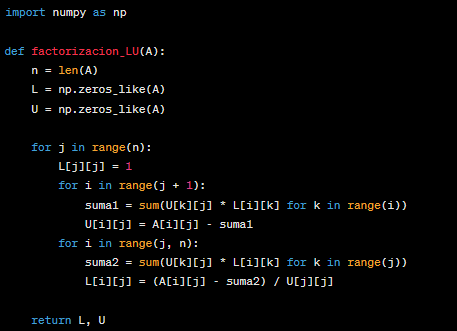
**Ejercicio 1**

Pregunta 1(a): Implementación de la Factorización LU

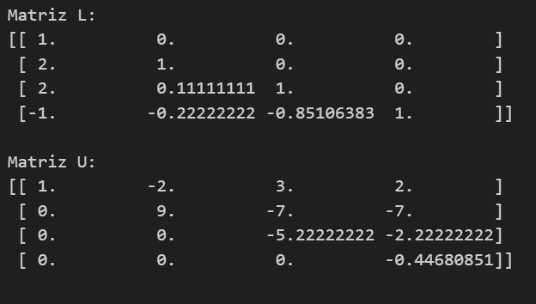
La factorización LU descompone una matriz *A* en dos matrices: una matriz triangular inferior que llamamos *L* y luego una matriz triangular superior llamada *U*. Este método es muy útil y frecuentemente usado para resolver sistemas de ecuaciones lineales. Esto se hace al transformar la matriz *A* en *U* mediante operaciones elementales, mientras que se registra o se guarda, cada paso de estas operaciones en *L*.

A = np.array([[1, -2, 3, 2],

[2, 5, -1, -3],

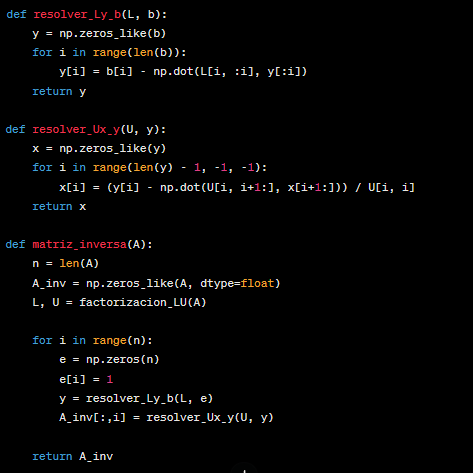
[2, -3, 0, 1],

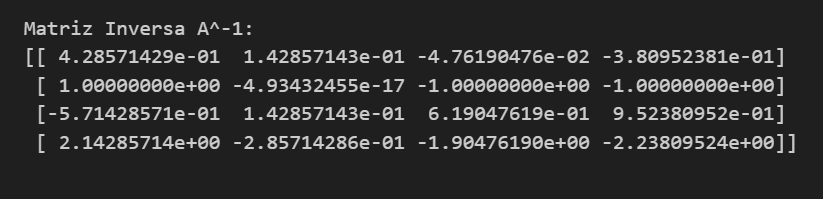
[-1, 0, 3, 1]])

Resultado:

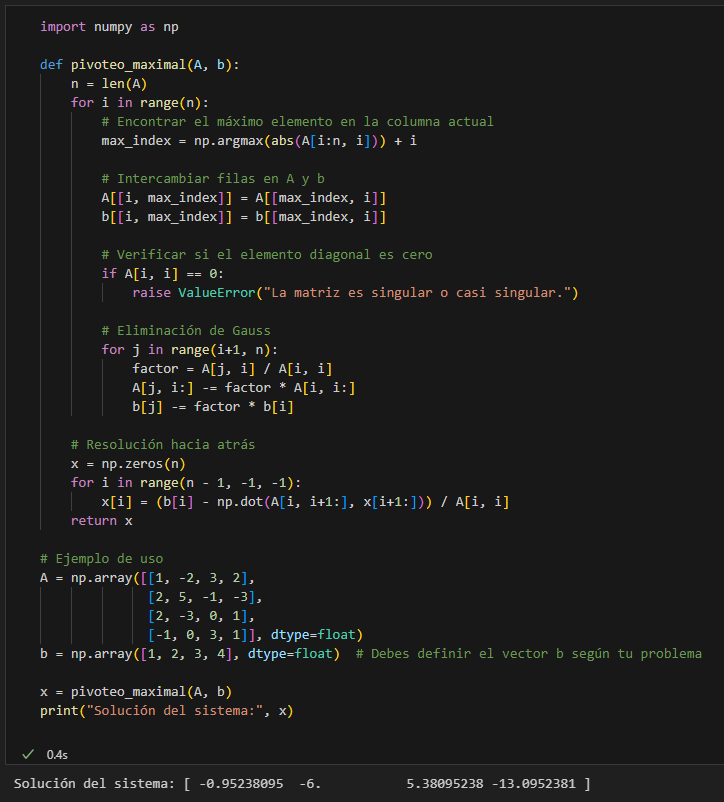
1(b): Implementación de la Rutina para la Matriz Inversa

Para calcular la matriz inversa *A*−1 de matriz no singular *A,* como se nos pide en la segunda parte del ejercicio, lo que haremos será resolver sistemas de ecuaciones lineales *A***c***i*​=**e***i*​ para cada columna **c***i*​ de la matriz inversa, donde **e***i*​ es el vector canónico correspondiente. Utilizando la factorización LU obtenida en el apartado anterior, cada sistema se resuelve de manera más eficiente.



Resultado:

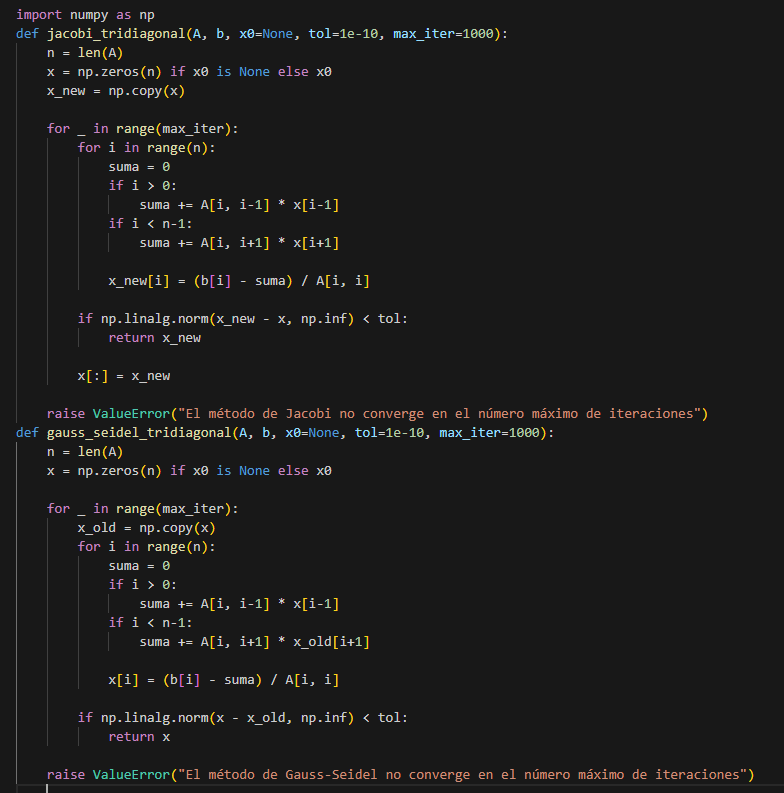
1(c): Implementación del Pivoteo Maximal por Columnas

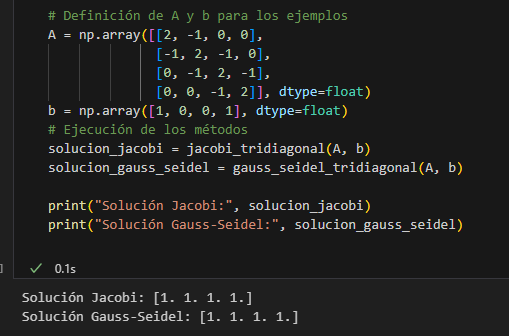
Para mejorar la estabilidad numérica del método de Gauss, usaremos el método de pivoteo maximal por columnas. Este se trata de seleccional el mayor elemento absoluto en cada columna como un pivote mientas se hace la eliminación de Gauss, con esto estaríamos logrando reducir el error de redondeo en los cálculos.

**Ejercicio 2**

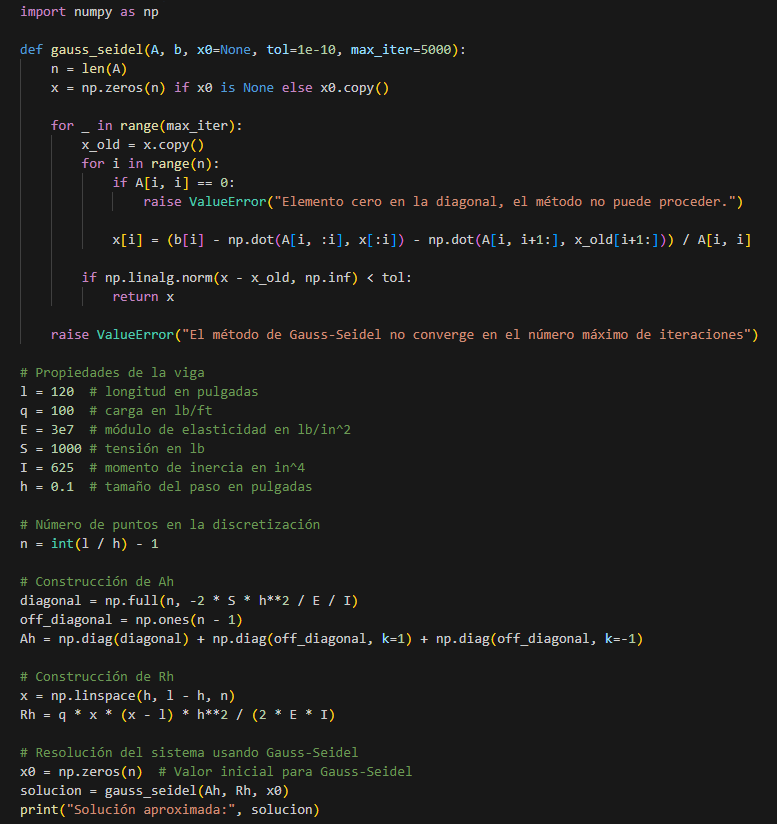
Parte (a): Algoritmos de Jacobi y Gauss-Seidel para Matrices Tridiagonales

Para matrices tridiagonales, podemos adaptar los algoritmos de Jacobi y Gauss-Seidel con esto lograríamos que sean más eficientes, para esto estaremos aprovechando la estructura de la matriz. La mayoría de las operaciones involucra solo tres elementos por fila, lo que reduce la complejidad computacional.





Parte (b): Resolución del Problema de la Viga

Para la resolución de este apartad, trabajaremos utilizando el método de las diferencias finitas, con esto logramos un aproximado, tomaremos las características dadas de la viga, puedes construir la matriz *Ah*​ y el vector *Rh*​, y luego aplicar los algoritmos de Jacobi o Gauss-Seidel para encontrar la solución aproximada.

Parte (c): Comparación con la Solución Exacta

Finalmente podremos generar una comparación de la solución aproximada con la solución exacta dada y con esto debemos calcular el error global máximo. Para esto, evaluamos ambas soluciones en una serie de puntos y grafica la diferencia entre ellas.

