ESCUELA POLITÉCNICA NACIONAL
MÉTODOS NUMÉRICOS

Leandro Bravo GR1CC

Tarea 10 - Descomposición LU

```
%load_ext autoreload
import numpy as np
from src import multiplicar_matrices, descomposicion_LU, resolver_LU
from src import eliminacion_gaussiana_L,eliminacion_gaussiana_U, determinante, inversa
```

Conjunto de Ejercicios

1. Realice las siguientes multiplicaciones matriz-matriz:

a.

$$\begin{bmatrix} 2 & -3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 1 & 5 \\ 2 & 0 \end{bmatrix}$$

```
%autoreload 2
A = np.array([[2,-3],[3,-1]])
B = np.array([[1,5],[2,0]])
C = multiplicar_matrices(A,B)
print("El resultado de la multiplicación es: \n",C)
```

```
El resultado de la multiplicación es:
 [[-4 10]
 [ 1 15]]
```

b.

$$\begin{bmatrix} 2 & -3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 1 & 5 & -4 \\ -3 & 2 & 0 \end{bmatrix}$$

```
%autoreload 2
A = np.array([[2,-3],[3,-1]])
B = np.array([[1,5,-4],[-3,2,0]])
C = multiplicar_matrices(A,B)
print("El resultado de la multiplicación es: \n",C)
```

```
El resultado de la multiplicación es:
 [[ 11   4  -8]
 [  6  13 -12]]
```

c.

$$\begin{bmatrix} 2 & -3 & 1 \\ 4 & 3 & 0 \\ 5 & 2 & -4 \end{bmatrix} \begin{bmatrix} 0 & 1 & -2 \\ 1 & 0 & -1 \\ 2 & 3 & -2 \end{bmatrix}$$

```
%autoreload 2
A = np.array([[2,-3,1],[4,3,0],[5,2,-4]])
B = np.array([[0,1,-2],[1,0,-1],[2,3,-2]])
C = multiplicar_matrices(A,B)
print("El resultado de la multiplicación es: \n",C)
```

```
El resultado de la multiplicación es:
 [[ -1   5  -3]
 [  3   4 -11]
 [ -6  -7  -4]]
```

d.

$$\begin{bmatrix} 2 & 1 & 2 \\ -2 & 3 & 0 \\ 2 & -1 & 3 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ -4 & 1 \\ 0 & 2 \end{bmatrix}$$

```
%autoreload 2
A = np.array([[2,1,2],[-2,3,0],[2,-1,3]])
B = np.array([[1,-2],[-4,1],[0,2]])
C = multiplicar_matrices(A,B)
print("El resultado de la multiplicación es: \n",C)
```

```
El resultado de la multiplicación es:
 [[ -2   1]
 [-14   7]
 [  6   1]]
```

2. Determine cuáles de las siguientes matrices son no singulares y calcule la inversa de esas matrices:

$$\begin{bmatrix} 4 & 2 & 6 \\ 3 & 0 & 7 \\ -2 & -1 & -3 \end{bmatrix}$$

```python
%autoreload 2
A = np.array([[4,2,6],[3,0,7],[-2,-1,-3]])
det = determinante(A)
if(det==0):
    print("El determinante es:",det, "\nPor tanto, la matriz es singular y no posee inversa")
else:
    inv = inversa(A)
    print("El determinante es:",det, "\nSu matriz inversa es la siguiente:", inv)
```

```
El determinante es: 0
Por tanto, la matriz es singular y no posee inversa
```

$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & -1 \\ 3 & 1 & 1 \end{bmatrix}$$

```python
%autoreload 2
A = np.array([[1,2,0],[2,1,-1],[3,1,1]])
det = determinante(A)
if(det==0):
    print("El determinante es:",det, "\nPor tanto, la matriz es singular y no posee inversa")
else:
    inv = inversa(A)
    print("El determinante es:",det, "\nSu matriz inversa es la siguiente:\n", inv)
```

```
El determinante es: -7.999999999999999
Su matriz inversa es la siguiente:
 [[-0.25   0.25   0.25 ]
 [ 0.625 -0.125 -0.125]
 [ 0.125 -0.625  0.375]]
```

c. $$\begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & 2 & -4 & -2 \\ 2 & 1 & 1 & 5 \\ -1 & 0 & -2 & -4 \end{bmatrix}$$

```python
%autoreload 2
A = np.array([[1,1,-1,1],[1,2,-4,-2],[2,1,1,5],[-1,0,-2,-4]])
det = determinante(A)
if(det==0):
    print("El determinante es:",det, "\nPor tanto, la matriz es singular y no posee inversa")
else:
    inv = inversa(A)
    print("El determinante es:",det, "\nSu matriz inversa es la siguiente:", inv)
```

```
El determinante es: 0
Por tanto, la matriz es singular y no posee inversa
```

d. $$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 6 & 7 & 0 & 0 \\ 9 & 11 & 1 & 0 \\ 5 & 4 & 1 & 1 \end{bmatrix}$$

```python
%autoreload 2
A = np.array([[4,0,0,0],[6,7,0,0],[9,11,1,0],[5,4,1,1]])
det = determinante(A)
if(det==0):
    print("El determinante es:",det, "\nPor tanto, la matriz es singular y no posee inversa")
else:
    inv = inversa(A)
    print("El determinante es:",det, "\nSu matriz inversa es la siguiente:\n", inv)
```

```
El determinante es: 27.999999999999993
Su matriz inversa es la siguiente:
```

```
[[ 0.25        0.          0.          0.        ]
 [-0.21428571  0.14285714  0.          0.        ]
 [ 0.10714286 -1.57142857  1.          0.        ]
 [-0.5         1.         -1.          1.        ]]
```

3. Resuelva los sistemas lineales 4 x 4 que tienen la misma matriz de coeficientes:

$$\begin{aligned} x_1 - x_2 + 2x_3 - x_4 &= 6, & x_1 - x_2 + 2x_3 - x_4 &= 1, \\ x_1 \qquad - x_3 + x_4 &= 4, & x_1 \qquad - x_3 + x_4 &= 1, \\ 2x_1 + x_2 + 3x_3 - 4x_4 &= -2, & 2x_1 + x_2 + 3x_3 - 4x_4 &= 2, \\ -x_2 + x_3 - x_4 &= 5, & -x_2 + x_3 - x_4 &= -1, \end{aligned}$$

Ya que ambos sistemas poseen la misma matriz de coeficientes

$$A = \begin{bmatrix} 1 & -1 & 2 & -1 \\ 1 & 0 & -1 & 1 \\ 2 & 1 & 3 & -4 \\ 0 & -1 & 1 & -1 \end{bmatrix}$$

primero realizaré la descomposición LU para usarla posteriormente para resolver cada sistema.

```
%autoreload 2
A = [[1,-1,2,-1],[1,0,-1,1],[2,1,3,-4],[0,-1,1,-1]]
L,U = descomposicion_LU(A)
print("Matriz L:\n",L)
print("Matriz U:\n",U)
```

```
Matriz L:
 [[ 1.    0.    0.    0.  ]
 [ 1.    1.    0.    0.  ]
 [ 2.    3.    1.    0.  ]
 [ 0.   -1.   -0.25  1.  ]]
Matriz U:
 [[ 1. -1.  2. -1.]
 [ 0.  1. -3.  2.]
 [ 0.  0.  8. -8.]
 [ 0.  0.  0. -1.]]
```

```
%autoreload 2
b_1 = [6,4,-2,5]
sol_1 = resolver_LU(L,U,b_1)
```

```
Calculando y
y
[ 6. -2. -8.  1.]
Verificación Ly=b:
[ 6.  4. -2.  5.]
Calculando x
x
[ 3. -6. -2. -1.]
Verificación Ux=y:
[ 6. -2. -8.  1.]
```

```
%autoreload 2
b_2 = [1,1,2,-1]
sol_2 = resolver_LU(L,U,b_2)
```

```
Calculando y
y
[ 1.  0.  0. -1.]
Verificación Ly=b:
[ 1.  1.  2. -1.]
Calculando x
x
[1. 1. 1. 1.]
Verificación Ux=y:
[ 1.  0.  0. -1.]
```

Por tanto, la soluciones obtenidas son: $sol_1 = [3, -6, -2, -1]$ y $sol_2 = [1, 1, 1, 1]$

4. Encuentre los valores de A que hacen que la siguiente matriz sea singular

$$A = \begin{bmatrix} 1 & -1 & \alpha \\ 2 & 2 & 1 \\ 0 & \alpha & -\frac{3}{2} \end{bmatrix}$$

Determinante de la matriz A

$$\begin{vmatrix} 1 & -1 & a \\ 2 & 2 & 1 \\ 0 & a & \frac{-3}{2} \end{vmatrix} = 2a^2 - a - 6$$

image.png

Para que cumpla

$$2a^2 - a - 6 = 0$$

Solución

$$a = 2,\ a = -\frac{3}{2}$$

image.png

5. Resuelva los siguientes sistemas lineales:

a. $$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & 1 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

```
%autoreload 2
L = np.array([[1,0,0],[2,1,0],[-1,0,1]])
U = np.array([[2,3,-1],[0,-2,1],[0,0,3]])
b = [2,-1,1]
resolver_LU(L,U,b)
```

```
Calculando y
y
[ 2. -5.  3.]
Verificación Ly=b:
[ 2. -1.  1.]
Calculando x
x
[-3.  3.  1.]
Verificación Ux=y:
[ 2. -5.  3.]
```

Por tanto, la solución es:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ 1 \end{bmatrix}$$

b. $$\begin{bmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ 3 & 2 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ 0 \end{bmatrix}$$

```
%autoreload 2
L = np.array([[2,0,0],[-1,1,0],[3,2,-1]])
U = np.array([[1,1,1],[0,1,2],[0,0,1]])
b = [-1,3,0]
resolver_LU(L,U,b)
```

```
Calculando y
y
[-0.5  2.5  3.5]
Verificación Ly=b:
[-1.  3.  0.]
Calculando x
x
[ 0.5 -4.5  3.5]
Verificación Ux=y:
[-0.5  2.5  3.5]
```

Por tanto, la solución es:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -4.5 \\ 3.5 \end{bmatrix}$$

6. Factorice las siguientes matrices en la descomposición LU mediante el algoritmo de factorización LU con $l_{ii} = 1$ para todas las i.

a. $$\begin{bmatrix} 2 & -1 & 1 \\ 3 & 3 & 9 \\ 3 & 3 & 5 \end{bmatrix}$$

```
%autoreload 2
A = [[2,-1,1],[3,3,9],[3,3,5]]
L,U = descomposicion_LU(A)
print("Matriz L:\n",L)
print("Matriz U:\n",U)
```

```
Matriz L:
 [[1.  0.  0. ]
 [1.5 1.  0. ]
 [1.5 1.  1. ]]
Matriz U:
 [[ 2.  -1.   1. ]
 [ 0.   4.5  7.5]
 [ 0.   0.  -4. ]]
```

b. $$\begin{bmatrix} 1.012 & -2.132 & 3.104 \\ -2.132 & 4.096 & -7.013 \\ 3.104 & -7.013 & 0.014 \end{bmatrix}$$

```
%autoreload 2
B = [[1.012,-2.132,3.104],[-2.132,4.096,-7.013],[3.104,-7.013,0.014]]
L,U = descomposicion_LU(B)
print("Matriz L:\n",L)
print("Matriz U:\n",U)
```

```
Matriz L:
 [[ 1.          0.          0.        ]
 [-2.10671937  1.          0.        ]
 [ 3.06719368  1.19775553  1.        ]]
Matriz U:
 [[ 1.012      -2.132       3.104     ]
 [ 0.         -0.39552569 -0.47374308]
 [ 0.          0.         -8.93914077]]
```

c. $$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1.5 & 0 & 0 \\ 0 & -3 & 0.5 & 0 \\ 2 & -2 & 1 & 1 \end{bmatrix}$$

```
%autoreload 2
C = [[2,0,0,0],[1,1.5,0,0],[0,-3,0.5,0],[2,-2,1,1]]
L,U = descomposicion_LU(C)
print("Matriz L:\n",L)
print("Matriz U:\n",U)
```

```
Matriz L:
 [[ 1.          0.          0.          0.        ]
 [ 0.5         1.          0.          0.        ]
 [ 0.         -2.          1.          0.        ]
 [ 1.         -1.33333333  2.          1.        ]]
Matriz U:
 [[2.  0.  0.  0. ]
 [0.  1.5 0.  0. ]
 [0.  0.  0.5 0. ]
 [0.  0.  0.  1. ]]
```

d. $$\begin{bmatrix} 2.1756 & 4.0231 & -2.1732 & 5.1967 \\ -4.0231 & 6.0000 & 0 & 1.1973 \\ -1.0000 & -5.2107 & 1.1111 & 0 \\ 6.0235 & 7.0000 & 0 & -4.1561 \end{bmatrix}$$

```python
%autoreload 2
D = [[2.1756,4.0231,-2.1732,5.1967],[-4.0231,6.0000,0,1.1973],
     [-1.0000,-5.2107,1.1111,0],[6.0235,7.0000,0,-4.1561]]
L,U = descomposicion_LU(D)
print("Matriz L:\n",L)
print("Matriz U:\n",U)
```

```
Matriz L:
 [[ 1.          0.          0.          0.        ]
 [-1.84919103  1.          0.          0.        ]
 [-0.45964332 -0.25012194  1.          0.        ]
 [ 2.76866152 -0.30794361 -5.35228302  1.        ]]
Matriz U:
 [[ 2.17560000e+00  4.02310000e+00 -2.17320000e+00  5.19670000e+00]
 [ 0.00000000e+00  1.34394804e+01 -4.01866194e+00  1.08069910e+01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  5.09169403e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01]]
```

7. Modifique el algoritmo de eliminación gaussiana de tal forma que se pueda utilizar para resolver un sistema lineal usando la descomposición LU y, a continuación, resuelva los siguientes sistemas lineales.

a.
$$\begin{aligned}2x_1 - x_2 + x_3 &= -1,\\ 3x_1 + 3x_2 + 9x_3 &= 0,\\ 3x_1 + 3x_2 + 5x_3 &= 4\end{aligned}$$

```python
%autoreload 2
A = [[2,-1,1],[3,3,9],[3,3,5]]
b = [1,0,4]
L,U = descomposicion_LU(A)
print("\nMatriz L:\n",L)
print("\nMatriz U:\n",U)
y = eliminacion_gaussiana_L(L,b)
print("\nValor de y:", y,"\n")
x = eliminacion_gaussiana_U(U,y)
print("\nValor de la solución x:", x)
```

```
Matriz L:
 [[1.  0.  0. ]
 [1.5 1.  0. ]
 [1.5 1.  1. ]]

Matriz U:
 [[ 2.  -1.   1. ]
 [ 0.   4.5  7.5]
 [ 0.   0.  -4. ]]

[[ 1.   0.   0.   1. ]
 [ 0.   1.   0.  -1.5]
 [ 1.5  1.   1.   4. ]]

[[ 1.   0.   0.   1. ]
 [ 0.   1.   0.  -1.5]
 [ 0.   1.   1.   2.5]]

[[ 1.   0.   0.   1. ]
 [ 0.   1.   0.  -1.5]
 [ 0.   0.   1.   4. ]]

Valor de y: [ 1.  -1.5  4. ]

[[ 2.  -1.   1.   1. ]
 [ 0.   4.5  7.5 -1.5]
 [ 0.   0.  -4.   4. ]]

[[ 2.  -1.   1.   1. ]
 [ 0.   4.5  0.   6. ]
 [ 0.   0.  -4.   4. ]]

[[ 2.  -1.   0.   2. ]
 [ 0.   4.5  0.   6. ]
 [ 0.   0.  -4.   4. ]]

[[ 2.          0.          0.          3.33333333]
 [ 0.          4.5         0.          6.        ]
 [ 0.          0.         -4.          4.        ]]
```

```
Valor de la solución x: [ 1.6666666  1.3333334 -1.       ]
```

b.

$$
\begin{aligned}
1.012x_1 - 2.132x_2 + 3.104x_3 &= 1.984,\\
-2.132x_1 + 4.096x_2 + -7.013x_3 &= -5.049,\\
3.104x_1 - 7.013x_2 + 0.014x_3 &= -3.895
\end{aligned}
$$

```python
%autoreload 2
A = [[1.012,-2.132,3.104],[-2.132,4.096,-7.013],[3.104,-7.013,0.014]]
b = [1.984,-5.049,-3.895]
L,U = descomposicion_LU(A)
print("\nMatriz L:\n",L)
print("\nMatriz U:\n",U)
y = eliminacion_gaussiana_L(L,b)
print("\nValor de y:", y,"\n")
x = eliminacion_gaussiana_U(U,y)
print("\nValor de la solución x:", x)
```

```
Matriz L:
 [[ 1.          0.          0.        ]
 [-2.10671937  1.          0.        ]
 [ 3.06719368  1.19775553  1.        ]]

Matriz U:
 [[ 1.012      -2.132       3.104     ]
 [ 0.         -0.39552569 -0.47374308]
 [ 0.          0.         -8.93914077]]

[[ 1.          0.          0.          1.984     ]
 [ 0.          1.          0.         -0.86926877]
 [ 3.06719368  1.19775553  1.         -3.895     ]]

[[ 1.          0.          0.          1.984     ]
 [ 0.          1.          0.         -0.86926877]
 [ 0.          1.19775553  1.         -9.98031225]]

[[ 1.          0.          0.          1.984     ]
 [ 0.          1.          0.         -0.86926877]
 [ 0.          0.          1.         -8.93914077]]

Valor de y: [ 1.984     -0.8692688 -8.93914   ]

[[ 1.012      -2.132       3.104       1.98399997]
 [ 0.         -0.39552569 -0.47374308 -0.86926877]
 [ 0.          0.         -8.93914077 -8.93914032]]

[[ 1.012      -2.132       3.104       1.98399997]
 [ 0.         -0.39552569  0.         -0.39552572]
 [ 0.          0.         -8.93914077 -8.93914032]]

[[ 1.012      -2.132       0.         -1.11999987]
 [ 0.         -0.39552569  0.         -0.39552572]
 [ 0.          0.         -8.93914077 -8.93914032]]

[[ 1.012       0.          0.          1.01200026]
 [ 0.         -0.39552569  0.         -0.39552572]
 [ 0.          0.         -8.93914077 -8.93914032]]

Valor de la solución x: [1.0000002  1.0000001  0.99999994]
```

c.

$$
\begin{aligned}
2x_1 \quad\quad\quad\quad\quad &= 3,\\
x_1 + 1.5x_2 \quad\quad\quad &= 4.5,\\
-3x_2 + 0.5x_3 \quad &= -6.6,\\
2x_1 - 2x_2 + \quad x_3 + x_4 &= 0.8
\end{aligned}
$$

```python
%autoreload 2
A = [[2,0,0,0],[1,1.5,0,0],[0,-3,0.5,0],[2,-2,1,1]]
b = [3,4.5,-6.6,0.8]
L,U = descomposicion_LU(A)
print("\nMatriz L:\n",L)
print("\nMatriz U:\n",U)
y = eliminacion_gaussiana_L(L,b)
print("\nValor de y:", y,"\n")
x = eliminacion_gaussiana_U(U,y)
print("\nValor de la solución x:", x)
```

```
Matriz L:
 [[ 1.          0.          0.          0.        ]
 [ 0.5         1.          0.          0.        ]
 [ 0.         -2.          1.          0.        ]
 [ 1.         -1.33333333  2.          1.        ]]

Matriz U:
 [[2.  0.  0.  0. ]
 [0.  1.5 0.  0. ]
 [0.  0.  0.5 0. ]
 [0.  0.  0.  1. ]]

[[ 1.          0.          0.          0.          3.        ]
 [ 0.          1.          0.          0.          3.        ]
 [ 0.         -2.          1.          0.         -6.6       ]
 [ 1.         -1.33333333  2.          1.          0.8       ]]

[[ 1.          0.          0.          0.          3.        ]
 [ 0.          1.          0.          0.          3.        ]
 [ 0.         -2.          1.          0.         -6.6       ]
 [ 1.         -1.33333333  2.          1.          0.8       ]]

[[ 1.          0.          0.          0.          3.        ]
 [ 0.          1.          0.          0.          3.        ]
 [ 0.         -2.          1.          0.         -6.6       ]
 [ 0.         -1.33333333  2.          1.         -2.2       ]]

[[ 1.          0.          0.          0.          3.        ]
 [ 0.          1.          0.          0.          3.        ]
 [ 0.          0.          1.          0.         -0.6       ]
 [ 0.         -1.33333333  2.          1.         -2.2       ]]

[[ 1.   0.   0.   0.   3. ]
 [ 0.   1.   0.   0.   3. ]
 [ 0.   0.   1.   0.  -0.6]
 [ 0.   0.   2.   1.   1.8]]

[[ 1.   0.   0.   0.   3. ]
 [ 0.   1.   0.   0.   3. ]
 [ 0.   0.   1.   0.  -0.6]
 [ 0.   0.   0.   1.   3. ]]

Valor de y: [ 3.   3.  -0.6  3. ]

[[ 2.          0.          0.          0.          3.        ]
 [ 0.          1.5         0.          0.          3.        ]
 [ 0.          0.          0.5         0.         -0.60000002]
 [ 0.          0.          0.          1.          3.        ]]

[[ 2.          0.          0.          0.          3.        ]
 [ 0.          1.5         0.          0.          3.        ]
 [ 0.          0.          0.5         0.         -0.60000002]
 [ 0.          0.          0.          1.          3.        ]]

[[ 2.          0.          0.          0.          3.        ]
 [ 0.          1.5         0.          0.          3.        ]
 [ 0.          0.          0.5         0.         -0.60000002]
 [ 0.          0.          0.          1.          3.        ]]

[[ 2.          0.          0.          0.          3.        ]
 [ 0.          1.5         0.          0.          3.        ]
 [ 0.          0.          0.5         0.         -0.60000002]
 [ 0.          0.          0.          1.          3.        ]]

[[ 2.          0.          0.          0.          3.        ]
 [ 0.          1.5         0.          0.          3.        ]
 [ 0.          0.          0.5         0.         -0.60000002]
 [ 0.          0.          0.          1.          3.        ]]

[[ 2.          0.          0.          0.          3.        ]
 [ 0.          1.5         0.          0.          3.        ]
 [ 0.          0.          0.5         0.         -0.60000002]
 [ 0.          0.          0.          1.          3.        ]]

[[ 2.          0.          0.          0.          3.        ]
 [ 0.          1.5         0.          0.          3.        ]
 [ 0.          0.          0.5         0.         -0.60000002]
```

```
 [ 0.         0.         0.         1.         3.        ]]

Valor de la solución x: [ 1.5  2.  -1.2  3. ]
```

d.

$$\begin{aligned}2.1756x_1 + 4.0231x_2 - 2.1732x_3 + 5.1967x_4 &= 17.102,\\ -4.0231x_1 + 6.0000x_2 \qquad\quad + 1.1973x_4 &= -6.1593,\\ -1.0000x_1 - 5.2107x_2 + 1.1111x_3 \qquad\quad &= 3.0004,\\ 6.0235x_1 + 7.0000x_2 + \qquad\quad -4.1561x_4 &= 0.0000\end{aligned}$$

```
%autoreload 2
A = [[2.1756,4.0231,-2.1732,5.1967],[-4.0231,6.0000,0,1.1973],[-1,-5.2107,1.1111,0],[6.0235,7.0000,0,-4.
b = [17.102,-6.1593,3.0004,0.0000]
L,U = descomposicion_LU(A)
print("\nMatriz L:\n",L)
print("\nMatriz U:\n",U)
y = eliminacion_gaussiana_L(L,b)
print("\nValor de y:", y,"\n")
x = eliminacion_gaussiana_U(U,y)
print("\nValor de la solución x:", x)
```

```
Matriz L:
 [[ 1.          0.          0.          0.        ]
 [-1.84919103  1.          0.          0.        ]
 [-0.45964332 -0.25012194  1.          0.        ]
 [ 2.76866152 -0.30794361 -5.35228302  1.        ]]

Matriz U:
 [[ 2.17560000e+00  4.02310000e+00 -2.17320000e+00  5.19670000e+00]
 [ 0.00000000e+00  1.34394804e+01 -4.01866194e+00  1.08069910e+01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  5.09169403e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01]]

[[ 1.         0.         0.         0.        17.102     ]
 [ 0.         1.         0.         0.        25.46556496]
 [-0.45964332 -0.25012194  1.         0.         3.0004    ]
 [ 2.76866152 -0.30794361 -5.35228302  1.         0.        ]]

[[ 1.         0.         0.         0.        17.102     ]
 [ 0.         1.         0.         0.        25.46556496]
 [ 0.        -0.25012194  1.         0.        10.86122    ]
 [ 2.76866152 -0.30794361 -5.35228302  1.         0.        ]]

[[  1.           0.           0.           0.          17.102     ]
 [  0.           1.           0.           0.          25.46556496]
 [  0.          -0.25012194   1.           0.          10.86122   ]
 [  0.          -0.30794361  -5.35228302   1.         -47.34964929]]

[[  1.           0.           0.           0.          17.102     ]
 [  0.           1.           0.           0.          25.46556496]
 [  0.           0.           1.           0.          17.23071662]
 [  0.          -0.30794361  -5.35228302   1.         -47.34964929]]

[[  1.           0.           0.           0.          17.102     ]
 [  0.           1.           0.           0.          25.46556496]
 [  0.           0.           1.           0.          17.23071662]
 [  0.           0.          -5.35228302   1.         -39.50769122]]

[[ 1.         0.         0.         0.        17.102     ]
 [ 0.         1.         0.         0.        25.46556496]
 [ 0.         0.         1.         0.        17.23071662]
 [ 0.         0.         0.         1.        52.71598078]]

Valor de y: [17.102     25.465565 17.230717 52.71598 ]

[[ 2.17560000e+00  4.02310000e+00 -2.17320000e+00  5.19670000e+00
   1.71019993e+01]
 [ 0.00000000e+00  1.34394804e+01 -4.01866194e+00  1.08069910e+01
   2.54655647e+01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  5.09169403e+00
   1.72307167e+01]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01
   5.27159805e+01]]

[[ 2.17560000e+00  4.02310000e+00 -2.17320000e+00  5.19670000e+00
   1.71019993e+01]
```

```
 [ 0.00000000e+00  1.34394804e+01 -4.01866194e+00  1.08069910e+01
   2.54655647e+01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  0.00000000e+00
  -5.06994697e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01
   5.27159805e+01]]

[[ 2.17560000e+00  4.02310000e+00 -2.17320000e+00  5.19670000e+00
   1.71019993e+01]
 [ 0.00000000e+00  1.34394804e+01 -4.01866194e+00  0.00000000e+00
  -2.18670265e+01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  0.00000000e+00
  -5.06994697e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01
   5.27159805e+01]]

[[ 2.17560000e+00  4.02310000e+00 -2.17320000e+00  0.00000000e+00
  -5.65857084e+00]
 [ 0.00000000e+00  1.34394804e+01 -4.01866194e+00  0.00000000e+00
  -2.18670265e+01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  0.00000000e+00
  -5.06994697e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01
   5.27159805e+01]]

[[ 2.17560000e+00  4.02310000e+00 -2.17320000e+00  0.00000000e+00
  -5.65857084e+00]
 [ 0.00000000e+00  1.34394804e+01  0.00000000e+00  0.00000000e+00
   9.49870688e-01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  0.00000000e+00
  -5.06994697e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01
   5.27159805e+01]]

[[ 2.17560000e+00  4.02310000e+00  0.00000000e+00  0.00000000e+00
   6.68028266e+00]
 [ 0.00000000e+00  1.34394804e+01  0.00000000e+00  0.00000000e+00
   9.49870688e-01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  0.00000000e+00
  -5.06994697e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01
   5.27159805e+01]]

[[ 2.17560000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   6.39593946e+00]
 [ 0.00000000e+00  1.34394804e+01  0.00000000e+00  0.00000000e+00
   9.49870688e-01]
 [ 0.00000000e+00  4.44089210e-16 -8.92952394e-01  0.00000000e+00
  -5.06994697e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.20361280e+01
   5.27159805e+01]]

Valor de la solución x: [2.9398508  0.07067764 5.677735   4.3798122 ]
```

Link del repositorio:

https://github.com/ElAlfa3007/M-todos_Num-ricos.git