

```
# -*- coding: utf-8 -*-
# Fichier tp3.py

from numpy import * # importation du module numpy
from numpy.linalg import * # importation du module numpy.linalg
from matplotlib.pyplot import * # importation du module matplotlib.pyplot
from mpl_toolkits.mplot3d import Axes3D # importation du module mpl_toolkits.mplot3d

# Demande la méthode à utiliser

print("Choix du schéma pour calcul des U(j): ")
print("1- schéma explicite")
print("2- schéma implicite")
meth = int(input('Choix = '))

# Demande le nombre de points N

print("Choix du nombre N de points intérieurs de l'intervalle [0,1]")
N = int(input('N = '))

# Demande le pas en temps dt

print('Choix du pas dt en temps')
dt = float(input('dt = '))

# Demande le temps final T

print('Choix du temps final T')
T = float(input('T = '))

# Maillage

h = 1./(N + 1.)
X = arange(0.,1.,h)
Xh = X[1:N+1]
M = int((T/dt) - 1)

# Calcul de la matrice A du système

A = (2/h**2)*eye(N) - (1/h**2)*(diag(ones(N-1),1) + diag(ones(N-1),-1))

# Fonction définissant U0(x)

def U0(x):
    y = sin(pi*x) + 0.25*sin(10*pi*x)
    return y

# Fonction définissant la solution exacte de l'équation

def solex(x,t):
    z = sin(pi*x)*exp(-1*pi**2*t) + 0.25*sin(10*pi*x)*exp(-100*pi**2*t)
    return z

# Calcul du vecteur U des solutions exactes aux points du maillage
# Calcul itératif des vecteurs U(j) par le schéma choisi
# Calcul de l'erreur max sur i et j de la valeur absolue de U(x(i),t(j)) - Uh(x(i),t(j))

Uh = U0(Xh)
Err = amax(absolute(solex(Xh,0) - Uh))
if (meth == 1):
    for i in arange(1,M+1):
```

```
        Uh = Uh - dt*dot(A,Uh)
        U = solex(Xh,(i*dt))
        if (Err < amax(absolute(U - Uh))):
            Err = amax(absolute(U - Uh))

if (meth == 2):
    A = linalg.inv((dt*A) + eye(N))
    for i in arange(1,M+1):
        Uh = dot(Uh,A)
        U = solex(Xh,(i*dt))
        if (Err < amax(absolute(U - Uh))):
            Err = amax(absolute(U - Uh))

print("l'erreur max vaut : " , Err)

# Tracé du graphe de la fonction Uh(x,T)

plot(Xh, Uh, label = 'sol approchee')
if (meth == 1):
    title('Uh(x,T) Methode explicite, T=' + str(T))
if (meth == 1):
    title('Uh(x,T) Methode implicite, T=' + str(T))
xlabel('x')
ylabel('Uh(x,T)')
legend()
show()
```