Université de Liège
Faculté des Sciences Appliquées

Introduction à l'Analyse Numérique

Edition 2013

Professeur Q. Louveaux
Département d'Électricité, Électronique et Informatique
Institut Montefiore

Table des matières

Chapitre 1

Interpolations et Régressions

Dans le cours d'introduction aux méthodes numériques, nous avons vu quelques méthodes numériques courantes et avons insisté sur l'analyse du comportement de celles-ci. Nous allons poursuivre l'approche dans ce cours-ci. Dans ce chapitre, nous commençons par un rappel des outils principaux d'analyse des méthodes numériques, à savoir l'approximation d'une fonction inconnue par un polynôme et le développement de Taylor. Outre le rappel de l'interpolation polynomiale de Lagrange, nous abordons également la question de l'approximation d'une fonction inconnue sous un angle différent, celui de la régression. Le principe de la régression est assez similaire à celui de l'interpolation. La différence majeure réside dans le fait qu'on va cette fois chercher à approximer une fonction sans pour autant imposer que l'approximation passe exactement par les points que l'on connaît de la fonction.

1.1 Développement de Taylor

Rappelons brièvement l'expansion d'une fonction en série de Taylor.

Théorème 1.1 *Soit f, une fonction possédant ses $(n+1)$ premières dérivées continues sur un intervalle fermé $[a, b]$, alors pour chaque $c, x \in [a, b]$, f peut s'écrire comme*

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(c)}{k!}(x - c)^k + E_{n+1}$$

où le terme d'erreur E_{n+1} peut s'écrire sous la forme

$$E_{n+1} = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-c)^{n+1},$$

et ξ est un point situé entre c et x.

On dit que le terme d'erreur est *d'ordre* $n+1$. Il est parfois utile d'écrire cela de manière plus compacte sous la forme $E_{n+1} = \mathcal{O}(h^{n+1})$, où h représente $x - c$. La notation $\mathcal{O}(x^p)$ permet de décrire le fait que la fonction tend au moins aussi vite vers 0 que x^p lorsque x tend vers 0.

Définition 1.1 *Soit $f : \mathbb{R} \mapsto \mathbb{R}$. On dit que $f = \mathcal{O}(x^p)$ au voisinage de 0 si il existe $C \in \mathbb{R}_+$ et $x_0 \in \mathbb{R}_+$ tels que*

$$|f(x)| \leq C|x^p| \quad \textit{pour tout } -x_0 \leq x \leq x_0.$$

Cette définition sera tres souvent utilisée. En effet, le degré minimal d'un polynôme donne une idée très précise de la vitesse de convergence vers 0. Si on approxime une valeur V en l'approchant itérativement par une fonction $f(h)$ quand h tend vers 0, une évaluation de $f(h) - V$ en notation $\mathcal{O}$ donne une mesure de la vitesse de convergence du processus itératif. Plus le degré de convergence est élevé, plus la convergence se fait rapidement. Le théorème de Taylor est un outil extrêmement puissant pour pouvoir moduler l'ordre de précision que l'on souhaite d'une fonction.

1.2 Interpolation polynomiale

Le développement de Taylor donne une approximation d'une fonction f à partir de la connaissance de f en un point ainsi que de plusieurs de ses dérivées en ce même point. L'interpolation polynomiale est conceptuellement différente puisque on ne va pas travailler avec les dérivées de la fonction mais sa valeur en plusieurs points. Cela peut être très important lorsque l'on n'a pas accès à ces dérivées.

Soit donc $u : \mathbb{R} \mapsto \mathbb{R}$ une fonction inconnue, mais donnée en n points $x_1, \ldots, x_n$. On recherche un polynôme de degré $n-1$,

$$P(x) = \sum_{i=0}^{n-1} a_i x^i \tag{1.1}$$

qui satisfait $P(x_i) = u(x_i)$ pour tout $i = 1, \ldots, n$. Premièrement, il est utile de remarquer que ce problème a une solution unique si tous les x_i sont différents. Le théorème suivant formalise le cadre de l'interpolation polynomiale.

Théorème 1.2 *Soit un ensemble de n paires $(x_i, u(x_i))$. Si $x_i \neq x_j$ pour tout $i \neq j$, alors il existe un et un seul polynôme $P(x)$ de degré au plus $n-1$ qui satisfait $P(x_i) = u(x_i)$ pour $i = 1, \ldots, n$.*

La formule de Lagrange permet de calculer directement le polynôme d'interpolation. Pour la dériver, définissons d'abord la fonction

$$\begin{aligned} l_i(x) &= \frac{(x - x_1)(x - x_2) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_1)(x_i - x_2) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} \\ &= \frac{\prod_{k \neq i}(x - x_k)}{\prod_{k \neq i}(x_i - x_k)}. \end{aligned}$$

Premièrement remarquons que $l_i(x)$ est un polynôme de degré $n-1$. Il suffit de remarquer que le dénominateur n'est, en fait, rien d'autre qu'un nombre réel non nul. Ensuite, nous voyons que l_i satisfait

$$\begin{aligned} l_i(x_i) &= 1 \\ l_i(x_k) &= 0 \quad \text{pour tout } k \neq i. \end{aligned}$$

Il s'ensuit que le polynôme

$$P(x) = \sum_{i=1}^{n} u(x_i) l_i(x)$$

est la solution unique de notre problème. C'est ce qu'on appelle le *polynôme d'interpolation de Lagrange*. La formule est aisée à dériver mais requiert néanmoins pas mal de calculs. Lorsque l'on voudra obtenir une approximation polynomiale en réduisant le nombre de calculs, on utilisera la formule de Newton (vue dans le cours de méthodes numériques). Cependant le fait que la formule de Lagrange soit une formule "fermée" est très utile lorsque l'on veut l'appliquer de manière théorique.

Exemple 1.1 Considérons les points $(0,4),(2,0),(3,1)$ et tentons de faire passer un polynôme quadratique par ces trois points. Nous définissons successivement les trois polynômes

$$\begin{aligned} l_1(x) &= \frac{(x-2)(x-3)}{(-2)(-3)} = \frac{1}{6}(x^2-5x+6) \\ l_2(x) &= \frac{(x-0)(x-3)}{2(-1)} = -\frac{1}{2}(x^2-3x) \\ l_3(x) &= \frac{(x-0)(x-2)}{(3-0)(3-2)} = \frac{1}{3}(x^2-2x). \end{aligned}$$

Nous pouvons vérifier par exemple que $l_1(0)=1, l_1(2)=0, l_1(3)=0$. Il nous est maintenant possible de construire un polynôme quadratique passant par les trois points initiaux en écrivant

$$\begin{aligned} P(x) &= \frac{4}{6}(x^2-5x+6)+\frac{1}{3}(x^2-2x) \\ &= x^2-4x+4 \\ &= (x-2)^2. \end{aligned}$$

Nous pouvons vérifier que $P(x)$ passe bien par les trois points annoncés initialement. ■

Il est important de savoir quelle est l'erreur que l'on fait sur la fonction $u(x)$ lorsque l'on utilise $P(x)$ à sa place. En d'autres termes, on s'intéresse à la fonction $e(x)=u(x)-P(x)$. Le théorème suivant indique l'erreur réalisée.

Théorème 1.3 *Soit $u:\mathbb{R}\mapsto\mathbb{R}$ et $P(x)$ un polynôme interpolant les points $(x_1,u(x_1)),\ldots,(x_n,u(x_n))$. Si on suppose que $x_i\in[a,b]$ pour tout i, et que $u(x)$ est n fois continûment dérivable sur $[a,b]$, alors pour tout $x\in[a,b]$, il existe $\xi\in[a,b]$ tel que*

$$e(x)=u(x)-P(x)=\frac{u^{(n)}(\xi)}{n!}(x-x_1)\cdots(x-x_n). \tag{1.2}$$

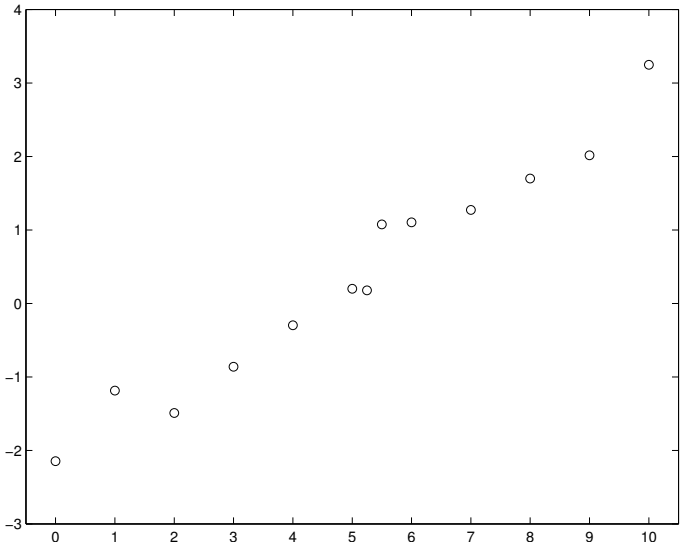

FIGURE 1.1 – Un nuage de points $(x_i, u(x_i))$

1.3 Approximation

Pour l'interpolation , nous avons contraint la fonction recherchée à *passer* par les points $(x_1, u(x_1)), \ldots, (x_n, u(x_n))$. Dans certaines situations, c'est en effet crucial. Mais quelle serait notre attitude si nous savions que certaines données sont entâchées d'erreurs ? C'est souvent le cas quand, par exemple, on recherche une fonction qui prédit au mieux un phénomène pour lequel on dispose de *mesures expérimentales*. Les mesures expérimentales étant, par essence, imprécises, il y aura lieu d'essayer de *lisser* les erreurs faites lors des mesures. C'est le sujet de cette section.

1.3.1 Régression linéaire

Considérons le nuage de points donnés dans la Figure 1.1. A l'oeil nu, il semble que les points suivent une relation *plus ou moins* linéaire. L'interpolation polynomiale et l'interpolation par spline cubique sont représentées sur la Figure 1.2. Il apparaît clairement que la relation qu'elles donnent sont peu satisfaisantes du point de vue de la prédiction. Les courbes dépendent en effet trop des mesures particulières considérées ici. Le problème de la régression

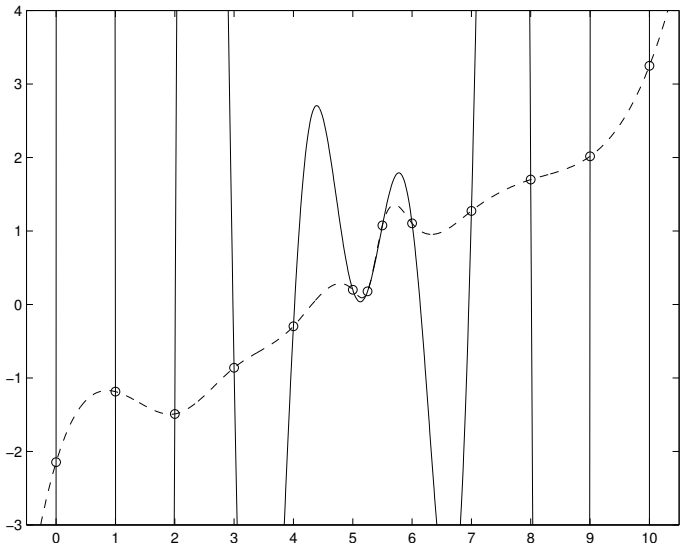

FIGURE 1.2 – L'interpolation polynomiale (en trait plein) et la spline cubique (en pointillés) interpolant le nuage

linéaire va être de trouver une relation linéaire qui décrit au mieux le nuage de points. Plus précisément, on part des points $(x_1, u(x_1)), \ldots, (x_n, u(x_n))$, et on cherche les coefficients a et b d'une droite $y = ax + b$ tels que l'on a $ax_i + b \approx u(x_i)$ pour tout i. Comme nous l'avons vu pour l'interpolation polynomiale, si on a deux points, on peut trouver a et b tels que l'on a à chaque fois une égalité. Mais en général, si on a plus de deux points, pour chaque abscisse x_i, nous introduisons une erreur $e_i = ax_i + b - u(x_i)$. Il faut alors minimiser cette erreur selon un critère précis. Un critère particulièrement populaire (parce que relativement aisé à mettre en oeuvre) consiste à trouver a et b qui minimisent la somme des carrés des erreurs, à savoir $E(a, b) := \sum_{i=1}^{n} e_i^2$. Il s'agit d'un bon critère car à la fois les erreurs positives et négatives sont comptées positivement. De plus, cela pénalise les fortes erreurs. On peut supposer que la courbe qui en résultera passera au mieux *entre* les points. Remarquons que ce critère peut ne pas être le bon choix si on sait que certaines mesures peuvent être entâchées de très grosses erreurs dont il faudrait, idéalement, ne pas tenir compte. Tâchons à présent,

de trouver les coefficients a et b. On cherche à minimiser la fonction

$$E(a,b) = \sum_{i=1}^{n} (ax_i + b - u(x_i))^2.$$

La fonction est certainement deux fois continûment dérivable. Une condition nécessaire pour trouver un minimum est donc d'annuler le Jacobien de E, à savoir

$$\frac{\partial E(a,b)}{\partial a} = 0, \qquad \frac{\partial E(a,b)}{\partial b} = 0.$$

On obtient donc

$$\sum_{i=1}^{n} 2x_i(ax_i + b - u(x_i)) = 0$$
$$\sum_{i=1}^{n} 2(ax_i + b - u(x_i)) = 0.$$

Remarquons que, comme a et b sont nos variables, le système est en réalité *linéaire*. On peut également le réécrire comme

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i u(x_i) \\ \sum_{i=1}^{n} u(x_i). \end{pmatrix}.$$

Ces équations sont appelées les *équations normales*. Il est possible de montrer que la solution de ces équations fournit bien la solution qui minimise la fonction $E(a,b)$. Appliqué à l'exemple du début de la section, on obtient la droite représentée sur la Figure 1.3.

1.3.2 Régression non linéaire

La méthode que l'on a vue dans le paragraphe précédent peut être appliquée à n'importe quel ensemble de fonctions de base, du moment que celles-ci sont linéairement indépendantes. On peut alors appliquer les mêmes approches, à savoir, annuler les dérivées partielles de la fonction erreur pour

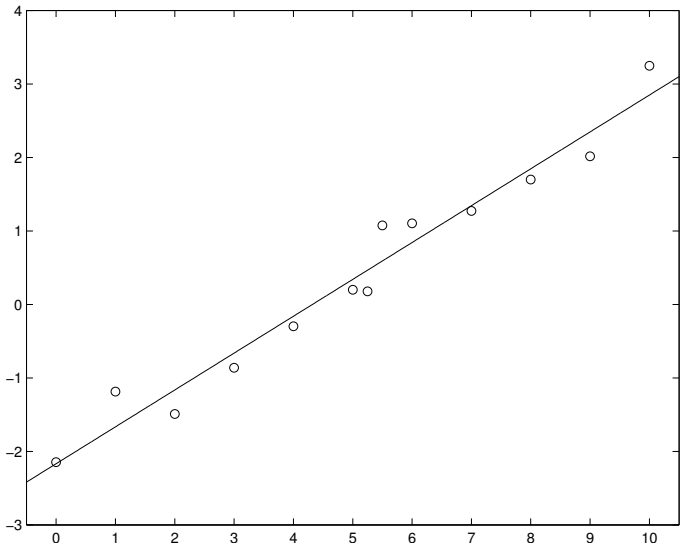

FIGURE 1.3 – La régression linéaire approximant le nuage de points

chaque paramètre. Imaginons que nous souhaitions trouver les meilleurs coefficients $a_1, \ldots, a_m$ tels que la fonction

$$\phi(x) = \sum_{j=1}^{m} a_j \phi_j(x)$$

approxime au mieux les points $(x_i, u(x_i))$. Ici on suppose que les fonctions $\phi_1(x), \ldots, \phi_m(x)$ sont linéairement indépendantes. On définit la fonction erreur

$$E(a_1, \ldots, a_m) := \sum_{i=1}^{n} (\sum_{j=1}^{m} a_j \phi_j(x_i) - u(x_i))^2.$$

Comme on cherche à minimiser E, les système d'équations normales est donné par le calcul de

$$\frac{\partial E(a_1, \ldots, a_m)}{\partial a_1} = 0, \quad , \ldots , \quad , \frac{\partial E(a_1, \ldots, a_m)}{\partial a_m} = 0.$$

En calculant les dérivées partielles, on trouve

$$\frac{\partial E(a_1,\ldots,a_m)}{\partial a_1} = 2\sum_{i=1}^{n}\phi_1(x_i)(\sum_{j=1}^{m}a_j\phi_j(x_i) - u(x_i))$$

$$\vdots$$

$$\frac{\partial E(a_1,\ldots,a_m)}{\partial a_m} = 2\sum_{i=1}^{n}\phi_m(x_i)(\sum_{j=1}^{m}a_j\phi_j(x_i) - u(x_i)).$$

On obtient dès lors le système

$$\sum_{j=1}^{m}(\sum_{i=1}^{n}\phi_1(x_i)\phi_j(x_i))a_j = \sum_{i=1}^{n}\phi_1(x_i)u(x_i) \tag{1.3}$$

$$\vdots$$

$$\sum_{j=1}^{m}(\sum_{i=1}^{n}\phi_m(x_i)\phi_j(x_i))a_j = \sum_{i=1}^{n}\phi_m(x_i)u(x_i) \tag{1.4}$$

qu'on appelle le *système d'équations normales*.

1.3.3 Choix de la base de fonctions

Un choix courant de fonctions de base pour effectuer une régression est de considérer des polynômes. On parlera ainsi de régression quadratique lorsque l'on cherche le meilleur polynôme quadratique approximant un nuage de points. Lorsque l'on fait tendre le degré du polynôme qui approxime un nuage de points vers l'infini, on obtient à un moment donné l'interpolation exacte du nuage de points. Nous avons vu cependant qu'il est souvent dangereux d'utiliser des polynômes de degré trop élevé car cela mène à des oscillations non désirées dans le comportement de la fonction recherchée. On préférera en général approximer un phénomène par des polynômes de petit degré.

Bien que l'approximation par des polynômes de haut degré soit déconseillée, nous allons nous intéresser dans cette section à la *résolution numérique* des équations normales dans le cas d'un grand nombre de fonctions. Il apparaît en effet que si l'on ne choisit pas la base de fonctions avec précaution, le système des équations normales peut être mal conditionné et fort sujet aux problèmes numériques. Imaginons que l'on cherche le polynôme de degré 5

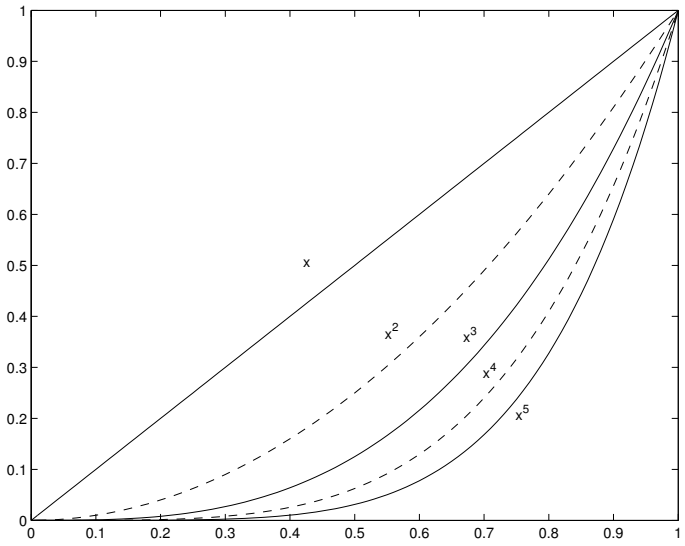

FIGURE 1.4 – Les 5 monômes de base sont de plus en plus similaires

qui approxime au mieux un nuage de points. Un choix naturel est d'écrire le polynôme que l'on cherche comme

$$\phi(x) = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$$

Cela consiste, de manière équivalente, à considérer que les 6 fonctions de base sont $x^5, x^4, \ldots, x, 1$. Il s'agit en général d'un mauvais choix du point de vue numérique. En effet, si on considère les polynômes sur l'intervalle $[0, 1]$, on peut voir sur la Figure 1.4 que les polynômes choisis sont beaucoup trop similaires. On aura en général un système linéaire dont le déterminant est trop proche de 0. Nous verrons dans le chapitre sur la résolution de systèmes linéaires que ces systèmes sont mal conditionnés et difficiles à résoudre. Il sera, par contre, plus judicieux d'utiliser une base de polynômes *orthogonaux*. Il existe toute une série de familles de polynômes orthogonaux qui peuvent être un bon choix pour calculer les coefficients. Nous avons déjà vu dans le cours de méthodes numériques une famille de polynômes orthogonaux. Il s'agit des *polynômes de Chebyshev*. Sur la Figure 1.5, les cinq premiers polynômes de Chebyshev sont représentés. On voit que les polynômes sont moins similaires que les monômes qu'il aurait été naturel de choisir.

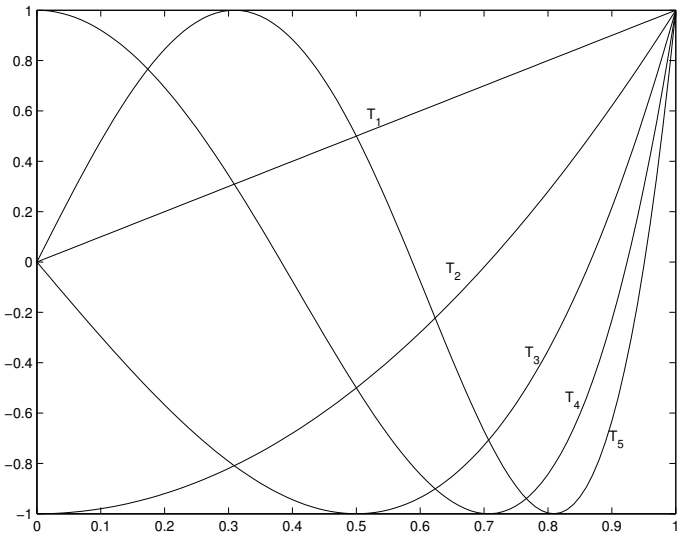

FIGURE 1.5 – Les 5 premiers polynômes de Chebyshev sont plus différents

1.3.4 Régression polynomiale

Dans les sections précédentes, nous avons montré comment on peut trouver la meilleure fonction approximant un nuage de points, lorsque l'on connaît a priori la forme de la fonction recherchée. Dans certaines applications, on recherche le meilleur polynôme sans savoir a priori le degré approprié que l'on souhaite. Il est alors utile de pouvoir résoudre plusieurs fois les équations normales très rapidement pour déterminer finalement quel est le degré optimal. Nous allons développer dans cette section une façon de faire qui utilise efficacement les polynômes orthogonaux. Dans un premier temps, nous montrons que le calcul de la *variance* de l'erreur peut nous permettre de déterminer le degré optimal. Ensuite, nous définissons le type de polynômes orthogonaux dont nous avons besoin et montrons finalement que leur utilisation permet de simplifier grandement les calculs successifs.

Supposons que nous disposions de n points expérimentaux. Nous savons qu'un polynôme de degré $n-1$ passe exactement par tous ces points. Il est également fréquent que ce polynôme admette des oscillations peu souhaitables. Il est donc préférable de considérer les polynômes de degré inférieur qui approximent au mieux tous ces points. Mais si le degré choisi est trop bas, il se peut que la solution aux équations normales soit également peu satisfai-

sante. On peut théoriquement penser calculer *toutes* les solutions pour tous les degrés inférieurs à $n-1$. Cela nous donne une suite de polynômes $q_j(x)$ de degré j. Pour chaque polynôme de degré $j < n$, on définit une *variance*

$$\sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}(u(x_i) - q_j(x_i))^2. \tag{1.5}$$

On peut prouver, en théorie statistique, que ces variances satisfont la propriété de monotonie

$$\sigma_0^2 > \sigma_1^2 > \sigma_2^2 > \cdots > \sigma_{n-1}^2 = 0.$$

Une façon de trouver le degré optimal est de réaliser que, tant que le polynôme de degré j n'est pas satisfaisant, la variance $\sigma_{j+1}^2 \ll \sigma_j^2$. Si on a, par contre, $\sigma_{j+1}^2 \approx \sigma_j^2$, il y a peu d'intérêt à considérer un polynôme de degré plus élevé que j. L'approche, pour déterminer le degré optimal, est donc de calculer les variances successives $\sigma_0^2, \sigma_1^2, \ldots$ et de s'arrêter lorsque les variances ne diminuent plus significativement. Nous allons maintenant développer la théorie des polynômes orthogonaux qui nous permettra plus tard de calculer les variances de manière très rapide.

Définition 1.2 *Le produit scalaire de deux polynômes $\langle f, g\rangle$ est une opération qui satisfait les propriétés suivantes*

(i) $\langle f, g\rangle = \langle g, f\rangle$

(ii) $\langle f, f\rangle \geq 0$ *et* $\langle f, f\rangle = 0 \implies f = 0$

(iii) $\langle af, g\rangle = a\langle f, g\rangle$ *pour tout* $a \in \mathbb{R}$

(iv) $\langle f, g + h\rangle = \langle f, g\rangle + \langle f, h\rangle$

Fixons à présent les abscisses $x_1, \ldots, x_n$ et définissons l'opération, définie pour deux polynômes p et q,

$$\sum_{i=1}^{n} p(x_i)q(x_i). \tag{1.6}$$

Il est aisé de vérifier que l'opération satisfait toutes les propriétés du produit scalaire de la Définition 1.2 excepté la propriété (ii). Cependant, si on se restreint aux polynômes de degré inférieur ou égal à $n-1$, la propriété (ii) est également satisfaite. Dans la suite de la section, nous définissons dès lors notre produit scalaire comme $\langle f, g\rangle = \sum_{i=1}^{n} p(x_i)q(x_i)$. Nous pouvons à présent définir ce qu'est un système de polynômes orthogonaux.

Définition 1.3 *L'ensemble de polynômes $(p_0, \ldots, p_t)$ est un système de polynômes orthogonaux si*

$$\langle p_i, p_j \rangle = 0$$

pour tout $i \neq j$.

Remarquons que cette définition est valable pour tout produit scalaire choisi. On peut construire une famille de polynômes orthogonaux en utilisant la formule de récurrence suivante.

Proposition 1.1 *La construction*

$$\begin{aligned} &p_0(x) = 1 \\ &p_1(x) = x - \alpha_0 \\ &p_{i+1}(x) = xp_i(x) - \alpha_i p_i(x) - \beta_i p_{i-1}(x) \quad \textit{pour } i \geq 1, \end{aligned}$$

où

$$\begin{aligned} \alpha_i &= \frac{\langle xp_i, p_i \rangle}{\langle p_i, p_i \rangle}, \\ \beta_i &= \frac{\langle xp_i, p_{i-1} \rangle}{\langle p_{i-1}, p_{i-1} \rangle} \end{aligned}$$

fournit une famille $(p_0, \ldots, p_k)$ de polynômes orthogonaux pour tout k.

La preuve de cette proposition est laissée comme exercice. La proposition nous permet, en particulier, de construire une famille orthogonale pour notre produit scalaire $\langle p, q \rangle$. Nous allons montrer que la construction de cette famille nous permet de réaliser toutes les opérations qui nous intéressent de manière efficace.

Retournons au problème des moindres carrés, et en particulier aux équations normales (1.3)-(1.4). Utilisant notre définition du produit scalaire, on voit qu'on peut réécrire le système, en utilisant $p_0, \ldots, p_k$ comme fonctions de base, comme

$$\begin{pmatrix} \langle p_0, p_0 \rangle & \langle p_0, p_1 \rangle & \cdots & \langle p_0, p_k \rangle \\ \langle p_1, p_0 \rangle & \langle p_1, p_1 \rangle & \cdots & \langle p_1, p_k \rangle \\ \vdots & & \ddots & \vdots \\ \langle p_k, p_0 \rangle & \langle p_k, p_1 \rangle & \cdots & \langle p_k, p_k \rangle \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} \langle u, p_0 \rangle \\ \langle u, p_1 \rangle \\ \vdots \\ \langle u, p_k \rangle \end{pmatrix}. \tag{1.7}$$

Mais comme notre système de fonctions de base est orthogonal, on en déduit que le système (1.7) prend l'expression très simple

$$\begin{pmatrix} \langle p_0, p_0\rangle & 0 & \cdots & 0 \\ 0 & \langle p_1, p_1\rangle & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \langle p_k, p_k\rangle \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} \langle u, p_0\rangle \\ \langle u, p_1\rangle \\ \vdots \\ \langle u, p_k\rangle \end{pmatrix}.$$

Le polynôme des moindres carrés de degré k q_k est donc donnée par $q_k(x) = \sum_{i=0}^{k} a_i p_i(x)$ et

$$a_i = \frac{\langle u, p_i\rangle}{\langle p_i, p_i\rangle}. \tag{1.8}$$

Il est particulièrement important, à ce stade, de noter que les coefficients a_i *ne dépendent pas* du nombre de polynômes dans la base (ce qui ne serait pas le cas si nous n'avions pas utilisé une base orthogonale). Nous avons donc, en particulier, que l'*interpolation polynomiale* de u est

$$q_{n-1}(x) = \sum_{i=1}^{n-1} a_i p_i(x) \tag{1.9}$$

et que les différentes approximations aux moindres carrés de degré inférieur proviennent exactement de la même somme (1.9) où cette dernière est simplement *tronquée*. Nous nous occupons à présent de calculer les variances successives $\sigma_0^2, \sigma_1^2, \ldots$ Pour ce faire, nous utilisons le résultat suivant.

Proposition 1.2 *L'ensemble de polynômes* $(p_0, \ldots, p_j, u - q_j)$ *est un système orthogonal pour tout* $0 \leq j \leq n-1$.

Démonstration: Il nous suffit de prouver que le dernier polynôme que nous avons ajouté, à savoir $u - q_j$, est orthogonal à tous les autres. En effet, on a

$$\begin{aligned} \langle u - q_j, p_k\rangle &= \langle u, p_k\rangle - \sum_{i=0}^{j} a_i \langle p_i, p_k\rangle \\ &= \langle u, p_k\rangle - a_k \langle p_k, p_k\rangle \\ &= 0 \end{aligned}$$

où la dernière égalité est obtenue en considérant (1.8). ∎

Proposition 1.3 *Les variances successives sont données par*

$$\sigma_k^2 = \frac{1}{n}(\langle u, u\rangle - \sum_{i=0}^{k} \frac{\langle u, p_i\rangle^2}{\langle p_i, p_i\rangle}).$$

Démonstration: La définition de la variance nous donne

$$\sigma_k^2 = \frac{1}{n}\sum_{i=1}^{n}(u(x_i) - q_k(x_i))^2$$

ce que nous pouvons réécrire comme

$$\begin{aligned}\sigma_k^2 &= \frac{1}{n}\langle u - q_k, u - q_k\rangle \\ &= \frac{1}{n}(\langle u - q_k, u\rangle - \langle u - q_k, q_k\rangle).\end{aligned}$$

En utilisant la Proposition 1.2 et le fait que $q_k = \sum_{i=0}^{k} a_i p_i$, on en déduit que $\langle u - q_k, q_k\rangle = 0$. Nous avons dès lors

$$\begin{aligned}\sigma_k^2 &= \frac{1}{n}(\langle u, u\rangle - \langle q_k, u\rangle) \\ &= \frac{1}{n}(\langle u, u\rangle - \sum_{i=0}^{k} a_i\langle p_i, u\rangle) \\ &= \frac{1}{n}(\langle u, u\rangle - \sum_{i=0}^{k} \frac{\langle p_i, u\rangle^2}{\langle p_i, p_i\rangle}).\end{aligned}$$

■

Le calcul des variances ne dépend, à nouveau, pas du degré final approprié. On peut donc les calculer successivement jusqu'à obtenir des variances qui ne décroissent plus significativement. Et dans ce cas, le degré du polynôme est satisfaisant.

Chapitre 2

Dérivation et intégration numérique

2.1 Dérivation

2.1.1 Introduction

Lorsque l'on dispose de la forme analytique d'une fonction, il est souvent possible de la dériver analytiquement. C'est souvent une bonne façon de faire afin d'obtenir ses dérivées numériques en différents points. Il peut arriver néanmoins que la forme analytique d'une dérivée soit etrêmement lourde à manipuler. Dans ce cas, il peut être préférable de pouvoir évaluer une dérivée par des méthodes numériques. De même, il arrive que la forme analytique d'une fonction ne soit pas connue mais que l'on souhaite tout de même disposer d'une évaluation numérique de sa dérivée. C'est le cas, par exemple, lorsque la fonction qui nous intéresse provient de la solution d'un problème ou d'une simulation. Dans ce cas, seule une évaluation numérique de la dérivée peut être réalisée. Nous allons voir dans ce chapitre que l'évaluation d'une dérivée, bien que simple conceptuellement, est assez ardue du point de vue numérique. En particulier, de nombreuses erreurs d'arrondi viennent perturber l'application de formules intuitives simples. Nous verrons que, si l'on peut calculer la valeur d'une fonction avec n chiffres significatifs, il est cependant difficile d'obtenir la même précision dans le calcul d'une dérivée. Nous pourrons néanmoins gagner de la précision en appliquant l'extrapolation de Richardson. Cette très importante technique sera également utilisée pour améliorer la précision dans le cas de l'intégration numérique et dans le

cas de la résolution d'équations différentielles ordinaires.

2.1.2 Méthode naïve d'ordre 1

Dans le reste de la section sur la dérivation numérique, une fonction $f : \mathbb{R} \mapsto \mathbb{R}$ nous est donnée. Notre tâche est d'évaluer sa dérivée $f'(x)$ en un point x fixé. Rappelons la définition de la dérivée $f'(x)$

$$f'(x) = \lim_{t \to x} \frac{f(t) - f(x)}{t - x}.$$

Partant de cette formule, une définition alternative est bien sûr de considérer un pas h se rapprochant de 0. On aura donc

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

La méthode la plus simple pour évaluer une dérivée est donc de tabuler les valeurs $\frac{f(x+h)-f(x)}{h}$ pour différentes valeurs de h. On s'attend à ce que, plus la valeur de h se rapproche de 0, plus l'approximation de la dérivée soit précise. Ceci est exprimé dans la proposition suivante

Proposition 2.1 *Soit une fonction f deux fois dérivable dont on souhaite évaluer la dérivée en x. On définit l'erreur comme $E(h) := \frac{f(x+h)-f(x)}{h} - f'(x)$. Pour tout h, il existe $\xi \in [x, x+h]$ et $C > 0$ tels que*

$$|E(h)| \leq \frac{C}{2} h.$$

Démonstration: Par un développement de Taylor, on obtient $f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\xi)$. En réarrangeant, on obtient

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(\xi)$$

c'est-à-dire

$$|E(h)| \leq \frac{C}{2} h$$

si on suppose que $|f''|$ est bornée par C sur $[x, x+h]$. ∎

h	$\lvert E(h)\rvert$
10^{-1}	0.6410000000000062
10^{-2}	0.0604010000000024
10^{-3}	0.0060040009994857
10^{-4}	0.0006000399994690
10^{-5}	0.0000600004308495
10^{-6}	0.0000059997602477
10^{-7}	0.0000006018559020
10^{-8}	0.0000000423034976
10^{-9}	0.0000003309614840
10^{-10}	0.0000003309614840
10^{-11}	0.0000003309614840
10^{-12}	0.0003556023293640
10^{-13}	0.0031971113494365
10^{-14}	0.0031971113494365
10^{-15}	0.4408920985006262
10^{-16}	4.0000000000000000

TABLE 2.1 – L'erreur sur le calcul de la dérivée $f'(1)$ en fonction du pas

La théorie nous dit donc que la méthode converge linéairement vers la dérivée recherchée. L'exemple suivant va nous montrer que, malheureusement, les erreurs d'arrondi vont empêcher la Proposition 2.1 de s'appliquer en pratique.

Exemple 2.1 Calculons la dérivée de $f(x) = x^4$ en $x = 1$. La théorie nous dit que l'on doit avoir $f'(1) = 4$. Si on calcule $\frac{f(x+h)-f(x)}{h}$ pour toutes les puissances négatives de 10, on obtient le tableau 2.1. Précisons que les calculs sont effectués avec un epsilon machine de $2\ 10^{-16}$. La Figure 2.1 représente un graphe logarithmique de l'erreur en fonction de h. On voit que la meilleure erreur est obtenue pour $h = 10^{-8}$. De plus, malgré des opérations précises à 16 décimales, on ne parvient pas à avoir une précision de plus de 8 décimales sur la valeur de la dérivée. La dégradation, même pour une fonction aussi banale, est donc flagrante. ■

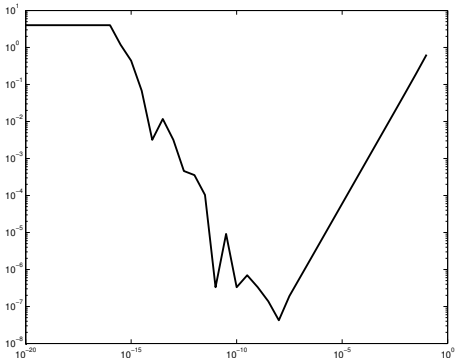

FIGURE 2.1 – Graphe logarithmique de l'erreur de calcul lors de l'évaluation de $f'(1)$

2.1.3 Différences centrées

Il est possible d'adapter légèrement la méthode précédente pour obtenir une approximation d'ordre 2 qui est également moins sensible aux erreurs d'arrondi. Cependant, pour se débarrasser totalement du problème d'instabilité numérique dû aux erreurs d'arrondi, il faudra utiliser l'extrapolation de Richardson que nous développerons dans la Section 2.2.

Dans le cas de différences centrées, on évalue la fonction f en $x+h$ et en $x-h$. La droite reliant ces deux points a une pente qui approxime la dérivée de f en x. Il s'agit en général d'une meilleure approximation que celle que nous avons développée dans la section précédente. La Figure 2.2 illustre ce fait. En pointillés, les différences centrées approximent mieux que l'expression traditionnelle en trait plein. La Proposition suivante donne la formule exacte et montre qu'il s'agit d'une méthode d'ordre 2.

Proposition 2.2 *Soit f une fonction trois fois continûment dérivable. La suite $\frac{f(x+h)-f(x-h)}{2h}$ converge vers $f'(x)$ et jouit d'une convergence d'ordre 2.*

Démonstration: Nous développons f en série de Taylor autour de x pour évaluer $f(x+h)$ et $f(x-h)$. On obtient donc respectivement

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(\xi_+)}{6}h^3 \tag{2.1}$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(\xi_-)}{6}h^3, \tag{2.2}$$

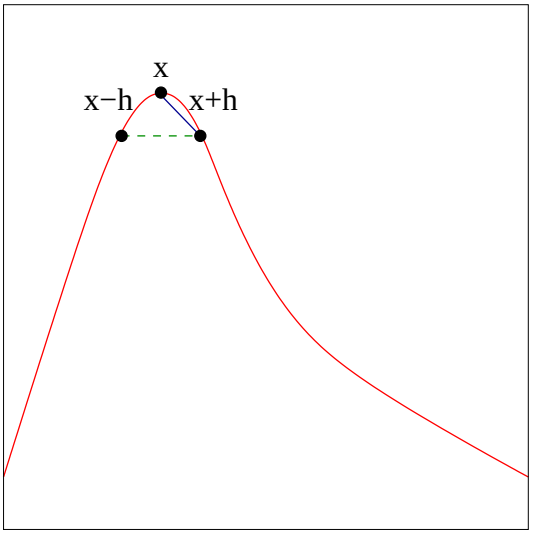

FIGURE 2.2 – Comparaison entre les différences centrées (pointillés) et la méthode naïve (trait plein)

avec $\xi_+ \in [x, x+h]$ et $\xi_- \in [x-h, x]$. En effectuant (2.1) − (2.2), les termes $f(x)$ et en h^2 se simplifient et on obtient dès lors

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f'''(\xi_+) + f'''(\xi_-)}{12} h^2.$$

Comme f''' est continue, on peut trouver une valeur $\xi \in [\xi_-, \xi_+]$ telle que $f'''(\xi) = (f'''(\xi_+) + f'''(\xi_-))/2$. On obtient donc finalement

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f'''(\xi)}{6} h^2, \tag{2.3}$$

ce qui prouve à la fois la convergence et l'ordre quadratique. ■

Exemple 2.2 Nous calculons à nouveau $f'(1)$ où $f(x) = x^4$. Cette fois, nous utilisons la méthode des différences centrées. Les résultats sont reportés dans le Tableau 2.2. ■

On peut obtenir une méthode d'ordre 4 si on utilise l'évaluation de la fonction f en deux points à droite et deux points à gauche. Cette méthode est dite de

Pas h	$\frac{f(x+h)-f(x-h)}{2h}$	$\lvert E(h)\rvert$
10^{-1}	4.0400000000000018	0.0400000000000018
10^{-2}	4.0004000000000035	0.0004000000000035
10^{-3}	4.0000039999997234	0.0000039999997234
10^{-4}	4.0000000399986746	0.0000000399986746
10^{-5}	4.0000000003925784	0.0000000003925784
10^{-6}	3.9999999999484892	0.0000000000515108
10^{-7}	4.0000000001150227	0.0000000001150227
10^{-8}	3.9999999867923464	0.0000000132076536
10^{-9}	4.0000001089168791	0.0000001089168791
10^{-10}	4.0000003309614840	0.0000003309614840
10^{-11}	4.0000003309614840	0.0000003309614840
10^{-12}	4.0001335577244390	0.0001335577244390
10^{-13}	3.9990233346998139	0.0009766653001861
10^{-14}	3.9968028886505635	0.0031971113494365
10^{-15}	4.2188474935755949	0.2188474935755949
10^{-16}	2.2204460492503131	1.7795539507496869
10^{-17}	0.0000000000000000	4.0000000000000000
10^{-18}	0.0000000000000000	4.0000000000000000
10^{-19}	0.0000000000000000	4.0000000000000000
10^{-20}	0.0000000000000000	4.0000000000000000

TABLE 2.2 – Approximation de la dérivée de x^4 en 1 par les différences centrées

différences centrées lorsque le choix des abscisses est symétrique par rapport à x. La façon la plus naturelle d'obtenir la formule est de considérer le polynôme de degré 3 passant par les 4 points $(x-2h, f(x-2h))$, $(x-h, f(x-h))$, $(x+h, f(x+h))$, et $(x+2h, f(x+2h))$ et de le dériver. Le résultat obtenu est donné dans la Proposition suivante.

Proposition 2.3 *Soit f une fonction cinq fois continûment dérivable en x. On a*

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} + \frac{h^4}{30} f^{(5)}(\xi).$$

Cette formule est dite de différence centrée d'ordre 4.

2.1.4 Différences décentrées

Dans certains cas, il n'est pas possible de calculer la valeur de f des deux côtés de x. Il peut alors être utile d'approximer la dérivée en n'utilisant que des évaluations de f venant d'un seul côté. La manière de procéder dans ce cas est identique. On fixe les abscisses où f est évaluée. On calcule ensuite le polynôme d'interpolation que l'on dérive. Illustrons le processus pour le choix d'abscisses $x, x+h, x+2h$. En utilisant la formule d'interpolation de Lagrange, le polynôme quadratique P interpolant $(x, f(x))$, $(x+h, f(x+h))$ et $(x+2h, f(x+2h))$ est donné par

$$P(t) = f(x)\frac{(t-x-h)(t-x-2h)}{2h^2} + f(x+h)\frac{(t-x)(t-x-2h)}{-h^2} + f(x+2h)\frac{(t-x)(t-x-h)}{2h^2}.$$

Si on dérive par rapport à t, on obtient

$$P'(t) = \frac{f(x)}{2h^2}((t-x-h) + (t-x-2h)) + \frac{f(x+h)}{-h^2}((t-x) + (t-x-2h)) + \frac{f(x+2h)}{2h^2}((t-x) + (t-x-h)).$$

Si on évalue en x, on obtient à présent

$$P'(x) = -\frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} \approx f'(x)$$

qui est une formule dite de *différence décentrée*. On peut prouver que la formule a une convergence quadratique. Il est intéressant de remarquer que les formules de différences décentrées nécessitent plus d'évaluations de fonctions pour obtenir le même ordre de convergence qu'une méthode de différences centrées. On peut ainsi construire une méthode d'ordre 4 qui nécessite l'évaluation de f en 5 points.

2.1.5 Dérivées d'ordre supérieur

Pour obtenir des formules d'approximation numérique des dérivées secondes ou d'ordre supérieur, on peut procéder de manière totalement similaire au cas des dérivées premières. Par exemple, on peut calculer le polynôme d'interpolation de Lagrange que l'on dérive plusieurs fois ensuite. Pour déterminer l'ordre de convergence, on se sert alors d'un développement en série de Taylor. La Proposition suivante détermine une formule centrée d'ordre 2 qui calcule une approximation de la dérivée seconde de f en x.

Proposition 2.4 *Soit f une fonction quatre fois continûment dérivable en x. On a*

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi). \tag{2.4}$$

Démonstration: Nous déduisons d'abord l'approximation (2.4) en dérivant deux fois le polynôme d'interpolation de Lagrange. Nous prouverons ensuite l'ordre de convergence en utilisant le développement de Taylor.

Le polynôme $P(t)$ passant par les points $(x, f(x))$, $(x-h, f(x-h))$ et $(x+h, f(x+h))$ peut être obtenu par la formule d'interpolation de Lagrange comme

$$P(t) = f(x)\frac{(t-x+h)(t-x-h)}{-h^2} + f(x-h)\frac{(t-x)(t-x-h)}{2h^2} + f(x+h)\frac{(t-x)(t-x+h)}{2h^2}.$$

Si on dérive deux fois P, on obtient

$$P''(t) = \frac{2f(x)}{-h^2} + \frac{f(x-h)}{h^2} + \frac{f(x+h)}{h^2}.$$

En particulier, on a bien comme approximation

$$f''(x) \approx P''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}.$$

A présent que nous disposons de la forme de l'approximation de la dérivée seconde, nous pouvons utiliser le développement en série de Taylor de f autour de x calculée en $x-h$ et $x+h$ pour déterminer l'ordre de convergence de l'approximation. Nous obtenons respectivement

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(x)}{6}h^3 + \frac{f^{(4)}(\xi_-)}{24}h^4 \qquad (2.5)$$

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \frac{f^{(4)}(\xi_+)}{24}h^4, \qquad (2.6)$$

avec $\xi_- \in [x-h, x]$ et $\xi_+ \in [x, x+h]$. Utilisant (2.4), nous déduisons par conséquent que la combinaison $(2.5) + (2.6) - 2f(x)$ permet de supprimer les termes en $f(x)$ et $f'(x)$. On obtient en effet

$$f(x-h) + f(x+h) - 2f(x) = f''(x)h^2 + \frac{f^{(4)}(\xi_-) + f^{(4)}(\xi_+)}{24}h^4. \qquad (2.7)$$

Comme $f^{(4)}$ est continue par hypothèse, on peut utiliser le théorème de la valeur intermédiaire. Il existe par conséquent $\xi \in [\xi_-, \xi_+]$ tel que $f^{(4)}(\xi) = (f^{(4)}(\xi_-) + f^{(4)}(\xi_+))/2$. On a donc finalement, en réécrivant (2.7)

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{f^{(4)}(\xi)}{12}h^2,$$

ce qui est le résultat désiré. ■

La formule centrée pour le calcul d'une dérivée seconde est encore plus rapidement dégradée par les erreurs d'arrondi comme l'indique la Table 2.3.

2.1.6 Calcul de l'erreur et choix du pas

Nous avons vu dans l'Exemple 2.1 que la stabilité numérique des approximations de la dérivée est très mauvaise. Dans cette section, nous tentons de

Pas h	$\frac{f(x+h)-2f(x)+f(x-h)}{h^2}$	$\lvert E(h)\rvert$
10^{-1}	12.0200000000000724	0.0200000000000724
10^{-2}	12.0001999999996833	0.0001999999996833
10^{-3}	12.0000019995236684	0.0000019995236684
10^{-4}	12.0000000047859601	0.0000000047859601
10^{-5}	12.0000054337765540	0.0000054337765540
10^{-6}	11.9996235170560794	0.0003764829439206
10^{-7}	12.0348175869366987	0.0348175869366987
10^{-8}	7.7715611723760949	4.2284388276239051
10^{-9}	444.0892098500625593	432.0892098500625593
10^{-10}	0.0000000000000000	12.0000000000000000
10^{-11}	0.0000000000000000	12.0000000000000000
10^{-12}	444089209.8500626	444089197.850062668
10^{-13}	-44408920985.0062	44408920997.0062
10^{-14}	0.0000000000000000	12.0000000000000000
10^{-15}	444089209850062.5	444089209850050.5
10^{-16}	-44408920985006264	44408920985006272
10^{-17}	0.0000000000000000	12.0000000000000000
10^{-18}	0.0000000000000000	12.0000000000000000
10^{-19}	0.0000000000000000	12.0000000000000000
10^{-20}	0.0000000000000000	12.0000000000000000

TABLE 2.3 – Calcul d'une dérivée seconde par une formule de différences centrées

quantifier l'erreur introduite. Nous partons de la formule de différence centrée donnée par la Proposition 2.2 et en particulier de (2.3). Nous avons donc

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f'''(\xi)}{6}h^2.$$

Cependant, nous ne disposons pas d'une valeur précise de $f(x+h)$ et de $f(x-h)$ mais de valeurs approchées $\tilde{f}(x+h)$ et $\tilde{f}(x-h)$. On suppose que l'erreur réalisée est de l'ordre de l'epsilon machine. On a donc $\tilde{f}(x+h) = f(x+h)(1+\delta_+)$ avec $|\delta_+| \leq \epsilon_M$ et $\tilde{f}(x-h) = f(x-h)(1+\delta_-)$ avec $|\delta_-| \leq \epsilon_M$ où ϵ_M est l'epsilon machine. Notre approximation calculée de la derivée peut donc s'écrire

$$\tilde{f}'_h(x) \approx \frac{f(x+h)(1+\delta_+) - f(x-h)(1+\delta_-)}{2h}(1+\delta_3),$$

si on suppose qu'une erreur ($|\delta_3| \leq \epsilon_M$) est encore réalisée sur la soustraction uniquement. La valeur absolue de l'erreur, en utilisant h comme pas, peut donc être bornée par

$$\begin{aligned} |E(h)| &= |f'(x) - \tilde{f}'_h(x)| \qquad (2.8)\\ &\approx |\frac{(\delta_- + \delta_3)f(x-h) - (\delta_+ + \delta_3)f(x+h)}{2h} - \frac{f'''(\xi)}{6}h^2| \\ &\leq \frac{2C\epsilon_M}{h} + \frac{D}{6}h^2, \qquad (2.9)\end{aligned}$$

où $|f(x)| \leq C$ dans l'intervalle $[x-h, x+h]$ et $|f'''(x)| \leq D$ dans l'intervalle $[x-h, x+h]$. Nous repérons deux termes dans (2.9). Le premier correspond à l'erreur due à l'arithmétique en virgule flottante (l'arrondi), tandis que le deuxième terme correspond à l'erreur dûe à l'approximation de la dérivée. Nous voyons que leurs dépendances respectives sont inverses. Plus le pas h est petit, plus l'erreur d'approximation est petite et l'erreur d'arrondi grande. Mais quand le pas h est grand, c'est le contraire.

Pour trouver, à présent, le pas qui donnera la plus petite erreur, il suffit de trouver h qui annule la dérivée de (2.9). Dans ce cas-ci, on trouve $-\frac{2C\epsilon_M}{h^2} + \frac{2Dh}{6} = 0$. Le pas optimal satisfait donc $h = \sqrt[3]{\frac{12C\epsilon_M}{2D}}$.

Dans le cas du calcul de $f'(x) = x^4$ en 1, on a que $f(x) \approx 1$ dans l'intervalle $[x-h, x+h]$. On peut donc utiliser $C = 1$. Quant à la dérivée troisième de f, on a $f'''(x) = 24x$. On a donc $D \approx 24$ sur l'intervalle. Rappelons que,

dans le cas d'une arithmétique "double précision", nous avons $\epsilon_M = 2\ 10^{-16}$. Cela nous donne donc comme pas optimal

$$h = \sqrt[3]{\frac{1}{2}10^{-16}} \approx 3.68\ 10^{-6}.$$

Cela correspond bien au calcul reporté dans la Table 2.2. En effet, il apparaissait que, parmi les pas considérés, le meilleur pas calculé était $h = 10^{-6}$.

2.2 Extrapolation de Richardson

2.2.1 Introduction

Jusqu'à présent, nous n'avons pu obtenir une estimation satisfaisante de la dérivée d'une fonction. En effet, malgré un calcul en double précision (avec environ 16 décimales correctes), nous ne pouvons obtenir une approximation de la dérivée ayant plus de 11 décimales correctes. Dans cette section, nous introduisons la technique d'extrapolation de Richardson qui va nous permettre de calculer une approximation de la dérivée avec 16 décimales correctes. La technique exposée ici est en réalité extrêmement générale et pourra s'appliquer également au cas du calcul d'une intégrale définie. C'est pourquoi nous lui consacrons une section et nous l'exposons dans sa généralité.

2.2.2 Extrapolation de Richardson

Considérons une quantité g que l'on souhaite évaluer. Considérons, de plus, que nous disposons d'une série d'approximations $\tilde{g}_h$ de g pour différents pas $h, \frac{h}{2}, \frac{h}{4}, \frac{h}{8}, \ldots$ en progression géométrique. Nous avons, de plus, $\lim_{h\to 0} \tilde{g}_h = g$. La formule d'approximation $\tilde{g}$ est par exemple une formule de différences centrées qui approxime une dérivée. Dans ce cas-ci et dans d'autres cas généraux, on peut prouver que $\tilde{g}$ approxime g comme

$$\tilde{g}_h = g + c_1 h + c_2 h^2 + c_3 h^3 + \cdots \tag{2.10}$$

On voit que l'on dispose d'une approximation de g d'ordre linéaire. Grâce à d'astucieuses combinaisons linéaires, nous allons être capable de créer une approximation de g plus précise que toutes celles dont nous disposons jusque là et ce, sans calculer de nouvelles approximations $\tilde{g}$.

Nous disposons donc, au départ, d'une série d'approximations de g qui convergent *linéairement*. Voyons à présent comment nous pouvons obtenir une suite qui converge *quadratiquement*. Ecrivons (2.10) pour deux pas consécutifs. Nous avons

$$\tilde{g}_h = g + c_1 h + c_2 h^2 + c_3 h^3 + \cdots \tag{2.11}$$

$$\tilde{g}_{\frac{h}{2}} = g + c_1 \frac{h}{2} + c_2 \frac{h^2}{4} + c_3 \frac{h^3}{8} + \cdots \tag{2.12}$$

Pour se débarrasser du terme en h, il suffit d'effectuer $-\frac{1}{2}(2.11) + (2.12)$. Après cette opération, nous obtenons donc

$$(\tilde{g}_{\frac{h}{2}} - \frac{1}{2}\tilde{g}_h) = \frac{1}{2}g - \frac{1}{4}c_2 h^2 - \frac{3}{8}c_3 h^3 + \cdots \tag{2.13}$$

Une nouvelle approximation de g est donc $2(\tilde{g}_{\frac{h}{2}} - \frac{1}{2}\tilde{g}_h)$. En réalisant une telle combinaison linéaire pour toute paire d'approximations consécutives, on obtient une nouvelle suite d'approximations. Mais cette suite **converge quadratiquement** vers g. Trouver cette combinaison linéaire adéquate qui permet de se débarrasser du terme d'ordre inférieur est ce qu'on appelle l'*extrapolation de Richardson*.

Nous ne sommes pas obligés de nous arrêter en si bon chemin. Il est aussi possible de se débarrasser du terme d'ordre quadratique et d'obtenir une série d'approximations qui converge cubiquement vers g. Pour ce faire, nous considérons la forme (2.13) écrite pour la paire h et $h/2$ ainsi que (2.13) écrite pour la paire $h/2$ et $h/4$. Cela nous donne respectivement

$$(2\tilde{g}_{\frac{h}{2}} - \tilde{g}_h) = g - \frac{1}{2}c_2 h^2 - \frac{3}{4}c_3 h^3 + \cdots \tag{2.14}$$

$$(2\tilde{g}_{\frac{h}{4}} - \tilde{g}_{\frac{h}{2}}) = g - \frac{1}{8}c_2 h^2 - \frac{3}{32}c_3 h^3 + \cdots \tag{2.15}$$

A présent pour éliminer le terme en h^2, nous allons réaliser $-\frac{1}{4}(2.14)+(2.15)$. Nous obtenons donc

$$(\frac{1}{4}\tilde{g}_h - \frac{3}{2}\tilde{g}_{\frac{h}{2}} + 2\tilde{g}_{\frac{h}{4}}) = \frac{3}{4}g - \frac{1}{32}c_3 h^3 + \cdots \tag{2.16}$$

La dernière équation nous donne donc une nouvelle suite d'approximations

$$g \approx \frac{1}{3}\tilde{g}_h - 2\tilde{g}_{\frac{h}{2}} + \frac{8}{3}\tilde{g}_{\frac{h}{4}}$$

qui converge **cubiquement** vers g.

Nous pouvons bien entendu répéter le processus à l'infini. Quelques remarques s'imposent. Dans toute la présentation de la méthode, nous avons considéré des pas qui se divisent à chaque fois par 2. Le processus peut bien entendu être appliqué avec des poids qui se divisent par 3 ou par 10, ou par n'importe quelle constante. Dans ce cas, la combinaison linéaire qui permet de se débarrasser du terme d'ordre inférieur change également. On peut même imaginer procéder à l'extrapolation de Richardson sur des pas qui ne sont pas en progression géométrique. Mais alors, les combinaisons linéaires qui permettent d'annuler le terme d'ordre inférieur sont à recalculer pour chaque paire de points.

Pour mieux visualiser l'extrapolation de Richardson, on peut également la représenter "graphiquement". Pour ce faire, introduisons tout d'abord quelques notations. On note par $G_{i,0} = \tilde{g}_{\frac{h}{2^i}}$. Le 0 indique que l'on a supprimé le terme d'ordre 0 et que les approximations sont d'ordre 1. Ensuite, on va noter par $G_{i,j}$ les approximations de g où l'on a supprimé le terme d'ordre j (approximation d'ordre $j+1$) et où l'approximation utilise les valeurs $g_{\frac{h}{2^{i-j}}}$ jusque $g_{\frac{h}{2^i}}$. L'expression (2.13) nous donne donc

$$G_{i,1} = 2G_{i,0} - G_{i-1,0}.$$

La Proposition suivante indique comment on obtient l'extrapolation pour j arbitraire. Nous l'acceptons sans démonstration.

Proposition 2.5 *Les valeurs $G_{i,j}$ de l'extrapolation de Richardson sont données par*

$$G_{i,j} = \frac{G_{i,j-1} - \frac{1}{2^j} G_{i-1,j-1}}{1 - \frac{1}{2^j}}.$$

Remarquons que la Proposition est bien en adéquation avec les formules (2.13) et (2.16) que nous avions trouvées plus tôt. Nous pouvons positionner

les calculs effectués comme dans le tableau suivant.

$$
\begin{array}{lccccccccc}
h & G_{0,0} & & & & & & & & \\
 & & \searrow & & & & & & & \\
\frac{h}{2} & G_{1,0} & \rightarrow & G_{1,1} & & & & & & \\
 & & \searrow & & \searrow & & & & & \\
\frac{h}{4} & G_{2,0} & \rightarrow & G_{2,1} & \rightarrow & G_{2,2} & & & & \\
 & & \searrow & & \searrow & & \searrow & & & \\
\frac{h}{8} & G_{3,0} & \rightarrow & G_{3,1} & \rightarrow & G_{3,2} & \rightarrow & G_{3,3} & & \\
 & & \searrow & & \searrow & & \searrow & & \searrow & \\
\frac{h}{16} & G_{4,0} & \rightarrow & G_{4,1} & \rightarrow & G_{4,2} & \rightarrow & G_{4,3} & \rightarrow & G_{4,4} \\
 & \mathcal{O}(h) & & \mathcal{O}(h^2) & & \mathcal{O}(h^3) & & \mathcal{O}(h^4) & & \mathcal{O}(h^5)
\end{array}
\tag{2.17}
$$

2.2.3 Application au calcul de la dérivée numérique

Dans cette section, nous allons appliquer l'extrapolation de Richardson au cas du calcul de la dérivée d'une fonction f. Pour ce faire, nous partons de la formule de différences centrées (2.3)

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f'''(\xi)}{3}h^2. \tag{2.18}$$

Lors de la présentation de l'extrapolaton de Richardson, nous cherchions à approximer g. Cette fois, notre g est en réalité $f'(x)$. Remarquons également que, dans (2.18), il n'y a pas de terme en h. On peut prouver qu'il en est ainsi de tout terme de puissance impaire de h. Pour supprimer le terme en h^2 de (2.18), il faut d'abord considérer (2.18) écrite pour h et pour $h/2$. On a ainsi

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f'''(\xi)}{3}h^2 \tag{2.19}$$

$$f'(x) = \frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} - \frac{f'''(\xi)}{3}\frac{h^2}{4} \tag{2.20}$$

A présent en effectuant (2.20) $-\frac{1}{4}$(2.19), on obtient une nouvelle approximation de $f'(x)$ d'ordre h^4 qui s'écrit

$$f'(x) \approx \frac{\left(\frac{f(x+\frac{h}{2})-f(x-\frac{h}{2})}{h}\right) - \frac{1}{4}\left(\frac{f(x+h)-f(x-h)}{2h}\right)}{1-\frac{1}{4}}.$$

h	$G_{i,0}$	$G_{i,1}$	$G_{i,2}$
10^{-1}	4.04		
10^{-2}	4.0004	4.000000000000000	
10^{-3}	4.00000399999972	3.99999999999972	3.99999999999972

TABLE 2.4 – Extrapolation de Richardson aux différences centrées pour $f(x) = x^4$.

h	$G_{i,0}$	$G_{i,1}$	$G_{i,2}$	$G_{i,3}$
2^{-1}	1.0421906109874948			
2^{-2}	1.0104492672326730	0.9998688193143991		
2^{-3}	1.0026062019289235	0.9999918468276737	1.0000000486618921	
2^{-4}	1.0006511688350699	0.9999994911371187	1.0000000007577483	0.9999999999973651
2^{-5}	1.0001627683641381	0.9999999682071609	1.0000000000118303	0.9999999999999903

TABLE 2.5 – Extrapolation de Richardson pour le calcul de $f'(x) = e^x$ en 0

On voit qu'on a simplement appliqué la Proposition 2.5 en sautant la première étape en profitant du fait qu'aucun terme d'ordre h n'existait au départ. Si on veut obtenir une meilleure approximation de la dérivée, on peut continuer à appliquer le procédé (2.17) en sautant chaque étape correspondant à la suppression d'un terme d'ordre impair.

Exemple 2.3 Revenons au calcul de la dérivée de $f(x) = x^4$ calculée en 1. Dans les sections précédentes, nous n'avons pu obtenir plus de 11 décimales correctes même en calculant avec une double précision (16 décimales). Appliquons à présent l'extrapolation de Richardson aux différences centrées obtenues précédemment. On obtient le Tableau 2.4. On voit que le premier élément calculé par l'extrapolation de Richardson est déjà correct avec toutes les décimales. Les éléments suivants du tableau sont moins corrects car déjà infectés par des erreurs d'arrondi. Ceci dit, le cas d'un polynôme est "trop facile" pour l'extrapolation. Pour voir sa puissance sur d'autres fonctions, considérons le calcul de la dérivée de $g(x) = e^x$ en $x = 0$, ce qui doit logiquement faire 1. Le calcul de l'extrapolation de Richardson à partir des différences centrées est reporté dans la Table 2.5.

Cette fois, nous avons choisi des puissances successives de 2 comme pas

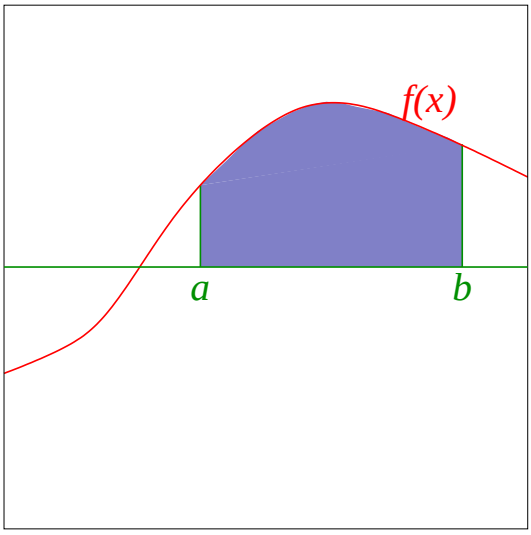

FIGURE 2.3 – $\int_a^b f(x)dx$ représente l'aire de la surface comprise entre f et l'axe des abscisses

afin que les valeurs initiales ne soient pas entâchées d'erreurs d'arrondi. On voit que les approximations successives convergent très rapidement vers la solution 1. Remarquons, à ce sujet, que la cinquième colonne (non présente dans le tableau) indiquerait une valeur exacte à 16 décimales. ■

2.3 Intégration numérique

Nous cherchons à évaluer $\int_a^b f(x)dx$. Remarquons ici que nous nous occupons uniquement du calcul d'une *intégrale définie* pour laquelle une *valeur* existe. Pour ce faire, toutes les méthodes que nous allons mettre au point tentent d'approximer l'aire de la surface sous la courbe $f(x)$ entre a et b (voir Figure 2.3). Les techniques que nous utilisons dans cette section sont très similaires au cas du calcul de la dérivée : approximation de f par un polynôme, utilisation d'un pas h tendant vers 0 et extrapolation de Richardson. Nous commençons par les méthodes approximant f par un polynôme.

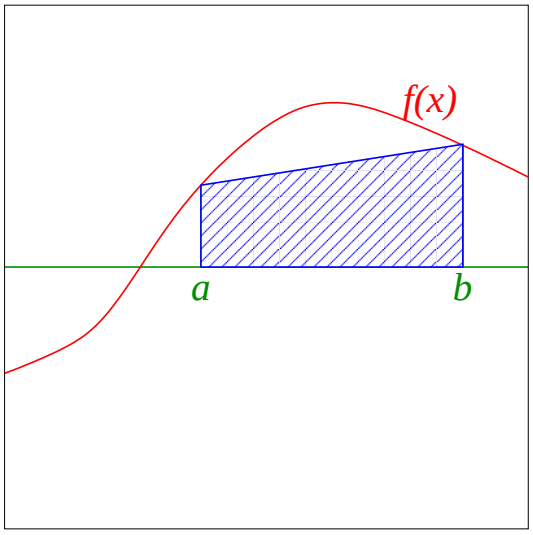

FIGURE 2.4 – La méthode des trapèzes approxime f par une droite et calcule l'aire du trapèze ainsi généré

2.3.1 Méthodes de Newton-Cotes

Dans les méthodes de Newton-Cotes, on divise l'intervalle d'intégration en $n-1$ sous-intervalles égaux à partir des n points $a = x_1, x_2, x_3, \ldots, x_{n-1}, x_n = b$. Ensuite, on calcule le polynôme d'interpolation de degré $n-1$ qui passe par ces n points et on intègre ce polynôme. Le *degré de précision* d'une méthode est le degré maximum p d'un polynôme qui est intégré de manière exacte par la méthode. Nous allons prouver que lorsque les points sont équidistants, les formules que l'on obtient sont des sommes pondérées des $f(x_i)$ multipliées par la taille de l'intervalle $(b-a)$.

La méthode la plus simple de ce type est la méthode des trapèzes. Nous considérons les deux points a et b de l'intervalle et approximons f par la droite reliant $(a, f(a))$ à $(b, f(b))$. L'aire de ce "trapèze" est égale à $\frac{(b-a)}{2}(f(a)+f(b))$. Il s'agit d'une méthode dont le degré de précision est 1. En effet, toutes les droites sont intégrées exactement par la méthode. La Figure 2.4 représente l'intuition géométrique de la méthode.

La méthode d'ordre 2 consiste à prendre deux intervalles équidistants, c'est-à-dire trois points $a, x_2 = \frac{a+b}{2}, b$. Par interpolation, on trouve le polynôme quadratique qui passe par ces trois points et on l'intègre. Par la

formule de Lagrange, on obtient

$$\begin{aligned}\int_a^b f(x)dx &\approx \int_a^b (f(a)\frac{(x-x_2)(x-b)}{(a-x_2)(a-b)} + f(x_2)\frac{(x-a)(x-b)}{(x_2-a)(x_2-b)} \\ &\qquad + f(x_3)\frac{(x-a)(x-x_2)}{(b-a)(b-x_2)})dx \\ &= \frac{1}{(b-a)^2}\int_a^b (2f(a)(x-x_2)(x-b) - 4f(x_2)(x-a)(x-b) \\ &\qquad + 2f(b)(x-a)(x-x_2))dx \\ &= \frac{1}{(b-a)^2}(2f(a)[\frac{1}{3}x^3 - \frac{x_2+b}{2}x^2 + bx_2x]_a^b \\ &\qquad - 4f(x_2)[\frac{1}{3}x^3 - \frac{a+b}{2}x^2 + abx]_a^b \\ &\qquad + 2f(b)[\frac{1}{3}x^3 - \frac{a+x_2}{2}x^2 + ax_2x]_a^b) \\ &= \frac{1}{(b-a)^2}(\frac{1}{6}f(a)(b-a)^3 + \frac{4}{6}f(\frac{a+b}{2})(b-a)^3 + \frac{1}{6}f(b)(b-a)^3) \\ &= \frac{b-a}{2}(\frac{1}{3}f(a) + \frac{4}{3}f(\frac{a+b}{2}) + \frac{1}{3}f(b)).\end{aligned}$$

Cette formule est connue sous le nom de *méthode de Simpson*. Son degré de précision est 3, c'est-à-dire que, bien que partant d'une interpolation quadratique, elle intègre parfaitement tous les polynômes ayant un degré jusque 3. L'interprétation géométrique de la formule est donnée dans la Figure 2.5.

En utilisant plus de points, on peut bien entendu intégrer un polynôme de degré plus élevé. Dans ce cas, le degré de précision sera lui aussi plus élevé. Cependant les coefficients qui apparaissent dans les formules deviennent très déséquilibrés et rendent les formules peu stables du point de vue numérique. A titre d'illustration, on obtient les formules suivantes.

$$\int_a^b f(x)dx \approx \frac{(b-a)}{2}(\frac{1}{4}f(a) + \frac{3}{4}f(\frac{2a+b}{3}) + \frac{3}{4}f(\frac{a+2b}{3}) + \frac{1}{4}f(b))$$

(formule de Simpson $\frac{3}{8}$)

$$\approx \frac{(b-a)}{2}(\frac{7}{45}f(a) + \frac{32}{45}f(\frac{3a+b}{4}) + \frac{12}{45}f(\frac{a+b}{2}) + \frac{32}{45}f(\frac{a+3b}{4}) + \frac{7}{45}f(b))$$

(formule de Boole)

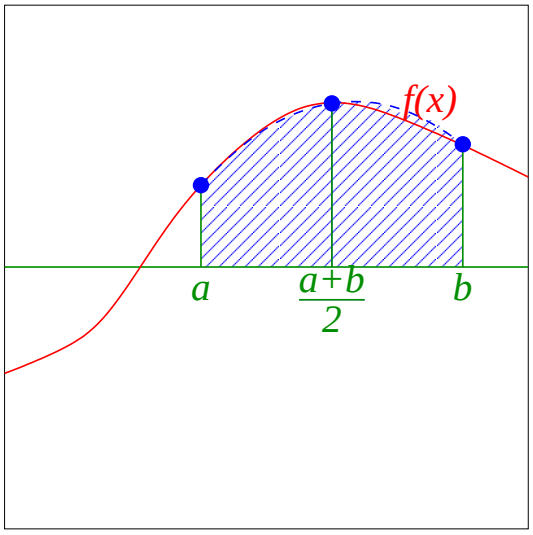

FIGURE 2.5 – La méthode de Simpson approxime f par la courbe en pointillés et calcule l'aire sous cette courbe

2.3.2 Juxtaposition d'intervalles

On applique rarement une règle de Newton-Cotes à l'entièreté de l'intervalle. Il sera, en général, plus judicieux de subdiviser l'intervalle $[a, b]$ en plusieurs sous-intervalles et d'appliquer une règle de Newton-Cotes pour chaque sous-intervalle. Par exemple, si on subdivise l'intervalle de départ en n sous-intervalles pour lesquels une méthode de Simpson doit être appliquée, nous obtenons donc $2n + 1$ points $a = x_0, x_1, x_2, \ldots, x_{2n-1}, x_{2n} = b$. Pour chaque intervalle $[x_{2k}, x_{2k+2}]$, nous appliquons la formule de Simpson. Nous obtenons dès lors par linéarité de l'intégrale

$$\begin{aligned}\int_a^b f(x)dx &= \sum_{k=0}^{n-1} \int_{x_{2k}}^{x_{2k+2}} f(x)dx \\ &\approx \sum_{k=0}^{n-1} \frac{(b-a)}{2n}\left(\frac{1}{3}f(x_{2k}) + \frac{4}{3}f(x_{2k+1}) + \frac{1}{3}f(x_{2k+2})\right) \\ &= \frac{(b-a)}{6n}(f(a) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + f(b)).\end{aligned}$$

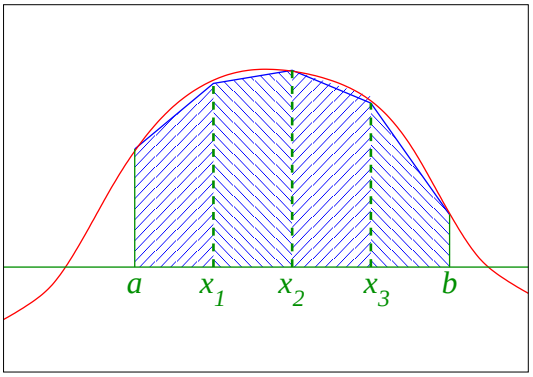

FIGURE 2.6 – La méthode composite des trapèzes

Si on note par h, l'écart entre deux abscisses consécutives, la formule de Simpson juxtaposée s'écrit de manière plus classique comme

$$\int_a^b f(x)dx \approx \frac{h}{3}(f(a) + 4f(a+h) + 2f(a+2h) + \cdots + 4f(b-h) + f(b)).$$

Similairement, on peut écrire une formule composite pour la méthode des trapèzes. On subdivise l'intervalle $[a, b]$ en n intervalles égaux. On a donc $n+1$ points $a = x_0, x_1, x_2 \ldots, x_n = b$. Pour chaque intervalle $[x_i, x_{i+1}]$, on approxime l'intégrale de f par l'aire d'un trapèze. En dénotant par h l'écart entre deux abscisses consécutives, on obtient la formule

$$\int_a^b f(x)dx \approx \frac{h}{2}(f(a)+2f(a+h)+2f(a+2h)+2f(a+3h)+\cdots+2f(b-h)+f(b)),$$

dite méthode composite des trapèzes. La Figure 2.6 indique l'intuition géométrique de la méthode.

2.3.3 Analyse de l'erreur

Dans cette section, nous déterminons l'ordre de convergence des méthodes composites. Il est, en effet, intéressant de connaître la rapidité des méthodes que nous mettons en oeuvre. Cela nous permettra plus tard également d'appliquer l'extrapolation de Richardson afin d'accélérer encore la convergence vers la valeur réelle de l'intégrale. Nous présentons dans le détail l'analyse de l'erreur dans le cas d'une formule composite des trapèzes.

Proposition 2.6 *Soit f deux fois continûment dérivable sur l'intervalle $[a, b]$. Soit $I := \int_a^b f(x)dx$ et soit T_h l'approximation obtenue par la méthode composite des trapèzes avec un pas h. On a*

$$I - T_h = -\frac{1}{12}(b-a)h^2 f''(\xi)$$

pour $\xi \in [a, b]$.

Démonstration: Nous allons tout d'abord analyser l'erreur commise sur chaque intervalle de taille h $[x_i, x_i + h]$. Nous étudions donc sans perte de généralité l'intégrale

$$\int_a^{a+h} f(x)dx$$

comparée à son approximation par l'aire d'un trapèze $\frac{h}{2}(f(a)+f(a+h))$. Par linéarité de l'intégrale, l'erreur commise en comparant ces deux intégrales est exactement égale à l'intégrale de l'erreur d'interpolation commise en utilisant le polynôme de degré 1. Dénotons donc l'erreur d'*interpolation* par $E(t)$. Elle représente donc l'erreur commise lorsqu'on remplace $f(x)$ par le polynôme de degré 1 obtenu par l'interpolation linéaire des points $(a, f(a))$ et $(b, f(b))$. Par le Théorème 1.3, on sait que l'erreur peut s'écrire, pour tout t comme

$$E(t) = \frac{f''(\xi_t)}{2}(t-a)(t-a-h),$$

où ξ_t dépend de t et se trouve dans l'intervalle $[a, a+h]$. On peut également prouver que ξ_t dépend de manière continue de t. De plus $(t-a)(t-a-h)$ ne change pas de signe sur l'intervalle $[a, a+h]$. On peut donc utiliser le lemme suivant.

Lemme *Soient f et g deux fonctions continues dont g ne change pas de signe sur $[t_0, t_1]$. Alors $\int_{t_0}^{t_1} f(t)g(t)dt = f(\xi)\int_{t_0}^{t_1} g(t)dt$ pour un certain $\xi \in [t_0, t_1]$.*

Nous avons dès lors

$$\begin{aligned}
\int_a^{a+h} E(t)dt &= \int_a^{a+h} \frac{f''(\xi_t)}{2}(t-a)(t-a-h)dt \\
&= \frac{f''(\zeta)}{2}\int_a^{a+h}(t-a)(t-a-h)dt \qquad \text{pour } \zeta \in [a, a+h] \\
&= \frac{f''(\zeta)}{2}[\frac{t^3}{3} + \frac{t^2}{2}(-2a-h) + t(a^2+ha)]_a^{a+h} \\
&= \frac{f''(\zeta)}{2}\left(a^2h + ah^2 + \frac{h^3}{3} - 2a^2h - ah^2 - ah^2 - \frac{h^3}{2} + a^2h + h^2a\right) \\
&= \frac{f''(\zeta)}{2}\left(-\frac{h^3}{6}\right) \qquad (2.21)
\end{aligned}$$

Cette dernière expression ne dépend pas de l'intervalle précis choisi. Il existe donc $\zeta_i \in [x_i, x_i + h]$ tel que l'erreur sur $[x_i, x_i + h]$ est donnée par (2.21). Nous obtenons donc finalement que l'erreur totale commise est de

$$\sum_{i=1}^{n} \int_{a+(i-1)h}^{a+ih} E(t)dt = \sum_{i=1}^{n} -\frac{f''(\zeta_i)h^3}{12}. \qquad (2.22)$$

On a, par ailleurs, que $h = \frac{b-a}{n}$. On peut donc réécrire (2.22) comme

$$\begin{aligned}
\sum_{i=1}^{n} \int_{a+(i-1)h}^{a+ih} E(t)dt &= -\frac{(b-a)}{12}h^2\left(\frac{1}{n}\sum_{i=1}^{n} f''(\zeta_i)\right) \\
&= -\frac{(b-a)}{12}h^2 f''(\xi). \qquad (2.23)
\end{aligned}$$

On obtient (2.23) par le théorème de la moyenne. On a donc $\xi \in [a, b]$. ∎

On peut de même calculer l'ordre de convergence des méthodes ayant un degré de précision supérieur, comme par exemple la méthode de Simpson ou la méthode de Boole.

Proposition 2.7 *Soit f quatre fois continûment dérivable sur l'intervalle $[a, b]$. Soit $I := \int_a^b f(x)dx$ et soit S_h l'approximation obtenue par la méthode composite de Simspon avec un pas h. On a*

$$I - S_h = -\frac{1}{90}(b-a)h^4 f^{(4)}(\xi)$$

pour $\xi \in [a, b]$.

Proposition 2.8 *Soit f six fois continûment dérivable sur l'intervalle $[a,b]$. Soit $I := \int_a^b f(x)dx$ et soit B_h l'approximation obtenue par la méthode composite de Boole avec un pas h c'est-à-dire*

$$B_h = \frac{2h}{45}(7f(a) + (32f(a+h) + 12f(a+2h) + 32f(a+2h)) + 14f(a+3h) \\ + (32f(a+4h) + 12f(a+5h) + 32f(a+6h)) + \cdots + 7f(b)).$$

On a

$$I - B_h = -\frac{8}{945}(b-a)h^6 f^{(6)}(\xi)$$

pour $\xi \in [a,b]$.

2.3.4 Méthode de Romberg

Maintenant que nous connaissons l'ordre de convergence des méthodes que nous avons étudiées précédemment, il est à présent possible d'accélérer leur convergence en tirant profit de l'extrapolation de Richardson exposée à la Section 2.2. La méthode de Romberg n'est en fait rien d'autre que l'application de l'extrapolation de Richardson à une méthode composite des trapèzes. Rappelons que la méthode composite des trapèzes avec un pas h consiste en l'approximation de l'intégrale $I = \int_a^b f(x)dx$ par

$$T_h = \frac{h}{2}(f(a) + 2f(a+h) + 2f(a+2h) + \cdots + 2f(b-h) + f(b)). \quad (2.24)$$

De plus, par la Proposition 2.6, on a

$$I - T_h = -\frac{1}{12}(b-a)h^2 f''(\xi).$$

Si on divise le pas par 2, on obtient

$$T_{\frac{h}{2}} = \frac{h}{4}(f(a) + 2f(a+\frac{h}{2}) + 2f(a+h) + \cdots + f(b)) \quad (2.25)$$

et l'erreur

$$I - T_{\frac{h}{2}} = -\frac{1}{12}(b-a)\frac{h^2}{4}f''(\xi).$$

Remarquons que pour calculer (2.25), la moitié des valeurs nécessaires ont déjà été calculées pour obtenir (2.24). Il suffit à présent d'appliquer l'extrapolation de Richardson donnée par la Proposition 2.5. Afin d'utiliser le même

genre de notation, nous dénotons par $I_{i,0} = T_{\frac{h}{2^i}}$ l'approximation de l'intégrale obtenue par la méthode des trapèzes avec un pas de $\frac{h}{2^i}$. Comme les erreurs commises ne contiennent que des puissances paires de h, nous devons appliquer l'extrapolation de Richardson dans le même esprit que dans le cas des formules de différences centrées pour la dérivation. Nous aurons donc

$$\begin{aligned} I_{i,1} &= \frac{\left(I_{i,0} - \frac{1}{2^2} I_{i-1,0}\right)}{\left(1 - \frac{1}{2^2}\right)} \\ &\vdots \\ I_{i,k} &= \frac{\left(I_{i,k-1} - \frac{1}{2^{2k}} I_{i-1,k-1}\right)}{\left(1 - \frac{1}{2^{2k}}\right)} \end{aligned}$$

La façon d'effectuer les calculs est de nouveau typique de l'extrapolation de Richardson. On obtient donc un tableau du type

$$\begin{array}{lccccccccc} h & I_{0,0} & & & & & & & & \\ & & \searrow & & & & & & & \\ \frac{h}{2} & I_{1,0} & \rightarrow & I_{1,1} & & & & & & \\ & & \searrow & & \searrow & & & & & \\ \frac{h}{4} & I_{2,0} & \rightarrow & I_{2,1} & \rightarrow & I_{2,2} & & & & \\ & & \searrow & & \searrow & & \searrow & & & \\ \frac{h}{8} & I_{3,0} & \rightarrow & I_{3,1} & \rightarrow & I_{3,2} & \rightarrow & I_{3,3} & & \\ & & \searrow & & \searrow & & \searrow & & \searrow & \\ \frac{h}{16} & I_{4,0} & \rightarrow & I_{4,1} & \rightarrow & I_{4,2} & \rightarrow & I_{4,3} & \rightarrow & I_{4,4} \\ & \mathcal{O}(h^2) & & \mathcal{O}(h^4) & & \mathcal{O}(h^6) & & \mathcal{O}(h^8) & & \mathcal{O}(h^{10}) \end{array} \tag{2.26}$$

En toute logique, la première colonne de (2.26) $I_{i,0}$ consiste en les valeurs données par la méthode des trapèzes. On peut montrer que la deuxième colonne obtenue, à savoir $I_{i,1}$, consiste en les valeurs que l'on aurait obtenues grâce à la méthode de Simpson. Les valeurs de la troisième colonne sont les valeurs de la méthode de Boole. Et ainsi de suite, on obtient ainsi toutes les valeurs des méthodes de Newton-Cotes de degré $2i$. Cet algorithme est connu sous le nom d'algorithme de Romberg.

2.3.5 Quadratures de Gauss-Legendre

La plupart des formules d'intégration consistent en une somme pondérée d'évaluations de f en certains points de l'intervalle. Les formules de Newton-

Cotes utilisent des points équidistants dans l'intervalle. En ce qui concerne la facilité d'utilisation et de présentation, c'est certainement un bon choix. Cependant rien ne dit qu'il s'agit d'un choix pertinent en ce qui concerne l'ordre et le degré de la méthode obtenue. Dans cette section, le choix des noeuds fait également partie des degrés de liberté dont on dispose pour obtenir une bonne méthode. Ainsi la méthode d'intégration qu'on recherche dans cette section est du type

$$\int_a^b f(x)dx \approx a_0 f(x_0) + \cdots + a_n f(x_n), \tag{2.27}$$

où cette fois, $x_0, \ldots, x_n$ ne sont pas nécessairement équidistants. On peut déjà faire la remarque, à ce stade, qu'un intérêt supplémentaire de ce type de méthodes est que l'on ne nécessite pas l'évaluation de la fonction f aux *bords de l'intervalle*. Cela peut être très utile si on veut évaluer une intégrale où la fonction tend vers l'infini à un des bords de l'intervalle.

Revenons à (2.27). Supposons pour commencer que $x_0, \ldots, x_n$ soient fixés. La façon la plus naturelle de déterminer les coefficients $a_0, \ldots, a_n$ est d'imposer que la formule (2.27) intègre convenablement le polynôme d'interpolation $P(t)$ passant par les $(x_i, f(x_i))$. On a dès lors

$$\begin{aligned}
\int_a^b f(x)dx &\approx \int_a^b P(t)dt \\
&= \sum_{i=0}^{n} f(x_i) \int_a^b \frac{\prod_{j\neq i}(t - x_j)}{\prod_{j\neq i}(x_i - x_j)} dt \qquad (2.28) \\
&= \sum_{i=0}^{n} a_i f(x_i) \qquad (2.29)
\end{aligned}$$

où (2.28) est obtenu grâce à la formule d'interpolation de Lagrange. Si on identifie les coefficients de (2.28) et (2.29), on voit donc que les coefficients a_i sont obtenus en intégrant les polynômes de Lagrange sur l'intervalle, c'est-à-dire

$$\begin{aligned}
a_i &= \int_a^b \frac{\prod_{j\neq i}(t - x_j)}{\prod_{j\neq i}(x_i - x_j)} dt \qquad (2.30) \\
&= \int_a^b l_i(t)dt,
\end{aligned}$$

où $l_i(t)$ représente donc les polynômes de Lagrange. Le théorème suivant, dit des quadratures de Gauss, indique comment choisir avantageusement les points x_i pour obtenir une méthode dont le degré de précision est le plus élevé possible.

Théorème 2.1 *Soit q un polynôme non nul de degré $n+1$ tel que pour tout $0 \leq k \leq n$, on a*

$$\int_a^b x^k q(x)dx = 0. \tag{2.31}$$

Soit $x_0, x_1, \ldots, x_n$ les racines de $q(x)$. Alors la formule

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i),$$

où $a_i = \int_a^b l_i(t)dt$, est exacte pour tout polynôme de degré inférieur ou égal à $2n+1$.

Démonstration: Soit f un polynôme de degré inférieur ou égal à $2n+1$. Divisons f par q, on obtient

$$f = pq + r,$$

où p est le quotient et r le reste. Ce sont deux polynômes de degré inférieur ou égal à n. Dès lors, on a $\int_a^b p(x)q(x)dx = 0$. De plus, comme x_i est une racine de q, on en déduit que $f(x_i) = r(x_i)$. On a finalement

$$\begin{aligned}\int_a^b f(x)dx &= \int_a^b p(x)q(x)dx + \int_a^b r(x)dx \\ &= \int_a^b r(x)dx.\end{aligned}$$

Pour conclure, comme r est de degré au plus n, il s'en suit que son intégrale peut être calculée par la somme pondérée. On a donc bien

$$\int_a^b r(x)dx = \sum_{i=0}^{n} a_i r(x_i) = \sum_{i=0}^{n} a_i f(x_i).$$

■

Pour résumer, on voit que la formule (2.29) est exacte pour tout polynôme de degré au plus n pour un choix arbitraire de points x_i. En revanche, si on choisit pour x_i les racines de q, la formule (2.29) est exacte pour tout polynôme de degré au plus $2n+1$.

Exemple 2.4 Déterminons la formule de quadrature de Gauss pour trois noeuds pour calculer la formule $\int_{-1}^{1} f(x)dx$.

On doit d'abord trouver le polynôme q de degré 3 qui satisfait

$$\int_{-1}^{1} q(x)dx = \int_{-1}^{1} xq(x)dx = \int_{-1}^{1} x^2 q(x)dx = 0.$$

Remarquons que si l'on prend un polynôme n'ayant que des degrés impairs, les intégrales $\int_{-1}^{1} q(x)dx$ et $\int_{-1}^{1} x^2 q(x)dx$ seront automatiquement nulles. Nous notons donc $q(x) = q_1 x + q_3 x^3$. Tentons de trouver les coefficients de q en annulant la dernière intégrale. Nous avons

$$\begin{aligned}\int_{-1}^{1} xq(x)dx &= \int_{-1}^{1} (q_1 x^2 + q_3 x^4)dx \\ &= [\frac{q_1}{3}x^3 + \frac{q_3}{5}x^5]_{-1}^{1} \\ &= \frac{2q_1}{3} + \frac{2q_3}{5}.\end{aligned}$$

Remarquons que le polynôme est défini à un facteur près. Nous pouvons choisir $q_1 = -3$ et $q_3 = 5$ comme solution. Nous avons donc finalement

$$q(x) = 5x^3 - 3x,$$

dont les racines sont $-\sqrt{\frac{3}{5}}, 0$ et $\sqrt{\frac{3}{5}}$. Notre formule d'intégration est donc provisoirement $\int_{-1}^{1} f(x)dx = a_1 f(-\sqrt{\frac{3}{5}}) + a_2 f(0) + a_3 f(\sqrt{\frac{3}{5}})$. Pour obtenir la formule complète, il nous suffit à présent d'intégrer les polynômes de Lagrange (2.30) sur l'intervalle. On a donc respectivement (sachant que

l'intégrale des parties impaires est nulle)

$$\begin{aligned}
a_1 &= \int_{-1}^{1} \frac{t(t-\sqrt{\frac{3}{5}})}{\left(\sqrt{\frac{3}{5}}\right)\left(2\sqrt{\frac{3}{5}}\right)}dt \\
&= \frac{5}{6}[\frac{t^3}{3}]_{-1}^{1} &= \frac{5}{9} \\
a_2 &= \int_{-1}^{1} \frac{(t+\sqrt{\frac{3}{5}})(t-\sqrt{\frac{3}{5}})}{\left(\sqrt{\frac{3}{5}}\right)\left(-\sqrt{\frac{3}{5}}\right)}dt \\
&= -\frac{5}{3}[\frac{t^3}{3} - \frac{3}{5}]_{-1}^{1} = \frac{5}{3}(\frac{6}{5} - \frac{2}{3}) = \frac{8}{9} \\
a_3 &= \int_{-1}^{1} \frac{t(t+\sqrt{\frac{3}{5}})}{\left(2\sqrt{\frac{3}{5}}\right)\left(\sqrt{\frac{3}{5}}\right)}dt \\
&= \frac{5}{6}[\frac{t^3}{3}]_{-1}^{1} &= \frac{5}{9}
\end{aligned}$$

Notre formule d'intégration est donc finalement

$$\int_{-1}^{1} f(x)dx \approx \frac{5}{9}f(-\sqrt{\frac{3}{5}}) + \frac{8}{9}f(0) + \frac{5}{9}f(\sqrt{\frac{3}{5}}).$$

Appliquons cette formule au calcul de $\int_{-1}^{1} e^x dx$ dont la valeur exacte est $e - \frac{1}{e} \approx 2.350402$ avec 7 chiffres significatifs. On trouve cette fois

$$\begin{aligned}
\int_{-1}^{1} e^x dx &\approx \frac{5}{9}e^{-\sqrt{\frac{3}{5}}} + \frac{8}{9} + \frac{5}{9}e^{\sqrt{\frac{3}{5}}} \\
&\approx 2.350337,
\end{aligned}$$

ce qui fait encore 5 chiffres significatifs. Si on compare à la formule de Simpson classique qui utilise trois points, on obtient

$$\begin{aligned}
\int_{-1}^{1} e^x dx &\approx \frac{1}{3}e^{-1} + \frac{4}{3} + \frac{1}{3}e \\
&\approx 2.36205
\end{aligned}$$

qui n'a que trois chiffres significatifs. ■

n	racines x_i	Poids a_i
1	$-\sqrt{\frac{1}{3}}$	1
	$\sqrt{\frac{1}{3}}$	1
2	$-\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
	0	$\frac{8}{9}$
	$\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
3	$-\sqrt{\frac{1}{7}(3-4\sqrt{0.3})}$	$\frac{1}{2}+\frac{1}{12}\sqrt{\frac{10}{3}}$
	$-\sqrt{\frac{1}{7}(3+4\sqrt{0.3})}$	$\frac{1}{2}-\frac{1}{12}\sqrt{\frac{10}{3}}$
	$+\sqrt{\frac{1}{7}(3-4\sqrt{0.3})}$	$\frac{1}{2}+\frac{1}{12}\sqrt{\frac{10}{3}}$
	$+\sqrt{\frac{1}{7}(3+4\sqrt{0.3})}$	$\frac{1}{2}-\frac{1}{12}\sqrt{\frac{10}{3}}$

TABLE 2.6 – Les racines des trois premiers polynômes de Legendre et les poids correspondants pour l'intervalle $[-1, 1]$

Les polynômes q du Théorème 2.1 sont appelés *les polynômes de Legendre*. Il s'agit, tout comme les polynômes de Chebyshev, d'une famille de polynômes orthogonaux. Ils peuvent se trouver par la formule de récurrence

$$\begin{aligned} q_0(x) &= 1 \\ q_1(x) &= x \\ q_n(x) &= \left(\frac{2n-1}{n}\right) x q_{n-1}(x) - \left(\frac{n-1}{n}\right) q_{n-2}(x). \end{aligned}$$

Les racines x_i et les poids a_i sont tabulées. Dans la Table 2.6, on reprend les trois premiers cas.

Chapitre 3

Systèmes linéaires

Les opérateurs linéaires sont les plus simples parmi les opérateurs mathématiques et constituent de ce fait des modèles naturels pour un ingénieur. Tout problème mathématique à structure linéaire se traduira toujours au niveau du calcul numérique par un problème d'algèbre linéaire. Ces problèmes sont fréquents : on a estimé que septante cinq pourcents des problèmes scientifiques font intervenir la résolution d'un système d'équations linéaires. Il est donc important d'être à même de résoudre ces problèmes rapidement et avec précision.

L'algèbre linéaire est un excellent exemple de la différence entre les mathématiques classiques et l'analyse numérique. Bien que la théorie ait été connue depuis des siècles, ce n'est qu'au cours des dernières décennies que le traitement numérique est apparu. Les règles classiques de Cramer sont particulièrement mal adaptées au calcul numérique car, si n représente la dimension du problème, le nombre d'opérations à effectuer est d'ordre $n\,!$ alors que pour les méthodes que nous allons exposer, ce nombre d'opérations est d'ordre n^3. De même, nous calculerons rarement explicitement l'inverse d'une matrice pour résoudre un système linéaire. Le nombre d'opérations requises pour calculer l'inverse étant souvent beaucoup plus élevé que le nombre d'opérations réellement nécessaire pour résoudre un problème usuel.

3.1 Méthodes directes pour la résolution de systèmes linéaires

Une *méthode directe* pour la résolution d'un système d'équations linéaires est une méthode qui donne, aux erreurs d'arrondi près, la solution exacte après un nombre fini d'étapes de calcul. Pour un système d'équations $Ax = b$, où la matrice A est dense (i.e., la plupart des éléments de A sont non-nuls), il n'existe pas d'algorithme qui soit meilleur, aussi bien du point de vue temps que du point de vue précision, que la méthode d'élimination systématique de Gauss. Par contre, lorsque la matrice A est creuse (i.e., un grand nombre des éléments de A sont nuls), les *méthodes itératives* offrent certains avantages et elles sont indispensables dans certains cas de très grands systèmes. Les méthodes itératives fournissent une suite de solutions approchées, convergente lorsque le nombre d'étapes tend vers l'infini. Pour certains systèmes ayant une structure spéciale, les méthodes itératives peuvent donner des résultats utiles avec moins d'opérations que les méthodes directes. Le choix entre une méthode directe et une méthode itérative dépendra de la proportion et de la répartition des éléments non-nuls de la matrice A.

3.1.1 Systèmes triangulaires

Un système d'équations linéaires dont la matrice est triangulaire est particulièrement simple à résoudre. Considérons un système linéaire $Lx = b$ dont la matrice $L = [l_{ij}]$ est triangulaire inférieure. Si l'on suppose que $l_{ii} \neq 0,\ i = 1, 2, ..., n,$ alors les inconnues peuvent être déterminées dans l'ordre direct $x_1,\ x_2, ...,\ x_n$ au moyen des formules :

$$x_i = \frac{b_i - \sum_{k=1}^{i-1} l_{ik}\, x_k}{l_{ii}} \qquad i = 1, 2, ...,\ n \tag{3.1}$$

L'algorithme est appelé *substitution avant*. Dans le cas d'une matrice triangulaire supérieure, on procède à la *substitution arrière*. D'après la formule (3.1), chaque étape i de la résolution d'un système triangulaire d'équations linéaires nécessite $i - 1$ multiplications, $i - 1$ additions et 1 division, soit un total de

$$[2\,(i-1) + 1]$$

opérations pour l'étape i. Utilisant alors la formule

$$\sum_{i=1}^{n} i = \frac{1}{2}\, n(n+1) \tag{3.2}$$

on calcule que le nombre total d'opérations nécessaires à la résolution d'un système linéaire triangulaire est égal à

$$\sum_{i=1}^{n} [2\,(i-1)+1] = n^2. \tag{3.3}$$

3.1.2 Elimination Gaussienne

La méthode d'élimination de Gauss devrait vous être familière. Il s'agit en effet de la méthode classique pour résoudre un système linéaire. L'idée de la méthode est d'éliminer les inconnues d'une façon systématique jusqu'à l'obtention d'un système triangulaire que nous pouvons résoudre aisément ainsi qu'on l'a vu dans le paragraphe précédent. Considérons le système

$$\begin{array}{ccccccccc} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \cdots & + & a_{1n}\,x_n & = & b_1 \\ a_{21}\,x_1 & + & a_{22}\,x_2 & + & \cdots & + & a_{2n}\,x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}\,x_1 & + & a_{n2}\,x_2 & + & \cdots & + & a_{nn}\,x_n & = & b_n \end{array} \tag{3.4}$$

Nous supposerons dans ce qui suit que la matrice $A = [a_{ij}]$ est non-singulière ; par conséquent, le système (3.4) a une solution unique.

Supposons alors que $a_{11} \neq 0$. Sous cette hypothèse, on peut éliminer x_1 des $(n-1)$ dernières équations en soustrayant de l'équation i le multiple

$$m_{i1} = \frac{a_{i1}}{a_{11}}\,, \quad i = 2, 3, ..., n$$

de la première équation. Les $(n-1)$ dernières équations deviennent ainsi

$$\begin{array}{ccccccccc} a_{22}^{(2)}\,x_2 & + & a_{23}^{(2)}\,x_3 & + & \cdots & + & a_{2n}^{(2)}\,x_n & = & b_2^{(2)} \\ \vdots & & & & & & & & \\ a_{n2}^{(2)}\,x_2 & + & a_{n3}^{(2)}\,x_3 & + & \cdots & + & a_{nn}^{(2)}\,x_n & = & b_n^{(2)}. \end{array}$$

où les nouveaux coefficients sont donnés par

$$a_{ij}^{(2)} = a_{ij} - m_{i1}a_{1j}\,, \qquad b_i^{(2)} = b_i - m_{i1}b_1 \qquad i,\ j = 2,3,...,n$$

Il s'agit donc d'un système de $(n-1)$ équations aux $(n-1)$ inconnues $x_2,\ x_3,...,\ x_n$. Si $a_{22}^{(2)} \neq 0$, on peut d'une façon similaire éliminer x_2 des $(n-2)$ dernières équations. Nous obtiendrons alors au moyen des multiplicateurs

$$m_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}\,, \qquad i = 3,\ 4,...,\ n$$

un système de $(n-2)$ équations aux $(n-2)$ inconnues $x_3,\ x_4,\ ,...,\ x_n$ dont les coefficients seront

$$a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2}\,a_{2j}^{(2)}\,, \qquad b_i^{(3)} = b_i^{(2)} - m_{i2}\,b_2^{(2)} \qquad i\ ,= 3,\ 4,...,\ n.$$

Les éléments $a_{11},\ a_{22}^{(2)},\ a_{33}^{(3)},...$ intervenant dans la déterminations des multiplicateurs des étapes successives de l'élimination sont appelés les *pivots*. Si ces éléments sont tous différents de zéro, nous pouvons poursuivre l'élimination jusqu'à l'étape $(n-1)$ qui nous fournira l'unique équation :

$$a_{nn}^{(n)}\,x_n = b_n^{(n)}$$

Rassemblant alors la première équation de chacune des étapes d'élimination, on obtient le système triangulaire suivant :

$$\begin{array}{rcccccl} a_{11}^{(1)}\,x_1 & + & a_{12}^{(1)}\,x_2 & + \cdots + & a_{1n}^{(1)}\,x_n & = & b_1^{(1)} \\ & & a_{22}^{(2)}\,x_2 & + \cdots + & a_{2n}^{(2)}\,x_n & = & b_2^{(2)} \\ & & & & \vdots & & \\ & & & & \vdots & & \\ & & & & a_{nn}^{(n)}\,x_n & = & b_n^{(n)} \end{array} \tag{3.5}$$

où, par simple souci de cohérence des notations, on a introduit $a_{ij}^{(1)} = a_{ij}$, $j = 1,\ 2,...,\ n$ et $b_1^{(1)} = b_1$. Ce système triangulaire supérieur peut alors être résolu par substitution arrière comme vu au paragraphe précédent.

Remarquons que, durant tout l'algorithme, les opérations réalisées sur les lignes de A le sont aussi sur les lignes de b. Ce constat permet de considérer

b comme une colonne supplémentaire de A. De la même façon, si le système doit être résolu pour plusieurs membres de droite, il sera aisé de simplement considérer toutes les variantes de b comme des colonnes supplémentaires de A. Les opérations effectuées sur A ne sont pas affectées par ces ajouts.

3.1.3 Complexité de l'élimination de Gauss

Nous allons maintenant estimer le nombre d'opérations nécessaires à l'algorithme de Gauss pour obtenir un système triangulaire.

Théorème 3.1 *Considérons les p systèmes $Ax = b_i$, $i = 1, \ldots, p$. Si on effectue l'élimination de Gauss pour obtenir simultanément p systèmes triangulaires, le nombre d'opérations nécessaires est*

$$\frac{2}{3}n^3 + (p - \frac{1}{2})n^2 - (p + \frac{1}{6})n.$$

Démonstration: Considérons l'étape i de l'élimination Gaussienne où i va de 1 à $n-1$. Pour chaque ligne à éliminer, on réalise 1 division pour obtenir le multiplicateur, $(n - i + p)$ multiplications et $(n - i + p)$ soustractions pour éliminer le coefficient en dessous du pivot. Or à l'étape i, il reste $(n-i)$ lignes à éliminer. Cela nous fait donc un total de $(n-i)(2n-2i+2p+1)$ opérations à l'étape i. Le nombre total d'opérations est donc de

$$\sum_{i=1}^{n-1}[(n - i)(2n - 2i + 2p + 1)]. \tag{3.6}$$

En utilisant les deux formules

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$
$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6},$$

on peut donc écrire (3.6) successivement comme

$$\begin{aligned}
&\sum_{i=1}^{n-1}[2i^2+(-4n-2p-1)i+(2n^2+2pn+n)]\\
=&\frac{(n-1)n(2n-1)}{3}+(-4n-2p-1)\frac{(n-1)n}{2}+(n-1)(2n^2+2pn+n)\\
=&(\frac{2}{3}n^3-n^2+\frac{1}{3}n)+(-2n^3+(2-p-\frac{1}{2})n^2+(p+\frac{1}{2})n)\\
&\qquad\qquad\qquad\qquad\qquad\qquad+2n^3+(2p-1)n^2+(-2p-1)n\\
=&\frac{2}{3}n^3+(p-\frac{1}{2})n^2+(-p-\frac{1}{6})n.
\end{aligned}$$

■

3.1.4 Choix des pivots

D'après la description même de l'algorithme, on se rend compte que la méthode d'élimination de Gauss n'est plus applicable si pour une certaine valeur de k l'élément pivot $a_{kk}^{(k)}=0$. Considérons par exemple le système :

$$\begin{array}{rcrcrcl}
x_1 & + & x_2 & + & x_3 & = & 1\\
x_1 & + & x_2 & + & 2x_3 & = & 2\\
x_1 & + & 2x_2 & + & 2x_3 & = & 1
\end{array}\qquad(3.7)$$

Ce système est non-singulier et a pour solution unique $x_1=-x_2=x_3=1$. Néanmoins, après la première étape d'élimination, on obtient :

$$\begin{array}{rcrcl}
 & & x_3 & = & 1\\
x_2 & + & x_3 & = & 0
\end{array}$$

de sorte que $a_{22}^{(2)}=0$ et que l'algorithme tel qu'il a été décrit précédemment n'est plus applicable. Le remède est évidemment de permuter les équations 2 et 3 avant l'étape suivante d'élimination, ce qui d'ailleurs dans ce cas-ci, donne directement le système triangulaire recherché. Une autre façon de faire serait de permuter les colonnes 2 et 3. Il ne faut alors pas oublier de tenir compte également de la même permutation de l'ordre des inconnues.

Dans le cas général où à l'étape k, nous avons $a_{kk}^{(k)}=0$, alors au moins un des éléments $a_{ik}^{(k)},\ i=k,\ k+1,...,\ n$ de la colonne k doit être non-nul sinon les

k premières colonnes de $A^{(k)} = [a_{ij}^{(k)}]$ seraient linéairement dépendantes d'où A serait singulière. Supposons $a_{rk}^{(k)} \neq 0$: on peut alors permuter les lignes k et r et continuer l'élimination. Il s'ensuit que tout système d'équations non singulier peut être réduit à une forme triangulaire par élimination de Gauss et des permutations éventuelles des lignes.

Afin d'assurer la *stabilité numérique* de l'algorithme, il sera souvent nécessaire de permuter des lignes non seulement quand un élément pivot est exactement égal à zéro, mais également quand il est proche de zéro. A titre d'exemple, revenons au système (3.7) et supposons que l'élément a_{22} est modifié et devient 1.0001 au lieu de 1 précédemment. L'élimination de Gauss effectuée sans permutation de lignes conduit au système triangulaire suivant :

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ 0.0001\, x_2 + x_3 &= 1 \\ 9999\, x_3 &= 10000 \end{aligned}$$

La substitution arrière, utilisant l'arithmétique en virgule flottante à quatre chiffres fournit la solution :

$$x_1' = 0 \ , \quad x_2' = 0 \ , \quad x_3' = 1.000$$

tandis que la véritable solution, arrondie à quatre décimales, est :

$$x_1 = 1.000 \ , \quad -x_2 = x_3 = 1.0001$$

Par contre, si l'on permute les lignes 2 et 3, on obtient le système triangulaire :

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ x_2 + x_3 &= 0 \\ 0.9999\, x_3 &= 1 \end{aligned}$$

qui fournit par substitution arrière (en utilisant la même précision que précédemment) la solution :

$$x_1 = -x_2 = x_3 = 1.000$$

qui est correcte à trois décimales.

La question des erreurs d'arrondi sera étudiée plus en détail au paragraphe 3.2.4. Nous noterons seulement ici qu'afin d'éviter d'éventuelles erreurs catastrophiques comme illustré dans l'exemple ci-dessus, il est généralement nécessaire de choisir l'élément pivot de l'étape k au moyen d'une des deux stratégies suivantes :

(i) *Pivotage partiel.* Choisir r comme le plus petit entier pour lequel on a :

$$|a_{rk}^{(k)}| = \max |a_{ik}^{(k)}| \ , \quad k \leq i \leq n$$

et permuter les lignes k et r.

(ii) *Pivotage complet.* Choisir r et s comme les plus petits entiers pour lesquels on a :

$$|a_{rs}^{(k)}| = \max |a_{ij}^{(k)}| \ , \quad k \leq i, \ j \leq n$$

et permuter les lignes k et r ainsi que les colonnes k et s.

Le pivotage partiel consiste donc à sélectionner comme pivot de l'étape k l'élément le plus grand en valeur absolue et le plus proche de $a_{kk}^{(k)}$ dans le dessous de la colonne k. Le pivotage complet consiste à sélectionner comme pivot de l'étape k l'élément le plus grand en valeur absolue et le plus proche de $a_{kk}^{(k)}$ dans l'ensemble des éléments qu'il reste encore à traiter. En pratique, le pivotage partiel est suffisant et le pivotage complet est donc rarement utilisé étant donné l'importance beaucoup plus grande du travail de recherche.

3.1.5 Décomposition LU

On a vu que l'élimination de Gauss permet de traiter plusieurs seconds membres à la fois pour autant que l'on connaisse tous ces seconds membres dès le début de l'opération d'élimination. Il existe cependant des cas où il n'en est pas ainsi ; on peut par exemple avoir à résoudre les systèmes $Ax_1 = b_1$ et $Ax_2 = b_2$ où b_2 est une fonction de x_1. La décomposition LU permet d'éviter de devoir recommencer toute l'opération. Le principe est le suivant : si l'on connaît une décomposition de A en une matrice triangulaire inférieure L et une matrice triangulaire supérieure U, c'est-à-dire :

$$A = LU$$

alors le système $Ax = b$ est équivalent au système $LUx = b$ qui se décompose en deux systèmes triangulaires :

$$Ly = b \ , \quad Ux = y$$

que l'on résoudra avec $2\,n^2$ opérations au lieu des $(2/3\,n^3 + 1/2\,n^2 - 7/6\,n)$ que nécessiterait une nouvelle élimination de Gauss.

Une telle décomposition LU n'existe pas toujours. Cependant, si on considère une matrice dont l'élimination de Gauss a pu se dérouler en choisissant à chaque étape le pivot diagonal (c'est-à-dire sans permutation des lignes), alors cette décomposition LU existe et ses éléments peuvent être aisément récupérés par l'élimination de Gauss.

Théorème 3.2 *Soit A une matrice donnée d'ordre n pour laquelle l'élimination de Gauss peut être effectuée sans permutation de lignes. Alors cette matrice a une décomposition LU dont les éléments de L et U sont donnés par les éléments de l'élimination de Gauss.*

Démonstration: Lorsque l'élimination de Gauss peut être effectuée sans permutation de lignes, l'algorithme peut être décrit comme la détermination de la suite de matrices :

$$A = A^{(1)},\ A^{(2)}, ...,\ A^{(n)}$$

au moyen des $n-1$ transformations

$$A^{(k+1)} = M_k A^{(k)}, \qquad k = 1, 2, ...,\ n-1 \tag{3.8}$$

où

$$M_k = \left(\begin{array}{ccc|ccc} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & -m_{k+1,k} & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -m_{n,k} & 0 & \cdots & 1 \end{array}\right)$$

$$= I - m_k e_k^T = \left(\begin{array}{c|c} I & 0 \\ \hline -X & I \end{array}\right)$$

avec

$$m_k^T = (0,\ 0, ...,\ 0,\ m_{k+1,k}, ...,\ m_{n,k})$$

$$e_k^T = (0,\ 0, ...,\ \underset{\uparrow}{1},\ 0,\ 0, ...,\ 0)$$
$$k$$

et

$$X = \begin{pmatrix} 0 & \cdots & 0 & m_{k+1,k} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & m_{n,k} \end{pmatrix}.$$

Quant à l'expression $m_k e_k^T$, il s'agit d'un produit dyadique du genre

$$uv^{(T)} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \begin{pmatrix} v_1 & \cdots & v_n \end{pmatrix} = (u_i\, v_j)$$

Les formules (3.8) fournissent

$$A^{(n)} = M_{n-1}\, M_{n-2} \ldots M_2\, M_1\, A^{(1)}$$

d'où

$$A = A^{(1)} = M_1^{-1}\, M_2^{-1} \ldots M_{n-2}^{-1}\, M_{n-1}^{-1}\, A^{(n)}$$

La matrice $A^{(n)}$ étant triangulaire supérieure, il reste à prouver que le produit des matrices M_k^{-1} est une matrice triangulaire inférieure. Constatons tout d'abord que

$$M_k^{-1} = \begin{pmatrix} I & 0 \\ -X & I \end{pmatrix}^{-1} = \begin{pmatrix} I & 0 \\ X & I \end{pmatrix} = I + m_k e_k^T$$

Ensuite, définissons les matrices

$$L_k := M_1^{-1}\, M_2^{-1} \ldots M_k^{-1}$$

et montrons que ces matrices sont de la forme

$$L_k = I + m_1 e_1^T + m_2 e_2^T + \ldots + m_k e_k^T$$

En effet, c'est vrai pour $k = 1$ puisque $L_1 = M_1^{-1} = I + m_1 e_1^T$. Montrons ensuite que si c'est vrai pour k quelconque, c'est vrai pour $k+1$:

$$\begin{aligned} L_{k+1} &= L_k M_{k+1}^{-1} \\ &= \left(I + m_1 e_1^T + \ldots + m_k e_k^T\right)\left(I + m_{k+1}\, e_{k+1}^T\right) \\ &= \left(I + m_1 e_1^T + \ldots + m_k e_k^T + m_{k+1}\, e_{k+1}^T\right) \\ &\quad + \left(m_1 \left(e_1^T m_{k+1}\right) + m_2 \left(e_2^T m_{k+1}\right) + \ldots + m_k \left(e_k^T m_{k+1}\right)\right) e_{k+1}^T \\ &\qquad = 0 \qquad\qquad = 0 \qquad\qquad = 0 \end{aligned}$$

Par conséquent $L_{n-1} = M_1^{-1} M_2^{-1} ... M_{n-1}^{-1}$ est une matrice triangulaire inférieure. On a donc obtenu une décomposition LU de la matrice A où L contient les multiplicateurs et U les éléments transformés de la méthode d'élimination de Gauss :

$$L = (m_{ik}) \ , \quad i \geq k \ , \quad U = \left(a_{kj}^{(k)} \right) , \quad k \leq j \tag{3.9}$$

■

Il est également possible de prouver que cette décomposition LU est unique. On constate donc que pour obtenir la décomposition LU d'une matrice A, il suffit de procéder à l'élimination de Gauss et de conserver les multiplicateurs. Sur un ordinateur, on procèdera de la façon suivante : puisque le multiplicateur $m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ est déterminé de façon à rendre $a_{ik}^{(k+1)}$ égal à zéro, il suffit de remplacer en mémoire $a_{ik}^{(k)}$ par m_{ik}. Les éléments de la diagonale principale de L n'ont pas besoin d'être stockés puisque l'on sait qu'ils sont tous égaux à un. En procédant de la sorte, on n'a besoin d'aucune mémoire supplémentaire et l'élimination de Gauss a pour effet la transformation suivante :

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \rightarrow \begin{pmatrix} a_{11}^{(n)} & a_{12}^{(n)} & \cdots & a_{1,n-1}^{(n)} & a_{1n}^{(n)} \\ m_{21} & a_{22}^{(n)} & \cdots & a_{2,n-1}^{(n)} & a_{2n}^{(n)} \\ \vdots & & & \vdots & \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & a_{nn}^{(n)} \end{pmatrix}$$

3.2 Analyse de l'erreur dans la résolution des systèmes linéaires

3.2.1 Introduction

Dans la résolution pratique d'un système d'équations linéaires, il y a deux sources principales d'erreurs à considérer. Tout d'abord, il arrivera souvent que les éléments de A et b ne soient pas connus exactement : il convient alors d'examiner l'erreur sur la solution x résultant de cette incertitude. Le second type d'erreurs est évidemment constitué par les erreurs d'arrondi en cours de calcul. On comprendra l'importance d'une analyse correcte des effets de ce type d'erreurs en se rappelant que pour un système de grande taille on devra effectuer plusieurs millions d'opérations.

Si $\bar{x}$ est une solution calculée du système $Ax = b$, on appelle *vecteur résidu*, le vecteur $r = b - A\bar{x}$. On pourrait alors penser, puisque $r = 0$ implique $\bar{x} = A^{-1}b$, que si r est petit, alors $\bar{x}$ est une solution précise. Il n'en est pas toujours ainsi, comme on peut s'en rendre compte par l'exemple suivant :

Exemple 3.1 Soit le système

$$A = \begin{pmatrix} 1.2969 & 0.8648 \\ \\ 0.2161 & 0.1441 \end{pmatrix} \quad , \quad b = \begin{pmatrix} 0.8642 \\ \\ 0.1440 \end{pmatrix} \tag{3.10}$$

et supposons que l'on a obtenu la solution

$$\bar{x} = (0.9911 \ , \ \ -0.4870)^T.$$

Le vecteur résidu correspondant à cet $\bar{x}$ est égal à

$$r = (-10^{-8} \ , \ \ 10^{-8})^T$$

et l'on pourrait donc s'attendre à ce que l'erreur sur $\bar{x}$ soit faible. En fait, il n'en est rien, car aucun des chiffres de $\bar{x}$ n'est significatif ! La solution exacte est

$$x = (2, \ -2)^T$$

Dans ce cas particulier, il est facile de se rendre compte du très mauvais conditionnement du système (3.10). En effet, l'élimination de x_1, dans la deuxième équation, conduit à l'équation

$$a_{22}^{(2)} \, x_2 = b_2^{(2)}$$

où

$$a_{22}^{(2)} = 0.1441 - \frac{0.2161}{1.2969} \, 0.8648 = 0.1441 - 0.1440999923 \simeq 10^{-8}$$

Il est donc évident qu'une petite perturbation de l'élément $a_{22} = 0.1441$ provoquera une grande modification de $a_{22}^{(2)}$ et donc finalement de x_2. Il en résulte que si les coefficients de A et b ne sont pas donnés avec une précision supérieure à 10^{-8}, il n'y a aucun sens à parler d'une solution à (3.10). ■

3.2.2 Normes vectorielles et normes matricielles

Pour l'analyse des erreurs, il est utile de pouvoir associer à tout vecteur et à toute matrice un scalaire non-négatif qui mesure en quelque sorte leur grandeur. Un tel scalaire, lorsqu'il satisfait certains axiomes, sera appelé une norme.

Définition 3.1 *On dit que* $\|x\|$ *est une* norme vectorielle *si les axiomes suivants sont satisfaits.*

(i) $\|x\| > 0$ *pour tout* $x \neq 0$ *et* $\|x\| = 0$ *implique* $x = 0$

(ii) $\|x + y\| \leq \|x\| + \|y\|$

(iii) $\|\alpha x\| = |\alpha|\|x\|$ *pour tout* $\alpha \in \mathbb{R}$

Les normes vectorielles les plus souvent utilisées appartiennent à la famille des normes l_p définies par

$$\|x\|_p = (|x_1|^p + |x_2|^p + ... + |x_n|^p)^{1/p} \qquad 1 \leq p < \infty \tag{3.11}$$

Les valeurs de p les plus souvent utilisées sont

$$p = 1, \qquad \|x\|_1 = |x_1| + |x_2| + ... + |x_n| \tag{3.12}$$

$$p = 2, \qquad \|x\|_2 = (|x_1|^2 + |x_2|^2 + ... + |x_n|^2)^{1/2} \tag{3.13}$$

$$p \to \infty \qquad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i| \tag{3.14}$$

Le cas $p = 2$ correspond bien entendu à la norme euclidienne habituelle. On peut montrer que d'une façon générale, les normes du type (3.11) (y compris le cas limite $p \to \infty$) satisfont bien les axiomes (i) à (iii).

Définition 3.2 *On dit que* $\|A\|$ *est une* norme matricielle *si les axiomes suivants sont satisfaits.*

(i) $\|A\| > 0$ *pour toute* $A \neq 0$, *et* $\|A\| = 0$ *implique* $A = 0$

(ii) $\|A + B\| \leq \|A\| + \|B\|$

(iii) $\|\alpha A\| = |\alpha|\|A\|$ *pour tout* $\alpha \in \mathbb{R}$

Si de plus, les deux axiomes supplémentaires suivants sont satisfaits

(iv) $\|Ax\| \leq \|A\|\|x\|$

(v) $\|AB\| \leq \|A\|\|B\|$

alors on dit que la norme matricielle $\|A\|$ *est* compatible *avec la norme vectorielle* $\|x\|$.

Bien que l'axiome (v) ne fasse pas intervenir de norme vectorielle, on peut montrer que s'il n'est pas satisfait, alors $\|A\|$ ne peut être compatible avec aucune norme vectorielle.

Soit alors $\|A\|$ une norme matricielle compatible avec une certaine norme vectorielle $\|x\|$. Si pour toute matrice A, il existe un vecteur $x \neq 0$, tel que l'axiome (iv) est satisfait avec le signe d'égalité, alors on dira que $\|A\|$ est une norme matricielle *subordonnée* à la norme vectorielle $\|x\|$. On peut montrer que toute norme matricielle subordonnée donne l'unité pour valeur de la norme de la matrice unité. A toute norme vectorielle, il correspond au moins une norme matricielle subordonnée (et donc au moins une norme matricielle compatible) donnée par la relation

$$\|A\| = \max_{\|x\|=1} \|Ax\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \tag{3.15}$$

Nous utiliserons toujours des normes matricielles de ce dernier type. On peut calculer que les normes subordonnées répondant à (3.15) et correspondant aux normes matricielles (3.12) à (3.14) sont données par

$$\begin{aligned} p = 1, \qquad & \|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}| \\ p = 2, \qquad & \|A\|_2 = (\text{valeur propre maximum de } A^T A)^{1/2} \\ p \to \infty \qquad & \|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}| \end{aligned}$$

Dans le cas où $p = 2$, on utilise aussi parfois, étant donné la difficulté de calculer la valeur propre maximum, la norme de *Frobenius*

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

On peut montrer que cette norme est compatible avec la norme vectorielle euclidienne, mais elle n'est pas subordonnée à cette norme. On a en effet $\|I\|_F = \sqrt{n}$.

Exemple 3.2 Calculons les normes usuelles du vecteur $x = (\ -1 \quad 2 \quad -3\ )^T$. On trouve respectivement

$$\begin{aligned}
\|x\|_1 &= |-1| + |2| + |-3| &&= 6,\\
\|x\|_2 &= \sqrt{1+4+9} &&= \sqrt{14} \approx 3.74,\\
\|x\|_\infty &= \max\{|-1|, |2|, |-3|\} &&= 3.
\end{aligned}$$

Calculons à présent quelques normes de la matrice

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}.$$

On trouve respectivement

$$\begin{aligned}
\|A\|_1 &= \max\{1+4+7, 2+5+8, 3+6+9\} = 18,\\
\|A\|_2 &= \max\{\text{val. propre de } A^T A\}^{1/2}\\
&= \max\{0, 1.07, 16.85\} &&\approx 16.85\\
\|A\|_\infty &= \max\{1+2+3, 4+5+6, 7+8+9\} = 24\\
\|A\|_F &= \sqrt{1+4+9+16+\cdots+81} &&\approx 16.88.
\end{aligned}$$

Remarquons que le fait que $\det(A) = 0$ n'implique nullement que $\|A\| = 0$. ■

3.2.3 Effets des perturbations sur les données

Dans cette section, nous allons voir que certains systèmes sont *mal conditionnés*, c'est-à-dire qu'une petite perturbation des données induit une grande perturbation de la solution. Cet effet est résumé par le *nombre de conditionnement* d'une matrice. Plus ce nombre est élevé, plus les sytèmes résolus avec la matrice comme membre de gauche sont sensibles aux variations sur les données.

Définition 3.3 *Soit $A \in \mathbb{R}^{n\times n}$ une matrice non singulière, on définit*

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

Ce nombre est appelé le nombre de conditionnement *de A.*

Nous examinons à présent l'effet d'une perturbation sur les données, et nous verrons que le nombre de conditionnement intervient dans la borne que l'on peut trouver sur l'erreur. Nous commençons par une perturbation sur le second membre.

Proposition 3.1 *Soit $A \in \mathbb{R}^{n\times n}$ une matrice non singulière, $b \in \mathbb{R}^n$ et la solution $x \in \mathbb{R}^n$ de $Ax = b$. On peut borner l'erreur relative faite sur x, lorsque l'on résout $Ax = (b + \delta b)$ au lieu de $Ax = b$, par*

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A)\frac{\|\delta b\|}{\|b\|}.$$

Démonstration: La solution de notre système modifié est $x + \delta x$. On a dès lors

$$A(x + \delta x) = (b + \delta b).$$

Comme $Ax = b$, on a donc $\delta x = A^{-1}(\delta b)$. Par conséquent,

$$\|\delta x\| \leq \|A^{-1}\|\|\delta b\|. \tag{3.16}$$

En divisant (3.16) par $\|x\|$ et en utilisant $\|b\| \leq \|A\|\|x\|$, c'est-à-dire $\|x\| \geq \frac{\|b\|}{\|A\|}$, on obtient donc

$$\begin{aligned}\frac{\|\delta x\|}{\|x\|} &\leq \|A^{-1}\|\|\delta b\|\frac{\|A\|}{\|b\|}\\ &\leq \kappa(A)\frac{\|\delta b\|}{\|b\|}\end{aligned}$$

■

Etudions à présent l'effet d'une perturbation sur la matrice A elle-même.

Proposition 3.2 *Soit $A \in \mathbb{R}^{n\times n}$ une matrice non singulière, $b \in \mathbb{R}^n$ et la solution $x \in \mathbb{R}^n$ de $Ax = b$. On peut borner l'erreur relative faite sur x, lorsque l'on résout $(A + \delta A)x = b$ au lieu de $Ax = b$, par*

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq \kappa(A)\frac{\|\delta A\|}{\|A\|}.$$

Démonstration: La solution de notre système modifié peut s'écrire $x+\delta x$, on a, dès lors,

$$(A + \delta A)(x + \delta x) = b.$$

En utilisant le fait que $Ax = b$, on en déduit que

$$A\delta x + \delta A(x + \delta x) = 0,$$

c'est-à-dire $\delta x = -A^{-1}\delta A(x + \delta x)$ et par conséquent

$$\|\delta x\| \leq \|A^{-1}\|\|\delta A\|\|x + \delta x\|$$

que l'on peut écrire sous la forme

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq \kappa(A)\frac{\|\delta A\|}{\|A\|}$$

■

Exemple 3.3 La matrice A de l'exemple 3.1 a pour inverse

$$A^{-1} = 10^8 \begin{pmatrix} 0.1441 & -0.8648 \\ -0.2161 & 1.2969 \end{pmatrix}$$

Par conséquent, $\|A^{-1}\|_\infty = 1.5130 \times 10^8$. Or, $\|A\|_\infty = 2.1617$ donc

$$\kappa(A) = 2.1617 \times 1.5130 \times 10^8 \approx 3.3 \times 10^8$$

Ce système est donc extrêmement mal conditionné. ■

Notons, pour terminer, que dans le cas d'une norme matricielle subordonnée à une norme vectorielle, on a toujours $\|I\| = 1$ et dès lors, $\kappa(A) \geq 1$.

3.2.4 Erreurs d'arrondi dans la méthode d'élimination de Gauss

Nous avons vu plus tôt que, par le simple jeu des erreurs d'arrondi en cours de calcul, l'élimination de Gauss pouvait conduire à des solutions totalement erronées. On a alors proposé des stratégies de choix des pivots permettant

d'obtenir la véritable solution du système. L'analyse des erreurs d'arrondi en cours de calcul que nous allons effectuer dans ce paragraphe, va nous permettre de justifier ces stratégies.

Pour évaluer l'erreur effectivement commise, nous allons rechercher quelle est la matrice initiale qui *aurait donné* le résultat calculé avec les erreurs d'arrondi.

Théorème 3.3 *Soit $\bar{L} = (\bar{m}_{ik})$ et $\bar{U} = (\bar{a}_{kj}^{(n)})$ les facteurs triangulaires calculés par l'algorithme d'élimination de Gauss. Alors il existe une matrice E telle que $\bar{L}\,\bar{U}$ est la décomposition exacte de $A + E$, c'est-à-dire*

$$\bar{L}\,\bar{U} = A + E \tag{3.17}$$

Si l'on a utilisé une stratégie de pivotage (partiel ou complet) et une arithmétique en virgule flottante avec un epsilon machine ϵ_M alors la matrice E est bornée par

$$\|E\|_\infty \leq n^2 g_n \epsilon_M \|A\|_\infty$$

où

$$g_n = \frac{\max\limits_{i,j,k} |\bar{a}_{ij}^{(k)}|}{\max\limits_{i,j} |a_{ij}|}. \tag{3.18}$$

Démonstration: Rappelons qu'à l'étape k de l'élimination, les éléments de la matrice $A^{(k)}$ sont transformés selon les formules

$$\begin{gathered} m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\ , \qquad a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}\, a_{kj}^{(k)} \\ i,\ j = k+1,\ k+2, ...,\ n \end{gathered} \tag{3.19}$$

Dénotons par une barre supérieure les valeurs $\bar{m}_{ik}$ et $\bar{a}_{ij}^{(k+1)}$ effectivement calculées par l'arithmétique en virgule flottante et considérons que ces valeurs sont les résultats d'opérations exactes du type (3.19) effectuées non plus sur les $\bar{a}_{ij}^{(k)}$, mais sur ces éléments perturbés par des quantités $\varepsilon_{ij}^{(k)}$. On a donc

$$\bar{m}_{ik} = \frac{\bar{a}_{ik}^{(k)} + \varepsilon_{ik}^{(k)}}{\bar{a}_{kk}^{(k)}} \tag{3.20}$$

$$\bar{a}_{ij}^{(k+1)} = \bar{a}_{ij}^{(k)} + \varepsilon_{ij}^{(k)} - \bar{m}_{ik}\bar{a}_{kj}^{(k)}. \tag{3.21}$$

Si l'on prend $\bar{m}_{ii} = 1$ et que l'on fait la somme des équations (3.21) pour $k = 1,\ 2, ...,\ n-1$, on obtient les relations

$$a_{ij} = \sum_{k=1}^{p} \bar{m}_{ik}\, \bar{a}_{kj}^{(k)} - e_{ij}\ , \qquad e_{ij} = \sum_{k=1}^{r} \varepsilon_{ij}^{(k)}$$
$$p = \min(i,\ j)\ , \qquad r = \min(i-1,\ j) \tag{3.22}$$

Les relations (3.22) ne sont évidemment rien d'autre que les relations (3.17) écrites composante par composante. Il nous reste à déterminer une borne pour $\|E\|_\infty$. Rappelons que les éléments calculés par l'arithmétique en virgule flottante ne satisfont pas les relations (3.19) mais plutôt les relations :

$$\bar{m}_{ik} = \frac{\bar{a}_{ik}^{(k)}}{\bar{a}_{kk}^{(k)}}\,(1+\delta_1) \tag{3.23}$$

$$\bar{a}_{ij}^{(k+1)} = \left(\bar{a}_{ij}^{(k)} - \bar{m}_{ik}\,\bar{a}_{kj}^{(k)}(1+\delta_2)\right)(1+\delta_3) \tag{3.24}$$

où

$$|\delta_i| \le \epsilon_M\ , \qquad i = 1,\ 2,\ 3.$$

De la comparaison de (3.20) et (3.23) on déduit immédiatement que

$$\varepsilon_{ik}^{(k)} = \bar{a}_{ik}^{(k)}\delta_1. \tag{3.25}$$

Si l'on écrit (3.24) sous la forme

$$\bar{m}_{ik}\,\bar{a}_{kj}^{(k)} = \frac{\bar{a}_{ij}^{(k)} - \bar{a}_{ij}^{(k+1)}/(1+\delta_3)}{1+\delta_2}$$

et que l'on porte ce résultat dans (3.21), on obtient

$$\varepsilon_{ij}^{(k)} = \bar{a}_{ij}^{(k+1)}\left(1-(1+\delta_3)^{-1}(1+\delta_2)^{-1}\right) - \bar{a}_{ij}^{(k)}\left(1-(1+\delta_2)^{-1}\right) \tag{3.26}$$

Négligeant les puissances supérieures de ϵ_M, (3.25) et (3.26) nous fournissent les bornes supérieures suivantes :

$$|\varepsilon_{ik}^{(k)}| \le \epsilon_M|\bar{a}_{ik}^{(k)}|, \quad |\varepsilon_{ij}^{(k)}| \le 3\epsilon_M \max(\,|\bar{a}_{ij}^{(k)}|, |\bar{a}_{ij}^{(k+1)}|)\ , \quad j \ge k+1. \tag{3.27}$$

Ces résultats sont valables sans que l'on ait fait la moindre hypothèse sur les multiplicateurs $\bar{m}_{ik}$. On voit donc que *ce qu'il convient d'éviter, c'est la*

croissance des éléments $\bar{a}_{ik}$, et que la grandeur des multiplicateurs est sans effet direct. Le choix d'une stratégie pivotale sera donc dicté par la nécessité d'éviter une trop forte croissance des éléments transformés. Si l'on retourne alors aux formules de transformation (3.19) on se rend compte que le choix d'un pivot maximal semble rationnel. Nous supposerons donc dans ce qui suit que l'on a procédé aux pivotages, soit partiels soit complets. Dans l'un comme dans l'autre cas, on a alors $|\bar{m}_{ik}| \leq 1$. Eliminant $\bar{a}_{ij}^{(k)}$ des équations (3.21) et (3.24), on obtient

$$\varepsilon_{ij}^{(k)} = \bar{a}_{ij}^{(k+1)} \left(1 - (1+\delta_3)^{-1}\right) - \bar{m}_{ik}\bar{a}_{kj}^{(k)}\delta_2$$

Négligeant alors les puissances supérieures de ϵ_M et puisque $|\bar{m}_{ik}| \leq 1$, on obtient la borne supérieure

$$|\varepsilon_{ij}^{(k)}| \leq 2\epsilon_M \max(\,|\bar{a}_{ij}^{(k+1)}|, |\bar{a}_{kj}^{(k)}|), \quad j \geq k+1. \tag{3.28}$$

La définition (3.18) et la définition de la norme maximum ($p = \infty$) permettent d'écrire

$$|\bar{a}_{ij}^{(k)}| \leq \max_{i,j,k} |\bar{a}_{ij}^{(k)}| \leq g_n \max_{i,j} |a_{ij}| \leq g_n \|A\|_\infty$$

d'où les relations (3.28) et (3.27) fournissent les bornes suivantes

$$|\varepsilon_{ij}^{(k)}| \leq g_n \|A\|_\infty \cdot \begin{cases} \epsilon_M & \text{si} \quad i \geq k+1\,, \quad j = k \\ 2\epsilon_M & \text{si} \quad i \geq k+1\,, \quad j \geq k+1 \end{cases}$$

Retournant alors aux relations (3.22 b), on obtient

$$\begin{aligned} i \leq j \quad \Rightarrow \quad r = i-1, \quad |e_{ij}| \leq \sum_{k=1}^{i-1} |\varepsilon_{ij}^{(k)}| \quad &\leq \quad \sum_{k=1}^{i-1} g_n \|A\|_\infty 2\epsilon_M \\ &= g_n \|A\|_\infty 2\epsilon_M (i-1) \\ i > j \quad \Rightarrow \quad r = j, \qquad |e_{ij}| \leq \sum_{k=1}^{j} |\varepsilon_{ij}^{(k)}| \quad &\leq \quad g_n \|A\|_\infty \left(\sum_{k=1}^{j-1} 2\epsilon_M + \epsilon_M \right) \\ &= g_n \|A\|_\infty \epsilon_M (2j-1) \end{aligned}$$

soit finalement

$$(|e_{ij}|) \leq g_n \epsilon_M \|A\|_\infty \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 2 & 2 & \cdots & 2 & 2 \\ 1 & 3 & 4 & \cdots & 4 & 4 \\ & \vdots & & \ddots & \vdots & \\ 1 & 3 & 5 & \cdots & 2n-4 & 2n-4 \\ 1 & 3 & 5 & \cdots & 2n-3 & 2n-2 \end{pmatrix}$$

où l'inégalité est satisfaite composante par composante. La norme maximum de la matrice intervenant dans le membre de droite de cette dernière inégalité étant

$$\sum_{j=1}^{n} (2\,j-1) = n^2$$

on obtient bien finalement le résultat annoncé. ■

S'il est bien clair, d'après les formules (3.27) que ce qu'il faut éviter c'est la croissance des éléments transformés $a_{ij}^{(k)}$, il est moins évident que le choix systématique d'un élément maximal comme pivot, empêchera cette croissance. On ne prétendra donc pas que le choix d'un élément maximal comme pivot constitue la meilleure stratégie, et il est en fait des cas où cette stratégie sera loin d'être la meilleure. Ce que nous constaterons par contre, c'est que jusqu'à présent, aucune alternative pratique n'a été proposée.

Une analyse similaire des erreurs apparaissant dans la résolution d'un système triangulaire permet d'en borner l'erreur. Nous ne rentrons, cette fois, pas dans les détails de la preuve.

Théorème 3.4 *Soit le système linéaire $Lx = b$ où la matrice $L = (\,l_{ij})$ est triangulaire inférieure. Le vecteur $\bar{x}$ calculé par substitution avant est la solution exacte du système triangulaire perturbé*

$$(L + \delta L)\bar{x} = b \tag{3.29}$$

où la matrice de perturbation δL est bornée par

$$\|\delta L\|_\infty \leq \frac{n(n+1)}{2} \epsilon_M \max_{i,j} |l_{ij}| \tag{3.30}$$

3.2.5 Changements d'échelle et équilibrage des équations

Dans un système linéaire $Ax = b$, les inconnues x_j ainsi que les seconds membres b_i ont souvent une signification physique. Un changement des unités dans lesquelles sont mesurées les inconnues sera équivalent à un changement d'échelle de ces inconnues (càd $x_j = \alpha_j\, x'_j$) tandis qu'un changement des unités dans lesquelles sont mesurés les seconds membres sera équivalent à la multiplication de la i^e équation par un facteur β_i. Le système original sera donc transformé en un système équivalent :

$$A'\, x' = b'$$

où

$$A' = D_2\, A\, D_1\ , \qquad b' = D_2\, b\ , \qquad x = D_1\, x'$$

avec

$$D_1 = \mathrm{diag}\,(\alpha_1,\ \alpha_2, ...,\ \alpha_n)\ , \qquad D_2 = \mathrm{diag}\,(\beta_1,\ \beta_2, ...,\ \beta_n)$$

Il semble naturel que la précision de la solution calculée ne soit pas affectée par les transformations ci-dessus. Dans une certaine mesure, il en est bien ainsi comme nous l'indique le théorème suivant que nous énonçons sans le démontrer.

Théorème 3.5 *Soient $\bar{x}$ et $\bar{x}'$ les solutions calculées des deux systèmes $Ax = b$ et $(D_2\, A\, D_1)\, x' = D_2\, b$. Si D_1 et D_2 sont des matrices diagonales dont les éléments sont des puissances entières de la base du système de numération utilisé, de sorte que les changements d'échelles n'introduisent pas d'erreurs d'arrondi, alors l'élimination de Gauss effectuée en arithmétique à virgule flottante sur les deux systèmes produira, à condition que l'on choisisse les mêmes pivots dans les deux cas, des résultats qui ne diffèrent que par leurs exposants, et l'on aura exactement $\bar{x} = D_1\, \bar{x}'$.*

On voit donc que l'effet d'un changement d'échelle est essentiellement d'influencer le choix des pivots. Par conséquent, à toute séquence de choix de pivots correspondent certains changements d'échelle tels que ce choix soit réalisé. Il est donc évident que des changements d'échelle inappropriés peuvent conduire à de mauvais choix des pivots.

Exemple 3.4 Soit le système linéaire

$$\begin{pmatrix} 1 & 10000 \\ 1 & 0.0001 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10000 \\ 1 \end{pmatrix}$$

dont la solution, correctement arrondie à quatre décimales, est $x_1 = x_2 = 0.9999$. La stratégie de pivotage partiel sélectionne a_{11} comme pivot, ce qui nous donne, si l'on utilise une arithmétique en virgule flottante à trois chiffres, le résultat

$$\bar{x}_2 = 1.00 \qquad \bar{x}_1 = 0.00$$

qui est évidemment de peu de valeur. Par contre, si l'on multiplie d'abord la première équation par 10^{-4}, on obtient le système équivalent

$$\begin{pmatrix} 0.0001 & 1 \\ 1 & 0.0001 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Cette fois, le pivot est a_{21} et, toujours avec la même arithmétique, on obtient maintenant le résultat

$$\bar{x}_2 = 1.00 \qquad \bar{x}_1 = 1.00$$

nettement meilleur que le précédent. ■

On recommande souvent *d'équilibrer les équations* avant de procéder à l'élimination. On dit que les équations sont équilibrées lorsque les conditions suivantes sont satisfaites :

$$\max_{1 \le j \le n} |a_{ij}| = 1 \ , \qquad i = 1,\ 2, ...,\ n$$

On peut remarquer que dans l'Exemple 3.4, c'est précisément ce que l'on a fait. On a procédé à l'équilibrage des équations.

Il convient cependant de ne pas conclure de façon hâtive. Il n'est pas nécessairement vrai qu'un système équilibré ne donne pas lieu à des difficultés. En effet, les changements d'échelle éventuellement effectués sur les inconnues influencent manifestement l'équilibrage et par conséquent certains choix d'échelles peuvent finalement mener à des situations délicates ainsi qu'on va le voir dans l'exemple qui suit.

Exemple 3.5 Soit le système équilibré $Ax = b$ avec

$$A = \begin{pmatrix} \varepsilon & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad A^{-1} = \frac{1}{4} \begin{pmatrix} 0 & -2 & 2 \\ -2 & 1-\varepsilon & 1+\varepsilon \\ 2 & 1+\varepsilon & 1-\varepsilon \end{pmatrix}$$

où $|\,\varepsilon\,| << 1$. Il s'agit d'un système bien conditionné : $\kappa\,(A) = 3$ en norme maximum. La méthode d'élimination de Gauss avec pivotage partiel donne donc une solution précise. Par contre, le choix de $a_{11} = \varepsilon$ comme pivot a un effet désastreux sur la précision de la solution calculée.

Considérons ensuite les changements d'échelle $x'_2 = x_2/\varepsilon$ et $x'_3 = x_3/\varepsilon$. Si le système qui en résulte est à nouveau équilibré, on obtient $A'\,x' = b'$ où

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

Dans ce dernier cas, le pivotage partiel et même complet, sélectionne $a'_{11} = 1$ comme premier pivot, c'est-à-dire le même pivot que dans le cas de la matrice A. On en déduit donc, d'après le Théorème 3.5 que ce choix de pivot conduit également à une situation désastreuse pour la matrice A'. Il n'est pas difficile, dans ce cas-ci, de se rendre compte de ce qui s'est passé. En fait, *les changements d'échelle ont modifié le nombre de conditionnement de la matrice* comme on peut s'en rendre compte en observant que :

$$(A')^{-1} = \frac{1}{4}\begin{pmatrix} 0 & -2 & 2 \\ -2 & \dfrac{1-\varepsilon}{\varepsilon} & \dfrac{1+\varepsilon}{\varepsilon} \\ 2 & \dfrac{1+\varepsilon}{\varepsilon} & \dfrac{1-\varepsilon}{\varepsilon} \end{pmatrix}$$

d'où l'on calcule $\|A'\|_\infty = 3$, $\|(A')^{-1}\|_\infty = \dfrac{1+\varepsilon}{2\,\varepsilon}$ et par conséquent :

$$\kappa\,(A') = \frac{3\,(1+\varepsilon)}{2\,\varepsilon} >> \kappa\,(A)$$

c'est-à-dire que le problème de résoudre $A'\,x' = b'$ est beaucoup plus mal conditionné que celui de résoudre $Ax = b$. ■

3.3 Méthodes itératives

3.3.1 Introduction et principe de base

Dans certaines applications, on doit résoudre des systèmes linéaires de très grande taille. C'est souvent le cas, par exemple, lorsqu'on doit résoudre des

systèmes d'équations aux dérivées partielles. Typiquement on trouve alors des systèmes contenant des centaines de milliers de lignes et de colonnes. Cependant dans la plupart des cas, les matrices que l'on retrouve sont extrêmement creuses. La technique de l'élimination de Gauss devient alors peu appropriée. En effet, outre le fait que sa complexité devient importante pour de telles tailles de problèmes, on voit également apparaître le phénomène de *fill in*, c'est-à-dire que, alors que la matrice initiale A est creuse, la factorisation LU est pleine et ne profite pas de la structure de A. On préférera donc, pour de telles matrices, utiliser les méthodes itératives que nous développons dans cette section.

Le principe de base des méthodes itératives est, comme pour le cas de systèmes non linéaires, de générer des vecteurs $x^{(1)}, \ldots, x^{(k)}$ qui s'approchent de plus en plus d'une solution $\bar{x}$ du système $Ax = b$. Nous allons appliquer ici une méthode de point fixe que nous verrons au chapitre suivant dans le cadre de la résolution de systèmes non linéaires. Pour ce faire, nous devons réécrire le système $Ax = b$ en isolant x. Il serait bien entendu possible d'écrire $x = A^{-1}b$ mais cela nécessite de connaître A^{-1}, ce qui revient à résoudre le système. Au lieu de cela, nous sélectionnons une matrice régulière Q, et écrivons le système initial comme

$$Qx = Qx - Ax + b. \tag{3.31}$$

Pour un choix judicieux de matrice Q, la méthode itérative consiste donc à résoudre le système (3.31) à chaque itération et ainsi écrire

$$x^{(k)} = Q^{-1}[(Q - A)x^{(k-1)} + b]. \tag{3.32}$$

Tout l'art des méthodes itératives est de choisir convenablement la matrice Q. Pour ce faire, il convient de prêter attention à deux points. Le premier point important est de pouvoir résoudre efficacement et rapidement le système (3.31) ou en d'autres termes de pouvoir calculer efficacement Q^{-1}. Le deuxième point important est, comme nous l'avons vu dans le Chapitre 4, de garantir la convergence de la méthode du point fixe. Idéalement, on voudra choisir une matrice Q qui implique une convergence *rapide* pour le processus. Avant de passer à l'exposé des méthodes en tant que telles, nous allons faire une première analyse de la convergence de la méthode afin de diriger notre choix.

Proposition 3.3 *Soit un système $Ax = b$ dont la solution est $\bar{x}$. Si on applique le processus* (3.32) *à partir d'un point de départ $x^{(1)}$, l'erreur $e^{(k)} :=$*

$x^{(k)} - \bar{x}$ suit la règle

$$e^{(k)} = (I - Q^{-1}A)e^{(k-1)}. \tag{3.33}$$

Démonstration: On calcule, en utilisant (3.32),

$$\begin{aligned} x^{(k)} - \bar{x} &= (I - Q^{-1}A)x^{(k-1)} - \bar{x} + Q^{-1}b \\ &= (I - Q^{-1}A)x^{(k-1)} - (I - Q^{-1}A)\bar{x} \\ &= (I - Q^{-1}A)e^{(k-1)}. \end{aligned}$$

■

La Proposition 3.3 nous indique quelle est la voie à choisir pour obtenir une bonne convergence. En effet, le facteur $(I-Q^{-1}A)$ dans (3.33) doit être le plus proche possible de 0 pour obtenir une bonne convergence. Le choix limite est évidemment de choisir $Q = A$, mais cela nécessite, comme nous l'avons dit plus haut, de résoudre le problème initial. Néanmoins, la conclusion vague de la Proposition 3.3 est qu'il faut choisir une matrice Q *proche* de A.

3.3.2 Méthodes de Jacobi et de Gauss-Seidel

Le choix de la méthode de Jacobi consiste à prendre, pour matrice Q, la diagonale de A. Il est, en effet, aisé de voir qu'une matrice Q diagonale mènera à un système (3.31) facile à résoudre. De plus, si on veut prendre une matrice diagonale, celle qui est la *plus proche de* A est bien sûr la diagonale de A elle-même. On peut donc décrire la méthode de Jacobi, en dénotant par $D := \mathrm{diag}(A)$,

$$x^{(k)} = D^{-1}(D - A)x^{(k-1)} + D^{-1}b. \tag{3.34}$$

Si on explicite (3.34), cela donne

$$x_i^{(k)} = \sum_{j \neq i} \frac{-a_{ij}}{a_{ii}} x_j^{(k-1)} + \frac{b}{a_{ii}}. \tag{3.35}$$

Dans (3.35), on voit que, pour calculer $x^{(k)}$, on se sert uniquement des valeurs $x^{(k-1)}$ calculées à l'itération précédente. Par exemple, pour calculer $x_j^{(k)}$, on utilise $x_1^{(k-1)}, \ldots, x_n^{(k-1)}$. Cependant, si on suppose que le processus est convergent, lorsque l'on calcule $x_j^{(k)}$, on a déjà à sa disposition des valeurs

approchées plus précises de $\bar{x}_1, \ldots, \bar{x}_{j-1}$, à savoir $x_1^{(k)}, \ldots, x_{j-1}^{(k)}$. L'idée de la méthode de Gauss-Seidel est donc d'utiliser à l'itération k *toutes les nouvelles valeurs* qui ont déjà été calculées à l'itération k.

Il est également possible de voir la méthode de Gauss-Seidel en utilisant la structure (3.31). Nous avons dit plus haut qu'il fallait choisir une matrice Q facilement inversible, ou du moins dont le système qui résulte de son emploi puisse être résolu facilement. La méthode de Jacobi utilise une matrice diagonale. Un autre type de matrice qui mène à un système facile à résoudre est la matrice triangulaire. Comme de plus, nous préférons choisir une matrice Q proche de A, pour la méthode de Gauss-Seidel, nous choisissons de considérer la partie triangulaire inférieure de A. Si on décompose $A = L + D + U$, où L est la partie triangulaire inférieure de A, D sa diagonale, et U sa partie triangulaire supérieure, on obtient le processus

$$x^{(k)} = (L + D)^{-1}(L + D - A)x^{(k-1)} + (L + D)^{-1}b \tag{3.36}$$

dit algorithme de *Gauss-Seidel*. Bien que la description matricielle (3.36) semble plus compliquée que dans le cas de la méthode de Jacobi (3.34), la mise en oeuvre de l'algorithme de Gauss-Seidel est en réalité plus simple. A chaque iération, il n'est, en effet, pas nécessaire de *sauvegarder* les itérés précédents. Chaque nouvelle valeur calculée est automatiquement réutilisée pour les calculs suivants.

On peut, dans certains cas, améliorer la convergence de l'algorithme de Gauss-Seidel en introduisant un paramètre que l'on pourra modifier en fonction de la matrice A de départ. Pour ce faire, on introduit un facteur $0 < \omega < 2$. Au lieu de considérer $Q = L+D$, on va plutôt utiliser $Q = \omega L+D$. Dans le cas de $\omega > 1$, on parle de *surrelaxation*. Dans le cas de $\omega < 1$, on parle de *sousrelaxation*. En fait de manière équivalente, c'est comme si on considérait à chaque nouvelle itération le nouvel itéré

$$x^{(k)} = (1 - \omega)x^{(k-1)} + \omega g(x^{(k-1)})$$

où $g(x^{(k-1)})$ est l'itéré calculé selon la méthode de Gauss-Seidel, c'est-à-dire $g(x^{(k-1)}) = (L + D)^{-1}(L + D - A)x^{(k-1)} + (L + D)^{-1}b$. A chaque itération, on fait donc une sorte de *moyenne* entre l'itéré calculé par la méthode de Gauss-Seidel et l'ancien itéré. Dans le cas de $\omega = 1$, on retrouve exactement la méthode de Gauss-Seidel. Dans le cas de $\omega > 1$, on procède à la surrelaxation, c'est-à-dire que l'on tente de forcer une convergence plus forte du côté de

$g(x^{(k-1)})$. Dans le cas de la sousrelaxation, on freine le processus en gardant une partie provenant de l'ancien itéré. Nous verrons dans la section suivante que c'est la question de convergence qui détermine le facteur ω optimal.

Exemple 3.6 On souhaite résoudre le système

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 8 \\ -5 \end{pmatrix}.$$

Pour appliquer la méthode de Jacobi, on écrira à chaque itération $x^{(k)} = (I - D^{-1}A)x^{(k-1)} + D^{-1}b$, c'est-à-dire dans notre cas,

$$x^{(k)} = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 \end{pmatrix} x^{(k-1)} + \begin{pmatrix} \frac{1}{2} \\ \frac{8}{3} \\ -\frac{5}{2} \end{pmatrix}.$$

La Table 3.1 indique l'évolution de l'algorithme avec 0 comme point de départ.

Si, en revanche, on utilise l'algorithme de Gauss-Seidel, cela revient à considérer la matrice

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

et à effectuer l'itération $x^{(k)} = (I - Q^{-1}A)x^{(k-1)} + Q^{-1}b$, c'est-à-dire

$$x^{(k)} = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} \\ 0 & \frac{1}{12} & \frac{1}{6} \end{pmatrix} x^{(k-1)} + \begin{pmatrix} \frac{1}{2} \\ \frac{17}{6} \\ -\frac{13}{12} \end{pmatrix}.$$

La Table 3.2 indique l'évolution de l'algorithme avec 0 comme point de départ. On peut remarquer que la méthode converge environ deux fois plus vite que celle de Jacobi. ■

3.3.3 Convergence des méthodes itératives

Les méthodes itératives ne convergent pas dans tous les cas. Nous allons dans cette section montrer quels sont les cas où elles convergent. La Proposition 3.3 nous donne une idée de l'évolution de l'erreur. Si on considérait la

Itér. k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$Ax^{(k)} - b$		
1	0.0000	0.0000	0.0000	-1.0000	-8.0000	5.0000
2	0.5000	2.6667	-2.5000	-2.6667	2.0000	-2.6667
3	1.8333	2.0000	-1.1667	0.6667	-2.6667	0.6667
4	1.5000	2.8889	-1.5000	-0.8889	0.6667	-0.8889
5	1.9444	2.6667	-1.0556	0.2222	-0.8889	0.2222
6	1.8333	2.9630	-1.1667	-0.2963	0.2222	-0.2963
7	1.9815	2.8889	-1.0185	0.0741	-0.2963	0.0741
8	1.9444	2.9877	-1.0556	-0.0988	0.0741	-0.0988
9	1.9938	2.9630	-1.0062	0.0247	-0.0988	0.0247
10	1.9815	2.9959	-1.0185	-0.0329	0.0247	-0.0329
11	1.9979	2.9877	-1.0021	0.0082	-0.0329	0.0082
12	1.9938	2.9986	-1.0062	-0.0110	0.0082	-0.0110
13	1.9993	2.9959	-1.0007	0.0027	-0.0110	0.0027
14	1.9979	2.9995	-1.0021	-0.0037	0.0027	-0.0037
15	1.9998	2.9986	-1.0002	0.0009	-0.0037	0.0009
16	1.9993	2.9998	-1.0007	-0.0012	0.0009	-0.0012
17	1.9999	2.9995	-1.0001	0.0003	-0.0012	0.0003
18	1.9998	2.9999	-1.0002	-0.0004	0.0003	-0.0004
19	2.0000	2.9998	-1.0000	0.0001	-0.0004	0.0001
20	1.9999	3.0000	-1.0001	-0.0001	0.0001	-0.0001
21	2.0000	2.9999	-1.0000	0.0000	-0.0001	0.0000
22	2.0000	3.0000	-1.0000	-0.0000	0.0000	-0.0000

TABLE 3.1 – La méthode de Jacobi

Itér. k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$Ax^{(k)} - b$		
1	0.0000	0.0000	0.0000	-1.0000	-8.0000	5.0000
2	0.5000	2.8333	-1.0833	-2.8333	1.0833	0.0000
3	1.9167	2.9444	-1.0278	-0.1111	-0.0556	0.0000
4	1.9722	2.9815	-1.0093	-0.0370	-0.0185	0.0000
5	1.9907	2.9938	-1.0031	-0.0123	-0.0062	0.0000
6	1.9969	2.9979	-1.0010	-0.0041	-0.0021	0.0000
7	1.9990	2.9993	-1.0003	-0.0014	-0.0007	0.0000
8	1.9997	2.9998	-1.0001	-0.0005	-0.0002	0.0000
9	1.9999	2.9999	-1.0000	-0.0002	-0.0001	0.0000
10	2.0000	3.0000	-1.0000	-0.0001	-0.0000	0.0000

TABLE 3.2 – La méthode de Gauss-Seidel

résolution d'*équations*, on dirait qu'il faut que $(I - Q^{-1}A)$ soit inférieur à 1 en valeur absolue. Bien entendu cette condition ne peut pas être formulée telle quelle dans le cas de matrices. Cependant, la Proposition suivante indique que les module des *valeurs propres* de $I - Q^{-1}A$ doivent être inférieures à 1 si on veut avoir la convergence dans tous les cas.

Proposition 3.4 *Si les valeurs propres λ_i de $I - Q^{-1}A$ sont telles que $|\lambda_i| < 1$ pour tout i, alors le processus* (3.32) *converge vers la solution $\bar{x}$ du système $Ax = b$ pour tout point de départ $x^{(1)}$.*

Démonstration: Nous ne prouvons le théorème que dans le cas où tous les vecteurs propres de $I - Q^{-1}A$ sont linéairement indépendants. Nous dénotons par $v_1, \ldots, v_n$ ces n vecteurs propres indépendants et où v_i correspond à la valeur propre λ_i. On part d'une erreur $e^{(1)}$ à la première itération. Nous pouvons écrire cette erreur comme combinaison linéaire des vecteurs propres. On a donc

$$e^{(1)} = \alpha_1 v_1 + \cdots + \alpha_n v_n.$$

Par la Proposition 3.3, on a succesivement

$$\begin{aligned} e^{(k)} &= (I - Q^{-1}A)e^{(k-1)} \\ &= (I - Q^{-1}A)^2 e^{(k-2)} \\ &\vdots \\ &= (I - Q^{-1}A)^{k-1} e^{(1)} \\ &= (I - Q^{-1}A)^{k-1}(\alpha_1 v_1 + \cdots + \alpha_n v_n) \\ &= (I - Q^{-1}A)^{k-2}(\alpha_1 \lambda_1 v_1 + \cdots + \alpha_n \lambda_n v_n) \\ &\vdots \\ &= \alpha \lambda_1^{k-1} v_1 + \cdots + \alpha_n \lambda_n^{k-1} v_n \end{aligned}$$

Si on a $|\lambda_i| < 1$ pour tout i, on voit que le vecteur d'erreur $e^{(k)}$ tend vers 0 quand $k \to \infty$ puisque chaque terme de la somme tend individuellement vers 0. ■

La condition de convergence est donc que le *rayon spectral* de la matrice $I - Q^{-1}A$ soit inférieur à 1. La Proposition 3.4 reste vraie quel que soit le choix de la matrice Q. Il s'applique donc en particulier au cas des méthodes de Jacobi et de Gauss-Seidel. Cette condition n'est malheureusement pas évidente à vérifier. Nous allons cependant montrer qu'il existe une classe simple de matrices qui satisfont toujours cette propriété.

Proposition 3.5 *Soit $M = (m_{ij}) \in \mathbb{R}^{n \times n}$ une matrice qui a n vecteurs propres linéairement indépendants. Alors, pour chaque $i = 1, \ldots, n$, il existe $t \in \{1, \ldots, n\}$ tel que*

$$|\lambda_i| \leq \sum_{j=1}^{n} |m_{tj}|$$

où λ_i est une valeur propre de M.

Démonstration: Considérons une valeur propre λ de M et un vecteur propre associé v. Nous pouvons écrire la définition de la valeur propre. On a donc $Mv = \lambda v$. Considérons la composante de v la plus élevée en valeur absolue.

Dénotons-la par j. Nous avons donc

$$\begin{aligned}
\lambda v_j &= \sum_{k=1}^{n} m_{jk} v_k \\
|\lambda||v_j| &\leq \sum_{k=1}^{n} |m_{jk}||v_k| \\
&\leq \sum_{k=1}^{n} |m_{jk}||v_j| \\
|\lambda| &\leq \sum_{k=1}^{n} |m_{jk}|
\end{aligned}$$

■

On peut déduire de la Proposition 3.5, qu'une condition suffisante pour que toutes les valeurs propres soient de module inférieur à 1 est que la somme des valeurs absolues des éléments de chaque ligne soit inférieure à 1. Si on transpose cette condition à la matrice utilisée dans la méthode de Jacobi, on obtient la Proposition suivante.

Proposition 3.6 *On considère la système $Ax = b$. Si A est à diagonale dominante, c'est-à-dire si*

$$\sum_{j \neq i} |a_{ij}| < |a_{ii}|$$

pour tout $i = 1, \ldots, n$, alors la méthode de Jacobi (3.34) *converge pour tout point de départ $x^{(1)}$.*

Démonstration: La matrice intervenant dans la méthode de Jacobi est

$$I - D^{-1}A = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \cdots & -\frac{a_{2n}}{a_{22}} \\ \vdots & & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{pmatrix}.$$

Si on écrit la condition que chaque ligne doit avoir une somme inférieure à 1 (en valeur absolue), en se fiant à la Proposition 3.5, on obtient, pour la ligne i,

$$\sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|} < 1,$$

ce qui peut se réécrire comme

$$\sum_{j \neq i} |a_{ij}| < |a_{ii}|$$

pour tout $i = 1, \ldots, n$. ■

On peut montrer que cette condition suffit également pour assurer la convergence de l'algorithme de Gauss-Seidel.

3.4 Calcul de valeurs propres

Les valeurs propres d'une matrice sont un outil précieux pour l'ingénieur. Comme application principale, on notera le calcul des oscillations propres d'un système qu'il soit mécanique ou électrique. Les valeurs propres peuvent également donner l'information contenue dans un grand ensemble de données, telles les directions principales d'un nuage de points, ou l'information contenue dans un grand graphe. Et la liste ne s'arrête évidemment pas là. Rappelons tout d'abord la définition mathématique d'une valeur propre et d'un vecteur propre associé.

Définition 3.4 *Soit $A \in \mathbb{R}^{n \times n}$ une matrice réelle. La valeur $\lambda \in \mathbb{C}$ est une valeur propre de A s'il existe $v \in \mathbb{C}^n \setminus \{0\}$ (appelé vecteur propre associé à λ) tel que $Av = \lambda v$.*

La question du calcul des valeurs propres d'une matrice est fondamentale. Il est cependant peu pratique de devoir calculer les racines du polynôme caractéristique d'une matrice $\det(A - \lambda I)$ afin d'en connaître les valeurs propres. Dans cette section, nous allons montrer comment on peut obtenir très rapidement quelques valeurs propres et leurs vecteurs propres associés en appliquant une méthode itérative, connue sous le nom de *méthode de la puissance*. La question du calcul de toutes les valeurs propres d'une matrice, bien qu'importante, déborde du cadre de ce cours.

3.4.1 Méthode de la puissance

Soit une matrice réelle $A \in \mathbb{R}^{n \times n}$. Dans cette section, nous supposons que les valeurs propres de A sont telles que

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

et que chaque valeur propre λ_i a un vecteur propre associé $v^{(i)}$. La méthode de la puissance est une méthode itérative qui sert à trouver une approximation de λ_1. Notons l'importance de la condition de stricte inégalité $|\lambda_1| > |\lambda_2|$.

Nous supposons d'abord que les vecteurs propres de A forment une base linéaire de $\mathbb{C}^n$. Nous partons d'un vecteur complexe arbitraire $w^{(0)} \in \mathbb{C}^n$. Celui-ci peut s'écrire comme une combinaison linéaire des différents vecteurs propres de A. On a donc

$$w^{(0)} = \alpha_1 v^{(1)} + \alpha_2 v^{(2)} + \cdots + \alpha_n v^{(n)}, \tag{3.37}$$

avec $\alpha_i \in \mathbb{C}$ pour tout i. Supposons encore que $\alpha_1 \neq 0$. Nous procédons à présent aux différentes itérations de la méthode de la puissance. On calcule successivement

$$\begin{aligned} w^{(1)} &= Aw^{(0)} \\ w^{(2)} &= Aw^{(1)} = A^2 w^{(0)} \\ &\vdots \\ w^{(k)} &= Aw^{(k-1)} = A^k w^{(0)}. \end{aligned}$$

Si on reprend (3.37), on peut également écrire

$$\begin{aligned} w^{(k)} &= A^k w^{(0)} \\ &= A^k(\alpha_1 v^{(1)} + \cdots + \alpha_n v^{(n)}) \\ &= \alpha_1 \lambda_1^k v^{(1)} + \cdots + \alpha_n \lambda_n^k v^{(n)} \end{aligned}$$

en utilisant la propriété des vecteurs propres. Finalement, la dernière expression se réécrit

$$w^{(k)} = \lambda_1^k \left(\alpha_1 v^{(1)} + \alpha_2 (\frac{\lambda_2}{\lambda_1})^k v^{(2)} + \cdots + \alpha_n (\frac{\lambda_n}{\lambda_1})^k v^{(n)} \right).$$

Comme $|\lambda_1| > |\lambda_j|$ pour tout $j \neq 1$, tous les termes $(\frac{\lambda_j}{\lambda_1})^k$ tendent vers 0 quand k tend vers l'infini. A l'infini, la quantité $w^{(k)}$ tend donc vers la direction du vecteur propre dominant de A. En général évidemment, soit si $|\lambda_1| > 1$, la quantité tend vers l'infini, soit si $|\lambda_1| < 1$, la quantité tend vers 0, et il sera difficile de localiser la vraie direction. Dans la pratique, on procède donc à la normalisation des vecteurs après chaque étape :

$$z^{(k)} = \frac{w^{(k)}}{\|w^{(k)}\|}, \quad w^{(k+1)} = Az^{(k)}.$$

Le processus converge alors vers le vecteur propre dominant. On peut obtenir la valeur propre dominante en calculant à chaque étape

$$\sigma_k = z^{(k)^T} w^{(k+1)}$$

qui converge vers λ_1. En effet, on a

$$\sigma_k = \frac{z^{(k)^T} A z^{(k)}}{z^{(k)^T} z^{(k)}}$$

qui converge vers λ_1 si z_k converge vers $v^{(1)}$. Remarquons que le processus que nous venons de décrire converge vers le vecteur propre avec une vitesse dépendant du ratio $|\lambda_2|/|\lambda_1|$. Plus ce quotient est petit, plus la convergence est rapide. Le processus ne convergera donc pas vite pour des matrices pour lesquelles les deux valeurs propres dominantes sont proches. Une façon d'accélérer le processus est de travailler avec une matrice $B = A - mI$. En effet toutes les valeurs propres de B sont exactement $\lambda_i - m$. Si on connaît une approximation des valeurs propres, il peut être possible d'améliorer le ratio $|\lambda_2 - m|/|\lambda_1 - m|$. En règle générale, on n'a évidemment pas accès aux valeurs propres et il est donc difficile de trouver le m optimal. Remarquons qu'on peut aussi se servir de cette astuce pour calculer une autre valeur propre. En effet, si on applique la méthode de la puissance à $B := A - mI$, celle-ci va converger vers la valeur propre la plus éloignée de m.

3.4.2 Calcul de la valeur propre de plus petit module

Il est également possible de calculer la valeur propre de plus petit module (à condition qu'elle soit non nulle) avec la méthode de la puissance. En effet, si A est inversible, on peut voir que si λ est une valeur propre de A, alors on a

$$Ax = \lambda x \Longleftrightarrow x = A^{-1}(\lambda x) \Longleftrightarrow A^{-1}x = \frac{1}{\lambda}x.$$

Dès lors, si λ est une valeur propre de A, $\frac{1}{\lambda}$ est une valeur propre de A^{-1}. On en déduit aussi que si λ est la valeur propre de plus petit module de A, $\frac{1}{\lambda}$ sera la valeur propre de plus grand module de A^{-1}. On peut donc appliquer la méthode de la puissance de manière totalement similaire avec A^{-1} et écrire

$$z^{(k)} = \frac{w^{(k)}}{\|w^{(k)}\|}, \quad w^{(k+1)} = A^{-1} z^{(k)}.$$

Remarquons que dans la dernière expression, il n'est pas nécessaire d'effectuer la coûteuse opération de l'inversion de la matrice A. On peut tout à fait se contenter d'une factorisation LU de la matrice et résoudre le système $Aw^{(k+1)} = z^{(k)}$ à chaque itération. Finalement, on peut remarquer que l'on peut se servir de la méthode inverse de la puissance, associée à un changement de matrice $B = A - mI$ pour trouver la valeur propre qui est la plus proche d'un scalaire m.

3.4.3 Calcul d'autres valeurs propres

Supposons que l'on ait trouvé la valeur propre dominante λ_1 de A. On souhaite à présent calculer la deuxième valeur propre de plus grand module, à savoir λ_2.

La méthode que nous décrirons en premier lieu ne convient que pour une matrice A symétrique. Si λ_1 et v^1 sont respectivement la valeur propre et le vecteur propre déjà calculés, on forme la matrice

$$A_1 = A - \lambda_1 \, v^1 v^{1^T} \tag{3.38}$$

Comme la matrice A est symétrique, A_1 l'est aussi. On calcule que $A_1 v^1 = 0$ et que $A_1 v^j = \lambda_j v^j$ pour tout vecteur propre v^j associé à une valeur propre λ_j, $j = 2, 3, \ldots, n$. Par conséquent, A_1 a tous les vecteurs propres de A et toutes ses valeurs propres excepté λ_1 qui est remplacé par zéro.

Lorsque λ_2 et v^2 ont été calculés à partir de A_1, le processus peut être répété en formant $A_2 = A_1 - \lambda_2 v^2 v^{2^T}$, et ainsi de suite pour la détermination des valeurs propres et vecteurs propres restant.

Une autre méthode de déflation consiste à trouver une matrice non-singulière P telle que $Pv^1 = e_1$ où e_1 est le vecteur canonique $e_1 = (1,\ 0\ \cdots\ 0)^T$. On obtient alors de $Av^1 = \lambda_1 v^1$ que

$$\begin{aligned} PAP^{-1}Pv^1 &= \lambda_1 Pv^1 \\ (PAP^{-1})e_1 &= \lambda_1 e_1. \end{aligned}$$

La dernière égalité signifie que la matrice PAP^{-1}, qui a les mêmes valeurs propres que A, doit être de la forme

$$PAP^{-1} = \left(\begin{array}{c|c} \lambda_1 & b^T \\ \hline 0 & \\ 0 & X_1 \end{array}\right)$$

et la matrice d'ordre $(n-1)$ occupant le coin inférieur droit de PAP^{-1} possède donc bien les propriétés recherchées. Comme pour l'autre méthode de déflation, on peut répéter le processus en calculant X_2 à partir de X_1 une fois λ_2 et v^2 calculés.

3.4.4 Algorithme QR

La méthode que nous allons étudier maintenant s'est révélée dans la pratique comme l'une des plus efficaces pour la recherche de toutes les valeurs propres d'une matrice symétrique ou non-symétrique.

L'algorithme QR consiste à construire une suite de matrices $A = A_1,\ A_2,\ A_3, ...$ au moyen des relations

$$A_k = Q_k R_k,\ A_{k+1} = R_k Q_k\ , \ldots \tag{3.39}$$

où les matrices Q_k sont orthogonales et les matrices R_k sont triangulaires supérieures. Rappelons ce théorème d'algèbre concernant la décomposition QR découlant du principe d'orthogonalisation de Gram-Schmidt.

Théorème 3.6 *Toute matrice $A \in \mathbb{R}^{n\times n}$ carrée non singulière peut être écrite sous la forme $A = QR$ où R désigne une matrice non singulière triangulaire supérieure et où Q désigne une matrice unitaire, c'est-à-dire $QQ^T = Q^TQ = I$.*

On peut montrer que la matrice A_k tend vers une matrice triangulaire supérieure dont les éléments diagonaux sont les valeurs propres de A.

Théorème 3.7 (Convergence de l'algorithme QR.) *Si les valeurs propres $\lambda_i,\ i = 1,\ 2, ...,\ n$ de A satisfont les conditions*

$$|\,\lambda_1\,| > |\,\lambda_2\,| > ... > |\,\lambda_n\,| \tag{3.40}$$

alors la matrice A_k définie en (3.39) tend vers une matrice triangulaire supérieure dont les éléments diagonaux sont les valeurs propres de A, rangées dans l'ordre des modules décroissants.

Démonstration: Puisque les valeurs propres de A sont toutes différentes (et réelles), il existe une matrice non-singulière (et réelle) X, telle que :

$$A = X\,D\,X^{-1} \tag{3.41}$$

où

$$D := \text{diag } (\lambda_1, \lambda_2, \ldots, \lambda_n)$$

Définissons les matrices Q, R, L et U par les relations

$$X = QR \qquad X^{-1} = LU. \tag{3.42}$$

Les matrices R et U sont triangulaires supérieures, la matrice L est triangulaire inférieure avec tous ses éléments diagonaux égaux à 1 et la matrice Q est orthogonale. La matrice R est non-singulière puisque X l'est également. La décomposition QR existe toujours, tandis que la décomposition LU existe seulement si tous les mineurs principaux de X^{-1} sont non nuls.

Analysons maintenant en détail une étape de l'algorithme QR. On a

$$A_{k+1} = R_k Q_k = Q_k^T A_k Q_k. \tag{3.43}$$

A partir de cette dernière relation on déduit

$$A_{k+1} = P_k^T A P_k \tag{3.44}$$

où

$$P_k := Q_1 Q_2 \cdots Q_k. \tag{3.45}$$

Si l'on pose alors

$$U_k := R_k R_{k-1} \cdots R_1, \tag{3.46}$$

on calcule

$$\begin{aligned} P_k U_k &= Q_1 Q_2 \cdots Q_{k-1} (Q_k R_k) R_{k-1} \cdots R_2 R_1 \\ &= P_{k-1} A_k U_{k-1} \\ &= A P_{k-1} U_{k-1} \end{aligned} \tag{3.47}$$

où la dernière égalité est obtenue grâce à (3.44). On obtient alors par récurrence de (3.47)

$$P_k U_k = A^k. \tag{3.48}$$

Cette dernière relation montre que P_k et U_k sont les facteurs de la décomposition QR de la matrice A^k.

Si on reprend à présent (3.42), on a successivement d'après (3.44), (3.41) et (3.42)

$$\begin{aligned} A_{k+1} &= P_k^T A P_k \\ &= P_k^T X D X^{-1} P_k \\ &= P_k^T Q (R D R^{-1}) Q^T P_k. \end{aligned} \tag{3.49}$$

La matrice R étant triangulaire supérieure, la matrice R^{-1} l'est aussi et a pour éléments diagonaux les inverses des éléments diagonaux de R. Le produit RDR^{-1} est donc une matrice triangulaire supérieure dont la diagonale est égale à D. Il suffit donc pour établir le théorème de montrer que $P_k \rightarrow Q$.

Pour établir ce dernier fait, considérons la matrice A^k qui, étant donné (3.41) et (3.42) peut s'écrire sous les formes

$$A^k = XD^kX^{-1} = QRD^kLU = QR(D^kLD^{-k})D^kU. \tag{3.50}$$

On constate que la matrice D^kLD^{-k} est une matrice triangulaire inférieure dont les éléments diagonaux sont égaux à 1, tandis que l'élément (i,j) est égal à $l_{ij}(\lambda_i/\lambda_j)^k$ lorsque $i > j$, et nous pouvons donc écrire

$$D^kLD^{-k} = I + E_k \quad \text{où} \quad \lim_{k\rightarrow\infty} E_k = 0$$

L'équation (3.50) donne alors

$$\begin{aligned} A^k &= QR(I+E_k)D^kU \\ &= Q(I+RE_kR^{-1})RD^kU \\ &= Q(I+F_k)RD^kU \end{aligned}$$

où

$$\lim_{k\rightarrow\infty} F_k = 0$$

On peut alors effectuer la décomposition QR de la matrice $(I+F_k)$, soit

$$(I+F_k) = \tilde{Q}_k\tilde{R}_k$$

où $\tilde{Q}_k$ et $\tilde{R}_k$ tendent toutes deux vers I puisque F_k tend vers zéro. On a donc finalement

$$A^k = (Q\tilde{Q}_k)(\tilde{R}_kRD^kU). \tag{3.51}$$

Le premier des facteurs de (3.51) est orthogonal et le second est triangulaire supérieur : on a donc bien obtenu une décomposition QR de A^k. Mais cette décomposition est unique puisque A^k est non-singulière. Comparant alors (3.51) et (3.48) on a $P_k = Q\tilde{Q}_k$ et donc $P_k \rightarrow Q$ puisque $\tilde{Q}_k \rightarrow I$. ∎

On a dû supposer pour pouvoir démontrer ce théorème que X^{-1} a une décomposition LU. Au cas ou cette décomposition n'existe pas, on sait qu'il existe une matrice de permutation P telle que l'on puisse écrire $PX^{-1} = LU$.

On peut alors montrer que dans ce cas la matrice P_k tend vers le facteur Q obtenu par la décomposition QR de la matrice XP^T. La matrice A_k tend toujours vers une matrice triangulaire supérieure dont les éléments diagonaux restent les valeurs propres de A, mais celles-ci ne seront plus rangées par ordre des modules croissants.

On a pu constater dans la démonstration du théorème 3.7 que la convergence de l'algorithme QR dépend essentiellement des rapports λ_i/λ_j. Tout comme pour la méthode de la puissance, la convergence sera donc ralentie en cas de valeurs propres voisines. On modifiera alors l'algorithme de base de la façon suivante :

$$A_k - \alpha_k I = Q_k R_k \ , \qquad R_k Q_k + \alpha_k I = A_{k+1} \ , \qquad k = 1,\ 2,\ 3, ... \tag{3.52}$$

où les facteurs α_k sont des *facteurs d'accélération de la convergence*. De façon analogue à (3.44) et (3.48) on aura maintenant

$$A_{k+1} = P_k^T A P_k \tag{3.53}$$

et

$$P_k U_k = (A - \alpha_0 I)(A - \alpha_1 I)...(A - \alpha_k I). \tag{3.54}$$

Les valeurs propres de A_{k+1} restent donc toujours les mêmes que celles de A, tandis que la convergence dépendra maintenant des rapports $(\lambda_i - \alpha_k)/(\lambda_j - \alpha_k)$. Les points délicats de la mise en œuvre de tels algorithmes sont évidemment dans les choix des facteurs de convergence. Nous renvoyons à la littérature spécialisée en ce qui concerne la description de stratégies des choix des α_k.

Dans le cas de valeurs propres identiques, une analyse détaillée montre qu'en fait, la convergence n'est pas affectée.

Le cas de valeurs propres complexes conjuguées est un peu plus délicat. Il est bien certain que des transformations de similitudes réelles ne peuvent pas fournir une matrice triangulaire avec les valeurs propres sur la diagonale. On peut montrer que la matrice A_k tendra vers une matrice "presque" triangulaire supérieure : la matrice limite contiendra des blocs d'ordre deux, centrés sur la diagonale principale, et dont les valeurs propres seront les valeurs propres complexes conjuguées de la matrice A.

Pour une matrice pleine, une seule itération de l'algorithme QR nécessitera de l'ordre de $10n^3/3$ opérations, ce qui est beaucoup trop élevé. En pratique, pour pouvoir utiliser l'algorithme, il faudra transformer la matrice en une

forme dite de Hessenberg pour laquelle la factorisation QR est beaucoup moins chère. Le détail de l'implémentation sort du cadre de ce cours et nous nous référons à la littérature spécialisée pour les détails d'implémentation.

3.5 Optimisation linéaire

L'ingénieur ou l'informaticien est souvent confronté à la recherche de la meilleure solution d'un problème donné. Lorsque cela est possible, il est souvent intéressant de modéliser les différentes décisions possibles d'un problème comme des variables mathématiques devant répondre à une série de contraintes. *Prendre la meilleure décision* revient alors à trouver le maximum ou le minimum que peut prendre une fonction si on lui affecte des variables satisfaisant les contraintes. Cette approche générale est appelée *optimisation* et la branche mathématique qui s'occupe de la théorie et des algorithmes pour résoudre les problèmes d'optimisation est appelée *programmation mathématique*. Nous développons à présent un exemple qui illustre cette démarche.

Exemple 3.7 La compagnie Steel a reçu une commande de 500 tonnes d'acier qui doit être utilisé en construction navale. L'acier doit avoir les caractéristiques suivantes.

Elément chimique	Valeur Minimum	Valeur Maximum
Carbone (C)	2	3
Cuivre (Cu)	0.4	0.6
Manganèse (Mn)	1.2	1.65

La compagnie possède différents matériaux de base qu'elle peut utiliser pour la production de son acier. La table suivante renseigne les valeurs de chacun de ces matériaux, leurs quantités disponibles et leurs prix.

Matériau	C %	Cu %	Mn %	Disp. (T)	Coût (€/T)
Alliage de fer 1	2.5	0	1.3	400	200
Alliage de fer 2	3	0	0.8	300	250
Alliage de fer 3	0	0.3	0	600	150
Alliage de cuivre 1	0	90	0	500	220
Alliage de cuivre 2	0	96	4	200	240
Alliage d'aluminium 1	0	0.4	1.2	300	200
Alliage d'aluminium 2	0	0.6	0	250	165

L'objectif est de déterminer la composition optimale de l'acier qui permet de minimiser le coût de celui-ci.

La démarche pour résoudre un tel problème est de proposer une formulation mathématique de celui-ci. Pour ce faire, nous allons définir des *variables de décision*, des contraintes à respecter et un objectif à maximiser ou minimiser.

Variables de décision
Les décisions possibles dans ce problème portent sur les quantités de chaque alliage qui se retrouveront dans la composition finale. Nous pouvons donc considérer les variables de décision suivantes :

x_{Fi} : quantité (T) de l'alliage de fer i entrant dans la composition
x_{Ci} : quantité (T) de l'alliage de cuivre i entrant dans la composition
x_{Ai} : quantité (T) de l'alliage d'aluminium i entrant dans la composition

Fonction objectif
L'objectif dans ce problème est de minimiser le coût de production. On peut formuler cela aisément en exprimant l'objectif comme

$$\min 200x_{F1} + 250x_{F2} + 150x_{F3} + 220x_{C1} + 240x_{C2} + 200x_{A1} + 165x_{A2}.$$

On remarque que la fonction objectif est *linéaire*.

Contraintes
Une série de contraintes sont imposées au problème. Nous allons maintenant modéliser chacune d'elles comme une inégalité mathématique. Il y a principalement trois types de contraintes. Premièrement, nous avons les contraintes de disponibilité de chaque matériau. Nous avons également les contraintes sur la composition finale de l'acier dans les trois éléments chimiques. Finalement nous avons la contrainte décrivant la quantité totale d'acier à produire. Commençons par l'expression de la disponibilité de chaque alliage. Cela peut s'exprimer aisément comme

$$x_{F1} \leq 400, x_{F2} \leq 300, x_{F3} \leq 600, x_{C1} \leq 500, x_{C2} \leq 200, x_{A1} \leq 300, x_{A2} \leq 250.$$

■

Les contraintes de satisfaction des pourcentages de chaque élément chimique seront obtenues en réalisant les combinaisons linéaires

$$\begin{aligned} 2 &\leq 2.5x_{F1} + 3x_{F2} &&\leq 3 \\ 0.4 &\leq 0.3x_{F3} + 90x_{C1} + 96x_{C2} + 0.4x_{A1} + 0.6x_{A2} &&\leq 0.6 \\ 1.2 &\leq 1.3x_{F1} + 0.8x_{F2} + 4x_{C2} + 1.2x_{A1} &&\leq 1.65. \end{aligned}$$

Nous voyons que les contraintes ne comportent que des combinaisons linéaires des variables. Elles sont donc également *linéaires*. La contrainte concernant la production totale d'acier s'exprime comme la somme de l'utilisation de tous les alliages c'est-à-dire

$$x_{F1} + x_{F2} + x_{F3} + x_{C1} + x_{C2} + x_{A1} + x_{A2} = 500.$$

Remarquons finalement que toutes les variables, pour avoir un sens, doivent être positives. On peut résumer le problème d'optimisation obtenu en utilisant la formulation de programmation mathématique

$$\begin{array}{rlllllll} \min & 200x_{F1}+250x_{F2}+150x_{F3}+220x_{C1}+240x_{C2}+200x_{A1}+165x_{A2} & & & & & & \\ \text{s.c.q.} & 2.5x_{F1}+ 3x_{F2} & & & & & & \geq 2 \\ & 2.5x_{F1}+ 3x_{F2} & & & & & & \leq 3 \\ & & 0.3x_{F3}+ 90x_{C1}+ 96x_{C2}+ 0.4x_{A1}+ 0.6x_{A2} & & & & & \geq 0.4 \\ & & 0.3x_{F3}+ 90x_{C1}+ 96x_{C2}+ 0.4x_{A1}+ 0.6x_{A2} & & & & & \leq 0.6 \\ & 1.3x_{F1}+ 0.8x_{F2} & & & + 4x_{C2}+ 1.2x_{A1} & & & \geq 1.2 \\ & 1.3x_{F1}+ 0.8x_{F2} & & & + 4x_{C2}+ 1.2x_{A1} & & & \leq 1.65 \\ & x_{F1} & & & & & & \leq 400 \\ & & x_{F2} & & & & & \leq 300 \\ & & & x_{F3} & & & & \leq 600 \\ & & & & x_{C1} & & & \leq 500 \\ & & & & & x_{C2} & & \leq 200 \\ & & & & & & x_{A1} & \leq 300 \\ & & & & & & & x_{A2} \leq 250 \\ & x_{F1}, & x_{F2}, & x_{F3}, & x_{C1}, & x_{C2}, & x_{A1}, & x_{A2} \in \mathbb{R}_+. \end{array}$$

Lorsque la fonction objectif et les contraintes sont linéaires, on parle de *programmation linéaire*. Ce genre de problème possède de très belles propriétés

théoriques et peut être résolu de manière très efficace. Dans cette section, nous allons étudier l'*algorithme du simplexe* qui est un algorithme efficace pour résoudre les problèmes de programmation linéaire. L'autre famille d'algorithmes efficaces pour résoudre les programmes linéaires est composé des méthodes de points intérieurs que nous n'aborderons pas dans ce cours.

3.5.1 Forme standard de la programmation linéaire

Le modèle de l'Exemple 3.7 est un exemple de programme linéaire. Nous allons à présent voir qu'il existe plusieurs façons équivalentes de représenter les programmes linéaires. Rappelons qu'un programme linéaire est un problème de la forme

$$\begin{aligned} \min\ & g(x) \\ \text{s.c.q.}\ & f_I(x) \geq 0 \\ & f_E(x) = 0 \\ & x \in \mathbb{R}^n \end{aligned}$$

où $g : \mathbb{R}^n \to \mathbb{R}$ est une fonction linéaire que l'on appelle la fonction objectif, $f_I : \mathbb{R}^n \to \mathbb{R}^{m_1}$ est une fonction affine définissant les contraintes d'inégalité et $f_E : \mathbb{R}^n \to \mathbb{R}^{m_2}$ est une fonction affine définissant les contraintes d'égalité. En utilisant le formalisme matriciel, on obtient une forme générique du type

$$\begin{aligned} \min\ & c^T x && (3.55)\\ \text{s.c.q.}\ & A^{\geq} x \geq b^{\geq} && (3.56)\\ & A^{\leq} x \leq b^{\leq} && (3.57)\\ & A^{=} x = b^{=} && (3.58)\\ & x \in \mathbb{R}^n. && (3.59) \end{aligned}$$

Nous allons maintenant voir comment on peut écrire un même problème sous différentes formes pratiquement équivalentes. Ceci nous permettra de traiter tous les problèmes de programmation linéaire sous une même forme standard plus simple que la forme (3.55)-(3.59).

Observation 3.1 *Tout problème de maximisation peut être ramené à un problème de minimisation et inversement.*

Démonstration: En effet, si on cherche le point x qui maximise $g(x)$ sur un ensemble X, on peut remarquer que le même point x *minimise* $-g(x)$ sur l'ensemble X. Dès lors les deux problèmes

$$\begin{array}{ll} \min c^T x & \max\ -c^T x \\ \text{s.c.q } x \in X & \text{s.c.q } x \in X \end{array}$$

ont les mêmes ensembles de solutions optimales atteignant des valeurs opposées de la fonction objectif. ■

Observation 3.2 *Toute contrainte d'égalité peut être exprimée de manière équivalente par deux inégalités.*

Démonstration: Il suffit de voir que les deux ensembles

$$\begin{aligned} X^{=} &= \{x \in \mathbb{R}^n \mid A^{=}x = b^{=}\} \text{ et} \\ X^{\leq,\geq} &= \{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}, A^{=}x \geq b^{=}\} \end{aligned}$$

sont égaux. ■

Dans l'autre direction, on peut également exprimer une contrainte d'inégalité par une contrainte d'égalité, mais on doit l'exprimer dans un autre espace (en rajoutant une variable).

Observation 3.3 *Toute contrainte d'inégalité peut être exprimée de manière équivalente dans un espace étendu, c'est-à-dire avec une variable de plus, appelée* variable d'écart *par une contrainte d'égalité.*

Démonstration: Considérons $A^{\leq} \in \mathbb{R}^{m \times n}$. Dans le cas d'une contrainte $\leq$, on peut voir que les deux ensembles

$$\begin{aligned} X^{\leq} &= \{x \in \mathbb{R}^n \mid A^{\leq}x \leq b^{\leq}\} \text{ et} \\ X^{s,\leq} &= \{x \in \mathbb{R}^n \mid \exists s \in \mathbb{R}^m_+ \text{ tel que } A^{\leq}x + Is = b^{\leq}\} \end{aligned}$$

sont égaux. Dans le cas d'une contrainte $\geq$ et pour une matrice $A^{\geq} \in \mathbb{R}^{m \times n}$, on peut voir que les deux ensembles

$$\begin{aligned} X^{\geq} &= \{x \in \mathbb{R}^n \mid A^{\geq}x \geq b^{\geq}\} \text{ et} \\ X^{s,\geq} &= \{x \in \mathbb{R}^n \mid \exists s \in \mathbb{R}^m_+ \text{ tel que } A^{\geq}x - Is = b^{\geq}\} \end{aligned}$$

sont égaux. ■

Nous terminons ce catalogue d'équivalences en montrant que l'on peut toujours se ramener à un problème ne contenant que des variables restreintes à des valeurs positives. Cela dit, il ne s'agit cette fois pas d'une stricte équivalence.

Observation 3.4 *Tout problème linéaire contenant des variables non restreintes quant au signe peut se ramener à un problème linéaire ne contenant que des variables dont les valeurs doivent être positives.*

Démonstration: Considérons le problème $\max\{c^T x + \bar{c}y \mid Ax + \bar{A}y \leq b, x \in \mathbb{R}^n_+, y \in \mathbb{R}\}$ où la variable y peut être aussi bien de signe positif que négatif et tentons d'écrire un problème associé où toutes les variables doivent être positives. On peut voir que si on introduit les deux variables $y_+ \in \mathbb{R}_+$ et $y_- \in \mathbb{R}_+$, on peut toujours exprimer $y = y_+ - y_-$. Dès lors si on résout le problème $\max\{c^T x + \bar{c}y_+ - \bar{c}y_- \mid Ax + \bar{A}y_+ - \bar{A}y_- \leq b, x \in \mathbb{R}^n_+, y_+, y_- \in \mathbb{R}\}$, toute solution optimale de ce problème peut se ramener à une solution optimale du problème initial en écrivant simplement $y = y_+ - y_-$. Remarquons simplement que les deux problèmes ne sont pas strictement équivalents. ■

Grâce aux quatre observations précédentes, nous sommes maintenant prêts à décrire la forme standard de la programmation linéaire.

Théorème 3.8 *Tout problème de programmation linéaire peut être ramené à une forme standard où la fonction objectif est une minimisation, où toutes les contraintes sont des égalités et où toutes les variables doivent prendre des valeurs positives. Mathématiquement, on peut donc ramener tout problème linéaire à la forme standard*

$$\begin{aligned} \min \ & c^T x \\ s.c.q. \ & Ax = b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

Exemple 3.8 Dans cet exemple, nous mettons en pratique les différentes observations citées plus haut de façon à mettre un problème sous sa forme standard. Considérons donc le problème

$$\max \ 2x_1 + 3x_2 \tag{3.60}$$

$$\text{s.c.q.} \ x_1 - x_2 \geq 1 \tag{3.61}$$

$$2x_1 + 3x_2 \leq 7 \tag{3.62}$$

$$x_1 \in \mathbb{R}_-, x_2 \in \mathbb{R}. \tag{3.63}$$

Nous allons le transformer en forme standard. Considérons tout d'abord la fonction objectif (3.60). Il s'agit ici d'un maximum. Pour se ramener à un minimum, il suffit de considérer la fonction opposée à minimiser : min $-2x_1 - 3x_2$. Considérons à présent les contraintes (3.61),(3.62) qui sont des inégalités. Elles peuvent être aisément transformées en contraintes d'égalité en y ajoutant des variables d'écart respectivement $s_1, s_2 \in \mathbb{R}_+$. On réécrit donc les contraintes comme $x_1 - x_2 - s_1 = 1$ et $2x_1 + 3x_2 + s_2 = 7$. Finalement, on doit traiter les bornes (ou l'absence de bornes) des variables. La variable x_1 est bornée mais dans le mauvais sens. On introduit donc $\bar{x}_1 := -x_1$. Remarquons qu'il suffit ensuite simplement de prendre l'opposé des coefficients de x_1 dans l'objectif et les contraintes pour obtenir les coefficients de $\bar{x}_1$ dans l'objectif et les contraintes. Si on considère x_2 qui n'a pas de bornes, on introduit deux variables x_2^+ et x_2^- et on écrit $x_2 = x_2^+ - x_2^-$. Les coefficients de x_2^+ sont ceux de x_2, quant à ceux de x_2^-, ils sont obtenus comme l'opposé de ceux de x_2. En résumé, on obtiendra donc le problème (3.60)-(3.63) mis sous forme standard

$$\begin{array}{lrcl} \max & -2\bar{x}_1+3x_2^+-3x_2^- & & \\ \text{s.c.q.} & -\bar{x}_1- \ x_2^+ + \ x_2^- - s_1 & & = 1 \\ & -2\bar{x}_1+3x_2^+-3x_2^- & +s_2 & = 7 \\ & \bar{x}_1, \quad x_2^+, \quad x_2^-, & s_1, \ s_2 & \in \mathbb{R}_+. \end{array}$$

■

Dans la suite, lorsque nous parlerons d'un problème de programmation linéaire, il pourra s'agir aussi bien de contraintes de la forme $Ax \leq b$ que $Ax \geq b$ ou de la forme standard $Ax = b, x \in \mathbb{R}_+^n$. L'ensemble des solutions *réalisables* d'un programme linéaire c'est-à-dire l'ensemble des points satisfaisant l'ensemble des contraintes est un *polyèdre*. Nous allons exprimer quelques propriétés géométriques utiles de ces polyèdres qui nous permettront d'exprimer l'algorithme du simplexe.

3.5.2 Géométrie des polyèdres

En trois dimensions, un polyèdre est un solide dont toutes les "faces" sont "rectilignes". On peut généraliser cette intuition à toute dimension et interpréter géométriquement un polyèdre comme étant l'intersection des en-

sembles réalisables d'un ensemble fini de contraintes linéaires. On utilisera donc la définition suivante.

Définition 3.5 *Un polyèdre en dimension n est l'ensemble $\{x \in \mathbb{R}^n \mid Ax \leq b\}$ pour une matrice $A \in \mathbb{R}^{m \times n}$ et un vecteur $b \in \mathbb{R}^m$ donnés.*

Nous avons décidé de définir un polyèdre comme étant l'ensemble réalisable d'un système d'inégalités $\leq$. Comme nous l'avons discuté dans la section précédente, nous pouvons également définir un polyèdre (en représentation standard) comme étant l'ensemble $\{x \in \mathbb{R}^n_+ \mid Ax = b, x \geq 0\}$.

Exemple 3.9 Considérons l'ensemble

$$\begin{aligned} X = \{x \in \mathbb{R}^2 \mid & -x_1 + 2x_2 \leq 1 && (a)\\ & -x_1 + \ x_2 \leq 0 && (b)\\ & 4x_1 + 3x_2 \leq 12 && (c)\\ & x_1, \quad x_2 \geq 0 \quad \}. && (d) \end{aligned}$$

En deux dimensions, un polyèdre est un polygone convexe où chaque côté est représenté par une inégalité de l'ensemble. Le polyèdre (polygone) X est la zone hachurée représentée à la Figure 3.1. ■

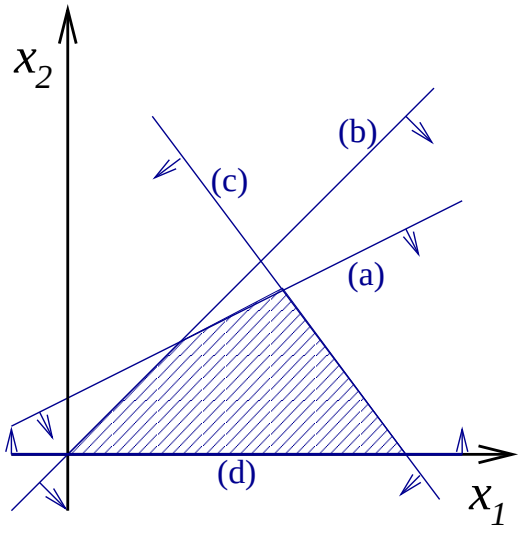

FIGURE 3.1 – Le polyèdre de l'Exemple 3.9

La convexité d'un polyèdre est une propriété fondamentale.

Définition 3.6 *Un ensemble $S \subseteq \mathbb{R}^n$ est* convexe *si pour tout $x, y \in S$ et tout $\lambda \in [0,1]$, on a également que $\lambda x + (1-\lambda)y \in S$.*

Géométriquement la convexité implique que si on considère deux points d'un ensemble convexe, alors le segment qui les lie est également entièrement inclus dans l'ensemble. Le segment liant deux points est ce qu'on appelle une combinaison convexe Ce concept peut être généralisé à plusieurs points.

Définition 3.7 *Soit $x^{(1)}, \ldots, x^{(k)} \subseteq \mathbb{R}^n$.*

(i) La combinaison linéaire $\sum_{i=1}^{k} \lambda_i x^{(i)}$, où $\sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0$, est appelée combinaiason convexe *des vecteurs $x^{(1)}, \ldots, x^{(k)}$.*

*(ii) L'*enveloppe convexe *des vecteurs $x^{(1)}, \ldots x^{(k)}$ est l'ensemble de toutes les combinaisons convexes de $x^{(1)}, \ldots, x^{(k)}$.*

Nous sommes maintenant en mesure de prouver le fait que tout polyèdre tel que nous l'avons défini est un ensemble convexe.

Proposition 3.7 *(i) Le demi-espace $\{x \in \mathbb{R}^n \mid a^T x \leq b\}$ est un ensemble convexe.*

(ii) L'intersection d'un nombre quelconque d'ensembles convexes est convexe.

(iii) Un polyèdre est un ensemble convexe.

Démonstration: (i) Soit $x, y \in \mathbb{R}^n$ tel que $a^T x \leq b$ et $a^T y \leq b$. Par linéarité, nous déduisons ensuite que $a^T(\lambda x + (1-\lambda)y) = \lambda(a^T x) + (1-\lambda)(a^T y) \leq \lambda b + (1-\lambda)b$ ce qui nous permet de conclure que la combinaison convexe de x et y satisfait la même contrainte.
(ii) Soit X_i, $i \in I$ des ensembles convexes et considérons $x, y \in S_i$ pour tout $i \in I$. Par conséquent par convexité de tous les ensembles S_i, on en déduit que $\lambda x + (1-\lambda)y \in S_i$ pour tout $i \in I$. Il s'ensuit dès lors que la combinaison convexe appartient aussi à l'intersection des ensembles.
(iii) Comme un polyèdre est une intersection de demi-espaces, il s'ensuit qu'il s'agit d'un ensemble convexe. ∎

Nous allons maintenant caractériser de trois façons équivalentes les "coins" d'un polyèdre. Nous verrons que ces points sont en réalité les candidats naturels lorsque l'on veut résoudre un problème de programmation linéaire. Nous utiliserons par la suite cette caractérisation pour décrire l'algorithme du simplexe.

Définition 3.8 *Soit $P \subseteq \mathbb{R}^n$ un polyèdre. Un point $x \in P$ est un* point extrême *de P s'il n'existe pas deux points $y, z \in P$ tels que x est une combinaison convexe (stricte) de y et z, c'est-à-dire tels qu'il existe $0 < \lambda < 1$ et $x = \lambda y + (1 - \lambda)z$.*

La définition de point extrême est relativement géométrique et indique que ceux-ci doivent se trouver aux "coins" d'un polyèdre. La définition suivante s'avère équivalente et est d'une grande importance du point de vue de la programmation linéaire.

Définition 3.9 *Soit $P \subseteq \mathbb{R}^n$ un polyèdre. Un point $x \in P$ est un* sommet *de P si il existe une fonction objectif $c \in \mathbb{R}^n$ telle que $c^T x < c^T y$ pour tout $y \in P \setminus \{x\}$.*

Cette définition indique qu'un point est un sommet s'il existe une fonction objetif pour laquelle x est la solution optimale du programme linéaire correspondant. Ceci nous donne déjà une idée de la méthode de résolution des programmes linéaires. Nous allons explorer les différents sommets afin de trouver celui qui minimise la fonction objectif. Cette caractérisation des sommets n'est malheureusement pas très utile algébriquement. C'est en se servant de la définition de solution de base que nous pourrons calculer explicitement les différents sommets. Avant de l'introduire, nous nécessitons encore la définition suivante.

Définition 3.10 *Soit un ensemble d'égalités ou d'inégalités $(a^{(i)})^T x \begin{matrix} \leq \\ = \\ \geq \end{matrix} b_{(i)}$, $i \in N$. Soit $x \in \mathbb{R}^n$ un point satisfaisant les contraintes avec égalité $(a^{(i)})^T x = b_{(i)}$ pour $i \in M \subseteq N$. On dit que les contraintes de l'ensemble M sont* actives *ou* serrées *en x.*

Définition 3.11 *Soit $P \subseteq \mathbb{R}^n$ un polyèdre défini par des contraintes $(a^{(i)})^T x \leq b_{(i)}, i \in M_{\leq}$ et $(a^{(i)})^T x = b_{(i)}, i \in M_{=}$. Un point $x \in \mathbb{R}^n$ est une* solution de base *si*

(i) x satisfait toutes les contraintes d'égalité $(a^{(i)})^T x = b_{(i)}, i \in M_{=}$,

(ii) n contraintes linéairement indépendantes sont serrées en x.

Un point $x \in \mathbb{R}^n$ est une solution de base réalisable *si x est une solution de base et satisfait aussi toutes les contraintes qui ne sont pas serrées. En d'autres termes, $x \in P$.*

Exemple 3.10 On peut aisément interpréter géométriquement le concept de solution de base et de solution de base réalisable dans le cas bidimensionnel. Reprenons l'ensemble de l'Exemple 3.9 à savoir

$$\begin{aligned} X = \{x \in \mathbb{R}^2 \mid & -x_1+2x_2 \leq 1 && (a)\\ & -x_1+\ x_2 \leq 0 && (b)\\ & 4x_1+3x_2 \leq 12 && (c)\\ & x_1,\quad x_2 \geq 0 \quad \}. && (d) \end{aligned}$$

représenté à la Figure 3.2. Puisque nous travaillons en 2 dimensions, une so-

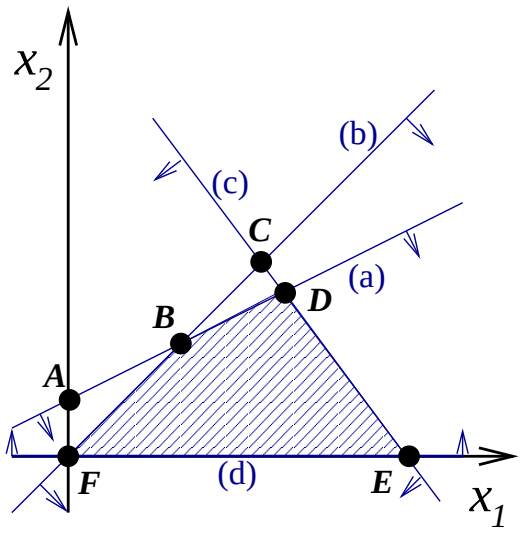

FIGURE 3.2 – Un polyèdre en deux dimensions et ses solutions de base

lution de base peut être obtenue en considérant deux inégalités linéairement indépendantes satisfaites avec égalité. Nous obtenons donc que les solutions de base sont les intersections géométriques de chaque paire d'inégalités. On obtient donc les points A, B, C, D, E, F (entre autres). Remarquons que les points B, D, E, F appartiennent au polyèdre et sont donc des solutions de base réalisables tandis que A et C sont des solutions de base non réalisables. Remarquons finalement que F est l'intersection non pas de deux mais de trois contraintes (dont évidemment seulement deux sont linéairement indépendantes). On parle dans ce cas de dégénérescence. Il s'agit d'un fait qui peut poser cer-

tains problèmes théoriques mais qui est extrêmement fréquent en pratique. ■

Nous allons maintenant prouver que les trois définitions (point extrême, sommet et solution de base réalisable) correspondent aux mêmes points. Ces mêmes points seront les points que nous investiguerons lorsque nous rechercherons une solution optimale d'un programme linéaire.

Théorème 3.9 *Soit $P \subseteq \mathbb{R}^n$ un polyèdre non vide et $x \in P$. Les trois propositions suivantes sont équivalentes :*

(i) x est un sommet de P,

(ii) x est un point extrême de P,

(iii) x est une solution de base réalisable de P.

Démonstration: Supposons que

$$P = \{x \in \mathbb{R}^n \mid (a^{(i)})^T x \geq b_{(i)}, i \in M^{\geq} \\ (a^{(i)})^T x = b_{(i)}, i \in M^{=} \quad \}.$$

où $M^{\geq}$ représente l'ensemble des contraintes d'inégalité et $M^{=}$ représente l'ensemble des contraintes d'égalité.
$(i) \Rightarrow (ii)$ Comme x est un sommet de P, en appliquant la définition, on en déduit qu'il existe $c \in \mathbb{R}^n$ tel que $c^T x < c^T y$ pour tout $y \in P \setminus \{x\}$. Supposons maintenant par l'absurde que x n'est pas un point extrême. Alors il existe $y, z \in P \setminus \{x\}$, $0 < \lambda < 1$ tels que $x = \lambda y + (1-\lambda)z$. Mais alors $c^T x = \lambda c^T y + (1-\lambda) c^T z > c^T x$ puisque $c^T y > c^T x$ et $c^T z > c^T x$, ce qui est une contradiction.

$(ii) \Rightarrow (iii)$ Nous allons montrer la contraposée. Supposons donc que x n'est pas une solution de base réalisable, nous allons montrer que x ne peut pas être un point extrême. Si x n'est pas une solution de base réalisable, alors au plus $n-1$ contraintes linéairement indépendantes sont serrées pour x. Appelons I l'ensemble des indices des contraintes qui sont serrées pour x. Il existe alors une direction $d \in \mathbb{R}^n$ telle que $(a^{(i)})^T d = 0$ pour tout $i \in I$. Remarquons qu'il existe $\epsilon > 0$ tel que $x + \epsilon d \in P$ et $x - \epsilon d \in P$. En effet, $(a^{(i)})^T (x \pm \epsilon d) = b_{(i)}$ pour $i \in I$. De plus $(a^{(i)})^T x > b_{(i)}$ pour $i \notin I$, ce qui implique le choix possible de $\epsilon > 0$. Nous obtenons donc finalement la négation de (iii) puisque $x = \frac{1}{2}(x + \epsilon d) + \frac{1}{2}(x - \epsilon d)$, ce qui montre que x

peut s'écrire comme une combinaison convexe de deux solutions réalisables distinctes de x qui n'est donc pas un point extrême.

$(iii) \Rightarrow (i)$ Supposons que x soit une solution de base réalisable et soit I l'ensemble des indices des contraintes qui sont serrées pour x, c'est-à-dire $(a^{(i)})^T x = b_{(i)}$ pour tout $i \in I$. Nous allons montrer qu'en définissant $c := \sum_{i \in I} a^{(i)}$, on satisfait la propriété de sommet pour x. Remarquons que l'on a $c^T x = \sum_{i \in I} (a^{(i)})^T x = \sum_{i \in I} b_{(i)}$. Maintenant considérons un point $y \in P \backslash \{x\}$. Remarquons que $(a^{(i)})^T y \geq b_{(i)}$ pour tout $i \in I$ puisque y doit satisfaire toutes les contraintes définissant le polyèdre. De plus, comme il y a n contraintes linéairement indépendantes dans les contraintes de I, nous savons que le système linéaire qui consiste à rendre toutes les contraintes serrées a une solution unique. Cette solution est donc forcément x. En conséquence, il existe une contrainte j telle que $(a^{(j)})^T y > b_{(j)}$ et dès lors $c^T y = \sum_{i \in I} (a^{(i)})^T y > \sum_{i \in I} b_{(i)}$ ce qui est bien le résultat que nous voulions prouver. ■

Nous avons caractérisé de plusieurs façons différentes le fait qu'un point x est un "coin" du polyèdre P. Nous avons toujours travaillé jusqu'à présent avec des polyèdres génériques. Nous allons maintenant plus précisément traiter les solutions de base réalisables d'un polyèdre lorsque celui-ci est en forme standard. Rappelons qu'un polyèdre en forme standard s'écrit $P = \{x \in \mathbb{R}^n_+ \mid Ax = b, x \geq 0\}$ où $A \in \mathbb{R}^{m \times n}$. On peut montrer que l'on peut supposer que les lignes de A sont linéairement indépendantes (ce qui n'est pas le cas en forme non-standard). On en déduit dès lors que le nombre de lignes de A est inférieur ou égal à son nombre de colonnes. Si on inspecte le nombre de contraintes du problème, nous obtenons donc m contraintes d'égalité et n contraintes de positivité sur les variables, c'est-à-dire un total de $m + n$ contraintes pour n variables. La définition d'une solution de base requiert que les m contraintes d'égalité soient satisfaites et qu'un total de n contraintes soient serrées. Au total, nous aurons donc $n - m$ contraintes de type $x_i \geq 0$ qui seront serrées. En d'autres termes, $n - m$ variables sont nulles dans une solution de base. Les m autres contraintes de positivité ne sont pas nécessairement serrées. On aura donc au plus m variables non nulles. La description suivante résume donc les propriétés d'une solution de base dans le cas d'un polyèdre en forme standard.

Observation 3.5 *Soit $P = \{x \in \mathbb{R}^n_+ \mid Ax = b, x \geq 0\}$ un polyèdre en représentation standard. Une solution de base x satisfait*

(i) $x_i = 0$ *pour* $i \in N$ *où* $|N| = n - m$. *Ces variables sont appelées* hors-base.

(ii) $B = \{1, \ldots, n\} \setminus N$ *est l'ensemble des* variables de base. *On a* $|B| = m$ *et ces variables sont potentiellement non-nulles.*

(iii) $A_B x_B = b$ *où* A_B *consiste en les différentes colonnes de* A *qui correspondent aux colonnes des variables de base.*

Une solution de base est réalisable *si* $x_B = A_B^{-1} b$ *est tel que* $x_i \geq 0$ *pour tout* $i \in B$.

Exemple 3.11 Reprenons le système de l'Exemple 3.9 que nous écrivons cette fois en forme standard. Le polyèdre s'écrit donc

$$\begin{aligned} X = \{(x,s) \in \mathbb{R}^2_+ \times \mathbb{R}^3_+ \mid & -x_1 + 2x_2 + s_1 = 1 \\ & -x_1 + x_2 + s_2 = 0 \\ & 4x_1 + 3x_2 + s_3 = 12 \\ & x_1, \quad x_2, \; s_1, \; s_2, \; s_3 \geq 0 \; \}. \end{aligned}$$

Toute solution de base correspond à trois variables parmi x_1, x_2, s_1, s_2, s_3 dans la base (potentiellement non nulles) et deux variables hors-base (c'est-à-dire nulles). Le point A de la Figure 3.2 correspond au choix de x_2, s_2, s_3 dans la base et x_1, s_1 hors base. On a bien pour ce point $x_1 = 0$ et $s_1 = 0$. Si on veut obtenir les autres valeurs, on doit considérer la sous-matrice correspondant aux trois colonnes de la base et résoudre

$$\begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 12 \end{pmatrix}.$$

Cela nous donne donc $(x_1, x_2, s_1, s_2, s_3) = (0, \frac{1}{2}, 0, -\frac{1}{2}, \frac{21}{2})$ est une solution de base. On voit que c'est une solution de base *non réalisable* puisque $s_2 < 0$. Le point B de la Figure 3.2 quant à lui, correspond au choix de x_1, x_2, s_3 dans la base et s_1, s_2 hors base. En résolvant

$$\begin{pmatrix} -1 & 2 & 0 \\ -1 & 1 & 0 \\ 4 & 3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 12 \end{pmatrix},$$

on trouve $(x_1, x_2, s_1, s_2, s_3) = (1, 1, 0, 0, 5)$ qui est cette fois une solution de base réalisable puisque toutes les variables sont bien positives. Remarquons que deux variables ont échangé leur place en base et hors-base. Il s'agit de x_2 qui est hors-base pour A et en base pour B. En revanche s_2 est en base pour A et hors-base pour B. Lorsqu'on peut passer d'une solution de base à une autre en échangeant le rôle (base ou hors base) de deux variables, on parle de solutions de base *adjacentes*. ■

3.5.3 L'algorithme du simplexe

Nous allons maintenant étudier le premier algorithme efficace qui fut proposé pour résoudre des problèmes de programmation linéaire, à savoir l'algorithme du simplexe. L'idée de l'algorithme est de partir d'un sommet dont on a la représentation en solution de base. On va ensuite vérifier si ce sommet est optimal ou pas. S'il n'est pas optimal, on pourra trouver un sommet *adjacent* dont la valeur de l'objectif est inférieure. Ecrire le nouveau sommet consistera en une sorte d'élimination gaussienne que l'on appellera *pivot*.

Considérons donc un polyèdre $P = \{x \in \mathbb{R}^n_+ \mid Ax = b, x \geq 0\}$ en forme standard. Supposons donc que l'on connaisse une solution de base réalisable et l'indice des variables de base B et l'indice des variables hors base N. On a donc $B \cup N = \{1, \ldots, n\}$. On peut réécrire les contraintes d'égalité comme

$$A_B x_B + A_N x_N = b,$$

où A_B représente les colonnes de A qui correspondent aux indices présents dans B. Remarquons que par construction, A_B est une matrice carrée alors que A_N ne l'est pas nécessairement. On peut inverser A_B pour obtenir une représentation de la valeur des variables de base. On a donc

$$x_B + A_B^{-1} A_N x_N = A^{-1} b. \tag{3.64}$$

En d'autres termes, la solution qui est représentée est $x_B = A_B^{-1} b, x_N = 0$. Cette solution est bien réalisable si on a $x_B = A_B^{-1} b \geq 0$.

Changer de solution de base Nous allons maintenant essayer de changer de solution pour éventuellement pouvoir améliorer la solution courante par rapport à l'objectif. Supposons donc que $\bar{x}_B = A_B^{-1} b, \bar{x}_N = 0$ est la solution

courante. Si nous voulons changer de solution courante, il nous faut rendre au moins une des variables hors base non nulle. Décidons donc arbitrairement d'une variable $j \in N$ que nous souhaitons rendre non nulle. On peut écrire $\tilde{x}_j = \theta$ et se demander ce que cela implique pour les variables de base que nous écrivons $\tilde{x}_B = \bar{x}_B + \theta d_B$. On doit toujours satisfaire les égalités décrivant le problème. On a donc $A_B\tilde{x}_B + A_j\theta = b$. Remarquons que l'on a $A_B\bar{x} = b$. Si on le soustrait de l'équation précédente, on obtient donc $A_B(\tilde{x}_B - \bar{x}_B) + A_j\theta = 0$ c'est-à-dire $A_B d_B\theta + A_j\theta = 0$. Finalement, on obtient

$$d_B = -A_B^{-1}A_j. \tag{3.65}$$

En conclusion, si on écrit $\tilde{x}_B = \bar{x}_B + \theta d_B, x_j = \theta, x_i = 0, i \in N \setminus \{j\}$, les contraintes d'égalité restent toujours satisfaites. Si on veut conserver toutes les contraintes satisfaites, il nous faut donc maintenant tenir compte des contraintes de positivité sur les variables. Nous le ferons lorsque nous discuterons du changement de base ou pivot. Nous allons distinguer deux cas liés à la définition suivante.

Définition 3.12 *Soit $P = \{x \in \mathbb{R}^n_+ \mid Ax = b, x \geq 0\}$ un polyèdre en forme standard et soit $\bar{x}$ une solution de base réalisable pour P. On dit que $\bar{x}$ est une solution de base* dégénérée *si il existe $i \in B$ tel que $x_i = 0$.*

Si on en revient maintenant à la satisfaction des contraintes de positivité des variables de base, on peut maintenant distinguer deux cas.
Cas 1 La solution de base $\bar{x}$ n'est pas dégénérée.
On a donc $\bar{x}_B > 0$ et en particulier il existe $\theta > 0$ tel que $x_B + \theta d \geq 0$. La direction d nous mène donc dans une direction réalisable. En revanche si il existe $d_i < 0$ pour $i \in B$, on n'a pas que $\bar{x}_B + \theta d \geq 0$ pour tout θ. On ne pourra donc pas suivre la direction jusqu'à l'infini. Le θ est limité par l'ensemble des directions négatives et on a

$$\theta_{max} = \min_{i \in B | d_i < 0} \frac{x_i}{|d_i|}. \tag{3.66}$$

Appelons k l'indice qui réalise le minimum de (3.66). On peut remarquer que si on calcule $\tilde{x}_B = \bar{x}_B + \theta_{max} d, \tilde{x}_i = \theta_{max}$, on a en réalité $\tilde{x}_k = 0$ et $\tilde{x}_i > 0$. en d'autres termes, on a généré un nouveau sommet de P (adjacent à $\bar{x}$) où i est maintenant dans la base et k est sorti de la base. L'opération que nous venons de décrire s'appelle *un pivot*.
Cas 2 La solution de base $\bar{x}$ est dégénérée.

Supposons que $k \in B$ soit un indice de la base tel que $\bar{x}_k = 0$. Si $d_k < 0$, on voit que l'expression (3.66) nous donnerait $\theta_{max} = 0$. Dans ce cas, on ne peut pas utiliser la direction d pour changer de sommet. Néanmoins, on peut changer de *base*. En effet, on peut échanger i et k, similairement au cas précédent. On effectue dans ce cas un pivot dégénéré, c'est-à-dire que l'on change de base sans changer de sommet.

Vérifier l'optimalité d'une base Nous allons maintenant évaluer l'effet d'un pivot au niveau de la fonction objectif. Dans ce paragraphe, nous appellons le *coût* d'une solution la valeur que cette solution donne dans l'objectif. Ainsi le coût de la base initiale $\bar{x}$ est $\bar{c} = c_B^T \bar{x}_B$ où c_B est la restriction du vecteur objectif aux indices correspondant à la base. Si on pivote vers le sommet $\tilde{x}$ où $\tilde{x}_B = \bar{x}_B + \theta d$ et $\tilde{x}_j = \theta$, le nouveau coût est $\tilde{c} = c_B^T \bar{x}_B + \theta c_B^T d + c_j \theta$. Si on remplace d par sa valeur calculée en (3.65), on obtient finalement

$$\tilde{c} = \bar{c} + \theta(c_j - c_B^T A_B^{-1} A_j). \tag{3.67}$$

La quantité dans la parentèse en (3.67) indique le taux de changement de l'objectif lorsque l'on suit la direction du pivot qui fait rentrer x_j dans la base. Nous le mettons en évidence dans une définition

Définition 3.13 *Soit $\bar{x}$ une solution de base réalisable d'un polyèdre P en forme standard et soit x_j une variable hors-base. On définit le* coût réduit *de x_j comme*

$$\bar{c}_j := c_j - c_B^T A_B^{-1} A_j.$$

Ce coût réduit est très important. En effet, s'il est négatif, cela signifie que l'on peut diminuer la fonction objectif en effectuant le pivot consistant à faire entrer x_j dans la base. S'il est positif, cela veut dire qu'il n'est pas intéressant d'effectuer le pivot en ce qui concerne la fonction objectif. Pour clôturer ce paragraphe, nous allons voir que les coûts réduits contiennent toute l'information dans le sens où, si tous sont positifs, on ne peut, non seulement pas trouver de sommet adjacent améliorant la solution courante, mais la solution courante est tout simplement optimale.

Théorème 3.10 *Soit une solution de base réalisable x d'un polyèdre P en forme standard. Si $\bar{c}_j \geq 0$ pour tout $j \in N$ alors x est une solution optimale du problème linéaire correspondant.*

Démonstration: Supposons que $\bar{c}_j \geq 0$ pour tout $j \in N$. Considérons une autre solution réalisable y de P. On a $A(y-x) = 0$. Nous notons par $v = y-x$. On a donc $A_B v_B + \sum_{i \in N} A_i v_i = 0$. En inversant, on trouve

$$v_B = -\sum_{i \in N} A_B^{-1} A_i v_i$$

et finalement

$$\begin{aligned} c^T v &= c_B^T v_B + \sum_{i \in N} c_i v_i \\ &= \sum_{i \in N} (c_i - c_B^T A_B^{-1} A_i) v_i \\ &= \sum_{i \in N} \bar{c}_i v_i. \end{aligned}$$

Remarquons que $v_i \geq 0$ pour tout $i \in N$ puisque $x_i = 0$ et $y_i \geq 0$ pour $i \in N$. De plus $\bar{c}_i \geq 0$ pour tout $i \in N$ par hypothèse. Dès lors aucune autre solution réalisable y ne peut améliorer la solution courante x. ■

L'algorithme et le tableau du simplexe Nous avons maintenant tous les éléments en main pour énoncer l'algorithme du simplexe, ou plus exactement la *phase 2* de l'algorithme du simplexe. En effet, nous supposons dans ce développement que nous disposons au départ d'une solution de base réalisable (ce qui ne va pas toujours de soi). Pour développer l'algorithme, nous introduisons maintenant une représentation extrêmement utile de chaque solution de base et que nous appelons le tableau du simplexe. Considérons donc une solution de base réalisable x avec un ensemble d'indices de variables de base $B = 1, \ldots, m$ et un ensemble d'indices de variables hors base $N = \{m+1, \ldots, n\}$. Si on reprend (3.64), on peut écrire en forme détaillée le tableau des contraintes du simplexe.

$$\begin{array}{cccc} x_1 & + \bar{a}_{1,m+1} x_{m+1} & + \cdots + & \bar{a}_{1,n} x_n = \bar{b}_1 \\ \ddots & \vdots & \ddots & \vdots \\ & x_m + \bar{a}_{m,m+1} x_{m+1} & + \cdots + & \bar{a}_{m,n} x_n = \bar{b}_m, \end{array}$$

où $\bar{A} = A_B^{-1} A_N$ et $\bar{b} = A_B^{-1} b$. On représente en quelque sorte dans le tableau les contraintes déjà multipliées par A_B^{-1}. Dans le tableau, il est également utile de représenter les différents coûts réduits des variables hors base. On peut le faire en calculant la formule de la Définition 3.13 ou en opérant une élimination gaussienne des coûts des variables de base dans l'objectif. Le tableau du simplexe s'écrit donc

$$\begin{array}{rrrrrrrrr} \min & & & & \bar{c}_{m+1}x_{m+1} & + \cdots + & \bar{c}_n x_n & & \\ & x_1 & & + \bar{a}_{1,m+1}x_{m+1} & + \cdots + & \bar{a}_{1,n}x_n & = \bar{b}_1 & & \\ & & \ddots & \vdots & & \ddots & \vdots & & \\ & & & x_m + \bar{a}_{m,m+1}x_{m+1} & + \cdots + & \bar{a}_{m,n}x_n & = \bar{b}_m. & & \end{array} \tag{3.68}$$

Le tableau suivant indique les différentes étapes de l'algorithme du simplexe.

Algorithme du simplexe

1. On part d'un tableau du type (3.68) avec une base B et des variables hors base N. La solution représentée est donc $x_B = \bar{b}, x_N = 0$.
2. Si tous les coûts réduits $\bar{c}_j \geq 0$ pour tout $j \in N$, alors la solution x est optimale, STOP !
3. Choisir $j \in N$ tel que $\bar{c}_j < 0$. Nous allons faire rentrer x_j dans la base.
4. Pour déterminer quelle variable sort de la base, nous calculons

$$\theta^* = \min_{i \in B | \bar{a}_{ij} > 0} \frac{\bar{b}_i}{\bar{a}_{ij}}. \tag{3.69}$$

   Remarquons que s'il n'existe pas de i tel que $\bar{a}_{ij} > 0$, alors on peut augmenter indéfiniment la variable x_j sans jamais violer les contraintes. Dans ce cas, le problème est non borné, c'est-à-dire que l'on peut toujours améliorer n'importe quelle solution, STOP !
5. Soit k l'indice qui réalise le minimum (3.69). Nous pouvons effectuer un pivot, c'est-à-dire faire rentrer x_j dans la base et en faire sortir x_k. Il n'est pas nécessaire de recalculer l'inverse de la matrice pour cette opération. On peut également procéder à une élimination gaussienne afin d'obtenir un 1 dans la k^e ligne pour l'élément j et des 0 dans tous les autres éléments de la colonne j. On retourne ensuite à l'étape 1 avec la nouvelle solution de base ayant une valeur inférieure de la fonction objectif.

Il n'est pas difficile de voir que si toutes les bases sont non-dégénérées, l'algorithme du simplexe va toujours terminer soit avec une solution optimale, soit avec une preuve que le problème est non borné. Dans le cas où certaines bases sont dégénérées, il peut arriver que, par un mauvais choix des pivots, l'algorithme cycle et revienne un nombre infini de fois à la même base. Il existe différentes règles qui permettent d'éviter de cycler, mais cela sort du cadre de ce cours.

Phase I Pour terminer cette section sur l'algorithme du simplexe, nous allons brièvement aborder la question de trouver une base réalisable initiale. Pour débuter, nous allons voir un cas particulier où il est très simple de trouver une base réalisable initiale. Supposons en effet que nous voulions résoudre le problème $\min c^T x$ s.c.q. $Ax \leq b$ avec $b \in \mathbb{R}_+$. Si nous rajoutons des variables d'écart, on peut facilement écrire $\min c^T x$ s.c.q $Is + Ax = b$. On voit que l'on obtient directement la forme du tableau avec une matrice identité et un membre de droite positif. On peut donc mettre toutes les variables d'écart dans la base initiale et toutes les autres variables hors base.

Dans quel cas ne peut-on pas faire cela ? Ce sera plus difficile de trouver une base initiale si on a des contraintes d'égalité dans le problème de départ ou si on a des contraintes $\geq$. Nous pouvons cependant supposer, qu'après avoir éventuellement introduit des variables d'écart, le problème s'écrit $Ax = b$ avec $b \geq 0$. On peut maintenant modéliser la recherche d'une base initiale par le problème linéaire auxiliaire suivant

$$\begin{aligned} \min\ & \xi_1 + \cdots + \xi_m \\ \text{s.c.q. } & Ax + I\xi = b \\ & \xi_i \geq 0. \end{aligned} \tag{3.70}$$

Remarquons tout d'abord qu'il est aisé de trouver une base initiale réalisable pour ce problème. Il suffit pour ce faire de mettre les variables artificielles ξ_i dans la base. Ensuite nous pouvons remarquer qu'il existe une solution réalisable au problème initial si et seulement si il existe une solution optimale du problème (3.70) qui a une valeur de l'objectif de 0. Résoudre le problème (3.70) s'appelle résoudre la phase 1 du simplexe. Cela peut se faire en utilisant les étapes classiques de l'algorithme du simplexe.

Chapitre 4

Systèmes non linéaires

Dans certains cas, il est possible d'adapter les méthodes de résolution d'équations non linéaires à la résolution de systèmes d'équations non linéaires. Dans ce chapitre, nous allons donc nous attarder à la résolution de $F(x) = 0$ où $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ et où x et 0 sont considérés comme des vecteurs de dimension n. Remarquons tout d'abord qu'il n'est pas possible d'adapter la méthode de la bisection au cas multidimensionnel. Par contre, la méthode du point fixe et celle de Newton-Raphson s'adaptent sans trop de problèmes au cas de la résolution d'un système non-linéaire. Pour éviter toute confusion, nous utilisons la convention que tous les vecteurs seront systématiquement soulignés.

4.1 Méthode du point fixe pour un système

Si nous souhaitons résoudre $F(\underline{x}) = 0$, on peut à nouveau écrire de manière équivalente $\underline{x} = G(\underline{x})$ pour une certaine fonction G. Il existe d'ailleurs une multitude de choix pour G. La méthode du point fixe appliquée aux systèmes est totalement similaire au cas à une variable. On part d'un "point" (qui est ici un vecteur) $\underline{x}_1$. Ensuite on itère $\underline{x}_{k+1} = G(\underline{x}_k)$. La condition de Lipschitz est à nouveau suffisante pour garantir la convergence.

Proposition 4.1 *Soit $G : \mathbb{R}^n \mapsto \mathbb{R}^n$ une fonction dont $\underline{\bar{x}}$ est un point fixe $G(\underline{\bar{x}}) = \underline{\bar{x}}$. Considérons la boule $B = \{x \mid \|x - \underline{\bar{x}}\| \leq r\}$ pour $r > 0$. Si G satisfait la condition de* Lipschitz

$$\|G(\underline{x}) - G(\underline{\bar{x}})\| \leq L\|\underline{x} - \underline{\bar{x}}\| \qquad \textit{pour tout } \underline{x} \in B,$$

avec $0 < L < 1$, *alors l'itération* $\underline{x}_{k+1} = G(\underline{x}_k)$ *converge vers* $\bar{\underline{x}}$ *pour tout point de départ* $\underline{x}_1 \in B$.

Souvenons-nous que dans le cas d'équations scalaires non linéaires, la convergence d'une telle méthode était assurée si la dérivée de g était inférieur ou égale à 1 dans un intervalle centré autour de la racine. Nous ne pouvons pas répéter telle quelle cette condition. Néanmoins il existe une condition similaire sur la *Jacobienne* de G. Rappelons que l'on peut écrire

$$G = (G_1(x_1, \ldots, x_n), \ldots, G_n(x_1, \ldots, x_n)).$$

La Jacobienne de G s'écrit

$$J(x) = \begin{pmatrix} \frac{\partial G_1}{\partial x_1} & \cdots & \frac{\partial G_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_n}{\partial x_1} & \cdots & \frac{\partial G_n}{\partial x_n} \end{pmatrix}.$$

Formellement, la condition $\|J(x)\| \leq 1$ est suffisante pour obtenir la convergence. En pratique, le choix de la norme 2 traditionnelle sur les matrices donne n conditions.

Proposition 4.2 *Soit* $G : \mathbb{R}^n \mapsto \mathbb{R}^n$ *une fonction différentiable qui admet un point fixe* $\bar{\underline{x}}$. *Considérons la boule* $B = \{\underline{x} \mid \|\underline{x} - \bar{\underline{x}}\| \leq r\}$. *Si pour tout* $\underline{x} \in B$, *on a*

$$\sum_{i=1}^{n} \left| \frac{\partial G_1(\underline{x})}{\partial x_i} \right| < 1$$

$$\vdots$$

$$\sum_{i=1}^{n} \left| \frac{\partial G_n(\underline{x})}{\partial x_i} \right| < 1$$

alors l'itération $\underline{x}_{k+1} = G(\underline{x}_k)$ *converge vers* $\bar{\underline{x}}$ *pour tout point de départ* $\underline{x}_1 \in B$.

Exemple 4.1 Nous cherchons à résoudre le problème

$$x^2 + y^2 = 4 \tag{4.1}$$

$$\cos(xy) = 0. \tag{4.2}$$

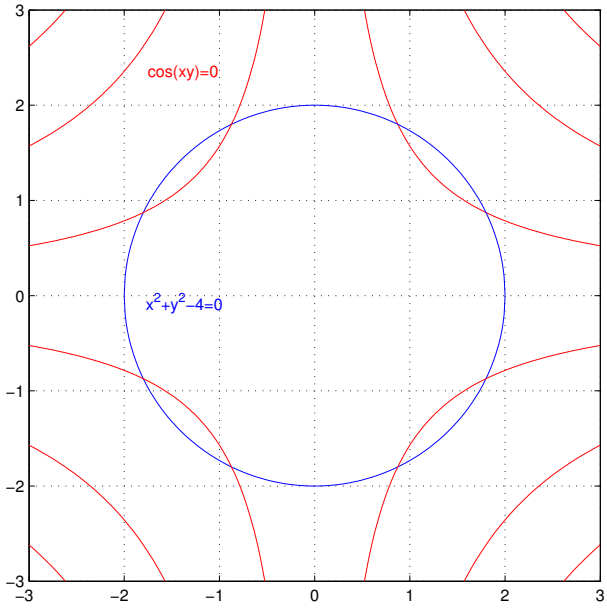

FIGURE 4.1 – Le système (4.1)-(4.2)

Les courbes des solutions à (4.1) et (4.2) sont représentées sur la Figure 4.1. On voit que le problème admet 8 racines. Il est évidemment possible de résoudre le problème analytiquement. Mais nous allons procéder ici numériquement. Premièrement, nous remarquons que le problème est symétrique. Si $x = a, y = b$ est une solution, on aura aussi $x = b, y = a$ mais également $x = -a, y = b, x = -a, y = -b$ et ainsi de suite comme solutions. Contentons-nous donc de rechercher une solution (x, y) avec $x, y \geq 0$. Pour appliquer la méthode du point fixe, on doit d'abord écrire les équations (4.1)-(4.2) sous une forme $\underline{x} = G(\underline{x})$. On pourra par exemple écrire

$$x = G_1(x, y) = \sqrt{4 - y^2} \tag{4.3}$$
$$y = G_2(x, y) = y + \cos(xy). \tag{4.4}$$

Il est utile à présent d'analyser ces réécritures. Premièrement, on voit que l'on peut écrire (4.3) puisqu'on s'intéresse à une solution avec $x \geq 0$. Voyons

maintenant si la méthode a une chance de converger. On calcule

$$\frac{\partial G_1(x,y)}{\partial x} = 0 \qquad \frac{\partial G_1(x,y)}{\partial y} = -\frac{y}{\sqrt{4-y^2}} \tag{4.5}$$

$$\frac{\partial G_2(x,y)}{\partial x} = -y\sin(xy) \qquad \frac{\partial G_2(x,y)}{\partial y} = 1 - x\sin(xy). \tag{4.6}$$

Remarquons que dans les dérivées partielles de G_2, si on considère (x, y) proche de la racine, comme $\cos(xy) \approx 0$, on aura $\sin(xy) \approx 1$. Cela suggère que la somme des valeurs absolues des deux dérivées partielles sera proche de $x + y - 1$ (en supposant que les deux dérivées partielles sont négatives). Comme on a $x^2 + y^2 = 4$, on a aussi $x^2 + y^2 + 2xy - 2xy = 4$ c'est-à-dire $(x + y)^2 - 2xy = 4$. En d'autres termes, dans le quadrant $x, y \geq 0$, on a $x + y \geq 2$. Les formes (4.5)-(4.6) risquent donc bien de ne pas converger car il y a de grandes chances que

$$\left|\frac{\partial G_2(x,y)}{\partial x}\right| + \left|\frac{\partial G_2(x,y)}{\partial y}\right| \approx x + y - 1 \geq 1$$

aux alentours de la racine. Peut-on s'en sortir ? Pour cela, il suffit de voir que pour obtenir (4.4), on s'est servi du fait que $F_2(x, y) = 0$ est équivalent à $y = y + F_2(x, y)$. Mais on aurait pu également écrire $y = y + F_2(x, y)/2$. Cela nous donne une deuxième forme des équations (4.1)-(4.2)

$$x = G_1(x,y) = \sqrt{4-y^2} \tag{4.7}$$

$$y = G_2(x,y) = y + \frac{1}{2}\cos(xy). \tag{4.8}$$

Nous lançons donc la méthode du point fixe avec les deux formes (4.3)-(4.4) et (4.7)-(4.8) en partant de $(x_1, y_1) = (1, 1)$. Les valeurs obtenues sont renseignées dans le Tableau 4.1 et le Tableau 4.2.

Il est intéressant de noter qu'aucune des deux formes ne diverge. Cependant, la première forme ne converge pas. Si on analyse attentivement les valeurs, on voit que la méthode *cycle*. Les valeurs obtenues pour (x_k, y_k) se répètent approximativement toutes les deux itérations. Quand on analyse la valeur donnée par $F(x_k, y_k)$, on voit que nous ne sommes pas proches d'une racine. La deuxième forme, quant à elle, converge vers une racine. Calculons, pour les deux formes, la valeur de la Jacobienne à la racine trouvée $(1.7989, 0.8736)$. On obtient repsectivement

$$J_1 \approx \begin{pmatrix} 0 & -0.49 \\ -0.87 & -0.8 \end{pmatrix}, \quad J_2 \approx \begin{pmatrix} 0 & -0.49 \\ -0.44 & 0.1 \end{pmatrix}.$$

Itération k	x_k	y_k	$F_1(x_k, y_k)$	$F_2(x_k, y_k)$	$G_1(x_k, y_k)$	$G_2(x_k, y_k)$
1	1	1	-2.0000	0.5403	1.7321	1.5403
2	1.7321	1.5403	1.3725	-0.8899	1.2757	0.6504
3	1.2757	0.6504	-1.9495	0.6751	1.8913	1.3255
4	1.8913	1.3255	1.3338	-0.8052	1.4977	0.5203
5	1.4977	0.5203	-1.4862	0.7115	1.9311	1.2317
6	1.9311	1.2317	1.2465	-0.7228	1.5757	0.5089
7	1.5757	0.5089	-1.2582	0.6953	1.9342	1.2043
8	1.9342	1.2043	1.1912	-0.6878	1.5968	0.5165
9	1.5968	0.5165	-1.1835	0.6788	1.9322	1.1952
10	1.9322	1.1952	1.1619	-0.6733	1.6036	0.5220
11	1.6036	0.5220	-1.1562	0.6697	1.9307	1.1917
12	1.9307	1.1917	1.1476	-0.6668	1.6062	0.5248

TABLE 4.1 – La méthode du point fixe avec la forme (4.3)-(4.4)

Itération k	x_k	y_k	$F_1(x_k, y_k)$	$F_2(x_k, y_k)$	$G_1(x_k, y_k)$	$G_2(x_k, y_k)$
1	1	1	-2.0000	0.5403	1.7321	1.2702
2	1.7321	1.2702	0.6133	-0.5885	1.5449	0.9759
3	1.5449	0.9759	-0.6609	0.0631	1.7457	1.0074
4	1.7457	1.0074	0.0625	-0.1868	1.7277	0.9140
5	1.7277	0.9140	-0.1795	-0.0084	1.7789	0.9098
6	1.7789	0.9098	-0.0077	-0.0477	1.7811	0.8860
7	1.7811	0.8860	-0.0428	-0.0072	1.7931	0.8824
8	1.7931	0.8824	-0.0064	-0.0114	1.7948	0.8767
9	1.7948	0.8767	-0.0100	-0.0027	1.7976	0.8753
10	1.7976	0.8753	-0.0024	-0.0027	1.7983	0.8740
11	1.7983	0.8740	-0.0024	-0.0009	1.7989	0.8736
12	1.7989	0.8736	-0.0007	-0.0007	1.7991	0.8732

TABLE 4.2 – La méthode du point fixe avec la forme (4.7)-(4.8)

On voit que la condition de la Proposition 4.2 est respectée pour la deuxième forme et pas pour la première. Rappelons que, par symétrie, $(0.87, 1.8)$ est également une racine du problème. Si on calcule les Jacobiennes en ce point, on obtient

$$J_1 \approx \begin{pmatrix} 0 & -2.06 \\ -1.8 & 0.13 \end{pmatrix}, \quad J_2 \approx \begin{pmatrix} 0 & -2.06 \\ -0.9 & 0.56 \end{pmatrix}.$$

On voit qu'aucune des deux formes ne pourra converger vers cette racine. ■

4.2 Méthode de Newton-Raphson pour un système

La méthode de Newton qui est un cas particulier amélioré de la méthode du point fixe, peut aussi être étendue relativement aisément au cas d'un système d'équations. La méthode converge, à nouveau, assez bien. C'est pour cette raison que la méthode est très populaire. Cependant la question de la convergence globale est très délicate. Dans le cas de plusieurs racines, il est en effet très difficile de prévoir vers quelle racine le processus sera attiré. La forme de la méthode de Newton dans le cas d'un système est très similaire au cas unidimensionnel. Supposons à nouveau que l'on veuille résoudre $F(\underline{x}) = \underline{0}$ où $F : \mathbb{R}^n \mapsto \mathbb{R}^n$. En partant d'un vecteur de départ $\underline{x}_1$, on calcule

$$\underline{x}_{k+1} = \underline{x}_k - \left(\frac{\partial F(\underline{x}_k)}{\partial \underline{x}}\right)^{-1} F(\underline{x}_k),$$

où $\left(\frac{\partial F(\underline{x}_k)}{\partial \underline{x}}\right)$ représente la *Jacobienne* de F calculée en $\underline{x}_k$. On voit que dans ce cas, au lieu de diviser par la dérivée, on devra résoudre un système linéaire où le membre de gauche est la matrice Jacobienne. On peut donc réécrire le processus comme

$$\underline{x}_{k+1} = \underline{x}_k - \underline{\Delta}_k$$

où $\underline{\Delta}_k \in \mathbb{R}^n$ est solution du système linéaire

$$\begin{pmatrix} \frac{\partial F_1(\underline{x}_k)}{\partial x_1} & \cdots & \frac{\partial F_1(\underline{x}_k)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n(\underline{x}_k)}{\partial x_1} & \cdots & \frac{\partial F_n(\underline{x}_k)}{\partial x_n} \end{pmatrix} \begin{pmatrix} \Delta_{k,1} \\ \vdots \\ \Delta_{k,n} \end{pmatrix} = \begin{pmatrix} F_1(\underline{x}_k) \\ \vdots \\ F_n(\underline{x}_k) \end{pmatrix}.$$

Exemple 4.2 Nous tentons à nouveau de résoudre le système (4.1)-(4.2), à savoir

$$\begin{aligned} x^2 + y^2 - 4 &= 0 \\ \cos(xy) &= 0. \end{aligned}$$

On va devoir résoudre un système où le membre de gauche est la Jacobienne de F et le membre de droite la valeur F au point de départ. Calculons tout d'abord la Jacobienne de F. On obtient

$$\frac{\partial F}{\partial \underline{x}} = \begin{pmatrix} 2x & 2y \\ -y\sin(xy) & -x\sin(xy) \end{pmatrix}.$$

A chaque étape, on doit donc résoudre le système

$$\begin{pmatrix} 2x_k & 2y_k \\ -y_k\sin(x_k y_k) & -x_k\sin(x_k y_k) \end{pmatrix} \begin{pmatrix} \Delta_{k,1} \\ \Delta_{k,2} \end{pmatrix} = \begin{pmatrix} x_k^2 + y_k^2 - 4 \\ \cos(x_k y_k) \end{pmatrix}.$$

Il faut maintenant se demander quel va être le point de départ de la méthode. Remarquons qu'en prenant un vecteur de la forme (a, a) (comme $(1, 1)$ dans l'exemple précédent), on obtient une Jacobienne singulière. De même, choisir 0 dans l'une des composantes mène à une Jacobienne singulière. Nous choisissons donc $(0.1, 1)$. Le système à résoudre est donc, pour la première itération,

$$\begin{pmatrix} 0.2 & 2 \\ -\sin(0.1) & -\frac{1}{10}\sin(0.1) \end{pmatrix} \begin{pmatrix} \Delta_{1,1} \\ \Delta_{1,2} \end{pmatrix} = \begin{pmatrix} -2.99 \\ \cos(0.1) \end{pmatrix}.$$

On obtient comme solution $\Delta_1 = (-9.9163, -0.5034)^T$ et donc $x_2 = x_1 - \Delta_1 = (10.0163\ 1.5034)^T$. Les différentes itérations suivantes sont reportées dans la Tableau 4.3.

Une fois encore, on voit que pour des itérés très proches de la racine, la méthode converge très vite. Bien qu'ayant convergé vers la même racine que précédemment, cette fois, rien n'empêche de converger vers n'importe laquelle des 8 racines. La difficulté de trouver une valeur initiale qui n'annule pas le déterminant de la Jacobienne indique une des difficultés de la méthode de Newton. Si dans le courant de l'algorithme, on trouve une Jacobienne proche de la singularité, l'itéré suivant risque d'être très éloigné et de rendre la convergence difficile. ■

Itération k	x_k	y_k	$F_1(x_k, y_k)$	$F_2(x_k, y_k)$
1	0.1000	1.0000	-2.9900	0.9950
2	10.0163	1.5034	98.5865	-0.7962
3	5.0018	2.1246	25.5316	-0.3604
4	1.8476	3.5417	11.9571	0.9663
5	4.5137	0.4628	16.5879	-0.4953
6	2.6698	0.5256	3.4040	0.1669
7	1.9936	0.7221	0.4958	0.1309
8	1.8229	0.8501	0.0456	0.0211
9	1.8000	0.8724	0.0010	0.0005
10	1.7994	0.8729	0.0000	0.0000

TABLE 4.3 – 10 itérations de la méthode de Newton pour le système (4.1)-(4.2)

4.3 Méthode Quasi-Newton

La méthode de Newton est très attrayante de par sa bonne convergence. Elle a cependant un gros défaut : on doit disposer de la forme analytique de la Jacobienne pour pouvoir l'utiliser. Or dans de nombreux cas, on souhaite résoudre un système d'équations où la fonction n'est donnée qu'implicitement. C'est par exemple le résultat d'un autre calcul comme la résolution d'un système d'équations différentielles ou aux dérivées partielles. Dans ce cas, ni la méthode de point fixe ni la méthode de Newton ne peut nous aider puisqu'elles requièrent toutes deux l'expression analytique de la fonction. Une première approche que l'on pourrait considérer pour utiliser tout de même la méthode de Newton est la suivante. A chaque itération, on peut évaluer *numériquement* la Jacobienne de la fonction F en $\underline{x}_k$. On va donc choisir un petit pas h et calculer successivement $\frac{F(\underline{x}_k+he_i)-F(\underline{x}_k)}{h}$ où e_i représente le vecteur unité ayant 1 dans sa i^e coordonnée. Remarquons que l'on n'a pas besoin d'une très grande précision sur la Jacobienne dans ce cas-ci et qu'il n'y a pas de réel intérêt à prendre h très petit ou à vouloir utiliser une formule centrée de la dérivée partielle, ce qui doublerait le nombre d'évaluations de fonctions. Une fois l'évaluation numérique de la Jacobienne obtenue, il nous suffit ensuite d'appliquer la méthode de Newton comme dans la section précédente.

Quelques remarques s'imposent au sujet de cette implémentation "numérique"

de la méthode de Newton. Tout d'abord, on voit que la Jacobienne doit être recalculée à chaque itération, alors qu'il est fort probable que près de la racine, celle-ci ne change plus drastiquement. De plus, à chaque itération, on doit réaliser $n+1$ évaluations de la fonction F. Cela peut être extrêmement coûteux surtout si n est très élevé. Si on se rappelle de la méthode de la sécante dans le cas de la résolution numérique d'une équation non linéaire, dans ce cas, la dérivée était approximée en se servant de l'évaluation de f à l'itération précédente. On avait, pour caricaturer,

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

Ainsi, une seule évaluation de fonction était nécessaire à chaque étape. Nous allons essayer de construire une méthode similaire dans le cas de la recherche d'une racine d'une fonction à plusieurs variables. On veut donc obtenir une approximation de la Jacobienne de F en $\underline{x}_k$ à partir de deux évaluations de F, à savoir $F(\underline{x}_k)$ et $F(\underline{x}_{k-1})$. Etant donné que l'on dispose de vecteurs et non pas de scalaires, on ne peut pas effectuer un quotient tel quel. On voudrait cependant obtenir une matrice A_k qui vérifie $A_k(\underline{x}_k - \underline{x}_{k-1}) = F(\underline{x}_k) - F(\underline{x}_{k-1})$. Si on note par $\underline{d}_{k-1} := \underline{x}_k - \underline{x}_{k-1}$ la différence entre deux itérés consécutifs et par $\underline{y}_{k-1} := F(\underline{x}_k) - F(\underline{x}_{k-1})$ la différence entre deux évaluations consécutives de la fonction, on recherche donc comme approximation de la Jacobienne une matrice A_k satisfaisant

$$A_k \underline{d}_{k-1} = \underline{y}_{k-1}. \tag{4.9}$$

Dans (4.9), appelée équation sécante, les inconnues sont les éléments de la matrice A_k, on voit que l'on a donc à faire à un système sous-déterminé, puisque on dispose de n équations pour n^2 inconnues. Il y a donc une infinité de solutions à ce système. Il semble naturel de choisir comme solution à (4.9), une matrice A_k qui ne soit pas trop éloignée de la matrice A_{k-1} que l'on a utilisée à l'itération précédente. Le théorème suivant indique quelle est cette matrice.

Théorème 4.1 (Optimalité de Broyden) *Soit $A_{k-1} \in \mathbb{R}^{n\times n}, \underline{d}_{k-1}, \underline{y}_{k-1} \in \mathbb{R}^n$ et soit S l'ensemble des matrices carrées d'ordre n satisfaisant l'équation sécante, c'est-à-dire $S = \{A \in \mathbb{R}^{n\times n} \mid A\underline{d}_{k-1} = \underline{y}_{k-1}\}$. Une solution optimale du problème d'optimisation*

$$\min_{A\in S} \|A - A_{k-1}\|_2$$

est donnée par

$$A_k = A_{k-1} + \frac{(\underline{y}_{k-1} - A_{k-1}\underline{d}_{k-1})\underline{d}_{k-1}^T}{\underline{d}_{k-1}^T\underline{d}_{k-1}}.$$

Démonstration: Soit $A \in S$ une matrice arbitraire satisfaisant l'équation sécante. On a successivement

$$\begin{aligned}
\|A_k - A_{k-1}\|_2 &= \left\| \frac{(\underline{y}_{k-1} - A_{k-1}\underline{d}_{k-1})\underline{d}_{k-1}^T}{\underline{d}_{k-1}^T\underline{d}_{k-1}} \right\|_2 \quad \text{par définition de } A_k \\
&= \left\| \frac{(A\underline{d}_{k-1} - A_{k-1}\underline{d}_{k-1})\underline{d}_{k-1}^T}{\underline{d}_{k-1}^T\underline{d}_{k-1}} \right\|_2 \quad \text{car } A \in S \\
&= \left\| \frac{(A - A_{k-1})\underline{d}_{k-1}\underline{d}_{k-1}^T}{\underline{d}_{k-1}^T\underline{d}_{k-1}} \right\|_2 \\
&\leq \|A - A_{k-1}\|_2 \left\| \frac{\underline{d}_{k-1}\underline{d}_{k-1}^T}{\underline{d}_{k-1}^T\underline{d}_{k-1}} \right\|_2 \quad \text{en utilisant Définition 3.2.}(iv)
\end{aligned}$$

On peut prouver que pour deux vecteurs quelconques $x, y \in \mathbb{R}^n$, on a $\|xy^T\|_2 = \|x\|_2\|y\|_2$. Ceci nous permet de conclure finalement que $\|A_k - A_{k-1}\|_2 \leq \|A - A_{k-1}\|_2$ pour toute matrice $A \in S$ ce qui est le résultat désiré. ■

On peut maintenant énoncer l'algorithme de Quasi-Newton pour la résolution d'un système d'équations non linéaires et que nous résumons ci-dessous.

Méthode de Quasi-Newton

On cherche à résoudre $F(\underline{x}) = 0$ où $F : \mathbb{R}^n \to \mathbb{R}^n$.

1. **Initialisation** : $\underline{x}_0$ et une première approximation de la Jacobienne, par exemple $A_0 = I$.
   $\underline{x}_1 = \underline{x}_0 - A_0^{-1}F(\underline{x}_0)$,
   $\underline{d}_0 = \underline{x}_1 - \underline{x}_0$,
   $\underline{y}_0 = F(\underline{x}_1) - F(\underline{x}_0)$,
   $k = 1$
2. `while` $\|F(\underline{x}_k)\| > \epsilon$ `faire`
   $A_k = A_{k-1} + \frac{(\underline{y}_{k-1} - A_{k-1}\underline{d}_{k-1})\underline{d}_{k-1}^T}{\underline{d}_{k-1}^T\underline{d}_{k-1}}$

Calculer la solution de l'équation linéaire $A_k \underline{d}_k = -F(\underline{x}_k)$
Mettre à jour $\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k$ et $\underline{y}_k = F(\underline{x}_{k+1}) - F(\underline{x}_k)$,
$k = k + 1$

3. Retourner $\underline{x}_k$ comme solution

Exemple 4.3 Nous allons à nouveau tenter de trouver une racine de (4.1)-(4.2) à savoir

$$x^2 + y^2 - 4 = 0$$
$$\cos(xy) = 0.$$

Nous commençons par

$$\underline{x}_1 = (0, 0), F(\underline{x}_1) = (-4, 1), A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

La première étape nous donne naturellement

$$\underline{x}_2 = -F(\underline{x}_1) = (4, -1), F(\underline{x}_2) = (13, -0.65).$$

On calcule donc une mise à jour de la matrice A, servant de Jacobienne bien que probablement très éloignée de celle-ci. On a tout d'abord $\underline{y}_1 = (17, -1.65)$ et $\underline{d}_1 = (4, -1)$. On obtient donc

$$\begin{aligned} A_2 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{\left(\begin{pmatrix} 17 \\ -1.65 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ -1 \end{pmatrix} \right) (4 \ -1)}{((4 \ -1) \begin{pmatrix} 4 \\ -1 \end{pmatrix}} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{\begin{pmatrix} 13 \\ -0.65 \end{pmatrix} (4 \ -1)}{17} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{17} \begin{pmatrix} 52 & -13 \\ -2.6 & 0.65 \end{pmatrix} \\ &\approx \begin{pmatrix} 4.06 & -0.76 \\ -0.15 & 1.04 \end{pmatrix} \end{aligned}$$

Il nous suffit maintenant de résoudre $A_2 \underline{d}_2 = -F(x_2)$ pour obtenir l'itéré suivant comme $\underline{x}_3 = \underline{x}_2 + \underline{d}_2$. La Table 4.4 présente les 11 premières itérations

Itération k	x_k	y_k	$F_1(x_k, y_k)$	$F_2(x_k, y_k)$
1	0.000000	0.000000	-4.000000	1.000000
2	4.000000	-1.000000	13.000000	-0.653644
3	0.827158	-0.840469	-2.609422	0.767925
4	1.268661	-1.405328	-0.415551	-0.210503
5	1.376995	-1.192440	-0.681973	-0.071127
6	2.112109	-0.608250	0.830972	0.282219
7	1.651033	-1.025011	-0.223444	-0.121231
8	1.764100	-0.906102	-0.066931	-0.027654
9	1.804358	-0.868436	0.009890	0.003826
10	1.799299	-0.873067	-0.000279	-0.000111
11	1.799439	-0.872937	-0.000001	0.000000

TABLE 4.4 – 11 itérations de la méthode de Quasi-Newton

de la méthode de Quasi-Newton. On voit que l'on a pratiquement convergé aussi vite que la méthode de Newton. Ce qui est très étonnant, c'est que l'on ne nécessite pas une bonne approximation de la matrice Jacobienne pour converger néanmoins. Le gain de temps est, en revanche, considérable puisque, même si l'on ne dispose pas de la forme analytique de la fonction, nous n'effectuons qu'un appel de la fonction et une résolution de système linéaire à chaque itération et cela, quelle que soit la taille des variables $\underline{x}$. ■