```python
# -*- 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
import time
from pylab import *

def U0(X):
    Y = zeros(shape(X))
    Y=sin(pi*X)+.25*sin(10.*pi*X)
    return Y
def U1(X):
    Y = zeros(shape(X))
    return Y
def solex(X,ct):
    return sin(pi*X)*cos(ct*pi)+.25*sin(10.*pi*X)*cos(10.*ct*pi)

print('Choix de la vitesse de transport c')
#c = float(input('c = '))
c = -2

Ns = 500
h = 1./(Ns + 1.)


X = linspace(0.,1.,Ns+1)
Xh = X[0:Ns]

dt = .001
T=1.
M = int((T/dt) - 1)

meth = 2
#Uj temps actuel
#Ujm temps precedent
#Tjn temps suivant
Uj = U0(Xh)
Ujm = zeros(shape(U0))
Ujn = zeros(shape(U0))

#Iteration 1

Ujn = Uj+dt*U1(Xh)

Uj, Ujm = Ujm, Uj
Ujn, Uj = Uj, Ujn

A= diag(-ones(Ns-1),1)-diag(ones(Ns-1),-1)+2.*eye(Ns)
A=A/h/h
#Erreur
Err = 0
Errn = 0

#line1, = plot(linspace(0,1,100), solex(linspace(0,1,100),T), label = 'sol exacte')

for j in arange(1, M):
    if( meth == 1):
        for i in arange(1,Ns):
                Ujn = 2.*Uj-Ujm-c*c*dt*dt*(A.dot(Uj))
```

```
    if( meth == 2 ):
        Ujn = solve( (eye(Ns)+c*c*dt*dt*A), 2.*Uj-Ujm)

    #Calcul de l'erreur
    U=solex(Xh,j*dt*c)
    Errn = amax(absolute(U - Ujn))
    if (Err < Errn):
        Err = Errn

    Uj, Ujm = Ujm, Uj
    Ujn, Uj = Uj, Ujn

plot(Xh, Uj,label="Approché")
plot(linspace(0,1,500),solex(linspace(0,1,500),T*c),label='exacte')

xlabel('X')
ylabel('Y')
legend()
show()
disp(Err)
```