Numerical Solution to 2D Flat Plate Problem

Nitin Kuamr 202211059
Tejas Pakhale 202211061
Rahul Gupta 202211069
Rajat Kumar Thakur 202211070

April 19, 2024

1 Project Overview

This project aims to employ numerical techniques to solve the 2D Flat Plate problem, a classic problem of heat transfer. The problem involves determining the temperature distribution over a flat plate subjected to heat transfer. By employing numerical methods, we seek to approximate the solution to this problem. Analytical solutions to these problems are often challenging due to their complex nature, necessitating the sue of numerical methods for approximation.

2 Solution Approach

We aim to solve the 2D Flat Plate problem numerically using the Finite Difference Method (FDM). FDM involves discretizing the domain into a grid and approximately derivatives using finite difference approximations. By giving the discretized equations iteratively, we can obtain numerical approximations of the temperature fields over the flat plate.

3 Finite Difference Method

- **Finite Difference Approximations:** In FDM, derivatives in the governing equations are approximated using finite difference formulae. Commonly used finite difference approximations include Central, Forward and Backward differences, which estimates derivatives at a point based on function values at neighbouring points.
- **Discretization of Domain:** The domain is discretized into a grid with spacing Δx and Δy in the x and y directions, respectively. The governing equations are applied at discrete points on this grid, , resulting in a system of algebraic equations that can be solved iteratively.

- **Iterative Solution Procedure:** The iterative solution procedure involves updating the solution at each grid point based on the neighboring values and boundary conditions. This process continues until convergence criteria are met, ensuring that the numerical solution approximates the true solution to the governing equations.

4 Selection of Derivative Method for FDM

For this project, we will use central finite difference approximations for the first-order derivatives in the x and y directions. Central differences offer second-order accuracy and better numerical stability compared to forward or backward differences, making them suitable for our purposes. Central differences provide accurate approximations of derivatives while minimizing numerical errors, ensuring reliable solutions to the governing equations.

5 Formulae Used for the Solution

- **Heat Conduction Equation:** The governing equation for heat conduction in 2D steady-state problems is given by:

  $$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

  where,
  T is the Temperature distribution,
  x and y are the spatial coordinates

- **Temperature Equations:**
  x-direction:

  $$\frac{\partial T}{\partial x} \approx \frac{T_{i+1,j} - T_{i-1,j}}{2\Delta x}$$

  y-direction:

  $$\frac{\partial T}{\partial y} \approx \frac{T_{i,j+1} - T_{i,j-1}}{2\Delta y}$$

- **Second-Order Derivative of Temperature Equations:**
  x-direction:

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

  y-direction:

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

Here,
$T_{i,j}$ ***represents the temperature at node***(i, j).
Δx and Δy are the grid spacings in the x and y directions, respectively.

- ***Discretized Equation:***
  On substituting the above finite difference approximations into the heat conduction equation, we get the discrete equation for each node:

  $$T_{i+1,j} + T_{i-1,j} - 4T_{i,j} + T_{i,j+1} + T_{i,j-1} = 0$$

  This equation will be solved iteratively to obtain the temperature distribution.

6 Algorithm

- Initialize the temperature distribution over the grid.
- Iterate until convergence:
  - For each interior node, update the temperature using the discretized equation.
  - Apply boundary conditions.
  - Set up a linear system of equations.
  - Solve the linear system using either the Gauss-Seidel method.
  - Check for convergence.
- Once convergence is achieved, the temperature distribution is obtained.
- Create a ".txt" file and for each grid point, write the indices (i, j) and the corresponding temperature value T[i][j] to the file and close the file.
- Use "gnuplot" function to generate a 3D plot of the temperature distribution from the data file.

7 Case Study

In this case study, we took the example case where the boundary conditions are as follows:-

- Top-edge Temperature: 800
- Bottom-edge Temperature: 800
- Left-edge Temperature: 800

- Right-edge Temperature: 800

Maximum allowed iterations: 60

The output of the example is shown in the following images:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\tnpak\Desktop>cd "c:\Users\tnpak\Desktop\" && gcc maths.c -o maths && "c:\Users\tnpak\Desktop\"maths
Enter the temperature for the top edge: 800
Enter the temperature for the bottom edge: 800
Enter the temperature for the left edge: 800
Enter the temperature for the right edge: 800
Grid Temperature:
0.00    800.00  800.00  800.00  800.00  800.00  800.00  800.00  800.00  0.00
800.00  799.61  799.30  799.10  799.03  799.08  799.23  799.46  799.73  800.00
800.00  799.28  798.72  798.36  798.23  798.32  798.59  799.01  799.50  800.00
800.00  799.05  798.30  797.82  797.65  797.76  798.13  798.68  799.34  800.00
800.00  798.91  798.05  797.51  797.31  797.44  797.87  798.50  799.24  800.00
800.00  798.87  797.99  797.42  797.22  797.36  797.80  798.45  799.21  800.00
800.00  798.93  798.09  797.55  797.36  797.49  797.91  798.52  799.25  800.00
800.00  799.07  798.33  797.87  797.70  797.81  798.17  798.71  799.35  800.00
800.00  799.26  798.68  798.32  798.18  798.27  798.56  798.98  799.49  800.00
800.00  799.50  799.11  798.86  798.76  798.83  799.02  799.31  799.65  800.00
800.00  799.75  799.56  799.43  799.39  799.42  799.52  799.66  799.83  800.00
0.00    800.00  800.00  800.00  800.00  800.00  800.00  800.00  800.00  0.00
```