gaxpy

April 29, 2019

1 Ejercicio clase 20 de Marzo, 2019

1.1 Gaxpy

Equipo 9

```
In [1]: import numpy as np
```

```
In [2]: A = np.array([
            [3,1,1,1],
            [2,1,1,1],
            [1,5,1,1],
            [7,9,8,4]])
        B = np.array([ -9, 6, 2, 9 ])
        print(A,B)
```

```
[[3 1 1 1]
 [2 1 1 1]
 [1 5 1 1]
 [7 9 8 4]] [-9  6  2  9]
```

Resultado de Gaxpy

```
In [3]: def resolver_gaxpy(A, B):
            N = B.shape[0]
            if N != A.shape[1]:
                raise Exception("Error: no es una matriz cuadrada")
        #     if not np.allclose(L, np.tril(L)):
        #         raise Exception("Error: no es una matriz triangular inferior")

            X = np.zeros((N,1))
            B_ = B.copy().astype(np.double)
            A_ = A.copy().astype(np.double)
            L_ = np.eye(N)
            U_ = np.zeros((N,N))
            V_ = np.zeros(N)
            for i in range(0,N):
```

```
            if i == 0:
                V_ = A_[:, 0]
            else:
                â = A_[:,i]
                z = np.linalg.solve(L_[0:i, 0:i], â[0:i])
                U_[0:i,i] = z
                V_[i:N] = â[i:N] - np.dot(L_[i:N, 0:i],z)
            U_[i,i] = V_[i]
            L_[(i+1):N,i] = V_[(i+1):N]/V_[i]

        return L_, U_
    (L,U) = resolver_gaxpy(A, B)
    L, U
```

```
Out[3]: (array([[ 1.        ,  0.        ,  0.        ,  0.        ],
                [ 0.66666667,  1.        ,  0.        ,  0.        ],
                [ 0.33333333, 14.        ,  1.        ,  0.        ],
                [ 2.33333333, 20.        ,  0.25      ,  1.        ]]),
         array([[ 3.        ,  1.        ,  1.        ,  1.        ],
                [ 0.        ,  0.33333333,  0.33333333,  0.33333333],
                [ 0.        ,  0.        , -4.        , -4.        ],
                [ 0.        ,  0.        ,  0.        , -4.        ]]))
```

Dado que

$$A = LU$$

```
In [4]: L@U
```

```
Out[4]: array([[3., 1., 1., 1.],
               [2., 1., 1., 1.],
               [1., 5., 1., 1.],
               [7., 9., 8., 4.]])
```

```
In [5]: A
```

```
Out[5]: array([[3, 1, 1, 1],
               [2, 1, 1, 1],
               [1, 5, 1, 1],
               [7, 9, 8, 4]])
```