Desafío 6

📄 **Sistema de Ecuaciones con Matriz de Hilbert (3×3)**

🧠 **Concepto:**

- Utilizamos una **matriz de Hilbert 3×3**, conocida por su mal condicionamiento, para resolver un sistema de ecuaciones.
- La matriz es sensible a pequeñas variaciones en los datos, afectando la precisión de la solución.

📊 **Resolución:**

1️⃣ **Matriz Original (A):**

$$A = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}$$

2️⃣ **Vector (b):**

$$b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

3️⃣ **Solución:**

- Resolvemos $(A \cdot x = b)$ usando `np.linalg.solve`:

$$x = [3, -24, 30]$$

4️⃣ **Verificaciones:**

- **Determinante de A**:

  $$\det(A) = 3.749 \times 10^{-5}$$
  ⚠️ El determinante es muy pequeño, lo que indica un sistema potencialmente mal condicionado.

- **Número de condición**:

  $$\text{cond}(A) = 748$$
  🔍 Un número elevado sugiere que el sistema es extremadamente sensible a pequeños cambios.

- *Matriz identidad aproximada* $(A * \ inv(A))$:

  $$\text{Identidad} \approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
  Pequeñas desviaciones de la identidad confirman el mal condicionamiento.

Código en Python

▼ Solución del ejercicio utilizando Python

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Matriz de Hilbert 3x3
A = np.array([[1, 1/2, 1/3],
              [1/2, 1/3, 1/4],
              [1/3, 1/4, 1/5]])
```

```
# Vector de resultados (b)
b = np.array([1, 1, 1])

# Resolver el sistema Ax = b
x = np.linalg.solve(A, b)

# Mostrar la solución
print(f"Solución del sistema: x = {x}")

# Calcular el determinante de A
det_A = np.linalg.det(A)
print(f"Determinante de A: {det_A}")

# Calcular el número de condición de A
cond_A = np.linalg.cond(A)
print(f"Número de condición de A: {cond_A}")

# Calcular la matriz identidad aproximada (A * inv(A))
inv_A = np.linalg.inv(A)
identity_approx = np.dot(A, inv_A)
print("Matriz identidad aproximada:\\n", identity_approx)

# Graficar los planos correspondientes a las ecuaciones
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Definir un rango para los valores de x e y
x_vals = np.linspace(-1, 1, 100)
y_vals = np.linspace(-1, 1, 100)
x_vals, y_vals = np.meshgrid(x_vals, y_vals)

# Definir las ecuaciones de los planos (z = f(x, y)) para
cada ecuación
z1 = (b[0] - A[0, 0] * x_vals - A[0, 1] * y_vals) / A[0,
2]
```

```
z2 = (b[1] - A[1, 0] * x_vals - A[1, 1] * y_vals) / A[1,
2]
z3 = (b[2] - A[2, 0] * x_vals - A[2, 1] * y_vals) / A[2,
2]

# Graficar los tres planos
ax.plot_surface(x_vals, y_vals, z1, alpha=0.5, rstride=10
0, cstride=100, color='red')
ax.plot_surface(x_vals, y_vals, z2, alpha=0.5, rstride=10
0, cstride=100, color='blue')
ax.plot_surface(x_vals, y_vals, z3, alpha=0.5, rstride=10
0, cstride=100, color='green')

# Etiquetas de los ejes
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

plt.title("Planos de las Ecuaciones del Sistema de Hilbert
3x3")
plt.show()
```

Explicación adicional:

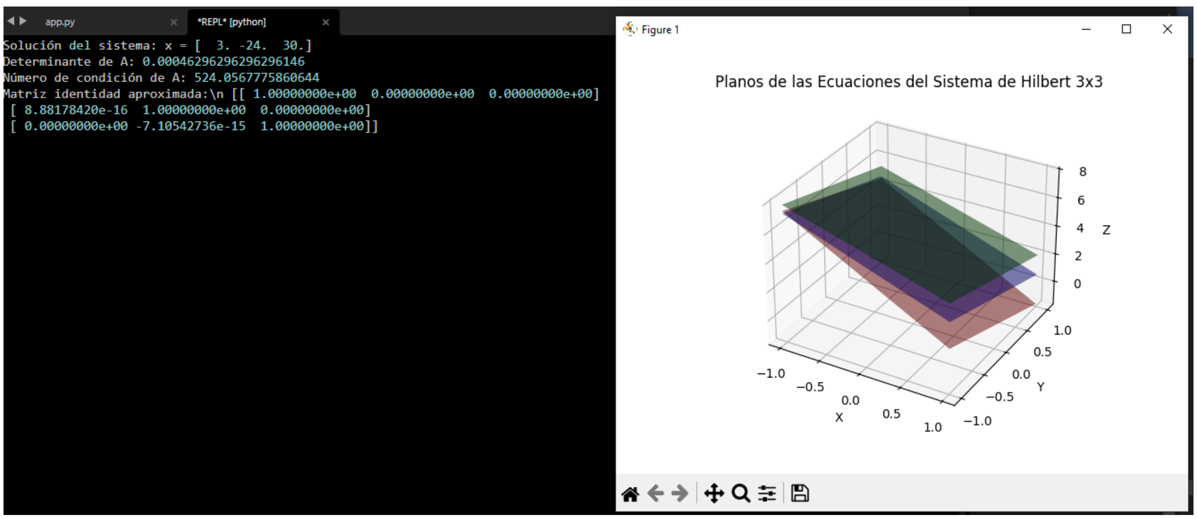
1. **Determinante de A**:
    - El determinante (`det_A`) indica si la matriz es singular o no. Si es cercano a cero, significa que el sistema puede estar mal condicionado.
2. **Número de condición**:
    - El número de condición (`cond_A`) mide la sensibilidad de la solución ante pequeños cambios en los datos. Un número de condición alto sugiere un sistema mal condicionado.
3. **Matriz identidad aproximada**:
    - Multiplicar la matriz original por su inversa debería dar la matriz identidad. Si los valores de la identidad aproximada no son

exactamente los esperados, esto puede confirmar el mal condicionamiento.

📉 Gráfica:

Representamos los tres planos correspondientes a las ecuaciones.

- El punto de intersección de estos planos representa la solución del sistema. 📊 Gráfico en $(\mathbb{R}^3)$.



🔗 Conclusión:

- El sistema, debido al mal condicionamiento, es muy inestable. El uso de la matriz de Hilbert genera números de condición elevados, lo que provoca una gran sensibilidad ante cualquier error de cálculo.
