

ASSIGNMENT 2

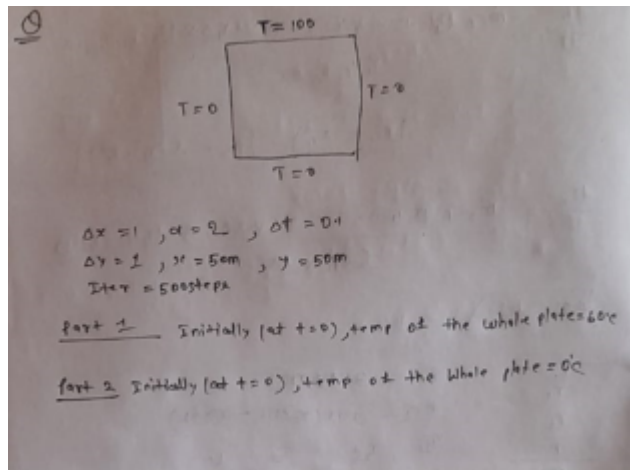
Name :- RAVI SHANKAR

Roll no :- 200778

TOPIC :- 2D Heat Equation Numerically using Python

2D PLATE

ASSIGNMENT QUESTION



CODE

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from matplotlib.animation import FuncAnimation

plate_length=int(input("enter the plate_length "))

u_top=float(input("enter the initial temp of top boundary "))
u_left=float(input("enter the initial temp of left boundary "))
u_bottom=float(input("enter the initial temp of bottom boundary "))
u_right=float(input("enter the initial temp of right boundary "))

u_initial =float(input("enter the initial temp of whole plate "))
```

```

max_iter_time=int(input("enter max_iter_time "))

alpha = 2
delta_x = 1

delta_t = (delta_x ** 2)/(4 * alpha)
gamma = (alpha * delta_t) / (delta_x ** 2)

u = np.empty((max_iter_time, plate_length, plate_length))

u.fill(u_initial)

u[:, (plate_length-1):, :] = u_top
u[:, :, :1] = u_left
u[:, :1, 1:] = u_bottom
u[:, :, (plate_length-1):] = u_right

def calculate(u):
    for k in range(0, max_iter_time-1, 1):
        for i in range(1, plate_length-1, delta_x):
            for j in range(1, plate_length-1, delta_x):
                u[k + 1, i, j] = gamma * (u[k][i+1][j] + u[k][i-1][j] + u[k][i][j+1] + u[k][i][j-1] - 4*u[k][i][j])
            + u[k][i][j]

    return u

def plotheatmap(u_k, k):

    plt.clf()

    plt.title(f"Temperature at t = {k*delta_t:.3f} unit time")
    plt.xlabel("x")
    plt.ylabel("y")

    plt.pcolormesh(u_k, cmap=plt.cm.jet, vmin=0, vmax=100)
    plt.colorbar()

    return plt

u = calculate(u)

def animate(k):
    plotheatmap(u[k], k)

anim = animation.FuncAnimation(plt.figure(), animate, interval=1, frames=max_iter_time,
repeat=False)
anim.save("heat_equation_solution.gif")

```

FINAL TEMPERATURE PROFILE

