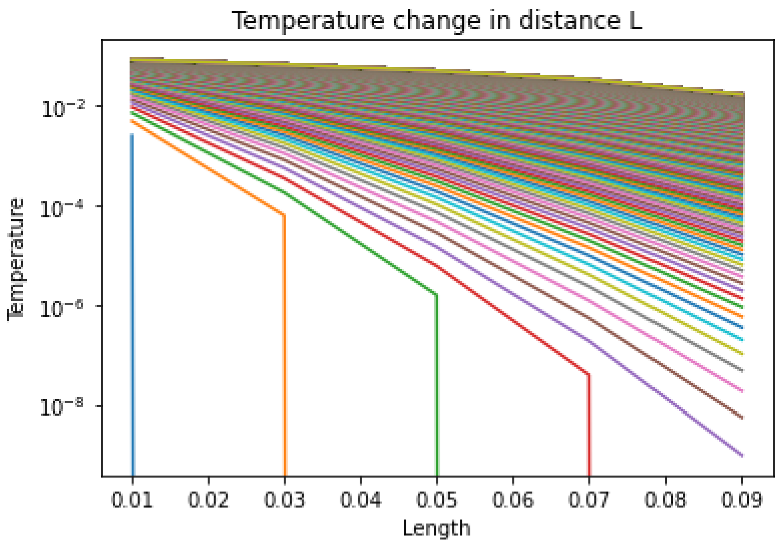
Python code for One dimensional heat equation

```
#we will first transform the one dimensional heat equation using the finite difference for
#then we impose start and end boundary condition values on the equtaion
#finally we plot the changes in temperature per distance
```

```
#importing important libraries
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")


#setting a random alpha value for the heat equation
cons=0.0001 #NOTE ,this can be changed accordingly based on the heat equation formulation

s=5  # x number of steps
T0=0 # setting initial condition temperature to start from zero

L=0.1  # critical length value
dx=L/s # change in step length
final_temp=100 #setting final temperature
dt=0.1 # time step


t=np.arange(0,final_temp,dt)
x=np.linspace(dx/2,L-dx/2,s) # setting x in the specified interval

T=np.ones(s)*T0 #converting initial Temp condition from scalar to vector
U=np.empty(s) #derivertive initialization


left_boundary_condition=L # left  boundary condition
right_boundary_condition=0 # right  boundary condition

for _ in range(1,len(t)):
  for k in range(1,s-1):

    #the finite difference transformation of the one dimensional heat Pde equation
    U[k]=((T[k+1]-T[k])/dx**2-(T[k]-T[k-1])/dx**2)*cons

    #applying the set boundary conditions
  U[s-1]=((right_boundary_condition-T[s-1])/dx**2-(T[s-1]-T[s-2])/dx**2)*cons
  U[0]=((T[1]-T[0])/dx**2-(T[0]-left_boundary_condition)/dx**2)*cons


  #plotting the results
  T+=U*dt
  plt.semilogy(x,T)
  plt.title('Temperature change in distance L')
  plt.ylabel('Temperature')
  plt.xlabel('Length')
```

```
# plt.legend()
```



0s completed at 3:28 PM