ejercicio_clase_eliminacion_gaussiana_2

March 13, 2019

0.1 Resolver sust hacia adelante en su versión "saxpy" de el sistema:

```
In [2]: import numpy as np
        A=np.array([[3,0,0,0],[2,1,0,0],[1,5,1,0],[7,9,8,4]])
        b=np.array([-9,6,2,5])
        print(A)
        print(b)

[[3 0 0 0]
 [2 1 0 0]
 [1 5 1 0]
 [7 9 8 4]]
[-9  6  2  5]
```

```
In [3]: x=np.zeros((4,1))
        x

Out[3]: array([[0.],
               [0.],
               [0.],
               [0.]])
```

```
In [4]: N=A.shape[0]
        N

Out[4]: 4
```

```
In [5]: n=N

        for i in range(0,N):
            x[i,0] = b[0]/A[0,0]
            b=b[1:n] - x[i,0]*A[1:n,0]
            A=A[1:n,1:n]
            n=A.shape[0]
```

```
In [6]: x
```

```
Out[6]: array([[ -3. ],
               [ 12. ],
               [-55. ],
               [ 89.5]])
```

El vector anterior es el resultado al que se había llegado en clase.

0.2 Resolver con sust hacia adelante en su versión saxpy por bloques

```
In [39]: A=np.matrix([[3,0,0,0],[2,1,0,0],[1,5,1,0],[7,9,8,4]])
         B=np.matrix([[-9,12],[6,-1],[2,0],[5,1]])
         print(A)
         print(B)

[[3 0 0 0]
 [2 1 0 0]
 [1 5 1 0]
 [7 9 8 4]]
[[-9 12]
 [ 6 -1]
 [ 2  0]
 [ 5  1]]
```

El resultado al que hay que llegar es:

```
In [40]: np.linalg.solve(A,B)

Out[40]: matrix([[ -3. ,   4. ],
                 [ 12. ,  -9. ],
                 [-55. ,  41. ],
                 [ 89.5, -68.5]])

In [41]: X=np.matrix(np.zeros((4,2)))
         print(X)

[[0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]]


In [42]: tam=2 #tamño del bloque es tam
         n=int(A.shape[0]/tam) #numero total de bloques por columna o por renglón

In [43]: for j in np.linspace(start=0,stop=A.shape[0]-tam,num=n,dtype=int):
             Ljj=A[j:j+tam,j:j+tam]

             X[j:j+tam,:] = np.linalg.solve(Ljj, B[j:j+tam,:])
```

```
            for i in range(j+1,n):

                B[i-1+tam:i-1+tam+tam,0:tam]

                A[i-1+tam:i-1+tam+tam,0:tam]

                X[j:j+tam,:]

                B[i-1+tam:i-1+tam+tam,0:tam]=B[i-1+tam:i-1+tam+tam,0:tam] - A[i-1+tam:i-1+tam
```

```
In [44]: X
```

```
Out[44]: matrix([[ -3. ,   4. ],
                 [ 12. ,  -9. ],
                 [-55. ,  41. ],
                 [ 89.5, -68.5]])
```

Con lo anterior vemos que sí se llegó al resultado.

0.3 Resolver con eliminación gaussiana simple :

```
In [483]: A=np.array([[1,2,1],[2,2,3],[-1,-3,0]])
          A
```

```
Out[483]: array([[ 1,  2,  1],
                 [ 2,  2,  3],
                 [-1, -3,  0]])
```

```
In [484]: b=(np.array([0,3,2])).reshape((3,1))
          b.shape
```

```
Out[484]: (3, 1)
```

```
In [485]: x=np.zeros((3,1))
          x.shape
```

```
Out[485]: (3, 1)
```

```
In [486]: n= A.shape[0] -1
          print(n)
```

```
2
```

```
In [487]: for k in range(0,n):
           print("k=",k)
           pivote = A[k,k]
           print("pivote=",pivote)
```

```
        i = np.linspace(start=k+1,stop=n,num=n-(k+1)+1,endpoint=True,dtype=int)
        print("i=",i)
        print("lk=",A[i,k]/A[k,k])
        lk = (A[i,k]/A[k,k])
        print("A[k,i]=",A[k,i])

        A[k+1:n+1,:]=A[k+1:n+1,:]-lk*np.array([A[k,:],A[k,:]])



        print("A=",A)

        b[k+1:n+1]=b[k+1:n+1] - lk*np.array([b[k],b[k]])


        print("cambio de ciclo")
```

```
k= 0
pivote= 1
i= [1 2]
lk= [ 2. -1.]
A[k,i]= [2 1]
```

```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

<ipython-input-487-d770b9d87daa> in <module>()
     10  print("A[k,i]=",A[k,i])
     11
---> 12  A[k+1:n+1,:]=A[k+1:n+1,:]-lk*np.array([A[k,:],A[k,:]])
     13
     14


ValueError: operands could not be broadcast together with shapes (2,) (2,3)
```