INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Analizador de Normas Matriciales

Matemáticas avanzadas para la ingeniería

Profesor: David Correa Coyac

Alumnos:

Chávez Torreblanca Angel Alexis
Chico Lampart Ariadna
Sánchez Butrón Eduardo
Segura Ramírez Itziar

4BM2

25/06/2025

Índice

Índice de figuras

Justificación del problema

En primer lugar, comenzamos aclarando el significado de una **norma de matriz**: se define a una **norma de matriz** como una extensión del concepto de norma vectorial aplicada a matrices, una función que mide el tamaño o magnitud de un vector o matriz, permitiendo estudiar su comportamiento, como la tendencia a otra matriz o la amplificación que produce sobre vectores.

Por ejemplo, la norma inducida mide cuánto puede estirar una matriz un vector en el espacio, es decir, la ganancia máxima que la matriz puede proporcionar a un vector unitario.

Existen diferentes tipos de normas matriciales, como la norma 1, máximo de la suma de valores absolutos por columnas, la norma 2, el valor mayor singular de una matriz y la norma de Frobenius. En este proyecto se aborda el *desarrollo de una herramienta que calcule dichas normas de matrices y las compare gráficamente*, además del análisis de sensibilidad.

En contexto de ingeniería, las **normas matriciales** son fundamentales, pues permiten medir el tamaño o magnitud de una matriz, lo cual es muy útil para analizar y controlar sistemas que se representan mediante matrices. Por ejemplo, en el análisis estructural, las normas ayudan a evaluar la estabilidad y el comportamiento de una estructura al estudiar las matrices que describen la rigidez o deformación, lo que facilita predecir cómo reaccionará la estructura ante diferentes cargas.

En el campo del control de sistemas dinámicos, las **normas matriciales** permiten medir cuánto una matriz puede amplificar o atenuar señales o perturbaciones, lo que es clave para diseñar sistemas de control eficientes y estables. Por otro lado, en el ámbito de los cálculos numéricos, estas normas son esenciales para estimar errores y analizar la estabilidad de los métodos utilizados, ya que ayudan a determinar la sensibilidad de un sistema de ecuaciones a pequeñas variaciones o errores en los datos.

Podemos decir que las **normas matriciales** son herramientas clave que permiten a los ingenieros modelar, analizar y garantizar la precisión y estabilidad en una amplia variedad de problemas que involucran matrices.

Con estos beneficios en mente es que se realiza el desarrollo de esta herramienta, pues cada norma ofrece una perspectiva diferente sobre las propiedades de una matriz. Calcularlas y visualizarlas juntas facilita una comprensión más completa y profunda del comportamiento de la matriz en distintos contextos.

Asimismo, el análisis de sensibilidad permite evaluar cómo pequeñas variaciones en los elementos de la matriz afectan las normas, lo que ayuda a identificar posibles puntos críticos o variables que influyen significativamente en la estabilidad o precisión de un sistema. Esta capacidad es vital para anticipar errores, mejorar el diseño y garantizar la robustez de sistemas en áreas como control automático, análisis estructural o procesamiento de señales.

Fundamento matemático aplicado

Una norma de una matriz nos permite cuantificar el *tamaño* o *magnitud* de una matriz. Para que se concidere una norma matricial, debe cumplir tres propiedades fundamentales [1].

- **No negatividad:** La norma de una matriz debe ser mayor o igual a cero, y solo será cero si la matriz en nula.
$$||A|| \geq 0 \quad \text{y} \quad ||A|| = 0 \iff A = 0 \tag{1}$$
- **Homogeneidad:** La norma de una matriz escalada por un factor α es igual al valor absoluto de α multiplicado por la norma de la matriz original.
$$||\alpha A|| = |\alpha| \cdot ||A|| \tag{2}$$
- **Desigualdad triangular:** La norma de la suma de dos matricees es menor o igual a la suma de sus normas individuales.
$$||A + B|| \leq ||A|| + ||B|| \tag{3}$$

Existe una cuarta propiedad que es importante para aplicaciones en análisis númerico, la cual es la **Submultiplicatividad**, estipula que la norma del producto de dos matrices es menor o igual al producto de sus normas.
$$||AB|| \leq ||A|| \cdot ||B|| \tag{4}$$
Existen diversas formas de definir una norma matricial, cada uno con sus propias características y aplicaciones. Las más utilizadas se dividen en dos categorías principales.

Normas Inducidas o Subordinadas

Estas normas se derivan de una norma vectorial. Miden la máxima amplificación que una matriz puede ejercer sobre un vector unitario bajo una norma vectorial específica, por ejemplo:

- **Norma 1:** La norma de columna L_1 vectorial, se calcula como la máxima suma de los valores absolutos de los elementos en cada columna [2].
$$||A||_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}| \tag{5}$$
- **Norma 2:** La norma espectral L_2 es inducida por la norma euclidiana vectorial, es igual a la raíz cuadrada del mayor valor propio de la transpuesta conjugada de A es decir, $A^H A$ [3].
$$||A||_2 = \sqrt{\rho(A^H A)} = \sigma_{\text{máx}}(A) \tag{6}$$

Norma de Frobenius

A diferencia de las normas inducidas, esta norma no se deriva de una norma vectorial específica, aunque comparte similitudes. Se define como la raíz cuadrada de la suma de los cuadrados de los valores absolutos de todos los elementos de la matriz [1].
$$||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\text{traza}(A^H A)} \tag{7}$$

Resultados

Ejemplo 1

Primero, se debe ingresar una matriz, en este caso ingresamos la matriz $A = \begin{bmatrix} 1 & -2 & 0 \\ 4 & 3 & -1 \end{bmatrix}$

Figura 1: Matriz A.

Si se desea agregar una matriz con diferente tamaño, simplemente agregamos las filas y columnas deseadas seleccionando la opción.

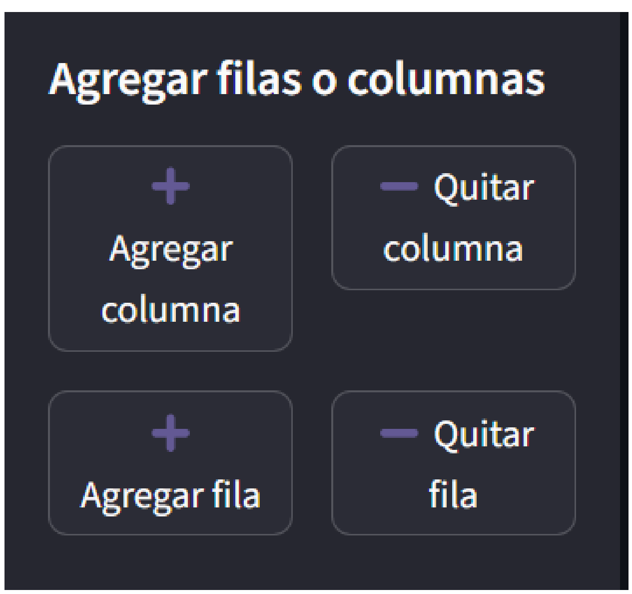


Figura 2: Selección del tamaño de la matriz.

Una vez que se ingresa la matriz, se muestra el resultado de cada norma.

📋 Normas calculadas

	Tipo de norma	Valor
0	Norma L1 (máx suma columnas)	5.0000
1	Norma Frobenius (cuadrática)	5.5678
2	Norma L2 / Espectral (σ máx)	5.1175

Figura 3: Normas calculadas.

Y se muestra una gráfica en donde se compara cada norma.

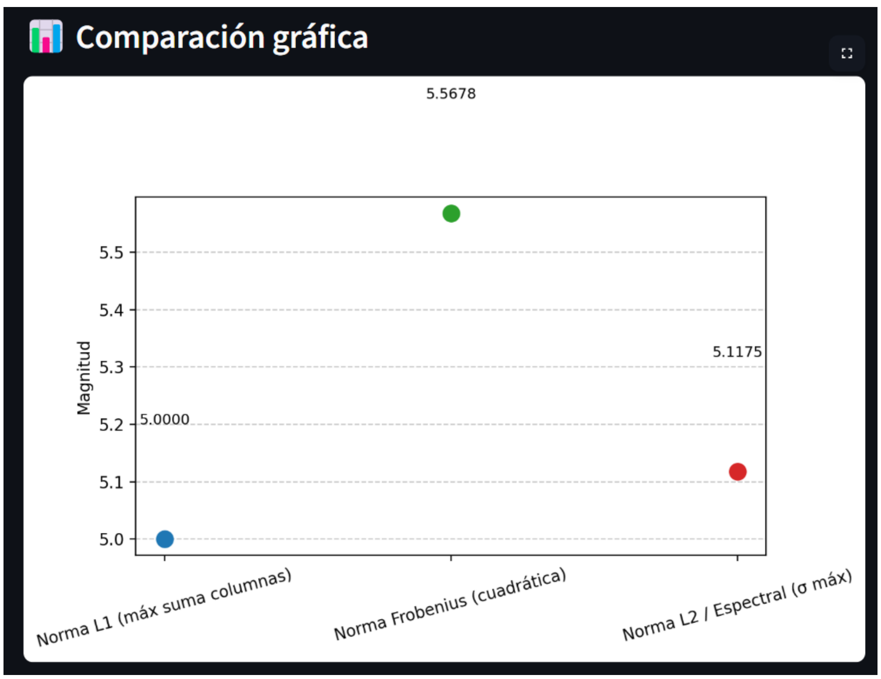


Figura 4: Gráfica.

Al final se muestra el análisis obtenido, en este caso, la matiz A es muy estable númericamente.

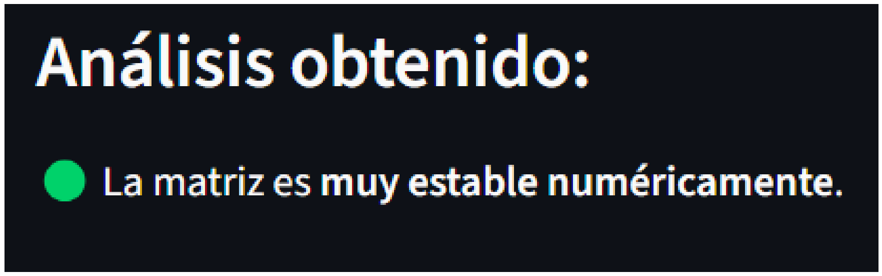
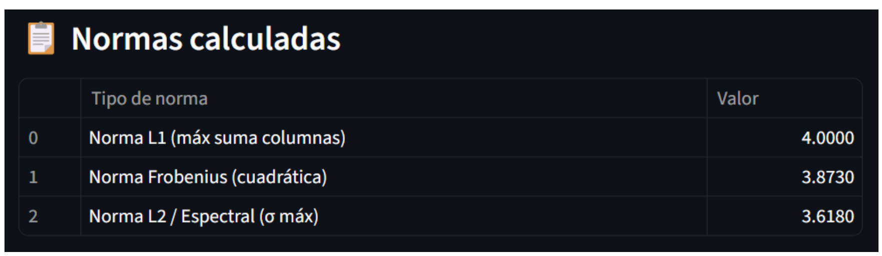


Figura 5: Análisis.

Ejemplo 2

Ahora si ingresmos la matriz B = $\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$ Obtenemos los siguientes resultados:



Normas calculadas

	Tipo de norma	Valor
0	Norma L1 (máx suma columnas)	4.0000
1	Norma Frobenius (cuadrática)	3.8730
2	Norma L2 / Espectral (σ máx)	3.6180

Figura 6: Normas calculadas ejemplo 2.

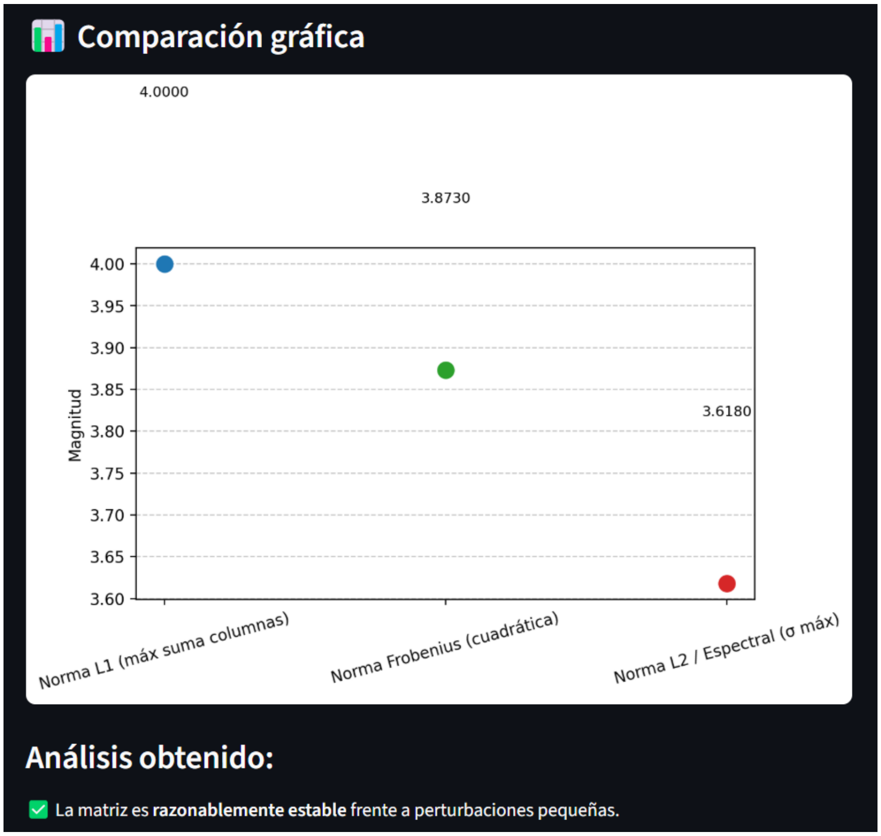


Figura 7: Gráfica y análisis.

Ejemplo 3

Por último ingresmos la matriz C = $\begin{bmatrix} 10 & 9 \\ 11 & 10 \end{bmatrix}$ Obtenemos los siguientes resultados:

Normas calculadas

	Tipo de norma	Valor
0	Norma L1 (máx suma columnas)	21.0000
1	Norma Frobenius (cuadrática)	20.0499
2	Norma L2 / Espectral (σ máx)	20.0499

Figura 8: Normas calculadas ejemplo 3.

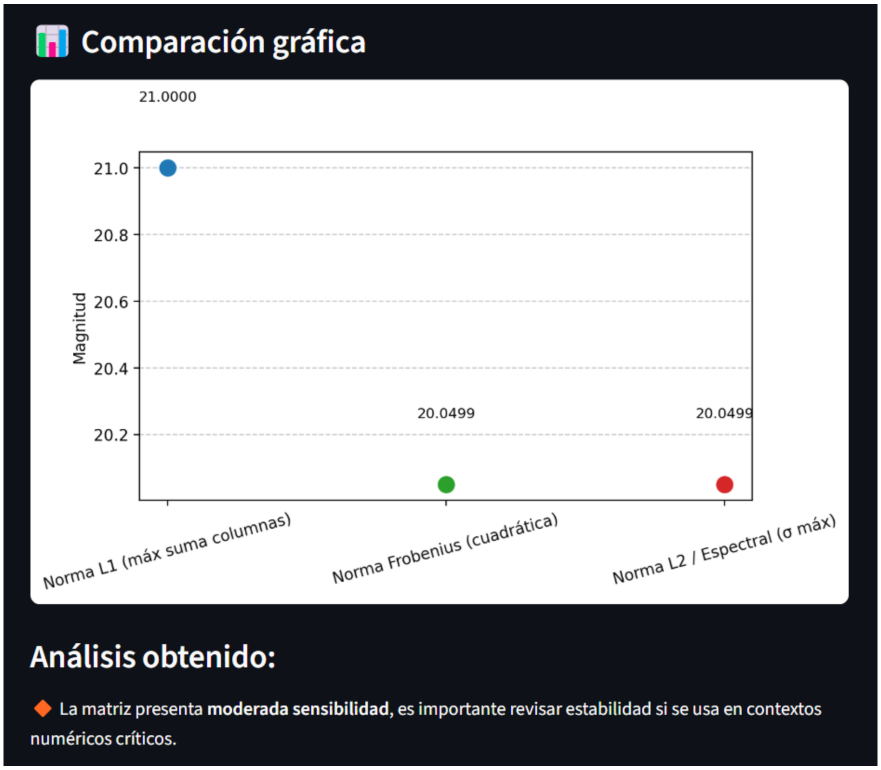


Figura 9: Gráfica y análisis.

Conclusiones

Por sus aplicacines y los beneficios que se pueden obtener de saber cómo aprovechar las **normas matriciales**, consideramos que es fundamental que los ingenieros comprendan las operaciones con matrices, ya que estas herramientas matemáticas permiten modelar y resolver problemas complejos de manera eficiente y organizada.

Esto gracias a que las matrices facilitan la representación de sistemas de ecuaciones lineales. Al dominar las operaciones con matrices, los ingenieros pueden analizar relaciones entre múltiples variables, predecir comportamientos, optimizar recursos y garantizar la estabilidad y precisión en sus soluciones.

Incluso podemos afirmar que el conocimiento de las matrices es esencial para manejar grandes volúmenes de datos y realizar simulaciones, lo que es cada vez más relevante en el contexto tecnológico actual. Por decir algunos ejemplos, en ingeniería civil, las matrices permiten calcular desplazamientos y fuerzas en estructuras complejas; en ingeniería eléctrica, ayudan a determinar corrientes y voltajes en circuitos; y en robótica, facilitan la descripción de movimientos y transformaciones espaciales.

En conclusión, el dominio de las operaciones matriciales dota a los ingenieros de una base sólida para enfrentar desafíos técnicos, desarrollar soluciones innovadoras y tomar decisiones informadas en proyectos multidisciplinarios y de alta complejidad.

Código Fuente

Main

```python
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from matrix_utils import (
    norma_1, norma_infinita, norma_frobenius, norma_2,
    sensibilidad_por_perturbacion
)
from plot_utils import graficar_normas


import matplotlib.pyplot as plt

def graficar_normas_con_puntos(normas_dict):
    nombres = list(normas_dict.keys())
    valores = list(normas_dict.values())

    colores = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728']  # azul, naranja, verde, rojo

    fig, ax = plt.subplots(figsize=(8, 5))

    # Crear el grafico de puntos
    for i, (nombre, valor) in enumerate(zip(nombres, valores)):
        ax.plot(valor, i, 'o', markersize=12, color=colores[i], label=nombre)
        ax.text(valor + 0.1, i, f'{valor:.4f}', va='center', fontsize=10)

    ax.set_yticks(range(len(nombres)))
    ax.set_yticklabels(nombres, fontsize=12)
    ax.set_xlabel('Magnitud de la norma', fontsize=12)
    ax.set_title('Comparacion de Normas Matriciales ', fontsize=14, fontweight='bold')
    ax.grid(axis='x', linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.show()


if __name__ == "__main__":
    A = np.array([[1, -2, 0], [4, -3, 4]])  # Matriz

    print("

    ")
    print("Matriz de entrada A:")
    print(A)
    print("

    ")

    normas = {
        'Norma L1 (max suma columnas)': norma_1(A),
        'Norma    (max suma filas)': norma_infinita(A),
        'Norma L2 / Espectral (   max)': norma_2(A),
```

```
        'Norma Frobenius': norma_frobenius(A),
    }

    print("\n Normas calculadas:")
    print("

    ")
    for nombre, valor in normas.items():
        print(f"{nombre:<35}: {valor:>10.4f}")
    print("

    ")

    graficar_normas_con_puntos(normas)


    delta, cambio = sensibilidad_por_perturbacion(A)
    print("\n Analisis de sensibilidad ante perturbacion:")
    print(f"    Perturbacion (norma Frobenius)      : {delta:.6f}")
    print(f"    Cambio en norma L2 (espectral)      : {cambio:.6f}")
    print("

    ")
```

Matrix utils

```

import numpy as np

def norma_1(A):
    return np.linalg.norm(A, 1)


def norma_frobenius(A):
    return np.linalg.norm(A, 'fro')

def norma_2(A):
    return np.linalg.norm(A, 2)

def valores_propios(A):
    return np.linalg.eigvals(A)

def descomposicion_svd(A):
    return np.linalg.svd(A)

def sensibilidad_por_perturbacion(A, perturbacion=1e-3):
    A_pert = A + perturbacion * np.random.randn(*A.shape)
    delta = np.linalg.norm(A_pert - A, 'fro')
    cambio_norma = abs(norma_2(A_pert) - norma_2(A))
    return delta, cambio_norma
```

Dashboard

```python

import streamlit as st
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from matrix_utils import (
    norma_1, norma_frobenius, norma_2,
    sensibilidad_por_perturbacion
)

# Configuraci n inicial de la app
st.set_page_config(page_title="Analizador de Normas Matriciales", layout="centered")
st.title("    Analizador de Normas Matriciales    ")

# --- Ingreso din mico de matriz ---
st.sidebar.header("Agregar filas o columnas")

if "filas" not in st.session_state:
    st.session_state.filas = 2
if "columnas" not in st.session_state:
    st.session_state.columnas = 3

col1, col2 = st.sidebar.columns(2)
if col1.button("    Agregar columna"):
    st.session_state.columnas += 1
if col2.button("    Quitar columna") and st.session_state.columnas > 1:
    st.session_state.columnas -= 1

col3, col4 = st.sidebar.columns(2)
if col3.button("    Agregar fila"):
    st.session_state.filas += 1
if col4.button("    Quitar fila") and st.session_state.filas > 1:
    st.session_state.filas -= 1

if "df_valores" not in st.session_state:
    st.session_state.df_valores = pd.DataFrame(
        np.zeros((st.session_state.filas, st.session_state.columnas)),
        columns=[f"C{j+1}" for j in range(st.session_state.columnas)]
    )
else:
    df_prev = st.session_state.df_valores
    filas_actual = st.session_state.filas
    columnas_actual = st.session_state.columnas

    if df_prev.shape[0] < filas_actual:
        filas_nuevas = pd.DataFrame(
            np.zeros((filas_actual - df_prev.shape[0], df_prev.shape[1])),
            columns=df_prev.columns
        )
        df_prev = pd.concat([df_prev, filas_nuevas], ignore_index=True)
    elif df_prev.shape[0] > filas_actual:
        df_prev = df_prev.iloc[:filas_actual, :]

    if df_prev.shape[1] < columnas_actual:
```

```
        cols_nuevas = [f"C{j+1}" for j in range(df_prev.shape[1], columnas_actual)]
        for c in cols_nuevas:
            df_prev[c] = 0.0
    elif df_prev.shape[1] > columnas_actual:
        df_prev = df_prev.iloc[:, :columnas_actual]

    st.session_state.df_valores = df_prev

st.subheader(" Ingresa los valores de la matriz directamente")
df_editado = st.data_editor(
    st.session_state.df_valores,
    num_rows="dynamic",
    use_container_width=True
)

st.session_state.df_valores = df_editado

A = df_editado.to_numpy()

#


# Visualizar matriz
#


st.subheader(" Matriz ingresada")
st.dataframe(df_editado.style.format("{:.1f}"), use_container_width=True)


normas = {
    'Norma L1 (m x suma columnas)': norma_1(A),
    'Norma Frobenius (cuadr tica)': norma_frobenius(A),
    'Norma L2 / Espectral (    m x)': norma_2(A),
}

st.subheader("Normas calculadas")
df_normas = pd.DataFrame({
    "Tipo de norma": list(normas.keys()),
    "Valor": list(normas.values())
})
st.table(df_normas.style.format({"Valor": "{:.4f}"}))

st.subheader("Comparaci n gr fica")
fig, ax = plt.subplots(figsize=(7, 4))
colores = ['#1f77b4', '#2ca02c', '#d62728']
nombres = list(normas.keys())
valores = list(normas.values())

for i, valor in enumerate(valores):
    ax.plot(i, valor, 'o', markersize=10, color=colores[i])
    ax.text(i, valor + 0.2, f'{valor:.4f}', ha='center', fontsize=9)

ax.set_xticks(range(len(nombres)))
ax.set_xticklabels(nombres, rotation=15)
ax.set_ylabel("Magnitud")
ax.grid(axis='y', linestyle='--', alpha=0.7)
st.pyplot(fig)
```

```python

# Calcular sensibilidad
delta, cambio = sensibilidad_por_perturbacion(A)
sensibilidad_relativa = cambio / delta if delta != 0 else 0

if sensibilidad_relativa > 1:
    conclusion = "La matriz muestra **alta sensibilidad num rica** ante perturbaciones. Esto podr a indicar mal condicionamiento."
elif sensibilidad_relativa > 0.5:
    conclusion = "La matriz presenta **moderada sensibilidad**, es importante revisar estabilidad si se usa en contextos num ricos cr ticos."
elif sensibilidad_relativa > 0.2:
    conclusion = "La matriz es **razonablemente estable** frente a perturbaciones peque as."
else:
    conclusion = "La matriz es **muy estable num ricamente**."

# Mostrar conclusiones
st.markdown(f"""
### An lisis obtenido:
{conclusion}
""")
```

Referencias

[1] R. L. Burden and J. D. Faires, *Análisis Numérico*, 9th ed. México: Cengage Learning, 2021.

[2] D. Kincaid and W. Cheney, *Análisis Numérico: Las Matemáticas del Cálculo Científico*, 1st ed. México: Cengage Learning, 2007.

[3] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, 1st ed. Philadelphia, PA, USA: SIAM, 1997.