Chapitre 1

Les systèmes linéaires

La résolution d'un système linéaire algébrique est au cœur de la plupart des calculs en analyse numérique. Il paraît donc naturel de débuter un cours de calcul scientifique par là. Ici, nous décrivons les algorithmes de résolution les plus populaires qui sont appliqués à des systèmes généraux. Nous considérons le problème suivant : trouver le vecteur x solution de

$$A\,x \;=\; b,$$

où A est une matrice carrée et b un vecteur donné à coefficients réels ou complexes. La discrétisation d'équations différentielles ordinaires ou d'équations aux dérivées partielles, la modélisation de problèmes en physique, chimie ou économie conduit souvent à la résolution de systèmes linéaires de grande taille avec plusieurs milliers d'inconnues et il devient pratiquement impossible de résoudre ces systèmes d'équations sans l'aide d'un ordinateur. Il s'agit alors de trouver des algorithmes de résolution efficaces où le nombre d'opérations et donc le temps de calcul, n'est pas prohibitif. C'est un problème classique mais difficile en analyse numérique.

L'objectif de ce chapitre est de proposer différentes méthodes numériques de résolution des systèmes linéaires et de sensibiliser le lecteur à l'importance du choix de la méthode en fonction des propriétés du système. Nous distinguerons deux types de méthodes : *les méthodes directes* où nous calculons exactement la solution et *les méthodes itératives* où nous calculons une solution approchée.

1.1 Quelques exemples de systèmes linéaires

1.1.1 Exemple d'une analyse de l'offre et de la demande

Nous étudions d'abord un modèle simplifié en économie. Imaginons que plusieurs artisans décident de coopérer pour fabriquer différents produits utiles à chacun d'eux. Afin d'éviter le coût du stockage des produits fabriqués, nous recherchons une situation d'équilibre entre l'offre et la demande. Pour cela, considérons n artisans, sachant que chacun fabrique un produit spécifique. Ces artisans ont choisi de coopérer, c'est-à-dire qu'ils souhaitent adapter leur production, d'une part, à leurs **propres besoins** déterminés par la quantité de produit nécessaire aux autres artisans pour qu'ils puissent fabriquer leur propre produit et, d'autre part, aux **besoins du marché**.

Puisque chaque artisan fabrique un produit différent, nous notons par $i \in \{1, \ldots, n\}$ le produit fabriqué par un artisan, x_i désigne le nombre total de produits i fabriqués par un artisan et b_i la demande du marché en ce produit. D'autre part $(c_{i,j})_{1\leq i,j\leq n}$ exprime la quantité de produit i nécessaire

à la confection d'une unité de produit j. En supposant que la relation qui lie les différents produits est linéaire, nous recherchons l'équilibre entre les besoins et la production, c'est-à-dire en demandant que la quantité de produit i fabriqué soit égale à la somme des besoins des autres artisans en produit i pour fabriquer leur propre produit et des besoins du marché en i

$$x_i \;=\; \sum_{j=1}^{n} C_{i,j}\, x_j \;+\; b_i, \quad i = 1, \ldots, n$$

ou encore

$$x \;=\; C\, x \;+\; b,$$

où la matrice C est formée par les coefficients $(c_{i,j})_{1 \leq i,j \leq n}$ tandis que le vecteur b correspond à la demande du marché $b := (b_1, \ldots, b_n)^T$. Par conséquent, la production totale $x = (x_1, \ldots, x_n)^T$ est la solution du système linéaire $A\, x \;=\; b$, où la matrice A est donnée par $A \;=\; I_n - C$ et I_n est la matrice identité composée de 1 sur sa diagonale et de 0 ailleurs.

Cette modélisation pose plusieurs difficultés mathématiques. Tout d'abord nous nous interrogeons sur les propriétés de la matrice A permettant d'assurer que ce problème a bien une solution. Il vient ensuite la question du calcul de cette solution. Dans la pratique lorsque le nombre d'artisans considérés devient grand, il n'est plus possible de calculer l'inverse de la matrice A, il faudra donc fournir des algorithmes qui ne nécessitent pas l'inversion de cette matrice. C'est l'objectif de ce premier chapitre.

1.1.2 L'équation de la chaleur

Nous examinons maintenant un autre problème issu de la physique consistant à décrire l'évolution de la chaleur dans une pièce fermée. Imaginons une chambre dans laquelle une source de chaleur est appliquée aux bords et au centre. L'ouvert $\Omega \subset \mathbb{R}^2$ désigne le domaine de calcul, $\Gamma := \partial\Omega$ est le bord de Ω et $u(t, x, y)$ la température de la pièce au temps $t \geq 0$ et au point de l'espace $(x, y) \in \Omega$. En choisissant les constantes physiques égales à 1, la chaleur va se répandre à l'intérieur de la pièce en suivant une dynamique de diffusion décrite par une équation aux dérivées partielles, appelée équation de la chaleur linéaire.

Pour $t \geq 0$ et $(x, y) \in \Omega$, nous considérons

$$\begin{cases} \dfrac{\partial u}{\partial t} - \Delta u = f, \\ \\ u(t=0) = u_0, \text{ dans } \Omega, \\ \\ u(x,y) = g, \text{ sur } \Gamma, \end{cases}$$

où f désigne la puissance surfacique, g est la condition aux bords, u_0 une donnée initiale et le Laplacien Δ est donné par

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

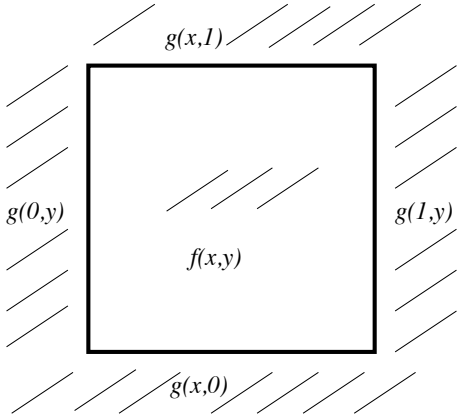

FIG. 1.1 – Étude de l'équation de la chaleur dans une pièce en dimension deux.

Lorsque la température aux bords est maintenue constante, la distribution de chaleur dans la pièce converge vers un état stationnaire. Dans ce cas, la dérivée temporelle dans l'équation précédente disparaît et nous obtenons l'équation de Poisson[1] :

$$-\Delta u = f, \quad (x, y) \in \Omega.$$

La résolution de cette équation représente un problème aux bords : la solution dépend fortement de la condition imposée aux bords du domaine $\Gamma = \partial\Omega$,

$$u = g, \quad (x, y) \in \Gamma,$$

où g désigne la source de chaleur aux bords (par exemple un radiateur attaché au mur).

Prenons alors $\Omega :=]0, 1[\times]0, 1[$ et recouvrons le domaine Ω par une grille. Soit P un point de la grille, dont les voisins sont notés E (Est), O, (Ouest), N (Nord) et S (Sud) représentés sur la Figure 1.1.2. Une formule de Taylor appliquée en chaque point voisin de P et pour une fonction u

[1] En référence au mathématicien français Denis Poisson (1781-1840). Son œuvre est considérable et touche aussi bien à la physique qu'aux mathématiques (série de Fourier, théorie des probabilités).

régulière, donne

$$\left\{\begin{array}{rcl} u(E) & = & u(P) + h\,\frac{\partial u}{\partial x}(P) + \frac{h^2}{2}\,\frac{\partial^2 u}{\partial x^2}(P) + \varepsilon(h^3), \\ u(O) & = & u(P) - h\,\frac{\partial u}{\partial x}(P) + \frac{h^2}{2}\,\frac{\partial^2 u}{\partial x^2}(P) + \varepsilon(h^3), \\ u(N) & = & u(P) + h\,\frac{\partial u}{\partial y}(P) + \frac{h^2}{2}\,\frac{\partial^2 u}{\partial y^2}(P) + \varepsilon(h^3), \\ u(S) & = & u(P) - h\,\frac{\partial u}{\partial y}(P) + \frac{h^2}{2}\,\frac{\partial^2 u}{\partial y^2}(P) + \varepsilon(h^3), \end{array}\right. \tag{1.1}$$

où $\varepsilon(h^p)$ signifie qu'il existe une constante $C > 0$ telle que $\|\varepsilon(h^p)\| \leq Ch^p$.

Les points de la grille sont équidistants $(x_i, y_j) = (i\,h, j\,h)$, avec $h = 1/(n+1)$. Ainsi, en écrivant $P = (x_i, y_j)$, $E = (x_{i+1}, y_j)$, $N = (x_i, y_{j+1})$, $O = (x_{i-1}, y_j)$ et $S = (x_i, y_{j-1})$. Puis, en sommant les quatre égalités de (1.1), il en découle une approximation de l'opérateur de Laplace de u au point P :

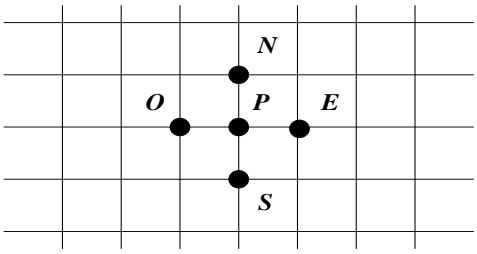

FIG. 1.2 – Localisation des points voisins d'un point arbitraire P où nous effectuons un développement de Taylor.

$$\begin{array}{rcl} h^2\Delta u(x_i, y_j) & = & -4u(x_i, y_j) + u(x_{i-1}, y_j) + u(x_{i+1}, y_j) + u(x_i, y_{j-1}) + u(x_i, y_{j+1}) \\ & & + 4\,\varepsilon(h^3). \end{array}$$

En notant $v_{i,j}$ l'inconnue approchant la solution u de l'équation de Poisson aux points (x_i, y_j), nous obtenons le système suivant de n^2 équations en négligeant les termes d'ordre supérieur à trois, c'est-à-dire les termes $\varepsilon(h^3)$,

$$\frac{4\,v_{i,j} - v_{i-1,j} - v_{i+1,j} - v_{i,j-1} - v_{i,j+1}}{h^2} = f(x_i, y_j) =: f_{i,j}$$

avec également n^2 inconnues en interprétant la condition aux limites $u = g$ sur Γ selon

$$v_{0,j} = g(0, y_j), \quad v_{n+1,j} = g(1, y_j), \quad j = 0, \ldots, n+1$$

et

$$v_{i,0} = g(x_i, 0), \quad v_{i,n+1} = g(x_i, 1), \quad i = 0, \ldots, n+1.$$

Nous verrons au Chapitre 7 comment justifier rigoureusement que nous obtenons bien une approximation de la solution du problème de l'équation de Poisson.

Ainsi, nous aboutissons à la résolution d'un système linéaire de grande taille : $A\,v = b$, avec

$$A = \begin{pmatrix} D & -I_n & 0 & \dots & 0 \\ -I_n & D & -I_n & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -I_n \\ 0 & \dots & 0 & -I_n & D \end{pmatrix}$$

et D est une matrice carrée de taille $n \times n$

$$D = \begin{pmatrix} 4 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 4 \end{pmatrix}$$

et v est le vecteur des inconnues et b la donnée

$$v = \begin{pmatrix} v_{1,1} \\ v_{2,1} \\ \vdots \\ v_{n,1} \\ v_{1,2} \\ \vdots \\ v_{n,n} \end{pmatrix}, \quad b = \begin{pmatrix} h^2\, f_{1,1} + g(0,h) + g(h,0) \\ h^2\, f_{2,1} + g(2\,h,0) \\ \vdots \\ h^2\, f_{1,2} + g(0,2\,h) \\ h^2\, f_{2,2} \\ \vdots \\ h^2\, f_{n,n} + g(1,1-h) + g(1-h,1) \end{pmatrix}.$$

Lorsque n devient grand, nous souhaitons développer des algorithmes numériques calculant la solution v, de manière exacte ou approchée, avec un nombre d'opérations (additions, divisions et multiplications) le plus faible possible. Observons également que cette matrice contient beaucoup de zéros (nous parlons de système creux) et nous verrons sur un exemple simple (Exercice 1.4.8) comment tirer profit de cette structure pour mettre en place des algorithmes rapides.

1.2 Rappels sur les matrices

Avant de s'intéresser à la résolution numérique de systèmes linéaires, présentons quelques rappels d'algèbre linéaire.

Soit $\mathcal{M}_{m,n}(\mathbb{K})$ l'ensemble des matrices à m lignes, n colonnes et à coefficient dans le corps des réels $\mathbb{K} = \mathbb{R}$ ou des complexes $\mathbb{K} = \mathbb{C}$. Pour un vecteur colonne $u \in \mathbb{K}^n$, la valeur u_i avec $1 \leq i \leq n$, désigne la i-ème composante dans la base canonique de $\mathbb{K}^n$ du vecteur u. Nous appelons *adjoint* du vecteur colonne u, le vecteur ligne $u^\star$ de $\mathbb{K}^n$ tel que $u^\star = (\overline{u}_1, \ldots, \overline{u}_n)$ où $\overline{u}_i$ désigne le conjugué de u_i et transposé de u est le vecteur ligne $u^T = (u_1, \ldots, u_n)$.

Rappelons également le produit matrice-vecteur. Soient A une matrice à m lignes et n colonnes et u un vecteur de $\mathbb{K}^n$, nous définissons $v = A\,u \in \mathbb{K}^m$ le vecteur dont les composantes sont données par

$$v_i = \sum_{j=1}^{n} a_{i,j}\, u_j, \quad i = 1, \ldots, m$$

et pour B une matrice à n lignes et p colonnes, le produit $A\,B$ fournit une matrice $C \in \mathcal{M}_{m,p}$ donnée par

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k}\, b_{k,j}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, p.$$

Introduisons également la notion de matrice adjointe et transposée.

Définition 1.2.1 *Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$, la matrice adjointe de A, notée $A^\star$, à n lignes et m colonnes, est donnée par $A^\star = (a^\star_{i,j})_{\substack{1\leq i\leq n\\ 1\leq j\leq m}}$ et $a^\star_{i,j} = \overline{a}_{j,i}$, pour $1 \leq i \leq n$ et $1 \leq j \leq m$.*

La matrice transposée de A, notée A^T, à n lignes et m colonnes, est donnée par $A^T = (a^T_{i,j})_{\substack{1\leq i\leq n\\ 1\leq j\leq m}}$ et $a^T_{i,j} = a_{j,i}$, pour $1 \leq i \leq n$ et $1 \leq j \leq m$.

Dans la base canonique de $\mathbb{K}^n$, définissons le produit scalaire ou produit hermitien entre deux vecteurs u et $v \in \mathbb{K}^n$, le scalaire de $\mathbb{K}$ donné par

$$\langle u, v \rangle = u^\star\, v = \sum_{i=1}^{n} u_i^\star\, v_i.$$

1.2.1 Cas des matrices carrées

Considérons maintenant le cas particulier des matrices carrées, pour lesquelles le nombre de lignes est égal au nombre de colonnes. Nous rappelons les définitions suivantes :

Définition 1.2.2 *Une matrice carrée $A \in \mathcal{M}_{n,n}(\mathbb{K})$ est dite inversible, ou régulière, ou encore non singulière, s'il existe une matrice $B \in \mathcal{M}_{n,n}(\mathbb{K})$ telle que $A\,B = B\,A = I_n$. Dans ce cas, la matrice B est unique et s'appelle la matrice inverse de A, elle est notée A^{-1}.*

Définition 1.2.3 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$, alors*

- *A est une matrice normale si $A^\star\, A = A\, A^\star$.*
- *A est une matrice unitaire si $A^\star\, A = A\, A^\star = I_n$, où I_n désigne la matrice identité. Dans le cas où $\mathbb{K} = \mathbb{R}$, nous parlons de matrice orthogonale et $A^T\, A = A\, A^T = I_n$, c'est-à-dire $A^T = A^{-1}$.*

– *A est une matrice hermitienne si $A^\star = A$. Dans le cas où $\mathbb{K} = \mathbb{R}$, nous parlons de matrice symétrique et $A^T = A$.*

Il est alors facile de faire le lien entre les matrices hermitiennes et normales.

Proposition 1.2.1 *Toute matrice hermitienne est une matrice normale.*

Comme nous l'avons vu en début de chapitre, l'objectif de cette partie est de mettre au point des algorithmes de résolution numérique pour un système de la forme

$$A\,x = b, \tag{1.2}$$

où $A \in \mathcal{M}_{n,n}(\mathbb{K})$ et $b \in \mathbb{K}^n$ sont donnés et $x \in \mathbb{K}^n$ est l'inconnue. Nous donnons d'abord une condition nécessaire et suffisante sur la matrice A pour que ce système admette une solution unique. Le système linéaire (1.2) admet une solution unique dès que la matrice A est inversible et cette solution est donnée par $x = A^{-1}\,b$. Il est alors important de connaître quelques propriétés des matrices inversibles [16].

Proposition 1.2.2 *Soient A, $B \in \mathcal{M}_{n,n}(\mathbb{K})$ inversibles, nous avons alors*
– *pour tout $\alpha \in \mathbb{K}$, $(\alpha\,A)^{-1} = \dfrac{1}{\alpha}\,A^{-1}$.*
– $(A\,B)^{-1} = B^{-1}\,A^{-1}$.
– $(A^\star)^{-1} = (A^{-1})^\star$.

Énonçons quelques propriétés des matrices inversibles, qui sont autant d'outils permettant de s'assurer que le problème (1.2) admet bien une solution [16].

Théorème 1.2.1 (Théorème des matrices inversibles) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$, les propositions suivantes sont équivalentes :*
– *A est inversible,*
– *le déterminant de A est non nul,*
– *le rang de A est égal à n, c'est-à-dire que les n vecteurs colonnes forment une famille libre,*
– *le système homogène $A\,x = 0$ a pour unique solution $x = 0$,*
– *pour tout b dans $\mathbb{K}^n$, le système linéaire $A\,x = b$ a exactement une solution.*

Pour conclure cette partie, rappelons qu'une valeur propre de A est donnée par $\lambda \in \mathbb{K}$ telle que $\det(A - \lambda\,I_n) = 0$. Ainsi, il existe au moins un vecteur v non nul, dit *vecteur propre*, vérifiant $A\,v = \lambda\,v$. Définissons alors le spectre d'une matrice.

Définition 1.2.4 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$. Nous appelons spectre de A l'ensemble des valeurs propres de A :*

$$\mathrm{Sp}(A) = \{\lambda_i \in \mathbb{K}, 1 \leq i \leq n, \exists\, v_i \in \mathbb{K}^n : v_i \neq 0 \quad A\,v_i = \lambda_i\,v_i\}\,.$$

Nous appelons rayon spectral de A le nombre réel positif $\rho(A)$ tel que

$$\rho(A) = \max_{1\leq i\leq n} \{|\lambda_i|, \quad \lambda_i \in \mathrm{Sp}(A)\}\,.$$

À partir de la définition d'une valeur propre, nous vérifions facilement qu'une matrice est inversible si et seulement si 0 n'est pas valeur propre.

La conséquence de ces propriétés est que l'ensemble des matrices carrées inversibles forme un groupe, appelé le groupe linéaire et noté habituellement $Gl_n(\mathbb{K})$. En général, ¡¡ presque toutes ¿¿ les matrices sont inversibles. Sur le corps $\mathbb{K}$, cela peut être formulé de façon plus précise : l'ensemble des matrices non inversibles, considéré comme sous-ensemble de $\mathcal{M}_{n,n}(\mathbb{K})$, est un ensemble négligeable, c'est-à-dire de mesure de Lebesgue nulle. Intuitivement, cela signifie que si vous choisissez au hasard une matrice carrée à coefficients réels, la probabilité pour qu'elle soit non inversible est égale à zéro. La raison est que des matrices non inversibles peuvent être considérées comme racines d'une fonction polynôme donnée par le déterminant [16].

Par la suite, nous nous intéresserons au calcul proprement dit de la solution de (1.2). Proposons d'abord un premier algorithme naïf qui permet de calculer la solution d'un système linéaire ; cet algorithme est dû à Cramer[2] [23].

Théorème 1.2.2 (Méthode de Cramer) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice inversible. Alors la solution du système $Ax = b$ est donnée par $x_i = \det(A_i)/\det(A)$, pour tout $i = 1, \ldots, n$, où A_i est la matrice A pour laquelle la i-ème colonne est remplacée par le vecteur b.*

Cette méthode bien que très élégante est très coûteuse puisqu'elle nécessite plus de $n!$ opérations où $n!$ est la factorielle de n et est donnée par $n! = n \times (n-1) \times \ldots \times 2 \times 1$. Elle n'est donc jamais utilisée dans la pratique sauf en dimension $n = 2$. Cet algorithme revient pratiquement à calculer explicitement la matrice A^{-1}. Hélas, ce calcul est souvent long et fastidieux, même pour un ordinateur, c'est pourquoi nous avons recours à des algorithmes de résolution exacte d'une complexité moindre (nous parlons alors d'une méthode directe), ou des méthodes itératives qui consistent à construire une suite de solutions approchées qui converge vers la solution exacte. Avant de présenter de tels algorithmes, nous introduirons des matrices dont la structure particulière est bien adaptée à la résolution du système linéaire (1.2).

1.2.2 Quelques matrices particulières

Cette partie constitue la base théorique permettant de mettre au point des méthodes pour la résolution exacte de (1.2).

Matrices triangulaires Nous introduisons la notion de matrice triangulaire et montrons que pour les matrices de ce type, la résolution du système linéaire (1.2) devient très facile.

Définition 1.2.5 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$,*

- *la matrice A est une matrice triangulaire inférieure si $A = (a_{i,j})_{1 \leq i,j \leq n}$ avec $a_{i,j} = 0, \quad 1 \leq i < j \leq n$.*
- *A est une matrice triangulaire supérieure lorsque $a_{i,j} = 0, 1 \leq j < i \leq n$.*
- *A est une matrice diagonale dès lors que $a_{i,j} = 0$ pour $i \neq j$.*

Vérifions d'abord que l'ensemble des matrices triangulaires supérieures (resp. inférieures) est stable par la somme, le produit et l'inversion.

[2] En référence au mathématicien suisse Gabriel Cramer (1704-1752).

Proposition 1.2.3 *Soit $L \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice triangulaire inférieure inversible, ce qui signifie que tous les éléments diagonaux sont non nuls. Alors L^{-1} est aussi une matrice triangulaire inférieure.*
Soit U une matrice triangulaire supérieure inversible. Alors U^{-1} est aussi une matrice triangulaire supérieure.

Les matrices triangulaires jouent un rôle important en analyse numérique car elles sont facilement inversibles ou du moins, nous pouvons facilement trouver la solution $x \in \mathbb{K}^n$ du système linéaire $A\,x = b$. En effet, considérons le cas d'une matrice triangulaire supérieure, alors la solution x se calcule par un algorithme dit de remontée. Nous observons d'abord qu'une matrice triangulaire inversible a tous ses éléments diagonaux non nuls, c'est pourquoi $x_n = b_n/a_{n,n}$ puis pour tout $i = n-1, n-2, \ldots, 1$, nous pouvons calculer x_i de la manière suivante :

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

Exemple 1.2.1 *Soit $A \in \mathcal{M}_{3,3}(\mathbb{R})$ donnée par*

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix},$$

nous cherchons une solution de $A\,x = b$ avec $b = (3,4,3)^T$. En commençant par la dernière ligne, nous trouvons $x_3 = 1$. Puis en injectant cette valeur dans l'avant-dernière équation, cela donne $x_2 = 1$. Finalement, connaissant x_2 et x_3, la première équation donne directement $x_1 = 1$.

Remarque 1.2.1 *Nous allons voir par la suite qu'une méthode directe calcule la solution exacte de (1.2) et consiste le plus souvent à trouver un système triangulaire équivalent.*

Nous énonçons ensuite le Théorème de Shur [7] qui sera utile pour rendre certaines matrices triangulaires par simple changement de base.

Théorème 1.2.3 (Théorème de Shur) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice quelconque. Alors il existe une matrice unitaire $U \in \mathcal{M}_{n,n}(\mathbb{K})$, c'est-à-dire $U^\star U = I_n$ et une matrice $T \in \mathcal{M}_{n,n}(\mathbb{K})$ triangulaire dont la diagonale est composée par l'ensemble des valeurs propres de A telles que $T = U^\star A U$.*

Matrices hermitiennes Dans le cas des matrices hermitiennes, ce dernier résultat peut être sensiblement amélioré.

Corollaire 1.2.1 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice hermitienne. Alors il existe une matrice unitaire $U \in \mathcal{M}_{n,n}(\mathbb{K})$ et une matrice diagonale $D \in \mathcal{M}_{n,n}(\mathbb{K})$ dont la diagonale est composée par l'ensemble des valeurs propres de A, telles que $D = U^\star A U$.*

Nous rappelons enfin quelques propriétés de base des matrices hermitiennes, définies positives. Nous introduisons d'abord la notion de sous-matrice principale.

Définition 1.2.6 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice quelconque. Nous appelons sous-matrice principale d'ordre i (pour $1 \leq i \leq n$) de A, la matrice $A_i \in \mathcal{M}_{i,i}(\mathbb{K})$ obtenue en ne gardant que les i premières lignes et les i premières colonnes de A.*

Définition 1.2.7 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$, nous disons que A est positive si pour tout $x \in \mathbb{K}^n$, elle vérifie $x^T\, A\, x \geq 0$ et A est définie positive si pour tout $x \in \mathbb{K}^n \setminus \{0\}$, elle vérifie $x^T\, A\, x > 0$ et $x^T\, A\, x = 0$ implique $x = 0$.*

Nous avons alors les résultats suivants dont la preuve est laissée en exercice.

Proposition 1.2.4 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice hermitienne définie positive. Alors elle vérifie*
- *toute sous-matrice principale A_i, $1 \leq i \leq n$ est hermitienne définie positive ;*
- *tous les coefficients diagonaux de A sont des réels strictement positifs ;*
- *A est diagonalisable et ses valeurs propres sont strictement positives ;*
- *le déterminant de A est strictement positif, c'est-à-dire que A est inversible ;*
- *il existe une constante $\alpha > 0$, telle que $x^\star\, A\, x \geq \alpha \|x\|^2$, pour n'importe qu'elle norme vectorielle $\|.\|$.*

Matrices de permutations Par la suite, nous utiliserons souvent les matrices de permutations pour réordonner les lignes ou les colonnes d'une matrice quelconque.

Définition 1.2.8 *Une matrice de permutations est une matrice carrée qui ne possède que des 0 et des 1 comme coefficients, telle qu'il y ait un seul 1 par ligne et par colonne.*

Une matrice de permutations vérifie alors

Proposition 1.2.5 *Soient σ et τ deux permutations des indices $\{1, \ldots, n\}$, nous avons alors*
- *la matrice de permutations P_σ correspondant à la permutation σ s'écrit comme*

$$(P_\sigma)_{i,j} = \delta_{i,\sigma(j)} = \begin{cases} 1, & \text{si } i = \sigma(j), \\ 0, & \text{sinon.} \end{cases}$$

- $P_\sigma\, P_\tau = P_{\sigma \circ \tau}$.
- $(P_\sigma)^T\, P_\sigma = I_n$, *c'est une matrice orthogonale.*

1.2.3 Conditionnement de matrices

Avant de décrire des algorithmes de résolution de systèmes linéaires (1.2), nous mettons en évidence une difficulté supplémentaire : la sensibilité de la solution par rapport aux perturbations des données $b \in \mathbb{K}^n$ ou des coefficients de la matrice A. En effet, considérons par exemple la matrice de Hilbert donnée par

$$a_{i,j} = \frac{1}{i+j-1}, \quad i,j = 1, \ldots, n.$$

Nous avons pour $n = 4$,

$$\begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{pmatrix}.$$

Ainsi, pour un vecteur donné $b \in \mathbb{R}^4$ tel que

$$b^T = \left(\frac{25}{12}, \frac{77}{60}, \frac{57}{60}, \frac{319}{420}\right) \simeq (2,08\,,\ 1,28\,,\ 0,95\,,\ 0,76)\,,$$

nous calculons alors $x = A^{-1}\,b$ et vérifions facilement que $x^T = (1,1,1,1)$. Ensuite, si nous modifions légèrement la source $\tilde{b}^T = (2{,}1\,,1{,}3\,,1\,,0{,}8)$, la solution $\tilde{x}$ du système $\tilde{x} = A^{-1}\tilde{b}$ est donnée par $\tilde{x}^T = (5{,}6\,,\text{-}48\,,114\,,\text{-}70)$, ce qui est très loin de la solution de départ x.

En définitive, nous constatons qu'une petite perturbation de la donnée b conduit à une grande modification de la solution x, ce qui engendre des instabilités lors de la résolution du système.

Norme matricielle Pour mesurer l'ampleur de cette instabilité, nous introduisons la notion de *norme matricielle*. En effet, l'ensemble $\mathcal{M}_{n,n}(\mathbb{K})$ peut être considéré comme étant un $\mathbb{K}$-espace vectoriel muni d'une norme $\|.\|$.

Définition 1.2.9 *Nous appelons norme matricielle toute application $\|.\|$ de $\mathcal{M}_{n,n}(\mathbb{K})$ à valeur dans $\mathbb{R}^+ := [0, +\infty[$ qui vérifie les propriétés suivantes :*

- *pour toute matrice $A \in \mathcal{M}_{n,n}(\mathbb{K})$, $\|A\| = 0 \Rightarrow A = 0_{\mathbb{K}^{n\times n}}$,*
- *pour toute matrice $A \in \mathcal{M}_{n,n}(\mathbb{K})$, et pour tout $\alpha \in \mathbb{K}$, $\|\alpha A\| = |\alpha|\,\|A\|$,*
- *pour toutes matrices A et $B \in \mathcal{M}_{n,n}(\mathbb{K})$, $\|A + B\| \leq \|A\| + \|B\|$, c'est l'inégalité triangulaire,*
- *pour toutes matrices A et $B \in \mathcal{M}_{n,n}(\mathbb{K})$, $\|A\,B\| \leq \|A\|\,.\,\|B\|$.*

Notons bien que définir une norme matricielle requiert une condition supplémentaire par rapport à la définition d'une norme vectorielle (la dernière propriété de la définition). Il n'est donc pas évident *a priori* de pouvoir construire une telle application seulement à partir d'une norme vectorielle. À titre d'exemple la norme de Froebenus $\|.\|_F$ donnée par

$$\|A\|_F = \left(\sum_{i,j=1}^{n} |a_{i,j}|^2\right)^{1/2} \tag{1.3}$$

est bien une norme matricielle. Les trois premières propriétés sont connues puisque $\|.\|_F$ est une norme vectorielle de $\mathbb{K}^{n^2}$. Il reste à démontrer que pour toutes matrices A et $B \in \mathcal{M}_{n,n}(\mathbb{K})$, $\|A\,B\|_F \leq \|A\|_F\,\|B\|_F$. En effet,

$$\|A\,B\|_F^2 = \|C\|_F^2 = \sum_{i,j=1}^{n} c_{i,j}^2 = \sum_{i,j=1}^{n} \left(\sum_{k=1}^{n} a_{i,k}\,b_{k,j}\right)^2.$$

En utilisant l'inégalité de Cauchy-Schwarz, nous avons

$$\|A\,B\|_F^2 \leq \sum_{i,j=1}^{n} \left(\sum_{k=1}^{n} a_{i,k}^2\right) \left(\sum_{k=1}^{n} b_{k,j}^2\right) = \left(\sum_{i,k=1}^{n} a_{i,k}^2\right) \left(\sum_{k,j=1}^{n} b_{k,j}^2\right), = \|A\|_F^2\,\|B\|_F^2.$$

Pour construire une norme matricielle à partir d'une norme vectorielle quelconque, nous introduisons la définition suivante :

Définition 1.2.10 *Soit $\|.\|$ une norme vectorielle sur $\mathbb{K}^n$. Nous définissons la norme matricielle $\|.\|$ subordonnée à la norme vectorielle $\|.\|$ comme étant l'application donnée par*

$$A \in \mathcal{M}_{n,n}(\mathbb{K}) \mapsto \|A\| := \sup_{\substack{v\in\mathbb{K}^n \\ v\neq 0}} \frac{\|A\,v\|}{\|v\|}.$$

Nous vérifions facilement que cette application définit bien une norme matricielle.

Par exemple pour $1 \leq p \leq \infty$, nous savons que l'application qui à $v \in \mathbb{K}^n$ fait correspondre le réel positif

$$\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p\right)^{1/p}$$

ou $\|v\|_\infty = \max_{1\leq i\leq n} |v_i|$ pour $p = \infty$, est une norme vectorielle sur $\mathbb{K}^n$. Posons alors pour $p \in [1, +\infty]$ et $A \in \mathcal{M}_{n,n}(\mathbb{K})$

$$\|A\|_p = \sup_{\substack{v\in\mathbb{K}^n \\ v\neq 0}} \frac{\|A\,v\|_p}{\|v\|_p},$$

qui est une norme matricielle subordonnée à la norme vectorielle $\|.\|_p$. En général nous ne pouvons pas calculer $\|A\|_p$ directement en fonction de $(a_{i,j})_{1\leq i,j\leq n}$ sauf pour $p = 1$ et $p = \infty$.

Proposition 1.2.6 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice quelconque. Alors nous avons*

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|$$

et

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}|.$$

Démonstration : Soit $v \in \mathbb{K}^n$. D'une part, nous avons

$$\|A\,v\|_1 = \sum_{i=1}^{n} \left|\sum_{j=1}^{n} a_{i,j}\,v_j\right| \leq \sum_{i=1}^{n}\sum_{j=1}^{n} |a_{i,j}|\,|v_j|,$$
$$\leq \sum_{j=1}^{n} \left(\sum_{i=1}^{n} |a_{i,j}|\right) |v_j| \leq \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}| \|v\|_1.$$

D'où

$$\|A\|_1 \leq \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|.$$

D'autre part, nous montrons qu'il existe $w \in \mathbb{K}^n$ unitaire $\|w\|_1$ tel que

$$\|A\,w\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|.$$

En effet, choisissons $j_0 \in \{1,\ldots,n\}$ tel que

$$\sum_{i=1}^{n} |a_{i,j_0}| = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|,$$

puis prenons $w \in \mathbb{K}^n$ tel que $w_i = 0$ pour $i \neq j_0$ et $w_{j_0} = 1$. Alors

$$\|A\,w\|_1 = \sum_{i=1}^{n} |(A\,w)_i| = \sum_{i=1}^{n} |a_{i,j_0}| = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j_0}|.$$

Par un raisonnement analogue, nous montrons le résultat pour la norme matricielle $\|.\|_\infty$. □

Dans le cas particulier de la norme subordonnée à la norme euclidienne de $\mathbb{K}^n$ définie pour $A \in \mathcal{M}_{n,n}(\mathbb{K})$ par

$$\|A\|_2 = \sup_{\substack{v\in\mathbb{K}^n \\ v\neq 0}} \frac{\|A\,v\|_2}{\|v\|_2},$$

nous avons le résultat suivant.

Proposition 1.2.7 *Pour $A \in \mathcal{M}_{n,n}(\mathbb{K})$, nous avons $\|A\|_2 = \sqrt{\rho(A^\star\,A)} = \sqrt{\rho(A\,A^\star)}$.*

Démonstration : Prenons $\mathbb{K} = \mathbb{C}$ ou $\mathbb{R}$, nous avons par définition de la norme subordonnée à la norme euclidienne

$$\begin{aligned} \|A\|_2^2 &= \sup_{\substack{v\in\mathbb{K}^n \\ v\neq 0}} \frac{\|A\,v\|_2^2}{\|v\|_2^2} = \sup_{\substack{v\in\mathbb{K}^n \\ v\neq 0}} \frac{(A\,v)^\star\,(A\,v)}{v^\star\,v}, \\ &= \sup_{\substack{v\in\mathbb{K}^n \\ v\neq 0}} \frac{v^\star\,A^\star\,A\,v}{v^\star\,v}. \end{aligned}$$

Or, nous vérifions que

$$(A^\star\,A)^\star = A^\star\,(A^\star)^\star = A^\star\,A.$$

Ainsi, d'après le Corollaire 1.2.1 puisque la matrice $A^\star\,A$ est hermitienne, elle est donc diagonalisable et il existe $U \in \mathcal{M}_{n,n}(\mathbb{K})$ unitaire telle que $U\,A^\star\,A\,U = \mathrm{diag}(\mu_k)$, où $\mathrm{diag}(\mu_k)$ représente une matrice diagonale formée à partir des valeurs μ_k sur la diagonale. Les scalaires $(\mu_k)_{1\leq k\leq n}$ sont les valeurs propres de la matrice hermitienne $A^\star\,A$. Notons que $\mu_k \geq 0$ puisque de la relation $A^\star\,Ap_k = \mu_k\,p_k$, nous déduisons que $(A\,p_k)^\star\,Ap_k = \mu_k\,p_k^\star\,p_k$, c'est-à-dire $\mu_k = \|A\,p_k\|_2^2/\|p_k\|_2^2 \geq 0$. Il vient ensuite

$$\sup_{\substack{v\in\mathbb{K}^n \\ v\neq 0}} \frac{v^\star\,A^\star\,A\,v}{v^\star\,v} = \sup_{\substack{v\in\mathbb{K}^n \\ v\neq 0}} \frac{v^\star\,U^\star\,U\,A^\star\,A\,U^\star\,U\,v}{v^\star\,U^\star\,U\,v}.$$

Puis, comme U est inversible, nous avons par changement de variable $w = U\,v$

$$\|A\|_2^2 \;=\; \sup_{\substack{w\in\mathbb{K}^n\\ w\neq 0}} \frac{w^\star\,U\,A^\star\,A\,U^\star\,w}{w^\star\,w} \;=\; \sup_{\substack{w\in\mathbb{K}^n\\ w\neq 0}} \frac{\sum_{k=1}^n \mu_k |w_k|^2}{\sum_{k=1}^n |w_k|^2},$$

où les μ_k sont les valeurs propres de $A^\star\,A$. Enfin, en prenant le vecteur de la base canonique dont toutes les composantes sont nulles exceptée la k-ème qui correspond à la plus grande valeur propre μ_k en module, nous obtenons $\|A\|_2^2 = \rho(A\,A^\star)$. □

Conditionnement d'une matrice Essayons maintenant de comprendre de manière plus générale le phénomène d'instabilité par rapport à la donnée $b \in \mathbb{K}^n$. Soient $A \in \mathcal{M}_{n,n}(\mathbb{K})$ inversible et b un vecteur de $\mathbb{K}^n$, non nul. Nous désignons par x la solution du système $A\,x = b$ et pour une perturbation δb du vecteur b, le vecteur $x + \delta x$ est la solution de $A\,(x + \delta x) = b + \delta b$.

Pour une norme vectorielle $\|.\|$ de $\mathbb{K}^n$, nous cherchons à contrôler l'erreur relative $\|\delta x\| \,/\, \|x\|$ en fonction de l'erreur relative $\|\delta b\| \,/\, \|b\|$ et de la norme matricielle subordonnée $\|A\|$.

Par linéarité et puisque A est inversible, nous avons d'une part $A\,\delta x = \delta b \Rightarrow \|\delta x\| \leq \|A^{-1}\|\,\|\delta b\|$ et d'autre part $A\,x = b \Rightarrow \|b\| \leq \|A\|\,\|x\|$, ou encore de manière équivalente puisque b est non nul

$$\frac{1}{\|x\|} \;\leq\; \|A\|\,\frac{1}{\|b\|}.$$

Ainsi, nous obtenons l'estimation suivante :

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

et proposons la définition suivante :

Définition 1.2.11 *Nous appelons conditionnement de la matrice A relativement à la norme matricielle $\|.\|$ subordonnée à la norme vectorielle $\|.\|$, le nombre* $\mathrm{cond}(A) = \|A\|\,\|A^{-1}\|$.

Nous observons bien que le conditionnement sert à mesurer la sensibilité du système aux perturbations de b et de A.

Définition 1.2.12 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice inversible. Nous disons que*
- *un système linéaire est bien conditionné, si* $\mathrm{cond}(A)$ *n'est pas trop grand par rapport à un, ce qui correspond au conditionnement de l'identité ;*
- *un système linéaire est mal conditionné, si* $\mathrm{cond}(A)$ *est grand par rapport à un.*

Vérifions sur l'exemple précédent la cohérence de cette définition. Nous avons pour la norme $\|.\|_1$, $\mathrm{cond}_1(A) = \|A\|_1\,\|A^{-1}\|_1 \simeq 28\,375 \gg 1$ ou pour la norme $\|.\|_2$, $\mathrm{cond}_2(A) = \|A\|_2\,\|A^{-1}\|_2 \simeq 15\,514 \gg 1$. Les valeurs du conditionnement de la matrice A semblent être assez élevées indépendamment de la norme choisie.

Pour conclure cette partie, nous donnons quelques propriétés sur le conditionnement

Proposition 1.2.8 *Soient $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice carrée inversible et $\|.\|$ une norme matricielle subordonnée à la norme vectorielle $\|.\|$. Alors nous avons*
- $\mathrm{cond}(A^{-1}) = \mathrm{cond}(A)$.
- *Soit* $\alpha \in \mathbb{K}$, $\mathrm{cond}(\alpha\,A) = \mathrm{cond}(A)$.

- $\mathrm{cond}(I_n) = 1$.
- $\mathrm{cond}(A) \geq 1$.

L'inconvénient de la définition du conditionnement est qu'il fait apparaître $\|A^{-1}\|$ qui n'est pas facile à calculer d'autant plus que nous ne connaissons pas la forme explicite de la matrice A^{-1}. Dans le cas particulier d'une matrice hermitienne et pour la norme matricielle $\|.\|_2$, nous avons néanmoins le résultat suivant qui ne nécessite pas le calcul de A^{-1} mais seulement la connaissance des valeurs propres de A.

Proposition 1.2.9 *Soit A une matrice hermitienne ($A^\star = A$). Alors $\|A\|_2 = \rho(A)$. De plus, si A est une matrice hermitienne inversible et $(\lambda_i)_{1\leq i\leq n}$ ses valeurs propres. Alors,*

$$\mathrm{cond}_2(A) = \frac{\max\{|\lambda_i|, i = 1, \ldots, n\}}{\min\{|\lambda_i|, i = 1, \ldots, n\}}.$$

Démonstration : Appliquons la Proposition 1.2.7 et puisque A est hermitienne, nous montrons que $\|A\|_2^2 = \rho(A)^2$.

Supposons ensuite que A est une matrice hermitienne inversible. Pour calculer $\mathrm{cond}_2(A)$, il suffit de remarquer que $1/\lambda_i$, où $\lambda_i \neq 0$, est valeur propre de A^{-1} et donc en appliquant le résultat précédent à A^{-1} (qui est également une matrice hermitienne), nous avons

$$\|A^{-1}\|_2 = \rho(A^{-1}) = \frac{1}{\min\{|\lambda_i|, i = 1, \ldots, n\}}.$$

Nous en déduisons le résultat

$$\mathrm{cond}_2(A) = \|A\|_2 \, \|A^{-1}\|_2 = \frac{\max\{|\lambda_i|, i = 1, \ldots, n\}}{\min\{|\lambda_i|, i = 1, \ldots, n\}}.$$

□

Préconditionnement d'un système linéaire Pour remédier au problème du mauvais conditionnement d'une matrice, nous pouvons appliquer une méthode de préconditionnement. En effet, en vue de résoudre $Ax = b$ nous multiplions ce système d'équation à gauche par une matrice inversible P, il vient alors $P\,A\,x = P\,b$, avec P choisie de manière à ce que la matrice $P\,A$ soit mieux conditionnée que A (dans le cas le plus favorable, nous aurions $P = A^{-1}$). Cependant, il n'y a pas de méthode standard pour trouver la matrice P, le plus souvent nous chercherons une matrice à la fois facile à inverser et ¡¡ assez proche ¿¿ de A^{-1}.

Présentons à présent les deux types de méthodes pour la résolution d'un système linéaire : les méthodes directes et les méthodes itératives.

1.3 Méthodes directes

1.3.1 Méthodologie générale

Soient $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice inversible et un vecteur $b \in \mathbb{K}^n$, nous recherchons $x \in \mathbb{K}^n$ solution de $A\,x = b$. Pour cela, construisons des matrices M et $N \in \mathcal{M}_{n,n}(\mathbb{K})$ telles que $A = M\,N$, où M est facile à inverser (triangulaire ou unitaire) et N triangulaire. Le système s'écrit alors $N\,x = M^{-1}\,b$, que nous résolvons en deux étapes de la manière suivante :

$$\begin{cases} \text{trouver } y \in \mathbb{K}^n \text{ tel que } M\,y \,=\, b, \\ \\ \text{trouver } x \in \mathbb{K}^n \text{ tel que } N\,x \,=\, y. \end{cases}$$

1.3.2 Méthode de Gauss avec et sans pivot

Les méthodes directes permettent de calculer la solution exacte du problème (1.2) en un nombre fini d'étapes (en l'absence d'erreurs d'arrondi). La méthode directe la plus classique est la méthode d'élimination de Gauss ou Gauss-Jordan, qui consiste à décomposer la matrice A comme le produit $L\,U$ où L est une matrice triangulaire inférieure et U une matrice triangulaire supérieure[3].

L'élimination de Gauss ou l'élimination de Gauss-Jordan est un algorithme d'algèbre linéaire pour déterminer les solutions d'un système d'équations linéaires, pour déterminer le rang d'une matrice ou pour calculer l'inverse d'une matrice carrée inversible.

Méthode de Gauss sans pivot Avant de décrire l'algorithme général, nous commençons par présenter un exemple de la méthode d'élimination Gauss. Elle consiste seulement à remplacer le système initial par un système triangulaire équivalent. Considérons le système d'équations suivant :

$$\begin{cases} x_1 \;+\; 2\,x_2 \;+\; 2\,x_3 \;=\; 2, & l_1^{(1)} \\ x_1 \;+\; 3\,x_2 \;+\; 3\,x_3 \;=\; 2, & l_2^{(1)} \\ 3\,x_1 \;+\; 7\,x_2 \;+\; 8\,x_3 \;=\; 8. & l_3^{(1)} \end{cases}$$

La matrice $A^{(1)}$ et le vecteur $b^{(1)}$ sont donnés par

$$A^{(1)} = \begin{pmatrix} 1 & 2 & 2 \\ 1 & 3 & 3 \\ 3 & 7 & 8 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad b^{(1)} = \begin{pmatrix} 2 \\ 2 \\ 8 \end{pmatrix}.$$

Appliquons la première étape de la méthode d'élimination de Gauss. Pour cela, nous conservons intacte la première ligne et ajoutons un multiple de la première ligne aux deux autres lignes pour annuler les premiers coefficients de ces lignes : nous faisons la différence entre la deuxième ligne et la première, puis la différence entre la troisième et trois fois la première soit $l_1^{(2)} \leftarrow l_1^{(1)}, l_2^{(2)} \leftarrow$

[3]Cette méthode fut nommée d'après Carl Friedrich Gauss, mathématicien allemand (1777-1855) surnommé le prince des mathématiciens. Cependant, si l'on se réfère au livre chinois ¡¡ Les neuf chapitres sur l'art du calcul ¿¿ sa naissance remonte à la dynastie Han au premier siècle de notre ère. Elle constitue le huitième chapitre de ce livre sous le titre de la ¡¡ disposition rectangulaire ¿¿ et est présentée au moyen de dix-huit exercices [6].

$l_2^{(1)} - l_1^{(1)}$ et $l_3^{(2)} \leftarrow l_3^{(1)} - 3\,l_1^{(1)}$, il vient alors

$$\begin{cases} x_1 + 2\,x_2 + 2\,x_3 = 2, & l_1^{(2)} \\ 0 + x_2 + x_3 = 0, & l_2^{(2)} \\ 0 + x_2 + 2\,x_3 = 2, & l_3^{(2)} \end{cases}$$

ou encore $A^{(2)}\,x = b^{(2)}$ avec

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 2 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix}.$$

Remarquons que les opérations réalisées sur les lignes du système sont linéaires et réversibles, comme nous le verrons par la suite. Par conséquent, les systèmes $(l_1^{(1)}, l_2^{(1)}, l_3^{(1)})$ et $(l_1^{(2)}, l_2^{(2)}, l_3^{(2)})$ sont équivalents.

Ensuite, nous conservons les deux premières lignes et modifions la troisième en faisant apparaître un zéro sur la deuxième colonne. Pour cela, il suffit de remplacer la troisième ligne par la différence entre la troisième et la deuxième ligne soit $l_1^{(3)} \leftarrow l_1^{(2)}, l_2^{(3)} \leftarrow l_2^{(2)}$ et $l_3^{(3)} \leftarrow l_3^{(2)} - l_2^{(2)}$:

$$\begin{cases} x_1 + 2x_2 + 2x_3 = 2, & l_1^{(3)} \\ x_2 + x_3 = 0, & l_2^{(3)} \\ + x_3 = 2, & l_3^{(3)} \end{cases}$$

ou encore $A^{(3)}\,x = b^{(3)}$ avec

$$A^{(3)} = \begin{pmatrix} 1 & 2 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad b^{(3)} = \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix}.$$

Puisque la matrice $A^{(3)}$ est triangulaire, nous pouvons résoudre facilement ce système et la solution est donnée par $x = (2, -2, 2)^T$.

Le principe de la méthode consiste donc à se ramener par des opérations simples (combinaisons linéaires) à un système triangulaire équivalent qui sera alors facile à résoudre. Avant de décrire précisément l'algorithme, nous introduisons une forme de matrice bien particulière et présentons quelques-unes de ses propriétés.

Lemme 1.3.1 *Soit $B^{(k)} \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice de la forme*

$$B^{(k)} := \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & b_{k+1}^{(k)} & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & b_n^{(k)} & 0 & \dots & 0 \end{pmatrix}.$$

Alors nous avons

(i) $B^{(k)}\,B^{(l)} = 0$, *pour tout* $1 \leq k \leq l \leq n$,
(ii) *pour* $L^{(k)} = I_n - B^{(k)}$, $L^{(k)}$ *est inversible et* $[L^{(k)}]^{-1} = I_n + B^{(k)}$,
(iii) $L^{(k)}\,L^{(l)} = I_n - (B^{(k)} + B^{(l)})$, *pour tout* $1 \leq k \leq l \leq n$.

Démonstration : Il suffit d'effectuer le produit $B^{(k)}\,B^{(l)}$ et montrer que ce produit est nul lorsque $0 \leq k \leq l \leq n$. Puis les propriétés (ii) et (iii) se déduisent facilement.

□

Voyons maintenant comment étendre la méthode de Gauss-Jordan au cas d'un système quelconque. Nous nous intéressons au système linéaire $A\,x = b$, où A est une matrice carrée inversible de taille $n \times n$ et $b \in \mathbb{K}^n$.

Posons $A^{(1)} = A$ et $b^{(1)} = b$, à partir de ce changement de notation le système s'écrit

$$\begin{cases} a_{1,1}^{(1)}\,x_1 + a_{1,2}^{(1)}\,x_2 + \quad \dots \quad + a_{1,n}^{(1)}\,x_n &= b_1^{(1)}, \\ a_{2,1}^{(1)}\,x_1 + a_{2,2}^{(1)}\,x_2 + \quad \dots \quad + a_{2,n}^{(1)}\,x_n &= b_2^{(1)}, \\ \vdots & \vdots \\ a_{n,1}^{(1)}\,x_1 + a_{n,2}^{(1)}\,x_2 + \quad \dots \quad + a_{n,n}^{(1)}\,x_n &= b_n^{(1)}. \end{cases}$$

Supposons que $a_{1,1}^{(1)} \neq 0$ et appelons ce coefficient le premier *pivot*. Pour $i = 2, \dots, n$, remplaçons simplement la ligne i par une combinaison linéaire des lignes 1 et i de manière à faire apparaître des 0 sur la première colonne. Pour cela, posons pour $i = 2, \dots, n$

$$\alpha_i^{(1)} = \frac{a_{i,1}^{(1)}}{a_{1,1}^{(1)}}$$

et appliquons la transformation $l_i \mapsto l_i - \alpha_i^{(1)}\, l_1$, ou encore

$$\left\{\begin{array}{lllll} a_{1,1}^{(1)}\, x_1 & + a_{1,2}^{(1)}\, x_2 + & \ldots & + a_{1,n}^{(1)}\, x_n & = b_1^{(1)}, \\ 0\, x_1 & + (a_{2,2}^{(1)} - \alpha_2^{(1)}\, a_{1,2}^{(1)})\, x_2 + & \ldots & + (a_{2,n}^{(1)} - \alpha_2^{(1)}\, a_{1,n}^{(1)})\, x_n & = b_2^{(1)} - \alpha_2^{(1)}\, b_1^{(1)}, \\ \vdots & & & & \\ 0\, x_1 & + (a_{n,2}^{(1)} - \alpha_n^{(1)}\, a_{1,2}^{(1)})\, x_2 + & \ldots & + (a_{n,n}^{(1)} - \alpha_n^{(1)}\, a_{1,n}^{(1)})\, x_n & = b_n^{(1)} - \alpha_n^{(1)}\, b_1^{(1)}. \end{array}\right.$$

Nous aboutissons à un nouveau système

$$\left\{\begin{array}{lllll} a_{1,1}^{(1)}\, x_1 + & a_{1,2}^{(1)}\, x_2 + & \ldots & + a_{1,n}^{(1)}\, x_n & = b_1^{(1)}, \\ & a_{2,2}^{(2)}\, x_2 + & \ldots & + a_{2,n}^{(2)}\, x_n & = b_2^{(2)}, \\ & \vdots & & & \\ & a_{n,2}^{(2)}\, x_2 + & \ldots & + a_{n,n}^{(2)}\, x_n & = b_n^{(2)}, \end{array}\right.$$

avec

$$\left\{\begin{array}{l} \alpha_i^{(1)} = \dfrac{a_{i,1}^{(1)}}{a_{1,1}^{(1)}}, \quad i = 2, \ldots, n, \\ a_{i,j}^{(2)} = a_{i,j}^{(1)} - \alpha_i^{(1)}\, a_{1,j}^{(1)}, \quad i = 2, \ldots, n,\ j = 2, \ldots, n, \\ b_i^{(2)} = b_i^{(1)} - \alpha_i^{(1)}\, b_1^{(1)}, \quad i = 2, \ldots, n, \end{array}\right.$$

et la première ligne est inchangée, c'est-à-dire $a_{1,j}^{(2)} = a_{1,j}^{(1)}$ pour tout $1 \leq j \leq n$. D'un point de vue matriciel, ceci revient à résoudre le système équivalent $A^{(2)}\, x = b^{(2)}$, où $A^{(2)}$ et $b^{(2)}$ sont obtenus par $A^{(2)} = L^{(1)}\, A^{(1)}, \quad b^{(2)} = L^{(1)}\, b^{(1)}$, où $L^{(1)} = I_n - B^{(1)}$ et $B^{(1)}$ est donnée par

$$B^{(1)} := \begin{pmatrix} 0 & 0 & \ldots & \ldots & 0 \\ \alpha_2^{(1)} & 0 & & & \vdots \\ \vdots & \vdots & & & \vdots \\ \alpha_n^{(1)} & 0 & \ldots & \ldots & 0 \end{pmatrix}.$$

Vérifions le nombre d'opérations que cette première étape nécessite. La transformation de $A^{(1)}x = b^{(1)}$ en $A^{(2)}\, x = b^{(2)}$ s'opère en

- $(n-1)$ divisions,

– $n\,(n-1) = (n-1)^2 + (n-1)$ multiplications,
– $n\,(n-1) = (n-1)^2 + (n-1)$ additions.

Pour poursuivre l'algorithme, raisonnons alors par récurrence. Supposons qu'à l'étape $(k-1)$, nous avons construit le système $A^{(k)}\,x \,=\, b^{(k)}$ qui s'écrit sous la forme

$$\left\{\begin{array}{lllllll} a_{1,1}^{(1)}\,x_1 + & a_{1,2}^{(1)}\,x_2 + & \ldots & \ldots & + a_{1,n}^{(1)}\,x_n & = & b_1^{(1)}, \\ & a_{2,2}^{(2)}\,x_2 + & \ldots & \ldots & + a_{2,n}^{(2)}\,x_n & = & b_2^{(2)}, \\ & & \ddots & \vdots & \vdots & \vdots & \\ & & a_{k,k}^{(k)}\,x_k + & \ldots & + a_{k,n}^{(k)}\,x_n & = & b_k^{(k)}, \\ & & \vdots & \vdots & \vdots & \vdots & \\ & & a_{n,k}^{(k)}\,x_k + & \ldots & + a_{n,n}^{(k)}\,x_n & = & b_n^{(k)}. \end{array}\right.$$

L'étape suivante consiste à faire apparaître des 0 sur la k-ème colonne à partir de la $(k+1)$-ème ligne. Pour cela supposons que le coefficient $a_{k,k}^{(k)}$, aussi appelé pivot, est non nul. Alors les k premières lignes restent inchangées tandis que pour $i \geq k+1$,

$$\left\{\begin{array}{l} \alpha_i^{(k)} = \dfrac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}},\ i \,=\, k+1, \ldots, n, \\ \\ \\ a_{i,j}^{(k+1)} = a_{i,j}^{(k)} \,-\, \alpha_i^{(k)}\,a_{k,j}^{(k)},\ i \,=\, k+1, \ldots, n, \quad j = k+1, \ldots, n, \\ b_i^{(k+1)} = b_i^{(k)} \,-\, \alpha_i^{(k)}\,b_k^{(k)},\ i = k+1, \ldots, n. \end{array}\right.$$

À nouveau d'un point de vue matriciel cela revient à faire le produit $A^{(k+1)} \,=\, L^{(k)}\,A^{(k)}$ et $b^{(k+1)} = L^{(k)}\,b^{(k)}$, où $L^{(k)} \,=\, I_n - B^{(k)}$ et $B^{(k)}$ est donnée par

$$B^{(k)} := \left(\begin{array}{ccccccc} 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \alpha_{k+1}^{(k)} & 0 & \ldots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \ldots & 0 & \alpha_n^{(k)} & 0 & \ldots & 0 \end{array}\right)$$

et les $(\alpha_j^{(k)})_{k+1\leq j\leq n}$ sont donnés par

$$\alpha_j^{(k)} \;=\; \frac{a_{j,k}^{(k)}}{a_{k,k}^{(k)}}, \quad k+1 \leq j \leq n.$$

Cette étape permet de passer du système $A^{(k)}\,x \;=\; b^{(k)}$ à $A^{(k+1)}\,x \;=\; b^{(k+1)}$ et nécessite

- $(n-k)$ divisions,
- $(n-k+1)\,(n-k) = (n-k)^2 \;+\; (n-k)$ multiplications,
- $(n-k+1)\,(n-k) = (n-k)^2 \;+\; (n-k)$ additions.

En itérant ce processus jusqu'à la $(n-1)$-ème étape, nous obtenons finalement un système de la forme $A^{(n)}\,x \;=\; b^{(n)}$, c'est-à-dire

$$\left\{\begin{array}{lllll} a_{1,1}^{(1)}\,x_1 + & a_{1,2}^{(1)}\,x_2 + & \ldots & + a_{1,n}^{(1)}\,x_n & = \; b_1^{(1)}, \\ & a_{2,2}^{(2)}\,x_2 + & \ldots & + a_{2,n}^{(2)}\,x_n & = \; b_2^{(2)}, \\ & & \ddots & \vdots \quad\quad \vdots & \vdots \\ & & & a_{n,n}^{(n)}\,x_n & = \; b_n^{(n)}. \end{array}\right.$$

Remarquons que $A^{(n)}$ est une matrice triangulaire supérieure. Pour conclure la méthode d'élimination de Gauss, il reste à résoudre le système $A^{(n)}\,x \;=\; b^{(n)}$: nous commençons par la dernière ligne $x_n = b_n^{(n)}/a_{n,n}^{(n)}$. Ensuite, connaissant x_n nous calculons x_{n-1} en utilisant la $(n-1)$-ème ligne

$$x_{n-1} \;=\; \frac{1}{a_{n-1,n-1}^{(n-1)}} \left(b_{n-1}^{(n-1)} - a_{n-1,n}^{(n-1)} x_n\right).$$

Puis poursuivons la remontée

$$x_k = \frac{1}{a_{k,k}^{(k)}} \left(b_k^{(k)} - \sum_{i=k}^{n} a_{k,i}^{(k)} x_i\right), \quad k = n-2, \ldots, 1.$$

Au final, pour résoudre le système triangulaire nous avons recours à n divisions, $\sum_{k=1}^{n} k = n(n+1)/2$ multiplications et additions, ce qui signifie qu'au total,

- le nombre de divisions est de

$$(n-1) + \ldots + 1 \;+\; n \;=\; \frac{n(n+1)}{2},$$

- le nombre de multiplications est de

$$\begin{array}{rl} & (n-1)^2 + \ldots + 1^2 \;+\; (n-1) + \ldots + 1 \;+\; \dfrac{n(n+1)}{2} \\ = & \dfrac{n(n-1)(2n-1)}{6} + \dfrac{n^2}{2} \;=\; \dfrac{2n^3+n}{6}, \end{array}$$

– le nombre d'additions est de

$$\begin{aligned} & (n-1)^2 + \ldots + 1^2 \; + \; (n-1) + \ldots + 1 \; + \; \frac{n(n+1)}{2} \\ = \quad & \frac{n(n-1)(2n-1)}{6} + \frac{n^2}{2} = \frac{2n^3+n}{6}. \end{aligned}$$

En conclusion la méthode est en $O\left(2n^3/3\right)$, ce qui est nettement meilleur que le coût de calcul en $n!$ de l'algorithme de Cramer.

Pour mieux comprendre et analyser la méthode d'élimination de Gauss, nous pouvons ré-écrire l'algorithme de manière plus abstraite en utilisant seulement des produits matriciels. En effet, la méthode d'élimination de Gauss présentée plus haut consiste en fait à décomposer la matrice A comme le produit d'une matrice triangulaire inférieure L (comme ¡¡ Low ¿¿, bas en anglais) et d'une matrice triangulaire supérieure U (comme ¡¡ Up ¿¿, haut en anglais). Plus précisément, $A^{(n)} = L^{(n-1)}\, A^{(n-1)} = L^{(n-1)} \ldots L^{(1)}\, A$ et donc en appliquant le résultat du Lemme 1.3.1[(ii)], il vient

$$\begin{aligned} A &= \left(L^{(n-1)} \ldots L^{(1)}\right)^{-1} A^{(n)}, \\ &= \left(L^{(1)}\right)^{-1} \ldots \left(L^{(n-1)}\right)^{-1} A^{(n)}, \\ &= \left(I_n + B^{(1)}\right) \ldots \left(I_n + B^{(n-1)}\right) A^{(n)}. \end{aligned}$$

Puis, en appliquant le Lemme 1.3.1[(iii)] en changeant $B^{(k)}$ en $-B^{(k)}$, nous avons

$$\left(I_n + B^{(1)}\right) \ldots \left(I_n + B^{(n-1)}\right) = \left(I_n + B^{(1)} + \ldots + B^{(n-1)}\right)$$

et obtenons finalement $A = \left(I_n + B^{(1)} + \ldots + B^{(n-1)}\right) A^{(n)}$, où la matrice $L = \left(I_n + B^{(1)} + \ldots + B^{(n-1)}\right)$ est une matrice triangulaire inférieure tandis que $U = A^{(n)}$ est une matrice triangulaire supérieure. Cependant, cette décomposition n'est pas toujours possible pour une matrice A quelconque puisqu'il est nécessaire qu'à chaque étape le pivot $a_{k,k}^{(k)}$ soit non nul. Démontrons alors un résultat qui donne une condition suffisante pour l'application de la méthode d'élimination de Gauss sans pivot.

Théorème 1.3.1 (Existence et unicité de la décomposition $L\,U$) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice inversible. Supposons que toutes les sous-matrices principales d'ordre 1 à $n-1$ de A soient inversibles. Alors il existe une unique matrice L triangulaire inférieure à diagonale unité (ne comportant que des 1 sur la diagonale) et une matrice U triangulaire supérieure inversible telles que $A = L\,U$.*

Démonstration : Établissons d'abord le résultat d'unicité. Soient (L_α, U_α) et (L_β, U_β) telles que $L_\alpha\, U_\alpha = A = L_\beta\, U_\beta$, où L_α et L_β sont des matrices triangulaires inférieures à diagonale unité et donc inversibles. Étant donné qu'U_α et U_β sont des matrices triangulaires supérieures inversibles, nous pouvons écrire

$$L_\beta^{-1}\, L_\alpha = U_\beta\, U_\alpha^{-1}.$$

Or, $L_\beta^{-1}\, L_\alpha$ est une matrice triangulaire inférieure à diagonale unité et $U_\beta\, U_\alpha^{-1}$ est une matrice triangulaire supérieure, elle est donc diagonale et $L_\beta^{-1}\, L_\alpha = I_n$ et $U_\beta\, U_\alpha^{-1} = I_n$. Ainsi, $L_\alpha = L_\beta$ et $U_\alpha = U_\beta$, ce qui prouve que la décomposition est unique.

Démontrons maintenant l'existence d'une telle décomposition. Nous avons vu que pour pouvoir appliquer la méthode d'élimination de Gauss sans pivot, il suffit qu'à chaque étape le pivot $a_{i,i}^{(i)}$ soit

non nul. Prouvons alors par récurrence que lorsque la matrice A a toutes ses sous-matrices principales inversibles, $a_{i,i}^{(i)}$ est bien différent de zéro.

Tout d'abord, puisque la première sous-matrice est de déterminant non nul, nous avons $a_{1,1} \neq 0$ et donc le premier pivot $a_{1,1}^{(1)} = a_{1,1}$ est bien non nul et pouvons effectuer la première étape de l'élimination de Gauss.

Supposons ensuite que pour tout $k \in \{1, \ldots, i-1\}$ le coefficient $a_{k,k}^{(k)}$ est différent de zéro, dans ce cas en suivant la méthode d'élimination de Gauss, nous avons

$$A = A^{(1)} = \left(L^{(i-1)} \ldots L^{(1)}\right)^{-1} A^{(i)}.$$

Développons alors le produit par blocs, il vient

$$\begin{pmatrix} A_i & \times \\ \times & \times \end{pmatrix} = \begin{pmatrix} L_i & 0 \\ \times & \times \end{pmatrix} \begin{pmatrix} A_i^{(i)} & \times \\ \times & \times \end{pmatrix},$$

où A_i (respectivement $A_i^{(i)}$) est la sous-matrice principale de A (respectivement $A^{(i)}$) tandis que $L_i \in \mathcal{M}_{i,i}(\mathbb{K})$ est une matrice tridiagonale inférieure avec uniquement des 1 sur la diagonale. Puisque $A_i^{(i)}$ est une matrice triangulaire supérieure, nous avons

$$\det(A_i) = \det\left(L_i A_i^{(i)}\right) = \det(L_i) \times \det\left(A_i^{(i)}\right) = 1 \times a_{1,1}^{(1)} \ldots a_{i,i}^{(i)}.$$

Or, puisque toutes les sous-matrices principales sont inversibles $\det(A_i)$ est différent de zéro et donc le produit $\prod_{k=1}^{i} a_{k,k}^{(k)}$ est aussi non nul et en particulier $a_{i,i}^{(i)} \neq 0$. Nous pouvons donc effectuer une nouvelle étape de la méthode d'élimination de Gauss.

Au final, nous obtenons la décomposition

$$\begin{aligned} A &= (L^{(1)})^{-1} L^{(1)} A^{(1)}, \\ &= (L^{(1)})^{-1} A^{(2)}, \\ &= \vdots \\ &= \left(I_n + B^{(1)} \ldots + B^{(n-1)}\right) A^{(n-1)}. \end{aligned}$$

Notons U la matrice triangulaire supérieure $A^{(n-1)}$ et $L = (L^{(1)})^{-1} \ldots (L^{(n-1)})^{-1}$. Nous obtenons bien $A = LU$. □

À l'aide de ce résultat, nous pouvons démontrer le corollaire suivant qui est particulièrement important pour les applications.

Corollaire 1.3.1 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice hermitienne définie positive. Alors A admet une décomposition $L\,U$.*

Démonstration : Les sous-matrices principales sont symétriques définies positives et donc inversibles. Par application du Théorème 1.3.1, A admet une décomposition $L\,U$. □

La décomposition $L\,U$ fournit donc un algorithme exact pour résoudre un système linéaire dont la solution $x \in \mathbb{K}^n$ vérifie $A\,x = b$. En écrivant la matrice A sous la forme $A = LU$, la résolution se décompose en deux étapes :

$$\begin{cases} \text{trouver } y \in \mathbb{K}^n \text{ tel que } L\,y = b, \\ \\ \text{trouver } x \in \mathbb{K}^n \text{ tel que } U\,x = y, \end{cases}$$

et chaque étape porte sur un système triangulaire. Or, nous avons vu que les systèmes linéaires avec des matrices triangulaires peuvent être aisément résolus en utilisant un algorithme de descente puis de remontée.

Remarque 1.3.1 *Notons que lorsque nous voulons résoudre ce système pour différents $b \in \mathbb{K}^n$, il est plus optimal de réaliser la décomposition $L\,U$ une fois pour toutes et de résoudre les systèmes linéaires avec les matrices triangulaires pour les différents b plutôt que d'utiliser l'élimination de Gauss-Jordan à de multiples reprises*.

L'algorithme correspondant peut être décrit sous une forme compacte :

Algorithme 1. Élimination de Gauss sans pivot

Pour $k = 1, \ldots, n-1$

Pour $i = k+1, \ldots, n$:

- calculer

$$\alpha_i^{(k)} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}.$$

Pour $j = k+1, \ldots, n$

$$a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - \alpha_i^{(k)}\, a_{k,j}^{(k)}.$$

Fin de la boucle sur j,

- puis le second membre

$$b_i^{(k+1)} = b_i^{(k)} - \alpha_i^{(k)}\, b_k^{(k)}.$$

Fin de la boucle sur i

Fin de la boucle sur k.

Cet algorithme peut être implanté sur ordinateur pour résoudre des systèmes avec des milliers d'inconnues et d'équations. Il est cependant numériquement peu stable, c'est-à-dire que les erreurs d'arrondi effectuées durant le calcul sont accumulées et le résultat obtenu peut être loin de la solution surtout lorsque le système est mal conditionné. Pour remédier à ce problème, nous ajoutons un étape supplémentaire de permutation qui améliore la stabilité de l'algorithme.

Méthode de Gauss avec pivot Jusqu'ici nous avons supposé qu'à chaque étape de la méthode d'élimination de Gauss, le pivot $a_{k,k}^{(k)} \neq 0$, l'inconvénient est que cette condition n'est pas assurée pour une matrice inversible quelconque. D'autre part, même lorsque cette condition est satisfaite, le pivot peut prendre des valeurs $|a_{k,k}^{(k)}|$ proches de 0, ce qui conduit à des instabilités numériques. C'est pourquoi en général, nous avons recours à une étape supplémentaire de permutation de lignes pour remplacer un pivot éventuellement nul par un autre pivot qui lui sera non nul à coup sûr. Cette étape permet également de stabiliser l'algorithme par rapport aux erreurs d'arrondi. En effet, considérons simplement le système à deux inconnues. Soit $\varepsilon > 0$ un petit paramètre,

$$\begin{cases} \varepsilon\, x_1 + x_2 &= 1, \\ x_1 + x_2 &= 2, \end{cases}$$

dont la solution exacte est $x_1 = 1/(1-\varepsilon)$ et $x_2 = (1-2\varepsilon)/(1-\varepsilon)$. D'une part, lorsque nous appliquons rigoureusement l'algorithme d'élimination de Gauss sans pivot avec des erreurs d'arrondi de l'ordre de ε, nous obtenons

$$\begin{cases} \varepsilon\, x_1 + x_2 &= 1, \\ 1 - \dfrac{1}{\varepsilon}\, x_2 &= 2 - \dfrac{1}{\varepsilon}. \end{cases}$$

Par conséquent,

$$x_2 = \left(2 - \frac{1}{\varepsilon}\right)\left(1 - \frac{1}{\varepsilon}\right)^{-1}$$

et en supposant que l'erreur de calcul est de l'ordre de ε, nous obtenons une valeur approchée de x_2 donnée par $x_2^{app} = 1$ et en utilisant la première ligne, nous avons ensuite $x_1^{app} = 0$, ce qui est très loin de la valeur exacte. Le résultat approché obtenu pour x_1 n'est pas correct. En revanche, si avant d'appliquer la méthode d'élimination de Gauss nous échangeons les deux équations, nous avons alors le système

$$\begin{cases} x_1 + x_2 &= 2, \\ \varepsilon\, x_1 + x_2 &= 1. \end{cases}$$

Cette fois-ci, le pivot est égal à un et la méthode d'élimination de Gauss donne pour la deuxième équation $(1 - \varepsilon)\, x_2 = 1 - 2\,\varepsilon$ et donc le système est équivalent à

$$\begin{cases} x_1 + x_2 &= 2, \\ (1-\varepsilon)\, x_2 &= 1 - 2\varepsilon. \end{cases}$$

De nouveau comme valeur approchée de x_2, sachant que l'erreur est de l'ordre ε, nous obtenons $x_2^{app} = 1$. Puis en substituant cette valeur approchée dans la première équation, nous trouvons comme valeur approchée de $x_1, x_1^{app} = 1$. Cette fois-ci les deux valeurs numériques sont très proches de la solution exacte.

En définitive, la méthode que nous avons appliquée ici est l'algorithme d'élimination de Gauss avec pivot. C'est une méthode directe de résolution de système linéaire qui permet de transformer un système en un système équivalent composé d'une matrice triangulaire. L'algorithme est le même que celui décrit pour la méthode d'élimination de Gauss sans pivot mais avec une étape supplémentaire de permutation permettant d'obtenir le pivot le plus grand possible. Cela permet d'éviter de travailler avec un pivot proche de zéro, ce qui peut introduire des erreurs numériques grossières et au final donner une mauvaise approximation de la solution. Comme pour la méthode d'élimination de Gauss sans pivot après avoir obtenu un système triangulaire, nous résolvons le système à l'aide d'un algorithme de remontée.

L'algorithme est directement inspiré de l'algorithme sans pivot avec une étape supplémentaire de permutation.

Algorithme 2. Élimination de Gauss avec pivot

Pour $k = 1, \ldots, n-1$

Pour $i = k+1, \ldots, n$:

- rechercher $k_0 \in \{k, \ldots, n\}$ tel que

$$|a_{k_0,k}^{(k)}| = \max\{|a_{l,k}^{(k)}|, l \in \{k, \ldots, n\}\}$$

et permuter les lignes k et k_0,

- calculer

$$\alpha_i^{(k)} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}.$$

Pour $j = k+1, \ldots, n$

$$a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - \alpha_i^{(k)} a_{k,j}^{(k)}.$$

Fin de la boucle sur j,

- puis le second membre

$$b_i^{(k+1)} = b_i^{(k)} - \alpha_i^{(k)} b_k^{(k)}.$$

Fin de la boucle sur i

Fin de la boucle sur k.

Nous avons donc le résultat suivant qui assure l'existence d'une décomposition LU pour n'importe quelle matrice inversible.

Théorème 1.3.2 (Décomposition LU d'une matrice inversible) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice inversible. Alors la matrice A admet une décomposition LU à quelques permutations près $PA = LU$ ou $A = P^T LU$, où P est une matrice de permutations (de même pour P^T), L est une matrice triangulaire inférieure ne contenant que des 1 sur sa diagonale et U une matrice triangulaire supérieure.*

Démonstration : Nous ne présentons pas le détail de la preuve qui est essentiellement technique, nous renvoyons le lecteur au livre [7] pour plus de détails. L'idée de la preuve consiste à démontrer

que la méthode d'élimination de Gauss avec pivot est comme la méthode d'élimination de Gauss sans pivot une décomposition $L\,U$ mais avec en plus à chaque étape une permutation des lignes. Cette permutation permet d'assurer à chaque étape que le pivot $a_{k,k}^{(k)}$ est bien non nul. En effet, la méthode d'élimination de Gauss avec pivot peut s'écrire comme le produit suivant

$$A^{(k)} \;=\; L^{(k-1)}\,P^{(k-1)}\ldots L^{(1)}\,P^{(1)}\,A^{(1)},$$

où $L^{(i)}$ est la matrice déjà construite lors de la méthode d'élimination de Gauss sans pivot et $P^{(i)}$ est une matrice de permutations. Ainsi,

$$\begin{aligned}
|\det(A)| \;&=\; |\det(A^{(k)})| = \left|\det\begin{pmatrix}
a_{1,1}^{(1)} & a_{1,2}^{(1)} & \ldots & \ldots & a_{1,n}^{(1)} \\
 & a_{2,2}^{(2)} & \ldots & \ldots & a_{2,n}^{(2)} \\
 & & \ddots & & \vdots \\
 & & a_{k,k}^{(k)} & \ldots & a_{k,n}^{(k)} \\
 & & \vdots & & \vdots \\
 & & a_{n,k}^{(k)} & \ldots & a_{n,n}^{(k)}
\end{pmatrix}\right|, \\
&=\; a_{1,1}^{(1)}\ldots a_{k-1,k-1}^{(k-1)}\left|\det\begin{pmatrix}
a_{k,k}^{(k)} & \ldots & a_{k,n}^{(k)} \\
\vdots & & \vdots \\
a_{n,k}^{(k)} & \ldots & a_{n,n}^{(k)}
\end{pmatrix}\right|.
\end{aligned}$$

Puisque $\det(A)$ est non nul, le déterminant de la sous-matrice est aussi non nul et donc il existe forcément un coefficient $a_{i,k}^{(k)}$ qui est non nul. Nous pouvons donc appliquer la stratégie de l'élimination de Gauss en ayant au préalable effectué si nécessaire une permutation des lignes.

Pour conclure la preuve, nous avons vu qu'à la k-ème étape, nous sélectionnons un pivot à la ligne $p_k \in \{k,\ldots,n\}$ et permutons ensuite les lignes k et p_k. Nous observons alors que la factorisation LU que nous obtenons est la même que si nous permutions les lignes de la matrice A, $k \leftrightarrow p_k$, $k = 1,\ldots,n-1$ avant d'effectuer la factorisation, ce qui correspondrait à une factorisation sans pivot de la matrice $P\,A$. □

1.3.3 Factorisation de Cholesky

La factorisation de Cholesky[4] consiste pour une matrice hermitienne définie positive A à déterminer une matrice triangulaire supérieure R telle que : $A \;=\; R^{\star}\,R$. La matrice R est en quelque sorte la ¡¡ racine carrée ¿¿ de A. Cette décomposition permet notamment de résoudre des systèmes linéaires du type (1.2) ou de calculer le déterminant de A qui n'est rien d'autre que le carré du produit des éléments

[4]Cette méthode a d'abord été mise au point par le mathématicien et militaire français André-Louis Cholesky (1875-1918), puis redécouverte seulement dans les années 1940.

diagonaux de R. Par la suite, nous démontrons l'existence et l'unicité d'une telle décomposition pour une matrice hermitienne, définie positive et proposons ensuite un algorithme de calcul.

Théorème 1.3.3 (Factorisation de Cholesky d'une matrice) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice hermitienne et définie positive. Alors il existe au moins une matrice triangulaire supérieure R telle que $A = R^\star R$.*

Si de plus les éléments diagonaux de la matrice R sont tous positifs alors la factorisation correspondante est unique.

Démonstration : Nous présentons ici la preuve, proposée par P. S. Dwyer en 1945 [15], qui se décompose en deux étapes : pour l'existence, il s'agit d'effectuer une décomposition LU puis de modifier L et U pour qu'elles aient la même diagonale. Puis, sachant que cette décomposition existe, nous construisons l'algorithme en effectuant le produit $R^\star R$ ce qui permet de calculer les coefficients de la matrice R de manière unique en imposant que les coefficients diagonaux soient positifs.

Tout d'abord, puisque la matrice A est définie positive, nous appliquons le Corollaire 1.3.1, ce qui assure que la matrice A admet une unique décomposition $L\,U$. De plus, en notant A_k la sous-matrice principale d'ordre k de A et en développant le produit par bloc, il vient alors

$$\det(A_k) \,=\, \det((L\,U)_k) \,=\, 1 \,\times\, u_{1,1} \,\times\, \ldots \,\times\, u_{k,k}.$$

D'une part, $\det(A_k)$ correspond au produit des valeurs propres de A_k qui sont strictement positives, ce qui signifie que $\det(A_k) > 0$. Ainsi, en raisonnant par récurrence, nous démontrons que $u_{k,k} > 0$ pour tout $1 \leq k \leq n$, ce qui permet de définir les matrices $\Lambda,\ R,\ U \in \mathcal{M}_{n,n}(\mathbb{K})$ comme suit :

$$\Lambda \,=\, \mathrm{diag}(\sqrt{u_{1,1}}, \ldots, \sqrt{u_{n,n}}), \qquad S \,=\, L\,\Lambda, \qquad R \,=\, \Lambda^{-1}\,U$$

et nous avons bien

$$A \;=\; L\,\Lambda\,\Lambda^{-1}\,U \;=\; S\,R.$$

De plus, comme la matrice A est hermitienne, $S\,R \;=\; R^\star\,S^\star$, ou encore puisque la matrice R est inversible $(R^\star)^{-1}\,S \;=\; S^\star\,R^{-1}$.

D'une part, la matrice de gauche est le produit de deux matrices triangulaires inférieures S et $(R^\star)^{-1}$. D'après la Proposition 1.2.3, la matrice $(R^\star)^{-1}\,S$ est donc triangulaire inférieure. D'autre part, la matrice de droite $S^\star\,R^{-1}$ est pour des raisons identiques, une matrice triangulaire supérieure. Cela signifie donc que $S^\star\,R^{-1}$ est tout simplement une matrice diagonale ne contenant que des 1 sur sa diagonale, c'est donc l'identité autrement dit $S^\star \;=\; R$. Ainsi, la matrice A se décompose comme $A \;=\; R^\star\,R$.

Enfin, dans la pratique pour construire la matrice R, nous cherchons la matrice R sous la forme

$$R \,=\, \begin{pmatrix} r_{1,1} & r_{1,2} & \ldots & r_{1,n} \\ & r_{2,2} & \ldots & r_{2,n} \\ & & \ddots & \vdots \\ & & & r_{n,n} \end{pmatrix}.$$

De l'égalité $A = R^\star R$, nous déduisons :

$$a_{i,j} = (R^\star R)_{i,j} = \sum_{k=1}^{n} \overline{r}_{k,i}\, r_{k,j} = \sum_{k=1}^{\min(i,j)} \overline{r}_{k,i}\, r_{k,j}, \qquad 1 \leq i,j \leq n,$$

puisque $r_{p,q} = 0$ si $1 \leq q < p \leq n$.

La matrice A étant hermitienne, il suffit que les relations ci-dessus soient vérifiées pour $i \leq j$, c'est-à-dire

$$a_{i,j} = \sum_{k=1}^{i} \overline{r}_{k,i}\, r_{k,j}, \qquad 1 \leq i \leq j \leq n.$$

Pour $i = 1$, nous déterminons la première colonne de R : le coefficient $r_{1,1}$ est choisi strictement positif

$$[j=1] \quad a_{1,1} = \overline{r}_{1,1}\, r_{1,1} \quad \Rightarrow \quad r_{1,1} = \sqrt{a_{1,1}} > 0,$$

et le choix des autres coefficients est alors déterminé par

$$\begin{aligned}
&[j=2] \quad a_{1,2} = r_{1,1}\, r_{1,2} \quad \Rightarrow \quad r_{1,2} = \frac{a_{1,2}}{r_{1,1}},\\
&\vdots\\
&[j=n] \quad a_{1,n} = r_{1,1}\, r_{1,n} \quad \Rightarrow \quad r_{1,n} = \frac{a_{1,n}}{r_{1,1}}.
\end{aligned}$$

Raisonnons ensuite par récurrence. Pour $i \geq 1$ fixé, nous supposons que les coefficients $(r_{i-1,j})_{j=i-1,\ldots,n}$ sont connus et déterminons alors la i-ème ligne de R. Pour cela, écrivons d'abord le produit par bloc pour le calcul de la sous-matrice principale A_i de A, il vient alors

$$A_i = \begin{pmatrix} A_{i-1} & \alpha \\ \alpha^\star & a_{i,i} \end{pmatrix} = \begin{pmatrix} R_{i-1}^\star & 0 \\ \rho^\star & \overline{r}_{i,i} \end{pmatrix} \begin{pmatrix} R_{i-1} & \rho \\ 0 & r_{i,i} \end{pmatrix},$$

avec $\alpha = (a_{1,i}, \ldots, a_{i-1,i})^T \in \mathbb{K}^{i-1}$ et $\rho = (r_{1,i}, \ldots, r_{i-1,i})^T \in \mathbb{K}^{i-1}$ lequel est déjà déterminé. Par identification, nous obtenons d'abord

$$a_{i,i} = \rho^\star\, \rho + \overline{r}_{i,i}\, r_{i,i} \tag{1.4}$$

ou encore en utilisant les propriétés du déterminant

$$\det(A_i) = \overline{r}_{i,i} \det(R_{i-1}^\star) \times r_{i,i} \det(R_{i-1}) = \overline{r}_{i,i}\, r_{i,i} \det(A_{i-1})$$

et puisque A_i et A_{i-1} sont définies positives, nous avons bien

$$\overline{r}_{i,i}\, r_{i,i} = \frac{\det(A_i)}{\det(A_{i-1})} > 0.$$

Nous sommes alors en mesure de déterminer le seul réel positif vérifiant l'égalité (1.4)

$$r_{i,i} = \sqrt{a_{i,i} - \sum_{k=1}^{i-1} |r_{k,i}|^2}.$$

Ensuite, le calcul du terme $a_{i,j}$ pour $j > i$ permet de définir les autres composantes

$$\begin{aligned}
[j = i+1] \quad & a_{i,i+1} = \overline{r}_{1,i}\, r_{1,i+1} + \ldots + \overline{r}_{i,i}\, r_{i,i+1} \\
& \Rightarrow r_{i,i+1} = \frac{a_{i,i+1} - \sum_{k=1}^{i-1} \overline{r}_{k,i}\, r_{k,i+1}}{r_{i,i}}, \\
\vdots & \\
[j = n] \qquad & a_{i,n} = \overline{r}_{1,i}\, r_{1,n} + \ldots + \overline{r}_{i,i}\, r_{i,n} \\
& \Rightarrow r_{i,n} = \frac{a_{i,n} - \sum_{k=1}^{i-1} \overline{r}_{k,i}\, r_{k,n}}{r_{i,i}}.
\end{aligned}$$

□

L'algorithme correspondant peut être décrit sous une forme compacte comme suit :

Algorithme 3. Factorisation de Cholesky

Pour $i = 1, \ldots, n$

- calculer

$$r_{i,i} = \sqrt{a_{i,i} - \sum_{k=1}^{i-1} |r_{k,i}|^2}.$$

Pour $j = i+1, \ldots, n$

$$r_{i,j} = \frac{1}{r_{i,i}} \left(a_{i,j} - \sum_{k=1}^{i-1} \overline{r}_{k,i}\, r_{k,j} \right).$$

Fin de la boucle sur j,

Fin de la boucle sur i.

Exemple 1.3.1 *Soit $A \in \mathcal{M}_{3,3}(\mathbb{R})$ donnée par*

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix},$$

nous cherchons une matrice R triangulaire supérieure vérifiant $R^T R = A$. Pour cela, écrivons simplement

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ 0 & r_{2,2} & r_{2,3} \\ 0 & 0 & r_{3,3} \end{pmatrix},$$

et en effectuant le produit $R^T R$, calculons d'abord $r_{1,1} = 1$. Puis nous en déduisons $r_{1,2} = 1$ puis $r_{1,3} = 1$. En passant à la deuxième ligne, il vient $r_{2,2} = 1$, puis $r_{2,3} = 1$. Enfin la dernière ligne donne $r_{3,3} = 1$.

1.3.4 Factorisation $Q\,R$

L'objectif de cette partie est de proposer une méthode de factorisation des matrices A inversibles non nécessairement définies positives. Nous allons voir que toute matrice inversible peut être obtenue comme le produit de deux matrices Q et R. L'originalité de cette factorisation réside dans le fait que nous ne cherchons pas Q triangulaire mais pleine et orthogonale, c'est-à-dire $Q^T Q = I_n$, alors que la matrice R, pour sa part, est triangulaire supérieure.

Remarquons que lorsqu'une telle décomposition est connue, le système linéaire de la forme $A\,x = b$ se résout facilement. En effet, cela revient à trouver $x \in \mathbb{R}^n$ tel que $Q\,R\,x = b$. En multipliant alors par Q^T, puisque Q est orthogonale, il vient $R\,x = Q^T\,b$ qui peut être résolu facilement puisque R est triangulaire supérieure. Nous verrons également au chapitre suivant que la décomposition $Q\,R$ permet de définir un algorithme itératif pour la recherche de valeurs propres et de vecteurs propres d'une matrice. Cette décomposition est donc très utile dans la pratique bien qu'un peu plus coûteuse en termes de complexité que la factorisation $L\,U$.

Il existe plusieurs méthodes pour réaliser cette décomposition :

- la méthode de Householder où Q est obtenue par produits successifs de matrices orthogonales élémentaires ;
- la méthode de Givens où Q est obtenue par produits successifs de matrices de rotations planes ;
- la méthode de Schmidt, basée sur le procédé de Gram-Schmidt (voir Exercice 1.4.5).

Nous avons choisi ici de présenter la méthode de Householder qui mène à la factorisation $Q\,R$ d'une matrice A inversible.

Définition 1.3.1 *Soit $u \in \mathbb{R}^n$. Nous appelons matrice de Householder, la matrice $H_u \in \mathcal{M}_{n,n}(\mathbb{R})$ donnée par*

$$H_u = I_n - 2\,\frac{u\,u^T}{\|u\|^2}, \qquad (1.5)$$

où $\|u\|^2 = u^T\,u$ lorsque u est non nul, et $H_u = I_n$ lorsque $u = 0$.

La matrice de Householder satisfait quelques propriétés particulières.

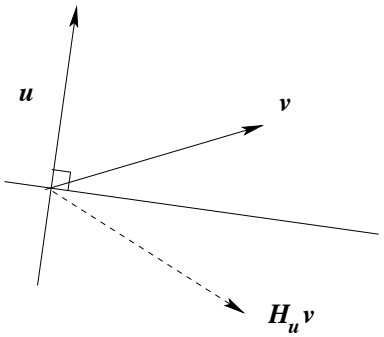

FIG. 1.3 – La matrice de Householder représente une réflection par rapport au vecteur u.

Proposition 1.3.1 *Soit $u \in \mathbb{R}^n$. Alors,*
(i) H_u est symétrique, c'est-à-dire $H_u = H_u^T$,
(ii) H_u est inversible et $H_u^{-1} = H_u$,
(iii) $H_u\,v = -v$, pour tout $v \in \mathbb{R}^n$ colinéaire à u et $H_u\,v = v$ pour tout $v \in \mathbb{R}^n$ orthogonal à u,
(iv) le déterminant de H_u est égal à -1 lorsque u est non nul et égal à $+1$ lorsque $u = 0$.
En définitive H_u est la matrice de la réflection d'hyperplan $\mathrm{Vect}\{u\}^T$.

Démonstration : La démonstration étant triviale lorsque $u = 0$, considérons seulement le cas où le vecteur u est non nul. D'après la définition de la matrice de Householder, nous vérifions facilement que la matrice H_u est symétrique.

Nous vérifions ensuite la propriété (ii). Pour cela, calculons

$$H_u\,H_u \;=\; I_n \;-\; 4\,\frac{u\,u^T}{\|u\|^2} \;+\; 4\,\frac{u\,u^T}{\|u\|^2}\,\frac{u\,u^T}{\|u\|^2}.$$

Or, comme $(u\,u^T)\,(u\,u^T) \;=\; u\,(u^T\,u)\,u^T = \|u\|^2\,u\,u^T$, il vient

$$H_u\,H_u \;=\; I_n \;-\; 4\,\frac{u\,u^T}{\|u\|^2} \;+\; 4\,\frac{u\,u^T}{\|u\|^2} \;=\; I_n$$

et donc $H_u^{-1} \;=\; H_u$.

Ensuite pour démontrer (iii), prenons d'une part $v \in \mathbb{R}^n$ tel que $\langle v, u\rangle = u^T\,v = 0$. Alors

$$H_u\,v \;=\; v \;-\; \frac{2}{\|u\|^2}\,u\,(u^T\,v) \;=\; v.$$

D'autre part, pour $v \;\in\; \mathbb{R}^n$ colinéaire à u, c'est-à-dire qu'il existe $\alpha \in \mathbb{R}$ tel que $v \;=\; \alpha\,u$, nous avons

$$H_u\,v \;=\; \alpha\,u \;-\; \frac{2\,\alpha}{\|u\|^2}\,(u\,u^T)\,u \;=\; \alpha\,u \;-\; 2\,\alpha\,u \;=\; -v.$$

Enfin, nous démontrons (iv). Pour $u \in \mathbb{R}^n$, posons $u_1 \;=\; u$, puis appliquons le résultat précédent en formant une base de $\mathbb{R}^n$ en complétant u_1 par une famille de vecteurs libres $(u_j)_{2\le j\le n}$ et orthogonaux à u_1. Nous notons alors U la matrice composée des colonnes $(u_j)_{1\le j\le n}$. En utilisant la propriété

(iii), nous obtenons

$$H_u\,U \;=\; H_u\,(u_1,u_2,\ldots,u_n) \;=\; (u_1,u_2,\ldots,u_n)\begin{pmatrix} -1 & 0^T_{\mathbb{R}^{n-1}} \\ 0_{\mathbb{R}^{n-1}} & I_{n-1} \end{pmatrix}$$

et donc en appliquant les propriétés du déterminant

$$\det(H_u)\det(U) \;=\; \det(H_u\,U) \;=\; -1 \,\times\, \det(U).$$

Puisque la matrice U est inversible, nous avons $\det(U) \neq 0$ et $\det(H_u) = -1$. □

À partir de la Proposition 1.3.1, nous vérifions que pour tout $v \in \mathbb{R}^n$ qui s'écrit aussi $v = x + y$ avec x colinéaire à u et y orthogonal à u, $H_u\,v \;=\; H_u\,(x \,+\, y) \;=\; y \,-\, x$, c'est bien une réflection d'hyperplan $\mathrm{Vect}\{u\}^T$. C'est cette propriété que nous allons maintenant exploiter pour démontrer le résultat qui suit.

Proposition 1.3.2 *Pour tout $v \in \mathbb{R}^m$, $m \geq 1$ tel que $\|v\| = 1$, il existe $u \in \mathbb{R}^m$ tel que $H_u\,v \;=\; e_1$, où $e_1 = (1,0,\ldots,0)^T \;\in \mathbb{R}^m$ le vecteur de la base canonique dont seule la première composante est non nulle.*

Démonstration : Tout d'abord si v est colinéaire à e_1, posons $u \;=\; 0$ et $H_u \;=\; I_m$. En revanche lorsque v n'est pas colinéaire à e_1, posons $u \;=\; v \,-\, e_1$. Un calcul direct montre d'une part

$$H_u\,v \;=\; v \,-\, 2\,\frac{(v \,-\, e_1)(v \,-\, e_1)^T}{\|v \,-\, e_1\|^2}\,v \;=\; v \,-\, 2\,\frac{(v \,-\, e_1)^T\,v}{\|v \,-\, e_1\|^2}\;(v \,-\, e_1).$$

D'autre part, puisque $e_1^T\,e_1 \;=\; 1 \;=\; v^T\,v$ et en utilisant la symétrie du produit scalaire, nous avons

$$\begin{aligned} \|v \,-\, e_1\|^2 \;&=\; (v \,-\, e_1)^T\,v \,-\, v^T\,e_1 + 1 \;=\; (v \,-\, e_1)^T\,v \,-\, v^T\,(e_1 - v) \\ &=\; 2\,(v - e_1)^T\,v. \end{aligned}$$

En combinant, ces deux derniers résultats, nous obtenons $H_u\,v = \,e_1$. □

Maintenant que nous avons étudié les propriétés des matrices de Householder, nous pouvons effectuer la factorisation $Q\,R$.

Théorème 1.3.4 (Existence et unicité de la décomposition $Q\,R$) *Soit $A \;\in\; \mathcal{M}_{n,n}(\mathbb{R})$ une matrice inversible. Alors il existe une matrice orthogonale Q et une matrice triangulaire supérieure R telles que $A \;=\; Q\,R$.*

Démonstration : Soit $A \;\in\; \mathcal{M}_{n,n}(\mathbb{R})$ une matrice inversible. L'idée de la factorisation $Q\,R$ consiste à multiplier la matrice A par des matrices de Householder de manière à faire apparaître au final une matrice triangulaire comme nous l'avons fait pour la décomposition $L\,U$.

Par récurrence, posons $A^{(1)} \;=\; A$ et pour $i \;\in\; \{1,\ldots,n\}$, notons $a_i^{(1)}$ la i-ème colonne de la matrice $A^{(1)}$, c'est-à-dire $A^{(1)} = \left(a_1^{(1)},\ldots,a_n^{(1)}\right)$.

En appliquant la Proposition 1.3.2 au vecteur $a_1^{(1)} \in \mathbb{R}^n$: il existe $u_1 \in \mathbb{R}^n$ tel que $H_{u_1}\, a_1^{(1)} = \|a_1^{(1)}\|\, e_1$. Ainsi en notant $H_1 := H_{u_1}$, nous obtenons

$$A^{(2)} = H_1\, A^{(1)} = \begin{pmatrix} \|a_1^{(1)}\| & A_{1,2}^{(2)} \\ 0_{\mathbb{R}^{n-1}} & A_{2,2}^{(2)} \end{pmatrix}.$$

En poursuivant ce procédé et en s'inspirant de ce qui précède, nous mettons en place un algorithme, dit d'élimination de Householder. Supposons qu'au cours de l'étape $(k-1)$, nous ayons construit une matrice $A^{(k)}$ donnée par

$$A^{(k)} = \begin{pmatrix} A_{1,1}^{(k)} & A_{1,2}^{(k)} \\ 0 & A_{2,2}^{(k)} \end{pmatrix},$$

avec $A_{1,1}^{(k)} \in \mathcal{M}_{k-1,k-1}(\mathbb{R})$ une matrice triangulaire supérieure, $A_{1,2}^{(k)}$ une matrice rectangulaire réelle à $k-1$ lignes et $n-k+1$ colonnes et $A_{2,2}^{(k)}$ une matrice carrée réelle à $n-k+1$ lignes et colonnes.

La k-ème étape consiste donc à multiplier la matrice $A_{2,2}^{(k)}$ par une matrice de Householder pour faire apparaître des 0 sur la première colonne. Pour cela, nous notons $a_k^{(k)}$ le vecteur de $\mathbb{R}^{n-k+1}$ formé à partir des $(n-k+1)$ composantes de la première colonne de $A_{2,2}^{(k)} := \left(a_k^{(k)}, \ldots, a_n^{(k)}\right)$ et $e_1^{(k)} := (1, 0, \ldots, 0) \in \mathbb{R}^{n-k+1}$ le vecteur de la base canonique dont seule la première composante est non nulle.

Nous appliquons la Proposition 1.3.2 au vecteur $a_k^{(k)} \in \mathbb{R}^{n-k+1}$: il existe $u_k \in \mathbb{R}^{n-k+1}$ tel que $H_{u_k}\, a_k^{(k)} = \|a_k^{(k)}\|\, e_1^{(k)}$. Nous construisons alors la matrice $H_k \in \mathcal{M}_{n,n}(\mathbb{R})$ de réflection

$$H_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H_{u_k} \end{pmatrix}$$

d'où nous déduisons la nouvelle matrice $A^{(k+1)}$

$$A^{(k+1)} = H_k\, A^{(k)} = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H_{u_k} \end{pmatrix} \begin{pmatrix} A_{1,1}^{(k)} & A_{1,2}^{(k)} \\ 0 & A_{2,2}^{(k)} \end{pmatrix}.$$

En effectuant le produit par blocs, nous obtenons plus précisément

$$A^{(k+1)} = \begin{pmatrix} A_{1,1}^{(k)} & A_{1,2}^{(k)} \\ 0 & H_{u_k}\, A_{2,2}^{(k)} \end{pmatrix}.$$

Ainsi, la matrice $A^{(k+1)}$ s'écrit finalement sous la forme

$$A^{(k+1)} = \begin{pmatrix} A_{1,1}^{(k+1)} & A_{1,2}^{(k+1)} \\ 0 & A_{2,2}^{(k+1)} \end{pmatrix}$$

et $A_{1,1}^{(k+1)} \in \mathcal{M}_{k,k}(\mathbb{R})$ une matrice triangulaire supérieure, $A_{1,2}^{(k+1)} \in \mathcal{M}_{k,n-k}(\mathbb{R})$ et $A_{2,2}^{(k+1)} \in \mathcal{M}_{n-k,n-k}(\mathbb{R})$.

Ainsi, après n itérations, nous obtenons une matrice $A^{(n)}$ qui est triangulaire supérieure et telle que $H_{n-1} \dots H_1 A = A^{(n)}$. Puisque, d'après la Proposition 1.3.1 chaque matrice H_k est symétrique et vérifie $H_k^{-1} = H_k$, la matrice H_k est orthogonale. Ainsi, en posant $Q = H_1 \dots H_{n-1}$ et $R = A^{(n)}$, la matrice A s'écrit sous la forme $A = Q\,R$ avec Q une matrice orthogonale et R une matrice triangulaire supérieure. La démonstration de l'unicité de cette décomposition est facile et est laissée en exercice.

□

Remarque 1.3.2 *Une autre méthode pour démontrer qu'une matrice inversible A réelle admet une factorisation $Q\,R$, consiste à considérer la matrice $A^T A$ qui est symétrique et définie positive, elle admet donc une factorisation de Cholesky $A^T A = R^T R$. Il suffit donc de vérifier que $Q = (A^T)^{-1} R^T$ est orthogonale. Néanmoins, la méthode de Householder est constructive tandis qu'ici le calcul de Q n'est pas évident !*

1.4 Méthodes itératives

Les méthodes directes sont certes intéressantes puisqu'elles fournissent la solution exacte mais lorsque le système devient de grande taille, les calculs deviennent trop fastidieux. À titre d'exemple, Carl Friedrich Gauss s'était rendu compte que sa méthode n'était pas adaptée aux systèmes de plus de vingt équations qu'il devait résoudre en cartographie (triangulation de Hannovre). D'autre part, les calculs sur ordinateur ne fournissent qu'une solution approchée à cause des erreurs d'arrondi et nous pouvons rencontrer des problèmes d'instabilité (voir la partie Méthode de Gauss sans pivot). Il peut donc s'avérer judicieux de ne pas rechercher la solution exacte mais plutôt une solution approchée.

C'est pourquoi dans cette partie, nous mettons en place un type de méthodes numériques tout à fait différent. Il ne s'agit plus de résoudre exactement un système linéaire de grande taille mais plutôt de construire une suite vectorielle convergeant vers la solution du système linéaire (1.2). C'est ce que nous appelons une méthode itérative. Dans un premier temps, nous donnons les grands principes de ces méthodes : construction de la suite et critère de convergence. Ensuite, nous détaillons deux méthodes classiques : la méthode de Jacobi puis celle de Gauss-Seidel.

1.4.1 Méthodologie générale

Soient $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice inversible et un vecteur $b \in \mathbb{K}^n$, recherchons une approximation de $x \in \mathbb{K}^n$ solution de $A\,x = b$.

Pour cela, posons $A = M - N$, où M est une matrice inversible et surtout facile à inverser (par exemple une matrice diagonale, triangulaire ou orthogonale). Alors résoudre le système $A\,x = b$ sera équivalent à résoudre le système $M\,x = N\,x + b$ ou encore, dès que M est facile à inverser,

$x = M^{-1} N x + M^{-1} b =: F(x)$, où F est une fonction affine. À partir de cette dernière égalité, nous proposons la méthode itérative suivante : pour $x^{(0)}$ donné, et $k \geq 0$

$$x^{(k+1)} = F(x^{(k)}) = M^{-1} N x^{(k)} + M^{-1} b. \tag{1.6}$$

Si, par chance, la suite $(x^{(k)})_{k\in\mathbb{N}}$ converge, alors sa limite x^∞ doit satisfaire $x^\infty = F(x^\infty)$ car F est continue. Donc $x^\infty = M^{-1} N x^\infty + M^{-1} b$ soit $(M - N) x^\infty = b$, c'est-à-dire $A x^\infty = b$. Nous disons alors que la solution $x^\infty = A^{-1} b \in \mathbb{K}^n$ est un point fixe ou un point stationnaire de l'algorithme (1.6). Plusieurs questions découlent de cette construction :

- Pour quelle décomposition de la matrice A obtenons-nous la convergence de la méthode ? Bien sûr, la décomposition la plus simple serait $M = I_n$ et $N = I_n - A$ mais nous rencontrons généralement des problèmes de convergence pour cette méthode itérative (voir la méthode de Richardson de l'Exercice 1.4.7).
- Est-il possible d'améliorer la vitesse de convergence en fonction de la décomposition ? Nous voyons bien que le choix de $M = A$ et $N = 0$ permet la convergence de la méthode en une seule itération mais ce choix n'est en général pas intéressant puisque nous ne connaissons pas $M^{-1} = A^{-1}$.
- Comment déterminer l'arrêt des itérations ? En effet, nous ne pouvons pas calculer l'écart exact entre $x^{(k)}$ et la solution x puisque nous ne connaissons pas la solution.

La construction d'une méthode itérative revient à chercher des réponses à ces trois questions. Nous commençons par la notion de convergence de la méthode itérative. Ensuite, nous proposons plusieurs décompositions possibles (méthode de Jacobi, Gauss-Seidel) et tentons d'évaluer la vitesse de convergence des différentes méthodes.

Notion de convergence et vitesse de convergence d'un algorithme itératif

Définition 1.4.1 *Soit $A = M - N \in \mathcal{M}_{n,n}(\mathbb{K})$ avec M une matrice inversible. L'algorithme itératif (1.6) converge si pour tout $b \in \mathbb{K}^n$ et tout $x^{(0)} \in \mathbb{K}^n$, la suite $(x^{(k)})_{k\geq 0}$ converge vers la solution $x = A^{-1} b$ dans $\mathbb{K}^n$, c'est-à-dire*

$$\lim_{k\to\infty} \|x^{(k)} - x\| = 0.$$

Rappelons que toutes les normes sont équivalentes sur $\mathbb{K}^n$ et que la notion de convergence ne dépend pas du choix de la norme.

Notons bien qu'ici le critère de convergence porte essentiellement sur les propriétés de la décomposition de $A = M - N$, puisqu'aucune restriction n'est imposée sur le vecteur initial $x^{(0)} \in \mathbb{K}^n$ et la donnée $b \in \mathbb{K}^n$. Nous verrons au chapitre suivant que les choses se compliquent lorsque nous nous intéressons à des systèmes non linéaires, nous introduirons alors différentes notions de convergence.

Pour l'instant, essayons d'établir un critère sur les matrices M et N pour assurer la convergence de la méthode. Pour cela, introduisons l'erreur $e^{(k)}$ entre la solution approchée $x^{(k)}$ et la solution exacte x qui est donnée par $e^{(k+1)} = x^{(k+1)} - x$.

Rappelons qu'*a priori* nous ne connaissons pas la solution exacte x, ce qui signifie que dans la pratique nous ne pouvons pas calculer l'erreur $e^{(k)}$. Ici, notre objectif est plutôt de trouver un critère sur la décomposition assurant que cette erreur $e^{(k)}$ converge vers zéro lorsque k tend vers l'infini.

En utilisant l'algorithme itératif (1.6) et le fait que la solution x est une solution de $x = A^{-1} b = M^{-1} (N x - b)$, nous sommes en mesure de calculer l'erreur $e^{(k+1)}$ à l'étape $k + 1$ en fonction de l'erreur $e^{(k)}$ à l'étape k : $e^{(k+1)} = M^{-1} N (x^{(k)} - x) = M^{-1} N e^{(k)}$.

Posons alors $B = M^{-1} N$, ce qui donne $e^{(k+1)} = B\, e^{(k)} = B^{k+1}\, e^{(0)}$. D'après la Définition 1.4.1, l'algorithme itératif va converger lorsque pour n'importe quel vecteur $e^{(0)} \in \mathbb{K}^n$

$$\lim_{k\to\infty} B^k\, e^{(0)} = 0. \tag{1.7}$$

Le théorème suivant établit des critères sur la matrice B pour que la propriété (1.7) soit vérifiée.

Théorème 1.4.1 *Soit $B \in \mathcal{M}_{n,n}(\mathbb{K})$, alors les quatre propositions suivantes sont équivalentes :*

(i) Pour une norme matricielle subordonnée quelconque, nous avons $\lim_{k\to\infty} \|B^k\| = 0$.

(ii) Pour tout vecteur $v \in \mathbb{K}^n$, nous avons $\lim_{k\to\infty} B^k\, v = 0_{\mathbb{K}^n}$.

(iii) Le rayon spectral de B vérifie $\rho(B) < 1$.

(iv) Il existe une norme matricielle subordonnée (dont le choix dépend de B) telle que $\|B\| < 1$.

Démonstration : Nous allons montrer que $(i) \Rightarrow (ii) \ldots \Rightarrow (iv)$ et enfin $(iv) \Rightarrow (i)$.

Montrons d'abord que $(i) \Rightarrow (ii)$. Supposons que $\lim_{k\to\infty} \|B^k\| = 0$, pour $\|.\|$ une norme quelconque. Vu qu'en dimension finie toutes les normes sont équivalentes, ceci est en particulier vraie pour une norme subordonnée. Soit $u \in \mathbb{R}^n$, nous avons pour une norme subordonnée $0 \leq \|B^k\, u\| \leq \|B^k\|\, \|u\|$. Or, d'après (i), nous savons que $\|B^k\|$ converge vers zéro lorsque k tend vers l'infini et donc pour tout vecteur $u \in \mathbb{R}^n$

$$\lim_{k\to\infty} \|B^k\, u\| = 0,$$

ce qui signifie bien que $B^k\, u$ tend vers zéro lorsque k tend vers l'infini.

Supposons maintenant que (ii) est vraie et montrons (iii). Pour cela, nous considérons $\lambda \in \mathrm{Sp}(B)$ et w un vecteur propre associé à la valeur propre λ : $B\, w = \lambda\, w$, nous avons alors

$$B^k\, w = B^{k-1}\, (B\, w) = B^{k-1}\, (\lambda\, w) = \ldots = \lambda^k w.$$

Or, nous savons que $\lim_{k\to\infty} \|B^k\, w\| = 0$, donc

$$\lim_{k\to\infty} |\lambda^k|\, \|w\| = 0,$$

où $w \in \mathbb{R}^n$ ne dépend pas de k et est non nul puisque c'est un vecteur propre. Nous avons alors $\lim_{k\to\infty} |\lambda|^k = 0$, ce qui implique que $|\lambda| < 1$. Ainsi, pour tout $\lambda \in \mathrm{Sp}(B)$, $|\lambda| < 1$, c'est-à-dire $\rho(B) < 1$.

Montrons ensuite que $(iii) \Rightarrow (iv)$. Supposons que pour tout $\lambda \in \mathrm{Sp}(B)$, nous ayons $|\lambda| < 1$. En appliquant le théorème de Shur, il existe une matrice unitaire telle que

$$T = U^\star\, B\, U = \begin{pmatrix} \lambda_1 & t_{1,2} & \ldots & t_{1,n} \\ & \ddots & \ddots & \vdots \\ & & \lambda_{n-1} & t_{n-1,n} \\ 0 & & & \lambda_n \end{pmatrix}$$

et nous ayons $|\lambda_i| < 1$ pour tout $i \in \{1, \ldots, n\}$. En appliquant la Proposition 1.2.6, nous pouvons alors calculer $\|T\|_\infty$ ou $\|T\|_1$ qui est donnée par

$$\|T\|_1 = \max\left(|\lambda_1|, \ldots, |\lambda_n| + \sum_{i=1}^{n-1} |t_{i,n}|\right).$$

Nous ne pouvons pas conclure directement ! Introduisons alors pour $\delta > 0$, un petit paramètre qui reste à déterminer, la matrice diagonale $D = \mathrm{diag}(1, \delta, \ldots, \delta^{n-1})$ et $D^{-1} = \mathrm{diag}(1, \delta^{-1}, \ldots, \delta^{1-n})$. Nous vérifions alors

$$D^{-1}\,T\,D = \begin{pmatrix} \lambda_1 & \delta t_{1,2} & \ldots & \delta^{n-1} t_{1,n} \\ & \ddots & \ddots & \vdots \\ & & & \\ & & \lambda_{n-1} & \delta t_{n-1,n} \\ & & & \\ 0 & & & \lambda_n \end{pmatrix}.$$

Ainsi, pour δ assez petit, nous avons $\|D^{-1}TD\|_1 < 1$ ou alors

$$\|D^{-1}\,U^\star\,B\,U\,D\|_1 < 1.$$

Choisissons alors l'application $\|.\|_B$ qui pour $A \in \mathcal{M}_{n,n}(\mathbb{R})$ associe

$$\|A\|_B = \|D^{-1}\,U^\star\,A\,U\,D\|_1.$$

Nous vérifions aisément que $\|.\|_B$, dépend de B par l'intermédiaire de U et D et est une norme matricielle subordonnée à la norme vectorielle

$$v \mapsto \|v\|_B = \|D^{-1}\,U^\star\,v\|_1.$$

Cette norme matricielle subordonnée est telle que par construction $\|B\|_B < 1$. Nous avons donc construit une norme matricielle subordonnée vérifiant $\|B\|_B < 1$, ce qui démontre (iv).

Finalement, montrons que $(iv) \Rightarrow (i)$. Supposons que pour une norme subordonnée donnée $\|.\|_\alpha$, $\|B\|_\alpha < 1$. En utilisant le fait que toutes les normes sont équivalentes, pour une norme quelconque $\|.\|$, il existe $C_1 > 0$ telle que

$$0 \leq C_1\,\|B^k\| \;\leq\; \|B^k\|_\alpha \;\leq\; \|B\|_\alpha^k \to 0.$$

Ce qui montre que $\|B^k\|$ converge vers zéro lorsque k tend vers l'infini. □

Ce théorème donne dans le même temps une indication sur la vitesse de convergence. En effet, puisque $\|e^{(k)}\| \;\leq\; \|B\|^k\,\|e^{(0)}\|$ pour tout $k \geq 0$, la vitesse de convergence de l'algorithme dépend directement du nombre $\|B\| < 1$ et nous chercherons à rendre cette norme la plus petite possible, ce qui permettra d'améliorer la vitesse de convergence.

Test d'arrêt et nombre d'itérations Nous venons de définir des critères de convergence mais dans la pratique il faudra stopper le processus lorsque nous estimerons que la solution numérique est suffisamment proche de la solution exacte. C'est pourquoi nous définissons un test d'arrêt et utilisons pour cela le vecteur résidu $r^{(k)} = b - A\,x^{(k)}$. En effet, le vecteur x est solution du problème $A\,x = b$

pour lequel le résidu $r = b - A\,x$ est nul. Il est donc clair que nous serons d'autant plus proche de la solution que le résidu sera petit. Ainsi, pour une précision donnée ε, nous poursuivons les itérations jusqu'à ce que le résidu $r^{(k)}$ vérifie

$$\frac{\|r^{(k)}\|}{\|b\|} = \frac{\|b - A\,x^{(k)}\|}{\|b\|} \leq \varepsilon.$$

Remarquons aussi que pour une méthode itérative dont nous connaissons pour une norme donnée la valeur $\|B\| < 1$, il est alors possible de calculer le nombre d'itérations maximal en fonction de l'erreur souhaitée. En effet, comme nous l'avons déjà démontré $\|e^{(k)}\| \leq \|B\|^k \, \|e^{(0)}\|$.

D'autre part, nous savons aussi que

$$\|e^{(0)}\| \leq \|x^{(0)} - x^{(1)}\| + \|e^{(1)}\| \leq \|x^{(0)} - x^{(1)}\| + \|B\| \, \|e^{(0)}\|$$

et donc puisque $\|B\| < 1$, nous obtenons

$$\|e^{(0)}\| \leq \frac{1}{1 - \|B\|} \, \|x^{(0)} - x^{(1)}\|,$$

d'où en regroupant les deux résultats

$$\|e^{(k)}\| \leq \frac{\|B\|^k}{1 - \|B\|} \, \|x^{(1)} - x^{(0)}\|.$$

Ainsi, si nous souhaitons que l'erreur $\|e^{(k)}\|$ soit inférieure à 10^{-N}, il suffit de calculer un nombre d'itérations $k_0 \in \mathbb{N}^\star$ tel que

$$\frac{\|B\|^{k_0}}{1 - \|B\|} \, \|x^{(1)} - x^{(0)}\| \leq 10^{-N},$$

c'est-à-dire

$$k_0 \geq \frac{\log\left(\frac{1-\|B\|}{\|x^{(1)} - x^{(0)}\|}\right) - N\log(10)}{\log(\|B\|)},$$

avec $\log(\|B\|) < 0$ puisque $\|B\| < 1$.

Intéressons-nous maintenant aux méthodes itératives les plus utilisées dans la pratique : les méthodes de Jacobi et de Gauss-Seidel.

1.4.2 Méthode de Jacobi

Pour construire l'algorithme de Jacobi[5], nous choisissons la décomposition suivante : $A = D - E - F$, où la matrice D est formée par la diagonale de A, $-E$ représente la partie sous-diagonale de A tandis que $-F$ désigne la partie sur-diagonale de A.

Ainsi, en supposant que la diagonale de A ne contient pas d'éléments nuls, nous reprenons la méthode exposée précédemment en posant $M = D$ (qui est supposée inversible) et $N = E + F$.

[5]En référence à Charles Gustave Jacob Jacobi, mathématicien allemand (1804-1851). Ses principales contributions furent en analyse et calcul différentiel. Une autre contribution importante est la théorie de Hamilton-Jacobi en mécanique newtonienne. Il s'est également illustré en algèbre (fonction thêta de Jacobi et méthode de Jacobi pour la résolution approchée de systèmes linéaires) ainsi qu'en théorie des nombres.

La suite $(x^{(k)})_{k\geq 0}$ est alors donnée par :

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

Nous pouvons aussi écrire la méthode de Jacobi sous la forme

$$\begin{cases} x^{(0)} \in \mathbb{K}^n, \\ a_{i,i}x_i^{(k+1)} = -\sum_{j<i} a_{i,j}\,x_j^{(k)} - \sum_{j>i} a_{i,j}x_j^{(k)} + b_i,\ 1 \leq i \leq n,\ k \geq 0. \end{cases} \tag{1.8}$$

L'algorithme correspondant peut être décrit sous une forme compacte comme suit. Un petit paramètre $\varepsilon > 0$ étant fixé et $x_0 \in \mathbb{K}^n$ donné,

Algorithme 4. Méthode de Jacobi

Poser $x^{(0)} = x_0, \varepsilon^{(0)} = 2\,\varepsilon$ et $k = 0$.

Tant que $\varepsilon^{(k)} \geq \varepsilon$

- calculer

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

- calculer un résidu qui doit être proche de zéro

lorsque $x^{(k+1)}$ approche la solution $x = A^{-1}\,b$

$$\varepsilon^{(k+1)} = \|A\,x^{(k+1)} - b\|,$$

- itérer $k \leftarrow k+1$.

Fin de tant que.

Cette méthode n'est pas toujours bien définie. En effet, il suffit qu'au moins un élément diagonal soit nul pour que l'algorithme ne soit plus valide. Néanmoins, lorsque la matrice A est une matrice définie positive, ses coefficients diagonaux sont strictement positifs et nous pouvons donc appliquer cet algorithme. Aussi lorsque A est à diagonale strictement dominante, la méthode est correctement définie et nous sommes même en mesure de présenter un résultat de convergence.

Théorème 1.4.2 (Convergence de la méthode de Jacobi) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice à diagonale strictement dominante, c'est-à-dire telle que*

$$|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|, \quad \forall\ i \in \{1, \ldots, n\}.$$

Alors pour tout $x^{(0)} \in \mathbb{K}^n$, la suite $(x^{(k)})_{k \in \mathbb{N}}$, donnée par la méthode de Jacobi (1.8), est bien définie et converge vers la solution x du système $A\,x = b$.

Démonstration : D'une part puisque A est à diagonale strictement dominante, tous les coefficients diagonaux de A sont strictement positifs et $M = D$ est bien inversible.

D'autre part, d'après le Théorème 1.4.1, il suffit de prouver qu'il existe une norme matricielle subordonnée $\|.\|_*$ telle que la matrice $B = D^{-1}(E + F)$ vérifie $\|B\|_* < 1$.

Considérons par exemple la norme $\|B\|_\infty$, nous savons d'après la Proposition 1.2.6 que cette norme peut être calculée exactement à partir des coefficients de B

$$\|B\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |b_{i,j}|.$$

Or, par construction de B, nous avons aussi

$$b_{i,j} = \begin{cases} 0, & \text{si } i = j, \\ -\dfrac{a_{i,j}}{a_{i,i}}, & \text{si } i \neq j. \end{cases}$$

Puisque A est à diagonale strictement dominante, nous vérifions facilement que pour tout $i \in \{1, \ldots, n\}$

$$\sum_{j=1}^{n} |b_{i,j}| = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{i,j}|}{|a_{i,i}|} < 1$$

et donc $\|B\|_\infty < 1$. Ainsi, par application du Théorème 1.4.1, la méthode de Jacobi est convergente. □

L'inconvénient de cette méthode est que le critère de convergence est assez restrictif pour pouvoir traiter des systèmes généraux. Citons le cas d'une matrice symétrique définie positive pour laquelle la méthode de Jacobi ne converge pas.

Exemple 1.4.1 *Soit $A \in \mathcal{M}_{3,3}(\mathbb{R})$ donnée par*

$$A = \begin{pmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{pmatrix},$$

où a est un réel. Nous voulons approcher la solution du système $A\,x = b$ par la méthode de Jacobi. Puisque les éléments diagonaux sont non nuls, la suite $(x^{(k)})_{k \in \mathbb{N}}$ est bien définie mais ne converge pas vers la solution.

D'une part, nous vérifions que A est symétrique définie positive si et seulement si $a \in]-1/2, 1[$. En effet, en calculant les valeurs propres de A, nous obtenons que le polynôme caractéristique est donné par

$$P(\lambda) = (1-\lambda)^3 - 3\,a^2\,(1-\lambda) + 2a^3 = -(\lambda - (1-a))^2\,(\lambda - (1+2\,a))$$

et les valeurs propres sont $\lambda_1 = \lambda_2 = 1 - a$ et $\lambda_3 = 1 + 2\,a$.

Lorsque $-1/2 < a < 1$, les valeurs propres sont strictement positives, ce qui assure que A est définie positive. En revanche, la méthode Jacobi converge seulement lorsque $a \in]-1/2, 1/2[$. En effet, la suite est donnée par $x^{(k+1)} = D^{-1}(D-A)\,x^{(k)} + D^{-1}\,b$, avec ici $D = I_3$. Les valeurs propres de $D - A$ sont de la forme $\mu = 1 - \lambda$ où λ est une valeur propre de A et donc $\mu_1 = \mu_2 = a$ et $\mu_3 = -2\,a$. Nous en concluons que la méthode de Jacobi converge pour $-1/2 < a < 1/2$ puisqu'il est nécessaire que $\rho(D^{-1}(D-A)) < 1$ et ici $D^{-1}(D-A) = I_3 - A$.

Cet exemple montre bien que la méthode de Jacobi n'est pas toujours la plus adaptée pour traiter des applications concrètes, où les systèmes linéaires à résoudre font intervenir des matrices définies positives. Une alternative possible est de recourir à la méthode de Gauss-Seidel.

1.4.3 Méthode de Gauss-Seidel

Décomposons également la matrice A de la façon suivante : $A = D - E - F$, avec le même choix pour D, E et F que celui de la méthode de Jacobi. Cependant, pour la méthode de Gauss-Seidel[6] nous choisissons $M = D - E$ et $N = F$

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

Dans ce cas, $D - E$ est une matrice triangulaire inférieure, elle est donc facilement inversible en utilisant un algorithme de descente (nous calculons d'abord x_1, puis $x_2, \ldots$). Nous avons cette fois-ci

$$\begin{cases} x^{(0)} \in \mathbb{K}^n, \\ a_{i,i}\,x_i^{(k+1)} = -\sum_{j<i} a_{i,j}\,x_j^{(k+1)} - \sum_{j>i} a_{i,j}\,x_j^{(k)} + b_i,\ i = 1, \ldots, n, \quad k \geq 0. \end{cases}$$

Notons bien qu'ici le terme de droite dépend de $x^{(k+1)}$, il faut donc utiliser une méthode de descente à chaque itération.

L'algorithme correspondant peut être décrit sous une forme compacte comme suit. Un petit paramètre $\varepsilon > 0$ étant fixé et $x_0 \in \mathbb{K}^n$ donné,

[6] En référence à Carl Friedrich Gauss, qui mit au point cet algorithme pour le calcul approché de la solution d'un système linéaire. Cet algorithme fut ensuite redécouvert en 1847 par le mathématicien allemand Philipp Ludwig von Seidel (1821-1896). L'algorithme de Gauss-Seidel est toujours utilisé de nos jours pour la résolution numérique de très grands systèmes.

Algorithme 5. Méthode de Gauss-Seidel

Poser $x^{(0)} = x_0, \varepsilon^{(0)} = 2\,\varepsilon$ et $k = 0$.

Tant que $\varepsilon^{(k)} \geq \varepsilon$

- pour $k \geq 0$, résoudre par un algorithme de descente

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

- calculer le résidu qui doit être proche de zéro

lorsque $x^{(k+1)}$ approche la solution :

$$\varepsilon^{(k+1)} = \| A\, x^{(k+1)} - b \|,$$

- itérer $k \leftarrow k + 1$.

Fin de tant que.

Nous pouvons démontrer que la méthode de Gauss-Seidel converge dans le cas de matrices symétriques définies positives. Avant cela, énonçons un lemme qui se révèle souvent utile pour démontrer la convergence d'une méthode itérative.

Lemme 1.4.1 *Soit $A \in \mathcal{M}_{n,n}(\mathbb{R})$ une matrice symétrique définie positive, décomposée sous la forme $A = M - N$ avec M inversible. Si $M^T + N$ est symétrique définie positive, alors $\rho(M^{-1}\, N) < 1$.*

Démonstration : Soit $A = M - N$ telle que $M^T + N$ soit symétrique définie positive. D'après le Théorème 1.4.1, la condition $\rho(M^{-1}\, N) < 1$ est équivalente à démontrer qu'il existe une norme matricielle subordonnée $\|.\|_\star$ telle que $\|M^{-1}\, N\|_\star < 1$.

Puisque la matrice A est définie positive, nous pouvons considérer la norme matricielle subordonnée à la norme vectorielle donnée par $\|x\|_\star = \sqrt{x^T\, A\, x}$. Il vient alors

$$\begin{aligned} \|M^{-1}\, N\|_\star^2 &= \sup_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{\|M^{-1}\, N\, x\|_\star^2}{\|x\|_\star^2}, \\ &= \sup_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{(M^{-1}\, N\, x)^T\, A\, M^{-1}\, N\, x}{x^T\, A\, x}. \end{aligned}$$

Or, en écrivant $N = M - A$ puis $y = M^{-1}\, A\, x$, nous avons pour $x \in \mathbb{R}^n$ non nul

$$\begin{aligned} (M^{-1}\, N\, x)^T\, A\, M^{-1}\, N\, x &= ((I_n - M^{-1}\, A)\, x)^T\, A\, (I_n - M^{-1}\, A)\, x\,, \\ &= x^T\, A\, x - y^T\, A\, x - x^T\, A\, y + y^T\, A\, y \end{aligned}$$

et puisque A est symétrique et $\|x\|_\star^2 = x^T\,A\,x$,

$$\frac{\|M^{-1}\,N\,x\|_\star^2}{x^T\,A\,x} = \frac{(M^{-1}\,N\,x)^T\,A\,M^{-1}\,N\,x}{x^T\,A\,x} \;=\; \frac{x^T\,A\,x \,-\, 2\,y^T\,A\,x \,+\, y^T\,A\,y}{x^T\,A\,x}.$$

Ce dernier terme est positif et donc pour démontrer que $\|M^{-1}\,N\|_\star^2 < 1$, il suffit de prouver que

$$\sup_{\substack{x\in\mathbb{R}^n\setminus\{0\}\\ Ax=My}} \frac{2\,y^T\,A\,x - y^T\,A\,y}{x^T\,A\,x} > 0.$$

Puisque $A\,x = M\,y$, nous vérifions que

$$2\,y^T\,A\,x - y^T\,A\,y \;=\; 2\,y^T\,M\,y \;-\; y^T\,A\,y.$$

Or, par symétrie du produit euclidien $y^T\,M\,y \;=\; y^T\,M^T\,y$, nous obtenons

$$2\,y^T\,A\,x \;-\; y^T\,A\,y \;=\; y^T\,M^T\,y \,+\, y^T\,(M-A)\,y \;=\; y^T\,(M^T+N)\,y.$$

Finalement, comme $M^T + N$ est définie positive, ce dernier terme est strictement positif et puisque x est non nul, il vérifie également $x^T\,A\,x > 0$. Ainsi,

$$\sup_{\substack{x\in\mathbb{R}^n\setminus\{0\}\\ Ax=My}} \frac{2\,y^T\,A\,x - y^T\,A\,y}{x^T\,A\,x} = \sup_{\substack{x\in\mathbb{R}^n\setminus\{0\}\\ Ax=My}} \frac{y^T\,(M^T+N)\,y}{x^T\,A\,x} > 0,$$

ce qui montre que $\|M^{-1}\,N\|_\star < 1$ ou encore $\rho(M^{-1}\,N) < 1$. □

Nous démontrons alors le résultat suivant.

Théorème 1.4.3 (Convergence de la méthode de Gauss-Seidel) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{R})$ une matrice symétrique définie positive. Alors pour tout $x^{(0)}$ la méthode de Gauss-Seidel est bien définie et converge vers la solution x du système $A\,x \;=\; b$.*

Démonstration : Vérifions d'abord que la méthode de Gauss-Seidel est bien définie. Posons $A = M - N$, avec $M = D - E$ et $N = F$, où la matrice D représente la diagonale de A, la matrice $-E$ est la partie inférieure de A et la matrice $-F$ la partie supérieure.

Pour que la méthode de Gauss-Seidel soit bien définie, il suffit de vérifier que M est inversible. Puisque M est une matrice triangulaire, nous avons

$$\det(M) \;=\; \det(D) \;=\; \prod_{i=1}^{n} a_{i,i}$$

et puisque A est définie positive tous les termes diagonaux sont positifs, nous avons alors $\det(M) > 0$.

Maintenant, pour prouver que la méthode de Gauss-Seidel converge, nous appliquons le Lemme 1.4.1, il suffit donc de vérifier que la matrice $M^T + N$ est symétrique définie positive $M^T + N \;=\; (D-E)^T + F = \; D - E^T + F$ et puisque A est symétrique $E^T \;=\; F$ et donc $M^T + N \;=\; D$, qui est bien symétrique définie positive. Par application du Lemme 1.4.1, nous obtenons $\rho(M^{-1}\,N) < 1$, ce qui implique que la méthode de Gauss-Seidel est convergente. □