

# ejercicio\_clase\_marzo13

March 13, 2019

## 0.1 Ejercicio 2:

## 0.2 Algoritmo de factorización LU en su versión outer product

```
In [620]: A=np.matrix([[1.0,2.0,1.0],[2.0,2.0,3.0],[-1.0,-3.0,0.0]])  
A
```

```
Out[620]: matrix([[ 1.,  2.,  1.],  
                  [ 2.,  2.,  3.],  
                  [-1., -3.,  0.]])
```

```
In [621]: n= A.shape[0]  
print(n)
```

3

```
In [622]: for k in range(0,n-1):  
  
    print(k)  
    print("k+1=",k+1)  
    print("n+1=",n+1)  
    A[k+1:n,k] = (A[k+1:n,k]/A[k,k])  
  
    A[k+1:n,k+1:n]=A[k+1:n,k+1:n]-A[k+1:n,k]*A[k,k+1:n]  
  
    #print(A)
```

```
0  
k+1= 1  
n+1= 4  
1  
k+1= 2  
n+1= 4
```

```
In [623]: A
```

```
Out[623]: matrix([[ 1. ,  2. ,  1. ],
                  [ 2. , -2. ,  1. ],
                  [-1. ,  0.5,  0.5]])
```

#### COMPROBACION

```
In [626]: L=np.matrix([[1,0,0],[2,1,0],[-1,0.5,1]])
          U=np.matrix([[1,2,1],[0,-2,1],[0,0,0.5]])
          L*U
```

```
Out[626]: matrix([[ 1.,  2.,  1.],
                  [ 2.,  2.,  3.],
                  [-1., -3.,  0.]])
```