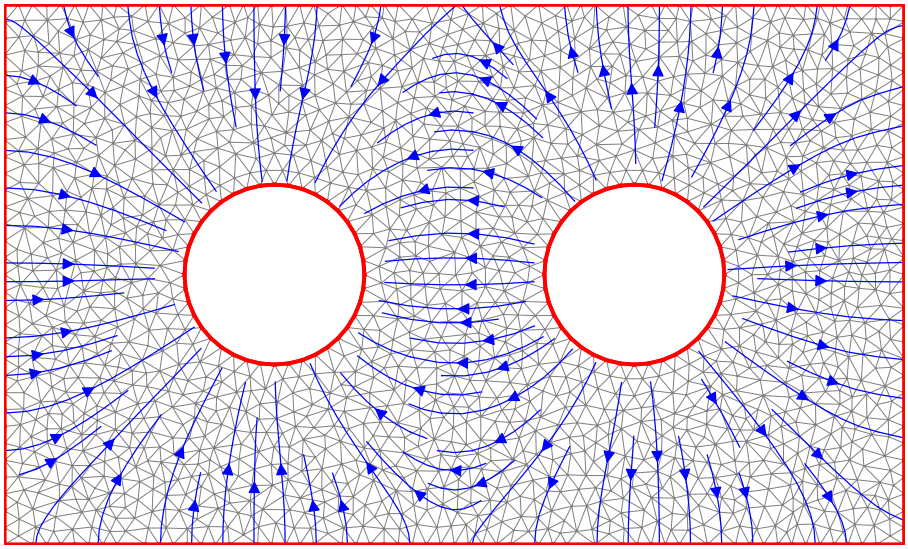
MÉTHODES NUMÉRIQUES ET SIMULATIONS

par

David SÉNÉCHAL
Ph.D., Professeur Titulaire



UNIVERSITÉ DE SHERBROOKE
Faculté des sciences
Département de physique
(mars 2020)

Ce manuel électronique fut utilisé dans le cadre du cours PHQ404/PHQ405 (Méthodes numériques et simulations) à l'Université de Sherbrooke, de 2011 à 2020. Il fait partie d'une collection de manuels électroniques diffusés par des professeurs du département de physique de l'Université de Sherbrooke. Il a été revisité pour une diffusion sous licence libre en collaboration avec la fabriqueREL en mars 2020. Il est diffusé sous licence *Creative Commons* dans sa version BY-NC, sauf indications contraires.

L'auteur, David Sénéchal, est professeur titulaire à l'Université de Sherbrooke. Son domaine de recherche est la modélisation numérique des matériaux quantiques. C'est dans un esprit de partage et de collaboration qu'il a décidé de partager cette ressource éducative libre. La liste de ses publications est disponible sur Google Scholar.

fabrique REL
RESSOURCES ÉDUCATIVES LIBRES

UNIVERSITÉ DE SHERBROOKE

TABLE DES MATIÈRES

1 Approche numérique aux problèmes physiques **1**
A Introduction . . . 1
B Grands types de problèmes . . . 2
1 Évolution temporelle de variables discrètes . . . 2
2 Problèmes aux limites . . . 2
3 Problèmes spatio-temporels . . . 3
4 Problèmes linéaires extrêmes . . . 3
5 échantillonnage . . . 3
6 Optimisation . . . 3

2 Représentation des nombres sur ordinateur **5**
A Nombres entiers . . . 5
B Nombres à virgule flottante . . . 6
1 Dépassements de capacité . . . 7
C Erreurs d'arrondi . . . 7
D Types fondamentaux en C, C++ et Python . . . 8

3 Équations différentielles ordinaires **11**
A Exemple : Mouvement planétaire . . . 12
B Méthode d'Euler . . . 13
1 Précision de la méthode d'Euler . . . 13
2 Stabilité de la méthode d'Euler . . . 13
3 Méthode prédicteur-correcteur . . . 14
C Méthode de Runge-Kutta . . . 15
1 Méthode du deuxième ordre . . . 15
2 Méthode du quatrième ordre . . . 16
3 Contrôle du pas dans la méthode de Runge-Kutta . . . 16
D Méthode de Richardson . . . 18
E Méthode d'Adams . . . 19
1 Méthode d'Adams-Bashforth . . . 19
2 Méthode d'Adams-Moulton . . . 20

4 Simulation de particules I : méthode de Verlet **21**
A Algorithme de Verlet . . . 21
1 Exemple : force constante . . . 22
B Complexité algorithmique des simulations de particules . . . 23
C Aspects quantiques et statistiques . . . 24

5 Transformées de Fourier rapides **25**
A Introduction . . . 25
B Transformées de Fourier discrètes . . . 26

C Algorithme de Danielson et Lanczos (ou Cooley-Tukey) . . . 27
1 Description de l'algorithme . . . 27
2 Cas des dimensions supérieures . . . 29
3 Fonctions réelles . . . 30

6 Simulation de particules II : écoulement d'un plasma **33**
A Description de la méthode . . . 33
1 Mise à l'échelle du problème . . . 34
2 Algorithme . . . 34
B Plasma en deux dimensions en présence d'un champ magnétique . . . 37

7 Opérations matricielles **39**
A Systèmes d'équations linéaires . . . 40
1 Système général et types de matrices . . . 40
2 Système triangulaire . . . 40
3 Élimination gaussienne . . . 41
4 Décomposition LU . . . 42
5 Système tridiagonal . . . 43
6 Décomposition QR et procédure de Gram-Schmidt . . . 44
7 Procédure de Householder . . . 46
B Valeurs et vecteurs propres . . . 48
1 Généralités . . . 48
2 Méthode de Jacobi . . . 49
3 Algorithme QR . . . 50
4 Méthode de Householder . . . 50
5 Problème aux valeurs propres généralisé . . . 52
C Décomposition en valeurs singulières . . . 53

8 Méthodes pour matrices creuses **57**
A Matrices creuses . . . 57
B Méthode du gradient conjugué . . . 58
1 Directions conjuguées . . . 58
2 Algorithme du gradient . . . 59
3 Minimisation le long de directions conjuguées . . . 60
C Méthode de Lanczos . . . 61
1 Convergence vers les valeurs propres extrêmes . . . 63
2 Calcul des vecteurs propres . . . 65
D Application : chaînes de spins et modèle de Heisenberg . . . 66
1 Produits tensoriels . . . 66
2 Modèle de Heisenberg . . . 67
3 Chaîne de spin 1/2 : analyse des effets de taille . . . 68
4 Chaîne de spin 1 : gap de Haldane . . . 69
5 Annexe : algorithme epsilon . . . 71

9 Interpolation des fonctions **73**
A Polynômes interpolants . . . 73
B Cubiques raccordées . . . 74

C Approximants de Padé . . . 77

10 Polynômes orthogonaux **81**
A Généralités . . . 81
1 Polynômes de Legendre . . . 83
2 Polynômes de Tchébychev . . . 83
3 Autres polynômes classiques . . . 85
4 Théorème sur les racines . . . 85
5 Approximation d'une fonction par un polynôme . . . 86

11 Intégration numérique **89**
A Formules élémentaires d'intégration d'une fonction . . . 89
1 Erreur de troncature dans la formule des trapèzes . . . 90
2 Formule de Simpson . . . 90
B Quadratures gaussiennes . . . 91
1 Quadratures de Gauss-Legendre . . . 94
2 Quadratures de Gauss-Tchébychev . . . 95
3 Quadratures de Gauss-Kronrod . . . 96
C Approximation de Tchébychev . . . 97

12 Problèmes aux limites en dimension 1 **101**
A Méthode du tir . . . 101
B Base de fonctions tentes . . . 103
C Méthode de Galerkin et méthode de collocation . . . 106
1 Imposition des conditions aux limites de Dirichlet . . . 107
2 Calcul du laplacien en dimension 1 . . . 108
3 Problème aux valeurs propres . . . 108
4 Exemple : équation de Helmholtz. . . . 109
5 Exemple : équation de Schrödinger . . . 111
D Méthodes spectrales . . . 111
1 Bases de polynômes orthogonaux et fonctions cardinales . . . 112
2 Opérateur différentiel . . . 113
3 Quadratures de Lobatto . . . 114
4 Exemple : équation de Helmholtz. . . . 115
5 Conditions aux limites périodiques . . . 116

13 Problèmes aux limites : dimension 2 **121**
A Triangulations . . . 121
B Fonctions tentes . . . 123
C Évaluation du laplacien. . . . 125

14 Équations aux dérivées partielles dépendant du temps **129**
A Introduction . . . 129
B Évolution directe en dimension un . . . 130
1 Analyse de stabilité de von Neumann . . . 130
C Méthode implicite de Crank-Nicholson. . . . 131
1 Analyse de stabilité . . . 132
D Méthode du saute-mouton . . . 133

E Application basée sur une représentation spectrale . . . 135
F Équation de Schrödinger dépendant du temps . . . 136
G Propagation d'une onde et solitons . . . 137
1 Équation d'advection . . . 137
2 Équation de Korteweg-de Vries . . . 138
3 Solitons . . . 139

15 Nombres aléatoires **143**
A Générateurs d'entier aléatoires . . . 143
1 Générateur à congruence linéaire . . . 144
2 Générateurs de Fibonacci . . . 144
B Générateurs de distributions continues. . . . 145
1 Distribution uniforme . . . 145
2 Méthode de transformation . . . 145
3 Méthode du rejet . . . 146
4 Méthode du rapport des aléatoires uniformes . . . 147

16 Méthode de Monte-Carlo **149**
A Intégrales multidimensionnelles . . . 149
1 Exemple simple : calcul de l'aire d'un disque . . . 152
B L'algorithme de Metropolis . . . 153
1 Chaînes de Markov . . . 154
2 Analyse d'erreur . . . 156
C Le modèle d'Ising . . . 159
1 Définition . . . 159
2 Algorithme de Metropolis appliqué au modèle d'Ising . . . 161
3 Changements de phase . . . 163
D Simulations de particules . . . 164

17 Équations non linéaires et optimisation **169**
A Équations non linéaires à une variable . . . 169
1 Cadrage et dichotomie . . . 169
2 Méthode de la fausse position . . . 170
3 Méthode de la sécante . . . 170
4 Méthode d'interpolation quadratique inverse . . . 171
5 Méthode de Newton-Raphson . . . 171
B Équations non linéaires à plusieurs variables . . . 173
1 Méthode de Newton-Raphson . . . 174
2 Méthode de Broyden . . . 174
3 Méthode itérative directe . . . 175
C Optimisation d'une fonction . . . 177
1 Méthode de Newton-Raphson . . . 177
2 Méthodes de quasi-Newton . . . 178
3 Méthode de Powell . . . 179
4 Méthode du simplexe descendant . . . 180
D Lissage d'une fonction . . . 182
1 Méthode des moindres carrés et maximum de vraisemblance . . . 182

2 Combinaisons linéaires de fonctions de lissage . . . 183
3 Lissages non linéaires . . . 185
E La méthode du recuit simulé . . . 186

18 Dynamique des fluides **189**
A Équations fondamentales . . . 189
1 Équations d'Euler et de Navier-Stokes . . . 189
2 Démonstration de l'équation d'Euler . . . 190
3 Démonstration de l'équation de Navier-Stokes . . . 192
4 Cas particulier d'un fluide incompressible . . . 192
5 Fluide incompressible en deux dimensions . . . 193
6 Écoulement irrotationnel . . . 193
B Équation de Boltzmann. . . . 194
1 Moments de la distribution . . . 194
2 Équation de Boltzmann . . . 195
3 Approximation des collisions moléculaires . . . 195
4 Approximation du temps de relaxation . . . 196
C Méthode de Boltzmann sur réseau . . . 197
1 Généralités . . . 197
2 Le schéma D2Q9 : dimension 2, 9 vitesses . . . 198
3 Conditions aux limites . . . 200
4 Algorithme . . . 200
5 Exemple . . . 200

CHAPITRE 1

APPROCHE NUMÉRIQUE AUX PROBLÈMES PHYSIQUES

A Introduction

La description du monde physique repose sur plusieurs concepts représentés par des objets mathématiques dont la définition précise a demandé de longues réflexions aux mathématiciens des siècles passés : l'infini, les nombres réels et complexes, les fonctions continues, les distributions de probabilités, etc. Les principes de la physique sont généralement exprimés par des relations entre ces objets.

Dans plusieurs cas, ces objets mathématiques peuvent être manipulés symboliquement et les équations qui les gouvernent résolues analytiquement. Les modèles les plus simples de la physique se prêtent à ces calculs, et leur solution analytique permet de comprendre l'effet des divers paramètres impliqués. Notre compréhension de base de la physique doit donc énormément à notre capacité à résoudre exactement certains modèles simples.

Dans la plupart des cas, dont les plus réalistes, les modèles ne peuvent être résolus analytiquement et tout un attirail de méthodes d'approximations analytiques a été développé, dans le but de conserver autant que possible les avantages d'une solution analytique, même approchée. La théorie des perturbations, que ce soit en mécanique classique ou quantique, est l'exemple le plus évident de méthode de calcul approchée.

Mais ces approches approximatives ont leurs limites. Elles reposent généralement sur des hypothèses, telle la petitesse de certains paramètres, qui ne sont souvent pas respectées en pratique. La vaste majorité des problèmes d'intérêt dans les sciences physiques se prêtent difficilement à une solution purement analytique, approximative ou non. On doit alors avoir recours à des méthodes numériques.

Une formation minimale sur les méthodes numériques est donc essentielle à toute personne s'intéressant à la modélisation du monde physique. En fait, l'expression «modélisation» est parfois utilisée pour signifier la description d'un système physique par un modèle qui ne peut être résolu que par des méthodes numériques.

B Grands types de problèmes

Les problèmes résolus en calcul scientifique sont très variés. Dans tous les cas, la résolution numérique du problème requiert un algorithme, une méthode ; on pourrait même parler de «recette». Or, toute recette complexe fait appel à d'autres recettes, plus générales, et donc plus basiques. Par exemple, plusieurs problèmes de physique ou de génie font appel à l'intégration de fonctions. Indépendamment du domaine d'études précis, il est donc important de comprendre comment intégrer des fonctions numériquement. De même, les opérations courantes d'algèbre linéaire (solution d'un système d'équations linéaires, diagonalisation d'une matrice, etc.) sont omniprésentes en calcul scientifique, quel que soit le domaine d'application.

Mais tout n'est pas qu'algorithme. La représentation numérique des objets mathématiques ou physiques est également très importante. Prenons par exemple le concept de fonction à une variable. Dans plusieurs problèmes, on cherche à déterminer une fonction ψ inconnue *a priori*. Cette fonction doit être représentée numériquement et la manière la plus appropriée d'y arriver peut dépendre de la formulation du problème : s'il s'agit d'une équation différentielle avec valeurs initiales, l'algorithme peut nous faire «avancer dans le temps» de manière à calculer la fonction $\psi(t)$ au fur et à mesure, sans avoir à la représenter dans son ensemble. Par contre, s'il s'agit d'une équation différentielle avec valeurs aux frontières, il peut être nécessaire de représenter l'ensemble de la fonction $\psi(x)$ dans un domaine précis à chaque étape de l'algorithme. La quantité d'information requise pour représenter les données nécessaires à l'algorithme dépend donc de l'algorithme utilisé, pas uniquement du type d'objet traité (ici, une fonction à une variable).

Les grands types de problèmes que nous allons considérer sont les suivants :

B.1 Évolution temporelle de variables discrètes

Dans ce type de problèmes, l'évolution dans le temps d'un ensemble de variables discrètes est régie par un ensemble de règles, par exemple des équations différentielles ordinaires. Par exemple, un problème de mécanique se traduit par un système d'équations différentielles sur un nombre bien défini de variables (les positions et les vitesses des particules en jeu). Un problème d'évolution temporelle se résout normalement en avançant dans le temps de manière discrète, en définissant un pas temporel h (pas nécessairement constant) et en déterminant les valeurs des variables au temps $t+h$ en fonction des variables au temps t, de proche en proche, sans avoir à conserver en mémoire les valeurs calculées depuis le début. La difficulté numérique dans ce type de problème est d'assurer une précision contrôlée aux prédictions, en dépit de la valeur finie du pas temporel h.

B.2 Problèmes aux limites

Dans ce type de problèmes, une ou plusieurs fonctions de la position $\psi_a(\mathbf{r})$ doivent être déterminées dans une région de l'espace de forme plus ou moins complexe, en respectant (i) certaines conditions aux limites exprimées en fonction des valeurs des fonctions $\psi_a(\mathbf{r})$ à la frontière du domaine et (ii) certaines équations constitutives valables dans le domaine considéré. Un exemple simple est l'équation de Laplace $\nabla^2\phi = 0$ pour le potentiel électrique ϕ dans l'espace vide, avec des frontières conductrices maintenues à des valeurs connues de ϕ. Dans une telle situation, la représentation des

fonctions inconnues passe généralement par une représentation du domaine spatial considéré par une grille de points ou *maillage*. La détermination du maillage constitue en soi un sous-domaine important du calcul scientifique.

B.3 Problèmes spatio-temporels

On peut combiner les deux catégories précédentes dans un type de problème qui implique à la fois une évolution temporelle et une dépendance spatiale. C'est le cas des équations différentielles aux dérivées partielles qui dépendent du temps : l'équation d'onde, l'équation de Navier-Stokes, les équations de Maxwell, les équations de la relativité générale, etc. Appartiennent aussi à cette catégorie des problèmes comportant plusieurs composantes ou systèmes en interaction, dont l'exemple extrême est la modélisation météorologique ou climatique. Il s'agit d'une catégorie particulièrement difficile et répandue de problèmes.

B.4 Problèmes linéaires extrêmes

L'algèbre linéaire est omniprésente en calcul scientifique. Par exemple, les équations aux dérivées partielles peuvent souvent être formulées dans le langage de l'algèbre linéaire. Par contre, la linéarité est une propriété fondamentale de la mécanique quantique. Les problèmes quantiques sont souvent traités à l'aide de plusieurs techniques d'algèbre linéaire numérique, notamment via l'utilisation de matrices creuses et l'application répétée de matrices de très grande taille sur des vecteurs représentant des états physiques.

B.5 échantillonnage

Plusieurs problèmes visent à calculer les valeurs moyennes de quantités définies sur des espaces de configurations très vastes, trop vastes pour permettre un calcul direct. Par exemple, on pourrait chercher à calculer la valeur moyenne de la vitesse d'une molécule dans un gaz réel (c.-à-d. non parfait) en fonction de la température. Ou encore, le nombre moyen de particules d'un certain type pouvant frapper un détecteur à un endroit précis suite à une collision de haute énergie dans un détecteur de particules. Dans ces cas, le nombre de possibilités est si grand («astronomique» serait un euphémisme) qu'on doit procéder à un *échantillonnage* des possibilités en suivant une certaine loi de probabilités. La difficulté est alors de procéder efficacement à un échantillonnage qui respecte précisément des probabilités connues *a priori*. Les méthodes de ce type sont généralement connues sous le qualificatif générique de *Monte-Carlo*. Notons qu'on désire généralement connaître non seulement les valeurs moyennes des quantités d'intérêt, mais aussi l'erreur commise par l'échantillonnage lui-même sur ces valeurs moyennes.

B.6 Optimisation

Plusieurs problèmes visent à trouver une configuration optimale, généralement le minimum ou le maximum d'une fonction de plusieurs variables. La détermination des paramètres d'un modèle afin de représenter un ensemble de données (lissage de courbes) appartient à cette catégorie. Parfois les configurations ne sont pas des points dans $\mathbb{R}^n$, mais des objects discrets, par exemple le problème

du commis-voyageur. Parfois on cherche non pas un minimum mais une valeur précise de la fonction (recherche de racines). Plusieurs algorithmes ont été mis au point pour cette catégorie de problèmes ; ils procèdent généralement par une «marche» dans l'espace considéré, marche guidée par la fonction à optimiser évaluée en plusieurs points. Cette marche d'arrête lorsque certains critères de convergence sont satisfaits.

CHAPITRE 2

REPRÉSENTATION DES NOMBRES SUR ORDINATEUR

Le premier problème rencontré en modélisation numérique est la représentation des objets mathématiques courants (nombres, fonctions, etc.) sur un ordinateur. Nous nous heurtons alors au fait qu'un ordinateur ne peut représenter ni l'infini, ni le continu.

Un calculateur électronique représente les données (y compris les instructions visant à les manipuler) par un ensemble d'états électroniques, chacun ne pouvant prendre que deux valeurs, tel un commutateur qui est soit ouvert ou fermé. Chacun de ces systèmes abrite donc un atome d'information, ou bit, et l'état de chaque bit peut prendre soit la valeur 0, soit la valeur 1.

Le bit étant l'unité fondamentale d'information, l'*octet* (angl. *Byte*) est défini comme un ensemble de 8 bits et sert couramment d'unité pratique d'information. Le Kilooctet (ko) désigne 10^3 octets, le Mégaoctet (Mo) désigne 10^6 octets, le Gigaoctet (Go) 10^9 octets, et ainsi de suite. On utilise aussi des multiples basés sur les puissances de $2^{10} = 1024$, portant le même nom mais notés différemment : 1 kio = 1024 octets, 1 Mio = 1024 kio, 1 Gio = 1024 Mio, etc. En 2017, la mémoire d'un ordinateur personnel typique se situe autour de 10^{10} octets.

A Nombres entiers

À partir des états binaires, un nombre entier naturel (l'un des concepts les plus simples, et probablement le plus ancien, des mathématiques) peut être représenté en base 2. Par exemple, on a la représentation binaire des nombres entiers suivants :

$$57 = 111001_2 \qquad 2532 = 100111100100_2 \tag{2.1}$$

Les entiers relatifs ($\mathbb{Z}$) sont représentés de la même manière, sauf qu'un bit supplémentaire est requis pour spécifier le signe ($\pm$). Bien évidemment, une quantité donnée d'information (un nombre donné de bits) ne peut représenter qu'un intervalle fini de nombres entiers. Un entier naturel de 4 octets (32 bits) peut donc prendre les valeurs comprises de 0 à $2^{32} - 1 = 4\,294\,967\,295$. Un entier relatif peut donc varier entre $-2\,147\,483\,648$ et $2\,147\,483\,647$. Un entier relatif de 8 octets (64 bits) peut varier entre $-2^{63} = -9\,223\,372\,036\,854\,775\,808$ et $2^{63} - 1 = 9\,223\,372\,036\,854\,775\,807 \sim -9.10^{18}$.

Les opérations élémentaires sur les entiers (addition, multiplication, etc.) ne sont donc pas fermées sur ces entiers : ajouter 1 à l'entier naturel maximum redonne la valeur 0. Les opérations sont effectuées modulo la valeur maximale admissible. Il est donc important se s'assurer que les entiers

manipulés dans un code soient toujours en deçà des bornes maximales permises si on désire qu'ils se comportent effectivement comme des entiers.

Un processeur ayant une architecture à 64 bits permet d'utiliser des entiers de 8 octets pour représenter les adresses des données en mémoire, alors qu'une architecture à 32 bits n'utilise que des entiers de 4 octets. Conséquemment, cette dernière ne peut pas adresser plus que $2^{32} = 4$ Gio de mémoire, alors que la première, omniprésente de nos jours, permet d'adresser une quantité de mémoire bien au-delà des capacités courantes des ordinateurs. Tous les systèmes d'exploitation récents sont basés sur une architecture à 64 bits, mais le mode 32 bits est parfois nécessaire afin d'assurer la compatibilité avec des logiciels plus anciens. La longueur naturelle (par défaut) des entiers dans une architecture de 64 bits sera effectivement de 8 octets.

B Nombres à virgule flottante

La représentation binaire des nombres réels pose un problème plus complexe. On introduit le concept de *nombre à virgule flottante* (NVF) pour représenter de manière approximative un nombre réel. Un NVF comporte un *signe*, une *mantisse* et un *exposant*, comme suit :

$$x \rightarrow \pm b_p\, b_{p-1} \ldots b_1 \times 2^{\pm e_q\, e_{q-1} \ldots e_1} \tag{2.2}$$

où les b_i sont les bits de la mantisse et les e_i ceux de l'exposant binaire. Au total, $p+q+2$ bits sont requis pour encoder un tel nombre (2 bits pour les signes de la mantisse et de l'exposant).

Il a fallu plusieurs années avant qu'un standard soit développé pour les valeurs de p et q en 1987, fruit d'une collaboration entre l'*Institute of Electrical and Electronics Engineers* (IEEE) et l'*American National Standards Institute* (ANSI). Ce standard, appelé IEEE 754, prend la forme suivante :

$$x \rightarrow \pm 1.f \times 2^{e-\text{decal.}} \tag{2.3}$$

où f est la partie fractionnaire de la mantisse et où un décalage est ajouté à l'exposant e. L'exposant e comporte 8 bits, de sorte que $0 \leq e \leq 255$, et le décalage, pour des nombres à 32 bits (précision simple) est 127. Le fait d'utiliser un décalage nous dispense de coder le signe de l'exposant. La mantisse comporte, elle, 23 bits, qui représentent la partie fractionnaire f de la mantisse. Par exemple, dans l'expression binaire à 32 bits

$$\underbrace{0}_{\text{signe}} : \underbrace{01111100}_{\text{exposant}} : \underbrace{01000000000000000000000}_{\text{mantisse}} \tag{2.4}$$

le signe est nul (donc +), l'exposant est $124-127 = -3$, et la mantisse est $1,01_2 = 1,25$. Le nombre représenté est donc $+1.25 \times 2^{-3} = 0.15625$.

Signalons que des subtilités se produisent lorsque l'exposant est nul ($e = 0$) ou maximum ($e = 255$), mais nous n'entrerons pas dans ces détails ici. [1]

Un NVF de 32 bits peut effectivement représenter des nombres réels compris entre 1.4×10^{-45} et 3.4×10^{38}, avec 6 ou 7 chiffres significatifs ($23 \times \log_{10}(2) \approx 6.9$). Un tel nombre est qualifié de nombre à *précision simple*.

1. Voir par exemple les pages sur IEEE 754 dans Wikipedia.

La précision simple est généralement insuffisante pour les applications scientifiques. On utilise plutôt la *précision double*, basée sur une représentation à 64 bits (8 octets) des NVF : l'exposant est codé en 11 bits et la mantisse en 52 bits, ce qui permet de représenter effectivement des nombres réels compris entre

$$4.9 \times 10^{-324} < \text{double précision} < 1.8 \times 10^{308} \tag{2.5}$$

avec 15 chiffres significatifs en base 10.

B.1 Dépassements de capacité

Comme la représentation en virgule flottante ne représente qu'un intervalle de nombres possibles sur l'ensemble des réels, certaines opérations sur ces nombres produisent des nombres qui sont trop grands ou trop petits pour être représentés par un NVF. C'est ce qu'on appelle un *dépassement de capacité* (*overflow* ou *underflow*, en anglais). Un nombre trop grand pour être représenté par un NVF est plutôt représenté par le symbole `NaN` (pour *not a number*). Ce symbole est aussi utilisé pour représenter le résultat d'opérations impossibles sur les réels, par exemple la racine carrée d'un nombre négatif. Les nombres trop petits sont généralement remplacés par zéro.

C Erreurs d'arrondi

Les NVF représentent exactement un sous-ensemble des réels, en fait un sous-ensemble des nombres rationnels, ceux qui s'expriment exactement en base 2 par une mantisse de 52 bits et un exposant de 11 bits (pour un nombre à double précision). Cet ensemble n'est pas fermé sous les opérations arithmétiques habituelles (addition, multiplication, inversion). L'erreur ainsi générée dans la représentation des réels et dans les opérations arithmétiques est qualifiée d'*erreur d'arrondi*.

Exemple 2.1 Erreur d'arrondi dans une addition simple
Voyons comment l'erreur d'arrondi se manifeste dans l'opération simple

$$7 + 1 \times 10^{-7} \tag{2.6}$$

en précision simple. Chacun des deux termes s'exprime, en binaire, comme suit :

$$\begin{aligned} 7 &= 0 : 10000001 : 11000000000000000000000 \\ 1 \times 10^{-7} &= 0 : 01100111 : 10101101011111100101001 \end{aligned} \tag{2.7}$$

Expliquons : l'exposant du nombre 7 est 129, moins le décalage de 127, ce qui donne 2. La mantisse est $1+2^{-1}+2^{-2} = \frac{7}{4}$, et donc on trouve bien $\frac{7}{4} \times 2^2 = 7$. Pour le deuxième nombre, l'exposant est $103-127 = -24$ et la mantisse est $1+2^{-1}+2^{-3}+2^{-5}+2^{-6}+\cdots = 1.6777216$, ce qui donne $1.6777216 \times 2^{-24} = 1.0 \times 10^{-7}$.
Additionnons maintenant ces deux nombres. Pour ce faire, on doit premièrement les mettre au même exposant, ce qui se fait en déplaçant les bits du deuxième nombre de 26 positions vers la droite, ce qui fait disparaître toute la mantisse. Le deuxième nombre devient donc effectivement nul, et on trouve

$$7 + 1.0 \times 10^{-7} = 7 \quad \text{(simple précision)} \tag{2.8}$$

Nom	type	octets
`bool`	logique	1
`char`	caractère	1
`short`	entier	2
`int`	entier	4
`long`	entier	8
`float`	NVF	4
`double`	NVF	8
`long double`	NVF	16
	pointeur	8

TABLE 2.1
Types simples en C/C++, sur une machine à 64 bits

On définit la *précision-machine* comme le nombre le plus grand qui, ajouté à 1, redonne 1. Pour un nombre à précision simple, la précision-machine est de `5.96e-08`. Pour un nombre à double précision, elle est de `1.11e-16`.

Les erreurs d'arrondi se produisent essentiellement à chaque fois que des NVF sont l'objet d'opérations arithmétiques (additions, soustractions, multiplications, divisions) et peuvent être aussi bien négatives que positives. Si un calcul comporte N opérations arithmétiques, il est plausible que l'erreur d'arrondi accumulée lors de ces opérations soit de l'ordre de $\sqrt{N}$ fois la précision-machine, car l'erreur se propage un peu comme une marche aléatoire. La précision-machine n'est donc pas la précision des calculs effectués, mais seulement une précision maximale théorique.

D Types fondamentaux en C, C++ et Python

Que l'on programme en C, C++ ou Python, on doit composer avec des types de nombres pré-définis en fonction desquels les opérations courantes sont efficaces sur les micro-processeurs courants. Les types propres à C/C++ sont indiqués dans le tableau 2.1. Python n'est pas un langage à types implicites ; on peut y connaître les caractéristiques des types fondamentaux en invoquant les fonctions suivantes :

```
import sys
print(sys.int_info)
print(sys.float_info)
```

Les nombres complexes ne constituent pas un type fondamental en C, mais font partie de la bibliothèque standard de C++ (STL) ; il s'agit alors d'un type générique `complex<>` qui peut se spécialiser selon la précision requise, par exemple en `complex<double>` ou en `complex<int>`. En Python, les complexes sont un type fondamental, représenté par deux `float`. NumPy permet de spécifier les types utilisés de manière plus précise (tableau 2.2).

type	description
`bool_`	booléen (True ou False), stocké dans un octet
`int_`	type entier par défaut (= `long` en C ; habituellement int64 ou int32)
`intc`	= `int` de C (habituellement int32 or int64)
`intp`	entier utilisé pour l'indexation (= `size_t` en C ; habituellement int32 ou int64)
`int8`	octet (-128 à 127)
`int16`	entier (-32 768 à 32 767)
`int32`	entier (-2 147 483 648 à 2 147 483 647)
`int64`	entier (-9 223 372 036 854 775 808 à 9 223 372 036 854 775 807)
`uint8`	entier non négatif (0 à 255)
`uint16`	entier non négatif (0 à 65 535)
`uint32`	entier non négatif (0 à 4 294 967 295)
`uint64`	entier non négatif (0 à 18 446 744 073 709 551 615)
`float_`	= float64.
`float16`	NVF à demi-précision : bit de signe, exposant à 5 bits, mantisse à 10 bits
`float32`	NVF à précision simple : bit de signe, exposant à 8 bits, mantisse à 23 bits
`float64`	NVF à précision double : bit de signe, exposant à 11 bits, mantisse à 52 bits
`complex_`	= complex128.
`complex64`	Nombre complexe de deux NVF à 32 bits
`complex128`	Nombre complexe de deux NVF à 64 bits

TABLE 2.2
Types disponibles avec NumPy

CHAPITRE 3

ÉQUATIONS DIFFÉRENTIELLES ORDINAIRES

Ce chapitre est consacré à la solution des systèmes d'équations différentielles ordinaires, c'est-à-dire aux systèmes d'équations de la forme

$$\boxed{\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, t)} \tag{3.1}$$

où $\mathbf{x}(t)$ est une collection de N fonctions dépendant d'une variable indépendante t, et $\mathbf{f}$ est un ensemble de N fonctions de $\mathbf{x}$ et de t.

Remarques :

- La variable t joue typiquement le rôle du temps en mécanique, d'où la notation utilisée. Mais elle peut en général avoir une interprétation différente.
- La forme (3.1) peut sembler restrictive, car il s'agit d'un système du premier ordre en dérivées seulement. Cependant tout système d'équations différentielles d'ordre plus élevé peut être ramené à un système du type (3.1) en ajoutant des variables au besoin pour tenir la place des dérivées d'ordre sous-dominant. Par exemple, considérons une équation différentielle du deuxième ordre pour une variable y :

  $$\ddot{y} = f(y, \dot{y}, t) \tag{3.2}$$

  En définissant les variables $x_1 = y$ et $x_2 = \dot{y}$, cette équation peut être ramenée au système suivant :

  $$\dot{x}_2 = f(x_1, x_2, t) \qquad \dot{x}_1 = x_2 \tag{3.3}$$

  qui est bien de la forme (3.1) où $f_1(\mathbf{x}, t) = x_2$ et $f_2 = f$.
- Le passage d'un système d'équations du deuxième ordre à M variables vers un système du premier ordre à N = 2M variables est justement ce qui est réalisé en passant de la mécanique de Lagrange à la mécanique de Hamilton. Les équations du mouvement de Hamilton sont de la forme (3.1), mais leur structure symplectique est spécifique, alors que la forme (3.1) est plus générale.

A Exemple : Mouvement planétaire

Considérons premièrement un problème dont nous connaissons la solution analytique : celui du mouvement d'un corps de masse m dans le champ gravitationnel d'un objet de masse M, fixe à l'origine. Nous savons que le problème peut être limité à deux dimensions d'espace, en raison de la conservation du moment cinétique.

Adoptons les coordonnées cartésiennes (x, y), et soit (v_x, v_y) les vitesses correspondantes. Les équations du mouvement pour ce problème sont, en forme vectorielle,

$$\dot{\mathbf{r}} = \mathbf{v} \qquad \dot{\mathbf{v}} = -\frac{k}{r^3}\mathbf{r} \tag{3.4}$$

où nous avons introduit la constante $k = \mathrm{GM}$. Nous pouvons toujours choisir un système d'unités pour le temps dans lequel cette constante vaut 1 (voir ci-dessous). Le système d'équations se ramène donc à l'ensemble suivant, si on numérote les variables dans l'ordre suivant : (x, y, v_x, v_y) :

$$\dot{x}_1 = x_3 \qquad \dot{x}_2 = x_4 \qquad \dot{x}_3 = -\frac{x_1}{r^3} \qquad \dot{x}_4 = -\frac{x_2}{r^3} \qquad r := (x_1^2 + x_2^2)^{1/2} \tag{3.5}$$

Choix du système d'unités

Dans la résolution numérique de problèmes physiques, il est très important d'utiliser des grandeurs relatives à certaines échelles caractéristiques du problème, de manière à ce que les nombres à virgules flottantes qui sont manipulés n'aient pas des exposants extravagants. Non seulement cela nous protège-t-il des limites numériques inhérentes à ces nombres, mais cela permet aussi de donner un sens physique plus évident aux résultats. Manipuler des valeurs relatives équivaut bien sûr à choisir un système d'unités particulier, différent d'un système standard, comme le SI.

Par exemple, supposons que le temps t soit défini en rapport avec un temps caractéristique τ, et les distances définies en rapport avec une longueur caractéristique a. Le temps mesuré en SI serait alors τt, où t est maintenant un rapport 'sans unités', et pareillement pour la position $a\mathbf{r}$. Remplaçons alors, dans les équations (3.4), le temps t par $t\tau$, le rayon r par ra et la vitesse v par va/τ. On trouve alors les équations suivantes :

$$\dot{\mathbf{r}} = \mathbf{v} \qquad \dot{\mathbf{v}} = -\frac{\mathrm{GM}\tau^2}{a^3 r^3}\mathbf{r} \tag{3.6}$$

La constante k de l'équation (3.4) vaut donc $\mathrm{GM}\tau^2/a^3$, alors que les variables t, $\mathbf{r}$ et $\mathbf{v}$ sont sans unités, c'est-à-dire exprimées comme multiples de τ, a et a/τ, respectivement.

Si on choisit a comme étant le demi-grand axe de l'orbite de la Terre autour du soleil (une unité astronomique) et τ comme étant la période $\tau_\oplus$ de cette orbite, alors, étant donné la relation connue pour cette période,

$$\tau_\oplus^2 = a^3 \frac{(2\pi)^2}{\mathrm{GM}} \tag{3.7}$$

il s'ensuit que $k = (2\pi)^2$. Si, au contraire, on insiste pour que $k = 1$ – le choix que nous avons fait ci-dessus – alors le temps caractéristique est $\tau = \tau_\oplus/2\pi$.

Une analyse d'échelle comme celle-ci doit en principe être faite pour chaque problème résolu de manière numérique.

B Méthode d'Euler

La méthode la plus élémentaire pour solutionner numériquement l'équation différentielle (3.1) est la *méthode d'Euler*. Sa simplicité est cependant contrebalancée par son manque de précision et de stabilité. Elle consiste à remplacer l'équation (3.1) par une équation aux différences finies : l'axe du temps est remplacé par une suite d'instants également espacés $t \to t_n = nh$, $(n = 0, 1, 2\ldots)$, où h est le *pas temporel*. La fonction $\mathbf{x}(t)$ est alors remplacée par une suite $\{\mathbf{x}_n\}$ et la dérivée est estimée par l'expression [1]

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{h} \tag{3.8}$$

Le système (3.1) est alors modélisé par l'équation aux différences suivante :

$$\boxed{\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(\mathbf{x}_n, t_n)} \tag{3.9}$$

Cette expression définit une application explicite $\mathbf{x}_n \mapsto \mathbf{x}_{n+1}$ qui permet, par récurrence, d'obtenir la suite complète à partir des valeurs initiales $\mathbf{x}_0$.

B.1 Précision de la méthode d'Euler

La principale source d'erreur de la méthode d'Euler provient du pas temporel h, qui doit être pris suffisamment petit. Cette erreur est qualifiée d'*erreur de troncature*. Si nous avions accès aux dérivées d'ordre arbitraire de la fonction $\mathbf{x}(t)$, nous pourrions envisager le développement de Taylor suivant :

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \left.\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right|_{t_n} h + \frac{1}{2}\left.\frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}t^2}\right|_{t_n} h^2 + \cdots \tag{3.10}$$

Le remplacement (3.9) revient à négliger les termes en h^2 (ou plus grand) dans ce développement. On dit alors que la méthode d'Euler est du *premier ordre* en h.

B.2 Stabilité de la méthode d'Euler

Au-delà des considérations de précision, l'application (3.9) est aussi sujette à des erreurs d'arrondi. Supposons à cet effet qu'une telle erreur produise une déviation $\delta\mathbf{x}_n$ de la solution numérique, par rapport à la solution exacte de l'équation aux différences (3.9). La méthode sera *stable* si cette déviation décroît avec le temps et *instable* dans le cas contraire. Appliquons un développement limité à l'éq. (3.9) :

$$\delta\mathbf{x}_{n+1} = \delta\mathbf{x}_n + h\frac{\partial}{\partial x_\alpha}\mathbf{f}(\mathbf{x}_n, t_n)\delta x_{\alpha,n} \quad \text{ou} \quad \delta\mathbf{x}_{n+1} = (\mathbb{I} + h\mathbf{M})\,\delta\mathbf{x}_n \tag{3.11}$$

où la matrice $\mathbf{M}$ possède les composantes suivantes :

$$\mathrm{M}_{\alpha\beta} = \frac{\partial f_\alpha}{\partial x_\beta} \tag{3.12}$$

1. Les éléments du vecteur $\mathbf{x}_n$ sont notés $x_{\alpha,n}$, où $\alpha = 1, 2, \ldots, \mathrm{N}$.

et est évaluée au temps t_n.

De toute évidence, la méthode d'Euler sera instable si la matrice $(\mathbb{I} + h\mathbf{M})$ possède au moins une valeur propre en dehors de l'intervalle $(-1, 1)$, de manière répétée en fonction du temps. En effet, dans ce cas, l'itération de la procédure va produire des différences $\delta\mathbf{x}_n$ de plus en plus grandes. Dans le cas d'une seule variable (N = 1), la condition de stabilité revient à

$$-1 < 1 + h\frac{\mathrm{d}f}{\mathrm{d}x} < 1 \Longrightarrow \frac{\mathrm{d}f}{\mathrm{d}x} < 0 \quad \text{et} \quad \left|\frac{\mathrm{d}f}{\mathrm{d}x}\right| < \frac{2}{h} \tag{3.13}$$

Exemple 3.1 Équation linéaire à une variable
Considérons l'équation

$$\dot{x} = \lambda x \tag{3.14}$$

dont la solution analytique est $x(t) = x(0)\mathrm{e}^{\lambda t}$. La méthode d'Euler produit l'équation aux différences suivante :

$$x_{n+1} = x_n + h\lambda x_n = (1 + h\lambda)x_n \tag{3.15}$$

L'analyse de stabilité produit la relation suivante pour les déviations :

$$\delta x_{n+1} = (1 + h\lambda)\delta x_n \tag{3.16}$$

ce qui montre que la méthode d'Euler est stable seulement si $\lambda < 0$ (décroissance exponentielle) et instable dans le cas d'une croissance exponentielle ($\lambda > 0$) ou d'une oscillation (λ imaginaire). Elle peut même être instable dans le cas d'une décroissance exponentielle si $h > 2/|\lambda|$.

B.3 Méthode prédicteur-correcteur

Une façon simple d'améliorer la méthode d'Euler est de la scinder en deux étapes :

1. On procède à une *prédiction* sur la valeur au temps t_{n+1} : $\mathbf{x}^{\text{pr.}}_{n+1} = \mathbf{x}_n + h\mathbf{f}(\mathbf{x}_n, t_n)$.
2. On procède ensuite à une *correction* de cette prédiction, en remplaçant la dérivée évaluée à $\mathbf{x}_n$ par la moyenne de la dérivée évaluée à $(\mathbf{x}_n, t_n)$ et $(\mathbf{x}^{\text{pr.}}_n, t_{n+1})$:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \tfrac{1}{2}h\left(\mathbf{f}(\mathbf{x}_n, t_n) + \mathbf{f}(\mathbf{x}^{\text{pr.}}_{n+1}, t_{n+1})\right) \tag{3.17}$$

On montre que cette méthode, qui requiert manifestement deux fois plus d'évaluations de $\mathbf{f}$ que la méthode d'Euler simple, est cependant du deuxième ordre en h, c'est-à-dire que l'erreur de troncature est d'ordre h^3. Voyons la démonstration : écrivons premièrement une expression explicite de l'éq. (3.17) :

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \tfrac{1}{2}h\mathbf{f}(\mathbf{x}_n, t_n) + \tfrac{1}{2}h\mathbf{f}(\mathbf{x}_n + h\mathbf{f}(\mathbf{x}_n, t_n), t_{n+1}) \tag{3.18}$$

On développe ensuite le dernier terme au premier ordre en série de puissances de h :

$$\begin{aligned}\mathbf{f}(\mathbf{x}_n + h\mathbf{f}(\mathbf{x}_n, t_n), t_n + h) &= \mathbf{f}(\mathbf{x}_n, t_n) + h\frac{\partial \mathbf{f}}{\partial x_i}f_i + h\frac{\partial \mathbf{f}}{\partial t} + \mathcal{O}(h^2) \\ &= \mathbf{f}(\mathbf{x}_n, t_n) + h\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}t} + \mathcal{O}(h^2)\end{aligned} \tag{3.19}$$

où les dérivées sont évaluées à $(\mathbf{x}_n, t_n)$ et la dérivée totale de $\mathbf{f}$ par rapport au temps est obtenue par la règle habituelle :

$$\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}t} = \frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x_i} f_i \tag{3.20}$$

car $\mathbf{f}$ est la dérivée de $\mathbf{x}$ par rapport au temps. On peut combiner les trois dernières équations en une seule :

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\left.\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right|_{t_n} + \tfrac{1}{2}h^2\left.\frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}t^2}\right|_{t_n} + \mathcal{O}(h^3) \tag{3.21}$$

ce qui n'est rien d'autre que le développement limité de $\mathbf{x}_{n+1}$ au deuxième ordre, avec les coefficients appropriés. On en conclut donc que la formule (3.17) fournit une estimation de $\mathbf{x}_{n+1}$, correcte à l'ordre h^2.

C Méthode de Runge-Kutta

La méthode de Runge-Kutta est une amélioration notable de la méthode d'Euler, et constitue en fait une généralisation de la méthode prédicteur-correcteur. Elle consiste à évaluer la dérivée $\mathbf{f}$ non pas au temps t_n, ni au temps t_{n+1}, mais entre les deux en général.

C.1 Méthode du deuxième ordre

Commençons par démontrer la méthode de Runge-Kutta d'ordre 2. Nous allons supposer qu'une étape de la méthode prend la forme suivante :

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + a\mathbf{k}_1 + b\mathbf{k}_2 \\ \mathbf{k}_1 &= h\mathbf{f}(\mathbf{x}_n, t_n) \\ \mathbf{k}_2 &= h\mathbf{f}(\mathbf{x}_n + \beta\mathbf{k}_1, t_n + \alpha h) \end{aligned} \tag{3.22}$$

et nous allons chercher à déterminer les constantes a, b, α et β de manière à maximiser la précision de la méthode. Notons que les valeurs $a = b = \frac{1}{2}$ et $\alpha = \beta = 1$ correspondent à la méthode prédicteur-correcteur (3.17). Mais d'autres valeurs donnent aussi une prédiction du deuxième ordre.

Pour déterminer ces paramètres, appliquons encore une fois un développement limité à $\mathbf{k}_2$, en tenant compte de (3.20) :

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + a\,h\mathbf{f} + b\,h\mathbf{f}\big(\mathbf{x}_n + h\beta\mathbf{f}(\mathbf{x}_n, t_n), t_n + \alpha h\big) \\ &= \mathbf{x}_n + (a+b)h\mathbf{f} + b\beta h^2 \frac{\partial \mathbf{f}}{\partial x_i}\frac{\partial x_i}{\partial t} + b\alpha h^2 \frac{\partial \mathbf{f}}{\partial t} + \mathcal{O}(h^3) \end{aligned} \tag{3.23}$$

où toutes les expressions sont évaluées à $(\mathbf{x}_n, t_n)$ dans la dernière ligne. Pour que cette dernière expression coïncide avec le développement de Taylor correct (3.21), les paramètres a, b, α et β doivent respecter les contraintes suivantes :

$$a + b = 1 \qquad b\alpha = \tfrac{1}{2} \qquad b\beta = \tfrac{1}{2} \tag{3.24}$$

La solution à ces contraintes n'est pas uniquement $a = b = \frac{1}{2}$ et $\alpha = \beta = 1$ (méthode prédicteur-correcteur). Un autre choix possible est $a = 0$, $b = 1$ et $\alpha = \beta = \frac{1}{2}$, qui correspond à ce qui est généralement appelé la méthode de Runge-Kutta du deuxième ordre :

$$\begin{aligned} \mathbf{k}_1 &= h\mathbf{f}(\mathbf{x}_n, t_n) \\ \mathbf{k}_2 &= h\mathbf{f}(\mathbf{x}_n + \mathbf{k}_1/2, t_n + h/2) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{k}_2 + \mathcal{O}(h^3) \end{aligned} \tag{3.25}$$

Cela revient à utiliser un prédicteur pour évaluer la dérivée au milieu de l'intervalle $[t_n, t_{n+1}]$ et à utiliser cette dérivée pour calculer la valeur de $\mathbf{x}_{n+1}$.

C.2 Méthode du quatrième ordre

La version la plus utilisée de la méthode de Runge-Kutta est celle du quatrième ordre, dans laquelle l'erreur commise est d'ordre h^5. La formule aux différences correspondante est la suivante :

$$\begin{aligned} \mathbf{k}_1 &= h\mathbf{f}(\mathbf{x}_n, t_n) \\ \mathbf{k}_2 &= h\mathbf{f}(\mathbf{x}_n + \mathbf{k}_1/2, t_n + h/2) \\ \mathbf{k}_3 &= h\mathbf{f}(\mathbf{x}_n + \mathbf{k}_2/2, t_n + h/2) \\ \mathbf{k}_4 &= h\mathbf{f}(\mathbf{x}_n + \mathbf{k}_3, t_n + h) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \tfrac{1}{6}\mathbf{k}_1 + \tfrac{1}{3}\mathbf{k}_2 + \tfrac{1}{3}\mathbf{k}_3 + \tfrac{1}{6}\mathbf{k}_4 + \mathcal{O}(h^5) \end{aligned} \tag{3.26}$$

Voyons le sens de chacune de ces étapes :

1. Pour commencer, on évalue la dérivée $\mathbf{k}_1/h$ au point $(\mathbf{x}_n, t_n)$.
2. On utilise cette dérivée pour obtenir un premier point médian $\mathbf{x}_n + \mathbf{k}_1/2$.
3. On calcule ensuite la dérivée $\mathbf{k}_2/h$ à ce point médian.
4. On calcule une deuxième estimation $\mathbf{x}_n + \mathbf{k}_2/2$ de ce point médian et on y calcule encore une fois la dérivée $\mathbf{k}_3/h$.
5. On calcule ensuite la dérivée $\mathbf{k}_4/h$ à une première estimation du point final $\mathbf{x}_n + \mathbf{k}_3$.
6. Enfin, le point final estimé par la méthode est obtenu par une combinaison des quatre dérivées calculées aux étapes précédentes.

C.3 Contrôle du pas dans la méthode de Runge-Kutta

Un solutionneur d'équations différentielles qui procède en suivant un pas temporel h constant est condamné soit à l'inefficacité (h trop petit), soit à commettre des erreurs de troncature non contrôlées (h trop grand). Il est impératif que le pas temporel s'adapte à chaque instant à l'erreur de troncature commise à chaque étape. Cette dernière peut être estimée en procédant à deux étapes de Runge-Kutta du quatrième ordre, chacune de pas $h/2$, et en comparant le résultat $\mathbf{x}_{n+1}$ obtenu après la deuxième étape à celui obtenu en procédant directement à une étape de pas h. Si la différence est inférieure à une précision demandée à l'avance, on peut alors augmenter la valeur de h pour la prochaine itération, sinon on peut la réduire.

Des calculs supplémentaires sont bien sûr nécessaires pour évaluer l'erreur de troncature, mais l'algorithme au total demande moins de calculs, car son erreur est *contrôlée* : le pas h peut augmenter

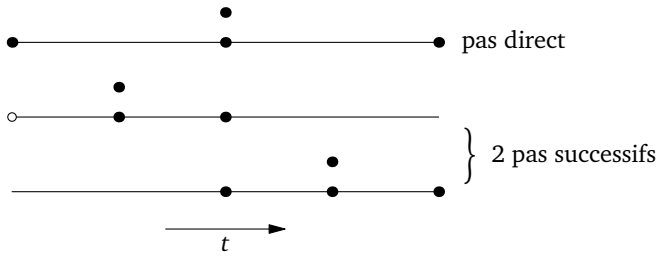


FIGURE 3.1
Nombre d'évaluation des dérivées dans une étape de la méthode de Runge-Kutta adaptative. Un cercle noir représente une évaluation des dérivées et un cercle blanc une dérivée qui a été gardée en mémoire.

si l'erreur est tolérable, et de cette manière alléger le coût des calculs. D'un autre côté, si h doit être diminué, c'est que les calculs auraient autrement été erronés à un degré jugé inacceptable.

Une façon élémentaire de procéder au contrôle du pas est la méthode du *doublement* :

1. On effectue un pas temporel h à l'aide de la méthode du quatrième ordre (`RK4`). On obtient une variable mise à jour $\mathbf{x}_{n+1}^{(1)}$.
2. On effectue deux pas temporels successifs $h/2$, à partir du même point de départ, pour arriver à une valeur $\mathbf{x}_{n+1}^{(2)}$.
3. La différence $|\mathbf{x}_{n+1}^{(2)} - \mathbf{x}_{n+1}^{(1)}|$ nous donne une estimation de l'erreur de troncature Δ_1. Nous pouvons par la suite modifier h (en l'augmentant ou le diminuant) pour viser une erreur de troncature constante Δ_0. En posant que l'erreur a la forme $\Delta = Ch^5$, où C est une constante indépendante de h, on trouve, pour deux valeurs différentes h_0 et h_1, la relation

$$\frac{\Delta_0}{\Delta_1} = \frac{h_0^5}{h_1^5} \quad \text{et donc} \quad h_0 = h_1 \left(\frac{\Delta_0}{\Delta_1} \right)^{1/5}$$

   À partir d'une erreur Δ_1, nous devons donc réajuster h ainsi afin d'espérer obtenir une erreur Δ_0 :

$$h \to h \left(\frac{\Delta_0}{\Delta_1} \right)^{1/5} \tag{3.27}$$

   On peut aussi multiplier par un facteur de sécurité (par exemple 0.9) afin de tempérer quelque peu l'augmentation de h, sans jamais faire plus que de quadrupler le pas.
4. On note que le nombre d'évaluations de la dérivée dans cette procédure est de 11 par étape, alors que la méthode utilisée avec un pas de $h/2$ (qui nous donne une précision semblable) nécessite 8 évaluations. Il y a donc un coût (surcharge) de $\frac{11}{8} = 1.375$ dans l'estimation de l'erreur. Cependant, ce coût est largement recouvré par le contrôle accru de la méthode.

Quoique la méthode du doublement soit simple à comprendre, elle est maintenant dépassée par la méthode dite de *Runge-Kutta-Fehlberg*, qui effectue une étape du 5e ordre en même temps qu'une étape du 4e ordre et compare les deux afin d'estimer l'erreur de troncature (la méthode est souvent désignée par son acronyme RKF45). Au total, le nombre d'évaluations nécessaire par étape est de 6, soit presque deux fois moins que la méthode de doublement. [2]

2. Voir *Numerical Recipes*, ou encore les articles de Wikipédia écrits sur le sujet.

D Méthode de Richardson

Une façon ingénieuse d'augmenter la précision d'une méthode est de procéder, sur un intervalle H donné, à une subdivision en m sous-intervalles équidistants. On traite ensuite le problème pour chacun des sous-intervalles de dimension $h = \mathrm{H}/m$, de manière à obtenir un prédicteur $\mathbf{x}(t+\mathrm{H})$ pour quelques valeurs de m (par exemple m =2, 4, 6) et on extrapole vers $m \to \infty$ (ou $h \to 0$). La première étape consiste à construire un prédicteur pour une valeur donnée de m. La deuxième étape, qui applique une extrapolation vers $h \to 0$, suppose que le prédicteur $\mathbf{x}(t+\mathrm{H})$ est une fonction analytique de H autour de $\mathrm{H} = 0$, et qu'une extrapolation, par exemple polynomiale, nous donne accès à cette limite. Le tout constitue la *méthode de Richardson*.

On se trouve ici à travailler avec un pas H/m fixe dans un intervalle donné, mais un contrôle de précision peut aussi être appliqué subséquemment à la valeur de H elle-même, de manière à s'adapter à une précision requise.

Une façon simple de procéder avec m sous-intervalles est la méthode aux différences modifiées (angl. *modified midpoint method*), qui évalue la dérivée à partir de l'instant suivant et de l'instant précédent (sauf aux extrémités) :

$$\begin{aligned}
\tilde{\mathbf{x}}_0 &= \mathbf{x}_n \\
\tilde{\mathbf{x}}_1 &= \tilde{\mathbf{x}}_0 + h\mathbf{f}(\tilde{\mathbf{x}}_0, t) \qquad\qquad h \coloneqq \frac{\mathrm{H}}{m} \\
\tilde{\mathbf{x}}_2 &= \tilde{\mathbf{x}}_0 + 2h\mathbf{f}(\tilde{\mathbf{x}}_1, t+h) \\
&\cdots \\
\tilde{\mathbf{x}}_{j+1} &= \tilde{\mathbf{x}}_{j-1} + 2h\mathbf{f}(\tilde{\mathbf{x}}_j, t+jh) \qquad\qquad j = 1, 2, \ldots, m-1 \\
&\cdots \\
\mathbf{x}(t+\mathrm{H}) \approx \mathbf{x}_{n+1} &= \tfrac{1}{2}(\tilde{\mathbf{x}}_m + \tilde{\mathbf{x}}_{m-1} + h\mathbf{f}(\tilde{\mathbf{x}}_m, t+\mathrm{H}))
\end{aligned} \tag{3.28}$$

Nous avons introduit une série de prédicteurs $\tilde{\mathbf{x}}_j$ désignant la j^{e} valeur dans le n^{e} sous-intervalle. Cette méthode requiert de conserver en mémoire non seulement la valeur courante de $\tilde{\mathbf{x}}_j$, mais la valeur précédente $\tilde{\mathbf{x}}_{j-1}$, ce qui n'est pas un problème. Par contre, les premières et dernières valeurs ($j = 0$ et $j = m$) requièrent un traitement spécial. La dernière valeur $\tilde{\mathbf{x}}_m$ sert de prédicteur pour $\mathbf{x}(t+\mathrm{H})$, qui est ensuite amélioré à la dernière équation ci-dessus. [3]

Suite à ce calcul pour différentes valeurs de m, on peut procéder à une extrapolation pour obtenir une estimation optimale de $\mathbf{x}_n$, et ensuite passer à l'intervalle H suivant. La méthode de *Bulirsch-Stoer* propose une façon particulière de réaliser cette extrapolation, et pour modifier la valeur de H au besoin (pas adaptatif). Les valeurs de m utilisées dans la méthode de Bulirsch-Stoer peuvent facilement atteindre la centaine ; donc le pas H est généralement assez grand, et correspond au pas d'affichage des résultats. Voir *Numerical Recipes* pour plus de détails.

3. Si cette dernière équation peut sembler mystérieuse, elle garantit par contre que l'erreur théorique commise par cette méthode, lorsque développée en puissances de h, ne contient que des termes de degré pair, c'est-à-dire des puissances entières de h^2.

E Méthode d'Adams

L'idée de base derrière la méthode d'Adams est d'utiliser les s points précédents d'une solution numérique afin de prédire le prochain point, à l'aide d'une extrapolation polynomiale. Il existe deux variantes :

1. la méthode d'Adams-Bashforth, qui prédit $\mathbf{x}_{n+1}$ à l'aide des s points $\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{x}_{n-s+1}$ et des dérivées correspondantes $\mathbf{f}_i \coloneqq \mathbf{f}(\mathbf{x}_i, t_i)$. C'est une méthode d'ordre s.
2. La méthode d'Adams-Moulton, qui prédit $\mathbf{x}_{n+1}$ à l'aide, en plus des s points précédents, d'un prédicteur $\mathbf{x}_{n+1}^{\text{pr.}}$ sur la valeur de $\mathbf{x}_{n+1}$. C'est une méthode d'ordre $s+1$.

E.1 Méthode d'Adams-Bashforth

Pour $s = 1$, la méthode d'Adams-Bashforth coïncide avec la méthode d'Euler. Voyons ce qu'elle donne pour $s = 2$. Afin de simplifier la notation, posons $n = 1$, de sorte qu'on vise à prédire $\mathbf{x}_2$ à l'aide de $\mathbf{x}_1$, $\mathbf{x}_0$, $\mathbf{f}_1$ et $\mathbf{f}_0$. On suppose en outre que les temps t_i sont disposés uniformément, avec un pas temporel h. On remplace la fonction $\mathbf{f}(\mathbf{x}(t), t)$, qui est inconnue, car $\mathbf{x}(t)$ n'est pas connu explicitement, par une extrapolation linéaire obtenue des deux points précédents :

$$\begin{aligned}\mathbf{f}(\mathbf{x}(t), t) \to \mathbf{P}_1(t) &= \frac{t-t_1}{t_0-t_1}\mathbf{f}_0 + \frac{t-t_0}{t_1-t_0}\mathbf{f}_1 \\ &= \frac{1}{h}\left[-(t-t_1)\mathbf{f}_0 + (t-t_0)\mathbf{f}_1\right] = \frac{1}{h}\left[t(\mathbf{f}_1-\mathbf{f}_0) + t_1\mathbf{f}_0 - t_0\mathbf{f}_1\right]\end{aligned} \tag{3.29}$$

Ensuite, on calcule simplement l'intégrale de cette fonction pour obtenir $\mathbf{x}(t_2)$:

$$\begin{aligned}\mathbf{x}_2 &= \mathbf{x}_1 + \int_{t_1}^{t_2} \mathrm{P}_1(t)\,\mathrm{d}t \\ &= \mathbf{x}_1 + \frac{1}{2h}(\mathbf{f}_1-\mathbf{f}_0)(t_2^2-t_1^2) + \frac{1}{h}(t_1\mathbf{f}_0 - t_0\mathbf{f}_1)(t_2-t_1) \\ &= \mathbf{x}_1 + \frac{1}{2}(\mathbf{f}_1-\mathbf{f}_0)(t_2+t_1) + t_1\mathbf{f}_0 - t_0\mathbf{f}_1 \\ &= \mathbf{x}_1 + \frac{1}{2}\mathbf{f}_1(t_2+t_1-2t_0) + \frac{1}{2}\mathbf{f}_0(t_1-t_2) \\ &= \mathbf{x}_1 + \frac{3}{2}h\mathbf{f}_1 - \frac{1}{2}h\mathbf{f}_0\end{aligned} \tag{3.30}$$

On constate qu'on obtient un autre cas particulier de la méthode générale du deuxième ordre, avec $a = \frac{3}{2}$, $b = -\frac{1}{2}$, $\alpha = -1$ et $\beta = -1$.

À l'ordre $s = 3$, on devrait utiliser les trois derniers points $\mathbf{x}_{0,1,2}$, et prédire $\mathbf{x}_4$ à l'aide du polynôme d'ordre deux unique qui passe par ces trois points, donné par la formule de Lagrange (éq. (9.1), à la page 73) :

$$f(\mathbf{x}(t), t) \to \mathrm{P}_2(t) = \frac{(t-t_1)(t-t_2)}{(t_0-t_1)(t_0-t_2)}\mathbf{f}_0 + \frac{(t_0-t)(t-t_2)}{(t_0-t_1)(t_1-t_2)}\mathbf{f}_1 + \frac{(t_1-t)(t_0-t)}{(t_1-t_2)(t_0-t_2)}\mathbf{f}_2 \tag{3.31}$$

et ainsi de suite, pour les ordres supérieurs.

Au total, pour les cinq premiers ordres, on obtient les règles suivantes :

$$s=1 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+h\mathbf{f}_n \qquad \text{(Euler)} \tag{3.32}$$

$$s=2 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{2}\left(3\mathbf{f}_n-\mathbf{f}_{n-1}\right) \tag{3.33}$$

$$s=3 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{12}\left(23\mathbf{f}_n-16\mathbf{f}_{n-1}+5\mathbf{f}_{n-2}\right) \tag{3.34}$$

$$s=4 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{24}\left(55\mathbf{f}_n-59\mathbf{f}_{n-1}+37\mathbf{f}_{n-2}-9\mathbf{f}_{n-3}\right) \tag{3.35}$$

$$s=5 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{720}\left(1901\mathbf{f}_n-2774\mathbf{f}_{n-1}+2616\mathbf{f}_{n-2}-1274\mathbf{f}_{n-3}+251\mathbf{f}_{n-4}\right) \tag{3.36}$$

E.2 Méthode d'Adams-Moulton

La méthode d'Adams-Moulton utilise les s points précédents et le point désiré $\mathbf{x}_{n+1}$ dans l'extrapolation. Considérons par exemple le cas $s=1$. Reprenons la formule (3.29) ci-dessus, sauf que cette fois $n=0$ et $\mathbf{x}_1$ est inconnu. Dans ce cas,

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{x}_0+\int_{t_0}^{t_1}\mathbf{P}_1(t)\,\mathrm{d}t \\ &= \mathbf{x}_0+\frac{1}{2h}(\mathbf{f}_1-\mathbf{f}_0)(t_1^2-t_0^2)+\frac{1}{h}(t_1\mathbf{f}_0-t_0\mathbf{f}_1)(t_1-t_0) \\ &= \mathbf{x}_0+\frac{1}{2}(\mathbf{f}_1-\mathbf{f}_0)(t_1+t_0)+t_1\mathbf{f}_0-t_0\mathbf{f}_1 \\ &= \mathbf{x}_0+\frac{h}{2}(\mathbf{f}_1+\mathbf{f}_0)\end{aligned} \tag{3.37}$$

On reconnait dans cette dernière équation la méthode prédicteur-correcteur (3.17), pourvu que la valeur de $\mathbf{x}_1$ utilisée dans le membre de droite soit un prédicteur, par exemple obtenu par la méthode d'Euler.

Pour les quatre premiers ordres, on obtient les règles suivantes par cette méthode :

$$s=1 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{2}\left(\mathbf{f}_{n+1}+\mathbf{f}_n\right) \tag{3.38}$$

$$s=2 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{12}\left(5\mathbf{f}_{n+1}+8\mathbf{f}_n-\mathbf{f}_{n-1}\right) \tag{3.39}$$

$$s=3 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{24}\left(9\mathbf{f}_{n+1}+19\mathbf{f}_n-5\mathbf{f}_{n-1}+\mathbf{f}_{n-2}\right) \tag{3.40}$$

$$s=4 \qquad \mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{720}\left(241\mathbf{f}_{n+1}+646\mathbf{f}_n-264\mathbf{f}_{n-1}+106\mathbf{f}_{n-2}-19\mathbf{f}_{n-3}\right) \tag{3.41}$$

En pratique, la valeur de $\mathbf{f}_{n+1}$ doit être obtenue par une méthode explicite, à l'ordre précédent, soit par la méthode d'Adams-Bashforth au même s. Ainsi, en substituant (3.32) dans (3.38) pour la valeur de $\mathbf{x}_{n+1}$ dans $\mathbf{f}_{n+1}$, on obtient la méthode prédicteur-correcteur (3.17). En substituant plutôt la valeur (3.35) de $\mathbf{x}_{n+1}$ dans (3.41), on obtient une méthode d'ordre 5 qui ne demande que deux évaluations de la fonction $\mathbf{f}$ par étape! Cependant, cette méthode demande un pas temporel h constant.

CHAPITRE 4

SIMULATION DE PARTICULES I : MÉTHODE DE VERLET

L'une des applications les plus répandues du calcul scientifique est la simulation du mouvement d'un grand nombre de particules sous l'influence de forces mutuelles et externes. L'objectif de ces simulations est typiquement de comprendre le comportement statistique de la matière, d'où le nom *dynamique moléculaire* donné à ce champ d'applications. Même s'il est en principe préférable d'utiliser la mécanique quantique pour décrire le mouvement des atomes et des molécules, il est acceptable d'utiliser la mécanique classique pour ce faire si les longueurs d'onde impliquées sont suffisamment petites par rapport aux distances intermoléculaires. D'autres applications de ce genre sont aussi très éloignées du domaine quantique, par exemple en astronomie.

A Algorithme de Verlet

La dynamique moléculaire est un domaine vaste ; il est hors de question de lui rendre justice dans cette section. Nous allons uniquement décrire l'*algorithme de Verlet*, utilisé pour résoudre les équations du mouvement des particules impliquées. Au fond, il s'agit ici de résoudre un système d'équations différentielles, représentant les équations du mouvement de Newton, mais à un nombre plutôt grand de particules. Pourquoi ne pas simplement utiliser la méthode de Runge-Kutta avec pas contrôlé ? C'est une possibilité, mais l'algorithme que nous présenterons plus bas est plus simple, du deuxième ordre, et se compare à la méthode prédicteur-correcteur. La solution ne sera peut-être pas aussi précise que celle obtenue par Runge-Kutta à pas adapté, mais le calcul sera par contre plus rapide. [1] Notons que l'intérêt de la simulation n'est pas ici de suivre à la trace chaque particule, mais de dégager le comportement de l'ensemble.

Soit donc un ensemble de N particules, de positions $\mathbf{r}_i$ et de vitesses $\mathbf{v}_i$. Chaque particule ressent une force $\mathbf{F}_i$ qui dépend en principe le la position de toutes les particules, ainsi que de la vitesse $\mathbf{v}_i$, si on veut tenir compte de processus d'amortissement (par exemple le rayonnement ou la diffusion de chaleur). Nous avons donc à résoudre le système d'équations différentielles suivant pour les 2N vecteurs $\mathbf{r}_i$ et $\mathbf{v}_i$:

$$\frac{\mathrm{d}\mathbf{r}_i}{\mathrm{d}t} = \mathbf{v}_i \qquad \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \frac{1}{m_i}\mathbf{F}_i(\mathbf{r}_j, \mathbf{v}_i, t) \tag{4.1}$$

1. Notez qu'il est possible d'utiliser un pas temporel variable, mais que ce raffinement ne sera pas expliqué ici.

La méthode de Verlet est basée sur une formule du deuxième ordre pour l'évaluation des dérivées :

$$\boxed{\begin{aligned}\mathbf{r}_i(t+h) &= \mathbf{r}_i(t) + h\mathbf{v}_i(t) + \frac{h^2}{2m_i}\mathbf{F}_i(t) + \mathcal{O}(h^4)\\ \mathbf{v}_i(t+h) &= \mathbf{v}_i(t) + \frac{h}{2m_i}\Big[\mathbf{F}_i(t+h) + \mathbf{F}_i(t)\Big] + \mathcal{O}(h^3)\end{aligned}} \tag{4.2}$$

L'important ici est que toutes les positions doivent être mises à jour avant que les vitesses le soient, car la force $\mathbf{F}_i(t+h)$ doit être calculée entretemps. Ceci suppose que la force ne dépend que des positions. Noter que nous avons supposé que la masse m_i peut être différente d'une particule à l'autre.

On peut aussi de cette manière traiter d'une force qui dépend linéairement de la vitesse, par exemple dans le but de modéliser un amortissement ou, éventuellement, l'effet d'un champ magnétique sur des particules chargées. Voyons comment procéder dans le cas d'une force d'amortissement $-m\gamma\mathbf{v}$ qui s'oppose linéairement à la vitesse (γ est la force d'amortissement par unité de masse). Afin d'avoir une évolution valable au deuxième ordre, on doit exprimer cette force en fonction de la vitesse moyenne aux temps t et $t+h$:

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) + \frac{h}{2m_i}\left[\mathbf{F}_i(t+h) + \mathbf{F}_i(t)\right] - \frac{h\gamma}{2}\left[\mathbf{v}_i(t+h) + \mathbf{v}_i(t)\right] \tag{4.3}$$

($\mathbf{F}_i$ représente les forces qui ne dépendent que de la position). On peut ensuite isoler $\mathbf{v}_i(t+h)$, à la manière d'un schéma implicite, et on trouve

$$\mathbf{v}_i(t+h) = \frac{1}{1+\dfrac{h\gamma}{2}}\left\{\left(1-\frac{h\gamma}{2}\right)\mathbf{v}_i(t) + \frac{h}{2m_i}\left[\mathbf{F}_i(t+h) + \mathbf{F}_i(t)\right]\right\} \tag{4.4}$$

A.1 Exemple : force constante

Nous allons étudier par la méthode de Verlet un ensemble de particules exerçant les unes sur les autres une force centrale constante en deçà d'un certain domaine. Cette force sera dérivée du potentiel illustré à la figure 4.1, et aura la forme suivante :

$$\mathrm{F}(r) = \begin{cases} \dfrac{\mathrm{V}_0 - \mathrm{V}_1}{r_0} & \text{si} \quad r < r_0 \\ \dfrac{\mathrm{V}_1}{r_m - r_0} & \text{si} \quad r_0 < r < r_m \\ 0 & \text{si} \quad r > r_m \end{cases} \tag{4.5}$$

Cette loi de force est une caricature simple d'une attraction à distances modérées, et d'une répulsion à courte distance, avec une distance d'équilibre r_0. Un potentiel d'interaction plus réaliste serait le célèbre potentiel de Lennard-Jones :

$$\mathrm{V}(r) = 4\mathrm{V}_0\left[\left(\frac{r_0}{r}\right)^{12} - \left(\frac{r_0}{r}\right)^{6}\right] \tag{4.6}$$

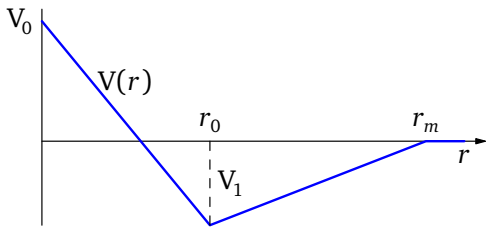


FIGURE 4.1

B Complexité algorithmique des simulations de particules

La méthode de Verlet que nous avons présentée a, dans sa version simple, un grave défaut : le temps de calcul des forces est proportionnel au nombre de paires de particules, soit $\frac{1}{2}\mathrm{N}(\mathrm{N}-1)$, qui se comporte comme $\mathcal{O}(\mathrm{N}^2)$ quand N est grand. Si on désire simuler un très grand nombre de particules, l'évaluation des forces va simplement devenir trop onéreuse et la simulation impossible.

La solution, bien sûr, est que les forces décroissent rapidement avec la distance. Donc il n'est pas nécessaire de calculer tous les détails des forces pour des particules qui sont éloignées l'une de l'autre. Deux possibilités générales se présentent :

1. Si la force est à courte portée, en particulier si elle décroît plus rapidement que $1/r^2$ en dimension trois, alors on peut négliger les particules qui sont trop éloignées. Supposons aux fins de l'argument qu'on peut négliger toutes les forces mutuelles au-delà d'une certaine distance R. C'est effectivement ce qui est fait dans le code présenté ci-haut. Cependant, ce code doit tout de même effectuer une boucle sur toutes les paires de particules pour vérifier si les distances sont inférieures à R, ce qui ne règle pas le problème de la complexité algorithmique d'ordre N^2. Pour s'en sortir, il faut une représentation des données différentes, dans laquelle on peut avoir accès directement aux particules qui sont dans une région donnée, sans avoir à chercher le tableau contenant les positions des particules. Par exemple, on pourrait diviser l'espace en un réseau de cellules identiques de taille R, et construire une liste dynamique des particules qui résident à un instant donné dans chaque cellule. Dans le calcul des forces, nous n'aurions alors qu'à considérer les particules qui résident dans une cellule donnée et les cellules immédiatement voisines. La complexité algorithmique serait réduite à $\mathcal{O}(\mathrm{M}n^2) = \mathcal{O}(\mathrm{N}n)$, $\mathrm{M} = \mathrm{N}/n$ étant le nombre de cellules et n le nombre de particules par cellule. Un tel schéma demande bien sûr à garder trace des particules qui passent d'une cellule à l'autre lors d'un pas temporel. En pratique, on définirait des cellules qui se chevauchent sur une distance égale à la portée de la force, comportant chacune une région intérieure et une région périphérique : il ne serait alors pas requis de sortir de la cellule pour calculer les forces et seules les particules de la région intérieure seraient propagées : celles de la région périphérique étant en même temps dans la région intérieure d'une cellule voisine, elles ne servent dans la cellule courante qu'au calcul des forces.
2. Si les forces sont à longue portée, comme la force gravitationnelle, alors on ne peut pas simplement ignorer les particules éloignées. La raison étant que même si la force décroît comme $1/r^2$, le nombre de particules situées à une distance d'ordre r d'un point donné croît comme r^2 quand la densité est du même ordre partout ; donc on commettrait une erreur grave en

ignorant les particules éloignées. Cependant, on peut appliquer dans ce cas le développement multipolaire et considérer l'influence de groupes de particules et non de particules individuelles. Une façon de procéder est de construire un réseau de cellules comme au cas précédent, mais en plus de les organiser en une hiérarchie de super-cellules à plusieurs niveaux : les cellules de taille R sont contenues dans des super-cellules de taille 3R, elles-mêmes contenues dans des super-cellules de taille 3^2R, et ainsi de suite. Dans le calcul des forces, on calcule quelques multipôles produits par chaque cellule d'un niveau donné, qui ensuite servent à calculer les multipôles du niveau supérieur de cellules, etc. De cette manière, la complexité algorithmique du problème passe de $\mathcal{O}(N^2)$ à $\mathcal{O}(N \log N)$ (le nombre de niveaux de cellules étant d'ordre log(N). Ce qui présente est une caricature de ce qui est en fait accompli dans ces algorithmes dits de *multipôles rapides*, mais l'idée de base est la même.

C Aspects quantiques et statistiques

Dans le domaine microscopique, on peut se demander si la mécanique classique est appropriée pour décrire le mouvement des molécules. En réalité, seul le mouvement des ions est déterminé classiquement ; les forces intermoléculaires dépendent cependant des configurations électroniques qui, elles, ne peuvent être déterminées que par la mécanique quantique. Les véritables simulations de dynamique moléculaire tiennent donc compte de la mécanique quantique, à un certain degré d'approximation, dans le calcul des forces. Une avancée considérable a été accomplie dans ce domaine par Car et Parinello en 1985 avec l'introduction d'une méthode rapide permettant d'avancer dans les temps les configurations électroniques en même temps que les positions des ions. Lorsqu'un calcul quantique – même approximatif – des forces inter ioniques est effectué, on parle de *dynamique moléculaire ab initio*. Sinon, il s'agit de *dynamique moléculaire classique*.

Les systèmes simples qui sont simulés par le code décrit plus haut sont décrits par l'ensemble microcanonique : le nombre de particules et l'énergie du système sont constants (sauf si on inclut un amortissement). Il peut être cependant plus réaliste, en simulant un système complexe, de supposer que ce système est lui-même un sous-ensemble d'un système encore plus grand, avec lequel il échange de l'énergie et, à la rigueur, des particules. Autrement dit, on peut simuler le système en le mettant en contact avec un environnement caractérisé par une température T (ensemble canonique). La simulation doit alors comporter une procédure pour communiquer aux particules qui parviennent à la frontière du domaine étudié une signature de leur interaction avec l'environnement, par exemple une variation de l'énergie, caractéristique de l'ensemble canonique. On peut aussi ajouter une force aléatoire agissant sur chaque particule qui, combinée à un terme d'amortissement γ, permet de reproduire une distribution canonique (dynamique de Langevin).

CHAPITRE 5

TRANSFORMÉES DE FOURIER RAPIDES

A Introduction

Plusieurs méthodes numériques reposent sur notre capacité à calculer rapidement des transformées de Fourier. Celles-ci sont généralement utiles dans les problèmes où il y a invariance par translation dans l'espace ou dans le temps, de sorte que la représentation en vecteur d'onde ou en fréquence est, en quelque sorte, plus simple. La pièce de résistance de ce chapitre est l'algorithme de transformée de Fourier rapide (TdFR), considéré comme l'un des algorithmes les plus importants du calcul numérique.

Une fonction complexe $\psi(x)$ définie sur un intervalle de longueur L peut être représentée par ses coefficients de Fourier $\tilde{\psi}_k$, définis comme suit :

$$\psi(x) = \sum_{k\in\mathbb{Z}} \mathrm{e}^{2\pi i k x/\mathrm{L}} \tilde{\psi}_k \qquad \tilde{\psi}_k = \frac{1}{\mathrm{L}} \int_{-\mathrm{L}/2}^{\mathrm{L}/2} \mathrm{d}x\ \mathrm{e}^{-2\pi i k x/\mathrm{L}} \psi(x) \tag{5.1}$$

Cette représentation de la fonction $\psi(x)$ est explicitement périodique de période L. Dans la limite $\mathrm{L} \to \infty$, on définit une variable continue $q = 2\pi k/\mathrm{L}$ et les relations ci-dessus deviennent la transformation de Fourier (TdF) :

$$\psi(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{d}q\ \mathrm{e}^{iqx} \tilde{\psi}(q) \quad \text{et} \quad \tilde{\psi}(q) = \int_{-\infty}^{\infty} \mathrm{d}x\ \mathrm{e}^{-iqx} \psi(x) \tag{5.2}$$

où $\tilde{\psi}_n \to \frac{1}{\mathrm{L}} \tilde{\psi}(2\pi k/\mathrm{L})$.

Bien que la notation utilisée fasse référence à une interprétation spatiale de la transformation, ce qui suit est bien sûr également valable dans le domaine temps-fréquence ; il suffit d'adapter la notation en conséquence.

B Transformées de Fourier discrètes

Numériquement, une fonction $\psi(x)$ sera représentée par un ensemble de N valeurs ψ_j ($j = 0,\dots,\mathrm{N}-1$) associées à une grille régulière de pas a et d'étendue $\mathrm{L} = \mathrm{N}a$. L'équivalent dans ce cas de la relation (5.1) est la *transformée de Fourier discrète* :

$$\psi_j = \frac{1}{\mathrm{N}} \sum_{k=0}^{\mathrm{N}-1} \mathrm{e}^{2\pi i jk/\mathrm{N}} \tilde{\psi}_k \qquad\qquad \tilde{\psi}_k = \sum_{j=0}^{\mathrm{N}-1} \mathrm{e}^{-2\pi i jk/\mathrm{N}} \psi_j \tag{5.3}$$

On peut également exprimer ces relations en fonction de la N^{me} racine complexe de l'unité $\omega \coloneqq \mathrm{e}^{-2\pi i/\mathrm{N}}$:

$$\psi_j = \frac{1}{\mathrm{N}} \sum_{k=0}^{\mathrm{N}-1} \bar{\omega}^{jk} \tilde{\psi}_k \qquad\qquad \tilde{\psi}_k = \sum_{j=0}^{\mathrm{N}-1} \omega^{jk} \psi_j \qquad\qquad \bar{\omega} \coloneqq \omega^* \tag{5.4}$$

Remarques :

- La quantité ψ_j est périodique de période N : $\psi_{j+\mathrm{N}}$ doit être identifié à ψ_j.
- Idem pour $\tilde{\psi}_k$, aussi périodique de période N.
- La correspondance avec une fonction continue est la suivante :

$$x \to ja \qquad\qquad q \to \frac{2\pi k}{\mathrm{L}} \qquad\qquad \psi_j \to \frac{a}{\mathrm{L}} \psi(x) \tag{5.5}$$

  La série de Fourier habituelle se retrouve dans la limite $a \to 0$, $\mathrm{N} \to \infty$, en gardant $\mathrm{L} = \mathrm{N}a$ constant. La transformée de Fourier habituelle se retrouve en faisant également tendre $\mathrm{L} \to \infty$.

La transformée de Fourier discrète peut être vue comme l'application d'une matrice $\mathrm{N} \times \mathrm{N}$ sur le vecteur ψ :

$$\tilde{\psi} = \mathrm{U}\psi \quad \text{où} \quad \mathrm{U}_{jk} = \omega^{jk} \tag{5.6}$$

Par exemple, pour $\mathrm{N} = 8$, cette matrice est la suivante ($\omega^4 = -1$) :

$$\mathrm{U} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & -1 & -\omega & -\omega^2 & -\omega^3 \\ 1 & \omega^2 & -1 & -\omega^2 & 1 & \omega^2 & -1 & -\omega^2 \\ 1 & \omega^3 & -\omega^2 & \omega & -1 & -\omega^3 & \omega^2 & -\omega \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\omega & \omega^2 & -\omega^3 & -1 & \omega & -\omega^2 & \omega^3 \\ 1 & -\omega^2 & -1 & \omega^2 & 1 & -\omega^2 & -1 & \omega^2 \\ 1 & -\omega^3 & -\omega^2 & -\omega & -1 & \omega^3 & \omega^2 & \omega \end{pmatrix} \tag{5.7}$$

La relation inverse, $\psi = \frac{1}{\mathrm{N}} \mathrm{U}^* \tilde{\psi}$, provient de l'identité suivante :

$$\mathrm{U}^\dagger \mathrm{U} = \mathrm{N} \quad \text{ou encore} \quad \boxed{\sum_{k=0}^{\mathrm{N}-1} \omega^{(j-j')k} = \mathrm{N}\delta_{jj'}} \tag{5.8}$$

La preuve de cette relation est très simple, car en général, pour tout complexe z,

$$\sum_{k=0}^{N-1} z^k = \frac{1-z^N}{1-z} \quad \text{et donc} \quad \sum_{k=0}^{N-1} \omega^{(j-j')k} = \frac{1-\omega^{(j-j')N}}{1-\omega^{(j-j')}} \tag{5.9}$$

Mais le numérateur de cette expression est toujours nul, car $\omega^N = 1$. Le dénominateur n'est nul que si $j = j'$; dans ce dernier cas, la fraction est indéterminée, mais il est trivial d'évaluer directement la somme, qui donne manifestement N, d'où le résultat (5.8).

C Algorithme de Danielson et Lanczos (ou Cooley-Tukey)

À première vue, il semble que le nombre d'opérations impliquées dans une TdF discrète soit d'ordre $\mathcal{O}(N^2)$, comme tout produit matrice vecteur. Or il est possible d'effectuer la TdF discrète à l'aide d'un nombre d'opérations beaucoup plus petit, d'ordre $\mathcal{O}(N \log N)$. [1] Cette différence est considérable : si un signal est échantillonné à l'aide de 1 000 points, ce qui n'est pas énorme, le rapport entre N^2 et $N \log N$ est de l'ordre de 100. Le gain de performance est énorme. Les méthodes qui tirent parti de cette réduction dans le nombre d'opérations portent le nom de *transformées de Fourier rapides* (TdFR, en anglais *fast Fourier transforms*, ou FFT). L'algorithme de TdFR est l'un des plus utilisés dans la vie courante : traitement du son et de l'image, etc. Les téléphones mobiles partout dans le monde effectuent des TdFR constamment. La TdFR a été reconnue comme l'un des dix algorithmes les plus importants de l'histoire du calcul. [2]

C.1 Description de l'algorithme

La version la plus simple de l'algorithme de TdFR se présente lorsque le nombre de points est une puissance de deux : $N = 2^M$. C'est ce que nous allons supposer dans la suite. La clé de l'algorithme est la possibilité d'écrire la deuxième des relations (5.4) comme la somme de deux transformées de Fourier comportant N/2 points : l'une comportant les termes pairs, l'autre les termes impairs :

$$\tilde{\psi}_k = \sum_{j=0}^{N/2-1} (\omega^2)^{jk} \psi_{2j} + \omega^k \sum_{j=0}^{N/2-1} (\omega^2)^{jk} \psi_{2j+1} \tag{5.10}$$

Cela entraîne qu'une TdF à N points est la somme de deux TdF à N/2 points, si on arrange le vecteur ψ à N composantes en deux vecteurs $\psi^{(0)}$ et $\psi^{(1)}$ de N/2 composantes chacun, contenant respectivement les composantes paires et impaires de ψ. Dans chacune des deux TdF, la valeur de k doit être

1. Ce fait a été découvert plusieurs fois depuis l'époque de Gauss, en particulier par Danielson et Lanczos en 1942. Ce sont cependant les noms de Cooley et Tukey qui sont habituellement associés à cette découverte (1965).
2. Par la revue *Computing in Science and Engineering*, Jan/Feb 2000.

prise modulo N/2, sauf dans l'exposant ω^k. Cependant, on peut grouper $\tilde{\psi}_k$ et $\tilde{\psi}_{k+\mathrm{N}/2}$ ensemble et exprimer la TdF ainsi :

$$\begin{aligned}\tilde{\psi}_k &= \sum_{j=0}^{\mathrm{N}/2-1} (\omega^2)^{jk}\psi_{2j} + \omega^k \sum_{j=0}^{\mathrm{N}/2-1} (\omega^2)^{jk}\psi_{2j+1} \\ \tilde{\psi}_{k+\mathrm{N}/2} &= \sum_{j=0}^{\mathrm{N}/2-1} (\omega^2)^{jk}\psi_{2j} - \omega^k \sum_{j=0}^{\mathrm{N}/2-1} (\omega^2)^{jk}\psi_{2j+1}\end{aligned} \tag{5.11}$$

car $\omega_{k+\mathrm{N}/2} = -\omega_k$.

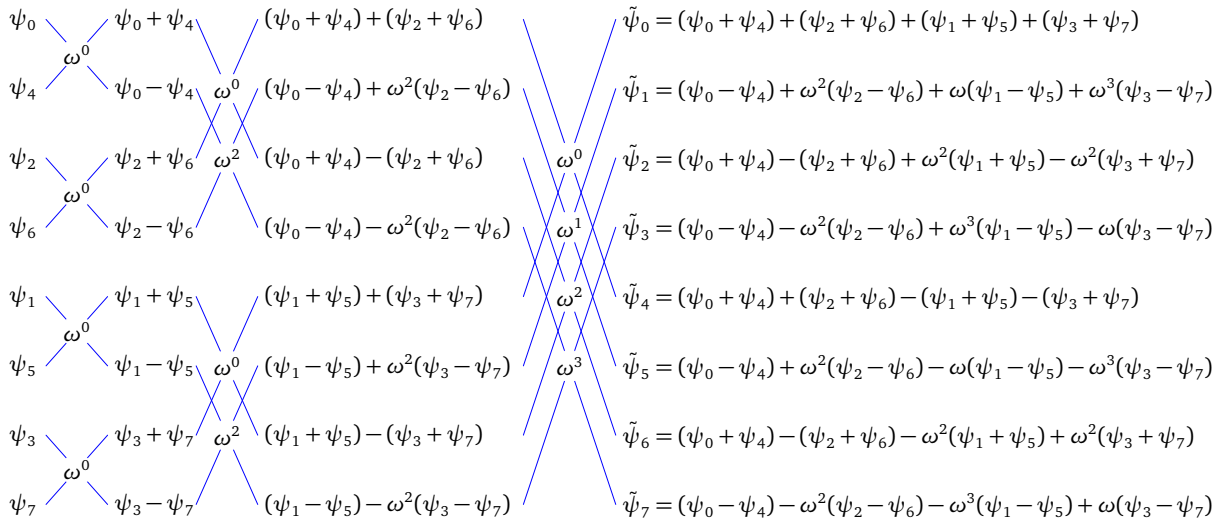


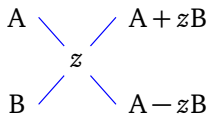
FIGURE 5.1

Chacun des deux termes de (5.11) peut, à son tour, est séparé en deux TdF à N/4 termes, et ainsi de suite jusqu'à ce que le problème soit réduit à une somme de N TdFs à 1 terme chacune. À l'étape 2, les quatre sous-vecteurs de longueur N/4 peuvent être notés $\psi^{(00)}$, $\psi^{(01)}$, $\psi^{(10)}$ et $\psi^{(11)}$. Le sous-vecteur $\psi^{(00)}$ contient les termes pairs de $\psi^{(0)}$, dont les indices originaux sont 0 modulo 4, alors que $\psi^{(01)}$ contient les termes impairs de $\psi^{(0)}$, dont les indices originaux sont 2 modulo 4, et ainsi de suite. À la dernière étape, chaque sous-vecteur ne contient qu'une seule composante et est indexé par une représentation binaire à M bits, par exemple $\psi^{(0011001011)} = \psi^{\mathrm{J}}$ pour $\mathrm{M} = 10$ et $\mathrm{N} = 1024$. Cet élément est indexé par un entier J dont la représentation binaire (0011001011 dans notre exemple) est précisément l'inverse de celle de l'indice j du même élément dans le vecteur original ψ (c'est-à-dire $1101001100 = 844$). Si on dispose les éléments de ψ non pas dans leur ordre original (celui des indices j), mais dans l'ordre des indices binaires inversés J, alors les TdF partielles qui doivent être réalisées à chaque étape sont plus facilement effectuées.

Cet algorithme est illustré dans le cas $\mathrm{N} = 8$ sur la figure 5.1. La dernière colonne contient les 8 valeurs $\tilde{\psi}_k$, dans l'ordre des k croissants. Le passage de la dernière colonne à l'avant-dernière se fait en vertu de l'équation (5.11). Les valeurs de k sont groupées en paires séparées de N/2, c'est-à-dire apparaissent comme ω^k et $-\omega^k$. La somme (5.11) apparaît donc dans le diagramme comme une somme de deux termes avec poids relatif ω^k ($k = 0,1,2,3$), ou comme une différence des deux mêmes termes. Cette double combinaison (somme et différence avec poids relatif ω^k) forme graphiquement ce qu'on appelle un «papillon» (voir fig. 5.2). Les éléments de la troisième colonne de

la figure 5.1 forment deux groupes de TdF partielles, et une autre application de l'équation (5.11) sur ces deux séries nous amène à la deuxième colonne. À cette étape les papillons ne comportent que 4 valeurs de l'exposant, soit $\pm\omega^2 = \pm i$ et $\pm\omega^0 = \pm 1$. Il en résulte 4 TdF partielles. Enfin, une dernière application de l'équation (5.11) nous amène à a première colonne, qui ne comporte que des papillons appliquant les combinaisons $\pm\omega^0$. La première colonne est faite des éléments de la fonction directe dans l'ordre des indices binaires inversés J.

FIGURE 5.2
L'opérateur du «papillon» prend les deux données A et B et les remplace par les combinaisons $A \pm zB$ comme illustré.



Évidemment, l'algorithme de TdFR doit procéder dans l'ordre inverse, c'est-à-dire partir des ψ_j pour aboutir aux $\tilde{\psi}_k$. Ceci se fait en procédant aux étapes suivantes :

1. Placer les ψ_j dans l'ordre des indices binaires inversés. Soit $\mathrm{I}(j)$ l'indice obtenu en inversant l'ordre des bits de j. Comme le carré de l'inversion est l'opérateur identité, on a $\mathrm{I}(\mathrm{I}(j)) = j$ et le changement d'ordre se fait simplement en appliquant les transpositions

$$\psi_j \longleftrightarrow \psi_{\mathrm{I}(j)} \tag{5.12}$$

   L'opération ne nécessite pratiquement pas de stockage supplémentaire et le tableau inversé remplace simplement le tableau original.
2. On effectue ensuite une boucle externe sur les niveaux. Au premier niveau, on effectue tous les «papillons» au sein des paires successives d'éléments du tableau, autrement dit entre les paires d'indices qui ne diffèrent que par le premier bit. Encore une fois le tableau est remplacé par un nouveau tableau sans besoin d'espace supplémentaire (sauf une valeur transitoire).
3. On continue dans cette boucle en passant à la colonne suivante. Les opérations papillons se font maintenant en combinant les indices qui ne diffèrent que par leur deuxième bit. Le tableau de l'itération précédente est remplacé par un nouveau tableau, et ainsi de suite.
4. Quand cette boucle interne est terminée, le tableau original a été remplacé par sa transformée de Fourier.
5. Chaque papillon comporte le même nombre d'opérations arithmétiques, et le nombre de papillons est $\mathrm{N}/2$ à chaque étape de la boucle, qui comporte elle-même $\log_2 \mathrm{N}$ étapes ; donc la complexité algorithmique de la TdFR est $\mathcal{O}(\mathrm{N}\log_2 \mathrm{N})$.

C.2 Cas des dimensions supérieures

La TdF peut être appliquée à une fonction de plus d'une variable. Considérons par exemple le cas de deux dimensions et introduisons les variables discrètes x et y, chacune allant de 0 à $\mathrm{N}-1$, ainsi que les indices réciproques correspondants k_x et k_y. La TdF prend alors la forme suivante :

$$\psi_{x,y} = \frac{1}{\mathrm{N}^2} \sum_{k_x,k_y} \bar{\omega}^{xk_x+yk_y} \tilde{\psi}_{k_x,k_y} \qquad \tilde{\psi}_{k_x,k_y} = \sum_{x,y} \omega^{xk_x+yk_y} \psi_{x,y} \tag{5.13}$$

La façon la plus directe de procéder à la TdFR est de commencer par effectuer N TdFR sur la variable y, une pour chaque valeur de x. On obtient alors un objet intermédiaire Ψ_{x,k_y} qui réside dans l'espace réciproque en y, mais dans l'espace direct en x. Ensuite on procède à N TdFR sur la variable x,

une pour chaque valeur de k_y. Il y a donc 2N TdFR à calculer, chacune de complexité $\mathrm{N}\log_2\mathrm{N}$, ce qui donne une complexité totale $2\mathrm{N}^2\log_2\mathrm{N} = \mathrm{N}^2\log_2\mathrm{N}^2$, où N^2 est le nombre de points échantillonnés en deux dimensions.

C.3 Fonctions réelles

L'algorithme de TdFR peut être appliqué tel quel à des fonctions réelles, mais il entraîne un certain gaspillage d'espace et de temps de calcul, car une fonction réelle discrétisée ψ_j ne comporte que la moitié des degrés de liberté d'une fonction complexe. D'autre part, sa TdF $\tilde{\psi}_k$ est toujours complexe, mais jouit de la symétrie suivante :

$$\tilde{\psi}_k^* = \tilde{\psi}_{\mathrm{N}-k} \tag{5.14}$$

(en particulier, $\tilde{\psi}_0$ et $\tilde{\psi}_{\mathrm{N}/2}$ sont réels).

Deux procédures différentes peuvent être utilisées pour rendre le calcul plus efficace :

1. Calculer deux TdFR simultanément. Souvent les TdF doivent être faites en série. Si on combine deux fonctions réelles ψ' et ψ'' en une seule fonction complexe $\psi = \psi' + i\psi''$, appliquer la TdFR sur ψ nous permet d'obtenir simultanément les TdF des parties réelle et imaginaire, en appliquant la formule simple

$$\tilde{\psi}'_k = \frac{1}{2}\left(\tilde{\psi}_k + \tilde{\psi}^*_{\mathrm{N}-k}\right) \qquad \tilde{\psi}''_k = \frac{1}{2i}\left(\tilde{\psi}_k - \tilde{\psi}^*_{\mathrm{N}-k}\right) \tag{5.15}$$

2. Combiner les parties paire et impaire de la fonction ψ en une seule fonction complexe Ψ de N/2 composantes : $\Psi = \psi^{(0)} + i\psi^{(1)}$. On montre que

$$\tilde{\psi}_k = \frac{1}{2}\left(\tilde{\Psi}_k + \tilde{\Psi}^*_{\mathrm{N}/2-k}\right) - \frac{i}{2}\left(\tilde{\Psi}_k - \tilde{\Psi}^*_{\mathrm{N}/2-k}\right)\mathrm{e}^{2\pi ik/\mathrm{N}} \tag{5.16}$$

Boîte à outils

Les transformées de Fourier rapides sont implantées en Python dans la bibliothèque `scipy.fft`. Voir aussi le tutoriel.
Les principales fonctions de ce module sont

- `fft` : calcule la TdeF directe pour une séquence complexe
- `ifft` : calcule la TdeF inverse pour une séquence complexe
- `rfft` : calcule la TdeF directe pour une séquence réelle
- `irfft` : calcule la TdeF inverse pour une séquence réelle
- `fft2` : calcule la TdeF directe pour une séquence complexe en dimension 2
- `ifft2` : calcule la TdeF inverse pour une séquence complexe en dimension 2

Ces fonctions prennent des tableaux NumPy comme arguments.

Problème 5.1 :
Démontrez les relations (5.15) et (5.16).

CHAPITRE 6

SIMULATION DE PARTICULES II : ÉCOULEMENT D'UN PLASMA

Nous allons dans cette section procéder à une simulation d'un autre genre, qui se rapproche plus de la dynamique moléculaire : celle de l'écoulement d'un plasma en dimension 1. Cette simulation va révéler une instabilité qui se manifeste dans l'interaction de deux faisceaux opposés. [1]

Un plasma est un gaz d'électrons en coexistence avec des ions, de sorte que le système est neutre au total. Par contre, comme les électrons sont beaucoup plus légers que les ions, les temps caractéristiques associés au mouvement des deux composantes du plasma sont très différents. Nous n'allons étudier ici que le mouvement des électrons, et supposer que le rôle des ions n'est que de neutraliser le système dans son ensemble.

A Description de la méthode

Considérons donc un ensemble de N électrons de charge $e < 0$, positions r_μ et de vitesses v_μ en dimension 1 ($\mu = 1, 2, \ldots, \mathrm{N}$). Si nous procédions à une simulation de dynamique moléculaire classique, nous devrions calculer, à chaque instant de la simulation, la force électrique agissant sur chaque particule, ce qui est un processus de complexité $\mathcal{O}(\mathrm{N}^2)$. Nous allons réduire la complexité du calcul à $\mathcal{O}(\mathrm{N})$ à l'aide de l'astuce suivante :

1. Les forces seront représentées par un champ électrique $\mathrm{E}(x)$, dérivant du potentiel électrique $\phi(x)$ et de la densité de charge $e\rho(x)$ ($\rho(x)$ étant la densité volumique de particules). Ces différents champs seront représentés sur une grille de M points, où $\mathrm{M} \ll \mathrm{N}$.
2. Les positions r_μ des particules seront utilisées pour calculer la densité ρ.
3. Le potentiel $\phi(x)$ sera déterminé par la solution de l'équation de Poisson : $\mathrm{d}^2\phi/\mathrm{d}x^2 = -e\rho(x)/\varepsilon_0$, à l'aide de transformées de Fourier rapides.
4. Le champ électrique $\mathrm{E}(x)$ sera dérivé du potentiel électrique : $\mathrm{E}(x) = -\mathrm{d}\phi/\mathrm{d}x$.

C'est l'utilisation de transformées de Fourier rapides (TdFR) qui rend cette méthode plus rapide : sa complexité sera la moindre de $\mathcal{O}(\mathrm{N})$ et de $\mathcal{O}(\mathrm{M}\log\mathrm{M})$.

1. Cette section est inspirée des notes de R. Fitzpatrick, de l'Université du Texas à Austin : http://farside.ph.utexas.edu/teaching/329/329.html.

A.1 Mise à l'échelle du problème

Il est important lors d'une simulation numérique de rapporter les différentes quantités physiques en fonction de quantités normalisées dont les grandeurs caractéristiques sont typiques au problème. En définissant un potentiel normalisé Φ tel que

$$\phi \coloneqq \frac{e\bar{\rho}}{\varepsilon_0}\Phi \tag{6.1}$$

où $\bar{\rho}$ est la densité moyenne d'électrons, l'équation du mouvement de la particule μ est

$$\dot{v}_\mu = \frac{e}{m}\mathrm{E}(r_\mu) = -\frac{e^2\bar{\rho}}{m\varepsilon_0}\Phi'(r_\mu) \coloneqq -\omega_p^2\Phi'(r_\mu) \tag{6.2}$$

où ω_p est la *fréquence plasma*, soit la fréquence à laquelle un plasma uniforme qui serait déplacé latéralement par rapport à son fond neutralisant se mettrait à osciller. Parallèlement, l'équation de Poisson devient alors

$$\Phi'' = \frac{\varepsilon_0}{e\bar{\rho}}\phi'' = -\frac{\varepsilon_0}{e\bar{\rho}}e\frac{\rho}{\varepsilon_0} = -\frac{\rho}{\bar{\rho}} \coloneqq -n \tag{6.3}$$

où n est la densité normalisée d'électrons. Les équations sont donc particulièrement simples si on les exprime en fonction de $n(x)$ et du potentiel normalisé $\Phi(x)$. La fréquence plasma inverse ω_p^{-1} servira ici d'unité naturelle de temps. Autrement dit, on définit un temps sans unités $\tau = \omega_p t$, en fonction duquel on définit une vitesse renormalisée V_μ, avec les équation du mouvement suivantes :

$$\mathrm{V}_\mu \coloneqq \frac{\mathrm{d}r_\mu}{\mathrm{d}\tau} = \omega_p^{-1}v_\mu \qquad\qquad \frac{\mathrm{dV}_\mu}{\mathrm{d}\tau} = -\Phi'(r_\mu) \tag{6.4}$$

A.2 Algorithme

Revoyons donc en détail chaque étape de la méthode :

1. La densité normalisée n, le potentiel électrique normalisé Φ et le champ électrique normalisé $\mathrm{F} = -\Phi'$ seront définis sur une grille uniforme périodique de M points et de pas $a = \mathrm{L}/\mathrm{M}$. La présence d'un électron à la position r modifie la densité n uniquement sur les points x_j et x_{j+1} situés de part et d'autre de r.

   $$n_j \to n_j + \frac{\mathrm{L}}{\mathrm{N}a^2}(x_{j+1} - r) \qquad n_{j+1} \to n_{j+1} + \frac{\mathrm{L}}{\mathrm{N}a^2}(r - x_j) \qquad x_j < r < x_{j+1} \tag{6.5}$$

   Autrement dit, on répartit le poids de la particule sur les deux points voisins, en proportion de la distance qui sépare la particule de l'autre point. Pour se convaincre que le préfacteur est correct, on constate que l'ajout d'une particule a l'effet suivant sur la somme des densités aux différents points :

   $$\sum_j n_j \to \sum_j n_j + \frac{\mathrm{L}}{\mathrm{N}a} \tag{6.6}$$

   et donc que l'effet de toutes les particules est $\sum_j n_j = \mathrm{L}/a$. L'intégrale de la densité serait alors

   $$\int_0^{\mathrm{L}} n(x)dx \longrightarrow \sum_j a\,n_j = \mathrm{L} \tag{6.7}$$

   ce qui est bien le résultat attendu pour une densité dont la moyenne est égale à 1.

2. Sur une grille de pas a, l'équation de Poisson prend la forme suivante :

$$\frac{\Phi_{j+1}-2\Phi_j+\Phi_{j-1}}{a^2}=-n_j \tag{6.8}$$

où on reconnaît la forme discrète de la deuxième dérivée. En fonction des transformées de Fourier discrètes $\tilde{\Phi}(q)$ et $\tilde{n}(q)$, cela se traduit par

$$(2\cos(q)-2)\tilde{\Phi}(q)=-a^2\tilde{n}(q) \quad \text{et donc} \quad \tilde{\Phi}(q)=\gamma_q\tilde{n}(q) \qquad \gamma_q \coloneqq \frac{a^2}{2-2\cos(q)} \tag{6.9}$$

Notons que sur une grille de M sites, $q \in 2\pi\mathbb{Z}/\mathrm{M}$ et si q est petit, alors $2-2\cos(q)\approx q^2$. En pratique, on procède à une TdFR de n et on calcule $\tilde{\Phi}$ en multipliant par γ_q comme ci-dessus ($q\neq 0$). On met également à zéro la composante $q=0$ de $\tilde{n}$, ce qui entraîne que la charge totale est nulle. Cette étape impose la condition de neutralité et est le seul endroit du calcul où la présence des ions positifs est prise en compte. Enfin, on procède à une TdFR inverse pour retrouver $\Phi_j=\Phi(x_j)$.

3. Le champ électrique normalisé est l'opposé de la dérivée du potentiel électrique normalisé :

$$\mathrm{F}_j=-\frac{\Phi_{j+1}-\Phi_{j-1}}{2a} \tag{6.10}$$

4. La méthode de Verlet sera employée pour évoluer dans le temps les positions et les vitesses, selon les équations suivantes :

$$r_\mu(t+h)=r_\mu(t)+\mathrm{V}_\mu h+\frac{h^2 f_\mu}{2} \tag{6.11}$$

$$\mathrm{V}_\mu(t+h)=\frac{1}{(1+\gamma h/2)}\left\{\mathrm{V}_\mu(t)(1-\gamma h/2)+\frac{f_\mu(t)+f_\mu(t+h)}{2}\right\} \tag{6.12}$$

où γ est un amortissement dû à des causes diverses (rayonnement, etc.). La force f_i à un instant donné sera calculée en interpolant le champ électrique défini sur la grille :

$$f_\mu=\frac{r_\mu-x_j}{a}\mathrm{F}_j+\frac{x_{j+1}-r_\mu}{a}\mathrm{F}_{j+1} \tag{6.13}$$

où x_j est le point no j de la grille périodique (il est implicite que $j+\mathrm{M}\to j$, c'est-à-dire que j est défini modulo M) et F_j est la valeur du champ électrique à ce point. On retourne ensuite à l'étape 1 (calcul mis à jour de la densité), jusqu'à ce que le temps de simulation soit écoulé.

Les conditions initiales seront les suivantes :

1. Une distribution initiale aléatoire des positions r_i dans l'intervalle $[0,\mathrm{L}]$.
2. Une distribution gaussienne des vitesses v_i autour de deux valeurs $\pm v_0$, ce qui représente deux faisceaux contraires de particules. L'écart-type de cette distribution des vitesses est $v_\mathrm{T}=\sqrt{\mathrm{T}/m}$, où T est la température absolue. Cette distribution est donc maxwellienne.

Le problème est véritablement unidimensionnel, ce qui signifie que, d'une certaine manière, le faisceau est confiné dans un canal étroit et que le mouvement des charges dans la direction transversale est soit interdit, soit trop rapide pour être pris en compte. Outre la fréquence plasma ω_p, une échelle caractéristique du plasma est la longueur de Debye $\lambda_\mathrm{D}=v_\mathrm{T}/\omega_p$, au-delà de laquelle les particules

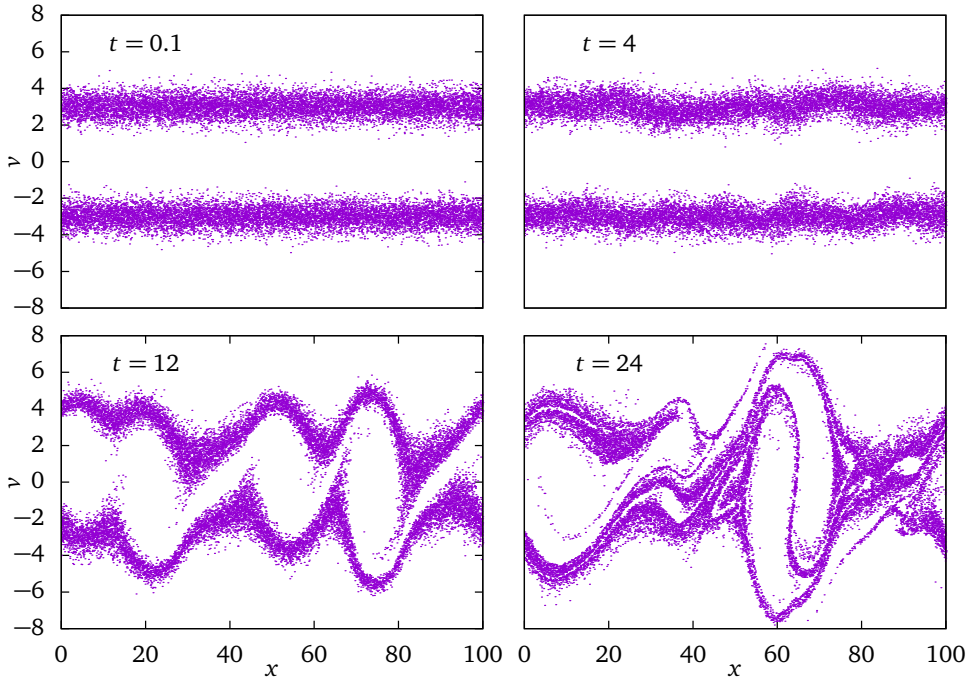


FIGURE 6.1
Distribution des positions et des vitesses des électrons du plasma dans l'espace des phases, pour 4 valeurs du temps t. L'instabilité est visible dès $t = 4$. Les paramètres utilisés sont $v_0 = 1$, $v_T = 0.5$, $h = 0.001$, $\gamma = 0$, L = 100, M = 1024 et N = 20 000.

du plasma démontrent un comportement collectif au lieu d'un comportement strictement individuel.

Le phénomène observé lors de cette simulation est illustré à la figure 6.1 : une instabilité du mouvement apparaît progressivement. Le mouvement uniforme des particules du plasma se transforme en une combinaison de mouvement uniforme et d'oscillation, menant à une inhomogénéité de la densité.

B Plasma en deux dimensions en présence d'un champ magnétique

Étendons la méthode ci-dessus pour traiter le problème d'un plasma en deux dimensions, en présence d'un champ magnétique perpendiculaire au plan : $\mathbf{B} = \mathrm{B}\mathbf{z}$. Le domaine de variation des particules sera périodique dans les deux directions, soit un tore de dimensions $\mathrm{L} \times \mathrm{L}$. L'équation du mouvement des particules est la suivante :

$$\dot{\mathbf{v}}_\mu = \frac{e}{m}\mathbf{E}(r_\mu) + \frac{e\mathrm{B}}{m}\mathbf{v}_\mu \wedge \mathbf{z} = \omega_p^2 \nabla\Phi(r_\mu) + \omega_c \mathbf{v}_\mu \wedge \mathbf{z} \tag{6.14}$$

où ω_c est la fréquence cyclotron et Φ est le potentiel normalisé, comme à la section précédente. En fonction du temps sans unités $\tau = \omega_p t$, on a plutôt les équations suivantes :

$$\frac{\mathrm{d}\mathbf{r}_\mu}{\mathrm{d}\tau} = \omega_p^{-1}\mathbf{v}_\mu \coloneqq \mathbf{V}_\mu \qquad \frac{\mathrm{d}\mathbf{V}_\mu}{\mathrm{d}\tau} = \nabla\Phi(r_\mu) + \frac{\omega_c}{\omega_p}\mathbf{V}_\mu \wedge \mathbf{z} \tag{6.15}$$

où

$$\frac{\omega_c}{\omega_p} = \mathrm{B}\sqrt{\frac{\varepsilon_0}{m\bar{\rho}}} \coloneqq \tilde{\mathrm{B}} \tag{6.16}$$

est un champ magnétique renormalisé sans unités. Les équations de Verlet sont alors

$$\mathbf{r}_\mu(t+h) = \mathbf{r}_\mu(t) + \mathbf{V}_\mu h + \frac{h^2\mathbf{f}_\mu}{2} \tag{6.17}$$

$$\mathbf{V}_\mu(t+h) = \mathbf{V}_\mu(t) - \frac{\gamma h}{2}\big(\mathbf{V}_\mu(t) + \mathbf{V}_\mu(t+h)\big) + \tag{6.18}$$

$$\frac{\tilde{\mathrm{B}}h}{2}\big(\mathbf{V}_\mu(t) + \mathbf{V}_\mu(t+h)\big) \wedge \mathbf{z} + h\frac{\mathbf{f}_\mu(t) + \mathbf{f}_\mu(t+h)}{2} \tag{6.19}$$

La deuxième équation est implicite pour $\mathbf{V}_\mu(t+h)$: il faut résoudre un système linéaire simple pour obtenir une expression explicite de $\mathbf{V}_\mu(t+h)$.

Le temps caractéristique en présence d'un champ magnétique est

$$\mathrm{T}_c = \frac{2\pi}{\omega_c} = \frac{2\pi m}{e\mathrm{B}} \tag{6.20}$$

En fonction du temps sans unités τ, ce temps caractéristique devient

$$\omega_p \mathrm{T}_c = \frac{2\pi\omega_p}{\omega_c} = \frac{2\pi}{\tilde{\mathrm{B}}} \tag{6.21}$$

En deux dimensions, les quantités définies sur la grille portent deux indices (i, j). l'analogue de l'éq. 6.5 est

$$\begin{aligned} n_{i,j} &\to n_{i,j} + \frac{\mathrm{L}}{\mathrm{N}a^4}((j+1)a-x)((i+1)a-y) \\ n_{i+1,j} &\to n_{i+1,j} + \frac{\mathrm{L}}{\mathrm{N}a^4}((j+1)a-x)(y-ia) \\ n_{i,j+1} &\to n_{i,j+1} + \frac{\mathrm{L}}{\mathrm{N}a^4}(x-ja)((i+1)a-y) \\ n_{i+1,j+1} &\to n_{i+1,j+1} + \frac{\mathrm{L}}{\mathrm{N}a^4}(x-ja)(y-ia) \end{aligned} \tag{6.22}$$

où la position (x, y) de la particule est située dans le carré délimité par (i, j) et $(i+1, j+1)$. En plus, afin de calculer la force $\mathbf{f}(\mathbf{r})$ à une position intermédiaire $\mathbf{r}$, on doit procéder à une interpolation des quatre valeurs sur la grille qui entourent la position $\mathbf{r}$. La formule à cette fin est la suivante :

$$\begin{aligned} \mathbf{f} = \frac{1}{a^2}\Big\{ &\mathbf{E}_{i,j}\Big[((j+1)a-x)((i+1)a-y))\Big] - \mathbf{E}_{i,j+1}\Big[(ja-x)((i+1)a-y))\Big] \\ &- \mathbf{E}_{i+1,j}\Big[((j+1)a-x)(ia-y))\Big] + \mathbf{E}_{i+1,j+1}\Big[(ja-x)(ia-y))\Big]\Big\} \end{aligned} \tag{6.23}$$

Un autre changement notable par rapport au problème unidimensionnel est la solution de l'équation de Poisson en fonction des transformées de Fourier, qui devient

$$(2\cos q_x + 2\cos q_y - 4)\tilde{\Phi}(q) = -a^2\tilde{n}(q) \implies \tilde{\Phi}(q) = \gamma_q \tilde{n}(q) \tag{6.24}$$

où $\gamma_q \coloneqq a^2/(4-2\cos q_x - 2\cos q_y)$.

Référentiel tournant Si on considère ce problème du point de vue d'un référentiel tournant à une fréquence Ω, les équations du mouvement sont modifiées par l'ajout de l'accélération de Coriolis $-2\Omega\mathbf{z}\wedge\mathbf{v}$ et de l'accélération centrifuge $\Omega^2\mathbf{r}$:

$$\dot{\mathbf{v}}_\mu = \omega_p^2 \nabla\Phi(r_\mu) + \omega_c \mathbf{v}_\mu \wedge \mathbf{z} - 2\Omega\mathbf{z}\wedge\mathbf{v}_\mu + \Omega^2\mathbf{r}_\mu \tag{6.25}$$

Si on choisit précisément $\Omega = -\omega_c/2$, les deux termes proportionnels à la vitesse s'annulent mutuellement et on reste avec

$$\dot{\mathbf{v}}_\mu = \omega_p^2 \nabla\Phi(r_\mu) + \frac{1}{4}\omega_c^2 \mathbf{r}_\mu \tag{6.26}$$

Dans ce référentiel, le mouvement du plasma est donc le même qu'en l'absence de champ magnétique, sauf pour la présence d'une force centrifuge qui croît comme le carré du champ. Par contre, une telle description est en contradiction avec les conditions aux limites périodiques qu'on serait tenté d'utiliser pour éliminer les effets de bord.

CHAPITRE 7

OPÉRATIONS MATRICIELLES

L'algèbre linéaire est au coeur des méthodes numériques en science, même dans l'étude des phénomènes non linéaires. Si on pouvait recenser les cycles de calcul sur tous les ordinateurs de la planète, il est probable qu'une fraction proche de l'unité serait dédiée à des calculs impliquant des matrices. Il est donc important de survoler les méthodes utilisées pour procéder aux deux opérations les plus importantes de l'algèbre linéaire : la résolution des systèmes d'équations linéaires, et le calcul des valeurs et vecteurs propres.

Mise en garde : le sujet des méthodes numériques en algèbre linéaire est vaste et complexe. Il existe toute une littérature sur le sujet et des livres entiers qui tentent de le résumer. Ce court chapitre ne constitue qu'un hors-d'oeuvre, pour ainsi dire.

Quelques mots sur la notation

On utilisera la notation de Dirac pour les vecteurs abstraits, par exemple $|x\rangle$. Par contre, le symbole x tout seul désignera l'ensemble des composantes de ce vecteur dans une base particulière, et les composantes elles-mêmes seront notées x_i. Le vecteur conjugué sera noté $\langle x|$, mais le vecteur-rangée associé dans une base particulière sera noté x^{T} (ou $x^\dagger$ s'il s'agit de composantes complexes). Ainsi, le produit scalaire de deux vecteurs sera noté soit $y^{\mathrm{T}}x$ (ou $y^\dagger x$ pour les vecteurs complexes) dans une base particulière, ou encore $\langle y|x\rangle$, ce qui ne change rien puisque ce produit est indépendant de la base utilisée. Pour une matrice A dont les composantes sont A_{ij}, on pourra noter $\mathrm{A}_{i\bullet}$ l'ensemble des composantes formant la rangée i et $\mathrm{A}_{\bullet j}$ l'ensemble des composantes formant la colonne j. La grandeur d'un vecteur $|x\rangle$ sera notée $|x|$. Une matrice A appliquée sur un vecteur x sera notée $\mathrm{A}x$. L'opérateur abstrait (c'est-à-dire indépendant de la base) associé à la matrice A sera noté $\hat{\mathrm{A}}$.

A Systèmes d'équations linéaires

A.1 Système général et types de matrices

Le problème de base de l'algèbre linéaire est la solution d'un système d'équations linéaires :

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \quad \text{ou encore} \quad \mathrm{A}x = b \tag{7.1}$$

Nous avons supposé que le nombre d'inconnues est égal au nombre d'équations, et donc que la matrice est carrée. Si ce n'est pas le cas, le système est soit surdéterminé (aucune solution exacte n'existe), ou sous-déterminé (certaines variables doivent être déterminées autrement).

La méthode indiquée pour résoudre un tel système dépend de la configuration de la matrice A. Par exemple, on distingue les types suivantes :

matrice pleine La matrice ne comporte aucune catégorie d'éléments ou disposition particulière : ses éléments sont en général non nuls.

matrice triangulaire supérieure Les éléments situés au-dessous de la diagonale sont nuls.

matrice triangulaire inférieure Les éléments situés au-dessus de la diagonale sont nuls.

matrice tridiagonale Les éléments non nuls sont situés sur la diagonale et sur les deux diagonales voisines (la matrice possède donc trois diagonales).

matrice multidiagonale Les éléments non nuls sont situés sur la diagonale ainsi que sur $\ell - 1$ diagonales de chaque côté de la diagonale. On dit alors que la matrice multidiagonale est de demi-largeur ℓ.

matrice creuse La vaste majorité des éléments sont nuls, mais les éléments non nuls sont dispersés un peu partout. Une telle matrice doit être stockée de telle sorte que seuls les éléments non nuls sont repérés (plusieurs schémas de stockage sont possibles).

A.2 Système triangulaire

Le système (7.1) est très simple à résoudre si la matrice A est triangulaire. Par exemple, si elle est *triangulaire supérieure*, le système se résout trivialement à partir du bas, par *rétro substitution* (ou *substitution vers l'arrière*) :

$$\begin{aligned} x_n &= \frac{b_n}{a_{nn}} \\ x_{n-1} &= \frac{1}{a_{n-1,n-1}} \left(b_{n-1} - a_{n-1,n} x_n \right) \\ x_{n-2} &= \frac{1}{a_{n-2,n-2}} \left(b_{n-2} - a_{n-2,n-1} x_{n-1} - a_{n-2,n} x_n \right) \\ &\text{etc}\dots \end{aligned} \tag{7.2}$$

La solution n'existe manifestement que si tous les éléments diagonaux a_{ii} sont non nuls.

Si la matrice est *triangulaire inférieure*, la solution est également simple, par substitution vers l'avant (c'est-à-dire en commençant par x_1) :

$$\begin{aligned} x_1 &= \frac{b_1}{a_{11}} \\ x_2 &= \frac{1}{a_{2,2}}\left(b_2 - a_{2,1}x_1\right) \\ x_3 &= \frac{1}{a_{3,3}}\left(b_3 - a_{3,2}x_2 - a_{3,1}x_1\right) \\ &\text{etc}\ldots \end{aligned} \tag{7.3}$$

A.3 Élimination gaussienne

Supposons maintenant que nous ayons affaire à une matrice pleine. L'algorithme de base pour résoudre le système (7.1) est l'*élimination gaussienne*. Il consiste à soustraire de chaque équation (rangée i) une combinaison des équations précédentes (rangées $< i$) de manière à ramener le système à une forme triangulaire supérieure. Une fois ceci accompli, la solution est immédiate, comme démontré ci-dessus.

Illustrons l'algorithme d'élimination gaussienne à l'aide d'un exemple. Considérons le système suivant :

$$\begin{pmatrix} 4 & 8 & 12 \\ 3 & 8 & 13 \\ 2 & 9 & 18 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \\ 11 \end{pmatrix} \tag{7.4}$$

La première étape consiste à diviser la première rangée du système par 4 afin de ramener la première valeur diagonale à l'unité :

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 8 & 13 \\ 2 & 9 & 18 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \\ 11 \end{pmatrix} \tag{7.5}$$

Ensuite, on soustrait les multiples appropriés de la première rangée des autres rangées, afin d'annuler le reste de la première colonne :

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 2 & 4 \\ 0 & 5 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 9 \end{pmatrix} \tag{7.6}$$

Ensuite, on recommence avec le deuxième élément de la diagonale : on divise par 2 cette fois et il reste

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 5 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 9 \end{pmatrix} \tag{7.7}$$

On soustrait de la dernière équation 5 fois la seconde, pour trouver enfin une forme triangulaire supérieure :

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 4 \end{pmatrix} \tag{7.8}$$

Ce système se résout alors comme expliqué plus haut. Dans ce cas-ci la solution est $(x_1, x_2, x_3) = (1, -3, 2)$.

On montre que le nombre d'opérations arithmétiques nécessaires à cet algorithme est

$$\frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} \tag{7.9}$$

ce qui se comporte comme $\frac{1}{3}n^3$ quand n est grand. Autrement dit, inverser un système 2 fois plus grand prend 8 fois plus de temps, quand n est grand.

Pivotage
L'algorithme de Gauss simple illustré ci-dessus ne fonctionne que si les éléments diagonaux sont non nuls à chaque étape. Si ce n'est pas le cas, on doit avoir recours au *pivotage*, c'est-à-dire à une permutation des rangées de manière à repousser au bas de la matrice la rangée qui pose problème. Ceci produira toujours une solution, pourvu que la matrice soit non singulière.

A.4 Décomposition LU

La décomposition LU est la représentation d'une matrice A comme produit d'une matrice triangulaire inférieure L par une matrice triangulaire supérieure U, modulo une possible permutation des rangées :

$$\mathrm{A} = \mathrm{PLU} \quad \text{où} \quad \mathrm{L} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{pmatrix} \qquad \mathrm{U} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{pmatrix} \tag{7.10}$$

et où P est une matrice de permutation, c'est-à-dire une matrice qui effectue une permutation p des rangées de la matrice sur laquelle elle s'applique, ou les composantes d'un vecteur sur lequel elle s'applique. Cette matrice est obtenue de la matrice identité en permutant ses colonnes à l'aide de la permutation p. On peut supposer sans perte de généralité que les éléments diagonaux de la matrice L sont l'unité.

La décomposition LU s'effectue d'une manière analogue à l'élimination de Gauss, par l'*algorithme de Crout*, que nous n'expliquerons pas ici (voir les références standards, comme [PTVF07]). Notons cependant que cet algorithme a en général recours au pivotage des rangées, et qu'il est toujours stable numériquement (en supposant bien sûr que la matrice A est non singulière). La complexité algorithmique de la décomposition LU est la même que celle de la multiplication de deux matrices, soit $\mathcal{O}(n^3)$.

Une fois la décomposition effectuée, la solution du système linéaire (7.1) s'effectue en deux étapes : on résout premièrement le système triangulaire $\mathrm{PL}y = b$ (ou $\mathrm{L}y = \mathrm{P}^{-1}b$) pour y (par substitution

vers l'avant), et ensuite le système triangulaire $\mathrm{U}\,x = y$ (par rétro-substitution). De plus, la décomposition a de multiples avantages collatéraux :

1. Une fois effectuée pour une matrice A, elle permet de résoudre le système avec plusieurs vecteurs b différents, même si les vecteurs b ne sont pas connus au moment d'effectuer la décomposition.
2. Elle permet de calculer simplement l'inverse de la matrice, en résolvant le système avec n vecteurs b tels que $b_i = \delta_{ij}$ $(j = 1, 2, \ldots, n)$.
3. Elle permet de calculer facilement le déterminant de A, comme le produit des éléments diagonaux de la matrice U.

La décomposition LU peut être effectuée à l'aide de la fonction `dgetrf_` de la bibliothèque Lapack. Cette dernière est à la base de la fonction `linalg.lu_factor()` du module `scipy.linalg` de Python.

A.5 Système tridiagonal

Supposons maintenant que la matrice A est tridiagonale. La résolution du système (7.1) ou la décomposition LU est alors beaucoup simplifiée, et s'effectue par un algorithme de complexité $\mathcal{O}(n)$. Expliquons : désignons par a_i, b_i et c_i les coefficients apparaissant respectivement sur la diagonale inférieure, la diagonale principale et la diagonale supérieure. le système peut alors s'écrire comme suit :

$$\begin{aligned} b_1 x_1 + c_1 x_2 &= r_1 \\ a_2 x_1 + b_2 x_2 + c_2 x_3 &= r_2 \\ a_3 x_2 + b_3 x_3 + c_3 x_4 &= r_3 \\ &\ldots\ldots \\ a_{n-1} x_{n-2} + b_{n-1} x_{n-1} + c_{n-1} x_n &= r_{n-1} \\ a_n x_{n-1} + b_n x_n &= r_n \end{aligned} \tag{7.11}$$

Notons tout de suite que la première et la dernière équation sont particulières, car les membres de gauche n'ont que deux termes. Éliminons x_1 de la première équation :

$$x_1 = \frac{r_1 - c_1 x_2}{b_1} \tag{7.12}$$

Substituons maintenant dans la deuxième équation :

$$\left(b_2 - \frac{c_1 a_2}{b_1} \right) x_2 + c_2 x_3 = r_2 - \frac{a_2 r_1}{b_1} \tag{7.13}$$

Cette équation a maintenant la même forme que la première, si on effectue les substitutions

$$b_2 \leftarrow b_2 - \frac{c_1 a_2}{b_1} \qquad r_2 \leftarrow r_2 - \frac{a_2 r_1}{b_1} \tag{7.14}$$

et on peut reprendre le processus avec la troisième équation, etc. Bref, on peut procéder à la substitution récurrente suivante, dans l'ordre des i croissants :

$$b_i \leftarrow b_i - \frac{c_{i-1} a_i}{b_{i-1}} \qquad r_i \leftarrow r_i - \frac{a_i r_{i-1}}{b_{i-1}} \qquad i = 2, 3, \ldots, n \tag{7.15}$$

Les nouveaux coefficients b_i et r_i ainsi construits, on demeure avec le système d'équations suivant :

$$x_i = \frac{r_i - c_i x_{i+1}}{b_i} \qquad i = 1, 2, \ldots, n-1 \tag{7.16}$$

alors que la dernière équation devient simplement $x_n = r_n / b_n$. Il suffit alors d'appliquer ces équations à partir de $i = n$ jusqu'à $i = 1$ pour déterminer toutes les inconnues x_i (rétrosubstitution). Nous avons supposé ici qu'aucun des b_i ne s'annule.

La complexité algorithmique de cette méthode est manifestement $\mathcal{O}(n)$. Il existe aussi des routines spéciales pour la solution de systèmes multidiagonaux qui sont avantageux si le nombre de diagonales $2\ell + 1$ est beaucoup plus petite que l'ordre de la matrice.

A.6 Décomposition QR et procédure de Gram-Schmidt

Toute matrice carrée A peut être exprimée sous la forme du produit d'une matrice orthogonale Q (ou unitaire, dans le cas complexe) par une matrice triangulaire supérieure R : A = QR. Si la matrice A est non singulière, alors cette décomposition est unique si on demande que les éléments diagonaux de R soient positifs. Cette décomposition est à la base de l'algorithme QR qui permet de calculer les valeurs et vecteurs propres d'une matrice (voir plus bas).

Procédure de Gram-Schmidt Afin de démontrer l'existence de la décomposition QR, nous allons utiliser la procédure d'orthogonalisation de Gram-Schmidt. Considérons donc les différentes colonnes $\mathrm{A}_{\bullet i}$ de A et désignons les vecteurs correspondants par $|a_i\rangle$ ($i = 1, \ldots, \mathrm{N}$). Ils ne sont pas orthogonaux en général. Par contre, on peut construire un ensemble de vecteurs orthogonaux $|e_i\rangle$ à partir des $|a_i\rangle$ en suivant la *procédure de Gram-Schmidt* :

$$\begin{aligned}
|u_1\rangle &= |a_1\rangle \\
|u_2\rangle &= |a_2\rangle - \frac{\langle u_1|a_2\rangle}{\langle u_1|u_1\rangle}|u_1\rangle \\
|u_3\rangle &= |a_3\rangle - \frac{\langle u_1|a_3\rangle}{\langle u_1|u_1\rangle}|u_1\rangle - \frac{\langle u_2|a_3\rangle}{\langle u_2|u_2\rangle}|u_2\rangle \\
&\vdots \\
|u_k\rangle &= |a_k\rangle - \sum_{j=1}^{k-1} \frac{\langle u_j|a_k\rangle}{\langle u_j|u_j\rangle}|u_j\rangle
\end{aligned} \tag{7.17}$$

Les vecteurs orthogonaux $|u_i\rangle$ sont donc construits en soustrayant de $|a_i\rangle$ ses projections sur les $|u_i\rangle$ trouvés précédemment. On peut ensuite définir les vecteurs normalisés et orthonormés

$$|e_i\rangle \coloneqq \frac{|u_i\rangle}{|u_i|} \qquad |u_i| \coloneqq \sqrt{\langle u_i|u_i\rangle} \tag{7.18}$$

En exprimant les vecteurs $|a_j\rangle$ en fonction des $|e_j\rangle$, on trouve évidemment

$$\begin{aligned}
|a_1\rangle &= |e_1\rangle\langle e_1|a_1\rangle \\
|a_2\rangle &= |e_1\rangle\langle e_1|a_2\rangle + |e_2\rangle\langle e_2|a_2\rangle \\
|a_3\rangle &= |e_1\rangle\langle e_1|a_3\rangle + |e_2\rangle\langle e_2|a_3\rangle + |e_3\rangle\langle e_3|a_3\rangle
\end{aligned}$$

$$\vdots$$
$$|a_k\rangle = \sum_{j=1}^{k} |e_j\rangle\langle e_j|a_k\rangle \tag{7.19}$$

Or, si on définit la matrice Q dont les colonnes sont les vecteur orthonormés $|e_j\rangle$, cette relation n'est autre que A = QR, où la matrice R est définie par

$$\mathrm{R} = \begin{pmatrix} \langle e_1|a_1\rangle & \langle e_1|a_2\rangle & \langle e_1|a_3\rangle & \dots \\ 0 & \langle e_2|a_2\rangle & \langle e_2|a_3\rangle & \dots \\ 0 & 0 & \langle e_3|a_3\rangle & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{7.20}$$

La procédure de Gram-Schmidt est cependant numériquement instable si elle est programmée de manière simple et directe. Par contre, elle peut-être considérablement améliorée en procédant de la manière suivante :

$$\begin{aligned}
|u_1\rangle &= |a_1\rangle \\
|u_2\rangle &= |a_2\rangle - \frac{\langle u_1|a_2\rangle}{\langle u_1|u_1\rangle}|u_1\rangle \\
|u_3^{(1)}\rangle &= |a_3\rangle - \frac{\langle u_1|a_3\rangle}{\langle u_1|u_1\rangle}|u_1\rangle \\
|u_3\rangle &= |u_3^{(1)}\rangle - \frac{\langle u_2|u_3^{(1)}\rangle}{\langle u_2|u_2\rangle}|u_2\rangle \\
|u_4^{(1)}\rangle &= |a_4\rangle - \frac{\langle u_1|a_4\rangle}{\langle u_1|u_1\rangle}|u_1\rangle \\
|u_4^{(2)}\rangle &= |u_4^{(1)}\rangle - \frac{\langle u_2|u_4^{(1)}\rangle}{\langle u_2|u_2\rangle}|u_2\rangle \\
|u_4\rangle &= |u_4^{(2)}\rangle - \frac{\langle u_3|u_4^{(2)}\rangle}{\langle u_3|u_3\rangle}|u_3\rangle \\
&\vdots
\end{aligned} \tag{7.21}$$

Autrement dit, chaque nouveau vecteur $|u_k\rangle$ est construit progressivement en orthogonalisant par rapport à $|u_j\rangle$ et ensuite en orthogonalisant le résultat par rapport à $|u_{j+1}\rangle$, etc jusqu'à $j = k-1$. Le résultat est bien sûr le même qu'en appliquant la formule (7.17), mais l'erreur numérique est moindre. La procédure (7.21) porte le nom de *procédure de Gram-Schmidt modifiée*.

La décomposition QR peut être utilisée afin de résoudre un système linéaire ou d'inverser une matrice, comme la décomposition LU, en utilisant le fait que l'inverse de Q est sa transposée et que l'inverse de R se calcule facilement du fait de sa forme triangulaire. Par contre, la décomposition LU est plus économique pour cette tâche. La véritable utilité de la décomposition QR est le calcul des valeurs et vecteurs propres.

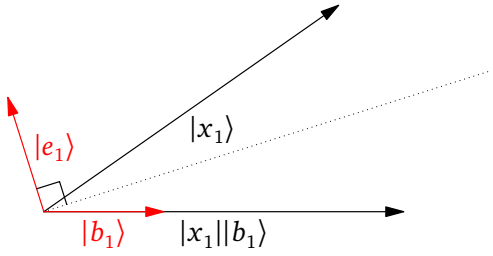
A.7 Procédure de Householder

En pratique, la décomposition QR est souvent exécutée à l'aide de la *procédure de Householder* et non via la procédure de Gram-Schmidt. Expliquons. Une *transformation de Householder* $\mathrm{P}(e)$ est une réflexion par rapport à un plan défini par un vecteur unitaire $|e\rangle$ qui lui est perpendiculaire :

$$\mathrm{P}(e) = \mathbb{I} - 2|e\rangle\langle e| \tag{7.22}$$

Notez que $|e\rangle\langle e|$ est un opérateur qui projette tout vecteur sur lequel il est appliquée sur $|e\rangle$. La matrice correspondante est évidemment orthogonale, car elle préserve la longueur de tout vecteur. Par un choix judicieux du vecteur $|e\rangle$, on peut éliminer les élément situés sous un élément donné d'une matrice, comme nous allons maintenant le montrer.

FIGURE 7.1
Transformation de Householder



Soit $|x_1\rangle$ le vecteur formé de la première colonne de la matrice A. Choisissons alors

$$|e_1\rangle = \frac{|u_1\rangle}{|u_1|} \quad \text{où} \quad |u_1\rangle = |x_1\rangle - |x_1||b_1\rangle \tag{7.23}$$

où $|b_1\rangle$ est le premier vecteur de base, dont les composantes sont $(1,0,0,\ldots,0)$. Soit $\mathrm{P}_1 = \mathrm{P}(e_1)$. On montre que $\mathrm{P}_1|x_1\rangle = |x_1\rangle - |u_1\rangle = |x_1||b_1\rangle$, donc que l'effet sur $|x_1\rangle$ est l'annulation de toutes ses composantes sauf la première :

$$\begin{aligned}
\mathrm{P}_1|x_1\rangle &= |x_1\rangle - |u_1\rangle \frac{(\langle x_1| - |x_1|\langle b_1|)|x_1\rangle}{\langle u_1|u_1\rangle/2} \\
&= |x_1\rangle - |u_1\rangle \frac{2(\langle x_1|x_1\rangle - |x_1|x_{11})}{2\langle x_1|x_1\rangle - 2|x_1|x_{11}} \qquad x_{11} \coloneqq \langle b_1|x_1\rangle \\
&= |x_1\rangle - |u_1\rangle \\
&= |x_1||b_1\rangle
\end{aligned} \tag{7.24}$$

Cette opération est illustrée à la figure 7.1

Dans ce cas, la matrice $\mathrm{P}_1\mathrm{A}$ verra tous ses éléments situés sous A_{11} réduits à zéro.

On peut ensuite définir une matrice de Householder P_2 d'ordre $\mathrm{N}-1$ qui n'agit que sur le bloc inférieur $(\mathrm{N}-1)\times(\mathrm{N}-1)$ de $\mathrm{P}_1\mathrm{A}$, en la définissant à partir du vecteur $|x_2\rangle$ formé des $\mathrm{N}-1$ composantes de la première colonne de ce bloc. P_2 n'agira pas sur la première colonne ou la première rangée de $\mathrm{P}_1\mathrm{A}$, et la matrice $\mathrm{P}_2\mathrm{P}_1\mathrm{A}$ n'aura pas d'éléments en dessous de A_{11} et de A_{22}, et ainsi de suite. À la fin, on trouve une matrice complètement triangulaire supérieure :

$$\mathrm{R} = \mathrm{P}_{\mathrm{N}-1}\ldots\mathrm{P}_2\mathrm{P}_1\mathrm{A} = \mathrm{Q}^{\mathrm{T}}\mathrm{A} \qquad \mathrm{Q}^{\mathrm{T}} \coloneqq \mathrm{P}_{\mathrm{N}-1}\ldots\mathrm{P}_2\mathrm{P}_1 \tag{7.25}$$

Comme les matrices P_i sont orthogonales, Q^{T} est orthogonale aussi et donc $\mathrm{Q}\mathrm{Q}^{\mathrm{T}} = \mathbb{I}$. On a donc obtenu la décomposition $\mathrm{A} = \mathrm{Q}\mathrm{R}$.

Nom	Forme	Description
LU	$A = PLU$	$A_{M\times N}$ $P_{M\times M}$: matrice de permutation $L_{M\times \min(M,N)}$: triangulaire inférieure $U_{\min(M,N)\times N}$: triangulaire supérieure
Cholesky	$A = LL^{\dagger}$	$A_{N\times N}$ hermitienne définie positive $L_{N\times N}$: triangulaire inférieure
QR	$A = QR$	$A_{M\times N}$ $Q_{M\times M}$: unitaire $R_{M\times N}$: triangulaire supérieure
valeurs singulières (SVD)	$A = U\Sigma V^{\dagger}$	$A_{M\times N}$ $U_{M\times M}$: : unitaire $\Sigma_{M\times N}$: diagonale, réelle $V_{N\times N}$: unitaire
	$A = WDV^{\dagger}$	$W_{M\times N}$ (isométrie, $M > N$) $D_{N\times N}$ (bloc non nul de Σ)
Schur	$A = ZTZ^{\dagger}$	$A_{N\times N}$ $Z_{N\times N}$: unitaire $T_{N\times N}$: triangulaire supérieure
Hessenberg	$A = QHQ^{\dagger}$	$A_{N\times N}$ $Q_{N\times N}$: unitaire $H_{N\times N}$: Hessenberg supérieure (zéros sous la première sous-diagonale inférieure)

TABLE 7.1
Principales décompositions de matrices utilisées en algèbre linéaire numérique.

B Valeurs et vecteurs propres

B.1 Généralités

L'un des problèmes les plus fréquents de l'algèbre linéaire est la recherche des valeurs propres et vecteurs propres d'une matrice. En physique, ce problème surgit dans plusieurs contextes, le plus souvent dans la détermination des modes d'oscillations en mécanique, en électromagnétisme ou en mécanique quantique. Le calcul des niveaux d'énergie d'un système décrit par la mécanique quantique (atome, molécule, structure de bandes d'un solide, etc.) entre dans cette dernière catégorie.

L'équation aux valeurs propres d'une matrice carrée A est

$$\mathrm{A}x = \lambda x \tag{7.26}$$

Étant donné A, le problème est de trouver les valeurs λ pour lesquelles cette équation a une solution non nulle. Le vecteur propre x correspondant définit en fait un sous-espace propre et peut être multiplié par une constante quelconque. La dimension du sous-espace propre est le *degré de dégénérescence* de la valeur propre λ.

Si une matrice d'ordre n possède n vecteurs propres linéairement indépendants, on dit qu'elle est *diagonalisable*. Cela signifie qu'on peut construire une base de vecteurs propres de A, et que par conséquent la matrice A est diagonale dans cette base. Spécifiquement, si les n vecteurs propres sont normalisés et forment les colonnes d'une matrice U, alors

$$\mathrm{AU} = \mathrm{UD} \implies \mathrm{U}^{-1}\mathrm{AU} = \mathrm{D} \tag{7.27}$$

où D est une matrice diagonale dont les éléments sont les valeurs propres de A (dans le même ordre que les vecteurs propres correspondants).

On montre qu'une condition suffisante pour que les vecteurs propres d'une matrice soient orthogonaux et qu'ils forment une base complète de dimension n est que la matrice A soit *normale*, c'est-à-dire qu'elle commute avec son conjugué hermitien :

$$\mathrm{AA}^\dagger = \mathrm{A}^\dagger\mathrm{A} \qquad \text{(matrice normale)} \tag{7.28}$$

Dans ce cas, les vecteurs propres étant orthogonaux, la matrice U est unitaire : $\mathrm{U}^\dagger\mathrm{U} = 1$. Une matrice hermitienne ($\mathrm{A}^\dagger = \mathrm{A}$) ou symétrique ($\mathrm{A}^\mathrm{T} = \mathrm{A}$) est nécessairement normale. Si, au contraire, la matrice n'est pas normale, alors elle peut soit avoir un ensemble complet de n vecteurs propres qui ne sont pas orthogonaux, ou encore un ensemble incomplet de vecteurs propres.

Une littérature très vaste décrit les méthodes de diagonalisation des matrices. Ces méthodes sont typiquement itératives et visent à rendre la matrice A progressivement diagonale en appliquant une suite de transformations de similitude :

1. La méthode de Jacobi est la plus ancienne et est expliquée à la sous-section suivante (B.2).
2. L'algorithme QR, expliqué par la suite, est basé sur une séquence de décompositions QR.
3. Une autre méthode consiste à réduire une matrice symétrique à une matrice tridiagonale (méthode de Householder) et ensuite à diagonaliser cette matrice tridiagonale.

En vérité, les méthodes efficaces de diagonalisation complète, par lesquelles toutes les valeurs propres et tous les vecteurs propres sont calculés, sont relativement complexes. Nous ferons appel aux méthodes préprogrammées de la bibliothèque Lapack plutôt que de les programmer. En particulier, la fonction Lapack qui permet de diagonaliser une matrice symétrique réelle porte le nom `dsyev_`. En Python, c'est la fonction `scipy.linalg.eig` qui est utilisée pour une matrice générale, ou encore la fonction `scipy.linalg.eigh` pour une matrice hermitienne ou symétrique.

B.2 Méthode de Jacobi

Considérons une matrice réelle symétrique A, qu'on désire diagonaliser. Commençons par le cas très simple d'une matrice d'ordre 2. La diagonalisation se fera via l'application d'une matrice orthogonale, dans ce cas une simple matrice de rotation dans le plan :

$$\mathrm{U}^{-1}\mathrm{AU} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{12} & \mathrm{A}_{22} \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \tag{7.29}$$

L'angle θ est déterminé par l'annulation des composantes hors diagonale :

$$-\sin\theta\cos\theta(\mathrm{A}_{11}-\mathrm{A}_{22})+\mathrm{A}_{12}(\sin^2\theta-\cos^2\theta)=0 \tag{7.30}$$

ou encore

$$\frac{1}{2}\sin(2\theta)(\mathrm{A}_{22}-\mathrm{A}_{11})=\mathrm{A}_{12}\cos(2\theta) \Longrightarrow \theta=\frac{1}{2}\arctan\frac{\mathrm{A}_{12}}{\mathrm{A}_{22}-\mathrm{A}_{11}} \tag{7.31}$$

Supposons maintenant que la matrice est d'ordre $n>2$. La méthode de Jacobi consiste à appliquer une suite de transformation du type (7.29), cette fois à l'aide de matrices de rotation $\mathrm{G}(i,j,\theta)$ qui effectuent une rotation dans le plan $x_i - x_j$:

$$\mathrm{G}(i,j,\theta)=\begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots & & \vdots & \\ 0 & \dots & \cos\theta & \dots & -\sin\theta & \dots & 0 \\ \vdots & & \vdots & \ddots & & \vdots & \\ 0 & \dots & \sin\theta & \dots & \cos\theta & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \tag{7.32}$$

Cette matrice est essentiellement la matrice identité, sauf pour les rangées et colonnes i et j, où on retrouve à la place une sous-matrice de rotation comme en (7.29). La méthode de Jacobi est itérative : à chaque étape, on sélectionne le plan (x_i, x_j) dans lequel la rotation est effectuée, en fonction de la valeur hors diagonale maximale de A (en valeur absolue). On modifie alors A ainsi :

$$\mathrm{A} \longleftarrow \mathrm{G}^{-1}(i,j,\theta)\mathrm{AG}(i,j,\theta) \tag{7.33}$$

où l'angle θ est donnée par la relation (7.31), en fonction des composantes appropriées :

$$\theta=\frac{1}{2}\arctan\frac{\mathrm{A}_{ij}}{\mathrm{A}_{jj}-\mathrm{A}_{ii}} \tag{7.34}$$

On recommence jusqu'à ce que l'angle θ soit plus petit qu'une précision donnée. En parallèle, on accumule les rotations dans une matrice U :

$$\mathrm{U} \longleftarrow \mathrm{UG}(i, j, \theta) \tag{7.35}$$

À la fin du processus, les colonnes de la matrice U contiennent les vecteurs propres de la valeur initiale de A, alors que la diagonale de la valeur finale de A contient les valeurs propres correspondantes. Cette méthode est destructive, donc il faut travailler sur une copie de A si on veut préserver l'original. Notons que la transformation (7.33) ne doit pas être implémentée comme un produit ordinaire de matrices, car la matrice G étant très simple, la transformation n'affecte qu'un petit sous-ensemble des éléments de matrice.

B.3 Algorithme QR

L'algorithme QR pour trouver les valeurs et vecteurs propres se base sur une séquence itérée de décompositions QR. Appelons $\mathrm{A} = \mathrm{A}_1$ la matrice originale, et A_k la séquence de matrices qui, par cet algorithme, se rapprochent d'une matrice diagonale. La séquence est définie de la manière suivante :

$$\mathrm{A}_k = \mathrm{Q}_k \mathrm{R}_k \qquad\qquad \mathrm{A}_{k+1} = \mathrm{R}_k \mathrm{Q}_k \tag{7.36}$$

Autrement dit, on procède à une décomposition QR de A_k et on construit A_{k+1} en multipliant les deux facteurs dans l'ordre inverse. Comme Q_k est orthogonal, $\mathrm{Q}_k^{\mathrm{T}}\mathrm{Q}_k = 1$ et donc

$$\mathrm{A}_{k+1} = \mathrm{Q}_k^{\mathrm{T}}\mathrm{Q}_k\mathrm{R}_k\mathrm{Q}_k = \mathrm{Q}_k^{-1}\mathrm{A}_k\mathrm{Q}_k \tag{7.37}$$

et donc A_{k+1} est obtenu de A_k par une transformation de similitude (c'est-à-dire un changement de base) et possède donc les mêmes valeurs propres que A_k. En itérant cette procédure, dans la plupart des cas la matrice A_k est triangulaire supérieure avec une précision suffisante pour une valeur modérée M de k et on peut écrire $\mathrm{T} = \mathrm{A}_{\mathrm{M}}$ ainsi que

$$\mathrm{A} = \mathrm{Q}_1\mathrm{Q}_2\ldots\mathrm{Q}_{\mathrm{M}-1}\mathrm{T}(\mathrm{Q}_1\mathrm{Q}_2\ldots\mathrm{Q}_{\mathrm{M}-1})^{-1} = \mathrm{QTQ}^{-1} \quad \text{où} \quad \mathrm{Q} = \mathrm{Q}_1\mathrm{Q}_2\ldots\mathrm{Q}_{\mathrm{M}-1} \tag{7.38}$$

Cette décomposition $\mathrm{A} = \mathrm{QTQ}^{-1}$ est appele *décomposition de Schur*. Comme la matrice T est triangulaire supérieure, ses valeurs propres figurent sur la diagonale et les vecteurs propres correspondants sont facilement calculables.

B.4 Méthode de Householder

La méthode de Householder permet de diagonaliser efficacement une matrice symétrique. Elle comporte deux grandes étapes : (1) la réduction de la matrice symétrique A à une matrice tridiagonale : $\mathrm{A} = \mathrm{Q}^{\mathrm{T}}\mathrm{TQ}$, où T est tridiagonale et Q est une matrice orthogonale. (2) Le calcul des valeurs et vecteurs propres de cette matrice tridiagonale, ce qui se fait par une algorithme de complexité N, N étant l'ordre de la matrice.

Expliquons la première étape. On procède comme pour la décomposition QR explique plus haut, à l'aide de réflexions de Householder, sauf qu'on adopte comme premier vecteur non pas la première colonne de A, mais le vecteur à $\mathrm{N}-1$ composantes formé des $\mathrm{N}-1$ derniers éléments de la première

colonne de A. Autrement dit, on construit un réflecteur P_1 qui n'agit pas sur la première rangée ou la première colonne de A :

$$P_1 A = \left(\begin{array}{c|cc} 1 & 0 & \\ \hline 0 & {}^{(N-1)}P_1 & \end{array}\right) A = \left(\begin{array}{c|ccc} A_{11} & A_{12} & A_{13} & \dots \\ \hline k & & & \\ 0 & & * & \\ 0 & & & \\ \vdots & & & \end{array}\right) \tag{7.39}$$

Considérons ensuite la matrice suivante :

$$A_2 = P_1 A P_1 = \left(\begin{array}{c|ccc} A_{11} & k & 0 & 0 & \dots \\ \hline k & & & & \\ 0 & & {}^{(N-1)}A_2 & & \\ 0 & & & & \\ \vdots & & & & \end{array}\right) \tag{7.40}$$

Cette matrice a bel et bien cette forme pour les raisons suivantes : (1) la post-multiplication par P_1 n'agit pas sur la première colonne de $P_1 A$. (2) $P_1 A P_1$ est symétrique ; en effet, $A_2^T = P_1^T A^T P_1^T$; comme P_1 est une réflexion, il s'ensuit que $P_1^2 = 1$ et donc que $P^{-1} = P$; d'autre part, comme P_1 est orthogonale, on a $P_1^T = P^{-1}$, et donc $P^T = P$. Donc enfin $A_2^T = P_1 A P_1 = A_2$.

La raison pour démarrer à la deuxième rangée au lieu de la première dans la construction de P_1 est précisément que nous voulons pouvoir post-multiplier par P_1 pour obtenir une transformation orthogonale : $A_2 = P_1 A P_1^T$. Une fois cette première étape passée, on recommence avec la sous-matrice ${}^{(N-1)}A_2$, et ainsi de suite, ce qui mène à une séquence de réflexions $Q = P_{N-2} P_{N-3} \cdots P_2 P_1$ qui, étant un produit de réflexions, est une matrice orthogonale, sans être elle-même une réflexion. La matrice résultante est tridiagonale :

$$T = QAQ^T \tag{7.41}$$

Cette première étape est un processus de complexité N^3 : chaque P_i doit être multipliée, ce qui est un processus de complexité N^2, et le nombre de matrices P_i distinctes est aussi d'ordre N, pour une complexité totale en N^3.

La deuxième étape consiste à diagonaliser une matrice tridiagonale : $T = UDU^{-1}$. La diagonalisation complète est alors $A = Q^T T Q = Q^T U D U^{-1} Q$. Les valeurs propres de A sont les mêmes que celles de T et les vecteurs propres de A sont les colonnes de $Q^T U$. Afin de diagonaliser T, on doit premièrement trouver les valeurs propres, qui sont les racines du polynôme caractéristique. Supposons que T, qui est symétrique, ait la forme explicite suivante :

$$T = \begin{pmatrix} a_0 & b_1 & 0 & \dots & 0 \\ b_1 & a_1 & b_2 & \dots & 0 \\ 0 & b_2 & a_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{N-1} \end{pmatrix} \tag{7.42}$$

Le polynôme caractéristique est le déterminant suivant :

$$\mathrm{D}_0 = \begin{vmatrix} a_0-\lambda & b_1 & 0 & \dots & 0 \\ b_1 & a_1-\lambda & b_2 & \dots & 0 \\ 0 & b_2 & a_2-\lambda & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{\mathrm{N}-1}-\lambda \end{vmatrix} \tag{7.43}$$

Soit D_n le sous-déterminant obtenu en enlevant les n premières rangées et colonnes de la matrice ci-dessus. D'après la définition même du déterminant, on a la relation suivante :

$$\mathrm{D}_0 = (a_0-\lambda)\mathrm{D}_1 - b_1^2\mathrm{D}_2 \tag{7.44}$$

Comme les sous-déterminants ont la même forme tridiagonale que le déterminant complet, cette relation s'applique à tous les niveaux :

$$\mathrm{D}_{n-1} = (a_{n-1}-\lambda)\mathrm{D}_n - b_n^2\mathrm{D}_{n+1} \tag{7.45}$$

On peut donc utiliser cette relation à partir de la fin ($\mathrm{D}_{\mathrm{N}-1}$ est le dernier sous-déterminant, égal à $a_{\mathrm{N}-1}-\lambda$) et ainsi calculer $\mathrm{D}_0(\lambda)$ à loisir à l'aide d'un processus qui est clairement de complexité N.

Il faut ensuite identifier les racines de $\mathrm{D}_0(\lambda)$, c'est-à-dire les valeurs propres. Nous verrons plus loin dans ce cours comment procéder, mais il s'agit aussi d'un processus de complexité N. Une fois les valeurs propres identifiées, il suffit de résoudre le système linéaire tridiagonal sous-déterminé

$$(\mathrm{T}-\lambda)x_\lambda = 0 \tag{7.46}$$

afin de trouver les vecteurs propres x_λ, ce qui est encore une fois un processus de complexité N.

B.5 Problème aux valeurs propres généralisé

Dans plusieurs applications, notamment dans la méthode des éléments finis, on fait face à un problème aux valeurs propres formulé comme suit :

$$\mathrm{A}x = \lambda \mathrm{M}x \tag{7.47}$$

où M est la matrice de masse, ou plus généralement une matrice définie positive quelconque. Ce problème porte le nom de *problème aux valeurs propres généralisé.*

Si la matrice M est régulière, alors en principe on peut l'inverser et ramener ce problème à la forme suivante : $\mathrm{M}^{-1}\mathrm{A}x = \lambda x$. Cependant, il n'est généralement pas pratique d'inverser la matrice M complètement. Par contre, comme cette matrice est définie positive, elle admet une *décomposition de Cholesky* :

$$\mathrm{M} = \mathrm{L}\mathrm{L}^{\mathrm{T}} \tag{7.48}$$

où L est une matrice triangulaire inférieure ne comportant aucun élément nul sur la diagonale. La décomposition de Cholesky se fait par un algorithme standard de complexité n^3, mais en général moins coûteux qu'une inversion complète. LAPACK permet de procéder à cette décomposition à l'aide de la méthode `dpotrf` (pour une matrice de `double`).

Une fois la décomposition de Cholesky complétée, on peut facilement inverser les matrices L et L^T (car elles sont triangulaires) et on écrit

$$L^{-1}A(L^{-1})^T L^T x = \lambda L^T x \quad \text{ou encore} \quad Cy = \lambda y \tag{7.49}$$

où on a défini $C = L^{-1}A(L^{-1})^T$ et $y = L^T x$. Le problème est donc ramené à celui des valeurs propres ordinaires.

C Décomposition en valeurs singulières

Soit A une matrice complexe rectangulaire $M \times N$. La décomposition en valeurs singulières (DVS, ou en angl. *singular value decomposition* : SVD) de A prend la forme suivante :

$$A = U\Sigma V^\dagger \tag{7.50}$$

où U est une matrice unitaire $M \times M$, V une matrice unitaire $N \times N$ et Σ une matrice réelle $M \times N$ diagonale (les valeurs diagonales de Σ sont notées σ_i et peuvent toujours être choisies positives). Cette décomposition généralise la diagonalisation d'une matrice normale à tout type de matrice, même rectangulaire. [1] Si la matrice A est réelle et non complexe, alors les matrices U et V sont orthogonales. De toute manière, $V^{-1} = V^\dagger$ et $U^{-1} = U^\dagger$.

La matrice A agit sur un espace X (son domaine) vers une espace d'arrivée Y, c'est-à-dire $A : X \to Y$. Sur un élément $x \in X$, A produit un élément y de Y : $y = Ax$. Si on procède à un changement de base dans X défini par la matrice V, le vecteur x est dorénavant représenté par $x' = V^{-1}x$. De même, en procédant à un changement de base dans Y défini par la matrice U^{-1}, le vecteur y devient $y' = U^{-1}y$ et donc la relation $y = Ax$ devient

$$y' = U^{-1}Ax = U^{-1}AVx' \quad \text{ou} \quad y' = \Sigma x' \tag{7.51}$$

La matrice Σ étant diagonale, cette forme fait ressortir clairement les vecteurs singuliers associés à la matrice A.

Considérons par exemple un système d'équations en apparence sous-déterminé (plus de variables que d'équations) : $Ax = y$, où $M < N$ et où on cherche les N composantes de x, connaissant les M composantes de y. Dans sa forme décomposée, ce système s'écrit plutôt $y' = \Sigma x'$, ou

$$\begin{pmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_M \end{pmatrix} = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \dots & 0 \\ 0 & 0 & \dots & \sigma_m & \dots & 0 \end{pmatrix} \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_M \\ \vdots \\ x'_N \end{pmatrix} \tag{7.52}$$

1. La diagonalisation d'une matrice carrée est un cas particulier de la DVS. Le fait que les valeurs propres ne soient pas toujours positives n'est pas en contradiction avec le fait que les valeurs singulières sont positives, car le signe des valeurs propres peut être caché dans la matrice V (qui n'est pas nécessairement égale à U, contrairement au cas d'une diagonalisation de matrice).

La solution à ce système est la suivante :

$$x'_i = \begin{cases} \dfrac{y'_i}{\sigma_i} & \text{si } i \leq \mathrm{M} \\ \text{arbitraire} & \text{si } \mathrm{M} < i \leq \mathrm{N} \end{cases} \tag{7.53}$$

On retourne ensuite à x à l'aide de la matrice V : $x = \mathrm{V}x'$.

Considérons maintenant un système en apparence surdéterminé (plus d'équations que de variables, ou M > N). Dans sa forme décomposée, ce système s'écrit plutôt $y' = \Sigma x'$, ou

$$\begin{pmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_n \\ \vdots \\ y'_{\mathrm{M}} \end{pmatrix} = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{\mathrm{M}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_{\mathrm{N}} \end{pmatrix} \tag{7.54}$$

La solution à ce système n'est possible que si $y'_i = 0$ ($\mathrm{N} < i \leq \mathrm{M}$). Dans ce cas, elle est simplement donnée par

$$x'_i = \frac{y'_i}{\sigma_i} \qquad (i = 1, \dots, n) \tag{7.55}$$

On retourne ensuite à x à l'aide de la matrice V : $x = \mathrm{V}x'$.

Si la matrice A est carrée (M = N), la DVS est possible même si la matrice n'est pas diagonalisable. Si A est diagonalisable, alors on a la relation simple $\mathrm{V} = \mathrm{U}^{-1}$. Si A n'est pas diagonalisable, alors $\mathrm{AA}^\dagger$ et $\mathrm{A}^\dagger\mathrm{A}$ le sont. On constate que

$$\mathrm{AA}^\dagger = \mathrm{U}\Sigma\Sigma^\dagger\mathrm{U}^\dagger \quad \text{et} \quad \mathrm{A}^\dagger\mathrm{A} = \mathrm{V}\Sigma^\dagger\Sigma\mathrm{V}^\dagger \tag{7.56}$$

Donc les valeurs singulières (les éléments de Σ) sont les racines carrées des valeurs propres de $\mathrm{AA}^\dagger$ et de $\mathrm{A}^\dagger\mathrm{A}$. Les colonnes de V sont les vecteurs propres de $\mathrm{A}^\dagger\mathrm{A}$, alors que les colonnes de U sont les vecteurs propres de $\mathrm{AA}^\dagger$.

La DVS permet aussi d'interpréter une matrice A comme une somme de produits tensoriels de vecteurs. En notation indicielle, la DVS s'écrit

$$\mathrm{A}_{ij} = \sum_k \mathrm{U}_{ik}\Sigma_{kk}\mathrm{V}^*_{jk} = \sum_k \sigma_k \mathrm{U}_{ik}\mathrm{V}^*_{jk} \tag{7.57}$$

Si la colonne n$^{\underline{o}}$ k de U est notée $\mathrm{U}_{\bullet k}$ et la colonne n$^{\underline{o}}$ k de V est notée $\mathrm{V}_{\bullet k}$, cela signifie que la matrice A peut s'écrire ainsi :

$$\mathrm{A} = \sum_k \sigma_k \mathrm{U}_{\bullet k} \otimes \mathrm{V}^*_{\bullet k} = \sum_k \sigma_k \mathrm{U}_{\bullet k}\mathrm{V}^\dagger_{\bullet k} \tag{7.58}$$

Cette représentation est importante en information quantique et est souvent appelée *décomposition de Schmidt* dans ce contexte.

isométries Si M > N, la DVS peut également s'exprimer de la manière suivante :

$$\mathrm{A} = \mathrm{W}\mathrm{D}\mathrm{V}^{\dagger} \tag{7.59}$$

où W est une matrice M × N formée des N premières colonnes de U, alors que D est une matrice N × N diagonale, soit le bloc non nul de la matrice Σ. La matrice W respecte la relation suivante : $\mathrm{W}^{\dagger}\mathrm{W} = \mathbb{I}$, qui vient du fait que ses colonnes sont orthonormées. Une matrice rectangulaire de ce type est appelée *isométrie*, car elle préserve le produit scalaire de deux vecteurs de l'espace X qui sont mis en correspondance dans l'espace Y par W :

$$y = \mathrm{W}x \ , \ y' = \mathrm{W}x' \implies y^{\dagger}y' = x^{\dagger}\mathrm{W}^{\dagger}\mathrm{W}x' = x^{\dagger}x' \tag{7.60}$$

Boîte à outils

En pratique, les méthodes d'algèbre linéaire impliquant des matrices pleines ne sont pas codées par l'utilisateur, qui a plutôt recours à une bibliothèque spécialisée et bien rodée, telle Lapack (*Linear Algebra PACKage*). Cette bibliothèque a été initialement écrite en FORTRAN, mais peut être utilisée en C ou en C++ avec les fichiers d'entête appropriés, en prenant garde au fait que les noms de fonctions établis en FORTRAN doivent être suivis d'un caractère de soulignement (`dgesv` en FORTRAN devient `dgesv_` en C ou C++), et que tous les arguments doivent être passés en référence.
En **Python**, les méthodes d'algèbre linéaire sont disponibles via `numpy.linalg` et `scipy.linalg`. Le premier module est entièrement accessible du deuxième et ce dernier est potentiellement plus rapide (il est basé sur Lapack). Il est donc conseillé d'utiliser `scipy.linalg` pour les besoins en algèbre linéaire. L'introduction à scipy.linalg constitue une excellente référence qu'il est conseillé de lire.

CHAPITRE 8

MÉTHODES POUR MATRICES CREUSES

A Matrices creuses

Une matrice est dite *creuse* si la vaste majorité de ses éléments sont nuls. Si la matrice est de grande taille, il n'est alors pas raisonnable de garder en mémoire tous les éléments de la matrice et on a recours à des représentations différentes dans lesquelles seuls les éléments non nuls sont stockés. Il existe plusieurs schémas de stockage des éléments d'une matrice creuse. Des livres entiers ont été écrits sur ce sujet.

Le schéma le plus simple consiste à conserver en mémoire trois tableaux définis comme suit :

1. Un tableau `v[N]` qui stocke les valeurs des `N` éléments de matrices non nuls, dans un ordre quelconque, typiquement l'ordre dans lequel les éléments de matrice sont produits ou calculés.
2. Un tableau `I[N]` qui stocke les indices de rangée des éléments de matrice contenus dans le tableau `v[N]`.
3. Un tableau `J[N]`, qui stocke les indices de colonne des éléments de matrice contenus dans le tableau `v[N]`.

Une routine qui ajoute au vecteur `y[]` le produit de la matrice A par un autre vecteur `x[]` contiendrait la boucle suivante :

```
  for i in range(N): y[I[i]] += v[i]*x[J[i]];
```

Dans le module scipy.sparse, ce format est appelé COO (pour «COOrdinate»).

Si on est en mesure de trier les éléments de matrice de A et qu'on sait que chaque rangée compte en moyenne plus d'un élément non nul, une façon plus économique de stocker la matrice consiste à modifier le sens d'un des trois tableaux, comme suit :

1. Le tableau `v[N]` stocke les valeurs des `N` éléments de matrices non nuls, *en progressant de gauche à droite, en ensuite de haut en bas.*
2. Le tableau `J[N]` stocke les indices de colonne des éléments du tableau `v[N]`.
3. On définit ensuite un tableau `I[n]` (n est la dimension de la matrice), qui stocke les indices des deux tableaux précédents où une nouvelle rangée commence. Ainsi, `I[i]` est l'indice du premier élément de `v[]` ou de `J[]` appartenant à la rangée `i`.

Une routine qui ajoute au vecteur `y[]` le produit de la matrice A par un autre vecteur `x[]` contiendrait cette fois la boucle suivante :

```
  for i in range(n):
      for j in range(I[i],I[i+1]): y[i] += v[j]*x[J[j]]
```

Dans le module scipy.sparse, ce format est appelé CSR (pour *compressed sparse row*). L'avantage du format CSR et aussi que l'opération de multiplication par un vecteur est plus efficace dans le cas de très grandes matrices. En effet, les éléments de matrice défilent successivement en mémoire dans le même ordre que `y` et, pour une rangée donnée, dans le même ordre que les éléments de `x`. Si le vecteur `x` est énorme et ne peut pas être stocké en entier dans la mémoire cache, alors cet arrangement minimise le nombre de «ratés de cache» (angl. *cache misses*) qui forceraient le processeur à vider et recharger la mémoire cache trop souvent.

Le module scipy.sparse offre d'autres formats de matrices creuses, et la documentation décrit sommairement les avantages et inconvénients de chacun.

L'utilisation de matrices creuses au lieu de matrices pleines permet d'accélérer les calculs et de s'attaquer à des problèmes impossibles à résoudre autrement. Par contre, les manipulations d'éléments par rangée ou par colonne, telles que requises par beaucoup de méthodes de décomposition (par exemple la décomposition LU), sont impossibles sans revenir effectivement à un stockage dense de la matrice. Il faut donc mettre au point des algorithmes qui reposent essentiellement sur l'opération matrice × vecteur, la seule qui soit efficace en fonction de matrices creuses.

B Méthode du gradient conjugué

Supposons que le système (7.1) est très grand (par exemple $n = 10^5$) et que la matrice A est creuse, de sorte qu'il est hors de question d'utiliser l'élimination gaussienne de complexité $\mathcal{O}(n^3)$. En général, un tel problème ne pourra être résolu exactement. Cependant, si la matrice A est symétrique ($\mathrm{A} = \mathrm{A}^{\mathrm{T}}$) et *définie positive*, c'est-à-dire si le produit $x^{\mathrm{T}}\mathrm{A}x$ est positif pour tout vecteur x, on peut utiliser la méthode du *gradient conjugué*, qui ne demande accès qu'à une procédure permettant de multiplier la matrice A par un vecteur quelconque. On sait généralement a priori, selon la nature du problème étudié, si la matrice A est définie positive ou non.

B.1 Directions conjuguées

On dit que deux vecteurs $|u\rangle$ et $|v\rangle$ sont *conjugués* selon A si $\langle u|\hat{\mathrm{A}}|v\rangle = 0$. En fait, comme A est symétrique définie positive, on peut définir un produit scalaire $\langle u|v\rangle_{\mathrm{A}}$ ainsi :

$$\langle u|v\rangle_{\mathrm{A}} \coloneqq \langle u|\hat{\mathrm{A}}|v\rangle = u^{\mathrm{T}}\mathrm{A}v \tag{8.1}$$

La conjugaison par rapport à A signifie simplement l'orthogonalité par rapport à ce produit. On peut donc supposer qu'il existe une base de vecteurs $|p_k\rangle$ ($k = 1, \ldots, n$) qui sont tous mutuellement conjugués. La solution recherchée $x^\star$ à l'équation $\mathrm{A}x = b$ peut donc en principe s'exprimer sur cette base :

$$|x^\star\rangle = \sum_{k=1}^{n} \alpha_k |p_k\rangle \tag{8.2}$$

et les coefficients α_k de ce développement se trouvent par projection :

$$\alpha_k = \frac{\langle p_k|x^\star\rangle_\mathrm{A}}{\langle p_k|p_k\rangle_\mathrm{A}} = \frac{\langle p_k|\hat{\mathrm{A}}|x^\star\rangle}{\langle p_k|p_k\rangle_\mathrm{A}} = \frac{\langle p_k|b\rangle}{\langle p_k|p_k\rangle_\mathrm{A}} \tag{8.3}$$

L'utilité des vecteurs conjugués vient précisément de la possibilité de remplacer $\mathrm{A}x^\star$ par b dans l'équation ci-dessus. La stratégie de la méthode sera donc de trouver une séquence de m vecteurs $|p_k\rangle$ qui nous rapproche le plus possible de la solution $|x^\star\rangle$, tout en maintenant $m \ll n$.

B.2 Algorithme du gradient

Commençons par formuler le problème linéaire en tant que problème de minimisation. Étant donné le système (7.1), on forme la fonction à n variables suivante :

$$f(x) = \frac{1}{2}\langle x|x\rangle_\mathrm{A} - \langle b|x\rangle = \frac{1}{2}x^\mathrm{T}\mathrm{A}x - b^\mathrm{T}x = \frac{1}{2}\sum_{ij}\mathrm{A}_{ij}x_i x_j - \sum_i b_i x_i \tag{8.4}$$

La solution recherchée est précisément le point $x^\star$ qui minimise la fonction f, car la condition de dérivée nulle donne précisément

$$\nabla f = \mathrm{A}x - b = 0 \tag{8.5}$$

Afin de converger vers la solution $x^\star$ à partir d'un point initial $x_0 = 0$, une méthode simple consiste à trouver la position x_1 du minimum de f dans la direction du gradient ∇f à x_0. Ensuite, on calcule de nouveau le gradient $r_1 = \nabla f|_{x_1}$ au point x_1 et on trouve la position x_2 du minimum dans cette nouvelle direction, et ainsi de suite jusqu'à convergence.

À l'étape k, le gradient au point x_k est simplement $r_k = \mathrm{A}x_k - b$. On doit ensuite minimiser la fonction $f(x_k + \alpha r_k)$ en fonction du paramètre α, ce qui mène à la condition

$$\begin{aligned} 0 &= \frac{\partial}{\partial\alpha}f(x_k+\alpha r_k) = \frac{\partial}{\partial\alpha}\left[\frac{1}{2}\langle x_k+\alpha r_k|x_k+\alpha r_k\rangle_\mathrm{A} - \langle x_k+\alpha r_k|b\rangle\right] \\ &= \alpha\langle r_k|r_k\rangle_\mathrm{A} + \langle r_k|x_k\rangle_\mathrm{A} - \langle r_k|b\rangle = \alpha\langle r_k|r_k\rangle_\mathrm{A} + \langle r_k|r_k\rangle \end{aligned} \tag{8.6}$$

et donc à la solution

$$\alpha = -\frac{\langle r_k|r_k\rangle}{\langle r_k|r_k\rangle_\mathrm{A}} \quad \text{et donc} \quad |x_{k+1}\rangle = |x_k\rangle - \frac{\langle r_k|r_k\rangle}{\langle r_k|r_k\rangle_\mathrm{A}}|r_k\rangle \tag{8.7}$$

Notons que le signe de r_k n'est pas important, puisque seule la direction compte. Il suffit donc, pour appliquer cette méthode, de construire de manière récurrente les vecteurs suivants :

$$|r_k\rangle = \hat{\mathrm{A}}|x_k\rangle - |b\rangle \qquad\qquad |x_{k+1}\rangle = |x_k\rangle - \frac{\langle r_k|r_k\rangle}{\langle r_k|r_k\rangle_\mathrm{A}}|r_k\rangle \tag{8.8}$$

avec la condition initiale $x_0 = 0$.

Cette méthode itérative est appelée *l'algorithme du gradient* (angl. *steepest descent method*). Typiquement, même en dimension 2, cette méthode requiert un nombre infini d'itérations avant de converger (voir la figure 8.1). Ce n'est donc pas la route à suivre.

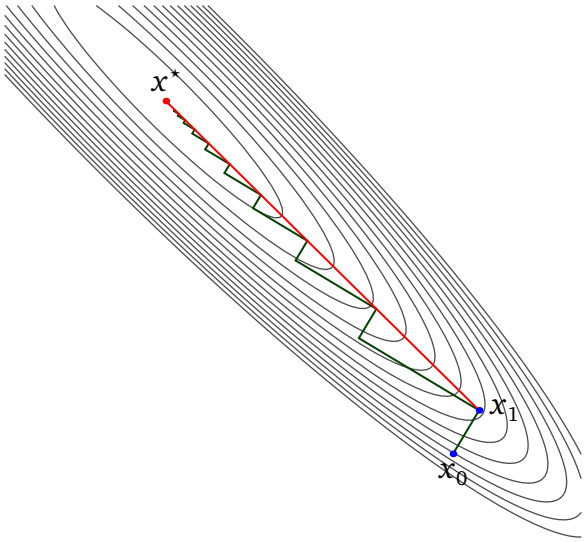


FIGURE 8.1
Comparaison entre l'algorithme du gradient (en vert) et la méthode du gradient conjugué (en rouge) dans le cas $n = 2$. Les courbes de niveau de la fonction f sont indiquées en gris. La première nécessite une infinité d'étapes avant de converger, alors que la seconde converge en deux étapes.

B.3 Minimisation le long de directions conjuguées

En fait, la direction r_k du gradient à un point donné n'est pas conjuguée aux directions précédentes. Pour que l'algorithme converge rapidement, les directions successives doivent être conjuguées selon A, ce qui permet d'épuiser systématiquement, sans répétition, les directions disponibles. L'algorithme du gradient conjugué construit plutôt des directions p_k qui sont les plus proches possible des gradients r_k, mais mutuellement conjuguées, ce qui se calcule par récurrence comme suit :

$$\begin{aligned} |r_k\rangle &= \hat{\mathrm{A}}|x_k\rangle - |b\rangle \\ |p_{k+1}\rangle &= |r_k\rangle - \sum_{i=1}^{k} \frac{\langle p_i|r_k\rangle_{\mathrm{A}}}{\langle p_i|p_i\rangle_{\mathrm{A}}}|p_i\rangle \\ |x_{k+1}\rangle &= |x_k\rangle + \frac{\langle p_{k+1}|b\rangle}{\langle p_{k+1}|p_{k+1}\rangle_{\mathrm{A}}}|p_{k+1}\rangle \end{aligned} \tag{8.9}$$

La première équation est simplement la direction du gradient au point x_k. La deuxième décrit une direction p_{k+1} obtenue de r_k en soustrayant les composantes de r_k qui sont parallèles aux directions p_i antérieures (au sens du produit $\langle\,|\,\rangle_{\mathrm{A}}$ défini plus haut). Enfin, la nouvelle position x_{k+1} correspond au minimum de la fonction f le long de cette nouvelle direction, ce qui mène à la condition

$$\alpha\langle p_{k+1}|p_{k+1}\rangle_{\mathrm{A}} + \langle p_{k+1}|x_k\rangle_{\mathrm{A}} - \langle p_{k+1}|b\rangle = 0 \tag{8.10}$$

comme $|p_{k+1}\rangle$ est une direction conjuguée à toutes les précédentes, elle est conjuguée à $|x_k\rangle$: $\langle p_{k+1}|x_k\rangle_{\mathrm{A}} = 0$. Il reste

$$\alpha = \frac{\langle p_{k+1}|b\rangle}{\langle p_{k+1}|p_{k+1}\rangle_{\mathrm{A}}} \tag{8.11}$$

d'où la dernière des relations (8.9), qui coïncide avec l'expression (8.3) pour les approximants successifs de $x^\star$.

L'algorithme est interrompu lorsque le résidu $|r_k\rangle$ est suffisamment petit. La méthode du gradient conjugué converge vers la solution exacte en au plus n itérations. En particulier, la solution en deux dimensions (voir figure 8.1) est atteinte après deux itérations seulement.

Le mérite principal de la méthode du gradient conjugué est qu'elle permet de résoudre un système linéaire d'ordre n en m étapes, où $m \ll n$, avec un nombre d'opérations de l'ordre $\mathcal{O}(nm^2) \sim \mathcal{O}(n \log n)$. Bien sûr, si on choisissait les directions p_k au hasard, le nombre d'itérations nécessaires pour converger serait de l'ordre de n et le gain serait nul. C'est le choix judicieux des directions p_k en rapport avec le gradient de f qui permet cette convergence accélérée.

Remarquons en outre que la méthode est la même si la matrice A est *définie négative* au lieu d'être *définie positive*. Il suffit pour cela de changer à la fois le signe de A et celui de b. La nouvelle matrice A est alors définie positive, mais la solution (8.9) est la même, comme on le constate immédiatement (c'est-à-dire que changer simultanément les signes de A et b n'y change rien).

C Méthode de Lanczos

La méthode de Lanczos permet de calculer les valeurs et vecteurs propres extrêmes – c'est-à-dire les plus élevées et les plus basses – d'une matrice de très grande taille. Elle se base sur une application itérative de la matrice sur des vecteurs : on doit fournir une façon d'appliquer la matrice A sur un vecteur quelconque, indépendamment de la manière dont cette matrice est emmagasinée, ce qui en fait une méthode particulièrement adaptée aux matrices creuses de très grande dimension. Certaines applications calculent même les éléments de matrice au fur et à mesure qu'ils sont requis, sans stocker la matrice d'aucune façon. [1]

L'idée de base derrière la méthode de Lanczos est de construire une projection de la matrice H (de dimension N) dont nous voulons les valeurs propres sur un sous-espace de petite dimension, mais qui contient les vecteurs propres extrêmes avec une assez bonne précision. Ce sous-espace de dimension $\mathrm{M} \ll \mathrm{N}$, appelé *sous-espace de Krylov*, est obtenu en appliquant à répétition $\mathrm{M}-1$ fois l'opérateur $\hat{\mathrm{H}}$ sur un vecteur de départ $|\phi_0\rangle$ qui peut être choisi au hasard :

$$\mathcal{K}(\phi_0, \mathrm{H}, \mathrm{M}) = \mathrm{span}\left\{|\phi_0\rangle, \hat{\mathrm{H}}|\phi_0\rangle, \hat{\mathrm{H}}^2|\phi_0\rangle, \cdots, \hat{\mathrm{H}}^{\mathrm{M}-1}|\phi_0\rangle\right\} \tag{8.12}$$

Les vecteurs $\hat{\mathrm{H}}^j|\phi_0\rangle$ ne sont pas orthogonaux, mais une base de vecteurs orthogonaux du même sous-espace est obtenue en appliquant la relation de récurrence suivante :

$$|\phi_{n+1}\rangle = \hat{\mathrm{H}}|\phi_n\rangle - a_n|\phi_n\rangle - b_n^2|\phi_{n-1}\rangle \tag{8.13}$$

avec les coefficients suivants :

$$a_n = \frac{\langle\phi_n|\hat{\mathrm{H}}|\phi_n\rangle}{\langle\phi_n|\phi_n\rangle} \qquad b_n^2 = \frac{\langle\phi_n|\phi_n\rangle}{\langle\phi_{n-1}|\phi_{n-1}\rangle} \tag{8.14}$$

1. Une référence importante sur les méthodes de solution des problèmes aux valeurs propres, en particulier pour les matrices de grande taille, est disponible en ligne : http://web.eecs.utk.edu/~dongarra/etemplates/book.html

et les conditions initiales $b_0 = 0$, $|\phi_{-1}\rangle = 0$.

Théorème 8.1 Relation d'orthogonalité dans la méthode de Lanczos
La relation de récurrence (8.13), qui peut être considérée comme une définition des vecteurs $|\phi_n\rangle$, avec les coefficients donnés en (8.14), entraîne que ces vecteurs sont mutuellement orthogonaux.

Preuve Supposons que la séquence de vecteurs construits, jusqu'à $|\phi_n\rangle$ inclus, soit déjà orthogonale. Démontrons alors que le vecteur suivant, $|\phi_{n+1}\rangle$, est orthogonal aux vecteurs précédents, en vertu des relations (8.13) et (8.14). Commençons par le produit

$$\langle\phi_n|\phi_{n+1}\rangle = \langle\phi_n|\hat{\mathrm{H}}|\phi_n\rangle - a_n\langle\phi_n|\phi_n\rangle - b_n^2\langle\phi_n|\phi_{n-1}\rangle \tag{8.15}$$

Le dernier terme à droite s'annule par orthogonalité, et les deux premiers se compensent en vertu de l'expression de a_n. Donc le produit s'annule. Considérons ensuite le produit

$$\langle\phi_{n-1}|\phi_{n+1}\rangle = \langle\phi_{n-1}|\hat{\mathrm{H}}|\phi_n\rangle - a_n\langle\phi_{n-1}|\phi_n\rangle - b_n^2\langle\phi_{n-1}|\phi_{n-1}\rangle \tag{8.16}$$

Le premier terme du membre de droite peut être récrit comme suit, en utilisant la relation de récurrence (8.13) appliquée à n au lieu de $n+1$ et en la conjuguant :

$$\langle\phi_{n-1}|\hat{\mathrm{H}}|\phi_n\rangle == \big(\langle\phi_n| + a_{n-1}\langle\phi_{n-1}| + b_{n-1}^2\langle\phi_{n-2}|\big)|\phi_n\rangle = \langle\phi_n|\phi_n\rangle \tag{8.17}$$

Il reste donc

$$\langle\phi_{n-1}|\phi_{n+1}\rangle = \langle\phi_n|\phi_n\rangle - b_n^2\langle\phi_{n-1}|\phi_{n-1}\rangle \tag{8.18}$$

ce qui s'annule en vertu de (8.14). Enfin, considérons le produit

$$\langle\phi_{n-2}|\phi_{n+1}\rangle = \langle\phi_{n-2}|\hat{\mathrm{H}}|\phi_n\rangle - a_n\langle\phi_{n-2}|\phi_n\rangle - b_n^2\langle\phi_{n-2}|\phi_{n-1}\rangle \tag{8.19}$$

Les deux derniers termes s'annulent par orthogonalité. Le premier terme du membre de droite s'annule aussi, car $\langle\phi_{n-2}|\hat{\mathrm{H}}$ ne contient pas de terme proportionnel à $\langle n|$. Il est manifeste que les produits avec les états précédents de la séquence sont nuls pour la même raison.

Pour compléter la démonstration, il faut également considérer les deux premiers termes de la séquence, qui est particulière en raison de l'inexistence de $|\phi_{-1}\rangle$. Spécifiquement,

$$|\phi_1\rangle = \hat{\mathrm{H}}|\phi_0\rangle - a_0|\phi_n\rangle \tag{8.20}$$

La même démonstration que ci-dessus s'applique pour montrer que $\langle\phi_1|\phi_0\rangle = 0$.

À chaque étape du calcul, trois vecteurs sont gardés en mémoire (ϕ_{n+1}, ϕ_n et ϕ_{n-1}). Ces vecteurs ne sont pas normalisés, mais on peut définir les vecteurs normalisés suivants :

$$|n\rangle = \frac{|\phi_n\rangle}{\sqrt{\langle\phi_n|\phi_n\rangle}} \tag{8.21}$$

En fonction de ces vecteurs orthonormés, la relation de récurrence (8.13) prend la forme suivante :

$$b_{n+1}|n+1\rangle = \hat{\mathrm{H}}|n\rangle - a_n|n\rangle - b_n|n-1\rangle \quad \text{ou encore} \quad \hat{\mathrm{H}}|n\rangle = b_n|n-1\rangle + a_n|n\rangle + b_{n+1}|n+1\rangle \tag{8.22}$$

Si on tronque l'espace de Krylov à l'étape M, l'action de la matrice H sur les vecteurs de base (8.21) peut être représentée par la matrice d'ordre M suivante :

$$\mathrm{T} = \begin{pmatrix} a_0 & b_1 & 0 & 0 & \cdots & 0 \\ b_1 & a_1 & b_2 & 0 & \cdots & 0 \\ 0 & b_2 & a_2 & b_3 & \cdots & 0 \\ 0 & 0 & b_3 & a_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{\mathrm{M}-1} \end{pmatrix} \tag{8.23}$$

Il est alors très simple de diagonaliser cette matrice tridiagonale, dont l'ordre est petit par rapport à N, et dont les valeurs propres extrêmes convergent rapidement vers les valeurs propres extrêmes de $\hat{\mathrm{H}}$.

C.1 Convergence vers les valeurs propres extrêmes

Comment peut-on démontrer de manière heuristique que les valeurs propres de la matrice T convergent vers les valeurs propres extrêmes de la matrice H ? Ceci est en fait une propriété de l'espace de Krylov. On constate sans peine les deux propriétés suivantes du sous-espace de Krylov :

1. $\mathscr{K}(\sigma\phi_0, \tau\mathrm{H}, \mathrm{M}) = \mathscr{K}(\phi_0, \mathrm{H}, \mathrm{M})$. Autrement dit, multiplier le vecteur de départ ou la matrice par une constante ne change pas le sous-espace. C'est une conséquence de la définition d'un sous-espace vectoriel.
2. $\mathscr{K}(\phi_0, \mathrm{H}+\sigma, \mathrm{M}) = \mathscr{K}(\phi_0, \mathrm{H}, \mathrm{M})$. Autrement dit, ajouter une constante à la matrice H ne change pas l'espace de Krylov. Ceci est une conséquence immédiate de la définition de l'espace de Krylov : la puissance $(\mathrm{H}+\sigma)^j$ est, après développement, une combinaison linéaire des puissances de H qui sont égales ou inférieures à j, et donc qui font déjà partie de l'espace de Krylov.

En vertu de ces deux propriétés, il n'y a aucune perte de généralité à supposer que les deux valeurs propres extrêmes de H sont -1 et 1, toutes les autres valeurs propres étant contenues entre ces deux valeurs. En effet, on peut toujours trouver les constantes σ et τ appropriées afin que les valeurs propres extrêmes de $\tau\mathrm{H}+\sigma$ soient ± 1, et l'espace de Krylov est le même pour $\tau\mathrm{H}+\sigma$ que pour H. Soit maintenant les vecteurs et valeurs propres exacts de la matrice H :

$$\hat{\mathrm{H}}|e_i\rangle = e_i|e_i\rangle \tag{8.24}$$

Le vecteur de départ $|\phi_0\rangle$ peut être développé sur une base des vecteurs propres de H :

$$|\phi_0\rangle = \sum_i \gamma_i |e_i\rangle \tag{8.25}$$

et l'application de la puissance $\hat{\mathrm{H}}^m$ sur $|\phi_0\rangle$ donne

$$\hat{\mathrm{H}}^m|\phi_0\rangle = \sum_i \gamma_i e_i^m |e_i\rangle \tag{8.26}$$

Quand m est suffisamment grand, on constate que $\hat{\mathrm{H}}^m|\phi_0\rangle$ est dominé par les deux vecteurs propres $|1\rangle$ et $|-1\rangle$. Donc ces deux vecteurs propres seront bien représentés dans l'espace de Krylov : leur projection sur ce sous-espace convergera rapidement vers l'unité en fonction du nombre d'itérations M.

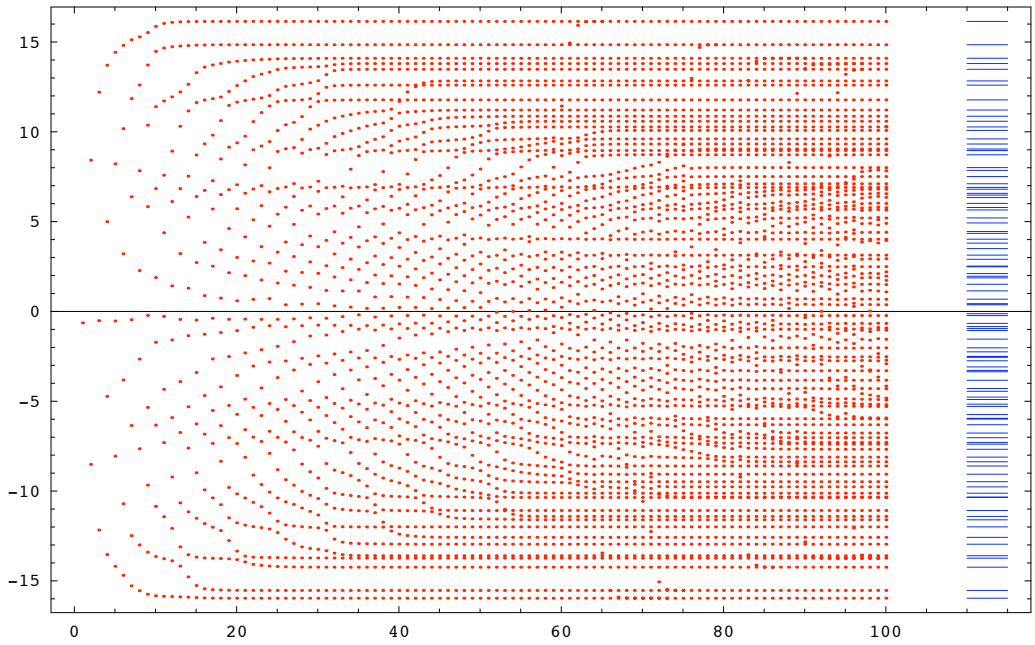


FIGURE 8.2
Illustration de la convergence des valeurs propres dans la procédure de Lanczos, en fonction du nombre d'itérations effectuées, jusqu'à M = 100. La matrice $\hat{H}$ a une dimension N = 600. On constate que les valeurs propres extrêmes convergent rapidement. À chaque itération, une valeur propre additionnelle s'ajoute à l'ensemble. On note aussi qu'à l'itération 72, un valeur propre superflue apparaît, qui se fond avec la valeur propre la plus basse un peu après ; ceci est la manifestation d'une fuite d'orthogonalité.

C.2 Calcul des vecteurs propres

À chaque itération de la procédure de Lanczos, on doit calculer les valeurs propres et vecteurs propres de la matrice tridiagonale (8.23). Une procédure efficace de complexité $\mathcal{O}(\mathrm{M})$ est disponible au sein de LAPACK pour ce type de diagonalisation. On doit ensuite utiliser un critère de convergence pour arrêter la procédure. Une critère heuristique est que le changement relatif de la valeur propre la plus basse (ou la plus élevée) d'une itération à l'autre est inférieur à une limite donnée. Un meilleur critère, que nous ne démontrerons pas ici, est de cesser les itérations quand le résidu de Ritz

$$|\mathrm{R}\rangle \coloneqq \hat{\mathrm{H}}|e_0\rangle - e_0|e_0\rangle \tag{8.27}$$

est suffisamment petit. Dans cette expression e_0 est la valeur propre la plus petite estimée à l'étape M, et $|e_0\rangle$ le vecteur propre correspondant. On montre qu'une estimation de la norme de ce résidu est donnée par la dernière composante du vecteur propre de T associé à la valeur propre la plus basse, multipliée par $b_{\mathrm{M}-1}$.

Une fois la convergence atteinte, on peut obtenir quelques vecteurs propres estimés de H (ceux associés aux valeurs propres extrêmes et quelques autres qui leur sont proches) à l'aide des vecteurs propres de la matrice T (qui sont connus dans la base de Lanczos (8.21)) et des vecteurs de cette base.

Remarques :

- Même si la relation de récurrence (8.13) garantit en principe l'orthogonalité des vecteurs de Lanczos, les erreurs d'arrondi vont éventuellement briser cette orthogonalité : il y aura des «fuites d'orthogonalité» et les M vecteurs résultants ne seront pas tous orthogonaux. Il est en fait assez courant que les états propres extrêmes soient représentés plus d'une fois dans la base de Lanczos. Ceci n'est pas grave si on ne cherche que les deux vecteurs propres extrêmes.
- La méthode de Lanczos ne peut pas déterminer la dégénérescence des valeurs propres : si les valeurs propres extrêmes sont dégénérées, alors un seul vecteur par sous-espace propre sera généralement trouvé.
- La convergence vers les vecteurs propres extrêmes est d'autant plus rapide que la différence entre ces valeurs propres et les suivantes est grande.
- La méthode de Lanczos est également valable pour estimer quelques valeurs et vecteurs propres sous-dominants (c'est-à-dire la deuxième plus grande ou petite, la troisième, etc.), mais la précision se détériore rapidement quand on s'éloigne des valeurs extrêmes.
- Une généralisation de l'algorithme de Lanczos existe pour traiter le problème aux valeurs propres généralisé $\mathrm{A}x = \lambda \mathrm{M}\,x$.

Boîte à outils

Le module scipy.sparse.linalg offre des fonctionnalités basées sur la bibliothèque ARPACK, qui implémente principalement la méthode d'Arnoldi, une généralisation de la méthode de Lanczos aux matrices non symétriques également basée sur le sous-espace de Krylov.

En particulier, les fonctions `eigs` et `eigsh` permettent de calculer les valeurs propres extrêmes de matrices respectivement non-hermitiennes et hermitiennes. Le problème aux valeurs propres généralisées peut aussi se traiter en fournissant la matrice définie positive M en argument optionnel.

D Application : chaînes de spins et modèle de Heisenberg

Dans cette section nous allons appliquer les méthodes de calcul des valeurs propres de matrices creuses à une problème physique : la détermination du gap d'excitation dans une chaîne de spins en interaction.

D.1 Produits tensoriels

Considérons un assemblage unidimensionnel (une chaîne) d'atomes en interaction mutuelle. Les différents atomes seront étiquetés par un entier $i = 0, \ldots, \mathrm{N}$. Seul le degré de liberté de spin de la dernière orbitale occupée sera considéré. L'espace de Hilbert associé à chaque atome est donc engendré par deux états : $|\uparrow\rangle$ et $|\downarrow\rangle$, désignant respectivement les états de spin up et down de l'électron qui occupe cette orbitale. Dans cet espace de dimension 2, chacune des trois composantes du spin constitue un opérateur représenté par une matrice de dimension 2 :

$$\mathrm{S}^x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \mathrm{S}^y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \mathrm{S}^z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{8.28}$$

On peut définir un vecteur $\mathbf{S}_i$ formé des trois composantes ci-dessus, associé à l'atome n$^{\underline{o}}$ i. Il s'agit d'un vecteur de matrices. En principe, il faudrait multiplier les matrices ci-dessus par $\hbar$ pour avoir le véritable opérateur de spin (c'est-à-dire pour lui donner les unités d'un moment cinétique). Nous supposerons ici que $\hbar = 1$, c'est-à-dire que nous travaillons dans un système d'unités particulier dans lequel $\hbar = 1$.

Considérons maintenant deux atomes voisins, étiquetés 1 et 2. Chaque atome aura sa copie de l'espace de Hilbert des opérateurs ci-dessus (appelons-les V_1 et V_2). D'après les principes de la mécanique quantique, l'espace de Hilbert décrivant le système complet de deux atomes sera le produit tensoriel (ou *produit de Kronecker*) $\mathrm{V}_1 \otimes \mathrm{V}_2$, de dimension 4. L'opérateur du spin du premier atome sera, dans cette espace, le produit tensoriel $\mathbf{S}_1 = \mathbf{S} \otimes \mathbb{I}$, alors que le spin du deuxième atome sera représenté par $\mathbf{S}_2 = \mathbb{I} \otimes \mathbf{S}$.

Les états de base du produit tensoriel $\mathrm{V} = \mathrm{V}_1 \otimes \mathrm{V}_2$ sont les produits tensoriels des états de base des deux facteurs et ont peut les mettre dans un ordre arbitraire. Par convention, on fait varier les états associés au facteur le plus à droite le plus rapidement. Les 4 états de base de l'espace V seraient alors

$$|\uparrow\rangle \otimes |\uparrow\rangle \qquad |\uparrow\rangle \otimes |\downarrow\rangle \qquad |\downarrow\rangle \otimes |\uparrow\rangle \qquad |\downarrow\rangle \otimes |\downarrow\rangle \tag{8.29}$$

On les note habituellement de manière plus concise :

$$|\uparrow,\uparrow\rangle \qquad |\uparrow,\downarrow\rangle \qquad |\downarrow,\uparrow\rangle \qquad |\downarrow,\downarrow\rangle \tag{8.30}$$

Dans cette base et dans cet ordre, les opérateurs S_1^x et S_2^x seront représentés par les matrices suivantes :

$$S_1^x = \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad S_2^x = \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{8.31}$$

On construit de même les autres composantes de l'opérateur de spin.

Si on ajoute un troisième atome à la chaîne, l'espace de Hilbert global devient $V = V_1 \otimes V_2 \otimes V_3$, de dimension $2^3 = 8$. Dans ce cas, une base possible pour cet espace est la suivante :

$$|\uparrow,\uparrow,\uparrow\rangle \quad |\uparrow,\uparrow,\downarrow\rangle \quad |\uparrow,\downarrow,\uparrow\rangle \quad |\uparrow,\downarrow,\downarrow\rangle \quad |\downarrow,\uparrow,\uparrow\rangle \quad |\downarrow,\uparrow,\downarrow\rangle \quad |\downarrow,\downarrow,\uparrow\rangle \quad |\downarrow,\downarrow,\downarrow\rangle \tag{8.32}$$

Les représentations, dans cette base, des opérateurs S_1^x et S_2^x sont

$$S_1^x = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad S_2^x = \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{8.33}$$

Dans le cas d'une chaîne comportant N spins, la dimension de l'espace de Hilbert est 2^N alors que le nombre d'éléments non nuls de chaque matrice décrivant l'une des composantes de l'un des spins est aussi 2^N, sur un total de 2^{2N} éléments possible. Les matrices sont donc très creuses, avec en moyenne un élément non nul par rangée.

D.2 Modèle de Heisenberg

Le modèle de Heisenberg est le premier modèle quantique du magnétisme. Il s'exprime uniquement en fonction des opérateurs de spin des atomes et est défini par l'expression ci-dessous du Hamiltonien (ou opérateur d'énergie). Pour une chaîne linéaire d'atomes, le hamiltonien a la forme suivante :

$$H = J\sum_{i=1}^{N} \mathbf{S}_i \cdot \mathbf{S}_{i+1} \tag{8.34}$$

où J est une constante ayant les unités de l'énergie et où il est compris que $\mathbf{S}_{N+1} \coloneqq \mathbf{S}_1$ (conditions aux limites périodiques). Chaque atome interagit avec son voisin immédiat. Nous allons supposer dans cette section que cette constante est positive ($J > 0$). Dans cette forme, le modèle peut être défini pour des spins de grandeur S quelconque ($S = \frac{1}{2}, 1, \frac{3}{2}, 2$, etc.). On peut montrer que la limite $S \to \infty$ correspond à la limite classique. Dans cette limite, les composantes des spins peuvent prendre des valeurs quelconques, pourvu que leur grandeur soit fixe. Si $J > 0$ et dans la limite classique, l'énergie est minimisée en adoptant une configuration de spins antiparallèles d'un site à l'autre. Une

telle configuration est qualifiée d'*antiferromagnétique*. Si $J < 0$, les spins doivent être mutuellement parallèles afin de minimiser l'énergie, ce qui constitue une configuration *ferromagnétique*.

En mécanique quantique, le problème de la minimisation de l'énergie revient à trouver la valeur propre la plus petite de la matrice H. Cette matrice est de dimension 2^N et est très creuse. Elle peut se construire par une succession de produits tensoriels.

Par exemple, dans le cas $N = 2$, on a

$$H = \frac{J}{4}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8.35}$$

L'état fondamental associé à ce hamiltonien est le vecteur propre de plus basse énergie, soit le vecteur $(0,1,-1,0)/\sqrt{2}$, de valeur propre $-3J/4$.

Dans le cas $N = 3$, la matrice d'ordre 8 est

$$H = \frac{J}{4}\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix} \tag{8.36}$$

et sa valeur propre minimale est encore $E_0 = -3J/4$.

Boîte à outils

Le module Numpy comporte la fonction kron(), qui construit le produit de Kronecker de deux tableaux, en particulier de deux matrices.
Il ne faut pas confondre avec la fonction kron() de `scipy.sparse` qui effectue la même opération, sur des matrices creuses ou pleines, mais qui retourne une matrice creuse.

D.3 Chaîne de spin $1/2$: analyse des effets de taille

Même à l'aide de matrices creuses, il est difficile d'effectuer la diagonalisation partielle du modèle de Heisenberg au-delà d'un nombre de sites de l'ordre de $N = 32$ sur un ordinateur domestique robuste. On peut espérer atteindre $N = 48$ sur un superordinateur et en programmant de manière très serrée. Par contre, on peut procéder à une extrapolation des quantités physiques d'intérêt vers la limite $N \to \infty$.

La façon la plus simple d'extrapoler vers $N \to \infty$ est d'exprimer ces quantités en fonction de $x = 1/N$, de calculer un lissage polynomial des points calculés et d'évaluer ce polynôme à $x = 0$. Il faut cependant faire preuve de prudence afin d'éviter le phénomène de Runge et s'en tenir, par exemple, à des polynômes d'ordre 2 ou 3. La figure 8.3 illustre le comportement de l'énergie du fondamental en fonction de la longueur inverse de la chaîne et les extrapolations qu'on peut utiliser pour estimer la limite $N \to \infty$. La figure 8.4 illustre les mêmes techniques démontrant que la différence d'énergie entre le premier niveau excité et l'état fondamental (ou *gap*) tend vers zéro dans cette limite.

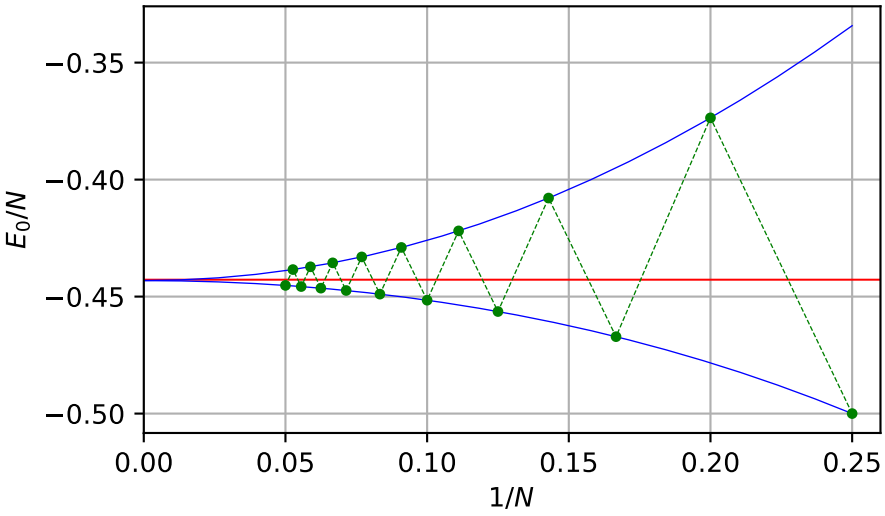


FIGURE 8.3
Énergie du fondamental de la chaîne de spin $\frac{1}{2}$ en fonction de $1/N$. Les courbes sont des lissages sur des polynômes de degré 4 à partir des points pairs et impairs. La droite rouge est l'extrapolation obtenue à l'aide de l'algorithme epsilon sur les tailles de $N = 4$ à $N = 20$.

D.4 Chaîne de spin 1 : gap de Haldane

Dans le cas où les spins de chaque atome sont de grandeur 1 au lieu de $1/2$, les opérateurs du spin ne sont représentés par les matrices de Pauli, mais plutôt par les matrices suivantes :

$$S^x = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad S^y = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & -i & 0 \\ i & 0 & -i \\ 0 & i & 0 \end{pmatrix} \qquad S^z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{8.37}$$

Le modèle de Heisenberg se définit comme pour la chaîne de spin $\frac{1}{2}$, sauf que l'espace de Hilbert d'une chaîne de N atomes est de dimension 3^N au lieu de 2^N. On peut procéder à la même analyse en taille finie que pour la chaîne de spin $\frac{1}{2}$, sauf qu'à puissance de calcul égale, nous devons nous limiter à des chaînes plus petites.

La surprise qui nous attend dans ce cas est que la valeur du gap entre le fondamental et le premier état excité ne tend pas vers zéro quand $N \to \infty$. L'extrapolation vers $N \to \infty$ obtenue à l'aide de

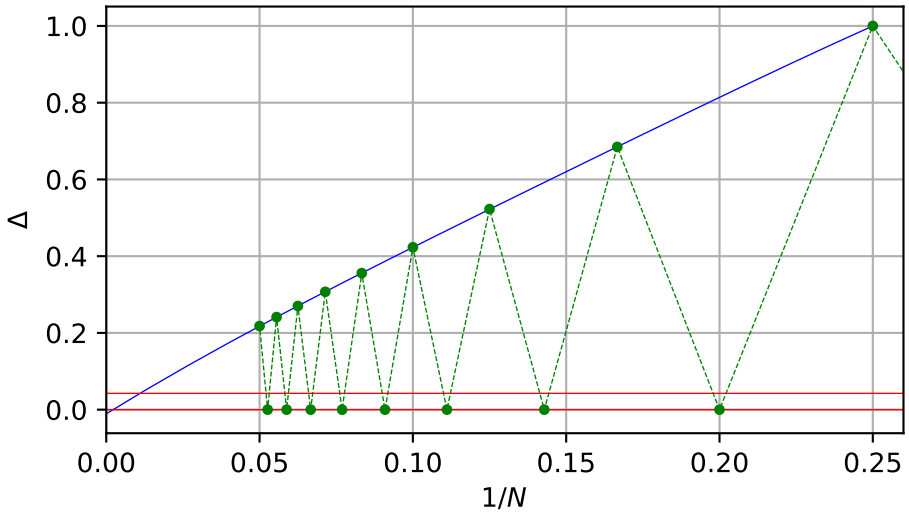


FIGURE 8.4
Différence entre l'énergie de l'état fondamental et celle du premier état excité (gap) dans la chaîne de spin $\frac{1}{2}$ en fonction de 1/N. Le fondamental est dégénéré pour N impair, en raison du théorème de Kramers et donc le gap est nul dans ce cas. La courbe est un lissage sur polynôme de degré 4 sur les valeurs impaires, qui extrapole à $\Delta_\infty \sim 0.01$. Les droites rouges sont les extrapolations obtenues à l'aide de l'algorithme epsilon, soit avec des valeurs extrêmes paires ($\Delta_\infty \sim 0.04$) ou des valeurs extrêmes impaires ($\Delta_\infty \sim 2\times10^{-9}$).

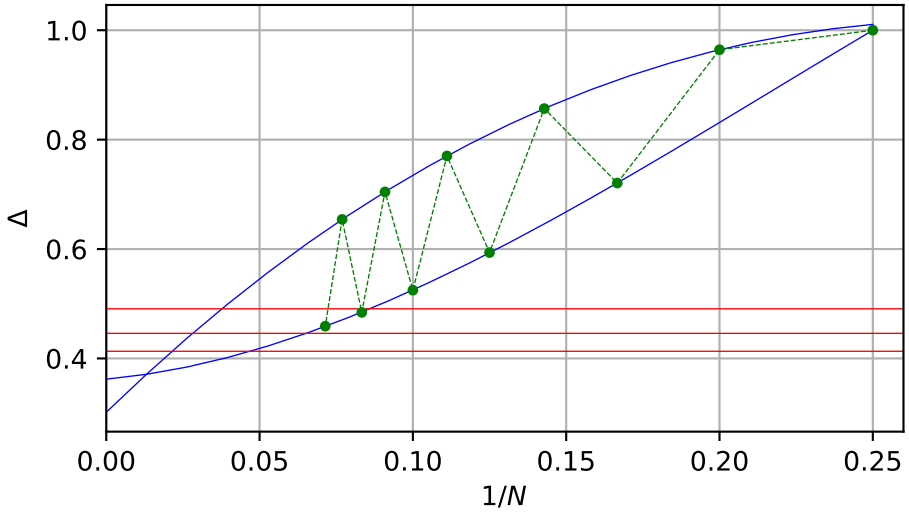


FIGURE 8.5
Différence entre l'énergie de l'état fondamental et celle du premier état excité (gap) dans la chaîne de spin 1 en fonction de 1/N. Les courbes sont des lissages sur des polynômes de degré 3, pour les valeurs paires et impaires de N. La courbe est un lissage sur polynôme de degré 4 sur les valeurs impaires, qui extrapole à $\Delta_\infty \sim 0.01$. Les droites rouges sont les extrapolations obtenues à l'aide de l'algorithme epsilon, soit avec des valeurs paires ($\Delta_\infty \sim 0.413$) ou impaires, ou les deux ensemble.

l'algorithme epsilon pour des valeurs de N allant de 4 à 14 est $\Delta_\infty \approx 0.413$ (voir figure 8.5). Une valeur plus précise, obtenue à l'aide d'une méthode sophistiquée appelée *Density matrix renormalization group* (DMRG), est $\Delta_\infty = 0.41047925(4)$. [2] Haldane (prix Nobel, 2016) a montré, à l'aide d'arguments basés sur la théorie quantique de champs et la topologie, qu'une chaîne de spin demi-entier ($\frac{1}{2}$, $\frac{3}{2}$, etc.) décrite par le modèle de Heisenberg possède un gap nul, alors qu'un chaîne de spin entier (1, 2, etc.) décrite par le même modèle possède un gap non nul, appelé *gap de Haldane*.

D.5 Annexe : algorithme epsilon

Supposons qu'on dispose d'une suite de valeurs S_n ($n = 0, 1, 2, 3, \ldots, N$) et qu'on désire connaître la limite S_∞, sans avoir d'information particulière sur le comportement théorique de cette suite. Une méthode d'*accélération de la convergence* consiste à définir une autre suite A_n à l'aide de la suite S_n qui en principe converge vers la même limite ($A_\infty = S_\infty$), mais plus rapidement. L'un des algorithmes connus en ce sens est la méthode du Δ^2 de Aitken. Les transformations de Shanks sont aussi très utilisées. L'algorithme epsilon [3] est proche de la méthode de Shanks, et constitue un généralisation de la méthode du Δ^2. Nous allons décrire l'algorithme sans justification, car cela nous mènerait loin des objectifs de ce chapitre.

La procédure consiste à construire un tableau triangulaire $\epsilon_k^{(j)}$ de la manière suivante :

1. On initialise la frontière gauche du tableau en posant

$$\epsilon_{-1}^{(n)} = 0 \qquad \epsilon_0^{(n)} = S_n \tag{8.38}$$

2. Ensuite on complète le reste du tableau en appliquant la règle suivante :

$$\epsilon_{k+1}^{(n)} = \epsilon_{k-1}^{(n+1)} + \frac{1}{\epsilon_k^{(n+1)} - \epsilon_k^{(n)}} \qquad \forall n, k \tag{8.39}$$

Cette règle nous fait progresser de la gauche vers la droite du tableau, qui prend alors une forme triangulaire :

$$\begin{array}{ccccccccc}
0 & & & & & & & & \\
 & \epsilon_0^{(0)} = S_0 & & & & & & & \\
0 & & \epsilon_1^{(0)} & & & & & & \\
 & \epsilon_0^{(1)} = S_1 & & \epsilon_2^{(0)} & & & & & \\
0 & & \epsilon_1^{(1)} & & \epsilon_3^{(0)} & & & & \\
 & \epsilon_0^{(2)} = S_2 & & \epsilon_2^{(1)} & & \epsilon_4^{(0)} & & & \\
0 & & \epsilon_1^{(2)} & & \epsilon_3^{(1)} & & & & \\
 & \epsilon_0^{(3)} = S_3 & & \epsilon_2^{(2)} & & & & & \\
0 & & \epsilon_1^{(3)} & & & & & & \\
 & \epsilon_0^{(4)} = S_4 & & & & & & & \\
0 & & & & & & & &
\end{array} \tag{8.40}$$

2. White, S. R., & Affleck, I. (2008). Spectral function for the S = 1 Heisenberg antiferromagetic chain. Physical Review **B**, 77(13), 134437.

3. Voir Graves-Morris, P. R., Roberts, D. E. & Salam, A. The epsilon algorithm and related topics. J. Comput. Appl. Math. 122, 5180 (2000).

Chaque élément du tableau est déterminé par les trois éléments qui figurent directement à gauche, en haut à gauche et en bas à gauche, ce qui donne au tableau sa forme triangulaire. Le dernier élément à droite du tableau constitue le prédicteur S_∞.

Considérons l'exemple suivant : La série de Gregory pour la fonction $\arctan(x)$ est connue pour sa convergence très lente :

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots \tag{8.41}$$

En posant $x = 1$, on trouve

$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots\right) \tag{8.42}$$

Les cinq premières sommes partielles sont

n	S_n
0	4.0
1	2.666666666666667
2	3.466666666666667
3	2.8952380952380956
4	3.3396825396825403

En appliquant l'algorithme epsilon à cette suite, on trouve le tableau suivant :

0					
	4.0				
0		−0.75			
	2.666666666666667		3.16666667		
0		1.25		−28.75	
	3.466666666666667		3.13333333		3.14234234
0		−1.75		82.25	
	2.8952380952380956		3.1452381		
0		2.25			
	3.3396825396825403				
0					

(8.43)

Le prédicteur de π, soit 3.142, est bien meilleur que la somme partielle $S_4 \approx 3.340$

CHAPITRE 9

INTERPOLATION DES FONCTIONS

A Polynômes interpolants

L'existence supposée d'un développement de Taylor limité pour la fonction $\psi(x)$ autour de chaque point signifie que la fonction peut être adéquatement représentée par un polynôme en tronquant cette série. Nous pouvons donc évaluer approximativement la fonction $\psi(x)$ en tout point, même si elle n'est connue que sur une grille discrète, en procédant à une interpolation polynomiale. La précision de cette approximation sera en proportion du degré du polynôme utilisé.

Les polynômes en question se trouvent par la *formule de Lagrange*. Si on connait la fonction $\psi_i = \psi(x_i)$ à M points notés $x_0, \ldots, x_{\mathrm{M}-1}$ (attention : M n'est pas nécessairement égal à N, mais peut ne représenter qu'un petit sous-ensemble de points contigus) alors le polynôme de degré M−1 unique qui passe par tous ces points est

$$\begin{aligned} \mathrm{P}(x) &= \frac{(x-x_1)(x-x_2)\cdots(x-x_{\mathrm{M}-1})}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_{\mathrm{M}-1})}\psi_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_{\mathrm{M}-1})}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_{\mathrm{M}-1})}\psi_1 + \cdots \\ &\qquad + \frac{(x-x_0)(x-x_1)\cdots(x-x_{\mathrm{M}-2})}{(x_{\mathrm{M}-1}-x_0)(x_{\mathrm{M}-1}-x_1)\cdots(x_{\mathrm{M}-1}-x_{\mathrm{M}-2})}\psi_{\mathrm{M}-1} \\ &= \sum_{j=0}^{\mathrm{M}-1}\left[\psi_j \prod_{\substack{i \\ (i\neq j)}}^{\mathrm{M}-1} \frac{x-x_i}{x_j-x_i}\right] \end{aligned} \tag{9.1}$$

Il est manifeste que ce polynôme est de degré M−1, car chaque terme est le produit de M−1 facteurs impliquant x. D'autre part, le coefficient du terme de degré M − 1 est

$$\sum_{j=0}^{\mathrm{M}-1}\left[\prod_{i\ (i\neq j)}^{\mathrm{M}-1} \frac{1}{x_j-x_i}\psi_j\right] \tag{9.2}$$

et ce terme est généralement non nul (il peut l'être accidentellement, bien sûr). Ensuite, ce polynôme passe manifestement par tous les points (x_i, ψ_i), car si $x = x_k$, seul le terme $j = k$ de la somme est non nul et la fraction correspondante est alors égale à l'unité, ne laissant que $\mathrm{P}(x_k) = \psi_k$. La contrainte que le polynôme passe par M−1 points donnés fixe complètement les M−1 paramètres nécessaires pour spécifier uniquement un polynôme de degré M − 1. La solution de Lagrange est donc unique.

Par exemple, la formule d'interpolation linéaire entre deux points x_0 et x_1 est

$$\mathrm{P}(x) = \frac{x-x_1}{x_0-x_1}\psi_0 + \frac{x-x_0}{x_1-x_0}\psi_1 \tag{9.3}$$

et la formule d'interpolation quadratique entre trois points $x_{0,1,2}$ est

$$\mathrm{P}(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\psi_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\psi_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\psi_2 \tag{9.4}$$

Il est possible de coder directement la formule de Lagrange (9.1), mais un algorithme plus efficace existe pour cela, et donne accès à l'erreur commise entre les interpolations de degré M−2 et M−1. [1]

B Cubiques raccordées

L'interpolation linéaire d'une fonction, d'un point à l'autre de la grille, produit une fonction approchée dont la dérivée seconde est nulle partout, sauf aux points de grille où elle est infinie. Pour pallier ce problème, on a fréquemment recours aux cubiques raccordées (angl. *cubic splines*) : il s'agit d'une approximation cubique qui a l'avantage de produire une fonction approchée dont les dérivées premières et secondes sont continues aux points de grilles.

Étant donnés deux points x_i et x_{i+1} de la grille, le polynôme cubique $\mathrm{P}_i(x)$ à 4 paramètres doit alors obéir à 4 conditions :

$$\mathrm{P}_i(x_i) = \psi(x_i) \qquad \mathrm{P}_i(x_{i+1}) = \psi(x_{i+1}) \qquad \mathrm{P}'_i(x_i) = \mathrm{P}'_{i-1}(x_i) \qquad \mathrm{P}''_i(x_i) = \mathrm{P}''_{i-1}(x_i) \tag{9.5}$$

ce qui le détermine uniquement. Notez que les contraintes d'égalité des dérivées première et seconde au point x_{i+1} s'appliquent au polynôme $\mathrm{P}_{i+1}(x)$ dans cette convention. Reste l'éternel problème des extrémités. Plusieurs possibilités existent ;

1. On peut imposer une valeur nulle des dérivées secondes aux deux extrémités, et obtenir ce qu'on appelle des cubiques raccordées *naturelles*.
2. On peut obéir à des conditions aux limites externes qui fixent les valeurs de la dérivée première aux deux extrémités, ce qui permet de déterminer les dérivées secondes.

Démontrons maintenant explicitement comment produire le polynôme interpolant. Rappelons que l'interpolation linéaire a la forme

$$\mathrm{P}(x) = \mathrm{A}\psi_i + \mathrm{B}\psi_{i+1} \quad \text{où} \quad \mathrm{A} = \frac{x-x_{i+1}}{x_i-x_{i+1}} \quad \text{et} \quad \mathrm{B} = 1-\mathrm{A} = \frac{x-x_i}{x_{i+1}-x_i} \tag{9.6}$$

Une interpolation cubique peut être mise sous la forme suivante :

$$\mathrm{P}_i(x) = a(x-x_i) + c(x-x_i)^3 + b(x-x_{i+1}) + d(x-x_{i+1})^3 \tag{9.7}$$

Cette forme contient 4 paramètres à déterminer, donc suffisamment générale pour un polynôme cubique. Pour déterminer ces 4 paramètres, nous allons imposer les valeurs de la fonction aux deux

1. Voir *Numerical Recipes*, section 3.2.

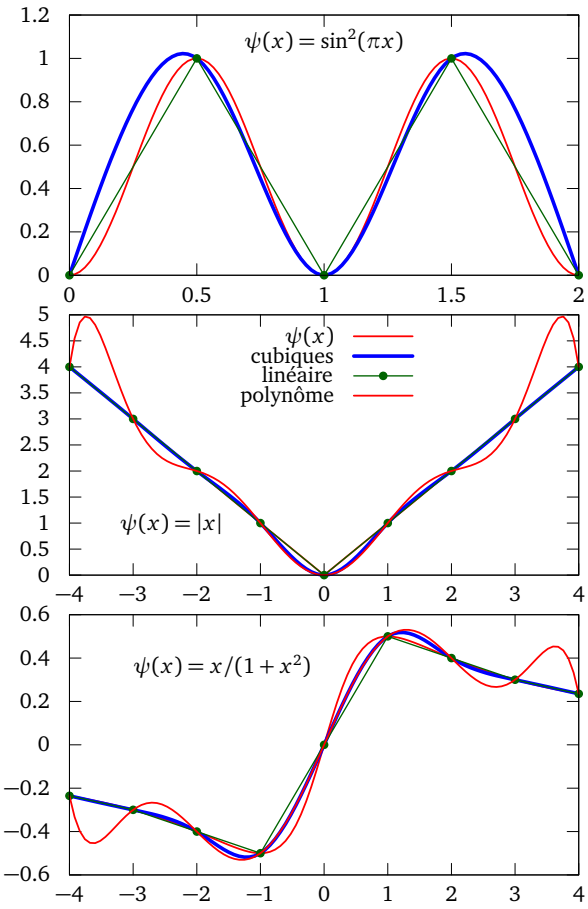


FIGURE 9.1
Exemples d'interpolation à l'aide de cubiques raccordées. Remarquons que les cubiques naturelles s'accordent mal aux extrémités lorsque la dérivée seconde est non négligeable à ces points (graphique du haut) et qu'elles représentent en général assez mal les points où la dérivée seconde est infinie (graphique du milieu). Les points de la grille sont indiqués, et sont peu nombreux. L'interpolant de Lagrange basé sur les 9 points de l'échantillon est aussi illustré ; son comportement est très oscillant par rapport aux cubiques raccordées.

extrémités de l'intervalle : $\mathrm{P}(x_i)=\psi_i$ et $\mathrm{P}(x_{i+1})=\psi_{i+1}$. Nous allons également supposer connues les deuxièmes dérivées aux mêmes points : $\mathrm{P}''(x_i)=\psi''_i$ et $\mathrm{P}''(x_{i+1})=\psi''_{i+1}$. Notons à cet effet que

$$\mathrm{P}''_i(x)=6c(x-x_i)+6d(x-x_{i+1}) \tag{9.8}$$

Les conditions ci-dessus mènent aux équations suivantes :

$$\begin{aligned}\psi_i &= b(x_i-x_{i+1})+d(x_i-x_{i+1})^3\\ \psi_{i+1} &= a(x_{i+1}-x_i)+c(x_{i+1}-x_i)^3\\ \psi''_i &= 6d(x_i-x_{i+1})\\ \psi''_{i+1} &= 6c(x_{i+1}-x_i)\end{aligned} \tag{9.9}$$

ou, de manière équivalente,

$$\begin{aligned} a &= \frac{\psi_{i+1}}{x_{i+1}-x_i}-\frac{1}{6}\psi''_{i+1}(x_{i+1}-x_i) & c &= \frac{1}{6}\frac{\psi''_{i+1}}{x_{i+1}-x_i}\\ b &= \frac{\psi_i}{x_i-x_{i+1}}-\frac{1}{6}\psi''_i(x_i-x_{i+1}) & d &= \frac{1}{6}\frac{\psi''_i}{x_i-x_{i+1}}\end{aligned} \tag{9.10}$$

ce qui équivaut à la forme suivante pour le polynôme interpolant :

$$\mathrm{P}_i(x)=\mathrm{A}\psi_i+\frac{1}{6}(\mathrm{A}^3-\mathrm{A})\psi''_i(x_i-x_{i+1})^2+\mathrm{B}\psi_{i+1}+\frac{1}{6}(\mathrm{B}^3-\mathrm{B})\psi''_{i+1}(x_i-x_{i+1})^2 \tag{9.11}$$

A et B étant définis en (9.6). Attention à la notation cependant : les valeurs ψ''_i et ψ''_{i+1} ne sont pas nécessairement les vraies dérivées secondes de la fonction $\psi(x)$ sous-jacente, alors qu'on suppose que les valeurs ψ_i sont véritablement $\psi(x_i)$. Ce sont plutôt les dérivées secondes des polynômes interpolants, que nous allons raccorder à chaque point.

La condition qui nous manque pour déterminer ψ''_i est la continuité de la dérivée à chaque point. On calcule à cette fin que

$$\mathrm{P}'_i(x)=\frac{\psi_{i+1}-\psi_i}{x_{i+1}-x_i}-\frac{1}{6}\left[(3\mathrm{A}^2-1)\psi''_i-(3\mathrm{B}^2-1)\psi''_{i+1}\right](x_{i+1}-x_i) \tag{9.12}$$

et donc que

$$\begin{aligned}\mathrm{P}'_i(x_i) &= \frac{\psi_{i+1}-\psi_i}{x_{i+1}-x_i}-\frac{1}{6}(2\psi''_i+\psi''_{i+1})(x_{i+1}-x_i)\\ \mathrm{P}'_i(x_{i+1}) &= \frac{\psi_{i+1}-\psi_i}{x_{i+1}-x_i}+\frac{1}{6}(\psi''_i+2\psi''_{i+1})(x_{i+1}-x_i)\end{aligned} \tag{9.13}$$

La condition manquante est alors $\mathrm{P}'_i(x_i)=\mathrm{P}'_{i-1}(x_i)$, ou encore

$$\frac{\psi_i-\psi_{i+1}}{x_{i+1}-x_i}-\frac{1}{6}(2\psi''_i+\psi''_{i+1})(x_{i+1}-x_i)=\frac{\psi_{i-1}-\psi_i}{x_i-x_{i-1}}+\frac{1}{6}(\psi''_{i-1}+2\psi''_i)(x_i-x_{i-1}) \tag{9.14}$$

ce qui s'exprime également comme

$$\frac{x_i-x_{i-1}}{6}\psi''_{i-1}+\frac{x_{i+1}-x_{i-1}}{3}\psi''_i+\frac{x_{i+1}-x_i}{6}\psi''_{i+1}=\frac{\psi_{i+1}-\psi_i}{x_{i+1}-x_i}-\frac{\psi_i-\psi_{i-1}}{x_i-x_{i-1}} \tag{9.15}$$

Il s'agit d'un système d'équations linéaires qui nous permet de déterminer les dérivées secondes ψ_i'', connaissant les points x_i et les valeurs ψ_i de la fonction. Ce système est tridiagonal, ce qui est un avantage calculatoire : les systèmes tridiagonaux à N variables étant résolus par un algorithme de complexité $\mathcal{O}(\mathrm{N})$.

Résumons. Chacun des $\mathrm{N}-1$ intervalles comporte un polynôme interpolant de degré 3, associé à 4 paramètres inconnus. On doit donc disposer de 4 conditions par intervalle pour fixer la valeur de ces paramètres, soit $4\mathrm{N}-4$ conditions au total. Ces conditions sont données à l'éq. (9.5). Chacun des $\mathrm{N}-2$ points intérieurs fournit 4 conditions complètes, alors que les 2 points aux extrémités n'apportent qu'une seule condition, donc au total le nombre de conditions posées est $4(\mathrm{N}-2)+2 = 4\mathrm{N}-6$. Il manque donc deux conditions, qui sont déterminées par le choix de conditions aux limites : soit on fixe les deuxièmes dérivées à 0 aux extrémités (cubiques raccordées naturelles) ou on fixe les dérivées premières ; dans les deux cas, on obtient deux conditions supplémentaires qui permettent de fermer la boucle et de déterminer tous les paramètres des cubiques, à l'aide d'un système linéaire tridiagonal.

C Approximants de Padé

On appelle *fonction rationnelle* le quotient de deux polynômes :

$$\mathrm{R}_{m,n}(x) = \frac{p_m(x)}{q_n(x)} \tag{9.16}$$

où p_m est un polynôme de degré m et q_n un polynôme de degré n. Contrairement aux polynômes, les fonctions rationnelles peuvent avoir des singularités (pôles) dans un domaine donné, là où sont les racines de q_n. Cette caractéristique leur permet de servir d'approximant à des fonctions qui possèdent elles-aussi de telles singularités.

Un cas particulier d'approximation par fonction rationnelle est l'*approximant de Padé*, qui sert en quelque sorte de «rallonge» à une série de Taylor et qui est défini comme suit : étant donnée une fonction $f(x)$ qui possède le développement de Taylor suivant :

$$f(x) = \sum_{k=0}^{\infty} c_k x^k \tag{9.17}$$

l'approximant de Padé $\mathrm{R}_{m,n}(x)$ de cette fonction est une fonction rationnelle telle que les $m+n+1$ premiers termes du développement de Taylor de $\mathrm{R}_{m,n}(x)$ coïncident avec ceux de $f(x)$.

Le numérateur et le dénominateur de $\mathrm{R}_{m,n}(x)$ sont définis par des coefficients :

$$p_m(x) = \sum_{k=0}^{m} a_k x^k \qquad\qquad q_n(x) = \sum_{k=0}^{n} b_k x^k \tag{9.18}$$

où on peut supposer sans perte de généralité que $b_0 = 1$. En pratique, on procède comme suit pour déterminer les coefficients inconnus a_k et b_k : on applique la relation $f(x)q_n(x) = p_m(x)$ aux développements ci-dessus et on fait correspondre les puissances semblables de x. On se retrouve alors

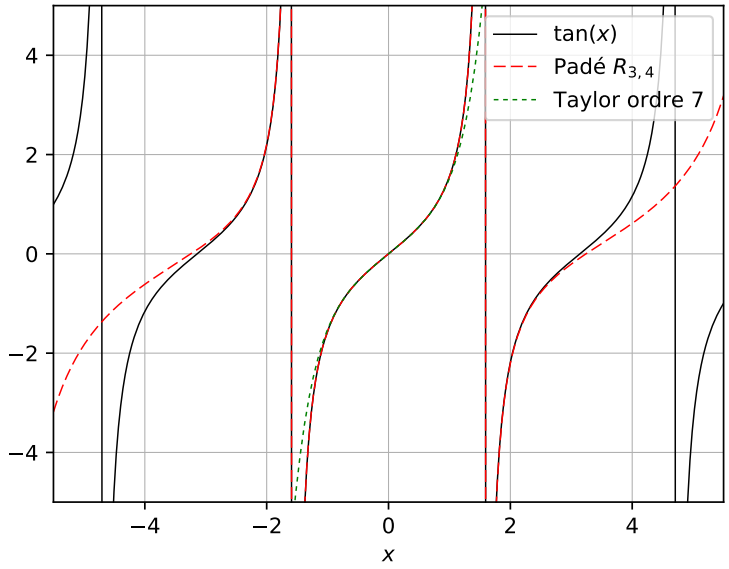


FIGURE 9.2
La fonction tan(x), sa série de Taylor à l'ordre 7 et l'approximant de Padé $\mathrm{R}_{3,4}(x)$ tirée de ce développement limité.

avec un ensemble d'équations linéaires pour a_k et b_k qui peut se résoudre par les méthodes habituelles.

Par exemple, considérons le développement de Taylor de la fonction tan(x) à l'ordre 7 :

$$\tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \mathrm{O}\left(x^9\right) \tag{9.19}$$

et voyons comment représenter ce développement par un approximant de Padé $\mathrm{R}_{3,4}(x)$. Comme la fonction tan(x) est impaire, nous pouvons supposer que p_3 est impair et q_4 pair, ce qui élimine d'emblée la moitié de leurs coefficients : $a_0 = a_2 = 0$ et $b_1 = b_3 = 0$. Les coefficients restants doivent respecter l'identité suivante à l'ordre 7 :

$$(1 + b_2 x^2 + b_4 x^4)\left(x + \frac{1}{3}x^3 + \frac{2}{15}x^5 + \frac{17}{315}x^7\right) = (a_1 x + a_3 x^3) \tag{9.20}$$

ce qui entraîne

$$1 = a_1 \qquad \frac{1}{3} + b_2 = a_3 \qquad \frac{2}{15} + \frac{b_2}{3} + b_4 = 0 \qquad \frac{17}{315} + \frac{2}{15}b_2 + \frac{1}{3}b_4 = 0 \tag{9.21}$$

La solution à ces équations linéaires couplées est

$$a_1 = 1 \qquad a_3 = -\frac{2}{21} \qquad b_2 = -\frac{3}{7} \qquad b_4 = \frac{1}{105} \tag{9.22}$$

et l'approximant de Padé est

$$\mathrm{R}_{3,4}(x) = \frac{x - \frac{2}{21}x^3}{1 - \frac{3}{7}x^2 + \frac{1}{105}x^4} \tag{9.23}$$

Le dénominateur $q_4(x)$ possède des racines à $x = \pm\sqrt{45/2 \pm \sqrt{1605}/2)} \approx (1.57123, 6.5216)$. La première est proche de la singularité $\pi/2 \approx 1.57080$ de la fonction $\tan(x)$.

La figure 9.2 compare la fonction $\tan(x)$, sa série de Taylor à l'ordre 7, ainsi que l'approximant de Padé $R_{3,4}(x)$. Il est remarquable que l'approximant de Padé réussit à décrire de manière satisfaisante la fonction $\tan(x)$, y compris sa première singularité et au-delà, à partir d'un développement de Taylor dont le rayon de convergence s'arrête à cette singularité.

L'approximant de Padé est un outil potentiellement très puissant. Par contre, il s'agit d'une approximation non contrôlée, c'est-à-dire que l'erreur produite n'est pas théoriquement bornée. Il faut donc prendre garde et ne pas l'utiliser sans réfléchir ou, du moins, sans tenter de la vérifier d'une certaine manière.

CHAPITRE 10

POLYNÔMES ORTHOGONAUX

A Généralités

Considérons les fonctions définies en une dimension sur l'intervalle $[a,b]$. Définissons sur cet intervalle un produit scalaire de fonctions par l'expression suivante :

$$\langle\psi|\psi'\rangle = \int_a^b \mathrm{d}x\, w(x)\psi(x)\psi'(x) \tag{10.1}$$

où $w(x)$ est la *fonction poids* associée au produit scalaire. Cette fonction est par hypothèse partout *positive* dans l'intervalle $[a,b]$. Nous pouvons toujours construire une base de polynômes $p_j(x)$ qui sont orthogonaux par rapport à ce produit scalaire. En effet, il suffit de commencer par un polynôme de degré 0, et ensuite d'ajouter à cette base des polynômes de degrés croissants obtenus par la procédure d'orthogonalisation de Gram-Schmidt. Il n'est pas nécessaire de les normaliser : souvent, on préfère travailler avec des polynômes dont le coefficient du plus haut degré est l'unité, c'est-à-dire $p_j(x) = x^j + c_{j,j-1}x^{j-1} + \ldots$ (on dit qu'ils sont *unitaires* (angl. *monic*)). Ces polynômes ont donc la propriété que

$$\langle p_i|p_j\rangle = \delta_{ij}\gamma_i \quad \text{ou} \quad \int_a^b w(x)p_i(x)p_j(x)\mathrm{d}x = \delta_{ij}\gamma_i \qquad (i,j=0,1,2,\ldots) \tag{10.2}$$

Les polynômes orthogonaux unitaires s'obtiennent de la relation de récurrence suivante :

$$\begin{aligned} p_{-1}(x) &\coloneqq 0 \\ p_0(x) &= 1 \\ p_{j+1}(x) &= (x-a_j)p_j(x) - b_j^2 p_{j-1}(x) \end{aligned} \tag{10.3}$$

où

$$a_j = \frac{\langle xp_j|p_j\rangle}{\langle p_j|p_j\rangle} \quad (j\geq 0) \qquad\qquad b_j^2 = \frac{\langle p_j|p_j\rangle}{\langle p_{j-1}|p_{j-1}\rangle} \quad (j\geq 1) \tag{10.4}$$

La similitude avec la relation de récurrence de l'algorithme de Lanczos n'est pas accidentelle. Ici x joue le rôle de $\hat{\mathrm{H}}$, mais sinon la preuve de l'orthogonalité qui résulte de cette relation de récurrence est identique.

Généralement, les polynômes orthogonaux ne sont pas définis comme étant unitaires, mais respectent plutôt une autre condition de normalisation. Par exemple, les polynômes de Legendre

Nom	intervalle	$w(x)$	relation de récurrence	norme h_j
Legendre	$[-1,1]$	1	$(j+1)\mathrm{P}_{j+1}=(2j+1)x\mathrm{P}_j-j\mathrm{P}_{j-1}$	$2/(2j+1)$
Tchébychev	$[-1,1]$	$(1-x^2)^{-1/2}$	$\mathrm{T}_{j+1}=2x\mathrm{T}_j-\mathrm{T}_{j-1}$	$\frac{1}{2}\pi(1+\delta_{j0})$
Hermite	$[-\infty,\infty]$	e^{-x^2}	$\mathrm{H}_{j+1}=2x\mathrm{H}_j-2j\mathrm{H}_{j-1}$	$\sqrt{\pi}2^{\mathrm{N}}\mathrm{N}!$
Laguerre	$[0,\infty]$	$x^\alpha\mathrm{e}^{-x}$	$(j+1)\mathrm{L}^\alpha_{j+1}=(-x+2j+\alpha+1)\mathrm{L}^\alpha_j$ $-(j+\alpha)\mathrm{L}^\alpha_{j-1}$	$\frac{\Gamma(\alpha+j+1)}{\mathrm{N}!}$

TABLE 10.1
Propriétés des polynômes orthogonaux les plus courants

prennent la valeur +1 à $x=1$. Cela ne fait pas de différence sur la position des racines, mais change les coefficients de la relation de récurrence (10.3). Ce changement de normalisation n'affecte en rien les propriétés générales que nous avons décrites plus haut, en particulier la relation (11.26). Les propriétés de base des polynômes les plus courants sont indiquées au tableau 10.1.

Problème 10.1 :
Démontrez les relations (10.4), en supposant la récurrence (10.3) et l'orthogonalité. La solution trouvée démontre alors que la relation de récurrence est correcte, ainsi que l'orthogonalité.

Solution

On multiplie premièrement la relation de récurrence par $\langle p_j|$ et $\langle p_{j-1}|$, en supposant l'orthogonalité. On trouve alors

$$0=\langle xp_j|p_j\rangle-a_j\langle p_j|p_j\rangle \quad \text{et} \quad 0=\langle xp_{j-1}|p_j\rangle-b_j^2\langle p_{j-1}|p_{j-1}\rangle$$

La première de ces relations est la première des équations (10.4). Dans la deuxième relation, on applique la relation de récurrence à $|p_j\rangle$:

$$\langle xp_{j-1}|p_j\rangle=\langle(p_j+a_{j-1}p_{j-1}+b_{j-1}^2p_{j-2})|p_j\rangle=\langle p_j|p_j\rangle$$

et donc on trouve la deuxième des équations (10.4). Il y a donc équivalence entre ces équations et l'orthogonalité entre p_j et les deux polynômes précédents. Il reste à confirmer l'orthogonalité avec les polynômes restants de la base. Par exemple avec p_{j-2} :

$$\langle p_{j-2}|p_{j+1}\rangle=\langle xp_{j-2}|p_j\rangle$$

car on suppose (par récurrence) que p_{j-2} est déjà orthogonal à p_j et p_{j-1}. mais en utilisant encore une fois la relation de récurrence, on trouve

$$\langle xp_{j-2}|=\langle p_{j-1}|+a_{j-2}\langle p_{j-2}|+b_{j-2}^2\langle p_{j-3}|$$

et tous les termes du membre de droite sont par hypothèse déjà orthogonaux à $|p_j\rangle$. On voit facilement comment le résultat se généralise immédiatement à p_{j-3}, p_{j-4}, etc.

A.1 Polynômes de Legendre

Les polynômes de Legendre correspondent au cas $a=-1$, $b=1$ et $w(x)=1$. En appliquant la relation de récurrence, on trouve

$$\begin{array}{lll} p_0(x)=1 & \langle p_0|p_0\rangle=2 & \langle x\,p_0|p_0\rangle=0 \\ p_1(x)=x & \langle p_1|p_1\rangle=\dfrac{2}{3} & \langle x\,p_1|p_1\rangle=0 \\ p_2(x)=x^2-\dfrac{1}{3} & \langle p_2|p_2\rangle=\dfrac{8}{45} & \langle x\,p_2|p_2\rangle=0 \\ p_3(x)=x^3-\dfrac{3}{5}x & \cdots & \cdots \end{array} \tag{10.5}$$

Les polynômes de Legendre proprement dits (notés $\mathrm{P}_n(x)$) ne sont pas unitaires, mais sont normalisés par la condition $\mathrm{P}_n(1)=1$. En multipliant les polynômes unitaires ci-dessus par la constante appropriée, on trouve alors

$$\mathrm{P}_0(x)=1 \quad , \quad \mathrm{P}_1(x)=x \quad , \quad \mathrm{P}_2(x)=\frac{3}{2}x^2-\frac{1}{2} \quad , \quad \mathrm{P}_3(x)=\frac{5}{2}x^3-\frac{3}{2} \quad , \quad \cdots \tag{10.6}$$

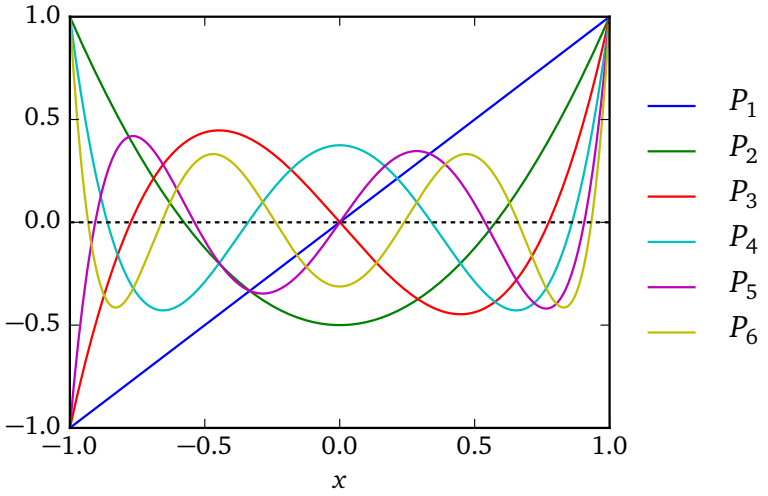


FIGURE 10.1
Les polynômes de Legendre P_1 à P_6.

A.2 Polynômes de Tchébychev

Les polynômes de Tchébychev correspondent au cas $a=-1$, $b=1$ et $w(x)=1/\sqrt{1-x^2}$. Les polynômes de Tchébychev sont en un sens plus simples que les polynômes de Legendre, en raison de leur relation avec les fonctions trigonométriques. On les définit habituellement comme

$$\mathrm{T}_j(x)=\cos(j\theta) \quad \text{où} \quad x=\cos\theta \tag{10.7}$$

La relation d'orthogonalité se comprend alors facilement via les fonctions trigonométriques :

$$\int_{-1}^{1} \mathrm{d}x \, \frac{\mathrm{T}_j(x)\mathrm{T}_k(x)}{\sqrt{1-x^2}} = \int_0^{\pi} \mathrm{d}\theta \, \cos(j\theta)\cos(k\theta) = \delta_{jk}\frac{\pi}{2}(1+\delta_{j0}) \tag{10.8}$$

La représentation d'une fonction sur la base des polynômes de Tchébychev s'apparente donc à une série de Fourier ; cependant la fonction n'est pas périodique en x, mais en θ ! Les polynômes de Tchébychev sont encore plus utiles que les polynômes de Legendre pour représenter une fonction définie sur un intervalle fini.

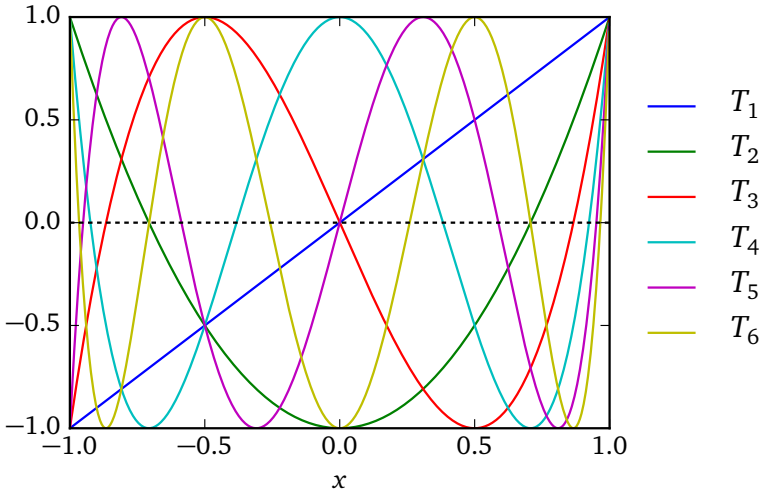


FIGURE 10.2
Les polynômes de Tchébychev T_1 à T_6.

Les premiers polynômes de Tchébychev sont :

$$\begin{aligned} \mathrm{T}_0(x) &= 1 & \mathrm{T}_1(x) &= x \\ \mathrm{T}_2(x) &= 2x^2-1 & \mathrm{T}_3(x) &= 4x^3-3x \\ \mathrm{T}_4(x) &= 8x^4-8x^2+1 & \mathrm{T}_5(x) &= 16x^5-20x^3+5x \end{aligned} \tag{10.9}$$

Problème 10.2 :
Montrez que si on définit les polynômes de Tchébychev par la relation (10.7), alors la relation de récurrence $\mathrm{T}_{j+1} = 2x\mathrm{T}_j - \mathrm{T}_{j-1}$ s'ensuit.

Solution

Cette relation, en fonction de θ, s'écrit

$$\cos((j+1)\theta) = 2\cos\theta\cos(j\theta) - \cos((j-1)\theta)$$

mais les relations d'addition des angles nous donnent

$$\cos((j+1)\theta) = \cos(j\theta)\cos\theta - \sin(j\theta)\sin\theta \qquad \cos((j-1)\theta) = \cos(j\theta)\cos\theta + \sin(j\theta)\sin\theta$$

En additionnant ces deux relations, on trouve bel et bien la relation de récurrence voulue.

A.3 Autres polynômes classiques

Polynômes d'Hermite

Les polynômes d'Hermite correspondent au cas $a=-\infty$, $b=\infty$ et $w(x)=\mathrm{e}^{-x^2}$. On les rencontre en mécanique quantique, dans l'expression des fonctions propres de l'oscillateur harmonique. En fonction de $u=\sqrt{m\omega}x$, où x est la coordonnée, celles-ci s'expriment ainsi :

$$\varphi_n(x)=\frac{1}{\sqrt{2^n n!}}\left(\frac{m\omega}{\pi}\right)^{1/4}\mathrm{e}^{-u^2/2}\mathrm{H}_n(u) \tag{10.10}$$

L'orthogonalité des fonctions propres correspond alors à celle des polynômes d'Hermite avec poids e^{-u^2}.

Polynômes de Laguerre

Les polynômes de Laguerre correspondent au cas $a=0$, $b=\infty$ et $w(x)=x^\alpha\,\mathrm{e}^{-x}$. Ils apparaissent en mécanique quantique dans le problème de l'atome d'hydrogène, en tant que parties radiales des fonctions propres du hamiltonien.

A.4 Théorème sur les racines

Théorème 10.1 Entrelacement des racines
Les racines de p_j sont réelles et celles de p_{j-1} s'intercalent entre celles de p_j.

Preuve Ce théorème se prouve par récurrence. Il est évident pour $j=0$ (aucune racine). Supposons qu'il est vrai pour p_j et montrons qu'il doit alors être vrai pour $j+1$. Soit x_i ($i=1,2,\ldots,j$) les j racines de p_j dans l'intervalle $[a,b]$. Appliquons la relation de récurrence (10.3) au point x_i :

$$p_{j+1}(x_i)=-b_j^2 p_{j-1}(x_i) \qquad\qquad p_j(x_i):=0 \tag{10.11}$$

Comme les racines de p_{j-1} sont par hypothèse entrelacées avec celles de p_j, le signe de $p_{j-1}(x_i)$ doit être l'opposé de celui de $p_{j-1}(x_{i+1})$ (voir figure 10.3), car p_{j-1} change de signe exactement une fois entre x_i et x_i+1. Comme b_j^2 est toujours positif, cela entraîne que p_{j+1} aussi change de signe entre ces deux valeurs, et qu'il a au moins une racine entre x_i et x_i+1. Il reste à montrer qu'en plus, p_{j+1} possède une racine dans l'intervalle $[x_j,\infty]$ et une autre dans $[-\infty,x_1]$, c'est-à-dire à l'extérieur des racines de p_j. Comme p_{j+1} a au plus $j+1$ racines, cela montrerait aussi qu'il y a exactement une racine de p_{j+1} entre x_i et x_{i+1}. Or comme p_{j-1} est unitaire et que sa dernière racine est plus petite que x_j, il doit être positif à x_j. Donc $p_{j+1}(x_j)<0$, alors que p_{j+1} est lui aussi unitaire ; donc il possède une racine plus élevée que x_j car $p_{j+1}(x)>0$ pour x suffisamment grand. Le même argument s'applique pour $x<x_1$, car p_{j+1} tend vers $\pm\infty$ quand $x\to-\infty$, selon que j est impair ou pair.

Théorème 10.2 Existence des racines dans l'intervalle d'orthogonalité
Le polynôme $p_j(x)$ a exactement j racines dans l'intervalle $[a,b]$.

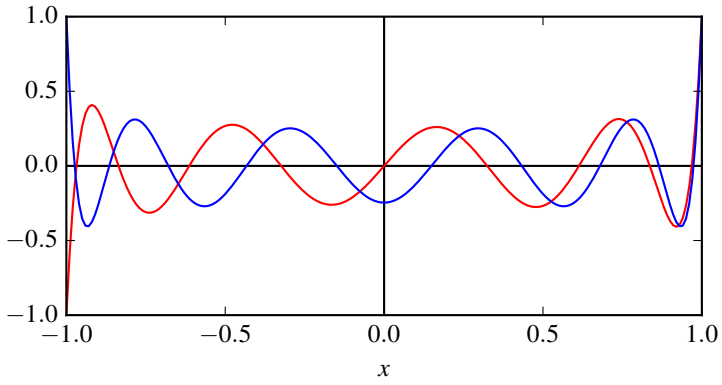


FIGURE 10.3
Polynômes unitaires de Legendre $p_{10}(x)$ (en bleu) et $p_9(x)$ (en rouge), avec leurs racines entrelacées.

Preuve Encore une fois, nous procédons par récurrence, comme dans le théorème d'entrelacement. Par rapport à la preuve précédente, il nous reste à prouver que les racines extrêmes de p_j sont comprise dans l'intervalle $[a,b]$, si celles de p_{j-1} le sont. À cette fin, considérons la fonction

$$s(x) = \prod_{x_i \in [a,b]} (x - x_i) \tag{10.12}$$

où le produit est restreint aux racines x_i de p_j qui sont dans l'intervalle $[a,b]$. La fonction $s(x)$ est égale à $p_j(x)$ si le théorème est vrai, car il s'agit alors simplement de la factorisation de p_j en fonction de toutes ses racines. Supposons au contraire que le théorème soit faux et montrons qu'on arrive à une contradiction. La fonction $s(x)$ est alors un polynôme de degré $m < j$. Ce polynôme peut alors s'exprimer comme une combinaison linéaire des $p_k(x)$, $(k = 1,2,\ldots,j-1)$. Il est donc orthogonal à p_j et

$$\int_a^b w(x)s(x)p_j(x) = 0 \tag{10.13}$$

Mais comme $s(x)$ s'annule exactement aux mêmes points que $p_j(x)$ dans l'intervalle, il possède toujours le même signe que $p_j(x)$ et donc l'intégrant est strictement positif dans l'intervalle et l'intégrale ne peut être nulle. Donc $s(x) = p_j(x)$.

A.5 Approximation d'une fonction par un polynôme

Supposons qu'on veuille représenter une fonction $\psi(x)$ de manière approchée par un polynôme de degré N dans l'intervalle $[a,b]$, de manière à minimiser l'écart quadratique entre la fonction et le polynôme, pondéré par le poids positif $w(x)$, c'est-à-dire de manière à minimiser la quantité suivante :

$$\chi^2 = \int_a^b \mathrm{d}x\, w(x)[\psi(x) - p(x)]^2 = \langle(\psi - p)|(\psi - p)\rangle \tag{10.14}$$

Ce polynôme $p(x)$ peut toujours être exprimé dans une base de polynômes orthogonaux dans l'intervalle $[a,b]$ sur le poids $w(x)$: $p(x) = \sum_{j=0}^{\mathrm{N}} a_j p_j(x)$. Il s'ensuit que

$$\chi^2 = \int_a^b \mathrm{d}x\, w(x) \left\{ \psi(x) - \sum_{j=0}^{\mathrm{N}} a_j p_j(x) \right\}^2 \tag{10.15}$$

On doit alors trouver les coefficients a_j qui minimisent cette expression. Il suffit pour cela d'en calculer la dérivée par rapport à a_j et de l'annuler :

$$\frac{\partial \chi^2}{\partial a_j} = 2\int_a^b \mathrm{d}x\, w(x)\left\{\psi(x) - \sum_{k=0}^{\mathrm{N}} a_k p_k(x)\right\} p_j(x) = 0 \tag{10.16}$$

Comme les polynômes sont orthogonaux, cette expression se simplifie en

$$\frac{\partial \chi^2}{\partial a_j} = 2\int_a^b \mathrm{d}x\, w(x)\psi(x)p_j(x) - 2a_j\langle p_j|p_j\rangle = 0 \tag{10.17}$$

ce qui entraîne que

$$a_j = \frac{1}{\langle p_j|p_j\rangle}\int_a^b \mathrm{d}x\, w(x)\psi(x)p_j(x) = \frac{\langle \psi|p_j\rangle}{\langle p_j|p_j\rangle} \tag{10.18}$$

Autrement dit, la meilleure approximation à une fonction par un polynôme, au sens de la minimisation de χ^2 défini ci-dessus, est une combinaison de polynômes orthogonaux dans l'intervalle $[a, b]$ dont les coefficients sont les produits scalaires de ψ avec les polynômes de base.

CHAPITRE 11

INTÉGRATION NUMÉRIQUE

A Formules élémentaires d'intégration d'une fonction

L'interpolation d'une fonction permet d'en définir les intégrales et les dérivées, en fonction de l'intégrale et des dérivées de l'interpolant. Considérons une fonction $\psi(x)$ qu'on désire intégrer sur un intervalle $[0,\ell]$. On peut définir une grille de N points $x_0,\ldots,x_{\mathrm{N}-1}$ et effectuer une interpolation linéaire entre ces points pour représenter la fonction à intégrer :

$$\int_0^\ell \mathrm{d}x\,\psi(x) \to \sum_{i=0}^{\mathrm{N}-2} \int_{x_i}^{x_{i+1}} \left\{ \frac{x-x_{i+1}}{x_i-x_{i+1}}\psi_i + \frac{x-x_i}{x_{i+1}-x_i}\psi_{i+1} \right\} = \sum_{i=0}^{\mathrm{N}-2} \frac{1}{2}(\psi_i+\psi_{i+1})(x_{i+1}-x_i) \tag{11.1}$$

Chaque terme du membre de droite est en fait la largeur d'un intervalle, multipliée par la moyenne des valeurs de la fonction aux extrémités de l'intervalle, comme illustré à la figure 11.1. Les colonnes rectangulaires peuvent être redécoupées pour former des trapèzes, d'où le nom *méthode des trapèzes* pour désigner cette méthode d'intégration élémentaire. En considérant comment chaque terme de cette somme se combine avec les termes voisins, on trouve

$$\int_0^\ell \mathrm{d}x\,\psi(x) \to \tfrac{1}{2}\left[\psi_0(x_1-x_0)+2\psi_1(x_2-x_0)+2\psi_2(x_3-x_1)+\cdots+\psi_{\mathrm{N}-1}(x_{\mathrm{N}-1}-x_{\mathrm{N}-2})\right] \tag{11.2}$$

Jusqu'ici les points de la grille ne sont pas nécessairement équidistants. Par contre, si la grille est uniforme ($x_{i+1}-x_i=a$), ceci devient simplement

$$\int_0^\ell \mathrm{d}x\,\psi(x) \to \tfrac{1}{2}a(\psi_0+\psi_{\mathrm{N}-1}) + a\sum_{i=1}^{\mathrm{N}-2}\psi_i \tag{11.3}$$

Notez encore une fois que les extrémités sont particulières.

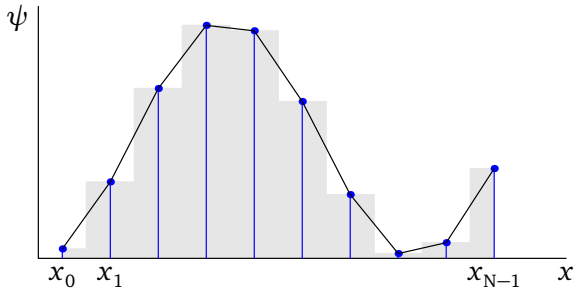


FIGURE 11.1
Illustration de la méthode des trapèzes. Notez que l'aire des rectangles ombragés est la même que l'aire limitée par les segments linéaires qui forment des trapèzes.

A.1 Erreur de troncature dans la formule des trapèzes

La formule des trapèzes se base sur l'intégrale suivante dans un intervalle élémentaire de largeur a :

$$\int_0^a \mathrm{d}x\,\psi(x) \approx \frac{1}{2}a\big[\psi(0)+\psi(a)\big] \tag{11.4}$$

Nous allons montrer que cette expression s'accompagne d'une erreur de troncature d'ordre $\mathcal{O}(a^3)$. Pour ce faire, considérons les deux développements limités suivants, l'un autour de $x=0$, l'autre autour de $x=a$:

$$\begin{aligned}\psi(x) &= \psi(0)+\psi'(0)x+\tfrac{1}{2}\psi''(0)x^2+\mathcal{O}(a^3)\\ \psi(x) &= \psi(a)+\psi'(a)(x-a)+\tfrac{1}{2}\psi''(a)(x-a)^2+\mathcal{O}(a^3)\end{aligned} \tag{11.5}$$

Les intégrales de ces deux développements sur $[0,a]$ donnent respectivement

$$\begin{aligned}\int_0^a \mathrm{d}x\,\psi(x) &= a\psi(0)+\tfrac{1}{2}a^2\psi'(0)+\tfrac{1}{6}a^3\psi''(0)+\mathcal{O}(a^4)\\ \int_0^a \mathrm{d}x\,\psi(x) &= a\psi(a)-\tfrac{1}{2}a^2\psi'(a)+\tfrac{1}{6}a^3\psi''(a)+\mathcal{O}(a^4)\end{aligned} \tag{11.6}$$

La moyenne de ces deux approximations est

$$\int_0^a \mathrm{d}x\,\psi(x) = \tfrac{1}{2}a\big[\psi(0)+\psi(a)\big]+\tfrac{1}{4}a^2\big[\psi'(0)-\psi'(a)\big]+\tfrac{1}{12}a^3\big[\psi''(0)+\psi''(a)\big]+\mathcal{O}(a^4) \tag{11.7}$$

Mais, d'après le théorème de la moyenne,

$$\big[\psi'(0)-\psi'(a)\big] = \psi''(\bar{x})a \tag{11.8}$$

où $\bar{x}$ est une valeur quelque part entre 0 et a. Donc finalement

$$\int_0^a \mathrm{d}x\,\psi(x) = \frac{1}{2}a\big[\psi(0)+\psi(a)\big]+\mathcal{O}(a^3) \tag{11.9}$$

En sommant les contributions des différents intervalles, on accumule les erreurs de troncature, commettant une erreur d'ordre $\mathcal{O}(\mathrm{N}a^3)=\mathcal{O}(a^2)=\mathcal{O}(1/\mathrm{N}^2)$ car $a=\ell/(\mathrm{N}-1)$.

A.2 Formule de Simpson

Procédons maintenant à un ordre supérieur d'approximation et faisons passer une parabole entre trois points de la grille (disons x_1, x_2 et x_3 pour simplifier). Le polynôme unique passant pas ces points est donné à l'équation (9.4). Supposons pour simplifier que la grille soit uniforme, de sorte que $x_3-x_2=x_2-x_1=a$. L'expression (9.4) devient

$$\mathrm{P}(x) = \psi_1+\frac{x}{2a}\big(3\psi_1-4\psi_2+\psi_3\big)+\frac{x^2}{2a^2}(\psi_1-2\psi_2+\psi_3) \tag{11.10}$$

L'intégrale de cette expression entre x_1 et x_3 est

$$\int_{x_1}^{x_3} \mathrm{d}x\,\mathrm{P}(x) = a\left(\frac{1}{3}\psi_1+\frac{4}{3}\psi_2+\frac{1}{3}\psi_3\right) \tag{11.11}$$

Il s'agit de la *règle de Simpson* pour un intervalle simple. Appliquons maintenant cette formule à une grille, en considérant les intervalles successifs $[x_0, x_2]$, $[x_2, x_4]$, etc. On trouve alors la *règle de Simpson étendue* :

$$\int_0^\ell \mathrm{d}x\, \psi(x) \approx a\left(\frac{1}{3}\psi_0 + \frac{4}{3}\psi_1 + \frac{2}{3}\psi_2 + \frac{4}{3}\psi_3 + \frac{2}{3}\psi_4 + \cdots + \frac{4}{3}\psi_{\mathrm{N}-2} + \frac{1}{3}\psi_{\mathrm{N}-1}\right) \tag{11.12}$$

(nous avons supposé que N est pair). On montre, de la même manière que pour la formule des trapèzes mais avec plus d'algèbre, que l'erreur commise par la règle de Simpson étendue se comporte comme $1/\mathrm{N}^4$.

Pour ceux qui pourraient croire que l'alternance des $\frac{2}{3}$ et des $\frac{4}{3}$ dans la règle de Simpson étendue cache une subtile compensation d'erreurs d'un terme à l'autre, signalons une autre formule d'intégration d'ordre $\mathcal{O}(1/\mathrm{N}^4)$: [1]

$$\int_0^\ell \mathrm{d}x\, \psi(x) \approx a\left(\frac{3}{8}\psi_0 + \frac{7}{6}\psi_1 + \frac{23}{24}\psi_2 + \psi_3 + \psi_4 + \cdots + \psi_{\mathrm{N}-4} + \frac{23}{24}\psi_{\mathrm{N}-3} + \frac{7}{6}\psi_{\mathrm{N}-2} + \frac{3}{8}\psi_{\mathrm{N}-1}\right) \tag{11.13}$$

Problème 11.1 :
Montrez que l'erreur de troncature de la formule de Simpson étendue (11.12) est $\mathcal{O}(1/\mathrm{N}^4)$. Utilisez la même méthode que pour la méthode des trapèzes, en poussant le développement à un ordre plus élevé.

B Quadratures gaussiennes

Les polynômes orthogonaux sont à la base d'une technique d'intégration très répandue : la quadrature gaussienne.

Une formule de quadrature pour une fonction $\psi(x)$ dans l'intervalle $[a, b]$ prend en général la forme suivante :

$$\int_a^b \mathrm{d}x\, \psi(x) \approx \sum_{i=1}^{\mathrm{N}} w_i \psi(x_i) \tag{11.14}$$

où les x_i sont les *abscisses* et les w_i les *poids*. Par exemple, la méthode des trapèzes (11.3) correspond à des abscisses équidistantes entre a et b (extrémités incluses) et des poids $w_i = (b-a)/(\mathrm{N}-1)$, sauf pour ceux des extrémités, qui en valent la moitié. Dans la méthode de Simpson (11.12) les abscisses sont les mêmes, mais les poids alternent entre deux valeurs dans l'intervalle et sont différents aux deux extrémités.

On peut considérablement augmenter l'efficacité d'une formule de quadrature si on permet aux abscisses d'être disposées de manière non uniforme. On dispose alors de 2N paramètres ajustables (les N abscisses et les N poids) qui peuvent être choisis de manière à ce que la formule de quadrature soit exacte pour tous les polynômes de degré $2\mathrm{N}-1$ ou moins (car un polynôme général de degré

1. Voir [PTVF07], Section 4.1.3.

$2\mathrm{N}-1$ comporte $2\mathrm{N}$ paramètres). On peut également choisir les paramètres de manière à ce que la formule soit exacte si $\psi(x)$ est une fonction positive $w(x)$ fois un polynôme de degré $2\mathrm{N}-1$ ou moins. C'est l'idée générale derrière les quadratures gaussiennes.

Supposons qu'on veuille intégrer une fonction $\psi(x) \coloneqq w(x)\tilde{\psi}(x)$ dans l'intervalle $[a,b]$. L'approximation (11.14) est optimale si les n points de grille x_i sont les racines du polynôme $p_\mathrm{N}(x)$ dans $[a,b]$, et les poids w_i la solution du système linéaire (11.15) ci-dessous. De plus, la formule de quadrature est alors exacte si $\tilde{y}$ est un polynôme de degré $2\mathrm{N}-1$ ou moins.

Théorème 11.1 quadratures gaussiennes
Soit x_i les N racines du polynôme orthogonal $p_\mathrm{N}(x)$, et w_i la solution du système linéaire suivant :

$$\begin{pmatrix} p_0(x_1) & p_0(x_2) & p_0(x_3) & \cdots & p_0(x_\mathrm{N}) \\ p_1(x_1) & p_1(x_2) & p_1(x_3) & \cdots & p_1(x_\mathrm{N}) \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ p_{\mathrm{N}-1}(x_1) & p_{\mathrm{N}-1}(x_2) & p_{\mathrm{N}-1}(x_3) & \cdots & p_{\mathrm{N}-1}(x_\mathrm{N}) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_\mathrm{N} \end{pmatrix} = \begin{pmatrix} \langle p_0|p_0\rangle \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{11.15}$$

Alors $w_i > 0$ et

$$\int_a^b w(x)p(x)\,\mathrm{d}x = \sum_{i=1}^{\mathrm{N}} w_i\, p(x_i) \tag{11.16}$$

pour tout polynôme $p(x)$ de degré $2\mathrm{N}-1$ ou moins.

Preuve La preuve suit la référence [SB02]. Considérons la matrice

$$\mathrm{A} \coloneqq \begin{pmatrix} p_0(x_1) & \cdots & p_0(x_\mathrm{N}) \\ \vdots & \ddots & \vdots \\ p_{\mathrm{N}-1}(x_1) & \cdots & p_{\mathrm{N}-1}(x_\mathrm{N}) \end{pmatrix} \tag{11.17}$$

Cette matrice est non singulière, car les polynômes sont linéairement indépendants. Autrement dit, si elle était singulière, il y aurait un vecteur $(c_0,\ldots,c_{\mathrm{N}-1})$ tel que $c^\mathrm{T}\mathrm{A}=0$ et donc le polynôme

$$q(x) \coloneqq \sum_{i=0}^{\mathrm{N}-1} c_i\, p_i(x) \tag{11.18}$$

qui est au plus de degré $\mathrm{N}-1$, aurait N racines distinctes $x_1, x_2, \ldots, x_\mathrm{N}$, ce qui est impossible, à moins qu'il soit identiquement nul, c'est-à-dire $c=0$. Donc A est une matrice non singulière. Par conséquent, les coefficients w_i existent et leur valeur est unique.

Considérons ensuite un polynôme $p(x)$ de degré $2\mathrm{N}-1$ au plus. Ce polynôme peut donc s'écrire comme $p(x)=p_\mathrm{N}(x)q(x)+r(x)$, où $q(x)$ et $r(x)$ sont des polynômes de degré $\mathrm{N}-1$ au plus. Ceci est vrai par construction de la division longue des polynômes (comme pour la division des nombres en représentation décimale, qui sont des polynômes en puissances de la base utilisée, à savoir 10). Le quotient $q(x)$ et le reste $r(x)$ peuvent donc être développés sur la base des p_j :

$$q(x)=\sum_{j=0}^{\mathrm{N}-1}\alpha_j p_j(x) \qquad\qquad r(x)=\sum_{j=0}^{\mathrm{N}-1}\beta_j p_j(x) \tag{11.19}$$

Comme $p_0(x) = 1$, il s'ensuit que

$$\int_a^b w(x)p(x) = \langle p_\mathrm{N}|q\rangle + \langle r|p_0\rangle = \beta_0\langle p_0|p_0\rangle \tag{11.20}$$

Notons que $\langle p_\mathrm{N}|q\rangle = 0$ car p_N est orthogonal à tous les polynômes p_j ($j < \mathrm{N}$) et que $q(x)$ est une combinaison de ces derniers. En plus, l'orthogonalité des p_j fait que le produit $\langle r|p_0\rangle$ se réduit à $\beta_0\langle p_0|p_0\rangle$. D'un autre côté, comme $p_\mathrm{N}(x_i) = 0$,

$$\sum_{i=1}^{\mathrm{N}} w_i p(x_i) = \sum_{i=1}^{\mathrm{N}} w_i \big(p_\mathrm{N}(x_i)q(x_i) + r(x_i)\big) = \sum_{i=1}^{\mathrm{N}} w_i r(x_i) = \sum_{i=1}^{\mathrm{N}}\sum_{j=0}^{\mathrm{N}-1} \beta_j w_i p_j(x_i) \tag{11.21}$$

Mais les poids w_i sont précisément choisis afin que

$$\sum_{i=1}^{\mathrm{N}} w_i p_j(x_i) = \delta_{j0}\langle p_0|p_0\rangle \tag{11.22}$$

Donc

$$\sum_{i=1}^{\mathrm{N}} w_i p(x_i) = \beta_0\langle p_0|p_0\rangle = \int_a^b w(x)p(x) \tag{11.23}$$

et l'équation (11.16) est satisfaite. Il reste à démontrer que les poids w_i sont positifs. À cet effet, considérons le polynôme de degré $2\mathrm{N}-2$

$$\bar{p}_j(x) \coloneqq \prod_{k=1,k\neq j}^{\mathrm{N}} (x - x_k)^2 \qquad j = 1,\ldots,\mathrm{N} \tag{11.24}$$

Ce polynôme est strictement positif, et donc l'intégrale suivante aussi :

$$0 < \int_a^b \mathrm{d}x\, w(x)\bar{p}_j(x) = \sum_{i=1}^{\mathrm{N}} w_i \bar{p}_j(x_i) = w_j \prod_{\substack{k=1\\k\neq j}}^{\mathrm{N}} (x_j - x_k)^2 \tag{11.25}$$

(seul le terme $i = j$ de la somme survit). Comme le produit est nécessairement positif, w_j l'est aussi.

Le choix de la classe de polynômes orthogonaux utilisés dépend du type de fonction qu'on veut intégrer. Si on sait que les fonctions d'intérêt sont bien représentées par des polynômes, alors on choisit $w(x) = 1$ et les polynômes correspondants sont les polynômes de Legendre. Par contre, si la fonction d'intérêt $\psi(x)$ a des singularités ou un comportement tel que $\tilde{\psi}(x) = \psi(x)/w(x)$ est doux (ou bien représenté par un polynôme), alors on choisit les polynômes qui sont orthogonaux par le poids $w(x)$.

On démontre l'expression explicite suivante pour les poids w_j :

$$w_j = \frac{\langle p_{\mathrm{N}-1}|p_{\mathrm{N}-1}\rangle}{p_{\mathrm{N}-1}(x_j)p'_\mathrm{N}(x_j)} \tag{11.26}$$

où $p'_\mathrm{N}(x)$ est la dérivée de $p_\mathrm{N}(x)$. Les poids w_i et les abscisses x_i sont tabulés pour les polynômes les plus communs, mais on peut également les calculer explicitement.

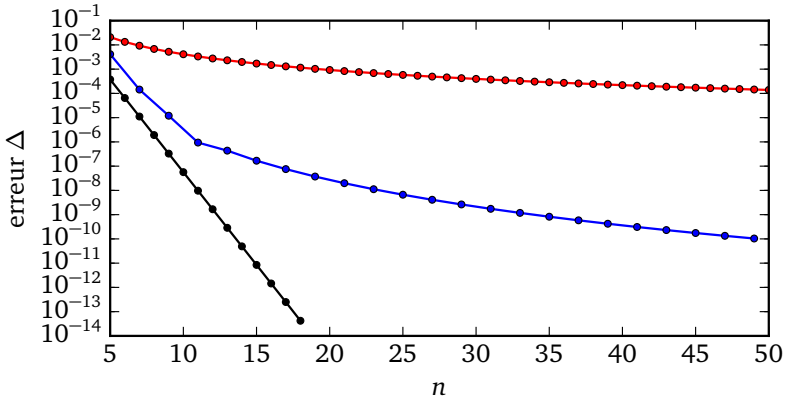


FIGURE 11.2
Graphique semi-logarithmique de l'erreur effectuée sur l'intégrale $\int_{-1}^{1} \mathrm{d}x\,(1+x)/(1+x^2)$ en fonction du nombre n d'évaluations de l'intégrant (le résultat exact est $\pi/2$ et donc l'erreur est connue exactement). La courbe supérieure provient de la méthode du trapèze; la courbe médiane de la méthode de Simpson et la courbe du bas de l'intégrale gaussienne avec polynômes de Legendre. Notez que dans ce dernier cas, l'erreur diminue exponentiellement avec n.

Le théorème de la quadrature gaussienne nous permet de définir le produit scalaire suivant :

$$\langle \psi | \psi' \rangle_g \coloneqq \sum_{i=1}^{\mathrm{N}} w_i \psi(x_i) \psi'(x_i) \tag{11.27}$$

En autant que le produit des fonctions impliquées soit un polynôme de degré $2\mathrm{N}-1$ ou moins, ce produit scalaire est identique au produit (10.1). On montre que la précision de la quadrature gaussienne augmente exponentiellement avec N. C'est là que réside en principe son avantage principal. Notre pratique sera donc d'utiliser la forme (11.27) du produit scalaire, en supposant qu'elle représente effectivement le produit (10.1), et nous ne ferons plus de distinction entre les deux du point de vue de la notation.

B.1 Quadratures de Gauss-Legendre

Les polynômes de Legendre sont utiles dans le cas d'un intervalle fini. En effet, tout intervalle fini $[a,b]$ peut être ramené à l'intervalle $[-1,1]$ par un changement de variable linéaire :

$$x = \frac{b+a}{2} + \frac{b-a}{2} u \tag{11.28}$$

et donc

$$\int_a^b \mathrm{d}x\ \psi(x) = \frac{b-a}{2} \int_{-1}^{1} \mathrm{d}u\ \psi(x(u)) \tag{11.29}$$

Les racines de $\mathrm{P}_\mathrm{N}(x)$ dans $[-1,1]$ doivent être trouvées numériquement, par la méthode de Newton par exemple (voir. la section A.5). Ceci est généralement vrai pour tous les polynômes orthogonaux (sauf les polynômes de Tchébychev pour lesquels une expression explicite est connue). Mais

TABLE 11.1
Racines et poids pour les premiers polynômes de Legendre

N	$\pm x_i$	w_i
2	0.5773502691896258	1.0
3	0	0.8888888888888889
	0.7745966692414834	0.5555555555555556
4	0.3399810435848563	0.6521451548625462
	0.8611363115940526	0.3478548451374539
5	0	0.5688888888888889
	0.5384693101056831	0.4786286704993665
	0.9061798459386640	0.2369268850561891
6	0.2386191860831969	0.4679139345726910
	0.6612093864662645	0.3607615730481386
	0.9324695142031520	0.1713244923791703

la propriété d'entrelacement des racines simplifie beaucoup cette recherche : on peut procéder par récurrence et on sait alors que chaque racine de $\mathrm{P}_n(x)$ se trouve entre deux racines successives de $\mathrm{P}_{n-1}(x)$ ou entre les racines extrêmes de $\mathrm{P}_{n-1}(x)$ et les extrémités de l'intervalle.

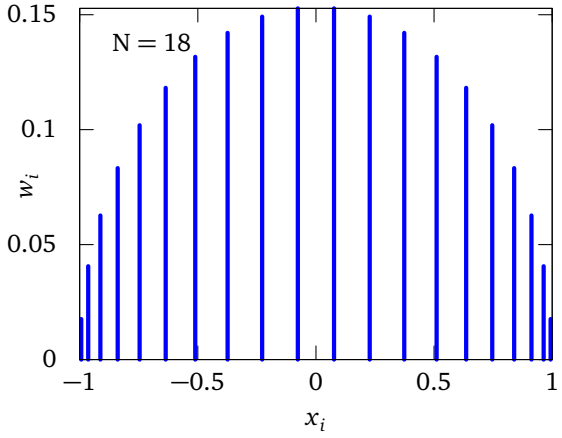


FIGURE 11.3
Les racines et poids du polynôme de Legendre $\mathrm{P}_{18}(x)$.

On utilise souvent les polynômes de Legendre dans des routines d'intégration numérique, où l'intervalle d'intégration est divisé en segments plus petits, de sorte qu'un petit nombre de points est requis pour calculer l'intégrale sur chaque segment. Les racines et poids des polynômes de Legendre pour de modestes valeurs de N sont alors requises ; elles sont données au tableau 11.1

B.2 Quadratures de Gauss-Tchébychev

On peut également définir une formule de quadrature basée sur les polynômes de Tchébychev. La relation (10.7) nous indique immédiatement où se trouvent les racines x_i de T_N :

$$x_i = \cos\left(\frac{\pi}{2}\frac{2i-1}{\mathrm{N}}\right) \tag{11.30}$$

De plus, les poids w_i correspondants sont tous égaux à π/N. Un façon de s'en convaincre est de contempler la formule intégrale (11.16) du point de vue de la variable angulaire θ :

$$\int_{-1}^{1} \frac{\psi(x)\,\mathrm{d}x}{\sqrt{1-x^2}} = \int_0^{\pi} \psi(\cos\theta)\,\mathrm{d}\theta \to \sum_{i=1}^{\mathrm{N}} \psi(\cos\theta_i) w_i \qquad (x_i = \cos\theta_i) \tag{11.31}$$

Les angles θ_i sont donnés par

$$\theta_i = \frac{\pi}{2}\frac{2i-1}{\mathrm{N}} \in \left\{ \frac{\pi}{2\mathrm{N}}, \frac{3\pi}{2\mathrm{N}}, \frac{5\pi}{2\mathrm{N}}, \ldots, \pi - \frac{\pi}{2\mathrm{N}} \right\} \tag{11.32}$$

Cet ensemble d'angles peut être doublé en y ajoutant $\pi - \theta_i$, sans que les valeurs de $\cos\theta$ soient affectées, car $\cos(\pi - \theta_i) = \cos\theta_i$. Après ce doublement, on se trouve en présence d'un ensemble d'angles équidistants le long du cercle unité, dans le but d'intégrer une fonction périodique en θ. Il n'y a donc pas de raison que les poids w_i soient différents d'un angle à l'autre, d'où la relation $w_i = \pi/\mathrm{N}$.

B.3 Quadratures de Gauss-Kronrod

Même si la quadrature de Gauss-Legendre converge rapidement en fonction du nombre de points, en pratique il ne s'agit pas d'une approximation contrôlée : elle ne fournit pas un estimation de l'erreur commise. De plus, on ne s'attend pas à ce qu'elle soit pratique pour des fonctions dont la comportement varie beaucoup d'une région à l'autre du domaine d'intégration.

La solution consiste à adopter une méthode adaptative contrôlée par une approximation d'ordre supérieur. En pratique, on doit stocker en mémoire l'ensemble des intervalles (ou régions) utilisés dans le calcul de l'intégrale, avec une estimation de l'erreur Δ_R commise dans chaque région R. On s'efforce alors de subdiviser les régions de manière à ce que l'erreur totale

$$\Delta = \left\{ \sum_{\mathrm{R}} \Delta_\mathrm{R}^2 \right\}^{1/2} \tag{11.33}$$

soit inférieure à une borne fixée à l'avance. À cette fin, on démarre avec un ensemble minimal de régions, toutes égales. On calcule pour chacune d'entre elles une estimation $\mathrm{I_R}$ de l'intégrale et l'erreur associée Δ_R. On sélectionne ensuite la région ayant l'erreur la plus grande et on la divise en deux, remplaçant sa contribution à l'intégrale totale I par celles des deux sous-régions. Ensuite on recommence avec la nouvelle région ayant la plus grande erreur, et ainsi de suite jusqu'à ce que l'erreur totale soit inférieure à la précision requise, ou qu'un nombre maximum de régions ait été atteint.

Reste le problème du calcul de l'erreur dans chaque région. Pour ce faire, on pourrait, dans une région donnée, utiliser une quadrature de Gauss-Legendre à N points et une deuxième quadrature à M points ($\mathrm{M} > \mathrm{N}$), obtenant ainsi deux estimations différentes $\mathrm{G_N}$ et $\mathrm{G_M}$ de l'intégrale dans cette région, ainsi qu'une estimation $\Delta_\mathrm{R} \sim |\mathrm{G_M} - \mathrm{G_N}|$ de l'erreur. Malheureusement, les points calculés à l'ordre M ne sont pas les mêmes qu'à l'ordre N (sauf $u = 0$ si M et N sont impairs), et donc cette façon de faire est coûteuse en évaluations de l'intégrant.

La façon la plus pratique de procéder est en fait la *méthode de Gauss-Kronrod*, dans laquelle on ajoute aux N points de la quadrature de Gauss-Legendre $\mathrm{N}+1$ points supplémentaires. Au total, les

$\pm x_i$	w_i
Noeuds de Gauss-Legendre	
±0.94910 79123 42759	0.12948 49661 68870
±0.74153 11855 99394	0.27970 53914 89277
±0.40584 51513 77397	0.38183 00505 05119
0.00000 00000 00000	0.41795 91836 73469
Noeuds de Kronrod	
±0.99145 53711 20813	0.02293 53220 10529
±0.94910 79123 42759	0.06309 20926 29979 *
±0.86486 44233 59769	0.10479 00103 22250
±0.74153 11855 99394	0.14065 32597 15525 *
±0.58608 72354 67691	0.16900 47266 39267
±0.40584 51513 77397	0.19035 05780 64785 *
±0.20778 49550 07898	0.20443 29400 75298
0.00000 00000 00000	0.20948 21410 84728 *

TABLE 11.2
Racines et poids utilisés dans la règle de Gauss-Kronrod G_7K_{15}. Notez que 7 des points sont communs à G_7 et K_{15} (marqués par *), mais que les poids sont différents.

$2N+1$ points, avec des poids appropriés, permettent de procéder à une *quadrature de Kronrod*. Par contre, les poids w_i ne sont pas les mêmes pour les deux quadratures. La quadrature de Kronrod compte alors $N+1$ points ajustables et $2N+1$ poids ajustables, pour un total de $3N+2$ paramètres, ce qui lui permet d'évaluer exactement l'intégrale d'un polynôme de degré $3N+1$.

Le tableau 11.2 énumère les points et les poids associés à la méthode dite G_7K_{15}, correspondant à $N=7$. On obtient de cette manières deux estimations de l'intégrale : G_7, calculée à l'aide des 7 points et poids de Gauss-Legendre, et K_{15}, calculée à l'aide des 15 points et poids de Kronrod, mais qui ne nécessite que 8 évaluations supplémentaires de l'intégrant. On montre que l'estimation de l'erreur est alors donnée par

$$\Delta_R = 200|K_{15} - G_7|^{3/2} \tag{11.34}$$

La méthode décrite ci-dessus est très utilisée, notamment dans beaucoup de bibliothèques, dont QUADPACK, qui à son tour est utilisée dans GSL et dans NumPy.

C Approximation de Tchébychev

Les avantages des polynômes de Tchébychev sont multiples : Non seulement les positions x_i des racines de T_N sont facilement calculables et les poids sont triviaux, mais $T_j(x)$ est toujours compris entre -1 et 1, ce qui est intéressant sur le plan du contrôle de l'approximation.

Supposons en effet qu'on effectue un développement limité d'une fonction $\psi(x)$ sur la base des polynômes de Tchébychev :

$$\psi(x) \approx \sum_{j=0}^{N-1} \psi_j T_j(x) \tag{11.35}$$

En vertu de la relation d'orthogonalité, les coefficients ψ_j sont donnés par

$$\begin{aligned}\psi_j &= \frac{1}{\gamma_j}\int_{-1}^{1}\frac{\mathrm{d}x}{\sqrt{1-x^2}}\psi(x)\mathrm{T}_j(x) = \frac{2}{\mathrm{N}(1+\delta_{j0})}\sum_{i=1}^{\mathrm{N}}\psi(x_i)\mathrm{T}_j(x_i)\\ &= \frac{2}{\mathrm{N}(1+\delta_{j0})}\sum_{i=1}^{\mathrm{N}}\psi\left(\cos\left(\frac{\pi(i+1/2)}{\mathrm{N}}\right)\right)\cos\left(\frac{\pi j(i+1/2)}{\mathrm{N}}\right)\end{aligned} \tag{11.36}$$

En supposant que ce développement limité soit quasi-exact, en raison de la valeur élevée de N, on peut ensuite le tronquer en amputant les termes plus élevés de la somme ($j > m$), tout en gardant la même valeur de N. Alors le premier terme négligé dans ce développement, $\psi_{m+1}\mathrm{T}_{m+1}(x)$, sera partout borné par ψ_{m+1}, car $-1 \leq \mathrm{T}_{m+1}(x) \leq 1$. L'erreur est donc contrôlée par le coefficient ψ_{m+1} dans tout l'intervalle $[-1,1]$. En somme, il est simple, en se fixant une précision voulue ϵ, de trouver une approximation polynomiale de degré m qui soit valable en deçà de cette précision, si on tronque le développement quand $|\psi_{m+1}| < \epsilon$.

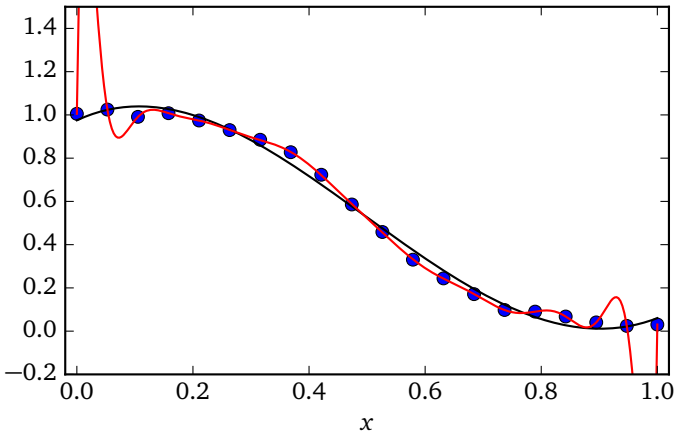


FIGURE 11.4
Approximant de Tchébychev pour un ensemble de 20 données bruitées. Le polynôme interpolant exact de degré 19 est indiqué en rouge, et souffre du *phénomène de Runge*, soit des oscillations violentes près des bords. L'approximant de degré 4 $p^{(4)}(x)$ (en noir) est beaucoup plus doux.

Cette approximation peut être appliquée à un ensemble de N données $\{y_k\}$ (par exemple des données expérimentales) dont les abscisses sont notées u_k. Ces N points définissent de manière unique un polynôme interpolant $p(x)$ de degré $\mathrm{N}-1$ qui passe par tous les points : c'est le polynôme donné par la formule de Lagrange (9.1) :

$$p(x) = \sum_{j=1}^{\mathrm{N}}\left[y_j\prod_{\substack{i=1\\(i\neq j)}}^{\mathrm{N}}\frac{x-u_i}{u_j-u_i}\right] \tag{11.37}$$

Or, ce polynôme admet un développement unique en fonction des polynômes de Tchébychev :

$$p(x) = \sum_{j=0}^{\mathrm{N}-1}\psi_j\mathrm{T}_j(x) \qquad \text{où} \qquad \psi_j = \frac{2}{\mathrm{N}(1+\delta_{j0})}\sum_{k=1}^{\mathrm{N}}p(x_k)\mathrm{T}_j(x_k) \tag{11.38}$$

Notons la différence entre les abscisses u_k des points de données et les racines x_k de $\mathrm{T}_{\mathrm{N}}(x)$.

On peut ensuite définir l'approximant suivant ($x = \cos\theta$) :

$$p^{(m)}(x) = \sum_{j=0}^{m}\psi_j\mathrm{T}_j(x) \tag{11.39}$$

Le polynôme $p(x)$ est le polynôme de degré $\mathrm{N}-1$ qui passe par tous les points y_k. On remarque que l'ensemble des points est relativement bien représenté par un approximant $f^{(m)}(x)$ de degré bien moindre, comme l'illustre la figure 11.4. La différence entre les approximants $f^{(m)}(x)$ et $f^{(m+1)}(x)$ est toujours inférieure, en valeur absolue, au coefficient ψ_{m+1}.

CHAPITRE 12

PROBLÈMES AUX LIMITES EN DIMENSION 1

Les problèmes de la physique classique et beaucoup de ses applications en génie peuvent être formulés via des équations différentielles. Souvent ces équations différentielles n'impliquent que des coordonnées spatiales, c'est-à-dire sont indépendantes du temps. Dans ce cas, leur solution est la plupart du temps contrainte par des conditions aux limites : c'est la valeur de la fonction recherchée, ou ses dérivées, sur la frontière du domaine qui détermine la solution. On parle alors de *problèmes aux limites*. Nous allons nous concentrer sur deux méthodes de solution des problèmes aux limites : la méthode des éléments finis, et les méthodes spectrales.

Les méthodes décrites dans ce chapitre pourront ensuite être appliquées au cas d'équations différentielles aux dérivées partielles qui dépendent du temps : la dépendance spatiale des fonctions impliquées étant alors représentée par éléments finis ou par représentation spectrale, et la dépendance temporelle faisant l'objet d'un traitement différent, purement séquentiel, comme ce qui a été fait au chapitre 3.

A Méthode du tir

Dans un *problème aux limites*, on doit résoudre une équation différentielle du deuxième ordre dont la solution est fixée non pas par des conditions initiales (c'est-à-dire la valeur de la fonction et de sa dérivée en un point), mais par les valeurs de la fonction (ou les valeurs de sa dérivée) aux extrémités de l'intervalle d'intérêt $[a,b]$; autrement dit, on impose les conditions $\psi(a)=\psi_a$ et $\psi(b)=\psi_b$.

Pour une équation différentielle générale du deuxième ordre dont les paramètres sont fixes, la méthode du tir consiste à résoudre un problème aux valeurs initiales à partir de $x=a$, et d'ajuster la valeur de la dérivée à $x=a$ jusqu'à ce que la condition $\psi(b)=\psi_b$ soit satisfaite. Autrement dit, on pose $\psi(a)=\psi_a$ et $\psi'(a)=\xi$, où ξ est une valeur d'essai, et on résout l'équation par les méthodes habituelles appliquées aux problèmes aux valeurs initiales. On obtient alors une solution $\psi(x|\xi)$ dont la valeur à $x=b$ ne respecte pas a priori la condition $\psi(b|\xi)=\psi_b$. On traite alors la relation $\psi(b|\xi)=\psi_b$ comme une équation pour ξ qu'il faut résoudre numériquement, à l'aide d'un algorithme de recherche de racines.

On peut aussi traiter de cette manière une équation aux valeurs propres, par exemple l'équation de Helmholtz :

$$\psi''+k^2\psi=0 \qquad \psi(0)=\psi(1)=0 \tag{12.1}$$

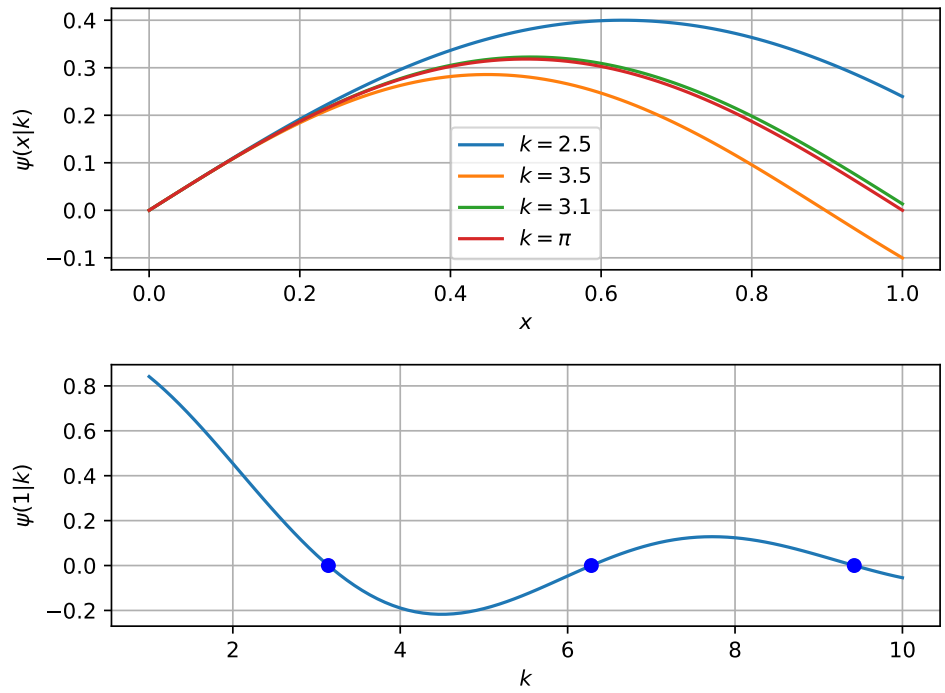


FIGURE 12.1
En haut : trois essais successifs visant à satisfaire la condition $\psi(1|k) = 0$. En bas, $\psi(1|k)$ en fonction de k, illustrant la position des trois premières solutions (π, 2π et 3π).

Cette équation n'a de solution que pour certaines valeurs de k à déterminer. Comme l'équation est homogène et linéaire, on peut multiplier la solution par un nombre quelconque et le résultat est encore une solution. Donc la valeur de la dérivée $\psi'(a)$ n'a pas d'impact sur la solution, à une constante multiplicative près, pourvu que $\psi'(a) \neq 0$ Par contre, la valeur propre k est traitée ici comme un paramètre qu'on doit ajuster. La solution $\psi(x|k)$ se trouve par les méthodes habituelles pour résoudre les problèmes aux valeurs initiales. On pose $\psi(0|k) = 0$ et $\psi'(0|k) = 1$ et on obtient une solution $\psi(1|k)$. On recherche alors les racines en k de $\psi(1|k)$, qui sont en général multiples (voir fig. 12.1).

L'efficacité de la méthode du tir dépend beaucoup du problème étudié. Si le domaine est impropre (c'est-à-dire infini ou semi-infini) et que les solutions et leurs dérivées tendent vers zéro à l'infini, il faut tout de même choisir un domaine fini et le choix des conditions aux limites peut être numériquement délicat. De plus, un problème aux valeurs propres dont les valeurs propres sont presque dégénérées sera très difficile à résoudre de cette manière. La méthode des éléments finis sera toujours plus stable et sûre.

B Base de fonctions tentes

La méthode des *éléments finis* repose sur le développement d'une fonction $\psi(x)$ sur une base de fonctions localisées. Elle est très utilisée dans la solution des équations aux dérivées partielles, et se combine habituellement à une discrétisation de l'espace en fonction de simplexes (ou de triangles en deux dimensions).

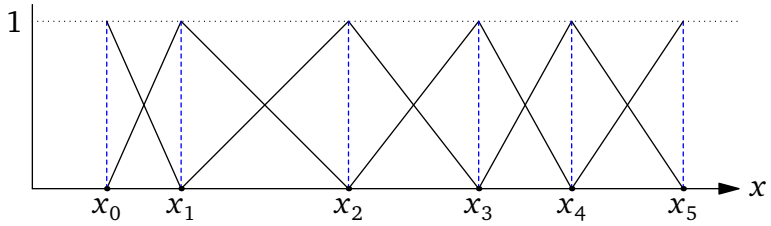


FIGURE 12.2
Fonctions-tentes en dimension 1, avec conditions aux limites ouvertes.

Commençons par le cas unidimensionnel. Définissons, sur le segment $[0,\ell]$, en se basant sur une grille de points $\{x_i\}$ qui n'est pas nécessairement régulière, un ensemble de fonctions $u_i(x)$ définies comme suit :

$$u_i(x) = \begin{cases} 0 & \text{si} \quad x > x_{i+1} \quad \text{ou} \quad x < x_{i-1} \\ \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{si} \quad x_{i-1} \leq x \leq x_i \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}} & \text{si} \quad x_i < x \leq x_{i+1} \end{cases} \tag{12.2}$$

Ces fonctions linéaires par morceaux sont appelées *fonctions tentes* en raison de leur forme. Voir la figure 12.2 pour un exemple. Le cas des extrémités ($i = 0$ et $i = \mathrm{N}-1$) est particulier si des conditions aux limites ouvertes sont utilisées. Dans ce cas, la définition est la même, à la différence que la fonction s'annule en dehors de l'intervalle $[0,\ell]$

Nous allons fréquemment utiliser la notation de Dirac dans cette section et les suivantes. La fonction $u_i(x)$ correspond à alors au produit bilinéaire $\langle x|u_i\rangle$ du vecteur abstrait $|u_i\rangle$ représentant la fonction u_i, et de la fonction propre de la position au point x ; l'utilisation de cette notation sera tout à fait semblable à ce qui est pratiqué en mécanique quantique, à la différence que nous utiliserons un espace de fonctions sur les réels et non sur les complexes. Le produit bilinéaire sera, à moins d'avis contraire, défini par l'intégrale

$$\langle f|g\rangle = \int_0^\ell \mathrm{d}x\, f(x)g(x) \tag{12.3}$$

La propriété fondamentale des fonctions tentes est qu'elles sont nulles en dehors du domaine $[x_{i-1}, x_{i+1}]$ et égales à l'unité à $x = x_i$. Ces propriétés entraînent premièrement que

$$u_i(x_j) = \delta_{ij} \quad \text{ou, en notation de Dirac,} \quad \langle x_j|u_i\rangle = \delta_{ij} \tag{12.4}$$

et deuxièmement que

$$\mathrm{M}_{ij} \coloneqq \langle u_i|u_j\rangle = \int_0^\ell \mathrm{d}x\, u_i(x)u_j(x) \tag{12.5}$$

n'est non nul que pour les sites i et j qui sont des voisins immédiats. La matrice M_{ij} est appelée *matrice de masse*. Comme $\mathrm{M}_{ij} \neq \delta_{ij}$, les fonctions tentes ne forment pas une base orthonormée.

L'ensemble des fonctions $u_i(x)$ génère un sous-espace $\mathcal{V}_u$ de l'espace des fonctions définies dans l'intervalle $[0,\ell]$. Toute fonction appartenant à ce sous-espace admet par définition le développement suivant :

$$\psi(x) = \sum_i \psi_i\, u_i(x) \quad \text{ou, en notation de Dirac,} \quad |\psi\rangle = \sum_i \psi_i|u_i\rangle \tag{12.6}$$

Comme les fonctions tentes ont la propriété $u_i(x_j) = \delta_{ij}$, on constate immédiatement que

$$\psi_i = \psi(x_i) \tag{12.7}$$

L'approximation que nous ferons dans la résolution de problèmes aux limites ou d'équations différentielles aux dérivées partielles est que les solutions à ces équations appartiendront au sous-espace $\mathcal{V}_u$. Autrement dit, nous ne procèderons pas à une approximation de l'équation différentielle elle-même – par exemple en remplaçant les dérivées continues par des dérivées discrètes – mais nous allons restreindre l'espace de fonctions dans lequel nous chercherons des solutions. Toute combinaison linéaire des fonctions tentes étant linéaire par morceaux, cette approximation revient à remplacer une fonction par une interpolation linéaire basée sur une grille de points. Ceci constitue donc l'approximation de base de la méthode des éléments finis.

Dans le sous-espace $\mathcal{V}_u$, le produit bilinéaire s'exprime comme suit :

$$\langle\psi|\psi'\rangle = \sum_{i,j}\langle u_i|u_j\rangle\psi_i\psi'_j = \psi^{\mathrm{T}}\mathrm{M}\psi' \tag{12.8}$$

où ψ^{T} est le transposé du vecteur ψ. Ce produit scalaire étant défini positif, la matrice M est, elle aussi, définie positive.

Considérons ensuite un opérateur différentiel $\mathscr{L}$, comme ceux qui apparaissent dans une équation différentielle. Par exemple, $\mathscr{L}$ pourrait être une combinaison de dérivées :

$$\mathscr{L} = u(x) + v(x)\partial_x + w(x)\partial_x^2 \tag{12.9}$$

L'action de l'opérateur $\mathscr{L}$ n'est pas fermée dans $\mathscr{V}_u$, c'est-à-dire que si $|\psi\rangle \in \mathscr{V}_u$, la fonction $\mathscr{L}|\psi\rangle$ n'appartient pas complètement à $\mathscr{V}_u$, mais possède un résidu $|\perp\rangle$ à l'extérieur de $\mathscr{V}_u$. Que veut-on dire exactement par là? Cette affirmation n'a de sens, en fait, que via le produit scalaire : le résidu $|\perp\rangle$ doit être orthogonal à $\mathscr{V}_u$: $\langle u_i|\perp\rangle = 0$. On peut donc écrire, pour chaque fonction tente,

$$\mathscr{L}|u_i\rangle = \sum_j \mathrm{L}^c_{ji}|u_j\rangle + |\perp\rangle \qquad (\langle u_k|\perp\rangle = 0) \tag{12.10}$$

où la matrice L^c définit l'action de $\mathscr{L}$ sur les fonctions tentes. En particulier, sur une fonction quelconque de $\mathscr{V}_u$, on a

$$\mathscr{L}|\psi\rangle = \sum_i \psi_i \mathscr{L}|u_i\rangle = \sum_{i,j} \psi_i \mathrm{L}^c_{ji}|u_j\rangle + |\perp\rangle \coloneqq \sum_j \psi'_j|u_j\rangle + |\perp\rangle \tag{12.11}$$

(notons que $|\perp\rangle$ ne désigne pas une fonction en particulier, mais toute fonction orthogonale à $\mathscr{V}_u$). Donc l'image de ψ par $\mathscr{L}$ est $\psi' = \mathrm{L}^c\psi$ (en notation vectorielle) lorsque projeté sur $\mathscr{V}_u$.

Définissons maintenant la matrice L ainsi :

$$\mathrm{L}_{ki} \coloneqq \langle u_k|\mathscr{L}|u_i\rangle = \sum_j \mathrm{L}^c_{ji}\langle u_k|u_j\rangle = \sum_j \mathrm{M}_{kj}\mathrm{L}^c_{ji} = (\mathrm{ML}^c)_{ki} \quad \text{ou} \quad \boxed{\mathrm{L} = \mathrm{ML}^c} \tag{12.12}$$

La distinction entre les matrices L et L^c provient du fait que les fonctions de base $|u_i\rangle$ ne forment pas une base orthonormée.

Problème 12.1 : Matrice de masse (dimension 1)
Montrez que

$$\langle u_i|u_j\rangle = \begin{cases} \frac{1}{6}|x_i - x_j| & \text{si} \quad |i-j| = 1 \\ \frac{1}{3}|x_{i+1} - x_{i-1}| & \text{si} \quad i = j,\ i \neq 0,\ i \neq \mathrm{N}-1 \end{cases} \quad \text{et} \quad \begin{array}{c} \langle u_0|u_0\rangle = \frac{1}{3}|x_1 - x_0| \\ \langle u_{\mathrm{N}-1}|u_{\mathrm{N}-1}\rangle = \frac{1}{3}|x_{\mathrm{N}-1} - x_{\mathrm{N}-2}| \end{array} \tag{12.13}$$

où les fonction $u_i(x)$ sont définies en (12.2).

C Méthode de Galerkin et méthode de collocation

Appliquons la représentation en éléments finis à la solution d'un problème aux limites, c'est-à-dire d'une équation différentielle pour la fonction $\psi(x)$, avec des conditions précises sur la valeur de $\psi(x)$ ou de ses dérivées à $x=0$ et $x=\ell$.

Nous allons écrire l'équation différentielle sous la forme générale suivante :

$$\mathscr{L}|\psi\rangle = |\rho\rangle \tag{12.14}$$

où $\mathscr{L}$ est un opérateur différentiel, qu'on peut supposer linéaire pour le moment, et $|\rho\rangle$ une fonction déterminée, souvent appelée *fonction de charge* ou *vecteur de charge*. Par exemple, dans le cas de l'équation de Helmholtz avec des sources émettrices distribuées selon la densité $\rho(x)$,

$$\psi''(x)+k^2\psi(x)=\rho(x) \quad \text{et donc} \quad \mathscr{L}=\frac{\mathrm{d}^2}{\mathrm{d}x^2}+k^2 \tag{12.15}$$

Nous cherchons une solution approchée de l'équation (12.14) sous la forme de valeurs $\{\psi_i\}$ définies sur la grille. Nous allons en fait considérer une *forme faible* de l'équation différentielle :

$$\langle w_j|\mathscr{L}|\psi\rangle = \int_0^\ell \mathrm{d}x\; w_j(x)\mathscr{L}\psi = \langle w_j|\rho\rangle \tag{12.16}$$

où $w_j(x)$ est une fonction ou une collection de fonctions qu'il faut spécifier et qui définissent précisément la forme faible utilisée. On qualifie cette forme de *faible*, parce que toute solution à l'équation (12.14) est nécessairement une solution à l'équation (12.16), alors que l'inverse n'est pas vrai.

On considère généralement les deux approches suivantes :

1. La méthode de **collocation** : On suppose alors que $|w_i\rangle = |x_i\rangle$, c'est-à-dire $w_i(x)=\delta(x-x_i)$. Ceci équivaut à demander que l'équation différentielle (12.14) soit respectée exactement sur la grille et mène à la relation suivante :

   $$\langle x_i|\mathscr{L}|\psi\rangle = \langle x_i|\rho\rangle = \rho_i \tag{12.17}$$

   où

   $$\langle x_i|\mathscr{L}|\psi\rangle = \langle x_i|\Big(\mathrm{L}^c_{kj}\psi_j|u_k\rangle+|\perp\rangle\Big) = \mathrm{L}^c_{ij}\psi_j+\langle x_i|\perp\rangle \tag{12.18}$$

   En négligeant le résidu $|\perp\rangle$, nous avons donc la relation matricielle

   $$\boxed{\mathrm{L}^c\psi=\rho} \tag{12.19}$$

2. La méthode de **Galerkin** : On prend plutôt $|w_i\rangle = |u_i\rangle$, ce qui revient à demander que l'équation différentielle soit respectée en moyenne sur le domaine de chaque fonction tente. Il s'ensuit que

   $$\langle u_i|\mathscr{L}|\psi\rangle = \langle u_i|\rho\rangle \tag{12.20}$$

   où

   $$\langle u_i|\mathscr{L}|\psi\rangle = \langle u_i|\Big(\mathrm{L}^c_{kj}\psi_j|u_k\rangle+|\perp\rangle\Big) = \mathrm{M}_{ik}\mathrm{L}^c_{kj}\psi_j = \mathrm{L}_{ij}\psi_j \tag{12.21}$$

On obtient donc la relation

$$(\mathrm{L}\psi)_i = \langle u_i|\rho\rangle = \langle u_i|\left(\sum_j \rho_j|u_j\rangle + |\perp\rangle\right) = \sum_j \mathrm{M}_{ij}\rho_j \quad \text{ou encore} \quad \boxed{\mathrm{L}\psi = \mathrm{M}\rho} \tag{12.22}$$

La subtilité ici est que ρ_i ne représente pas exactement la valeur $\rho(x_i)$, mais plutôt la valeur à x_i de la projection sur $\mathscr{V}_u$ de la fonction ρ.

Comme $\mathrm{L} = \mathrm{ML}^c$, les deux méthodes semblent superficiellement équivalentes. Cependant une approximation différente a été faite dans chacune avant d'arriver aux équations matricielles équivalentes $\mathrm{L}^c\psi = \rho$ et $\mathrm{L}\psi = \mathrm{M}\rho$. Dans la méthode de collocation, nous avons traité $\rho(x)$ exactement, mais négligé le résidu $|\perp\rangle$ résultant de l'application de $\mathscr{L}$ sur les fonctions tentes. Dans la méthode de Galerkin, ce résidu disparaît de lui-même, mais par contre seule la projection de $\rho(x)$ sur l'espace $\mathscr{V}_u$ est prise en compte.

C.1 Imposition des conditions aux limites de Dirichlet

La discussion qui précède a passé sous silence la question des conditions aux limites. Elle est certainement valable lorsqu'on impose des conditions aux limites périodiques, mais doit être raffinée dans les autres cas. Nous allons supposer ici qu'on impose à la fonction $\psi(x)$ des valeurs particulières à $x = 0$ et $x = \ell$ (conditions aux limites de type *Dirichlet*).

Supposons qu'on réordonne les indices vectoriels (et matriciels) de manière à ce que les indices associés à la frontière F apparaissent avant les indices associés à l'intérieur I. Un vecteur ψ est alors composé de deux parties :

$$\psi = \begin{pmatrix} \psi^{\mathrm{F}} \\ \psi^{\mathrm{I}} \end{pmatrix} \tag{12.23}$$

où ψ^{F} est un vecteur à 2 composantes (pour les deux points à la frontière en dimension 1) et ψ^{I} un vecteur à $\mathrm{N}-2$ composantes, pour les points intérieurs de l'intervalle. Une matrice L serait de même décomposée comme suit :

$$\mathrm{L} = \begin{pmatrix} \mathrm{L}^{\mathrm{F}} & \mathrm{L}^{\mathrm{FI}} \\ \mathrm{L}^{\mathrm{IF}} & \mathrm{L}^{\mathrm{I}} \end{pmatrix} \tag{12.24}$$

où L^{I} est une matrice d'ordre $\mathrm{N}-2$ décrivant l'action de l'opérateur L sur les points intérieurs, L^{IF} est une matrice $(\mathrm{N}-2)\times 2$ décrivant l'effet des points intérieurs via L sur les 2 extrémités, et ainsi de suite. L'équation différentielle ne devrait pas être imposée sur les points situés à la frontière du domaine : ce sont des conditions aux limites qui sont imposées en lieu et place. Donc, en appliquant l'équation différentielle aux points intérieurs seulement, on trouve l'équation matricielle suivante :

$$\mathrm{L}^{\mathrm{I}}\psi^{\mathrm{I}} + \mathrm{L}^{\mathrm{IF}}\psi^{\mathrm{F}} = \mathrm{M}^{\mathrm{I}}\rho^{\mathrm{I}} + \mathrm{M}^{\mathrm{IF}}\rho^{\mathrm{F}} \tag{12.25}$$

La solution se trouve en solutionnant un problème linéaire de la forme $\mathrm{L}^{\mathrm{I}}\psi^{\mathrm{I}} = b$, où $b = \mathrm{M}^{\mathrm{I}}\rho^{\mathrm{I}} + \mathrm{M}^{\mathrm{IF}}\rho^{\mathrm{F}} - \mathrm{L}^{\mathrm{IF}}\psi^{\mathrm{F}}$ est connu. Ceci suppose bien sûr que la matrice intérieure L^{I} est non singulière.

Si les conditions aux limites sont homogènes (c'est-à-dire si $\psi^{\mathrm{F}} = 0$) et que l'équation différentielle aussi est homogène ($\rho = 0$), alors la seule possibilité de solution survient lorsque la matrice L^{I} est singulière. Cela correspond au problème des modes propres (voir ci-dessous).

Dans le cas de conditions aux limites périodiques, il n'existe pas de frontière (F est vide) et donc L et L^{I} sont identiques.

C.2 Calcul du laplacien en dimension 1

Calculons maintenant les éléments de matrice D^2_{ij} de l'opérateur de Laplace en dimension 1, c'est-à-dire la dérivée seconde. En appliquant la définition de l'élément de matrice, on trouve

$$\mathrm{D}^2_{ij} = \langle u_i | \partial_x^2 | u_j \rangle = \int_0^\ell \mathrm{d}x \; u_i(x) \frac{\mathrm{d}^2}{\mathrm{d}x^2} u_j(x) \tag{12.26}$$

Comme les fonctions u_i sont linéaires, on pourrait naïvement conclure que leur dérivée seconde est nulle et donc que $\mathrm{D}^2_{ij} = 0$. Cependant, il n'en est rien, car la dérivée seconde n'est pas définie partout sur la fonction tente : elle est nulle presque partout, et infinie sur les bords et au sommet de la tente. Il faut donc procéder par limite : supposer que la tente est très légèrement arrondie à ces points, de manière à ce que la dérivée seconde soit partout bien définie. On procède ensuite à une intégration par parties :

$$\mathrm{D}^2_{ij} = \Big[u_i(x) u_j'(x) \Big]_0^\ell - \int_0^\ell \mathrm{d}x \; u_i'(x) u_j'(x) \tag{12.27}$$

On peut maintenant procéder à la limite, car cette expression est bien définie : les dérivées premières des fonctions tentes sont des constantes à l'intérieur du domaine de chaque fonction. Il est manifeste que D^2_{ij} s'annule si $|i-j|>1$ car les fonctions tentes ne se recouvrent pas dans ce cas. Ignorons pour le moment le problème des extrémités ($i=0$ et $i=\mathrm{N}-1$). Commençons par évaluer l'élément diagonal D^2_{ii} : le premier terme est manifestement nul car la fonction u_i est nulle aux extrémités ; le deuxième terme donne, quant à lui,

$$\mathrm{D}^2_{ii} = -\frac{1}{x_{i+1}-x_i} - \frac{1}{x_i - x_{i-1}} \tag{12.28}$$

Supposons ensuite que $j = i+1$. On trouve immédiatement que

$$\mathrm{D}^2_{i,i+1} = \frac{1}{x_{i+1}-x_i} \tag{12.29}$$

Comme l'opérateur de Laplace est hermitien, c'est-à-dire symétrique, il s'ensuit que $\mathrm{D}^2_{i+1,i} = \mathrm{D}^2_{i,i+1}$.

C.3 Problème aux valeurs propres

Certaines équations différentielles prennent la forme d'un problème aux valeurs propres :

$$\mathscr{L}|\psi\rangle = \lambda |\psi\rangle \tag{12.30}$$

où λ est la valeur propre inconnue (on suppose que des conditions aux limites appropriées sont appliquées sur $|\psi\rangle$. En appliquant la méthode de Galerkin à cette équation, on trouve

$$\langle u_i | \mathscr{L} | \psi \rangle = \lambda \langle u_i | \psi \rangle \tag{12.31}$$

Ensuite, substituons le développement $|\psi\rangle = \sum_j \psi_j |u_j\rangle$:

$$\sum_j \langle u_i | \mathscr{L} | u_j \rangle \psi_j = \lambda \sum_j \langle u_i | u_j \rangle \psi_j \tag{12.32}$$

c'est-à-dire

$$\sum_j \mathrm{L}_{ij}\psi_j = \lambda \sum_j \mathrm{M}_{ij}\psi_j \quad \text{ou encore} \quad \mathrm{L}\psi = \lambda \mathrm{M}\psi \tag{12.33}$$

Il s'agit de l'équation matricielle du problème aux valeurs propres généralisé.

Il faut tout de même appliquer les conditions aux limites. En séparant les valeurs aux frontières des valeurs intérieures, on a le système suivant :

$$\begin{pmatrix} \mathrm{L}^{\mathrm{F}} & \mathrm{L}^{\mathrm{FI}} \\ \mathrm{L}^{\mathrm{IF}} & \mathrm{L}^{\mathrm{I}} \end{pmatrix} \begin{pmatrix} \psi^{\mathrm{F}} \\ \psi^{\mathrm{I}} \end{pmatrix} = \lambda \begin{pmatrix} \mathrm{M}^{\mathrm{F}} & \mathrm{M}^{\mathrm{FI}} \\ \mathrm{M}^{\mathrm{IF}} & \mathrm{M}^{\mathrm{I}} \end{pmatrix} \begin{pmatrix} \psi^{\mathrm{F}} \\ \psi^{\mathrm{I}} \end{pmatrix} \tag{12.34}$$

En réalité seule la composante intérieure de cette équation est applicable, car les valeurs aux frontières sont régies par les conditions aux limites et non par l'équation différentielle :

$$\mathrm{L}^{\mathrm{I}}\psi^{\mathrm{I}} + \mathrm{L}^{\mathrm{IF}}\psi^{\mathrm{F}} = \lambda\left(\mathrm{M}^{\mathrm{I}}\psi^{\mathrm{I}} + \mathrm{M}^{\mathrm{IF}}\psi^{\mathrm{F}}\right) \tag{12.35}$$

Posons premièrement les conditions aux limites de Dirichlet : $\psi(0) = \psi(\ell) = 0$. Dans ce cas, $\psi^{\mathrm{F}} = 0$ et le problème aux valeurs propres se réduit à

$$\mathrm{L}^{\mathrm{I}}\psi^{\mathrm{I}} = \lambda \mathrm{M}^{\mathrm{I}}\psi^{\mathrm{I}} \tag{12.36}$$

Par contre, si les conditions de dérivées nulles (ou de Neumann) sont imposées, alors les choses sont différentes. En pratique, la condition de dérivées nulles aux frontières revient à $\psi_0 = \psi_1$ et $\psi_{\mathrm{N}-1} = \psi_{\mathrm{N}-2}$. Donc ψ^{F} est non nul mais sa valeur est liée à celle de certaines composantes de ψ^{I}, ce qui revient en pratique à modifier L^{I} de manière à lui ajouter certaines composantes de L^{IF}. Par exemple, dans le cas du laplacien en dimension 1, l'élément D^2_{10} appartient à $\mathrm{D}^{2\mathrm{F}}$ et doit être ajouté à D^2_{11}, qui appartient à $\mathrm{D}^{2\mathrm{I}}$, afin de tenir compte des conditions aux dérivées nulles. De même, $\mathrm{D}^2_{\mathrm{N}-1,\mathrm{N}-2}$ doit être ajouté à $\mathrm{D}^2_{\mathrm{N}-2,\mathrm{N}-2}$ et pareillement pour la matrice de masse M. Ces changements étant faits, les matrices ainsi modifiées $\tilde{\mathrm{D}}^{2\mathrm{I}}$ et $\tilde{\mathrm{M}}^{\mathrm{I}}$ figurent dans le problème aux valeurs propres généralisé approprié à ces conditions aux limites : $\tilde{\mathrm{L}}^{\mathrm{I}}\psi^{\mathrm{I}} = \lambda\tilde{\mathrm{M}}^{\mathrm{I}}\psi^{\mathrm{I}}$.

C.4 Exemple : équation de Helmholtz

Considérons une corde vibrante de longueur ℓ, fixée à ses extrémités. Le déplacement transversal de la corde, noté ψ, obéit à l'équation d'onde :

$$\frac{\partial^2\psi}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2\psi}{\partial t^2} = 0 \tag{12.37}$$

Supposons que la corde vibre à une fréquence ω. L'équation se réduit alors à l'équation de Helmholtz

$$\frac{\partial^2\psi}{\partial x^2} + k^2\psi = 0 \qquad\qquad k := \frac{\omega}{c} \tag{12.38}$$

avec les conditions aux limites $\psi(0) = \psi(\ell) = 0$.

L'opérateur différentiel pertinent ici est donc $\mathscr{L} = \partial_x^2 + k^2$. Les éléments de matrice de ∂_x^2 ont été calculés ci-haut. Les éléments de matrice de la constante k^2 sont donnés par la matrice de masse $\mathrm{M}_{ij} = \langle u_i | u_j \rangle$. Donc l'équation de Helmholtz sous forme matricielle se réduit à

$$(\mathrm{D}^2 + k^2\mathrm{M})\psi = 0 \tag{12.39}$$

Appliquons maintenant la condition aux limites $\psi_0 = \psi_{N-1} = 0$ du problème de la corde vibrante. L'équation $L^I \psi^I = 0$ se réduit alors à

$$D^{2I}\psi = -k^2 M^I \psi \quad \text{ou encore} \quad (M^I)^{-1} D^{2I}\psi = -k^2 \psi \tag{12.40}$$

L'équation (12.40) est une équation aux valeurs propres : seules les valeurs de k qui sont des valeurs propres de $(M^I)^{-1}D^{2I}$ sont admissibles. Ces valeurs propres déterminent alors les fréquences propres du problème, et les vecteurs propres correspondants sont les modes propres d'oscillation de la corde vibrante.

En pratique, il n'est pas courant d'inverser la matrice M^I, surtout en dimension supérieure à un. On tente alors de résoudre directement l'équation aux valeurs propres généralisée : $Ax = \lambda Bx$, où B est une matrice définie positive. Ce problème est plus difficile à résoudre que le problème aux valeurs propres ordinaire, et la recherche de méthodes efficaces pour ce faire, quand les matrices A et B sont énormes, est encore un champ actif de recherche.

Problème 12.2 : Cas d'une grille uniforme
Considérons l'équation de Helmholtz en dimension 1 :

$$\frac{\partial^2 \psi}{\partial x^2} + k^2 \psi = 0 \qquad \psi(0) = \psi(\ell) = 0 \tag{12.41}$$

A Donnez une expression exacte (analytique) des valeurs propres k^2 et des vecteurs propres correspondants dans ce problème. Portez une attention particulière aux valeurs propres dégénérées.

B Calculez les matrices M^I et D^{2I} dans le cas d'une grille uniforme, pour $\ell = 1$ et $N = 20$.

C Décrivez comment l'application de la matrice D^{2I} sur le vecteur ψ se compare à la formule élémentaire pour la dérivée seconde par différences finies :

$$(D_x^2 \psi)_i = \frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{a^2} = \psi_i'' + \mathcal{O}(a^2) \tag{12.42}$$

D Vérifiez numériquement que, dans ce cas d'une grille uniforme, les matrice D^{2I} et M^I commutent entre elles. Quelle relation s'ensuit-il entre les vecteurs propres de D^{2I} et ceux de $(M^I)^{-1}D^{2I}$?

E Comparez les 4 valeurs propres les plus petites (en valeur absolue) de la solution exacte, de D^{2I} et de $(M^I)^{-1}D^{2I}$. Dressez un tableau de ces valeurs propres et de l'écart relatif avec la valeur exacte.

F Portez les 4 premiers vecteurs propres en graphique, en les comparant aux fonctions propres exactes du problème (superposez un graphique continu et un graphique discret).

C.5 Exemple : équation de Schrödinger

Considérons l'équation de Schrödinger indépendante du temps en dimension 1, pour une particule se déplaçant dans un potentiel $\mathrm{V}(x)$:

$$-\frac{1}{2m}\frac{\partial^2\psi}{\partial x^2}+\mathrm{V}(x)\psi=\mathrm{E}\psi \tag{12.43}$$

Nous avons posé $\hbar=1$, ce qui revient à supposer que E et $\mathrm{V}(x)$ ont les unités de la fréquence. L'opérateur différentiel approprié à ce problème est donc

$$\mathscr{L}=-\frac{1}{2m}\partial_x^2+\mathrm{V}(x) \tag{12.44}$$

et il faut en pratique calculer les éléments de matrice du potentiel :

$$\mathrm{V}_{ij}=\int \mathrm{d}x\ u_i(x)u_j(x)\mathrm{V}(x) \tag{12.45}$$

L'élément V_{ij} n'est non nul que si $|i-j|\leq 1$, sinon les fonctions u_i et u_j ne se recouvrent pas. Bref, le support matriciel de V est le même que celui de la matrice de masse M. Dans le détail, on a :

$$\mathrm{V}_{ii}=\int_{x_{i-1}}^{x_i}\left(\frac{x-x_{i-1}}{x_i-x_{i-1}}\right)^2\mathrm{V}(x)+\int_{x_i}^{x_{i+1}}\left(\frac{x-x_i}{x_i-x_{i+1}}\right)^2\mathrm{V}(x) \tag{12.46}$$

$$\mathrm{V}_{i,i+1}=\int_{x_i}^{x_{i+1}}\left(\frac{x-x_{i-1}}{x_i-x_{i-1}}\right)\left(\frac{x-x_i}{x_i-x_{i+1}}\right)\mathrm{V}(x) \tag{12.47}$$

Le problème aux valeurs propres généralisé à résoudre est donc

$$\left(-\frac{1}{2m}\mathrm{D}^2+\mathrm{V}\right)\psi=\mathrm{EM}\psi \tag{12.48}$$

Boîte à outils

Le module scipy.sparse.linalg a déjà été signalé dans un chapitre précédent. La méthode des éléments finis produit des matrices creuses, qui sont naturellement traitées dans ce module de scipy. En particulier, la fonction `eigsh()` permet de calculer les valeurs propres extrêmes d'une matrice hermitiennes. Le problème aux valeurs propres généralisé peut aussi se traiter en fournissant la matrice définie positive M en argument optionnel.

D Méthodes spectrales

La méthode des éléments finis est caractérisée par l'utilisation de fonctions de base locales (les fonctions tentes des sections précédentes). Ces fonctions sont bien adaptées aux géométries compliquées, en raison de la possibilité de faire des triangulations, mais ne constituent pas la meilleure solution dans le cas des géométries simples. Dans cette section, nous allons expliquer comment utiliser une base de polynômes afin de représenter les fonctions plus efficacement (c'est-à-dire avec moins de fonctions de base).

D.1 Bases de polynômes orthogonaux et fonctions cardinales

Les méthodes spectrales utilisent les polynômes orthogonaux comme fonctions de base, ou encore des fonctions trigonométriques dans le cas de conditions aux limites périodiques. La discussion qui suit supposera qu'un intervalle ouvert (c.-à-d. non périodique) est considéré. Le cas périodique sera traité à la section D.5 ci-dessous.

Fixons un entier N. Les N racines x_i de $p_{\mathrm{N}}(x)$ sont utilisées comme points de grille. Définissons aussi les polynômes orthonormés

$$\phi_k(x) = \frac{1}{\sqrt{\gamma_k}} p_k(x) \qquad \langle \phi_k | \phi_m \rangle = \delta_{km} \qquad k, m = 0, \ldots, \mathrm{N}-1 \tag{12.49}$$

Une fonction $\psi(x)$ admet alors le développement limité suivant :

$$\psi(x) \approx \sum_{k=0}^{\mathrm{N}-1} \bar{\psi}_k \phi_k(x) \qquad \bar{\psi}_k = \langle \phi_k | \psi \rangle = \sum_{i=1}^{\mathrm{N}} w_i \psi_i \phi_k(x_i) \qquad \psi_i := \psi(x_i) \tag{12.50}$$

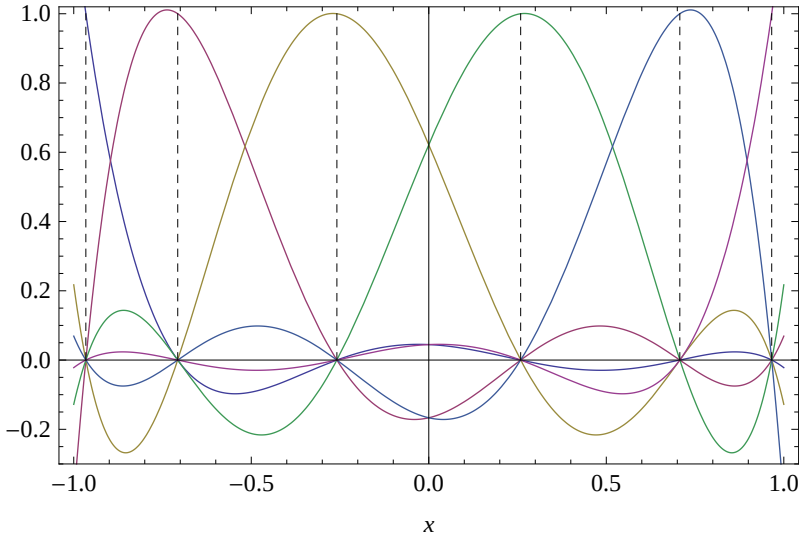


FIGURE 12.3
Les fonctions cardinales associées aux 6 racines du polynôme de Tchébychev $\mathrm{T}_6(x)$.

Les polynômes orthogonaux sont des fonctions oscillantes étendues sur tout l'intervalle $[a, b]$. En ce sens ils sont l'analogue des ondes planes (fonctions trigonométriques). Il est généralement plus utile d'avoir recours à des fonctions localisées, comme dans la méthode des éléments finis, mais qui nous permettent de profiter des excellentes propriétés de convergence des séries définies sur des bases de polynômes orthogonaux. C'est pour cela qu'on définit les *fonctions cardinales* :

$$\mathrm{C}_i(x) := \prod_{\substack{j=1 \\ j \neq i}}^{\mathrm{N}} \frac{x - x_j}{x_i - x_j} \tag{12.51}$$

La fonction $\mathrm{C}_i(x)$ est un polynôme de degré $\mathrm{N}-1$ qui s'annule à tous les points x_j de la grille, sauf au point x_i où elle est égale à l'unité : $\mathrm{C}_j(x_i) = \delta_{ij}$.

Nous avons donc à notre disposition deux bases de fonctions pour l'intervalle $[a,b]$: les polynômes orthogonaux normalisés $\phi_j(x)$ (ou $|\phi_j\rangle$) et les fonctions cardinales $\mathrm{C}_j(x)$ (ou $|\mathrm{C}_j\rangle$). Ces deux bases sont toutes les deux polynomiales de même degré, et sont reliées par une transformation de similitude, qu'on trouve facilement en exprimant les fonctions cardinales sur la base des polynômes orthogonaux :

$$|\mathrm{C}_j\rangle = \sum_i \mathrm{M}_{ij}|\phi_i\rangle \quad \text{où} \quad \mathrm{M}_{ij} \coloneqq \langle\phi_i|\mathrm{C}_j\rangle = \sum_k w_k \phi_i(x_k)\mathrm{C}_j(x_k) = w_j\phi_i(x_j) \tag{12.52}$$

car $\mathrm{C}_j(x_k) = \delta_{jk}$.

Le développement d'une fonction $\psi(x)$ sur la base des fonctions cardinales est très simple :

$$\psi(x) \approx \sum_{i=0}^{\mathrm{N}-1} \psi(x_i)\mathrm{C}_i(x) = \sum_{i=0}^{\mathrm{N}-1} \psi_i \mathrm{C}_i(x) \quad \text{ou encore} \quad |\psi\rangle \approx \sum_{i=0}^{\mathrm{N}-1} \psi_i |\mathrm{C}_i\rangle \tag{12.53}$$

On démontre ce développement en substituant $x = x_j$ ($j = 0, 1, \cdots, \mathrm{N}-1$). Le signe $\approx$ devient une égalité si la fonction ψ est un polynôme de degré $\mathrm{N}-1$ ou moins.

La relation existant entre les coefficients $\bar{\psi}_i$ et ψ_i est

$$\bar{\psi}_i = \langle\phi_i|\psi\rangle = \sum_j \langle\phi_i|\mathrm{C}_j\rangle\psi_j = \sum_j \mathrm{M}_{ij}\psi_j \tag{12.54}$$

ou, en notation matricielle : $\bar{\psi} = \mathrm{M}\psi$. On vérifie d'ailleurs, d'après l'expression ci-dessus de M_{ij} et de $\bar{\psi}_i$, que

$$\sum_j \mathrm{M}_{kj}\psi_j = \sum_j w_j \phi_k(x_j)\psi_j = \bar{\psi}_k \tag{12.55}$$

comme démontré plus haut.

Notons que la base des fonctions cardinales est orthogonale, mais pas orthonormée :

$$\langle\mathrm{C}_i|\mathrm{C}_j\rangle = \sum_k w_k \mathrm{C}_i(x_k)\mathrm{C}_j(x_k) = \sum_k w_k \delta_{ik}\delta_{jk} = w_i\delta_{ij} \tag{12.56}$$

D.2 Opérateur différentiel

Considérons maintenant un opérateur différentiel linéaire $\mathscr{L}$, par exemple le laplacien. Sur la base des fonctions cardinales, cet opérateur a une forme matricielle L telle que

$$\mathscr{L}|\mathrm{C}_i\rangle = \sum_j \mathrm{L}_{ji}|\mathrm{C}_j\rangle \tag{12.57}$$

(notez l'ordre des indices de la matrice L). L'équation différentielle $\mathscr{L}\psi(x) = f(x)$, où $f(x)$ est le vecteur de charge (connu) et $\psi(x)$ la fonction recherchée, s'écrit abstraitement de la manière suivante : $\mathscr{L}|\psi\rangle = |f\rangle$. L'utilisation des fonctions cardinales est étroitement liée à la méthode de collocation, selon laquelle l'équation différentielle est imposée aux points de grille, à savoir

$$\langle x_i|\mathscr{L}|\psi\rangle = \langle x_i|f\rangle = f(x_i) = f_i \tag{12.58}$$

D'après la définition de L_{ij} ci-haut, le membre de gauche de cette équation devient

$$\begin{aligned}\langle x_i|\mathscr{L}|\psi\rangle &= \sum_j \psi_j \langle x_i|\mathscr{L}|\mathrm{C}_j\rangle \\ &= \sum_{j,k} \psi_j \mathrm{L}_{kj}\langle x_i|\mathrm{C}_k\rangle \\ &= \sum_{j,k} \psi_j \mathrm{L}_{kj}\delta_{ik} \\ &= \sum_j \mathrm{L}_{ij}\psi_j\end{aligned} \tag{12.59}$$

ce qui s'exprime comme $\mathrm{L}\psi$ en notation matricielle, ψ étant le vecteur colonne des coefficients ψ_i. L'équation différentielle devient donc l'équation matricielle

$$\mathrm{L}\psi = f \tag{12.60}$$

Les matrices associées aux dérivées d'ordre quelconque $\mathscr{L} = \partial_x^{(n)}$ peuvent être calculées relativement facilement à l'aide de l'expression explicite (12.51) des fonctions cardinales. L'opérateur de différentiation ∂_x a la représentation générale

$$\partial_x \mathrm{C}_i(x) = \sum_k \mathrm{D}^{(1)}_{ki}\mathrm{C}_k(x) \tag{12.61}$$

et la matrice $\mathrm{D}^{(1)}_{ji}$ n'est rien d'autre que la valeur de la dérivée de $\mathrm{C}_i(x)$ évaluée à $x = x_j$, comme on peut le voir en posant $x = x_j$ dans l'équation ci-dessus :

$$\mathrm{C}'_i(x_j) = \sum_k \mathrm{D}^{(1)}_{ki}\mathrm{C}_k(x_j) = \sum_k \mathrm{D}^{(1)}_{ki}\delta_{kj} = \mathrm{D}^{(1)}_{ji} \tag{12.62}$$

Les matrices associées aux dérivées d'ordre supérieur peuvent être calculées simplement en prenant les puissances appropriées de la matrice $\mathrm{D}^{(1)}$.

Résumé : Propriétés des fonctions cardinales

$$\mathrm{C}_i(x) = \prod_{j=1, j\neq i}^{\mathrm{N}} \frac{x - x_j}{x_i - x_j} \qquad \mathrm{C}_i(x_j) = \delta_{ij} \qquad \langle \mathrm{C}_i|\mathrm{C}_j\rangle = w_i\delta_{ij}$$

$$|\mathrm{C}_j\rangle = \sum_i \mathrm{M}_{ij}|\phi_i\rangle \qquad \mathrm{M}_{ij} = w_j\phi_i(x_j)$$

$$\partial_x \mathrm{C}_i(x) = \sum_k \mathrm{D}^{(1)}_{ki}\mathrm{C}_k(x) \qquad \mathrm{D}^{(1)}_{ji} = \mathrm{C}'_i(x_j)$$

D.3 Quadratures de Lobatto

La quadrature gaussienne se base sur des abscisses qui sont strictement contenues à l'intérieur du domaine $[a, b]$. Cela pose un problème si on veut représenter des fonctions qui ont des valeurs précises aux frontières de l'intervalle. On peut remédier à ce problème en définissant une approche

légèrement différente basée sur la formule de quadrature de Lobatto qui, elle, inclut les points extrêmes. Cette formule existe en général pour une fonction poids $w(x)$ quelconque et s'obtient en imposant que les extrémités de l'intervalle (a et b) fassent partie de la grille d'intégration, qui comporte alors $\mathrm{N}-2$ points intérieurs et 2 points sur la frontière. Les poids w_i et les noeuds intérieurs forment un ensemble de $2\mathrm{N}-2$ paramètres ajustables qui sont déterminés afin de rendre l'intégrale exacte pour des polynômes du plus haut degré possible, c'est-à-dire $2\mathrm{N}-3$.

La formule de Lobatto pour les polynômes de Legendre est

$$\int_{-1}^{1} \mathrm{d}x\, f(x) \approx \sum_{i=1}^{\mathrm{N}} w_i f(x_i) \tag{12.63}$$

où

$$\begin{aligned} &x_1 = -1 \qquad x_{\mathrm{N}} = 1 \qquad w_1 = w_{\mathrm{N}} = \frac{2}{\mathrm{N}(\mathrm{N}-1)} \\ &w_i = \frac{2}{\mathrm{N}(\mathrm{N}-1)[\mathrm{P}_{\mathrm{N}-1}(x_i)]^2} \qquad \mathrm{P}'_{\mathrm{N}-1}(x_i) = 0 \qquad i = 2, \ldots, \mathrm{N}-1 \end{aligned} \tag{12.64}$$

Autrement dit, les points intérieurs sont les $\mathrm{N}-2$ racines de la dérivée de $\mathrm{P}_{\mathrm{N}-1}$, qui est un polynôme d'ordre $\mathrm{N}-2$.

La formule de Lobatto pour les polynômes de Tchébychev est

$$\begin{aligned} &\int_{-1}^{1} \mathrm{d}x\, \frac{f(x)}{\sqrt{1-x^2}} \approx \sum_{i=1}^{\mathrm{N}} w_i f(x_i) \qquad x_i = \cos\frac{(i-1)\pi}{\mathrm{N}-1} \\ &w_i = \frac{\pi}{\mathrm{N}-1} \quad (0 < i < \mathrm{N}), \qquad w_0 = w_{\mathrm{N}} = \frac{1}{2}\frac{\pi}{\mathrm{N}-1} \end{aligned} \tag{12.65}$$

Notons que la position des abscisses et les poids sont particulièrement simples avec ces polynômes.

Nous ne démontrerons pas les formules de Lobatto ici, comme nous l'avons fait pour la formule d'intégration gaussienne en général. Ces formules fournissent des approximations différentes au produit scalaire (11.27), qui sont d'ordre plus bas : exactes pour les polynômes de degré $2\mathrm{N}-3$ au lieu de $2\mathrm{N}-1$. Par contre, leur avantage est que les fonctions cardinales associées $\mathrm{C}_i(x)$ permettent de représenter les valeurs aux frontières de l'intervalle. En ce sens, ces fonctions sont une alternative aux fonctions tentes utilisées dans la méthode des éléments finis, et le traitement des conditions aux limites peut se faire exactement comme dans le section C, c'est-à-dire en définissant une représentation L^{I} de l'opérateur différentiel pour les points intérieurs

D.4 Exemple : équation de Helmholtz

Revisitons le problème des valeurs propres de l'équation de Helmholtz étudié à la section C.4. Il s'agit ici de calculer les valeurs propres du laplacien $\mathscr{L} = \partial_x^2$, donc de résoudre l'équation aux valeurs propres

$$\mathscr{L}|\psi\rangle = \lambda|\psi\rangle \tag{12.66}$$

En projetant sur les fonctions $\langle x_i|$, cette équation devient

$$\langle x_i|\mathscr{L}|\psi\rangle = \mathrm{L}_{ij}\psi_j = \lambda\psi_i \quad \text{ou encore} \quad \mathrm{L}\psi = \lambda\psi \tag{12.67}$$

n	exact	éléments finis	méthode spectrale
1	2.4674	2.4676	2.4674
2	9.8696	9.8728	9.8696
3	22.206	22.223	22.206
4	39.478	39.530	39.472
5	61.685	61.812	61.900

TABLE 12.1
Les 5 premières valeurs propres de l'opérateur ∂_x^2 dans l'intervalle $[-1,1]$. À droite : valeurs exactes $\lambda_n = (n\pi/2)^2$. Au milieu, valeurs obtenues à l'aide de la méthode des éléments finis et N = 101 points de grille. À gauche, valeurs obtenues à l'aide d'une méthode spectrale et d'une grille de Gauss-Lobatto de N = 12 points.

en notation matricielle. Il s'agit donc d'une équation aux valeurs propres en fonction de la matrice L. La méthode de collocation mène au problème ordinaire des valeurs propres et non au problème généralisé. De plus, la dimension de la matrice est généralement beaucoup plus petite.

Il faut cependant tenir compte des conditions aux limites, c'est-à-dire imposer l'annulation de la fonction aux extrémités de l'intervalle. Dans l'intervalle $[-1,1]$ associé aux polynômes de Tchébychev, cela revient à demander $\psi(-1) = \psi(1) = 0$. Pour appliquer cette contrainte, nous avons besoin de points de grille aux extrémités, donc de la grille de Tchébychev-Lobatto. L'équation aux valeurs propres elle-même n'est valable que pour les points intérieurs de la grille.

Nous devons donc partitionner la matrice L comme expliqué à la fin de la section C. L'équation aux valeurs propres prend alors la forme

$$\begin{pmatrix} \mathrm{L}^{\mathrm{F}} & \mathrm{L}^{\mathrm{FI}} \\ \mathrm{L}^{\mathrm{IF}} & \mathrm{L}^{\mathrm{I}} \end{pmatrix} \begin{pmatrix} 0 \\ \psi^{\mathrm{I}} \end{pmatrix} = \lambda \begin{pmatrix} 0 \\ \psi^{\mathrm{I}} \end{pmatrix} \tag{12.68}$$

ce qui revient à demander

$$\mathrm{L}^{\mathrm{I}} \psi^{\mathrm{I}} = \lambda \psi^{\mathrm{I}} \tag{12.69}$$

Autrement dit, nous devons chercher les valeurs propres de la matrice intérieure seulement.

Le tableau 12.1 illustre les résultats obtenus de cette manière et les compare aux résultats exacts et à ceux obtenus par la méthode des éléments finis. Le fait remarquable est que la méthode spectrale avec seulement 12 points (donc 10 points intérieurs) est beaucoup plus précise que la méthode des éléments finis avec 101 points, du moins pour les valeurs propres les plus basses. En fait, les valeurs obtenues avec la méthode spectrale cessent d'être fiables après $\sim n/2$ valeurs propres, n étant le nombre de points utilisés. Mais souvent ce sont les valeurs propres les plus basses qui sont recherchées.

D.5 Conditions aux limites périodiques

Lorsqu'on a affaire à des conditions aux limites périodiques, c'est-à-dire lorsque le problème est défini sur un cercle au lieu d'un segment ouvert, les fonctions spectrales les plus utiles ne sont pas

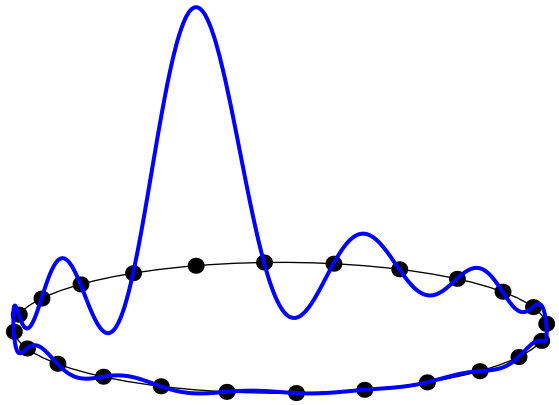


FIGURE 12.4
Fonction cardinale sur une grille périodique de 24 points. Les 24 fonctions sont identiques à celle illustrée ici, sauf pour une translation. L'espace périodique est ici représenté comme un cercle.

les polynômes de Legendre ou de Tchébychev, mais plutôt les fonctions trigonométriques qui sont d'emblée périodiques. Dans cette section nous allons décrire les fonctions cardinales utilisées dans les problèmes périodiques.

Il est toujours possible, par un changement d'échelle approprié, de fixer la longueur de l'intervalle périodique à 2π. Nous allons adopter une grille de N points également espacés (N étant pair) [1] :

$$x_j = \frac{2\pi j}{\mathrm{N}} \qquad j = 0, 1, 2, \ldots, \mathrm{N}-1 \tag{12.70}$$

Les fonctions cardinales appropriées à cette grille sont définies comme suit :

$$\mathrm{C}_i(x) = \frac{1}{\mathrm{N}} \frac{\sin\left(\frac{1}{2}\mathrm{N}(x_i - x)\right)}{\sin\left(\frac{1}{2}(x_i - x)\right)} \cos\left(\tfrac{1}{2}(x_i - x)\right) \tag{12.71}$$

Ces fonctions ne sont pas des polynômes en x (les polynômes ne sont pas périodiques de toute manière). Par contre, elles ont les propriétés suivantes :

1. Elles sont périodiques, de période 2π, à condition que N soit pair. Voir à cet effet la figure 12.4.
2. $\mathrm{C}_i(x_j) = \delta_{ij}$. Cela se vérifie immédiatement si $i \neq j$. Pour $i = j$, il s'agit d'un processus de limite standard pour le rapport des sinus. Donc le développement d'une fonction périodique $\psi(x)$ sur la base des fonctions cardinales s'effectue comme auparavant :

$$\psi(x) = \sum_{i=0}^{\mathrm{N}-1} \psi_i \mathrm{C}_i(x) \quad \text{ou} \quad \psi_i \coloneqq \psi(x_i) \tag{12.72}$$

1. il n'y a aucune raison que les points ne soient pas également espacés, étant donnée l'invariance par translation

3. En fonction d'exponentielles complexes, la fonction $C_0(x)$ s'exprime comme

$$\begin{aligned} C_0(x) &= \frac{1}{2N}\frac{e^{iNx/2}-e^{-iNx/2}}{e^{ix/2}-e^{-ix/2}}\left(e^{ix/2}+e^{-ix/2}\right) \\ &= \frac{1}{2N}\frac{z^{N/2}-z^{-N/2}}{z^{1/2}-z^{-1/2}}(z^{1/2}+z^{-1/2}) \qquad\qquad z := e^{ix} \\ &= \frac{z^{-N/2}}{2N}\frac{z^N-1}{z-1}(z+1) \\ &= \frac{z^{-N/2}}{2N}(1+z+z^2+\cdots+z^{N-1})(z+1) \\ &= \frac{1}{N}\left(\tfrac{1}{2}z^{-N/2}+z^{-N/2+1}+z^{-N/2+2}+\cdots+z^{N/2-1}+\tfrac{1}{2}z^{N/2}\right) \end{aligned} \tag{12.73}$$

   Autrement dit, $C_0(x)$ est un *polynôme trigonométrique* de degré $N/2$, c'est-à-dire une combinaison des puissances entières de $z = e^{ix}$, de $-N/2$ at $N/2$.

4. La fonction $C_j(x)$ s'obtient en remplaçant x par $x - x_j$, ou encore z par $w := \omega^j z$, où $\omega := e^{-2\pi i/N}$:

$$C_j(x) = \frac{1}{N}\left[\tfrac{1}{2}w^{-N/2}+w^{-N/2+1}+w^{-N/2+2}+\cdots+w^{N/2-1}+\tfrac{1}{2}w^{N/2}\right] \tag{12.74}$$

   Les N fonctions cardinales sont toutes des polynômes trigonométriques de degré $N/2$. Elles sont linéairement indépendantes. Elles sont aussi réelles, alors que w est complexe, en raison de leur invariance lors du remplacement $w \to w^{-1}$. Le nombre de degrés de liberté de l'ensemble des polynômes respectant cette condition est $N+1$. Comme il n'y a que N fonctions cardinales, l'un de ces polynômes ne peut pas être exprimé comme une combinaison de fonctions cardinales : il s'agit de

$$i\left(z^{-N/2}-z^{N/2}\right) = 2\sin\frac{Nx}{2} \tag{12.75}$$

   comme on peut le vérifier aisément en vérifiant que les coefficients de développement (12.72) sont tous nuls.

5. On montre facilement que l'action de la dérivée première sur les fonctions cardinales est la suivante :

$$C_j'(x_i) = D^{(1)}_{ij} = \begin{cases} 0 & (i=j) \\ \frac{1}{2}(-1)^{i+j}\cot\left(\frac{x_i-x_j}{2}\right) & (i\neq j) \end{cases} \tag{12.76}$$

   La matrice $D^{(2)}$ représentant la dérivée seconde est simplement le carré de $D^{(1)}$.

Problème 12.3 :
Montrez que

$$C_i(x) = \frac{p_N(x)}{(x-x_i)p_N'(x_i)} \tag{12.77}$$

où p_N' est la dérivée du polynôme p_N. Indice : les racines du polynôme C_i sont presque les mêmes

que celles de p_{N}.

Solution

On peut faire le calcul directement. L'expression de p_{N} et de sa dérivée en fonction de ses pôles est

$$p_{\mathrm{N}}(x)=\prod_{j=1}^{\mathrm{N}}(x-x_j) \qquad\qquad p'_{\mathrm{N}}(x)=\sum_{k=1}^{\mathrm{N}}\prod_{\substack{j=1\\ i\neq k}}^{\mathrm{N}}(x-x_j)$$

En substituant $x=x_i$ dans $p'_{\mathrm{N}}(x)$, on constate qu'un seul des N termes ($k=i$) survit :

$$p'_{\mathrm{N}}(x_i)=\prod_{\substack{j=1\\ j\neq i}}^{\mathrm{N}}(x_i-x_j)$$

L'équation (12.77) s'ensuit immédiatement (on divise par $(x-x_i)$ pour enlever le pôle à x_i).

Problème 12.4 :
Montrez que $\langle \mathrm{C}_i|\mathscr{L}\mathrm{C}_j\rangle = \mathrm{L}_{ij}\,w_i$. Quelle relation s'applique à la matrice L si $\mathscr{L}$ est un opérateur hermitien ?

Solution

Comme $\langle \mathrm{C}_i|\mathrm{C}_j\rangle = w_i\delta_{ij}$, on trouve immédiatement

$$\langle \mathrm{C}_i|\mathscr{L}|\mathrm{C}_j\rangle = \langle \mathrm{C}_i|\sum_k \mathrm{L}_{kj}|\mathrm{C}_k\rangle = \sum_k \mathrm{L}_{kj}\,w_i\delta_{ik} = \mathrm{L}_{ij}\,w_i$$

Si $\mathscr{L}$ est hermitien et réel, alors

$$\langle \mathrm{C}_i|\mathscr{L}\mathrm{C}_j\rangle = \langle \mathscr{L}\mathrm{C}_i|\mathrm{C}_j\rangle = \langle \mathrm{C}_j|\mathscr{L}\mathrm{C}_i\rangle$$

et donc

$$\mathrm{L}_{ij}\,w_i = \mathrm{L}_{ji}\,w_j$$

La matrice L n'est donc pas symétrique.

Problème 12.5 :
Montrez que les N fonctions cardinales (12.74) sont linéairement indépendantes. Indice : exprimez dans la base des puissances z^n, et calculez le déterminant formé des N fonctions cardinales. C'est un déterminant de Vandermonde.

Solution

Le déterminant des N fonctions cardinales est

$$\begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \cdots & \omega^{(N-1)(N-1)} \end{vmatrix} = \prod_{i<j} (\omega^i - \omega^j)$$

Comme les N premières puissances de ω sont toutes différentes (ce sont les N racines N^{eme} de l'unité), ce déterminant est non nul est les fonctions sont linéairement indépendantes.

CHAPITRE 13

PROBLÈMES AUX LIMITES : DIMENSION 2

La théorie générale des éléments finis en dimension supérieure à un n'est pas différente de ce qu'elle est en dimension 1. En particulier, la discussion de la section C se transpose sans modification en dimension supérieure, sauf pour le nombre de points à la frontière, qui est bien sûr plus grand que 2. La difficulté principale associée à une dimension supérieure vient de la forme plus complexe des fonctions tentes. La grille unidimensionnelle est typiquement remplacée, en dimension 2, par une *triangulation* (voir par exemple la figure 13.1).

A Triangulations

La construction de triangulations est en soi un sujet vaste, qui s'insère dans un champ d'études appelé *géométrie algorithmique* (angl. *computational geometry*). Partons d'un ensemble de points (les *noeuds*) qui forment la grille de points physique d'intérêt en dimension 2. Le problème est de construire un ensemble de triangles (la *triangulation*) à partir de ces points. Chaque triangle constitue alors une facette de l'espace et aucun noeud ne doit se trouver à l'intérieur d'un triangle. La solution à ce problème n'est pas unique.

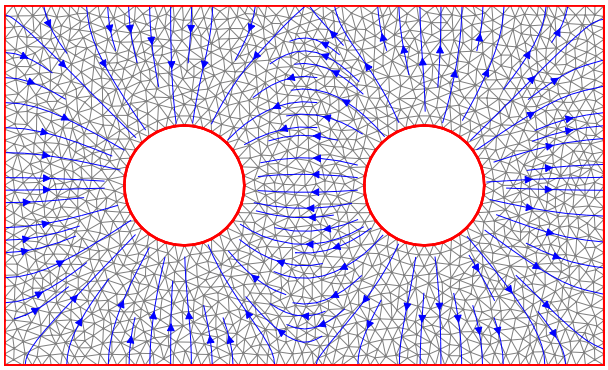


FIGURE 13.1
Triangulation de l'espace autour de deux conducteurs circulaires formant un condensateur. Le problème est bidimensionnel, mais représente un condensateur 3D beaucoup plus long que large, coupé en son milieu. Les parois du domaine forment un troisième conducteur, mis à terre, alors que les deux conducteurs principaux sont maintenus à des potentiels opposés. Les lignes de champ électrique sont indiquées.

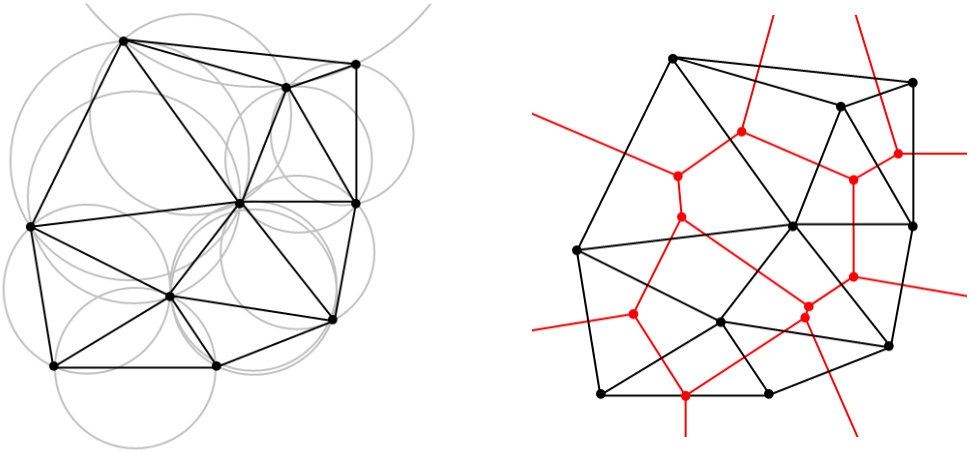


FIGURE 13.2
À gauche : triangulation de Delaunay. Le cercle circonscrit à chaque triangle ne contient aucun autre noeud. À gauche : diagramme de Voronoï (en rouge)[source : wikipedia]

Par contre, si on demande que le cercle circonscrit à chaque triangle ne contienne aucun autre noeud (les noeuds sur la circonférence étant permis), alors la solution devient unique. Les triangulations qui respectent cette condition sont appelées *triangulations de Delaunay*. La figure 13.2 illustre une triangulation de Delaunay avec les cercles circonscrits à chaque triangle. La partie droite de la figure illustre le *diagramme de Voronoï* correspondant. L'intérieur de chaque polygone du diagramme de Voronoï est l'ensemble des points qui sont plus proches de chaque noeud que de tout autre point.

L'avantage des triangulations de Delaunay est que les triangles sont les plus compacts possible et qu'un algorithme existe pour les construire. Cet algorithme est basé sur le *basculement* (voir figure 13.3) : les deux triangles ABD et CBD ne respectent pas la condition de Delaunay, comme on peut le voir à l'image du centre. On montre que la condition est violée à chaque fois que la somme

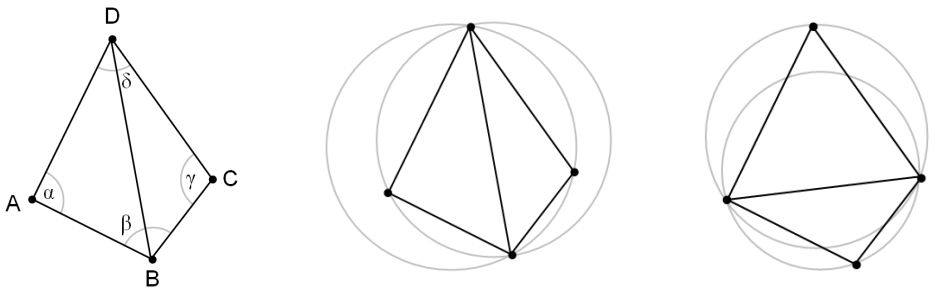


FIGURE 13.3
Illustration du basculement. Les deux triangles ABD et CBD ne respectent pas la condition de Delaunay. Mais en remplaçant le segment BD par le segment AC, on obtient deux nouveaux triangles (ABC et ADC) qui, eux, respectent la condition.[source : wikipedia]

des angles opposés (ici α et γ) est supérieure à π, ce qui est le cas ici. Par contre, la somme $\beta + \delta$ est alors forcément inférieure à π, et il suffit donc de remplacer le segment BD par le segment AC pour obtenir deux nouveaux triangles qui respectent la condition de Delaunay. De cette manière, on peut progressivement arriver à une triangulation entièrement conforme à la condition de Delaunay. Chaque paire de triangles opposés (c'est-à-dire partageant une arête) détermine au total 6 angles intérieurs. L'angle le plus petit, ou angle minimum, est plus grand si les deux triangles respectent la condition de Delaunay, que dans le cas contraire. En ce sens, les triangulations de Delaunay évitent plus que toutes les autres les angles petits, c'est-à-dire les triangles effilés. Notons cependant que la triangulation n'est définie qu'une fois l'ensemble des noeuds spécifié, et que le choix des noeuds peut entraîner l'existence de triangles effilés, même dans une triangulation de Delaunay.

Une triangulation de Delaunay est dite *contrainte* si elle se base non seulement sur un ensemble de noeuds, mais aussi sur un ensemble de segments, par exemple délimitant une région. Ces segments forment la frontière de la région physique d'intérêt, par exemple le bord de la région illustrée sur la figure 13.1, ainsi que le périmètre de chacune des deux plaques. La triangulation contrainte doit contenir les segments d'origine parmi les arêtes des triangles, en plus des noeuds.

En pratique, dans la solution d'un problème aux limites, on cherchera à raffiner la triangulation, c'est-à-dire ajouter des noeuds, jusqu'à ce qu'une certaine précision soit atteinte. Le raffinement des triangulations de Delaunay est un sujet de recherche actif. Nous utiliserons dans les travaux pratiques un programme utilisant l'algorithme de Ruppert. [1] Le raffinement procède par ajout de noeuds :

1. Chaque triangle est caractérisé par un cercle circonscrit. Si le rayon de ce cercle dépasse un certain maximum prescrit, un noeud supplémentaire est ajouté en son centre.
2. Des noeuds sont ajoutés sur le périmètre de la région au point milieu de segments, de manière à conserver un angle minimum prescrit dans tous les triangles. Ces noeuds additionnels sont appelés *points de Steiner*. L'ajout de ces noeuds permet d'augmenter la qualité locale des triangles, c'est-à-dire d'augmenter l'angle minimum.

Il y a naturellement un compromis à atteindre entre la qualité des triangles et le nombre de points : on doit augmenter le nombre de noeuds afin d'améliorer la qualité minimale des triangles.

B Fonctions tentes

En deux dimensions, les fonctions tentes ressemblent plus à de véritables tentes, comme illustré à la figure 13.4. Elles conservent les caractéristiques suivantes :

1. Leur valeur est 1 sur le noeud correspondant.
2. Elles n'ont de recouvrement qu'avec les fonctions des noeuds voisins, c'est-à-dire les noeuds reliés par un seul segment de la triangulation.

Nous allons utiliser la notation suivante dans ce qui suit :

- Les noeuds seront indexés un indice latin : leurs positions seront notées $\mathbf{r}_i$, $\mathbf{r}_j$, etc.
- Les triangles (ou faces) seront notés T_a ($a = 0, 1, \ldots, \mathrm{N_T} - 1$), $\mathrm{N_T}$ étant le nombre de faces.
- La surface de la face T_a sera notée A_a.

1. Voir http://en.wikipedia.org/wiki/Ruppert's_algorithm

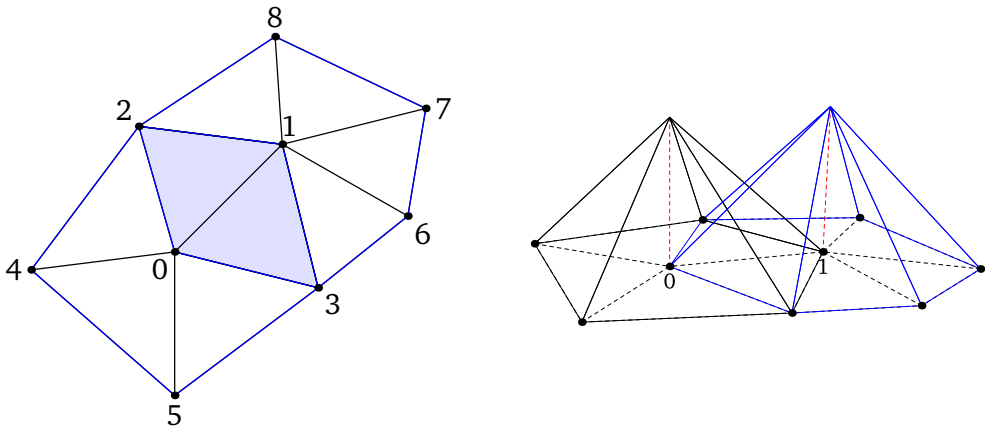


FIGURE 13.4
Fonctions tentes associées à une triangulation. À gauche : les triangles associés à deux fonctions tentes qui se recouvrent, centrées aux sites 0 et 1 respectivement. La zone de recouvrement est indiquée en bleu. À droite : vue 3D des deux fonctions tentes, pour les mêmes triangles.

- Les trois noeuds de chaque face seront notés $\mathbf{r}_{a1}$, $\mathbf{r}_{a2}$ et $\mathbf{r}_{a3}$; chacun de ces noeuds étant bien sûr partagé par plusieurs faces.
- Inversement, le triangle formé par les trois noeuds i, j et k (dans le sens antihoraire) sera noté T_{ijk}.

L'expression analytique de chaque fonction tente dépend bien sûr de la face considérée. Considérons à cet effet trois noeuds $\mathbf{r}_{1,2,3}$ délimitant une face, et trouvons l'expression de la fonction tente centrée à $\mathbf{r}_1$ sur cette face. Définissons d'abord la fonction

$$\begin{aligned}\gamma(\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3) &= \mathbf{z}\cdot[(\mathbf{r}_2-\mathbf{r}_1)\wedge(\mathbf{r}_3-\mathbf{r}_1)]\\ &= x_2y_3-x_3y_2-x_1y_3+x_3y_1+x_1y_2-x_2y_1\end{aligned} \tag{13.1}$$

D'après les propriétés du produit vectoriel, cette fonction est 2 fois l'aire orientée de la face formée des trois points en question, si ces points sont pris dans le sens antihoraire. Elle possède les propriétés suivantes :

1. Elle est antisymétrique lors de l'échange de deux arguments et inchangée lors d'une permutation cyclique de ses trois arguments.
2. Elle est linéaire dans chacun de ses arguments

La fonction tente centrée à $\mathbf{r}_1$ est linéaire en (x,y), s'annule à $\mathbf{r}=\mathbf{r}_2$ et $\mathbf{r}=\mathbf{r}_3$, et est égale à 1 si $\mathbf{r}=\mathbf{r}_1$. la seule possibilité est

$$u_{1,2,3}(\mathbf{r}) = \frac{\gamma(\mathbf{r},\mathbf{r}_2,\mathbf{r}_3)}{\gamma(\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3)} \tag{13.2}$$

Cette expression répond manifestement aux critères, et de plus est unique, car une seule fonction linéaire (un seul plan) passe par les trois points $(x_1,y_1,1)$, $(x_2,y_2,0)$ et $(x_3,y_3,0)$. Nous utiliserons la notation abrégée

$$\gamma_{ijk} = \gamma(\mathbf{r}_i,\mathbf{r}_j,\mathbf{r}_k) \tag{13.3}$$

où il est sous-entendu que les trois points forment une face de la triangulation.

La fonction tente complète centrée en $\mathbf{r}_0$ sur la figure 13.4 sera donc

$$u_0(\mathbf{r}) = \begin{cases} u_{012}(\mathbf{r}) & \text{si} \quad \mathbf{r} \in \mathrm{T}_{012} \\ u_{024}(\mathbf{r}) & \text{si} \quad \mathbf{r} \in \mathrm{T}_{024} \\ \cdots & \cdots \\ u_{031}(\mathbf{r}) & \text{si} \quad \mathbf{r} \in \mathrm{T}_{031} \end{cases} \tag{13.4}$$

Les fonctions tente de deux noeuds voisins partagent deux faces situées de part et d'autre du lien qui relie les deux noeuds, comme illustré à la figure 13.4.

Problème 13.1 :
Démontrez les propriétés de la fonction $\gamma(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$: (i) antisymétrie lors de l'échange de deux arguments et (ii) linéarité dans chacun de ses arguments.

C Évaluation du laplacien

Supposons que nous ayons à notre disposition une triangulation de l'ensemble des noeuds $\mathbf{r}_i$ en deux dimensions. L'opérateur de Laplace ∇^2 intervient dans un grand nombre d'équations différentielles et il est donc nécessaire dans ces cas d'en calculer une représentation dans la base des fonctions tente. Le problème est donc de calculer les éléments de matrice

$$\mathrm{D}^2_{ij} = \langle u_i | \nabla^2 | u_j \rangle = \int \mathrm{d}^2 r \; u_i(\mathbf{r}) \nabla^2 u_j(\mathbf{r}) \tag{13.5}$$

L'évaluation se fait encore une fois par intégration par parties, en utilisant l'identité de Green :

$$\int_{\mathrm{R}} \mathrm{d}^2 r \, f \nabla^2 g = \oint_{\partial \mathrm{R}} \mathrm{d}\mathbf{a} \cdot \nabla g \, f - \int_{\mathrm{R}} \mathrm{d}^2 r \, \nabla f \cdot \nabla g \tag{13.6}$$

où la première intégrale est menée sur le périmètre $\partial \mathrm{R}$ de la région R considérée.

Les fonctions $u_i(\mathbf{r})$ étant linéaires, leur gradient est constant et les intégrales ne présentent donc pas de difficulté, sinon dans l'organisation des différentes faces impliquées. De plus, les fonctions s'annulant sur le périmètre de leur domaine, le terme de périmètre ne contribue jamais.

Chaque fonction u_i est une combinaison des fonctions u_{ijk} définies en (13.2). On calcule sans peine que

$$\nabla u_{ijk}(\mathbf{r}) = \frac{1}{\gamma_{ijk}} \mathbf{z} \wedge (\mathbf{r}_k - \mathbf{r}_j) \tag{13.7}$$

et ensuite que

$$\int_{\mathrm{T}(ijk)} \mathrm{d}^2 r \, (\nabla u_{ijk})^2 = \frac{\ell^2_{jk}}{4 \mathrm{A}_{ijk}} \tag{13.8}$$

où ℓ_{jk} est la longueur du segment reliant j à k. D'autre part,

$$\int_{\mathrm{T}(ijk)} \mathrm{d}^2 r \, \nabla u_{ijk} \cdot \nabla u_{jik} = -\frac{\mathbf{r}_{ik} \cdot \mathbf{r}_{jk}}{4\mathrm{A}_{ijk}} \tag{13.9}$$

où $\mathbf{r}_{ij} \coloneqq \mathbf{r}_i - \mathbf{r}_j$. Ces relations permettent de calculer les éléments de matrice du laplacien simplement en sommant les contributions des différentes faces :

$$\begin{aligned} \mathrm{D}^2_{ii} &= -\frac{1}{4} \sum_{a, i \in \mathrm{T}_a} \frac{\ell^2_{ia}}{\mathrm{A}_a} \\ \mathrm{D}^2_{ij} &= -\frac{\ell^2_{ij} - \ell^2_{ik} - \ell^2_{jk}}{8\mathrm{A}_{ijk}} - \frac{\ell^2_{ij} - \ell^2_{ik'} - \ell^2_{jk'}}{8\mathrm{A}_{jik'}} \end{aligned} \tag{13.10}$$

La notation utilisée est la suivante : ℓ_{ij} est la longueur du lien reliant les noeuds i et j ; ℓ_{ia}, a étant un indice de face, est la longueur du lien de la face a opposée au site i. Dans la deuxième équation, k et k' sont les noeuds qui complètent les faces (ijk) et (jik') situées de part et d'autre du lien (ij). Ainsi, en se référant à la figure 13.4, on a par exemple

$$\begin{aligned} \mathrm{D}^2_{00} &= -\frac{1}{4}\left(\frac{\ell^2_{12}}{\mathrm{A}_{012}} + \frac{\ell^2_{24}}{\mathrm{A}_{024}} + \frac{\ell^2_{45}}{\mathrm{A}_{045}} + \frac{\ell^2_{53}}{\mathrm{A}_{053}} + \frac{\ell^2_{31}}{\mathrm{A}_{031}} \right) \\ \mathrm{D}^2_{01} &= -\frac{\ell^2_{01} - \ell^2_{02} - \ell^2_{12}}{8\mathrm{A}_{012}} - \frac{\ell^2_{01} - \ell^2_{03} - \ell^2_{13}}{8\mathrm{A}_{103}} \end{aligned} \tag{13.11}$$

Problème 13.2 : Matrice de masse
L'objectif de ce problème est de calculer la matrice de masse $\mathrm{M}_{ij} = \langle u_i | u_j \rangle$ des fonctions tentes en dimension 2.

A Montrez que l'élément de matrice diagonal est

$$\langle u_i | u_i \rangle = \frac{1}{6} \sum_{a, i \in \mathrm{T}_a} \mathrm{A}_a \tag{13.12}$$

où la somme est effectuée sur les faces a qui touchent le site i.

B Montrez que l'élément de matrice entre sites voisins est

$$\langle u_i | u_j \rangle = \frac{1}{12}(\mathrm{A}_a + \mathrm{A}_b) \tag{13.13}$$

où a et b indexent les triangles situés de part et d'autre du lien ij.
Indice : procéder à un changement de variable pour effectuer les intégrales. Par exemple, pour trois noeuds 1, 2 et 3, il faut paramétrer les points à l'intérieur du triangle par les variables s et t telles que

$$\mathbf{r} = \mathbf{r}_1 + s(\mathbf{r}_2 - \mathbf{r}_1) + t(\mathbf{r}_3 - \mathbf{r}_1) \tag{13.14}$$

et intégrer dans le domaine approprié du plan (s,t), après avoir calculé le jacobien associé à ce changement de variables. Les propriétés de $\gamma(\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3)$ (linéarité, antisymétrie) contribuent à simplifier considérablement le calcul.

Problème 13.3 :
Démontrez les relations (13.10).

CHAPITRE 14

ÉQUATIONS AUX DÉRIVÉES PARTIELLES DÉPENDANT DU TEMPS

Nous allons traiter dans ce chapitre de la solution numérique des équations aux dérivées partielles qui impliquent une évolution temporelle, et donc qui requièrent des conditions initiales. Ce type de problème est différent des problèmes aux limites statiques, par exemple la solution de l'équation de Laplace, qui sont déterminées par des conditions aux frontières uniquement. Dans les problèmes dynamiques qui seront étudiés ici, nous aurons à la fois à tenir compte de conditions aux limites et de conditions initiales.

L'approche sera donc de considérer la représentation discrète d'un champ $\psi(\mathbf{r}, t)$ à un instant donné et de faire évoluer dans le temps cette représentation discrète. Celle-ci pourra être basée sur l'une des méthodes décrites au chapitre précédent : soit une grille régulière, une représentation par éléments finis ou par une méthode spectrale.

A Introduction

L'équation de diffusion, ou équation de la chaleur, prend la forme suivante :

$$\frac{\partial \psi}{\partial t} = \kappa \nabla^2 \psi \tag{14.1}$$

où κ est le coefficient de diffusion. Le champ ψ peut représenter la température locale à l'intérieur d'un matériau, ou la densité locale d'un soluté dans un solvant, etc. On peut justifier l'équation (14.1) de manière assez générale en supposant que la quantité ψ représente une quantité conservée (c'est-à-dire qui n'est pas localement créée) dont la diffusion d'un endroit à l'autre est caractérisée par une densité de courant $\mathbf{j}$ proportionnelle au gradient de ψ :

$$\mathbf{j} = -\kappa \nabla \psi \tag{14.2}$$

où le signe $-$ signifie que la quantité en question a tendance à se répartir là où elle est la plus faible. En combinant cette relation à l'équation de continuité

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \mathbf{j} = 0 \tag{14.3}$$

qui ne fait que représenter la conservation de la quantité en question, on retrouve bien l'équation de diffusion.

Dans le cas de la chaleur, l'équation (14.2) est l'expression de la loi de conduction linéaire de la chaleur (ou loi de Fourier) qui stipule que le courant de chaleur **q** (en W/m^2) est proportionnel au gradient de température :

$$\mathbf{q} = -k\nabla \mathrm{T} \qquad k : \text{conductivité thermique, en W/m.K} \tag{14.4}$$

La variation d'énergie interne par unité de volume (ΔQ) est proportionnelle à la variation de température :

$$\Delta \mathrm{Q} = c_p \rho \Delta \mathrm{T} \tag{14.5}$$

où c_p est la chaleur spécifique à pression constante et ρ la densité du milieu. Le courant de chaleur **q** étant associé à Q, le courant de température **j** sera donc proportionnel à **q** : $\mathbf{j} = \mathbf{q}/(c_p\rho)$. La loi de Fourier se réduit donc à la relation (14.2), avec $\kappa = k/(c_p\rho)$.

B Évolution directe en dimension un

Considérons l'équation de diffusion en une dimension d'espace, entre les frontières $x = 0$ et $x = \ell$. La méthode la plus simple pour résoudre l'équation numériquement est d'adopter une discrétisation uniforme de l'intervalle $[0, \ell]$, et une discrétisation également uniforme du temps, de sorte qu'on doit traiter l'évolution d'un vecteur ψ_r, en fait une suite $\psi_{r,n}$, où r est l'indice spatial et n l'indice temporel. En utilisant la valeur de la dérivée temporelle calculée au temps t_n et la méthode d'Euler pour l'évolution dans le temps, on obtient le système d'équations suivant :

$$\psi_{r,n+1} = \psi_{r,n} + \eta\left[\psi_{r+1,n} - 2\psi_{r,n} + \psi_{r-1,n}\right] \qquad \eta = \frac{h\kappa}{a^2} \tag{14.6}$$

où $h = \Delta t$ est le pas temporel et $a = \Delta x$ le pas spatial. On doit appliquer cette relation aux valeurs intérieures de ψ, et imposer les conditions aux limites en tout temps aux frontières :

$$\psi_{0,n} = \psi_{\text{gauche}} \qquad \psi_{\mathrm{N},n} = \psi_{\text{droite}} \tag{14.7}$$

Il faut également spécifier les conditions initiales $\psi_{r,0}$. Une fois cela fait, la solution est immédiate par récurrence : les valeurs $\psi_{r,n+1}$ sont données de manière explicite en fonction des $\psi_{r,n}$. Pour cette raison, cette méthode simple est qualifiée d'*explicite*.

B.1 Analyse de stabilité de von Neumann

La méthode explicite décrite ci-dessus court cependant le danger d'être instable. Von Neumann a étudié ce problème en supposant que la solution du problème discrétisé, qui est encore un problème linéaire, serait une combinaison de modes propres. Il a supposé une forme exponentielle pour les modes propres, qui seraient indexés par un nombre d'onde k, en fonction de la position et du temps $t = nh$:

$$\psi_{r,n} = \xi(k)^n \, \mathrm{e}^{ikar} \tag{14.8}$$

où ξ est un nombre complexe qui peut dépendre de k. En substituant cette forme dans l'équation discrète (14.6), on trouve

$$\xi(k) = 1 + 2\eta(\cos ka - 1) = 1 - 4\eta \sin^2(ka/2) \tag{14.9}$$

La solution numérique sera stable seulement si toutes les valeurs possibles de ξ respectent la condition $|\xi(k)| < 1$. Ceci n'est vrai, pour l'équation de diffusion, que si la condition suivante est respectée :

$$\eta < \frac{1}{2} \quad \text{ce qui revient à} \quad \kappa h < \frac{a^2}{2} \tag{14.10}$$

Autrement dit, le pas temporel doit être suffisamment petit en comparaison du pas spatial. Dans le cas de l'équation de diffusion, le pas temporel doit même varier en raison quadratique du pas spatial.

Problème 14.1 : Test de la condition de von Neumann
Écrivez un programme très simple qui permet de tester la condition de stabilité de von Neumann pour l'équation de la diffusion en une dimension. Posez $\ell = 1$ et les conditions aux limites $\psi(0) = 0$ et $\psi(\ell) = 1$.

C Méthode implicite de Crank-Nicholson

Le défaut principal de la méthode simple décrite à la section précédente est que la dérivée temporelle utilisée pour passer du temps t au temps $t + h$ est évaluée au temps t, c'est-à-dire au début de l'intervalle, comme dans la méthode d'Euler. Cette méthode est du premier ordre en h. La précision et la stabilité de la méthode sont grandement améliorées si la dérivée est estimée au milieu de l'intervalle.

Supposons que l'évolution temporelle ait la forme suivante :

$$\frac{\partial \psi}{\partial t} = \mathscr{L}\psi \tag{14.11}$$

où $\mathscr{L}$ est un opérateur différentiel linéaire, mais qui n'agit que sur la dépendance spatiale de ψ. La méthode simple décrite plus haut équivaut à la récurrence suivante :

$$\psi(t+h) = \psi(t) + h\mathscr{L}\psi(t) \tag{14.12}$$

Nous allons améliorer cette façon de faire en partant d'une valeur inconnue $\psi(t + h/2)$ évaluée au milieu de l'intervalle et en la propageant de $\Delta t = h/2$ vers $\psi(t + h)$ et de $\Delta t = -h/2$ vers $\psi(t)$:

$$\psi(t) = \left[1 - \tfrac{1}{2}h\mathscr{L}\right]\psi(t + h/2) \qquad \psi(t+h) = \left[1 + \tfrac{1}{2}h\mathscr{L}\right]\psi(t + h/2) \tag{14.13}$$

La première de ces équations est un système linéaire qui peut être résolu pour $\psi(t + h/2)$, connaissant $\psi(t)$. En pratique, nous aurons une version discrète du champ ψ et de l'opérateur $\mathscr{L}$, suite à une représentation du champ ψ selon l'une des méthodes décrites au chapitre précédent. L'équation ci-haut devient donc

$$\psi(t) = \left[1 - \tfrac{1}{2}h\mathrm{L}\right]\psi(t + h/2) \qquad \psi(t+h) = \left[1 + \tfrac{1}{2}h\mathrm{L}\right]\psi(t + h/2) \tag{14.14}$$

où ψ est maintenant un vecteur fini et L une matrice carrée. Une fois $\psi(t + h/2)$ connu par solution de la première équation, on applique la deuxième équation de manière directe pour obtenir $\psi(t+h)$.

La résolution de la première des équations (14.14) pour $\psi(t+h/2)$ est particulièrement simple en dimension 1 si on utilise une représentation par grille simple ou par éléments finis de la fonction ψ, car la matrice L est alors tridiagonale si $\mathscr{L}$ contient des dérivées du deuxième ordre ou moins. Un tel système s'inverse en un temps d'ordre $\mathscr{O}(\mathrm{N})$, N étant le nombre de points sur la grille. Par contre, une représentation spectrale demanderait d'inverser une matrice pleine. Si l'équation différentielle est linéaire et homogène dans le temps, et que le pas h est constant, alors le calcul d'une seule matrice inverse est suffisant pour tout le calcul. Si l'équation est non linéaire, alors l'opérateur L dépend de ψ et l'inversion doit être faite à chaque étape $t \to t+h$.

C.1 Analyse de stabilité

Supposons pour simplifier les choses que l'équation différentielle soit linéaire et homogène dans le temps, ce qui revient à dire que la matrice L ci-haut est constante. Dans ce cas, l'évolution du système du temps t au temps $t+h$ se fait par application d'une matrice d'évolution constante U :

$$\psi(t+h) = \mathrm{U}\psi(t) \quad \text{où} \quad \mathrm{U} \coloneqq \frac{1+\frac{1}{2}h\mathrm{L}}{1-\frac{1}{2}h\mathrm{L}} \tag{14.15}$$

La stabilité de cette évolution est régie par les valeurs propres de l'opérateur U : si au moins une valeur u de U est telle que $|u|>1$, alors la méthode est instable, car la moindre composante de ψ le long du vecteur propre correspondant va être amplifiée par l'évolution temporelle. Si λ désigne une valeur propre de l'opérateur L, alors la valeur propre correspondante de U est simplement

$$u = \frac{1+\frac{1}{2}h\lambda}{1-\frac{1}{2}h\lambda} \tag{14.16}$$

car U est une fonction directe de la matrice L. La condition de stabilité $|u| \leq 1$ se traduit donc par

$$-1 < \frac{1+\frac{1}{2}h\lambda}{1-\frac{1}{2}h\lambda} < 1 \implies \tfrac{1}{2}h\lambda < 0 \tag{14.17}$$

Dans le cas particulier de l'équation de la chaleur, les valeurs propres λ du laplacien D^2 étant toujours négatives, la stabilité de la méthode devrait être assurée. Bien sûr, ce sont les valeurs propres de L qui comptent et non celles de $\mathscr{L}$; il faut donc que le nombre N de points de grille soit suffisant pour que la condition $\lambda < 0$ soit respectée pour L, ce qui est plausible quand N est suffisamment grand.

D Méthode du saute-mouton

Une façon alternative de procéder à l'évolution temporelle sans avoir à résoudre le système implicite (14.14) tout en conservant une précision du deuxième ordre en h est la méthode dite du «saute-mouton» (angl. *leapfrog*). Il s'agit simplement de calculer la dérivée au temps t à l'aide de la fonction (inconnue) au temps $t+h$ et de la fonction (conservée en mémoire) au temps $t-h$. Autrement dit :

$$\mathscr{L}\psi(t) = \frac{\partial\psi}{\partial t} \approx \frac{1}{2h}\big[\psi(t+h) - \psi(t-h)\big] \tag{14.18}$$

ce qui mène à une expression *explicite* pour $\psi(t+h)$:

$$\psi(t+h) = \psi(t-h) + 2h\mathscr{L}\psi(t) \tag{14.19}$$

Cette approche requiert de garder en mémoire, à chaque étape, la valeur du champ ψ à trois temps différents : elle est plus coûteuse en mémoire, mais potentiellement plus rapide qu'une méthode implicite dans les cas où le système (14.14) doit être résolu à chaque étape, surtout en dimensions supérieures quand les matrices en jeu ne sont pas tridiagonales.

instabilité de la méthode pour l'équation de diffusion Appliquons à cette méthode la même analyse de von Neumann que nous avons appliquée à la méthode directe, c'est-à-dire en nous basant sur les différences finies dans l'espace, pour l'équation de la diffusion. L'équation aux différences est

$$\psi_{r,n+1} = \psi_{r,n-1} + 2\eta\big[\psi_{r+1,n} - 2\psi_{r,n} + \psi_{r-1,n}\big] \qquad \eta = \frac{h\kappa}{a^2} \tag{14.20}$$

En posant $\psi_{r,n} = \xi(k)^n\, \mathrm{e}^{ikar}$, on trouve la relation

$$\xi(k) = \xi^{-1}(k) + 4\eta(\cos ka - 1) \Longrightarrow \xi^2 - 1 = 4\eta\xi(\cos ka - 1) \tag{14.21}$$

Il s'agit d'une équation quadratique en ξ, dont la solution est

$$\xi = 2\eta(\cos ka - 1) \pm \sqrt{1 + [2\eta(\cos ka - 1)]^2} = \mathrm{A} \pm \sqrt{1+\mathrm{A}^2} \quad \text{où} \quad \mathrm{A} \equiv 2\eta(\cos ka - 1) \tag{14.22}$$

La condition de stabilité $|\xi| < 1$ revient donc à demander

$$\mathrm{A} - \sqrt{1+\mathrm{A}^2} > -1 \quad \text{et} \quad \mathrm{A} + \sqrt{1+\mathrm{A}^2} < 1 \tag{14.23}$$

La première condition n'est pas respectée si $\mathrm{A} < 0$, ce qui est toujours le cas. La méthode du saute-mouton, en dépit d'être du deuxième ordre en temps, est donc instable, même pour des h très petits, pour l'équation de diffusion. Mais elle peut l'être pour d'autres équations différentielles.

L'équation d'onde Considérons à cet effet l'équation d'onde :

$$\frac{\partial^2\psi}{\partial t^2} - v^2\frac{\partial^2\psi}{\partial x^2} = 0 \tag{14.24}$$

où v est la vitesse de propagation. Exprimons cette équation dans un formalisme du premier ordre dans le temps, en définissant $\phi = \dfrac{\partial \psi}{\partial t}$:

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= v^2 \frac{\partial^2 \psi}{\partial x^2} \\ \frac{\partial \psi}{\partial t} &= \phi \end{aligned} \tag{14.25}$$

Sur une grille régulière dans le temps et dans l'espace, les équations de récurrence correspondant à la méthode du saute-mouton dans ce cas sont les suivantes :

$$\begin{aligned} \phi_{r,n+1} &= \phi_{r,n} + \frac{h v^2}{a^2}\left[\psi_{r+1,n} - 2\psi_{r,n} + \psi_{r-1,n}\right] \\ \psi_{r,n+1} &= \psi_{r,n} + h\phi_{r,n+1} \end{aligned} \tag{14.26}$$

Notez que ψ est évolué du temps n au temps $n+1$ à l'aide de ϕ au temps $n+1$. C'est ce détail qui fait de cette relation une application de la méthode du saute-mouton.

stabilité de la méthode Procédons maintenant à une analyse de la stabilité de cette procédure. Posons

$$\phi_{r,n} = \phi_0 \xi^n \mathrm{e}^{ikar} \qquad \psi_{r,n} = \psi_0 \xi^n \mathrm{e}^{ikar} \tag{14.27}$$

(il est important que le même ξ et le même k s'appliquent aux deux variables, mais que leurs amplitudes puissent être différentes). En substituant cette forme dans la relation d'évolution ci-dessus, on trouve

$$\begin{aligned} \xi\phi_0 &= \phi_0 - \frac{4h v^2}{a^2}\psi_0 \sin^2(ka/2) \\ \xi\psi_0 &= \psi_0 + h\xi\phi_0 \end{aligned} \tag{14.28}$$

Cette relation peut aussi s'écrire ainsi, sous forme matricielle :

$$\begin{pmatrix} 1-\xi & -\dfrac{4h v^2}{a^2}\sin^2(ka/2) \\ h\xi & 1-\xi \end{pmatrix} \begin{pmatrix} \phi_0 \\ \psi_0 \end{pmatrix} = 0 \tag{14.29}$$

Les valeurs possibles de ξ se trouvent donc en annulant le déterminant de cette matrice, ce qui mène à la condition suivante :

$$(1-\xi)^2 = -\frac{4h^2 v^2}{a^2}\xi \sin^2(ka/2) \tag{14.30}$$

La procédure est instable si $\xi^2 > 1$. La limite de stabilité peut donc être obtenue en posant $\xi = \pm 1$. La limite $\xi = 1$ n'est atteinte que si $k = 0$, ce qui n'est pas menaçant car cela correspond à une solution uniforme. La limite $\xi = -1$ correspond à la condition

$$\frac{h^2 v^2}{a^2}\sin^2(ka/2) = 1 \tag{14.31}$$

Cette limite est impossible à atteindre si $a > v\,h$. Cette condition est appelée *condition de Courant*[1] La procédure sera stable si cette condition est respectée. Elle correspond physiquement à l'impossibilité pour des conditions situées à plus d'un pas de réseau d'une position donnée de se transmettre causalement (à la vitesse v) du temps n au temps $n+1$.

1. D'après Richard Courant, l'un des pionniers de la méthode des éléments finis dans la première moitié du XX$^{\text{e}}$siècle.

E Application basée sur une représentation spectrale

Le code déployé ci-dessous utilise la représentation spectrale d'une fonction en dimension 1, dans l'intervalle $[-1,1]$, et résout l'équation de diffusion en fonction du temps, en affichant la solution de manière interactive à l'aide de gnuplot. Le schéma de Crank-Nicholson est utilisé pour propager la solution d'un temps à l'autre. Un opérateur d'évolution U, indépendant du temps, est construit au début du calcul et est ensuite appliqué à la solution à chaque instant t pour la propager vers l'instant suivant $(t+h)$.

La grille de Gauss-Lobatto est utilisée dans le calcul, mais seuls les points intérieurs doivent être propagés. Précisons ce point : dans la méthode de collocation, l'équation $\dot{\psi} = \mathscr{L}\psi$ doit être imposée aux points intérieurs seulement. En fonction des sous-vecteurs ψ^{F} et ψ^{I} représentant respectivement les valeurs aux extrémités et à l'intérieur du domaine, la version discrétisée de l'équation prend la forme

$$\frac{\partial}{\partial t}\begin{pmatrix}\psi^{\mathrm{F}}\\ \psi^{\mathrm{I}}\end{pmatrix} = \begin{pmatrix}\mathrm{L}^{\mathrm{F}} & \mathrm{L}^{\mathrm{FI}}\\ \mathrm{L}^{\mathrm{IF}} & \mathrm{L}^{\mathrm{I}}\end{pmatrix}\begin{pmatrix}\psi^{\mathrm{F}}\\ \psi^{\mathrm{I}}\end{pmatrix} \quad \text{ou encore} \quad \frac{\partial \psi^{\mathrm{I}}}{\partial t} = \mathrm{L}^{\mathrm{I}}\psi^{\mathrm{I}} + \mathrm{L}^{\mathrm{IF}}\psi^{\mathrm{F}} \tag{14.32}$$

où seule la deuxième rangée de l'équation matricielle est appliquée, ψ^{F} étant fixé par les conditions aux limites, indépendamment du temps.

Supposons maintenant que les conditions aux limites sont homogènes, c'est-à-dire $\psi^{\mathrm{F}} = 0$. On retrouve alors la même forme que l'équation différentielle originale obtenue sans tenir compte des conditions aux limites, mais pour les points intérieurs seulement. La méthode de Crank-Nicholson implique donc l'opérateur d'évolution suivant :

$$\mathrm{U} = \frac{1+\frac{1}{2}h\mathrm{L}^{\mathrm{I}}}{1-\frac{1}{2}h\mathrm{L}^{\mathrm{I}}} \tag{14.33}$$

qui doit être construit et appliqué sur le vecteur ψ^{I} de manière répétée.

Si les conditions aux limites ne sont pas homogènes, alors on peut sans trop de peine ramener le problème à celui de conditions aux limites homogènes de la manière suivante : on pose

$$\psi(x,t) = \phi(x,t) + \psi_s(x) \tag{14.34}$$

où par définition $\psi_s(x)$ est une solution statique (indépendante du temps) à l'équation différentielle, et qui en plus respecte les bonnes conditions aux limites. Comme l'équation est linéaire, on a donc

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \mathscr{L}\phi \quad \text{où} \quad \phi^{\mathrm{F}} = 0\\ \mathscr{L}\psi_s &= 0 \quad \text{où} \quad \psi_s^{\mathrm{F}} = \psi^{\mathrm{F}} \end{aligned} \tag{14.35}$$

La solution au problème indépendant du temps $\mathscr{L}\psi_s = 0$ est généralement simple. Dans le cas de l'équation de diffusion, $\partial_x^2 \psi_s = 0$ et donc $\psi_s(x)$ est une fonction linéaire qui interpole entre les valeurs aux extrémités de l'intervalle. Cette fonction est en fait la valeur asymptotique de $\psi(x,t)$ dans la limite $t \to \infty$. Il nous reste alors à déterminer numériquement $\phi(x,t)$, c'est-à-dire à résoudre un problème aux limites homogènes.

F Équation de Schrödinger dépendant du temps

Dans cette section nous nous intéresserons à résoudre l'équation de Schrödinger dépendant du temps dans un potentiel unidimensionnel. Si $\psi(x,t)$ est la fonction d'onde, alors son évolution dans le temps est régie par l'équation suivante :

$$i\frac{\partial\psi}{\partial t} = \hat{\mathrm{H}}\psi \qquad\qquad \hat{\mathrm{H}} = \hat{\mathrm{K}} + \hat{\mathrm{V}} \tag{14.36}$$

où $\hat{\mathrm{K}}$ est l'opérateur d'énergie cinétique et $\hat{\mathrm{V}}$ l'opérateur d'énergie potentielle (nous adoptons des unités telles que $\hbar = 1$). Dans la représentation X, l'opérateur $\hat{\mathrm{V}}$ est diagonal : $\langle x|\hat{\mathrm{V}}|x'\rangle = \mathrm{V}(x)\delta(x-x')$. Dans la représentation P, l'opérateur $\hat{\mathrm{K}}$ est diagonal : $\langle k|\hat{\mathrm{K}}|k'\rangle = (k^2/2m)\delta(k-k')$.

L'évolution temporelle de la fonction d'onde $\psi(x,t)$ peut s'accomplir via l'opérateur d'évolution :

$$\psi(x,t) = \mathrm{e}^{-i\hat{\mathrm{H}}t}\psi(x,0) \tag{14.37}$$

Il est simple, dans la représentation P, de calculer l'exponentielle $\mathrm{e}^{-i\hat{\mathrm{K}}t}$, comme il est simple, dans la représentation X, de calculer l'exponentielle $\mathrm{e}^{-i\hat{\mathrm{V}}t}$, car les exposants sont diagonaux dans chaque cas. Par contre, il est très difficile de calculer l'exponentielle $\mathrm{e}^{-i(\hat{\mathrm{K}}+\hat{\mathrm{V}})t}$, car les opérateurs $\hat{\mathrm{K}}$ et $\hat{\mathrm{V}}$ ne commutent pas.

En effet, si A et B sont deux matrices (ou opérateurs abstraits), on doit multiplier les exponentielles selon la formule de Campbell-Baker-Hausdorff :

$$\mathrm{e}^{\mathrm{A}}\mathrm{e}^{\mathrm{B}} = \mathrm{e}^{\mathrm{C}} \qquad \mathrm{C} = \mathrm{A}+\mathrm{B}+\frac{1}{2}[\mathrm{A},\mathrm{B}]+\frac{1}{12}([\mathrm{A},[\mathrm{A},\mathrm{B}]]+[\mathrm{A},[\mathrm{B},\mathrm{B}]])+\cdots \tag{14.38}$$

La matrice C est exprimée comme une série infinie en fonction de commutateurs imbriqués de A avec B, chaque terme comportant un niveau de commutation de plus que le précédent. Si le commutateur [A, B] est constant, cette série ne comporte que les trois premiers termes. Si [A, [A, B]] est constant, elle s'arrête aux termes explicitement décrits ci-dessus, et ainsi de suite.

La stratégie d'approximation que nous allons appliquer est de procéder à une évolution temporelle en N étapes :

$$\mathrm{e}^{-i\hat{\mathrm{H}}t} = \left(\mathrm{e}^{-i\hat{\mathrm{H}}h}\right)^{\mathrm{N}} \qquad (t = \mathrm{N}h) \tag{14.39}$$

À chaque étape, nous devons calculer $\mathrm{e}^{(\mathrm{A}+\mathrm{B})h}$, où $\mathrm{A} = -i\hat{\mathrm{K}}$ et $\mathrm{B} = -i\hat{\mathrm{V}}$. D'après ce que nous avons dit plus haut,

$$\mathrm{e}^{(\mathrm{A}+\mathrm{B})h} = \mathrm{e}^{\mathrm{A}h}\,\mathrm{e}^{\mathrm{B}h}\,\mathrm{e}^{\mathcal{O}(h^2)} \tag{14.40}$$

Cela peut aussi se démontrer en développant les exponentielles en séries jusqu'à l'ordre h^3. Cette décomposition de l'exponentielle d'une somme en un produit d'exponentielles de matrices non commutantes porte le nom de *décomposition de Trotter-Suzuki*. Une version plus précise de cette décomposition est la suivante :

$$\mathrm{e}^{(\mathrm{A}+\mathrm{B})h} = \mathrm{e}^{\mathrm{B}h/2}\,\mathrm{e}^{\mathrm{A}h}\,\mathrm{e}^{\mathrm{B}h/2}\,\mathrm{e}^{\mathcal{O}(h^3)} \tag{14.41}$$

Nous allons donc faire évoluer la fonction d'onde sur un court pas temporel h, en mettant à profit la relation (14.40) :

$$\psi(x,t+h) = \mathrm{e}^{-i\hat{\mathrm{K}}h}\,\mathrm{e}^{-i\hat{\mathrm{V}}h}\psi(x,t) \tag{14.42}$$

Nous allons adopter la représentation X, donc une description de ψ en fonction de x, à l'aide d'une grille régulière de pas a qui s'étale dans l'espace de –L à L. Dans cette représentation, l'opérateur $\hat{\mathrm{V}}$ est diagonal et son action est simplement

$$\psi(x_i,t) \to \mathrm{e}^{-i\mathrm{V}(x_i)h}\psi(x_i,t) \tag{14.43}$$

Il suffit pour l'accomplir de stocker la diagonale $\mathrm{e}^{-i\mathrm{V}(x_i)h}$ après l'avoir calculée au début du programme. Afin d'utiliser la même procédure pour l'opérateur $\hat{\mathrm{K}}$, nous allons procéder à une transformée de Fourier rapide avant l'application de $\hat{\mathrm{K}}$, et à une transformée de Fourier rapide inverse juste après. Schématiquement, l'évolution sur un pas temporel h prendra donc la forme suivante :

1. $\psi(x_i,t)=\psi_i \to \mathrm{e}^{-i\mathrm{V}(x_i)h}\psi_i$
2. $\psi_i \to \mathrm{TdF}(\psi_i)$
3. $\psi_i \to \mathrm{e}^{-ihk_i^2/2m}\psi_i$
4. $\psi_i \to \mathrm{TdFi}(\psi_i)=\psi(x_i,t+h)$

où TdF représente la transformée de Fourier et TdFi la transformée de Fourier inverse.

En pratique, il faut remplacer k_i^2 par $2(1-\cos(ak_i))/a^2$, où a est le pas de grille, afin d'avoir une opération périodique en k_i, sinon on s'expose à un comportement incorrect aux grandes valeurs de k_i.

Cette façon de procéder à l'évolution temporelle est de complexité $\mathrm{N}\ln\mathrm{N}$ et est relativement efficace. L'erreur commise est d'ordre h^2 et, ce qui est plus important, l'évolution est unitaire.

Si on voulait appliquer la décomposition plus précise (14.41), le coût serait à peine plus élevé :

1. $\psi(x_i,t)=\psi_i \to \mathrm{e}^{-i\mathrm{V}(x_i)h/2}\psi_i$
2. $\psi_i \to \mathrm{TdF}(\psi_i)$
3. $\psi_i \to \mathrm{e}^{-ihk_i^2/2m}\psi_i$
4. $\psi_i \to \mathrm{TdFi}(\psi_i)$
5. $\psi_i \to \mathrm{e}^{-i\mathrm{V}(x_i)h/2}\psi_i=\psi(x_i,t+h)$

La seule différence dans ce dernier algorithme et qu'on applique l'énergie potentielle en deux temps consécutifs, ce qui ne demande que peu de ressources supplémentaires, et qu'on 'mesure' $\psi(x_i,t+h)$ entre ces deux applications de $\mathrm{e}^{-i\mathrm{V}(x_i)h/2}$.

G Propagation d'une onde et solitons

G.1 Équation d'advection

L'équation la plus simple décrivant la propagation d'une onde est l'*équation d'advection* en une dimension d'espace :

$$\frac{\partial\psi}{\partial t}+v\frac{\partial\psi}{\partial x}=0 \tag{14.44}$$

Cette équation est un cas particulier de l'équation de continuité

$$\frac{\partial\psi}{\partial t}+\nabla\cdot\mathbf{j}=0 \quad \text{où} \quad \mathbf{j}=(v\psi,0,0) \tag{14.45}$$

Le champ ψ représente donc la densité d'une quantité conservée, mais qui ne peut se propager que vers la droite (si $v > 0$).

La solution analytique de l'équation d'advection est très simple : on procède au changement de variables

$$\xi = x + v\,t \qquad \eta = x - v\,t \tag{14.46}$$

de sorte que

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \xi} + \frac{\partial}{\partial \eta} \qquad \frac{\partial}{\partial t} = v\frac{\partial}{\partial \xi} - v\frac{\partial}{\partial \eta} \tag{14.47}$$

L'équation d'advection s'écrit comme suit en fonction des variables (ξ, η) :

$$\frac{\partial \psi}{\partial \xi} = 0 \quad \text{et donc} \quad \psi(x,t) = \psi_0\, u(\eta) = \psi_0\, u(x - v\,t) \tag{14.48}$$

où $u(\eta)$ est une fonction différentiable quelconque, représentant la forme du paquet d'ondes qui se propage vers la droite.

G.2 Équation de Korteweg-de Vries

L'étude de la propagation des vagues – l'archétype des ondes dans l'histoire des sciences – est un sujet relativement complexe. La vague elle-même est une interface entre deux milieux (le liquide en bas, l'air ou le vide en haut) dont la forme varie en fonction du temps. La manière dont cette interface se déplace repose sur une description de l'écoulement du fluide sous-jacent, par les équations standards de l'hydrodynamique (éq. de Navier-Stokes et de continuité).

L'étude théorique et expérimentale de la propagation des vagues a été motivée en bonne partie par l'observation en 1834, par l'ingénieur naval John Scott Russell, de la propagation d'une vague singulière dans un canal en Écosse :

> I was observing the motion of a boat which was rapidly drawn along a narrow channel by a pair of horses, when the boat suddenly stoppednot so the mass of water in the channel which it had put in motion; it accumulated round the prow of the vessel in a state of violent agitation, then suddenly leaving it behind, rolled forward with great velocity, assuming the form of a large solitary elevation, a rounded, smooth and well-defined heap of water, which continued its course along the channel apparently without change of form or diminution of speed. I followed it on horseback, and overtook it still rolling on at a rate of some eight or nine miles an hour, preserving its original figure some thirty feet long and a foot to a foot and a half in height. Its height gradually diminished, and after a chase of one or two miles I lost it in the windings of the channel. Such, in the month of August 1834, was my first chance interview with that singular and beautiful phenomenon which I have called the Wave of Translation.

Russell procéda par la suite à une étude contrôlée de la propagation des vagues dans un petit bassin artificiel qu'il fit creuser chez lui. L'une des découvertes qualitatives qu'il fit est que la vitesse de propagation augmente avec la profondeur du bassin (pourvu qu'il ne soit pas trop profond). Donc, si ψ représente la hauteur de la vague, la vitesse est proportionnelle à ψ.

Les calculs théoriques du Français Boussinesq, et ensuite des Hollandais Korteweg et de Vries, on mené à la forme simplifiée suivante pour l'équation décrivant la propagation d'une onde en une dimension, dans un canal étroit et plat :

$$\frac{\partial \psi}{\partial t} + \epsilon\psi\frac{\partial \psi}{\partial x} + \mu\frac{\partial^3 \psi}{\partial x^3} = 0 \tag{14.49}$$

C'est l'*équation de Korteweg-de Vries* (ou KdeV). Elle diffère de l'équation d'advection de deux manières :

1. La vitesse de propagation dépend de l'amplitude de l'onde : v a été remplacé par $\epsilon\psi$.
2. Un terme de dispersion a été ajouté, proportionnel à la dérivée troisième de ψ. Ce terme provoque la dispersion – l'étalement d'un paquet d'ondes initial – car la vitesse de propagation dépend alors du nombre d'onde, et donc n'est pas constante pour une onde non sinusoïdale.

L'équation d'advection est retrouvée si on néglige la dispersion, et si on linéarise le terme non linéaire : on pose $\psi = \psi_0 + \delta\psi$, ψ_0 étant l'élévation moyenne de la surface, et on néglige les termes quadratiques en $\delta\psi$. On retrouve alors l'équation d'advection pour la déviation $\delta\psi$ par rapport à l'élévation moyenne.

Il est pratique de conserver les paramètres ϵ et μ dans l'étude de cette équation, car on peut alors facilement retrouver diverses limites. Par exemple, en posant $\mu = 0$, on trouve l'*équation de Burgers*, qui constitue un modèle simple des ondes de choc. En supposant cependant que ϵ et μ sont non nuls, il est toujours possible de procéder à un changement d'échelle de x pour fixer μ à l'unité, et à un changement d'échelle de ψ (l'équation étant non linéaire) pour fixer $\epsilon = 6$. On obtient ainsi une forme normalisée de l'équation de KdeV :

$$\frac{\partial\psi}{\partial t} + \frac{\partial^3\psi}{\partial x^3} + 6\psi\frac{\partial\psi}{\partial x} = 0 \tag{14.50}$$

G.3 Solitons

Nous allons maintenant démontrer que l'équation de KdeV (14.49) admet comme solution particulière des ondes d'étendue finie qui se propagent sans déformation, ou *solitons* (parfois aussi appelée *ondes solitaires*). Celles-ci correspondent à l'onde de translation (*wave of translation*) observée par Scott Russell.

Commençons par supposer une solution de la forme $\psi(x,t) = u(x - v\,t)$ et substituons dans l'équation de KdeV; on obtient une équation différentielle ordinaire en fonction de $\eta = x - v\,t$, ou en fonction de x si on pose $t = 0$:

$$-v\,u' + u''' + 6\,u\,u' = 0 \quad \text{ou} \quad -v\,u' + u''' + 3(u^2)' = 0 \tag{14.51}$$

ce qui peut s'intégrer une fois par rapport à x pour donner

$$-v\,u + u'' + 3\,u^2 = \mathrm{A} \tag{14.52}$$

où A est une constante d'intégration. Si on suppose que la solution u tend vers zéro en même temps que ses dérivées (loin du maximum du paquet d'onde), on doit poser $\mathrm{A} = 0$. En multipliant par u', qui sert de facteur intégrant, on trouve

$$-v\,u\,u' + u'\,u'' + 3\,u'\,u^2 = 0 = \frac{\mathrm{d}}{\mathrm{d}x}\left[-\frac{1}{2}v\,u^2 + \frac{1}{2}(u')^2 + u^3\right] \tag{14.53}$$

Donc l'expression entre crochets est indépendante de x, et doit être nulle quand $x \to \pm\infty$. On peut alors isoler u' :

$$-\frac{1}{2}v\,u^2 + \frac{1}{2}(u')^2 + u^3 = 0 \implies \frac{\mathrm{d}u}{\mathrm{d}x} = u\sqrt{v - 2u} \tag{14.54}$$

ce qui nous permet d'intégrer :

$$x = \int \frac{\mathrm{d}u}{u\sqrt{v-2u}} = \frac{2}{\sqrt{v}} \operatorname{arctanh} \sqrt{1-\frac{2u}{v}} \tag{14.55}$$

En isolant u, on trouve

$$u(x) = \frac{v}{2}\left[1-\tanh^2\left(\frac{\sqrt{v}\,x}{2}\right)\right] = \frac{v/2}{\cosh^2\left(\frac{\sqrt{v}\,x}{2}\right)} \tag{14.56}$$

Nous avons donc trouvé une solution à l'équation de KdeV de la forme suivante :

$$\psi(x,t) = \frac{v/2}{\cosh^2\left[\frac{\sqrt{v}}{2}(x-v\,t)\right]} \tag{14.57}$$

Notons que l'amplitude du soliton est proportionnelle à sa vitesse.

Problème 14.2 : Mise à l'échelle des solitons
On peut écrire la solution solitonique comme suit :

$$\psi(x,t) = \frac{\mathrm{A}}{\cosh^2[(x-v\,t)/\sigma]} \tag{14.58}$$

où A est l'amplitude maximale du soliton et σ sa largeur caractéristique. En partant de la solution (14.57) à l'équation (14.50), écrivez la solution correspondante à l'équation (14.49) sous la forme ci-haut en procédant aux transformations d'échelle nécessaires. Exprimez A, σ en fonction de ϵ, μ et de la vitesse du soliton.

Solution

La transformation $x = a\,x'$ et $\psi = b\,\psi'$ ramène l'équation (14.50) à la forme

$$b\frac{\partial\psi'}{\partial t} + \frac{b}{a^3}\frac{\partial^3\psi'}{\partial x'^3} + 6\frac{b^2}{a}\psi'\frac{\partial\psi'}{\partial x'} = 0$$

ou

$$\frac{\partial\psi'}{\partial t} + \frac{1}{a^3}\frac{\partial^3\psi'}{\partial x'^3} + 6\frac{b}{a}\psi'\frac{\partial\psi'}{\partial x'} = 0$$

d'où la correspondance $\mu = 1/a^3$ et $\epsilon = 6b/a$. La solution (14.57) s'exprime alors comme suit :

$$\psi'(x,t) = \frac{v/2b}{\cosh^2\left[\frac{\sqrt{v}}{2}(a\,x'-v\,t)\right]}$$

En définissant $v' = v/a$, on trouve

$$\psi'(x,t) = \frac{a v'/2b}{\cosh^2\left[\dfrac{\sqrt{a^3 v'}}{2}(x' - v' t)\right]} = \frac{3v'/\epsilon}{\cosh^2\left[\dfrac{1}{2}\sqrt{\dfrac{v'}{\mu}}(x' - v' t)\right]}$$

Donc

$$\sigma = 2\sqrt{\frac{\mu}{v'}} \qquad \text{A} = \frac{3v'}{\epsilon} = \frac{12\mu}{\epsilon\sigma^2}$$

CHAPITRE 15

NOMBRES ALÉATOIRES

A Générateurs d'entier aléatoires

Les méthodes stochastiques reposent toutes sur la possibilité de générer des nombres de manière aléatoire, c'est-à-dire des séquences de nombres entiers pris au hasard dans un intervalle donné. Que veut-on dire par là précisément ? Premièrement, remarquons qu'un nombre n'est pas aléatoire en soi : le qualificatif s'applique à une suite infinie de nombres, chacun étant compris entre 0 et M (M peut être très grand, par exemple 2^{64}, ou simplement M = 2 pour une production de bits aléatoires). Le caractère aléatoire est fondamentalement lié à l'impossibilité de prédire quel sera le nombre suivant de la suite, ou de déceler des corrélations significatives entre les membres différents de la suite. Par exemple, une suite de nombres aléatoires sera *incompressible*, au sens informatique du terme : aucun algorithme de compression ne pourrait y être appliqué avec un gain supérieur à un dans la limite d'une suite infinie.

La façon idéale de générer une séquence aléatoire est d'avoir recours à un processus physique fondamentalement aléatoire, gouverné par les lois de la mécanique quantique ou statistique. Un dispositif produisant des bits aléatoires liés à un processus optique (le passage ou non d'un photon au travers d'un miroir semi-transparent) existe sur le marché [1], mais est généralement trop lent pour les besoins du calcul. Un dispositif plus efficace et suffisamment rapide, basé sur le bruit électronique dans un circuit comportant une jonction tunnel, est en instance de brevet à l'Université de Sherbrooke. [2] En pratique, il est plus économique et simple d'utiliser des générateurs de nombres *pseudo-aléatoires*, même si ceux-ci ne sont pas parfaitement aléatoires, car étant basés sur un algorithme déterministe. Par contre, pour des fins de vérification et de développement, il est utile de disposer d'un générateur de nombres pseudo-aléatoires produisant la même séquence sur demande.

Un générateur pseudo-aléatoire est une contradiction dans les termes : il s'agit d'une méthode *déterministe* pour générer une suite de nombres qui se comporte pratiquement comme une suite réellement aléatoire. En particulier, chaque nombre de la suite est déterminé de manière unique par un ou quelques-uns des nombres qui le précèdent dans la suite, mais la loi déterministe est «non naturelle» et n'a pratiquement aucune chance d'avoir un effet sur le calcul, si on la compare à un processus réellement aléatoire.

1. Voir par exemple http://www.idquantique.com/
2. Dans le groupe de recherche du Prof. Bertrand Reulet.

A.1 Générateur à congruence linéaire

La manière classique de générer des entiers aléatoires est la relation de récurrence suivante, dite à *congruence linéaire* :

$$x_{n+1} = (a\,x_n + c) \mod \mathrm{M} \tag{15.1}$$

où les entiers a et c doivent être choisis judicieusement, ainsi qu'un premier entier non nul dans la séquence. La séquence définie par (15.1) est périodique, et sa période est au plus égale à M, car la séquence se répète exactement après M nombres. En effet, le nombre suivant est déterminé par le nombre courant, et au plus M possibilités existent. Dans le meilleur des cas, si les entiers a et c sont correctement choisis, la période est égale à M, sinon elle est égale à un diviseur de M.

On montre le théorème suivant :

Théorème 15.1 Hull-Dobell
Si l'incrément c est non nul, alors la relation $x_{n+1} = (a\,x_n + c) \mod \mathrm{M}$ donne une séquence de période M si et seulement si les trois conditions suivantes sont respectées :

1. c et M n'ont pas de facteur commun
2. $a-1$ est divisible par tous les facteurs premiers de M
3. $a-1$ est un multiple de 4 si M est un multiple de 4.

Par contre, la combinaison $a = 16807$, $c = 0$ et $\mathrm{M} = 2^{31}-1$ est un choix courant (le théorème de Hull-Dobell ne s'applique pas dans ce cas, mais la séquence est tout de même de période M).

Les générateurs simples de ce type ont longtemps été la règle. Ils sont à proscrire absolument dans toute application ou la qualité de la séquence aléatoire est importante. En particulier, sur une machine moderne, la séquence complète peut être générée en quelques secondes, ce qui démontre clairement son insuffisance pour les calculs sérieux. En fait, on montre que si on considère des multiplets à k composantes formés des nombres de cette séquence, ces multiplets formeront au plus $\mathrm{M}^{1/k}$ plans dans l'espace $\mathbb{R}^k$. En trois dimensions, si $\mathrm{M} \sim 2^{32}$, cela représente $\sim 1\,600$ plans, et moins que cela si a et c ne sont pas bien choisis. Un générateur célèbre, nommé RANDU et populaire dans les années 1970, était basé sur $a = 65539$ et $\mathrm{M} = 2^{31}$; ce choix d'entiers était très mauvais et de nombreux résultats publiés ont été remis en question en raison de la mauvaise qualité de RANDU.

A.2 Générateurs de Fibonacci

Cette méthode est basée sur la récurrence suivante :

$$x_{n+1} = x_{n-p} + x_{n-q} \mod \mathrm{M} \tag{15.2}$$

On doit spécifier les entiers p, q et M et fournir les premiers éléments de la séquence par une autre méthode (par exemple une congruence linéaire). Parmi les choix acceptables de (p,q), signalons

$$(607,273) \qquad (2281,1252) \qquad (9689,5502) \qquad (44497,23463) \tag{15.3}$$

Des générateurs plus avancés encore sont obtenus en se basant sur la méthode de Fibonacci et en ajoutant un mélange des bits du nombre à chaque étape.

B Générateurs de distributions continues

B.1 Distribution uniforme

À partir d'un générateur de nombres aléatoires entiers, compris entre 0 et M−1, on peut générer des nombres aléatoires à virgule flottante compris entre 0 et 1 simplement en divisant par M (au sens des NVF). Ces nombres aléatoires suivent une distribution de probabilité uniforme : $p(x) = 1$ dans l'intervalle $x \in [0,1]$. Le fait que la distribution soit uniforme est précisément l'une des conditions qui caractérisent un bon générateur de nombres aléatoires. Plusieurs tests statistiques de la qualité des générateurs pseudo-aléatoires peuvent être appliqués pour s'en assurer.

À partir d'une distribution uniforme dans l'intervalle $[0,1]$, on peut générer une distribution uniforme dans tout intervalle fini par transformation affine $x' = a\,x + b$. Plus important : on peut générer des distributions de probabilité plus complexes, par les méthodes expliquées ci-dessous.

B.2 Méthode de transformation

Une méthode simple pour simuler une variable aléatoire y qui suit une distribution de probabilité non uniforme est d'appliquer une transformation $x \mapsto y = f(x)$ à la variable uniforme x. En général, si une variable aléatoire x suit une distribution $p_1(x)$, les aléatoires [3] situés dans l'intervalle $\mathrm{d}x$ autour de x correspondent alors à des valeurs aléatoires de y situées dans un intervalle $\mathrm{d}y$ autour de y, distribués selon une fonction $p_2(y)$ telle que

$$p_1(x)|\,\mathrm{d}x| = p_2(y)|\,\mathrm{d}y| \quad \text{ou encore} \quad p_2(y) = p_1(x)\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| \tag{15.4}$$

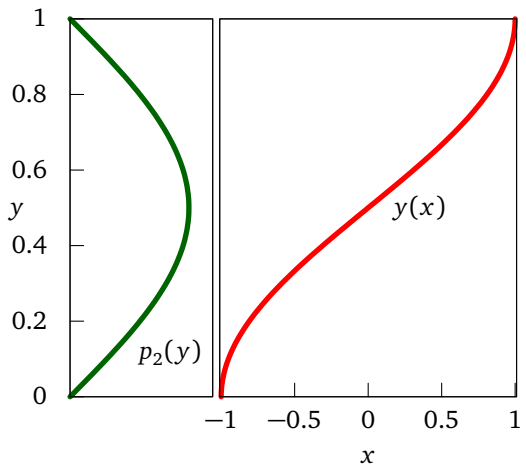


FIGURE 15.1
Méthode de transformation.

3. Nous désignerons couramment une variable aléatoire par le nom *aléatoire*, correspondant à l'anglais *deviate*.

La méthode est illustrée sur la figure 15.1, dans le cas $p_1(x) = \frac{1}{2}$ sur $x \in [-1,1]$ et $p_2(y) = \frac{1}{2}\pi \sin \pi y$ sur $y \in [0,1]$. D'après la discussion ci-dessus, essayons de trouver la transformation appropriée $y = f(x)$:

$$\begin{aligned} p_2(y) = \frac{1}{2}\pi \sin \pi y &= p_1(x)\frac{\mathrm{d}x}{\mathrm{d}y} \\ \Longrightarrow \int_{-1}^{x} \mathrm{d}x' &= \pi \int_0^y \sin \pi y' \, \mathrm{d}y' \\ \Longrightarrow 1 + x = 1 - \cos \pi y \quad &\text{ou enfin} \quad x = -\cos \pi y \end{aligned} \tag{15.5}$$

Donc $y = \arccos(-x)/\pi$ est la transformation désirée. Remarquons, d'après la forme de la relation $y = f(x)$, que si x est distribué uniformément dans l'intervalle $[0,1]$, alors y sera plus fréquent là où $\mathrm{d}y/\mathrm{d}x$ est le plus faible, et vice-versa.

Autre exemple, cette fois obtenu à l'envers, c'est-à-dire en partant de la transformation. Considérons $y = -\ln(x)$ dans l'intervalle $y \in [0,\infty]$ et $p_1(x) = 1$ dans $x \in [0,1]$. Alors

$$p_2(y) = \left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| = x = \mathrm{e}^{-y} \tag{15.6}$$

On parvient alors à la distribution exponentielle.

Plus généralement, produire une distribution p_2 quelconque à partir d'une distribution uniforme requiert la solution analytique de l'équation différentielle suivante :

$$p_2(y)\,\mathrm{d}y = \mathrm{d}x \Longrightarrow \int_0^y \mathrm{d}z\, p_2(z) \coloneqq \mathrm{F}_2(y) = x \tag{15.7}$$

(on a supposé $\mathrm{d}y/\mathrm{d}x > 0$), où $\mathrm{F}_2(y)$ est l'intégrale de la distribution p_2. En pratique, tirer un aléatoire selon la distribution p_2 revient à tirer un aléatoire x selon une distribution uniforme, puis à effectuer la correspondance $y = \mathrm{F}_2^{-1}(x)$ pour obtenir la valeur de y distribuée selon p_2. La méthode de transformation est utile et efficace uniquement si F_2 peut être calculé et inversé analytiquement.

B.3 Méthode du rejet

Une méthode plus générale pour générer une distribution $p(x)$ est la méthode du rejet, illustrée à la figure 15.2. Supposons que nous désirions générer un aléatoire x qui suive la distribution $p(x)$ (en rouge). Sur la figure le domaine de x est fini (de a à b), mais cela n'est pas nécessaire. Supposons en outre que nous sachions comment échantillonner une distribution $f(x)$ (en noir sur la figure) définie sur le même intervalle, et qui a été normalisée de manière à ce qu'elle soit partout plus grande que $p(x)$. La méthode du rejet procède comme suit :

1. On tire une valeur $x \in [a,b]$, distribuée selon $f(x)$. Par exemple, $f(x)$ pourrait être échantillonné par la méthode de transformation décrite ci-haut.
2. On tire un deuxième nombre y uniformément distribué entre 0 et $f(x)$.
3. Si $y < p(x)$, on accepte la valeur de x ainsi produite. Sinon, on la rejette et on recommence jusqu'à ce que la valeur de x soit acceptée.

L'ensemble des valeurs de x ainsi générées suit la distribution $p(x)$.

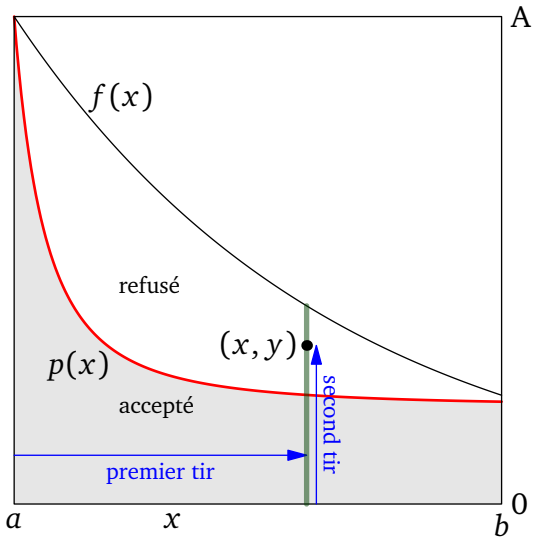


FIGURE 15.2
Méthode du rejet.

On peut comprendre la méthode du rejet géométriquement comme suit : en échantillonnant x selon $f(x)$ et y de manière uniforme entre 0 et $f(x)$, on se trouve à générer des points (x, y) qui sont uniformément répartis au-dessous de la courbe $f(x)$. En rejetant systématiquement tous les points compris entre $p(x)$ et $f(x)$, on se trouve à échantillonner la distribution $p(x)$.

Plus la fonction $f(x)$ est proche de $p(x)$, plus la méthode est efficace, c'est-à-dire plus les rejets sont rares. Le cas particulier d'une distribution $f(x)$ uniforme peut toujours servir en pratique, mais risque d'être très inefficace si la fonction $p(x)$ est piquée autour d'une valeur en particulier.

B.4 Méthode du rapport des aléatoires uniformes

Une méthode puissante pour générer un grand nombre de distributions est la méthode du rapport des aléatoires uniformes (Kinderman et Monahan), que nous allons maintenant expliquer. La méthode du rejet peut se représenter ainsi dans le plan (x, p) :

$$p(x)\,\mathrm{d}x = \int_0^{p(x)} \mathrm{d}p'\,\mathrm{d}x \tag{15.8}$$

Ce qui revient à dire que si on échantillonne uniformément la région bornée par l'axe des x et par la courbe $p(x)$ dans le plan (x, p), on se trouve à échantillonner x selon la distribution $p(x)$. Cette affirmation, tout évidente qu'elle soit, repose sur le fait que l'intégrant de l'intégrale ci-dessus est l'unité.

Procédons maintenant à un changement de variables :

$$x = \frac{v}{u} \qquad p = u^2 \tag{15.9}$$

Le jacobien associé est une constante :

$$\frac{\partial(p,x)}{\partial(u,v)} = \begin{vmatrix} 2u & -v/u^2 \\ 0 & 1/u \end{vmatrix} = 2 \tag{15.10}$$

et donc on peut écrire, sur le plan (u,v),

$$p(x)\,\mathrm{d}x = 2\int_0^{\sqrt{p(x)}} \mathrm{d}u\,\mathrm{d}v = 2\int_0^{\sqrt{p(v/u)}} \mathrm{d}u\,\mathrm{d}v \tag{15.11}$$

Le domaine d'intégration est défini par la courbe représentant la distribution $p(x)$, d'une part, et par la droite $p = 0$, d'autre part, exprimées en fonction de u et v. Comme l'intégrant est encore une fois constant, un échantillonnage uniforme à l'intérieur de cette courbe dans le plan (u,v) revient à échantillonner x selon la distribution $p(x)$, la valeur de x étant simplement donnée par le rapport v/u. Nous avons ainsi généré x par un rapport de variables aléatoires (u et v) distribuées uniformément, avec cependant des conditions de rejet.

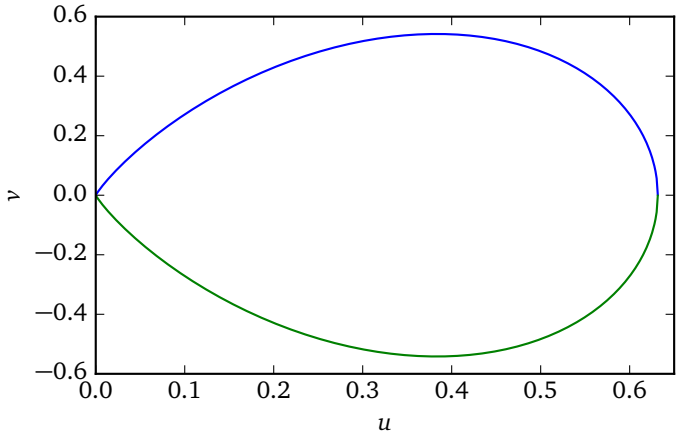


FIGURE 15.3
Domaine du plan (u,v) permettant d'échantillonner une distribution gaussienne pour le rapport $x = v/u$ (on a choisi $\sigma = 1$ et $\mu = 0$).

Considérons par exemple la distribution gaussienne :

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{(x-\mu)^2}{2\sigma^2} \tag{15.12}$$

où σ est l'écart-type et μ la moyenne. Posons $\mu = 0$, ce que nous pouvons faire sans perte de généralité. En fonction de u et v, l'équation qui définit la distribution devient

$$u^2 = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{v^2}{2u^2\sigma^2} \quad \text{ou} \quad v = \pm u\sigma\sqrt{2\ln\left(\sqrt{2\pi}\sigma\,u^2\right)} \tag{15.13}$$

Cette équation est représentée par la courbe illustrée à la figure 15.3.

Pour échantillonner efficacement une distribution gaussienne, on doit alors échantillonner de manière uniforme u et v dans le domaine du graphique, rejeter les points qui tombent à l'extérieur de la courbe, et ensuite retourner la valeur $x = v/u$. Pour augmenter l'efficacité de la procédure, on définit des bornes externe et interne (angl. *squeeze*) à la courbe, bornes qui sont rapidement calculables en comparaison de la formule (15.13). On accepte alors les points à l'intérieur de la borne interne et on les rejette à l'extérieur de la borne externe. Ce n'est que dans les cas plus rares où le point tombe entre les deux bornes qu'on doit tester la frontière exacte (15.13). On peut ainsi proposer un code qui produit un aléatoire gaussien au coût moyen de 2.74 aléatoires uniformes.

CHAPITRE 16

MÉTHODE DE MONTE-CARLO

L'épithète «Monte-Carlo» est appliquée généralement à des méthodes de calcul qui reposent sur l'échantillonnage aléatoire d'un ensemble très vaste. Elle fait bien sûr référence à la célèbre maison de jeu de Monaco, et a été proposée par J. von Neumann et Stanislaw Ulam lors du développement des premières applications de ce genre à Los Alamos dans les années 1950. [1] L'idée générale est la suivante : plusieurs problèmes d'intérêt requièrent de sommer (ou de faire une moyenne) sur un très grand nombre d'états. Ceux-ci peuvent être les états possibles d'un gaz ou d'un liquide, d'un système magnétique, ou les trajectoires d'une particule dans un milieu. Le mot «état» est ici employé au sens de «configuration», mais peut désigner quelque chose d'aussi simple qu'un point dans un domaine d'intégration en plusieurs dimensions. Au lieu de sommer systématiquement sur tous les états, ce qui est pratiquement impossible en raison de leur nombre astronomique, on échantillonne ces états à l'aide de nombres aléatoires. La difficulté consiste généralement à (1) échantillonner de manière efficace et (2) bien estimer les erreurs statistiques ainsi commises.

A Intégrales multidimensionnelles

Supposons qu'on doive intégrer une fonction $\mathrm{A}(x)$ de d variables, ou plutôt la moyenne de A dans son domaine de définition, ce qui revient au même :

$$\langle f \rangle = \frac{\displaystyle\int_\Omega \mathrm{d}^d x\, \mathrm{A}(x)}{\displaystyle\int_\Omega \mathrm{d}^d x} \tag{16.1}$$

On considère en fait x comme une variable aléatoire, et de ce fait A, en tant que fonction de x, est aussi une variable aléatoire. La densité de probabilité $p(x)$ est alors une constante égale à $1/\mathrm{vol}(\Omega)$ à l'intérieur du domaine Ω, et nulle à l'extérieur de Ω. La technique la plus simple consiste à échantillonner x de manière uniforme dans un domaine de définition hypercubique. Si le domaine Ω n'est pas hypercubique, il suffit alors de définir un domaine hypercubique contenant Ω tout juste et de définir $\mathrm{A}(x) = 0$ à l'extérieur de Ω (voir fig. 16.1).

1. L'oncle de Ulam était, semble-t-il, un joueur compulsif qui empruntait de l'argent pour jouer à la roulette à Monte-Carlo.

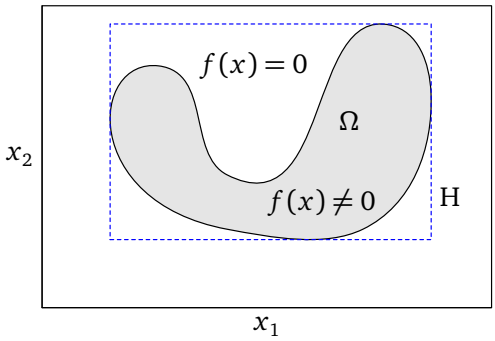


FIGURE 16.1
Définition d'une région d'intégration hypercubique H autour du domaine Ω d'une fonction.

Le principe de l'intégration Monte-Carlo est d'échantillonner le domaine de A avec N points x_i et d'estimer la moyenne ainsi :

$$\langle \mathrm{A} \rangle \approx \bar{\mathrm{A}} := \frac{1}{\mathrm{N}} \sum_i \mathrm{A}(x_i) \tag{16.2}$$

Quelle erreur commet-on en procédant de la sorte ? Autrement dit, quel est l'écart-type de $\bar{\mathrm{A}}$? La réponse à cette question provient du théorème de la limite centrale :

Théorème 16.1 Limite centrale
Considérons une variable aléatoire A qu'on mesure N fois. Les résultats de chaque mesure, A_i, sont des variables aléatoires indépendantes, quoiqu'ayant les mêmes propriétés. Alors la meilleure estimation de la moyenne $\langle \mathrm{A} \rangle$ est l'espérance mathématique

$$\bar{\mathrm{A}} = \frac{1}{\mathrm{N}} \sum_i \mathrm{A}_i \tag{16.3}$$

et l'écart-type de cette estimation est

$$\Delta(\bar{\mathrm{A}}) = \frac{1}{\sqrt{\mathrm{N}}} \Delta(\mathrm{A}) \tag{16.4}$$

Preuve: En calculant simplement la valeur moyenne de l'espérance mathématique, on constate qu'elle coïncide avec la valeur moyenne de A :

$$\langle \bar{\mathrm{A}} \rangle = \frac{1}{\mathrm{N}} \sum_i \langle \mathrm{A}_i \rangle = \frac{1}{\mathrm{N}} \sum_i \langle \mathrm{A} \rangle = \langle \mathrm{A} \rangle \tag{16.5}$$

car chaque mesure A_i obéit à la même loi de probabilité qui régit la variable A, de sorte que $\langle \mathrm{A}_i \rangle = \langle \mathrm{A} \rangle$. L'estimation de l'erreur s'obtient ensuite en calculant la variance de $\bar{\mathrm{A}}$:

$$\mathrm{Var}\bar{\mathrm{A}} = \langle \bar{\mathrm{A}}^2 \rangle - \langle \bar{\mathrm{A}} \rangle^2 = \langle \bar{\mathrm{A}}^2 \rangle - \langle \mathrm{A} \rangle^2 \tag{16.6}$$

Or

$$\langle \bar{\mathrm{A}}^2 \rangle = \frac{1}{\mathrm{N}^2} \sum_{i,j} \langle \mathrm{A}_i \mathrm{A}_j \rangle = \frac{1}{\mathrm{N}^2} \sum_i \langle \mathrm{A}_i^2 \rangle + \frac{1}{\mathrm{N}^2} \sum_{i,j\ (i \neq j)} \langle \mathrm{A}_i \mathrm{A}_j \rangle \tag{16.7}$$

Dans l'hypothèse où les mesures successives de A ne sont pas corrélées, alors $\langle A_i A_j \rangle = \langle A_i \rangle \langle A_j \rangle = \langle A \rangle^2$ si $i \neq j$ (il y a $N(N-1)$ termes comme celui-là). On trouve alors

$$\langle \bar{A}^2 \rangle = \frac{1}{N} \langle A^2 \rangle + \frac{N-1}{N} \langle A \rangle^2 \tag{16.8}$$

Au total, la variance de $\bar{A}$ est

$$\mathrm{Var}\bar{A} = \frac{1}{N} \langle A^2 \rangle - \frac{1}{N} \langle A \rangle^2 = \frac{1}{N} \mathrm{Var}A \tag{16.9}$$

On retrouve donc le théorème de la limite centrale, que nous venons en fait de démontrer : l'erreur Δ_A commise sur l'estimation de $\langle A \rangle$ est l'écart-type $\sqrt{\mathrm{Var}A}$, divisé par $\sqrt{N}$.

Bref, l'erreur sur la valeur moyenne $\langle A \rangle$ commise lorsqu'on la remplace par $\bar{A}$ est

$$\Delta = \sqrt{\frac{\mathrm{Var}A}{N}} \qquad \mathrm{Var}A := \langle A^2 \rangle - \langle A \rangle^2 \tag{16.10}$$

La variance de A dépend beaucoup du degré de variation de $A(x)$ dans Ω. Si la fonction varie beaucoup dans ce domaine, VarA sera très élevé, et vice-versa. Mais le point important est que cette erreur diminue comme $1/\sqrt{N}$.

Par exemple, si on tire N aléatoires uniformément distribués dans $[-1,1]$, la moyenne de chaque aléatoire est 0, et la variance est $1/3$. Chacun des tirs suit la même loi de probabilité, et donc l'espérance mathématique des résultats, $\bar{A}$, suivra une loi normale de moyenne 0 et de variance $1/3N$. En clair, cela signifie qu'en générant N «mesures» A_i de A, on se trouve à générer une instance de la variable $\bar{A}$, instance qui tombe à l'intérieur d'un écart-type $\Delta(A)/\sqrt{N}$ de la moyenne $\langle A \rangle$ 68% du temps, et à l'intérieur de deux écarts-types 95% du temps.

Si on intègre la même fonction en utilisant une grille fixe de points, par exemple avec la méthode de Simpson, l'erreur commise sera de l'ordre de $1/n^4$, où n est le nombre de points par direction, menant à un nombre total de points $N = n^d$ (on suppose un nombre de points égal dans chaque direction de l'espace). L'erreur commise dans la méthode de Simpson est donc

$$\Delta_{\mathrm{Simpson}} \sim \frac{1}{N^{4/d}} \tag{16.11}$$

On voit que la méthode Monte-Carlo converge plus rapidement que la méthode de Simpson pour des dimensions $d > 8$: pour les intégrales dans des domaines de grande dimension, le Monte-Carlo est la meilleure méthode! Le fait de remplacer la méthode de Simpson par une autre méthode convergeant plus rapidement ne change pas fondamentalement ce résultat, mais ne fait que repousser son impact à des dimensions plus grandes. Or, en mécanique statistique classique, on doit calculer les valeurs moyennes en intégrant sur l'espace des phases Ω de M particules. En trois dimensions d'espace, la dimension de l'espace Ω est $d = 6M$ (ou $d = 4M$ en deux dimensions d'espace). Donc quelques particules seulement suffisent à rendre l'intégration Monte-Carlo incontournable.

N	$4\bar{A}$	σ	erreur/σ
10	2.4	0.52	1.4
10^2	3.12	0.16	0.13
10^3	3.176	0.052	−0.66
10^4	3.1148	0.016	1.63
10^5	3.1498	0.052	−1.6
10^6	3.1417	0.0016	−0.068
10^7	3.14177	0.00052	−0.35

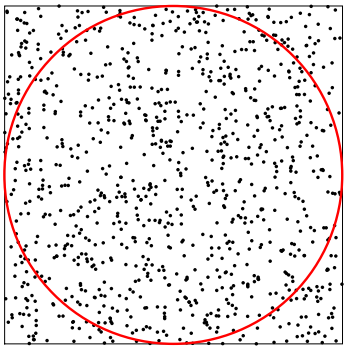


FIGURE 16.2
Calcul de l'aire d'un cercle par la méthode de Monte-Carlo. À gauche : valeur estimée de π en fonction du nombre de points, écart-type de l'estimation et erreur commise en rapport avec l'écart-type.

A.1 Exemple simple : calcul de l'aire d'un disque

Appliquons la méthode d'intégration Monte-Carlo au calcul de l'aire d'un disque en dimension 2. La fonction A est alors définie comme suit :

$$A(x) = \begin{cases} 1 & \text{si} \quad (x_1^2 + x_2^2) < 1 \\ 0 & \text{sinon} \end{cases} \tag{16.12}$$

le domaine Ω est l'intérieur du disque, et le domaine hypercubique qui enveloppe Ω est le carré qui s'étend de $(-1,-1)$ à $(1,1)$. On échantillonne uniformément des points dans ce carré, en tirant deux aléatoires uniformes $x \in [-1,1]$ et $y \in [-1,1]$. La valeur moyenne de $A(x)$ doit être $p = \pi/4$, et l'estimation $\bar{A}$ doit tendre vers $\pi/4$ quand $N \to \infty$. La variance de A est

$$\mathrm{Var}A = p - \langle A \rangle^2 = p - p^2 = \frac{\pi}{4}(1 - \frac{\pi}{4}) \tag{16.13}$$

L'estimation de la valeur de A est donnée par l'*espérance mathématique* :

$$\bar{A} = \frac{1}{N} \sum_i A_i \tag{16.14}$$

La valeur de $\bar{A}$ ainsi calculée (multipliée par 4), dans un exemple de simulation, est illustrée à la figure 16.2 en fonction du nombre de points tirés. Selon le théorème de la limite centrale, l'erreur commise sur la moyenne est alors

$$\sigma = \sqrt{\frac{1}{N} \frac{\pi}{4} \left(1 - \frac{\pi}{4}\right)} \tag{16.15}$$

Boîte à outils

Le module python `vegas` permet de calculer des intégrales multidimensionnelles à l'aide d'un algorithme d'échantillonnage selon l'importance. Ce module peut être installé à l'aide de la commande `pip` sur la plupart des plates-formes :
`pip install vegas`
La documentation peut être trouvée sur le site de distribution. Une documentation plus courte, en format html et restreinte aux fonctions et classes Python, peut être générée par la commande
`pydoc -w vegas`
(cette fonctionnalité existe pour tous les modules bien construits). La documentation plus longue du module contient une section qui explique l'algorithme.
Un court exemple d'utilisation de `vegas` suit :

```
1 import vegas
2 import math
3 def f(x):
4  return 16*x[0]*x[1]*x[2]*x[3]
5 result = integ(f, nitn=10, neval=10000)
6 print(result.summary())
7 print('result = %s Q = %.2f' % (result, result.Q))
```

`vegas` divise chaque axe d'intégration selon une grille, et donc l'ensemble du domaine d'intégration est divisé en régions hyperrectangulaires. Dans chacune de ces régions, un échantillonnage de points est utilisé pour calculer une estimation de l'intégrale comme expliqué ci-dessus, et les résultats des différentes régions sont ajoutés. Comme la valeur de la fonction peut varier beaucoup d'une région à l'autre, l'utilisation d'une grille uniforme est inefficace et mène à une variance trop grande des résultats. Pour pallier ce problème, `vegas` procède à `nint` itérations ; à chaque itération, la grille est modifiée de manière non uniforme dans chaque direction (les points ne sont plus également espacés) afin que la contribution de chaque région soit à peu près égale, ce qui minimise la variance de la somme de ces contributions.

B L'algorithme de Metropolis

Raffinons maintenant la méthode de Monte-Carlo. Soit Ω l'espace sur lequel est définie une variable aléatoire multidimensionnelle x, et soit $\mu(x)$ la distribution de probabilité associée (dans le problème défini par (16.1), $\mu(x)=1$). Le problème est de calculer la valeur moyenne d'une fonction $\mathrm{A}(x)$ dans cet espace :

$$\langle \mathrm{A} \rangle = \frac{\displaystyle\int_\Omega \mathrm{d}x\, \mu(x)\mathrm{A}(x)}{\displaystyle\int_\Omega \mathrm{d}x\, \mu(x)} \tag{16.16}$$

En mécanique statistique, x représente un état dynamique du système (un point dans l'espace des phases, par exemple) ; $\mathrm{A}(x)$ représente une quantité physique (une fonction dans l'espace des états), comme l'énergie ou l'aimantation. Enfin, la distribution $\mu(x)$ est la probabilité relative d'occupation de l'état x, par exemple la loi de Boltzmann $\mu(x) = \exp -\mathrm{E}(x)/\mathrm{T}$, T étant la température absolue en

unités de l'énergie E. Si on suppose que l'espace des configurations Ω forme un ensemble discret et non continu, l'expression ci-dessus s'écrit plutôt

$$\langle \mathrm{A} \rangle = \frac{\sum_x \mu_x \mathrm{A}_x}{\sum_x \mu_x} \tag{16.17}$$

B.1 Chaînes de Markov

Pour calculer la moyenne (16.17), on doit en pratique générer un ensemble de configurations aléatoires x, ce qui n'est pas toujours facile et économique dans un espace de grande dimension. En particulier, la méthode du rejet simple peut dans ces cas devenir très inefficace si une proportion très importante des configurations x ont des probabilités très petites. L'*algorithme de Metropolis* vise à échantillonner l'espace des configurations Ω non pas en produisant des valeurs successives qui sont absolument indépendantes, mais plutôt à produire une «marche aléatoire» dans Ω, qui puisse traverser Ω en passant plus de temps dans les régions où μ_x est plus grand, c'est-à-dire en générant une séquence de points x_i qui finit par être distribuée selon μ_x lorsqu'elle est suffisamment longue.

L'algorithme vise en fait à produire une *chaîne de Markov*, c'est-à-dire une succession de valeurs aléatoires dont chacune provient de la précédente selon une loi de probabilité bien précise :

$$x_1 \longrightarrow x_2 \longrightarrow x_3 \longrightarrow \cdots \longrightarrow x_i \longrightarrow x_{i+1} \longrightarrow \cdots \tag{16.18}$$

Le passage d'une valeur x à un valeur y à chaque étape de la chaîne est effectué avec une probabilité $\mathrm{W}(x \longrightarrow y)$, appelée *probabilité de transition*. Dans tout calcul numérique, le nombre de valeurs possibles de x est fini (disons M), et chaque valeur possible de x pourrait alors être étiquetée à l'aide d'un indice variant de 1 à M. Dans ce cas, les probabilités de transition W forment une matrice finie d'ordre M notée $\mathrm{W}_{yx} = \mathrm{W}(x \longrightarrow y)$ (noter l'ordre inverse des indices). Afin de simplifier la notation, nous étiquetons directement les rangées et colonnes de cette matrice par les configurations x et y et non par les entiers correspondants. Un vecteur X de dimension M représente alors une certaine distribution de la variable aléatoire x, qui n'est pas nécessairement la distribution recherchée (μ_x). Par exemple, le vecteur X pourrait n'avoir qu'une composante non nulle au départ.

Exemple 16.1 chaîne de Markov à deux états

L'exemple le plus simple d'une chaîne de Markov se produit sur un espace à deux états, qu'on notera A et B. Appelons p la probabilité de passer de B vers A : $\mathrm{W}(\mathrm{B} \longrightarrow \mathrm{A}) = \mathrm{W}_{\mathrm{AB}} = p$. La probabilité de rester sur B est donc $\mathrm{W}_{\mathrm{BB}} = 1 - p$. De même, soit q la probabilité de passer de A vers B : $\mathrm{W}(\mathrm{A} \longrightarrow \mathrm{B}) = \mathrm{W}_{\mathrm{BA}} = q$, alors que la probabilité de rester sur A est $\mathrm{W}_{\mathrm{AA}} = 1 - q$. Ces probabilités forment la matrice

$$\mathrm{W} = \begin{pmatrix} 1-q & p \\ q & 1-p \end{pmatrix} \tag{16.19}$$

Si le premier maillon de la chaîne de Markov représente l'état A avec certitude, il est représenté par le vecteur

$X_1 = (1,0)$. Le deuxième maillon est alors obtenu en appliquant la matrice W sur X :

$$X_2 = WX_1 = \begin{pmatrix} 1-q \\ q \end{pmatrix} \quad \text{et ensuite} \quad X_3 = WX_2 = \begin{pmatrix} (1-q)^2 + pq \\ q(2-p-q) \end{pmatrix} \qquad \text{etc.} \tag{16.20}$$

À la longue, le vecteur X_k va converger vers le vecteur propre de W possédant la plus grande valeur propre λ_1, modulo une constante multiplicative λ_1^k.
Or, dans ce cas, on montre facilement que $\lambda = 1$ est une valeur propre. En effet,

$$W - \mathbb{I} = \begin{pmatrix} -q & p \\ q & -p \end{pmatrix} \tag{16.21}$$

qui est une matrice singulière (la deuxième rangée est l'opposée de la première). Le vecteur propre correspondant est $X = (p,q)$. Comme la somme des valeurs propres est la trace $2-p-q$, la deuxième valeur propre est simplement $\lambda_2 = 1-p-q$, dont la valeur absolue est forcément inférieure à un, puisque $0 \leq p,q \leq 1$. Donc, en appliquant W à répétition, seul le vecteur propre $X = (p,q)$ associé à $\lambda_1 = 1$ va survivre. À la fin de la chaîne, la probabilité d'être dans l'état A sera $p/(p+q)$, et celle d'être dans l'état B sera $q/(p+q)$ (nous avons normalisé les probabilités afin que leur somme soit 1). Donc, en partant d'une certitude d'être dans l'état A, nous avons abouti à une distribution de probabilité déterminée par la matrice W.

Ceci est une propriété générale : en appliquant de manière répétée la matrice W sur un vecteur X représentant une distribution de probabilité, la distribution évolue, et peut tendre vers une limite stable, ce qui correspond à un vecteur propre de W de valeur propre unité :

$$WX = X \tag{16.22}$$

Le vecteur propre X correspond alors à une distribution de probabilité qui n'est pas affectée par la marche, idéalement celle qu'on désirait générer au départ : $X = (\mu_x)$.

Pour ce faire, les probabilités de transition W_{yx} doivent respecter les contraintes suivantes :

1. Normalisation : $\sum_y W_{yx} = 1$, en langage discret. Cette propriété est nécessaire afin que W soit effectivement une matrice de probabilités. Elle impose M contraintes sur les M^2 composantes de W.
2. Ergodicité : toute valeur de y doit être éventuellement accessible à partir de toute valeur de x si on applique la matrice W un nombre suffisant de fois : il existe un entier n tel que $(W^n)_{yx} \neq 0 \ \forall x, y$. Cette propriété nous assure que toutes les régions de l'espace Ω seront visitées, quel que soit le point de départ de la chaîne.
3. Le bilan détaillé :

   $$\mu_x W_{yx} = \mu_y W_{xy} \tag{16.23}$$

   où on ne somme pas sur l'indice répété. Cette relation impose $\frac{1}{2}M(M-1)$ contraintes à la matrice W. Elle suffit cependant à s'assurer que le vecteur μ_x est un vecteur propre de W de valeur propre unité. En effet,

   $$\sum_y W_{xy}\mu_y = \sum_y \frac{\mu_x}{\mu_y} W_{yx}\mu_y = \sum_y W_{yx}\mu_x = \mu_x \tag{16.24}$$

   où on a utilisé la condition de normalisation et la condition du bilan détaillé. Ces deux conditions ensemble ne fixent pas W de manière unique, mais seulement $\frac{1}{2}M(M+1)$ composantes.

Notons cependant qu'il n'est pas nécessaire de connaître précisément la normalisation de la distribution μ_x. Autrement dit, μ_x peut être une probabilité relative, dont la somme est la fonction de partition

$$\mathrm{Z} = \sum_x \mu_x \tag{16.25}$$

Dans ce cas, la condition de normalisation ci-dessus se ramène à

$$\sum_y \mathrm{W}_{yx} = \mathrm{Z} \quad \forall x \quad \text{et} \quad \sum_y \mathrm{W}_{xy}\mu_y = \sum_y \frac{\mu_x}{\mu_y}\mathrm{W}_{yx}\mu_y = \sum_y \mathrm{W}_{yx}\mu_x = \mathrm{Z}\mu_x \tag{16.26}$$

ce qui revient à demander que le vecteur X soit un vecteur propre de W avec valeur propre Z :

$$\mathrm{WX} = \mathrm{ZX} \tag{16.27}$$

L'algorithme de Metropolis utilise la forme suivante de la matrice W :

$$\mathrm{W}_{xy} = \min\left(1, \frac{\mu_x}{\mu_y}\right) \tag{16.28}$$

ce qui peut également se représenter par le tableau suivant :

	$\mathrm{W}(x \to y) = \mathrm{W}_{yx}$	$\mathrm{W}(y \to x) = \mathrm{W}_{xy}$
$\mu_x > \mu_y$	μ_y/μ_x	1
$\mu_x < \mu_y$	1	μ_x/μ_y

On constate que cette prescription respecte la condition du bilan détaillé. Par contre, elle ne respecte pas la condition de normalisation stricte, mais il suffit de multiplier W par une constante de normalisation inconnue Z^{-1} pour régler ce problème. L'avantage de l'algorithme de Metropolis est que la connaissance préalable de cette constante de normalisation n'est pas requise ; seules les probabilités relatives μ_x/μ_y le sont.

B.2 Analyse d'erreur

L'aspect le plus subtil de la méthode de Monte-Carlo est l'estimation des résultats et de leur incertitude à partir des données statistiques recueillies. Considérons un ensemble de valeurs mesurées A_i ($i = 1, \ldots, \mathrm{N}$) d'une observable A, lors d'un processus markovien, par exemple basé sur l'algorithme de Metropolis. On suppose que la variable aléatoire A possède une moyenne exacte $\langle \mathrm{A} \rangle$ et une variance exacte VarA définie par

$$\mathrm{VarA} = \langle \mathrm{A}^2 \rangle - \langle \mathrm{A} \rangle^2 \tag{16.29}$$

L'objectif principal de la simulation Monte-Carlo est d'obtenir une estimation de la valeur moyenne $\langle \mathrm{A} \rangle$, ainsi que de l'incertitude Δ_A sur cette valeur.

Nous avons vu ci-haut que la valeur moyenne de A sera estimée par la moyenne statistique des mesures, et que l'erreur associée sera l'écart-type de A divisé par $\sqrt{\mathrm{N}}$, d'après le théorème de la limite centrale :

$$\bar{\mathrm{A}} = \frac{1}{\mathrm{N}} \sum_i \mathrm{A}_i \qquad \Delta(\bar{\mathrm{A}}) = \frac{1}{\sqrt{\mathrm{N}}} \Delta(\mathrm{A}) \tag{16.30}$$

temps d'autocorrélation Le problème avec l'analyse ci-dessus est que les mesures successives A_i dans un processus markovien non idéal sont corrélées. En pratique, les mesures successives sont corrélées sur un certain «temps» caractéristique noté τ_A et appelé *temps d'autocorrélation* :

$$\tau_A := \frac{1}{\mathrm{Var}A}\sum_{t=1}^{\infty}(\langle A_{1+t}A_1\rangle - \langle A\rangle^2) \tag{16.31}$$

Nous employons le mot «temps» dans le sens markovien, c'est-à-dire désignant la position dans la chaîne de Markov. On peut supposer généralement que la corrélation chute exponentiellement avec le temps, de sorte que seuls quelques termes de la somme sur t contribuent de manière appréciable, et donc la borne supérieure infinie de la somme sur les temps n'est pas réellement importante.

justification de la forme de τ_A. Afin de justifier la définition (16.31), remarquons qu'elle suppose une décroissance exponentielle des corrélation en fonction du temps :

$$\langle A_{1+t}A_1\rangle - \langle A\rangle^2 = \mathrm{Var}A\, e^{-t/\tau_A} \qquad t = 0,1,2,3,\ldots \tag{16.32}$$

Le cas $t = 0$ justifie que le préfacteur soit la variance de A. En substituant cette forme dans la définition (16.31), on trouve

$$\sum_{t=1}^{\infty}(\langle A_{1+t}A_1\rangle - \langle A\rangle^2) = \sum_{t=1}^{\infty}\mathrm{Var}A\, e^{-t/\tau_A} = \mathrm{Var}A\frac{e^{-1/\tau_A}}{1-e^{-1/\tau_A}} \tag{16.33}$$

En supposant que $\tau \gg 1$, on peut procéder à un développement limité et

$$\sum_{t=1}^{\infty}(\langle A_{1+t}A_1\rangle - \langle A\rangle^2) \approx \mathrm{Var}A\,\tau_A \tag{16.34}$$

Ceci motive la définition (16.31).

corrections à la limite centrale Retournons à l'éq. (16.7), sans toutefois supposer que les mesures sont non corrélées :

$$\langle \bar{A}^2\rangle = \frac{1}{N}\langle A^2\rangle + \frac{N-1}{N}\langle A\rangle^2 + \frac{1}{N^2}\sum_{i,j\ (i\neq j)}\left(\langle A_iA_j\rangle - \langle A\rangle^2\right) \tag{16.35}$$

et donc

$$\begin{aligned}
\mathrm{Var}\bar{A} &= \frac{1}{N}\mathrm{Var}A + \frac{1}{N^2}\sum_{i,j\ (i\neq j)}\left(\langle A_iA_j\rangle - \langle A\rangle^2\right)\\
&= \frac{1}{N}\mathrm{Var}A + \frac{2}{N^2}\sum_{i}^{N}\sum_{t=1}^{N-i}\left(\langle A_iA_{i+t}\rangle - \langle A\rangle^2\right)\\
&\approx \frac{1}{N}\mathrm{Var}A + \frac{2}{N}\sum_{t=1}^{\infty}\left(\langle A_1A_{1+t}\rangle - \langle A\rangle^2\right)\\
&= \frac{1}{N}\mathrm{Var}A(1+2\tau_A)
\end{aligned} \tag{16.36}$$

Nous avons donc une estimation corrigée de l'erreur commise sur l'estimation de la valeur moyenne d'une observable. Il nous faut cependant une méthode efficace pour calculer le temps d'autocorrélation τ_A, car la définition directe de cette quantité ne se prête pas à un calcul pratique.

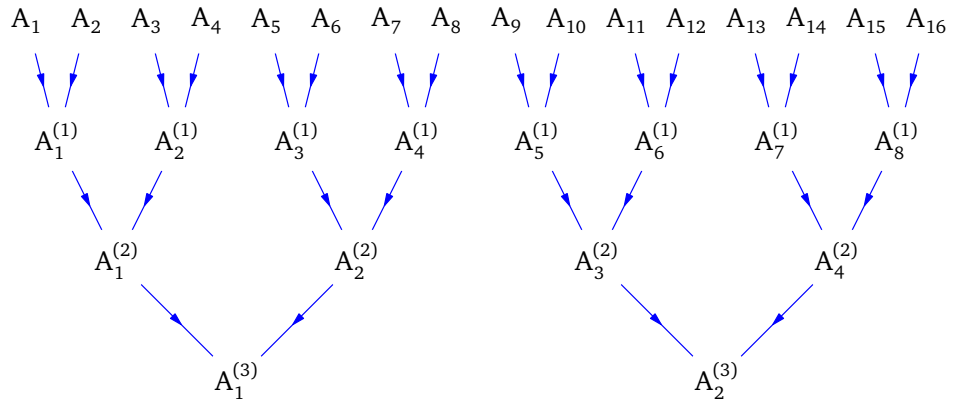


FIGURE 16.3
Schéma de l'analyse logarithmique d'erreurs.

À cette fin, nous allons procéder à une *analyse logarithmique* des erreurs (angl. *binning*), dont le principe est illustré à la figure 16.3. Au fur et à mesure que les mesures A_i sont prises, on construit des séries accessoires de moyennes binaires

$$\mathrm{A}^{(l)} = \tfrac{1}{2}\left(\mathrm{A}^{(l-1)}_{2i-1} + \mathrm{A}^{(l-1)}_{2i}\right) \tag{16.37}$$

où l est l'indice du niveau de la série accessoire. Chaque série accessoire contient la moitié du nombre de termes de la série précédente, jusqu'à n'en contenir qu'un seul (voir la figure). Si on procède à 2^{M} mesures, alors $\mathrm{M}+1$ séries sont ainsi formées, y compris la série principale ($l = 0$). Pour chaque série on peut calculer l'estimation de la moyenne et de la variance :

$$\mathrm{Var}\mathrm{A}^{(l)} = \frac{1}{\mathrm{N}_l(\mathrm{N}_l - 1)} \sum_{i=1}^{\mathrm{N}_l} \left(\mathrm{A}^{(l)}_i - \overline{\mathrm{A}^{(l)}}\right)^2 \tag{16.38}$$

où $\mathrm{N}_l = 2^{\mathrm{M}-l} = 2^{-l}\mathrm{N}$ est le nombre de termes dans la série de niveau l. Les erreurs associées à chaque série, $\Delta^{(l)}_{\mathrm{A}} = \sqrt{\mathrm{Var}\mathrm{A}^{(l)}}$, sont initialement une sous-estimation de l'erreur véritable, mais convergent vers celle-ci dans la limite $l \to \infty$, car les termes consécutifs des séries accessoires deviennent non corrélés dans cette limite. Notons que les erreurs $\Delta^{(l)}_{\mathrm{A}}$ ne croissent pas indéfiniment, même si N_l décroît exponentiellement, parce que la variable $\mathrm{A}^{(l)}$, étant déjà une moyenne, n'a pas la même variance que A. En pratique, on ne construit pas toutes les séries accessoires jusqu'au niveau maximum possible, car l'estimation de l'erreur requiert un nombre M_l de données qui n'est pas trop petit ; par exemple on peut arrêter de construire les séries accessoires en deçà de $\mathrm{M}_l = 2^5 = 32$. On doit être en mesure d'observer que l'erreur estimée $\Delta^{(l)}_{\mathrm{A}}$ sature en fonction de l, et on peut adopter cette valeur limite comme estimation de l'erreur véritable $\Delta^{(\infty)}_{\mathrm{A}}$, et en même temps estimer le temps d'autocorrélation en fonction de cette valeur. En effet, l'estimation initiale $\Delta^{(0)}_{\mathrm{A}}$ de l'erreur, qui est erronée, est donnée par $\sqrt{\mathrm{Var}\mathrm{A}/\mathrm{N}}$, alors que la valeur limite $\Delta^{(\infty)}_{\mathrm{A}}$ est plutôt $\sqrt{\mathrm{Var}\mathrm{A}(1+2\tau_{\mathrm{A}})/\mathrm{N}}$. De là on déduit que

$$\tau_{\mathrm{A}} = \frac{1}{2}\left[\left(\frac{\Delta^{(\infty)}_{\mathrm{A}}}{\Delta^{(0)}_{\mathrm{A}}}\right)^2 - 1\right] \tag{16.39}$$

En pratique, cette méthode ne requiert pas de stocker toutes les séries accessoires ni la série principale. Le nombre de termes serait beaucoup trop grand. Comme on ne requiert que le calcul des valeurs moyennes et des variances, il suffit de garder en mémoire la somme courante des valeurs et la somme courante des valeurs au carré, pour chaque niveau l :

$$\Sigma^{(l)} = \sum_i \mathrm{A}_i^{(l)} \qquad \mathrm{T}^{(l)} = \sum_i \left(\mathrm{A}_i^{(l)}\right)^2 \tag{16.40}$$

1. À chaque fois qu'une mesure est effectuée au niveau 0, on met à jour $\Sigma^{(0)}$ et $\mathrm{T}^{(0)}$.
2. Si le nombre de mesures courant i est impair, on garde en mémoire A_i.
3. S'il est pair, alors on forme la valeur courante de $\mathrm{A}^{(1)}$ selon l'équation (16.37) et on reprend à l'étape 1 ci-dessus, cette fois au niveau $l = 1$, et ainsi de suite.

Cette méthode est mise en oeuvre dans les méthodes `collecte` et `calcul_erreur` de la classe `observable` listée plus bas.

Problème 16.1 :
Montrez que la variance VarA peut être estimée comme suit à partir des valeurs mesurées :

$$\mathrm{VarA} = \frac{\mathrm{N}}{\mathrm{N}-1}\left(\overline{\mathrm{A}^2} - \bar{\mathrm{A}}^2\right) \quad \text{où} \quad \overline{\mathrm{A}^2} := \frac{1}{\mathrm{N}}\sum_i \mathrm{A}_i^2 \tag{16.41}$$

C Le modèle d'Ising

C.1 Définition

Pour motiver la méthode de Monte-Carlo, considérons premièrement le modèle d'Ising, l'une des premières théories proposées pour tenter de comprendre le ferromagnétisme, et l'un des modèles les plus simples de la physique statistique.

Wilhelm Lenz a proposé en 1920 un modèle simple pour décrire le changement de phase magnétique dans un ferroaimant. Ce modèle fut étudié par son étudiant Ernst Ising qui l'a résolu en 1925 dans le cas d'une dimension d'espace, et qui lui a laissé son nom. On suppose dans ce modèle que les atomes du matériau sont disposés régulièrement, par exemple sur un réseau carré ou cubique. Chaque atome porte un spin, et un moment magnétique associé, représenté par une variable s_i pouvant prendre deux valeurs : $+1$ et -1 (i étant l'indice de l'atome, ou du site cristallin). On peut se figurer que ces deux valeurs représentent deux orientations opposées de l'aimantation (ou du spin) de chaque atome. Le modèle est ensuite défini par l'expression de l'énergie totale du système :

$$\mathrm{H}[s] = -\mathrm{J}\sum_{\langle i\,j\rangle} s_i\, s_j \tag{16.42}$$

où la somme est effectuée sur les paires de sites qui sont des voisins immédiats sur le réseau, ce qu'on appelle couramment les *premiers voisins* (ces paires sont notées $\langle i\,j\rangle$). Deux spins voisins

contribuent une énergie −J s'ils sont parallèles, et +J s'ils sont antiparallèles. La tendance énergétique est donc de favoriser les configurations ferromagnétiques (spins alignés), mais celles-ci sont rares en comparaison des configurations d'aimantation nulle : l'entropie favorise l'état paramagnétique.

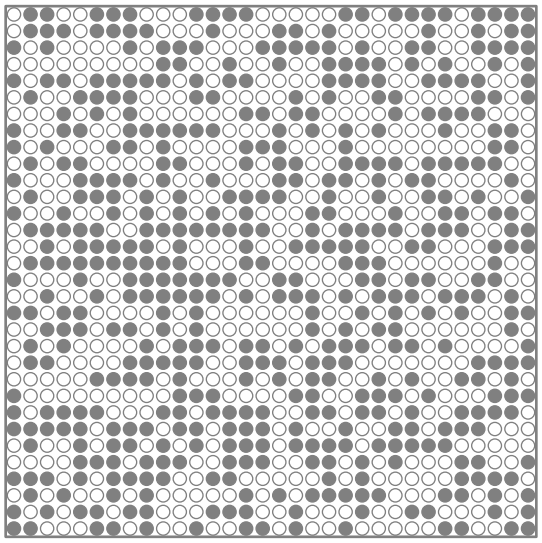


FIGURE 16.4
Configuration aléatoire du modèle d'Ising défini sur un carré de 32×32 sites. Les spins $s_i = +1$ sont indiqués par un carré gris, et les spins $s_i = -1$ par un carré blanc.

Un système comportant N sites supporte donc 2^N configurations de spins différentes (notées s). Selon la mécanique statistique, les quantités observables (énergie ou aimantation) sont obtenues en moyennant leur valeur sur toutes les configurations, chacune étant prise avec un poids

$$\mathrm{W}[s] = \frac{1}{\mathrm{Z}} \exp -\frac{\mathrm{H}[s]}{\mathrm{T}} = \frac{1}{\mathrm{Z}} \exp -\beta \mathrm{H}[s] \qquad \beta := \frac{1}{\mathrm{T}} \quad , \quad \mathrm{Z} := \sum_s \mathrm{e}^{-\beta \mathrm{H}[s]} \tag{16.43}$$

où T est la température absolue, dans des unités telles que $k_\mathrm{B} = 1$. L'énergie E et l'aimantation M moyennes sont alors données par

$$\begin{aligned} \langle \mathrm{E} \rangle &= \frac{1}{\mathrm{Z}} \sum_s \mathrm{H}[s] \mathrm{e}^{-\beta \mathrm{H}[s]} \\ \langle \mathrm{M} \rangle &= \frac{1}{\mathrm{Z}} \sum_s \mathrm{M}[s] \mathrm{e}^{-\beta \mathrm{H}[s]} \quad \text{où} \quad \mathrm{M}[s] = \sum_i s_i \end{aligned} \tag{16.44}$$

L'objectif de Lenz et d'Ising était de voir si un modèle aussi simple que celui-ci pouvait expliquer l'existence d'un changement de phase entre un état ferromagnétique à basse température ($\mathrm{T} < \mathrm{T}_c$), dans lequel $\langle \mathrm{M} \rangle \neq 0$, et un état paramagnétique $\langle \mathrm{M} \rangle = 0$ à haute température ($\mathrm{T} > \mathrm{T}_c$). La température critique T_c correspondrait alors à la température de Curie d'un matériau ferromagnétique.

Ising réussit à résoudre exactement ce modèle en dimension 1 seulement, et observa qu'il n'y avait pas de changement de phase à température non nulle, autrement dit que la température de Curie T_c était nulle. Par contre, en 1944, Lars Onsager réussit à résoudre analytiquement le modèle d'Ising

en deux dimensions d'espace, et trouva une température critique non nulle, donnée par

$$\frac{T_c}{J} = \frac{2}{\ln(1+\sqrt{2})} = 2.2691853142130221\ldots \tag{16.45}$$

Ce résultat confirma la pertinence du modèle et stimula l'étude de modèles plus réalistes.

Un modèle plus réaliste du magnétisme doit tenir compte plus exactement de la nature vectorielle et quantique du spin. Par exemple, le modèle de Heisenberg pour le magnétisme est défini à l'aide des opérateurs du spin :

$$H = -J \sum_{\langle i\,j\rangle} \mathbf{S}_i \cdot \mathbf{S}_j \tag{16.46}$$

Ce modèle peut être traité de manière classique (les vecteurs $\mathbf{S}_i$ sont alors de norme constante) ou de manière quantique (les vecteurs $\mathbf{S}_i$ sont alors des opérateurs quantiques). Le traitement quantique est bien sûr plus réaliste. Le couplage spin-spin du modèle de Heisenberg est isotrope, c'est-à-dire invariant par rotation. Par contre, dans certains matériaux, l'interaction spin-orbite peut mener à un couplage spin-spin qui n'est pas invariant par rotation dans l'espace des spins :

$$H = -J_{xy} \sum_{\langle i\,j\rangle} \left(S_i^x \cdot S_j^x + S_i^y \cdot S_j^y \right) - J_z \sum_{\langle i\,j\rangle} S_i^z \cdot S_j^z \tag{16.47}$$

Dans les cas où $J_z > J_{xy}$, on peut montrer que le modèle se comporte effectivement comme un modèle d'Ising proche de la transition ferromagnétique (les composantes S_i^z ne pouvant prendre que deux valeurs opposées $\pm\frac{1}{2}$). Tout cela pour dire que le modèle d'Ising, malgré sa simplicité extrême, n'est pas tout à fait irréaliste.

Solution numérique du modèle d'Ising

Si la solution analytique du modèle d'Ising était impossible (c'est le cas de la vaste majorité des modèles étudiés en physique statistique), il faudrait se résigner à le résoudre numériquement. Naïvement, il faudrait calculer la fonction de partition Z en procédant à une somme sur toutes les configurations possibles. Or, pour un petit système 32×32 comme celui illustré sur la figure 16.4, le nombre de configurations possibles est $2^{1024} \sim 10^{308}$. Comme on estime le nombre de protons dans l'Univers à $\sim 10^{80}$, on voit à quel point ce nombre est astronomique et empêche tout calcul systématique de la fonction de partition.

Une approche plus raisonnable est d'échantillonner les configurations, en suivant leur distribution de probabilité naturelle, soit la distribution de Boltzmann, et à remplacer les moyennes exactes par des moyennes partielles, cependant entachées d'erreurs statistiques. C'est ce que vise à accomplir la méthode de Monte-Carlo.

C.2 Algorithme de Metropolis appliqué au modèle d'Ising

Le modèle d'Ising est véritablement le plus simple qu'on puisse imaginer : il comporte un nombre fini de configurations pour un nombre fini N de sites (ou degrés de liberté), contrairement à un modèle de gaz basé sur les positions et vitesses continues des particules. Et pourtant le calcul direct des valeurs moyennes (16.44) est peu pratique, le nombre de configurations étant trop élevé (2^N) pour toute valeur intéressante de N. À basse température, la très vaste majorité des configurations

ont une énergie trop élevée pour contribuer de manière significative aux valeurs moyennes. En revanche, à très haute température, les configurations sont toutes à peu près équivalentes, et il n'est pas plus sage alors de les compter toutes.

Voici comment procède le calcul des valeurs moyennes du modèle d'Ising par l'algorithme de Metropolis :

1. Choisir une configuration initiale des spins. On peut la choisir de manière aléatoire, ce qui est alors typique d'une température élevée et demandera un plus long temps d'équilibration (ou thermalisation) si $T < T_c$.
2. À l'intérieur d'une boucle :
   (a) Effectuer une mise à jour de la configuration, en choisissant un site au hasard et en renversant le spin sur ce site.
   (b) Calculer la différence d'énergie ΔE entre cette configuration et la précédente.
   (c) Si $\Delta E < 0$ (énergie plus basse), accepter le changement.
   (d) Si $\Delta E > 0$ (énergie plus élevée), accepter ce changement avec probabilité $e^{-\Delta E/T}$, sinon conserver l'ancienne configuration.
   (e) Mesurer les observables et collecter les statistiques périodiquement (c'est-à-dire à toutes les R mises à jour), pourvu que le nombre de mises à jour déjà effectuées soit suffisant pour que le processus markovien ait convergé vers la distribution de Boltzmann – ce qu'on appelle le temps de thermalisation.
3. Arrêter lorsque L niveaux de séries accessoires ont été complétés. Calculer l'erreur et le temps d'autocorrélation τ_A pour chaque observable. S'assurer que τ_A est considérablement plus petit que 2^L (le nombre de mesures effectuées au niveau 0 pour une valeur donnée au niveau L).

On reconnaît dans l'étape 2 ci-dessus l'algorithme de Metropolis.

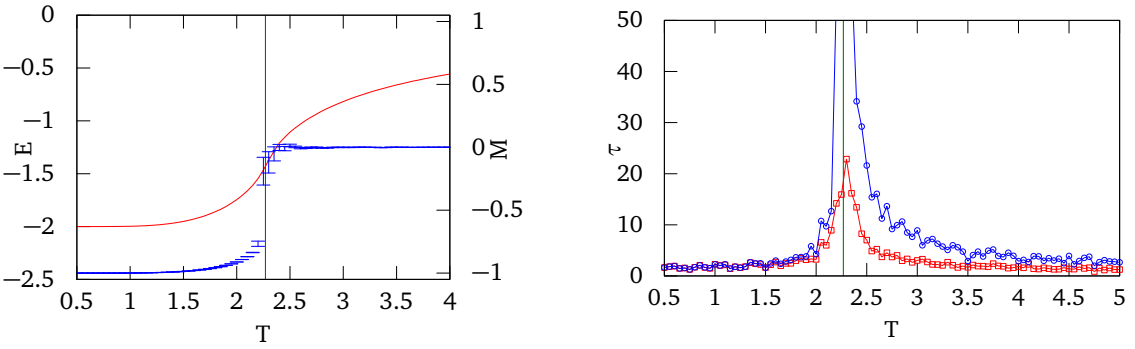


FIGURE 16.5
À gauche : valeurs moyennes de l'énergie E (rouge) et de l'aimantation M (bleu) en fonction de la température dans le modèle d'Ising 2D, sur un amas 32×32. Les barres d'erreur sont affichées pour M. À droite : temps d'autocorrélation pour les mêmes quantités, avec le même code de couleurs. Le temps τ_M pour l'aimantation dépasse les bornes acceptables très proches de la transition ; les valeurs de M ne sont donc pas fiables à cet endroit.

Les résultats d'une simulation typique réalisée avec un code simple sont illustrés à la figure 16.5.

C.3 Changements de phase

L'algorithme de Metropolis fonctionne généralement bien, mais des problèmes peuvent survenir à proximité des changements de phase. Malheureusement, ce sont généralement les changements de phase qui sont intéressants !

Longueur de corrélation

L'un des concepts fondamentaux de la physique statistique est celui de *longueur de corrélation*. Il s'agit, *grosso modo*, de la distance caractéristique ξ en deçà de laquelle les spins de sites différents sont corrélés. On définit la *fonction de corrélation* χ_{ij} entre les sites i et j comme suit :

$$\chi_{ij} = \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle \tag{16.48}$$

Cette fonction est nulle lorsque les sites i et j sont suffisamment éloignés l'un de l'autre, mais décroît généralement de manière exponentielle en fonction de la distance :

$$\chi_{ij} \propto \exp - \frac{|\mathbf{r}_i - \mathbf{r}_j|}{\xi} \tag{16.49}$$

$\mathbf{r}_i$ étant la position du site no i. Cette forme exponentielle définit plus rigoureusement la longueur de corrélation ξ.

Types de changements de phase

On distingue deux types généraux de changements de phase :

1. Les changements de phase *continus*, souvent appelés *changements de phase du deuxième ordre*. Ceux-ci sont caractérisés par une divergence de la longueur de corrélation ξ à l'approche de la transition. Les différentes quantités thermodynamiques, comme l'énergie ou l'aimantation, sont cependant continues à la transition. La transition magnétique du modèle d'Ising est de ce type.
2. Les changements de phase *discontinus*, aussi appelés changements de phase du *premier ordre*. Ceux-là séparent deux phases dont les énergies libres sont identiques exactement à la transition ; mais comme dans ce cas la longueur de corrélation ne diverge pas, différentes régions de l'espace peuvent se retrouver dans des phases différentes : on dit qu'il y a coexistence de phase à la transition. Sous certaines conditions, il peut aussi y avoir hystérésis : une phase de haute température peut se prolonger en deçà de la température de transition pendant un certain temps et se trouver dans un état métastable ; à l'inverse, une phase de basse température peut se retrouver en équilibre métastable au-delà de la température de transition. L'exemple type d'une telle transition est l'ébullition de l'eau : les phases liquide et gazeuse sont en coexistence à la température de transition.

Ralentissement critique

En principe, un véritable changement de phase ne peut exister que dans la limite thermodynamique, c'est-à-dire la limite de taille infinie du système. En pratique, il suffit que le système soit suffisamment grand pour que toutes les caractéristiques d'un véritable changement de phase se manifestent, physiquement ou numériquement. L'un des phénomènes physiques qui se manifestent à proximité d'un changement de phase continu est le *ralentissement critique*, c'est-à-dire des fluctuations de plus en plus lentes qui se manifestent par un temps d'autocorrélation qui diverge. Dans le

modèle d'Ising, on observe en fait que, à la transition elle-même ($T = T_c$), le temps d'autocorrélation croît comme le carré de la taille L du système :

$$\tau \propto L^2 \tag{16.50}$$

On observe aussi que, lors d'une transition du premier ordre qu'on franchit de manière progressive dans une simulation, le système reste dans la phase métastable, et le temps nécessaire pour plonger dans la phase de plus basse énergie se comporte comme

$$\tau \sim \exp L^{d-1} \tag{16.51}$$

où L est la taille du système et d la dimension de l'espace.

Problème 16.2 : Dénombrement des configurations

A Pour un système comportant N sites, combien y a-t-il de configurations d'aimantation M ?

B Tracez le logarithme du nombre de configurations d'aimantation M en fonction de M, pour $N = 36$.

Solution

Une configuration d'aimantation M comporte $N_\uparrow$ spins up et $N_\downarrow$ spins down, tels que $N_\uparrow + N_\downarrow = N$ et $N_\uparrow - N_\downarrow = M$. Donc $N_\uparrow = (N+M)/2$ et $N_\downarrow = (N-M)/2$. Le nombre de façon de distribuer $N_\uparrow$ spins up parmi N sites quand l'ordre n'a pas d'importance est

$$\frac{N!}{N_\uparrow!(N-N_\uparrow)!} = \frac{N!}{((N+M)/2)!((N-M)/2)!} = \binom{N}{(N+M)/2} \tag{16.52}$$

D Simulations de particules

Les méthodes stochastiques sont très utiles dans l'étude des processus faisant intervenir des collisions ou réactions entre particules. La physique des radiations (radiobiologie) met à profit de telles méthodes afin de modéliser les processus en cours dans un tissu irradié. La physique expérimentale des particules élémentaires utilise la méthode de Monte-Carlo pour simuler les cascades de particules au sein des détecteurs de particules des grands accélérateurs. Nous allons illustrer ces méthodes à l'aide d'un exemple simple, tiré de la physique nucléaire.

Supposons qu'un faisceau de neutrons, d'énergie cinétique E, soit incident sur un mur d'épaisseur h, constitué d'une substance – ci-après appelée le *modérateur* – pouvant diffuser élastiquement les neutrons, ou les absorber. Soit $d\sigma_{el}/d\Omega$ la section différentielle de diffusion entre un neutron et un atome de cette substance, dans le repère du centre de masse. Cette section différentielle de diffusion mène à une section efficace σ_{el}. Soit en outre σ_{abs} la section efficace pour l'absorption d'un neutron par le noyau de cette substance. Si ce sont là les deux seuls processus possibles impliquant les neutrons, la section efficace totale $\sigma = \sigma_{el} + \sigma_{abs}$ correspond à une certaine distance caractéristique $\xi = (n\sigma)^{-1}$, où n est la densité volumique des diffuseurs, c'est-à-dire le nombre d'atomes de

modérateur par unité de volume. On montre sans peine que la probabilité qu'un neutron puisse se propager sur une distance x sans subir de collision ou d'absorption est

$$p(x) = \mathrm{e}^{-x/\xi} = \mathrm{e}^{-\rho\sigma x} \tag{16.53}$$

La question qui se pose est la suivante : quelle fraction du faisceau incident a réussi à franchir le mur, et quelle est l'énergie moyenne des neutrons qui l'ont franchi ?

Pour répondre à cette question, nous pouvons simuler le système de la manière suivante :

1. On suppose qu'un neutron a une position initiale $\mathbf{r} = (0,0,0)$ et une quantité de mouvement initiale $\mathbf{p} = p\mathbf{x}$. Le mur est perpendiculaire à l'axe des x et débute à $x = 0$ pour se terminer à $x = h$.
2. Pour cette paire $(\mathbf{r}, \mathbf{p})$, on tire une valeur aléatoire de la distance de propagation d selon la distribution exponentielle. On suppose que le neutron se propage librement sur cette distance d, et on met à jour la position du neutron : $\mathbf{r} \to \mathbf{r} + \hat{\mathbf{p}}d$, où $\hat{\mathbf{p}}$ est le vecteur unitaire dans la direction de $\mathbf{p}$.
3. Si la nouvelle position $\mathbf{r}$ est au-delà du mur ($x > h$), on arrête la simulation pour ce neutron tout en mesurant les quantités physiques requises, et on retourne à l'étape 1 pour en simuler un autre. On procède de même si la nouvelle position est en deçà du mur ($x < 0$), ce qui ne saurait se produire lors de la première passe.
4. À cette nouvelle position, le neutron doit subir une interaction. La probabilité que le neutron soit absorbé est $\Gamma_{\text{abs}} = \sigma_{\text{abs}}/\sigma$ (ce qu'on appelle dans le jargon le *rapport d'embranchement* pour ce processus), alors que la probabilité qu'il soit diffusé de manière élastique est $\Gamma_{\text{el}} = \sigma_{\text{el}}/\sigma = 1 - \Gamma_{\text{abs}}$. On tire alors un aléatoire x uniforme dans $[0,1]$, et on suppose que le neutron est absorbé si $x < \Gamma_{\text{abs}}$; c'est alors la fin de l'histoire de ce neutron, on collecte la statistique et on retourne à l'étape 1 ci-dessus pour traiter un autre neutron.
5. Si, au contraire, $x > \Gamma_{\text{abs}}$, on suppose qu'il est diffusé par un noyau au repos. On se déplace alors dans le repère du centre de masse, dans lequel la quantité de mouvement du neutron devient
$$\tilde{\mathbf{p}} = \mathbf{p} - m_n \mathbf{v}_{\text{cm}} = \mathbf{p} - \frac{m_n}{m_n + m_a}\mathbf{p} = \frac{m_a}{m_n + m_a}\mathbf{p} \tag{16.54}$$
où m_n et m_a sont les masses du neutron et du noyau, respectivement. On choisit ensuite une direction aléatoire pour le neutron diffusé dans le référentiel du centre de masse : on suppose que la section différentielle de diffusion est indépendante de la direction, comme pour la diffusion de sphères dures (ceci est une approximation, valable quand l'énergie du neutron n'est pas trop grande). Puisque la mesure de l'angle solide est $\mathrm{d}\Omega = \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\varphi = \mathrm{d}\cos\theta\,\mathrm{d}\varphi$, cela revient à tirer un uniforme dans $[0, 2\pi]$ pour φ, et un autre uniforme dans $[-1, 1]$ pour $z = \cos\theta$. Une fois cette direction choisie, la quantité de mouvement $\tilde{\mathbf{p}}'$ du neutron dans le repère du centre de masse est fixée par la conservation de l'énergie : $|\tilde{\mathbf{p}}'| = |\tilde{\mathbf{p}}|$ et la quantité de mouvement du neutron dans le repère du laboratoire est simplement
$$\mathbf{p}' = \tilde{\mathbf{p}}' + \frac{m_n}{m_n + m_a}\mathbf{p} \tag{16.55}$$
On met alors à jour la quantité de mouvement $\mathbf{p} \to \mathbf{p}'$ et on retourne à l'étape 2 ci-dessus. Notons que comme l'énergie du neutron est maintenant plus faible (puisqu'il a cédé de l'énergie au modérateur), la section d'absorption σ_{abs} sera généralement différente (cette quantité dépend de l'énergie E du neutron).

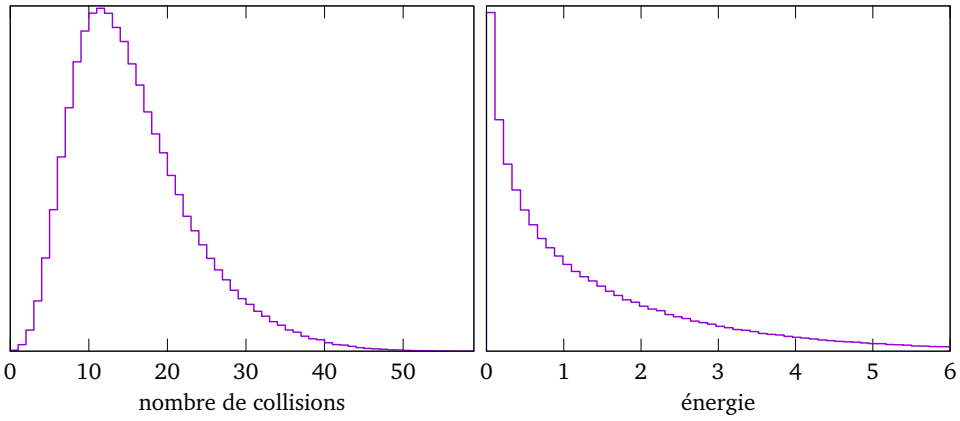


FIGURE 16.6
Histogramme du nombre de collisions (à gauche) et de l'énergie des neutrons à la sortie du modérateur (à droite).

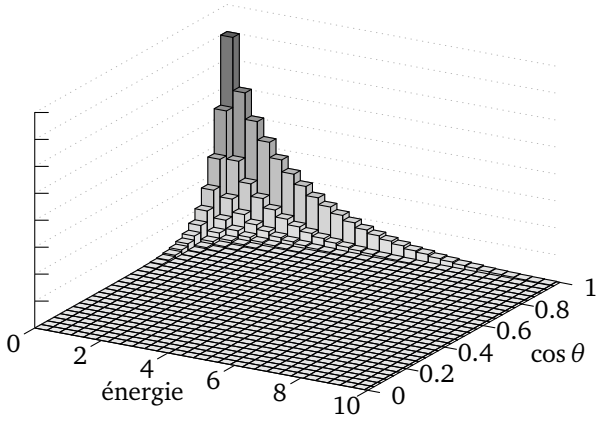


FIGURE 16.7
Histogramme conjoint de l'énergie des neutrons à la sortie du modérateur et du cosinus de l'angle avec la normale.

6. On arrête la simulation lorsqu'un nombre prédéfini de neutrons a été traité.

Les figures 16.6 et 16.7 illustrent les résultats d'une simulation menée sur 10^6 particules, traversant un modérateur d'épaisseur h. On a supposé que le modérateur avait une masse 10 fois plus grande que celle du neutron, et que les longueurs de pénétrations associées aux processus élastiques et d'absorption étaient respectivement données par

$$\xi_{\text{él.}}^{-1} = 0.04 \qquad \xi_{\text{abs.}}^{-1} = \frac{0.0003}{\sqrt{\mathrm{E}}} \tag{16.56}$$

et que l'énergie des neutrons incidents est $\mathrm{E}_0 = 10$ (unités arbitraires). L'épaisseur du modérateur est $h = 1\,000$. Pour ces paramètres, le nombre le plus probable de collisions élastiques est 12, et 64,6 % des neutrons traversent le modérateur, le reste (35,4%) étant absorbé. Plus les neutrons subissent de collisions, plus leur énergie diminue et plus la probabilité d'absorption augmente. L'histogramme conjoint (fig. 16.7) nous apprend que plus l'énergie des neutrons est basse à la sortie, plus leur étalement angulaire est grand, alors que les neutrons de haute énergie sont fortement concentrés dans la direction normale au mur.

CHAPITRE 17

ÉQUATIONS NON LINÉAIRES ET OPTIMISATION

A Équations non linéaires à une variable

Le problème traité dans cette section consiste à résoudre une équation non linéaire, ce qui revient à chercher la ou les racines d'une fonction $y(x)$:

$$y(x) = 0 \tag{17.1}$$

La fonction $y(x)$ peut avoir une forme analytique explicite ou être le résultat d'un autre calcul numérique ne correspondant pas à une forme connue.

A.1 Cadrage et dichotomie

Une méthode simple et robuste pour trouver une racine de f consiste premièrement à *cadrer* la racine, c'est-à-dire à trouver deux points x_1 et x_2 entre lesquels au moins une racine existe. Cela ne peut se faire que si on fait l'hypothèse que la fonction est continue, de sorte que si $y_1 = y(x_1) < 0$ et $y_2 = y(x_2) > 0$, on a l'assurance qu'une racine $x^\star$ existe dans l'intervalle $[x_1, x_2]$. Nous n'expliquerons pas ici d'algorithmes particuliers pour trouver les valeurs x_1 et x_2 ; nous supposons plutôt que, selon la fonction désirée, ces valeurs peuvent se trouver sans trop de difficulté.

La *méthode de dichotomie* consiste à diviser l'intervalle de recherche en deux parties égales, et ce de manière répétée, jusqu'à ce que la largeur de l'intervalle soit comparable à la précision recherchée ϵ sur la position de la racine. Plus précisément :

1. On calcule $y(\bar{x})$, où $\bar{x} = (x_1 + x_2)/2$.
2. Si $y(\bar{x})$ est du même signe que $y(x_1)$, alors on met à jour $x_1 \leftarrow \bar{x}$, sinon on fait $x_2 \leftarrow \bar{x}$ et on recommence à l'étape 1.
3. On arrête la procédure lorsque $x_2 - x_1 < \epsilon$.

La méthode de dichotomie est sûre, mais inefficace : elle converge linéairement. Par là on veut dire que la différence $\delta_n \coloneqq \bar{x} - x^\star$ entre la solution véritable et la meilleure estimation de la solution à l'étape n de la procédure se comporte comme

$$\delta_{n+1} \sim \frac{1}{2}\delta_n \tag{17.2}$$

En général, le *degré de convergence* η d'une méthode itérative est défini par

$$\delta_{n+1} \sim \mathrm{A}\delta_n^\eta \tag{17.3}$$

où A est une constante ; plus l'exposant η est élevé, plus la convergence est rapide.

A.2 Méthode de la fausse position

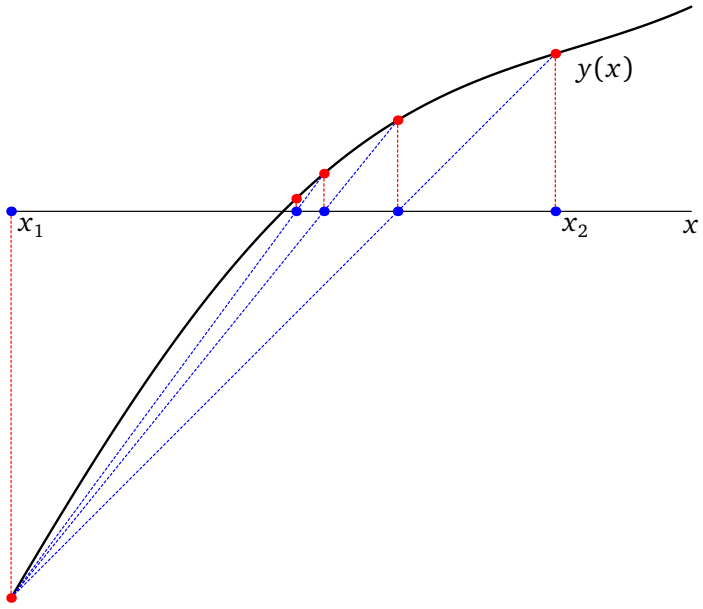


FIGURE 17.1
Méthode de la fausse position.

Une amélioration sensible par rapport à la méthode de dichotomie est la méthode de la *fausse position*, qui consiste non pas à adopter la moyenne $\bar{x} = (x_1 + x_2)/2$ comme estimation de la racine à chaque étape, mais plutôt l'intersection avec l'axe des y de l'interpolant linéaire entre les deux points x_1 et x_2 :

$$\frac{\bar{x} - x_2}{x_1 - x_2} y_1 + \frac{\bar{x} - x_1}{x_2 - x_1} y_2 = 0 \Longrightarrow \bar{x} = \frac{x_1 y_2 - x_2 y_1}{y_2 - y_1} \tag{17.4}$$

On met ensuite là jour les valeurs de x_1 et x_2 comme dans la méthode de dichotomie (voir la figure 17.1). La méthode de la fausse position est robuste, en ce sens que la racine est toujours cadrée et donc on ne peut pas la rater. Son degré de convergence n'est pas bien défini, mais la méthode converge en général plus rapidement que la méthode de dichotomie, cette dernière n'étant plus rapide que dans quelques cas pathologiques.

Problème 17.1 :
Illustrez schématiquement une fonction dont la racine serait trouvée plus rapidement par la méthode de dichotomie que par la méthode de la fausse position.

A.3 Méthode de la sécante

La méthode de la sécante est une variation de la méthode de la fausse position, dans laquelle on procède à une interpolation ou extrapolation linéaire pour trouver une estimation de la position de

la racine, même si cette nouvelle estimation x_{n+1} est en dehors de l'intervalle formé par les deux points précédents :

$$\boxed{x_{n+1} = \frac{x_{n-1}y_n - x_n y_{n-1}}{y_n - y_{n-1}}} \tag{17.5}$$

À la différence de la méthode de la fausse position, on base chaque nouvelle estimation x_{n+1} de la racine sur les deux estimations précédentes (x_n et x_{n-1}), au lieu de rejeter l'une des deux frontières de l'intervalle, qui une fois sur deux en moyenne correspond à x_{n-1}.

On montre que le degré de convergence de la méthode de la sécante est le nombre d'or :

$$\eta = \frac{1+\sqrt{5}}{2} = 1.618034\ldots \tag{17.6}$$

Par contre, sa convergence n'est pas garantie, car la racine n'est pas systématiquement cadrée.

A.4 Méthode d'interpolation quadratique inverse

Au lieu d'utiliser les deux valeurs précédentes, on peut utiliser les trois valeurs précédentes afin de lisser une parabole décrivant la fonction réciproque f^{-1}. Plus précisément, en partant de trois paires (x_n, y_n), (x_{n-1}, y_{n-1}) et (x_{n-2}, y_{n-2}), l'interpolant de Lagrange pour la fonction $f^{-1}(y)$ est

$$f^{-1}(y) \approx \frac{(y-y_{n-1})(y-y_n)x_{n-2}}{(y_{n-2}-y_{n-1})(y_{n-2}-y_n)} + \frac{(y-y_{n-2})(y-y_n)x_{n-1}}{(y_{n-1}-y_{n-2})(y_{n-1}-y_n)} + \frac{(y-y_{n-1})(y-y_{n-2})x_n}{(y_n-y_{n-1})(y_n-y_{n-2})}$$

et donc la prédiction pour x_{n+1} s'obtient en posant $y = 0$:

$$x_{n+1} = \frac{y_{n-1}y_n x_{n-2}}{(y_{n-2}-y_{n-1})(y_{n-2}-y_n)} + \frac{y_{n-2}y_n x_{n-1}}{(y_{n-1}-y_{n-2})(y_{n-1}-y_n)} + \frac{y_{n-1}y_{n-2}x_n}{(y_n-y_{n-1})(y_n-y_{n-2})}$$

Cette méthode est utilisée afin d'accélérer la convergence de la méthode de la sécante et est habituellement combinée à la méthode de la sécante et à la méthode de dichotomie dans une stratégie robuste qui porte le nom de *méthode de Brent*. Cette dernière effectue des tests à chaque étape afin de garder la racine cadrée. [1].

A.5 Méthode de Newton-Raphson

L'une des méthodes numériques les plus anciennes a été proposée indépendamment par I. Newton et J. Raphson à la fin du XVII^e siècle pour trouver les racines d'une fonction. Elle se base sur le calcul différentiel et non sur un cadrage de la racine (voir la figure 17.2) :

1. On choisit une estimation initiale x_1 de la racine.
2. On calcule la valeur de la fonction $y_1 = y(x_1)$ et de sa dérivée $y_1' = y'(x_1)$.
3. On trace la droite de pente y_1' passant par (x_1, y_1) et on trouve l'intersection de cette droite avec l'axe des x. Il s'agit de la prochaine valeur x_2 de l'estimateur :

$$x_2 = x_1 - \frac{y_1}{y_1'} \quad \text{ou, à l'étape } n, \quad \boxed{x_{n+1} = x_n - \frac{y_n}{y_n'}} \tag{17.7}$$

1. Pour plus de détails, voir l'article de Wikipédia

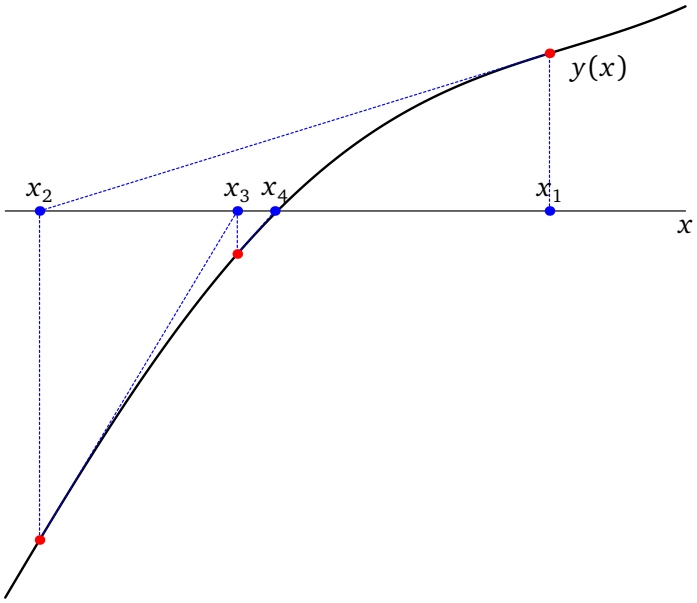


FIGURE 17.2
Méthode de Newton-Raphson pour la recherche des racines.

4. On répète jusqu'à convergence.

On montre que le degré de convergence de la méthode de Newton-Raphson est $\eta = 2$. Cela signifie qu'à l'approche de la solution, le nombre de chiffres significatifs de l'estimateur double à chaque résolution! Par contre, la méthode n'est pas robuste : si on la démarre trop loin de la solution, elle peut facilement diverger ou rester sur un cycle limite. La dynamique discrète définie par l'équation (17.7) peut être complexe et intéressante en soi.

Problème 17.2 :
Montrez que le degré de convergence de la méthode de Newton-Raphson est $\eta = 2$.

Solution

Posons $x_n = x^\star + \delta_n$, où $x^\star$ est la racine. L'équation de récurrence de la méthode de Newton s'écrit alors ainsi :

$$\delta_{n+1} = \delta_n - \frac{y(x^\star + \delta_n)}{y'(x^\star + \delta_n)}$$

Développons le deuxième terme au deuxième ordre en δ_n au numérateur et au premier ordre au

dénominateur ; dans ce qui suit la fonction y et ses dérivées sont évaluées à $x^\star$, et donc $y = 0$:

$$\begin{aligned}
\delta_{n+1} &\approx \delta_n - \frac{y + y'\delta_n + \frac{1}{2}y''\delta_n^2}{y' + y''\delta_n} \\
&= \delta_n \left\{ 1 - \frac{y' + \frac{1}{2}y''\delta_n}{y' + y''\delta_n} \right\} \\
&= \delta_n \left\{ 1 - \frac{1 + \frac{y''}{2y'}\delta_n}{1 + \frac{y''}{y'}\delta_n} \right\} \\
&\approx \delta_n \left\{ 1 - \left(1 + \frac{y''}{2y'}\delta_n \right) \left(1 - \frac{y''}{y'}\delta_n \right) \right\} \\
&\approx \frac{y''}{2y'}\delta_n^2
\end{aligned}$$

Nous avons utilisé le développement du binôme pour le dénominateur et nous avons négligé les termes d'ordre δ_n^2 à l'intérieur de l'accolade.

Boîte à outils

La méthode de Brent est disponible via `scipy.optimize.brentq()`. On doit spécifier un intervalle qui cadre la racine ainsi que les précisions requises).

B Équations non linéaires à plusieurs variables

La résolution d'un système d'équations non linéaires couplées est un problème pour lequel il n'existe pas de méthode sûre et générale. Considérons par exemple le cas de deux variables x et y obéissant à deux équations non linéaires simultanées qu'on peut toujours mettre sous la forme

$$f(x, y) = 0 \qquad g(x, y) = 0 \tag{17.8}$$

où f et g sont des fonctions qui définissent le problème.

L'équivalent de ces équations en dimension 1 est une équation unique $y(x) = 0$; si cette fonction est continue et que deux valeurs $y(x_1)$ et $y(x_2)$, l'une positive et l'autre négative, sont connues, alors une racine existe certainement. On ne peut affirmer l'existence d'une solution avec autant de généralité en dimension 2 : chacune des équations (17.8) définit une certaine courbe de niveau (une pour f, l'autre pour g). Une solution existe seulement si ces deux courbes de niveau se croisent. Il est également possible que l'une ou l'autre des équations (17.8) n'ait pas de solution, c'est-à-dire que la courbe de niveau associée à la valeur 0 n'existe pas.

Une façon quelque peu brutale de tenter la solution des équations (17.8) est de les traiter comme une succession de problèmes unidimensionnels : On résout la première équation en fonction de x pour une valeur donnée de y, qui est alors traité comme un paramètre. L'un des algorithmes à une variable décrits ci-dessus est mis à contribution pour cela. On obtient ainsi une fonction $x(y)$, qu'on injecte ensuite dans la deuxième équation : $g(x(y),y)=0$. Cette dernière équation est alors résolue, encore une fois par un algorithme à une variable. Cette approche a le mérite de la simplicité, mais est inefficace, car elle ne traite pas les deux variables sur un pied d'égalité, et accordera trop de précision à des valeurs intermédiaires de y.

B.1 Méthode de Newton-Raphson

Nous allons décrire la méthode de Newton-Raphson appliquée à N équations à N variables, qu'on écrira comme

$$\mathbf{f}(\mathbf{x})=0 \quad \text{ou} \quad f_i(\mathbf{x})=0 \qquad i=1,\ldots,\mathrm{N} \tag{17.9}$$

Commençons par un point de départ $\mathbf{x}_0$. Autour de ce point, les fonctions f_i admettent un développement de Taylor :

$$f_i(\mathbf{x})=f_i(\mathbf{x}_0)+\sum_j \left.\frac{\partial f_i}{\partial x_j}\right|_{\mathbf{x}_0} \delta x_j + \mathcal{O}(\delta\mathbf{x}^2) \qquad\qquad \delta\mathbf{x}\coloneqq\mathbf{x}-\mathbf{x}_0 \tag{17.10}$$

On note habituellement la matrice des dérivées premières (le jacobien) comme

$$\mathrm{J}_{ij}\coloneqq\frac{\partial f_i}{\partial x_j} \tag{17.11}$$

et donc l'équation non linéaire peut s'écrire, à cet ordre d'approximation, comme

$$0=f_i(\mathbf{x}_0)+\mathrm{J}_{ij}\delta x_j \quad \text{ou} \quad \mathbf{J}(\mathbf{x}_0)\delta\mathbf{x}=-\mathbf{f}(\mathbf{x}_0) \tag{17.12}$$

ce qui constitue un système linéaire simple qu'on résout dans le but d'obtenir une nouvelle estimation de la racine cherchée :

$$\delta\mathbf{x}=-\mathbf{J}^{-1}(\mathbf{x}_0)\mathbf{f}(\mathbf{x}_0) \tag{17.13}$$

Répéter cette procédure revient à poser la relation de récurrence suivante :

$$\boxed{\mathbf{x}_{n+1}=\mathbf{x}_n-\mathbf{J}^{-1}(\mathbf{x}_n)\mathbf{f}(\mathbf{x}_n)} \tag{17.14}$$

Cette relation est la généralisation à plusieurs variables de l'équation unidimensionnelle (17.7).

B.2 Méthode de Broyden

La méthode de Broyden est une généralisation à plusieurs variables de la méthode de la sécante. Elle consiste à construire une suite d'approximations $\mathbf{J}_n$ ($n=0,1,2\ldots$) au jacobien $\partial\mathbf{f}/\partial\mathbf{x}$, en utilisant la relation suivante, qui est une approximation aux différences :

$$\mathbf{J}_n(\mathbf{x}_n-\mathbf{x}_{n-1})=\mathbf{f}_n-\mathbf{f}_{n-1} \tag{17.15}$$

où $\{\mathbf{x}_n\}$ est la suite de points obtenus par la méthode et $\mathbf{f}_n \coloneqq \mathbf{f}(\mathbf{x}_n)$. Cette relation est pratiquement la même que (17.14), si on écrit cette dernière comme $\mathbf{J}(\mathbf{x}_{n-1})(\mathbf{x}_n - \mathbf{x}_{n-1}) = -\mathbf{f}_{n-1}$ et en posant $\mathbf{f}_n = 0$. Définissons $\delta\mathbf{x}_n = \mathbf{x}_n - \mathbf{x}_{n-1}$ et $\delta\mathbf{f}_n = \mathbf{f}_n - \mathbf{f}_{n-1}$. L'équation (17.15) prend alors la forme plus concise $\mathbf{J}_n \delta\mathbf{x}_n = \delta\mathbf{f}_n$.

La relation (17.15) ne détermine pas complètement la matrice $\mathbf{J}_n$. Il nous faut donc une prescription explicite pour passer de $\mathbf{J}_n$ à $\mathbf{J}_{n+1}$ qui soit compatible avec la relation (17.15). Dans la méthode de Broyden, cette prescription est la suivante :

$$\mathbf{J}_n = \mathbf{J}_{n-1} + \frac{(\delta\mathbf{f}_n - \mathbf{J}_{n-1}\delta\mathbf{x}_n)\delta\mathbf{x}_n^{\mathrm{T}}}{\delta\mathbf{x}_n^{\mathrm{T}}\delta\mathbf{x}_n} \tag{17.16}$$

où le numérateur est le produit d'un vecteur-colonne par un vecteur-rangée, donc une matrice. On vérifie facilement que la prescription (17.16) respecte bien la condition (17.15).

La procédure de Broyden est donc la suivante :

1. On pose un point de départ $\mathbf{x}_0$.
2. On calcule une valeur initiale $\mathbf{J}_0$ du jacobien à $\mathbf{x}_0$ en procédant à une approximation aux différences.
3. On pose ensuite $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{J}_0^{-1}\mathbf{f}_0$, ce qui nous donne $\delta\mathbf{x}_1$ et $\delta\mathbf{f}_1$.
4. On calcule ensuite la prochaine valeur $\mathbf{J}_1$ du jacobien en appliquant l'éq. (17.16) pour $n = 1$. En fait, on accomplit en boucle la transformation suivante pour $n = 1, 2, 3, \ldots$:

$$\boxed{\mathbf{J}_n = \mathbf{J}_{n-1} + \frac{(\delta\mathbf{f}_n - \mathbf{J}_{n-1}\delta\mathbf{x}_n)\delta\mathbf{x}_n^{\mathrm{T}}}{\delta\mathbf{x}_n^{\mathrm{T}}\delta\mathbf{x}_n} \qquad \text{et} \qquad \mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J}_n^{-1}\mathbf{f}_n} \tag{17.17}$$

   Remarquons qu'il n'est pas nécessaire d'inverser $\mathbf{J}_n$ pour cela : on peut simplement résoudre le système linéaire $\mathbf{J}_n\delta\mathbf{x}_{n+1} = \mathbf{f}_n$ pour $\delta\mathbf{x}_{n+1}$ et poser $\mathbf{x}_{n+1} = \mathbf{x}_n + \delta\mathbf{x}_{n+1}$.
5. On arrête lorsque $\delta\mathbf{x}_n$ est suffisamment petit.

Une variante de la procédure consiste à itérer non pas $\mathbf{J}_n$, mais son inverse $\mathbf{I}_n = \mathbf{J}_n^{-1}$. On montre que, dans ce cas, la relation de récurrence est

$$\boxed{\mathbf{I}_n = \mathbf{I}_{n-1} + \frac{(\delta\mathbf{x}_n - \mathbf{I}_{n-1}\delta\mathbf{f}_n)\delta\mathbf{x}_n^{\mathrm{T}}\mathbf{I}_{n-1}}{\delta\mathbf{x}_n^{\mathrm{T}}\mathbf{I}_{n-1}\delta\mathbf{f}_n} \qquad \text{et} \qquad \mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{I}_n\mathbf{f}_n} \tag{17.18}$$

Cette variante ne nécessite pas d'inverser une matrice à chaque étape.

La méthode de Broyden est l'une des plus utilisées dans la résolution d'équations non linéaires à plusieurs variables.

B.3 Méthode itérative directe

Il arrive dans plusieurs applications que les équations non linéaires aient la forme suivante :

$$\mathbf{x} = \mathbf{K}(\mathbf{x}|\alpha) \tag{17.19}$$

où α désigne un paramètre ou un ensemble de paramètres. Ceci n'est évidemment pas restrictif, car tout système non linéaire peut être mis sous cette forme. C'est le cas notamment dans l'approximation du champ moyen en physique statistique ou dans le problème à N corps en mécanique quantique. Les paramètres α dans ce cas pourraient être la température, ou encore la force d'une interaction entre particules. Ces équations pourraient bien sûr être traitées par la méthode de Newton-Raphson, avec la fonction $\mathbf{f}(\mathbf{x}) = \mathbf{K}(\mathbf{x}|\alpha) - \mathbf{x}$. Il est cependant fréquent de leur appliquer une méthode de solution beaucoup plus simple, qui suppose que l'on connaît à l'avance une solution dans un cas limite (par exemple pour une valeur précise α_0 des paramètres). On suppose alors que, si α n'est pas trop différent de α_0, on peut adopter comme première approximation la solution correspondante $\mathbf{x} = \mathbf{x}_0$ et appliquer la relation de récurrence suivante :

$$\mathbf{x}_{n+1} = \mathbf{K}(\mathbf{x}_n|\alpha) \tag{17.20}$$

L'application répétée de cette relation, si elle converge, mène effectivement à la solution recherchée. En pratique, si la solution est requise pour plusieurs valeurs de α, il est avantageux de procéder par *proximité*, en mettant sur pied une boucle sur α qui recycle la solution $\mathbf{x}(\alpha)$ de l'étape précédente comme point de départ de la nouvelle recherche pour la valeur suivante de α.

Cette méthode itérative directe est simple, mais converge moins rapidement que la méthode de Newton-Raphson.

Problème 17.3 :
Appliquez la méthode itérative directe à la solution de l'équation transcendante $x = \frac{1}{4}\,\mathrm{e}^x$, en adoptant comme point de départ $x = 0$ (écrivez un court programme Python). Quel est le degré de convergence de cette méthode ?

Solution

On applique la récurrence $x_{n+1} = \frac{1}{4}\,\mathrm{e}^{x_n}$ avec $x_0 = 0$, et on trouve la suite

$$\{x_n\} = 0\,,\ 0.25\,,\ 0.321006\,,\ 0.344629\,,\ 0.352866\,,\ 0.355785\,,\ 0.356825,\ \cdots,0.357402\ldots \tag{17.21}$$

Les erreurs successives $\delta x_n \coloneqq x_n - x^\star$, où $x^\star$ est la solution exacte, se comportent manifestement comme

$$\delta x_{n+1} = \mathrm{K}'(x^\star)\delta x_n \tag{17.22}$$

et donc le degré de convergence est linéaire.

Problème 17.4 :
Supposons qu'on veuille appliquer la méthode itérative directe à la solution de l'équation $x = \lambda x(1-x)$, en adoptant comme point de départ $x = 0$ et en faisant progresser λ de 0 jusqu'à 4. Quel problème rencontrerions-nous ? L'application $x \mapsto \lambda x(1-x)$ porte le nom de *carte logistique*.

Boîte à outils

La fonction `scipy.optimize.root()` peut appliquer plusieurs méthodes afin de trouver la racine d'une fonction vectorielle à N variables. On doit spécifier le point de départ (un vecteur à N composantes). La méthode utilisée est spécifiée par le mot-clé `method`. L'une des possibilités est la méthode de Broyden (`method = broyden1`), mais ce n'est pas la méthode par défaut.

C Optimisation d'une fonction

Les problèmes d'optimisation sont parmi les plus fréquents rencontrés en calcul scientifique. Dans plusieurs cas, il s'agit de trouver le minimum d'une fonction différentiable E($\mathbf{x}$) à N variables. Certaines méthodes supposent une connaissance des dérivées de E, qui doivent être calculées séparément ; d'autres ne requièrent pas la dérivée, ce qui est crucial si cette dernière est inconnue, par exemple si E est le résultat d'un calcul numérique complexe. Dans certains problèmes, on cherche un point de dérivée nulle de E, qui n'est pas nécessairement un minimum, mais pourrait être un maximum ou un point d'inflexion (un point de selle, en plusieurs variables).

D'autres problèmes, comme le problème du commis voyageur, ne peuvent pas être formulés en fonction d'une ou plusieurs variables continues, mais sont plutôt de nature discrète.

Dans tous les cas, il faut distinguer les minimums locaux du minimum global. La plupart des méthodes ne peuvent trouver que des minimums locaux. La recherche du minimum global est un problème très difficile qui n'a pas de solution certaine. La méthode du recuit simulé, vue en fin de section, est la meilleure stratégie dans ce cas.

Une condition nécessaire à l'existence d'un minimum d'une fonction E est la condition de dérivée nulle :

$$\mathbf{f}(\mathbf{x}) \coloneqq \frac{\partial \mathrm{E}}{\partial \mathbf{x}} = 0 \tag{17.23}$$

On peut donc réduire le problème du minimum local à celui de la recherche de racines. Par contre, l'inverse n'est pas vrai : une fonction vectorielle $\mathbf{f}(\mathbf{x})$ générale comme celle traitée à la section précédente n'est pas nécessairement le gradient d'une fonction. Il faudrait pour cela qu'elle soit intégrable. Par exemple, pour N = 3 variables, il faudrait que le rotationnel du vecteur $\mathbf{f}(\mathbf{x})$ soit nul. Par contre, le fait que $\mathbf{f}$ soit le gradient d'une fonction E nous permet de mettre en place des algorithmes plus efficaces.

C.1 Méthode de Newton-Raphson

La méthode de Newton-Raphson requiert de pouvoir calculer non seulement les premières dérivées $\mathbf{f} = \partial \mathrm{E}/\partial \mathbf{x}$, mais aussi les dérivées secondes

$$\mathbf{H} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \frac{\partial^2 \mathrm{E}}{\partial \mathbf{x} \partial \mathbf{x}} \quad \text{ou encore} \quad \mathrm{H}_{ij} = \frac{\partial \mathrm{E}}{\partial x_i \partial x_j} \tag{17.24}$$

cette matrice porte le nom de *matrice hessienne*, ou simplement *hessien*. (dans le contexte d'une recherche de racines, l'équivalent de cette matrice était notée $\mathbf{J}$). Si on peut produire le hessien,

alors la méthode de Newton-Raphson peut être appliquée comme décrit à la section B.1. Lorsque la forme des dérivées de la fonction E n'est pas connue explicitement, la méthode peut quand même être appliquée à condition de procéder à un calcul numérique des dérivées. Par exemple, on pourrait procéder de la manière suivante : une fonction de N variables à un point donné comporte 1 valeur, N dérivées premières et $\frac{1}{2}\mathrm{N}(\mathrm{N}+1)$ dérivées secondes distinctes, soit au total $1+\mathrm{N}(\mathrm{N}+3)/2$ quantités qui doivent être estimées à chaque itération de la méthode. Le calcul numérique de ces quantités requiert donc un nombre égal de valeurs E calculées au point $\mathbf{x}$ et dans son voisinage immédiat : il s'agit alors de lisser une forme quadratique sur ces valeurs de la fonction E

$$\mathrm{E}(\mathbf{x}) = \mathrm{E}(\mathbf{x}_0) + \mathbf{f}\cdot(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^{\mathrm{T}}\mathbf{H}(\mathbf{x}-\mathbf{x}_0) \tag{17.25}$$

afin de déterminer les N composantes de $\mathbf{f}$ et les $\frac{1}{2}\mathrm{N}(\mathrm{N}+1)$ composantes indépendantes de $\mathbf{H}$. Si la fonction E est longue à calculer et que N est grand, ceci n'est pas la meilleure stratégie, car le nombre d'évaluations de E se comporte comme N^2. On préfèrera dans ce cas les méthodes de quasi-Newton décrites ci-dessous.

C.2 Méthodes de quasi-Newton

Les méthodes de quasi-Newton sont semblables à la méthode de Broyden, mais appliquées à la minimisation d'une fonction. L'idée générale est de commencer une procédure itérative à l'aide d'une valeur très imparfaite du hessien, et de mettre à jour le hessien à chaque étape en même temps que la position $\mathbf{x}$.

À chaque étape de la méthode le gradient $\mathbf{f}_n = \mathbf{f}(\mathbf{x}_n)$ doit être calculé à l'aide d'une approximation aux différences finies :

$$f_{n,i} = \frac{\mathrm{E}(\mathbf{x}_n + \epsilon\mathbf{e}_i) - \mathrm{E}(\mathbf{x}_n - \epsilon\mathbf{e}_i)}{2\epsilon} \tag{17.26}$$

où $\mathbf{e}_i$ est le vecteur unitaire dans la direction n$^{\underline{o}}$ i et ϵ un pas de différentiation petit. Ensuite, la position $\mathbf{x}$ est mise à jour à l'aide de la formule

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{I}_n\mathbf{f}_n \qquad\qquad \mathbf{I}_n \coloneqq \mathbf{H}_n^{-1} \tag{17.27}$$

Enfin, dans la méthode dite de DFP (Davidson-Fletcher-Powell) le hessien inverse $\mathbf{I}_n$ est mis à jour à l'aide de la formule suivante :

$$\boxed{\mathbf{I}_n = \mathbf{I}_{n-1} + \frac{\delta\mathbf{x}_n\delta\mathbf{x}_n^{\mathrm{T}}}{\delta\mathbf{x}_n^{\mathrm{T}}\delta\mathbf{f}_n} - \frac{\mathbf{I}_{n-1}\delta\mathbf{f}_n\delta\mathbf{f}_n^{\mathrm{T}}\mathbf{I}_{n-1}}{\delta\mathbf{f}_n^{\mathrm{T}}\mathbf{I}_{n-1}\delta\mathbf{f}_n}} \tag{17.28}$$

On remarque que cette formule préserve la relation $\delta\mathbf{x}_n = \mathbf{I}_n\delta\mathbf{f}_n$:

$$\mathbf{I}_n\delta\mathbf{f}_n = \mathbf{I}_{n-1}\delta\mathbf{f}_n + \delta\mathbf{x}_n - \mathbf{I}_{n-1}\delta\mathbf{f}_n = \delta\mathbf{x}_n \tag{17.29}$$

La méthode dite BFGS (Broyden-Fletcher-Goldfarb-Shanno) modifie la relation (17.28) de la manière suivante :

$$\boxed{\mathbf{I}_n = \mathbf{I}_{n-1} + \frac{\delta\mathbf{x}_n\delta\mathbf{x}_n^{\mathrm{T}}}{\delta\mathbf{x}_n^{\mathrm{T}}\delta\mathbf{f}_n} - \frac{\mathbf{I}_{n-1}\delta\mathbf{f}_n\delta\mathbf{f}_n^{\mathrm{T}}\mathbf{I}_{n-1}}{\delta\mathbf{f}_n^{\mathrm{T}}\mathbf{I}_{n-1}\delta\mathbf{f}_n} + \delta\mathbf{f}_n^{\mathrm{T}}\mathbf{I}_{n-1}\delta\mathbf{f}_n\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}} \tag{17.30}$$

où

$$\mathbf{u}_n = \frac{\delta \mathbf{x}_n}{\delta \mathbf{x}_n^{\mathrm{T}} \delta \mathbf{f}_n} - \frac{\mathbf{I}_{n-1} \delta \mathbf{f}_n}{\delta \mathbf{f}_n^{\mathrm{T}} \mathbf{I}_{n-1} \delta \mathbf{f}_n} \tag{17.31}$$

Encore une fois, on constate que la relation $\delta \mathbf{x}_n = \mathbf{I}_n \delta \mathbf{f}_n$ est préservée, car le vecteur $\mathbf{u}_n$ est orthogonal à $\delta \mathbf{f}_n$:

$$\mathbf{u}_n^{\mathrm{T}} \delta \mathbf{f}_n = 1 - 1 = 0 \tag{17.32}$$

La méthode BFGS est généralement jugée meilleure de la méthode DFP, au sens de la convergence.

C.3 Méthode de Powell

Lorsqu'elle converge, la méthode de Newton-Raphson le fait rapidement. Cependant, elle n'est pas robuste et la solution trouvée n'est pas nécessairement un minimum. La méthode de Powell, que nous allons maintenant décrire, corrige ces deux travers, au prix d'un plus grand nombre d'évaluations de la fonction E.

Approximation quadratique en dimension 1

La méthode de Powell repose sur la capacité de trouver le minimum d'une fonction E($\mathbf{x}$) dans une direction donnée de l'espace, c'est-à-dire de résoudre un problème de minimisation à une variable. Si on désire minimiser une fonction à une variable $y(x)$, la stratégie générale est la suivante :

1. On doit premièrement *cadrer* le minimum, c'est-à-dire trouver trois points $x_1 < x_2 < x_3$ tels que $y(x_1) > y(x_2)$ et $y(x_3) > y(x_2)$.
2. Ensuite on lisse une parabole sur ces trois points (la solution est unique). Le minimum $x_{\text{min.}}$ de cette parabole est alors l'estimation suivante du minimum $x^\star$ de la fonction. $x_{\text{min.}}$ est forcément compris entre x_1 et x_3, mais peut être situé à droite ou à gauche de x_2.
3. On doit conserver deux autres points, afin d'itérer la procédure. L'une des deux bornes (x_1 ou x_3) sera remplacée, et on retourne à l'étape 2 en utilisant les trois nouveaux points x_i' définis par le tableau suivant :

condition	x_1'	x_2'	x_3'
$x_{\text{min.}} < x_2$ et $y(x_{\text{min.}}) < y(x_2)$	x_1	$x_{\text{min.}}$	x_2
$x_{\text{min.}} < x_2$ et $y(x_{\text{min.}}) > y(x_2)$	$x_{\text{min.}}$	x_2	x_3
$x_{\text{min.}} > x_2$ et $y(x_{\text{min.}}) < y(x_2)$	x_2	$x_{\text{min.}}$	x_3
$x_{\text{min.}} > x_2$ et $y(x_{\text{min.}}) > y(x_2)$	x_1	x_2	$x_{\text{min.}}$

4. On arrête la procédure lorsque $x_{\text{min.}} \approx x^\star$ varie par une valeur inférieure à une précision ϵ.

Directions conjuguées

La méthode de Powell proprement dite comporte les étapes suivantes :

1. Choisir un point de départ $\mathbf{x}_0$ et un ensemble de directions $\mathbf{u}_i$ ($i = 0, 1, \ldots, \mathrm{N}-1$, où N est la dimension de l'espace dans lequel on minimise). Les directions initiales peuvent simplement être les vecteurs unitaires mutuellement orthogonaux définis par chaque axe.
2. Pour `[i=0,i<N,i++]`, trouver la position du minimum $\mathbf{x}_{i+1}$ à partir du point $\mathbf{x}_i$, dans la direction $\mathbf{u}_i$. Autrement dit, on se déplace ainsi : $\mathbf{x}_0 \to \mathbf{x}_1 \to \mathbf{x}_2 \to \cdots \to \mathbf{x}_{\mathrm{N}}$, où le déplacement de $\mathbf{x}_i$ à $\mathbf{x}_{i+1}$ est fait dans la direction $\mathbf{u}_i$, de manière à minimiser la fonction E($\mathbf{x}_i + \lambda \mathbf{u}_i$).

3. On met à jour les étiquettes des directions en sacrifiant $\mathbf{u}_0$: $\mathbf{u}_i \leftarrow \mathbf{u}_{i+1}$.
4. On définit la nouvelle direction $\mathbf{u}_{N-1} = \mathbf{x}_N - \mathbf{x}_0$. C'est la direction de plus grande décroissance de la fonction jusqu'ici.
5. On minimise à partir de $\mathbf{x}_N$ dans la direction $\mathbf{u}_{N-1}$ et remplace $\mathbf{x}_0$ par le point ainsi trouvé.
6. On retourne à l'étape 2, jusqu'à ce que la fonction E ne change plus de manière appréciable.

On montre que cette procédure, si elle est appliquée à une fonction quadratique, produit, après k itérations, une séquence de directions $\mathbf{u}_i$ dont les k derniers membres sont conjugués, au sens de la méthode du gradient conjugué. Cependant, en pratique, les fonctions sur lesquelles on applique la méthode ne sont pas quadratiques et la procédure ci-dessus tend à produire des directions qui éventuellement deviennent linéairement dépendantes, ce qui mène bien sûr à la mauvaise réponse. Cela est dû au fait qu'on jette la direction $\mathbf{u}_0$ au profit de $\mathbf{x}_N - \mathbf{x}_0$, et que toutes les directions ont tendance à «converger» vers une direction commune.

Il faut donc, en pratique, modifier la procédure ci-dessus, ce qui peut se faire de plusieurs façons. La manière proposée par [PTVF07] consiste à éliminer la direction qui produit la plus grande diminution de E, et qui donc est la plus susceptible d'être proche de $\mathbf{x}_N - \mathbf{x}_0$.

C.4 Méthode du simplexe descendant

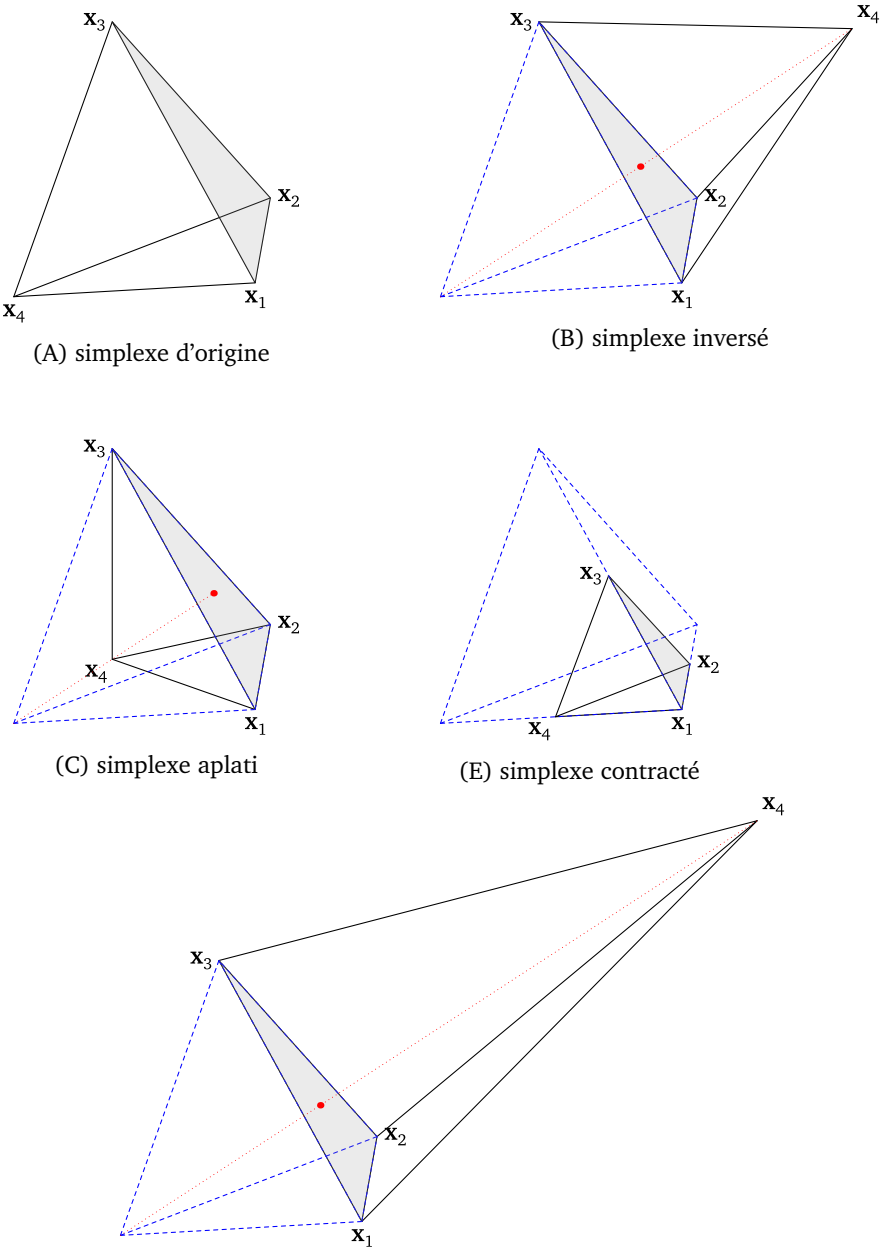
S'il n'est pas possible d'utiliser l'expression des dérivées de la fonction E, ou si la forme de la fonction ne laisse pas espérer qu'une méthode basée sur des dérivées puisse avoir du succès, alors on peut se rabattre sur la méthode du *simplexe descendant* (angl. *downhill simplex*), proposée par Nelder et Mead.

Un *simplexe* en d dimensions est une figure géométrique de dimension d bornée par $d+1$ simplexes de dimension $d-1$. Par exemple :

1. Un simplexe de dimension 0 est un point.
2. Un simplexe de dimension 1 est un segment, borné par deux points.
3. Un simplexe de dimension 2 est un triangle, c'est-à-dire une portion de $\mathbb{R}^2$ bornée par trois segments.
4. Un simplexe de dimension 3 est un tétraèdre irrégulier, c'est-à-dire une portion de $\mathbb{R}^3$ bornée par 4 triangles, et ainsi de suite.

L'idée générale de la méthode du simplexe descendant est de faire évoluer un simplexe dans l'espace $\mathbb{R}^d$ sur lequel est définie la fonction E($\mathbf{x}$). Cette évolution se fait via différentes transformations du simplexe, notamment des réflexions et des contractions, qui amène progressivement le simplexe vers la région où la fonction E($\mathbf{x}$) comporte un minimum local. Certaines de ces transformations sont illustrées à la figure 17.3. Il y a plusieurs variantes de la méthode du simplexe. L'une d'entre elles est décrite ci-dessous :

1. Choisir un simplexe de départ, comportant $n = d+1$ sommets en dimension d. Les positions sont notées $\mathbf{x}_1, \ldots, \mathbf{x}_n$.
2. Trier les sommets dans l'ordre croissant de la fonction f à minimiser : $f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \cdots < f(\mathbf{x}_n)$.
3. Calculer le centre de gravité $\mathbf{x}_0$ de tous les points sauf $\mathbf{x}_n$.



(A) simplexe d'origine

(B) simplexe inversé

(C) simplexe aplati

(E) simplexe contracté

(D) simplexe inversé et allongé

FIGURE 17.3
Transformations d'un simplexe en trois dimensions utilisées dans l'algorithme de Nelder et Mead.

4. Calculer le point réfléchi $\mathbf{x}_r = \mathbf{x}_0 + \alpha(\mathbf{x}_0 - \mathbf{x}_n)$. Ici α est le coefficient de réflexion, habituellement égal à 1. Si $f(\mathbf{x}_1) < f(\mathbf{x}_r) < f(\mathbf{x}_{n-1})$, alors remplacer $\mathbf{x}_n \rightarrow \mathbf{x}_r$ (fig. 17.3B) et retourner à l'étape 2.
5. Si, au contraire, le point réfléchi est le meilleur à date, c'est-à-dire si $f(\mathbf{x}_r) < f(\mathbf{x}_1)$, alors calculer le point étiré $\mathbf{x}_e = \mathbf{x}_0 + \gamma(\mathbf{x}_0 - \mathbf{x}_n)$, où γ est le coefficient d'élongation, habituellement égal à 2. Si $f(\mathbf{x}_e) < f(\mathbf{x}_r)$, alors remplacer $\mathbf{x}_n \rightarrow \mathbf{x}_e$ (fig. 17.3D) et retourner à l'étape 2. Sinon remplacer $\mathbf{x}_n \rightarrow \mathbf{x}_r$ et retourner à l'étape 2.
6. Si le point réfléchi n'est pas meilleur que $\mathbf{x}_{n-1}$ (c'est-à-dire si $f(\mathbf{x}_r) > f(\mathbf{x}_{n-1})$, alors calculer le point contracté $\mathbf{x}_c = \mathbf{x}_0 + \rho(\mathbf{x}_0 - \mathbf{x}_n)$, où ρ est le coefficient de contraction, habituellement -0.5. Si $f(\mathbf{x}_c) < f(\mathbf{x}_n)$, alors remplacer $\mathbf{x}_n \rightarrow \mathbf{x}_c$ (fig. 17.3C) et retourner à l'étape 2.
7. Sinon, alors, contracter le simplexe en remplaçant $\mathbf{x}_i \rightarrow \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1)$ pour tous les points sauf le meilleur ($\mathbf{x}_1$, fig. 17.3E). Ensuite, retourner à l'étape 2. Le coefficient σ (deuxième coefficient de contraction) est typiquement égal à 0.5.

Une condition de convergence est requise. Par exemple, la valeur

$$r = \frac{|f(\mathbf{x}_n) - f(\mathbf{x}_1)|}{|f(\mathbf{x}_n) + f(\mathbf{x}_1)| + 10^{-6}} \tag{17.33}$$

peut être calculée à la fin de l'étape 2 et le programme terminé si cette valeur est inférieure à une précision choisie à l'avance.

D Lissage d'une fonction

D.1 Méthode des moindres carrés et maximum de vraisemblance

Supposons que nous disposions d'un modèle pour décrire une certaine quantité y, qui prend la forme d'une fonction $y(x|a)$, où x représente une variable sur laquelle nous avons le contrôle et a est un ensemble de paramètres du modèle qu'on cherche à déterminer. On suppose en outre qu'un ensemble de mesures entachées d'erreurs a produit N observations (x_i, y_i), avec une erreur σ_i sur la valeur de y_i $(i = 1, \ldots, \mathrm{N})$. Le problème décrit dans cette section consiste à estimer les meilleures valeurs possible des paramètres a_j $(j = 1, \ldots, \mathrm{M})$ qui découlent de ces observations.

Le principe de base que nous allons suivre consiste à *maximiser la vraisemblance*, c'est-à-dire à trouver les valeurs a_j les plus probables. Pour cela, nous devons définir une probabilité d'observer les y_i, étant données des valeurs de a, c'est-à-dire étant donné le modèle. Nous allons supposer que cette probabilité suit une loi gaussienne, c'est-à-dire qu'elle prend la forme suivante :

$$\mathrm{P}(\text{observations}|a) = \prod_{i=1}^{\mathrm{N}} \exp\left[-\frac{1}{2}\left(\frac{y_i - y(x_i|a)}{\sigma_i}\right)^2\right] \Delta y \tag{17.34}$$

où $\sigma_i = \sigma(x_i)$ est un écart-type qui dépend de x et qui est lié au processus de mesure lui-même; Δy est un intervalle conventionnel de y nécessaire ici parce que nous avons affaire à une densité de probabilité (cet intervalle doit être petit en comparaison de σ_i). L'emploi d'une loi gaussienne se justifie en supposant que le processus de mesure est perturbé par une suite d'événements non

corrélés et de variances identiques, dont la résultante, en vertu de la loi des grands nombres, suit une distribution gaussienne.

La probabilité (17.34) est une *probabilité conditionnelle*, c'est-à-dire qu'elle exprime la probabilité d'un événement A (les observations) étant donné la certitude sur l'événement B (le modèle). Or ce qui nous intéresse ici est l'inverse : quelle est la probabilité du modèle B étant donnée une certitude sur les observations A : P(B|A). Ces deux probabilités sont reliées par le théorème de Bayes :

$$\mathrm{P(A\cap B)=P(A|B)P(B)=P(B|A)P(A)}\quad\text{ou encore}\quad\mathrm{P(A|B)}=\frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}} \tag{17.35}$$

Autrement dit : la probabilité que A et B se produisent est la probabilité de A étant donné B, fois la probabilité de B. Ceci est manifestement identique à la probabilité de B étant donné A, fois la probabilité de A. Ce théorème est donc extrêmement simple, en dépit des apparences.

Donc, appliqué à notre problème, ce théorème stipule que :

$$\mathrm{P}(a|\text{observations})=\mathrm{P}(\text{observations}|a)\mathrm{P}(a)/\mathrm{P}(\text{observations}) \tag{17.36}$$

Nous allons supposer ici que la probabilité absolue du modèle est indépendante de a, c'est-à-dire que toutes les valeurs de a sont a priori équiprobables, en l'absence d'observation. Donc seul le premier facteur du membre de droite dépend de a. Nous devons chercher les valeurs de a_j qui maximisent cette probabilité, ou qui minimisent l'opposé de son logarithme, ce qui veut dire minimiser l'expression suivante :

$$\boxed{\chi^2=\sum_i\left(\frac{y_i-y(x_i|a)}{\sigma_i}\right)^2} \tag{17.37}$$

En conclusion : la méthode du maximum de vraisemblance, alliée à quelques hypothèses générales, nous dicte la manière d'inférer les paramètres a_j : il faut minimiser la somme des écarts au carré entre les observations et le modèle, somme pondérée par l'erreur σ. Ceci est accompli en annulant les dérivées premières par rapport à a_j, c'est-à-dire en résolvant les M équations suivantes :

$$\sum_i\frac{y_i-y(x_i|a)}{\sigma_i^2}\frac{\partial\, y(x_i|a)}{\partial\, a_j}=0\qquad(j=1,\ldots,\mathrm{M}) \tag{17.38}$$

D.2 Combinaisons linéaires de fonctions de lissage

Les équations (17.38) sont en général difficiles à résoudre, car la fonction (17.37) est en général non linéaire en fonction des paramètres a_j. Par contre, le cas linéaire est assez fréquent et simple : le modèle est une superposition de fonctions $\mathrm{Y}_j(x)$, dont les coefficients sont les paramètres a_j :

$$y(x)=\sum_j a_j\mathrm{Y}_j(x) \tag{17.39}$$

Les équations (17.38) deviennent alors

$$\sum_i\frac{1}{\sigma_i^2}\left(y_i-\sum_k a_k\mathrm{Y}_k(x_i)\right)\mathrm{Y}_j(x_i)=0\qquad(j=1,\ldots,\mathrm{M}) \tag{17.40}$$

ce qui peut également s'écrire sous la forme concise suivante :

$$\sum_k \alpha_{jk} a_k = \beta_j \quad \text{où} \quad \alpha_{jk} = \sum_i \frac{1}{\sigma_i^2} \mathrm{Y}_j(x_i) \mathrm{Y}_k(x_i) \quad \text{et} \quad \beta_j = \sum_i \frac{y_i}{\sigma_i^2} \mathrm{Y}_j(x_i) \tag{17.41}$$

L'équation de gauche est un système linéaire qui se résout par les méthodes matricielles usuelles. On l'appelle le *système des équations normales* pour le problème du lissage de la fonction $y(x|a)$.[2]

Si on définit la matrice N × M

$$\mathrm{A}_{ij} = \frac{\mathrm{Y}_j(x_i)}{\sigma_i} \tag{17.42}$$

et le vecteur colonne à N composantes $b_i = y_i/\sigma_i$, alors les quantités α et β s'expriment comme suit :

$$\alpha = \mathrm{A}^{\mathrm{T}}\mathrm{A} \qquad \beta = \mathrm{A}^{\mathrm{T}} b \tag{17.43}$$

La matrice inverse $\mathrm{C} = \alpha^{-1}$ est appelée *matrice de covariance* et donne accès aux incertitudes sur les valeurs des paramètres inférés a_j. En effet, ces paramètres sont

$$a_j = \sum_k \mathrm{C}_{jk} \beta_k = \sum_k \sum_i \mathrm{C}_{jk} \frac{y_i \mathrm{Y}_k(x_i)}{\sigma_i^2} \tag{17.44}$$

or la variance $\sigma^2(a_j)$ se trouve par la propagation normale des erreurs de la variable y_i vers la variable a_j (notons que la matrice α est indépendante des y_i) :

$$\begin{aligned}
\sigma^2(a_j) &= \sum_i \sigma_i^2 \left(\frac{\partial a_j}{\partial y_i} \right)^2 \\
&= \sum_i \sigma_i^2 \left(\sum_k \frac{\mathrm{C}_{jk} \mathrm{Y}_k(x_i)}{\sigma_i^2} \right)^2 \\
&= \sum_{l,k} \sum_i \frac{1}{\sigma_i^2} \mathrm{C}_{jk} \mathrm{C}_{jl} \mathrm{Y}_k(x_i) \mathrm{Y}_l(x_i) \\
&= \sum_{l,k} \mathrm{C}_{jk} \mathrm{C}_{jl} \alpha_{kl} \\
&= \sum_l \delta_{jl} \mathrm{C}_{jl} = \mathrm{C}_{jj}
\end{aligned} \tag{17.45}$$

Nous avons utilisé la relation $\alpha^{-1} = \mathrm{C}$ dans la dernière équation. En somme, l'élément diagonal n$^{\underline{\mathrm{o}}}$ j de la matrice de covariance est la variance associée au paramètre a_j, suite à la propagation des erreurs σ_j.

Qu'arrive-t-il si on ne connaît pas les erreurs σ_i ? Dans ce cas on peut formellement prétendre que toutes les erreurs σ_i sont égales à une constante σ. Les valeurs des paramètres tirées de l'équation (17.44) seront alors manifestement indépendantes de σ : la matrice α comporte un facteur $1/\sigma^2$ et la matrice C comporte donc le facteur inverse, qui annule le $1/\sigma^2$ apparaissant dans l'équation.

2. La solution du système linéaire peut être délicate, cependant, car le problème est souvent proche d'un système singulier et une méthode du genre SVD (décomposition en valeurs singulières) est recommandée. Voir *Numerical Recipes* à cet effet.

D.3 Lissages non linéaires

Si la fonction à lisser comporte un ou plusieurs paramètres qui apparaissent de manière non linéaire, le problème est plus difficile, et doit être traité par une méthode de minimisation du χ^2 comme celles décrites plus haut, par exemple la méthode de Newton-Raphson. Cependant il est fréquent dans ce contexte d'utiliser la méthode de *Levenberg-Marquardt*, que nous expliquons sommairement dans ce qui suit.

Cette méthode requiert la connaissance des premières et deuxièmes dérivées de la fonction à minimiser (le χ^2 de l'éq. (17.37)). A un point a_0 proche du minimum de la fonction χ^2, on peut utiliser l'approximation quadratique et écrire

$$\chi^2(a) = \chi^2(a_0) + \nabla\chi^2(a_0)\delta a + \frac{1}{2}\delta a^{\mathrm{T}} \mathrm{D}\delta a \qquad \delta a \coloneqq a - a_0 \tag{17.46}$$

où D est la matrice hessienne de χ^2 (la matrice des deuxièmes dérivées, ou hessien), évaluée au point a_0 :

$$\mathrm{D}_{ij} = \left.\frac{\partial^2\chi^2}{\partial a_i \partial a_j}\right|_{a_0} \tag{17.47}$$

Dans cette approximation, le gradient au point a est donné par

$$\nabla\chi^2(a) = \nabla\chi^2(a_0) + \mathrm{D}\delta a \tag{17.48}$$

et il s'annule si

$$\delta a = -\mathrm{D}^{-1}\nabla\chi^2(a_0) \quad \text{ou encore} \quad a = a_0 - \mathrm{D}^{-1}\nabla\chi^2(a_0) \tag{17.49}$$

Cette relation constitue donc une mise à jour adéquate de la valeur de a, si elle n'est pas trop éloignée de la solution. Si elle est trop éloignée, alors il est plus efficace de faire le pas suivant, le long du gradient de la fonction :

$$a = a_0 - \gamma\nabla\chi^2(a_0) \tag{17.50}$$

où la constante γ est suffisamment petite. Ceci équivaut à adopter un hessien diagonal. La méthode de Levenberg-Marquardt combine ces deux approches, en utilisant la deuxième initialement, et en se rapprochant de la première au fur et à mesure qu'on approche de la solution : on définit la matrice

$$\alpha_{ij} = \frac{1}{2}\frac{\partial^2\chi^2}{\partial a_i \partial a_j} \quad \text{ainsi que} \quad \beta_k = -\frac{\partial\chi^2}{\partial a_k} \tag{17.51}$$

On définit ensuite un matrice modifiée

$$\alpha'_{ij} = \alpha_{ij} + \lambda\delta_{ij} \tag{17.52}$$

où λ est une constante qui est initialement petite ($\lambda \sim 0.001$). La méthode consiste alors à résoudre le système linéaire

$$\alpha'\delta a = \beta \implies \delta a = (\alpha')^{-1}\beta \tag{17.53}$$

à répétition, en suivant la prescription suivante :

1. Choisir une valeur initiale a des paramètres et calculer $\chi^2(a)$.
2. Choisir une petite valeur de λ (ex. 0.001) et calculer α'.
3. Calculer δa selon l'éq. (17.53). Sortir de la boucle si δa est suffisamment petit.

4. Si $\chi^2(a+\delta a) \geq \chi^2(a)$, augmenter λ d'un facteur important (ex. 10) et retourner à l'étape 3. Sinon, diminuer λ par un facteur important (ex. 10), mettre à jour $a \rightarrow a+\delta a$ et retourner à l'étape 3.

Les éléments diagonaux de la matrice de covariance $C = \alpha^{-1}$ sont les variances des paramètres obtenus, compte tenu des erreurs σ_i.

Boîte à outils

La méthode de Levenberg-Marquardt est utilisée par la plupart des programmes de lissage, incluant par la commande `fit` de gnuplot ainsi que `scipy.optimize.curve_fit`. Cette dernière fonction peut aussi utiliser d'autres méthodes (selon l'argument optionnel `method`) qui peuvent être invoqués si la méthode par défaut (`method='lm'` pour «Levenberg-Marquardt») n'arrive pas à converger.

E La méthode du recuit simulé

Une grande partie des problèmes d'optimisation vise à minimiser une fonction $E(x)$ définie sur un espace Ω de configurations discret, sur lequel la notion de continuité n'existe pas. Le problème le plus connu de ce genre est celui du *commis voyageur* : un voyageur de commerce doit visiter N villes dont les positions sont connues, tout en minimisant le chemin parcouru au total. On peut supposer pour simplifier l'argument qu'il peut parcourir la distance entre chaque paire de villes selon une ligne droite, mais cela n'est pas essentiel à la formulation du problème. Le voyageur doit donc choisir un itinéraire, qui prend la forme d'une permutation p_i de la liste des villes ($i = 1, \ldots, N$), qu'il devra parcourir dans cet ordre, en commençant et en terminant par la même ville (la trajectoire est fermée). La figure 17.4 illustre un exemple de chemin minimal reliant $N = 36$ points.

Ce problème ne peut pas être résolu par les méthodes classiques d'optimisation, car la variable sur laquelle nous devons minimiser est une permutation p, et non une variable continue. D'un autre côté, la simple énumération de toutes les possibilités est hors de question, car le nombre d'itinéraires possibles est N!, un nombre qui défie l'imagination même pour une valeur modérée de N, comme 36. [3]

La méthode de choix pour résoudre ce type de problème est l'algorithme du *recuit simulé* (angl. *simulated annealing*). [4] L'idée générale est de considérer la fonction E comme une énergie qu'on veut minimiser, d'explorer différentes configurations par l'algorithme de Metropolis, et de diminuer progressivement la température T jusqu'à un minimum proche de zéro. Autrement dit, on procède à une marche aléatoire dans l'espace des configurations Ω. Cette marche favorise les configurations de basse «énergie» E, mais permet tout de même de passer pendant un certain temps à des configurations d'énergie plus grande, quand la température est suffisamment élevée. Cet aspect est crucial,

3. $36! = 371\,993\,326\,789\,901\,217\,467\,999\,448\,150\,835\,200\,000\,000 \sim 3.10^{41}$.

4. En métallurgie, le *recuit* est un procédé par lequel un alliage est porté à haute température et ensuite refroidi lentement, ce qui permet aux défauts cristallins de se propager et d'être évacués, par opposition à la *trempe*, qui est un refroidissement soudain du matériau, qui gèle sur place les différents défauts cristallins et augmente la dureté de l'alliage.

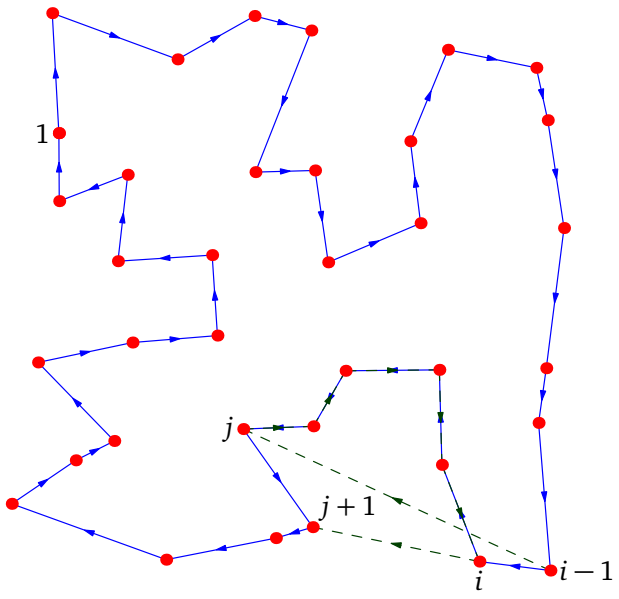


FIGURE 17.4
Le problème du commis voyageur : comment relier N points (ou «villes») par un chemin qui passe au moins une fois par chaque ville et minimise la distance totale parcourue. La portion pointillée illustre un changement local qu'il est possible d'apporter à l'itinéraire dans le recherche d'un minimum de la distance totale.

car il permet de surmonter les barrières de potentiel qui entourent souvent les minimums locaux (l'adjectif «local» doit être correctement interprété dans le contexte d'un espace Ω où la notion de distance n'est pas définie de manière évidente).

La méthode du recuit simulé requiert donc les ingrédients suivants :

1. Une définition claire de ce que constitue une configuration. Dans le cas du problème du commis voyageur, une configuration est une permutation p de la liste des villes à visiter.
2. Une fonction de type «énergie» $\mathrm{E}(p)$, qu'on désire minimiser. Dans le problème du voyageur, c'est la longueur de l'itinéraire :

$$\mathrm{E}(p) = \sum_{i=0}^{\mathrm{N}-1} d(p_i, p_{i+1}) \tag{17.54}$$

   où $d(j,k)$ est la distance entre les villes j et k, et p_i est l'indice n$^{\underline{o}}$ i de la permutation p. Il est sous-entendu que les indices sont traités de manière périodique, c'est-à-dire modulo N (l'indice $\mathrm{N}+i$ étant considéré synonyme de i).
3. Une procédure de changement local, l'équivalent du renversement local du spin dans le modèle d'Ising. Dans le problème du voyageur, une procédure possible est de sélectionner un indice i au hasard, ainsi qu'un deuxième indice j également au hasard, mais suivant une distribution exponentielle en fonction de la séparation $|j-i|$, de sorte que les paires (i,j) d'indices proches sont plus probables que les paires éloignées. Ensuite, on sectionne le chemin entre les indices $i-1$ et i, et entre les indices j et $j+1$ et on recolle la portion du chemin comprise entre i et j dans l'autre sens, c'est-à-dire de $i-1$ à j, ensuite $j-1$, etc. jusqu'à i et ensuite $j+1$. Cette procédure est illustrée par les traits pointillés sur la figure 17.4.

4. Un mode de refroidissement, c'est-à-dire une séquence de températures décroissantes, entrecoupées d'une série de changements locaux à chaque température. Le mode choisi dans le code décrit ci-dessous est caractérisé par une température initiale T_1, une température finale T_2, un facteur $1-\epsilon$ par lequel la température est multipliée à chaque étape, et un nombre m de mises à jour de la configuration effectuées à chaque valeur de T.

Au total, un code de recuit simulé ressemble beaucoup, mais en plus simple, à un code de simulation de physique statistique dans l'ensemble canonique. Il n'y a pas lieu, cependant, de procéder à une analyse d'erreur, car c'est l'état fondamental du système qui nous intéresse et non une moyenne statistique. Ceci dit, il n'y a aucune garantie que l'algorithme va converger vers le minimum absolu et non vers un minimum local. Il est cependant très peu probable que ce minimum local soit mauvais, au sens que son «énergie» E soit élevée. Donc, même si le minimum absolu n'est pas garanti, l'algorithme est tout de même extrêmement utile, car le résultat qu'il produit est toujours intéressant d'un point de vue pratique. Dans tous les cas, l'algorithme est rapide, mais demande une certaine exploration des paramètres afin d'optimiser sa convergence vers un minimum de qualité ; plusieurs simulations répétées (avec des configurations initiales aléatoires différentes) sont à conseiller, afin de valider le point d'arrivée.

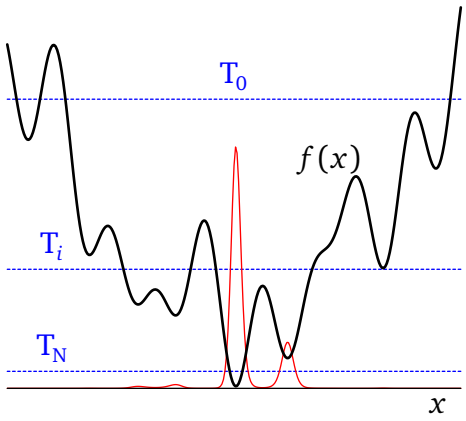


FIGURE 17.5
Application de la méthode du recuit simulé à la minimisation d'une fonction à une variable $f(x)$ possédant plusieurs minimums locaux. En rouge : probabilité $e^{-f(x)/T}$ d'une configuration x à la température finale T_N.

Application à une fonction ordinaire

Rien n'empêche d'utiliser la méthode du recuit simulé lors de la minimisation d'une fonction définie sur les réels. Dans ce cas, la mise à jour de la configuration consiste simplement en un déplacement $\delta\mathbf{x}$ dans une direction aléatoire dont la grandeur est, par exemple, distribuée uniformément dans un certain intervalle. Le mérite de la méthode dans ce cas est sa capacité à trouver souvent le minimum global de la fonction et non un minimum local. Voir la figure 17.5 pour un exemple unidimensionnel : la température initiale T_0 est suffisamment élevée pour permettre une exploration de l'espace dans un domaine large où plusieurs minimums relatifs existent. À mesure que la température diminue, la distribution de probabilité régissant l'algorithme de Metropolis devient de plus en plus concentrée autour du minimum global, de sorte que ce minimum est celui qui est atteint le plus souvent à la fin de la simulation.

CHAPITRE 18

DYNAMIQUE DES FLUIDES

L'étude du mouvement des fluides est l'une des applications les plus courantes du calcul scientifique. Elle est particulièrement répandue en génie mécanique, dans la conception de véhicules (automobile, aéronautique) et de réacteurs. En physique, elle constitue le problème de base de la prévision météorologique et climatique, et de la dynamique stellaire. Bref, il s'agit d'un problème extrêmement important à la fois en sciences fondamentales et, surtout, en sciences appliquées.

Le problème du mouvement des fluides va bien au-delà de l'étude de l'écoulement d'un fluide unique dans une géométrie statique. Dans les problèmes concrets, le fluide comporte plusieurs composantes, c'est-à-dire plusieurs substances formant un mélange ; ces substances peuvent se transformer l'une dans l'autre par réactions (chimiques, nucléaires, etc.). Il peut s'agir de phases différentes (liquide et gaz, ou même liquide et solide). La géométrie dans laquelle le ou les fluides s'écoulent peut aussi changer dans le temps, les parois solides peuvent être élastiques (par ex. en aéronautique) ou impliquer une croissance (ex. propagation d'une frontière de phase liquide-solide). Bref, la richesse et la complexité des problèmes impliquant l'écoulement des fluides peut être considérable, et nous ne ferons qu'effleurer le sujet dans ce chapitre, en nous concentrant sur une méthode de calcul : la méthode de Boltzmann sur réseau.

A Équations fondamentales

A.1 Équations d'Euler et de Navier-Stokes

Pendant très longtemps, la mécanique des fluides s'est réduite à une tentative de résolution des équations fondamentales du mouvement des fluides, soit l'équation d'Euler ou, pour les fluides visqueux, l'équation de Navier-Stokes. Ces équations gouvernent l'évolution dans le temps d'un champ de vitesse $\mathbf{u}(\mathbf{r}, t)$, qui décrit la vitesse au point $\mathbf{r}$ d'un fluide à une composante. La densité du fluide est décrite par un champ scalaire $\rho(\mathbf{r})$, et le produit $\rho\mathbf{u}$ définit une densité de courant qui obéit à l'équation de continuité, reflétant la conservation de la masse :

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0 \tag{18.1}$$

L'équation d'Euler gouverne le mouvement d'un fluide non visqueux et prend la forme suivante :

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla \mathrm{P} + \rho\mathbf{f} \tag{18.2}$$

où $\mathrm{P}(\mathbf{r}, t)$ est la pression au point $\mathbf{r}$ et au temps t et $\mathbf{f}$ est une force externe (par unité de masse) agissant au point $\mathbf{r}$. Si cette force dérive d'un potentiel, comme la force de gravité, on peut l'exprimer comme le gradient d'une fonction : $\mathbf{f} = -\nabla \mathrm{V}$, où V serait par exemple le potentiel gravitationnel. En divisant par ρ, on obtient la forme alternative

$$\boxed{\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla \mathrm{P} + \mathbf{f}} \tag{18.3}$$

L'équation (18.2) par elle-même est incomplète : on doit lui ajouter l'équation de continuité (18.1), qui représente la contrainte de conservation de la masse. On doit également disposer d'une équation d'état qui détermine la pression $\mathrm{P}(\rho, \mathrm{T})$ en fonction de la densité et d'un paramètre externe comme la température (en supposant que celle-ci soit uniforme et constante). Si la température n'est pas uniforme, alors il faut ajouter un champ de température $\mathrm{T}(\mathbf{r}, t)$ qui sera gouverné par la conservation de l'énergie, sous la forme d'une équation de continuité pour l'énergie interne, avec un courant de chaleur associé.

Si le fluide est visqueux, alors une force supplémentaire apparaît, par laquelle la vitesse du fluide a tendance à s'uniformiser. Cette force prend la forme suivante :

$$\frac{1}{\rho}\left[\eta\nabla^2\mathbf{u} + \left(\zeta + \frac{\eta}{3}\right)\nabla\nabla\cdot\mathbf{u}\right] \tag{18.4}$$

où η est le *coefficient de viscosité dynamique* et ζ la *seconde viscosité*. Quand cette force est prise en compte, on trouve alors *l'équation de Navier-Stokes* :

$$\boxed{\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla \mathrm{P} + \eta\nabla^2\mathbf{u} + \left(\zeta + \frac{\eta}{3}\right)\nabla\nabla\cdot\mathbf{u} + \rho\mathbf{f}} \tag{18.5}$$

Si le fluide est *incompressible*, la densité ρ est une constante et donc, par l'équation de continuité, $\nabla\cdot\mathbf{u} = 0$. Dans ce cas, l'équation de Navier-Stokes se réduit à

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla \mathrm{P} + \eta\nabla^2\mathbf{u} + \rho\mathbf{f} \tag{18.6}$$

ou encore, en divisant par ρ,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla \mathrm{P} + \nu\nabla^2\mathbf{u} + \mathbf{f} \tag{18.7}$$

où $\nu = \eta/\rho$ est appelée la *viscosité cinématique*.

A.2 Démonstration de l'équation d'Euler

L'équation (18.2) peut être démontrée de la manière suivante. Le fluide porte une certaine quantité de mouvement. La i^{e} composante de la quantité de mouvement ($i = 1, 2, 3$) est portée par une densité ρu_i et une densité de courant $\rho\mathbf{u}u_i$. En effet, ρu_i est la quantité de mouvement par unité de volume dans la direction i, et naturellement la densité de courant correspondante doit être obtenue simplement en multipliant par la vitesse du fluide. La quantité de mouvement du fluide, cependant,

n'est pas conservée : elle est modifiée par une certaine force par unité de volume. L'équation de continuité dans ce cas doit donc comporter un membre de droite, soit la force par unité de volume, de la forme donnée en (18.2) :

$$\frac{\partial \rho u_i}{\partial t} + \nabla\cdot(\rho \mathbf{u} u_i) = -\frac{\partial \mathrm{P}}{\partial x^i} + \rho f_i \tag{18.8}$$

En appliquant au membre de gauche les relations sur la dérivée d'un produit, on trouve

$$\rho \frac{\partial u_i}{\partial t} + u_i \frac{\partial \rho}{\partial t} + u_i \nabla\cdot(\rho \mathbf{u}) + (\rho \mathbf{u}\cdot\nabla) u_i \tag{18.9}$$

Or, la somme des deuxième et troisième termes de cette expression s'annule en vertu de l'équation de continuité (18.1). Il reste donc

$$\rho \frac{\partial u_i}{\partial t} + \rho(\mathbf{u}\cdot\nabla) u_i = -\frac{\partial \mathrm{P}}{\partial x^i} + \rho f_i \tag{18.10}$$

ce qui n'est rien d'autre que l'éq. (18.2), composante par composante.

Une façon différente d'arriver au même résultat est la suivante : L'opérateur

$$\frac{\partial}{\partial t} + \mathbf{u}\cdot\nabla \tag{18.11}$$

est une *dérivée en co-mouvement* ou *dérivée convective*, qui calcule la dérivée par rapport au temps d'une quantité liée à un élément de masse qui se déplace à une vitesse $\mathbf{u}$. Si on suit un élément de fluide pendant un court instant, la dérivée temporelle d'une quantité relative à cet élément doit tenir compte du déplacement de cet élément pendant cet instant. La dérivée totale par rapport au temps est alors

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\partial}{\partial t} + \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\cdot\frac{\partial}{\partial \mathbf{r}} = \frac{\partial}{\partial t} + \mathbf{u}\cdot\nabla \tag{18.12}$$

Le membre de gauche de l'équation (18.2) représente donc la dérivée totale de la quantité de mouvement par unité de volume par rapport au temps. Le deuxième terme du membre de gauche de l'éq. (18.2) est appelé *terme de convection*.

Tenseur de la densité de flux d'impulsion

On peut récrire l'équation (18.8) ainsi (en l'absence de force externe) :

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial \Pi_{ik}}{\partial x_k} = 0 \tag{18.13}$$

où le tenseur Π_{ik} dit « de la densité de flux d'impulsion» est défini ainsi :

$$\Pi_{ik} = \mathrm{P}\delta_{ik} + \rho u_i u_k \tag{18.14}$$

Autrement dit, Π_{ik} est la k^{e} composante du flux de la i^{e} composante de la quantité de mouvement, incluant non seulement le flux provenant du mouvement du fluide lui-même, mais aussi des forces agissant sur le fluide localement. Le tenseur Π_{ik} est symétrique.

A.3 Démonstration de l'équation de Navier-Stokes

Nous allons maintenant ajouter au tenseur Π_{ik} un terme qui tient compte de la force de viscosité. Cette dernière tend à diminuer les variations de vitesses et donc s'annule lorsque la vitesse du fluide est homogène. Le terme correspondant du tenseur Π_{ik} doit donc contenir la dérivée de la vitesse, en plus d'être symétrique. [1] Les possibilités les plus simples, linéaires en dérivées de la vitesse, sont

$$\frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_k} \quad \text{et} \quad \delta_{ik}\frac{\partial u_j}{\partial x_j} \tag{18.15}$$

Un fluide pour lequel ces deux possibilités suffisent à décrire la viscosité est qualifié de *newtonien*. On écrira donc

$$\Pi_{ik} = \mathrm{P}\delta_{ik} + \rho u_i u_k - \eta\left\{\frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_k} - \frac{2}{3}\delta_{ik}\frac{\partial u_j}{\partial x_j}\right\} - \zeta\delta_{ik}\frac{\partial u_j}{\partial x_j} \tag{18.16}$$

où on a distingué une contribution sans trace avec coefficient $-\eta$ et une trace pure avec coefficient $-\zeta$. Les contributions additionnelles à la force par unité de volume $-\partial_k\Pi_{ik}$ provenant des deux nouveaux termes sont

$$\begin{aligned}
&\eta\frac{\partial}{\partial x_k}\left\{\frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_k} - \frac{2}{3}\delta_{ik}\frac{\partial u_j}{\partial x_j}\right\} + \zeta\delta_{ik}\frac{\partial}{\partial x_k}\frac{\partial u_j}{\partial x_j} \\
&\quad = \eta\left\{\frac{\partial}{\partial x_i}\nabla\cdot\mathbf{u} + \nabla^2 u_i - \frac{2}{3}\frac{\partial}{\partial x_i}\nabla\cdot\mathbf{u}\right\} + \zeta\frac{\partial}{\partial x_i}\nabla\cdot\mathbf{u} \\
&\quad = \eta\nabla^2 u_i + (\zeta + \tfrac{1}{3}\eta)\frac{\partial}{\partial x_i}\nabla\cdot\mathbf{u}
\end{aligned} \tag{18.17}$$

En notation vectorielle, ces deux termes sont

$$\eta\nabla^2\mathbf{u} + (\zeta + \tfrac{1}{3}\eta)\nabla\nabla\cdot\mathbf{u} \tag{18.18}$$

On obtient donc les deux termes correspondants de l'éq. (18.5). Notez que les signes des deux nouveaux termes sont choisis de telle manière que si $\eta > 0$ et $\zeta > 0$, alors les inhomogénéités de la vitesse ont tendance à diminuer avec le temps.

A.4 Cas particulier d'un fluide incompressible

Le problème de l'écoulement se simplifie quelque peu dans le cas d'un fluide incompressible, c'est-à-dire dont la densité est constante. L'équation de continuité impose alors que l'écoulement est sans divergence : $\nabla\cdot\mathbf{u} = 0$. Nous sommes alors dispensés de résoudre une équation pour la variation de ρ, mais la pression P demeure variable : au fond, un fluide incompressible est la limite d'un fluide dont la compressibilité est extrêmement faible, dans lequel un très faible changement de densité mène à de grandes différences de pression. La pression P doit alors être déterminée dynamiquement, de manière à ce que l'écoulement respecte la contrainte $\nabla\cdot\mathbf{u} = 0$. La technique de «correction de pression», que nous n'expliquerons pas ici, peut accomplir ceci à chaque étape de l'évolution temporelle.

1. Une partie antisymétrique à ce tenseur aurait comme conséquence qu'un mouvement de rotation rigide d'un fluide, spécifiquement $\mathbf{u} = \boldsymbol{\omega}\wedge\mathbf{r}$, entrainerait une force de viscosité, ce qui n'a pas de sens.

A.5 Fluide incompressible en deux dimensions

Dans le cas d'un écoulement bidimensionnel (ou effectivement bidimensionnel), la condition d'incompressibilité $\nabla\cdot\mathbf{u}=0$ s'écrit

$$\frac{\partial u_x}{\partial x}=-\frac{\partial u_y}{\partial y} \tag{18.19}$$

et peut être généralement résolue par l'introduction de la «fonction de courant» ψ, telle que

$$u_x=\frac{\partial\psi}{\partial y} \qquad u_y=-\frac{\partial\psi}{\partial x} \tag{18.20}$$

En prenant le rotationnel de l'équation de Navier-Stokes, ce qui élimine le terme en gradient de pression et toute force conservative $\mathbf{f}$, on trouve l'équation suivante pour ψ :

$$\frac{\partial\chi}{\partial t}+\frac{\partial\psi}{\partial y}\frac{\partial\chi}{\partial x}-\frac{\partial\psi}{\partial x}\frac{\partial\chi}{\partial y}=\nu\nabla^2\chi \qquad \chi\coloneqq\nabla^2\psi \tag{18.21}$$

Il s'agit en fait d'un ensemble de deux équations aux dérivées partielles couplées (pour χ et ψ). Une simplification similaire existe si on suppose que l'écoulement est à symétrie cylindrique.

A.6 Écoulement irrotationnel

Tout champ de vecteur peut être décomposé en une partie longitudinale et une partie transverse (théorème de Helmholtz) :

$$\mathbf{u}=\nabla\phi+\nabla\wedge\mathbf{w} \tag{18.22}$$

où

$$\nabla\cdot\mathbf{u}=\nabla^2\phi \qquad \text{et} \qquad \nabla\wedge\mathbf{u}=-\nabla^2\mathbf{w} \qquad (\nabla\cdot\mathbf{w}=0) \tag{18.23}$$

La fonction ϕ est appelée «potentiel de l'écoulement». Des solutions particulières à l'équation de Navier-Stokes peuvent être obtenues en supposant que l'écoulement est irrotationnel ($\mathbf{w}=0$). En utilisant la relation exacte

$$\frac{1}{2}\nabla(\mathbf{u}^2)=\mathbf{u}\wedge(\nabla\wedge\mathbf{u})+(\mathbf{u}\cdot\nabla)\mathbf{u} \tag{18.24}$$

le terme de convection peut alors être remplacé par un gradient et on obtient

$$\frac{\partial\mathbf{u}}{\partial t}+\frac{1}{2}\nabla(\mathbf{u}^2)=-\frac{1}{\rho}\nabla\mathrm{P}+\nu\nabla^2\mathbf{u}-\nabla\mathrm{V} \tag{18.25}$$

où on a supposé que la force externe par unité de masse était conservative ($\mathbf{f}=-\nabla\mathrm{V}$). Si en plus on néglige la viscosité et qu'on suppose un écoulement stationnaire (indépendant du temps) et incompressible (ρ constant), on trouve l'équation de Bernoulli :

$$\nabla\left(\frac{1}{2}\mathbf{u}^2+\frac{1}{\rho}\mathrm{P}+\mathrm{V}\right)=0\implies\frac{1}{2}\mathbf{u}^2+\mathrm{P}+\mathrm{V}=\mathrm{cte} \tag{18.26}$$

Pour un fluide incompressible en écoulement irrotationnel, la relation $\nabla\cdot\mathbf{u}=0$ devient simplement l'équation de Laplace $\nabla^2\phi=0$. En dimension deux, la relation entre ϕ et la fonction de courant ψ est alors

$$\frac{\partial\psi}{\partial y}=\frac{\partial\phi}{\partial x} \qquad \frac{\partial\psi}{\partial x}=-\frac{\partial\phi}{\partial y} \tag{18.27}$$

ce qui constitue les équations de Cauchy-Riemann pour les parties réelle et imaginaire d'une fonction analytique d'une variable complexe $z=x+iy$. Toute fonction analytique $f(z)$ nous donne donc accès à un écoulement irrotationnel d'un fluide incompressible.

B Équation de Boltzmann

L'équation de Navier-Stokes est valable dans la limite macroscopique, ou *hydrodynamique*, dans laquelle les longueurs caractéristiques du problème (les dimensions des conduits, par exemple) sont très grandes par rapport au libre parcours moyen des molécules qui forment le fluide, afin que la notion d'élément de fluide ait un sens physique.

L'équation de Navier-Stokes peut être dérivée d'une équation plus fondamentale, c'est-à-dire plus microscopique : l'équation de Boltzmann. Cette dernière régit l'évolution dans le temps d'une distribution de probabilité de positions et de vitesses dans l'espace des phases et est l'équation fondamentale de la mécanique statistique hors équilibre.

Considérons à cet effet un fluide comportant un très grand nombre N de particules. Si nous avions la prétention d'étudier la dynamique classique exacte de ce système complexe, nous devrions considérer l'espace des phases de ce système, qui comporte 6N dimensions, pour les 3N positions et 3N impulsions du système. L'état classique du système serait alors défini par un point dans cet espace et l'évolution temporelle du système serait une trajectoire suivie par ce point dans l'espace des phases. Comme il est irréaliste de procéder ainsi, nous allons plutôt raisonner sur la base de l'espace des phases pour une seule molécule, en supposant que les N molécules du fluide sont distribuées dans cet espace des phases. Une telle distribution est définie par une fonction $f(\mathbf{r},\mathbf{p},t)$ de la position et de l'impulsion ; la probabilité de trouver une molécule dans un élément de volume $\mathrm{d}^3 r$ à la position $\mathbf{r}$, possédant une impulsion contenue dans un élément de volume $\mathrm{d}^3 p$ autour de $\mathbf{p}$, est alors proportionnelle à $f(\mathbf{r},\mathbf{p})\mathrm{d}^3 r\ \mathrm{d}^3 p$. Nous allons normaliser la distribution f par le nombre total N de particules :

$$\int \mathrm{d}^3 r\ \mathrm{d}^3 p\ f(\mathbf{r},\mathbf{p}) = \mathrm{N} \tag{18.28}$$

B.1 Moments de la distribution

Si on intègre la distribution f sur toutes les valeurs de l'impulsion $\mathbf{p}$ à une position donnée $\mathbf{r}$, on génère des *moments partiels* de la distribution de probabilité. Les trois premiers moments correspondent à la densité, la vitesse moyenne et l'énergie cinétique moyenne par unité de volume :

$$\begin{aligned}
\rho(\mathbf{r}) &= m \int \mathrm{d}^3 p\ f(\mathbf{r},\mathbf{p}) \\
\rho(\mathbf{r})\mathbf{u}(\mathbf{r}) &= \int \mathrm{d}^3 p\ \mathbf{p} f(\mathbf{r},\mathbf{p}) \\
\rho(\mathbf{r})\epsilon &= \frac{1}{2}\int \mathrm{d}^3 p\ (\mathbf{p}-m\mathbf{u})^2 f(\mathbf{r},\mathbf{p})
\end{aligned} \tag{18.29}$$

En effet, la première intégrale nous donne naturellement le nombre de particules par unité de volume, ou encore la densité massique ρ si on multiplie par la masse m de chaque particule. La deuxième expression est l'impulsion par unité de volume à une position $\mathbf{r}$, soit précisément la densité ρ multipliée par la vitesse $\mathbf{u}$ du fluide à cet endroit. Enfin, la dernière expression est proportionnelle à la variance de l'impulsion à la position $\mathbf{r}$, ce qui n'est autre que l'énergie cinétique par unité

de volume, telle que définie dans le référentiel qui se déplace à la vitesse moyenne des particules à cet endroit ; c'est donc l'énergie cinétique interne par unité de volume, ϵ étant l'énergie moyenne d'une particule de fluide dans ce référentiel.

B.2 Équation de Boltzmann

Si les N molécules sont indépendantes les unes des autres et ne répondent qu'à des forces externes, alors l'évolution dans le temps de la distribution f suit les règles de la mécanique de Hamilton à une particule : au temps $t + \mathrm{d}t$, la molécule à $(\mathbf{r}, \mathbf{p})$ s'est déplacée vers $(\mathbf{r} + \mathbf{p}\,\mathrm{d}t/m, \mathbf{p} + \mathbf{f}\,\mathrm{d}t)$, où $\mathbf{f}$ est une force externe conservative agissant sur chaque particule. Donc

$$f(\mathbf{r} + \mathbf{p}\,\mathrm{d}t/m, \mathbf{p} + \mathbf{f}\,\mathrm{d}t, t + \mathrm{d}t) = f(\mathbf{r}, \mathbf{p}, t) \tag{18.30}$$

ou encore, en développant,

$$\frac{\partial f}{\partial t} + \frac{1}{m}\mathbf{p} \cdot \frac{\partial f}{\partial \mathbf{r}} + \mathbf{f} \cdot \frac{\partial f}{\partial \mathbf{p}} = 0 \tag{18.31}$$

Cette équation décrit des particules indépendantes qui se déplacent sans interagir. On peut se figurer la distribution f comme un nuage de probabilité qui se déplace dans l'espace des phases. Selon le théorème de Liouville,[2] une portion donnée de l'espace des phases évolue dans le temps en se déformant, mais en conservant son volume. Le nuage de probabilité associé à f s'écoule donc dans l'espace des phases en se déformant, mais de manière incompressible.

Ceci n'est plus vrai si on tient compte des collisions entre les particules élémentaires du fluide. À cette fin, on modifie la relation (18.31) comme suit :

$$\frac{\partial f}{\partial t} + \frac{1}{m}\mathbf{p} \cdot \frac{\partial f}{\partial \mathbf{r}} + \mathbf{f} \cdot \frac{\partial f}{\partial \mathbf{p}} = \left.\frac{\delta f}{\delta t}\right|_{\text{coll.}} \tag{18.32}$$

où le membre de droite représente tout changement dans f (à un point donné de l'espace des phases) causé par les interactions entre les particules (ou molécules).

B.3 Approximation des collisions moléculaires

Le terme de collision est souvent représenté dans *l'approximation des collisions moléculaires*, qui suppose que les molécules collisionnent par paires et que deux molécules incidentes d'impulsions $\mathbf{p}$ et $\mathbf{p}'$ se retrouvent après collision avec des impulsions $\mathbf{p}+\mathbf{q}$ et $\mathbf{p}'-\mathbf{q}$: $\mathbf{q}$ est le transfert d'impulsion entre les deux particules et la quantité de mouvement totale des deux particules est conservée lors de la collision. Cette collision se produit avec une densité de probabilité par unité de temps $g(\mathbf{p}-\mathbf{p}', \mathbf{q})$ qui ne dépend que de deux variables en raison de l'invariance galiléenne : l'une est la vitesse relative des deux particules, $(\mathbf{p}-\mathbf{p}')/m$, l'autre le paramètre d'impact de la collision, qui se traduit par un angle de diffusion θ, ou de manière équivalente par un transfert d'impulsion $\mathbf{q}$.

La variation de la fonction f à $\mathbf{p}$ due aux collisions avec des molécules d'impulsion $\mathbf{p}'$ est alors

$$\left.\frac{\delta f}{\delta t}\right|^{(1)}_{\text{coll.}} = -\int \mathrm{d}^3p'\,\mathrm{d}^3q\, g(\mathbf{p}-\mathbf{p}', \mathbf{q}) f(\mathbf{r}, \mathbf{p}, t) f(\mathbf{r}, \mathbf{p}', t) \tag{18.33}$$

2. Voir le cours de Mécanique II – PHQ310

Il s'agit bien sûr d'une diminution du nombre de particules d'impulsion $\mathbf{p}$ due aux collisions qui diffusent ces particules vers d'autres impulsions. Par contre, les collisions peuvent aussi augmenter f à $\mathbf{p}$ en raison des particules qui diffusent *vers* l'impulsion $\mathbf{p}$. Cette augmentation s'écrit naturellement comme

$$\left.\frac{\delta f}{\delta t}\right|_{\text{coll.}}^{(2)} = \int \mathrm{d}^3p'\,\mathrm{d}^3q\,g(\mathbf{p}-\mathbf{p}',\mathbf{q})f(\mathbf{r},\mathbf{p}+\mathbf{q},t)f(\mathbf{r},\mathbf{p}'-\mathbf{q},t) \tag{18.34}$$

qu'il faut lire comme suit : les particules d'impulsion $\mathbf{p}+\mathbf{q}$ diffusent sur les particules d'impulsion $\mathbf{p}'-\mathbf{q}$, ce qui résulte, après un transfert d'impulsion $\mathbf{q}$, vers les impulsions $\mathbf{p}$ et $\mathbf{p}'$. On intègre sur toutes les possibilités $\mathbf{p}'$ et $\mathbf{q}$ pour obtenir la variation ci-dessus. Au total, l'équation de Boltzmann dans l'approximation des collisions moléculaires (*Stosszahl Ansatz*) prend la forme suivante :

$$\frac{\partial f}{\partial t}+\frac{1}{m}\mathbf{p}\cdot\frac{\partial f}{\partial \mathbf{r}}+\mathbf{f}\cdot\frac{\partial f}{\partial \mathbf{p}} = \int \mathrm{d}^3p'\,\mathrm{d}^3q\,g(\mathbf{p}-\mathbf{p}',\mathbf{q})\Big[f(\mathbf{r},\mathbf{p}+\mathbf{q},t)f(\mathbf{r},\mathbf{p}'-\mathbf{q},t)-f(\mathbf{r},\mathbf{p},t)f(\mathbf{r},\mathbf{p}',t)\Big] \tag{18.35}$$

L'équation (18.35) est difficile à résoudre : c'est une équation intégro-différentielle, qui implique une intégrale double sur les impulsions à chaque évaluation de la dérivée temporelle de f. On peut montrer (mais nous ne le ferons pas ici, car c'est une proposition assez complexe) que l'équation de Navier-Stokes est une conséquence de l'équation (18.35).

Remarquons qu'en l'absence de forces externes, l'équation de Boltzmann sans le terme de collisions entraîne que les distributions partielles $f(\mathbf{r},\mathbf{p},t)$ associées à des valeurs différentes de $\mathbf{p}$ sont indépendantes, c'est-à-dire qu'elles ne s'influencent nullement en fonction du temps. Le terme de collision se trouve à coupler entre elles des valeurs différentes de $\mathbf{p}$.

B.4 Approximation du temps de relaxation

On s'attend naturellement à ce que la distribution f tende vers une distribution d'équilibre $f^{\text{eq.}}$ en raison des collisions. Cette convergence vers $f^{\text{eq.}}$ va s'effectuer en général avec un certain temps caractéristique τ si la distribution initiale n'est pas trop éloignée de $f^{\text{eq.}}$. Une façon courante de simplifier considérablement l'équation de Boltzmann est de remplacer le terme de collision par une simple relaxation vers la distribution d'équilibre :

$$\frac{\partial f}{\partial t}+\frac{1}{m}\mathbf{p}\cdot\frac{\partial f}{\partial \mathbf{r}}+\mathbf{f}\cdot\frac{\partial f}{\partial \mathbf{p}} = \frac{1}{\tau}\Big[f^{\text{eq.}}(\mathbf{r},\mathbf{p})-f(\mathbf{r},\mathbf{p},t)\Big] \tag{18.36}$$

La distribution $f^{\text{eq.}}$ se calcule, bien sûr, en mécanique statistique des systèmes à l'équilibre. Si la distribution décrit un fluide qui se déplace à une vitesse $\mathbf{u}(\mathbf{r})$, ce sont les déviations par rapport à cette vitesse qui seront l'objet d'une distribution de Maxwell-Boltzmann :

$$f^{\text{eq.}}(\mathbf{r},\mathbf{p}) = \frac{\rho}{m(2\pi m\mathrm{T})^{d/2}}\exp-\frac{[\mathbf{p}-m\mathbf{u}(\mathbf{r})]^2}{2m\mathrm{T}} \tag{18.37}$$

où d est la dimension de l'espace et T la température absolue (on pose $k_{\mathrm{B}}=1$). Notez que, dans cette dernière équation, la vitesse $\mathbf{u}$ dépend en général de la position. Cette distribution des impulsions à l'équilibre respecte les règles de somme décrites à l'éq. (18.29), sauf que dans ce cas la densité d'énergie cinétique ϵ est donnée par le théorème d'équipartition : $\epsilon = \mathrm{T}\,d/2$.

Remarques :

- ✦ Notons que l'approximation du temps de relaxation couple effectivement les différentes valeurs de $\mathbf{p}$: la distribution à l'équilibre $f^{\text{eq.}}$ dépend de la vitesse du fluide $\mathbf{u}$ à un point donné, et celle-ci à son tour se calcule par une intégrale sur les différentes impulsions.
- ✦ La distribution (18.37) est définie de manière auto-cohérente. Par cela on veut dire que les quantités $\rho(\mathbf{r})$ et $\mathbf{u}(\mathbf{r})$ sont en principe déterminés par la même distribution qu'elles veulent définir. En pratique, on ne se soucie pas de cela : on insère dans (18.37) les valeurs de $\rho(\mathbf{r})$ et $\mathbf{u}(\mathbf{r})$ provenant de $f(\mathbf{r},\mathbf{p},t)$ à un instant donné, en supposant que ces dernières ne sont pas trop différentes des valeurs à l'équilibre, et de toute manière l'éq. de Boltzmann tend à les conserver proches de leurs valeurs à l'équilibre.

C Méthode de Boltzmann sur réseau

C.1 Généralités

Depuis les années 1990, l'hydrodynamique numérique se tourne de plus en plus vers l'équation de Boltzmann, au lieu de s'attaquer à l'équation de Navier-Stokes. L'avantage de l'équation de Boltzmann est qu'elle permet plus facilement de considérer des géométries complexes (des milieux poreux, par exemple) et des fluides à plusieurs composantes, en tenant compte des réactions entre ces composantes : le terme de collision peut en effet convertir des espèces de molécules en d'autres espèces, c'est-à-dire décrire la cinétique des réactions chimiques. Le traitement numérique de l'équation de Boltzmann dans le but de décrire le mouvement des fluides constitue ce qu'on appelle la *méthode de Boltzmann sur réseau* (angl. *lattice Boltzmann method* ou LBM). Beaucoup de logiciels commerciaux ou ouverts utilisés en génie ou ailleurs sont maintenant basés sur cette méthode.

Cependant, l'équation (18.35) est encore trop complexe. La méthode LBM est basée plutôt sur la version simplifiée (18.36) de l'équation de Boltzmann. C'est ce qu'on appelle dans ce contexte le modèle de Bhatnagar-Gross-Krook (BGK).

La numérisation de l'équation (18.36) passe nécessairement par une discrétisation de l'espace des phases. En pratique, la partie spatiale a besoin d'une discrétisation fine, car souvent on s'intéresse au passage des fluides à travers des géométries compliquées. Par contre, la discrétisation de l'espace des vitesses (ou des impulsions) est extrêmement simplifiée : on considère généralement 9 valeurs de la vitesse en deux dimensions (schéma D2Q9), 15 ou 27 en dimension 3 (schémas D3Q15 et D3Q27). Il peut sembler surprenant à première vue qu'un aussi petit nombre de vitesses puisse bien représenter le mouvement des fluides, mais il faut garder à l'esprit qu'on décrit ici une distribution de probabilité f, et que la vitesse du fluide qui sera produite par cette approche est la moyenne pondérée des probabilités associées aux 9 (ou 15, ou 27) valeurs formant la grille des vitesses, et donc qu'elle pourra prendre un continuum de valeurs dans l'espace borné par cette grille.

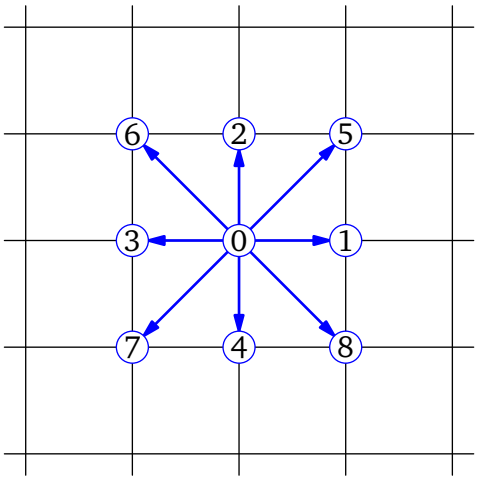


FIGURE 18.1
Les 9 vitesses possibles dans le schéma D2Q9 de la méthode de Boltzmann sur réseau, en relation avec les sites de la grille.

C.2 Le schéma D2Q9 : dimension 2, 9 vitesses

Voyons plus précisément comment se formule la méthode dans le schéma D2Q9 (voir fig. 18.1). Les sites forment un réseau carré de pas a. Définissons les 9 vecteurs sans unités $\mathbf{e}_i$ ($i = 0, \ldots, 8$) :

$$\mathbf{e}_0 = (0,0) \qquad \begin{aligned} \mathbf{e}_1 &= (1,0) & \mathbf{e}_5 &= (1,1) \\ \mathbf{e}_2 &= (0,1) & \mathbf{e}_6 &= (-1,1) \\ \mathbf{e}_3 &= (-1,0) & \mathbf{e}_7 &= (-1,-1) \\ \mathbf{e}_4 &= (0,-1) & \mathbf{e}_8 &= (1,-1) \end{aligned} \tag{18.38}$$

Les 9 déplacements possibles sont $a\mathbf{e}_i$ et sont illustrés par les flèches sur la figure. Les impulsions correspondantes sont définies de telle manière qu'en un intervalle de temps h (le pas temporel de la méthode), le déplacement $a\mathbf{e}_i$ soit la différence de deux positions appartenant au réseau : $\mathbf{p}_i = (ma/h)\mathbf{e}_i$.

La distribution $f(\mathbf{r}, \mathbf{v})$ devient dans ce cas un ensemble de 9 distributions $f_i(\mathbf{r})$, où les $\mathbf{r}$ appartiennent au réseau. La densité du fluide au point $\mathbf{r}$ est alors

$$\rho(\mathbf{r}) = m \sum_i f_i(\mathbf{r}) \tag{18.39}$$

alors que la vitesse $\mathbf{u}$ du fluide est telle que

$$\rho(\mathbf{r})\mathbf{u}(\mathbf{r}) = \frac{ma}{h} \sum_i f_i(\mathbf{r})\mathbf{e}_i \tag{18.40}$$

La version discrète de l'équation de Boltzmann (18.36) est alors

$$f_i(\mathbf{r} + \mathbf{e}_i a, t + h) = f_i(\mathbf{r}, t) - \frac{h}{\tau}\Big[f_i(\mathbf{r}, t) - f_i^{\text{eq.}}(\mathbf{r}, t)\Big] \tag{18.41}$$

Le premier terme du membre de droite effectue l'*écoulement* du fluide, alors que le deuxième terme représente l'effet des collisions. Nous avons supposé ici qu'aucune force externe **f** n'est à l'oeuvre. Notons qu'en l'absence de terme de collision, cette équation revient simplement à déplacer rigidement la distribution f_i sur une distance $a\mathbf{e}_i$ lors d'un pas temporel h : encore une fois il n'y a aucun mélange des vitesses et les différentes distributions sont indépendantes. Le terme de relaxation va modifier ce comportement en couplant les distributions partielles f_i via le calcul de la vitesse du fluide **u**.

Nous allons déterminer la forme de la distribution à l'équilibre $f_i^{\text{eq.}}$ en nous inspirant de la distribution de Maxwell-Boltzmann (18.37) et en imposant les règles de somme (18.29), qui deviennent dans ce contexte

$$\begin{aligned}\rho(\mathbf{r}) &= m\sum_i f_i^{\text{eq.}}(\mathbf{r})\\ \rho(\mathbf{r})\mathbf{u}(\mathbf{r}) &= \sum_i \mathbf{p}_i f_i^{\text{eq.}}\\ \rho(\mathbf{r})\mathrm{T} &= \frac{1}{2}\sum_i (\mathbf{p}_i - m\mathbf{u})^2 f_i^{\text{eq.}}\end{aligned} \tag{18.42}$$

En principe, ces règles de somme devraient s'appliquer pour toute valeur de **u**. Malheureusement, le nombre limité de vitesses (neuf), ne nous permet pas d'y arriver : nous ne pourrons respecter ces contraintes que pour les termes constants, linéaires et quadratiques en u. Autrement dit, il ne sera possible d'imposer ces contraintes que dans l'approximation des petites vitesses (petits nombres de Mach). En développant la distribution (18.37) à cet ordre en u, nous obtenons la forme suivante :

$$f_i^{\text{eq.}} = w_i\rho(\mathbf{r})\left\{1 + \frac{ma}{h\mathrm{T}}\mathbf{e}_i\cdot\mathbf{u} - \frac{m}{2\mathrm{T}}\mathbf{u}^2 + \frac{1}{2}\left(\frac{ma}{h\mathrm{T}}\right)^2(\mathbf{e}_i\cdot\mathbf{u})^2\right\} \tag{18.43}$$

où les constantes w_i sont considérées comme ajustables. En raison de la symétrie des 9 vecteurs $\mathbf{e}_i$, il est clair que trois seulement de ces constantes sont indépendantes : w_0, $w_1 = w_2 = w_3 = w_4$ et $w_5 = w_6 = w_7 = w_8$. L'imposition des trois règles de somme (18.42) à l'ordre u^2 fixe de manière unique ces constantes. Pour simplifier les choses, il est de coutume de fixer le pas temporel h de manière à ce que

$$\frac{ma^2}{h^2\mathrm{T}} = 3 \tag{18.44}$$

On démontre alors que

$$w_0 = \frac{4}{9} \qquad w_{1,2,3,4} = \frac{1}{9} \qquad w_{5,6,7,8} = \frac{1}{36} \tag{18.45}$$

On montre que ce modèle spécifique nous ramène, dans la limite continue, à l'équation de Navier-Stokes pour un fluide incompressible :

$$\nabla\cdot\mathbf{u} = 0 \qquad \rho\frac{\partial\mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla\mathrm{P} + \rho\,\nu\nabla^2\mathbf{u} \tag{18.46}$$

où la pression P est reliée à la densité par

$$\mathrm{P} = c_s^2\rho \qquad c_s = \quad \text{vitesse du son} \quad = \frac{1}{\sqrt{3}} \tag{18.47}$$

et le coefficient de viscosité est relié au temps de relaxation par

$$\nu = c_s^2\left(\frac{\tau}{h} - \frac{1}{2}\right) \tag{18.48}$$

C.3 Conditions aux limites

Comment traiter les conditions aux limites dans le méthode de Boltzmann sur réseau ? Autrement dit, quel doit être le comportement du fluide au contact d'une surface ? La façon la plus simple et la plus courante de traiter une paroi, en fait l'une des forces de la méthode, est la *condition de rebond* (angl. *no slip condition*), qui stipule que la particule qui collisionne avec la paroi est rétrodiffusée ($\mathbf{e}_i \rightarrow -\mathbf{e}_i$). En pratique, cela signifie que l'équation (18.41) ne s'applique pas aux sites du réseau situés sur la paroi. Au contraire, sur un tel site, la distribution $f_i(\mathbf{r}, t+h)$ est simplement donnée par $f_j(\mathbf{r}, t)$, où $\mathbf{e}_j = -\mathbf{e}_i$. Cette condition de rebond revient à dire que le fluide ne peut pas glisser le long de la paroi : celle-ci exerce sur le fluide un frottement statique parfait.

C.4 Algorithme

Résumons ici l'algorithme de la méthode LBM :

1. On commence par mettre en place une distribution initiale $f_i(\mathbf{r})$ associée à des valeurs initiales de la densité ρ et de la vitesse $\mathbf{u}$ (gardons en tête que $\mathbf{r}$ est maintenant un indice discret défini sur une grille régulière). À cette fin, on initialise $f_i = f_i^{\text{eq.}}(\mathbf{u}_0)$ en fonction d'une vitesse initiale spécifiée à l'avance $\mathbf{u}_0(\mathbf{r})$.
2. À chaque pas temporel, on propage les distributions selon l'équation (18.41) sur les noeuds intérieurs de la grille, et on applique les conditions de rebond sur les frontières.
3. On calcule ensuite les nouvelles densités et vitesses selon les expressions (18.39) et (18.40), ce qui nous permet de calculer les nouvelles distributions à l'équilibre.
4. On retourne à l'étape 2 jusqu'à ce que le temps de simulation soit écoulé.

C.5 Exemple

Au lieu d'illustrer la méthode de Boltzmann sur réseau à l'aide d'un code maison, nous allons utiliser l'un des codes libres disponibles : `openLB`. [3] L'un des exemples inclus dans la distribution illustre le passage d'un fluide en deux dimensions entre deux parois horizontales, entre lesquelles figure un obstacle cylindrique – donc à section circulaire. En trois dimensions, cette situation correspondrait au passage d'un fluide dans un conduit cylindrique au milieu duquel un obstacle sphérique a été inséré.

Un instantané de la simulation est illustré à la figure 18.2. La grandeur $|\mathbf{u}|$ de la vitesse d'écoulement est représentée en fonction de la position par un code de couleur. La partie inférieure de la figure représente plutôt le tourbillon (angl. *vorticity*) $\nabla \wedge \mathbf{u}$ associé à la vitesse $\mathbf{u}$. Les deux sections de gauche correspondent à un écoulement caractérisé par un petit nombre de Reynolds (Re=1), alors que les sections de droite correspondent à un écoulement turbulent, caractérisé par Re=800. Le nombre de Reynolds, en hydrodynamique, est le rapport des forces d'inertie aux forces visqueuses. Sa définition pratique est la suivante :

$$\mathrm{Re} = \frac{u\mathrm{L}}{\nu} \tag{18.49}$$

où

3. Voir `http://www.numhpc.org/openlb/`

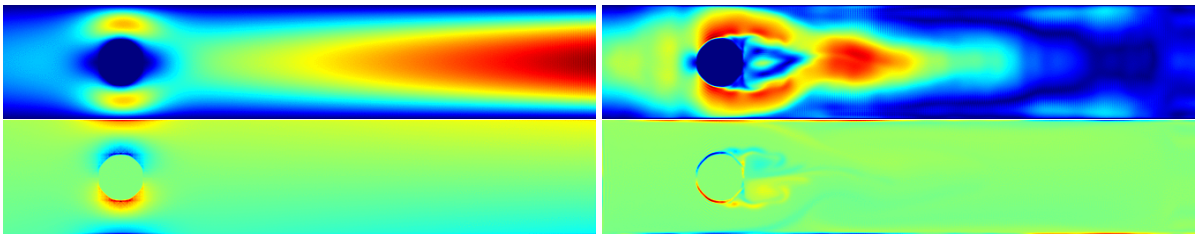


FIGURE 18.2
Simulation du passage d'un fluide 2D entre deux parois, avec un obstacle cylindrique, réalisée à l'aide du logiciel libre `openLB`. En haut : profil de la grandeur de la vitesse $|\mathbf{u}|$ en fonction de la position. Les vitesses plus grandes sont en rouge, les plus faibles en bleu. L'écoulement s'effectue de la gauche vers la droite, et l'obstacle est visible au cinquième de la longueur, à partir de la gauche. En bas, profil du tourbillon $\nabla\wedge\mathbf{u}$ associé au même écoulement. Le tourbillon n'a ici qu'une seule composante (en z) et le code de couleur va des valeurs négatives (bleues) à positives (rouge) en passant par les valeurs nulles (vert pâle). À gauche : écoulement à petit nombre de Reynolds (Re=1), à droite, écoulement turbulent (Re=800).

1. u est la vitesse moyenne du fluide, ou la vitesse d'un objet dans un fluide au repos.
2. L est la longueur caractéristique du système (la distance entre les deux parois dans notre exemple).
3. ν est le coefficient de viscosité cinématique (le même qui figure dans l'équation de Navier-Stokes (18.7)).

Cette définition, comme on le voit, n'est pas universelle : elle dépend du problème étudié. Plus le nombre de Reynolds est petit, plus la viscosité est importante ; plus il est grand, plus la turbulence a de chances de s'établir.

On remarque que le tourbillon $\nabla\wedge\mathbf{u}$ est important autour de l'obstacle dans les deux cas (Re=1 et Re=800), ce qui reflète l'existence de la couche limite, c'est-à-dire une couche mince autour des parois où la vitesse varie de zéro sur la paroi à une valeur non nulle dans un espace assez étroit. Le fait que le tourbillon $\nabla\wedge\mathbf{u}$ soit non nul n'entraîne pas nécessairement l'existence d'un écoulement tourbillonnaire : il faut pour cela déterminer les lignes d'écoulement, ce qui n'est pas visible sur les graphiques de la figure 18.2. Cependant, l'écoulement illustré sur la partie droite de la figure est clairement turbulent.

Problème 18.1 :
Obtenez explicitement les valeurs (18.45) à partir des conditions (18.42) et de la distribution à l'équilibre (18.43), en négligeant les termes d'ordre supérieur à u^2.

Problème 18.2 :
Décrivez comment on pourrait appliquer la méthode décrite dans cette section à une force inter particule générale qui dérive d'un potentiel central U(r) en dimension 3, de sorte que l'énergie

potentielle d'une particule à la position $\mathbf{r}_i$ en présence des autres particules est

$$V(\mathbf{r}_i) = \sum_{j \neq i} U(|\mathbf{r}_i - \mathbf{r}_j|) \tag{18.50}$$

A Formulez la méthode en fonction des transformées de Fourier $\tilde{U}(\mathbf{q})$ de $U(\mathbf{r}) = U(r)$ et $\tilde{\rho}$ de la densité volumique des particules $\rho(\mathbf{r})$.

B Quelle condition le potentiel $U(r)$ doit-il respecter pour que $\tilde{U}$ soit bien défini? Comment peut-on modifier $U(r)$ au besoin, sans conséquences physiques, afin que cette condition soit respectée? Considérez le potentiel de Lennard-Jones comme exemple.

BIBLIOGRAPHIE

[AS64] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover publications, 1964.

[AT10] Vinay Ambegaokar and Matthias Troyer. Estimating errors reliably in monte carlo simulations of the ehrenfest model. *Am. J. Phys.*, 78:150, 2010.

[BB01] J.P. Boyd and J.P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Pubns, 2001.

[CD98] Shiyi Chen and Gary D. Doolen. Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30:329–364, 1998.

[Cha] L. Champaney. Méthodes numériques pour la mécanique. Notes de cours, Université de Versailles, St-Quentin en Yvelines.

[DD01] H.M. Deitel and P.J. Deitel. *Comment programmer en C+*. Éditions R. Goulet, 2001.

[Fit] Richard Fitzpatrick. Computational physics. Lecture notes, University of Texas at Austin.

[For98] B. Fornberg. Calculation of weights in finite difference formulas. *SIAM review*, 40(3):685–691, 1998.

[For08] A. Fortin. *Analyse numérique pour ingénieurs*. Presses inter Polytechnique, 2008.

[GO89] Gene H. Golub and Dianne P. O'Leary. Some history of the conjugate gradient and lanczos algorithms : 1948-1976. *SIAM Review*, 31(1):pp. 50–102, 1989.

[GT88] H. Gould and J. Tobochnik. *An introduction to computer simulation methods : applications to physical systems*. Addison-Wesley Reading (MA), 1988.

[HL97] Xiaoyi He and Li-Shi Luo. Theory of the lattice boltzmann method : From the boltzmann equation to the lattice boltzmann equation. *Phys. Rev. E*, 56(6):6811–6817, Dec 1997.

[LPB08] Rubin H. Landau, Manuel José Páez, and Cristian C. Bordeianu. *A survey of computational physics*. Princeton University Press, 2008.

[Mac] Angus MacKinnon. Computational physics — 3rd/4th year option. Lecture notes, Imperial College, London.

[Pai72] C.C. Paige. Computational variants of the Lanczos method for the eigenproblem. *J. Inst. Maths Applics*, 10:373–381, 1972.

[Pai80] C.C. Paige. Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. *Linear algebra and its applications*, 34:235–258, 1980.

[PTVF07] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes, the art of scientific computing*. Cambridge, 2007.

[SB02] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*. Springer, 2002.