UNIVERSITÉ MOHAMED V
FACULTÉ DES SCIENCES RABAT
Département de Mathématiques

Filière :
Sciences de la Matière Physique
PC- S_4

Analyse Numérique

Par
Prof : Mohamed Najib LAATABI
najib.laatabi@um5r.ac.ma

Remarque importante

Ce document est un **cours complémentaire**, destiné uniquement à accompagner la correction des travaux dirigés (TD). Il ne se substitue **en aucun cas** au cours magistral dispensé par le **professeur responsable de la matière**. L'auteur de ce document assure uniquement les **séances de TD**.

Toute remarque de votre part est la bienvenue.
Si vous identifiez une erreur, veuillez la signaler.

Année Universitaire : **2025–2026**

Table des matières

2 Résolution des systèmes linéaires **1**

2.1 Objectifs et cadre général 1
2.1.1 Rappels sur les systèmes linéaires 1
2.2 Méthodes directes 2
2.2.1 Définition 2
2.2.2 Méthodes directes 2
2.2.3 Méthode de Gauss 6
2.2.4 Méthode de LU 8
2.2.5 Appplications de la Factorisation LU 11
2.2.6 Méthode de Cholesky 13
2.2.7 Application de la Factorisation de Cholesky 15
2.3 Méthode itératives 16
2.3.1 Principe Général 17
2.3.2 Matrice d'Itération et Erreur 17
2.4 La Méthode de Jacobi 18
2.4.1 Construction de la méthode 18
2.4.2 Formule composante par composante 19
2.4.3 Convergence de Jacobi 19
2.4.4 Exemple numérique 20
2.5 La Méthode de Gauss-Seidel 20
2.5.1 Construction de la méthode 21
2.5.2 Formule composante par composante 21
2.5.3 Convergence de Gauss-Seidel 21
2.5.4 Exemple numérique 22
2.6 Comparaison et Critères de Choix 22
2.6.1 Tableau récapitulatif 22
2.6.2 Exemple : Cas où Jacobi et Gauss-Seidel diverge 22

Symboles et Abréviations

Symbole	Description / Signification
$\mathbb{K}$	Corps des scalaires ($\mathbb{R}$ ou $\mathbb{C}$)
$\mathcal{M}_n(\mathbb{K})$	Matrices carrées d'ordre n sur $\mathbb{K}$
$\mathcal{M}_{m,n}(\mathbb{K})$	Matrices de type (m, n) sur $\mathbb{K}$
I_n	Matrice identité d'ordre n
a_{ij}	Coefficient ligne i, colonne j de A
$\overline{z}$	Conjugué complexe du scalaire $z \in \mathbb{C}$
A^T ou ${}^tA$	Transposée de A : $(A^T)_{ij} = a_{ji}$
A^*	Adjointe de A : $(A^*)_{ij} = \overline{a_{ji}}$, $\quad A^* = \overline{A^T}$
$\mathrm{tr}(A)$	Trace de A : somme des termes diagonaux $\sum_{i=1}^{n} a_{ii}$
$\det(A)$	Déterminant de A
A^{-1}	Inverse de A : unique matrice vérifiant $AB = BA = I_n$
$P_A(\lambda) = \det(A - \lambda I_n)$	Polynôme caractéristique de A
$\mathrm{sp}(A)$	Spectre de A
λ_i	Valeurs propres de A
$\rho(A) = \max_i \lvert\lambda_i\rvert$	Rayon spectral de A
$E_\lambda = \ker(A - \lambda I_n)$	Sous-espace propre associé à $\lambda \quad (Ax = \lambda x,\ x \neq 0)$
$\lVert\cdot\rVert,\ \lvert\!\lvert\!\lvert\cdot\rvert\!\rvert\!\rvert$	Norme vectorielle sur $\mathbb{K}^n$; norme matricielle sur $\mathcal{M}_n(\mathbb{K})$
$\lvert\!\lvert\!\lvert A\rvert\!\rvert\!\rvert_1,\ \lvert\!\lvert\!\lvert A\rvert\!\rvert\!\rvert_\infty$	Norme 1 (max sommes colonnes) ; norme ∞ (max sommes lignes)
$\rho(A) = \max_i \lvert\lambda_i\rvert$	Rayon spectral de A
$\lvert\!\lvert\!\lvert A\rvert\!\rvert\!\rvert_2 = \sqrt{\rho(A^*A)}$	Norme spectrale ; si $A^*A = AA^*$ alors $\lvert\!\lvert\!\lvert A\rvert\!\rvert\!\rvert_2 = \rho(A)$
DSD	Matrice à Diagonale Strictement Dominante
SDP	Matrice Symétrique Définie Positive

Les notations suivent les conventions standards de l'Analyse Numérique.

Chapitre

2 Résolution des systèmes linéaires

2.1 Objectifs et cadre général

On note $\mathcal{M}_n$ l'ensemble des matrices carrées d'ordre n à coefficients réels. Soit $A \in \mathcal{M}_n$ une matrice **inversible** et $b \in \mathbb{R}^n$. L'objectif est de résoudre le *système linéaire* :

$$\text{Trouver } x \in \mathbb{R}^n \text{ tel que } Ax = b. \tag{2.1}$$

Puisque A est inversible, le système (2.1) admet une **unique solution** $x = A^{-1}b \in \mathbb{R}^n$.

Ce problème est fondamental en sciences de l'ingénieur ; la taille des systèmes rencontrés peut atteindre plusieurs millions d'inconnues. Nous étudierons deux grandes familles de méthodes :

- les **méthodes directes** : solution exacte en un nombre fini d'opérations élémentaires (hors erreurs d'arrondi) ;
- les **méthodes itératives** : construction d'une suite convergeant vers la solution.

2.1.1 Rappels sur les systèmes linéaires

Un système d'équations linéaires est un ensemble de m équations portant sur n inconnues $x_1, x_2, \ldots, x_n$ appartenant à un corps $\mathbb{K}$ ($\mathbb{R}$ ou $\mathbb{C}$). Il s'écrit sous la forme :

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n & = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n & = b_2 \\ \vdots & \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n & = b_m \end{cases} \tag{2.2}$$

Pour plus de concision, on utilise la notation matricielle :

$$Ax = b \tag{2.3}$$

où :

- $A = (a_{i,j}) \in \mathcal{M}_{m,n}(\mathbb{K})$ est la matrice des coefficients ;
- $x = (x_j)_{1 \leq j \leq n} \in \mathbb{K}^n$ est le vecteur des inconnues ;
- $b = (b_i)_{1 \leq i \leq m} \in \mathbb{K}^m$ est le vecteur du second membre.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

On distingue trois cas selon le rapport entre le nombre d'équations m et le nombre d'inconnues n :

- **Cas $m < n$ (Système sous-déterminé) :** Il y a moins d'équations que d'inconnues. L'ensemble des solutions est généralement un sous-espace affine de dimension non nulle (infinité de solutions).
- **Cas $m > n$ (Système sur-déterminé) :** Il y a plus d'équations que d'inconnues. En général, le système n'admet aucune solution exacte (on cherche alors souvent une solution au sens des moindres carrés).
- **Cas $m = n$ (Système carré) :** C'est le cas fondamental. Le système admet une unique solution pour tout $b \in \mathbb{K}^n$ si et seulement si la matrice A est inversible ($\det(A) \neq 0$).

Exemple 1:

Considérons le système linéaire suivant :

$$\begin{cases} 2x_1 + x_2 - x_3 = 8 \\ -3x_1 - x_2 + 2x_3 = -11 \\ -2x_1 + x_2 + 2x_3 = -3 \end{cases}$$

Ce système se traduit par l'équation matricielle $Ax = b$ suivante :

$$\underbrace{\begin{pmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} 8 \\ -11 \\ -3 \end{pmatrix}}_{b}$$

2.2 Méthodes directes

2.2.1 Définition

Définition 2.2.1: Méthode directe

On appelle *méthode directe* de résolution de (2.1) une méthode qui donne la solution exacte x après un **nombre fini d'opérations élémentaires** : addition, soustraction, multiplication, division (et extraction de racine carrée pour la méthode de Cholesky).

2.2.2 Méthodes directes

Méthode de résolution pour les systèmes triangulaires inférieurs

Définition 2.2.2:

Une matrice carrée $(a_{i,j})_{1\leq i,j\leq n}$ est dite triangulaire inférieure (ou trigonale inférieure) si tous les éléments situés au-dessus de la diagonale principale sont nuls, c'est-à-dire si :

$$\forall (i,j) \in \{1,\dots,n\}^2, \quad i < j \implies a_{i,j} = 0$$

Une matrice triangulaire inférieure a la forme suivante :

$$A = \begin{pmatrix} a_{1,1} & 0 & \cdots & \cdots & 0 \\ a_{2,1} & a_{2,2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ a_{n,1} & a_{n,2} & \cdots & \cdots & a_{n,n} \end{pmatrix}$$

Remarque 1:

Si une matrice A est triangulaire inférieure alors $\det(A) = \prod_{i=1}^{n} a_{i,i}$.

On en déduit que A est inversible si et seulement si $a_{i,i} \neq 0$ pour tout $i = 1,\dots,n$.

Algorithme de résolution pour les systèmes triangulaires inférieurs

Considérons un système inversible 3×3 triangulaire inférieur :

$$\begin{pmatrix} l_{1,1} & 0 & 0 \\ l_{2,1} & l_{2,2} & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

La matrice étant inversible, ses termes diagonaux $l_{i,i}$ $(i = 1, 2, 3)$ sont non nuls. On peut donc déterminer successivement les valeurs inconnues x_i pour $i = 1, 2, 3$:

$$\begin{cases} x_1 = \frac{b_1}{l_{1,1}} \\ x_2 = \frac{b_2 - l_{2,1}x_1}{l_{2,2}} \\ x_3 = \frac{b_3 - l_{3,1}x_1 - l_{3,2}x_2}{l_{3,3}} \end{cases}$$

Cet algorithme peut être étendu aux systèmes $n \times n$. Dans le cas d'un système $Lx = b$, où L est une matrice triangulaire inférieure d'ordre n $(n \geq 2)$, la méthode s'écrit :

$$\begin{cases} x_1 = \frac{b_1}{l_{1,1}} \\ x_i = \frac{1}{l_{i,i}} \left(b_i - \sum_{j=1}^{i-1} l_{i,j} x_j \right), \quad i = 2,\dots,n \end{cases}$$

Algorithme 1 Descente d'un système triangulaire inférieur

Entrée : Matrice triangulaire inférieure $L \in \mathcal{M}_n(\mathbb{K})$ inversible ($l_{i,i} \neq 0$), vecteur $b \in \mathbb{K}^n$

Sortie : Vecteur solution $x \in \mathbb{K}^n$ tel que $Lx = b$

$x_1 \leftarrow \dfrac{b_1}{l_{1,1}}$

for $i = 2$ **à** n **do**

  $s \leftarrow 0$ // initialisation de la somme partielle

  for $j = 1$ **à** $i - 1$ **do**

    $s \leftarrow s + l_{i,j}\, x_j$

  end

  $x_i \leftarrow \dfrac{1}{l_{i,i}}(b_i - s)$ // calcul de la composante x_i

end

Retourner $x = (x_1, \ldots, x_n)^T$

Exemple 2:

Considérons le système triangulaire inférieur suivant :

$$\begin{pmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 9 \\ 7 \\ 14 \end{pmatrix}$$

En utilisant l'algorithme de descente, on obtient :

$$\begin{cases} x_1 = \frac{b_1}{l_{1,1}} = \frac{9}{3} = 3 \\ x_2 = \frac{b_2 - l_{2,1}x_1}{l_{2,2}} = \frac{7-1\times 3}{2} = 2 \\ x_3 = \frac{b_3 - l_{3,1}x_1 - l_{3,2}x_2}{l_{3,3}} = \frac{14-3\times 3-2\times 2}{1} = 1 \end{cases}$$

Méthode de résolution pour les systèmes triangulaires supérieurs

Définition 2.2.3:

Une matrice carrée $(a_{i,j})_{1\leq i,j\leq n}$ est dite triangulaire supérieure (ou trigonale supérieure) si tous les éléments situés en-dessous de la diagonale principale sont nuls, c'est-à-dire si :

$$\forall (i,j) \in \{1, \ldots, n\}^2, \quad i > j \implies a_{i,j} = 0$$

Une matrice triangulaire supérieure a la forme suivante :

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 & a_{n,n} \end{pmatrix}$$

Remarque 2:

Si une matrice A est triangulaire supérieure alors $\det(A) = \prod_{i=1}^{n} a_{i,i}$.
On en déduit que A est inversible si et seulement si $a_{i,i} \neq 0$ pour tout $i = 1, \ldots, n$.

Algorithme de résolution pour les systèmes triangulaires supérieurs

Considérons un système inversible 3×3 triangulaire supérieur :

$$\begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ 0 & u_{2,2} & u_{2,3} \\ 0 & 0 & u_{3,3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

La matrice étant inversible, ses termes diagonaux $u_{i,i}$ $(i = 1, 2, 3)$ sont non nuls. On peut donc déterminer successivement les valeurs inconnues en commençant par la dernière x_i pour $i = 3, 2, 1$ (c'est la phase de remontée) :

$$\begin{cases} x_3 = \frac{b_3}{u_{3,3}} \\ x_2 = \frac{b_2 - u_{2,3}x_3}{u_{2,2}} \\ x_1 = \frac{b_1 - u_{1,2}x_2 - u_{1,3}x_3}{u_{1,1}} \end{cases}$$

Cet algorithme peut être étendu aux systèmes $n \times n$. Dans le cas d'un système $Ux = b$, où U est une matrice triangulaire supérieure d'ordre n $(n \geq 2)$, la méthode s'écrit :

$$\begin{cases} x_n = \frac{b_n}{u_{n,n}} \\ x_i = \frac{1}{u_{i,i}} \left(b_i - \sum_{j=i+1}^{n} u_{i,j}x_j \right), \quad i = n-1, \ldots, 1 \end{cases}$$

Algorithme 2 Remontée d'un système triangulaire supérieur

Entrée : Matrice triangulaire supérieure $U \in \mathcal{M}_n(\mathbb{K})$ inversible ($u_{i,i} \neq 0$), vecteur $b \in \mathbb{K}^n$

Sortie : Vecteur solution $x \in \mathbb{K}^n$ tel que $Ux = b$

$x_n \leftarrow \dfrac{b_n}{u_{n,n}}$

for $i = n-1$ **à** 1 en décroissant **do**

  $s \leftarrow 0$ // initialisation de la somme partielle

  for $j = i+1$ **à** n **do**

    $s \leftarrow s + u_{i,j}\, x_j$

  end

  $x_i \leftarrow \dfrac{1}{u_{i,i}}(b_i - s)$ // calcul de la composante x_i

end

Retourner $x = (x_1, \ldots, x_n)^T$

Exemple 3:

Considérons le système triangulaire supérieur suivant :

$$\begin{pmatrix} 3 & 2 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 7 \\ 6 \end{pmatrix}$$

En utilisant l'algorithme de remontée, on obtient :

$$\begin{cases} x_3 = \frac{b_3}{u_{3,3}} = \frac{6}{2} = 3 \\ x_2 = \frac{b_2 - u_{2,3}x_3}{u_{2,2}} = \frac{7-1\times 3}{2} = 2 \\ x_1 = \frac{b_1 - u_{1,2}x_2 - u_{1,3}x_3}{u_{1,1}} = \frac{10-2\times 2-1\times 3}{3} = 1 \end{cases}$$

Le **principe général** consiste à écrire une décomposition $A = BC$, où B et C ont des propriétés favorables (triangularité, orthogonalité...). Résoudre $Ax = b$ revient alors à résoudre successivement deux systèmes plus simples :

$$By = b \qquad \text{puis} \qquad Cx = y.$$

La méthode directe la plus utilisée est la **méthode d'élimination de Gauss**, dont la forme matricielle conduit à la **décomposition LU**.

2.2.3 Méthode de Gauss

Soit $A \in \mathcal{M}_n(\mathbb{R})$ inversible et $b \in \mathbb{R}^n$. Le principe de la **méthode de Gauss** est de se ramener, par des **combinaisons linéaires de lignes**, à un système triangulaire supérieur équivalent $Ux = b'$, puis de résoudre ce système par **substitution arrière**.

Algorithme 3 Élimination de Gauss — vue générale

Entrée : Matrice $A \in \mathcal{M}_n(\mathbb{R})$ inversible, vecteur $b \in \mathbb{R}^n$
Sortie : Matrice triangulaire supérieure U, vecteur transformé $b^{(n)}$

```
Poser A^(1) ← A,   b^(1) ← b
for k = 1 à n − 1 do
    // Pivot courant : a_kk^(k)
    for i = k + 1 à n do
        Calculer le multiplicateur
            ℓ_ik ← a_ik^(k) / a_kk^(k)
        L_i ← L_i − ℓ_ik L_k // annulation de a_ik
    end
end
U ← A^(n)
```

Exemple 4:

Considérons le système $Ax = b$ avec

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 3 \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}.$$

Étape 1 : Élimination de la colonne 1

Le pivot est $a_{11}^{(1)} = 1$. Le multiplicateur vaut

$$\ell_{21} = \frac{a_{21}^{(1)}}{a_{11}^{(1)}} = \frac{1}{1} = 1.$$

On effectue $L_2 \leftarrow L_2 - \ell_{21} L_1$:

$$\underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 3 \end{pmatrix}}_{A^{(1)}} \qquad \begin{array}{c} \ell_{21} = 1 \\ L_2 \leftarrow L_2 - L_1 \\ (L_3 \text{ inchangée}) \end{array} \qquad \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 3 \end{pmatrix}}_{A^{(2)}}$$

Matriciellement, cette opération revient à multiplier $A^{(1)}$ à gauche par

$$E_2^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad A^{(2)} = E_2^{(1)} A^{(1)}.$$

Étape 2 : Élimination de la colonne 2

Le pivot est $a_{22}^{(2)} = 1$. Le multiplicateur vaut

$$\ell_{32} = \frac{a_{32}^{(2)}}{a_{22}^{(2)}} = \frac{1}{1} = 1.$$

On effectue $L_3 \leftarrow L_3 - \ell_{32}\, L_2$:

$$\underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 3 \end{pmatrix}}_{A^{(2)}} \qquad \begin{array}{c} \ell_{32} = 1 \\ L_3 \leftarrow L_3 - L_2 \\ (L_1,\, L_2 \text{ inchangées}) \end{array} \qquad \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix}}_{A^{(3)} = U}$$

Matriciellement :

$$E_3^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \qquad U = A^{(3)} = E_3^{(2)}\, A^{(2)} = E_3^{(2)}\, E_2^{(1)}\, A.$$

Bilan de la factorisation. On obtient la matrice triangulaire supérieure

$$U = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix}.$$

Les multiplicateurs utilisés sont $\ell_{21} = 1$ et $\ell_{32} = 1$, et la matrice triangulaire inférieure associée est

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \qquad \text{de sorte que } A = LU.$$

Remontée (substitution arrière). Le système $Ux = b'$ se résout par substitution arrière :

$$\begin{cases} 2\, x_3 = b'_3 & \Longrightarrow \; x_3 = \frac{b'_3}{2}, \\ x_2 + x_3 = b'_2 & \Longrightarrow \; x_2 = b'_2 - x_3, \\ x_1 + x_2 = b'_1 & \Longrightarrow \; x_1 = b'_1 - x_2. \end{cases}$$

2.2.4 Méthode de LU

En appliquant la méthode d'élimination de Gauss *sans stratégie de pivot*, on construit successivement des matrices élémentaires $E_1, E_2, \ldots, E_{n-1}$ telles que

$$\mathbf{A}^{(n)} = E_{n-1} \cdots E_2\, E_1\, \mathbf{A}^{(1)} = \mathbf{M}\, \mathbf{A},$$

où $\mathbf{A}^{(n)}$ est triangulaire supérieure. En posant

$$\mathbf{U} = \mathbf{A}^{(n)}, \qquad \mathbf{L} = \mathbf{M}^{-1} = E_1^{-1} E_2^{-1} \cdots E_{n-1}^{-1},$$

on obtient $\mathbf{A} = \mathbf{L}\,\mathbf{U}$, où $\mathbf{L}$ est triangulaire inférieure avec des 1 sur la diagonale.

Théorème 2.2.1: Existence et unicité de la décomposition LU

Soit $A \in \mathcal{M}_n(\mathbb{R})$. Si toutes les **sous-matrices principales** A_k $(k = 1, \ldots, n)$ sont inversibles, alors il existe une **unique** décomposition

$$A = LU,$$

où L est triangulaire inférieure avec des 1 sur la diagonale, et U est triangulaire supérieure inversible.

Mineur principal d'ordre i. Le **mineur principal d'ordre** i de $\mathbf{A}$ est le déterminant de la sous-matrice formée par les i premières lignes *et* les i premières colonnes :

$$\det(\mathbf{A}_i) = \begin{vmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,i} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,i} \end{vmatrix}.$$

La condition du théorème est équivalente à demander $\det(\mathbf{A}_i) \neq 0$ pour tout $i \in \{1, \ldots, n\}$.

Proposition 2.2.1: Conditions suffisantes d'existence

La factorisation LU de $\mathbf{A}$ par la méthode de Gauss existe si l'une des conditions suivantes est vérifiée :

1. La matrice $\mathbf{A}$ est **définie positive** (i.e. $\mathbf{A}$ est symétrique et $x^\top \mathbf{A} x > 0$ pour tout $x \neq 0$).
2. La matrice $\mathbf{A}$ est à **diagonale strictement dominante** (i.e. $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ pour tout i).

Exemple 5: Exemple détaillé — matrice 3×3

Soit la matrice

$$\mathbf{A} = \begin{pmatrix} 2 & 4 & -2 \\ 1 & 5 & 3 \\ -3 & 6 & 2 \end{pmatrix}.$$

On vérifie les mineurs principaux :

$$\det(\mathbf{A}_1) = 2 \neq 0, \quad \det(\mathbf{A}_2) = \begin{vmatrix} 2 & 4 \\ 1 & 5 \end{vmatrix} = 10 - 4 = 6 \neq 0, \quad \det(\mathbf{A}_3) = \det(\mathbf{A}) \neq 0.$$

La décomposition LU existe donc et est unique.

Étape 1 — élimination dans la colonne 1. On cherche à annuler les entrées a_{21} et a_{31}. Les multiplicateurs sont

$$m_{21} = \frac{a_{21}}{a_{11}} = \frac{1}{2}, \qquad m_{31} = \frac{a_{31}}{a_{11}} = \frac{-3}{2}.$$

La matrice élémentaire correspondante est

$$E_1 = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{3}{2} & 0 & 1 \end{pmatrix},$$

et on calcule $\mathbf{A}^{(2)} = E_1\,\mathbf{A}$:

$$L_2 \leftarrow L_2 - \tfrac{1}{2}\,L_1 : \quad \Big(1 - \tfrac{1}{2}\cdot 2,\ 5 - \tfrac{1}{2}\cdot 4,\ 3 - \tfrac{1}{2}\cdot(-2)\Big) = \Big(0,\ 3,\ 4\Big),$$
$$L_3 \leftarrow L_3 - \Big(-\tfrac{3}{2}\Big)\,L_1 : \quad \Big(-3 + \tfrac{3}{2}\cdot 2,\ 6 + \tfrac{3}{2}\cdot 4,\ 2 + \tfrac{3}{2}\cdot(-2)\Big) = \Big(0,\ 12,\ -1\Big).$$

On obtient :

$$\mathbf{A}^{(2)} = \begin{pmatrix} 2 & 4 & -2 \\ 0 & 3 & 4 \\ 0 & 12 & -1 \end{pmatrix}.$$

Étape 2 — élimination dans la colonne 2. Le seul multiplicateur est

$$m_{32} = \frac{a_{32}^{(2)}}{a_{22}^{(2)}} = \frac{12}{3} = 4.$$

La matrice élémentaire est

$$E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{pmatrix},$$

et on calcule $\mathbf{A}^{(3)} = E_2\,\mathbf{A}^{(2)}$:

$$L_3 \leftarrow L_3 - 4\,L_2 : \quad \Big(0,\ 12 - 4\cdot 3,\ -1 - 4\cdot 4\Big) = \Big(0,\ 0,\ -17\Big).$$

On obtient la matrice triangulaire supérieure

$$\mathbf{U} = \mathbf{A}^{(3)} = \begin{pmatrix} 2 & 4 & -2 \\ 0 & 3 & 4 \\ 0 & 0 & -17 \end{pmatrix}.$$

Assemblage de L. Les inverses des matrices élémentaires s'obtiennent en *changeant le signe* des multiplicateurs hors-diagonaux :

$$E_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix}, \qquad E_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 1 \end{pmatrix}.$$

Une propriété remarquable de l'élimination de Gauss est que $\mathbf{L} = E_1^{-1}E_2^{-1}$ s'obtient simplement en **plaçant les multiplicateurs dans le triangle inférieur** :

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{3}{2} & 4 & 1 \end{pmatrix}.$$

Vérification.

$$\mathbf{L}\,\mathbf{U} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{3}{2} & 4 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 0 & 3 & 4 \\ 0 & 0 & -17 \end{pmatrix} = \begin{pmatrix} 2 & 4 & -2 \\ 1 & 5 & 3 \\ -3 & 6 & 2 \end{pmatrix} = \mathbf{A}.$$

Propriété 2.2.1: Lecture directe des multiplicateurs

Lors de l'élimination de Gauss *sans pivot*, à chaque étape k, on met à jour les entrées restantes par :

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \underbrace{\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}}_{\ell_{ik}} \cdot a_{kj}^{(k)}, \qquad i, j > k.$$

Le coefficient ℓ_{ik} ainsi obtenu est **directement** l'entrée (i, k) de $\mathbf{L}$, et les entrées de $\mathbf{U}$ sont les lignes successives issues de cette élimination. On résume :

$$\ell_{ij} = \frac{a_{ij}^{(j)}}{a_{jj}^{(j)}} \quad (i > j), \qquad \ell_{ii} = 1, \qquad u_{ij} = a_{ij}^{(i)} \quad (i \leq j).$$

Ainsi, $\mathbf{L}$ et $\mathbf{U}$ se **construisent simultanément** au fil de l'élimination, sans calculer explicitement les matrices E_k.

2.2.5 Appplications de la Factorisation LU

Si l'on doit résoudre **plusieurs systèmes** $\mathbf{A}x = b_1, b_2, \ldots$ où seul le second membre change, ou si l'on veut calculer l'inverse d'une matrice, il est avantageux d'effectuer la réduction triangulaire **une seule fois**.

En effet, si $\mathbf{A} = \mathbf{LU}$, résoudre $\mathbf{A}x = b$ revient à résoudre **deux systèmes triangulaires successifs** :

$$\mathbf{A}x = b \iff \begin{cases} \mathbf{L}\,z = b & \text{(descente)}, \\ \mathbf{U}\,x = z & \text{(remontée)}. \end{cases}$$

La cohérence est immédiate :

$$\mathbf{A}x = \mathbf{LU}x = \mathbf{L}(\mathbf{U}x) = \mathbf{L}z = b.$$

Exemple 6:

Résolvons $\mathbf{A}x = b$ avec $b = \begin{pmatrix} 2 \\ 4 \\ -6 \end{pmatrix}$.

Descente — résolution de $\mathbf{L}y = b$ *:*

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{3}{2} & 4 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ -6 \end{pmatrix} \Longrightarrow \begin{cases} y_1 = 2, \\ y_2 = 4 - \frac{1}{2}\cdot 2 = 3, \\ y_3 = -6 + \frac{3}{2}\cdot 2 - 4\cdot 3 = -6 + 3 - 12 = -15. \end{cases}$$

Remontée — résolution de $\mathbf{U}x = y$ *:*

$$\begin{pmatrix} 2 & 4 & -2 \\ 0 & 3 & 4 \\ 0 & 0 & -17 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ -15 \end{pmatrix} \Longrightarrow \begin{cases} x_3 = \dfrac{-15}{-17} = \dfrac{15}{17}, \\ x_2 = \dfrac{1}{3}\left(3 - 4\cdot\frac{15}{17}\right) = \dfrac{1}{3}\cdot\dfrac{51-60}{17} = -\dfrac{3}{17}, \\ x_1 = \dfrac{1}{2}\left(2 - 4\cdot\left(-\frac{3}{17}\right) + 2\cdot\frac{15}{17}\right) = \dfrac{1}{2}\cdot\dfrac{34+12+30}{17} = \dfrac{76}{34} = \dfrac{38}{17}. \end{cases}$$

La solution est donc

$$\boxed{x = \frac{1}{17}\begin{pmatrix} 38 \\ -3 \\ 15 \end{pmatrix}.}$$

Algorithme 4 Factorisation LU — méthode de Doolittle

Entrée : Matrice $A \in \mathcal{M}_n(\mathbb{R})$ avec mineurs principaux non nuls
Sortie : Matrices L (triangulaire inférieure, $\ell_{ii} = 1$) et U (triangulaire supérieure)

// Initialisation - première ligne et première colonne
$u_{1,1} \leftarrow a_{1,1}$
for $j = 2$ **à** n **do**
  $u_{1,j} \leftarrow a_{1,j}$
  $\ell_{j,1} \leftarrow \dfrac{a_{j,1}}{u_{1,1}}$
end
// Boucle principale
for $i = 2$ **à** $n-1$ **do**
  // Pivot diagonal de U
  $u_{i,i} \leftarrow a_{i,i} - \displaystyle\sum_{k=1}^{i-1} \ell_{i,k}\, u_{k,i}$
  for $j = i+1$ **à** n **do**
    // Ligne i de U
    $u_{i,j} \leftarrow a_{i,j} - \displaystyle\sum_{k=1}^{i-1} \ell_{i,k}\, u_{k,j}$
    // Colonne i de L
    $\ell_{j,i} \leftarrow \dfrac{1}{u_{i,i}}\left(a_{j,i} - \displaystyle\sum_{k=1}^{i-1} \ell_{j,k}\, u_{k,i}\right)$
  end
end
// Dernier pivot
$u_{n,n} \leftarrow a_{n,n} - \displaystyle\sum_{k=1}^{n-1} \ell_{n,k}\, u_{k,n}$

Remarque 3:

Coût comparé pour m seconds membres :

— Factorisation LU (une seule fois) : $\dfrac{2n^3}{3}$ opérations.

— Chaque résolution triangulaire ($\mathbf{L}z = b$ puis $\mathbf{U}x = z$) : $2n^2$ opérations.

Le coût total avec LU est donc :

$$\underbrace{\frac{2n^3}{3}}_{\text{factorisation}} + \underbrace{2m\,n^2}_{m\text{ résolutions}} \ll \underbrace{\frac{2m\,n^3}{3}}_{m\text{ Gauss indépendants}}.$$

Exemple ($n = 100$, $m = 50$) : **1 666 667** opérations avec LU, contre **33 333 333** sans, soit un gain d'un facteur **20**.

2.2.6 Méthode de Cholesky

La méthode de Cholesky est une *variante optimisée de la factorisation LU* qui s'applique exclusivement aux matrices **symétriques définies positives**. Elle exploite la symétrie pour réduire de moitié le nombre d'opérations par rapport à la décomposition LU classique.

Théorème 2.2.2: Décomposition de Cholesky

Si $\mathbf{A} \in \mathcal{M}_n(\mathbb{R})$ est une matrice symétrique définie positive, alors il existe au moins une matrice triangulaire inférieure $\mathbf{L}$ telle que

$$\mathbf{A} = \mathbf{L}\,\mathbf{L}^\top.$$

Si de plus les éléments diagonaux $\ell_{i,i}$ de $\mathbf{L}$ sont imposés strictement positifs, alors la décomposition est *unique*.

Proposition 2.2.2: Conditions équivalentes

Soit $\mathbf{A} \in \mathcal{M}_n(\mathbb{R})$ symétrique. Les assertions suivantes sont équivalentes :

1. $\mathbf{A}$ est **définie positive** : $x^\top \mathbf{A}\, x > 0$ pour tout $x \in \mathbb{R}^n$, $x \neq 0$.
2. Toutes les **valeurs propres** de $\mathbf{A}$ sont strictement positives : $\lambda_i > 0$.
3. Tous les **mineurs principaux** sont strictement positifs : $\det(\mathbf{A}_k) > 0$ pour $k = 1, \ldots, n$.
4. La décomposition $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ (Cholesky) existe avec $\ell_{ii} > 0$.

Exemple 7: Exemple détaillé — matrice 3×3

Soit la matrice symétrique

$$\mathbf{A} = \begin{pmatrix} 4 & 2 & -2 \\ 2 & 5 & 3 \\ -2 & 3 & 14 \end{pmatrix}.$$

On vérifie les mineurs principaux :

$$\det(\mathbf{A}_1) = 4 > 0, \qquad \det(\mathbf{A}_2) = \begin{vmatrix} 4 & 2 \\ 2 & 5 \end{vmatrix} = 20-4 = 16 > 0, \qquad \det(\mathbf{A}_3) = \det(\mathbf{A}) = 144 > 0.$$

$\mathbf{A}$ est donc symétrique définie positive : la décomposition de Cholesky $\mathbf{A} = \mathbf{L}\,\mathbf{L}^\top$ existe et est unique (avec $\ell_{ii} > 0$).

On construit $\mathbf{L}$ *colonne par colonne* à l'aide des formules

$$\ell_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} \ell_{jk}^2}, \qquad \ell_{ij} = \frac{1}{\ell_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} \ell_{ik}\,\ell_{jk} \right), \quad i > j.$$

Étape 1 — traitement de la colonne 1. L'élément diagonal est

$$\ell_{11} = \sqrt{a_{11}} = \sqrt{4} = 2.$$

Les entrées sous-diagonales de la première colonne sont ensuite

$$\ell_{21} = \frac{a_{21}}{\ell_{11}} = \frac{2}{2} = 1, \qquad \ell_{31} = \frac{a_{31}}{\ell_{11}} = \frac{-2}{2} = -1.$$

La matrice partielle après la colonne 1 est :

$$\mathbf{L}^{(1)} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & ? & 0 \\ -1 & ? & ? \end{pmatrix}.$$

Étape 2 — traitement de la colonne 2. On soustrait la contribution de la colonne 1 déjà fixée :

$$\ell_{22} = \sqrt{a_{22} - \ell_{21}^2} = \sqrt{5 - 1^2} = \sqrt{4} = 2.$$

L'unique entrée sous-diagonale restante dans cette colonne est

$$\ell_{32} = \frac{a_{32} - \ell_{31}\,\ell_{21}}{\ell_{22}} = \frac{3 - (-1)(1)}{2} = \frac{4}{2} = 2.$$

La matrice partielle après la colonne 2 est :

$$\mathbf{L}^{(2)} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ -1 & 2 & ? \end{pmatrix}.$$

Étape 3 — traitement de la colonne 3. On soustrait les contributions des deux colonnes précédentes :

$$\ell_{33} = \sqrt{a_{33} - \ell_{31}^2 - \ell_{32}^2} = \sqrt{14 - (-1)^2 - 2^2} = \sqrt{14 - 1 - 4} = \sqrt{9} = 3.$$

Résultat. La décomposition de Cholesky est $\mathbf{A} = \mathbf{L}\,\mathbf{L}^\top$ avec

$$\mathbf{L} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ -1 & 2 & 3 \end{pmatrix}.$$

Vérification.

$$\mathbf{L}\,\mathbf{L}^\top = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ -1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 2 & 1 & -1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 4 & 2 & -2 \\ 2 & 5 & 3 \\ -2 & 3 & 14 \end{pmatrix} = \mathbf{A}.$$

Propriété 2.2.2: Formules de construction de la décomposition de Cholesky

Soit $\mathbf{A} \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive. On construit $\mathbf{L}$ colonne par colonne, de $j = 1$ à n, par les formules :

$$\ell_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} \ell_{jk}^2}, \qquad \ell_{ij} = \frac{1}{\ell_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} \ell_{ik}\,\ell_{jk} \right), \quad i > j,$$

avec la convention $\ell_{ij} = 0$ pour $i < j$ et $\ell_{jj} > 0$.

2.2.7 Application de la Factorisation de Cholesky

Si $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$, résoudre $\mathbf{A}x = b$ revient à résoudre **deux systèmes triangulaires successifs** :

$$\mathbf{A}x = b \iff \begin{cases} \mathbf{L}\,y = b & \text{(descente)}, \\ \mathbf{L}^\top x = y & \text{(remontée)}. \end{cases}$$

La cohérence est immédiate :

$$\mathbf{A}x = \mathbf{L}\mathbf{L}^\top x = \mathbf{L}(\mathbf{L}^\top x) = \mathbf{L}y = b.$$

Exemple 8:

Résolvons $\mathbf{A}x = b$ avec $b = \begin{pmatrix} 2 \\ 5 \\ 12 \end{pmatrix}$, en utilisant la décomposition $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ établie précédemment.

Descente — résolution de $\mathbf{L}y = b$ *:*

$$\begin{pmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ -1 & 2 & 3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 12 \end{pmatrix} \implies \begin{cases} y_1 = \dfrac{2}{2} = 1, \\ y_2 = \dfrac{1}{2}(5 - 1\cdot 1) = \dfrac{4}{2} = 2, \\ y_3 = \dfrac{1}{3}(12 - (-1)\cdot 1 - 2\cdot 2) = \dfrac{12 + 1 - 4}{3} = \dfrac{9}{3} = 3. \end{cases}$$

Remontée — résolution de $\mathbf{L}^\top x = y$ *:*

$$\begin{pmatrix} 2 & 1 & -1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \implies \begin{cases} x_3 = \dfrac{3}{3} = 1, \\ x_2 = \dfrac{1}{2}(2 - 2{\cdot}1) = 0, \\ x_1 = \dfrac{1}{2}(1 - 1{\cdot}0 + 1{\cdot}1) = \dfrac{2}{2} = 1. \end{cases}$$

La solution est donc

$$\boxed{x = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.}$$

Algorithme 5 Factorisation de Cholesky

Entrée : Matrice $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive
Sortie : Matrice L triangulaire inférieure telle que $A = LL^\top$

// Initialisation - première colonne
$\ell_{1,1} \leftarrow \sqrt{a_{1,1}}$
for $i = 2$ **à** n **do**
  $\ell_{i,1} \leftarrow \dfrac{a_{i,1}}{\ell_{1,1}}$
end
// Boucle principale
for $j = 2$ **à** $n-1$ **do**
  // Pivot diagonal
  $\ell_{j,j} \leftarrow \sqrt{a_{j,j} - \sum_{k=1}^{j-1} \ell_{j,k}^2}$
  for $i = j+1$ **à** n **do**
    // Entrées sous-diagonales de la colonne j
    $\ell_{i,j} \leftarrow \dfrac{1}{\ell_{j,j}} \left(a_{i,j} - \sum_{k=1}^{j-1} \ell_{i,k}\, \ell_{j,k} \right)$
  end
end
// Dernier pivot diagonal
$\ell_{n,n} \leftarrow \sqrt{a_{n,n} - \sum_{k=1}^{n-1} \ell_{n,k}^2}$

2.3 Méthode itératives

Pour des systèmes linéaires de grande taille, les méthodes directes (telles que l'élimination de Gauss, la factorisation de Cholesky ou LU) s'avèrent souvent trop coûteuses en termes de nombre d'opérations, de temps de calcul et d'espace mémoire.

L'idée fondamentale des méthodes itératives est de **ne plus chercher à résoudre exactement** le système en un nombre fini d'étapes. On cherche plutôt à approcher sa solution par

itérations successives, où chaque itération consiste à résoudre un système linéaire beaucoup moins coûteux à l'aide d'une formule de récurrence simple.

2.3.1 Principe Général

Soit le système linéaire

$$Ax = b, \tag{2.4}$$

où $A \in \mathcal{M}_n(\mathbb{C})$ est inversible et $x, b \in \mathbb{C}^n$.

Définition 2.3.1: Décomposition de la matrice

Le principe des méthodes itératives est d'écrire A comme la différence de deux matrices :

$$A = M - N,$$

où M est une matrice **inversible** choisie de sorte que le système linéaire $My = c$ soit **facile à résoudre** (faible coût de calcul). Typiquement, M sera diagonale, triangulaire, ou diagonale/triangulaire par blocs.

On approche la solution de (2.4) par la suite $\left(x^{(k)}\right)_{k\geq 0}$ définie par la **formule de récurrence** :

$$\boxed{x^{(k+1)} = M^{-1}\left(b + N\,x^{(k)}\right),} \tag{2.5}$$

à partir d'un vecteur initial $x^{(0)}$ quelconque.

Remarque 4: Mise en œuvre pratique

On ne calcule **jamais** explicitement M^{-1}. On résout à chaque itération le système triangulaire (ou diagonal) $M\,x^{(k+1)} = b + N\,x^{(k)}$, ce qui est peu coûteux.

Proposition 2.3.1: Cohérence de la méthode

Si la suite $\left(x^{(k)}\right)$ converge vers une limite ℓ, alors ℓ est solution du système (2.4).

Démonstration. En passant à la limite dans (2.5), on obtient $M\,\ell = b + N\,\ell$, soit $A\,\ell = (M - N)\ell = b$. L'unicité de la solution assure $\ell = x$. ■

2.3.2 Matrice d'Itération et Erreur

Définition 2.3.2: Matrice d'itération

La matrice $\mathcal{L} = M^{-1}N$ est appelée **matrice d'itération** de la méthode. On définit également l'erreur à l'étape k par $e^{(k)} = x - x^{(k)}$.

En soustrayant (2.5) de $x = M^{-1}(b+Nx)$, on obtient la relation de récurrence sur l'erreur :

$$\begin{aligned} e^{(k+1)} &= x - x^{(k+1)}, \\ &= M^{-1}(b + Nx) - M^{-1}(b + Nx^{(k)}), \\ &= M^{-1}N(x - x^{(k)}), \\ &= \mathcal{L}\, e^{(k)}, \\ &\vdots \\ &= \mathcal{L}^{k+1}\, e^{(0)}. \end{aligned}$$

Théorème 2.3.1: Rayon spectral

La suite $\left(x^{(k)}\right)$ converge vers x pour *tout* choix de $x^{(0)}$ si et seulement si

$$\rho(\mathcal{L}) = \rho(M^{-1}N) < 1,$$

où $\rho(\mathcal{L}) = \max\left\{|\lambda| : \lambda \text{ valeur propre de } \mathcal{L}\right\}$ est le **rayon spectral** de $\mathcal{L}$.

Démonstration. La suite converge pour tout $e^{(0)}$ si et seulement si $\mathcal{L}^n \to 0$. Or, d'après le résultat classique d'algèbre linéaire numérique, $\mathcal{L}^n \to 0$ si et seulement si $\rho(\mathcal{L}) < 1$. ■

Théorème 2.3.2: Estimation de la vitesse de convergence

Pour toute norme matricielle subordonnée $\|\cdot\|$, on a l'estimation

$$\left\|e^{(k)}\right\| \leq \|\mathcal{L}\|^k \left\|e^{(0)}\right\|.$$

Plus le rayon spectral $\rho(\mathcal{L})$ est petit, plus la convergence est rapide.

Remarque 5: Critère pratique d'arrêt

On arrête les itérations lorsque le résidu relatif vérifie :

$$\frac{\|b - Ax^{(k+1)}\|}{\|b\|} < \varepsilon, \quad \text{ou} \quad \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} < \varepsilon.$$

En pratique, on prend $\varepsilon \in [10^{-6}, 10^{-10}]$ selon la précision souhaitée.

2.4 La Méthode de Jacobi

2.4.1 Construction de la méthode

Soit $A = (a_{ij}) \in \mathcal{M}_n(\mathbb{R})$ inversible, avec $a_{ii} \neq 0$ pour tout i. On décompose A en :

$$A = D - E - F,$$

où $D = \text{diag}(a_{11}, a_{22}, \ldots, a_{nn})$ est la partie diagonale, $-E$ est la partie triangulaire inférieure stricte, et $-F$ est la partie triangulaire supérieure stricte.

Définition 2.4.1: Méthode de Jacobi

La **méthode de Jacobi** correspond au choix

$$M = D, \qquad N = E + F = D - A,$$

ce qui donne la formule d'itération :

$$\boxed{x^{(k+1)} = D^{-1}(E+F)\,x^{(k)} + D^{-1}b.}$$

La matrice d'itération de Jacobi est $\mathcal{L}_J = D^{-1}(E+F) = I - D^{-1}A$.

2.4.2 Formule composante par composante

Pour la i-ème composante, la mise à jour s'écrit explicitement :

$$\boxed{x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1\\ j\neq i}}^{n} a_{ij}\,x_j^{(k)}\right), \quad i = 1,\dots,n.} \tag{2.6}$$

Remarque 6: Parallélisme

À chaque itération, **toutes les composantes** $x_i^{(k+1)}$ ne dépendent que de $x^{(k)}$. Les calculs sont donc **indépendants** et peuvent être effectués en parallèle, ce qui est un avantage sur les architectures multi-cœurs.

2.4.3 Convergence de Jacobi

Définition 2.4.2: Matrice à diagonale strictement dominante

Une matrice $A = (a_{ij})$ est dite **à diagonale strictement dominante** (en lignes) si

$$|a_{ii}| > \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}|, \quad \forall\, i = 1,\dots,n.$$

Théorème 2.4.1: Convergence de Jacobi

Si A est à diagonale strictement dominante, alors la méthode de Jacobi converge pour tout $x^{(0)}$.

Démonstration. Il suffit de montrer que $\rho(\mathcal{L}_J) < 1$. On a $\|\mathcal{L}_J\|_\infty = \max_i \sum_{j\neq i} \frac{|a_{ij}|}{|a_{ii}|} < 1$ par hypothèse. Or $\rho(\mathcal{L}_J) \leq \|\mathcal{L}_J\|_\infty < 1$. ■

Corollaire 2.4.1: Cas des matrices tridiagonales

Si A est tridiagonale à diagonale strictement dominante, Jacobi converge et son rayon spectral vérifie $\rho(\mathcal{L}_J) < 1$.

2.4.4 Exemple numérique

Exemple 9: Application de Jacobi sur un système 3×3

Considérons le système $Ax = b$ avec :

$$A = \begin{pmatrix} 4 & -1 & 1 \\ 2 & 5 & -1 \\ -1 & 2 & 6 \end{pmatrix}, \qquad b = \begin{pmatrix} 4 \\ 6 \\ 7 \end{pmatrix}.$$

Vérification de la dominance diagonale :

- Ligne 1 : $|4| = 4 > |-1| + |1| = 2$ ✓
- Ligne 2 : $|5| = 5 > |2| + |-1| = 3$ ✓
- Ligne 3 : $|6| = 6 > |-1| + |2| = 3$ ✓

La matrice est à diagonale strictement dominante, donc **Jacobi converge**.

Formules de récurrence de Jacobi :

$$\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{4}\left(4 + x_2^{(k)} - x_3^{(k)}\right), \\ x_2^{(k+1)} &= \tfrac{1}{5}\left(6 - 2x_1^{(k)} + x_3^{(k)}\right), \\ x_3^{(k+1)} &= \tfrac{1}{6}\left(7 + x_1^{(k)} - 2x_2^{(k)}\right). \end{aligned}$$

Itérations à partir de $x^{(0)} = (0, 0, 0)^\top$:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\left\|e^{(k)}\right\|_\infty$
0	0.000000	0.000000	0.000000	1.000000
1	1.000000	1.200000	1.166667	0.343434
2	1.008333	0.946667	0.766667	0.233333
3	1.044999	0.990667	1.019167	0.054167
5	1.001412	0.998845	0.999623	0.002155
10	1.000002	0.999998	1.000001	0.000003
⋮	⋮	⋮	⋮	
∞	1.000000	1.000000	1.000000	0.000000

La solution exacte est $x^* = (1, 1, 1)^\top$.

2.5 La Méthode de Gauss-Seidel

2.5.1 Construction de la méthode

Définition 2.5.1: Méthode de Gauss-Seidel

La **méthode de Gauss-Seidel** correspond au choix

$$M = D - E, \qquad N = F,$$

ce qui donne la formule d'itération :

$$\boxed{x^{(k+1)} = (D-E)^{-1}F\,x^{(k)} + (D-E)^{-1}b.}$$

La matrice d'itération de Gauss-Seidel est $\mathcal{L}_{GS} = (D-E)^{-1}F$.

2.5.2 Formule composante par composante

L'idée clé est d'utiliser **immédiatement** les nouvelles valeurs $x_j^{(n+1)}$ dès qu'elles sont calculées :

$$\boxed{x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}\,x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}\,x_j^{(k)}\right), \quad i = 1, \ldots, n.} \tag{2.7}$$

Remarque 7: Différence avec Jacobi

Contrairement à Jacobi qui utilise uniquement $x^{(k)}$, Gauss-Seidel utilise les composantes **déjà mises à jour** dans la même itération. Cela accélère souvent la convergence, mais rend les calculs **séquentiels** (non parallélisables directement).

2.5.3 Convergence de Gauss-Seidel

Théorème 2.5.1: Convergence de G-S — Matrices SDP

Si A est **symétrique définie positive** (SDP), alors la méthode de Gauss-Seidel converge pour tout $x^{(0)}$.

Théorème 2.5.2: Convergence de G-S — Diagonale strictement dominante

Si A est à diagonale strictement dominante, alors la méthode de Gauss-Seidel converge pour tout $x^{(0)}$.

Remarque 8: Comparaison des rayons spectraux

Pour les matrices **tridiagonales à diagonale strictement dominante**, on dispose du résultat remarquable :

$$\rho(\mathcal{L}_{GS}) = \left[\rho(\mathcal{L}_J)\right]^2.$$

Gauss-Seidel converge donc **deux fois plus vite** que Jacobi (en termes d'itérations logarithmiques) dans ce cas.

2.5.4 Exemple numérique

Exemple 10: Reprise du système 3×3 avec Gauss-Seidel

On reprend le même système. Les formules de Gauss-Seidel sont :

$$\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{4}\left(4 + x_2^{(k)} - x_3^{(k)}\right), \\ x_2^{(k+1)} &= \tfrac{1}{5}\left(6 - 2\,\mathbf{x}_1^{(k+1)} + x_3^{(k)}\right) && \leftarrow \text{utilise } x_1^{(k+1)}, \\ x_3^{(k+1)} &= \tfrac{1}{6}\left(7 + \mathbf{x}_1^{(k+1)} - 2\,\mathbf{x}_2^{(k+1)}\right) && \leftarrow \text{utilise } x_1^{(k+1)},\ x_2^{(k+1)}. \end{aligned}$$

Itérations à partir de $x^{(0)} = (0, 0, 0)^\top$:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\left\|e^{(k)}\right\|_\infty$
0	0.000000	0.000000	0.000000	1.000000
1	1.000000	0.800000	0.866667	0.200000
2	0.983333	1.006667	0.995000	0.016667
3	1.002917	0.999167	0.999097	0.002917
5	1.000003	0.999999	1.000000	0.000003

Gauss-Seidel atteint la même précision en **5 itérations** contre 10 pour Jacobi, confirmant la relation $\rho(\mathcal{L}_{GS}) \approx \rho(\mathcal{L}_J)^2$.

2.6 Comparaison et Critères de Choix

2.6.1 Tableau récapitulatif

Critère	**Jacobi**	**Gauss-Seidel**
Décomposition	$M = D,\ N = E + F$	$M = D - E,\ N = F$
Matrice d'itération	$\mathcal{L}_J = D^{-1}(E + F)$	$\mathcal{L}_{GS} = (D - E)^{-1}F$
Convergence (DSD)	Oui	Oui
Convergence (SDP)	Pas nécessairement	Oui (Théorème)

2.6.2 Exemple : Cas où Jacobi et Gauss-Seidel diverge

Exemple 11: Divergence possible de Jacobi

Considérons la matrice

$$A = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

La matrice de Jacobi est $\mathcal{L}_J = -D^{-1}(E+F) = \begin{pmatrix} 0 & -2 \\ 2 & 0 \end{pmatrix}$, de valeurs propres $\pm 2i$, donc $\rho(\mathcal{L}_J) = 2 > 1$: **Jacobi diverge**.

En revanche, la matrice de Gauss-Seidel est $\mathcal{L}_{GS} = (D-E)^{-1}F$. On calcule :

$$D - E = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}, \quad F = \begin{pmatrix} 0 & -2 \\ 0 & 0 \end{pmatrix},$$

$$\mathcal{L}_{GS} = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 & -2 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -2 \\ 0 & 4 \end{pmatrix}.$$

Les valeurs propres de $\mathcal{L}_{GS}$ sont 0 et 4, donc $\rho(\mathcal{L}_{GS}) = 4 > 1$: **Gauss-Seidel diverge aussi** ici (la matrice n'est ni DSD ni SDP). Cet exemple illustre qu'aucune convergence n'est garantie sans hypothèse sur A.

Bibliographie

[1] **Bannouna**, J.
Notes de cours : Analyse Numérique.
Université Sidi Mohamed Ben Abdellah, Faculté des Sciences Dhar Mahraz, Fès. Département de Mathématiques — Deuxième année SMA (S4), 2015–2016.

[2] **Pujo-Menjouet**, L.
Analyse Numérique — Troisième année de Licence.
Université Claude Bernard Lyon I, Licence Sciences, Technologies & Santé, Spécialité Mathématiques. 43, boulevard du 11 novembre 1918, 69622 Villeurbanne, France.
pujo@math.univ-lyon1.fr

[3] **Gallouët**, Thierry & **Herbin**, Raphaèle.
Cours d'Analyse Numérique.
Université Aix-Marseille, Licence de Mathématiques. 4 août 2023.

[4] **Manceau**, David.
Analyse Numérique Matricielle.
Licence 3 Mathématiques.

Les références sont présentées par ordre d'apparition dans le texte.