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

Richard Tipantiza

Tabla de Contenidos

```python
%load_ext autoreload
%autoreload 2
from src import eliminacion_gaussiana, gauss_jacobi, gauss_seidel
import numpy as np
```

```
[01-28 23:46:48][INFO][DELL] 2026-01-28 23:46:48.862541
[01-28 23:46:50][INFO][DELL] 2026-01-28 23:46:50.857008
```

CONJUNTO DE EJERCICIOS

1. Encuentre las primeras dos iteraciones del método de Jacobi para los siguientes sistemas lineales, por medio de $\mathbf{x}^{(0)} = \mathbf{0}$:

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.

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

c.

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

d.

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

```python
#definir la matriz A y el vector b
A = np.array([[3,-1, 1], [3, 6, 2], [3, 3, 7]])
b = np.array([1, 0, 4])

#definir n
n = len(b)

gauss_jacobi(A=A, b=b, x=[0] * n, tol=1e-5, max_iter=10)  # type: ignore
```

```
[01-28 23:52:58][INFO][DELL] i= 0 x: [[0. 0. 0.]]
[01-28 23:52:58][INFO][DELL] i= 10 x: [[ 0.03507839 -0.23692617  0.65780145]]
```

```
array([[ 0.03507839],
       [-0.23692617],
       [ 0.65780145]])
```

```
A1 = np.array([[10, -1, 0], [-1, 10, -2], [0, -2, 10]])
b1 = np.array([9, 7, 6])
n1 = len(b1)
gauss_jacobi(A=A1, b=b1, x=[0] * n1, tol=1e-5, max_iter=10)  # type: ignore
```

```
[01-28 23:57:25][INFO][DELL] i= 0 x: [[0. 0. 0.]]
```

```
array([[0.99578888],
       [0.95789313],
       [0.79157775]])
```

```
A2 = np.array([[10, 5, 0, 0], [5, 10, -4, 0], [0, -4, 8 , -1], [0, 0, -1, 5]])
b2 = np.array([6, 25, -11, -11])
n2 = len(b2)
gauss_seidel(A=A2, b=b2, x=[0] * n2, tol=1e-5, max_iter=10)  # type: ignore
```

```
[01-28 23:58:26][INFO][DELL] i= 0 x: [[0. 0. 0. 0.]]
[01-28 23:58:26][INFO][DELL] i= 1 x: [[ 0.6    2.2   -0.275 -2.255]]
[01-28 23:58:26][INFO][DELL] i= 2 x: [[-0.5       2.64     -0.336875 -2.267375]]
[01-28 23:58:26][INFO][DELL] i= 3 x: [[-0.72        2.72525    -0.29579687 -2.25915938]]
[01-28 23:58:26][INFO][DELL] i= 4 x: [[-0.762625    2.76299375 -0.27589805 -2.25517961]]
[01-28 23:58:26][INFO][DELL] i= 5 x: [[-0.78149687  2.78038922 -0.26670284 -2.25334057]]
[01-28 23:58:26][INFO][DELL] i= 6 x: [[-0.79019461  2.78841617 -0.26245949 -2.2524919 ]]
[01-28 23:58:26][INFO][DELL] i= 7 x: [[-0.79420808  2.79212025 -0.26050136 -2.25210027]]
[01-28 23:58:26][INFO][DELL] i= 8 x: [[-0.79606012  2.79382952 -0.25959778 -2.25191956]]
[01-28 23:58:26][INFO][DELL] i= 9 x: [[-0.79691476  2.79461827 -0.25918081 -2.25183616]]
[01-28 23:58:26][INFO][DELL] i= 10 x: [[-0.79730913  2.79498224 -0.2589884  -2.25179768]]
```

```
array([[-0.79730913],
       [ 2.79498224],
       [-0.2589884 ],
       [-2.25179768]])
```

```
A3 = np.array([[4 ,1, 1, 0, 1], [-1, -3, 1, 1, 0], [2, 1, 5, -1, -1], [-1, -1, -1, 4, 0], [0
b3 = np.array([6, 6, 6, 6, 6])
n3 = len(b3)
gauss_seidel(A=A3, b=b3, x=[0] * n3, tol=1e-5, max_iter=10)  # type: ignore
```

```
[01-29 00:01:10][INFO][DELL] i= 0 x: [[0. 0. 0. 0. 0.]]
[01-29 00:01:10][INFO][DELL] i= 1 x: [[ 1.5     -2.5      1.1      1.525    2.64375]]
[01-29 00:01:10][INFO][DELL] i= 2 x: [[ 1.1890625  -1.52135417  1.86239583  1.88252604  2.255
[01-29 00:01:10][INFO][DELL] i= 3 x: [[ 0.85082845 -1.03530219  1.89436317  1.92747236  2.009
[01-29 00:01:10][INFO][DELL] i= 4 x: [[ 0.7828913  -0.98701859  1.87161643  1.91687229  1.982
[01-29 00:01:10][INFO][DELL] i= 5 x: [[ 0.78330171 -0.998271    1.86614704  1.91279444  1.987
[01-29 00:01:10][INFO][DELL] i= 6 x: [[ 0.78616258 -1.00240703  1.86606999  1.91245638  1.989
[01-29 00:01:10][INFO][DELL] i= 7 x: [[ 0.78668253 -1.00271872  1.86628339  1.9125618   1.989
[01-29 00:01:10][INFO][DELL] i= 8 x: [[ 0.78666139 -1.0026054   1.86632683  1.91259571  1.989
[01-29 00:01:10][INFO][DELL] i= 9 x: [[ 0.78663577 -1.00257108  1.86632614  1.91259771  1.989
```

```
array([[ 0.78663182],
       [-1.00256932],
       [ 1.86632421],
       [ 1.91259668],
       [ 1.98971654]])
```

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

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

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

5. El sistema lineal

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

```
gauss_seidel(A=A, b=b, x=[8] * n, tol=1e-5, max_iter=10)  # type: ignore
```

```
[01-29 00:05:10][INFO][DELL] i= 0 x: [[8. 8. 8.]]
[01-29 00:05:10][INFO][DELL] i= 1 x: [[ 0.33333333 -2.83333333  1.64285714]]
[01-29 00:05:10][INFO][DELL] i= 2 x: [[-1.15873016  0.03174603  1.05442177]]
[01-29 00:05:10][INFO][DELL] i= 3 x: [[-0.00755858 -0.34769463  0.72367995]]
[01-29 00:05:10][INFO][DELL] i= 4 x: [[-0.02379153 -0.22933089  0.67990961]]
[01-29 00:05:10][INFO][DELL] i= 5 x: [[ 0.03025317 -0.24176312  0.66207569]]
[01-29 00:05:10][INFO][DELL] i= 6 x: [[ 0.03205373 -0.23671876  0.65914216]]
```

```
[01-29 00:05:10][INFO][DELL] i= 7 x: [[ 0.03471303 -0.23707057  0.65815323]]
[01-29 00:05:10][INFO][DELL] i= 8 x: [[ 0.0349254  -0.23684711  0.65796645]]
[01-29 00:05:10][INFO][DELL] i= 9 x: [[ 0.03506215 -0.23685322  0.65791046]]
[01-29 00:05:10][INFO][DELL] i= 10 x: [[ 0.03507877 -0.23684287  0.6578989 ]]
```

```
array([[ 0.03507877],
       [-0.23684287],
       [ 0.6578989 ]])
```

```
gauss_jacobi(A=A, b=b, x=[0] * n, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:06:42][INFO][DELL] i= 0 x: [[0. 0. 0.]]
```

```
array([[ 0.03507839],
       [-0.23692617],
       [ 0.65780145]])
```

```
gauss_seidel(A=A, b=b, x=[8] * n, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:07:07][INFO][DELL] i= 0 x: [[8. 8. 8.]]
[01-29 00:07:07][INFO][DELL] i= 1 x: [[ 0.33333333 -2.83333333  1.64285714]]
[01-29 00:07:07][INFO][DELL] i= 2 x: [[-1.15873016  0.03174603  1.05442177]]
[01-29 00:07:07][INFO][DELL] i= 3 x: [[-0.00755858 -0.34769463  0.72367995]]
[01-29 00:07:07][INFO][DELL] i= 4 x: [[-0.02379153 -0.22933089  0.67990961]]
[01-29 00:07:07][INFO][DELL] i= 5 x: [[ 0.03025317 -0.24176312  0.66207569]]
[01-29 00:07:07][INFO][DELL] i= 6 x: [[ 0.03205373 -0.23671876  0.65914216]]
[01-29 00:07:07][INFO][DELL] i= 7 x: [[ 0.03471303 -0.23707057  0.65815323]]
```

```
array([[ 0.0349254 ],
       [-0.23684711],
       [ 0.65796645]])
```

```
gauss_seidel(A=A1, b=b1, x=[8] * n1, tol=1e-5, max_iter=10)  # type: ignore
```

```
[01-29 00:05:22][INFO][DELL] i= 0 x: [[8. 8. 8.]]
[01-29 00:05:22][INFO][DELL] i= 1 x: [[1.7   2.47  1.094]]
[01-29 00:05:22][INFO][DELL] i= 2 x: [[1.147  1.0335 0.8067]]
[01-29 00:05:22][INFO][DELL] i= 3 x: [[1.00335  0.961675 0.792335]]
[01-29 00:05:22][INFO][DELL] i= 4 x: [[0.9961675  0.95808375 0.79161675]]
[01-29 00:05:22][INFO][DELL] i= 5 x: [[0.99580837 0.95790419 0.79158084]]
[01-29 00:05:22][INFO][DELL] i= 6 x: [[0.99579042 0.95789521 0.79157904]]
```

```
array([[0.99578952],
       [0.95789476],
       [0.79157895]])
```

```python
gauss_jacobi(A=A1, b=b1, x=[0] * n1, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:08:29][INFO][DELL] i= 0 x: [[0. 0. 0.]]
```

```
array([[0.995725],
       [0.957775],
       [0.79145 ]])
```

```python
gauss_seidel(A=A1, b=b1, x=[8] * n1, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:08:35][INFO][DELL] i= 0 x: [[8. 8. 8.]]
[01-29 00:08:35][INFO][DELL] i= 1 x: [[1.7   2.47  1.094]]
[01-29 00:08:35][INFO][DELL] i= 2 x: [[1.147  1.0335 0.8067]]
[01-29 00:08:35][INFO][DELL] i= 3 x: [[1.00335  0.961675 0.792335]]
[01-29 00:08:35][INFO][DELL] i= 4 x: [[0.9961675  0.95808375 0.79161675]]
```

```
array([[0.99580837],
       [0.95790419],
       [0.79158084]])
```

```python
gauss_seidel(A=A2, b=b2, x=[8] * n2, tol=1e-5, max_iter=10)  # type: ignore
```

```
[01-29 00:05:34][INFO][DELL] i= 0 x: [[8. 8. 8. 8.]]
[01-29 00:05:34][INFO][DELL] i= 1 x: [[-3.4    7.4    3.325 -1.535]]
[01-29 00:05:34][INFO][DELL] i= 2 x: [[-3.1       5.38      1.123125 -1.975375]]
[01-29 00:05:34][INFO][DELL] i= 3 x: [[-2.09        3.99425     0.37520312 -2.12495937]]
[01-29 00:05:34][INFO][DELL] i= 4 x: [[-1.397125    3.34864375  0.03370195 -2.19325961]]
[01-29 00:05:34][INFO][DELL] i= 5 x: [[-1.07432187  3.05064172 -0.12383659 -2.22476732]]
[01-29 00:05:34][INFO][DELL] i= 6 x: [[-0.92532086  2.91312579 -0.19653302 -2.2393066 ]]
[01-29 00:05:34][INFO][DELL] i= 7 x: [[-0.8565629   2.84966824 -0.2300792  -2.24601584]]
[01-29 00:05:34][INFO][DELL] i= 8 x: [[-0.82483412  2.82038538 -0.24555929 -2.24911186]]
[01-29 00:05:34][INFO][DELL] i= 9 x: [[-0.81019269  2.80687263 -0.25270267 -2.25054053]]
[01-29 00:05:34][INFO][DELL] i= 10 x: [[-0.80343631  2.80063709 -0.25599902 -2.2511998 ]]
```

```
array([[-0.80343631],
       [ 2.80063709],
       [-0.25599902],
       [-2.2511998 ]])
```

```
gauss_jacobi(A=A2, b=b2, x=[0] * n2, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:08:41][INFO][DELL] i= 0 x: [[0. 0. 0. 0.]]
[01-29 00:08:41][INFO][DELL] i= 10 x: [[-0.79125437  2.74356006 -0.26558238 -2.26361792]]
```

```
array([[-0.79125437],
       [ 2.74356006],
       [-0.26558238],
       [-2.26361792]])
```

```
gauss_seidel(A=A2, b=b2, x=[8] * n2, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:08:46][INFO][DELL] i= 0 x: [[8. 8. 8. 8.]]
[01-29 00:08:46][INFO][DELL] i= 1 x: [[-3.4    7.4    3.325 -1.535]]
[01-29 00:08:46][INFO][DELL] i= 2 x: [[-3.1       5.38      1.123125 -1.975375]]
[01-29 00:08:46][INFO][DELL] i= 3 x: [[-2.09        3.99425     0.37520312 -2.12495937]]
[01-29 00:08:46][INFO][DELL] i= 4 x: [[-1.397125    3.34864375  0.03370195 -2.19325961]]
[01-29 00:08:46][INFO][DELL] i= 5 x: [[-1.07432187  3.05064172 -0.12383659 -2.22476732]]
[01-29 00:08:46][INFO][DELL] i= 6 x: [[-0.92532086  2.91312579 -0.19653302 -2.2393066 ]]
[01-29 00:08:46][INFO][DELL] i= 7 x: [[-0.8565629   2.84966824 -0.2300792  -2.24601584]]
[01-29 00:08:46][INFO][DELL] i= 8 x: [[-0.82483412  2.82038538 -0.24555929 -2.24911186]]
[01-29 00:08:46][INFO][DELL] i= 9 x: [[-0.81019269  2.80687263 -0.25270267 -2.25054053]]
[01-29 00:08:46][INFO][DELL] i= 10 x: [[-0.80343631  2.80063709 -0.25599902 -2.2511998 ]]
```

```
array([[-0.80343631],
       [ 2.80063709],
       [-0.25599902],
       [-2.2511998 ]])
```

```
gauss_seidel(A=A3, b=b3, x=[8] * n3, tol=1e-5, max_iter=10)  # type: ignore
```

```
[01-29 00:05:37][INFO][DELL] i= 0 x: [[8. 8. 8. 8. 8.]]
[01-29 00:05:37][INFO][DELL] i= 1 x: [[-4.5         4.83333333  5.23333333  2.89166667 -0.33
[01-29 00:05:37][INFO][DELL] i= 2 x: [[-0.93385417  1.01961806  1.88170139  1.99186632  0.96
[01-29 00:05:37][INFO][DELL] i= 3 x: [[ 0.5340077  -0.88681333  1.7546628   1.85046429  1.91
[01-29 00:05:37][INFO][DELL] i= 4 x: [[ 0.80317356 -1.06601549  1.84591779  1.89576897  2.02
[01-29 00:05:37][INFO][DELL] i= 5 x: [[ 0.79988819 -1.01940048  1.8671876   1.91191883  1.99
[01-29 00:05:37][INFO][DELL] i= 6 x: [[ 0.78842386 -1.00310581  1.86733887  1.91316423  1.99
[01-29 00:05:37][INFO][DELL] i= 7 x: [[ 0.78641759 -1.0019715   1.86647942  1.91273138  1.98
[01-29 00:05:37][INFO][DELL] i= 8 x: [[ 0.78651733 -1.00243551  1.866311    1.9125982   1.98
[01-29 00:05:37][INFO][DELL] i= 9 x: [[ 0.78661964 -1.00257015  1.866315    1.91259112  1.98
[01-29 00:05:37][INFO][DELL] i= 10 x: [[ 0.78663477 -1.00257622  1.86632277  1.91259533  1.98
```

```
array([[ 0.78663477],
       [-1.00257622],
       [ 1.86632277],
       [ 1.91259533],
       [ 1.98971997]])
```

```
gauss_jacobi(A=A3, b=b3, x=[0] * n3, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:09:03][INFO][DELL] i= 0 x: [[0. 0. 0. 0. 0.]]
[01-29 00:09:03][INFO][DELL] i= 10 x: [[ 0.78718101 -1.00174151  1.8658388   1.91274157  1.98
```

```
array([[ 0.78718101],
       [-1.00174151],
       [ 1.8658388 ],
       [ 1.91274157],
       [ 1.98672138]])
```

```
gauss_seidel(A=A3, b=b3, x=[8] * n3, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:09:17][INFO][DELL] i= 0 x: [[8. 8. 8. 8. 8.]]
[01-29 00:09:17][INFO][DELL] i= 1 x: [[-4.5         4.83333333  5.23333333  2.89166667 -0.331
[01-29 00:09:17][INFO][DELL] i= 2 x: [[-0.93385417  1.01961806  1.88170139  1.99186632  0.962
[01-29 00:09:17][INFO][DELL] i= 3 x: [[ 0.5340077  -0.88681333  1.7546628   1.85046429  1.919
[01-29 00:09:17][INFO][DELL] i= 4 x: [[ 0.80317356 -1.06601549  1.84591779  1.89576897  2.020
[01-29 00:09:17][INFO][DELL] i= 5 x: [[ 0.79988819 -1.01940048  1.8671876   1.91191883  1.998
[01-29 00:09:17][INFO][DELL] i= 6 x: [[ 0.78842386 -1.00310581  1.86733887  1.91316423  1.990
[01-29 00:09:17][INFO][DELL] i= 7 x: [[ 0.78641759 -1.0019715   1.86647942  1.91273138  1.989
```

```
array([[ 0.78651733],
       [-1.00243551],
       [ 1.866311  ],
       [ 1.9125982 ],
       [ 1.98964595]])
```

A) Muestre que el método de Jacobi con x(0) = 0 falla al proporcionar una buena aproximación después de 25 iteraciones.

```
A4 = np.array([[2, -1, 1], [2, 2, 2], [-1, -1, 2]])
b4 = np.array([-1, 4, 5])
n4 = len(b4)
gauss_jacobi(A=A4, b=b4, x=[0] * n4, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:11:55][INFO][DELL] i= 0 x: [[0. 0. 0.]]
[01-29 00:11:55][INFO][DELL] i= 10 x: [[-0.06933594 -1.28125     4.57128906]]
```

```
array([[-0.06933594],
       [-1.28125   ],
       [ 4.57128906]])
```

B) Utilice el método de Gauss-Siedel con x(0) = 0:para aproximar la solución para el sistema lineal dentro de 10^{-5}

```
gauss_seidel(A=A4, b=b4, x=[0] * n4, tol=1e-3, max_iter=10)  # type: ignore
```

```
[01-29 00:12:00][INFO][DELL] i= 0 x: [[0. 0. 0.]]
[01-29 00:12:00][INFO][DELL] i= 1 x: [[-0.5  2.5  3.5]]
[01-29 00:12:00][INFO][DELL] i= 2 x: [[-1.   -0.5   1.75]]
[01-29 00:12:00][INFO][DELL] i= 3 x: [[-1.625  1.875  2.625]]
[01-29 00:12:00][INFO][DELL] i= 4 x: [[-0.875   0.25    2.1875]]
[01-29 00:12:00][INFO][DELL] i= 5 x: [[-1.46875  1.28125  2.40625]]
[01-29 00:12:00][INFO][DELL] i= 6 x: [[-1.0625    0.65625   2.296875]]
[01-29 00:12:00][INFO][DELL] i= 7 x: [[-1.3203125  1.0234375  2.3515625]]
[01-29 00:12:00][INFO][DELL] i= 8 x: [[-1.1640625   0.8125      2.32421875]]
[01-29 00:12:00][INFO][DELL] i= 9 x: [[-1.25585938  0.93164062  2.33789062]]
[01-29 00:12:00][INFO][DELL] i= 10 x: [[-1.203125    0.86523438  2.33105469]]
```

```
array([[-1.203125  ],
       [ 0.86523438],
       [ 2.33105469]])
```