

Polytech

L3

Année 2023/2024

Premier semestre

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	14
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	30
2.2.1	Convergence théorique d'une méthode de type I	31
2.2.2	Critères d'arrêt	34
2.3	Diagonalisation de matrice	36

Chapitre 1

Systeme linéaire et décompositions LU

L'objectif de ce cours est de résoudre efficacement des systèmes d'équations linéaires. Rappelons qu'il s'agit de résoudre l'équation

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

où $A = (a_{ij})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}}$ est une matrice, b est un vecteur à coefficients complexes ou réels de dimension n et x est un vecteur de dimension m . Il y a deux aspects fondamentaux pour ce genre de problèmes, comme pour tout problème inverse :

1. l'aspect théorique de (1.1) :
 - (a) Le problème a-t-il une ou plusieurs solutions ?
 - (b) Existe-t-il une solution préférable ?
 - (c) Comment varie qualitativement cette ou ces solutions quand b varie ?
2. l'aspect pratique du problème :
 - (a) Comment trouve-t-on pratiquement cette solution ?
 - (b) Quelle est le coût algorithmique de la recherche de solution ?
 - (c) Le problème est-il stable d'un point de vue algorithmique ?

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 nous attacherons à répondre à ces six questions pour le problème (1.1) dans le cas d'un système carré inversible (cf Section 1.1 pour un rappel sur cette notion). Il est dans ce cas déjà connu que le système (1.1) a une unique solution 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 tout ce qui suit, on considère des matrices réelles ou complexes, mais la plupart des définitions et propriétés restent vraies pour des matrices à valeurs dans un corps \mathbb{K} . Dans ce qui suit, $\mathbb{K} = \mathbb{R}$ ou \mathbb{C} .

Matrice complexe : Une matrice A à n lignes et m colonnes est la donnée de $n \cdot m$ nombres complexes $a_{ij}, 1 \leq i, j \leq n$. On note $M_{n,m}(\mathbb{K})$ l'espace vectoriel des matrices à coefficients dans le corps \mathbb{C} . Si $n = m$, la matrice est dite carrée et on écrit simplement $M_n(\mathbb{K})$ (n est alors appelée la taille de la matrice). La matrice $0 \in M_{n,m}(\mathbb{C})$ est la matrice avec tous les coefficients nuls. Si $A \in M_{n,m}(\mathbb{K})$ et $1 \leq i \leq n, 1 \leq j \leq m$, on note $A[i, \cdot]$ la i -ème ligne de A et $A[\cdot, j]$ la j -ème colonne de A . Formellement, $A[i, \cdot] \in M_{1,m}(\mathbb{K})$ est une matrice ligne telle que la j -ème entrée vaut a_{ij} . De même, $A[\cdot, j] \in M_{n,1}(\mathbb{K})$ est une matrice colonne telle que la i -ème entrée vaut a_{ij} .

Transposée d'une matrice : On définit la transposée d'une matrice $A \in M_{n,m}(\mathbb{K})$ comme la matrice $A^T \in M_{m,n}(\mathbb{K})$ telle que

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

Produit de matrices : Si A et B sont deux matrices de taille $n \times m$ et $m \times l$, le produit $AB \in M_{n \times l}(\mathbb{K})$ est la matrice $(c_{ij})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq l}}$ donnée par

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

où a_{ij}, b_{kl} représentent les coefficients de A et B .

Base de $M_n(\mathbb{K})$: Pour $1 \leq k, l \leq n$, on note $E_{kl} = (\delta_{ik} \delta_{jl})_{1 \leq i, j \leq n}$ la matrice avec des zéros partout sauf dans l'entrée (i, j) . Les règles de multiplication énoncées ci-dessus impliquent que

$$E_{ij} E_{kl} = \delta_{jk} E_{il}.$$

Matrice identité : La matrice $I_n = \sum_{i=1}^n E_{ii} = (\delta_{ij})_{1 \leq i, j \leq n}$ 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 quand il existe $B \in M_n(\mathbb{K})$ 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

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

La matrice A est inversible si et seulement si son déterminant est non nul, et alors

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

où $(\text{Com}A)_{ij} = (-1)^{i+j} \det A^{[ij]}$ et $A^{[ij]}$ est la matrice A où les i -ème ligne et j -ème colonne ont été enlevées. On note $GL_n(\mathbb{K})$ l'ensemble des matrices inversibles à coefficients dans \mathbb{K} .

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

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

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

Dans toute la suite, nous nous intéresserons principalement aux matrices carrés. Il faut garder en tête que de nombreux résultats existent également dans le cas de 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

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

telle que

- $\|x\| = 0$ si et seulement si $x = 0$,
- $\|\lambda x\| = |\lambda| \|x\|$ pour tout $x \in E, \lambda \in \mathbb{K}$,
- $\|x + y\| \leq \|x\| + \|y\|$ 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 considérerons trois normes particulières :

- la norme euclidienne : $\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$,
- la norme 1 : $\|x\|_1 = \sum_{i=1}^n |x_i|$,
- la norme infini : $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$.

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

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

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

$$\|Ax\| = \|x\| \frac{\|Ax\|}{\|x\|} \leq \|x\| \sup_{x \in \mathbb{K}^n \setminus \{0\}} \frac{\|Ax\|}{\|x\|} \leq \|A\| \|x\|.$$

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

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

En effet, comme $A^T A$ est symétrique, elle est diagonalisable et les vecteurs propres forment une base orthonormée de \mathbb{K}^n . On notera D la matrice diagonale associée. Ainsi, on note $\lambda_1, \dots, \lambda_n$ les valeurs propres de $A^T A$, et pour tout $\|x\|_2 = 1$, on note $\tilde{x} = (\tilde{x}_1, \dots, \tilde{x}_n)$ l'écriture de x dans la nouvelle base orthonormée, il vient $\|\tilde{x}\|_2 = 1$ et

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

On a donc,

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

où λ_{\max} est la plus grande valeur propre de $A^T A$. De plus, si on considère v_{\max} un vecteur propre unitaire associée à la valeur propre λ_{\max} , on a

$$\|Av_{\max}\|^2 = v_{\max}^T A^T A v_{\max} = \lambda_{\max} \|v_{\max}\|^2 = \lambda_{\max}.$$

On a donc $\| \|A\| \|^2 \geq \lambda_{\max}$ et donc $\| \|A\| \|^2 = \lambda_{\max}$.

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

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

et

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

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 $(E_{kl})_{1 \leq k, l \leq n}$ 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

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

1.2.2 Conditionnement d'une matrice

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

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

Le conditionnement d'une matrice caractérise la stabilité du système linéaire (1.1) associé.

Proposition 1.2.2. Soit $A \in GL_n(\mathbb{K})$ et $b, b' \in \mathbb{K}^n$. Les solutions respectives x et x' de $Ax = b$ et $Ax' = b'$ satisfont l'erreur relative

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

Démonstration. Nous avons tout d'abord

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

D'autre part,

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

i.e $\frac{1}{\|x\|} \leq \frac{\| \|A\| \|}{\|b\|}$. En combinant ces deux inégalités, on obtient que

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

Le résultat est déduit par la définition de $\text{Cond}(A)$. □

En d'autres termes, une matrices avec un "bon" conditionnement, i.e une matrice avec un conditionnement faible permet que si on modifie légèrement b , alors la solution associée ne bouge pas trop par rapport à la solution d'origine. On a donc des méthodes numériques qui seront beaucoup plus stables. Inversement, si A a un grand conditionnement, on risque de rencontrer de nombreux problèmes de stabilité.

1.3 Méthode d'élimination de Gauss

Nous allons voir une première méthode numérique pour résoudre efficacement l'équation (1.1) quand A est inversible.

Remarque 1.3.1. *Il est illusoire de penser qu'on pourrait utiliser (1.3) pour calculer A^{-1} . En effet rien que le calcul du déterminant de A implique une somme sur l'ensemble des permutations possibles de n : il y a $n! = n \cdot (n-1) \dots 2 \cdot 1$ permutations de n . Cela implique qu'il faut par exemple environ $2,5 \cdot 10^{18}$ opérations pour inverser une matrice de taille 20. Avec un ordinateur ayant un processeur de 1 GHz, cela prendrait environ $2,5 \cdot 10^9 \cdot s$ soit 77 ans.*

La première méthode classique et utile de résolution de l'équation $Ax = b$ est la méthode de Gauss, qui est sans doute déjà connue. Pour rappel, celle-ci consiste à transformer le système

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 & = b_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}x_1 + \dots & + a_{nn}x_n = b_n \end{cases} \quad (1.4)$$

en un système triangulaire supérieur

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

avec $u_{ii} \neq 0$ pour $1 \leq i \leq n$, par des opérations sur les lignes.

1.3.1 Méthode sans pivot

L'algorithme se fait de manière suivante : posons $A^{(0)} = A, b^{(0)} = b$, puis à chaque étape $1 \leq \ell \leq n-1$, définissons

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

ce qui n'est bien sûr possible que si $a_{\ell\ell}^{(\ell-1)} \neq 0$ (nous verrons plus loin comment traiter le cas dégénéré $a_{\ell\ell}^{(\ell-1)} = 0$). En d'autres termes, on laisse inchangées les ℓ premières lignes de $A^{(\ell-1)}$ et $b^{(\ell-1)}$ et les $\ell-1$ premières colonnes de $A^{(\ell-1)}$. Remarquons qu'à l'étape ℓ , la matrice $A^{(\ell)} =$

$(a_{ij}^{(\ell)})_{1 \leq i, j \leq n}$ satisfait donc la condition

$$a_{ij}^{(\ell)} = 0, \quad j < i, j \leq \ell,$$

et donc finalement $A^{(n-1)}$ est triangulaire supérieure.

Autre façon de réécrire l'algorithme : Pour $A^{(l)}$, on peut réécrire l'algorithme de la façon suivante :

$$A^{(\ell)}[i,] = \begin{cases} A^{(\ell-1)}[i,] & \text{si } i \leq l \\ A^{(\ell-1)}[i,] - \frac{a_{i\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}} A^{(\ell-1)}[\ell,] & \text{sinon.} \end{cases} \quad b_i^{(\ell)} = \begin{cases} b_i^{(\ell-1)} & \text{si } i \leq \ell, \\ b_i^{(\ell-1)} - \frac{a_{i\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}} b_\ell^{(\ell-1)} & \text{si } i > \ell. \end{cases}$$

On retrouve ainsi le pivot de Gauss "classique". Dans tout ce qui suit on notera, on notera $\Delta_{i,\ell} = \frac{a_{i\ell}^{(\ell-1)}}{a_{\ell\ell}^{(\ell-1)}}$.

En posant $u_{ij} = A^{(n-1)}[i, j]$ pour $1 \leq i \leq j \leq n$ et $b'_i = b^{(n-1)}[i]$, nous avons transformé le système linéaire (1.4) en (1.5), et la solution y du premier coïncide avec celle du second. Le fait que A soit inversible assure que $u_{nn} \neq 0$. L'avantage est que le système (1.5) est simple à résoudre. Il suffit de poser successivement en partant de $i = n$

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

Exemple : On considère $Ax = b$ avec

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

On obtient donc

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

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

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 ne fonctionnait que si à chaque étape $1 \leq \ell \leq n-1$, nous avons $a_{i_\ell}^{(\ell-1)} \neq 0$. Dans le cas où cette condition n'est pas satisfaite pour $1 \leq \ell \leq n-1$, il suffit de permuter la ℓ -ième ligne de $A^{(\ell-1)}$ avec une ligne $i_\ell > \ell$ telle $a_{i_\ell}^{(\ell-1)} \neq 0$ puis d'appliquer l'opération (\mathcal{L}_ℓ) à la nouvelle matrice. Cela donne l'opération suivante :

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

où σ est la transposition échangeant ℓ et i_ℓ , c'est à dire

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

Le choix de l'entier i_ℓ apparaît à première vue arbitraire. Pour des raisons de stabilité de la méthode de Gauss, il est recommandé de choisir i_ℓ tel que $|a_{i_\ell\ell}^{(\ell-1)}|$ soit le plus grand possible. En effet, il faut ensuite diviser par $a_{i_\ell\ell}^{(\ell-1)}$ dans l'étape \mathcal{L}'_ℓ , et un $|a_{i_\ell\ell}^{(\ell-1)}|$ faible entraîne de plus grande valeurs dans la procédure. Un choix optimal de i_ℓ est donc

$$i_\ell = \operatorname{argmax}_{\ell \leq i \leq n} |a_{i\ell}^{(\ell-1)}|. \quad (1.6)$$

En fait, la méthode avec pivot est utile même quand la méthode sans pivot fonctionnait pour ces raisons de stabilité de l'algorithme.

Autre façon d'écrire l'algorithme : La aussi, on peut réécrire l'algorithme de la façon suivante

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

avec

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

Exemple : On considère $Ax = b$ avec

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

On a alors

$$A^{(1)} = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \begin{array}{l} \leftarrow L_1 \\ \leftarrow L_2 + L_1 \\ \leftarrow L_3 - 2L_1 \end{array} \quad b^{(1)} = \begin{pmatrix} 2 \\ 1 \\ -4 \end{pmatrix} \begin{array}{l} \leftarrow L_1 \\ \leftarrow L_2 + L_1 \\ \leftarrow L_3 - 2L_1 \end{array}$$

$$A^{(2)} = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{pmatrix} \begin{array}{l} \leftarrow L_1 \\ \leftarrow L_3 \\ \leftarrow L_2 \end{array} \quad b^{(2)} = \begin{pmatrix} 2 \\ -4 \\ 1 \end{pmatrix} \begin{array}{l} \leftarrow L_1 \\ \leftarrow L_3 \\ \leftarrow L_2 \end{array}$$

On résout alors

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

On vérifie bien que l'on a

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

1.3.3 Coût de la méthode de Gauss

Chaque étape \mathcal{L}'_ℓ nécessite $O((n - \ell)^2)$ opérations, et donc la transformations en un système triangulaire nécessite $O(n^3)$ opérations. L'opération finale (\mathcal{R}) nécessite elle $O(n^2)$ opération. La méthode de Gauss résout donc le système $Ay = b$ en $O(n^3)$ opérations. Cette vitesse est bien sûr plus grande dans le cas où la matrice A a de nombreuses entrées nulle : c'est par exemple le cas quand A est tridiagonale (ce qui n'est pas si rare!), auquel cas la résolution du système $Ay = b$ est en $O(n^2)$ opérations.

Une étude plus poussée de la complexité permettrait de montrer que la méthode de Gauss est asymptotiquement réalisée en $\frac{2}{3}n^3$ opérations.

1.4 Décomposition LU

Nous avons vu que la méthode de Gauss était une méthode relativement rapide pour résoudre (1.1). Il arrive souvent qu'on n'ait pas un seul système linéaire mais de nombreux systèmes linéaires à résoudre, tous de la forme $Ay^{(i)} = b^{(i)}$ pour une suite $(b^{(i)})_{1 \leq i \leq m}$ de données et une

suite $(y^{(i)})$ d'inconnues et avec la même matrice A . C'est typiquement le cas lorsque l'on cherche à inverser A , i.e dans ce cas on prend $b^{(i)} = e_i$ où e_i est le i -ème élément de la base canonique. Il est donc souhaitable d'automatiser la méthode de Gauss de manière à ne pas avoir à refaire à chaque fois la mise sous forme triangulaire du système. Cette automatisation est précisément le rôle de la décomposition LU de la matrice A .

1.4.1 Théorème de décomposition et conséquences

Nous énonçons dans un premier temps le théorème de décomposition LU et son application principale. La preuve de cette décomposition est 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 $a_{ij} = 0$ quand $i > j$ (resp. $i < j$). Remarquez que le produit de deux matrices triangulaires inférieures (resp. supérieures) est encore triangulaire inférieure (resp. supérieure), et la diagonale du produit est simplement le produit des diagonales. Une matrice triangulaire est donc inversible si et seulement si sa diagonale ne s'annule jamais. La matrice est dite unitriangulaire (inférieure ou supérieure) si elle est triangulaire (inférieure ou supérieure) et a tous ses coefficients diagonaux égaux à 1.

Une matrice $P = (p_{ij})_{1 \leq i, j \leq n}$ de taille n est une matrice de permutation si il existe une permutation σ de $\{1, \dots, n\}$ telle que $p_{ij} = \delta_{j\sigma(i)}$. On note alors P_σ la matrice de permutation associée à la permutation 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 allons le voir, cette décomposition est toujours 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*

$$PA = LU.$$

Plus que le résultat du théorème, c'est sa preuve, que nous verrons en section suivante, qui présente un intérêt pratique important : elle fournit en effet 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 autre version équivalente du théorème, moins répandue mais légèrement plus facile à démontrer, énonce qu'on peut toujours décomposer une matrice inversible A comme $AP = LU$, avec P matrice de permutation, L triangulaire inférieure avec des 1 sur la diagonale et U triangulaire su-

périeure. Dans ce dernier cas, la multiplication par P correspond à une permutation des colonnes de A et non des lignes.

Remarquez qu'il n'y a en général pas unicité. Cette non unicité se traduit essentiellement dans le choix de P . En effet, on peut prouver que pour chaque choix de P , il existe au plus un couple (L, U) de matrices unitriangulaire inférieure/triangulaire supérieure tel que $PA = LU$. D'un point de vue algorithmique, il y a un choix de P optimal qui assure 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 $x_1 = b_1$ puis $x_i = b_i - \sum_{k=1}^{i-1} l_{ik}x_k$. Cette solution est obtenue en $O(n^2)$ opérations.

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

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

avec $u_{ii} \neq 0$ pour $1 \leq i \leq n$ (car U est inversible). Ce système est également facilement résoluble : il suffit de poser $x_n = y_n/u_{nn}$ puis par récurrence $x_k = \frac{1}{u_{kk}}(y_k - \sum_{i=k+1}^n u_{ki}x_i)$. Cette solution est également obtenue en $O(n^2)$ opérations.

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

1.4.2 Preuve de la décomposition $A = LU$

On va d'abord s'intéresser à la preuve dans le cas "simple" ou $P = I_n$, i.e dans le cas où $A = LU$. Afin de faciliter les preuves, on va commencer par définir les vecteurs et la matrice de Gauss dans le cas sans pivot.

Definition 1.4.1 (Vecteurs et matrice de Gauss). *On rappelle que la ℓ -ième étape de la méthode de Gauss (sans pivot) peut s'écrire*

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

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

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

et "enregistre" donc toutes les transformations appliquées à A au temps ℓ . En notant $\{e_1, \dots, e_n\}$ la base

canonique de \mathbb{K}^n , la matrice de Gauss associée est définie par

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

La proposition suivante est cruciale pour reconstruire la matrice triangulaire inférieure L .

Proposition 1.4.1. 1. La matrice M_ℓ est inversible et

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

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 $(I_n - \tau_\ell e_\ell^T) (I_n + \tau_\ell e_\ell^T) = I_n$. On a donc

$$L_{n-1} \dots L_2 \underbrace{L_1 M_1}_{=I_n} M_2 \dots M_{n-1} = L_{n-1} \dots \underbrace{L_2 M_2}_{=I_n} \dots M_{n-1} = L_{n-1} M_{n-1} = I_n.$$

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

$$\begin{aligned}
L &= \left(I_n + \tau_1 e_1^T \right) \dots \underbrace{\left(I_n + \tau_{n-2} e_{n-2}^T \right) \left(I_n + \tau_{n-1} e_{n-1}^T \right)}_{= I_n + \tau_{n-2} e_{n-2}^T + \tau_{n-1} e_{n-1}^T + \tau_{n-2} \underbrace{e_{n-2}^T \tau_{n-1}}_{=0} e_{n-1}^T} \\
&= \left(I_n + \tau_1 e_1^T \right) \dots \underbrace{\left(I_n + \tau_{n-3} e_{n-3}^T \right) \left(I_n + \tau_{n-2} e_{n-2}^T + \tau_{n-1} e_{n-1}^T \right)}_{= I_n + \tau_{n-3} e_{n-3}^T + \tau_{n-2} e_{n-2}^T + \tau_{n-1} e_{n-1}^T + \tau_{n-3} \underbrace{e_{n-3}^T \tau_{n-2}}_0 e_{n-2}^T + \tau_{n-3} \underbrace{e_{n-3}^T \tau_{n-1}}_{=0} e_{n-1}^T} \\
&= \left(I_n + \tau_1 e_1^T \right) \dots \left(I_n + \tau_{n-4} e_{n-4}^T \right) \left(I_n + \tau_{n-3} e_{n-3}^T + \tau_{n-2} e_{n-2}^T + \tau_{n-1} e_{n-1}^T \right) \\
&\vdots \\
&= I_n + \tau_1 e_1^T + \dots + \tau_{n-1} e_{n-1}^T.
\end{aligned}$$

□

Afin de finir la preuve de la décomposition $A = LU$, il suffit maintenant de vérifier que l'on a $U = M_{n-1} \dots M_1 A$, 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 $(M_\ell[i,] A^{(\ell-1)}[1,], \dots, M_\ell[i,] A^{(\ell-1)}[n,])$. Comme les ℓ premières lignes de M_ℓ correspondent aux ℓ premières lignes de la matrices identité, il est évident que l'on a pour tout $i \leq \ell$

$$\left(M_\ell[i,] A^{(\ell-1)}[1,], \dots, M_\ell[i,] A^{(\ell-1)}[n,] \right) = \left(A^{(\ell-1)}[i,1], \dots, A^{(\ell-1)}[i,n] \right) = A^{(\ell-1)}[i,]$$

i.e M_ℓ laisse les ℓ premières lignes inchangées. Pour tout $i > \ell$, on a

$$\begin{aligned}
&\left(M_\ell[i,] A^{(\ell-1)}[1,], \dots, M_\ell[i,] A^{(\ell-1)}[n,] \right) \\
&= \left(-\Delta_{i,\ell} A^{(\ell-1)}[\ell,1] + A^{(\ell-1)}[i,1], \dots, -\Delta_{i,\ell} A^{(\ell-1)}[\ell,n] + A^{(\ell-1)}[i,n] \right) \\
&= \left(A^{(\ell-1)}[i,1], \dots, A^{(\ell-1)}[i,n] \right) - \Delta_{i,\ell} \left(A^{(\ell-1)}[\ell,1], \dots, A^{(\ell-1)}[\ell,n] \right) \\
&= A^{(\ell-1)}[i,] - \Delta_{i,\ell} A^{(\ell-1)}[\ell,].
\end{aligned}$$

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

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

On obtient donc

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

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

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 $PA = LU$, on va d'abord donner des exemples simples de l'utilité de cette décomposition par rapport à la décomposition $A = LU$. Le premier exemple simple est celui donné lors de l'introduction de la méthode de Gauss avec pivot : on peut rencontrer des cas où $A^{(\ell-1)}[\ell, \ell] = 0$ et comme on ne peut pas diviser par 0, la méthode sans pivot ne peut plus marcher. Une autre utilité de la décomposition $PA = LU$ est pour des questions de stabilité. Prenons l'exemple suivant¹ considérons

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

1. Cette exemple est tiré du cours de B. Gauzère, Gilles Gasso et Stéphane Canu.

Ces énormes différences d'échelles peuvent entraîner de nombreuses complications numériques. En effet, dans ce cas, on a $Cond(U) = cond(A) = 10^8$ et ainsi, pour résoudre ces problèmes, on peut considérer la méthode avec pivot, en prenant à chaque itération la plus haute valeur comme pivot. On aurait alors

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

et on règle ainsi les problèmes de différences d'échelle. En effet, on a $Cond(L) = 1.0001$ et $Cond(U) \simeq 2.61$.

On va maintenant définir plus précisément les matrices de permutation, qui seront particulièrement utiles pour la décomposition $PA = LU$.

Definition 1.4.2. Une matrice de permutation 2 à 2 est définie par

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

A titre d'exemple, on a

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

est la matrice identité où l'on a permuté les 2-ème et 4-ème lignes (ou colonnes). A noter que les matrices de permutation vérifient les propriétés suivantes :

- Pour toute matrice A , $P_{ij}A$ est la matrice A où l'on a permuté les i -ème et j -ème lignes.
- Pour toute matrice A , AP_{ij} est la matrice A où l'on a permuté les i -ème et j -ème colonnes.
- 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

$$U = M_{n-1}P^{(n-1)} \dots M_2P^{(2)}M_1P^{(1)}A.$$

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

$$\begin{aligned} U &= M_{n-1}P^{(n-1)}M_{n-2}P^{(n-2)} \dots M_1P^{(1)}A \\ &= M_{n-1} \underbrace{P^{(n-1)}M_{n-2}P^{(n-1)}}_{=:\hat{M}_{n-2}} P^{(n-1)}P^{(n-2)}M_{n-3}P^{(n-3)} \dots \tilde{M}_1P^{(1)}A \\ &= M_{n-1} \hat{M}_{n-2} \underbrace{P^{(n-1)}P^{(n-2)}M_{n-3}P^{(n-2)}P^{(n-1)}}_{=:\hat{M}_{n-3}} P^{(n-1)}P^{(n-2)}P^{(n-3)}M_{n-4} \dots M_1P^{(1)}A \\ &\vdots \\ &= \hat{M}PA \end{aligned}$$

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*

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

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

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

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

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

et on obtient donc le résultat pour \hat{M}_ℓ . 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 décompo-

sition de Gauss avec pivot. La matrice $P = P^{(n-1)} \dots P^{(1)}$ est le produit de toutes 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

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

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.

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

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

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

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

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

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

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

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

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

Chapitre 2

Méthodes itératives

Nous avons vu qu'un moyen efficace de résoudre l'équation $Ax = b$ était de décomposer la matrice A comme $PA = LU$ (ou $AP = LU$, ou $PAQ = LU$ suivant les autres versions possibles), puis de résoudre successivement $Ly = b$, $Ux = y$. La décomposition de A se fait en $O(n^3)$ opérations, puis la résolution de l'équation se fait ensuite en $O(n^2)$ opérations. On peut donc résoudre le système $Ax = b$ en $O(n^3)$ opérations. Ceci est bien sûr beaucoup mieux que les $O(n^2n!)$ opérations nécessaires au calcul naïf de l'inverse de A à l'aide du déterminant.

Toutefois, quand le système est très grand, même la décomposition LU devient trop coûteuse. Par exemple, un écran 8K contient 33.10^6 pixels ce qui donne des représentations d'images dans un espace de dimension $n = 33.10^6$. Dans ce cas $n^3 \simeq 3,5.10^{22}$, ce qui reste un coût algorithmique important pour un ordinateur actuel. Des méthodes alternatives dites *itératives* existent pour diminuer ce coût en renonçant à calculer de manière exacte la solution x . D'un point de vue pratique, ce dernier point n'est pas handicapant si l'erreur finale reste négligeable par rapport à l'erreur intrinsèque de représentation des données (donnée par exemple par la précision de la virgule flottante dans un ordinateur).

Nous ne démontrerons aucun résultat précis sur la comparaison de complexité entre la méthode LU et les méthodes itératives, mais il faut retenir que les méthodes itératives sont utiles quand :

1. La dimension n du système est très grande : il faut que n soit assez grand pour qu'un nombre $O(n^3)$ d'opérations soit trop coûteux,
2. Quand la matrice A n'a beaucoup de zéros. Inversement, si la matrice A est une matrice par bande (par exemple), ce qui signifie que $a_{ij} = 0$ si $|i - j| > l$ pour un certain entier l , alors les matrices L et U de la décomposition LU satisfont aussi cette propriété. La décomposition $PA = LU$ nécessite alors seulement $O(ln^2)$ opérations.
3. On est seulement intéressé par la résolution de l'équation $Ax = b$ pour peu de vecteurs b . Inversement, si on veut résoudre cette équation de manière systématique pour de nombreux vecteurs, il est plus efficace de faire la décomposition $PA = LU$ de la matrice A et de stocker les matrices P, L, U pour résoudre l'ensemble des systèmes linéaires.

2.1 Types de méthodes itératives et résultats généraux

Il existe principalement deux méthodes itératives, dites de type *I* et *II* (qui est en fait une variante de la méthode de type *I*). Dans toute cette section, l'objectif est de résoudre $Ax = b$ avec $A \in GL_n(\mathbb{K})$ et $b \in \mathbb{K}^n$.

Méthode itérative de Type *I*

Le principe est de déterminer une suite $(x_k) \in \mathbb{K}^n$ définie itérativement par $x_0 \in \mathbb{K}^n$ puis pour tout $k \geq 0$,

$$x_{k+1} = Bx_k + c, \quad (2.1)$$

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 $A^{-1}b$.*

Autrement dit, pour une méthode consistante, c est uniquement déterminée 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

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

soit

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

Supposons que la méthode soit convergente et consistante. Alors pour tout x_0 , la suite $(x_k)_{k \geq 0}$ converge vers une limite $\ell(x_0)$. Par passage à 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* est en réalité irréaliste. En effet, pour qu'elle soit intéressante il faut au moins qu'elle soit consistante, et pour cela il faudrait choisir $c = (I_n - B)A^{-1}b$. Le problème est

que l'on ne veut pas inverser A ! Une solution alternative, dite méthode itérative de type II , est de définir l'itération par $x_0 \in \mathbb{K}^n$ puis

$$Mx_{k+1} = Nx_k + b, \quad (2.2)$$

où $M, N \in M_n(\mathbb{K})$ sont choisies telles que M soit facile à inverser : pour calculer x_{k+1} , il faut en effet alors poser

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

On voit ainsi qu'une méthode de type II est une méthode de type I particulière : la matrice $M^{-1}N$ est alors appelée la matrice d'itération. En particulier, les notions de consistance et convergence s'étendent au type II . L'avantage est que la méthode est maintenant explicite, i.e on n'a pas à calculer l'inverse de A pour l'implémenter.

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

Démonstration. Par (2.3), la méthode itérative (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

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

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. \square

Méthodes de Jacobi et Gauss-Seidel

Nous allons voir ici deux exemples importants de méthodes itératives de type II , les méthodes de Jacobi et Gauss-Seidel. Les deux méthodes reposent sur la décomposition $A = D - E - F$, où D est une matrice diagonale, E une matrice triangulaire inférieure et F est une matrice triangulaire supérieure. De manière formelle, nous avons

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

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 ! Les méthodes de Jacobi et Gauss-Seidel décrites ci-dessous ne sont définies que lorsque $a_{ii} \neq 0$ pour tout $1 \leq i \leq n$. Nous supposons donc que la matrice A satisfait cette propriété dans le reste de la sous-section.

La méthode de Jacobi

La méthode de Jacobi est la méthode de type II consistant à prendre $M = D$ et $N = E + F$. La matrice d'itération correspondante est donc

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

Cette méthode est bien consistante par la propriété 2.1.2, puisque

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

L'avantage de la méthode de Jacobi est que la matrice M^{-1} est très rapide à calculer, puisqu'elle ne nécessite que $O(n)$ opérations (inversion d'une matrice diagonale). En revanche, la convergence de cette méthode n'est souvent garantie que dans certains 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

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

Comme M est triangulaire inférieure, nous avons vu dans le chapitre précédente que résoudre l'équation $Mx_{k+1} = Nx_k + b$ est explicite et ne nécessite que $O(n^2)$ opérations. Sous de bonnes conditions, la méthode de Gauss-Seidel est donc plus rapide que 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

Nous allons voir dans cette section des critères de convergence et d'arrêt pour les méthodes de type I. Comme les méthodes de type II sont des méthodes de type I particulières, tous les résultats

de la section s'appliquent également aux méthodes itératives de type II.

2.2.1 Convergence théorique d'une méthode de type I

Nous nous intéressons donc ici à la convergence de la suite définie par

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

où $B \in M_n(\mathbb{K})$ et $c \in \mathbb{K}^n$. Afin de donner des résultats de convergence pour les méthodes de type I, on va d'abord donner une écriture directe de x_k (et non une écriture récursive). Cette écriture 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 $B^K \xrightarrow{K \rightarrow \infty} 0$, 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

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

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

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

□

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

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

Le rayon spectral d'une matrice est fortement lié aux normes opérateurs de la matrice. La relation la plus forte, donnée par le 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]

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

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

Condition nécessaire 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 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. $\|Ax_k - b\| \leq \epsilon$: dans ce cas, nous arrêtons la méthode quand le vecteur x_k satisfait l'équation ' $Ax = b$ ' à une approximation près que nous jugeons acceptable. Attention ! Suivant le conditionnement de A , x_k peut alors être assez éloigné de \bar{x} .
2. $\|x_k - x_{k-1}\| \leq \epsilon$: dans ce cas, nous arrêtons la méthode quand la variation du vecteur x_k à chaque itération devient négligeable devant l'erreur acceptable. Attention ! Suivant les propriétés de la matrice d'itération B , x_k peut être très éloigné de \bar{x} et Ax_k aussi 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 seulement de l'équation $Ax = b$. En revanche, le deuxième critère d'arrêt dépend fortement du choix de B , comme nous allons le voir ci-dessous. 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

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

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

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

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

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

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

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

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

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

□

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 $\|x_{k+1} - x_k\| \xrightarrow{k \rightarrow \infty} 0$ 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 connaître la plus grande valeur propre ou plus petite valeur propre d'une matrice : pour pouvoir par exemple connaître la convergence d'une méthode itérative ou pour avoir une borne sur son conditionnement. Comme pour l'inversion d'une matrice, il n'est pas envisageable d'utiliser la caractérisation des valeurs propres comme zéros du polynôme caractéristique de la matrice : le calcul de ce polynôme fait appel au calcul d'un déterminant, calcul que nous voulons éviter pour des raisons de coût algorithmique. De plus, extraire les racines d'un polynôme est une opération qui peut s'avérer extrêmement instable en pratique.

Nous allons donc voir une méthode itérative, appelée *méthode de la puissance*, qui permet de caractériser la plus grande/plus petite valeur propre d'une matrice, ainsi que le vecteur propre associé.

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

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

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 $x_0 \notin \text{Vect}(v_i, i \geq 2)$, alors*

- θ_k converge vers λ_1 .
- $\left(\left(\frac{\lambda_1}{|\lambda_1|} \right)^k x_k \right)_{k \geq 1}$ 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

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

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

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

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

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

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

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

Et donc, en passant à la limite,

$$\frac{1}{|\lambda_1|^k} \|A^k x_0\| \xrightarrow{k \rightarrow +\infty} |\mu_1|.$$

En reprenant l'égalité (2.4), on a

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

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

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

et il vient donc

$$\left(\frac{\bar{\lambda}_1}{|\lambda_1|}\right)^k x_{k+1} \xrightarrow{k \rightarrow +\infty} \frac{\mu_1}{|\mu_1|} v_1,$$

qui est bien un vecteur unitaire.

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

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

□

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, $\left(\frac{\bar{\lambda}_n}{|\lambda_n|}\right)^{-k} (x_k)_{k \geq 1}$ 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 mêmes 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).