REPORT

ASSIGNMENT 2, Python in ChemE

ARMEET LUTHRA

Problem statement

In this assignment we were required to find the TEMPERATURE PROFILE OVER TIME FOR a flat plate, subject to certain fixed boundary conditions and also animate the changes.

Attempted solution

Physics of problem

The heat conduction equation at steady state is given by:

$$\frac{dT}{dt} = \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right)$$

$$\alpha = \frac{k}{\rho Cp}, k = thermal\ conductivity, \rho = density\ of\ material, Cp = specific\ heat\ capacity$$

Algorithm

1) The flat plate is divided into square cells
2) We use the approximation:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T(x+\ \Delta x\ ,y) +\ T(x - \Delta x, y) - 2 * T(x,y)}{\Delta x^2}$$

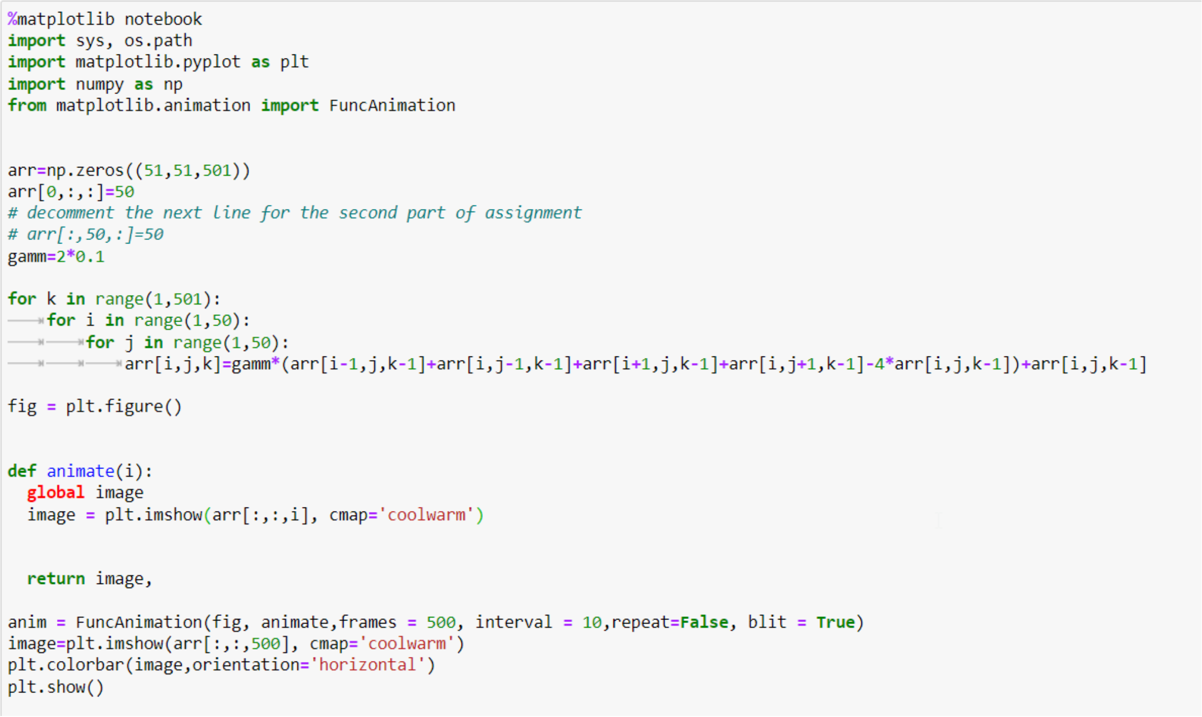
3) Using this approximation, the heat conduction equation becomes:

$$T(x,y,t+\Delta t) \\ = \gamma\big(T(x+\ \Delta x\ ,y,t)+\ T(x-\Delta x,y,t)+T(x,y-\Delta y,t) \\ +T(x,y+\Delta y,t)-4*T(x,y,t)\big)+T(x,y,t)$$

$$\gamma = \frac{\alpha\Delta t}{\Delta x^2}\,,\, assume\ \Delta x = \Delta y\ ,for\ numerical\ stability{:}\ \Delta t \leq \frac{\Delta x^2}{4\alpha}$$

4) We solve this using 3 loops where we apply the above equation for each cell and save the temperature value for different time stamps in a 3d array.
5) Using FuncAnimation we animate the temperature profile.

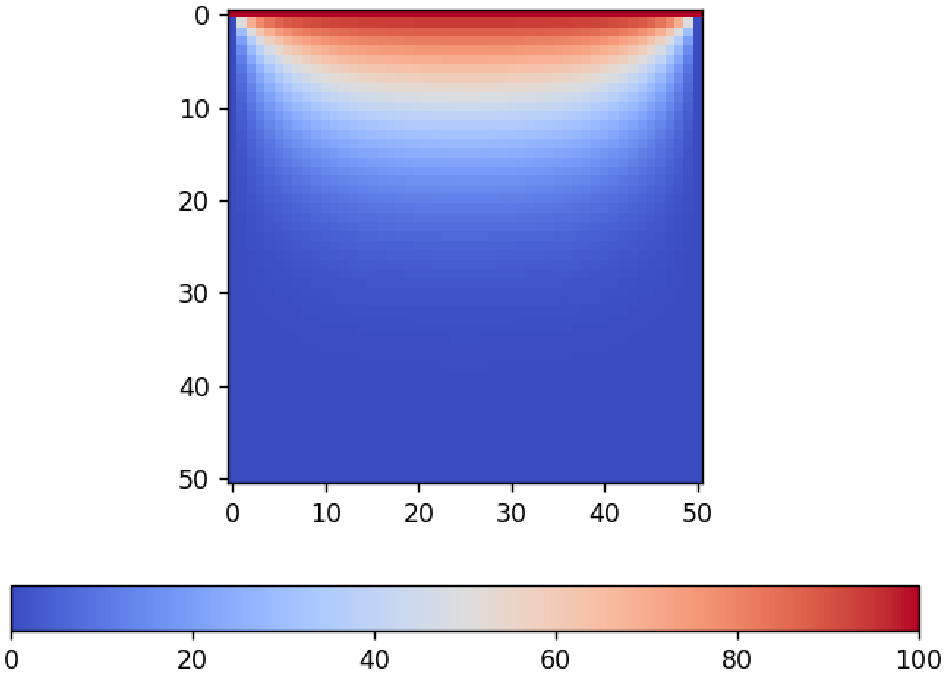
Code



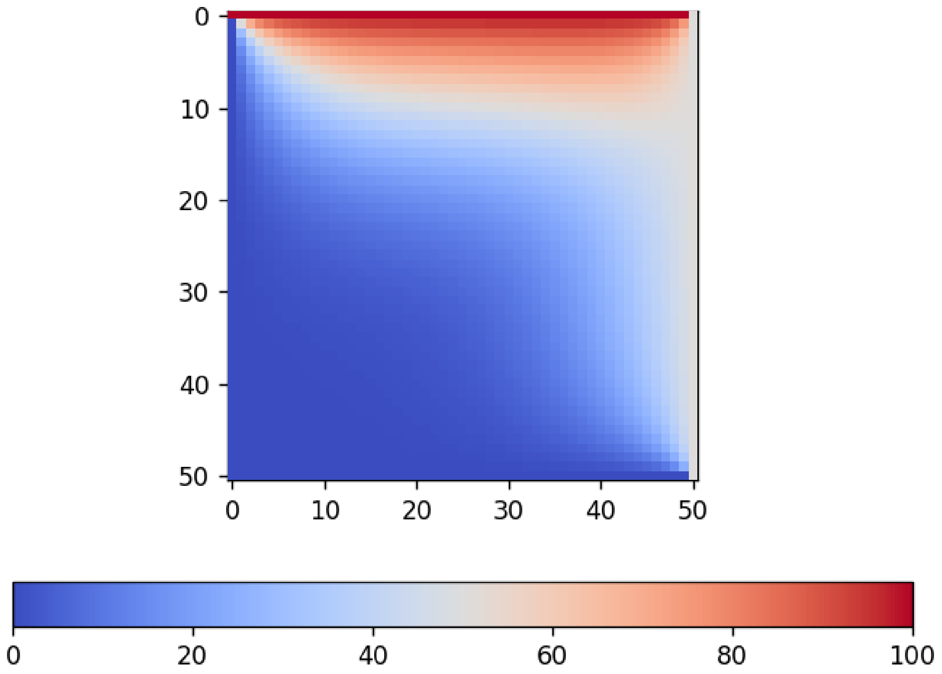
```
%matplotlib notebook
import sys, os.path
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation


arr=np.zeros((51,51,501))
arr[0,:,:]=50
# decomment the next line for the second part of assignment
# arr[:,50,:]=50
gamm=2*0.1

for k in range(1,501):
    for i in range(1,50):
        for j in range(1,50):
            arr[i,j,k]=gamm*(arr[i-1,j,k-1]+arr[i,j-1,k-1]+arr[i+1,j,k-1]+arr[i,j+1,k-1]-4*arr[i,j,k-1])+arr[i,j,k-1]

fig = plt.figure()


def animate(i):
  global image
  image = plt.imshow(arr[:,:,i], cmap='coolwarm')


  return image,

anim = FuncAnimation(fig, animate,frames = 500, interval = 10,repeat=False, blit = True)
image=plt.imshow(arr[:,:,500], cmap='coolwarm')
plt.colorbar(image,orientation='horizontal')
plt.show()
```

Output

1. Initial boundary condition: Top edge at 100°, rest plate at 0°.



2) Initial boundary condition: Top edge at 100°, Right edge at 50°, rest plate at 0°.


0
10
20
30
40
50
0
10
20
30
40
50
0
20
40
60
80
100