Project 5: 2D Steady-State Heat Problem

Robert Spragg

April 27, 2017

1 Introduction

The goal of this assignment is to model a laser heating process. To begin, our model will analyze a steady-state, 2D case. This is good practice, because bugs can be detected before advancing to more complicated 3D or transient problems.
The analysis method developed in this project is extremely useful for optimizing existing additive manufacturing processes. These processes require a detailed understanding of the mechanical, thermal, and even magnetic fields within bodies.

2 Problem Setup

The steady-state heat transfer problem is governed by the following equation.

$$0 = \frac{\partial}{\partial x}\left(K\frac{\partial \theta}{\partial x}\right) + \frac{\partial}{\partial y}\left(K\frac{\partial \theta}{\partial y}\right) + z$$

The problem consists of a 1mm square piece of metal. A figure is shown below.

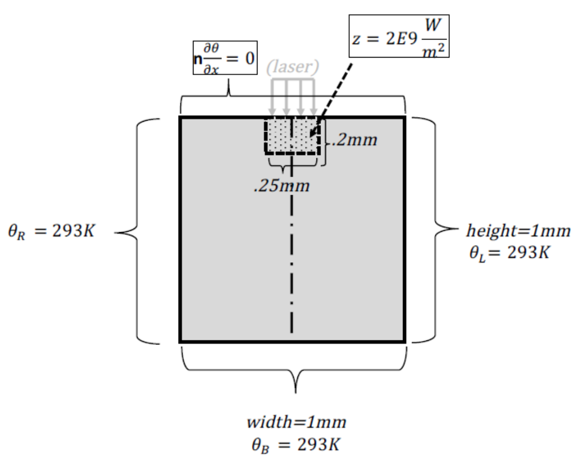

Figure 1: Problem Geometry and Boundary Conditions

As shown in the previous figure, the object has three isothermal sides (each at 293K) and one adiabatic side. The laser is applied to a .25 by .2 mm area in the top region of the square, adjacent to the adiabatic edge. The laser heat source is the following.

$$z = 2E9\frac{W}{m^2}$$

The material properties for our object are shown in the following figure.

$Property$	PLA
K	$.13\frac{W}{m \cdot K}$
$Melting\ \theta$	$473K$

Figure 2: Conductivity and Melting point of PLA

3 Results

Introduction: This report compares our 2D FEM model with the solution generated using ANSYS, a commercial FEM software package.

Part 1
Prompt
Check your solution by producing 2D color plots (heat maps) of the solution at steady state; both in ANSYS and using your own code, and compare them. I recommend using the *pcolor* command in MATLAB (Make sure you use the same temperature axis for all the plots).
Result
The plots for both MATLAB and ANSYS are presented on the next page.

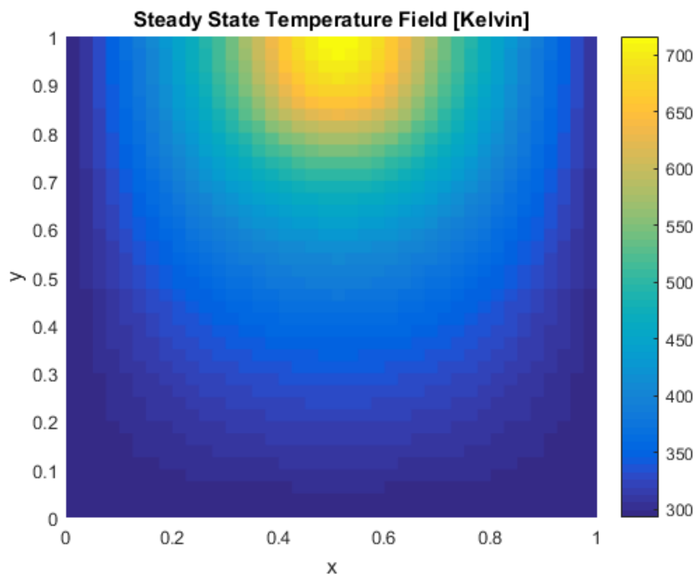

Figure 3: Steady-state Temperature Field —MATLAB

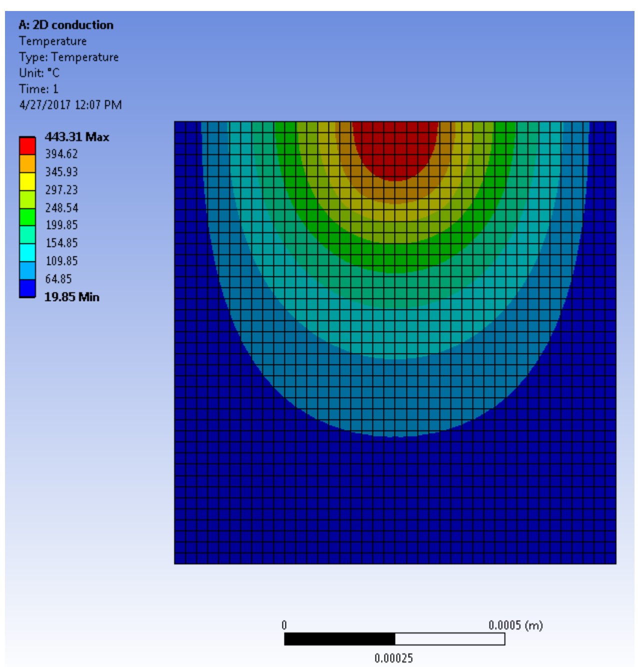

Figure 4: Steady-state Temperature Field —ANSYS [Celsius]

Discussion: As expected, the two solutions are in agreement. The temperatures converge to 293K on the left, bottom, and right edges, which corresponds to the isothermal boundary condition. The temperature gradients shown in both results are also very similar. Most importantly, the maximum temperatures are the same;716 Kelvin, to be exact.

Part 2
Prompt
Provide a plot showing the region of the geometry whose temperature exceeds the melting temperature of PLA.

Result
The melting of PLA is 473K (200°C). To identify where melting occurs, I adjusted the colormap such that it would only plot regions at or above the melting temperature.

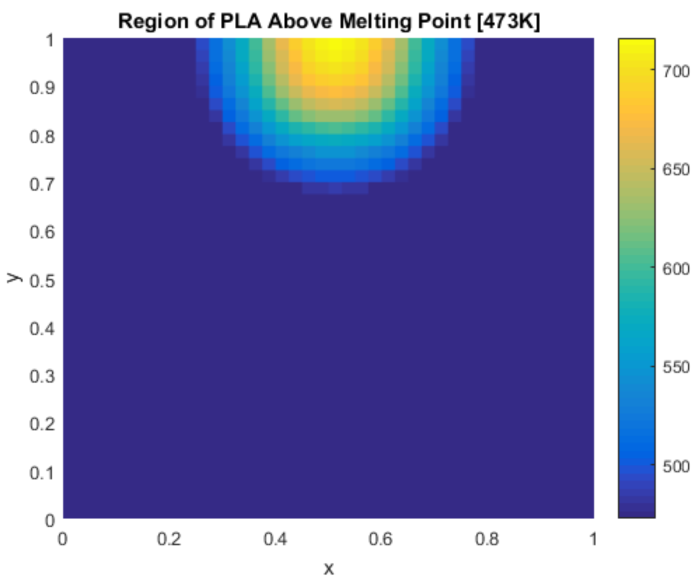

Figure 5: Plot of region above melting point [473K]. Dark blue corresponds to temperatures below melting point

Discussion: It is important to know which region melts when using additive manufacturing. This is because materials will be placed on top of existing materials, which need to be melted to form a solid connection. Furthermore, it will be important t also understand the transient melted region, to better understand how quickly an operation must be completed or how long the time between operations must be. For our laser problem, the melted region consists of the entire area where the laser source is applied, as well as the immediate surroundings, which intuitively makes sense.

Part 3
Prompt
Provide the full derivation of the FEM weakform and FEM matrix form for this problem.
Result
FEM Weak Form Derivation We will start with the strong form of the heat equation. In the equations below, z is the source term.

$$\rho c \frac{dT}{dt} = \nabla \cdot (K \cdot \nabla T) + z$$

Since we are dealing with steady-state, the problem can be simplified to the following.

$$\nabla \cdot (K \cdot \nabla T) + z = 0$$

Per usual, we start by multiplying by a test function v and then integrating over the body. Now, we have the following.

$$\int_{\Omega} \Big(\nabla \cdot (K \cdot \nabla T) + z \Big) \cdot v \, d\Omega = 0$$

Before continuing, we must review the product rule and divergence theorem.
Product Rule: (let $\sigma = K \cdot \nabla T$)

$$\nabla \cdot (\sigma \cdot v) = (\nabla \cdot \sigma) \cdot v + \nabla v : \sigma$$

Divergence Theorem: (let F = $K \cdot \nabla T$)

$$\int_{\Omega} (\nabla \cdot F) d\Omega = \int_{d\Omega} (F \cdot n) dA$$

Using the above product rule analog for our heat transfer problem, we can rewrite our equation in the following manner.

$$\int_{\Omega} \nabla \cdot (K \nabla T \cdot v) d\Omega - \int_{\Omega} \nabla v : K \nabla T d\Omega + \int_{\Omega} z \cdot v = 0$$

Using the divergence theorem, we can rewrite the first term as:

$$\int_{d\Omega} K \nabla T v \cdot n \, dA$$

We now have the weak form of the heat equation.

$$\int_{\Omega} \nabla v : K \nabla T d\Omega = \int_{\Omega} z \cdot v + \int_{d\Omega} K \nabla T v \cdot n \, dA$$

The last term, which is over an area instead of a volume, is our flux term. The flux is defined as:

$$q = K \nabla T \cdot n$$

Rewriting our equation, we now have the **weak form** for the heat equation.

$$\boxed{\int_\Omega \nabla v : K \nabla T \, d\Omega = \int_\Omega zv \, d\Omega + \int_{d\Omega} qv \, dA}$$

Note: For the 2D case, replace the $d\Omega$ terms with dA_e, and replace dA with dL (line integral along the edge of object).

FEM Matrix Form Derivation This section will be very similar to the notes provided to us on 2D thermo implementation. We will start by focusing on the left side of our weak form equation, which corresponds to the **stiffness matrix** term. Since we are in 2D, the gradient matrix will be 2x1, K will be 2x2.

$$\int_{A_e} \left(\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} v \right)^T \begin{bmatrix} K \end{bmatrix} \left(\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} T \right) dA_e$$

Now, we do not know what T is yet. Just like in our 1D problem where we measured deflection, we now define T as the summation of N test functions.

$$T = \sum_{j=1}^{N} a_j \phi_j$$

We do the same for v.

$$v = \sum_{i=1}^{N} b_i \phi_i$$

We can then rewrite these test functions in matrix form, where $\begin{bmatrix} b \end{bmatrix}$ is a vector of all b_i and $\begin{bmatrix} \phi \end{bmatrix}$ is a vector of all ϕ_i.

$$\begin{bmatrix} b \end{bmatrix}^T \left[\int_{A_e} \left(\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} \phi \end{bmatrix} \right)^T \begin{bmatrix} K \end{bmatrix} \left(\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} \phi \end{bmatrix} \right) dA_e \right] \begin{bmatrix} a \end{bmatrix}$$

Now that we have the setup for the integration across the entire body, we then focus our effort on setting up the problem for a single element. First, convert the ∇ functions to the master element space.

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \zeta_1} \frac{\partial \zeta_1}{\partial x} + \frac{\partial}{\partial \zeta_2} \frac{\partial \zeta_2}{\partial x}$$

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial \zeta_1} \frac{\partial \zeta_1}{\partial y} + \frac{\partial}{\partial \zeta_2} \frac{\partial \zeta_2}{\partial y}$$

These equations can be rewritten in matrix form, as desired.

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \zeta_1}{\partial x} & \frac{\partial \zeta_2}{\partial x} \\ \frac{\partial \zeta_1}{\partial y} & \frac{\partial \zeta_2}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial \zeta_1} \\ \frac{\partial}{\partial \zeta_2} \end{bmatrix}$$

The square matrix is known as F^{-T}, or the **deformation gradient**, or the inverse transpose matrix.

We must also define the Jacobian for the transformation to the master element space.

$$J = det(F)$$

F is defined as $\begin{bmatrix} \frac{\partial x}{\partial \zeta_1} & \frac{\partial x}{\partial \zeta_2} \\ \frac{\partial y}{\partial \zeta_1} & \frac{\partial y}{\partial \zeta_2} \end{bmatrix}$

Finally, we can write the matrix form for the stiffness matrix of one element

$$\boxed{\int_{\hat{A}_e} \left([F^{-T}] \begin{bmatrix} \frac{\partial}{\partial \zeta_1} \\ \frac{\partial}{\partial \zeta_2} \end{bmatrix} [\hat{\phi}] \right)^T [K] \left([F^{-T}] \begin{bmatrix} \frac{\partial}{\partial \zeta_1} \\ \frac{\partial}{\partial \zeta_2} \end{bmatrix} [\hat{\phi}] \right) J \, d\hat{A}_e}$$

Next, we will rewrite the **Force Matrix** for the element. Just like before, v can be rewritten as a summation and then as a matrix form of $b * \phi$. Since the b's cancel out, we are left with the following.

$$R = \int_{A_e} [\phi] \, z \, dA_e$$

Next, we can rewrite the element force matrix in the master element domain.

$$\boxed{R_e = \int_{\hat{A}_e} [\hat{\phi}] \, zJ \, dA_e}$$

Note: The same procedure could be done for the flux term, but in both of our reports the flux term equals zero.

Deriving the Penalty Term

We can add the following **Penalty** term to our differential equation, if we desire to solve the problem without removing rows and columns from the stiffness and force matrices.

$$P^* \int_L v(T^* - T) dL$$

For our model, the penalty term forces nodes with a Dirichlet boundary condition (i.e. nodes on the isothermal boundary) to be T^* (293K). This is useful because it removes the post-processing and re-indexing associated with modifying the stiffness matrix.

Using the test functions described previously, our penalty term can be rewritten as the following. Note that the penalty method for 2D Heat flux is a line integral along the edge of the object.

$$P^* \left[b\right]^T \int_L \left[\phi\right]^T \Big(T^* - \left[\phi\right] \left[a\right] \Big) dL$$

After expanding out the equation and converting to the master-element space, we have the following.

$$P^* \int_{\hat{L}} \Big(\left[\hat{\phi}\right]^T T^* - \left[\hat{\phi}\right]^T \left[\hat{\phi}\right] \left[a\right] \Big) J_L dL$$

J_L is the ratio of the lengths of the edge in the real space to the master-element space. Finally, the appropriate parts of the penalty term are applied to the stiffness and force matrices.

$$\textit{Added to Stiffness Matrix: } \Big(P^* \int_{\hat{L}} \left[\hat{\phi}\right]^T \left[\hat{\phi}\right] J_L dL \Big) \left[a\right]$$

$$\textit{Added to Force Matrix: } P^* \int_{\hat{L}} \left[\hat{\phi}\right]^T T^* J_L dL$$

Discussion: The weak form and 2D derivations for the steady-state heat equation are much more complicated than our simple 1D case from past reports. It requires a more thorough understanding of multivariable calculus. However, it is rewarding to have gone through this derivation and actually understand how a program like ANSYS actually solves these types of problems.

Part 4

Prompt

Describe the 2D FEM theory you used to construct your code.

Result This section will discuss how to implement Gaussian quadrature in 2D, as well as the essential components for mapping from the element to global matrices. I will start by discussing Gaussian quadrature.

Gaussian Quadrature

Just like in previous reports, I used four Gauss points. However, in 2D, the problem must be integrated in both the ζ_1 and ζ_2 directions. Therefore, there are actually 16 locations within each element that are used for calculating the element stiffness and force matrices. Furthermore, since each element has four nodes, there are now four $\hat{\phi}$ equations.

$$\hat{\phi}_1 = \frac{1}{4}(1 - \zeta_1) * (1 - \zeta_2)$$

$$\hat{\phi}_2 = \frac{1}{4}(1 + \zeta_1) * (1 - \zeta_2)$$

$$\hat{\phi}_3 = \frac{1}{4}(1 + \zeta_1) * (1 + \zeta_2)$$

$$\hat{\phi}_4 = \frac{1}{4}(1 - \zeta_1) * (1 + \zeta_2)$$

We also have two mapping functions now, which map from the master-element space to the corresponding x or y value in the real domain.

$$x_{map} = X_1 * \hat{\phi}_1 + X_2 * \hat{\phi}_2 + X_3 * \hat{\phi}_3 + X_4 * \hat{\phi}_4$$

$$y_{map} = Y_1 * \hat{\phi}_1 + Y_2 * \hat{\phi}_2 + Y_3 * \hat{\phi}_3 + Y_4 * \hat{\phi}_4$$

The matrix F is found by taking derivatives of the mapping functions x_{map} and y_{map}.

$$F_{11} = \frac{\partial x_{map}}{\partial \zeta_1} \;\; F_{12} = \frac{\partial x_{map}}{\partial \zeta_2}$$

$$F_{21} = \frac{\partial y_{map}}{\partial \zeta_1} \;\; F_{22} = \frac{\partial y_{map}}{\partial \zeta_2}$$

Mapping from Local to Global

Three matrices had to be established before assembling the global matrices.

- Element Matrix (keeps track of nodes for each element) [4-by-# Elements]
- BC Matrix (keeps track of boundary conditions for each element)[2-by-# Elements]
- Position Matrix (keeps track of x and y location of each node [2-by-# Nodes]

To ensure our method was in line with proper FEM theory, we took precautions to ensure the following

- Element node indexing done in a consistent manner (counter-clockwise)
- Stiffness Matrix symmetrical positive definite
- Laser source term only applied to elements within source region

To ensure our numbering was consistent, we numbered nodes row by row. To ensure the laser source term was applied correctly, we made sure that nodes lying on the edge of the source for elements outside of the source area did not receive a source value in their force matrix.

Discussion: There were many challenges associated with implementing the 2D FEM code. Some issues I faced involved improperly implementing Gaussian Quadrature. At first, I only checked four points within the element, which corresponds to the diagonal of the element. Also, I originally calculated the F-inverse-transpose matrix wrong, which gave me NaN for my stiffness matrix. Finally, I had trouble implementing the penalty method, and ended up solving the problem using the row/column elimination method before going back and attempting penalty method again.

4 Appendix

This section contains the .m files that were used in this report.

- hw5_nopenalty_final.m
- gauss_quad4_hw5_final.m