Analyse Numérique

Antoine Godichon-Baggioni

Ce polycopié reprend les notes de cours de P. Tarrago

Table des matières

1 Système linéaire et décompositions LU **5**

1.1 Rappels sur les matrices . . . 6
1.2 Conditionnement de matrices . . . 7
1.2.1 Normes matricielles . . . 7
1.2.2 Conditionnement d'une matrice . . . 10
1.3 Méthode d'élimination de Gauss . . . 11
1.3.1 Méthode sans pivot . . . 11
1.3.2 Méthode avec pivot . . . 13
1.3.3 Coût de la méthode de Gauss . . . 14
1.4 Décomposition LU . . . 15
1.4.1 Théorème de décomposition et conséquences . . . 15
1.4.2 Preuve de la décomposition $A = LU$ . . . 17
1.4.3 Preuve de la décomposition $PA = LU$ . . . 20

2 Méthodes itératives **27**

2.1 Types de méthodes itératives et résultats généraux . . . 28
2.2 Convergence des méthodes itératives . . . 31
2.2.1 Convergence théorique d'une méthode de type I . . . 31
2.2.2 Critères d'arrêt . . . 35
2.3 Diagonalisation de matrice . . . 36

Chapitre 1

Système linéaire et décompositions LU

L'objectif de ce cours est de proposer des méthodes efficaces pour résoudre des systèmes d'équations linéaires. Le problème revient à résoudre l'équation

$$Ax = b, \tag{1.1}$$

où $A = (a_{ij})_{\substack{1 \le i \le n \\ 1 \le j \le m}}$ est une matrice de dimensions $n \times m$, b est un vecteur de dimension n à coefficients réels ou complexes, et x est un vecteur de dimension m. Pour analyser ce type de problème, nous distinguons deux aspects fondamentaux :

1. **Aspect théorique de (1.1) :**
   (a) Le problème admet-il une ou plusieurs solutions ?
   (b) Existe-t-il une solution optimale ou préférable ?
   (c) Comment les solutions évoluent-elles qualitativement lorsque b varie ?
2. **Aspect pratique du problème :**
   (a) Comment obtenir concrètement cette solution ?
   (b) Quel est le coût algorithmique de la recherche de solution ?
   (c) Le problème est-il numériquement stable ?

Quand nous sommes confrontés au problème (1.1) sur un problème concret, l'ensemble des réponses à ces six questions est importante : la connaissance de l'existence d'une solution unique mais inaccessible numériquement (ou en un temps déraisonnable) est inutile.

Dans ce cours, nous répondrons à ces six questions pour le cas particulier d'un système carré inversible (cf. Section 1.1 pour un rappel de cette notion). Dans ce cas, il est bien connu que l'équation (1.1) admet une solution unique, donnée par

$$x = A^{-1}b. \tag{1.2}$$

Ceci répond directement aux points $1.(a)$ et $1.(b)$.

1.1 Rappels sur les matrices

Dans ce cours, nous travaillerons avec des matrices à coefficients réels ou complexes, bien que la plupart des définitions et propriétés restent valides pour des matrices dont les coefficients appartiennent à un corps arbitraire $\mathbb{K}$. Dans ce qui suit, nous considérons que $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$.

Matrice complexe : Une matrice A de dimensions $n \times m$ est déterminée par $n \times m$ éléments $a_{ij} \in \mathbb{K}$ pour $1 \leq i \leq n$ et $1 \leq j \leq m$. L'espace vectoriel des matrices à coefficients dans $\mathbb{K}$ est noté $M_{n,m}(\mathbb{K})$. Si $n = m$, la matrice est dite carrée et on note simplement cet espace $M_n(\mathbb{K})$ (où n est appelé la taille de la matrice). La matrice nulle dans $M_{n,m}(\mathbb{K})$ est celle dont tous les coefficients sont nuls. Pour $A \in M_{n,m}(\mathbb{K})$, on note $A[i,]$ la i-ème ligne de A et $A[,j]$ la j-ème colonne. Formellement, $A[i,] \in M_{1,m}(\mathbb{K})$ et $A[,j] \in M_{n,1}(\mathbb{K})$.

Transposée d'une matrice : La transposée d'une matrice $A \in M_{n,m}(\mathbb{K})$ est la matrice $A^T \in M_{m,n}(\mathbb{K})$ définie par

$$[A^T]_{ij} = a_{ji}.$$

Produit de matrices : Si A est une matrice $n \times m$ et B une matrice $m \times \ell$, le produit $AB \in M_{n,\ell}(\mathbb{K})$ est la matrice (c_{ij}) définie par

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj},$$

où a_{ik} et b_{kj} représentent les coefficients de A et B.

Base de $M_n(\mathbb{K})$: Pour $1 \leq k, \ell \leq n$, notons $E_{k\ell} = (\delta_{ik}\delta_{j\ell})_{1\leq i,j\leq n}$ la matrice ayant des zéros partout sauf en position (k, ℓ). Les règles de multiplication des matrices impliquent que

$$E_{ij}E_{k\ell} = \delta_{jk}E_{i\ell}.$$

Matrice identité : La matrice identité $I_n = \sum_{i=1}^{n} E_{ii} = (\delta_{ij})_{1\leq i,j\leq n}$ est l'unique matrice dans $M_n(\mathbb{K})$ telle que

$$I_n A = A I_n = A$$

pour toute matrice $A \in M_n(\mathbb{K})$.

Matrice inversible : Une matrice carrée $A \in M_n(\mathbb{K})$ est dite inversible s'il existe une matrice $B \in M_n(\mathbb{K})$ telle que

$$AB = BA = I_n.$$

On note alors $B = A^{-1}$.

Déterminant d'une matrice : Soit $A \in M_n(\mathbb{K})$. Son déterminant est défini par

$$\det A := \sum_{\sigma \in S_n} \epsilon(\sigma) a_{1\sigma(1)} \cdots a_{n\sigma(n)}.$$

La matrice A est inversible si et seulement si son déterminant est non nul, et dans ce cas

$$A^{-1} = \frac{1}{\det A}(\mathrm{Com}\, A)^T, \tag{1.3}$$

où $(\mathrm{Com}\, A)_{ij} = (-1)^{i+j} \det A^{[ij]}$ et $A^{[ij]}$ est la matrice obtenue en supprimant la i-ème ligne et la j-ème colonne de A. L'ensemble des matrices inversibles à coefficients dans $\mathbb{K}$ est noté $GL_n(\mathbb{K})$.

Vecteurs : Un vecteur $x \in \mathbb{K}^n$ est identifié avec la matrice colonne $(x_i)_{1\leq i\leq n} \in M_{n,1}(\mathbb{K})$. Pour toute matrice $A \in M_{m,n}(\mathbb{K})$, la multiplication

$$x \mapsto Ax = \left(\sum_{k=1}^{n} a_{ik}x_k\right)_{1\leq i\leq m}$$

définit une application linéaire de $\mathbb{K}^n$ vers $\mathbb{K}^m$. Le noyau de A, noté $\ker A$, est l'ensemble $\{x \in \mathbb{K}^n \mid Ax = 0\} \subset \mathbb{K}^n$ et l'image de A, notée $\mathrm{Im}\, A$, est l'ensemble $\{Ax \mid x \in \mathbb{K}^n\} \subset \mathbb{K}^m$.

Dans ce cours, nous nous concentrerons principalement sur les matrices carrées, tout en gardant en tête que plusieurs résultats restent valides pour les matrices rectangulaires.

1.2 Conditionnement de matrices

1.2.1 Normes matricielles

Rappelons qu'une norme $\|\cdot\|$ sur un espace vectoriel E est une application

$$\|\cdot\| : E \to \mathbb{R}^+$$

vérifiant les axiomes suivants :

- **Séparation :** $\|x\| = 0$ si et seulement si $x = 0$,
- **Homogéinité :** $\|\lambda x\| = |\lambda|\|x\|$ pour tout $x \in E$, $\lambda \in \mathbb{K}$,
- **Inégalité triangulaire :** $\|x + y\| \leq \|x\| + \|y\|$ pour tout $x, y \in E$.

Suivant le problème que l'on considère, il est parfois utile de considérer des normes différentes. Par exemple sur $\mathbb{K}^n$, nous considérons trois normes particulières :

- **Norme euclidienne :** $\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$.
- **Norme** 1 **:** $\|x\|_1 = \sum_{i=1}^n |x_i|$.
- **Norme infinie :** $\|x\|_\infty = \max_{1\leq i\leq n} |x_i|$.

Nous laissons en exercice la preuve que ces trois applications sont bien des normes.

Definition 1.2.1. *Soit $\|\cdot\|$ une norme sur $\mathbb{K}^n$. La norme opérateur sur $M_n(\mathbb{K})$ associée à $\|\cdot\|$ est l'application*

$$|||.||| : A \mapsto |||A||| = \sup_{x\in\mathbb{K}^n\setminus\{0\}} \frac{\|Ax\|}{\|x\|} = \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|=1}} \|Ax\|.$$

Proposition 1.2.1. *Soit $\|\cdot\|$ une norme sur $\mathbb{K}^n$. Alors*

1. *L'application $|||\cdot|||$ est une norme.*
2. *Pour tout $x \in \mathbb{K}^n$,*
$$\|Ax\| \leq |||A|||\|x\|.$$
3. *Pour tout $A, B \in M_n(\mathbb{K})$,*
$$|||AB||| \leq |||A|||||B|||.$$

Démonstration. 1. Il faut prouver les trois axiomes d'une norme. Premièrement, $|||A||| = 0$ si et seulement si pour tout vecteur de norme 1, $Ax = 0$. Cela signifie donc que pour tout vecteur non nul, on a $Ax = \|x\|A\left(\frac{x}{\|x\|}\right) = 0$, et donc $A = 0$. D'autre part, pour $A, B \in M_n(\mathbb{K})$ et $\lambda \in \mathbb{K}$, on a par les propriétés de $\|\cdot\|$

$$|||\lambda A||| = \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|=1}} \|\lambda Ax\| = |\lambda| \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|=1}} \|Ax\| = |\lambda|||A|||$$

et

$$\begin{aligned} |||A+B||| &= \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|=1}} \|(A+B)x\| \\ &\leq \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|=1}} (\|Ax\| + \|Bx\|) \\ &\leq \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|=1}} \|Ax\| + \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|=1}} \|Bx\| \leq |||A||| + |||B|||. \end{aligned}$$

2. Pour $x = 0$ le résultat est clair. Par définition de $|||\cdot|||$, pour $x \in \mathbb{K}^n$ non nul, on a
$$\|Ax\| = \|x\|\frac{\|Ax\|}{\|x\|} \leq \|x\| \sup_{x\in\mathbb{K}^n\setminus\{0\}} \frac{\|Ax\|}{\|x\|} = |||A|||\|x\|.$$

3. Soit $A, B \in M_n(\mathbb{K})$. En utilisant le point précédent pour A puis pour B, pour tout $x \in \mathbb{K}^n$ de norme 1 on a
$$\|ABx\| \leq \|Bx\|||A||| \leq |||A|||||B|||.$$
Par passage au supremum on en déduit que
$$|||AB||| \leq |||A|||||B|||.$$

□

Exemple 1 : La norme induite par la norme $\|\cdot\|_2$ est également souvent appelée simplement norme

opérateur ou norme spectrale. Elle vaut donc

$$\|A\|_2 = \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|_2=1}} \|Ax\|_2.$$

Cette norme est importante car elle mesure la transformation de la norme euclidienne (la norme la plus naturelle sur $\mathbb{K}^n$) par A. Le problème est qu'elle n'a pas de forme explicite. On peut cependant montrer que

$$\|A\|_2^2 = \sup\{|\lambda|, \lambda \text{ valeur propre de } A^*A\}.$$

A noter que A^*A est au semi définie positive. En effet, on a

$$\langle x, A^*Ax\rangle = \langle Ax, Ax\rangle = \|Ax\|^2 \geq 0.$$

De plus, comme A^*A est symétrique, elle est diagonalisable et les vecteurs propres forment une base orthonormée de $\mathbb{K}^n$. On note D la matrice diagonale associée, et les valeurs propres de A^TA sont $\lambda_1, \ldots, \lambda_n$. Pour tout $\|x\|_2 = 1$, si on écrit x dans cette base orthonormée, comme $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n)$ avec $\|\tilde{x}\|_2 = 1$, alors

$$\|Ax\|_2^2 = \langle x, A^*Ax\rangle = \tilde{x}^T D\overline{\tilde{x}} = \sum_{i=1}^n \lambda_i |\tilde{x}_i|^2 \leq \max\{\lambda_i\} \underbrace{\sum_{i=1}^n \tilde{x}_i^2}_{=1}.$$

Ainsi,

$$|||A|||_2^2 = \left(\sup_{\substack{x\in\mathbb{K}^n \\ \|x\|_2=1}} \|Ax\|_2\right)^2 = \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|_2=1}} \|Ax\|_2^2 \leq \lambda_{\max},$$

où $\lambda_{\max}$ est la plus grande valeur propre de A^TA. De plus, en prenant un vecteur propre unitaire $v_{\max}$ associé à $\lambda_{\max}$, on a

$$\|Av_{\max}\|_2^2 = \langle Av_{\max}, Av_{\max}\rangle = \lambda_{\max}^2 \|v_{\max}\|_2^2 = \lambda_{\max}^2.$$

Ainsi, on conclut que $|||A|||_2^2 = \lambda_{\max}^2$.

Exemple 2 : Les normes induites par $\|\cdot\|_1$ et $\|\cdot\|_\infty$ sont en revanche explicites : on a

$$\|A\|_1 = \max\left\{\sum_{i=1}^n |a_{ij}|, 1 \leq j \leq n\right\},$$

$$\|A\|_\infty = \max\left\{\sum_{j=1}^n |a_{ij}|, 1 \leq i \leq n\right\}.$$

Exemple 3 : Toutes les normes sur $M_n(\mathbb{K})$ ne sont pas forcément induites par une norme sur $\mathbb{K}^n$. C'est par exemple le cas de la norme de Frobenius, qui est simplement la norme euclidienne sur

l'espace vectorielle $M_n(\mathbb{K})$ (vue comme $\mathbb{K}^{n^2}$ avec la base $(E_{k\ell})_{1\leq k,\ell\leq n}$ comme base orthonormée), donnée par

$$\|A\|_F = \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2}.$$

A noter que le produit scalaire associé est défini pour tout $A, B \in M_n(\mathbb{K})$ par

$$\langle A, B\rangle_F = \text{Trace}\left(A^T B\right).$$

1.2.2 Conditionnement d'une matrice

Definition 1.2.2. *Soit $\|\cdot\|$ une norme sur $\mathbb{K}^n$ et soit $A \in GL_n(\mathbb{K})$. Le conditionnement de A par rapport à la norme $\|\cdot\|$ est défini par*

$$Cond(A) = |||A|||\,|||A^{-1}|||.$$

Le conditionnement d'une matrice caractérise la sensibilité du système linéaire associé à (1.1) face aux variations de b, ce que nous donne la proposition suivante :

Proposition 1.2.2. *Soit $A \in GL_n(\mathbb{K})$ et $b, b' \in \mathbb{K}^n$. Les solutions respectives x et x' des systèmes $Ax = b$ et $Ax' = b'$ satisfont l'inégalité d'erreur relative suivante :*

$$\frac{\|x - x'\|}{\|x\|} \leq Cond(A)\frac{\|b - b'\|}{\|b\|}.$$

Démonstration. En effet, nous avons :

$$\|x - x'\| = \|A^{-1}(b - b')\| \leq |||A^{-1}|||\|b - b'\|.$$

Par ailleurs,

$$\|b\| = \|Ax\| \leq |||A|||\|x\|,$$

ce qui implique que $\frac{1}{\|x\|} \leq \frac{|||A|||}{\|b\|}$. En combinant ces deux inégalités, on obtient :

$$\frac{\|x - x'\|}{\|x\|} \leq |||A|||\,|||A^{-1}|||\frac{\|b - b'\|}{\|b\|},$$

ce qui conclut la preuve par définition de Cond(A). □

En d'autres termes, une matrice avec un faible conditionnement est dite bien conditionnée, ce qui signifie que de petites perturbations de b induisent de faibles variations dans la solution x, assurant ainsi une meilleure stabilité numérique. En revanche, une matrice avec un conditionnement élevé est dite mal conditionnée, ce qui peut entraîner des problèmes de stabilité significatifs dans les calculs numériques.

1.3 Méthode d'élimination de Gauss

Nous introduisons ici une première méthode numérique pour résoudre efficacement l'équation (1.1) lorsque A est inversible.

Remarque 1.3.1. *Il serait irréaliste de chercher à utiliser* (1.3) *pour calculer explicitement* A^{-1}. *En effet, le calcul du déterminant de* A *implique une somme sur toutes les permutations possibles des* n *indices, avec un nombre de permutations égal à* $n! = n \cdot (n-1) \cdots 2 \cdot 1$. *Cela entraîne une complexité exponentielle en* n, *nécessitant par exemple environ* $2,5 \cdot 10^{18}$ *opérations pour inverser une matrice de taille* 20. *Avec un processeur d'ordinateur fonctionnant à 1 GHz, le temps de calcul s'élèverait à environ* $2,5 \cdot 10^9$ *secondes, soit environ 77 ans.*

La méthode classique de résolution de l'équation $Ax = b$ est l'élimination de Gauss. Elle consiste à transformer le système

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n, \end{cases} \tag{1.4}$$

en un système triangulaire supérieur :

$$\begin{cases} u_{11}x_1 + u_{12}x_2 + \cdots + u_{1n}x_n = b'_1, \\ \vdots \\ u_{(n-1)(n-1)}x_{n-1} + u_{(n-1)n}x_n = b'_{n-1}, \\ u_{nn}x_n = b'_n, \end{cases} \tag{1.5}$$

où $u_{ii} \neq 0$ pour tout $1 \leq i \leq n$, en appliquant des opérations élémentaires sur les lignes du système. Ce processus de "triangularisation" permet de simplifier la résolution, car on peut ensuite résoudre les équations de la dernière ligne vers la première.

1.3.1 Méthode sans pivot

L'algorithme de la méthode sans pivot fonctionne comme suit : on commence par poser $A^{(0)} = A$ et $b^{(0)} = b$. À chaque étape $1 \leq \ell \leq n-1$, on définit les matrices $A^{(\ell)}$ et le vecteur $b^{(\ell)}$ comme suit :

$$a_{ij}^{(\ell)} = \begin{cases} a_{ij}^{(\ell-1)} & \text{si } i \leq \ell \text{ ou } j < \ell, \\ a_{ij}^{(\ell-1)} - \frac{a_{i\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}} a_{\ell j}^{(\ell-1)} & \text{si } i > \ell \text{ et } j \geq \ell, \end{cases} \quad b_i^{(\ell)} = \begin{cases} b_i^{(\ell-1)} & \text{si } i \leq \ell, \\ b_i^{(\ell-1)} - \frac{a_{i\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}} b_i^{(\ell-1)} & \text{si } i > \ell. \end{cases} \tag{$\mathcal{L}_\ell$}$$

Cela n'est possible que si $a_{\ell\ell}^{(\ell-1)} \neq 0$ (nous verrons plus loin comment traiter le cas où $a_{\ell\ell}^{(\ell-1)} = 0$). En d'autres termes, on laisse inchangées les ℓ premières lignes de $A^{(\ell-1)}$ et $b^{(\ell-1)}$, ainsi que les $\ell - 1$ premières colonnes de $A^{(\ell-1)}$.

Remarquons qu'à l'étape ℓ, la matrice $A^{(\ell)} = (a_{ij}^{(\ell)})_{1\le i,j\le n}$ satisfait la condition suivante :

$$a_{ij}^{(\ell)} = 0 \quad \text{pour} \quad j < i \text{ ou } j \le \ell,$$

ce qui implique que la matrice $A^{(n-1)}$ est triangulaire supérieure.

Autre manière de réécrire l'algorithme : Pour la matrice $A^{(\ell)}$, on peut reformuler l'algorithme de la manière suivante :

$$A^{(\ell)}[i,] = \begin{cases} A^{(\ell-1)}[i,] & \text{si } i \le \ell, \\ A^{(\ell-1)}[i,] - \frac{a_{i\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}} A^{(\ell-1)}[\ell,] & \text{sinon,} \end{cases}$$

et on a toujours

$$b_i^{(\ell)} = \begin{cases} b_i^{(\ell-1)} & \text{si } i \le \ell, \\ b_i^{(\ell-1)} - \frac{a_{i\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}} b_i^{(\ell-1)} & \text{si } i > \ell. \end{cases}$$

Cela donne la procédure classique de l'élimination de Gauss. Dans la suite, nous noterons $\Delta_{i,\ell} = \frac{a_{i\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}}$.

En posant $u_{ij} = A^{(n-1)}[i,j]$ pour $1 \le i \le j \le n$ et $b'_i = b^{(n-1)}[i]$, nous avons transformé le système linéaire (1.4) en un système triangulaire supérieur (1.5). La solution du premier système coïncide avec celle du second. Le fait que A soit inversible assure que $u_{nn} \neq 0$. L'avantage est que le système triangulaire supérieur est facile à résoudre en procédant par substitution rétroactive, en partant de $i = n$:

$$x_n = \frac{b'_n}{u_{nn}}, \quad x_i = \frac{1}{u_{ii}} \left(b'_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \quad \text{pour} \quad 1 \le i \le n-1. \tag{$\mathcal{R}$}$$

Exemple : Considérons le système $Ax = b$ avec

$$A = \begin{pmatrix} 3 & 5 & 0 \\ 6 & 7 & 1 \\ 3 & 2 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix}.$$

Après application de la méthode sans pivot, nous obtenons : On obtient donc

$$A^{(1)} = \begin{pmatrix} 3 & 5 & 0 \\ 0 & -3 & 1 \\ 0 & -3 & 3 \end{pmatrix} \begin{matrix} \leftarrow L_1 \\ \leftarrow L_2 - 2L_1 \\ \leftarrow L_3 - L_1 \end{matrix} \qquad b^{(1)} = \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix} \begin{matrix} \leftarrow L_1 \\ \leftarrow L_2 - 2L_1 \\ \leftarrow L_3 - L_1 \end{matrix}$$

$$A^{(2)} = \begin{pmatrix} 3 & 5 & 0 \\ 0 & -3 & 1 \\ 0 & 0 & 4 \end{pmatrix} \begin{matrix} \leftarrow L_1 \\ \leftarrow L_2 \\ \leftarrow L_3 - L_2 \end{matrix} \qquad b^{(2)} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \begin{matrix} \leftarrow L_1 \\ \leftarrow L_2 \\ \leftarrow L_3 - L_2 \end{matrix}$$

On cherche donc à résoudre

$$\begin{pmatrix} 3 & 5 & 0 \\ 0 & -3 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \Leftrightarrow \begin{cases} 3x_1 + 5x_2 & = 0 \\ -3x_2 + x_3 & = 1 \\ 2x_3 & = 2 \end{cases} \Leftrightarrow \begin{cases} x_1 = 0 \\ x_2 = 0 \\ x_3 = 1 \end{cases}$$

On peut vérifier que l'on a bien

$$\begin{pmatrix} 3 & 5 & 0 \\ 6 & 7 & 1 \\ 3 & 2 & 3 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix}.$$

1.3.2 Méthode avec pivot

La méthode précédente nécessite que, à chaque étape $1 \leq \ell \leq n-1$, l'élément $a_{\ell\ell}^{(\ell-1)}$ soit non nul. Cependant, si cette condition n'est pas remplie pour un certain ℓ, il suffit de permuter la ℓ-ième ligne de $A^{(\ell-1)}$ avec une ligne $i_\ell > \ell$ telle que $a_{i_\ell \ell}^{(\ell-1)} \neq 0$. Ensuite, on applique l'opération ($\mathcal{L}_\ell$) à la nouvelle matrice. Cela donne l'opération suivante :

$$a_{ij}^{(\ell)} = \begin{cases} a_{ij}^{(\ell-1)} & \text{si } i < \ell \text{ ou } j < \ell, \\ a_{i_\ell j}^{(\ell-1)} & \text{si } i = \ell \text{ et } j \geq \ell, \\ a_{\sigma(i)j}^{(\ell-1)} - \dfrac{a_{\sigma(i)\ell}^{(\ell-1)}}{a_{i_\ell \ell}^{(\ell-1)}} a_{i_\ell j}^{(\ell-1)} & \text{si } i > \ell \text{ et } j \geq \ell. \end{cases} \quad b_i^{(\ell)} = \begin{cases} b_i^{(\ell-1)} & \text{si } i < \ell, \\ b_{i_\ell}^{(\ell-1)} & \text{si } i = \ell, \\ b_{\sigma(i)}^{(\ell-1)} - \dfrac{a_{\sigma(i)\ell}^{(\ell-1)}}{a_{i_\ell \ell}^{(\ell-1)}} b_{\sigma(i)}^{(\ell-1)} & \text{si } i > \ell. \end{cases} \quad (\mathcal{L}'_\ell)$$

Ici, σ représente la transposition qui échange ℓ et i_ℓ, c'est-à-dire

$$\sigma(\ell) = i_\ell, \quad \sigma(i_\ell) = \ell \quad \text{et} \quad \sigma(i) = i \quad \text{pour} \quad i \notin \{\ell, i_\ell\}.$$

Le choix de l'indice i_ℓ peut sembler arbitraire au premier abord, mais pour des raisons de stabilité numérique de la méthode de Gauss, il est recommandé de choisir i_ℓ tel que $|a_{i_\ell \ell}^{(\ell-1)}|$ soit maximal. En effet, dans l'étape $\mathcal{L}'_\ell$, on divise par $a_{i_\ell \ell}^{(\ell-1)}$, et un petit $|a_{i_\ell \ell}^{(\ell-1)}|$ pourrait provoquer des erreurs d'arrondi. Le choix optimal de i_ℓ est donc donné par

$$i_\ell = \arg \max_{\ell \leq i \leq n} |a_{i\ell}^{(\ell-1)}|. \tag{1.6}$$

En pratique, la méthode avec pivot est souvent préférable, même lorsque la méthode sans pivot aurait suffi, pour garantir une meilleure stabilité de l'algorithme.

Autre façon d'écrire l'algorithme : On peut également réécrire l'algorithme sous la forme sui-

vante :

$$A^{(\ell)}[i,] = \begin{cases} A^{(\ell-1)}[i,] & \text{si } i < \ell, \\ A^{(\ell-1)}[i_\ell,] & \text{si } i = \ell, \\ A^{(\ell-1)}[i,] - \Delta_{i,\ell} A^{(\ell-1)}[i_\ell,] & \text{si } i > \ell \text{ et } i \neq i_\ell, \\ A^{(\ell-1)}[\ell,] - \Delta_{i,\ell} A^{(\ell-1)}[i_\ell,] & \text{si } i = i_\ell. \end{cases}$$

avec

$$\Delta_{i,\ell} = \frac{a^{(\ell-1)}_{\sigma(i)\ell}}{a^{(\ell-1)}_{i_\ell \ell}}.$$

Exemple : Considérons le système $Ax = b$ avec

$$A = \begin{pmatrix} 2 & 1 & -1 \\ -2 & -1 & 0 \\ 4 & 3 & -1 \end{pmatrix}, \qquad \text{et} \quad b = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}.$$

La méthode de Gauss avec pivot nous donne

$$A^{(1)} = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \begin{matrix} \\ \leftarrow L_2 + L_1, \\ \leftarrow L_3 - 2L_1. \end{matrix} \qquad b^{(1)} = \begin{pmatrix} 2 \\ 1 \\ -4 \end{pmatrix} \begin{matrix} \leftarrow L_1, \\ \leftarrow L_2 + L_1, \\ \leftarrow L_3 - 2L_1. \end{matrix}$$

$$A^{(2)} = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{pmatrix} \begin{matrix} \leftarrow L_1, \\ \leftarrow L_3, \\ \leftarrow L_2. \end{matrix} \qquad b^{(2)} = \begin{pmatrix} 2 \\ -4 \\ 1 \end{pmatrix} \begin{matrix} \leftarrow L_1, \\ \leftarrow L_3, \\ \leftarrow L_2. \end{matrix}$$

On résout alors le système triangulaire supérieur :

$$\begin{pmatrix} 2 & 1 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -4 \\ 1 \end{pmatrix} \Leftrightarrow \begin{cases} 2x_1 + x_2 - x_3 = 2, \\ x_2 + x_3 = -4, \\ -x_3 = 1. \end{cases} \Leftrightarrow \begin{cases} x_1 = 2, \\ x_2 = -3, \\ x_3 = -1. \end{cases}$$

Enfin, on vérifie que la solution est correcte :

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

1.3.3 Coût de la méthode de Gauss

Chaque étape $\mathcal{L}'_\ell$ de la méthode nécessite $O((n-\ell)^2)$ opérations, ce qui implique que la transformation en un système triangulaire nécessite un total de $O(n^3)$ opérations. L'opération finale, correspondant à la résolution du système triangulaire, nécessite $O(n^2)$ opérations supplémentaires. En conséquence, la méthode de Gauss résout le système $Ay = b$ en un temps global de $O(n^3)$

opérations.
Ce coût peut toutefois être réduit si la matrice A contient de nombreuses entrées nulles. Par exemple, si A est tridiagonale (ce qui est relativement fréquent dans certains problèmes), la résolution du système $Ay = b$ peut être effectuée en $O(n^2)$ opérations.
Une analyse plus détaillée de la complexité permet de montrer que, dans le cas général, la méthode de Gauss est réalisée en environ $\frac{2}{3}n^3$ opérations, ce qui est asymptotiquement plus précis.

1.4 Décomposition LU

La méthode de Gauss est une approche rapide pour résoudre un système linéaire donné, tel que (1.1). Cependant, il arrive fréquemment que l'on ait à résoudre non pas un seul système, mais une famille de systèmes linéaires, tous de la forme $Ay^{(i)} = b^{(i)}$, où A est une matrice fixe et $\left(b^{(i)}\right)_{1\leq i\leq m}$ représente une série de données avec pour inconnues $\left(y^{(i)}\right)$. C'est typiquement le cas dans le processus d'inversion de A, en prenant $b^{(i)} = e_i$, avec e_i le i-ème vecteur de la base canonique. Il devient donc pertinent d'automatiser la méthode de Gauss pour éviter de refaire la mise en forme triangulaire à chaque nouvelle résolution. La décomposition LU permet précisément cette automatisation.

1.4.1 Théorème de décomposition et conséquences

Nous énonçons le théorème de décomposition LU et son application principale. La démonstration de cette décomposition sera donnée dans la section suivante.

On rappelle qu'une matrice $A = (a_{ij})_{1\leq i,j\leq n}$ est dite triangulaire supérieure (resp. inférieure) si $a_{ij} = 0$ pour $i > j$ (resp. $i < j$). Notons que le produit de deux matrices triangulaires inférieures (resp. supérieures) reste triangulaire inférieure (resp. supérieure), et que la diagonale du produit est le produit des diagonales. Une matrice triangulaire est donc inversible si et seulement si tous ses éléments diagonaux sont non nuls. Elle est dite unitriangulaire si elle est triangulaire et a des 1 sur sa diagonale.

Une matrice $P = (p_{ij})_{1\leq i,j\leq n}$ de taille n est une matrice de permutation s'il existe une permutation σ de $\{1, \ldots, n\}$ telle que $p_{ij} = \delta_{j\sigma(i)}$. On note P_σ la matrice de permutation associée à σ.

La décomposition LU permet d'écrire une matrice A inversible comme le produit d'une matrice U triangulaire supérieure et d'une matrice L triangulaire inférieure. Comme nous le verrons, cette décomposition est possible, à une permutation près des lignes de A.

Théorème 1.4.1 (Décomposition $PA = LU$)**.** *Soit A une matrice inversible de taille $n \geq 1$. Il existe une matrice de permutation P, une matrice triangulaire supérieure inversible U et une matrice triangulaire inférieure L avec des 1 sur la diagonale telles que*

$$PA = LU.$$

Au-delà du théorème, sa preuve (section suivante) offre un intérêt pratique : elle fournit un moyen explicite de construire les matrices P, L, et U. Pour le moment, nous pouvons juste retenir que la décomposition LU est une manière de décrire la méthode de Gauss. La matrice U est exactement la matrice U obtenue par la méthode de Gauss en section précédente, la matrice P enregistre les choix successifs des i_ℓ pour $1 \leq \ell \leq n-1$ et la matrice L garde en mémoire les opérations successives sur les lignes que nous avons faites pour transformer A en U.

Une variante équivalente du théorème, moins courante mais plus simple à démontrer, stipule que l'on peut décomposer A comme $AP = LU$, où P est une matrice de permutation, L est triangulaire inférieure avec des 1 sur la diagonale, et U est triangulaire supérieure. Dans cette variante, la multiplication par P correspond à une permutation des colonnes de A et non des lignes.

Il est important de noter que la décomposition LU d'une matrice n'est généralement pas unique, en raison principalement du choix de la matrice de permutation P. On peut cependant montrer que pour chaque choix de P, il existe au plus un couple de matrices (L, U) — où L est une matrice triangulaire inférieure avec des 1 sur la diagonale (dite *unitriangulaire*) et U est triangulaire supérieure — tel que $PA = LU$. D'un point de vue algorithmique, on peut choisir P de manière optimale pour assurer que $Cond(L)$ et $Cond(U)$ ne soient pas trop grand.

Avant de le prouver, mentionnons pourquoi une telle décomposition nous est utile. Rappelons nous que l'objectif est de résoudre le système

$$Ax = b,$$

avec $b \in \mathbb{K}^n$. Supposons que nous ayons trouvé L, U respectivement triangulaire inférieure et supérieure et P matrice de permutation telles que $PA = LU$. Il faut donc résoudre le système

$$LUx = PAx = Pb.$$

Nous allons successivement résoudre l'équation $Ly = Pb$ (*algorithme de descente*) puis $Ux = y$ (*algorithme de remontée*). Remarquons que nous aurons alors

$$Ax = P^{-1}LUx = P^{-1}Ly = P^{-1}Pb = b,$$

ce qui résoudra notre problème. Soit σ la permutation telle que $P = P_\sigma$. Nous avons alors pour $1 \leq i \leq n$

$$(Pb)_i = \sum_{j=1}^{n} \delta_{j\sigma(i)} b_j = b_{\sigma(i)}.$$

Plus précisément, une fois la décomposition $PA = LU$ obtenue, on peut retrouver la solution x à l'aide des deux étapes suivantes :

Algorithme de descente : Comme $L = (l_{ij})_{1 \leq i,j \leq n}$ est triangulaire inférieure (avec des 1 sur la

diagonale), le système $Ly = b$ est équivalent à

$$\begin{cases} y_1 & & & & & & & = & b_{\sigma(1)} \\ l_{21}y_1 & + & y_2 & & & & & = & b_{\sigma(2)} \\ \vdots & & \vdots & & \ddots & & & = & \vdots \\ l_{n1}y_1 & + & \dots & + & l_{n(n-1)}y_{n-1} & + & y_n & = & b_{\sigma(n)} \end{cases}$$

Il peut être résolu par récurrence en posant $y_1 = b_{\sigma(1)}$ puis $y_i = b_{\sigma(i)} - \sum_{k=1}^{i-1} l_{ik}y_k$. Cette solution est obtenue en $O(n^2)$ opérations.

Algorithme de remontée : De même, l'équation $Ux = y$ est équivalente au système

$$\begin{cases} u_{11}x_1 & + & \dots & + & u_{1(n-1)}x_{n-1} & + & u_{1n}x_n & = & y_1 \\ & & \ddots & & \vdots & & \vdots & = & \vdots \\ & & & & u_{(n-1)(n-1)}x_{n-1} & + & u_{(n-1)n}x_n & = & y_{n-1} \\ & & & & & & u_{nn}x_n & = & y_n \end{cases}$$

Étant donné que U est inversible, ses éléments diagonaux vérifient $u_{ii} \neq 0$ pour $1 \leq i \leq n$. Ce système est donc facilement résoluble : on peut poser $x_n = y_n/u_{nn}$, puis, par récurrence, calculer chaque x_k avec la formule

$$x_k = \frac{1}{u_{kk}} \left(y_k - \sum_{i=k+1}^{n} u_{ki}x_i \right).$$

Cette solution est obtenue en un nombre d'opérations de l'ordre de $O(n^2)$.

Ainsi, une fois la décomposition LU d'une matrice A réalisée, la résolution de l'équation $Ax = b$ se fait en $O(n^2)$ opérations.

1.4.2 Preuve de la décomposition $A = LU$

Nous nous intéressons ici à la preuve de la décomposition $A = LU$ dans le cas où $P = I_n$, c'est-à-dire sans pivot. Pour simplifier les notations et les arguments, nous allons introduire les vecteurs de Gauss ainsi que les matrices associées, pour chaque étape de l'algorithme de Gauss sans pivot.

Definition 1.4.1 (Vecteurs et matrice de Gauss)**.** *À chaque étape ℓ de la méthode de Gauss (sans pivot), les lignes de la matrice $A^{(\ell)}$ sont mises à jour selon*

$$A^{(\ell)}[i,] = \begin{cases} A^{(\ell-1)}[i,] & si\ i \leq \ell \\ A^{(\ell-1)}[i,] - \Delta_{i,\ell} A^{(\ell-1)}[\ell,] & si\ i > \ell \end{cases} \tag{1.7}$$

où $\Delta_{i,\ell} = \frac{a_{i,\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}}$. *Le vecteur de Gauss* τ_ℓ *est défini par*

$$\tau_\ell = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \Delta_{\ell+1,\ell} \\ \vdots \\ \Delta_{n,\ell} \end{pmatrix},$$

et enregistre les transformations appliquées aux lignes de A *à l'étape* ℓ. *En utilisant la base canonique* $\{e_1, \ldots, e_n\}$ *de* $\mathbb{K}^n$, *la matrice de Gauss associée à* τ_ℓ *est définie par*

$$M_\ell = I_n - \tau_\ell e_\ell^T = \begin{pmatrix} 1 & 0 & \ldots & & & \\ 0 & \ddots & 0 & & & \\ \vdots & & 1 & \ddots & & \\ \vdots & \vdots & -\Delta_{\ell+1,\ell} & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ & & -\Delta_{n,\ell} & 0 & \ldots & 1 \end{pmatrix}$$

La proposition suivante permet de reconstituer la matrice triangulaire inférieure L.

Proposition 1.4.1. *Les matrices* M_ℓ *vérifient les propriétés suivantes :*

1. *La matrice* M_ℓ *est inversible et*

$$L_\ell := M_\ell^{-1} = I_n + \tau_\ell e_\ell^T = \begin{pmatrix} 1 & 0 & \ldots & & & \\ 0 & \ddots & 0 & & & \\ \vdots & & 1 & \ddots & & \\ \vdots & \vdots & \Delta_{\ell+1,\ell} & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ & & \Delta_{n,\ell} & 0 & \ldots & 1 \end{pmatrix}$$

2. *Soient* $M = M_{n-1}M_{n-2}\dots M_1$ *et* $L = L_1L_2,\dots L_{n-1}$, *on a* $LM = I_n$ *et*

$$L = I_n + \sum_{\ell=1}^{n-1} \tau_\ell e_\ell^T = \begin{pmatrix} 1 & & \dots & & & \\ \Delta_{2,1} & \ddots & 0 & & & \\ \vdots & & 1 & \ddots & & \\ \vdots & \vdots & \Delta_{\ell+1,\ell} & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ \Delta_{n,1} & & \Delta_{n,\ell} & \Delta_{n,\ell+1} & \dots & 1 \end{pmatrix}$$

Démonstration. On vérifie facilement que $e_\ell^T\tau_\ell = 0$, et donc

$$L_\ell M_\ell = \left(I_n + \tau_\ell e_\ell^T\right)\left(I_n - \tau_\ell e_\ell^T\right) = I_n - \tau_\ell e_\ell^T\tau_\ell e_\ell^T = I_n - \tau_\ell\left(e_\ell^T\tau_\ell\right)e_\ell^T = I_n.$$

On vérifie de la même façon que $\left(I_n - \tau_\ell e_\ell^T\right)\left(I_n + \tau_\ell e_\ell^T\right) = I_n$. On a donc

$$L_{n-1}\dots L_2\underbrace{L_1M_1}_{=I_n}M_2\dots M_{n-1} = L_{n-1}\dots\underbrace{L_2M_2}_{=I_n}\dots M_{n-1} = L_{n-1}M_{n-1} = I_n.$$

De plus, pour tout $\ell < \ell'$, on a $e_\ell^T\tau_{\ell'} = 0$. Par induction, on obtient donc

$$\begin{aligned} L &= \left(I_n + \tau_1e_1^T\right)\dots \underbrace{\left(I_n + \tau_{n-2}e_{n-2}^T\right)\left(I_n + \tau_{n-1}e_{n-1}^T\right)}_{=I_n+\tau_{n-2}e_{n-2}^T+\tau_{n-1}e_{n-1}^T+\tau_{n-2}\underbrace{e_{n-2}^T\tau_{n-1}}_{=0}e_{n-1}^T} \\ &= \left(I_n + \tau_1e_1^T\right)\dots \underbrace{\left(I_n + \tau_{n-3}e_{n-3}^T\right)\left(I_n + \tau_{n-2}e_{n-2}^T + \tau_{n-1}e_{n-1}^T\right)}_{I_n+\tau_{n-3}e_{n-3}^T+\tau_{n-2}e_{n-2}^T+\tau_{n-1}e_{n-1}^T+\tau_{n-3}\underbrace{e_{n-3}^T\tau_{n-2}}_{0}e_{n-2}^T+\tau_{n-3}\underbrace{e_{n-3}^T\tau_{n-1}}_{=0}e_{n-1}^T} \\ &= \left(I_n + \tau_1e_1^T\right)\dots\left(I_n + \tau_{n-4}e_{n-4}^T\right)\left(I_n + \tau_{n-3}e_{n-3}^T + \tau_{n-2}e_{n-2}^T + \tau_{n-1}e_{n-1}^T\right) \\ &\vdots \\ &= I_n + \tau_1e_1^T + \dots + \tau_{n-1}e_{n-1}^T. \end{aligned}$$

□

Afin de finir la preuve de la décomposition $A = LU$, il suffit maintenant de vérifier que l'on a $U = M_{n-1}\dots M_1A$, i.e que pour tout ℓ, on a $A^{(\ell)} = M_\ell A^{(\ell-1)}$. Pour cela, par définition du produit matriciel, on vérifie facilement que $M_\ell A^{(\ell-1)}$ est la matrice dont la i-ème ligne est donnée par $M_\ell[i,]A^{(\ell-1)}$. Comme les ℓ première lignes de M_ℓ correspondent aux ℓ premières lignes de la matrices identité, on a pour tout $i \leq \ell$

$$M_\ell[i,]A^{(\ell-1)} = e_i^TA^{(\ell-1)} = A^{(\ell-1)}[i,]$$

i.e M_ℓ laisse les ℓ premières lignes inchangées. Pour tout $i > \ell$, on a $M_\ell[i,] = e_i^T - \Delta_{i,\ell} e_\ell^T$ et donc

$$M_\ell[i,]A^{(\ell-1)} = e_i^T A^{(\ell-1)} - \Delta_{i,\ell} e_\ell^T A^{(\ell-1)} = A^{(\ell-1)}[i,] - \Delta_{i,\ell} A^{(\ell-1)}[\ell,]$$

D'après l'équation (1.7), faire $M_\ell A^{(\ell-1)}$ correspond donc bien à faire la ℓ-ème étape de l'algorithme de Gauss (sans pivot).

Exemple : On considère $Ax = b$ avec

$$A = \begin{pmatrix} 3 & 5 & 0 \\ 6 & 7 & 1 \\ 3 & 2 & 3 \end{pmatrix}, \qquad \text{et} \qquad b = \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix}.$$

On obtient donc

$$A^{(1)} = \begin{pmatrix} 3 & 5 & 0 \\ 0 & -3 & 1 \\ 0 & -3 & 3 \end{pmatrix} \begin{matrix} \leftarrow L_1 \\ \leftarrow L_2 - 2L_1 \\ \leftarrow L_3 - L_1 \end{matrix} \qquad L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{matrix} \\ \leftarrow \Delta_{2,1} = 2 \\ \leftarrow \Delta_{3,1} = 1 \end{matrix}$$

$$U = A^{(2)} = \begin{pmatrix} 3 & 5 & 0 \\ 0 & -3 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{matrix} \leftarrow L_1 \\ \leftarrow L_2 \\ \leftarrow L_3 - L_2 \end{matrix} \qquad L = L_1 L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{matrix} \\ \\ \leftarrow \Delta_{3,2} = 1 \end{matrix}$$

On vérifie bien que l'on a

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 3 & 5 & 0 \\ 0 & -3 & 1 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 3 & 5 & 0 \\ 6 & 7 & 1 \\ 3 & 2 & 3 \end{pmatrix}$$

On résout donc $Ly = b$, i.e on a

$$\begin{cases} y_1 = 0 \\ 2y_1 + y_2 = 1 \\ y_1 + y_2 + y_3 = 3 \end{cases} \Leftrightarrow \begin{cases} y_1 = 0 \\ y_2 = 1 \\ y_3 = 2 \end{cases}$$

Et on résout maintenant $Ux = y$, i.e

$$\begin{pmatrix} 3 & 5 & 0 \\ 0 & -3 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \Leftrightarrow \begin{cases} 3x_1 + 5x_2 = 0 \\ -3x_2 + x_3 = 1 \\ 2x_3 = 2 \end{cases} \Leftrightarrow \begin{cases} x_1 = 0 \\ x_2 = 0 \\ x_3 = 1 \end{cases}$$

1.4.3 Preuve de la décomposition $PA = LU$

Avant de donner la preuve de la décomposition $PA = LU$, illustrons son utilité par rapport à la décomposition $A = LU$. En effet, la méthode sans pivot peut rencontrer des difficultés dans cer-

tains cas, notamment si l'on a $A^{(\ell-1)}[\ell,\ell] = 0$, car cela implique une division par zéro impossible. Dans de telles situations, la méthode sans pivot devient inapplicable.
Une autre raison d'utiliser la décomposition $PA = LU$ est pour des questions de stabilité numérique. Prenons l'exemple suivant [1] :

$$A = \begin{pmatrix} 0.0001 & 1 \\ 1 & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 10000 & 1 \end{pmatrix}}_{=L} \underbrace{\begin{pmatrix} 0.0001 & 1 \\ 0 & 10000 \end{pmatrix}}_{=U}.$$

Les grandes différences d'échelle dans les éléments de cette matrice peuvent entraîner de nombreuses complications numériques. Dans ce cas, on a les conditionnements $Cond(U) = Cond(A) = 10^8$. Ces valeurs de conditionnement très élevées peuvent rendre la résolution numérique instable. Pour surmonter ce problème, on peut utiliser la méthode de pivot, qui consiste à sélectionner, à chaque étape, l'élément de plus grande valeur comme pivot. En appliquant cette méthode, on obtient :

$$\underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{=P} \underbrace{\begin{pmatrix} 0.0001 & 1 \\ 1 & 0 \end{pmatrix}}_{=A} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0.0001 & 1 \end{pmatrix}}_{=L} \underbrace{\begin{pmatrix} 1 & 1 \\ 0 & 0.999 \end{pmatrix}}_{=U}$$

Ainsi, les problèmes liés aux différences d'échelle sont résolus, avec un conditionnement de L réduit à $Cond(L) = 1.0001$ et $Cond(U) \approx 2.61$, ce qui est beaucoup plus stable.

Pour prouver la décomposition $PA = LU$, définissons les matrices de permutation deux à deux, qui joueront un rôle clé.

Definition 1.4.2. *Une matrice de permutation deux à deux, notée P_{jk}, est définie par :*

$$\begin{aligned} &P_{jk}[i,i] = 1 \quad \textit{si } i \neq j,k, \\ &P_{jk}[i,i] = 0 \quad \textit{si } i = j \textit{ ou } i = k, \\ &P_{jk}[i,i'] = 0 \quad \textit{si } i \neq i' \textit{ et } ((i,i') \neq (j,k) \textit{ ou } (i,i') \neq (k,j)), \\ &P_{jk}[j,k] = P_{jk}[k,j] = 1. \end{aligned}$$

En d'autres termes, une matrice de permutation est obtenue en prenant la matrice identité et en permutant deux de ses lignes (ou, de manière équivalente, deux de ses colonnes).
Par exemple, on a :

$$P_{24} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

1. Cet exemple est tiré du cours de B. Gaüzère, Gilles Gasso, et Stéphane Canu.

qui est la matrice identité dans laquelle les deuxièmes et quatrièmes lignes (ou colonnes) ont été permutées.

Les matrices de permutation vérifient les propriétés suivantes :

- Pour toute matrice A, $P_{ij}A$ est la matrice A où les lignes i et j sont permutées.
- Pour toute matrice A, AP_{ij} est la matrice A où les colonnes i et j sont permutées.
- La matrice P_{ij} vérifie :
$$P_{ij} = P_{ij}^T = P_{ij}^{-1}.$$

De la même façon que pour la décomposition $A = LU$, on peut alors remarquer que la ℓ-ème opération dans la méthode de Gauss avec pivot peut s'écrire

$$A^{(\ell)} = M_\ell P^{(\ell)} A^{(\ell-1)}$$

où $P^{(\ell)}$ est la permutation utilisée pour le pivot, et M_ℓ est la matrice de Gauss définie par

$$M_\ell = I_n - \tau_\ell e_\ell^T = \begin{pmatrix} 1 & 0 & \dots & & & \\ 0 & \ddots & 0 & & & \\ \vdots & & 1 & \ddots & & \\ \vdots & \vdots & -\Delta_{\ell+1,\ell} & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ & & -\Delta_{n,\ell} & 0 & \dots & 1 \end{pmatrix}$$

avec, si $P^{(\ell)} = P_{i_\ell \ell}$,

$$\Delta_{i,\ell} = \frac{a_{\sigma(i)\ell}^{(\ell-1)}}{a_{i_\ell \ell}^{(\ell-1)}}$$

On a ainsi

$$U = M_{n-1} P^{(n-1)} \dots M_2 P^{(2)} M_1 P^{(1)} A.$$

L'objectif est maintenant de réussir à réécrire cela sous la forme $U = L^{-1}PA$. Pour cela, on remarque que

$$\begin{aligned} U &= M_{n-1} P^{(n-1)} M_{n-2} P^{(n-2)} \dots M_1 P^{(1)} A \\ &= M_{n-1} \underbrace{P^{(n-1)} M_{n-2} P^{(n-1)}}_{=:\hat{M}_{n-2}} P^{(n-1)} P^{(n-2)} M_{n-3} P^{(n-3)} \dots \tilde{M}_1 P^{(1)} A \\ &= M_{n-1} \hat{M}_{n-2} \underbrace{P^{(n-1)} P^{(n-2)} M_{n-3} P^{(n-2)} P^{(n-1)}}_{=:\hat{M}_{n-3}} P^{(n-1)} P^{(n-2)} P^{(n-3)} M_{n-4} \dots M_1 P^{(1)} A \\ &\vdots \\ &= \hat{M} P A \end{aligned}$$

avec $P = P^{(n-1)} \dots P^{(1)}$, $\hat{M} = M_{n-1}\hat{M}_{n-2} \dots \hat{M}_1$, où

$$\hat{M}_\ell = P^{(n-1)} \dots P^{(\ell+1)} M_\ell P^{(\ell+1)} \dots P^{(n-1)}.$$

De la même façon que pour la décomposition $A = LU$, on peut facilement montrer que

$$M_{(\ell)}^{-1} = L_\ell := I_n + \tau_\ell e_\ell^T = \begin{pmatrix} 1 & 0 & \dots & & & \\ 0 & \ddots & 0 & & & \\ \vdots & & 1 & \ddots & & \\ \vdots & \vdots & \Delta_{\ell+1,\ell} & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ & & \Delta_{n,\ell} & 0 & \dots & 1 \end{pmatrix}$$

La proposition suivante nous donne une autre écriture de $\hat{M}^{(\ell)}$ ainsi que de son inverse.

Proposition 1.4.2. *On a*

$$\hat{M}_\ell = I_n - P^{(n-1)} \dots P^{(\ell+1)} \tau_\ell e_\ell^T \qquad \text{et} \qquad \hat{M}_\ell^{-1} =: \hat{L}_\ell = I_n + P^{(n-1)} \dots P^{(\ell+1)} \tau_\ell e_\ell^T$$

Preuve de la Proposition 1.4.2. On a

$$\hat{M}_\ell = P^{(n-1)} \dots P^{(\ell+1)} I_n P^{(\ell+1)} \dots P^{(n-1)} - P^{(n-1)} \dots P^{(\ell+1)} \left(\tau_\ell e_\ell^T\right) P^{(\ell+1)} \dots P^{(n-1)}.$$

Comme $P^{(\ell')} P^{(\ell')} = I_n$, il vient

$$P^{(n-1)} \dots P^{(\ell+1)} I_n P^{(l+1)} \dots P^{(n-1)} = P^{(n-1)} \dots P^{(\ell+2)} P^{(\ell+2)} \dots P^{(n-2)} = \dots = I_n$$

On a vu que la multiplication à droite par les matrices de permutation revient à faire des permutations sur les colonnes. Or pour tout $\ell' > \ell$, la permutation $P^{(\ell')}$ ne peut intervenir que sur des lignes strictement plus grandes que ℓ, et donc la multiplication à droite par $P^{(\ell')}$ ne fait intervenir que des permutation sur les colonnes strictement plus grande que ℓ. Or, ces colonnes, pour la matrice $\tau_\ell e_\ell^T$, sont toutes nulles. Ainsi, on a

$$\tau_\ell e_\ell^T P^{(\ell+1)} \dots P^{(n-1)} = \tau_\ell e_\ell^T$$

et on obtient donc le résultat pour $\hat{M}_\ell$. De plus, comme $P^{(n-1)} \dots P^{(\ell+1)} \tau_\ell$ ne consiste qu'en des permutations des composantes de τ_ℓ on peut montrer que $\hat{M}_\ell = \hat{L}_\ell$ de la même façon que pour la

décomposition $A = LU$, i.e on a

$$\begin{aligned}
\hat{M}_\ell \hat{L}_\ell &= I_n - P^{(n-1)} \dots P^{(\ell+1)} \tau_\ell e_\ell^T P^{(n-1)} \dots P^{(\ell+1)} \tau_\ell e_\ell^T \\
&= I_n - P^{(n-1)} \dots P^{(\ell+1)} \left(\tau_\ell e_\ell^T P^{(n-1)} \dots P^{(\ell+1)} \right) \tau_\ell e_\ell^T \\
&= I_n - P^{(n-1)} \dots P^{(\ell+1)} \tau_\ell e_\ell^T \tau_\ell e_\ell^T \\
&= I_n - P^{(n-1)} \dots P^{(\ell+1)} \tau_\ell \underbrace{\left(e_\ell^T \tau_\ell \right)}_{=0} e_\ell^T \\
&= I_n.
\end{aligned}$$

□

De la même façon, on a $M_{n-1}^{-1} = L_{n-1}$ où $L_{n-1} = I_n + \tau_{n-1} e_{n-1}^T$. Ainsi, de manière analogue à la décomposition $A = LU$, on a

$$L = \hat{M}^{-1} = \hat{L}_1 \dots \hat{L}_{n-2} L_{n-1} = I_n + \sum_{\ell=1}^{n-2} P^{(n-1)} \dots P^{(\ell+1)} \tau_\ell e_\ell^T + \tau_{n-1} e_{n-1}^T.$$

Pour résumer : L'obtention de la matrice U consiste en la matrice que l'on obtient avec la décomposition de Gauss avec pivot. La matrice $P = P^{(n-1)} \dots P^{(1)}$ est le produit de toute les permutations effectuées pour obtenir U. La seule "difficulté" est la construction de L, que l'on peut faire de la manière itérative suivante : partir de la matrice identité, et à l'itération ℓ

1. Pour les $\ell - 1$ premières colonnes, appliquer la permutation $P^{(\ell)}$.
2. Remplir la ℓ-ème colonne avec les coefficients $\tau_{i,\ell}$.

Résolution de $Ax = b$: On a $LUx = PAx = Pb$, et on cherche donc à résoudre $Ly = Pb$ avant de résoudre $Ux = y$.

Exemple 1 : On cherche à résoudre $Ax = b$ avec

$$A = \begin{pmatrix} 2 & 1 & -1 \\ -2 & -1 & 0 \\ 4 & 3 & 0 \end{pmatrix} \qquad \text{et} \qquad b = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}.$$

On fait d'abord la décomposition $PA = LU$. Afin de prendre les bons réflexes pour la suite, on fera les permutations nécessaires pour avoir le plus grand coefficient à chaque fois.

$$A^{(1)} = \begin{pmatrix} 4 & 3 & 0 \\ 0 & 1/2 & 0 \\ 0 & -1/2 & -1 \end{pmatrix} \begin{matrix} L_3 = L_1 \\ L_2 = L_1 + 1/2 L_3 \\ L_3 = L_1 - 1/2 L_3 \end{matrix} \quad L^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} \begin{matrix} \\ \Delta_{2,1} = -1/2 \\ \Delta_{3,1} = 1/2 \end{matrix} \quad P^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$U = A^{(2)} = \begin{pmatrix} 4 & 3 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad L_3 = L_3 + L_2 \qquad L = L^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/2 & -1 & 1 \end{pmatrix} \quad \Delta_{3,2} = -1 \qquad P^{(2)} = I_3$$

Ainsi, on obtient la décomposition $PA = LU$ avec

$$U = \begin{pmatrix} 4 & 3 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & -1 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/2 & -1 & 1 \end{pmatrix} \qquad P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

On vérifie que l'on a

$$PA = \begin{pmatrix} 4 & 3 & 0 \\ -2 & -1 & 0 \\ 2 & 1 & -1 \end{pmatrix} = LU = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/2 & -1 & 1 \end{pmatrix} \begin{pmatrix} 4 & 3 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 4 & 3 & 0 \\ -2 & -1 & 0 \\ 2 & 1 & -1 \end{pmatrix}$$

On va donc d'abord résoudre $Ly = Pb$ avec $Pb = (0, -1, 2)^T$. On a donc

$$\begin{cases} y_1 &= 0 \\ -1/2y_1 + y_2 &= -1 \\ 1/2y_1 - y_2 + y_3 &= 2 \end{cases} \Leftrightarrow \begin{cases} y_1 &= 0 \\ y_2 &= -1 \\ y_3 &= 1 \end{cases}$$

On résout maintenant $Ux = y$, ce qui nous donne

$$\begin{cases} 4x_1 + 3x_2 &= 0 \\ 1/2x_2 &= -1 \\ -x_3 = 1 \end{cases} \Leftrightarrow \begin{cases} x_1 &= 3/2 \\ x_2 &= -2 \\ x_3 &= -1 \end{cases} \leftrightarrow x = \begin{pmatrix} 3/2 \\ -2 \\ -1 \end{pmatrix}$$

On vérifie bien que l'on a

$$\begin{pmatrix} 2 & 1 & -1 \\ -2 & -1 & 0 \\ 4 & 3 & 0 \end{pmatrix} \begin{pmatrix} 3/2 \\ -2 \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}$$

Exemple 2 : On considère la matrice

$$A = \begin{pmatrix} 3 & 17 & 10 \\ 2 & 4 & -2 \\ 6 & 18 & -12 \end{pmatrix}.$$

On a

$$A^{(1)} = \begin{pmatrix} 6 & 18 & -12 \\ 0 & -2 & 2 \\ 0 & 8 & 16 \end{pmatrix} \begin{matrix} L_1 = L_3 \\ L_2 = L_2 - 1/3L_3 \\ L_3 = L_1 - 1/2L_3 \end{matrix} \quad L^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} \begin{matrix} \Delta_{2,1} = 1/3 \\ \Delta_{3,1} = 1/2 \end{matrix} \quad P^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} 6 & 18 & -12 \\ 0 & 8 & 16 \\ 0 & 0 & 6 \end{pmatrix} \begin{matrix} \\ L_2 = L_3 \\ L_3 = L_2 + 1/4 L_3 \end{matrix} \quad L^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/3 & -1/4 & 1 \end{pmatrix} \begin{matrix} \\ \\ \Delta_{3,1} = -1/4 \end{matrix} \quad P^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

On remarquera que l'on a permuté les 2èmes et 3èmes composantes de la première colonne de $A^{(1)}$ pour mettre à jour $A^{(2)}$. On a donc la décomposition $PA = LU$ avec $U = A^{(2)}$, $L = L^{(2)}$ et

$$P = P^{(2)} P^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

On vérifie que l'on a bien

$$PA = \begin{pmatrix} 6 & 18 & -12 \\ 3 & 17 & 10 \\ 2 & 4 & -2 \end{pmatrix} = LU = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/3 & -1/4 & 1 \end{pmatrix} \begin{pmatrix} 6 & 18 & -12 \\ 0 & 8 & 16 \\ 0 & 0 & 6 \end{pmatrix}.$$

Autre façon de calculer P : Plutôt que de noter toutes les projections intermédiaire, on peut partir de l'identité et mettre à jour la matrice de permutation au fur et à mesure de l'algorithme. Reprenons le cas de l'exemple 2, on a

$$A^{(1)} = \begin{pmatrix} 6 & 18 & -12 \\ 0 & -2 & 2 \\ 0 & 8 & 16 \end{pmatrix} \begin{matrix} L_1 = L_3 \\ L_2 = L_2 - 1/3 L_3 \\ L_3 = L_1 - 1/2 L_3 \end{matrix} \quad L^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} \begin{matrix} \\ \Delta_{2,1} = 1/3 \\ \Delta_{3,1} = 1/2 \end{matrix} \quad P^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} 6 & 18 & -12 \\ 0 & 8 & 16 \\ 0 & 0 & 6 \end{pmatrix} \begin{matrix} \\ L_2 = L_3 \\ L_3 = L_2 + 1/4 L_3 \end{matrix} \quad L^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/3 & -1/4 & 1 \end{pmatrix} \begin{matrix} \\ \\ \Delta_{3,1} = -1/4 \end{matrix} \quad P^{(2)} P^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

On retrouve bien

$$P = P^{(2)} P^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Chapitre 2

Méthodes itératives

Nous avons vu qu'un moyen efficace pour résoudre l'équation $Ax = b$ consiste à décomposer la matrice A sous la forme $PA = LU$ (ou $AP = LU$, ou $PAQ = LU$ selon les variantes), puis à résoudre successivement les systèmes triangulaires $Ly = b$ et $Ux = y$. La décomposition de A requiert un coût algorithmique de l'ordre de $O(n^3)$ opérations, tandis que la résolution des systèmes associés s'effectue en $O(n^2)$ opérations. Ainsi, la résolution complète de $Ax = b$ a une complexité de $O(n^3)$, ce qui est nettement plus efficace que les $O(n^2 n!)$ opérations nécessaires pour calculer naïvement l'inverse de A via son déterminant.

Cependant, pour des systèmes de très grande dimension, même la décomposition LU peut devenir prohibitive. Par exemple, un écran $8K$ contient environ 33×10^6 pixels, correspondant à un espace de dimension $n = 33 \times 10^6$. Dans ce cas, $n^3 \approx 3.5 \times 10^{22}$, ce qui représente un coût algorithmique extrêmement élevé, même pour un ordinateur moderne. Des méthodes alternatives, dites *itératives*, permettent de réduire ce coût en approximant la solution x au lieu de la calculer de manière exacte. Cette approximation est souvent acceptable en pratique si l'erreur finale reste négligeable par rapport aux erreurs intrinsèques liées à la représentation numérique des données (la précision de la virgule flottante par exemple).

Bien que nous n'étudierons pas en détail la complexité des méthodes itératives par rapport à la méthode LU, il est utile de noter que les méthodes itératives sont avantageuses dans les situations suivantes :

1. **Grande dimension du système** : lorsque n est suffisamment grand pour que le coût de $O(n^3)$ opérations devienne prohibitif.
2. **Matrices denses** : lorsque A ne contient beaucoup de zéros. Inversement, si A est creuse, i.e si A contient beaucoup de zéros (par exemple, une matrice par bande, où $a_{ij} = 0$ si $|i - j| > \ell$ pour un certain entier ℓ), les matrices L et U issues de la décomposition $PA = LU$ conservent cette propriété. Dans ce cas, la décomposition peut être réalisée en seulement $O(\ell n^2)$ opérations.
3. **Résolution ponctuelle** : lorsque l'on ne cherche à résoudre $Ax = b$ que pour un petit nombre de vecteurs b. Inversement, si de nombreux systèmes avec la même matrice A

doivent être résolus, il est plus efficace de calculer une fois la décomposition $PA = LU$ et de stocker les matrices P, L et U pour réutilisation.

2.1 Types de méthodes itératives et résultats généraux

Il existe principalement deux méthodes itératives, appelées respectivement de type I et II (cette dernière étant une variante de la méthode de type I). Dans cette section, nous nous intéressons à la résolution de l'équation $Ax = b$, où $A \in GL_n(\mathbb{K})$ et $b \in \mathbb{K}^n$.

Méthode itérative de type I

Le principe de cette méthode consiste à construire une suite $(x_k) \in \mathbb{K}^n$ définie itérativement par un point de départ arbitraire $x_0 \in \mathbb{K}^n$, et pour tout $k \geq 0$:

$$x_{k+1} = Bx_k + c, \tag{2.1}$$

avec $B \in M_n(\mathbb{K})$ et $c \in \mathbb{K}^n$ indépendants de $k \geq 0$.

Definition 2.1.1. *La méthode itérative est dite*

- *consistante si l'équation $x = Bx + c$ a une unique solution égale à $A^{-1}b$,*
- *convergente si pour tout $x_0 \in \mathbb{K}^n$, la suite $(x_k)_{k\geq 1}$ définie par (2.1) converge.*

Proposition 2.1.1. *La méthode est consistante si $c = (I_n - B)A^{-1}b$. Si elle est consistante et convergente, toute suite $(x_k)_{k\geq 0}$ définie par (2.1) converge vers $A^{-1}b$.*

En d'autres termes, pour une méthode consistante, le vecteur c est entièrement déterminé par A, B, et b.

Démonstration. Si la méthode est consistante, l'unique solution de $x = Bx + c$ est $A^{-1}b$. On en déduit donc que

$$A^{-1}b = BA^{-1}b + c,$$

ce qui implique que

$$c = (I_n - B)A^{-1}b.$$

Supposons maintenant que la méthode soit à la fois consistante et convergente. Pour tout $x_0 \in \mathbb{K}^n$, la suite $(x_k)_{k\geq 0}$ converge vers une limite $\ell(x_0)$. En prenant la limite dans (2.1), on a

$$\ell(x_0) = B\ell(x_0) + c.$$

La limite $\ell(x_0)$ est donc un point fixe de (2.1), et par consistance on a

$$\ell(x_0) = A^{-1}b.$$

□

Méthode itérative de type II

La méthode itérative de type I s'avère en réalité inapplicable. En effet, pour qu'elle soit intéressante, elle doit au moins être consistante, ce qui implique que c doit être choisi comme $c = (I_n - B)A^{-1}b$. Le principal problème est que cette expression nécessite de calculer l'inverse de la matrice A, ce que l'on ne veut surtout pas faire ! Une alternative, appelée méthode itérative de type II, est de définir l'itération par $x_0 \in \mathbb{K}^n$ puis de considérer

$$Mx_{k+1} = Nx_k + b, \tag{2.2}$$

où $M, N \in M_n(\mathbb{K})$ sont des matrices choisies de manière à ce que M soit facilement inversible. Cette condition permet de réécrire l'itération sous la forme suivante :

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b. \tag{2.3}$$

l apparaît ainsi que la méthode de type II est une version particulière de la méthode de type I, où la matrice d'itération est donnée par $M^{-1}N$. Les concepts de consistance et de convergence s'appliquent donc également à cette méthode de type II. L'avantage principal de la méthode de type II est qu'elle est explicite, c'est-à-dire qu'elle ne nécessite pas de calculer l'inverse de la matrice A pour l'implémenter.

Proposition 2.1.2. *La méthode itérative* (2.2) *est consistante si* $N = M - A$.

Démonstration. D'après l'égalité (2.3), la méthode itérative de type II (2.2) est une méthode itérative de type I avec $B = M^{-1}N$ et $c = M^{-1}b$. Pour qu'elle soit consistante, il faut donc que

$$M^{-1}b = (I_n - M^{-1}N)A^{-1}b,$$

c'est à dire, en multipliant par M (à gauche) dans l'égalité précédente,

$$b = (M - N)A^{-1}b.$$

Ceci implique que si $N = M - A$, alors $M - N = A$ et la consistance est vérifiée. On a donc proposé ici une méthode beaucoup plus réaliste. □

Méthodes de Jacobi et Gauss-Seidel

Dans cette section, nous présentons deux méthodes itératives de type II particulièrement importantes : les méthodes de Jacobi et de Gauss-Seidel. Les deux reposent sur la décomposition de la matrice A sous la forme $A = D - E - F$, où D est une matrice diagonale, E est une matrice triangulaire inférieure et F une matrice triangulaire supérieure. De manière explicite, on a

$$D = \left(a_{ii}\delta_{ij}\right)_{1\leq i,j\leq n}, \; E = \left(-a_{ij}\delta_{i>j}\right)_{1\leq i,j\leq n} \text{ et } F = \left(-a_{ij}\delta_{i<j}\right)_{1\leq i,j\leq n}.$$

Par exemple, si $A = \begin{pmatrix} 2 & -2 & 1 \\ 1 & 3 & -1 \\ -1/2 & 2 & -3 \end{pmatrix}$, nous avons

$$D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & -3 \end{pmatrix}, E = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 1/2 & -2 & 0 \end{pmatrix}, F = \begin{pmatrix} 0 & 2 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Attention ! Il est important de noter que les méthodes de Jacobi et de Gauss-Seidel ne sont définies que lorsque $a_{ii} \neq 0$ pour tous les i allant de 1 à n. Nous supposerons donc cette condition est satisfaite par la matrice A dans tout ce qui suit.

La méthode de Jacobi

La méthode de Jacobi est une méthode itérative de type II où l'on choisit $M = D$ et $N = E + F$. La matrice d'itération associée est donc :

$$M^{-1}N = D^{-1}(E + F).$$

Cette méthode est consistante grâce à la Propriété 2.1.2, puisque

$$M - N = D - (E + F) = A.$$

Un des avantages de la méthode de Jacobi réside dans la facilité de calcul de la matrice M^{-1}, qui peut être obtenue en $O(n)$ opérations, étant donné que M est une matrice diagonale. Cependant, sa convergence n'est souvent garantie que dans des cas très spécifiques, ce qui empêche de l'utiliser de manière généralisée.

La méthode de Gauss-Seidel

La méthode de Gauss-Seidel est la méthode de type II consistant à prendre $M = D - E$ et $N = F$. La matrice d'itération correspondante a alors deux expressions pratiques données par

$$M^{-1}N = (D - E)^{-1}F = I_n - (D - E)^{-1}A,$$

où on a utilisé le fait que $F = D - E - A$ dans la dernière égalité. Comme la méthode de Jacobi, la méthode de Gauss-Seidel est consistante en appliquant la Propriété 2.1.2 car

$$M - N = D - E - F = A.$$

Étant donné que M est une matrice triangulaire inférieure, résoudre l'équation $Mx_{k+1} = Nx_k + b$ est une opération explicite qui nécessite seulement $O(n^2)$ opérations. Ainsi, sous de bonnes conditions, la méthode de Gauss-Seidel peut s'avérer plus rapide que de faire la décomposition LU.

De plus, la convergence de la méthode de Gauss-Seidel est plus souvent garantie que celle de la méthode Jacobi, et cette convergence a lieu plus rapidement.

2.2 Convergence des méthodes itératives

Dans cette section, nous introduisons des critères de convergence et d'arrêt pour les méthodes itératives de type I. Étant donné que les méthodes de type II peuvent être considérées comme des cas particuliers des méthodes de type I, tous les résultats présentés ici s'appliquent également aux méthodes itératives de type II. Ces critères permettent non seulement de garantir la validité des solutions obtenues, mais aussi d'optimiser le temps de calcul en déterminant efficacement quand arrêter l'itération.

2.2.1 Convergence théorique d'une méthode de type I

Considérons la suite définie par une méthode itérative de type I :

$$\begin{cases} x_0 \in \mathbb{K}^n \\ x_{k+1} = Bx_k + c, \quad k \geq 0 \end{cases}$$

où $B \in M_n(\mathbb{K})$ et $c \in \mathbb{K}^n$. L'étude de la convergence de cette méthode repose sur une écriture explicite dex_k (et non une écriture récursive), qui est donnée par la proposition suivante.

Proposition 2.2.1. *Soit $(x_k)_{k\geq 1}$ une suite itérative définie par (2.1). Alors pour tout $k \geq 1$,*

$$x_k = B^k x_0 + \left(\sum_{i=0}^{k-1} B^i \right) c.$$

Démonstration. Cette propriété se démontre par récurrence sur $n \geq 0$. Le cas $k = 1$ est donné par la définition de $(x_k)_{k\geq 1}$. Supposons $k > 1$ et le résultat vrai pour $k - 1$. On a alors

$$x_{k-1} = B^{k-1} x_0 + \left(\sum_{i=0}^{k-2} B^i \right) c.$$

On en déduit par (2.1) que

$$\begin{aligned} x_k = Bx_{k-1} + c = B \left[B^{k-1} x_0 + \left(\sum_{i=0}^{k-2} B^i \right) c \right] + c &= B^k x_0 + B \left(\sum_{i=0}^{k-2} B^i \right) c + c \\ &= B^k x_0 + \left(\sum_{i=1}^{k-1} B^i \right) c + c \\ &= B^k x_0 + \left(\sum_{i=0}^{k-1} B^i \right) c. \end{aligned}$$

□

Grâce à la Proposition 2.2.1, nous remarquons qu'un rôle particulier est joué par les matrices $\sum_{i=0}^{k} B^i$ pour $k \geq 1$. La proposition suivante donne leur convergence (sous certaines hypothèses).

Proposition 2.2.2. *Soit $\|\cdot\|$ une norme multiplicative sur $M_n(\mathbb{K})$ et $B \in M_n(\mathbb{K})$ telle que $\|B\| < 1$. Alors $\sum_{k\geq 0} B^k$ est bien définie et*

1. $\|\sum_{k\geq 0} B^k\| \leq \frac{1}{1-\|B\|}$,
2. $\sum_{k\geq 0} B^k = (I_n - B)^{-1}$.

Démonstration. Supposons que $B \in M_n(\mathbb{K})$ soit telle que $\|B\| < 1$. Alors, comme $\|\cdot\|$ est multiplicative, pour tout $k \geq 1$ on a

$$\|B^k\| \leq \|B\|^k.$$

Utilisant le fait que $\|B\| < 1$, on en déduit que

$$\sum_{k\geq 0} \|B^k\| \leq \sum_{k\geq 0} \|B\|^k \leq \frac{1}{1-\|B\|} < +\infty.$$

On en déduit que la suite $\sum_{k\geq 0} B^k$ est absolument convergente et donc convergente dans l'espace complet $(M_n(\mathbb{K}), \|\cdot\|)$, et

$$\left\| \sum_{k\geq 0} B^k \right\| \leq \sum_{k\geq 0} \|B\|^k \leq \frac{1}{1-\|B\|}.$$

Pour montrer que $(I_n - B)\sum_{k\geq 0} B^k = I_n$, notons tout d'abord que pour $K \geq 1$,

$$(I_n - B)\sum_{k=0}^{K} B^k = \sum_{k=0}^{K} B^k - \sum_{k=1}^{K+1} B^k = I_n - B^{K+1}.$$

En passant à la limite à gauche et à droite de la dernière égalité quand K tend vers l'infini, et en utilisant que $B^K \xrightarrow[K\to\infty]{} 0$, on obtient

$$(I_n - B)\sum_{k\geq 0} B^k = I_n.$$

□

Par la Proposition 2.2.2, nous pouvons déduire une condition suffisante de convergence pour une méthode itérative de type I.

Corollaire 2.2.1. *Une méthode de type I définie par $B \in M_n(\mathbb{K}), c \in \mathbb{K}^n$ est convergente si il existe une norme opérateur $\|\cdot\|$ sur $M_n(\mathbb{K})$ qui vérifie $\|B\| < 1$.*

Démonstration. Si $\|B\| < 1$ pour une norme opérateur $\|\cdot\|$, par la Proposition 2.2.2, la série $\sum_{i\geq 0} B^i$ est convergente dans $M_n(\mathbb{K})$. La suite $\sum_{i=0}^{k-1} B^i c$ converge donc vers $\sum_{i\geq 0} B^i c$ quand k tend vers

l'infini. D'autre part, comme pour $k \geq 1$ on $\|B^k\| \leq \|B\|^k$, on a

$$\|B^k x_0\| \leq \|B\|^k \|x_0\| \xrightarrow{k\to\infty} 0.$$

A l'aide des Propositions 2.2.1 et 2.2.2, on a

$$x_k = B^k x_0 + \sum_{i=0}^{k-1} B^i c \xrightarrow[k\to\infty]{} \sum_{i=0}^{+\infty} B^i c = (I_n - B)^{-1} c.$$

□

Le problème du critère précédent pour garantir la convergence de la méthode est qu'il dépend du choix d'une norme opérateur particulière qui vérifie $\|B\| < 1$, ce qui n'est pas forcément évident à trouver suivant le contexte (il se peut très bien que $\|B\|_a < 1$ et $\|B\|_b > 1$ pour deux normes opérateurs $\|\cdot\|_a$ et $\|\cdot\|_b$). Nous allons donc énoncer une condition nécessaire et suffisante pour garantir la convergence de la méthode.

Definition 2.2.1. *Le rayon spectral $\rho(A)$ d'une matrice $A \in M_n(\mathbb{K})$ est définie comme*

$$\rho(A) = \max\{|\lambda|,\ \textit{valeur propre complexe de } A\}.$$

Le rayon spectral d'une matrice est fortement lié aux normes opérateurs de la matrice. La relation la plus forte, donnée par le point 3 de la propriété suivante, sera admise dans ce cours (cf le TD 3 dans le cas d'une matrice diagonalisable).

Proposition 2.2.3.

1. *Si $A = A^*$, nous avons en particulier $\rho(A) = \|A\|_2$.*
2. *Soit $\|\cdot\|$ une norme sur $\mathbb{K}^n$ et $\|\cdot\|$ sa norme associée sur $M_n(\mathbb{K})$. Alors*

$$\rho(A) \leq \|A\|.$$

3. *[Admis]*

$$\rho(A) = \inf(\|A\|,\ \|\cdot\| \textit{ norme opérateur sur } M_n(\mathbb{K})\}).$$

Démonstration. 1. Nous avons vu que

$$\|A\|_2 = \max(|\lambda|^{1/2}, \lambda \text{ valeur propre de } A^*A).$$

Si A est hermitienne, alors $A^* = A$, et donc

$$\begin{aligned}\max(|\lambda|^{1/2}, \lambda \text{ valeur propre de } A^*A) &= \max(|\lambda|^{1/2}, \lambda \text{ valeur propre de } A^2)\\ &= \max(|\lambda^2|^{1/2}, \lambda \text{ valeur propre de } A)\\ &= \max(|\lambda|, \lambda \text{ valeur propre de } A)\\ &= \rho(A).\end{aligned}$$

2. Soit $\lambda_0 \in \mathbb{K}$ tel que $|\lambda_0| = \rho(A)$ et $\bar{x}$ un vecteur propre associé pour A. Nous avons alors pour toute norme $\|\cdot\|$ sur $\mathbb{K}^n$,

$$\frac{\|A\bar{x}\|}{\|\bar{x}\|} = \frac{\|\lambda_0\bar{x}\|}{\|\bar{x}\|} = |\lambda_0| = \rho(A).$$

On en déduit que pour toute norme opérateur sur A associée à une norme $\|\cdot\|$,

$$\|A\| = \sup_{x\in\mathbb{K}^n, x\neq 0} \frac{\|Ax\|}{\|x\|} \geq \frac{\|A\bar{x}\|}{\|\bar{x}\|} = \rho(A).$$

□

En combinant la propriété précédente avec le Corollaire 2.2.1, nous obtenons un nouveau critère de convergence pour les méthodes itératives. Il se trouve que ce critère est nécessaire et suffisant, comme le montre la proposition suivante.

Proposition 2.2.4. *Une méthode consistante est convergente si et seulement si*

$$\rho(B) = \sup\{|\lambda|, \text{ valeur propre complexe de } B\} < 1.$$

Démonstration. Supposons que $\rho(B) < 1$. Par le point (3) de la Propriété 2.2.3, il existe donc une norme opérateur $\|\cdot\|$ telle que $\|B\| \leq \rho(B) + \frac{1-\rho(B)}{2} < 1$. Par le Corollaire 2.2.1, la méthode converge donc.
Supposons maintenant $\rho(B) \geq 1$ et soit x tel que $x = Bx + c$ (qui existe car la méthode est consistante). Comme $\rho(B) \geq 1$, il existe donc v tel que $Bv = \lambda v$ avec $|\lambda| \geq 1$. Posons alors $x_0 = v + x$. Alors la méthode itérative initiée à x_0 satisfait pour tout $k \geq 1$

$$x_k - x = B(x_{k-1} - x) = \cdots = B^k(x_0 - x) = \lambda^k v.$$

Comme $|\lambda| \geq 1$, le terme de droite ne tend pas vers 0 quand k tend vers l'infini. La suite $(x_k)_{k\geq 0}$ ne converge donc pas vers $\bar{x}$. On en déduit qu'elle ne peut pas converger par le second point de la Propriété 2.1.1. La méthode n'est donc pas convergente. □

En combinant la correspondance entre méthode de type I et méthode de type II avec le proposition précédente et le Corollaire 2.2.1, nous obtenons les critères de convergence suivants pour la méthode de type II.

Corollaire 2.2.2. *On a les conditions de convergence suivante pour la méthode (2.2) :*

Condition nécéssaire : *La méthode (2.2) est convergente si $\|M^{-1}N\| < 1$ pour une certaine norme opérateur $\|\cdot\|$.*

Condition nécéssaire et suffisante : *La méthode (2.2) est convergente si et seulement si*

$$\rho(M^{-1}N) < 1.$$

2.2.2 Critères d'arrêt

Nous supposons ici que nous implémentons une méthode de type I consistante et convergente. Comme cette méthode produit une suite (infinie) $(x_k)_{k\geq 0}$, il se pose la question pratique de savoir quand arrêter l'algorithme. Rappelons que l'objectif est de résoudre l'équation $Ax = b$ et que, par consistance et convergence de la méthode, la solution est théoriquement donnée par la limite $\bar{x}$ de la méthode itérative.
Un critère évident, mais irréaliste, serait donc d'arrêter l'algorithme quand $\|x_k - \bar{x}\| \leq \epsilon$, où ϵ est la marge d'erreur que l'on accepte. Le problème est qu'on ne connaît pas $\bar{x}$, ce qui empêche d'évaluer $\|x_k - \bar{x}\|$! Il existe alors deux critères d'arrêts alternatifs :

1. $\|Ax_k - b\| \leq \epsilon$: Ce critère garantit que x_k satisfait approximativement l'équation $Ax = b$. L'approximation est jugée acceptable car inférieure à une valeur subjective ϵ. Cependant, la précision de x_k par rapport à $\bar{x}$ dépend du conditionnement de A. Si A est mal conditionnée, x_k peut être assez éloigné de $\bar{x}$, même si ce critère est satisfait.
2. $\|x_k - x_{k-1}\| \leq \epsilon$:
   dans ce cas, nous arrêtons la méthode lorsque la variation du vecteur x_k à chaque itération devient négligeable. Attention ! Suivant les propriétés de la matrice d'itération B, x_k peut être très éloigné de $\bar{x}$ (et donc Ax_k peut être également éloigné de b).

Notons que le premier critère d'arrêt ne dépend pas du choix de la méthode (i.e du choix de B, c), mais plutôt de l'approximation Ax par Ax_k, et donc du conditionnement de A. En revanche, le deuxième critère d'arrêt dépend fortement du choix de B. Plus précisément, pour ces deux critères, nous pouvons en fait relier la distance de x_k à $\bar{x}$ après arrêt de l'algorithme.

Proposition 2.2.5. *Supposons la méthode consistante et soit $\bar{x}$ tel que $A\bar{x} = b$. Pour toute norme $\|\cdot\|$ sur $\mathbb{K}^n$, en notant également $\|\cdot\|$ pour la norme associée sur $M_n(\mathbb{K})$,*

1. *Si $\|Ax_k - b\| \leq \epsilon$, alors*

$$\|x_k - \bar{x}\| \leq \|A^{-1}\|\epsilon, \qquad et \qquad \frac{\|x_k - \bar{x}\|}{\|\bar{x}\|} \leq Cond(A)\frac{\epsilon}{\|b\|}.$$

2. *Si $\|x_k - x_{k-1}\| \leq \epsilon$, alors*

$$\|x_{k-1} - \bar{x}\| \leq \|(I_n - B)^{-1}\|\epsilon, \qquad et \qquad \frac{\|x_{k-1} - \bar{x}\|}{\|\bar{x}\|} \leq Cond(I_n - B)\frac{\epsilon}{\|c\|}.$$

Démonstration. 1. Supposons que $\|Ax_k - b\| \leq \epsilon$. Alors comme $A\bar{x} = b$,

$$\|A(x_k - \bar{x})\| = \|Ax_k - b\| \leq \epsilon.$$

On en déduit que

$$\|x_k - \bar{x}\| = \left\|A^{-1}\left[A(x_k - \bar{x})\right]\right\| \leq \|A^{-1}\| \cdot \|A(x_k - \bar{x})\| \leq \|A^{-1}\|\epsilon.$$

Comme $\|b\| = \|A\bar{x}\| \leq \|A\| \cdot \|\bar{x}\|$, on a

$$\frac{\|x_k - \bar{x}\|}{\|\bar{x}\|} \leq \frac{\|A\|}{\|b\|} \cdot \|A^{-1}\|\epsilon = Cond(A)\frac{\epsilon}{\|b\|}.$$

2. Supposons que $\|x_k - x_{k-1}\| \leq \epsilon$. Alors comme $\bar{x} = B\bar{x} + c$,

$$\|(I_n - B)(x_{k-1} - \bar{x})\| = \|x_{k-1} - x_k + c - c\| = \|x_{k-1} - x_k\| \leq \epsilon.$$

On en déduit donc que

$$\begin{aligned}\|x_{k-1} - \bar{x}\| = \|(I_n - B)^{-1} \cdot (I_n - B)(x_{k-1} - \bar{x})\| &\leq \|(I_n - B)^{-1}\| \cdot \|(I_n - B)(x_{k-1} - \bar{x})\| \\ &\leq \|(I_n - B)^{-1}\| \cdot \epsilon.\end{aligned}$$

De même, comme $\|c\| = \|\bar{x} - B\bar{x}\| \leq \|I_n - B\| \cdot \|\bar{x}\|$,

$$\frac{\|x_{k-1} - \bar{x}\|}{\|\bar{x}\|} \leq \|(I_n - B)^{-1}\| \cdot \epsilon \cdot \frac{\|I_n - B\|}{\|c\|} \leq Cond(I_n - B)\frac{\epsilon}{\|c\|}.$$

□

En d'autres termes, pour le premier critère d'arrêt, la majoration de l'erreur relative dépend du conditionnement de A, tandis que pour le deuxième critère, cette majoration dépend des choix de c et de B.

Remarque 2.2.1. *Concernant le deuxième critère, il faut faire attention à ce que de manière générale, une suite $(x_k)_{k\geq 0}$ peut satisfaire $\|x_{k+1} - x_k\| \xrightarrow{k\to\infty} 0$ sans que $(x_k)_{k\geq 0}$ ne converge. C'est par exemple le cas pour $x_k = x_{k-1} + \frac{1}{k}$. Dans le cas spécifique des méthodes itératives de ce cours, ce phénomène ne se passe pas pour nos méthodes convergentes car $\rho(B) < 1$ (cf le point 2. de la proposition précédente).*

2.3 Diagonalisation de matrice

Nous avons vu dans la section précédente qu'il était important de pouvoir déterminer la plus grande ou la plus petite valeur propre d'une matrice. Ces informations sont cruciales pour, par exemple, analyser la convergence d'une méthode itérative ou obtenir une borne sur le conditionnement d'une matrice.
Cependant, tout comme pour l'inversion d'une matrice, il n'est pas envisageable de recourir à la caractérisation des valeurs propres comme les racines du polynôme caractéristique de la matrice. En effet :

- Le calcul explicite de ce polynôme implique le calcul d'un déterminant, une opération coûteuse en termes de complexité algorithmique.
- Extraire les racines d'un polynôme est une opération numériquement instable en pratique, particulièrement pour des matrices de grande taille ou mal conditionnées.

Afin de pallier ces limitations, nous introduisons une méthode itérative appelée *méthode de la puissance*. Cette méthode permet d'approcher efficacement la plus grande ou la plus petite valeur propre d'une matrice, ainsi que le vecteur propre correspondant.

Remarque 2.3.1. *Rappelons que si A est une matrice, on dit que A a pour valeur propre $\lambda \in \mathbb{K}$ si il existe un vecteur non nul x tel que*

$$Ax = \lambda x.$$

Le vecteur x est alors appelé vecteur propre pour A associé à la valeur propre λ. Pour toute valeur propre λ et vecteur propre associé x, on a

$$\frac{\langle Ax, x\rangle}{\|x\|_2^2} = \lambda.$$

Une matrice A est diagonalisable si il existe une base B de $\mathbb{K}^n$ qui consiste en des vecteurs propres de A.

Algorithme de la puissance

L'algorithme de la puissance est défini ainsi :

1. on fixe $x_0 \in \mathbb{K}^n$ (attention! contrairement aux méthodes précédentes, ce choix ne doit pas être complètement quelconque).
2. pour $k \geq 0$, on pose
$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|_2}.$$
3. Après arrêt de l'algorithme, on pose
$$\theta_k = \langle Ax_k, x_k\rangle.$$

Pour $A \in M_n(\mathbb{K})$ diagonalisable, nous désignons $\{\lambda_i\}_{1\leq i\leq n}$ les valeurs propres de A ordonnées par module décroissant. Nous avons donc

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|.$$

Nous désignons également $\{v_i\}_{1\leq i\leq n}$ une base de $\mathbb{K}^n$ telle que v_i est un vecteur propre unitaire de A pour λ_i pour tout $1 \leq i \leq n$.

Proposition 2.3.1. *Supposons que A soit diagonalisable et que $|\lambda_1| > |\lambda_2|$. Si $x_0 \notin Vect(v_i, i \geq 2)$, alors*

- *θ_k converge vers λ_1.*
- *$\left(\left(\frac{\overline{\lambda_1}}{|\lambda_1|}\right)^k x_k\right)_{k\geq 1}$ converge vers un vecteur propre unitaire de A associé à λ_1.*

Démonstration. Soit $x_0 \in \mathbb{K}^n$, et écrivons

$$x_0 = \sum_{i=1}^{n} \mu_i v_i,$$

où $\{v_i\}_{1\leq i\leq n}$ est la base de vecteurs propres de A introduite précédemment et $\mu_i \in \mathbb{C}, 1 \leq i \leq n$. Remarquons tout d'abord que

$$x_k = \frac{Ax_{k-1}}{\|Ax_{k-1}\|_2} = \cdots = \frac{A^k x_0}{\|Ax_{k-1}\|_2 \cdot \|Ax_{k-2}\|_2 \ldots \|Ax_0\|_2}.$$

De plus, comme $\|x_k\|_2 = 1$, on a $\|A^k x_0\|_2 = \|Ax_{k-1}\|_2 \ldots \|Ax_0\|_2$, et donc

$$x_k = \frac{A^k x_0}{\|A^k x_0\|_2}.$$

Comme $x_0 \notin Vect(v_i, i \geq 2)$, $\mu_1 \neq 0$, et donc

$$\frac{1}{|\lambda_1|^k} A^k x_0 = \frac{1}{|\lambda_1|^k} \sum_{i=1}^{n} \lambda_i^k \mu_i v_i = \left(\frac{\lambda_1}{|\lambda_1|}\right)^k \mu_1 v_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{|\lambda_1|}\right)^k \mu_i v_i \tag{2.4}$$

Ainsi, comme $\|a\| - \|b\| \leq \|a+b\| \leq \|a+b\|$,

$$\underbrace{\left(\frac{|\lambda_1|}{|\lambda_1|}\right)^k}_{=1} |\mu_1| - \sum_{i=2}^{n} \underbrace{\left(\frac{|\lambda_i|}{|\lambda_1|}\right)^k}_{\to 0} |\mu_i| \leq \frac{1}{|\lambda_1|^k} \left\|A^k x_0\right\| \leq \underbrace{\left(\frac{|\lambda_1|}{|\lambda_1|}\right)^k}_{=1} |\mu_1| + \sum_{i=2}^{n} \underbrace{\left(\frac{|\lambda_i|}{|\lambda_1|}\right)^k}_{\to 0} |\mu_i|$$

Et donc, en passant à la limite,

$$\frac{1}{|\lambda_1|^k} \left\|A^k x_0\right\| \xrightarrow[k\to+\infty]{} |\mu_1|.$$

En reprenant l'égalité (2.4), on a

$$\begin{aligned} x_k = \frac{A^k x_0}{\|A^k x_0\|} = \frac{|\lambda_1^k|}{\|A^k x_0\|} \frac{A^k x_0}{|\lambda_1|^k} &= \frac{|\lambda_1^k|}{\|A^k x_0\|} \left(\left(\frac{\lambda_1}{|\lambda_1|}\right)^k \mu_1 v_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{|\lambda_1|}\right)^k \mu_i v_i \right) \\ &= \underbrace{\frac{|\lambda_1^k|}{\|A^k x_0\|}}_{\to 1/|\mu_1|} \left(\frac{\lambda_1}{|\lambda_1|}\right)^k \underbrace{\left(\mu_1 v_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1}\right)^k \mu_i v_i \right)}_{=w_k}. \end{aligned}$$

On a clairement w_k qui converge vers $\mu_1 v_1$. De plus, on a

$$\left(\frac{\overline{\lambda}_1}{|\overline{\lambda}_1|}\right)^k \left(\frac{\lambda_1}{|\lambda_1|}\right)^k = 1,$$

et il vient donc

$$\left(\frac{\overline{\lambda}_1}{|\lambda_1|}\right)^k x_{k+1} \xrightarrow[k\to+\infty]{} \frac{\mu_1}{|\mu_1|} v_1,$$

qui est bien un vecteur unitaire.

De plus, comme $\left(\frac{\overline{\lambda}_1}{|\lambda_1|}\right)^k \left(\frac{\lambda_1}{|\lambda_1|}\right)^k = 1$, et par continuité,

$$\theta_k = \langle Ax_k, x_k\rangle = \left\langle A\left(\frac{\overline{\lambda}_1}{|\lambda_1|}\right)^k x_k, \left(\frac{\overline{\lambda}_1}{|\lambda_1|}\right)^k x_k \right\rangle \xrightarrow[k\to+\infty]{} \frac{\mu_1\overline{\mu}_1}{|\mu_1|^2}\langle Av_1, v_1\rangle = \lambda_1\|v_1\|^2 = \lambda_1.$$

□

Méthode de la puissance inverse

Si A est diagonalisable et inversible avec valeurs propres $\{\lambda_1, \dots, \lambda_n\}$, alors A^{-1} est encore diagonalisable avec pour valeurs propres $\left\{\frac{1}{\lambda_n}, \dots, \frac{1}{\lambda_1}\right\}$. On en déduit que la plus petite valeur propre de A est l'inverse de la plus grande valeur propre de A^{-1}. En appliquant ce raisonnement à la méthode de la puissance, cela donne également une méthode pour trouver la plus petite valeurs propre de A. L'algorithme de la puissance inverse est définie ainsi :

1. on fixe $x_0 \in \mathbb{K}^n$ (attention ! ici aussi, il ne faut pas prendre x_0 n'importe comment).
2. pour $k \geq 0$, on résout (par exemple grâce au premier chapitre de ce cours)

   $$A\tilde{x}_{k+1} = x_k,$$

   puis on pose

   $$x_{k+1} = \frac{\tilde{x}_{k+1}}{\|\tilde{x}_{k+1}\|}.$$

3. Après arrêt de l'algorithme, on pose

   $$\bar{\theta}_k = \langle Ax_k, x_k\rangle.$$

   (Remarquons que x_k a déjà pour norme 1, donc il n'est pas nécessaire de diviser ici par $\|x_k\|$.)

Si $|\lambda_n| < |\lambda_{n-1}|$, par un raisonnement similaire à celui de la Proposition 2.3.1, $\left(\frac{\overline{\lambda}_n}{|\lambda_n|}\right)^{-k}(x_k)_{k\geq 1}$ converge vers un vecteur propre associé à la plus petite grande valeur propre de A^{-1} et donc à la plus petite valeur propre de A. Avec les même calculs que pour la Proposition 2.3.1, $\bar{\theta}_k$ converge vers cette plus petite valeur propre (pour un choix générique de x_0).