scalar MATLAB expression.

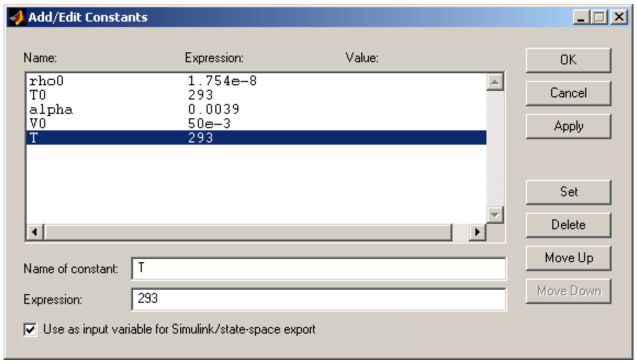
Symbolic *expressions* are added and modified using the **Expression Variable Settings** dialog box, also in the **Options** menu. Expression variables can have different meanings on different parts of the model geometry, and may contain any variables and expressions computable at the time the FEM problem is solved, for example the solution itself, other expression variables, the space variables, and time. As the expression definitions are linked to the geometry, they cannot be created until the geometry has been drawn.

The most powerful option is the *Coupling variables*. These are a generalization of expressions, which can define non-local couplings, including mesh transformations, integrals over subdomains, and projections. The coupling variables are also the foundation of FEMLAB's *Extended Multiphysics* capabilities.

This model, in the end, uses all three kinds of variables. Now, before the geometry is drawn, you can enter some constants that will be needed later on.

- Select the **Options** menu at the top of the FEMLAB window and choose **Add/Edit Constants**.
- First add the resistivity of copper at room temperature: Enter `rho0` in the **Name of constant** field. Press **Tab** to move the cursor to the **Expression** field and enter `1.754e-8`. The value is saved when you press **Enter**, press the **Set** button, or otherwise leave the **Expression** field.

- Go on adding the reference temperature, `T0=293`; the temperature coefficient, `alpha=3.9e-3`; and the input voltage, `V0= 50e-3`.



Add/Edit Constants

Name:	Expression:	Value:
rho0	1.754e-8	
T0	293	
alpha	0.0039	
V0	50e-3	
T	293	

Name of constant: T

Expression: 293

☑ Use as input variable for Simulink/state-space export

- Finally, add a constant `T` with value `293`, that will represent the real temperature during the first part of the model. Later it will be replaced by the solution of the coupled heat equation. Press **OK**.

Draw Mode

With the constants set, your first task in running a simulation is to define the model's geometry. For this task you need to work in draw mode. Conveniently, the graphical interface automatically enters this mode when you exit the **Model Navigator** by pressing the **OK** button.

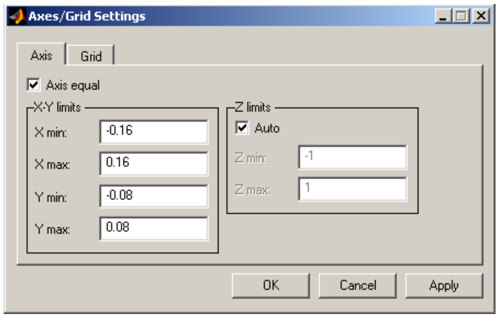
When drawing objects, it's often useful to take advantage of FEMLAB's snap-to-grid option, which is active by default (see page 4-35 of the *Reference Manual* for details on making sure it is set or how to change its setting). Snapping makes it easier to align the cursor to the grid and reach exact coordinate values.

The first part of this model is drawn in a 2D environment known as a *work plane*. See "Boundary Modeling" on page 1-180 on how to create and edit work planes. The default work plane is the x-y plane at $z=0$.

- Select **Draw** on the main menu bar of the FEMLAB window. Choose **Work plane 1** at the very bottom to enter the default work plane. The toolbar on the left changes, and the user interface is transfered to a 2D mode, almost identical to the draw mode when creating a 2D model.

It is often necessary to adjust the settings of the graphical interface. For example, you might wish to modify the axis limits if the geometry you wish to draw doesn't fit the default settings or, as in this example, is too small.

- On the menu bar at the top of the screen select the **Options** menu, and from the drop-down choices choose **Axes/Grid Settings...**


Axes/Grid Settings

Axis | Grid

☑ Axis equal

X-Y limits

X min:	-0.16
X max:	0.16
Y min:	-0.08
Y max:	0.08

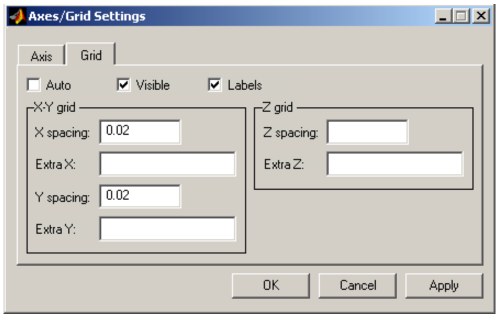
Z limits

☑ Auto

Z min:	-1
Z max:	1

OK | Cancel | Apply

- A dialog box opens up, and on the **Axis** page enter `-0.16`, `0.16`, `-0.08`, and `0.08` as **X min**, **X max**, **Y min**, and **Y max**, respectively.
- Click the tab for the **Grid** page and deselect the **Auto** check box. Enter `0.02` as both **X spacing** and **Y spacing**.


Axes/Grid Settings

Axis | Grid

☐ Auto ☑ Visible ☑ Labels

X-Y grid

X spacing:	0.02
Extra X:	
Y spacing:	0.02
Extra Y:	

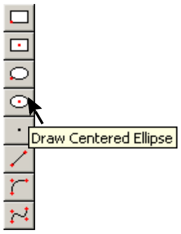
Z grid

Z spacing:	
Extra Z:	

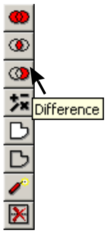
OK | Cancel | Apply

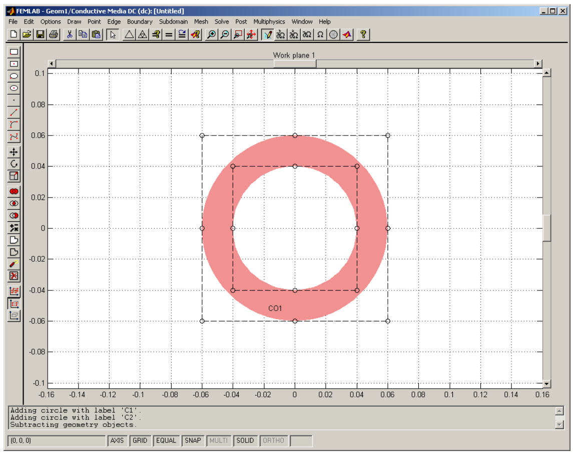
- Press **OK** to close the dialog box and to apply the settings.

Your next task is to draw one half of the S-shaped bus bar profile. This is most conveniently accomplished using Boolean operations on primitives. When drawing in a work plane or in a 2D model, you have access to rectangles and ellipses, as well as polygonal lines, and second and third order Bézier curves. Using these, together with Boolean operations, scaling, translation and rotation, almost anything can be drawn.



- First draw a circle with radius 0.06, centered at the origin. Press the **Draw Centered Ellipse** draw toolbar button. It is the fourth button on the draw toolbar, which is located on the left side of the graphical interface. Then, using the right mouse button, click at the origin and drag the mouse, keeping the button down, until the circle has the desired radius. Using the right mouse button ensures that a circle, and not an ellipse, is drawn.
- Draw a second circle, also centered at the origin, but with radius 0.04.
- Select both circles. Either you can draw a rubber band box around both them, or you can use **Select All** from the **Edit** menu, or simply press **Ctrl-a** to quickly select all objects.
- Press the **Difference** button on the draw toolbar.

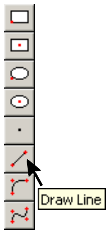



FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 1
CO1
(0, 0, 0)
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

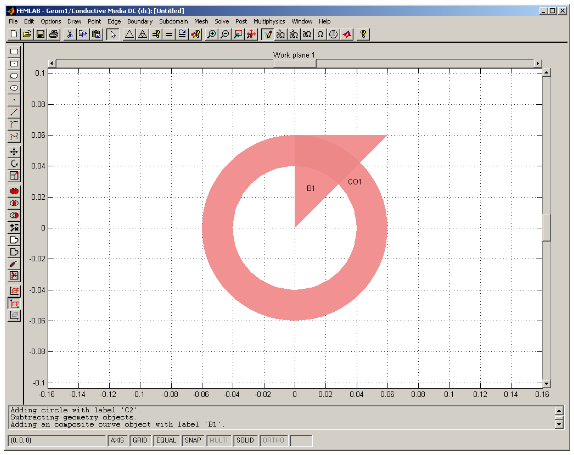
```
Adding circle with label 'C1'.
Adding circle with label 'C2'.
Subtracting geometry objects.
```

The next step is to cut one-eighth out of the ring created so far. This is done most easily covering the desired part with a polygon and taking the intersection.

- On the draw toolbar, press the **Draw Line** button.

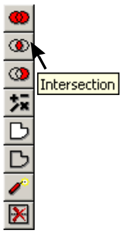


- Draw a triangle using the left mouse button: click in turn at (0,0), (0.06,0.06), and(0,0.06). Press the right mouse button to close and solidify the curve.

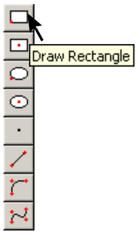

FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 1
B1
CO1
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

```
Adding circle with label 'C2'.
Subtracting geometry objects.
Adding an composite curve object with label 'B1'.
```

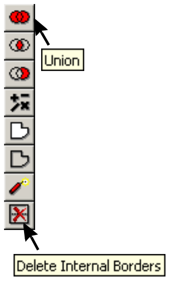
- Select both the circle and the triangle, using one of the methods above or by clicking them in turn, keeping the **Ctrl** key down.
- Press the **Intersection** button on the draw toolbar.



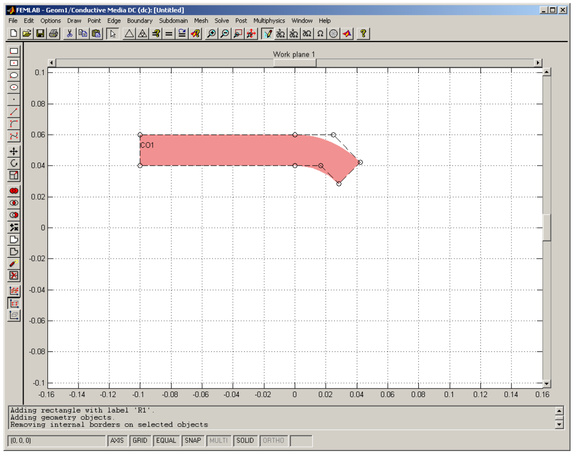
After this, a rectangle is added and joined to the left side of the ring section. When unioning, intersecting and subtracting geometrical objects, the resulting composite object still contains all original boundaries. These, so called, internal borders define a number of subdomains, which can have different physics. In this case, you want the S-shaped part to contain only two subdomains—the part you're drawing, and its mirror reflection. Therefore, you have to remove the inner border between the rectangle and the ring section before continuing.



- Draw a rectangle with corners at (-0.1, 0.04) and (0, 0.06). Select the best drawing tool for the job by pressing the **Draw Rectangle** button. Point the mouse at (-0.1,0.04), press the left mouse button, drag the mouse to (0,0.06) and release the button.

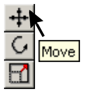


- Once again, select all objects using one of the suggested techniques. Press the **Union** button on the draw toolbar.
- The composite object still contains a border between the rectangle and the ring section. Make sure the object is selected, and remove the border by pressing the **Delete Internal Borders** toolbar button.

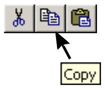

FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 1
CO1
(0, 0, 0)
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

```
Adding rectangle with label 'R1'.
Adding geometry objects.
Removing internal borders on selected objects
```

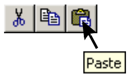
One half of the S-shaped profile is done. The next step is to move it to its final position and make a copy, which is mirrored to produce the desired shape.



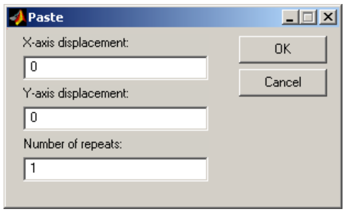
- Select the single composite object still remaining and press the **Move** toolbar button. A small dialog box opens up.
- Enter `-0.05/sqrt(2)` both in the **X-displacement** and **Y-displacement** fields. Press **OK** to move the symmetry line of the final shape to the origin.



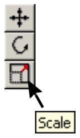
- Copy the composite solid object by pressing the **Copy** button in the main toolbar, choosing **Copy** from the **Edit** menu, or pressing **Ctrl-c**.



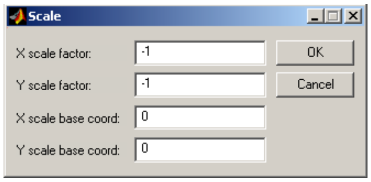
- Press the **Paste** button, or use the short-cut **Ctrl-v**, to open the **Paste** dialog box. Enter 0 in the fields for both the **X-axis displacement** and the **Y-axis displacement** and press **OK**.



X-axis displacement: 0
Y-axis displacement: 0
Number of repeats: 1

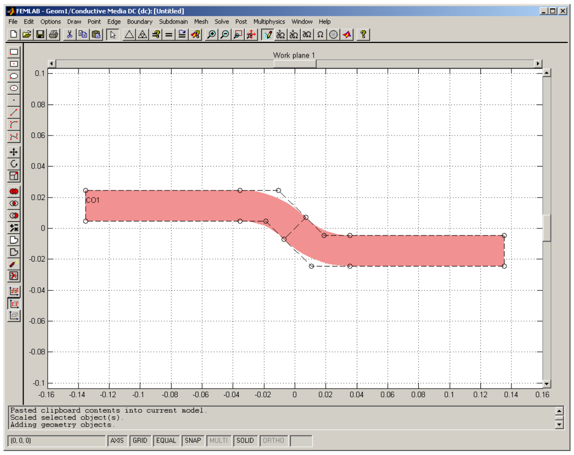


- With the copy still selected, press the **Scale** button on the draw toolbar. Enter -1 as both **X scale factor** and **Y scale factor**. Press **OK**. This mirrors the copy in the line $y = x$.



X scale factor: -1
Y scale factor: -1
X scale base coord: 0
Y scale base coord: 0

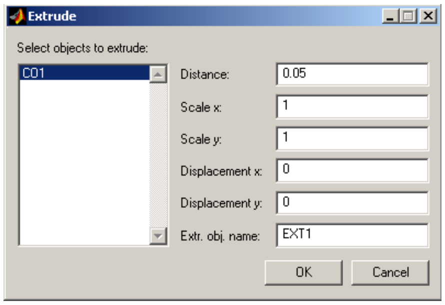
- Select both parts of the profile and press the **Union** toolbar button.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 1
CO1
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

```
Pasted clipboard contents into current model.
Scaled selected object(s).
Adding geometry objects.
```

The 2D profile should now be extruded to produce a 3D solid object. FEMLAB can create 3D objects from 2D profiles by extruding them along a line or revolving them around an axis. As will be seen later on, it's also possible to embed 0D, 1D and 2D objects directly into the 3D space.

- Choose **Extrude...** on the **Draw** menu. Make sure the single object CO1 is selected in the list at the left of the dialog box that opens up.


Extrude

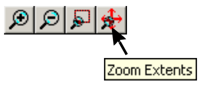
Select objects to extrude:

CO1

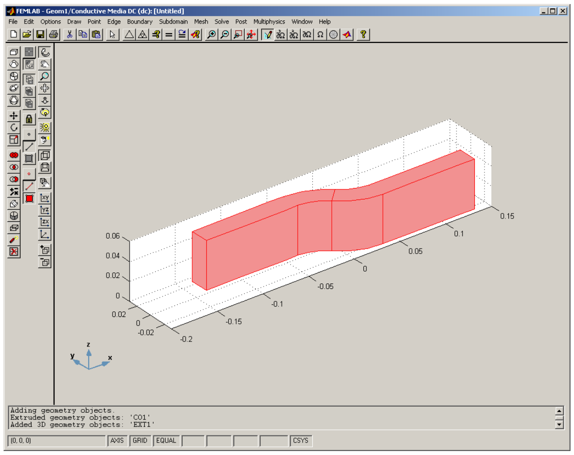
Distance:	0.05
Scale x:	1
Scale y:	1
Displacement x:	0
Displacement y:	0
Extr. obj. name:	EXT1

OK Cancel

- Enter `0.05` in the **Distance** field and press **OK**. The user interface switches to the 3D view.



- Press the **Zoom Extents** button located on the main toolbar just below the menu bar at the top of the screen. When you press it, FEMLAB automatically adjusts the axes' scales so the graphical interface displays the model geometry for optimal viewing and manipulation.

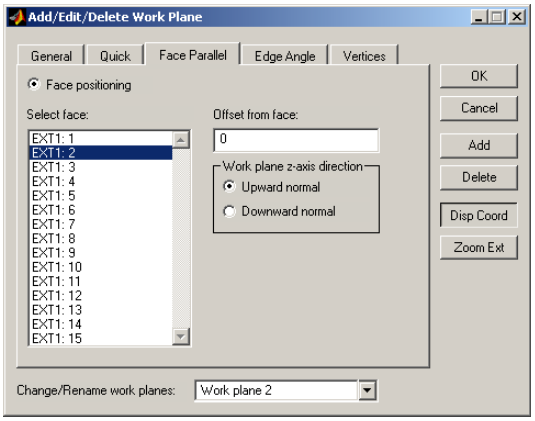

FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
0.06
0.04
0.02
0
0.02
0
-0.02
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
y
z
x
(0, 0, 0)
AXIS GRID EQUAL CSYS

```
Adding geometry objects.
Extruded geometry objects: 'CO1'
Added 3D geometry objects: 'EXT1'
```

The two holes for the fastening screws can be drilled using the cylinder primitive and 3D Boolean operations. It is easier, though, to position the holes and the surrounding contact areas using work planes parallel to the surfaces in which the contact areas should be embedded.

New work planes are created using the **Add/Edit/Delete Work Plane** dialog box, which can be accessed from the **Draw** menu. This dialog box presents a number of alternative ways for positioning new work planes, or moving the existing ones around. See further "Boundary Modeling" on page 1-180. The only one you'll need right now is the utility that positions a work plane parallel to a face in the selected geometry object.

- Select the S-shaped object and then choose **Add/Edit/Delete Work Plane** from the **Draw** menu.
- Press the **Add** button to create a new work plane called `Work plane 2`, and then go to the **Face Parallel** page.
- Click the **Face positioning** radio button to enable the face positioning mode. Select face `EXT1:2` from the list on the left side of the page. You can see the face highlighted in the 3D view.



[✓ Face positioning]

Select face:

EXT1: 1
EXT1: 2
EXT1: 3
EXT1: 4
EXT1: 5
EXT1: 6
EXT1: 7
EXT1: 8
EXT1: 9
EXT1: 10
EXT1: 11
EXT1: 12
EXT1: 13
EXT1: 14
EXT1: 15

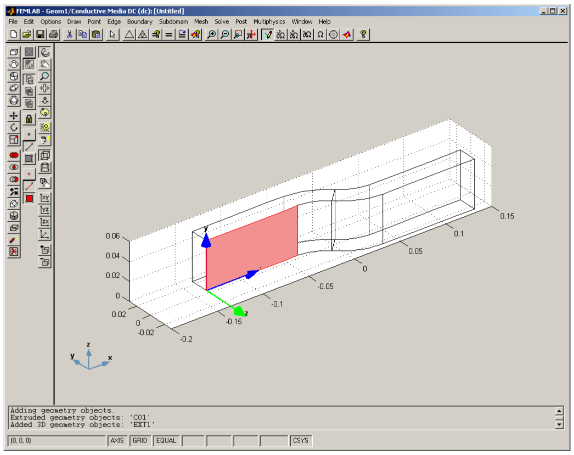
Offset from face: 0

Work plane z-axis direction: [✓ Upward normal] Downward normal

Change/Rename work planes: Work plane 2

- To check how the local coordinate system of the new work plane is embedded in 3D space, press the **Disp Coord** button. A small triple of axes appear in the 3D view. The blue *x*- and *y*-axes represent the local system of the created work plane, while the green *z*-axis shows the direction of positive extrusion from the plane.

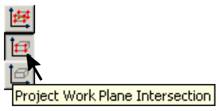
Apparently, you'll have to extrude a circle in the *negative* direction, to drill a hole in the solid object.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
0.06
0.04
0.02
0
0.02
0
-0.02
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

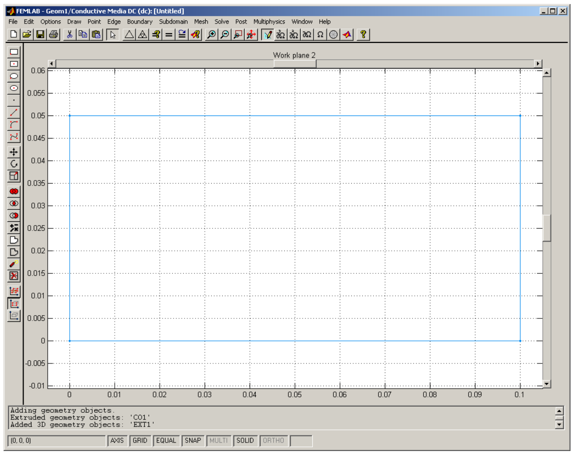
```
Adding geometry objects.
Extruded geometry objects: 'CO1'
Added 3D geometry objects: 'EXT1'
```

- Press **OK** to close the dialog box and enter straight into the new work plane.

While modeling in the work plane, you can choose how much you want to see of the full 3D geometry. The options are to project all 3D geometries, meaning that all 3D edges are projected as blue lines onto the work plane; just show the edges actually lying in the work plane; or show nothing at all. The second option is default.



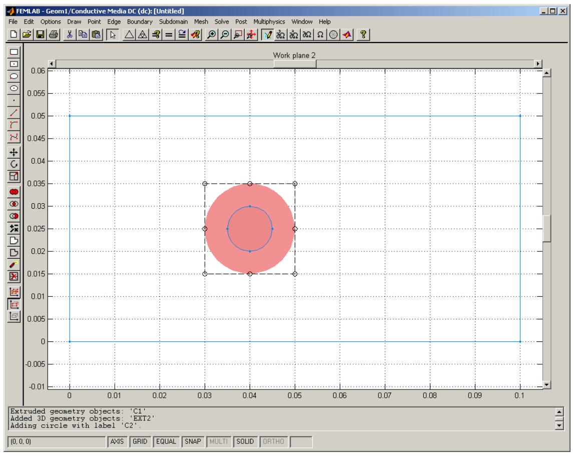
- Check that the **Project Work Plane Intersection** draw toolbar button is selected. This means that you can see only the contours of the face, parallel to which the work plane is positioned. If you press the **Zoom Extents** button now, the view adapts to the blue projected lines, and not to the objects you have drawn in the work plane. Note also that the origin of the local system of coordinates is the lower left corner of the projected face.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 2
0.06
0.055
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
-0.005
-0.01
0 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1
(0, 0, 0)
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

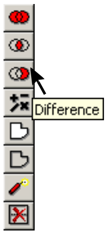
```
Adding geometry objects.
Extruded geometry objects: 'CO1'
Added 3D geometry objects: 'EXT1'
```

- Again, use **Axes/Grid Settings**, accessible from the **Options** menu, to change the density of the grid. On the **Grid** page of the dialog box, deselect **Auto**, enter `0.01` as **X spacing** and `0.005` as **Y spacing**, and press **OK**.
- Draw a circle, centered at (0.04,0.025) and with radius 0.005.
- Create a solid 3D cylinder by selecting the circle, choosing **Extrude...** from the **Draw** menu, entering a **Distance** of `-0.02` and pressing **OK**.
- You're now in 3D view and can see the new cylinder inside the S-shaped bar. Choose **Work plane 2** from the **Draw** menu to return to the work plane, from which you also can embed the necessary contact area.

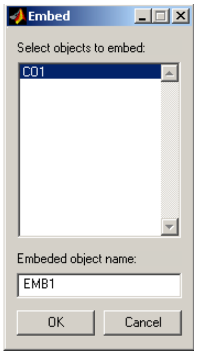
- Everything you draw remains in the work plane until you delete it. Therefore you can re-use the circle employed in creating the cylinder. Draw a second circle using the same center point, but with radius 0.01.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 2
(0, 0, 0)
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

```
Extruded geometry objects: 'C1'
Added 3D geometry objects: 'EXT2'
Adding circle with label 'C2'.
```



- Select both circles and press the **Difference** draw toolbar button.
- Choose **Embed...** from the **Draw** menu, make sure the composite object CO1 is selected in the list, and press **OK**.



Embed

Select objects to embed:

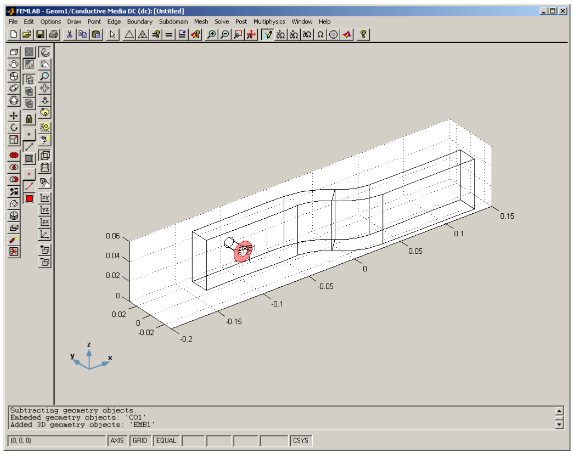
CO1

Embeded object name:

EMB1

OK Cancel

The annular contact area is now embedded in the 3D view.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
EMB1
0.06
0.04
0.02
0
0.02
0
-0.02
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
z
y
x
(0, 0, 0)
AXIS GRID EQUAL
CSYS

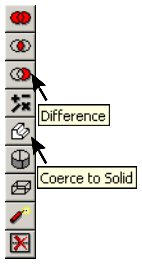
```
Subtracting geometry objects.
Embeded geometry objects: 'CO1'
Added 3D geometry objects: 'EMB1'
```

Now, you have to repeat the same steps for the second hole and contact area.

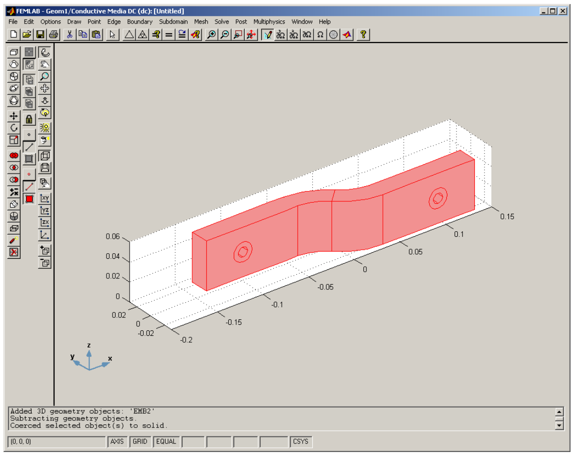
- Create a new work plane as before, but this time parallel to face `EXT1:5`. Check the orientation of the local system of coordinates using the **Disp Coord** button.
- Adjust the grid settings and then draw a circle with radius 0.005, centered at (0.06,0.025).
- Extrude the circle 0.02 units in the negative direction to create a cylinder.

- Return to Work plane 3 and add a second circle with radius 0.01. Select both circles and press the **Difference** button.
- Embed the annular contact area into the 3D view using **Embed...** from the **Draw** menu.

The final step in the geometry modeling is to actually drill the holes and glue the contact areas to the bar.

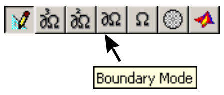


- Select all objects and press the **Difference** button on the draw toolbar. Note that Boolean operations on solid objects do not affect embedded faces.
- Again select all objects, but now press the **Coerce to Solid** draw toolbar button. This incorporates the contact faces into the bus bar.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
0.06
0.04
0.02
0
0.02
0
-0.02
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Added 3D geometry objects: 'EMB2'
Subtracting geometry objects.
Coerced selected object(s) to solid.
```

Boundary Mode



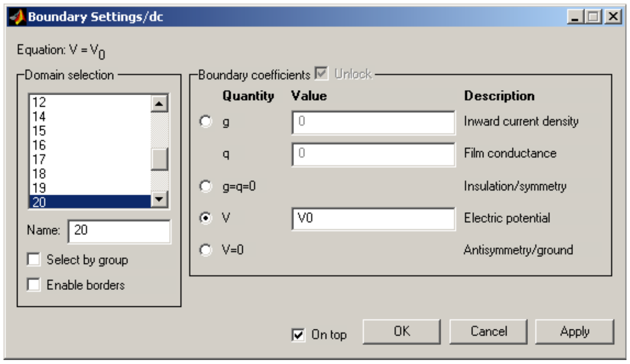
Now that you've created the model's physical geometry, it's time to set its boundary conditions. Everything that has to do with boundary conditions is handled in *boundary mode*, using the options under **Boundary** in the main menu bar. To enter boundary mode, press the **Boundary Mode** button in the main toolbar, or choose **Boundary Mode** from the **Boundary** menu.

Boundary conditions are set in the **Boundary Settings** dialog box, which can be opened from the **Boundary** menu, or by double-clicking anywhere in the model window outside the model geometry. The **Boundary Settings** dialog box is customized to fit the active application mode, because different modes support various types of boundary conditions. The short-hand name of the active mode—in this case `dc`—can be seen after the slash (/) in the title bar of the dialog box. This is important when using different modes simultaneously in a multiphysics model.

When setting boundary conditions, you can select boundaries either by clicking or drawing a rubber band box in the geometry; or by selecting from the **Boundary selection** list in the **Boundary Settings** dialog box. To find out more about the flexible selection tools in FEMLAB, see "Object Selection Methods in 3D" on page 1-95.

The bus bar is electrically insulated everywhere, except on the contact plates, of which one is grounded, and the other held at a constant potential of 50 mV.

- Choose **Boundary Settings** from the **Boundary** menu to open the **Boundary Settings** dialog box.
- First set an insulation condition everywhere: select all boundaries and click the **Insulation/symmetry** radio button. The conditions on the contact plates will be changed in the following steps.
- Select the first contact plate—boundary number 6—and click the **Antisymmetry/ ground** radio button. In the edit fields you can enter real or complex numbers or even MATLAB expressions.
- Select the second contact plate, which is boundary number 20, click the **Electric potential** radio button and enter `V0` in the edit field.



Boundary Settings/dc

Equation: $V = V_0$

Domain selection: 12, 14, 15, 16, 17, 18, 19, [✓ 20]

Name: 20

☐ Select by group

☐ Enable borders

Boundary coefficients ☑ Unlock

	Quantity	Value	Description
○	g	0	Inward current density
	q	0	Film conductance
○	g=q=0		Insulation/symmetry
●	V	V0	Electric potential
○	V=0		Antisymmetry/ground

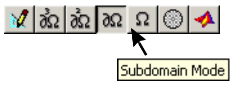
☑ On top

- Close the **Boundary Settings** dialog box, pressing the **OK** button.

Subdomain Mode

Subdomains, in FEMLAB, are the different cells that the geometry is divided into. In the *subdomain mode*, you set material properties (PDE coefficients). It is often necessary to set different parameters for different subdomains.

In the parameters, you can use MATLAB expressions that contain application mode dependent variables, such as the electric potential V and its derivatives, making a model nonlinear. In a multiphysics model, the dependent variables from all included application modes are available to create couplings between the modes. Further, you can always use the space and time variables, for example, x, y, z, and t.



You enter subdomain mode either by pressing the **Subdomain Mode** button in the main toolbar, or by choosing **Subdomain Mode** from the **Subdomain** menu. Also, opening the **Subdomain Settings** dialog box, by choosing **Subdomain Settings...** from the **Subdomain** menu, automatically transfers the user interface to subdomain mode.

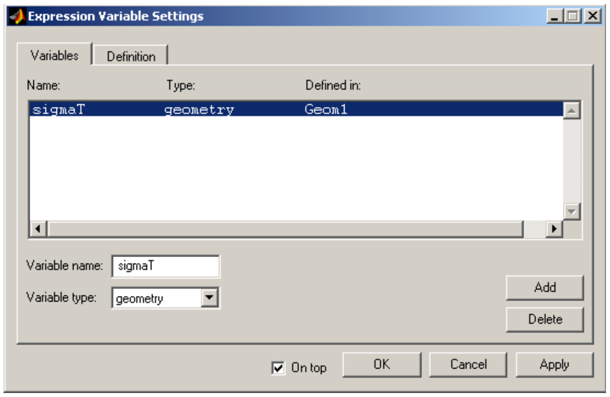
The only necessary parameters for the Conductive media DC application mode are the conductivity, σ, and the distributed current source, Q. In order to prepare for the multiphysics coupling introduced below, the conductivity is written as a nonlinear expression of temperature, T, reference temperature, T_0, resistivity at the reference temperature, ρ_0, and temperature coefficient, α, all of which have been entered as constants into the model:

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

As the expression for σ will be needed more than once in the future, this is the right time to introduce an expression variable.

- Choose **Add/Edit Expressions** from the **Options** menu. This opens the **Expression Variable Settings** dialog box. As you can see, there are no expression variables defined yet.

- Enter the name `sigmaT` in the **Variable name** field and press **Add**. This creates a new, as yet undefined, expression variable named *sigmaT*.

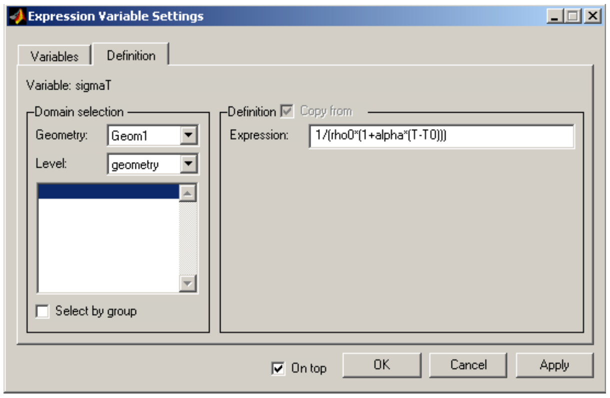


sigmaT geometry Geom1

Variable name: sigmaT

Variable type: geometry

- Go to the **Definition** page. Here you can give the expression variable different meanings on different parts of the geometry or, as is more apropriate in this case, the same meaning everywhere. Just enter `1/(rho0*(1+alpha*(T-T0)))` in the Expression field, and press OK.



Variable: sigmaT

Geometry: Geom1

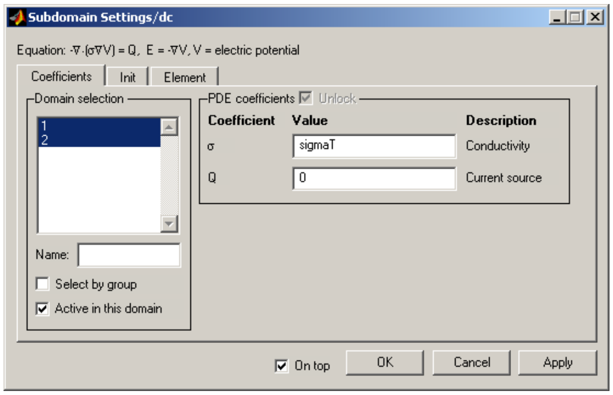
Level: geometry

Expression: 1/(rho0*(1+alpha*(T-T0)))

Using the expression for σ, setting the actual material parameters is simple.

- Open the **Subdomain Settings** dialog box, by choosing **Subdomain Settings...** from the **Subdomain** menu, or by double-clicking anywhere in the axes outside the geometry.
- Select both subdomains. You can select subdomains both in the list on the left side of the **Subdomain Settings** dialog box, and directly in the user interface, using almost the same selection techniques as for boundaries.

- Enter `sigmaT` in the **Conductivity** field, and set the **Current** source to `0`.


Subdomain Settings/dc

Equation: -∇·(σ∇V) = Q, E = -∇V, V = electric potential

Coefficient	Value	Description
σ	sigmaT	Conductivity
Q	0	Current source

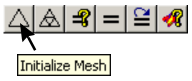
- Press **OK** to apply the subdomain settings and close the dialog box.

In subdomain mode you can also enter initial values. The initial values are used as an initial guess by the nonlinear and iterative solvers, and as initial conditions when solving a time-dependent problem. As this particular model is linear and stationary, the initial data has no effect and can be left at the default value, which is 0.

Mesh Mode

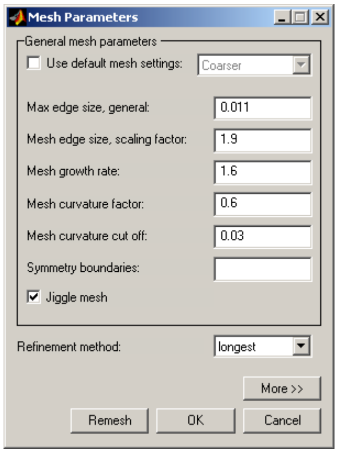
Because FEMLAB is based on the finite element method (FEM), it needs a subdivision of the geometry known as a mesh. To create this representation you work in *mesh mode*. For simple models, chiefly in 2D, you typically just click one of the mesh buttons on the main toolbar; alternatively, you can ignore mesh mode and solve the model directly.

For more complex models, more often than not in 3D, you're better off trying to create the most appropriate mesh for your specific needs. The default mesh is sometimes, as in this case, too dense for a first solution. To change the mesh parameters, you use the **Mesh Parameters** dialog box.



- Go to mesh mode by pressing the **Mesh Mode** button in the main toolbar, pressing the **Initialize Mesh** button, or choosing **Mesh Mode** from the **Mesh** menu. This creates a default mesh with roughly three thousand node points and twelve thousand elements, as you can see in the message log at the bottom of the FEMLAB window. Note that the automatic settings have created a mesh that is rather dense close to curved surfaces, but quite coarse far from the smaller details in the geometry.

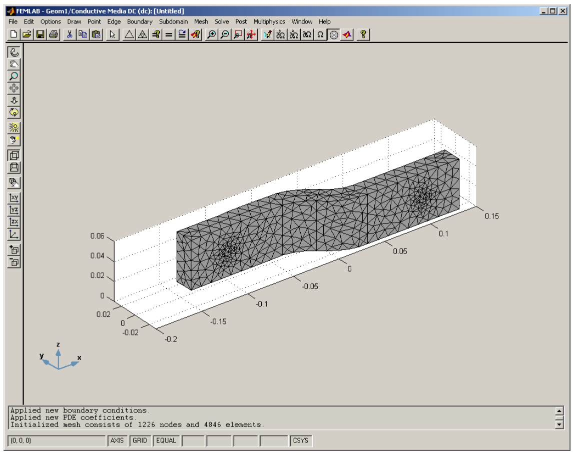
- If you're using second order elements (default) you should create a coarser mesh to save memory and time. Open the **Mesh Parameters** dialog box by choosing **Parameters...** from the **Mesh** menu.
- Make sure the **Use default mesh settings** check box is checked and select **Coarser** from the drop-down list. This will create a much coarser mesh, but still one where the element size varies greatly. To make the mesh smoother, also add a global limit on the mesh element size: enter `0.011` in the **Max edge size, general** field.


Mesh Parameters

General mesh parameters

☐ Use default mesh settings:	Coarser
Max edge size, general:	0.011
Mesh edge size, scaling factor:	1.9
Mesh growth rate:	1.6
Mesh curvature factor:	0.6
Mesh curvature cut off:	0.03
Symmetry boundaries:	
☑ Jiggle mesh	
Refinement method:	longest

- Press **Remesh** and then **OK**. The new mesh consists of slightly more than one thousand nodes and almost five thousand elements.

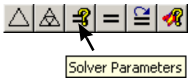

FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]

```
Applied new boundary conditions.
Applied new PDE coefficients.
Initialized mesh consists of 1226 nodes and 4846 elements.
```

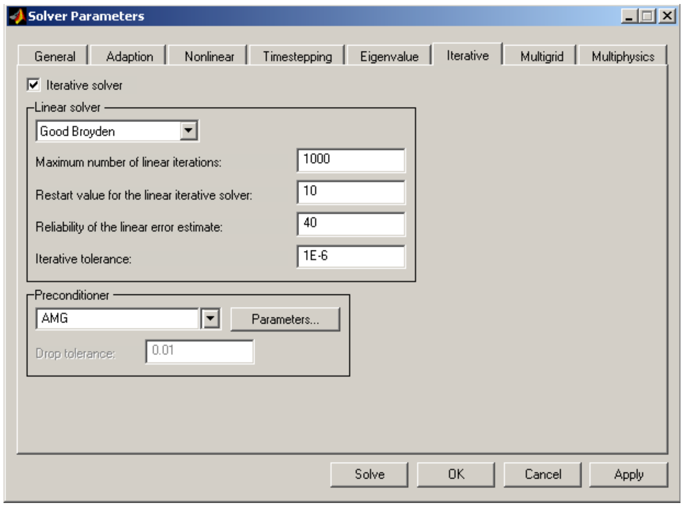
Note that the number of nodes in the mesh is the same as the number of degrees of freedom only when using linear elements. If higher-order elements are used, extra degrees of freedom are introduced.

Solving

Before solving, you can fine tune your solver parameters in the **Solver Parameters** dialog box, a step not strictly necessary for this model. You can adjust solver parameters to achieve higher accuracy or to gain greater control over nonlinear or time-dependent models. For linear 3D models, the iterative solver with *Incomplete LU* preconditioner is default, and does a fair job on this problem. Equations of *Poisson* type are, however, solved much more efficiently using the *Algebraic Multigrid* preconditioner.



- Open the **Solver Parameters** dialog box, by choosing **Parameters...** from the **Solve** menu, or pressing the corresponding toolbar button.
- Go to the **Iterative** page and select AMG from the drop-down list under **Preconditioner**. Press **OK**.



☑ Iterative solver

Linear solver: Good Broyden

Maximum number of linear iterations: 1000

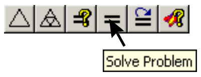
Restart value for the linear iterative solver: 10

Reliability of the linear error estimate: 40

Iterative tolerance: 1E-6

Preconditioner: AMG

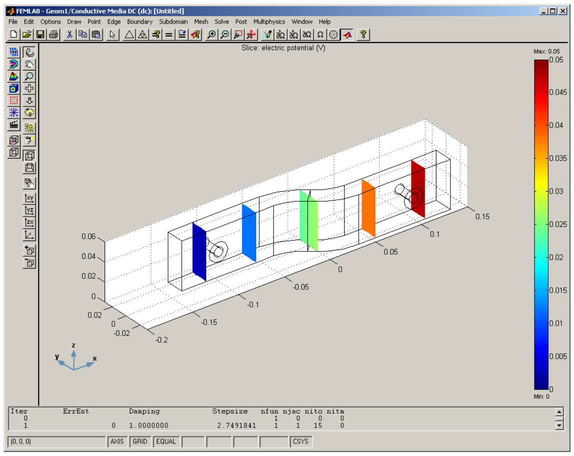
Drop tolerance: 0.01



- To solve the model, press the **Solve Problem** button on the main toolbar. Solving can require anything from 13 seconds up to a couple of minutes depending on your computer.

Post Mode

When the solution is ready, FEMLAB automatically transfers to *post mode* and the solution is plotted. The default 3D plot is displayed on five evenly spaced cross sections of the geometry parallel to the *y*-*z* plane. Which quantity is plotted depends on the application mode. In the Conductive media DC application mode, it is the potential, *V*.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Slice: electric potential (V)
Max: 0.05
Min: 0
AXIS GRID EQUAL CSYS
(0, 0, 0)

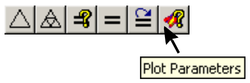
```
Iter     ErrEst          Damping            Stepsize   nfun njac nito nitm
   0                                                      1    0    0    0
   1             0    1.0000000           2.7491841      1    1   15    0
```

It is extremely useful to examine solution results in a graphical form. Therefore FEMLAB's post mode provides plenty of visualization tools, which you access through the **Plot Parameters** dialog box. You can switch quickly between the standard plot types by simply pressing corresponding buttons on the plot toolbar, the default toolbar in this mode. Other powerful options include plotting expressions evaluated on faces, edges, cross sections and lines in separate windows and integrating expressions over subdomains or boundaries.

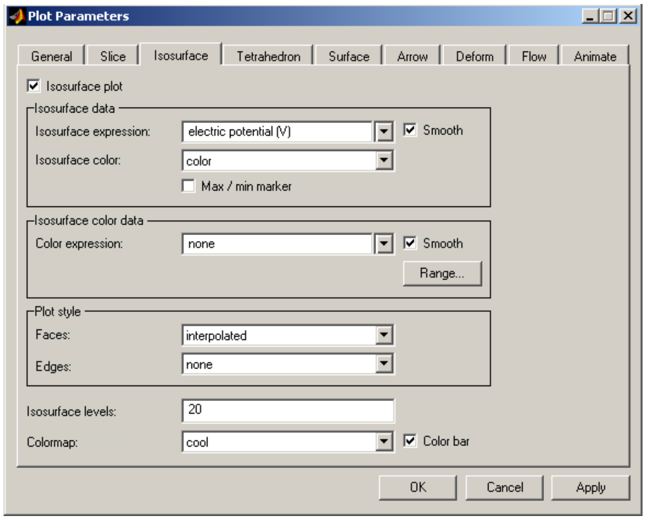
- Try out the buttons on the plot toolbar.

The **Slice plot**, **Isosurface plot** and **Tetrahedron plot** all by default plot the same scalar variable, in this case the potential *V*. The **Arrow** and **Flow** plots are used to visualize vector fields. The default vector field for the Conductive media DC application mode is the current density vector, *J*. The **Surface plot**, finally, plots the normal component of the default vector field. If you try it out, you can see that the current enters at one of the contact plates and exits at the other, the rest is insulated, just as expected.

The easiest way to get a good visualization of the solution is often to start from the default plot given by one of the buttons on the plot toolbar. Then you open the **Plot Parameters** dialog box, and change the parameters until you're satisfied.



- For example, first press the **Isosurface** toolbar button. Open the **Plot Parameters** dialog box by pressing the **Plot Parameters** button in the main toolbar, or by choosing **Plot Parameters** from the **Post** menu.
- Increase the number of isosurfaces: Go to the **Isosurface** page and enter `20` in the **Isosurface levels** field. You can also specify manually, as a MATLAB vector, at what values the isosurfaces are drawn. Try, for example, `linspace(0,0.01,10)`. Always press **Apply** to see the changes.



Plot Parameters

General | Slice | Isosurface | Tetrahedron | Surface | Arrow | Deform | Flow | Animate

☑ Isosurface plot

Isosurface data
Isosurface expression: electric potential (V) ☑ Smooth
Isosurface color: color
☐ Max / min marker

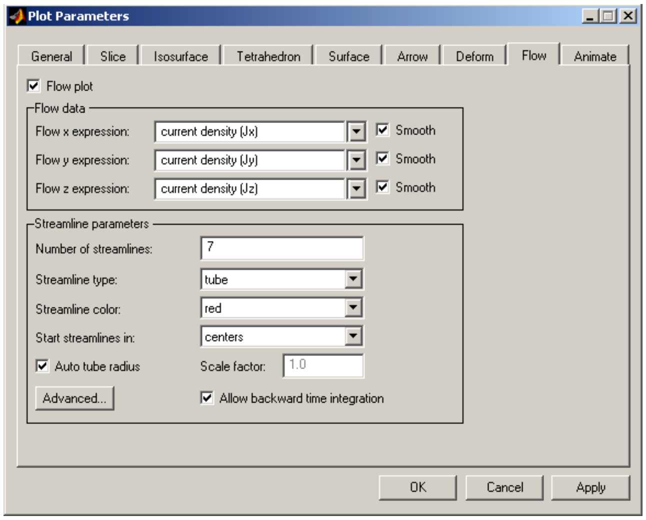
Isosurface color data
Color expression: none ☑ Smooth
Range...

Plot style
Faces: interpolated
Edges: none

Isosurface levels: 20
Colormap: cool ☑ Color bar

OK | Cancel | Apply

- Then add, for example, a flow plot. Go to the **Flow** page and check the **Flow plot** check box on the top left. Then decrease the number of streamlines by entering `7` in the top field in the **Streamline parameters** frame, and make them red, selecting from the **Streamline color** drop-down list.


Plot Parameters

General | Slice | Isosurface | Tetrahedron | Surface | Arrow | Deform | Flow | Animate

☑ Flow plot

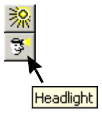
Flow data

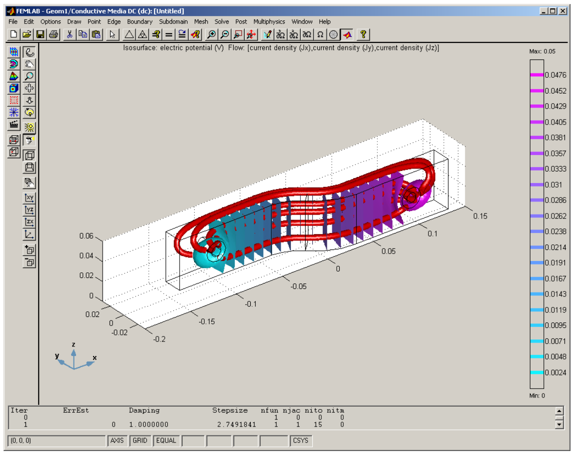
Flow x expression:	current density (Jx)	☑ Smooth
Flow y expression:	current density (Jy)	☑ Smooth
Flow z expression:	current density (Jz)	☑ Smooth

Streamline parameters

Number of streamlines:	7
Streamline type:	tube
Streamline color:	red
Start streamlines in:	centers
☑ Auto tube radius	Scale factor: 1.0
Advanced...	☑ Allow backward time integration

- Press **OK** to close the dialog box and see the final result. Add some light by pressing the **Scenelight** or the **Headlight** button on the draw toolbar.



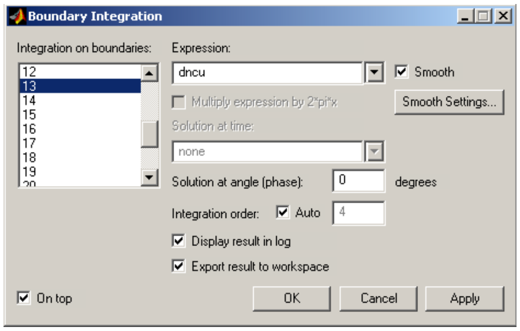

FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Isosurface: electric potential (V) Flow: [current density (Jx),current density (Jy),current density (Jz)]
Max: 0.05
0.0476
0.0452
0.0429
0.0405
0.0381
0.0357
0.0333
0.031
0.0286
0.0262
0.0238
0.0214
0.0191
0.0167
0.0143
0.0119
0.0095
0.0071
0.0048
0.0024
Min: 0
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0.06
0.04
0.02
0
0.02
0
-0.02
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Iter        ErrEst         Damping           Stepsize   nfun njac nito nitm
   0                                                       1    0    0    0
   1                  0    1.0000000          2.7491841    1    1   15    0
```

Integration on Boundaries

One interesting property of a bus bar is its lumped resistance, that is, the total resistance of the bus bar when incorporated in an electric network. The lumped resistance is defined as the applied voltage divided by the associated total current. The total current is found by integrating the normal component of the current density over the internal border between subdomains 1 and 2. Integration on boundaries is controlled from the **Boundary Integration** dialog box, which can be opened from the **Post** menu.

- Choose **Boundary Integration...** from the **Post** Menu. Select boundary 13 in the list on the left. This is the internal border between the two subdomains in the model.



- Enter `dncu` in the **Expression** field, check the **Export result to workspace** box and press **OK**. The reason that you cannot use the predefined current density variable *nJ* is that it represents a jump in the current density rather than the current density itself. To read more about the `dncu` variable, see "Boundary Coupled Equation Variables" on page 3-70 in the *Reference Guide*.

The total current is now present in the MATLAB workspace, as a variable called `ans`. The `ans` variable always contains the result of the last calculation. Therefore, it will be overwritten by the next operation in the workspace. So, you should make a copy under a different name. Then calculate the resistance.

- At the MATLAB prompt, write

```
Itot = ans
```

- Now find the lumped resistance by writing

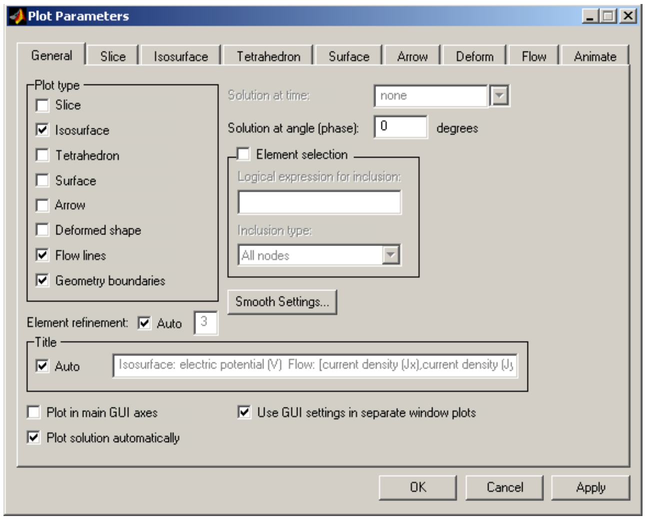
```
R = 0.05/Itot
```

also at the prompt. The result is displayed immediately.

Annotations in Separate Figure Window

If you choose to plot the solution not in the user interface, but in a separate MATLAB figure window, you gain access to MATLAB's tools for setting figure properties. Further, you can use the versatile printing routines available in MATLAB to produce prints and image files on various formats.

- To plot results in a separate MATLAB figure window, again call up the **Plot Parameters** dialog box. Go to the **General** page and deselect the **Plot in main GUI axes** check box. Press **OK** to create the plot.


Plot Parameters

General | Slice | Isosurface | Tetrahedron | Surface | Arrow | Deform | Flow | Animate

Plot type
- [ ] Slice
- [x] Isosurface
- [ ] Tetrahedron
- [ ] Surface
- [ ] Arrow
- [ ] Deformed shape
- [x] Flow lines
- [x] Geometry boundaries

Solution at time: none
Solution at angle (phase): 0 degrees

- [ ] Element selection

Logical expression for inclusion:
Inclusion type: All nodes

Smooth Settings...

Element refinement: [x] Auto 3

Title
- [x] Auto Isosurface: electric potential (V) Flow: [current density (Jx),current density (J

- [ ] Plot in main GUI axes
- [x] Use GUI settings in separate window plots
- [x] Plot solution automatically

OK | Cancel | Apply

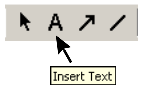
- To remove the axes from the figure, you can enter

  ```
  axis off
  ```

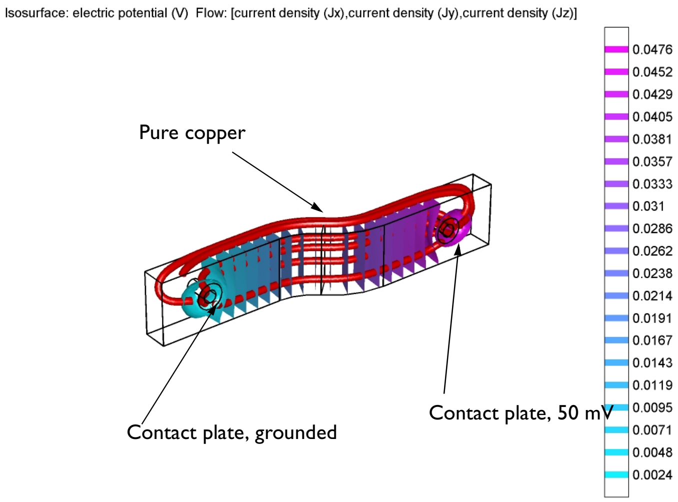
  at the MATLAB prompt. If you use MATLAB 5.3 or later, you can write

  ```
  cameramenu
  ```

  to get access to a number of tools letting you move the camera around, change the lighting and other rendering options.



- You also have access to all of MATLAB's annotation tools. For instance, to add some descriptive text inside the figure window, press the **Add text** toolbar button, click in the window and start writing.
- You can also add objects such as lines and arrows. MATLAB's documentation provides details on how to create annotations in figure windows.



Saving

- To save the model, including all settings in the graphical user interface, go to the **File** menu on the menu bar at the top, then select **Save As** and **Model MAT-file...**.

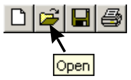
It is also possible to save the model as a MATLAB script file for parameterized models and optimal design by selecting **Save As** and **Model M-file...**. A description of the Model M-file concept appears in this book in the section "Model M-files" on page 1-51.

Thermo-Electric Heating in a Bus Bar with Multiphysics

In the first step of modeling the thermo-electric heating in a bus bar, the stationary DC current distribution at constant temperature was found. See "Thermo-Electric Heating in a Bus Bar" on page 1-109 above for instructions on how to create and save the first part of the model. Here you'll add an application mode for *heat transfer* and couple the equations. You also need an extra point in the geometry, where you can define the total current, using a coupling variable. Then a time-dependent simulation will be run, and the total current and the lumped resistance of the bus bar as functions of time can be plotted.

Loading Previously Saved Model

If you've previously saved the results from the first part of the model as a Model MAT-file (.mat) or Model M-file (.m), you should now load it into the user interface. When you, as in this case, want to continue modeling in the user interface, you should save your model as a binary Model MAT-file. These can be loaded complete with solution and all, while the text-based Model M-files have to recreate the final state by repeating all steps in the solution. The Model M-files, however, have other advantages; see "Model M-files" on page 1-51.



- To load the previously saved model, choose **Open** from the **File** menu, and then **Model MAT-file** or **Model M-file**, as appropriate, in the submenu. You can also press the **Open** button on the main toolbar to load a Model MAT-file.
- Locate your model file and press **OK**.

Options and Settings

The first thing you have to do is remove the constant *T,* otherwise it will conflict with the solution *T* in the heat transfer equation.

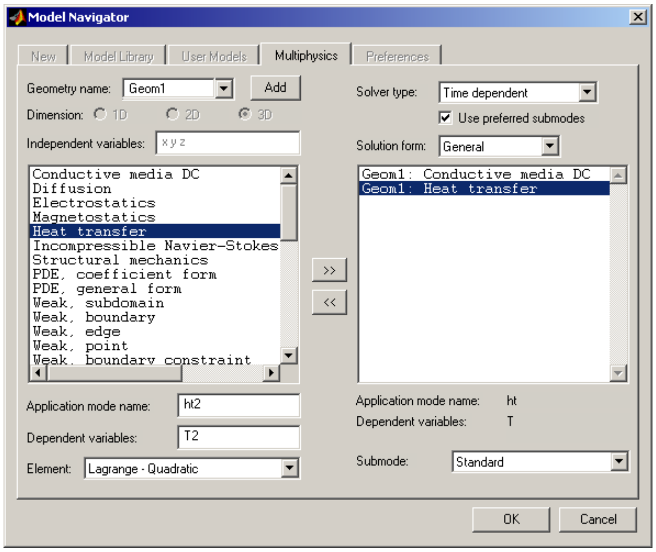
- Choose **Add/Edit Constants...** from the **Options** menu. Select the constant `T` and press the **Delete** button. Press **OK** to close the dialog box.

Multiphysics

Now add the Heat transfer application mode. Additional modes are added using the **Model Navigator**, which for this purpose can be accessed from the **Multiphysics** menu. You also should change the solution form from *Coefficient form* to *General form*. This is necessary because in Coefficient form, the nonlinear couplings between the electric potential *V* and the temperature *T* aren't considered when assembling the *Jacobian* (essentially the same as the stiffness matrix for a linear stationary problem). The correct Jacobian is essential in order to obtain good convergence from the nonlinear and time-dependent solvers. See "The Nonlinear Solver" on page 3-108 and "The Time-Dependent Solver" on page 3-112, both in the *Reference Guide*.

- Choose **Add/Edit Modes...** form the **Multiphysics** menu. The **Model Navigator**—the same dialog box that is used for creating new models—opens up on the **Multiphysics** page. The other pages are inactive; they are only accessible when creating a new model, or loading an old one.

- Select **Heat transfer** in the application mode list on the left. Note that the default name of the dependent variable is `T` and the Element type is **Lagrange - Quadratic**. Accept this and add the Heat transfer mode to the model by pressing the right arrow (**>>**) in the middle of the dialog box.
- Change **Solver type** to `Time dependent` and **Solution form** to `General`, by selecting from the corresponding drop-down lists.



Geometry name: Geom1
Dimension: 1D 2D [✓ 3D]
Independent variables: x y z

```
Conductive media DC
Diffusion
Electrostatics
Magnetostatics
Heat transfer
Incompressible Navier-Stokes
Structural mechanics
PDE, coefficient form
PDE, general form
Weak, subdomain
Weak, boundary
Weak, edge
Weak, point
Weak, boundary constraint
```

Application mode name: ht2
Dependent variables: T2
Element: Lagrange - Quadratic

Solver type: Time dependent
[✓] Use preferred submodes
Solution form: General

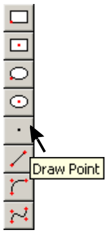
```
Geom1: Conductive media DC
Geom1: Heat transfer
```

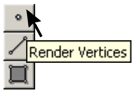
Application mode name: ht
Dependent variables: T
Submode: Standard

- Press **OK** to accept the choices and close the **Model Navigator**. The user interface is now in **Heat Transfer** mode, as can be seen in the title bar of the FEMLAB window.

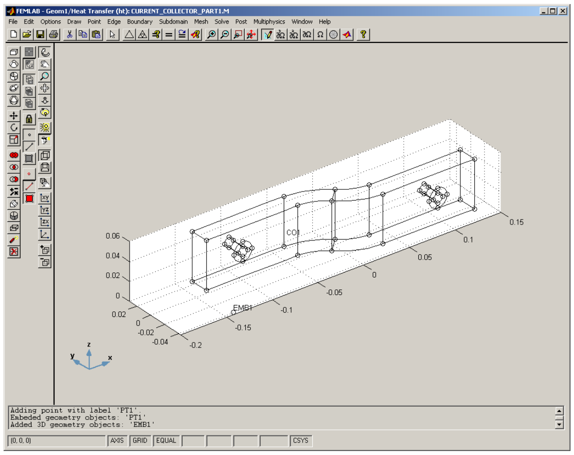
Draw Mode

A coupling variable representing the total current flux from subdomain 2 to subdomain 1 will be introduced below. Coupling variables must have a source—in this case boundary number 13—and a destination. As the total current is a scalar value, it makes sense to have it defined on a point, and also to embed a new point for this purpose.





- Enter **Work plane 1** by choosing from the **Draw** menu. Press the **Draw Point** button on the draw toolbar and draw a single point at (-0.14,-0.04).
- Choose **Embed...** from the **Draw** menu. Make sure only the point PT1 is selected in the list, and press **OK**.
- The new embeded point is invisible in the 3D view until you press the **Render Vertices** button on the select toolbar next to the draw toolbar.


FEMLAB - Geom1/Heat Transfer (ht): CURRENT_COLLECTOR_PART1.M
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
EMB1
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Adding point with label 'PT1'.
Embeded geometry objects: 'PT1'
Added 3D geometry objects: 'EMB1'
```

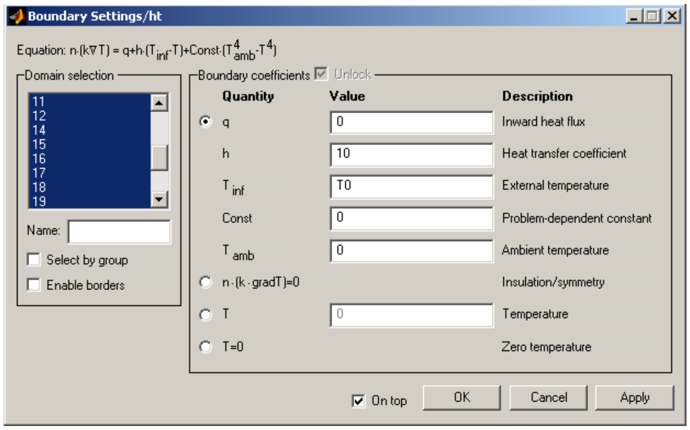
Boundary Mode

The boundary conditions for the Heat transfer mode are somewhat more complicated than for the Conductive media DC mode: One face, the inside of one of the fastening holes, and the contact plates are kept at a constant temperature equal to the temperature of the surrounding air. The free faces have a heat transfer coefficient of 10, while on the inside of the other hole it's only 5. Finally, one of the end surfaces has an Insulation/symmetry condition, representing the fact that the modeled bus bar is mounted end-to-end with another, of similar shape.

The following table contains an overview of the boundary conditions. You'll meet more tables like this one in the rest of the examples in this book, and also throughout the *Model Library*.

BOUNDARY	1-4, 11-19	5-10, 20	21-24	25
Type	Flux	Temperature	Flux	Insulation/symmetry
q	0	-	0	-
h	10	-	5	-
T_{inf}	T0	-	T0	-
Const	0	-	0	-
T_{amb}	0	-	0	-
T	-	T0	-	-

- Choose **Boundary Settings** from the **Boundary** menu. Note that the title bar of the dialog box now reads **Boundary Settings/ht**, to show that you're working with the Heat transfer mode.
- Select boundaries 1-4, 11-19 and 21-24, and press the top radio button for a flux condition. Enter `10` in the **h** field and `T0` in the **Tinf** field.


Boundary Settings/ht

Equation: $n\cdot(k\nabla T) = q+h\cdot(T_{inf}-T)+Const\cdot(T_{amb}^4-T^4)$

Domain selection: 11, 12, 14, 15, 16, 17, 18, 19

Name:

☐ Select by group

☐ Enable borders

Boundary coefficients ☑ Unlock

	Quantity	Value	Description
●	q	0	Inward heat flux
	h	10	Heat transfer coefficient
	T_{inf}	T0	External temperature
	Const	0	Problem-dependent constant
	T_{amb}	0	Ambient temperature
○	n·(k·gradT)=0		Insulation/symmetry
○	T	0	Temperature
○	T=0		Zero temperature

☑ On top

- Deselect all boundaries except 21-24 and change **h** to `5`.
- Select boundaries 5-10 and 20, select the **Temperature** radio button and enter `T0` in the edit field.
- Finally, select boundary 25 and select **Insulation/Symmetry**. Press **OK**.

Subdomain Mode

The application mode specific material parameters for the Heat transfer mode are density, ρ, heat capacity, C, thermal conductivity, k, and volume heat source Q. The heat source in this case is the electrical heating, which depends on the solution from the Conductive media DC problem. The heating power per unit volume from the current density J is:

$$P = \frac{J^2}{\sigma} = J \cdot E = \sigma E^2 = \sigma|\nabla V|^2$$

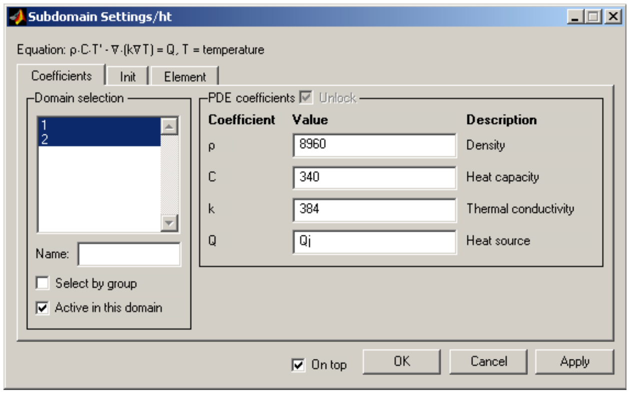
where E is the electric field and V the potential—the dependent variable in the Conductive media DC application mode. The conductivity, σ, is temperature dependent, as before. The heat source is best entered using an expression variable.

- Choose **Add/Edit Expressions...** from the **Options** menu. Enter the new expression name `Qj` and press the **Add** button.
- On the **Definition** page, set the variable to mean `sigmaT*(Vx^2+Vy^2+Vz^2)` on the geometry level. Press **OK**. Note that the new expression variable includes the old *sigmaT* in its definition.

Now enter the material parameters (PDE coefficients) according to the following table, and set the initial conditions:

SUBDOMAIN	1,2
ρ	8960
C	340
k	384
Q	`Qj`

- Choose **Subdomain Settings** from the **Subdomain** menu. Select both subdomains and enter the values for ρ, C, and k, and write `Qj` in the Q field.



Equation: ρ·C·T' - ∇·(k∇T) = Q, T = temperature

Domain selection: 1, 2

Name:

Active in this domain ✓

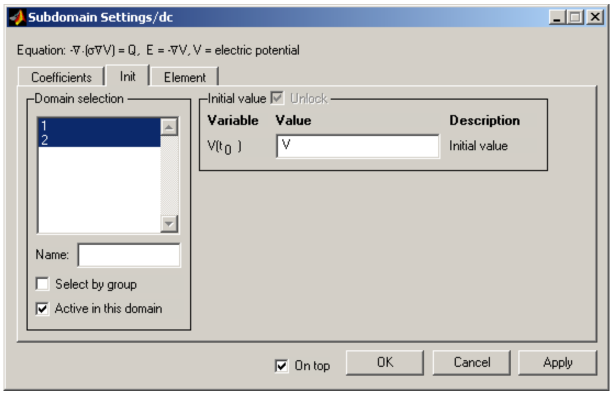
PDE coefficients ✓ Unlock

Coefficient	Value	Description
ρ	8960	Density
C	340	Heat capacity
k	384	Thermal conductivity
Q	Qj	Heat source

- Click the **Init** tab to show the **Init** page. Still with both subdomains selected, enter `T0` in the **T(t_0)** field. This begins the simulation from room temperature.

It is important that your initial conditions are consistent. Now that you have set an initial temperature, you should set an initial condition for the electric potential corresponding to the stationary solution of the Conductive media DC problem at the same temperature. This you can accomplish by setting a previously stored solution as initial condition for the following calculations.

- Choose the **Conductive media DC** application mode from the bottom of the **Multiphysics** menu. This transfers the user interface and all open dialog boxes to the Conductive media DC mode.


Subdomain Settings/dc

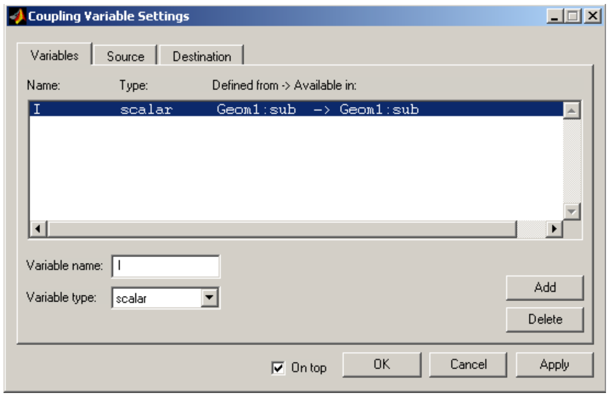
Equation: $-\nabla\cdot(\sigma\nabla V) = Q$, $E = -\nabla V$, V = electric potential

Variable	Value	Description
$V(t_0)$	V	Initial value

- Enter `V` in the $V(t_0)$ field and press **OK** to confirm and close the Subdomain Settings dialog box.

Now add a coupling variable which puts the value of the integral of the current flux over the border between subdomains 1 and 2 into point number 1, the single point embedded above.

- Choose **Add/Edit Coupling Variables...** from the **Options** menu. This opens the **Coupling Variable Settings** dialog box. Enter the variable name `I` of type **Scalar** and press **Add**.

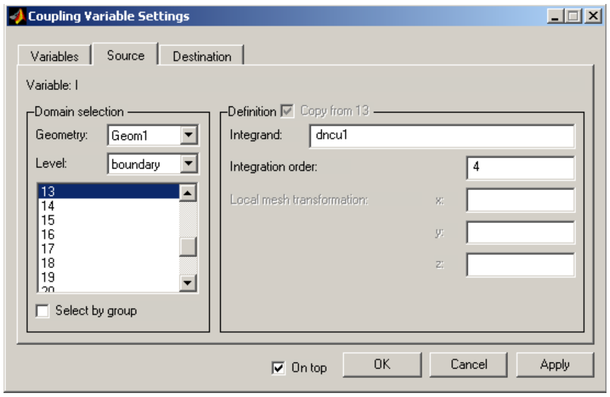

Coupling Variable Settings

Name:	Type:	Defined from -> Available in:
I	scalar	Geom1:sub -> Geom1:sub

Variable name: I

Variable type: scalar

- Go on to the **Source** page. Choose **Level** `boundary` from the drop down list, and then select boundary 13.
- Enter the **Integrand** `unga1` and **Integration order** `4`.


Coupling Variable Settings

Variable: I

Domain selection

Geometry: Geom1

Level: boundary

13
14
15
16
17
18
19
20

☐ Select by group

Definition ☑ Copy from 13

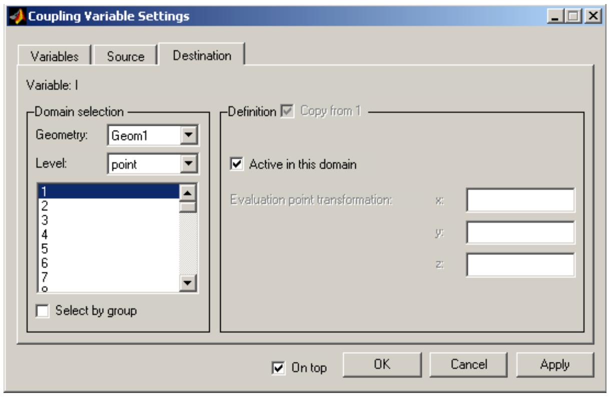
Integrand: dncu1

Integration order: 4

Local mesh transformation: x: y: z:

☑ On top

The variable `nga1` is a so called coupled boundary variable that in general form defines the normal flux on the boundary. On an outer boundary you would use just `nga1`, or even simpler, the postprocessing variable `nJ`—the normal component of the current density vector. But, for internal borders these are averages of the currents as evaluated on the up and down side of the border, which is generally zero; that is `nga1` is the jump in the flux. `unga1`, on the other hand, is the flux out of the upper subdomain, and therefore into the subdomain on the down side. Se further "The FEMLAB PDE Language" on page 1-336 in this book.


Coupling Variable Settings

Variable: I

Domain selection

Geometry: Geom1

Level: point

1
2
3
4
5
6
7
8

☐ Select by group

Definition ☑ Copy from 1

☑ Active in this domain

Evaluation point transformation: x: y: z:

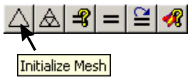
☑ On top

- Proceed to the **Destination** page, where you choose **Level** `point` and select point 1. Check the box **Active in this domain** and press **OK**. You now have a variable *I* defined on point number 1, whose value is always the total current through the bus bar.

Mesh Mode

As you have changed the geometry, the mesh will be re-initialized when you enter mesh mode. The settings in the **Mesh Parameters** dialog box are also lost, so you have to reenter them to create practically the same mesh as before.

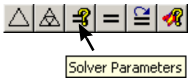
- First select **Use default mesh settings** and `Coarser`. Then enter a **Max edge size, general** of `0.011` and press **OK**.
- Press the **Initialize mesh** button on the main toolbar.



Solving

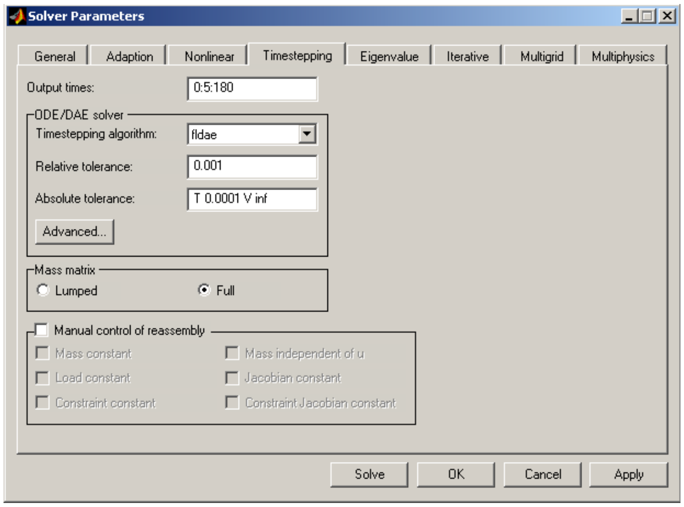
This time you're going to use the time-dependent solver. You always have to specify the time interval for the solution. Normally, it's best to specify exactly at what times the solution should be stored, as the timestepping algorithms calculate the solution at more points in time, than you probably want to keep. That is, the timestepping algorithms use automatic time step control, but you can specify certain times for which the solution will be interpolated and stored.

If the output times are given as a MATLAB vector of more than two elements, they are considered to be times for storing the solution. If, on the other hand, the output times are given as a vector of only two numbers, these are taken as start and end points of an interval where all time steps, as decided by the timestepping algorithm, should be saved.



- Open the **Solver Parameters** dialog box by pressing the **Solver Parameters** button on the main toolbar. Among the **Advanced** options on the **General** page, you find **Direct linear solver**. If you are running MATLAB 6.0 or later, select `SuperLU`; this solver works better than the built-in MATLAB solver for large problems, which may require virtual memory.
- Click the **Timestepping** tab.
- In the **Output times** field, enter `0:5:180`, which means that the solution will be saved every five seconds during three minutes.

- Select the `fldae` **Timestepping algorithm**. The ODE system at hand is a *differential-algebraic equation* (DAE), because it contains no time derivatives of the potential *V*. In other words, the mass matrix is singular.



Output times: 0:5:180

ODE/DAE solver

Timestepping algorithm: fldae

Relative tolerance: 0.001

Absolute tolerance: T 0.0001 V inf

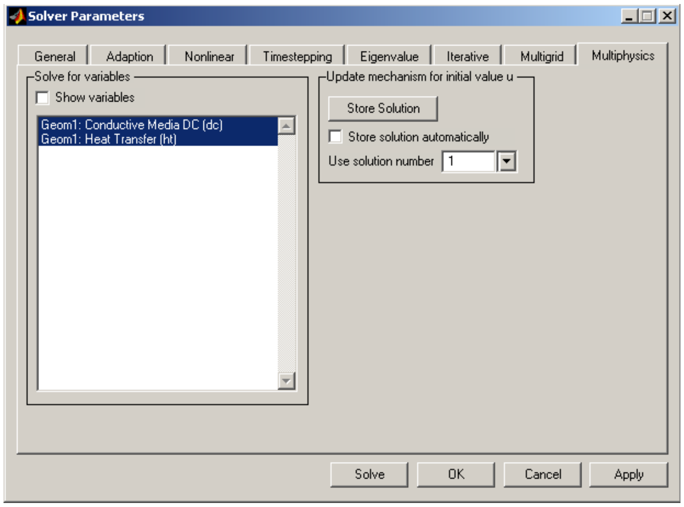
Mass matrix: Lumped [✓ Full]

☐ Manual control of reassembly

☐ Mass constant	☐ Mass independent of u
☐ Load constant	☐ Jacobian constant
☐ Constraint constant	☐ Constraint Jacobian constant

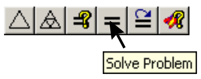
- As the system is a DAE, the accuracy in the potential *V* will be within tolerance, given enough accuracy in the temperature *T*. Therefore set the **Absolute tolerance** to `T 0.0001 V inf`. This means that there is no explicit limit on the accuracy in *V*. Also, change the **Relative tolerance** to `0.001`.
- Go to the **Multiphysics** page and press the **Store Solution** button. This stores the previous solution for the electric potential *V*, so that it can be used as initial value

for the multiphysics problem. Press **OK** to confirm and close the Solver Parameters dialog box.


Solver Parameters

Geom1: Conductive Media DC (dc)
Geom1: Heat Transfer (ht)

Note This problem requires approximately 500 MB of memory. When using the SuperLU linear solver, solving in virtual memory (“swap”) is reasonably efficient. It is also possible to reduce the memory requirements by selecting **Solution form** `coefficient` on the **General** page of the Solver Parameters dialog box (300 MB) or switching to linear elements (150 MB). The former option gives an accurate result, as the tolerances are always met, but forces the timestepping algorithm to take more and shorter steps. The latter alternative gives a much faster, but less accurate, solution.



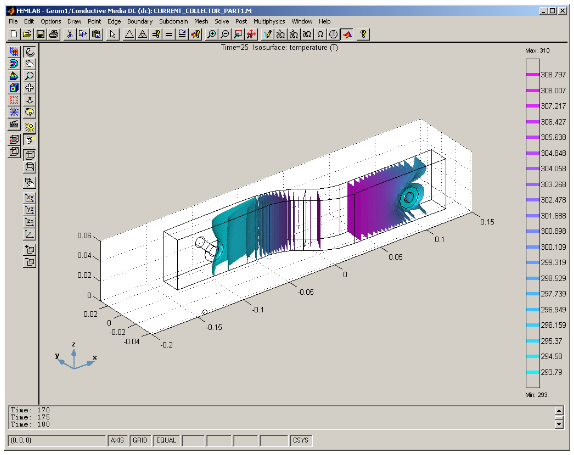
- Press the **Solve Problem** button in the main toolbar to solve the problem. The time required to solve the problem on a 1.5 GHz Pentium 4 is approximately 529 seconds.

Post Mode

The current plot settings in the **Plot Parameters** dialog box are used when the new solution is plotted. Only the last time step is plotted automatically. This time, you’ll rather want to have a look at the temperature distribution in the bar.

- Open the **Plot Parameters** dialog box. Deselect **Flow lines** under **Plot type** on the **General** page.

- Continue to the **Isosurface** page and select `temperature (T)` from the **Isosurface expression** drop-down list. Press **Apply**.
- Return to the **General** page and have a look at the solution at some earlier timesteps. You choose which time to display by selecting from the **Solution at time** drop-down list. Do not forget to press **Apply**.

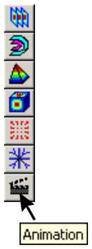


```
Time: 170
Time: 175
Time: 180
```

Animating the Solution

The time evolution of a model can be visualized by displaying the saved solutions as an animation. Surface plots often look particularly good as animations.

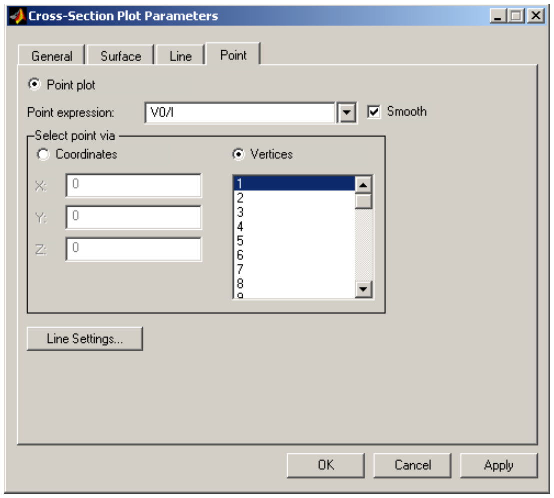
- Open the **Plot Parameters** dialog box, deselect the **Isosurface** plot, select **Surface** instead and change the **Surface Expression** on the **Surface** page to `temperature (T)`. Press **OK**.
- Press the **Animation** button on the plot toolbar. A MATLAB figure window opens, where the animation is first recorded frame by frame and then played five times. If you press the animation button again, the movie is played directly without the recording step.



Plotting the Resistance

To plot the time evolution of the lumped resistance, you can use the cross-section plot facilities for nodes. As the total current already has been extracted out to the free-lying point, all that is needed is to plot the specified voltage 50 mV divided by this value for all time steps.

- Open the **Cross-Section Plot Parameters** dialog box, by choosing **Cross-Section Plot Parameters...** from the **Post** menu. Click **Point** under **Plot type** and select all time steps in the **Solution at time** list.


Cross-Section Plot Parameters

General | Surface | Line | Point

Point plot: selected

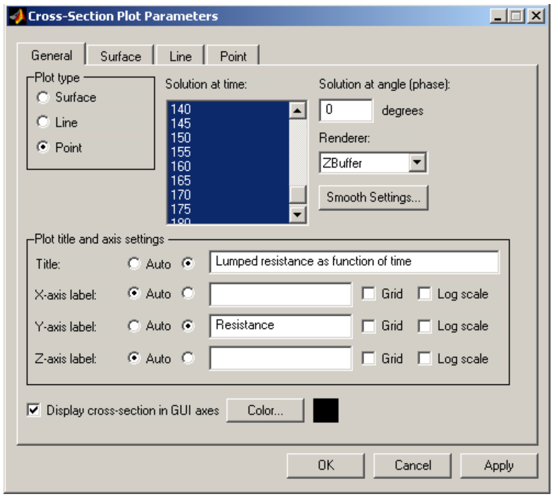
Point expression: V0/I

Smooth: ✓

Select point via: Vertices (Coordinates X: 0, Y: 0, Z: 0)

Vertices: 1 (selected), 2, 3, 4, 5, 6, 7, 8

- On the **Point** page, choose to plot data for vertices rather than for specified coordinates by clicking the **Vertices** radio button under **Select point via**. Then select vertex number 1 from the list. Enter `V0/I` in the **Point Expression** field.


Cross-Section Plot Parameters

General | Surface | Line | Point

Plot type: Point

Solution at time: 140, 145, 150, 155, 160, 165, 170, 175 (selected)

Solution at angle (phase): 0 degrees

Renderer: ZBuffer

Plot title and axis settings

Title: Lumped resistance as function of time

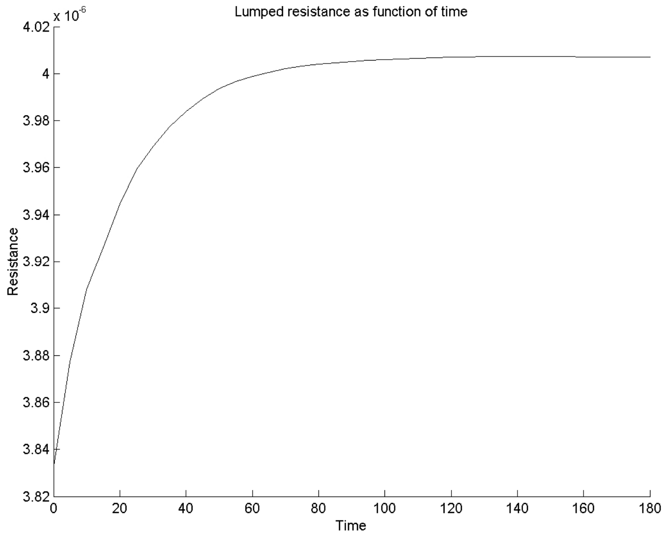
X-axis label: Auto

Y-axis label: Resistance

Z-axis label: Auto

Display cross-section in GUI axes: ✓

Return to the **General** page and set a title for the plot: Activate the **Title** edit field by clicking the radio button just to the left of it. Enter, for example, `Lumped resistance as function of time`. Then change the **Y-axis label** to read `Resistance`, and press **OK**.



Exporting Data to Text File

The plotted data in the cross sectional plot above can also be exported to a text file. The data in the text file can then be loaded into other programs, such as Microsoft Excel. A text file with the plot data is generated by first extracting the values used to generate the plot above, using the corresponding command line function `postcrossplot` at the MATLAB command prompt, followed by issuing the MATLAB command `save -ascii` to save the plot data to a text file.

- Start by exporting the FEM structure to the command line by selecting **Export FEM structure as 'fem'** in the **File** menu or by pressing the **ctrl-f** key.
- Extract the plotted data in the cross sectional plot by issuing the following command at the MATLAB command prompt (for a description of `postcrossplot` type `help postcrossplot` at the MATLAB command prompt).

```
[h,data] = postcrossplot(fem,0,1,'Pointdata','V0/I',...
  'Solnum',1:37);
```

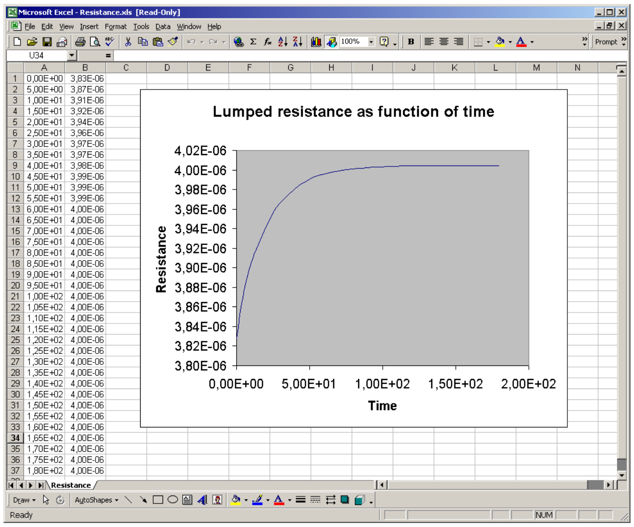
- Create a matrix with the x-data and y-data values from the structure `data`.

```
resistance = [data.XData(:) data.YData(:)];
```

- Save the data values in the matrix `resistance` to a text file called `Resistance.txt`.

```
save Resistance.txt resistance -ascii
```

The data in the text file `Resistance.txt` can now be loaded into Microsoft Excel and be used in creating a diagram.



	A	B
1	0,00E+00	3,83E-06
2	5,00E+00	3,87E-06
3	1,00E+01	3,91E-06
4	1,50E+01	3,92E-06
5	2,00E+01	3,94E-06
6	2,50E+01	3,96E-06
7	3,00E+01	3,97E-06
8	3,50E+01	3,97E-06
9	4,00E+01	3,98E-06
10	4,50E+01	3,99E-06
11	5,00E+01	3,99E-06
12	5,50E+01	3,99E-06
13	6,00E+01	4,00E-06
14	6,50E+01	4,00E-06
15	7,00E+01	4,00E-06
16	7,50E+01	4,00E-06
17	8,00E+01	4,00E-06
18	8,50E+01	4,00E-06
19	9,00E+01	4,00E-06
20	9,50E+01	4,00E-06
21	1,00E+02	4,00E-06
22	1,05E+02	4,00E-06
23	1,10E+02	4,00E-06
24	1,15E+02	4,00E-06
25	1,20E+02	4,00E-06
26	1,25E+02	4,00E-06
27	1,30E+02	4,00E-06
28	1,35E+02	4,00E-06
29	1,40E+02	4,00E-06
30	1,45E+02	4,00E-06
31	1,50E+02	4,00E-06
32	1,55E+02	4,00E-06
33	1,60E+02	4,00E-06
34	1,65E+02	4,00E-06
35	1,70E+02	4,00E-06
36	1,75E+02	4,00E-06
37	1,80E+02	4,00E-06

Controlling Temperature, Exporting to Simulink®

This model simulates a heated metal block with a thermal controller, and demonstrates how to set up communication between FEMLAB and Simulink.

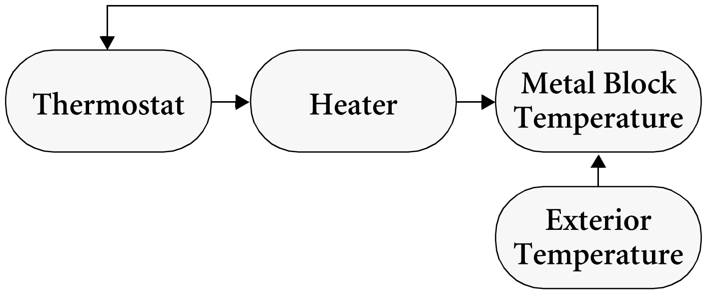
It is also possible to simulate this system without Simulink by using a FEMLAB model with two geometries and non-local coupling variables to implement the interaction between the equations in the two geometries. The model `thermal_controller_ode` follows on page 2-288 in the *Model Library,* showing how to set up this alternative formulation.

Note The second part of this model requires Simulink on your system.

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 a package such as Simulink.
- One of the model's elements, the controller, contains a subsystem that involves heat distribution. You can model that phenomenon with a PDE, the well-known heat equation. It is easy to break this portion of the problem down and describe it in FEMLAB.

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, which works in a simple manner: The thermostat turns the heater on or off when the temperature becomes too low or 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}.

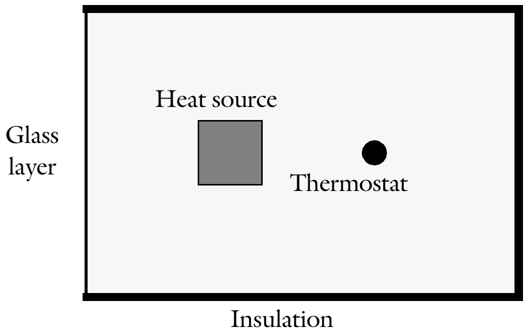
As its output, the model supplies the temperature at the thermostat's location.

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 rapid such that for all practical purposes the heat distribution is constant (in space, not in time), a single state is sufficient. One way to simplify the solution is to replace the PDE with a gain function, thereby reducing the modeling process to computing the effective heat conductance. Otherwise, you must merge FEMLAB into the Simulink environment, a task this model illustrates.

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 $\boldsymbol{n} \cdot (k\nabla T) = 0$. The poorly insulated sides involve the Neumann condition $\boldsymbol{n} \cdot (k\nabla T) = k_g/l_g \cdot (T_{out} - T)$, 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.



For the purposes of this discussion, assign system objects with some specific characteristics as noted below. Since we are only interested in the temperature distribution in the xy-plane, the model is carried out in 2D. For the units to make sense, you have to think of the domain as having a depth (z-direction) of 1 m.

FEMLAB doesn't specify any units, which allows you to choose any self-consistent system. This model, as the rest of the Model library, 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 all involved quantities whose units contain the basic unit of time must be scaled by a factor 60 compared to their standard SI values. This conversion is carried out below for the thermal conductivity k and the heat source Q, before the values are entered into FEMLAB.

Metal block:

- *Dimensions* 30×20 x 100 cm $=$ 0.3×0.2 x 1 m
- *Density* $\rho = 7.82 \cdot 10^3$ kg/m^3

- *Heat capacity* C = 449 J/kg·K
- *Thermal conductivity* k = 82 W/m·K = $4.92 \cdot 10^3$ J/m·min·K

Glass sheet:

- *Thermal conductivity* k_g = 0.9 W/m·K = 54 J/m·min·K
- *Thickness* l_g = 1 mm = 10^{-3} m

Heater:

- *Dimensions* $4 \times 4 \times 100$ cm = $0.04 \times 0.04 \times 1$ m
- *Heat source* Q = 2 kW / V_{heater} = 2e3 / $(1 \cdot 0.04^2)$ W/m^3 = $7.5 \cdot 10^7$ J/m^3·min

MODELING THE TIME-DEPENDENT TEMPERATURE DISTRIBUTION

Before exporting the model description to Simulink, you should run a time-dependent simulation in FEMLAB.

Model Library `FEMLAB/Multidisciplinary/thermal_controller`

Model Navigator

- On the **New** page, double-click on **Physics modes**, then double-click on the **Heat transfer** tree control to expand it. Select the **Time-dependent** tree control and press **OK** to close the **Model Navigator**. Second order Lagrange elements are used by default.

Draw Mode

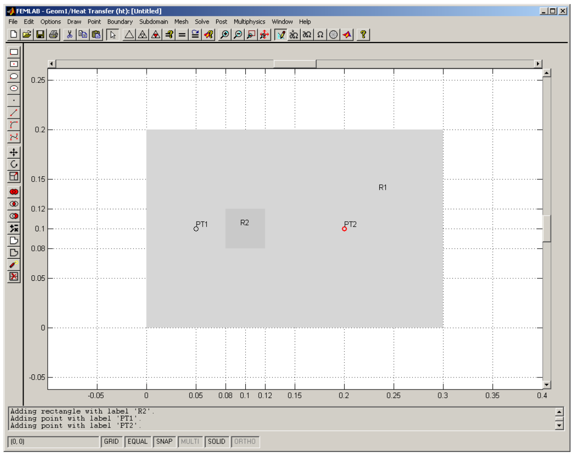
- First modify the scales of the user-interface coordinate axes to fit the size of the metal block (0.3×0.2 m). Choose **Axes/Grid Settings** from the **Options** menu. On the **Axis** page, enter –0.1, 0.4, –0.05, and 0.25 as **X min**, **X max**, **Y min**, and **Y max**, respectively.
- On the **Grid** page, deselect the **Auto** check box and insert 0.08 and 0.12 as extra ticks for the *x*- and *y*-axis. Enter 0.05 for both *x*-spacing and *y*-spacing. Press **OK** to close the dialog box.
- Press the **Draw Rectangle** button to create the object that represents the metal plate.
- Position the pointer at (0,0), press the left mouse button, move the pointer to (0.3,0.2) and release the button. A rectangle denoted `R1` appears. (By double-clicking on the rectangle you can check its coordinates.)

Next model the heater as a small rectangle within Rectangle `R1`:

- Press the **Draw Rectangle** button.
- Draw a rectangle with corners at (0.08,0.08) and (0.12,0.12). An object designated `R2` appears.

You now need two points in the geometry model to measure temperature in the metal plate.

- Press the **Draw Point** button.
- Click at (0.05,0.1) to draw the first point.
- Press the **Draw Point** button again and draw another point at (0.2,0.1).


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
R1
R2
PT1
PT2
(0, 0)
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

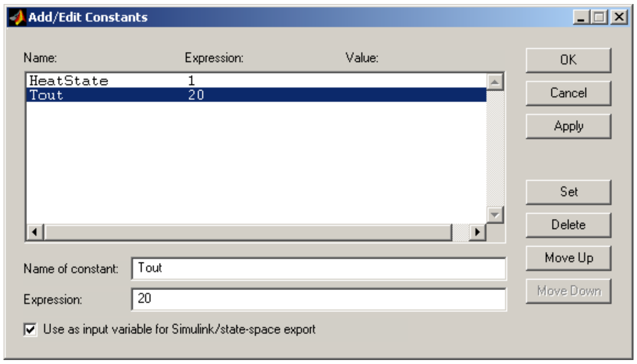
```
Adding rectangle with label 'R2'.
Adding point with label 'PT1'.
Adding point with label 'PT2'.
```

Options and Settings

Before specifying the PDE and boundary parameters, define the constants `HeatState` and `Tout` as below.

NAME	EXPRESSION
`HeatState`	1
`Tout`	20

- Open the **Add/Edit Constants** dialog box on the **Options** menu. Enter the name `HeatState` and the value `1`, then press the **Set** button. Do the same for the constant `Tout` using the value `20`. Press the **OK** button. Note that at this time you're specifying arbitrary values for the constants; they assume new values when you run the model in Simulink.



Name:	Expression:	Value:
HeatState	1	
Tout	20	

Name of constant: Tout

Expression: 20

[✓] Use as input variable for Simulink/state-space export

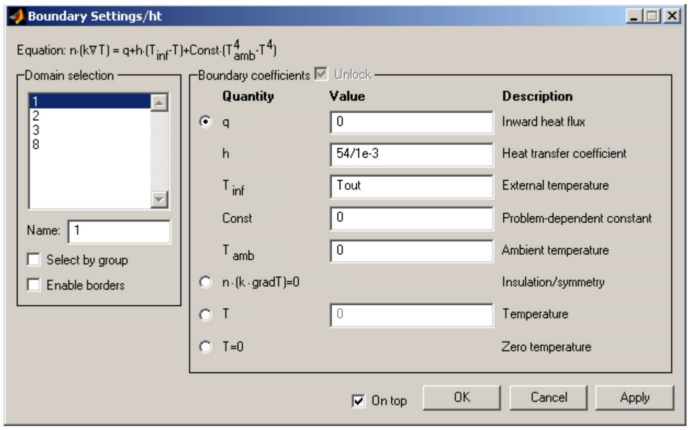
Boundary Mode

Enter the following boundary conditions:

BOUNDARY	**1**	**2,3,8**
Type	Convection	Insulation/ symmetry
q	`54/1e-3*Tout`	
h	`54/1e-3`	

- Open the **Boundary Settings** dialog box by selecting **Boundary Settings...** from the **Boundary** menu.

- For the insulated sides (boundaries 2, 3 and 8) use the **Insulation/symmetry** boundary condition.
- For the poorly insulated side (boundary 1) specify an **Inward heat flux** of `54/1e-3*Tout` and a **Heat transfer coefficient** of `54/1e-3`.


Boundary Settings/ht

Equation: $n\cdot(k\nabla T) = q+h\cdot(T_{inf}-T)+Const\cdot(T_{amb}^4-T^4)$

Domain selection: 1 (selected), 2, 3, 8

Name: 1

☐ Select by group

☐ Enable borders

Boundary coefficients ☑ Unlock

	Quantity	Value	Description
●	q	0	Inward heat flux
	h	54/1e-3	Heat transfer coefficient
	T_{inf}	Tout	External temperature
	Const	0	Problem-dependent constant
	T_{amb}	0	Ambient temperature
○	n·(k·gradT)=0		Insulation/symmetry
○	T	0	Temperature
○	T=0		Zero temperature

☑ On top

- Press **OK** to close the dialog box.

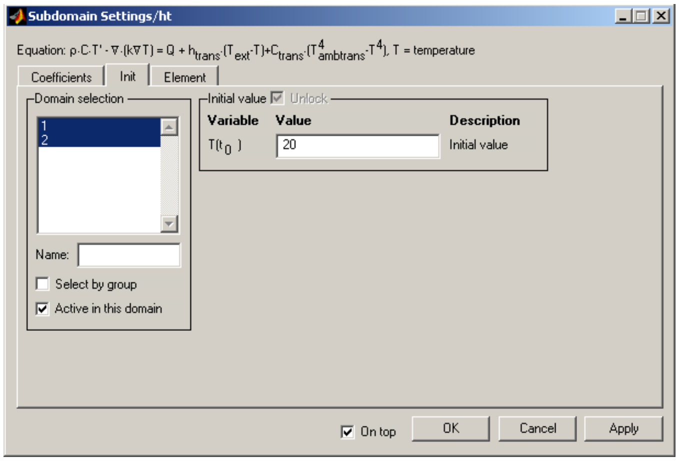
Subdomain Mode

Enter PDE coefficients according to the table:

SUBDOMAIN	1	2
ρ	`7.82e3`	`7.82e3`
C	`449`	`449`
k	`4.92e3`	`4.92e3`
Q	`0`	`7.5e7*HeatState`

- Open the **Subdomain Settings** dialog box by selecting **Subdomain Settings…** from the **Subdomain** menu.
- Assume the metal plate is iron. For both subdomains enter the values `ρ = 7.82e3`, `C = 449`, and `k = 4.92e3`, corresponding to the **Density**, **Heat capacity** and **Coeff. of heat conduction**, respectively.

- The **Heat source** `Q` equals `0` for the metal block (Subdomain 1) and `7.5e7*HeatState` for the heater (Subdomain 2).
- Click the **Init** tab and enter the initial value `20` for **T(t_0)** for both subdomains. Press **OK** to close the dialog box.


Subdomain Settings/ht

Equation: $\rho \cdot C \cdot T' - \nabla \cdot (k \nabla T) = Q + h_{trans} \cdot (T_{ext} - T) + C_{trans} \cdot (T^4_{ambtrans} - T^4)$, T = temperature

Coefficients | Init | Element

Domain selection: 1, 2

Name:

☐ Select by group

☑ Active in this domain

Initial value ☑ Unlock

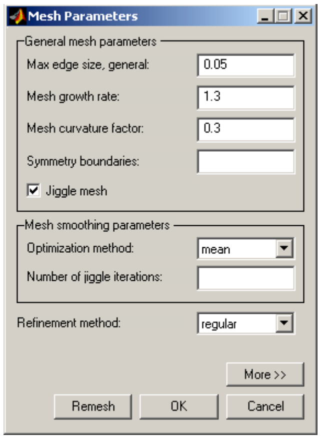
Variable	Value	Description
T(t_0)	20	Initial value

☑ On top

OK | Cancel | Apply

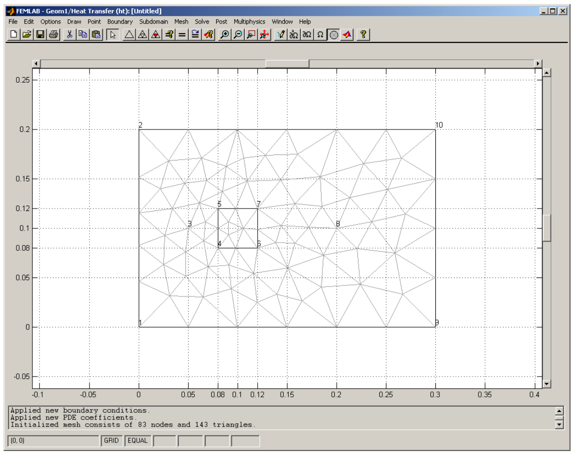
Mesh Mode

- Open the **Mesh Parameters** dialog box by selecting **Parameters…** from the **Mesh** menu.
- Set the **Max. edge size, general** to 0.05 and press **OK**.



Max edge size, general: 0.05
Mesh growth rate: 1.3
Mesh curvature factor: 0.3
Symmetry boundaries:
[✓] Jiggle mesh
Optimization method: mean
Number of jiggle iterations:
Refinement method: regular

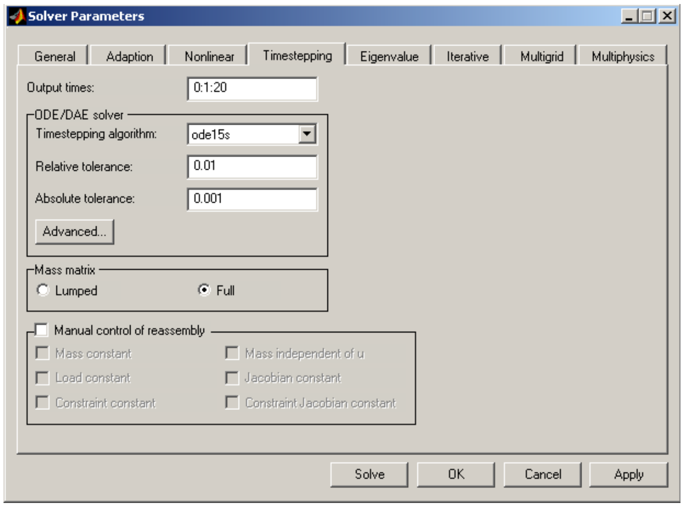
- Press the **Initialize Mesh** button. The mesh parameter settings generate a mesh coarser than the default.


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
GRID EQUAL

```
Applied new boundary conditions.
Applied new PDE coefficients.
Initialized mesh consists of 83 nodes and 143 triangles.
```

Solving

- Select **Show Vertex Labels** on the **Options** menu. In the GUI it's clear that the two points where you expect to detect the temperature have the numbers 3 and 8.
- Open the **Solver Parameters** dialog box by selecting **Parameters...** on the **Solve** menu.
- To run a simulation from 0 to 20 minutes, go to the **Timestepping** page and set `0:1:20` for the **Output times**.



Output times: 0:1:20

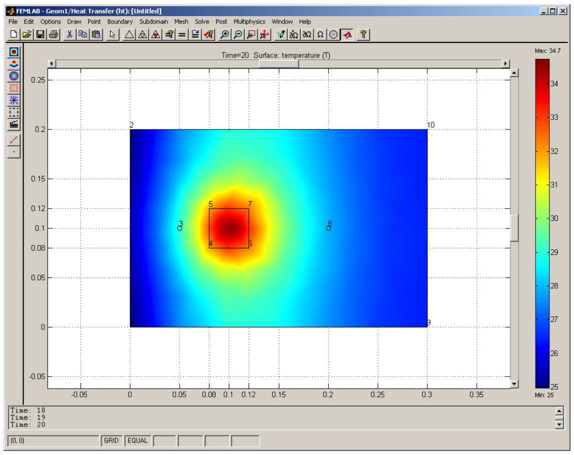
ODE/DAE solver
Timestepping algorithm: ode15s
Relative tolerance: 0.01
Absolute tolerance: 0.001

Mass matrix
Lumped [✓ Full]

☐ Manual control of reassembly
☐ Mass constant ☐ Mass independent of u
☐ Load constant ☐ Jacobian constant
☐ Constraint constant ☐ Constraint Jacobian constant

- Press the **Solve Problem** button to start the simulation. (Time to solve: 8 s)

A plot of the surface temperature at Time = 20 appears as in this figure:


FEMLAB - Geom1/Heat Transfer (ht): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Time=20 Surface: temperature (T)
Max: 34.7
Min: 25
34
33
32
31
30
29
28
27
26
25
0.25
0.2
0.15
0.12
0.1
0.08
0.05
0
-0.05
-0.05
0
0.05
0.08
0.1
0.12
0.15
0.2
0.25
0.3
0.35
2
10
5
7
3
8
4
6
9
(0, 0)
GRID
EQUAL

```
Time: 18
Time: 19
Time: 20
```

You can also run an animation by pressing the **Animation** button.

SIMULINK EXPORT AND TEMPERATURE CONTROL

As mentioned before, it's possible to model the temperature control system without using Simulink and an implementation of this follows from "Controlling Temperature using Non-local Coupling Variables" on page 2-288, along with a short discussion of some of the advantages and disadvantages of each approach. In this case it's preferable to use Simulink because of the nature of the control mechanism, but the alternative is included as an illustration.

The result of exporting a model from FEMLAB to Simulink is a structure in the MATLAB main workspace. To use the model in Simulink, simply drag the **FEMLAB Subsystem** block to your Simulink model as described below.

The following variables affect the temperature of the metal block:

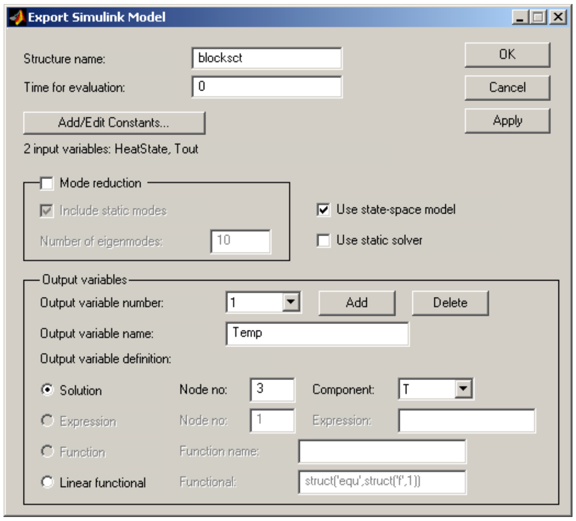
- the exterior temperature, `Tout`
- the state of the heater, `HeatState`, which can assume the values 1 or 0, corresponding to On or Off, respectively.

The output from the FEMLAB model is the temperature in the block where the thermostat is located:

- thermostat temperature, `Temp`

You'll need this variable to control the heater.

- Open the **Export Simulink Model** dialog box from the **File** menu.
- Specify the **Structure name** `blocksct`. This structure contains the FEMLAB data.
- Check the **Use state-space model** check box to linearize the model. Because the FEMLAB model is linear, the linearization doesn't change the solution's accuracy. Using the state-space version of the FEMLAB model prevents MATLAB from calling the FEMLAB kernel from within Simulink, saving computation time.
- To verify that you're working with the correct input variables, check that the variable names `HeatState` and `Tout` appear below the **Add/Edit Constants** button.
- Press the **Add** button, then set the output to be the first component of the solution at node 3 by entering `3` in the edit field for **Node no**.
- Set the **Output variable name** to `Temp`.
- Press **OK** to export the model.

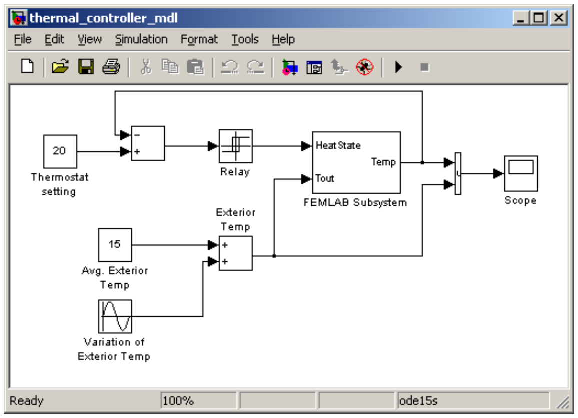

Export Simulink Model

Structure name: blocksct
Time for evaluation: 0
2 input variables: HeatState, Tout

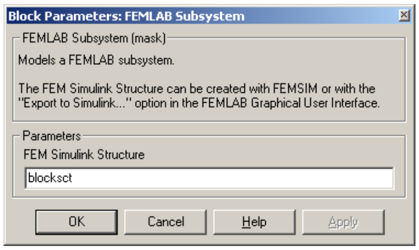
Mode reduction: ☐
Include static modes: ☑
Number of eigenmodes: 10
Use state-space model: ☑
Use static solver: ☐

Output variables
Output variable number: 1
Output variable name: Temp
Output variable definition:
● Solution — Node no: 3 — Component: T
○ Expression — Node no: 1 — Expression:
○ Function — Function name:
○ Linear functional — Functional: struct('equ',struct('f',1))

- Start Simulink by typing `simulink` in the MATLAB Command Window. Immediately draw a model as shown below. You can find the **FEMLAB Subsystem** block under **FEMLAB** in the Simulink library. Refer to the Simulink manual for more information about creating models. Alternatively, you can load a prewritten model from the FEMLAB Model Library into Simulink: `.../models/FEMLAB/Multidisciplinary/thermal_controller_mdl.mdl`.



- To set up the input and output ports of the **FEMLAB Subsystem** block, double-click on that block and set the structure name to `blocksct`. Press **OK** to close the dialog box.


Block Parameters: FEMLAB Subsystem

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.

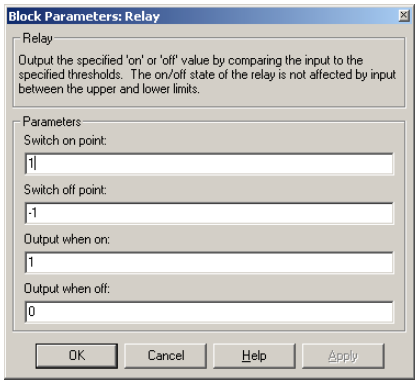
Parameters

FEM Simulink Structure

blocksct

- Specify the values of the constant blocks for **Thermostat Setting** and **Average Exterior Temperature** as in the previous figure.
- 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

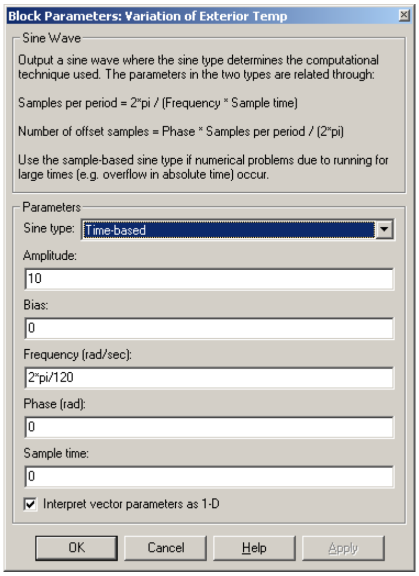
Switch on point: 1

Switch off point: -1

Output when on: 1

Output when off: 0

- 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

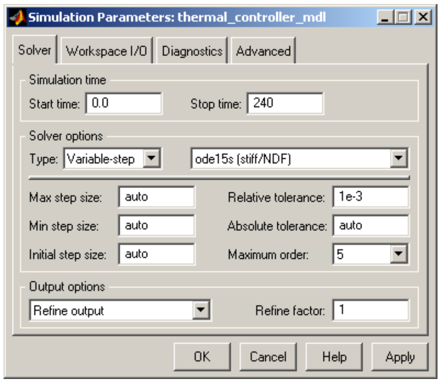
Frequency (rad/sec): 2*pi/120

Phase (rad): 0

Sample time: 0

[✓] Interpret vector parameters as 1-D

- 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.


Simulation Parameters: thermal_controller_mdl

Simulation time

Start time: 0.0 Stop time: 240

Solver options

Type: Variable-step ode15s (stiff/NDF)

Max step size: auto Relative tolerance: 1e-3

Min step size: auto Absolute tolerance: auto

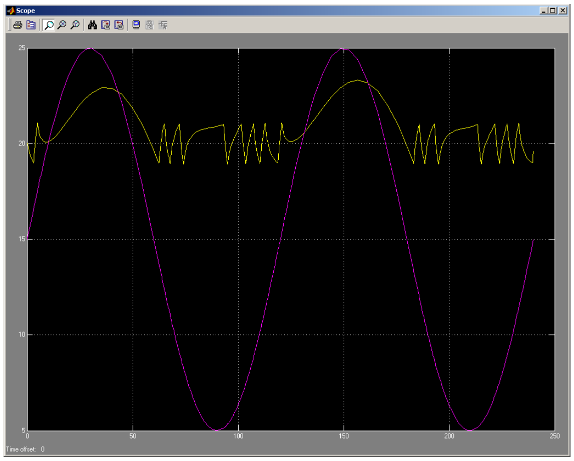
Initial step size: auto Maximum order: 5

Output options

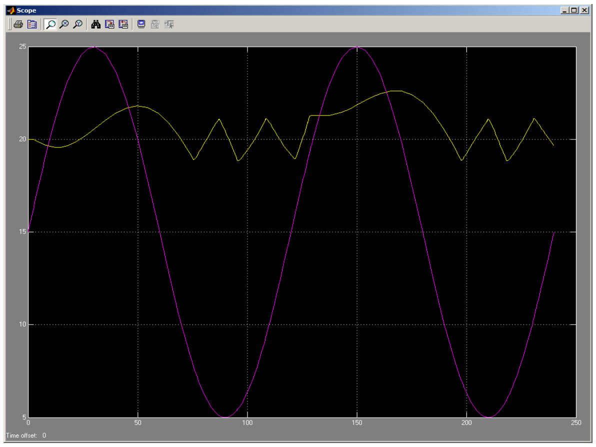
Refine output Refine factor: 1

- Now solve the system by selecting **Start** from the **Simulation** menu.

- To view the temperature at node 3, double-click on the **Scope** block. The result from the simulation appears in the figure below.



- Finally switch node 3 for node 8 in the **Export Simulink Model** dialog box, and press **OK** once more. Then restart the simulation in Simulink.



Note 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 away from the exterior compared to node 3, and the delay is obviously larger in the second plot than in the first.

Note also the qualitative differences between the two simulations. This extra information comes as a bonus for solving a detailed heat-flow problem that takes into account geometry and heat-diffusion properties. Of course, you pay a price in terms of computing speed compared to other modeling approaches such as the one used in the demonstration model `thermo` in Simulink.

Geometry Modeling

This section describes geometry modeling within FEMLAB and explains the concepts of solid modeling and boundary modeling. The examples demonstrate how to use the graphical interface and the FEMLAB functions in programming.

In FEMLAB you can use solid modeling or boundary modeling to create geometry objects.

During solid modeling, you form a geometry as a combination of solid objects using Boolean operations. These geometry-modeling operations—including union, intersection and difference—have been popular since the early days of computer aided design (CAD) because they resemble traditional manufacturing methods such as welding, milling, drilling, and cutting. The use of Boolean operations together with the boundary-modeling method presented here continue to represent the most widespread modeling paradigms in CAD. In FEMLAB, objects you form as a combination of solid objects using Boolean operations are known as *composite solid objects*.

Boundary modeling is the process of defining a solid in terms of its boundaries.

It is also possible to overlay additional non-solid objects on top of solid objects to control the distribution of the mesh and to improve postprocessing capabilities. For example, a `curve3` object can be added to a geometry to be able to control the mesh element size along a 3D curve.

Additionally, you can import geometry objects from, or export to, the MATLAB main workspace. A 2D geometry can also be imported from or exported to a DXF file, and a 3D geometry can be imported from an IGES file. See "Importing CAD Models" on page 1-209 for a description of how to use these CAD file formats.

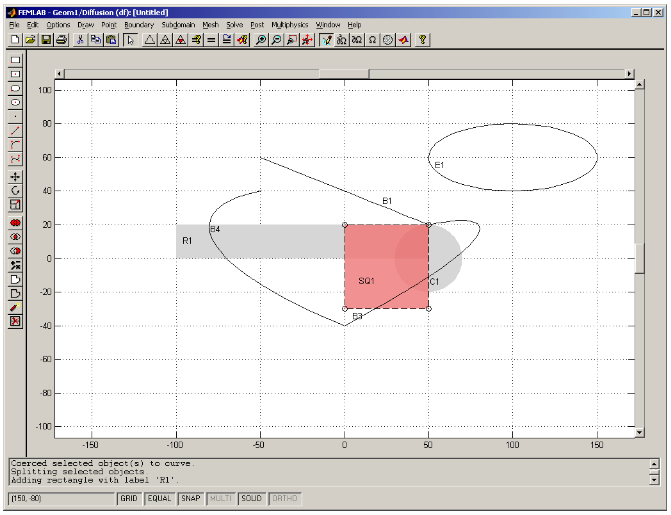
If your geometry data is available in image format or as magnetic resonance imaging (MRI) data, you can convert this into FEMLAB geometry objects, using the programming modeling tools described in the section "Image and MRI Import" on page 1-220.

You can create 2D and 3D spline curves and surface interpolation of measured data from the MATLAB prompt and insert the resulting geometry objects into the GUI. For a description of this, see "Spline Interpolation for Creating Geometry Objects" on page 1-225 and "Surface Interpolation" on page 1-228.

FEMLAB provides the tools for creating 1D, 2D, or 3D geometry objects. This chapter starts with a discussion of 2D modeling because it forms the basis of 3D geometry modeling. For details on the other geometry dimensions, please refer to the sections "Creating a 3D Geometry Model" on page 1-186 or "Creating a 1D Geometry Model" on page 1-207.

Creating a 2D Geometry Model

Using the draw toolbar, you can easily create many different types of geometry objects.


FEMLAB - Geom1/Diffusion (di): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
E1
B1
B4
R1
SQ1
C1
B3
(150, -80)
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

```
Coerced selected object(s) to curve.
Splitting selected objects.
Adding rectangle with label 'R1'.
```

The graphical interface automatically assigns each geometry object a unique name. It supplies defaults in a predefined sequence: for circles, C1, C2, C3, and so on; for rectangles, R1, R2, and R3; for ellipses, E1, E2, and E3. Although squares are just a special type of rectangle, the interface gives them a dedicated name of the type SQ1, SQ2, or SQ3. Composite objects take on designations such as CO1, CO2, and CO3, and Bézier curves become B1, B2, and B3.

The names appear on the objects. In draw mode, the act of double-clicking on an object opens a dialog box in which you can edit the name and other geometry properties. You can change the default name to any unique name as long as it contains no spaces.

SOLID OBJECTS

A *solid object* consists of a boundary part and an interior part. Sometimes a solid object has internal *borders* that separate internal subdomains. A solid object always consists of one or more subdomains that can serve, for instance, to specify material properties in different regions of a solid. Additional points and borders can control the distribution of the mesh inside a solid object, which can be useful for postprocessing purposes.

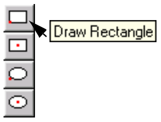
A solid's boundary, including borders, consists of a number of *edge segments*. A solid is completely defined in terms of its boundary (and optionally its internal borders) and is thus said to have a *boundary representation*. For example, a primitive solid rectangle or solid square has a boundary that consists of four line segments and one subdomain. A primitive solid circle or solid ellipse has a boundary that consists of four circular or elliptical arcs and a single subdomain.

SOLID MODELING USING THE GUI

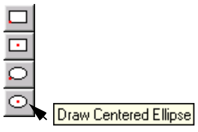
This section demonstrates the use of Boolean operations (union, intersection and difference) on solid objects. These operations are suited for rapid generation of complex geometries. The examples show how to create composite solids starting from primitive solids.

Start by selecting **Geometry only** from the **Model Navigator** tree.

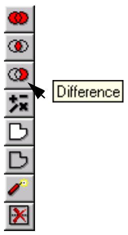
Creating Holes



Let us first learn how to drill a hole through a solid rectangle. Start by creating the solid rectangle. If not already selected, select snap by double-clicking on **SNAP** on the **Status** bar. Go to the top of the draw toolbar and press the button that corresponds to a corner-aligned rectangle, or select **Rectangle/Square** from the **Draw** menu. Using the left mouse button, click and drag from (-1,-1) to (1,1) to describe the rectangle's corners. Depending on settings in the graphical interface, you might need to adjust axis and grid settings to reach these coordinates.

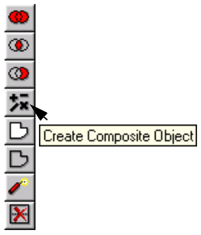


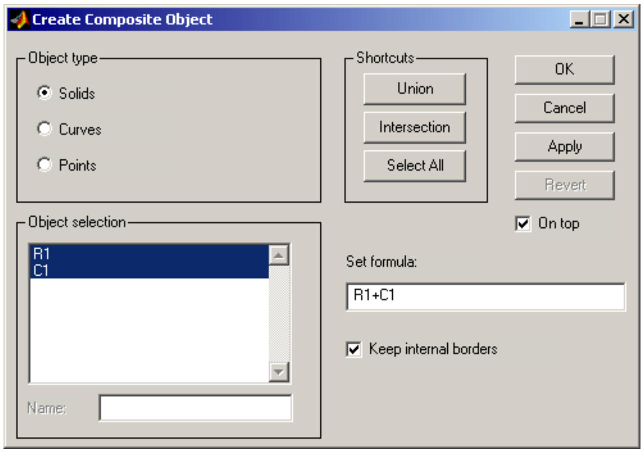
To create the circular hole, press the draw toolbar button for centered ellipse. Click and drag, this time using the right mouse button (the left button lets you create ellipses instead of just circles). Start the center at (0,0) and release the button at a position that results in a solid circle with a radius of 0.5. Depending on the settings you might find it helpful to add an extra grid line, in this case at $x = 0.5$. You can undo the last drawn solid by pressing the **Delete** or **Backspace** keys or by selecting **Undo** from the **Edit** menu.



The act of drilling a hole through the solid rectangle actually involves the creation of a composite solid object. First select both objects presently in the graphical interface (go to **Select All** on the **Edit** menu, or use the shortcut **Ctrl-a)**. Now simply press the **Difference draw toolbar** button and the hole appears. This button subtracts objects with smaller areas from the object with the largest area.

You can also try an alternative approach. Before drilling the hole, select **Create Composite Object...** from the **Draw** menu (or press the corresponding button on the draw toolbar) and this dialog box opens up:




Create Composite Object

Object type
- (●) Solids
- ( ) Curves
- ( ) Points

Shortcuts: Union | Intersection | Select All

OK | Cancel | Apply | Revert

[✓] On top

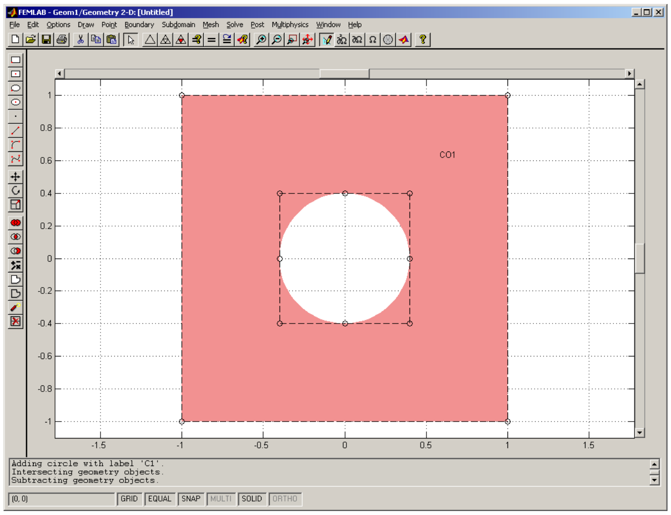
Object selection: R1, C1

Name:

Set formula: R1+C1

[✓] Keep internal borders

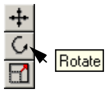
To select both objects, choose **Select all**. Edit the **Set formula** by changing the + sign to a - sign. Doing so results in the set formula `R1-C1`, here meaning that the interface subtracts the solid circle from the solid rectangle. Press **OK** to apply the set formula and close the dialog box. You can see the result of this operation in the figure below.

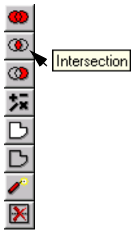

FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
(0, 0)
GRID EQUAL SNAP MULTI SOLID ORTHO

```
Adding circle with label 'C1'.
Intersecting geometry objects.
Subtracting geometry objects.
```

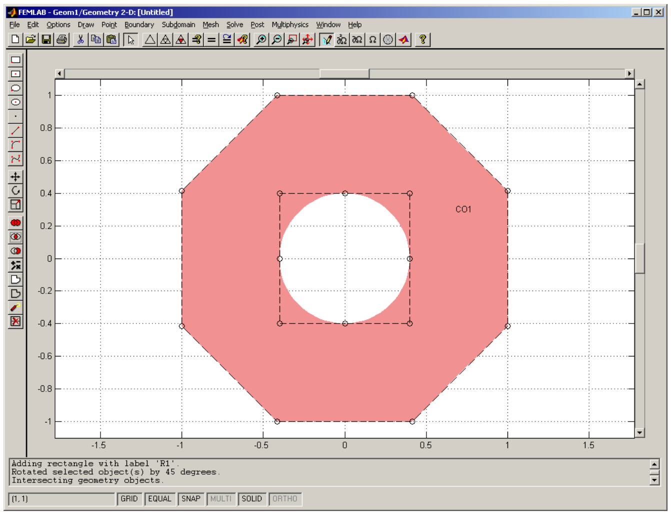
Trimming Solids

Suppose it is necessary to remove the corners on the solid you just created. To do so, create a second solid rectangle with the same size as the previously created rectangle, rotate it 45 degrees, and then form the intersection of the two objects.





In more detail, create a new solid rectangle with corners at (-1,1) and (1,1). Press the **Rotate** button on the draw toolbar to open the **Rotate** dialog box. Enter 45 degrees of rotation, press **OK** to apply that value and close the dialog box. To cut the corners off, first select all objects with, for example, **Ctrl-a**. Press the **Intersection** button on the draw toolbar to form the intersection of the first and the second solid objects. The figure below shows the result of these operations.

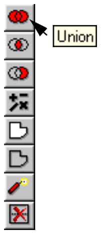

FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.5
-1
-0.5
0
0.5
1
1.5
(1, 1)
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

```
Adding rectangle with label 'R1'.
Rotated selected object(s) by 45 degrees.
Intersecting geometry objects.
```

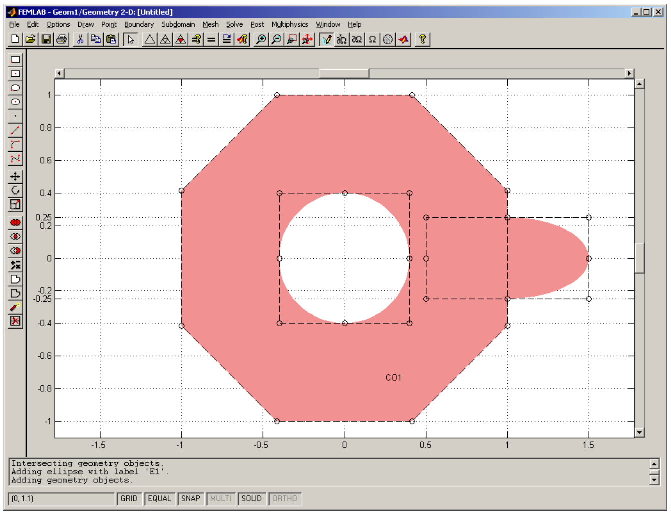
Adding Domains

What if you want to add a domain to a solid? Use the union operation. Continuing with the solid with the trimmed corners, let us assume you want to attach an elliptical domain on its right side.

First open the **Axes/Grid Settings** dialog box from the **Options** menu. On the **Grid** page, deselect automatic grid spacing by unchecking the **Auto** check box. Then add extra ticks in the y-direction by entering `-0.25 0.25` in the **Extra Y** ticks edit field. In addition, add a grid spacing of 0.5 in the *x*-direction and 0.2 in the *y*-direction. Press **OK** to apply these values and close the dialog box.



Now draw an ellipse centered at (1,0) with semi-axes of length 0.5 and 0.25 in the x- and y-directions, respectively. Use the draw toolbar button for a centered ellipse. Finally select all the objects and press the draw toolbar icon for union.


FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
(0, 1.1)
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

```
Intersecting geometry objects.
Adding ellipse with label 'E1'.
Adding geometry objects.
```

USING THE PROGRAMMING LANGUAGE

Now let us learn how to run these same Boolean operations on solid objects from MATLAB.

The following examples illustrate how to create composite solids starting with primitive solids, repeating what you just did with the graphical interface. Some readers might want to skip over this section during a first reading of this manual because you generally perform geometry modeling with the graphical interface. However, the programming language can prove useful for creating geometry objects such as when sequences of commands compute exact coordinates for objects. You can import the resulting geometry objects into the graphical interface for further manipulations.

Creating Holes

To learn how to drill a hole, first create a solid rectangle with corners at (-1,-1) and (1,1):

```
s1 = rect2(-1,1,-1,1);
geomplot(s1)
axis equal
```

In the first line, the function `rect2` creates a solid rectangular object `s1`. The function `geomplot` plots the rectangle, which in this case is a square. The command `axis equal` is necessary for proper visualization of the rectangle as a square.

Now create a circular hole with a radius of 0.5 and center it at (0,0).

```
s2 = circ2(0,0,0.5);
```

The function `circ2` creates a circular object `s2`. Use it to effectively drill a hole in the first object (the square) by forming the difference

```
s3 = s1-s2;
```

or by entering

```
s3 = geomcomp({s1,s2},'ns',{'s1','s2'},'sf','s1-s2');
```

Finally, visualize the result

```
geomplot(s3)
axis equal
```

When entering `s3 = s1-s2` you take advantage of MATLAB's object-orientated capabilities by means of object arithmetic. In this case, both `s1` and `s2` are solid objects, and the operation you run is equivalent to calling the function `geomcomp` with the parameters shown. The intersection operator "`*`" and the union operator "+" work in a similar fashion. The function `geomcomp` analyzes the specified geometric objects and returns the analyzed geometric object. The analyzed object is of the same class as the input objects.

Trimming Solids

To remove the corners from the solid rectangle, create a second solid rectangle of the same size, rotate it 45 degrees, and then take the intersection of the two. Notice that this sequence effectively cuts material from the outside of the rotated solid rectangle.

First rotate `s1` about (0,0)

```
s2 = rotate(s1,45*(pi/180),0,0);
```

then form the intersection

```
s = s3*s2;
```

and visualize the result

```
geomplot(s)
axis equal
```

The reason for `(pi/180)` in the argument for `rotate` is that the function expects you to express the angle of rotation in radians rather than degrees.

Adding Domains

To add a domain to a solid, use the union operator (`+`). Take the object you're creating, still named solid `s`, and attach an elliptical domain on its right side.

First create a solid ellipse centered at (0,0) and with semi-axes (0.5, 0.25):

```
e = ellip2(0,0,0.5,0.25,45*pi/180);
```

The final argument, `45*pi/180`, expresses the angle of rotation in radians with respect to the semi-axes and the coordinate axes. Now rotate the solid ellipse by -45 degrees to get an ellipse with semi-axes parallel to the coordinate axes. We include this step merely to illustrate how to rotate objects using FEMLAB functions:

```
e = rotate(e,-45*pi/180);
```

Move the center of the ellipse from (0,0) to (1,0):

```
e = move(e,1,0);
```

The second and third parameters in the function call correspond to the *x*- and *y*-displacements. Finally, use the union operation to add the solid ellipse to the solid `s`

```
s = s+e;
```

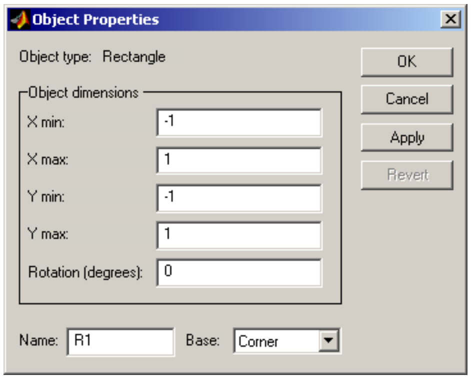
and visualize the result

```
geomplot(s)
```

EDITING SOLIDS IN THE GUI

You can move objects by clicking and dragging them using the left mouse button. To edit the intrinsic geometrical properties of a geometry object, choose **Object**

Properties from the **Draw** menu or double-click on the object using the left mouse button.



Object type: Rectangle

Object dimensions

X min:	-1
X max:	1
Y min:	-1
Y max:	1
Rotation (degrees):	0

Name: R1 Base: Corner

The contents of the **Object Properties** dialog box vary depending on which type of object you have selected for editing. For primitive solid objects, you can modify object-specific information such as center and semi-axes for ellipse primitives and corner positions for rectangle primitives. For composite solids, the entries allow editing of boundary curves and the removal of borders. For details see “Object Properties…” on page 4-55. You can also edit the name of both primitive solids and composite solids.

An alternative way for modifying control vertices is a drag-and-drop technique. Click on one of the control vertices of a solid’s boundary using the left mouse button, hold the button down and drag the vertex to a new position, then release the button. Any change that alters a solid’s topology, such as one that creates new intersections or wraps a solid’s domain, results in a warning message.

SUBDOMAINS AND THE SPLIT OPERATION

When you work with the union operator, the internal borders will partition the resulting solid into subdomains. These subdomains can, for example, specify different material properties. If you do not want FEMLAB to create these borders, deselect the check box **Keep internal borders** in the **Create Composite Object** dialog box.

It is possible to split a solid consisting of multiple subdomains into a collection of minimal solids that each consists of only one domain. To do so in the graphical interface, press the **Split** button on the draw toolbar. Using programming for the same task, the line of code

```
ss = split(s);
```

divides the solid s into multiple solid objects, all of which are now contained in the cell array ss.

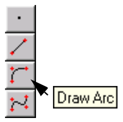
CURVE OBJECTS

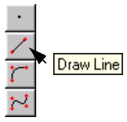
When drawing a line, arc, or any other curve, you are creating a *curve object*. Primitive curve objects in FEMLAB include lines, arcs (or 2nd degree Bézier curves), and 3rd degree Bézier curves. A composite curve object consists of a collection of primitive curve objects, and to form one, select **Create Composite Object…** from the **Draw** menu. Before doing so, hold down the **Ctrl** key to select the constituent objects and click on the curves using the left mouse button; alternatively, enclose them with a rubber band box.

BOUNDARY MODELING

When using boundary modeling, you explicitly define a composite solid object's boundary by drawing each curve segment of it. The example below shows how to create a composite solid by means of boundary modeling. The solid consists of two line segments and four arc segments.

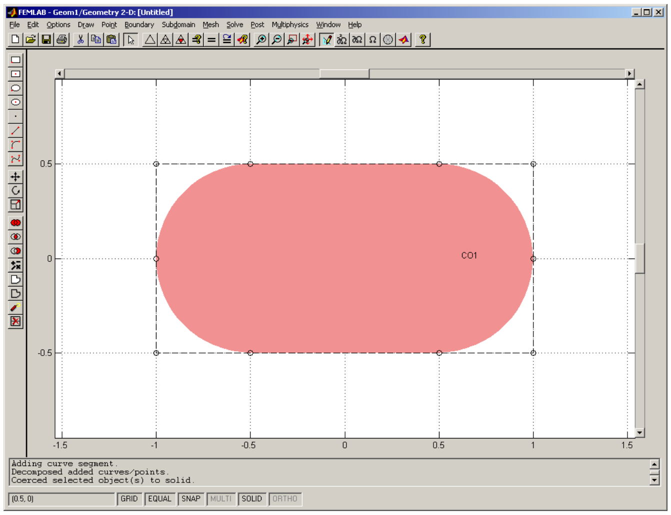
USING THE GRAPHICAL USER INTERFACE





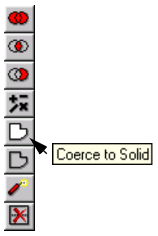
If you haven't already done so, select grid snapping by double-clicking **SNAP** on the **Status** bar. Open the **Axes/Grid Settings** dialog box from the **Options** menu. On the **Grid** page, deselect the **Auto** check box and enter a grid spacing of 0.5 for both the *x*- and *y*-directions. Press **OK** to apply these selections and close the dialog box.

Choose **Arc** from the **Draw** menu or press the **Draw Arc** button on the draw toolbar. Create the boundary by clicking at (-0.5,-0.5), (-1,-0.5), (-1,0), (-1,0.5) and (-0.5,0.5) using the left mouse button. This action generates two arcs. Now choose **Line** from the **Draw** menu or press the **Draw Line** button on the draw toolbar. Click at (0.5,0.5) to create a horizontal line segment. Choose **Arc** again and click at (1,0.5), (1,0), (1,-0.5) and (0.5,-0.5). Finally, clicking on the right mouse button creates a solid object.

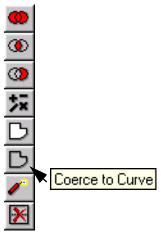

FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
GRID EQUAL SNAP MULTI SOLID ORTHO

```
Adding curve segment.
Decomposed added curves/points.
Coerced selected object(s) to solid.
```

(0.5, 0)



You can force the transformation of a solid object into a curve object by choosing **Coerce Object(s) To > Curve** from the **Draw** menu or by pressing the **Coerce to Curve** button on the draw toolbar. This action coerces the boundary into one composite curve object. To split a composite curve object into its constituent primitive objects, select **Split Object** from the **Draw** menu or press the **Split** button on the draw toolbar. You can also coerce a curve object to a solid object by using the **Coerce Object(s) To > Solid** functionality, either by using the corresponding **Draw** menu item or draw toolbar button.



USING THE PROGRAMMING LANGUAGE

Let us now create the solid from the previous example using boundary modeling from the MATLAB prompt. First create the six boundary curve objects:

```
w = 1/sqrt(2);
c1 = curve2([-0.5 -1 -1],[-0.5 -0.5 0],[1 w 1]);
c2 = curve2([-1 -1 -0.5],[0 0.5 0.5],[1 w 1]);
c3 = curve2([-0.5 0.5],[0.5 0.5]);
c4 = curve2([0.5 1 1],[0.5 0.5 0],[1 w 1]);
c5 = curve2([1 1 0.5],[0 -0.5 -0.5],[1 w 1]);
c6 = curve2([0.5 -0.5],[-0.5 -0.5]);
```

The objects `c1,c2,c3,c4,c5`, and `c6` are all `curve2` objects. The vector `[1 w 1]` specifies the weights for a rational Bézier curve that is equivalent to a quarter-circle arc. The graphical interface automatically adjusts weights so that it always creates elliptical or circular arcs. For more information on available geometry objects see the *Reference Manual*.

Create a solid object `s` of the object type `solid2`

```
  s = geomcoerce('solid',{c1,c2,c3,c4,c5,c6});
```

and visualize the result

```
  geomplot(s)
```

You can create a single-curve object by converting a `solid2` object to a `curve2` object

```
  c = curve2(s);
```

You can accomplish the same thing with a call to `geomcomp`

```
  c = geomcomp({c1,c2,c3,c4,c5,c6});
```

ARCS

The solid you created in the previous example has rounded corners consisting of circular arcs. When drawing arcs, whether you create circular or elliptical curve segments depends on the control polygon. *FEMLAB draws a circular arc only if the control polygon is isosceles.* In other words, to ensure an arc is circular, keep the distance between the first and second control vertices equal to the distance between the second and third control vertices.

The underlying representation of arcs is a 2nd degree Bézier curve with weights that depend on the shape of the control polygon. When using the graphical interface, FEMLAB automatically adapts the weights to the shape of the control polygon in order to create elliptical or circular curve segments. For more information on the underlying representation of curves, see the *Reference Manual*.

RATIONAL BÉZIER CURVES

This section gives an elementary introduction to some of the more important properties of rational Bézier curves. It also describes how to use the graphical interface or the programming language to generate them.

Why Rational Bézier Curves?

When you work in draw mode and create a geometry object with a curved boundary, FEMLAB represents that object in terms of rational Bézier curves. This representational scheme has become a standard in the CAD community because of its intuitive behavior and representational power. Rational Bézier curves of degree 2 can represent all different types of conic sections including circles, ellipses, parabolas, and hyperbolas. FEMLAB includes 3rd degree rational Bézier curves to assist in the creation of additional free-form shapes such as curves with inflection points (S-shaped curves) and advanced fillets.

Control Vertices and Weights

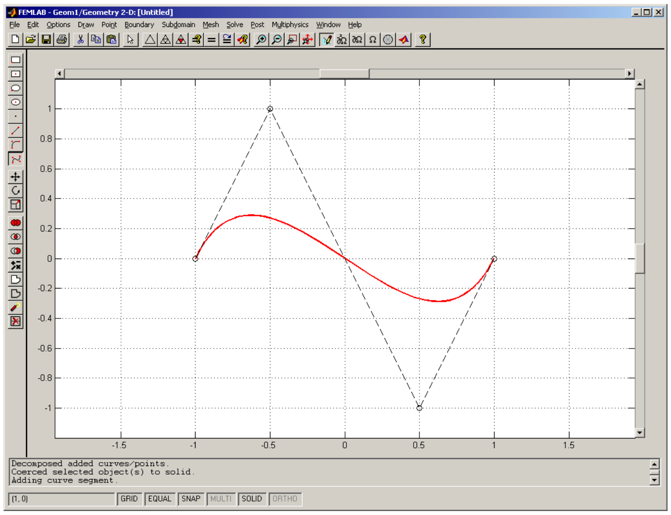
A rational Bézier curve of degree n is completely defined in terms of a control polygon consisting of $n+1$ control vertices and $n+1$ associated control weights. The curve always interpolates the two end points of the control polygon. A change in the control polygon's shape, such as by moving a control vertex, produces a corresponding change in the curve's appearance. Its shape, in a sense, always mimics that of the control polygon. The higher curve degree, n, the more complicated are the shapes you can create due to the increased number of control vertices. If you increase the weight associated with one of the control polygon points, that action pulls the curve toward the corresponding control vertex.

This simple and intuitive interaction between the control polygon, with its weights, and the curve makes the rational Bézier curve representation so useful. Below you can examine some simple examples of how alterations in control vertices and weights affect curve shape.

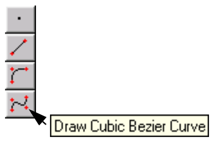
Using the Graphical User Interface

Start by examining an S-shaped 3rd degree rational Bézier curve. To produce this shape, position the control vertices at $\mathbf{b}_0 = (-1,0)$, $\mathbf{b}_1 = (-0.5,1)$, $\mathbf{b}_2 = (0.5,-1)$, and $\mathbf{b}_3$

= (1,0) in a zigzag pattern. FEMLAB starts with the control weights w_0, w_1, w_2, and w_3 all set to 1.

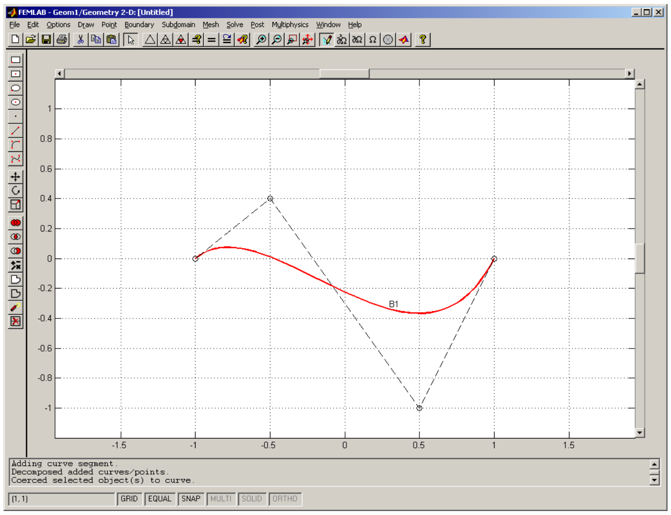

FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
(1, 0)
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

```
Decomposed added curves/points.
Coerced selected object(s) to solid.
Adding curve segment.
```



To create the curve shown above, first open the graphical interface. If you haven't already done so, select grid snapping by double-clicking on **SNAP** on the **Status** bar. Deselect the solidifying mode by double-clicking on **SOLID** on the **Status** bar. Press the **Draw Cubic Bézier Curve** button on the draw toolbar, or select **3rd Degree Bézier Curve** from the **Draw** menu. Using the left mouse button, click in order at (-1,0), (-0.5,1), (0.5,-1) and (1,0). You can also click and hold down the left mouse button at the starting point, move to the next point and release the button, repeating this process for each point. This drag-and-drop technique gives you visual feedback and shows the control polygon's edges while creating the curve.

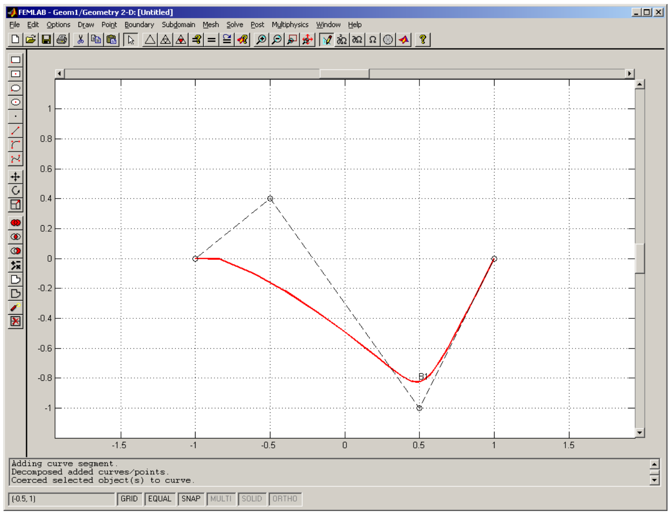
Deselect the 3rd degree curve mode by clicking on the right mouse button or by pressing the toolbar for cubic Bézier curves a second time. Doing so coerces the curve into an editable curve object. Next move $\mathbf{b}_1$ to (-0.5,0.4) by clicking on the second control vertex at (-0.5,1), holding down the left mouse button and releasing it at the new location. The curve adapts to the altered shape of the control polygon:


FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
B1
GRID EQUAL SNAP MULTI SOLID ORTHO

```
Adding curve segment.
Decomposed added curves/points.
Coerced selected object(s) to curve.
```

(1, 1)

Increasing weight w_2 (the third weight) from 1 to another value, for example 7, pulls the curve towards the corresponding control vertex $\mathbf{b}_2$:


FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
B1
GRID EQUAL SNAP MULTI SOLID ORTHO

```
Adding curve segment.
Decomposed added curves/points.
Coerced selected object(s) to curve.
```

(-0.5, 1)

You can perform this action in the graphical interface by opening the **Object Properties** dialog box from the **Draw** menu or by double-clicking on the curve object.

Choose the **Weights** page, select Curve 1 and alter Weight 3 from 1 to 7. Finish by pressing the **OK** button. Be especially aware that weights mathematically denoted w_0, w_1, w_2, and w_3 correspond to Weights 1, 2, 3, and 4 in the graphical interface.

Using the Programming Language

To achieve the same result as above, use this code:

```
c = curve2([-1 -0.5 0.5 1],[0 1 -1 0],[1 1 1 1]);
geomplot(c,'ctrlmode','on')
```

In the first line the constructor creates a `curve2` object named `c`, which corresponds to the S-shaped curve. In the next line the method `geomplot` plots this object. You can find a detailed discussion of various geometry methods and the geometry class structure in the *Reference Manual*.

When you move the control vertex $\mathbf{b}_1$ to (-0.5,0.4), the curve adapts to the altered shape of the control polygon.

Change the position of the second control vertex by entering

```
c = change(c,2,-0.5,0.4);
```

or equivalently

```
c = curve2([-1 -0.5 0.5 1],[0 0.4 -1 0]);
```

and plot the changed curve

```
geomplot(c,'ctrlmode','on')
```

Note that you can omit the weights when calling `curve2`; FEMLAB sets all weights to 1 in this case.

If you now increase weight w_2 from 1 to 7, that change pulls the curve towards the corresponding control vertex $\mathbf{b}_2$.

Enter the following code to change the weights:

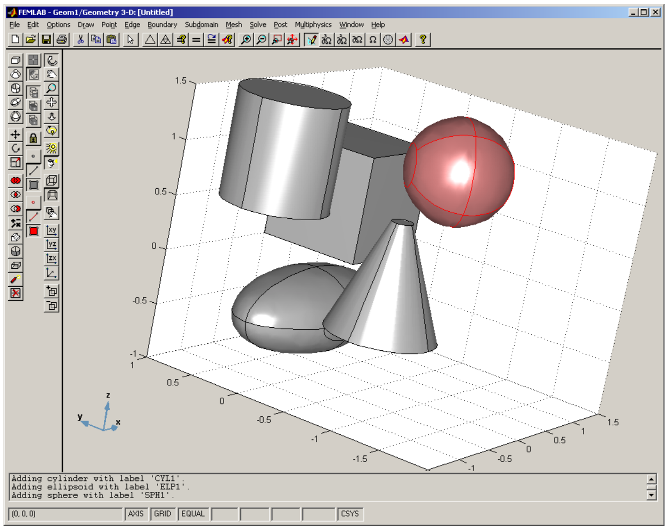
```
c = change(c,1,[1 1 7 1]);
geomplot(c,'ctrlmode','on')
```

Creating a 3D Geometry Model

The computer's two-dimensional screen limits the possibilities of defining a 3D geometry object by means of clicking and dragging the mouse. The standard way of creating 3D geometries is to model a cross section in 2D and then use operations such as *extrude* or *revolve* to extend it into a 3D geometry object. You can implement even more complex geometries by using Boolean operations as described below. In FEMLAB, you model 2D cross sections in local 2D coordinate systems called *work planes* that you position anywhere in space.

Clearly 2D modeling is an essential part of the 3D modeling process. If you are not familiar with 2D modeling, you may want to review the previous section, “Creating a 2D Geometry Model” on page 1-171 before proceeding with this chapter.

FEMLAB also supplies a set of 3D primitives you can create easily without first activating a work plane. These include *blocks*, *cones*, *cylinders, ellipsoids*, and *spheres* and appear at the top of the draw toolbar.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
1.5
1
0.5
0
-0.5
-1
z
y
x
AXIS
GRID
EQUAL
CSYS

```
Adding cylinder with label 'CYL1'.
Adding ellipsoid with label 'ELP1'.
Adding sphere with label 'SPH1'.
```

(0, 0, 0)

When you create a geometry object, the graphical interface automatically assigns it a unique name. Default names for cylinders start with CYL1, moving on to CYL2, CYL3, and so on; it uses BLK1, BLK2, BLK3, and so on for blocks; while CON1, CON2, and CON3 are used for cones. A geometry object you create from a work plane with an extrusion operation takes on names such as EXT1, EXT2, or EXT3. Similarly, a revolve operation results in names such as REV1, REV2, or REV3.

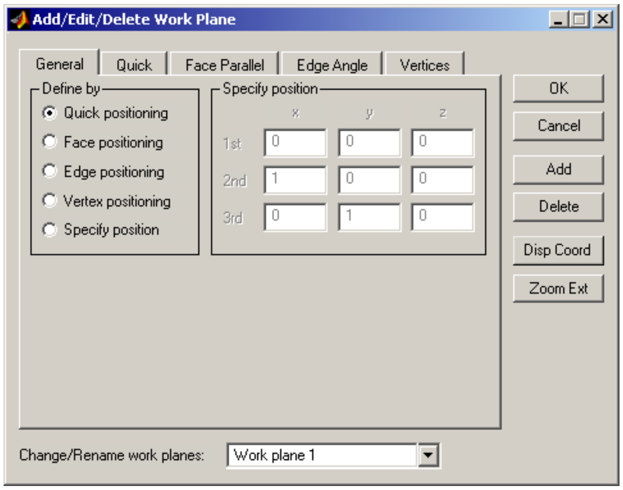
CREATING WORK PLANES

A work plane is a 2D plane positioned anywhere in 3D space. To completely define such a plane you need three points and a normal direction. In FEMLAB you can specify a work plane’s orientation by specifying these properties either explicitly, by specifying its coordinates, or by using existing 3D geometry objects as reference points.

To create a work plane, open the **Add/Edit/Delete Work Plane...** dialog box from the draw menu. Note that if you have any objects selected at this time, all others temporarily disappear. In the dialog box you can choose among five ways to specify a work plane. You can add new work planes, edit or delete existing work planes, and rename them.

On the **General** page you select a method: **Quick positioning**, **Face positioning**, **Edge positioning**, **Vertex positioning** or **Specify position**. Use quick positioning to obtain a work plane equal to the *xy*-, *yz*-, or *zx*-plane. When creating a work plane using face positioning, edge positioning, or vertex positioning, you select existing 3D objects as reference points. Another possibility to define a work plane is to specify three points in 3D space. In this case, the normal direction depends on the order of the specified points. For more information on the different methods to create a work plane, see the section “3D Draw Menu” on page 4-59 in the *Reference Manual*.

Pressing the **Disp Coord** toggle button displays a small axis in the 3D view that indicates a work space’s origin and its normal vector direction. Pressing **OK** for this dialog allows you to then enter the currently selected work plane and start working in that local 2D coordinate system.



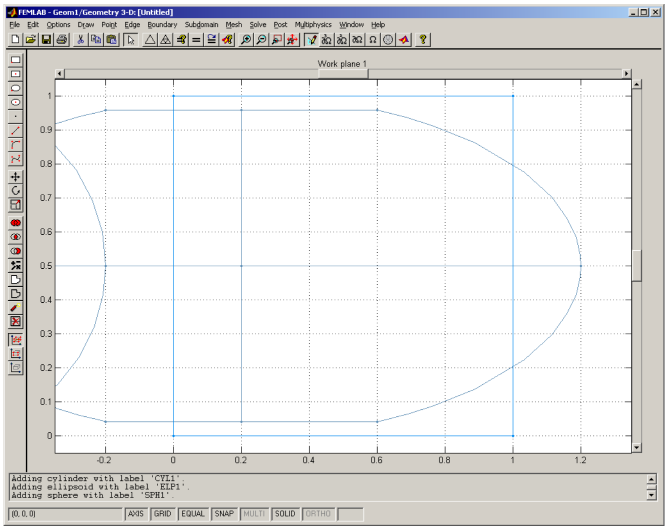
Define by
[✓ Quick positioning]
Face positioning
Edge positioning
Vertex positioning
Specify position

Specify position

	x	y	z
1st	0	0	0
2nd	1	0	0
3rd	0	1	0

Change/Rename work planes: Work plane 1

When you enter a work plane, a 2D draw toolbar replaces the 3D draw toolbar. It contains the same buttons as the standard 2D toolbar but adds three more at the bottom that define which portion of the 3D geometry to project onto the plane. This method helps you capture the orientation of a work plane in 3D space. Here you can choose to see a projection of the entire 3D geometry, see parts of the geometry that intersect the current plane, or see nothing of the 3D geometry. FEMLAB considers this geometry projection as a background plot to which you can snap the cursor when drawing 2D objects.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 1
(0, 0, 0)
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

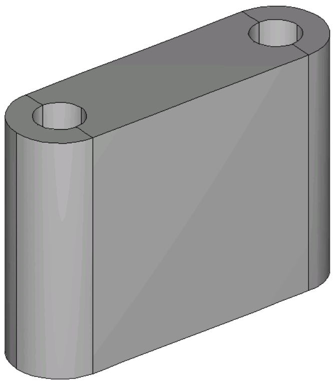
```
Adding cylinder with label 'CYL1'.
Adding ellipsoid with label 'ELP1'.
Adding sphere with label 'SPH1'.
```

Drawing in a work plane is essentially the same as 2D geometry modeling. Here you create a cross-section geometry using standard 2D solid modeling or boundary modeling techniques. It is possible to embed, extrude, or rotate objects to form a 3D geometry object.

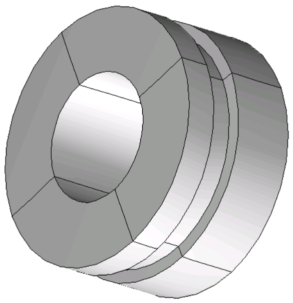
EXTRUDE, REVOLVE, AND EMBED OPERATIONS

The **Extrude** item in the **Draw** menu opens a dialog box in which you select objects to extrude. You also specify the length of the extruded object in the direction perpendicular to the plane and other optional parameters. See "Extrude…" on page

4-69 for a detailed description of this dialog box. The figure below shows the result of an extrusion.



The **Revolve** menu item rotates the selected 2D geometry objects between two angles and about an axis that you specify in the **Revolve** dialog box. See “Revolve…” on page 4-70 for a details. Below, a result of a revolution is shown.



You can also embed 2D geometry objects in 3D space, yet still in their existing form, by choosing the **Embed** menu item.

SOLID OBJECTS

As is the case with 2D solid objects, a *3D solid object* consists of a boundary part and an interior part. The boundary, including any borders, consists of a number of *face segments*. You can completely define a solid in terms of its boundary and optionally its internal borders, and thus we say it has a *boundary representation*. For example, a primitive solid block object or solid cube object has a boundary that consists of six rectangular face segments and one subdomain. A primitive solid cylinder has a boundary that consists of two circular faces (top and bottom), four bent rectangular faces building the lateral area, and a single subdomain.

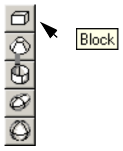
SOLID MODELING AND BOOLEAN OPERATIONS

This section demonstrates the use of work planes and the Boolean operations (union, intersection and difference) on solid objects in the graphical interface. The Boolean operations are suited for rapid generation of complex geometries.

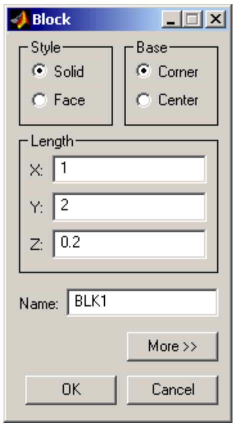
The following examples show how to use the graphical interface to create composite solids from those you previously created using work planes.

Start by selecting the **3D** radio button in the **Dimension** frame. Then select **Geometry only** from the **Model Navigator** tree.

Using the Graphical User Interface



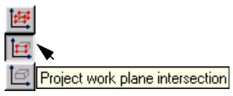
The first operation shows how to drill holes through a block primitive. Start by pressing the **Block** toolbar button, or select **Block** from the **3D Primitives** submenu in the upper part of the **Draw** menu. This action opens a dialog box in which you specify the primitive's dimensions and position.



Style: [✓ Solid] Face
Base: [✓ Corner] Center
X: 1
Y: 2
Z: 0.2
Name: BLK1

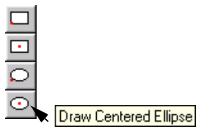
In the **Style** frame you can choose to create a face object or a solid object. The **Base** frame defines where to put the origin of the block's internal coordinate system. Pressing the **More** button reveals additional parameters that specify the block's position in 3D space. For this exercise, enter the length of the block's sides as 1, 2, and 0.2 according to the figure above and press **OK**.

Open the **Add/Edit/Delete Work Plane** dialog box from the **Draw** menu. On the **Quick** page select the **Quick positioning** radio button. Select the **x-y** radio button to create a work plane in the *xy*-plane. To view its position and normal direction, simply press the **Disp Coord** button. You can change the work plane's name by clicking on its title at the bottom of the dialog box. Press **OK** and you automatically enter the work plane.

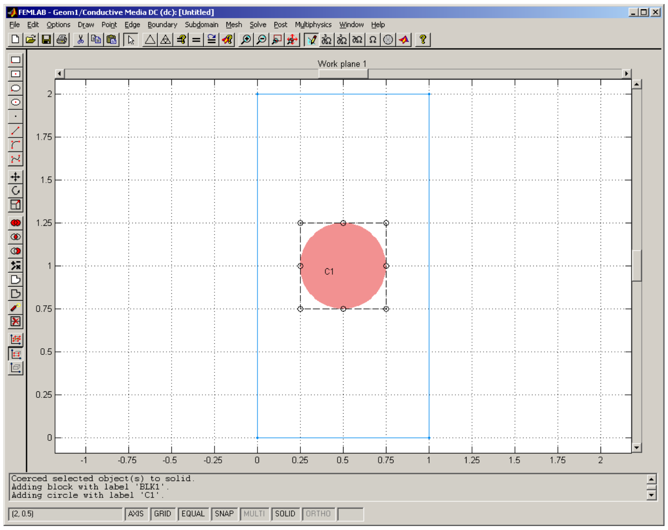


Now that you have entered that local 2D coordinate environment, press the **Project Work Plane Intersection** toolbar button at the bottom of the draw toolbar if it's not already selected. Doing so shows a blue wireframe plot of the 3D geometry. The three bottom toolbar buttons together with the **Geometry Object Suppression** dialog box on the **Draw** menu control how much of the 3D geometry is displayed in the work plane. See "Suppressing Geometry Objects" on page 1-106 for more information on how to suppress geometry objects.

Using the background image for orientation, you can easily find the middle of the block. Open the **Axes/Grid Settings** dialog from the **Options** menu. On the **Grid** tab, uncheck the **Auto** check box and enter 0.25 in the **X-spacing**, and **Y-spacing** edit fields. Press **OK**

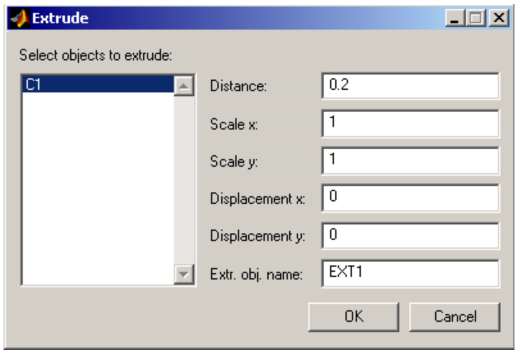


To create the circular hole, press the toolbar button for centered ellipse. Click the right mouse button and drag the cursor starting at the circle's center at (0.5,1) and releasing the mouse button when you've created a solid circle with a radius of 0.25.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 1
C1
2
1.75
1.5
1.25
1
0.75
0.5
0.25
0
-1
-0.75
-0.5
-0.25
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
(2, 0.5)
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

```
Coerced selected object(s) to solid.
Adding block with label 'BLK1'.
Adding circle with label 'C1'.
```

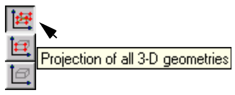
Now open the **Extrude** dialog box from the **Draw** menu to turn the circle into a 3D cylinder.



Distance: 0.2
Scale x: 1
Scale y: 1
Displacement x: 0
Displacement y: 0
Extr. obj. name: EXT1

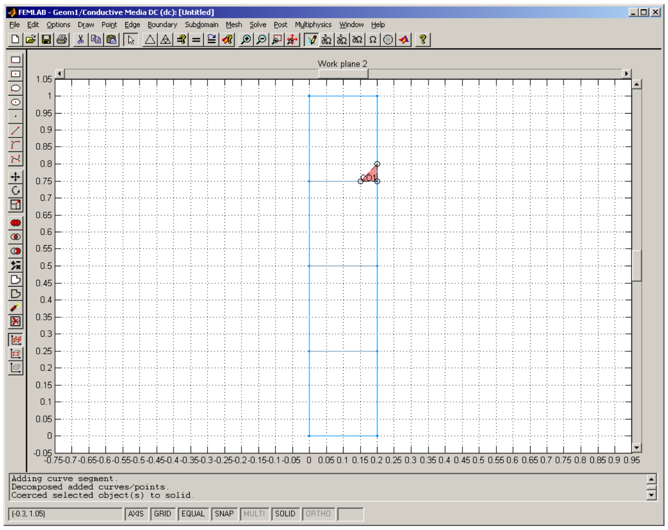
Select C1 in the object list box and enter the distance 0.2 in the **Distance** edit field. Press **OK**. You can extrude an object in numerous ways. This includes scaling the geometry in the *x*- or *y*-direction to create conic geometries and skew the geometry by setting an *x*- or *y*-displacement. You can also make piece-wise linear and cubic extrusions by entering matrices in these edit fields. See the reference entry on `extrude` on page 5-93 in the *Function Reference* for details.

To create a chamfer, that is, a flattened corner, select the block by clicking it, and open the **Add/Edit/Delete Work Plane** dialog box. Press the **Add** button to create a new work plane, which FEMLAB names *Work plane 2*. On the **Face Parallel** page, check the **Face positioning** radio button and select face 3. Enter an offset of 1 in the **Offset from face** edit field. Choose **Upward normal** in the **Work plane z-axis direction** frame. Press the **Disp Coord** button to view the work plane's position and normal direction. Press **OK**, which puts you automatically into the new work plane.



Select the **Project All 3D Geometries** button in the draw toolbar, and click the **Zoom Extents** button in the main toolbar to view the entire projection of the 3D geometry. Open the **Axis and Grid Settings** dialog box from the **Options** menu. On the **Grid** tab, clear the **Auto** check box and enter 0.05 in both the **X-spacing** and **Y-spacing** edit fields. Press **OK**.

Make sure that **SNAP** and **SOLID** are selected in the **Status** bar. Create a solid triangle by selecting the **Draw Line** button and clicking at (0.15,0.75), (0.2,0.75), and (0.2, 0.8) using the left mouse button. Right-click anywhere in the axes to create the solid triangular object.

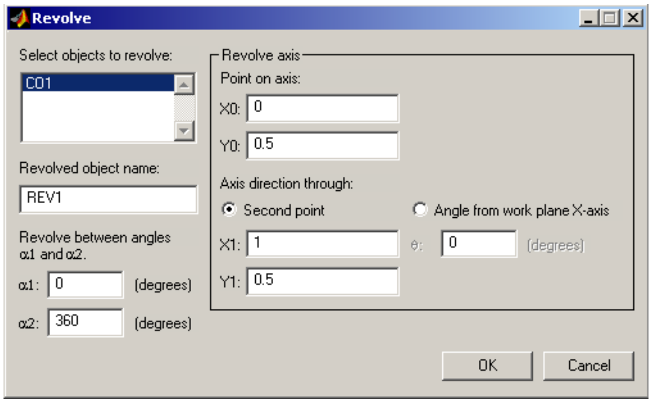

FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]

```
Adding curve segment.
Decomposed added curves/points.
Coerced selected object(s) to solid.
```

(-0.3, 1.05)

File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 2
CO1
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

Now rotate this triangular object by selecting **Revolve…** from the **Draw** menu. In the resulting **Revolve** dialog box, choose the axis about which you want to rotate the object, and enter the angles between which it should rotate.


Revolve

Select objects to revolve: CO1

Revolved object name: REV1

Revolve between angles $\alpha1$ and $\alpha2$.

$\alpha1$: 0 (degrees)

$\alpha2$: 360 (degrees)

Revolve axis

Point on axis:

X0: 0

Y0: 0.5

Axis direction through:

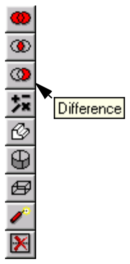
[✓ Second point] Angle from work plane X-axis

X1: 1

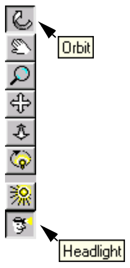
Y1: 0.5

θ: 0 (degrees)

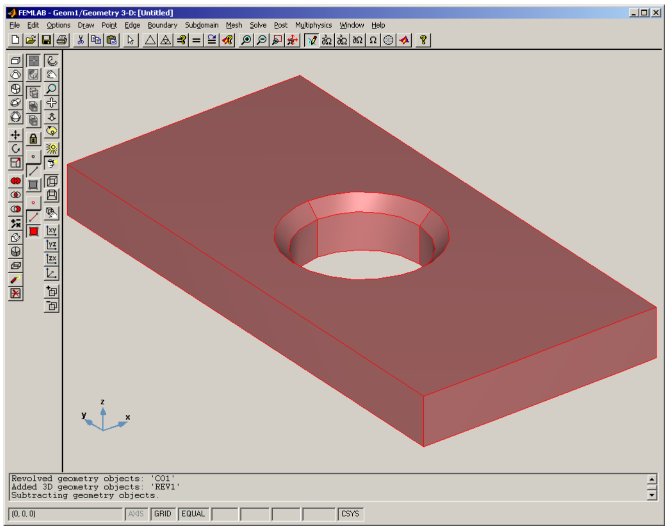
Select the composite triangle object `CO1`. Specify a point on the axis of rotation by entering 0 and 0.5 in the **X0** and **Y0** fields, respectively. Specify the axis direction by selecting the **Second point** radio button and entering 1 and 0.5 in the **X1** and **Y1** fields, respectively. Use the default angles to revolve between 0 and 360 degrees, and press the **OK** button. The resulting revolved 3D object is selected in the 3D view.



Now let us drill the hole through the solid block by creating a composite solid object. First choose **Select All** from the **Edit** menu or use the shortcut **Ctrl-a**. Press the draw toolbar button for difference, which creates the hole. The difference toolbar button subtracts objects with small volumes from the one with the largest volume.

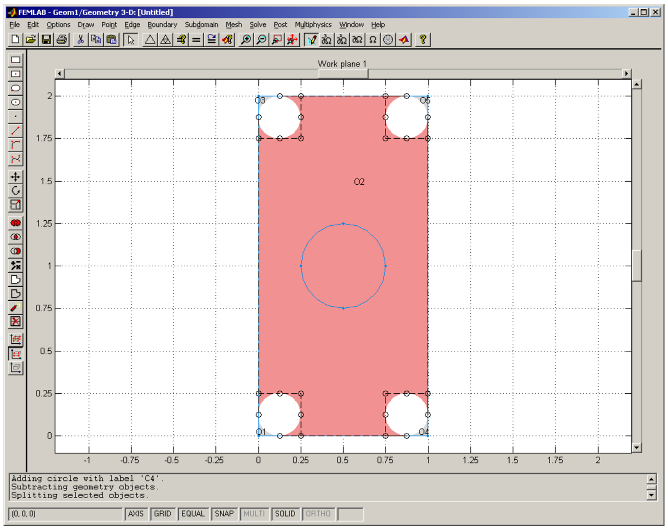


To get a better view of the 3D geometry use the **Headlight** button and the **Orbit** button on the orbit/pan/zoom toolbar. Turn the axis on or off by double-clicking on **AXIS** in the **Status** bar.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Revolved geometry objects: 'CO1'
Added 3D geometry objects: 'REV1'
Subtracting geometry objects.
```

To round off the edges, re-enter the first work plane by selecting **Work plane 1** from the **Draw** menu. Next, select the old circle object, `C1`, and press the **Delete** key or choose **Clear** from the **Edit** menu. Draw a rectangle from (0,0) to (1,2), covering the entire projection of the 3D geometry. Double-click **MULTI** on the **Status** bar to be able to draw multiple objects without the need to reselect the draw tool. Choose **Draw Ellipse** from the drawing toolbar, and then draw four circles with radii of 0.125, using the right mouse button, aligned to the corners of the rectangular geometry as shown below. Then select all objects by pressing **Ctrl-a**, and press the **Difference** toolbar button in the draw toolbar.

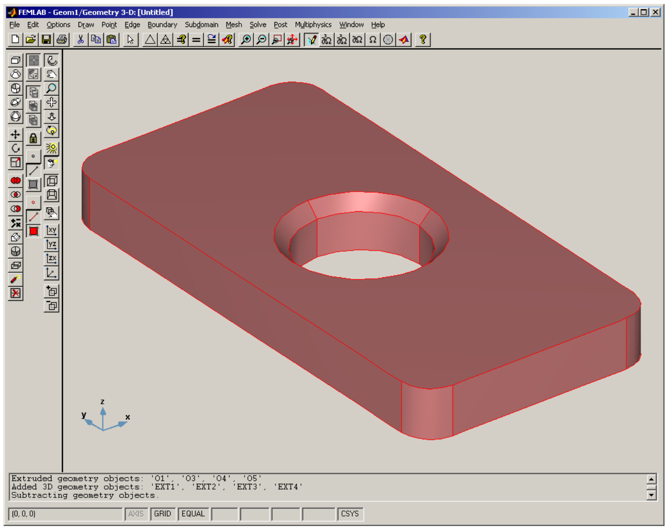

FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 1
O1
O2
O3
O4
O5
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

```
Adding circle with label 'C4'.
Subtracting geometry objects.
Splitting selected objects.
```

(0, 0, 0)

Use the **Split Object** toolbar button to split the composite solid object into five objects. To remove the largest middle object, click on it with the mouse and press delete, or choose **Clear** from the **Edit** menu. Now extrude the remaining objects a distance of 0.2.

In the 3D view, select all objects once again and press the **Difference** toolbar button to get the final geometry object.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Extruded geometry objects: 'O1', 'O3', 'O4', 'O5'
Added 3D geometry objects: 'EXT1', 'EXT2', 'EXT3', 'EXT4'
Subtracting geometry objects.
```

Using the Programming Language

This section demonstrates the use of Boolean operations (union, intersection, and difference) on solid objects in the programming language.

The following examples show how to use the FEMLAB functions to create composite solids starting from primitive solids, repeating the work you just completed within the graphical interface. Some users might opt to postpone a reading of this section because you generally perform geometry modeling from the graphical interface. However, using the programming language for creating geometry objects can be useful, for instance, when you need a sequence of commands to compute exact coordinates of geometry objects or, as in this case, to access advanced functionality not yet implemented in the graphical interface. It is then easy to import geometry objects you create on the MATLAB prompt into the graphical interface for further manipulation.

To create a geometry similar to the one in the previous example, first create a solid rectangle with corners at (0,0) and (1,2).

```
r = rect2(0,1,0,2);
```

Then use the fillet utility to take off its edges.

```
rf = fillet(r,'radii',0.125);
```

Create a circle object with radius 0.25, and form a triangle using three curve objects.

```
c1 = circ2(0.5,1,0.25);
t1 = curve2([0.15 0.2],[0.75 0.75]);
t2 = curve2([0.2 0.2],[0.75 0.8]);
t3 = curve2([0.2 0.15],[0.8 0.75]);
```

Create a solid triangle by a call to `geomcoerce`.

```
t = geomcoerce('solid',{t1,t2,t3});
```

The function `geomcoerce` analyzes the specified geometric objects and coerces the resulting geometric object into the specified class.

Extrude the rectangle along the *z*-axis from 0 to 0.2, and extrude circle `c1` from 0 to 0.2.

```
blk = extrude(rf,0.2);
hole = extrude(c1,0.2);
```

Create the chamfer by revolving the triangle object, `t`, 360 degrees about the center of the hole. To do so, first define a work plane going through the center of the hole, then define an axis of rotation.

```
wrkpln = [[0;1;0],[0;1;1],[1;1;0]];
ax = [[0;0.5],[1;0]];
chamf = revolve(t,'angles',[0,2*pi],...
  'revaxis',ax,'wrkpln',wrkpln);
chamf = solid3(chamf);
```

Use the resulting 3D objects to create the composite solid object by subtracting the chamfer and hole from the filleted block using the Boolean operations "+" and "-".

```
plate = blk-(chamf+hole);
```

Plot the final geometry using the `geomplot` method. The last line adds light to the plot.

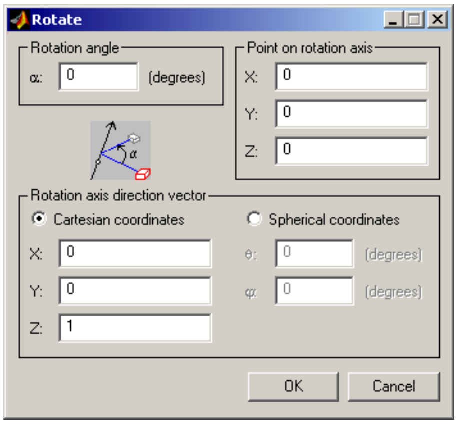
```
geomplot(plate)
axis equal
camlight
```

EDITING SOLIDS IN THE GUI

It is not possible to move 3D objects by clicking and dragging them using the mouse as when working with 2D objects. Instead, you edit a geometry object's intrinsic properties by choosing **Object Properties** from the **Draw** menu or by double-clicking on it using the right mouse button. Note that you must use the right button when double-clicking due to the ambiguity of selection in 3D, but this choice implies that the selected object is locked. See "Object Selection Methods in 3D" on page 1-95 for information about selection in 3D. The only objects that FEMLAB allows you to edit are primitive objects; you cannot edit any object created using a Boolean operation or a work plane in this way.

Choosing **Object Properties** opens a primitive's dialog box, which is the same as the one that appears when you first created it. Further, you can open it only when exactly one primitive is selected. The contents of the dialog box differ depending on which type of primitive you're working on. You can edit object-specific information such as its name, dimensions and orientation in 3D space. To rename a non-primitive object, use the **Create Composite Object** dialog box.

You can move, scale, or rotate a 3D object with the **Move**, **Scale**, and **Rotate** toolbar buttons or by selecting **Linear Transformation** on the **Draw** menu. In the **Rotate** dialog box, you specify an angle of rotation plus the axis of rotation. You define the axis of rotation with a point on the axis plus a direction vector specified in either Cartesian or spherical coordinates.


Rotate

Rotation angle
α: 0 (degrees)

Point on rotation axis
X: 0
Y: 0
Z: 0

Rotation axis direction vector
[✓ Cartesian coordinates] Spherical coordinates
X: 0
Y: 0
Z: 1
θ: 0 (degrees)
φ: 0 (degrees)

SUBDOMAINS AND THE SPLIT OPERATION

The union operator partitions the resulting solid into subdomains defined by the internal borders. These subdomains can, for example, specify different material properties. If you do not want to create borders, deselect the check box **Keep internal borders** in the **Create Composite Object** dialog box.

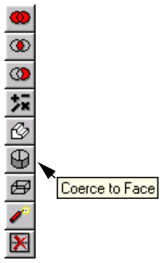
It is possible to split a solid with multiple subdomains into a collection of minimal solids that each consists of only one domain. To do so in the graphical interface, press the **Split** button on the draw toolbar. The same process from the MATLAB prompt is

```
ss = split(s);
```

which splits a solid object `s` into a number of solid objects contained in the cell array `ss`.

FACE OBJECTS

When embedding a 2D solid, or extruding or revolving a curve object, you create a *3D face object*. A composite face object consists of a collection of face objects. You can create it by selecting **Create Composite Object...** from the **Draw** menu or by coercing a 3D solid into a face object by pressing the **Coerce to Face** toolbar button.



BOUNDARY MODELING

When using boundary modeling, you explicitly define a composite solid object's boundary by drawing each of its face segments. When modeling 3D geometries there are actually two possible levels of boundary modeling because you can create 3D boundaries using 2D solid modeling or 2D boundary modeling.

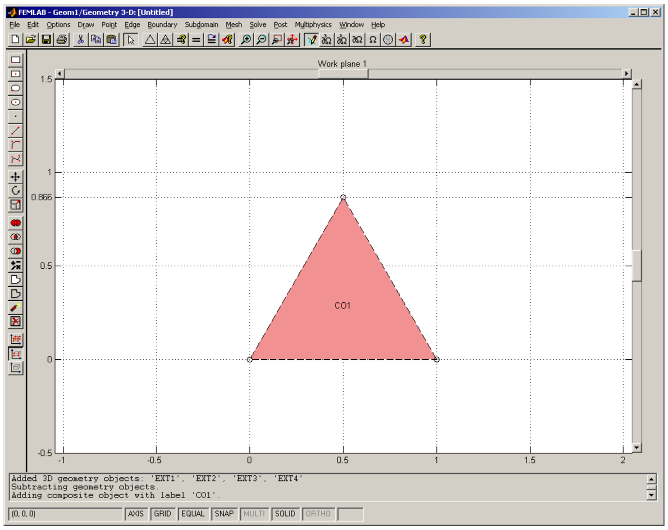
The following example shows how to create a composite solid tetrahedron with boundary modeling.

Using the Graphical User Interface

Open the Model Navigator, select the **3D** radio button, choose **Geometry Only** and press **OK**. Enter the default work plane, **Work plane 1**, from the lower part of the **Draw** menu. By default this work plane is the *xy*-plane.

Using boundary modeling, create a solid triangle with equal sides. For this task, open the **Axis and Grid Settings** dialog box from the **Options** menu, uncheck **Auto** on the **Grid** page, set the *x*- and *y*-spacing to 0.5, and add an extra *y*-grid line at `sqrt(3)/2`. Check that **SOLID** and **SNAP** are enabled on the **Status** bar.

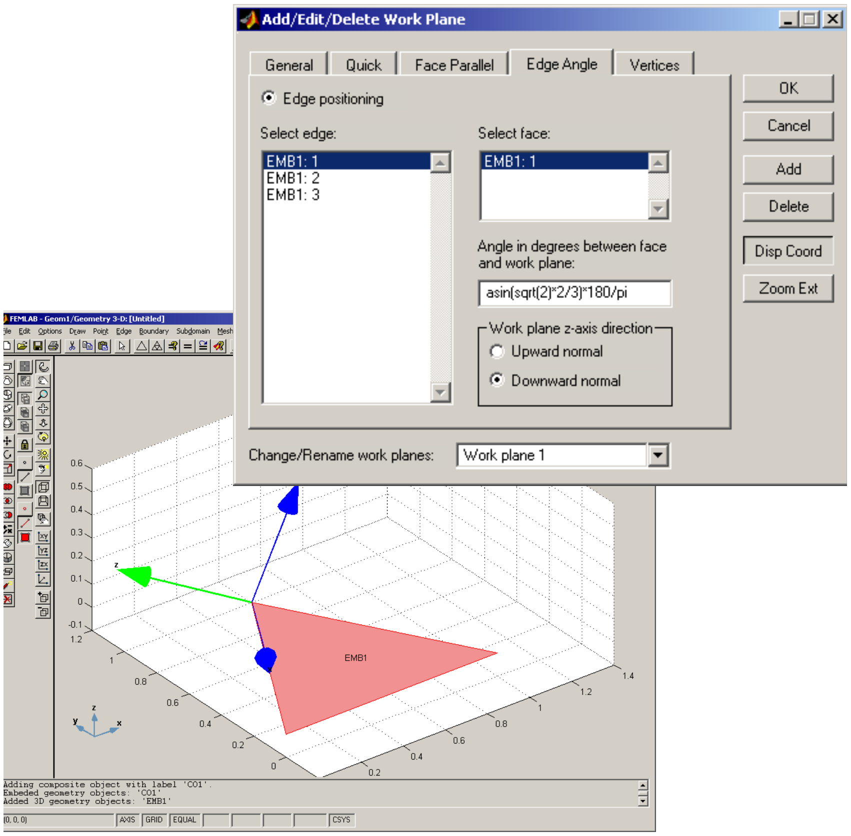
Select **Draw Line** from the toolbar, then click with the left mouse button at (0,0), (1,0) and $(0.5,(\sqrt{3})/2)$. Click anywhere in the axes using the right mouse button to complete the triangle and solidify it.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Work plane 1
1.5
1
0.866
0.5
0
-0.5
-1
-0.5
0
0.5
1
1.5
2
CO1
(0, 0, 0)
AXIS GRID EQUAL SNAP MULTI SOLID ORTHO

```
Added 3D geometry objects: 'EXT1', 'EXT2', 'EXT3', 'EXT4'
Subtracting geometry objects.
Adding composite object with label 'CO1'.
```

Open the **Embed** dialog box from the **Draw** menu and select the triangle before pressing **OK**.

To create the next face, open the **Add/Edit/Delete Work Plane** dialog box and select the **Edge Angle** page. Check the **Edge positioning** radio button and select the first edge in the list box. Select the face in the list box to the right, then enter the expression `asin(sqrt(2)*2/3)*180/pi` in the edit field for the angle between the face and the work plane. Change the *z*-axis direction to **Downward normal** and press the **Disp Coord** button to see the new definition of Work plane 1 before pressing **OK** and entering the plane.



(●) Edge positioning

Select edge: EMB1: 1 (selected), EMB1: 2, EMB1: 3

Select face: EMB1: 1 (selected)

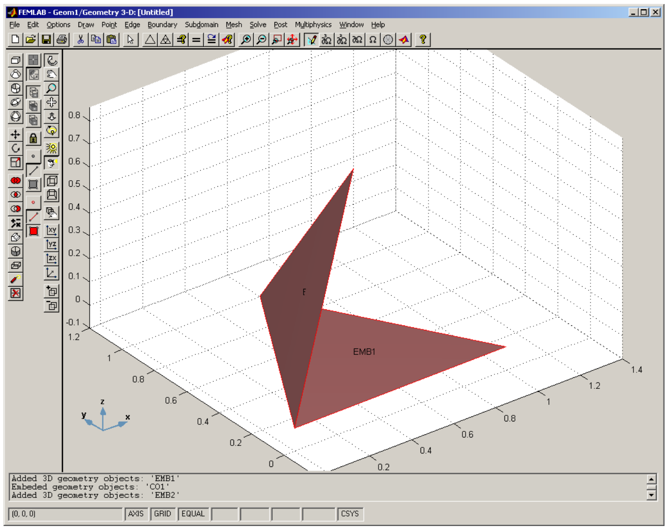
Angle in degrees between face and work plane: asin(sqrt(2)*2/3)*180/pi

Work plane z-axis direction: ( ) Upward normal [✓ (●) Downward normal]

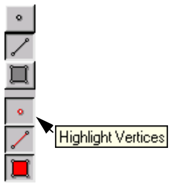
Change/Rename work planes: Work plane 1

```
Adding composite object with label 'CO1'.
Embeded geometry objects: 'CO1'
Added 3D geometry objects: 'EMB1'
```

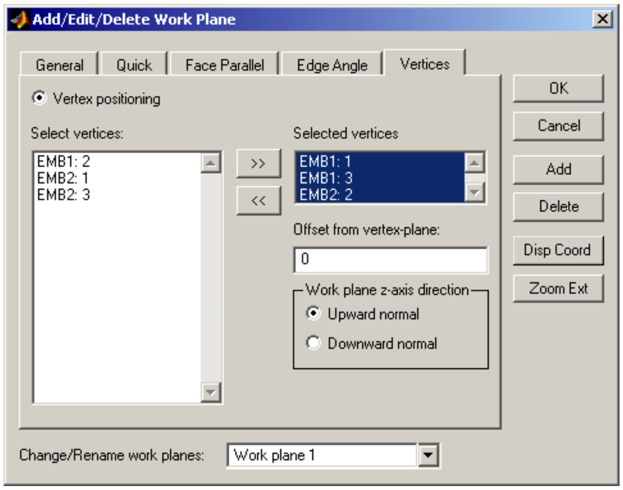
Once you are in that local coordinate system, embed the triangle once again without making any changes.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
EMB1
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Added 3D geometry objects: 'EMB1'
Embeded geometry objects: 'CO1'
Added 3D geometry objects: 'EMB2'
```



Select both faces by pressing **Ctrl-a** or clicking the faces while holding the **Shift** key down. Turn on vertex highlighting by pressing the corresponding toolbar button in the select toolbar (if not already selected), and open the **Add/Edit/Delete Work Plane** dialog again. Now use **Vertex positioning**, and in the list box to the left select vertices 1 and 3 from triangle EMB1 and vertex 2 from triangle EMB2. Press the **Add** button to move the selected vertices to the right list box. Choose upward normal direction and confirm the orientation of the work plane using the **Disp Coord** button.



[✓ Vertex positioning]

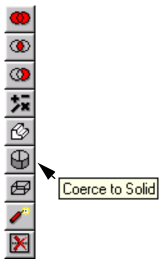
Select vertices:
EMB1: 2
EMB2: 1
EMB2: 3

Selected vertices
EMB1: 1
EMB1: 3
EMB2: 2

Offset from vertex-plane: 0

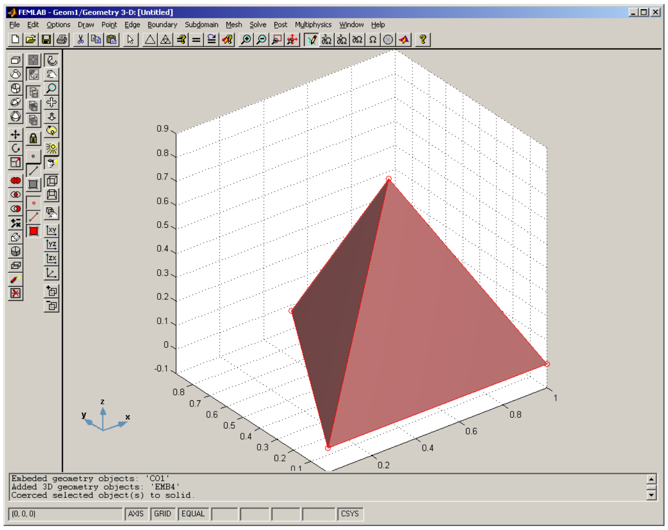
Work plane z-axis direction
[✓ Upward normal] Downward normal

Change/Rename work planes: Work plane 1



Press **OK** and embed the triangle again from the work plane. At this point you've created three of the four faces. To create the last one, make sure to select all existing faces before opening the **Add/Edit/Delete Work Plane** dialog box one more time. Remove the vertices from the right list box on the **Vertices** page by selecting them and pressing the **Remove** button. Select vertex 3 from triangle `EMB1`, vertex 1 from triangle `EMB2`, and vertex 2 from triangle `EMB3`, and then add them to the list box on the right-hand side. Press the **Disp Coord** button to verify that the work plane is correctly oriented before pressing **OK**. By embedding the triangle a last time, you create the final face object.

To create a 3D solid object from these face objects, select all the faces by pressing **Ctrl-a** or by choosing **Select All** from the **Edit** menu. Press the **Coerce to Solid** button in the draw toolbar. FEMLAB automatically names the final solid object `CO1`.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Embeded geometry objects: 'CO1'
Added 3D geometry objects: 'EMB4'
Coerced selected object(s) to solid.
```

Using the Programming Language

This section demonstrates how to create the solid from the previous example using boundary modeling and FEMLAB functions. First create the three boundary curve objects, forming the triangle using the method `curve2`.

```
c1 = curve2([0 0.5],[0 sqrt(3)/2]);
c2 = curve2([0.5 1],[sqrt(3)/2 0]);
c3 = curve2([1 0],[0 0]);
```

For more information on the various geometry objects available, see the *Reference Manual*.

Create a solid triangle object `s` of the object type `solid2`.

```
s = geomcoerce('solid',{c1,c2,c3});
```

In FEMLAB, you can perform extrude, revolve or embed operations, using the low-level methods `extrude`, `revolve`, and `embed` directly on 2D objects. Let us look at this in detail.

Begin by defining three constants; they contain the height for the base triangle in the *xy*-plane, the top of the tetrahedron, and the center of the base triangle, respectively:

```
height= [0.5;sqrt(3)/2;0];
top = [0.5;sqrt(1/12);sqrt(2/3)];
center = top.*[1;1;0];
```

Create the 3D face objects by embedding the solid triangle, `s`, using the `embed` function. The second argument to this utility is a 3-by-3 matrix containing three points that define a work plane. The first point is the origin of the work plane. The second and the third point define the work plane *x*- and *y*-axis, respectively.

```
f1 = embed(s,[[0;0;0],[1;0;0],[0;1;0]]);
f2 = embed(s,[[0;0;0],[1;0;0],top]);
f3 = embed(s,[height,[0;0;0],top]);
f4 = embed(s,[[1;0;0],height,top]);
```

Finally, create the solid 3D object of type `solid3` from the `face3` objects by

```
t = geomcoerce('solid',{f1,f2,f3,f4});
geomplot(t)
```

The last line plots the tetrahedron in a figure window.

RATIONAL BÉZIER PATCHES

This section gives an overview of rational Bézier patches, as used in 3D geometry modeling in FEMLAB.

Why Rational Bézier Surfaces?

The reason for using rational Bézier surfaces in 3D is the same as that for using rational Bézier curves in 2D—it is the current industry standard. When you work in the 3D draw mode and create a geometry object with a curved boundary, FEMLAB represents that object in terms of rational Bézier surfaces. Two types of Bézier surfaces are supported: rectangular and triangular surfaces. A rectangular Bézier surface is assigned a mixed degree (m, n), which represents the degree of the surface in terms of two parameters, often named s and t. A triangular Bézier surface is assigned a single degree, m, just as a Bézier curve.

FEMLAB supports rectangular surfaces of mixed degree at most (3,3) and triangular surfaces of degree 1 to represent planar surfaces. Rectangular rational Bézier surfaces, of mixed degree up to (2,2) can represent all the common CAD-surfaces including bilinear patches, cylinders, cones, spheres, ellipsoids, and tori. FEMLAB also includes (3,3)-degree rational Bézier curves to assist in the creation of additional free-form surfaces. Rational Bézier surfaces are closely connected to so called NURBS-surfaces (non-uniform rational basis spline surfaces) in that a NURBS-surface can always be decomposed into a net of rational Bézier surfaces. You can envision this as considering rational Bézier surfaces as the atomic constituents of a NURBS-surface.

Control Vertices and Weights

A rectangular rational Bézier surface of degree (m,n) is completely defined in terms of a control net consisting of $(m+1)\times(n+1)$ control vertices, each assigned an associated positive control weight. The surface always interpolates the four corner points of the control net. A change in the control net's shape, such as by moving a control vertex, produces a corresponding change in the surface's appearance. Its shape, in a sense, always mimics that of the control net. The higher the surface degree the more complicated are the shapes you can create due to the increased number of control vertices. If you increase the weight associated with one of the control vertex points, that action pulls the curve toward the corresponding control vertex.

This simple and intuitive interaction between the control net, with its weights, and the surface makes the rational Bézier surface representation so useful. By using the `face3` command, you can examine how alterations in control vertices and weights affect surface shape. You can import free-form face objects into the GUI and incorporate them with your solid models.

3D Geometry Objects

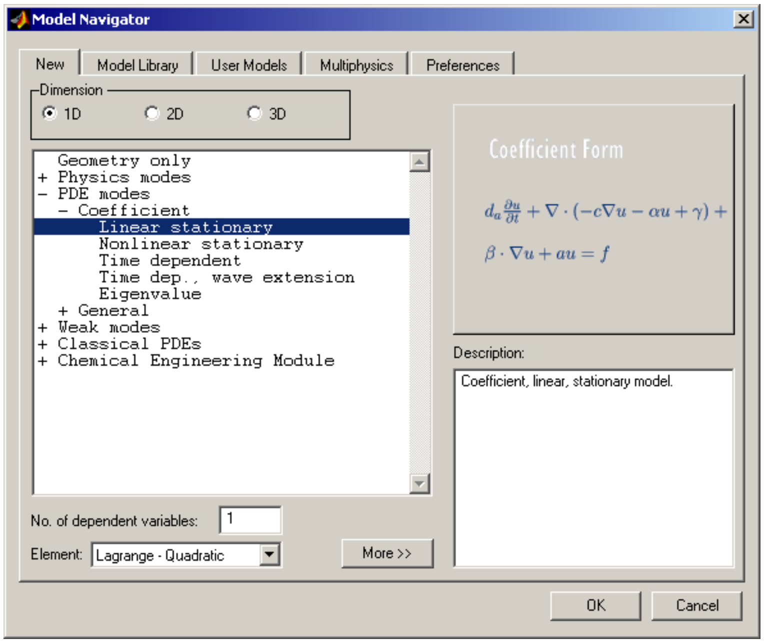
The 3D geometry objects in FEMLAB are formed by a set of trimmed and untrimmed rational Bézier surfaces. A cylinder, for instance, consists of four untrimmed rectangular degree (2,1) patches and two trimmed triangular planar patches. The planar patches are trimmed, so only a circular portion of each planar patch is actually used. When using geometry modeling operations, the Bézier patches are trimmed by the intersection curves between surfaces. By trimming surfaces, patch boundaries can take virtually any shape. The connected trimmed and untrimmed patches of a 3D geometry object are called faces. It is possible for a patch to be divided into any number of faces—curved areas bounded by trimming (intersection) curves.

For more information on rational Bézier patches, see "Rational Bézier Patches" on page 3-13 in the *Reference Manual*.

Creating a 1D Geometry Model

USING THE GRAPHICAL USER INTERFACE

You can also perform geometry modeling in 1D. From the graphical interface, start by pressing the 1D radio button in the **Model Navigator**.



Dimension: [✓ 1D] 2D 3D

```
   Geometry only
+  Physics modes
-  PDE modes
   - Coefficient
       Linear stationary
       Nonlinear stationary
       Time dependent
       Time dep., wave extension
       Eigenvalue
   + General
+  Weak modes
+  Classical PDEs
+  Chemical Engineering Module
```

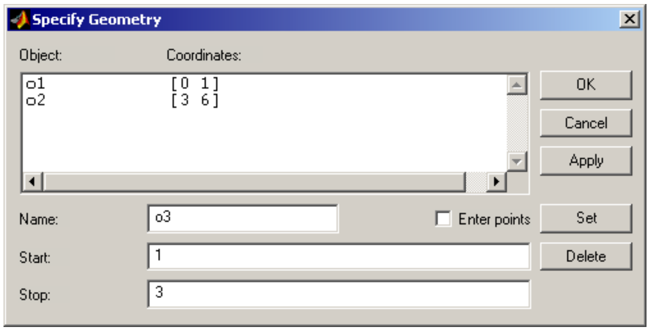
$$d_a\frac{\partial u}{\partial t} + \nabla\cdot(-c\nabla u - \alpha u + \gamma) + \beta\cdot\nabla u + au = f$$

Description: Coefficient, linear, stationary model.

No. of dependent variables: 1

Element: Lagrange - Quadratic

In draw mode you open the **Specify Geometry** dialog box from the **Draw** menu. Here you enter names and coordinates for 1D solid objects and points. The dialog box is similar to the **Add/Edit Constants** dialog box.



```
o1                 [0 1]
o2                 [3 6]
```

Name: o3
Start: 1
Stop: 3

For detailed examples of 1D geometry modeling in the graphical interface, go to the *Model Library* and examine "The KdV Equation and Solitons" on page 2-172 and "The Burgers Equation" on page 2-165.

USING THE PROGRAMMING LANGUAGE

From the MATLAB prompt, you create 1D geometry model using the constructors `solid1`, `point1`, and the function `geomcsg`.

The command

```
s=solid1([0 1 2])
```

creates a 1D solid object with two subdomains.

Entering

```
p=point1(0.5)
```

creates a 1D point object, and

```
g=geomcsg({s},{p})
```

merges the two geometry objects to a 1D analyzed geometry.

See the model "Tubular Reactor" on page 1-378 of this book for an example of 1D geometry modeling using the programming language.

Importing CAD Models

You can import both 2D and 3D CAD models into FEMLAB. In 2D, the DXF file format is used, and in 3D the IGES format is used.

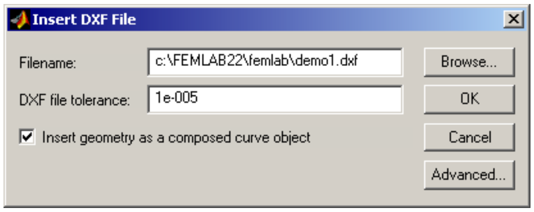
DXF IMPORT

FEMLAB allows you to import 2D CAD drawings via the DXF file format, which is an industry standard format for CAD drawings. To be able to utilize the automatic mesh generation, FEMLAB requires a high degree of accuracy within the CAD drawing. If these requirements are not met, the geometry needs to be updated to meet FEMLAB's requirements. This can be done by applying different "geometry repair" algorithms on the drawing when importing the DXF file into FEMLAB.

Often there is a need for closing small gaps and eliminate short edges in the imported geometry. If small gaps or exceedingly short edges are present, FEMLAB might fail in creating either a valid 2D solid or a mesh of acceptable quality. To remove these short edges and to close small gaps between the end points of curves, the import property DXF tolerance often need to be adjusted. This tolerance specifies the largest distance between the end points of curves allowed in the imported geometry. For the FEMLAB function, `dxfread`, this property is called `tol`. In the GUI you define this property in the **DXF file tolerance** field found in the **Insert DXF file** dialog box. In order to close gaps between end point of curves and curves within the specified tolerance, the `dxfread` property `curvesnap` has to be activated. The corresponding GUI setting can be made in the **Advanced DXF File Properties** dialog box. See "Insert from File" on page 4-21 in the *Reference Manual* for details.

An Example Using the Graphical User Interface

To import the DXF file `demo1.dxf` into the GUI, open the **Insert DXF file** dialog from the **Insert from File** submenu found in the **File** menu. Start by importing the geometry in `demo1.dxf` from the `femlab` directory as a composed curve object using the default settings.

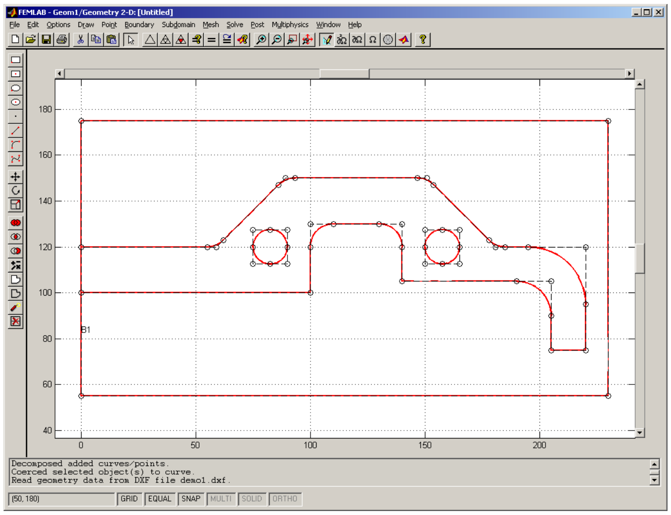

Insert DXF File

Filename: c:\FEMLAB22\femlab\demo1.dxf

DXF file tolerance: 1e-005

☑ Insert geometry as a composed curve object

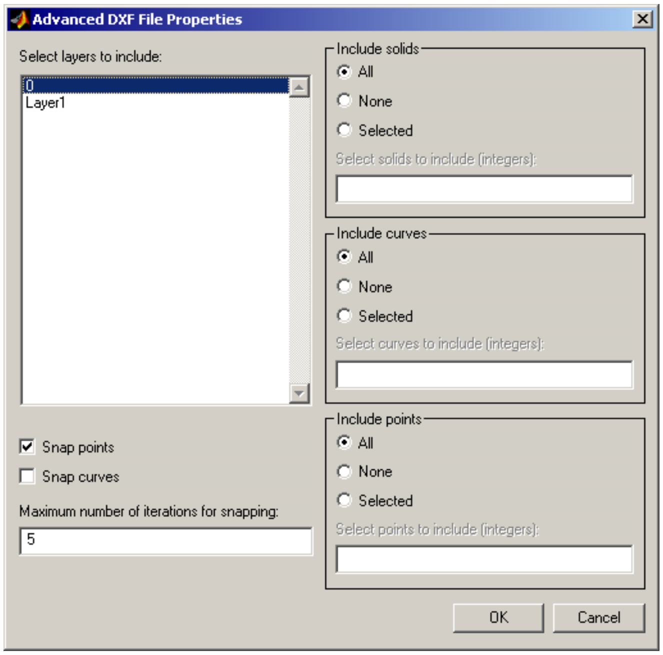
You can use the **Browse...** button to find the file. Pressing **OK** results in the following plot.


FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
B1
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

```
Decomposed added curves/points.
Coerced selected object(s) to curve.
Read geometry data from DXF file demo1.dxf.
```

(50, 180)

The actual geometry in `demo1.dxf` is found in the DXF layer `0`. To import this layer only (excluding the frame lying in the DXF layer named `Layer1`) select layer `0` in the **Advanced DXF File Properties** dialog. First remove the current geometry by choosing **Undo Import DXF File** from the **Edit** menu. Now, open the **Insert DXF File** dialog box again and press the **Advanced...** button to open the **Advanced DXF File Properties** dialog box. Here, select layer `0` and press **OK** in both dialog boxes. This results in a geometry without the bounding box.


Advanced DXF File Properties

Select layers to include:
- 0 (selected)
- Layer1

[✓] Snap points
[ ] Snap curves

Maximum number of iterations for snapping: 5

Include solids: [✓ All] None Selected
Select solids to include (integers):

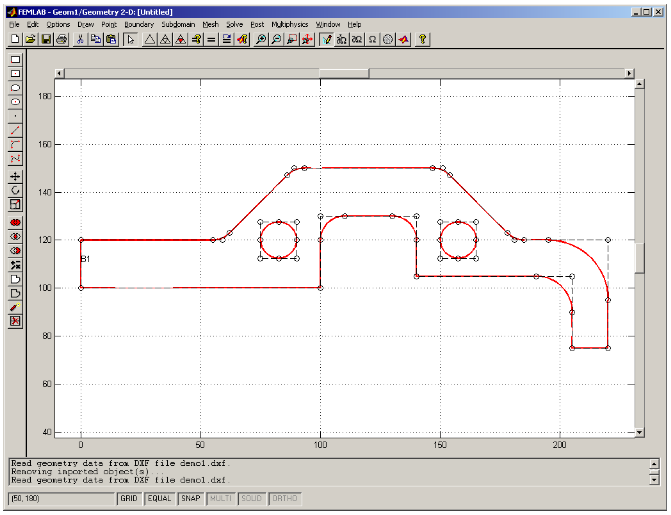
Include curves: [✓ All] None Selected
Select curves to include (integers):

Include points: [✓ All] None Selected
Select points to include (integers):

OK Cancel

Alternatively this can be done by importing both layers but only the curves belonging to layer 0. In `demo1.dxf` the first 22 curves belong to layer `0`. This curve selection is done by checking **Selected** in the **Include curves** frame in the **Advanced DXF File**

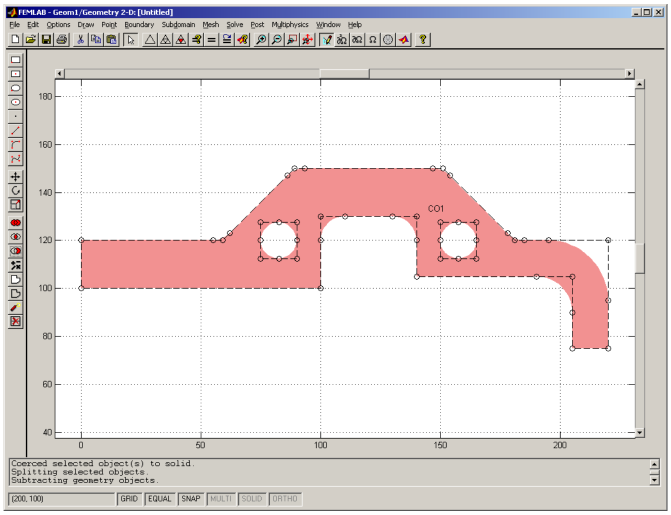
Properties dialog and specifying curves `1:22` in the field **Select curves to include (integers):**.


FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
180
160
140
120
100
80
60
40
0
50
100
150
200
B1
(50, 180)
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

```
Read geometry data from DXF file demo1.dxf.
Removing imported object(s)...
Read geometry data from DXF file demo1.dxf.
```

By solidifying the imported geometry it can be seen that parts of the geometry disappear. This depends on small gaps in the geometry.

These gaps can be closed when importing the DXF file by setting the **DXF file tolerance** in the **Insert DXF file** dialog to `0.001` and checking **Snap curves** in the **Advanced DXF File Properties** dialog. If you clear the previously imported geometry (for example by choosing **Undo Import DXF File**) and import the file once again with these new settings, you can press the **Coerce to Solid** button in the draw toolbar to get a solidified geometry. By pressing the **Split Object** button, you can split the object into its subdomains. Now, pressing the **Difference** button in the draw toolbar removes the smaller geometries from the largest one. The result is the following geometry:


FEMLAB - Geom1/Geometry 2-D: [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
180
160
140
120
100
80
60
40
0
50
100
150
200
GRID EQUAL SNAP MULTI SOLID ORTHO

```
Coerced selected object(s) to solid.
Splitting selected objects.
Subtracting geometry objects.
```

(200, 100)

IGES IMPORT

3D CAD geometric models can be imported to FEMLAB via the IGES format (Initial Graphics Exchange Specification). The IGES format is a neutral standard format for exchanging geometric models between different CAD systems. Geometric models saved in the IGES format do not always pass flawlessly between different CAD systems. This is unavoidable since the IGES format uses the lowest common denominator approach to describe a geometric model which means that information is lost. Furthermore, the IGES format allows a certain geometric model to have different representations, and some representations are less compatible with the receiving system than others.

In FEMLAB you get a printed report of the encountered problems during the translation. This error summary is printed to the message log when you run the GUI and to the MATLAB prompt when you run the FEMLAB function `igesread`. Often there is a need for adjusting the geometry tolerance parameter used when translating the geometric entities in the IGES file. This tolerance specifies the smallest distance between coordinates that is considered as discernible. For the FEMLAB function, this property is called `agtol`. In the GUI this property is defined in the **IGES file tolerance** field found in the **Insert IGES file** dialog box.

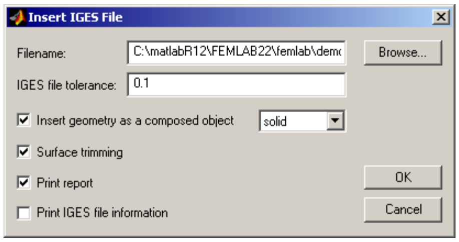
The surfaces that define the boundary of a solid in an IGES file are represented by untrimmed surfaces and their corresponding trimming curves. If FEMLAB encounters problems trying to trim some surface, the message 'Unable to trim surface.' is printed. Then you are advised to try to import the IGES file without applying the surface trimming or to adjust the geometry tolerance parameter. The property that determines if the surfaces in an IGES file are trimmed is called `surftrim` for the FEMLAB function, and in the **Insert IGES file** dialog box it is defined in the **Surface trimming** check box.

If you just want to have a quick view of the geometry in the IGES file you should import the surfaces in the IGES file without composing and coercing them into a solid. Working from the GUI this is done by unchecking the check box **Insert geometry as a composed object** in the **Insert IGES file** dialog box, and from the MATLAB prompt by setting the property `coercion` to `off`. You can also compose your imported surfaces into one face object without coercing it into a solid by checking the **Insert geometry as a composed object** check box and then choosing **face** in the popup menu. On the MATLAB prompt this is done by setting the `coercion` property to `face`.

For more information on the **Insert IGES file** dialog box, see "Insert from File" on page 4-21 in the *Reference Manual* for details.

Using the Graphical User Interface

We will now study two examples of IGES file import working from the GUI. To import the IGES file `demo1.igs` into the GUI, open the **Insert IGES file** dialog box from the **Insert from File** submenu found in the **File** menu. Start by importing the geometric model in `demo1.igs` from the `femlab` directory using the default settings.


Insert IGES File

Filename: C:\matlabR12\FEMLAB22\femlab\demo

IGES file tolerance: 0.1

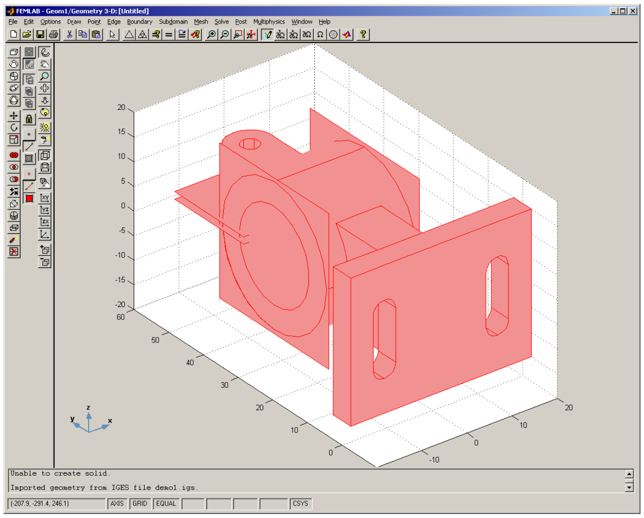
☑ Insert geometry as a composed object: solid

☑ Surface trimming

☑ Print report

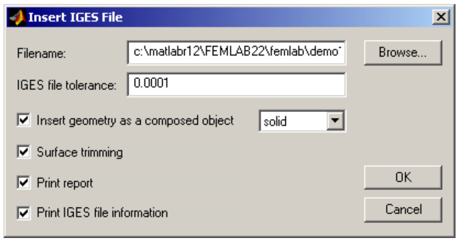
☐ Print IGES file information

You can use the **Browse...** button to find the file. Click **OK** to get the following plot.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
20
15
10
5
0
-5
-10
-15
-20
60
50
40
30
20
10
0
-10
0
10
20
z
y
x
(-207.9, -291.4, 246.1)
AXIS
GRID
EQUAL
CSYS

```
Unable to create solid.
Imported geometry from IGES file demo1.igs.
```

By studying the error summary in the message log, it can be seen that FEMLAB encountered some problems while importing this IGES file. These problems can be avoided by adjusting the **IGES file tolerance** in the **Insert IGES file** dialog box. First, remove the current geometry by selecting **Undo Import IGES File** from the **Edit** menu. Now, open the **Insert IGES file** dialog box again and set the **IGES file tolerance** field to `0.0001`. By checking the **Print IGES file information** check box you get information on the geometry tolerance and the entities defining the geometric model in the IGES file printed to the message log.



Insert IGES File

Filename: c:\matlabr12\FEMLAB22\femlab\demo`

IGES file tolerance: 0.0001

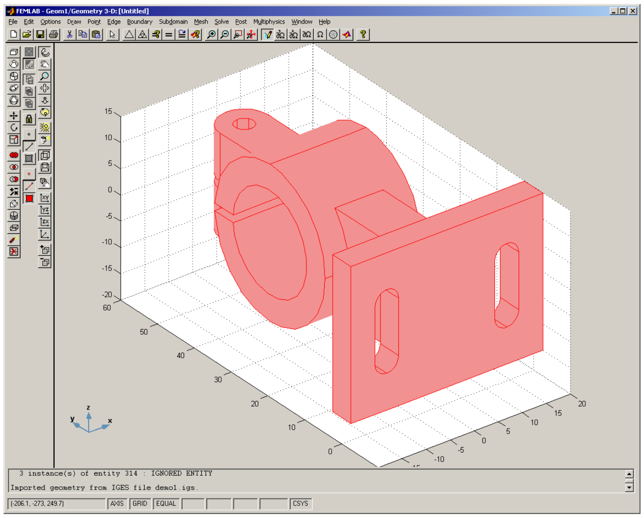
☑ Insert geometry as a composed object — solid

☑ Surface trimming

☑ Print report

☑ Print IGES file information

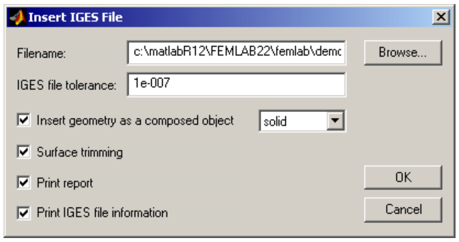
By clicking **OK**, the file is imported smoothly using the new settings.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
15
10
5
0
-5
-10
-15
-20
60
50
40
30
20
10
0
-10
-5
0
5
10
15
20
z
y
x
AXIS
GRID
EQUAL
CSYS

```
  3 instance(s) of entity 314 : IGNORED ENTITY
Imported geometry from IGES file demo1.igs.
```

(-206.1, -273, 249.7)

As mentioned earlier, a single geometric model can have many IGES format representations. Several different combinations of IGES entities can be used to represent a geometric model in the IGES format, and each combination is more or less compatible with the receiving system. You can try to import another IGES file storing the same geometric model, by using another combination of IGES entities to represent the geometry. Remove the previously imported geometry and import the IGES file `demo2.igs`, found in the `femlab` directory as well, with the **Print IGES file information** check box checked.



Filename: c:\matlabR12\FEMLAB22\femlab\demo

IGES file tolerance: 1e-007

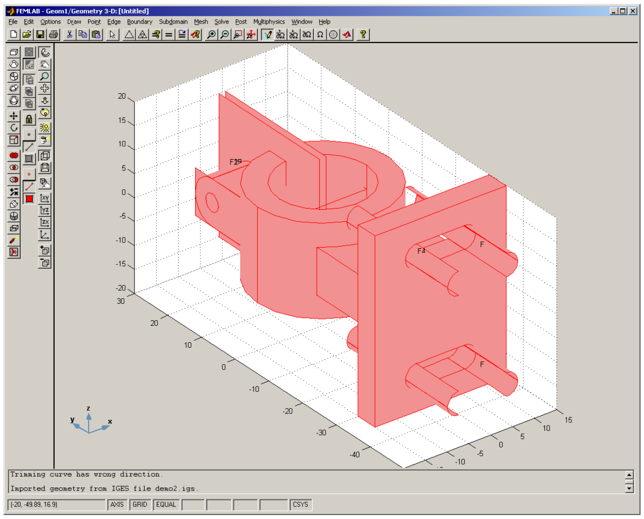
[✓] Insert geometry as a composed object — solid

[✓] Surface trimming

[✓] Print report

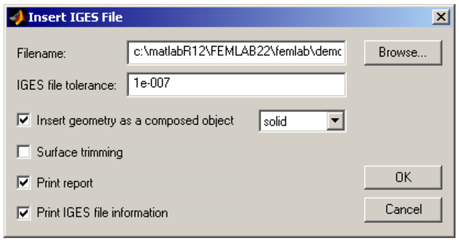
[✓] Print IGES file information

By studying the entity summary in the message log, you can see that the file `demo2.igs` does not use the same set of IGES entities.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
AXIS GRID EQUAL CSYS

```
Trimming curve has wrong direction.
Imported geometry from IGES file demo2.igs.
```

In the error summary, it can be seen that the translator encountered problems when trying to trim some surfaces in the file `demo2.igs`. The surface trimming can be turned off, when importing an IGES file, by unchecking the **Surface trimming** check box. Remove the previously imported geometry, and try to import `demo2.igs` with the surface trimming turned off.


Insert IGES File

Filename: c:\matlabR12\FEMLAB22\femlab\demc

IGES file tolerance: 1e-007

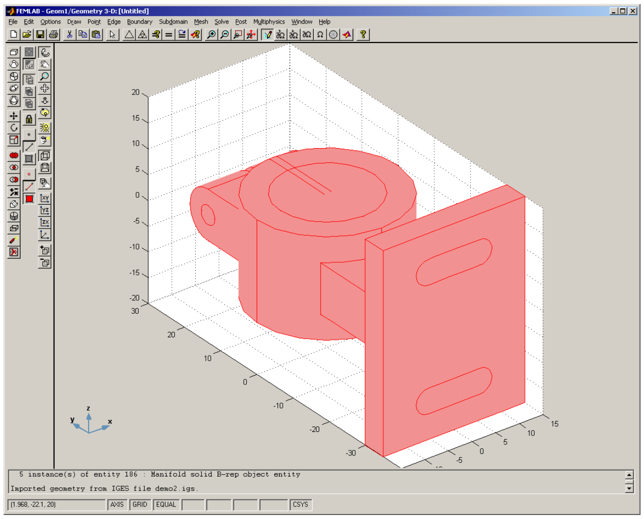
☑ Insert geometry as a composed object: solid

☐ Surface trimming

☑ Print report

☑ Print IGES file information

By not trimming the surfaces in the IGES file, the translator succeeded in creating a valid solid. There is a need for removing some subdomains, however, to obtain the desired solid object.

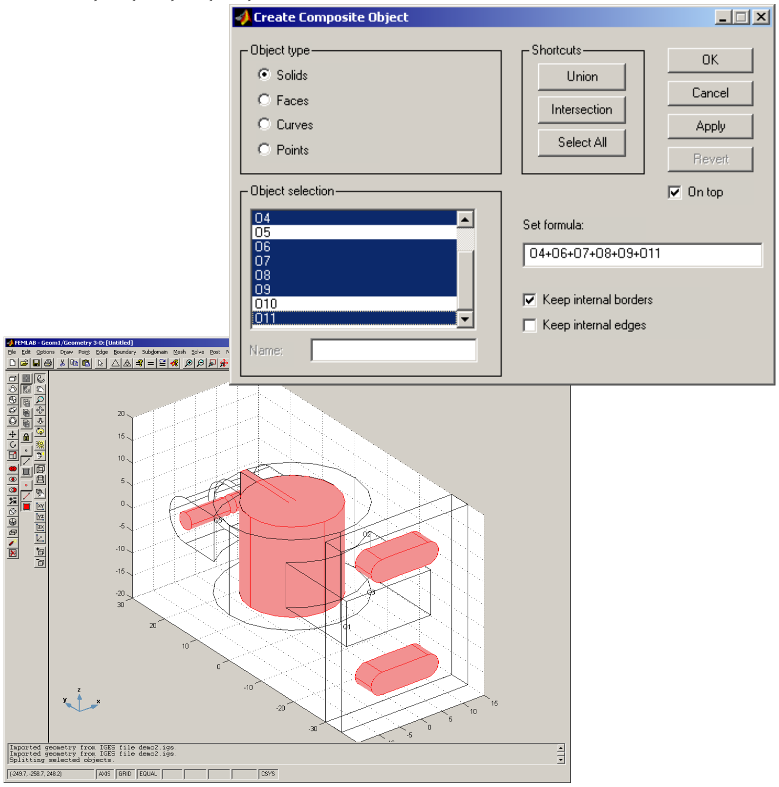

FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
AXIS GRID EQUAL CSYS

```
 5 instance(s) of entity 186 : Manifold solid B-rep object entity
Imported geometry from IGES file demo2.igs.
```

(1.968, -22.1, 20)

Split the solid object into its constituents by pressing the **Split Object** toolbar button. Press the **Create Composite Object** button on the draw toolbar to open the **Create**

Composite Object dialog box where the solids to be deleted can be selected. Select the solids **04**, **06**, **07**, **08**, **09**, and **011**.



Create Composite Object

Object type: (● Solids) ○ Faces ○ Curves ○ Points

Object selection: **04** 05 **06** **07** **08** **09** 010 **011**

Name:

Set formula: 04+06+07+08+09+011

☑ Keep internal borders

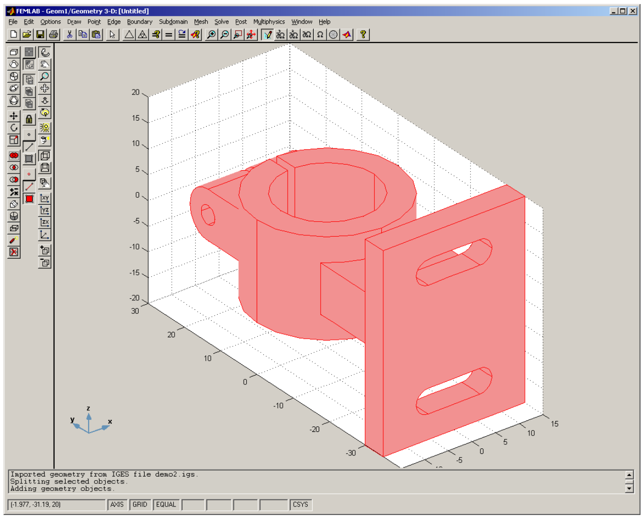
☐ Keep internal edges

☑ On top

```
Imported geometry from IGES file demo2.igs.
Imported geometry from IGES file demo2.igs.
Splitting selected objects.
```

Shortcuts
Union
Intersection
Select All
OK
Cancel
Apply
Revert
FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post
AXIS GRID EQUAL CSYS

Click **Cancel** to close the dialog box and select **Clear** from the **Edit** menu (or press the **Ctrl-r** key) to remove the selected solids. Select all objects by selecting **Select All** from the **Edit** menu (or by pressing the **Ctrl-a** key) and press the **Union** toolbar button to create the final solid object.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
20
15
10
5
0
-5
-10
-15
-20
30
20
10
0
-10
-20
-30
15
10
5
0
-5
z
y
x
AXIS
GRID
EQUAL
CSYS

```
Imported geometry from IGES file demo2.igs.
Splitting selected objects.
Adding geometry objects.
```

(-1.977, -31.19, 20)

Image and MRI Import

Magnetic resonance imaging (MRI) is an imaging technique used primarily in medical settings to produce high quality images of the inside of the human body. The more general name for the analytical technique is NMR (Nuclear Magnetic Resonance spectroscopy). MRI data is typically represented as a sequence of 2D images.

MRI import consists of two steps. First the images need to be imported, and the corresponding geometry objects in FEMLAB generated. After that, 3D geometry objects can be created using a lofting technique.

IMPORT OF IMAGES

Images of the formats JPEG, TIFF, GIF, BMP, PNG, HDF, PCX, XWD, ICO, and CUR can be converted into 2D or 3D geometry objects, working from the MATLAB prompt. To read graphic file format images use the MATLAB function `imread`. This function converts images on graphic file format to a matrix format in the MATLAB work space. Other useful functions are `imwrite`, for saving images, `image`, and `imagesc`, for displaying images on matrix format, and `imfinfo`, for obtaining information about graphics file format images. See the MATLAB documentation for more information on the MATLAB functions available for image analysis. Note that the Image Processing Toolbox contains additional functionality for advanced image processing.

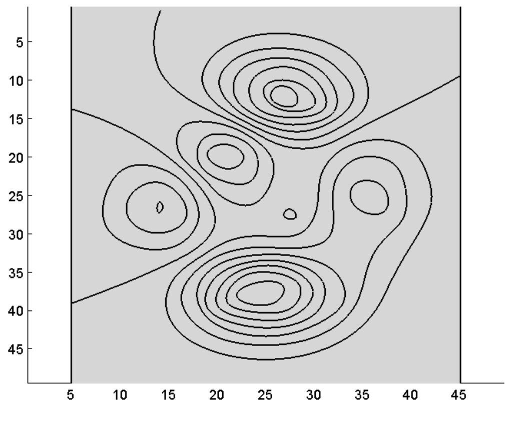
In FEMLAB, a 2D curve object can be created from an image using the function `flim2curve`, and a 3D solid object from a sequence of images using the functions `flim2curve` and `loft`. For more information, see below in this section and the *Function Reference* entries `flim2curve` and `loft`.

A 2D curve describing the contour curves of an image is created by calling `flim2curve` with appropriate property values. The function `flim2curve` utilizes the MATLAB function `imcontour` for detecting contours in images. The following example shows an image on matrix format, generated from MATLAB's sample function `peaks`, that is converted to a 2D curve object.

```
p = (peaks+7)*5;
figure
image(p)
v = axis;
c = flim2curve(p,{[],[5:5:75]});
g = geomcsg({rect2(5,45,0,50)},{c})
s = solid2(g);
```

Visualize the geometry object with the following commands.

```
figure
geomplot(s,'pointmode','off','sublabels','on');
axis(v)
axis ij
```



You can continue working on this model either from the MATLAB prompt or from the GUI. To insert this geometry model into the GUI, select **Insert from Workspace** and then **Geometry Object(s)...** from the **File** menu in the FEMLAB GUI. Once you have done this, you can use the **Split** and **Union** buttons for creating the geometry to be used in the simulation.

If the original image is noisy, the resulting geometry object may contain too many edge segments to be practical to work with. The function `flim2curve` provides basic functionality for filtering data by supporting the same input argument properties as `flmesh2spline` (see the *Function Reference* entries for these functions). Also, MATLAB provides functionality for filtering images before converting them to FEMLAB geometry objects. See the MATLAB documentation on the functions `filter2` and `conv2` for more information on how to do this.

For more information on reducing noise in point data, see “Spline Interpolation for Creating Geometry Objects” on page 1-225.

If the processed image is large, the computation time for the MRI import functions can be long. To reduce the computation time one can often subsample the image before processing it. That is, reduce the number of pixels in the image. The most simple subsampling method is to include every second pixel in the x and y direction, respectively. More advanced subsampling methods preprocess the image by low-pass filtering (also referred to as softening or blurring) before reducing the number of pixels. This often gives a higher quality of the subsampled image due to the fact that low-pass filtering reduces the amount of high frequency components in the subsampled image (due to rasterization effects).

MRI IMPORT

When creating a 3D geometry object from a sequence of images, for example, MRI data, the work flows as follows.

- Load MRI data to MATLAB. The 3D MRI data is typically represented as a sequence of 2D images.
- Create a set of 2D geometry sections from the images using the `flim2curve` command.
- Use the `loft` command to create an interpolating 3D surface between the 2D geometry sections.
- Define your FEM problem on the resulting geometry.

The example below shows how to create a FEMLAB 3D solid object from MRI-data.

Start by loading MRI-data from MATLAB.

```
load mri
```

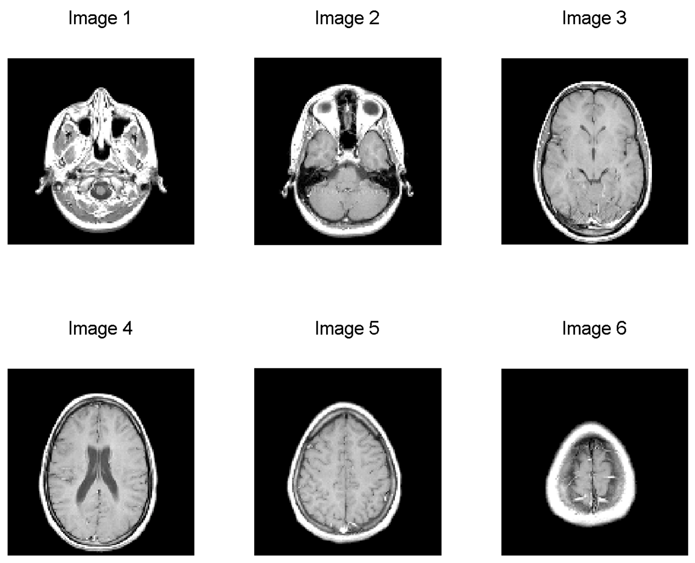
This loads a sequence of images stored in multi-dimensional array called `mri`. It also loads a color map named `map`, for illustrating the images.

Create an index vector indicating which sections to include.

```
i = [1 6 12 17 22 27];
```

Visualize the sections.

```
figure
for k = 1:6
  subplot(2,3,k)
  image(D(:,:,1,i(k)))
  title(sprintf('Image %d',k))
  axis equal,axis off
end
colormap(map)
```



Before creating curve objects from the images, the `threshold` and `keepfrac` data need to be defined. The threshold value is used for creating a binary image, where all parts of the image with a lower value than the threshold value is turned white. This binary image is used for creating contour curve as in the function `imcontour`. When the threshold value is set to 1, the contour curves are created on the boundaries of the regions of the images that are completely black. The `keepfrac` property in `flim2curve` is used for determining how many points in the contours created from the images should be used for creating a curve object. This value need to be tuned for all images so that every created curve object consists of the same number of segments. This is crucial for creating a 3D geometry model of the cross-section images.

```
th = [1 1 1 1 1 1];
kf = [0.11 0.10 0.112 0.115 0.129 0.165];
```

Loop over sections and save curve-data in the cell-array `c`.

```
for k = 1:6
  [c{k},r] = ...
  flim2curve(D(:,:,1,i(k)),{th(k),[]},'KeepFrac',kf(k));
end
```

Convert 2D curve objects to 2D solid objects.

```
for k = 1:6
  c{k} = solid2(c{k});
end
```

Now, each of these solid objects could be used for creating a 2D mesh, and for solving 2D problems. Alternatively, the objects could be given specific names and then imported into FEMLAB. For example, create

```
s1=c{1}
```

and import the solid object `s1` into the GUI using **Insert from Workspace, Geometry Objects** from the **File** menu. You can then continue modeling from the GUI, creating a mesh, specifying the boundary conditions and material parameters, solving, and visualizing the results.

Instead, in this example, the 2D solids are seen as 2D cross-sections of a 3D geometry, as follows.

Create a mapping table that maps edges between sections. (See `loft` help text or *Function Reference* entry.)

```
el = {18 18 18 19 19 18};
```

This means that edge 18 of section 1 is mapped on edge 18 of section 2. The rest of the edge mappings between sections 1 and 2 are then automatically derived from this. The mappings between sections 2,3; 3,4; 4,5; and 5,6, then are: 18-18, 18-19, 19-19 and 19-18, respectively.

Create 3D positioning data for the sections.

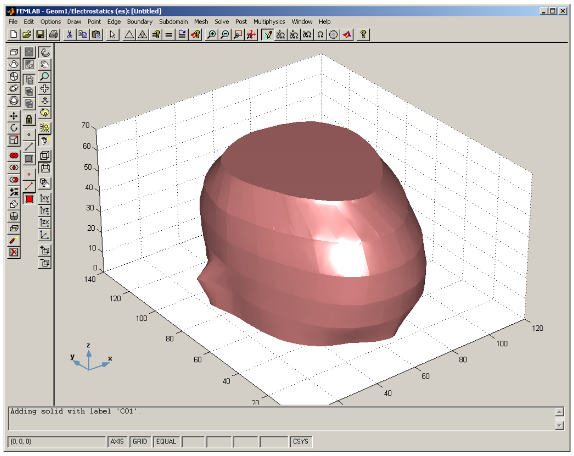
```
dvr = {repmat(12.5,1,5),repmat(0,2,6),repmat(0,1,6)};
```

Loft between the sections using cubic lofting (the default lofting method), with the lofting weights explicitly given.

```
lg = loft(c,'loftedge',el,'loftsecpos',dvr,...
  'loftweights',repmat(0.1,2,5));
```

This geometry object can now be imported to the GUI.

- In the **Model Navigator**, select the **3D**, **Geometry Only** application mode.
- Select **Insert from Workspace** from the **File** menu. To import a geometry object, select **Geometry object(s)...** and then type the name `lg`.
- Click the **Headlight** and **Perspective Projection** toolbar buttons.
- Click the **Render Edges** and **Highlight Edges** toolbar buttons, to avoid visualizing the edges in the geometry.


FEMLAB - Geom1/Electrostatics (es): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
70
60
50
40
30
20
10
0
140
120
100
80
60
40
20
120
100
80
60
40
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Adding solid with label 'CO1'.
```

You can now continue and use this geometry as part of a simulation: create a mesh, solve, and post process the solution. Note that to be able to mesh this on a computer with limited memory resources, you need to select a coarse mesh setting.

Spline Interpolation for Creating Geometry Objects

To create a curve object from a set of data points, the function `geomspline` can be used. It creates C^1 or C^2 continuous splines for the interpolation between the points.

The use of `geomspline` is illustrated below by creating a geometry object with splines created from irregularly distributed points on a circle.

First create the circle data, sorted by the angle, and remove some of the points.

```
phi = 0:0.2:2*pi;
phi([1 3 6 7 10 20 21 25 28 32]) = [];
p = [cos(phi);sin(phi)];
```

Add some noise to the data points.

```
randn('state',17)
p = p+0.01*randn(size(p));
```

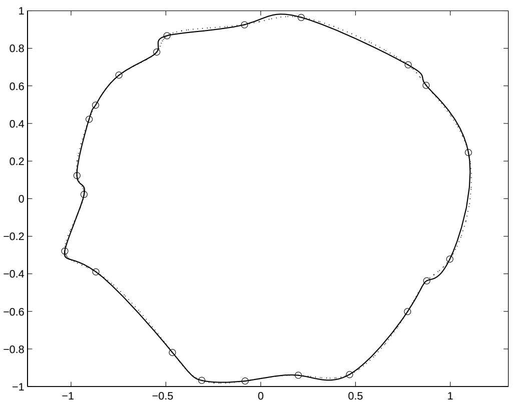
Different global parametrization techniques can be used in `geomspline`. The global parametrization is a parameter that varies from 0 in the first interpolated point to 1 in the last interpolated point. For a closed curve these two points coincide. For details on the different parametrization techniques, see the reference entry on `geomspline` in the *Function Reference*.

First use a uniform parametrization technique. This is the most straightforward technique, but it does not always work well on irregularly distributed data.

```
plot(p(1,:),p(2,:),'ro')
c = geomspline(p,'splinemethod','uniform','closed','on')
hold on
geomplot(c,'pointmode','off')
```

Then interpolate using centripetal parametrization and note the difference.

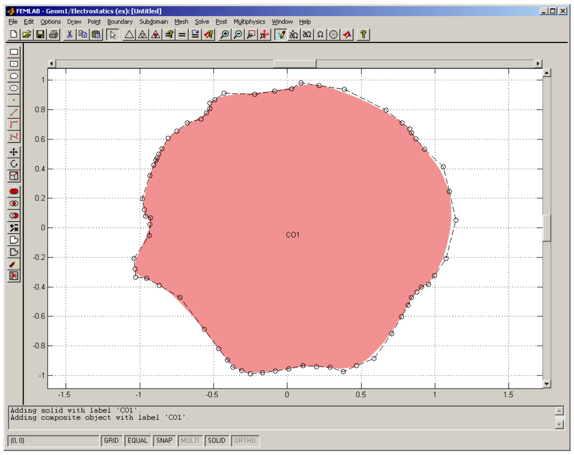
```
c = geomspline(p,'splinemethod','centripetal','closed','on')
hold on
geomplot(c,'pointmode','off','edgecolor','b')
axis equal
```



For creating a solid geometry from this, on which you can set up a FEM problem, simply enter

```
s = solid2(c);
```

To use this geometry for modeling in the GUI select **Insert from Workspace**, and **Geometry Object(s)...** from the **File** menu in the FEMLAB GUI. This is done when the **Draw Mode** is active, either when setting up a 2D problem or when a work plane has been defined when setting up a 3D problem. Once this is done you can use the ordinary geometry operations to continue working on the model.


FEMLAB - Geom1/Electrostatics (es): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.5
-1
-0.5
0
0.5
1
1.5
(0, 0)
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

```
Adding solid with label 'CO1'.
Adding composite object with label 'CO1'.
```

Occasionally, point data can be too dense or contain too much noise to be practical to work with. The functions `flcontour2curve` and `flmesh2spline` contain basic functionality to reduce the number of edge segments in a point set. To understand the MATLAB contour data format, see the MATLAB functions `imcontour`, `contour`, and `contourc`.

Note that you can also use `geomspline` for creating 3D curve objects. Using the data created in the previous example, define the z coordinates to create the following 3D curve object.

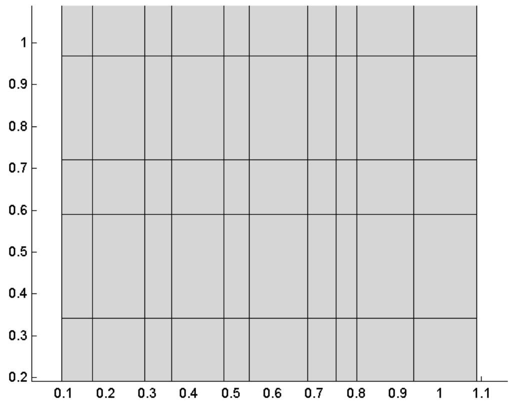
```
p = [p;linspace(0,3,22)];
c = geomspline(p,'splinemethod','centripetal','closed','off')
```

A 3D face object or a 2D solid object can be created from a 3D point set or a 2D point set respectively using the function `geomsurf`; see the *Function Reference* entry `geomsurf` for more information.

2D SOLID OBJECTS

A solid object with rectangular subdomains can be created from a rectangular point set using `geomsurf`. In the following example a solid object with non-uniformly distributed subdomains is generated from a non-uniform grid.

```
[x,y] = meshgrid(0:0.1:1,0:0.2:1);
rand('state',1);
x = x+repmat((0.1*rand(1,size(x,2))),size(x,1),1);
rand('state',1);
y= y+repmat((0.2*rand(size(y,1),1)),1,size(y,2));
s = geomsurf(x,y);
figure
geomplot(s,'pointmode','off')
axis equal
```



This can be used for defining a domain with a non-uniform distribution of some material parameter.

You can import this geometry object into the GUI.

- From the **File** menu, select **Insert from Work Space**, **Geometry Object(s)**.
- Type the name of the object, `s`, in the edit field of the dialog box opened, and press **OK**.

3D FACE OBJECTS

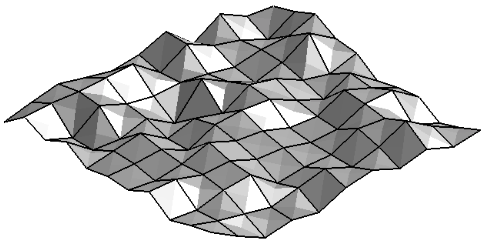
By using the function `geomsurf`, a surface can be created from height data defined on a grid by bilinear interpolation. The surface created is identical in shape to the surface created with the MATLAB command `surf`. The following example illustrates how to generate a surface from simulated or measured data and how to insert the generated surface as the top surface of a solid block.

The height data is defined as deviations, with a maximum deviation of 0.03, from a reference plane situated at *z* equal to 0.7.

```
dev_max = 0.03;
ref_level = 0.7;
[x,y] = meshgrid(0:0.1:1,0:0.1:1);
randn('state',1);
z = ref_level+dev_max*randn(size(x));
```

Create a surface from the height data.

```
f = geomsurf(x,y,z);
```



Create a solid block whose top surface lies above the generated surface.

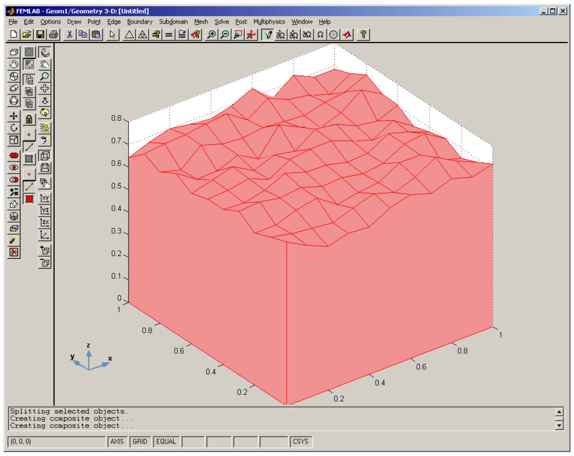
```
b = block3(1,1,ref_level+3*dev_max);
```

The geometry objects `f` and `b` can now be inserted in the GUI.

- Start the FEMLAB GUI. In the **Model Navigator**, select the **3D**, **Geometry Only** application mode.

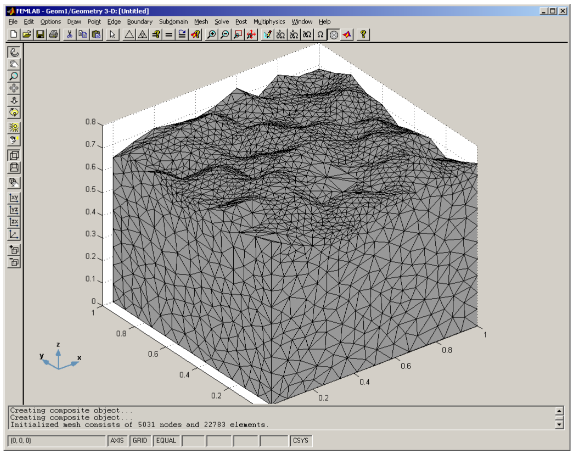
- On the **File** menu, select **Insert from Work Space**, **Geometry Object(s)**.
- In the dialog box edit field, type a space-separated list of object names: `b f`, and press **OK**.
- In the GUI, select both objects and click the **Coerce to Solid** toolbar button. This will merge the objects, one face object and one solid object, into a single solid object.
- Now, split the object by pressing the **Split Object** toolbar button.
- Delete the object on the top.

This is the resulting geometry object.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Splitting selected objects.
Creating composite object...
Creating composite object...
```

- You can now create a mesh by pressing the **Initialize Mesh** button, and add any application mode to solve a problem on this geometry.


FEMLAB - Geom1/Geometry 3-D: [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0
1 0.8 0.6 0.4 0.2
0.2 0.4 0.6 0.8 1
z y x
(0, 0, 0)
AXIS GRID EQUAL CSYS

```
Creating composite object...
Creating composite object...
Initialized mesh consists of 5031 nodes and 22783 elements.
```

From the prompt, the same operations can be made as follows.

Intersect the solid block with the generated surface and split the resulting solid object.

```
s = geomcoerce('solid',{b,f});
ss = split(s);
```

Mesh the solid object with the interpolated surface as top surface.

```
m = meshinit(ss{1});
figure
meshplot(m)
axis equal
cameramenu
```

This will visualize the mesh in a MATLAB figure window.

If the original data set is noisy, the resulting geometry object may contain too many faces to be practical to work with. MATLAB provides functionality for filtering data sets before converting them to FEMLAB geometry objects. See the MATLAB documentation on the functions `filter2` and `conv2` for more information on how to do this.

Insertion of Points in Geometry Models

In some modeling situations it is desirable to insert a series of points into a geometry model created in FEMLAB. This is typically the case in any of the following two situations:

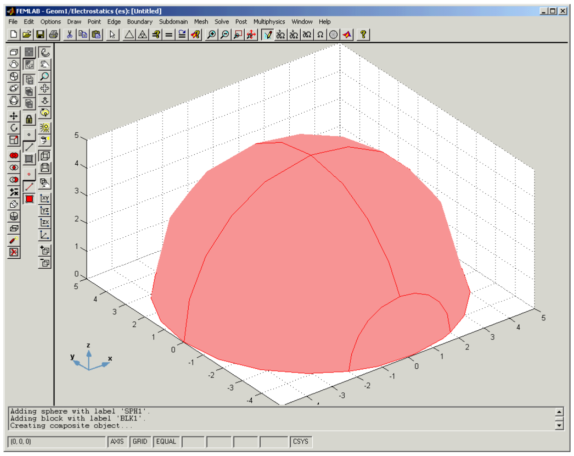
- You want to set point constraints or point sources, see "Example: Poisson's Equation with a Point Source" on page 3-85.
- You want to retrieve values of the solution for certain coordinates. Inserting explicit points at the coordinate values is not necessary but is sometimes convenient. This will ensure that the mesh have node points in the explicitly added points.

Points are most easily inserted into a FEMLAB geometry model using the Programming Language. You can also load the coordinate values from a file, see "Importing Data from Text File" on page 1-294.

The following model illustrates how points can be inserted into a geometry.

- Select **3D**, **Electrostatics** application mode from the **Model Navigator**.
- Draw a sphere with radius 5 centered at the origin, and a block with side lengths of (10,10,5) centered at the point (0,0,-2.5).

- Open the **Create Composite Object** dialog box, select both objects, type the formula `SPH1-BLK1` in the **Set formula** field and press **OK**. This results in the hemisphere object CO1.


FEMLAB - Geom1/Electrostatics (es): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help

```
Adding sphere with label 'SPH1'.
Adding block with label 'BLK1'.
Creating composite object...
```

(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

- Export the geometry object by selecting **Export to Workspace**, **Geometry as Objects...** from the **File** menu. Then press **OK** to use the same name for the object in the MATLAB workspace as in the FEMLAB GUI.
- Run the following script to create the vertices equally distributed on constant radius in a spherical coordinate system.

```
% Radius as parameter
r = 5;
% Create a grid of points using spherical coordinates
% on the surface
m = 11; n = 11;
[theta,phi] = meshgrid(...
    linspace(0,pi*(1-1/m),m),...
    linspace(0,2*pi*(1-1/n),n));

% Evaluate coordinate values from parameters
x = r*sin(theta).*cos(phi);
y = r*sin(theta).*sin(phi);
z = r*cos(theta);

% Create cell array of POINT3 objects.
v = {};
for k = 1:prod(size(x))
  v{k} = point3(x(k),y(k),z(k));
end
```

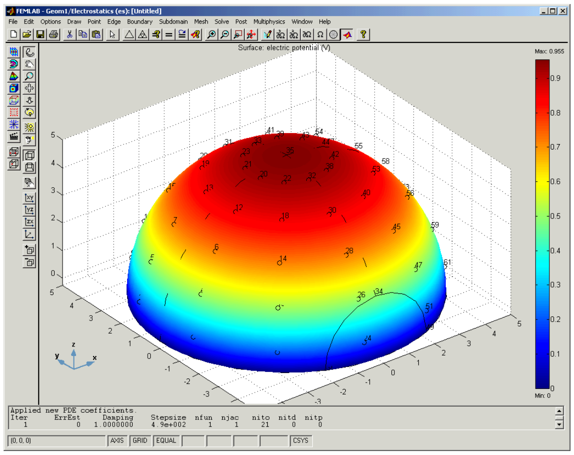
```
% Insert the point object into original object and coerce to solid
CO1 = geomcoerce('solid',{CO1 v{:}});
```

- The new object can now be inserted into the FEMLAB GUI. First remove the original object named `CO1`. Then select **Insert from Workspace** from the **File** menu. Choose **Geometry as Object(s)...** and give the name `CO1`. Press **OK**.

  You can also insert the points directly into the GUI and draw the rest of the geometry using the graphical CAD tools. To do this you create a composite point object instead of a solid object by entering the MATLAB command

  ```
  CO1 = geomcoerce('point',v);
  ```

- To solve a simple problem, select **Ground** as boundary condition for boundary 1. Then press the **Solve Problem** button.


FEMLAB - Geom1/Electrostatics (es): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Surface: electric potential (V)
Max: 0.955
Min: 0
AXIS GRID EQUAL CSYS

```
Applied new PDE coefficients.
Iter      ErrEst      Damping     Stepsize  nfun  njac    nito  nitd  nitp
   1           0    1.0000000    4.9e+002     1     1      21     0     0
```

- To study the values of the solution in the vertices you created above, select **Export FEM structure as 'fem'** from the **File** menu. Then give the following command at the MATLAB prompt.

  ```
  V1 = posteval(fem,'V','edim',0,'dl',[3 5 7 13 21 35]);
  ```

The variable `V1` is a cell array of length 1. The values of the vertices can be visually inspected by selecting **Surface** plot in the GUI and giving `V` as the **Surface expression** in the **Plot Parameters** dialog box. Also select **Show Vertex Labels** from the **Options**, **Labels** menu.

Parameterization of Curves and Faces

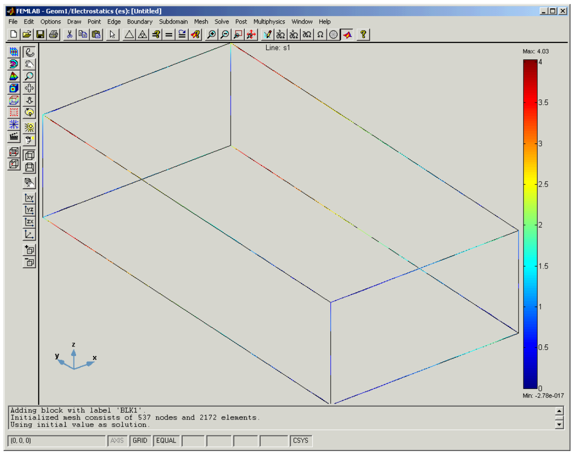
The curves and surfaces of a geometry object may have several different mathematical representations. In FEMLAB, rational parameterizations are used in the representation of curves and surfaces. For a detailed description of these representations, see “Geometry Modeling” on page 3-10 in the *Reference Manual*. This means that there is a parameter `s1` for curves and two parameters, `s1` and `s2`, for faces that can be used when setting up a model or in the postprocessing functions available in FEMLAB. More precisely, this means that for each value of the curve parameter `s1` there is a unique point lying on the curve. For faces, each pair of values `(s1,s2)` corresponds to a unique point that lies on the face.

The faces and edges used in the FEMLAB geometry representation are trimmed surfaces and curves respectively. This means that there is a well-defined boundary in the parameter domain that determines the valid values of `s1` and `s2`. In 2D, the possible values of `s1` will in almost all cases be the interval [0,1], but in 3D the parameter domain is more complicated. For surfaces as well as for curves.

The best way of determining the parameterization is to plot the parameter values. You can do this for a block as described below.

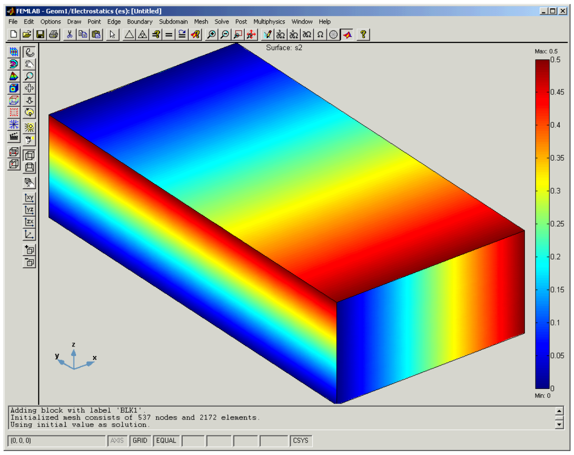
- Start any 3D application mode in the FEMLAB GUI.
- Draw a **Block** object with the side lengths 2, 4, and 1 respectively.
- To see how the faces and curves are parameterized before setting up the problem, select **Get Initial Value** from the **Post** menu. This enables you to use the plotting tools without solving the problem first.
- Deselect **Slice** plot and select **Line** plot instead. At the **Line** page, set the **Line expression** to `s1`. The different geometric variables that are available for plotting

are listed in the tables in“Geometric Variables” on page 1-352 and “Geometric Variables” on page 3-56 in the *Reference Manual*.


FEMLAB - Geom1/Electrostatics (es): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Line: s1
Max: 4.03
4
3.5
3
2.5
2
1.5
1
0.5
0
Min: -2.78e-017
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Adding block with label 'BLK1'.
Initialized mesh consists of 537 nodes and 2172 elements.
Using initial value as solution.
```

- To visualize the surface parameters on the faces, select **Surface** plot and give either `s1` or `s2` as **Surface expression**. Make sure to deselect **Smooth** before plotting, to avoid an incorrect smoothing over the edges.


FEMLAB - Geom1/Electrostatics (es): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Surface: s2
Max: 0.5
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Min: 0
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

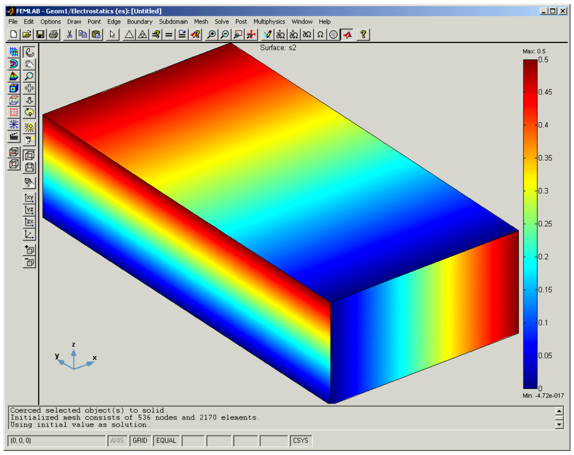
```
Adding block with label 'BLK1'.
Initialized mesh consists of 537 nodes and 2172 elements.
Using initial value as solution.
```

Sometimes you may need to modify the parameterization of one or several surfaces or curves. This is often the case when using periodic boundary conditions in 3D. Periodic boundary conditions means that two boundaries should be regarded as the same when assembling the problem. This means that they must have identical parameter representations.

The best way of accomplishing equal parameterizations is to use the extrude and revolve functionality when creating the geometry models. These functions are naturally used together with periodic boundary conditions, since they are both two forms of connecting several faces that are identical with respect to the parametrization. In some cases, the geometry cannot be created this way. However, there are alternative ways for obtaining matching parameterizations.

If you study the parameterization of the upper and lower face, with respect to the *z*-axis, in the block object above, you can see that the `s2` parameter is increasing along the *y*-axis for one of the faces and decreasing for the other. This means that these two faces cannot be used for setting up periodic boundary conditions. There may also be other reasons than using periodic boundary conditions for wanting to change the parameterization. Matching parameterizations can be accomplished as described below.

- Push the **Draw mode** button and select the object `BLK1`. Press the **Coerce to Face** button followed by the **Split Object** button.
- Delete face object `F4`.
- Copy face `F3` and set the **z-displacement** to 1, and the other two displacements to 0 when pasting the new object.
- Select all objects and press the **Coerce to Solid** button. A new solid object with the desired parameterization has now been created. To see the new parameterization, select **Get Initial Value** from the **Post** menu. Set the surface expression to `s1` or `s2` to see that the parametrization is the same for the two faces.


FEMLAB - Geom1/Electrostatics (es): [Untitled]
File Edit Options Draw Point Edge Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Surface: s2
Max: 0.5
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Min: -4.72e-017
z
y
x
(0, 0, 0)
AXIS
GRID
EQUAL
CSYS

```
Coerced selected object(s) to solid.
Initialized mesh consists of 536 nodes and 2170 elements.
Using initial value as solution.
```

Associative Geometry

Associative geometry is a notion for automatic updating of applied physical properties, such as boundary conditions and PDE coefficients, under geometric transformations. This means that if you have defined the physical properties of your FEM model and return to draw mode to modify the geometric model, the physical properties are updated according to the geometrical modifications as you leave draw mode.

The associative geometry functionality is essentially built upon geometrical mapping information between the domain groups, that is, vertices, edges, boundaries, and subdomains in 3D, in the analyzed geometric object, and the corresponding domain groups in the geometric model available in draw mode.

Note however that this geometrical mapping is not without ambiguities. This means that some decisions have to be made by the associative geometry functionality when mapping the physical properties between the analyzed geometric object, that is, the object on which the physical properties are imposed, and the geometric model available in draw mode. This implies that in some cases, the resulting updated physical properties might not be as desired.