

SYSTEMES LINEAIRES EN PYTHON

Pivot de Gauss

Objectif

Résolution numérique du système $AX = b$ avec $A = (a_{ij}) \in \mathcal{M}_n(\mathbb{R})$, $X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, $b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$

1) CAS D'UN SYSTÈME TRIANGULAIRE

$$(S) : \begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,p}x_p = b_1 \\ a_{2,2}x_2 + \cdots + a_{2,p}x_p = b_2 \\ \vdots \\ a_{n,n}x_n = b_n \end{cases} \quad \text{On a alors :} \quad \begin{cases} x_n = \dots\dots\dots \\ \forall i \in [1, n-1], x_i = \dots\dots\dots \end{cases}$$

- Écrire une fonction python **remonter(M)** qui prend en argument une matrice M concaténant A et b et qui renvoie la solution du système triangulaire.
- Tester le programme sur le système suivant :

$$(S) : \begin{cases} x - 3y + 2z = 3 \\ -3y - z = 0 \\ -z = -7 \end{cases}$$

2) SYSTÈME DE n ÉQUATIONS À n INCONNUES

$$(S) : \begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 & (L_1) \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 & (L_2) \\ \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n = b_n & (L_n) \end{cases}$$

Opérations élémentaires sur les lignes

Les opérations élémentaires sur les lignes $L_1, L_2, \dots, L_n$ du système (S).

- $L_i \leftarrow \alpha L_i$
- $L_i \leftarrow L_i + \beta L_j \quad \beta \in \mathbb{R}$
- $L_i \xleftrightarrow{\alpha \neq 0} \alpha L_i + \beta L_j$
- $L_i \longleftrightarrow L_j$



Méthode du pivot de Gauss

Effectuons **des opérations élémentaires** sur les lignes du système (S) jusqu'à obtenir un système équivalent simple, dont la solution se détermine facilement.

- **Étape 1 :** On suppose $a_{1,1} \neq 0$ par exemple (quitte à permuter les équations ou les inconnues).

On supprime la variable x_1 , en effectuant les opérations élémentaires :

$$L_i \leftarrow a_{1,1}L_i - a_{i,1}L_1 \quad \text{pour } 2 \leq i \leq n$$

On obtient alors un nouveau système :

$$(S') : \begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ a'_{2,2}x_2 + a'_{2,3}x_3 + \dots + a'_{2,n}x_n = b'_2 & L_2 \leftarrow a_{1,1}L_2 - a_{2,1}L_1 \\ a'_{3,2}x_2 + a'_{3,3}x_3 + \dots + a'_{3,n}x_n = b'_3 & L_3 \leftarrow a_{1,1}L_3 - a_{3,1}L_1 \\ \vdots \\ a'_{n,2}x_2 + a'_{n,3}x_3 + \dots + a'_{n,n}x_n = b'_n & L_n \leftarrow a_{1,1}L_n - a_{n,1}L_1 \end{cases}$$

- **Étape 2 :** On applique à ces $(n-1)$ lignes la même méthode que précédemment pour « éliminer » x_2 des $(n-2)$ dernières lignes.

$$(S'') : \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a'_{22}x_2 + a'_{23}x_3 + \dots + a'_{2n}x_n = b'_2 \\ a''_{33}x_3 + \dots + a''_{3n}x_n = b''_3 & L_3 \leftarrow a'_{1,1}L_3 - a'_{3,1}L_2 \\ \vdots \\ a''_{n3}x_3 + \dots + a''_{nn}x_n = b''_n & L_n \leftarrow a'_{1,1}L_n - a'_{n,1}L_2 \end{cases}$$

En itérant le procédé, on finit par obtenir un système triangulaire du type :

$$(S) : \begin{cases} \alpha_{1,1}x_1 + \alpha_{1,2}x_2 + \dots + \alpha_{1,n}x_n = \beta_1 & L_1 \\ 0 + \alpha_{2,2}x_2 + \dots + \alpha_{2,n}x_n = \beta_2 & L_2 \\ \vdots & \vdots \\ 0 + \dots + 0 + \alpha_{r,r}x_r + \dots + \alpha_{r,n}x_n = \beta_r & L_r \end{cases}$$

où $\alpha_{i,i} \in \mathbb{K}^*$ pour $1 \leq i \leq r$.

- **Étape 3 :** On résout ce système triangulaire.

3) LES FONCTION PRINCIPALES DE LA MÉTHODE EN PYTHON

Q.1 : Écrire une fonction **transvection**(M,i,j,a) qui reçoit en paramètres une matrice M, deux entiers i et j représentant les indices de deux lignes dans la matrice M, et un nombre réel a la fonction remplace la ligne L_i par $L_i + aL_j$.

Q.2 : Écrire une fonction **echanger-lignes**(M,i,j) qui reçoit en paramètres une matrice M, la fonction échange les deux lignes L_i et L_j .

Q.3 : Meilleur choix du pivot

Écrire la fonction : **ligne-pivot** (M, j), qui reçoit en paramètres une matrice M , et un entier j représentant une colonne dans la matrice M .

La fonction retourne l'indice p tel que :

$$|M[p, j]| = \max\{|M[i, j]| / 0 \leq i \leq j\}$$

NB : Afin de minimiser l'influence des erreurs d'arrondi, on doit choisir à chaque étape j le pivot le plus grand en valeur absolue.

Q.4 : Écrire une fonction **concatener**(A, b) qui reçoit les matrices A et b et renvoie la matrice $M = (A \mid b)$.

Q.5 : Écrire une fonction **triangulaire-sup**(M) qui reçoit une matrice M qui transforme la matrice $M = (A \mid b)$ en un système triangulaire supérieure équivalent.