```python
# -*- coding: utf-8 -*-

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


Ns = 20

# Maillage

h = 1./(Ns + 1)
X = linspace(0,1,Ns+2)
Xh = X[1:Ns+1]

# Matrice du système linéaire

A = -1*(diag(ones(Ns*Ns-3),3) + diag(ones(Ns*Ns-3),-3))

B = 4*eye(Ns) -1*(diag(ones(Ns-1),1) + diag(ones(Ns-1),-1))

for i in arange(0,Ns):
  A[Ns*i:Ns*i+Ns,Ns*i:Ns*i+Ns] = B

def Ud(x):
  y = sin(2*pi*x)*sinh(2*pi)
  return y

# Fonction définissant la solution exacte de l'équation

def solex(x, y):
  z = sin(2*pi*x)*sinh(2*pi*y)
  return z

# Second membre

b = zeros(Ns*Ns)
b[Ns*(Ns-1):Ns*Ns] = Ud(Xh)

# Resolution du systeme lineaire

A_inv = linalg.inv(A)
Uh = solve(A_inv, b)

#Mise en forme de la matrice Zh
Zh = array( 0*Ud(X))
for i in arange (0, Ns, 1):
  newrow = Uh[ i*(Ns):i*Ns+Ns]
  newrow =concatenate([[0], newrow, [0]])
  Zh = vstack([newrow, Zh])
Zh = vstack([Ud(X), Zh])


#Calcul du maillage
coordX, coordY= np.meshgrid(X, flip(X,0))

#Calcul de la solution exacte sur le maillage
U = solex(coordX,coordY)

#Calcul de l'erreur
Err = amax(absolute(U-Zh))

fig = figure()
ax = Axes3D(fig, azim = 30, elev = 30)
ax.plot_surface(coordX, coordY, Zh, cmap = cm.jet)
#ax.plot_surface(coordX, coordY, U, cmap = cm.jet)
fig.show()

```