

PYTHON # 6

Determinantes, Inversas y Regla de Cramer.

Comandos Numpy: se utilizará la sublibrería linalg de Álgebra Lineal

<https://docs.scipy.org/doc/numpy/reference/routines.linalg.html>

- np.linalg.det()
- np.linalg.inv()

Comandos Sympy:

1. A.det()
2. A.inv()

Comandos Time

1. time.time()

En Python06.py encontrará la definición ReglaCramer() que le permitirá resolver un sistema de ecuaciones utilizando la Regla de Cramer.

```
def ReglaCramer(B,C)

    for i in range(0,len(B)):
        for j in range(0,len(B)):
            C[j][i]=B[j]
            if i>0:
                C[j][i-1]=A[j][i-1]
        X.append(round(linalg.det(C)/linalg.det(A),1))

    return X
```

Ejercicios

Debe imprimir en consola cada cambio u operación realizada.

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

$$b_2 = \begin{bmatrix} 1 \\ 8 \\ 4 \\ 5 \end{bmatrix}$$

1. Calcule los determinantes e inversas (usando numpy y sympy) de A y B. **(20 pts.)**

2. Resuelva los sistemas de ecuaciones $Ax = b_1$ $Cx = b_2$. **(45 pts.)**
 - a. **Utilice el método de la inversa de numpy**
 - b. **Utilice el método de la inversa de sympy**
 - c. **Utilice la Regla de Cramer**
3. Cree una matriz C 100×100 con entradas aleatorias y un vector d 100×1 con entradas aleatorias. Imprima el tiempo que se tarda cada método en resolver el sistema de ecuaciones $Cx=d$ utilizando los tres métodos. **(10 pts.)**
4. Cree una matriz C 500×500 con entradas aleatorias y un vector d 100×1 con entradas aleatorias. Imprima el tiempo que se tarda cada método en resolver el sistema de ecuaciones $Cx=d$ utilizando los tres métodos. **(10 pts.)**
5. Explique qué diferencias nota en el uso de los 3 métodos para resolver sistemas de ecuaciones $Cx=d$. **(15 pts.)**