

## Computación

Aula Virtual: <https://famaf.aulavirtual.unc.edu.ar/course/view.php?id=747>

Resguardo tutoriales: <https://www.famaf.unc.edu.ar/~moreschi/docencia/Computacion/>

### Tutorial Problema 3 de la Guía N° 5

**Problema 3:** Escriba una función en Python llamada `egauss` que implemente el algoritmo de eliminación Gaussiana y retorne una matriz triangular superior  $U$  y un vector  $y$  tal que  $Ux = y$  es equivalente a  $Ax = b$ . La función debe tener como entrada la matriz  $A$  y el vector  $b$ . Guarde la función en el archivo “guia5.py”.

#### Tutorial:

- Guarde en el archivo `p3.py` las siguientes instrucciones:

```
1
2 import numpy as np
3
4 # leemos arreglos de archivos para usar en testeos -----
5 A = np.loadtxt("datos/array_int_6x6_A.dat", dtype=np.int)
6 #A = np.loadtxt("datos/array_float_8x8_A.dat", dtype=np.float128)
7 print('A=')
8 print(A)
9 print('type(A)=', type(A))
10 print('A[1,1]=', A[1,1])
11 print('type(A[1,1])=', type(A[1,1]))
12
13 print()
14 b = np.loadtxt("datos/array_int_6x6_b.dat", dtype=np.int)
15 #b = np.loadtxt("datos/array_float_8x8_b.dat", dtype=np.float128)
16 print('b=')
17 print(b)
18 print('type(b)=', type(b))
19 print('b[1]=', b[1])
20 print('type(b[1])=', type(b[1]))
21
22 print()
23 b2 = np.transpose(b)
24 print('b2 = np.transpose(b) =')
25 print(b2)
26 print('type(b2)=', type(b2))
27 print('b2[1]=', b2[1])
28 print('type(b2[1])=', type(b2[1]))
29
30 # dimensión del problema
31 print()
32 N = len(b)
33 print('N =', N)
34 print()
35
```

```

36 # -----
37
38 def egauss(AA,bb):
39     nn = len(bb)
40     A2 = 1.*np.copy(AA) # esto hace que si la matriz tiene enteros,
41                          # no moleste
42     b2 = 1.*np.copy(bb) # esto hace que si la matriz tiene enteros,
43                          # no moleste
44
45     #A2 = np.copy(AA)
46     #b2 = np.copy(bb)
47     filacambiada = []
48     for k in range(nn-1):
49         if (abs(A2[k][k]) <1.e-10):
50             filan = np.copy(A2[k,:])
51             A2[k,:] = A2[k+1,:]
52             A2[k+1,:] = filan
53             filan = np.copy(b2[k])
54             b2[k] = b2[k+1]
55             b2[k+1] = filan
56             filacambiada.append(k)
57         for i in range(k + 1, nn):
58             em = -A2[i][k] / A2[k][k]
59             b2[i] = b2[i] + em * b2[k]
60             A2[i][:] = A2[i][:] + A2[k][:] * em
61     print('filacambiada =',filacambiada)
62     if (abs(A2[nn-1][nn-1]) <1.e-10):
63         print('Atención: La matriz tiene último elemento de la diagonal\
64               muy pequeño.')
65     return A2, b2
66
67 """
68 Pruebe de cambiar la definición de A2 y b2 sin usar 1.*, para
69 datos con entradas enteras, en vez de punto flotante. Notará
70 con la matriz de 6x6 que hay inestabilidad numérica; a pesar
71 que python3 trata los cocientes de enteros como float.
72 """
73 # -----
74 # testeos
75
76 # notar que en la próxima ejecución se imprime: 'filacambiada ='
77 U, bs = egauss(A,b)
78
79 print()
80 print(' U =',U)
81 print(' bs =',bs)
82
83 # -----
84 # hacemos uso del código hecho precedentemente grabado en guia5.py
85 from guia5 import trsup
86
87 x = trsup(U,bs)
88 print()
89 print(' la solución usando el método de Gauss es:')
90 print('      x=',x)
91

```

```

92 #-----
93 from scipy.linalg import solve
94
95 xsc = solve(A, b)
96 print()
97 print(' la solución usando (scipy.linalg solve) es:')
98 print('   xsc=', xsc)
99 print()
100
101 print(' segundo chequeo:')
102 xscsup = solve(U, bs)
103 print(' xscsup=', xscsup)
104 print()
105
106 #-----
107 print(' Esto debería ser cero:')
108
109 print('   np.dot(A, x) - b =', np.dot(A, x) - b)
110 print()
111 print('   np.dot(A, xsc) - b =', np.dot(A, xsc) - b)
112 print()
113 print(' np.dot(A, xscsup) - b =', np.dot(A, xscsup) - b)
114
115 print()
116 print(' ¿Es suficientemente cerca de cero?')

```

- Crear el subdirectorio datos.
- Guarde en el archivo datos/array\_int\_6x6\_A.dat los siguientes datos:

```

1  2 -60  13  62 -29 -36
2  74 -64  22  26 -67  34
3  38  4 -33 -28  78  46
4  63 -77  3  60 -7 -68
5  16 -22  75 -57 -12  0
6 -2  91  80 -20 -9  45

```

- Guarde en el archivo datos/array\_int\_6x6\_b.dat los siguientes datos:

```

1 -23  71  37  20 -40 -34

```

- Guarde en el archivo datos/array\_float\_8x8\_A.dat los siguientes datos:

```

1 -9.97767787 -0.64753585 -3.70971741 -2.87798655  2.13669268 -2.60726176  2.55800501  8.40907991
2 -8.80113154  4.76482543  3.62453937  3.57787486  2.79109368  5.22459831  4.22710241  5.79936931
3 -4.79590658 -0.1319525  0.82075613 -8.85692047 -2.80388128  2.8801654  6.65329085 -9.25686548
4 7.99468074  2.27949564  6.83503322 -7.39496915 -7.33866531  9.30035835 -0.31156596  7.49248931
5 9.39527155 -3.71993612  3.70829887 -6.97098755 -5.09961001 -1.45161831  2.76310624 -0.30995156
6 3.11737289  2.66676431 -1.94085424  7.76199252 -9.34875905  4.5856315 -8.5956168  7.27482296
7 6.11367461  3.16664267  2.01439378  9.2836568  0.78876935 -0.13831311 -2.48420319 -7.30951381
8 -6.91713659  0.9681366  5.99039025 -2.12293836 -7.29898362 -0.82001581 -7.75894713 -5.83576998

```

- Guarde en el archivo datos/array\_float\_8x8\_b.dat los siguientes datos:

```

1 0.14877337  4.58615183  8.68606466 -5.21007811 -0.92711778  7.16338501 -0.39250293 -7.87375225

```

- Desde la terminal ejecute:  
`python3 p3.py`  
e interprete el resultado.  
Alternativamente ejecute:  
`python3`  
y vaya agregando uno a uno los bloques del programa.
- ¿Dieron bien los tests?
- Guarde la función `egauss` en el archivo `guia5.py`.
- Altere el programa para probar distintas cosas; en particular utilice los dos conjuntos de matrices.