Pregunta 4

April 25, 2019

Autor:
Lazaro Camasca Edson
Analisis Numerico
Universidad Nacional de Ingenieria

Programe el método de eliminación gaussiana y aplíquelo para obtener la factorización LU de la matriz

$$A=\begin{pmatrix} 4 & -1 & 1/2 \\ -1 & 4 & -1 \\ 1/2 & -1 & 4 \end{pmatrix}$$

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

```
toolNick se ha importado correctamente.
```

```
In [2]: def factorizacion_LU(A, p = False, v = False):

            return L@U
```

Eliminacion Gaussiana
Similar al anterior, pero especificamos el metodo de eliminacion gausiana

```
In [ ]: def get_L(A, i, v=False):
```

Esta funcion nos permite: Encontrar la matriz de transformacion de gauss,L_i Para una solución detallada, pasar v = True como parámetro.

Guarda el filas y columnas

```
In [ ]: (fil, col) = A.shape
```

Hallar alfa

```
In [ ]: alpha_i = np.zeros(fil) # alpha_i es declarado fila
            alpha_i[i + 1:] = - A[i + 1: fil, i] / A[i, i]
            alpha_i = alpha_i.reshape(fil, 1) # transforma alpha_i en columna
```

Hallar e_i

```
In [ ]: e_i = np.zeros(fil); e_i[i] = 1
```

Hallar L_i

```
In [ ]: L_i = np.identity(fil) - alpha_i * e_i
```

0.1 Calculado en la siguiente matriz

```
In [2]: A = np.array([[4, -1, 0.5],
                      [-1, 4, -1],
                      [-0.5, -1, 4]])
        A
```

```
Out[2]: array([[ 4. , -1. ,  0.5],
               [-1. ,  4. , -1. ],
               [-0.5, -1. ,  4. ]])
```

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

```
alpha_1:
[[0.   ]
 [0.25 ]
 [0.125]]
alpha_1 * e_1:
[[0.    0.    0.   ]
 [0.25  0.    0.   ]
 [0.125 0.    0.   ]]
L_1:
[[ 1.     0.     0.   ]
 [-0.25   1.     0.   ]
 [-0.125  0.     1.   ]]
L_1 * A:
[[ 4.     -1.      0.5   ]
 [-2.      4.25   -1.125 ]
 [-1.     -0.875   3.9375]]
=========================================
alpha_2:
[[0.        ]
 [0.        ]
 [0.20588235]]
alpha_2 * e_2:
[[0.         0.         0.        ]
 [0.         0.         0.        ]
 [0.         0.20588235 0.        ]]
L_2:
[[ 1.          0.          0.        ]
 [ 0.          1.          0.        ]
```

```
 [ 0.         -0.20588235  1.        ]]
L_2 * A:
[[ 4.         -1.          0.5       ]
 [-2.          4.25       -1.125     ]
 [-0.58823529 -1.75        4.16911765]]
=========================================
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:
[[ 4.         -1.          0.5       ]
 [-2.          4.25       -1.125     ]
 [-0.58823529 -1.75        4.16911765]]
=========================================
L_:
[[1.         0.         0.        ]
 [0.25       1.         0.        ]
 [0.125      0.20588235 1.        ]]
U_:
[[ 4.         -1.          0.5       ]
 [-2.          4.25       -1.125     ]
 [-0.58823529 -1.75        4.16911765]]
P_:
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

```
Out[4]: array([[ 4. , -1. ,  0.5],
               [-1. ,  4. , -1. ],
               [-0.5, -1. ,  4. ]])
```