

Assignment in The Finite Element Method, 2017

Division of Solid Mechanics

The task is to write a finite element program and then use the program to analyse aspects of a surface mounted resistor. The problem should be solved with Matlab along with suitable sub-routines included in the CALFEM toolbox. A well structured report that presents the findings should be written.

Problem description

Today, many electronic devices do not only have to stand up to the demand of functions, but also have to meet the demand of size. One example is the mobile phone industry, where many applications are integrated, but the size of the phone is limited. Here Surface Mounted Devices (SMD) have an advantage due to their small size. A simple sketch of a surfaced mounted resistor attached to a Printed Circuit Board (PCB) is given in Figure 1 together with the solder. A number

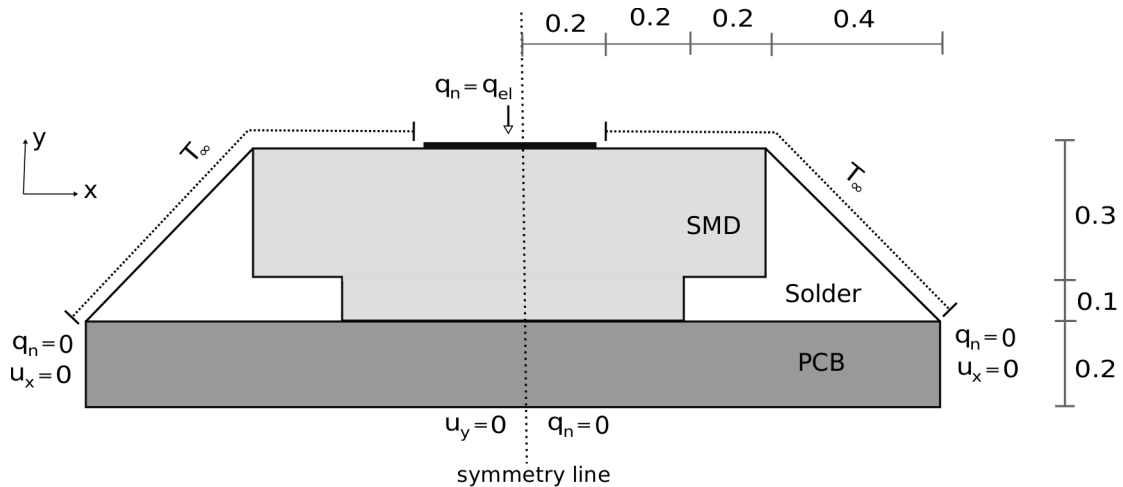


Figure 1: Sketch of the PCB, SMD and Solder that is to be analysed. (Dimensions in mm)

of resistors is mounted next to each other, i.e. the left and right side of the PCB can be assumed to be thermally insulated. Furthermore, the left and right side of the PCB are assumed to be restricted from moving in the x-direction while the bottom side is isolated and restricted from moving in the vertical direction.

A thin film on top of the resistor is exposed to an electric current, producing a heat flow throughout the structure given by q_{el} . The thermal conductivity is k_{smd} , k_{solder} and k_{pcb} for the resistor, solder and the PCB respectively.

On the free surfaces of the SMD and solder, convection is present, which can be modelled using Newton convection law, i.e. $q_n = \alpha_c(T - T_\infty)$. The same convection coefficient α_c applies to all

convection surfaces and it is assumed that the surrounding temperature is T_∞ . Furthermore, the initial temperature of the device is given by T_0 .

Due to the heat generation, thermal stresses will be introduced in the structure. A linear elastic material behaviour, together with the assumption of plane strain conditions, is assumed for the resistor, the PSB and the solder. The material properties are given by E_{smd} , ν_{smd} and α_{smd} for the resistor, E_{pcb} , ν_{pcb} and α_{pcb} for the PCB and E_{sol} , ν_{sol} and α_{sol} for the solder. Here E , ν and α are Young's modulus, Poisson's ratio and thermal expansion coefficient respectively. Furthermore the mass density ρ and specific heat coefficient c are provided for the materials in the numerical table 1.

Problem formulation

- Determine how the temperature distribution changes with respect to time and find the stationary temperature distribution.
- Determine the effective von Mises stress field and the corresponding displacement field due to thermal expansion from the stationary temperature distribution.

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, C^e , is available on the course homepage. 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.

Numerical data

$E_{smd} = 105 \text{ GPa}$	$E_{pcb} = 105 \text{ GPa}$	$E_{sol} = 50 \text{ GPa}$	$T_0 = 30 \text{ C}^\circ$
$\nu_{smd} = 0.118$	$\nu_{pcb} = 0.136$	$\nu_{sol} = 0.36$	$a_c = 40 \text{ W}/(\text{m}^2\text{K})$
$k_{smd} = 0.29 \text{ W}/(\text{mK})$	$k_{pcb} = 1.059 \text{ W}/(\text{mK})$	$k_{sol} = 66.8 \text{ W}/(\text{mK})$	$q_{el} = 9 \cdot 10^3 \text{ W}/\text{m}^2$
$\rho_{smd} = 1850 \text{ kg}/\text{m}^3$	$\rho_{pcb} = 1850 \text{ kg}/\text{m}^3$	$\rho_{sol} = 7265 \text{ kg}/\text{m}^3$	$T_\infty = 20 \text{ C}^\circ$
$c_{smd} = 950 \text{ J}/(\text{kgK})$	$c_{pcb} = 950 \text{ J}/(\text{kgK})$	$c_{sol} = 210 \text{ J}/(\text{kgK})$	
$\alpha_{smd} = 1.2 \cdot 10^{-5} \text{ 1}/\text{K}$	$\alpha_{pcb} = 2 \cdot 10^{-5} \text{ 1}/\text{K}$	$\alpha_{sol} = 1.2 \cdot 10^{-5} \text{ 1}/\text{K}$	

Table 1: Numerical data

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 e.t.c. A suitable structure for the report is:

- **Introduction:** Description of the problem, geometry e.t.c. Keep this section as short as possible.
- **Procedure:** How the problems are solved (weak formulation, application of boundary conditions, thermal strains e.t.c.). Note that you are encouraged to make references to textbooks e.t.c. 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 e.t.c.)
- **Discussion:** A discussion of the results. 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 so on.

A well structured report containing all steps from the strong formulation to the FE formulation shall be returned to the Division of Solid Mechanics no later than **2017-05-24 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 augmented to the points obtained at the exam in May 2017. The assignment should be approved no later than 2017-06-12. 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 May 2017.

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.

A Generating mesh

In this assignment you should create your own mesh using the built-in PDEtool in MATLAB. 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. As an example consider the segment of a dog bone geometry that should be meshed, see figure 2.

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

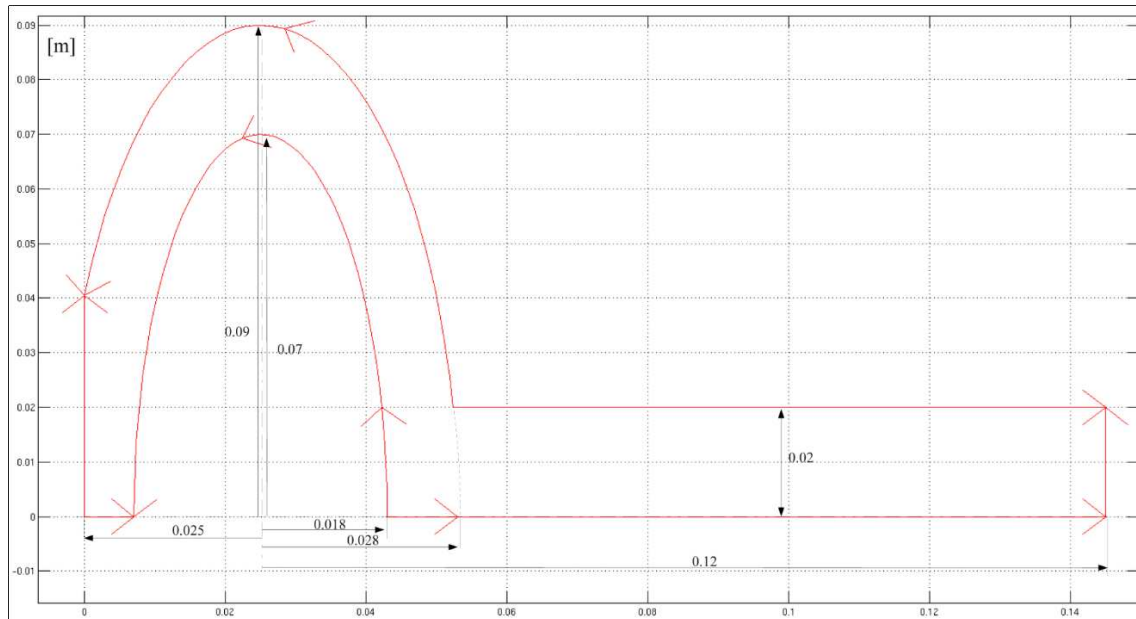


Figure 2: Dog toy geometry.

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).

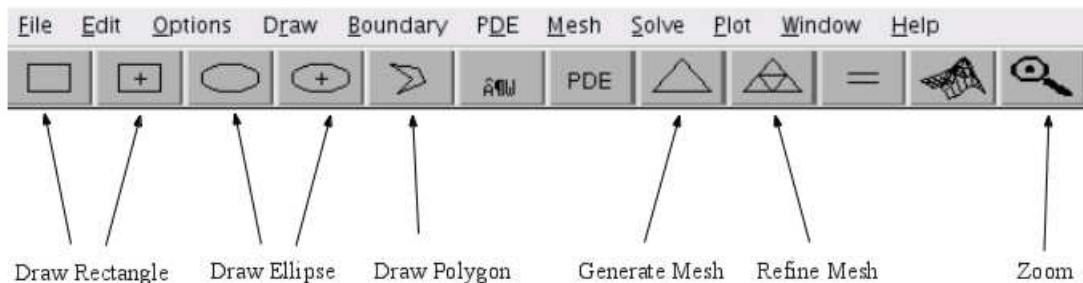


Figure 3: Basic tools for drawing in PDEtool.

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.

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 $\mathbf{R1+E1-E2} \neq \mathbf{R1-E2+E1}$ in general. Below figures showing drawn features and the resulting body after combining by grouping and changing signs in the set formula row (see

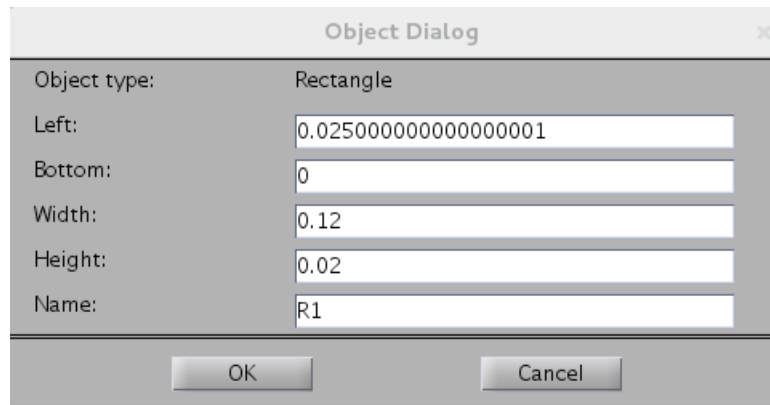


Figure 4: Rectangle object dialog.

figure 5, 7 and 6). Note that to see the resulting geometry we need to go to either meshing or boundary view (see toolbar).

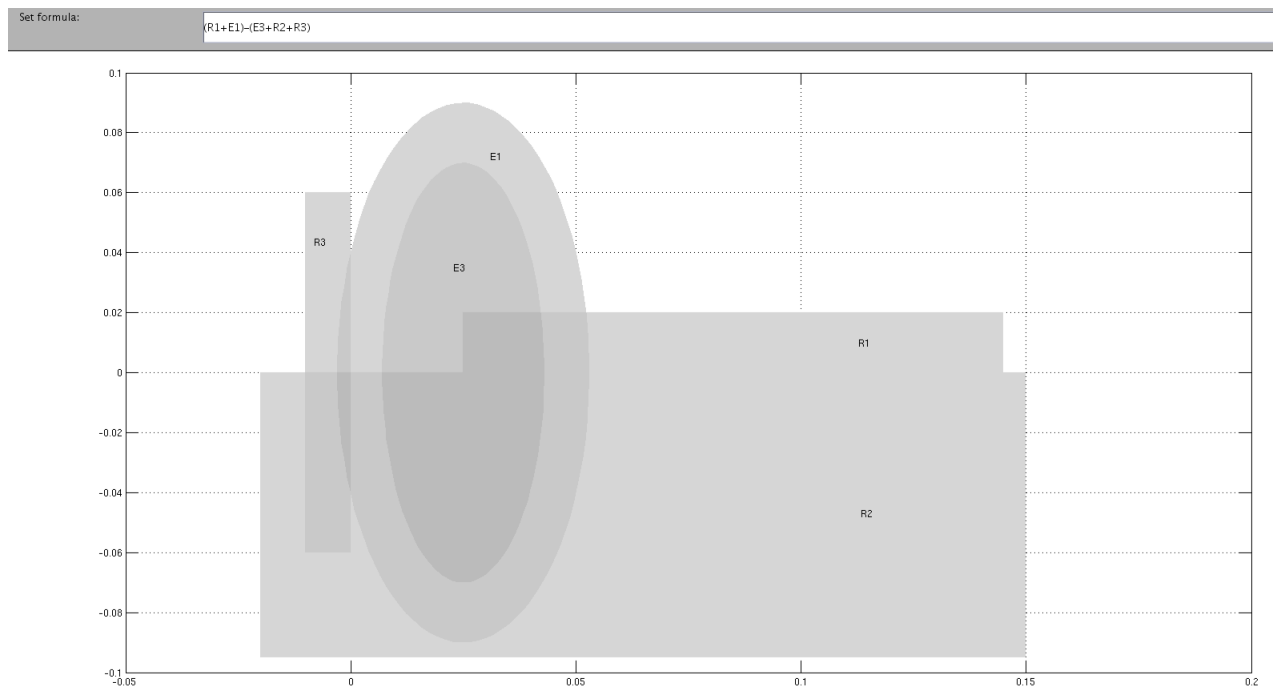


Figure 5: Set of combined shapes.

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.

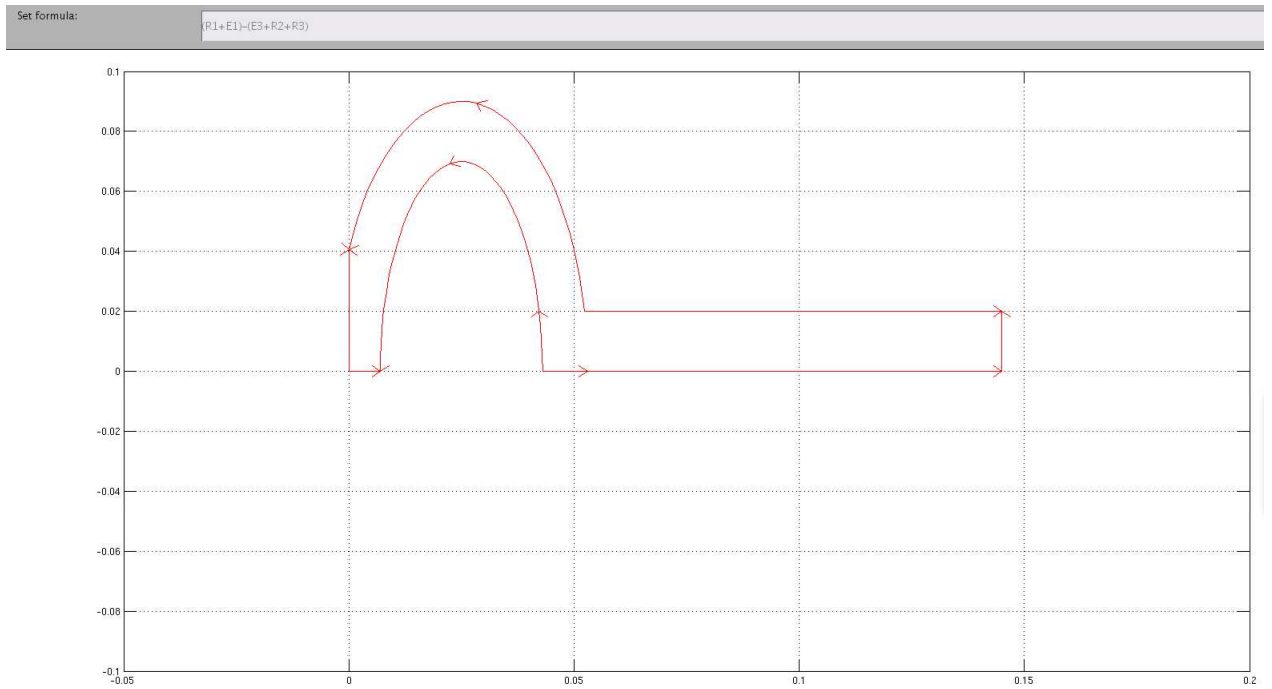


Figure 6: Resulting geometry seen in boundary mode (ctrl+B) after all subdomains have been removed.

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.



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.

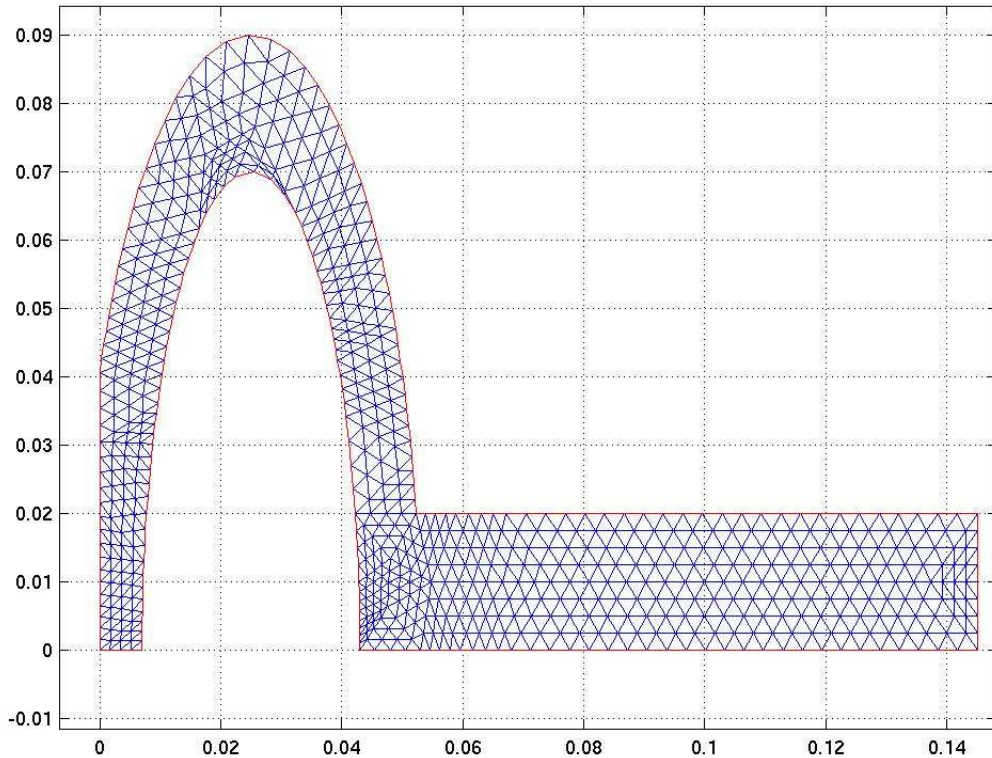


Figure 8: Mesh after two refinements.

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 traditional 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.