# Assignment in
# The Finite Element Method, 2020

Division of Solid Mechanics

The task is to write a finite element program and then use the program to analyze temperature and stress distributions in an integrated circuit. The problem should be solved with Matlab along with suitable subroutines included in the CALFEM toolbox. A well structured report that presents the findings should be written.

## Problem description

Due to the recent COVID-19 outbreak, people have been forced to stay in quarantine in their homes. As a result, Netflix has emerged as a popular mean to relive the boredom. Netflix uses large data and server centers around the world to store and distribute their content. Somewhere in one of their centers, we find an integrated circuit. One major issue when designing electronics is how to deal with the heat that is generated when electric current passes though the component, and to model the heat generation, a finite element analysis can be performed.

To limit the demand on the internet during the crisis, the European Union asked Netflix to reduce their streaming quality. We assume that the heat, $Q$, that is generated in the integrated circuit due to the internal resistance is proportional to the data consumption, i.e.

$$Q \propto \text{data} \tag{1}$$

The heat generation is assumed to take place inside the die in the middle of the integrated circuit. The die is a small block of semiconducting material on which the electronic circuit is fabricated.

Our simplified model of the integrated circuit consists of three different components made from different materials. The package is made from silver (Ag) filled epoxy, the die is made from silicon (Si) and the conducting frame is made from copper (Cu). A cross section of the integrated circuit that should be analyzed is depicted in Fig. 1. Due to the symmetry of the cross-section, only half of the circuit needs to be analyzed.

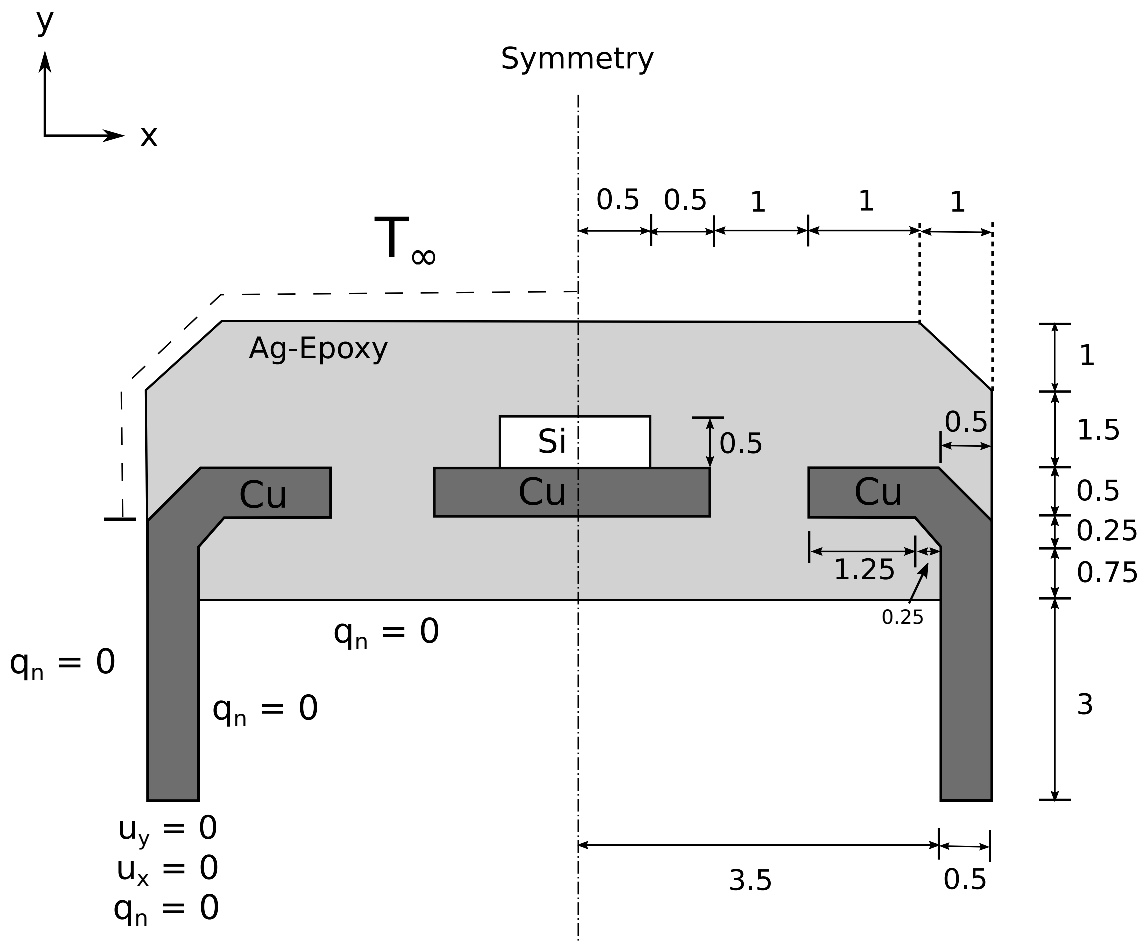


Figure 1: Sketch of a cross section of the integrated circuit. The boundary conditions and dimensions are symmetric around the symmetry line. All dimensions are in mm.

The material data are provided in Table 1.

| | Ag-epoxy | Silicon | Copper |
|---|---|---|---|
| Young's modulus, $E$ [GPa] | 7 | 165 | 128 |
| Poisson's ratio, $\nu$ [-] | 0.3 | 0.22 | 0.36 |
| Expansion coefficient, $\alpha$ $[1/K]$ | $4 \cdot 10^{-5}$ | $2.6 \cdot 10^{-6}$ | $17.6 \cdot 10^{-6}$ |
| Density, $\rho$ [kg/m$^3$] | 2500 | 2530 | 8930 |
| Specific heat, $c_p$ [J/(kg K)] | 1000 | 703 | 386 |
| Thermal conductivity, $k$ [W/(m K)] | 5 | 149 | 385 |

Table 1: Material data

The integrated circuit protrudes 10 mm out-of-plane (i.e. thickness). It is mounted on a PCB-board such that the bottom boundaries of the conductive frame are fixated in the $x$ and $y$-direction, i.e. $u_x = 0$ and $u_y = 0$. Fig. 1 illustrates the boundary conditions. Plane strain conditions are assumed to hold. Due to a fan placed above the circuit, Newton convection is present along the top boundary, cf. Fig. 1, i.e., $q_n = \alpha_c(T - T_\infty)$, where $\alpha_c = 40$ W/(m$^2$ K). The rest of the boundaries are thermally insulated. The initial temperature of the circuit is $T_0 = 30$ °C.

## Problem formulation

a) Find the stationary temperature distribution when we assume the that the integrated circuit is in an environment with $T_\infty = 18$ °C and the heat generated in the die is $Q = 5 \cdot 10^7$ W/m$^3$. The COVID-19 outbreak forced Netflix to decrease the video quality with 25%, which we assume is proportional to the data consumption. How will that affect the maximum temperature? Present the temperature distribution.

b) Determine how the temperature distribution changes with respect to time during the first 20 minutes after startup and then find the stationary temperature distribution, assuming $T_\infty = 18$ °C and $Q = 5 \times 10^7$ W/m$^3$. Present the temperature distribution at representative instants and at stationary conditions. What is the peak temperature for full quality and reduced quality?

c) Determine the effective von Mises stress field and the corresponding displacement field due to thermal expansion from the stationary temperature distribution when $T_\infty = 18$ °C and $Q = 5 \times 10^7$ W/m$^3$ is assumed. Investigate once again the effect of decreasing the video quality with 25%. Present the stress distributions and displacement fields. What are the peak stresses and where are they found?

**Hint:** The von Mises stress is defined as:

$$\sigma_{eff} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 - \sigma_{xx}\sigma_{yy} - \sigma_{xx}\sigma_{zz} - \sigma_{yy}\sigma_{zz} + 3\tau_{xy}^2 + 3\tau_{xz}^2 + 3\tau_{yz}^2}$$

## Procedure

A fully implicit time integration scheme should be used. Note that the element function for forming, $\boldsymbol{C}^e$, is available in Canvas. A suitable element is the linear triangular element. As a start point, you have the strong formulations of the heat and the mechanical problems.

The contour plots of the stress distribution are based on the stress at the nodal points. The stress of the nodal points can be approximated by taking the mean value of the stresses in the elements connected to a node. The following Matlab code can be useful:

```
for i=1:size(coord,1)
    [c0,c1]=find(edof(:,2:4)==i);
    Seff_nod(i,1)=sum(Seff_el(c0))/size(c0,1);
end
```

where `Seff_nod` and `Seff_el` is the von Mises effective stress at the nodal points and in the elements, respectively. Note that here the topology matrix (`edof`) is associated with the temperature problem.

## Report

A fundamental ingredient in all research is that it should be possible to regenerate the results obtained based on the report. In the present situation this implies that the appended matlab code should only be considered as supporting material. Moreover, note that one variable for grading the report is the structure of the computer code, i.e. you should choose suitable names for variables. A suitable structure for the report is:

- **Introduction**: Description of the problem, geometry and boundary conditions.
- **Procedure**: How the problems are solved (weak formulation, application of boundary conditions, treatment of thermal strains and so on). Derivation of the FE formulation and other important theoretical aspects should be included here. Note that you are encouraged to make references to textbooks. Though, it is important to carefully present all calculations that are not available in the literature.

- **Results**: Present the results in illustrative figures and/or tables. Note that the results should be commented such that the reader can not misunderstand the results (correct labels, units, figure texts etc.)
- **Discussion**: A discussion of the results, their meaning and significance. You might want to discuss sources of errors and accuracy in this section.
- **Computer Code**: Note that the code should be easy to follow and all declared variables should have intuitive names and be well commented.

A well structured should be handed to the Division of Solid Mechanics no later than **2020-05-26 16.00**. Matlab/CALFEM files (appendix) should be well commented. The reader of the report is assumed to have the same knowledge level as the author. If the report contains major programming or theoretical errors, the report is returned in order to be corrected. It is possible to obtain up to 5 points which are added to the points obtained at the exam in June 2020. The assignment should be approved no later than 2020-06-11. You should submit your report in **PDF** format to FHLF01@solid.lth.se or FHLF10@solid.lth.se. In addition to your report you should also attach your m-files in the email. Moreover, a paper version should also be handed in to the division of Solid Mechanics. Note that the bonus points obtained is only valid for the examination in June 2020.

### Collaboration

The task should be solved in groups of *two* or *individually*. For further details see www.solid.lth.se and navigate to the course homepage.

## 1 Generating mesh

In this assignment you should create your own mesh using the built-in PDEtool in MATLAB. This tutorial considers the example geometry seen in figure 2, which of course must be changed for your specific application.

PDEtool can be used directly to create simple geometries with little effort. To start just type `pdetool` into the MATLAB terminal to open PDEtool user interface. Although the tool is originally designed for solving partial differential equations only the meshing aspect are of interest in this assignment.

A step by step tutorial of how to use some of the basic features of PDEtool are provided here. Before starting to draw it is often convenient to activate grid and change the axis scales. This is done using <Options> menu. To build our geometry simple geometries are added one by one. The basic shapes used in PDEtool are rectangles, ellipses and polygons. Basic drawing tools can be found on the toolbar (see figure 3, note polygons are not necessary to create the geometry for the lab).

**Draw a rectangle:** Create a rectangle by using the rectangle tool (see figure 3) and simply hold left mouse button and move the mouse cursor to obtain the desired size. To adjust the size and position double click on the rectangle and a dialog will appear where coordinates and dimensions can be provided, see figure 4.

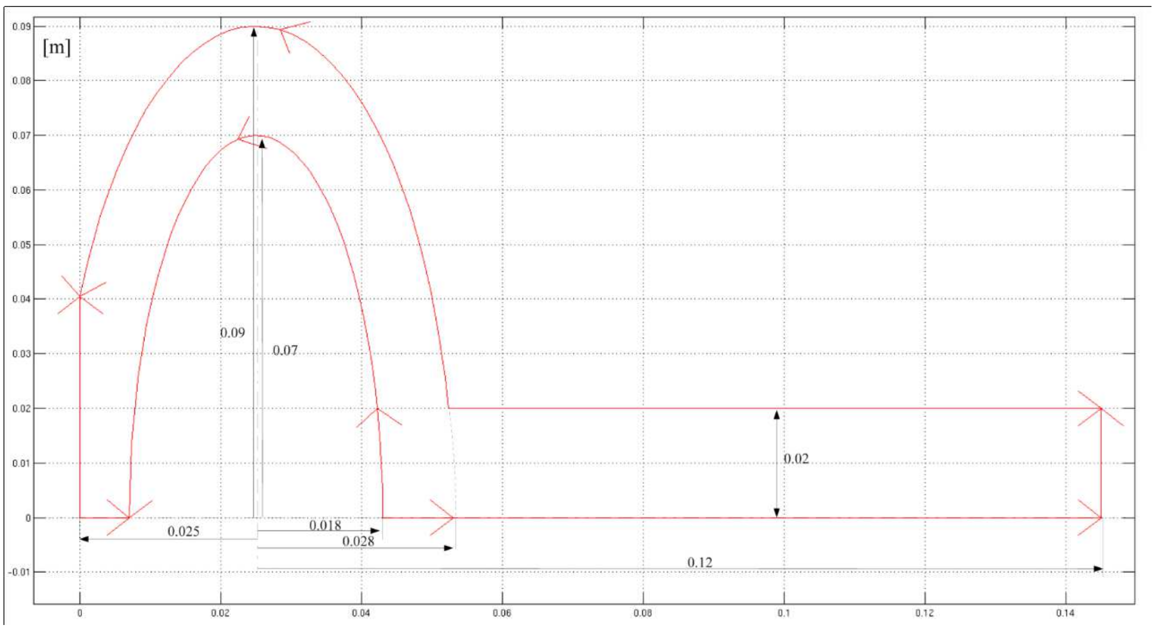


Figure 2: Dog toy geometry.

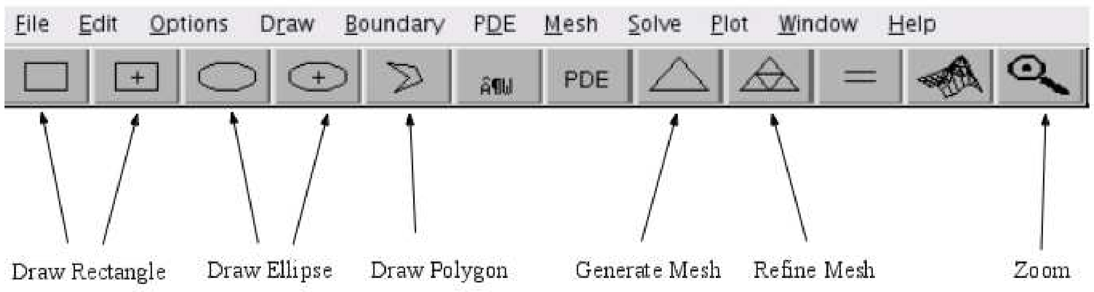
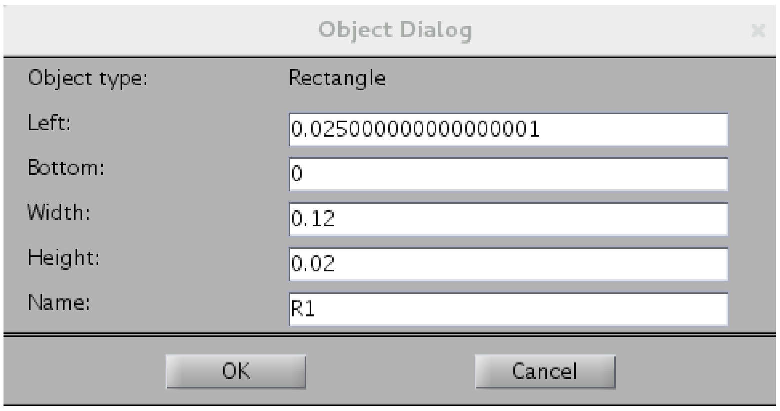


Figure 3: Basic tools for drawing in PDEtool.

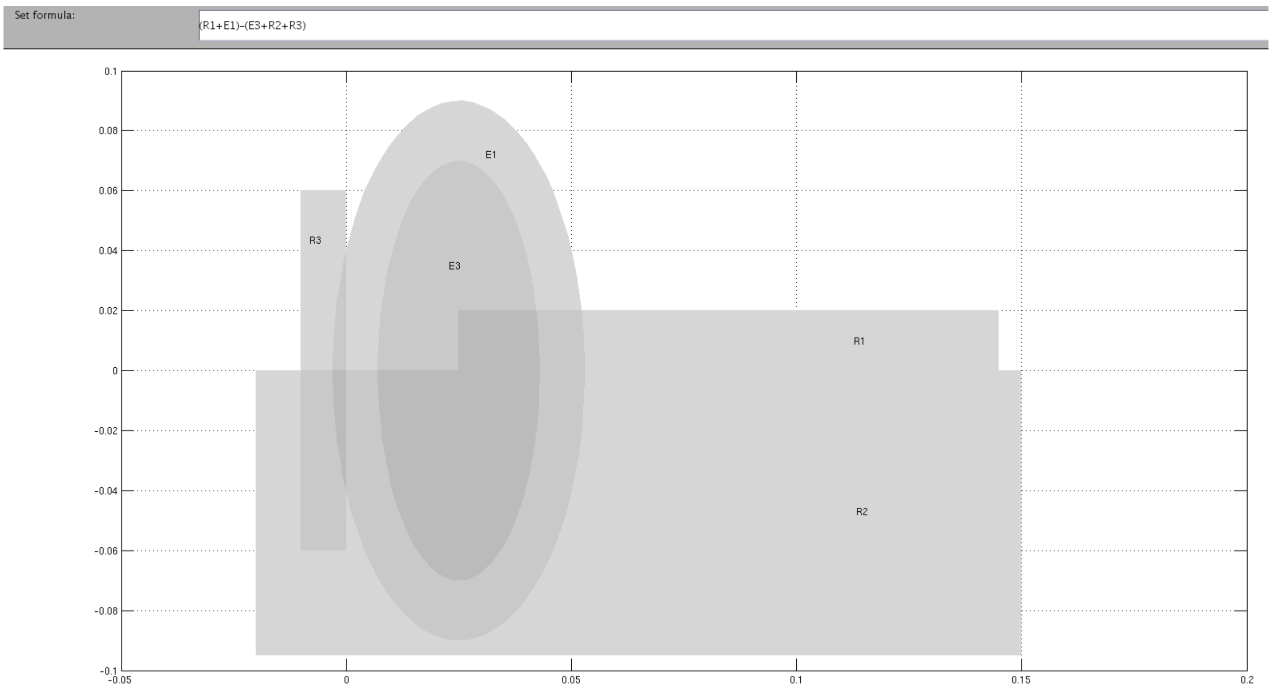


| | |
|---|---|
| Object type: | Rectangle |
| Left: | 0.025000000000000001 |
| Bottom: | 0 |
| Width: | 0.12 |
| Height: | 0.02 |
| Name: | R1 |

Figure 4: Rectangle object dialog.

**Draw an ellipse:** Create a ellipse by using the draw ellipse tool (see figure 3) and simply hold left mouse button and move the mouse cursor to obtain the desired size. Double click to adjust position and size in similar fashion as for rectangles.

**Determining combined geometry:** In the box below shape-buttons the present features are presented by name (which may be edited using object dialog). The plus sign indicates that geometries are added and minus signs indicate that they are cut out. Because of this the order is important i.e **R1+E1-E2 ≠ R1-E2+E1 in general**. Below figues showing drawn features and the resulting body after combining by grouping and changing signs in the set formula row (see figure 5, 7 and 6). Note that to see the resulting geometry we need to go to either meshing or boundary view (see toolbar).



Set formula: (R1+E1)-(E3+R2+R3)

Figure 5: Set of combined shapes.

When the necessary features are added go to <Boundary> / <Boundary mode> to see which are the boundaries of your geometry. The red arrows define the main borders and the grey contours represent the so called subdomains. For instance, subdomains appear where features overlap. To get a uniform mesh it is convenient to remove the subdomains. To do so use <Boundary> / <Remove all Subdomain Borders>. If the mesh consist of different natural domains such as different materials it is favourable to keep the subdomains.

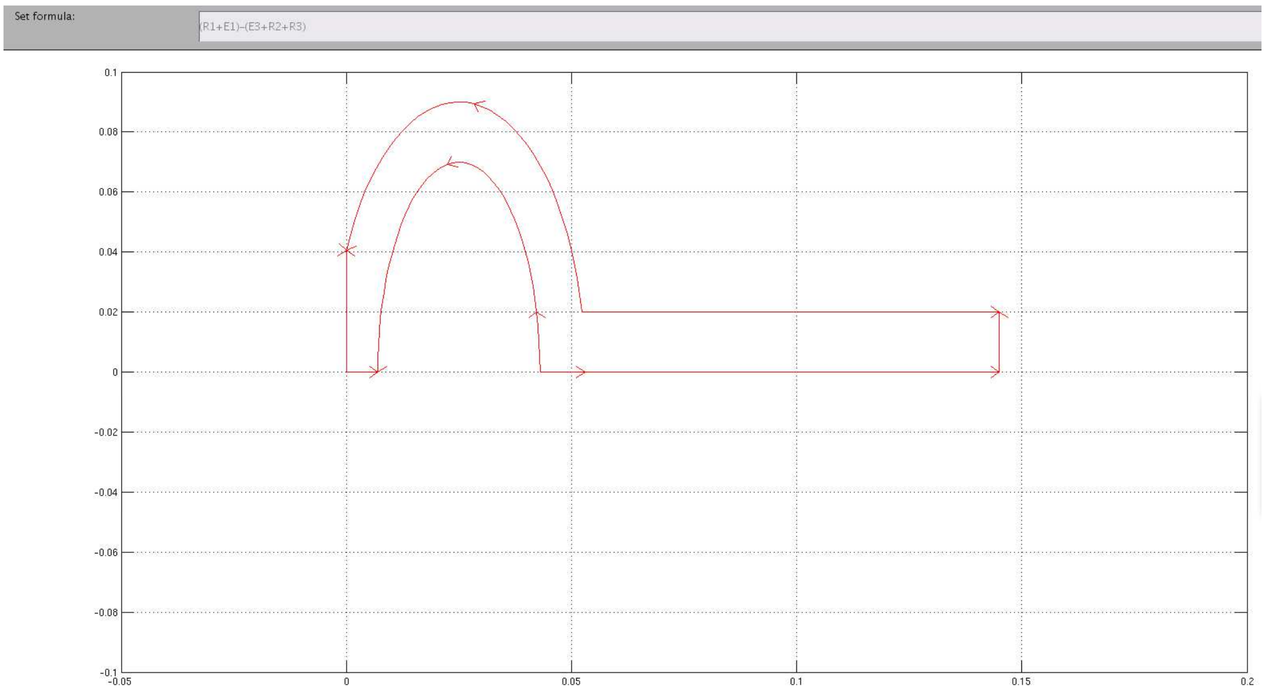
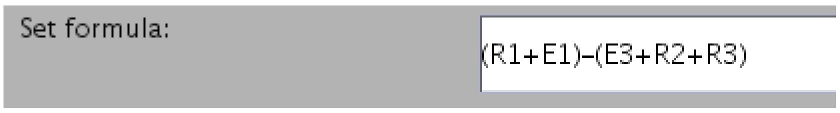


Figure 6: Resulting geometry seen in boundary mode (ctrl+B) after all subdomains have been removed.

To obtain the geometry provided in figure 6 the formula used is given in figure 7 and the object names is seen in figure 5.



Set formula: (R1+E1)-(E3+R2+R3)

Figure 7: Formula determining active areas.

To generate a mesh simply press mesh tool when the geometry seen in boundary mode is satisfactory (see figure 3). To refine the mesh, i.e. create more elements, press refine mesh until enough elements are generated (remember that finer mesh means higher computational cost but better accuracy). Note that a high number of elements increases the computational time.

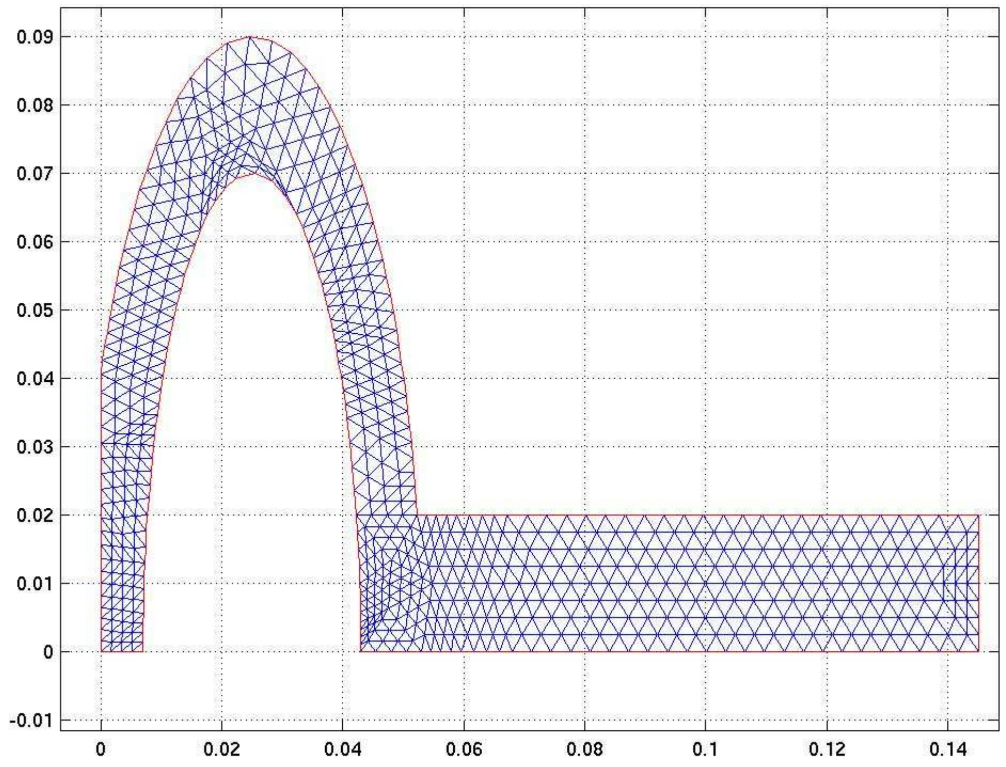


Figure 8: Mesh after two refinements.

When you are content with your mesh use <Mesh> / <Export mesh> to save the topology matrices associated with the current mesh (note you have to save the work separately to keep geometries etc. since <Export mesh> will only save matrices). The topology matrices are `p,e,t` (points, edges, triangles). The exported matrices will directly appear in the active MATLAB workspace. It is strongly advised to save them directly in a `.mat` file (mark variables in workspace, right click and save) so it can be loaded in the scrips you write. From `p,e,t` the CALFEM quantities `edof`, `dof`, `coord` etc can be extracted. On the course web page for FEM FAQ are some instructions of how this is done.

## 2 General tips

The output data from PDEtool is in the form `p,e,t` where `p` is *points*, it is a 2 row matrix where the first row is the x-coordinates and the second row is the y-coordinates. There are one column for each node i.e. in CALFEM notation `coord = p'`.

The output `t` is *triangles* and it has dimensions $4 \times \text{nelm}$. The three first rows are the node numbers associated with each element (three nodes for linear triangular elements) while the fourth row is its subdomain. The subdomain is great to use when you want to associate specific material properties to each element. Rows 1,2 and 3 can be used to calculate our `edof` matrices, as an example

```
enod=t(1:3,:)';                       % nodes of elements
nelm=size(enod,1);                    % number of elements
nnod=size(coord,1);                   % number of nodes
dof=(1:nnod)';                        % dof number is node number
dof_S=[(1:nnod)',(nnod+1:2*nnod)'];   % give each dof a number
for ie=1:nelm
    edof_S(ie,:)=[ie dof_S(enod(ie,1),:), dof_S(enod(ie,2),:),dof_S(enod(ie,3),:)];
    edof(ie,:)=[ie,enod(ie,:)];
end
```

The output `e` is *edges* and contains among other things node-pairs belonging to a certain boundary segment. Thus if you activate the option in PDEtool, <Show Edge Labels> and <Show Subdomain Labels> in boundary mode you can see how the different segments are numbered in `e` and subdomains in `t`. For our intents and purposes the rows 1,2 and 5 are the relevant ones. Row 1 and 2 includes the node numbers of the element-segment and row 5 is the edge label, i.e. which boundary segment it belongs to. For instance a list of all the convective boundaries for the heat problems can be found from

```
% Check which segments that should have convections
er = e([1 2 5],:);   % Reduced e

%conv_segments = [10 11 12]; % Choosen boundary segments
edges_conv = [];
for i = 1:size(er,2)
    if ismember(er(3,i),conv_segments)
        edges_conv = [edges_conv er(1:2,i)];
    end
end
```

`edges_conv` now contains the node numbers of the node pairs that belongs to the convective part of the global boundary. From this the length of each segment $L$ can be determined when calculating the contributions to boundary force vector and the stiffness matrix. Figure 9 shows an example of how the segment numbering and subdomains can look like.

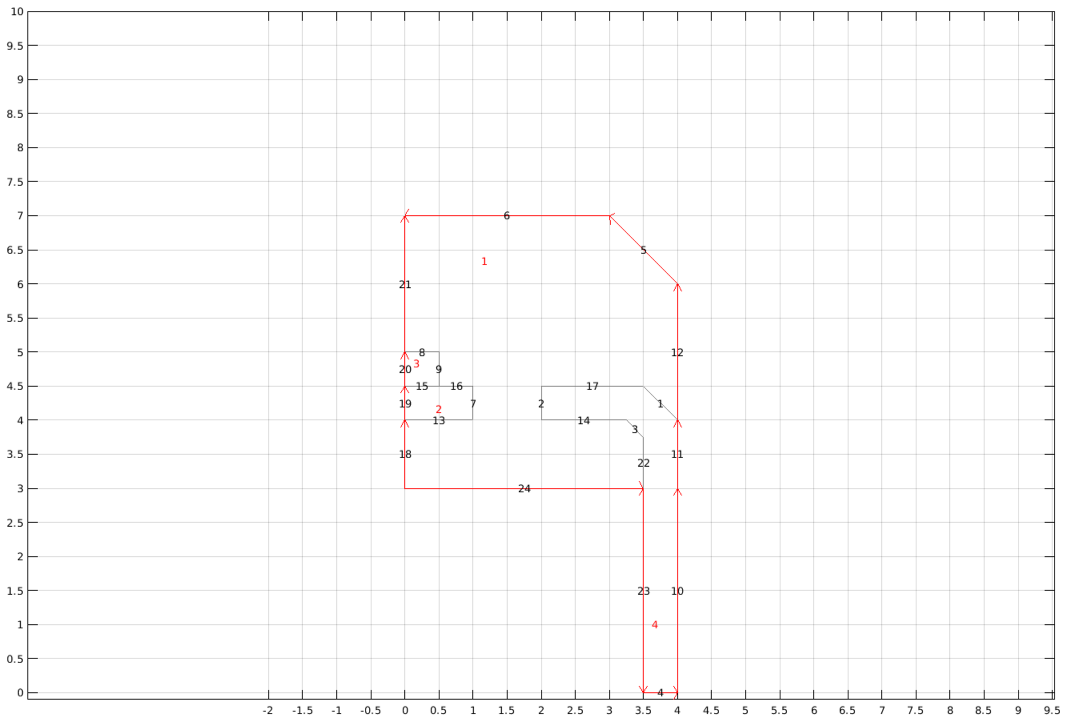


Figure 9: Integrated circuit geometry with subdomains (red numbers) and edge labels (black numbers). Note that the meshing will be constrained by the edges (grey lines) so that an element is wholly whin a single subdomain. Boundary segment 11 and 12 in this case corresponds to where the boundary conditions are changing at the outer boundary, e.g. 11 is isolated while 12 has convection.

Having made a mesh in PDE it is trivial to refine the mesh so it is advantageous to use a small mesh to test your code since it is much faster. When you think it is working use a finer mesh to get better resolved results.

From experience the CALFEM function `assem` and `insert` are very slow. The following code rows does the same thing:

```
%  Kt = assem(edof(el,:),Kt,Kte);
indx = edof(el,2:end);
Kt(indx,indx) = Kt(indx,indx)+Kte;

% f = insert(edof(el,:),f,fe);
indx = edof(el,2:end);
f(indx) = f(indx) + fe;
```

It is always nice to present nice plots, here are some short tips for plotting.

For instance the Matlab function `patch` can be used to generate field plots, e.g. `patch(ex',ey',eT')` where `eT` is the element temperatures obtained from `eT=extract(edof,T)` with `T` as the nodal temperatures. To plot the whole component (both sides of the symmetry cut) you can therefore write:

```
patch(ex',ey',eT')
hold on
patch(-ex',ey',eT')
```

If you use a fine mesh, the mesh lines can sometimes obscure the actual result. To remove the mesh lines from the plotting the option `'EdgeColor'` can be turned off (note that somewhere in the report the mesh used for the results should be shown), i.e. `patch(ex',ey',eT','EdgeColor','none')`.

Don't forget to turn on the colorbar and choose an illustrative color scale, e.g. `colormap(hot)` and set the axis scales to the correct proportions. Example:

```
figure()
patch(ex',ey',eT','EdgeColor','none')
title('Temperature distribution [C]')
colormap(hot);
colorbar;
xlabel('x-position [m]')
ylabel('y-position [m]')
axis equal
```

In order to compare several plots with each other it is recommended to use the same scale. By setting option `caxis([Tmin Tmax])` where `Tmin` and `Tmax` can be calculated or chosen.

To compare the deformation patterns it is preferred they are plotted on top of each other. This can be done in several ways but if `patch` is used the opacity can be set by the option `'FaceAlpha'` where 1 is completely opaque and 0 completely transparent. Example:

```
% Calculate displaced coordinates
mag = 100; % Magnification (due to small deformations)
exd = ex + mag*ed(:,1:2:end);
eyd = ey + mag*ed(:,2:2:end);

figure()
patch(ex',ey',[0 0 0],'EdgeColor','none','FaceAlpha',0.3)
hold on
patch(exd',eyd',[0 0 0],'FaceAlpha',0.3)
axis equal
title('Displacement field [Magnitude enhancement 100]')
```

where the third argument is a color triplet, i.e. `[0 0 0]` is black, so with `'FaceAlpha'=0.3` the result will be a grey-scale plot. You might have to change the magnitude to get nice plots.