Name : Mudit sand

Roll NO. : 203100068

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

```
## 1-D solution
## INPUT PARAMETERS
L = 1
alpha = 1/5000 # Random
dx = 0.1
dt = 1
t_curr = 0
t_final = 50
n = L/dx + 1
```

```
### Initial condition temp function along x
x_array = np.arange(0,L,dx)
initial_temp_func = '12*x + 10' #Random function taken
temp_0 = []
for x in x_array:
  temp_0.append(eval(initial_temp_func))
```

```
temp_0
```

```
[10.0,
 11.2,
 12.4,
 13.600000000000001,
 14.8,
 16.0,
 17.200000000000003,
 18.4,
 19.6,
 20.8]
```

```
plt.plot(x_array,temp_0)
```

```
##Defining boundary condition
temp_array = temp_0.copy()
temp_array[0] = 20
temp_array[-1] = 40 ## Random values

## deep copy this array to change value through formulae
temp_final = temp_array.copy()
```

```
## Step 4

while t_curr <= t_final:
  for i in range(1,len(x_array)-1):
    temp_final[i] = temp_array[i] + (alpha * (dt / (dx)**2)) * (temp_array[i + 1] - 2 * temp_array[i] + temp_array[i - 1])

  plt.plot(x_array, temp_final)
  temp_array = temp_final.copy()
  t_curr += dt
plt.xlabel('Length')
plt.ylabel('Temperature')
```

```
plt.show()
```

2 Dimensional Heat equation from FDM

```
## INPUT PARAMETERS
Lx = 1
Ly = 1.5
alpha = 1/5000 # Random
dx = 0.1
dy = 0.1
dt = 1
t_curr = 0
t_final = 500
nx = int(Lx/dx + 1)
ny = int(Ly/dy + 1)
```

```
## Initial condition assumend at uniform temp 40 degree centigrade
temp_0 = np.ones((nx,ny))
temp_0 = temp_0 * 40
temp_0
```

```
array([[40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.],
       [40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40., 40.,
        40., 40., 40.]])
```

```
import seaborn as sns
sns.heatmap(temp_0)
```

```
##Defining boundary condition
temp_array = temp_0.copy()
temp_array[0,:] = 60
temp_array[-1,:] = 80 ## Random values
temp_array[:,-1] = 45

## deep copy this array to change value through formulae
temp_final = temp_array.copy()
```

```
## Step 4

while t_curr <= t_final:
  for i in range(1,nx-1):
    for j in range(1,ny-1):
      temp_final[i,j] = temp_array[i,j] + ((alpha * (dt / (dx)**2)) * (temp_array[i + 1,j] - 2 * temp_array[i,j] + temp_array[i - 1,j])) + ((alph


  temp_array = temp_final.copy()
  t_curr += dt
```

```
sns.heatmap(temp_final,cmap='jet')
```

✓ 0s completed at 10:24 PM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.