# Résolution numérique des systèmes linéaires

Un système linéaire est un ensemble d'équations linéaires qui peuvent être représentées sous la forme $Ax = B$. Les méthodes de résolution des systèmes linéaires, comme Pivot de Gauss, la décomposition LU et Cholesky, sont choisies en fonction des propriétés de la matrice. Par exemple la méthode de Pivot de gauss peut être choisie pour les petites matrices, la décomposition LU pour les matrices générales de grandes tailles, la méthode de Cholesky pour les matrices symétriques définies positives.

## Résolution du système linéaire par la méthode du pivot de Gauss

### Description de la méthode

Le but de cette méthode est de transformer la matrice augmentée $M = (A|B)$ (qui a $n$ lignes et $n+1$ colonnes) en une matrice augmentée équivalente $M' = (A'|B')$ où $A'$ est une matrice triangulaire supérieure à coefficients diagonaux non nuls), puis de résoudre le système linéaire associé à cette nouvelle matrice augmentée.
L'algorithme du pivot de Gauss se déroule donc en deux étapes :

**Etape 1 : Triangularisation**

Cette étape consiste à transformer progressivement la matrice $A$ par opérations élémentaires sur les lignes en une matrice triangulaire supérieure par des itérations successives.

Si on désigne par $L_i$ la $i^{me}$ ligne du système linéaire, les opérations élémentaires sont :

- Permuter les lignes $i$ et $j$ de $A$ ainsi que celle de $B$ : $L_i \leftrightarrow L_j$
- Multiplier la ligne $i$ par $\lambda$ (un scalaire non nul) : $L_i \leftarrow \lambda L_i$
- Remplacer la ligne $i$ par $L_i + \lambda L_j$ : $L_i \leftarrow L_i + \lambda L_j$

Ces opérations effectuées dans n'importe quel ordre et en nombre quelconque, laissent invariant l'ensemble des solutions du système linéaire.

Le système devient alors triangulaire :

$$\mathbf{S'} = \begin{cases} a'_{11}x_1 + a'_{12}x_2 + \ldots + a'_{1n}x_n = b'_1 \\ a'_{22}x_2 + \ldots + a'_{2n}x_n = b'_2 \\ \ddots \quad \vdots \\ a'_{nn}x_n = b'_n \end{cases}$$

Essayer d'appliquer le principe qui suit sur ce système linéaire :

$$S = \begin{cases} x - 3y - 2z = 6 \\ 2x - 4y - 3z = 8 \\ -3x + 6y + 8z = -5 \end{cases}$$

Soit $M = (A|B)$ :

$$\left(\begin{array}{ccc|c} 1 & -3 & -2 & 6 \\ 2 & -4 & -3 & 8 \\ -3 & 6 & 8 & -5 \end{array}\right)$$

Itération 1

- On cherche un coefficient $a_{11}$ devant $x_1$ qui soit non nul, appelé *premier pivot*. Si ce coefficient apparaît en ligne $i$, on permute les lignes 1 et la ligne $i$. Si tous les coefficients sont nuls, on affiche "PIVOT NUL STOP".
- On élimine par la suite tous les coefficients devant $x_1$ à partir de la ligne 2 jusqu'à la ligne $n$ en appliquant la relation $L_i \leftarrow L_i - \frac{a_{i1}}{a_{11}}L_1$ aux coefficients des lignes entre 2 à $n$. $L_i$ représente les coefficients de la ligne $i$ de $A$ .

A l'itération suivante, la ligne 1 reste inchangée, et on réapplique le même processus à la sous matrice à partir de la ligne 2.

**Etape 2 : Remontée**
La deuxième étape de remontée consiste à résoudre, de bas en haut, le système triangulaire obtenu $S'$ :

$$\mathbf{S'} = \begin{cases} a'_{11}x_1 + a'_{12}x_2 + \ldots + a'_{1n}x_n = b'_1 \\ a'_{22}x_2 + \ldots + a'_{2n}x_n = b'_2 \\ \ddots \quad \vdots \\ a'_{nn}x_n = b'_n \end{cases}$$

avec $a'_{11} \neq 0, \ldots, a'_{nn} \neq 0$.
Ce système est triangulaire, sa solution s'obtient par substitutions successives. On commence par la résolution de la dernière équation, à une seule inconnue, pour déterminer le $x_n = \frac{b'_n}{a'_{nn}}$ puis, on substitue $x_n$ dans l'équation de la ligne $n-1$ pour déterminer $x_{n-1}$ et ainsi de suite.
On calcule donc les valeurs de $x_1, \ldots, x_n$ par récurrence descendante.
Nous avons :

$$\begin{cases} x_n = \dfrac{b'_n}{a'_{nn}} \\ x_i = \dfrac{1}{a'_{ii}} \left( b'_i - \displaystyle\sum_{j=i+1}^{n} a'_{ij}xj \right) \qquad \forall i \in \{n-1, \ldots, 1\} \end{cases}$$

**Implémentation de la méthode du pivot de Gauss en python**

**Indication :**

- Penser à travailler sur une copie de la matrice et du vecteur pour ne pas altérer les données d'origine. Modifier une donnée d'une extraction de tableau entraîne aussi une modification du tableau initial. Pour remédier à ce problème on peut utiliser la commande `np.copy()`

**Travail à faire :**

1. Ecrire une fonction python sans paramètres, appelée **taille**, qui permet de saisir et retourner un entier superieur à 2.
2. Ecrire une fonction python, nommée **saisie_Vect** qui permet la saisie des $n$ coefficients réels d'un vecteur. Cette fonction doit retourner une matrice $B$ unicolonne d'ordre $(n, 1)$.
3. Ecrire une fonction python, nommée **saisie_Mat** qui permet la saisie des coefficients réels d'une matrice carrée d'ordre $n$. Cette fonction doit retourner une matrice $A$ carrée d'ordre $(n, n)$.
4. Ecrire une fonction python, nommée **triangulaire**, qui prend en paramètres une matrice **M** d'ordre $(n, n+1)$ et qui rend triangulaire supérieure la matrice **M**.
5. Ecrire une fonction python, nommée **remontee**, qui prend en paramètres une matrice **T** d'ordre $(n, n+1)$ triangulaire supérieure et retourne un vecteur **X** solution du système triangulaire.
6. Ecrire une fonction python, nommée **pivot_Gauss**, qui prend en paramètres une matrice **A** et un vecteur **B** et retourne le vecteur **X**, solution du système.
7. Déterminer la matrice des inconnus $X$ :

$$A = \begin{pmatrix} 2 & -5 & 1 & 3 \\ 4 & 7 & 8 & 2 \\ 3 & 1 & 1 & 6 \\ 4 & 1 & 7 & 9 \end{pmatrix} \qquad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \qquad B = \begin{pmatrix} 5 \\ 10 \\ 2 \\ 6 \end{pmatrix}$$

# 1 Déjà disponible en Python

Comme pour tous les algorithmes de ce chapitre, nous ne ferons que réimplémenter des fonctions déjà existantes en Python. Concernant les méthodes de résolution des systèmes linéaires, tout se trouve dans le module `numpy.linalg`. Il contient entre autres les fonctions suivantes :

`numpy.linalg.solve(A,B)` : permet de résoudre le système linéaire $AX = B$