

TP01_ Méthodes Numériques

Exo_01 :

Compléter le code ci-dessous pour afficher la solution d'un système d'équations linéaires ($Ax=Y$, $A(n,n)$, $X(n)$, $Y(n)$) en utilisant la fonction python (**linalg.solve**).

```
import numpy as np
n=int(input("Entrer l'ordre de la matrice : "))
""" Lire la matrice A """
A=np.ones((n,n))
print ("entrer les éléments de la matrice A : ")
for i in range (n):
    for j in range (n):
        A[i,j]= int(input("A["+str(i)+","+str(j)+"]: "))
""" Lire le tableau y """
y=np.ones(n)
print ("entrer les éléments du tableau y : ")
.....
.....
""" Calculer et afficher la solution """
.....
.....
```

Exo_02 :

Ecrire un script python qui permet de calculer et d'afficher la solution d'un système d'équations linéaires ($Ax=Y$) à matrice triangulaire supérieure (Sans utiliser des fonctions prédéfinies).

Exo_03 :

Soit un un système d'équations linéaires ($Ax=Y$) d'ordre (n).

1- Ecrire un script python permettant de réaliser les opérations de la méthode de GAUSS sur la première ligne de A.

c-à-d : la ligne pivot est L_0 , rendre A_{00} à "1" et l'éléments au-dessous de A_{00} à "0".

Exemple:

$$\left(\begin{array}{cccc|c} 2 & 4 & -2 & 0 & -6 \\ 1 & 3 & 0 & 1 & 0 \\ 3 & -1 & 1 & 2 & 8 \\ 0 & -1 & 2 & 1 & 6 \end{array} \right) \rightarrow \left(\begin{array}{cccc|c} 1 & 2 & -1 & 0 & -3 \\ 0 & 1 & 1 & 1 & 3 \\ 0 & -7 & 4 & 2 & 17 \\ 0 & -1 & 2 & 1 & 6 \end{array} \right)$$

2- Modifier le script pour effectuer les opérations de la méthode de GAUSS sur une ligne "i" introduite au clavier.