Project 1: Steady State Heat Conduction

Alex Hagen

March 10, 2015

1 Code Description

The provided code attempts to use finite element method to solve the heat conduction equation in two dimensions. As requested, the code uses quadrilateral 8 noded and triangular 6 noded elements to provide solutions. The code has been prepared to provide solutions to three problems:

- A 2 dimensional plate with temperature fixed at both x boundaries
- A 2 dimensional plate with temperature fixed at one x boundary and heat flux fixed at the other x boundary
- A 2 dimensional plate with a circular hole with temperature fixed at the internal boundary and at the two x boundaries, and heat flux fixed at the two y boundaries

The code provides the solutions to these as plots of temperature, heat flux, and error (which is defined as $\sigma = T_{FEM} - T_{exact}$) compared to the analytical solutions, which are a linear temperature profile, a quadratic temperature profile, and a more complicated, but provided temperature profile, respectively.

2 Salient Features

The code was written with an attempt to be highly general and object oriented. Although MATLAB is not ideally suited for object oriented programming, the benefits of using object oriented were great. Using an object oriented mesh, which consisted of a structure array of elements, each with their own connectivity matrices and connections to global, allowed for easy adaptation between quadrilateral and triangular elements. This way the object could be passed between objects and, if generalized well, could provide a black box for each step.

Generality of input parameters was also practiced. For the boundaries and their values, it could be possible to simply define specific boundaries and give the values of those boundary nodes explicitly. However, the code provided uses MATLAB's ability to pass function handlers through other functions as arguments. The code then is able to evaluate these functions at the corresponding global values. For example:

```
% creating a function handle anonymously for the criteria
criteria = @(x,y) x == -b;  % if x is on the left boundary, return true
% now pass in this criteria and a group of x and y coordinates
function [nodes_matching_criteria]=check_boundaries(x,y,criteria)
```

```
5     % we can now find the indices where the criteria returns true
6     indices_of_nodes_matching_criteria = find(feval(criteria,x,y));
7     % then we can find the nodes of the x and y values of those points
8     nodes_matching_criteria = [ x(indices_of_nodes_matching_criteria); ...
9       y(indices_of_nodes_matching_criteria) ];
10  end
```

This allowed for the application of functions to boundary conditions and for criteria (which was immensely helpful in dealing with the square plate with a hole).

An attempt was also made to add visual clues to help debugging. In the mesh importation routine, each element is plotted with yellow fill color, with all nodes plotted with a small 'x' to indicate their position. Upon addition of boundary conditions, these are highlighted in green or purple (depending on their type) and an annotation of the value is provided in the visual.

3 Computational Effort Description

Because of MATLAB's inability in object oriented programming, the resulting code is not computationally efficient. In the mesh, every element has a defined matrix including the isoparametric coordinates. In a different language, such as C++ or Python, each element could simply have a pointer to these matrices, defined once in memory. This would also decrease the amount of iterating through structures that is currently required in the code. In general, the code is slow to run (around 80 seconds for a 100 element mesh). By using a lower level language (such as C++ or Fortran) or a more memory efficient language (such as C++ or Python), the computational burden of this code could be drastically reduced.

4 Results and Concerns

For the Dirichlet boundary only condition, the code provides exact results.

For the analyses which have neumann boundary conditions, the results seem to be non-sensical. A good example of this is Figure 4. In Figure 4, there are only boundary conditions on the left and right side, as indicated in the mesh plot. This dictates that our solution should ONLY be in the x direction, and should not vary in the y direction. Unfortunately, the application of the neumann boundary condition is flawed, and there becomes a y direction gradient. This effect is not relieved when increasing the amount of nodes, in fact the magnitude of error increases as the amount of nodes increases. The neumann boundary condition was removed but the uniform source included to check for results, and this provided sensical and exact results. This indicates that the application of the neumann boundary condition is creating some sort of 2nd order effect which is non-physical.

Part I
Figures

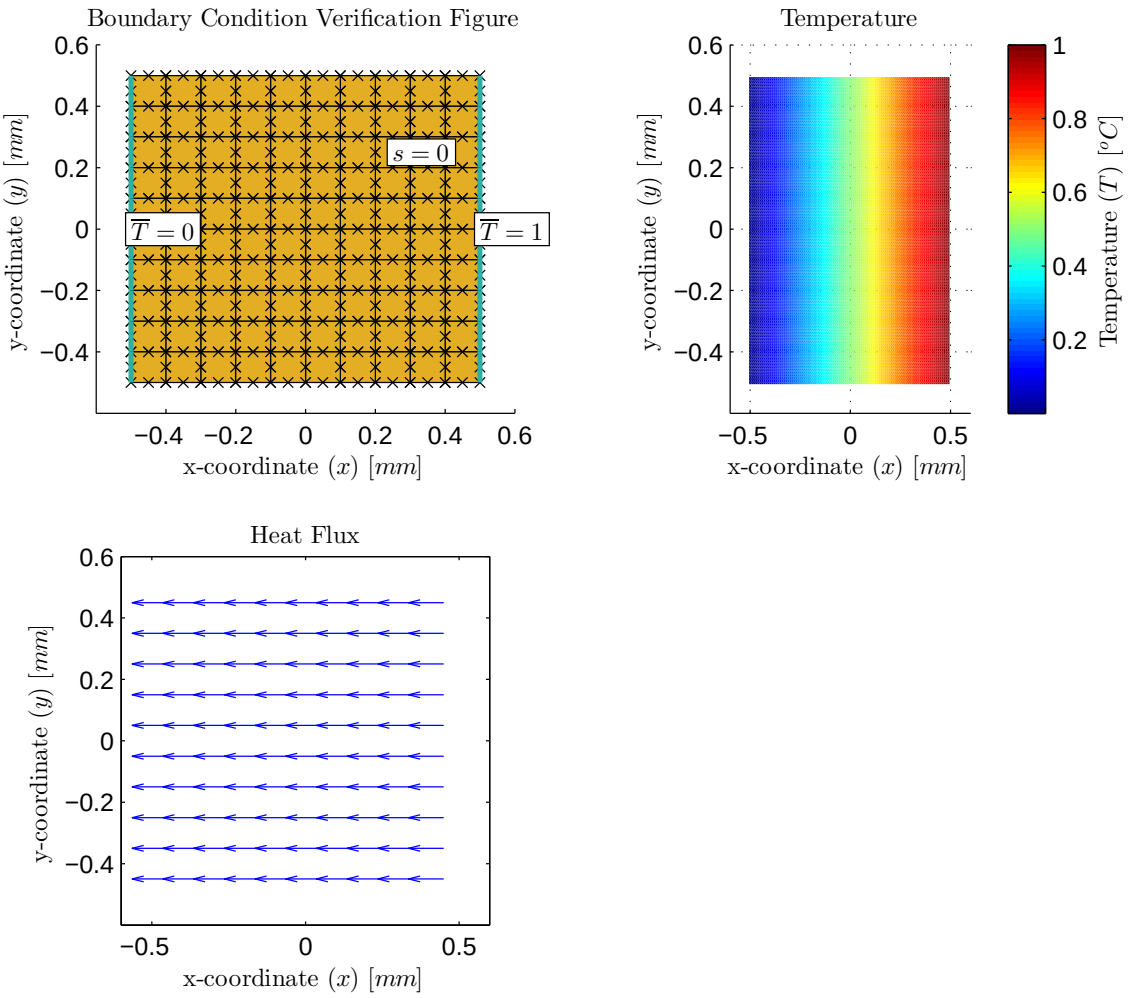

Figure 1: Dirichlet Only Problem with Quadrilateral Elements

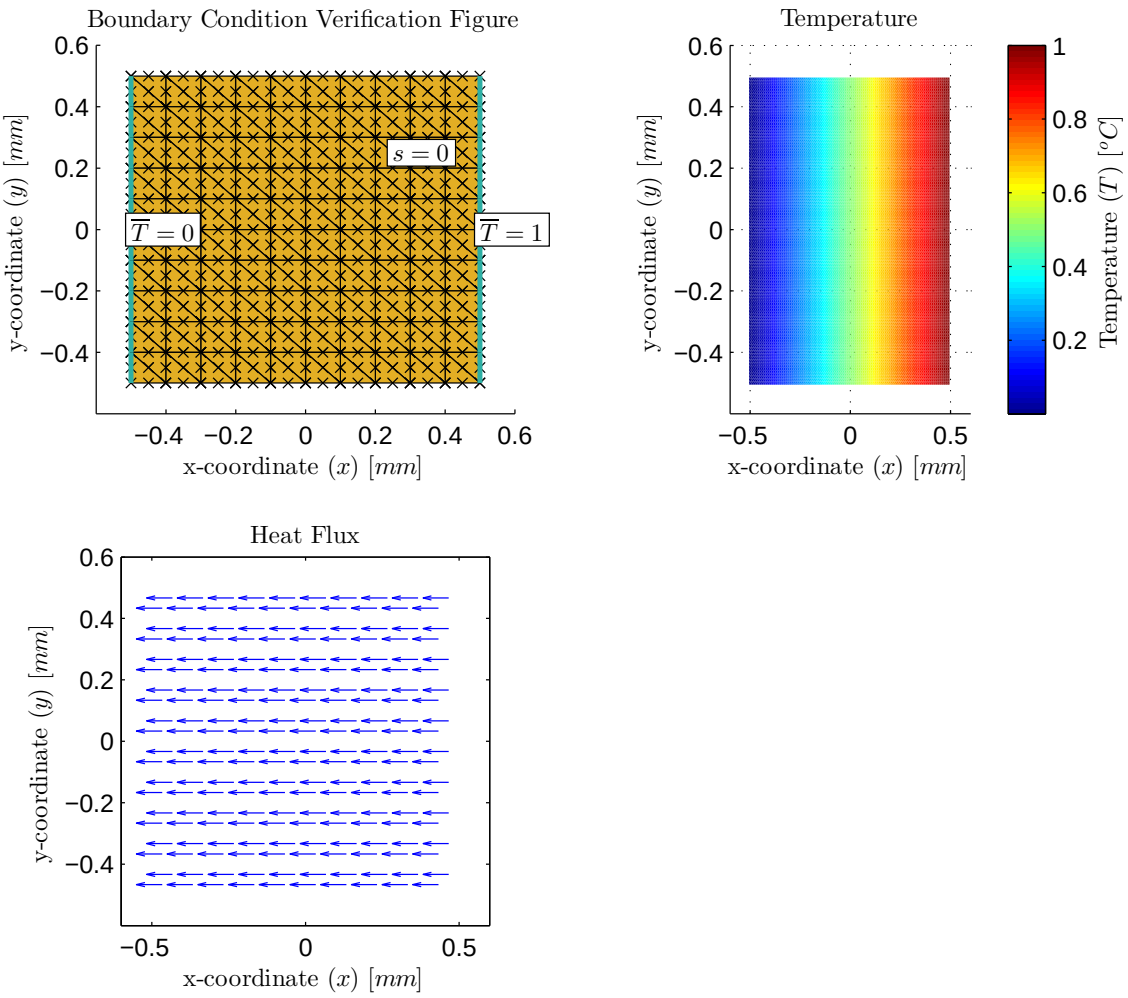

Figure 2: Dirichlet Only Problem with Triangular Elements

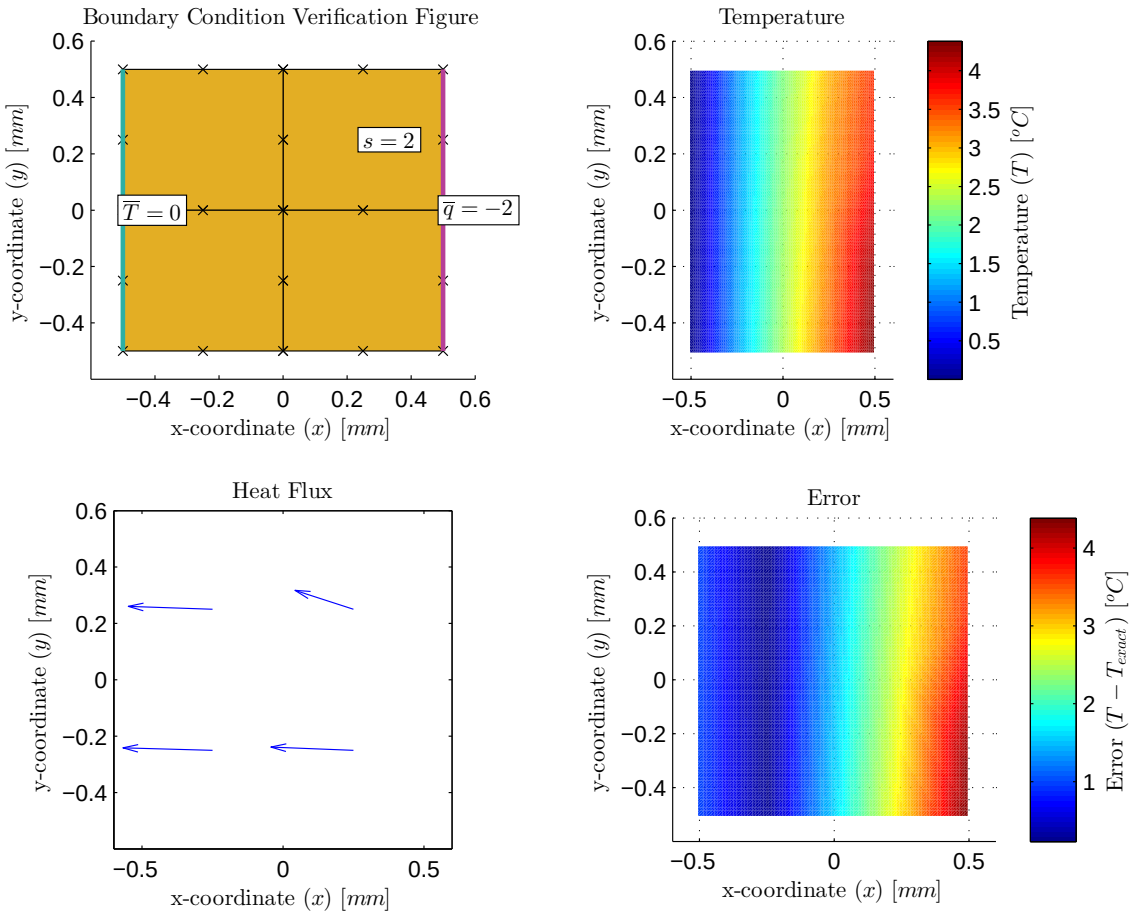

Figure 3: Plate Problem with Low Density Quadrilateral Elements

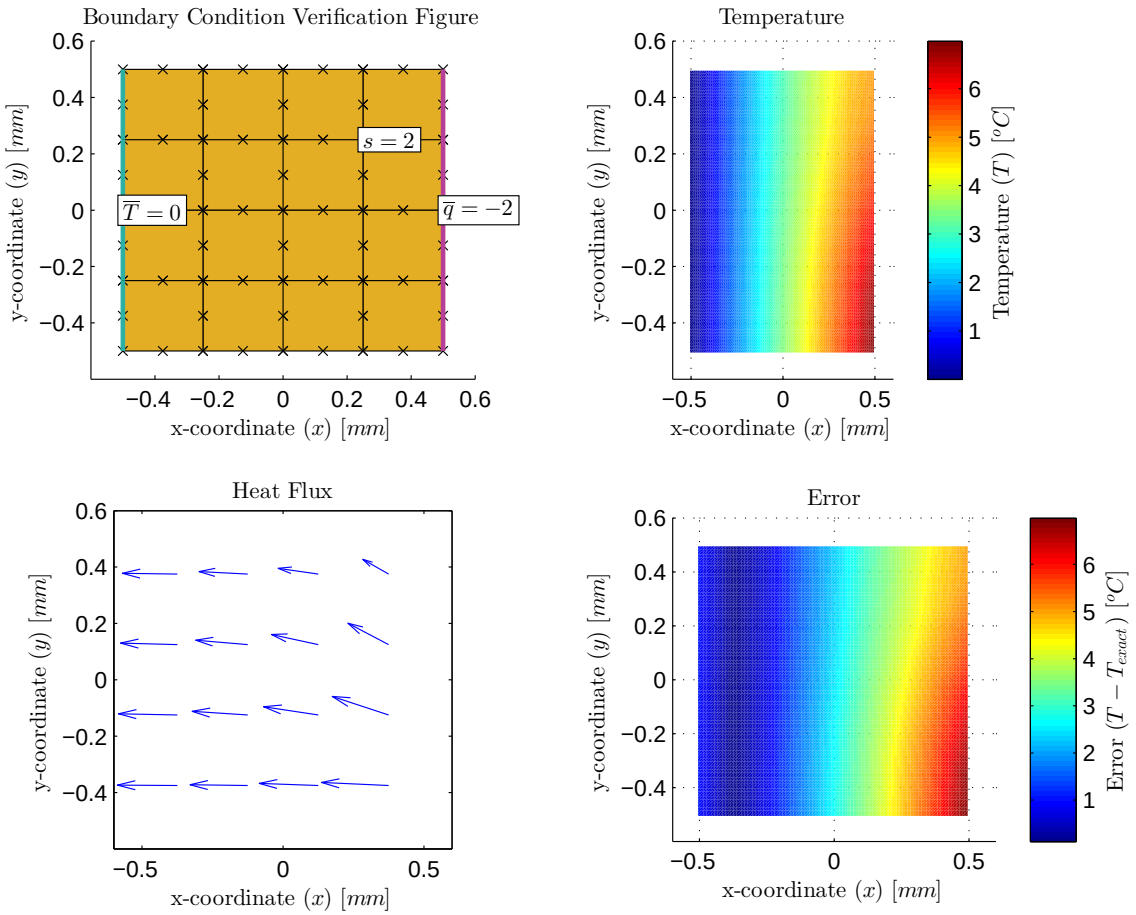

Figure 4: Plate Problem with Moderate Density Quadrilateral Elements

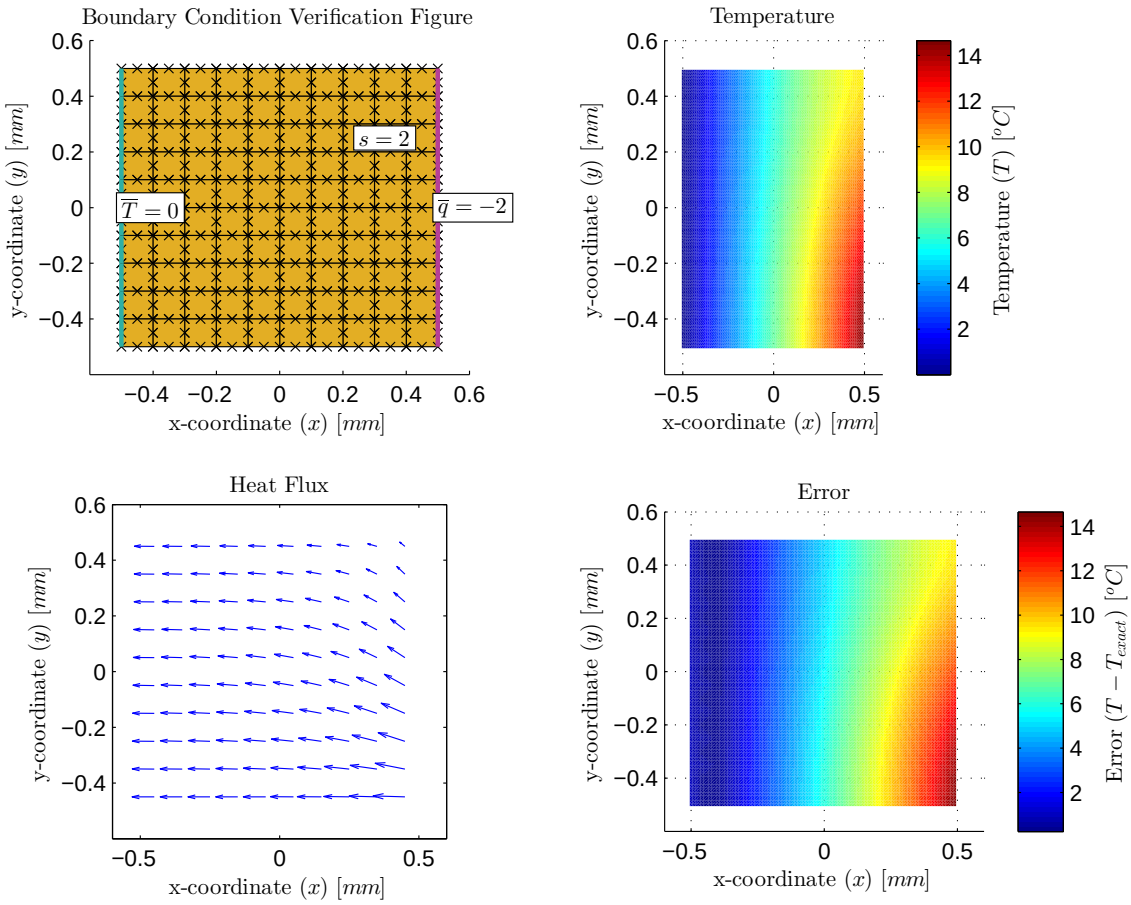

Figure 5: Plate Problem with High Density Quadrilateral Elements

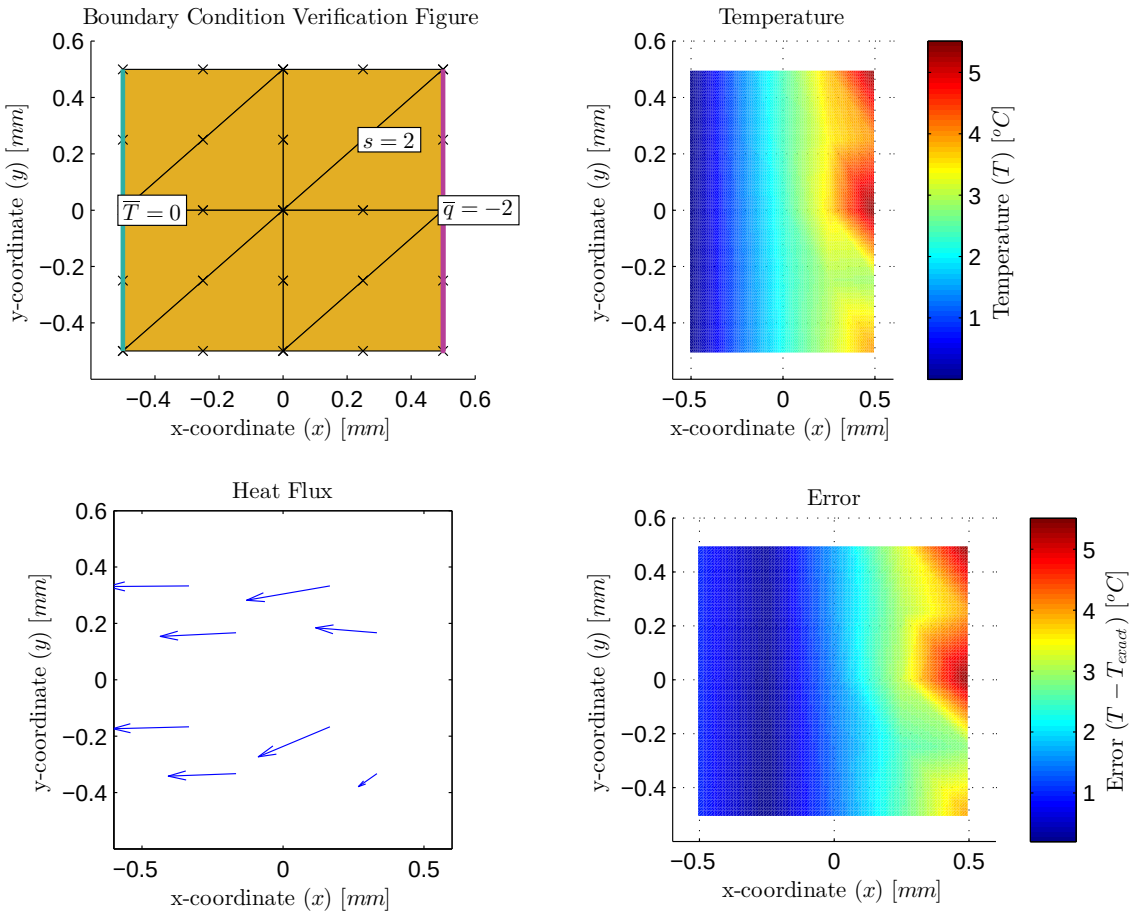

Figure 6: Plate Problem with Low Density Triangular Elements

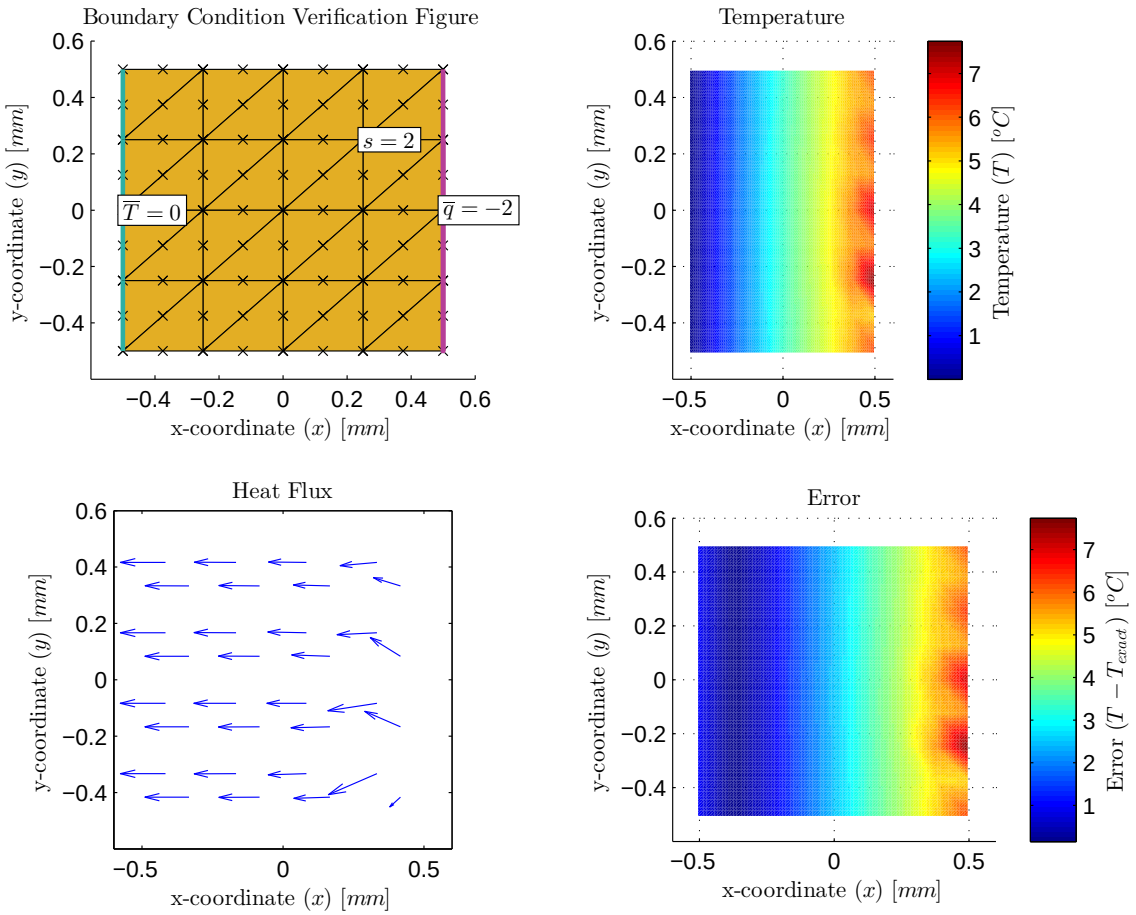

Figure 7: Plate Problem with Moderate Density Triangular Elements

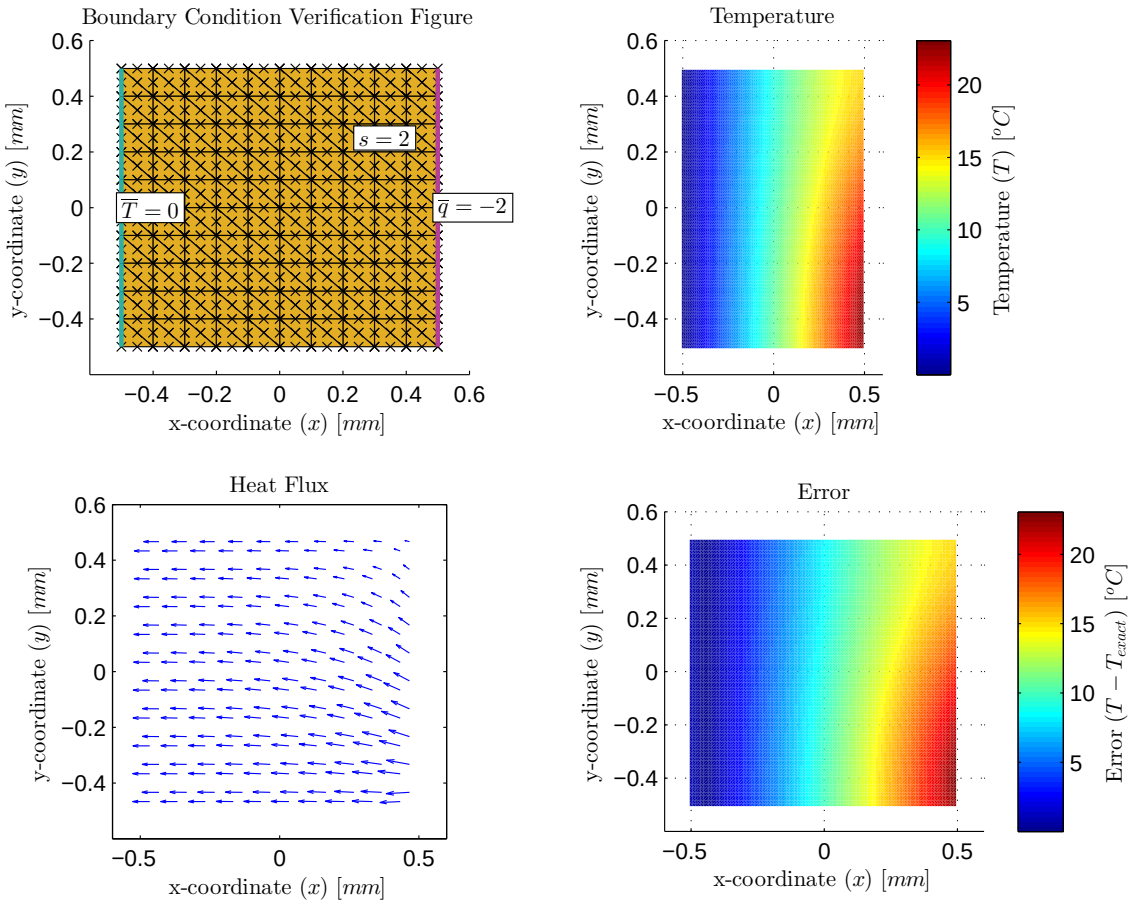

Figure 8: Plate Problem with High Density Triangular Elements

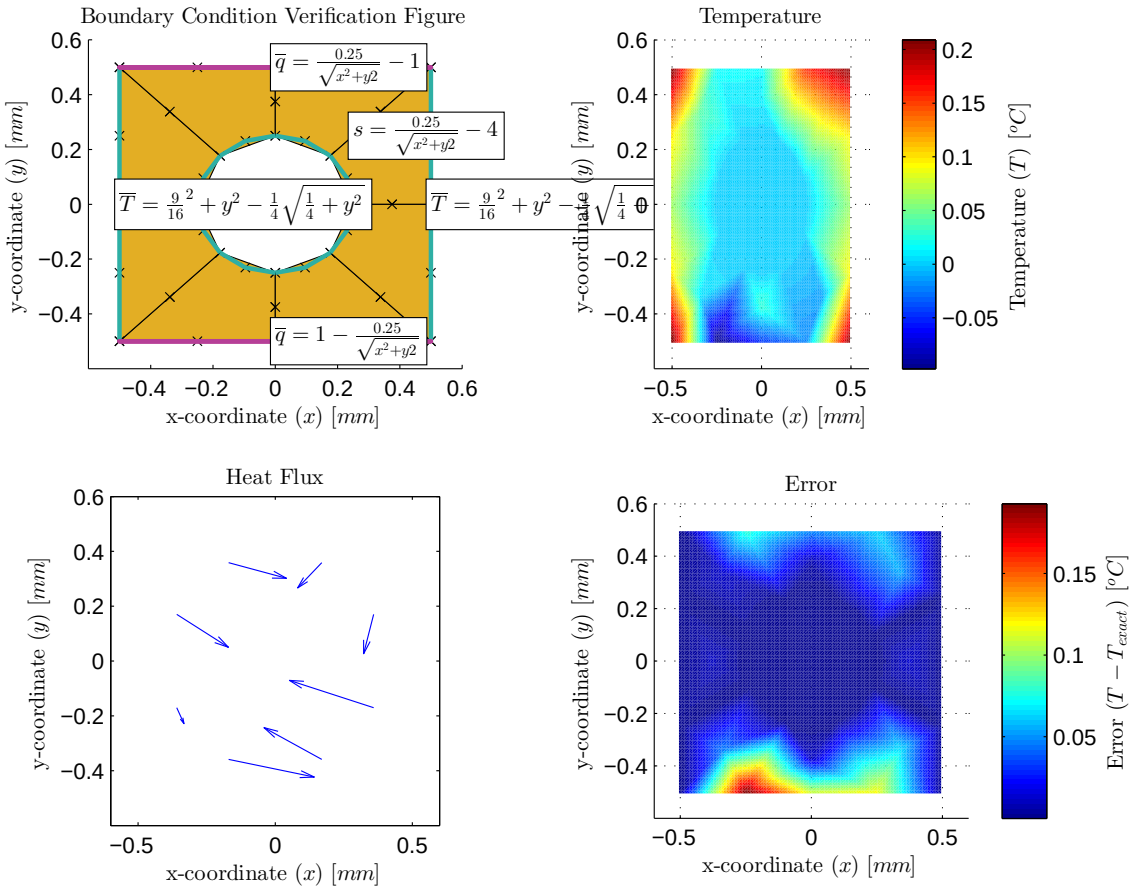

Figure 9: Plate with Hole Problem with Low Density Quadrilateral Elements

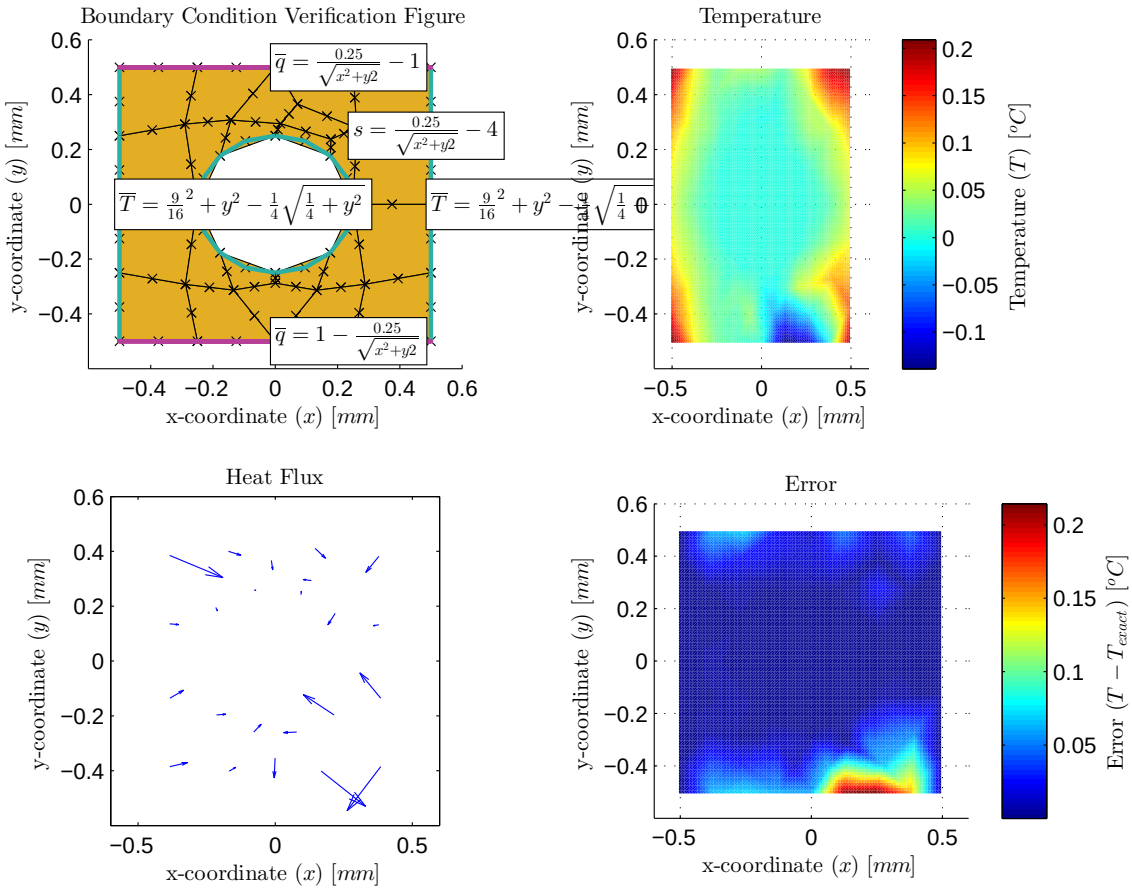

Figure 10: Plate with Hole Problem with Moderate Density Quadrilateral Elements

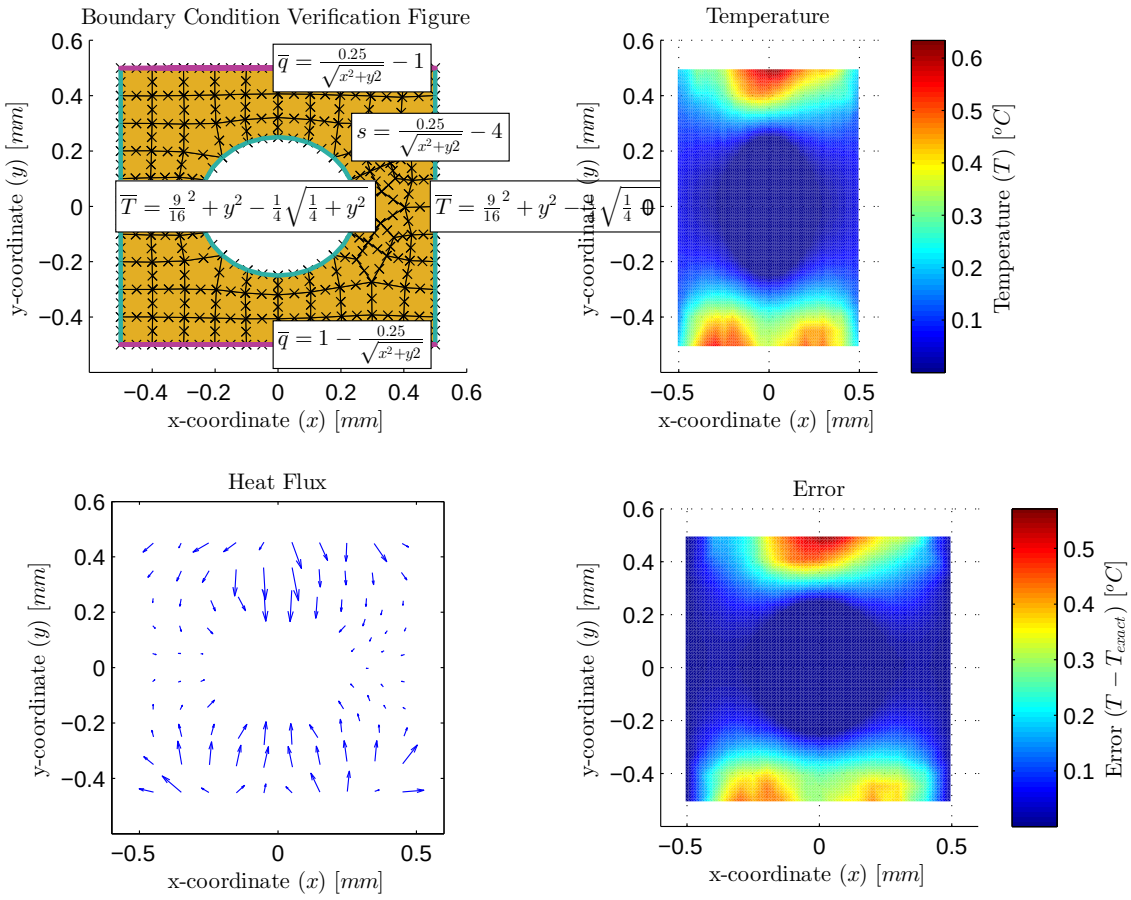

Figure 11: Plate with Hole Problem with High Density Quadrilateral Elements

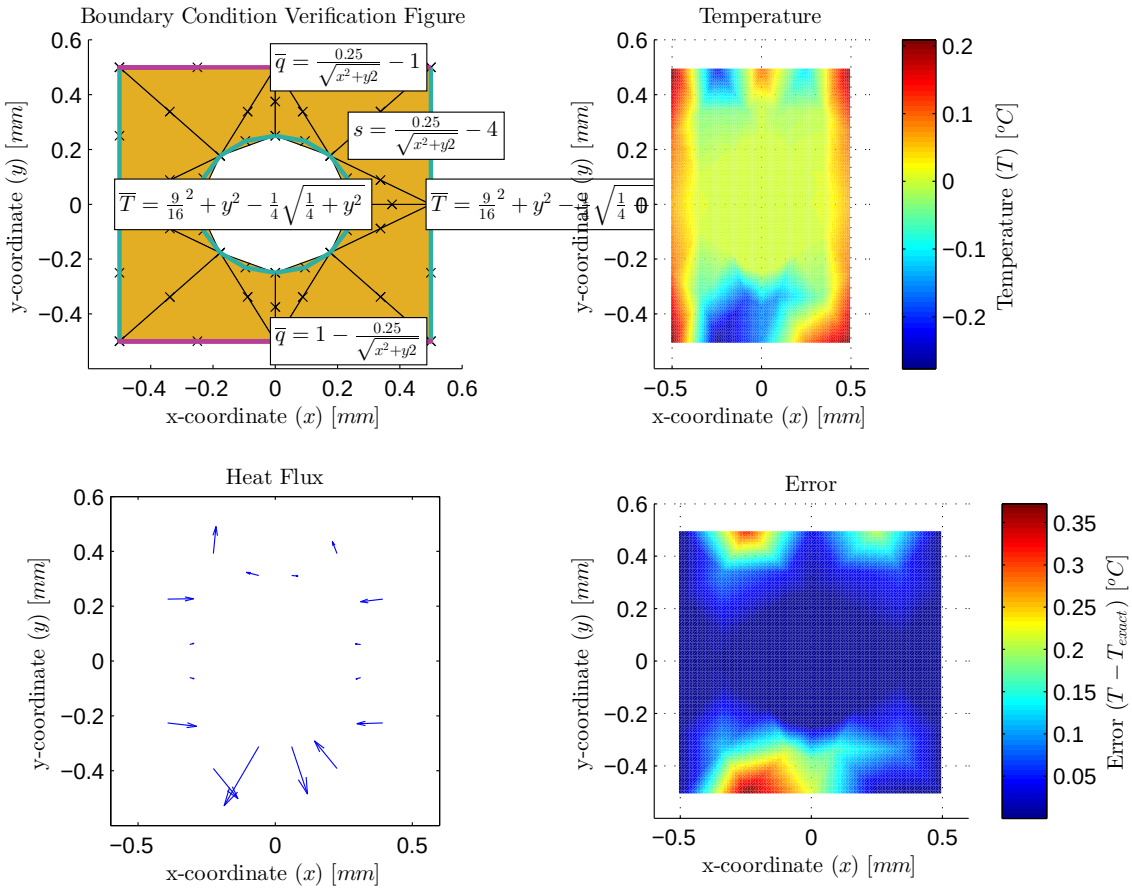

Figure 12: Plate with Hole Problem with Low Density Triangular Elements

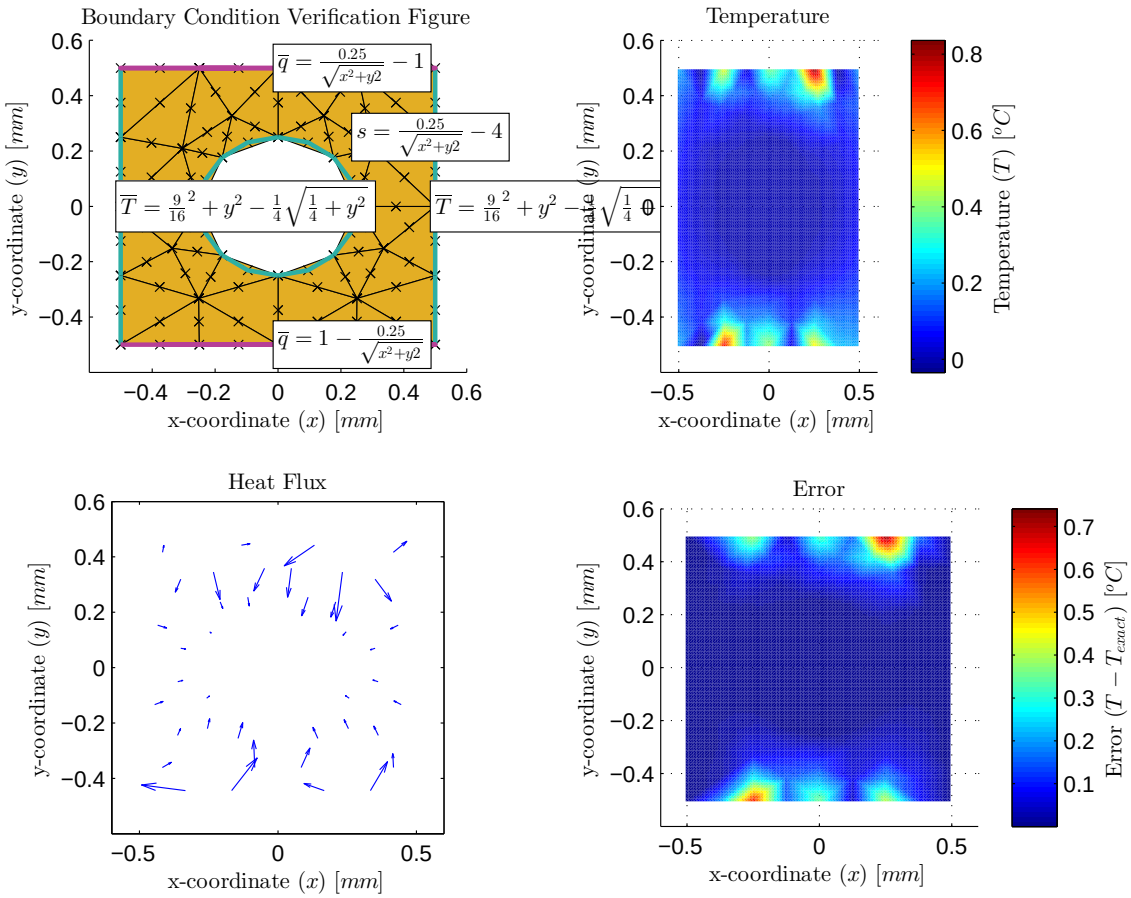

Figure 13: Plate with Hole Problem with Moderate Density Triangular Elements

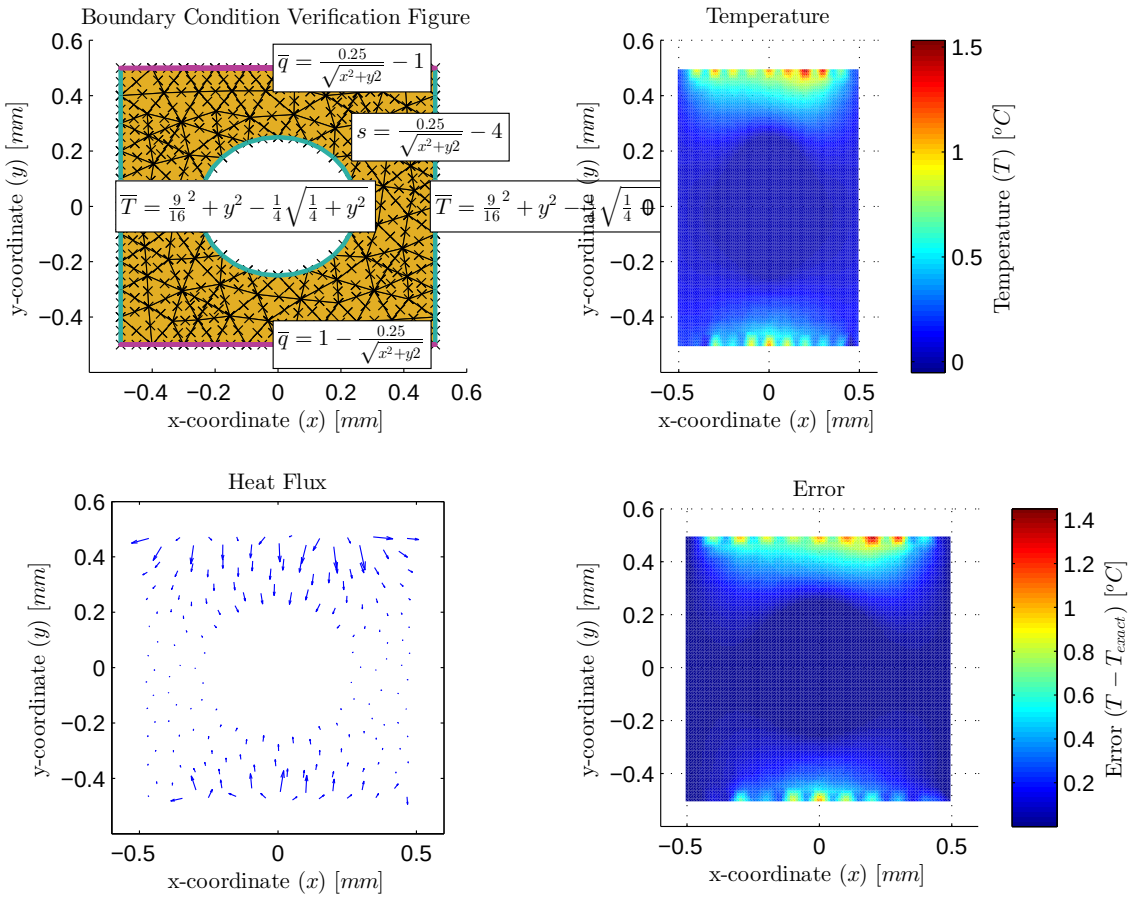

Figure 14: Plate with Hole Problem with High Density Triangular Elements