

```

# ===== #
# ===== #
# ALGEBRE LINEAIRE #
# Décomposition LU #
# JANVIER 2016 #
# ===== #
# ===== #

from numpy import array, identity, zeros, dot
from numpy.linalg import det

def ValidLU(M):
    """
    Fonction renvoyant un booléen True/False selon que la matrice M (supposée
    inversible) admet ou non une décomposition LU.
    """
    (n,p) = M.shape
    for i in range(1,n-1):
        # L'un des déterminants principaux (hormis le déterminant de la
        # matrice) est nul.
        if det(M[:i,:i]) == 0:
            return(False)
    return(True)

def DecompoLU_Ident(M):
    """
    Fonction effectuant la décomposition LU d'une matrice M supposée inversible
    par identification.
    """
    (n,p) = M.shape
    # La matrice n'est pas carrée.
    if n != p:
        return("La matrice n'est pas carrée !")
    # La matrice est carrée.
    elif det(M) == 0:
        # La matrice M n'est pas inversible.
        return("La matrice n'est pas inversible !")
    # La matrice est inversible.
    elif ValidLU(M):
        # La matrice admet une décomposition LU (unique !).
        # Initialisation de L et U
        L = identity(n)
        U = zeros((n,n))
        # Coeur de l'algorithme
        for i in range(n):
            # Détermination de la i-ème ligne de U
            for j in range(i,n):
                # On peut calculer la somme via un produit matriciel...
                S = dot(L[i,:i],U[:i,j])
                # OU de façon plus classique !
                """
                S = 0
                for k in range(i):
                    S += L[i,k]*U[k,j]
                """
                U[i,j] = M[i,j] - S
            # Détermination de la i-ème colonne de L
            # Remarque : quand i prend la valeur n-1, le for ci-dessous ne fait
            # ... rien !
            for j in range(i+1,n):
                # On peut calculer la somme via un produit matriciel...
                S = dot(L[j,:i],U[:i,i])
                # OU de façon plus classique !
                """
                S = 0
                for k in range(i):
                    S += L[j,k]*U[k,i]

```

```

        """
        L[j,i] = (M[j,i] - S)/U[i,i]
    return(L,U)
else:
    return("La matrice n'admet pas de décomposition LU !")

# Matrices inversibles admettant une décomposition LU
A1 = array([[1,2],[0,3]],dtype=float)
A2 = array([[3,1,1],[0,1,2],[1,1,1]],dtype=float)
A3 = array([[2,1,-1,1],[0,-1,2,2],[4,3,-3,-3],[2,3,-6,3]],dtype=float)
A4 = array([[5,-1,2],[4,3,-3],[0,2,4]],dtype=float)

# Matrices inversibles n'admettant pas de décomposition LU
B1 = array([[0,3],[1,2]]) # Le coefficient B1(1,1) est nul...
B2 = array([[0,0,1],[0,1,0],[0,0,1]]) # idem...
B3 = array([[1,-3,11],[2,-1,7],[-4,-3,1]]) # B3 n'est pas inversible

(L,U)=DecompoLU_Ident(A3)
print("La matrice initiale")
print(A3)
print("La matrice L")
print(L)
print("La matrice U")
print(U)
print("Check")
print(dot(L,U))

```