Introduction

 **Computer Project (CP)**

In order to meet the demands of your future career where computational skills are necessary to compete in today’s (and tomorrow’s) industries, this project will utilize what you have learned in this class to solve complex heat transfer problems. For this project, you are to analyze a 2D steady state conduction problem in MATLAB using the finite differencing techniques located in your textbook and presented in class. There will be two options for performing this project depending on what you feel you can accomplish: **Option A**: the simple geometry (max 80% grade) or **Option B**: the complex geometry (max 100% grade). Only choose one to submit for a grade.

Project Requirements

Your Deliverable for this project must meet the following requirements:

* •  The final deliverable must be submitted into a ***single word document*** that includes your entire code in the appendix of that document. This code will be extracted from the document in order to confirm your results presented in your paper. If your code doesn’t work as written in your paper, ***you will receive a ZERO.***
* •  Your code must handle a variable mesh in multiples of 10. If not, ***you will receive a ZERO.***

o For the simpler geometry: 10x10, 20x20, etc.
o For the complex geometry: 5x10, 10x20, 20x40, etc.

• Finite differencing techniques using MATRIX math MUST be used to solve this problem. Do not use any other technique such as Gauss-Seidel ***or you will receive a ZERO***.

Project Deliverable

Your report must include the following sections:

1. ***(5 pts)*** Introduction – a succinct summary of the project
2. ***(15 pts)*** Code Validation – checks that helped you to assure that your code was presenting

correct results:

* 1. Visual comparison versus the key (to be posted before the due date).
	2. Net heat flow summed ***over the boundaries of your simulation*** should match the heat

generated in the system (if correct, the difference between these should be within 10-10)

* 1. Your temperature profile should match what is physically expected given EACH of the

prescribed boundary conditions (***describe SPECIFICALLY what EACH boundary condition should do to the temperature profile, and verify that this is reflected in your 3D temperature map***)

1. Doubling your mesh density should not affect your solution (i.e. your calculated heat flows) by more than 0.5%
2. ***If you do not get a reasonable answer, show how your solution is NOT valid and discuss HOW you would use these to check whether your code is valid***
3. ***(15 pts)*** Results
	1. A table listing the quantity of heat transfer per unit depth along each of the non-

insulated edges, using a positive value for heat transfer into the solid and negative for heat transfer out of the solid. This should be the TOTAL (summed) heat transfer along each edge that you use for your code validation.

* 1. A 3D projection of the temperature distribution (the ‘surf’ command may prove useful), arranged in a way that clearly shows the full temperature distribution, *in the correct orientation*.
	2. Plots of the temperature profiles along each of the edges. Put these together into a single figure with a 3x2 subplot (not a single plot) using the subplot command. Begin with the bottom edge, then work your way clockwise around the geometry.
1. Code – a FULL copy of your code so we can confirm your results presented in your paper. ***This must be the full code TEXT and not an image of your code so we can directly copy and paste it into Matlab. The code must at least include:***
	1. ***(40 pts, 30 for easier geometry)*** Correct coefficient matrix ‘A’ and known vector ‘b’
	2. ***(15 pts, 10 for easier geometry)*** Correct temperature matrix and plotting
	3. ***(10 pts, 5 for easier geometry)*** Validation calculations: nodal calcs and edges vs.

generation

**Other grading comments:**

* •  **Up to -10 points for grammar and writing clarity**
* •  **100% deduction for plagiarized code**
* •  **100% deduction for reporting results not generated by your submitted code**
* •  **100% deduction for submitting code that cannot handle a variable mesh size**
* •  **NO LATE DELIVERABLES ALLOWED**

Project Timeline

There will be one final deliverable for this project. But to keep you on track with your work, there will be weekly milestones to follow that will not be graded. To keep you from waiting till the last minute to work on this project, my office hours (and my TA’s) will give priority to students who have questions on that week’s milestones. If you are not asking current milestone questions, I will ask you to go to the end of the line (if there is one). The weekly milestones are as follows:

**Week 1 Milestones (completed by February 18th):**

* •  Comprehension on finite differencing techniques to be used (see ch4 and lecture notes).
* •  Choose a numbering scheme for your nodes and draw it out by hand to visualize those nodes
* •  Identify each node “type” for those nodes and determine the appropriate energy balance for each node type.
* •  Determine your “logic” to identify any node in your domain based on the node number ‘I’ and the size of your mesh ‘n’. This will ultimately be used in your if statements to identify nodes in your for loop.
* •  Write a pseudocode/outline in Matlab to use as a skeleton to populate your code with. **Week 2 Milestones (completed by February 25th):**
* •  Have code that accurately identifies nodes and neighbors of each node based on the node number (i).
* •  Have code that will identify the appropriate energy balance equation for each node (i) that uses the appropriate neighboring nodes. This will involve populating the respected locations within your computational matrix.

**Week 3 Milestones (completed by March 4th):**

* •  Have code that will correctly populate the coefficient matrix ‘A’ and the known vector ‘b’.
* •  Have code that will solve for the unknown temperature vector ‘T’.
* •  Reshape the temperature vector into a matrix that can ultimately be plotted for visualization.

**Week 4 Milestones (completed by the DUE DATE):**

* •  Code that creates the required plots
* •  Code that calculates the required heat flows
* •  Completed Code Validation
* •  Completed final report with code text copied into the word document

Example Skeleton Pseudocode

\*\*\*\*\*\*
%Choose mesh size (e.g. n=10 for 10x10, 20 for 20x20, etc.)
%Initialize your variables
%Calculate number of nodes (probably something related to n2)
%Run a for loop to define all the nodetypes and put them in their proper location for i=1:allnodes

if i == xxxxx %use if statements (may need more than one) to determine where node ‘i’ is %e.g. left boundary, bottom right corner, etc. Should be able to determine

%using only i and n %Must be node type 1 (for example)

elseif i==xxxxx %use if statements to determine where node ‘i’ is %must be node type 2 (for example)

elseif i ==xxxx %continue for all node types

end end

%coefficient matrix and b vector should now be populated. Solve for T values %rearrange T values into matrix to make surface plot
%verify sum of qin to each node adds to 0
%calculate heat flows on edges, verify that it matches generation

%make edge plots

Geometry Options
Complex geometry (max 100% grade)

Simple geometry (max 80% grade)

*Figure 1: Complex Geometry*

*Figure 2: Simple Geometry*