

Série de TP 3
Méthodes itératives : Jacobi & Gauss-Siedel

Exercice 1 :

En utilisant la fonction ci-dessous, écrire en python le script qui permet de résoudre le système suivant, par la méthode de Jacobi en initialisant $(0, 0, 0, 0)^T$.

$$\begin{pmatrix} 5 & -1 & -1 & -1 \\ -1 & 10 & -1 & -1 \\ -1 & -1 & 5 & -1 \\ -1 & -1 & -1 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -4 \\ 12 \\ 8 \\ 34 \end{pmatrix}$$

```
def jacobi(A,b,N=25,x=None):
    # Faire une initialisation si nécessaire
    if x is None:
        x = np.zeros(len(A[0]))

    D = np.diag(A)
    R = A - np.diagflat(D)

    # Itérer N fois
    for i in range(N):
        x = (b - np.dot(R,x)) / D
    return x
```

Exercice 2 :

En utilisant la fonction ci-dessous, écrire en python le script qui permet de résoudre le système suivant, par la méthode de Gauss Seidel en initialisant $(0, 0, 0, 0)^T$.

$$\begin{pmatrix} 5 & -1 & -1 & -1 \\ -1 & 10 & -1 & -1 \\ -1 & -1 & 5 & -1 \\ -1 & -1 & -1 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -4 \\ 12 \\ 8 \\ 34 \end{pmatrix}$$

```
def seidel(a, x ,b):
    n = len(a)
    for j in range(0, n):
        # variable temporaire d pour sauvegarder b[j]
        d = b[j]

        # Calculer x1, x2, x3,x4
        for i in range(0, n):
            if(j != i):
                d-=a[j][i] * x[i]
        # Mise à jour de la solution
        x[j] = d / a[j][j]
    # renvoyer la solution mise à jour
    return x
```