

Taller1B_LeninAmangandi

December 25, 2025

1 Escuela Politécnica Nacional

1.1 Métodos Numéricos

Nombre: Lenin Amangandi

Tema: Eliminación gaussiana vs Gauss-Jordan

[Link al repositorio Taller N6](#)

1. Resuelva el bug encontrado en el código

- El error de precisión suele ocurrir en operaciones como divisiones y multiplicaciones, especialmente cuando los valores que estamos usando son enteros. Para evitar esto, es importante asegurarnos de que tanto las matrices como los vectores estén definidos como tipo float. De esta manera, todas las operaciones se harán con precisión de punto flotante, lo que evita problemas de redondeo o truncamiento que pueden ocurrir si usamos enteros.

2. Implemente el conteo de sumas y restas para ambos métodos

```
[1]: import matplotlib.pyplot as plt
import numpy as np

# Definimos el rango de valores para n
n = np.arange(1, 20)

# Calculamos las expresiones para Elimination Gaussiana y Gauss-Jordan
gaussiana = (n**3)/3 + (n**2)/2 - (5*n)/6
gauss_jordan = (n**3)/2 - n/2

# Creamos la figura y los ejes
plt.figure(figsize=(8, 6))

# Graficamos las dos funciones
plt.plot(n, gaussiana, label=r'Eliminación gaussiana: $\frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}$', color='blue')
plt.plot(n, gauss_jordan, label=r'Gauss-Jordan: $\frac{n^3}{2} - \frac{n}{2}$', color='red')

# Añadimos etiquetas y título
plt.xlabel('n')
```

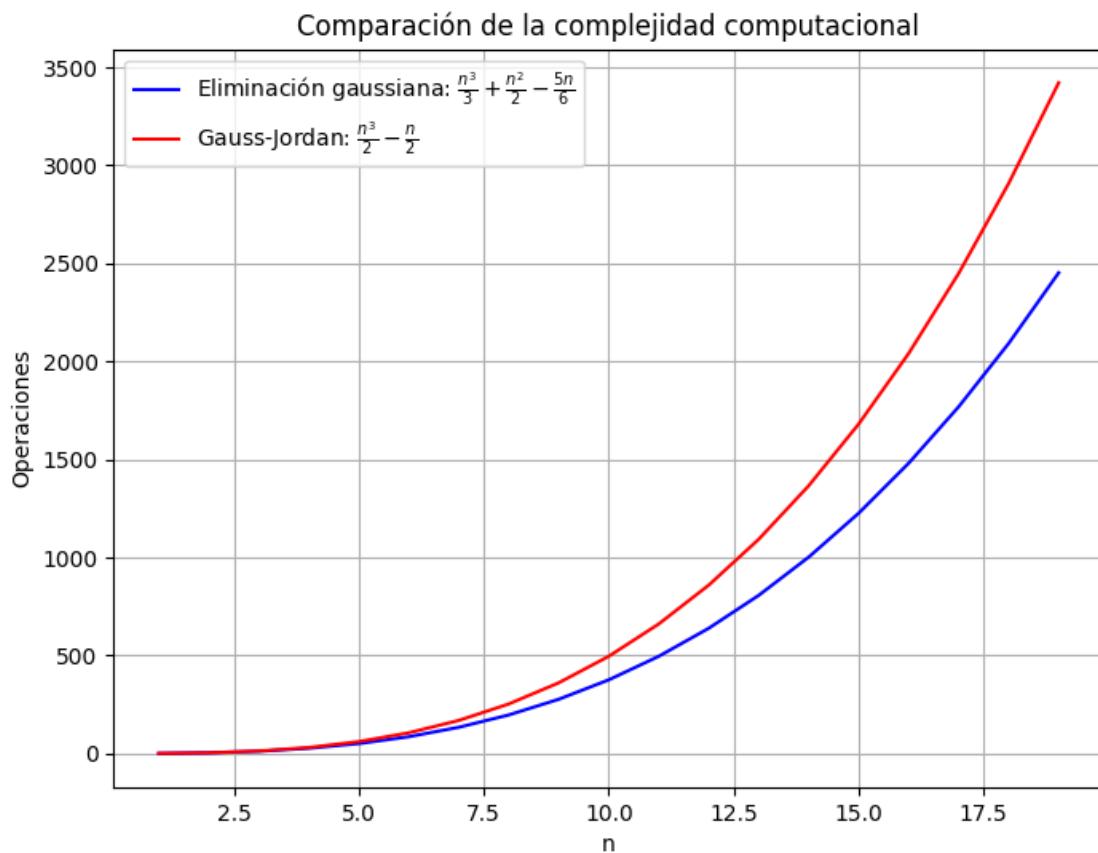
```

plt.ylabel('Operaciones')
plt.title('Comparación de la complejidad computacional')

# Añadimos la leyenda
plt.legend()

# Mostramos la gráfica
plt.grid(True)
plt.show()

```



```

[2]: import matplotlib.pyplot as plt
import numpy as np

# Definimos el rango de valores para n
n = np.arange(1, 20)

# Calculamos las expresiones para multiplicaciones y divisiones
multiplicaciones_divisiones_gaussiana = (n**3)/3 + n**2 - n/3
multiplicaciones_divisiones_gauss_jordan = (n**3)/2 + n**2 - n/2

```

```

# Creamos la figura y los ejes
plt.figure(figsize=(8, 6))

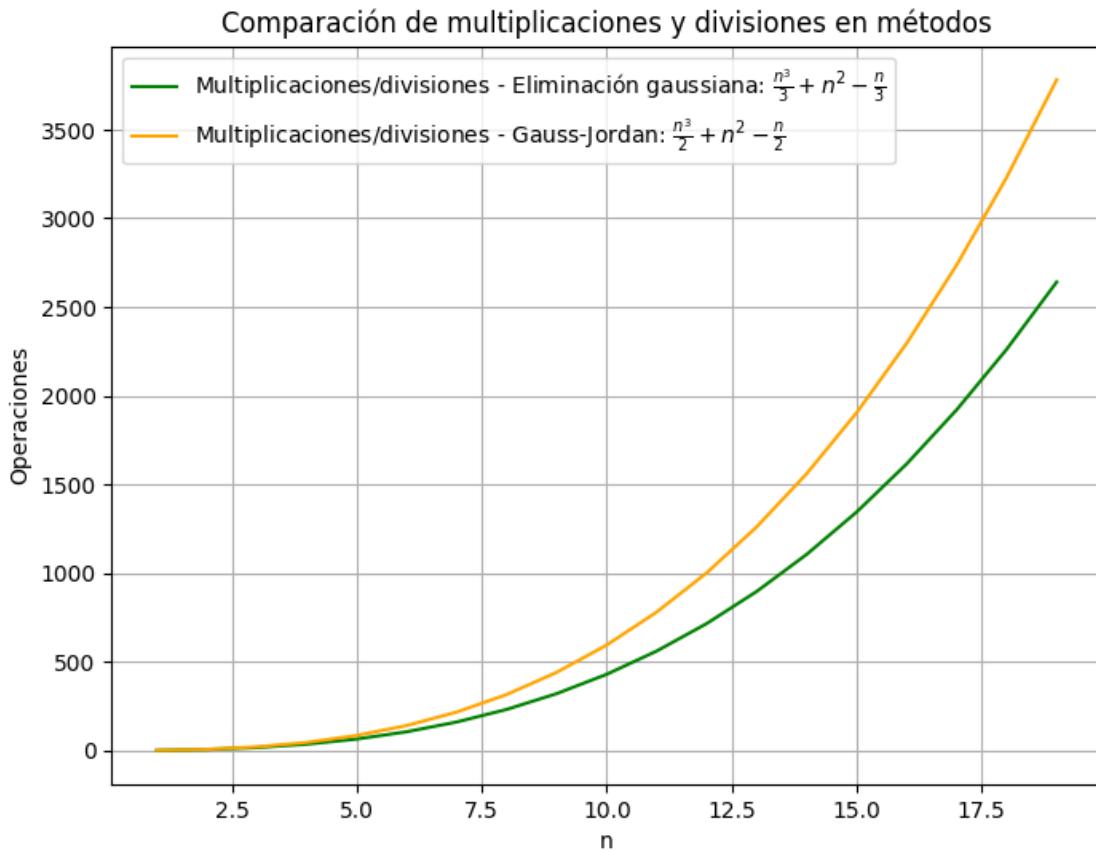
# Graficamos las dos funciones
plt.plot(n, multiplicaciones_divisiones_gaussiana, label=r'Multiplicaciones/
    ↵divisiones - Eliminación gaussiana: $\frac{n^3}{3} + n^2 - \frac{n}{3}$', ↵
    ↵color='green')
plt.plot(n, multiplicaciones_divisiones_gauss_jordan, label=r'Multiplicaciones/
    ↵divisiones - Gauss-Jordan: $\frac{n^3}{2} + n^2 - \frac{n}{2}$', ↵
    ↵color='orange')

# Añadimos etiquetas y título
plt.xlabel('n')
plt.ylabel('Operaciones')
plt.title('Comparación de multiplicaciones y divisiones en métodos')

# Añadimos la leyenda
plt.legend()

# Mostramos la gráfica
plt.grid(True)
plt.show()

```



3. Compare la complejidad computacional del método de eliminación gaussiana vs Gauss-Jordan

El método de **Eliminación Gaussiana** tiene una complejidad de

$$O(n^2)$$

porque solo necesita reducir la matriz a una forma triangular inferior, lo que implica operaciones sobre las filas sin afectar a los elementos por encima de la diagonal. En cambio, **Gauss-Jordan** tiene una complejidad de

$$O(n^3)$$

porque necesita reducir la matriz a la forma escalonada reducida, lo que implica trabajar tanto con la matriz triangular inferior como la superior. Este proceso adicional de eliminación de los elementos sobre la diagonal requiere más operaciones, incrementando la complejidad.

4. Determine el número de intercambios de filas usando el código base

```
[5]: from src import eliminacion_gaussiana
```

```
Ab = [
    [4, 2, -3, 2, -6, 14],
    [2, 1, 3, -2, 2, 5],
```

```
[1, -3, -2, 2, -2, -8],  
[5, 2, -5, 1, 6, 10],  
[2, -2, -2, -2, 2, 6]  
]  
]
```

```
eliminacion_gaussiana(Ab)
```

```
[12-25 21:38:18] [INFO]  
[[ 1 -3 -2 2 -2 -8]  
 [ 0 7 7 -6 6 21]  
 [ 0 14 5 -6 2 46]  
 [ 0 17 5 -9 16 50]  
 [ 0 4 2 -6 6 22]]  
[12-25 21:38:18] [INFO]  
[[ 1 -3 -2 2 -2 -8]  
 [ 0 4 2 -6 6 22]  
 [ 0 0 -2 15 -19 -31]  
 [ 0 0 -3 16 -9 -43]  
 [ 0 0 3 4 -4 -17]]  
[12-25 21:38:18] [INFO]  
[[ 1 -3 -2 2 -2 -8]  
 [ 0 4 2 -6 6 22]  
 [ 0 0 -2 15 -19 -31]  
 [ 0 0 0 -6 19 3]  
 [ 0 0 0 26 -32 -63]]  
[12-25 21:38:18] [INFO]  
[[ 1 -3 -2 2 -2 -8]  
 [ 0 4 2 -6 6 22]  
 [ 0 0 -2 15 -19 -31]  
 [ 0 0 0 -6 19 3]  
 [ 0 0 0 0 50 -50]]
```

```
[5]: array([ 0.58333333,  2.75        , -2.5        , -3.66666667, -1.        ])
```

5. Determine el número de multiplicaciones/divisiones y sumas/restas usando el código base

```
[7]: from src import eliminacion_gaussiana
```

```
Ab = [  
      [3, 2, -1, 2, -4, 10],  
      [2, 1, 3, -2, 2, 6],  
      [1, -1, -2, 3, -2, -4],  
      [4, 2, -3, 1, 6, 8],  
      [2, -2, -1, -2, 3, 5]  
    ]
```

```
eliminacion_gaussiana(Ab)
```

```
[12-25 21:40:22] [INFO]
[[ 1 -1 -2 3 -2 -4]
 [ 0 3 7 -8 6 14]
 [ 0 5 5 -7 2 22]
 [ 0 6 5 -11 14 24]
 [ 0 0 3 -8 7 13]]
[12-25 21:40:22] [INFO]
[[ 1 -1 -2 3 -2 -4]
 [ 0 3 7 -8 6 14]
 [ 0 0 -6 6 -8 -1]
 [ 0 0 -9 5 2 -4]
 [ 0 0 3 -8 7 13]]
[12-25 21:40:22] [INFO]
[[ 1 -1 -2 3 -2 -4]
 [ 0 3 7 -8 6 14]
 [ 0 0 3 -8 7 13]
 [ 0 0 0 -19 23 35]
 [ 0 0 0 -10 6 25]]
[12-25 21:40:22] [INFO]
[[ 1 -1 -2 3 -2 -4]
 [ 0 3 7 -8 6 14]
 [ 0 0 3 -8 7 13]
 [ 0 0 0 -10 6 25]
 [ 0 0 0 0 11 -12]]
[7]: array([ 2.22929293,  2.01414141, -1.53333333, -3.15454545, -1.09090909])
```