Universidad Autónoma del Estado de México

Centro Universitario UAEM Zumpango

Ingeniería en Computación

Graficación Computacional

Alumno: Diego Argel Navarrete Godines

Profesor: Hazem

Fecha: 23 de octubre del 2024

Matrices y sistemas de ecuaciones

import numpy as np A = np.matrix([[2, 3],[1, -2]]) b = np.matrix([[8],[-10]]) x = (A**-1)*b

$$
\begin{aligned}
2x_1 + 3x_2 &= 8 \\
x_1 - 2x_2 &= -10
\end{aligned}
$$

Sistema de ecuaciones lineales

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

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

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

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

Lo que se puede reescribir de forma compacta utilizando:

$$Ax = b$$

donde (A) es una matriz de dimensión (n \times n) y (x) y (b) son dos vectores columna de longitud (n). En esta ecuación, se busca despejar el valor de (x). Para ello, asumiendo que la matriz (A) es regular, se puede multiplicar la expresión por la inversa de (A), es decir, (A^{-1}):

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

En este caso, (A^{-1} A) es la matriz identidad, por lo que se puede escribir la solución del sistema como:

$$x = A^{-1}b$$

Esta operación es lo que se ha escrito anteriormente en el código Python. La matriz inversa de (A) se puede obtener mediante `A**-1` y, al multiplicar esta matriz por (b), se obtiene el resultado buscado.

Determinar si el sistema se puede resolver

In [4]:
```
if np.linalg.det(A) == 0:
    x = None
    print("No se puede resolver")
else:
    x = (A**-1)*b
```

In [7]:
```
print("Matriz A: ", A)
print("Matriz B: ", b)
print("Inversa: ", x)
```

```
Matriz A:  [[ 2  3]
 [ 1 -2]]
Matriz B:  [[  8]
 [-10]]
Inversa:  [[-2.]
 [ 4.]]
```

Calcular determinantes en Python

4. Empleando

$$\begin{aligned} 3x + 2y - \quad z &= \quad 1 \\ 2x - 2y + 4z &= -2 \\ -x + \frac{1}{2}y - \quad z &= \quad 0 \end{aligned}$$

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

In [11]:
```
A4 = np.matrix([[3, 2, -1], [2, -2, 4], [-1, 1/2, -1]])
b4 = np.matrix([[1],[-2],[0]])

print('Esto es A', A4)
print('Esto es b', b4)
```

```
Esto es A [[ 3.   2.  -1. ]
 [ 2.  -2.   4. ]
 [-1.   0.5 -1. ]]
Esto es b [[ 1]
 [-2]
 [ 0]]
```

In [12]:
```
x = (A4**-1)*b4
print(x)
```

```
[[ 1.]
 [-2.]
 [-2.]]
```

In [14]:
```
print(np.linalg.det(A4))
```

```
-3.0000000000000036
```

EJERCICIOS

1. A = (9)

In [2]:
```
import numpy as np

A = np.matrix([[9]])

print("this is A\n",A)
det = np.linalg.det(A)
print("this is the determinante\n",det)
```

```
this is A
 [[9]]
this is the determinante
 9.000000000000002
```

Determine la solución y el determinante para:

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

In [19]:
```python
B = np.array ([[4,  -1],[-2,   0]])
print(B)
```

```
[[ 4 -1]
 [-2  0]]
```

In [23]:
```python
print("Determinante es: ", np.linalg.det (B))
```

```
Determinante es:  -2.0
```

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

In [17]:
```python
C = np.array ([[5,   0,   2],[3,   1,   1],[0,   1,   2]])
print(C)
```

```
[[5 0 2]
 [3 1 1]
 [0 1 2]]
```

In [24]:
```python
print("Determinante es: ", np.linalg.det (C))
```

```
Determinante es:  11.000000000000002
```