

# UNIVERSIDAD MAYOR DE SAN ANDRÉS

## FACULTAD DE CIENCIAS PURAS Y NATURALES



### CASO EJEMPLO: SISTEMAS MAL CONDICIONADOS DE 3X3 CON GRÁFICO

**Universitario:** Cosme Pamuri Marcelo Santos  
**Docente:** Lic. Carvajal Blanco Brigida  
**Materia:** Análisis Numérico  
**Sigla** MAT - 156  
**Paralelo:** A

Para la prueba de matriz mal condicionada, se usa una matriz en específico a ser probada en este caso

```
a =  
  
    2.0000    4.0000    5.0000  
    6.0000    9.0000    8.0000  
    4.1000    5.0000    3.0000
```

con b

```
b =  
  
    220  
    490  
    274
```

se desea probar si la matriz está mal condicionada, además se desea conocer la determinante, la condicional y la matriz identidad

haciendo uso del siguiente código python

```
import numpy as np  
  
# Función que calcula la matriz inv(a) * b, el determinante de a, la  
# condicional de a y a * inv(a)  
def matriz_operaciones(a, b):  
    if a.shape[0] != a.shape[1]:  
        raise ValueError("La matriz a debe ser cuadrada para calcular  
la inversa.")  
  
    # Calcular el determinante de a  
    determinante = np.linalg.det(a)  
  
    if determinante == 0:  
        raise ValueError("La matriz a es singular y no tiene inversa.")  
  
    # Calcular la matriz inversa de a  
    a_inv = np.linalg.inv(a)  
  
    # Multiplicar la matriz inversa de a por la matriz b  
    resultado = np.dot(a_inv, b)  
  
    # Calcular la condicional de la matriz a  
    condicional = np.linalg.cond(a)  
  
    # Calcular a * inv(a) para mostrar la matriz identidad
```

```

    identidad = np.dot(a, a_inv)

    return resultado, determinante, condicional, identidad

# Ejemplo de uso con matrices 3x3
a = np.array([[2, 4, 5],
              [6, 9, 8],
              [4.1, 5, 3]])

b = np.array([220,
              490,
              274])

resultado, determinante, condicional, identidad = matriz_operaciones(a,
b)

print("Matriz inv(a) * b:")
print(resultado)
print("\nDeterminante de a:")
print(determinante)
print("\nCondicional de a:")
print(condicional)
print("\nMatriz identidad (a * inv(a)):")
print(identidad)

```

para el primer caso obtenemos los siguientes resultados

```

Matriz inv(a) * b:
[40. 10. 20.]

```

```

Determinante de a:
-1.2999999999999999

```

```

Condicional de a:
449.85632265472333

```

```

Matriz identidad (a * inv(a)):
[[ 1.00000000e+00  4.44089210e-15 -8.88178420e-16]
 [ 7.10542736e-15  1.00000000e+00  0.00000000e+00]
 [ 4.44089210e-15  4.44089210e-15  1.00000000e+00]]

```

Ahora si variamos el valor de 4.1 en la fila 3, columna 1 añadiendo un decimal

a =

```
2.0000  4.0000  5.0000
6.0000  9.0000  8.0000
4.2000  5.0000  3.0000
```

Matriz inv(a) \* b:

```
[20.          31.53846154 10.76923077]
```

Determinante de a:

```
-2.599999999999997
```

Condicional de a:

```
230.19749072982788
```

Matriz identidad (a \* inv(a)):

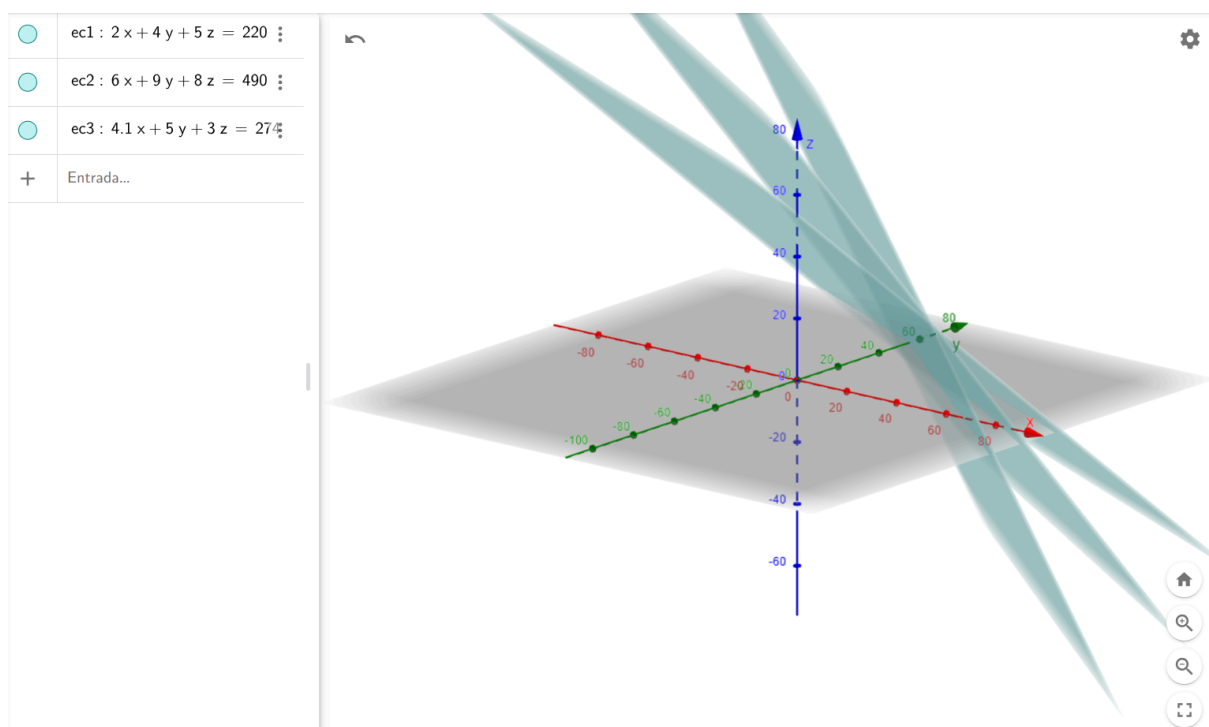
```
[[ 1.00000000e+00  8.88178420e-16  0.00000000e+00]
 [-3.55271368e-15  1.00000000e+00 -3.55271368e-15]
 [ 8.88178420e-16 -1.77635684e-15  1.00000000e+00]]
```

Los resultados cambian abruptamente, también se observa que aunque los valores de la matriz identidad son cercanos a 1 estos varían mucho y sus valores son muy delicados.

Además el valor de la condicional se dispara, de inicialmente 449.86 a 230.20, si bien ya se tenía un valor de condicional muy alta el que varíe tanto y que los resultados ambas matrices sean tan variados indica que efectivamente la matriz está mal condicionada.

## Graficas

### caso 1 con 4.1



Caso 2 con 4.2

