Analyse numérique

Thomas Cluzeau
École Nationale Supérieure d'Ingénieurs de Limoges
16 rue d'atlantis, Parc ester technopole
87068 Limoges CEDEX
cluzeau@ensil.unilim.fr
http://www.unilim.fr/pages_perso/thomas.cluzeau

Table des matières

Chapitre 1

Arithmétique des ordinateurs et analyse d'erreurs

Dans tout ce cours, nous manipulerons des nombres réels. L'objet de ce premier chapitre est de décrire comment ces nombres réels sont représentés dans un ordinateur.

1.1 L'arithmétique flottante

1.1.1 Le système des nombres à virgule flottante

Théorème 1.1. *Soit β un entier strictement supérieur à 1. Tout nombre réel x non nul peut se représenter sous la forme*

$$x = \mathrm{sgn}(x)\, \beta^e \sum_{k\geq 1} \frac{d_k}{\beta^k},$$

où $\mathrm{sgn}(x) \in \{+,-\}$ est le signe de x, les d_k sont des entiers tels que $0 < d_1 \leq \beta - 1$ et $0 \leq d_k \leq \beta - 1$ pour $k \geq 2$, et $e \in \mathbb{Z}$. De plus, cette écriture est unique (sauf pour les décimaux : $2,5 = 2,499999\ldots$).

D'ordinaire, nous utilisons le système décimal, *i.e.*, $\beta = 10$ et les chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 pour les d_k. Nous avons par exemple :

- $0,0038 = 0,38.10^{-2} = +\,10^{-2}\,(\frac{3}{10} + \frac{8}{10^2})$.
  Remarque : en MATLAB, on peut écrire $\mathtt{0.38e-2}$ au lieu de $\mathtt{0.38*10\hat{}(-2)}$.
- $\frac{1}{7} = 0,142857\ldots = +\,10^{0}\,(\frac{1}{10} + \frac{4}{10^2} + \frac{2}{10^3} + \frac{8}{10^4} + \cdots)$. Notons que le développement décimal d'un nombre rationnel est périodique (ici, $\frac{1}{7} = 0,\mathbf{142857}142857142857\ldots$).
- $-\sqrt{2} = -1,4142\ldots = -\,10^{1}\,(\frac{1}{10} + \frac{4}{10^2} + \frac{1}{10^3} + \frac{4}{10^4} + \cdots)$.
- $\pi = 3,14159\ldots = +\,10^{1}\,(\frac{3}{10} + \frac{1}{10^2} + \frac{4}{10^3} + \frac{1}{10^4} + \cdots)$.

Historiquement, le choix $\beta = 10$ est lié à une particularité anatomique de la race humaine (nous avons 10 doigts). Les ordinateurs utilisent quant à eux $\beta = 2$ (numération binaire), $\beta = 8$ (numération octale), ou encore $\beta = 16$ (numération hexadécimale).

Remarquons que l'on perd l'unicité si on autorise $d_1 = 0$: en effet, on a par exemple :

$$\begin{array}{rcrcl} 0,0038 & = & 0,38.10^{-2} & = & +10^{-2}\,(\frac{3}{10} + \frac{8}{10^2}) \\ & = & 0,038.10^{-3} & = & +10^{-1}\,(\frac{0}{10} + \frac{3}{10^2} + \frac{8}{10^3}). \end{array}$$

On définit l'ensemble $F \subset \mathbb{R}$ par :

$$F = \{y \in \mathbb{R} \mid y = \pm\,\beta^e\,(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_t}{\beta^t}),\ e_{\min} \leq e \leq e_{\max}\},$$

ou encore

$$F = \{y \in \mathbb{R} \mid y = \pm\, m\,\beta^{e-t},\ e_{\min} \leq e \leq e_{\max}\}.$$

Ceci correspond aux deux écritures $0,0038 = +10^{-2}\,(\frac{3}{10} + \frac{8}{10^2}) = +38.10^{-4}$ avec $e = -2$, $t = 2$, $e - t = -4$. Le nombre m s'appelle *la mantisse* et on utilise la notation $m = \overline{d_1 d_2 \ldots d_t}^{\beta}$.

Notons que $0 \notin F$.

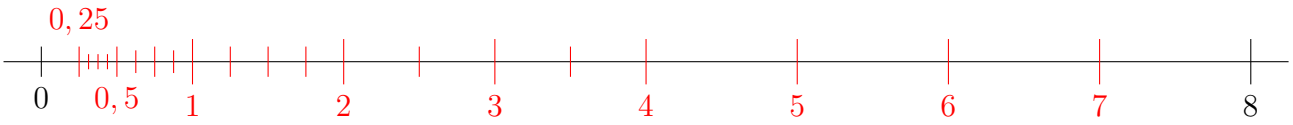
Pour $y \neq 0$, on a $m\,\beta^{e-t} = \beta^e\,(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_t}{\beta^t}) \geq \beta^e\,\frac{1}{\beta}$ car $d_1 \geq 1$. D'où $m \geq \beta^{t-1}$. D'autre part, $m = \overline{d_1 d_2 \ldots d_t}^{\beta} = d_1\,\beta^{t-1} + \cdots + d_{t-k}\,\beta^k + \cdots + d_{t-1}\,\beta + d_t < \beta^t$. On a donc montré que

$$\beta^{t-1} \leq m < \beta^t.$$

F est un *système de nombres à virgule flottante* (floating point number system) noté $F(\beta, t, e_{\min}, e_{\max})$. Il dépend de quatre paramètres :

1. *la base* β (chiffres utilisés 0, 1, ..., $\beta - 1$),
2. *la précision* t (nombre de chiffres utilisés pour représenter la mantisse),
3. $e_{\min}$ et $e_{\max}$ qui définissent *le domaine des exposants.*

Par exemple, pour $F(2, 3, -1, 3)$, on obtient les nombres représentés en rouge sur la figure ci-dessous :



On constate que l'écart entre deux nombres consécutifs est multiplié par 2 à chaque puissance de 2.

Dans le standard IEEE 754 utilisé par MATLAB, on a $\beta = 2$ et :

- en simple précision : $t = 24$, $e_{\min} = -125$, $e_{\max} = 128$,
- en double précision : $t = 53$, $e_{\min} = -1021$, $e_{\max} = 1024$.

Définition 1.2. *On appelle* epsilon machine *et on note ϵ_M la distance de* 1 *au nombre flottant suivant.*

Par exemple, pour $F(2, 3, -1, 3)$, on a $\epsilon_M = 0,25$. Dans MATLAB, c'est `eps`.

Proposition 1.3. *Pour $F(\beta, t, e_{\min}, e_{\max})$, on a $\epsilon_M = \beta^{1-t}$.*

Démonstration. On a $1 = \beta \frac{1}{\beta} = \beta^{1-t} \overline{10\ldots0}^{\beta}$. Le nombre suivant dans le système de nombres à virgule flottante $F(\beta, t, e_{\min}, e_{\max})$ est alors $\beta^{1-t} \overline{10\ldots1}^{\beta} = \beta^{1-t}(\beta^{t-1} + 1) = 1 + \beta^{1-t}$. □

Lemme 1.4. *Dans le système de nombres à virgule flottante $F(\beta, t, e_{\min}, e_{\max})$, l'écart $|y - x|$ entre un nombre flottant x (non nul) et un nombre flottant y (non nul) adjacent vérifie $\beta^{-1}\,\epsilon_M\,|x| \leq |y - x| \leq \epsilon_M\,|x|$.*

Démonstration. On a $x = m\,\beta^{e-t}$, donc $|y - x| = 1\,\beta^{e-t}$. Or $\beta^{t-1} \leq m < \beta^t$ donc $m\,\beta^{-t} < 1$ et $m\,\beta^{1-t} \geq 1$. Il vient donc $m\,\beta^{-t}\,\beta^{e-t} < 1\,\beta^{e-t} \leq m\,\beta^{1-t}\,\beta^{e-t}$ d'où le résultat puisque $\epsilon_M = \beta^{1-t}$. □

1.1.2 Représentation effective des réels et sa formalisation

• *Représentation « physique »* : par exemple, en simple précision 32 bits (bit = binary digit), 8 bits sont réservés à l'exposant et 24 bits (dont 1 pour le signe) à la mantisse. En double précision 64 bits, 11 bits sont réservés à l'exposant et 53 bits (dont 1 pour le signe) à la mantisse.

• Arrondi :

1. par troncature : par exemple avec 3 chiffres, $0,8573\ldots$ devient $0,857$.
2. au plus près : $0,8573\ldots$ devient $0,857$.
3. au représentant le plus proche dont la dernière décimale est paire (rounding to even) : $0,8573\ldots$ devient $0,858$.

• Formalisation :

Définition 1.5. *Soit $G = G(\beta, t) = \{y \in \mathbb{R} \mid y = \pm m\,\beta^{e-t}\}$ sans conditions sur l'exposant e. L'application* $\mathrm{fl} : \mathbb{R} \to G,\ x \mapsto \mathrm{fl}(x)$ *est appelée* opération d'arrondi.

Étant donné un domaine $F(\beta, t, e_{\min}, e_{\max})$, il y a alors dépassement de capacité si :

1. $|\mathrm{fl}(x)| > \max\{|y| \mid y \in F\}$. On parle d'« overflow ».
2. $|\mathrm{fl}(x)| < \min\{|y| \mid y \in F\}$. On parle d'« underflow ».

Sinon, x est dans le domaine de F.

1.1.3 Unité d'erreur d'arrondi, estimation d'erreurs

Définition 1.6. *Soit x un réel et $\overline{x}$ une valeur approchée de x. L'*erreur absolue *e est défini par $e = |x - \overline{x}|$. L'*erreur relative *est $|\frac{e}{x}|$. Le* pourcentage d'erreur *est l'erreur relative multipliée par* 100.

En pratique, on ne connait en général pas la valeur exacte x (c'est le cas dans la plupart des mesures physiques) mais on peut souvent avoir une idée de l'erreur maximale e que l'on a pu commettre : dans ce cas, on majore la quantité $|\frac{e}{\overline{x}}|$.

Théorème 1.7 (Estimation de l'erreur d'arrondi)**.** *Soit x un réel. Si x est dans le domaine $F(\beta, t, e_{\min}, e_{\max})$, alors il existe $\delta \in \mathbb{R}$ avec $|\delta| < u = \frac{1}{2}\,\beta^{1-t} = \frac{1}{2}\,\epsilon_M$ tel que* $\mathrm{fl}(x) = x\,(1+\delta)$.

Démonstration. Admis pour ce cours. □

L'erreur relative sur l'arrondi est égale à $|\delta| < u$: le nombre u s'appelle *unité d'erreur d'arrondi.*

Par exemple, dans le standard IEEE 754 utilisé par MATLAB, on a $u = 2^{-24} \approx 5,96.10^{-8}$ en simple précision et $u = 2^{-53} \approx 1,11.10^{-16}$ en double précision.

1.1.4 Modèle de l'arithmétique flottante

Le modèle suivant de l'arithmétique flottante est celui utilisé par le standard IEEE.

Modèle Standard : Soit $x,\ y \in F(\beta, t, e_{\min}, e_{\max})$. Pour $\mathrm{op} \in \{+, -, \times, \div, \sqrt{}\}$, on définit $x \boxed{\mathrm{op}} y = \mathrm{fl}(x \,\mathrm{op}\, y) = (x \,\mathrm{op}\, y)\,(1+\delta)$ avec $|\delta| < u = \frac{1}{2}\,\beta^{1-t} = \frac{1}{2}\,\epsilon_M$.

Nous allons maintenant nous intéressé aux erreurs faites par $\boxed{\mathrm{op}}$.

1.2 L'analyse d'erreurs

1.2.1 Non-associativité

En général, contrairement à op, l'opération $\boxed{\mathrm{op}}$ n'est pas associative. Ceci est dû aux erreurs d'arrondi. Par exemple, supposons que les réels soient calculés avec 3 chiffres significatifs et arrondis à la décimale la plus proche et cherchons à calculer la somme $x \boxed{+} y \boxed{+} z$ avec $x = 8,22$, $y = 0,00317$ et $z = 0,00432$.

- $x \boxed{+} y = 8,22$ donc $(x \boxed{+} y) \boxed{+} z = 8,22$,
- $y \boxed{+} z = 0,01$ donc $x \boxed{+} (y \boxed{+} z) = 8,23$.

1.2.2 Erreurs d'arrondi sur une somme

Supposons que l'on souhaite calculer une somme $S = u_1 + u_2 + \cdots + u_n$ de n réels positifs dans $F(\beta, t, e_{\min}, e_{\max})$. On calcule alors les sommes partielles S_i par la récurrence $S_0 = 0$, $S_i = S_{i-1} + u_i$. Si l'on suppose les u_i connus exactement, alors les erreurs d'arrondi ΔS_i commises sur le calcul des sommes partielles S_i vérifient $\Delta S_i \leq \Delta S_{i-1} + \delta\,(S_{i-1} + u_i) = \Delta S_{i-1} + \delta S_i$ où $|\delta| < u$. L'erreur globale sur $S = S_n$ vérifie donc

$$\Delta S \leq \delta\,(S_2 + \cdots + S_n),$$

ou encore

$$\Delta S \leq \delta\,(u_n + 2\,u_{n-1} + 3\,u_{n-2} + \cdots + (n-1)\,u_2 + (n-1)\,u_1).$$

On voit donc que pour minimiser cette erreur on a tout intérêt à sommer d'abord les termes les plus petits (Cf exemple de la sous-section précédente).

1.2.3 Erreurs d'arrondi sur un produit

Supposons que l'on souhaite calculer un produit $P = u_1\,u_2 \ldots u_n$ de n réels positifs dans $F(\beta, t, e_{\min}, e_{\max})$. On calcule alors les produits P_i par la récurrence $P_0 = 1$, $P_i = P_{i-1}\,u_i$. Si l'on suppose les u_i connus exactement, alors les erreurs d'arrondi ΔP_i commises sur le calcul des produits P_i vérifient $\Delta P_i \leq \Delta P_{i-1}\,u_i + \delta\,(S_{i-1}\,u_i) = \Delta P_{i-1}\,u_i + \delta P_i$ où $|\delta| < u$. L'erreur globale sur $P = P_n$ vérifie donc

$$\Delta P \leq (k-1)\delta\,P_n.$$

On voit donc que contrairement au cas de l'addition, la majoration de l'erreur ne dépend pas de l'ordre des facteurs.

1.2.4 Phénomènes de compensation

Les phénomènes de compensation sont ceux qui se produisent lorsque l'on tente de soustraire des nombres très proches. Nous illustrons ces phénomènes sur deux exemples et donnons des astuces pour les contourner.

Exemple 1 : On considère l'expression $E = \sqrt{x+1} - \sqrt{x}$ avec $x > 0$. Sous MATLAB, en calculant E pour $x = 10^9$, on va obtenir $1,5811.10^{-5}$ mais pour $x = 10^{16}$, on va obtenir 0 ! Si l'on remarque que $E = \frac{1}{\sqrt{x+1}+\sqrt{x}}$, alors, en utilisant cette nouvelle formule, on trouvera $E = 1,5811.10^{-5}$ pour $x = 10^9$ et $E = 5,000.10^{-9}$ pour $x = 10^{16}$.

Exemple 2 : On considère l'équation du second degré $x^2 - 1634\,x + 2 = 0$. Supposons que les calculs soient effectués avec 10 chiffres significatifs. Les formules habituelles donnent $\Delta' = (\frac{1634}{2})^2 - 2 = 667487$, $\sqrt{\Delta'} = 816,9987760$, d'où les solutions

$$x_1 = \frac{1634}{2} + \sqrt{\Delta'} = 817 + 816,9987760 = 1633,998776,$$

$$x_2 = \frac{1634}{2} - \sqrt{\Delta'} = 817 - 816,9987760 = 0,0012240.$$

On voit donc qu'en procédant ainsi on a une perte de 5 chiffres significatifs sur x_2. Pour y remédier, on peut utiliser la relation $x_1\, x_2 = 2$ et calculer

$$x_2 = \frac{2}{x_1} = \frac{2}{1633,998776} = 0,001223991125.$$

1.2.5 Phénomènes d'instabilité numérique

Les phénomènes d'instabilité numérique sont des phénomènes d'amplification d'erreur d'arrondi. Ils se produisent en général pour des calculs récurrents ou itératifs. Nous illustrons ces phénomènes sur deux exemples.

Exemple 1 : On considère l'intégrale

$$I_n = \int_0^1 \frac{x^n}{10+x}\,\mathrm{d}x, \quad n \in \mathbb{N},$$

que l'on cherche à évaluer numériquement. Un calcul direct montre que $I_0 = \ln(\frac{11}{10})$. De plus, on a

$$I_n = \int_0^1 \frac{x}{10+x}\,x^{n-1}\,\mathrm{d}x = \int_0^1 \left(1 - \frac{10}{10+x}\right) x^{n-1}\,\mathrm{d}x = \frac{1}{n} - 10\,I_{n-1}.$$

On peut donc calculer successivement les valeurs de I_n en utilisant la récurrence $I_0 = \ln(\frac{11}{10})$, $I_n = \frac{1}{n} - 10\,I_{n-1}$. Numériquement, cela conduit à des résultats très mauvais. Cela provient du fait que l'erreur d'arrondi ΔI_n sur le calcul de I_n vérifie $\Delta I_n \approx 10\,\Delta I_{n-1}$, si l'on néglige l'erreur d'arrondi sur $\frac{1}{n}$. On voit donc que l'erreur croit exponentiellement : l'erreur sur I_0 est multipliée par 10^n sur I_n. Par conséquent cette formule de récurrence ne peut pas nous permettre de calculer la valeur de I_{36} par exemple. Pour remédier à ce problème, on peut *renverser la récurrence* c'est-à-dire considérer la formule :

$$I_{n-1} = \frac{1}{10}\left(\frac{1}{n} - I_n\right).$$

Toujours en négligeant l'erreur d'arrondi sur $\frac{1}{n}$, on obtient alors $\Delta I_{n-1} \approx \frac{1}{10}\,\Delta I_n$. En utilisant l'encadrement $10 \leq 10 + x \leq 11$ pour $x \in [0,1]$, on montre que

$$\frac{1}{11\,(n+1)} \leq I_n \leq \frac{1}{10\,(n+1)}.$$

L'approximation $I_n \approx \frac{1}{11\,(n+1)}$ nous permet alors de calculer une valeur de départ pour notre récurrence renversée. Par exemple, si l'on part de $I_{46} \approx \frac{1}{11\,(46+1)}$, on obtiendra pour I_{36} une erreur relative meilleure que 10^{-10}.

On constate ici l'importance du coefficient d'amplification d'erreur pour ce genre de calcul.

Exemple 2 : On considère la suite définie par :

$$\begin{cases} u_0 &= 2, \\ u_1 &= -4, \\ u_n &= 111 - \dfrac{1130}{u_{n-1}} + \dfrac{3000}{u_{n-1}\,u_{n-2}}, \end{cases}$$

introduite par J.-M. Muller. On peut alors montrer que la limite de cette suite est égale à 6 et malgré cela, quelque soit le système et la précision utilisés, cette suite semblera tendre vers 100. L'explication de ce phénomène étrange est assez simple : la solution générale de la récurrence $u_n = 111 - \dfrac{1130}{u_{n-1}} + \dfrac{3000}{u_{n-1}\,u_{n-2}}$ est donnée par :

$$u_n = \frac{\alpha\,100^{n+1} + \beta\,6^{n+1} + \gamma\,5^{n+1}}{\alpha\,100^{n} + \beta\,6^{n} + \gamma\,5^{n}},$$

où α, β et γ dépendent des valeurs initiales u_0 et u_1. Par conséquent, si $\alpha \neq 0$, la suite converge vers 100 et sinon (si $\beta \neq 0$) la suite converge vers 6. Dans notre exemple, les valeurs initiales $u_0 = 2$ et $u_1 = -4$ correspondent à $\alpha = 0$, $\beta = -3$ et $\gamma = 4$. Par conséquent, la limite exacte de la suite est 6. Mais à cause des erreurs d'arrondi, même les premiers termes calculés seront différents des termes exacts et donc la valeur de α correspondant à ces termes calculés sera très petite mais non-nulle ce qui suffira à faire en sorte que la suite converge vers 100 au lieu de 6.

1.2.6 Erreur amont et erreur aval

Considérons un problème que l'on résout à l'aide d'un algorithme numérique et appelons f la fonction qui fait correspondre à l'entrée x de l'algorithme la solution algébrique $y = f(x)$. En pratique, compte tenu des erreurs d'arrondis, étant donnée une entrée x, nous allons obtenir une sortie $\overline{y}$ qui sera distincte de la solution algébrique $y = f(x)$. L'*erreur aval* est alors la différence entre le résultat $\overline{y}$ obtenu et la solution algébrique y. L'*erreur amont* ou *erreur inverse* est le plus petit δx tel que la solution algébrique $f(x + \delta x)$ correspondant à l'entrée $x + \delta x$ soit égale à $\overline{y}$. Ces deux erreurs sont liées par le *conditionnement* (voir Chapitre 3) : l'erreur aval étant du même ordre que l'erreur amont multipliée par le conditionnement. L'erreur amont est en général plus intéressante que l'erreur aval car elle nous renseigne sur le problème qui est réellement résolu par l'algorithme numérique. De plus, en pratique, nous ne connaissons en général qu'une valeur approchée de l'entrée (par exemple obtenue par une mesure).

1.2.7 Outils théoriques de l'analyse d'erreurs

Considérons la formule $(x \times y) + z$ pour trois réels x, y et z d'un domaine $F(\beta, t, e_{\min}, e_{\max})$.

On a alors :

$$\begin{aligned} \mathrm{fl}((x \times y) + z) &= [\mathrm{fl}(x \times y) + z]\,(1 + \delta_1) \\ &= [(x \times y)\,(1 + \delta_2) + z]\,(1 + \delta_1) \\ &= (x \times y)\,(1 + \delta_2)\,(1 + \delta_1) + z\,(1 + \delta_1), \end{aligned}$$

d'après le théorème 1.7, avec de plus $|\delta_i| < u$, pour $i = 1, 2$.

Lemme 1.8. *Si pour tout $i = 1, \ldots, k$, on a $|\delta_i| < u$ et si $k\,u < 1$, alors il existe θ_k tel que $|\theta_k| \leq \frac{k\,u}{1-k\,u}$ et $\prod_{i=1}^{k}(1+\delta_i) \leq 1+\theta_k$.*

Démonstration. La démonstration se fait par récurrence. □

En utilisant la notation $< k >$ « = » $\prod_{i=1}^{k}(1+\delta_i)$ avec $< j > . < k > = < j+k >$, on a alors

$$\begin{array}{rcl} \mathrm{fl}((x \times y) + z) & = & (x \times y) < 2 > + z < 1 > \\ & \leq & (x \times y)\,(1 + \frac{2\,u}{1-2\,u}) + z\,(1 + \frac{u}{1-u}). \end{array}$$

Chapitre 2

Résolution d'un système d'équations linéaires (Partie 1) : méthodes directes

Beaucoup de problèmes se réduisent à la résolution numérique d'un système d'équations linéaires. Il existe deux grandes classes de méthodes pour résoudre ce type de systèmes :

1. les méthodes *directes* qui déterminent explicitement la solution après un nombre fini d'opérations arithmétiques,
2. les méthodes *itératives* qui consistent à générer une suite qui converge vers la solution du système.

Remarquons que les méthodes itératives ne s'appliquent que dans le cas de systèmes à coefficients dans $\mathbb{R}$ ou $\mathbb{C}$ mais pas dans le cas des corps finis $\mathbb{F}_p$. Notons qu'il existe aussi des méthodes intermédiaires telles que les méthodes de *Splitting* ou de décomposition incomplètes, et des méthodes probabilistes comme celle de Monte-Carlo qui ne seront pas abordées dans ce cours.
Dans ce deuxième chapitre nous nous intéressons aux méthodes directes. Les méthodes itératives seront abordées au chapitre 4.

2.1 Introduction et motivation

2.1.1 Objet

On considère un système (S) de n équations à n inconnues de la forme

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

Les données sont les coefficients $a_{i,j}$ du système qui appartiennent à un corps $\mathbb{K}$ avec $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$ ainsi que les coefficients du second membre $b_1, \ldots, b_n$. Les inconnues sont $x_1, \ldots, x_n$ qui

appartiennent à $\mathbb{K}$. Un système est dit *homogène* si son second membre est nul c'est-à-dire si tous les b_i sont nuls. Nous écrirons ce système sous forme matricielle

$$(S) \quad A\,x = b,$$

$$\text{avec } A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n,1} & \dots & \dots & a_{n,n} \end{pmatrix} \in \mathbb{M}_{n\times n}(\mathbb{K}),\ x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{K}^n \text{ et } b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{K}^n.$$

Dans tout ce chapitre, on supposera que la matrice A **est inversible !**

2.1.2 Motivation

Le lecteur peut se demander pourquoi le problème de la résolution d'un tel système se pose alors que les formules de Cramer nous donnent la solution d'un tel système (S) :

$$\forall\, i \in \{1,\dots,n\}, \quad x_i = \frac{\begin{vmatrix} a_{1,1} & \dots & a_{1,(i-1)} & b_1 & a_{1,(i+1)} & \dots & a_{1,n} \\ \vdots & & & \vdots & & & \vdots \\ a_{n,1} & \dots & a_{n,(i-1)} & b_n & a_{n,(i+1)} & \dots & a_{n,n} \end{vmatrix}}{\det(A)}.$$

Pour comprendre le problème, essayons de compter le nombre d'opérations nécessaires pour calculer la solution en utilisant ces formules de Cramer. Nous devons calculer $n+1$ déterminants de taille n. On sait que le déterminant de A est donné par la formule

$$\det(A) = \sum_{\sigma\in\mathcal{S}_n} \epsilon_\sigma\, a_{\sigma(1),1}\, a_{\sigma(2),2} \cdots a_{\sigma(n),n},$$

où $\mathcal{S}_n$ est l'ensemble des permutations de $\{1,\dots,n\}$ et ϵ_σ est la signature de la permutation σ ($\epsilon_\sigma \in \{-1,1\}$). L'ensemble $\mathcal{S}_n$ contient $n!$ éléments donc le calcul de $\det(A)$ se ramène à $n!-1$ additions et $n!\,(n-1)$ multiplications soit au total $n!-1+n!\,(n-1) = n\,n!-1$ opérations à virgule flottante. Comme nous avons $n+1$ déterminants à calculer, le nombre d'opérations nécessaires pour résoudre le système à l'aide des formules de Cramer est de $(n+1)\,(n\,n!-1)$ opérations à virgule flottante qui est équivalent quand la dimension n tend vers l'infini à $n\,(n+1)!$. Essayons d'évaluer cette quantité lorsque $n = 100$. Pour ceci on peut utiliser la formule de Stirling $n! \sim n^{n+\frac{1}{2}}\,\mathrm{e}^{-n}\,\sqrt{2\,\pi}$ qui est très précise pour évaluer la factorielle :

$$\begin{aligned} 100.101! &= 100.101.100! \\ &\simeq 100.101.100^{100,5}.\mathrm{e}^{-100}.\sqrt{2\,\pi} \\ &\simeq 100.101.100^{100,5}.10^{-43,43}.\sqrt{2\,\pi} \qquad (\log_{10}(\mathrm{e}) \simeq 0,4343) \\ &\simeq 10^{205-44}.10^{0,57}.\sqrt{2\,\pi} \\ &\simeq 9,4.10^{161}. \end{aligned}$$

Par exemple, avec ordinateur fonctionnant à 100 megaflops (flops = opérations à virgule flottante par secondes - floating point operations per second), il faudrait environ 3.10^{146} années pour résoudre notre système !

On en déduit donc qu'il est impossible d'utiliser les formules de Cramer pour des systèmes de grande taille. En pratique, on utilise les méthodes directes pour des systèmes de dimension peu élevée ($n \leq 100$). Pour de très grands systèmes qui peuvent par exemple apparaitre en économie ou pour l'approximation d'équations aux dérivées partielles (voir Chapitre 4), on préfère les méthodes itératives.

2.1.3 Résolution d'un système triangulaire

L'idée des méthodes directes est de se ramener à la résolution d'un (ou de deux) système(s) triangulaire(s).

Supposons que A soit une matrice triangulaire supérieure. Le système s'écrit alors :

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

Puisque A est supposée inversible, aucun des $a_{i,i}$ n'est nul et on peut résoudre ce système en utilisant *l'algorithme de substitution rétrograde (ou substitution arrière)* suivant :

Entrèes : $A = (a_{i,j})_{1\leq i,j\leq n} \in \mathbb{M}_{n\times n}(\mathbb{K})$ triangulaire supérieure et $b = (b_i)_{1\leq i\leq n} \in \mathbb{K}^n$.
Sortie : $x = (x_i)_{1\leq i\leq n} \in \mathbb{K}^n$ tel que $A\,x = b$.

1. $x_n = \frac{b_n}{a_{n,n}}$;
2. Pour i de $n-1$ à 1 par pas de -1, faire :
   - $x_i = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=i+1}^{n} a_{i,j}\,x_j \right)$;
3. Retourner $x = (x_i)_{1\leq i\leq n} \in \mathbb{K}^n$.

Exemple : On considère le système triangulaire supérieur

$$(S) \begin{cases} x_1 & + & 2\,x_2 & + & 5\,x_3 & = & 1, \\ & & -4\,x_2 & - & 16\,x_3 & = & -\frac{5}{2}, \\ & & & & -17\;x_3 & = & -\frac{17}{8}. \end{cases}$$

En résolvant la troisième équation, on trouve $x_3 = \frac{1}{8}$. La deuxième équation donne alors $x_2 = \frac{-5/2+16\,x_3}{-4} = \frac{1}{8}$ et finalement la première équation fournit $x_1 = \frac{1-2\,x_2-5\,x_3}{1} = \frac{1}{8}$.

De façon analogue, lorsque A est triangulaire inférieure, on obtient *l'algorithme de substitution progressive (ou substitution avant).*

Proposition 2.1. *La résolution d'un système d'équations linéaires triangulaire se fait en n^2 opérations à virgule flottante.*

Démonstration. Calculer x_n nécessite 1 opération (division), calculer x_{n-1} nécessite 3 opérations (une multiplication, une soustraction et une division), et calculer x_i nécessite $2\,(n-i)+1$ opérations ($n-i$ multiplications, $(n-i-1)$ additions, 1 soustraction et 1 division). Au total, le nombre d'opérations est donc

$$\sum_{i=1}^{n}(2\,(n-i)+1) = 2\sum_{i=1}^{n}(n-i)+n = 2\sum_{j=0}^{n-1} j + n = 2\,\frac{(n-1)\,n}{2} + n = n^2.$$

□

Proposition 2.2. *Soient A, $B \in \mathbb{M}_{n\times n}(\mathbb{K})$ deux matrices triangulaires supérieures. On a alors les résultats suivants :*

1. *$A\,B$ est triangulaire supérieur ;*
2. *Si A et B sont à diagonale unité (*i.e.*, n'ont que des* 1 *sur la diagonale), alors $A\,B$ est à diagonale unité ;*
3. *Si A est inversible, alors A^{-1} est aussi triangulaire supérieure ;*
4. *Si A est inversible et à diagonale unité, alors A^{-1} est aussi à diagonale unité.*

Démonstration. Exercice. □

2.1.4 Les méthodes directes étudiées

Dans la suite de ce chapitre, nous allons considérer les trois méthodes directes suivantes pour la résolution de (S) : $A\,x = b$:

1. Méthode de Gauss : le principe est de réduire le système à $(M\,A)\,x = M\,b$ avec $M\,A$ triangulaire supérieure sans calculer explicitement M. On se ramène donc à la résolution d'un système triangulaire supérieur. Cette méthode est associée à la factorisation $A = L\,U$ de la matrice A avec L triangulaire inférieure (**L**ower) et U triangulaire supérieure (**U**pper). Étant donnée une telle factorisation $A = L\,U$, on peut résoudre le système $A\,x = b$ en résolvant successivement les deux systèmes triangulaires $L\,y = b$ puis $U\,x = y$.
   En MATLAB, on peut appliquer la méthode de Gauss pour résoudre (S) : $A\,x = b$ en tapant `x = A \ b` (attention, le symbole entre la matrice A et le vecteur colonne b est « backslash » et non un « slash » !) et calculer la factorisation LU d'une matrice en utilisant `[L,U]=lu(A)`.

2. Méthode de Cholesky associée à la factorisation de Cholesky $A = R^T R$ avec R triangulaire supérieure. Cette méthode est valable pour une matrice A symétrique et définie positive (voir Définitions 2.14 et 2.16). On résout alors $A\,x = b$ en résolvant successivement les systèmes triangulaires $R^T\,y = b$ puis $R\,x = y$.
En MATLAB, on peut calculer la factorisation de Cholesky d'une matrice A en tapant `R=chol(A)`. On obtient un message d'erreur lorsque A n'est pas symétrique définie positive.

3. Méthode de Householder associée à la factorisation $A = Q\,R$ avec R triangulaire supérieure et Q orthogonale (voir Définition 2.20). La matrice Q est un produit de $n-1$ « matrices de Householder » H_i (voir Définition 2.19). Le système $A\,x = b$ s'écrit alors $H_{n-1}\cdots H_2\,H_1\,A\,x = H_{n-1}\cdots H_2\,H_1\,b$ que l'on résout facilement grâce au fait que le produit $H_{n-1}\cdots H_2\,H_1\,A$ est une matrice triangulaire supérieure.
En MATLAB, on calcule la factorisation QR d'une matrice A en tapant `[Q,R]=qr(A)`.

2.2 Méthode de Gauss et factorisation LU

2.2.1 Description de la méthode

On considère le système linéaire (S) : $A\,x = b$ en supposant toujours que A est inversible et on pose $b^{(1)} = b$ et $A^{(1)} = A = (a_{i,j}^{(1)})_{1\leq i,j\leq n}$. Le système (S) s'écrit alors $A^{(1)}\,x = b^{(1)}$ que l'on note $S^{(1)}$.

Étape 1 : Puisque A est inversible, quitte à échanger la première ligne de $A^{(1)}$ avec une autre, on peut supposer que $a_{1,1}^{(1)} \neq 0$. Le nombre $a_{1,1}^{(1)}$ est le premier *pivot* de l'élimination de Gauss. Pour $i = 2,\ldots,n$, on multiplie la première équation de $(S^{(1)})$ par $g_{i,1} = \frac{a_{i,1}^{(1)}}{a_{1,1}^{(1)}}$ et on retranche l'équation obtenue à la $i$ème équation de $(S^{(1)})$. La $i$ème ligne L_i de $(S^{(1)})$ devient donc $L_i - g_{i,1}\,L_1$. On obtient alors un nouveau système $(S^{(2)})$: $A^{(2)}\,x = b^{(2)}$ avec :

$$\left\{\begin{array}{lcll} a_{1,j}^{(2)} & = & a_{1,j}^{(1)}, & j = 1,\ldots,n,\\ a_{i,1}^{(2)} & = & 0, & i = 2,\ldots,n,\\ a_{i,j}^{(2)} & = & a_{i,j}^{(1)} - g_{i,1}\,a_{1,j}^{(1)}, & i,j = 2,\ldots,n,\\ b_1^{(2)} & = & b_1^{(1)}, & \\ b_i^{(2)} & = & b_i^{(1)} - g_{i,1}\,b_1^{(1)}, & i = 2,\ldots,n. \end{array}\right.$$

La matrice $A^{(2)}$ et le vecteur $b^{(2)}$ sont donc de la forme :

$$A^{(2)} = \begin{pmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \dots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & \dots & a_{2,n}^{(2)} \\ 0 & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & a_{n,2}^{(2)} & \dots & a_{n,n}^{(2)} \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{pmatrix}.$$

Étape k : On a ramené le système à $(S^{(k)})$: $A^{(k)}\, x = b^{(k)}$ avec

$$A^{(k)} = \begin{pmatrix} a_{1,1}^{(1)} & & \dots & \dots & a_{1,k}^{(1)} & \dots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & & & a_{2,k}^{(2)} & \dots & a_{2,n}^{(2)} \\ 0 & 0 & a_{3,3}^{(3)} & & a_{3,k}^{(3)} & \dots & a_{3,n}^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & a_{k,k}^{(k)} & \dots & a_{k,n}^{(k)} \\ \vdots & & \vdots & 0 & a_{k+1,k}^{(k)} & \dots & a_{k+1,n}^{(k)} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & a_{n,k}^{(k)} & \dots & a_{n,n}^{(k)} \end{pmatrix}.$$

On peut alors se ramener au cas $a_{k,k}^{(k)} \neq 0$ et $a_{k,k}^{(k)}$ est le $k$ième *pivot* de l'élimination de Gauss. Par le même principe qu'à l'étape 1 et en utilisant $g_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}$ pour $i > k$, on obtient alors $(S^{(k+1)})$: $A^{(k+1)}\, x = b^{(k+1)}$ avec

$$A^{(k+1)} = \begin{pmatrix} a_{1,1}^{(1)} & & \dots & \dots & a_{1,k+1}^{(1)} & \dots & \dots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & & & a_{2,k}^{(2)} & \dots & \dots & a_{2,n}^{(2)} \\ 0 & 0 & a_{3,3}^{(3)} & & a_{3,k}^{(3)} & \dots & \dots & a_{3,n}^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots & & & \vdots \\ 0 & \dots & 0 & 0 & a_{k,k}^{(k)} & \dots & \dots & a_{k,n}^{(k)} \\ \vdots & & \vdots & 0 & 0 & a_{k+1,k+1}^{(k+1)} & \dots & a_{k+1,n}^{(k+1)} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & a_{n,k+1}^{(k+1)} & \dots & a_{n,n}^{(k+1)} \end{pmatrix}.$$

Étape $n-1$: Le système $(S^{(n)})$: $A^{(n)}\,x = b^{(n)}$ obtenu est triangulaire supérieure avec

$$A^{(n)} = \begin{pmatrix} a_{1,1}^{(1)} & & \cdots & \cdots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & & & a_{2,n}^{(2)} \\ 0 & 0 & a_{3,3}^{(3)} & & a_{3,n}^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & a_{n,n}^{(n)} \end{pmatrix},$$

et peut donc être résolu par l'algorithme de substitution rétrograde de la sous-section 2.1.3.

Exemple : Considérons le système suivant :

$$(S) = (S^{(1)}) \begin{cases} x_1 + 2\,x_2 + 5\,x_3 = 1, \\ 3\,x_1 + 2\,x_2 - x_3 = \frac{1}{2}, \\ 5\,x_2 + 3\,x_3 = 1. \end{cases}$$

Le premier pivot de l'élimination de Gauss est donc $a_{1,1}^{(1)} = 1$ et on a $g_{2,1}^{(1)} = 3$, $g_{3,1}^{(1)} = 0$. La première étape fournit donc

$$(S^{(2)}) \begin{cases} x_1 + 2\,x_2 + 5\,x_3 = 1, \\ -4\,x_2 - 16\,x_3 = -\frac{5}{2}, \\ 5\,x_2 + 3\,x_3 = 1. \end{cases}$$

Le second pivot de l'élimination de Gauss est donc $a_{2,2}^{(2)} = -4$ et on a $g_{3,2}^{(2)} = -\frac{5}{4}$. On obtient donc le système

$$(S^{(3)}) \begin{cases} x_1 + 2\,x_2 + 5\,x_3 = 1, \\ -4\,x_2 - 16\,x_3 = -\frac{5}{2}, \\ -17\,x_3 = -\frac{17}{8}. \end{cases}$$

On résout alors ce système triangulaire supérieur par l'algorithme de substitution rétrograde pour trouver : $x_1 = x_2 = x_3 = \frac{1}{8}$ (voir l'exemple de la sous-section 2.1.3).

2.2.2 Point de vue numérique : stratégies de choix du pivot

Au cours de l'exécution de l'élimination de Gauss, si on tombe sur un pivot nul, alors on permute la ligne en question avec une ligne en dessous pour se ramener à un pivot non nul (ceci est toujours possible car A est supposée inversible). Cependant, certains choix de pivots peuvent s'avérer plus judicieux que d'autres.

Exemple :

Considérons le système (S) : $A\,x = b$ où

$$A = \begin{pmatrix} \alpha & 1 \\ 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix},$$

avec α réel non nul. On suppose de plus $\alpha \neq 1$ de sorte que A est inversible. La solution de ce système est alors donnée par $x_1^* = \frac{1}{1-\alpha}$, $x_2^* = \frac{1-2\,\alpha}{1-\alpha}$. Supposons maintenant que α est « très petit » $(0 \ll \alpha < 1)$ et appliquons l'élimination de Gauss décrite dans la sous-section précédente. Le premier pivot est alors α et $g_{2,1} = \frac{1}{\alpha}$. La première étape transforme donc le système (S) en $(S^{(2)})$: $A^{(2)}\,x = b^{(2)}$ avec

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

La deuxième équation entraîne $-\frac{1}{\alpha}\,x_2 \approx -\frac{1}{\alpha}$ d'où $x_2 \approx 1$ et la première nous donne alors $x_1 \approx 0$ ce qui est faux. L'erreur ne provient pas seulement du fait que α est « très petit » car si on multiplie la première ligne par une puissance de 10 quelconque, on va trouver la même erreur. Notons $x_2 = x_2^* + \delta x_2$ où $|\delta x_2|$ est l'erreur absolue sur x_2. On a alors

$$x_1 = \frac{1 - x_2}{\alpha} = \frac{1 - x_2^*}{\alpha} - \frac{\delta x_2}{\alpha},$$

et on voit donc que l'erreur $\delta x_1 = \frac{1}{\alpha}\,\delta x_2$ sur x_1 est très amplifiée par rapport à celle sur x_2. L'anomalie provient du déséquilibre entre les coefficients de x_1 et x_2 de la ligne du pivot. Pour y remédier, échangeons maintenant les deux lignes de notre système et appliquons l'élimination de Gauss avec 1 comme premier pivot. On obtient alors

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

qui entraîne alors $x_2 \approx 1$ et $x_1 \approx 1$ ce qui est correct.

Élimination de Gauss à pivot partiel

À l'étape k, on échange les lignes k et k' $(k' \geq k)$ de $A^{(k)}$ de telle sorte que : $|a_{k,k}^{(k)}| = \max\{|a_{i,k}^{(k)}|,\, i \geq k\}$.
Par exemple, pour le système (S) de la sous-section précédente, à l'étape 1, on va permuter les lignes 1 et 2 et considérer

$$(S') : \begin{cases} 3\,x_1 & + & 2\,x_2 & - & x_3 & = & \frac{1}{2}, \\ x_1 & + & 2\,x_2 & + & 5\,x_3 & = & 1, \\ & & 5\,x_2 & + & 3\,x_3 & = & 1. \end{cases}$$

Élimination de Gauss à pivot total

À l'étape k, on échange à la fois les lignes k et k' ($k' \geq k$) et les colonnes k et k'' ($k'' \geq k$) de telle sorte que : $|a_{k,k}^{(k)}| = \max\{|a_{i,j}^{(k)}|,\ i \geq k,\ j \geq k\}$.
Attention ! Si on fait des échanges de colonnes cela modifie l'ordre des composantes du vecteur solution x donc il faut penser à rétablir le bon ordre des composantes à la fin.
Par exemple, pour le système (S) de la sous-section précédente, à l'étape 1, on va permuter les colonnes 1 et 3 et considérer

$$(S') : \left\{ \begin{array}{rcrcrcl} 5\,x_3 & + & 2\,x_2 & + & x_1 & = & 1, \\ -x_3 & + & 2\,x_2 & + & 3\,x_1 & = & \frac{1}{2}, \\ 3\,x_3 & + & 5\,x_2 & & & = & 1. \end{array} \right.$$

2.2.3 Lien avec la factorisation LU d'une matrice

Définition 2.3. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$. On appelle* factorisation LU *de A une factorisation $A = L\,U$ avec $L \in \mathbb{M}_{n\times n}(\mathbb{K})$ triangulaire inférieure et $U \in \mathbb{M}_{n\times n}(\mathbb{K})$ triangulaire supérieure.*

Lemme 2.4. *Avec les notations de la sous-section 2.2.1, à l'étape k de l'élimination de Gauss, on a $A^{(k+1)} = G_k\,A^{(k)}$ où*

$$G_k = \begin{pmatrix} 1 & & (0) & & 0 & \dots & 0 \\ & & \ddots & & \vdots & & \vdots \\ & (0) & & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & -g_{k+1,k} & 1 & & (0) \\ \vdots & & \vdots & \vdots & & \ddots & \\ 0 & \dots & 0 & -g_{n,k} & (0) & & 1 \end{pmatrix}, \quad g_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}},\ i = k+1, \dots, n.$$

On a de plus $b^{(k+1)} = G_k\,b^{(k)}$.

Démonstration. Il suffit d'effectuer le produit $G_k\,A^{(k)}$ et de vérifier que l'on retrouve $A^{(k+1)}$. □

Définition 2.5. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$. Les* mineurs fondamentaux D_k, $k = 1, \dots, n$ de A *sont les déterminants des sous-matrices de A formées par les k premières lignes et les k premières colonnes de A : $D_k = \det\left((a_{i,j})_{1\leq i,j\leq k}\right)$ pour $k = 1, \dots, n$.*

Théorème 2.6. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ une matrice carrée inversible. Les propriétés suivantes sont équivalentes :*

(i) L'élimination de Gauss s'effectue sans permutation de lignes ;

(ii) Il existe $L \in \mathbb{M}_{n\times n}(\mathbb{K})$ triangulaire inférieure inversible et $U \in \mathbb{M}_{n\times n}(\mathbb{K})$ triangulaire supérieure inversible telles que $A = L\,U$;

(iii) Tous les mineurs fondamentaux de A sont non nuls.

Démonstration. Pour la suite de ce cours, la partie de la démonstration qui nous intéresse est l'implication $(i) \Rightarrow (ii)$ qui nous permettra de calculer la factorisation LU de A. D'après le lemme 2.4, on a $A^{(n)} = G_{n-1}\, G_{n-2} \cdots G_1\, A$. La matrice $G_{n-1}\, G_{n-2} \cdots G_1$ étant inversible, on peut donc écrire $A = (G_{n-1}\, G_{n-2} \cdots G_1)^{-1}\, A^{(n)}$. Les matrices G_k étant triangulaires inférieures, la proposition 2.2 affirme que $(G_{n-1}\, G_{n-2} \cdots G_1)^{-1}$ est aussi triangulaire inférieure. De plus, par construction $A^{(n)}$ est triangulaire supérieure d'où le résultat en posant $L = (G_{n-1}\, G_{n-2} \cdots G_1)^{-1}$ et $U = A^{(n)}$. □

Corollaire 2.7. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ une matrice carrée inversible. La matrice A admet une factorisation LU si et seulement si tous ses mineurs fondamentaux sont non nuls.*

Le lemme suivant nous permet d'expliciter la matrice L de la factorisation LU de A obtenue à partir de l'élimination de Gauss.

Lemme 2.8. *Avec les notations précédentes, on a*

$$(G_{n-1}\, G_{n-2} \cdots G_1)^{-1} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ g_{2,1} & 1 & \ddots & & \vdots \\ g_{3,1} & g_{3,2} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & 1 \end{pmatrix}.$$

Démonstration. Faire le calcul. □

Corollaire 2.9. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ une matrice carrée inversible. Si tous les mineurs fondamentaux de A sont non nuls, alors avec les notations précédentes, l'élimination de Gauss fournit la factorisation LU de A suivante :*

$$A = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ g_{2,1} & 1 & \ddots & & \vdots \\ g_{3,1} & g_{3,2} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} a_{1,1}^{(1)} & & \dots & \dots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & & & a_{2,n}^{(2)} \\ 0 & 0 & a_{3,3}^{(3)} & & a_{3,n}^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & a_{n,n}^{(n)} \end{pmatrix}.$$

On remarque que la matrice L obtenue est à diagonale unité.

Exemple : En reprenant l'exemple de la sous-section 2.2.1, on trouve :

$$\underbrace{\begin{pmatrix} 1 & 2 & 5 \\ 3 & 2 & -1 \\ 0 & 5 & 3 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & -\frac{5}{4} & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} 1 & 2 & 5 \\ 0 & -4 & -16 \\ 0 & 0 & -17 \end{pmatrix}}_{U}.$$

Proposition 2.10. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ une matrice carrée inversible admettant une factorisation LU. Alors il existe une unique factorisation LU de A avec L à diagonale unité.*

Démonstration. Si A admet une factorisation LU, alors l'existence d'une factorisation LU avec L à diagonale unité est claire d'après le corollaire 2.9. Supposons maintenant que A admette deux factorisations LU : $A = L_1 U_1$ et $A = L_2 U_2$ avec L_1 et L_2 à diagonale unité. L'égalité $L_1 U_1 = L_2 U_2$ implique $U_1 U_2^{-1} = L_1^{-1} L_2$. D'après la proposition 2.2, $U_1 U_2^{-1}$ est triangulaire supérieure et $L_1^{-1} L_2$ est triangulaire inférieure à diagonale unité. La seule possibilité pour que ces deux matrices soient égales est donc $U_1 U_2^{-1} = L_1^{-1} L_2 = \mathbb{I}_n$ ce qui implique $L_1 = L_2$ et $U_1 = U_2$. □

Lorsque A admet une factorisation LU, la résolution du système d'équations linéaires $(S) : A\,x = b$ se ramène à la résolution de deux systèmes linéaires triangulaires. En effet :

$$A\,x = b \Longleftrightarrow L\,U\,x = b \Longleftrightarrow \left\{ \begin{array}{rcl} L\,y & = & b, \\ U\,x & = & y. \end{array} \right.$$

En pratique, on résout donc d'abord $L\,y = b$ puis connaissant y on résout $U\,x = y$.

Définition 2.11. *On appelle* matrice de permutation associée à une permutation $\sigma \in \mathcal{S}_n$, *la matrice* $\mathcal{P}_\sigma = (\delta_{i\sigma(j)})_{1\leq i,j\leq,n}$ *où* $\delta_{ij} = 1$ *si* $i = j$ *et* $\delta_{ij} = 0$ *si* $i \neq j$.

Par exemple si l'on considère la permutation, $\sigma : (1,2,3,4,5) \mapsto (3,2,5,1,4)$, on obtient la « matrice de permutation élémentaire »

$$\mathcal{P}_\sigma = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Multiplier une matrice A à gauche (resp. à droite) par une matrice de permutation revient alors à permuter les lignes (resp. les colonnes) de la matrice. Par exemple, en multipliant une matrice à gauche par la matrice $\mathcal{P}_\sigma$ ci-dessus, la troisième ligne devient la première, la seconde ligne reste inchangée, la cinquième ligne devient la troisième .... Les matrices de permutation sont orthogonales (voir Définition 2.20) c'est-à-dire que $\forall\,\sigma \in \mathcal{S}_n,\ \mathcal{P}_\sigma^{-1} = \mathcal{P}_\sigma^T$.

Le corollaire 2.7 nous donne une condition nécessaire et suffisante pour qu'une matrice inversible admette une factorisation LU. Lorsque cette factorisation LU n'existe pas, on peut tout de même utiliser le théorème suivant :

Théorème 2.12. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ une matrice carrée inversible. Il existe une matrice de permutation $\mathcal{P}$ telle que $\mathcal{P}\,A$ admette une factorisation LU.*

Démonstration. Admis pour ce cours. □

Notons que dans ce cas, on a :

$$A\,x = b \Longleftrightarrow \mathcal{P}\,A\,x = \mathcal{P}\,b \Longleftrightarrow L\,U\,x = \mathcal{P}\,b \Longleftrightarrow \left\{ \begin{array}{rcl} L\,y & = & \mathcal{P}\,b, \\ U\,x & = & y. \end{array} \right.$$

En pratique, on résout donc d'abord $L\,y = \mathcal{P}\,b$ puis connaissant y on résout $U\,x = y$.

2.2.4 Coût de l'algorithme

Résolution d'un système via l'élimination de Gauss

Nous allons nous intéresser au nombre d'opérations à virgule flottante nécessaires à la résolution d'un système (S) : $A\,x = b$ en utilisant l'élimination de Gauss puis en résolvant le système triangulaire $A^{(n)}\,x = b^{(n)}$.

Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ une matrice carrée inversible et supposons que A admette une factorisation LU. À l'étape k de l'élimination de Gauss, on doit faire $n-k$ divisions pour calculer les $g_{i,k}$ puis nous avons $(n-k)^2$ coefficients $a_{i,j}^{(k+1)}$ à calculer et $(n-k)$ coefficients $b_i^{(k+1)}$. Cela nécessite donc $(n-k)\,(n-k+1)$ multiplications puis $(n-k)\,(n-k+1)$ soustractions. Au total, pour toute l'élimination de Gauss, nous aurons donc à faire

$$\begin{array}{rcl} \sum_{k=1}^{n-1}(n-k) + 2\,\sum_{k=1}^{n-1}(n-k)\,(n-k+1) & = & \sum_{k=0}^{n-1} k + 2\,\sum_{k=0}^{n-1} k\,(k+1) \\ & = & \frac{n\,(n-1)}{2} + 2\,(\sum_{k=1}^{n-1} k + \sum_{k=1}^{n-1} k^2) \\ & = & \frac{n\,(n-1)}{2} + 2\,(\frac{n\,(n-1)}{2} + \frac{n\,(n-1)\,(2\,n-1)}{6}) \\ & = & n\,(n-1)\,\left(\frac{3+6+2\,(2\,n-1)}{6}\right) \\ & = & \left(\frac{n\,(n-1)\,(4\,n+7)}{6}\right) \\ & = & \left(\frac{4\,n^3+3\,n^2-7\,n}{6}\right) \end{array}$$

opérations à virgule flottante. D'après la proposition 2.1, la résolution d'un système triangulaire coûte n^2 opérations à virgule flottantes donc au total, la résolution du système (S) par cette méthode coûtera $G(n) = n^2 + \left(\frac{4\,n^3+3\,n^2-7\,n}{6}\right) = \frac{4\,n^3+9\,n^2-7\,n}{6}$ opérations à virgule flottante. Lorsque n tend vers l'infini, on a $G(n) \sim \frac{2\,n^3}{3}$ opérations à virgule flottante. Notons que d'après ce qui précède, cette estimation représente aussi le coût asymptotique du calcul de la factorisation LU d'une matrice. On a donc obtenu :

Lemme 2.13. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ une matrice carrée inversible. Résoudre un système linéaire (S) : $A\,x = b$ via l'élimination de Gauss nécessite un nombre d'opérations à virgule flottante équivalent à $\frac{2\,n^3}{3}$ lorsque n tend vers l'infini. Ce coût asymptotique est aussi celui du calcul de la factorisation $L\,U$ de A.*

Pour comparer avec le résultat obtenu en utilisant les formules de Cramer, si l'on suppose que $n = 100$, alors cela donne environ $6,6.10^5$ opérations à virgule flottante et avec un

ordinateur fonctionnant à 100 megaflops, cela prendra moins de 7 millièmes de secondes. Cela ne se fait donc pas à la main mais votre ordinateur personnel pourra le faire en quelques (dizaines de) secondes.

Faut-il inverser une matrice ?

Nous admettrons qu'étant donnée la factorisation LU de A, le coût du calcul de l'inverse A^{-1} de A lorsque n tend vers l'infini est de $\frac{4\,n^3}{3}$ opérations à virgule flottante. Au total, lorsque n tend vers l'infini, il faut donc $2\,n^3$ opérations à virgule flottante pour calculer l'inverse de A. D'après ce qui précède cela signifie donc qu'asymptotiquement (*i.e.*, lorsque n tend vers l'infini), il faut 3 fois plus d'opérations à virgule flottante pour calculer l'inverse de A que pour résoudre le système linéaire $A\,x = b$ en utilisant l'élimination de Gauss. Cela implique qu'il ne faut pas calculer l'inverse d'une matrice pour résoudre un système linéaire.

Cas de la résolution de plusieurs systèmes de même matrice A

Soit $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ une matrice carrée inversible et supposons que l'on ait à résoudre K systèmes linéaires avec la même matrice A et N seconds membres $b^{[1]}, \ldots, b^{[K]}$. Si l'on applique l'élimination de Gauss à chacun de ces systèmes pour les résoudre, alors d'après ce qui précède cela nous coûtera $K\,\frac{4\,n^3+9\,n^2-7\,n}{6}$ opérations à virgule flottante. Si maintenant on calcule une fois pour toute une factorisation LU de A (via l'élimination de Gauss) et que l'on résout ensuite successivement les $2\,K$ systèmes triangulaires cela ne nous coutera que $\left(\frac{4\,n^3+3\,n^2-7\,n}{6}\right) + 2\,K\,n^2$ opérations à virgule flottante ce qui est asymptotiquement meilleur. Notons que si l'on calcule une fois pour toute l'inverse A^{-1} de A et que l'on résout ensuite chaque système en posant $x^{[i]} = A^{-1}\,b^{[i]}$ cela coutera $2\,n^3$ opérations pour le calcul de l'inverse puis $n\,(2\,n-1)$ opérations pour calculer chacun des K produits $A^{-1}\,b^{[i]}$ soit au total $2\,n^3 + 2\,K\,n^2$ ce qui est moins avantageux que d'utiliser la factorisation LU.

2.3 Méthode de Cholesky

La méthode de Cholesky est une alternative à l'élimination de Gauss qui s'applique aux matrices symétriques et définies positives.

Définition 2.14. *Une matrice $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ est dite* symétrique *si elle est égale à sa transposée,* i.e.*, $A^T = A$.*

Définition 2.15. *Soit $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$.* Le produit scalaire canonique sur $\mathbb{K}^n$ *est défini comme l'application $\langle .\,,.\rangle : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K},\ (u,v) \mapsto \langle u, v\rangle$ qui vérifie :*

- *Si $\mathbb{K} = \mathbb{R}$, $\langle u, v\rangle = v^T\,u = \sum_{i=1}^n u_i\,v_i$ (produit scalaire euclidien),*
- *Si $\mathbb{K} = \mathbb{C}$, $\langle u, v\rangle = \overline{v}^T\,u = \sum_{i=1}^n u_i\,\overline{v_i}$ (produit scalaire hermitien).*

Définition 2.16. *Une matrice $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ est dite* définie positive*, resp.* semi définie positive *si pour tout $x \in \mathbb{R}^n$ non nul, on a $\langle A\,x, x\rangle > 0$, resp. $\langle A\,x, x\rangle \geq 0$.*

On montre facilement les propriétés suivantes :

1. Une matrice définie positive est inversible ;
2. Si $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ est inversible, alors $A^T A$ est symétrique et définie positive ;
3. Si $A = (a_{i,j})_{1\leq i,j\leq n} \in \mathbb{M}_{n\times n}(\mathbb{K})$ est définie positive, alors $a_{i,i} > 0$ pour tout $i = 1, \ldots, n$.

Théorème 2.17 (Caractérisation des matrices symétriques définies positives)**.** *Une matrice réelle $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ est symétrique définie positive si et seulement si il existe une matrice $L = (l_{i,j})_{1\leq i,j\leq n} \in \mathbb{M}_{n\times n}(\mathbb{R})$ triangulaire inférieure inversible telle que $A = L\,L^T$. De plus, si pour tout $i = 1, \ldots, n$, $l_{i,i} \geq 0$, alors L est unique.*

Démonstration. Admis pour ce cours. □

Remarquons que si L est triangulaire inférieure, alors L^T est triangulaire supérieure de sorte que la factorisation de Cholesky $A = L\,L^T$ peut être vue comme un cas particulier de la factorisation LU d'une matrice étudiée dans la section précédente.

On donne maintenant *l'algorithme de Cholesky* qui étant donnée une matrice carrée réelle A symétrique et définie positive calcule une matrice L telle que $A = L\,L^T$. On admettra que cet algorithme est correct.

Entrèe : $A = (a_{i,j})_{1\leq i,j\leq n} \in \mathbb{M}_{n\times n}(\mathbb{R})$ symétrique et définie positive.
Sortie : $L = (l_{i,j})_{1\leq i,j\leq n} \in \mathbb{M}_{n\times n}(\mathbb{R})$ tel que $A = L\,L^T$.

1. $l_{1,1} = \sqrt{a_{1,1}}$;
2. Pour i de 2 à n par pas de 1, faire :
   - $l_{i,1} = \frac{a_{i,1}}{l_{1,1}}$;
3. Pour j de 2 à n par pas de 1, faire :
   - Pour i de 1 à $j-1$ par pas de 1, faire :
     $l_{i,j} = 0$;
   - $l_{j,j} = \sqrt{a_{j,j} - \sum_{k=1}^{j-1} l_{j,k}^2}$;
   - Pour i de $j+1$ à n par pas de 1, faire :
     $l_{i,j} = \frac{a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}\,l_{j,k}}{l_{j,j}}$;
4. Retourner $L = (l_{i,j})_{1\leq i,j\leq n} \in \mathbb{M}_{n\times n}(\mathbb{R})$.

Notons que dans ce cas, comme pour la factorisation LU, on a :

$$A\,x = b \Longleftrightarrow L\,L^T\,x = b \Longleftrightarrow \left\{ \begin{array}{rcl} L\,y & = & b, \\ L^T\,x & = & y. \end{array} \right.$$

En pratique, on résout donc d'abord $L\,y = b$ puis connaissant y on résout $L^T\,x = y$.

Proposition 2.18. *L'algorithme de Cholesky décrit ci-dessus nécessite n extractions de racines carrées et un nombre d'opérations à virgule flottante équivalent à $\frac{n^3}{3}$ lorsque n tend vers l'infini.*

Démonstration. Admis pour ce cours. □

On voit donc que cet algorithme requiert asymptotiquement « presque » (cela dépendra de la méthode utilisée pour le calcul des racines carrées qui peut-être plus ou moins rapide) deux fois moins d'opérations à virgule flottante que celui calculant la factorisation LU via l'élimination de Gauss. Par conséquent, il est conseillé de l'utiliser lorsque A est réelle symétrique et définie positive.

2.4 Méthode de Householder et factorisation QR

Dans cette section, on suppose que $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ est une matrice réelle inversible.

2.4.1 Transformation (élémentaire) de Householder

Définition 2.19. *On appelle* matrice (élémentaire) de Householder *une matrice H de la forme $H_u = \mathbb{I}_n - 2\,u\,u^T$, où $u \in \mathbb{R}^n$ est un vecteur unitaire c'est-à-dire de norme 1 pour la* norme associée au produit scalaire canonique sur $\mathbb{R}^n$ *définie par $\|u\| = \sqrt{\langle u, u\rangle}$.*

Par exemple, pour $n = 3$, on peut considérer le vecteur $u = \frac{1}{\sqrt{6}} \begin{pmatrix} -1 & 1 & 2 \end{pmatrix}^T$ qui vérifie bien $\|u\| = 1$. On obtient alors la matrice de Householder $H_u = \frac{1}{3} \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & -1 \end{pmatrix}$.

Définition 2.20. *Une matrice $A \in \mathbb{M}_{n\times n}(\mathbb{K})$ est dite* orthogonale *si elle est réelle,* i.e.*, $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ et si $A\,A^T = A^T\,A = \mathbb{I}_n$.*

Par exemple que les matrices de permutation (voir Définition 2.11) sont orthogonales.

Proposition 2.21. *Toute matrice de Householder H est symétrique et orthogonale.*

Démonstration. Facile à vérifier. □

Proposition 2.22. *Pour tout vecteur $u \in \mathbb{R}^n$ tel que $\|u\| = 1$, on a $H_u\,u = -u$. De plus, si $v \in \mathbb{R}^n$ est orthogonal à u,* i.e.*, $\langle u, v\rangle = 0$, alors $H_u\,v = v$.*

Démonstration. Facile à vérifier. □

Géométriquement, la matrice H_u représente donc la symétrie orthogonale par rapport au plan u^T orthogonal à u.

Lemme 2.23. *Soit x et y deux vecteurs de $\mathbb{R}^n$ tels que $x \neq y$ et $\|x\| = \|y\|$. Alors il existe un vecteur unitaire $u \in \mathbb{R}^n$ tel que $H_u\,x = y$.*

Démonstration. Prendre $u = \frac{x-y}{\|x-y\|}$. □

2.4.2 Principe de la méthode de Householder

La méthode de Householder pour la résolution d'un système linéaire $A\,x = b$ est basée sur les deux propositions suivantes que nous admettrons.

Proposition 2.24. *Soit v un vecteur non nul de $\mathbb{R}^n$. Alors il existe une matrice de Householder H et un réel α tels que $H\,v = \alpha\,e_1$, où $e_1 = (1, 0, \ldots, 0)^T$ est le premier vecteur de la base canonique de $\mathbb{R}^n$.*

Proposition 2.25. *Soit $u = (u_i)$ un vecteur unitaire de $\mathbb{R}^n$ tel que $u_1 = \cdots = u_p = 0$ pour $p < n$. On décompose alors u en deux blocs : $u = (0 \quad z)^T$ avec $z \in \mathbb{R}^{n-p}$. La matrice de Householder H_u se décompose alors par blocs de la manière suivante : $H_u = \begin{pmatrix} \mathbb{I}_p & 0 \\ 0 & H_z \end{pmatrix}$.*

2.4.3 Exemple de résolution d'un système linéaire par la méthode de Householder

Considérons le système linéaire suivant :

$$(S) : \begin{cases} 2\,x_1 & + & x_2 & + & 2\,x_3 & = & 1, \\ x_1 & + & x_2 & + & 2\,x_3 & = & 1, \\ 2\,x_1 & + & x_2 & + & x_3 & = & 1. \end{cases}$$

Étape 1 : On considère le vecteur obtenu à partir de la première colonne de la matrice A de (S), *i.e.*, $a_1 = (2 \quad 1 \quad 2)^T$. Le but de cette étape est de trouver la matrice H apparaissant dans la proposition 2.24 et correspondant à $v = a_1$ de sorte qu'en multipliant la matrice A du système (S) à gauche par H on obtienne des zéros sous le premier coefficient. Pour ceci, en concordance avec la preuve du lemme 2.23, on introduit le vecteur $v_1 = \frac{a_1}{\|a_1\|} - e_1 = \frac{1}{3}(-1 \quad 1 \quad 2)^T$, le vecteur $u_1 = \frac{v_1}{\|v_1\|} = \frac{1}{\sqrt{6}} \begin{pmatrix} -1 & 1 & 2 \end{pmatrix}^T$ et on considère la matrice élémentaire de Householder $H_{u_1} = \frac{1}{3} \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & -1 \end{pmatrix}$. On a alors :

$$(S) \ : \ A\,x = b \Longleftrightarrow H_{u_1}\,A\,x = H_{u_1}\,b \Longleftrightarrow \begin{pmatrix} 9 & 5 & 8 \\ 0 & 1 & 4 \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ -1 \end{pmatrix}.$$

Étape 2 : On se place maintenant dans $\mathbb{R}^2$ et on considère le vecteur $a_2 = (1 \quad -1)^T$ auquel nous allons appliquer la même technique qu'au vecteur a_1 à l'étape précédente. On utilisera ensuite la proposition 2.25 pour obtenir une matrice de Householder de la bonne taille. On pose alors $z_2' = \frac{a_2}{\|a_2\|} - e_1'$ où $e_1' = (1, 0)$ est le premier vecteur de la base canonique de $\mathbb{R}^2$ et $z_2 = \frac{z_2'}{\|z_2'\|}$. On considère ensuite la matrice de Householder $H_{z_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$

puis $u_2 = \begin{pmatrix} 0 & z_2 \end{pmatrix}^T$ et $H_{u_2} = \begin{pmatrix} 1 & 0 \\ 0 & H_{z_2} \end{pmatrix}$ (Voir Proposition 2.25). On a alors :

$$A\,x = b \Longleftrightarrow H_{u_2}\,H_{u_1}\,A\,x = H_{u_2}\,H_{u_1}\,b \Longleftrightarrow \begin{pmatrix} 9 & 5 & 8 \\ 0 & \sqrt{2} & \frac{5}{\sqrt{2}} \\ 0 & 0 & -\frac{3}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ \sqrt{2} \\ 0 \end{pmatrix},$$

d'où en résolvant ce système triangulaire $x = (0 \quad 1 \quad 0)^T$.

2.4.4 Factorisation QR d'une matrice

Définition 2.26. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ une matrice carrée réelle inversible. On appelle* factorisation QR de A *une factorisation de la forme $A = Q\,R$ avec $Q \in \mathbb{M}_{n\times n}(\mathbb{R})$ orthogonale et $R \in \mathbb{M}_{n\times n}(\mathbb{R})$ triangulaire supérieure.*

En généralisant la méthode expliquée dans la sous-section précédente pour une matrice carrée réelle inversible quelconque de taille n, on obtient $n-1$ matrices de Householder $H_1, \ldots, H_{n-1}$ telles que le produit $R = H_{n-1}\,H_{n-2}\cdots H_1\,A$ est triangulaire supérieure. On pose alors $Q = (H_{n-1}\,H_{n-2}\cdots H_1)^{-1}$ de sorte que $A = Q\,R$. La matrice Q ainsi défini étant orthogonale, on a obtenu une factorisation QR de A. On obtient alors le résultat suivant :

Théorème 2.27. *Pour toute matrice réelle $A \in \mathbb{M}_{n\times n}(\mathbb{R})$, il existe une matrice orthogonale $Q \in \mathbb{M}_{n\times n}(\mathbb{R})$ produit d'au plus $(n-1)$ matrices de Householder et une matrice triangulaire supérieure $R \in \mathbb{M}_{n\times n}(\mathbb{R})$ telles que $A = Q\,R$.*

Proposition 2.28. *La méthode de Householder pour résoudre un système linéaire nécessite un nombre d'opérations à virgule flottante équivalent à $\frac{4\,n^3}{3}$ lorsque n tend vers l'infini.*

Démonstration. Admis pour ce cours. □

Le coût de cette méthode est donc relativement élevé comparé à l'élimination de Gauss ou à la méthode de Cholesky. Cependant un avantage de cette méthode est qu'elle est plus *stable numériquement* que les deux méthodes citées précédemment.

Notons enfin que la factorisation QR d'une matrice existe aussi pour des matrices rectangulaires et qu'elle est utilisée pour des problèmes de *moindres carrés*.

Factorisation QR et moindres carrés linéaires. Le problème des moindres carrés linéaires est le suivant : étants donné une matrice $A \in \mathbb{M}_{m\times n}(\mathbb{R})$ et un vecteur $b \in \mathbb{R}^m$, minimiser la quantité $\|A\,x - b\|_2^2$.

Ce problème peut être résolu en utilisant la factorisation QR de la matrice $A \in \mathbb{M}_{m\times n}(\mathbb{R})$ en procédant comme suit : la matrice A étant rectangulaire (on supposera ici $m > n$), on partitionne les matrices Q et R de la manière suivante :

$$A = (Q_1 \quad Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix},$$

où $Q_1 \in \mathbb{M}_{m\times n}(\mathbb{R})$, $Q_2 \in \mathbb{M}_{m\times(m-n)}(\mathbb{R})$ et $R_1 \in \mathbb{M}_{n\times n}(\mathbb{R})$. En remplaçant A par sa factorisation QR, on a :

$$\|A\,x - b\|_2^2 = \|Q\,R\,x - b\|_2^2\,.$$

On admettra qu'en appliquant une transformation orthogonale au vecteur des résidus $A\,x - b$ on ne modifie pas la norme euclidienne du vecteur et que le problème de minimisation conduit au même résultat. On multiplie alors le résidu par Q^T et on est ramené à minimiser

$$\left\|Q^T\,Q\,R\,x - Q^T\,b\right\|_2^2 = \left\|R\,x - Q^T\,b\,\right\|_2^2 = \left\|\begin{pmatrix} R_1 \\ 0 \end{pmatrix} x - \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} b\,\right\|_2^2,$$

ce qui revient à minimiser

$$\left\|R_1\,x - Q_1^T\,b\right\|_2^2 + \left\|Q_2^T\,b\right\|_2^2\,.$$

On résout alors le système triangulaire $R_1\,x = Q_1^T\,b$ et la somme des résidus au carré correspond à $\left\|Q_2^T\,b\right\|_2^2$.

Dans les problèmes qui apparaissent en économétrie ou statistique, on est aussi intéressé par la matrice des variances et covariances $\sigma^2\,(A^T\,A)^{-1}$, où $\sigma^2 = \dfrac{\|A\,x - b\|_2^2}{m-n}$. Pour obtenir cette matrice, on a donc besoin d'évaluer le résidu à la solution x trouvée et de calculer l'inverse de $A^T\,A$. Si on possède la factorisation QR de A, alors $A^T\,A = R^T\,Q^T\,Q\,R = R^T\,R$ car Q est orthogonale. On doit donc inverser la matrice $R^T\,R$. De plus, avec les notations précédentes, si on note $d = Q_2^T\,b$, on peut écrire $\sigma^2 = \dfrac{d^T\,d}{m-n}$.

Remarque : cela n'entre pas dans le cadre de ce cours mais il existe une méthode efficace et numériquement stable pour calculer l'inverse de matrices de la forme $A^T\,A$.

Chapitre 3

Conditionnement d'une matrice pour la résolution d'un système linéaire

3.1 Normes matricielles

3.1.1 Normes vectorielles

Soit E un espace vectoriel sur $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$.

Définition 3.1. *On appelle* norme sur E *une application* $\|.\| : E \to \mathbb{R}_+$ *telle que :*

- $\forall\, x \in E,\ (\|x\| = 0 \Rightarrow x = 0)$;
- $\forall\, \lambda \in \mathbb{K},\ \forall\, x \in E,\ \|\lambda\, x\| = |\lambda|\, \|x\|$;
- $\forall\, (x, y) \in E^2,\ \|x + y\| \leq \|x\| + \|y\|$.

Les exemples classiques de normes sur $\mathbb{R}^n$ sont les normes $\|.\|_1$, $\|.\|_2$ et $\|.\|_\infty$ définies par :

$$\forall\, x = (x_i)_{1 \leq i \leq n} \in \mathbb{R}^n, \quad \|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}} = \langle x, x \rangle^{\frac{1}{2}}, \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

3.1.2 Normes matricielles et normes subordonnées

Définition 3.2. *Une norme* $\|.\|$ *sur* $\mathbb{M}_{n \times n}(\mathbb{K})$ *est une* norme matricielle *si elle vérifie :* $\forall\, (A, B) \in \mathbb{M}_{n \times n}(\mathbb{K})^2, \quad \|A\, B\| \leq \|A\|\, \|B\|$.

L'exemple fondamental est celui des normes dites *subordonnées* qui sont associées à une norme vectorielle.

Théorème et Définition 3.3. *Soit* $\|.\|$ *une norme vectorielle sur* $\mathbb{K}^n$. *Pour toute matrice* $A \in \mathbb{M}_{n \times n}(\mathbb{K})$, *on définit* $\|.\|_M : \mathbb{M}_{n \times n}(\mathbb{K}) \to \mathbb{R}_+$ *par* $\|A\|_M = \sup_{x \in \mathbb{K}^n \setminus \{0\}} \frac{\|A\, x\|}{\|x\|}$. *Alors* $\|.\|_M$ *est une norme matricielle. Elle est dite* norme subordonnée à la norme vectorielle $\|.\|$.

Dans la suite on notera indifféremment $\|.\|$ pour une norme vectorielle ou une norme matricielle.

Démonstration. On utilise le lemme suivant qui découle de la définition d'une norme subordonnée :

Lemme 3.4. *Soit $\|.\|$ la norme subordonnée à une norme vectorielle $\|.\|$. Alors, pour tout vecteur $x \in \mathbb{K}^n$ et pour toute matrice $M \in \mathbb{M}_{n\times n}(\mathbb{K})$, on a $\|M\,x\| \leq \|M\|\,\|x\|$.*

Soit $(A, B) \in \mathbb{M}_{n\times n}(\mathbb{K})^2$ et $x \neq 0$. On a :

$$\frac{\|A\,B\,x\|}{\|x\|} \leq \frac{\|A\|\,\|B\,x\|}{\|x\|} \leq \frac{\|A\|\,\|B\|\,\|x\|}{\|x\|}.$$

d'où le résultat. □

Les normes subordonnées respectivement aux normes vectorielles $\|.\|_1$, $\|.\|_2$ et $\|.\|_\infty$ de $\mathbb{R}^n$ sont données par : $\forall\, A = (a_{i,j})_{1\leq i,j\leq n} \in \mathbb{M}_{n\times n}(\mathbb{K})$:

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|, \quad \|A\|_2 = \sqrt{\rho(A\,A^*)}, \quad \|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}|,$$

où $A^* = \overline{A}^T$ désigne *la matrice adjointe de A* et $\rho(M)$ désigne *le rayon spectral d'une matrice M*, *i.e.*, le maximum des modules des valeurs propres de M.

Notons que ces trois normes matricielles sont équivalentes : pour $A \in \mathbb{M}_{n\times n}(\mathbb{K})$, on a :

$$\frac{1}{\sqrt{n}}\|A\|_2 \leq \|A\|_1 \leq \sqrt{n}\,\|A\|_2, \quad \frac{1}{\sqrt{n}}\|A\|_\infty \leq \|A\|_2 \leq \sqrt{n}\,\|A\|_\infty, \quad \frac{1}{n}\|A\|_1 \leq \|A\|_\infty \leq n\,\|A\|_1.$$

3.2 Conditionnement d'une matrice

3.2.1 Exemple classique

Cet exemple est du à R. S. Wilson. Considérons le système linéaire (S) : $A\,x = b$ avec

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}.$$

Remarquons tout d'abord que la matrice A est symétrique, que son déterminant vaut 1 et que la solution de (S) est donnée par $x = (1 \quad 1 \quad 1 \quad 1)^T$.

Premier cas : b est perturbé. Perturbons légèrement le second membre b et considérons

$$b' = \begin{pmatrix} 32,1 \\ 22,9 \\ 33,1 \\ 30,9 \end{pmatrix}.$$

Si on résout le système (S') : $A\,x' = b'$, on trouve $x' = (9,2 \quad -12,6 \quad 4,5 \quad -1,1)^T$. La « petite » perturbation sur le second membre b entraîne donc une « forte » perturbation sur la solution du système.

D'une manière générale, on considère les deux systèmes $A\,x = b$ et $A\,(x + \delta x) = b + \delta b$. On a donc $A\,\delta x = \delta b$ de sorte que $\delta x = A^{-1}\,\delta b$ et on obtient la majoration suivante de l'erreur absolue sur la solution : $\|\delta x\| \leq \|A^{-1}\|.\|\delta b\|$. Or $A\,x = b$ donc $\|b\| \leq \|A\|.\|x\|$. Finalement, on obtient une majoration de l'erreur relative sur la solution en fonction de l'erreur relative sur la donnée :

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

Notons que cette majoration est optimale dans le sens où il n'existe pas de borne plus petite qui soit valable pour tout système. En effet, prenons $A = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$, $b = (1 \quad 0)^T$, et $\delta b = (0 \quad \frac{1}{2})^T$. La solution de $A\,x = b$ est alors $x = (1 \quad 0)^T$ et celle de $A\,\delta x = \delta b$ est $\delta x = (0 \quad 1)^T$. On a donc $\frac{\|\delta x\|}{\|x\|} = 1$, $\frac{\|\delta b\|}{\|b\|} = \frac{1}{2}$. Or $\|A^{-1}\|.\|A\| = 2$ donc la borne est atteinte.

Deuxième cas : A est perturbée. De façon analogue, si on perturbe légèrement la matrice A et que l'on considère

$$A'' = \begin{pmatrix} 10 & 7 & 8,1 & 7,2 \\ 7,08 & 5,04 & 6 & 5 \\ 8 & 5,98 & 9,89 & 9 \\ 6,99 & 4,99 & 9 & 9,98 \end{pmatrix},$$

alors la solution du système (S'') : $A''\,x'' = b$ est $x'' = (-81 \quad 107 \quad -34 \quad 22)^T$.

D'une manière générale, on considère les deux systèmes $A\,x = b$ et $(A + \Delta A)\,(x + \delta x) = b$. Il vient donc $\delta x = A^{-1}\,\Delta A\,(x + \delta x)$ d'où

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

3.2.2 Définition du conditionnement

Définition 3.5. *Soit $\|.\|$ une norme matricielle subordonnée et A une matrice inversible. Le nombre* $\mathrm{Cond}(A) = \|A^{-1}\|.\|A\|$ *s'appelle* le conditionnement de A relatif à la norme $\|.\|$.

Ce nombre mesure la « sensibilité » de la solution par rapport aux données du problème. Une matrice est bien conditionnée si $\mathrm{Cond}(A) \approx 1$ et mal conditionnée si $\mathrm{Cond}(A) \gg 1$. Le fait que les normes matricielles $\|.\|_1$, $\|.\|_2$ et $\|.\|_\infty$ soient équivalentes implique la même propriété d'équivalence pour les conditionnements associés : pour toute matrice $A \in \mathbb{M}_{n\times n}(\mathbb{K})$, on a :

$$\frac{1}{n}\mathrm{Cond}_2(A) \leq \mathrm{Cond}_1(A) \leq n\,\mathrm{Cond}_2(A), \quad \frac{1}{n}\mathrm{Cond}_\infty(A) \leq \mathrm{Cond}_2(A) \leq n\,\mathrm{Cond}_\infty(A),$$

$$\frac{1}{n^2}\mathrm{Cond}_1(A) \leq \mathrm{Cond}_\infty(A) \leq n^2\,\mathrm{Cond}_1(A).$$

Des exemples de matrices bien conditionnées sont les matrices de la forme

$$A = \begin{pmatrix} 4 & 1 & 0 & \dots & 0 \\ 1 & 4 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \dots & 0 & 1 & 4 \end{pmatrix},$$

qui vérifient que, quelque soit la dimension n de A, $\mathrm{Cond}_\infty(A) \leq 3$. Inversement, les *matrices de Hilbert* H_n et les *matrices de Vandermonde* V_n respectivement définies par

$$H_n = \left(\frac{1}{i+j-1}\right)_{1\leq i,j\leq n}, \quad V_n = \left(\binom{j}{n}^{i-1}\right)_{1\leq i,j\leq n},$$

sont très mal conditionnées : leur conditionnement pour la norme $\|.\|_\infty$ vérifie :

n	$\mathrm{Cond}(H_n)$	$\mathrm{Cond}(V_n)$
2	27	8
4	$2,8.10^4$	$5,6.10^2$
6	$2,9.10^7$	$3,7.10^4$

Proposition 3.6. *Soit A une matrice réelle et considérons la norme matricielle subordonnée $\|A\|_2 = \sqrt{\rho(A\,A^*)}$. On a*

$$\mathrm{Cond}_2(A) = \|A^{-1}\|_2.\|A\|_2 = \sqrt{\frac{\rho(A\,A^T)}{\sigma(A\,A^T)}},$$

où $\sigma(M)$ désigne le minimum des modules des valeurs propres de M. En particulier, si A est symétrique, alors on obtient $\mathrm{Cond}_2(A) = \frac{\rho(A)}{\sigma(A)}$.

Démonstration. Admis pour ce cours. □

3.2.3 Estimation théorique de l'erreur a priori

Premier cas : b est perturbé

Théorème 3.7. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ inversible et $b \in \mathbb{R}^n$ tels que $A\,x = b$ et $A\,(x+\delta x) = b+\delta b$ avec $x \neq 0$. Alors on a :*

$$\frac{1}{\mathrm{Cond}(A)}.\frac{\|\delta b\|}{\|b\|} \leq \frac{\|\delta x\|}{\|x\|} \leq \mathrm{Cond}(A).\frac{\|\delta b\|}{\|b\|},$$

Démonstration. La majoration a déjà été prouvée. La minoration s'obtient directement à partir des inégalités $\|\delta b\| \leq \|A\|.\|\delta x\|$ et $\|x\| \leq \|A^{-1}\|.\|b\|$. □

Deuxième cas : A est perturbée

Théorème 3.8. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ inversible, $b \in \mathbb{R}^n$ et $\Delta A \in \mathbb{M}_{n\times n}(\mathbb{R})$ tels que $\|A^{-1}\|.\|\Delta A\| < 1$. Alors $A + \Delta A$ est inversible. De plus si on suppose $A\,x = b$ et $(A + \Delta A)\,(x + \delta x) = b$ avec $x \neq 0$, alors on a :*

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\mathrm{Cond}(A).\frac{\|\Delta A\|}{\|A\|}}{1 - \mathrm{Cond}(A).\frac{\|\Delta A\|}{\|A\|}}.$$

Démonstration. On a $(A + \Delta A) = A\,(\mathbb{I}_n + A^{-1}\,\Delta A)$ avec par hypothèse $\|A^{-1}\,\Delta A\| < 1$. Le lemme suivant (admis pour ce cours - la preuve n'est pas difficile) implique que $(A + \Delta A)$ est inversible.

Lemme 3.9. *Soit $\|.\|$ une norme matricielle subordonnée et $B \in \mathbb{M}_{n\times n}(\mathbb{R})$ une matrice telle que $\|B\| < 1$. Alors la matrice $\mathbb{I}_n + B$ est inversible et on a : $\|(\mathbb{I}_n + B)^{-1}\| \leq \frac{1}{1-\|B\|}$.*

On a de plus $(A+\Delta A)^{-1} = (\mathbb{I}_n + A^{-1}\,\Delta A)^{-1}\,A^{-1}$. Maintenant, Si $A\,x = b$ et $(A+\Delta A)\,(x+\delta x) = b$, alors $(A + \Delta A)\,\delta x = -\Delta A.x$ d'où $\delta x = -(\mathbb{I}_n + A^{-1}\,\Delta A)^{-1}\,A^{-1}\,\Delta A.x$. On a donc

$$\frac{\|\delta x\|}{\|x\|} \leq \|\mathbb{I}_n + A^{-1}\,\Delta A)^{-1}\|.\|A^{-1}\|.\|\Delta A\|,$$

d'où d'après le lemme 3.9 ci-dessus,

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{1}{1 - \|A^{-1}\,\Delta A\|}.\|A^{-1}\|.\|\Delta A\| \leq \frac{\mathrm{Cond}(A).\frac{\|\Delta A\|}{\|A\|}}{1 - \mathrm{Cond}(A).\frac{\|\Delta A\|}{\|A\|}}.$$

□

Troisième cas : A et b sont perturbés

Théorème 3.10. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ inversible, $b \in \mathbb{R}^n$ et $\Delta A \in \mathbb{M}_{n\times n}(\mathbb{R})$ vérifiant $\|A^{-1}\|.\|\Delta A\| < 1$. Si l'on suppose que $A\,x = b$ et $(A + \Delta A)\,(x + \delta x) = b + \delta b$ avec $x \neq 0$, alors on a :*

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\mathrm{Cond}(A)}{1 - \mathrm{Cond}(A).\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right).$$

Démonstration. Exercice : on utilise les mêmes techniques que pour les preuves précédentes. □

3.2.4 Estimation théorique de l'erreur a posteriori

Étant donné un système linéaire $A\,x = b$, on va maintenant estimer, en fonction du conditionnement, l'erreur commise sur la solution réellement calculée. Soit x la solution exacte et y la solution obtenue par la machine. On pose $r = A\,y - b$; r s'appelle le *résidu*. On a alors le résultat suivant :

Théorème 3.11. *Avec les notations précédentes, si $x \neq 0$, alors*

$$\|y - x\| \leq \mathrm{Cond}(A).\frac{\|r\|}{\|b\|}.\|x\|.$$

Démonstration. On a $r = A\,y - b = A(\,y - x)$ d'où $y - x = A^{-1}\,r$ et $\|y - x\| \leq \|A^{-1}\|.\|r\|$ ce qui implique $\|y - x\| \leq \mathrm{Cond}(A).\frac{\|r\|}{\|A\|}$. Or $A\,x = b$ implique $\|A\| \geq \frac{\|b\|}{\|x\|}$ d'où le résultat. □

On voit donc que si le conditionnement est grand, l'erreur relative peut être grande. Cette majoration n'est pas très facile à utiliser car le conditionnement est en général inconnu. Soit C une approximation de A^{-1} (que l'on peut par exemple calculer via la méthode d'élimination de Gauss) et posons $R = A\,C - \mathbb{I}_n$. On a alors le résultat suivant :

Théorème 3.12. *Avec les notations précédentes, si $\|R\| < 1$, alors*

$$\|y - x\| \leq \frac{\|r\|.\|C\|}{1 - \|R\|}.$$

Démonstration. On a vu dans la preuve précédente que $\|y - x\| \leq \|A^{-1}\|.\|r\|$. Il suffit alors de remarquer que $A^{-1} = C\,(\mathbb{I}_n + R)^{-1}$ et d'utiliser le lemme 3.9.

□

Chapitre 4

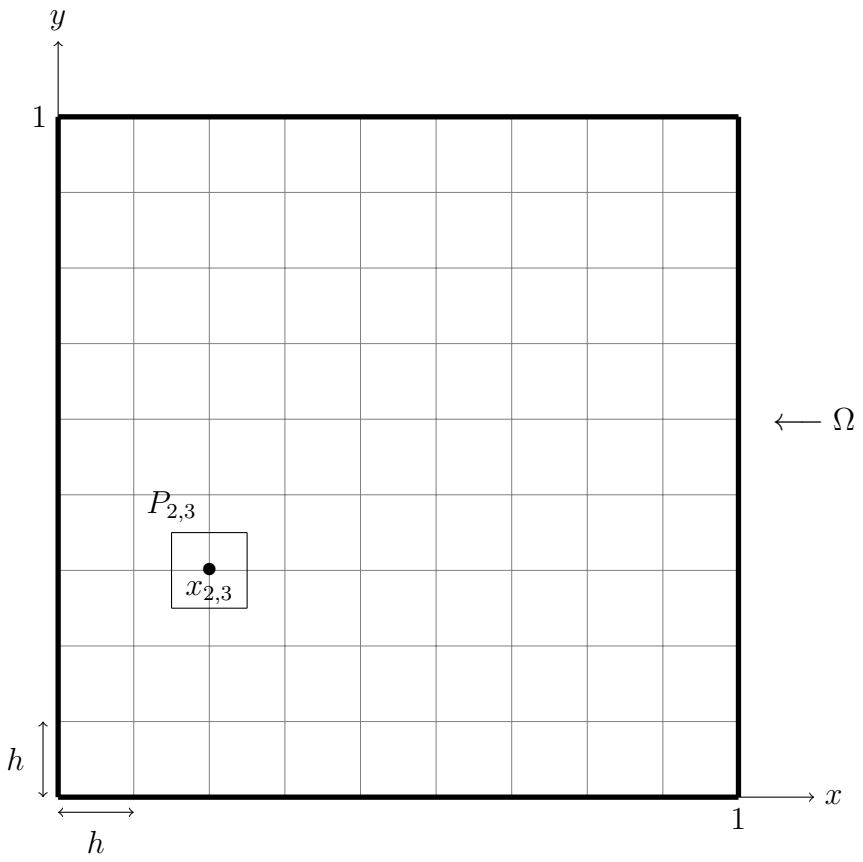
Résolution d'un système d'équations linéaires (Partie 2) : méthodes itératives

4.1 Motivation

Les méthodes itératives deviennent indispensables dès que la taille n du système est très grande. En effet, les méthodes directes exigent un nombre d'opérations à virgule flottante de l'ordre de n^3 lorsque n tend vers l'infini ce qui les rend lentes pour de grandes valeurs de n. De tels systèmes apparaissent par exemple dans les techniques de résolution numérique d'équations aux dérivées partielles. Les matrices des systèmes obtenus sont en général « creuses » (c'est-à-dire qu'elles ont beaucoup de 0) et (semi) définies positives. Voici un exemple classique. Étant donnée une fonction $f : \mathbb{R}^2 \to \mathbb{R}$, on se propose de trouver une solution approchée $\tilde{u} : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ du problème suivant :

$$\left\{ \begin{array}{rcll} -\Delta\tilde{u} & = & f, & \forall (x,y) \in \Omega =]0,1[\times]0,1[, \\ \tilde{u} & = & 0, & \forall (x,y) \in \partial\Omega, \end{array} \right.$$

où $\Delta\tilde{u} = \frac{\partial^2\tilde{u}}{\partial x^2} + \frac{\partial^2\tilde{u}}{\partial y^2}$ désigne le laplacien de la fonction $\tilde{u}$ et $\partial\Omega$ la frontière de Ω.
Pour ce faire, on se donne un réel $h > 0$, on effectue une *discrétisation de* $\Omega =]0,1[\times]0,1[$ *de pas* h (*i.e.*, on quadrille Ω à l'aide de « petits » pavés d'aire h^2) et on cherche une fonction étagée u (dépendante de h) telle que u tende vers $\tilde{u}$ lorsque h tend vers 0. On pose $h = \frac{1}{n+1}$ et on écrit $u = \sum_{i,j} u_{i,j}\, \chi_{i,j}$ où, pour $1 \leq i, j \leq n$, $\chi_{i,j}$ est la fonction caractéristique du pavé $P_{i,j} =](i - \frac{1}{2})\, h, (i + \frac{1}{2})\, h[\times](j - \frac{1}{2})\, h, (j + \frac{1}{2})\, h[$. On note alors $x_{i,j} = (i\, h, j\, h)$ les nœuds du quadrillage et $P_{i,j}$ est donc le pavé de centre $x_{i,j}$.



En se basant sur la définition de la dérivée d'une fonction, pour h suffisamment petit, on a les *approximations par différences finies* suivantes :

$$\frac{\partial \tilde{u}}{\partial x} \approx \frac{u(x+\frac{h}{2},y)-u(x-\frac{h}{2},y)}{h}, \quad \frac{\partial^2 \tilde{u}}{\partial x^2} \approx \frac{u(x+h,y)-2\,u(x,y)+u(x-h,y)}{h^2}.$$

De même,

$$\frac{\partial \tilde{u}}{\partial y} \approx \frac{u(x,y+\frac{h}{2})-u(x,y-\frac{h}{2})}{h}, \quad \frac{\partial^2 \tilde{u}}{\partial y^2} \approx \frac{u(x,y+h)-2\,u(x,y)+u(x,y-h)}{h^2}.$$

En décomposant f sous la forme $\sum_{i,j} f_{i,j}\,\chi_{i,j}$, notre problème se ramène alors à chercher les $u_{i,j},\ 1 \leq i,j \leq n$ satisfaisants les équations suivantes :

$$\left\{ \begin{array}{rcl} 4\,u_{i,j}-u_{i-1,j}-u_{i+1,j}-u_{i,j-1}-u_{i,j+1} & = & h^2\,f_{i,j}, \quad 1 \leq i,j \leq n, \\ u_{0,j}=u_{n+1,j}=u_{i,0}=u_{i,n+1}=0, & & 1 \leq i,j \leq n. \end{array} \right.$$

Ce schéma numérique est dit *implicite* par opposition à un schéma *explicite* pour lequel il est possible d'ordonner les inconnues de sorte que chacune d'entre elles puisse être déterminée « explicitement » en fonction des précédentes (système triangulaire). Les schémas numériques implicites ont l'avantage d'être numériquement stables ce qui n'est pas toujours le cas pour

les schémas explicites.

On remarque que ces équations forment un système linéaire que nous allons écrire sous forme matricielle. Pour ceci, on pose

$$M = \begin{pmatrix} 4 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 4 \end{pmatrix} \in \mathbb{M}_{n\times n}(\mathbb{R}),$$

$X_j = (u_{1,j} \quad u_{2,j} \quad \dots \quad u_{n,j})^T$ pour $1 \leq j \leq n$, $X = (X_1^T \quad X_2^T \quad \dots \quad X_n^T)^T$ ainsi que $F_j = (f_{1,j} \quad f_{2,j} \quad \dots \quad f_{n,j})^T$ pour $1 \leq j \leq n$, $F = (F_1^T \quad F_2^T \quad \dots \quad F_n^T)^T$. En définissant de plus $X_0 = X_{n+1} = 0$, le système précédent s'écrit

$$-X_{j-1} + M\,X_j - X_{j+1} = h^2\,F_j, \quad 1 \leq j \leq n,$$

ce qui conduit à

$$A\,X = h^2\,F, \quad A = \begin{pmatrix} M & -\mathbb{I}_n & 0 & \dots & 0 \\ -\mathbb{I}_n & M & -\mathbb{I}_n & \ddots & \vdots \\ 0 & -\mathbb{I}_n & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\mathbb{I}_n \\ 0 & \dots & 0 & -\mathbb{I}_n & M \end{pmatrix} \in \mathbb{M}_{n^2\times n^2}(\mathbb{R}).$$

La matrice A est symétrique réelle et on peut montrer qu'elle est définie positive donc en particulier inversible. En pratique, on a donc à résoudre un système linéaire tridiagonal par blocs de grande taille (notons que faire tendre h vers 0 équivaut à faire tendre n vers l'infini) à résoudre. Notons qu'il existe des méthodes efficaces (*e.g.*, l'algorithme de Thomas qui est une simplification de l'algorithme de Gauss dans le cas particulier des systèmes tridiagonaux) pour résoudre les systèmes linéaires tridiagonaux.

4.2 Notions générales

On rappelle que les méthodes itératives ne s'appliquent que dans le cas de systèmes à coefficients dans $\mathbb{R}$ ou $\mathbb{C}$ mais pas dans le cas des corps finis $\mathbb{F}_p$.

4.2.1 Modèle général d'un schéma itératif

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

$$(S) \; : \; A\,x = b.$$

Le principe général d'une méthode itérative pour résoudre (S) est de générer une suite de vecteurs qui converge vers la solution $A^{-1}\,b$. Pour ce faire l'idée est d'écrire le système (S) sous une forme équivalente permettant de voir la solution comme le point fixe d'une certaine fonction, *e.g.* :

$$(S) \iff B\,x + c = x, \tag{4.1}$$

avec $B \in \mathbb{M}_{n\times n}(\mathbb{K})$ et $c \in \mathbb{K}^n$ bien choisis c'est-à-dire $\mathbb{I} - B$ inversible et $c = (\mathbb{I} - B)\,A^{-1}\,b$. Par exemple, si $A = M - N$ pour deux matrices M, $N \in \mathbb{M}_{n\times n}(\mathbb{K})$ avec M inversible, on peut choisir $B = M^{-1}\,N$ et $c = M^{-1}\,b$. Dans la suite on supposera toujours que $B \in \mathbb{M}_{n\times n}(\mathbb{K})$ et $c \in \mathbb{K}^n$ sont choisis tels que $\mathbb{I} - B$ inversible et $c = (\mathbb{I} - B)\,A^{-1}\,b$ (*e.g.*, méthode itérative *consistante*). On se donne alors un vecteur $x^{(0)} \in \mathbb{K}^n$ et on construit une suite de vecteurs $x^{(k)} \in \mathbb{K}^n$ à l'aide du schéma itératif

$$x^{(k+1)} = B\,x^{(k)} + c, \quad k = 1,\, 2, \ldots. \tag{4.2}$$

Si la suite $(x^{(k)})_{k\in\mathbb{N}}$ est convergente, alors elle converge vers la solution $A^{-1}\,b$ de (S). En effet, si elle existe, la limite x^* est un point fixe de la fonction $x \mapsto B\,x + c$, *i.e.*, vérifie $x^* = B\,x^* + c$ qui est équivalent à $A\,x^* = b$ d'après (4.1).

La mise en oeuvre pratique d'une méthode itérative de la forme (4.2) nécessite la donnée d'un point de départ $x^{(0)}$ (en général, sauf si l'on possède des informations *a priori* sur la solution, on choisit le vecteur nul) et d'une tolérance sur la solution que l'on cherche à calculer. On calcule ensuite les itérés $x^{(k)}$, $k = 1,\, 2, \ldots$ en utilisant la formule (4.2) jusqu'à ce que le résidu $b - A\,x^{(k)}$ soit plus petit que la tolérance.

4.2.2 Convergence

Définition 4.1. *La méthode itérative (4.2) pour résoudre $A\,x = b$ est dite* convergente *si pour toute valeur initiale $x^{(0)} \in \mathbb{K}^n$, on a $\lim_{k\to+\infty} x^{(k)} = A^{-1}\,b$.*

Lemme 4.2. *Si la méthode itérative (4.2) est convergente et si on note $x = A^{-1}\,b$ la solution, alors*

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

Démonstration. On a $c = (\mathbb{I}_n - B)\,A^{-1}\,b = (\mathbb{I}_n - B)\,x$ d'où $x^{(k+1)} = B\,x^{(k)} + (\mathbb{I}_n - B)\,x$ ou encore $x^{(k+1)} - x = B(x^{(k)} - x)$ d'où le résultat. □

Remarquons que $x^{(k)} - x$ représente l'erreur à la k-ième itération de sorte que la formule ci-dessus permet d'estimer cette erreur en fonction de l'erreur initiale.

Le résultat suivant nous donne des critères pour tester la convergence de la méthode itérative (4.2).

Théorème 4.3. *Les assertions suivantes sont équivalentes :*

(i) La méthode itérative (4.2) est convergente ;

(ii) Pour tout $y \in \mathbb{K}^n$, $\lim_{k\to+\infty} B^k\, y = 0$;

(iii) Pour toute norme matricielle $\|.\|$ *sur* $\mathbb{M}_{n\times n}(\mathbb{K})$, *on a* $\lim_{k\to+\infty} \|B^k\| = 0$.

Démonstration. Admis pour ce cours. □

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

Théorème 4.4. *Les assertions suivantes sont équivalentes :*

(i) La méthode itérative (4.2) est convergente ;

(ii) $\rho(B) < 1$, *où* $\rho(B)$ *désigne* le rayon spectral de la matrice B, i.e., *le maximum des modules des valeurs propres de* B ;

(iii) Il existe une norme matricielle $\|.\|$ *sur* $\mathbb{M}_{n\times n}(\mathbb{K})$ *subordonnée à une norme vectorielle sur* $\mathbb{K}^n$ *telle que* $\|B\| < 1$.

Démonstration. Admis pour ce cours. □

4.2.3 Vitesse de convergence

L'égalité $x^{(k)} - x = B^k(x^{(0)} - x)$ donnée précédemment implique que c'est la norme des puissances de la matrice B qui va nous renseigner sur la vitesse de convergence de la méthode itérative. Nous définissons ici les outils permettants de comparer les vitesses de convergence de différentes méthodes itératives.

Définition 4.5. *Considérons le schéma itératif (4.2) convergent. Soit* $\|.\|$ *une norme matricielle sur* $\mathbb{M}_{n\times n}(\mathbb{K})$ *et soit* k *un entier tel que* $\|B^k\| < 1$ *(l'existence d'un tel* k *découle du théorème 4.3). On appelle* taux moyen de convergence associé à la norme $\|.\|$ pour k itérations de (4.2) *le nombre positif*

$$R_k(B) = -\ln\left(\left[\|B^k\|\right]^{\frac{1}{k}}\right).$$

Définition 4.6. *Considérons deux méthodes itératives consistantes et convergentes :*

$$(1)\quad x^{(k+1)} = B_1\, x^{(k)} + c_1, \quad k = 1, 2, \ldots,$$

$$(2)\quad x^{(k+1)} = B_2\, x^{(k)} + c_2, \quad k = 1, 2, \ldots.$$

Soit k *un entier tel que* $\|B_1^k\| < 1$ *et* $\|B_2^k\| < 1$. *On dit que* (1) est plus rapide que (2) relativement à la norme $\|.\|$ *si* $R_k(B_1) \geq R_k(B_2)$.

En pratique le calcul des $R_k(B)$ est trop couteux car il nécessite l'évaluation des B^k. On préfère donc estimer le taux asymptotique de convergence.

Définition 4.7. *On appelle* taux asymptotique de convergence *le nombre*

$$R_\infty(B) = \lim_{k\to+\infty} R_k(B) = -\ln(\rho(B)).$$

Théorème 4.8. *Avec les notations précédentes, une méthode itérative est d'autant plus rapide que son taux asymptotique de convergence est grand c'est-à-dire que* $\rho(B)$ *est petit.*

4.3 Les méthodes itératives classiques

4.3.1 Principe

On considère un système linéaire (S) : $A\,x = b$ avec A inversible. L'idée est de déduire un schéma itératif de la décomposition de A sous la forme $A = M - N$ où M est une matrice inversible. En pratique on suppose que les systèmes de matrice M sont « faciles » à résoudre (par exemple M diagonale, triangulaire, ...). Le système (S) s'écrit alors $M\,x = N\,x + b$ c'est-à-dire $x = B\,x + c$ avec $B = M^{-1}\,N$ et $c = M^{-1}\,b$ et on considère le schéma itératif associé :

$$x^{(0)} \in \mathbb{K}^n, \quad M\,x^{(k+1)} = N\,x^{(k)} + b.$$

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

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

Exemple : Considérons la matrice

$$A = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}. \tag{4.3}$$

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

$$\underbrace{\begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}}_{D} - \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}}_{E} - \underbrace{\begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}}_{F}.$$

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

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

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

4.3.2 Méthode de Jacobi

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

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

Définition 4.9. *La matrice* $B_J = D^{-1}\,(E+F)$ *s'appelle* la matrice de Jacobi associée à A.

Mise en œuvre et complexité arithmétique : On se propose d'estimer le nombre d'opérations à virgule flottante nécessaires pour calculer $x^{(k+1)}$ à partir de $x^{(k)}$. On a $D\,x^{(k+1)} = (E+F)\,x^{(k)} + b$ donc pour tout $i = 1, \ldots, n$, $\left(D\,x^{(k+1)}\right)_i = \left((E+F)\,x^{(k)}\right)_i + b_i$ c'est-à-dire

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

Pour calculer $x_i^{(k+1)}$ à partir de $x^{(k)}$, on a donc besoin de $n-1$ multiplications, $n-1$ additions et 1 division soit $2\,n-1$ opérations à virgule flottante. Par conséquent il nous faudra $n\,(2\,n-1)$ opérations à virgule flottante pour calculer $x^{(k+1)}$ à partir de $x^{(k)}$ et pour K itérations, on aura besoin de $K\,n\,(2\,n-1)$ opérations à virgule flottante. Pour comparaison, pour $n = 1000$, l'élimination de Gauss coûte environ $\frac{2}{3}\,n^3 = 6,6.10^8$ opérations à virgule flottante alors que par exemple $K = 100$ itérations de la méthode de Jacobi coûtent approximativement $2\,K\,n^2 = 2.10^8$ opérations à virgule flottante.

Convergence : D'après le théorème 4.4, on a le résultat suivant :

Théorème 4.10. *La méthode de Jacobi converge si et seulement si* $\rho(B_J) < 1$.

Exemple : Pour la matrice A donnée par (4.3), on obtient :

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

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

4.3.3 Méthode de Gauss-Seidel

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

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

Définition 4.11. *La matrice* $B_{GS} = (D-E)^{-1}\,F$ *s'appelle* la matrice de Gauss-Seidel associée à A.

Mise en œuvre et complexité arithmétique : On se propose d'estimer le nombre d'opérations à virgule flottante nécessaires pour calculer $x^{(k+1)}$ à partir de $x^{(k)}$. On a $(D-E)\,x^{(k+1)} = F\,x^{(k)} + b$ donc pour tout $i = 1, \ldots, n$, $\left((D-E)\,x^{(k+1)}\right)_i = \left(F\,x^{(k)}\right)_i + b_i$ c'est-à-dire

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

ce qui entraîne

$$x_1^{(k+1)} = \frac{1}{a_{1,1}}\left[-\sum_{j=2}^{n} a_{1,j}\,x_j^{(k)} + b_1\right],$$

et pour $i = 2, \ldots, n$,

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

La complexité arithmétique de la méthode de Gauss-Seidel est la même que celle de la méthode de Jacobi. Cependant, on peut remarquer que la méthode de Gauss-Seidel est plus intéressante en ce qui concerne la gestion de la mémoire. En effet, on peut écraser au fur et à mesure la valeur de $x_i^{(k)}$ et ne stocker au cours des calculs qu'un seul vecteur de taille n, *e.g.*, le vecteur $(x_1^{(k+1)} \ldots x_i^{(k+1)} x_{i+1}^{(k)} \ldots x_n^{(k)})^T$, au lieu de deux vecteurs pour la méthode de Jacobi.

Convergence : D'après le théorème 4.4, on a le résultat suivant :

Théorème 4.12. *La méthode de Gauss-Seidel converge si et seulement si* $\rho(B_{GS}) < 1$.

Exemple : Pour la matrice A donnée par (4.3), on obtient :

$$B_{GS} = (D-E)^{-1}\,F = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & -1 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix},$$

d'où

$$B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}.$$

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

4.3.4 Méthode de relaxation

On considère un système linéaire (S) : $A\,x = b$ avec A inversible. Soit ω un paramètre réel non nul. On pose $A = M - N$ avec $M = \frac{1}{\omega}(D - \omega\,E)$ inversible et $N = \left(\frac{1-\omega}{\omega}\right) D + F$. Le schéma itératif s'écrit alors

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

qui est équivalent à :

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

Définition 4.13. *La matrice* $B_R(\omega) = (D - \omega\,E)^{-1}\left[(1-\omega)\,D + \omega\,F\right]$ *s'appelle* la matrice de relaxation associée à A *et* ω *est* le facteur de relaxation. *Si* $\omega < 1$, *on parle de* sous-relaxation, *si* $\omega = 1$, *on retrouve la méthode de Gauss-Seidel et si* $\omega > 1$, *on parle de* sur-relaxation.

D'après le théorème 4.4, on a le résultat suivant :

Théorème 4.14. *La méthode de relaxation converge si et seulement si* $\rho(B_R(\omega)) < 1$.

Exemple : Pour la matrice A donnée par (4.3), on obtient :

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

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

4.3.5 Résultats de convergence dans des cas particuliers

On s'intéresse tout d'abord au cas des matrices symétriques définies positives.

Théorème 4.15. *Soit* A *une matrice symétrique définie positive et écrivons* $A = M - N$ *avec* M *inversible et* $M^T + N$ *définie positive. Alors la méthode itérative*

$$x^{(0)} \in \mathbb{K}^n, \quad x^{(k+1)} = M^{-1}\,N\,x^{(k)} + M^{-1}\,b,$$

converge.

Démonstration. Admis pour ce cours. □

Corollaire 4.16. *Soit A une matrice symétrique définie positive. Alors la méthode de Gauss-Seidel converge.*

Démonstration. Pour la méthode de Gauss-Seidel, on a $M = D - E$ et $N = F$. La matrice M est inversible car A est supposée définie positive et est donc inversible (voir la section 2.3). On a de plus $M^T + N = D - E^T + F$. Or A étant supposée symétrique, on a $E^T = F$ d'où $M^T + N = D$. La matrice $M^T + N$ est donc définie positive car, pour tout $i = 1, \ldots, n$, $\langle D\, e_i, e_i \rangle = a_{i,i}$ et $a_{i,i} > 0$ puisque A est définie positive (voir la section 2.3). Le théorème 4.15 précédent permet alors de conclure. □

Considérons maintenant le cas des matrices *à diagonale strictement dominante.*

Définition 4.17. *Une matrice $A = (a_{i,j})_{1 \leq i,j \leq n}$ est dite* à diagonale strictement dominante *si :*

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

Par exemple, la matrice du système linéaire obtenu à la fin de la section 4.1 pour la résolution d'un système d'équations aux dérivées partielles est à diagonale strictement dominante.

Théorème 4.18. *Soit A une matrice à diagonale strictement dominante. Alors A est inversible et les méthodes de Jacobi et de Gauss-Seidel convergent toutes les deux.*

Démonstration. Admis pour ce cours. □

4.4 Méthode du gradient conjugué

Il s'agit d'une méthode itérative qui permet de résoudre un système linéaire $(S) : A\,x = b$ lorsque A est une matrice symétrique et définie positive. Dans cette méthode, la matrice A du système intervient une seule fois à chaque itération, lorsqu'on calcule son produit par un vecteur. Par conséquent, la méthode du gradient conjugué est particulièrement bien adaptée aux systèmes creux et de grande taille. Pour ce type de systèmes, la méthode du gradient conjugué est souvent plus efficace que les méthodes déjà présentées, à la fois en termes de complexité arithmétique et en termes d'espace mémoire nécessaire : elle est donc très utilisée en pratique.

Soit $(S) : A\,x = b$ avec $A \in \mathbb{M}_{n \times n}(\mathbb{R})$ symétrique et définie positive, et $b \in \mathbb{R}^n$. La méthode du gradient conjugué construit une suite de vecteurs $(x^{(k)})_{k=0,1,\ldots}$ telle que $x^{(m)} = A^{-1}\,b$ pour un indice $m \leq n$. En principe, il s'agit donc d'une méthode exacte. En pratique, à cause des erreurs numériques, le gradient conjugué est considéré comme une méthode itérative. Dans les applications, le nombre d'itérations nécéssaires pour atteindre la précision voulue

est significativement plus petit que la taille n du système, en particulier dans le cas où on utilise des techniques de préconditionnement.

Dans la suite, le gradient conjugué sera présenté comme un cas particulier d'une famille de méthodes itératives connue sous le nom de *méthodes du gradient*.

Définition 4.19. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ symétrique et définie positive. On définit la fonction $\|\cdot\|_A : \mathbb{R}^n \longrightarrow \mathbb{R}_+$ par $\|x\|_A = \sqrt{x^T\, A\, x}$.*

Proposition 4.20. *La fonction $\|\cdot\|_A$ est une norme vectorielle.*

Démonstration. En utilisant le fait que A est symétrique et définie positive, on montre que la fonction $\|\cdot\|_A$ vérifie les trois propriétés de la définition 3.1, avec $E = \mathbb{R}^n$ et $\mathbb{K} = \mathbb{R}$:

- $\forall x \in \mathbb{R}^n, \|x\|_A = 0 \Rightarrow x^T\, A\, x = 0 \Rightarrow x = 0$, car A est définie positive ;
- $\forall \lambda \in \mathbb{R}, \forall x \in \mathbb{R}^n, \|\lambda x\|_A = \sqrt{(\lambda x)^T\, A\, (\lambda x)} = \sqrt{\lambda^2\, x^T\, A\, x} = |\lambda|\sqrt{x^T\, A\, x} = |\lambda|\|x\|_A$;
- $\forall x, y \in \mathbb{R}^n, \|x+y\|_A = \sqrt{(x+y)^T\, A\, (x+y)} = \sqrt{x^T\, A\, x + y^T\, A\, y} \leq \sqrt{x^T\, A\, x} + \sqrt{x^T\, A\, x} = \|x\|_A + \|y\|_A$.

□

Définition 4.21. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ symétrique et définie positive. On dit que les vecteurs u et v de $\mathbb{R}^n$ sont* A-conjugués *si $u^T A v = 0$.*

On remarque que l'application $(u, v) \in (\mathbb{R}^n)^2 \longmapsto u^T A v \in \mathbb{R}$ est un produit scalaire sur $\mathbb{R}^n$. Deux vecteurs u et v de $\mathbb{R}^n$ sont donc A-conjugués s'ils sont orthogonaux (pour ce produit scalaire).

4.4.1 Méthodes du gradient

On considère le problème suivant : minimiser sur $\mathbb{R}^n$ la fonction ϕ définie par

$$\phi(x) = \frac{1}{2}\, x^T\, A\, x - b^T\, x,$$

où la matrice A est symétrique et définie positive. L'intérêt de la question vient du fait que le minimum de ϕ est atteint pour $x^* = A^{-1}\, b$, et cette solution est unique. En effet, le gradient de ϕ en $x = (x_1\ \ldots\ x_n)^T \in \mathbb{R}^n$ est le vecteur

$$\nabla\phi(x) = \left(\frac{\partial\phi}{\partial x_1}\ \frac{\partial\phi}{\partial x_2}\ \ldots\ \frac{\partial\phi}{\partial x_n}\right)^T = \frac{1}{2}\, A\, x + \frac{1}{2}\, A^T\, x - b = A\, x - b,$$

qui s'annule seulement pour $x^* = A^{-1}\, b$. Notons que dans ce calcul on a utilisé le fait que A est une matrice symétrique. Le vecteur $x^* = A^{-1}\, b$ est donc l'unique point critique de ϕ et, puisque A est définie positive, il s'agit d'un minimum global (admis).

On peut donc conclure de ce qui précède que minimiser ϕ sur $\mathbb{R}^n$ et résoudre le système linéaire $A\,x = b$ sont deux problèmes équivalents.

Définition 4.22. *Soit* (S) : $A\,x = b$ *avec* $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ *symétrique et définie positive, et* $b \in \mathbb{R}^n$. *La quantité* $r(x)$ *définie par*

$$r(x) = b - A\,x = -\nabla\phi(x),$$

est appelée résidu du système (S) en x. *En particulier, on notera*

$$r^{(k)} = b - A\,x^{(k)} = -\nabla\phi(x^{(k)}),$$

le résidu à l'itération k.

Remarque : on rappelle que la valeur $\nabla\phi(x)$ du gradient de ϕ en x donne la direction de plus forte pente pour la fonction ϕ au point x.

Les méthodes du gradient procèdent généralement en choisissant à l'étape k une *direction de descente* pour ϕ c'est-à-dire un vecteur $p^{(k)} \in \mathbb{R}^n$ tel que $p^{(k)T}\nabla\phi(x^{(k)}) < 0$. Pour calculer $x^{(k+1)}$ à partir de $x^{(k)}$, on minimise alors la fonction ϕ sur la droite de vecteur directeur $p^{(k)}$ et passant par $x^{(k)}$: on choisit donc

$$x^{(k+1)} = x^{(k)} + \alpha_k\,p^{(k)}, \tag{4.4}$$

où $\alpha_k \in \mathbb{R}$ est tel que

$$\phi(x^{(k+1)}) = \min_{\alpha\in\mathbb{R}} \phi(x^{(k)} + \alpha\,p^{(k)}).$$

En dérivant $\phi(x^{(k)} + \alpha\,p^{(k)})$ par rapport à α, on obtient

$$\begin{aligned}
\frac{\partial}{\partial\alpha}\left(\phi(x^{(k)} + \alpha\,p^{(k)})\right) &= \frac{\partial}{\partial\alpha}\left(\frac{1}{2}\,(x^{(k)} + \alpha\,p^{(k)})^T\,A\,(x^{(k)} + \alpha\,p^{(k)}) - (x^{(k)} + \alpha\,p^{(k)})^T\,b\right),\\
&= \frac{\partial}{\partial\alpha}\left(\frac{1}{2}\,(x^{(k)T}\,A\,x^{(k)} + \alpha\,p^{(k)T}\,A\,x^{(k)} + \alpha\,x^{(k)T}\,A\,p^{(k)} + \alpha^2\,p^{(k)T}\,A\,p^{(k)}) - x^{(k)T}\,b - \alpha\,p^{(k)T}\,b\right),\\
&= \frac{1}{2}\,\left(p^{(k)T}\,A\,x^{(k)} + x^{(k)T}\,A\,p^{(k)} + 2\,\alpha\,p^{(k)T}\,A\,p^{(k)}\right) - p^{(k)T}\,b,\\
&= (x^{(k)} + \alpha\,p^{(k)})^T\,A\,p^{(k)} - b^T\,p^{(k)},
\end{aligned}$$

car A étant symétrique, $p^{(k)T}\,A\,x^{(k)} = x^{(k)T}\,A\,p^{(k)}$. En imposant $\frac{\partial}{\partial\alpha}\left(\phi(x^{(k)} + \alpha\,p^{(k)})\right) = 0$, on trouve alors

$$\alpha_k = \frac{(b - A\,x^{(k)})^T\,p^{(k)}}{p^{(k)T}\,A\,p^{(k)}} = \frac{r^{(k)T}\,p^{(k)}}{p^{(k)T}\,A\,p^{(k)}}. \tag{4.5}$$

En particulier, on observe que $\alpha_k > 0$ puisque $r^{(k)T}\,p^{(k)} = -p^{(k)T}\,\nabla\phi(x^{(k)}) > 0$.

Proposition 4.23. *À chaque itération, le résidu* $r^{(k+1)}$ *est orthogonal à la direction de descente* $p^{(k)}$ *utilisée à l'étape précédente,* i.e., $r^{(k+1)T}\,p^{(k)} = 0$.

Démonstration. D'après (4.4), pour $k \in \mathbb{N}$, on a $b - A\,x^{(k+1)} = b - A\,x^{(k)} - \alpha_k\,A\,p^{(k)}$ et donc $r^{(k+1)} = r^{(k)} - \alpha_k\,A\,p^{(k)}$. D'où , à partir de (4.5), $r^{(k+1)T}\,p^{(k)} = (r^{(k)} - \alpha_k\,A\,p^{(k)})^T\,p^{(k)} = r^{(k)T}\,p^{(k)} - \alpha_k\,p^{(k)T}\,A\,p^{(k)} = 0$. □

4.4.2 Méthode de la plus forte pente

Dans cette méthode, on choisit à chaque itération la direction de descente

$$p^{(k)} = r^{(k)} = -\nabla\phi(x^{(k)}),$$

c'est-à-dire la direction de plus forte pente pour ϕ au point $x^{(k)}$. La proposition 4.23 implique alors qu'à chaque itération, on choisit une direction de descente orthogonale à la précédente, *i.e.*, $p^{(k+1)T}\, p^{(k)} = 0$.

Théorème 4.24. *Pour la méthode de la plus forte pente on a, à l'itération k :*

$$\|x^* - x^{(k)}\|_A \leq \left(\frac{\mathrm{Cond}_2(A) - 1}{\mathrm{Cond}_2(A) + 1}\right)^k \|x^* - x^{(0)}\|_A.$$

Démonstration. Admis pour ce cours. □

Le choix $p^{(k)} = -\nabla\phi(x^{(k)})$ peut paraitre intuitivement assez efficace pour minimiser ϕ vu qu'on se déplace le long de la direction de plus forte décroissance de la fonction. Pourtant, le théorème 4.24 suggère que dans certains cas la convergence de cette méthode pourrait être lente, notamment si la matrice A est mal conditionnée, *i.e.*, lorsque $\mathrm{Cond}_2(A) \gg 1$ (Voir Chapitre 3). Soient $\lambda_1 \geq \cdots \geq \lambda_n > 0$ les valeurs propres de A : géométriquement, le fait que le nombre de conditionnement $\mathrm{Cond}_2(A) = \lambda_1/\lambda_n$ (voir Proposition 3.6) soit grand est équivalent au fait que les courbes de niveau de ϕ soient des hyperellipsoïdes très allongés.

Exemple : Soit A une matrice symétrique de taille 2×2 ayant pour valeurs propres $\lambda_1 = e^4 \geq \lambda_2 = e^{-2} > 0$, et b le vecteur $(1 \quad 1)^T$. Le conditionnement de A est alors $\mathrm{Cond}_2(A) \approx 403,43$ de sorte que A est mal conditionnée. La figure 4.1 montre la surface définie dans $\mathbb{R}^3$ par la fonction ϕ : on voit que cette surface ressemble à une vallée au fond assez plat.

La solution du système est approximativement $x^* \approx (8,7361 \quad 2,3478)^T$. On choisit un vecteur initial $x^{(0)} = (8 \quad 2)^T$ assez proche de x^* et on effectue 10 itérations de la méthode de la plus forte pente. La figure 4.2 montre, en échelle logarithmique, les résidus rélatifs $\frac{\|r^{(k)}\|_2}{\|b\|_2}$ obtenus à chaque itération : on constate que 10 itérations ne suffisent pas à atteindre un résidu de norme inférieure à 10^{-6}. Géométriquement, ceci peut s'expliquer par le fait qu'à chaque itération de la méthode de la plus forte pente on choisit une direction orthogonale à la précédente et donc on va rebondir sur les parois de la vallée, ce qui fait qu'on s'approche très lentement du minimum situé sur le fond.

Par contre, pour ce même exemple, la méthode du gradient conjugué converge en 2 itérations avec un résidu rélatif comparable à la précision machine (voir la section suivante).

4.4.3 Gradient conjugué

Pour la méthode du gradient conjugué, le choix de $p^{(k)}$ dans le schéma itératif (4.4) se fait en tenant compte des directions $p^{(j)}$, $j = 0, 1, \ldots, k-1$, calculées aux itérations précédentes.

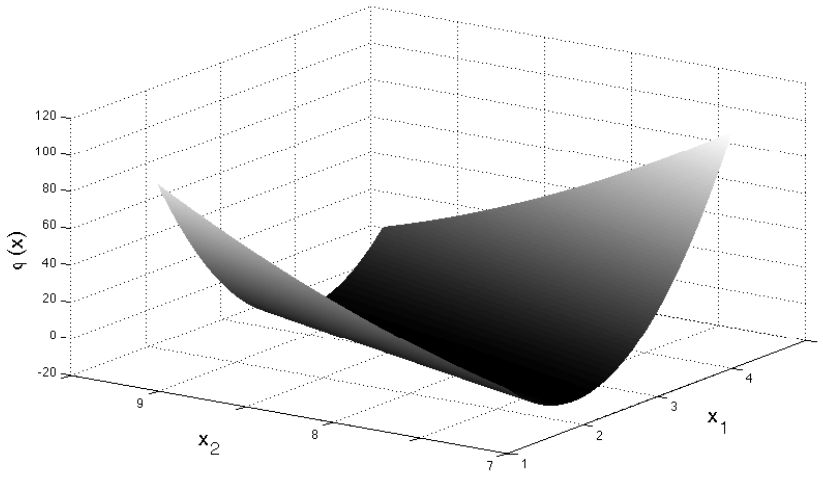


FIGURE 4.1 : Surface définie par la fonction ϕ.

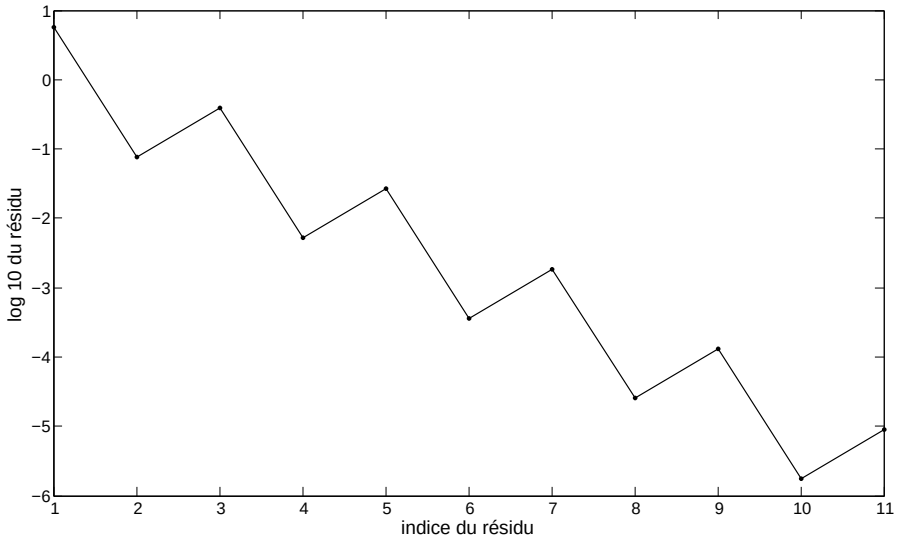


FIGURE 4.2 : Résidus rélatifs calculés à chaque itération de la méthode de la plus forte pente.

On définit

$$p^{(k)} = \begin{cases} r^{(0)} & \text{si } k = 0, \\ r^{(k)} + \beta_k \, p^{(k-1)} & \text{si } k \geq 1, \end{cases} \tag{4.6}$$

où $\beta_k \in \mathbb{R}$ est tel que

$$p^{(k)T} \, A \, p^{(k-1)} = 0. \tag{4.7}$$

La condition (4.7) revient à dire que les directions de descente $p^{(k-1)}$ et $p^{(k)}$ calculées à deux itérations consécutives de la méthode du gradient conjugué sont A-conjuguées.

En utilisant (4.6) et (4.7), on peut écrire β_k sous la forme

$$\beta_k = -\frac{r^{(k)T} \, A \, p^{(k-1)}}{p^{(k-1)T} \, A \, p^{(k-1)}}. \tag{4.8}$$

On vérifie alors que $p^{(k)}$ est effectivement une direction de descente pour ϕ : si $r^{(k)} \neq 0$, c'est-à-dire $x^{(k)} \neq x^*$, on a

$$p^{(k)T} \, \nabla\phi(x^{(k)}) = -p^{(k)T} \, r^{(k)} = -r^{(k)T} \, r^{(k)} - \beta_k \, p^{(k-1)T} \, r^{(k)} = -r^{(k)T} \, r^{(k)} < 0.$$

En particulier, on constate que $p^{(k)T} r^{(k)} = r^{(k)T} r^{(k)}$ donc l'expression (4.5) pour α_k devient

$$\alpha_k = \frac{r^{(k)T} \, r^{(k)}}{p^{(k)T} \, A \, p^{(k)}}.$$

Lemme 4.25. *À chaque itération, le résidu $r^{(k)}$ est orthogonal au résidu $r^{(k-1)}$ calculé à l'itération précédente,* i.e., $r^{(k)T} \, r^{(k-1)} = 0$.

Démonstration. On a à la fois $r^{(k)T} \, r^{(k-1)} = r^{(k)T} \, p^{(k-1)} - \beta_{k-1} \, r^{(k)T} \, p^{(k-2)} = -\beta_{k-1} \, r^{(k)T} \, p^{(k-2)}$ et $r^{(k)T} \, p^{(k-2)} = r^{(k-1)T} \, p^{(k-2)} - \alpha_k \, p^{(k-1)T} \, A \, p^{(k-2)} = 0$, d'où $r^{(k)T} \, r^{(k-1)} = 0$. □

Lemme 4.26. *La quantité β_k peut s'écrire sous la forme*

$$\beta_k = \frac{r^{(k)T} r^{(k)}}{r^{(k-1)T} r^{(k-1)}}.$$

Démonstration. Pour démontrer cette formule, on écrit la quantité $p^{(k)T} \, r^{(k-1)}$ de deux manières différentes. On a

$$p^{(k)T} \, r^{(k-1)} = r^{(k)T} \, r^{(k-1)} + \beta_k \, p^{(k-1)T} \, r^{(k-1)} = \beta_k \, p^{(k-1)T} \, r^{(k-1)} = \beta_k \, r^{(k-1)T} \, r^{(k-1)},$$

et

$$p^{(k)T} \, r^{(k-1)} = p^{(k)T} \, r^{(k)} + \alpha_{k-1} \, p^{(k)T} \, A \, p^{(k-1)} = p^{(k)T} \, r^{(k)} = r^{(k)T} \, r^{(k)},$$

d'où le résultat. □

Théorème 4.27. *Soit $\mathcal{S}_k$ le sous-espace vectoriel de $\mathbb{R}^n$ engendré par les vecteurs $p^{(0)}, \ldots, p^{(k-1)}$. Alors le vecteur $x^{(k)}$ défini par la méthode du gradient conjugué à l'itération k minimise la fonction ϕ sur $\mathcal{S}_k$:*

$$\phi(x^{(k)}) = \min_{x \in \mathcal{S}_k} \phi(x), \qquad k \geq 1.$$

Démonstration. Admis pour ce cours. □

Théorème 4.28. *Soit $r^{(0)} \neq 0$ et $h \geq 1$ tels que $r^{(k)} \neq 0$ pour tout $k \leq h$. Alors pour $k, j \in \{0, \ldots, h\}$ avec $k \neq j$, on a :*

$$r^{(k)T} r^{(j)} = 0 \qquad \text{et} \qquad p^{(k)T} A p^{(j)} = 0.$$

Autrement dit, dans la méthode du gradient conjugué, les résidus forment un ensemble de vecteurs orthogonaux et les directions de descente $p^{(k)}$ forment un ensemble de vecteurs A-conjugués.

Démonstration. Par récurrence sur h (exercice). □

Corollaire 4.29. *Il existe $m \leq n$ tel que $r^{(m)} = 0$. Autrement dit, le gradient conjugué calcule la solution $x^* = A^{-1}b$ en au plus n itérations.*

Pour récapituler, l'*algorithme du gradient conjugué* se déroule de la manière suivante :

Entrèe : $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ symétrique et définie positive, $b \in \mathbb{R}^n$, et $x^{(0)} \in \mathbb{R}^n$.
Sortie : $x^* \in \mathbb{R}^n$ tel que $A x^* = b$.

1. $k = 0$;
2. $r^{(0)} = b - A x^{(0)}$;
3. Tant que $r^{(k)} \neq 0$, faire :
   - Si $k = 0$, alors faire :
     $p^{(0)} = r^{(0)}$;
     Sinon faire :
     $\beta_k = r^{(k)T} r^{(k)} / r^{(k-1)T} r^{(k-1)}$;
     $p^{(k)} = r^{(k)} + \beta_k p^{(k-1)}$;
   - $\alpha_k = r^{(k)T} r^{(k)} / p^{(k)T} A p^{(k)}$;
   - $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$;
   - $r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$;
   - $k = k + 1$;
4. Retourner $x^* = x^{(k)}$.

Les propriétés de convergence de la méthode du gradient conjugué sont données par le résultat suivant (à comparer au théorème 4.24) :

Théorème 4.30. *Pour la méthode du gradient conjugué, on a, à l'itération k :*

$$\|x^* - x^{(k)}\|_A \leq 2 \left(\frac{\sqrt{\mathrm{Cond}_2(A)} - 1}{\sqrt{\mathrm{Cond}_2(A)} + 1} \right)^k \|x^* - x^{(0)}\|_A$$

Démonstration. Admis pour ce cours. □

Quelques remarques :

- Dans la pratique, le critère d'arrêt $r^{(k)} = 0$ est remplacé par $\|r^{(k)}\|_2 < \epsilon_M \|b\|_2$, et k est borné par $k_{\max} \ll n$. En effet, la méthode du gradient conjugué est souvent appliquée à des systèmes de grande taille, où on espère atteindre une bonne approximation de la solution x^* après un nombre d'itérations significativement plus petit que n.

- En principe, le vecteur initial $x^{(0)}$ peut être choisi de manière arbitraire, par exemple comme le vecteur nul. Évidemment, le choix de $x^{(0)}$ a un effet sur le nombre d'itérations nécessaires pour atteindre la solution.

- À chaque itération, l'opération la plus couteuse du point de vue de la complexité arithmétique est la multiplication matrice-vecteur $A\,p^{(k)}$, qui nécessite en général n^2 opérations. On peut donc estimer qu'asymptotiquement le cout de la méthode du gradient conjugué est de l'ordre de $k_{\max}\,n^2$ opérations. De plus, si la matrice A est creuse, comme c'est souvent le cas dans les applications (*e.g.*, résolution d'équations aux dérivées partielles par discrétisation- voir Section 4.1), le produit matrice-vecteur $A\,p^{(k)}$ peut se faire en seulement n opérations, ce qui rend la méthode du gradient conjugué plus avantageuse qu'une méthode directe comme celle de Cholesky (voir la proposition 2.18).

- Si on veut résoudre un système linéaire $A\,x = b$ où la matrice A est inversible mais n'est pas symétrique et définie positive, on peut toujours appliquer la méthode du gradient conjugué au système équivalent $A^T A\,x = A^T b$ (*méthode des équations normales*). Cependant, cette approche n'est pas recommandée lorsque A est mal conditionnée car le passage aux équations normales élève le conditionnement de la matrice au carré. Dans ce cas, il existe des versions de la méthode du gradient conjugué spécialement adaptées aux matrices non symétriques : l'une des plus utilisées est la méthode GMRES.

4.4.4 Gradient conjugué avec préconditionnement

Le préconditionnement est une technique qui vise à accélérer la convergence d'une méthode itérative. On rappelle que, dans le cas de la méthode du gradient conjugué, la convergence est très rapide si la matrice A est proche de la matrice identité, ou si ses valeurs propres sont bien regroupées (voir Théorème 4.30).

Description : On considère un système linéaire $(S) : A\,x = b$ avec A symétrique et définie positive. Étant donnée une matrice $C \in \mathbb{M}_{n\times n}(\mathbb{R})$ inversible, on définit le système transformé

$$(\tilde{S}) : \tilde{A}\,\tilde{x} = \tilde{b}, \qquad \text{avec} \qquad \tilde{A} = C^{-1}A(C^{-1})^T, \quad \tilde{x} = C^T x, \quad \text{et} \quad \tilde{b} = C^{-1}b.$$

On remarque que $\tilde{A}$ est aussi symétrique et définie positive donc on peut appliquer la méthode du gradient conjugué à $(\tilde{S})$.

Définition 4.31. *La matrice $M = C\,C^T$ est appelée* préconditionneur *du système* $(\tilde{S})$.

Le choix du préconditionneur M est généralement fait de sorte que :

- la matrice $\tilde{A}$ soit mieux conditionnée que A, ou idéalement proche de la matrice identité, pour que la méthode du gradient conjugué appliquée à $(\tilde{S})$ converge rapidement,
- M soit « facilement » inversible car dans l'algorithme il faut résoudre un système linéaire de matrice M à chaque itération (voir ci-dessous), donc cette opération doit pouvoir être effectuée de manière stable et rapide, idéalement en un nombre d'opérations de l'ordre de n.

Pour écrire l'*algorithme du gradient conjugué préconditionné*, on observe que, si on note $s^{(k)}$ le résidu de la méthode préconditionnée à l'itération k, on a, avec les notations précédentes, $s^{(k)} = C^{-1}\,r^{(k)}$, et donc $s^{(k)T}\,s^{(k)} = r^{(k)T}\,M^{-1}\,r^{(k)}$. Si $z^{(k)}$ est tel que $M\,z^{(k)} = r^{(k)}$, alors $s^{(k)T}\,s^{(k)} = z^{(k)T}\,r^{(k)}$. On obtient donc l'algorithme suivant :

Entrèe : $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ symétrique et définie positive, $b \in \mathbb{R}^n$, $x^{(0)} \in \mathbb{R}^n$, et un préconditionneur $M \in \mathbb{M}_{n\times n}(\mathbb{R})$ symétrique et défini positif.
Sortie : $x^* \in \mathbb{R}^n$ tel que $A\,x^* = b$.

1. $k = 0$;
2. $r^{(0)} = b - A\,x^{(0)}$;
3. Tant que $r^{(k)} \neq 0$, faire :
   - résoudre le système linéaire $M\,z^{(k)} = r^{(k)}$;
   - Si $k = 0$, alors faire :
     $p^{(0)} = z^{(0)}$;

     Sinon faire :
     $\beta_k = z^{(k)T}\,r^{(k)} / z^{(k-1)T}\,r^{(k-1)}$;
     $p^{(k)} = z^{(k)} + \beta_k\,p^{(k-1)}$;
   - $\alpha_k = z^{(k)T}\,r^{(k)} / p^{(k)T}\,A\,p^{(k)}$;
   - $x^{(k+1)} = x^{(k)} + \alpha_k\,p^{(k)}$;
   - $r^{(k+1)} = r^{(k)} - \alpha_k\,A\,p^{(k)}$;
   - $k = k + 1$;
4. Retourner $x^* = x^{(k)}$.

Remarque : la matrice C est utilisée dans la description théorique de la méthode mais n'apparait pas explicitement dans l'algorithme.

Le choix d'un préconditionneur est un problème délicat et il existe une vaste littérature à ce sujet. Ici nous présentons deux exemples simples qui peuvent s'appliquer de manière assez générale.

Exemple 1 : Préconditionnement diagonal. On note $A = (a_{i,j})_{1\leq i,j\leq n}$, et on choisit le conditionneur $M = (m_{i,j})_{1\leq i,j\leq n}$ défini par

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

Exemple 2 : Préconditionnement de *Cholesky incomplet*. On choisit $M = L\,L^T$, où $L = (\ell_{i,j})_{1\leq i,j\leq n}$ est une matrice triangulaire inférieure définie de la manière suivante :

$$\ell_{i,i} = \sqrt{a_{i,i} - \sum_{r=1}^{i-1} \ell_{i,r}^2}, \qquad i = 1, \ldots, n,$$

$$\ell_{i,j} = \begin{cases} 0 & \text{si } a_{i,j} = 0, \\ \frac{1}{\ell_{i,j}} \left(a_{i,j} - \sum_{r=1}^{j-1} \ell_{i,r}\ell_{j,r} \right) & \text{si } a_{i,j} \neq 0, \end{cases} \qquad j = 1, \ldots, i-1, \quad i = 2, \ldots, n.$$

On remarque que la matrice L est définie de manière à préserver l'éventuelle structure creuse de A. Les éléments non nuls de L sont calculés comme pour le facteur de Cholesky de A (voir l'algorithme de Cholesky dans la section 2.3).

Chapitre 5

Interpolation polynomiale

5.1 Le problème considéré

Dans ce chapitre, on notera $\mathcal{P}_n = \mathbb{R}_n[x]$ l'ensemble des polynômes de degré inférieur ou égal à n et à coefficients dans $\mathbb{R}$. On rappelle que $\mathcal{P}_n$ est un espace vectoriel de dimension $n+1$ sur $\mathbb{R}$. Soit $(a,b) \in \mathbb{R}^2$ avec $a < b$ et $f : [a,b] \to \mathbb{R}$ une fonction que l'on suppose continue sur $[a,b]$. On considère $n+1$ points $x_0, \ldots, x_n$ de l'intervalle $[a,b]$ tels que $a \leq x_0 \leq x_1 \leq \cdots \leq x_n \leq b$. Le *problème d'interpolation polynomiale* $(I)^f_{m,n}$ consiste à chercher s'il existe un polynôme $P_m \in \mathcal{P}_m$ qui coïncide avec f aux nœuds $(x_i)_{0\leq i\leq n}$, *i.e.*, tel que, pour tout $i = 0, \ldots, n$, $P_m(x_i) = f(x_i)$. Si on pose $P_m(x) = \lambda_0 + \lambda_1\, x + \cdots + \lambda_m\, x^m$ avec les λ_i dans $\mathbb{R}$, alors le problème se ramène à trouver $\lambda_0, \ldots, \lambda_m$ tels que :

$$(S) : \underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^m \\ 1 & x_1 & x_1^2 & \ldots & x_1^m \\ \vdots & & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{pmatrix}}_{V} \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_m \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_m) \end{pmatrix}. \tag{5.1}$$

Nous avons donc un système de $n+1$ équations linéaires en $m+1$ inconnues.

Proposition 5.1. *Le problème d'interpolation $(I)^f_{m,n}$ admet une unique solution si et seulement si $m = n$ et les nœuds $(x_i)_{0\leq i\leq n}$ sont deux à deux distincts.*

Démonstration. Cela découle des propriétés des *matrices de Vandermonde* comme V, *e.g.*, lorsque $n = m$, $\det(V) = \prod_{0\leq i<j\leq n}(x_j - x_i)$. □

Dans la suite, on s'intéresse au cas où le problème d'interpolation admet une unique solution et on le notera $(I)^f_n$.

Définition 5.2. *Soit $f : [a,b] \to \mathbb{R}$ et $n+1$ nœuds $(x_i)_{0\leq i\leq n}$ deux à deux distincts. La solution (unique) du problème $(I)^f_n$ est appelée* polynôme d'interpolation de f aux nœuds $(x_i)_{0\leq i\leq n}$ *: ce polynôme est noté $P_n(x\,;f)$.*

Notons que même si cela n'apparait pas dans la notation, $P_n(x\,;f)$ dépend des nœuds choisis.

Ce problème d'interpolation apparaît dans un contexte expérimental. Supposons que l'on souhaite calculer les valeurs d'une fonction f mais qu'on ne connaisse pas la fonction f et qu'il soit difficile d'en calculer des valeurs. Par exemple lorsque les valeurs de f s'obtiennent par intégration ou par résolution d'une équation non linéaire ou en sommant une série compliquée. On va donc faire une table de valeurs de f pour des valeurs du paramètre x convenablement choisies et construire une règle de calcul qui permette d'approcher les valeurs de f en dehors de ces valeurs discrètes. Cette règle de calcul peut par exemple être une formule d'interpolation. Le même problème apparaît lorsque la fonction f est obtenue comme résultat de mesures. Dans ce cas, on fait un nombre fini de mesures pour des valeurs distinctes de x et on souhaite avoir une formule pour calculer les valeurs de f en dehors des valeurs pour lesquelles on a fait des mesures.
Notons que pour espérer la réussite de ces techniques, il est naturel de supposer que l'on connaisse un minimum d'information sur la fonction f à interpoler.

5.2 La méthode d'interpolation de Lagrange

Définition 5.3. *Pour $j \in \{0,\dots,n\}$, le polynôme $L_j^{(n)}$ défini par*

$$L_j^{(n)}(x) = \prod_{\substack{i=0\\ i\neq j}}^{n} \frac{x-x_i}{x_j-x_i} = \frac{(x-x_0)\cdots(x-x_{j-1})\,(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)\cdots(x_j-x_{j-1})\,(x_j-x_{j+1})\cdots(x_j-x_n)},$$

est appelé interpolant de base de Lagrange *ou* polynôme de base de Lagrange associé à la suite $(x_i)_{0\leq i\leq n}$ et relatif au point x_j.

Proposition 5.4. *Pour $n \in \mathbb{N}$ fixé, les $(L_j^{(n)}(x))_{0\leq j\leq n}$ forment une base de l'espace vectoriel $\mathcal{P}_n$ que l'on appelle* base de Lagrange.

Proposition 5.5. *Les interpolants de base de Lagrange vérifient les propriétés suivantes :*

1. *Pour tout $j = 0,\dots,n$, si on note g_j la fonction de $[a,b]$ dans $\mathbb{R}$ définie par $\forall\, i = 0,\dots,n,\ g_j(x_i) = \delta_{ij}$, alors $P_n(x\,;g_j) = L_j^{(n)}(x)$;*
2. *Si on pose*

$$\pi_{n+1}(x) = \prod_{j=0}^{n}(x-x_j) \in \mathcal{P}_{n+1},$$

alors, pour tout $j = 0,\dots,n$,

$$L_j^{(n)}(x) = \frac{\pi_{n+1}(x)}{(x-x_j)\,\pi'_{n+1}(x_j)}.$$

3. Pour tout $k = 0, \ldots, n$, $x^k = \sum_{j=0}^{n} x_j^k \, L_j^{(n)}(x)$.

Démonstration. Exercice. □

La méthode d'interpolation de Lagrange consiste à écrire le polynôme d'interpolation sur la base de Lagrange.

Théorème 5.6. *Soit* $f : [a, b] \to \mathbb{R}$ *et* $n + 1$ *nœuds* $(x_i)_{0 \leq i \leq n}$ *deux à deux distincts. Le polynôme d'interpolation de* f *aux nœuds* $(x_i)_{0 \leq i \leq n}$ *s'écrit alors :*

$$P_n(x\,; f) = \sum_{j=0}^{n} f(x_j) \, L_j^{(n)}(x).$$

Démonstration. On doit montrer que le polynôme $Q(x) = \sum_{j=0}^{n} f(x_j) \, L_j^{(n)}(x)$ ainsi défini est solution du problème d'interpolation $(I)_n^f$. Il est clair que $Q \in \mathcal{P}_n$, *i.e.*, Q est de degré inférieur ou égal à n. De plus, pour tout $i = 0, \ldots, n$, $L_j^{(n)}(x_i) = \delta_{ij}$ où $\delta_{ij} = 1$ si $i = j$ et $\delta_{ij} = 0$ si $i \neq j$ d'où $Q(x_i) = f(x_i)$. □

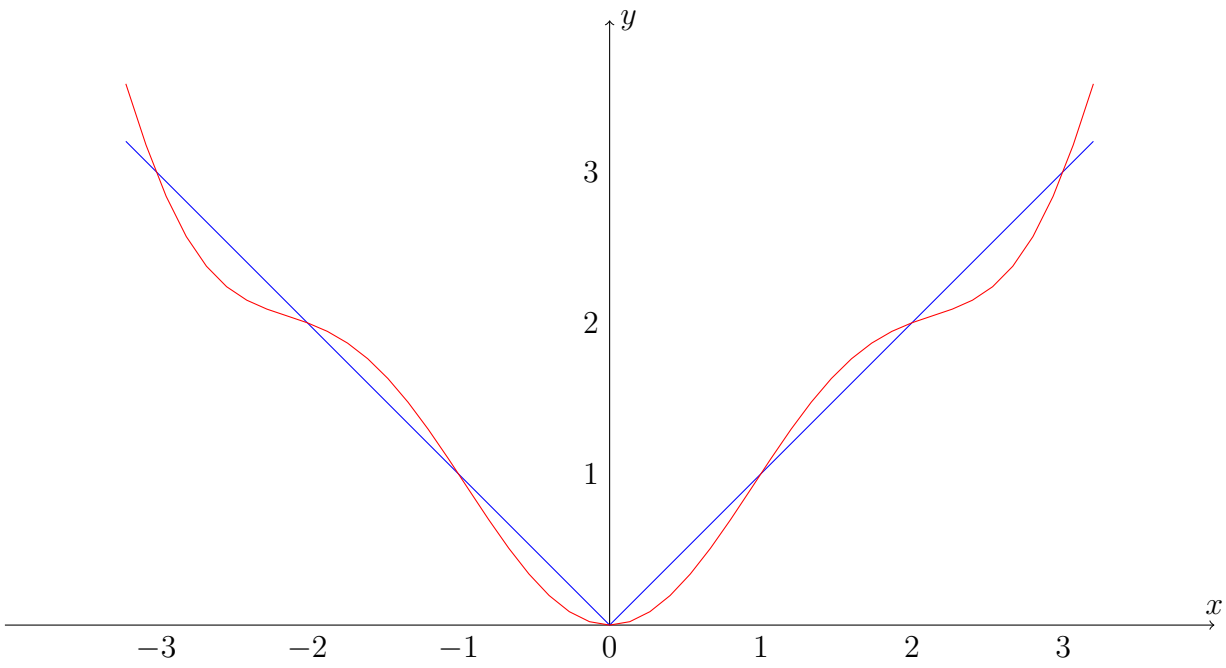
L'intérêt de la base de Lagrange par rapport à la base constituée des monômes est que nous n'avons pas besoin de résoudre un système linéaire de la forme (5.1) pour écrire le polynôme d'interpolation. L'expression de celui-ci dans la base de Lagrange s'écrit facilement. Par exemple si on choisit les nœuds $-1, 0, 1$, on obtient :

$$\begin{aligned} P_2(x\,; f) &= f(-1) \, \frac{(x-0)\,(x-1)}{(-1-0)\,(-1-1)} + f(0) \, \frac{(x+1)\,(x-1)}{(0+1)\,(0-1)} + f(1) \, \frac{(x+1)\,(x-0)}{(1+1)\,(1-0)}, \\ &= f(-1) \, \frac{x(x-1)}{2} - f(0)\,(x^2-1) + f(1) \, \frac{x\,(x+1)}{2}, \\ &= \frac{f(-1) - 2\,f(0) + f(1)}{2} \, x^2 + \frac{f(1) - f(-1)}{2} \, x + f(0). \end{aligned}$$

Exemple : Si on considère la fonction $f : [-4, 4] \to \mathbb{R}$, $x \mapsto |x|$. Alors le polynôme d'interpolation de f relatif aux nœuds $(x_j)_{0 \leq j \leq 8} = (-4, -3, -2, -1, 0, 1, 2, 3, 4)$ se décompose sur la base de Lagrange sous la forme $P_8(x\,; f) = \sum_{j=0}^{8} |x_j| \, L_j^{(8)}(x)$ et en développant on obtient :

$$P_8(x\,; f) = \frac{533}{420} \, x^2 - \frac{43}{144} \, x^4 + \frac{11}{360} \, x^6 - \frac{1}{1008} \, x^8.$$

Le graphe ci-dessous montre en bleu la courbe de f et en rouge celle de l'interpolant de Lagrange $P_8(x\,; f)$.



D'un point de vue efficacité pratique, on ne développe les $L_j^{(n)}(x)$ pour écrire le polynôme d'interpolation dans la base monomiale. Si l'on veut évaluer le polynôme d'interpolation en un point, on utilise la formule

$$P_n(x\,;f) = \pi_{n+1}(x) \sum_{j=0}^{n} \frac{f(x_j)}{\pi'_{n+1}(x_j)\,(x-x_j)},$$

qui nécessite moins d'opérations à virgule flottante. Ce calcul demeure tout de même coûteux.

Le principal inconvénient de la méthode d'interpolation de Lagrange est que le fait de rajouter un nœud change complètement les interpolants de base de Lagrange et on doit donc recalculer entièrement le polynôme $P_n(x\,;f)$. On va donc considérer une autre approche qui se comporte mieux lorsqu'on rajoute des nœuds.

Notons enfin que cette méthode permet aussi d'interpoler un nuage de points. Au lieu de se donner une fonction f, on se donne une suite de valeurs discrètes $(b_i)_{0\leq i\leq n}$ aux nœuds $(x_i)_{0\leq i\leq n}$ et on cherche un polynôme P_n tel que $P_n(x_i) = b_i$ pour $i = 0, \ldots, n$. On obtient exactement les mêmes résultats en remplaçant $f(x_i)$ par b_i dans tout ce qui précède.

5.3 Effectivité de l'interpolation : interpolant de Newton

5.3.1 Base d'interpolation de Newton

Définition 5.7. *Les polynômes* $N_j^{(n)}$ *définis pour* $j = 0, \ldots, n$ *par :*

$$\begin{cases} N_0^{(n)}(x) &= 1, \\ N_1^{(n)}(x) &= (x - x_0), \\ N_2^{(n)}(x) &= (x - x_0)\,(x - x_1), \\ \quad\vdots & \\ N_j^{(n)}(x) &= (x - x_0)\,(x - x_1)\cdots(x - x_{j-1}), \\ \quad\vdots & \\ N_n^{(n)}(x) &= (x - x_0)\,(x - x_1)\cdots(x - x_{n-1}), \end{cases}$$

sont appelés interpolants de base de Newton *ou* polynômes de base de Newton relatifs à la suite de points $(x_i)_{i=0,\ldots,n-1}$.

On remarque que là où on avait besoin de $n+1$ points pour définir les $L_j^{(n)}(x)$, $j = 0, \ldots, n$, la définition des $N_j^{(n)}(x)$, $j = 0, \ldots, n$, ne nécessite que n points.

Proposition 5.8. *Pour* $n \in \mathbb{N}$ *fixé, les* $(N_j^{(n)}(x))_{0 \le j \le n}$ *forment une base de l'espace vectoriel* $\mathcal{P}_n$ *que l'on appelle* base de Newton.

5.3.2 Expression de l'interpolant de Newton

Soit $f : [a, b] \to \mathbb{R}$ et n nœuds $(x_i)_{0 \le i \le n-1}$. Essayons d'écrire le polynôme d'interpolation sur la base de Newton. Autrement dit, cherchons des nombres α_i, $i = 0, \ldots, n$ tels que $P_n(x\,; f) = \sum_{i=0}^{n} \alpha_i\, N_i^{(n)}(x)$. On a :

$$P_n(x_0\,; f) = \alpha_0 = f(x_0) \Longrightarrow \alpha_0 = f(x_0)$$

$$P_n(x_1\,; f) = f(x_0) + \alpha_1\,(x_1 - x_0) = f(x_1) \Longrightarrow \alpha_1 = \frac{f(x_0) - f(x_1)}{x_0 - x_1}$$

$$P_n(x_2\,; f) = f(x_0) + \tfrac{f(x_0)-f(x_1)}{x_0-x_1}\,(x_2 - x_0) + \alpha_2\,(x_2 - x_0)\,(x_2 - x_1) = f(x_2)$$

$$\Longrightarrow \alpha_2 = \frac{\dfrac{f(x_0) - f(x_2)}{x_0 - x_2} - \dfrac{f(x_0) - f(x_1)}{x_0 - x_1}}{x_2 - x_1}$$

En posant

$$f[u, v] = \frac{f(u) - f(v)}{u - v},$$

on a alors

$$\alpha_1 = f[x_0, x_1], \quad \alpha_2 = \frac{f[x_0, x_2] - f[x_0, x_1]}{x_2 - x_1} = \frac{f[x_0, x_1] - f[x_1, x_2]}{x_0 - x_2}.$$

Définition 5.9. *Pour tout $k \in \mathbb{N}$, on appelle* différence divisée d'ordre k de f associée à la suite de points deux à deux distincts $(x_j)_{j\in\mathbb{N}}$ *la quantité* $f[x_0, x_1, \ldots, x_k]$ *définie par :*

$$f[x_0] = f(x_0), \quad \forall\, k \in \mathbb{N}^*,\ f[x_0, x_1, \ldots, x_k] = \frac{f[x_0, x_1, \ldots, x_{k-1}] - f[x_1, x_2, \ldots, x_k]}{x_0 - x_k}.$$

Théorème 5.10. *Avec les notations précédentes, on a*

$$P_n(x\,; f) = \sum_{k=0}^{n} f[x_0, x_1, \ldots, x_k]\, N_k^{(n)}(x).$$

Démonstration. Admis pour ce cours. □

Même si la définition de la base d'interpolation de Newton de $\mathcal{P}_n$ ne nécessite que la donnée de n nœuds, le coefficient $f[x_0, x_1, \ldots, x_n]$ de $N_n^{(n)}(x)$ fait intervenir le nœud x_n.

Corollaire 5.11. *Avec les notations précédentes, on a*

$$P_n(x\,; f) = P_{n-1}(x\,; f) + f[x_0, x_1, \ldots, x_n]\, N_n^{(n)}(x).$$

La formule de ce corollaire montre que si l'on écrit le polynôme d'interpolation sur la base de Newton, alors il est facile de rajouter un point.

L'évaluation du polynôme d'interpolation dans la base de Newton est relativement simple comparé à celui sur la base de Lagrange (schéma d'évaluation type Horner).

Proposition 5.12. *Pour tout $k \in \mathbb{N}$, on a :*

$$f[x_0, x_1, \ldots, x_k] = \sum_{j=0}^{k} \frac{f(x_j)}{\prod_{\substack{l=0 \\ l\neq j}}^{k} (x_j - x_l)} = \sum_{j=0}^{k} \frac{f(x_j)}{\pi'_{k+1}(x_j)}$$

Démonstration. Admis pour ce cours. □

Corollaire 5.13. *Soit $\mathcal{S}_{k+1}$ l'ensemble des permutations sur $\{0, 1, \ldots, k+1\}$. Pour tout $\sigma \in \mathcal{S}_{k+1}$, on a $f[x_{\sigma(0)}, x_{\sigma(1)}, \ldots, x_{\sigma(k)}] = f[x_0, x_1, \ldots, x_k]$.*

5.3.3 Algorithme de calcul des différences divisées

Le calcul pratique des différences divisées est basé sur la formule de récurrence de la définition 5.9. Nous avons le tableau suivant :

$$
\begin{array}{llllll}
x_0 & f[x_0] & & & & \\
 & & \searrow & & & \\
x_1 & f[x_1] & \longrightarrow f[x_0,x_1] & & & \\
 & & \searrow & \searrow & & \\
x_2 & f[x_2] & \longrightarrow f[x_1,x_2] & \longrightarrow f[x_0,x_1,x_2] & & \\
\vdots & \vdots & & & & \\
\vdots & \vdots & & & \ddots & \\
\vdots & \vdots & & & & \\
x_{k-2} & f[x_{k-2}] & & & & \\
 & & \searrow & & & \\
x_{k-1} & f[x_{k-1}] & \longrightarrow f[x_{k-2},x_{k-1}] & \quad\cdots & \cdots\ f[x_0,\ldots,x_{k-1}] & \\
 & & \searrow & \searrow & & \searrow \\
x_k & f[x_k] & \longrightarrow f[x_{k-1},x_k] & \longrightarrow f[x_{k-2},x_{k-1},x_k] & \cdots\ f[x_1,\ldots,x_k] & \longrightarrow f[x_0,\ldots,x_k]
\end{array}
$$

On en déduit immédiatement un algorithme de calcul des différences divisées.

On note que contrairement à ce qu'il se passait pour l'interpolation de Lagrange, l'ajout d'un nouveau nœud n'oblige pas à recalculer toutes les différences divisées. Plus précisément, passer de n à $n+1$ nœuds demande simplement le calcul de n différences divisées. Par exemple, pour passer de 3 à 4 points, on rajoute une ligne x_3 en dessous de la ligne x_2 dans notre tableau et on doit calculer $(f[x_3])$, $f[x_2,x_3]$, $f[x_1,x_2,x_3]$ et $f[x_0,x_1,x_2,x_3]$ soit 3 différences divisées.

5.4 Erreur d'interpolation

On va maintenant voir comment estimer l'erreur ponctuelle $|f(x) - P_n(x\,;f)|$.

Lemme 5.14. *Soit $(x_i)_{0\le i\le n}$ tels que, pour tout $i = 0,\ldots,n$, $x_i \in [a,b]$ et soit $P_n(x\,;f)$ le polynôme d'interpolation de f aux nœuds $(x_i)_{0\le i\le n}$. Alors, avec les notations précédentes, pour tout $x \in [a,b]$ tel que, pour tout $i = 0,\ldots,n$, $x \neq x_i$, on a :*

$$f(x) - P_n(x\,;f) = f[x_0,x_1,\ldots,x_n,x]\, N_{n+1}^{(n+1)}(x).$$

Démonstration. Admis pour ce cours. □

Lemme 5.15. *Si $f \in \mathcal{C}^n([a,b])$, alors :*

$$\exists\, \xi \in]a,b[, \quad f[x_0,x_1,\ldots,x_n] = \frac{1}{n!}\, f^{(n)}(\xi).$$

Démonstration. Admis pour ce cours. □

Les deux lemmes précédents mènent alors au résultat suivant et à son corollaire :

Théorème 5.16. *Soit $(x_i)_{0\leq i\leq n}$ tels que, pour tout $i = 0, \ldots, n$, $x_i \in [a, b]$ et soit $P_n(x\,; f)$ le polynôme d'interpolation de f aux nœuds $(x_i)_{0\leq i\leq n}$. Si $f \in \mathcal{C}^{n+1}([a, b])$, alors :*

$$\forall\, x \in [a, b],\, \exists\, \xi_x \in]a, b[, \quad f(x) - P_n(x\,; f) = \frac{1}{(n+1)!}\, f^{(n+1)}(\xi_x)\, N_{n+1}^{(n+1)}(x).$$

Corollaire 5.17. *Avec les mêmes hypothèses, on a :*

$$\forall\, x \in [a, b], \quad |f(x) - P_n(x\,; f)| \leq \frac{|N_{n+1}^{(n+1)}(x)|}{(n+1)!} \sup_{y\in[a,b]} |f^{(n+1)}(y)|.$$

Chapitre 6

Intégration numérique

Dans ce chapitre, nous allons voir comment approcher de façon numérique la valeur d'intégrales de la forme $I(f) = \int_a^b f(x)\,\mathrm{d}x$. Ce problème est d'autant plus intéressant qu'en pratique on ne connait pas forcément l'expression symbolique de f et que même si c'est le cas, la plupart des fonctions n'admettent pas de primitives pouvant s'exprimer à l'aide de fonctions élémentaires.

6.1 Introduction et méthodes classiques

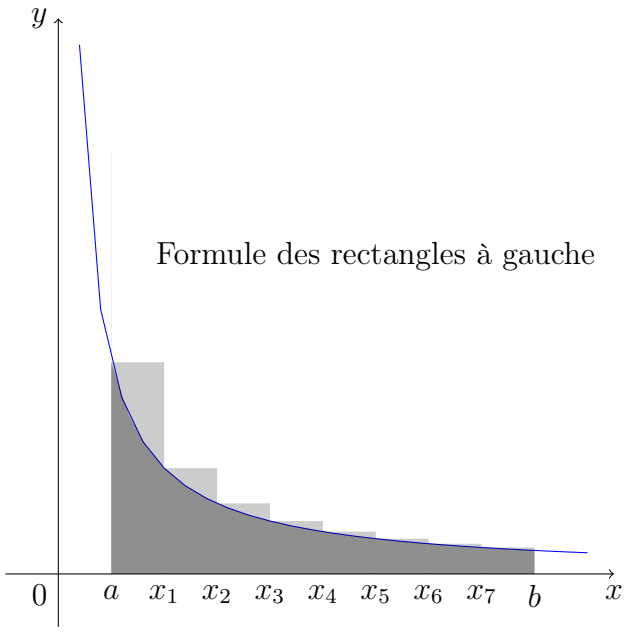
On supposera dans ce qui suit que les fonctions que l'on cherche à intégrer numériquement sont continues sur l'intervalle $[a, b]$. Soit $x_0 = a < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$ une subdivision de l'intervalle $[a, b]$. La théorie élémentaire de l'intégration implique :

$$I(f) = \int_a^b f(x)\mathrm{d}x = \lim_{n\to+\infty} \underbrace{\sum_{j=0}^{n-1} f(\xi_j)(x_{j+1} - x_j)}_{\text{Somme de Riemann}}, \quad \text{avec } \forall\, j,\ \xi_j \in [x_j, x_{j+1}].$$

Différents choix des ξ_j mènent aux méthodes classiques :

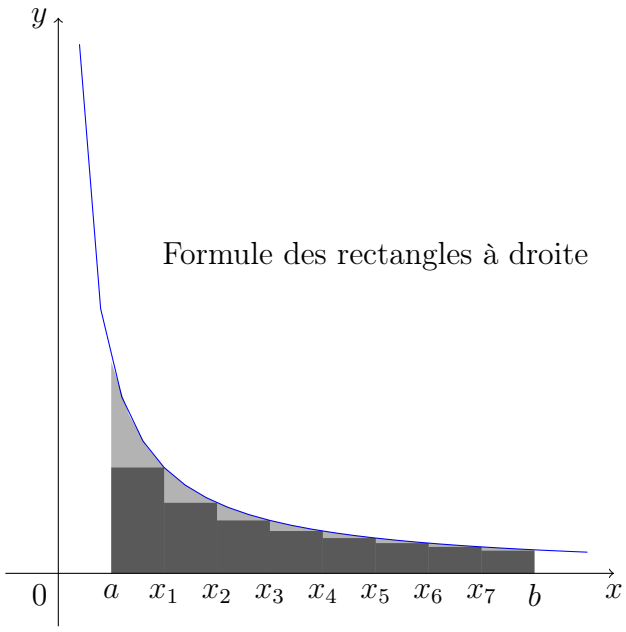
1. $\xi_j = x_j$ donne la *formule des rectangles à gauche* $I_{rg}(f)$:

$$I_{rg}(f) = \sum_{j=0}^{n-1} f(x_j)\,(x_{j+1} - x_j).$$



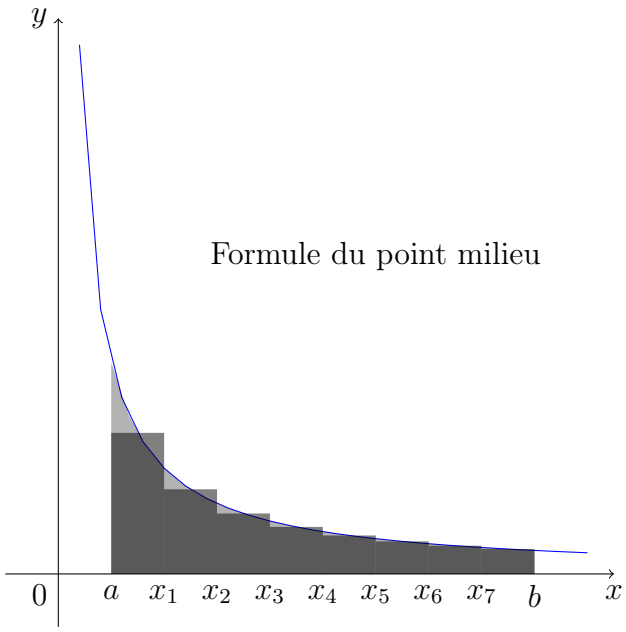
2. $\xi_j = x_{j+1}$ donne la *formule des rectangles à droite* $I_{rd}(f)$:

$$I_{rd}(f) = \sum_{j=0}^{n-1} f(x_{j+1})\,(x_{j+1} - x_j).$$



3. $\xi_j = \frac{x_j + x_{j+1}}{2}$ donne la *formule du point milieu* $I_{pm}(f)$:

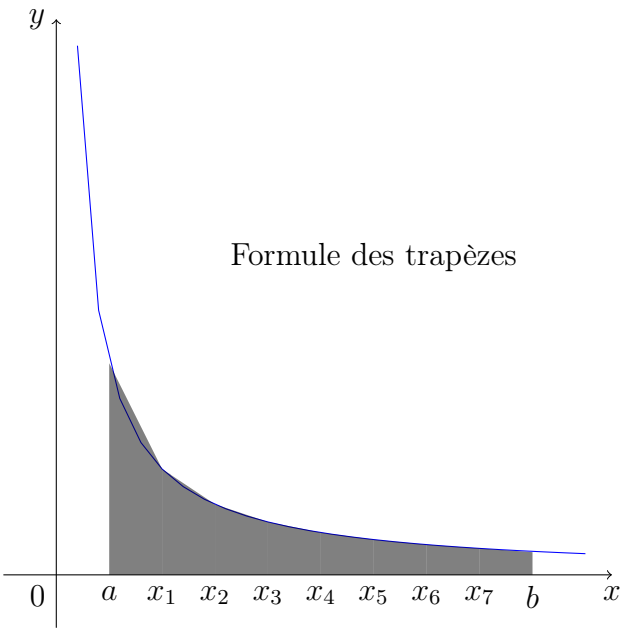
$$I_{pm}(f) = \sum_{j=0}^{n-1} f\left(\frac{x_j + x_{j+1}}{2}\right)\,(x_{j+1} - x_j).$$



Remarquons que les méthodes précédentes reviennent à interpoler f sur chaque intervalle $[x_j, x_{j+1}]$ par le polynôme d'interpolation de degré 0 relatif à l'unique nœud ξ_j. Ces formules seront donc exactes pour les fonctions constantes sur $[a, b]$ et en particulier pour $f \in \mathcal{P}_0$.

L'autre méthode classique est la *méthode des trapèzes* basée sur la formule :

$$I_t(f) = \sum_{j=0}^{n-1} \frac{f(x_j) - f(x_{j+1})}{2} (x_{j+1} - x_j).$$



La méthode des trapèzes revient à interpoler f sur chaque intervalle $[x_j, x_{j+1}]$ par le polynôme d'interpolation de degré 1. Cette formule sera donc exacte pour $f \in \mathcal{P}_1$.

6.2 Formalisation de l'intégration approchée

Nous allons maintenant définir un cadre d'étude général au problème de l'intégration approchée et donner un certain nombre de résultats généraux qui seront admis pour ce cours.

Soit $\mathcal{C}([a,b])$ l'espace vectoriel des fonctions continues sur l'intervalle $[a,b]$ de $\mathbb{R}$ et f une fonction de $\mathcal{C}([a,b])$. On suppose que l'on connaît au moins les valeurs de f en certains points $x_0, x_1, \ldots, x_n$ de l'intervalle $[a,b]$. On cherche alors une *formule d'intégration approchée* de la forme

$$I(f) = \int_a^b f(x)\,\mathrm{d}x \approx \sum_{k=0}^{n} \lambda_k\, f(x_k) = \tilde{I}^{(n)}(f),$$

où les λ_k sont à déterminer. On parle aussi de *méthode d'intégration numérique* ou *formule de quadrature*.

Définition 6.1. *Une méthode d'intégration numérique est dite* d'ordre N *($N \in \mathbb{N}$) si elle est exacte sur $\mathcal{P}_N$.*

Par exemple, la méthode des rectangles à gauche ou à droite et la méthode du point milieu sont d'ordre 0 et celle des trapèzes est d'ordre 1.

En pratique, connaissant les valeurs de f aux points $x_0, \ldots, x_n$, on remplace f par le polynôme d'interpolation $\sum_{k=0}^{n} f(x_k)\, L_k^{(n)}(x)$ écrit dans la base de Lagrange. On a alors la formule d'intégration approchée :

$$\tilde{I}^{(n)}(f) = \sum_{k=0}^{n} A_k^{(n)}\, f(x_k), \quad A_k^{(n)} = \int_a^b L_k^{(n)}(x)\,\mathrm{d}x, \tag{6.1}$$

qui est exacte sur $\mathcal{P}_n([a,b])$ (admis).

Théorème 6.2. *Soit $f \in \mathcal{C}^{n+1}([a,b])$ et $\tilde{I}^{(n)}(f)$ donnée par (6.1). Alors on a la majoration suivante de l'erreur d'intégration :*

$$|I(f) - \tilde{I}^{(n)}(f)| \leq \frac{M_{n+1}}{(n+1)!} \int_a^b |\pi_{n+1}(x)|\,\mathrm{d}x,\ M_{n+1} = \sup_{x\in[a,b]} |f^{(n+1)}(x)|,\ \pi_{n+1}(x) = \prod_{j=0}^{n}(x - x_j).$$

6.3 Formules de Newton-Côtes

D'après ce qui précède, pour obtenir notre formule d'intégration approchée, on doit donc calculer les

$$A_k^{(n)} = \int_a^b L_k^{(n)}(x)\,\mathrm{d}x.$$

Pour ceci, nous allons supposer que les points d'interpolation sont équidistants, *i.e.*, $x_{j+1} - x_j$ ne dépend pas de j, que $x_0 = a$, $x_n = b$ et $n \geq 1$.

Proposition 6.3. *Pour $k = 0, 1, \ldots, n$, on a :*

(i)

$$A_k^{(n)} = \frac{(b-a)}{n}\,\frac{(-1)^{n-k}}{k!\,(n-k)!}\int_0^n \prod_{\substack{j=0\\ j\neq k}}^{n} (y-j)\,\mathrm{d}y.$$

(ii) $A_{n-k}^{(n)} = A_k^{(n)}$.

Démonstration. Le (i) s'obtient par un changement de variable $x = a + y\,\frac{b-a}{n}$. Pour le (ii), on fait le changement d'indice $k = n - k$ en remarquant que $n - (n-k) = k$. □

Pour $n = 1$, on obtient $A_0^{(1)} = A_1^{(1)} = \frac{b-a}{2}$ d'où $\tilde{I}^{(1)}(f) = \frac{b-a}{2}(f(a) + f(b))$ et on retrouve la formule des trapèzes.

Pour $n = 2$, on trouve $A_0^{(2)} = A_2^{(2)} = \frac{b-a}{6}$ et $A_1^{(2)} = \frac{4\,(b-a)}{6}$ d'où

$$\tilde{I}^{(2)}(f) = \frac{b-a}{6}\left(f(a) + 4\,f(\frac{a+b}{2}) + f(b)\right) (\textit{formule de Simpson}).$$

Calcul pratique des coefficients : Pour calculer les coefficients $A_k^{(n)}$, on peut utiliser le fait que $\tilde{I}^{(n)}$ est exacte sur $\mathcal{P}_n([a,b])$. Par exemple pour $n = 1$, $a = -1$ et $b = 1$, on a $\tilde{I}^{(1)}(f) = A_0^{(1)}\,f(-1) + A_1^{(1)}\,f(1)$. Or $\tilde{I}^{(1)}$ est exacte sur $\mathcal{P}_1([-1,1])$ donc $\tilde{I}^{(1)}(1) = I(1) = \int_{-1}^{1} 1\,\mathrm{d}x = 2$ et $\tilde{I}^{(1)}(x) = I(x) = \int_{-1}^{1} x\,\mathrm{d}x = 0$. On obtient donc le système linéaire

$$\left\{\begin{array}{rcl} A_0^{(1)} + A_1^{(1)} & = & 2, \\ -A_0^{(1)} + A_1^{(1)} & = & 0, \end{array}\right.$$

d'où $A_0^{(1)} = A_1^{(1)} = 1$. Remarquons que cette formule n'est pas exacte sur $\mathcal{P}_2([-1,1])$ puisque $I(x^2) = \int_{-1}^{1} x^2\,\mathrm{d}x = \frac{2}{3}$ alors que $\tilde{I}^{(1)}(x^2) = 2$.

Théorème 6.4. *Considérons l'erreur $\mathcal{E}_n(f) = I(f) - \sum_{i=0}^{n} A_i^{(n)}\,f(x_i)$. Alors :*

1. *Si n est impair et si $f \in \mathcal{C}^{n+1}([a,b])$, alors il existe $\xi \in [a,b]$ tel que :*

$$\mathcal{E}_n(f) = \left(\frac{b-a}{n}\right)^{n+2} \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t\,(t-1)\cdots(t-n)\,\mathrm{d}t.$$

2. *Si n est pair et si $f \in \mathcal{C}^{n+2}([a,b])$, alors il existe $\xi \in [a,b]$ tel que :*

$$\mathcal{E}_n(f) = \left(\frac{b-a}{n}\right)^{n+3} \frac{f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t^2\,(t-1)\cdots(t-n)\,\mathrm{d}t.$$

Pour $n = 2$, si $f \in \mathcal{C}^4([a,b])$, alors l'erreur d'approximation commise par la formule de Simpson vaut $-h^5 \dfrac{f^{(4)}(\xi)}{90}$ où $h = \dfrac{b-a}{2}$ et $\xi \in [a,b]$.

6.4 Stabilité des méthodes d'intégration

La stabilité d'une méthode numérique mesure la « sensibilité de la méthode aux erreurs de calculs ». Considérons une formule d'intégration approchée $\tilde{I}^{(n)}(f) = \sum_{k=0}^n A_k^{(n)} f(x_k)$. Supposons maintenant que les valeurs calculées des $f(x_k)$ ne soient pas exactes. On a : $\sum_{k=0}^n A_k^{(n)} (f(x_k) + \epsilon_k) - \sum_{k=0}^n A_k^{(n)} f(x_k) = \sum_{k=0}^n A_k^{(n)} \epsilon_k$. Par conséquent,

$$\left| \sum_{k=0}^n A_k^{(n)} \epsilon_k \right| \leq \left(\max_{0 \leq k \leq n} |\epsilon_k| \right) \sum_{k=0}^n |A_k^{(n)}|,$$

et le terme $\sum_{k=0}^n |A_k^{(n)}|$ dépend de la méthode.

Définition 6.5. *La formule d'intégration numérique $\tilde{I}^{(n)}(f) = \sum_{k=0}^n A_k^{(n)} f(x_k)$ est dite* stable *s'il existe $M \in \mathbb{R}$ tel que : $\forall\, n \in \mathbb{N},\ \forall\, (\epsilon_0, \ldots, \epsilon_n) \in \mathbb{R}^{n+1},\ |\sum_{k=0}^n A_k^{(n)} \epsilon_k| \leq M \max_{0 \leq k \leq n} |\epsilon_k|$.*

Théorème 6.6. *Avec les notations précédentes, une condition nécessaire et suffisante de stabilité est qu'il existe $M \in \mathbb{R}$ (indépendant de n) tel que $\sum_{k=0}^n |A_k^{(n)}| \leq M$.*

Concernant les formules de Newton-Côtes, on peut montrer que pour certaines valeurs de k, $\lim_{n \to \infty} |A_k^{(n)}| = +\infty$ de sorte que pour de grandes valeurs de n ces formules ne sont pas stables.

6.5 Formules d'intégration composées

Les formules d'intégration composées sont les plus utilisées en pratique. Le principe de ces formules consiste à décomposer l'intervalle $[a,b]$ en k intervalles $[a_i, a_{i+1}]$, $i = 0, \ldots, k-1$. Grâce à la relation de Chasles, on écrit alors :

$$I(f) = \int_a^b f(x)\,\mathrm{d}x = \sum_{i=0}^{k-1} \underbrace{\int_{a_i}^{a_{i+1}} f(x)\,\mathrm{d}x}_{I_i(f)},$$

et on approche chaque $I_i(f)$ par une formule d'intégration numérique vu précédemment. Notons que pour la stabilité, il est judicieux de choisir une formule avec un n petit comme par exemple celle de Simpson ($n = 2$).

Ces méthodes composées sont d'autant plus intéressantes que l'erreur d'approximation diminue lorsque la taille de l'intervalle diminue. Par exemple, avec la formule de Simpson obtenue précédemment, si l'on subdivise l'intervalle $[a, b]$ en k sous intervalles avec k pair, on obtient la formule d'intégration suivante :

$$\frac{h}{3}\left(f(a_0) + 2\sum_{i=1}^{k/2-1} f(a_{2\,i}) + 4\sum_{i=1}^{k/2} f(a_{2\,i-1}) + f(a_k)\right) \text{ (formule de Simpson composée),}$$

avec $h = \frac{b-a}{k}$, $a_0 = a$, $a_k = b$ et $a_i = a_{i-1} + h$. Lorsque $f \in \mathcal{C}^4([a, b])$, l'erreur d'approximation de cette formule composée est alors de $-k\,h^5\,\dfrac{f^{(4)}(\xi)}{180}$ où $h = \dfrac{b-a}{k}$ et $\xi \in [a, b]$.

Chapitre 7

Résolution d'équations et de systèmes d'équations non linéaires

Le dernier chapitre de ce cours est consacré à la résolution d'équations et de systèmes d'équations non linéaires.

On considère tout d'abord une fonction $f : \mathbb{R} \to \mathbb{R}$ d'une seule variable réelle et on cherche à résoudre l'équation $f(x) = 0$ c'est-à-dire trouver une valeur approchée $\overline{x}$ d'un réel $\tilde{x}$ vérifiant $f(\tilde{x}) = 0$.

La mise en oeuvre pratique des méthodes que nous allons voir nécessite la donnée d'une tolérance sur la solution que l'on cherche à calculer. L'algorithme numérique utilisé doit alors avoir un *critère d'arrêt* dépendant de cette tolérance et nous assurant que la solution calculée a bien la précision recherchée. En fonction de la méthode utilisée, on peut parfois savoir à l'avance combien d'étapes de l'algorithme sont nécessaires pour obtenir la précision recherchée (méthode de dichotomie) ou alors il nous faut à chaque étape vérifier une condition nous permettant d'arrêter le processus lorsque l'on est certain d'avoir obtenu la précision requise sur la solution (méthodes de points fixes).

Pour comparer les différentes méthodes de résolution que l'on va considérer, on utilise les notions suivantes de *vitesse de convergence* d'une suite :

Définition 7.1. *Soit $(x_n)_{n\in\mathbb{N}}$ une suite convergente et soit $\tilde{x}$ sa limite.*

1. *On dit que la convergence de $(x_n)_{n\in\mathbb{N}}$ est* linéaire de facteur $K \in]0, 1[$ *s'il existe $n_0 \in \mathbb{N}$ tel que, pour tout $n \geq n_0$, $|x_{n+1} - \tilde{x}| \leq K\,|x_n - \tilde{x}|$.*

2. *On dit que la convergence de $(x_n)_{n\in\mathbb{N}}$ est* superlinéaire d'ordre $p \in \mathbb{N}$, $p > 1$ *s'il existe $n_0 \in \mathbb{N}$ et $K > 0$ tels que, pour tout $n \geq n_0$, $|x_{n+1} - \tilde{x}| \leq K\,|x_n - \tilde{x}|^p$. Si $p = 2$, on parle de* convergence quadratique *et si $p = 3$ on parle de* convergence cubique.

Remarquons que K n'est pas unique et qu'en pratique il peut être difficile de prouver la

convergence d'une méthode d'autant plus qu'il faut tenir compte des erreurs d'arrondis. On utilise en général les notions de convergence plus faible suivantes :

Définition 7.2. *Soit $(x_n)_{n\in\mathbb{N}}$ une suite convergent vers une limite $\tilde{x}$. On dit que la convergence de $(x_n)_{n\in\mathbb{N}}$ est* linéaire de facteur K *(resp.* superlinéaire d'ordre p*) s'il existe une suite $(y_n)_{n\in\mathbb{N}}$ convergent vers 0, linéaire de facteur K (resp. superlinéaire d'ordre p) au sens de la définition 7.1 telle que $|x_n - \tilde{x}| \leq y_n$.*

Soit $d_n = -\log_{10}(|x_n - \tilde{x}|)$ une « mesure » du nombre de décimales exactes de x_n. Si la convergence est d'ordre p, alors asymptotiquement, on a $|x_{n+1} - \tilde{x}| \sim K\,|x_n - \tilde{x}|^p$ d'où $-d_{n+1} \sim \log_{10}(K) - p\,d_n$ et donc asymptotiquement x_{n+1} a p fois plus de décimales exactes que x_n. Ainsi, l'ordre p de la convergence représente asymptotiquement le facteur multiplicatif du nombre de décimales exactes que l'on gagne à chaque itération. Nous avons donc intérêt à ce qu'il soit le plus grand possible.

7.1 Méthode de dichotomie

La méthode classique de dichotomie est une méthode de localisation des racines d'une équation $f(x) = 0$ basée sur le théorème des valeurs intermédiaires : si f est continue sur $[a, b]$ et $f(a)\,f(b) < 0$, alors il existe $\tilde{x} \in]a, b[$ tel que $f(\tilde{x}) = 0$. L'idée est donc de partir d'un intervalle $[a, b]$ vérifiant la propriété $f(a)\,f(b) < 0$, de le scinder en deux intervalles $[a, c]$ et $[c, b]$ avec $c = \frac{a+b}{2}$, et de tester les bornes des nouveaux intervalles (on calcule $f(a)\,f(c)$ et $f(c)\,f(b)$) pour en trouver un (au moins) qui vérifie encore la propriété, *i.e.*, $f(a)\,f(c) < 0$ ou/et $f(c)\,f(b) < 0$. On itère ensuite ce procédé un certain nombre de fois dépendant de la précision que l'on recherche sur la solution (voir Théorème 7.3 ci-dessous). On obtient l'algorithme suivant :

Entrèes : la fonction[1] f, $(a, b) \in \mathbb{R}^2$ tels que f est continue sur $[a, b]$ et $f(a)\,f(b) < 0$ et la précision ϵ.
Sortie : x_{k+1} valeur approchée de $\tilde{x}$ solution de $f(\tilde{x}) = 0$ à ϵ près.

1. $x_0 \leftarrow a$, $y_0 \leftarrow b$;
2. Pour k de 0 à $E\left(\dfrac{\ln(b-a) - \ln(\epsilon)}{\ln(2)}\right)$ par pas de 1, faire :
   - Si $f(x_k)\,f\left(\dfrac{x_k + y_k}{2}\right) > 0$, alors $x_{k+1} \leftarrow \dfrac{x_k + y_k}{2}$, $y_{k+1} \leftarrow y_k$;
   - Si $f(x_k)\,f\left(\dfrac{x_k + y_k}{2}\right) < 0$, alors $x_{k+1} \leftarrow x_k$, $y_{k+1} \leftarrow \dfrac{x_k + y_k}{2}$;
   - Sinon retourner $\dfrac{x_k + y_k}{2}$;

[1] Il suffit en fait de connaitre un moyen d'évaluer les valeurs de la fonction

3. Retourner x_{k+1}.

Cet algorithme construit une suite de segments emboîtés contenant tous la solution $\tilde{x}$. À chaque passage dans la boucle nous devons calculer une évaluation de f. Remarquons qu'en pratique, avec les arrondis, « > 0 » et « < 0 » ne veulent rien dire ! On démontre maintenant que cet algorithme est correct dans le sens où il calcule bien une valeur approchée de la solution à ϵ près.

Théorème 7.3. *Le nombre minimum d'itérations de la méthode de dichotomie nécessaire pour approcher $\tilde{x}$ à ϵ près est $E\left(\frac{\ln(b-a)-\ln(\epsilon)}{\ln(2)}\right)+1$, où $E(x)$ désigne la partie entière d'un réel x.*

Démonstration. À la première itération, la longueur de l'intervalle est $\frac{b-a}{2}$, ..., à la $n$ième itération, la longueur de l'intervalle est $\frac{b-a}{2^n}$. L'erreur commise à l'étape n est donc majorée par $\frac{b-a}{2^n}$. Le nombre n d'itérations à effectuer doit alors vérifier $\frac{b-a}{2^n} \leq \epsilon$ qui est équivalent à $n \geq \frac{\ln(b-a)-\ln(\epsilon)}{\ln(2)}$ d'où le résultat. □

Proposition 7.4. *La convergence de la méthode de dichotomie est linéaire de facteur $\frac{1}{2}$.*

Démonstration. Prendre $y_n = \frac{b-a}{2^n}$ dans la définition 7.2. □

7.2 Méthode du point fixe

La méthode itérative du point fixe que nous allons décrire est aussi appelée méthode des approximations successives.

Définition 7.5. *Soit $g : \mathbb{R} \to \mathbb{R}$. On dit que $x \in \mathbb{R}$* est un point fixe de *g si $g(x) = x$.*

Le principe de la méthode du point fixe est d'associer à l'équation $f(x) = 0$ une équation de point fixe $g(x) = x$ de sorte que trouver une solution de $f(x) = 0$ équivaut à trouver un point fixe de g. La technique pour approximer le point fixe de g est alors basée sur le résultat suivant :

Lemme 7.6. *Soit $(x_n)_{n\in\mathbb{N}}$ la suite définie par $x_0 \in \mathbb{R}$ donné et $x_{n+1} = g(x_n)$. Si $(x_n)_{n\in\mathbb{N}}$ est convergente et g est continue, alors la limite de $(x_n)_{n\in\mathbb{N}}$ est un point fixe de g.*

Nous devons donc trouver des conditions sur g pour que la suite $(x_n)_{n\in\mathbb{N}}$ définie ci-dessus converge.

Définition 7.7. *Soit $g : \Omega \subseteq \mathbb{R} \to \mathbb{R}$. On dit que g* est lipschitzienne sur Ω de constante de lipschitz γ (ou γ-lipschitzienne) *si pour tout $(x, y) \in \Omega^2$, on a $|g(x) - g(y)| \leq \gamma\,|x - y|$. On dit que g* est strictement contractante sur Ω *si g est γ-lipschitzienne sur Ω avec $\gamma < 1$.*

On remarque que g lipschitzienne sur Ω implique en particulier g continue sur Ω.

Théorème 7.8 (Théorème du point fixe)**.** *Soit g une application strictement contractante sur un intervalle $[a,b] \subset \mathbb{R}$ de constante de Lipschitz $\gamma < 1$. Supposons que l'intervalle $[a,b]$ soit stable sous g,* i.e.*, $g([a,b]) \subseteq [a,b]$ ou encore pour tout $x \in [a,b]$, $g(x) \in [a,b]$. Alors g admet un unique point fixe $x^* \in [a,b]$ et la suite définie par $x_{n+1} = g(x_n)$ converge linéairement de facteur γ vers x^* pour tout point initial $x_0 \in [a,b]$. De plus,*

$$\forall\, n \in \mathbb{N},\ |x_n - x^*| \leq \frac{\gamma^n}{1-\gamma}\,|x_1 - x_0|.$$

Démonstration. Admis pour ce cours. □

On remarque que l'erreur est d'autant plus petite que γ est proche de 0. De plus, on peut montrer que l'on a aussi

$$\forall\, n \in \mathbb{N},\ |x_n - x^*| \leq \frac{\gamma}{1-\gamma}\,|x_n - x_{n-1}|,$$

et si $\gamma \leq \frac{1}{2}$, alors $|x_n - x^*| \leq |x_n - x_{n-1}|$. Dans ce cas, on pourra utiliser le test d'arrêt $|x_n - x_{n-1}| < \epsilon$ qui certifiera une précision ϵ sur le résultat.

La proposition suivante donne un critère pour tester le fait qu'une fonction soit contractante sur un intervalle donné.

Proposition 7.9. *Soit g une fonction dérivable sur l'intervalle $[a,b]$. Si sa dérivée g' vérifie $\max_{x\in[a,b]} |g'(x)| = L < 1$, alors g est strictement contractante sur $[a,b]$ de constante de Lipschitz L.*

Démonstration. Utiliser le théorème des accroissements finis. □

On en déduit alors la proposition suivante :

Proposition 7.10. *Soit $x^* \in [a,b]$ un point fixe d'une fonction $g \in \mathcal{C}^1([a,b])$.*

- *Si $|g'(x^*)| < 1$, alors il existe un intervalle $[\alpha,\beta] \subseteq [a,b]$ contenant x^* pour lequel la suite définie par $x_0 \in [\alpha,\beta]$ et $x_{n+1} = g(x_n)$ converge vers x^* ;*
- *Si $|g'(x^*)| > 1$, alors pour tout $x_0 \neq x^*$, la suite définie par x_0 et $x_{n+1} = g(x_n)$ ne converge pas vers x^* ;*
- *Si $|g'(x^*)| = 1$, on ne peut pas conclure.*

En pratique, on « estime » $g'(x^*)$, *i.e.*, on en a une valeur approchée $\overline{g'(x^*)}$. Si $|\overline{g'(x^*)}| > 1$, alors on élimine la méthode et si $|\overline{g'(x^*)}| < 1$, on cherche un intervalle $[\alpha,\beta] \subseteq [a,b]$ dans lequel $\max_{x\in[\alpha,\beta]} |g'(x)| < 1$ et $g([\alpha,\beta]) \subseteq [\alpha,\beta]$.

Revenons maintenant à notre problème initial où on cherche à résoudre une équation $f(x) = 0$. Posons $g(x) = x - f(x)$ de sorte que trouver les solutions de $f(x) = 0$ soit équivalent

à trouver les points fixes de g. D'après le théorème du point fixe (Théorème 7.8), une condition suffisante pour que g admette un point fixe dans l'intervalle $[a, b]$ est que $[a, b]$ soit stable sous g et que g soit strictement contractante sur $[a, b]$ de constante de Lipschitz $\gamma < 1$. On a alors (conséquence directe de la définition d'une fonction contractante)

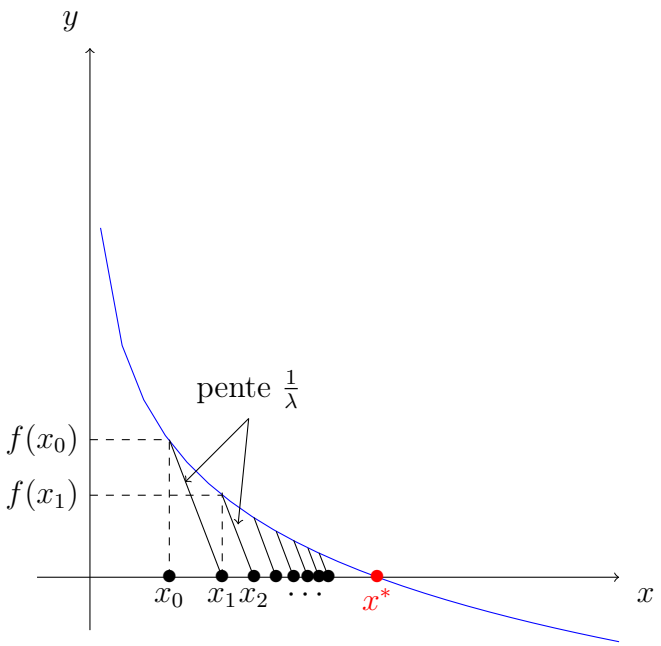
$$\forall\, x \in [a, b], \quad |g'(x)| < \gamma \Longleftrightarrow |1 - f'(x)| < \gamma,$$

qui implique en particulier que f' ne s'annule pas sur $[a, b]$ et que f est donc monotone sur $[a, b]$.

Étant donnée f, le choix précédent de g est restrictif et on peut choisir $g(x) = x - \lambda\, f(x)$ où λ est une constante arbitraire non nulle. Comme précédemment, une condition suffisante pour que g ait un point fixe dans $[a, b]$ est que $[a, b]$ soit stable sous g et que g soit strictement contractante sur $[a, b]$ de constante de Lipschitz $\gamma < 1$. On obtient alors

$$\forall\, x \in [a, b], \quad |1 - \lambda\, f'(x)| < \gamma < 1, \tag{7.1}$$

ce qui implique en particulier que f' ne change pas de signe sur $[a, b]$ et que λ est du même signe que f'. Géométriquement, on construit la suite des itérés $x_{n+1} = x_n - \lambda\, f(x_n)$. En remarquant que la droite de pente μ et passant par le point $(x_n, f(x_n))$ a pour équation $y = f(x_n) + \mu\,(x - x_n)$ et coupe l'axe des abscisses en $x = x_n - \frac{f(x_n)}{\mu}$, on voit que x_{n+1} s'obtient comme point d'intersection de la droite de pente $\frac{1}{\lambda}$ passant par le point $(x_n, f(x_n))$ avec l'axe des abscisses. En itérant ce procédé, on obtient la méthode illustrée sur le graphique suivant :



La proposition suivante donne des précisions sur l'orde de convergence d'une méthode de point fixe.

Proposition 7.11. *On considère l'équation $g(x) = x$ où g est une fonction au moins $p+1$ fois dérivable avec $p \geq 1$. Supposons que les hypothèses du théorème 7.8 soient vérifiées de sorte que g admette un unique point fixe $x^* \in [a,b]$. Si $g'(x^*) = g''(x^*) = \cdots = g^{(p)}(x^*) = 0$ et $g^{(p+1)}(x^*) \neq 0$, alors la convergence de la méthode $x_{n+1} = g(x_n)$ est superlinéaire d'ordre $p+1$.*

Démonstration. Utiliser la formule de Taylor. □

7.3 Méthode de Newton

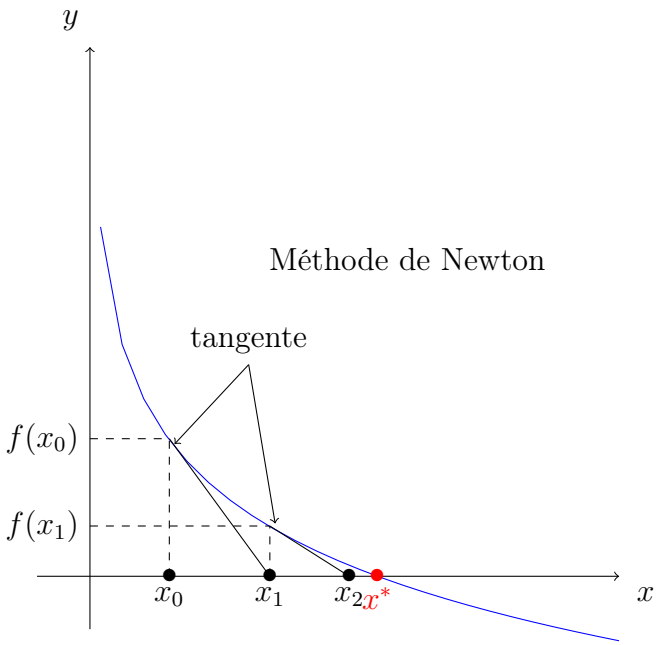
Si on regarde l'équation (7.1), on voit que la méthode convergera d'autant plus vite que la constante γ est petite. Ceci motive donc le fait de remplacer la constante λ par $\frac{1}{f'(x)}$ et de poser $g(x) = x - \frac{f(x)}{f'(x)}$.

Définition 7.12. *La* fonction d'itération de Newton associée à l'équation $f(x) = 0$ sur $[a,b]$ *est*

$$\mathcal{N} : \left\{ \begin{array}{ccc} [a,b] & \to & \mathbb{R}, \\ x & \mapsto & \mathcal{N}(x) = x - \dfrac{f(x)}{f'(x)}. \end{array} \right.$$

Cette fonction est définie pour f dérivable sur $[a,b]$ et telle que f' ne s'annule pas sur $[a,b]$.

Géométriquement, la suite des itérés de Newton se construit comme suit : étant donné $x_n \in [a,b]$, on cherche à construire x_{n+1} tel que $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$. En reprenant, le raisonnement développé dans la sous-section précédente, on voit que x_{n+1} s'obtient comme point d'intersection de la droite tangente au graphe de f en $(x_n, f(x_n))$ avec l'axe des abscisses. On illustre ceci sur le graphique suivant :



Théorème 7.13 (Théorème de convergence locale)**.** *Soit f une fonction de classe $\mathcal{C}^2$ sur un intervalle $[a,b]$ de $\mathbb{R}$. On suppose qu'il existe $\tilde{x} \in [a,b]$ tel que $f(\tilde{x}) = 0$ et $f'(\tilde{x}) \neq 0$ ($\tilde{x}$ est un zéro simple de f). Alors il existe $\epsilon > 0$, tel que pour tout $x_0 \in [\tilde{x} - \epsilon, \tilde{x} + \epsilon]$, la suite des itérés de Newton donnée par $x_{n+1} = \mathcal{N}(x_n)$ pour $n \geq 1$ est bien définie, reste dans l'intervalle $[\tilde{x} - \epsilon, \tilde{x} + \epsilon]$ et converge vers $\tilde{x}$ quand n tend vers l'infini. De plus, cette convergence est (au moins) quadratique.*

Démonstration. Admis pour ce cours. □

Exemple : Prenons l'exemple du calcul de la racine carrée d'un nombre réel $a > 0$. On cherche alors à résoudre l'équation $f(x) = 0$ avec $f(x) = x^2 - a$. On a alors

$$\mathcal{N}(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - a}{2\,x} = \frac{1}{2}\left(x + \frac{a}{x}\right).$$

Dans le cas $a = 2$, si l'on prend comme point de départ $x_0 = 1$ on obtient les valeurs suivantes des itérés de Newton :

$$\begin{aligned} x_0 &= 1,000000000000000 \\ x_1 &= 1,500000000000000 \\ x_2 &= 1,416666666666667 \\ x_3 &= 1,414215686274510 \\ x_4 &= 1.414213562374690 \\ x_5 &= 1,414213562373095 \end{aligned}$$

La valeur de $\sqrt{2}$ donnée par MATLAB en format `long` est $1,414213562373095$. Le nombre de décimales justes double approximativement à chaque itération ce qui est bien cohérent avec le fait que la convergence de la méthode de Newton est quadratique. Notons que si on utilise la méthode de dichotomie sur l'intervalle $[1,2]$, alors nous avons besoin de 51 itérations pour avoir une valeur approchée de $\sqrt{2}$ à 10^{-15} près (voir Théorème 7.3).

Le théorème 7.13 suppose que $\tilde{x}$ est un zéro simple de f. Nous avons la généralisation suivante au cas d'un zéro de multiplicité quelconque.

Théorème 7.14. *Avec les notations, précédentes, si $\tilde{x}$ est un zéro de multiplicité m de f,* i.e., *$f(x^*) = f'(x^*) = \cdots = f^{(m-1)}(x^*) = 0$ et $f^{(m)}(x^*) \neq 0$, alors la méthode itérative définie par $x_{n+1} = \mathcal{N}_m(x_n)$ avec $\mathcal{N}_m(x_n) = x - m\,\dfrac{f(x)}{f'(x)}$ est d'ordre supérieure ou égal à 2.*

Finalement, nous avons le résultat « global » suivant :

Théorème 7.15 (Théorème de convergence globale)**.** *Soit f une fonction de classe $\mathcal{C}^2$ sur un intervalle $[a,b]$ de $\mathbb{R}$ vérifiant :*

- $f(a)\,f(b) < 0$,
- $\forall\, x \in [a,b],\ f'(x) \neq 0$ *(f strictement monotone),*

- $\forall\, x \in [a,b],\ f''(x) \neq 0$ *(concavité dans le même sens sur* $[a,b]$*).*

Alors, en choisissant $x_0 \in [a,b]$ *tel que* $f(x_0)\, f''(x_0) > 0$*, la suite* $(x_n)_{n\in\mathbb{N}}$ *définie par* x_0 *et* $x_{n+1} = \mathcal{N}(x_n)$ *converge vers l'unique solution de* $f(x) = 0$ *dans* $[a,b]$*.*

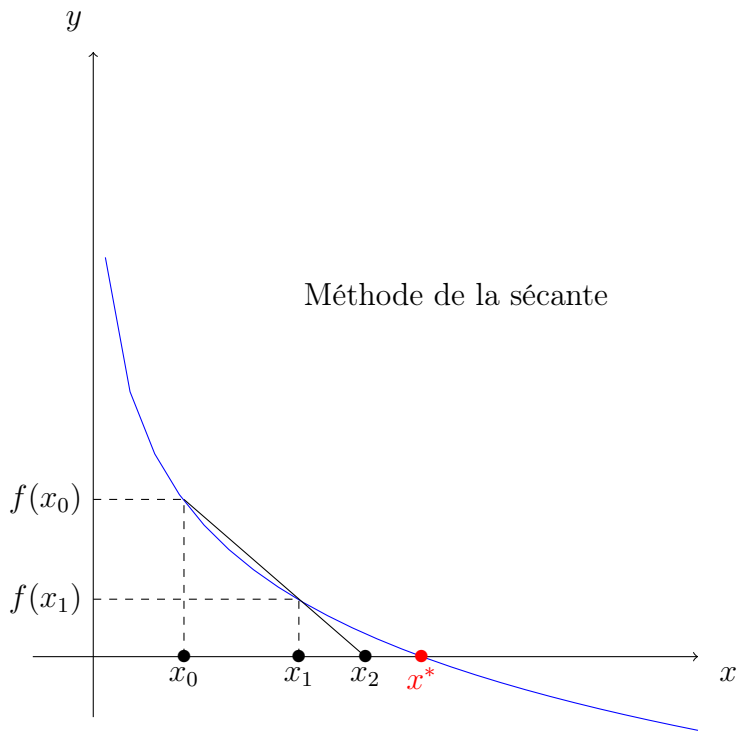
Démonstration. Admis pour ce cours. □

7.4 Méthode de la sécante

La méthode de Newton étudiée dans la sous-section précédente présente le désavantage de nécessiter le calcul de la dérivée de la fonction f qui peut s'avérer difficile. Gardons en tête qu'on ne connait pas nécessairement une expression symbolique de f. Partant de ce constat, l'idée de la *méthode de la sécante* est de remplacer la dérivée f' de f qui apparait dans la méthode de Newton par une différence divisée (voir la définition 5.9). La méthode de la sécante est alors donnée par l'itération suivante :

$$x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, x_{n-1}]}, \quad \text{où} \quad f[x_n, x_{n-1}] = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}. \tag{7.2}$$

Cette méthode doit alors être initialisée par deux points x_0 et x_1 à partir desquels on peut utiliser la formule précédente pour calculer x_2 et les itérés suivants.



Théorème 7.16. *Soit* f *une fonction de classe* $\mathcal{C}^2$ *sur un intervalle* $[a,b]$ *de* $\mathbb{R}$*. On suppose qu'il existe* $\tilde{x} \in [a,b]$ *tel que* $f(\tilde{x}) = 0$ *et* $f'(\tilde{x}) \neq 0$ *(*$\tilde{x}$ *est un zéro simple de* f*). Alors il existe*

$\epsilon > 0$, tel que pour tout x_0, $x_1 \in [\tilde{x} - \epsilon, \tilde{x} + \epsilon]$, la suite des itérés de la méthode de la sécante donnée par (7.2) pour $n \geq 1$ est bien définie, reste dans l'intervalle $[\tilde{x} - \epsilon, \tilde{x} + \epsilon]$ et converge vers $\tilde{x}$ quand n tend vers l'infini. De plus, cette convergence est d'ordre $p = \frac{1+\sqrt{5}}{2} \approx 1,618$ (nombre d'or).

Démonstration. Admis pour ce cours. □

Exemple : Reprenons l'exemple du calcul de la racine carrée d'un réel $a > 0$ en résolvant $f(x) = 0$ pour $f(x) = x^2 - a$. La suite des itérés de la méthode de la sécante est donnée par :

$$x_{n+1} = x_n - \frac{x_n^2 - a}{\frac{(x_n^2 - a) - (x_{n-1}^2 - a)}{x_n - x_{n-1}}} = x_n - \frac{x_n^2 - a}{x_n + x_{n-1}}.$$

Dans le cas $a = 2$, si l'on prend comme points de départ $x_0 = x_1 = 1$ on obtient les valeurs suivantes des itérés de la méthode de la sécante :

$$\begin{aligned} x_0 &= 1,000000000000000 \\ x_1 &= 1,000000000000000 \\ x_2 &= 1,500000000000000 \\ x_3 &= 1,400000000000000 \\ x_4 &= 1,413793103448276 \\ x_5 &= 1,414215686274510 \\ x_6 &= 1,414213562057320 \\ x_7 &= 1,414213562373095 \end{aligned}$$

On voit donc que l'on a besoin de plus d'itérations que pour la méthode de Newton pour avoir la même précision. Ceci vient du fait que l'ordre de cette méthode est plus petit que celui de la méthode de Newton. Par contre, on n'a pas besoin de calculer la dérivée.

7.5 Systèmes d'équations non linéaires

On considère maintenant un système d'équations non linéaires donné par une fonction $f : \mathbb{R}^n \to \mathbb{R}^n$, $x = (x_1 \dots x_n)^T \mapsto f(x) = (f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))^T$. On cherche donc un vecteur $x = (x_1 \dots x_n)^T \in \mathbb{R}^n$ tel que

$$f(x) = 0_{\mathbb{R}^n} \iff \begin{cases} f_1(x_1, \dots, x_n) &= 0, \\ \qquad \vdots & \\ f_n(x_1, \dots, x_n) &= 0. \end{cases}$$

La méthode décrite à la fin de la section 7.2 se généralise immédiatement à ce cadre en définissant l'itération

$$x^{(n+1)} = x^{(n)} + M^{-1} f(x^{(n)}), \tag{7.3}$$

où M est une certaine matrice, et nous avons les mêmes résultats de convergence que dans le cas d'une seule équation.

Définition 7.17. *La* matrice jacobienne *d'une fonction $f : \mathbb{R}^n \to \mathbb{R}^n$ notée J_f est définie (lorsqu'elle existe) par :*

$$\forall\, x = (x_1 \dots x_n)^T \in \mathbb{R}^n, \quad J_f(x) = \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1}(x) & \dfrac{\partial f_1}{\partial x_2}(x) & \dots & \dfrac{\partial f_1}{\partial x_n}(x) \\ \dfrac{\partial f_2}{\partial x_1}(x) & \dfrac{\partial f_2}{\partial x_2}(x) & \dots & \dfrac{\partial f_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial f_n}{\partial x_1}(x) & \dfrac{\partial f_n}{\partial x_2}(x) & \dots & \dfrac{\partial f_n}{\partial x_n}(x) \end{pmatrix}.$$

La méthode de Newton se généralise naturellement au cas des systèmes d'équations non linéaires de la manière suivante : on choisit $x^{(0)} \in \mathbb{R}^n$ et on utilise la formule d'itération

$$x^{(n+1)} = x^{(n)} - J_f\left(x^{(n)}\right)^{-1} f(x^{(n)}), \tag{7.4}$$

où $J_f\left(x^{(n)}\right)^{-1}$ désigne l'inverse de la matrice jacobienne de f évaluée en $x^{(n)}$.

Théorème 7.18. *Soit $f : \mathbb{R}^n \to \mathbb{R}^n$ une fonction de classe $\mathcal{C}^2$ sur une boule fermée B de $\mathbb{R}^n$. On suppose qu'il existe un zéro $\tilde{x}$ de f dans B et que $J_f(\tilde{x})$ est inversible. Alors il existe $\epsilon > 0$ tel que pour tout $x^{(0)} \in B$ tel que $\|x^{(0)} - \tilde{x}\| \leq \epsilon$, la suite des itérés de la méthode de Newton définie par (7.4) est bien définie et converge vers $\tilde{x}$ quand n tend vers l'infini.*

Calculer l'itéré $n+1$ à partir de l'itéré n en utilisant la formule (7.4) nécessite d'inverser la matrice $J_f\left(x^{(n)}\right)$. Or, calculer l'inverse d'une matrice peut s'avérer couteux. Par conséquent, nous ré-écrivons la formule d'itération (7.4) sous la forme

$$J_f\left(x^{(n)}\right)\left(x^{(n+1)} - x^{(n)}\right) = -f\left(x^{(n)}\right), \tag{7.5}$$

de sorte qu'à chaque itération, le calcul de l'inverse d'une matrice est remplacé par la résolution d'un système d'équations linéaires ce qui est asymptotiquement moins couteux en nombre d'opérations à virgule flottante (voir la sous-section 2.2.4 du chapitre 2).

Exemple : Considérons le système d'équations non linéaires :

$$(S) : \begin{cases} x_1^2 + 2\,x_1 - x_2^2 - 2 &= 0, \\ x_1^3 + 3\,x_1\,x_2^2 - x_2^3 - 3 &= 0. \end{cases}$$

Avec les notations précédentes, cela correspond à $n = 2$, $f_1(x_1, x_2) = x_1^2 + 2\,x_1 - x_2^2 - 2$, et $f_2(x_1, x_2) = x_1^3 + 3\,x_1\,x_2^2 - x_2^3 - 3$. La matrice jacobienne de f est alors :

$$J_f(x_1, x_2) = \begin{pmatrix} 2\,x_1 + 2 & -2\,x_2 \\ 3\,(x_1^2 + x_2^2) & 6\,x_1\,x_2 - 3\,x_2^2 \end{pmatrix}.$$

Partant du point $x^{(0)} = (1 \quad -1)^T$, calculons le premier itéré de la méthode de Newton pour résoudre le système (S). Pour $n = 1$, la formule d'itération (7.5) s'écrit :

$$J_f\left(x^{(0)}\right)\left(x^{(1)} - x^{(0)}\right) = -f\left(x^{(0)}\right),$$

c'est-à-dire

$$\begin{pmatrix} 4 & 2 \\ 6 & -9 \end{pmatrix} \begin{pmatrix} x_1^{(1)} - 1 \\ x_2^{(1)} + 1 \end{pmatrix} = - \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

En résolvant ce système linéaire, on trouve $x_1^{(1)} - 1 = -\frac{1}{12}$ et $x_2^{(1)} + 1 = \frac{1}{6}$ de sorte que $x^{(1)} = \left(\frac{11}{12} \quad -\frac{5}{6}\right)^T$.

La méthode de la sécante ne se généralise pas facilement au cas de plusieurs équations. En pratique, pour résoudre un système de plusieurs équations non linéaires, soit on utilise la méthode de Newton, soit on utilise une méthode type (7.3) mais en ajustant la matrice M au bout d'un certain nombre d'itérations. En général, on prend pour M une matrice assez proche de la jacobienne J_f de f pour que la convergence soit d'un ordre supérieur à 1. On obtient ainsi des *méthodes de Newton généralisées* qui sont surtout utilisées dans le cadre de l'optimisation.

Remerciements

Ce document a été rédigé avec l'aide de Samir Adly, Paola Boito et Marc Rybowicz, enseignants-chercheurs à la faculté des sciences et techniques de l'Université de Limoges. Je remercie en particulier Paola Boito pour la rédaction de la section 4.4.

Bibliographie

[1] Jean-Claude Bajard et Jean-Michel Muller. Calcul et arithmétique des ordinateurs. Informatique et Systèmes d'Information, Hermes Science publications, Lavoisier, 2004.

[2] Jean-Pierre Demailly. Analyse numérique et équations différentielles. Collections Grenoble Sciences, 1991.

[3] Nicolas J. Higham. Acuracy and Stability of Numerical Algorithms. Second Edition, SIAM, 2002.

[4] Jean-Michel Muller. Arithmétique des ordinateurs. Masson Paris, 1989.

[5] Michelle Schatzman. Analyse numérique, cours et exercices pour la licence. InterEditions, Paris, 1991.

[6] Alain Yger et Jacques-Arthur Weil. Mathématiques L3 - Mathématiques appliquées (Cours complet avec 500 tests et exercices corrigés, 890p et Dvd). Pearson, 2009.