

Tarea 3

Métodos Numéricos

Benjamin Rivera

20 de septiembre de 2020

Índice

1. Tarea 3	2
1.0.1. Como ejecutar	2
1.1. Ejercicio 1	3
1.1.1.	3
1.1.2.	3
1.2. Ejercicio 2	4
1.3. Ejercicio 3	7

1. Tarea 3

Tarea 3 de Benjamín Rivera para el curso de **Métodos Numéricos** impartido por Joaquín Peña Acevedo. Fecha límite de entrega **20 de Septiembre de 2020**.

1.0.1. Como ejecutar

Requerimientos Este programa se ejecuto en mi computadora con la version de **Python 3.8.2** y con estos [requerimientos](#)

Jupyter En caso de tener acceso a un *servidor jupyter* ,con los requerimientos antes mencionados, unicamente basta con ejecutar todas las celdas de este *notebook*. Probablemente no todas las celdas de *markdown* produzcan el mismo resultado por las [Nbextensions](#).

Consola Habrá archivos e instrucciones para poder ejecutar cada uno de los ejercicios desde la consola.

Si todo sale mal En caso de que todo salga mal, tratare de dejar una copia disponible en **GoogleColab** que se pueda ejecutar con su versión de **Python**.

1.1. Ejercicio 1

1.1.1. .

Muestre que la inversa de L_k es $L_k^{-1} = I + l_k e_k^T$. Así, la inversa de una matriz triangular inferior elemental es otra matriz triangular inferior elemental.

Sabemos que, si la matriz es invertible, entonces $L_k L_k^{-1} = I$.

Procedemos por contradicción. Supongamos que $L_k^{-1} \neq I + l_k e_k^T$, de manera que $L_k(I + l_k e_k^T) \neq I$; si desarrollamos lo anterior resulta que

$$\begin{aligned} L_k(I + l_k e_k^T) &\neq I \\ (I - l_k e_k^T)(I + l_k e_k^T) &\neq I \\ I^2 - (l_k e_k^T)^2 &\neq I \\ I - (l_k e_k^T)^2 &\neq I \end{aligned}$$

lo cual es cierto siempre que $(l_k e_k^T)^2 \neq \bar{0}$. Sin embargo, por la manera en que están definidas $l_k e_k^T$, $(l_k e_k^T)^2 = 0$. Por lo que llegamos a una contradicción y $L_k^{-1} = I + l_k e_k^T$.

1.1.2. .

Muestre que $L_{k-1}^{-1} L_k^{-1} = I + l_{k-1} e_{k-1}^T + l_k e_k^T$ y con esto demostrar que

$$L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1} = I + \sum_{k=1}^{n-1} l_k e_k^T$$

por lo que L es una matriz triangular inferior con 1's en la diagonal

Parte 1 Por definición y el punto anterior

$$\begin{aligned} L_{k-1}^{-1} L_k^{-1} &= (I + l_{k-1} e_{k-1}^T)(I + l_k e_k^T) \\ &= I^2 + l_{k-1} e_{k-1}^T (l_k e_k^T) + l_{k-1} e_{k-1}^T + l_k e_k^T \\ &\quad \text{y como sabemos que la multiplicación de las submatrices } l, e \text{ es } 0 \\ &= I + \bar{0} + l_{k-1} e_{k-1}^T + l_k e_k^T \\ &= I + l_{k-1} e_{k-1}^T + l_k e_k^T \end{aligned}$$

Parte 2 ...

1.2. Ejercicio 2

```
[2]: def show1D( x,sz,/, max_sz=8, prnt=True):
    """ Funcion que recibe un arreglo 1D y, en caso de
    que no sea muy grande, lo imprime en pantalla. En
    caso de que lo sea, imprime datos representativos.
    Input:
        x := Apuntador al arreglo 1D para imprimir
        sz := Tamano del arreglo. (para la imple_
            mentacion en python es inecesario, pe_
            ro lo solicita el ejercicio)
        max_sz := Maximo de elementos a imprimir
        prnt := Indica si el string obtenido se
            debe imprimir en pantalla
    Output:
        Esta funcion regresa
    _Doctest:
        >>> show1D([1,2,3,8,5,6,7,8,9,0], 10, prnt=False)
        '1, 2, 3, 8, ... , 7, 8, 9, 0'

        >>> show1D([1,2,3,8], 4, prnt=False)
        '1, 2, 3, 8'
    """
    return ret
```

```
[3]: def forwardSubstitution( L, n, b, t,/,dtype=np.float64):
    """ Funcion que implementa el metodo desustitucion pa_
    ra adelante dada una matriz triangular inferior.
    Input:
        L := apuntador a la matriz L
        n := tamano de la matriz L
        b := apuntador al vector b
        t := tolerancia de 0's
    Otput:
        ret := vector x de respuestas si se encontro, o
            None en caso de que hay habido algun error
    _Doctest:
        >>> forwardSubstitution(np.matrix([[1,0],[0,1]]), 1, np.
        ↪matrix([[3,4]]), 0.1)
        matrix([[3.],
                [4.]])
    """

    if len(L) == b.size:
        return np.asmatrix(x, dtype=dtype).transpose()
    else:
        return None
```

```
[4]: def readFile(file,/, path='datos/npv/', ext='.npv', dtype=np.
      ↪float64):
      return np.asmatrix(np.load(file= str(path+file+ext),

def Ejercicio2(f_L, f_b):
    """ Programa para ejecutar el ejercicio 2. espera el nombre
    de los archivos de la matriz y el vector soluciones para
    funcionar.
    """
    L = readFile(f_L)
    b = readFile(f_b)
    dtype = np.float64
    t = (2.2e-16)**(2/3) # De la tarea 1 y como estamos usando
    ↪float64

    if True:
        for sz in [5, 50, 500]:
            Ejercicio2('matL'+str(sz), 'vecb'+str(sz))
            print('\n')
```

```
5 5
[ 0.33006882]
[-0.78428134]
[ 0.70227498]
[ 0.48272233]
[-0.61319569]
Error =
12.755173555608748
```

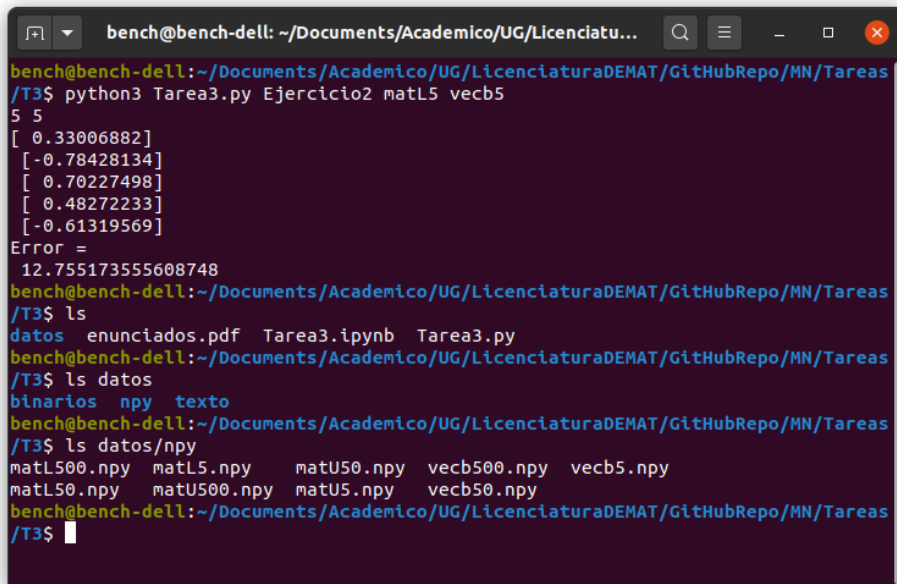
```
50 50
Err: 0 o casi cero en diagonal
El sistema no tiene solucion.
Y es singular.
```

```
500 500
[ 1.27622082]
[ 1.39905809]
[-3.80410368]
[ 2.83344826], ... , [ 4.01907914e+110]
[ 8.50132313e+110]
[-6.48257593e+110]
[ 1.66110939e+111]
Error =
8.245074985591727e+111
```

Como ejecutar Para ejecutar en consola se necesita estar en el mismo directorio que la carpeta de la tarea datos. Como nombre de los archivos no se espera la direccion completa ni la extension del archivo. Se infiere automaticamente que los datos sera obtenidos del subdirectorio datos/npv y con la extension .npv. De manera qu la ejecucion debe parecerse a lo siguiente

```
python3 Tarea3.py Ejercicio2 matL5 vecb5
```

lo que debe producir la salida



```
bench@bench-dell: ~/Documents/Academico/UG/Licenciaturu...
/T3$ python3 Tarea3.py Ejercicio2 matL5 vecb5
5 5
[ 0.33006882]
[-0.78428134]
[ 0.70227498]
[ 0.48272233]
[-0.61319569]
Error =
12.755173555608748
bench@bench-dell:~/Documents/Academico/UG/LicenciaturaDEMAT/GitHubRepo/MN/Tareas
/T3$ ls
datos enunciados.pdf Tarea3.ipynb Tarea3.py
bench@bench-dell:~/Documents/Academico/UG/LicenciaturaDEMAT/GitHubRepo/MN/Tareas
/T3$ ls datos
binarios npv texto
bench@bench-dell:~/Documents/Academico/UG/LicenciaturaDEMAT/GitHubRepo/MN/Tareas
/T3$ ls datos/npv
matL500.npv matL5.npv matU50.npv vecb500.npv vecb5.npv
matL50.npv matU500.npv matU5.npv vecb50.npv
bench@bench-dell:~/Documents/Academico/UG/LicenciaturaDEMAT/GitHubRepo/MN/Tareas
/T3$
```

Otros [Norma de Numpy](#)

1.3. Ejercicio 3

```
[5]: def backwardSubstitution( U, n, b, t,/,dtype=np.float64):  
    """ Funcion que implementa el metodo desustitucion pa_  
ra atras dada una matriz triangular inferior.  
    Input:  
        U := apuntador a la matriz U  
        n := tamaño de la matriz U  
        b := apuntador al vector b  
        t := tolerancia de 0's  
    Output:  
        ret := vector x de respuestas si se encontro, o  
        None en caso de que hay habido algun error  
    """  
    return None  
  
[8]: def Ejercicio3(f_U, f_b):  
    """ Programa para ejecutar el ejercicio 3. espera el nombre  
de los archivos de la matriz y el vector soluciones para  
funcionar.  
    """  
  
    if True:  
        for sz in [5, 50, 500]:  
            Ejercicio3('matU'+str(sz), 'vecb'+str(sz))  
            print('\n')
```

5 5

Err: 0 o casi cero en diagonal

El sistema no tiene solucion.

Y es singular.

50 50

```
[ 1.76160471e+15]  
[-4.50692072e+15]  
[ 7.27394101e+13]  
[ 6.62574710e+14], ... , [-2.96030701]  
[ 1.04582814]  
[ 2.64891114]  
[-0.38446395]
```

Error =

```
15.155075012507066
```

500 500

```
[ 7.41601169e+141]  
[-4.27114460e+142]  
[-1.15567637e+141]  
[ 6.22209212e+141], ... , [-3.67045535]  
[ 1.29204355]  
[ 0.70500787]
```

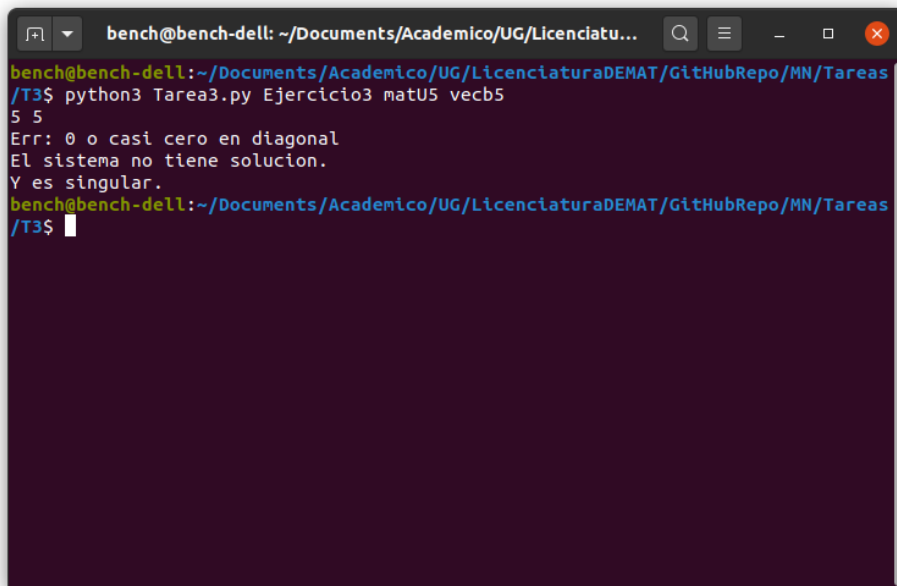
```
[ 1.29749623]
Error =
1.9978749809209134e+128
```

Como ejecutar

Como ejecutar Para ejecutar en consola se necesita estar en el mismo directorio que la carpeta de la tarea datos. Como nombre de los archivos no se espera la direccion completa ni la extension del archivo. Se infiere automaticamente que los datos sera obtenidos del subdirectorio datos/.npz y con la extension .npz. De manera que la ejecucion debe parecerse a lo siguiente

```
python3 Tarea3.py Ejercicio3 matU5 vecb5
```

lo que debe producir la salida



```
bench@bench-dell: ~/Documents/Academico/UG/LicenciaturaDEMAT/GitHubRepo/MN/Tareas
/T3$ python3 Tarea3.py Ejercicio3 matU5 vecb5
5 5
Err: 0 o casi cero en diagonal
El sistema no tiene solucion.
Y es singular.
bench@bench-dell:~/Documents/Academico/UG/LicenciaturaDEMAT/GitHubRepo/MN/Tareas
/T3$
```