

Pregunta 3

April 25, 2019

Autor

Lazaro Camasca Edson
Analisis Numerico
Universidad Nacional de Ingenieria

Programe el método de factorización LU y aplíquelo la matriz $A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$

```
In [1]: from toolNick import *  
import numpy as np
```

toolNick se ha importado correctamente.

```
In [2]: def factorizacion_LU(A, p = False, v = False):  
    #v nos muestra el procedimiento detallado de la factorizacion  
    #p se utiliza si requiere pivotacion total al inicio  
    line = "=====  
    fil, col = A.shape #Guarda el #filas y #columnas  
    if p:  
        A = pivoteo_Total(A, v)  
  
    #Creamos una lista para almacenar las matrices L y P  
    L_list = []  
    P_list = [] #Matrices de permutacion  
  
    for i in range(fil): # se condiera dim(a)  
        if A[i, i] == 0 :  
            P = pivoteo(A, i)  
  
            if v : print("P_{}:\n{}".format(i+1,P))  
  
            A = P @ A  
            P_list.append(P)  
  
    else: #En caso de no permutar agragamos la identidad
```

```

        P_list.append(np.identity(fil))

        #Obtenemos el L_i
        L_i = get_L(A, i, v)
        L_list.append(L_i)

        A = L_i @ A

        #mostrar L(i) * (A)
        if v : print("L_{} * A:\n{}\n{}".format(i+1,A,line))

    U = A
    L,P = get_L_and_P_LU(L_list, P_list)
    print("L_:\n{}\nU_:\n{}\nP_:\n{}".format(L,U,P))

    return L@U

In [3]: A = np.array([[2, -1, 0],
                      [1, 2, -1],
                      [0, -1, 2]])

        A

Out[3]: array([[ 2, -1,  0],
               [ 1,  2, -1],
               [ 0, -1,  2]])

In [4]: factorizacion_LU(A, v=True)

alpha_1:
[[ 0. ]
 [-0.5]
 [ 0. ]]
alpha_1 * e_1:
[[ 0.  0.  0. ]
 [-0.5 -0. -0. ]
 [ 0.  0.  0. ]]
L_1:
[[1.  0.  0. ]
 [0.5 1.  0. ]
 [0.  0.  1. ]]
L_1 * A:
[[ 2. -1.  0. ]
 [ 2.  1.5 -1. ]
 [ 0. -1.  2. ]]
=====
alpha_2:
[[0.  ]
 [0.  ]

```

```

[0.66666667]]
alpha_2 * e_2:
[[0.          0.          0.          ]
 [0.          0.          0.          ]
 [0.          0.66666667 0.          ]]
L_2:
[[ 1.          0.          0.          ]
 [ 0.          1.          0.          ]
 [ 0.          -0.66666667 1.          ]]
L_2 * A:
[[ 2.          -1.          0.          ]
 [ 2.          1.5         -1.          ]
 [-1.33333333 -2.          2.66666667]]
=====
alpha_3:
[[0.]
 [0.]
 [0.]]
alpha_3 * e_3:
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
L_3:
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
L_3 * A:
[[ 2.          -1.          0.          ]
 [ 2.          1.5         -1.          ]
 [-1.33333333 -2.          2.66666667]]
=====
L_:
[[ 1.          0.          0.          ]
 [-0.5         1.          0.          ]
 [ 0.          0.66666667 1.          ]]
U_:
[[ 2.          -1.          0.          ]
 [ 2.          1.5         -1.          ]
 [-1.33333333 -2.          2.66666667]]
P_:
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

Out[4]: array([[ 2., -1.,  0.],
               [ 1.,  2., -1.],
               [ 0., -1.,  2.]])

```