

# Universidad Autonoma del Estado de México

## Centro Universitario UAEM Zumpango

### Licenciatura en Ingenieria en Computación

Yair Uriel Flores Hidalgo

Hazem Alvarez Rodriguez

In [ ]:

## Sistema de ecuaciones lineales

Un sistema de  $n$  ecuaciones con  $n$  incógnitas se puede escribir de la siguiente forma:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

El sistema también se puede escribir en forma matricial de la forma:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

Lo que se puede reescribir de forma compacta utilizando  $\mathbf{Ax} = \mathbf{b}$  donde  $\mathbf{A}$  es una matriz de dimensión  $n \times n$  y  $\mathbf{x}$  y  $\mathbf{b}$  son dos vectores columna de longitud  $n$ . En esta ecuación se ha para despejar el valor de  $\mathbf{x}$ .

Para ello, asumiendo que la matriz  $\mathbf{A}$  es regular se puede multiplicar la expresión por la inversa de  $\mathbf{A}$ ,  $\mathbf{A}^{-1}$

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}$$

In [9]:

```
import numpy as np

A = np.matrix([[2, 3],[1, -2]])
b = np.matrix([[8],[-10]])

print("This is A \n",A)
print("This is b \n", b)

x=(A**-1)*b
print("Inverse A-1\n", x)

print (np.linalg.det(A))
```

```

This is A
[[ 2  3]
 [ 1 -2]]
This is b
[[ 8]
 [-10]]
Inverse A-1
[[-2.]
 [ 4.]]

```

Usamos Numpy para calcular los determinantes en Python. Veamos un ejercicio en el que vamos a calcular el determinante de la siguiente matriz A:  $A = (9)$

Como es lógico, el determinante de dicha matriz A de orden 1 va a ser el único valor que contiene dicha matriz.

```

In [10]: #determinantes en Python
import numpy as np

A = np.array ([[9]])
print (A.shape)

print (np.linalg.det (A))

```

```

(1, 1)
9.0000000000000002

```

Ese residuo infinitesimal es un redondeo de las operaciones en bytes.

## Orden

Cuando se habla de orden en una matriz, nos referimos al tamaño  $m \times n$  de la matriz. En las matrices cuadradas  $m = n$ .

Calcula el determinante de la matriz de segundo orden B:

$$B = \begin{pmatrix} 4 & -1 \\ -2 & 0 \end{pmatrix}$$

Aplicando la definición, obtenemos que:

$$|B| = a_{11} \cdot a_{22} - a_{21} a_{12}$$

$$|B| = 4 \cdot 0 - (-2) \cdot (-1) = -2$$

Por lo tanto:

```

In [12]: #determinantes en Python
B = np.array ([[4,  -1],
               [-2,  0]])

np.linalg.det (B)

```

Out[12]: -2.0

Calcula el determinante de la matriz de tercer orden C:

$$C = \begin{bmatrix} 5 & 0 & 2 \\ 3 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

```
In [13]: #determinantes en Python
C = np.array([[5, 0, 2],
              [3, 1, 1],
              [0, 1, 2]])

np.linalg.det(C)
```

Out[13]: 11.000000000000002

```
In [17]: import numpy as np

A4 = np.matrix([[3, 2, -1],[2, -2,4] , [-1, 1/2, -1]])
b4 = np.matrix([[1],[-2] , [0]])

print("This is A \n",A4)
print("This is b \n", b4)

x=(A4**(-1))*b4
print("Inverse A-1\n", x)

print (np.linalg.det(A4))
```

```
This is A
[[ 3.  2. -1. ]
 [ 2. -2.  4. ]
 [-1.  0.5 -1. ]]
This is b
[[ 1]
 [-2]
 [ 0]]
Inverse A-1
[[ 1.]
 [-2.]
 [-2.]]
-3.0000000000000036
```

Recordemos...

Un **determinante** es un número que puede calcularse sobre las matrices cuadradas. Su valor se da apartir de la suma de los productos de las diagonales de la matriz en una dirección menos la suma de los productos de las diagonales en la otra dirección y se representa con el símbolo  $|A|$ .

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$|A| = (a_{11} a_{22} a_{33} + a_{12} a_{23} a_{31} + a_{13} a_{21} a_{32}) - (a_{31} a_{22} a_{13} + a_{32} a_{23} a_{11} + a_{33} a_{21} a_{12})$$

- A demás una matriz se considera singular solo si su determinante es nulo y, en ese caso, no existe su matriz inversa.

Recuperado de Calcular determinantes en Python en <https://keepcoding.io/blog/calcular-determinantes-en-python/>

Empleando **Cómo utilizar Markdown para escribir documentación técnica** de <https://experienceleague.adobe.com/es/docs/contributor/contributor-guide/writing-essentials/markdown>

## Actividad.

1. Empleando Jupyter Notebook, resuelva los siguientes ejercicios,
2. Ingrese a su git hub,
3. Cree un repositorio con nombre 2doPar24B,
4. Suba su código fuente y el PDF correspondiente, donde se muestre la ejecución,
5. Recuerde agregar su nombre, periodo y UA a su código,

### Ejercicios:

\*Determine la solución y el determinante para:

1.  $A = (9)$

2.  $B = \begin{pmatrix} 4 & -1 \\ -2 & 0 \end{pmatrix}$

3.  $C = \begin{pmatrix} 5 & 0 & 2 \\ 3 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$

4. Empleando  $\begin{cases} 3x + 2y - z = 1 \\ 2x - 2y + 4z = -2 \\ -x + \frac{1}{2}y - z = 0 \end{cases}$

- Identifique la forma matricial empleando  $Ax=b$
- Calcule el valor de  $x$
- Calcule su determinante

## Ejercicio 1

```
In [29]: import numpy as np

B= np.matrix([[4, -1],[ -2, 0 ] ])

print("This is B \n",B)

print (" Determinante: ")
print (np.linalg.det(B))
```

```
This is B
[[ 4 -1]
 [-2  0]]
This is c
[[5 0 2]
 [3 1 1]
 [0 1 2]]
Determinante:
-2.0
```

## Ejercicio 2

```
In [30]: import numpy as np
c = np.matrix([[5, 0, 2],[3, 1, 1], [0, 1, 2]])
print("This is c \n", c)

print (" Determinante: ")
print (np.linalg.det(c))
```

```
This is c
[[5 0 2]
 [3 1 1]
 [0 1 2]]
Determinante:
11.000000000000002
```

## Ejercicio 3

```
In [32]: A = np.matrix([[3, 2, -1], [2,-2,4], [-1,1/2,1]])
b = np.matrix([[1],[-2],[0]])

x = (A**-1)*b
print("Matriz Inversa ", x)
print("Determinante de A", np.linalg.det(A))
```

```
Matriz Inversa [[-0.04347826]
 [ 0.43478261]
 [-0.26086957]]
Determinante de A -23.000000000000001
```

```
In [ ]:
```