

Chapitre 1

Systèmes linéaires

1.1 Introduction

Un système d'équations linéaires est un ensemble d'équations portant sur les mêmes inconnues.

En général, un système de m équations linéaires à n inconnues peut être écrit sous la forme suivante :

$$\left\{ \begin{array}{lcl} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n & = & b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n & = & b_2 \\ \vdots & & \vdots \quad \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n & = & b_m \end{array} \right. \quad (1.1)$$

Les données sont les coefficients $a_{i,j}$ ($1 \leq i \leq m$ et $1 \leq j \leq n$) du système qui appartiennent à un corps \mathbb{K} avec $\mathbb{K} = \mathbb{R}$ ou \mathbb{C} ainsi que les coefficients du second membre b_1, \dots, b_m .

Les inconnues sont x_1, x_2, \dots, x_n qui appartiennent à \mathbb{K} .

On peut utiliser la notation matricielle, qui beaucoup plus pratique et surtout plus compacte. On écrit alors le système (1.1) sous la forme matricielle

$$Ax = b \quad (1.2)$$

où A est une matrice de taille $m \times n$, x est un vecteur de taille n et b est un vecteur de taille m .

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

Remarque 1.1.1

- Un système est dit **homogène** si son second membre est nul c'est-à-dire si tous les b_i sont nuls
- Si $m > n$ on dit que le système est **surdéterminer**.
- Si $m = n$ alors la matrice A est carrée et d'ordre n

Remarque 1.1.2

Pour résoudre un système d'équations linéaires on cherche à remplacer le système initial par un autre, plus simple, ayant la même solution, sachant que la résolution de système (1.2) est très simple lorsque :

- A est diagonale ;
- A est orthogonale ;
- A est tridiagonale ;
- A est triangulaire supérieure ou inférieure ;
- A est creuse (en général).

Dans tout ce qui suit, on considérera une matrice carré $A \in \mathcal{M}_n(\mathbb{R})$ et un vecteur b contenant n réels.

Seulement trois cas sont possibles pour le système linéaire (1.2) :

- Le système n'a pas de solution.
- Le système a une solution unique.
- Le système a une infinité de solutions.

Exemple 1.1.3

Les systèmes suivants ont-ils une solution unique, une infinité de solution ou pas de solution ?

$$\begin{cases} 2x_1 + 6x_2 = 4 \\ -4x_1 - 12x_2 = -8 \end{cases}$$

Ce système a une infinite de solutions : $(x_1, x_2) = (-1 - 3\lambda, 1 + \lambda)$, $\lambda \in \mathbb{R}$.

$$\begin{cases} x_1 + 3x_2 = 7 \\ 2x_1 - x_2 = 0 \end{cases}$$

Ce système admet une unique solution $(x_1, x_2) = (1, 2)$.

$$\begin{cases} 4x_1 - 3x_2 = 14 \\ 16x_1 - 12x_2 = 2 \end{cases}$$

Ce système n'a pas de solution.

1.2 Résolution de systèmes linéaires

Si la matrice A est inversible alors le système linéaire (1.2) admet une unique solution $x = A^{-1}b$ où A^{-1} est la matrice inverse de A . Ainsi théoriquement le problème revient à calculer A^{-1} . Mais en pratique ce calcul est difficile. Il existe plusieurs méthodes classiques pour résoudre le système linéaire (1.2) sans calculer A^{-1} . Le choix de la méthode dépend

fortement du type (forme) de la matrice A .

Les méthodes de résolution des systèmes linéaires (1.2) se groupent dans deux grandes catégories :

1. Les méthodes directes qui permettent d'obtenir la solution en un nombre fini d'opérations soit par triangulation ou soit par décomposition de la matrice A . Les principales méthodes sont :
 - L'élimination de Gauss,
 - La décomposition LU ,
 - La décomposition de Cholesky,
2. Les méthodes itératives qui consistent à construire une suite $(x_n)_{n \in \mathbb{N}}$ qui converge vers la solution. Les principales méthodes sont :
 - Méthode de Jacobi,
 - Méthode de Gauss-Seidel,
 - Méthode de Relaxation.

1.3 Méthodes directes

1.3.1 Principe des méthodes directes

Nous nous ramenons en un nombre fini d'opérations à un système simple à résoudre en général triangulaire (parfois une matrice orthogonale). Ces méthodes reviennent souvent à écrire une décomposition de la matrice $A = BC$, où B et C ont des propriétés particulières. Ainsi résoudre

$$Ax = b$$

équivaut à résoudre

$$B(Cx) = b$$

ce qui est équivalent à résoudre

$$By = b,$$

puis

$$Cx = y.$$

1.3.2 Méthode de résolution pour les systèmes triangulaires inférieurs

Définition 1.3.3

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 :

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

Remarque 1.3.4

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 $i = 1, \dots, n$.

1.3.4.1 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 $i = 1, 2, 3$.

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

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

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

```

 $x_1 = b_1/l_{1,1};$ 
Pour i de 2 à n faire
     $x_i = \left( b_i - \sum_{j=1}^{i-1} l_{i,j} x_j \right) / l_{i,i}$ 
Fin Pour

```

Algorithme 1 – Algorithme de descente d'un système triangulaire inférieur

Exemple 1.3.5

Considérons le système triangulaire inférieure 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

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

1.3.6 Méthode de résolution pour les systèmes triangulaires supérieurs

Définition 1.3.7

Une matrice carrée $(a_{i,j})_{1 \leq i,j \leq n}$ est dite triangulaire supérieure si tous les éléments situés au-dessous 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 supérieure a la forme type

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

Remarque 1.3.8

Une matrice triangulaire supérieure est simplement la transposée d'une matrice triangulaire inférieure. Alors $\det(A) = \det({}^t A) = \prod_{i=1}^n a_{i,i}$.

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

1.3.8.1 Algorithme de résolution pour les systèmes triangulaires supérieurs

Considérons un système inversible 3×3 triangulaire inférieur

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

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 x_i pour $i = 1, 2, 3$.

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

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

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

$$x_n = b_n/l_{n,n};$$

Pour i de n-1 à 1 faire

$$x_i = (b_i - \sum_{j=i+1}^n u_{i,j} x_j)/u_{i,i}$$

Fin Pour

Algorithme 2 – *Algorithme de remontée d'un système triangulaire supérieur*

Exemple 1.3.9

Considérons le système triangulaire supérieur suivant

$$\begin{pmatrix} 2 & 5 & 3 \\ 0 & 4 & 2 \\ 0 & 0 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 49 \\ 30 \\ 21 \end{pmatrix}$$

En utilisant l'algorithme de remontée, on obtient

$$\begin{cases} x_3 = b_3/u_{3,3} = 21/7 = 3 \\ x_2 = (b_2 - u_{2,3} x_3)/u_{2,2} = (30 - 2 \times 3)/4 = 6 \\ x_1 = (b_1 - u_{1,2} x_2 - u_{2,3} x_3)/u_{1,1} = (49 - 5 \times 6 - 3 \times 3)/2 = 5 \end{cases}$$

1.3.10 Complexité des algorithmes de descente et remontée**Proposition 1.3.11**

La résolution d'un système d'équations linéaires triangulaire se fait en n^2 opérations à virgule flottante.

- $\frac{n(n-1)}{2}$ multiplications
- $\frac{n(n-1)}{2}$ additions
- n divisions
- Au total n^2 opérations.

Preuve

Calculer x_n nécessite 1 opération (division), calculer x_{n-1} nécessite 3 opérations (une multiplication, une soustraction et une division), et calculer x_i nécessite $2(n-i) + 1$ opérations ($n-i$) multiplications, $(n-i-1)$ additions, 1 soustraction et 1 division)

Au total $\sum_{i=1}^n (2(n-i) + 1) = n^2$

1.3.12 Cas particulier matrice diagonale

Exercice 1.3.13

Soit $D = (d_{i,i})_{i=1,\dots,n}$ une matrice diagonale d'ordre $n > 0$.

$$D = \begin{pmatrix} d_{1,1} & & & & \\ & \ddots & & & 0 \\ & & d_{i,i} & & \\ 0 & & & \ddots & \\ & & & & d_{n,n} \end{pmatrix}$$

- Donner une condition nécessaire et suffisante pour que D soit inversible.
- Donner un algorithme qui permet de résoudre l'équation d'inconnue x :

$$D x = b.$$

1.3.14 Méthode d'élimination de Gauss

La méthode d'élimination de Gauss a pour but de transformer le système $A x = b$ en un système équivalent (c'est-à-dire ayant la même solution) de la forme $U x = \tilde{b}$, où U est une matrice triangulaire supérieure et \tilde{b} est un second membre convenablement modifié. Ce dernier système peut être alors résolu par la méthode de remontée.

Au cours de la transformation, on utilise essentiellement la propriété selon laquelle on ne change pas la solution du système quand on ajoute à une équation donnée une combinaison linéaire des autres équations.

Propriétés 1.3.15

On ne change pas la solution d'un système linéaire lorsque :

- on permute deux lignes,
- on permute deux colonnes,
- on multiplie une ligne par un réel non nul,
- on ajoute une ligne à une autre.

1.3.15.1 Méthode de Gauss sans pivot

Considérons une matrice invisible $A \in \mathcal{M}_n(\mathbb{R})$ dont le terme diagonal $a_{1,1}$ est supposé non nul. Le nombre $a_{1,1}$ est le premier pivot de l'élimination de Gauss. On pose $A^{(1)} = A$ et $b^{(1)} = b$ et on introduit les multiplicateurs

$$m_{i,1} = \frac{a_{i,1}^{(1)}}{a_{1,1}^{(1)}}, \quad i = 2, 3, \dots, n.$$

où les $a_{i,j}^{(1)}$ désignent les éléments de $A^{(1)}$. On peut éliminer l'inconnue x_1 des lignes $i = 2, \dots, n$ en leur retranchant $m_{i,1}$ fois la première ligne et en faisant de même pour le membre de droit. On définit alors

$$\begin{aligned} a_{i,j}^{(2)} &= a_{i,j}^{(1)} - m_{i,1} a_{1,j}^{(1)}, \quad i, j = 2, \dots, n, \\ b_i^{(2)} &= b_i^{(1)} - m_{i,1} b_1^{(1)}, \quad i = 2, \dots, n. \end{aligned}$$

où les $b_i^{(1)}$ sont les composantes de $b^{(1)}$ et on obtient un nouveau système de la forme

$$\left(\begin{array}{cccccc} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots & a_{1,n}^{(1)} & & \\ 0 & a_{2,2}^{(2)} & \cdots & a_{2,n}^{(2)} & & \\ \vdots & \vdots & \ddots & \vdots & & \\ 0 & a_{n,2}^{(2)} & \cdots & a_{n,n}^{(2)} & & \end{array} \right) \left(\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right) = \left(\begin{array}{c} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{array} \right)$$

que l'on note $A^{(2)}x = b^{(2)}$ et qui est équivalent au système de départ.

On peut à nouveau transformer ce système de façon à éliminer l'inconnue x_2 des lignes $3, \dots, n$. En poursuivant ainsi, on obtient une suite finie de systèmes

$$A^{(k)}x = b^{(k)}, \quad 1 \leq k \leq n,$$

où, pour $k \geq 2$, la matrice $A^{(k)}$ est de la forme suivante

$$\left(\begin{array}{cccccc} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots & a_{1,k}^{(1)} & \cdots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & \cdots & a_{2,k}^{(2)} & \cdots & a_{2,n}^{(2)} \\ \vdots & \ddots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{k,k}^{(k)} & \cdots & a_{n,n}^{(1)} \end{array} \right)$$

où on a supposé $a_{i,i}^{(i)} \neq 0$ pour $i = 1, \dots, k-1$. Il est clair que pour $k = n$ on obtient alors le système triangulaire supérieur $A^{(n)}x = b^{(n)}$ suivant

$$\left(\begin{array}{cccccc} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots & a_{1,n}^{(1)} & & \\ 0 & a_{2,2}^{(2)} & \cdots & a_{2,n}^{(2)} & & \\ \vdots & \ddots & \ddots & \vdots & & \\ 0 & \cdots & 0 & a_{n,n}^{(n)} & & \end{array} \right) \left(\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right) = \left(\begin{array}{c} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(n)} \end{array} \right)$$

Pour être consistant avec les notations introduites précédemment, on note U la matrice triangulaire supérieur $A^{(n)}$. Les termes $a_{k,k}^{(k)}$ sont appelés pivots et doivent être évidemment non

nuls pour $k = 1, \dots, n - 1$.

Afin d'expliciter les formules permettant de passer du $k^{\text{ième}}$ système au $(k + 1)^{\text{ième}}$, pour $k = 1, \dots, n - 1$, on suppose que $a_{k,k}^{(k)} \neq 0$ et on définit les multiplicateurs

$$m_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}, \quad i = k + 1, \dots, n$$

On pose alors

$$\begin{aligned} a_{i,j}^{(k+1)} &= a_{i,j}^{(k)} - m_{i,k} a_{k,j}^{(k)}, \quad i, j = k + 1, \dots, n, \\ b_i^{(k+1)} &= b_i^{(k)} - m_{i,k} b_k^{(k)}, \quad i = k + 1, \dots, n. \end{aligned}$$

Remarque 1.3.16

Le procédé suppose que tous les $a_{k,k}^{(k)} \neq 0$. Si à une étape k on a $a_{k,k}^{(k)} = 0$ et s'il y a au moins un des $a_{i,k}^{(k)} \neq 0$, $i = k + 1, \dots, n$, on permute les lignes k et i et on continue, sinon ça voudrait dire que la matrice A n'est pas inversible.

Pour bien comprendre la méthode, on considère l'exemple suivant :

Exemple 1.3.17

$$\begin{pmatrix} 2 & -1 & 2 \\ 4 & 3 & -3 \\ -2 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 15 \\ -25 \\ -4 \end{pmatrix}$$

On pose $A^{(1)} = A$ et $b^{(1)} = b$, i.e.

$$A^{(1)} = \begin{pmatrix} 2 & -1 & 2 \\ 4 & 3 & -3 \\ -2 & 2 & 1 \end{pmatrix} \quad b^{(1)} = \begin{pmatrix} 15 \\ -25 \\ -4 \end{pmatrix}$$

On note $a_{i,j}^{(1)}$, $1 \leq j \leq 3$ l'élément (i, j) de la matrice A . Le premier pivot de l'élimination de Gauss est donc $a_{1,1}^{(1)} = 2$. On effectue alors pour toute ligne L_i pour $i = 2, 3$

$$L_i \leftarrow L_i - \frac{a_{i,1}^{(1)}}{a_{1,1}^{(1)}} L_1$$

Dans notre exemple on obtient

$$A^{(2)} = \begin{pmatrix} 2 & -1 & 2 \\ 0 & 5 & -7 \\ 0 & 1 & 3 \end{pmatrix} \quad b^{(2)} = \begin{pmatrix} 15 \\ -55 \\ 11 \end{pmatrix}$$

On refait la même procédure, dans cette étape le pivot est $a_{2,2}^{(2)} = 5$. On effectue

$$L_3 \leftarrow L_3 - \frac{1}{5} L_2$$

On obtient finalement

$$A^{(3)} = \begin{pmatrix} 2 & -1 & 2 \\ 0 & 5 & -7 \\ 0 & 0 & \frac{22}{5} \end{pmatrix} \quad b^{(3)} = \begin{pmatrix} 15 \\ -55 \\ 22 \end{pmatrix}$$

La matrice obtenue est maintenant triangulaire supérieure, et on peut résoudre le système $A^{(3)}x = b^{(3)}$ par l'algorithme de remontée, on trouve

$$\begin{cases} x_1 = 1/2 \\ x_2 = -4 \\ x_3 = 5 \end{cases}$$

On donne maintenant l'algorithme d'élimination de Gauss sans pivot. On admettra que les termes (pivots) $a_{k,k}^{(k)}$ doivent être évidemment non nuls pour $k = 1, \dots, n$.

```

Pour k de 1 à n-1 faire
  Pour i de k+1 à n faire
     $m_{i,k} = a_{i,k}/a_{k,k}$ 
     $b_i = b_i - m_{i,k}b_k$ 
     $a_{i,k} = 0$ 
    Pour j de k+1 à n faire
       $a_{i,j} = a_{i,j} - m_{i,k}a_{k,j}$ 
  Fin Pour
Fin Pour
Fin Pour
→ Résolution du système triangulaire sup.
```

Algorithme 3 – *Algorithme d'élimination de Gauss*

Exercice 1.3.18

En appliquant l'algorithme de Gauss, résoudre le système linéaire suivant

$$(S) \begin{cases} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 = \frac{11}{6} \\ \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 = \frac{13}{12} \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 = \frac{47}{60} \end{cases}$$

La solution du système (S) est $x =^t (1, 1, 1)$.

A la première étape on cherche les multiplicateurs $m_{2,1} = \frac{1}{2}$ et $m_{3,1} = \frac{1}{3}$, et on soustrait de la deuxième (resp. troisième) équation la première ligne multipliée par $m_{2,1}$ (resp. $m_{3,1}$). On obtient le système équivalent

$$(S) \begin{cases} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 = \frac{11}{6} \\ 0 + \frac{1}{12}x_2 + \frac{1}{12}x_3 = \frac{1}{6} \\ 0 + \frac{1}{12}x_2 + \frac{4}{45}x_3 = \frac{31}{180} \end{cases}$$

Si on soustrait à présent la troisième ligne la seconde multipliée par $m_{3,2} = 1$, on obtient le système triangulaire supérieur suivant

$$(S) \begin{cases} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 = \frac{11}{6} \\ 0 + \frac{1}{3}x_2 + \frac{1}{4}x_3 = \frac{13}{12} \\ 0 + 0 + \frac{1}{180}x_3 = \frac{1}{180} \end{cases}$$

à partir duquel on calcule immédiatement $x_3 = 1$ et, par substitution (remontée), les autres inconnus $x_1 = x_2 = 1$.

1.3.18.1 Méthode de Gauss avec pivot

Le choix du pivot permet de minimiser les erreurs d'arrondis :

- si un pivot est nul, on permute deux lignes
- si tous les pivots restant sont nuls \Rightarrow la matrice est singulière (i.e. le système d'équations n'admet pas de solution unique)

Pour minimiser les erreurs d'arrondis, on choisit le plus grand pivot possible (en valeur absolue) et donc on permute les lignes (voir les colonnes associées) c'est la stratégie du pivot maximal (partiel (lignes) ou total).

La méthode de Gauss avec pivot consiste à choisir à l'étape k : $a_{k,k}^{(k)}$ tel que

$$|a_{k,k}^{(k)}| = \max_{k \leq i \leq n} |a_{i,k}^{(k)}|$$

Remarque 1.3.19 (Influence du pivotage sur la précision)

Soit à résoudre le système

$$\begin{cases} 10^{-10}x_1 + x_2 = 1, \\ x_1 - x_2 = 0. \end{cases}$$

La solution théorique est $x_1 = x_2 = \frac{1}{1+10^{-10}} = 0.9999999999$. Cependant, la résolution de ce système par la méthode de Gauss donne des résultats différents selon qu'on l'applique avec ou sans pivot.

Trouver x_1 et x_2 en gardant 10 chiffres significatifs après la virgule.

Que se passe t'il si on prend le système à l'envers?

1. Si on applique la méthode de Gauss **sans pivot** on obtient

$$m_{2,1} = \frac{a_{2,1}^{(1)}}{a_{1,1}^{(1)}} = \frac{1}{10^{-10}} = 10^{10}$$

et

$$\begin{cases} 10^{-10}x_1 + x_2 = 1, \\ (-1 - 10^{10})x_2 = -10^{10}. \end{cases}$$

qui donne pour solution approchée $x_2 = \frac{-10^{10}}{-1-10^{10}} = 0.9999999999$ et $x_1 = \frac{1-x_2}{10^{-10}} = 1.000000082740371$.

2. Si on adopte la stratégie du pivot partiel qui consiste à mettre en première ligne celle dont le coefficient de x_1 est le plus grand en module alors on permute les lignes pour obtenir le système

$$\begin{cases} x_1 - x_2 = 0, \\ 10^{-10}x_1 + x_2 = 1. \end{cases}$$

Pour lequel $m_{2,1} = \frac{a_{2,1}^{(1)}}{a_{1,1}^{(1)}} = \frac{10^{-10}}{1} = 10^{-10}$ et qui conduit à la solution approchée : $x_2 = \frac{1}{1+10^{-10}} = 0.9999999999$ et $x_1 = 0.9999999999$.

Exercice 1.3.20

Appliquer la méthode de Gauss avec pivot à l'Exemple 6.8.4.

1.3.20.1 Application : calcul de A^{-1}

On considère la matrice $A = \begin{pmatrix} -1 & -2 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix}$.

A l'aide de l'algorithme d'élimination de Gauss sans pivot (Gauss-Jordan) calculer A^{-1} .

Phase 1 : \downarrow

$$\left| \begin{array}{ccc|ccc} -1 & -2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 1 \end{array} \right| \quad \left| \begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right|$$

$$\left| \begin{array}{ccc|ccc} -1 & -2 & 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 1 & 1 & 0 \\ 0 & -3 & 1 & 1 & 0 & 1 \end{array} \right| \quad \left| \begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right|$$

ligne 2 = ligne 2 + 1*ligne 1
ligne 3 = ligne 3 + 1*ligne 1

$$\left(\begin{array}{ccc|ccc} -1 & -2 & 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 2 & 2 & 3 & 1 \end{array} \right) \quad \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 3 & 1 \\ 2 & 3 & 1 & 2 & 3 & 1 \end{array} \right)$$

ligne 3 = ligne 3 - 3*ligne 2

$$\left(\begin{array}{ccc|ccc} 1 & 2 & -1 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 1 & 1 & \frac{3}{2} & \frac{-1}{2} \end{array} \right) \quad \left(\begin{array}{ccc|ccc} -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & -1 & 0 & -1 & -1 & 0 \\ 1 & \frac{3}{2} & \frac{-1}{2} & 1 & \frac{3}{2} & \frac{-1}{2} \end{array} \right)$$

ligne 1 = -1*ligne 1
ligne 2 = -1*ligne 2
ligne 3 = (-1/2)*ligne 3

Phase 2 : ↑

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 0 & 0 & \frac{3}{2} & \frac{-1}{2} \\ 0 & 1 & 0 & 0 & \frac{1}{2} & \frac{-1}{2} \\ 0 & 0 & 1 & 1 & \frac{3}{2} & \frac{-1}{2} \end{array} \right) \quad \left(\begin{array}{ccc|ccc} 0 & \frac{3}{2} & \frac{-1}{2} & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{-1}{2} & 0 & \frac{1}{2} & \frac{-1}{2} \\ 1 & \frac{3}{2} & \frac{-1}{2} & 1 & \frac{3}{2} & \frac{-1}{2} \end{array} \right)$$

ligne 1 = ligne 1 + 1*ligne 3
ligne 2 = ligne 2 + 1*ligne 3

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 & 0 & \frac{1}{2} & \frac{-1}{2} \\ 0 & 0 & 1 & 1 & \frac{3}{2} & \frac{-1}{2} \end{array} \right)$$

ligne 1 = ligne 1 - 2*ligne 2

Ainsi $A^{-1} = \frac{1}{2} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & -1 \\ 2 & 3 & -1 \end{pmatrix}$

Vérifier que $A * A^{-1} = Id$.

1.3.21 Méthode par factorisation LU

Le principe de la méthode est de se ramener à deux systèmes triangulaires.

On cherche à décomposer la matrice A en deux matrices L (triangulaire inférieure) et U (triangulaire supérieure) telles que le produit LU soit égal à A .

Si la matrice A peut s'écrire sous la forme $A = LU$, où L et U sont des matrices triangulaires inférieure et supérieure respectivement, alors le système $Ax = b$ peut se décomposer en deux sous-systèmes $Ly = b$ et $Ux = y$.

Les matrices L et U étant triangulaires, la résolution de chacun de ces deux sous-systèmes est immédiate :

- $Ly = b$: on calcule y_1, y_2, \dots, y_n (Algorithme de descente),
- $Ux = y$: on calcule x_n, x_{n-1}, \dots, x_1 (Algorithme de remontée).

Il reste maintenant à savoir comment faire cette décomposition LU .

Commençons par un résultat théorique

Théorème 1.3.22

Soit A une matrice inversible d'ordre n dont les mineurs principaux sont non nuls. Alors il existe une unique matrice L triangulaire inférieure avec des 1 sur la diagonale, et une unique matrice U triangulaire supérieure telles que $A = LU$. De plus $\det(A) = \prod_{i=1}^n u_{i,i}$.

Rappelons que le mineur principal d'ordre i de A est le déterminant des premières lignes et premières colonnes

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

La démonstration se fait par récurrence sur n

Proposition 1.3.23

La factorisation LU de la matrice A de taille n par la méthode de Gauss existe si l'une des conditions est vérifiée :

- La matrice A est définie positive.
- La matrice A est à diagonale strictement dominante par lignes ou par colonnes.

Pour mieux comprendre la méthode, on prend l'exemple suivant

$$A = A^{(1)} = \begin{pmatrix} 4 & -9 & 2 \\ 2 & -4 & 4 \\ -1 & 2 & 2 \end{pmatrix}$$

En effectuant les opérations suivantes :

$$\begin{aligned} L_2 &= L_2 - \frac{1}{2} L_1 \hookrightarrow l_{2,1} = \frac{1}{2} \\ L_3 &= L_3 + \frac{1}{4} L_1 \hookrightarrow l_{3,1} = -\frac{1}{4} \end{aligned}$$

On obtient

$$A^{(2)} = \begin{pmatrix} 4 & -9 & 2 \\ 0 & \frac{1}{2} & 3 \\ 0 & -\frac{1}{4} & \frac{9}{2} \end{pmatrix}$$

En refaisant la même procédure

$$L_3 = L_3 + \frac{1}{2} L_2 \hookrightarrow l_{3,2} = -\frac{1}{2}$$

On obtient finalement

$$U = A^{(3)} = \begin{pmatrix} 4 & -9 & 2 \\ 0 & \frac{1}{2} & 3 \\ 0 & 0 & 4 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{1}{4} & -\frac{1}{2} & 1 \end{pmatrix}$$

De l'exemple précédent on peut s'inspirer pour donner l'algorithme suivant

```

 $u_{1,1} = a_{1,1};$ 
Pour j de 2 à n faire
   $u_{1,j} = a_{1,j};$ 
   $l_{j,1} = a_{j,1}/u_{1,1};$ 
Fin Pour
Pour i de 2 à n-1 faire
   $u_{i,i} = a_{i,i} - \sum_{k=1}^{i-1} l_{i,k} u_{k,i}$ 
  Pour j de i+1 à n faire
     $u_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} l_{j,k} u_{k,j}$ 
     $l_{j,i} = (a_{j,i} - \sum_{k=1}^{i-1} l_{j,k} u_{k,j})/u_{i,i}$ 
  Fin Pour
Fin Pour
 $u_{n,n} = a_{n,n} - \sum_{k=1}^{n-1} l_{n,k} u_{k,n}$ 

```

Algorithme 4 – Algorithme de factorisation LU

Exemple 1.3.24

Déterminer, par cette méthode, la décomposition LU de la matrice suivante :

$$A = \begin{pmatrix} 1 & 4 & -3 \\ -2 & 8 & 5 \\ 3 & 4 & 7 \end{pmatrix}$$

En effectuant les opérations suivantes :

$$L_2 = L_2 + 2 L_1 \rightarrow l_{2,1} = -2$$

$$L_3 = L_3 - 3 L_1 \rightarrow l_{3,1} = -3$$

On obtient

$$A^{(2)} = \begin{pmatrix} 1 & 4 & -3 \\ 0 & 16 & -1 \\ 0 & -8 & 16 \end{pmatrix}$$

En refaisant la même procédure

$$L_3 = L_3 + \frac{1}{2} L_2 \rightarrow l_{3,2} = -\frac{1}{2}$$

On obtient finalement

$$U = A^{(3)} = \begin{pmatrix} 1 & 4 & -3 \\ 0 & 16 & -1 \\ 0 & 0 & 15.5 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & -\frac{1}{2} & 1 \end{pmatrix}$$

1.3.25 Méthode de Cholesky

La méthode de Cholesky est une alternative à la factorisation LU qui s'applique aux matrices symétriques et définie positives.

Rappelons qu'une matrice carrée symétrique $A \in \mathcal{M}_n(\mathbb{R})$ est dite symétrique définie positive si on a

$${}^t x A x > 0 \text{ pour tout } x \in \mathbb{R}^n \text{ et } \neq 0$$

et que pour une matrice symétrique définie positive les sous-matrices principales ont toutes des déterminants positifs.

Nous donnons maintenant une nouvelle caractérisation de ces matrices.

Théorème 1.3.26

Si $A \in \mathcal{M}_n(\mathbb{R})$ une matrice symétrique définie positive alors il existe au moins une matrice triangulaire inférieure L telle que $A = L {}^t L$. Si les éléments diagonaux $l_{i,i}$ de L sont strictement positifs alors la décomposition est unique.

La méthode de Cholesky ne s'applique qu'aux matrices réelles symétriques définies positives. Elle consiste en une factorisation $A = L {}^t L$, où L est une matrice triangulaire inférieure, dans le but de ramener la résolution de l'équation linéaire $Ax = b$ à la résolution de deux équations $Ly = b$ et ${}^t Lx = y$. Ceci est intéressant lorsqu'on a à résoudre plusieurs systèmes avec la même matrice et différents second membres car la décomposition est effectuée une fois pour toutes.

1.3.26.1 Algorithme de la factorisation de Cholesky

Soit A une matrice symétrique définie positive à coefficients réels, mettons A sous la forme $A = L {}^t L$. Les coefficients $L_{i,i}$ de L peuvent être calculés comme suit :

- $l_{1,1} = \sqrt{a_{1,1}}$
- pour tout $i = 2, \dots, n$

$$\begin{aligned} l_{i,j} &= (a_{i,j} - \sum_{k=1}^{j-1} l_{i,k} l_{k,j}) / l_{j,j}, \quad j = 1, \dots, i-1 \\ l_{i,i} &= (a_{i,i} - \sum_{k=1}^{i-1} l_{i,k}^2) / l_{j,j}. \end{aligned}$$

On résout ensuite le système $Ly = b$ puis ${}^t Lx = y$ par les algorithmes de descente et de remontée, respectivement.

```


$$l_{1,1} = \sqrt{a_{1,1}};$$

Pour i de 2 à n faire
  
$$l_{i,1} = a_{i,1}/l_{1,1};$$

Fin Pour
Pour j de 2 à n-1 faire
  
$$l_{j,j} = \sqrt{a_{j,j} - \sum_{k=1}^{j-1} l_{j,k}^2}$$

  Pour i de j+1 à n faire
    
$$l_{i,j} = (a_{i,j} - \sum_{k=1}^{j-1} l_{i,k} l_{j,k})/l_{j,j}$$

  Fin Pour
Fin Pour

$$l_{n,n} = \sqrt{a_{n,n} - \sum_{k=1}^{n-1} l_{n,k}^2}$$


```

Algorithme 5 – Algorithme de la factorisation de Cholesky

Exemple 1.3.27

Soit A une matrice définie positive tel que

$$A = \begin{pmatrix} 4 & 2 & -2 \\ 2 & 10 & 2 \\ -2 & 2 & 3 \end{pmatrix}$$

On a

$$l_{1,1} = \sqrt{a_{1,1}} = \sqrt{4} = 2$$

$$l_{2,1} = a_{2,1}/l_{1,1} = 1$$

$$l_{3,1} = a_{3,1}/l_{1,1} = -1$$

$$l_{2,2} = \sqrt{a_{2,2} - l_{2,1}^2} = 3$$

$$l_{3,2} = (a_{3,2} - l_{3,1}l_{2,1})/l_{2,2} = 1$$

$$l_{3,3} = \sqrt{a_{3,3} - (l_{3,2}^2 + l_{2,1}^2)} = 1$$

Donc, finalement matrice L obtenu est

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

Résolvant maintenant le système $Ax = b$ avec $b = {}^t(1, 1, 1)$.

Le système $Ax = b$ est équivalent à résoudre les deux systèmes suivants :

$$Ax = b \iff L^t L x = b \iff \begin{cases} Ly = b & (1) \\ {}^t L x = y & (2) \end{cases}$$

La solution du système (1) est

$$\begin{pmatrix} 1/2 \\ 1/6 \\ 4/3 \end{pmatrix}$$

$${}^t L x = y \iff \begin{pmatrix} 2 & 1 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/6 \\ 4/3 \end{pmatrix}$$

La solution du système (2) est

$$x = \begin{pmatrix} 10/9 \\ -7/18 \\ 4/3 \end{pmatrix}$$

Remarque 1.3.28

Comme la matrice A s'écrit sous la forme de multiplication d'une matrice L et sa transposée, donc $\det(A) = \prod_{i=1}^n l_{i,i}^2$.

Rappels : Matrice définie positive

Soit A une matrice symétrique réelle d'ordre n . Elle est dite définie positive si elle vérifie l'une des trois propriétés équivalentes suivantes :

1. Pour toute matrice colonne non nulle x à n éléments réels, on a

$${}^t x A x > 0.$$

2. Les valeurs propres de M sont strictement positives.
3. Les mineurs principaux dominants de A sont tous > 0 .

Exemple 1.3.29

On considère la matrice suivante :

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

Montrer que cette matrice est symétrique définie positive. Déterminer sa factorisation de Cholesky.

La matrice A est symétrique car $A = {}^t A$, montrons que A est définie positive

- Soit $x \in \mathbb{R}^3 \setminus \{0\}$, avec

$$x = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \quad 0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Montrons que ${}^t x A x > 0$

$${}^t x A x = (a, b, c) A \begin{pmatrix} a \\ b \\ c \end{pmatrix} = f(a, b, c)$$

$$\begin{aligned} f(a, b, c) &= 2a^2 - 2ab + 2b^2 - 2bc + c^2 \\ &= a^2 - 2ab + b^2 + a^2 + b^2 - 2bc + c^2 \\ &= (a - b)^2 + a^2 + (b - c)^2 > 0 \end{aligned}$$

- Cherchons les valeurs propres de A .

Les valeurs propres de A sont les racines du polynôme caractéristique

$$P(\lambda) = \det(A - \lambda I) = 0$$

Ceci conduit à calculer

$$\begin{vmatrix} 2 - \lambda & -1 & 0 \\ -1 & 2 - \lambda & -1 \\ 0 & -1 & 1 - \lambda \end{vmatrix} = 0$$

On trouve ainsi

$$P(\lambda) = 1 - 6\lambda + 5\lambda^2 - \lambda^3$$

Les valeurs propres sont donc

$$\lambda_1 = 0.198062, \quad \lambda_2 = 1.55496, \quad \lambda_3 = 3.24698$$

- Les mineurs principaux dominants de A sont

$$\Delta_1 = \det(2) = 2 > 0, \quad \Delta_2 = \det \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} = 3 > 0, \quad \Delta_3 = \det \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} = 1 > 0.$$

La matrice A est symétrique définie positive donc elle admet une factorisation de Cholesky. On a

$$l_{1,1} = \sqrt{a_{1,1}} = \sqrt{2} = 1.4142$$

$$l_{2,1} = a_{2,1}/l_{1,1} = -0.7071$$

$$l_{3,1} = a_{3,1}/l_{1,1} = 0$$

$$l_{2,2} = \sqrt{a_{2,2} - l_{2,1}^2} = 1.2247$$

$$l_{3,2} = (a_{3,2} - l_{3,1}l_{2,1})/l_{2,2} = -0.8165$$

$$l_{3,3} = \sqrt{a_{3,3} - (l_{3,2}^2 + l_{2,1}^2)} = 0.5774$$

Donc, finalement matrice L obtenu est

$$L = \begin{pmatrix} 1.4142 & 0 & 0 \\ -0.7071 & 1.2247 & 0 \\ 0 & -0.8165 & 0.5774 \end{pmatrix}$$

1.4 Méthode itératives

Les méthodes directes sont très coûteuses en termes de nombre d'opérations, donc de temps de calcul lorsque la taille du système linéaire est assez élevée. On peut alors avoir recours à des méthodes où la solution est obtenue par itérations successives et où chaque itération consiste à résoudre un système linéaire moins coûteux.

On considère une matrice $A \in \mathcal{M}_n(\mathbb{R})$ inversible, un vecteur $b \in \mathbb{R}^n$ et un système linéaire

$$(S) : Ax = b$$

Résoudre le système linéaire (S) par une méthode itérative consiste à construire une suite de

vecteurs $(x^{(k)})_k$ de \mathbb{R}^n qui converge vers la solution exacte $x = A^{-1}b$, c'est-à-dire

$$\lim_{k \rightarrow +\infty} x^{(k)} = x \quad (1.3)$$

pour n'importe quelle donnée initiale $x^{(0)} \in \mathbb{R}^n$.

La suite $(x^{(k)})_k$ est obtenu, à partir d'un vecteur initial arbitraire $x^{(0)}$, par une relation de récurrence de la forme

$$x^{(k+1)} = Bx^{(k)} + c, \quad \forall k \in \mathbb{N} \quad (k \geq 0) \quad (1.4)$$

où B est une matrice bien choisi (dépendant de A) appelée matrice d'itération de la méthode et c est un vecteur (dépendant de A et b), qui vérifient la relation de consistance

$$x = Bx + c, \quad \forall k \in \mathbb{N} \quad (k \geq 0) \quad (1.5)$$

Comme $x = A^{-1}b$, ceci implique $c = (I_n - B)A^{-1}b$

1.4.1 Convergence des méthodes itératives

Définition 1.4.2

On appelle erreur (resp résidu) d'ordre k , $k \in \mathbb{N}$, de la méthode d'itération le vecteur $e^{(k)} = x^{(k)} - x$ (resp $r^{(k)} = b - Ax^{(k)}$), où $x = A^{-1}b$ est la solution de $Ax = b$.

La mise en oeuvre pratique d'une méthode itérative de la forme (1.4) nécessite la donnée d'un point de départ $x^{(0)}$ (en général, sauf si l'on possède des informations a priori sur la solution, on choisit le vecteur nul) et d'une tolérance sur la solution que l'on cherche à calculer. On calcule ensuite les itérés $x^{(k)}$, $k = 1, 2, \dots$ en utilisant la formule (1.4) jusqu'à ce que le résidu $r^{(k)} = b - Ax^{(k)}$ soit plus petit que la tolérance.

On déduit de cette définition qu'une méthode itérative consistante de la forme $Ax = b$ converge si et seulement si $\lim_{k \rightarrow +\infty} e^k = 0$ (soit encore si $\lim_{k \rightarrow +\infty} r^k = \lim_{k \rightarrow +\infty} Ae^k = 0$).

Lemme 1.4.3

Si la méthode itérative (1.4) est convergente et si on note $x = A^{-1}b$ la solution exacte du système linéaire (S), alors

$$x^{(k)} - x = B^k(x^{(0)} - x)$$

Preuve

On a $c = (I_n - B)A^{-1}b = (I_n - B)x$ d'où $x^{(k+1)} = Bx^{(k)} + (I_n - B)x$ ou encore $x^{(k+1)} - x = B(x^{(k)} - x)$ d'où le résultat.

Définition 1.4.4 (Norme matricielle)

On dit que l'application $\|\cdot\| : \mathcal{M}_n(\mathbb{R}) \rightarrow \mathbb{R}$ est une norme matricielle si :

- $\forall A \in \mathcal{M}_n(\mathbb{R}), \quad \|A\| = 0 \iff A = 0.$
- $\forall \alpha \in \mathbb{R}, \forall A \in \mathcal{M}_n(\mathbb{R}), \quad \|\alpha A\| = |\alpha| \|A\|.$
- $\forall A, B \in \mathcal{M}_n(\mathbb{R}), \quad \|A + B\| \leq \|A\| + \|B\|.$
- $\forall A, B \in \mathcal{M}_n(\mathbb{R}), \quad \|AB\| \leq \|A\| \|B\|.$

Définition 1.4.5 (Norme matricielle subordonnée)

Soit $\|\cdot\|$ une norme vectorielle sur \mathbb{R}^n . On définit la norme matricielle subordonnée à la norme vectorielle $\|\cdot\|$ (que l'on notera aussi $\|\cdot\|$) par

$$\forall A \in \mathcal{M}_n(\mathbb{R}), \quad \|A\| = \sup_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ax\|}{\|x\|}$$

On dit aussi que c'est une norme matricielle subordonnée à une norme vectorielle ou associée à la norme vectorielle.

Le résultat suivant nous donne des critères pour tester la convergence de la méthode itérative (1.4).

Théorème 1.4.6

Les assertions suivantes sont équivalentes :

- i) La méthode itérative (1.4) est convergente ;
- ii) Pour tout $y \in \mathbb{R}^n$, $\lim_{k \rightarrow +\infty} B^k y = 0$;
- iii) Pour toute norme matricielle $\|\cdot\|$ sur $\mathcal{M}_n(\mathbb{R})$, on a $\lim_{k \rightarrow +\infty} \|B^k\| = 0$.

Preuve

Admis pour ce cours.

En pratique, les caractérisations précédentes de la convergence d'une méthode itérative ne sont pas faciles à vérifier. On utilise plutôt le résultat suivant :

Théorème 1.4.7

Les assertions suivantes sont équivalentes :

- i) La méthode itérative (1.4) est convergente ;
- ii) $\rho(B) < 1$, où $\rho(B)$ désigne le rayon spectral de la matrice B , i.e., le maximum des modules des valeurs propres de B ;
- iii) Il existe une norme matricielle $\|\cdot\|$ sur $\mathcal{M}_n(\mathbb{R})$ subordonnée à une norme vectorielle sur \mathbb{R}^n telle que $\|B\| < 1$.

Preuve

Admis pour ce cours.

1.4.8 Quelques méthodes itératives

Comment construire une méthode itérative ?

Une technique générale pour construire des méthodes itératives est basée sur une décomposition (splitting) de la matrice A sous la forme $A = M - N$, où M et N sont des matrices à déterminer avec M non singulière. La matrice M est appelée matrice de préconditionnement. Plus précisément, $x^{(0)}$ étant donné, on peut calculer $x^{(k)}$, pour $k \geq 1$, en résolvant le système

$$Mx^{(k+1)} = Nx^{(k)} + b, \quad k \geq 0 \quad (1.6)$$

Clairement, la solution exacte x satisfait $Mx = Nx + b$ et donc $Ax = b$. Le système (1.6) peut être écrit également sous la forme (1.4), avec $B = M^{-1}N$, et $c = M^{-1}b$. Une relation de récurrence équivalente à (1.6) est la suivante :

$$M(x^{(k+1)} - x^{(k)}) = r^{(k)}, \quad k \geq 0 \quad (1.7)$$

où $r^{(k)} = b - Ax^{(k)}$ est le résidu à l'itération k . On peut généraliser la relation précédente par

$$M(x^{(k+1)} - x^{(k)}) = \alpha_k r^{(k)}, \quad k \geq 0 \quad (1.8)$$

où on a introduit un paramètre α_k (qui peut être différent à chaque itération k) afin d'accélérer la convergence. Cette méthode est dite de Richardson. Les relations (1.6), (1.7) et (1.8) montrent qu'on doit résoudre un système linéaire de matrice M à chaque itération ; donc M doit être telle que la résolution du système ait un coût raisonnable. Par exemple, on pourra choisir M diagonale ou triangulaire.

Nous allons maintenant considérer trois exemples classiques : les méthodes de Jacobi, Gauss-Seidel et de relaxation. Le point de départ de chacune de ces méthodes est l'unique décomposition de la matrice $A = (a_{i,j})_{1 \leq i,j \leq n}$ sous la forme $A = D - E - F$ avec :

- $D = (d_{i,j})_{1 \leq i,j \leq n}$ diagonale, telle que $d_{i,i} = a_{i,i}$ et $d_{i,j} = 0$ pour $i \neq j$;
- $E = (e_{i,j})_{1 \leq i,j \leq n}$ triangulaire inférieure **stricte** telle que $e_{i,j} = -a_{i,j}$ si $i > j$ et $e_{i,j} = 0$ si $i \leq j$;
- $F = (f_{i,j})_{1 \leq i,j \leq n}$ triangulaire supérieure **stricte** telle que $f_{i,j} = -a_{i,j}$ si $i < j$ et $f_{i,j} = 0$ si $i \geq j$.

Exemple 1.4.9

Considérons le système

$$A = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}$$

La décomposition de A sous la forme $A = D - E - F$ décrite ci-dessus s'écrit alors

$$\underbrace{\begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}}_A = \underbrace{\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}}_D - \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ -1 & -1 & 0 \end{pmatrix}}_E - \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ -1 & -1 & 0 \end{pmatrix}}_F$$

On supposera de plus que D est inversible et on distingue les trois méthodes suivantes :

- Méthode de Jacobi : $M = D$, $N = E + F$;
- Méthode de Gauss-Seidel : $M = D - E$, $N = F$;
- Méthode de relaxation : $M = \frac{1}{\omega}(D - \omega E)$, $N = \frac{1-\omega}{\omega}D + F$ avec ω paramètre réel non nul.

On remarque que la méthode de Gauss-Seidel est un cas particulier de la méthode relaxation pour $\omega = 1$.

1.4.9.1 Méthode de Jacobi

On considère un système linéaire (S) : $Ax = b$ avec A inversible. On pose $A = M - N$ avec $M = D$ inversible et $N = E + F$. Le schéma itératif s'écrit alors

$$\begin{cases} x^{(0)} \in \mathbb{R}^n \\ Dx^{(k+1)} = (E + F)x^{(k)} + b, \quad \forall k \geq 0. \end{cases} \iff \begin{cases} x^{(0)} \in \mathbb{R}^n \\ x^{(k+1)} = D^{-1}(E + F)x^{(k)} + D^{-1}b, \quad \forall k \geq 0. \end{cases}$$

Chaque itération consiste donc à résoudre un système linéaire diagonal.

Définition 1.4.10

La matrice $B_J = D^{-1}(E + F)$ s'appelle la matrice de Jacobi associée à A .

On calcul $x^{(k+1)}$ par l'algorithme suivant

$$x_i^{(k+1)} = (b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{i,j}x_j^{(k)}) / a_{i,i}, \quad i = 1, \dots, n.$$

```

Données :  $A$ ,  $b$ ,  $x^0$ ,  $n$ ,  $\varepsilon$  et  $N_{max}$ 
Pour i de 1 à n faire
     $x_i^{new} \leftarrow x_i^0$ 
Fin Pour
     $nb \leftarrow 0$ 
Tant que ( $\|Ax^{new} - b\| > \varepsilon$  et  $nb < N_{max}$ ) faire
         $nb \leftarrow nb + 1$ 
        Pour i de 1 à n faire
             $x_i^{old} \leftarrow x_i^{new}$ 
        Fin Pour
        Pour i de 1 à n faire
             $x_i^{new} \leftarrow (b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{i,j}x_j^{old}) / a_{i,i}$ 
        Fin Pour
    Fait
```

Algorithme 6 – Algorithme de la méthode de Jacobi

Exemple 1.4.11

Considérons le système

$$\begin{pmatrix} 4 & 2 & 1 \\ -1 & 2 & 0 \\ 2 & 1 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ 9 \end{pmatrix}$$

mis sous la forme

$$\begin{cases} x = 1 - y/2 - z/4 \\ y = 1 + x/2 \\ z = 9/4 - x/2 - y/4 \end{cases}$$

Soit $x^{(0)} = (0, 0, 0)$ le vecteur initial, en calculant les itérées on trouve

$$\begin{aligned} x^{(1)} &= (1, 1, 9/4) & x^{(2)} &= (-1/16, 3/2, 3/2) \\ x^{(3)} &= (-1/8, -1/32, 3/2) & x^{(4)} &= (5/128, 15/16, 265/128) \\ x^{(5)} &= (7/512, 261/256, 511/256) \end{aligned}$$

La suite $x^{(k)}$ converge vers la solution du système $(0, 1, 2)$.

D'après le théorème 1.4.7, on a le résultat suivant :

Théorème 1.4.12

La méthode de Jacobi converge si et seulement si $\rho(B_J) < 1$.

Exemple 1.4.13

Pour la matrice A donnée par 1.4.9, on obtient :

$$B_J = D^{-1}(E + F) = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ -2 & 0 & 2 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

Les valeurs propres de la matrices B_J sont 0 et $\pm i\frac{\sqrt{5}}{2}$. On a donc $\rho(B_J) = \frac{\sqrt{5}}{2} > 1$ et la méthode de Jacobi diverge.

1.4.13.1 Méthode de Gauss-Seidel

Dans la méthode de Gauss-Seidel, on choisi $M = D - E$ et $N = F$ ce qui conduit à considérer la relation de récurrence

$$\left\{ \begin{array}{l} x^{(0)} \in \mathbb{R}^n \\ (D - E)x^{(k+1)} = Fx^{(k)} + b, \quad \forall k \geq 0. \end{array} \right. \iff \left\{ \begin{array}{l} x^{(0)} \in \mathbb{R}^n \\ x^{(k+1)} = (D - E)^{-1}Fx^{(k)} + (D - E)^{-1}b, \quad \forall k \geq 0. \end{array} \right.$$

Définition 1.4.14

La matrice $B_{GS} = (D - E)^{-1}F$ s'appelle la matrice de Gauss-Seidel associée à A .

On calcul $x^{(k+1)}$ par l'algorithme suivant

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^n a_{i,j} x_j^{(k)} \right) / a_{i,i}, \quad i = 1, \dots, n.$$

Remarque 1.4.15

La méthode de Gauss-Seidel est une amélioration de la méthode de jacobi dans laquelle les vecteurs calculées sont utilisées au fur et à mesure du calcul et non à l'issue d'une itération comme dans la méthode de Jacobi. On améliore ainsi la vitesse de convergence.

```

Données :  $A, b, x^0, n, \varepsilon$  et  $N_{max}$ 
Pour i de 1 à n faire
   $x_i^{new} \leftarrow x_i^0$ 
Fin Pour
   $nb \leftarrow 0$ 
Tant que ( $\|Ax^{new} - b\| > \varepsilon$  et  $nb < N_{max}$ ) faire
   $nb \leftarrow nb + 1$ 
  Pour i de 1 à n faire
     $x_i^{old} \leftarrow x_i^{new}$ 
  Fin Pour
  Pour i de 1 à n faire
     $x_i^{new} \leftarrow \left( b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{new} - \sum_{j=i+1}^n a_{i,j} x_j^{old} \right) / a_{i,i}$ 
  Fin Pour
Fait

```

Algorithme 7 – Algorithme de la méthode de Gauss-Seidel

Exemple 1.4.16

Considérons le système

$$\begin{cases} x = 1 - y/2 - z/4 \\ y = 1 + x/2 \\ z = 9/4 - x/2 - y/4 \end{cases}$$

Partant du point $x^{(0)} = (0, 0, 0)$, on calcule successivement

$$\begin{aligned} x^{(1)} &= (1, 3/2, 11/8) & x^{(2)} &= (-3/32, 61/64, 527/256) \\ x^{(3)} &= (9/1024, 2047/2048, 16349/8192) \end{aligned}$$

Cet ensemble de points converge vers la solution exacte $(0, 1, 2)$.

D'après le théorème 1.4.7, on a le résultat suivant :

Théorème 1.4.17

La méthode de Gauss-Seidel converge si et seulement si $\rho(B_{GS}) < 1$.

Exemple 1.4.18

Pour la matrice A donnée par 1.4.9, on obtient :

$$B_{GS} = (D - E)^{-1}F = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & -1 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

Les valeurs propres de la matrices B_{GS} sont 0 et $-\frac{1}{2}$ (de multiplicité 2). On a donc $\rho(B_{GS}) = \frac{1}{2} < 1$ donc la méthode de Gauss-Seidel converge.

La méthode de Gauss-Seidel est aussi utilisé pour résoudre des système non-linéaires.

Exemple 1.4.19

Considérons le système

$$\begin{cases} x = \sin(xy) - y/2\pi \\ y = 2\pi x - (\pi - 1/4)(e^{2x-1} - 1) \end{cases}$$

Partant du point $x^{(0)} = (0, 0, 0)$, on calcule successivement

$$x^{(1)} = (0.455, 3.03) \quad x^{(2)} = (0.499, 3.11) \quad x^{(3)} = (0.505, 3.14)$$

qui converge vers la solution exacte $(1/2, \pi)$.

Remarque 1.4.20

Pour que la méthode de Gauss-Seiled soit bien définie, il faut que la matrice D soit inversible, mais, là encore, cette condition n'est pas très restrictive en pratique. On peut également introduire dans cette méthode un paramètre de relaxation ω . On parle alors de la méthode de Relaxation.

1.4.20.1 Méthode de Relaxation

Dans la méthode de Relaxation, on pose $A = M - N$ avec $M = \frac{1}{\omega}(D - \omega E)$ inversible et $N = \frac{1-\omega}{\omega}D + F$. Le schéma itératif s'écrit alors

$$\begin{cases} x^{(0)} \in \mathbb{R}^n \\ \frac{1}{\omega}(D - \omega E)x^{(k+1)} = (\frac{1-\omega}{\omega}D + F)x^{(k)} + b, \quad \forall k \geq 0. \end{cases}$$

qui est équivalent à :

$$\begin{cases} x^{(0)} \in \mathbb{R}^n \\ x^{(k+1)} = (D - \omega E)^{-1}((1 - \omega)D + F)x^{(k)} + \omega(D - \omega E)^{-1}b, \quad \forall k \geq 0. \end{cases}$$

Définition 1.4.21

La matrice $B_R(\omega) = (D - \omega E)^{-1}((1 - \omega)D + F)$ s'appelle la matrice de relaxation associée à A et ω est le facteur de relaxation. Si $\omega < 1$, on parle de sous-relaxation, si $\omega = 1$, on retrouve la méthode de Gauss-Seidel et si $\omega > 1$, on parle de sur-relaxation.

On calcul $x^{(k+1)}$ par l'algorithme suivant

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{i,i}}(b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{(k+1)} - \sum_{j=i+1}^n a_{i,j}x_j^{(k)}), \quad i = 1, \dots, n.$$

```

Données :  $A, b, x^0, n, \varepsilon, \omega$  et  $N_{max}$ 
Pour i de 1 à n faire
   $x_i^{new} \leftarrow x_i^0$ 
Fin Pour
 $nb \leftarrow 0$ 
Tant que ( $\|Ax^{new} - b\| > \varepsilon$  et  $nb < N_{max}$ ) faire
   $nb \leftarrow nb + 1$ 
  Pour i de 1 à n faire
     $x_i^{old} \leftarrow x_i^{new}$ 
  Fin Pour
  Pour i de 1 à n faire
     $x_i^{new} \leftarrow \omega x_{GS,i}^{new} + (1 - \omega)x_i^{old}$ 
  Fin Pour
Fait

```

Algorithme 8 – Algorithme de la méthode de Relaxation

Avec

$$x_{GS,i}^{new} = (b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{new} - \sum_{j=i+1}^n a_{i,j}x_j^{old}) / a_{i,i}$$

Remarque 1.4.22

Pour $\omega = 1$, nous retrouvons la méthode de Gauss Seidel. Pour $\omega < 1$ on appelle cette méthode méthode de sous-relaxation et pour $\omega > 1$ méthode de sur-relaxation.

D'après le théorème 1.4.7, on a le résultat suivant :

Théorème 1.4.23

La méthode de Relaxation converge si et seulement si $\rho(B_R(\omega)) < 1$.

Exemple 1.4.24

Pour la matrice A donnée par 1.4.9, on obtient :

$$B_R(\omega) = \begin{pmatrix} 1 - \omega & \frac{1}{2}\omega & -\frac{1}{2}\omega \\ \omega(1 - \omega) & -\frac{1}{2}\omega^2 + 1 - \omega & -\frac{1}{2}\omega^2 - \omega \\ \frac{1}{2}\omega(1 - \omega)^2 & -\frac{1}{4}\omega^3 - \frac{1}{4}\omega^2 + \frac{1}{2}\omega & \frac{1}{4}\omega^3 - \frac{3}{4}\omega^2 + 1 - \omega \end{pmatrix}$$

Les valeurs propres de la matrice $B_R(\omega)$ dépendent en général de ω donc la convergence de la méthode de relaxation dépendra aussi de la valeur de ω .

1.4.25 Résultats de convergence dans des cas particuliers

Nous allons maintenant examiner la convergence de ces méthodes itératives dans le cas d'une matrice symétrique définie positive.

Théorème 1.4.26

Soit A une matrice symétrique définie positive et soit $A = M - N$ une décomposition de A où M est inversible. On suppose que la matrice ${}^tM + N$ est symétrique définie positive. Alors, la méthode itérative $Mx^{(k+1)} = Nx^{(k)} + b$ converge.

On peut déduire de ce résultat le corollaire suivant :

Corollaire 1.4.27

Si A est symétrique définie positive, la méthode de relaxation avec $0 < \omega < 2$ converge.

Remarque 1.4.28

Soit A une matrice symétrique définie positive. Alors la méthode de Gauss-Seidel converge.

Corollaire 1.4.29

Soit A une matrice symétrique définie positive telle que $2D - A$ soit définie positive. Alors, la méthode de Jacobi converge.

Une autre catégorie de matrices pour lesquelles l'utilisation des méthodes de décomposition est intéressante est celle des matrices à diagonale dominante.

Définition 1.4.30

On dit qu'une matrice $A = (a_{i,i})_{1 \leq i,j \leq n}$ est à diagonale dominante si on a

$$\forall i = 1, \dots, n \quad |a_{i,i}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{i,j}|$$

On dit qu'elle est à diagonale strictement dominante si l'inégalité ci-dessus est stricte.

Théorème 1.4.31

Soit A une matrice à diagonale strictement dominante. Alors A est inversible. De plus, les méthodes de Jacobi et de Gauss-Seidel convergent.

La méthode de Gauss-Seidel converge généralement plus vite et plus souvent que la méthode de Jacobi. La méthode de relaxation nécessite la connaissance d'une valeur optimale du paramètre de relaxation ω .

La convergence des méthodes itératives se teste généralement en utilisant un paramètre de tolérance ε . On arrête ainsi les calculs dès que l'erreur relative est assez petite :

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} < \varepsilon$$

Une autre alternative est de tester le résidu :

$$\|Ax^{(k+1)} - b\| < \varepsilon$$