TELECOM SudParis

Calcul Scientifique

François Desbouvries
Emmanuel Monfrini
Yohan Petetin
Wojciech Pieczynski

Septembre 2016

Table des matières

Introduction **3**

1 Solutions approchées de certaines équations différentielles **5**
1.1 Introduction 5
1.2 Méthode des différences finies 5
1.2.1 En dimension un 5
1.2.2 En dimension deux 8
1.2.3 Conclusion 9
1.3 Formulation variationnelle des problèmes aux limites elliptiques 9
1.3.1 Problème de Dirichlet 9
1.3.2 Formulation abstraite 10
1.3.3 Approximation dans un espace de dimension finie 11
1.3.4 Convergence de la solution approchée 12
1.3.5 Synthèse 13
1.4 Interpolation de Lagrange 14
1.4.1 Éléments finis de Lagrange 14
1.4.2 Solutions approchées 16
1.4.3 Analyse de la méthode et convergence des solutions approchées 19
1.5 Conclusion 22

2 Analyse Numérique Matricielle **25**
2.1 Rappels et Compléments d'algèbre linéaire 25
2.1.1 Similarité et diagonalisation d'une matrice 25
2.1.2 Équivalence et Décomposition en valeurs singulières 26
2.1.3 Triangularisation unitaire de Schur et applications 27
2.1.4 Normes vectorielles et matricielles 28
2.2 Conditionnement d'un problème d'algèbre linéaire 30
2.2.1 Conditionnement d'un système linéaire 30
2.2.2 Conditionnement d'un problème de valeurs propres 33
2.2.3 Remarques importantes sur le conditionnement 34
2.3 Algorithmes de résolution de systèmes linéaires 35
2.3.1 Méthode de Gauss et factorisation LU 36
2.3.2 Méthode de Householder et factorisation QR 38
2.3.3 Résolution d'un système au sens des Moindres Carrés 41
2.4 Algorithmes de calcul de valeurs propres 43

2.4.1 Méthode de Jacobi 44
2.4.2 Algorithme QR 46

3 Approximations Stochastiques **49**
3.1 Introduction 49
3.2 Intégration par la méthode de Monte Carlo 49
3.3 Générations des variables aléatoires 52
3.3.1 Fonction de répartition inversible 52
3.3.2 Loi de Gauss et lois associées 53
3.3.3 Méthode des lois marginales 54
3.3.4 Méthode d'acceptation-rejet 54
3.4 Méthodes de Monte Carlo par Chaînes de Markov (MCMC) 56
3.4.1 Cas discret 56
3.4.2 Cas continu 58

Travaux dirigés **61**

Travaux pratiques **71**

Bibliographie **75**

Introduction

Origine des problèmes de calcul scientifique

Dans de très nombreux problèmes de la physique ou des sciences de l'ingénieur il est nécessaire d'effectuer des calculs numériques sur ordinateur. Considérons en effet les quelques exemples suivants :

— on souhaite connaître (même de façon approximative) la solution d'une équation différentielle partielle. Si cette équation n'admet pas de solution analytique, la démarche consiste à remplacer le problème exact (que l'on ne sait pas résoudre explicitement) par un problème approché plus simple à résoudre. Une solution possible consiste ainsi à discrétiser l'équation ainsi que l'espace des solutions. Après discrétisation, le nouveau problème consiste alors à résoudre un système linéaire, éventuellement de très grande dimension ;

— Connaissant la loi initiale et la matrice de transition $\mathbf{P}$ d'une chaîne de Markov d'ordre 1, on souhaite calculer la loi stationnaire de la chaîne. Le calcul de cette loi stationnaire revient au calcul d'un vecteur propre à gauche de $\mathbf{P}$ associé à la valeur propre 1 ;

— On souhaite évaluer une intégrale (ou un moment probabiliste : espérance, variance ...) que l'on ne sait pas calculer exactement. La seule solution consiste alors à proposer une approximation numérique, par exemple en découpant le segment d'intégration en petits intervalles consécutifs, ou en utilisant des techniques de simulation stochastique.

Problématique

Dans la plupart des systèmes informatiques les calculs numériques sont effectués avec une arithmétique à virgule flottante. Tout nombre réel x est donc représenté sous la forme $x = m \times \beta^e$. Dans cette écriture, $m = \pm \sum_{i=1}^{t} d_i \beta^{-i}$ est la *mantisse* et e *l'exposant* du nombre réel considéré, β étant la base de numération.

Le fait de devoir proposer une valeur numérique en faisant pour cela tourner un algorithme sur un ordinateur pose donc d'emblée un certain nombre de problèmes :

— $e \in \{-M, +N\}$ où M et N sont deux entiers. Cette limitation des valeurs de l'exposant a pour conséquence importante que les nombres de valeur absolue trop grande ne peuvent être représentés **(problème d' overflow)**, tandis que les

nombres de valeur absolue trop petite sont remplacés indûment par zéro **(problème d' underflow)** ;

— du fait de la longueur t finie de la mantisse, la représentation des nombres réels fait apparaître une **erreur d'arrondi** ; on ne résoudra donc pas nécessairement le problème qui correspond aux données exactes dont on dispose, mais à une représentation approchée de ces données ;

— un algorithme de calcul consiste en une suite d'opérations qui transforme progressivement les données d'un problème en sa solution. Lors de ces étapes successives les erreurs d'arrondi peuvent s'accumuler, voire s'amplifier de façon dramatique. Du fait de cette **propagation d'erreurs d'arrondi**, l'algorithme peut alors être mis en défaut, ou retourner une valeur aberrante, très différente de la vraie solution. **Quelle confiance peut-on alors accorder à la valeur qui apparaît sur l'écran ?**

— Certains problèmes (tels que la recherche des valeurs propres d'une matrice : cf. le paragraphe 2.4 du chapitre 1) ne peuvent être résolus en un nombre fini d'opérations. Les méthodes numériques employées seront donc nécessairement itératives, et proposeront une suite de résultats approchés convergeant vers la solution exacte. En pratique, il sera nécessaire d'arrêter l'algorithme au bout d'un nombre fini d'itérations ; **aux erreurs d'arrondi (inévitables) se rajouteront donc dans ce type de méthodes les erreurs de troncature** ;

— Enfin, au delà de la précision numérique obtenue (- c'est-à-dire de la confiance que l'on peut accorder au résultat qui s'affiche sur l'écran), se rajoutent deux critères de qualité importants d'un algorithme : la place mémoire requise, et le *coût* (et donc le temps ...) de calcul. Ces problèmes ne sont pas anodins : en effet, si chacun sait que les performances des ordinateurs augmentent de façon exponentielle avec le temps, la contribution à la rapidité de calcul de l'amélioration sans cesse croissante des techniques mathématiques semble moins bien connue. C'est ainsi par exemple que dans la période allant de 1973 à 1983, les capacités de calcul des ordinateurs les plus puissants étaient multipliés par 1000 ; *mais dans le même temps, l'amélioration de certaines techniques numériques faisait gagner un autre facteur 1000.* Et c'est la conjugaison de ces deux performances qui fait qu'aujourd'hui il est possible de calculer un avion complet en moins d'une journée de calcul d'un Cray I.

Objectifs du cours

Ce cours a pour but de sensibiliser les étudiants aux problèmes posés par la **résolution numérique** d'un problème donné. Le premier chapitre présente quelques éléments de résolution numérique d'équations différentielles aux dérivées partielles. Le second chapitre est consacré à l'analyse numérique matricielle, c'est-à-dire à un ensemble de techniques permettant numériquement : soit de résoudre un système linéaire, soit d'extraire les éléments propres d'une matrice. Enfin le chapitre 3 s'intéresse aux techniques stochastiques de calcul en particulier pour approcher numériquement certaines intégrales.

Chapitre 1

Solutions approchées de certaines équations différentielles

1.1 Introduction

Un grand nombre de phénomènes peut être décrit par les solutions des équations différentielles aux dérivées partielles. Ainsi les différentes variables intervenant en mécanique ou électromagnétisme sont liées par le biais des équations différentielles aux dérivées partielles et des conditions «aux limites» propres à une situation donnée. Dans le domaine de l'économie ou de la santé, les choses sont plus complexes et donc moins faciles à appréhender ; cependant, les équations différentielles peuvent encore jouer un rôle, ne serait ce qu'en fournissant une «première approximation». En particulier, les possibilités d'enrichissement rapide offertes par la bourse ces dernières années sont à l'origine d'une activité de recherche fébrile visant les modélisations mathématiques, utilisant notamment différentes équations différentielles, du comportement des marchés financiers.

L'objet de ce chapitre est de présenter, en utilisant un exemple simple, la méthode des différences finies et celle des éléments finis. Cette dernière méthode est bien adaptée aux traitements modernes des équations différentielles et l'une des plus largement utilisées.

1.2 Méthode des différences finies

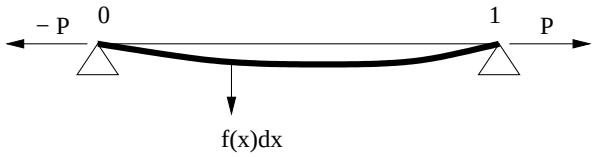
1.2.1 En dimension un

Considérons le problème suivant : étant donné 2 fonctions c et f continues sur $[0,1]$ et 2 constantes α et β, trouver une fonction u 2 fois dérivable sur $[0,1]$ qui vérifie

$$\begin{cases} -u''(x) + c(x)u(x) = f(x), \quad 0 < x < 1 \\ u(0) = \alpha,\ u(1) = \beta. \end{cases} \tag{1.1}$$

Une telle équation régit, par exemple, le phénomène de fléchissement d'une poutre fixée à ces extrémités.

Exemple 1.2.1 *Soit une poutre de longueur 1 appuyée à ses extrémités. Elle est étirée selon son axe par une force P et soumise à une charge transversale $f(x)dx$ par unité de longueur.*



Alors le moment fléchissant $u(x)$ est solution de l'équation (1.1) avec $c(x)$ une fonction dépendant de P, du type de matériau et du moment d'inertie de la section de la poutre en x et $\alpha = \beta = 0$ ■

On peut montrer que, lorsque $c \geq 0$ sur $[0, 1]$, le problème a une solution et une seule, qui sera notée φ.

Mais en règle générale, il n'existe pas de méthode qui permette de calculer exactement la valeur de φ en un point quelconque de $[0, 1]$. Alors il reste la question de savoir comment approcher les valeurs de la solution d'aussi près que l'on veut. La méthode des différences finies est une méthode qui permet d'y parvenir en discrétisant l'intervalle $[0, 1]$ et l'équation et en approchant la solution du problème continu par la solution du problème discret.

Principe de la méthode des différences finies

On définit un maillage uniforme de pas $h = \frac{1}{N+1}$ avec $N \geq 1$ comme l'ensemble des points $x_i = ih$ avec $0 \leq i \leq N + 1$, appelés aussi nœuds. On définit alors le problème discret comme la recherche d'un vecteur $u_h = (u_1, u_2, \ldots, u_N)^T$ tel que u_i soit voisin de $\varphi(x_i)$. Le pas h est destiné à tendre vers 0 et permettra d'ajuster la qualité de l'approximation. La méthode des différences finies consiste alors à remplacer le problème continu par le problème discret, en calculant une approximation de la solution φ précisément aux nœuds du maillage.

Construction du problème discret

Il s'agit à présent de construire le problème discret. Pour cela, nous devons exprimer $\varphi''(x_i)$. Supposons φ quatre fois dérivable sur $[0, 1]$. La formule de Taylor permet d'écrire :

$$\varphi(x_{i+1}) = \varphi(x_i) + h\varphi'(x_i) + \frac{h^2}{2}\varphi''(x_i) + \frac{h^3}{6}\varphi^{(3)}(x_i) + \frac{h^4}{24}\varphi^{(4)}(x_i + \theta_i^+ h)$$

et

$$\varphi(x_{i-1}) = \varphi(x_i) - h\varphi'(x_i) + \frac{h^2}{2}\varphi''(x_i) - \frac{h^3}{6}\varphi^{(3)}(x_i) + \frac{h^4}{24}\varphi^{(4)}(x_i + \theta_i^- h)$$

avec $-1 < \theta_i^- < 0 < \theta_i^+ < 1$
On en déduit

$$-\varphi(x_{i+1}) + 2\varphi(x_i) - \varphi(x_{i-1}) = -h^2\varphi''(x_i) - \frac{h^4}{24}\left(\varphi^{(4)}(x_i + \theta_i^+ h) + \varphi^{(4)}(x_i + \theta_i^- h)\right)$$

D'après le théorême des valeurs intermédiaires :

— on sait que $\frac{1}{2}\left(\varphi^{(4)}(x_i+\theta_i^+h)+\varphi^{(4)}(x_i+\theta_i^-h)\right)$ est compris entre $\varphi^{(4)}(x_i+\theta_i^+h)$ et $\varphi^{(4)}(x_i+\theta_i^-h)$ et $\varphi^{(4)}$ continue sur $[0,1]$

— donc il existe $c \in [x_i+\theta_i^-h, x_i+\theta_i^+h]$ tel que $\varphi^{(4)}(c) = \frac{1}{2}\left(\varphi^{(4)}(x_i+\theta_i^+h)+\varphi^{(4)}(x_i+\theta_i^-h)\right)$

— on peut écrire c en fonction de x_i et h et on obtient

$$\varphi^{(4)}(x_i+\theta_i^+h)+\varphi^{(4)}(x_i+\theta_i^-h) = 2\varphi^{(4)}(x_i+\theta_i h) \quad \text{avec } |\theta_i| < \max(\theta_i^+, -\theta_i^-) < 1$$

D'où, en tout point x_i

$$-\varphi''(x_i) = \frac{1}{h^2}\left(-\varphi(x_{i+1})+2\varphi(x_i)-\varphi(x_{i-1})\right)+\frac{h^2}{12}\varphi^{(4)}(x_i+\theta_i h) \quad \text{avec } |\theta_i|<1 \quad (1.2)$$

Pour alléger l'écriture, posons $\varphi_i = \varphi(x_i)$, $c_i = c(x_i)$ et $f_i = f(x_i)$. Nous allons, pour chaque nœud du maillage, substituer l'égalité (1.2) dans l'équation (1.1 en tenant compte des conditions aux limites. Ceci aboutit au système ci-dessous, dont la forme des membres de gauche est à l'origine du nom de la méthode.

$$\begin{cases} -\dfrac{\alpha}{h^2}+\dfrac{2\varphi_1-\varphi_2}{h^2}+c_1\varphi_1 = f_1-\dfrac{h^2}{12}\varphi^{(4)}(x_1+\theta_1 h) \\ \dfrac{-\varphi_{i-1}+2\varphi_i-\varphi_{i+1}}{h^2}+c_i\varphi_i = f_i-\dfrac{h^2}{12}\varphi^{(4)}(x_i+\theta_i h), \quad 2\leq i\leq N-1 \\ \dfrac{-\varphi_{N-1}+2\varphi_N}{h^2}-\dfrac{\beta}{h^2}+c_N\varphi_N = f_N-\dfrac{h^2}{12}\varphi^{(4)}(x_N+\theta_N h) \end{cases}$$

Ce système d'équation peut se résumer par la forme matricielle $A_h\varphi_h = b_h+\epsilon_h(\varphi)$ en posant

$$A_h = \frac{1}{h^2}\begin{pmatrix} 2+c_1h^2 & -1 & & & \\ -1 & 2+c_2h^2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2+c_{N-1}h^2 & -1 \\ & & & -1 & 2+c_Nh^2 \end{pmatrix},$$

$$\varphi_h = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_{N-1} \\ \varphi_N \end{pmatrix}, \; b_h = \begin{pmatrix} f_1+\alpha/h^2 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N+\beta/h^2 \end{pmatrix} \text{ et } \epsilon_h(\varphi) = -\tfrac{h^2}{12}\begin{pmatrix} \varphi^{(4)}(x_1+\theta_1h) \\ \varphi^{(4)}(x_2+\theta_2h) \\ \vdots \\ \varphi^{(4)}(x_{N-1}+\theta_{N-1}h) \\ \varphi^{(4)}(x_N+\theta_Nh) \end{pmatrix}.$$

On remarquera que $\epsilon_h(\varphi)$ sera d'autant plus petit que h est petit, ce qui induit la définition suivante du problème discret associé au problème continu et correspondant au pas h :

Définition 1.2.1 (Problème discret) *Trouver $u_h \in \mathbb{R}^N$ solution de $A_hu_h = b_h$.*

Etude du problème discret et convergence

Le problème discret étant posé, il faut maintenant s'assurer que

(i) le système $A_h\varphi_h = b_h$ a bien une solution et une seule ;

(ii) la méthode converge lorsque h tend vers zéro, c'est-à-dire que $u_h - \varphi_h$ tend vers zéro.

Le premier point vient très aisément en constatant que A_h est inversible. En effet, A_h est symétrique et de plus, elle est définie positive lorsque la fonction c est positive ; ce qui est prouvé par l'égalité vraie pour tout vecteur v de $\mathbb{R}^N$

$$v^T A_h v = \sum_{i=1}^{N} c_i v_i^2 + \frac{1}{h^2}\left(v_1^2 + v_N^2 + \sum_{i=2}^{N}(v_i - v_{i-1})^2\right).$$

Le second point est plus délicat. En effet, la convergence dépend du comportement asymptotique de l'erreur de consistance $\epsilon_h(\varphi)$ et de la régularité de la solution. Néanmoins on peut montrer que la convergence est d'ordre h^2, ce qui est justifié par le théorème ci-dessous (qui sera admis).

Théorème 1.2.1 *Supposons la fonction c positive. Si la solution φ du problème continue est 4 fois dérivable sur $[0,1]$, on a la majoration :*

$$\max_{1\le i\le N} |u_i - \varphi(x_i)| = ||u_h - \varphi_h||_\infty \le h^2\Big(\frac{1}{96}\sup_{0\le x\le 1}|\varphi^{(4)}(x)|\Big)$$

Dire que la convergence de la méthode des différences finies appliquées au problème considéré est d'ordre h^2 s'interprète donc comme le fait que l'erreur au sens de la norme infinie entre la valeur vraie et l'apporximation est $O(h^2)$.

1.2.2 En dimension deux

Il est tout à fait possible d'utiliser la méthode des différences finies lorsque le nombre de dimensions du problème augmente. Le principe reste identique : on remplace le problème continu en un problème discret qui va approximer la solution en un nombre fini de points du domaine de définition.

Cependant, le choix des points n'est pas toujours aussi naturel qu'en dimension un. En effet, le plus simple est de construire un maillage uniforme du plan (avec le même pas h dans les deux directions) dont les nœuds seraient les intersections de lignes horizontales et verticales. L'exemple d'un tel cas est traité dans l'Annexe 1. Dans le cas d'un domaine dont les frontières ne coïncident pas avec les nœuds du maillage, il restera le problème de la bonne prise en compte des conditions aux limites. Par ailleurs, un maillage si régulier n'est peut-être pas la meilleure solution pour approximer la solution sur un domaine biscornu.

1.2.3 Conclusion

Ce qu'il faut retenir c'est que la méthode des différences finies est assez simple à mettre en œuvre, mais qu'elle ne permet d'obtenir une approximation de la solution d'une équation différentielle qu'en un nombre de points finis.
L'intérêt de la méthode variationnelle présentée dans la suite de ce chapitre est d'aboutir à une approximation de la solution en tous points du domaine de définition de l'équation.

1.3 Formulation variationnelle des problèmes aux limites elliptiques

1.3.1 Problème de Dirichlet

Soit Ω un sous-ensemble de $\mathbb{R}^2$ de frontière Γ, qui sera supposée de classe $\mathcal{C}^1$ par morceaux. On note $L^2(\Omega)$ l'ensemble des fonctions de carré intégrable, qui est un espace de Hilbert pour le produit scalaire $< f, g >= \int_\Omega f(x)g(x)dx$. Pour toute fonction réelle u sur Ω deux fois différentiable, on note Δu le Laplacien de u : $\Delta u(x,y) = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}$.
Rappelons la formule de Stokes, valable pour deux fonctions u et v «suffisamment régulières», ce que l'on supposera tout au long de ce chapitre :

$$-\int_\Omega (\Delta u) v dxdy = \int_\Omega \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} \right) dxdy - \int_\Gamma \frac{\partial u}{\partial v} v d\sigma \tag{1.3}$$

Considérons alors le «problème de Dirichlet » suivant :

Définition 1.3.1 (Problème de Dirichlet) *Pour $f \in L^2(\Omega)$, trouver u définie sur Ω et vérifiant*

$$-\Delta u = f \ \textit{dans}\ \Omega \tag{1.4}$$
$$u = 0 \ \textit{sur}\ \Gamma \tag{1.5}$$

Prenons une fonction v définie dans Ω, nulle sur Γ, et suffisamment régulière pour que la formule de Stokes soit vérifiée. En multipliant 1.4 par v, en appliquant la formule de Stokes, et en intégrant sur Ω, on a

$$\int_\Omega \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} \right) dxdy = \int_\Omega f v dxdy \tag{1.6}$$

pour tout v définie dans Ω, nulle sur Γ, et suffisamment régulière. 1.6 est appelé la formulation variationnelle du problème de Dirichlet.
Réciproquement, on montre que si 1.6 est vérifiée pour tout v définie dans Ω, nulle sur Γ, et suffisamment régulière, alors u est solution du problème de Dirichlet 1.4, 1.5.

1.3.2 Formulation abstraite

L'objet de ce sous-paragraphe est de présenter une version générale du problème 1.6, susceptible de fournir un cadre permettant d'affirmer l'existence et l'unicité de la solution u. On reconnaît dans le membre de droite de l'équation 1.6 une **forme linéaire continue** (f est fixée), et dans le membre de gauche une **forme bilinéaire symétrique**. Il s'agit alors de trouver u tel que l'image de tout v par la forme bilinéaire (à u fixée) soit égale à son image par la forme linéaire de droite (à f fixée). C'est ainsi que l'on définit les problèmes variationnels abstraits :

Définition 1.3.2 (Problème variationnel) *On se donne*

- *un espace de Hilbert E doté d'un produit scalaire $< \cdot, \cdot >$ et de la norme associée $\| \cdot \|$ (*i.e. $\|f\| = \sqrt{< f, f >}$*)*
- *une forme bilinéaire $u, v \mapsto a(u, v)$ continue sur $E \times E$,* i.e. *pour laquelle il existe une constante $k > 0$ telle que*

$$\forall u \in E,\ \forall v \in E,\ a(u, v) \leq k\|u\|\|v\| \tag{1.7}$$

- *une forme linéaire $v \mapsto L(v)$ continue sur E*

Trouver $u \in E$ tel que

$$\forall v \in E,\ a(u, v) = L(v) \tag{1.8}$$

est appelé un problème variationnel.

Remarque : Dans notre problème de Dirichlet précédent, $L(v) =< f, v >$.

Sans condition sur $a(u, v)$, rien n'assure que le problème ait une solution (ex : $a(\cdot, \cdot) = 0$). Par contre, on connaît une condition suffisante pour assurer l'existence, et même l'unicité, d'une solution du problème variationnel. Ceci est donné par le théorème suivant :

Théorème 1.3.1 (Lemme de Lax-Milgram) *Soit le problème variationnel : trouver $u \in E$ tel que $\forall v \in E,\ a(u, v) = L(v)$.*
Si $a(\cdot, \cdot)$ est coercitive, i.e. *il existe une constante $\alpha > 0$ telle que*

$$\forall v \in E,\ a(v, v) \geq \alpha\|v\|^2 \tag{1.9}$$

alors le problème admet une solution unique dans E.
De plus, si $a(\cdot, \cdot)$ est symétrique, l'élément $u \in E$ est par ailleurs caractérisé par

$$\frac{1}{2}a(u, u) - L(u) = \min_{v \in E} \left(\frac{1}{2}a(v, v) - L(v) \right). \tag{1.10}$$

La preuve de la caractérisation de la solution par la minimisation d'une fonctionnelle quadratique est laissée en exercice.

Exemple 1.3.1 *Un exemple très simple consiste à considérer $E = \mathbb{R}$; les formes bilinéaires symétriques continues coercitives sont alors de la forme $a(u, v) = auv$ (avec*

$a > 0$) et $L(v) = fv$ (toujours avec f fixée).
On a alors $u = \frac{f}{a}$ et le problème 1.10 est celui de la minimisation d'une parabole. En effet, $\frac{1}{2}a(v,v) - L(v) = \frac{av^2}{2} - fv$. ■

1.3.3 Approximation dans un espace de dimension finie

Le Théorème 1.3.1 permet d'affirmer l'existence et l'unicité de la solution d'un grand nombre d'équations différentielles aux dérivées partielles ; cependant, les solutions sont généralement impossibles à donner analytiquement. Comment rechercher des solutions approchées ? Comment évaluer l'erreur commise lors de l'approximation ? L'idée que nous allons suivre dans ce sous-paragraphe est de considérer un sous-espace $(E_h, < \cdot, \cdot >)$ de dimension finie (h est un paramètre dont l'interprétation apparaîtra ultérieurement). Dans un premier temps, nous allons voir que résoudre le problème «pour $f \in E$ donnée, trouver $u \in E$ tel que $\forall v \in E$, $a(u,v) =< f, v >$ »dans un sous-espace $E_h \subset E$ de dimension finie se ramène à l'inversion d'une matrice, ce qui fera le lien avec le chapitre 2 de ce cours. Dans un deuxième temps, nous verrons pourquoi la solution $u_h \in E_h$ tend vers la vraie solution $u \in E$ lorsque la dimension de E_h tend vers l'infini quand h évolue.

Notons $(\mathcal{P})$ le problème
$(\mathcal{P})$: pour $f \in E$ donnée, trouver $u \in E$ tel que $\forall v \in E$, $a(u,v) =< f, v >$
et $(\mathcal{P}_h)$ le même problème considéré dans un sous-espace E_h de dimension fini n
$(\mathcal{P}_h)$: pour $f_h \in E_h$ donnée, trouver $u_h \in E_h$ tel que $\forall v_h \in E_h$, $a(u_h, v_h) =< f_h, v_h >$
Montrons que résoudre $(\mathcal{P}_h)$ revient à inverser une matrice.

Le Théorème 1.3.1 s'appliquant en dimension finie, nous savons que $(\mathcal{P}_h)$ admet une solution unique. On peut démontrer ce résultat sans utiliser le lemme, ce qui nous permettra de voir que la résolution de $(\mathcal{P}_h)$ se ramène à la résolution d'un système linéaire.

Considérons $(b_1, \ldots, b_n)$ une base de E_h. Tout élément v_h de E_h peut être décomposé dans cette base : $v_h = \sum_{i=1}^{n} v_h^i b_i$. L'assertion «$\forall v \in E_h$, $a(u_h, v_h) =< f_h, v_h >$ » est alors équivalente à l'assertion «$a(u_h, b_i) =< f_h, b_i >$ pour tout $1 \leq i \leq n$ ». En notant $u_h = \sum_{i=1}^{n} u_h^i b_i$ et $f_h = \sum_{i=1}^{n} f_h^i b_i$, et en utilisant la bilinéarité de $a(\cdot,\cdot)$, ces dernières équations s'écrivent $\sum_{j=1}^{n} u_h^j a(b_j, b_i) =< f_h, b_i >= \sum_{j=1}^{n} f_h^j < b_j, b_i >$ pour tout $1 \leq i \leq n$. On

reconnaît alors un système linéaire qui peut s'écrire sous forme matricielle :

$$\begin{bmatrix} u_h^1 & \dots & u_h^n \end{bmatrix} \underbrace{\begin{bmatrix} a(b_1,b_1) & \dots & a(b_1,b_n) \\ \vdots & & \vdots \\ a(b_n,b_1) & \dots & a(b_n,b_n) \end{bmatrix}}_{A} = \begin{bmatrix} < f_h, b_1 > & \dots & < f_h, b_n > \end{bmatrix}$$

$$= \begin{bmatrix} f_h^1 & \dots & f_h^n \end{bmatrix} \underbrace{\begin{bmatrix} < b_1, b_1 > & \dots & < b_1, b_n > \\ \vdots & & \vdots \\ < b_n, b_1 > & \dots & < b_n, b_n > \end{bmatrix}}_{B} \tag{1.11}$$

A est la matrice associée à la forme bilinéaire a dans la base $(b_1, \dots, b_n) : A = [a_{ij}]_{1 \leq i,j \leq n}$ avec $a_{ij} = a(b_i, b_j)$. Pour montrer que le système a une solution unique, il reste à vérifier que la matrice A est inversible.

Preuve : Soient (ξ_i) tels que $\forall i, 1 \leq i \leq n$, $\sum_{j=1}^{n} \xi^j a(b_j, b_i) = 0$. Montrons que cela implique que $\forall i, 1 \leq i \leq n, \xi_i = 0$.

- d'après l'hypothèse de coercitivité (1.9), on a $\alpha \left\| \sum_{j=1}^{n} \xi^j b_j \right\|^2 \leq a\left(\sum_{j=1}^{n} \xi^j b_j, \sum_{i=1}^{n} \xi^i b_i \right)$.
- or $a\left(\sum_{j=1}^{n} \xi^j b_j, \sum_{i=1}^{n} \xi^i b_i \right) = \sum_{j=1}^{n} \sum_{i=1}^{n} \xi^j \xi^i a(b_j, b_i) = 0$.
- d'où $\sum_{j=1}^{n} \xi^j b_j = 0$ et donc $\forall i, 1 \leq i \leq n, \xi_i = 0$ car les b_j forment une base de E_h.
- donc A est inversible car les vecteurs qui la composent sont libres.

■

On doit alors décider de la dimension n d'une part, et du choix de la base $(b_1, \dots, b_n)$ d'autre part. n doit être «suffisamment grand »pour que la solution approchée soit proche de la vraie solution ; néanmoins, il doit être «suffisamment petit » pour que le système 1.11 admette des solutions stables et calculables dans un temps raisonnable.

1.3.4 Convergence de la solution approchée

On cherche à répondre à la question : est-ce que u_h tend vers u quand E_h grossit ? Nous avons le résultat suivant :

Théorème 1.3.2 *Soit $(E, <\cdot,\cdot>)$ un espace de Hilbert, a une forme bilinéaire, continue et cœrcitive, $f \in E$ et $u \in E$ solution du problème $(\mathcal{P})$. Soit E_h un sous-espace de $(E, <\cdot,\cdot>)$ de dimension finie, et $f_h \in E_h$ le projeté de f sur E_h. Soit $u_h \in E_h$ solution du porblème $(\mathcal{P}_h)$ (associé à f_h) dans E_h. Alors il existe une constante $C > 0$ indépendante de l'espace E_h telle que*

$$\|u - u_h\| \leq C \inf_{v_h \in E_h} \|u - v_h\|. \tag{1.12}$$

Preuve : Soient E, E_h, $f \in E$, $f_h \in E_h$, $u \in E$ et $u_h \in E_h$ comme dans l'énoncé.

- Soit v_h un élément quelconque de E_h et posons $w_h = v_h - u_h$.
- $w_h \in E_h$ donc on a $a(u_h, w_h) =< f_h, w_h >$ (a)
- $w_h \in E$ donc on a $a(u, w_h) =< f, w_h >$ Par ailleurs, $f_h \in E_h$ étant le projeté de f sur E_h, on a $< f_h, w_h >=< f, w_h >$ et il en résulte $a(u, w_h) =< f_h, w_h >$ (b)
- En soustrayant (b) de (a) on arrive à $a(u - u_h, w_h) = 0 = a(u - u_h, v_h - u_h)$
- En soustrayant $a(u-u_h, v_h-u_h)$ de $a(u-u_h, u-u_h)$ et en utilisant la linéarité par rapport à la deuxième variable on obtient : $a(u - u_h, u - u_h) = a(u - u_h, u - v_h)$
- La coercitivité de $a(\cdot,\cdot)$ nous dit que $\alpha\|u - u_h\|^2 \leq a(u - u_h, u - u_h)$
- La continuité de $a(\cdot,\cdot)$ nous dit que $a(u - u_h, u - v_h) \leq k\|u - u_h\|\|u - v_h\|$
- On en déduit $\alpha\|u - u_h\|^2 \leq k\|u - u_h\|\|u - v_h\|$
- En simplifiant par $\|u - u_h\|$, on a $\alpha\|u - u_h\| \leq k\|u - v_h\|$, d'où le résultat du théorème avec $C = \frac{k}{\alpha}$.

■

Ainsi pour montrer que la solution approchée u_h tend vers la vraie solution u lorsque h évolue (étant donné les E_h utilisés plus loin, nous verrons que h sera un paramètre qui tendra vers 0), il suffit de choisir une famille d'espaces (E_h) telle que tout élément E puisse être approché aussi près que l'on veut, en choisissant convenablement h, par un élément d'un espace E_h. Quelques précisions sur la convergence seront abordées en exercice.

1.3.5 Synthèse

En résumé :

- nous avons mis notre problème sous la forme d'un problème variationnel équivalent ;
- cela nous a permis d'assurer l'existence et l'unicité de la solution ;
- nous avons proposé une technique générale d'approximation de la solution dans un espace de dimension finie ;
- nous avons étudié l'erreur commise par cette approximation.

Il reste à préciser la technique d'approximation en choisissant l'espace de dimension fini et une base de cet espace. Une première idée venant à l'esprit est de considérer des espaces de Hilbert admettant une base orthonormée. Supposons que$(b_1, \ldots, b_n, \ldots)$ est une base orthonormée de E ; pour tout $v \in E$, on a alors $v = \sum_{i=1}^{+\infty} \alpha_i b_i$ avec $\alpha_i =< v, b_i >$. En prenant pour E_h (ici $h = \frac{1}{n}$) l'espace engendré par les n premiers éléments de la base $(b_1, \ldots, b_n)$, on constate que l'on peut approcher $v \in E$ par $\sum_{i=1}^{n} \alpha_i b_i$ appartenant à E_h. On a alors $\|v - v_h\|^2 = \left\| \sum_{i=n+1}^{+\infty} \alpha_i b_i \right\|^2 = \sum_{i=n+1}^{+\infty} \alpha_i^2$ qui tend vers 0 lorsque n tend vers l'infini. Une solution possible serait alors de choisir une famille libre quelconque (formée des polynômes par exemple), d'en déduire une base orthonormée (par l'orthogonalisation de Schmidt par exemple), et rechercher la solution approchée dans E_h l'espace engendré par les n premiers vecteurs de la base.

Il existe ainsi beaucoup de solutions approchées «théoriques» et le problème est plutôt de choisir celles qui sont les plus intéressantes au plan algorithmique (bon compromis entre rapidité des calculs et qualité de l'approximation).

La méthode des éléments finis, qui fait l'objet du paragraphe suivant, présente de bonnes qualités sur ce plan et est fréquemment utilisée dans la pratique.

1.4 Interpolation de Lagrange

Le principe général de la méthode des éléments finis consiste à découper le domaine Ω de définition de l'équation différentielle en petits sous-domaines finis sur lesquels on cherche une approximation de la solution par des polynômes.Il va donc falloir réaliser une «triangulation» de Ω, puis choisir un espace E_h dont les fonctions sont polynômiales par morceaux et enfin construire une base de E_h dont les éléments ont des «petits» supports, propriété qui entraîne une structure de matrice-bande pour le système linéaire associé.

1.4.1 Éléments finis de Lagrange

Dans toute la suite, on considère :

1. K une partie compacte de $\mathbb{R}^2$, connexe et d'intérieur non vide
2. $\Sigma = \{s_j\}_{j=1}^{N}$ un ensemble fini de N points distincts de K
3. P un espace vectoriel de dimension finie et composé de fonctions définies sur K à valeurs réelles

Définition 1.4.1 (Unisolvance) *Σ est P-unisolvant si et seulement si étant donné N réels quelconques $(\alpha_1, \ldots, \alpha_N)$ il existe une unique fonction p de P telle que $p(s_j) = \alpha_j$ pour $1 \leq j \leq N$.*

Définition 1.4.2 (Éléments finis de Lagrange) *Le triplet (K, P, Σ) est dit élément fini de Lagrange si Σ est P-unisolvant.*

Définition 1.4.3 (Fonctions de forme (ou fonctions de base)) *Étant donné un élément fini de Lagrange (K, P, Σ), on appelle fonctions de forme les N fonctions notées $p_1, \ldots, p_N$ vérifiant $p_i(s_j) = \delta_{ij} = \begin{cases} 0 \text{ si } i \neq j \\ 1 \text{ si } i = j \end{cases}$*

Remarques pratiques. Pour montrer qu'un triplet (K, P, Σ) est un élément fini de Lagrange, nous avons trois méthodes à notre disposition.

Première méthode : vérifier que $\dim(P) = \mathrm{Card}(\Sigma) = N$ et que la seule fonction de P qui s'annule sur Σ est la fonction identiquement nulle (*i.e.* résoudre le système d'équations $p(s_j) = 0$ pour $1 \leq j \leq N$) ;

Deuxième méthode : vérifier que $\dim(P) = \mathrm{Card}(\Sigma) = N$ et exhiber toutes les fonctions de base p_i ;

Troisième méthode : appliquer directement la définition de l'unisolvance et résoudre le système d'équations $p(s_j) = \alpha_j$ pour $1 \leq j \leq N$.

Les deux premières méthodes sont justifiées d'une part, par le fait que la condition $\dim(P) = \mathrm{Card}(\Sigma)$ est une condition nécessaire pour que Σ soit P-unisolvant et d'autre part, par l'étude de l'application $\mathcal{L} : P \to \mathbb{R}^N$ définie par $\mathcal{L}(p) = (p(s_j))_{j=1}^N$: dans le premier cas, on montre que $\mathcal{L}$ est injective et dans le second qu'elle est surjective ; ce qui implique dans les deux cas que $\mathcal{L}$ est bijective car P est de dimension N.

Avant d'aller plus loin précisons la philosophie générale de la recherche d'une approximation de la solution d'une équation aux dérivées partielles. Le domaine sur lequel on recherche une solution approchée est appelé à être divisé en «éléments finis élémentaires». Ensuite, on définira des fonctions formant une base de l'espace de dimension finie dans lequel on recherche la solution approchée. Conformément à ce qui précède, la solution approchée sera obtenue en résolvant un système linéaire. Dans le cas simple considéré ici, tous les éléments finis formant une «triangulation» du domaine sont tous d'une même forme, obtenus à partir d'un élément fini de référence.

Définition 1.4.4 (Éléments finis équivalents) *Deux éléments finis de Lagrange (K, P, Σ) et $(\hat{K}, \hat{P}, \hat{\Sigma})$ sont dit équivalents s'il existe une bijection continue F de $\hat{K}$ sur K vérifiant*

- *$K = F(\hat{K})$ et $\Sigma = F(\hat{\Sigma})$*
- *$P = \{p : K \to \mathbb{R};\ p \circ F \in \hat{P}\}$*

De plus, si F est affine inversible, les éléments finis sont dits affine-équivalents.

Dans un cas simple où la «triangulation» ne contient que des éléments finis équivalents, on peut se restreindre à étudier uniquement l'élément de référence $(\hat{K}, \hat{P}, \hat{\Sigma})$.

Exemple 1.4.1 *Soit $\hat{K}$ le triangle de référence de sommets $s_1 = (0,0)$, $s_2 = (0,1)$ et $s_3 = (1,0)$. Notons s_0 son centre de gravité et s_{12}, s_{13} et s_{23} les milieux, respectivement, des segments $[s_1s_2]$, $[s_1s_3]$ et $[s_2s_3]$. Par ailleurs, notons P_k l'ensemble des polynômes (de deux variables) de degré inférieur ou égal à k. Ainsi $f \in P_k$ si et seulement si $f(x,y) = \sum_{0 \leq i+j \leq k} \alpha_{ij} x^i y^j$ avec i et j des entiers naturels.*

Voici par exemple trois types possibles d'éléments finis.

- *Triangle de type (0) : $\hat{T}_0 = (\hat{K}, \hat{P} = P_0, \hat{\Sigma} = \{s_0\})$*
- *Triangle de type (1) : $\hat{T}_1 = (\hat{K}, \hat{P} = P_1, \hat{\Sigma} = \{s_1, s_2, s_3\})$*
- *Triangle de type (2) : $\hat{T}_2 = (\hat{K}, \hat{P} = P_2, \hat{\Sigma} = \{s_1, s_2, s_3, s_{12}, s_{13}, s_{23}\})$*

Considérons maintenant un triangle quelconque T^b dans $\mathbb{R}^2$ de sommets (a_1, a_2, a_3). On montre qu'il existe une transformation affine unique F telle que $a_1 = F(s_1)$, $a_2 = F(s_2)$ et $a_3 = F(s_3)$. On peut alors considérer des éléments finis T_0^b, T_1^b ou T_2^b affine-équivalents respectivement à $\hat{T}_0$, $\hat{T}_1$ et $\hat{T}_2$. ■

1.4.2 Solutions approchées

L'idée générale de la méthode des éléments finis consiste à faire une triangulation du domaine et à considérer les fonctions continues telles que leur restrictions à chaque triangle K faisant partie de la triangulation soit un élément de P. L'ensemble de telles fonctions est un espace vectoriel de dimension finie ; on peut alors envisager la solution approchée dans un tel espace. Comme nous le verrons plus loin, il existe en plus des théorèmes de convergence, permettant d'évaluer la différence entre les solutions approchées et la vraie solution, lorsque la triangulation devient «de plus en plus fine».

Triangulation de Ω

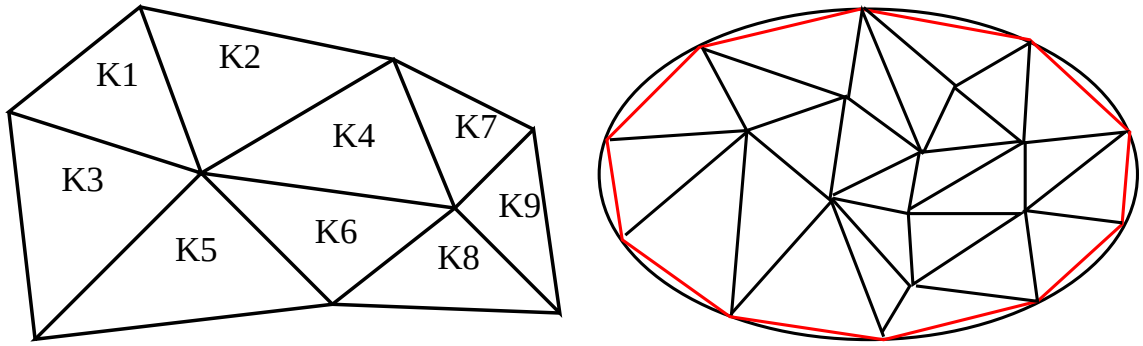
La triangulation de Ω consiste à partitionner Ω en petits domaines. Il est possible de donner une définition très précise d'une triangulation d'un domaine polygonal. Dans le cas où Ω ne serait pas polygonal, il semble naturel alors d'approcher Ω par un ouvert polygonal.

Définition 1.4.5 (Triangulation) *Soit Ω un ouvert polygonal de $\mathbb{R}^2$ et $\overline{\Omega}$ son adhérence. On considère une décomposition Φ_h finie du domaine telle que :*

- *$\overline{\Omega} = \bigcup_{K \in \Phi_h} K$;*
- *chaque élément K de Φ_h est un triangle de $\mathbb{R}^2$;*
- *les intérieurs de deux triangles de Φ_h distincts sont disjoints ;*
- *toute côté d'un triangle est soit un côté d'un autre triangle (auquel cas les triangles sont dits adjacents), soit une partie de la frontière Γ de Ω.*

Alors Φ_h est appelée triangulation de Ω.

Deux exemples de triangulation sont présentés sur la figure 1.1.



(a) (b)

FIGURE 1.1 – Triangulation (a) d'un ouvert polygonal en neuf triangles élémentaires et (b) d'un ouvert quelconque

Se donner une triangulation et définir des éléments finis sur les sous-domaines de cette triangulation permet d'introduire des espaces de dimension finie E_h dans lesquels il est possible de rechercher des solutions approchées de la formulation variationnelle de certaines équations différentielles.

Construction de l'espace des solutions

Considérons un ouvert polygonal Ω et Φ_h une triangulation. Choisissons l'un des éléments finis $\hat{T}_1$ ou $\hat{T}_2$ et supposons qu'à tout élément de Φ_h est associé un élément fini affinement équivalent à l'élément fini choisi. Pour fixer les idées, considérons $\hat{T}_1$. Considérons X_h le sous-ensemble de l'ensemble $\mathcal{C}^0(\overline{\Omega})$ des fonctions continues sur $\overline{\Omega}$, défini de la façon suivante :

$$X_h = \{v \in \mathcal{C}^0(\overline{\Omega}) : \forall K \in \Phi_h,\ v_{|K} \in P_K\} \tag{1.13}$$

où $v_{|K}$ désigne la restriction de v à K. De plus, dans le cas où le système d'équations différentielles contient une condition de nullité de la fonction sur les bords du domaine (comme dans notre cas particulier du problème de Dirichlet, cf l'équation 1.5), il faudra ne considérer que les éléments de X_h nuls sur la frontière de Ω, soit l'ensemble $X'_h = \{v \in \mathcal{C}^0(\overline{\Omega}) : \forall K \in \Phi_h,\ v_{|K} \in P_K \text{ et } v_{|\Gamma} = 0\}$.

La solution approchée sera recherchée dans l'espace $E_h = X_h$ (ou X'_h) ; pour ce faire, il nous faut préciser pourquoi ce dernier est de dimension fini et choisir une base. Dans la mesure où le nombre de triangles dans la triangulation est fini, et où sur chaque triangle K la fonction $v_{|K}$ est dans un espace de dimension finie, l'espace des v définies sur $\overline{\Omega}$ telles que $v_{|K} \in P_K$ est nécessairement de dimension finie.

Construction d'une base de X_h

Pour définir une base, considérons $\Sigma_h = \{a_j\}_{1 \leq j \leq I}$ tous les points faisant partie des éléments finis formant la triangulation. Dans le cas où la condition 1.5 est vérifiée, il n'est pas nécessaire de prendre en compte les points se situant sur la frontière de Ω et Σ_h sera limité à l'ensemble des points intérieurs de la triangulation.

Considérons alors la famille, dont on montre aisément le caractère libre et génératrice, suivante :

$$\text{pour } 1 \leq i,j \leq I,\ \varphi_i(a_j) = \delta_{ij} = \begin{cases} 0 \text{ si } i \neq j \\ 1 \text{ si } i = j \end{cases} \tag{1.14}$$

On remarquera que les fonctions φ_i ont un petit support : c'est l'ensemble des K auxquels le point a_i appartient. Par ailleurs, il existe des liens entre les $(\varphi_i)_{1\leq i\leq I}$ et les fonctions de forme associées à chacun des K qui sont définies de manière analogue mais en se restreignant aux points de l'élément fini considéré. En effet, considérons un point a_i appartenant à un seul triangle K de la triangulation ; alors la restriction de φ_i au triangle K est égale à p_j, c'est-à-dire la fonction de forme de $(K, P, \Sigma = (s_j))$ qui s'annule en tous points de Σ sauf $s_j = a_i$, et sa restriction aux autres triangles est égale à la fonction nulle. Supposons que a_i appartienne à plusieurs triangles $K_1, \ldots, K_m$ de la triangulation. Alors la restriction de φ_i à chaque K_q est égale à la fonction p_j^q, qui est la fonction de forme de $(K_q, P, \Sigma_q = (s_j^q))$ liée à $s_j^q = a_i$ au sens de la définition 1.4.3.

Exemple 1.4.2 *Considérons notre problème de Dirichlet initial, l'ouvert polygonal Ω de $\mathbb{R}^2$, inclus dans le plan horizontal, et la triangulation Φ_h de la figure 1.4.2. Associons à chaque triangle un élément fini affine équivalent au Triangle de type (1) $\hat{T}_1 = (\hat{K}, \hat{P} = P_1, \hat{\Sigma} = \{s_1, s_2, s_3\})$ (voir l'exemple 1.4.1). Φ_h est constituée de dix triangles $K_1, \ldots, K_{10}$ (tous situés sur le même plan horizontal). L'ensemble Σ_h est alors constitué des sommets des triangles qui ne se trouvent pas sur la frontière de Ω. Or, il n'y a que trois sommets ne se trouvant pas sur la frontière et on les note a1, a2, et a3. Donc $\Sigma_h = \{a_1, a_2, a_3\}$ et il y a trois fonctions φ_1, φ_2 et φ_3 formant une base de l'espace E_h.*

Intéressons-nous au point a_1 et à sa fonction associée φ_1. De par la définition 1.14, le support de φ_1 correspond aux triangles auxquels le point a_1 appartient, c'est-à-dire l'union de K_1, K_2, K_5 et K_6. Étant donné que pour chacun de ces triangles les fonctions p_j sont des polynômes de $\mathbb{R}^2$ de degré 1 de la forme $p_j(x,y) = d_1^j x + d_2^j y + d_3^j$, la fonction φ_1 a une forme de pyramide. Ce raisonnement est évidemment valable pour tous les autres points de Σ_h : le graphe représentatif de toutes les fonctions φ_i sera une pyramide.

On peut déterminer plus précisement la fonction φ_1 avec les fonctions de forme des triangles de son support. On introduit les points a_4 et a_5 comme dans la figure 1.4.2 et on pose l'ordre des sommets dans les triangles : $K1 = \{a_1, a_2, a_4\}$, $K2 = \{a_3, a_1, a_4\}$, $K5 = \{a_5, a_2, a_1\}$ et $K6 = \{a_3, a_5, a_1\}$.

La définition 1.4.3 des fonctions de forme implique que pour le triangle K_1 :

- *p_1^1 est nul en a_2 et a_4 et égal à 1 en a_1,*
- *p_2^1 est nul en a_1 et a_4 et égal à 1 en a_2,*
- *p_3^1 est nul en a_1 et a_2 et égal à 1 en a_4.*

On a les mêmes conclusions pour les fonctions de forme des autres triangles et le lien entre les (φ_i) et les (p_j^q) permet en particulier de construire φ_1 de la manière suivante :

- *la restriction de φ_1 à K_1 est la fonction p_1^1 de K_1,*

— *la restriction de* φ_1 *à* K_2 *est la fonction* p_2^2 *de* K_2,

— *la restriction de* φ_1 *à* K_5 *est la fonction* p_3^5 *de* K_5,

— *la restriction de* φ_1 *à* K_6 *est la fonction* p_3^6 *de* K_6.

■

Conséquences sur le système linéaire à résoudre

Nous avons donc défini un espace de dimension finie et une base $(\varphi_i)_{1\leq i\leq I}$. Rappelons que l'on recherche un vecteur u_h de X_h tel que pour tout v_h de X_h, $a(u_h, v_h) = < f_h, v_h >$ et que cela peut se ramener à résoudre le système linéaire 1.11 avec $A = [a(\varphi_i, \varphi_j)]_{1\leq i,j\leq I}$ et $B = [< \varphi_i, \varphi_j >]_{1\leq i,j\leq I}$.

En résumé, on doit alors calculer les éléments de A et B. Ensuite, on résout le système linéaire 1.11, ce qui nous fournit une solution approchée. Dans notre problème de Dirichlet initial, on a

$$a(\varphi_i, \varphi_j) = \int_\Omega \left(\frac{\partial \varphi_i(x,y)}{\partial x} \frac{\partial \varphi_j(x,y)}{\partial x} + \frac{\partial \varphi_i(x,y)}{\partial y} \frac{\partial \varphi_j(x,y)}{\partial y} \right) dxdy \quad (1.15)$$

$$< \varphi_i, \varphi_j > = \int_\Omega \varphi_i(x,y)\varphi_j(x,y) dxdy \quad (1.16)$$

Nous ne pousserons pas plus avant les calculs 1.15 et 1.16 ; notons cependant que les fonctions à intégrer figurant dans 1.15 et 1.16 sont des polynômes par morceaux. Il en résulte que la recherche des intégrales ne pose aucun problème de calcul.
Par ailleurs, on remarquera que la matrice A est symétrique et qu'elle sera plutôt creuse, c'est-à-dire qu'elle contiendra beaucoup de 0, du fait que les fonctions φ_i ont des petits supports. En fait, les seuls termes $a(\varphi_i, \varphi_j)$ non nuls seront ceux pour lesquels l'intersection des supports de φ_i et φ_j est non vide. Enfin si l'on numérote convenablement les points de Σ_h, A possèdera une structure bande (par exemple tridiagonale par blocs).

1.4.3 Analyse de la méthode et convergence des solutions approchées

Rappelons que l'on cherche une approximation d'une fonction continue u. D'ailleurs, X_h ne contient par définition que des fonctions continues. Une première question se pose alors : est-on certain que les fonctions φ_i définies par l'équation 1.14 sont continues (en particulier au niveau des côtés des triangles) et forment donc bien une base de X_h ? Pour assurer qu'une triangulation donnée aboutisse réellement à une base, il faut vérifier quelques hypothèses de compatibilité.

Proposition 1.4.1 *Les hypothèses de compatibilité sont les suivantes :*

1. *Pour tout couple d'éléments finis $(K_1, P_{K_1}, \Sigma_{K_1})$ et $(K_2, P_{K_2}, \Sigma_{K_2})$ adjacents de côté commun $K' = K_1 \cap K_2$. On doit avoir*
   - $\Sigma_{K_1} \cap K' = \Sigma_{K_2} \cap K'$,
   - $P_{K_1|K'} = P_{K_2|K'}$.
2. *Pour tout côté K' d'un élément fini (K, P, Σ), l'ensemble des points du côté $\Sigma' = \Sigma \cap K'$ est unisolvant par rapport aux restrictions des fonctions de forme de K à ce même côté (Σ' et P'-unisolvant avec $P' = \{p_{|K'},\ p \in P\}$).*

Sous ces hypothèses, il existe un théorème qui assure que X_h est l'image de $\mathcal{C}^0(\overline{\Omega})$ par une fonction Π_h continue sur $\overline{\Omega}$, construite à partir des opérateurs d'interpolation sur tous les triangles K de la triangulation.

Définition 1.4.6 (Opérateur de P-interpolation) *Soit (K, P, Σ) un élément fini de Lagrange (avec $\Sigma = \{s_j\}_{j=1}^N$). On appelle opérateur de P-interpolation de Lagrange sur Σ l'opérateur qui à toute fonction v, définie sur K, associe la fonction $\Pi_K v = \sum_{i=1}^{N} v(s_i) p_i$, où les p_i sont les fonctions de forme de l'élément fini.*
$\Pi_K v$ est dit le P-interpolé de v.

Grâce à cette définition, nous savons que pour chaque triangle K, auquel l'on a associé l'élément fini (K, P_K, Σ_K), il existe un opérateur de P_K-interpolation Π_K. L'opérateur Π_h, défini sur l'ensemble $\mathcal{C}^0(\overline{\Omega})$ des fonctions continues sur $\overline{\Omega}$, est construit de la façon suivante : sur les points appartenant à l'intérieur d'un élément de la triangulation, on pose

$$\forall K \in \Phi_h,\ \forall x \in \mathrm{Int}(K),\ \Pi_h v(x) = \Pi_K v(x). \tag{1.17}$$

Les hypothèses de compatibilités permettent d'étendre l'égalité 1.17 aux points se trouvant sur les côtés des triangles formant la triangulation, et donc à tout point de $\overline{\Omega}$. De plus, étant donnée 1.14, on vérifie que $\Pi_h v$ peut également s'écrire $\Pi_h v = \sum_{i=1}^{N} v(a_i)\varphi_i$ et donc pour tout v de $\mathcal{C}^0(\overline{\Omega})$, $\Pi_h v$ est un élément de X_h. Cette dernière remarque sera utile pour l'étude de la convergence des solutions approchées.

On rappelle que la méthode des éléments finis sera convergente si les solutions approchées u_h tendent vers la solution vraie u quand X_h grossit, c'est-à-dire lorsque l'on augmente le nombre de points dans Σ_h. Selon le théorème 1.3.2, on doit chercher à majorer la distance de tout élément de $\mathcal{C}^0(\overline{\Omega})$ à X_h. Nous allons voir comment cette distance tend vers 0 lorque le paramètre h évolue (en fait, il aura une signification géométrique liée à la forme des triangles et il va tendre vers 0).

Avant de poursuivre, introduisons, pour chaque triangle K de $\mathbb{R}^2$ deux paramètres :

- h_K appelé diamètre de K correspondant au maximum des distances euclidiennes entre deux points de K

— ρ_K appelé rondeur de K correspondant au rayon maximum d'un cercle inscrit dans K.

Par ailleurs, considérons un deuxième produit scalaire défini sur l'ensemble des fonctions dont les dérivées partielles appartiennent à $L^2(\Omega)$ (ici $\Omega \subset \mathbb{R}^2$) :

$$< f, g >_1 = \int_\Omega \left[f(x,y)g(x,y) + \frac{\partial f}{\partial x}(x,y)\frac{\partial g}{\partial x}(x,y) + \frac{\partial f}{\partial y}(x,y)\frac{\partial g}{\partial y}(x,y) \right] dxdy$$

et la norme associée

$$\|f\|_1 = \sqrt{\int_\Omega \left[f^2(x,y) + \left(\frac{\partial f}{\partial x}(x,y)\right)^2 + \left(\frac{\partial f}{\partial y}(x,y)\right)^2 \right] dxdy} \tag{1.18}$$

On note immédiatement que cette nouvelle norme majore la norme de $L^2(\Omega)$ (qui est $\|f\| = \sqrt{\int_\Omega f^2(x,y)}$), ce qui signifie que la convergence au sens de la norme (1.18) implique la convergence dans $L^2(\Omega)$.

Nous avons alors le résultat suivant :

Théorème 1.4.1 *Soit Ω un ouvert polygonal de $\mathbb{R}^2$ et Φ_h une famille des triangulations de Ω telles que leurs éléments soient affine-équivalents et vérifient les conditions de compatibilité.*
On suppose de plus deux conditions sur la géométrie des éléments :

— $h = \max_{K \in \Phi_h} h_K \to 0$;

— *il existe $\sigma \geq 1$ telle que $\forall h$, $\forall K \in \phi_h$, $\frac{h_K}{\rho_K} \leq \sigma$.*

Alors la solution u_h du problème $(\mathcal{P}_h)$ dans l'espace X_h converge au sens de la norme (1.18) (donc également dans $L^2(\Omega)$) vers la vraie solution u.
De plus, il existe une constante C telle que $\|u - u_h\|_1 \leq Ch$ si on a choisi des triangles de type (1), et $\|u - u_h\|_1 \leq Ch^2$ si on a choisi des triangles de type (2).

Preuve : La démonstration complète et rigoureuse dépasse largement le cadre de ce cours. Mais nous en donnons tout de même quelques éléments.

— Le théorème 1.3.2 et la remarque sur l'opérateur d'interpolation Π_h permettent de dire que $\|u - u_h\|_1 \leq C_1 \|u - \Pi_h u\|_1$.

— Par définition de Π_h et de Φ_h, on a $\|u - \Pi_h u\|_1 = \sqrt{\sum_{K \in \Phi_h} \|u - \Pi_K u\|_1^2}$.

— Un résultat général sur l'erreur d'interpolation permet d'obtenir la majoration suivante : il existe deux constantes C_2 et C_3, ne dépendant que de l'élément fini de référence et une constante C_4 ne dépendant que de u, telles que $\|u - \Pi_K u\|_1 \leq C_4 \frac{h_K}{\rho_K} \sqrt{C_2^2 + C_3^2 \rho_K^2} h_K^k$ où k est le numéro du type d'élément fini ($k = 1$ si ce sont des triangles de type (1) par exemple).

— En utilisant les conditions du théorème, on a : $\|u - \Pi_h u\|_1 \leq C_5 h^k$, C_5 dépendant de σ, du diamètre de $\overline{\Omega}$ et de u. D'où $\lim_{h\to 0} \|u - \Pi_h u\|_1 = 0$.

— En appliquant le théorème démontré en travaux dirigés avec $r_h = \Pi_h$, on obtient la convergence de la méthode.

■

1.5 Conclusion

La démarche générale que nous avons suivi pour résoudre notre équation différentielle partielle du type Dirichlet, se résume de la façon suivante.

1. Mettre l'équation différentielle partielle sous la forme d'un problème variationnel équivalent : $\forall v \in E,\ a(u,v) =< f, v >$ (avec u l'inconnue et f donnée).
2. Approximer ce dernier par le problème approché :

$$\forall v_h \in E_h,\ a(u_h, v_h) =< f_h, v_h > .$$

3. Dans le cadre des éléments finis de Lagrange, construire E_h de dimension finie n et sa base $(\varphi_1, \ldots, \varphi_n)$ à l'aide d'une triangulation Φ_h et d'une base de polynômes sur chaque éléments de Φ_h.
4. Résoudre le problème approché en trouvant la solution du système linéaire :
   pour $1 \leq i \leq n$, $\sum_{j=1}^{n} u_h^j a(\varphi_j, \varphi_i) =< f_h, \varphi_i >$.

Remarque. Les résultats ci-dessus se généralisent à $\mathbb{R}^n$ pour n quelconque. On remplace alors les triangles par des n-simplexes. Les éléments finis peuvent aussi être de forme parallélépipédique ou prismatique.

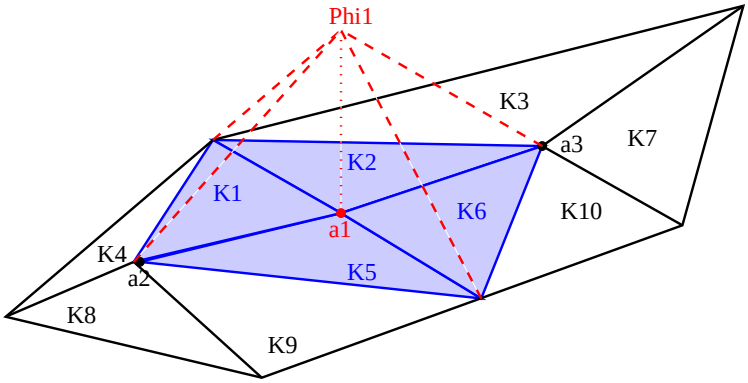


FIGURE 1.2 – Exemple de domaine Ω avec sa triangulation par des éléments finis affines-équivalents au Triangle de type (1). a_1 est un point de la triangulation et φ_1 (en pointillé) la fonction correspondante par la définition 1.14. φ_1 est non nulle sur les triangles K_1, K_2, K_5, K_6 et nulle sur tous les autres triangles.

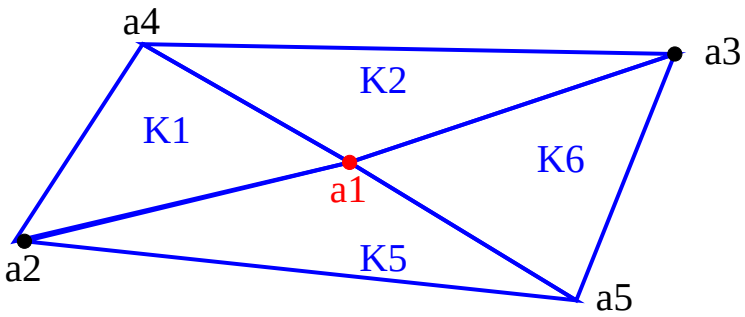


FIGURE 1.3 – Support de la fonction φ_1

Chapitre 2

Analyse Numérique Matricielle

Dans ce chapitre nous supposons que le problème que l'on étudie se ramène, au bout d'un certain nombre d'étapes : soit en la résolution d'un système linéaire $\mathbf{Ax} = \mathbf{b}$; soit en l'extraction d'éléments (valeurs et/ou vecteurs) propres d'une matrice donnée. Le but de ce chapitre consiste à proposer pour de tels problèmes des méthodes de résolution numérique et de les comparer entre elles.

2.1 Rappels et Compléments d'algèbre linéaire

Nous commençons par quelques résultats d'algèbre linéaire qui seront utiles dans les paragraphes 2.2, 2.3 et 2.4. Commençons par une remarque sur les notations utilisées dans ce chapitre : les lettres en caractères gras désignent des quantités vectorielles ou matricielles, selon qu'elles sont en minuscules ou en majuscules ($\mathbf{A}$, $\mathbf{u}$ et x désignent donc respectivement la matrice $\mathbf{A}$, le vecteur $\mathbf{u}$ et le nombre scalaire x). Les notations $\overline{z}$, $\mathbf{A}^H$ et $\mathbf{A}^T$ désignent respectivement le complexe conjugué de z, la matrice transposée conjuguée de $\mathbf{A}$ et la matrice transposée de $\mathbf{A}$.

2.1.1 Similarité et diagonalisation d'une matrice

Une même application linéaire de $\mathbb{C}^n$ dans $\mathbb{C}^n$ admet différentes représentations matricielles, chacune dépendant de la base choisie. Toutes ces matrices sont "similaires" ou "semblables" ; elles sont reliées les unes aux autres par la matrice de passage $\mathbf{P}$ de l'ancienne base dans la nouvelle base :

Définition 2.1.1 *Soit* $\mathbf{A}$ *et* $\mathbf{B}$ *deux matrices carrées d'ordre* n. $\mathbf{A}$ *et* $\mathbf{B}$ *sont* semblables *si et seulement si* $\exists\ \mathbf{P}$ *inversible, tel que* $\mathbf{B} = \mathbf{P}^{-1}\mathbf{AP}$.

Le problème de la réduction de matrices consiste, partant d'une représentation de l'application linéaire (c'est-à-dire d'une matrice $\mathbf{A}$), à trouver une base dans laquelle la représentation matricielle de cette même application linéaire soit la plus simple possible. La forme la plus intéressante est bien entendu la forme diagonale :

Définition 2.1.2 *Soit* $\mathbf{A}$ *une matrice carrée d'ordre* n. $\mathbf{A}$ *est* diagonalisable *si et seulement si* $\exists \mathbf{P}$ *inversible, tel que* $\mathbf{P}^{-1}\mathbf{AP} = diag.(\lambda_i)$.

Les nombres complexes λ_i (les valeurs propres de $\mathbf{A}$) sont également les racines du polynôme caractéristique dét$(\mathbf{A} - \lambda\mathbf{I})$. Bien entendu, toute matrice n'est pas diagonalisable. De façon générale, $\mathbf{A}$ est diagonalisable si et seulement s'il existe une base de vecteurs propres, ou encore si et seulement si $\mathbf{A}$ admet n vecteurs propres indépendants. Une caractérisation de la diagonalisabilité d'une matrice est également fournie par le résultat classique suivant :

Théorème 2.1.1 *Supposons que la matrice carrée* $\mathbf{A}$ *de dimension n admette m valeurs propres distinctes* $\{\lambda_i\}_{i=1}^m$*, chacune étant de* multiplicité algébrique m_i *(c'est-à-dire, dét(*$\mathbf{A} - \lambda\mathbf{I}$*) = cte* $\times \prod_{i=1}^m (\lambda - \lambda_i)^{m_i}$*, avec* $\sum_{i=1}^m m_i = n$*). Soit* $g_i = \dim Ker(\mathbf{A} - \lambda_i\mathbf{I})$ *la* multiplicité géométrique *de* λ_i*, c'est-à-dire la dimension de l'espace propre associé à* λ_i*. Alors*

- $g_i \leq m_i \ \forall i$ *; et*
- $\mathbf{A}$ *est diagonalisable si et seulement si* $g_i = m_i \ \forall i$.

Corollaire 2.1.1 $\mathbf{A}_{n\times n}$ *est diagonalisable si* $\mathbf{A}$ *admet n valeurs propres distinctes.* □

2.1.2 Équivalence et Décomposition en valeurs singulières

Définition 2.1.3 $\mathbf{A}_{q\times p}$ *et* $\mathbf{B}_{q\times p}$ *sont* équivalentes *si et seulement si* $\exists\ \mathbf{P}_{p\times p}$ *et* $\mathbf{Q}_{q\times q}$ *inversibles, telles que* $\mathbf{B}_{q\times p} = \mathbf{Q}_{q\times q}\mathbf{A}_{q\times p}\mathbf{P}_{p\times p}$.

La notion d'équivalence est bien sûr plus faible que celle de similarité puisque l'on n'exige, ni que $\mathbf{A}$ soit carrée, ni que $\mathbf{Q}$ soit égal à $\mathbf{P}^{-1}$ (la relation $\mathbf{B} = \mathbf{QAP}$ n'est donc pas un changement de base). Elle fournit une *relation d'équivalence* (- d'où la dénomination). Pour des raisons pratiques que nous verrons plus loin, on privilégie l'*équivalence unitaire* (c'est-à-dire que l'on exige que $\mathbf{P}$ et $\mathbf{Q}$ soient unitaires). La classe d'équivalence à laquelle appartient une matrice contient des représentants canoniques intéressants. En effet, si une matrice (carrée) n'est pas nécessairement semblable à une matrice diagonale, en revanche toute matrice $\mathbf{A}_{q\times p}$ est (unitairement) équivalente à une matrice diagonale :

Théorème 2.1.2 (Décomposition en valeurs singulières) *Soit* $\mathbf{A}$ *une matrice* $q \times p$*.* $\mathbf{A}$ *peut se factoriser en :* $\mathbf{A} = \mathbf{U\Sigma V}^H$*, où* $\mathbf{U}$ *(de dim.* $q \times q$*) et* $\mathbf{V}$ *(de dim.* $p \times p$*) sont des matrices unitaires,* $\mathbf{\Sigma}_{q\times p} = \begin{bmatrix} \mathbf{\Delta}_{r\times r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$*,* $\mathbf{\Delta} = \mathrm{diag}(\sigma_1, \cdots, \sigma_r)$*, et r (avec* $r \leq \min(p, q)$*) est le rang de* $\mathbf{A}$*.* $\mathbf{U}$ *et* $\mathbf{V}$ *ne sont pas uniques. Les constantes* σ_i *(les* valeurs singulières *non nulles de* $\mathbf{A}$*) sont uniques, et vérifient* $\sigma_i \in \mathbb{R}^{+*}$*,* $\sigma_1 \geq \cdots \geq \sigma_r > 0$.

Il existe un lien entre les valeurs singulières d'une matrice $\mathbf{A}$ et les valeurs propres de $\mathbf{AA}^H$ (ou de $\mathbf{A}^H\mathbf{A}$) : comme $\mathbf{AA}^H = \mathbf{U\Sigma\Sigma}^T\mathbf{U}^H$ (resp. $\mathbf{A}^H\mathbf{A} = \mathbf{V\Sigma}^T\mathbf{\Sigma V}^H$), $\mathbf{AA}^H$ (resp. $\mathbf{A}^H\mathbf{A}$) est similaire à $\mathbf{\Sigma\Sigma}^T$ (resp. à $\mathbf{\Sigma}^T\mathbf{\Sigma}$). Par conséquent :

Théorème 2.1.3 *Les valeurs singulières non nulles de* $\mathbf{A}$ *sont les racines carrées des valeurs propres non nulles de* $\mathbf{AA}^H$ *ou de* $\mathbf{A}^H\mathbf{A}$.

2.1.3 Triangularisation unitaire de Schur et applications

On a vu qu'une matrice carrée n'était pas nécessairement similaire à une matrice diagonale. A défaut de pouvoir réduire $\mathbf{A}$ sous la forme diagonale, on peut se contenter de formes moins sympathiques mais encore très utiles. C'est ainsi que l'on dispose de deux résultats importants de similarité : le *théorème de Jordan* et le *théorème de Schur*. Nous n'aurons besoin ici que de ce deuxième résultat :

Théorème 2.1.4 (Triangularisation unitaire de Schur.) *Soit* $\mathbf{A}$ *une matrice carrée quelconque. Alors il existe* $\mathbf{U}$ *unitaire, telle que* $\mathbf{U}^H\mathbf{A}\mathbf{U} = \mathbf{T}$, $\mathbf{T}$ *étant une matrice triangulaire supérieure (c'est-à-dire,* $t_{i,j} = 0$ *si* $i > j$*), avec les valeurs propres de* $\mathbf{A}$ *sur la diagonale (ni* $\mathbf{U}$ *ni* $\mathbf{T}$ *ne sont uniques).*

Le théorème de Schur est un résultat important qui admet de nombreuses applications. En particulier, il permet par exemple (entre autres démonstrations possibles) de démontrer le théorème de Cayley-Hamilton, selon lequel toute matrice annule son polynôme caractéristique. Il permet également de montrer que toute matrice normale est unitairement diagonalisable, et que toute matrice est arbitrairement proche d'une matrice diagonalisable. Nous allons maintenant préciser ces deux derniers résultats.

Applications à la diagonalisation de matrices.

Définition 2.1.4 *Une matrice* $\mathbf{A}$ *est* normale *si et seulement si* $\mathbf{A}\mathbf{A}^H = \mathbf{A}^H\mathbf{A}$.

Les matrices unitaires, de même que les matrices hermitiennes, sont des cas particuliers importants des matrices normales. Mais il existe également des matrices normales qui ne sont ni unitaires, ni hermitiennes : par exemple la matrice $\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$.

Théorème 2.1.5 (Théorème spectral pour les matrices normales) *Toute matrice normale se diagonalise dans une base unitaire :* $\forall \mathbf{A}$ *t.q.* $\mathbf{A}\mathbf{A}^H = \mathbf{A}^H\mathbf{A}$, $\exists\, \mathbf{U}$ *unitaire et* $\mathbf{\Lambda}$ *diagonale, t.q.* $\mathbf{A} = \mathbf{U}^H\mathbf{\Lambda}\mathbf{U}$.

Preuve : $\mathbf{A}$ peut se factoriser en $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^H$ en vertu du théorème 2.1.4. Or $\mathbf{A}$ normale implique $\mathbf{T}$ normale. Mais une matrice triangulaire normale est nécessairement diagonale. ■

Le théorème spectral admet deux cas particuliers intéressants : toute matrice unitaire, de même que toute matrice hermitienne, est diagonalisable dans une base unitaire :

Corollaire 2.1.2 $\forall \mathbf{A}$ *hermitienne,* $\exists \mathbf{U}$ *unitaire et* $\mathbf{\Lambda}$ *diagonale réelle, t.q.* $\mathbf{A} = \mathbf{U}^H\mathbf{\Lambda}\mathbf{U}$. □

Corollaire 2.1.3 $\forall \mathbf{A}$ *symétrique réelle,* $\exists \mathbf{Q}$ *orthogonale et* $\mathbf{\Lambda}$ *diagonale réelle, t.q.* $\mathbf{A} = \mathbf{Q}^T\mathbf{\Lambda}\mathbf{Q}$. □

Corollaire 2.1.4 $\forall \mathbf{A}$ *unitaire,* $\exists \mathbf{U}$ *unitaire et* $\mathbf{\Lambda}$ *diagonale, dont les éléments diagonaux sont de module égal à 1, t.q.* $\mathbf{A} = \mathbf{U}^H\mathbf{\Lambda}\mathbf{U}$. □

Densité des matrices diagonalisables dans l'ensemble des matrices

Une matrice donnée n'est pas nécessairement diagonalisable ; en revanche, toute matrice est similaire à une matrice triangulaire supérieure, arbitrairement proche d'une matrice diagonale :

Théorème 2.1.6 *Soit* $\mathbf{A}$ *une matrice carrée quelconque.* $\forall \epsilon > 0$, $\exists\, \mathbf{P}(\epsilon)$ *t. q.*
- $\mathbf{P}(\epsilon)^{-1}\mathbf{A}\mathbf{P}(\epsilon) = \mathbf{T}(\epsilon) = [t_{i,j}(\epsilon)]_{i,j=1}^{n}$ *est triangulaire supérieure ;*
- $\forall i, j, 1 \leq i < j \leq n, |t_{i,j}(\epsilon)| < \epsilon$.

Un autre résultat complète le point de vue apporté par le théorème précédent. Il montre que l'ensemble des matrices diagonalisables est dense [1] dans l'ensemble des matrices carrées :

Théorème 2.1.7 *Soit* $\mathbf{A} = [a_{i,j}]_{i,j=1}^{n}$ *une matrice carrée quelconque.* $\forall \epsilon > 0$, $\exists \mathbf{A}(\epsilon) = [a_{i,j}(\epsilon)]_{i,j=1}^{n}$ *telle que*
- $\mathbf{A}(\epsilon)$ *a n v. p. distinctes, et est donc diagonalisable ;*
- $\sum_{i,j=1}^{n} |a_{i,j} - a_{i,j}(\epsilon)|^2 < \epsilon$.

2.1.4 Normes vectorielles et matricielles

Définitions

Dans ce paragraphe nous établissons un certain nombre de résultats sur les normes matricielles (qu'elles soient induites ou non par une norme vectorielle). Commençons par rappeler les définitions des normes vectorielles les plus utilisées, à savoir les normes-p, ainsi que la norme infinie :

Définition 2.1.5 *Soit* $\mathbf{x} = [x_1 \cdots x_n]^T \in \mathbb{C}^n$. $\forall p \in \mathbb{N}^*$, $\| \mathbf{x} \|_p \stackrel{def}{=} (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$ *est une norme (la* norme-p*) de* $\mathbf{x}$. *L'application qui à tout* $\mathbf{x}$ *associe* $\| \mathbf{x} \|_\infty \stackrel{def}{=} \max(|x_i|)$ *définit également une norme, appelée* norme infinie *de* $\mathbf{x}$.

Nous passons maintenant de la norme d'un vecteur à la norme d'une matrice. L'ensemble des matrices carrées de dimension n étant un espace vectoriel de dimension n^2, on pourrait "mesurer" une matrice en utilisant n'importe quelle norme vectorielle sur $\mathbb{C}^{n^2}$ (ou sur $\mathbb{R}^{n^2}$). Cependant, les matrices étant aussi naturellement dotées d'une multiplication, il est souvent d'intérêt de pouvoir relier la "taille" du produit $\mathbf{AB}$ aux tailles de $\mathbf{A}$ et de $\mathbf{B}$. Il est donc d'usage, dans la définition de la norme d'une matrice, de rajouter aux propriétés requises pour toute norme la propriété (2.1) de sous-multiplicativité :

Définition 2.1.6 (Norme matricielle.) *Soit* $\mathcal{M}_n$ *l'ensemble des matrices carrées de dimension n. Une fonction* $\| \cdot \|$ *de* $\mathcal{M}_n$ *dans* $\mathbb{R}$ *est une norme matricielle si* $\| \mathbf{A} \| \geq 0$, $\| \mathbf{A} \| = 0$ *si et seulement si* $\mathbf{A} = \mathbf{0}$, $\| \alpha\mathbf{A} \| = |\alpha| \, \| \mathbf{A} \| \; \forall \alpha \in \mathbb{C}$, $\| \mathbf{A} + \mathbf{B} \| \leq \| \mathbf{A} \| + \| \mathbf{B} \|$, *et*

$$\| \mathbf{AB} \| \leq \| \mathbf{A} \| \| \mathbf{B} \| . \tag{2.1}$$

1. au sens de la norme euclidienne, cf. le paragraphe suivant.

Exemples classiques

— A toute norme vectorielle $\| \ . \ \|$ on peut associer la *norme subordonnée* (ou *induite*) par $\| \ . \ \|$ de l'opérateur linéaire représenté par la matrice [2]. On montre alors que les conditions rassemblées dans la définition 2.1.6 sont vérifiées, et donc que cette norme induite définit bien une norme matricielle :

Proposition 2.1.1 *Soit* $\|.\|$ *une norme vectorielle quelconque. La fonction* $|| \ . \ ||$ *de* $\mathcal{M}_n$ *dans* $\mathbb{R}$*, définie* $\forall \mathbf{A} \in \mathcal{M}_n$ *par* $|| \ \mathbf{A} \ || \stackrel{def}{=} \sup_{\mathbf{x} \neq \mathbf{0}} \frac{\| \ \mathbf{Ax} \ \|}{\| \ \mathbf{x} \ \|} = \sup_{\|\mathbf{x}\|=\mathbf{1}} \| \ \mathbf{Ax} \ \|$*, est une norme matricielle sur* $\mathcal{M}_n$*.*

En particulier, $\forall p \in \mathbb{N}^* \cup \{+\infty\}$, la norme$-p$ vectorielle $\|.\|_p$ induit donc une norme matricielle (qui sera également dénotée $\|.\|_p$). Il est utile d'observer que pour toute norme subordonnée, outre la propriété de sous-multiplicativité (2.1), on a par définition même :

$$\forall \mathbf{A} \in \mathcal{M}_n, \forall \mathbf{x} \in \mathbb{C}^n, \| \ \mathbf{Ax} \ \| \ \leq \ \| \ \mathbf{A} \ \| \| \ \mathbf{x} \ \| \ . \tag{2.2}$$

— Cependant toutes les normes matricielles ne sont pas subordonnées. La norme suivante, qui n'est subordonnée à aucune norme vectorielle, joue un rôle important :

Définition 2.1.7 (Norme euclidienne.) *Soit* $\mathbf{A} = [a_{i,j}]_{i,j=1}^n$ *une matrice carrée, de valeurs singulières non-nulles* $\{\sigma_i\}_{i=1}^p$*. La norme* euclidienne *(ou* de Frobenius*, ou* de Schur*) de* $\mathbf{A}$ *est la norme définie par :*

$$\| \ \mathbf{A} \ \|_E \stackrel{def}{=} (\sum_{i,j=1}^{n} |a_{i,j}|^2)^{\frac{1}{2}} = \sqrt{\sigma_1^2 + \cdots + \sigma_p^2}.$$

— $\| \ . \ \|_E$ n'est autre que la norme euclidienne de $\mathbf{A}$ (d'où son nom), *considérée comme un élément de* $\mathbb{C}^{n^2}$ (attention à ne pas confondre $\| \ . \ \|_E$ avec $\| \ . \ \|_2$!) ; c'est également une norme matricielle car (2.1) est vérifiée. Cependant toutes les "normes$-p$" de matrices $n \times n$, *considérées comme des vecteurs de* $\mathbb{C}^{n^2}$, ne définissent pas nécessairement des normes matricielles. C'est ainsi que l'application $\mathbf{A} \mapsto \sum_{i,j=1}^n |a_{i,j}|$ est une norme matricielle, mais pas $\mathbf{A} \mapsto \max_{1 \leq i,j \leq n} |a_{i,j}|$ (car (2.1) n'est pas vérifiée pour $\mathbf{A} = \mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$).

2. attention ! il est donc sous-entendu que l'on travaille dans une base donnée : deux matrices semblables n'ont en effet aucune raison d'avoir la même norme (même si elles représentent un même opérateur linéaire dans deux bases différentes).

Propriétés

Proposition 2.1.2 *Soit* $\mathbf{A} = [a_{i,j}]_{i,j=1}^{n}$ *une matrice de valeurs singulières non nulles* $\{\sigma_i\}_{i=1}^{p}$, $\sigma_1 \geq \cdots \geq \sigma_p > 0$. *Alors*

— $\| \mathbf{A} \|_1 = \max_j \sum_{i=1}^{n} |a_{i,j}|$,

— $\| \mathbf{A} \|_\infty = \max_i \sum_{j=1}^{n} |a_{i,j}|$,

— $\| \mathbf{A} \|_2 = \sigma_1$.

On voit donc que les normes 1 et ∞ d'une matrice peuvent être calculées très simplement à partir des éléments de cette matrice, ce qui n'est pas le cas de la norme 2. La norme 2 jouant toutefois un rôle particulier (cf. le paragraphe 2.2 sur le conditionnement), il est important de pouvoir estimer rapidement l'ordre de grandeur de la norme 2 d'une matrice. On peut alors utiliser par ex. le résultat suivant : $\| \mathbf{A} \|_2 \leq \| \mathbf{A} \|_E \leq \sqrt{n} \, \| \mathbf{A} \|_2$.

2.2 Conditionnement d'un problème d'algèbre linéaire

2.2.1 Conditionnement d'un système linéaire

Exemple introductif

Commençons par introduire, au travers d'un exemple, la notion du conditionnement d'un système linéaire $\mathbf{Ax} = \mathbf{b}$. Le vecteur $\mathbf{x} = [1111]^T$ est la solution exacte du système ci-dessous :

$$\underbrace{\begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}}_{\mathbf{b}}.$$

Cependant, une petite modification du membre de droite $\mathbf{b}$ entraîne une grande modification de la solution :

$$\begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \begin{bmatrix} 9,2 \\ -12,6 \\ 4,5 \\ -1,1 \end{bmatrix} = \begin{bmatrix} 32,1 \\ 22,9 \\ 33,1 \\ 30,9 \end{bmatrix}.$$

De même, une petite modification de la matrice $\mathbf{A}$ entraîne une solution très éloignée de la solution du système originel :

$$\begin{bmatrix} 10 & 7 & 8,1 & 7,2 \\ 7,08 & 5,04 & 6 & 5 \\ 8 & 5,98 & 9,89 & 9 \\ 6,99 & 4,99 & 9 & 9,98 \end{bmatrix} \begin{bmatrix} -81 \\ 137 \\ -34 \\ 22 \end{bmatrix} = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}.$$

On voit tout de suite les conséquences désastreuses entraînées par ce genre de problème : si on dispose de données bruitées, ou si, du fait de la précision finie des mots machine, les données sur lesquelles travaillera l'ordinateur sont (même légèrement) différentes des données exactes dont on dispose, alors le résultat obtenu numériquement peut être radicalement différent de la solution exacte du système, et on ne sait plus quelle confiance accorder au résultat affiché sur l'écran ...

Conditionnement

Comment expliquer le phénomène ci-dessus ? Constatons tout de suite sur l'exemple que le problème ne vient pas de ce que $\mathbf{A}$ serait "proche" d'une matrice singulière. En effet, son déterminant est égal à 1, et

$$\mathbf{A}^{-1} = \begin{bmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{bmatrix}.$$

Considérons d'abord le cas où seul le membre de droite a été perturbé. Pour mieux cerner le problème, appelons $\mathbf{x}$ la solution exacte du système originel, et $\mathbf{x} + \delta\mathbf{x}$ la solution exacte du système dont le membre de droite a été perturbé :

$$\begin{cases} \mathbf{A}\mathbf{x} & = & \mathbf{b} \\ \mathbf{A}(\mathbf{x} + \delta\mathbf{x}) & = & \mathbf{b} + \delta\mathbf{b} \end{cases}. \tag{2.3}$$

Soit $\| \, . \, \|$ une norme matricielle subordonnée. Du fait de (2.2), (2.3) implique successivement :

$$\begin{cases} \mathbf{A}\mathbf{x} & = & \mathbf{b} \\ \delta\mathbf{x} & = & \mathbf{A}^{-1}\delta\mathbf{b} \end{cases} \Rightarrow \begin{cases} \| \, \mathbf{b} \, \| & \leq & \| \, \mathbf{A} \, \| \| \, \mathbf{x} \, \| \\ \| \, \delta\mathbf{x} \, \| & \leq & \| \, \mathbf{A}^{-1} \, \| \| \, \delta\mathbf{b} \, \| \end{cases} \Rightarrow \frac{\| \, \delta\mathbf{x} \, \|}{\| \, \mathbf{x} \, \|} \leq \underbrace{\| \, \mathbf{A} \, \| \| \, \mathbf{A}^{-1} \, \|}_{} \times \frac{\| \, \delta\mathbf{b} \, \|}{\| \, \mathbf{b} \, \|}. \tag{2.4}$$

Soit maintenant $\mathbf{x} + \delta\mathbf{x}$ la solution exacte du système dont la matrice a été perturbée : $(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b}$. On obtient de même :

$$\frac{\| \, \delta\mathbf{x} \, \|}{\| \, \mathbf{x} + \delta\mathbf{x} \, \|} \leq \underbrace{\| \, \mathbf{A} \, \| \| \, \mathbf{A}^{-1} \, \|}_{} \times \frac{\| \, \delta\mathbf{A} \, \|}{\| \, \mathbf{A} \, \|}. \tag{2.5}$$

A partir de (2.4) comme de (2.5), on voit donc que relativement à l'erreur relative sur le membre de droite (respectivement sur la matrice), l'erreur relative sur la solution $\mathbf{x}$ *peut* être d'autant plus grande que le nombre positif $\| \, \mathbf{A} \, \| \| \, \mathbf{A}^{-1} \, \|$ est grand, et par

ailleurs que si ce nombre $\| \mathbf{A} \| \| \mathbf{A}^{-1} \|$ est petit, une petite perturbation (quelle qu'elle soit) sur le membre de droite (resp. sur la matrice) n'entraînera pas de modification importante de la solution $\mathbf{x}$.

Cette plus ou moins grande sensibilité de la solution d'un système linéaire à une perturbation sur la matrice ou sur le membre de droite dépend donc du *conditionnement du système linéaire* $\mathbf{A}$ (*condition number*, en anglais), défini comme étant :

$$cond(\mathbf{A}) \stackrel{def}{=} \| \mathbf{A} \| \| \mathbf{A}^{-1} \| . \tag{2.6}$$

Propriétés

Les propriétés suivantes (qui se déduisent de façon immédiate de la définition 2.1.6) sont vraies pour toute norme matricielle subordonnée :

$$\begin{aligned} \text{cond}(\mathbf{A}) &\geq 1; && (2.7)\\ \text{cond}(\mathbf{A}) &= \text{cond}(\mathbf{A}^{-1}); && (2.8)\\ \alpha \neq 0, \quad \text{cond}(\alpha\mathbf{A}) &= \text{cond}(\mathbf{A}). && (2.9) \end{aligned}$$

Commentons maintenant ces résultats. D'après la propriété (2.7), on dira qu'un système linéaire est *bien conditionné* si son conditionnement est égal à 1 (ou est proche de 1), et mal conditionné dans le cas contraire. La propriété (2.9) implique que dans le cas où le système $\mathbf{Ax} = \mathbf{b}$ serait mal conditionné, il est illusoire de penser qu'il serait préférable de résoudre le système équivalent $\alpha\mathbf{Ax} = \alpha\mathbf{b}$ (α étant une constante quelconque), puisque ces deux systèmes admettent le même conditionnement.

En revanche, un même vecteur $\mathbf{x}$, solution du système originel $\mathbf{Ax} = \mathbf{b}$, est également solution de systèmes différents qui, eux, peuvent être mieux conditionnés. Cette idée est à l'origine de la technique de *l' équilibrage d'une matrice* :

— minimiser cond $(\Delta_1\mathbf{A}\Delta_2)$, Δ_1, Δ_2 diagonales inversibles ;

— résoudre $(\Delta_1\mathbf{A}\Delta_2)\mathbf{v} = \Delta_1\mathbf{b}$, puis calculer $\mathbf{x} = \Delta_2\mathbf{v}$.

Venons en maintenant aux propriétés spécifiques de la norme-2.

Proposition 2.2.1 *Les propriétés suivantes sont spécifiques de la norme* $\| . \|_2$ *:*

— $cond_{\|.\|_2}(\mathbf{A}) = \frac{\sigma_{\max}}{\sigma_{\min}}$ *;*

— $cond_{\|.\|_2}(\mathbf{A}) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$ *si* $\mathbf{A}$ *est normale ;*

— $cond_{\|.\|_2}(\mathbf{A})$ *est invariant par transformation unitaire : si* $\mathbf{U}\mathbf{U}^H = \mathbf{I}$,

$$cond_{\|.\|_2}(\mathbf{A}) = cond_{\|.\|_2}(\mathbf{AU}) = cond_{\|.\|_2}(\mathbf{U}^H\mathbf{A}) = cond_{\|.\|_2}(\mathbf{U}^H\mathbf{AU}); \tag{2.10}$$

— $cond_{\|.\|_2}(\mathbf{A}) = 1 \Leftrightarrow \mathbf{A} = \alpha\mathbf{U}$, *avec* $\alpha \neq 0$ *et* $\mathbf{U}$ *unitaire (resp. orthogonale).*

La propriété (2.10) justifie le rôle privilégié joué par les matrices unitaires (ou orthogonales) en analyse numérique matricielle (nous reviendrons sur ce point lors de la description de la méthode de Householder).

2.2.2 Conditionnement d'un problème de valeurs propres

Exemple introductif

De même que pour la résolution des systèmes linéaires, le problème de l'extraction de valeurs propres d'une matrice peut être très sensible à une variation des éléments de cette matrice. Considérons par exemple le cas de la recherche des valeurs propres $\{\lambda_i\}$ de la matrice $n \times n$

$$\mathbf{A}(\epsilon) = \begin{bmatrix} 0 & & & \epsilon \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} .$$

Si $\epsilon = 0$, alors $\lambda_i = 0 \ \forall \ i$; mais si $n = 40$, et $\epsilon = 10^{-40}$, $|\lambda_i| = 10^{-1} \ \forall i$. Dans cet exemple, on voit qu'une petite variation d'un des éléments de la matrice entraîne une variation du module des valeurs propres 10^{39} fois plus importante ...

Conditionnement d'un problème de valeurs propres

Le théorème de Bauer-Ficke assure que si on perturbe une matrice diagonalisable $\mathbf{A}$ en $\mathbf{A} + \delta\mathbf{A}$, alors les valeurs propres de la matrice perturbée sont situées, dans le plan complexe, à l'intérieur de la réunion d'un ensemble de disques centrés sur les valeurs propres de la matrice originelle :

Théorème 2.2.1 (Bauer-Fike) *Soit* $\mathbf{A}$ *une matrice diagonalisable* $n \times n$, $\mathbf{P}$ *une matrice telle que* $\mathbf{P}^{-1}\mathbf{AP} = diag(\lambda_i)$, *et* $\| \ . \ \|$ *une norme matricielle telle que* $\| \ diag(\lambda_i) \ \| = \max_i |\lambda_i|$ *(ce qui est vérifié pour les normes* $\| \ . \ \|_1$, $\| \ . \ \|_2$ *et* $\| \ . \ \|_\infty$*). Alors* $\forall \ \delta\mathbf{A}$ *telle que* $\mathbf{A} + \delta\mathbf{A}$ *est diagonalisable, Spectre* $(\mathbf{A} + \delta\mathbf{A}) \subset \bigcup_{i=1}^{n} D_i$, $D_i = \{z \in \mathbb{C}, |z - \lambda_i| \leq cond(\mathbf{P}) \times \| \ \delta\mathbf{A} \ \|\}$.

Comme le théorème est vrai quelle que soit la matrice de passage $\mathbf{P}$, on en déduit immédiatement le corollaire suivant :

Corollaire 2.2.1 $\forall \ \mathbf{A}$ *diagonalisable* $n \times n$, $\forall \ \delta\mathbf{A}$ *telle que* $\mathbf{A} + \delta\mathbf{A}$ *diagonalisable,*

$$Spectre(\mathbf{A}+\delta\mathbf{A}) \subset \bigcup_{i=1}^{n} \{z \in \mathbb{C}, |z-\lambda_i| \leq \underbrace{\inf\{cond(\mathbf{P}), t.q. \mathbf{P}^{-1}\mathbf{AP} = diag(\lambda_i)\}}_{\Gamma(\mathbf{A})} \times \| \ \delta\mathbf{A} \ \|\}.$$

Le nombre $\Gamma(\mathbf{A})$ *est le* Conditionnement de $\mathbf{A}$ relativement au calcul de ses valeurs propres[3]. □

Les valeurs propres d'une matrice perturbée $\mathbf{A} + \delta\mathbf{A}$ sont donc situés dans des disques dont les rayons sont d'autant plus petits que $\Gamma(\mathbf{A})$ est petit. A ce sujet, remarquons tout de suite que (2.7) implique $\Gamma(\mathbf{A}) \geq 1$. Par conséquent, une matrice sera bien conditionnée (pour la recherche de ses valeurs propres) si $\Gamma(\mathbf{A}) = 1$, ou est proche de 1 :

3. Attention : dans la définition de $\Gamma(\mathbf{A})$, le terme cond$(\mathbf{P})$ est le conditionnement de $\mathbf{P}$ *relativement à la résolution d'un système linéaire.*

Exemple 2.2.1 Soit $\mathbf{A}$ une matrice normale. D'après le théorème 2.1.5, $\exists\, \mathbf{P}$ unitaire et $\mathbf{\Lambda}$ diagonale, telles que $\mathbf{PAP}^H = \mathbf{\Lambda}$. Or d'après la proposition 2.2.1, $\text{cond}_{\|.\|_2}(\mathbf{P}) = 1$, donc $\Gamma(\mathbf{A}) = 1$. Par conséquent, les matrices normales (et donc, en particulier, les matrices hermitiennes, les matrices unitaires (dans le cas complexe), les matrices symétriques, ainsi que les matrices orthogonales (dans le cas réel)) sont bien conditionnées pour la recherche de leurs valeurs propres. ■

2.2.3 Remarques importantes sur le conditionnement

— **Conditionnements des différents problèmes liés à une matrice A.**
Dire qu'une matrice $\mathbf{A}$ est "plus ou moins bien conditionnée" n'a pas de sens : le conditionnement dépend du problème que l'on cherche à résoudre (résolution d'un système linéaire de matrice $\mathbf{A}$, recherche des valeurs propres de $\mathbf{A}$, recherche des vecteurs propres de $\mathbf{A}$ ...). Une matrice peut être bien conditionnée pour la résolution de systèmes linéaires, et mal conditionnée pour la recherche de valeurs propres (ou vice-versa) ; une matrice peut par ailleurs être bien conditionnée pour la recherche de ses valeurs propres, et mal conditionnée pour la recherche de ses vecteurs propres.

Considérons à titre d'exemple une matrice symétrique réelle $\mathbf{A}$ de valeurs propres $\{\lambda_i\}$:

- Comme $\Gamma_{\|.\|_2}(\mathbf{A}) = 1$, la recherche des valeurs propres de $\mathbf{A}$ est un problème très bien conditionné (et ce, quelle que soit la répartition effective des valeurs propres de $\mathbf{A}$) ;
- comme $\mathbf{A}$ est normale, le conditionnement (selon la norme 2) de la résolution d'un système linéaire de matrice $\mathbf{A}$ est $\frac{\max(|\lambda_i|)}{\min(|\lambda_i|)}$, et est donc d'autant meilleur que les différentes valeurs propres sont sensiblement de même module ;
- en revanche, on montre que la recherche des vecteurs propres sera d'autant mieux conditionnée que les valeurs propres seront bien séparées (et donc, intuitivement, que les espaces propres seront bien séparés) : si le problème de résolution de système linéaire est bien conditionné, celui de recherche de vecteurs propres ne le sera pas (et vice-versa).

— **Deux phénomènes distincts à ne pas confondre.**
Il faut par ailleurs veiller à ne pas confondre *le conditionnement d'un problème* et *la stabilité numérique d'un algorithme* :

- Soit à calculer $y = f(x)$. Le conditionnement traduit la sensibilité de $y = f(x)$ aux petites variations sur x ; c'est une propriété *de l'application* f, *pas* de l'algorithme employé en pratique. Cette propriété de la fonction f est indépendante du mode de représentation des nombres adoptés dans l'ordinateur ; elle existerait même si les nombres étaient représentés avec une précision infinie.
- En pratique, un algorithme de calcul fournit une solution numérique. Les calculs se font nécessairement en précision finie ; il en résulte des *erreurs d'arrondi*, qui peuvent s'amplifier d'une itération à l'autre par un phénomène

de *propagation d'erreurs* : c'est ce que traduit la notion de stabilité (ou d'instabilité) numérique d'un algorithme.

2.3 Algorithmes de résolution de systèmes linéaires

Nous abordons désormais une description rapide de quelques algorithmes utilisés en pratique pour résoudre un système linéaire.

Soit à résoudre le système linéaire $\mathbf{Ax} = \mathbf{b}$, où $\mathbf{A}$ est une matrice carrée inversible de dimension n. La solution *analytique* est bien évidemment $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. Cependant, *en pratique* ce n'est pas l'implantation directe de cette formule qui est utilisée pour calculer $\mathbf{x}$, ne serait-ce que pour des raisons de coût de calcul : calculer la matrice $\mathbf{A}^{-1}$ revient en effet à calculer ses n colonnes, c'est-à-dire à résoudre les n systèmes linéaires $\mathbf{Au}_i = [0 \cdots 010 \cdots 0]^T$, le "1" du membre de droite étant en $i^{ème}$ position.

Comment procède-t-on donc ? Commençons par un exemple pratique de résolution de système linéaire. Considérons le système

$$\left\{ \begin{array}{rcr} 2x + y & = & 1 \\ x - 3y & = & -2 \end{array} \right. .$$

Une solution simple consiste à additionner 3 fois la première ligne à la seconde ligne : on obtient ainsi l'équation $7x = 1$ d'où l'on déduit immédiatement $x = 1/7$. En procédant de la même façon on obtient y et le système est entièrement résolu.

De façon générale, les algorithmes classiques de résolution de systèmes linéaires [4] consistent à transformer le système originel $\mathbf{Ax} = \mathbf{b}$ en un système équivalent plus simple à résoudre. Supposons qu'on ne s'autorise comme ci-dessus que des opérations élémentaires (additions, multiplications) sur les lignes de la matrice $\mathbf{A}$; alors un algorithme donné est décrit de manière synthétique par l'équivalence

$$\mathbf{Ax} = \mathbf{b} \Longleftrightarrow \underbrace{\mathbf{MA}}_{\mathbf{U}}\mathbf{x} = \underbrace{\mathbf{Mb}}_{\mathbf{b'}} \ , \tag{2.11}$$

$\mathbf{M}$ étant une matrice inversible qui regroupe l'ensemble des opérations successives effectuées sur les lignes de $\mathbf{A}$, et $\mathbf{U}$ étant une matrice qui est telle que l'équation $\mathbf{Ux} = \mathbf{b'}$ soit de résolution aisée.

Les deux méthodes que nous allons décrire maintenant : *la méthode de Gauss* et *la méthode de Householder*, procèdent de cette façon. Elles consistent à transformer progressivement la matrice originelle $\mathbf{A}$ en une matrice $\mathbf{U} = \mathbf{MA}$, où $\mathbf{U}$ est triangulaire supérieure (d'où la notation classique $\mathbf{U}$ pour *upper triangular matrix*). Du fait de la structure de $\mathbf{U}$, la résolution du système $\mathbf{Ux} = \mathbf{b'}$ est alors immédiate : on commence par calculer x_n, puis (une fois que l'on connaît x_n) on calcule x_{n-1}, et l'on remonte ainsi de suite jusqu'à x_1. Les deux méthodes diffèrent essentiellement dans la façon de construire la matrice $\mathbf{U}$.

4. du moins les algorithmes directs ; nous ne parlerons pas ici des méthodes itératives qui constituent une autre classe d'algorithmes.

2.3.1 Méthode de Gauss et factorisation LU

Méthode de Gauss

La triangulation de $\mathbf{A}_{n\times n}$ est obtenue en $n-1$ étapes successives :

— Supposons tout d'abord que le premier élément $a_{1,1}$ de $\mathbf{A}$ soit différent de 0. Il est alors possible, en effectuant des combinaisons linéaires bien choisies d'une ligne quelconque avec la première ligne, de transformer $\mathbf{A}$ en une matrice $\mathbf{A}'$ dont la première colonne est constituée de zéros, à l'exception du premier élément :

$$\underbrace{\begin{bmatrix} 1 & & & 0 \\ -\frac{a_{2,1}}{a_{1,1}} & 1 & & \\ \vdots & & \ddots & \\ -\frac{a_{n,1}}{a_{1,1}} & & & 1 \end{bmatrix}\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & \ddots & & \vdots \\ \vdots & & & \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}}_{\mathbf{A}'=\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ 0 & a'_{2,2} & \cdots & a'_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & a'_{n,2} & \cdots & a'_{n,n} \end{bmatrix}}\begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}$$

— Supposons maintenant que $a'_{2,2}$ soit différent de zéro. La deuxième étape consiste, en effectuant des combinaisons linéaires bien choisies d'une ligne quelconque $i \in [3, \cdots, n]$ avec la deuxième ligne, à transformer $\mathbf{A}'$ en une matrice $\mathbf{A}''$ dont la deuxième colonne est identiquement nulle, à l'exception des deux premiers éléments :

$$\begin{bmatrix} 1 & & & 0 \\ 0 & 1 & & \\ \vdots & -a'_{3,2}/a'_{2,2} & \ddots & \\ 0 & -a'_{n,2}/a'_{2,2} & 0 & 1 \end{bmatrix}\underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ 0 & a'_{2,2} & \cdots & a'_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & a'_{n,2} & \cdots & a'_{n,n} \end{bmatrix}}_{\mathbf{A}'} = \underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & & \cdots a_{1,n} \\ 0 & a'_{2,2} & & \cdots a'_{2,n} \\ \vdots & 0 & a''_{3,3} & \cdots a''_{3,n} \\ & \vdots & \vdots & \vdots \\ 0 & 0 & a''_{n,3} & \cdots a''_{n,3} \end{bmatrix}}_{\mathbf{A}''} \tag{2.12}$$

— en continuant de cette façon, à la $(n-1)^{ème}$ étape, la matrice du système linéaire est devenue triangulaire supérieure :

$$\begin{bmatrix} a_{1,1} & * & \cdots & * \\ 0 & a'_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \cdots & 0 & a_{n,n}^{(n-1)} \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ \vdots \\ b_n^{(n-1)} \end{bmatrix}$$

et le calcul de x_n, puis de $x_{n-1}, \cdots, x_1$ est alors immédiat.

Remarques.

— Dans la description de l'algorithme nous avons supposé qu'à chaque étape l'élément $a_{i,i}^{(i-1)}$ (le *pivot de Gauss*) était différent de zéro. Dans le cas contraire, il suffit de réordonner au préalable les lignes de la sous-matrice sur laquelle s'effectuera la transformation, de façon à placer en première position un élément différent de zéro (ce qui est toujours possible puisque dét($\mathbf{A}$) $\neq 0$) ;

— Par ailleurs, même dans le cas où $a_{i,i}^{(i-1)} \neq 0$, il est pertinent (pour des considérations numériques) de réordonner de façon systématique les lignes, de façon à choisir comme pivot, parmi les éléments de la colonne à annuler, l'élément (ou un des éléments) de plus grande valeur absolue ; c'est la *stratégie du pivot partiel* (d'autres statégies sont également possibles) ;

— Observons au passage que l'algorithme de Gauss est également un algorithme de calcul du déterminant d'une matrice, puisque le déterminant de $\mathbf{A}$ est égal au produit des pivots successifs ;

— Effectuons une dernière remarque concernant le coût de calcul de l'algorithme. Prenons $n = 10$, et posons $\mathbf{x} = [x_1 \cdots x_n]^T$. L'application directe des formules de Cramer :

$$x_i = \frac{\textit{dét.}(\mathbf{B}_i)}{\textit{dét.}(\mathbf{A})}, \quad \mathbf{B}_i = \begin{bmatrix} a_{1,1}..a_{1,i-1} & b_1 & a_{1,i+1}..a_{1,n} \\ \vdots & \vdots & \vdots \\ a_{n,1}..a_{n,i-1} & b_n & a_{1,i+1}..a_{1,n} \end{bmatrix}$$

nécessite 400.000.000 opérations élémentaires (multiplications et additions), alors que l'algorithme de Gauss n'en requiert que 900 (no comment ...)

Factorisation LU, Factorisation de Choleski

Revenons sur l'algorithme de Gauss. A chaque étape, la matrice de transformation est triangulaire inférieure avec des "1" sur la diagonale. Le produit de matrices triangulaires inférieures étant également triangulaire inférieure, la matrice $\mathbf{M}$ qui résulte de la succession des $n-1$ étapes (et qui apparaît dans la factorisation (2.11)) est également triangulaire inférieure. L'inverse d'une matrice triangulaire inférieure étant triangulaire inférieure, on voit en définitive que $\mathbf{A}$ peut se factoriser en $\mathbf{A} = \mathbf{M}^{-1}\mathbf{U}$, avec $\mathbf{L} = \mathbf{M}^{-1}$ triangulaire inférieure (d'où la notation $\mathbf{L}$ pour *lower triangular matrix*) et $\mathbf{U}$ triangulaire supérieure. Il reste à exprimer le fait que le résultat n'est valable que si les pivots $a_{i,i}^{(i-1)}$ de $\mathbf{A}$ sont tous inversibles. Le résultat est résumé dans le théorème suivant :

Théorème 2.3.1 Factorisation LU. *Soit* $\mathbf{A}$ *une matrice carrée* fortement régulière*, c'est à dire telle que tous les mineurs principaux* $\{a_{1,1}$*, dét.(*$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$*),* $\cdots$*, dét.(*$\mathbf{A}$*)*$\}$*, soient tous différents de zéro. Alors* $\mathbf{A}$ *peut se factoriser de façon unique en* $\mathbf{A} = \mathbf{LU}$*,* $\mathbf{L}$ *étant une matrice triangulaire inférieure avec des "1" sur la diagonale, et* $\mathbf{U}$ *étant une matrice triangulaire supérieure.*

Remarquons que la condition : "$\mathbf{A}$ est fortement régulière", est une condition plus forte que la condition : "$\mathbf{A}$ est régulière" (c'est-à-dire inversible) (prenons l'exemple de la matrice $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$). Cependant, il est toujours possible de réordonner les lignes d'une

matrice inversible pour la rendre fortement inversible, et de ce point de vue (c'est-à-dire, à une permutation des lignes près), toute matrice inversible admet une factorisation LU.

Considérons maintenant le cas des matrices symétriques définies positives. Toute matrice symétrique définie positive est fortement régulière ; en appliquant la factorisation LU à de telles matrices (et au prix de quelques manipulations algébriques), on obtient aisément le résultat suivant :

Théorème 2.3.2 Factorisation de Choleski. *Soit* $\mathbf{A}$ *une matrice carrée, réelle, symétrique, et définie positive. Alors* $\mathbf{A}$ *peut se factoriser en* $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, $\mathbf{L}$ *étant une matrice triangulaire inférieure dont tous les éléments diagonaux sont strictement positifs. Une telle factorisation est unique.*

Remarque. Une "racine carrée" d'une matrice $\mathbf{A}$ est une matrice $\mathbf{M}$ satisfaisant $\mathbf{A} = \mathbf{M}\mathbf{M}^T$. La décomposition de Choleski montre que toute matrice symétrique définie positive ($> \mathbf{0}$) admet au moins une racine carrée (le facteur $\mathbf{L}$ étant une racine très particulière). Une matrice $> \mathbf{0}$ a une infinité de racines carrées : en effet, si $\mathbf{A} = \mathbf{M}\mathbf{M}^T = \mathbf{N}\mathbf{N}^T$, alors $(\mathbf{M}^{-1}\mathbf{N})\ (\mathbf{M}^{-1}\mathbf{N})^T = \mathbf{I}$, donc $\mathbf{N} = \mathbf{M}\mathbf{Q}$ pour une certaine matrice orthogonale $\mathbf{Q}$. Ce résultat généralise donc la décomposition classique de tout nombre réel positif x en $x = (\sqrt{x})^2 = (-\sqrt{x})^2$, $+1$ et -1 étant les deux seules matrices réelles orthogonales en dimension 1.

Applications. Un intérêt majeur de la factorisation LU (ou de la factorisation de Choleski) apparaît lorsqu'on est amené à résoudre plusieurs systèmes linéaires $\{\mathbf{A}\mathbf{x}_i = \mathbf{b}_i\}_{i=1}^m$ de même matrice $\mathbf{A}$. En effet, en appliquant la méthode de Gauss une première fois, on obtient implicitement deux facteurs $\mathbf{L}$ et $\mathbf{U}$ de $\mathbf{A}$. Pour résoudre ensuite les systèmes

$$\mathbf{L}\underbrace{\mathbf{U}\ \mathbf{x}_i}_{\mathbf{v}_i} = \mathbf{b}_i\ ,$$

on peut donc résoudre successivement les deux systèmes $\mathbf{L}\mathbf{v}_i = \mathbf{b}_i$, puis $\mathbf{U}\mathbf{x}_i = \mathbf{v}_i$; chacun de ces deux systèmes étant très aisé à résoudre puisque les deux matrices $\mathbf{L}$ et $\mathbf{U}$ sont triangulaires.

2.3.2 Méthode de Householder et factorisation QR

Nous allons maintenant décrire la méthode de Householder. Cette méthode transforme également, par actions successives sur ses lignes, la matrice $\mathbf{A}$ en une matrice triangulaire supérieure, mais utilise des matrices de transformation différentes de celles utilisées par l'algorithme de Gauss.

Méthode de Householder

Soit $\mathbf{u} = [u_1 \cdots u_n]^T$. Introduisons la matrice de Householder élémentaire $\mathbf{H}(\mathbf{u})$, définie par

$$\mathbf{H}(\mathbf{u}) = \mathbf{I} - \frac{2}{\mathbf{u}^H \mathbf{u}} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} [\overline{u}_1 \cdots \overline{u}_n]. \tag{2.13}$$

Posons $\mathbf{e}_1 = [1 0 \cdots 0]^T$. On vérifie aisément que $\mathbf{H}(\mathbf{u})$ est hermitienne, *unitaire*, et que

$$\mathbf{H}(\mathbf{a} \pm \| \mathbf{a} \| \mathbf{e}_1) \, \mathbf{a} = \mp \| \mathbf{a} \| \mathbf{e}_1. \tag{2.14}$$

Appelons maintenant $\mathbf{a} = [a_{1,1} \cdots a_{n,1}]^T$ la première colonne de $\mathbf{A}$. Grâce à (2.14), on a :

$$\mathbf{H}(\mathbf{a} \pm \| \mathbf{a} \| \mathbf{e}_1) \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & \ddots & & \vdots \\ \vdots & & & \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} = \begin{bmatrix} a'_{1,1} & a'_{1,2} & \cdots & a'_{1,n} \\ 0 & a'_{2,2} & \cdots & a'_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & a'_{n,2} & \cdots & a'_{n,n} \end{bmatrix}.$$

On voit donc qu'il est possible, grâce à des combinaisons linéaires bien choisies (-combinaisons décrites par la matrice $\mathbf{H}(\mathbf{a} \pm \| \mathbf{a} \| \mathbf{e}_1)$) des lignes de $\mathbf{A}$, de transformer $\mathbf{A}$ en une matrice $\mathbf{A}'$ dont la première colonne est nulle, à l'exception du premier élément.

Soit $\mathbf{a}' = [a'_{2,2} \cdots a'_{n,2}]^T$ et $\mathbf{e}'_1 = [1 0 \cdots 0]^T$ (vecteurs de dimension $n-1$). La deuxième étape affecte les lignes 2 à n de $\mathbf{A}'$:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & \multicolumn{3}{c}{\mathbf{H}(\mathbf{a}' \pm \| \mathbf{a}' \| \mathbf{e}_1)} \\ 0 & & & \end{bmatrix} \underbrace{\begin{bmatrix} a'_{1,1} & a'_{1,2} & \cdots & a'_{1,n} \\ 0 & a'_{2,2} & \cdots & a'_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & a'_{n,2} & \cdots & a'_{n,n} \end{bmatrix}}_{\mathbf{A}'} = \underbrace{\begin{bmatrix} a'_{1,1} & * & \cdots & * \\ 0 & a''_{2,2} & \cdots & * \\ \vdots & 0 & * & * \\ 0 & 0 & * & * \end{bmatrix}}_{\mathbf{A}''}.$$

De même que la méthode de Gauss, on voit donc que la méthode de Householder va permettre, en $n-1$ étapes, de transformer la matrice originelle en une matrice triangulaire supérieure, c'est-à-dire, implicitement, de transformer le système linéaire originel en un système linéaire simple à résoudre.

L'intérêt de cet algorithme par rapport à l'algorithme de Gauss provient de ce que, par construction, toutes les matrices de transformation élémentaire utilisées sont des matrices unitaires ; du fait de la propriété (2.10), le conditionnement du système originel (de matrice $\mathbf{A}$) est égal au conditionnement du système simplifié (de matrice triangulaire supérieure $\mathbf{U}$) : *les transformations successives de* $\mathbf{A}$ *ne peuvent donc pas dégrader son conditionnement.* En définitive, l'algorithme de Householder s'avère être mieux conditionné que l'algorithme de Gauss ; la contrepartie réside en un coût de calcul à peu près deux fois plus élevé.

Factorisation QR

De même que la méthode de Gauss fournit implicitement la factorisation LU d'une matrice $\mathbf{A}$, nous allons voir que la méthode de Householder fournit implicitement une autre factorisation, appelée *factorisation QR* de $\mathbf{A}$. Afin de simplifier les notations, appelons $\mathbf{Q}_i$ la matrice de Householder utilisée lors de la $i^{ème}$ étape de l'algorithme. Graphiquement, les étapes successives de l'algorithme peuvent être représentées de la façon suivante :

$$\mathbf{Q}_1 \underbrace{\begin{bmatrix} * & * & \cdots & * \\ * & * & \cdots & * \\ \vdots & \vdots & & \vdots \\ * & * & \cdots & * \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \cdots & * \end{bmatrix}}_{\mathbf{A}'},$$

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \mathbf{Q}_2 & \\ 0 & & & \end{bmatrix} \underbrace{\begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \cdots & * \end{bmatrix}}_{\mathbf{A}'} = \underbrace{\begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & 0 & * & * \\ 0 & 0 & * & * \end{bmatrix}}_{\mathbf{A}''}, \quad \cdots$$

Toutes les matrices de transformation sont unitaires ; le produit de matrices unitaires étant unitaire, on obtient en définitive le résultat suivant :

Théorème 2.3.3 Factorisation QR. *Soit* $\mathbf{A}$ *une matrice carrée de dimension* n*. Alors* $\mathbf{A}$ *peut se factoriser en* $\mathbf{A} = \mathbf{QR}$*,* $\mathbf{Q}$ *étant une matrice unitaire, et* $\mathbf{R}$ *une matrice triangulaire supérieure. On peut s'arranger pour que les éléments diagonaux* $r_{i,i}$ *de* $\mathbf{R}$ *soient positifs ou nuls ; en ce cas, la factorisation est unique si* $\mathbf{A}$ *est inversible.*

Remarques.

— La factorisation QR se généralise aux matrices non carrées. Considérons le cas où $\mathbf{A}$ est $q \times p$ avec $q > p$. Si $\mathbf{A}$ est de rang complet p, $\mathbf{A}$ peut s'écrire de façon unique $\mathbf{A} = \mathbf{QR}$, où $\mathbf{Q}$ est une matrice $q \times p$ constituée de vecteurs orthonormés (une sous-matrice d'une matrice unitaire), et $\mathbf{R}$ une matrice triangulaire supérieure inversible.

— Remarquons également que dans ce cas il existe un lien immédiat entre *factorisation QR* et *procédé d'orthonormalisation de Gram Schmidt* (il suffit de remarquer que si $\mathbf{A} = \mathbf{QR}$, alors $\mathbf{Q} = \mathbf{AR}^{-1}$ avec $\mathbf{R}^{-1}$ triangulaire supérieure ; et cette dernière égalité n'exprime rien d'autre que la construction progressive d'une base orthonormée de l'espace vectoriel engendré par les colonnes de $\mathbf{A}$ à partir de vecteurs de cet espace).

— Remarquons enfin que si $\mathbf{A} = \mathbf{QR}$, alors

$$\mathbf{A}^H\mathbf{A} = \mathbf{R}^H \underbrace{\mathbf{Q}^H\mathbf{Q}}_{\mathbf{I}} \mathbf{R} = \mathbf{R}^H\mathbf{R} .$$

La factorisation QR de $\mathbf{A}$ fournit donc implicitement la factorisation de Choleski de $\mathbf{A}^H\mathbf{A}$.

2.3.3 Résolution d'un système au sens des Moindres Carrés

Jusqu'ici nous avons parlé de la résolution de systèmes linéaires carrés et inversibles, mais il est également important de pouvoir proposer des solutions à des systèmes sur- ou sous-dimensionnés, ou de proposer des solutions approchées pertinentes à des systèmes linéaires n'admettant pas de solution exacte. Nous évoquerons donc la très importante *Méthode des Moindres Carrés*, qui est largement utilisée du fait de ses bonnes propriétés statistiques (cf. cours de deuxième année ...un peu de patience s.v.p.).

Exemple.

Commençons par un exemple de façon à fixer les idées. Considérons un mobile en mouvement rectiligne uniforme, dont la position $y(t) = v \times t + y_0$, et dont on voudrait connaître la vitesse v et la position initiale y_0. Pour celà, effectuons des observations à des instants régulièrement espacés $t = 0, 2, \cdots, N$. En pratique, de multiples sources d'erreurs présentes dans les appareils de mesure font que ces observations sont imparfaites. Cette imprécision est généralement modélisée de la façon suivante : on n'observe pas $y_k = v \times k + y_0$, mais $y_k = (v \times k + y_0) + b_k$, b_k étant un "bruit" de mesure. On obtient donc le système

$$\underbrace{\begin{bmatrix} y_0 \\ \vdots \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & N \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} y_0 \\ v \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} b_0 \\ \vdots \\ \vdots \\ b_N \end{bmatrix}}_{\mathbf{b}}, \tag{2.15}$$

dans lequel $\mathbf{x}$ et $\mathbf{b}$ sont inconnus.

Ce système admet une infinité de couples $(\mathbf{x}, \mathbf{b})$ solutions, car pour tout $\mathbf{x}$, (2.15) est vérifiée si l'on choisit $\mathbf{b} = \mathbf{y} - \mathbf{A}\mathbf{x}$. Il convient donc de retenir une solution raisonnable, c'est-à-dire une solution pour laquelle le bruit de mesure reste "petit" (- après tout, $\mathbf{b}$ ne modélise qu'un terme de perturbation) :

— On ne peut pas purement et simplement prendre $\mathbf{b} = \mathbf{0}$, car dans ce cas (2.15) devient

$$\underbrace{\begin{bmatrix} y_0 \\ \vdots \\ \vdots \\ y_N \end{bmatrix}}_{\notin \mathrm{Im}(\mathbf{A})} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & N \end{bmatrix} \mathbf{x}}_{\in \mathrm{Im}(\mathbf{A}) \,\forall\, \mathbf{x}}, \tag{2.16}$$

système qui en général n'admet pas de solution (même si, comme c'est le cas dans cet exemple, $\mathbf{A}$ est de rang complet) : en effet, le vecteur de mesures $\mathbf{y}$ n'a

aucune raison d'appartenir à l'espace image de la matrice $\mathbf{A}$ (on dit dans ce cas que l'équation $\mathbf{y} = \mathbf{Ax}$ est *inconsistente*).

— En revanche, il est pertinent de retenir la (ou les) solution(s) $\mathbf{x}_{MC}$ qui soi(en)t telle(s) que $\sum_{i=0}^{N} b_i^2$ soit minimum ; c'est le principe de la *résolution de systèmes linéaires au sens des Moindres Carrés*. Dit d'une autre façon, puisque le système linéaire (2.16) n'admet pas de solution exacte, il faut se contenter de retenir une solution approchée ; une solution $\mathbf{x}_{MC}$ au sens des moindres carrés est une solution qui minimise, sur tous les vecteurs $\mathbf{x}$, la norme $\| \, . \, \|_2$ de l'erreur $\mathbf{b} = \mathbf{y} - \mathbf{Ax}$.

Méthode des Moindres Carrés

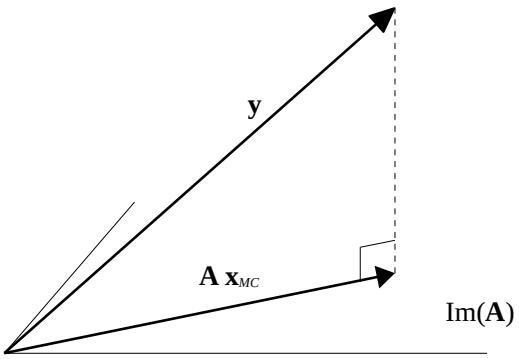
Considérons donc le système

$$\mathbf{y} = \mathbf{Ax} + \mathbf{b}, \tag{2.17}$$

dans lequel $\mathbf{y}$ est un vecteur d'observations, $\mathbf{A}$ une matrice (connue) de dimensions $q \times p$, $\mathbf{x}$ un vecteur (inconnu) de dimension p, et $\mathbf{b}$ un bruit additif (inconnu). Pour simplifier, nous considérerons uniquement le cas où $\mathbf{y}$, $\mathbf{A}$, $\mathbf{x}$ et $\mathbf{b}$ sont réelles. Une solution de (2.17) au sens des Moindres Carrés, si elle existe, est un vecteur $\mathbf{x}_{MC}$ solution du problème d'optimisation suivant :

$$\mathbf{x}_{MC} = \arg\min_{\mathbf{x}} \| \, \mathbf{y} - \mathbf{Ax} \, \|_2^2 \, . \tag{2.18}$$

On peut résoudre (2.18) : soit analytiquement (- en écrivant que les dérivées partielles de $\| \, \mathbf{y} - \mathbf{Ax} \, \|_2^2$ par rapport à chacune des composantes de $\mathbf{x}$ sont toutes nulles), soit géométriquement. L'approche géométrique consiste à observer que $\forall \, \mathbf{x}$, $\mathbf{Ax}$ doit appartenir à l'espace $\mathrm{Im}(\mathbf{A})$ engendré par les colonnes de la matrice $\mathbf{A}$. On cherche donc un vecteur $\mathbf{x}_{MC}$ qui soit tel que la distance entre $\mathbf{y}$ (qui n'appartient pas à $\mathrm{Im}(\mathbf{A})$) et $\mathbf{Ax}_{MC}$ (qui appartient à $\mathrm{Im}(\mathbf{A})$) soit minimale : *par conséquent, $\mathbf{Ax}_{MC}$ est la projection orthogonale de $\mathbf{y}$ sur Im($\mathbf{A}$).*



Afin de donner une expression analytique de $\mathbf{Ax}_{MC}$, observons que $\mathbf{Ax}_{MC} = \mathbb{P}_{\mathrm{Im}(\mathbf{A})}(\mathbf{y})$ si et seulement si le résidu de projection $(\mathbf{y} - \mathbf{Ax}_{MC})$ est orthogonal à $\mathrm{Im}(\mathbf{A})$. $\mathbf{x}_{MC}$ vérifie donc $\mathbf{A}^T(\mathbf{y} - \mathbf{Ax}_{MC}) = \mathbf{0}$, c'est-à-dire

$$(\mathbf{A}^T\mathbf{A}) \, \mathbf{x}_{MC} = \mathbf{A}^T\mathbf{y} \, . \tag{2.19}$$

On voit donc que $\mathbf{x}_{MC}$ est la solution exacte du système linéaire carré (2.19) (ces équations sont souvent appelées *les équations normales*). Contrairement au système linéaire $\mathbf{y} = \mathbf{A}\mathbf{x}$, on montre que les équations normales sont toujours consistentes et admettent donc toujours au moins une solution. Cependant, comme pour toute matrice $\mathbf{A}$ on a $\text{rang}(\mathbf{A}) = \text{rang}(\mathbf{A}\mathbf{A}^T) = \text{rang}(\mathbf{A}^T\mathbf{A})$, la matrice $(\mathbf{A}^T\mathbf{A})$ n'est inversible (et donc la solution de (2.19) n'est unique) que si $\mathbf{A}$ est de rang colonne complet. Plaçons nous pour simplifier dans ce cas de figure ; la solution de (2.18) est alors donnée par

$$\mathbf{x}_{MC} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y} \; . \tag{2.20}$$

Remarquons enfin que dans le cas où $\mathbf{A}$ est carrée et de rang complet, (2.20) devient $\mathbf{x}_{MC} = [\mathbf{A}^{-1}(\mathbf{A}^T)^{-1}]\mathbf{A}^T\mathbf{y} = \mathbf{A}^{-1}\mathbf{y}$, et coïncide donc avec la solution exacte du système $\mathbf{y} = \mathbf{A}\mathbf{x}$, ce qui est logique puisque (2.16) est devenue consistente.

Résolution pratique

Il reste à calculer $\mathbf{x}_{MC}$ numériquement. Il se trouve qu'il y a mieux à faire que d'implanter directement (2.20) en calculant le produit $\mathbf{A}^T\mathbf{A}$, en l'inversant, puis finalement en calculant le produit (2.20). Considérons en effet la factorisation QR de la matrice $\mathbf{A}_{q\times p}$:

$$\begin{bmatrix} \\ \mathbf{A} \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \mathbf{Q} \\ \\ \end{bmatrix} \underbrace{\begin{bmatrix} * & * \\ & \ddots \\ 0 & * \end{bmatrix}}_{\mathbf{R}} \; .$$

Comme les colonnes de $\mathbf{Q}$ sont orthonormées, l'équation $(\mathbf{A}^T\mathbf{A})\mathbf{x} = \mathbf{A}^T\mathbf{b}$ s'écrit $(\mathbf{R}^T \underbrace{(\mathbf{Q}^T\mathbf{Q})}_{\mathbf{I}_p}\mathbf{R})\mathbf{x} = \mathbf{R}^T\mathbf{Q}^T\mathbf{b}$, et donc

$$\mathbf{R}\mathbf{x} = \mathbf{Q}^T\mathbf{b} \; . \tag{2.21}$$

L'intérêt de cette nouvelle écriture peut se voir immédiatement : en utilisant la proposition 2.2.1, le théorème 2.1.3 et la propriété (2.7), on obtient successivement : $\text{cond}_{\|.\|_2}(\mathbf{R}) = \frac{\sigma_{\max}(\mathbf{R})}{\sigma_{\min}(\mathbf{R})} = (\frac{\lambda_{\max}(\mathbf{R}^T\mathbf{R})}{\lambda_{\min}(\mathbf{R}^T\mathbf{R})})^{1/2} = (\frac{\lambda_{\max}(\mathbf{A}^T\mathbf{A})}{\lambda_{\min}(\mathbf{A}^T\mathbf{A})})^{1/2} = (\text{cond}_{\|.\|_2}(\mathbf{A}^T\mathbf{A}))^{1/2} \leq \text{cond}_{\|.\|_2}(\mathbf{A}^T\mathbf{A})$. On voit donc que *le système linéaire (2.21) est toujours mieux conditionné que le système linéaire (2.19)*, et donc qu'il est systématiquement préférable d'implanter (2.21) plutôt que (2.19). Finalement, l'algorithme pratique consiste à effectuer une factorisation QR de $\mathbf{A}$; à calculer le membre de droite de (2.21) ; et enfin à résoudre l'équation (2.21), ce qui est immédiat puisque $\mathbf{R}$ est triangulaire supérieure.

2.4 Algorithmes de calcul de valeurs propres

Il existe de nombreux algorithmes de calcul d'éléments propres d'une matrice $\mathbf{A}$. Il n'est pas question ici de les exposer toutes, ni dans tous leurs détails ; nous allons juste

évoquer brièvement deux méthodes importantes, l'algorithme de Jacobi et l'algorithme QR. Ainsi que nous allons le voir, ces algorithmes ne cherchent pas à calculer le polynôme caractéristique pour ensuite en extraire les racines, mais fournissent les valeurs propres par transformations successives de la matrice $\mathbf{A}$.

2.4.1 Méthode de Jacobi

La méthode de Jacobi est utilisée lorsque l'on cherche toutes les valeurs propres d'une matrice réelle symétrique. Elle se base sur le résultat classique rappelé dans le corollaire 2.1.3 : toute matrice symétrique réelle se diagonalise dans une base orthonormée. La méthode de Jacobi va donc chercher à construire progressivement une matrice orthogonale $\mathbf{Q}$ satisfaisant $\mathbf{Q}^T\mathbf{A}\mathbf{Q} = diag.\{\lambda_i\}$.

Rotations élémentaires de Givens et algorithme de Jacobi

Une telle matrice $\mathbf{Q}$ sera obtenue comme produit de matrices orthogonales élémentaires appelées les "rotations de Givens" ($\mathbf{Q}$ sera bien orthogonale comme produit de matrices orthogonales). Une rotation de Givens $\mathbf{Q}_{p,q}(c,s)$ est une matrice du type

$$\mathbf{Q}_{p,q}(c,s) = \begin{bmatrix} \mathbf{I} & & & & \\ & c & \dots & s & \\ & \vdots & \mathbf{I} & \vdots & \\ & -s & \dots & c & \\ & & & & \mathbf{I} \end{bmatrix} ;$$

$\mathbf{Q}_{p,q}(c,s)$ coïncide donc avec la matrice identité, à l'exception des éléments (p,p), (p,q), (q,p) et (q,q) ; c et s sont respectivement le cosinus et le sinus d'un certain angle θ.

Considérons maintenant une matrice réelle symétrique $\mathbf{A}$. Un calcul simple montre que si l'on prend $\mathrm{cotg}(2\theta) = \frac{a_{q,q}-a_{p,p}}{2a_{p,q}}$, avec $\theta \in]-\pi/4, 0[\cup]0, +\pi/4[$, alors le produit $\mathbf{Q}_{p,q}^T(\cos(\theta),\sin(\theta))\ \mathbf{A}\ \mathbf{Q}_{p,q}(\cos(\theta),\sin(\theta))$ produit une matrice symétrique $\mathbf{A}'$, qui ne diffère de $\mathbf{A}$ que par les $p^{ème}$ et $q^{ème}$ lignes et colonnes, et dont l'élément en position (p,q) est forcé à zéro :

$$\begin{matrix} \\ p \\ \\ q \\ \\ \end{matrix}\!\!\begin{bmatrix} \mathbf{I} & & & & \\ & c & \dots & -s & \\ & \vdots & \mathbf{I} & \vdots & \\ & s & \dots & c & \\ & & & & \mathbf{I} \end{bmatrix}\underbrace{\begin{bmatrix} & \vdots & & \vdots & \\ \dots & a_{p,p} & \dots & a_{p,q} & \dots \\ & \vdots & & \vdots & \\ \dots & a_{q,p} & \dots & a_{q,q} & \dots \\ & \vdots & & \vdots & \end{bmatrix}}_{\mathbf{A}\ symétrique}\begin{bmatrix} \mathbf{I} & & & & \\ & c & \dots & s & \\ & \vdots & \mathbf{I} & \vdots & \\ & -s & \dots & c & \\ & & & & \mathbf{I} \end{bmatrix} = \underbrace{\begin{bmatrix} & \vdots & & \vdots & \\ \dots & a'_{p,p} & \dots & 0 & \dots \\ & \vdots & & \vdots & \\ \dots & 0 & \dots & a'_{q,q} & \dots \\ & \vdots & & \vdots & \end{bmatrix}}_{\mathbf{A}'\ symétrique} \tag{2.22}$$

Le principe de la méthode apparaît alors clairement : partant de la matrice symétrique réelle $\mathbf{A}^{(1)} = \mathbf{A}$, on va forcer à zéro un élément $a_{p,q}$ en dehors de la diagonale. Puis à partir de $\mathbf{A}^{(1)}$ on construira une matrice $\mathbf{A}^{(2)}$ en forçant à zéro un nouvel élément hors-diagonal, et ainsi de suite. Il existe différentes versions de l'algorithme. C'est ainsi qu'en pratique, il est numériquement pertinent, à chaque étape, d'annuler parmi les

éléments non diagonaux celui qui est de plus grand module (la contrepartie étant bien sûr qu'au temps requis pour calculer (2.22), s'additionne le coût de calcul engendré par cette recherche du plus grand élément). Cette version de l'algorithme de Jacobi s'écrit finalement :

Algorithme de Jacobi

- `Initialisation.` $\mathbf{A}^{(1)} \leftarrow \mathbf{A}$;
- `Étape 1.` $|a^{(1)}_{p_1,q_1}| = \max_{i \neq j} |a^{(1)}_{i,j}|$
  $\mathbf{Q}_1^T \mathbf{A}^{(1)} \mathbf{Q}_1 = \mathbf{A}^{(2)}$, $a^{(2)}_{p_1,q_1} = 0$;
- `Étape 2.` $|a^{(2)}_{p_2,q_2}| = \max_{i \neq j} |a^{(2)}_{i,j}|$
  $\mathbf{Q}_2^T \mathbf{A}^{(2)} \mathbf{Q}_2 = \mathbf{A}^{(3)}$, $a^{(3)}_{p_2,q_2} = 0$;
- $\cdots$

Convergence de l'algorithme

On a vu que l'algorithme de Jacobi était une méthode itérative qui fournira en définitive une matrice dont tous les éléments non-diagonaux auront été un-à-un forcés à zéro, c'est-à-dire une matrice diagonale ; il suffira alors de lire la diagonale pour obtenir les valeurs propres.

Mais . . .est-on bien sûr de la convergence de ce processus itératif ? Il existe en fait ici une difficulté par rapport aux algorithmes de résolution de systèmes linéaires. Revenons par exemple sur la méthode de Gauss. Lors de la première étape, tous les éléments (à l'exception du premier) de la première colonne de $\mathbf{A}$ ont été forcés à zéro. Lors de la deuxième étape, la matrice de transformation effectue des combinaisons linéaires entre la deuxième ligne et les lignes $3, 4, \cdots, n$ (cf. éq. (2.12)) ; ces combinaisons linéaires vont créer de nouveaux zéros (dans la deuxième colonne), *sans affecter pour autant les zéros créés à l'étape précédente*.

Ici en revanche, le fait qu'à une étape donnée k $a^{(k)}_{p,q}$ ait été forcé à zéro, *n'implique pas* qu'aux étapes suivantes $m > k$, les éléments $a^{(m)}_{p,q}$ restent égaux à zéro . . .car si tel était le cas, celà signifierait que l'on serait capable, en un nombre fini d'opérations élémentaires, de calculer les valeurs propres d'une matrice de dimension quelconque, c'est-à-dire les racines d'un polynôme (le polynôme caractéristique de $\mathbf{A}$) de degré quelconque ; ce qui contredirait le célèbre théorème d'Abel, qui affirme l'impossibilité de "résoudre par radicaux" les polynômes de degré ≥ 5.

Fort heureusement, il se trouve néanmoins que l'algorithme converge. La clé de cette convergence réside dans le résultat suivant : dans l'équation (2.22),

$$\left\{ \begin{array}{lcl} \sum_{i,j=1}^n a_{i,j}^2 & = & \sum_{i,j=1}^n (a'_{i,j})^2 \ , \ \text{mais} \\ \sum_{i=1}^n a_{i,i}^2 + 2a_{p,q}^2 & = & \sum_{i=1}^n (a'_{i,i})^2 \quad ! \end{array} \right.$$

Dit d'une autre façon, la norme de Frobenius de $\mathbf{A}'$ reste identique à celle de $\mathbf{A}$, mais le poids dans la nouvelle matrice a tendance à se concentrer sur la diagonale principale. En définitive, il est possible de montrer le résultat suivant :

Théorème 2.4.1 *Soit $\mathbf{A}^{(k)}$ la suite de matrices produites par l'algorithme de Jacobi. Alors $\lim_{k\to\infty} \mathbf{A}^{(k)} = diag.(\lambda_i)$, où $\{\lambda_i\}$ sont les valeurs propres de $\mathbf{A}$.*

En vertu de la discussion précédente, il est clair que la convergence n'aura lieu *qu'à l'infini*. En pratique bien sûr, il est nécessaire d'arrêter l'algorithme après un certain nombre (nécessairement fini) d'itérations : à d'éventuels problèmes de *conditionnement*, à la propagation d'une étape à l'autre d'*erreurs d'arrondi*, se rajoutent donc des **erreurs de troncature** : on voit ainsi apparaître un troisième phénomène, inévitable dans toute méthode itérative.

Terminons par une dernière remarque. Soit $\mathbf{Q}_i$ la rotation de Givens utilisée à la $i^{ème}$ étape de l'algorithme. Au bout de k itérations, on a

$$\underbrace{(\mathbf{Q}_k^T \cdots \mathbf{Q}_1)}_{\mathbf{Q}_{1:k}^T} \mathbf{A} \underbrace{(\mathbf{Q}_1 \cdots \mathbf{Q}_k)}_{\mathbf{Q}_{1:k}} = \mathbf{A}^{(k)} \approx \text{diag.}(\lambda_i)$$

Aussi, si $\lambda_i \neq \lambda_j$, $i \neq j$, $\mathbf{Q}_{1:k}$ converge vers un ensemble orthonormal de vecteurs propres de $\mathbf{A}$, et constitue donc une approximation d'une base orthonormée de vecteurs propres de $\mathbf{A}$.

2.4.2 Algorithme QR

Nous évoquons maintenant le très célèbre algorithme QR, qui peut être utilisé lorsque l'on recherche toutes les valeurs propres d'une matrice quelconque (non nécessairement symétrique).

L'algorithme QR est basé sur la factorisation QR (cf. le Théorème 2.3.3) : Toute matrice réelle $\mathbf{A}$ peut se factoriser en $\mathbf{A} = \mathbf{QR}$, où $\mathbf{Q}$ est une matrice orthogonale et $\mathbf{R}$ est triangulaire supérieure. L'algorithme QR est une méthode itérative qui consiste à écrire la factorisation QR de $\mathbf{A}$, puis à calculer le produit $\mathbf{RQ}$ (noter que l'on a interverti l'ordre des facteurs) ; on calcule alors la factorisation QR de la matrice obtenue, puis le produit des facteurs dans l'ordre inverse, et ainsi de suite :

Algorithme QR

- `Initialisation.` $\mathbf{A}^{(1)} \leftarrow \mathbf{A}$;
- `Étape 1.` $\mathbf{A}^{(1)} = \mathbf{Q}^{(1)}\mathbf{R}^{(1)}$, $\mathbf{A}^{(2)} = \mathbf{R}^{(1)}\mathbf{Q}^{(1)}$;
- `Étape 2.` $\mathbf{A}^{(2)} = \mathbf{Q}^{(2)}\mathbf{R}^{(2)}$, $\mathbf{A}^{(3)} = \mathbf{R}^{(2)}\mathbf{Q}^{(2)}$;
- $\cdots$

A l'étape k, $\mathbf{A}^{(k)} = \mathbf{Q}^{(k)}\mathbf{R}^{(k)}$. Par conséquent,

$$\mathbf{A}^{(k+1)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)} = (\mathbf{Q}^{(k)})^{-1}\mathbf{A}^{(k)}\mathbf{Q}^{(k)} : \tag{2.23}$$

cette dernière égalité signifie que $\mathbf{A}^{(k+1)}$ et $\mathbf{A}^{(k)}$ sont semblables et donc ont les mêmes valeurs propres : *les valeurs propres de* $\mathbf{A}^{(k)}$ *sont donc identiques* $\forall$ k *aux valeurs propres de* $\mathbf{A}$. On montre enfin que sous certaines conditions, les matrices $\mathbf{A}^{(k)}$ deviennent triangulaires supérieures lorsque $k \to \infty$:

$$\lim_{k\to\infty} \mathbf{A}^{(k)}(i,j) = 0 \text{ si } j < i$$

(ce qui n'implique pas que la suite $\{\mathbf{A}^{(k)}\}$ converge, car les valeurs situées dans la partie sur-diagonale de $\mathbf{A}^{(k)}$ peuvent ne pas admettre de limite). Par conséquent il suffira de lire les éléments diagonaux de $\mathbf{A}^{(k)}$ (pour k "suffisamment grand") pour avoir les valeurs propres de $\mathbf{A}$.

Chapitre 3

Approximations Stochastiques

3.1 Introduction

La possibilité de simulation des réalisations de variables aléatoires peut s'avérer être un puissant outil de calcul. L'exemple le plus naturel est celui de l'utilisation de la loi des grands nombres pour le calcul d'intégrales : la moyenne des réalisations d'une variable aléatoire simulée converge vers l'espérance, qui est une intégrale. Dans certaines situations les méthodes stochastiques sont en concurrence avec les méthodes d'analyse numérique «déterministes», dans d'autres, elles constituent le seul outil adapté. A titre d'exemple, comment calculer la probabilité d'avoir «pile» dans un lancement d'une pièce de monnaie voilée ? Le calcul théorique est inextricable alors que la possibilité d'effectuer un certain nombre de lancement de cette pièce permet l'estimation, avec la précision aussi grande que l'on souhaite, de ladite probabilité. L'objectif premier de ce chapitre est de présenter quelques méthodes de base de calcul de certaines approximations stochastiques parmi les plus répandues. Des techniques plus récentes, comme le filtrage particulaire ou l'algorithme de Hasting-Metropolis, sont également brièvement décrites.

Quelques remarques préliminaires Faites bien attention que

- simuler selon une loi de probabilité ne consiste surtout pas à calculer cette loi en un point ;
- simuler une variable aléatoire ne signifie pas choisir la valeur la plus probable.

3.2 Intégration par la méthode de Monte Carlo

Définition 3.2.1 *On appelle méthode de Monte-Carlo toute méthode visant à calculer une valeur numérique en utilisant des procédés aléatoires.*

Le nom de ces méthodes fait allusion aux jeux de hasard pratiqués à Monte-Carlo et leur développement s'est effectué au cours de la Seconde Guerre mondiale et des recherches sur la fabrication de la bombe atomique pour résoudre des équations aux dérivées partielles. Ces méthodes sont très employées pour le calcul des intégrales

en dimensions plus grandes que 1 (surfaces, volumes, etc...) et une application importante est le filtrage adaptatif qui permet de faire de la prédiction en finance, de la trajectographie, ...

L'intégration par la méthode de Monte Carlo est fondée sur la loi des grands nombres et le théorème de transfert : pour une densité de probabilité f sur $\mathbb{R}^n$ et h une application quelconque sur $\mathbb{R}^n$ telle que fh soit intégrable sur $\mathbb{R}^n$, nous avons

$$\frac{h(X_1) + \ldots + h(X_k)}{k} \xrightarrow{k\to\infty} \mathbb{E}[h(X_1)] = \int_{\mathbb{R}^n} h(x)f(x)dx \tag{3.1}$$

où $X_1, \ldots, X_k$ sont des variables aléatoires indépendantes dont la loi commune admet f pour densité. On en déduit que l'intégrale $\int_{\mathbb{R}^n} h(x)f(x)dx$ pourra être approchée par $\frac{1}{k}\sum_{i=1}^{k} h(X_i)$. Cela signifie qu'il «suffit » donc de tirer suffisamment d'échantillons selon f et de calculer la somme pour estimer l'intégrale.
La convergence (3.1) a lieu presque sûrement ; de plus, lorsque les variables $h(X_1), \ldots, h(X_k)$ sont de carré intégrable on peut appliquer le théorème central limite et avoir des précision sur sa vitesse. La convergence (3.1) peut déjà être utilisée dans un grand nombre de situations ; cependant, elle ne donne pas nécessairement des résultats plus intéressants que des méthodes d'intégration numérique.

Remarque. Pour $X_1 = x_1, \ldots, X_k = x_k$, la quantité $\frac{1}{k}\sum_{i=1}^{k} h(X_i)$ peut être vue comme une intégrale par rapport à la mesure «empirique» définie par cette réalisation, cette dernière consistant à associer à chaque ensemble $B \subset \mathbb{R}^n$ la proportion des $x_1, \ldots, x_k$ qu'il contient.

Exemple 3.2.1 *Une illustration simple de l'emploi de cette méthode est l'estimation de l'aire d'une surface. Prenons l'exemple de la figure 3.1.*

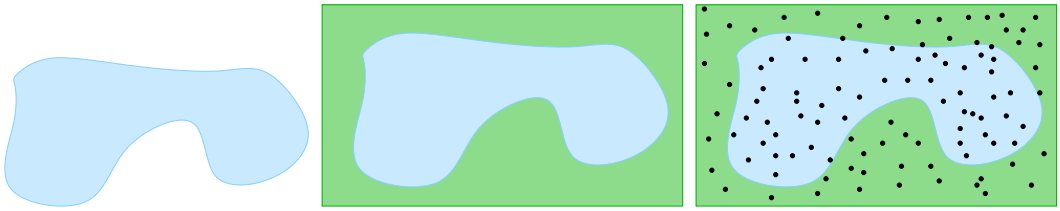


FIGURE 3.1 – Illustration de la méthode de Monte-Carlo pour estimer une surface

La technique consiste à englober la surface dans un rectangle dans lequel on tire des points uniformément répartis. Ensuite la méthode de Monte Carlo approxime l'aire de

la façon suivante :

$$\frac{\textit{Nb de points dans } S}{\textit{Nb total de points}} \approx \frac{\textit{Aire de } S}{\textit{Aire du rectangle}}$$

Dans ce cas, f est la loi uniforme sur le rectangle et h est la fonction indicatrice de l'appartenance à S.
Par le même procédé, il est possible d'estimer le nombre π en prenant le rectangle $[0,1]\times[0,1]$ et le disque inscrit pour S. ■

Une des possibilités d'amélioration de la vitesse de convergence de (3.1) est de noter que pour une densité de probabilité quelconque g l'intégrale d'intérêt $\int_{\mathbb{R}^n} h(x)f(x)dx$ peut aussi s'écrire $\int_{\mathbb{R}^n} h(x)f(x)dx = \int_{\mathbb{R}^n} \frac{h(x)f(x)g(x)}{g(x)}dx$, et donc pour une suite $X_1,\ldots,X_k$ de variables aléatoires indépendantes dont la loi commune admet g pour densité, nous avons

$$\frac{\frac{h(X_1)f(X_1)}{g(X_1)}+\ldots+\frac{h(X_k)f(X_k)}{g(X_k)}}{k} \xrightarrow{k\to\infty} \int_{\mathbb{R}^n} h(x)f(x)dx \tag{3.2}$$

En effet, par la loi des grands nombres $\frac{\frac{h(X_1)f(X_1)}{g(X_1)}+\ldots+\frac{h(X_k)f(X_k)}{g(X_k)}}{k}$ tend vers $\mathbb{E}\left[\frac{h(X_1)f(X_1)}{g(X_1)}\right] = \int_{\mathbb{R}^n}\frac{h(x)f(x)}{g(x)}g(x)dx = \int_{\mathbb{R}^n} h(x)f(x)dx$, la première égalité ayant lieu en vertu du théorème de transfert. Il est ainsi possible de rechercher des densités g pour lesquelles la vitesse de la convergence vers $\int_{\mathbb{R}^n} h(x)f(x)dx$ est la plus intéressante. On a alors le résultat suivant :

Proposition 3.2.1 *La densité $g^*(x) = \frac{|h(x)|f(x)}{\int_{\mathbb{R}^n}|h(t)|f(t)dt}$ procure la variance minimum pour les estimateurs de type (3.2).*

Exemple 3.2.2 *Considérons une suite de variables aléatoires réelles $X_1,\ldots,X_n,\ldots$ dont la loi admet, pour chaque $n\in\mathbb{N}$, une densité de la forme*

$$f(x_1,\ldots,x_n) = f(x_1)f(x_2|x_1)f(x_3|x_2)\ldots f(x_n|x_{n-1}) \tag{3.3}$$

Une telle suite est appelée «chaîne de Markov». On montre que l'on a la factorisation (3.3) (à comparer avec (3.9)) si et seulement si pour tout $n\in\mathbb{N}$ la loi de X_n conditionnelle à $X_1,\ldots,X_{n-1}$ est égale à sa loi conditionnelle à X_{n-1}. Une telle suite est également dite «en mémoire d'ordre 1» ; en effet, si l'on observe $X_{n-1}=x_{n-1}$, la loi de X_n ne dépend que de x_{n-1} et l'information contenue dans $x_1,\ldots,x_{n-2}$, qui est la mémoire de ce qui s'est passé avant l'instant $n-1$, n'a aucune influence sur le comportement de X_n. Si l'on suppose que $n-1$ est l'instant présent, les instants $1,\ldots,n-2$ sont le passé, et l'instant n est le futur, on dit aussi que «le futur et le passé sont

indépendants conditionnellement au présent». Attention, si x_{n-1} n'est pas observé (on supprime le conditionnement), X_n n'est plus indépendant de $X_1, \ldots, X_{n-1}$.
Supposons que $X_1, \ldots, X_n, \ldots$ n'est pas observable et l'on observe une suite $Y_1, \ldots, Y_n, \ldots$ telle que pour tout $n \in \mathbb{N}$, la loi de $(Y_1, \ldots, Y_n)$ conditionnelle à $(X_1, \ldots, X_n) = (x_1, \ldots, x_n)$ est donnée par

$$f(y_1, \ldots, y_n | x_1, \ldots, x_n) = f(y_1|x_1) f(y_2|x_2) \ldots f(y_n|x_n) \tag{3.4}$$

Le problème, admettant de multiples applications, est alors d'estimer $X_1, \ldots, X_n, \ldots$ à partir de $Y_1, \ldots, Y_n, \ldots$. Une approche particulière consiste à calculer $f(x_n|y_1, \ldots, y_n)$ à partir de $f(x_{n-1}|y_1, \ldots, y_{n-1})$ et y_n. Dans une telle approche (appelée filtrage adaptatif) on considère généralement deux étapes :

$$\begin{aligned} f(x_n|y_1, \ldots, y_{n-1}) &= \int_{\mathbb{R}} f(x_{n-1}, x_n | y_1, \ldots, y_{n-1}) dx_{n-1} \\ &= \int_{\mathbb{R}} f(x_n|x_{n-1}) f(x_{n-1}|y_1, \ldots, y_{n-1}) dx_{n-1} \end{aligned} \tag{3.5}$$

$$f(x_n|y_1, \ldots, y_n) = \frac{f(x_n, y_n|y_1, \ldots, y_{n-1})}{f(y_n|y_1, \ldots, y_{n-1})} = \frac{f(x_n|y_1, \ldots, y_{n-1}) f(y_n|x_n)}{f(y_n|y_1, \ldots, y_{n-1})} \tag{3.6}$$

Dans des cas simples, typiquement linéaires et Gaussiens, (3.6) admet une solution analytique (célèbre filtre de Kalman). Dans le cas général on peut approcher l'intégration dans (3.5) en utilisant des tirages stochastiques. De telles méthodes, dites filtrage particulaire en référence aux points, ou particules, obtenues par les tirages, sont actuellement très à la mode et font objet de nombreuses recherches. ■

3.3 Générations des variables aléatoires

Nous considérons le problème de génération des réalisations d'une variable aléatoire réelle (à valeurs dans l'ensemble des nombres réels $\mathbb{R}$) dont la loi admet une densité f par rapport à la mesure de Lebesgue. Pour simplifier, nous dirons que l'on « simule f».

3.3.1 Fonction de répartition inversible

Supposons que l'on sache simuler des réalisations d'une variable notée U, de la loi uniforme sur $[0, 1]$ (possibilité offerte par la plupart des ordinateurs). Supposons que la fonction de répartition F correspondant à f (donc $F(t) = \int_{-\infty}^{t} f(w)dw$) est inversible. Alors la variable aléatoire $V = F^{-1}(U)$ suit la loi donnée par f. En effet, la fonction de répartition de $V = F^{-1}(U)$ s'écrit :

$$P(V \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = \int_0^{F(x)} dt = F(x).$$

Cette technique ne pourra évidemment être utilisée que si l'on connaît une formulation explicite de la fonction F^{-1}.

À titre d'exemple, il est ainsi possible de simuler les lois de densités exponentielles. En effet, la fonction de répartition d'une loi exponentielle à valeurs dans $\mathbb{R}^+$ admettant pour densité $f(x) = \lambda e^{-\lambda x}\mathbb{1}_{[0,+\infty[}(x)$ est $F(x) = 1 - e^{-\lambda x}\mathbb{1}_{[0,+\infty[}(x)$. Donc si U suit la loi uniforme sur $[0,1]$, la variable aléatoire $F^{-1}(U) = -\dfrac{\log(1-U)}{\lambda}$ suit la loi exponentielle de densité $f(x) = \lambda e^{-\lambda x}\mathbb{1}_{[0,+\infty[}(x)$. Notons que les lois exponentielles interviennent beaucoup dans des phénomènes d'attentes et, dans certaines études des systèmes complexes, la possibilité de leur simulation directe peut s'avérer très utile.

Pour plus de précisions, nous reportons le lecteur au cours de probabilité de Première Année.

3.3.2 Loi de Gauss et lois associées

La loi de Gauss intervient fréquemment en applications ayant trait aux traitements des signaux ou des images. En effet, les divers «bruitages» ont souvent modélisés par des réalisations des variables gaussiennes. Cette modélisation est souvent justifiée par le théorème central limite selon lequel, grosso modo, une somme «grande» des quantités aléatoires indépendantes tend en loi vers une distribution Gaussienne. Cependant, même lorsque ce type de justification ne peut être mis en avant, la modélisation Gaussienne est quand même souvent utilisée à cause des facilités de calcul qu'elle offre. La loi de Gauss ne fait pas partie de la famille du paragraphe précédent ; en effet, sa fonction de répartition n'est pas exprimable analytiquement. Cependant, il est possible de trouver un **changement de variables** qui transforme deux variables indépendantes U_1, U_2, chacune étant de la loi uniforme sur $[0,1]$, en un couple de variables gaussiennes indépendantes (cf. le cours de probabilité de Première Année). En particulier, on peut montrer que la loi de la variable

$$X = \sqrt{-2log(U_1)}cos(2\pi U_2) \tag{3.7}$$

est la loi normale de moyenne 0 et de variance 1. En conséquence

$$\sigma\sqrt{-2log(U_1)}cos(2\pi U_2) + m \tag{3.8}$$

suit $\mathcal{N}(m, \sigma^2)$ (qui désigne la loi normale de moyenne m et de variance σ^2).
Notons que la possibilité de simuler une variable aléatoire gaussienne réelle implique la possibilité de simuler tout vecteur gaussien. En effet, soit $X = (X_1, \ldots, X_n)$ un vecteur gaussien et f la densité de sa loi sur $\mathbb{R}^n$. Nous pouvons écrire

$$f(x) = f(x_1, \ldots, x_n) = f(x_1)f(x_2|x_1)f(x_3|x_1, x_2)\ldots f(x_n|x_1, x_2, \ldots, x_{n-1}) \tag{3.9}$$

et il est connu que chacune des densités conditionnelles figurant dans le produit à droite de l'égalité (3.9) est une densité gaussienne. On simule ainsi les composantes du vecteur de proche en proche : on commence par $X_1 = x_1$ en utilisant (3.8), ensuite on calcule la moyenne et la variance de $f(x_2|x_1)$ et on simule $X_2 = x_2$ (toujours par (3.8)), et ainsi de suite ...
Il existe une autre manière de simuler un vecteur gaussien quelconque $X = (X_1, \ldots, X_n)$ de moyenne $m \in \mathbb{R}^n$ et de matrice de covariance C. On sait générer un vecteur gaussien

Y centré et de matrice de covariance l'identité, *i.e.* dont toutes les composantes sont indépendantes (cf. cours de probabilité). Les propriétés des vecteurs gaussiens nous disent que le vecteur $Z = AY + m$ est aussi gaussien avec une espérance $\mathbb{E}(Z) = A\mathbb{E}(Y) + m = m$ et une matrice de covariance de $C_Z = AC_YA^T = AA^T$. On en déduit que pour simuler X à partir d'une réalisation de Y, il suffit d'appliquer le changement de variable proposé en choisissant A parmi les racines de C, *i.e.* $C = AA^T$ (numériquement, il sera intéressant de choisir A triangulaire : voir la décomposition de Cholesky dans le chapitre 2).

Un certain nombre de lois classiques sont des lois des variables aléatoires liées aux lois normales de manière déterministe : $V = g(X_1, \ldots, X_n)$, avec $X = (X_1, \ldots, X_n)$ un vecteur gaussien et g une application de $\mathbb{R}^n$ dans $\mathbb{R}$. Bien entendu, de telles lois sont immédiatement simulables à partir des simulations des lois gaussiennes.

3.3.3 Méthode des lois marginales

Il est parfois commode - voir indispensable - de considérer la loi à simuler comme étant la loi marginale d'un couple de variables aléatoires.
Considérons l'exemple de la loi de probabilité de la variable «taille» d'un individu pris au hasard dans une foule. L'individu peut être un homme ou une femme, les deux populations présentant des tailles distribuées selon les Gaussiennes f_H et f_F. On sait donc simuler la taille d'un homme pris au hasard, ainsi que la taille d'une femme choisie au hasard. Comment simuler la taille d'un individu pris dans une foule comportant des hommes et des femmes ? On introduit une variable aléatoire X prenant ses valeurs dans l'ensemble $\Omega = \{H, F\}$, avec $P(X = H) = \Pi(H)$ (proportion des hommes dans la foule), et $P(X = F) = \Pi(F)$. Si Y est la variable aléatoire modélisant la taille d'un individu pris au hasard, f_H est la loi de Y conditionnelle à $X = H$, et f_F est la loi de Y conditionnelle à $X = F$. Il en résulte que la loi de (X, Y) est donnée par $\Pi(x)f_x(y)$, et donc loi de Y est la loi marginale de celle de (X, Y), obtenue en sommant sur les x : $f(y) = \Pi(H)f_H(y) + \Pi(F)f_F(y)$. On voit que si l'on sait simuler $(X, Y) = (x, y)$ on saura simuler $Y = y$ (il suffit de ne regarder que $Y = y$). Comment simuler $(X, Y) = (x, y)$? On simule d'abord $X = x$; et ensuite $Y = y$ (cette dernière simulation est faite selon f_H si $x = H$, et elle est faite selon f_F si $x = F$).

C'est une démarche générale ; on sait parfois simuler la réalisation d'une variable Y conditionnellement à une autre variable X. Si l'on sait simuler X, on saura simuler Y ; en effet, la loi de Y est la loi marginale de la loi de (X, Y), cette dernière étant simulable.

3.3.4 Méthode d'acceptation-rejet

La méthode d'acceptation-rejet peut être utile lorsque l'on ne peut pas utiliser la fonction de répartition inverse ou s'il n'y a pas de changement de variables connu aboutissant à la densité que l'on cherche à simuler. Dans ce cas, nous allons exploiter la seule connaissance de la densité de probabilité f et l'«imiter» avec une densité g que l'on sait simuler.

Considérons une densité à simuler f, une densité simulable g, et une constante connue M pour laquelle

$$f(x) \leq Mg(x), \quad \forall x \tag{3.10}$$

Alors f peut être simulée de la manière suivante :

1. Simuler $Z = z$ selon g, et $U = u$ selon la loi uniforme sur $[0, 1]$;
2. Prendre $x = z$ si $u \leq \dfrac{f(z)}{Mg(z)}$; retourner en (1) sinon.

Ainsi, si $z_1, \ldots, z_n$ est une suite des valeurs simulées selon g, certaines seront «acceptées» et d'autres seront «rejetées», d'où l'appellation de la méthode.

Pour comprendre l'origine de la méthode, considérons l'exemple d'une densité f à support borné $[a, b]$; prenons g la densité uniforme sur $[a, b]$ et M tel que $\frac{M}{b-a} = c \geq \max(f)$. Nous sommes bien dans les conditions d'emploi de la méthode d'acceptation-rejet. Celle-ci peut alors s'interprêter de la façon suivante (cf. la figure 3.2(a)) : on tire des points uniformément sur le rectangle $[a, b] \times [0, c]$ et on ne conserve que les points sous la courbe $f(x)$. Les abscisses des points conservés seront des réalisations selon f.

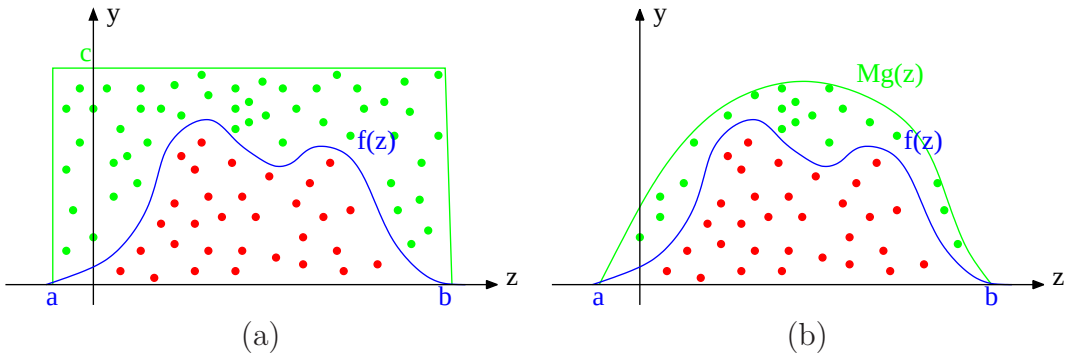


FIGURE 3.2 – Illustration de la méthode d'acceptation-rejet

Généralisons et remplaçons la loi uniforme par une densité quelconque simulable g. Plaçons nous dans les conditions de la méthode, soit $f \leq Mg$. La figure 3.2(b) illustre la méthode qui consiste à

(i) tirer des points uniformément sous la courbe $Mg(x)$;

(ii) ne conserver que les points sous la courbe $f(x)$.

Les abscisses des points conservés seront des réalisations selon f. Cette affirmation sera démontrée en séance de travaux dirigés (voir TD-4). Par ailleurs le point (i) est réalisé de la façon suivante :

— tirer z selon g ;

— tirer y selon la loi uniforme sur $[0, Mg(z)]$, *i.e.* tirer u selon la loi uniforme sur $[0, 1]$ et poser $y = uMg(z)$.

Puis le point (ii) doit vérifier si $uMg(z) = y \leq f(z)$ et dans ce cas, on pose $x = z$. Alors, on retrouve bien l'algorithme présenté au début du paragraphe.

Notons la grande généralité de cette méthode. En particulier, la densité f peut être connue à un facteur multiplicatif près. En effet, l'algorithme sera utilisable dès lors que le rapport $\frac{f(z)}{Mg(z)}$ est calculable. Enfin, la probabilité d'acceptation d'une réalisation selon g étant $\frac{1}{M}$, on recherchera une fonction g simulable pour laquelle la majoration (3.10) est vraie pour M aussi petit que possible.

3.4 Méthodes de Monte Carlo par Chaînes de Markov (MCMC)

3.4.1 Cas discret

La problématique abordée dans ce paragraphe est la même que celle du paragraphe 3.3 : génération des réalisations d'une variable aléatoire dont la loi admet une densité f par rapport une certaine mesure. La différence est que les variables considérées prennent leurs valeurs dans des espaces trop compliquées (souvent, simplement trop vastes) pour que les méthodes du paragraphe 3.3 puissent être appliquées. L'idée de base est de construire une chaîne de Markov homogène (signifiant que les lois conditionnelles $p(x_n|x_{n-1})$, que l'on appelle également transitions, ne dépendent pas de n) telle que :

1. on sait simuler ses réalisations ;
2. la loi de probabilité donnée par f est la loi marginale limite de la chaîne.

Notons bien que le fait que les transitions ne dépendent pas de n n'implique pas que les lois marginales ne dépendent pas de n, ces dernières se calculant à partir des transitions et de la loi $p(x_1)$ de la première variable X_1.
Nous nous intéressons dans ce premier sous-paragraphe aux chaînes de Markov à états discrets. On considère donc une suite de variables aléatoires $X_1, \ldots, X_n, \ldots$ prenant leurs valeurs dans l'espace fini d'état $\Omega = \{e_1, \ldots, e_k\}$. Pour chaque $n \in \mathbb{N}$, la loi de $(X_1, \ldots, X_n)$ s'écrit

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2|x_1)p(x_3|x_2)\ldots p(x_n|x_{n-1}) \tag{3.11}$$

où $p(x_1)$ est une probabilité sur Ω, et $p(x_2|x_1), \ldots, p(x_n|x_{n-1})$ ont des probabilités conditionnelles (dites aussi des transitions), qui sont données par des matrices de transition carrées de dimension k. Notons bien que nous utilisons la même lettre p pour simplifier l'écriture, mais les $k-1$ matrices de transition définissant $p(x_2|x_1), \ldots, p(x_n|x_{n-1})$ sont, dans le cas général, différentes.
Supposons que ces matrices sont égales (la chaîne est dite homogène) à une matrice M (sur la ligne i de la matrice on met les probabilités de $e_1, \ldots, e_k$ conditionnelles à e_i). La probabilité $p(x_1, \ldots, x_n)$ donnée par (3.11) est ainsi déterminée par $p(x_1)$, que nous noterons Π, et par la matrice M. Considérons le problème de l'évolution des lois de X_n lorsque $n \in \mathbb{N}$ varie. La loi de X_1 est $\Pi = (\Pi_1, \ldots, \Pi_k)$, la loi de X_2 est $p(x_2) = \sum_{1 \leq i \leq k} p(x_1 = e_i, x_2) = \sum_{1 \leq i \leq k} p(x_1 = e_i)p(x_2|x_1 = e_i)$. Si l'on considère $p(x_2)$ comme un vecteur ligne à k composantes, on note qu'il peut s'écrire ΠM (produit à

gauche, Π étant un vecteur ligne). En faisant le même raisonnement (la loi de X_3 est obtenue à partir de la loi de X_2 de la même manière que la loi de X_2 est obtenue à partir de celle de X_1), on note que la loi de X_3 est $(\Pi M)M = \Pi M^2$. De proche en proche, on peut affirmer que la loi de X_n est ΠM^{n-1}.
Une question intéressante est de savoir si ΠM^{n-1} converge vers une limite indépendante de Π et il existe différents résultats précisant des conditions suffisantes pour qu'il en soit ainsi. Pour l'application qui nous intéresse ici mentionnons les conditions suivantes :

Proposition 3.4.1 *Sous les conditions*

(i) les termes diagonaux de M sont non nuls ;

(ii) pour tous (e_i, e_j), la probabilité de passer de e_i à e_j en un temps fini est non nulle,

ΠM^{n-1} *converge vers une limite indépendante de* Π.

Notons également

Proposition 3.4.2 *Si ΠM^{n-1} converge vers une limite L indépendante de Π, alors $LM = L$ (la limite est un vecteur propre à gauche de la matrice de transition).*

En utilisant ces deux propositions, il suffit donc de vérifier (i) et (ii) de la proposition 3.4.1, et trouver un vecteur propre à gauche de M.

Dans les méthodes de Monte Carlo par Chaînes de Markov (MCMC) considérées ici le problème est le suivant. On dispose d'une loi de probabilité L trop compliquée pour pouvoir être simulée directement. On cherche alors à construire une chaîne de Markov homogène (de matrice de transition M) vérifiant deux conditions suivantes :

(i) les transition sont faciles à simuler ;

(ii) L est la limite des lois des X_n.

Exemple 3.4.1 *Afin d'illustrer l'intérêt des chaînes de Markov en approximation stochastique considérons l'exemple suivant. Soit un ensemble de points disposés en carré de coté m (un réseau simple). Chaque point peut être, de manière aléatoire, 0 ou 1. L'ensemble des réalisation possibles d'un tel réseau est $\Omega = \{0,1\}^{m\times m}$. Comment simuler les réalisations d'un tel réseau ? On voit que très rapidement le cardinal de Ω est trop grand pour utiliser les méthodes du paragraphe 3.3.*
En considérant les couples (s,t) de points voisins (horizontalement ou verticalement), soit une probabilité L définie sur $\Omega = \{0,1\}^{m\times m}$ par

$$L(x) = \gamma \exp\left[\sum_{(s,t)} \varphi_{(s,t)}(x_s, x_t)\right] \tag{3.12}$$

La constante γ est inconnue et ne peut pas être calculée ; cependant, pour chaque point s la loi de X_s conditionnelle aux autres X_t est calculable : si V_s désigne le voisinage de

s, on a

$$L(x_s|x_t, t \in V_s) = \frac{\exp\left[\sum_{t\in V_s} \varphi_{(s,t)}(x_s, x_t)\right]}{\exp\left[\sum_{t\in V_s} \varphi_{(s,t)}(x_s = 0, x_t)\right] + \exp\left[\sum_{t\in V_s} \varphi_{(s,t)}(x_s = 1, x_t)\right]} \tag{3.13}$$

L'idée est alors de balayer le carré (ligne par ligne par exemple) et faire un tirage, sur chaque point, selon la très simple loi conditionnelle (3.13). Ce faisant, on utilise des transitions dans l'espace $\Omega = \{0,1\}^{m\times m}$ *: bien que ce dernier soit excessivement riche, les transitions sont très simples car la matrice de transition est très creuse (sur une ligne de* $2^{m\times m}$ *éléments, seuls 2 éléments sont non nuls). Par ailleurs, on montre que les hypothèses des Propositions 3.4.1 et 3.4.2 sont vérifiées, et donc* L *est bien la loi limite d'une chaîne de Markov ainsi construite.*

À titre d'exemple, on présente sur la figure 3.3 des tirages effectués avec $m = 128$ *et*

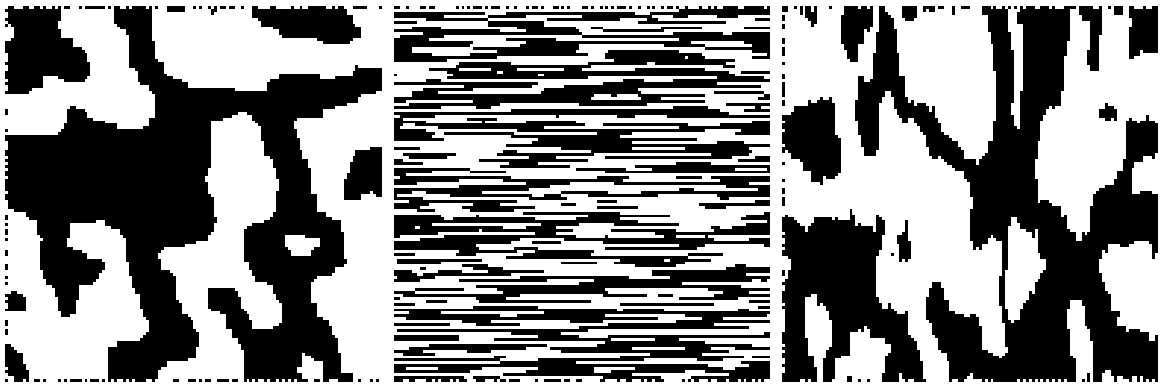


FIGURE 3.3 – Réalisations de $X = (X_s)_{s\in S}$, avec S un carré de côté comportant $m = 128$ points, obtenues avec 40 balayages.

quarante balayages du carré, soit $40 \times 128 \times 128$ *tirages élémentaires. Malgré ce nombre élevé le tirage final, dont trois versions obtenues pour des paramètres différents sont visualisées sur la figure 3.3, apparaît comme quasi instantané et prend moins de temps que le lancer d'une pièce de monnaie.* ■

3.4.2 Cas continu

Considérons le cas plus général où l'espace des états est $\mathbb{R}^m$. Dans une chaîne de Markov vérifiant l'écriture (3.3) la matrice de transitions est remplacée par le noyau de transitions, qui est une application $N : \mathbb{R}^m \times \mathcal{B}(\mathbb{R}^m) \to [0,1]$ vérifiant

(N1) pour tout $x \in \mathbb{R}^m$, l'application $B \in \mathcal{B}(\mathbb{R}^m) \mapsto N(x,B)$ est une probabilité sur $\mathcal{B}(\mathbb{R}^m)$;

(N2) pour tout $B \in \mathcal{B}(\mathbb{R}^m)$, l'application $x \in \mathbb{R}^m \mapsto N(x,B)$ est mesurable.

Cela veut dire, plus simplement, que chaque $X_i = x_i \in \mathbb{R}^m$ définit la loi de X_{i+1} sur $\mathbb{R}^m$.
Comme précédemment, la loi d'une chaîne de Markov homogène est entièrement déterminée par la loi de la première variable X_1 et le noyau de transitions.
Pour une densité de probabilité f trop compliquée pour pouvoir être simulée directement, le problème est donc de rechercher un noyau (qui devient une matrice de transition dans le cas discret) exploitable et tel que la loi limite soit la loi donnée par f. L'étude des chaînes de Markov où l'espace des états est $\mathbb{R}^m$ est notablement plus compliquée que celle des cas où cet espace est discret. Nous nous contentons d'énoncer deux manière de rechercher un noyau de transition convenable.

Algorithme de Hasting-Metropolis

Soit f la loi que l'on souhaite simuler. Soit une famille des densités conditionnelles simulables $q(y|x)$, avec x et y dans $\mathbb{R}^m$. On suppose que l'une de deux conditions suivantes est vérifiée :

(i) $q(\cdot|x)$ est disponible analytiquement (à une constante indépendante de x près) ;

(ii) q est symétrique, soit $q(y|x) = q(x|y)$.

À $X_n = x_n$ donné, on simule alors X_{n+1} de la manière suivante (ce qui donne le noyau recherché) :

1. générer $Y_n = y_n$ selon $q(y|x_n)$;
2. poser $x_n = \begin{cases} y_n & \text{avec la probabilité} \quad \rho(x_n, y_n) \\ x_n & \text{avec la probabilité} \quad 1 - \rho(x_n, y_n) \end{cases}$ où

$$\rho(x_n, y_n) = \min\left[\frac{f(y_n)q(x_n|y_n)}{f(x_n)q(y_n|x_n)}, 1\right]. \tag{3.14}$$

Un processus ainsi obtenu est bien une chaîne de Markov (la loi de X_{n+1} conditionnelle à $X_1 = x_1, \ldots, X_n = x_n$ ne dépend que de x_n).
On montre alors que la loi marginale de la chaîne (X_n) converge bien vers la loi donnée par la densité f. Le principe de la démonstration est analogue aux démarches utilisées dans le cas discret ci-dessus : on montre d'abord qu'il existe une loi limite indépendante de la loi de X_1 (régularité de la chaîne), et ensuite on montre que f est «invariante » (ce qui correspond aux vecteurs propres à gauche ci-dessus) au sens suivant :

$$\text{Pour tout } B \in \mathcal{B}(\mathbb{R}^m), \text{ on a } \int_B f(t)dt = \int_{\mathbb{R}^m} f(t)N(t, B)dt \tag{3.15}$$

Remarque Remarquons la très grande généralité de cet algorithme. Tout d'abord, la densité f peut n'être connue qu'à une constante multiplicative près (seul le rapport intervient dans (3.14)). Ensuite on a une grande latitude de choix dans les densités conditionnelles q.

Échantillonneur de Gibbs

Supposons que l'on peut décomposer le vecteur $x \in \mathbb{R}^m$ en k composantes $x = (x^1, \ldots, x^k)$ tel que chaque X^i soit simulable selon sa distribution conditionnelle aux autres composantes $f_i(x^i|x^1, \ldots, x^{i-1}, x^{i+1}, \ldots, x^k)$. L'échantillonnage de Gibbs consiste alors en la simulation de $X_n = x_n = (x_n^1, \ldots, x_n^k)$ en k étapes :

1. Simuler x_{n+1}^1 selon $f_1(x^1|x_n^2, \ldots, x_n^k)$;
2. Simuler x_{n+1}^2 selon $f_2(x^2|x_{n+1}^1, x_n^3, \ldots, x_n^k)$;
3. Simuler x_{n+1}^3 selon $f_3(x^3|x_{n+1}^1, x_{n+1}^2, x_n^4, \ldots, x_n^k)$;

   $\vdots$

k. Simuler x_{n+1}^k selon $f_k(x^k|x_{n+1}^1, \ldots, x_{n+1}^{k-1})$.

Remarquons qu'à l'étape j on doit prendre en compte les $j-1$ nouvelles composantes simulées aux $j-1$ étapes précédentes.

Travaux dirigés

TD-1 : Equation de diffusion de la chaleur

Soit f continue sur $[a, b]$. On cherche à résoudre numériquement le problème suivant (équation de Laplace aux conditions aux limites de Dirichlet) :

$$\begin{cases} -u''(x) & = & f(x)\ \forall x \in]a, b[\\ u(a) = \alpha & \text{et} & u(b) = \beta \end{cases} \tag{3.16}$$

où a et b sont des réels donnés tels que $a < b$.

1. On propose d'utiliser la méthode de différences finies pour calculer une solution approchée de (3.16). Donner le système d'équations obtenu en utilisant un maillage uniforme de pas $h = \frac{b-a}{N+1}$.
2. Mettre les N équations du système précédent sous forme matricielle de type $Au = h^2 w$, avec $u = (u_1, u_2, \ldots, u_N)^T$, les u_i étant une approximation de u au point $a + ih$. Donner A et w.
3. Vérifier que les vecteurs colonnes p_k de coordonnées $(\sin(\frac{k\pi}{N+1}), \sin(\frac{2k\pi}{N+1}), \ldots, \sin(\frac{Nk\pi}{N+1}))$ sont vecteurs propres de A avec les valeurs propres associées $\lambda_k = 2(1 - \cos(\frac{k\pi}{N+1}))$. En déduire que A est inversible.
4. En remarquant que A est normale, calculer le conditionnement de A vis-à-vis de la résolution du système. En faisant tendre N vers l'infini, vérifier que le conditionnement de A se comporte en $\frac{4(N+1)^2}{\pi^2}$. Que peut-on en conclure sur la résolution du système ?
5. Résoudre le problème (3.16) dans le cas où $a = 0$, $b = \pi$, $\alpha = 2$, $\beta = 0$ et $f(x) = \cos(x) - \sin(x)$.
6. Que devient le système linéaire lorsqu'on utilise une condition aux limites de type Cauchy, $u'(b) = k(u(b) - C)$?

TD-2 : un premier aperçu de la méthode des éléments finis

Soit $E = L^2([0,1])$ l'ensemble des fonctions définies sur $[0,1]$ et de carré intégrable. Pour tout couple de fonctions $(u,v) \in E^2$, $< u, v >= \int_\Omega u(x)v(x)\mathrm{d}x$ définit un produit scalaire et $(E, <.,.>)$ est un espace de Hilbert. Soit $f \in E$. On considère le problème ponctuel (P) suivant. Trouver $u \in E$ telle que

$$\left\{ \begin{array}{rcl} \forall x \in]0,1[, -\frac{\mathrm{d}}{\mathrm{d}x}(1+x)\frac{\mathrm{d}u(x)}{\mathrm{d}x} & = & f(x) \\ u(0) = \alpha \text{ et } u(1) = \beta & & \end{array} \right. \tag{3.17}$$

On considèrera dans la suite que $\alpha = \beta = 0$. On se propose dans un premier temps d'utiliser la méthode des éléments finis pour calculer une solution approchée du problème.

Rappels de cours et préliminaires

1. En introduisant une fonction auxiliaire $v \in E$ dérivable telle que $v(0) = v(1) = 0$, donner en utilisant une intégration par parties la formulation variationnelle (P') du problème (P). On cherche maintenant à résoudre le problème (P').
2. Quelles sont les conditions suffisantes portant sur les applications a et L définies respectivement par $a(u,v) = \int_0^1 (1+x)u'(x)v'(x)\mathrm{d}x$ et $L(v) = \int_0^1 f(x)v(x)\mathrm{d}x$, pour que la solution de (P') soit unique ?
3. On cherche à résoudre le problème (P') dans un espace $E_h \in E$ de dimension finie. On note (P'_h) le problème approché de (P'). Expliciter (P'_h) et discuter l'unicité de la solution.
4. On note $(\phi_1, \cdots, \phi_M)$ une base de E_h. Montrer que la résolution du problème (P'_h) passe par la résolution d'un système linéaire.

L'objectif consiste donc à expliciter les termes intervenant dans le système linéaire.

Elements finis de Lagrange

5. Soit $S = [b, b+h]$ un segment de longueur $h = \frac{1}{N+1}$ et $\mathbb{P}_k$ l'ensemble des polynômes réels de degré inférieur ou égal à k. Montrer que les éléments $(S, \mathbb{P}_1, \Sigma_1 = \{b, b+h\})$ et $(S, \mathbb{P}_2, \Sigma_2 = \{b, b+\frac{h}{2}, b+h\})$ sont des éléments finis de Lagrange.
6. On considère l'élément de type 1 $(S, \mathbb{P}_1, \Sigma_1 = \{b, b+h\})$. Donner l'expression des fonctions de forme $p_b(.)$ et $p_{b+h}(.)$ associés à cet élément fini.

Construction de E_h et résolution du problème

En utilisant l'élément fini de type 1 décrit ci-dessus, on « triangule »pour N donné le segment $[0,1]$ par des segments $[ih, (i+1)h]$ avec $h = \frac{1}{N+1}$, $0 \leq i \leq N$. On définit l'espace E_h de la façon suivante : v_h appartient à E_h si la restriction de v_h à un segment

$[ih, (i+1)h]$ est un polynôme de degré 1. En d'autres termes, si l'on note $\mathbb{S}_h$ l'ensemble des segments $[ih, (i+1)h]$,

$$E_h = \{v \in \mathcal{C}([0,1]), \forall S \in \mathbb{S}_h, v_{|K} \in \mathbb{P}_1\}. \tag{3.18}$$

7. Pour $1 \leq i \leq N$, on considère la famille ϕ_i appartenant à E_h qui vérifie

$$\begin{aligned} \phi_i(jh) &= 0 \text{ si } j \neq i, \\ \phi_i(jh) &= 1 \text{ si } j = i. \end{aligned}$$

   Montrer que la famille $(\phi_1, \phi_2, \cdots, \phi_N)$ forme une base de E_h.

8. Déduire l'expression des fonctions de base $\phi_i(.)$ à partir des fonctions de forme $p_j(.)$ calculées précédemment.
9. Déduire la dimension du système linéaire à résoudre obtenu en 4. ainsi que l'expression des coefficients de la matrice de rigidité A. On montrera que $A_{i,i} = 2(\frac{1}{h} + i)$ et $A_{i,i+1} = \frac{-1}{h} - i - \frac{1}{2}$.
10. Dans le cas où l'on triangule $[0,1]$ par des éléments finis de type 2, $(S, P_2, \Sigma_2 = \{b, b + \frac{h}{2}, b + h\})$, quelle est la dimension du système à résoudre ? Commentaire ?

Résolution par la méthode des différences finies

11. Montrer que la discrétisation de l'équation différentielle (méthode des différences finies) permet de se ramener à la résolution d'un système linéaire $Au = 2h^2 w$. On explicitera A et w.

TD-3 : éléments finis dans $\mathbb{R}^2$

Soient Ω un carré dans $\mathbb{R}^2$ de coté de longueur 5, Γ sa frontière et f une fonction définie sur Ω de carré intégrable. On recherche la solution$u$ de l'équation différentielle aux dérivées partielles

$$-\left(\frac{\partial^2 u(x,y)}{\partial x^2}+\frac{\partial^2 u(x,y)}{\partial y^2}\right)=f(x,y) \text{ dans } \Omega \quad \text{et} \quad u(x,y)=0 \text{ sur } \Gamma.$$

On se propose d'approcher la solution par la méthode des éléments finis.

Eléments finis et fonctions de forme

On considère un triangle K de sommets $s_1=(0,0)$, $s_2=(1,0)$ et $s_3=(0,1)$. Soient les points $s_4=(\frac{1}{2},0)$, $s_5=(\frac{1}{2},\frac{1}{2})$ et $s_6=(0,\frac{1}{2})$.

1. Vérifier que les triangles de type (1) et de type (2) (voir l'exemple 1.4.1) associés à K sont bien des éléments finis de Lagrange.
2. Dans le cas de triangle de type (2), explicitez les fonctions de forme p_1 et p_6 correspondant aux points s_1 et s_6.
3. Remarquer et commenter la facilité de calcul de

$$\int_K \left(\frac{\partial p_1(x,y)}{\partial x}\frac{\partial p_6(x,y)}{\partial x}+\frac{\partial p_1(x,y)}{\partial y}\frac{\partial p_6(x,y)}{\partial y}\right)dxdy.$$

Propriétés du système linéaire issu d'une équation différentielle

On considère maintenant le découpage de Ω en 25 carrés élémentaire de coté 1, ensuite on divise chaque carré élémentaire en deux triangles.

4. Rappeler la formulation variationnelle issue de l'équation différentielle de départ.
5. Donner la dimension du système linéaire à résoudre lorsque l'on utilise, respectivement, des triangles de type (1) ou (2) pour approcher la solution du problème sous sa forme variationelle.
6. En rangeant les fonctions de base de l'espace E_h ligne par ligne en commençant en haut à gauche, donner la structure de la matrice A dans le cas de l'utilisation des triangles de type (1).

Convergence de la méthode des éléments finis

Soit $(E,<\cdot,\cdot>)$ un espace de Hilbert, $a(\cdot,\cdot)$ une forme bilinéaire, continue et coercitive et u de E la solution du problème $(\mathcal{P})$ du cours. Par ailleurs, on considère une famille $(E_h)_{h>0}$ de sous-espaces vectoriels de E de dimension finie, et l'on note u_h de E_h la solution du problème approché $(\mathcal{P}_h)$ dans $E_h \subset E$.
On suppose qu'il existe un sous-espace V de E dense dans E et une application r_h de V dans E_h tels que

$$\forall v \in V,\ \lim_{h\to 0}\|v-r_h(v)\|=0.$$

7. Montrer que $\lim_{h\to 0} \|u - u_h\| = 0$ (la méthode d'approximation variationnelle converge).

TD-4 : méthodes de Monte Carlo

Méthode de la fonction de répartition inverse

Soit X une variable aléatoire réelle dont la fonction de répartition est donnée par $G(x)$ et U une variable aléatoire de loi uniforme.

1. Quelle est la loi de la variable aléatoire $Y = G^{-1}(U)$? On calculera pour cela la fonction de répartition de la variable Y.
2. On considère la loi de Laplace de paramètre λ dont la densité est

$$\forall x \in \mathbb{R},\ g_\lambda(x) = \frac{\lambda}{2} e^{-\lambda|x|},\ \lambda > 0.$$

   Déduire une méthode de simulation de réalisations de cette loi.
3. Ecrire le code Matlab correspondant.
4. Peut on utiliser la méthode de la fonction de répartition inverse pour obtenir des réalisations de la loi normale centré réduite, $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2}}$?

Méthode d'acceptation-rejet

On considère la méthode d'acceptation-rejet, avec f la densité à simuler, g une densité simulable, et M une constante connue pour laquelle $f(x) \leq Mg(x)$ pour tout x.
La méthode consiste alors en :

(i) Simuler $Z = z$ selon g, et $U = u$ selon la loi uniforme sur $[0, 1]$;

(ii) Prendre $x = z$ si $u \leq \dfrac{f(z)}{Mg(z)}$; retourner en (i) sinon.

5. Remarquer que $M \geq 1$ et montrer que $P\left[U \leq \dfrac{f(Z)}{Mg(Z)}\right] = \dfrac{1}{M}$.
6. Montrer que la réalisation de X ainsi simulée suit bien la loi de densité f.
7. Commenter l'intérêt de choisir M aussi petit que possible.
8. Déduire une méthode de simulation pour obtenir des réalisations de la loi normale centré réduite à partir de réalisations de la loi de Laplace de paramètre λ.
9. En terme de coût de calcul, quel est l'inconvénient majeur lié à l'utilisation de la méthode d'acceptation-rejet ?

Echantillonnage d'importance

On s'intéresse maintenant au calcul approché de $I = \mathrm{E}(\phi(X))$, où $X \sim \mathcal{N}(0, 1)$.

10. Exprimer $I = \mathrm{E}(\phi(X))$ sous forme intégrale et proposer une méthode d'approximation de cette espérance à partir des réalisations de la loi normale centrée réduite obtenue par la méthode d'acceptation-rejet.

11. On cherche maintenant à contourner l'inconvénient de la méthode d'acceptation-rejet évoqué plus haut, et on ne travaille qu'avec des réalisations indépendantes X^i de la loi de Laplace. Montrer que l'estimateur

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X^i)}{g(X^i)} \phi(X^i) \tag{3.19}$$

est non biaisé.

Exercices sur l'analyse numérique matricielle

1. Elimination gaussienne et Décomposition LU sous Matlab

1. Appliquer l'algorithme du pivot de Gauss sans permutation sur les matrices A et B ci-dessous.
$$A = \begin{bmatrix} 1 & 3 & -4 \\ 0 & -1 & 5 \\ 2 & 0 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 0 & 4 \\ 1 & 3 & -4 \\ 0 & -1 & 5 \end{bmatrix}$$
2. Donner les lignes de commandes Matlab réalisant les calculs précédents.
3. En vous souvenant que la matrice triangulaire obtenue par la méthode de Gauss est exactement la matrice U dans l'algorithme LU, comparer vos résultats avec la décomposition LU de Matlab (voir `help lu` pour son utilisation). Qu'observe-t-on ? Quelle conclusion en tire-t-on sur le fonctionnement de la commande `lu` de Matlab ? (pour cela comparer A et B)
4. De l'intérêt de la permutation :
$$\begin{array}{rcrcl} 10^{-17}x & + & y & = & 1 \\ x & + & y & = & 2 \end{array}$$
   (a) Donner une solution approchée intuitive du système.
   (b) Résoudre le système à l'aide de la méthode de Gauss et proposer les lignes de code Matlab mettant en œuvre cette résolution.
   (c) Exécuter ces commandes. Qu'observe-t-on ? Résoudre le système en permutant les deux équations. Comment peut-on expliquer le phénomène observé ?

2. Décomposition de matrice et moindres carrés

Soit la matrice
$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1-\eta \end{bmatrix}$$

On suppose $\eta^2 < \varepsilon$, $\eta > 100\varepsilon$, où ε est la précision machine [1]. On définit $B = A^T A$, la matrice intervenant dans les équations normales pour un problème de minimisation.

1. Calculer la décomposition de Cholesky de la matrice B en $B = LL^T$ avec L matrice triangulaire inférieure.
2. Calculer la décomposition QR de A, avec Q matrice orthogonale et R triangulaire supérieure.
3. Proposer deux solutions pour la résolution d'un système linéaire de type $Ax = b$ avec $b = \begin{bmatrix} 2 & 2 & 2 \end{bmatrix}^T$ par la méthode des moindres carrés.
4. Programmer ces deux solutions sous Matlab. Qu'observe-t-on et comment l'expliquer ?

1. noté `eps` sous Matlab et défini comme la plus grande valeur telle la représentation en virgule flottante de $1.0 + \varepsilon$ soit égale à 1.0.

3. Calcul des valeurs propres

Soit $A(\varepsilon)$ la matrice carrée d'ordre n, définie par :

$$A(\varepsilon) = \begin{bmatrix} 0 & 1 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 1 & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 1 \\ \varepsilon & 0 & \ddots & \ddots & \ddots & 0 \end{bmatrix}$$

Évaluer les valeurs propres exactes de $A(\varepsilon)$ pour $\varepsilon = 0$ et pour $\varepsilon = 10^{-10}$. Qu'en conclue-t-on ?

Travaux Pratiques

TP-1

1. Equation de diffusion de la chaleur

L'objectif de la séance de TP de manipuler en pratique quelques unes des notions introduites dans les chapitres 1 et 2 et de mettre en application des techniques présentées dans ces chapitres. Soit f continue sur $[a, b]$. On cherche à résoudre numériquement le problème suivant (équation de Laplace aux conditions aux limites de Dirichlet) traité au TD-1 :

$$\begin{cases} -u''(x) & = & f(x) \ \forall x \in]a, b[\\ u(a) = \alpha & \text{et} & u(b) = \beta \end{cases} \tag{3.20}$$

où $a = 0$, $b = \pi$, $\alpha = 2$, $\beta = 0$ et $f(x) = \cos(x) - \sin(x)$.

L'ensemble des programmes sera réalisé sous Matlab.

1. Ecrire les instructions pour construire la matrice A définie au TD-1 avec N quelconque (utiliser la fonction `diag` ou `toeplitz`). Tester avec $N = 10$.
2. Ecrire les instructions permettant de tracer la courbe d'évolution du conditionnement de A pour N compris entre N_{min} et N_{max} (tester pour $N_{min} = 2$ et $N_{max} = 100$).

3. Ecrire les instructions résolvant le système sous sa forme matricielle dans le cas particulier résolu au TD-1. On utilisera pour cela l'opérateur \ (pour son emploi, taper `doc mldivide`). On tracera la solution exacte et la solution approchée sur un même graphe.
4. Comparer la solution obtenue avec la solution qui serait obtenue en perturbant légèrement la matrice A de manière déterministe, puis de manière aléatoire. Quel phénomène est illustré ici ?
5. Tracer le graphe des solutions exacte et approchée du problème en fonction de N puis calculer et afficher l'erreur $e_N = \max_{i=1,\ldots,N} |u_i - u(x_i)|$ entre la solution approchée et la solution exacte, ainsi que le rapport $\frac{e_N}{h^2}$.

6. Ecrire les instructions traçant pour $N = 2^m$, $m = 2, \ldots, 9$, l'erreur e_N et le rapport $\frac{e_N}{h^2}$ en fonction de N.
   Qu'observe-t-on ? Cette constatation était-elle prévisible ? Pourquoi ?

2. Polynôme de Wilkinson

Un polynôme est représenté sous Matlab par un vecteur contenant ses coefficients. Si v est un vecteur de N éléments, on lui associe le polynôme suivant :

$$V(z) = v(1)z^{N-1} + v(2)z^{N-2} + \cdots + v(N-1)z + v(N)$$

de degré $N-1$.

Si w est un autre vecteur, de longueur M, associé au polynôme $W(z)$, alors le produit $P(z) = V(z)W(z)$ est un polynôme de degré $N+M-2$ dont les coefficients sont obtenus par la convolution entre les coefficients de $V(z)$ et $W(z)$. Si l'on se ramène aux vecteurs, cette opération peut être réalisée à l'aide de la fonction `conv` de Matlab : `p=conv(v,w)`.

1. Considérons le polynôme de Wilkinson

$$P(z) = \prod_{j=1}^{20}(z-j) = \sum_{k=1}^{21} p(k)z^{21-k}$$

   Ecrire un programme qui calcule les coefficients du vecteur p. Quelle est la valeur de $p(1)$?

2. A partir des coefficients du vecteur p, construire une matrice (dite *matrice compagnon*) sous la forme

$$A = \begin{bmatrix} -p(2) & -p(3) & \cdots & -p(20) & -p(21) \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

   dont les valeurs propres sont théoriquement les racines du polynôme $P(z)$.

   (a) Calculer les valeurs propres de A avec Matlab. Sont-elles proches des vraies valeurs ?

   (b) Augmenter l'élément $A(1,20)$ de 0,5% et calculer à nouveau les valeurs propres. Que constatez-vous ? Quel est illustré ici ?

TP-2

1. Approximation de π par méthode de Monte-Carlo

On considère, dans un repère orthonormé, une cible carrée de côté R dont le coin inférieur gauche coïncide avec l'origine du repère. Un tireur se présente devant la cible et on supposera que chaque tir est aléatoirement uniformément réparti sur ce carré.

1. Quelle est la probabilité qu'un tir soit dans le quart de cercle de centre O et de rayon R ?
2. Ce qui permet de distinguer un bon tir d'un mauvais est qu'un bon tir est dans le quart de cercle. De quelle type de variable aléatoire peut on se servir pour décrire le problème ?
3. Notons X cette variable aléatoire. Quelle est sa loi ? Calculer son espérance et sa variance.
4. Rappeler le principe des méthodes de Monte Carlo pour le calcul approché de l'espérance de X, $\mathrm{E}(X)$.
5. Exprimer π en fonction de $\mathrm{E}(X)$ et déduire une méthode d'approximation de π à partir de simulations de variables aléatoires.

On se propose d'implémenter sous Matlab cette méthode d'approximation de π.

6. Implémenter le calcul approché de π de façon à pouvoir modifier aisément le nombre N de simulations effectuées.
7. Calculer l'erreur relative commise en comparant avec la valeur de π donnée par Matlab.
8. Stocker dans un vecteur les valeurs de π en fonction du nombre N de simulations (on utilisera la fonction *cumsum*).
9. Justifier pourquoi l'erreur ne décroît pas nécessairement lorsque le nombre de simulations augmente.

Partie facultative : on cherche enfin à interpréter les résultats précédents.

10. En utilisant le théorème de la limite centrale, calculer pour une simulation basée sur N=10000 tirages, la probabilité que l'erreur absolue commise soit inférieure à 0.1 puis à 0.01 et enfin à $0,001$. Quelles valeurs minimales faudrait-il choisir pour N pour que ces deux dernières probabilités soient supérieures à 0,99 ? (pour cette question on utilisera la fonction normcdf)
11. Plutôt que de faire un calcul approché de la probabilité d'erreur, il est possible de majorer celle-ci grâce à l'inégalité de Bienaymé-Tchebychev. Quelles sont alors les valeurs de N permettant que l'erreur commise soit inférieure à 0.01 puis à 0.001 avec des probabilités supérieures à 0.99 ?
12. Expliquer la différence du nombre de simulations à effectuer entre les deux questions précédentes.

2. Méthodes de Monte Carlo diverses

L'objectif de l'exercice consiste à implémenter les résultats obtenus au TD-3. Nous avions considéré la loi de Laplace de paramètre $\lambda > 0$ de densité $g_\lambda(x) = \frac{\lambda}{2} e^{-\lambda|x|}$, $\forall x \in \mathbb{R}$.

1. Simuler N réalisations de cette loi à l'aide de la méthode de la fonction de répartition inverse.
2. Déduire des approximations des moments d'ordre 1 et 2 puis de $G_\lambda(1)$, où $G_\lambda(.)$ désigne la fonction de répartition associée à la loi de Laplace. Comparer aux vraies valeurs.
3. A partir des N réalisations précédentes, implémenter la méthode d'acceptation-rejet pour obtenir des réalisations de la loi normale centrée réduite.
4. Tracer l'histogramme des effectifs des réalisations obtenues afin de s'assurer que les tirages obtenus suivent bien la loi normale centrée réduite.
5. Comparer le nombre de tirages acceptés par rapport à la valeur théorique moyenne. Commentaire ?

Bibliographie

Ouvrages sur la résolution d'équations aux dérivées partielles, les éléments finis (chapitre 1)

1. Introduction à l'analyse numérique des équations aux dérivées partielles, P.A. Raviart et J. M. Thomas, Masson, 1983.
2. Introduction aux méthodes numériques, F. Andrzejewski, Springer, 2001.

Ouvrages d'analyse numérique matricielle (chapitre 2)

Les références [3] à [6] sont des ouvrages classiques consacrés à l'analyse numérique matricielle. La référence [7] est un excellent ouvrage d'algèbre linéaire.

3. Matrix computation, G. H. Golub, C. F. Van Loan, John Hopkins, 1989.
4. Introduction to matrix computation, G. W. Stuart, Academic Press, 1973.
5. Introduction à l'analyse numérique matricielle et à l'optimisation, P. G. Ciarlet, Masson, 1982.
6. Analyse numérique matricielle appliquée à l'art de l'ingénieur, P. Lascaux, R. Théodor, Masson, tome 1, 1986 ; tome 2, 1987.
7. Matrix analysis, R. A. Horn and C. R. Johnson, Cambridge University Press, 1988.

Ouvrages sur les approximations stochastiques et les techniques de simulation (chapitre 3)

8. Méthodes de Monte Carlo par chaînes de Markov, C. Robert, Economica, 1996.