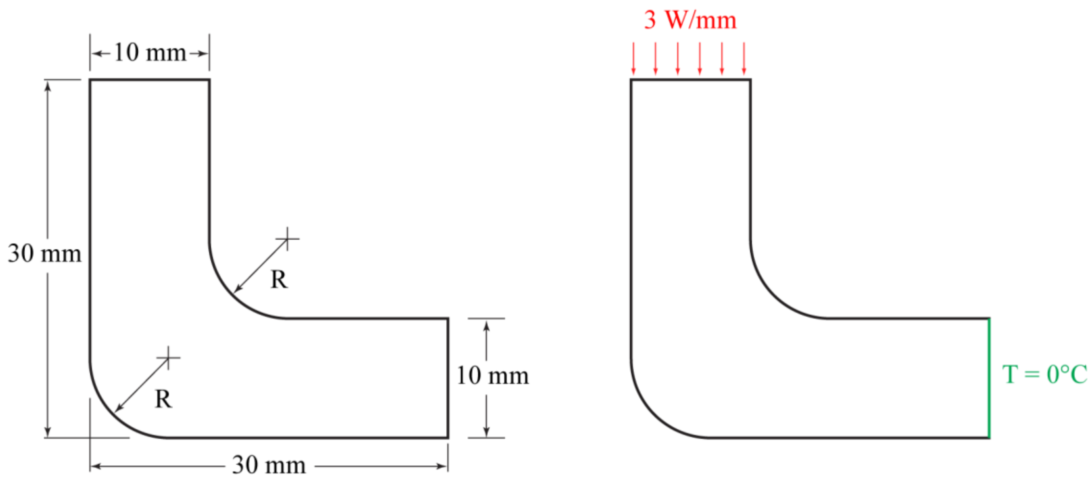
Project #2:



Write a finite element code to analyze the temperature and heat flux fields for a 2 mm thick copper plate subjected to a uniform heat flux of 300 W/mm along its top surface and a prescribed temperature of 0°C at the rightmost edge. Two of the corners of the plate have a blend radius *R*, which is a design parameter for this project. Over the surface of the plate, heat is conducted to the surrounding air with a heat transfer coefficient of 0.0001 W/mm^2-K and an ambient air temperature of $T_\infty = 20$°C. The thermal conductivity of copper is $k = 0.40$ W/mm-K.

Mesh generation

A family of meshes for this problem have been generated for you. The type of mesh is given by the file format in the form of etype-r#-h#.inp, where etype is the element type, the number following r is the radius of curvature in mm, and the number following h is the average element size.

A MATLAB function called **abaqus_reader.m** is also provided to read the mesh file into a MATLAB structure. To use this function, you should call the following:

```
mesh = abaqus_reader('quad4-r2-h0.5.inp');
```

The mesh object created by this command will contain the member variables, mesh.x and mesh.conn that are in the same format that we have used throughout the course.

For the 6-node and 8-node elements, the patch command will not correctly draw the elements. This is because the patch command expects that the vertices around a face are sequentially ordered, but typically in finite elements, the corner nodes are numbered first. For plotting purposes, you can reorder the element nodes using the following (e.g. for 8-node elements) you can do something like this:

```
if size(m.conn,1) == 8
    p.faces = m.conn([1;5;2;6;3;7;4;8],:)';
end
```

Task 1:

Write out the governing equations, including the heat transfer to the surrounding error and then develop the weak form and discretized finite element equations. The weak form must also account for both natural and essential boundary conditions.

Task 2:

Solve for the equilibrium temperature field of the plate. Write a finite element program in MATLAB to approximate the steady state temperature in the plate and compare a solution from ABAQUS (or any other commercial finite element code). Note: if you are using the student version in ABAQUS, you will want to compare solutions using a large radius of curvature as you will not be able to use enough nodes to get a fully resolved solution.

Grading scale:
Points will be awarded for correct completion of each of the tasks listed below with partial credit awarded for incomplete or incorrect attempts. Plots must contain axis labels and units as appropriate for full credit. You may attempt more points than required (however, your maximum score will be capped at 100%).

The total number of points needed for a full 100% depends on how many people are in your group:

General points [up to 30 points]
- [10] Report is clearly written. This means that all plots are correctly labelled, all results have a few sentences explaining the observed trends, the problem you are solving is clearly stated.
- [20] Derivation of the weak form and discretized equations (Task 1). Show how the strong form is developed into a weak form, and then how the test and trial functions are discretized to arrive at the discrete (matrix) equations.

Convergence testing – method of manufactured solutions [up to 70 points]
- [20] Use the method of manufactured solutions to verify that linear elements (either QUAD4 or TRI3) reproduce your manufactured solution at the ideal convergence rate. Test both essential and natural boundary conditions (using the boundaries defined in the figure).
- [20] Repeat with quadratic elements.
- [30] Describe your implementation of the method of manufactured solutions, be sure to describe:
    - the chosen solution for the temperature field
    - heat source term that produces the solution
    - boundary conditions (must use both essential and natural BCs)

MATLAB solution points [up to 200 points]
- [10] Plot the temperature field for linear TRI3 elements.
- [10] Plot the magnitude of the heat flux field for linear TRI3 elements.
- [10] Plot the temperature field for linear QUAD4 elements.
- [10] Plot the magnitude of the heat flux field for linear QUAD4 elements.
- [20] Plot the temperature field for quadratic TRI6 elements.
- [20] Plot the magnitude of the heat flux field for quadratic TRI6 elements.
- [20] Plot the temperature field for quadratic QUAD8 elements.
- [20] Plot the magnitude of the heat flux field for quadratic QUAD8 elements.
- [40] For any of the above mesh types, demonstrate that your finite element mesh is sufficiently refined for a specific radius of curvature. Generate a table where each row shows the approximate element size, the maximum temperature, and the maximum internal heat flux. The element sizes should follow a geometric sequence: h, h/2, h/4, h/8 …
- [40] Repeat the above mesh refinement study to determine the relationship between the radius of curvature and the element size needed to achieve convergence (e.g. to predict maximum heat flux to within 5% accuracy).

ABAQUS solution points [up to 100 points]
- [20] Plot temperature and heat flux fields for linear elements using two different element sizes, e.g. 1 and 2 mm.
- [20] Plot temperature and heat flux fields for quadratic elements using two different element sizes.

- [10] Use a locally refined mesh to obtain the most accurate solution possible when the radius R = 1 mm, with fewer than 1000 nodes. **The most accurate group in the class will receive +20 extra points.** Make sure to show a contour plot of the heat flux magnitude and give the total number of nodes in the model.
- [30] Given the node limitations you may have with ABAQUS or other commercial FEA software, explain whether you believe your ABAQUS solution is consistent with your MATLAB solution.
- [20] Describe how you created your ABAQUS model: what is the geometry used, what elements are defined, how are the loads and boundary conditions applied? What (if any) differences are there in the ABAQUS and MATLAB models you have solved.