18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Outline

- **Project 1: 2-D Thermal Analysis**
  - Project overview
  - Thermal analysis review
  - Project details

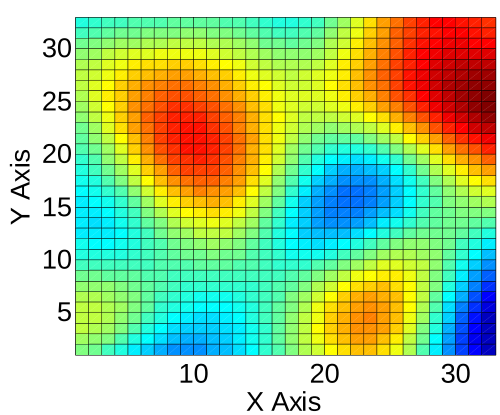
Project Overview

- In this project you are required to write a MATLAB program to simulate a 2-D steady-state thermal problem

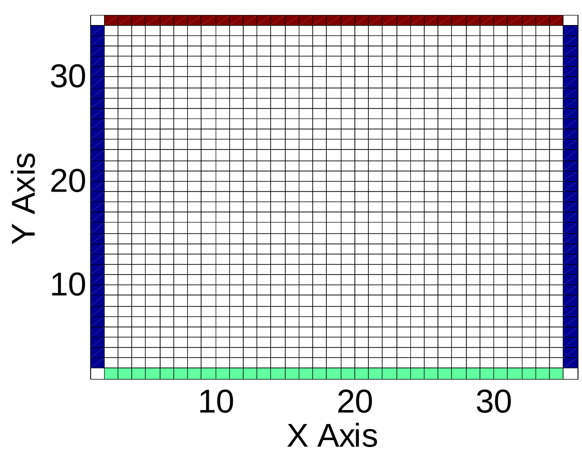
- The program mainly consists of the following two steps:
  - Formulate and discretize thermal PDEs
  - Solve the resulting linear systems by both Gaussian elimination and Cholesky factorization

Project Overview

- You will be given 3 test cases
- In each test case, you will be provided with:



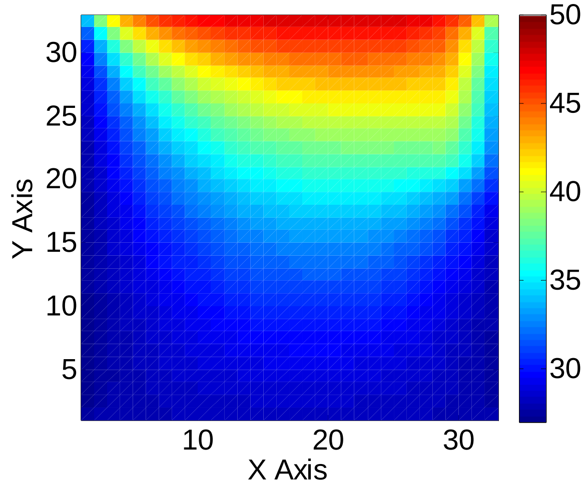
Power density within the medium



Temperature at the boundary

Project Overview

- **Your program will generate:**



Temperature plot within the medium

2-D Heat Equation

- **Heat equation is a 2nd-order linear PDE**

Density — Laplace operator

$$\rho \cdot C_p \cdot \frac{\partial T(x,y,t)}{\partial t} = \kappa \cdot \nabla^2 T(x,y,t) + f(x,y,t)$$

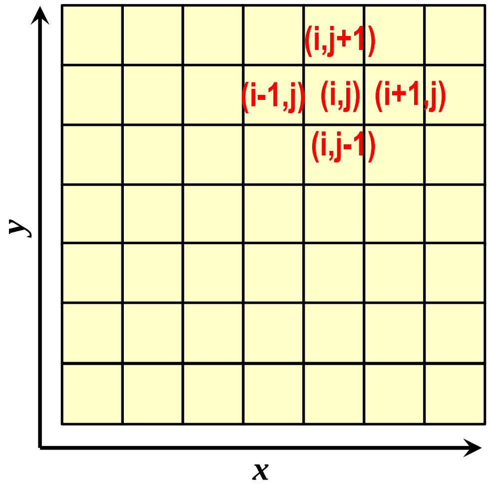
Thermal capacity — Thermal conductivity — Heat source

- **We assume that heat conduction has reached a steady state**
  - Heat equation can be simplified as

$$\kappa \cdot \nabla^2 T(x,y) + f(x,y) = 0$$

Equation Discretization

- **Discretize 2-D space into a number of small panels**



$$\kappa \cdot \nabla^2 T(x,y) + f(x,y) = 0$$

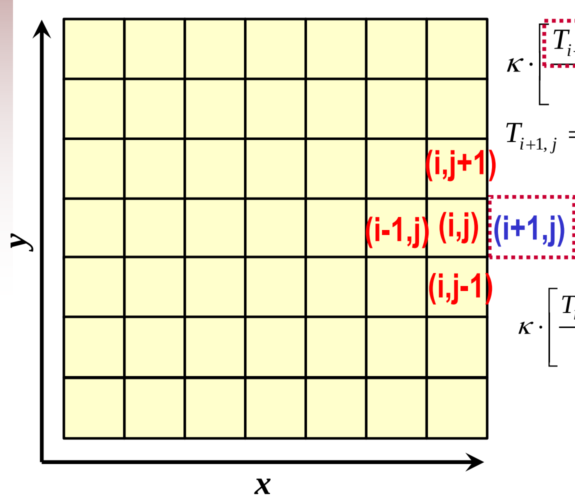
$$\kappa \cdot \left[\frac{\partial^2 T(i,j)}{\partial x^2} + \frac{\partial^2 T(i,j)}{\partial y^2} \right] = -f(i,j)$$

$$\frac{\partial^2 T(i,j)}{\partial x^2} = \frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{\Delta x^2}$$

$$\frac{\partial^2 T(i,j)}{\partial y^2} = \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2}$$

Boundary Condition

- **We assume a given temperature at the boundary**
  - You need to move the constant term to RHS of the equation



$$\kappa \cdot \left[\frac{\boxed{T_{i+1,j}} + T_{i-1,j} - 2T_{i,j}}{\Delta x^2} + \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2} \right] = -f(i,j)$$

$$T_{i+1,j} = T_c$$

$$\kappa \cdot \left[\frac{T_{i-1,j} - 2T_{i,j}}{\Delta x^2} + \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2} \right] = -f(i,j) - \frac{\kappa \cdot T_c}{\Delta x^2}$$

System of Linear Equations

- **Combine all linear equations**

$$\kappa \cdot \left[\frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{\Delta x^2} + \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2} \right] = -f(i,j)$$

$$(1 \le i \le N \quad 1 \le j \le M)$$

- **We get a system of linear equations**

$$A \cdot X = B$$

$$X = [T_{1,1}, T_{1,2}, \ldots, T_{N,M}]^T$$

- **The matrix A is symmetric and positive definite**
  - The linear system can be solved by using either Gaussian elimination or Cholesky factorization

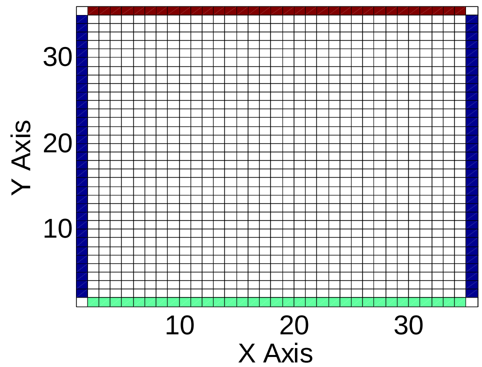
Project Files

- **All files for this project can be found from the distributed package**
  - A report template: Proj1.doc
  - A MATLAB function to generate thermal plot: thermalplot.m
  - Two templates: thermalsimGauss.m, thermalsimCholesky.m
  - Three sets of test data: case1.mat, case2.mat, case3.mat (use "load case<u>x</u>.mat" in MATLAB to import test data)

- **Each mat file contains the following variables:**
  - mediumX: x-dimension of the medium
  - mediumY: y-dimension of the medium
  - p: discretized power density
    p(i,j) means the power density at x = i, y = j
    size(p,1) is the number of panels in x-direction
    size(p,2) is the number of panels in y-direction

Project Data (cont'd)

- leftBound: temperature at the left boundary (x = 0)
  leftBound(j) means the temperature at T(0, j)
- rightBound: temperature at the right boundary (x = N+1)
- topBound: temperature at the top boundary (y = M+1)
- bottomBound: temperature at the bottom boundary (y = 0)



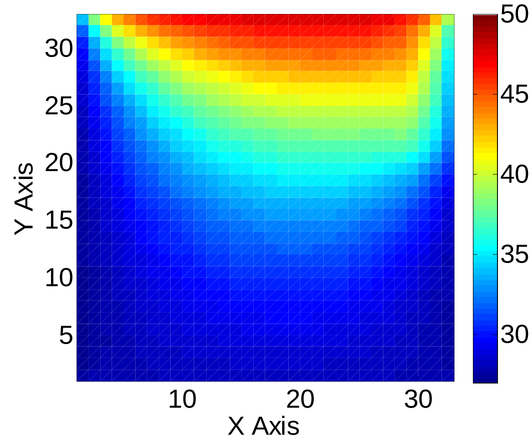
- **Thermal conductivity constant: κ = 157 W / m • K**

Project Requirements

- **You are required to implement MATLAB functions with the following format (templates included in the project package)**
  - [Temperature] = thermalsimGauss(p, mediumX, mediumY, leftBound, rightBound, topBound, bottomBound);
  - [Temperature] = thermalsimCholesky(p, mediumX, mediumY, leftBound, rightBound, topBound, bottomBound);
  - Temperature is a matrix with the same dimension as p
- **Your program must work on Windows or Linux computer without any modification**
- **You must implement Gaussian elimination and Cholesky factorization by yourself**
  - It is not allowed to use MATLAB functions such as *chol* or *backslash*

Temperature Plot

- **For each test case, you should generate the temperature plot using the following function**
  thermalplot(Temperature);



- **Please include the generated thermal plots in your report**

Project Submission

- **You should zip the MATLAB code into a single file and submit it to the course web site**

- **You should also submit a PDF report (at most 4 pages) to the course web site, including the following items**
  - A high level description of your implementation
  - Your approach for formulating the linear equation
  - Your approach for solving the linear equation
  - Temperature plot of each benchmark
  - Anything else that will make your program unique
  - A WORD template is provided for your project report

Grading Criteria

- **The total points will be distributed as the follows**
- **75% for MATLAB code and results**
  - We will compare your results to our "golden solution" for accuracy where the accuracy will be evaluated by:

$$Error = \sqrt{\frac{\sum_i \sum_j \left[T*(i,j) - T(i,j)\right]^2}{\sum_i \sum_j \left[T*(i,j)\right]^2}}$$

  where *T** is the golden solution and *T* is your solution. This error value must be less than 0.01
- **25% for project report**
  - You should complete all sections in the report template and clearly address all required points mentioned in the previous slide