1

Minicourse on FEMLAB

This minicourse is an introduction to working with FEMLAB, a program for the solution of a large class of partial differential equations (PDE) from different fields of physics. The program has been written in a way that makes it easy to use, both for the engineer who is unaccustomed to PDE´s, as well as for the specialist in PDE´s.

The object of the course is to present three examples that will get the user started in using FEMLAB modeling. If you are new to FEMLAB, it is recommended that you start with exercise 1.

You need to download comp.dxf. Also, answers to the exercises are provided in the files eldet.m,comp.m and difflow.m

Contents

FEMLAB

FEMLAB Structural Mechanics

FEMLAB Electromagnetics

FEMLAB in Simulink

Exercise 1: An electronic Detector

Exercise 2: Mechanical Component Model

Exercise 3: Chemical Engineering

1

FEMLAB

FEMLAB solves Partial Differential Equations (PDE) that arise in engineering and scientific applications, using the finite element method.

Areas of physics where FEMLAB could be applied are e.g.

- electromagnetics
- heat flow
- chemistry
- diffusion
- fluid mechanics
- structural mechanics
- general physics
- wave propagation

The common property of these problems are that they can be described by partial differential equations on the forms

Table 1-1: Classical PDEs in compact and standard notation

Equation	Compact Notation	Standard Notation
Laplace's equation	$-\nabla \cdot (\nabla u) = 0$	$-\frac{\partial}{\partial x}\frac{\partial u}{\partial x}-\frac{\partial}{\partial y}\frac{\partial u}{\partial y} = 0$
Poisson's equation	$-\nabla \cdot (c\nabla u) = f$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$
Helmholtz's equation	$-\nabla \cdot (c\nabla u) + au = f$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)+au = f$
Heat equation	$d_a\frac{\partial u}{\partial t}-\nabla \cdot (c\nabla u) = f$	$d_a\frac{\partial u}{\partial t}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$

Table 1-1: Classical PDEs in compact and standard notation

Equation	Compact Notation	Standard Notation
Wave equation	$d_a\frac{\partial^2 u}{\partial t^2}-\nabla\cdot(c\nabla u)=f$	$d_a\frac{\partial^2 u}{\partial t^2}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)=f$
Schrödinger equation	$-\nabla\cdot(c\nabla u)+au=\lambda u$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)+au=\lambda u$
Convection-reaction equation	$d_a\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)+\beta\cdot\nabla u=f$	$d_a\frac{\partial u}{\partial t}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)+\beta_x\frac{\partial u}{\partial x}+\beta_y\frac{\partial u}{\partial y}=f$

These PDE´s are solved in two dimensions. The boundary conditions on edges of domains are prescribed values of the solution and its derivatives, i. e. Dirichlet and Generalized Neumann conditions. Boundary conditions can also be applied to inner edges. An additional feature is the possibility to specify periodic boundary conditions.

To facilitate for the user, different physics modes have been created. In these the PDE and the boundary conditions are formulated in a way that is customary in different scientific areas. There are still modes for the mathematical formulation of PDE´s.

It is possible to solve coupled systems of PDE´s. Actually, FEMLAB has a special feature to make this easy and intuitive, which is called *multiphysics*. Using this, different modes can be added in the GUI and FEMLAB automatically sets up the set of coupled PDE´s. In the third exercise you will learn how to use the multiphysics feature to set up a coupled system of PDE.

Most problems are solved in a way, where the basic PDE´s are put on a form, suited for linear or almost linear problems. We call this the *coefficient form*. For strongly non-linear PDE´s, however, it is better to use a formulation, called the *general form*. You will encounter this latter formulation in the last exercise.

The finite element mesh is generated using a Delaunay algorithm in order to ensure compatibility with the geometry, while still keeping the finite element angles as large as possible. This increases accuracy and the speed of the adaptive solver. You can control the parameter of the mesh directly. Adjustable parameter are e.g. the global element size and the element size on an edge.

Using the adaptive mesh solver, the mesh is refined during the solution in areas where some error estimate is too large. The different solvers can handle linear, nonlinear, as well as eigenfrequency problems.

The logic of modelling in FEMLAB follows the natural steps for solving PDE´s, using the finite element method.

- Draw the domain geometry
- Specify boundary conditions
- Define the PDE´s, either by specifying physical parameters in the physics modes, or by inserting coefficients in mathematical PDE formulations.
- Generate a mesh
- Discretize the equations and solve on the mesh.

In the Graphical User Interface (GUI), FEMLAB lets you navigate easily back and forth among these steps. A set of CAD tools are provided to draw the geometry of the model, and a set of postprocessing functions offer a wide range of possibilities to analyze the results. Apart from plotting the results over the geometry, data can be obtained along curves. Integrals, both over the geometry and along lines, can also be computed. You will get acquainted to modeling in the GUI by doing the exercises below.

When modeling in the GUI, it is still possible to use matlab functions, by inserting their names in e.g. the fields for the specification of PDE coefficients.

It is also possible to use FEMLAB directly in the Matlab environment, using the special FEMLAB commands. This is not covered in this minicourse. But if you save your model as an m-file, you can see the different commands used for creating and solving the model. Command line modelling allows the user to do parameter studies on their model as Matlab programs. The *API*, furthermore, makes it possible for the user to customize the GUI, to incorporate this parameter study. For all this, we refer to the FEMLAB manuals

FEMLAB Structural Mechanics

Structural mechanics can be modeled in the basic FEMLAB program. FEMLAB Structural Mechanics[1], however, extends the capability of FEMLAB in this area. It provides several additional features of importance. Application modes are

- Plane stress problems
- Plane strain problems
- Axial symmetric problems
- Kirchhoff plate problems
- Mindlin plate problems

The first three types of application modes are basically two-dimensional. All displacements and loads are in the geometrical plane. The two plate modes, on the other hand, allows for the specification of some three-dimensional properties in a two-dimensional model. For these modes, it is possible to specify torsion and load directed out of the plane.

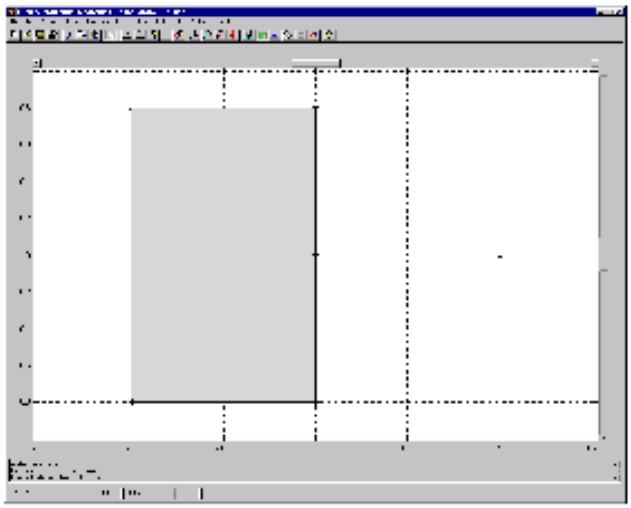
For the above modes, the models can be static or time-dependent. Analysis of eigenfrequency and frequency response can be made, using the corresponding solvers.

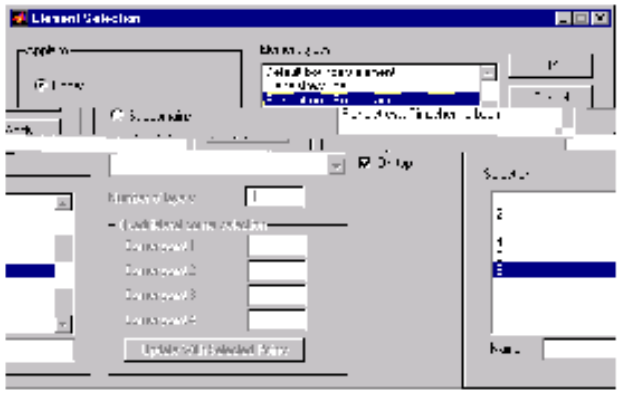
Bars and beams can be modeled together with two-dimensional objects in the plane stress mode and in the plate modes. A beam type element is a combination of a bending element and a bar element.

To create a beam, draw a straight line in the **Draw** mode and then open the **Specify Element** dialog box, in the **Elements** menu. Check the **Edges** radio button and select the appropriate beam element type. In the example below, a beam is to be attached to a rectangle. By default, the beam is fixed to another geometry object.

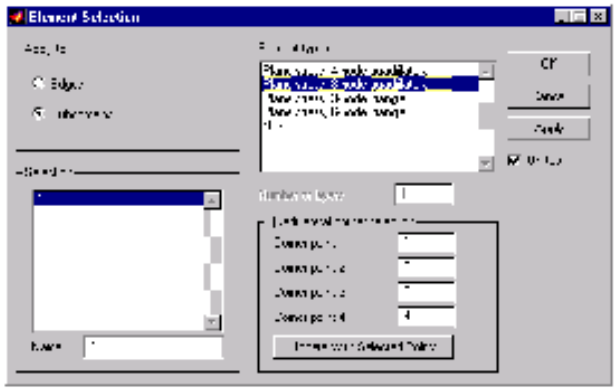
1. In the navigator, denoted Structural Mechanics Module.

However, you can disconnect one or several degrees of freedom, using the **Disconnect Edges** dialog box, also found in the **Elements** menu.



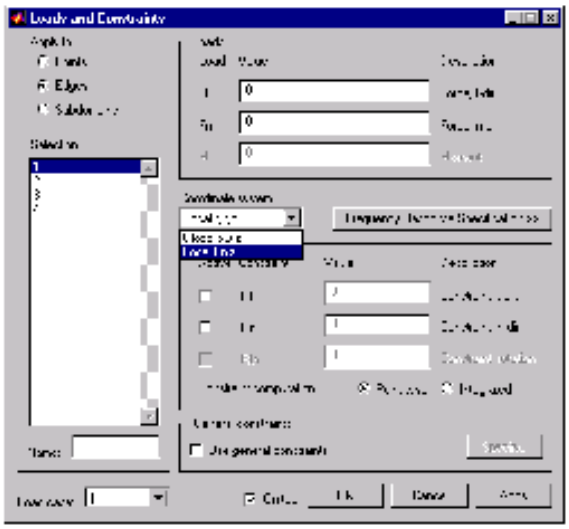


For the two-dimensional subdomain, you can specify different types of finite elements. Higher order elements, like 8-node quadrilaterals and 6-node triangles are available.



In FEMLAB Structural Mechanics you can use a local coordinate system when setting loads and constraints

Using this property, it is easy to set a normal load on an edge with a complex shape.



Other valuable properties of FEMLAB Structural Mechanics are

- The material library
- The possibility of specifying models with plasticity
- The possibility of modeling dynamic problems with damping.

1

FEMLAB Electromagnetics

FEMLAB Electromagnetics[2] is a collection of application modes, customized for engineers and physicists working within the field of electromagnetics.

The types of problem that can be solved, can be classified into

- In-plane quasi-statics
- Axisymmetric quasi-static
- In-plane waves
- Axisymmetric waves
- Perpendicular waves

For these problems, static, time-harmonic and time-dependent models can be specified.

Of the above, the first two types use similar approaches for simulating electromagnetic phenomena. They only differ in the coordinate system used for describing the geometric structures. In the same manner the last three types form a group with common simulation strategies. The last type describes 3-D problems with a harmonic propagation in one direction. These can be simplified to 2-D simulations.

The difference between the two groups is that the design of the models depend on the *electrical size* of the structure. The electrical size is a dimension-less measure given as the ratio between the largest distance between two points in the structure divided by the wavelength of the electromagnetic fields. This distinction is based on whether the retarded field[3] has to be taken into account or not. The latter case is often referred to as the *quasi-static approximation*.

Quasi-static models are suitable for simulations of structures with electrical sizes of up to 1/10.

When the variations in time of the sources of the electromagnetic fields are more rapid, the full Maxwell models have to be used. These models are appropriate for structures of electrical size 1/100 and larger. This means that there is an overlapping range where quasi-static or full Maxwell models can be used equally well.

For analysis of electro-mechanical devices, such as linear and rotary motors, the computation of the forces and force distribution is important. It turns out that the

2. In the navigator, denoted Electromagnetics Module.

3. Cheng, D. K. *Field and Wave Electromagnetic*, 1989, Addison-Wesley, Reading.

latter is difficult. To enable the calculation of force distribution, FEMLAB Electromagnetics has a special function, based on the principle of virtual work for magnetic energy.

FEMLAB Electromagnetics can handle

- Inhomogeneous materials
- Anisotropic materials
- Nonlinear materials
- Frequency dispersive materials.

1

FEMLAB in Simulink

Simulink is a Matlab product for simulation of dynamical systems. It provides you with a graphical user interface, enabling the building of models from a number of connected blocks, each representing an operation performed on a data flow. Simulink is used, e. g., for validation of control systems, signal processing and optimization.

A FEMLAB model can be exported as a *FEMLAB Simulink structure*. This structure provides an interface between Simulink and FEMLAB. Thus, the model can be incorporated as a block in a Simulink model.

It should always be considered, whether the FEMLAB model makes it possible to use a FEMLAB Simulink structure, that does not call FEMLAB during the simulation. Thus, a faster Simulink model can be obtained for a linear FEMLAB model, if the FEMLAB model has been exported as a *state-space* model. Also, if the time-scale of the FEMLAB model is significantly shorter than the time scale of the Simulink model, a *static* model of FEMLAB should be exported.

The input data to the FEMLAB Simulink structure is passed using variables. These variables are the ones specified in the **Add/Edit Variables** dialog box, and not the dependent variables. Simulink cannot handle dirichlet boundary conditions that depends on input variables from Simulink.

Output data can be on different forms, all derived from dependent variables.

- The solution or solution component at a given node in the mesh.
- An expression evaluated at a given node.
- A user defined Matlab function
- A linear functional of the solution

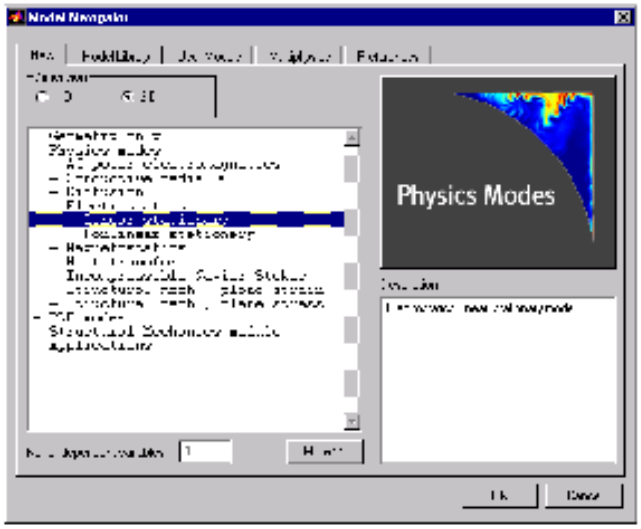
Exercise 1: An Electronic Detector

In this example you will calculate the electric field in a part of an electronic detector. The part of the detector that we will model consists of an array of electrodes (cathodes) between two electrodes at 100 V. This is described in more detail on the web site http://hsbpc1.ikf.physik.uni-frankfurt.de/detektor/Hand1/hand1.htm. Due to the symmetry property of the problem, we will only model a domain with three electrodes.

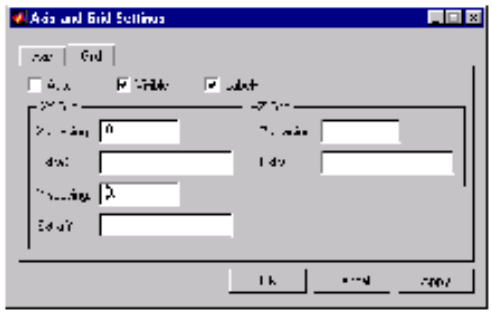
Start FEMLAB from the Matlab command window by typing

femlab

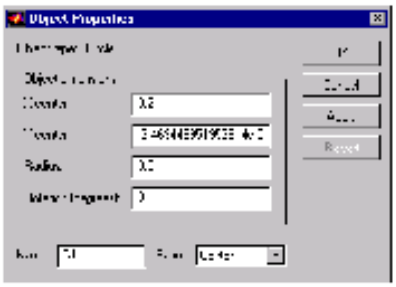
The **Model Navigator** now appears. If not, go to the **File** menu and select **New**. Select **Physics modes** and double click. Then choose **Electrostatics** and **Linear stationary**. Then press **OK**.



Select **Axis/Grid Settings** from the **Options** menu. In the **Grid** menu, deselect **Auto**. Write 0.1 in the fields **X spacing** and **Y spacing**. Click OK.



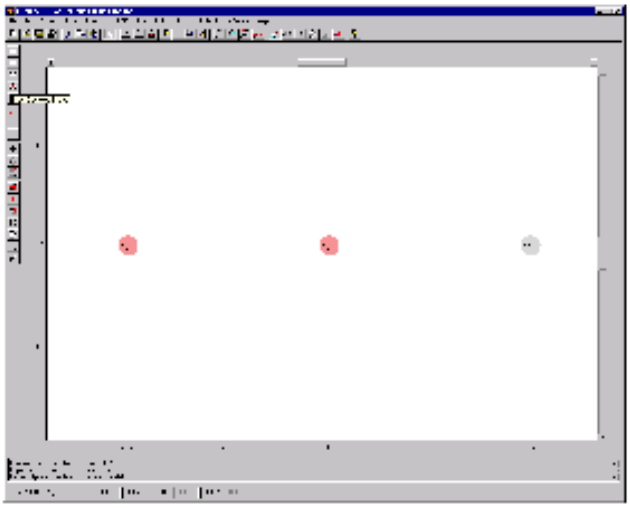
The model contains three circular holes of radius 0.01, and with centers at (-0.2,0), (0,0) and (0.2,0). See the picture on next page. In order to create circles for these; proceed in the following way. Select the **Draw Centered Ellipse** button. Right-click at (0.2,0) and drag, while holding down the right mouse button. When the circle have a radius of 0.1 (determined by the grid), release the button. Double-click on the circle. A dialog box appears where you can insert the radius 0.01. Leave with **OK**.



Select the circle and copy it from the **Edit** menu. When you use **Paste**, another dialog box appears. Insert the following numbers; x-axis displacement: -0.2, y-axis

displacement: 0 and **number of repeats**: 2. Use the zooming possibility provided

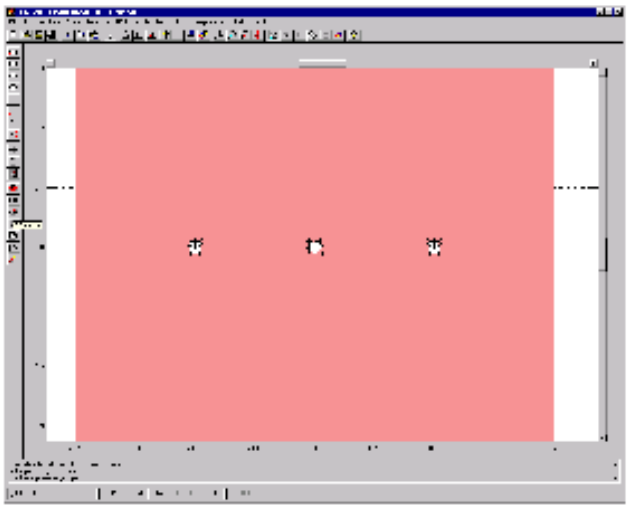
by buttons on the main (upper) toolbar to inspect the result closer. The three circles will now appear in the places where the holes in the figure on next page are situated.



Draw a rectangle by pressing the **Draw Rectangle** button (top), then clicking with the mouse on (-0.4, -1), dragging to (0.4,1), and finally releasing the button.

Select all objects (ctrl-a) and subtract the circles from the rectangle, using the difference button to the left (move the mouse over the buttons to inspect their functions).

You should now have obtained the following geometry.



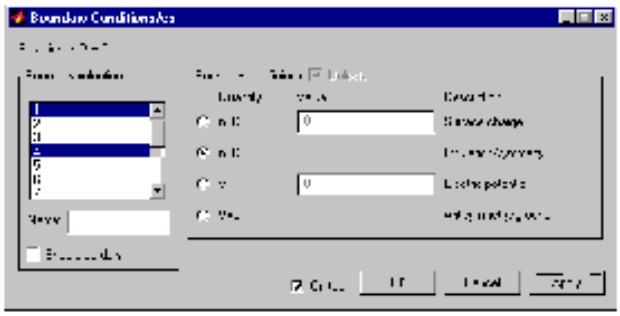
Enlarged part of the geometry

The small circles are three anodes at zero potential, while the upper and lower sides of the rectangle are electrodes held at a potential of 100V. On the vertical edges symmetry conditions are applied, since the real detector consists of an array of many electrodes. The space charge due to secondary electrons is totally insignificant. With this information we can specify the boundary conditions and the material properties in the PDE specification.

Press the **Boundary Condition** button $\delta\Omega$ on the upper toolbar. The boundaries are visible as arrows. Their default value is zero voltage, i.e. a Dirichlet condition, which is marked by solid arrows. Neumann- or Generalized Neumann conditions are shown as dashed arrows.

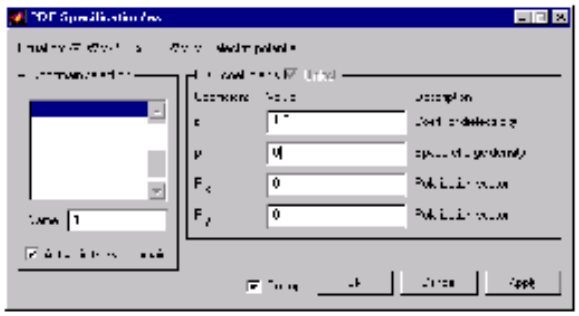
Open the **Specify Boundary Conditions** dialog box from the **Boundary** menu. Select among the numbers in the left table. When you select a number, the corresponding boundary becomes colored. You can select several boundaries simultaneously. You can also select boundaries in the dialog box, by selecting edges in the main GUI.

The three cathodes are already grounded by default. However, the upper and lower electrodes must be set to 100 V. The vertical edges should be set to **insulation/symmetry**. Notice that the boundary lines become dashed when this boundary condition is selected. This indicates that they are Neumann conditions.



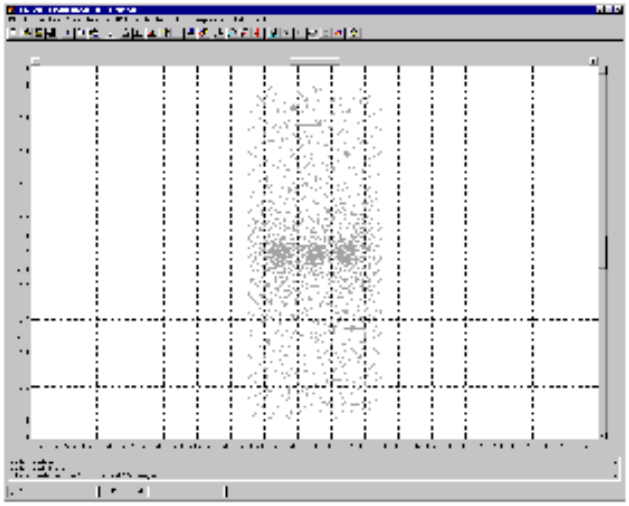
Press **OK**.

Next, open the **PDE Specification** dialog box in the **PDE** menu. Select domain 1 in the left table. We have vacuum and the electrons can be disregarded. Thus the space charge density is zero and the dielectric constant of no interest. Change **Space charge density** to zero.



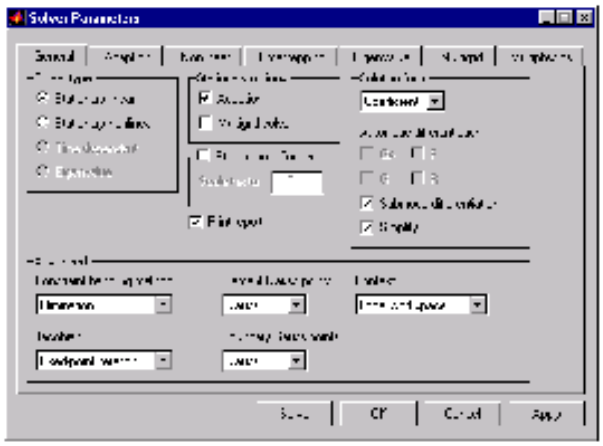
Press the **Mesh Mode** button to enter the mesh mode. A mesh is automatically generated. This initial mesh is the same as the one generated by the triangle button on the upper toolbar. If we would have wanted to refine the mesh, this could have

been done by pressing the **Refine Mesh** button. In our case, however, an adaptive mesh solver will be used to refine the mesh selectively during the solution.

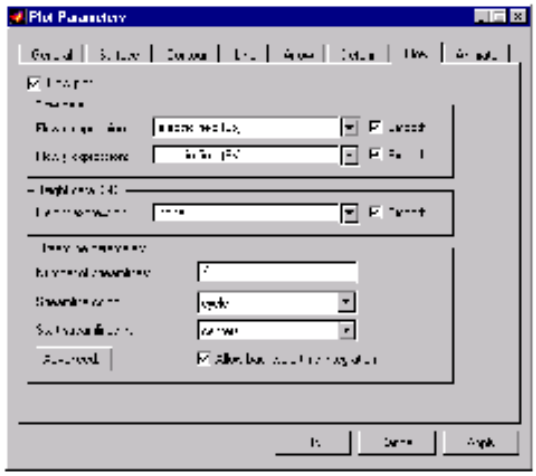


Change the solver parameters by pressing the **Solver Parameters** button. Alternatively, this can be done by selecting **Parameters** in the **Solver** menu. Check

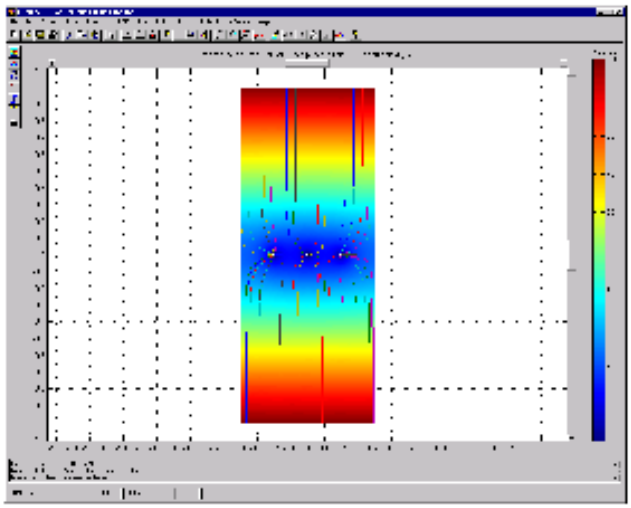
the **Adaption** box in the general page, and then solve the problem (the equality sign button).



In order to see the field lines in the plot, press the **Plot Parameters** button (a question mark on top of a Matlab symbol). Check the **Flow plot** check box on the **Flow** page. Then select electric field in the following way.



Note the change of the number of streamlines. You will get a plot similar to the one below.



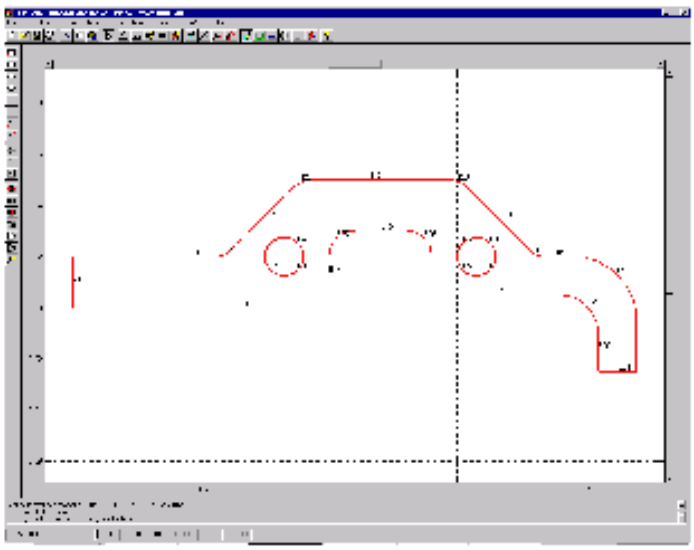
Exercise 2: Mechanical Component Model

For this example you need the file comp.dxf, which can be downloaded together with the rest of the course material.

In this exercise a plane stress model is created in the Structural Mechanics Engineering Module (SME). You can read more about this module in the introduction. We are going to look at different possibilities to analyse the properties of the model. These include static analysis, as well as eigen frequency and frequency response analysis.

Static Analysis

- Choose Plane Stress, Linear, Static in Structural Mechanics Module in the Model Navigator (select New from the File menu).



Options and Settings

You do not need any settings in this problem.

Draw Mode

- Import the DXF file comp.dxf using **Insert from File, DXF-file** on the **File** menu.

The DXF file is located in your course directory.

- First you should solidify the circles. Select all curve objects on a circle. This can be done, either by using the ctrl-key while clicking on each of the curve elements with the mouse, or by holding down the left mouse button and dragging it over the circle. Be sure that you only select curve objects of the circle. Coerce the selected object to a solid by pressing the **Coerce to Solid** button on the **Draw** toolbar (left, third from the bottom). Do this for both circles.
- Select all outer edge curve objects and coerce to a solid.
- You should now have three objects. Select them all (ctrl-a).
- Remove the circles by pressing the **Difference** button on the **Draw** toolbar.

Elements Mode

The imported component is drawn using meter as length scale so we use SI units for the material data.

- Specify elements according to the following table, using **Specify Elements** from the **Elements** menu.:

Subdomain	1
Element	Plane stress, 3-node triangle

- Enter material data according to the following table, using **Specify Material** from the **Elements** menu.:

Subdomain	1
E	2.1e11
nu	0.3
t	4e-3
rho	7.85e3

Edge	all
1	4e-3

Load Mode

Open the **Specify Loads and Constraints** dialog box under the **Load** menu. Edge 12 should have the load 15e6 in the x-direction. Notice the arrows that denote the load. Constrain edge 1 in the x- and y- directions, by checking **Rx=0** and **Ry=0** for edge 1. Leave the dialog box, using OK.

Mesh Mode

- Select **Parameters** from the **Mesh** menu.
- Enter 3e-3 as **Max edge size, general** in the **Mesh Parameters** dialog box.
- Select **Initialize mesh**, using e.g. the triangle button in the upper toolbar.

Solve Mode

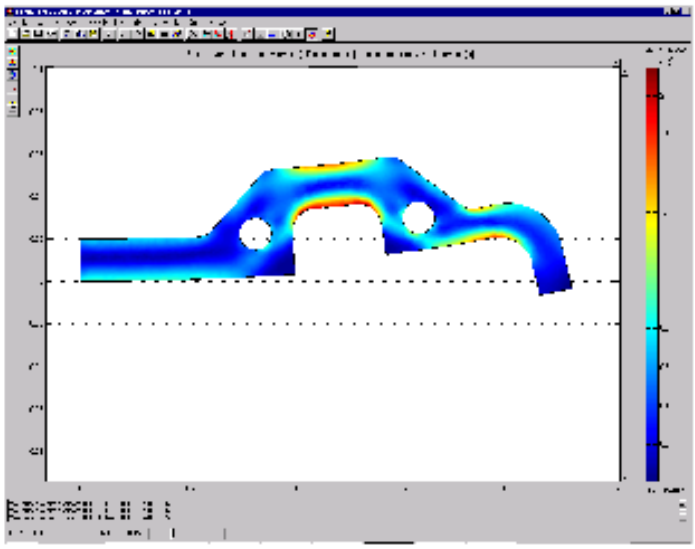
- Solve the problem by pressing the equal sign button.

Plot Mode

Plot the von Mises stresses and the deformed shape.

- Check **Deformed shape** and **Surface** in the **General** page in the **Plot Parameters** dialog box.

- Select von Mises stress (vonmises) as Surface expression, in the Surface page.

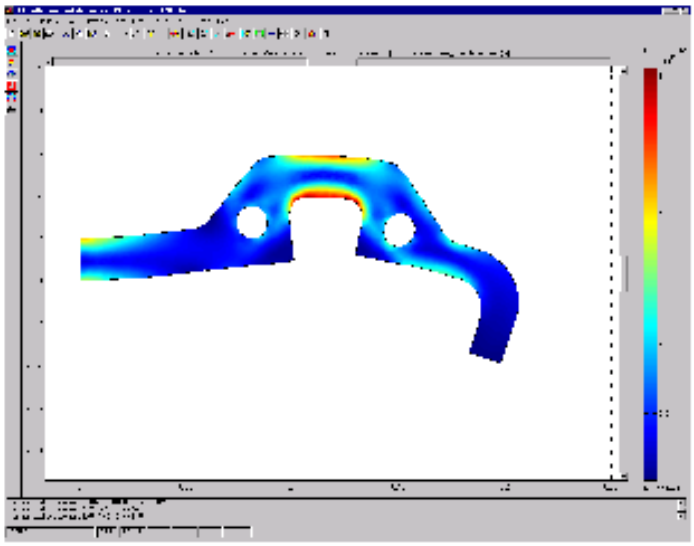


Eigenfrequency Analysis

Let us now study the eigenfrequencies of the component.

- Select Solver Parameters from the Solve menu and choose Eigenfrequency as Analysis on the general page.
- Solve the problem once more.

Go to the Plot Parameters dialog box and select the second eigenfrequency, at 1.4 kHz.



Frequency Response

In a frequency response analysis, the steady state response from harmonic loading of the model, is studied. The loads can be described as

$$F(t) = A(f)\sin(2\pi f + \gamma(f))$$

where the amplitude and phase can be functions of the excitation frequency f.

In the SME module, modal decomposition is used, and the user can specify the eigenfrequency to be used in the frequency response analysis. The damping can be specified individually.

The excitation frequency can be specified in a number of ways. In this example we will use the default settings, which means that we excite with frequencies spread around the eigenfrequencies.

- Open **Solver Parameters**.

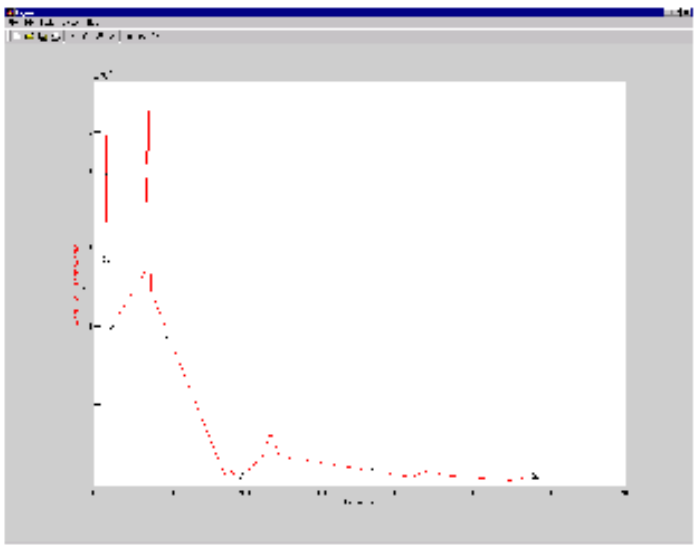
- Select **Frequency Response** as **Analysis** on the **General** page.
- Solve once more

Let us plot the result from an excitation with a frequency equal to the second eigenfrequency.

- Go to **Plot Parameters** and select the frequency 1.4 kHz.

Finally, we would like to view the deformation as a function of frequency in a x-y plot.

- First select **Labels**, **Show Point Labels** in the **Options** menu. We will study the x-deformation where the load is applied (point 27).
- Go to **Plot Parameters** and select the **Node** page. Insert 27 as the **Node number to plot** and press the **Node Plot** button. The amplitude of the response is shown in the figure below.



As expected, there is a peak in the displacement around the eigenfrequency, but the response is damped for higher frequencies.

Exercise 3: Chemical Engineering

Diffusion In Isothermal Laminar Flow Along a Flat Plate

Introduction

In the transport of chemical species in laminar flow, the flux time scale for diffusion compared to convection differs, in most cases, by several orders of magnitude. This can instructively be visualized by modeling these transports mechanisms in the simplest possible geometry. This is neatly shown in [1], where the diffusion in laminar flow along a soluble flat plate is treated analytically. However, the analytical solution requires a fairly large degree of simplification, where even this does not provide a straight-forward solution.

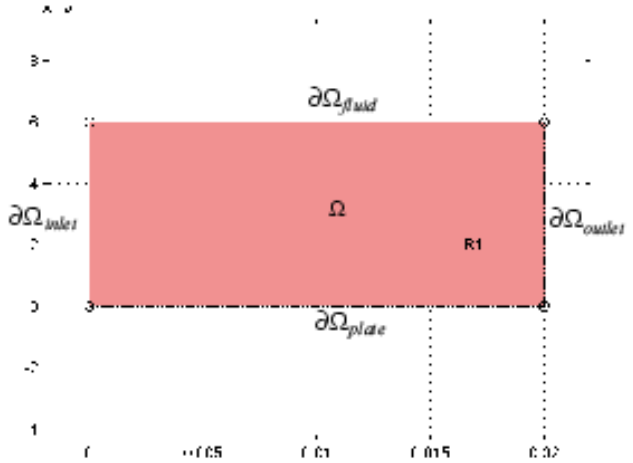
In this example, we will treat the same type of problem with a minimum of simplification. This simple model serves as an introduction to the modeling of systems where a mass balances is coupled to a momentum balance, and where the flux of dissolved species in a fluid is given by diffusion and convection.

We will study the concentration and flow distributions along a flat plate in a parallel channel. We will assume that a constant flux of a dissolved species, perpendicular to the surface, is generated along the flat plate. The solution will generate the developing structures of the viscous and diffusion layers in the parallel channel.

Definition of the problem

We treat the problem with one diffusing species, dissolved in water at room temperature. The geometry of the domain in this model is the simplest possible; a rectangle of 6 times 20 mm, see the below figure. The fluid inlet to our system is situated at the left boundary and the outlet at the right. The plate is represented by the lower boundary and symmetry is assumed at the top. The diffusing species is produced at the surface of the plate.

We use the Navier-Stokes equations, in combination with the continuity equation and a mass balance equation, for one dissolved chemical species. The transport of this species takes place by diffusion and convection. We further assume that the production of the diffusing species, at the surface of the plate, does not influence the viscosity and density of the fluid.



The assumptions listed above result in the following equations, for the momentum balance in the domain:

$$\mu\nabla^2\mathbf{v} + \rho(\mathbf{v}\cdot\nabla)\mathbf{v} + \nabla p = 0 \qquad in\ \Omega$$

$$\nabla\cdot\mathbf{v} = 0 \qquad in\ \Omega$$

where μ denotes the dynamic viscosity (kg m^{-1} s^{-1}), $\mathbf{v}$ the velocity vector (m s^{-1}), ρ the density of the fluid (kg m^{-3}) and p the pressure (Pa).

We solve these equations of motion coupled to the mass balance equations:

$$\nabla\cdot(-D\nabla c + c\mathbf{v}) = 0 \qquad in\ \Omega$$

where D denotes the diffusion coefficient (m^{2}s^{-1}) and c the concentration (mol m^{-3}). We can see that the flux vector, in the brackets on the left hand side of the equation, has a diffusive and a convective contribution.

We obtain the boundary conditions for the equations of motion by assuming a uniform velocity profile at the inlet and constant pressure at the outlet. Furthermore,

we assume symmetry along the boundary towards the free fluid. These assumptions give the following boundary conditions for the equations of motion:

$$\mathbf{v} \cdot \mathbf{n} = v_0 \qquad at\ \partial\Omega_{inlet}$$

$$\mathbf{v} \cdot \mathbf{n} = 0 \qquad at\ \partial\Omega_{fluid}$$

$$\mathbf{v} = (0, 0) \qquad at\ \partial\Omega_{plate}$$

$$p = 0 \qquad at\ \partial\Omega_{outlet}$$

We additionally obtain the corresponding boundary conditions, for the mass balance equations, by assuming that the concentration is known at the inlet and at the symmetry boundaries. We also know the production rate of the diffusing species at the surface of the plate, and assume that the dominating transport process in the direction of the flow, at the outlet, is transport by convection. This can be formulated by the following equations:

$$c = c_0 \qquad at\ \partial\Omega_{inlet}$$

$$c = c_0 \qquad at\ \partial\Omega_{fluid}$$

$$(-D\nabla c + c\mathbf{v}) \cdot \mathbf{n} = k \qquad at\ \partial\Omega_{plate}$$

$$(-D\nabla c + c\mathbf{v}) \cdot \mathbf{n} = (c\mathbf{v}) \cdot \mathbf{n} \qquad at\ \partial\Omega_{outlet}$$

The condition that determines the concentration at the symmetry boundary is only valid in the case when the diffusion layer does not reach this boundary. This assumption will be validated or falsified in the solution.

Solving the problem using the Graphical User Interface

Select Incompressible Navier-Stokes, from the Multiphysics menu in the Model Navigator, and add it as an application mode by moving it to the right field with the arrow button. Press **OK**.

Options and Settings

- Open **Axis/Grid Settings** in the **Options** menu. Unselect **Axes Equal** and set the axis settings according to the table below: Then, go to the **Grid** dialog box and set the grid values according to the table. You have to uncheck the **Auto** checkbox. Press **OK**.

Axis		Grid	
X min	-0.001	X spacing	0.005
X max	0.021		
Y min	-0.002	Y spacing	0.002
Y max	0.008		

- Enter the following variable names, for later use, in the **Add/Edit Variables** window, in the **Options** menu. Then press **OK**

Name	Expression
rho	1e3
miu	1e-3
D1	5e-9
flux	5.2e-2
vo	0.01
clo	0

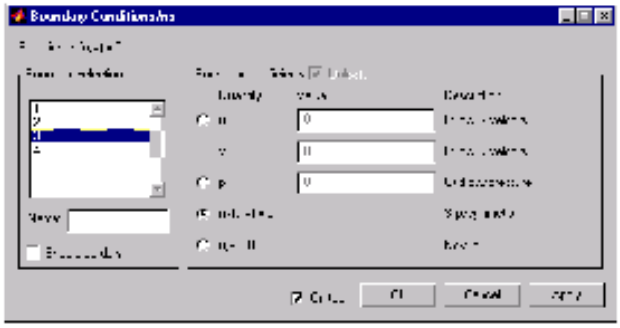
Draw Mode

- Press the **Draw Rectangle** button in the **Draw** toolbar. Press the left mouse button at the position (0,0) and draw to (0.02,0.006). Then release the mouse button. You have made a rectangle with the name R1.

Boundary Mode

- Select **Specify Boundary Conditions** from the **Boundary** menu. Enter boundary coefficients according to the following table. Then press **OK**

Boundary	1	2	3	4
Type	Inflow	No-slip	Slip	Outflow
u	v0			
v	0			
p				0



PDE Mode

We can start by defining the coefficients for the Navier-Stokes equations, which are the density and viscosity of the fluid, in this case water.

- Select **PDE Specifications** from the **PDE** menu. Enter the PDE coefficients, in subdomain 1, according to the following table. Then press **OK**

Subdomain	1
ρ	rho
η	mi u

We are now ready to define the mass transport equations for the diffusing species that is being produced at the surface of the plate. We do this by adding a new model equation in the Multiphysics Mode.

Multiphysics Mode

- Select **Add/Edit Modes** in the **Multiphysics** menu. Choose **PDE, general form**, label your application mode with the name, mass bal, and your Dependent variable, c1. This is done in the two bottom fields, before moving the selected mode to the right.

- Add the new application by moving it to the right field with the arrow button. Press **OK**. You should now be in the **1 variable general form PDE (massbal) mode**, as you can see at the top of the main GUI window. Check this by selecting the **Multiphysics** menu.

Boundary Mode

- Enter boundary coefficients according to the following table. Press **OK**.

Boundary	1,3	2	4
G	0	flux	-c1.*n
R	-c1+c1o	0	0

PDE Mode

We define the flux vector, in the **PDE Mode**, as an expression that consists of a diffusion and a convection term.

- Enter the PDE coefficients according to the following table. Note that Γ is a vector. Thus, there is a space in the middle of the expression. Press **OK**.

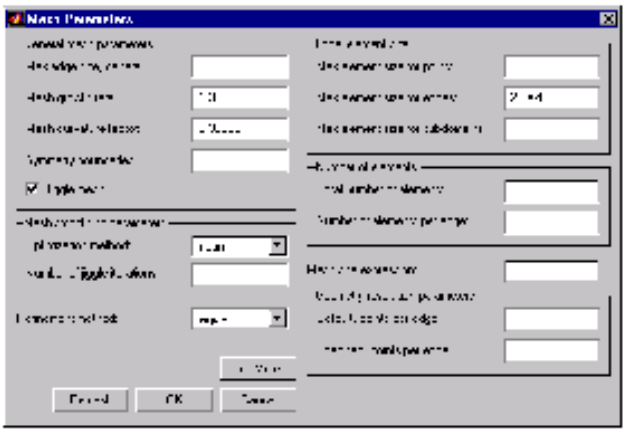
Subdomain	1
Γ	-D1.*c1x+c1.*u -D1.*c1y+c1.*v
F	0
d_a	0

Mesh Mode

This example requires a fairly dense mesh since the Reynolds number is relatively high. For that purpose, we define a maximum element size for the edge that represents the flat plate. This requires that we identify the edge number of our geometry. Select **Option, Labels, Show Edge Labels** in **Draw Mode**. In **Mesh Mode**:

- Choose **Parameters** and press the button labeled **More**.

- In the field **Max element size for edges**, set the maximum element size for edge 2 to 1e-4. This done by inserting "2 1e-4" in the field.

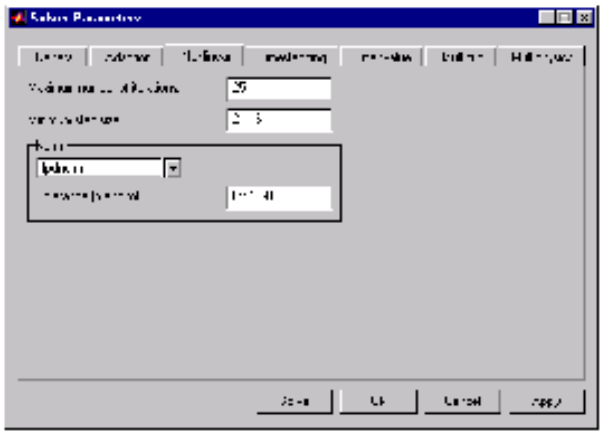


- Press **Remesh** and then **OK**.
- Refine the mesh once, using the **Refine Mesh**, button on the main toolbar, or by selecting **Refine Mesh** in the **Mesh** menu.

Solve Mode

- Select **Solver Parameters** on the main toolbar or in the **Solve** menu. Check that the **Stationary nonlinear solver** is used. **Streamline diffusion** should be off.

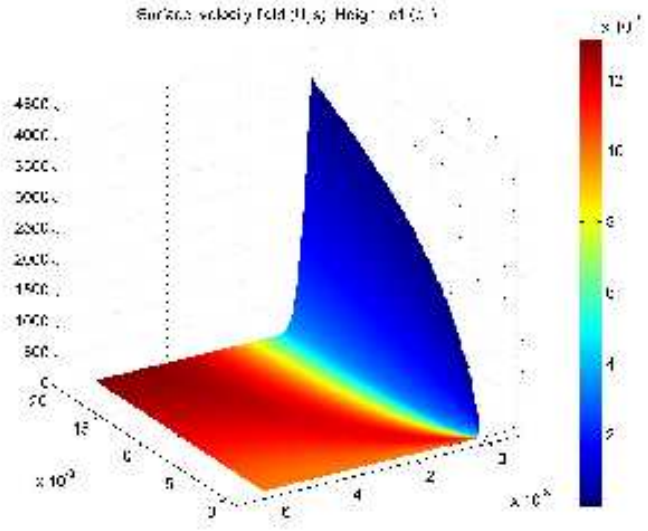
- Set the Tolerance to 1e- 8. This is done by selecting the dialog box **Nonlinear**, and inserting 1e-8 instead of 1e-4 in the bottom field. Solve the problem by pressing the **Solve** button. The solution takes a couple of minutes.



Plot Mode

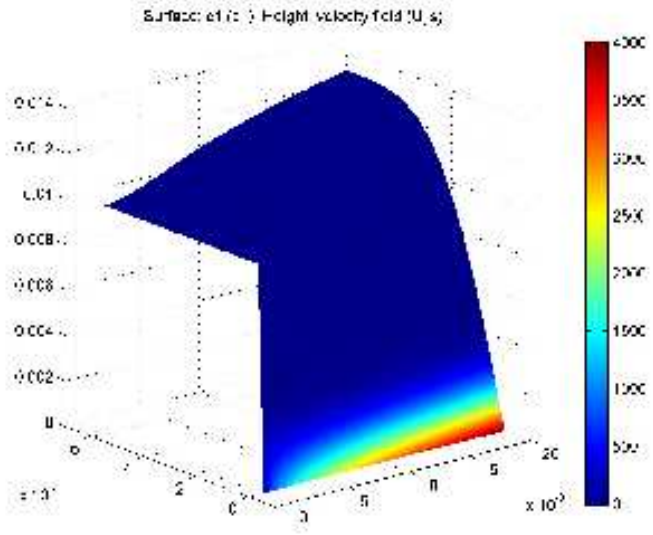
The default plot shows the concentration of the reacting species in the solution (in mole m^{-3}). The most interesting result from this simulation is a comparison between the thickness of the viscous layer and of the diffusion boundary layer, which is often given in the Schmidt number (Sc). We can obtain a notion of this relation by plotting the results in a 3-D surface plot. To do this, choose to plot the velocity field as

Surface expression, and the concentration as Height expression in the Plot Parameters dialog window. After rotating we obtain the resulting plot:



The difference between the viscous and diffusion layers can be clearly seen, in the figure above, by the amount that they extend into the fluid. This difference can be seen even more clearly if we reverse the plotting instructions. In order to do this, plot

the concentration as Surface expression and the velocity field as Height expression in the Plot Parameters dialog window. Rotate freely to generate a suitable view.



In addition, we can see in the above figure that we have almost a fully developed laminar flow, at the outlet of the domain, which supports our assumption of a constant pressure along this boundary.

References

[1] R. Bird, W. Stewart and E. Lightfoot, *"Transport Phenomena"*, John Wiley & Sons, New York, 1960.