

## 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]= float(input("A["+str(i)+","+str(j)+"]: "))
# Lire le tableau y
Y=np.ones(n)
print ("entrer les éléments du tableau y : ")
for i in range (n):
    Y[i]= float(input("Y["+str(i)+"]: "))

# Calculer et afficher la solution
X=np.ones(n)
X=np.linalg.solve(A,Y)
print ("la solution est =", X)
```

**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 diagonale.

```
import numpy as np
n=int(input("Entrer l'ordre de la matrice : "))
#Lire la matrice A
A=np.zeros((n,n))
print ("entrer les éléments non nul de la matrice A : ")
for i in range (n):
    A[i,i]= float(input("A["+str(i)+","+str(i)+"]: "))
# Lire le tableau y
Y=np.zeros(n)
print ("entrer les éléments du tableau y : ")
for i in range (n):
    Y[i]= float(input("Y["+str(i)+"]: "))

# Calculer et afficher la solution
```

```
X=np.zeros(n)
for i in range (n):
    X[i]= Y[i]/A[i,i]
print ("la solution est =", X)
```

### Exo\_03 :

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.

```
import numpy as np
n=int(input("Entrer l'ordre de la matrice : "))
#Lire la matrice A
A=np.zeros((n,n))
print ("entrer les éléments non nul de la matrice A : ")

for i in range (n):
    for j in range (i,n):
        A[i,j]= float(input("A["+str(i)+","+str(j)+"]: "))
print (A)
# Lire le tableau y
Y=np.zeros(n)
print ("entrer les éléments du tableau y : ")
for i in range (n):
    Y[i]= float(input("Y["+str(i)+"]: "))

# Calculer et afficher la solution
X=np.ones(n)
for i in range (n-1, -1, -1):
    S=0
    for k in range (i+1,n):
        S+= A[i,i]*X[k]
    X[i]= (Y[i]-S)/A[i,i]
print ("la solution est =", X)
```

### Exo\_04 :