Task Two Report

Task Description

In this task, you will implement simple 2D or 3D computational fluid dynamics (CFD) simulations for basic flow problems using finite difference methods (FDM) or finite element methods (FEM). These methods are foundational numerical techniques used in CFD to solve partial differential equations (PDEs) that describe fluid flow, heat transfer, and other physical phenomena. You will work on problems such as heat transfer, diffusion, or convection, which involve solving equations like the heat equation or the incompressible Navier-Stokes equations.

Implementation

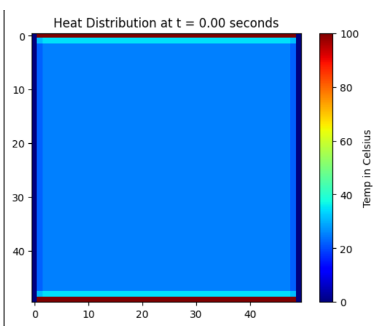
Solved the heat equation using FDM in 2D.

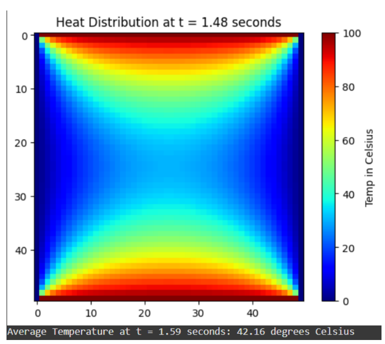
Read papers to gain some foundational knowledge about the task, followed by some blogs about CFD and heat equations in 1D and 2D and lastly followed some tutorials about this field again to gain a stronger understanding of the problem and to correctly implement its solution.

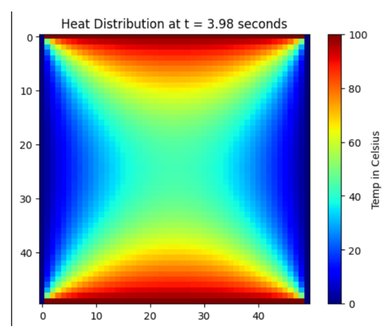
Faced some challenges like complexity of some papers, not having all the base knowledge needed at first and having to do the research which took quite some time.

0.1 Visualization

I have used matplotlib library to visualize heat convergence along a plate, here are the results:







Code

[33]
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[34]
```python
# Physical Parameters
plate_length = 50 #mm
plate_width = 50  #mm
alpha = 110 #How quickly heat spreads through the material
time = 4
nodes = 50
```

[35]
```python
# Numerical Parameters
dx = plate_length / nodes
dy = plate_width / nodes
dt = min((dx**2 / (4*alpha)),(dy**2 / (4*alpha)))
sim_time = int(time / dt)
```

[36]
```python
# Stability Check parameters
if dt > 0.5 * dx**2 / alpha:
  print("Stability Condition Not Satisfied")
else:
  print("Stability Condition Satisfied")
```

```
Stability Condition Satisfied
```

[37]
```python
u = np.zeros((nodes,nodes)) + 25 #Initial plate temp = 25 degrees

# Boundary Conditions

u[0, :] = 100  # Upper initial temp is 100
u[-1, :] = 100  # Lower initial temp is 100
u[:, 0] = 0
u[:, -1] = 0
```

2D Equation in continous form is ∂u/∂t = α(∂²u/∂x² + ∂²u/∂y²)

```python
from IPython.display import clear_output
fig = plt.figure()
plt.ion()


for t in range(sim_time):
  u_new = u.copy()
  for i in range(1, nodes-1):
    for j in range(1, nodes-1):
      u_new[i, j] = u[i, j] + (alpha * dt) * (u[i+1, j] + u[i-1, j] + u[i, j+1] + u[i, j-1] - 4*u[i, j]) / (dx**2 + dy**2) # Heat diffusion terms to achieve convergence
  u = u_new.copy()

  # Plot the current temperature distribution
  if t % 50 == 0:
    # Print average temp at each iteration
    print(f"Average Temperature at t = {t*dt:.2f} seconds: {np.mean(u):.2f} degrees Celsius")
    clear_output(wait=True)
    im = plt.imshow(u, cmap='jet', interpolation='nearest')
    plt.colorbar(im,label= 'Temp in Celsius')
    plt.title(f"Heat Distribution at t = {t*dt:.2f} seconds")
    plt.show()
```

Imported libraries. Initialized physical and numerical parameters. Made sure the stability condition is achieved. Nodes are used to like divide the plate into smaller parts to then have the heat transfer happen between each node and the other resulting in better results when having larger number of nodes as that is better off to replicate the real plate.