# Your FEMLAB Trial:

# An Introduction to FEMLAB's Multiphysics Modeling Capabilities

# Preface

.

This manual gives you an introduction to modeling in FEMLAB by taking you through the basic steps of the modeling process, from generating the geometry to postprocessing. It also provides you with a list of resources for more in-depth technical information.

Enjoy your modeling!

*COMSOL, Inc.*

*8 New England Executive Park*
*Burlington, MA 01803*
*Phone: 781-273-3322*
*Fax: 781-273-6603*
*www.comsol.com*

# Technical Resources

Throughout your trial period, you have access to the following sources of information: manuals, online tutorials and knowledge base, model library, online command reference and FEMLAB support team. Here is a brief overview of these resources:

### Manuals

Trial users receive the full set of manuals built into the software. To access these manuals, in FEMLAB go to the **Help** menu and select **FEMLAB Help Desk.** We provide the documentation in PDF and HTML format. These manuals include:

1. User's Guide and Introduction manual: This manual demonstrates the basics of how to create a model, how to use the graphical user interface and how to specify a geometry. It also describes application modes available in the core FEMLAB package and much more.
2. Model Library manual: This manual documents the models of the model library. These are problems that have been solved using FEMLAB that we thought might help you understand the range of problems that FEMLAB can model. These models can also be used for inspiration, as starting points for your own modeling.
3. Reference manual: This manual contains information on more advanced features of the software, a description of the various GUI windows, menus and toolbars, as well as a reference library of all FEMLAB commands.
4. Module manuals: The manual for each module describes the application modes of that module. You will also find extensive descriptions of the models of the model library that are based on the module in question.

*When you purchase the software you will also receive a set of printed manuals.*

### Online tutorials

On the webpage:

http://www.comsol.com/showroom/tutorials/

you will find animated tutorials that take you step by step through the set-up of multiphysics models.

### Online Knowledge Base

The webpage:

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

is your first stop for troubleshooting assistance. Search this area of our website if you are experiencing difficulty with an installation, receiving errors at run-time, or seeing unexpected behavior.

Here, we have also collected a large number of FEMLAB models provided by users. You can find tips and tricks as well as unusual solutions to tough modeling situations.

### Model library

The model library has been described above. It can be accessed either through the **New** item of the **File** menu or through the **Examples and Demos** entry of the **Help** menu. Note the **Documentation** button on the **Model Library** tab of the **Model Navigator** which gives you access to the online documentation associated with each model.

### Online command reference

While FEMLAB possesses a graphical user interface, you can also create a FEMLAB model by typing code at the MATLAB command line. These commands are documented in the Reference manual as explained earlier but you can also get information about a command from the MATLAB command line. For instance, to learn about the "postinterp" command, you can type

```
doc postinterp
```

or

```
help postinterp
```

at the MATLAB command line.

### Support team

At any point during your trial, you should feel free to contact our technical staff with your questions. Do not get stuck with a modeling problem! E-mail the support team at support@comsol.com for help! Alternatively, you can call support at 781-273-3322. Your trial is also a great opportunity for you to discover the quality of the support you will receive when you become a customer.

# Introduction

FEMLAB offers a complete modeling environment that allows you to perform all the steps in the modeling process. FEMLAB's graphical user interface includes functions for CAD modeling, import of drawings and images, physics or equation definition, mesh generation, equation solving, visualization and postprocessing. The parametric solver gives you a perfect way of examining a parameterized series of models. The parameter varied typically represents a material property, frequency or reaction rate.

The analysis in engineering design does not end with the computation of voltages, stresses, and velocity fields. Frequently, we have to perform extensive post processing on the computed output in order to arrive at useful engineering parameters like impedance, quality factor, conversion factors, rotation angles etc. In the examples in this primer, you will be able to translate the simulation results into engineering design numbers.

The following example model gives an overview of basic modeling and introduces the concepts of multiphysics in the two-way coupling of heat transport and current conduction in an electronic device. The concepts demonstrated in this example apply to all types of FEMLAB models, be they from chemical engineering, structural mechanics or electromagnetics. To demonstrate the various application modes available in FEMLAB, the heat transfer part of this multiphysics problem is defined first using the "heat transfer" physics mode (a parametric sweep is demonstrated), then the problem is solved using the PDE modes and finally we employ the weak mode feature of FEMLAB. For the PDE modes, both coefficient form and general form are illustrated.

As a last step, we perform a simple parametric study from the Graphical User Interface and demonstrate the use of the scripting language.

# Your First Multiphysics Model

Let us start by defining and solving our first model in FEMLAB. The model treats current conduction and heat generation in an aluminum film deposited on a silicon substrate. The purpose of the model is to investigate the temperature in the device after the current is applied. This model is based on a submission by M.S. Cohen at Aegis Semiconductors.

The model exemplifies the important steps in a typical modeling process. It also gives a quick overrview of some of the features available for advanced modeling in FEMLAB's graphical user interface. We will address the following important FEMLAB features:

- The use of predefined physics.
- The definition of a multiphysics problem.
- The expression feature to define physical properties that depend on the solution itself.
- The extraction of design numbers using the postprocessing tools.
- The parameter solver feature that provides fast and efficient parameter screening as well as smooth convergence for highly nonlinear problems.

In addition to the FEMLAB features, several modeling methods are exemplified in the exercise, among them:

- Reduction of the dimension of parts of the problem from three to two dimensions.
- Control of the obtained results to estimate the validity of the solutions.

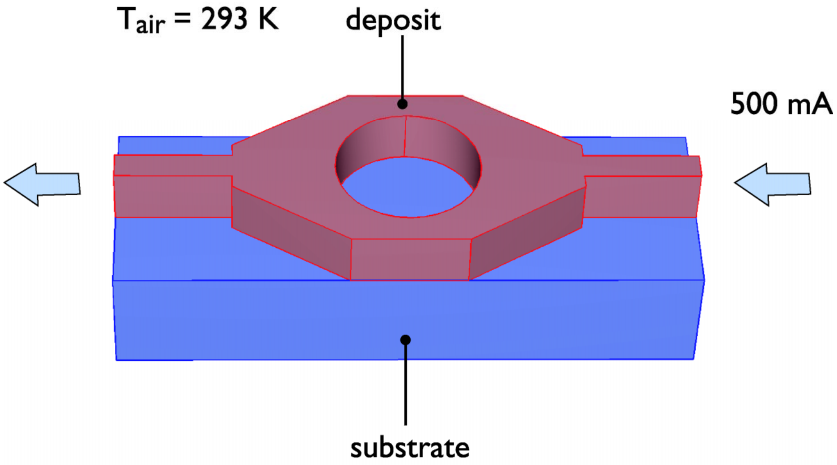
Apart from these highlights, we will go through the typical modeling steps, which include:

1. Definition of the geometry
2. Definition of the physics in the volume and at the boundaries
3. Meshing
4. Solving
5. Postprocessing
6. Parametric studies

Although in a typical modeling session the analyst would usually try to use the predefined physics modes as much as possible, we will also illustrate with this model how the PDE modes and weak modes of FEMLAB can be used to solve the heat transfer problem.

These applications modes are extremely useful when it comes to analyzing non-standard problems for which no physics modes are available.

## *General Description of the Physical Problem and Study*



A film of aluminum is deposited on a silicon substrate. The aluminum is about 1 µm thick, with the pattern shown above. The width of the pattern is about 1000 µm, with a hole in the center about 250 µm in diameter. A constant current is passed through the deposit starting at time t = 0.

We wish to study the temperature distribution in the aluminum pattern as a function of time. We also want to see the electrical power distribution as well as the total resistance as a function of time.

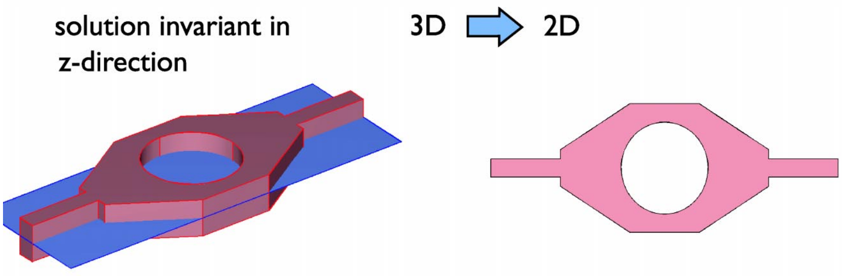
The problem couples two physical phenomena, a current balance and a heat balance. The electrical resistance increases with temperature while the current density itself serves as a heat source through ohmic losses in the current conduction. This implies that the multiphysics coupling between the two physics is a two-way coupling (i.e. we can not solve either problem separately from the other).

To simplify this problem and keep run-times low, we will look at the current and heat balances in a two-dimensional approximation of the real geometry. This will give us an estimation of the time scale of the problem and of the magnitude of the studied fields, potential and temperature.

A more advanced analysis would consist in solving a full-fledged 3D model of the film. Such a study could be performed in FEMLAB. Since the purpose of this document is only to familiarize ourselves with the software, we will not go through this analysis.

### CURRENT BALANCE AND HEAT BALANCES IN A TWO-DIMENSIONAL CROSS SECTION

Let us look at the current distribution in the device. Current is only conducted in the aluminum foil since the silicon substrate is an electrical insulator. Assuming isothermal conditions, the current density along the thickness should be perfectly uniform, since no variations are introduced by the geometry, current conduction, or boundary conditions. This means that we can reduce the problem down to two dimensions to study the current balance. The current passes through the device from right to left, in the figure above, and all the boundaries except the left and right boundaries are assumed to be insulated. We will study the current density distribution, and the influence of temperature on the total ohmic losses in the device.



The heat balance in the device however is clearly a three dimensional phenomenon. That being said, we can get an estimate of the production of heat in the device by performing a two-dimensional analysis. We can introduce the convective cooling perpendicular to the surface as a heat sink, in order to investigate the convective cooling's influence on the time scale of the phenomena. This would give a good start as we take the next step to three-dimensional modeling.

Below follows a detailed description of the modeling procedure in this exercise. The step-by-step instructions will guide us through the current balance at steady state and the introduction of the multiphysics content in the time-dependent heat transfer problem.

**NOTE:** *Although in reality the conductivity of the aluminum layer depends on its temperature, we first neglect this dependence and solve the electricity part of the problem with an assumed constant conductivity. This is meant to help us get familiar with basic FEMLAB concepts with a simpler problem. In a second step we will add this dependence and solve the actual physical problem where the electrical problem and the heat transfer problems are coupled and we need to solve for the electric potential and temperature simultaneously.*

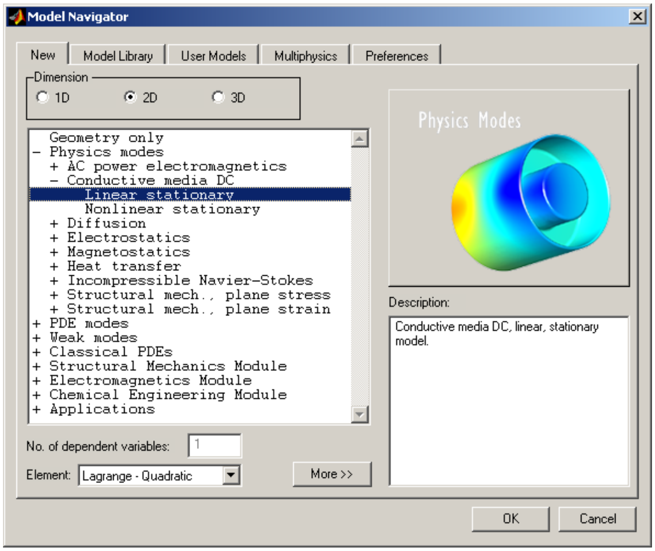
### *The Model Navigator*

- Start FEMLAB by double-clicking on the FEMLAB icon on the desktop or by typing `femlab` in the Matlab command window.
- In the **Model Navigator**, set **Dimension** to **2D** and then select **Physics mode** (double click), **Conductive Media DC** (double-click)**, Linear Stationary** in the list of available application modes.

**Note: Conductive Media DC** *is one of a number of physics problems for which FEMLAB possesses a tailored interface. We refer to those as application modes. You will find a description of the FEMLAB core application modes in the User's Guide and Introduction manual in the chapter called The Application Modes.*

*Application modes that come in the modules are documented in the module manuals.*

- Press **OK**.



Dimension: ○ 1D ● 2D ○ 3D

```
Geometry only
- Physics modes
  + AC power electromagnetics
  - Conductive media DC
      Linear stationary
      Nonlinear stationary
  + Diffusion
  + 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
+ Applications
```

No. of dependent variables: 1

Element: Lagrange - Quadratic

Description:

Conductive media DC, linear, stationary model.

### Draw Mode

The second step in the modeling process is to create or import the model geometry. FEMLAB is able to read and repair DXF and IGES files from other solid modeling packages. However, the geometry of the aluminum deposit is very easily created in FEMLAB's built-in solid modeling tool.

**Note:** *Extensive information on the various ways of creating a geometry is available from the Geometry Modeling chapter of the User's Guide and Introduction manual.*

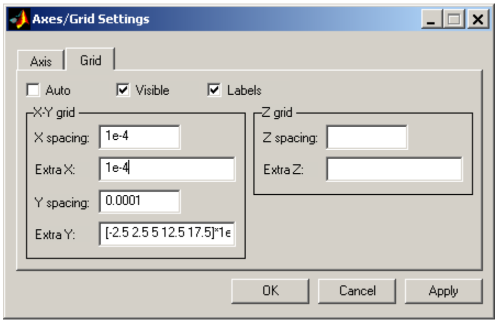
First we can specify the range of the drawing window and specify the grid spacings.

- Select **Axes/Grid settings** in the **Options** menu
- Type `-5.5e-5` in the **X min** and `1.05e-3` in the **X max** edit field

**Note:** *FEMLAB does not use a specific set of units: you can specify your problem in any consistent set of units. Quantities output by FEMLAB will then be in the same set of units. Here we use SI units.*

- Type `-3.5e-4` in the **Y min** and `3.5e-4` in the **Y max** edit field

- Click on the **Grid** tab
- Clear the **Auto** box
- Type `1e-4` in the **X spacing** and **Y spacing** edit fields
- Insert extra grid lines by typing `[-2.5 2.5 5 12.5 17.5]*1e-5` in the **Extra Y** edit field and press **OK**.



Axes/Grid Settings

Axis | Grid

☐ Auto ☑ Visible ☑ Labels

X-Y grid

X spacing: 1e-4

Extra X: 1e-4

Y spacing: 0.0001

Extra Y: [-2.5 2.5 5 12.5 17.5]*1e

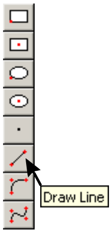
Z grid

Z spacing:

Extra Z:

OK | Cancel | Apply

This gives a proper grid spacing for the drawing of the device. You can now start drawing the geometry. The geometry can be created using solid primitive objects or line and curve tools. We will use the line tool to create the main part of the device.



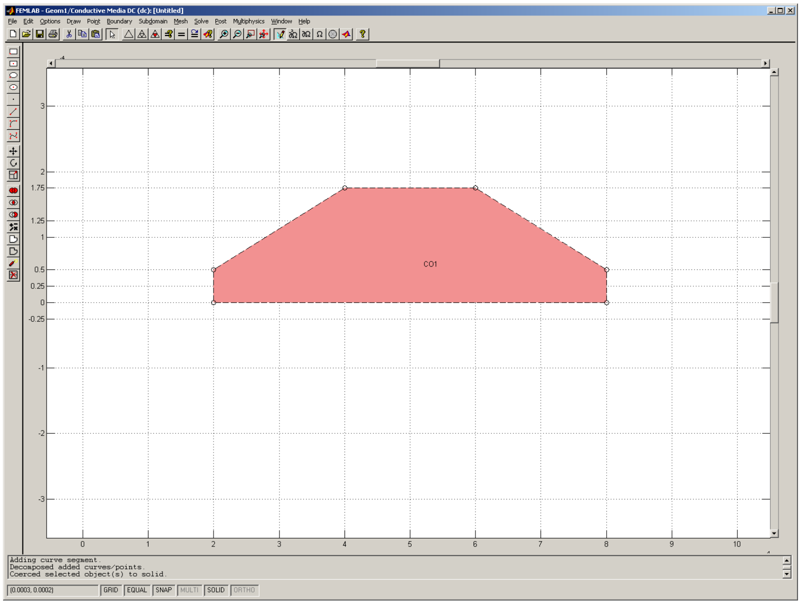
- Press the **Draw Line** tool button (It is sixth from the top in the draw toolbar on the left side of the FEMLAB window.

**Note** *If you hold the cursor steady on a tool button, its associated function is displayed. Also, note that you can see the location of the pointer in the left most box of the status bar at the bottom left corner of the FEMLAB window).*

- Click on the coordinates tabulated below to create the thick part of the device.

| X COORDINATES | Y COORDINATES |
|---|---|
| `2e-4` | `0` |
| `2e-4` | `0.5e-4` |
| `4e-4` | `1.75e-4` |
| `6e-4` | `1.75e-4` |
| `8e-4` | `0.5e-4` |
| `8e-4` | `0` |

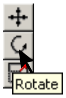
- Press the right mouse button to create a solid object denoted `CO1`, composite object 1.


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
CO1
3
2
1.75
1.25
1
0.5
0.25
0
-0.25
-1
-2
-3
0
1
2
3
4
5
6
7
8
9
10
GRID
EQUAL
SNAP
MULTI
SOLID
ORTHO

```
Adding curve segment.
Decomposed added curves/points.
Coerced selected object(s) to solid.
```

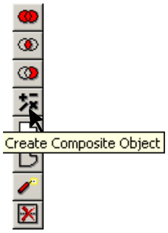
(0.0003, 0.0002)

- Copy CO1 by using the short-cut key combinations Ctrl-C and Ctrl-V or by selecting **Copy** followed by **Paste** in the **Edit** menu.
- Type 0 in the **X-displacement** and **Y-displacement** edit fields and press **OK**. Note that the new object, denoted CO2, is behind CO1 and therefore is indistinguishable.



- Press the **Rotate** button and type 180 in the **Rotation** and [5e-4 0] in the **Center coordinates** edit fields respectively.

You have created CO1 and CO2 and can continue with FEMLAB's **Create Composite Object** tool to unify these objects.



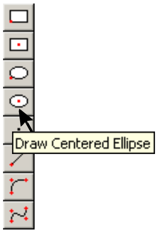
- Select CO1 and CO2 by using the key combination Ctrl-A and press the **Create Composite Object** button (fifth button from the bottom on the vertical toolbar). Check that the **Set formula** is CO1+CO2. This means that the resulting composite object is the union of CO1 and CO2.
- Clear the **Keep internal borders** box and press **OK**.

Draw the contact strips to the main body of the device:

- Press the **Draw Rectangle** tool (first button on the vertical toolbar).

- Hold down the left mouse button and drag the mouse between the coordinates `(0,-0.25e-4)` and `(1e-3,0.25e-4)`to create a rectangle.
- Select the rectangle `R1` and the composite `CO1` and unify them by using the **Create Composite Object** tool. The **Keep internal borders** box should be cleared.

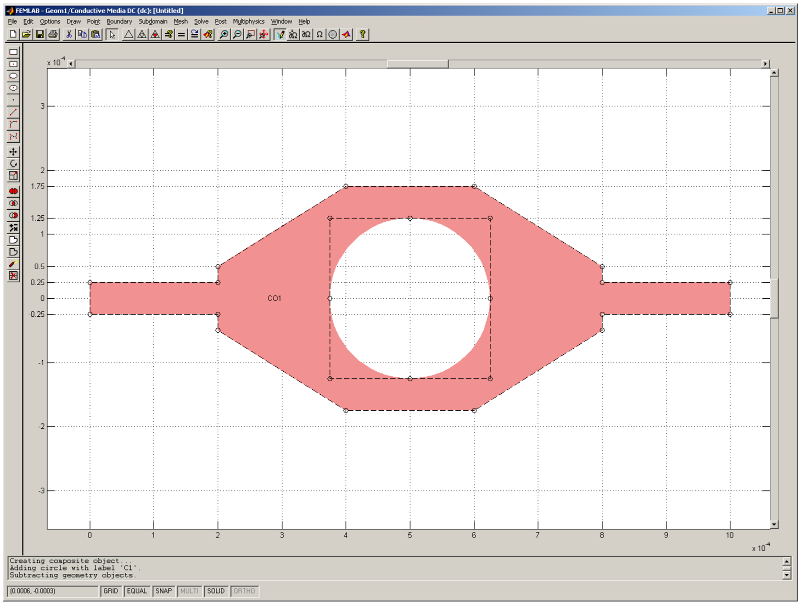
Most of the geometry is now done. You can now create the circle in the middle and drill the hole in the structure as follows:



- Select the **Draw Centered Ellipse** tool.
- Using the right mouse button, create a circle by dragging the mouse straight upwards from `(5e-4,0)` to `(5e-4,1.25e-4)`.

**Note:** *You can check that the disc has the proper center and radius by double-clicking on C1.*

- Select all the objects by using the key combination `Ctrl-A` and press the **Difference** button (sixth button from the bottom on the vertical toolbar).


FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
x 10^-4
3
2
1.75
1.25
1
0.5
0.25
0
-0.25
-1
-2
-3
CO1
0
1
2
3
4
5
6
7
8
9
10
x 10^-4

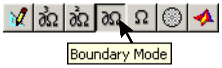
```
Creating composite object...
Adding circle with label 'C1'.
Subtracting geometry objects.
```

(0.0006, -0.0003) GRID EQUAL SNAP MULTI SOLID ORTHO

The geometry of the aluminum deposit is completed and you can go to the **Boundary** mode:

### *Boundary Mode*

- Press the **Boundary Mode** button on the main toolbar (fifth from right)



- Select all boundaries by using the key combination `Ctrl-A`
- Select **Boundary settings** in the **Boundary** menu
- Set **Insulation/symmetry** conditions at all boundaries
- Select the right hand side boundary, boundary 16, and press the **Inward current density** radio button. You can check that you have selected the correct boundary in the **Domain selection** list in the **Boundary settings** dialog box.
  Type `iin/(thick*side)` in the **g** edit field. `iin`, `thick`, and `side` are constants that you will define later.
- Select the left hand side boundary, number 1, and set the **Antisymmetry/ground** boundary condition.
- Press **OK**.

**Note:** *The boundaries are rendered differently based on what boundary conditions apply there.*

The next step is to define the constants used in the above expression in a temporary data base.

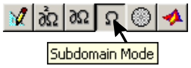
- Select **Add/edit constants** in the **Options** menu
- Define the constants tabulated below in the corresponding edit fields. This is done by entering the name and expression in the table below into the appropriate dialog boxes and then clicking either the **Set** or the **Apply** button. `sig` is a constant that you will use later on to define the electrical conductivity of the material.

| NAME OF CONSTANT | EXPRESSION |
|---|---|
| `iin` | `0.5` |
| `thick` | `1e-6` |
| `side` | `50e-6` |
| `sig` | `3.54e7` |

- Press **OK**

This defines the boundary conditions for the current density balance. Proceed to the subdomain mode.

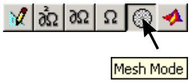
### Subdomain Mode



- Press the **Subdomain Mode** button in the main toolbar
- Double-click on the geometry or open **Subdomain settings** from the **Subdomain** menu
- Type `sig` in the **Conductivity** edit field
- Type `0` in the **Current source** edit field
- Press **OK**

The constant `sig` corresponds to the conductivity, in S m$^{-1}$, at 293 K.

### Mesh Mode



- Press the **Mesh Mode** button in the main toolbar to create the mesh

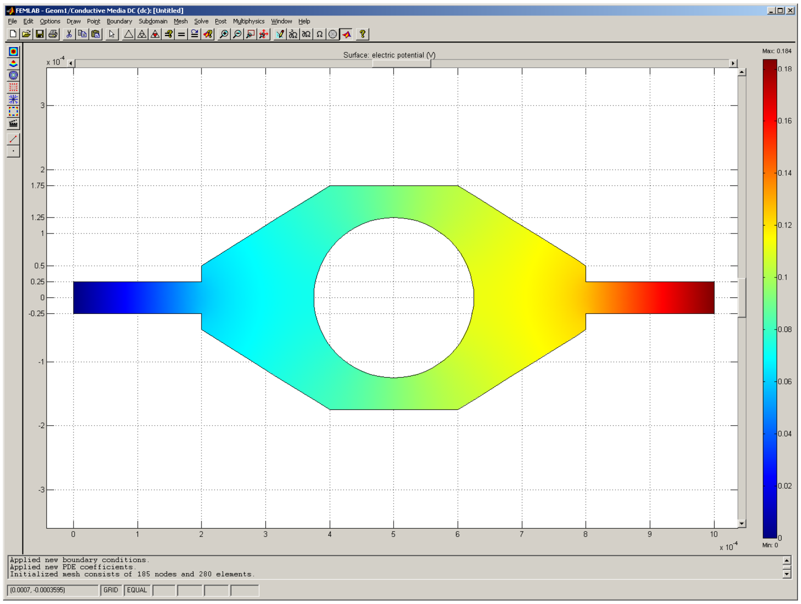
**Note:** *Here we are using the default settings for mesh generation, but very elaborate meshes can also be created from the Mesh Parameters dialog window, which is documented in the Graphical User Interface chapter of the Reference manual. Finer meshes usually yield more accurate solutions and it is always a good idea to compare the solution obtained with different meshes to make sure that a sufficiently accurate solution has been reached, as will be done later on in this document.*

### Solve Mode

- Solve the problem by pressing the **Solve Problem** button (=) in the main toolbar

### *Post Mode*

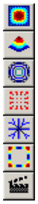
The graphical user interface automatically selects the **Post Mode** once the problem is solved. The default plot shows the voltage in the aluminum deposit.Post mode quick



FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post Multiphysics Window Help
Surface: electric potential (V)
x 10^-4
3
2
1.75
1.25
1
0.5
0.25
0
-0.25
-1
-2
-3
0 1 2 3 4 5 6 7 8 9 10
x 10^-4
Max: 0.184
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Min: 0

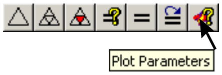
```
Applied new boundary conditions.
Applied new PDE coefficients.
Initialized mesh consists of 185 nodes and 280 elements.
```

(0.0007, -0.0003595) GRID EQUAL



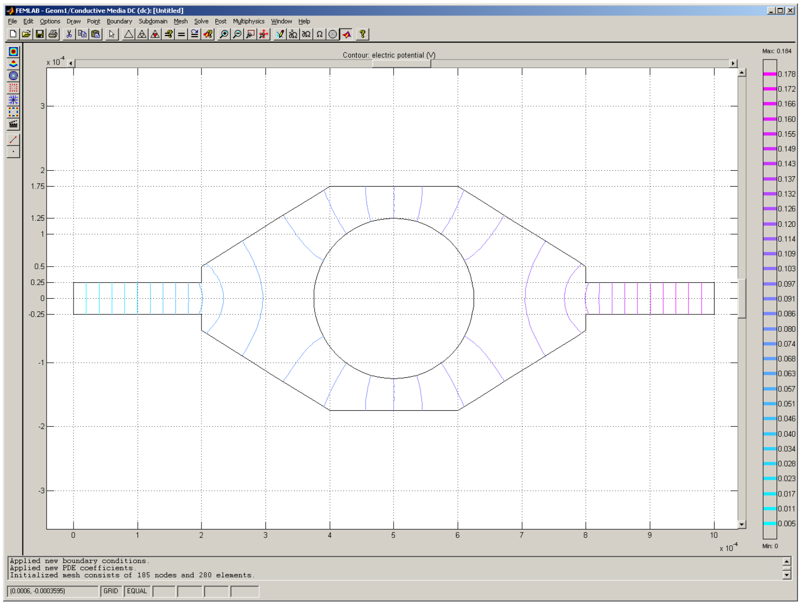
buttons

You can toggle between a number of visualization options by pressing the quick buttons in the **Post Mode**.



- Press the **Plot Parameters** button in the main toolbar
- Clear the **Surface** box and check the **Contour** box
- Select the **Contour** tab
- Type 31 in the **Contour levels** edit field. This will render 31 contour lines with constant potential increment
- Press **OK**

**Note:** *The iso-potential lines above and below the hole are perfectly vertical, see the figure below. This implies that anti-symmetry is present along these vertical lines while symmetry is obtained between the upper and lower part of the geometry. The presence of symmetry and anti-symmetry would allow for the modeling of only one fourth of the device.*



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

```
Applied new boundary conditions.
Applied new PDE coefficients.
Initialized mesh consists of 185 nodes and 280 elements.
```

(0.0006, -0.0003595) GRID EQUAL

**Note:** *While the FEMLAB graphical user interface allows advanced postprocessing, there are cases where you may need to export your solution to an other program for analysis of the solution. This is easily done in FEMLAB. Visit our Knowledge Base at the following URL to familiarize yourself with this:*

*http://www.comsol.com/support/knowledgebase/*

Now that we have familiarized ourselves with the FEMLAB interface and its main concepts, we can start modeling the actual problem of interest by introducing the heat transfer problem which, as indicated earlier, is coupled to the electric problem. To demonstrate FEMLAB's capabilities, the thermal part of the problem will be specified consecutively using four different application modes: heat transfer physics mode, coefficient form PDE mode, general form PDE mode and weak mode.

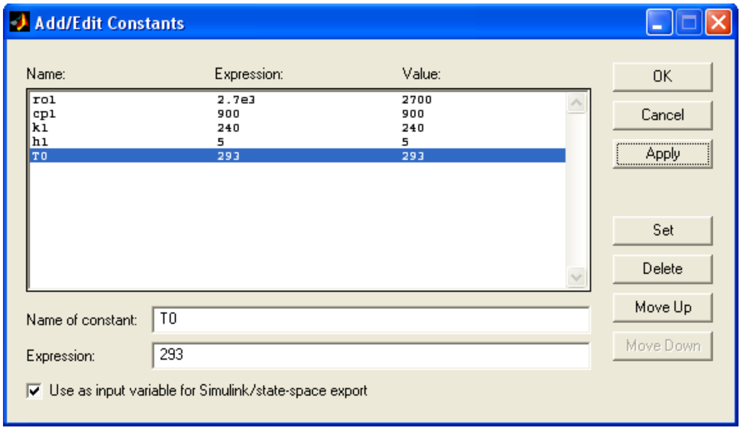
The next step introduces the concept of multiphysics. FEMLAB allows for arbitrary coupling of several physical phenomena. In this case, you will add the heat balance to the existing current balance.

Before we do that, let us define some constants and expressions that will be required for all four application modes.

First, the constants:

- Select **Add/edit constants** in the **Options** menu
- Define the constants according to the table below

| NAME OF CONSTANT | EXPRESSION |
|---|---|
| ro1 | 2.7e3 |
| cp1 | 900 |
| k1 | 240 |
| h1 | 5 |
| T0 | 293 |



| Name: | Expression: | Value: |
|---|---|---|
| ro1 | 2.7e3 | 2700 |
| cp1 | 900 | 900 |
| k1 | 240 | 240 |
| h1 | 5 | 5 |
| T0 | 293 | 293 |

Name of constant: T0

Expression: 293

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

- Press **OK**

In addition to constants, you can define expressions that are functions of the dependent variables in the model, in this case voltage V and temperature T.

**Note 1:** *The voltage and temperature were named V and T respectively automatically by FEMLAB when we selected the application modes from the Model Navigator. If these names had not suited our needs, we could have changed them from the Model Navigator.*

**Note 2:** *Expressions can also depend on independent variables (such as x and y) and include calls to Matlab functions, including user-specified functions.*

- Select **Add/edit expressions** in the **Options** menu
- Type `res` in the **Variable name** edit field
- Press **Add**
- Click the **Definition** tab
- Define the ohmic resistivity as a function of temperature according to the table below by typing in the expression in the corresponding edit field. When you enter a scalar term like the one below in a FEMLAB dialog box, make sure *not* to use spaces (In FEMLAB, spaces are used to separate the entries of a vector).

| VARIABLE NAME | EXPRESSION |
|---|---|
| `res` | `2.824e-8*(1+3.9e-5*(T-293))` |

- Press **OK**
- Now, save your model under the title `FilmBasic.mat` using the **File** menu item **Save.** We will reuse this file as a basis for future modeling.
- Save the working model under the file name `FilmHTPM.mat` using the **Save as** item of the **File** menu. The letters ‘HTPM’ are a reminder that we will be using the Heat Transfer Physics Mode.

**Note:** *There are two formats in which you can save you FEMLAB models. One is the .mat format which allows you at a later to time to re-open you model where you left it, with the computed solution immediately available. The second format is the .m format which is a script format: the .m file contains a log of all the Matlab prompt commands that are equivalent to the operations performed in the GUI. When an .m file is open, all these commands are executed again in sequence. It is possible to add comments to an .m file just like you would in any other Matlab .m file.*

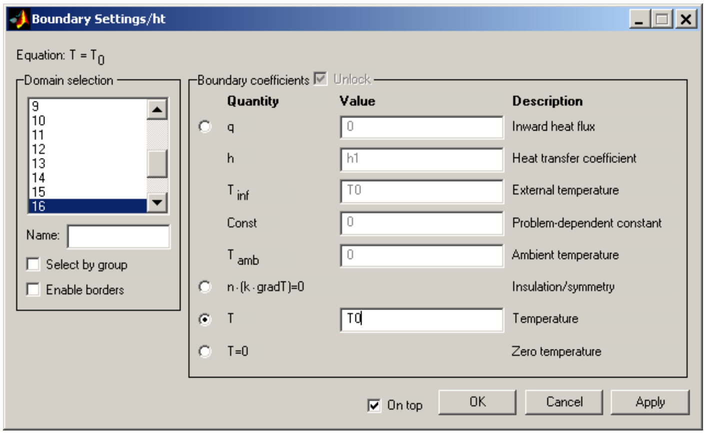
### *Multiphysics Mode*

- Select **Add/edit modes** in the **Multiphysics** menu
- Select the **Heat transfer** mode on the left hand side of the **Multiphysics** dialog box
- Press the **Add Selected Application Mode to Current Model (>>)** button
- Select **Time-dependent** as the **Solver** type
- Press **OK**

If you click on the **Multiphysics** item in the main menu bar, you can see that we have two application modes selected for our problem and that the currently-active one is the Heat Transfer application mode we just added.

### *Boundary Mode*

- Press the **Boundary Mode** button
- Select all boundaries by typing the key combination `Ctrl-A`
- Select **Boundary settings** in the **Boundary** menu
- Press the radio button to set the heat flux, which is the first button in the **Boundary coefficients** list
- Type `h1` in the **heat transfer coefficient** edit field
- Type `T0` in the **External temperature** edit field
- Select the left and right hand side boundaries, boundaries 1 and 16. You can also do this by holding down the Ctrl key and clicking on the boundaries in the drawing of the device if you do not know the numbers.
- Press the **Temperature** radio button and type `T0` in the corresponding edit field
- Press **O**K



Equation: $T = T_0$

Domain selection

9
10
11
12
13
14
15
16

Name:

☐ Select by group

☐ Enable borders

Boundary coefficients ☑ Unlock

| | Quantity | Value | Description |
|---|---|---|---|
| ○ | q | 0 | Inward heat flux |
| | h | h1 | 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 | T0 | Temperature |
| ○ | T=0 | | Zero temperature |

☑ On top

You will now continue by defining the domain physics for the heat transfer mode.

### *Subdomain Mode*

- Press the **Subdomain Mode** button
- Double-click on the domain

**Note:** *The equation solved for in the subdomain is displayed at the top of the Subdomain Settings window.*

The heat transfer through convection on the surface of the aluminum deposit is introduced as a source or a sink. Since invariance along the thickness of the device implies that the model is defined per unit length in the z direction, the heat transfer coefficient has to be divided by the thickness of the deposit. This gives the correct value of the cooling per unit volume of aluminum. The coupling between the heat balance and the current balance is present in the expression for the resistivity, denoted `res`, and the source term, which depends on the gradient of the potential, whose components are denoted by `Vx` and `Vy`. The equation we want to solve for in the subdomain is therefore:

$$\rho_1 c_{p1}\frac{\partial T}{\partial t} - \nabla \bullet (k_1 \nabla T) = \frac{\left(\frac{\partial V}{\partial x}\right)^2 + \left(\frac{\partial V}{\partial y}\right)^2}{\mathrm{res}} + \frac{h_1}{\mathrm{thick}}(T_0 - T)$$

With this in mind, define the **Subdomain settings** according to the table below:

| DESCRIPTION | VALUE |
|---|---|
| `Density` | `ro1` |
| `Heat capacity` | `cp1` |
| `Thermal conductivity` | `k1` |
| `Heat source` | `1/res*(Vx^2+Vy^2)` |
| `Convect. heat transf.coeff.` | `h1/thick` |
| `External temperature` | `T0` |

- Click the **Init** tab and type `T0` in the **Initial value** edit field
- Press **OK**

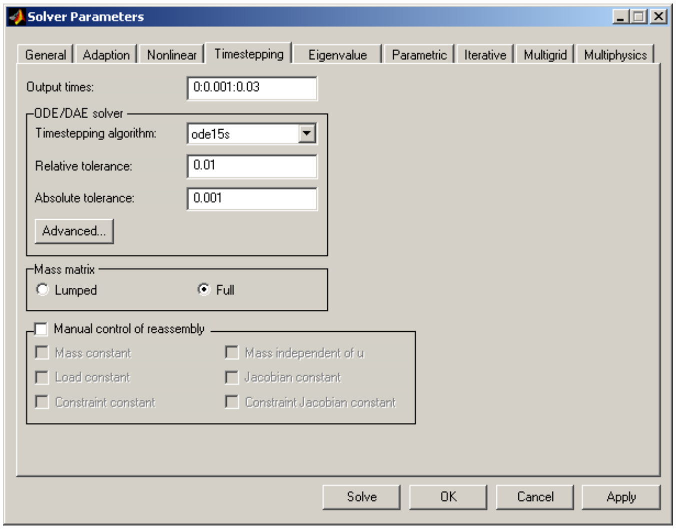
In order to make a two-way coupling between the heat and current balances, you have to introduce the expression for the resistivity, `res`, in the conductive media application.

- Select **1Geom1:Conductive media DC** in the **Multiphysics** menu
- Select **Subdomain settings** in the **Subdomain** menu
- Type `1/res` in the **Conductivity** edit field instead of `sig`. The resistivity `res` was specified earlier as an expression that depends on the temperature
- Press **OK**

### *Solve Mode*

- Select **Parameters** in the **Solve** menu
- Check that the **Time dependent** radio button in the **General** page is selected

- Click the **Timestepping** tab and type `0:0.001:0.03` in the **Output times** edit field
- Press **O**K



Output times: 0:0.001:0.03

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

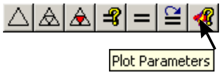
□ Load constant □ Jacobian constant

□ Constraint constant □ Constraint Jacobian constant

The vector expression for the output times gives the solution from 0 to 0.03 seconds with 0.001 second increments. The time stepping algorithm has built-in step length control and the output time only defines the time points that should be saved for post processing.

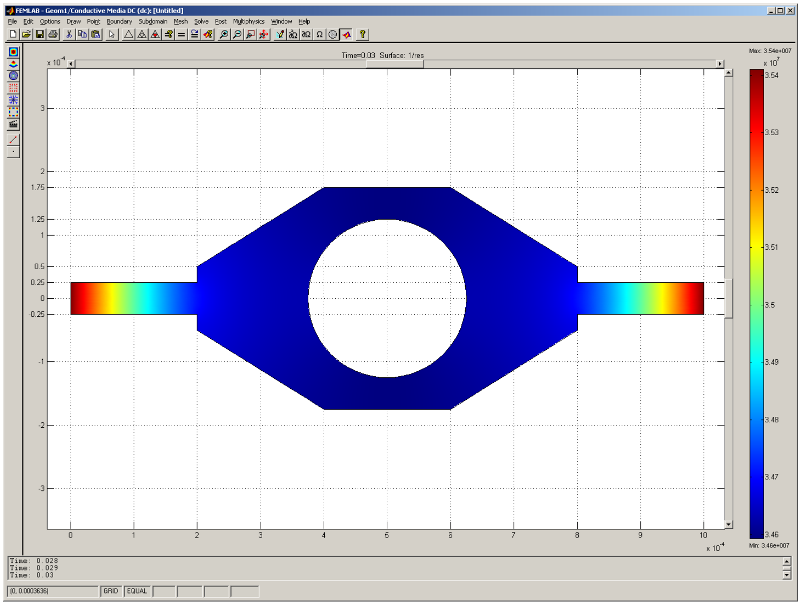
- Press the **Solve** button in the main tool bar

The solution shows the potential distribution in the aluminum deposit for the resulting temperature distribution. You can plot the conductivity as a function of temperature in the geometry:



- Press the **Plot parameters** button in the main toolbar
- Click the **General** tab
- Clear the **Contour** box and check the **Surface** box
- Select the **Surface** tab

- Type `1/res` in the **Surface expression** edit field
- Press **OK**



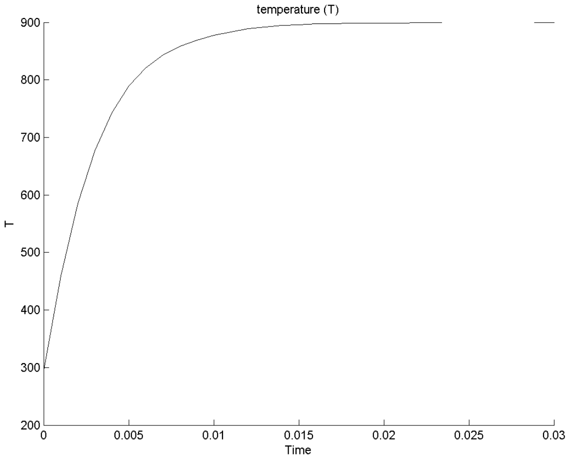
FEMLAB - Geom1/Conductive Media DC (dc): [Untitled]

```
Time: 0.028
Time: 0.029
Time: 0.03
```

You can study the development of the temperature profile by animating the solution through the Animate tab of the Plot Parameters window or by plotting the temperature as a function of time in the hottest part of the device.

- Select **Cross-Section plot parameters** in the **Post** menu
- Select the **Point** tab and press the **Point plot** radio button
- Select **temperature (T)** in the **Point expression** combo box
- Type `500e-6` in the **X:** and `125e-6` in **Y:** edit fields
- Select the **General** tab

- Select all time steps in the Solution at time list
- Press **OK**



The resulting plot shows that the problem has reached steady-state after 0.02 seconds. You can continue visualizing the current density, potential, and current density distribution by combining cross sectional, surface, contours, and flow plots.

### CONVERGENCE ANALYSIS AND POST-PROCESSING

To check the numerical accuracy of the solutions, we need to examine the sensitivity of the values to the mesh density. We will look at a couple of key numbers in the solution at the end time and compare them at different mesh densities.

- Change to a stationary problem by selecting the **Solve/Parameters...** menu and under the **General** tag select **Solver type** to **Stationary nonlinear** and **Solution form** to **General**.
- Solve the problem by clicking the **=** button.
- Evaluate the norm of the temperature gradient in (97e-5, 0) by selecting **Post/Cross Section Plot Parameters** from the **Post** menu and selecting **Point** on the **General** tab. On the **Point** tab, select **Point expression**: `temp. gradient (gradT_ht)` and **Coordinates: X:** `97e-5,` **Y:** `0.`
- Evaluate the mean temperature

$$\langle T\rangle = \int TdA/A$$

by selecting **Subdomain Integration** in the **Post** menu and specifying **Expression** to `temperature (T).` The result will show up in the status field. Divide this number by the total area, which can be obtained by a second subdomain integration, typing in `1` in the **Expression** field. The average temperature should be approximately 830.5 degrees.

- The last two steps can be automated with a few lines of MATLAB code. Create an m-file in `C:\MATLAB6p5\FEMLAB23` with the following contents:

```
grad1=postinterp(fem,'gradT_ht',[97e-5;0])
Tmean=postint(fem,'T')/postint(fem,'1')
```

**Note:** `postinterp` *and* `postint` *are FEMLAB commands that are documented in the Function Reference chapter of the Reference manual. You can also learn more about the syntax of these commands by typing*

```
doc postint
```

*or*

```
doc postinterp
```

*at the MATLAB command line.*

- In the GUI, do CTRL-F before running the m-file. This will export all of the data structure from FEMLAB to MATLAB and therefore enable post-processing of the FEMLAB solution from the MATLAB prompt.

If you wish, you can repeat the runs of the m-file for different mesh densities: select the **Mesh/Parameters...** menu and specify the mesh size: **More>>**, **Max element size for subdomains:** `1 1e-4.` Repeat to generate the following table.

| MESH ELEMENT SIZE | GRADT [97E-5, 0] | <T> |
|---|---|---|
| 1 1e-4 | 3.4428e6 | 830.4682 |
| 1 3e-5 | 3.4430e6 | 830.5551 |
| 1 1.5e-5 | 3.4474e6 | 831.7698 |
| 1 0.75e-5 | 3.4496e6 | 832.4117 |
| -* | 3.4503e6 | 832.6016 |

*Use the following expression in **Mesh size expression:**

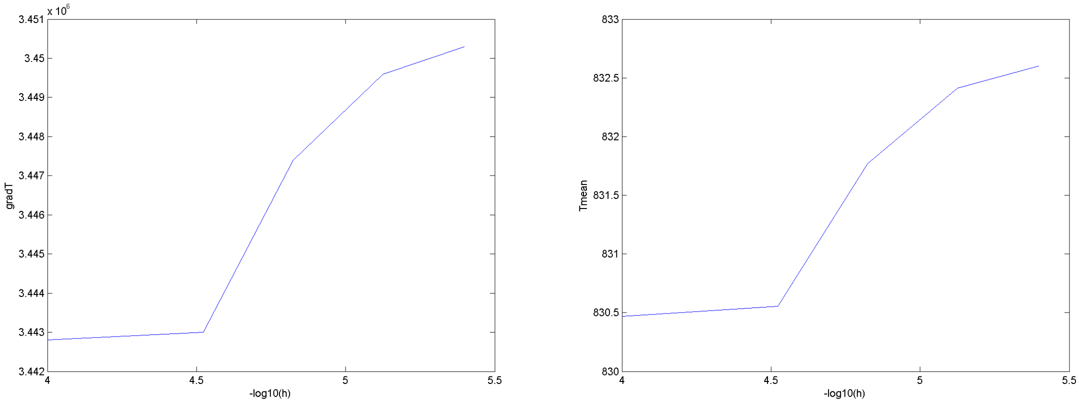
`(x<2e-4)*.4e-5+(x>=2e-4)*(x<8e-4)*1.5e-5+(x>=8e-4)*.4e-5`

The following MATLAB code is used to generate two convergence diagrams:

```
Tmeans=[830.4682 830.5551 831.7698 832.4117 832.6016];
gradTs=[3.4428e6 3.4430e6 3.4474e6 3.4496e6 3.4503e6];
Mesh=[1e-4 3e-5 1.5e-5 0.75e-5 0.4e-5];
figure
```

```
plot(-log10(Mesh),gradTs)
xlabel('-log10(h)')
ylabel('gradT')
figure
plot(-log10(Mesh),Tmeans)
xlabel('-log10(h)')
ylabel('Tmean')
```

This series of commands draws the the figure below where piecewise linear interpolation is employed to join the 5 data points presented in the table on the previous page. The fact that these diagrams flatten out when the elements get smaller is an indication that we have probably reached a reasonably accurate solution with the finer meshes and that the solution would not vary significantly if we refined the mesh further.



This concludes our analysis based on the Heat Transfer physics mode. Save your simulation by selecting **Save** from the **File** menu.

## *Coefficient Form PDE Mode*

We will now solve the exact same problem by use of the Coefficient Form PDE mode for the heat transfer part of the problem. This is done to demonstrate the Coefficient Form PDE mode. When there exists a physics mode for the problem you want to model, using that physics mode is by far the easiest and safest way to go as the user interface is then tailored for the problem in question. However, when you wish to solve an equation for which there is no physics mode available the PDE modes allow you to specify your own equation. The Coefficient Form PDE mode is most appropriate for modeling linear or weakly nonlinear problems.

**Note:** *the PDE modes are documented in the chapter of the User's Guide and Introduction manual entitled The FEMLAB PDE Language.*

- First, open the file `FilmBasic.mat` by selecting **Open** in the **File** menu
- Save the model as FilmCoeff.mat using the **Save** as item in the **File** menu
- Select **Add/edit modes** in the **Multiphysics** menu
- Highlight the mode named **PDE, coefficient form** on the left hand side of the **Multiphysics** dialog box and name T the **dependent variable**.
- Press the **Add Selected Application Mode to Current Model (>>)** button
- Change the **Solver type** to **Time dependent**
- Press **OK**

**Note:** *If you click on the Multiphysics item on the main menu bar, you can check that two application modes are selected and the active one is the Coefficient Form PDE mode.*

### *Subdomain Mode*

- Select **Subdomain Settings** from the **Subdomain** menu and click on subdomain 1.

The equation we need to solve in the subdomain is still

$$\rho_1 c_{p1}\frac{\partial T}{\partial t} - \nabla \bullet (k_1 \nabla T) = \frac{\left(\frac{\partial V}{\partial x}\right)^2 + \left(\frac{\partial V}{\partial y}\right)^2}{res} + \frac{h_1}{thick}(T_0 - T)$$

The equation solved for in the Coefficient Form PDE mode is

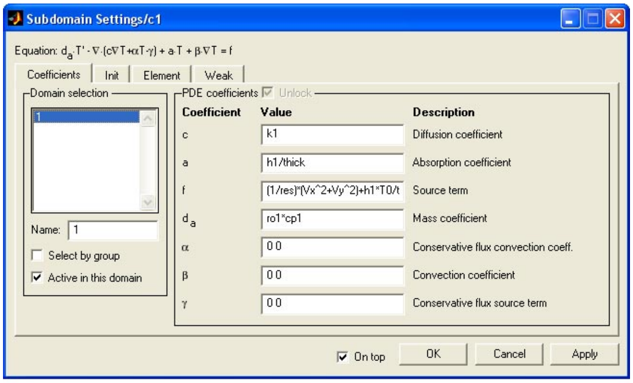
$$d_a\frac{\partial T}{\partial t} - \nabla \bullet (c\nabla T + \alpha T - \gamma) + aT + \beta \bullet \nabla T = f$$

**Note:** *This equation is displayed at the top of the Subdomain Settings window.*

- Therefore, enter the data from the following table in the dialog fields of the **Subdomain Settings** window after selecting domain 1 from the **Domain selection** list:

| DESCRIPTION | VALUE |
|---|---|
| c | k1 |
| a | h1/thick |
| f | (1/res*(Vx^2+Vy^2))+(h1/thick)*T0 |
| da | ro1*cp1 |

| DESCRIPTION | VALUE |
|---|---|
| $\alpha$ | 0 0 |
| $\beta$ | 0 0 |
| $\Upsilon$ | 0 0 |



Equation: $d_a \cdot T' - \nabla \cdot (c\nabla T + \alpha T - \gamma) + a \cdot T + \beta \cdot \nabla T = f$

Name: 1

☐ Select by group

☑ Active in this domain

| Coefficient | Value | Description |
|---|---|---|
| c | k1 | Diffusion coefficient |
| a | h1/thick | Absorption coefficient |
| f | (1/res)*(Vx^2+Vy^2)+h1*T0/t | Source term |
| $d_a$ | ro1*cp1 | Mass coefficient |
| $\alpha$ | 0 0 | Conservative flux convection coeff. |
| $\beta$ | 0 0 | Convection coefficient |
| $\gamma$ | 0 0 | Conservative flux source term |

☑ On top

- Click the **Init** tab and type `T0` in the **Initial value** edit field
- Press OK

### *Boundary Mode*

We want to impose Neumann boundary conditions on all boundaries except 1 and 16 and Dirichlet conditions on those two. The easy way to do this is to first impose Neumann boundary conditions on all boundaries and then modify the boundary conditions applied to 1 and 16 (Same as we did earlier).

- Press the **Boundary Mode** button (fifth from right on the main tool bar, below the menu bar)
- Select all boundaries using Ctrl+A
- Select **Boundary Settings** from the **Boundary** menu. Make sure to toggle on the **Neumann boundary condition**. As you can see at the top of the window, the Neuman boundary condition in the Coefficient Form PDE mode is:

$$n \bullet (c\nabla T + \alpha T - \Upsilon) + qT = g$$

The boundary condition we need to impose is as before:

$$\mathrm{n} \bullet (k_1 \nabla T) = h_1 (T_0 - T)$$

We have already defined c, $\alpha$ and $\Upsilon$. Enter the values in the table below into the proper dialog fields:

| DESCRIPTION | VALUE |
|---|---|
| q | h1 |
| g | h1*T0 |

- Select boundaries 1 and 16 from the **Domain selection** list (holding the control key)
- Select Dirichlet boundary condition

The Dirichlet boundary condition in the Coefficient Form PDE mode is:

$$\mathrm{n} \bullet (c \nabla T + \alpha T - \Upsilon) + qT = g - h^T \mu \qquad ; \qquad hT = r$$

The rightmost term in the first of these two equations corresponds to a Lagrange multiplier used to enforce the Dirichlet condition.

- Enter the values in the table below into the proper dialog fields

| DESCRIPTION | VALUE |
|---|---|
| q | 0 |
| g | 0 |
| h | l |
| r | T0 |

- Click **OK**

*Subdomain Mode*

Before we run the simulation, we need to finish specifying the two-way coupling by setting the electric conductivity to `1/res` in the **Subdomain Setting**s of the electrical application mode.

- Select **I Geom1: conductive media DC (dc)** from the **Multiphysics** menu to switch back to the electrical application mode
- In the **Subdomain Settings** (under the **Subdomain** menu) window, enter `1/res` instead of `sig` as the conductivity for subdomain 1
- Click OK

### *Solve Mode*

- Select **Parameters** in the **Solve** menu
- Check that the **Time dependent** radio button is selected on the **General** page
- Click the **Timestepping** tab and type `0:0.001:0.03` in the **Output times** edit field
- Press **OK**
- Press the **Solve** button in the main tool bar

The solution is the same as earlier. For instance, you could easily verify that the same integral for the temperature is obtained, with a value of 0.00010873.

- Save your model by selecting **Save** from the **File** menu. Alternatively, use Ctrl+S

## *General Form PDE Mode*

We will now solve the exact same problem by use of the General Form PDE mode for the heat transfer part of the problem. Our goal is to demonstrate this alternative way of specifying your own equations in FEMLAB. This form is particularly appropriate for non-linear problems.

**Note:** *the PDE modes are documented in the chapter of the User's Guide and Introduction manual entitled The FEMLAB PDE Language.*

- First, open the file `FilmBasic.mat` by selecting **Open** in the **File** menu
- Save the model as FilmGeneral.mat using the **Save as** item in the **File** menu
- Select **Add/edit modes** in the **Multiphysics** menu
- Highlight the mode named **PDE, general form** on the left hand side of the **Multiphysics** dialog box and name `T` the dependent variable
- Change the **Solution form** to **General**
- Press the **Add Selected Application Mode to Current Model (>>)** button
- Change the **Solver type** to **Time dependent**
- Press **OK**
- Note that if you click on the **Multiphysics** item on the main menu bar, you can check that two application modes are selected and the active one is the General Form PDE mode

### *Subdomain Mode*

- Select **Subdomain Settings** from the **Subdomain** menu and click on subdomain 1

Once again, the equation we need to solve in the subdomain is:

$$\rho_1 c_{p1}\frac{\partial T}{\partial t} - \nabla \bullet (k_1 \nabla T) = \frac{\left(\frac{\partial V}{\partial x}\right)^2 + \left(\frac{\partial V}{\partial y}\right)^2}{res} + \frac{h_1}{thick}(T_0 - T)$$

The equation solved for in the General Form PDE mode is:

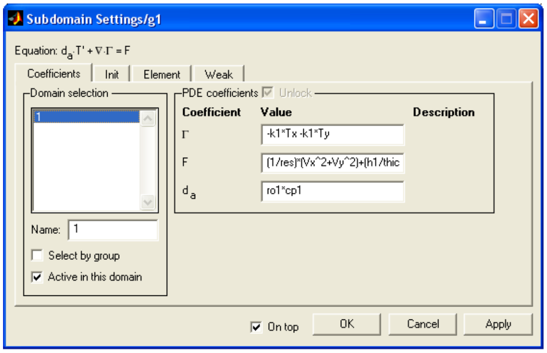
$$d_a\frac{\partial T}{\partial t} + \nabla \bullet \Gamma = F$$

Observe that this equation is displayed at the top of the **Subdomain Settings** window.

- Therefore, enter the data from the following table in the dialog fields of the **Subdomain Settin**gs window:

| DESCRIPTION | VALUE |
|---|---|
| $\Gamma$ | `-k1*Tx -k1*Ty` |
| F | `(1/res)*(Vx^2+Vy^2)+(h1/thick)*(T0-T)` |
| da | `ro1*cp1` |

It is important that you leave a space between `-k1*Tx` and `-k1*Ty` to specify that these are the two components of the $\Gamma$ vector.



Equation: $d_a \cdot T' + \nabla \cdot \Gamma = F$

1

| Coefficient | Value |
|---|---|
| $\Gamma$ | -k1*Tx -k1*Ty |
| F | (1/res)*(Vx^2+Vy^2)+(h1/thic |
| $d_a$ | ro1*cp1 |

Name: 1

☐ Select by group

☑ Active in this domain

☑ On top

- Click the **Init** tab and type `T0` in the **Initial value** edit field
- Press OK

### *Boundary Mode*

We want to impose Neumann boundary conditions on all boundaries except 1 and 16 and Dirichlet conditions on those two. As earlier, we will do this by first imposing Neumann boundary conditions on all boundaries and then modifying the boundary conditions applied to 1 and 16.

- Press the **Boundary Mode** button
- Select all boundaries using Ctrl+A
- Select **Boundary Settings** from the **Boundary** menu. Make sure that the Neumann boundary condition is toggled on. As you can see at the top of the window, the equation for boundary condition in the Coefficient Form PDE mode is:

$$-\mathrm{n} \bullet \Gamma = \mathrm{G} + \left(\frac{\partial \mathrm{R}}{\partial \mathrm{T}}\right)^{\mathrm{T}} \mu \qquad ; \qquad \mathrm{R} = 0$$

The boundary condition we need to impose is as before:

$$\mathrm{n} \bullet (k_1 \nabla \mathrm{T}) = h_1(\mathrm{T}_0 - \mathrm{T})$$

We have already defined $\Gamma$ .

- Enter `h1*(T0-T)` in the **G** dialog field
- Select boundaries 1 and 16 from the **Domain selection** list
- Select Dirichlet boundary condition
- Enter `0` in the **G** dialog field and `-T+T0` in the **R** dialog field
- Click **OK**

### *DC Subdomain Mode*

Before we run the simulation, we need to finish specifying the two-way coupling by setting again the electric conductivity to `1/res` in the **Subdomain Settings** of the electrical application mode.

- Select **1 Geom1: conductive media DC (dc)** from the Multiphysics menu to switch back to the electrical application mode
- In the **Subdomain Settings** (under the Subdomain menu) window, enter `1/res` instead of `sig` as the conductivity for subdomain 1
- Click OK

### *Solve Mode*

- Select **Parameters** in the **Solve** menu

- Press the **Time dependent** radio button in the **General** page
- Click the **Timestepping** tab and type `0:0.001:0.03` in the **Output times** edit field
- Press **OK**
- Press the **Solve** button in the main tool bar

The solution is the same as earlier. For instance, you could easily verify that the same integral for the temperature is obtained (Again, 0.00010873 should be its value).

- Save your model by selecting **Save** from the **File** menu. Alternatively, use Ctrl+S.

## *Weak Mode*

We will now solve the exact same problem by use of the Weak mode for the heat transfer part of the problem. The Weak mode is a very general way of specifying your governing equation in FEMLAB. It offers a great deal of flexibility and may be required in the creation some very advanced models.

- First, open the file `FilmBasic.mat` by selecting **Open** in the **File** menu
- Save the model as FilmWeak.mat using the **Save as** item in the **File** menu
- Select **Add/edit modes** in the **Multiphysics** menu
- Highlight the mode named **Weak, subdomain** on the left hand side of the **Multiphysics** dialog box and name `T` the dependent variable
- Press the **Add Selected Application Mode to Current Model (>>)** button
- Change the **Solver type** to **Time-dependent**
- Press **OK**

**Note:** *that if you click on the Multiphysics item on the main menu bar, you can check that two application modes are selected and the active one is the Subdomain weak form PDE mode.*

### *Subdomain Mode*

- Select **Subdomain Settings** from the **Subdomain** menu and click on subdomain 1.

The equation we need to solve in the subdomain is

$$\rho_1 c_{p1}\frac{\partial T}{\partial t} - \nabla \bullet (k_1 \nabla T) = \frac{\left(\frac{\partial V}{\partial x}\right)^2 + \left(\frac{\partial V}{\partial y}\right)^2}{res} + \frac{h_1}{thick}(T_0 - T)$$

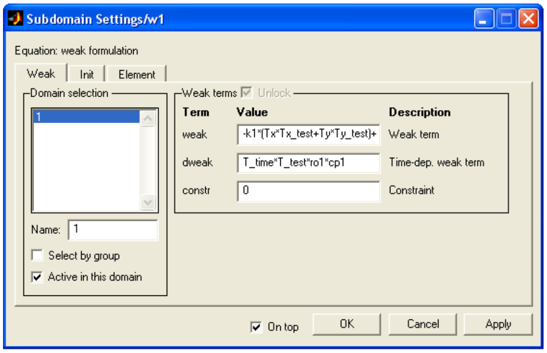
Multiplying by a test function $\bar{T}$, integrating over the subdomain, integrating by parts the diffusion term and rearranging terms yields:

$$\int_{\Omega} \bar{T}\rho_1 c_{p1}\frac{\partial T}{\partial t} = \ldots$$

$$\ldots = \int_{\Omega} -k_1\nabla\bar{T}\bullet\nabla T + \bar{T}\left(\frac{\left(\frac{\partial V}{\partial x}\right)^2+\left(\frac{\partial V}{\partial y}\right)^2}{res} + \frac{h_1}{thick}(T_0 - T)\right) + \int_{\partial\Omega_N} \bar{T}h_1(T_0 - T)$$

where $\partial\Omega_N$ denotes the part of the boundaries where Neumann conditions are imposed and the test function is chosen to vanish where Dirichlet conditions are satisfied by the solution.

- Select subdomain 1 from the **Domain selection** list
- Enter `-k1*(Tx_test*Tx+Ty_test*Ty)+T_test*((1/res)*(Vx^2+Vy^2)+(h1/thick)*(T0-T))` in the **weak** dialog field
- Enter `T_test*ro1*cp1*T_time` in the **dweak** dialog field
- Click the **Init** tab and type `T0` in the **Initial value** edit field
- Click **OK**



Subdomain Settings/w1

Equation: weak formulation

Domain selection: 1

Name: 1

□ Select by group

☑ Active in this domain

Weak terms ☑ Unlock

| Term | Value | Description |
|---|---|---|
| weak | -k1*(Tx*Tx_test+Ty*Ty_test)+ | Weak term |
| dweak | T_time*T_test*ro1*cp1 | Time-dep. weak term |
| constr | 0 | Constraint |

☑ On top

### *Boundary Mode*

We want to impose Neumann boundary conditions on all boundaries except 1 and 16 and Dirichlet conditions on those two. As earlier, we do this by first imposing Neumann boundary conditions on all boundaries and then modifying the boundary conditions applied to 1 and 16.

- Press the **Boundary Mode** button
- Select all boundaries using Ctrl+A
- Select **Boundary Settings** from the **Boundary** menu
- In the **weak** dialog field, enter `T_test*h1*(T0-T)`
- Enter `0` in the **constr** dialog field
- Select boundaries 1 and 16 from the **Domain selection** list
- Enter `0` in the **weak** field and `-T+T0` in the **constr** dialog field
- Click **OK**

### *DC Subdomain Mode*

Before we run the simulation, we need to finish specifying the two-way coupling by again setting the electric conductivity to `1/res` in the **Subdomain Setting**s of the electrical application mode.

- Select **1 Geom1: conductive media DC (dc)** from the Multiphysics menu to switch back to the electrical application mode
- In the **Subdomain Settings** (under the **Subdomain** menu) window, enter `1/res` instead of `sig` as the conductivity for subdomain 1
- Click OK

### *Solve Mode*

- Select **Parameters** in the **Solve** menu
- Press the **Time dependent** radio button in the **General** page
- Click the **Timestepping** tab and type `0:0.001:0.03` in the **Output times** edit field
- Press **OK**
- Press the **Solve** button in the main tool bar

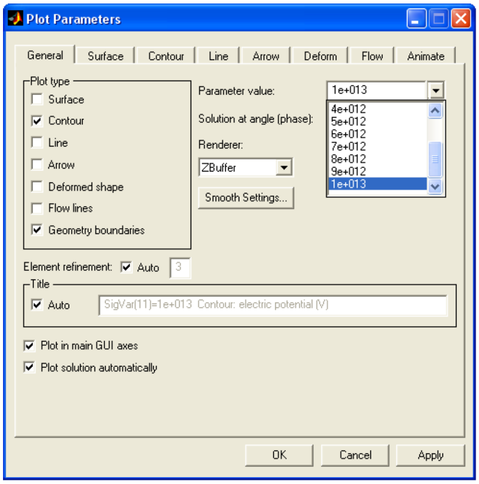
The solution is the same as earlier. Once again, you can check that the same integral as before is obtained for the temperature.

Save your model by selecting **Save** from the **File** menu. Alternatively, use Ctrl+S.

## *Parametric Study*

We conclude our study of this problem with a simple parametric study. In the original electric problem uncoupled to the heat transfer problem, we considered a uniform and constant conductivity `sig`. Now, we will consider a conductivity that is not uniform but instead varies in the form `sig+SigVar*x^2`. We will look at several values from `0` to `1e13` for the parameter SigVar.

- Open the file `FilmBasic.mat` from the **Open** item in the **File** menu
- Save as `FilmParam.mat` (under the **File** menu)
- On the **Subdomain Settings** page under the **Subdomain** menu, specify `sig+SigVar*x^2` for the conductivity in subdomain 1
- Go to the **General** tab on the **Solve**, **Parameters** window
- Select **Parametric** as the **Solver type**
- Under the **Parametric** tab, enter `SigVar` as the **Name of parameter** and `[0:1e12:1e13]` for the **List of parameter values**. This specifies that we will solve the problem for 11 equally-spaced values of `SigVar` between `0` and `1e13`.
- Press **Solve**
- In Post mode, you can plot the solution at any of the specified values of the parameter `SigVar`. Simply select that value from the **Parameter value** list under the **General** tab on the **Plot parameters** window.



Plot type
☐ Surface
☑ Contour
☐ Line
☐ Arrow
☐ Deformed shape
☐ Flow lines
☑ Geometry boundaries

Parameter value: 1e+013
4e+012
5e+012
6e+012
7e+012
8e+012
9e+012
1e+013

Solution at angle (phase):

Renderer: ZBuffer

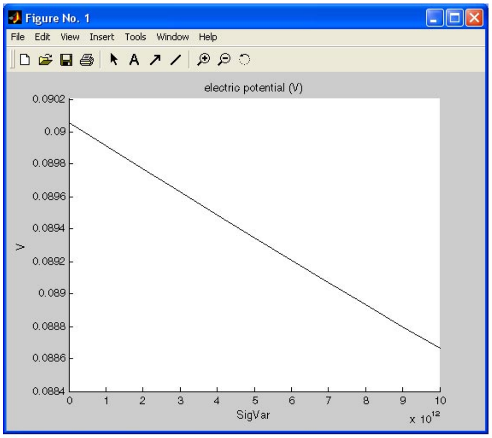
Element refinement: ☑ Auto 3

Title
☑ Auto SigVar(11)=1e+013 Contour: electric potential (V)

☑ Plot in main GUI axes
☑ Plot solution automatically

**Note:** *An other easy way to visually compare the solutions obtained with different values of the parameter consists in using the Animation feature in the Plot parameters window.*

Let us plot the voltage at the point of coordinates (5e-4; 1.25e-4). To do so, in the **Post** menu, select **Cross Section Plot Parameters** and under the **General** tab select all timesteps (using Ctrl+A). Under the **Point** tab, toggle **Point plot**, select `V` as the point expression and finally enter `5e-4` as the **X** coordinate of the point and `1.25e-4` as the **Y** coordinate. Click **OK**.



## *Note on command line scripting*

Whereas we have now created and run all the models above from the FEMLAB Graphical User Interface, it should be noted that in fact the same studies could have been performed entirely from the MATLAB command line, through scripting.

There are a number of reasons one might want to use scripting. For one, a script allows you to readily create several problems that are similar to each other in a minimal amount of time. Repetitive tasks can be automatized in a script. Also a script can include any Matlab command, including tests, loops, etc, which means you can base the definition of a problem on the output on another, automatically. Finally, certain functionalities that are not implemented in the GUI can be accessed only from the command line, or you may even add functionalities specific to your application.

For an example of such a script, in the GUI, select **Reset model m-file** from the **File** menu. This will generate a script that generates the input for the current model. Note that this m-file does not contain the mesh and solution.

For more information on the syntax of FEMLAB commands, it is always possible to type `help commandname` or `doc commandname` at the MATLAB prompt. Extensive information is also available from the FEMLAB Manuals, which you can access from the **FEMLAB Helpdesk** in the **Help** menu.

As a sample problem specified from the command line, type the following at the MATLAB prompt:

```
clear fem;
fem.geom=rect2;
fem.equ.c=1;
fem.equ.f=1;
fem.bnd.h=1;
fem.bnd.r=0;
fem.mesh=meshinit(fem);
fem.xmesh=meshextend(fem);
fem.sol=femlin(fem);
postsurf(fem,’u’)
```

These lines of code illustrate the solution of the equation $\nabla \bullet \nabla u = 1$ on the unit square with the boundary condition $u = 1$ on the boundary.