Université Abou Bekr Belkaid Tlemcen — **2019-2020**
Département de Mathématiques — **Fiche TP1**
Module: Analyse numérique 2

Exercice : Soit le système linéaire $Ax = b$ suivant :

$$\begin{pmatrix} 2 & -1 & 3 & -3 \\ 1 & -1 & 2 & -1 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 4 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -20 \\ -8 \\ -2 \\ 4 \end{pmatrix}$$

1. Résoudre le système linéaire par la méthode de GAUSS avec pivot.
2. Factoriser la matrice A, calculer le déterminant de A et calculer A^{-1}.

Solution :

```python
import numpy as np
import numpy.linalg as alg
def LU(A):
    n=A.shape[0]
    L,U=np.identity(n),np.zeros((n,n))
    U=np.array(A)
    for k in range(n-1):
        for i in range(k+1,n):
            L[i,k]=U[i,k]/U[k,k]
            U[i,k+1:n]=U[i,k+1:n]-L[i,k]*U[k,k+1:n]
            U[i,k]=0
    return L,U
def gauss(A,b):
    n=A.shape[0]
    L,U=LU(A)
    b1=np.array(b)
    for k in range(n-1):
        for i in range(k+1,n):
            b1[i]= b1[i]-L[i,k]*b1[k]
    return b1
def remente(U,b):
    n=U.shape[0]
    x=np.zeros((n,1))
    x[n-1]=b[n-1]/U[n-1,n-1]
    for i  in  range(n-2,-1,-1):
        x[i]=(b[i]-np.dot(U[i,i+1:n],x[i+1:n]))/U[i,i]
    return x
def descente(L,b):
    n=L.shape[0]
    x=np.zeros((n,1))
    x[0]=b[0]/L[0,0]
    for i  in  range(1,n):
        x[i]=(b[i]-np.dot(L[i,0:i],x[0:i]))/L[i,i]
    return x
def det(A):
    det=1
    L,U=LU(A)
    for i in range(A.shape[0]):
        det*=U[i,i]
    return det
def inv(A) :
    n=A.shape[0]
    L,U=LU(A)
    inv=np.zeros((n,n))
    for j in range(n):
        inv[:,[j]]=remente(U,descente(L,np.identity(n)[:,[j]]))
    return inv
A=np.array([[2,-1,3,-3],[1,-1,2,-1],[1,-2,1,0],[1,-1,4,3]],dtype='float')
b=np.array([[-20],[-8],[-2],[4]])
L,U=LU(A)
print("La matrice L est\n",L,"\nLa matrice U est \n",U)
print("Méthode de Gauss: la solution est:\n",remente(U,gauss(A,b)))
print("Numpy: la solution est:\n",alg.solve(A,b))
print("Factorisation LU: determinant de A est " ,det(A))
print("Factorisation LU: l'inverse de A est \n",inv(A))
print("Numpy: l'inverse de A est\n",alg.inv(A))
```

```
In [1]:

La matrice L est
 [[ 1.   0.   0.   0. ]
 [ 0.5  1.   0.   0. ]
 [ 0.5  3.   1.   0. ]
 [ 0.5  1.  -1.   1. ]]
La matrice U est
 [[ 2.  -1.   3.  -3. ]
 [ 0.  -0.5  0.5  0.5]
 [ 0.   0.  -2.   0. ]
 [ 0.   0.   0.   4. ]]
Méthode de Gauss: la solution est:
 [[-4. ]
 [-1.5]
 [-1. ]
 [ 3.5]]
Numpy: la solution est:
 [[-4. ]
 [-1.5]
 [-1. ]
 [ 3.5]]
Factorisation LU: determinant de A est 8.0

Factorisation LU: l'inverse de A est
 [[ 2.   -4.5   1.    0.5 ]
 [ 0.75 -1.5  -0.25  0.25]
 [-0.5   1.5  -0.5  -0.  ]
 [ 0.25 -1.    0.25  0.25]]
Numpy: l'inverse de A est
 [[ 2.   -4.5   1.    0.5 ]
 [ 0.75 -1.5  -0.25  0.25]
 [-0.5   1.5  -0.5   0.  ]
 [ 0.25 -1.    0.25  0.25]]
```