

```

#=====#

"""
    Projet_Simulation_Numerique_De_L'Equation_De_Chaleur
    ->En_Volume_Fini_Avec_Maillage_Non_Uniforme
    ->Yaya_Touré / L2 Modélisations_Mathématiques_Analyse_Et_Simulation_Numérique
    ->Centre_National_Calculs_Scientifique
"""

#=====#

#~~~~~Début_Projet~~~~~#
#Ceci est un encodage python / sur sublime text :->
#utf-8:coding
# Nous avons besoin que 2 Libraries
import numpy as np

import matplotlib.pyplot as plt

dx = 0.025
dt = 0.025

#Une vision de notre barre de longueur L (voir figure)
"""
t
^
|
|-----|
|
|----->x
"""

"""
    #sur l'axe des x,
    tient compte du nombres de points,
    qui se trouvent dans le domaine [0;1]
"""
x = np.arange(0, 1+dx, dx)

"""
    #sur l'axe des t,
    tient compte du nombres de points,
    qui se trouvent dans le domaine [0;1]
"""
t = np.arange(0, 0.1+dt, dt)

ConditionsAuxbors = [0, 0] #
ConditionInitial = np.sin(np.pi*x)

n = len(x)
m = len(t)
T = np.zeros((n,m))

#En tenant compte des conditions aux bords
T[0, :] = ConditionsAuxbors[0]
T[-1, :] = ConditionsAuxbors[1]
#Notre condition initial
T[:, 0] = ConditionInitial
#T.round(3)
Lamda = dt/dx**2

```

```

for j in range(1, m):
    for i in range(1, n-1):
        #codage de l'equation
        #T[i,0] = Lamda*T[i-1, j-1] + (1-2*Lamda)*T[i, j-1] + Lamda*T[i+1, j-1]
        T[i,j] = Lamda*T[i-1, j-1] + (1-2*Lamda)*T[i, j-1] + Lamda*T[i+1, j-1]
        #T[i,0] = Lamda*T[i-1,0] + (1-2*Lamda)*T[i, 0] + Lamda*T[i+1, 0]

```

```

#xi=centre(expl: xi=3), on lui dit fait le tour,
#Afin de tenir en compte chaque xi, en utilisant la fonction <<round()>>
T = T.round(3)
T

```

```

R = np.linspace(1,0,m)
B = np.linspace(0,1,m)
G = 0

```

```

for j in range(m):
    plt.plot(x, T[:, j], color = [R[j],G,B[j]]) # gestion d'Affichage

```

```

plt.xlabel('distance [m]')
plt.ylabel('Temperature [°C]')
plt.legend(f't = {value} s' for value in t)
plt.show() #rendre visible la figure

```

#~~~~~Fin_Projet~~~~~#