

```

1  # -*- coding: utf-8 -*-
2  # Fichier tp3.py
3
4  from numpy import * # importation du module numpy
5  from numpy.linalg import * # importation du module numpy.linalg
6  from matplotlib.pyplot import * # importation du module matplotlib.pyplot
7  from mpl_toolkits.mplot3d import Axes3D # importation du module mpl_toolkits.mplot3d
8
9  # Demande la méthode à utiliser
10
11 print("Choix du schéma pour calcul des U(j): ")
12 print("1- schéma explicite")
13 print("2- schéma implicite")
14 meth = int(input('Choix = '))
15
16 # Demande le nombres de points N
17
18 print("Choix du nombre N de points interieurs de l'intervalle [0,1]")
19 N = int(input('N = '))
20
21 # Demande le pas en temps dt
22
23 print('Choix du pas dt en temps')
24 dt = float(input('dt = '))
25
26 # Demande le temps final T
27
28 print('Choix du temps final T')
29 T = float(input('T = '))
30
31 # Maillage
32
33 h = 1./(N + 1.)
34 X = arange(0.,1.,h)
35 Xh = X[1:N+1]
36 M = int((T/dt) - 1)
37
38 # Calcul de la matrice A du système
39
40 A = (2/h**2)*eye(N) - (1/h**2)*(diag(ones(N-1),1) + diag(ones(N-1),-1))
41
42 # Fonction définissant U0(x)
43
44 def U0(x):
45     y = sin(pi*x) + 0.25*sin(10*pi*x)
46     return y
47
48 # Fonction définissant la solution exacte de l'équation
49
50 def solex(x,t):
51     z = sin(pi*x)*exp(-1*pi**2*t) + 0.25*sin(10*pi*x)*exp(-100*pi**2*t)
52     return z
53
54 # Calcul du vecteur U des solutions exactes aux points du maillage
55 # Calcul iteratif des vecteurs U(j) par le schéma choisi
56 # Calcul de l'erreur max sur i et j de la valeur absolue de U(x(i),t(j)) -
    Uh(x(i),t(j))
57
58 Uh = U0(Xh)
59 Err = amax(absolute(solex(Xh,0) - Uh))
60 if (meth == 1):
61     for i in arange(1,M+1):
62         Uh = Uh - dt*dot(A,Uh)
63         U = solex(Xh,(i*dt))
64         if (Err < amax(absolute(U - Uh))):
65             Err = amax(absolute(U - Uh))
66
67 if (meth == 2):
68     A = linalg.inv((dt*A) + eye(N))
69     for i in arange(1,M+1):
70         Uh = dot(Uh,A)
71         U = solex(Xh,(i*dt))

```

```
72         if (Err < amax(absolute(U - Uh))):
73             Err = amax(absolute(U - Uh))
74
75     print("l'erreur max vaut : " , Err)
76
77     # Tracé du graphe de la fonction Uh(x,T)
78
79     plot(Xh, Uh, label = 'sol approchee')
80     if (meth == 1):
81         title('Uh(x,T) Methode explicite, T=' + str(T))
82     if (meth == 2):
83         title('Uh(x,T) Methode implicite, T=' + str(T))
84     xlabel('x')
85     ylabel('Uh(x,T)')
86     legend()
87     show()
88
```