1 PSet-5.py

```
import numpy as np
import matplotlib.pyplot as plt

fileName = "Problem-Set-5"

num_time_steps = 100
delta = 1

# Grid points:
height = 30
width = 30

# Constants
rho = 3000              # kg/m^3
c = 840                 # J/(kg*C)
h = 28                  # W/(m^2*C)  Convective Heat Transfer Coefficient
k = 5.2                 # W/(m*C)    Thermal Conductivity
alpha = k / (rho * c)   # m^2/s      Thermal Diffusivity

dt = k * (delta ^ 2) / (2 * h * delta + 4 * k)  # Characteristic time
#dt = (delta ^ 2) / (4 * alpha)
Fo = alpha * dt / (delta ^ 2)                   # Fourier Number
Bi = h * delta / k                              # Biot Number

T_initial = 10
T_right = 38
T_inf = 0

# Create array and initialize to T-initial
data = np.zeros((width, height)) + T_initial

# Set the right boundary to T_right
for j in range(height):
```

```
    data[(width - 1), j] = T_right


for t in range(num_time_steps):
    data_old = data.copy()

    # Internal Nodes
    for m in range(1, width - 1):
        for n in range(1, height - 1):
            data[m, n] = (data_old[m + 1, n] + data_old[m - 1, n] + data_old[m, n + 1] + data_old[m, n - 1]) / 4

    # Convective Boundary Nodes (Left)
    for n in range(1, height - 1):
        m = 0
        data[m, n] = Fo * (2 * Bi * (T_inf - data_old[m, n]) + 2 * data_old[m + 1, n] + data_old[m, n + 1] + data_old[m, n - 1] - 4 * data_old[m, n]) + data_old[m, n]

    # Insulated Boundary Noes (Top)
    for m in range(1, width - 1):
        data[m, 0] = Fo * (2 * data_old[m, n - 1] + data_old[m - 1, n] + data_old[m + 1, n]) + (1 - 4 * Fo) * data_old[m, n]



# Print the data in the console (readable format)
#print(np.rot90(data))

data_printable = np.rot90(data) #np.flipud(np.rot90(data))

figNum = 1
plt.figure(figNum)
plt.axes().set_aspect('equal')
plt.style.use('classic')
heatmap = plt.pcolor(data_printable)
```

```
66
67 plt.text(0.5, -0.02, "T = " + str(T_initial) + "\N{DEGREE SIGN}C",
68          horizontalalignment='center',
69          verticalalignment='top',
70          rotation=0,
71          clip_on=False,
72          transform=plt.gca().transAxes)
73 plt.text(0, 0.5, "Convective Boundary",
74          horizontalalignment='right',
75          verticalalignment='center',
76          rotation=90,
77          clip_on=False,
78          transform=plt.gca().transAxes)
79 plt.text(0.5, 1, "Insulated Surface",
80          horizontalalignment='center',
81          verticalalignment='bottom',
82          rotation=0,
83          clip_on=False,
84          transform=plt.gca().transAxes)
85 plt.text(1, 0.5, "T = " + str(T_right) + "\N{DEGREE SIGN}C",
86          horizontalalignment='left',
87          verticalalignment='center',
88          rotation=270,
89          clip_on=False,
90          transform=plt.gca().transAxes)
91
92 plt.axis("off")
93
94 plt.xlim(0, width)
95 plt.ylim(0, height)
96
97 cbar = plt.colorbar(heatmap)
98 cbar.set_label("Temperature (\N{DEGREE SIGN}C)")
99 plt.clim(0, np.amax(data))
100
```

```
101 plt.savefig(fileName + "/images/" + fileName + "-" + str(figNum) + ".png")
102 plt.show()
```