

# [Tarea 11] Ejercicios Unidad 04-D | Gauss-Jacobi y Gauss-Seide

realizado por: Correa Adrian

Fecha: 05/02/2025

**link de GitHub:**

<https://github.com/afca2002/Tarea-11-Gauss-Jacobi-y-Gauss-Seidel.git>

## Ejercicio 1

Encuentre las primeras dos iteraciones del método de Gauss-Jacobi para resolver los siguientes sistemas lineales, por medio de un vector inicial  $\mathbf{x}^{(0)} = \mathbf{0}$

**Sistemas de ecuaciones:**

a.

$$\begin{aligned}3x_1 - x_2 + x_3 &= 1, \\3x_1 + 6x_2 + 2x_3 &= 0, \\3x_1 + 3x_2 + 7x_3 &= 4.\end{aligned}$$

b.

$$\begin{aligned}10x_1 - x_2 &= 9, \\-x_1 + 10x_2 - 2x_3 &= 7, \\-2x_2 + 10x_3 &= 6.\end{aligned}$$

c.

$$\begin{aligned}10x_1 + 5x_2 &= 6, \\5x_1 + 10x_2 - 4x_3 &= 25, \\-4x_2 + 8x_3 - x_4 &= -11, \\-x_3 + 5x_4 &= -11.\end{aligned}$$

d.

$$\begin{aligned}4x_1 + x_2 + x_3 + x_5 &= 6, \\-x_1 - 3x_2 + x_3 + x_4 &= 6, \\2x_1 + x_2 + 5x_3 - x_4 - x_5 &= 6, \\-x_1 - x_2 - x_3 + 4x_4 &= 6, \\2x_2 - x_3 + x_4 + 4x_5 &= 6.\end{aligned}$$

In [1]: `import numpy as np`

```

def gauss_jacobi(A, b, x0, iterations=2):
    """
        Método de Gauss-Jacobi para resolver sistemas de ecuaciones lineales.
        :param A: Matriz de coeficientes (numpy array)
        :param b: Vector de términos independientes (numpy array)
        :param x0: Vector inicial (numpy array)
        :param iterations: Número de iteraciones a realizar (default=2)
        :return: Lista con los vectores de cada iteración
    """
    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for _ in range(iterations):
        x_new = np.zeros_like(x)
        for i in range(n):
            sum_ = sum(A[i][j] * x[j] for j in range(n) if j != i)
            x_new[i] = (b[i] - sum_) / A[i][i]
        x = x_new.copy()
        results.append(x.copy())
    return results

# Definición de los sistemas de ecuaciones
systems = {
    "a": {
        "A": np.array([[3, -1, 1], [3, 6, 2], [3, 3, 7]], dtype=float),
        "b": np.array([1, 0, 4], dtype=float),
    },
    "b": {
        "A": np.array([[10, -1, 0], [-1, 10, -2], [0, -2, 10]], dtype=float),
        "b": np.array([9, 7, 6], dtype=float),
    },
    "c": {
        "A": np.array([[10, 5, 0, 0], [5, 10, -4, 0], [0, -4, 8, -1], [0, 0, -1, 0]], dtype=float),
        "b": np.array([6, 25, -11, -11], dtype=float),
    },
    "d": {
        "A": np.array([
            [4, 1, 1, 0, 1],
            [-1, -3, 1, 1, 0],
            [2, 1, 5, -1, -1],
            [-1, -1, -1, 4, 0],
            [0, 2, -1, 1, 4],
        ], dtype=float),
        "b": np.array([6, 6, 6, 6, 6], dtype=float),
    },
}

# Vector inicial y número de iteraciones
initial_guesses = {
    "a": np.zeros(3),
    "b": np.zeros(3),
    "c": np.zeros(4),
    "d": np.zeros(5),
}

# Calcular las primeras dos iteraciones para cada sistema
if __name__ == "__main__":
    for key, system in systems.items():
        print(f"\nResultados para el sistema {key}:")
        A, b = system["A"], system["b"]

```

```

x0 = initial_guesses[key]
results = gauss_jacobi(A, b, x0)
for i, res in enumerate(results):
    print(f"Iteración {i}: {res}")

```

Resultados para el sistema a:

```

Iteración 0: [0. 0. 0.]
Iteración 1: [0.33333333 0.           0.57142857]
Iteración 2: [ 0.14285714 -0.35714286  0.42857143]

```

Resultados para el sistema b:

```

Iteración 0: [0. 0. 0.]
Iteración 1: [0.9 0.7 0.6]
Iteración 2: [0.97 0.91 0.74]

```

Resultados para el sistema c:

```

Iteración 0: [0. 0. 0. 0.]
Iteración 1: [ 0.6   2.5  -1.375 -2.2  ]
Iteración 2: [-0.65  1.65  -0.4   -2.475]

```

Resultados para el sistema d:

```

Iteración 0: [0. 0. 0. 0. 0.]
Iteración 1: [ 1.5 -2.   1.2  1.5  1.5]
Iteración 2: [ 1.325 -1.6   1.6   1.675 2.425]

```

## Ejercicio 2

2. Repita el ejercicio 1 usando el método de Gauss-Siedel

```

In [2]: import numpy as np

def gauss_seidel(A, b, x0, iterations=2):

    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for _ in range(iterations):
        for i in range(n):
            sum1 = sum(A[i][j] * x[j] for j in range(i))
            sum2 = sum(A[i][j] * x[j] for j in range(i + 1, n))
            x[i] = (b[i] - sum1 - sum2) / A[i][i]
        results.append(x.copy())
    return results

# Definición de los sistemas de ecuaciones
systems = {
    "a": {
        "A": np.array([[3, -1, 1], [3, 6, 2], [3, 3, 7]], dtype=float),
        "b": np.array([1, 0, 4], dtype=float),
    },
    "b": {
        "A": np.array([[10, -1, 0], [-1, 10, -2], [0, -2, 10]]), dtype=float),
        "b": np.array([9, 7, 6], dtype=float),
    },
    "c": {
        "A": np.array([[10, 5, 0, 0], [5, 10, -4, 0], [0, -4, 8, -1], [0, 0, -1, 0]]), dtype=float),
        "b": np.array([6, 25, -11, -11], dtype=float),
    },
    "d": {

```

```

        "A": np.array([
            [4, 1, 1, 0, 1],
            [-1, -3, 1, 1, 0],
            [2, 1, 5, -1, -1],
            [-1, -1, -1, 4, 0],
            [0, 2, -1, 1, 4],
        ], dtype=float),
        "b": np.array([6, 6, 6, 6, 6], dtype=float),
    },
}

# Vector inicial y número de iteraciones
initial_guesses = {
    "a": np.zeros(3),
    "b": np.zeros(3),
    "c": np.zeros(4),
    "d": np.zeros(5),
}

# Calcular las primeras dos iteraciones para cada sistema
if __name__ == "__main__":
    print("\nResultados utilizando el método de Gauss-Seidel:")
    for key, system in systems.items():
        print(f"\nSistema {key}:")
        A, b = system["A"], system["b"]
        x0 = initial_guesses[key]
        results = gauss_seidel(A, b, x0)
        for i, res in enumerate(results):
            print(f"Iteración {i}: {res}")

```

Resultados utilizando el método de Gauss-Seidel:

Sistema a:

```

Iteración 0: [0. 0. 0.]
Iteración 1: [ 0.33333333 -0.16666667  0.5       ]
Iteración 2: [ 0.11111111 -0.22222222  0.61904762]

```

Sistema b:

```

Iteración 0: [0. 0. 0.]
Iteración 1: [0.9  0.79  0.758]
Iteración 2: [0.979 0.9495 0.7899]

```

Sistema c:

```

Iteración 0: [0. 0. 0. 0.]
Iteración 1: [ 0.6   2.2   -0.275 -2.255]
Iteración 2: [-0.5     2.64   -0.336875 -2.267375]

```

Sistema d:

```

Iteración 0: [0. 0. 0. 0. 0.]
Iteración 1: [ 1.5   -2.5     1.1     1.525   2.64375]
Iteración 2: [ 1.1890625 -1.52135417  1.86239583  1.88252604  2.25564453]

```

## Ejercicio 3

3. Utilice el método de Jacobi para resolver los sistemas lineales en el ejercicio 1, con  $TOL = 10^{-3}$

```
In [4]: import numpy as np

def gauss_jacobi(A, b, x0, tol=1e-3, max_iterations=100):

    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for k in range(max_iterations):
        x_new = np.zeros_like(x)
        for i in range(n):
            sum_ = sum(A[i][j] * x[j] for j in range(n) if j != i)
            x_new[i] = (b[i] - sum_) / A[i][i]
        results.append(x_new.copy())
        # Verificar convergencia
        if np.linalg.norm(x_new - x, ord=np.inf) < tol:
            print(f"Convergencia alcanzada en {k+1} iteraciones.")
            break
        x = x_new
    return results, k+1

# Definición de los sistemas de ecuaciones
systems = {
    "a": {
        "A": np.array([[3, -1, 1], [3, 6, 2], [3, 3, 7]], dtype=float),
        "b": np.array([1, 0, 4], dtype=float),
    },
    "b": {
        "A": np.array([[10, -1, 0], [-1, 10, -2], [0, -2, 10]], dtype=float),
        "b": np.array([9, 7, 6], dtype=float),
    },
    "c": {
        "A": np.array([[10, 5, 0, 0], [5, 10, -4, 0], [0, -4, 8, -1], [0, 0, -1, 0]], dtype=float),
        "b": np.array([6, 25, -11, -11], dtype=float),
    },
    "d": {
        "A": np.array([
            [4, 1, 1, 0, 1],
            [-1, -3, 1, 1, 0],
            [2, 1, 5, -1, -1],
            [-1, -1, -1, 4, 0],
            [0, 2, -1, 1, 4],
        ], dtype=float),
        "b": np.array([6, 6, 6, 6, 6], dtype=float),
    },
}

# Vector inicial y tolerancia
initial_guesses = {
    "a": np.zeros(3),
    "b": np.zeros(3),
    "c": np.zeros(4),
    "d": np.zeros(5),
}
tol = 1e-3

# Resolver los sistemas con el método de Gauss-Jacobi
if __name__ == "__main__":
    print("\nResultados utilizando el método de Gauss-Jacobi:")
    for key, system in systems.items():
        results, iterations = gauss_jacobi(system["A"], system["b"], initial_guesses[key])
        print(f"\nSistema {key}:")
        print(f"Matriz A: {system['A']}")
        print(f"Vector b: {system['b']}")
        print(f"Vector resultado: {results[-1]}")
        print(f"Número de iteraciones: {iterations}
```

```
print(f"\nSistema {key}:")
A, b = system["A"], system["b"]
x0 = initial_guesses[key]
results, iterations = gauss_jacobi(A, b, x0, tol)
for i, res in enumerate(results):
    print(f"Iteración {i}: {res}")
```

Resultados utilizando el método de Gauss-Jacobi:

Sistema a:

Convergencia alcanzada en 9 iteraciones.

Iteración 0: [ 0. 0. 0.]  
Iteración 1: [ 0.33333333 0. 0.57142857]  
Iteración 2: [ 0.14285714 -0.35714286 0.42857143]  
Iteración 3: [ 0.07142857 -0.21428571 0.66326531]  
Iteración 4: [ 0.04081633 -0.25680272 0.63265306]  
Iteración 5: [ 0.03684807 -0.23129252 0.66399417]  
Iteración 6: [ 0.03490444 -0.23975543 0.6547619 ]  
Iteración 7: [ 0.03516089 -0.23570619 0.65922185]  
Iteración 8: [ 0.03502399 -0.23732106 0.65737656]  
Iteración 9: [ 0.03510079 -0.23663751 0.65812732]

Sistema b:

Convergencia alcanzada en 6 iteraciones.

Iteración 0: [ 0. 0. 0.]  
Iteración 1: [ 0.9 0.7 0.6]  
Iteración 2: [ 0.97 0.91 0.74]  
Iteración 3: [ 0.991 0.945 0.782]  
Iteración 4: [ 0.9945 0.9555 0.789 ]  
Iteración 5: [ 0.99555 0.95725 0.7911 ]  
Iteración 6: [ 0.995725 0.957775 0.79145 ]

Sistema c:

Convergencia alcanzada en 21 iteraciones.

Iteración 0: [ 0. 0. 0. 0.]  
Iteración 1: [ 0.6 2.5 -1.375 -2.2 ]  
Iteración 2: [-0.65 1.65 -0.4 -2.475]  
Iteración 3: [-0.225 2.665 -0.859375 -2.28 ]  
Iteración 4: [-0.7325 2.26875 -0.3275 -2.371875]  
Iteración 5: [-0.534375 2.73525 -0.53710938 -2.2655 ]  
Iteración 6: [-0.767625 2.55234375 -0.2905625 -2.30742188]  
Iteración 7: [-0.67617187 2.7675875 -0.38725586 -2.2581125 ]  
Iteración 8: [-0.78379375 2.68318359 -0.27347031 -2.27745117]  
Iteración 9: [-0.7415918 2.78250875 -0.3180896 -2.25469406]  
Iteración 10: [-0.79125437 2.74356006 -0.26558238 -2.26361792]  
Iteración 11: [-0.77178003 2.78939423 -0.28617221 -2.25311648]  
Iteración 12: [-0.79469712 2.77142113 -0.26194244 -2.25723444]  
Iteración 13: [-0.78571057 2.79257158 -0.27144374 -2.25238849]  
Iteración 14: [-0.79628579 2.78427779 -0.26026277 -2.25428875]  
Iteración 15: [-0.79213889 2.79403779 -0.2646472 -2.25205255]  
Iteración 16: [-0.79701889 2.79021057 -0.25948768 -2.25292944]  
Iteración 17: [-0.79510528 2.79471438 -0.2615109 -2.25189754]  
Iteración 18: [-0.79735719 2.79294828 -0.25913 -2.25230218]  
Iteración 19: [-0.79647414 2.79502659 -0.26006363 -2.251826 ]  
Iteración 20: [-0.7975133 2.79421162 -0.25896495 -2.25201273]  
Iteración 21: [-0.79710581 2.79517067 -0.25939578 -2.25179299]

Sistema d:

Convergencia alcanzada en 12 iteraciones.

Iteración 0: [ 0. 0. 0. 0. 0.]  
Iteración 1: [ 1.5 -2. 1.2 1.5 1.5]  
Iteración 2: [ 1.325 -1.6 1.6 1.675 2.425]  
Iteración 3: [ 0.89375 -1.35 1.81 1.83125 2.28125]  
Iteración 4: [ 0.8146875 -1.08416667 1.935 1.8384375 2.1696875 ]  
Iteración 5: [ 0.74486979 -1.01375 1.89258333 1.91638021 2.06622396]  
Iteración 6: [ 0.76373568 -0.97863542 1.90132292 1.90592578 2.00092578]  
Iteración 7: [ 0.76909668 -0.98549566 1.87160313 1.92160579 1.98816699]

```

Iteración 8: [ 0.78143139 -0.99196259  1.87141502  1.91380104  1.98024716]
Iteración 9: [ 0.7850751 -0.99873844  1.8646296   1.91522095  1.98538479]
Iteración 10: [ 0.78718101 -1.00174151   1.8658388   1.91274157   1.98672138]
Iteración 11: [ 0.78729533 -1.00286688   1.86536849   1.91281957   1.98914507]
Iteración 12: [ 0.78708833 -1.00303576   1.86604817   1.91244923   1.98957067]

```

## Ejercicio 4

4. Utilice el método de Gauss-Seidel para resolver los sistemas lineales en el ejercicio 1, con  $TOL = 10^{-3}$

```

In [5]: import numpy as np

def gauss_seidel(A, b, x0, tol=1e-3, max_iterations=100):
    """
    Método de Gauss-Seidel para resolver sistemas de ecuaciones lineales.
    :param A: Matriz de coeficientes (numpy array)
    :param b: Vector de términos independientes (numpy array)
    :param x0: Vector inicial (numpy array)
    :param tol: Tolerancia para la convergencia (default=1e-3)
    :param max_iterations: Número máximo de iteraciones (default=100)
    :return: Lista con los vectores de cada iteración y número de iteraciones re
    """
    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for k in range(max_iterations):
        x_new = x.copy()
        for i in range(n):
            sum1 = sum(A[i][j] * x_new[j] for j in range(i))
            sum2 = sum(A[i][j] * x[j] for j in range(i + 1, n))
            x_new[i] = (b[i] - sum1 - sum2) / A[i][i]
        results.append(x_new.copy())
        # Verificar convergencia
        if np.linalg.norm(x_new - x, ord=np.inf) < tol:
            print(f"Convergencia alcanzada en {k+1} iteraciones.")
            break
        x = x_new
    return results, k+1

# Definición de los sistemas de ecuaciones
systems = {
    "a": {
        "A": np.array([[3, -1, 1], [3, 6, 2], [3, 3, 7]], dtype=float),
        "b": np.array([1, 0, 4], dtype=float),
    },
    "b": {
        "A": np.array([[10, -1, 0], [-1, 10, -2], [0, -2, 10]]], dtype=float),
        "b": np.array([9, 7, 6], dtype=float),
    },
    "c": {
        "A": np.array([[10, 5, 0, 0], [5, 10, -4, 0], [0, -4, 8, -1], [0, 0, -1, 0]]], dtype=float),
        "b": np.array([6, 25, -11, -11], dtype=float),
    },
    "d": {
        "A": np.array([
            [4, 1, 1, 0, 1],
            [-1, -3, 1, 1, 0],
        ], dtype=float),
        "b": np.array([1, 2, 3, 4, 5], dtype=float),
    }
}

```

```

        [2, 1, 5, -1, -1],
        [-1, -1, -1, 4, 0],
        [0, 2, -1, 1, 4],
    ], dtype=float),
    "b": np.array([6, 6, 6, 6, 6], dtype=float),
},
}

# Vector inicial y tolerancia
initial_guesses = {
    "a": np.zeros(3),
    "b": np.zeros(3),
    "c": np.zeros(4),
    "d": np.zeros(5),
}
tol = 1e-3

# Resolver los sistemas con el método de Gauss-Seidel
if __name__ == "__main__":
    print("\nResultados utilizando el método de Gauss-Seidel:")
    for key, system in systems.items():
        print(f"\nSistema {key}:")
        A, b = system["A"], system["b"]
        x0 = initial_guesses[key]
        results, iterations = gauss_seidel(A, b, x0, tol)
        for i, res in enumerate(results):
            print(f"Iteración {i}: {res}")

```

Resultados utilizando el método de Gauss-Seidel:

Sistema a:

Convergencia alcanzada en 6 iteraciones.

Iteración 0: [ 0. 0. 0.]  
Iteración 1: [ 0.33333333 -0.16666667 0.5 ]  
Iteración 2: [ 0.11111111 -0.22222222 0.61904762]  
Iteración 3: [ 0.05291005 -0.23280423 0.64852608]  
Iteración 4: [ 0.03955656 -0.23595364 0.65559875]  
Iteración 5: [ 0.0361492 -0.23660752 0.65733928]  
Iteración 6: [ 0.03535107 -0.23678863 0.65775895]

Sistema b:

Convergencia alcanzada en 4 iteraciones.

Iteración 0: [ 0. 0. 0.]  
Iteración 1: [ 0.9 0.79 0.758]  
Iteración 2: [ 0.979 0.9495 0.7899]  
Iteración 3: [ 0.99495 0.957475 0.791495]  
Iteración 4: [ 0.9957475 0.95787375 0.79157475]

Sistema c:

Convergencia alcanzada en 9 iteraciones.

Iteración 0: [ 0. 0. 0. 0.]  
Iteración 1: [ 0.6 2.2 -0.275 -2.255]  
Iteración 2: [-0.5 2.64 -0.336875 -2.267375]  
Iteración 3: [-0.72 2.72525 -0.29579688 -2.25915938]  
Iteración 4: [-0.762625 2.76299375 -0.27589805 -2.25517961]  
Iteración 5: [-0.78149687 2.78038922 -0.26670284 -2.25334057]  
Iteración 6: [-0.79019461 2.78841617 -0.26245949 -2.2524919 ]  
Iteración 7: [-0.79420808 2.79212025 -0.26050136 -2.25210027]  
Iteración 8: [-0.79606012 2.79382952 -0.25959778 -2.25191956]  
Iteración 9: [-0.79691476 2.79461827 -0.25918081 -2.25183616]

Sistema d:

Convergencia alcanzada en 7 iteraciones.

Iteración 0: [ 0. 0. 0. 0. 0.]  
Iteración 1: [ 1.5 -2.5 1.1 1.525 2.64375]  
Iteración 2: [ 1.1890625 -1.52135417 1.86239583 1.88252604 2.25564453]  
Iteración 3: [ 0.85082845 -1.03530219 1.89436317 1.92747236 2.0093738 ]  
Iteración 4: [ 0.7828913 -0.98701859 1.87161643 1.91687229 1.98219533]  
Iteración 5: [ 0.78330171 -0.998271 1.86614704 1.91279444 1.98747365]  
Iteración 6: [ 0.78616258 -1.00240703 1.86606999 1.91245638 1.98960692]  
Iteración 7: [ 0.78668253 -1.00271872 1.86628339 1.9125618 1.98978976]

## Ejercicio 5

5. El sistema lineal:

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

tiene la solución ((1, 2, -1)):

- a) Muestre que el método de Jacobi con  $\mathbf{x}^{(0)} = 0$  falla al proporcionar una buena aproximación después de 25 iteraciones.

b) Utilice el método de Gauss-Seidel con  $\mathbf{x}^{(0)} = \mathbf{0}$  para aproximar la solución para el sistema lineal dentro de  $10^{-5}$

```
In [6]: import numpy as np

def gauss_jacobi(A, b, x0, tol=1e-5, max_iterations=25):

    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for k in range(max_iterations):
        x_new = np.zeros_like(x)
        for i in range(n):
            sum_ = sum(A[i][j] * x[j] for j in range(n) if j != i)
            x_new[i] = (b[i] - sum_) / A[i][i]
        results.append(x_new.copy())
        if np.linalg.norm(x_new - x, ord=np.inf) < tol:
            break
        x = x_new
    return results, k+1

def gauss_seidel(A, b, x0, tol=1e-5, max_iterations=100):

    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for k in range(max_iterations):
        x_new = x.copy()
        for i in range(n):
            sum1 = sum(A[i][j] * x_new[j] for j in range(i))
            sum2 = sum(A[i][j] * x[j] for j in range(i + 1, n))
            x_new[i] = (b[i] - sum1 - sum2) / A[i][i]
        results.append(x_new.copy())
        if np.linalg.norm(x_new - x, ord=np.inf) < tol:
            break
        x = x_new
    return results, k+1

# Definir el sistema de ecuaciones
A = np.array([
    [2, -1, 1],
    [2, 2, 2],
    [-1, -1, 2]
], dtype=float)
b = np.array([-1, 4, -5], dtype=float)

# Vector inicial
x0 = np.zeros(3)

# Parte (a): Método de Gauss-Jacobi
print("\nMétodo de Gauss-Jacobi:")
results_jacobi, iterations_jacobi = gauss_jacobi(A, b, x0, tol=1e-5, max_iterations=25)
for i, res in enumerate(results_jacobi):
    print(f"Iteración {i}: {res}")

# Parte (b): Método de Gauss-Seidel
print("\nMétodo de Gauss-Seidel:")
results_seidel, iterations_seidel = gauss_seidel(A, b, x0, tol=1e-5)
for i, res in enumerate(results_seidel):
```

```
    print(f"Iteración {i}: {res}")
print(f"Convergencia alcanzada en {iterations_seidel} iteraciones con Gauss-Seid
```

Método de Gauss-Jacobi:

```
Iteración 0: [0. 0. 0.]
Iteración 1: [-0.5 2. -2.5]
Iteración 2: [ 1.75 5. -1.75]
Iteración 3: [2.875 2. 0.875]
Iteración 4: [ 0.0625 -1.75 -0.0625]
Iteración 5: [-1.34375 2. -3.34375]
Iteración 6: [ 2.171875 6.6875 -2.171875]
Iteración 7: [3.9296875 2. 1.9296875]
Iteración 8: [-0.46484375 -3.859375 0.46484375]
Iteración 9: [-2.66210938 2. -4.66210938]
Iteración 10: [ 2.83105469 9.32421875 -2.83105469]
Iteración 11: [5.57763672 2. 3.57763672]
Iteración 12: [-1.28881836 -7.15527344 1.28881836]
Iteración 13: [-4.7220459 2. -6.7220459]
Iteración 14: [ 3.86102295 13.4440918 -3.86102295]
Iteración 15: [8.15255737 2. 6.15255737]
Iteración 16: [-2.57627869 -12.30511475 2.57627869]
Iteración 17: [-7.94069672 2. -9.94069672]
Iteración 18: [ 5.47034836 19.88139343 -5.47034836]
Iteración 19: [12.1758709 2. 10.1758709]
Iteración 20: [-4.58793545 -20.35174179 4.58793545]
Iteración 21: [-12.96983862 2. -14.96983862]
Iteración 22: [ 7.98491931 29.93967724 -7.98491931]
Iteración 23: [18.46229827 2. 16.46229827]
Iteración 24: [-7.73114914 -32.92459655 7.73114914]
Iteración 25: [-20.82787284 2. -22.82787284]
```

Método de Gauss-Seidel:

```
Iteración 0: [0. 0. 0.]
Iteración 1: [-0.5 2.5 -1.5]
Iteración 2: [ 1.5 2. -0.75]
Iteración 3: [ 0.875 1.875 -1.125]
Iteración 4: [ 1. 2.125 -0.9375]
Iteración 5: [ 1.03125 1.90625 -1.03125]
Iteración 6: [ 0.96875 2.0625 -0.984375]
Iteración 7: [ 1.0234375 1.9609375 -1.0078125]
Iteración 8: [ 0.984375 2.0234375 -0.99609375]
Iteración 9: [ 1.00976562 1.98632812 -1.00195312]
Iteración 10: [ 0.99414062 2.0078125 -0.99902344]
Iteración 11: [ 1.00341797 1.99560547 -1.00048828]
Iteración 12: [ 0.99804688 2.00244141 -0.99975586]
Iteración 13: [ 1.00109863 1.99865723 -1.00012207]
Iteración 14: [ 0.99938965 2.00073242 -0.99993896]
Iteración 15: [ 1.00033569 1.99960327 -1.00003052]
Iteración 16: [ 0.99981689 2.00021362 -0.99998474]
Iteración 17: [ 1.00009918 1.99988556 -1.00000763]
Iteración 18: [ 0.99994659 2.00006104 -0.99999619]
Iteración 19: [ 1.00002861 1.99996758 -1.00000191]
Iteración 20: [ 0.99998474 2.00001717 -0.99999905]
Iteración 21: [ 1.00000811 1.99999094 -1.00000048]
Iteración 22: [ 0.99999571 2.00000477 -0.99999976]
Iteración 23: [ 1.00000226 1.9999975 -1.00000012]
```

Convergencia alcanzada en 23 iteraciones con Gauss-Seidel.

## Ejercicio 6

6. El sistema lineal:

[

$$\begin{aligned}x_1 - x_3 &= 0.2, \\-\frac{1}{2}x_1 + x_2 - \frac{1}{4}x_3 &= -1.425, \\x_1 - \frac{1}{2}x_2 + x_3 &= 2,\end{aligned}$$

]

tiene la solución ((0.9, -0.8, 0.7)):

a) ¿La matriz de coeficientes

$$A = \begin{bmatrix} 1 & 0 & -1 \\ -\frac{1}{2} & 1 & -\frac{1}{4} \\ 1 & -\frac{1}{2} & 1 \end{bmatrix} \text{ tiene diagonal estrictamente dominante?}$$

b) Utilice el método iterativo de Gauss-Seidel para aproximar la solución para el sistema lineal con una tolerancia de  $10^{-2}$  y un máximo de 300 iteraciones.

c) ¿Qué pasa en la parte (b) cuando el sistema cambia por el siguiente?

[

$$\begin{aligned}x_1 - 2x_3 &= 0.2, \\-\frac{1}{2}x_1 + x_2 - \frac{1}{4}x_3 &= -1.425, \\x_1 - \frac{1}{2}x_2 + x_3 &= 2.\end{aligned}$$

]

```
In [7]: # Parte (a): Verificar si la matriz es diagonal estrictamente dominante
import numpy as np

A1 = np.array([
    [1, 0, -1],
    [-0.5, 1, -0.25],
    [1, -0.5, 1]
], dtype=float)

def es_diagonal_dominante(A):
    for i in range(A.shape[0]):
        suma_fila = sum(abs(A[i, j])) for j in range(A.shape[1]) if i != j
        if abs(A[i, i]) <= suma_fila:
            return False
    return True

print("¿La matriz A1 es diagonal estrictamente dominante?", es_diagonal_dominante)
```

¿La matriz A1 es diagonal estrictamente dominante? False

In [9]: # b) Utilice el método iterativo de Gauss-Seidel para aproximar La solución para

```
def gauss_seidel(A, b, x0, tol=1e-2, max_iterations=300):

    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for k in range(max_iterations):
        x_new = x.copy()
        for i in range(n):
            sum1 = sum(A[i][j] * x_new[j] for j in range(i))
            sum2 = sum(A[i][j] * x[j] for j in range(i + 1, n))
            x_new[i] = (b[i] - sum1 - sum2) / A[i][i]
        results.append(x_new.copy())
        if np.linalg.norm(x_new - x, ord=np.inf) < tol:
            print(f"Convergencia alcanzada en {k+1} iteraciones.")
            break
        x = x_new
    return results, k+1

b1 = np.array([0.2, -1.425, 2], dtype=float)
# Vector inicial
x0 = np.zeros(3)

# Resolver el sistema inicial
results1, iterations1 = gauss_seidel(A1, b1, x0)
for i, res in enumerate(results1):
    print(f"Iteración {i}: {res}")
```

Convergencia alcanzada en 13 iteraciones.

```
Iteración 0: [0. 0. 0.]
Iteración 1: [ 0.2   -1.325   1.1375]
Iteración 2: [ 1.3375   -0.471875   0.4265625]
Iteración 3: [ 0.6265625  -1.00507813   0.87089844]
Iteración 4: [ 1.07089844  -0.67182617   0.59318848]
Iteración 5: [ 0.79318848  -0.88010864   0.7667572 ]
Iteración 6: [ 0.9667572   -0.7499321   0.65827675]
Iteración 7: [ 0.85827675  -0.83129244   0.72607703]
Iteración 8: [ 0.92607703  -0.78044223   0.68370185]
Iteración 9: [ 0.88370185  -0.81222361   0.71018634]
Iteración 10: [ 0.91018634  -0.79236024   0.69363354]
Iteración 11: [ 0.89363354  -0.80477485   0.70397904]
Iteración 12: [ 0.90397904  -0.79701572   0.6975131 ]
Iteración 13: [ 0.8975131   -0.80186517   0.70155431]
```

c) ¿Qué pasa en la parte (b) cuando el sistema cambia por el siguiente?

[

$$\begin{aligned}x_1 - 2x_3 &= 0.2, \\-\frac{1}{2}x_1 + x_2 - \frac{1}{4}x_3 &= -1.425, \\x_1 - \frac{1}{2}x_2 + x_3 &= 2.\end{aligned}$$

]

```
In [10]: #c) ¿Qué pasa en la parte (b) cuando el sistema cambia por el siguiente?  
# Sistema modificado  
A2 = np.array([  
    [1, 0, -2],  
    [-0.5, 1, -0.25],  
    [1, -0.5, 1]  
], dtype=float)  
  
# Resolver el sistema modificado  
results2, iterations2 = gauss_seidel(A2, b1, x0)  
for i, res in enumerate(results2):  

```

Iteración 0: [ 0. 0. 0.]  
Iteración 1: [ 0.2 -1.325 1.1375]  
Iteración 2: [ 2.475 0.096875 -0.4265625]  
Iteración 3: [-0.653125 -1.85820313 1.72402344]  
Iteración 4: [ 3.64804688 0.8300293 -1.23303223]  
Iteración 5: [-2.26606445 -2.86629028 2.83291931]  
Iteración 6: [ 5.86583862 2.21614914 -2.75776405]  
Iteración 7: [-5.31552811 -4.77220507 4.92942557]  
Iteración 8: [10.05885115 4.83678197 -5.64046016]  
Iteración 9: [-11.08092033 -8.3755752 8.89313272]  
Iteración 10: [ 17.98626545 9.79141591 -11.0905575 ]  
Iteración 11: [-21.98111499 -15.18819687 16.38701656]  
Iteración 12: [ 32.97403311 19.1587707 -21.39464777]  
Iteración 13: [-42.58929553 -28.06830971 30.55514068]  
Iteración 14: [ 61.31028136 36.86892585 -40.87581843]  
Iteración 15: [-81.55163687 -52.41977304 57.34175035]  
Iteración 16: [114.88350069 70.35218793 -77.70740673]  
Iteración 17: [-155.21481345 -98.45925841 107.98518425]  
Iteración 18: [ 216.1703685 133.65648031 -147.34212834]  
Iteración 19: [-294.48425668 -185.50266043 203.73292647]  
Iteración 20: [ 407.66585294 253.34115809 -278.9952739 ]  
Iteración 21: [-557.79054779 -350.06909237 384.75600161]  
Iteración 22: [ 769.71200321 479.62000201 -527.90200221]  
Iteración 23: [-1055.60400442 -661.20250276 727.00275304]  
Iteración 24: [1454.20550607 907.4284413 -998.49128543]  
Iteración 25: [-1996.78257085 -1249.43910678 1374.06301746]  
Iteración 26: [ 2748.32603492 1716.25377182 -1888.19914901]  
Iteración 27: [-3776.19829801 -2361.57393626 2597.41132988]  
Iteración 28: [ 5195.02265977 3245.43916236 -3570.30307859]  
Iteración 29: [-7140.40615718 -4464.20384824 4910.30423306]  
Iteración 30: [ 9820.80846613 6136.55529133 -6750.53082046]  
Iteración 31: [-13500.86164092 -8439.48852558 9283.11737813]  
Iteración 32: [ 18566.43475627 11602.57172267 -12763.14889494]  
Iteración 33: [-25526.09778987 -15955.26111867 17550.46723054]  
Iteración 34: [ 35101.13446107 21936.75903817 -24130.75494199]  
Iteración 35: [-48261.30988397 -30164.76867748 33180.92554523]  
Iteración 36: [ 66362.05109046 41474.83193154 -45622.63512469]  
Iteración 37: [-91245.07024939 -57029.61890587 62732.26079645]  
Iteración 38: [125464.72159291 78414.00099557 -86255.72109512]  
Iteración 39: [-172511.24219025 -107820.97636891 118602.7540058 ]  
Iteración 40: [ 237205.70801159 148252.11750725 -163077.64925797]  
Iteración 41: [-326155.09851594 -203848.38657246 224232.90522971]  
Iteración 42: [ 448466.01045942 280289.80653714 -308319.10719085]  
Iteración 43: [-616638.0143817 -385400.20898856 423939.90988742]  
Iteración 44: [ 847880.01977484 529923.56235927 -582916.2385952 ]  
Iteración 45: [-1165832.2771904 -728646.623244 801510.9655684]  
Iteración 46: [ 1603022.1311368 1001887.3819605 -1102076.44015655]  
Iteración 47: [-2204152.6803131 -1377596.87519569 1515356.24271526]  
Iteración 48: [ 3030712.68543051 1894193.97839407 -2083613.69623348]  
Iteración 49: [-4167227.19246695 -2604518.44529185 2864969.96982103]  
Iteración 50: [ 5729940.13964206 3581211.13727629 -3939332.57100392]  
Iteración 51: [-7878664.94200783 -4924167.0387549 5416583.42263039]  
Iteración 52: [10833167.04526077 6770727.95328798 -7447801.06861678]  
Iteración 53: [-14895601.93723356 -9309752.66077098 10240727.60684807]  
Iteración 54: [ 20481455.41369614 12800908.18356009 -14080999.3219161 ]  
Iteración 55: [-28161998.4438322 -17601250.47739512 19361375.20513464]  
Iteración 56: [ 38722750.61026928 24201717.6814183 -26621889.76956013]  
Iteración 57: [-53243779.33912025 -33277363.53695016 36605099.57064518]  
Iteración 58: [ 73210199.34129035 45756373.13830648 -50332010.77213712]  
Iteración 59: [-1.00664021e+08 -6.29150148e+07 6.92065159e+07]

Iteración 60: [ 1.38413032e+08 8.65081436e+07 -9.51589583e+07]  
Iteración 61: [-1.90317916e+08 -1.18948699e+08 1.30843569e+08]  
Iteración 62: [ 2.61687138e+08 1.63554460e+08 -1.79909906e+08]  
Iteración 63: [-3.59819812e+08 -2.24887384e+08 2.47376122e+08]  
Iteración 64: [ 4.94752244e+08 3.09220151e+08 -3.40142166e+08]  
Iteración 65: [-6.80284333e+08 -4.25177709e+08 4.67695480e+08]  
Iteración 66: [ 9.35390960e+08 5.84619349e+08 -6.43081284e+08]  
Iteración 67: [-1.28616257e+09 -8.03851606e+08 8.84236766e+08]  
Iteración 68: [ 1.76847353e+09 1.10529596e+09 -1.21582555e+09]  
Iteración 69: [-2.43165110e+09 -1.51978194e+09 1.67176014e+09]  
Iteración 70: [ 3.34352027e+09 2.08970017e+09 -2.29867019e+09]  
Iteración 71: [-4.59734037e+09 -2.87333773e+09 3.16067151e+09]  
Iteración 72: [ 6.32134301e+09 3.95083938e+09 -4.34592332e+09]  
Iteración 73: [-8.69184664e+09 -5.43240415e+09 5.97564457e+09]  
Iteración 74: [ 1.19512891e+10 7.46955571e+09 -8.21651128e+09]  
Iteración 75: [-1.64330226e+10 -1.02706391e+10 1.12977030e+10]  
Iteración 76: [ 2.25954060e+10 1.41221288e+10 -1.55343416e+10]  
Iteración 77: [-3.10686833e+10 -1.94179270e+10 2.13597197e+10]  
Iteración 78: [ 4.27194395e+10 2.66996497e+10 -2.93696147e+10]  
Iteración 79: [-5.87392293e+10 -3.67120183e+10 4.03832201e+10]  
Iteración 80: [ 8.07664403e+10 5.04790252e+10 -5.55269277e+10]  
Iteración 81: [-1.11053855e+11 -6.94086596e+10 7.63495256e+10]  
Iteración 82: [ 1.52699051e+11 9.54369070e+10 -1.04980598e+11]  
Iteración 83: [-2.09961195e+11 -1.31225747e+11 1.44348322e+11]  
Iteración 84: [ 2.88696644e+11 1.80435402e+11 -1.98478942e+11]  
Iteración 85: [-3.96957885e+11 -2.48098678e+11 2.72908546e+11]  
Iteración 86: [ 5.45817092e+11 3.41135682e+11 -3.75249251e+11]  
Iteración 87: [-7.50498501e+11 -4.69061563e+11 5.15967720e+11]  
Iteración 88: [ 1.03193544e+12 6.44959650e+11 -7.09455615e+11]  
Iteración 89: [-1.41891123e+12 -8.86819518e+11 9.75501470e+11]  
Iteración 90: [ 1.95100294e+12 1.21937684e+12 -1.34131452e+12]  
Iteración 91: [-2.68262904e+12 -1.67664315e+12 1.84430747e+12]  
Iteración 92: [ 3.68861493e+12 2.30538433e+12 -2.53592277e+12]  
Iteración 93: [-5.07184553e+12 -3.16990346e+12 3.48689380e+12]  
Iteración 94: [ 6.97378761e+12 4.35861726e+12 -4.79447898e+12]  
Iteración 95: [-9.58895796e+12 -5.99309873e+12 6.59240860e+12]  
Iteración 96: [ 1.31848172e+13 8.24051075e+12 -9.06456182e+12]  
Iteración 97: [-1.81291236e+13 -1.13307023e+13 1.24637725e+13]  
Iteración 98: [ 2.49275450e+13 1.55797156e+13 -1.71376872e+13]  
Iteración 99: [-3.42753744e+13 -2.14221090e+13 2.35643199e+13]  
Iteración 100: [ 4.71286398e+13 2.94553999e+13 -3.24009399e+13]  
Iteración 101: [-6.48018797e+13 -4.05011748e+13 4.45512923e+13]  
Iteración 102: [ 8.91025846e+13 5.56891154e+13 -6.12580269e+13]  
Iteración 103: [-1.22516054e+14 -7.65725336e+13 8.42297870e+13]  
Iteración 104: [ 1.68459574e+14 1.05287234e+14 -1.15815957e+14]  
Iteración 105: [-2.31631914e+14 -1.44769946e+14 1.59246941e+14]  
Iteración 106: [ 3.18493882e+14 1.99058676e+14 -2.18964544e+14]  
Iteración 107: [-4.37929088e+14 -2.73705680e+14 3.01076248e+14]  
Iteración 108: [ 6.02152496e+14 3.76345310e+14 -4.13979841e+14]  
Iteración 109: [-8.27959682e+14 -5.17474801e+14 5.69222281e+14]  
Iteración 110: [ 1.13844456e+15 7.11527852e+14 -7.82680637e+14]  
Iteración 111: [-1.56536127e+15 -9.78350796e+14 1.07618588e+15]  
Iteración 112: [ 2.15237175e+15 1.34523234e+15 -1.47975558e+15]  
Iteración 113: [-2.95951116e+15 -1.84969447e+15 2.03466392e+15]  
Iteración 114: [ 4.06932784e+15 2.54332990e+15 -2.79766289e+15]  
Iteración 115: [-5.59532578e+15 -3.49707861e+15 3.84678648e+15]  
Iteración 116: [ 7.69357295e+15 4.80848309e+15 -5.28933140e+15]  
Iteración 117: [-1.05786628e+16 -6.61166425e+15 7.27283068e+15]  
Iteración 118: [ 1.45456614e+16 9.09103835e+15 -1.00001422e+16]  
Iteración 119: [-2.00002844e+16 -1.25001777e+16 1.37501955e+16]

Iteración 120: [ 2.75003910e+16 1.71877444e+16 -1.89065188e+16]  
Iteración 121: [-3.78130376e+16 -2.36331485e+16 2.59964634e+16]  
Iteración 122: [ 5.19929268e+16 3.24955792e+16 -3.57451371e+16]  
Iteración 123: [-7.14902743e+16 -4.46814214e+16 4.91495636e+16]  
Iteración 124: [ 9.82991271e+16 6.14369545e+16 -6.75806499e+16]  
Iteración 125: [-1.35161300e+17 -8.44758124e+16 9.29233936e+16]  
Iteración 126: [ 1.85846787e+17 1.16154242e+17 -1.27769666e+17]  
Iteración 127: [-2.55539332e+17 -1.59712083e+17 1.75683291e+17]  
Iteración 128: [ 3.51366582e+17 2.19604114e+17 -2.41564525e+17]  
Iteración 129: [-4.83129050e+17 -3.01955657e+17 3.32151222e+17]  
Iteración 130: [ 6.64302444e+17 4.15189028e+17 -4.56707931e+17]  
Iteración 131: [-9.13415861e+17 -5.70884913e+17 6.27973404e+17]  
Iteración 132: [ 1.25594681e+18 7.84966756e+17 -8.63463431e+17]  
Iteración 133: [-1.72692686e+18 -1.07932929e+18 1.18726222e+18]  
Iteración 134: [ 2.37452444e+18 1.48407777e+18 -1.63248555e+18]  
Iteración 135: [-3.26497110e+18 -2.04060694e+18 2.24466763e+18]  
Iteración 136: [ 4.48933526e+18 2.80583454e+18 -3.08641799e+18]  
Iteración 137: [-6.17283598e+18 -3.85802249e+18 4.24382474e+18]  
Iteración 138: [ 8.48764948e+18 5.30478092e+18 -5.83525902e+18]  
Iteración 139: [-1.16705180e+19 -7.29407377e+18 8.02348115e+18]  
Iteración 140: [ 1.60469623e+19 1.00293514e+19 -1.10322866e+19]  
Iteración 141: [-2.20645732e+19 -1.37903582e+19 1.51693940e+19]  
Iteración 142: [ 3.03387881e+19 1.89617426e+19 -2.08579168e+19]  
Iteración 143: [-4.17158336e+19 -2.60723960e+19 2.86796356e+19]  
Iteración 144: [ 5.73592712e+19 3.58495445e+19 -3.94344990e+19]  
Iteración 145: [-7.88689979e+19 -4.92931237e+19 5.42224361e+19]  
Iteración 146: [ 1.08444872e+20 6.77780451e+19 -7.45558496e+19]  
Iteración 147: [-1.49111699e+20 -9.31948120e+19 1.02514293e+20]  
Iteración 148: [ 2.05028586e+20 1.28142867e+20 -1.40957153e+20]  
Iteración 149: [-2.81914306e+20 -1.76196441e+20 1.93816086e+20]  
Iteración 150: [ 3.87632171e+20 2.42270107e+20 -2.66497118e+20]  
Iteración 151: [-5.32994235e+20 -3.33121397e+20 3.66433537e+20]  
Iteración 152: [ 7.32867074e+20 4.58041921e+20 -5.03846113e+20]  
Iteración 153: [-1.00769223e+21 -6.29807641e+20 6.92788406e+20]  
Iteración 154: [ 1.38557681e+21 8.65985507e+20 -9.52584058e+20]  
Iteración 155: [-1.90516812e+21 -1.19073007e+21 1.30980308e+21]  
Iteración 156: [ 2.61960616e+21 1.63725385e+21 -1.80097923e+21]  
Iteración 157: [-3.60195847e+21 -2.25122404e+21 2.47634645e+21]  
Iteración 158: [ 4.95269289e+21 3.09543306e+21 -3.40497636e+21]  
Iteración 159: [-6.80995273e+21 -4.25622046e+21 4.68184250e+21]  
Iteración 160: [ 9.36368500e+21 5.85230313e+21 -6.43753344e+21]  
Iteración 161: [-1.28750669e+22 -8.04691680e+21 8.85160848e+21]  
Iteración 162: [ 1.77032170e+22 1.10645106e+22 -1.21709617e+22]  
Iteración 163: [-2.43419233e+22 -1.52137021e+22 1.67350723e+22]  
Iteración 164: [ 3.34701446e+22 2.09188404e+22 -2.30107244e+22]  
Iteración 165: [-4.60214488e+22 -2.87634055e+22 3.16397460e+22]  
Iteración 166: [ 6.32794921e+22 3.95496825e+22 -4.35046508e+22]  
Iteración 167: [-8.70093016e+22 -5.43808135e+22 5.98188948e+22]  
Iteración 168: [ 1.19637790e+23 7.47736186e+22 -8.22509804e+22]  
Iteración 169: [-1.64501961e+23 -1.02813726e+23 1.13095098e+23]  
Iteración 170: [ 2.26190196e+23 1.41368873e+23 -1.55505760e+23]  
Iteración 171: [-3.1101152e+23 -1.9438220e+23 2.1382042e+23]  
Iteración 172: [ 4.27640840e+23 2.67275525e+23 -2.94003077e+23]  
Iteración 173: [-5.88006154e+23 -3.67503846e+23 4.04254231e+23]  
Iteración 174: [ 8.08508462e+23 5.05317789e+23 -5.55849568e+23]  
Iteración 175: [-1.11169914e+24 -6.94811960e+23 7.64293156e+23]  
Iteración 176: [ 1.52858631e+24 9.55366445e+23 -1.05090309e+24]  
Iteración 177: [-2.10180618e+24 -1.31362886e+24 1.44499175e+24]  
Iteración 178: [ 2.88998350e+24 1.80623968e+24 -1.98686365e+24]  
Iteración 179: [-3.97372731e+24 -2.48357957e+24 2.73193752e+24]

Iteración 180: [ 5.46387505e+24 3.41492190e+24 -3.75641409e+24]  
Iteración 181: [-7.51282819e+24 -4.69551762e+24 5.16506938e+24]  
Iteración 182: [ 1.03301388e+25 6.45633672e+24 -7.10197040e+24]  
Iteración 183: [-1.42039408e+25 -8.87746299e+24 9.76520929e+24]  
Iteración 184: [ 1.95304186e+25 1.22065116e+25 -1.34271628e+25]  
Iteración 185: [-2.68543256e+25 -1.67839535e+25 1.84623488e+25]  
Iteración 186: [ 3.69246976e+25 2.30779360e+25 -2.53857296e+25]  
Iteración 187: [-5.07714593e+25 -3.17321620e+25 3.49053782e+25]  
Iteración 188: [ 6.98107565e+25 4.36317228e+25 -4.79948951e+25]  
Iteración 189: [-9.59897902e+25 -5.99936189e+25 6.59929807e+25]  
Iteración 190: [ 1.31985961e+26 8.24912259e+25 -9.07403485e+25]  
Iteración 191: [-1.81480697e+26 -1.13425436e+26 1.24767979e+26]  
Iteración 192: [ 2.49535958e+26 1.55959974e+26 -1.71555971e+26]  
Iteración 193: [-3.43111943e+26 -2.14444964e+26 2.35889461e+26]  
Iteración 194: [ 4.71778921e+26 2.94861826e+26 -3.24348008e+26]  
Iteración 195: [-6.48696017e+26 -4.05435011e+26 4.45978512e+26]  
Iteración 196: [ 8.91957023e+26 5.57473140e+26 -6.13220453e+26]  
Iteración 197: [-1.22644091e+27 -7.66525567e+26 8.43178124e+26]  
Iteración 198: [ 1.68635625e+27 1.05397265e+27 -1.15936992e+27]  
Iteración 199: [-2.31873984e+27 -1.44921240e+27 1.59413364e+27]  
Iteración 200: [ 3.18826728e+27 1.99266705e+27 -2.19193375e+27]  
Iteración 201: [-4.38386751e+27 -2.73991719e+27 3.01390891e+27]  
Iteración 202: [ 6.02781783e+27 3.76738614e+27 -4.14412475e+27]  
Iteración 203: [-8.28824951e+27 -5.18015594e+27 5.69817154e+27]  
Iteración 204: [ 1.13963431e+28 7.12271442e+27 -7.83498586e+27]  
Iteración 205: [-1.56699717e+28 -9.79373233e+27 1.07731056e+28]  
Iteración 206: [ 2.15462111e+28 1.34663820e+28 -1.48130202e+28]  
Iteración 207: [-2.96260403e+28 -1.85162752e+28 2.03679027e+28]  
Iteración 208: [ 4.07358054e+28 2.54598784e+28 -2.80058662e+28]  
Iteración 209: [-5.60117324e+28 -3.50073328e+28 3.85080661e+28]  
Iteración 210: [ 7.70161321e+28 4.81350826e+28 -5.29485908e+28]  
Iteración 211: [-1.05897182e+29 -6.61857385e+28 7.28043124e+28]  
Iteración 212: [ 1.45608625e+29 9.10053905e+28 -1.00105930e+29]  
Iteración 213: [-2.00211859e+29 -1.25132412e+29 1.37645653e+29]  
Iteración 214: [ 2.75291306e+29 1.72057066e+29 -1.89262773e+29]  
Iteración 215: [-3.78525546e+29 -2.36578466e+29 2.60236313e+29]  
Iteración 216: [ 5.20472626e+29 3.25295391e+29 -3.57824930e+29]  
Iteración 217: [-7.15649860e+29 -4.47281163e+29 4.92009279e+29]  
Iteración 218: [ 9.84018558e+29 6.15011599e+29 -6.76512759e+29]  
Iteración 219: [-1.35302552e+30 -8.45640948e+29 9.30205043e+29]  
Iteración 220: [ 1.86041009e+30 1.16275630e+30 -1.27903193e+30]  
Iteración 221: [-2.55806387e+30 -1.59878992e+30 1.75866891e+30]  
Iteración 222: [ 3.51733782e+30 2.19833614e+30 -2.41816975e+30]  
Iteración 223: [-4.83633950e+30 -3.02271219e+30 3.32498341e+30]  
Iteración 224: [ 6.64996682e+30 4.15622926e+30 -4.57185219e+30]  
Iteración 225: [-9.14370437e+30 -5.71481523e+30 6.28629676e+30]  
Iteración 226: [ 1.25725935e+31 7.85787094e+30 -8.64365804e+30]  
Iteración 227: [-1.72873161e+31 -1.08045725e+31 1.18850298e+31]  
Iteración 228: [ 2.37700596e+31 1.48562873e+31 -1.63419160e+31]  
Iteración 229: [-3.26838320e+31 -2.04273950e+31 2.24701345e+31]  
Iteración 230: [ 4.49402689e+31 2.80876681e+31 -3.08964349e+31]  
Iteración 231: [-6.17928698e+31 -3.86205436e+31 4.24825980e+31]  
Iteración 232: [ 8.49651960e+31 5.31032475e+31 -5.84135722e+31]  
Iteración 233: [-1.16827144e+32 -7.30169653e+31 8.03186618e+31]  
Iteración 234: [ 1.60637324e+32 1.00398327e+32 -1.10438160e+32]  
Iteración 235: [-2.2087632e+32 -1.3804770e+32 1.5185247e+32]  
Iteración 236: [ 3.03704940e+32 1.89815587e+32 -2.08797146e+32]  
Iteración 237: [-4.17594292e+32 -2.60996433e+32 2.87096076e+32]  
Iteración 238: [ 5.74192152e+32 3.58870095e+32 -3.94757105e+32]  
Iteración 239: [-7.89514209e+32 -4.93446381e+32 5.42791019e+32]

Iteración 240: [ 1.08558204e+33 6.78488774e+32 -7.46337651e+32]  
Iteración 241: [-1.49267530e+33 -9.32922064e+32 1.02621427e+33]  
Iteración 242: [ 2.05242854e+33 1.28276784e+33 -1.41104462e+33]  
Iteración 243: [-2.82208924e+33 -1.76380578e+33 1.94018635e+33]  
Iteración 244: [ 3.88037271e+33 2.42523294e+33 -2.66775624e+33]  
Iteración 245: [-5.33551247e+33 -3.33469530e+33 3.66816483e+33]  
Iteración 246: [ 7.33632965e+33 4.58520603e+33 -5.04372664e+33]  
Iteración 247: [-1.00874533e+34 -6.30465829e+33 6.93512412e+33]  
Iteración 248: [ 1.38702482e+34 8.66890516e+33 -9.53579567e+33]  
Iteración 249: [-1.90715913e+34 -1.19197446e+34 1.31117190e+34]  
Iteración 250: [ 2.62234381e+34 1.63896488e+34 -1.80286137e+34]  
Iteración 251: [-3.60572274e+34 -2.25357671e+34 2.47893438e+34]  
Iteración 252: [ 4.95786876e+34 3.09866798e+34 -3.40853478e+34]  
Iteración 253: [-6.81706955e+34 -4.26066847e+34 4.68673532e+34]  
Iteración 254: [ 9.37347063e+34 5.85841915e+34 -6.44426106e+34]  
Iteración 255: [-1.28885221e+35 -8.05532633e+34 8.86085896e+34]  
Iteración 256: [ 1.77217179e+35 1.10760737e+35 -1.21836811e+35]  
Iteración 257: [-2.43673621e+35 -1.52296013e+35 1.67525615e+35]  
Iteración 258: [ 3.35051229e+35 2.09407018e+35 -2.30347720e+35]  
Iteración 259: [-4.60695440e+35 -2.87934650e+35 3.16728115e+35]  
Iteración 260: [ 6.33456230e+35 3.95910144e+35 -4.35501158e+35]  
Iteración 261: [-8.71002317e+35 -5.44376448e+35 5.98814093e+35]  
Iteración 262: [ 1.19762819e+36 7.48517616e+35 -8.23369378e+35]  
Iteración 263: [-1.64673876e+36 -1.02921172e+36 1.13213289e+36]  
Iteración 264: [ 2.26426579e+36 1.41516612e+36 -1.55668273e+36]  
Iteración 265: [-3.11336546e+36 -1.94585341e+36 2.14043875e+36]  
Iteración 266: [ 4.28087751e+36 2.67554844e+36 -2.94310329e+36]  
Iteración 267: [-5.88620657e+36 -3.67887911e+36 4.04676702e+36]  
Iteración 268: [ 8.09353404e+36 5.05845877e+36 -5.56430465e+36]  
Iteración 269: [-1.11286093e+37 -6.95538081e+36 7.65091889e+36]  
Iteración 270: [ 1.53018378e+37 9.56364862e+36 -1.05200135e+37]  
Iteración 271: [-2.10400270e+37 -1.31500168e+37 1.44650185e+37]  
Iteración 272: [ 2.89300371e+37 1.80812732e+37 -1.98894005e+37]  
Iteración 273: [-3.97788010e+37 -2.48617506e+37 2.73479257e+37]  
Iteración 274: [ 5.46958513e+37 3.41849071e+37 -3.76033978e+37]  
Iteración 275: [-7.52067956e+37 -4.70042472e+37 5.17046720e+37]  
Iteración 276: [ 1.03409344e+38 6.46308400e+37 -7.10939239e+37]  
Iteración 277: [-1.42187848e+38 -8.88674049e+37 9.77541454e+37]  
Iteración 278: [ 1.95508291e+38 1.22192682e+38 -1.34411950e+38]  
Iteración 279: [-2.68823900e+38 -1.68014937e+38 1.84816431e+38]  
Iteración 280: [ 3.69632862e+38 2.31020539e+38 -2.54122593e+38]  
Iteración 281: [-5.08245186e+38 -3.17653241e+38 3.49418565e+38]  
Iteración 282: [ 6.98837130e+38 4.36773207e+38 -4.80450527e+38]  
Iteración 283: [-9.60901054e+38 -6.00563159e+38 6.60619475e+38]  
Iteración 284: [ 1.32123895e+39 8.25774344e+38 -9.08351778e+38]  
Iteración 285: [-1.81670356e+39 -1.13543972e+39 1.24898369e+39]  
Iteración 286: [ 2.49796739e+39 1.56122962e+39 -1.71735258e+39]  
Iteración 287: [-3.43470516e+39 -2.14669073e+39 2.36135980e+39]  
Iteración 288: [ 4.72271960e+39 2.95169975e+39 -3.24686972e+39]  
Iteración 289: [-6.49373944e+39 -4.05858715e+39 4.46444587e+39]  
Iteración 290: [ 8.92889174e+39 5.58055733e+39 -6.13861307e+39]  
Iteración 291: [-1.22772261e+40 -7.67326634e+39 8.44059297e+39]  
Iteración 292: [ 1.68811859e+40 1.05507412e+40 -1.16058153e+40]  
Iteración 293: [-2.32116307e+40 -1.45072692e+40 1.59579961e+40]  
Iteración 294: [ 3.19159922e+40 1.99474951e+40 -2.19422446e+40]  
Iteración 295: [-4.38844892e+40 -2.74278058e+40 3.01705863e+40]  
Iteración 296: [ 6.03411727e+40 3.77132329e+40 -4.14845562e+40]  
Iteración 297: [-8.29691124e+40 -5.18556953e+40 5.70412648e+40]  
Iteración 298: [ 1.14082530e+41 7.13015810e+40 -7.84317391e+40]

```
Iteración 299: [-1.56863478e+41 -9.80396739e+40  1.07843641e+41]
Iteración 300: [ 2.15687283e+41  1.34804552e+41 -1.48285007e+41]
```

## Ejercicio 7

Repita el ejercicio 11 usando el método de Jacobi

```
In [4]: import numpy as np

# b) con el metodo de gauss jacobi
def jacobi(A, b, x0, tol=1e-2, max_iterations=300):
    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for k in range(max_iterations):
        x_new = np.zeros_like(x)
        for i in range(n):
            sum1 = sum(A[i][j] * x[j] for j in range(n) if j != i)
            x_new[i] = (b[i] - sum1) / A[i][i]
        results.append(x_new.copy())
        if np.linalg.norm(x_new - x, ord=np.inf) < tol:
            print(f"Convergencia alcanzada en {k+1} iteraciones.")
            break
        x = x_new
    return results, k+1

A1 = np.array([
    [1, 0, -1],
    [-0.5, 1, -0.25],
    [1, -0.5, 1]
], dtype=float)
b1 = np.array([0.2, -1.425, 2], dtype=float)
x0 = np.zeros(3)

results1, iterations1 = jacobi(A1, b1, x0)
for i, res in enumerate(results1):
    print(f"Iteración {i}: {res}")
```

Convergencia alcanzada en 187 iteraciones.

Iteración 0: [ 0. 0. 0.]  
Iteración 1: [ 0.2 -1.425 2. ]  
Iteración 2: [ 2.2 -0.825 1.0875]  
Iteración 3: [ 1.2875 -0.053125 -0.6125 ]  
Iteración 4: [-0.4125 -0.934375 0.6859375]  
Iteración 5: [ 0.8859375 -1.45976563 1.9453125 ]  
Iteración 6: [ 2.1453125 -0.49570312 0.38417969]  
Iteración 7: [ 0.58417969 -0.25629883 -0.39316406]  
Iteración 8: [-0.19316406 -1.23120117 1.2876709 ]  
Iteración 9: [ 1.4876709 -1.19966431 1.57756348]  
Iteración 10: [ 1.77756348 -0.28677368 -0.08750305]  
Iteración 11: [ 0.11249695 -0.55809402 0.07904968]  
Iteración 12: [ 0.27904968 -1.34898911 1.60845604]  
Iteración 13: [ 1.80845604 -0.88336115 1.04645576]  
Iteración 14: [ 1.24645576 -0.25915804 -0.25013661]  
Iteración 15: [-0.05013661 -0.86430627 0.62396522]  
Iteración 16: [ 0.82396522 -1.294077 1.61798348]  
Iteración 17: [ 1.81798348 -0.60852152 0.52899628]  
Iteración 18: [ 0.72899628 -0.38375919 -0.12224424]  
Iteración 19: [ 0.07775576 -1.09106292 1.07912412]  
Iteración 20: [ 1.27912412 -1.11634109 1.37671278]  
Iteración 21: [ 1.57671278 -0.44125974 0.16270533]  
Iteración 22: [ 0.36270533 -0.59596728 0.20265735]  
Iteración 23: [ 0.40265735 -1.192983 1.33931103]  
Iteración 24: [ 1.53931103 -0.88884357 1.00085115]  
Iteración 25: [ 1.20085115 -0.4051317 0.01626719]  
Iteración 26: [ 0.21626719 -0.82050763 0.596583 ]  
Iteración 27: [ 0.796583 -1.16772066 1.373479 ]  
Iteración 28: [ 1.573479 -0.68333875 0.61955667]  
Iteración 29: [ 0.81955667 -0.48337133 0.08485162]  
Iteración 30: [ 0.28485162 -0.99400876 0.93875766]  
Iteración 31: [ 1.13875766 -1.04788477 1.218144 ]  
Iteración 32: [ 1.418144 -0.55108517 0.33729995]  
Iteración 33: [ 0.53729995 -0.63160301 0.30631342]  
Iteración 34: [ 0.50631342 -1.07977167 1.14689854]  
Iteración 35: [ 1.34689854 -0.88511866 0.95380075]  
Iteración 36: [ 1.15380075 -0.51310054 0.21054213]  
Iteración 37: [ 0.41054213 -0.79546409 0.58964898]  
Iteración 38: [ 0.78964898 -1.07231669 1.19172582]  
Iteración 39: [ 1.39172582 -0.73224405 0.67419268]  
Iteración 40: [ 0.87419268 -0.56058892 0.24215215]  
Iteración 41: [ 0.44215215 -0.92736562 0.84551286]  
Iteración 42: [ 1.04551286 -0.99254571 1.09416504]  
Iteración 43: [ 1.29416504 -0.62870231 0.45821428]  
Iteración 44: [ 0.65821428 -0.66336391 0.39148381]  
Iteración 45: [ 0.59148381 -0.99802191 1.01010376]  
Iteración 46: [ 1.21010376 -0.87673215 0.90950524]  
Iteración 47: [ 1.10950524 -0.59257181 0.35153016]  
Iteración 48: [ 0.55153016 -0.78236484 0.59420886]  
Iteración 49: [ 0.79420886 -1.00068271 1.05728742]  
Iteración 50: [ 1.25728742 -0.76357372 0.70544979]  
Iteración 51: [ 0.90544979 -0.61999384 0.36092572]  
Iteración 52: [ 0.56092572 -0.88204367 0.78455329]  
Iteración 53: [ 0.98455329 -0.94839882 0.99805244]  
Iteración 54: [ 1.19805244 -0.68321025 0.5412473 ]  
Iteración 55: [ 0.7412473 -0.69066195 0.46034244]  
Iteración 56: [ 0.66034244 -0.93929074 0.91342172]  
Iteración 57: [ 1.11342172 -0.86647335 0.87001219]  
Iteración 58: [ 1.07001219 -0.65078609 0.4533416 ]

Iteración 59: [ 0.6533416 -0.7766585 0.60459476]  
Iteración 60: [ 0.80459476 -0.94718051 0.95832914]  
Iteración 61: [ 1.15832914 -0.78312033 0.72181498]  
Iteración 62: [ 0.92181498 -0.66538168 0.45011069]  
Iteración 63: [ 0.65011069 -0.85156484 0.74549417]  
Iteración 64: [ 0.94549417 -0.91357111 0.92410689]  
Iteración 65: [ 1.12410689 -0.72122619 0.59772027]  
Iteración 66: [ 0.79772027 -0.71351649 0.51528001]  
Iteración 67: [ 0.71528001 -0.89731986 0.84552149]  
Iteración 68: [ 1.04552149 -0.85597962 0.83606006]  
Iteración 69: [ 1.03606006 -0.69322424 0.5264887 ]  
Iteración 70: [ 0.7264887 -0.7753478 0.61732782]  
Iteración 71: [ 0.81732782 -0.90742369 0.8858374 ]  
Iteración 72: [ 1.0858374 -0.79487674 0.72896033]  
Iteración 73: [ 0.92896033 -0.69984122 0.51672423]  
Iteración 74: [ 0.71672423 -0.83133878 0.72111906]  
Iteración 75: [ 0.92111906 -0.88635812 0.86760638]  
Iteración 76: [ 1.06760638 -0.74753887 0.63570188]  
Iteración 77: [ 0.83570188 -0.73227134 0.55862418]  
Iteración 78: [ 0.75862418 -0.86749301 0.79816245]  
Iteración 79: [ 0.99816245 -0.8461473 0.80762931]  
Iteración 80: [ 1.00762931 -0.72401145 0.5787639 ]  
Iteración 81: [ 0.7787639 -0.77649437 0.63036497]  
Iteración 82: [ 0.83036497 -0.87802681 0.83298891]  
Iteración 83: [ 1.03298891 -0.80157029 0.73062163]  
Iteración 84: [ 0.93062163 -0.72585014 0.56622594]  
Iteración 85: [ 0.76622594 -0.8181327 0.7064533 ]  
Iteración 86: [ 0.9064533 -0.8652737 0.82470771]  
Iteración 87: [ 1.02470771 -0.76559642 0.66090985]  
Iteración 88: [ 0.86090985 -0.74741868 0.59249408]  
Iteración 89: [ 0.79249408 -0.84642156 0.76538081]  
Iteración 90: [ 0.96538081 -0.83740776 0.78429514]  
Iteración 91: [ 0.98429514 -0.74623581 0.61591531]  
Iteración 92: [ 0.81591531 -0.7788736 0.64258695]  
Iteración 93: [ 0.84258695 -0.85639561 0.79464789]  
Iteración 94: [ 0.99464789 -0.80504455 0.72921524]  
Iteración 95: [ 0.92921524 -0.74537224 0.60282984]  
Iteración 96: [ 0.80282984 -0.80968492 0.69809864]  
Iteración 97: [ 0.89809864 -0.84906042 0.7923277 ]  
Iteración 98: [ 0.9923277 -0.77786876 0.67737115]  
Iteración 99: [ 0.87737115 -0.75949336 0.61873792]  
Iteración 100: [ 0.81873792 -0.83162994 0.74288217]  
Iteración 101: [ 0.94288217 -0.8299105 0.76544711]  
Iteración 102: [ 0.96544711 -0.76219714 0.64216258]  
Iteración 103: [ 0.84216258 -0.7817358 0.65345432]  
Iteración 104: [ 0.85345432 -0.84055513 0.76696952]  
Iteración 105: [ 0.96696952 -0.80653046 0.72626812]  
Iteración 106: [ 0.92626812 -0.75994821 0.62976525]  
Iteración 107: [ 0.82976525 -0.80442463 0.69375778]  
Iteración 108: [ 0.89375778 -0.83667793 0.76802243]  
Iteración 109: [ 0.96802243 -0.7861155 0.68790326]  
Iteración 110: [ 0.88790326 -0.76901297 0.63891982]  
Iteración 111: [ 0.83891982 -0.82131842 0.72759026]  
Iteración 112: [ 0.92759026 -0.82364253 0.75042098]  
Iteración 113: [ 0.95042098 -0.77359963 0.66058848]  
Iteración 114: [ 0.86058848 -0.78464239 0.66277921]  
Iteración 115: [ 0.86277921 -0.82901096 0.74709033]  
Iteración 116: [ 0.94709033 -0.80683781 0.72271531]  
Iteración 117: [ 0.92271531 -0.77077601 0.64949077]  
Iteración 118: [ 0.84949077 -0.80126965 0.69189669]

Iteración 119: [ 0.89189669 -0.82728045 0.74987441]  
Iteración 120: [ 0.94987441 -0.79158306 0.69446309]  
Iteración 121: [ 0.89446309 -0.77644702 0.65433407]  
Iteración 122: [ 0.85433407 -0.81418494 0.7173134 ]  
Iteración 123: [ 0.9173134 -0.81850462 0.73857347]  
Iteración 124: [ 0.93857347 -0.78169994 0.6734343 ]  
Iteración 125: [ 0.8734343 -0.78735469 0.67057657]  
Iteración 126: [ 0.87057657 -0.82063871 0.73288836]  
Iteración 127: [ 0.93288836 -0.80648963 0.71910408]  
Iteración 128: [ 0.91910408 -0.7787798 0.66386683]  
Iteración 129: [ 0.86386683 -0.79948125 0.69150602]  
Iteración 130: [ 0.89150602 -0.82019008 0.73639254]  
Iteración 131: [ 0.93639254 -0.79514885 0.69839894]  
Iteración 132: [ 0.89839894 -0.78220399 0.66603303]  
Iteración 133: [ 0.86603303 -0.80929227 0.71049906]  
Iteración 134: [ 0.91049906 -0.81435872 0.72932083]  
Iteración 135: [ 0.92932083 -0.78742026 0.68232158]  
Iteración 136: [ 0.88232158 -0.78975919 0.67696904]  
Iteración 137: [ 0.87696904 -0.81459695 0.72279883]  
Iteración 138: [ 0.92279883 -0.80581577 0.71573249]  
Iteración 139: [ 0.91573249 -0.78466746 0.67429328]  
Iteración 140: [ 0.87429328 -0.79856044 0.69193378]  
Iteración 141: [ 0.89193378 -0.81486991 0.7264265 ]  
Iteración 142: [ 0.9264265 -0.79742648 0.70063126]  
Iteración 143: [ 0.90063126 -0.78662893 0.67486026]  
Iteración 144: [ 0.87486026 -0.8059693 0.70605427]  
Iteración 145: [ 0.90605427 -0.8110563 0.72215509]  
Iteración 146: [ 0.92215509 -0.79143409 0.68841758]  
Iteración 147: [ 0.88841758 -0.79181806 0.68212786]  
Iteración 148: [ 0.88212786 -0.81025925 0.71567339]  
Iteración 149: [ 0.91567339 -0.80501772 0.71274251]  
Iteración 150: [ 0.91274251 -0.78897768 0.68181775]  
Iteración 151: [ 0.88181775 -0.79817431 0.69276865]  
Iteración 152: [ 0.89276865 -0.81089896 0.7190951 ]  
Iteración 153: [ 0.9190951 -0.7988419 0.70178187]  
Iteración 154: [ 0.90178187 -0.79000698 0.68148395]  
Iteración 155: [ 0.88148395 -0.80373808 0.70321464]  
Iteración 156: [ 0.90321464 -0.80845436 0.71664701]  
Iteración 157: [ 0.91664701 -0.79423093 0.69255818]  
Iteración 158: [ 0.89255818 -0.79353695 0.68623753]  
Iteración 159: [ 0.88623753 -0.80716153 0.71067335]  
Iteración 160: [ 0.91067335 -0.8042129 0.71018171]  
Iteración 161: [ 0.91018171 -0.7921179 0.6872202 ]  
Iteración 162: [ 0.8872202 -0.79810409 0.69375934]  
Iteración 163: [ 0.89375934 -0.80795006 0.71372775]  
Iteración 164: [ 0.91372775 -0.79968839 0.70226563]  
Iteración 165: [ 0.90226563 -0.79256972 0.68642805]  
Iteración 166: [ 0.88642805 -0.80226017 0.70144951]  
Iteración 167: [ 0.90144951 -0.80642359 0.71244186]  
Iteración 168: [ 0.91244186 -0.79616478 0.69533869]  
Iteración 169: [ 0.89533869 -0.7949444 0.68947575]  
Iteración 170: [ 0.88947575 -0.80496172 0.70718911]  
Iteración 171: [ 0.90718911 -0.80346485 0.70804339]  
Iteración 172: [ 0.90804339 -0.7943946 0.69107847]  
Iteración 173: [ 0.89107847 -0.79820869 0.69475931]  
Iteración 174: [ 0.89475931 -0.80577094 0.70981719]  
Iteración 175: [ 0.90981719 -0.80016605 0.70235522]  
Iteración 176: [ 0.90235522 -0.7945026 0.69009979]  
Iteración 177: [ 0.89009979 -0.80129744 0.70039348]  
Iteración 178: [ 0.90039348 -0.80485174 0.70925149]

```
Iteración 179: [ 0.90925149 -0.79749039  0.69718065]
Iteración 180: [ 0.89718065 -0.79607909  0.69200332]
Iteración 181: [ 0.89200332 -0.80340885  0.7047798 ]
Iteración 182: [ 0.9047798  -0.80280339  0.70629226]
Iteración 183: [ 0.90629226 -0.79603703  0.6938185 ]
Iteración 184: [ 0.8938185  -0.79839924  0.69568922]
Iteración 185: [ 0.89568922 -0.80416844  0.70698188]
Iteración 186: [ 0.90698188 -0.80040992  0.70222656]
Iteración 187: [ 0.90222656 -0.79595242  0.69281316]
```

```
In [3]: # c) con metodo de gauss jacobi
A2 = np.array([
    [1, 0, -2],
    [-0.5, 1, -0.25],
    [1, -0.5, 1]
], dtype=float)

results2, iterations2 = jacobi(A2, b1, x0)
for i, res in enumerate(results2):
    print(f"Iteración {i}: {res}")
```

Iteración 0: [ 0. 0. 0.]  
Iteración 1: [ 0.2 -1.425 2. ]  
Iteración 2: [ 4.2 -0.825 1.0875]  
Iteración 3: [ 2.375 0.946875 -2.6125 ]  
Iteración 4: [-5.025 -0.890625 0.0984375]  
Iteración 5: [ 0.396875 -3.91289062 6.5796875 ]  
Iteración 6: [13.359375 0.41835938 -0.35332031]  
Iteración 7: [ -0.50664063 5.16635742 -11.15019531]  
Iteración 8: [-22.10039062 -4.46586914 5.08981934]  
Iteración 9: [ 10.37963867 -11.20274048 21.86745605]  
Iteración 10: [ 43.93491211 9.23168335 -13.98100891]  
Iteración 11: [-27.76201782 17.04720383 -37.31907043]  
Iteración 12: [-74.43814087 -24.63577652 38.28561974]  
Iteración 13: [ 76.77123947 -29.0726655 64.12025261]  
Iteración 14: [128.44050522 52.99068289 -89.30757222]  
Iteración 15: [-178.41514444 40.46835955 -99.94516377]  
Iteración 16: [-199.69032755 -115.61886317 200.64932422]  
Iteración 17: [401.49864844 -51.10783272 143.88089597]  
Iteración 18: [ 287.96179193 235.29454821 -425.0525648 ]  
Iteración 19: [-849.9051296 36.29275477 -168.31451783]  
Iteración 20: [-336.42903565 -468.45619426 870.05150698]  
Iteración 21: [1740.30301397 47.87335892 104.20093852]  
Iteración 22: [ 208.60187705 894.77674162 -1714.36633451]  
Iteración 23: [-3428.53266902 -325.7156451 240.78649376]  
Iteración 24: [ 481.77298752 -1655.49471107 3267.67484647]  
Iteración 25: [ 6535.54969293 1056.38020537 -1307.52034305]  
Iteración 26: [-2614.8406861 2939.4697607 -6005.35959024]  
Iteración 27: [-12010.51918049 -2810.18524061 4086.57556645]  
Iteración 28: [ 8173.35113291 -4985.04069863 10607.42656018]  
Iteración 29: [ 21215.05312036 6737.1072065 -10663.87148222]  
Iteración 30: [-21327.54296445 7940.13368963 -17844.49951711]  
Iteración 31: [-35688.79903423 -15126.3213615 25299.60980926]  
Iteración 32: [ 50599.41961852 -11520.9220648 28127.63835348]  
Iteración 33: [ 56255.47670696 32330.19439763 -56357.88065092]  
Iteración 34: [-112715.56130184 14036.84319075 -40088.37950814]  
Iteración 35: [-80176.55901628 -66381.30052796 119735.98289722]  
Iteración 36: [239472.16579443 -10155.70878384 46987.9087523 ]  
Iteración 37: [ 93976.0175046 131481.63508529 -244548.02018635]  
Iteración 38: [-489095.8403727 -14150.42129429 -28233.19996196]  
Iteración 39: [-56466.19992391 -251607.64517684 482022.62972556]  
Iteración 40: [964045.45945112 92271.13246943 -69335.62266451]  
Iteración 41: [-138671.04532901 464687.39905943 -917907.8932164 ]  
Iteración 42: [-1835815.58643281 -298813.92096861 371016.74485873]  
Iteración 43: [ 742033.68971746 -825155.03200172 1686410.6259485 ]  
Iteración 44: [ 3372821.45189701 792618.07634585 -1154609.20571832]  
Iteración 45: [-2309218.21143664 1397756.99951892 -2976510.41372408]  
Iteración 46: [-5953020.62744816 -1898738.13414934 3008098.7111961 ]  
Iteración 47: [ 6016197.6223922 -2224487.06092505 5003653.56037349]  
Iteración 48: [10007307.32074698 4259010.77628947 -7128439.15285472]  
Iteración 49: [-14256878.10570945 3221542.44715981 -7877799.93260224]  
Iteración 50: [-15755599.66520448 -9097890.46100529 15867651.32928935]  
Iteración 51: [31735302.8585787 -3910888.4252799 11206656.43470184]  
Iteración 52: [ 22413313.06940368 18669314.11296481 -33690745.07121865]  
Iteración 53: [-67381489.9424373 2783968.84189718 -13078654.01292128]  
Iteración 54: [-26157307.82584255 -36960409.89944897 68773476.3633859 ]  
Iteración 55: [1.37546953e+08 4.11471375e+06 7.67710488e+06]  
Iteración 56: [ 1.53542100e+07 7.06927513e+07 -1.35489594e+08]  
Iteración 57: [-2.70979188e+08 -2.61952950e+07 1.99921677e+07]  
Iteración 58: [ 3.99843356e+07 -1.30491553e+08 2.57881542e+08]  
Iteración 59: [ 5.15763085e+08 8.44625520e+07 -1.05230110e+08]

Iteración 60: [-2.10460220e+08 2.31574014e+08 -4.73531807e+08]  
Iteración 61: [-9.47063614e+08 -2.23613063e+08 3.26247229e+08]  
Iteración 62: [ 6.52494458e+08 -3.91970001e+08 8.35257084e+08]  
Iteración 63: [ 1.67051417e+09 5.35061499e+08 -8.48479457e+08]  
Iteración 64: [-1.69695891e+09 6.23137219e+08 -1.40298342e+09]  
Iteración 65: [-2.80596683e+09 -1.19922531e+09 2.00852753e+09]  
Iteración 66: [ 4.01705505e+09 -9.00851537e+08 2.20635418e+09]  
Iteración 67: [ 4.41270836e+09 2.56011607e+09 -4.46748082e+09]  
Iteración 68: [-8.93496163e+09 1.08948397e+09 -3.13265032e+09]  
Iteración 69: [-6.26530065e+09 -5.25064340e+09 9.47970362e+09]  
Iteración 70: [ 1.89594072e+10 -7.62724419e+08 3.63997895e+09]  
Iteración 71: [ 7.27995790e+09 1.03896984e+10 -1.93407695e+10]  
Iteración 72: [-3.86815389e+10 -1.19521342e+09 -2.08510872e+09]  
Iteración 73: [-4.17021744e+09 -1.98620466e+10 3.80839322e+10]  
Iteración 74: [ 7.61678644e+10 7.43587433e+09 -5.76080588e+09]  
Iteración 75: [-1.15216118e+10 3.66437307e+10 -7.24499272e+10]  
Iteración 76: [-1.44899854e+11 -2.38732877e+10 2.98434771e+10]  
Iteración 77: [ 5.96869543e+10 -6.49890580e+10 1.32963211e+11]  
Iteración 78: [ 2.65926421e+11 6.30842798e+10 -9.21814832e+10]  
Iteración 79: [-1.84362966e+11 1.09917840e+11 -2.34384281e+11]  
Iteración 80: [-4.68768563e+11 -1.50777554e+11 2.39321886e+11]  
Iteración 81: [ 4.78643773e+11 -1.74553810e+11 3.93379786e+11]  
Iteración 82: [ 7.86759572e+11 3.37666833e+11 -5.65920678e+11]  
Iteración 83: [-1.13184136e+12 2.51899617e+11 -6.17926155e+11]  
Iteración 84: [-1.23585231e+12 -7.20402217e+11 1.25779116e+12]  
Iteración 85: [ 2.51558233e+12 -3.03478365e+11 8.75651203e+11]  
Iteración 86: [ 1.75130241e+12 1.47670396e+12 -2.66732151e+12]  
Iteración 87: [-5.33464302e+12 2.08820825e+11 -1.01295042e+12]  
Iteración 88: [-2.02590085e+12 -2.92055912e+12 5.43905343e+12]  
Iteración 89: [1.08781069e+13 3.46812935e+11 5.65621289e+11]  
Iteración 90: [ 1.13124258e+12 5.58045875e+12 -1.07047004e+13]  
Iteración 91: [-2.14094008e+13 -2.11055381e+12 1.65898680e+12]  
Iteración 92: [ 3.31797360e+12 -1.02899537e+13 2.03541239e+13]  
Iteración 93: [ 4.07082478e+13 6.74751777e+12 -8.46295045e+12]  
Iteración 94: [-1.69259009e+13 1.82383863e+13 -3.73344889e+13]  
Iteración 95: [-7.46689778e+13 -1.77965727e+13 2.60450940e+13]  
Iteración 96: [ 5.20901881e+13 -3.08232154e+13 6.57706914e+13]  
Iteración 97: [ 1.31541383e+14 4.24877669e+13 -6.75017957e+13]  
Iteración 98: [-1.35003591e+14 4.88952425e+13 -1.10297499e+14]  
Iteración 99: [-2.20594999e+14 -9.50761706e+13 1.59451213e+14]  
Iteración 100: [ 3.18902425e+14 -7.04346962e+13 1.73056914e+14]  
Iteración 101: [ 3.46113827e+14 2.02715441e+14 -3.54119774e+14]  
Iteración 102: [-7.08239547e+14 8.45269702e+13 -2.44756107e+14]  
Iteración 103: [-4.89512213e+14 -4.15308800e+14 7.50503032e+14]  
Iteración 104: [ 1.50100606e+15 -5.71303485e+13 2.81857813e+14]  
Iteración 105: [ 5.63715626e+14 8.20967486e+14 -1.52957124e+15]  
Iteración 106: [-3.05914248e+15 -1.00534997e+14 -1.53231883e+14]  
Iteración 107: [-3.06463766e+14 -1.56787921e+15 3.00887498e+15]  
Iteración 108: [ 6.01774996e+15 5.98986862e+14 -4.77475838e+14]  
Iteración 109: [-9.54951677e+14 2.88950602e+15 -5.71825653e+15]  
Iteración 110: [-1.14365131e+16 -1.90703997e+15 2.39970469e+15]  
Iteración 111: [ 4.79940937e+15 -5.11833036e+15 1.04829931e+16]  
Iteración 112: [ 2.09659861e+16 5.02045295e+15 -7.35857455e+15]  
Iteración 113: [-1.47171491e+16 8.64334943e+15 -1.84557597e+16]  
Iteración 114: [-3.69115193e+16 -1.19725145e+16 1.90388238e+16]  
Iteración 115: [ 3.80776476e+16 -1.36960537e+16 3.09252621e+16]  
Iteración 116: [ 6.18505242e+16 2.67701393e+16 -4.49256745e+16]  
Iteración 117: [-8.98513490e+16 1.96938435e+16 -4.84654545e+16]  
Iteración 118: [-9.69309090e+16 -5.70420381e+16 9.96982707e+16]  
Iteración 119: [ 1.99396541e+17 -2.35408868e+16 6.84098900e+16]

Iteración 120: [ 1.36819780e+17 1.16800743e+17 -2.11166985e+17]  
Iteración 121: [-4.22333970e+17 1.56181438e+16 -7.84194083e+16]  
Iteración 122: [-1.56838817e+17 -2.30771837e+17 4.30143042e+17]  
Iteración 123: [8.60286083e+17 2.91163521e+16 4.14528982e+16]  
Iteración 124: [ 8.29057964e+16 4.40506266e+17 -8.45727907e+17]  
Iteración 125: [-1.69145581e+18 -1.69979079e+17 1.37347337e+17]  
Iteración 126: [ 2.74694673e+17 -8.11391073e+17 1.60646628e+18]  
Iteración 127: [ 3.21293255e+18 5.38963905e+17 -6.80390210e+17]  
Iteración 128: [-1.36078042e+18 1.43636872e+18 -2.94345060e+18]  
Iteración 129: [-5.88690119e+18 -1.41625286e+18 2.07896478e+18]  
Iteración 130: [ 4.15792956e+18 -2.42370940e+18 5.17877477e+18]  
Iteración 131: [ 1.03575495e+19 3.37365847e+18 -5.36978426e+18]  
Iteración 132: [-1.07395685e+19 3.83632870e+18 -8.67072029e+18]  
Iteración 133: [-1.73414406e+19 -7.53746434e+18 1.26577329e+19]  
Iteración 134: [ 2.53154658e+19 -5.50628708e+18 1.35727084e+19]  
Iteración 135: [ 2.71454168e+19 1.60509100e+19 -2.80686093e+19]  
Iteración 136: [-5.61372186e+19 6.55555610e+18 -1.91199619e+19]  
Iteración 137: [-3.82399237e+19 -3.28485998e+19 5.94149966e+19]  
Iteración 138: [ 1.18829993e+20 -4.26621269e+18 2.18156238e+19]  
Iteración 139: [ 4.36312476e+19 6.48689026e+19 -1.20963100e+20]  
Iteración 140: [-2.41926199e+20 -8.42515108e+18 -1.11967964e+19]  
Iteración 141: [-2.23935927e+19 -1.23762299e+20 2.37713624e+20]  
Iteración 142: [ 4.75427247e+20 4.82316096e+19 -3.94875566e+19]  
Iteración 143: [-7.89751133e+19 2.27841735e+20 -4.51311443e+20]  
Iteración 144: [-9.02622885e+20 -1.52315417e+20 1.92895981e+20]  
Iteración 145: [ 3.85791961e+20 -4.03087447e+20 8.26465176e+20]  
Iteración 146: [ 1.65293035e+21 3.99512275e+20 -5.87335685e+20]  
Iteración 147: [-1.17467137e+21 6.79631255e+20 -1.45317422e+21]  
Iteración 148: [-2.90634843e+21 -9.50629239e+20 1.51448700e+21]  
Iteración 149: [ 3.02897399e+21 -1.07455247e+21 2.43103381e+21]  
Iteración 150: [ 4.86206762e+21 2.12224545e+21 -3.56625023e+21]  
Iteración 151: [-7.13250046e+21 1.53947126e+21 -3.80094490e+21]  
Iteración 152: [-7.60188980e+21 -4.51648645e+21 7.90223608e+21]  
Iteración 153: [ 1.58044722e+22 -1.82538588e+21 5.34364657e+21]  
Iteración 154: [ 1.06872931e+22 9.23814773e+21 -1.67171651e+22]  
Iteración 155: [-3.34343302e+22 1.16435530e+21 -6.06821928e+21]  
Iteración 156: [-1.21364386e+22 -1.82342199e+22 3.40165079e+22]  
Iteración 157: [6.80330157e+22 2.43590768e+21 3.01932860e+21]  
Iteración 158: [ 6.03865720e+21 3.47713400e+22 -6.68150619e+22]  
Iteración 159: [-1.33630124e+23 -1.36844369e+22 1.13470128e+22]  
Iteración 160: [ 2.26940256e+22 -6.39783087e+22 1.26787905e+23]  
Iteración 161: [ 2.53575811e+23 4.30439891e+22 -5.46831799e+22]  
Iteración 162: [-1.09366360e+23 1.13117110e+23 -2.32053816e+23]  
Iteración 163: [-4.64107632e+23 -1.12696634e+23 1.65924915e+23]  
Iteración 164: [ 3.31849830e+23 -1.90572587e+23 4.07759315e+23]  
Iteración 165: [ 8.15518630e+23 2.67864744e+23 -4.27136124e+23]  
Iteración 166: [-8.54272247e+23 3.00975284e+23 -6.81586258e+23]  
Iteración 167: [-1.36317252e+24 -5.97532688e+23 1.00475989e+24]  
Iteración 168: [ 2.00951978e+24 -4.30396286e+23 1.06440617e+24]  
Iteración 169: [ 2.12881234e+24 1.27086143e+24 -2.22471792e+24]  
Iteración 170: [-4.44943584e+24 5.08226692e+23 -1.49338163e+24]  
Iteración 171: [-2.98676326e+24 -2.59806333e+24 4.70354919e+24]  
Iteración 172: [ 9.40709838e+24 -3.17494331e+23 1.68773159e+24]  
Iteración 173: [ 3.37546319e+24 5.12548209e+24 -9.56584555e+24]  
Iteración 174: [-1.91316911e+25 -7.03729794e+23 -8.12722141e+23]  
Iteración 175: [-1.62544428e+24 -9.76902608e+24 1.87798262e+25]  
Iteración 176: [ 3.75596524e+25 3.88223441e+24 -3.25906876e+24]  
Iteración 177: [-6.51813752e+24 1.79650590e+25 -3.56185352e+25]  
Iteración 178: [-7.12370704e+25 -1.21637026e+25 1.55006670e+25]  
Iteración 179: [ 3.10013340e+25 -3.17433684e+25 6.51552191e+25]

Iteración 180: [ 1.30310438e+26 3.17894718e+25 -4.68730183e+25]  
Iteración 181: [-9.37460365e+25 5.34369645e+25 -1.14415702e+26]  
Iteración 182: [-2.28831405e+26 -7.54769438e+25 1.20464519e+26]  
Iteración 183: [ 2.40929038e+26 -8.42995726e+25 1.91092933e+26]  
Iteración 184: [ 3.82185865e+26 1.68237752e+26 -2.83078824e+26]  
Iteración 185: [-5.66157648e+26 1.20323227e+26 -2.98066989e+26]  
Iteración 186: [-5.96133979e+26 -3.57595571e+26 6.26319261e+26]  
Iteración 187: [ 1.25263852e+27 -1.41487174e+26 4.17336193e+26]  
Iteración 188: [ 8.34672386e+26 7.30653309e+26 -1.32338211e+27]  
Iteración 189: [-2.64676422e+27 8.64906658e+25 -4.69345732e+26]  
Iteración 190: [-9.38691463e+26 -1.44071854e+27 2.69000955e+27]  
Iteración 191: [5.38001910e+27 2.03156656e+26 2.18332192e+26]  
Iteración 192: [ 4.36664384e+26 2.74459260e+27 -5.27844077e+27]  
Iteración 193: [-1.05568815e+28 -1.10127800e+27 9.35631915e+26]  
Iteración 194: [ 1.87126383e+27 -5.04453280e+27 1.00062425e+28]  
Iteración 195: [ 2.00124851e+28 3.43719255e+27 -4.39353023e+27]  
Iteración 196: [-8.78706046e+27 8.90785999e+27 -1.82938888e+28]  
Iteración 197: [-3.65877776e+28 -8.96700243e+27 1.32409905e+28]  
Iteración 198: [ 2.64819809e+28 -1.49836412e+28 3.21042764e+28]  
Iteración 199: [ 6.42085528e+28 2.12670596e+28 -3.39738015e+28]  
Iteración 200: [-6.79476030e+28 2.36108260e+28 -5.35750231e+28]  
Iteración 201: [-1.07150046e+29 -4.73675573e+28 7.97530160e+28]  
Iteración 202: [ 1.59506032e+29 -3.36367691e+28 8.34662675e+28]  
Iteración 203: [ 1.66932535e+29 1.00619583e+29 -1.76324417e+29]  
Iteración 204: [-3.52648833e+29 3.93851633e+28 -1.16622744e+29]  
Iteración 205: [-2.33245487e+29 -2.05480102e+29 3.72341415e+29]  
Iteración 206: [ 7.44682830e+29 -2.35373898e+28 1.30505436e+29]  
Iteración 207: [ 2.61010872e+29 4.04967774e+29 -7.56451525e+29]  
Iteración 208: [-1.51290305e+30 -5.86074453e+28 -5.85269847e+28]  
Iteración 209: [-1.17053969e+29 -7.71083271e+29 1.48359933e+30]  
Iteración 210: [ 2.96719865e+30 3.12372847e+29 -2.68487666e+29]  
Iteración 211: [-5.36975332e+29 1.41647741e+30 -2.81101223e+30]  
Iteración 212: [-5.62202446e+30 -9.71240723e+29 1.24521404e+30]  
Iteración 213: [ 2.49042807e+30 -2.49970872e+30 5.13640410e+30]  
Iteración 214: [ 1.02728082e+31 2.52931506e+30 -3.74028243e+30]  
Iteración 215: [-7.48056487e+30 4.20133349e+30 -9.00815067e+30]  
Iteración 216: [-1.80163013e+31 -5.99232010e+30 9.58123161e+30]  
Iteración 217: [ 1.91624632e+31 -6.61284276e+30 1.50201413e+31]  
Iteración 218: [ 3.00402826e+31 1.33362669e+31 -2.24688846e+31]  
Iteración 219: [-4.49377692e+31 9.40292013e+30 -2.33721491e+31]  
Iteración 220: [-4.67442982e+31 -2.83119219e+31 4.96392293e+31]  
Iteración 221: [ 9.92784586e+31 -1.09623418e+31 3.25883372e+31]  
Iteración 222: [ 6.51766745e+31 5.77863136e+31 -1.04759629e+32]  
Iteración 223: [-2.09519259e+32 6.39842989e+30 -3.62835177e+31]  
Iteración 224: [-7.25670354e+31 -1.13830509e+32 2.12718474e+32]  
Iteración 225: [4.25436948e+32 1.68961008e+31 1.56517810e+31]  
Iteración 226: [ 3.13035619e+31 2.16631419e+32 -4.16988897e+32]  
Iteración 227: [-8.33977795e+32 -8.85954434e+31 7.70121476e+31]  
Iteración 228: [ 1.54024295e+32 -3.97735860e+32 7.89680073e+32]  
Iteración 229: [ 1.57936015e+33 2.74432166e+32 -3.52892225e+32]  
Iteración 230: [-7.05784451e+32 7.01457017e+32 -1.44214406e+33]  
Iteración 231: [-2.88428813e+33 -7.13428241e+32 1.05651296e+33]  
Iteración 232: [ 2.11302592e+33 -1.17801582e+33 2.52757401e+33]  
Iteración 233: [ 5.05514801e+33 1.68840646e+33 -2.70203383e+33]  
Iteración 234: [-5.40406766e+33 1.85206555e+33 -4.21094478e+33]  
Iteración 235: [-8.42188956e+33 -3.75477002e+33 6.33010043e+33]  
Iteración 236: [ 1.26602009e+34 -2.62841967e+33 6.54450455e+33]  
Iteración 237: [ 1.30890091e+34 7.96622657e+33 -1.39744107e+34]  
Iteración 238: [-2.79488214e+34 3.05090187e+33 -9.10589581e+33]  
Iteración 239: [-1.82117916e+34 -1.62508847e+34 2.94742723e+34]

Iteración 240: [ 5.89485447e+34 -1.73732773e+33 1.00863493e+34]  
Iteración 241: [ 2.01726986e+34 3.19958597e+34 -5.98172085e+34]  
Iteración 242: [-1.19634417e+35 -4.86795284e+33 -4.17476875e+33]  
Iteración 243: [-8.34953751e+33 -6.08609007e+34 1.17200441e+35]  
Iteración 244: [ 2.34400881e+35 2.51253414e+34 -2.20809129e+34]  
Iteración 245: [-4.41618257e+34 1.11680212e+35 -2.21838211e+35]  
Iteración 246: [-4.43676421e+35 -7.75404655e+34 1.00001932e+35]  
Iteración 247: [ 2.00003864e+35 -1.96837728e+35 4.04906189e+35]  
Iteración 248: [ 8.09812377e+35 2.01228479e+35 -2.98422728e+35]  
Iteración 249: [-5.96845455e+35 3.30300507e+35 -7.09198138e+35]  
Iteración 250: [-1.41839628e+36 -4.75722262e+35 7.61995709e+35]  
Iteración 251: [ 1.52399142e+36 -5.18699210e+35 1.18053514e+36]  
Iteración 252: [ 2.36107029e+36 1.05712949e+36 -1.78334102e+36]  
Iteración 253: [-3.56668205e+36 7.34699888e+35 -1.83250554e+36]  
Iteración 254: [-3.66501108e+36 -2.24146741e+36 3.93403199e+36]  
Iteración 255: [ 7.86806398e+36 -8.48997543e+35 2.54427738e+36]  
Iteración 256: [ 5.08855475e+36 4.57010133e+36 -8.29256275e+36]  
Iteración 257: [-1.65851255e+37 4.71136690e+35 -2.80350409e+36]  
Iteración 258: [-5.60700818e+36 -8.99343877e+36 1.68206938e+37]  
Iteración 259: [ 3.36413877e+37 1.40166937e+36 1.11028879e+36]  
Iteración 260: [ 2.22057758e+36 1.70982660e+37 -3.29405530e+37]  
Iteración 261: [-6.58811060e+37 -7.12484946e+36 6.32855544e+36]  
Iteración 262: [ 1.26571109e+37 -3.13584141e+37 6.23186813e+37]  
Iteración 263: [ 1.24637363e+38 2.19082258e+37 -2.83363180e+37]  
Iteración 264: [-5.66726359e+37 5.52346018e+37 -1.13683250e+38]  
Iteración 265: [-2.27366499e+38 -5.67571304e+37 8.42899368e+37]  
Iteración 266: [ 1.68579874e+38 -9.26107655e+37 1.98987934e+38]  
Iteración 267: [ 3.97975868e+38 1.34036920e+38 -2.14885256e+38]  
Iteración 268: [-4.29770513e+38 1.45266620e+38 -3.30957408e+38]  
Iteración 269: [-6.61914816e+38 -2.97624608e+38 5.02403823e+38]  
Iteración 270: [ 1.00480765e+39 -2.05356452e+38 5.13102512e+38]  
Iteración 271: [ 1.02620502e+39 6.30679451e+38 -1.10748587e+39]  
Iteración 272: [-2.21497174e+39 2.36231044e+38 -7.10865299e+38]  
Iteración 273: [-1.42173060e+39 -1.28520220e+39 2.33308727e+39]  
Iteración 274: [ 4.66617453e+39 -1.27593482e+38 7.79129499e+38]  
Iteración 275: [ 1.55825900e+39 2.52786964e+39 -4.72997127e+39]  
Iteración 276: [-9.45994254e+39 -4.03363319e+38 -2.94324179e+38]  
Iteración 277: [-5.88648357e+38 -4.80355232e+39 9.25826089e+39]  
Iteración 278: [ 1.85165218e+40 2.02024104e+39 -1.81312780e+39]  
Iteración 279: [-3.62625560e+39 8.80497893e+39 -1.75064012e+40]  
Iteración 280: [-3.50128025e+40 -6.18972811e+39 8.02874507e+39]  
Iteración 281: [ 1.60574901e+40 -1.54992150e+40 3.19179384e+40]  
Iteración 282: [ 6.38358769e+40 1.60082297e+40 -2.38070976e+40]  
Iteración 283: [-4.76141953e+40 2.59661640e+40 -5.58317620e+40]  
Iteración 284: [-1.11663524e+41 -3.77650381e+40 6.05972773e+40]  
Iteración 285: [ 1.21194555e+41 -4.06824427e+40 9.27810050e+40]  
Iteración 286: [ 1.85562010e+41 8.37925285e+40 -1.41535776e+41]  
Iteración 287: [-2.83071552e+41 5.73970610e+40 -1.43665746e+41]  
Iteración 288: [-2.87331492e+41 -1.77452212e+41 3.11770082e+41]  
Iteración 289: [ 6.23540165e+41 -6.57232252e+40 1.98605385e+41]  
Iteración 290: [ 3.97210771e+41 3.61421429e+41 -6.56401777e+41]  
Iteración 291: [-1.31280355e+42 3.45049410e+40 -2.16500056e+41]  
Iteración 292: [-4.33000113e+41 -7.10526791e+41 1.33005603e+42]  
Iteración 293: [ 2.66011205e+42 1.16013950e+41 7.77367170e+40]  
Iteración 294: [ 1.55473434e+41 1.34949020e+42 -2.60210508e+42]  
Iteración 295: [-5.20421015e+42 -5.72789552e+41 5.19271668e+41]  
Iteración 296: [ 1.03854334e+42 -2.47228716e+42 4.91781537e+42]  
Iteración 297: [ 9.83563075e+42 1.74872551e+42 -2.27468692e+42]  
Iteración 298: [-4.54937383e+42 4.34914365e+42 -8.96126799e+42]

Iteración 299: [-1.79225360e+43 -4.51500391e+42 6.72394565e+42]

Iteración 300: [ 1.34478913e+43 -7.28028158e+42 1.56650340e+43]

## Ejercicio 8

8. Un cable coaxial está formado por un conductor interno de 0.1 pulgadas cuadradas y un conductor externo de 0.5 pulgadas cuadradas. El potencial en un punto en la sección transversal del cable se describe mediante la ecuación de Laplace. Suponga que el conductor interno se mantiene en 0 volts y el conductor externo se mantiene en 110 volts. Aproximar el potencial entre los dos conductores requiere resolver el siguiente sistema lineal:

\$

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

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \\ w_{10} \\ w_{11} \\ w_{12} \end{bmatrix}$$

=

$$\begin{bmatrix} 220 \\ 110 \\ 110 \\ 220 \\ 110 \\ 110 \\ 110 \\ 110 \\ 220 \\ 110 \\ 110 \\ 220 \end{bmatrix}$$

a) ¿La matriz es estrictamente diagonalmente dominante?

b) Resuelva el sistema lineal usando el método de Jacobi con  $\mathbf{x}^{(0)} = \mathbf{0}$  y  $TOL = 10^{-2}$

c) Repita la parte (b) mediante el método de Gauss-Seidel.

```
In [8]: import numpy as np

# Parte (a): Verificar si la matriz es estrictamente diagonal dominante
def es_diagonal_dominante(A):
    for i in range(A.shape[0]):
        suma_fila = sum(abs(A[i, j])) for j in range(A.shape[1]) if i != j)
        if abs(A[i, i]) <= suma_fila:
            return False
    return True

A = np.array([
    [4, -1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0],
    [-1, 4, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, -1, 4, -1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, -1, 4, 0, -1, 0, 0, 0, 0, 0, 0],
    [-1, 0, 0, 0, 4, 0, -1, 0, 0, 0, 0, 0],
    [0, 0, 0, -1, 0, 4, 0, -1, 0, 0, 0, 0],
    [0, 0, 0, 0, -1, 0, 4, 0, -1, 0, 0, 0],
    [0, 0, 0, 0, 0, -1, 0, 4, 0, 0, 0, -1],
    [0, 0, 0, 0, 0, 0, -1, 0, 4, -1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, -1, 4, -1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, -1, 4, -1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 4, -1]
])

b = np.array([220, 110, 110, 220, 110, 110, 110, 220, 110, 110, 220], dtype
print("¿La matriz es diagonalmente dominante?", es_diagonal_dominante(A))
```

¿La matriz es diagonalmente dominante? True

```
In [9]: # Parte (b): Método de Jacobi
def gauss_jacobi(A, b, x0, tol=1e-2, max_iterations=100):
    n = len(A)
    x = x0.copy()
    results = [x.copy()]
```

```

for k in range(max_iterations):
    x_new = np.zeros_like(x)
    for i in range(n):
        sum_ = sum(A[i][j] * x[j] for j in range(n) if j != i)
        x_new[i] = (b[i] - sum_) / A[i][i]
    results.append(x_new.copy())
    if np.linalg.norm(x_new - x, ord=np.inf) < tol:
        print(f"Convergencia alcanzada en {k+1} iteraciones.")
        break
    x = x_new
return results, k+1

x0 = np.zeros(12)
results_jacobi, iterations_jacobi = gauss_jacobi(A, b, x0)
print("\nResultados del método de Jacobi:")
for i, res in enumerate(results_jacobi):
    print(f"Iteración {i}: {res}")

```

Convergencia alcanzada en 13 iteraciones.

Resultados del método de Jacobi:

```

Iteración 0: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Iteración 1: [55. 27.5 27.5 55. 27.5 27.5 27.5 27.5 55. 27.5 27.5 55. ]
Iteración 2: [68.75 48.125 48.125 68.75 48.125 48.125 48.125 48.125 68.75 48.1
25
48.125 68.75 ]
Iteración 3: [79.0625 56.71875 56.71875 79.0625 56.71875 56.71875 56.71875 56.7
1875
79.0625 56.71875 56.71875 79.0625 ]
Iteración 4: [83.359375 61.4453125 61.4453125 83.359375 61.4453125 61.4453125
61.4453125 61.4453125 83.359375 61.4453125 61.4453125 83.359375 ]
Iteración 5: [85.72265625 63.70117188 63.70117188 85.72265625 63.70117188 63.7011
7188
63.70117188 63.70117188 85.72265625 63.70117188 63.70117188 85.72265625]
Iteración 6: [86.85058594 64.85595703 64.85595703 86.85058594 64.85595703 64.8559
5703
64.85595703 64.85595703 86.85058594 64.85595703 64.85595703 86.85058594]
Iteración 7: [87.42797852 65.42663574 65.42663574 87.42797852 65.42663574 65.4266
3574
65.42663574 65.42663574 87.42797852 65.42663574 65.42663574 87.42797852]
Iteración 8: [87.71331787 65.71365356 65.71365356 87.71331787 65.71365356 65.7136
5356
65.71365356 65.71365356 87.71331787 65.71365356 65.71365356 87.71331787]
Iteración 9: [87.85682678 65.85674286 65.85674286 87.85682678 65.85674286 65.8567428
6
65.85674286 65.85674286 87.85682678 65.85674286 65.85674286 87.85682678]
Iteración 10: [87.92837143 65.92839241 65.92839241 87.92837143 65.92839241 65.92839241
65.92839241
65.92839241 65.92839241 87.92837143 65.92839241 65.92839241 87.92837143]
Iteración 11: [87.96419621 65.96419096 65.96419096 87.96419621 65.96419096 65.96419096
65.96419096
65.96419096 65.96419096 87.96419621 65.96419096 65.96419096 87.96419621]
Iteración 12: [87.98209548 65.98209679 65.98209679 87.98209548 65.98209679 65.98209679
65.98209679
65.98209679 65.98209679 87.98209548 65.98209679 65.98209679 87.98209548]
Iteración 13: [87.9910484 65.99104807 65.99104807 87.9910484 65.99104807 65.99104807 65.991
04807
65.99104807 65.99104807 87.9910484 65.99104807 65.99104807 87.9910484 ]

```

```
In [10]: # Parte (c): Método de Gauss-Seidel
def gauss_seidel(A, b, x0, tol=1e-2, max_iterations=100):
    n = len(A)
    x = x0.copy()
    results = [x.copy()]
    for k in range(max_iterations):
        x_new = x.copy()
        for i in range(n):
            sum1 = sum(A[i][j] * x_new[j] for j in range(i))
            sum2 = sum(A[i][j] * x[j] for j in range(i + 1, n))
            x_new[i] = (b[i] - sum1 - sum2) / A[i][i]
        results.append(x_new.copy())
        if np.linalg.norm(x_new - x, ord=np.inf) < tol:
            print(f"Convergencia alcanzada en {k+1} iteraciones.")
            break
        x = x_new
    return results, k+1

results_seidel, iterations_seidel = gauss_seidel(A, b, x0)
print("\nResultados del método de Gauss-Seidel:")
for i, res in enumerate(results_seidel):
    print(f"Iteración {i}: {res}")
```

Convergencia alcanzada en 10 iteraciones.

Resultados del método de Gauss-Seidel:

Iteración 0: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Iteración 1: [55. 41.25 37.8125 64.453125 41.25 43.6132  
8125 37.8125 38.40332031 64.453125 43.61328125 38.40332031 75.50415039]

Iteración 2: [75.625 55.859375 57.578125 80.29785156 55.859375 57.1752  
9297 57.578125 60.66986084 80.29785156 57.17529297 60.66986084 84.46128845]

Iteración 3: [82.9296875 62.62695312 63.23120117 85.10162354 62.62695312 63.9428  
7109 63.23120117 64.60103989 85.10162354 63.94287109 64.60103989 87.13597775]

Iteración 4: [86.31347656 64.88616943 64.99694824 87.23495483 64.88616943 65.4589  
9868 64.99694824 65.64874411 87.23495483 65.45899868 65.64874411 87.7769357 ]

Iteración 5: [87.44308472 65.61000824 65.71124077 87.79255986 65.61000824 65.8603  
2599 65.71124077 65.90931542 87.79255986 65.86032599 65.90931542 87.94241035]

Iteración 6: [87.80500412 65.87906122 65.91790527 87.94455782 65.87906122 65.9634  
6831 65.91790527 65.97646967 87.94455782 65.96346831 65.97646967 87.98498449]

Iteración 7: [87.93953061 65.96435897 65.9772292 87.98517438 65.96435897 65.9904  
1101 65.9772292 65.99384888 87.98517438 65.99041101 65.99384888 87.99606497]

Iteración 8: [87.98217949 65.98985217 65.99375664 87.99604191 65.98985217 65.9974  
727 65.99375664 65.99838442 87.99604191 65.9974727 65.99838442 87.99896428]

Iteración 9: [87.99492609 65.99717068 65.99830315 87.99894396 65.99717068 65.9993  
3209 65.99830315 65.99957409 87.99894396 65.99933209 65.99957409 87.99972655]

Iteración 10: [87.99858534 65.99922212 65.99954152 87.9997184 65.99922212 65.99  
82312 65.99954152 65.99988742 87.9997184 65.99982312 65.99988742 87.99992764]