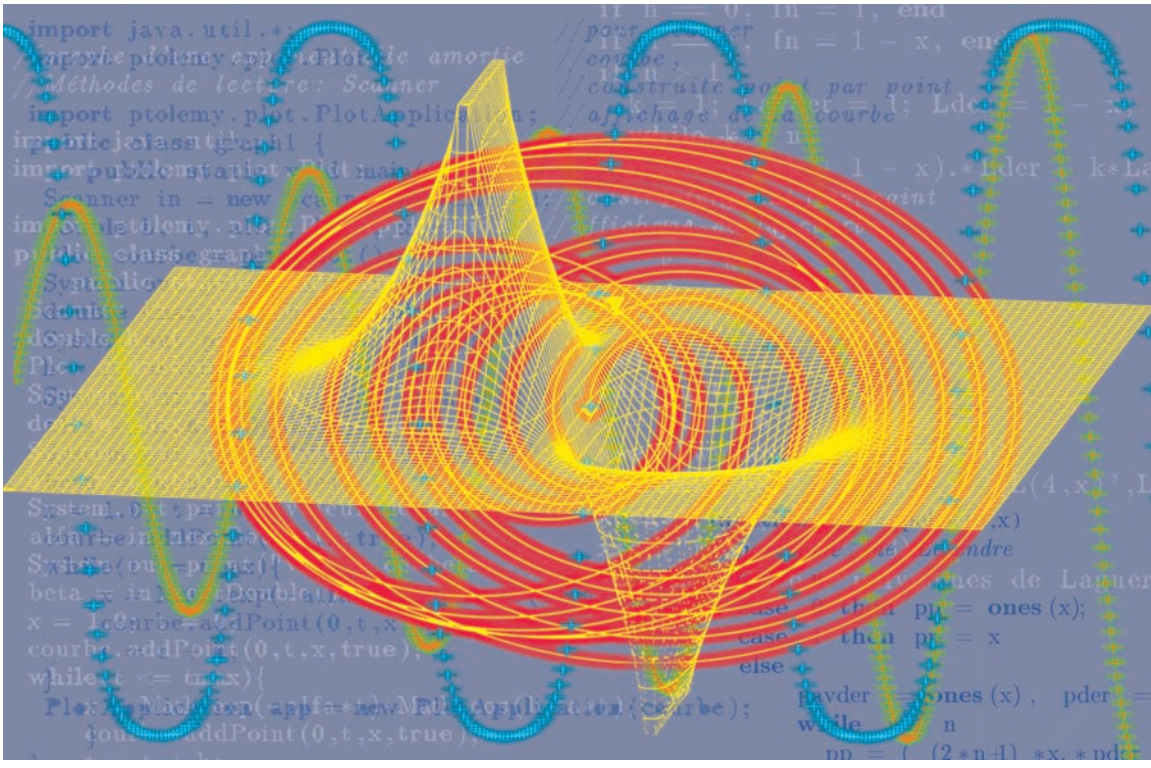
COLLECTION GRENOBLE SCIENCES
DIRIGÉE PAR JEAN BORNAREL

MÉTHODES NUMÉRIQUES APPLIQUÉES

POUR LE SCIENTIFIQUE ET L'INGÉNIEUR

Jean-Philippe GRIVET



EDP SCIENCES

This page intentionally left blank

MÉTHODES NUMÉRIQUES APPLIQUÉES
POUR LE SCIENTIFIQUE ET L'INGÉNIEUR

Grenoble Sciences

Grenoble Sciences poursuit un triple objectif :

- réaliser des ouvrages correspondant à un projet clairement défini, sans contrainte de mode ou de programme,
- garantir les qualités scientifique et pédagogique des ouvrages retenus,
- proposer des ouvrages à un prix accessible au public le plus large possible.

Chaque projet est sélectionné au niveau de Grenoble Sciences avec le concours de referees anonymes. Puis les auteurs travaillent pendant une année (en moyenne) avec les membres d'un comité de lecture interactif, dont les noms apparaissent au début de l'ouvrage. Celui-ci est ensuite publié chez l'éditeur le plus adapté.

Contact : Tél. : (33)4 76 51 46 95 - e-mail : Grenoble.Sciences@ujf-grenoble.fr
Information : ***http://grenoble-sciences.ujf-grenoble.fr***

Deux collections existent chez EDP Sciences :

- la ***Collection Grenoble Sciences***, connue pour son originalité de projets et sa qualité
- ***Grenoble Sciences - Rencontres Scientifiques***, collection présentant des thèmes de recherche d'actualité, traités par des scientifiques de premier plan issus de disciplines différentes.

Grenoble Sciences reçoit le soutien du **Ministère de l'Enseignement supérieur et de la Recherche** et de la **Région Rhône-Alpes**.
Grenoble Sciences est rattaché à l'**Université Joseph Fourier de Grenoble**.

Réalisation et mise en pages : **Centre technique Grenoble Sciences**
Illustration de couverture : **Alice GIRAUD**, d'après les éléments fournis par l'auteur

ISBN 978-2-7598-0386-6

MÉTHODES NUMÉRIQUES APPLIQUÉES
POUR LE SCIENTIFIQUE ET L'INGÉNIEUR

Jean-Philippe GRIVET

17, avenue du Hoggar
Parc d'Activité de Courtabœuf - BP 112
91944 Les Ulis Cedex A - France

Ouvrages Grenoble Sciences édités par EDP Sciences

Collection Grenoble Sciences

Chimie. Le minimum à savoir *(J. Le Coarer)* • Electrochimie des solides *(C. Déportes et al.)* • Thermodynamique chimique *(M. Oturan & M. Robert)* • CD de Thermodynamique chimique *(J.P. Damon & M. Vincens)* • Chimie organométallique *(D. Astruc)* • De l'atome à la réaction chimique *(sous la direction de R. Barlet)* • Spectroscopies infrarouge et Raman *(R. Poilblanc & F. Crasnier)* • Chemogénomique. Des petites molécules pour explorer le vivant *(sous la direction de E. Maréchal, S. Roy & L. Lafanechère)*

Introduction à la mécanique statistique *(E. Belorizky & W. Gorecki)* • Mécanique statistique. Exercices et problèmes corrigés *(E. Belorizky & W. Gorecki)* • La cavitation. Mécanismes physiques et aspects industriels *(J.P. Franc et al.)* • La turbulence *(M. Lesieur)* • Magnétisme : I - Fondements, II - Matériaux et applications *(sous la direction d'E. du Trémolet de Lacheisserie)* • Du Soleil à la Terre. Aéronomie et météorologie de l'espace *(J. Lilensten & P.L. Blelly)* • Sous les feux du Soleil. Vers une météorologie de l'espace *(J. Lilensten & J. Bornarel)* • Mécanique. De la formulation lagrangienne au chaos hamiltonien *(C. Gignoux & B. Silvestre-Brac)* • Problèmes corrigés de mécanique et résumés de cours. De Lagrange à Hamilton *(C. Gignoux & B. Silvestre-Brac)* • La mécanique quantique. Problèmes résolus, T. 1 et 2 *(V.M. Galitsky, B.M. Karnakov & V.I. Kogan)* • Description de la symétrie. Des groupes de symétrie aux structures fractales *(J. Sivardière)* • Symétrie et propriétés physiques. Du principe de Curie aux brisures de symétrie *(J. Sivardière)* • Physique des plasmas collisionnels. Application aux décharges haute fréquence *(M. Moisan & J. Pelletier)* • Energie et environnement. Les risques et les enjeux d'une crise annoncée *(B. Durand)* • Hydrothermalisme. Spéciation métallique hydrique et systèmes hydrothermaux *(M. Chenevoy & M. Piboule)* • Les roches, mémoire du temps *(G. Mascle)* • Physique des diélectriques *(J.C. Peuzin & D. Gignoux)*

Exercices corrigés d'analyse, T. 1 et 2 *(D. Alibert)* • Introduction aux variétés différentielles *(J. Lafontaine)* • Mathématiques pour les sciences de la vie, de la nature et de la santé *(F. & J.P. Bertrandias)* • Approximation hilbertienne. Splines, ondelettes, fractales *(M. Attéia & J. Gaches)* • Mathématiques pour l'étudiant scientifique, T. 1 et 2 *(Ph.J. Haug)* • Analyse statistique des données expérimentales *(K. Protassov)* • Nombres et algèbre *(J.Y. Mérindol)* • Analyse numérique et équations différentielles *(J.P. Demailly)* • Outils mathématiques à l'usage des scientifiques et ingénieurs *(E. Belorizky)*

Bactéries et environnement. Adaptations physiologiques *(J. Pelmont)* • Enzymes. Catalyseurs du monde vivant *(J. Pelmont)* • Endocrinologie et communications cellulaires *(S. Idelman & J. Verdetti)* • Eléments de biologie à l'usage d'autres disciplines *(P. Tracqui & J. Demongeot)* • Bioénergétique *(B. Guérin)* • Cinétique enzymatique *(A. Cornish-Bowden, M. Jamin & V. Saks)* • Biodégradations et métabolismes. Les bactéries pour les technologies de l'environnement *(J. Pelmont)* • Enzymologie moléculaire et cellulaire, T. 1 et 2 *(J. Yon-Kahn & G. Hervé)* • Glossaire de biochimie environnementale *(J. Pelmont)*

L'Asie, source de sciences et de techniques *(M. Soutif)* • La biologie, des origines à nos jours *(P. Vignais)* • Naissance de la physique. De la Sicile à la Chine *(M. Soutif)* • Science expérimentale et connaissance du vivant. La méthode et les concepts *(P. Vignais, avec la collaboration de P. Vignais)* • Histoire de la science des protéines *(J. Yon-Kahn)*

La plongée sous-marine à l'air. L'adaptation de l'organisme et ses limites *(Ph. Foster)* • Le régime oméga 3. Le programme alimentaire pour sauver notre santé *(A. Simopoulos, J. Robinson, M. de Lorgeril & P. Salen)* • Gestes et mouvements justes. Guide de l'ergomotricité pour tous *(M. Gendrier)*

Listening Comprehension for Scientific English *(J. Upjohn)* • Speaking Skills in Scientific English *(J. Upjohn, M.H. Fries & D. Amadis)* • Minimum Competence in Scientific English *(S. Blattes, V. Jans & J. Upjohn)* • Minimum Competence in Medical English *(J. Upjohn, J. Hay, P.E. Colle, J. Hibbert & A. Depierre)*

Grenoble Sciences - Rencontres Scientifiques

Radiopharmaceutiques. Chimie des radiotraceurs et applications biologiques *(sous la direction de M. Comet & M. Vidal)* • Turbulence et déterminisme *(sous la direction de M. Lesieur)* • Méthodes et techniques de la chimie organique *(sous la direction de D. Astruc)* • L'énergie de demain. Techniques, environnement, économie *(sous la direction de J.L. Bobin, E. Huffer & H. Nifenecker)* • Physique et biologie. Une interdisciplinarité complexe *(sous la direction de B. Jacrot)*

Avant-propos

Qu'est-ce que l'analyse numérique ? C'est un ensemble d'outils qui permet d'obtenir une solution numérique approchée d'un problème mathématique, lui-même modèle d'une question technique ou scientifique.

Pourquoi étudier (et enseigner) l'analyse numérique conçue de cette manière ? N'est-il pas suffisant d'appuyer sur la touche « solve » d'une calculette pour résoudre une équation algébrique ? Si l'on veut vraiment utiliser un logiciel, pourquoi faire plus que d'appeler, à l'intérieur d'un logiciel de haut niveau, la fonction « solve » ? En réalité, il est toujours profitable de connaître le principe de fonctionnement des outils que l'on utilise afin de les employer au mieux et pour être conscient de leurs limites. De plus, il peut arriver qu'un programme, bien qu'immédiatement disponible, ne soit pas parfaitement adapté à l'usage prévu ; seul l'utilisateur bien informé pourra le modifier en connaissance de cause et étendre son domaine de validité. Enfin, la curiosité est une qualité légitime chez l'ingénieur ou le scientifique et ce livre aide à soulever les couvercles des « boîtes noires » que sont les algorithmes numériques.

C'est dans cette optique qu'a été conçu l'ouvrage que vous avez entre les mains : le principe de chaque algorithme important est présenté simplement, puis son fonctionnement est illustré par des exemples et les limites sont précisées.

Le livre est fait pour des scientifiques de niveau L2, L3 ou M1 en physique et physique appliquée. L'ouvrage est donc destiné aux étudiants et élèves ingénieurs, mais aussi à tous les utilisateurs de l'outil numérique, en particulier ceux qui ont peu de goût ou de temps pour les démonstrations rigoureuses et qui souhaitent aborder rapidement les applications concrètes.

Le texte est orienté vers la « physique numérique ». Il y a quinze ans, ce terme (ou plutôt ses équivalents anglais, « numerical physics » ou « computational physics ») suscitait une centaine de réponses sur un moteur de recherche. Il en évoque plusieurs millions aujourd'hui. Bien que cet ouvrage ne soit pas, au sens strict, un livre de physique numérique, il s'en approche, par l'intermédiaire de certains exemples, exercices et projets.

Par rapport aux programmes habituels de mathématiques, sont inclus des chapitres qui ne font pas traditionnellement partie de l'analyse numérique comme les polynômes orthogonaux et un rappel de calcul des probabilités. Certains sujets qui, à l'expérience, semblent rébarbatifs, ont été omis, traités sommairement ou introduits assez tard dans le développement. D'autres, au contraire, qui paraissent plus motivants, ont été placés au début.

On trouve d'excellents livres d'analyse numérique en français. Ils s'adressent en général à un public de mathématiciens ou de futurs mathématiciens et sont aussi d'un niveau assez élevé. Nous renvoyons systématiquement à ces ouvrages (Crouzeix et Mignot, Schatzmann, Demailly, Allaire et Kaber, etc.) pour la plupart des démonstrations. Les aspects élémentaires, tels qu'on peut les assimiler pendant les deux premières années post-baccalauréat, ont été privilégiés.

Chaque fois que l'on traite de méthodes numériques, se pose la question du langage de programmation à utiliser. Si l'ouvrage contient quelques exemples de code rédigés en C/C++, en Java, en Python et en Maple, l'essentiel des exemples présentés est écrit en Scilab. Ce logiciel gratuit, puissant et facile à installer, intègre des fonctions graphiques de qualité raisonnable. Il permet la programmation à plusieurs niveaux : niveau global (fonctions telles que « fsolve » pour résoudre une équation non-linéaire ou « ode » pour intégrer une équation différentielle avec conditions initiales) jusqu'au niveau des opérations élémentaires. Cette souplesse est pédagogiquement utile et permet une vérification commode des programmes. Un autre avantage de Scilab est que la syntaxe en est fort simple et facilement assimilable par toute personne formée, même sommairement, à Fortran, C ou Java. Sa ressemblance avec Matlab, un logiciel très répandu, constitue un autre facteur positif.

Chaque chapitre est accompagné d'exercices qui ont été expérimentés par de nombreuses promotions d'étudiants et qui peuvent illustrer utilement le sujet traité. Sont également proposés des énoncés de projets en textes « ouverts » : les données peuvent être incomplètes, la méthode à suivre est parfois décrite assez sommairement. La réalisation du projet demandera donc un investissement personnel certain mais, en contrepartie, permettra au lecteur de se familiariser avec des applications de l'analyse numérique plus proches du monde réel. La liste des questions proposées pour chaque projet n'est pas limitative et peut être complétée selon les intérêts de chacun.

Les chapitres 1 à 3 constituent une sorte d'introduction ou de pré-requis avant d'appliquer une quelconque méthode numérique. Visualiser une fonction est la meilleure façon de découvrir ses propriétés. La programmation détaillée du calcul d'une fonction, même élémentaire, est un exercice profitable, tant du point de vue de l'algorithmique que du point de vue de l'analyse numérique. Enfin, la construction de variables sans dimension est une étape obligée de toute simulation numérique.

L'analyse numérique proprement dite apparaît au chapitre 4 avec l'interpolation. Si la pratique de l'interpolation a beaucoup diminué depuis l'apparition des ordinateurs, son rôle théorique, comme fondement des méthodes d'intégration et de résolution des équations différentielles, est resté. Les méthodes les plus simples de résolution des équations non-linéaires sont présentées (chapitre 5), en consacrant un paragraphe spécial aux polynômes. Puis suit une brève description des familles de polynômes orthogonaux et des leurs principales propriétés (chapitre 7). Bien que ces connaissances soient appliquées au chapitre suivant, lors de la construction de l'algorithme de Gauss-Legendre, cette partie peut être omise en première lecture. Les chapitres 8 et 9 sont consacrés au calcul numérique de dérivées et d'intégrales. Les algorithmes classiques sont passés en revue, y compris l'algorithme de Cooley–Tuckey pour la transformation de Fourier rapide.

L'algèbre linéaire est à l'honneur dans les chapitres 6 et 10. Tout d'abord, la résolution de systèmes d'équations linéaires (chapitre 6), y compris les systèmes surdéterminés, tels qu'on les rencontre lors de l'application de la méthode des moindres carrés. Ce chapitre est accompagné d'une annexe rassemblant quelques définitions et théorèmes d'algèbre linéaire. Le chapitre 10 traite du calcul des valeurs propres et des vecteurs propres.

Sont ensuite abordés les problèmes différentiels : équations différentielles ordinaires avec conditions initiales (chapitre 11), avec conditions aux limites (chapitre 12), puis équations aux dérivées partielles (chapitre 13).

L'ouvrage se termine par deux chapitres consacrés à des aspects non déterministes : quelques rudiments de probabilité appliqués à la propagation des erreurs expérimentales et au lissage par moindres carrés (chapitre 14) puis une description des méthodes de Monte Carlo.

De nombreux sujets ont dû être omis, soit parce qu'ils semblaient trop difficiles ou demandaient un exposé trop long. C'est notamment le cas des méthodes de descente et de gradient conjugué pour la résolution de systèmes linéaires, des méthodes modernes de résolution numérique des équations aux dérivées partielles (collocation, méthodes spectrales, éléments finis) et de la décomposition d'une matrice en valeurs singulières. Les lecteurs pourront compléter leur information grâce aux références et aux mots-clés fournis en fin de chapitre.

REMERCIEMENTS

Pour la préparation du cours qui est l'ancêtre de ce livre, j'ai bénéficié des conseils de nombreux collègues, particulièrement de D. Mas et J.L. Rouet : je leur exprime ici ma reconnaissance. Ce livre n'existerait pas sans l'aide apportée par Grenoble Sciences. Les experts qui ont lu le manuscrit ont, non seulement corrigé un nombre incalculable de fautes de frappe et d'erreurs, mais ont aussi suggéré de nombreuses améliorations. Je rends ici hommage à leur dévouement, leur patience, leur esprit d'observation et leur rigueur mathématique. Le personnel de Grenoble Sciences a aussi droit à ma reconnaissance. Mmes Capolo et Bordage ont géré la genèse du livre avec une extrême bonne volonté, elles ont détecté d'autres erreurs et, avec patience et habileté, redessiné toutes mes figures malhabiles ; je les en remercie vivement.

This page intentionally left blank

Chapitre 1

Représentation graphique de fonctions

L'une des activités les plus fréquentes en informatique scientifique consiste à représenter l'allure d'une fonction à l'aide d'un dessin, et de très nombreux logiciels peuvent répondre à cette attente légitime. Nous présentons dans ce chapitre des exemples choisis dans trois catégories de logiciels : un sous-programme que l'on doit incorporer dans un programme principal pour produire un tracé « en ligne », une application autonome capable de représenter graphiquement des données produites par un autre programme et enfin des logiciels puissants capables de calculer et de tracer. On peut encore distinguer deux cas un peu différents en pratique : ou bien la fonction est définie par une ou des formules ou un programme (c'est une fonction « analytique »), ou bien elle est représentée par un tableau de valeurs créé à l'avance (fonction « numérique »). Dans le premier cas, il faudra programmer la formule correspondante ; dans le deuxième cas, il faudra faire lire un fichier de données par le logiciel considéré, à moins de devoir entrer toutes les valeurs au clavier. Nous expliquons la marche à suivre, pour quelques logiciels pratiques et faciles d'accès, dans les paragraphes qui suivent.

1.1. Les tableurs

Tous les tableurs comportent des outils graphiques puissants. Pour cet exemple, nous utiliserons le programme `Calc` de la collection « OpenOffice.org ». Pour représenter les variations de la fonction $y = \exp(-x/3)\cos(2x)$, nous installons dans la colonne A la suite des valeurs de x. Il suffit de donner les deux premières valeurs (0 dans A1 et 0.05 dans A2 par exemple), de sélectionner ces deux cellules puis de créer toutes les valeurs suivantes (jusqu'à une limite choisie par l'utilisateur) dans la même colonne à l'aide de la « poignée de recopie ». Le programme demande si nous voulons engendrer une progression arithmétique, ce que nous confirmons. La cellule B1 contiendra l'expression de la fonction ; comme il s'agit d'une formule, nous écrivons `=exp(-A1/3)*cos(2*A1)`. Utilisant encore la poignée de recopie, nous reproduisons cette expression dans toutes les cellules utiles de la colonne B, en indiquant dans la boite de dialogue que nous voulons dupliquer une formule. Il faut ensuite cliquer sur l'icône « insertion de diagramme », choisir avec la souris l'emplacement et la taille du dessin et répondre aux questions posées par le logiciel : où se trouvent les données, quelle est la présentation souhaitée.

S'agissant d'un fichier de données, le seul obstacle mineur est la lecture du fichier. Nous supposons que ce fichier (au format ASCII) contient des données produites par un autre programme : ce sont des nombres décimaux (écrits avec un **point décimal**) séparés par plusieurs blancs. Le nom du fichier se terminera de préférence par l'extension « .csv » (pour éviter qu'OpenOffice ne le considère comme un texte, destiné au traitement de texte, « Writer »). Nous ouvrons le fichier et nous indiquons, dans la boite de dialogue, que les valeurs sont séparées par des blancs, qu'il faut regrouper les séparateurs et que le modèle des nombres est anglo-saxon. Le tracé s'effectue comme précédemment. Si le fichier contient deux colonnes, celles-ci sont utilisées automatiquement. Sinon, il faut préciser, avec la souris, la colonne des abscisses et celle des ordonnées.

1.2. JAVA ET PTPLOT

PtPlot est une collection de méthodes (issues d'un gros projet baptisé « Ptolemy ») que l'on peut inclure dans un programme en Java pour tracer des graphes point par point. Le listing suivant permet d'afficher une sinusoïde amortie.

Listing 1.1 – Sinusoïde amortie en Java

```
import java.util.*;                                                    1
import ptolemy.plot.Plot;                                              2
import ptolemy.plot.PlotApplication;                                   3
public class graph1 {                                                  4
    public static void main(String[] args) {                          5
 Scanner in = new Scanner(System.in);                                  6
 double h, t, alfa, beta, x;                                           7
 Plot courbe = new Plot();                                             8
 System.out.print("duree:  ");                                        9
 double tmax = in.nextDouble();                                        10
 System.out.print("valeur du pas: ");                                 11
 h = in.nextDouble();                                                  12
 System.out.print("valeur de alpha: ");                               13
 alfa = in.nextDouble();                                               14
 System.out.print("valeur de beta: ");                                15
 beta = in.nextDouble();                                               16
 x = 1.0; t = 0;                                                       17
 courbe.addPoint(0,t,x,true);                                          18
 while(t <= tmax){                                                     19
    x = Math.exp(-alfa*t)*Math.cos(beta*t);// construction de          20
    courbe.addPoint(0,t,x,true);            // la courbe,              21
    t = t + h;                              // point par point         22
 }                                                                     23
 PlotApplication app = new PlotApplication(courbe);//affichage        24
    }                                                                  25
}                                                                      26
```

Les lignes 1 à 3 permettent l'insertion de méthodes d'entrée-sortie et graphiques. La lecture des données fait appel à la classe `Scanner` (disponible dans les versions de Java postérieures à 2004). Les lignes 6 à 8 déclarent la méthode `in` et l'objet `courbe` ainsi que les variables nécessaires. Les valeurs correspondantes sont lues par les instructions 9 à 16. L'objet `courbe` est initialisé lignes 17 et 18. La boucle `while` construit la courbe point par point ; celle-ci est affichée par l'instruction 24.

Nous vous rappelons les étapes d'une utilisation simplifiée de Java. On crée le programme à l'aide d'un éditeur de texte et on l'enregistre sous le nom `graph1.java` en prenant garde que le nom de la classe et le nom du fichier **coïncident exactement**. Sous Windows, dans une fenêtre « invite de commande », on compile ce programme par l'instruction `javac graph1.java`. Si tout se passe bien, on peut alors lancer l'exécution par `java graph1`. L'ordinateur doit savoir où se trouve le compilateur (il faut donc initialiser correctement la variable d'environnement `PATH`) et où se trouvent les classes PtPlot (initialiser `CLASSPATH`!).

La figure apparaît dans une nouvelle fenêtre interactive : certains paramètres du tracé, comme les échelles des axes, sont modifiables par l'utilisateur (menu « Edit »).

1.3. Python et Matplotlib

Vous pouvez considérer Java et Python comme des descendants de C++, en plus simples et plus commodes. Python (et toutes ses bibliothèques de sous-programmes spécialisés) est, à notre avis, plus facile d'emploi et plus commode que Java dans le domaine scientifique. Il souffre cependant d'un inconvénient : c'est un langage « interprété » assez lent. Il faut l'utiliser pour de petits programmes ou apprendre à l'interfacer avec des modules Fortran ou C qui exécuteront les gros calculs.

Listing 1.2 – Sinusoïde amortie en Python

```
from pylab import *                                          1
t = arange(0,2*pi,0.01)                                      2
alfa = float(raw_input("valeur de alfa: "))                  3
beta = float(raw_input("valeur de beta: "))                  4
ca = exp(-alfa*t)*cos(beta*t)                                5
plot(t,ca)                                                   6
show()                                                       7
```

Pour utiliser Python, il vous faut installer Python, les bibliothèques scientifiques Scipy et Numpy et la bibliothèque graphique Matplotlib ; l'environnement de programmation interactif IPython (et ses accessoires pour Windows) est commode. Si vous voulez lancer Python depuis un répertoire quelconque, vous devrez faire figurer son chemin d'accès dans `PATH`. L'utilisation est semblable à celle de Java : avec votre éditeur de texte favori, vous créez un programme (`graph1.py` par exemple) dont vous demandez l'interprétation et l'exécution (depuis une fenêtre « invite de commande ») par `python graph1.py`. Vous pouvez tout aussi bien taper `%run graph1.py` dans IPython.

Comme le montre l'exemple (lui-même réduit à l'essentiel), ce langage est compact et puissant. Une grande partie du travail est fait, en coulisse, par la ligne 1 qui « importe » de très nombreuses fonctions mathématiques et graphiques. La ligne 2 crée une liste de valeurs de `t`, les lignes 3 et 4 permettent d'entrer, au clavier, les valeurs de α et de β et la ligne 5 calcule, point par point, les valeurs de la fonction. Celle-ci est affichée (selon la syntaxe de Matlab, d'où le nom de la bibliothèque) par l'instruction 6. La ligne 7 empêche la disparition de votre beau graphique dès la fin de l'exécution du programme.

1.4. Gnuplot

Gnuplot est un logiciel gratuit et puissant, disponible pour tous les systèmes d'exploitation ; il a l'avantage d'être assez peu encombrant (2,5 Mo environ). Il possède un analyseur syntactique qui connaît la plupart des fonctions élémentaires. Il est très facile d'accomplir avec Gnuplot les deux tâches que nous nous sommes proposées. Pour tracer une fonction définie par une formule :

```
gnuplot>  alfa = 0.3; beta = 4;
gnuplot>  plot [0:10] exp(-alfa*x)*cos(beta*x);
```

La fonction `plot` admet comme premier paramètre l'intervalle de variation de la variable indépendante, laquelle doit s'appeler x, par convention.

Pour afficher un fonction définie comme une suite de valeurs (x, y), rangées en colonnes dans le fichier `ex142.dta`, il faut écrire :

```
gnuplot>  plot [-2.5:0] [-3.5:1.5] "C:/an_poly/ex142.dta"
```

Les obliques remplacent les contre-obliques de MS-DOS ; nous avons précisé l'intervalle de variation pour les abscisses (premières valeurs entre crochets) et pour les ordonnées (valeurs suivantes). Si le fichier comporte plusieurs colonnes, nous pourrons indiquer que les nombres de la colonne 2 devront servir d'abscisses et ceux de la colonne 5 d'ordonnées en ajoutant simplement `using 2:5` à la fin de l'instruction précédente.

Il existe un grand nombre d'options, pour modifier les caractéristiques du tracé (affichage de légendes, utilisation de symboles ou de traits continus, épaisseur et couleur du trait) et pour choisir le type de graphe (paramétrique, polaire). Vous pourrez les découvrir en lisant l'aide en ligne ou encore en sauvegardant votre travail dans un fichier sur disque (touche `SAVE`) ; en ouvrant ce fichier dans un éditeur de texte, vous verrez que Gnuplot enregistre un grand nombre d'options et de paramètres. Vous pourrez alors modifier progressivement ces valeurs et afficher le tracé enrichi par l'instruction `load`.

Spontanément, gnuplot utilise une fenêtre comme dispositif de sortie. Cette fenêtre est interactive et certains paramètres du dessin peuvent être modifiés en cliquant sur la rubrique « option » du menu déroulant. On peut aussi envoyer le tracé dans un fichier destiné à une imprimante, selon le format souhaité (PNG, GIF, HPGL et bien d'autres).

Le même programme peut représenter des surfaces en perspective ou des courbes de niveau (voir le fichier `all.dem`).

1.5. Maple

Le logiciel Maple est orienté vers le calcul algébrique ; il possède cependant des possibilités graphiques extrêmement puissantes, dont voici un tout petit aperçu. Le dialogue suivant permet de tracer une sinusoïde amortie.

```
> y := exp(-alpha*x)*cos(beta*x) ;
```

$$y := e^{(-\alpha\, x)} \cos(\beta\, x)$$

```
> y1 := subs(alpha = 0.3, beta = 4,y) ;
```

$$y1 := e^{(-0.3\, x)} \cos(4\, x)$$

```
> plot(y1,x = 0..10) ;
```

Nous aurions pu nous contenter de l'instruction unique

```
> plot( exp(-0.3*x)*cos(4*x),x = 0..10);
```

mais la version précédente nous permet de définir une quantité y dépendant de deux paramètres dont nous pouvons modifier la valeur aisément par l'instruction `subs`.

La lecture d'un tableau de valeurs externe se fait au moyen de l'instruction `readdata` qui admet deux paramètres obligatoires, le nom du fichier et le nombre de colonnes. On peut aussi préciser s'il s'agit d'entiers ou de nombres fractionnaires. La manoeuvre est simple dans le cas d'un tableau à deux colonnes, comme le montre l'exemple ci-dessous. Nous traçons le graphe correspondant à l'aide des instructions

```
>    M := readdata("C:/an_poly/ex122.dta",2);
```

$$M := [[-2.,\ -3.1],\ [-1.1,\ -0.99],\ [0.,\ 1.],\ [2.222,\ 3.1415],\ [4.,\ 5.]]$$

```
>    plot(M);
```

Les nombres entiers présents dans le fichier ont été transformés en nombres fractionnaires. Remarquez aussi que les contre-obliques du nom de fichier sont remplacées par des obliques sous Maple. Il est un peu plus compliqué d'extraire d'un fichier multi-colonnes une colonne d'abscisses et une colonne d'ordonnées. Il faut procéder comme ceci. Le fichier dont le nom complet est `D:\an_poly\graph02.dta` contient les valeurs :

```
-1.9  -1     -2.2
-1    -0.04  -1
0      0.97   0.35
0.9    1.87   2.43
2      3      4
3.21   3.96   4.43
```

On lit et on extrait les bonnes valeurs grâce aux instructions

```
> M := readdata("D:/an_001/graph02.dta",3);
```

$M :=$ [[−1.9, −1., −2.2], [−1., −0.04, −1.], [0., 0.97, 0.35], [0.9, 1.87, 2.43], [2., 3., 4.], [3.21, 3.96, 4.43]]

```
> points := [seq([M[i,1],M[i,3]],i=1..6)];
```

points := [[−1.9, −2.2], [−1., −1.], [0., 0.35], [0.9, 2.43], [2., 4.], [3.21, 4.43]]

```
> plot(points,style=POINT,symbol=BOX,symbolsize=15);
```

Maple considère M comme une liste de listes, chaque liste élémentaire représentant une ligne. Le code extrait les nombres de la première colonne (abscisses) et ceux de la troisième colonne (ordonnées), ligne par ligne, puis reporte les points sur un graphique (figure 1.1).

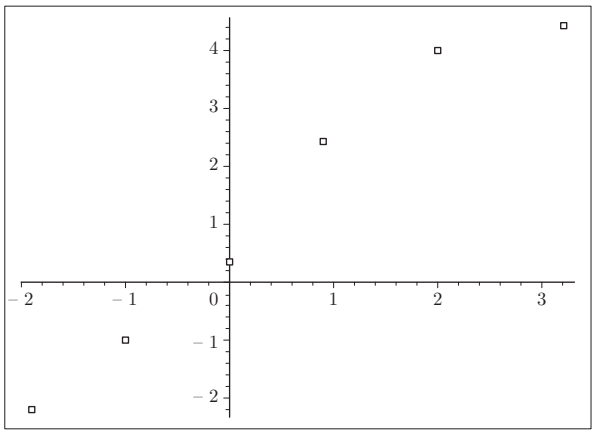


Figure 1.1 – Le tracé d'une suite de points par Maple.

1.6. SCILAB

Scilab est un logiciel gratuit, très bien adapté à l'algèbre linéaire et à la simulation des systèmes dynamiques. On peut l'utiliser de façon interactive (comme une calculette) ou préparer, à l'aide d'un éditeur de texte (par exemple celui qui est incorporé dans Scilab depuis la version 2.7), un programme que l'on fera exécuter ensuite. Nous utilisons, dans ce chapitre, essentiellement la partie graphique de Scilab.

Nous souhaitons encore tracer une sinusoïde amortie, définie par l'équation

$$x = \exp(-\alpha t)\cos(\beta t),$$

pour $\alpha = 0.3, \beta = 2$ et $0 \leq t \leq 10$. Les instructions suivantes répondent à notre souhait.

```
alfa = 0.3; beta = 2;
t = 0:0.1:10;
x = exp(-alfa*t).*cos(beta*t);
plot2d(t,x)
```

Le résultat apparaît sur la figure 1.2. Ce n'est pas ici le lieu de détailler la syntaxe de Scilab, qui est très bien expliquée dans l'aide en ligne, dans les manuels et sur divers sites (voir les références en fin de chapitre) ; nous nous contenterons de quelques indications.

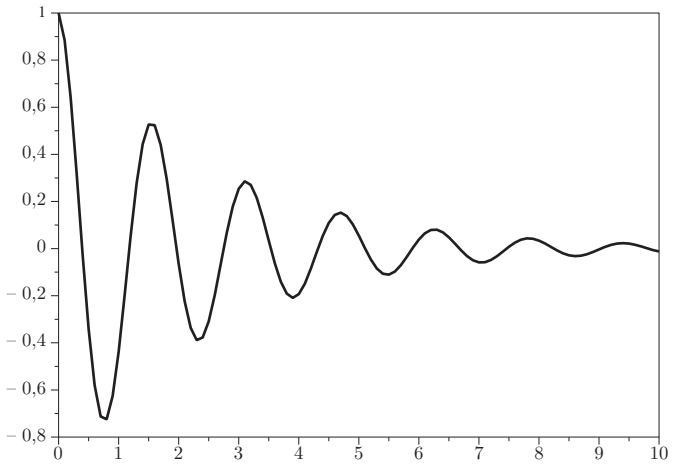


Figure 1.2 – Une sinusoïde amortie représentée par Scilab.

La première ligne définit et initialise les deux paramètres ; les points-virgules indiquent à Scilab qu'il ne doit pas afficher à l'écran les valeurs qui viennent d'être définies. La deuxième ligne initialise un vecteur ligne, t, dont les coordonnées sont $t_1 = 0, t_2 = 0.1, t_3 = 0.2, \ldots, t_{101} = 1.0$; les éléments de tableaux (vecteurs, matrices) sont toujours numérotés à partir de 1. La ligne suivante calcule le vecteur x. Vous pouvez constater qu'il se passe pas mal de choses « en douce ». Les composantes du vecteur t sont successivement utilisées comme arguments des fonctions exponentielle et cosinus. Ainsi, `exp(-alfa*t)` est un vecteur ligne de composantes $\exp(-\alpha * t_i)$. Les deux tableaux représentés par `exp(-alfa*t)` et `cos(beta*t)` sont ensuite multipliés **élément par élément** (c'est ce qu'indique la notation « **point-étoile** » du produit) pour former le résultat, un vecteur ligne de coordonnées $x_i = \exp(-\alpha t_i)\cos(\beta t_i)$. Le programme relie ensuite par des segments les points successifs de coordonnées (t_i, x_i), pour produire un tracé lisse.

La fonction `plot2d` peut recevoir des arguments supplémentaires, qui permettent de définir des axes, des graduations, de modifier l'épaisseur et la couleur des traits, d'introduire des légendes ou un titre.

En ajoutant la ligne `y = exp(-alfa*t).*sin(beta*t)` et en modifiant l'instruction de tracé, qui devient `plot2d(t,[x,y])`, nous obtenons, sur le même graphe, les

courbes représentant les deux fonctions x et y. Nous pouvons aussi considérer ces deux fonctions ensemble comme la représentation paramétrique d'une courbe ; celle-ci s'obtient facilement par l'instruction graphique `plot2d(x,y)`.

Comment procéder pour tracer la courbe correspondant à un fichier de valeurs numériques ? C'est encore plus simple.

Nous supposons que le fichier `C:/an_poly/ex112.dta` contient les données qui nous intéressent, à raison de trois par ligne, ces nombres étant séparés par des espaces ou une tabulation. Nous procédons alors comme suit

```
M = read("C:/an_poly/ex112.dta", -1,3);
plot2d(M(:,1),M(:,3),style = -3)
```

Scilab admet aussi bien les obliques que les contre-obliques dans les noms de fichiers. À la première ligne, on lit le contenu du fichier et on range les valeurs dans la matrice M. La valeur `-1` oblige Scilab à lire toutes les lignes de `C:/an_poly/ex112.dta` quel que soit leur nombre et le point-virgule est là pour l'empêcher d'afficher ces valeurs à l'écran. On représente ensuite graphiquement tous les nombres de la troisième colonne de M (ordonnées) en fonction des valeurs correspondantes de la première colonne (abscisses). Sauf indication contraire, Scilab relie les points par des segments. Pour éviter cela, nous avons indiqué un `style` de trait négatif (-3 ici) ; Scilab représente alors des points isolés à l'aide du symbole correspondant.

1.7. Grace

Les utilisateurs du système Linux ont à leur disposition de nombreux autres outils graphiques gratuits. Citons la collection de programmes « Plotutils » (qui se lancent depuis la ligne de commande), la bibliothèque « pgplot » conçue pour s'interfacer facilement avec des programmes en Fortran (ou C) et enfin le somptueux « xmgrace » interactif. Ce logiciel possède les mêmes fonctionnalités que gnuplot à l'exception des représentations de courbes ou surfaces à trois dimensions, mais il est complètement interactif. On peut entrer une formule dans une fenêtre de commande pour représenter une fonction analytique ou importer des données contenues dans un fichier. Toutes les options sont accessibles par des menus déroulants. On peut également sauvegarder un graphique sous forme de fichier texte (terminaison `.agr`). En relisant un fichier de ce type à l'aide de votre éditeur de texte favori, vous constaterez que Grace a enregistré de très nombreux paramètres du tracé, que vous pourrez modifier à loisir : ils seront disponibles lorsque vous relirez le fichier `.agr`.

1.8. Pour en savoir plus

– PtPlot :
`http://ptolemy.eecs.berkeley.edu`

– Python et cie :
`http://python.org`

http://scipy.org
http://ipython.scipy.org
http://matplotlib.sourceforge.net

– Scilab :
http://www.scilab.org
Distributions pour toutes plates-formes.
La page http://www.scilab.org/publications/ contient des références de livres et des liens vers de nombreux textes pédagogiques gratuits.

– Maple :
http://www.maplesoft.com/
http://lumimath.univ-mrs.fr/~jlm/cours/maple/maple.html
http://algo.inria.fr/ : voir la page personnelle de M. Dumas.
http://pagesperso-orange.fr/eddie.saudrais/index.html

– gnuplot :
http://www.gnuplot.info/
ftp://ftp.irisa.fr/pub/gnuplot/

– Grace :
http://plasma-gate.weizmann.ac.il/Grace/

– plotutils :
http://www.gnu.org/software/plotutils/

– pgplot :
http://www.astro.caltech.edu/~tjp/pgplot/

Il existe encore de très nombreux logiciels de visualisation ou de calcul que nous n'avons pas eu la possibilité de mentionner dans le texte. Voici les sites Internet relatifs aux plus connus d'entre eux.

– Mathematica : http://www.wolfram.com

– Mupad : http://www.mupad.de

– Matlab : http://www.mathworks.fr/

– O-Matrix : http://www.omatrix.com

– Origin : www.originlab.com/

– Octave : http://www.gnu.org/software/octave/

– R : http://www.r-project.org

– Root : http://root.cern.ch

– Sage : http://www.sagemath.org/

1.9. Exercices

Exercice 1

En utilisant le logiciel de votre choix, tracer (séparément ou sur un même graphique) les courbes représentant les deux fonctions

$$x = \exp(-\alpha t)\cos(\beta t) \quad ; \quad y = \exp(-\alpha t)\sin(\beta t)$$

avec $\alpha = 0.25, \beta = 6$, puis la courbe d'équations paramétriques $x(t), y(t)$. Reproduire enfin la figure 1.3 ci-dessous.

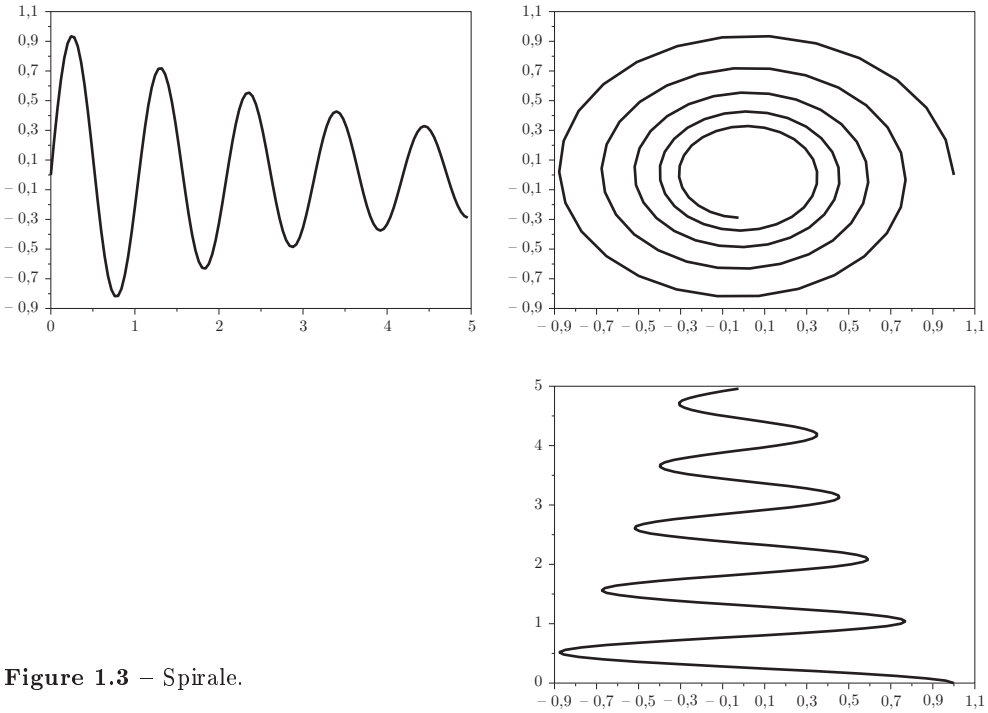


Figure 1.3 – Spirale.

Exercice 2

On a déterminé le nombre d'atomes radioactifs présents dans un échantillon au cours du temps, avec les résultats suivants.

t	0	1	2	3	4	5	10
$N(t)$	1000	370	130	50	17	8	1

Représenter graphiquement ces données, en coordonnées normales ou logarithmiques.

Exercice 3

Examiner, en fonction des valeurs des entiers L, M et N, l'aspect des courbes d'équations paramétriques

$$x = \sin(Lt)\cos(Mt); \quad y = \sin(Lt)\sin(Nt).$$

Exercice 4

a) Tracer, sur un même graphique, les courbes représentatives des fonctions de t

$$x(t) = \cos t \cos \frac{t}{2}; \quad y(t) = \sin t \cos 3t.$$

Quelle est la période de chacune de ces fonctions ?

b) Tracer la courbe d'équations paramétriques

$$\begin{cases} x = \cos t \cos \frac{t}{2}, \\ y = \sin t \cos 3t. \end{cases}$$

c) Cette courbe présente, pour $-1 \leq x \leq -0,8$ et $0,8 \leq x \leq 1$, deux boucles fermées. Déterminer aussi précisément que possible, les intervalles en t correspondants. Tracer chacune de ces boucles à grande échelle.

Exercice 5

On donne les deux équations

$$f = t - a \sin t; \quad g = 1 - a \cos t$$

où t est la variable et a un paramètre.

a) Représenter graphiquement les fonctions f et g pour $a = 0,5; 1; 1,5$.

b) Pour les mêmes valeurs de a, tracer les courbes d'équations paramétriques $x = f(t), y = g(t)$.

c) Pour $a = 1,5$, la courbe du (b) coupe l'axe horizontal. Déterminer les abscisses des deux points d'intersection les plus proches de l'origine. La courbe présente aussi des points doubles ; à quelle valeur de t correspond le point double situé sur l'axe vertical ?

Exercice 6

On considère la série de Fourier

$$s(t) = \frac{4}{\pi} \left[\cos x - \frac{1}{3} \cos 3x + \frac{1}{5} \cos 5x - \frac{1}{7} \cos 7x + \cdots \right]$$

dont le terme général s'écrit

$$\frac{(-1)^n}{2n+1} \cos(2n+1)x.$$

a) Quelle est la période de $s(t)$? Quelle est sa parité ? Représenter graphiquement chacun de trois premiers termes de la série.

b) On appelle s_n la somme partielle de la série limitée aux termes d'indices inférieurs ou égaux à n ; ainsi, s_2 comporte les termes en $\cos x, \cos 3x, \cos 5x$. Pour $-\pi \leq x \leq 2\pi$, tracer les courbes représentatives de s_3, s_4, s_6, s_{10}. Les sommes partielles semblent converger vers une limite, laquelle ?

Exercice 7

Utiliser la fonction Scilab `param3d` pour représenter en perspective l'hélice d'équations paramétriques

$$x = \cos 2\pi t; \quad y = \sin 2\pi t; \quad z = t.$$

Exercice 8

En 1933, paraissait chez Teubner (Leipzig et Berlin) le livre « Funktionentafeln mit Formeln und Kurven » des professeurs E. Jahnke et F. Emde. Ce livre est illustré d'une profusion de courbes et de surfaces qui restent remarquables encore actuellement. Les logiciels modernes permettent toutefois de reproduire sans trop de peine les travaux des étudiants disciplinés de Jahnke et Emde. Nous vous proposons d'utiliser Scilab (et la fonction `besselj(n,x)`) pour tracer la surface représentant $J_n(x)$, où l'indice n est considéré comme une variable continue. Vous devriez obtenir une représentation proche de celle de la figure 1.4. La distribution de Scilab comprend de nombreuses démonstrations ; parmi celles-ci, des représentations en perspective des fonctions élémentaires d'un argument complexe, tout à fait dans l'esprit de cet exercice.

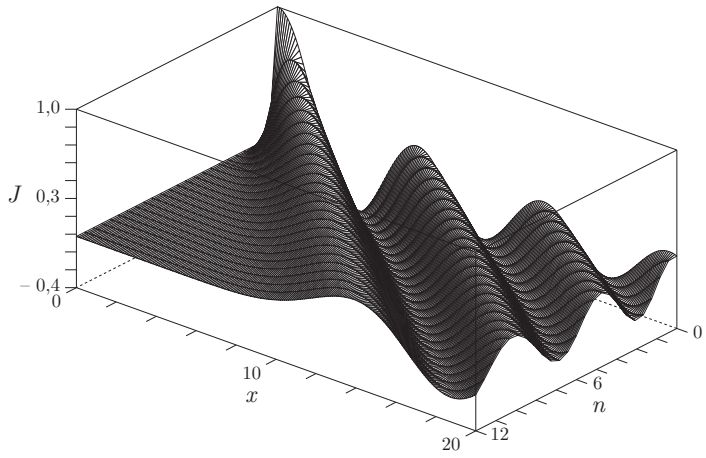


Figure 1.4 – Fonction $J_n(x)$.

Exercice 9

Deux charges électriques, $+q$ et $-q$, sont disposées respectivement aux points $(a, 0)$ et $(-a, 0)$. Exprimer le potentiel électrique U en un point (x, y) du plan et utiliser un logiciel pour tracer quelques courbes de niveaux de U (courbes équipotentielles). Vous devriez obtenir un dessin semblable à la figure 1.5.

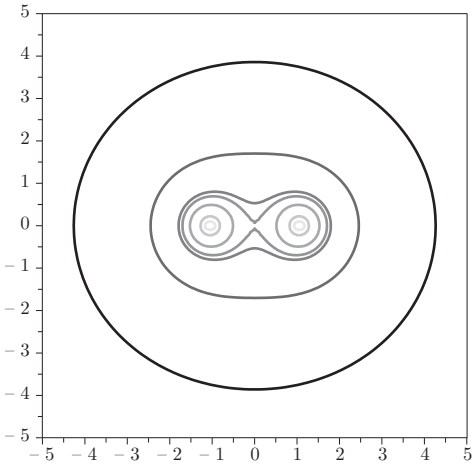


Figure 1.5 – Équipotentielles.

Exercice 10

On considère deux circuits électriques couplés par une inductance mutuelle (figure 1.6).

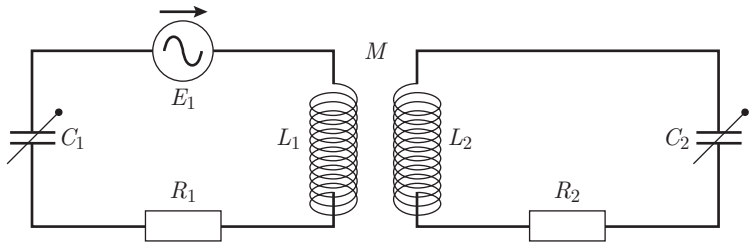


Figure 1.6 – Circuits couplés.

Le circuit 1 (le primaire) est alimenté par un générateur de tension sinusoïdale de pulsation ω et d'amplitude E_1. On s'intéresse à l'amplitude $|I_2|$ du courant dans le circuit 2 (le secondaire), en régime permanent. Pour cela, on introduit les notations $\omega_i = 1/\sqrt{L_i C_i}, \quad i = 1, 2$ (les pulsations de résonance de chaque circuit en l'absence de couplage) et $b_i = 1 - \omega_i^2/\omega^2$. On pose $Q_i = \omega L_i/R_i$ et $k = |M|/\sqrt{L_1 L_2}$ (le coefficient de couplage). Avec ces notations, l'amplitude du courant dans le circuit secondaire s'écrit

$$|I_2| = \frac{kE_1}{\omega\sqrt{L_1 L_2}} \frac{1}{\sqrt{\left(\dfrac{1}{Q_1 Q_2} - b_1 b_2 + k^2\right)^2 + \left(\dfrac{b_2}{Q_1} + \dfrac{b_1}{Q_2}\right)^2}}$$

En pratique, les résistances et les inductances sont fixes, mais les capacités sont réglables. Il est alors commode d'étudier $|I_2|$ en fonction des variables $w_1 = \omega_1/\omega$ et $w_2 = \omega_2/\omega$ Représenter, en perspective ou par des courbes de niveau, la surface $|I_2|$ fonction de w_1 et de w_2.

On peut choisir par exemple, $Q_1 = Q_2 = 8$ ou $Q_1 = 4, Q_2 = 16$. Le coefficient de couplage k est toujours inférieur à 1. Vérifier qu'il existe une valeur de k en-dessous de laquelle la surface ne présente qu'un seul maximum ; l'étude analytique montre que cette valeur critique est

$$k_c = \frac{1}{\sqrt{Q_1 Q_2}}$$

Chapitre 2

Calcul et approximation de fonctions

Sachant qu'un ordinateur ne connaît que les 4 opérations de l'arithmétique (addition, soustraction, multiplication et division), comment pouvons-nous le convaincre de calculer des valeurs numériques de $\sqrt{x}$, $\cos x$ ou $J_3(x)$ (une fonction de Bessel de première espèce) ? Nous devrons faire appel à un algorithme, tout comme le ferait un humain qui chercherait à calculer à la main $\sqrt{23}$. Un algorithme est une suite d'opérations élémentaires, **en nombre fini**, dont l'exécution correcte, **dans le bon ordre**, fournit le résultat souhaité. Il existe des algorithmes de calcul pour chaque fonction. Certains datent de l'antiquité (pour le calcul du plus grand commun diviseur de deux entiers ou le calcul des racines carrées par exemple), d'autres n'ont que quelques années d'existence. L'utilisateur devrait donc, en principe, écrire un programme (la traduction de l'algorithme en un langage informatique) pour calculer la ou les fonctions qui l'intéresse. En réalité, une grande partie de ce travail est déjà faite ; depuis quelques années, les microprocesseurs incorporent un opérateur mathématique capable de calculer vite et bien les fonctions élémentaires. Les compilateurs comportent également des sous-programmes de calcul de fonctions, plus ou moins nombreux selon le compilateur.

Cependant, on rencontre encore des « fonctions spéciales » telles que les fonctions de Bessel, les polynômes de Legendre ou les fonctions elliptiques pour lesquelles il n'existe pas de programme immédiatement disponible. On peut alors rechercher le programme convenable dans les livres ou dans les bibliothèques de sous-programmes ou écrire un programme soi-même.

L'utilisation d'un programme « tout fait » est particulièrement recommandée dans un cadre professionnel. Par contre, dans un contexte d'apprentissage, nous estimons qu'il est extrêmement instructif et utile d'apprendre à programmer le calcul d'une fonction : on doit découvrir le « bon » algorithme, on apprend à se défier des erreurs d'arrondi ou de troncation, enfin on s'entraîne à rédiger un programme correct. Dans tous les cas simples, la vérification du programme est immédiate, par comparaison avec le résultat fourni par le compilateur ou une calculette. C'est dans cet esprit que nous avons rédigé les paragraphes qui suivent.

Il existe une catégorie de fonctions qui se calculent exactement en utilisant uniquement des opérations arithmétiques : ce sont les polynômes et les fractions rationnelles. Ils font l'objet du prochain paragraphe.

2.1. Polynômes et fractions rationnelles

Un polynôme sous forme générale s'écrit

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n.$$

Les coefficients a_k et la variable x sont ici réels. S'il n'y a aucune difficulté pour calculer la valeur numérique de $p(x_0)$, pour la valeur $x = x_0$ de la variable, nous pouvons cependant nous demander quelle est la façon la plus rapide de parvenir au résultat.

La méthode « naïve » consiste à calculer séparément la valeur de chaque puissance de x_0, puis à multiplier chaque quantité x_0^k par le coefficient a_k correspondant, puis à faire la somme des résultats intermédiaires. Pour obtenir $a_k x_0^k$, on doit faire k multiplications ($n \geq k \geq 1$) et donc $n(n+1)/2$ multiplications en tout ; il s'y ajoute n additions. L'algorithme naïf demande donc un nombre d'opérations qui varie comme $\frac{1}{2}n^2$ lorsque n est grand ; on dit qu'il présente une complexité d'ordre n^2.

Une méthode plus économique est fondée sur la remarque que le calcul de x_0^k peut se faire simplement à partir de la valeur de x_0^{k-1} : $x_0^k = x_0 \times x_0^{k-1}$. On calcule l'ensemble des $x_0^k, k = 2 \ldots n$ en $n-1$ multiplications et on garde en mémoire tous les résultats. Il faut ensuite multiplier chaque puissance de x par le coefficient convenable ($n-1$ opérations) et faire la somme. Vous vérifierez sans peine que le nombre total d'opérations est équivalent à $3n$.

L'algorithme de Horner perfectionne le méthode précédente. Il repose sur le fait que le polynôme peut s'écrire

$$p(x) = ((a_n \times x + a_{n-1}) \times x + a_{n-2}) \times x + \cdots + a_1) \times x + a_0. \tag{2.1}$$

Sous cette forme, le calcul de la valeur numérique nécessite n multiplications et n additions, soit une « complexité » d'ordre n. Comment programmer pratiquement ce calcul ? Nous supposons que le polynôme est représenté en mémoire par le vecteur de ses coefficients, $a = [a_0, a_1, \ldots, a_n]$. Si le degré est élevé, il est fastidieux d'écrire en entier la formule précédente et il vaut mieux utiliser la récurrence

$$z_0 = a_n \quad ; \quad z_k = x z_{k-1} + a_{n-k}, \quad k = 1, 2, \ldots, n. \tag{2.2}$$

On peut faire ce calcul à la main, en utilisant une disposition comme ci-dessous ; il s'agit de calculer la valeur de $x^5 + 2x^3 - 3x^2 + 4x - 1$ pour $x = 2$.

1	0	2	-3	4	-1
1	2	6	9	22	43

La première ligne contient les coefficients du polynôme; remarquez qu'il faut faire figurer les coefficients nuls. Le premier élément de la deuxième ligne est z_0, il est égal à a_n ; le deuxième élément (z_1) vaut $2z_0 + a_{n-1} = 2z_0 = 2$, le troisième est $z_2 = 2z_1 + a_{n-2} = 2 \times 2 + 2 = 6$. La valeur de polynôme est $z_n = 43$.

Une fraction rationnelle est le quotient de deux polynômes : il suffit donc de calculer comme précédemment le numérateur et le dénominateur.

2.2. Relations de récurrence

Certaines fonctions utiles obéissent à des relations de récurrence, le plus souvent à trois termes. La fonction cosinus en fournit un exemple élémentaire :

$$\cos(k+1)x = 2\cos kx \cos x - \cos(k-1)x.$$

On peut utiliser cette relation pour établir rapidement une table des valeurs de $\cos kx$, les valeurs de l'argument étant espacées de x (« l'intervalle tabulaire »).

Exemple

Listing 2.1 – Calcul d'un polynôme de Legendre

```
function y = poleg(n,x)
//polynome de Legendre
select n
  case 0 then pp = ones(x);
  case 1 then pp = x
  else
    pavder = ones(x), pder = x; i = 1;
    while i < n
      pp = ( (2*n+1)*x.*pder - n*pavder  )/(n+1);
      pavder = pder; pder = pp;
      i = i + 1;
    end
   end
y = pp;
endfunction
```

Les polynômes de Legendre obéissent à la relation de récurrence suivante

$$(n+1)P_{n-1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

qui permet un calcul rapide et précis de $P_n(x)$ pourvu que l'on connaisse $P_0 = 1$ et $P_1(x) = x$. Les deux programmes ci-contre (listings 2.1 et 2.2) permettent le calcul puis l'affichage de P_n sous Scilab.

Listing 2.2 – Tracé d'un polynôme de Legendre

```
getf("C:\an_poly\legendre.sce");
n = input("degre du polynome: ");
x = linspace(-1,1,200);
y = poleg(n,x);
xset("window",0),xbasc(0)
plot2d(x,y)
```

Remarquez l'instruction `pp = ones(x)` qui crée un vecteur de la même taille que x et dont toutes les composantes sont égales à 1. Comme expliqué au § 1.6, la construction « .* » permet de faire le produit **composante par composante** de deux vecteurs : si $a = \{a_i\}$ et $b = \{b_i\}$, alors $a. * b = \{a_i b_i\}$. On effectuerait de façon analogue une division ou une élévation à la puissance. Ce type de calcul est beaucoup plus rapide qu'une boucle `for`. La figure 2.1 montre le résultat dans le cas $n = 9$.

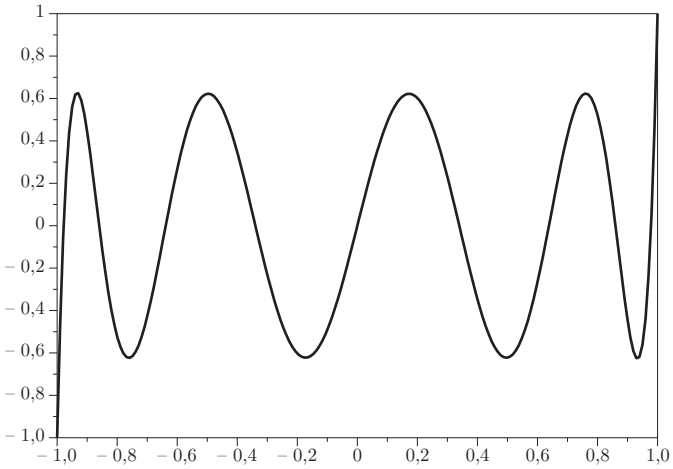


Figure 2.1 – Le polynôme de Legendre d'ordre 9.

2.3. Développement limité

La grande majorité des fonctions que l'on rencontre en sciences physiques admet un développement en série ; il est donc tentant d'utiliser un développement en série tronqué (un polynôme) pour le calcul numérique d'une telle fonction.

Dans les paragraphes précédents, il n'y avait aucune approximation ; si les résultats n'étaient pas tout à fait exacts, cela était du aux erreurs d'arrondi, sur lesquelles nous reviendrons à la fin de ce chapitre. L'utilisation d'un développement en série tronqué au terme de rang n fait apparaître une autre source d'erreur, liée à la méthode utilisée elle-même : au lieu d'une série infinie, nous manipulons un polynôme à n termes, commettant ainsi ce que l'on appelle une erreur de troncation (ou erreur de méthode).

Cette erreur est souvent assez facile à borner, à condition de savoir que la variable indépendante est contenue dans un intervalle défini.

Pour montrer les avantages et inconvénients de cette approche, nous choisissons l'exemple simple de la fonction e^{-x}, que nous voudrions approcher par son développement en série tronqué, avec une erreur absolue inférieure au millième, sur l'intervalle [0,10]. Le terme général du développement est $u_k = (-1)^k \frac{x^k}{k!}$. La série est absolument convergente quel que soit x, ce qui, malheureusement comme nous allons le voir, n'est pas synonyme de rapidement convergente. Nous pouvons estimer le nombre de termes nécessaires à partir du rapport $|u_{k+1}/u_k| = x/(k+1)$. Ce rapport est plus grand que un (les termes de la série sont croissants en valeur absolue) tant que $k+1 < |x|$. Cela implique qu'au bord de l'intervalle choisi, il faudra bien plus de 10 termes pour approcher la fonction exponentielle. En fait, pour $x = 10$, nous avons trouvé $u_9 \simeq -2755,7$ et $u_{10} \simeq 2755,7$. Plus ennuyeux encore, sachant que $e^3 \simeq 20$, nous estimons que $e^{-10} \simeq 5 \times 10^{-5}$. Ceci signifie que pour atteindre une précision relative du millième, le premier terme négligé devra être inférieur à 5×10^{-8}. Le premier terme qui réponde à ce critère est le 39ième ; on trouve alors $e^{-10} \simeq 0,0000454$. Il faut remarquer que chacun des 39 termes doit être calculé avec la même précision absolue de 5×10^{-8}, soit 12 (douze !) chiffres significatifs pour les plus grands d'entre eux, sous peine de perdre toute précision à cause des erreurs d'arrondi pendant l'addition de termes de signes différents. Cet exemple est certes caricatural. En pratique, on s'arrangerait pour réduire l'intervalle de définition de x. Il montre cependant que le calcul à l'aide de développements tronqués doit faire l'objet d'une attention certaine.

Listing 2.3 – Exponentielle par son développement en série

```
eps = 1E-8; kmax = 40;                                    1
terme = 1; somme = 1;k = 1;                               2
x = input("valeur de x: ");                               3
while (k<kmax) & (abs(terme) > eps)                       4
    terme = -x*terme/k;                                   5
    somme = somme + terme;                                6
    [res] = [k,terme,somme];                              7
    write(%io(2),res);                                    8
    k = k+1;                                              9
end                                                       10
somme, exp(-x)                                            11
```

Sans nous laisser décourager par les remarques précédentes, nous avons rédigé un programme de calcul de e^{-x} sous Scilab, reproduit ci-contre. Nous calculons chaque terme à partir du précédent (itération, ligne 5), en évitant soigneusement de former des factorielles ou des puissances de x : ce serait beaucoup plus long et surtout les résultats intermédiaires déborderaient de la capacité de l'ordinateur. Les lignes (7, 8) affichent des résultats intermédiaires.

2.4. Approximant de Padé

Une fonction comme tg x, avec ses asymptotes verticales, a un comportement très éloigné de celui d'un polynôme ; il sera donc malaisé de trouver un développement limité convenable. Au contraire, une fraction rationnelle peut présenter des branches asymptotiques et pourra plus facilement approcher tg x. Si bien que nous aurions intérêt, pour approcher une fonction qui n'a pas un comportement polynômial, à essayer une fraction rationnelle. Si la théorie générale de telles approximations existe, nous ne la décrirons pas ici, mais nous présenterons un cas particulier, connu sous le nom d'approximant de Padé. On peut décrire un approximant de Padé comme une fraction rationnelle qui obéit à des contraintes très semblables à celles d'un développement en série.

Nous considérons une fraction rationnelle $R_{m,n}$, quotient d'un polynôme P_m (le numérateur), de degré m et de coefficients $a_j, 0 \leq j \leq m$ par un polynôme Q_n (le dénominateur), de degré n et de coefficients $b_j, 0 \leq j \leq n$, avec $b_0 \neq 0$. Comme nous pouvons, sans changer la valeur de la fraction, diviser haut et bas par une même constante, nous imposons, sans restreindre la généralité, la condition $b_0 = 1$. La fraction $R_{m,n}$ contient donc $m+n+1$ coefficients à déterminer. $R_{m,n}$ sera, par définition, l'approximant de Padé d'ordre $N+1$ de la fonction $f(x)$, au voisinage du point $x = x_0$, si $R_{m,n}(x_0)$ et ses N premières dérivées coïncident respectivement avec $f(x_0)$ et ses N premières dérivées :

$$R_{m,n}^{(p)}(x_0) \equiv f^{(p)}(x_0), \qquad p = 0, 1, 2 \ldots, N. \tag{2.3}$$

en posant $f^{(0)} = f$. Nous pouvons toujours, à l'aide d'une translation, nous ramener au cas $x_0 = 0$, ce que nous supposerons réalisé dans la suite. Comme nous disposons de $m+n+1$ coefficients inconnus, il nous faut, pour déterminer entièrement la fraction rationnelle, autant de conditions ; il faut donc que $m + n = N$. Il est hors de question de calculer les dérivées successives de R et de f pour les identifier, ce qui serait en général impossible. Nous supposons que la fonction f admet un développement de MacLaurin

$$f(x) = \sum_{j=0}^{\infty} c_j x^j.$$

$R - f$ s'écrit

$$\frac{P_m(x) - Q_n(x) f(x)}{Q_n(x)}.$$

$R - f$ et ses N premières dérivées seront nuls à l'origine si le numérateur de cette expression commence par un terme en x^{N+1}, ce qui revient à dire que les coefficients des termes en $x^0, x^1 \cdots x^N$ sont tous nuls. Avec la convention $b_l = 0$ si $l > n$, nous obtenons les relations

$$\begin{aligned} a_0 &= b_0 c_0 \qquad (x^0), \\ a_1 &= b_0 c_1 + b_1 c_0 \qquad (x^1), \\ a_j &= \sum_{s=0}^{j} c_{j-s} b_s \qquad (x^j, j \leq m). \end{aligned}$$

Lorsque tous les coefficients a_j disponibles ont été utilisés, on continue sans eux

$$0 = \sum_{s=0}^{j} c_{j-s} b_s \qquad (x^j, j = m+1, m+2 \ldots, N).$$

Il est commode d'utiliser d'abord la dernière série d'équations, plus simples, pour déterminer certains des coefficients b_j, puis la première série pour les coefficients restants.

On a constaté empiriquement que le choix $m = n \pm 1$ était souvent meilleur que d'autres. Le raisonnement précédent ne permet pas d'estimer l'erreur d'approximation, mais l'expérience montre que l'approximation selon Padé est très bonne.

Exemple – Cherchons l'approximant de Padé $R_{2,2}$ de e^x au voisinage de l'origine. Ici, $m = n = 2, N = 4$; il nous faut donc connaître, pour commencer, les 5 premiers termes du développement de l'exponentielle. Les coefficients correspondants sont

$$c_0 = 1; \quad c_1 = 1; \quad c_2 = \frac{1}{2}; \quad c_3 = \frac{1}{6}; \quad c_4 = \frac{1}{24}.$$

D'autre part, $R_{2,2}$ s'écrit

$$R_{2,2}(x) = \frac{a_0 + a_1 x + a_2 x^2}{1 + b_1 x + b_2 x^2}.$$

Nous formons maintenant le numérateur de $R - e^x$ et nous identifions à zéro les coefficients de x^0, x^1, x^2, x^3, x^4, ce qui donne

$$a_0 = 1; \qquad a_1 = b_1 + 1; \qquad a_2 = b_1 + b_2 + 1/2;$$
$$0 = b_2 + b_1/2 + 1/6; \qquad 0 = b_1/6 + b_2/2 + 1/24.$$

Nous trouvons, en résolvant ce système :

$$b_1 = -1/2; \qquad b_2 = 1/12; \qquad a_0 = 1; \qquad a_1 = 1/2; \qquad a_2 = 1/12$$

et donc

$$R_{2,2} = \frac{12 + 6x + x^2}{12 - 6x + x^2} = \frac{(x+6)x + 12}{(x-6)x + 12}.$$

Le programme Scilab ci-dessous a produit le tracé présenté figure 2.2.

Listing 2.4 – Approximations de l'exponentielle

```
x = linspace(0,1,100);
p = 1+x+x.^2/2+x.^3/6+x.^4/24;
r = ((x+6).*x+12)./((x-6).*x+12);
xset("window",0),xbasc(0)
xtitle("approximations de exp(x)")
plot2d(x',[(p-exp(x))',(r-exp(x))']);
```

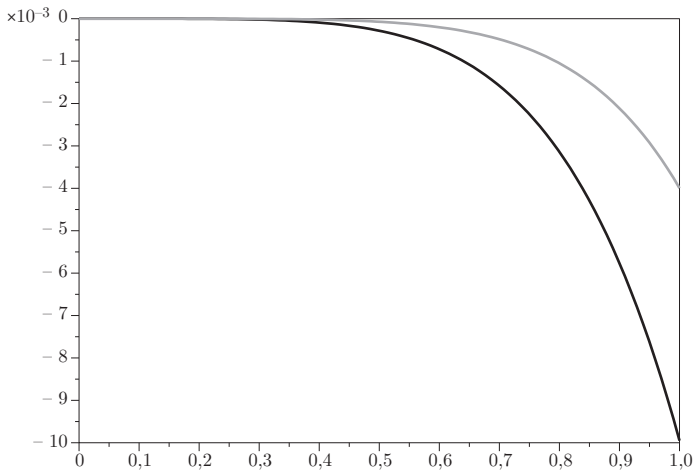


Figure 2.2 – Erreurs d'approximation de e^x par son développement limité (en noir) et par l'approximant de Padé $R_{2,2}$ (en gris).

Vous constatez que pour $x \simeq 1$ l'approximant de Padé est presque trois fois plus précis que le développement de Taylor.

2.5. Utilisation de bibliothèques de programmes

Les utilisateurs de logiciels de « haut niveau » comme Scilab ont accès à de nombreuses fonctions prédéfinies, comme les fonctions trigonométriques ou hyperboliques directes et inverses, la fonction d'erreur, les fonctions de Bessel. Les logiciels commerciaux (Maple, Mathematica) sont encore plus riches. Ceux qui écrivent leurs propres programmes ne sont pas tenus de coder le calcul des fonctions spéciales : ils peuvent importer les sous-programmes nécessaires à partir d'une bibliothèque. Il existe de très nombreuses collections de programmes scientifiques, recensées dans le « Guide to available mathematical software » ou « GAMS ». Nous proposons ici un tout petit exemple d'utilisation de la bibliothèque « Gnu Scientific Library » (ou « GSL ») écrite en C et compatible avec le C++. Ce programme définit et fait quelques calculs élémentaires sur des nombres complexes.

Les lignes 2 à 4 provoquent l'inclusion des en-têtes des constantes et des fonctions nécessaires (celles-ci sont groupées par rubriques à l'intérieur de l'ensemble `gsl`).

La GSL contient en particulier des programmes de calcul des fonctions spéciales, de génération de nombres aléatoires, de résolution d'équations différentielles et d'algèbre linéaire.

2.6. Approximation de fonctions

Nous avons présenté, dans les paragraphes précédents, un certain nombre de recettes ou de procédés destinés à fournir des approximations de fonctions, sans aucunement

Listing 2.5 – Manipulation de nombres complexes

```
#include <iostream>
#include <gsl/gsl_math.h>
#include <gsl/gsl_complex.h>
#include <gsl/gsl_complex_math.h>
using namespace std;
int main(void){
    gsl_complex x,y,z,a,b,c;
    x = gsl_complex_rect(1,0); y = gsl_complex_rect(0,1);
    z = gsl_complex_add(x,y);
    cout << GSL_REAL(z) << '\t' << GSL_IMAG(z)<< endl;
    a = gsl_complex_rect(M_E,0);
    b = gsl_complex_rect(0,M_PI/4);
    c = gsl_complex_pow(a,b);
    cout << GSL_REAL(c) << '\t' << GSL_IMAG(c)<< endl;
    c = gsl_complex_mul_real(c,sqrt(2));
    cout << gsl_complex_abs(c);
}
```

nous soucier de rigueur mathématique. En réalité, l'approximation des fonctions est un domaine bien établi des mathématiques, qui a connu un grand développement à partir de la fin du 19ème siècle. Dans les lignes qui suivent, nous donnons quelques idées générales sur l'approximation, considérée d'un point de vue plus mathématique. Le problème se pose en ces termes. Nous nous intéressons à une fonction $f(x)$ de la variable réelle x et nous désirons connaître ses valeurs en tous points d'un certain intervalle. Seulement voilà, f est compliquée et longue à calculer et nous n'avons pas le temps d'accumuler toutes les valeurs nécessaires de f. Nous décidons alors de remplacer f par une approximation f^* qui se calcule facilement. Pour savoir si cette substitution peut avoir un sens, nous devons répondre à plusieurs questions : dans quelle catégorie de fonctions allons nous choisir f^* ? Selon quel critère allons nous décider que l'approximation est « bonne » ou « mauvaise » ? Sur quel intervalle les propriétés précédentes doivent-elles être vérifiées ?

Nous nous limiterons ici au cas de l'approximation polynômiale, c'est-à-dire que f^* sera un polynôme de degré au plus égal à n (une combinaison linéaire de $x^k, k = 0, 1, 2, \ldots, n$). Le théorème suivant, dû à Weierstrass, affirme l'existence de f^* sous des conditions assez générales.

Théorème – Si f est continue sur l'intervalle fini $I = [a, b]$ et si ϵ est un nombre strictement positif donné, alors il existe un polynôme f^* tel que

$$\sup_{x \in I} |f(x) - f^*(x)| < \epsilon.$$

Remarquez que le théorème ne donne aucune indication sur la construction de f^*. Ici, la « distance » entre f et f^* est la « norme infinie » :

$$\| f - f^* \| = \sup_{x \in I} |f(x) - f^*(x)|.$$

On pourrait imaginer d'autres types d'approximation où la définition de la « distance » serait différente ; on pourrait penser par exemple à l'écart quadratique moyen, qui est défini comme $\| f - f^* \|^2 = \int_I |f - f^*|^2 dx$.

Tschebychef et de la Vallée–Poussin ont cherché à construire ce que l'on appelle le polynôme d'approximation « minimax » qui minimise la distance maximale (au sens de la norme infinie) entre f et f^*. Pour décrire qualitativement cette théorie, disons qu'un « bon » polynôme f^* est tel que l'erreur d'approximation, $f - f^*$, présente une série d'extremums assez régulièrement répartis sur I, d'amplitudes identiques et alternativement positifs et négatifs. On dispose, pour trouver explicitement f^*, d'un algorithme puissant d'approximations successives (second algorithme de Remes). Ainsi, le polynôme minimax du troisième degré qui approche e^x sur $[-1, 1]$ s'écrit :

$$p_3^*(x) = 0,994579 + 0,995668x + 0,542973x^2 + 0,179533x^3.$$

L'erreur $e^x - p_3^*(x)$ est représentée sur la figure 2.3.

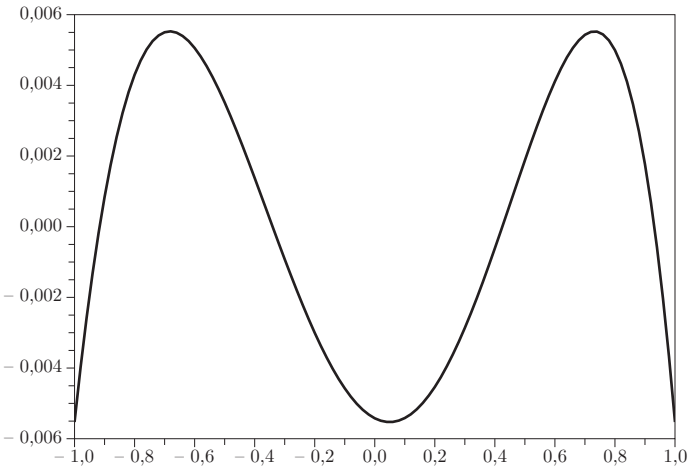


Figure 2.3 – Erreur d'approximation de e^x par le polynôme minimax cubique.

Les compilateurs utilisent ce genre d'approximation pour évaluer toutes les fonctions.

2.7. Développement asymptotique

Lorsque la variable indépendante devient très grande, il devient évidemment désespéré d'approcher une fonction par un développement construit au voisinage de 0 ; on peut bien sûr faire une translation d'origine pour chaque valeur particulière, mais cela complique notablement les calculs. Dans certains cas, on peut avoir recours à une

approximation valable lorsque $1/x$ est petit, ce que l'on appelle un développement asymptotique et dont voici la définition.

Soit la série infinie

$$c_0 + \frac{c_1}{x} + \frac{c_2}{x^2} + \cdots$$

Notons S_n la somme partielle des n premiers termes

$$S_n = c_0 + \frac{c_1}{x} + \frac{c_2}{x^2} + \cdots + \frac{c_{n-1}}{x^{n-1}}.$$

Habituellement, on s'intéresse à la convergence d'une série ou à la limite de ses sommes partielles S_n quand n tend vers l'infini. Ici, nous procédons différemment ; gardant n fixe, nous faisons croître x au delà de toute limite. Nous supposons qu'il existe une fonction f telle que la différence $|f(x) - S_n(x)|$ tende vers zéro plus vite que $\frac{1}{x^{n-1}}$ quand $x \longrightarrow \infty$. En d'autres termes

$$\lim_{x \longrightarrow \infty} x^{n-1} \, |f(x) - S_n(x)| = 0.$$

Si ces conditions sont remplies, nous disons que la série est un développement asymptotique de la fonction f.

Exemple – Considérons la fonction définie pour $x > 0$ par l'intégrale

$$f(x) = \int_x^\infty \frac{1}{t} e^{x-t} dt$$

Des intégrations par partie répétées nous permettent de transformer cette expression en

$$f(x) = \frac{1}{x} - \frac{1}{x^2} + \frac{2}{x^3} + \cdots + \frac{(-1)^{n-1}(n-1)!}{x^n} + (-1)^n n! \int_x^\infty \frac{e^{x-t}}{t^{n+1}} dt.$$

Ceci nous laisse supposer que la série

$$\frac{1}{x} - \frac{1}{x^2} + \frac{2!}{x^3} - \frac{4!}{x^4} \cdots$$

pourrait représenter le développement asymptotique de f. Pour le prouver, formons

$$f(x) - S_{n+1}(x) = (-1)^n n! \int_x^\infty \frac{e^{x-t}}{t^{n+1}} dt.$$

Dans l'intégrale, l'exponentielle est comprise entre 0 et 1, d'où la majoration

$$|f(x) - S_{n+1}(x)| < n! \int_x^\infty \frac{1}{t^{n+1}} dt = (n-1)! \frac{1}{x^n},$$

une quantité qui tend évidemment vers zéro lorsque x croît, à n constant. On écrit souvent

$$\int_x^\infty \frac{1}{t} e^{x-t} dt \sim \frac{1}{x} - \frac{1}{x^2} + \frac{2!}{x^3} - \frac{4!}{x^4} \cdots .$$

2.8. Représentation des nombres en machine

L'arithmétique des ordinateurs présente des propriétés plus compliquées qu'il n'y paraît à première vue. Ces complications sont dues à ce qu'un ordinateur (tout comme un humain) ne manipule qu'un nombre limité de chiffres ; il s'en suit que beaucoup de nombres concevables ne sont pas ou sont mal représentés en machine. c'est le cas par exemple pour 1/10 en base 2. Les choses se présentent différemment pour les entiers et pour les nombres fractionnaires. Commençons par examiner le cas des entiers.

2.8.1. Les nombres entiers

Nous avons choisi, pour fixer les idées, le compilateur FreePascal, mais des considérations presque identiques (au vocabulaire près) s'appliquent à tous les compilateurs. FreePascal définit dix types d'entiers, dont nous reproduisons ci-dessous les caractéristiques.

types entiers	représentation (nombre d'octets)	intervalle de définition
`byte`	1	$0 : 255$
`shortint`	1	$-128 : 127$
`smallint`	2	$-32768 : 32767$
`word`	2	$0 : 65535$
`integer`	2, 4 ou 8	
`cardinal`	2, 4 ou 8	
`longint`	4	$-2147483648 : 2147483647$
`longword`	4	$0 : 4294967295$
`int64`	8	$-9223372036854775808 : 9223372036854775807$
`qword`	8	$0 : 18446744073709551615$

Les types `byte, word, cardinal` et `qword` ne comportent pas de signe. La taille des types `integer` et `cardinal` dépend de la machine.

Le chiffre binaire le plus à gauche (le plus significatif, de poids le plus grand) indique le signe : 0 pour les nombres positifs, 1 pour les entiers négatifs. Dans le type « shortint » par exemple, le plus grand nombre positif s'écrit 01111111 ; il vaut $2^6+2^5+2^4+2^3+2^2+2^1+2^0 = 127$. Les nombres négatifs sont représentés selon la méthode dite « du complément à deux », qui prescrit que, si x est négatif, (et plus petit en valeur absolue que 2^7) sa représentation, que nous notons (x), est $x+2^8$. La recette suivante permet de construire facilement $(-x)$ à partir de (x) : changer tous les « 1 » en « 0 » et réciproquement, sauf pour le chiffre le plus à droite. Si $x = 23, (x) = 00010111$ et $(-23) = 11101001$. Vous pouvez vérifier, en appliquant les règles de l'arithmétique binaire, que $(23)+(-23) = 100000000$; la retenue (1) sera perdue, puisque les nombres n'ont que 8 chiffres.

L'ordinateur ne prévient pas lorsque le résultat d'une addition entre entiers est trop grand. On a par exemple $(127) + (3) = 01111111 + 00000011 = 10000010 = (-124)$. Cette représentation fonctionne un peu comme un compteur kilométrique de voiture qui, à 99999 km, repasse par zéro. La conséquence pratique est que l'amplitude des entiers est limitée. Un piège classique dans lequel chacun se fait prendre au moins une fois est le calcul de la factorielle. Avec le type `smallint`, $7!$ est calculé correctement, $8!$ est trouvé négatif. Le problème ne se pose pas avec Scilab qui ne connaît (sauf déclaration spéciale) que des nombres fractionnaires, ni avec Maple, qui peut manipuler autant de chiffres que la mémoire de l'ordinateur le permet.

2.8.2. Les nombres fractionnaires

Les nombres décimaux (on dit aussi fractionnaires, flottants, à virgule flottante) sont représentés par une partie fractionnaire (parfois appelée mantisse) et un exposant, comme par exemple $23 \longrightarrow 0{,}23\text{E}2$ (notation « scientifique »). Le nombre précédent peut aussi bien s'écrire 23E0 ou 0,023E3. Pour éviter ces ambiguïtés et rendre maximal le nombre de chiffres significatifs, on invoque une convention de normalisation, qui est en général équivalente à placer la virgule juste après le premier chiffre non nul (2,3E1). En binaire, il n'existe qu'un chiffre non nul (1) et celui-ci peut donc être sous-entendu (implicite). On atteint ainsi une résolution de $n+1$ chiffres pour un encombrement de n chiffres.

Les fabricants d'ordinateurs et de logiciels s'efforcent de respecter la norme édictée par l'Institute of Electrical and Electronics Engineers, IEEE 754. La conformité à cette norme est assez variable. La norme définit deux types de nombres flottants : `single` codé sur 32 bits et `double` qui comportent 64 chiffres binaires. La répartition de ces chiffres est la suivante

	signe	exposant	p. fractionnaire
`single`	1	8	23
`double`	1	11	52

Il existe une borne inférieure et une borne supérieure pour tout nombre représenté en machine et utilisant tous les chiffres significatifs prévus par son type. Cet intervalle est donné dans le tableau suivant.

type	nb équivalent de chiffres décimaux	intervalle de définition pour un nombre positif
`single`	$7-8$	$1{,}4\times10^{-45}$: $3{,}4\times10^{38}$
`double`	$15-16$	5×10^{-324} : 2×10^{308}

Il est clair que l'intervalle de définition (ou le nombre de chiffres significatifs) dépend du nombre de chiffres binaires affectés respectivement à la représentation de la partie fractionnaire et de l'exposant. L'intervalle de définition des nombres négatifs est obtenu en prenant l'opposé des valeurs ci-dessus. Sous Scilab, les nombres fractionnaires sont uniformément du type double.

Zéro est représenté conventionnellement par un nombre à exposant et partie fractionnaire nuls. Les nombres compris entre 0 et la borne inférieure de l'intervalle de définition (exposant nul, partie fractionnaire non nulle) sont dits « dénormalisés » ; ils comportent donc moins de chiffres significatifs. Ceux plus grands que la borne supérieure sont représentés par l'infini (partie fractionnaire nulle, exposant ne comportant que des 1). La norme prévoit comment chaque opération arithmétique doit traiter ces nombres.

Même si un nombre est de grandeur correcte, son calcul est souvent entaché d'une erreur d'arrondi. Supposons que nous utilisons un ordinateur fictif fonctionnant en numération décimale. Essayons de calculer 27/13,1 = 2,06106870229..., un nombre fractionnaire dont la représentation décimale ne se termine jamais et que l'unité centrale calcule avec 20 chiffres significatifs. Si nous avons déclaré que la variable correspondante était du type `single`, lorsque l'ordinateur rangera cette valeur en mémoire, il le fera selon la norme du type, soit avec 7 chiffres. Sur certaines machines on conserve la valeur 2,061068 (troncation), sur d'autres, on retient 2,061069 (arrondi). L'erreur est ici minime (inférieure à 10^{-7}en valeur absolue), mais il faut être conscient qu'elle peut se répéter des milliards de fois au cours d'un calcul. Heureusement, le signe de chaque erreur est pratiquement aléatoire, si bien que celles-ci ne s'ajoutent pas de façon cohérente.

L'erreur relative devient importante lorsque l'on forme la différence de deux nombres voisins. On ne remarque rien en général, sauf si, par hasard, ce premier résultat est immédiatement multiplié par un nombre très grand.

Comment faire des calculs plus précis ? Cela dépend du langage, du compilateur, du système d'exploitation et de la machine ! Par exemple, le compilateur Fortran IBM XL propose le type `REAL(16)` qui code les nombres fractionnaires sur 16 octets (34 chiffres décimaux) et le compilateur Gnu C++ dispose du type `long double` sur dix octets (19 chiffres décimaux). Pour aller plus loin, on peut utiliser des langages spéciaux comme « Pari » ou encore maîtriser les erreurs d'arrondi en recourant au « calcul par intervalle » (voir les références).

2.9. POUR EN SAVOIR PLUS

- FAQ du forum `sci.math.num-analysis` :
  `http://www.mathcom.com/corpdir/techinfo.mdir/index.html`
- GAMS : `http://gams.nist.gov`
- Bibliothèque scientifique GNU : `http://www.gnu.org/software/gsl`
- Norme IEEE 754 :
  - `http://www.rmn.uhp-nancy.fr/Grandclaude/archi2.pdf`

  - `http://www.irisa.fr/sage/jocelyne/cours/precision/insa-1200.pdf`

- histoire des mathématiques, définitions :
  `http://www.sciences-en-ligne.com`, voir dictionnaire scientifique interactif.
- calcul de fonctions et approximations :
  - R. Théodor : *Initiation à l'analyse numérique*, ch. 4 (Masson, Paris, 1994).
  - W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vettering : *Numerical Recipes, the art of scientific computing* (Cambridge University Press, Cambridge, 2007).
  - M. Abramowitz, I. Stegun : *Handbook of mathematical functions* (Dover, New York, 1965).
  - `http://perso.ens-lyon.fr/jean-michel.muller/french-index.html`
  - J.-M. Muller : *Elementary functions, algorithms and implementation* (Birkhauser, Boston, 1997).
  - J.-M. Muller : *Calcul et arithmétique des ordinateurs* (Hermès, traité IC2, Paris, 2004).
  - `http://www.math.technion.ac.il/hat`
- logiciels de calcul de précision arbitraire :
  `http://pari.math.u-bordeaux.fr`
  `http://www.mpfr.org`
- logiciel polyvalent, comprend PARI et Matplotlib :
  `http://sage.math.washington.edu/sage`
- Arithmétique par intervalle :
  `http://www.cs.utep.edu/interval-comp/main.html`
- Erreurs et bogues catastrophiques :
  `http://www5.in.tum.de/~huckle/bugse.html`

2.10. Exercices

Exercice 1

a) Calculer la valeur numérique du polynôme $p(x) = 2x^3 - 3x + 1$ pour $x = 2$ puis pour $x = 1$ en utilisant le schéma de Horner.

b) Même question pour $p(x) = 7x^4 + 5x^3 - 2x^2 + 8$ en $x = 0,5$.

Exercice 2

Les polynômes de Hermite $H_n(x)$ obéissent à la relation de récurrence

$$H_0(x) = 1; \quad H_1(x) = 2x; \quad H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x).$$

a) Former les polynômes H_3 et H_4.

b) Calculer numériquement $H_5(0,5)$ sans construire ce polynôme.

c) Représenter graphiquement les 6 premiers polynômes.

Exercice 3

La fonction $\arctan x$ admet le développement en série

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots$$

a) Soit s_n la somme partielle de la série tronquée après le n-ième terme. Quelle erreur de troncation commet-on si l'on remplace $\arctan x$ par s_n ?

b) On se propose de calculer π à l'aide de la relation $\pi/4 = \arctan 1$. Combien faudra-t-il de termes de la série pour que l'erreur sur π soit inférieure à 10^{-5} ?

c) La série ne converge pas lorsque $|x| > 1$; utiliser une relation entre $\arctan x$ et $\arctan(1/x)$ pour disposer d'un argument toujours inférieur à 1. Calculer $\arctan 2$.

Exercice 4

La fonction de Bessel d'ordre zéro, $J_0(x)$, intervient souvent en physique. Elle est définie par son développement en série

$$J_0(x) = \sum_0^\infty (-1)^k \left(\frac{x^k}{2^k k!}\right)^2.$$

a) Trouver, en fonction de x supposé entier, le rang du terme le plus grand en valeur absolue.

b) Écrire un programme pour calculer $J_0(x)$ avec 6 chiffres significatifs, pour $0 < x \leq 20$; quel critère d'arrêt allez-vous utiliser ? Comparez vos résultats aux valeurs exactes

$$J_0(5) = -0,1775967713; \qquad J_0(10) = -0,2459357644; \qquad J_0(15) = -0,0142244728$$

Exercice 5

Les fonctions d'Airy sont utilisées dans la théorie de la diffraction. Elles sont définies de façon conventionnelle à partir de développements en série. On pose

$$\begin{aligned} f(x) &= 1 + \frac{1}{3!}x^3 + \frac{1\cdot 4}{6!}x^6 + \frac{1\cdot 4\cdot 7}{9!}x^9 + \ldots \\ &= \sum_0^\infty 3^k \left(\frac{1}{3}\right)_k \frac{x^{3k}}{(3k)!}; \\ g(x) &= x + \frac{2}{4!}x^4 + \frac{2\cdot 5}{7!}x^7 + \frac{2\cdot 5\cdot 8}{10!}x^{10} + \ldots \\ &= \sum_0^\infty 3^k \left(\frac{2}{3}\right)_k \frac{x^{3k+1}}{(3k+1)!}. \end{aligned}$$

où $$(x)_n = x(x+1)(x+2)\cdots(x+n-1).$$

On a alors

$$Ai(x) = c_1 f(x) - c_2 g(x) \quad ; \quad Bi(x) = \sqrt{3}[c_1 f(x) + c_2 g(x)],$$

avec $c_1 = 0,3550281$ et $c_2 = 0,2588194$.

Programmer le calcul de Ai et Bi ; on assurera une précision absolue de 10^{-5} sur l'intervalle $[-10 \ldots +2]$. Représenter graphiquement ces fonctions sur le même intervalle.

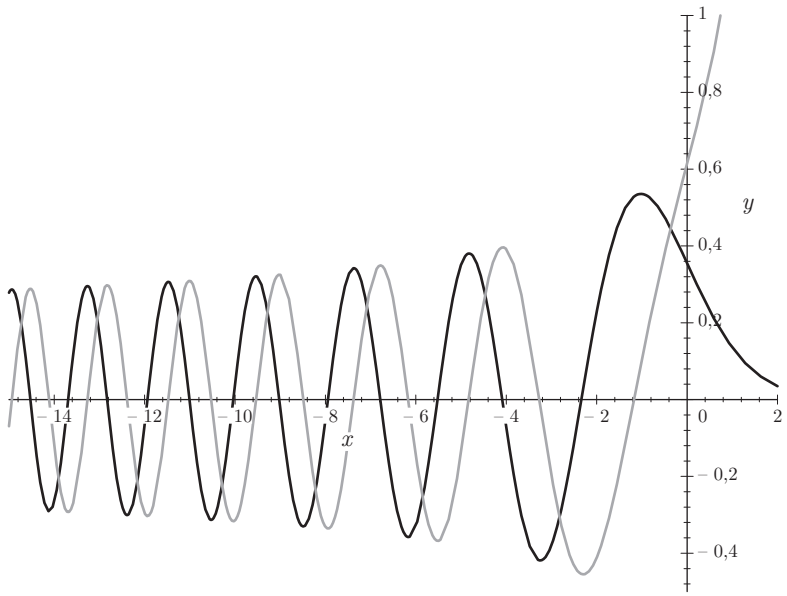


Figure 2.4 – Les fonctions d'Airy, $Ai(x)$ (en noir) et $Bi(x)$ (en gris).

Le résultat vous est montré figure 2.4. Le graphe de la fonction $Bi(x)$, qui présente une branche infinie, a été tronqué.

Exercice 6

Écrire un programme destiné à calculer $\cos x$ et $\sin x$ à partir de leur développement en série tronqué. Le programme comportera les étapes suivantes.

a) Utilisation de la périodicité, de la symétrie et des relations entre sinus et cosinus d'arcs complémentaires pour ramener l'argument dans l'intervalle $-\pi/4 \leq x \leq \pi/4$. On peut écrire $x = k(\pi/2) + r$, puis $n = k \bmod 4$ et enfin exprimer $\sin x$ et $\cos x$ en fonction de $\sin r$ et $\cos r$ pour chaque valeur de n.

b) Calcul par récurrence des termes de ces séries, pour atteindre une erreur absolue donnée à l'avance (par exemple 10^{-6}).

c) Affichage du résultat.

Exercice 7

L'objet mathématique représenté ci-dessous est une « fraction continue » :

$$f = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{a_4}{b_4 + \cdots}}}}$$

Pour simplifier le travail des imprimeurs, on l'écrit plutôt

$$f = b_0 + \frac{a_1}{b_1+}\frac{a_2}{b_2+}\frac{a_3}{b_3+}\frac{a_4}{b_4+}\cdots$$

Une expression de ce genre n'a d'intérêt que si les a_i et les b_i sont des constantes ou des fonctions simples de x, comme dans la représentation de la fonction tangente :

$$\operatorname{tg} x \simeq \frac{x}{1-}\frac{x^2}{3-}\frac{x^2}{5-}\frac{x^2}{7-}\cdots$$

En tronquant la fraction continue précédente après le terme $x^2/9$ et en réduisant au même dénominateur, on obtient

$$\operatorname{tg} x \simeq \frac{x}{15}\frac{945 - 105x^2 + x^4}{63 - 28x^2 + x^3}.$$

Quels sont les premiers zéros de numérateur ? Et ceux du dénominateur ? Comparez vos résultats aux valeurs exactes.

Exercice 8

a) Construire le polynôme p qui représente le développement limité à l'ordre 7 de la fonction $f(x) = \operatorname{argth} x$ au voisinage de l'origine. Cette fonction admet aussi la représentation en fraction continue

$$g = \frac{x}{1-}\frac{x^2}{3-}\frac{4x^2}{5-}\frac{9x^2}{7-}\cdots$$

b) Représenter sur un même graphique, les variations de f, p et g pour $|x| < 0,9$.

c) Vous avez sans doute procédé de façon « artisanale », en chassant les dénominateurs, pour calculer g. L'algorithme de Wallis, décrit ci-dessous, permet un calcul systématique. Soit

$$f(x) = b_0 + \frac{a_1}{b_1+}\frac{a_2}{b_2+}\frac{a_3}{b_3+}\cdots$$

la fraction continue générale et soit f_n la valeur de $f(x)$ calculée en incluant tous les termes jusqu'à a_n et b_n compris. Si on pose

$$f_n = \frac{A_n}{B_n},$$

on montre que les quantités A_n et B_n peuvent s'obtenir par les relations de récurrence suivantes

$$A_{-1} = 1; \qquad B_{-1} = 0; \qquad A_0 = b_0; \qquad B_0 = 1;$$
$$A_j = b_j A_{j-1} + a_j A_{j-2}; \quad B_j = b_j B_{j-1} + a_j B_{j-2}; \quad j = 1, 2, \ldots, n.$$

On interrompt le calcul lorsque deux valeurs successives, f_n et f_{n+1}, sont « suffisamment » proches l'une de l'autre. Écrire un programme pour calculer $\operatorname{argth} x$.

Exercice 9

On se propose de former l'approximant de Padé $R_{3,4}$ de la fonction $\operatorname{th} x$.

a) Construire le développement de MacLaurin de cette fonction, à l'ordre 8 en x ; il est commode d'utiliser la relation $(\operatorname{th} x)' = 1 - \operatorname{th}^2 x$.

b) En déduire l'approximant de Padé. On utilisera les propriétés de symétrie de la fonction pour éviter de calculer des termes inutiles.

c) Représenter graphiquement la fonction et son approximation.

Exercice 10

La fonction « exponentielle intégrale » est définie comme

$$Ei(x) = \int_x^\infty \frac{e^{-t}}{t} dt.$$

En intégrant par partie, on trouve successivement

$$Ei(x) = -\left.\frac{e^{-t}}{t}\right|_x^\infty - \int_x^\infty \frac{e^{-t}}{t^2} dt = \frac{e^{-x}}{x} + \left.\frac{e^{-t}}{t}\right|_x^\infty + 2\int_x^\infty \frac{e^{-t}}{t^3} dt.$$

À l'ordre n, il vient

$$Ei(x) = \frac{e^{-x}}{x}\left[1 - \frac{1}{x} + \frac{2!}{x^2} + \cdots + (-1)^n \frac{n!}{x^n}\right] + (-1)^{n+1}(n+1)! \int_x^\infty \frac{e^{-t}}{t^{n+2}} dt,$$

ce qui constitue le développement asymptotique de $Ei(x)$, sous une forme un peu plus générale qu'au § 2.7.

a) On appelle $S(x)$ la série qui figure entre crochets ; est-elle convergente ?

b) Majorer l'intégrale pour démontrer que

$$\lim_{x\to\infty} \left[x^n \left(xe^x Ei(x) - S_n\right)\right] = 0.$$

Ceci signifie que S_n, somme partielle de la série S, est une approximation de $xe^x Ei(x)$ et que cette approximation d'autant meilleure que x est plus grand, n étant donné.

c) Écrire un programme pour calculer et représenter graphiquement les approximations de $Ei(x)$ pour n variant de 2 à 20 ; on sait, par exemple, que $Ei(10) \simeq 0,4157\,10^{-5}$.

Exercice 11

Tout nombre réel compris entre zéro et un peut être représenté par une fraction continue de la forme suivante :

$$x = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

où la suite $a_1, a_2, a_3, \ldots$ constitue une représentation très précise de x. L'algorithme suivant permet de trouver les a_i ($[E(y)]$ désigne la partie entière de y) :

```
i := 1
répéter :
  y := 1/x ;
  ai := E(y);
  x := y - E(y) ;
  i = i + 1
jusqu'à fini.
```

Écrire un programme pour réaliser cet algorithme. Pour $\pi - 3 = 0,141592$, on trouve $\{7, 15, 1, 292, 1, 1, \ldots\}$. En chassant tous les dénominateurs, former les fractions qui représentent π et les calculer numériquement. Remarque : l'approximation $\pi \simeq 22/7$ était connue d'Archimède et $355/113$ s'appelle la fraction réduite d'Adrien Metius.

Exercice 12

Un nombre décimal positif peut être représenté approximativement par un quotient de deux entiers. L'algorithme suivant permet de déterminer systématiquement le numérateur et le dénominateur de cette fraction. Soit $x > 0$ le nombre décimal et $a_0/b_0, a_1/b_1$ deux fractions simples qui l'encadrent (les a_i, b_i sont toujours positifs). On forme

$$f = \frac{a_0 + a_1}{b_0 + b_1}.$$

Montrer que f est comprise entre a_0/b_0 et a_1/b_1. Si $f < x$, f devient l'approximation de x par défaut et on pose $f = a_2/b_2$; si $f > x$, f devient l'approximation par excès, et on pose $f = a_3/b_3$. On calcule alors une nouvelle valeur de f selon la formule précédente et on itère l'algorithme jusqu'à convergence. Il est commode de choisir $a_0/b_0 \equiv 0/1$ et $a_1/b_1 \equiv 1/0$. Comme cette dernière fraction n'est pas définie, il faut chasser les dénominateurs avant de comparer xb_1 et a_1.

Exemple – Pour $x = 3,14159\ldots$, on a la suite

$$\frac{0}{1} < \pi < \frac{1}{0}; \frac{1}{1} < \pi < \frac{1}{0}; \frac{2}{1} < \pi < \frac{1}{0}; \frac{3}{1} < \pi < \frac{1}{0}; \frac{3}{1} < \pi < \frac{4}{1};$$
$$\frac{3}{1} < \pi < \frac{7}{2}; \frac{3}{1} < \pi < \frac{10}{3};$$

Écrire un programme correspondant à cet algorithme.

Exercice 13

On dispose d'une machine où les nombres réels sont représentés (sous forme décimale) par une partie fractionnaire à trois chiffres, comprise entre 0 et 1, et un exposant à un chiffre : $42,3 \longrightarrow 0,423\ E2$ ou encore $0,00123 \longrightarrow 0,123\ E-2$. Avant d'effectuer une opération arithmétique, la machine multiplie ou divise par une puissance de 10 l'une des parties fractionnaires de telle façon que les deux opérandes présentent le même exposant. Prévoir le résultat des opérations suivantes, en distinguant le cas où le résultat est tronqué de celui où il est arrondi.

$$12,3 + 0,0234; \quad -0,0321 + 0,000136; \quad -321 + 32,1; \quad 132 \times 0.987\,;$$
$$-2,14/0,000137; \quad (-0,111 + 0,222) \times 0,00111/999.$$

2.11. Projets

Projet 1. Fonction de Riemann

La fonction ζ de Riemann est définie par la formule

$$\zeta(s) = 1 + \frac{1}{2^s} + \frac{1}{3^s} + \cdots + \frac{1}{n^s} + \cdots$$

où la variable s est complexe. Cette fonction joue un rôle important en mathématique, en particulier dans la théorie des nombres premiers. On demande de calculer numériquement $\zeta(x+iy)$, pour $0 \leq x \leq 3$ et $10 \leq y \leq 30$ et de représenter les résultats par une surface $|\zeta(x,y)|$. On peut faire cela en une unique instruction en Maple :

```
plot3d(abs(Zeta(x+I*y),x=0..3,y=10..30)
```

Il est plus intéressant de programmer soi-même le calcul. La manipulation de nombres complexes est banale sous Scilab. Cependant, vous vous rendrez compte rapidement que, malgré la simplicité de la définition, un calcul direct est très long et bien peu précis. Pour obtenir en un temps raisonnable une surface semblable à celle de la figure 2.5, il faut faire appel à l'un des algorithmes élégants décrits par X. Gourdon et P. Sebah (voir le site `http://numbers.computation.free.fr`).

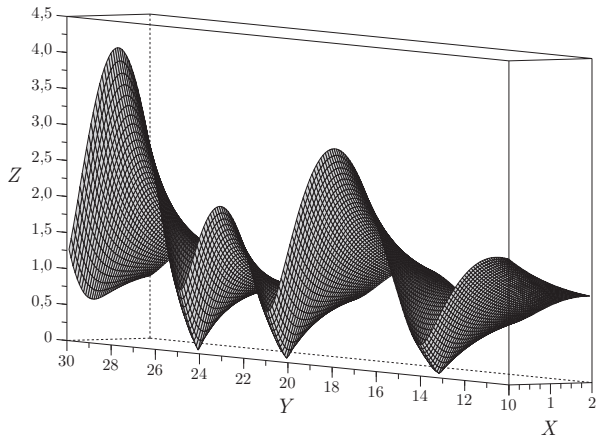


Figure 2.5 – Une vue de la fonction $|\zeta(s)|$.

Vous pourrez alors tenter de vérifier l'hypothèse de Riemann : tous les zéros de ζ se trouvent sur la droite verticale d'équation $x = 0,5$. Cette conjecture a été vérifiée numériquement pour quelques milliards de zéros, mais n'est pas démontrée.

Projet 2. Diffraction de Fresnel

Dans la figure 2.6, S est une source ponctuelle monochromatique, C est un écran percé d'une ouverture circulaire de rayon r et E est un écran d'observation.

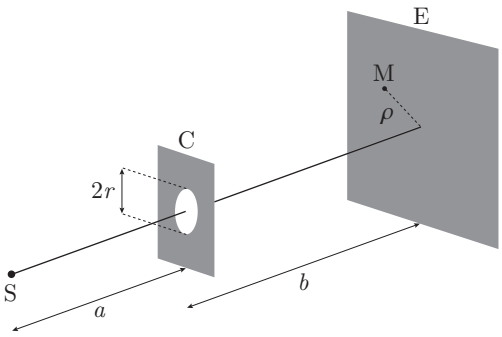


Figure 2.6 – Définition des variables pour le diffraction de Fresnel.

On désigne par ρ la distance d'un point courant de E (M) à l'axe, par a la distance de S à C, par b la distance de C à E et par I_0 l'éclairement de l'écran en l'absence d'obstacle. On pose

$$y = \frac{2\pi}{\lambda}\frac{a+b}{ab}; \quad z = \frac{2\pi}{\lambda}\frac{\rho r}{b}$$

L'intensité diffractée par cette ouverture circulaire, dans les conditions de l'approximation de Fresnel, s'écrit :

$$\begin{aligned} I_1(y,z)/I_0 &= U_1^2(y,z) + U_2^2(y,z) \\ &= \left[V_0(y,z) - \cos\left(\frac{y^2+z^2}{2y}\right)\right]^2 + \left[V_1(y,z) - \sin\left(\frac{y^2+z^2}{2y}\right)\right]^2. \end{aligned}$$

Les fonctions de Lommel U et V sont des séries de fonctions de Bessel :

$$U_i = \sum_{j=0}^{\infty}(-1)^j \left(\frac{y}{z}\right)^{i+2j} J_{i+2j}(z) \quad ; \quad V_i = \sum_{j=0}^{\infty}(-1)^j \left(\frac{z}{y}\right)^{i+2j} J_{i+2j}(z).$$

On demande de tracer la courbe représentant l'éclairement du plan E en fonction de la distance à l'axe ρ. Les quantités U_i, V_i étant des sommes infinies, il faudra choisir un critère de convergence qui permette d'interrompre le calcul à partir d'un certain rang. Cependant, il sera nécessaire d'utiliser des fonctions de Bessel d'indice au moins égal à 50. Ceci ne pose pas de difficulté avec un logiciel comme Scilab. Pour programmer vous-mêmes le calcul de J pour de grandes valeurs de l'indice ou de l'argument, reportez-vous par exemple au livre de Press et coll. En gros, il faut

calculer J_0 et J_1 à partir de leurs développements en série, $J_n(x), \quad n < x$, par récurrence et $J_n(x), \quad n > x$, à l'aide de son développement asymptotique.

Appliquer le théorème de Babinet pour montrer que l'intensité diffractée par un disque opaque vaut

$$\frac{I_2(y,z)}{I_0} = V_0^2(y,z) + V_1^2(y,z)$$

et calculer cette intensité. Vous devez trouver un résultat proche de la figure 2.7.

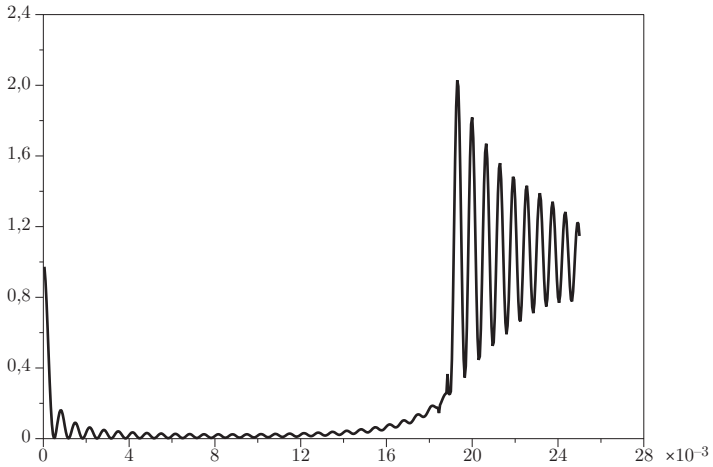


Figure 2.7 – Diffraction de Fresnel par un disque. Remarquer la « tache de Poisson » au centre de l'ombre géométrique.

Projet 3. Champ magnétique d'une boucle de courant

Une spire de rayon a, centrée à l'origine dans le plan xOy (voir fig. 2.8), parcourue par un courant I, crée, au point M de coordonnées cylindriques r, z, une induction donnée par les formules suivantes

$$\begin{cases} B_z &= \dfrac{B_0}{\pi\sqrt{Q}}\left[E(k)\dfrac{1-\rho^2-\zeta^2}{(1-\rho)^2+\zeta^2} + K(k)\right], \\ B_r &= \dfrac{B_0}{\pi\sqrt{Q}}\dfrac{z}{r}\left[E(k)\dfrac{1+\rho^2+\zeta^2}{(1-\rho)^2+\zeta^2} - K(k)\right] \end{cases}$$

avec :

$$\rho = r/A, \quad \zeta = z/a, \quad Q = (1+\rho)^2+\zeta^2, \quad k = \sqrt{\frac{4\rho}{Q}}, \quad B_0 = \frac{\mu_0 I}{2a}.$$

a) Le calcul des « intégrales elliptiques complètes » $E(k), K(k)$ se fait commodément par l'algorithme dit de la moyenne arithmético-géométrique de Gauss. Étant donnés trois nombres a_0, b_0, c_0, on applique les relations de récurrence suivantes

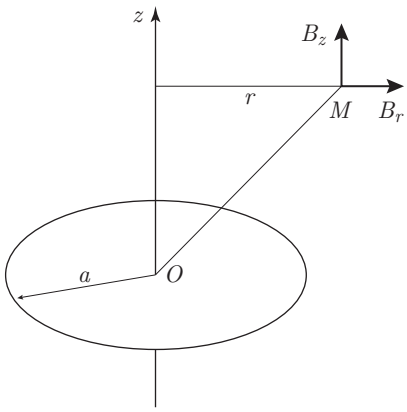


Figure 2.8 – Boucle de courant.

$$\begin{array}{lll} a_1 = \frac{1}{2}(a_0 + b_0) & b_1 = \sqrt{a_0 b_0} & c_1 = \frac{1}{2}(a_0 - b_0) \\ a_2 = \frac{1}{2}(a_1 + b_1) & b_2 = \sqrt{a_1 b_1} & c_1 = \frac{1}{2}(a_1 - b_1) \\ \vdots & \vdots & \vdots \\ a_n = \frac{1}{2}(a_{n-1} + b_{n-1}) & b_n = \sqrt{a_{n-1} b_{n-1}} & c_n = \frac{1}{2}(a_{n-1} - b_{n-1}). \end{array}$$

On arrête le calcul dès que $|c_n|$ est plus petit que la précision cherchée. Pour obtenir $E(k)$ et $K(k)$, il faut choisir

$$a_0 = 1, \quad b_0 = \sqrt{1 - k^2}, c_0 = k.$$

Il vient alors

$$K = \frac{\pi}{2a_n}, \quad \frac{K - E}{K} = \frac{1}{2}[c_0^2 + 2c_1^2 + 2^2 c_2^2 + \cdots + 2^n c_n^2].$$

Vérifiez que votre programme calcule correctement $K(0), E(0), E(1)$ et même $K(0,5) = 1,685750, E(0,5) = 1,467462$.

b) Utilisant les résultats précédents, tracer les lignes de champs de la spire.

Projet 4. Transformation conforme

Si $f(z)$ est une fonction analytique de la variable complexe z dans un domaine du plan complexe, la transformation $w = f(z)$ est appelée une transformation conforme. Lorsque l'image M de z se déplace sur une courbe (C), l'image N de w parcourt une courbe (D). Un logiciel tel que Scilab permet de déterminer (D) sans effort.

a) Nous choisissons comme exemple la fonction $w = 1/z$ (en excluant l'origine). Il suffit alors de créer un vecteur dont les éléments complexes sont les affixes des points de (C), d'exécuter l'instruction `w = z.^(-1)` et enfin de tracer la partie imaginaire de w en fonction de sa partie réelle. Reproduire la figure 2.9, ci-contre ; plus généralement, vérifier que l'ensemble droites+cercles est invariant lors de la transformation (inversion).

Soit $w = f(z)$ une fonction analytique de $z = x + iy$; on pose $w = u + iv$. Les variable u et v sont des fonctions de x et y. On démontre facilement, à partir des conditions de Cauchy, que chacune obéit à l'équation de Laplace. Ainsi, pour u :

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.$$

On dit que u et v sont des fonctions harmoniques conjuguées. De plus les courbes $u(x, y) =$ constante et $v(x, y) =$ constante sont orthogonales. Un certain nombre de phénomènes physiques à deux dimensions sont régis par l'équation de Laplace ; c'est le cas de l'électrostatique et aussi de l'hydrodynamique des fluides non visqueux. Si l'on sait résoudre l'équation de Laplace pour une géométrie simple, une transformation conforme bien choisie pourra permettre de déterminer la solution pour une géométrie plus compliquée. Nous vous proposons d'illustrer cette démarche à propos de deux exemples.

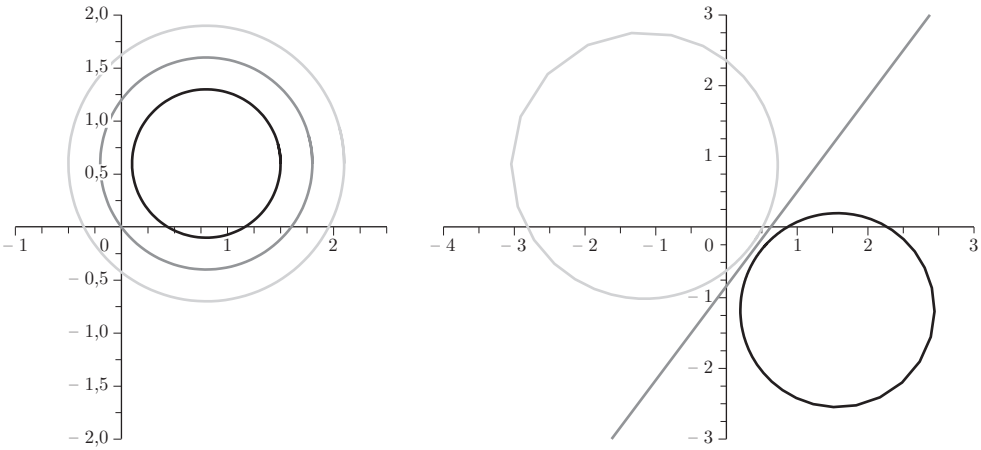


Figure 2.9 – Ensemble de départ (à gauche) et ensemble image (à droite) pour la transformation conforme $w = 1/z$.

b) La détermination de l'écoulement de l'air autour d'un profil d'aile d'avion a fait beaucoup travailler les mécaniciens dans les années 1920. Le problème a été résolu par Joukovsky, comme l'indique la figure 2.10.

La transformation de Joukovsky est de la forme

$$w = z + \frac{b^2}{z}.$$

Saurez-vous déterminer le paramètre b utilisé pour ce tracé ? Comment faut-il déplacer le cercle pour obtenir un profil cambré ?

c) La figure suivante montre en coupe un condensateur plan infiniment long, quelques équipotentielles et des lignes de champ, dans une description très simplifiée où tous les effets de bord ont été négligés. On applique à l'ensemble la suite de transformations

$$w_1 = exp(z); \quad w_2 = w_1 + d; \quad w_3 = 1/w_2.$$

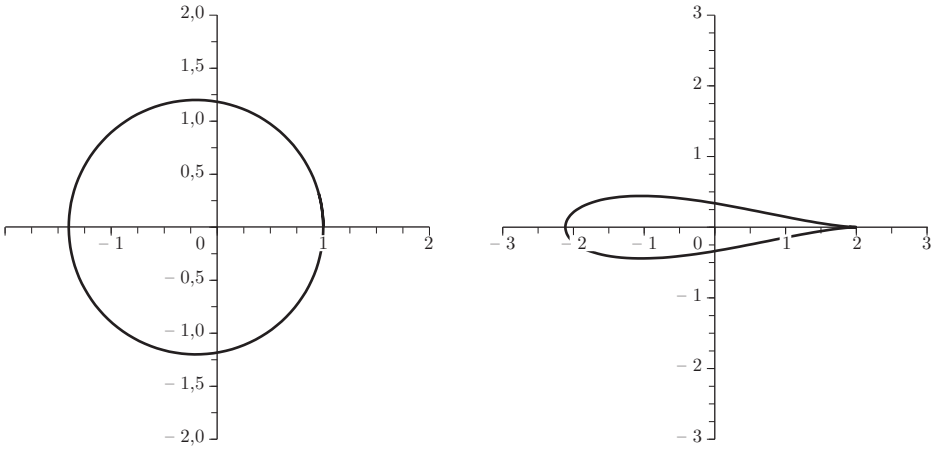


Figure 2.10 – La transformation de Joukovsky. Ensemble de départ (à gauche) et ensemble d'arrivée (à droite).

Les armatures du condensateur ont pour images deux conducteurs cylindriques parallèles dont on voit, sur la figure 2.11, les équipotentielles et les lignes de champ. Quelle est la capacité, par unité de longueur, de ce système de conducteurs ?

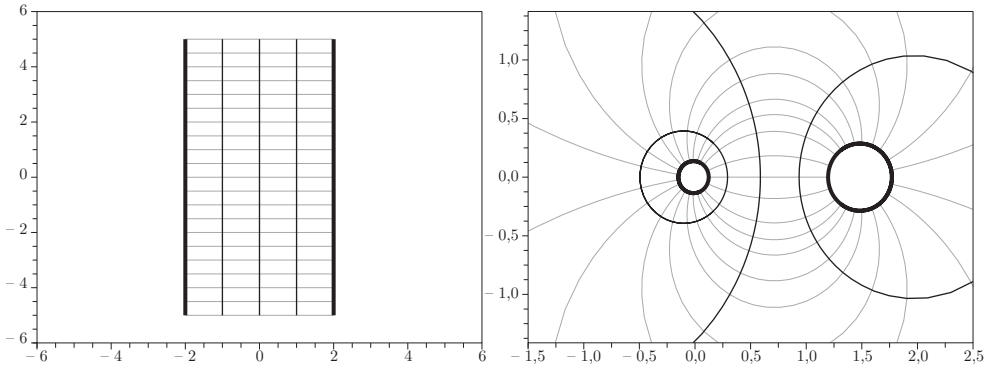


Figure 2.11 – Création d'une ligne bifilaire (à droite) par transformation conforme d'un condensateur plan infiniment long (à gauche).

Chapitre 3

Représentation des grandeurs physiques

Les grandeurs issues du monde réel ne sont pas des nombres purs, elles sont (fâcheusement ?) munies d'unités. On dit : un rôti de 1,5 kg, une solution tampon à la concentration de 15 millimolaire, l'inductance de cette bobine est de 0,27 microhenrys... Plus précisément, le résultat de la mesure d'une grandeur est le quotient de cette grandeur par une autre, de même nature, choisie comme unité. Il faut donc préciser l'unité pour pouvoir communiquer valablement un résultat de mesure. L'unité dans laquelle se mesure une grandeur est le reflet de sa dimension. La capacité d'un lac de barrage se mesure, en France, en km^3, aux États-Unis en pied-arpent : il s'agit toujours du cube d'une longueur. On sait que le système SI comporte sept grandeurs fondamentales ; les autres grandeurs, dites dérivées, s'en déduisent.

Le formalisme mathématique utilisé en sciences physiques ne s'accommode pas aisément du concept d'unité ou de variables « dimensionnées » ; il en est de même de l'analyse numérique. Les fonctions sinus ou logarithme, par exemple, n'admettent que des arguments sans dimension ; plus généralement, les mathématiques opèrent sur des nombres sans dimension. C'est pourquoi la construction d'un modèle mathématique implique le choix systématique de variables sans dimension.

Par ailleurs, « l'analyse dimensionnelle » est une activité bien connue des scientifiques et des ingénieurs. Ils l'utilisent pour vérifier la cohérence de leurs formules (et même, parfois, pour établir la forme d'une relation entre grandeurs) et pour réduire au minimum le nombre de paramètres physiques intervenant dans un modèle. Ces considérations rejoignent celles des programmeurs. Les programmes bien écrits manipulent des variables sans dimensions. Nous savons aussi qu'un programme ne peut traiter que des nombres « représentables en machine ». Il « coincera » si on lui soumet les nombres très grands ou très petits que fournissent les sciences physiques, comme le carré de la constante de Planck ou le rayon de la galaxie, en émettant le message d'erreur `arithmetic overflow`. De plus, l'expérience montre que les variables sans dimension ont souvent des valeurs « raisonnables », pas extrêmement différentes de l'unité.

Nous espérons que ces trois ensembles de raisons vous auront convaincus de n'utiliser que des grandeurs sans dimension dans vos modèles analytiques ou numériques. Dans ce chapitre, nous vous proposons donc, sous forme d'exemples, quelques méthodes pour « adimensionner » un modèle mathématique.

3.1. UNE MÉTHODE SIMPLE DE « DÉDIMENSIONNEMENT »

Commençons par traiter un exemple simple, la conduction de la chaleur dans un milieu à une dimension, comme une tige métallique isolée. Nous supposons que la température θ le long de la tige dépend uniquement de l'abscisse x et du temps t. La densité de flux de chaleur en un point s'écrit :

$$J = -k\frac{\partial\theta}{\partial x} \tag{3.1}$$

où J est la densité de flux ($\mathrm{Jm^{-2}s^{-1}}$), θ est la température et k le coefficient de conductivité. En écrivant le bilan thermique d'une section de la tige, nous obtenons la relation :

$$\rho c\frac{\partial\theta}{\partial t} = k\frac{\partial^2\theta}{\partial x^2} \tag{3.2}$$

où ρ désigne la masse volumique et c la chaleur massique ; tous les paramètres sont ici supposés indépendants de x et t. Cette équation aux dérivées partielles, associée à des conditions aux limites et à des conditions initiales, détermine la fonction $\theta(x,t)$. En apparence, cette équation dépend de trois paramètres : ρ, c et k, mais nous pouvons aussi l'écrire :

$$\frac{\partial\theta}{\partial t} = \kappa\frac{\partial^2\theta}{\partial x^2} \tag{3.3}$$

en introduisant le coefficient de diffusivité thermique κ (kappa). Ce réarrangement banal montre déjà que les solutions ne dépendent que de la combinaison de paramètres $\kappa = k/\rho c$. Mettre l'équation de la chaleur sous cette forme présente un autre avantage : elle devient pratiquement identique à l'équation qui décrit la diffusion d'un soluté dans un solvant ; si $C(x,t)$ est la concentration d'un soluté dont le coefficient de diffusion est D, alors C obéit à

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} \tag{3.4}$$

Toute solution de (3.3) est aussi solution de (3.4) aux notations près. Par ailleurs, les dimensions des deux membres de (3.3) doivent être identiques, soit encore :

$$[\theta]T^{-1} = [\kappa][\theta]L^{-2}$$

ou

$$[\kappa] = L^2T^{-1}$$

où nous utilisons la notation classique $[x]$ pour désigner les dimensions de x. En d'autres termes, κ se mesure en $\mathrm{m^2s^{-1}}$.

Soit maintenant x_0 une longueur (par exemple la longueur de la tige) et t_0 un temps. Nous introduisons une variable de position sans dimension $\xi \equiv x/x_0$ et une variable temporelle $\tau \equiv t/t_0$ également indépendante des unités. La fonction $\theta(x,t)$ devient, avec les nouvelles variables, $\theta(x_0\xi, t_0\tau) \equiv \Theta(\xi,\tau)$. Un calcul de dérivée élémentaire $[\partial\Theta/\partial\xi = (\partial\theta/\partial x)(\partial x/\partial\xi) = x_0(\partial\theta/\partial x)]$ montre que cette fonction obéit maintenant à l'équation :

$$\frac{\partial\,\Theta}{\partial\tau} = \kappa\frac{t_0}{x_0^2}\frac{\partial^2\Theta}{\partial\xi^2} \tag{3.5}$$

Pour que l'équation (3.5) soit indépendante des unités, il nous faut choisir les paramètres t_0 et x_0 de telle manière que $t_0 = x_0^2/\kappa$ et (3.5) s'écrira alors :

$$\frac{\partial\,\Theta}{\partial\tau} = \frac{\partial^2\Theta}{\partial\xi^2} \tag{3.6}$$

3.2. Construction systématique de variables sans dimension

Considérons un autre exemple, qui va nous permettre d'introduire un formalisme différent. Sous une forme un peu abstraite, on peut dire qu'un oscillateur harmonique est constitué d'une masse ponctuelle m se déplaçant le long d'un axe sous l'influence d'un potentiel $V(x) = (1/2)kx^2$. Mettons ce problème en équation, dans le cadre de la mécanique quantique. L'équation de Schrödinger indépendante du temps qui détermine les états stationnaires du système est :

$$-\frac{\hbar^2}{2m}\psi''(x) + \frac{1}{2}kx^2\psi(x) = E\psi(x). \tag{3.7}$$

Résoudre cette équation aux valeurs propres consiste à trouver les valeurs de E et les expressions de ψ compatibles avec (3.7) et avec les conditions aux limites. Ici encore, nous souhaitons travailler sur une forme sans dimensions. (3.7) est homogène en ψ et cette grandeur ne jouera aucun rôle dans le raisonnement. L'équation dépend de la variable indépendante x (une longueur) et des paramètres m, k et $\hbar$. Nous allons chercher une nouvelle unité de longueur, disons a, caractéristique du problème posé. Si une telle grandeur existe, il suffira de remplacer dans (3.7) x par x/a pour obtenir une forme sans dimension. Commençons par recenser les dimensions des paramètres du problème :

$$[m] = M; [k] = [\text{Newton/mètre}] = MT^{-2}; [\hbar] = [\text{Joule.seconde}] = L^2MT^{-1}.$$

Voyons maintenant si un monôme de la forme $a = h^\alpha k^\beta m^\gamma$ peut convenir. Pour le voir, il suffit de chercher les dimensions de a d'après sa définition, sachant que les symboles de dimensions obéissent aux règles habituelles de l'algèbre (ils sont définis comme des rapports d'unités de même nature). Il vient :

$$[a] = (L^2MT^{-1})^\alpha(MT^{-2})^\beta M^\gamma = L^{2\alpha}M^{\alpha+\beta+\gamma}T^{-\alpha-2\beta}.$$

La propriété « a est une longueur » est équivalente au système de trois équations à trois inconnues :

$$\left\{\begin{array}{rcl} \alpha & = & 1/2, \\ \alpha+\beta+\gamma & = & 0, \\ \alpha+2\beta & = & 0, \end{array}\right.$$

soit :

$$\alpha = \frac{1}{2}; \quad \beta = -\frac{1}{4}; \quad \gamma = -\frac{1}{4}.$$

et

$$a = \left(\frac{\hbar^2}{km}\right)^{1/4}. \tag{3.8}$$

Faisons maintenant le changement de variable $x = a\xi, \psi(x) = \psi(a\xi) = \Psi(\xi)$ et $\psi'(x) = (1/a)\Psi'(\xi)$. L'équation (3.7) devient :

$$-\frac{\hbar^2}{2m}\frac{\sqrt{km}}{\hbar}\Psi'' + \frac{k}{2}\frac{\hbar}{\sqrt{km}}\xi^2\Psi = E\Psi.$$

Posons $\omega = \sqrt{k/m}$, la pulsation de l'oscillateur selon la mécanique classique ; l'équation précédente se met alors sous la forme désirée ($\hbar\omega$ a les dimensions d'une énergie) :

$$\Psi'' + \left(\frac{2E}{\hbar\omega} - \xi^2\right)\Psi = 0. \tag{3.9}$$

Un autre avantage des variables sans dimension est qu'il est plus facile de reconnaître les termes petits (et donc négligeables) d'une équation lorsque celle-ci a été écrite en fonction de variables sans unité. Un contre-exemple caricatural peut illustrer ce dernier point. Imaginons une onde plane électromagnétique monochromatique, pour laquelle l'amplitude du champ électrique est de 1 V/m alors que celle du champ magnétique est de 3,333 $\times 10^{-9}$ T ($E = cB$ pour une onde plane). Pouvons-nous négliger un champ magnétique aussi faible ? La physique des équations de Maxwell nous l'interdit : une onde électromagnétique implique l'existence des deux champs. Le champ magnétique a, dans les unités en vigueur, une taille dérisoire, mais cet effet trompeur disparaît si nous comparons des choses comparables, comme les densités d'énergie. La densité d'énergie électrique est de l'ordre de $\varepsilon_0 E^2 \cong 8,85 \times 10^{12}$ Jm^{-3} alors que la densité d'énergie magnétique est environ $B^2/\mu_0 \cong E^2/\mu_0 c^2$, une valeur identique puisque $\varepsilon_0\mu_0 \mathrm{c}^2 = 1$.

Nous pouvons à juste titre nous demander si l'introduction de variables sans dimensions est toujours possible et si le choix en est unique. La réponse à la première question est oui, ce que l'on peut formaliser en énonçant le théorème de Buckingham :

> **Théorème** – Soit un problème physique décrit par n variables liées par une relation. Supposons que les dimensions de ces n variables fassent intervenir exactement m grandeurs fondamentales (comme longueur, masse, etc.). Alors, la relation peut être écrite à l'aide de $n-m$ quantités sans dimension.

Le choix de ces nouvelles grandeurs est parfois arbitraire, dicté soit par la tradition soit par la commodité.

3.3. Pour en savoir plus

– Site du Bureau International des Poids et Mesures :
`http://www.bipm.fr/fr/home/`

– Des sites qui traitent de l'analyse dimensionnelle :

- `http://pagesperso-orange.fr/eddie.saudrais/index.html`
- `http://formation.etud.u-psud.fr/pcsm/physique/outils_nancy/index.htm`
- `http://www.webphysique.fr/` : voir méthodes mathématiques.

3.4. Exercices

Exercice 1

Une onde plane électromagnétique arrive sur un électron animé de la vitesse v. Estimer le rapport de la force électrique f_e à la force magnétique f_m qui s'exercent sur la particule. A quelle condition la seconde est-elle 1000 plus petite que la première ?

Exercice 2

Nous avons construit, dans le même matériau, un modèle réduit de la tige considérée au § 3.1, à l'échelle $1/N$. Après avoir créé un gradient de température dans ce modèle, nous observons que la température redevient à peu près uniforme au bout de 120 s. Quel sera, approximativement, le temps de retour à l'équilibre thermique pour la tige réelle ?

Exercice 3

Le rayon de l'orbite terrestre, considérée comme circulaire, est $R \simeq 1,5 \times 10^{11}$ m, la masse de la terre est $m = 6 \times 10^{24}$ kg, la masse du soleil vaut $M \simeq 2 \times 10^{30}$ kg. Par ailleurs, la constante de la gravitation universelle vaut $G = 6,67 \times 10^{-11}$ SI. Pour simplifier des calculs astronomiques, on décide de choisir le rayon de l'orbite terrestre comme unité de longueur, la masse de la terre comme unité de masse et l'année comme unité de temps.

a) Déterminer la nouvelle valeur de G dans ce système d'unités en utilisant les dimensions de G.

b) Résoudre la même question par la méthode « artisanale » qui consiste à exprimer la période du mouvement circulaire uniforme de la terre autour du soleil, puis à faire $R = T = m = 1$ dans cette formule.

Exercice 4

Les propriétés d'une mole de gaz réel sont bien reproduites par l'équation de van der Waals :

$$\left(p + \frac{a}{v^2}\right)(v - b) = RT \tag{3.10}$$

(R est la constante des gaz parfaits, 8,32 J/K, T la température absolue, a et b deux constantes caractéristiques du gaz considéré). Considérée comme une équation en v, (3.10) admet trois racines tant que T est inférieure à une valeur critique T_c. Pour $T = T_c$, (3.10) admet une racine triple, v_c.

a) Calculer, en fonction de a, b et R, les coordonnées p_c, v_c et T_c du point critique.

b) On introduit les variables sans dimension $p^* \equiv p/p_c, v^* \equiv v/v_c$ et $T^* \equiv T/T_c$. Exprimer (3.10) en fonction de p^*, v^* et T^* et vérifier que toute référence à un corps particulier a disparu. Ce résultat est connu sous le nom de « loi des états correspondants » : lorsque l'on utilise les variables réduites p^*, v^* et T^*, tous les gaz ont le même jeu d'isothermes.

Exercice 5

Une méthode souvent utilisée pour « dédimensionnaliser » une équation consiste à prendre égales à l'unité toutes les constantes physiques qui y figurent, au prix, bien sûr, d'un changement d'unités. En physique atomique, on a souvent à considérer le mouvement d'un électron de charge $-q$ et de masse m autour d'un noyau immobile de charge $+q$. Le système d'unités de Hartree est défini implicitement par les conditions :

$$\hbar = m = \frac{q^2}{4\pi\varepsilon_0} = 1. \tag{3.11}$$

a) Montrer que ces conditions sont compatibles avec les dimensions de $\hbar, m$ et q^2/ε_0.

b) On demande quelles sont les unités de longueur, masse, temps, vitesse et énergie dans le système de Hartree. On peut se servir des équations aux dimensions, mais il est plus commode et parlant de considérer un problème pour lequel on peut obtenir simplement des résultats analytiques : le modèle de Bohr de l'atome. Les conditions précédentes seront imposées tout à la fin. L'électron décrit une orbite circulaire autour du noyau fixe, sous l'effet de l'attraction de celui-ci. Le mouvement est quantifié par la condition que la coordonnée du moment cinétique non nulle soit égale à $\hbar$. Vérifier qu'il en résulte les deux équations :

$$\frac{q^2}{4\pi\varepsilon_0 r^2} = m\frac{v^2}{r}\ ;\ mvr = \hbar.$$

Exprimer r, v et l'énergie totale de l'électron en fonction de $m, q, \hbar$.

c) On impose maintenant les conditions (3.11). En déduire les unités de longueur, vitesse, masse et énergie dans le système de Hartree. L'unité de temps est le temps que met l'électron, animé d'une vitesse unité, à parcourir la distance unité.

Exercice 6

Le mouvement d'un fluide visqueux est régi par l'équation de Navier–Stokes, que l'on peut écrire

$$\rho\frac{d\boldsymbol{v}}{dt} + \rho(\boldsymbol{v}.\nabla)\boldsymbol{v} - \mu\nabla^2\boldsymbol{v} + \nabla p = 0.$$

Les deux premiers termes du premier membre expriment l'inertie d'un élément de volume entourant le point $\boldsymbol{r}$ (proportionnelle à la masse volumique ρ), le troisième

terme traduit la résistance due à la viscosité (caractérisée par μ) et le quatrième représente les forces de pression.

Ce système physique est caractérisé par au moins cinq paramètres : le diamètre du tube, la vitesse du liquide, sa masse volumique et sa viscosité et le gradient de pression. Pour vérifier l'équation du mouvement, on devrait, en principe, faire varier indépendamment chaque paramètre. En fait, comme nous allons le voir en utilisant la méthode de Reynolds, un seul nombre sans dimension détermine la nature de l'écoulement.

Imaginons deux expériences réalisées avec deux tubes de diamètres respectifs a_1 et $a_2 = \alpha a_1$. Nous considérons le nombre α soit comme le facteur d'homothétie qui reflète le changement de taille du montage lorsque l'on passe de l'expérience 1 à l'expérience 2 (deux points homologues ont pour coordonnées respectives $[x_1, y_1, z_1]$ et $[\alpha x_1, \alpha y_1, \alpha z_1]$), soit comme l'effet d'un changement d'unité de longueur. Les vitesses du fluide dans deux sections homologues sont v_1 et $v_2 = \beta v_1$, ce qui définit β. Il faut admettre en conséquence que l'échelle de temps a changé (puisque nous modifions indépendamment la position et la vitesse). Nous supposons que les deux tubes sont remplis de liquides différents, dont les masses volumiques sont liées par $\rho_2 = \gamma \rho_1$ (ce qui implique un changement de l'unité de masse). Il est commode de définir la viscosité cinématique $\nu = \mu/\rho$. Nous posons encore $\nu_2 = \delta \nu_1$. Enfin, les pressions mesurées en des points homologues des deux expériences sont liées par $p_2 = \varepsilon p_1$. Nous pouvons écrire l'équation de Navier–Stokes sous la forme schématique :

$$(\text{accélération}) - \nu \, \nabla^2 \boldsymbol{v} + \frac{1}{\rho} \nabla p = 0.$$

a) Comment doit-on modifier l'échelle des temps, pour tenir compte des transformations des longueurs et des vitesses ? Répondre à la même question pour les masses.

b) Que devient chaque terme de l'équation précédente lorsque l'on passe de l'expérience 1 à l'expérience 2 ?

c) Énoncer deux relations entre les coefficients $\alpha, \beta, \gamma, \delta$ et ε qui expriment que l'équation de Navier–Stokes est vérifiée pour les deux expériences.

d) Déduire de ce qui précède que le nombre de Reynolds

$$R = va/\nu$$

caractérise l'écoulement.

This page intentionally left blank

Chapitre 4

L'interpolation

Dans le chapitre 2, nous avons décrit plusieurs méthodes d'approximation d'une fonction. L'interpolation, qui fait l'objet de ce chapitre, poursuit le même but que l'approximation : remplacer une fonction difficile à calculer par une expression plus simple pour, par exemple, pouvoir la calculer numériquement vite et souvent, ou pour évaluer sa dérivée ou son intégrale. Le cadre mathématique est aussi assez semblable. Nous nous intéressons à une fonction réelle continue $f(x)$ que nous allons encore remplacer par une fonction plus simple $f^*(x)$. La différence réside dans la manière d'imposer la « proximité » de $f(x)$ et de son « substitut » $f^*(x)$. Nous allons demander que $f(x) \equiv f^*(x)$ pour un certain nombre de valeurs de la variable indépendante x. On imposait que l'approximant reste dans une bande de largeur 2ϵ entourant la fonction ; on impose maintenant que l'interpolant coïncide avec la fonction pour certaines valeurs de la variable indépendante. On peut ainsi comprendre le terme d'approximation exacte autrefois employé pour désigner l'interpolation.

Dans un passé lointain, alors qu'il n'existait pas d'ordinateur, le calcul d'une fonction compliquée était laborieux. On dressait une table de la fonction pour quelques valeurs de l'argument puis on interpolait entre ces valeurs. De nos jours, cet usage de l'interpolation a beaucoup régressé, mais l'interpolation reste un outil théorique important. D'autre part, on a souvent besoin des valeurs de grandeurs physiques comme la densité, la conductivité électrique ou des chaleurs de réaction, qui, à leur tour, dépendent de la pression, de la température ou des concentrations. Ces grandeurs figurent dans des tables, mais seulement pour quelques valeurs des paramètres. Là encore, nous serons amenés à interpoler pour déterminer les valeurs qui nous intéressent.

L'échelle internationale de température constitue un bon exemple physique d'interpolation. La température légale est la température thermodynamique, associée à un unique point fixe, le point triple de l'eau à 273,15 K ; elle se mesure à l'aide d'un thermomètre à gaz à volume constant ou d'un thermomètre à rayonnement. Ces deux instruments sont malcommodes et impliquent des manipulations longues et compliquées. Aussi a-t-on choisi une série de points fixes secondaires (points d'ébullition de l'hydrogène, de l'eau, point de fusion du zinc par exemple) dont les températures ont été mesurées très précisément une fois pour toutes. On a choisi aussi quelques

thermomètres « pratiques », comme le thermomètre à résistance de platine. Une fois mesurée la résistance de la sonde pour chaque point fixe, la valeur d'une température intermédiaire est obtenue par interpolation.

4.1. Définition de l'interpolation

Abordons maintenant le formalisme.

Étant donnée une fonction f, une fonction $f^*(x; a_0, a_1, \ldots, a_n)$ dépendant de x et de $n+1$ paramètres $a_0, a_1, \ldots, a_n$ et un ensemble d'abscisses $x_0, x_1, \ldots, x_n$, le problème d'interpolation consiste à déterminer les a_k de telle manière que les $n+1$ équations

$$f^*(x_k; a_0, a_1, \ldots, a_n) = f(x_k), \qquad k = 0, 1, 2, \ldots, n \tag{4.1}$$

soient vérifiées. Dans la suite, nous utiliserons indifféremment les notations $f(x_k) = f_k$; chaque couple de valeurs (x_k, f_k) définit un point du plan que l'on appelle un pivot ou un noeud.

Il nous faut ensuite décider à quelle classe de fonctions appartient f^* ; le critère le plus important est la façon dont interviennent les paramètres a_k. On distingue les cas où f^* dépend linéairement des a_k de tous les autres. Une forme très générale de f^* est

$$f^*(x) = \sum_0^n a_k \varphi_k(x). \tag{4.2}$$

Les qualités de l'interpolation (commodité de calcul, sensibilité aux erreurs d'arrondi par exemple) dépendront du choix des « fonctions de base » $\varphi_k(x)$. L'interpolation polynômiale

$$f^*(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

et l'interpolation trigonométrique

$$f^*(x) = a_0 + a_1 e^{ix} + a_2 e^{2ix} + \cdots + a_n e^{inx}$$

relèvent de ce cas. Par contre, l'interpolation rationnelle

$$f^*(x; a_0, a_1, \ldots, a_m, b_0, b_1 \ldots, b_n) = \frac{a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m}{b_0 + b_1 x + \cdots + b_n x^n}$$

(qui par ailleurs fait appel à $m+n+2$ coefficients) dépend non-linéairement des b_k. Bien que l'interpolation rationnelle soit parfois utilisée, nous nous limiterons ici à l'interpolation polynômiale qui n'implique que des calculs simples. Les formes trigonométriques seront évoquées plus tard (chapitre 9).

Vous remarquez qu'à la différence de l'approximation, il ne parait pas urgent de définir un intervalle dans lequel l'interpolation est valable ; en réalité, cette définition est implicite. Si x_{min} est le plus petit des x_i et x_{max} le plus grand, nous parlons d'interpolation lorsque $x_{min} \leq x \leq x_{max}$; dans la cas contraire (x extérieur à l'intervalle $[x_{min}, x_{max}]$), nous parlons d'extrapolation. Au § 4.4, nous montrerons comment majorer l'erreur correspondante dans ces deux cas. Vous verrez que l'erreur d'extrapolation peut croître sans limite quand x s'éloigne des pivots. En pratique donc, l'extrapolation doit être maniée avec prudence.

4.2. Méthode des coefficients indéterminés

La méthode la plus directe pour trouver les coefficients inconnus a_k consiste à identifier, sur chaque pivot, fonction et polynôme d'interpolation. Montrons le principe de la méthode sur l'exemple de deux pivots. La fonction f est connue numériquement en deux points, ce qui définit les deux pivots $(x_0, f_0), (x_1, f_1)$. La fonction d'interpolation, notée maintenant p, puisqu'il s'agit d'un polynôme, dépend linéairement de deux paramètres inconnus et s'écrit $p(x) = a_0 + a_1 x$. Géométriquement, $p(x)$ est représentée par la droite qui passe par les deux pivots. Les conditions d'interpolation sont alors

$$\left\{ \begin{array}{lcl} a_0 + a_1 x_0 & = & f_0 \\ a_0 + a_1 x_1 & = & f_1 \end{array} \right.$$

Nous devons donc, pour déterminer les $\{a_i\}$, résoudre un système de deux équations linéaires à deux inconnues ; la solution est immédiate :

$$a_0 = \frac{x_0 f_1 - x_1 f_0}{x_0 - x_1} \quad ; \quad a_1 = \frac{f_0 - f_1}{x_0 - x_1}.$$

Ce procédé s'étend sans difficulté au cas d'une fonction du second degré, définie par trois paramètres inconnus ; la courbe représentative de $p(x)$ est une parabole que l'on oblige à passer par trois pivots. Nous pouvons même considérer le cas d'une fonction d'interpolation polynômiale de degré n, dépendant de $n+1$ constantes inconnues. Celles-ci seront déterminées en imposant que f et f^* coïncident sur $n+1$ noeuds :

$$\left\{ \begin{array}{lcl} a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n & = & f_0, \\ a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n & = & f_1, \\ a_0 + a_1 x_2 + a_2 x_2^2 + \cdots + a_n x_2^n & = & f_2, \\ \cdots & = & \cdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n & = & f_n. \end{array} \right. \tag{4.3}$$

Il apparaît ici encore un système d'équations linéaires dont les inconnues sont les $n+1$ nombres a_i et dont le déterminant s'écrit

$$\Delta = \left| \begin{array}{ccccc} x_0^0 & x_0^1 & x_0^2 & \cdots & x_0^n \\ x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^n \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n^0 & x_n^1 & x_n^2 & \cdots & x_n^n \end{array} \right|. \tag{4.4}$$

Δ, que l'on appelle un déterminant de van der Monde, s'annule chaque fois que la condition $x_i = x_j, \quad i \neq j$ est remplie ; le développement de ce déterminant contient donc le facteur $x_i - x_j$. En répétant ce raisonnement pour tous les couples possibles de pivots, nous voyons que Δ est proportionnel à $\prod_{i>j}(x_i - x_j)$. D'autre part, Δ est un polynôme en $x_0, x_1, \ldots, x_n$ de degré maximal n par rapport à chaque variable. Cela implique que le facteur de proportionnalité se réduit à une constante ; on peut montrer que celle-ci vaut 1. Nous déduisons de ce résultat que Δ est non nul, et donc que le système admet une solution unique, si et seulement si les pivots ont des abscisses strictement distinctes. Nous ne faisons pas d'autre hypothèse sur les pivots ; en particulier, ceux-ci peuvent être rangés dans un ordre arbitraire.

Nous disposons maintenant d'une méthode pour construire le polynôme d'interpolation ; si nous avons pris soin de choisir des pivots différents, ce polynôme est unique. Dans la pratique, la méthode des coefficients indéterminés est, en fait, rarement utilisée, car on connaît des algorithmes plus stables et/ou plus rapides. Ces algorithmes, inventés par des mathématiciens célèbres des siècles passés, prennent des formes très différentes et produisent des résultats apparemment dissemblables. En réalité, ils aboutissent tous à former l'unique polynôme d'interpolation à partir des pivots disponibles. La première méthode que nous présenterons est due à Lagrange.

4.3. Le polynôme d'interpolation de Lagrange

Dans cette méthode, nous faisons l'hypothèse (à vérifier) que le polynôme d'interpolation peut s'écrire, pour $n+1$ pivots $[x_i, f(x_i)]$

$$p(x) = \sum_{i=0}^{n} \ell_i(x) f(x_i). \tag{4.5}$$

Chaque polynôme élémentaire de Lagrange $\ell_i(x)$ est de degré n et, par conséquent, p est un polynôme de degré au plus égal à n. Vous voyez que le problème d'interpolation sera résolu si nous pouvons former des ℓ_i répondant aux conditions

$$\ell_i(x_k) = \delta_{i,k}, \qquad i,k = 0,1,2,\ldots,n. \tag{4.6}$$

$\delta_{i,k}$ est le symbole de Kronecker, qui vaut 1 lorsque ses deux indices sont égaux et qui vaut 0 dans tous les autres cas. Si, par exemple, $x = x_2$, alors $\ell_0(x_2) = \ell_1(x_2) = \ell_3(x_2) = \cdots = \ell_n(x_2) = 0$ tandis que $\ell_2(x_2) = 1$. Des $n+1$ termes de p ne subsiste que f_2. Ainsi, pour le pivot x_2, le polynôme d'interpolation coïncide avec la fonction et il en est de même pour tout autre noeud.

Le polynôme élémentaire ℓ_i s'annule pour tous les pivots sauf x_i ; il est donc proportionnel au polynôme $q(x) = (x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)$. La constante de proportionnalité s'obtient en formant $q(x_i) = (x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)$: ℓ_i est égal au polynôme normalisé $q(x)/q(x_i)$. Le polynôme d'interpolation de Lagrange est maintenant entièrement déterminé ; il s'écrit

$$p(x) = \sum_{i=0}^{n} \ell_i(x) f_i = \sum_{i=0}^{n} f_i \prod_{k=0, k\neq i}^{n} \frac{x - x_k}{x_i - x_k}. \tag{4.7}$$

Le calcul numérique de $p(x)$ se programme aisément à l'aide de deux boucles imbriquées ; il est bien plus rapide que la résolution du système linéaire du paragraphe précédent. Le fragment de programme ci-dessous réalise ce calcul.

Listing 4.1 – Calcul du polynôme de Lagrange de degré 3 (4 pivots)

```
x = linspace(xmin,xmax,npts);                                    1
pl = zeros(1,npts);                                              2
for i = 1:4                                                      3
    L(i,1:npts) = ones(1,npts);                                  4
    for k = 1:4                                                  5
        if k <> i                                                6
            L(i,:) = L(i,:).*(x-xp(k))/(xp(i)-xp(k));            7
        end                                                      8
    end                                                          9
    L(i,:) = L(i,:)*fn(xp(i));                                   10
    pl = pl + L(i,:);                                            11
end                                                              12
```

Les deux premières instructions créent un vecteur d'abscisses et un vecteur de zéros, lequel va recevoir les valeurs successives du polynôme d'interpolation. À la ligne 4, nous remplissons la ligne i de la matrice `L` par des 1. Le gros du travail est fait ligne 7, où nous calculons, à l'aide d'une itération, les valeurs du polynôme élémentaire ℓ_i, pour toutes les abscisses simultanément. Nous avons utilisé ce programme pour interpoler la fonction e^x sur les pivots d'abscisses $[0,5; -1; 1,5; 2]$. La figure 4.1 montre le résultat. Nous avons représenté chacune des quantités $\ell_i(x)f(x_i)$, ainsi que leur somme $p(x)$. Vous pouvez vérifier que les $\ell_i(x)f(x_i)$ passent par un pivot et un seul, alors que $p(x)$ passe par tous les noeuds.

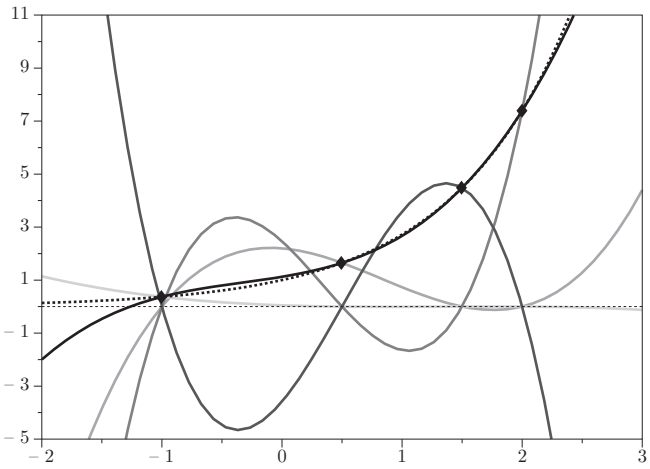


Figure 4.1 – Interpolation de Lagrange.

Cependant, il serait pénible de développer le polynôme de Lagrange et de regrouper les termes pour le mettre sous la forme d'une somme ordonnée selon les puissances de x. Si l'on souhaite vraiment obtenir cette forme, il vaut mieux utiliser un algorithme dû à Newton, la méthode des différences divisées.

4.4. Le polynôme de Newton

4.4.1. Interpolation linéaire

Il est commode de revenir à l'interpolation linéaire. Lorsque nous disons que la fonction $f(x)$ est bien représentée par une interpolation linéaire dans un certain intervalle, cela signifie que le rapport

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

(la pente moyenne de la courbe représentative) est à peu près indépendant de x_0 et x_1 dans cet intervalle. Ce rapport s'appelle la différence divisée du premier ordre, relative à x_0 et x_1, et se note généralement

$$f[x_0, x_1] \equiv \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \tag{4.8}$$

Remarquez que $f[x_0, x_1] = f[x_1, x_0]$. Nous pouvons aussi écrire la relation approchée

$$f[x_0, x] \cong f[x_0, x_1] \tag{4.9}$$

ou encore

$$f(x) \cong f(x_0) + (x - x_0) f[x_0, x_1]. \tag{4.10}$$

Vous reconnaissez, au second membre, le polynôme du premier degré qui interpole f sur les pivots (x_0, x_1), une expression que nous noterons $p_{0,1}$ dans la suite. Il est commode d'introduire aussi les notations

$$p_0(x) \equiv f[x_0] \equiv f(x_0), \tag{4.11}$$

si bien que $f[x_0]$ est la différence divisée d'ordre zéro et $p_0(x)$ est le polynôme d'interpolation de degré zéro qui coïncide avec f en x_0. La formule (4.10) prend alors une forme qu'il sera facile de généraliser plus tard :

$$p_{0,1} = f[x_0] + (x - x_0) f[x_0, x_1]. \tag{4.12}$$

À moins que la fonction f ne soit réellement linéaire en x, la pente de la sécante, $f[x_0, x_1]$, va dépendre des abscisses x_0 et x_1. Si f est du second degré en x, la pente de la sécante joignant les points d'abscisses x_0 et x est elle-même une fonction linéaire de x, à x_0 constant. Nous pouvons donc nous attendre à ce que la quantité

$$\frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

soit indépendante des trois arguments. Dans le cas général, on appelle ce rapport la différence divisée d'ordre 2 relative aux abscisses x_0, x_1, x_2 et on le note $f[x_0, x_1, x_2]$. Cette expression est encore symétrique par rapport à tous ses arguments. Nous pouvons alors écrire la relation (4.9)

$$f[x_0, x] - f[x_0, x_1] = f[x_0, x] - f[x_1, x_0] = (x - x_1) f[x_0, x_1, x]$$

d'où nous déduisons la relation exacte qui remplace (4.12)

$$f(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x].$$

Telle quelle, cette formule est peu utile, car connaître $f[x_0, x_1, x]$ revient à connaître $f(x)$, ce que nous cherchons justement à éviter. Mais l'erreur $E(x)$ commise en remplaçant $f(x)$ par $p_{0,1}$ est donnée par

$$f - p_{0,1} = (x - x_0)(x - x_1)f[x_0, x_1, x],$$

un résultat que nous généraliserons plus tard.

4.4.2. Les différences divisées

Les différences divisées d'ordres $0, 1, 2, \ldots, k$ sont définies par les relations de récurrence

$$\begin{aligned} f[x_0] &= f(x_0), \qquad f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}, \\ f[x_0, \ldots, x_k] &= \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{x_k - x_0}. \end{aligned} \tag{4.13}$$

Dans chaque numérateur, les $k - 1$ premiers arguments du premier terme sont identiques aux $k - 1$ derniers arguments du deuxième terme et le dénominateur est la différence entre les arguments qui diffèrent d'un terme à l'autre. Il est possible de démontrer par récurrence que $f[x_0, \ldots, x_k]$ est une fonction totalement symétrique de ses $k + 1$ arguments ; l'ordre de ceux-ci est donc indifférent. Nous en déduisons que $f[x_0, x_1, \ldots, x_k]$ peut s'exprimer comme le quotient de la différence de deux différences divisées d'ordre $k - 1$ (ayant $k - 1$ arguments en commun) par la différence entre arguments différents :

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_0, x_1, x_2] - f[x_1, x_2, x_3]}{x_0 - x_3} = \frac{f[x_0, x_2, x_3] - f[x_1, x_2, x_3]}{x_0 - x_1}.$$

Jusqu'à présent, nous avons fait semblant d'ignorer la possibilité que deux abscisses puissent être égales ; si cela se produit, il est possible de donner un sens à la différence divisée par passage à la limite :

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

4.4.3. La formule de Newton

D'après la définition (4.13), nous pouvons écrire, en remplaçant à chaque fois un des x_i par x

$$\begin{aligned} f(x) &= f[x_0] + (x - x_0)f[x_0, x], \\ f[x_0, x] &= f[x_0, x_1] + (x - x_1)f[x_0, x_1, x], \\ \cdots &= \cdots \\ f[x_0, \ldots, x_{n-1}, x] &= f[x_0, \ldots, x_n] + (x - x_n)f[x_0, \ldots, x_n, x]. \end{aligned}$$

En substituant la première équation dans la seconde, nous obtenons l'expression déjà vue

$$f(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x]$$

Par des substitutions successives, nous arrivons finalement à

$$\begin{aligned} f(x) = f[x_0] \quad &+ \quad (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] \\ &+ \quad (x - x_0)(x - x_1)\cdots(x - x_{n-1})f[x_0, x_1, \ldots, x_n] \\ &+ \quad (x - x_0)(x - x_1)\cdots(x - x_n)f[x_0, x_1, \ldots, x]. \end{aligned} \tag{4.14}$$

Nous passons de cette expression « formelle » (que nous ne savons pas utiliser puisque nous ignorons le dernier terme) à la formule d'interpolation de Newton en négligeant le terme inconnu :

$$\begin{aligned} p_{0,1,2\ldots,n} = f[x_0] \quad &+ \quad (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] \\ &+ \quad (x - x_0)(x - x_1)\cdots(x - x_{n-1})f[x_0, x_1, \ldots, x_n] \end{aligned} \tag{4.15}$$

L'erreur commise en remplaçant f par p vaut

$$E(x) = (x - x_0)(x - x_1)\cdots(x - x_n)f[x_0, x_1, \ldots, x].$$

Nous donnerons plus tard une forme plus pratique de cette erreur.

Exemple – Voici un exemple de calcul de différences divisées. Nous utilisons les mêmes données qu'au paragraphe précédent (valeurs de e^x aux points $[-1; 0,5; 1,5; 2]$) ; bien que cela ne soit pas nécessaire, nous avons rangé les abscisses par ordre croissant, ce qui rend la suite des opérations plus facile à suivre. Les pivots sont numérotés implicitement de 0 à 3.

ordre	0	1	2	3
-1	$0,367879$			
		$1,280842$		
		$1,5 \quad 0,853895$		
$0,5$	$1,648721$		$1,979073$	
		$2,83297$	$2,5 \quad 0,791629$	
		$1 \quad 2,83297$		$1,196215$
$1,5$	$4,481689$		$2,981766$	$3 \quad 0,398738$
		$2,907367$	$1,5 \quad 1,987844$	
		$0,5 \quad 5,814734$		
2	$7,389056$			

La première colonne contient les valeurs de x, la deuxième celles de f, qui sont identiques aux différences divisées d'ordre zéro. Nous donnons dans la colonne suivante les valeurs des quantités $x_i - x_j$ et $f_i - f_j$, dont le quotient fournit la différence divisée d'ordre 1, reportée dans la colonne numérotée 1.Le calcul se poursuit sans difficulté jusqu'à la différence divisée d'ordre 3, la seule que nous puissions calculer avec les données disponibles.

Le polynôme de Newton d'ordre 3 s'écrit alors

$$\begin{aligned} p_{0,1,2,3} &= 0,367879 + (x-0,5)0,853895 + (x-0,5)(x-1,5)0,791629 \\ &\quad +(x-0,5)(x-1,5)(x-2)0,398738. \end{aligned}$$

Listing 4.2 – Calcul du polynôme de Newton

```
//creation d'une serie de valeurs
 x = [0.5, -1.0, 2.0, 1.5];
 y = exp(x);
 neff = input("ordre de l interpolation (<= 3: ");
//calcul des differences
 for i = 1:neff+1
  t(i) = y(i);
  for j = i-1:-1:1
   t(j) = (t(j+1) - t(j))/(x(i) - x(j));
  end
  a(i) = t(1);
 end
//calcul du polynome
 pol = a(neff+1);
 z = linspace(-2,3);
 for i = neff:-1:1
  pol = pol.*(z-x(i))+a(i);
 end
 plot2d(z',[pol',exp(z)'])
```

Sa forme, très semblable à celle de Horner, rend le calcul facile pour toute valeur de x. Le listing ci-dessus montre un programme Scilab qui accomplit la même tâche. La figure 4.2 illustre le résultat ; comme le polynôme d'interpolation est unique, le résultat final est indiscernable de celui du paragraphe précédent.

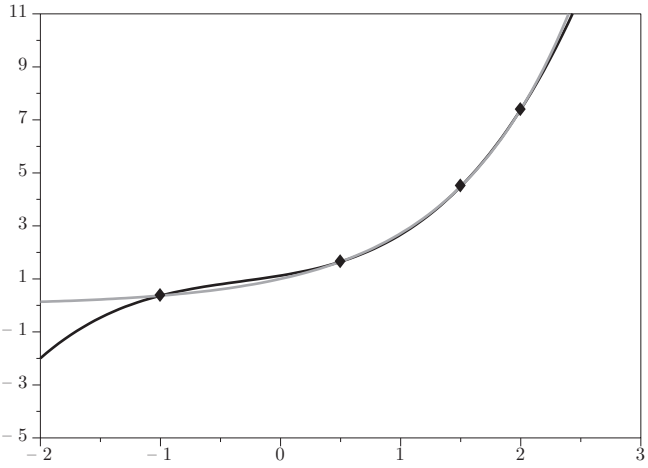


Figure 4.2 – Interpolation par la méthode de Newton.

4.5. L'ERREUR D'INTERPOLATION

Nous allons majorer l'erreur commise en remplaçant la fonction f par le polynôme p de degré n, lequel satisfait aux $n+1$ conditions d'interpolation

$$p(x_i) = f(x_i) \equiv f_i, \qquad i = 0, 1, 2 \ldots, n.$$

Définissons tout d'abord le polynôme de degré $n+1$

$$\pi(x) \equiv (x - x_0)(x - x_1) \cdots (x - x_n) = \prod_0^n (x - x_i)$$

puis la fonction auxiliaire

$$F(x) \equiv f(x) - p(x) - K\pi(x),$$

où K est une constante. Il est possible de choisir la constante K de telle manière que $F(X) = 0$, X étant un point quelconque de l'axe réel. Nous désignerons par I le plus petit intervalle contenant X et tous les x_i. Alors la fonction F admet $n+2$ zéros sur I. En effet, $f - p = 0$ lorsque $x = x_i$ et, simultanément, $\pi(x_i) = 0$. De plus, F s'annule en X.

Le théorème de Rolle appliqué à la fonction F nous permet d'affirmer que la fonction $F'(x)$ présente au moins $n+1$ zéros sur ce même intervalle ; en appliquant ce même théorème aux dérivées successives de F, nous démontrons que F'' admet au moins n zéros dans I, que $F^{(3)}$ en admet au moins $n-1$ et finalement que $F^{(n+1)}$ s'annule au moins une fois dans I (à condition que cette dérivée existe) ; notons ξ ce point. La dérivée d'ordre $n+1$ de F se calcule aisément ; $p^{(n+1)} = 0$ puisque p est de degré n. Le terme de degré le plus élevé de $\pi(x)$ est x^{n+1}, si bien que la dérivée cherchée est $\pi^{(n+1)} = (n+1)!$. Nous en déduisons que

$$F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - K(n+1)! = 0$$

ou encore que

$$K = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

d'où nous tirons que

$$f(X) - p(X) = K\pi(X) = \frac{\pi(X)}{(n+1)!} f^{(n+1)}(\xi).$$

Nous pouvons donc énoncer le théorème suivant.

> **Théorème** – Si une fonction f possède une dérivée d'ordre $n+1$, alors, pour toute valeur X de l'argument, il existe un nombre ξ appartenant au plus petit intervalle qui contient X et tous les pivots x_i et satisfaisant à
>
> $$f(X) - p_{0,1,2,\ldots,n}(X) = \frac{\pi(X) f^{(n+1)}(\xi)}{(n+1)!}. \tag{4.16}$$

En pratique, comme nous ne connaissons pas le nombre ξ et que nous cherchons un résultat valable pour tout l'intervalle I, nous utiliserons un majorant de la valeur absolue du second membre :

$$|f - p| \leq \frac{1}{(n+1)!} \sup_{x \in I} |\pi f^{(n+1)}|.$$

Cette relation est souvent assez pessimiste. D'autre part, son utilité pratique est assez faible parce que si f est compliquée (et c'est pour cela que nous souhaitons la remplacer par un polynôme), sa dérivée d'ordre $n+1$ a toutes les chances d'être impossible à majorer facilement.

On pourrait penser que l'erreur diminue lorsque le nombre de pivots augmente, ou plutôt, lorsque ceux-ci deviennent de plus en plus serrés. Il n'en est rien ; le théorème de Faber établit au contraire l'existence de fonctions pour lesquelles l'erreur augmente avec le nombre de noeuds. La figure 4.3 vous présente un exemple de cette propriété paradoxale, connue sous le nom de phénomène de Runge. Nous avons interpolé la fonction $1/(1+x^2)$ en utilisant 12 pivots régulièrement espacés sur l'intervalle $[-5, 5]$; vous voyez que l'erreur d'interpolation devient très grande aux bords de l'intervalle.

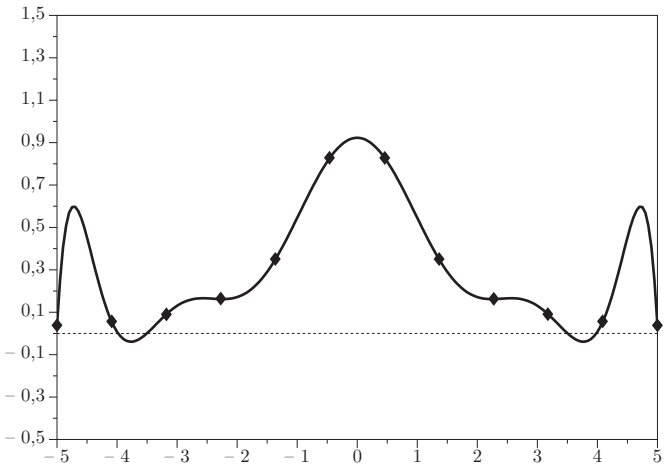


Figure 4.3 – Interpolation polynômiale de la fonction de Lorentz.

Nous pouvons cependant réaliser une interpolation plus précise en choisissant « bien » les pivots. Les pivots définis par l'instruction Scilab

```
for k = 1:npiv, xp(k) = 0.5*(xmax - xmin)*cos( (2*k-1)*%pi/(2*npiv) ); end;
```

sont particulièrement efficaces, comme le montre la figure 4.4.

À titre d'exercice, nous vous incitons à chercher l'explication de ce bon comportement. Elle tient à ce que ces abscisses sont proportionnelles aux zéros du polynôme de Tschebychef d'ordre `npiv`.

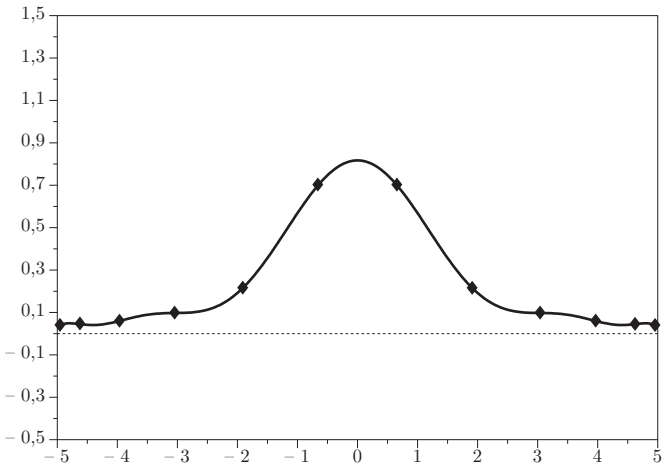


Figure 4.4 – Interpolation de la fonction de Lorentz sur des abscisses de Tschebychef.

4.6. Interpolation entre pivots équidistants

Nous supposons disposer d'une table des valeurs d'une fonction $f(x)$ calculées pour des valeurs entières de x et nous avons besoin de connaître $f(3,4)$. Nous pouvons bien sûr utiliser les connaissances acquises en lisant les paragraphes précédents et interpoler entre les valeurs $f(3)$ et $f(4)$ par exemple. Les algorithmes de Lagrange ou de Newton s'appliquent sans difficulté, mais ici les données présentent une caractéristique intéressante : les pivots sont régulièrement espacés. Une telle situation se rencontre fréquemment lorsque l'on calcule à la main et de nombreuses méthodes ont été développées pour tirer parti au mieux de ce cas particulier. D'autre part, le même formalisme est utilisé pour résoudre numériquement les équations différentielles et les équations aux dérivées partielles. Ceci justifie que nous consacrions quelques pages à l'interpolation entre pivots équidistants.

4.6.1. Les différences finies latérales

Étant donné une fonction $f(x)$ et une constante positive h, nous définissons la différence latérale ascendante

$$\Delta_h f(x) \equiv f(x+h) - f(x).$$

En général, la constante h est définie par le contexte, ce qui nous dispense de la faire figurer en indice. Comme nous ne travaillerons, dans la suite, qu'avec des abscisses équidistantes $x_i = x_0 + ih, \quad i = 0, 1, 2, 3, \ldots$, nous utiliserons la notation plus concise

$$\Delta f(x_i) \equiv \Delta f_i = f(x_{i+1}) - f(x_i) \equiv f_{i+1} - f_i. \tag{4.17}$$

h s'appelle le pas ou l'intervalle tabulaire et $f(x_{i+1}) = f(x_i + h)$. Définissons maintenant par récurrence les différences latérales d'ordre supérieur :

$$\Delta^{p+1} f \equiv \Delta^p f(x+h) - \Delta^p f(x), \quad p \geq 0. \tag{4.18}$$

Par convention, $\Delta^0 f(x) \equiv f(x)$. Ainsi, $\Delta^2 f = f(x+2h) - 2f(x+h) + f(x)$. Le calcul pratique des différences finies se fait commodément à l'aide d'un tableau (semblable à celui utilisé pour former les différences divisées, mais maintenant sans division).

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
x_0	f_0			
		Δf_0		
x_1	f_1		$\Delta^2 f_0$	
		Δf_1		$\Delta^3 f_0$
x_2	f_2		$\Delta^2 f_1$	
		Δf_2		$\Delta^3 f_1$
x_3	f_3		$\Delta^2 f_2$	
		Δf_3		$\Delta^3 f_2$
x_4	f_4		$\Delta^2 f_3$	
		Δf_4		
x_5	f_5	$\vdots$	$\vdots$	

4.6.2. La formule d'interpolation de Newton

Il existe une relation simple entre le polynôme de Newton (formé à l'aide de différences divisées) et les différences latérales : pour tout entier k positif,

$$f[x_0, x_1, x_2, \ldots, x_k] = \frac{1}{k! h^k} \Delta^k f_0. \tag{4.19}$$

Cette formule se démontre par récurrence. Elle est manifestement vraie pour $k = 0$. Nous avons, pour $k = 1$,

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{1}{h} \Delta f_0,$$

ce qui est conforme au résultat proposé. Supposons maintenant que la formule est vraie pour tout ordre $k \leq r$. Alors, pour $k = r + 1$, la propriété (4.13) implique que

$$f[x_0, x_1, \ldots, x_{k+1}] = \frac{f[x_1, x_1, \ldots, x_{r+1}] - f[x_0, \ldots, x_r]}{x_{r+1} - x_0}.$$

En remplaçant les différences divisées d'ordre r du second membre par leurs expressions, nous trouvons

$$\frac{1}{(r+1)h} \left[\frac{1}{r! h^r} \Delta^r f_1 - \frac{1}{r! h^r} \Delta^r f_0 \right] = \frac{1}{(r+1)! h^{r+1}} \Delta^{r+1} f_0,$$

ce qui établit le résultat pour $k = r + 1$. Il reste à utiliser (4.19) pour transformer le polynôme d'interpolation de Newton. Nous définissons une variable réduite

$$\mu \equiv \frac{x - x_0}{h}.$$

Nous voyons ensuite que $x - x_j = h(\mu - j)$ et que

$$(x - x_0)(x - x_1) \cdots (x - x_k) = \mu(\mu - 1) \cdots (\mu - k)h^{k+1}.$$

Insérant ces résultats dans (4.15), nous obtenons

$$p_{0,1,\ldots,n}(x) = f_0 + \mu h \frac{\Delta f_0}{h} + \mu(\mu - 1)h^2 \frac{\Delta^2 f_0}{2!h^2} + \cdots + \mu(\mu - 1) \cdots (\mu - n + 1) \frac{\Delta^n f_0}{n!h^n}.$$

Nous allons donner à cette formule une allure plus compacte en introduisant des analogues fractionnaires des coefficients du binôme

$$\mathcal{C}_\mu^k \equiv \frac{\mu(\mu - 1) \cdots (\mu - k + 1)}{k!}. \tag{4.20}$$

Avec cette définition, le polynôme d'interpolation s'écrit

$$p_{0,1,\ldots,n}(x) = \sum_{j=0}^{n} \mathcal{C}_\mu^j \Delta^j f_0. \tag{4.21}$$

Cette formule générale devient, pour $n = 1$

$$p_{0,1}(x) = f_0 + \mu \Delta f_0$$

et, pour $n = 2$,

$$p_{0,1,2} = f_0 + \mu \Delta f_0 + \frac{1}{2}\mu(\mu - 1)\Delta^2 f_0.$$

Exemple – Voici un exemple complet d'interpolation par les différences finies et la formule de Newton. Nous avons préparé un tableau de f et de ses différences latérales.

i	x_i	f_i	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$	$\Delta^5 f$
0	1	1					
			0,61051				
1	1,1	1,61051		0,26730			
			0,87781		0,0795		
2	1,2	2,48832		0,3468		0,0144	
			1,22461		0,0939		0,0012
3	1,3	3,71293		0,4407		0,0156	
			1,66531		0,1095		
4	1,4	5,37824		0,5502			
			2,21551				
5	1,5	7,59375					

Pour cette fonction assez régulière, les différences diminuent lorsque l'ordre croît, mais le nombre de chiffres significatifs diminue aussi.

Nous cherchons la valeur de $f(1,25)$. Nous décidons d'utiliser toutes les valeurs disponibles, à commencer par f_0. Pour cet exemple, $\mu = (1,25-1)/0,1 = 2,5$. Les termes successifs du polynôme d'interpolation s'écrivent

f_0	1
$(2,5)\Delta^1 f_0$	$1,526275$
$\frac{(2,5)(1,5)}{2}\Delta^2 f_0$	$0,501187$
$\frac{(2,5)(1,5)(0,5)}{6}\Delta^3 f_0$	$0,024844$
$\frac{(2,5)(1,5)(0,5)(-0,5)}{24}\Delta^4 f_0$	$-0,000563$
$\frac{(2,5)(1,5)(0,5)(-0,5)(-1,5)}{120}\Delta^5 f_0$	$0,000014$
$p(1,25)$	$3,051758$

Il est facile, avec cette méthode, de suivre les « progrès » de l'interpolation : nous arrêtons le calcul dès que nous rencontrons une contribution « assez petite ».

Il existe bien d'autres formules d'interpolation, associées aux noms de Gauss, Bessel et autres et décrites dans les ouvrages cités.

4.7. Le polynôme d'interpolation de Hermite

Dans le but de construire une fonction d'interpolation f^* encore plus précise, nous pouvons imposer à celle-ci des conditions supplémentaires. Hermite a proposé que les dérivées de f^* coïncident avec les dérivées correspondantes de f pour certains pivots. Nous allons suivre cette idée, en nous limitant à l'interpolation polynômiale et en ne considérant que la dérivée première. Pour tous les pivots, nous imposons que

$$p(x_i) \equiv f_i \quad ; \quad p'(x_i) \equiv f'(x_i) = f'_i, \quad i = 0, 1, 2, \ldots, n. \tag{4.22}$$

En d'autre termes, $p(x)$ interpole f et $p'(x)$ interpole f', sur les mêmes pivots. Nous supposons que le polynôme d'interpolation de Hermite, ainsi défini, peut s'écrire sous une forme analogue à celle de Lagrange

$$p(x) = \sum_{i=0}^{n} h_i(x) f_i + \sum_{i=0}^{n} \bar{h}_i(x) f'_i, \tag{4.23}$$

où les $h_i, \bar{h}_i$ sont des polynômes réels distincts. Quel en est le degré ? Le polynôme p doit satisfaire aux $2n+2$ conditions (4.22) : il comporte donc $2n+2$ coefficients

ajustables et il est de degré $2n+1$. Les polynômes élémentaires h_i et $\bar{h}_i$ ont donc un degré au plus égal à $2n+1$.

Les polynômes h_i sont tels que p interpole f ; ils obéissent donc à des conditions identiques à celles que nous avons écrites pour les polynômes élémentaires de Lagrange. Il paraît de même raisonnable que les polynômes dérivés $\bar{h}'_i$ soient choisis de façon à ce que p' interpole f', d'où une nouvelle série de conditions. Mais cela ne suffit pas : il ne faut pas que les h'_i viennent perturber l'interpolation de f, ni que les $\bar{h}_i$ détruisent l'interpolation de f' : ces polynômes doivent être nuls sur les pivots. Nous arrivons ainsi à quatre séries de conditions

$$\begin{array}{lll} h_i(x_k) = \delta_{i,k} & i,k = 0,1,2,\ldots,n, & (a) \\ h'_i(x_k) = 0 & i,k = 0,1,2,\ldots,n, & (b) \\ \bar{h}_i(x_k) = 0 & i,k = 0,1,2,\ldots,n, & (c) \\ \bar{h}'_i(x_k) = \delta_{i,k} & i,k = 0,1,2,\ldots,n. & (d) \end{array} \tag{4.24}$$

Chaque polynôme élémentaire obéit à $2n+2$ conditions et peut donc être de degré $2n+1$ au plus.

Commençons par construire le polynôme $\bar{h}_i$. Il s'annule ainsi que sa dérivée sur tous les pivots $k \neq i$; il admet donc des racines doubles en ces points et aussi une racine simple en x_i. Il s'exprime facilement à l'aide du polynôme élémentaire de Lagrange $\ell_i(x)$ construit sur les mêmes pivots et du facteur $x - x_i$:

$$\bar{h}_i(x) = C[\ell_i(x)]^2(x - x_i).$$

La dérivée $\bar{h}'_i(x_i)$ vaut 1 (condition (4.24d)), ce qui impose $C \equiv 1$.

Le polynôme $h_i(x)$ admet tous les pivots sauf x_i comme racines doubles. Il peut donc s'écrire

$$h_i(x) = [\ell_i(x)]^2 t_i(x),$$

en désignant par t_i un polynôme du premier degré tel que h_i respecte les conditions (4.24a,b) en x_i :

$$h_i(x_i) = [\ell_i(x_i)]^2 t_i(x_i) = t_i(x_i) = 1; \quad h'_i(x_i) = 2\ell_i(x_i)\ell'_i(x_i) + \ell_i^2(x_i)t'_i(x_i) = 0.$$

Un calcul sans difficulté montre que

$$t_i(x) = 1 - 2(x - x_i)\ell'_i(x_i).$$

L'erreur d'interpolation s'obtient par un raisonnement calqué sur celui de § 4.5 ; elle s'écrit

$$f(X) - p(X) = \frac{[\pi(X)]^2}{(2n+2)!} f^{(2n+2)}(\xi), \tag{4.25}$$

l'abscisse ξ appartenant au plus petit intervalle qui contient X et tous les pivots x_i.

4.8. L'INTERPOLATION INVERSE

Lorsque nous disposons d'une table de valeurs d'une fonction f (c'est-à-dire d'un ensemble de couples (x_i, f_i)) et que nous cherchons la valeur de l'argument x correspondant à une valeur de f ne figurant pas dans la table, nous devons effectuer une

« interpolation inverse ». Il s'agit en fait d'interpoler la fonction inverse f^{-1} sur des pivots qui ne sont généralement pas équidistants.

La recherche d'une racine de l'équation $f(x) = 0$ peut relever de ce formalisme. Si nous connaissons les valeurs de f pour des abscisses encadrant la racine, nous obtiendrons une bonne approximation de celle-ci par interpolation.

Exemple – Cherchons la solution de $\cos x = x$, sachant qu'elle est voisine de 0,7 radian. Nous construisons la table suivante :

x	$\cos x - x$
$0,5$	$0,377583$
$0,6$	$0,225336$
$0,7$	$0,064842$
$0,8$	$-0,103293$
$0,9$	$-0,278390$

La racine est telle que $0,7 < f^{-1}(0) < 0,8$; approchons la par interpolation linéaire selon Lagrange :

$$x^* = 0,7\frac{0-(-0,103293)}{0,064842-(-0,103293)} + 0,8\frac{0-0,064842}{-0,103293-0,064842}$$

soit $x^* = 0,738565$ et $f(x^*) = 0,00087$.

4.9. L'interpolation par intervalle

Cherchant à améliorer la qualité de l'interpolation d'une fonction donnée, nous sommes tentés d'augmenter le nombre n de pivots (ou de noeuds). Comme vous le savez, cette solution est souvent vouée à l'échec, parce que le terme d'erreur contient deux facteurs, le polynôme $\pi(x)$ et la dérivée d'ordre n (ou $2n$) de la fonction qui sont vraisemblablement rapidement croissants avec n.

On peut cependant augmenter à volonté le nombre de pivots, *sans* faire croître l'ordre d'interpolation. La figure 4.5 montre le principe de ce tout petit miracle. Pour interpoler la fonction représentée par la courbe en noir, nous avons défini 5 pivots (d'abscisses $-1,8; -0,5; 0; 0,5; 1,4$) et, ce faisant, 4 intervalles. Nous pratiquons une interpolation du premier degré dans chaque intervalle ; la fonction linéaire correspondante est définie pour tout x : elle est représentée par un trait noir. Cependant, nous n'utilisons que la restriction de cette fonction à l'intervalle considéré, c'est-à-dire que nous assimilons l'arc de courbe au segment représenté en trait gris épais.

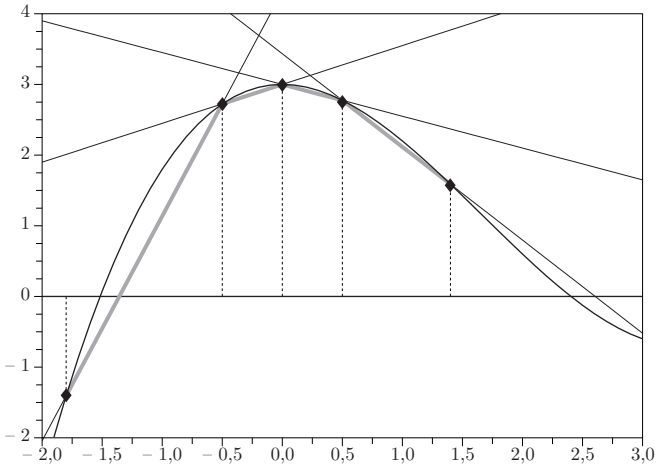


Figure 4.5 – Interpolation linéaire par morceaux.

La fonction d'interpolation f^* est maintenant plus compliquée : elle admet une définition différente dans chaque intervalle. De plus, sa dérivée n'est plus continue. Ce type d'interpolation peut être rendu aussi précis que l'on veut en multipliant le nombre d'intervalles. On peut aussi le perfectionner en choisissant une méthode plus précise dans chaque intervalle, interpolation parabolique ou interpolation de Hermite.

Cependant, dans certaines applications, la disparition de la continuité de la dérivée pose problème : imagine-t-on un bureau d'étude qui calcule une coque de bateau formée d'une série de facettes raccordées par des angles vifs ? Il existe un formalisme qui combine l'avantage d'une interpolation « par morceaux » et la continuité des dérivées de la fonction d'interpolation : c'est la méthode des splines, que nous expliquons dans le paragraphe suivant.

4.10. L'INTERPOLATION « SPLINE »

Le mot « spline » désigne en anglais une règle souple et élastique (une baguette de bois ou une latte) dont les dessinateurs se servaient pour tracer des courbes lisses (ou régulières) passant par des points imposés. Le calcul du profil d'un pont ou d'une aile d'avion fournit les coordonnées d'un ensemble de points ; il appartient ensuite au dessinateur de produire une courbe esthétique ou aérodynamique mais passant par les points calculés. La forme de la règle souple était ajustée en y suspendant des poids ou en y attachant des ressorts en des points bien choisis. L'interpolation spline est la traduction algébrique de ce savoir-faire.

Nous allons détailler un type particulier de fonctions spline, les splines cubiques, qui sont fréquemment utilisées et sont représentatives de cette catégorie de fonctions. Nous nous proposons d'interpoler la fonction $f(x)$ sur l'intervalle $I = [a, b]$. Choisissons pour cela une partition de I

$$a \equiv x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n \equiv b. \tag{4.26}$$

Ces $n+1$ abscisses définissent n sous-intervalles $[x_{i-1}, x_i]$. La fonction spline $s(x)$ répond alors aux contraintes suivantes

- Dans chaque intervalle $[x_{i-1}, x_i]$, la fonction d'interpolation est un polynôme de degré trois.
- La fonction s, sa dérivée première s' et sa dérivée seconde s'' sont continues dans I.
- $s(x)$ interpole la fonction f sur $[a,b]$, $s(x_i) = f_i, i = 0, 1, \ldots, n$.

Pour que le problème de la détermination de la fonction s soit soluble, il faut au moins que le nombre d'équations soit égal au nombre d'inconnues. Cela est-il le cas ? Nous pourrions poser

$$s(x) = a_i + b_i x + c_i x^2 + d_i x^3 \qquad x_{i-1} \leq x \leq x_i \qquad i = 1, 2, \ldots, n. \tag{4.27}$$

Les inconnues sont les $4n$ coefficients a_i, b_i, c_i, d_i. Les conditions d'interpolation fournissent $n+1$ contraintes. Les conditions de continuité s'appliquent à chaque frontière entre sous-intervalles ; il y a $n-1$ points communs à deux intervalles et 3 séries de conditions, ce qui engendre $3n-3$ contraintes. Nous disposons donc en tout de $4n-2$ relations pour déterminer $4n$ inconnues. Ce léger déficit n'est pas gênant et nous introduirons bientôt deux conditions supplémentaires qui nous permettront de résoudre entièrement le problème.

Plutôt que de faire intervenir directement les coefficients du polynôme s (comme dans (4.27)), il est commode d'utiliser comme inconnues les quantités $m_i \equiv s''(x_i), i = 0, 1, 2, \ldots, n$. Comme s doit être du troisième degré sur $[x_i, x_{i+1}]$, sa dérivée seconde est une fonction linéaire de x que nous écrivons sous une forme qui minimise les calculs à venir et assure la continuité de s'' :

$$\begin{aligned} &s''(x) \equiv \frac{1}{h_i}[(x_{i+1} - x)m_i + (x - x_i)m_{i+1}], \quad \text{pour } x \in [x_i, x_{i+1}], \\ &h_i \equiv x_{i+1} - x_i, \qquad\qquad i = 0, 1, 2, \ldots, n-1. \end{aligned} \tag{4.28}$$

Intégrons (4.28) deux fois par rapport à x ; il apparaît deux constantes arbitraires C_i et D_i :

$$s(x) = \frac{(x_{i+1} - x)^3 m_i + (x - x_i)^3 m_{i+1}}{6h_i} + C_i(x_{i+1} - x) + D_i(x - x_i).$$

Imposant maintenant les conditions d'interpolation, nous obtenons les expressions des constantes d'intégration

$$C_i = \frac{f_i}{h_i} - \frac{h_i m_i}{6} \qquad D_i = \frac{f_{i+1}}{h_i} - \frac{h_i m_{i+1}}{6}.$$

Le polynôme s prend alors une forme assez symétrique

$$\begin{aligned} s(x) = &\frac{(x_{i+1} - x)^3 m_i + (x - x_i)^3 m_{i+1}}{6h_i} + \frac{(x_{i+1} - x)f_i + (x - x_i)f_{i+1}}{h_i} \\ &- \frac{h_i}{6}[(x_{i+1} - x)m_i + (x - x_i)m_{i+1}], \\ &x_i \leq x \leq x_{i+1}, \qquad\qquad i = 0, 1, 2, \ldots, n-1. \end{aligned} \tag{4.29}$$

Vous pouvez vérifier que s est bien continue sur $[a,b]$. Il nous reste à prendre en compte la continuité de s' ; pour cela, exprimons cette dérivée dans deux intervalles successifs. Dans $[x_i, x_{i+1}]$

$$s'(x) = \frac{-(x_{i+1}-x)^2 m_i + (x-x_i)^2 m_{i+1}}{2h_i} + \frac{f_{i+1}-f_i}{h_i} - \frac{(m_{i+1}-m_i)h_i}{6}$$

alors que dans l'intervalle précédent $[x_{i-1}, x_i]$:

$$s'(x) = \frac{-(x_i-x)^2 m_{i-1} + (x-x_{i-1})^2 m_i}{2h_{i-1}} + \frac{f_i-f_{i-1}}{h_{i-1}} - \frac{(m_i-m_{i-1})h_{i-1}}{6}.$$

Ces deux expressions doivent vérifier

$$\lim_{x\to x_i^+} = \lim_{x\to x_i^-} \qquad i = 1, 2, \ldots, n-1.$$

Après un peu d'algèbre, nous aboutissons à l'équation suivante

$$\frac{h_{i-1}}{6} m_{i-1} + \frac{h_{i-1}+h_i}{3} m_i + \frac{h_{i+1}}{6} m_{i+1} = \frac{f_{i+1}-f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}, \quad i = 1, 2, \ldots, n-1. \tag{4.30}$$

Nous venons d'obtenir $n-1$ équations linéaires pour les $n+1$ inconnues m_i. De plus, ces équations ont une structure simple : chacune ne fait intervenir que trois inconnues.

Nous devons maintenant imaginer deux équations supplémentaires. Il est d'usage d'imposer une condition raisonnable à chaque extrémité (a,b) de l'intervalle. On peut, par exemple, supposer connues les pentes du polynôme d'interpolation en ces points :

$$s'(x_0) = f'_0 \qquad s'(x_n) = f'_n, \tag{4.31}$$

où f'_0, f'_n sont des constantes. En utilisant les expressions de s' vues précédemment, pour $i = 0$ et pour $i = n-1$, nous trouvons les deux équations manquantes

$$\frac{h_0}{3} m_0 + \frac{h_0}{6} m_1 = \frac{f_1 - f_0}{h_0} - f'_0,$$
$$\frac{h_{n-1}}{6} m_{n-1} + \frac{h_{n-1}}{3} m_n = f'_n - \frac{f_n - f_{n-1}}{h_{n-1}}.$$

L'ensemble des équations définissant les m_i peut être mis sous forme matricielle

$$\boldsymbol{A}\boldsymbol{m} = \boldsymbol{b}$$

en définissant

$$\boldsymbol{b}^T \equiv \left[\frac{f_1-f_0}{h_0} - f'_0, \frac{f_2-f_1}{h_1} - \frac{f_1-f_0}{h_0}, \ldots, \right.$$
$$\left. \frac{f_n - f_{n-1}}{h_{n-1}} - \frac{f_{n-1}-f_{n-2}}{h_{n-2}}, f'_n - \frac{f_n-f_{n-1}}{h_{n-1}} \right],$$

$$\boldsymbol{m}^T \equiv [m_0, m_1, \ldots, m_n],$$

et

$$\boldsymbol{A} = \begin{bmatrix} \frac{h_0}{3} & \frac{h_0}{6} & 0 & & & \cdots & 0 \\ \frac{h_0}{6} & \frac{h_0+h_1}{3} & \frac{h_1}{6} & & & & \\ 0 & \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & & \\ 0 & & & \ddots & & & \vdots \\ \vdots & & & & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} & \frac{h_{n-1}}{6} \\ 0 & & \cdots & & & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} \end{bmatrix}.$$

Ces équations définissent ce que l'on appelle parfois l'interpolant spline « complet » (ou « bloqué », « clamped » en anglais). Nous sommes parvenu à concilier nombre de pivots arbitraire et continuité des deux premières dérivées, mais il y a un prix à payer : nous devons maintenant résoudre un problème global (représenté par un système d'équations) et toute modification locale d'une donnée (par exemple de f_2) se répercutera sur toutes les valeurs de la fonction spline. Nous n'aborderons pas ici l'étude de l'erreur d'interpolation, qui nécessite de trop longs développements et que l'on trouvera dans les ouvrages spécialisés.

Le formalisme assez touffu qui précède est entièrement masqué à l'intérieur des deux fonctions de Scilab `splin` et `interp`. Remarquons tout d'abord que Scilab utilise, pour déterminer complètement les m_i et sauf indication contraire, deux conditions qui ne font intervenir aucune donnée supplémentaire. Dans ce logiciel, on impose en effet la continuité de la dérivée troisième de s pour le deuxième et l'avant-dernier pivot, qui ne relèvent donc plus de la définition stricte. Cela revient à dire que s est un interpolant spline pour les noeuds $x_0, x_2, x_3, \ldots, x_{n-2}, x_n$, mais interpole cependant sur tous les pivots $x_0, x_1, x_2, \ldots, x_{n-1}, x_n$; c'est ce que traduit le terme anglais de « not-a-knot ». Vous pouvez cependant choisir un autre type de spline (naturel ou bloqué par exemple) en insérant le mot clé correspondant dans les arguments de `splin`.

Exemple – Reprenons l'exemple du § 3, l'interpolation de la fonction $1/(1+x^2)$. Voici le programme Scilab, suivi du graphe correspondant (fig. 4.10).

Listing 4.3 – Construction d'un interpolant « spline »

```
deff("y = fn(x)", "y = (1.0)./(1+x.*x)");
npts = 200;
npiv = input("nombre de pivots: ");
xmin = -5; xmax = 5;
x = linspace(xmin,xmax,npts);
xp = linspace(xmin,xmax,npiv);
xset("window",0), xbasc(0), xset("mark size", 3)
plot2d(xp,fn(xp),-4,rect = [xmin, -0.5,xmax,1.5])
spl1 = splin(xp,fn(xp));
spl2 = interp(x,xp,fn(xp),spl1);
plot2d(x,spl2,2,"000")
```

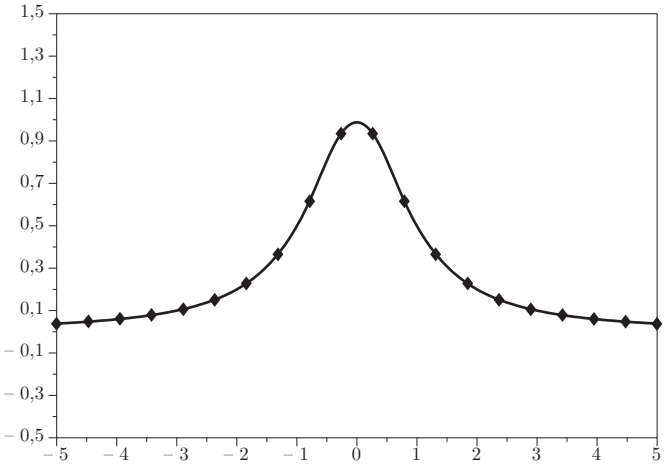


Figure 4.6 – Interpolation spline d'une fonction de Lorentz.

Le phénomène de Runge a disparu et l'interpolation est parfaitement régulière.

4.11. Interpolation à deux ou plusieurs dimensions

Les algorithmes exposés ci-dessus se généralisent de façon plus ou moins laborieuse à deux ou plusieurs dimensions. La méthode des éléments finis, utilisée pour résoudre les équations aux dérivées partielles, fait un usage intensif de l'interpolation à plusieurs variables ; vous trouverez les meilleures introductions à l'interpolation à plusieurs dimensions dans les livres traitant des éléments finis. Le graphisme sur ordinateur (CAO, DAO) est un autre « consommateur » important d'algorithmes d'interpolation. Mentionnons pour terminer une application classique de l'interpolation à deux variables : le tracé de courbes de niveau. Soit $f(x, y)$ une fonction définie dans une région du plan. Nous pouvons représenter cette fonction comme une surface $z = f(x, y)$ (en fait comme une projection sur le papier ou l'écran de cette surface). Nous pouvons aussi dessiner la courbe du plan xOy définie par l'équation $f(x, y) = C$. Les bulletins météo comportent souvent des cartes avec des isobares, lieux des points où la pression atmosphérique prend une valeur donnée. Pour dessiner automatiquement une courbe de niveau, nous devons d'abord construire un tableau de valeurs de f correspondant à tous les couples possibles d'abscisses x_i et d'ordonnées y_j, donc en tous points d'un quadrillage couvrant la région intéressante. Nous repérons ensuite, sur les droites horizontales $x = x_i$ et sur les verticales $y = y_j$, les points pour lesquels $f = C$. Enfin, nous relions ces points entre eux par une interpolation bidimensionnelle.

4.12. Pour en savoir plus

Avant l'arrivée des ordinateurs, l'interpolation dans une table de valeurs équidistantes et le calcul des différences finies jouaient un rôle très important ; on peut retrouver ce formalisme astucieux dans les ouvrages plus anciens cités ci-dessous.

– M. Schatzman : *Analyse numérique, une approche mathématique*, ch. 4 (Dunod, Paris, 2001).

– R. Théodor : *Initiation à l'analyse numérique*, ch. 3 (Masson, Paris, 1994).

– courbes de Bézier :

  – `http://d.krauss.free.fr/documents/Transverses/Bezier/Simulation_Bezier/Simulation_Bezier.htm`

  – Courbes de Bézier et courbes B-spline : `http://www.irit.fr/` : voir la page personnelle de Loic Barthe.

  – historique complet sur le site de l'INSA de Toulouse : `http://www-gmm.insa-toulouse.fr` : voir la page personnelle de M. Rabut.

  – J.-P. Pouget, G. Demengel : *Modèle de Bézier, des B.Splines et des NURBS, mathématique des courbes et des surfaces* (Ellipses, Paris, 1998).

– H. Mineur : *Techniques de calcul numérique* (Bérenger, Paris, 1952).

– F.B. Hildebrand : *Introduction to numerical analysis* (McGrawHill, New York, 1974).

4.13. Exercices

Exercice 1

On veut que le polynôme du second degré $p(x) = ax^2 + bx + c$ interpole les points $(-2, -27)$, $(0, -1)$ et (1,0). Former les équations auxquelles obéissent les coefficients indéterminés a, b, c et les résoudre pour exprimer $p(x)$.

Exercice 2

L'indice de réfraction du polystyrène, mesuré à différentes longueurs d'onde (correspondant à des raies intenses du spectre du sodium), est donné dans la table suivante.

λ (Å)	4358	4861	5896	6563	7679
n	1,6174	1,6062	1,5923	1,5870	1,5812

a) Déterminer l'indice de réfraction pour une longueur d'onde de 5000 Å par interpolation linéaire.

b) Répondre à la même question au moyen d'une interpolation quadratique (à trois points).

Exercice 3

Démontrer que le polynôme élémentaire de Lagrange $\ell_i(x)$ peut aussi s'écrire

$$\ell_i(x) = \frac{\pi(x)}{(x - x_i)\pi'(x_i)}, \text{ avec } \pi(x) = \prod_0^n (x - x_i).$$

Exercice 4

On donne les pivots

x_i	0	3	1
f_i	1	2	3

Construire le polynôme de Lagrange du second degré $p(x)$ qui utilise ces noeuds et calculer $p(2)$.

Exercice 5

a) Construire une table de la fonction $\sqrt{x}$ et de ses cinq premières différences latérales ascendantes pour x = 1(0,1)2 (cette notation classique signifie que x varie de 1 à 2 par pas de 0,1), en conservant 4 chiffres après la virgule. Il est instructif d'écrire un programme pour construire cette table ; le calcul se réduit à des soustractions successives, mais la rédaction du code pour obtenir une présentation plaisante demande un peu d'attention.

b) Interpoler dans cette table pour trouver $\sqrt{1,24}$ et $\sqrt{1,86}$.

c) Sachant que $\sqrt{x_1} = 1,69$, déterminer x_1 par interpolation inverse. L'erreur d'arrondi sur les valeurs de la table est d'environ $5\,10^{-5}$ en plus ou en moins. Quel ordre d'interpolation devez vous utiliser pour que l'erreur de troncation ne soit plus grande ?

Exercice 6

a) Démontrer que les différences latérales d'ordre n d'un polynôme quelconque de degré n sont constantes et trouver leur valeur. Indication : montrer que l'on peut se contenter d'étudier $p_n \equiv x^n$. Que dire des différences d'ordre $n+1$? Les valeurs suivantes sont celles d'un polynôme de degré strictement inférieur à 6 : quel est donc ce degré ?

x	-2	-1	0	1	2	3
$p(x)$	-5	1	1	1	7	25

b) Déterminer complètement ce polynôme par la méthode des coefficients indéterminés.

c) Reconstituer le polynôme en utilisant les différences divisées de Newton.

Exercice 7

On donne les valeurs de la fonction $f(x) = 1/x$ pour quelques valeurs entières de la variable :

x	2	3	4	5
y	0,5	0,3333	0,25	0,2

et on demande d'estimer, par interpolation, la valeur de la fonction en $x = 3,5$.

a) Que donne l'interpolation linéaire entre 3 et 4 ?

b) Déterminer $f(3,5)$ par interpolation quadratique sur les pivots $x = 2, 3, 4$ puis sur les pivots $x = 3, 4, 5$.

c) Estimer $f(3,5)$ à l'aide d'une interpolation à 4 points.

d) Quelles sont les erreurs réelles de ces diverses approximations ?

e) Appliquer la formule générale de l'erreur d'interpolation et vérifier que les erreurs réelles sont bien comprises dans l'intervalle prévu.

f) Extrapoler pour obtenir des estimations de $f(1)$ et $f(6)$.

Exercice 8

a) Former le polynôme d'interpolation de Lagrange, $P(x)$, qui passe par les points (0,1),(2,5), (3,10) et (4,15).

b) Combien faut-il d'opérations pour écrire ce polynôme, puis pour le calculer en un point ? Généraliser au cas d'un polynôme de degré n.

c) Soit $\bar{P}(x) = 1 + 2x + x(x-2) - \frac{1}{4}x(x-2)(x-3)$. Vérifier que $\bar{P} = P$.

d) Mettre $\bar{P}$ sous une forme voisine de celle de Horner et décompter les opérations nécessaires à son calcul.

e) On voudrait construire le polynôme $Q(x)$ qui passe par les points (0,1),(2,5), (3,10) et (4,15) et aussi par (5,25). On pose $Q(x) = \bar{P}(x) + ax(x-2)(x-3)(x-4)$; déterminer le coefficient a.

Exercice 9

a) On dispose d'une table de la fonction $\cos x$, où l'argument est en degrés et l'intervalle tabulaire est de 1 minute d'angle, pour $0° \leq x \leq 90°$. Les valeurs de la fonction sont données avec 5 chiffres significatifs. Selon la convention habituelle, on suppose donc qu'elles sont entachées d'une erreur d'arrondi de ± 0.000005. On pratique une interpolation linéaire dans cette table. Donner une borne de l'erreur totale : erreur d'interpolation + erreur d'arrondi.

b) Question inverse de la précédente. Vous devez établir une table de $\sin x$, avec un intervalle tabulaire de h, sous la condition suivante : l'erreur totale pour une interpolation linéaire doit être inférieure à 10^{-6} sur tout l'intervalle $(0° \ldots 90°)$. Quelle valeur de h faut-il choisir, combien de chiffres significatifs doit comporter la table ?

Exercice 10

Mettre les polynômes d'interpolation de Lagrange et de Newton sous la forme de l'équation (4.2). Expliciter, dans chaque cas, les fonctions φ_i et les coefficients a_i.

Exercice 11

a) Construire le polynôme d'interpolation de Hermite à partir des valeurs d'une fonction et de sa dérivée en deux points, x_0 et x_1, avec $h \equiv x_1 - x_0$. Appliquer au calcul de $\sin 45^0$ à partir des valeurs de sinus et cosinus pour 30^0 et 60^0.

b) Utiliser l'interpolation de Hermite inverse (à deux points) pour résoudre l'équation $x = \cos x$, à partir des approximations $x_0 = 0, x_1 = 1$.

Exercice 12

Former les équations de définition de la spline cubique « naturelle » pour laquelle

$$s''(a) = m_0 = s''(b) = m_n = 0.$$

4.14. PROJETS

Projet 1. La forme « barycentrique » de l'interpolation de Lagrange

On considère le polynôme de Lagrange qui interpole la fonction $f(x)$ sur les $n+1$ pivots d'abscisses $x_i, \quad 0 \leq i \leq n$.

a) Démontrer la relation $\ell_i(x) = \pi(x)/[(x - x_i)\pi'(x_i)]$, où $\pi(x) = (x - x_0)(x - x_1)\dots(x - x_n)$ et ℓ_i est le polynôme élémentaire de Lagrange.

b) On pose $w_i = 1/\pi'(x_i)$. Les poids w_i sont des nombres, indépendants de x, que l'on peut calculer une fois pour toutes dès que l'on a choisi les pivots. Exprimer le polynôme d'interpolation de Lagrange, $L(x)$, en fonction des nombres x_i, w_i et $f_i = f(x_i)$ et de $\pi(x)$.

c) Vous vous souvenez que l'erreur d'interpolation est nulle si la fonction f est un polynôme de degré au plus égal à n (pourquoi ?). Nous choisissons le polynôme particulier $f \equiv 1$. Écrire $L(x)$ dans ce cas et en déduire une expression de $\pi(x)$ ne faisant intervenir que x et les constantes x_i, w_i.

d) Exprimer $L(x)$ en fonction de x, des x_i et w_i.

e) Particulariser la formule précédente pour le cas de pivots équidistants, $x_i = x_0 + ih$.

f) Vérifier que le polynôme d'interpolation est invariant si l'on multiplie les coefficients w_i par une constante. Déterminer ce facteur commun de telle manière que w_0 prenne la valeur 1.

Vérifier que L prend la forme

$$L(x) = \frac{\displaystyle\sum_{i=0}^{n} (-1)^i C_n^i \frac{f_i}{x - x_i}}{\displaystyle\sum_{i=0}^{n} (-1)^i C_n^i \frac{1}{x - x_i}}.$$

g) Écrire un programme fondé sur les considérations précédentes. Comment éviter une division par zéro lorsque $x \simeq x_i$? On donne un extrait de la table de $\sin x$, pour x en degrés :

x	$\sin x$
42	0,66913061
44	0,69465837
46	0,71933980
48	0,74314483

Interpoler dans la table, à l'aide du programme, pour quelques valeurs de x et vérifier le résultat.

h) Transposer le formalisme précédent à l'interpolation de Hermite.

Projet 2. Courbes de Bézier

Une fonction (ou courbe)de Bézier n'est pas à proprement parler un interpolant. Sa souplesse, sa régularité et sa stabilité (pas d'oscillations aux bords) la rendent cependant très utile pour approcher des formes géométriques. Une fonction de Bézier est définie implicitement par des points, appelés « points de contrôle ». La courbe de Bézier d'ordre n répond à la définition suivante

$$\boldsymbol{P}(u) \equiv \sum_{i=0}^{n} C_n^i u^i (1-u)^{n-i} \boldsymbol{p}_i \tag{4.32}$$

où u est un nombre réel compris entre 0 et 1, C_n^i est un coefficient du binôme et les $\boldsymbol{p}_i$ sont des vecteurs colonnes $\boldsymbol{p}_i = [x_i, y_i]^T$ représentant $n+1$ points distincts du plan. La fonction de u qui multiplie $\boldsymbol{p}_i$ est un polynôme de Bernstein. Le point dont le rayon vecteur est $\boldsymbol{P}$ (le barycentre des $\boldsymbol{p}_i$) décrit la courbe de Bézier lorsque u varie. En pratique, on utilise essentiellement les fonctions de Bézier cubiques, qui dépendent de 4 points de commande.

a) Écrire les équations paramétriques d'une courbe de Bézier cubique (soit $x(u), y(u)$).

b) À quels points correspondent les valeurs $u = 0$ et $u = 1$?

c) La courbe passe-t-elle par les points $\boldsymbol{p}_1$ ou $\boldsymbol{p}_2$?

d) Construire la tangente à la courbe aux points de paramètre 0 et 1.

e) Comment faut-il disposer les points $\boldsymbol{p}_4 \ldots \boldsymbol{p}_7$ pour que la courbe qu'ils définissent se raccorde régulièrement à la précédente ?

Projet 3. Algorithme de Neville

On connaît les valeurs d'une fonction f pour $n+1$ valeurs $x_0, x_1, \ldots, x_n$ de la variable. On appelle $P_{a,b,\ldots,m}(x)$ le polynôme qui interpole f sur les points $x_a, x_b, \ldots, x_m$ où a, b et m sont des indices appartenant à l'intervalle $[0, n]$.

a) Démontrer la relation

$$P_{a,\ldots,m} = \frac{1}{x_i - x_j}[(x - x_j)P_{a,\ldots,j-1,j+1,\ldots,m} - (x - x_i)P_{1,\ldots,i-1,i+1,\ldots,m}]$$

avec $i, j \in \{a, b, \ldots, m\}$. En d'autres termes, le polynôme qui interpole sur l'ensemble de points $[x_a, \ldots, x_m]$ s'exprime simplement en fonction des deux polynômes qui interpolent respectivement sur les mêmes points sauf x_j et sur les mêmes points sauf x_i.

b) Si l'on pose $P_i \equiv f(x_i)$, on peut construire un tableau

x_0	P_0			
x_1	P_1	$P_{0,1}$		
x_2	P_2	$P_{1,2}$	$P_{0,1,2}$	
x_3	P_3	$P_{2,3}$	$P_{1,2,3}$	$P_{0,1,2,3}$

dans lequel chaque élément se calcule par la formule précédente à partir de ceux situés à sa gauche. Calculer effectivement les valeurs figurant dans le tableau pour $x = 3.5$ et les $x_i, f(x_i) = P_i$ de l'exercice 6. Ce procédé (appelé algorithme de Neville) permet de calculer numériquement le polynôme de Lagrange par approximations successives et de s'arrêter quand la valeur obtenue varie assez peu.

c) Comme la méthode de Neville remplit un tableau à deux dimensions, il est possible de redéfinir les éléments pour qu'ils ne dépendent que de deux indices. On désigne par $Q_{i,j}$ le polynôme de degré j qui interpole sur les $j + 1$ points $x_{i-j}, x_{i-j+1}, \ldots, x_{i-1}, x_i$, soit

$$Q_{i,j} \equiv P_{i-j,i-j+1,\ldots,i-1,i}.$$

Vérifier que la relation de récurrence s'écrit maintenant

$$Q_{i,j} = \frac{1}{x_i - x_{i-j}}[(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}]$$

alors que le tableau des approximations successives devient

x_0	$Q_{0,0}$			
x_1	$Q_{1,0}$	$Q_{1,1}$		
x_2	$Q_{2,0}$	$Q_{2,1}$	$Q_{2,2}$	
x_3	$Q_{3,0}$	$Q_{3,1}$	$Q_{3,2}$	$Q_{3,3}$

d) Écrire un programme pour former le tableau des $Q_{i,j}$ à partir des valeurs de x et de f.

Chapitre 5

Résolution d'équations non linéaires

Nous exposons, dans ce chapitre, quelques méthodes de recherche des racines d'une équation non-linéaire, de la forme $f(x) = 0$. Résoudre l'équation $f = 0$, chercher les zéros de f ou les pôles de $1/f$ sont des expressions équivalentes. Ce genre de problème se rencontre souvent, qu'il s'agisse de déterminer le point de fonctionnement d'une diode d'après sa caractéristique, la concentration d'une espèce chimique dans un mélange réactionnel ou la fréquence de coupure d'un filtre électrique. Nous n'envisagerons ici que des fonctions réelles. On peut être amené à chercher toutes les solutions de $f = 0$, ou seulement quelques unes ou une seule, la plus petite par exemple. Un cas particulier important est celui où f est un polynôme ; nous savons alors que les racines sont réelles ou deux à deux complexes conjuguées, en nombre égal au degré du polynôme.

Il est rare que nous puissions écrire une solution analytique de l'équation $f = 0$; en fait, cela ne se produira que pour les polynômes de degré inférieur ou égal à 4 ou pour quelques fonctions simples. En conséquence, toutes les méthodes générales de recherche de racine sont des méthodes itératives ; partant d'une solution approchée, nous obtiendrons une suite d'approximations de plus en plus précises. Nous devrons donc nous préoccuper de la convergence de la méthode et de la vitesse de convergence. Il nous faudra définir un critère d'arrêt des itérations et prévoir le rôle des erreurs d'arrondi inévitables dans tout calcul numérique. Insistons sur le fait qu'aucune méthode connue ne fonctionne vraiment « en aveugle » : on doit toujours avoir une connaissance au moins approximative de l'emplacement de la racine. Nous vous recommandons vivement de tracer le graphe de la fonction pour avoir une idée du nombre et de la position des zéros, ce qui, avec une calculette moderne, n'est pas bien difficile.

Les algorithmes de recherche d'une racine reposent plus ou moins directement sur le théorème suivant.

> **Théorème des valeurs intermédiaires** – Soit f une fonction continue dans $[a, b]$; alors, pour tout réel F compris entre $f(a)$ et $f(b)$, il existe au moins un réel c de $[a, b]$ tel que $f(c) = F$.

En d'autres termes, si $f(a)$ et $f(b)$ sont de signes contraires, la fonction continue f présente au moins un zéro entre a et b.

5.1. Méthode de bissection ou de dichotomie

Nous nous intéressons à une fonction f continue sur l'intervalle $I = [a, b]$ et telle que $f(a)f(b) < 0$; en d'autres termes, nous avons « encadré » la racine. Il découle de ces hypothèses que f s'annule au moins une fois dans I et c'est ce zéro que nous souhaitons localiser précisément. Voici un fragment de programme (en Scilab) correspondant à l'algorithme de bissection.

Listing 5.1 – Recherche d'une racine par bissection

```
NITMAX = 20;TOL = 1E-2;
deff("y = fn(x)","y =  x.^2 - cos(x) ")
nit = 0;
xg = input("valeur a gauche: " );
xd = input("valeur a droite: ");
fg = fn(xg); fd = fn(xd);
disp(fd,xd,fg,xg)
while( (nit < NITMAX) & (abs(xd - xg) > TOL) )
    xm = 0.5*(xg+xd);
    xi(nit+1) = xm;
    fm = fn(xm); yi(nit+1) = fm;
    if(fm*fg < 0)
       xd = xm; fd = fm;
    else
       xg = xm; fg = fm;
    end
    mprintf("%d  %10.6f  %10.6f\n",nit,xg,xd);
    nit = nit+1;
end
xm = 0.5*(xg+xd);
```

Le principe de la méthode est facile à comprendre. Nous calculons la valeur de la fonction au milieu (x_m) de l'intervalle $[x_g, x_d]$ et nous observons le changement de signe : s'il se produit entre x_g et x_m, c'est que la racine se trouve dans cet intervalle ; dans ce cas, nous redéfinissons la borne droite de l'intervalle et nous recommençons ; le raisonnement est semblable si le changement de signe se produit entre x_m et x_d ; dans ce cas, c'est x_m qui devient la nouvelle borne gauche. A chaque itération, l'intervalle qui contient la racine voit sa longueur divisée par 2.

Comme dans toute méthode itérative, nous avons dû définir un critère d'arrêt, ici le fait que la longueur du segment $[x_g, x_d]$ devienne inférieure à la constante TOL. Il se pourrait que cette condition ne soit pas remplie assez vite : c'est pourquoi nous surveillons aussi le nombre d'itérations et nous arrêtons le calcul si nit dépasse $NITMAX$. Nous aurions aussi bien pu choisir la condition $|f(x_m)| < SEUIL$ comme critère d'arrêt.

Cette ébauche est incomplète : rien n'est prévu pour le cas où $f(x_m) = 0$ exactement. Pouvez-vous prévoir ce qui arrivera si $a > b$ ou si l'intervalle I contient plusieurs racines ?

Cependant, une fois amélioré, l'algorithme de dichotomie présente de nombreux avantages. La convergence est certaine dès lors que les conditions précédentes sont remplies ; la programmation est très simple ; le calcul de f n'a pas besoin d'être très précis, parce que nous utilisons en fait le signe de la fonction et non sa valeur. Le désavantage est que la convergence est assez lente ; comme l'intervalle où peut se trouver la solution est divisé par deux à chaque itération, nous disons que la convergence est approximativement linéaire.

La figure 5.1 montre un exemple de recherche de racine de l'équation $x^2 = \cos x$ réalisé par dichotomie sous Scilab.

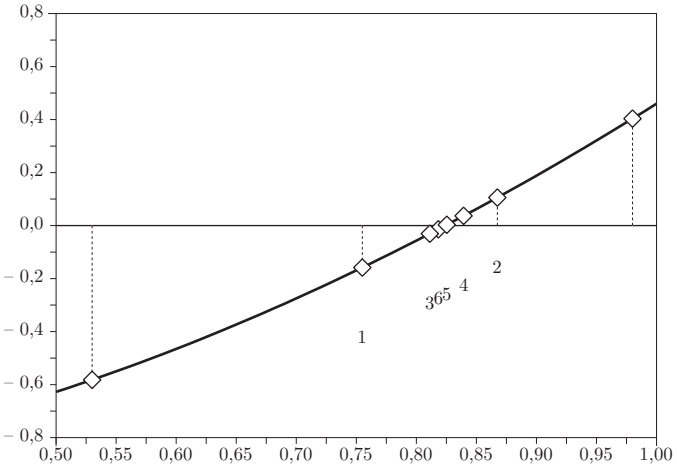


Figure 5.1 – Résolution de $x^2 = \cos x$ par bissection.
$TOL = 0,01$; $x_{0g} = 0,53$; $x_{0d} = 0,98$. Les valeurs successives de x_m sont numérotées.

5.2. Méthode « Regula falsi » ou des parties proportionnelles

Ces mots latins (règle des fausses positions, une expression qui date du 17ème siècle) désignent une variante de la méthode de dichotomie qui utilise une meilleure estimation de la nouvelle abscisse (x_m pour la bissection).

À chaque itération de l'algorithme précédent, nous connaissons deux abscisses, x_g et x_d, qui encadrent la racine inconnue x^*. Les nombres $y_g = f(x_g)$ et $y_d = f(x_d)$ sont donc de signes contraires si bien que les deux points $\mathrm{G}(x_g, y_g)$ et $\mathrm{D}(x_d, y_d)$ sont situés de part et d'autre de l'axe des x. La corde GD coupe donc cet axe en un point $\mathrm{M}(x_m, 0)$, lequel sera « assez proche » de x^* ; nous choisirons x_m comme nouvelle borne de l'intervalle contenant la racine, borne gauche si le changement de signe de f se produit entre x_m et x_d, borne droite dans le cas contraire. Il nous reste à déterminer

l'abscisse x_m :

$$x_m = x_g - y_g \frac{x_g - x_d}{y_g - y_d} = x_d - y_d \frac{x_g - x_d}{y_g - y_d}.$$

L'une de ces expressions devra remplacer la ligne 9 du listing 5.1.

La convergence est souvent plus rapide qu'avec la méthode de bissection. Nous avons choisi comme exemple (voir figure 5.2) la même équation qu'au paragraphe précédent, qui fait intervenir une fonction à variation régulière pour laquelle la convergence est très rapide. Nous en avons profité pour modifier le critère d'arrêt : le calcul s'interrompt dès que deux estimations successives de la racine sont « assez voisines » :

$$|x_m^{(n)} - x_m^{(n-1)}| < TOL.$$

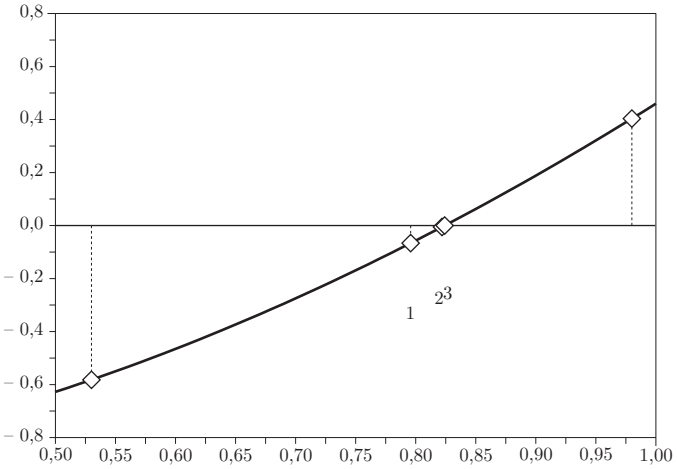


Figure 5.2 – Résolution de $x^2 = \cos x$ par *regula falsi* ; $TOL = 0,001$; $x_{og} = 0,53$; $x_{0d} = 0,98$.

5.3. Méthode du point fixe ou d'itération

Nous nous proposons de résoudre l'équation

$$x = g(x). \tag{5.1}$$

Toute solution de cette équation s'appelle un point fixe de l'application g (l'image de x est x lui-même). Nous allons procéder par itérations successives à partir d'une estimation $x^{(0)}$ de la solution :

$$x^{(n+1)} = g(x^{(n)}). \tag{5.2}$$

L'équation (5.2) admet une interprétation géométrique simple. La solution x^* de (5.1) est le point d'intersection de la première bissectrice (droite d'équation $y = x$) et de la courbe d'équation $y = g(x)$.

L'ordonnée $g(x^{(0)})$ est la nouvelle abscisse $x^{(1)}$; le point $(x^{(1)}, 0)$ se déduit du point $(x^{(0)}, g(x^{(0)}))$ par une symétrie orthogonale par rapport à la droite $y = x$. Cette construction se répète pour $x^{(2)}, x^{(3)}, \ldots$ En essayant de résoudre (5.1) pour diverses fonctions g, vous verrez que l'on peut rencontrer deux dispositions de la suite $x^{(0)}x^{(1)}, \ldots$, correspondant soit à un point fixe attractif, soit à un point fixe répulsif, comme le montrent les figures 5.3 et 5.4.

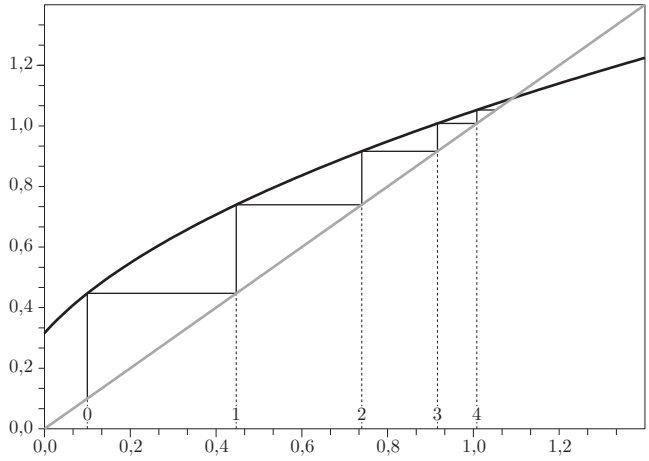


Figure 5.3 – Itération convergente.

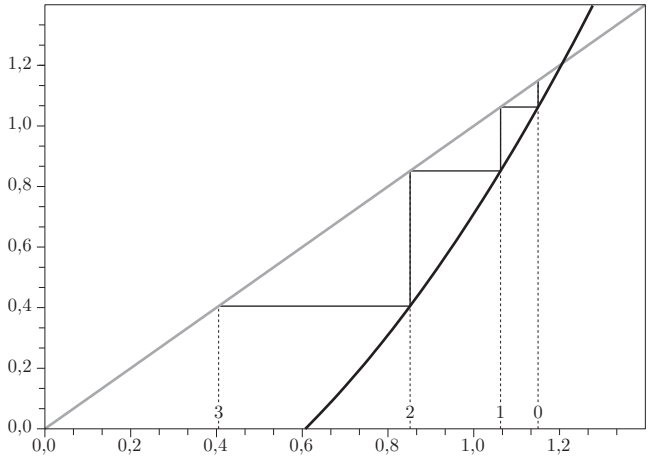


Figure 5.4 – Itération divergente.

D'où la question : à quelle(s) condition(s) la suite des $x^{(n)}$ converge-t-elle, et vers quelle valeur ? Le raisonnement, simple dans le cas présent, peut être considéré comme représentatif de nombreuses études de convergence. Nous supposons que la fonction g et sa dérivée g' sont continues dans un intervalle $I = [a, b]$ entourant la racine x^*. De plus, g est telle que $a \leq g(x) \leq b$ si $x \in I$. Cette condition implique que tous les $x^{(k)}$ appartiennent à I si l'approximation initiale $x^{(0)}$ appartient à cet intervalle.

La racine vérifie (5.1) :

$$x^* = g(x^*).$$

Introduisons l'erreur à l'itération k

$$e_k \equiv x^{(k)} - x^*.$$

L'erreur à l'étape $k+1$ s'exprime simplement en fonction de e_k

$$e_{k+1} = x^{(k+1)} - x^* = g(x^{(k)}) - g(x^*).$$

D'après le théorème des accroissements finis, le second membre s'écrit

$$g(x^{(k)}) - g(x^*) = (x^{(k)} - x^*)g'(\xi_k)$$

où ξ_k est compris entre x^* et $x^{(k)}$. La relation cherchée s'écrit

$$e_{k+1} = g'(\xi_k)e_k. \tag{5.3}$$

Pour que l'itération converge, il faut que l'erreur tende vers zéro lorsque $k \to \infty$, ce qui sera assuré si $|g'(\xi_k)| < 1$ quel que soit k. D'où le théorème

Théorème – Soit une fonction g continûment dérivable sur $[a,b]$ et telle que $g(x) \in [a,b]$ si $x \in [a,b]$. Soit encore M telle que

$$M = \sup_{a \le x \le b} |g'(x)| < 1.$$

Pour tout $x^{(0)} \in [a,b]$, la suite des itérés $x^{(0)}, x^{(1)}, \ldots, x^{(n)}, \ldots$ converge vers la solution de l'équation $x = g(x)$. On a de plus

$$\lim_{n\to\infty} \frac{x^* - x^{(n+1)}}{x^* - x^{(n)}} = g'(x^*).$$

Vous voyez que l'erreur est multipliée par un facteur inférieur à un à chaque itération : on dit que la convergence est linéaire.

La méthode du point fixe est d'application assez générale car beaucoup d'équations peuvent se mettre sous la forme $x = g(x)$. L'exemple suivant remonte à l'Antiquité. L'équation $x^2 = a$ peut se mettre sous l'une des formes $x = a/x$ ou $x = \frac{1}{2}(x + a/x)$. Les itérations correspondantes sont encore utilisées pour approcher $\sqrt{a}$.

La figure 5.5 montre l'application de la méthode du point fixe à l'équation $x = \sqrt{\cos x}$; les conditions de convergence sont remplies dans [0,1] et, effectivement, la méthode converge vers $x^* = 0,824132$. Pour que la figure reste lisible, nous avons limité à 5 le nombre d'itérations. Sinon, nous aurions dû choisir un critère de convergence, comme $|x^{[n]} - x^{(n-1)}| < TOL$.

5.4. Méthode de Newton

La méthode de Newton s'applique à la résolution d'une équation de la forme $f(x) = 0$. Étant donnée une approximation $x^{(0)}$ de la racine, nous construisons la tangente à

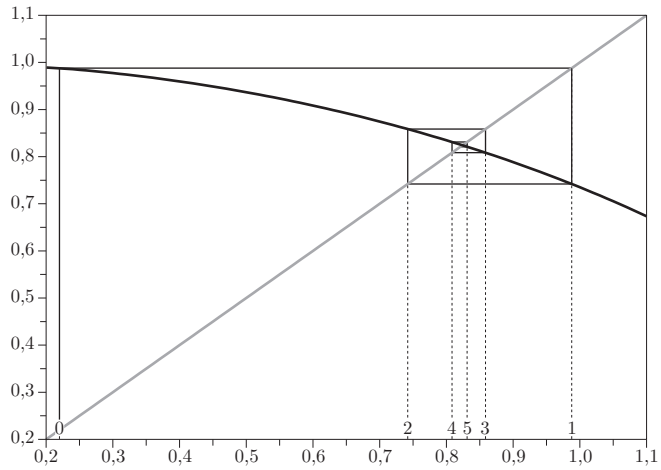


Figure 5.5 – Résolution de $x = \sqrt{\cos x}$ par itération.

la courbe d'équation $y = f(x)$ au point d'abscisse $x^{(0)}$; cette droite coupe l'axe horizontal en $x^{(1)}$; nous construisons une nouvelle tangente en cette abscisse, dont l'intersection avec l'axe des x nous donne $x^{(2)}$. Ce procédé est itéré jusqu'à convergence. L'application à l'équation $x^2 = \cos x$ est représentée figure 5.6. Nous sommes partis de $x^{(0)} = 0,22$ et avons effectué 5 itérations.

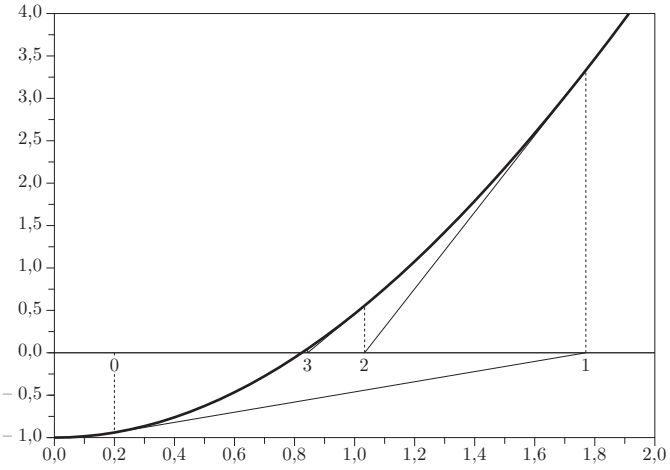


Figure 5.6 – Résolution de $x^2 = \cos x$ par la méthode de Newton.

Traduisons en formules la description précédente. L'équation de la droite de pente $f'(x^{(k)})$ passant par le point $(x^{(k)}, f(x^{(k)}))$ s'écrit $y - f(x^{(k)}) = f'(x^{(k)})(x - x^{(k)})$.

Cette droite coupe l'axe des x au point d'abscisse

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}. \tag{5.4}$$

Cette équation définit l'algorithme de Newton pour la résolution des équations non-linéaires. Vous constatez immédiatement que la méthode va diverger lorsque f' sera nulle ou même petite. En effet, si la pente de la courbe en x_k est faible, le point d'intersection (d'abscisse $x^{(k+1)}$) de la tangente avec l'axe sera très éloigné et l'algorithme a toutes les chances de se perdre.

Essayons maintenant d'établir la condition de convergence de façon précise ; la théorie devient très simple si l'on remarque qu'en posant

$$\phi(x) \equiv x - \frac{f(x)}{f'(x)},$$

on met l'équation (5.4) sous la forme d'une itération vers un point fixe

$$x^{(k+1)} = \phi(x^{(k)}).$$

La fonction $\phi(x)$ s'appelle parfois la fonction d'itération de la méthode de Newton. D'après le paragraphe précédent, nous savons que la convergence dépend de $\phi'(x)$; or

$$\phi'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}.$$

Désignons encore par x^* la racine cherchée ; elle vérifie $f(x^*) = 0$ et par conséquent

$$\phi'(x^*) = 0.$$

Comme nous supposons que f, f', f'' sont définies et continues dans la région qui nous intéresse, il existe un intervalle $I = [x^* - \delta, x^* + \delta]$ tel que

$$|\phi'(x)| \leq C < 1.$$

Il nous suffit donc de choisir $x^{(0)} \in I$ pour assurer la convergence de l'algorithme de Newton.

Pour estimer la vitesse de convergence, faisons un développement de Taylor de ϕ autour de x^* :

$$\phi(x^{(k)}) - \phi(x^*) = \frac{1}{2}(x^{(k)} - x^*)^2\phi''(\eta_k)$$

où η_k est compris entre x^* et $x^{(k)}$. Nous en déduisons que

$$e^{(k+1)} = \frac{1}{2}[e^{(k)}]^2\phi''(\eta_k).$$

Lorsque k croît, $x^{(k)}$ et η_k se rapprochent de x^*, ce que traduit l'expression

$$\lim_{k\to\infty} \frac{e^{(k+1)}}{[e^{(k)}]^2} = \frac{1}{2}\phi''(x^*).$$

Nous calculons $\phi''(x^*) = f''(x^*)/f'(x^*)$, d'où le résultat

$$\lim_{k\to\infty} \frac{e^{(k+1)}}{[e^{(k)}]^2} = \frac{f''(x^*)}{2f'(x^*)}. \tag{5.5}$$

Nous pouvons donc énoncer le théorème suivant.

> **Théorème** – Si f, f' et f'' sont continues dans un voisinage de x^* et si $f(x^*) = 0$ et $f'(x^*) \neq 0$, si $x^{(0)}$ est choisi assez près de x^*, alors la suite définie par (5.4) converge vers x^*, avec une vitesse de convergence donnée par (5.5) ; la convergence est dite quadratique.

5.5. Méthode de la sécante

La méthode de la sécante est une méthode « à deux points » : connaissant les approximations $x^{(k-1)}$ et $x^{(k)}$ de la racine, nous cherchons une meilleure approximation $x^{(k+1)}$. Le principe en est simple. La corde qui joint les points de coordonnées $[x^{(k-1)}, f(x^{(k-1)})]$ et $[x^{(k)}, f(x^{(k)})]$ coupe l'axe des x en un point d'abscisse $x^{(k+1)}$. Nous trouvons facilement l'expression de cette abscisse

$$x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}. \tag{5.6}$$

Ce procédé n'est pas très différent de la méthode de Newton. En effet, la fraction au second membre peut être considérée comme une approximation de $1/f'(x^{(k)})$. L'analyse de la convergence ne sera pas abordée ici ; contentons nous de citer le résultat.

> **Théorème** – Si f, f' et f'' sont continues dans un voisinage de x^* et si $f(x^*) = 0$ et $f'(x^*) \neq 0$, si $x^{(0)}$ et $x^{(1)}$ sont choisis assez près de x^*, alors la suite définie par 5.6 converge vers x^* avec une vitesse donnée par
>
> $$\lim_{k\to\infty} \frac{e^{(k+1)}}{[e^{(k)}]^p} = \left| \frac{f''(x^*)}{2f'(x^*)} \right|^{1/p}. \tag{5.7}$$
>
> L'ordre de convergence est $p = (\sqrt{5}+1)/2 \simeq 1,62$.

La convergence caractérisée par (5.7) est moins rapide que pour la méthode de Newton, mais le résultat peut être obtenu plus rapidement si le calcul de f' est très long.

5.6. Résolution de systèmes d'équations

Nous ne ferons qu'effleurer le sujet vaste et compliqué des systèmes d'équations non linéaires à plusieurs variables. Dans le cas de deux dimensions, nous cherchons les solutions du système :

$$\begin{cases} f(x,y) = 0, \\ g(x,y) = 0. \end{cases} \tag{5.8}$$

Chaque équation définit une courbe du plan (x, y) ; résoudre le système équivaut donc à chercher les points d'intersection de ces deux courbes. Nous recommandons vivement une étude graphique sommaire de ces deux équations avant toute recherche de racine par le calcul ; en effet, quel que soit l'algorithme utilisé, la convergence dépendra fortement de la qualité de l'approximation initiale.

Nous allons décrire une généralisation de la méthode de Newton. Supposons que f, g et leurs dérivées premières sont continues dans la région du plan qui nous intéresse. Nous supposons connue une approximation de la solution, soit (x_0, y_0). Nous pouvons toujours poser :

$$x = x_0 + u \quad ; \quad y = y_0 + v.$$

Substituons dans (5.8) et effectuons un développement de Taylor du premier ordre à deux variables :

$$f(x_0 + u, y_0 + v) = f(x_0, y_0) + uf_x + vf_y + \cdots = 0,$$

$$g(x_0 + u, y_0 + v) = g(x_0, y_0) + ug_x + vg_y + \cdots = 0,$$

où les dérivées partielles f_x, f_y, g_x et g_y sont calculées au point (x_0, y_0). Nous faisons l'hypothèse que les variations de f et g sont assez lentes pour que nous puissions négliger les termes d'ordre supérieur. En isolant les termes en u et v, nous trouvons

$$\left\{ \begin{array}{lcl} uf_x + vf_y & = & -f(x_0, y_0), \\ ug_x + vg_y & = & -g(x_0, y_0). \end{array} \right.$$

Les équations (5.9) constituent un système de deux équations linéaires à deux inconnues, u et v, dont la solution est immédiate. Il est maintenant facile d'imaginer l'algorithme de Newton à deux dimensions. Nous obtiendrons une approximation encore meilleure de la solution en itérant ce procédé

$$x_1 = x_0 + u \quad ; \quad y_1 = y_0 + v.$$

À l'étape suivante de l'itération, nous remplaçons (x_0, y_0) par (x_1, y_1) et ainsi de suite, jusqu'à satisfaire à un critère de convergence.

Ce formalisme se généralise facilement à un nombre quelconque d'équations ; il est alors commode d'utiliser une notation matricielle. Soit $\boldsymbol{f} \in \mathbb{R}^n$ un vecteur de coordonnées $f_i(\boldsymbol{r})$; nous supposons aussi que $\boldsymbol{r} \in \mathbb{R}^n$. Le système d'équations à résoudre s'écrit :

$$\boldsymbol{f} = 0.$$

Si $\boldsymbol{r}^{(0)}$ est une approximation de la solution, nous posons $\boldsymbol{r} = \boldsymbol{r}^{(0)} + \boldsymbol{u}$. La correction $\boldsymbol{u}$ est alors définie par l'équation

$$\boldsymbol{f}(\boldsymbol{r}^{(0)}) + \sum_1^n u_i \frac{\partial \boldsymbol{f}}{\partial u_i} = 0$$

ou encore, en définissant la "matrice jacobienne" $\boldsymbol{J}$, d'éléments $J_{ij} = \partial f_i / \partial u_j$:

$$\boldsymbol{u} = -\boldsymbol{J}^{-1} \boldsymbol{f}(\boldsymbol{r}^{(0)}).$$

Nous calculons ensuite $\boldsymbol{r}^{(1)} = \boldsymbol{r}^{(0)} + \boldsymbol{u}$ et, plus généralement

$$\boldsymbol{r}^{(n+1)} = \boldsymbol{r}^{(n)} - \boldsymbol{J}^{-1}\boldsymbol{f}(\boldsymbol{r}^{(n)}).$$

La convergence est rapide si le point de départ est proche de la solution ; dans le cas contraire, nous ne sommes pas assurés de voir l'algorithme converger.

Exemple – Soit le système de deux équations :

$$\begin{cases} f(x,y) &= x^2 + y^2 - 4 = 0, \\ g(x,y) &= e^x + y - 1 = 0. \end{cases}$$

Il est très simple de construire les deux courbes représentant les équations $f = 0$ et $g = 0$; elles se coupent en deux points voisins de $(1, -2)$ et $(-2, 1)$; il n'y a pas d'autre solution. Nous avons écrit un programme de résolution par la méthode de Newton à deux dimensions, que vous trouverez résumé ci-dessous.

Listing 5.2 – Méthode de Newton pour deux inconnues

```
//
//.....définition des fonctions f(x,y),
//.....g(x,y) et de leurs dérivées
//
tol = 1e-6; nitmax = 20;
a = input("abscisse initiale: ");
b = input("ordonnee initiale: ");
nit = 1; xx(1) = a; x = a; yy(1) = b; y = b;
u = 1; v = 1;
while ( ((abs(u) > tol) | (abs(v) > tol)) & nit < nitmax )
   delta = dfdx(x,y)*dgdy(x,y)-dfdy(x,y)*dgdx(x,y);
   u = (g(x,y)*dfdy(x,y)-f(x,y)*dgdy(x,y))/delta;
   v = (f(x,y)*dgdx(x,y)-g(x,y)*dfdx(x,y))/delta;
   nit = nit + 1
   x = x + u, y = y + v,
   xx(nit) = x; yy(nit) = y;
end
```

La figure 5.7 montre les itérations successives pour trois points de départ différents, $(3,5; -1,5)$, $(-1; -0,6)$ et $(3; 2)$. La convergence est rapide, bien que le point de coordonnées(x_i, y_i) ait tendance à « explorer » une assez grande région du plan. Comme toujours avec la méthode de Newton, le comportement dépend très fortement du point de départ. Pour certains systèmes, on peut montrer que les valeurs initiales qui conduisent à la convergence sont contenues dans une région du plan dont la frontière est une belle courbe fractale.

La fonction `fsolve` de Scilab est capable de résoudre un système d'équations non-linéaires. Il faut lui fournir comme arguments le vecteur initial $\boldsymbol{r}_0$, le nom effectif du vecteur $\boldsymbol{f}$ et celui du jacobien $\boldsymbol{J}$; les éléments de $\boldsymbol{f}$ et $\boldsymbol{J}$ doivent être définis auparavant. S'il y a convergence, la fonction renvoie le vecteur solution $\boldsymbol{x}$ et la valeur de $\boldsymbol{f}(\boldsymbol{x})$.

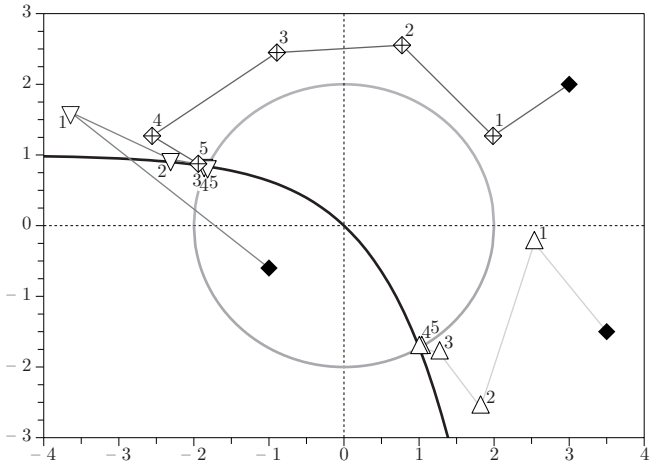


Figure 5.7 – Recherche des solutions d'un système non-linéaire, pour trois jeux de valeurs initiales (les losanges noirs).

5.7. Racines des polynômes

Soit p un polynôme de degré n à coefficients réels. On désigne parfois une équation comme $p(x) = 0$ sous le nom d'équation algébrique. Le théorème fondamental de l'algèbre énonce que p a n racines, réelles ou complexes conjuguées deux à deux. Pour de nombreuses applications, il est nécessaire de déterminer toutes les racines de p.

5.7.1. Division des polynômes

Dans ce paragraphe, nous détaillons la division des polynômes, un algorithme qui nous sera utile dans la suite. Étant donné un polynôme p de degré n (le dividende) et un polynôme b de degré $m < n$ (le diviseur), on sait trouver deux polynômes, le quotient q et le reste r tels que

$$p = bq + r, \text{ avec } \deg r < \deg b. \tag{5.9}$$

Remarquez que le degré de q est imposé $(n-m)$ mais ne joue aucun rôle dans la suite. Avec cette définition, la division des polynômes est très semblable à celle des nombres, pour laquelle le reste doit être inférieur au diviseur. Considérons maintenant un cas particulier, celui où le diviseur est de la forme $b(x) = x - \alpha$. Le reste est alors une constante (polynôme de degré < 1). Nous pouvons écrire

$$p(x) = (x - \alpha)q(x) + r$$

d'où nous tirons, en faisant $x = \alpha$,

$$p(\alpha) = r. \tag{5.10}$$

En d'autres termes, la valeur numérique de p en $x = \alpha$ est le reste de la division de p par $x - \alpha$. Ce résultat a une conséquence importante. Si a est un zéro de p, donc si $p(\alpha) = 0$, alors $r = 0$, ce qui signifie qu'un polynôme qui s'annule en $x = \alpha$ est divisible par $x - \alpha$. Nous avons déjà expliqué (chapitre 2) comment calculer r ou $p(a)$, mais nous reprendrons ce calcul plus bas.

Montrons comment obtenir une expression simple de la valeur numérique de la dérivée de p. Nous savons que

$$p'(x) = q(x) + (x - \alpha)q'(x)$$

et, en faisant à nouveau $x = \alpha$,

$$p'(\alpha) = q(\alpha). \tag{5.11}$$

Comme vous le verrez, il est facile de calculer $q(\alpha)$. Des équations analogues existent pour les dérivées d'ordre supérieur.

Nous allons maintenant construire le polynôme $q(x)$ par identification. Notons c_0, $c_1, \ldots, c_{n-1}$ les coefficients des puissances croissantes de x dans q. Ces nombres doivent vérifier, quel que soit x, la relation

$$a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = (x-\alpha)(c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_1 x + c_0) + r.$$

En identifiant les coefficients des puissances successives de x, nous trouvons

$$\begin{aligned} &a_n = c_{n-1}; \quad a_{n-1} = c_{n-2} - \alpha c_{n-1}; \quad \ldots \\ &a_1 = c_0 - \alpha c_1; \quad a_0 = r - \alpha c_0 \end{aligned}$$

soit encore, en inversant ces formules :

$$\begin{aligned} c_{n-1} &= a_n \\ c_{n-2} &= a_{n-1} + \alpha c_{n-1}, \\ \cdots &\quad \cdots \\ c_0 &= a_1 + \alpha c_1, \\ r &= a_0 + \alpha c_0. \end{aligned} \tag{5.12}$$

Vous voyez que r, qui est aussi la valeur numérique de $p(\alpha)$, s'obtient au moyen de $n-1$ multiplications. En fait la relation de récurrence précédente est équivalente à la règle suivante. Écrivons le polynôme p sous la forme

$$p(x) = (((\ldots(a_n x + a_{n-1})x + a_{n-2})x + \ldots + a_1)x + a_0, \tag{5.13}$$

puis faisons $x = \alpha$ avant d'effectuer les calculs. La parenthèse intérieure contient alors c_{n-2}, la suivante c_{n-3}, jusqu'à r. Vous avez reconnu la méthode de Horner. Donnons un exemple, bien qu'il fasse double emploi avec celui du chapitre 2.

Exemple – Nous cherchons la valeur de $p = 2x^5 - 71x^3 - 8x^2 + 12x + 3$ pour $x = 6$, ce qui revient à diviser p par $x - 6$. Il est commode de disposer le calcul comme ci-dessous. Seuls les coefficients des puissances décroissantes de x sont nécessaires, mais il faut les faire figurer tous, mêmes ceux qui sont nuls.

p	2	0	-71	-8	12	3
q	2	$6 \times 2 + 0 = 12$	$6 \times 12 - 71 = 1$	$6 \times 1 - 8 = -2$	$6 \times -2 + 12 = 0$	$6 \times 0 + 3 = 3$

Le reste est $r = 3$, le quotient est $q = 2x^4 + 12x^3 + x^2 - 2x$. La valeur numérique de $p'(6)$ est $q(6)$ qui s'obtient par le même procédé.

q	2	12	1	-2	0
p'	2	$6 \times 2 + 12 = 24$	$6 \times 24 + 1 = 145$	$870 - 2 = 868$	$6 \times 868 = 5208$

ce que vous pouvez vérifier par un calcul direct.

Lorsque l'on traite à la main un seul polynôme, les formes (5.12) et (5.13) sont équivalentes mais, lorsque l'on doit faire de nombreux calculs, il devient intéressant d'écrire un programme général. Nous donnons ci-dessous le squelette d'un programme qui calcule p et p' à partir du tableau `a` des coefficients de p et de la valeur de x.

```
p = a(n+1), dp = 0.0;                    1
for i = n:-1:1                           2
    dp = dp*x + p;                       3
    p = p*x+a(i);                        4
end                                      5
```

Lorsque la boucle se termine, `p` contient la valeur numérique du polynôme (accumulée selon la méthode de Horner) et `dp` celle de la dérivée.

Remarque : Vous vous souvenez que Scilab numérote les éléments d'un tableau à partir de 1, si bien que $a_n \Rightarrow$ `a(n+1)` et $a_0 \Rightarrow$ `a(1)`. De plus, l'incrément d'un compteur de boucle **doit** être précisé s'il est différent de $+1$ (ligne 2).

5.7.2. Séparation des racines

Avant de chercher les valeurs numériques des zéros d'un polynôme, il est commode d'avoir une idée de leur localisation. Si $a_0, a_1, \ldots, a_n$ est la suite des coefficients de $p(x)$ supposés réels (certains pouvant être nuls), nous appelons ν le nombre de changements de signe dans la suite des $\{a_i\}$ (en ignorant les coefficients nuls). Nous désignons par k le nombre de zéros réels positifs de $p(x)$, comptés selon leur multiplicité (une racine double compte pour 2, etc.). Nous pouvons alors énoncer la règle de Descartes :

$$k \leq \nu \text{ et de plus } \nu - k \text{ doit être pair.}$$

Exemple – Les coefficients de $x^6 - x - 1$ présentent un changement de signe ($\nu = 1$) ; k vaut donc 0 ou 1, mais la valeur 0 est à rejeter puisqu'alors $\nu - k = 1$. Ce polynôme a donc une racine réelle positive.

Nous pouvons, par le même procédé, trouver le nombre de racines réelles négatives : il suffit d'appliquer la règle de Descartes au polynôme $q(x) = p(-x)$. Le polynôme considéré admet un zéro réel négatif.

5.7.3. Suites de Sturm

Une localisation plus précise des racines d'un polynôme p de degré n peut être obtenue en construisant une « suite de Sturm » basée sur p. Commençons donc par définir ce qu'est une suite de Sturm.

Soit $p_0(x), p_1(x), \ldots, p_m(x)$ une séquence de polynômes ; elle constitue une suite de Sturm si

– Les zéros réels de p_0 sont simples.

– $p_1(\alpha)p_0'(\alpha) < 0$ si α est un zéro réel de p_0.

– Pour $k = 1, 2 \ldots, m-1$, $p_{k-1}(\alpha)p_{k+1}(\alpha) < 0$ si α est un zéro réel de p_k.

– Le dernier polynôme, p_m n'a pas de zéros réels.

Comment construire cette suite ? À l'aide de la relation de récurrence que nous allons exposer. Posons

$$p_0 \equiv p; \qquad p_1 \equiv p_0'$$

et

$$p_{k-1} = q_k p_k - p_{k+1} \text{ avec } \deg p_k > \deg p_{k+1}. \tag{5.14}$$

Si les p_k étaient des nombres entiers, vous auriez sûrement reconnu l'algorithme d'Euclide pour la construction du PGCD de p_0 et p_1. Le concept de PGCD se transpose facilement aux polynômes. Selon cette définition, p_{k+1} est, au signe près, le reste de la division de p_{k-1} par p_k. Comme le degré des polynômes décroît, l'algorithme s'arrête au bout de $m \leq n$ étapes :

$$p_{m-1} = q_m p_m, \quad p_m \neq 0.$$

Le dernier polynôme, p_m est un plus grand commun diviseur de p_0 et p_1. Comme $p = p_0$ n'a que des zéros simples, p_0 et $p_1 = p_0'$ n'ont pas de diviseurs communs (ne sont jamais nuls simultanément) et, par conséquent, p_m n'a pas de zéros réels. Si $p_k(\alpha) = 0$, la relation (5.14) implique que $p_{k-1}(\alpha) = -p_{k+1}(\alpha)$. Il pourrait se faire que $p_{k+1}(\alpha)$ soit nul ; mais alors, toujours d'après la relation de récurrence, tous les $p_k(\alpha)$ seraient nuls, y compris p_m, ce qui est contraire au résultat précédent. En tenant compte des définitions de p_0 et p_1, vous vous apercevez que les $\{p_k\}$ satisfont à toutes les propriétés qui définissent une suite de Sturm.

Remarque : Certains auteurs utilisent la définition $p_1 = -p_0'$, avec des résultats équivalents.

Énonçons maintenant le théorème de Sturm :

> **Théorème** – Le nombre de racines réelles de $p \equiv p_0$ dans l'intervalle $a \leq x < b$ est égal à $w(b) - w(a)$ où $w(x)$ est le nombre de changements de signe dans la suite $p_0(x), p_1(x), \ldots, p_m(x)$ au point x.

Nous ne donnerons pas ici la démonstration, simple mais assez longue : il faut examiner en détail les changements de signe de chaque polynôme au voisinage des zéros de l'un d'eux.

Exemple – Soit $p(x) = x^5 - x^3 - x - 1$. La suite de Sturm de p est

$$\begin{aligned}
p_0(x) &= x^5 - x^3 - x - 1,\\
p_1(x) &= 5x^4 - 3x^2 - 1,\\
p_2(x) &= \frac{2}{5}x^3 + \frac{4}{5}x + 1,\\
p_3(x) &= 13x^2 + \frac{25}{2}x + 1,\\
p_4(x) &= -\frac{385}{338}x - \frac{174}{169},\\
p_5(x) &= -\frac{47827}{148225}.
\end{aligned}$$

Il est permis de multiplier ou de diviser chaque p_i par une constante positive avant chaque division, ce qui peut permettre de faire disparaître des coefficients compliqués. Cependant, ces calculs sont en général trop laborieux pour être conduits à la main. Maple dispose de l'instruction `sturmseq(p,x)` qui construit la suite de Sturm de $p(x)$; il existe aussi `rem(a,b,x)` qui forme le reste de la division de $a(x)$ par $b(x)$. Scilab comporte la fonction `[r,q] = pdiv(a,b)`, laquelle forme le reste (r) et le quotient (q) de la division de a par b. Voici, en résumé, le même calcul que précédemment, fait avec Scilab. Nous définissons le polynôme par

```
-->x = poly(0,"x");
-->p = x^5-x^3-x-1;
```

puis nous calculons sa dérivée :

```
-->p1 = derivat(p)
          2    4
  - 1 -3x + 5x
```

et nous formons la suite des restes ;

```
-->[p2,q] = pdiv(p,p1); p2 = -p2
p2  =
                   3
   1 + 0.8x + 0.4x
-->[p3,q] = pdiv(p1,p2); p3 = -p3
p3  =
                   2
   1 + 12.5x + 13x
-->[p4,q] = pdiv(p2,p3); p4 = -p4
p4  =
 - 1.0295858 - 1.1390533x
-->[p5,q] = pdiv(p3,p4); p5 = -p5
p5  =

 - 0.3226649
```

Appliquons le théorème de Sturm, avec $m = 5$. Nous ne considérons que les valeurs $x = -\infty, 0, \infty$; nous construisons un tableau de signes :

x	$-\infty$	0	∞
p_0	$-$	$-$	$+$
p_1	$+$	$-$	$+$
p_2	$-$	$+$	$+$
p_3	$+$	$+$	$+$
p_4	$+$	$-$	$-$
p_5	$-$	$-$	$-$
$w(x)$	4	2	1

La suite présente quatre changements de signe lorsque x est très grand et négatif, deux changements pour $x = 0$ et un seul pour x grand et positif. Il y a donc deux zéros dans l'intervalle $[-\infty, 0[$ et un dans $[0, \infty[$. Les valeurs sont fournies par `roots(p)` et valent environ -1, $-0,82$ et $1,38$.

5.7.4. La méthode de Newton pour les polynômes

Nous disposons d'un algorithme pratique de calcul des valeurs numériques d'un polynôme et de sa dérivée : c'est tout ce qu'il faut pour mettre en oeuvre la méthode de Newton. En voici un exemple : la recherche du zéro de $p = x^3 - x^2 + 2x + 5$ voisin de -1. Les coefficients ont été rangés en colonne pour gagner de la place.

x	p	q	p'	p/p'
-1	1	1	1	
	-1	-2	-3	
	2	4	**7**	
	5	1		
				$0,142857$
$-1,142857$	1	1	1	
	-1	$-2,142857$	$-3,285714$	
	2	$4,448979$	**8,204081**	
	5	$-0,084546$		
				$-0,010305$
$-1,132552$	1	1	1	
	-1	$-2,132552$	$-3,265104$	
	2	$4,415226$	**8,113126**	
	5	-0.000473		
				$-0,000058$
$-1,132494$				

Les nombres soulignés sont les valeurs de p, alors que les valeurs de p' figurent en caractères gras. Vous constatez que la convergence (caractérisée par le terme correctif (p/p')) est très rapide. Il est possible d'étendre cet algorithme aux racines complexes ou aux polynômes à coefficients complexes.

p est de degré 3 et admet donc 3 zéros ; comment pourrons-nous trouver les deux racines qui nous manquent ? Très simplement : il suffit de diviser le polynôme de départ par $x + 1,132494$. La division doit se faire exactement, puisque $-1,132494$ est racine de $p(x)$; le polynôme quotient est ici du second degré et ses zéros s'obtiennent sans problème. Ce procédé s'appelle la déflation.

Il est recommandé d'appliquer la déflation dans le cas général d'un polynôme $p(x)$ de degré n. Dès que nous avons déterminé une racine x^* (par la méthode de Newton par exemple), nous divisons $p(x)$ par $x - x^*$ pour obtenir un polynôme de degré $n - 1$. Nous répétons cette démarche jusqu'au degré 2 ou 1. Si n est un tant soit peu grand, les erreurs d'arrondi risquent, en s'accumulant, de gâcher ce programme : le calcul doit être mené avec toute la précision possible.

5.7.5. SCILAB ET LES POLYNÔMES

Bien qu'il ne soit pas destiné au calcul symbolique, le logiciel Scilab permet de manipuler facilement des polynômes. Étant donné un vecteur à n éléments (par exemple `v=[1,2,3,4]`) nous pouvons définir un polynôme par ses coefficients

```
-->p2 = poly(v,"x","coeff")
p2  =
                2    3
  1 + 2x + 3x + 4x
```

ou par l'intermédiaire de ses zéros

```
-->p1 = poly(v,"x","roots")
p1  =
                    2     3   4
   24 - 50x + 35x - 10x + x
```

ou encore directement (sans utiliser `v`)

```
-->x = poly(0, "x");
-->p3 = 1+x+2*x^2+3*x^3+4*x^4
p3  =
                2    3    4
   1 + x + 2x + 3x + 4x
```

La valeur numérique de l'un de ces objets s'obtient comme ceci

```
-->horner(p2,-1)
  - 2.
```

Le calcul de la dérivée est tout aussi rapide :

```
-->dp3 = derivat(p3)
dp3  =
                2     3
   1 + 4x + 9x + 16x
```

La fonction `degree(p2)` renvoie le degré de $p2$, ici 3, `coeff(p3)` renvoie le vecteur des coefficients de $p3$. Le calcul des racines est tout aussi facile :

```
-->w = roots(p3)
w  =
    0.2166768 + 0.6157657i
    0.2166768 - 0.6157657i
  - 0.5916768 + 0.4864288i
  - 0.5916768 - 0.4864288i
```

5.7.6. Condition du problème

Listing 5.3 – calcul du polynome de Wilkinson

```
function fn = p(x)
fn = (x-1).*(x-2).*(x-3).*(x-4).*(x-5).*(x-6).*(x-7)..
          .*(x-8).*(x-9).*(x-10).*(x-11).*(x-12)
endfunction
function fn = q(x,k)
fn = p(x) + x.^11/2^k
endfunction
x = 0.5:0.05:12.5;
xset("window",0),xbasc(0)
plot2d(x',[p(x)',q(x,19)',q(x,18)',q(x,17)'],..
   rect=[0,-2e6,13,2e6], style = [1,4,5,6],..
   axesflag=5,nax=[0,14,0,3],frameflag=5)
```

En expérimentant avec des polynômes de degrés de plus en plus élevés, vous vous rendrez compte que le calcul des zéros est très sensible aux erreurs d'arrondi ; on dit que le problème de recherche des racines est « mal conditionné ». Un exemple classique est le polynôme de Wilkinson. L'idée consiste à partir d'un polynôme dont les racines sont évidentes

$$p(x) = \prod_{i=1}^{N} (x - i)$$

et à le perturber en lui ajoutant une très petite contribution en x^{N-1}. Comme le montre le programme Scilab (listing 5.3) et la figure 5.8, les racines évoluent très rapidement et plusieurs deviennent complexes conjuguées deux à deux lorsque la perturbation $(x^{11}/2^k)$ augmente.

C'est cette grande sensibilité à une petite modification d'un coefficient qui constitue le « mauvais conditionnement ».

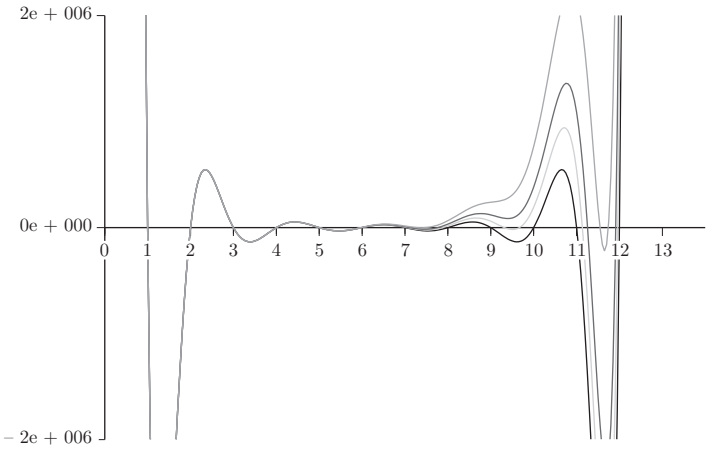


Figure 5.8 – Polynômes perturbés de Wilkinson.

5.8. Pour en savoir plus

– R. Théodor : *Initiation à l'analyse numérique*, ch. 3 (Masson, Paris, 1994).

– Polycopiés des cours d'analyse numérique de MM. E. Hairer et G. Wanner : ch. 6, méthodes itératives, équations non-linéaires :
`http://www.unige.ch/~hairer/polycop.html`

– M. Schatzman : *Analyse numérique, une approche mathématique*, ch. 9 (Dunod, Paris, 2001).

– R. Herbin : *Cours d'analyse numérique (L3)*, ch. 2, systèmes non-linéaires :
`http://www.cmi.univ-mrs.fr/~herbin/`

Il existe de nombreux algorithmes de recherche de racines qui n'ont pas été mentionnés dans le texte. Vous pourrez trouver les pages correspondantes sur le Web en partant du nom de leurs auteurs comme Müller, Brent ou Jenkins et Traub.

5.9. Exercices

Exercice 1

L'équation

$$x - 0,2 \sin x = 0,5$$

admet une racine dans $[0, 1]$. Trouvez cette racine en utilisant successivement les méthodes suivantes, après vous être assurés que la convergence était possible :

a) bissection ;

b) « regula falsi » ;

c) point fixe ;

d) Newton ;

e) sécante.

Exercice 2

Soit la fonction

$$y = x + e^{-kx^2} \cos x$$

qui admet un zéro dans l'intervalle $[-1, 0]$.

a) Utiliser la méthode de Newton pour calculer ce zéro, avec une erreur absolue sur x inférieure à 10^{-3} ; on donne $k = 1$ et x_0 (valeur initiale de x) $= 0$. On présentera les résultats sous la forme d'un tableau à quatre colonnes : numéro de l'itération, valeur de x, valeur de $y(x)$, valeur de $y'(x)$.

b) On choisit maintenant $k = 50$ et $x_0 = 0$. Faire 4 itérations de l'algorithme de Newton et expliquer le comportement de la suite x_i. (c) Déterminer la racine pour $k = 50$ et $x_0 = -0.1$.

Exercice 3

La fonction $\operatorname{cotg} x - x$ admet un zéro dans l'intervalle $[0, 5...1, 5]$.

a) Pourrait-on calculer cette racine par la méthode du point fixe ? Justifier votre réponse.

b) Localiser ce zéro, à 0,1 près, par la méthode de bissection (dichotomie).

c) Trouver précisément ce zéro par la méthode de Newton ; on arrêtera les itérations lorsque deux approximations successives différeront de moins de 0,0001.

Exercice 4

Trouver, par la méthode de votre choix, les trois plus petites solutions positives de l'équation

$$\tan x = x.$$

Exercice 5

Un épargnant dépose au début de chaque année, et ce pendant N_1 années, une somme de P_1 euros sur un compte. Ces versements rapportent des intérêts composés, au taux annuel de r. Ensuite, au début des années $N_1 + 1, \ldots, N_1 + N_2$, il retire une pension de P_2 euros. Son compte est exactement à zéro après le dernier retrait. On admet la relation

$$P_1 \left[(1 + r)^{N_1} - 1\right] = P_2 \left[1 - (1 + r)^{-N_2}\right].$$

Sachant que $N_1 = 30, N_2 = 20, P_1 = 2000, P_2 = 8000$, déterminer r.

Exercice 6

Selon un modèle simple, les énergies possible d'un électron dans une molécule diatomique sont données (pour des variables sans dimensions) par les solutions de l'équation

$$\operatorname{th} x = \frac{x}{\lambda - x}.$$

Pour quelles valeurs de λ cette équation admet-elle des racines réelles ? Déterminer la racine pour $\lambda = 2$.

Exercice 7

On donne le polynôme $p(x) = x^3 + x^2 - 3x - 3$.

a) Calculer $p(1), p(2)$ et $p(1,5)$ par la méthode de Horner.

b) Trouver la racine voisine de $x_0 = 2$ par la méthode de Newton.

Exercice 8

Trouver une racine, proche de $x_0 = 0$, de l'équation $x^4 - 9x^2 - 12x - 4 = 0$.

Exercice 9

La division d'un polynôme $p(x)$ par $x - \alpha$ est symbolisée par la relation

$$p(x) = (x - \alpha)q(x) + r.$$

En dérivant plusieurs fois, trouver les valeurs de $p'(\alpha), p''(\alpha)$ et $p'''(\alpha)$ en fonction de $q(\alpha), q'(\alpha)$ ou $q''(\alpha)$. Application. On donne le polynôme

$$p(x) = 2x^3 - 8x^2 + 10x - 4.$$

Évaluer $p(1), p'(1), p''(1), p'''(1)$.

Exercice 10

Le polynôme $p = x^5 - 2x^4 - 3x^3 - x^2 + 2x + 3$ admet la racine « évidente » $x = 1$. Diviser p par $x - 1$ pour obtenir $q(x)$, qui admet, lui aussi, une racine « évidente » $x = \beta$. Construire le polynôme $r(x) = q(x)/(x - \beta)$. $r(x)$ possède un zéro « simple » que l'on peut trouver par tâtonnement. Procéder à une nouvelle division pour former $s(x)$, un trinôme du second degré qui présente deux racines complexes conjuguées.

Exercice 11

Le polynôme $f(x) = x^3 - 3x + 2$ a une racine double en $x = 1$. Chercher quand même cette racine, à l'aide de la méthode de la sécante, avec $x_0 = 1,4$ et $x_1 = 1,2$. Reprendre la même question avec la méthode de Newton et l'approximation initiale $x_0 = 1,2$. Comparer les vitesses de convergence des deux algorithmes.

Exercice 12

Construire la suite de Sturm correspondant au polynôme du texte, $x^3 - x^2 + 2x + 5$ et localiser les zéros réels de ce polynôme.

Exercice 13

Le volume molaire v du gaz carbonique à la température absolue T, sous la pression p, obéit à l'équation de van der Waals

$$(p + a/v^2)(v - b) = RT$$

avec R =0,082054 litre.atm.mol^{-1}K^{-1}, a = 3.592 atm.litre^2.mol^{-2} et b = 0,04267 litre.mol^{-1}.

a) Montrer qu'à pression et température données, v doit être solution d'une équation algébrique que l'on explicitera.

b) Calculer, à 350 K et 100 atm, le volume molaire de ce gaz en résolvant l'équation précédente par la méthode de Newton. On pourra utiliser la valeur fournie par la loi de Boyle-Mariotte ($pv = RT$) comme valeur initiale. On arrêtera le calcul lorsque deux valeurs successives du volume différeront de moins de 0,001.

Exercice 14

Pour gagner du temps, on utilise parfois une version simplifiée de la méthode de Newton qui consiste à remplacer (5.4) par

$$x_{k+1} = x_k - f(x_k)/f'(x_0) \equiv x_k - f(x_k)/g.$$

La pente f' est considérée comme constante et égale à sa valeur initiale, soit g.

a) En posant $\psi(x) = x - f(x)/g$ et en vous inspirant de l'étude de la méthode de Newton, examinez la convergence de cet algorithme et énoncez la condition que x_0 doit remplir pour que la suite des x_k converge vers la solution de $f = 0$.

b) Utiliser cette méthode pour trouver le volume molaire du gaz carbonique dans les conditions de l'exercice précédent.

Exercice 15

Dans l'exemple du polynôme perturbé de Wilkinson, combien vaut la variation relative du coefficient de x^{11} ?

Exercice 16

Résoudre le système

$$\begin{cases} x^2 - \dfrac{4}{3}xy + y^2 & = & 1, \\ x^2 - 2x + y^2 - 2y & = & -\dfrac{3}{2}. \end{cases}$$

This page intentionally left blank

Chapitre 6

Résolution de systèmes d'équations linéaires

La résolution de systèmes d'équations linéaires est peut-être le problème numérique que l'on rencontre le plus fréquemment. Des questions aussi diverses que l'étude des réseaux électriques en régime permanent, la résistance des matériaux, les modèles économétriques conduisent à des systèmes d'équations linéaires. Les problèmes aux limites, associés à des équations différentielles ou à des équations aux dérivées partielles, sont souvent discrétisés ce qui engendre encore des systèmes d'équations linéaires. C'est aussi le domaine le plus connu et le mieux étudié de l'analyse numérique : ces études s'appuient sur l'ensemble des connaissances en algèbre linéaire.

Le système linéaire général, qui comporte m équations et n inconnues, s'écrit

$$\left\{\begin{array}{lcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k + \cdots + a_{1n}x_n & = & b_1, \\ & \vdots & \\ a_{\ell 1}x_1 + a_{\ell 2}x_2 + \cdots + a_{\ell k}x_k + \cdots + a_{\ell n}x_n & = & b_\ell, \\ & \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mk}x_k + \cdots + a_{mn}x_n & = & b_m, \end{array}\right. \tag{6.1}$$

En introduisons la matrice $\boldsymbol{A}$ d'éléments $a_{\ell k}$ (le premier indice ici désigne la ligne $1 \leq \ell \leq m$, le deuxième la colonne $1 \leq k \leq n$)

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

et les vecteurs colonnes

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \qquad \boldsymbol{b} = [b_1, b_2, \ldots, b_m]^T$$

nous pouvons écrire le système linéaire sous la forme

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{6.2}$$

Remarque : La notation z^T désigne le transposé de l'objet z, vecteur ou matrice.

Dans l'équation (6.2), $\mathbf{b}$ est un vecteur connu de $\mathbb{R}^m$ (le vecteur des seconds membres) et $\mathbf{x}$ est un vecteur inconnu de $\mathbb{R}^n$ (le vecteur solution).

Dans la suite de ce chapitre (sauf pour le paragraphe concernant la méthode des moindres carrés), nous allons supposer que le nombre d'équations est égal au nombre d'inconnues du système, ou encore que le nombre de lignes de la matrice $\mathbf{A}$ est égal au nombre de colonnes, $m = n$. Cette matrice est alors carrée et on dit qu'elle est d'ordre n (ou m). (6.2) représente alors, sous forme condensée, une application de $\mathbb{R}^n$ dans lui-même (voir l'annexe de ce chapitre pour un bref rappel d'éléments d'algèbre linéaire). Si la matrice $\boldsymbol{A}$ est régulière, son « image » (l'ensemble des vecteurs $\boldsymbol{Ax}$) est $\mathbb{R}^n$ lui-même ; quel que soit le vecteur $\boldsymbol{b}$, on pourra toujours trouver un $\boldsymbol{x}$ satisfaisant (6.1). Si, au contraire, $\boldsymbol{A}$ est singulière, son image est un sous-ensemble de $\mathbb{R}^n$. Deux possibilités se présentent alors. Si $\boldsymbol{b}$ n'appartient pas à l'image de $\boldsymbol{A}$, le système est impossible, puisqu'on ne pourra jamais trouver un vecteur $\boldsymbol{x}$ convenable. Si $\boldsymbol{b}$ appartient à l'image de $\boldsymbol{A}$, il y aura une infinité de solutions, car on pourra toujours ajouter à une solution un vecteur $\boldsymbol{y}$ appartenant au « noyau » de $\boldsymbol{A}$ (l'ensemble des vecteurs $\boldsymbol{y}$ tels que $\boldsymbol{Ay} = \mathbf{0}$).

Nous pouvons retrouver la même classification par des considérations élémentaires de géométrie analytique. Soit le système linéaire :

$$\begin{aligned} ax + by &= m, \\ cx + dy &= n. \end{aligned}$$

Divisons la première équation par $\sqrt{a^2 + b^2}$ et la seconde par $\sqrt{c^2 + d^2}$, pour obtenir

$$\begin{aligned} n_{1x}x + n_{1y}y &= d_1, \\ n_{2x}x + n_{2y}y &= d_2. \end{aligned}$$

Les coefficients satisfaisant maintenant aux relations $n_{ix}^2 + n_{iy}^2 = 1$, $i = 1, 2$, les vecteurs $\boldsymbol{n}_i$ de coordonnées (n_{ix}, n_{iy}) sont unitaires.

Le système linéaire ainsi écrit admet l'interprétation géométrique suivante. Chaque équation est celle d'une droite D_i du plan (x, y). La droite D_i est perpendiculaire au vecteur $\boldsymbol{n}_i$; la distance de la droite à l'origine est d_i. La solution cherchée est représentée par le point d'intersection des deux droites. On peut alors distinguer trois cas : D_1 confondue avec D_2 (système indéterminé ou infinité de solutions), D_1 parallèle à D_2 (système impossible), et le cas normal où D_1 coupe D_2.

Numériquement, il faut encore distinguer ici deux possibilités. Si D_1 et D_2 font entre elles un angle notable, le point d'intersection est bien défini et la solution du système sera stable (dans un sens que nous allons préciser). Si D_1 et D_2 se coupent sous un angle très faible, le système est dit « mal conditionné ». Dans ce cas, la moindre variation de l'un des d_i (les seconds membres) ou de l'un des $n_{\alpha,i}$ (les coefficients, $\alpha = x, y$) entraînera un déplacement très important du point d'intersection, comme le montrent les figures 6.1 et 6.2.

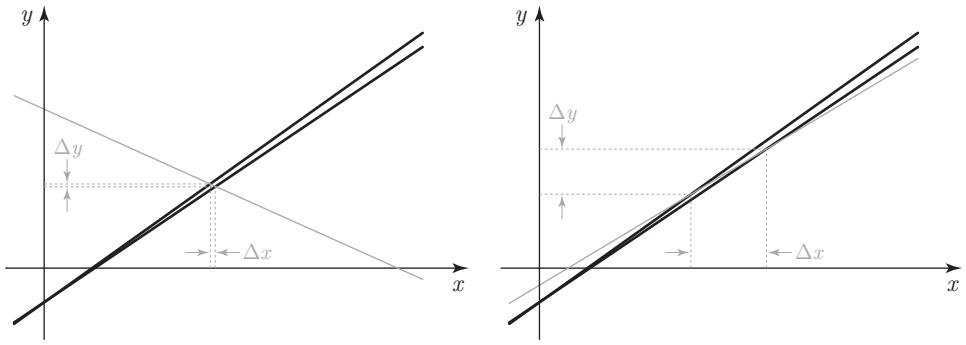


Figure 6.1 – Effet d'une perturbation d'un coefficient sur la solution d'un système linéaire ; système bien conditionné (à gauche) et système mal conditionné (à droite).

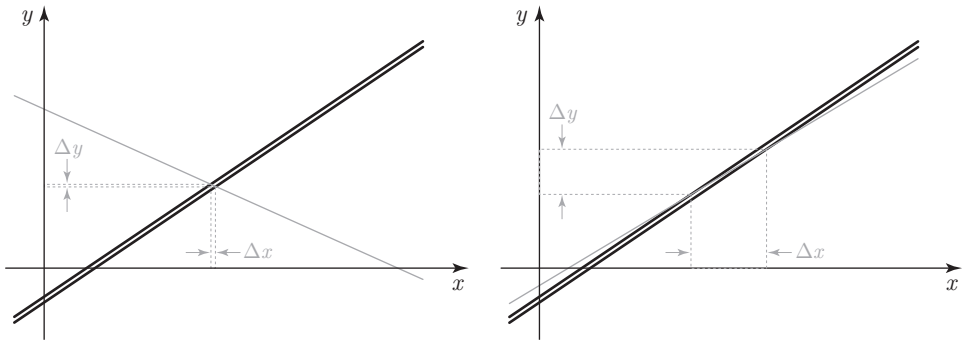


Figure 6.2 – Effet d'une perturbation d'un second membre sur la solution d'un système linéaire ; système bien conditionné (à gauche) et système mal conditionné (à droite).

Autrement dit, la solution est extrêmement sensible aux perturbations des coefficients ou des seconds membres. Une matrice dont le déterminant est nul est singulière. $\boldsymbol{A}$ est-elle « presque » singulière si $\det(\boldsymbol{A}) \simeq 0$? C'est le cas pour les systèmes mal conditionnés des figures 6.1 et 6.2 (droites presque parallèles). Malheureusement, il n'existe en général pas de corrélation entre $\det(\boldsymbol{A})$ et la condition du système $\boldsymbol{Ax} = \boldsymbol{b}$.

6.1. Le « conditionnement »

Pour l'analyse numérique, la difficulté de résolution d'un système linéaire tient aux effets conjugués de sa taille et de son « conditionnement ». Cet anglicisme désigne la sensibilité du système (ou plutôt de sa solution) aux perturbations des coefficients ou des seconds membres ; il s'agit en fait d'une généralisation des considérations géométriques du paragraphe précédent, sous un aspect plus quantitatif et qui peut, sans inconvénients, être sautée en première lecture. Admettons qu'il existe une norme vectorielle $\| \boldsymbol{x} \|$ et une norme matricielle $\| \boldsymbol{A} \|$, subordonnée à la précédente (c'est-à-dire que, pour tout $\boldsymbol{x}$, $\| \boldsymbol{A} \| \| \boldsymbol{x} \| \geq \| \boldsymbol{Ax} \|$).

En plus du système (6.2), nous considérons un système « perturbé » ; le second membre a subi une petite modification $\Delta \boldsymbol{b}$, ce qui a entraîné une petite variation $\Delta \boldsymbol{x}$ de la solution

$$\boldsymbol{A}(\boldsymbol{x} + \Delta \boldsymbol{x}) = \boldsymbol{b} + \Delta \boldsymbol{b}$$

d'où $\Delta \boldsymbol{x} = \boldsymbol{A}^{-1} \Delta \boldsymbol{b}$ et :

$$\| \Delta \boldsymbol{x} \| \leq \| \boldsymbol{A}^{-1} \| \| \Delta \boldsymbol{b} \| .$$

Nous en tirons une majoration de la variation relative :

$$\frac{\| \Delta \boldsymbol{x} \|}{\| \boldsymbol{x} \|} \leq \| \boldsymbol{A} \| \| \boldsymbol{A}^{-1} \| \frac{\| \Delta \boldsymbol{b} \|}{\| \boldsymbol{b} \|} = \mathrm{cond}(\boldsymbol{A}) \frac{\| \Delta \boldsymbol{b} \|}{\| \boldsymbol{b} \|}. \tag{6.3}$$

Le nombre $\mathrm{cond}(\boldsymbol{A}) \equiv \| \boldsymbol{A} \| \| \boldsymbol{A}^{-1} \|$ est appelé le conditionnement de la matrice $\boldsymbol{A}$. C'est une mesure de la sensibilité de l'erreur relative (sur la solution) aux variations du second membre.

Montrons maintenant que $\mathrm{cond}(\boldsymbol{A})$ représente aussi la sensibilité aux variations des coefficients des premiers membres. Nous considérons un nouveau système perturbé

$$(\boldsymbol{A} + \boldsymbol{\delta A})(\boldsymbol{x} + \boldsymbol{\delta x}) = \boldsymbol{b}.$$

En retranchant membre à membre (6.2), nous trouvons

$$\| \boldsymbol{\delta x} \| = - \| \boldsymbol{A}^{-1} \| \| \boldsymbol{\delta A} \| \| \boldsymbol{x} + \boldsymbol{\delta x} \|,$$

ce qui implique l'inégalité

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

soit encore

$$\frac{\| \boldsymbol{\delta x} \|}{\| \boldsymbol{x} + \boldsymbol{\delta x} \|} \leq \mathrm{cond}(\boldsymbol{A}) \frac{\| \boldsymbol{\delta A} \|}{\| \boldsymbol{A} \|}. \tag{6.4}$$

On peut montrer le résultat qualitatif suivant. Si $\mathrm{cond}(\boldsymbol{A}) \cong 10^m$ pour un entier $m \geq 0$, alors m chiffres significatifs seront perdus au cours du calcul de la solution de $\boldsymbol{Ax} = \boldsymbol{b}$, par rapport à la précision d'une opération arithmétique élémentaire.

Malheureusement, le calcul de $\boldsymbol{A}^{-1}$ est coûteux (nombre d'opérations de l'ordre de n^3, pour une matrice $n \times n$, voir §6.3.5) si bien que les considérations précédentes sont surtout d'intérêt théorique.

6.2. Orientation

Quelles sont les méthodes de résolution des systèmes linéaires que propose l'arsenal mathématique ? Tout d'abord une méthode que nous qualifierons de formelle : la solution du problème (où $\boldsymbol{A}$ est une matrice $n \times n$, $\boldsymbol{b}$ et $\boldsymbol{x}$ des vecteurs à n coordonnées) s'écrit $\boldsymbol{x} = \boldsymbol{A}^{-1} \boldsymbol{b}$. Puis la méthode de Cramer, dans laquelle chaque coordonnée de $\boldsymbol{x}$ est exprimée comme un rapport de déterminants. Elle se révèle inutilisable en pratique dès qu'il y a plus de trois ou quatre inconnues parce que le calcul d'un déterminant $n \times n$ nécessite $n!$ opérations, ce qui est prohibitif.

Nous rencontrons ensuite un groupe de méthodes qui procèdent par éliminations successives des inconnues : elles sont associées aux noms de Gauss et de Gauss–Jordan et sont d'application aisée. D'autres algorithmes décomposent la matrice des coefficients en un produit de facteurs ($\boldsymbol{A} \Longrightarrow \boldsymbol{LU}$) avant de pratiquer des éliminations successives qui sont alors très rapides à cause de la forme particulière des matrices $\boldsymbol{L}$ et $\boldsymbol{U}$: ce sont les méthodes dites de Crout, Doolittle ou Cholesky. Dans la suite, nous employons, de façon interchangeable, les termes de décomposition ou de factorisation (sauf au § 6). En dépit des apparences, les deux approches (élimination et factorisation) sont mathématiquement équivalentes ; elles sont aussi fréquemment employées en analyse numérique.

Dans un esprit bien différent, nous pourrons faire appel à des procédés itératifs de résolution, dont la convergence n'est pas toujours assurée : ce sont les méthodes associées aux noms de Jacobi et Gauss–Seidel.

Comme vous le verrez (§ 6.3.7), le calcul de l'inverse $\boldsymbol{A}^{-1}$ d'une matrice $\boldsymbol{A}$ est plus long que le calcul direct de la solution du système linéaire (6.1). L'analogie avec l'algèbre peut faire comprendre cette remarque. Pour résoudre $10x = 3$, nous pouvons faire $10^{-1} = 0,1 ; 0,1 \times 3 = 0,3$ ou directement $x = 3/10 = 0,3$. On évite donc de calculer l'inverse de la matrice des coefficients, à moins que le problème posé ne le demande explicitement. Si tel est le cas, nous obtiendrons $\boldsymbol{A}^{-1}$ en résolvant n systèmes linéaires particuliers, dont les seconds membres sont respectivement égaux à chacun des vecteurs de la base canonique.

Dans la mise en oeuvre des méthodes d'élimination ou de factorisation, l'ordinateur ne commet que des erreurs d'arrondi. Dans le cas d'une méthode itérative, nous devrons tenir compte aussi d'une erreur de troncation, puisque l'algorithme s'arrête au bout d'un nombre fini d'opérations.

6.3. Méthode de Gauss

6.3.1. Algorithme

L'algorithme de Gauss, enseigné dans les cours élémentaires d'algèbre, consiste à éliminer successivement les inconnues pour parvenir à un système d'équations simple à résoudre. Nous utilisons, pour le système initial, la notation plus détaillée

$$\boldsymbol{A}^{(1)}\boldsymbol{x} = \boldsymbol{b}^{(1)},$$

les éléments de $\boldsymbol{A}^{(1)}$ étant notés $a_{ij}^{(1)}$ et ceux de $\boldsymbol{b}^{(1)}$, $b_i^{(1)}$. Nous allons transformer le système de départ en un système équivalent $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{c}$ où la matrice $\boldsymbol{U}$ est triangulaire supérieure (U pour « upper »). Nous savons que la solution du système reste inchangée si :

- on multiplie une équation (une ligne de $\boldsymbol{A}$ et la coordonnée correspondante de $\boldsymbol{b}$) par une constante ;
- on ajoute une équation à une autre (une ligne de $\boldsymbol{A}$ à une autre et la coordonnée correspondante de $\boldsymbol{b}$ à l'autre).

À la première étape, nous éliminons x_1 de toutes les lignes sauf la première. Pour cela, nous supposons que le « pivot » $a_{11}^{(1)} \neq 0$ et nous définissons les « multiplicateurs »

$$m_{i1} = -\frac{a_{i1}^{(1)}}{a_{11}^{(1)}}.$$

Nous formons une nouvelle ligne de rang i en **ajoutant** m_{i1} fois la première ligne à la i-ème :

$$\begin{aligned} a_{ij}^{(2)} &= a_{ij}^{(1)} + m_{i1} a_{1j}^{(1)} \qquad i,j = 2,\ldots,n, \\ b_i^{(2)} &= b_i^{(1)} + m_{i1} b_1^{(1)} \qquad i = 2,\ldots,n. \end{aligned}$$

Remarque : De nombreux auteurs choisissent la convention opposée :

$$m_{i1} = +a_{i1}^{(1)}/a_{11}^{(1)}$$

et **retranchent** m_{i1} fois la première ligne à la i-ème.

La première ligne de $\boldsymbol{A}$ et la première composante de $\boldsymbol{b}$ sont inchangées. Le système d'équations a pris la forme

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(2)} \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{bmatrix}.$$

Vous constatez que l'inconnue x_1 a disparu des lignes $2,3,\ldots,n$. Le cas général est très semblable. Nous supposons avoir éliminé $x_1, x_2, \ldots, x_{k-1}$, si bien que le tableau des coefficients a pris l'aspect suivant

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & & \cdots & & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & & \cdots & & a_{2n}^{(2)} \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{bmatrix}$$

Définissons un nouveau pivot $a_{kk}^{(k)}$, que nous supposons encore différent de zéro et de nouveaux multiplicateurs

$$m_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1,\ldots,n. \tag{6.5}$$

À l'aide de ces éléments, nous éliminons l'inconnue x_k des équations de rang $k+1, k+2, \ldots, n$, ce qui se traduit par les équations

$$\begin{aligned} a_{ij}^{(k+1)} &= a_{ij}^{(k)} + m_{ik} a_{kj}^{(k)} \qquad i,j = k+1,\ldots,n, \\ b_i^{(k+1)} &= b_i^{(k)} + m_{ik} b_k^{(k)} \qquad i = k+1,\ldots,n. \end{aligned} \tag{6.6}$$

Nous ne touchons pas aux lignes de rang 1 à k et nous remplaçons par des zéros les éléments de la colonne k situés sous le pivot. Au bout de $n-1$ étapes analogues à celle qui vient d'être décrite, l'élimination est terminée et le système a pris la forme $\boldsymbol{A}^{(n)}\boldsymbol{x} = \boldsymbol{b}^{(n)}$, ou encore

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & a_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(n)} \\ b_2^{(n)} \\ \vdots \\ b_n^{(n)} \end{bmatrix}.$$

Posons, pour simplifier la notation, $\boldsymbol{U} = \boldsymbol{A}^{(n)}$ et $\boldsymbol{c} = \boldsymbol{b}^{(n)}$. Le système d'équations $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{c}$ est triangulaire supérieur et se résout facilement, à condition de commencer par la dernière ligne et de remonter. Nous trouvons

$$x_n = \frac{c_n}{u_{nn}}$$

puis

$$x_k = \frac{1}{u_{kk}} \left[c_k - \sum_{j=k+1}^{n} u_{kj} x_j \right], \quad k = n-1, n-2, \ldots, 1. \tag{6.7}$$

Exemple
Définition de $\boldsymbol{A}$ et $\boldsymbol{b}$ et construction de la matrice « augmentée » (qui inclut $\boldsymbol{b}$ comme dernière colonne).

```
-->A = [1,2,1;2,3,3;-1,-3,1]
 A  =   1.    2.    1.
        2.    3.    3.
      - 1.  - 3.    1.
-->Aa = [A,b]
 Aa  =   1.    2.    1.    1.
         2.    3.    3.    3.
       - 1.  - 3.    1.    2.
```

```
-->b = [1,3,2]'
     b  =  1.
           3.
           2.
```

Élimination de x_1, pivots -2 et $+1$.

```
-->Aa(2,:) = Aa(2,:)-2*Aa(1,:)
 Aa =    1.    2.    1.    1.
         0.  - 1.    1.    1.
       - 1.  - 3.    1.    2.
```

```
-->Aa(3,:) = Aa(3,:)+Aa(1,:)
 Aa =    1.    2.    1.    1.
         0.  - 1.    1.    1.
         0.  - 1.    2.    3.
```

Élimination de x_2, pivot -1.

```
-->Aa(3,:)=Aa(3,:)-Aa(2,:)
Aa =    1.    2.    1.    1.
        0.  - 1.    1.    1.
        0.    0.    1.    2.
```

Extraction de $\boldsymbol{U}$ et de $\boldsymbol{c}$

```
-->U = Aa(1:3,1:3)
 U  = 1.    2.    1.
      0.  - 1.    1.
      0.    0.    1.
```

```
-->c = Aa(:,$)
    c  =  1.
          1.
          2.
```

Calcul de $\boldsymbol{x}$.

```
-->x(3) = 2;
-->x(2) = -(1-x(3));
-->x(1) = 1-x(3)-2*x(2)
 x  =  - 3.
         1.
         2.
```

Remarque : Vous pouvez facilement conserver une copie d'un dialogue avec Scilab. Votre première instruction sera analogue à celle-ci : `diary(\"D:an\exemple\")`. Le dialogue sera enregistré en continu dans le fichier désigné. Vous arrêterez l'enregistrement par `diary(0)`.

La programmation sous Scilab de l'algorithme de Gauss peut se faire à différents niveaux de détail, depuis la « boîte noire » : `x =A\b` jusqu'à l'écriture de chaque opération arithmétique dans chaque boucle. Pour le programme ci-dessous, nous avons choisi une voie moyenne : les boucles considérées comme « banales » sont écrites en utilisant un vecteur d'indices ou l'opérateur « : ».

Listing 6.1 – Élimination de Gauss sans permutation de lignes

```
clear                                                                    1
N = 4;                                                                   2
A = rand(N,N)+eye(N,N);                                                  3
for k = 1:N-1                                                            4
  lignes = k+1:N;                                                        5
  A(lignes,k) = - A(lignes,k)/A(k,k);                                    6
  A(lignes,lignes) = A(lignes,lignes)+A(lignes,k)*A(k,lignes);           7
end                                                                      8
U = zeros(A); L = eye(A);                                                9
for k = 1:N                                                              10
  cols = k:N;                                                            11
  U(k,cols) = A(k,cols);                                                 12
  cols = 1:k-1;                                                          13
  L(k,cols) = - A(k,cols);                                               14
end                                                                      15
```

Les lignes 3 à 4 créent une matrice $n \times n$. L'algorithme de Gauss est mis en oeuvre par une boucle sur `k`. L'instruction 6 calcule tous les multiplicateurs et les range dans la colonne k, sous le pivot $a_{kk}^{(k)}$ (là où doivent apparaître les zéros). L'instruction 7 effectue l'élimination. L'expression `A(lignes,k)*A(k,lignes)` représente le produit

d'un vecteur colonne par un vecteur ligne ; le résultat est une matrice d'élément général $m_{ik}a_{ki}^{(k)}$, comme prévu par l'équation (6.6). La boucle suivante extrait la matrice $\boldsymbol{U}$ de $\boldsymbol{A}^{(n)}$ et construit une matrice $\boldsymbol{L}$ que nous définissons ci-dessous. Ce programme est incomplet, puisqu'il ne traite pas les seconds membres.

6.3.2. Méthode de Gauss–Jordan

Pour les personnes qui auraient pris goût à l'élimination, nous proposons ici une variante. Plutôt que de substituer dans le système triangulaire, nous poursuivons l'élimination au-dessus de la diagonale principale : c'est la méthode de Gauss–Jordan. On choisit un pivot et on définit un multiplicateur selon l'équation (6.5) puis on utilise les équations (6.6) avec les conditions $j = k, \ldots, n$ et $i = 1, \ldots, n$, $i \neq k$. La matrice des coefficients prend, à la fin, la forme diagonale et le système se résout à vue.

Exemple
Matrice augmentée après élimination de x_1 :

```
 Aa  =   1.    2.    1.    1.
         0.  - 1.    1.    1.
         0.  - 1.    2.    3.
```

Élimination de x_2

```
-->Aa(1,:)=Aa(1,:)+2*Aa(2,:);
-->Aa(3,:)=Aa(3,:)-Aa(2,:)
 Aa  =    1.    0.    3.    3.
          0.  - 1.    1.    1.
          0.    0.    1.    2.
```

Élimination de x_3

```
-->Aa(1,:)=Aa(1,:)-3*Aa(3,:);
-->Aa(2,:)=Aa(2,:)-Aa(3,:)
 Aa  =    1.    0.    0.  - 3.
          0.  - 1.    0.  - 1.
          0.    0.    1.    2.
```

Finalement, $x_1 = -3$; $x_2 = 1$; $x_3 = 2$.

6.3.3. Décomposition *LU*

Les multiplicateurs de la méthode de Gauss vont maintenant nous servir pour construire un algorithme d'apparence différente mais en fait complètement équivalent.

Définissons la matrice triangulaire inférieure (L pour « lower »)

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & & & \vdots \\ -m_{n1} & -m_{n2} & -m_{n3} & \cdots & 1 \end{bmatrix} \tag{6.8}$$

Nous avons alors le résultat remarquable que, si $\boldsymbol{L}$ et $\boldsymbol{U}$ résultent de l'élimination de Gauss portant sur la matrice $\boldsymbol{A}$,

$$\boldsymbol{A} = \boldsymbol{LU}. \tag{6.9}$$

Vérifions-le en calculant l'élément (i, j) du produit. C'est le produit scalaire du vecteur ligne $[-m_{i1}, -m_{i2}, \ldots, -m_{i,i-1}, 1, 0, \ldots, 0]$ de $\boldsymbol{L}$ par le vecteur colonne $[u_{1j}, u_{2j}, \ldots, u_{j,j}, 0, \ldots, 0]^T$ de $\boldsymbol{U}$. Nous distinguons deux cas : $i \leq j$ (au dessus de, ou sur, la diagonale principale) et $i > j$ et nous utilisons les définitions des éléments de $\boldsymbol{L}$ et $\boldsymbol{U}$, en particulier la première des équations (6.6).

Si $i \leq j$:

$$\begin{aligned} (LU)_{ij} &= -\sum_{k=1}^{i-1} m_{ik}u_{kj} + u_{ij} = -\sum_{k=1}^{i-1} m_{ik}a_{kj}^{(k)} + a_{ij}^{(i)} \\ &= -\sum_{k=1}^{i-1} \left[a_{ij}^{(k+1)} - a_{ij}^{(k)}\right] + a_{ij}^{(i)} = a_{ij}^{(1)} = a_{ij}. \end{aligned}$$

Si $i > j$:

$$\begin{aligned} (LU)_{ij} &= -\sum_{k=1}^{j} m_{ik}u_{kj} = -\sum_{k=1}^{j-1} m_{ik}a_{kj}^{(k)} + a_{ij}^{(j)} \\ &= -\sum_{k=1}^{j-1} \left[a_{ij}^{(k+1)} - a_{ij}^{(k)}\right] + a_{ij}^{(j)} = a_{ij}^{(1)} = a_{ij}. \end{aligned}$$

Vous savez que le déterminant d'un produit de matrices est égal au produit des déterminants de chaque facteur ; de plus, le déterminant d'une matrice triangulaire se calcule comme le produit des éléments diagonaux. La factorisation LU nous offre ainsi en prime la valeur du déterminant de $\boldsymbol{A}$:

$$\det(\boldsymbol{A}) = \det(\boldsymbol{L}) \det(\boldsymbol{U}) = u_{11}u_{22}\cdots u_{nn}. \tag{6.10}$$

Pour trouver la solution de (6.1), maintenant que nous connaissons les matrices $\boldsymbol{L}$ et $\boldsymbol{U}$, il nous suffit de résoudre successivement deux systèmes triangulaires. D'abord

$$\boldsymbol{Ly} = \boldsymbol{b}, \tag{6.11}$$

dont la solution est le vecteur $\boldsymbol{y}$, puis

$$\boldsymbol{Ux} = \boldsymbol{y} \tag{6.12}$$

dont la solution est le vecteur $\boldsymbol{x}$ cherché. En effet, en substituant $\boldsymbol{y}$ tiré de (6.12) dans (6.11), nous trouvons $\boldsymbol{LUx} = \boldsymbol{Ax} = \boldsymbol{b}$. Cette démarche est profitable si l'on doit résoudre plusieurs systèmes d'équations linéaires qui ne diffèrent que par leurs seconds membres : la factorisation est faite une fois pour toutes.

Exemple – La matrice de l'exemple de la section précédente nous a fourni les facteurs

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

et vous vérifiez facilement que $\boldsymbol{LU} = \boldsymbol{A}$ et que $\det(\boldsymbol{A}) = -1$. La solution du système linéaire s'obtient, comme nous l'avons dit, en deux étapes. La première consiste à résoudre $\boldsymbol{Ly} = \boldsymbol{b}$; nous commençons par la **première** ligne pour trouver successivement $y_1 = 1$, $y_2 = 3 - 2 = 1$ et $y_3 = 2 - 1 + 1 = 2$. À la deuxième étape, nous résolvons $\boldsymbol{Ux} = \boldsymbol{y}$, en commençant par la **dernière** ligne ce qui nous donne $x_3 = 2$, $x_2 = -(1 - 2) = 1$ et $x_1 = 1 - 2 - 2 = -3$.

Scilab fournit une fonction qui effectue la factorisation en une instruction :

`[L,U,P] = lu(A)`.

$\boldsymbol{P}$ est une matrice de permutation (voir plus loin). La fonction équivalente sous Maple s'appelle `LUdecomp`.

6.3.4. Représentation matricielle de l'élimination

Vous pouvez avoir le sentiment que la décomposition LU d'une matrice régulière quelconque est le fruit du hasard ; il n'en est rien ; nous allons en donner, sur un exemple, une autre présentation qui sera peut-être plus convaincante.

Multiplier la coordonnée 1 d'un vecteur par le coefficient m, puis ajouter le résultat à la coordonnée 2 revient à multiplier ce même vecteur par une matrice particulière :

$$\begin{bmatrix} 1 & 0 & 0 \\ m & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ mb_1 + b_2 \\ b_3 \end{bmatrix}$$

Plus généralement, pour ajouter m fois la ligne i à la ligne j, il suffit de disposer le nombre m à la place de l'élément j, i de la matrice unité ($i < j$). Ce qui est vrai pour un vecteur l'est également pour une matrice. La prémultiplication de $\boldsymbol{A}$ par la matrice précédente revient à multiplier successivement chaque colonne de $\boldsymbol{A}$, donc à ajouter la première ligne de $\boldsymbol{A}$ à la deuxième. Encore un exemple ; nous voulons ajouter -3 fois la 2-ième ligne à la 3-ième de la matrice générale : -3 est en ligne 3, colonne 2 :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ -3a_{21} + a_{31} & -3a_{22} + a_{32} & -3a_{23} + a_{33} \end{bmatrix}$$

Les matrices qui viennent d'être définies (dites matrices de Frobenius) jouissent de propriétés intéressantes. Si $\boldsymbol{f}(i, j, m), i \geq j$ désigne la matrice de Frobenius contenant

m en ligne i et colonne j, alors $\boldsymbol{f}^{-1} = \boldsymbol{f}(i,j,-m)$: soustraire m fois la ligne i de la ligne j est bien l'opération inverse d'ajouter m fois (i) à (j). $\boldsymbol{f}(i,j,m)\boldsymbol{f}(i',j,m') = \boldsymbol{f}(i',j,m')\boldsymbol{f}(i,j,m)$: la matrice produit contient m et m' en colonne j, aux lignes i et i', ce qui représente bien le fait qu'ajouter un multiple de la ligne j aux lignes i et i' sont des actions indépendantes (et que l'on peut réaliser dans un ordre quelconque).

Exemple – Reprenons le système du paragraphe précédent, représenté par la matrice de ses coefficients :

$$\boldsymbol{A}^{(1)} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 3 \\ -1 & -3 & 1 \end{bmatrix}$$

L'élimination de x_1 a fait intervenir les pivots -2 et 1, ce qui peut s'écrire comme

$$\boldsymbol{A}^{(2)} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 1 \\ 0 & -1 & 2 \end{bmatrix} = \boldsymbol{f}(2,1,-2)\boldsymbol{f}(3,1,1)\boldsymbol{A}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \boldsymbol{A}^{(1)} \equiv \boldsymbol{M}_1 A^{(1)}.$$

Nous désignons par $\boldsymbol{M}_1$ le produit des deux matrices de Frobenius ; vous voyez qu'il est facile à former, il suffit de recopier les multiplicateurs à leur place dans la matrice unité. L'élimination de x_2, dans la dernière ligne, s'est faite grâce au pivot -1, soit

$$\boldsymbol{A}^{(3)} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \boldsymbol{f}(3,2,-1)\boldsymbol{A}^{(2)} \equiv \boldsymbol{M}_2 A^{(2)} \equiv \boldsymbol{U}.$$

La matrice $\boldsymbol{M}_2$ est identique à la matrice de Frobenius et le résultat a été noté $\boldsymbol{U}$, comme dans le cas général. Nous avons les relations

$$\boldsymbol{U} = \boldsymbol{A}^{(3)} = \boldsymbol{M}_2\boldsymbol{A}^{(2)} = \boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{A}^{(1)},$$

ou encore (en multipliant par les inverses des $\boldsymbol{M}_i$)

$$\boldsymbol{A}^{(1)} = \boldsymbol{M}_1^{-1}\boldsymbol{M}_2^{-1}\boldsymbol{U}.$$

Les inverses de matrices $\boldsymbol{M}$ se calculent très facilement (comme pour les $\boldsymbol{f}$) en changeant les signes des éléments extra-diagonaux ; ainsi, par exemple

$$\boldsymbol{M}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \equiv \boldsymbol{L}_1.$$

Avec ces notations, il vient

$$\boldsymbol{A} = \boldsymbol{L}_1\boldsymbol{L}_2\boldsymbol{U}.$$

Vous pouvez vérifier que le produit de deux matrices $\boldsymbol{L}_i$ se forme comme pour les matrices $\boldsymbol{M}$ en recopiant les éléments hors diagonale à leur place. On retrouve ainsi la factorisation $\boldsymbol{LU}$ obtenue au paragraphe précédent.

6.3.5. Permutation de lignes

Tous les raisonnements des sections précédentes s'écroulent si nous rencontrons un pivot nul. Heureusement, nous savons que nous pouvons permuter deux équations du système sans changer la solution ; il suffit donc, pour éliminer x_k lorsque $a_{kk}^{(k)} = 0$, de choisir une ligne, disons la ligne k' telle que $a_{k'k}^{(k)} \neq 0$, puis de permuter les lignes k et k' (dans $\boldsymbol{A}^{(k)}$ **et** $\boldsymbol{b}^{(k)}$) puis de reprendre l'algorithme de Gauss. Si nous ne trouvons aucun pivot non nul, c'est que la matrice est singulière, contrairement à l'hypothèse faite au début.

En fait, il faut même éviter les « petits » pivots ; en effet ces nombres auraient peut-être été nuls lors d'un calcul exact et peuvent ne devoir leur existence qu'à des erreurs d'arrondi commises au cours des étapes précédentes. De plus, diviser par un nombre petit (ou multiplier par un grand facteur) va exalter toutes les erreurs d'arrondi et rendre le calcul instable. Il y a même intérêt, pour éviter d'amplifier les erreurs d'arrondi, à ce qu'aucun multiplicateur ne soit plus grand que un. Il faut donc, à chaque étape de l'algorithme de Gauss, rechercher non pas un pivot non nul mais le « pivot maximal ».

Où chercher ce pivot sympathique ? Il y a deux possibilités : la recherche du plus grand élément (en valeur absolue) dans la colonne k (recherche « partielle »), suivie d'une permutation de lignes ou la recherche de l'élément maximal dans la sous matrice $a_{i,j}^{(k)}, \quad i,j \geq k$ (recherche « complète »), suivie d'une permutation de lignes **et** de colonnes. La recherche complète est plus compliquée à programmer car permuter des colonnes revient à permuter des inconnues. Il faut absolument restaurer les identités des inconnues en fin de calcul. Quelle méthode choisir ? La théorie reste assez vague et on constate empiriquement que la recherche partielle du pivot maximal est aussi performante que la recherche complète ; comme elle est plus simple, c'est elle que nous recommandons, en accord avec la plupart des auteurs.

La décomposition $\boldsymbol{LU}$ est modifiée si l'on doit permuter des lignes, de même que sa représentation matricielle. Avant de citer le résultat, introduisons les matrices de permutation : ce sont des matrices qui se déduisent de la matrice unité par une permutation de lignes, comme par exemple

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Vous pouvez vérifier que $P[x_1, x_2, x_3]^T = [x_3, x_2, x_1]^T$: la prémultiplication d'un vecteur (ou d'une matrice) par $\boldsymbol{P}$ fait subir aux lignes du vecteur (ou de la matrice) la même permutation que celle qui fait passer de la matrice unité à $\boldsymbol{P}$. Les matrices de permutation sont des matrices orthogonales, $\boldsymbol{P}^{-1} = \boldsymbol{P}^T$.

Nous pouvons maintenant énoncer le résultat

$$\boldsymbol{LU} = \boldsymbol{PA}. \tag{6.13}$$

En d'autres termes, la factorisation s'applique à une matrice dont les lignes ont été convenablement permutées. Comme la démonstration est laborieuse, contentons-nous d'un exemple.

Exemple

Remarque : Nous effectuons ici des calculs exacts sur une petite matrice contenant de petits entiers ; dans ce cas, il n'est pas obligatoire de choisir comme pivot le plus grand élément de la colonne, on se contente d'éviter les pivots nuls.

Définition de la matrice des coefficients :

```
 A1  =    1.    2.    1.
          2.    4.    3.
        - 1.  - 3.    1.
```

Élimination de x_1 :

```
 M1  =    1.    0.    0.
        - 2.    1.    0.
          1.    0.    1.
-->A2 = M1*A1 =    1.    2.    1.
                   0.    0.    1.
                   0.  - 1.    2.
```

Permutation des lignes 2 et 3

```
-->P  =    1.    0.    0.
           0.    0.    1.
           0.    1.    0.
-->A21 = P*A2 =     1.    2.    1.
                    0.  - 1.    2.
                    0.    0.    1.
```

La triangularisation est terminée, `U = A21;`

```
-->L1  =    1.    0.    0.
            2.    1.    0.
          - 1.    0.    1.

-->L1*U =    1.    2.    1.
             2.    3.    4.
           - 1.  - 2.    0.
```

ce qui ne ressemble à rien. Procédons plus soigneusement ; nous avons :

`A1 = L1*A2, A2 = P*A21 = P*U`, d'où
`A1 = L1*P*U` et `P*A1 = P*L1*P*U`.

Vérifions :

```
-->P*L1*P*U  =    1.    2.    1.
                - 1.  - 3.    1.
                  2.    4.    3.
```

et

```
-->P*L1*P  =    1.    0.    0.
              - 1.    1.    0.
                2.    0.    1.
```

Les multiplicateurs doivent suivrent leurs équations dans l'échange de lignes.

Le programme suivant effectue ou une élimination de Gauss, avec recherche partielle du pivot maximal, ou une décomposition LU, au prix d'une petite modification.

Listing 6.2 – Élimination de Gauss avec permutation de lignes

```
// ... lecture de N et A ...                                              1
for k = 1:N-1                                                             2
  [m,km] = max(abs(A(k:N,k)));                                            3
  km = km + k -1;                                                         4
  temp = A(k,k:N); A(k,k:N) = A(km,k:N); A(km,k:N) = temp;                5
//   temp = A(k,1:N); A(k,1:N) = A(km,1:N); A(km,1:N) = temp;             6
  if A(k,k) <> 0                                                          7
    lignes = k+1:N;                                                       8
    A(lignes,k) = - A(lignes,k)/A(k,k);                                   9
    A(lignes,lignes)=A(lignes,lignes)+A(lignes,k)*A(k,lignes);            10
  end                                                                     11
end                                                                       12
A                                                                         13
//U = zeros(A);                                                           14
//L = eye(A);                                                             15
//for k = 1:N                                                             16
//   cols = k:N;                                                          17
//   U(k,cols) = A(k,cols);                                               18
//   cols = 1:k-1;                                                        19
//   L(k,cols) = - A(k,cols);                                             20
//end                                                                     21
//L,U,L*U                                                                 22
```

La ligne 3 recherche le plus grand élément en module de la colonne `k` (son rang, au sein du petit vecteur `A(k:N,k)`, est `km`, sa valeur est `m`) et la ligne 4 assure la numérotation correcte de cet élément. C'est l'instruction 5 qui est active pour une élimination de Gauss. C'est, au contraire, la ligne 6 qui doit l'être pour une factorisation LU. Les instructions 7 à 11 réalisent l'élimination proprement dite et les lignes 14 à 22 construisent, si nécessaire, les matrices $\boldsymbol{L}$ et $\boldsymbol{U}$, en extrayant les éléments convenables de $\boldsymbol{A}$.

Un algorithme tout à fait semblable peut être associé à la méthode de Gauss–Jordan.

Les erreurs d'arrondi sont plus faibles si les éléments des matrices successives restent à peu près du même ordre de grandeur. Pour cela, on peut, avant de résoudre le système, multiplier certaines lignes par des facteurs tels que le plus grand coefficient de chaque ligne soit compris entre 0,5 et 1 (équilibrage).

6.3.6. Nombre d'opérations

Pour estimer le nombre d'opérations arithmétiques nécessaires à la résolution d'un système linéaire, nous supposerons qu'aucun pivot n'est nul et donc qu'aucune permutation de lignes n'est nécessaire.

Commençons par la triangularisation. À la première étape de l'élimination, il nous faut $n-1$ divisions pour calculer les multiplicateurs m_{1i}, $2 \leq i \leq n$. Le calcul des éléments de la nouvelle matrice, $a_{ij}^{(2)}$, nous coûte ensuite $(n-1)^2$ multiplications et autant d'additions. L'étape suivante n'implique plus que $n-2$ divisions, $(n-2)^2$ multiplications et $(n-2)^2$ additions. Finalement, nous éliminons de x_{n-1} avec une division, une multiplication et une addition. Le nombre d'opérations se calcule à l'aide des identités

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

Si nous estimons que, sur une machine récente, toutes les opérations arithmétiques prennent pratiquement le même temps, nous pouvons ne pas distinguer additions, multiplications et divisions. On trouve alors un nombre total d'opérations égal à

$$NOP_t = \frac{1}{3}n(n+1)(2n+\frac{1}{2}) \sim \frac{2}{3}n^3. \tag{6.14}$$

La transformation de $\boldsymbol{b}^{(1)}$ en $\boldsymbol{b}^{(2)}$ nous coûte $n-1$ multiplications et $n-1$ additions, celle de $\boldsymbol{b}^{(2)}$ en $\boldsymbol{b}^{(3)}$ $n-2$ multiplications et $n-2$ additions et ainsi de suite. Au total, nous obtenons $\boldsymbol{b}^{(n)}$ en

$$NOP_b = n(n-1) \sim n^2 \tag{6.15}$$

opérations arithmétiques.

Il faut enfin résoudre le système triangulaire $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{c}$, ce qui coûte

$$NOP_r = n(n+1) \sim n^2 \tag{6.16}$$

opérations. Nous retiendrons le résultat asymptotique, valable quand n est « grand » :

$$NOP \sim \frac{2}{3}n^3, \tag{6.17}$$

avec une contribution à peu près équivalente d'additions et de multiplications/divisions. Ce nombre d'opérations paraît faible lorsqu'on le compare au cas apparemment simple du produit de deux matrices $n \times n$ qui coûte environ $2n^3$ opérations.

Nous ne considérerons pas ici les comparaisons, qui peuvent être importantes si on applique la recherche complète du pivot maximal, ni les permutations d'éléments, dont le nombre est difficile à estimer à l'avance.

L'algorithme de Gauss est assez économe en mémoire, si l'on accepte de détruire $\boldsymbol{A}$. En effet, à chaque étape, nous pouvons ranger les éléments de $\boldsymbol{L}$ à la place que devraient occuper les zéros de $\boldsymbol{A}$ après élimination (dans le triangle inférieur). Les 1 de la diagonale de $\boldsymbol{L}$ sont sous-entendus. Le triangle supérieur de $\boldsymbol{A}$ est progressivement occupé par les éléments de $\boldsymbol{U}$.

6.3.7. Calcul de l'inverse de A

Pour quelques rares applications, il faut calculer $\boldsymbol{A}^{-1}$ explicitement, soit encore trouver $\boldsymbol{X}$ telle que $\boldsymbol{AX} = \boldsymbol{I}$. La colonne j de $\boldsymbol{X}$ est solution du système linéaire :

$$\boldsymbol{A}\boldsymbol{x}_j = \boldsymbol{e}_j,$$

où $\boldsymbol{e}_j$ est un vecteur de base. Nous allons avoir à résoudre n systèmes linéaires dont les premiers membres sont identiques, mais les seconds membres différents. Nous savons maintenant qu'il est commode d'utiliser la décomposition $\boldsymbol{PA} = \boldsymbol{LU}$ effectuée une seule fois, puis de résoudre les systèmes triangulaires $\boldsymbol{L}\boldsymbol{y}_j = \boldsymbol{P}\boldsymbol{e}_j$ puis $\boldsymbol{U}\boldsymbol{x}_j = \boldsymbol{y}_j$. Le nombre d'opérations nécessaires est $\cong \frac{8}{3}n^3$, quatre fois plus que pour résoudre un système linéaire de même n. Si l'on tient compte de ce que les coordonnées de $\boldsymbol{y}_j$ sont nulles jusqu'à un certain rang, on peut n'utiliser que $2n^3$ opérations.

On peut également trouver l'inverse d'une matrice à l'aide de l'algorithme de Gauss–Jordan légèrement modifié. À l'étape k de l'élimination, après avoir choisi le pivot, on pose

$$a_{kj}^{(k+1)} = \frac{a_{kj}^{(k)}}{a_{kk}^{(k)}} \quad k \leq j \leq n; \quad b_k^{(k+1)} = \frac{b_k^{(k)}}{a_{kk}^{(k)}}.$$

puis on élimine x_j au-dessus et au-dessous de la ligne k selon

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{ik}^{(k)} a_{kj}^{(k+1)}; \quad b_i^{(k+1)} = b_k^{(k)} - a_{ik}^{(k)} b_k^{(k+1)}$$

pour $k \leq j \leq n$ et $1 \leq i \leq n, i \neq k$. Appliquée à la matrice augmentée $[\boldsymbol{A}|\boldsymbol{b}]$, cette méthode produit la matrice $[\boldsymbol{I}|\boldsymbol{c}]$; lorsque $\boldsymbol{b} = \boldsymbol{e}_j$, $\boldsymbol{c}$ est la colonne j de l'inverse. Le coût en opérations est le double de celui de la factorisation, mais l'encombrement de la mémoire est plus faible.

6.4. Factorisation directe

Nous savons que la matrice régulière $\boldsymbol{A}$ admet une représentation sous forme d'un produit de matrices triangulaires $\boldsymbol{LU}$. Nous pouvons alors nous poser la question suivante : est-il possible de trouver les éléments de $\boldsymbol{L}$ et $\boldsymbol{U}$ sans utiliser l'algorithme de Gauss ? La réponse est affirmative : il suffit d'identifier chaque élément dans l'équation $\boldsymbol{A} = \boldsymbol{LU}$. Comme la décomposition $\boldsymbol{LU}$ est unique, ce nouvel algorithme est équivalent à celui de Gauss.

Soient ℓ_{ij} les éléments de $\boldsymbol{L}$, avec $\ell_{ij} = 0$ si $i < j$; de même, les éléments de $\boldsymbol{U}$ sont tels que $u_{ij} = 0$ si $i > j$; enfin $\ell_{ii} = 1$. Les éléments de la première ligne de $\boldsymbol{A}$ vérifient

$$\begin{aligned}
a_{11} &= \sum \ell_{1k} u_{k1} = \ell_{11} u_{11} = u_{11}, \\
a_{12} &= \sum \ell_{1k} u_{k2} = u_{12}, \\
&\dots \\
a_{1n} &= \sum \ell_{1k} u_{kn} = u_{1n}.
\end{aligned}$$

En utilisant la première ligne de $\boldsymbol{A}$, nous avons trouvé la première ligne de $\boldsymbol{U}$. L'astuce pour continuer (due à Crout, d'où le nom de l'algorithme), consiste à identifier maintenant la première colonne de $\boldsymbol{A}$.

$$\begin{aligned} a_{21} &= \sum \ell_{2k}u_{k1} = \ell_{21}u_{11} + \ell_{22}u_{21} \Rightarrow \ell_{21} = a_{21}/u_{11} \\ a_{31} &= \sum \ell_{3k}u_{k1} = \ell_{31}u_{11} + \ell_{32}u_{21} + \ell_{33}u_{31} \quad \Rightarrow \ell_{31} = a_{31}/u_{11} \\ &\dots \end{aligned}$$

Nous déterminons ainsi la première colonne de $\boldsymbol{L}$. Vous pouvez imaginer que la deuxième ligne de $\boldsymbol{U}$ s'obtient à partir de la deuxième ligne de $\boldsymbol{A}$, puis que la deuxième colonne de $\boldsymbol{L}$ découle de la deuxième colonne de $\boldsymbol{A}$, etc... Les formules générales s'écrivent :

$$\begin{aligned} u_{ij} &= a_{ij} - \sum_{k=1}^{i-1} \ell_{ik}u_{kj}; \quad j = i, i+1, \dots n; \\ \ell_{ij} &= \frac{1}{u_{jj}} \left[a_{ij} - \sum_{k=1}^{j-1} \ell_{ik}u_{kj} \right]; \quad i = j+1, j+2, \dots n, \end{aligned} \tag{6.18}$$

avec toujours $\ell_{ii} = 1$. En procédant dans cet ordre, on s'assure que les éléments de $\boldsymbol{L}$ et $\boldsymbol{U}$ qui figurent dans les membres de droite de (6.18) sont disponibles quand on en a besoin. Le rôle de pivot est tenu par l'élément u_{jj}, qui doit être non-nul. Une condition équivalente est que $\det(\boldsymbol{A}(1:k, 1:k)) \neq 0$ pour $1 \leq k \leq n-1$; autrement dit, toutes les sous-matrices que l'on peut extraire du coin supérieur gauche de $\boldsymbol{A}$ doivent être régulières. En pratique, non seulement le pivot ne doit pas être nul mais, comme cet algorithme est sensible aux erreurs d'arrondis, il faut choisir le plus grand élément de la colonne j comme pivot et effectuer les permutations de lignes correspondantes.

6.4.1. Variantes

Nous avons montré que la matrice régulière $\boldsymbol{A}$ admet la factorisation $\boldsymbol{LU}$ en un produit de matrices, triangulaire inférieure pour $\boldsymbol{L}$ à diagonale unitaire, triangulaire supérieure pour $\boldsymbol{U}$. Cette répartition des rôles n'est pas la seule possible. On peut concevoir la décomposition $\boldsymbol{A} = \boldsymbol{L'U'}$ où $\boldsymbol{U'}$ est maintenant triangulaire à diagonale unitaire, $\boldsymbol{L'}$ étant simplement triangulaire ; un algorithme dû à Doolittle et très voisin du précédent permet d'obtenir $\boldsymbol{L'}$ et $\boldsymbol{U'}$.

Plutôt que de favoriser l'un des facteurs grâce à la possession d'une diagonale unitaire, on peut imposer que chaque facteur triangulaire présente des éléments diagonaux égaux à 1. On a alors :

$$\boldsymbol{A} = \boldsymbol{LDU} \quad , \quad \ell_{ii} = u_{ii} = 1, \quad 1 \leq i \leq n$$

où $\boldsymbol{D}$ est une matrice diagonale. Cette factorisation est unique et sera obtenue sans permutation de lignes si toutes les sous-matrices du coin supérieur gauche de $\boldsymbol{A}$ sont régulières.

Dans le cas particulier où $\boldsymbol{A}$ est symétrique, il vient :

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T.$$

6.5. Matrices particulières

Il y a toujours intérêt à prendre en compte une structure particulière de matrice ou de système linéaire quand celle-ci existe. C'est pourquoi nous examinons quelques cas particuliers dans ce paragraphe.

6.5.1. Matrice à diagonale dominante

Nous dirons que la matrice $\boldsymbol{A}$, $n \times n$, est à diagonale strictement dominante si chaque élément de la diagonale est supérieur (en valeur absolue) à la somme des valeurs absolues des autres éléments de la même ligne :

$$|a_{ii}| > \sum_{j=1 j \neq i}^{n} |a_{ij}|.$$

On démontre qu'alors l'algorithme de Gauss (ou la factorisation LU) ne nécessite aucune permutation.

6.5.2. Matrice symétrique définie positive

Nous supposons que $\boldsymbol{A}$ est définie positive. Rappelons la définition : $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0$ quel que soit le vecteur $\boldsymbol{x}$ non nul. Cette matrice est donc régulière et jouit de nombreuses propriétés intéressantes, dont certaines sont résumées dans le théorème suivant.

> **Théorème** – Si $\boldsymbol{A}$ est définie positive, la factorisation $\boldsymbol{LDU}$ existe et les éléments de la matrice diagonale $\boldsymbol{D}$ sont positifs. De plus, $\boldsymbol{A}$ admet aussi la factorisation unique $\boldsymbol{A} = \boldsymbol{G}\boldsymbol{G}^T$, où $\boldsymbol{G}$ est une matrice triangulaire inférieure dont les éléments diagonaux sont positifs.

Cette décomposition est appelée factorisation de Cholesky. On détermine les éléments de $\boldsymbol{G}$ ($g_{ij} = 0$ si $j > i$) par identification dans la colonne j de $\boldsymbol{A}$. Le calcul de a_{ij} fait intervenir la colonne j de $\boldsymbol{G}^T$, dont tous les éléments sont nuls à partir du rang $j+1$.

$$a_{ij} = \sum_{k=1}^{j} g_{ik}g_{jk} = \sum_{k=1}^{j-1} g_{ik}g_{jk} + g_{ij}g_{jj}$$

ou encore

$$g_{ij}g_{jj} = a_{ij} - \sum_{k=1}^{j-1} g_{ik}g_{jk} \equiv v_i,$$

en définissant une composante d'un vecteur $\boldsymbol{v}$. Comme cette relation est valable quel que soit $i \geq j$, elle définit la partie utile de $\boldsymbol{v}$. Nous pourrons calculer ce vecteur à condition de connaître les $j-1$ premières colonnes de $\boldsymbol{G}$. La composante j de $\boldsymbol{v}$ vérifie $g_{jj}^2 = v_j$ d'où nous tirons

$$g_{ij} = \frac{v_i}{\sqrt{v_j}}, \quad j \leq i \leq n.$$

Le programme suivant est une traduction « concentrée » en Scilab de l'algorithme de Cholesky, les boucles sur l'indice i étant remplacées par l'opérateur « : ».

Listing 6.3 – principe de l'algorithme de Cholesky

```
// initialisation                                                    1
//   .........                                                       2
//                                                                   3
G = zeros(A);                                                        4
//factorisation                                                      5
for j = 1:N                                                          6
    v(j:N) = A(j:N,j);                                               7
    for k = 1:j-1                                                    8
        v(j:N) = v(j:N) - G(j,k)*G(j:N,k);                           9
    end                                                              10
    G(j:N,j)= v(j:N)/sqrt(v(j));                                     11
end                                                                  12
```

Pour la première itération ($j = 1$), $\boldsymbol{v}$ est identifié à la première colonne de $\boldsymbol{A}$ (ligne 7) ; la boucle sur k n'est pas parcourue ($j - 1 = 0$) et la première colonne de $\boldsymbol{G}$ est définie comme proportionnelle à $\boldsymbol{v}$ (ligne 11). Le programme se poursuit en calculant une nouvelle version de $\boldsymbol{v}$ à partir de la première colonne de $\boldsymbol{G}$ (et de la première ligne de $\boldsymbol{G}^T$). Nous vous laissons le soin de modifier ce programme pour ranger les g_{ij} dans le triangle inférieur de $\boldsymbol{A}$.

Lorsque la forme quadratique $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ est positive ou nulle pour $\boldsymbol{x}$ non nul, on dit que $\boldsymbol{A}$ est semi-définie positive. Il est possible d'adapter le programme précédent pour former la décomposition de Cholesky d'une telle matrice.

6.5.3. Matrice bande

Dans certaines applications, on rencontre des systèmes linéaires tels que l'inconnue x_i n'apparaisse que dans quelques équations « proches » de l'équation de rang i. La matrice des coefficients a alors une structure en « bande ». $\boldsymbol{A}$ a une « largeur de bande supérieure » égale à q si $a_{ij} = 0$ quand $j > i + q$; de même, elle a une « largeur de bande inférieure » de valeur p si $a_{ij} = 0$ quand $i > j + p$. On rencontre souvent le cas particulier $p = q = 1$: la matrice est alors dite « tridiagonale ».

On démontre que si $\boldsymbol{A}$ présente une structure de bande et si $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$, alors, en l'absence de permutation de lignes, $\boldsymbol{L}$ hérite de la largeur de bande inférieure (p), tandis que $\boldsymbol{U}$ aura une largeur de bande supérieure égale à q, ce que nous représentons schématiquement, dans le cas $n = 5, p = 1, q = 2$, comme ceci, les croix représentant

des éléments non nuls

$$\begin{bmatrix} \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & 0 \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \times & 1 & 0 & 0 & 0 \\ 0 & \times & 1 & 0 & 0 \\ 0 & 0 & \times & 1 & 0 \\ 0 & 0 & 0 & \times & 1 \end{bmatrix} \begin{bmatrix} \times & \times & \times & 0 & 0 \\ 0 & \times & \times & \times & 0 \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

Les algorithmes de Gauss ou de factorisation peuvent facilement être modifiés pour tirer parti de la structure de $\boldsymbol{A}$.

6.5.4. Système tridiagonal

De nombreuses applications conduisent à un système d'équations linéaires tridiagonal à diagonale dominante ; nous savons que, dans ces conditions, la matrice des coefficients admet une décomposition LU sans permutation de lignes de la forme

$$\begin{bmatrix} a_1 & c_1 & 0 & \dots & \dots & 0 \\ b_2 & a_2 & c_2 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & b_{n-1} & a_{n-1} & c_{n-1} \\ 0 & \dots & \dots & 0 & b_n & a_n \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ \ell_2 & 1 & 0 & \dots & 0 \\ 0 & \dots & 1 & \dots & 0 \\ 0 & \dots & \ell_{n-1} & 1 & 0 \\ 0 & \dots & 0 & \ell_n & 1 \end{bmatrix} \begin{bmatrix} v_1 & u_1 & 0 & \dots & 0 \\ 0 & v_2 & u_2 & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & v_{n-1} & u_{n-1} \\ 0 & \dots & 0 & 0 & v_n \end{bmatrix}$$

L'identification fournit les relations suivantes :

$$\begin{aligned} &a_1 = v_1, \quad c_1 = u_1, \\ &c_i = u_i, \quad 1 \leq i \leq n-1; \\ &b_i = \ell_i v_{i-1}, \\ &a_i = \ell_i u_{i-1} + v_i, \quad 2 \leq i \leq n \end{aligned}$$

et les relations inverses

$$\begin{aligned} &v_1 = a_1, \quad u_1 = c_1, \\ &u_i = c_i, \quad 1 \leq i \leq n-1; \\ &\ell_i = b_i / v_{i-1}, \\ &v_i = a_i - \ell_i u_{i-1}, \quad 2 \leq i \leq n. \end{aligned} \tag{6.19}$$

La résolution du système $\boldsymbol{LUx} = \boldsymbol{s}$ se fait, comme d'habitude, en deux étapes, d'abord $\boldsymbol{Ly} = \boldsymbol{s}$ puis $\boldsymbol{Ux} = \boldsymbol{y}$, selon les formules

$$\begin{aligned} &y_1 = s_1, \\ &y_i = s_i - \ell_i y_{i-1}, \quad 2 \leq i \leq n; \\ &x_n = y_n / v_n, \\ &x_i = (y_i - u_i x_{i+1}) / v_i, \quad n-1 \geq i \geq 1. \end{aligned} \tag{6.20}$$

Une matrice $n \times n$ tridiagonale possède au plus $3n - 2$ éléments non nuls, qui représentent, si n est grand, une petite fraction des n^2 éléments théoriques. Il est recommandé, pour économiser de la mémoire, de ranger, comme nous l'avons fait, ces éléments dans trois tableaux à une dimension. La discrétisation des équations aux dérivées partielles fait souvent apparaître de très grandes matrices dites « creuses » (« sparse » en anglais) car comportant très peu d'éléments différents de zéro. Il existe des programmes spécialisés pour les manipuler sans encombrer la mémoire.

6.6. Méthodes itératives de résolution des systèmes linéaires

Les très grands systèmes linéaires posent un problème pratique : il peut être impossible de caser en mémoire centrale l'ensemble des coefficients. D'autre part, on rencontre, surtout lorsque le système linéaire provient de la discrétisation d'une équation aux dérivées partielles, une structure particulière des coefficients : les éléments diagonaux sont plus grands que tous les autres. Dans ces cas là, on peut utiliser avec profit des méthodes itératives de résolution, beaucoup moins gourmandes en mémoire et plus rapides.

6.6.1. Méthode de Jacobi

Nous cherchons la solution de (6.1), avec $a_{ii} \neq 0$, $1 \leq i \leq n$, en connaissant une approximation de la solution : $\boldsymbol{x}^{(0)}$. Nous allons utiliser une méthode d'approximations successives définie par les relations

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right) ; \quad 1 \leq i \leq n. \tag{6.21}$$

Le calcul de $x_i^{(k+1)}$ n'utilise que la ligne i de $\boldsymbol{A}$: cette matrice peut donc être rangée sur disque et lue ligne par ligne à la demande. Vous remarquez que nous n'utilisons pas l'information la plus récente pour calculer $x_i^{(k+1)}$.

Comme pour toute méthode d'itération, nous devons nous préoccuper d'un critère d'arrêt. On rencontre couramment l'une des deux définitions classiques suivantes. Définissons le « résiduel » à l'itération k :

$$\boldsymbol{r}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}. \tag{6.22}$$

Nous pouvons choisir la condition de convergence

$$\| \boldsymbol{r}^{(k)} \| \leq \epsilon \| \boldsymbol{b} \|,$$

ϵ étant un seuil fixé à l'avance ou la condition :

$$\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \| \leq \epsilon \| \boldsymbol{x}^{(k)} \| .$$

L'algorithme (6.21) admet une représentation matricielle commode pour les analyses théoriques. Décomposons $\boldsymbol{A}$ en une **somme** de trois matrices (qui n'ont rien à voir avec les $\boldsymbol{L}$ et $\boldsymbol{U}$ des paragraphes précédents) :

$$\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U}$$

où $\boldsymbol{D}$ est diagonale, $\boldsymbol{L}$ strictement triangulaire inférieure et $\boldsymbol{U}$ strictement triangulaire supérieure (et donc $a_{ii} = d_{ii}, \ell_{ii} = u_{ii} = 0$, $1 \leq i \leq n$). Comme $\boldsymbol{D}$ est diagonale, son inverse est également diagonale et ses éléments s'écrivent $[\boldsymbol{D}^{-1}]_{ii} = 1/d_{ii} = 1/a_{ii}$. Les équations (6.21) peuvent alors se résumer en

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\boldsymbol{x}^{(k)} + \boldsymbol{D}^{-1}\boldsymbol{b}. \tag{6.23}$$

Exemple – Étant donné le système linéaire ci-dessous et le vecteur initial $(0,0)^T$:

$$\left\{ \begin{array}{rcrcl} 3x & + & y & = & 2 \\ -x & + & 2y & = & -2 \end{array} \right. \quad \text{ou} \quad \left\{ \begin{array}{rcl} x & = & (1/3)(2-y) \\ y & = & \frac{1}{2}(x-2) \end{array} \right. ,$$

nous trouvons la suite de solutions approchées suivante :

x	2/3	1	8/9	5/6	23/27	31/36	139/162
y	-1	$-2/3$	$-1/2$	$-5/9$	$-7/12$	$-31/54$	$-41/72$

qui converge effectivement mais lentement vers la solution exacte $(6/7, -4/7)^T$. Le code suivant résout un système linéaire par la méthode itérative de Jacobi.

Listing 6.4 – Méthode de Jacobi pour les systemes linéaires

```
//    ... lecture de seuil, kmax, n, A et b ...                    1
//initialisation                                                   2
x = ones(n,1);                                                     3
//resolution                                                       4
for k = 1:kmax                                                     5
    for i = 1:n                                                    6
        s = b(i) - A(i,:)*x;                                       7
        y(i) = x(i) + s/A(i,i);                                    8
    end                                                            9
    if max(abs(y-x)) < seuil, break, end                           10
    x = y;                                                         11
end                                                                12
//affichage                                                        13
k, x', (A*x - b)'                                                  14
```

À chaque tour dans la boucle, `x` contient l'ancienne valeur $x^{(k)}$ et `y` contient la nouvelle valeur $x^{(k+1)}$ du vecteur inconnu. Plutôt que d'utiliser directement (6.21), il est commode de calculer $\sum_1^n a_{ij}x_j^{(n)}$ (ligne 7). Cela revient à calculer une composante du produit $\boldsymbol{Ax^{(k)}}$, celle-ci contient un terme de trop, ce que nous corrigons à la ligne suivante (ligne 8).

La méthode de Jacobi converge lentement, mais nous pouvons facilement modifier cet algorithme pour le rendre plus rapide.

6.6.2. Méthode de Gauss–Seidel

L'algorithme de Jacobi nous prescrit de calculer $x_i^{(k+1)}$ en utilisant les valeurs $x_1^{(k)}, x_2^{(k)}$, $\ldots$, $x_{i-1}^{(k)}, x_{i+1}^{(k)}, \ldots x_n^{(k)}$; il semble que nous pourrions gagner du temps ou de la précision, puisque nous connaissons déjà $x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_{i-1}^{(k+1)}$, qui sont plus précises que les valeurs des mêmes variables à l'itération précédente (k). C'est ce que fait l'algorithme de Gauss–Seidel. Les relations de récurrence s'écrivent :

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \tag{6.24}$$

Pour découvrir l'équivalent matriciel des équations (6.24), il suffit de les « déplier » pour les mettre sous la forme équivalente

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

Le membre de gauche représente la ligne i du produit $(-\boldsymbol{L} + \boldsymbol{D})\boldsymbol{x}^{(k+1)}$, tandis que le membre de droite est la ligne i de $\boldsymbol{b} + \boldsymbol{U}\boldsymbol{x}^{(k)}$. Le passage de $\boldsymbol{x^{(k)}}$ à $\boldsymbol{x^{(k+1)}}$ peut donc s'écrire

$$(\boldsymbol{D} - \boldsymbol{L})\boldsymbol{x}^{(k+1)} = \boldsymbol{U}\boldsymbol{x}^{(k)} + \boldsymbol{b}. \tag{6.25}$$

Exemple – Appliquant l'algorithme de Gauss–Seidel à l'exemple du paragraphe précédent, nous trouvons la suite de valeurs :

x	$2/3$	$8/9$	$23/27$	$139/162$
y	$-2/3$	$-5/9$	$-31/54$	$-185/324$

Certains de ces nombres sont apparus lors du calcul précédent ; on observe toutefois que la convergence est beaucoup plus rapide.

Les équations de l'algorithme de Gauss–Seidel sont peut-être plus faciles à programmer que celles qui découlent de la méthode de Jacobi. En effet, il n'est pas nécessaire de conserver l'ensemble des valeurs $x_i^{(k)}$ pendant le calcul des $x_i^{(k+1)}$: chaque nouvelle valeur remplace la précédente. De plus, comme dans le programme précédent, nous calculons la composante i de $\boldsymbol{Ax^{(k)}}$ (ligne 7), qui contient un terme de trop, ce qui sera compensé à la ligne suivante. Voici un programme qui réalise l'itération de Gauss–Seidel. Le vecteur $\boldsymbol{y}$ ne sert qu'à effectuer le test de convergence.

Listing 6.5 – Méthode de Gauss–Seidel

```
// .... lecture de seuil, kmax, n, A, b                    1
//initialisation                                           2
x = ones(n,1); y = x;                                       3
//resolution                                               4
for k = 1:kmax                                              5
 for i = 1:n                                                6
```

```
  s = A(i,:)*x;
  x(i) = x(i) + (b(i) - s)/A(i,i);
 end
 if max(abs(y-x)) < seuil, break, end
 y = x;
end
k, x', (A*x - b)'
```

6.6.3. Méthode de surrelaxation

Cette variante de l'algorithme de Gauss–Seidel (elle peut utiliser Jacobi aussi, mais sans avantage) procède comme suit. Connaissant un vecteur $\boldsymbol{x}^{(k)}$, nous calculons un vecteur provisoire $\hat{\boldsymbol{x}}^{(k+1)}$ à l'aide de l'algorithme (6.24), puis nous définissons le vecteur $\boldsymbol{x}^{(k+1)}$ définitif comme une combinaison linéaire du nouveau et de l'ancien vecteur :

$$\boldsymbol{x}^{(k+1)} = \omega\hat{\boldsymbol{x}}^{(k+1)} + (1-\omega)\boldsymbol{x}^{(k)}.$$

Soit en explicitant les composantes

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) + (1-\omega)x_i^{(k)}. \tag{6.26}$$

ω est un paramètre réel dont il faut optimiser la valeur pour chaque classe de problème. Le choix $\omega = 1$ nous ramène à (6.24). On a toujours $\omega < 2$. Réordonnons les relations précédentes :

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

Vous reconnaissez la ligne i de l'expression

$$(\boldsymbol{D} - \omega\boldsymbol{L})\boldsymbol{x}^{(k+1)} = [(1-\omega)\boldsymbol{D} + \omega\boldsymbol{U}]\boldsymbol{x}^{(k)} + \omega\boldsymbol{b}. \tag{6.27}$$

Le programme suivant met en oeuvre les équations (6.26). Chaque nouvelle valeur $x_i^{(k+1)}$ écrase la précédente ($x_i^{(k)}$).

Listing 6.6 – Exemple d'application de la méthode de surrelaxation

```
//    ...  lecture de seuil, kmax, n, A, b, omg
//initialisation
x = ones(n,1); y = x;
//resolution
for k = 1:kmax
    for i = 1:n
        s = A(i,:)*x;
        x(i) = x(i) + omg*(b(i) - s)/A(i,i);
    end
```

```
    if max(abs(y-x)) < seuil, break, end
    y = x;
end
k, x', (A*x - b)'
```

6.6.4. CONVERGENCE DES MÉTHODES ITÉRATIVES

Les trois algorithmes que nous venons de présenter peuvent tous s'écrire

$$\boldsymbol{M}\boldsymbol{x}^{(k+1)} = \boldsymbol{N}\boldsymbol{x}^{(k)} + \boldsymbol{b}, \tag{6.28}$$

où nous avons choisi une certaine décomposition de $\boldsymbol{A}$ sous la forme $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$. L'équation (6.28) représente un système d'équations linéaires pour les inconnues $x_i^{(k+1)}$ mais, grâce à un choix judicieux de la matrice $\boldsymbol{M}$, ce système se résout facilement par itération. Plus précisément

– pour Jacobi :

$$\boldsymbol{M} = \boldsymbol{D}, \quad \boldsymbol{N} = \boldsymbol{L} + \boldsymbol{U},$$

– pour Gauss–Seidel :

$$\boldsymbol{M} = \boldsymbol{D} - \boldsymbol{L}, \quad \boldsymbol{N} = \boldsymbol{U},$$

– pour la surrelaxation :

$$\boldsymbol{M} = \frac{1}{\omega}\boldsymbol{D} - \boldsymbol{L}, \quad \boldsymbol{N} = \left(\frac{1}{\omega} - 1\right)\boldsymbol{D} + \boldsymbol{U}.$$

Pour examiner la convergence de ces méthodes, définissons l'erreur à l'itération k comme

$$\boldsymbol{e}^{(k)} \equiv \boldsymbol{x}^{(k)} - \boldsymbol{x}. \tag{6.29}$$

La solution exacte $\boldsymbol{x}$ vérifie

$$\boldsymbol{M}\boldsymbol{x} = \boldsymbol{N}\boldsymbol{x} + \boldsymbol{b}$$

(d'après la décomposition de $\boldsymbol{A}$) et cela implique que l'erreur à l'itération $k+1$ est donnée par

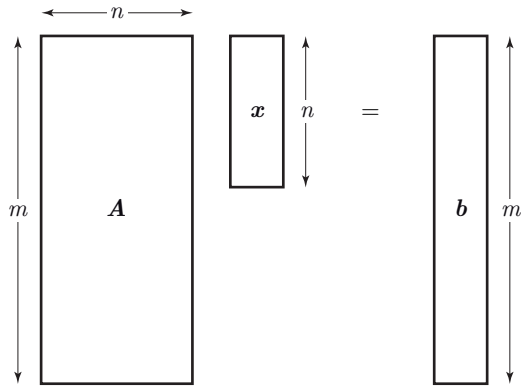
$$\boldsymbol{e}^{(k+1)} = \boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{e}^{(k)}.$$

La matrice $\boldsymbol{M}^{-1}\boldsymbol{N}$ est appelée matrice de relaxation de l'algorithme considéré. Pour savoir si cette erreur tendra vers zéro avec k, il faut examiner la plus grande des valeurs propres (en valeur absolue) de la matrice de relaxation (ce que l'on appelle le rayon spectral de $\boldsymbol{M}^{-1}\boldsymbol{N}$, noté $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$). On peut alors énoncer le théorème suivant.

Théorème – Soit $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ une matrice $n \times n$ régulière. Si la matrice $\boldsymbol{M}$ est régulière et si le rayon spectral de la matrice $\boldsymbol{M}^{-1}\boldsymbol{N}$ est inférieur à 1, alors le vecteur itéré défini par (6.28) converge vers la solution exacte $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ quelle que soit l'approximation de départ $\boldsymbol{x}^{(0)}$. Cette condition est remplie dans les deux cas particuliers suivants.
Pour la méthode de Jacobi, une condition suffisante de convergence est que $\boldsymbol{A}$ soit à diagonale dominante.
Si $\boldsymbol{A}$ est symétrique définie positive, alors l'itération de Gauss–Seidel converge.
La convergence de la méthode de surrelaxation doit être démontrée au cas par cas.

6.7. Système surdéterminé

Un système, où le nombre d'équations est supérieur au nombre d'inconnues, se rencontre souvent en pratique ; on dit que le système est surdéterminé. Dans l'équation (6.1), on a ainsi $m > n$ ou encore, dans la relation (6.2), la matrice $\boldsymbol{A}$ est « haute et étroite », comme ci-dessous Ce schéma représente une application de $\mathbb{R}^n$ dans $\mathbb{R}^m$.



Nous supposons que la matrice $\boldsymbol{A}$ est de rang maximal (ou encore que ses n vecteurs colonnes sont linéairement indépendants). L'image de la matrice $\boldsymbol{A}$ est alors un sous-espace de dimension n inclus dans $\mathbb{R}^m$ que nous désignons par $\mathcal{S}$. Si $\boldsymbol{b} \in \mathcal{S}$, le système admet une solution exacte. Que faire dans le cas contraire ? Depuis Gauss, on admet que la meilleure (ou la moins mauvaise) solution dans ce cas est celle que fournit la méthode des moindres carrés.

Étant donné un vecteur quelconque $\boldsymbol{x} \in \mathbb{R}^m$, nous définissons le vecteur résiduel (cf. (6.22)) comme $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$. La solution de (6.1), au sens des moindres carrés, est le vecteur $\boldsymbol{x}_0$ qui rend le vecteur résiduel aussi petit que possible. Autrement dit, $\boldsymbol{x}_0$ minimise la fonction

$$f(\boldsymbol{x}) = \| \boldsymbol{r} \|^2 = \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \|^2 .$$

Nous choisissons ici la norme euclidienne des vecteurs, qui rend les calculs à venir beaucoup plus commodes.

En faisant un schéma à 2 ou 3 dimensions, vous pouvez vous convaincre que la distance qui sépare $\boldsymbol{Ax}$ de $\boldsymbol{b}$ sera minimale si $\boldsymbol{r}$ est perpendiculaire à $\mathcal{S}$, ou encore si $\boldsymbol{r}$ est perpendiculaire à n'importe quel vecteur de $\mathcal{S}$. D'après la définition de $\mathcal{S}$, un vecteur de ce sous-espace est de la forme $\boldsymbol{y} = \boldsymbol{Au}, \boldsymbol{u} \in \mathbb{R}^m$. Nous imposons donc que, quel que soit $\boldsymbol{u}$, le produit scalaire de $\boldsymbol{r}$ par $\boldsymbol{y}$ soit nul :

$$(\boldsymbol{Au})^T(\boldsymbol{Ax}_0 - \boldsymbol{b}) = \boldsymbol{u}^T\boldsymbol{A}^T(\boldsymbol{Ax}_0 - \boldsymbol{b}) = 0.$$

Comme $\boldsymbol{A}$ est de rang maximal, nous concluons que le facteur qui multiplie $\boldsymbol{u}$ doit être identiquement nul :

$$\boldsymbol{A}^T\boldsymbol{Ax}_0 = \boldsymbol{A}^T\boldsymbol{b}. \tag{6.30}$$

Nous avons abouti à un système de n équations à n inconnues (les composantes de $\boldsymbol{x}_0$), appelées équations normales ou équations de Gauss. Le tableau des coefficients est la matrice $n \times n$, définie positive $\boldsymbol{A}^T\boldsymbol{A}$.

Le principal champ d'application de la méthode des moindres carrés concerne le traitement des données expérimentales. Nous y reviendrons au chapitre 14. Les résultats expérimentaux sont entachés d'erreurs aléatoires ; que peut-on dire alors d'une solution au sens des moindres carrés ? Cet aspect probabiliste sera abordé à cette occasion.

6.8. Pour en savoir plus

De nombreux algorithmes n'ont pas été mentionnés dans le texte. Les plus connus, que vous pourrez trouver dans les livres cités ou sur la Toile, sont : la méthode du gradient conjugué (systèmes à matrice symétrique définie positive) et la méthode GMRES, qui peuvent être associées à un « préconditionnement ».

– R. Théodor : *Initiation à l'analyse numérique*, ch. 2, (Masson, Paris, 1994).

– P. Lascaux, R. Théodor : *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, tomes I et II (Masson, Paris, 1993).

– J.G. Dion, R. Gaudet : *Méthodes d'analyse numérique – de la théorie à l'application*, ch. 7–9 (Modulo, Mont-Royal, Québec, 1996).

– W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vettering : *Numerical Recipes, the art of scientific computing*, ch. 2 (Cambridge University Press, Cambridge, 2007).

– G. Allaire, S.M. Kaber : *Algèbre linéaire numérique* (Ellipse, Paris, 2002).
G. Allaire, S.M. Kaber : *Introduction à Scilab. Exercices pratiques corrigés d'algèbre linéaire* (Ellipses, Paris, 2002).

– M. Schatzman : *Analyse numérique, une approche mathématique*, ch. 9 (Dunod, Paris, 2001).

– C. Brezinski, M. Redivo-Zaglia : *Méthodes numériques directes de l'algèbre matricielle - Niveau L3* (Ellipses, Paris, 2004).

– C. Brezinski, M. Redivo-Zaglia : *Méthodes numériques itératives – Niveau M1* (Ellipses, Paris, 2006).

– sur le site `http://www.librecours.org/` :
  – J.-P. Tignol : Algèbre linéaire.
  – O. Debarre : Algèbre, deuxième année.
– R. Herbin : Cours d'analyse numérique (L3), ch. 1, systèmes linéaires : `http://www.cmi.univ-mrs.fr/~herbin/`
– Polycopiés des cours d'analyse numérique de E. Hairer et G. Wanner : ch. 4, systèmes d'équations linéaires : `http://www.unige.ch/~hairer/polycop.html`

– `http://docs.ufrmd.dauphine.fr/lebourg/opti-iup1.html`

– Vie de Cholesky : `http://www.sabix.org/bulletin/b39/vie.html`

6.9. Exercices

Exercice 1

Soient $\boldsymbol{x} = [1, 2, 3]^T$ et $\boldsymbol{y} = [a, b, c]^T$ deux vecteurs colonnes. Former la matrice $\boldsymbol{P} = \boldsymbol{x}\boldsymbol{y}^T$. Quel est le rang de cette matrice ? Calculer $\boldsymbol{P}^2$. Quelle est l'image de $\boldsymbol{P}$?

Exercice 2

Soient $\boldsymbol{L}$ une matrice régulière triangulaire inférieure d'ordre N et $\boldsymbol{b}$ un vecteur de dimension N, tel que $b_i = 0$ pour $i < k$ et $b_k \neq 0$. Montrer que la solution $\boldsymbol{x}$ de l'équation $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$ est telle que $x_i = 0$ pour $i < k$ et $x_k = b_k/L_{kk}$. Déduire de ce qui précède que si $\boldsymbol{L}$ est une matrice régulière triangulaire inférieure, son inverse $\boldsymbol{L}^{-1}$ a la même structure. Quels sont les éléments diagonaux de $\boldsymbol{L}^{-1}$?

Exercice 3

Soient $\boldsymbol{L}$ et $\boldsymbol{M}$ deux matrices carrées triangulaires inférieures d'ordre N. Montrer que le produit $\boldsymbol{L}\boldsymbol{M}$ est également triangulaire inférieur. Déduire de ce résultat la propriété : si $\boldsymbol{A}$ est une matrice régulière d'ordre N qui possède une décomposition LU (avec $\ell_{ii} = 1$, $i = 1, 2 \ldots, N$, alors cette décomposition est unique.

Exercice 4

On considère la matrice $n \times n$ de Frobenius $\boldsymbol{f}(k, \ell, x)$ qui se déduit de la matrice identité d'ordre n par adjonction du réel x en ligne k et colonne ℓ.

a) Exprimer l'élément i, j de $\boldsymbol{f}(k, \ell, x)$ en fonction des symboles de Kronecker $\delta_{ij}, \delta_{i\ell}$ et δ_{kj}.

b) Calculer $\boldsymbol{f}(k, \ell, x)\boldsymbol{f}(k', \ell, y)$ en utilisant les propriétés des δ_{ij}.

Exercice 5

On donne la matrice

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

a) Former la décomposition $\boldsymbol{LU}$ de $\boldsymbol{A}$, où $\boldsymbol{L}$ est une matrice triangulaire inférieure dont les éléments diagonaux sont égaux à l'unité et $\boldsymbol{U}$ est une matrice triangulaire supérieure. Calculer $D = \det \boldsymbol{A}$.

b) Utiliser la factorisation précédente pour résoudre le système linéaire $\boldsymbol{Ax} = \boldsymbol{b}$, avec $\boldsymbol{b} = [1, 1, 1, 1]^T$.

c) Former la décomposition de Cholesky de $\boldsymbol{A}$, qui s'écrit $\boldsymbol{A} = \boldsymbol{BB}^T$, avec $\boldsymbol{B}$ une matrice triangulaire inférieure dont les éléments diagonaux peuvent être quelconques. À l'aide de ce résultat, calculer à nouveau le déterminant D.

Exercice 6

On donne les deux systèmes linéaires :

$$\begin{bmatrix} 10 & 1 \\ 2 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 12 \end{bmatrix} \quad \text{(A)} \; ; \quad \begin{bmatrix} 1 & 10 \\ 10 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 12 \end{bmatrix} \quad \text{(B)}$$

a) Trouver les solutions exactes.

b) Résoudre chaque système par la méthode itérative de Jacobi, à partir du vecteur initial $x^{(0)} = (0, 0)^T$, jusqu'au vecteur $x^{(4)}$ inclus.

c) Expliquer qualitativement le comportement différent des deux suites de vecteurs.

d) Résoudre les systèmes (A) et (B) de la question par la méthode itérative de Gauss–Seidel, en partant du vecteur nul et en poussant le calcul jusqu'à $\boldsymbol{x}^{(3)}$ inclus.

Exercice 7

On veut résoudre, par la méthode de Gauss–Seidel, le système :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}.$$

Soit X, Y la solution exacte et $x^{(k)}, y^{(k)}$ les valeurs obtenues à l'itération k. On pose :

$$\Delta^{(k)}x = X - x^{(k)}, \qquad \Delta^{(k)}y = Y - y^{(k)}.$$

Exprimer $\Delta^{(k)}x, \Delta^{(k)}y$ en fonction des mêmes quantités à l'étape $k-1$. En déduire $\Delta^{(k)}x$ en fonction de $\Delta^{(k-1)}x$, puis une condition nécessaire de convergence. Donner une interprétation géométrique de cette condition.

6.10. Projet

Simulation d'une usine chimique

Le modèle d'une usine de transformations chimiques comporte cinq unités notées U1 à U5 sur la figure (6.10). Le flux d'entrée (appelé S1) est composé d'un mélange des corps A, B et C dont les débits respectifs sont par exemple de S1A = 40 kg/h pour A, S1B = 40 kg/h pour B et S1C = 20 kg/h pour C. Le but de l'usine est de convertir une partie de A et un peu de B en C et d'effectuer ensuite une séparation partielle pour que le flux de sortie (S7) contienne surtout le produit C. On étudie le régime stationnaire. Chaque débit est la somme des débits partiels des trois composés : Sn = SnA + SnB + SnC, avec $n = 1 \ldots 9$.

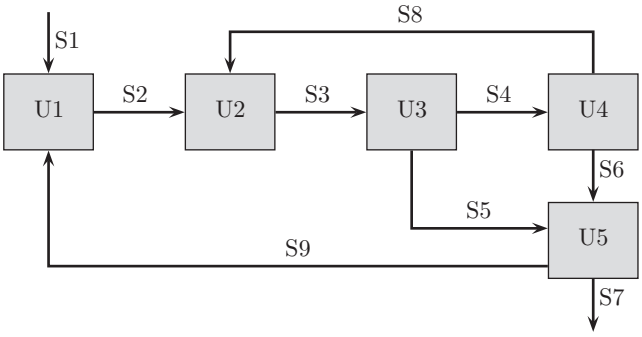


Figure 6.3 – Schéma de principe d'une usine chimique.

La première unité est un mélangeur-réchaufeur. Elle reçoit les flux d'entrée connus S1A, S1B et S1C, des flux recyclés depuis U5 (débits S9A, S9B, S9C) et fournit à U2 un flux de sortie sans transformation chimique caractérisé par les débits S2A, S2B et S2C.

L'unité 2 est un réacteur chargé de convertir une partie de A et B en C. Le flux d'entrée est la réunion de S2 (provenant de U1) et d'un flux S8 provenant de U4 et contenant les trois produits. Les rendements de réaction sont $k_{AC} \simeq 0,4$ pour la réaction A$\Longrightarrow$C et $k_{BC} \simeq 0,5$ pour la conversion B$\Longrightarrow$C. Le C formé, les fractions de A et de B qui n'ont pas réagi forment le flux de sortie S3.

U3 est un évaporateur. Le point d'ébullition de C est inférieur à ceux de A et B. Le liquide pénétrant dans U3 est partiellement vaporisé et la vapeur est enrichie en C. Elle est dirigée vers U5 avec un débit S5 tandis que le liquide restant est envoyé dans U4 (débit S4). On connaît, pour chaque espèce, la répartition entre phase liquide et phase vapeur (coefficient de partage) : k_A = S5A/S4A = 0,1, k_B = S5B/S4B = 0,2 et k_C = S5C/S4C= 1,2.

L'unité 4 est un décanteur-condenseur. Le liquide obtenu par refroidissement de la vapeur riche en A et B est partiellement recyclé vers U2 (flux S8), le reste est dirigé vers U5 (flux S6). Pour chaque espèce, on a les rapports de débit ℓ_A = S6A/S8A = 1,5, ℓ_B = S6B/S8B = 2, ℓ_C = S6C/S8C = 4.

U5 est une unité de distillation dont le rôle est similaire à celui de U3 : la sortie S7 doit contenir une plus grande proportion de C que l'entrée (S6+S5), elle représente le produit final. Le reste est recyclé vers U1 avec un débit S9. On connaît ici aussi les rapports m_A = S7A/S9A = 0,18, m_B = S7B/S9B = 0,25 et m_C = S7C/S9C = 9.

Le problème comporte 24 inconnues (les débits partiels SnA,SnB et SnC, n = 2..9) et 24 relations. Écrire les équations de conservation correspondantes et les résoudre.

On suppose que le fonctionnement de l'évaporateur U3 serait moins coûteux si le coefficient de partage k_C était ramené à 0,8. Comment seraient modifiées la pureté de C et la quantité finale produite ?

6.11. Annexe : rappels d'algèbre linéaire

Nous rappelons ici quelques définitions et théorèmes d'algèbre linéaire utiles pour suivre les développements de ce chapitre, en nous limitant aux espaces vectoriels réels.

6.11.1. Base et sous-espace

Soient $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$ k vecteurs de $\mathbb{R}^n$ (avec $k \leq n$). Ils sont dits linéairement indépendants si $\sum_1^n \alpha_i \boldsymbol{v}_i = 0$ implique $\alpha_i = 0$, $1 \leq i \leq n$. Ils sont linéairement dépendants dans le cas contraire. L'ensemble de toutes les combinaisons linéaires de la forme $\sum_1^k \beta_i \boldsymbol{v}_i$ est un sous-espace de $\mathbb{R}^m$; c'est le sous-espace $\mathcal{S}$ « engendré » par les $\boldsymbol{v}_i$. Si ces vecteurs sont linéairement indépendants, ils forment une base de $\mathcal{S}$, dont la dimension est alors k. On dit que $\mathcal{S}$ est « sous-tendu » par les $\boldsymbol{v}_i$. Si $k = n$, $\mathcal{S}$ se confond avec $\mathbb{R}^n$ et les n vecteurs indépendants forment une base de $\mathbb{R}^n$.

6.11.2. Image, noyau et rang

Une matrice rectangulaire $\boldsymbol{A}$, $m \times n$ (m lignes, n colonnes), représente une application linéaire de $\mathbb{R}^n$ dans $\mathbb{R}^m$, dès lors que nous avons choisi une base dans chacun de ces espaces. À un vecteur $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ de $\mathbb{R}^n$ correspond un vecteur $y = [y_1, y_2, \ldots, y_m]$ de $\mathbb{R}^m$ selon la relation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ ou, en termes de composantes

$$y_i = \sum_{j=1}^{n} a_{ij} x_j \quad 1 \leq i \leq n,$$

où l'élément a_{ij} de $\boldsymbol{A}$ se trouve en j-ième position dans la ligne i. Il est souvent utile d'écrire plutôt

$$\boldsymbol{y} = \sum_{i=1}^{n} x_i \boldsymbol{a}_i$$

où $\boldsymbol{a}_i$ est le vecteur colonne de rang i de $\boldsymbol{A}$.

« L'image » de $\boldsymbol{A}$ (notée Im$\boldsymbol{A}$) est l'ensemble des vecteurs $\boldsymbol{y}$ obtenus lorsque $\boldsymbol{x}$ « balaie » $\mathbb{R}^n$; d'après la relation précédente, c'est l'espace engendré par les vecteurs

colonnes de $\boldsymbol{A}$ (un sous-espace de $\mathbb{R}^m$). Le « rang » de $\boldsymbol{A}$ est la dimension de l'image, c'est-à-dire le nombre de vecteurs colonne linéairement indépendants ; en symboles :

$$\text{rang}(\boldsymbol{A}) = \dim(\text{Im}\boldsymbol{A}).$$

Le noyau de $\boldsymbol{A}$ (Ker($\boldsymbol{A}$))est l'ensemble des vecteurs $\boldsymbol{x}$ pour lesquels on a $\boldsymbol{Ax} = 0$. C'est un sous-espace de $\mathbb{R}^n$. On a la relation

$$\text{rang}\boldsymbol{A} + \dim(\text{Ker}(\boldsymbol{A})) = n.$$

6.11.3. Inverse et déterminant

À toute matrice carrée $\boldsymbol{A}$ d'ordre n, on associe un nombre, son déterminant, noté $\det(\boldsymbol{A})$ qui vaut

$$\det(\boldsymbol{A}) \equiv \sum_P (-1)^p a_{1,j(1)} a_{2,j(2)} \cdots a_{n,j(n)}.$$

P désigne l'ensemble de $n!$ permutations des entiers $1, 2, \ldots, n$; les nombres $j(1), j(2), \ldots, j(n)$ représentent l'une de ces permutations, dont la « signature » est p. $(-1)^p = 1$ si la permutation peut se ramener à un nombre pair d'échanges et $(-1)^p = -1$ si la permutation est un produit d'un nombre impair d'échanges.

Voici quelques propriétés de $\det(\boldsymbol{A})$.

$$\begin{aligned}
&\det(\boldsymbol{AB}) = \det(\boldsymbol{A})\det(\boldsymbol{B}) \\
&\det(\boldsymbol{A}^T) = \det(\boldsymbol{A}) \\
&\det(c\boldsymbol{A}) = c^n \det(\boldsymbol{A}) \\
&\det(\boldsymbol{A}^{-1}) = \frac{1}{\det(\boldsymbol{A})}, \text{ si } \det(\boldsymbol{A}) \text{ n'est pas nul.}
\end{aligned}$$

Une matrice carrée $n \times n$ de rang n (dont le noyau est 0) admet une matrice inverse $\boldsymbol{A}^{-1}$ qui vérifie

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$$

où $\boldsymbol{I}$ est la matrice unité $n \times n$. D'après les propriétés du déterminant, l'inverse de $\boldsymbol{A}$ existe si et seulement si $\det(\boldsymbol{A}) \neq 0$. Dans ce cas la matrice est dite régulière ou inversible.

6.11.4. Normes vectorielles

Une norme est une application f de $\mathbb{R}^n$ dans $\mathbb{R}$ qui vérifie les trois propriétés

$$\begin{aligned}
&f(\boldsymbol{x}) \geq 0 \text{ et } f(\boldsymbol{x}) = 0 \text{ ssi } \boldsymbol{x} = 0 \\
&f(\boldsymbol{x} + \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) \\
&f(c\boldsymbol{x}) = |c| f(\boldsymbol{x})
\end{aligned}$$

et que l'on note $\| x \|$. Les normes les plus employées sont appelées les « p-normes » ; elles répondent à la définition

$$\| x \|_p \equiv \left(\sum_1^n |x_i|^p \right)^{1/p}.$$

Les cas p = 1, 2 et ∞ sont les plus importants

$$\| x \|_1 \equiv \sum_1^n |x_i|,$$
$$\| x \|_2 \equiv \left(\sum_1^n |x_i|^2 \right)^{1/2},$$
$$\| x \|_\infty \equiv \max_{1 \le i \le n} |x_i|.$$

La norme euclidienne ($p = 2$) vérifie l'inégalité de Cauchy–Schwartz

$$|\boldsymbol{x}^T \boldsymbol{y}| \le \| x \|_2 \| y \|_2 .$$

6.11.5. Normes de matrices

Une norme matricielle doit vérifier les mêmes propriétés générales qu'une norme vectorielle ; c'est une application f de $\mathbb{R}^{m \times n}$ dans $\mathbb{R}$, notée $\| \boldsymbol{A} \|$, telle que

$$f(\boldsymbol{A}) \ge 0 \text{ et } f(\boldsymbol{A}) = 0 \text{ ssi } \boldsymbol{A} = 0,$$
$$f(\boldsymbol{A} + \boldsymbol{B}) \le f(\boldsymbol{A}) + f(\boldsymbol{B}),$$
$$f(c\boldsymbol{A}) = |c| f(\boldsymbol{A}),$$
$$f(\boldsymbol{A}\boldsymbol{B}) \le f(\boldsymbol{A}) f(\boldsymbol{B}).$$

Les normes les plus fréquemment utilisées sont les normes induites (ou subordonnées) et la norme « de Frobenius » ou « de Schur ». Les premières répondent à la définition

$$\| \boldsymbol{A} \|_p = \sup_{\boldsymbol{x} \ne 0} \frac{\| \boldsymbol{A}\boldsymbol{x} \|_p}{\| \boldsymbol{x} \|_p}.$$

Le plus souvent, $p = 1, 2$ ou ∞.

La norme de Frobenius ressemble, par sa définition seulement, à la norme euclidienne :

$$\| \boldsymbol{A} \|_F \equiv \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2}.$$

Citons quelques propriétés des normes induites, pour $\boldsymbol{A} \in \mathbb{R}^{m\times n}$:

$$\begin{aligned}
&\| \boldsymbol{Ax} \|_p \leq \| \boldsymbol{A} \|_p \| \boldsymbol{x} \|_p, \\
&\| \boldsymbol{A} \|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} |a_{ij}|, \\
&\| \boldsymbol{A} \|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{ij}|.
\end{aligned}$$

Si $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ est la matrice unité d'ordre n, alors $\| \boldsymbol{I} \|_p = 1$ quel que soit p, mais $\| \boldsymbol{I} \|_F = \sqrt{n}$.

6.11.6. Opérations sur des blocs

Considérons la matrice

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

Nous l'écrivons sous la forme

$$\left[\begin{array}{c:cc} a_{11} & a_{12} & a_{13} \\ \cdots & \cdots & \cdots \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array}\right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

avec les définitions

$$A_{11} \equiv [a_{11}]; \quad A_{12} \equiv [a_{12}\, a_{13}]; \quad A_{21} \equiv \begin{bmatrix} a_{21} \\ a_{31} \end{bmatrix}; \quad A_{22} \equiv \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}.$$

Nous dirons que la matrice $\boldsymbol{A}$ a été décomposée en blocs A_{ij}, $1 \leq i, j \leq 2$. Soit maintenant une deuxième matrice $\boldsymbol{B}$ de même forme que $\boldsymbol{A}$ et que nous décomposons de la même manière

$$\boldsymbol{B} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

avec, par exemple, $B_{12} \equiv [b_{12}\, b_{13}]$.

La matrice produit $\boldsymbol{C} = \boldsymbol{AB}$ admet, elle aussi, une décomposition en blocs

$$\boldsymbol{C} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}.$$

Vous vérifierez facilement que

$$\begin{aligned}
C_{11} &= A_{11}B_{11} + A_{12}B_{21}; \quad C_{12} = A_{11}B_{12} + A_{12}B_{22}, \\
C_{21} &= A_{21}B_{11} + A_{22}B_{21}; \quad C_{22} = A_{21}B_{12} + A_{22}B_{22}.
\end{aligned}$$

La quantité $A_{21}B_{12}$ est le produit d'un vecteur colonne par un vecteur ligne : le résultat est une matrice 2×2. Ce procédé s'étend sans difficulté aux matrices rectangulaires à condition de bien respecter les nombres de lignes et de colonnes pour les facteurs de chaque produit.

Au § 6.10.2, nous avons représenté le produit $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ sous la forme

$$\boldsymbol{y} = \sum_{i=1}^{n} x_i \boldsymbol{a}_i.$$

Nous avions alors décomposé la matrice $\boldsymbol{A}$ en considérant chaque colonne comme un bloc.

Chapitre 7

Polynômes orthogonaux

Les suites de polynômes orthogonaux apparaissent fréquemment en physique mathématique en particulier au cours de la résolution d'équations aux dérivées partielles (Laplace, Schrödinger) par la méthode de séparation des variables. Certaines d'entre elles sont aussi très utilisées en analyse numérique. Quelle que soit leur définition, l'orthogonalité impose que ces polynômes orthogonaux aient en commun un certain nombre de propriétés, en particulier celles de vérifier une relation de récurrence à trois termes et d'obéir à une équation différentielle linéaire du second ordre. Nous allons les décrire succinctement, en omettant le plus souvent les démonstrations.

7.1. Définition, existence

Soient $[a, b]$ un intervalle (appelé I dans la suite et qui peut être fini ou infini) et w une fonction strictement positive et intégrable sur cet intervalle. On appelle polynômes orthogonaux sur I, par rapport à la « fonction de poids » w, une suite de polynômes $G_0(x)$, $G_1(x)$,..., $G_n(x)$,... (où G_k est de degré k) tels que :

$$(G_k, G_n) \equiv \int_a^b w(x) G_k(x) G_n(x) dx = 0 \text{ si } k \neq n. \tag{7.1}$$

Le nombre

$$(f, g) \equiv \int f(x) g(x) w(x) dx \tag{7.2}$$

est souvent appelé « produit scalaire » de f par g ; lorsqu'il est nul, on dit que les fonctions f et g sont « orthogonales ». En faisant $f = g$ dans la relation précédente, on obtient le carré de la norme de f :

$$\| f \|^2 = (f, f) \equiv \int [f(x)]^2 w(x) dx. \tag{7.3}$$

Comme l'intégrale qui le définit, ce produit scalaire est linéaire et symétrique en f et g. On montre qu'il vérifie l'inégalité de Cauchy-Schwartz :

$$|(f,g)| \leq \| f \| \| g \| . \tag{7.4}$$

Les formules (7.1), (7.2) ne définissent pas entièrement les G_n ; il est commode d'imposer les conditions supplémentaires :

$$(G_n, G_n) = \int_a^b w[G_n]^2 dx = 1; \quad \int_a^b x w G_n G_{n+1} dx > 0.$$

On a alors :

$$(G_k, G_n) = \int_a^b w(x) G_k(x) G_n(x) dx = \delta_{k,n}.$$

Les polynômes sont ainsi orthonormés.

Pour prouver l'existence de ces polynômes, nous allons les construire en utilisant le procédé d'orthogonalisation de Gram–Schmidt, appliqué à l'ensemble des puissances successives de x. G_0 est une constante positive, telle que $(G_0, G_0) = 1$. g_1 est une fonction linéaire définie comme :

$$g_1 = x - (x, G_0) G_0.$$

g_1 est manifestement orthogonale à G_0. G_1 est défini comme g_1 normé :

$$G_1 = g_1 / \| g_1 \| .$$

Nous itérons ensuite ces relations ; g_2 est une fonction quadratique définie par :

$$g_2 = x^2 - (x^2, G_0) G_0 - (x^2, G_1) G_1,$$

$$G_2 = g_2 / \| g_2 \| .$$

En formant le produit scalaire de g_2 successivement par G_0 et G_1, nous vérifions que cette fonction est bien orthogonale à G_0 et G_1. Nous écrivons, à l'ordre n

$$g_n = x^n + a_{n,n-1} G_{n-1} + a_{n,n-2} G_{n-2} + \cdots + a_{n,0} G_0$$

et nous choisissons les constantes $a_{n,j}$ de telle façon que g_n soit orthogonal aux G_j pour $j = 0 \ldots n-1$. Ceci impose que

$$a_{n,j} = -(x^n, G_j).$$

Il reste à normaliser ce polynôme :

$$G_n = \frac{g_n}{\| g_n \|}.$$

Nous construisons ainsi la suite des polynômes G_k, à partir des nombres (x, G_0), (x^2, G_0), $\ldots$, $(x^k, G_l) \ldots$

7.2. Relation avec les polynômes habituels

Théorème – Tout polynôme de degré n est représentable comme combinaison linéaire de polynômes G_k, avec $k \leq n$.

Nous avons construit les G_k comme combinaisons linéaires des x^p, $p \leq k$; de plus, le coefficient de x^k dans G_k est non nul. Il en résulte que ces relations linéaires peuvent s'inverser et que x^p est fonction linéaire des G_k, $k \leq p$. Un polynôme quelconque Φ, lui-même combinaison linéaire des x^p, peut donc s'exprimer comme :

$$\Phi = a_0 G_0 + a_1 G_1 + \cdots + a_n G_n.$$

La détermination des a_i est facile : effectuons en effet le produit scalaire de l'égalité précédente par G_p ($p \leq n$). Il vient :

$$a_p = (\Phi, G_p), \quad p = 0, 1, 2, \ldots, n.$$

Nous avons au contraire $(\Phi, G_{n+1}) = 0$, quel que soit $\Phi(x)$ de degré n, puisque G_{n+1} est orthogonal à tous les G_k d'indice inférieur. Nous en déduisons le théorème suivant

Théorème – Le polynôme G_k est orthogonal à tout polynôme de degré inférieur.

Par ailleurs, $\Phi(x)$ sera identiquement nul si et seulement si : $(\Phi, G_p) = a_p = 0$; $p = 0, 1, \ldots, n$.

Remarque : Le produit scalaire (xG_{n-1}, G_n) est encore le produit d'un polynôme de degré n, xG_{n-1}, par G_n ; si b_n dénote le coefficient de G_n dans le développement de xG_{n-1}, le produit scalaire considéré se réduit à :

$$(xG_{n-1}, G_n) = b_n (G_n, G_n) \quad \neq 0.$$

7.3. Propriétés des zéros

Les zéros de G_n appartiennent à I et sont tous simples. Pour le prouver, nous supposons au contraire que G_n n'admet dans I que $p < n$ racines, toutes simples, et nous montrons que cette hypothèse conduit à une contradiction. Formons le polynôme Φ de degré $p < n$ qui admet ces mêmes racines. Le produit scalaire (G_n, Φ) doit être nul puisque $\deg(\Phi) < n$; or ce produit scalaire est en fait l'intégrale :

$$(G_n, \Phi) = \int w(x) \Phi(x) G_n(x) dx$$

qui ne peut être nulle, puisque l'intégrant garde un signe constant dans I. En effet, le produit ΦG_n admet les racines de G_n (ou de Φ) comme racines doubles et ne peut changer de signe lorsque x traverse l'une de ces racines ; de plus $w(x) > 0$ sur I. Il faut donc que $p = n$.

Prouvons maintenant que G_n ne peut avoir de racines multiples. Dans un raisonnement par l'absurde calqué sur le précédent, nous supposons, au contraire, que G_n

admet, dans I, $n-2$ racines simples et une racine double. Prenons alors pour Φ le polynôme de degré $n-2$ qui admet les mêmes racines simples que G_n. L'intégrale qui exprime le produit scalaire $(\Phi, G_n) \equiv 0$ ne peut s'annuler, parce que l'intégrant garde un signe constant dans I. On peut généraliser ce raisonnement en définissant Φ comme le polynôme de degré $< n$ qui admet comme racines tous les zéros d'ordre impair de G_n.

Les zéros des polynômes successifs adoptent une disposition particulière, résumée par le théorème ci-dessous.

Théorème – Les zéros de G_k séparent ceux de G_{k+1}.

Ce résultat est vrai pour G_0 et G_1 ; il se démontre par récurrence dans le cas général. On peut dire aussi que la suite des G_n forme une suite de Sturm.

7.4. Relation de récurrence

Nous allons vérifier l'existence d'une relation de récurrence, de la forme :

$$G_{n+1}(x) = (\alpha_n x + \beta_n) G_n(x) + \gamma_n G_{n-1}(x). \tag{7.5}$$

où les $\alpha_n, \beta_n, \gamma_n$ sont des constantes. Cette propriété est une conséquence de l'orthogonalité. Soit c_k le coefficient de x^k dans G_k (terme de plus haut degré) ; posons $a_k = c_{k+1}/c_k$. Considérons le polynôme

$$F \equiv G_{n+1} - a_n x G_n.$$

Il est de degré n au plus (à cause de la définition de a_n) et peut donc s'exprimer comme une combinaison linéaire des $G_i, i \leq n$:

$$F = \sum_0^n b_j G_j.$$

Formons le produit scalaire $(F, G_p) = (G_{n+1}, G_p) - a_n(xG_n, G_p) = -a_n(xG_n, G_p)$, à condition que $p < n+1$. De plus, l'examen de l'intégrale qui définit (xG_n, G_p) montre que : $(xG_n, G_p) = (G_n, xG_p)$. Comme xG_p est de degré au plus $p+1$, il est orthogonal à G_n tant que $p < n-1$. Nous venons de prouver que F était orthogonal à $G_p, 0 \leq p \leq n-2$. Le développement de F s'écrit alors :

$$F = G_{n+1} - a_n x G_n = b_n G_n + b_{n-1} G_{n-1}$$

si bien que

$$G_{n+1} = (a_n x + b_n) G_n + b_{n-1} G_{n-1}.$$

Cette relation est bien de la forme annoncée, à condition de choisir $\alpha_n = a_n, \beta_n = b_n$ et $\gamma_n = b_{n-1}$. En posant $G_{-1} = 0$, on définit complètement les G_n.

7.5. Équation différentielle

On peut montrer que chaque G_k obéit à une équation différentielle du second ordre. Il est plus facile de démontrer une réciproque partielle. Soit l'équation différentielle :

$$A(x)y'' + B(x)y' + C_n y = 0,$$

où C_n est une constante (avec $C_n \neq C_p$ si $n \neq p$), $A(x)$ et $B(x)$ deux fonctions régulières de x. Supposons que l'équation admette une solution polynômiale $y = G_n(x)$ de degré n pour chaque valeur de n.

On peut alors trouver un intervalle $[a, b]$ et une fonction de poids w tels que la suite des G_n soit orthogonale par rapport à ces éléments. En effet, G_n et G_p satisfont séparément à l'équation proposée :

$$AG_n'' + BG_n' + C_n G_n = 0,$$

$$AG_p'' + BG_p' + C_p G_p = 0.$$

Multiplions la première relation par wG_p, la seconde par wG_n et retranchons membre à membre ; il vient

$$Aw(G_n G_p'' - G_p G_n'') + Bw(G_n G_p' - G_p G_n') + w(C_p - C_n)G_n G_p = 0$$

équation qui s'écrit encore :

$$[Aw(G_n G_p' - G_p G_n')]' + [wB - (wA)'](G_n G_p' - G_p G_n') + w(C_p - C_n)G_n G_p = 0. \quad (7.6)$$

Considérons l'équation différentielle $uB - (uA)' = 0$ et notons w la solution qui obéit aux conditions aux limites $Aw|_a = Aw|_b = 0$. En intégrant terme à terme l'équation (7.6), avec cette définition de w, nous obtenons :

$$(C_p - C_n) \int_a^b w G_n G_p dx = 0,$$

ce qui signifie que les polynômes G_n sont orthogonaux sur l'intervalle I par rapport à la fonction de poids w.

7.6. Fonction génératrice

On peut en général trouver une fonction $g(u, x)$ telle que l'on puisse écrire :

$$g(u, x) = \sum_{k=0}^{\infty} C_k G_k(x) u^k \quad (7.7)$$

les C_k étant des constantes et u une variable réelle auxiliaire. Si la fonction g a une forme analytique « simple » par rapport à u et à x, on l'appelle la fonction génératrice de la suite des G_k. La fonction génératrice peut servir à définir les G_k ; les propriétés de ces polynômes se démontrent alors par des manipulations de g.

7.7. Formule de Rodrigues

Les G_n s'expriment en fonction de la dérivée d'ordre n d'une fonction $U_n(x)$ (formule de Rodrigues) :

$$G_n(x) = \frac{1}{w(x)} \frac{d^n}{dx^n} U_n(x) \tag{7.8}$$

laquelle est solution d'un problème différentiel :

$$\frac{d^{n+1}}{dx^{n+1}} \left[\frac{1}{w(x)} \frac{d^n U_n}{dx^n} \right] = 0$$

avec les conditions aux limites $U_n = U_n' = U_n'' = \ldots = U_n^{(n-1)} = 0$ en $x = a$ et $x = b$.

7.8. Identité de Darboux–Christofel

Une démonstration plutôt laborieuse permet d'établir l'identité de Darboux–Christofel :

$$\sum_{k=0}^{n} G_k(x) G_k(y) = \frac{1}{(x-y)a_n} [G_{n+1}(x) G_n(y) - G_n(x) G_{n+1}(y)] \tag{7.9}$$

souvent utilisée pour simplifier des expressions impliquant les G_n (la constante a_n a été définie au §7.4).

7.9. Polynômes particuliers

7.9.1. Legendre

Les polynômes de Legendre (déjà rencontrés à la section 2.2) sont orthogonaux sur $[-1, 1]$ par rapport à la fonction de poids $w = 1$. A partir de $P_0 = 1$, on peut orthogonaliser les puissances successives de x pour obtenir

$$P_0 = 1; \quad P_1(x) = x; \quad P_2(x) = (3x^2 - 1)/2; \quad P_3(x) = (5x^3 - 3x)/2,$$
$$P_4(x) = (35x^4 - 30x^2 + 3)/8; \quad P_5(x) = (63x^5 - 70x^3 + 15x)/8.$$

Le polynôme P_k a le degré et la parité de k. En intégrant par partie dans la relation d'orthogonalité $(P_k, P_l) = 0$, on démontre une relation d'Olinde Rodrigues :

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n$$

avec $P_n(1) = 1$. Avec ces définitions, les P_n ne sont pas normalisés à 1 mais à $(P_n, P_n) = 2/(2n + 1)$. Ils admettent la fonction génératrice :

$$g(ux) = \frac{1}{\sqrt{1 - 2ux + u^2}} = \sum_{k=0}^{\infty} P_k(x) u^k$$

d'où l'on tire la relation de récurrence (voir l'exercice 6) :

$$(n+1)P_{n+1} - (2n+1)xP_n + nP_{n-1} = 0.$$

Les polynômes de Legendre obéissent à l'équation différentielle :

$$(x^2-1)P_n'' + 2xP_n' - n(n+1)P_n = 0.$$

7.9.2. Hermite

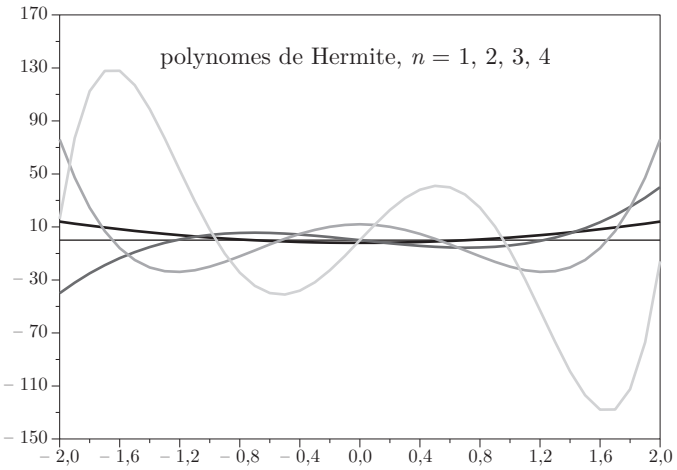


Figure 7.1 – Les premiers polynômes de Hermite.

Les polynômes de Hermite sont orthogonaux sur l'intervalle : $[-\infty, +\infty]$, par rapport à la fonction de poids : $w(x) = \exp(-x^2)$. Ils satisfont à la relation de récurrence :

$$H_{n+1} - 2xH_n + 2nH_{n-1} = 0$$

et à l'équation différentielle :

$$H_n'' - 2xH_n' + 2nH_n = 0.$$

Ils peuvent se déduire de la fonction génératrice :

$$g(u,x) = \exp(2ux - u^2) = \sum_k H_k(x)u^k/k!$$

On connaît aussi une formule de Rodrigues :

$$H_n(x) = (-1)^n \exp(x^2)\frac{d^n}{dx^n}\exp(-x^2).$$

Les polynômes d'Hermite admettent une représentation générale assez simple :

$$H_n(x) = \sum_0^{n/2}(-\frac{1}{2})^p \frac{1}{p!}\frac{1}{(n-2p)!}(2x)^{n-2p}$$

et les premiers représentants de l'espèce s'écrivent

$$H_0(x) = 1; \quad H_1(x) = 2x; \quad H_2(x) = 4x^2 - 2; \quad H_3(x) = 8x^3 - 12x;$$
$$H_4(x) = 16x^4 - 48x^2 + 12; \quad H_5(x) = 32x^5 - 160x^3 + 120x.$$

Vous pouvez constater que H_k a la parité de k.

7.9.3. Laguerre

On peut définir ces polynômes à partir d'une formule d'Olinde Rodrigues :

$$L_n(x) = \frac{1}{n!} e^x \frac{d^n}{dx^n} \left(e^{-x} x^n \right),$$

qui nous permet d'écrire les premiers membres de la suite :

$$L_0(x) = 1; \quad L_1(x) = -x + 1; \quad L_2(x) = \frac{1}{2}(x^2 - 4x + 2);$$
$$L_3(x) = \frac{1}{6}(-x^3 + 9x^2 - 18x + 6).$$

Ils admettent la fonction génératrice :

$$\frac{1}{1-t} \exp\left(-\frac{xt}{1-t} \right) = \sum_{k=0}^{\infty} L_k(x) t^k.$$

et la relation de récurrence :

$$(n+1)L_{n+1} + (x - 2n - 1)L_n + nL_{n-1} = 0.$$

Ils satisfont à l'équation différentielle :

$$xL_n'' + (1-x)L_n' + nL_n = 0.$$

Ces polynômes sont orthogonaux sur l'intervalle $[0, \infty]$ par rapport à la fonction de poids $w = e^{-x}$; ils sont normalisés à un : $(L_n, L_n) = 1$. Grâce au programme ci-dessous (listing 7.1), nous avons calculé et tracé les premiers polynômes de Laguerre (voir figure 7.2).

Listing 7.1 – Construction des polynômes de Laguerre

```
//polynomes de Laguerre                                          1
//(n+1)L_{n+1} + (x -2n -1)L_n + nL_{n-1} = 0                    2
function fn = L(n,x)                                             3
if n == 0, fn = 1, end                                           4
if n == 1, fn = 1 - x, end                                       5
if n > 1                                                         6
  k = 1; Lavder = 1; Lder = 1 - x;                               7
  while k < n                                                    8
    L = ((2*k + 1 - x).*Lder - k*Lavder)/(k+1);                  9
    Lavder = Lder;                                               10
```

```
    Lder = L;
    k = k+1;
  end
  fn = L;
end
endfunction
x = 0:0.1:6;
xset("window",0), xbasc(0)
plot2d(x,[L(2,x)',L(3,x)',L(4,x)',L(5,x)'])
xsegs([0,6],[0,0])
xset("font size", 3)
xstring( 2, 6, "polynomes de Laguerre, n = 1,2,3,4")
```

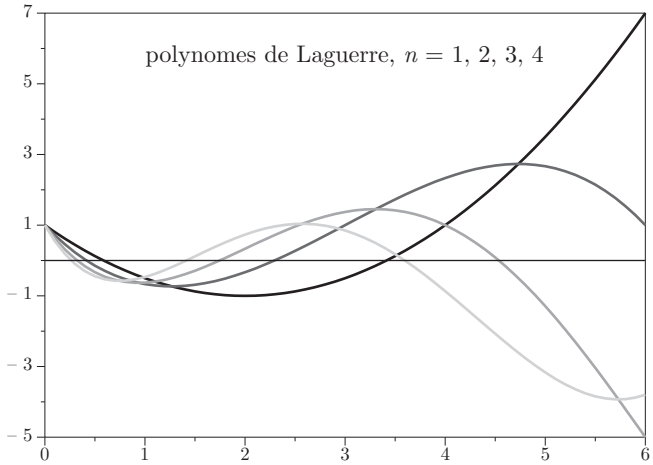


Figure 7.2 – Les premiers polynômes de Laguerre.

7.9.4. Tschebychef

Ces polynômes admettent la définition simple :

$$T_n(x) = \cos[n \arccos(x)].$$

On obtient une relation de récurrence entre T_{n-1}, T_n et T_{n+1} en calculant $\cos[(n\pm1)x]$:

$$T_{n+1}(x) + T_{n-1}(x) = 2xT_n(x).$$

Nous en déduisons les expressions des premiers polynômes

$$T_0 = 1; \quad T_1 = x; \quad T_2 = 2x^2 - 1; \quad T_3 = 4x^3 - 3x;$$
$$T_4 = 8x^4 - 8x^2 + 1.$$

Les T_n obéissent à l'équation différentielle

$$(1 - x^2)T_n'' - xT_n' + n^2T_n = 0.$$

Ils sont orthogonaux sur l'intervalle $[-1,1]$ par rapport à la fonction de poids

$$w = \frac{1}{\sqrt{1-x^2}}.$$

Leur fonction génératrice s'écrit

$$g(x,u) = \frac{1-ux}{1-2ux+u^2} = \sum_{k=0}^{\infty} u^k T_k(x)$$

7.10. Autres polynômes classiques

Voici les définitions de certaines suites moins fréquemment rencontrées.

7.10.1. Jacobi

$$I = [-1,1]; \quad w = (1-x)^\alpha (1-x)^\beta; \quad \alpha, \beta > -1.$$

Les polynômes de Jacobi recouvrent plusieurs cas particuliers intéressants : polynômes de Legendre, pour $\alpha = \beta = 0$, polynômes de Tschebychef de première espèce ($\alpha = \beta = -1/2$), de deuxième espèce ($\alpha = \beta = 1/2$), de troisième espèce ($\alpha = -\beta = -1/2$), de quatrième espèce ($\alpha = -\beta = 1/2$) et polynômes de Gegenbauer ($\alpha = \beta = \lambda - 1/2$).

7.10.2. Laguerre généralisé

$I = [0,\infty]$; $w = x^\alpha e^{-t}$; $\alpha > -1$. Les polynômes de Laguerre habituels sont obtenus en faisant $\alpha = 0$ dans la définition des polynômes généralisés.

7.11. Pour en savoir plus

– A. Angot : *Compléments de mathématiques à l'usage des ingénieurs* (Masson, Paris, 1972).

– M. Abramowitz, I.A. Stegun : *Handbook of mathematical functions*, ch. 22 (Dover, New York, 1972).

– É. Belorizky : *Outils mathématiques à l'usage des scientifiques et ingénieurs*, ch. 9 (EDP, Grenoble Sciences, 2007).

– Portrait de Olinde Rodrigues : `http://www.nebuleuse-rh.org/olinde`

7.12. Exercices

Exercice 1

Les polynômes de Tschebychef sont définis pour $-1 \leq x \leq 1$ par

$$T_n(x) = \cos(n \arccos\ x).$$

a) Former T_0, T_1, T_2 et T_3 comme fonctions de x ; il est cependant commode d'introduire la variable intermédiaire $\theta = \arccos x$; préciser le domaine de définition de cette variable.

b) Démontrer la relation de récurrence

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

Utiliser cette relation pour construire T_4 et T_5. Quelle est la parité de T_n ?

c) Déterminer les zéros de T_n. Montrer qu'entre deux zéros successifs de ce polynôme on rencontre un zéro de T_{n-1}.

d) Majorer T_n sur $[-1, 1]$.

e) Vérifier la relation

$$\int_{-1}^{1} \frac{T_k(x)T_\ell(x)}{\sqrt{1-x^2}} dx = 0 \quad si \quad k \neq \ell.$$

Que vaut cette intégrale quand $k = \ell$?

f) Exprimer x^0, x^1, x^2, x^3, x^4 et x^5 en fonction des polynômes T_0, T_1, T_2, T_3, T_4 et T_5.

g) On note $p_n(x)$ le polynôme de degré n qui représente le développement de Taylor tronqué à l'ordre n de e^x au voisinage de zéro et q_n le développement de e^x sur la base des T_i :

$$e^x \simeq \sum_{k=0}^{n} \frac{x^k}{k!} \equiv p_n(x)\ ; \quad e^x \simeq \sum_{k=0}^{n} a_k T_k(x) \equiv q_n(x).$$

Former $p_5(x)$ et $q_5(x)$.

h) Quelle erreur commet-on en remplaçant e^x par p_5 ou par q_5 sur le segment $[-1, 1]$?

i) Borner l'erreur commise, dans le même intervalle, lorsque l'on néglige les deux derniers termes de q_5 ; soit q_3^* ce nouveau polynôme. Exprimer q_3^* en fonction des seules puissances de x. Calculer $q_3^*(1)$ et vérifier que l'erreur en ce point est compatible avec les majorations précédentes.

j) Combien faut-il, au minimum, de multiplications pour calculer p_5 et q_3^* ?

Exercice 2

Déterminer, en utilisant le procédé d'orthogonalisation de Schmidt appliqué aux puissances successives de x, les quatre premiers polynômes P_k qui vérifient les relations :

$$P_0 = 2; \qquad \int_{-1}^{1} P_k(x) P_\ell(x) dx = \frac{2\delta_{k,l}}{2\ell + 1}$$

Les polynômes ainsi obtenus sont appelés polynômes de Legendre.

Exercice 3

Une charge électrique q est située au point A tel que $OA = a$. On cherche le potentiel électrique $V(\vec{r})$ en un point M, tel que $\overrightarrow{OM} = \overrightarrow{r}$ et $(\overrightarrow{OA}, \overrightarrow{OM}) = \theta$. Dans l'hypothèse où $a \ll r$, former le développement limité de $V(r)$, en fonction des puissances croissantes de a/r. Vérifier que les coefficients du développement sont des polynômes en $\cos\theta$ très proches de ceux de l'exercice précédent.

Exercice 4

a) Écrire l'équation de Laplace en coordonnées sphériques, dans le cas où le potentiel $V(r, \theta, \phi)$ est indépendant de ϕ. Chercher une solution de la forme $V(r, \theta) = F(r)G(\theta)$ et montrer que l'équation de Laplace est équivalente à deux équations différentielles ordinaires, l'une pour F, l'autre pour G.

b) Vérifier que l'équation en r admet la solution :

$$F(r) = Ar^n + B/r^{n+1}.$$

c) On fait, dans l'équation en θ, le changement de variable $x = \cos\theta$: que devient cette équation ? On admet que les solutions sont des polynômes $Q_n(x)$. Montrer que, si Q_n est une solution, Q_{-n-1} en est une autre. Déduire de ce qui précède la forme générale de la solution de l'équation de Laplace en symétrie axiale.

d) On sait que la fonction $U(r) = C/r$ est, hors de l'origine, solution de l'équation de Laplace. A quelle valeur de n correspond-elle ? On sait aussi que $\partial U/\partial z$ est une solution ; quel est le polynôme $Q(x)$ correspondant ? Comment étendre ce raisonnement pour former d'autres polynômes de la famille ? Quelle relation existe-t-il avec les polynômes des deux exercices précédents ?

Exercice 5

On considère les polynômes définis par la relation :

$$G_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}[(x^2 - 1)^n].$$

Former les premiers termes de la famille. On définit le produit scalaire de deux polynômes par la relation :

$$(G_k, G_\ell) \equiv I_{k,\ell} \equiv \int_{-1}^{1} G_k(x) G_\ell(x) dx.$$

Prouver, au moyen d'intégrations par partie, que (G_k, G_ℓ) est nul si $k \neq \ell$.

Exercice 6

On pose :

$$g(u,x) \equiv \frac{1}{\sqrt{u^2 - 2ux + 1}} \equiv \sum_0^\infty R_n(x) u^n.$$

Montrer que R_n est un polynôme de degré n en x. g est appelée la fonction génératrice des R_n. En dérivant par rapport à u les deux membres de cette égalité, trouver une relation de récurrence entre polynômes d'ordres successifs. En dérivant g par rapport à u, obtenir une relation entre R_n et les dérivées R'_{n-1} et R'_{n+1}. Combiner ces résultats pour former l'équation différentielle qui régit les R_n. Relier cette famille de polynômes à celles des questions précédentes.

This page intentionally left blank

Chapitre 8

Dérivation et intégration numériques

Dans ce chapitre, nous nous intéressons au calcul numérique de la dérivée en un point d'une fonction et au calcul (numérique) de l'intégrale d'une fonction sur un intervalle donné. Du point de vue du calculateur, ces deux opérations ne sont nullement équivalentes. On peut en général dériver une fonction exprimée par une formule. Écrire et exécuter le programme correspondant permet simplement de gagner du temps. Par contre, vous savez que l'on ne connaît de primitive que pour un petit nombre de fonctions. Pour l'immense majorité des cas, l'intégration (ou la quadrature, comme on dit parfois) n'est possible que de façon approchée, numérique. De plus, pour l'une ou l'autre tâche, il est bon de traiter séparément le cas d'une fonction définie analytiquement (par une formule) et le cas d'une fonction définie numériquement (par une table de valeurs). D'autre part, le calcul d'une intégrale indéfinie, considérée comme fonction de sa borne supérieure, est traité dans le chapitre 11, consacré à la résolution des problèmes différentiels.

Avant d'aborder les problèmes de dérivation ou d'intégration, qui forment le coeur de l'analyse mathématique, il nous parait utile de rappeler quelques théorèmes élémentaires ; c'est ce que nous faisons dans le prochain paragraphe.

8.1. Rappels d'analyse

Le résultat suivant est une généralisation du théorème de Rolle.

> **Théorème des accroissements finis** – Soit f une fonction continue sur l'intervalle fermé $[a, b]$ (avec $a < b$) et dérivable sur l'intervalle ouvert $]a, b[$; il existe au moins un point c de $]a, b[$ pour lequel
>
> $$f(b) - f(a) = f'(c)(b - a).$$

Géométriquement, cela signifie qu'il existe au moins un point de l'intervalle où la tangente à la courbe représentative de f est parallèle à la corde joignant les extrémités de la courbe.

Formules de la moyenne pour les intégrales – Si f est intégrable sur $[a,b] \equiv I$ et si l'on pose $m = \inf_{x\in I} f(x)$ et $M = \sup_{x\in I} f(x)$, alors

$$m(b-a) \leq \int_a^b f(t)dt \leq M(b-a).$$

Si, de plus, f est continue sur I, alors il existe un nombre réel c appartenant à l'intervalle ouvert $]a,b[$ tel que

$$\frac{1}{b-a}\int_a^b f(t)dt = f(c).$$

Ce résultat nous sera plus utile sous une forme généralisée (**deuxième formule de la moyenne**). Si f et g sont continues, $g \geq 0$ et intégrable sur I, alors

$$\int_a^b f(x)g(x)dx = f(c)\int_a^b g(x)dx$$

pour un nombre $c \in I$.

Nous en arrivons à la formule de Taylor, qui est à l'origine de bien des applications décrites dans ce chapitre et les suivants.

Formule de Taylor – Soit $f(x)$ une fonction possédant $n+1$ dérivées continues sur l'intervalle $[a,b]$ et soient x et x_0 deux points de cet intervalle. Alors

$$f(x) = p_n(x) + R_{n+1}(x),$$

avec :

$$p_n(x) = f(x_0) + \frac{x-x_0}{1!}f'(x_0) + \cdots + \frac{(x-x_0)^n}{n!}f^{(n)}(x_0),$$

$$R_{n+1}(x) = \frac{1}{n!}\int_{x_0}^{x}(x-t)^n f^{(n+1)}(t)dt$$

$$= \frac{(x-x_0)^{n+1}}{(n+1)!}f^{(n+1)}(\xi), \quad x_0 \leq \xi \leq x.$$

8.2. Dérivée d'une fonction analytique

8.2.1. Développements limités

Considérons une fonction f définie par une formule ou un algorithme, dérivable dans un intervalle. Cette fonction est trop compliquée pour que le calcul analytique de sa dérivée soit pratique, mais nous souhaitons pourtant connaître la valeur numérique

de $f'(x)$. Le procédé le plus simple s'inspire de la définition même de la dérivée. Soit $f(x)$ la fonction à dériver, nous savons que :

$$f'(x) \cong \frac{f(x+h) - f(x)}{h}.$$

Comment faut-il choisir h ? Ce paramètre doit être plus grand que le plus petit réel représentable (environ 10^{-38} en simple précision), sinon le calcul s'arrête et la machine affiche « Division par Zéro ! ». h doit aussi être assez grand pour que $f(x)$ et $f(x+h)$ soient reconnus comme différents et être à l'abri des erreurs d'arrondi. Il faut maintenant éviter de tomber de Charybde en Scylla, et garder h assez petit pour que les erreurs de troncation soient négligeables. Ces dernières sont faciles à estimer. Nous pouvons écrire une version exacte de la formule précédente, d'après le théorème de Taylor :

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

Vous voyez que l'erreur sur la dérivée est $O(h)$, ou encore que la formule est du premier ordre : elle est exacte pour une fonction linéaire, dont la dérivée seconde est nulle.

En faisant un petit effort d'imagination, nous pouvons construire une formule plus précise, mais qui ne nous coûtera aucun travail supplémentaire (même nombre d'évaluations de la fonction f). Appliquons deux fois le théorème de Taylor, pour les points $x-h$ et $x+h$:

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(\xi_+), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{6} f'''(\xi_-), \\ & x \leq \xi_+ \leq x+h; \quad x-h \leq \xi_- \leq x. \end{aligned} \tag{8.1}$$

Grâce à la parité des termes en h^2, ceux-ci disparaissent de la différence terme à terme des deux lignes précédentes :

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(\xi) \text{ où } x-h \leq \xi \leq x+h. \tag{8.2}$$

L'expression du terme d'erreur découle de l'application du théorème des valeurs intermédiaires. Les erreurs d'arrondi sont les mêmes que précédemment, mais l'erreur de troncation est maintenant en h^2 ; la formule est exacte pour les polynômes du second degré.

Exemple – On a extrait d'une table les valeurs suivantes :

x	$\ln x$
$0,159$	$-1,83885$
$0,160$	$-1,83258$
$0,161$	$-1,82635$

et on demande de calculer la dérivée de $\ln x$ en 0,160. Nous avons trouvé

$$\frac{f(0,161)-f(0,160)}{0.001}=6,23 \quad ; \quad \frac{f(0,161)-f(0,159)}{0,002}=6,25.$$

Quelle est la valeur exacte ?

Comment approcher une dérivée seconde ? Il suffit de reprendre la formule (8.1), de pousser les développements à l'ordre 4 et d'ajouter terme à terme (au lieu de retrancher) ; il vient

$$f(x+h)+f(x-h)=2f(x)+h^2f''(x)+\frac{h^4}{24}\left[f^{(4)}(\xi_+)+f^{(4)}(\xi_-)\right]$$

soit, en isolant f'',

$$f''(x)=\frac{1}{h^2}\left[f(x+h)-2f(x)+f(x-h)\right]-\frac{h^2}{12}f^{(4)}(\xi). \tag{8.3}$$

Ici ce sont les puissances impaires de h qui ont disparu, ce qui nous permet d'obtenir une erreur en $O(h^2)$ sans effort.

Remarque : La formule précédente suppose la continuité de la dérivée quatrième et la condition $x-h\leq\xi\leq x+h$.

Exemple – La dérivée seconde de $\ln x$, au point 0,160, vaut, d'après les données précédentes, -40.

8.2.2. Méthode des coefficients indéterminés

La méthode des coefficients indéterminés permet de retrouver très facilement les formules d'approximation des dérivées, sans toutefois donner d'indications sur le terme d'erreur. Nous cherchons trois nombres a_-, a_0 et a_+ tels que l'expression suivante soit « aussi vraie que possible » :

$$f'(x)=a_+f(x+h)+a_0f(x)+a_-f(x-h).$$

Nous allons imposer trois conditions : la relation devra être vérifiée, quel que soit x, pour $f=x^0,x^1$, et x^2 (les trois premières puissances de x). Ceci s'écrit :

$$\begin{aligned} 0 &= a_+ + a_0 + a_-, \\ 1 &= a_+(x+h) + a_0x + a_-(x-h), \\ 2x &= a_+(x+h)^2 + a_0x^2 + a_-(x-h)^2. \end{aligned}$$

Ce système de trois équations linéaires à trois inconnues se résout facilement. Compte tenu de la première équation, la deuxième s'écrit :

$$1/h=a_+-a_-.$$

Tenant compte des deux premières, la dernière équation devient :

$$0=a_++a_-.$$

D'où nous tirons facilement

$$a_0 = 0, \quad a_+ = -a_- = 1/2h.$$

L'heureuse disparition de certains termes dans les équations précédentes ne doit rien au hasard. Ces relations doivent être vérifiées quel que soit x : il faut donc que x ne figure pas dans le système qui détermine les coefficients a. Nous aurions donc pu partir d'une valeur particulièrement commode de x, comme $x = 0$. Avec l'habitude, on peut gagner encore un peu de temps en posant $h = 1$ et en restaurant h à la fin : f' a les dimensions de f/h.

Vous pourrez établir d'autres formules, en nombre limité uniquement par votre patience. Certaines impliqueront un grand nombre de valeurs de f, pour une erreur de troncation moindre, d'autres pourront être dissymétriques, pour s'approcher sans danger d'une discontinuité : on peut, par exemple, calculer f'_0 en fonction de f_0, f_1 et f_2. Elles s'obtiennent par la méthode des coefficients indéterminés, par le théorème de Taylor, ou à partir des différences latérales (voir plus loin).

8.2.3. Dérivée du polynôme d'interpolation

Il existe une autre méthode systématique pour construire des formules d'approximation de f' : il suffit de dériver un polynôme d'interpolation de f. Rappelons la formule de Newton impliquant les différences latérales :

$$p(x_0 + hm) = f_0 + m\Delta f_0 + \frac{m(m-1)}{2}\Delta^2 f_0 + \cdots$$

où $\Delta f_0 = f_1 - f_0$, $\Delta^2 f_0 = \Delta f_1 - \Delta f_0 = f_2 - 2f_1 + f_0$, $\Delta^k f_0 = \Delta^{k-1} f_1 - \Delta^{k-1} f_0$ et dont la dérivée (par rapport à $x = x_0 + hm$) est :

$$p'(x_0 + hm) = \frac{1}{h}\left[\Delta f_0 + \frac{1}{2}(2m-1)\Delta^2 f_0 + \cdots\right].$$

Au point x_0 ($m = 0$), nous trouvons :

$$p'_0 = \frac{1}{h}\left[\Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \cdots\right].$$

On peut obtenir les dérivées d'ordre supérieur par cette méthode et pas mal d'algèbre.

8.2.4. Accélération de la convergence

L'extrapolation de Richardson permet d'améliorer la précision d'une approximation en n'utilisant qu'une connaissance qualitative du terme d'erreur. Commençons par estimer la dérivée de f avec un pas h :

$$f'(x) \cong \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6}f'''(\xi_1) \equiv f'_1 + C_1 h^2,$$

puis avec un pas $2h$:

$$f'(x) \cong \frac{f(x+2h) - f(x-2h)}{4h} - 2\frac{h^2}{3} f'''(\xi_2) \equiv f'_2 + 4C_2 h^2.$$

À une certaine approximation, nous pouvons considérer que $C_1 = C_2$, si bien qu'en ajoutant 4 fois la première équation à l'opposée de la seconde, nous obtenons :

$$f' \cong (4f'_1 - f'_2)/3 = f'_1 + (f'_1 - f'_2)/3 \tag{8.4}$$

apparemment sans terme d'erreur. En réalité, l'erreur de troncation n'a pas disparu, mais elle est maintenant d'un ordre plus élevé (lequel ?). L'extrapolation de Richardson peut s'appliquer chaque fois que l'on utilise une approximation dépendant d'un paramètre lequel tend de façon continue vers zéro et à condition de connaître l'expression du terme principal de l'erreur.

8.3. Dérivée d'une fonction empirique

Nous pouvons avoir à calculer la dérivée d'une fonction définie par une table de valeurs, soit qu'elle ait été calculée par quelqu'un d'autre, soit qu'elle résulte de mesures expérimentales. Les méthodes précédentes se révèlent être malcommodes dans ce cas. D'une part, le pas h est maintenant imposé et n'est pas forcément adapté au but poursuivi. De plus, les entrées dans la table ne sont pas forcément équidistantes. D'autre part (et ici nous pensons surtout à des résultats expérimentaux), les inévitables erreurs de mesure vont être amplifiées par la division par h (qui est « petit »). Considérez en effet deux entrées successives prises dans une table de résultats expérimentaux :

variable indépendante	variable dépendante
x_k	$y_k + b_k$
x_{k+1}	$y_{k+1} + b_{k+1}$

Nous avons supposé que les valeurs de x étaient connues sans erreur, alors que les valeurs de y étaient entachées d'une erreur aléatoire (un bruit) b. Une valeur approchée de la dérivée est alors :

$$y' \cong \frac{y_{k+1} - y_k}{x_{k+1} - x_k} + \frac{b_{k+1} - b_k}{x_{k+1} - x_k}.$$

La première fraction est la dérivée cherchée ; sa valeur dépend peu de la différence $x_{k+1} - x_k$ (tant que l'on respecte les contraintes exposées au début de ce chapitre). La deuxième fraction est une fluctuation aléatoire d'autant plus grande que l'intervalle tabulaire h est petit.

Une première manière d'éviter ces difficultés consiste à construire un polynôme d'interpolation s'appuyant sur les points $x_{-m}, x_{-m+1}, \ldots, x_0, \ldots, x_m$. En supposant que

c'est f_0' qui nous intéresse, nous n'avons plus qu'à dériver ce polynôme. Le polynôme de Lagrange ou la fonction spline, avec 2 à 4 pivots, donnent de bons résultats. Une autre méthode consiste à construire un polynôme d'approximation au sens des moindres carrés, puis à dériver cette expression. Pour des abscisses équidistantes, le polynôme et ses dérivées ont des expressions simples (on parle souvent de polynômes de Golay). C'est la méthode que nous recommandons.

8.4. Généralités sur l'intégration numérique

Nous nous tournons maintenant vers le calcul numérique d'intégrales (on dit aussi quadrature numérique). Il importe de distinguer le cas des intégrales définies :

$$I = \int_a^b f(x)dx$$

où nous cherchons un nombre I connaissant un intervalle $[a, b]$ et la fonction f, des intégrales indéfinies (ou primitives) :

$$J(x) = \int_a^x f(u)du$$

où nous calculons en fait une suite de nombres, représentant la fonction $J(x)$. Ce dernier cas est généralement abordé comme un problème différentiel :

$$J' = f(x); \qquad J(a) = 0.$$

et ceux-ci seront traités au chapitre 11.

Nous allons donc chercher à approcher numériquement une intégrale définie. Toutes les méthodes que nous envisagerons peuvent s'écrire sous la forme :

$$I = \int_a^b f(x)dx = \sum_1^n w_j f(x_j) + E$$

où E est l'erreur de troncation, les x_j des abscisses (pivots ou noeuds) et les w_j des « poids ». Nous décrirons deux sortes d'algorithmes. Pour une première classe (Newton–Cotes), les pivots sont équidistants et les poids sont les seuls paramètres ajustables ; comme il y en a n, nous pourrons satisfaire n contraintes et rendre la méthode exacte pour un polynôme de degré $n-1$. Dans une deuxième classe (Gauss), les pivots et les poids seront considérés comme ajustables ; le choix de ces $2n$ paramètres nous permettra de rendre la méthode exacte pour un polynôme de degré $2n-1$. Évidemment, les méthodes de Gauss sont plus performantes que celles de Newton–Cotes, puisqu'elles permettent une précision plus grande pour un même temps de calcul, au pris d'une théorie plus compliquée et d'une programmation un peu plus lourde.

8.5. Méthodes élémentaires d'intégration

La figure 8.1 présente quatre méthodes d'intégration simples.

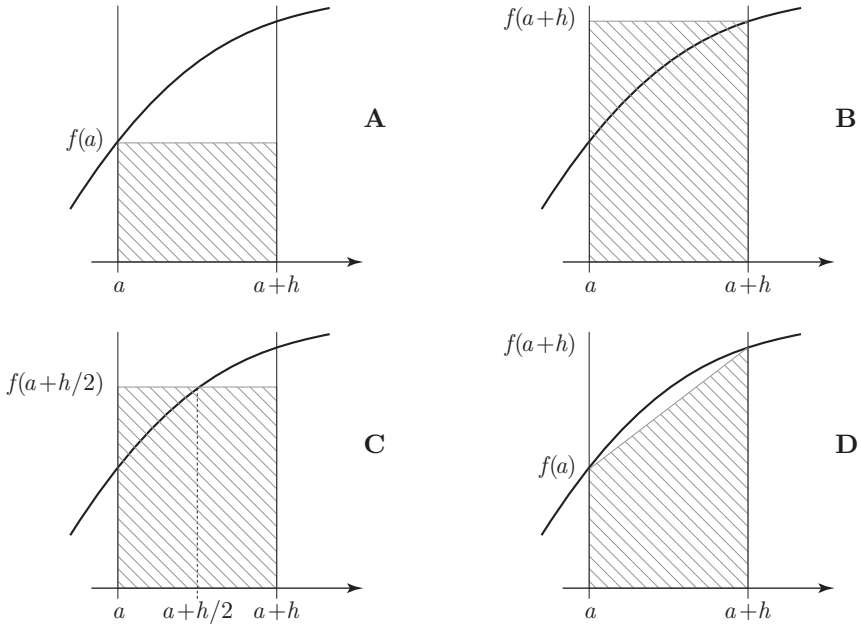


Figure 8.1 – Principe de quatre méthodes simples de quadrature numérique. A : point gauche ou rectangle à droite, B : point droit ou rectangle à gauche, C : « point milieu », D : trapèzes.

(A) (resp. (B)) représente ce que l'on peut appeler la méthode du point gauche (resp. point droit). L'intégrale (l'aire algébrique comprise entre le graphe de f, l'axe Ox et les deux verticales d'abscisses a et $a+h$) est approchée par l'aire du rectangle de même base et de hauteur $f(a)$ (resp. $f(a+h)$)

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

Le théorème de Taylor va nous permettre de borner l'erreur, par exemple dans le cas du rectangle à droite. Il s'écrit

$$f(x) = f(a) + (x-a)f'(\xi)$$

où ξ est un point (fonction de x) de l'intervalle $[a, x]$. Intégrons cette expression entre a et $a+h$.

$$\int_a^{a+h} f(x)dx = \int_a^{a+h} f(a)dx + \int_a^{a+h} (x-a)f'(\xi)dx.$$

Dans l'intervalle d'intégration, $x - a$ est toujours positif; en supposant que f' est continue dans ce même intervalle, nous pouvons appliquer la formule de la moyenne

$$\int_a^{a+h} f(x)dx = hf(a) + f'(\eta) \int_a^{a+h} (x-a)dx$$

où $a \leq \eta \leq a + h$. L'intégrale se calcule à vue après avoir fait le changement de variable $u = x - a$; nous trouvons

$$\int_a^{a+h} f(x)dx = hf(a) + \frac{h^2}{2} f'(\eta). \tag{8.5}$$

Dans le cas de la figure, $f' > 0$ et la valeur exacte de l'intégrale est en effet plus grande que sa valeur approchée. Appelant M_1 une borne supérieure de $|f'|$ dans l'intervalle $[a, a + h]$, nous pouvons encore écrire

$$\left| \int_a^{a+h} f(x)dx - hf(a) \right| \leq \frac{h^2}{2} M_1.$$

Vous pourrez établir une formule analogue dans le cas du point droit. Ces deux méthodes ne sont pas utilisées en pratique, car l'erreur est bien plus grande que celle associée à l'algorithme suivant qui, pourtant, implique un même nombre de calculs de f.

Le diagramme 8.1(C) illustre ce que nous appellerons la méthode du « point milieu », par analogie avec le terme anglais. Nous espérons que l'erreur sera plus petite, car elle est la somme des aires de deux triangles curvilignes qui sont de signes opposés et se compensent approximativement. La formule correspondante s'écrit

$$I \simeq hf\left(a + \frac{h}{2}\right).$$

Cherchons l'expression de l'erreur, en supposant $f(x)$ deux fois continûment dérivable dans l'intervalle d'intégration. Nous formons le développement de Taylor de f autour de l'abscisse $a + h/2$

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

Nous intégrons terme à terme entre a et $a + h$

$$I = hf\left(a + \frac{h}{2}\right) + f'\left(a + \frac{h}{2}\right) \int_a^{a+h} \left(x - a - \frac{h}{2}\right) dx + \frac{1}{2} \int_a^{a+h} \left(x - a - \frac{h}{2}\right)^2 f''(\xi)dx.$$

Dans la deuxième intégrale, le coefficient de f'' est toujours positif et nous appliquons encore le théorème de la moyenne pour obtenir

$$I = hf\left(a + \frac{h}{2}\right) + f'\left(a + \frac{h}{2}\right) \int_a^{a+h} \left(x - a - \frac{h}{2}\right) dx + \frac{1}{2} f''(\eta) \int_a^{a+h} \left(x - a - \frac{h}{2}\right)^2 dx.$$

Les deux intégrales se calculent aisément en posant $u = x - a - \frac{h}{2}$; la première est nulle, la seconde vaut $h^3/12$. Nous obtenons finalement

$$\int_a^{a+h} f(x)dx = hf\left(a + \frac{h}{2}\right) + f''(\eta)\frac{h^3}{24}. \tag{8.6}$$

Introduisant M_2, une borne supérieure de $|f''|$ sur $[a, a+h]$, nous écrivons aussi bien

$$\left| \int_a^{a+h} f(x)dx - hf\left(a + \frac{h}{2}\right) \right| \leq M_2 \frac{h^3}{24}.$$

Vous voyez que la compensation entre triangles curvilignes d'aires algébriques opposées est d'autant plus mauvaise que la courbure du graphe (la valeur de f'') est plus grande. L'erreur est ici en h^3, grâce à la symétrie qui a fait disparaître le terme en h^2.

La figure 8.1(D) représente un algorithme à peine plus compliqué. La surface cherchée est approchée par l'aire du trapèze inscrit, soit

$$\int_a^b f(x)dx \simeq \frac{1}{2}(b-a)\left[f(a) + f(b)\right], \tag{8.7}$$

une expression dont le terme d'erreur est comparable au précédent et que nous détaillons dans la section suivante.

8.6. Méthodes de Newton–Cotes

Comme nous ne savons pas intégrer analytiquement la fonction f, nous la remplaçons par une expression plus simple ; dans le paragraphe précédent, il s'agissait de constantes. On peut penser que remplacer f par un polynôme d'interpolation (dont l'intégration est banale) conduira à des formules plus précises, puisque l'interpolant peut en principe être aussi proche que l'on veut de f. Appelons h l'intervalle entre pivots. Les diverses méthodes de ce groupe se distinguent par le choix des pivots dans l'intervalle d'intégration, supposé fini.

8.6.1. Intervalle fermé

Nous voulons calculer l'intégrale de f sur le segment $[a, b]$, divisé en n intervalles de taille h ; posons donc

$$h = (b-a)/n, \quad x_0 = a, \quad x_n = b, \quad x_j = a + jh.$$

Le polynôme d'interpolation de Lagrange construit sur les pivots $x_j, 0 \leq j \leq n$, s'écrit :

$$p(x) = \sum_0^n \ell_j(x) f(x_j)$$

où $\ell_j(x)$ représente un polynôme élémentaire de Lagrange. L'intégrale de p est une approximation de I :

$$I = \int_a^b f(x)dx = \sum_0^n w_j f_j + E$$

avec les définitions $f_j \equiv f(x_j)$, $w_j \equiv \int_a^b \ell_j(x)dx$. Les « nombres de Cotes » w_j ont été calculés une fois pour toutes pour $n \leq 10$. Les formules de Newton–Cotes les plus courantes sont résumées ci-dessous, avec la notation : $w_j \equiv hAW_j$. E est l'erreur de la méthode ; dans chaque cas, l'argument de la dérivée de f est un nombre ξ appartenant à l'intervalle d'intégration.

n	A	W_0	W_1	W_2	W_3	W_4	E	
1	1/2	1	1				$-\frac{h^3}{12}f''$	trapèzes
2	1/3	1	4	1			$-\frac{h^5}{90}f^{(4)}$	Simpson I
3	3/8	1	3	3	1		$-\frac{3h^5}{80}f^{(4)}$	Simpson II
4	2/45	7	32	12	32	7	$-\frac{8h^7}{945}f^{(6)}$	Villarceau

Dans certains pays, la formule de Villarceau est connue sous le nom de formule de Boole. En pratique, les deux dernières méthodes sont peu utilisées.

Toutes ces formules se démontrent facilement par la méthode des coefficients indéterminés. Prenons l'exemple de la première formule de Simpson et cherchons trois nombres w_0, w_1 et w_2 tels que :

$$\int_a^b f(x)dx = w_0 f(a) + w_1 f(a+h) + w_2 f(a+2h)$$

où $b = a + 2h$. Cette relation doit être exacte pour $f = x^0, x^1$ et x^2, ce qui conduit au système :

$$\left\{ \begin{array}{rcl} 2h & = & w_0 + w_1 + w_2, \\ 2ah + 2h^2 & = & aw_0 + (a+h)w_1 + (a+2h)w_2, \\ (1/3)[6a^2h + 12ah^2 + 8h^3] & = & w_0a^2 + w_1(a+h)^2 + w_2(a+2h)^2. \end{array} \right.$$

En substituant la première équation dans la seconde, puis les deux premières dans la troisième, nous trouvons :

$$\left\{ \begin{array}{l} 2h = w_0 + w_1 + w_2, \\ 2h = w_1 + 2w_2, \\ (8/3)h = w_1 + 4w_2. \end{array} \right.$$

système dont la solution est : $w_0 = w_2 = h/3, w_1 = 4h/3$. Ici encore, nous aurions pu poser $a = 0$ dès le début, puisque les coefficients doivent être indépendants de l'abscisse initiale.

Le calcul de l'erreur dans le cas général est assez long et ne sera pas abordé ici ; nous nous limiterons au cas $n = 1$. Nous utilisons pour cela le polynôme de Lagrange d'ordre 1, avec son terme d'erreur

$$f(x) - \frac{(b-x)f(a) + (x-a)f(b)}{b-a} = \frac{(x-a)(x-b)}{2!} f''(\xi).$$

L'intégration terme à terme nous donne

$$E \equiv \int_a^b f(x)dx - \frac{b-a}{2}\left[f(a) + f(b)\right] = \int_a^b \frac{(x-a)(x-b)}{2!} f''(\xi).dx$$

Le théorème de la moyenne nous permet d'écrire

$$E = \frac{1}{2} f''(\eta) \frac{-1}{6} (b-a)^3.$$

En examinant la figure (8.5-D), vous vérifierez que $E > 0$ et $f'' < 0$, ce que confirme la formule précédente. Le fait que les algorithmes du point milieu et du trapèze aient des erreurs de signes opposés sera utilisé lors de la résolution d'équations différentielles.

8.6.2. Intervalle ouvert

Il peut arriver que la fonction à intégrer ne soit pas définie pour l'une des bornes de l'intervalle d'intégration (ou pour les deux) ou qu'elle y présente une singularité quelconque. Il est alors souhaitable de ne pas s'approcher « trop » de cette limite, ce qui est impossible avec le découpage de l'intervalle d'intégration choisi plus haut ($x_0 = a, x_n = b$). Il existe des formules de quadrature dites « ouvertes » (parce qu'elles impliquent des intervalles ouverts) qui n'utilisent pas les valeurs de la fonction aux bornes. Elles s'écrivent en général :

$$I = \int_a^b f(x)dx = \sum_1^{n-1} w_j f(x_j) + E$$

où les bornes x_0 et x_n **ne figurent pas**. La formule « ouverte » la plus courante a été détaillée au paragraphe précédent, c'est la formule du « point milieu ». Elle est définie, avec les notations de cette section, par $n = 2, A = 2$ et $W_1 = 1$, soit encore :

$$I = \int_a^{a+2h} f(x)dx = 2hf(a+h) + \frac{h^3}{3} f''(\xi). \tag{8.8}$$

8.6.3. Formules composites

Que faire si la précision d'une intégrale numérique paraît insuffisante ? Une première possibilité consiste à choisir une formule d'intégration d'ordre plus élevé. Comme dans le cas de l'interpolation polynômiale, cette solution est à rejeter, et pour les mêmes raisons : lorsque l'ordre n devient grand (quand le degré du polynôme que l'on intègre en lieu et place de la fonction devient grand), le terme d'erreur peut avoir un comportement anarchique en fonction de ξ. La bonne méthode est semblable à celle qui motive l'interpolation spline : découper l'intervalle d'intégration en plusieurs « sous-intervalles » et utiliser une méthode d'ordre peu élevé dans chaque sous-intervalle. Comme la borne droite du sous-intervalle de rang p coïncide, dans le cas d'intervalles fermés, avec la limite gauche du sous-intervalle de rang $p+1$, nous économiserons un certain nombre d'évaluations de la fonction.

Divisons l'intervalle global $[a, b]$ en m sous-intervalles. Il vient, d'après le théorème de Chasles

$$I \equiv \int_a^b f(x)dx = \sum_{i=0}^{m-1} \int_{x_i}^{x_{i+1}} f(x)dx.$$

Appliquons la méthode de Newton–Cotes « fermée » d'ordre un dans chaque sous-intervalle

$$I = (h/2)[f_0 + f_1 + f_1 + f_2 + \ldots + f_{m-2} + f_{m-1} + f_{m-1} + f_m] + \sum_{i=0}^{m-1} \frac{-h^3}{12} f''(\xi_i).$$

Nous aboutissons ainsi à la formule composite (ou composée ou encore étendue) de Newton–Cotes d'ordre 1 (souvent appelée encore méthode des trapèzes) :

$$\int_a^b f(x) = h[f_0/2 + f_1 + f_2 + \ldots + f_{m-2} + f_{m-1} + f_m/2] + E \tag{8.9}$$

où tous les termes « intérieurs » ont doublé. L'application du théorème des valeurs intermédiaires montre que

$$E = -\frac{h^2}{12}(b-a)f''(\eta), \quad \eta \in [a, b]. \tag{8.10}$$

Dans la pratique, on programme ce calcul de façon itérative. Supposez que nous connaissions $I(m)$, calculée avec m sous-intervalles, mais que la précision nous paraisse encore insuffisante. Nous calculons alors $I(2m)$ à partir de $I(m)$ et des seules nouvelles valeurs de f, aux points intermédiaires d'abscisses $(i + 1/2)h$. Nous recommençons tant que le critère de convergence n'est pas atteint. Quel critère d'arrêt allons-nous choisir ? Ce peut être la condition que la variation relative de l'intégrale d'une étape à l'autre est inférieure à un seuil choisi à l'avance : $|[I(m+1) - I(m)]/I(m)| \leq \varepsilon$.

La méthode du point milieu se généralise tout aussi facilement en une méthode composite. Un raisonnement calqué sur le précédent montre que

$$\int_a^b f(x)dx = h \sum_{i=0}^{m-1} f(x_i) + \frac{h^2(b-a)}{24} f''(\eta), \quad \eta \in [a, b]. \tag{8.11}$$

Comme précédemment, on peut programmer ce calcul de façon itérative.Cependant, doubler le nombre d'intervalles n'apporte aucune économie, puisque ceux-ci ne comportent aucun pivot commun. Nous vous suggérons de vérifier qu'il faut tripler le nombre d'intervalles à chaque itération pour pouvoir économiser des calculs.

La formule de Simpson donne aussi naissance à une formule composée qui s'écrit, pour $n \geq 2$ et pair ($n/2$ sous-intervalles)

$$\int_a^b f(x) = \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 2f_{n-2} + 4f_{n-1} + f_n] + E \qquad (8.12)$$

avec

$$E = -\frac{h^4(b-a)}{180} f^{(4)}(\eta), \quad \eta \in [a, b].$$

Il est aussi possible de combiner une formule de Newton–Cotes ouverte avec $m-1$ formules fermées : ceci pourrait servir à éviter une région « dangereuse » autour de $x = a$.

Comme les termes d'erreur de toutes les méthode de Newton–Cotes sont connus, il est tentant d'améliorer la précision d'une quadrature, sans trop d'effort, par l'extrapolation de Richardson. En réalité, cela se fait rarement, car on connaît un algorithme analogue mais plus puissant : c'est la méthode de Romberg que nous décrivons dans le paragraphe suivant.

Exemple – On demande de calculer l'intégrale de $1/x$ de 1 à 3 (dont la valeur exacte est $\ln 3 = 1,0986$) avec deux puis quatre sous-intervalles, par la méthode des trapèzes. Nous trouvons :

$$m = 2 : \quad I_2 \cong (1)\left[\frac{1}{2}(1) + 1/2 + \frac{1}{2}(1/3)\right] = 7/6 = 1,166666\ldots;$$

$$m = 4 : \quad I_4 \cong (1/2)\left[\frac{1}{2}(1) + 2/3 + 1/2 + 2/5 + \frac{1}{2}(1/3)\right] = 67/60 = 1,116666\ldots$$

La formule de Richardson donne : $I \cong I_4 + (1/3)(I_4 - I_2) = 11/10 = 1,1$, soit une erreur relative un peu supérieure à $1/1000$.

8.7. Méthode de Romberg

Nous utilisons la notation $I_{k,0}$ = valeur approchée de I, calculée par la méthode des trapèzes avec $m = 2^k$ sous-intervalles. En utilisant une expression du terme d'erreur plus précise que celle mentionnée (et non démontrée) au paragraphe précédent, nous pourrions écrire

$$I_{k,0} = I - \sum_{j=1}^{\infty} \alpha_j \left[\frac{b-a}{2^k}\right]^{2j}.$$

Si nous répétons le même calcul avec des intervalles deux fois plus nombreux et deux fois plus petits, nous avons

$$I_{k+1,0} = I - \sum_{j=1}^{\infty} \alpha_j \left[\frac{b-a}{2^{k+1}}\right]^{2j}.$$

Comme pour une extrapolation de Richardson, formons :

$$I_{k,1} = 4I_{k+1,0} - I_{k,0} = 3I - \sum_{j=2}^{\infty} \alpha_j \left[\frac{b-a}{2^k}\right]^{2j} \left[-1 + 4/2^{2j}\right].$$

Nous venons de gagner un ordre dans le développement de l'erreur en fonction des puissances de $(b-a)$. Le procédé se généralise aisément. Nous construisons un tableau triangulaire dont la première colonne est formée des $I_{k,0}$, la deuxième des $I_{k,1}$ et dont le terme général s'écrit :

$$I_{k+1,m+1} = \frac{4^m I_{k+1,m} - I_{k,m}}{4^m - 1}.$$

Le gros du travail est effectué lors du calcul des $I_{k,0}$, puisque c'est à ce moment-là que nous calculons la fonction compliquée f ; la suite n'est qu'une série de combinaisons linéaires. Si $k \leq K$, alors le dernier élément de la diagonale principale, $I_{K,K}$, est la meilleure approximation de I que l'on puisse obtenir en évaluant 2^K fois f.

Nous avons écrit le programme présenté ci-dessous pour mettre en oeuvre l'algorithme de Romberg. On commence par remplir la première colonne d'un tableau `J(l,c)` par des valeurs calculées selon la méthode des trapèzes. Vous pouvez constater qu'à chaque étape on ne calcule que les valeurs strictement nécessaires de la fonction. Nous garnissons ensuite les autres colonnes selon la formule de Romberg.

Listing 8.1 – Calcul d'une intégrale par la méthode de Romberg

```
function y = fn(x)
y = exp(x).*cos(x)
endfunction
lmax = 5;
a = 0; b = %pi;
h = b - a;
J(1,1) = 0.5*h*(fn(a) + fn(b));
for l = 2:lmax
  h = h/2;
  x = a+h:2*h:b-h;
  di = h*sum(fn(x))
  J(l,1) = 0.5*J(l-1) + di;
end
for c = 2:lmax
  for l = c:lmax
    J(l,c) = ((4^(c-1))*J(l,c-1)-J(l-1,c-1))/(4^(c-1)-1);
  end
end
J
Jex = -(exp(%pi)+1)/2
```

Ce programme a servi à calculer l'intégrale $\int_0^\pi e^x \cos x dx$ dont la valeur exacte est $-(e^\pi + 1)/2 = -12,070346$; voici les résultats obtenus :

$-34,778519$	0	0	0	0
$-17,389259$	$-11,59284$	0	0	0
$-13,336023$	$-11,984944$	$-12,011084$	0	0
$-12,382162$	$-12,064209$	$-12,069493$	$-12,07042$	0
$-12,148004$	$-12,069951$	$-12,070334$	$-12,070347$	$-12,070347$

Au prix de quelques opérations arithmétiques, l'erreur finale est passée de $7\ 10^{-2}$ à 10^{-6}.

8.8. Intégration de Gauss

Nous abandonnons maintenant l'hypothèse que les pivots sont régulièrement répartis sur l'axe ; nous cherchons au contraire à les répartir « au mieux » pour avoir un algorithme aussi exact que possible. Nous aurons toujours

$$I = \int_a^b f(x)dx = \sum_1^n w_j f(a_j) + E.$$

Ici, la méthode des coefficients indéterminés est assez malcommode ; lorsque nous imposons que la formule ci-dessus soit exacte pour $x^0, x^1, \ldots, x^k, \ldots$, nous obtenons un système d'équations non-linéaires (en a_j) dès que $k > 1$, système dont la solution est en général inaccessible.

Nous allons plutôt utiliser un raisonnement indirect, à partir du polynôme d'interpolation de Hermite. La formule d'interpolation de Hermite s'écrit (voir §4.7) :

$$f(x) = \sum_1^n h_j(x) f(a_j) + \sum_1^n \bar{h}_j(x) f'(a_j) + [\pi(x)]^2 \frac{f^{(2n)}(\xi)}{(2n)!}.$$

Le terme d'erreur disparaît si f est un polynôme de degré $2n-1$ au plus. Intégrons terme à terme cette relation entre les abscisses a et b :

$$I = \int_a^b f(x)dx = \sum_1^n H_j f(a_j) + \sum_1^n \bar{H}_j f'(a_j) + E$$

où H_j et $\bar{H}_j$ sont respectivement les intégrales de h_j et de $\bar{h}_j$ entre les mêmes bornes. Cette formule d'intégration est exacte pour tout polynôme de degré inférieur à $2n$: c'est le mieux que nous puissions faire avec les $2n$ paramètres H_j, a_j.

Pour obtenir la forme annoncée, il faut que les $\bar{H}_j$ soient tous nuls. Comment parvenir à ce résultat ? Les seuls paramètres encore ajustables dont nous disposons sont les a_j. Nous devons donc choisir ces nombres de telle manière que l'intégrale de $\bar{h}_j$ soit nulle. Rappelons la forme de ce polynôme : $\bar{h}_j(x) = (x - a_j)[\ell_j(x)]^2$, si ℓ_j est le polynôme élémentaire de Lagrange construit sur les $\{a_j\}$. En utilisant une fois la relation $\ell_j = \pi(x)/((x - a_j)\pi'(a_j))$, nous obtenons

$$\bar{H}_j = \int_a^b (x - a_j)[\ell_j(x)]^2 dx = \int_a^b \pi(x) \frac{\ell_j(x)}{\pi'(a_j)} dx.$$

Nous voulons que cette intégrale soit nulle. Autrement dit, nous voulons que les polynômes π et ℓ_j soient orthogonaux sur le segment $[a, b]$ par rapport à la fonction de poids $w \equiv 1$.

Vous savez que $\pi(x)$ (défini en (4.5)) est le produit de tous les termes de la forme $x - a_j$: ce polynôme est défini (implicitement) par l'ensemble de ses zéros. Il doit être orthogonal aux ℓ_j, polynômes élémentaires de Lagrange pour n pivots, donc de degré $n-1$. Plutôt que d'imposer cette condition particulière, nous imposons une condition plus générale : $\pi(x)$ devra être orthogonal à tout polynôme de degré $n-1$ ou inférieur. Nous vous laissons le soin de démontrer la proposition réciproque et nous admettons que la condition nécessaire et suffisante pour que la formule de Gauss soit valable est que $\pi(x)$ soit orthogonal à tout polynôme de degré inférieur, pour une fonction de poids égale à l'unité, sur $[a, b]$.

A quelques détails près, vous connaissez des polynômes répondant à cette définition : ce sont les polynômes de Legendre, les P_n de la section 7.9.1. Pour que l'identification soit parfaite, il faut ramener l'intervalle d'intégration à [-1,1] et, si l'on veut être très soigneux, normaliser correctement $\pi(x)$. Le changement de variable $x = \frac{1}{2}(a+b) - \frac{1}{2}(a-b)t$ permet d'intégrer sur $[-1, 1]$:

$$I = \int_a^b f(x)dx = \frac{1}{2}(b-a)\int_{-1}^{1} f(t)dt.$$

Si maintenant nous choisissons comme pivots a_j les zéros du polynôme de Legendre de degré n, $\pi(x)$ coïncidera, à un facteur constant près, avec ce polynôme ; sur le segment $[-1, 1]$ il sera donc orthogonal à tout polynôme de degré inférieur, et en particulier aux ℓ_j.

Il reste à calculer les poids, à partir de l'expression des h_j. Calculons pour cela l'intégrale de la fonction particulière ℓ_i. Nous trouvons

$$\int_{-1}^{1} \ell_i(x)dx = \sum_1^n H_j \ell_i(a_j).$$

Le terme d'erreur est nul (pourquoi ?). Nous savons que $\ell_i(a_j) = \delta_{ij}$ (formule (4.6)), si bien que

$$H_j \equiv w_j = \int_{-1}^{1} \ell_j(x)dx.$$

Á l'aide de la formule de Darboux–Christoffel et de beaucoup d'algèbre, on peut déduire l'expression plus pratique :

$$w_j = \frac{-2}{(n+1)P_n'(a_j)P_{n+1}(a_j)}, \quad j = 1, 2, \ldots, n. \tag{8.13}$$

Il existe des tables des arguments et des poids pour l'intégration de Gauss–Legendre, pour n allant jusqu'à au moins 128. Pour éviter les erreurs de transcription, il vaut mieux écrire un sous-programme pour les recalculer.

Exemple – Calculons la même intégrale qu'au paragraphe 8.6.3, avec trois pivots. Les tables donnent :

a_j	0	$\pm 0,774597$
w_j	8/9	5/9

Le changement de variable $y = x - 2$ transforme l'intervalle $[1, 3]$ en $[-1, 1]$ et l'intégrant en $1/(y + 2)$. Un calcul élémentaire donne alors $I \cong 1,098039$, pour une erreur relative un peu supérieure à $5\ 10^{-4}$.

8.9. Généralisations de la méthode de Gauss

En choisissant un autre intervalle d'intégration et une autre fonction de poids, on peut construire de nouvelles formules d'intégration de type gaussien. Nous savons par exemple que les polynômes de Laguerre sont orthogonaux sur $[0, \infty\ ]$ par rapport à la fonction de poids e^{-x}. Nous pouvons donc écrire :

$$\int_0^\infty e^{-x} f(x) dx = \sum_1^n w_j f(a_j) + E,$$

où les a_j sont les zéros du polynôme de Laguerre d'ordre n et les w_j des poids calculables à l'aide d'une formule semblable à (8.13). Chaque famille de polynômes orthogonaux donne ainsi naissance à une formule d'intégration. Il ne faut pas s'empresser de conclure que l'on peut intégrer n'importe quelle fonction entre zéro et l'infini, au prix du calcul des valeurs de f en quelques points. Par ailleurs, aucun algorithme numérique ne pourra transformer une intégrale divergente en intégrale convergente. Écrivons quelques formules pour préciser la question.

$$\int_0^\infty g(x) dx = \int_0^\infty e^{-x} [e^x g(x)] dx = \int_0^\infty e^{-x} f(x) dx$$

où $f = e^x g$; ces relations n'ont rien d'anormal si g est telle que ces intégrales convergent, mais c'est lorsque nous tenterons d'appliquer la formule de Gauss–Laguerre à f que les problèmes apparaîtront. Si g ne décroît pas assez vite à l'infini, le terme d'erreur, en $f^{(2n)}$, deviendra rapidement intolérable.

Que faire si la précision sur l'intégrale obtenue par la méthode de Gauss est insuffisante ? Il existe ici encore deux possibilités : utiliser une formule d'ordre supérieur ou employer une méthode composée. Comme d'habitude, la première solution est à rejeter. La seconde est assez simple à mettre en oeuvre. Nous divisons l'intervalle $[a, b]$ en m sous-intervalles, $[x_m, x_{m+1}]$. Pour chaque sous-intervalle, nous faisons un changement de variable qui ramène l'intervalle d'intégration à $[-1, 1]$ et nous employons une formule de Gauss d'ordre peu élevé. Comme aucun pivot ne coïncide avec les bords des intervalles, on n'économise aucun calcul, à la différence des méthodes de Newton fermées. Cependant, la précision est largement supérieure à ce que l'on obtient avec Newton–Cotes ; de plus, l'existence de discontinuités en a ou b est peu gênante, puisque ces points ne sont pas des pivots.

8.10. Les intégrales généralisées

Une intégrale généralisée convergente est mathématiquement bien définie ; l'ennui, c'est que l'ordinateur ne le sait pas. Par exemple, le calcul analytique de

$$I = \int_0^\infty x^n e^{-x} dx$$

est facile, mais, par définition, un algorithme comporte un nombre fini d'opérations. Comment alors intégrer numériquement jusqu'à $+\infty$? Vous connaissez une réponse dans ce cas particulier : on peut employer la méthode de Gauss–Laguerre.

Pour certains intégrants, nous pouvons calculer des intégrales généralisées en faisant un changement de variable astucieux. Par exemple, si $a > 0$, nous pouvons écrire :

$$\int_a^\infty f(x)dx = \int_0^{1/a} \frac{1}{t^2} f\left(\frac{1}{t}\right) dt.$$

Un changement de variable analogue est valable pour $a < 0$.

Il peut aussi arriver qu'un même changement de variable soit malcommode sur l'ensemble de l'intervalle d'intégration : il faut alors couper celui-ci en deux ou plusieurs morceaux.

Supposons maintenant que $f(x)$ est équivalente à $(x-a)^{-1/2}$ pour x proche de a ($a < b$). Le changement de variable $x = a + t^2$ fera disparaître cette singularité intégrable :

$$\int_a^b f(x)dx = \int_0^{\sqrt{b-a}} 2t f(a+t^2)dt.$$

8.11. Les intégrales multiples

Nous examinons, pour fixer les idées, le cas d'une intégrale double ; nous cherchons l'intégrale de $f(x,y)$ dans un domaine $\mathcal{D}$. Les valeurs extrêmes de x dans $\mathcal{D}$ sont x_1 et x_2 ; pour une valeur donnée de x, les valeurs extrêmes de y sont $y_1(x)$ et $y_2(x)$. Intégrons d'abord par rapport à y, puis par rapport à x :

$$J = \int\int f(x,y)dxdy = \int_{x_1}^{x_2} dx \int_{y_1(x)}^{y_2(x)} f(x,y)dy.$$

Après avoir fait le choix d'une méthode numérique pour approcher l'intégrale en y, nous pouvons programmer le calcul d'une fonction dont la valeur sera :

$$I(x) = \int_{y_1(x)}^{y_2(x)} f(x,y)dy$$

où les bornes y_1 et y_2 dépendent de x. Nous sommes maintenant capables de calculer l'intégrale de $I(x)$:

$$J = \int_{x_1}^{x_2} I(x)dx.$$

Remarque : Si l'on utilise un sous-programme d'intégration (« trapèze » par exemple), qui sera appelé aussi bien pour calculer $I(x)$ que J, il faut prendre garde à ne pas laisser croire au programme que l'on utilise une procédure récursive, s'appelant elle-même.

Cet algorithme ne peut guère se généraliser à plus de dimensions, car le nombre d'évaluations de la fonction f augmente de façon prohibitive avec le nombre de dimensions.

8.12. L'INTÉGRALE SANS PEINE

Le logiciel Scilab comprend plusieurs sous-programmes d'intégration numérique. Voici deux exemples, dans le cas d'une fonction définie par une formule. Le premier est la fonction `integrate` qui reçoit quatre arguments : deux chaînes de caractères définissant la fonction à intégrer puis la variable d'intégration et deux réels, les limites d'intégration.

```
-->integrate("(2/sqrt(%pi))*exp(-x*x)","x",0,1)
ans  =
   0.8427008
```

La fonction `intg` est plus souple ; elle admet trois arguments : les deux limites d'intégration et le nom de la fonction à intégrer. Celle-ci peut être définie dans un fichier différent et être aussi compliquée que vous voulez (tout en restant intégrable !). Elle fait appel à un algorithme « adaptatif », capable d'augmenter le nombre de noeuds dans les régions où la fonction varie rapidement.

```
-->function y = fn(x), y = (2/sqrt(%pi))*exp(-x*x), endfunction
-->intg(0,1,fn)
ans  =
   0.8427008
```

Finalement, vous pouvez vérifiez les résultats précédents :

```
-->erf(1)
ans  =
   0.8427008
```

Scilab peut aussi intégrer, par la méthode des trapèzes, une fonction calculée et rangée dans un tableau (`inttrap`).

Maple sait calculer de nombreuses intégrales sous forme analytique mais, dès qu'une formule contient un nombre réel, le logiciel effectue un calcul numérique :

```
> int(-x*log(x)/(1-x), x = 0..1);
```

$$-1 + \frac{\pi^2}{6}$$

```
> int(-x*log(x)/(1.0-x), x = 0..1);
```

$$0.6449340668$$

8.13. Pour en savoir plus

De nombreux algorithmes de quadrature numérique ont dû être omis du texte. Citons les plus courants (ces mots-clés sont un point de départ pour une recherche sur la Toile ou en bibliothèque) : intégration adaptative, algorithme de Clenshaw–Curtis, méthode de Gauss–Kronrod.

- R. Théodor : *Initiation à l'analyse numérique*, ch. 5 (Masson, Paris, 1994).
- M. Schatzman : *Analyse numérique, une approche mathématique*, ch. 8 (Dunod, Paris, 2001).
- Polycopiés des cours d'analyse numérique de MM. E. Hairer et G. Wanner : ch. 1, intégration numérique :
  `http://www.unige.ch/~hairer/polycop.html`
- Sur le site de l'École polytechnique fédérale de Lausane, cours d'analyse numérique de J. Rappaz : `http://iacs.epfl.ch/asn/teaching.html`
  et compléments du livre : J. Rappaz, M. Picasso : *Introduction à l'analyse numérique* (Presses Polytechniques et Universitaires Romandes, Lausanne, 2004).
- W.H. Press, S.A. Teukolsky, W.T. Vetterling et B.P. Flannery : *Numerical Recipes, The Art of Scientific Computing*, ch. 4 (Cambridge University Press, Cambridge, 2007).
- Intégrales multiples :
  `http://www-fp.mcs.anl.gov/ccst/research/reports_pre1998/mcs/numerical_integration/napierala.html`

8.14. Exercices

Exercice 1

Soit f une fonction suffisamment continue dans l'intervalle $[a-h, a+h]$. Supposons que l'expression $Df \equiv A_- f(x-h) + A_0 f(x) + A_+ f(x+h)$ soit une approximation numériquement convenable de sa dérivée seconde au point x. Nous estimons $f(x \pm h)$ grâce au théorème de Taylor

$$f(x \pm h) = f(x) \pm \frac{h}{1!} f'(x) + \frac{h^2}{2!} f''(x) \pm \frac{h^3}{3!} f^{(3)}(x) + \frac{h^4}{4!} f^{(4)}(\xi_\pm).$$

Déterminer les coefficients A_-, A_0, A_+ pour que $Df = f''$ jusqu'au terme en h^2 compris. Que vaut le terme d'erreur ?

Exercice 2

Soient T_1 et T_2 deux approximations d'une même intégrale

$$J \equiv \int_a^b f(x)dx$$

obtenues par la méthode des trapèzes composée avec les pas h et $h/2$, respectivement.

a) Quelles sont les abscisses utilisées pour ces deux calculs ?

b) On combine T_1 et T_2 pour obtenir une extrapolation de Richardson. Préciser les valeurs de la fonction f qui interviennent. Quel procédé d'intégration classique obtient-on alors ?

Exercice 3

On souhaite calculer une intégrale définie à l'aide de l'algorithme :

$$\int_a^b f(x)dx \simeq Af(a) + Mf(\frac{a+b}{2}) + Bf(b).$$

Utiliser la méthode des coefficients indéterminés pour trouver les paramètres A, M et B.

Exercice 4

On veut calculer l'intégrale

$$J = \int_{-1}^{1} \frac{e^{-x}}{\sqrt{1-x^2}} dx.$$

a) Cette intégrale est-elle définie ?

b) Pourrait-on calculer J par la méthode des trapèzes composée ?

c) On décide d'utiliser la méthode du point milieu composée. Effectuer ce calcul avec $h = 1$, pour obtenir le résultat J_1, puis avec $h = 1/3$, pour trouver le résultat $J_{1/3}$.

d) On voudrait faire un nouveau calcul, avec un pas plus petit ; quelle valeur de h faut-il choisir pour utiliser au mieux les valeurs de l'intégrant déjà calculées ?

e) L'étude théorique montre que l'erreur de troncation de la méthode du point milieu utilisant un pas h est proportionnelle à h^2 :

$$J = J_h + Ch^2.$$

Comment peut-on combiner les deux résultats précédents (J_1 et $J_{1/3}$) pour obtenir une valeur plus précise de l'intégrale, où le terme d'erreur en h^2 a disparu ?

Exercice 5

Une boucle de courant de rayon a est disposée dans le plan xOy ; son centre coïncide avec l'origine et son axe avec Oz. Les formules de Biot et Laplace impliquent que la composante verticale du champ magnétique en un point du plan xOy, à une distance r de l'axe est donnée par l'intégrale

$$B_z = C \int_0^{\pi} \frac{a^2 - ar\cos\theta}{(a^2 + r^2 - 2ar\cos\theta)^{3/2}} d\theta. \tag{8.14}$$

a) Trouver un changement de variable qui permette d'écrire B_z sous la forme

$$B_z = C' \int_0^{\pi} \frac{1 - r'\cos\theta}{(1 + r'^2 - 2r'\cos\theta)^{3/2}} d\theta. \tag{8.15}$$

b) Utiliser la méthode de Simpson composée pour calculer le champ à la distance $r = a/2$ de l'axe. On prendra des valeurs de θ distantes de $\pi/6$.

Exercice 6

a) On connaît les valeurs d'une fonction f pour trois abscisses équidistantes, x_0, x_1, x_2. Former le polynôme de Lagrange $L(x)$ qui interpole la fonction f aux points x_0, x_1, x_2. En désignant par h l'intervalle entre pivots, en posant $x = x_0 + mh$, exprimer $L(x)$ en fonction de m et h et des valeurs de f, soit f_0, f_1 et f_2.

b) On souhaite calculer l'intégrale définie :

$$J = \int_{x_0}^{x_2} f(x)dx$$

à l'aide d'une formule à trois points : $J \simeq a_0 f_0 + a_1 f_1 + a_2 f_2$. Utiliser le polynôme $L(x)$ trouvé en (a) pour déterminer les coefficients a_i.

c) Calculer numériquement l'intégrale :

$$J_1 = \frac{2}{\sqrt{\pi}} \int_0^2 \exp(-x^2)dx$$

avec $h = 0,5$.

Exercice 7

On considère deux points x_0 et x_1 distants de h, pour lesquels on dispose des valeurs de la fonction f et de sa dérivée f'.

a) Construire le polynôme de Hermite, $H(x)$ qui interpole la fonction et sa dérivée aux points x_0 et x_1. On pose $x = x_0 + mh$; exprimer alors ce polynôme en fonction de h, m et des valeurs f_0, f_1 de la fonction et f'_0, f'_1 de sa dérivée.

b) Utiliser le polynôme trouvé en (a) pour calculer les valeurs des coefficients a_i et b_i de la formule d'intégration suivante :

$$J = \int_{x_0}^{x_2} f(x)dx \simeq a_0 f_0 + a_1 f_1 + b_0 f'_0 + b_1 f'_1$$

c) Appliquer au calcul de J_1 (définie à l'exercice 6), avec $h = 1$.

Exercice 8

Utiliser la méthode des coefficients indéterminés pour calculer les arguments $\{x_i\}$ et les poids $\{w_i\}$ de la méthode d'intégration de Gauss à un ou à deux points pour l'intervalle $[-1, 1]$:

$$\int_{-1}^{1} f(x)dx \cong w_0 f(x_0); \quad \int_{-1}^{1} f(x)dx \cong w_1 f(x_1) + w_2 f(x_2)$$

Exercice 9

On donne les arguments et les poids de la méthode d'intégration de Gauss–Legendre d'ordre 5 :

a_i	w_i
$\pm 0,906180$	$0,236927$
$\pm 0,538469$	$0,478629$
0	$0,568889$

Calculer, en utilisant 5 valeurs de la fonction, l'intégrale

$$J = \int_0^1 \sqrt{u}du.$$

Quelle est la valeur exacte de J, quelle est l'erreur de cette approximation ?

Exercice 10

On demande de calculer :

$$J = \int_0^\infty \frac{1}{1+x^2}dx.$$

Pour cela, on décomposera l'intervalle d'intégration en deux parties : [0,2] et [2,∞], et on appliquera, après avoir effectué les changements de variables nécessaires, dans

chaque intervalle une formule de Gauss à 3 points dont les paramètres figurent ci-dessous.

$$w_1 = w_3 = 5/9; \quad w_2 = 8/9; \quad a_3 = -a_1 = 0,774596; \quad a_2 = 0.$$

Exercice 11

La densité d'énergie dans le rayonnement du corps noir s'exprime comme une intégrale sur toutes les fréquences :

$$G(T) = \frac{8\pi h}{c^3} \int_0^\infty \frac{\nu^3}{\exp(\frac{h\nu}{kT}) - 1} d\nu \equiv AT^4 \int_0^\infty \frac{x^3}{e^x - 1} dx \equiv AT^4 \int_0^\infty f(x)dx,$$

ce qui définit la fonction $f(x)$ et la constante A. ν est la fréquence, $h \simeq 6,6 \times 10^{-34}$ J.s la constante de Planck, $k = R/N_A \simeq 1,4 \times 10^{-23}$ JK^{-1} la constante de Boltzmann et T le température absolue. On prendra T= 1 K dans la suite.

a) Quel est le changement de variable qui permet de passer de ν à x ? Vérifier que x est sans dimensions. Donner l'expression de A et déterminer ses dimensions. Sont-elles compatibles avec le fait que G est une densité d'énergie ?

b) Une première approximation de G peut être obtenue de la façon suivante. On calcule numériquement l'intégrale $G_0 = \int_0^{x_1} f(x)dx$. Pour $x > x_1$, on remplace f par une fonction voisine f_1, dont l'intégrale $G_1 = \int_{x_1}^\infty f_1(x)dx$ est calculable analytiquement. On choisit $x_1 = 6$. Appliquer la méthode de Simpson composée, avec $h = 1$ pour calculer G_0. Proposer une forme de f_1 et terminer le calcul de G. Comment peut-on valider le choix $x_1 = 6$?

c) On veut calculer G par intégration de Gauss–Laguerre à 4 points. Sous quelle forme doit-on écrire l'intégrale pour que cette méthode puisse s'appliquer ? On donne, pour 4 points :

i	0	1	2	3
x_i	0,322548	1,74576	4,53662	9,39507
w_i	0,603154	0,357419	0,038888	0,00053929

Évaluer G.

Exercice 12

Les polynômes de Tschebychef ont été présentés au chapitre 7. Ils sont définis pour $-1 \leq x \leq 1$ par

$$T_n(x) = \cos(n \arccos\ x)$$

et vérifient la relation

$$\int_{-1}^{1} \frac{T_k(x)T_\ell(x)}{\sqrt{1-x^2}}dx = 0 \quad \text{si} \quad k \neq \ell.$$

a) Déterminer les zéros de T_n.

b) Les polynômes de Tschebychef sont à la base d'une méthode d'intégration dite de Gauss–Tschebychef. En tenant compte de la relation d'orthogonalité, montrer que la forme générale des intégrales que l'on pourra calculer par cette méthode est

$$I = \int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}dx \simeq \sum_{1}^{n} w_k f(x_k)$$

Quelles seront les abscisses à choisir ? On démontre que les poids correspondants sont $w_k \equiv \pi/n$ pour une méthode à n points et que le terme d'erreur est :

$$E = \frac{2\pi}{2^{2n}(2n)!} f^{(2n)}(\xi) \quad ; \quad -1 \leq \xi \leq 1.$$

(c) Calculer, avec 5 chiffres significatifs, l'intégrale

$$J = \int_{-1}^{1} \frac{e^x}{\sqrt{1-x^2}}dx.$$

8.15. PROJET

Champ magnétique et lignes de champ

Le champ magnétique créé en un point $\vec{r}_t$ de l'espace par un courant I circulant dans un circuit filiforme est donné par l'une ou l'autre des expressions

$$\vec{B} = \frac{\mu_0 I}{4\pi} \int \frac{\vec{dr}_s \wedge \vec{R}_{st}}{R_{st}^3} = \frac{\mu_0 I}{4\pi} \overrightarrow{\text{rot}} \left(\int \frac{\vec{dr}_s}{R_{st}} \right), \tag{8.16}$$

où $\vec{r}_s$ désigne un point quelconque de la source du champ (le conducteur) et $\vec{R}_{st} \equiv \vec{r}_t - \vec{r}_s$. On se propose de calculer numériquement le champ produit par quelques systèmes simples de conducteurs et de tracer les lignes de champs correspondantes.

1. Calculer numériquement le champ créé en un point quelconque de l'espace par une spire de rayon a centrée sur l'origine et disposée dans le plan xOy. On remarque que ce problème admet une symétrie cylindrique et que l'on peut donc se contenter de calculer le champ en tout point d'un plan méridien, par exemple xOz. Il est commode d'utiliser le rayon de la spire comme unité de longueur. On obtient une bonne précision en employant une méthode d'intégration d'ordre 4, comme la méthode de Simpson.

2. Une ligne de champ est une courbe telle que, en tout point, le vecteur $\vec{B}$ soit tangent à la ligne de champ passant en ce point. Si le point M (x, z) appartient à une ligne de champ, alors un petit déplacement (dx, dz) conduira à un point voisin sur la même ligne à condition que

$$\frac{dx}{B_x} = \frac{dz}{B_z}. \tag{8.17}$$

Ces relations représentent des équations différentielles pour x et z que l'on peut résoudre pas à pas par la méthode d'Euler (voir §11.2). Il est prudent de normaliser à l'unité le champ :

$$dx \sim \frac{B_x}{\sqrt{B_x^2 + B_z^2}}\,; \qquad dz \sim \frac{B_z}{\sqrt{B_x^2 + B_z^2}}. \tag{8.18}$$

Écrire un programme destiné à tracer les lignes de champ pour un champ dont les valeurs ont été précédemment calculées et rangées dans un tableau. Appliquer au cas de la spire. Comparer vos résultats avec le cas classique d'un dipole électrique.

3. Trouver la répartition de champ créée par deux spires en position de Helmholtz (deux spires identiques coaxiales distantes de D parcourues par un courant de même sens). Pour quelle valeur de D l'homogénéité du champ au centre du montage est-elle la meilleure ? Que se passe-t-il si on inverse le courant dans l'une des spires ? Quel est l'équivalent en électrostatique ? À quoi pourrait servir un tel champ ?

4. Tracer les lignes de champ d'un solénoïde, représenté comme l'assemblage de dix spires identiques, avec une distance entre spires voisines de l'ordre de $a/5$.

This page intentionally left blank

Chapitre 9

Analyse spectrale, transformation de Fourier numérique

L'analyse spectrale est une activité très courante, au moins sous forme qualitative. On attribue à Newton la première étude de la décomposition de la lumière blanche par un prisme. Un musicien peut reconnaître les trois ou quatre notes composant un accord. Rayleigh est sans doute l'un des premiers physiciens à utiliser l'analyse spectrale quantitativement. Il a calculé l'intensité lumineuse en sortie d'un interféromètre de Michelson en fonction de la composition spectrale de la lumière émise par la source. Depuis, l'usage de la transformée de Fourier (TF) est devenu presque universel, de la théorie des systèmes linéaires à la spectroscopie (infra-rouge et Raman, résonance magnétique nucléaire, spectroscopie de masse), la reconstruction d'images (imagerie par résonance magnétique, tomographie axiale par rayons X, « scanner », radioastronomie, exploration sismique), en passant par la diffraction, l'holographie et la traitement du signal.

Pendant longtemps, la transformation de Fourier a été un outil presque exclusivement théorique. L'apparition d'ordinateurs performants très peu coûteux et l'invention d'un algorithme de transformation de Fourier rapide ont complètement modifié les conditions d'application de ce formalisme.

Nous examinons, dans ce chapitre, ce qu'il faut entendre par calcul numérique d'une transformée de Fourier et nous montrons comment réaliser ce calcul. Avant d'entrer dans le vif du sujet, il nous paraît utile de présenter une vue d'ensemble des méthodes d'analyse spectrale associées au nom de Fourier.

9.1. Les méthodes de Fourier

9.1.1. Série de Fourier

Soit f une fonction périodique de période L, intégrable sur une période. Considérons d'autre part la série de Fourier S définie par la formule (9.1) :

$$S = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos \frac{2n\pi x}{L} + b_n \sin \frac{2n\pi x}{L} \right], \tag{9.1}$$

avec

$$a_n = \frac{2}{L}\int_0^L f(x)\cos\frac{2\pi nx}{L}dx, \quad n = 0, 1, 2, \ldots,$$

$$b_n = \frac{2}{L}\int_0^L f(x)\sin\frac{2\pi nx}{L}dx, \quad n = 1, 2, 3, \ldots.$$

Si f satisfait aussi aux conditions suivantes (dites conditions de Dirichlet) : f est bornée, monotone par morceaux et présente un nombre fini de singularités sur une période, alors la série de Fourier S converge vers la demi-somme

$$S(x) = \frac{1}{2}[f(x+0) + f(x-0)]. \tag{9.2}$$

Si f est continue en x, alors $S(x) = f(x)$.

Nous pouvons écrire une formule équivalente à 9.1 en termes d'exponentielles complexes :

$$f(x) = \sum_{-\infty}^{\infty} \alpha_n e^{2i\pi nx/L}, \tag{9.3}$$

$$\alpha_n = \frac{1}{L}\int_0^L f(x)e^{-2i\pi nx/L}dx. \tag{9.4}$$

Vous voyez que la fonction f est déterminée dès lors que nous connaissons les nombres L et $\{a_n, b_n\}$ ou encore L et $\{\alpha_n\}$. À une fonction périodique correspond une suite discrète (dénombrable) de valeurs.

9.1.2. Intégrale ou transformée de Fourier (TF)

La transformée de Fourier d'une fonction $f(x)$ est définie comme

$$F(s) \equiv \int_{-\infty}^{\infty} f(x)e^{-2i\pi sx}dx. \tag{9.5}$$

On connaît la transformation inverse qui s'écrit

$$f(x) \equiv \int_{-\infty}^{\infty} F(s)e^{2i\pi xs}ds, \tag{9.6}$$

Une condition suffisante (mais non nécessaire) pour que ces relations soient vraies est que f et F soient absolument intégrables et que f soit bornée en module. On dit que F est la transformée de Fourier de f ou encore que f, F forment une paire de fonctions conjuguées de Fourier. Une autre notation courante est $F = \mathcal{F}(f)$ et aussi $f \Leftrightarrow F$.

Remarque : Le « sinus cardinal » $\operatorname{sinc} x \equiv \frac{\sin x}{x}$ est un exemple d'une fonction non absolument intégrable mais dont la transformée de Fourier existe.

La transformation de Fourier fait correspondre à une fonction quelconque une autre fonction quelconque, non-périodique. Les variables x et s sont réelles mais les fonctions peuvent être à valeurs complexes.

Il est possible de montrer que ces définitions peuvent s'appliquer aussi aux « fonctions généralisées » (ou « distributions ») comme la « fonction » de Dirac ou la « fonction » de Heaviside, un aspect que nous n'aborderons pas ici, malgré son importance théorique.

9.1.3. Vocabulaire et notations

Dans chaque domaine d'application de la TF, on a tendance à utiliser un vocabulaire particulier. Ainsi, dans le cas du traitement de signal, on considérera un signal dépendant du temps, $g(t)$, et de son analyse en fréquence, ou spectre, $G(f)$. Dans le cas de l'optique de Fourier, on parle plus volontiers de déplacement et de fréquence (ou de pulsation) spatiale. Nous avons choisi une notation neutre qui n'avantage aucun domaine, mais il arrivera que, pour fixer les idées, nous utilisions le couple de variables temps-fréquence.

Il existe de nombreuses définitions de la TF légèrement différentes de la notre. L'une des plus courantes (mais moins symétrique) est

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx; \quad f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{ix\omega} d\omega.$$

Dès que l'on emploie les exponentielles complexes, on doit admettre l'existence de fréquences (ou de pulsations) négatives aussi bien que positives : c'est une conséquence inévitable du formalisme. Dans certaines applications (échographie Doppler, RMN), on considère des différences de fréquence par rapport à une référence (la « porteuse »), et ces différences peuvent être des deux signes. Les fréquences négatives s'introduisent alors assez naturellement.

9.1.4. Échantillonnage

Les anciens appareils de mesure fournissaient toujours des données continues, ou « analogiques ». Pour pouvoir être traitées par un ordinateur, ces données devaient être « numérisées » ou « échantillonnées » (on dit parfois, comme en anglais, digitalisées). Les appareils plus récents fournissent directement des données numériques.

Il y a deux « dimensions » à la numérisation. Supposons que la mesure fournisse une tension électrique fonction du temps, soit $v(t)$. L'échantillonnage implique que la tension n'est connue qu'en certains instants : la fonction continue $v(t)$ est remplacée par une suite de nombres $\{v(t_i)\}$, en général régulièrement espacés. D'autre part, la tension est convertie en nombre par un convertisseur analogue-numérique et $v(t)$ devient une fonction en escalier (fonction étagée). Cet aspect, plus difficile à traiter,

sera masqué dans la suite ; nous supposerons que les convertisseurs ont une résolution très grande.

Y a-t-il perte d'information lors d'un échantillonnage ? La réponse à cette question est contenue dans le théorème de Shannon, aussi appelé théorème d'échantillonnage. Soient $u(t), U(f)$ deux fonctions transformées de Fourier l'une de l'autre ; la première est définie (par exemple) sur l'espace des temps, la seconde sur l'espace des fréquences. Nous supposons que

$$U(f) = 0 \text{ si } |f| \geq f_0/2.$$

En d'autres termes, $u(t)$ n'a pas de composantes de fréquences supérieures à $f_0/2$, son spectre est limité. La quantité $f_0/2$ est souvent appelée fréquence de Nyquist, notée f_{Ny}. $1/f_{Ny} = 2/f_0$ est donc la plus courte période présente dans $u(t)$.

Nous supposons que nous nous sommes procurés des échantillons de u équidistants de τ (la période d'échantillonnage), les nombres $\{u(k\tau)\}$. Dans ces conditions, le théorème de Shannon affirme que la fonction u peut être exactement reconstituée à partir des $u(k\tau)$ selon la formule

$$u(t) = \sum_{k=-\infty}^{\infty} u(k\tau) \frac{\sin\left[\pi(f_0 t - k)\right]}{\pi(f_0 t - k)}, \tag{9.7}$$

à condition de choisir $\tau = 1/f_0$. Qualitativement, nous pouvons dire que $u(t)$ sera « bien reproduite » si nous disposons d'au moins deux valeurs de u par période de la composante la plus rapidement variable.

9.1.5. Transformée de Fourier d'une fonction échantillonnée (TFTD)

Une fonction nous est connue par ses échantillons f_n ; nous supposerons que

$$\sum_{n=-\infty}^{\infty} |f_n|^2 < \infty$$

pour assurer la convergence des expressions qui vont suivre. Vous remarquez que cette condition exclut les fonctions périodiques. Nous définissons sa transformée de Fourier par la relation

$$F(s) = \sum_{n=-\infty}^{\infty} f_n e^{-2i\pi ns}. \tag{9.8}$$

Tous les termes de la somme sont périodiques, de période multiple de $1/2\pi$: F est donc elle-même périodique. La situation est symétrique de celle décrite à propos des séries de Fourier : f est discrète, F est périodique.

Nous ne détaillerons pas ici les propriétés de la transformée de Fourier à temps discret (TFTD), bien que cette opération mathématique joue un rôle important en traitement du signal.

Il nous reste à envisager une quatrième combinaison de caractéristiques. La fonction de départ est discrète **et** périodique, la fonction image (ou d'arrivée) est, elle-aussi,

discrète **et** périodique. L'opération mathématique qui fait passer de l'une à l'autre s'appelle la transformée de Fourier discrète (TFD). Étant donné son importance pratique, nous lui consacrons un paragraphe spécial.

9.2. Transformée de Fourier discrète (TFD)

9.2.1. Définition

Étant donné un nombre τ (le pas d'échantillonnage), un entier N et une fonction réelle ou complexe f, nous considérons la suite de valeurs

$$\{f(n\tau)\} = \{f_n\} = f(0), f(\tau), f(2\tau), \ldots, f((N-1)\tau).$$

La transformée de Fourier discrète (TFD) du tableau (ou du vecteur) $\{f_n\}, 0 \leq n \leq N-1$ est aussi une suite finie de N nombres définis comme

$$F_k \equiv \sum_{n=0}^{N-1} f_n e^{-2i\pi kn/N}, \quad k = 0, 1, \ldots, N-1. \tag{9.9}$$

Remarque : Les échantillons $f_n = f(n\tau)$ sont **numérotés de 0 à $N-1$**. Tous les textes traitant de la TFD utilisent cette convention pour les indices, alors que bien des logiciels (Maple, Scilab, Matlab) numérotent à partir de 1, ce qui est une source certaine de confusion.

Il existe une transformation inverse qui permet de construire les f_n à partir de la séquence $\{F_k\}$:

$$f_n \equiv \frac{1}{N} \sum_{k=0}^{N-1} e^{2i\pi nk/N} F_k, \quad n = 0, 1, \ldots, N-1. \tag{9.10}$$

À cause des propriétés de l'exponentielle, tous les termes qui apparaissent dans les expressions (9.9) et (9.10) sont périodiques, de période N. Par exemple, p étant un entier,

$$e^{2i\pi n(k+pN)/N} = e^{2i\pi nk/N} e^{2i\pi np} = e^{2i\pi nk/N}.$$

Il en découle que F_k est périodique. La formule 9.10 peut être utilisée pour n extérieur à l'intervalle $[0, N-1]$ pour obtenir des répétitions de valeurs prises dans cet intervalle. La définition répond donc bien à ce que nous avions annoncé : les deux fonctions conjuguées par une TFD sont discrètes et périodiques ; elles sont représentées par le même nombre d'échantillons.

9.2.2. La TFD comme approximation de l'intégrale de Fourier

Pour justifier qualitativement la définition qui vient d'être donnée, examinons une approximation numérique simple de l'intégrale de Fourier. Nous considérons une fonction $f(t)$ nulle en dehors de l'intervalle $[0, T]$. Nous répartissons dans cet intervalle N

noeuds (pivots), régulièrement espacés de $\tau = T/N$, pour lesquels nous connaissons les valeurs de $f_n = f(n\tau)$ de la fonction. La TF s'écrit

$$F(s) = \int_{-\infty}^{\infty} f(t)e^{-2i\pi st}dt = \int_{0}^{T} f(t)e^{-2i\pi st}dt$$

et nous cherchons la valeur de cette intégrale en certains points à définir de l'axe des s. L'approximation la plus simple est celle des rectangles à droite (paragraphe 8.5) :

$$F(s) \simeq \tau \sum_{n=0}^{N-1} f_n \exp(-2i\pi st_n).$$

En utilisant les définitions précédentes de τ et de t_n et en choisissant $s_k = k/T$, il vient

$$F_k = F(s_k) \simeq \frac{T}{N} \sum_{n=0}^{N-1} f_n \exp\left(-2i\pi \frac{k}{T} n \frac{T}{N}\right)$$

ce qui reproduit, à un facteur près, la définition (9.9).

Supposons maintenant que la fonction $f(t)$ ne soit différente de zéro que dans l'intervalle $[-T/2, T/2]$. Il est alors commode d'approcher l'intégrale par la méthode des rectangles à gauche. Les noeuds pour l'intégration sont alors $t_n = nT/N$ avec $n = -N/2+1, -N/2+2, \ldots, N/2$. La TF est calculée aux abscisses $s_k = k/T$, mais $k = -N/2+1, \ldots, N/2$. Nous avons alors la relation

$$F_k \simeq \frac{T}{N} \sum_{n=-N/2+1}^{N/2} f_n e^{-2i\pi nk/N}.$$

Un raisonnement tout à fait analogue permet de relier les coefficients d'une série de Fourier aux valeurs de la TFD.

Les développements précédents nous paraissent satisfaisants du point de vue intuitif, mais ils sont, en toute rigueur, faux. Le problème n'est pas directement lié à la mauvaise qualité de la méthode des rectangles mais à l'hypothèse faite par souci de simplicité. La TF d'une fonction nulle en dehors d'un intervalle (comme notre $f(t)$) s'étend sur tout l'axe des s : la condition d'application du théorème de Shannon n'est pas remplie. Une analyse plus précise permet de calculer l'écart entre TF et TFD et de montrer que celui-ci tend vers zéro lorsque τ tend vers zéro.

9.2.3. Notation matricielle pour la TFD

Il est commode d'introduire maintenant une notation matricielle. Nous posons

$$W_N = e^{2i\pi/N}. \tag{9.11}$$

W_N est l'une des racines N-ièmes de l'unité (ou encore une solution de $z^N - 1 = 0$). Avec cette définition, la formule (9.9) s'écrit

$$F_k = \sum_{n=0}^{N-1} W_N^{-kn} f_n, \quad k = 0, 1, 2, \ldots, N-1.$$

Introduisons les vecteurs colonnes $\boldsymbol{f} \equiv [f_0, f_1, \ldots, f_{N-1}]^T$, $\boldsymbol{F} = [F_0, F_1, \ldots, F_{N-1}]^T$ et la matrice $\boldsymbol{V}$ d'éléments $V_{ij} \equiv W_N^{-ij}$. La TFD prend la forme d'un produit matrice × vecteur :

$$\boldsymbol{F} = \boldsymbol{V}\boldsymbol{f}. \tag{9.12}$$

Nous sommes tentés d'écrire aussi

$$\boldsymbol{f} = \boldsymbol{V}^{-1}\boldsymbol{F} \tag{9.13}$$

et, par identification avec la formule (9.10), nous trouvons que

$$\boldsymbol{V}^{-1} = \frac{1}{N}\boldsymbol{V}^*,$$

ce qui indique que $\mathbf{V}$ est unitaire, au facteur $1/N$ près. Vérifions directement ce résultat en calculant un élément de la matrice produit $\boldsymbol{V}\boldsymbol{V}^*$:

$$\begin{aligned}
(\boldsymbol{V}\boldsymbol{V}^*)_{mn} &= \sum_{k=0}^{N-1} V_{mk}(\boldsymbol{V}^*)_{kn} = \sum_{k=0}^{N-1} W_N^{-mk} W_N^{kn}, \\
&= \sum_{k=0}^{N-1} W_N^{k(n-m)} = \sum_{k=0}^{N-1} z^k, \text{ en posant } z \equiv W_N^{n-m}, \\
&= \begin{cases} N & \text{si } z = 1, \\ \dfrac{1-z^N}{1-z} & \text{sinon,} \end{cases} \\
&= N\delta_{mn}.
\end{aligned}$$

Nous avons utilisé une formule connue pour la somme des termes d'une progression géométrique (de raison z et de premier terme 1) et les propriétés des W : $z = W_N^{n-m} = 1$ si $n - m = 0 \bmod N$ et $z^N = W_N^{N(n-m)} = 1$ pour prouver que $\boldsymbol{V}\boldsymbol{V}^* = N\boldsymbol{I}$. En général, les W satisfont à

$$W_N^{-nk} = W_N^{-\ell}, \text{ avec } \ell = nk \bmod N.$$

Si nous nous intéressons au calcul numérique de $\boldsymbol{F}$, nous constatons qu'il faut effectuer N^2 multiplications (en nombres complexes) et $N(N-1)$ additions pour obtenir ce vecteur. La majeure partie des applications de la DFT concerne de grands vecteurs (dix mille échantillons ou plus). Pour d'autres applications comme le traitement d'images, on doit calculer les TFD de chaque ligne et de chaque colonne d'une grande matrice. Il est donc essentiel de diminuer le nombre d'opérations. Nous disposons pour cela de l'algorithme de Cooley et Tuckey, proposé en 1965, ainsi que de nombreuses variantes. Cet algorithme est de complexité $N \log_2 N$. Si $N = 2^{14} = 16384$, $N^2 \simeq 2,68\,10^8$ et $N \log_2 N \simeq 2,29\,10^5$, un gain d'un facteur supérieur à mille.

9.3. Transformée de Fourier rapide (TFR)

On appelle transformée de Fourier rapide un algorithme capable, comme son nom l'indique, de calculer « rapidement » une transformée de Fourier discrète. Le nombre de points N peut être quelconque mais les algorithmes ne sont simples que si N est une puissance de 2, $N = 2^p$, ce que nous supposerons dans la suite.

9.3.1. Algorithme de Cooley–Tukey ou « entrelacement en temps »

Décrivons maintenant un algorithme de TFR. Nous commençons par écrire F_k en séparant les f_n de rang pair et impair

$$\begin{aligned} F_k &= \sum_{n=0}^{N/2-1} [W_N^{-2kn} f_{2n} + W_N^{-k(2n+1)} f_{2n+1}] \\ &= \sum_{n=0}^{N/2-1} W_N^{-2kn} f_{2n} + W_N^{-k} \sum_{n=0}^{N/2-1} W_N^{-2kn} f_{2n+1}, \\ k &= 0, 1, \ldots, N-1. \end{aligned}$$

La remarque suivante est la clé de l'algorithme :

$$W_N^{-2kn} = W_{N/2}^{-kn}.$$

Elle nous permet d'écrire F_k sous la forme

$$F_k = G_k + W_N^{-k} H_k, \quad k = 0, 1, 2, \ldots, N/2-1,$$

où G_k et H_k sont des TFD portant sur $N/2$ points. De plus, ces quantités étant périodiques, de période $N/2$, nous pouvons exprimer les F_k manquants comme

$$F_k = G_{k-N/2} + W_N^{-k} H_{k-N/2}, \quad k = N/2, N/2+1, \ldots N-1.$$

Chaque F_k apparaît comme une combinaison linéaire d'un G et d'un H.

L'intérêt de cette décomposition est qu'elle économise des opérations. Rappelez-vous que le calcul direct des F_k nécessite N^2 multiplications. Supposons que les G_k, H_k aient été obtenus par la méthode directe : cela a coûté $2(N/2)^2$ multiplications, auxquelles il faut ajouter N multiplications par W_N^{-k}, pour un total de $N^2/2+N$ opérations. Ce nombre va encore diminuer, car nous allons en fait décomposer à nouveau, de façon récursive, les TFD partielles.

Renommons temporairement g_n les f_n de rang pair et h_n ceux de rang impair, avec $0 \leq n \leq N/2-1$. Les nombres intermédiaires G_k s'écrivent alors

$$\begin{aligned} G_k &= \sum_{n=0}^{N/2-1} g_n W_{N/2}^{-kn} = \sum_{n=0}^{N/4-1} [g_{2n} W_{N/2}^{-2nk} + g_{2n+1} W_{N/2}^{-k(2n+1)}] \\ &= \sum_{n=0}^{N/4-1} W_{N/4}^{-nk} g_{2n} + W_{N/2}^{-k} \sum_{n=0}^{N/4-1} W_{N/4}^{-nk} g_{2n+1} \\ &= \begin{cases} I_k + W_{N/2}^{-k} J_k, & k = 0, 1, \ldots, N/4-1 \\ I_{k-N/4} + W_{N/2}^{-k} J_{k-N/4} & k = N/4, N/4+1, \ldots, N/2-1. \end{cases} \end{aligned}$$

Nous avons utilisé la relation $W_{N/2}^{-2nk} = W_{N/4}^{-nk}$ pour exprimer les $\{G_k\}$ comme une somme de deux TFD à $N/4$ points chacune. Vous vérifierez facilement que le calcul des $\{F_k\}$, en utilisant deux décompositions, implique $N^2/4 + 2N$ multiplications.

La récurrence se poursuit tant que le nombre de points impliqués dans une TFD est divisible par 2. Comme nous avons supposé que $N = 2^p$, nous pouvons opérer $p = \log_2 N$ décompositions de sommes. La dernière étape fait intervenir des couples de valeurs de f comme f_n et $f_{n+N/2}$. Les coefficients W sont alors $W_N^0 = 1$ et $W_N^{N/2} = -1$.

Représentons par un schéma (qui est censé symboliser le flux de données d'une étape à l'autre de l'algorithme) le procédé qui vient d'être décrit, dans le cas $N = 8$ (voir figure 9.1) ; nous écrirons $W_8 = w = e^{i\pi/4}$. Les résultats du calcul (les coordonnées de $\boldsymbol{F}$) proviennent des G_k et H_k, qui sont eux-mêmes issus d'une TFD des éléments de rang pair ou impair de $\boldsymbol{f}$. La façon de réaliser cette TFD n'est pas précisée pour l'instant. Le diagramme signifie que, par exemple, $F_2 = G_2 + w^{-2}H_2$ ou encore que $F_4 = G_0 + w^{-4}H_0$.

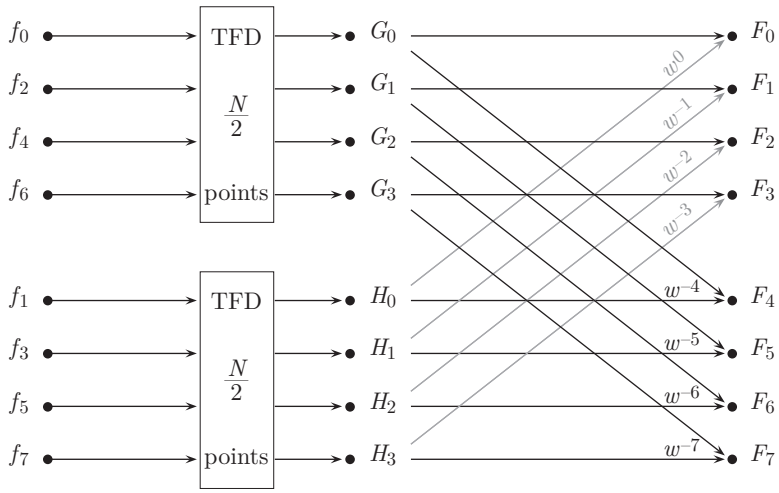


Figure 9.1 – Première étape de la construction d'une TFR.

Comment avons nous obtenu les G_k et les H_k que l'on trouve dans la figure 9.1 ? Par deux TFD portant sur 4 points chacune. Posons maintenant $w' = W_{N/2} = i$.

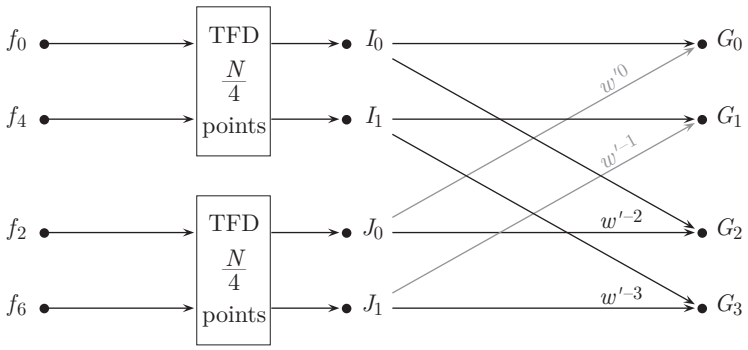


Figure 9.2 – Deuxième étape de la construction d'une TFR.

Le diagramme (voir figure 9.2) illustre la construction des quantités intermédiaires G_k à partir des I_k, J_k, eux-mêmes issus d'une TFD à deux points. Il faut à nouveau séparer les termes de rang pair de ceux de rang impair : f_0, f_4 d'un coté, f_2, f_6 de l'autre.

Ce diagramme nous indique qu'il faut effectuer, entre autres, l'opération $G_1 = I_1 + w'^{-1} J_1$. Les nombres H_k se calculent de façon analogue. Nous en arrivons au dernier stade logique de l'algorithme (mais le premier à être exécuté). Il faut réaliser les TFR à deux points qui permettent le calcul des I_k, J_K à partir des f_n. Interviennent maintenant les coefficients $W_8^0 = 1$ et $W_8^4 = -1$, selon un schéma souvent appelé le « papillon », figure 9.3.

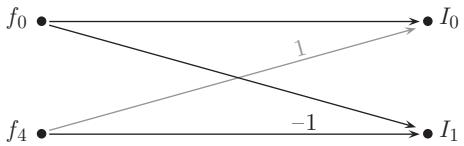


Figure 9.3 – Troisième étape de de la construction d'une TFR.

Nous pouvons réunir les trois diagrammes précédents en un seul, pour obtenir le schéma complet d'une TFR portant sur huit valeurs 9.4.

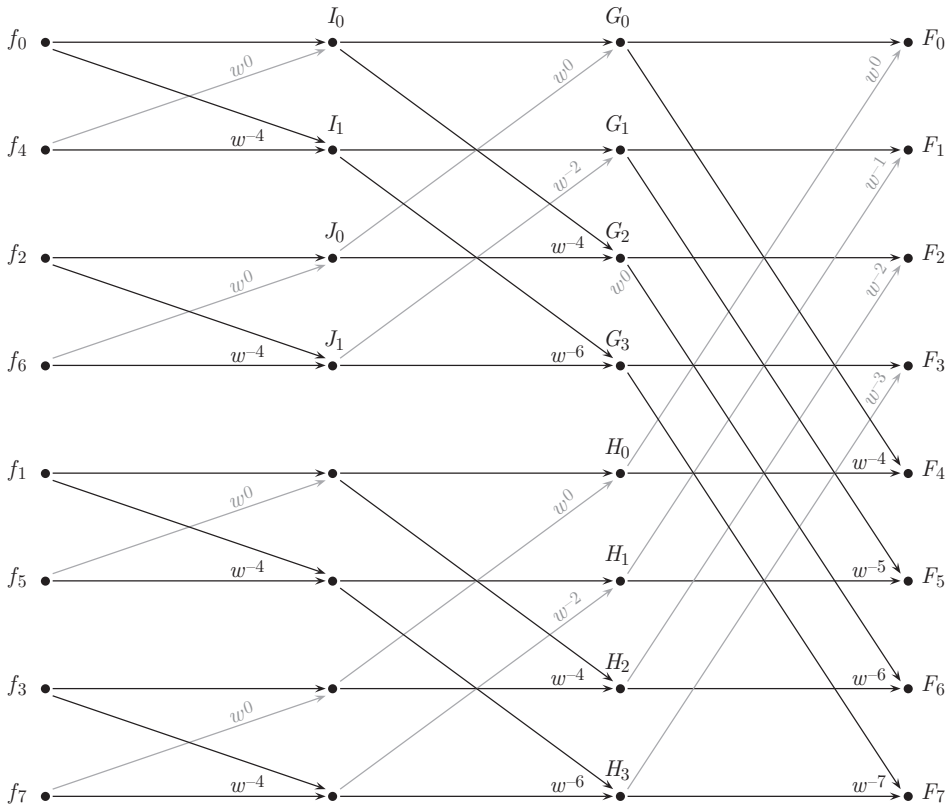


Figure 9.4 – Schéma complet d'une TFR à $N = 8$ valeurs.

Cet algorithme est qualifié « d'entrelacement en temps » (« decimation in time » en anglais). On imagine que les données sont les valeurs échantillonnées d'un signal temporel et qu'on les réorganise à la manière d'un mille-feuilles avant de les transformer. Vous constatez que, pour obtenir le résultat dans l'ordre normal $(F_0, F_1, \ldots, F_{N-1})$, il faut partir de données rangées dans un ordre bien particulier : $[f_0, f_4, f_2, f_6, f_1, f_5 f_3, f_7]$. Si nous utilisons des f_n rangés dans l'ordre habituel, nous obtiendrons des F_k permutés, qu'il faudra réorganiser. Ces deux possibilités sont en fait équivalentes. Tous les algorithmes de Fourier rapides incorporent une étape de permutation des données ou des résultats, pour retrouver l'ordre habituel.

9.3.2. Le renversement binaire

Lors d'un calcul portant sur huit ou seize valeurs, il n'est pas difficile de suivre les transformations successives des données pour établir la permutation convenable. Mais comment allons nous faire dans le cas, bien plus fréquent en pratique, de 10^4 ou 10^5 valeurs ? En utilisant la recette aussi simple qu'élégante qui suit, que nous appellerons le « retournement binaire ». Pour connaître l'emplacement que doit occuper la donnée f_k avant la transformation de Fourier rapide : écrire k en numération binaire (k_2), renverser ce nombre : le premier chiffre devient le dernier, le deuxième, l'avant-dernier et ainsi de suite (k'_2), reconvertir en décimal (k'), qui est le rang cherché. Voici l'illustration pour notre exemple à $N = 8$ points.

k_{10}	k_2	k'_2	k'_{10}
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Nous vous proposons, à titre d'exercice, d'écrire les quelques lignes d'un programme capable d'effectuer cette transformation.

9.3.3. Factorisation de la matrice V et variantes de l'algorithme TFR

Nous avons vu que la TFD pouvait s'écrire sous la forme

$$\boldsymbol{F} = \boldsymbol{V}\boldsymbol{f}.$$

Le passage de la TFD à la TFR peut se concevoir comme une décomposition de la matrice $\mathbf{V}$ en un produit de facteurs « plus simples » ou, plus précisément, contenant

plus de zéros. Nous nous contenterons d'énoncer le résultat, que vous pourrez vérifier avec un crayon, du papier et de la patience.

$$\boldsymbol{V} = \boldsymbol{V}_8 \boldsymbol{V}_4 \boldsymbol{V}_2 \boldsymbol{P}, \tag{9.14}$$

avec les définitions suivantes, en posant toujours $w = W_8$:

$$\boldsymbol{V}_8 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & w^0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & w^{-1} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & w^{-2} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & w^{-3} \\ 1 & 0 & 0 & 0 & w^{-4} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & w^{-5} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & w^{-6} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & w^{-7} \end{bmatrix},$$

$$\boldsymbol{V}_4 \equiv \left[\begin{array}{cccc:cccc} 1 & 0 & w^0 & 0 & & & & \\ 0 & 1 & 0 & w^{-2} & & & & \\ 1 & 0 & w^{-4} & 0 & & & 0 & \\ 0 & 1 & 0 & w^{-6} & & & & \\ \hdashline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hdashline & & & & 1 & 0 & w^0 & 0 \\ & & & & 0 & 1 & 0 & w^{-2} \\ & 0 & & & 1 & 0 & w^{-4} & 0 \\ & & & & 0 & 1 & 0 & w^{-6} \end{array}\right].$$

$\boldsymbol{V}_2$ est aussi une matrice bloc-diagonale, qu'il est commode d'écrire

$$\boldsymbol{V}_2 \equiv \begin{bmatrix} \boldsymbol{v}_2' & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{v}_2' & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{v}_2' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{v}_2' \end{bmatrix},$$

avec

$$\boldsymbol{v}_2' \equiv \begin{bmatrix} 1 & w^0 \\ 1 & w^{-4} \end{bmatrix}.$$

$\mathbf{0}$ est une matrice 2×2 dont tous les éléments sont nuls. Enfin, la matrice de permutation $\mathbf{P}$ vaut

$$\boldsymbol{P} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Vous remarquez que $\boldsymbol{P}$ est symétrique. La suite des opérations à effectuer apparaît clairement sur la formule (9.14) : permuter les données, effectuer les « papillons » d'ordre 2, puis ceux d'ordre 4 et enfin ceux d'ordre 8. Vous constatez aussi que chaque matrice $\boldsymbol{V}_p$ ne comporte que deux éléments non-nuls par ligne, dont l'un vaut un, prouvant ainsi que peu de multiplications sont nécessaires. En fait, l'algorithme requiert autant de multiplications qu'il y a de coefficients w^{-n} dans la figure 9.4.

Il existe de nombreuses variantes de l'algorithme que nous venons de décrire. Elles peuvent en général s'obtenir par des manipulations de la formule (9.14). En voici un exemple, qui utilise la symétrie de la matrice $\mathbf{V}$.

$$\boldsymbol{V} = \boldsymbol{V}^T = \boldsymbol{P}^T \boldsymbol{V}_2^T \boldsymbol{V}_4^T \boldsymbol{V}_8^T = \boldsymbol{P} \boldsymbol{V}_2^T \boldsymbol{V}_4^T \boldsymbol{V}_8^T. \tag{9.15}$$

En examinant les formules de définition, vous verrez que les matrices $\boldsymbol{V}_p^T$ ne comportent encore que deux éléments non nuls par ligne. L'algorithme représenté par la formule (9.15) est donc encore un algorithme de TFR qui ne diffère du précédent que par l'ordre des opérations et la valeur des coefficients. On le désigne par l'expression « entrelacement en fréquence » (« decimation in frequency »). D'autres algorithmes s'accommodent d'un nombre quelconque de points. L'algorithme de Cooley–Tukey est un assemblage d'opérations élémentaires impliquant chacune deux valeurs (le « papillon »). Pour N quelconque, on doit écrire une opération élémentaire pour chaque facteur premier de N.

9.4. Propriétés de la transformée de Fourier discrète

Nous allons examiner, de façon empirique, quelques propriétés de le TFD/TFR. Nous utiliserons pour cela la fonction `fft(y)` de Scilab. Cette fonction calcule en fait

$$Y_\ell = \sum_{m=1}^{N} y_m e^{-2i\pi(m-1)(\ell-1)/N}$$

puisque Scilab numérote les éléments de vecteurs à partir de 1. Vous pourrez obtenir les mêmes résultats, avec un temps de calcul plus long, avec `dft(y)`, qui n'utilise pas l'algorithme rapide. Dans un cas comme dans l'autre, N est automatiquement ajusté au nombre d'éléments du vecteur $\boldsymbol{y}$. Il existe aussi une fonction `ifft(Y)` qui effectue la transformation inverse. Un premier exemple concerne les échantillons de la fonction

$$y = e^{-\alpha x},\ x > 0, \quad y = 0,\ x < 0 \text{ et } \alpha \geq 0.$$

Nous savons que la TF de cette fonction s'écrit

$$Y(s) = \frac{1}{\alpha + 2i\pi s} = \frac{\alpha}{\alpha^2 + 4\pi^2 s^2} - \frac{2i\pi s}{\alpha^2 + 4\pi^2 s^2}.$$

Ces deux fonctions sont représentées ci-contre (figure 9.5), pour $\alpha = 6$.

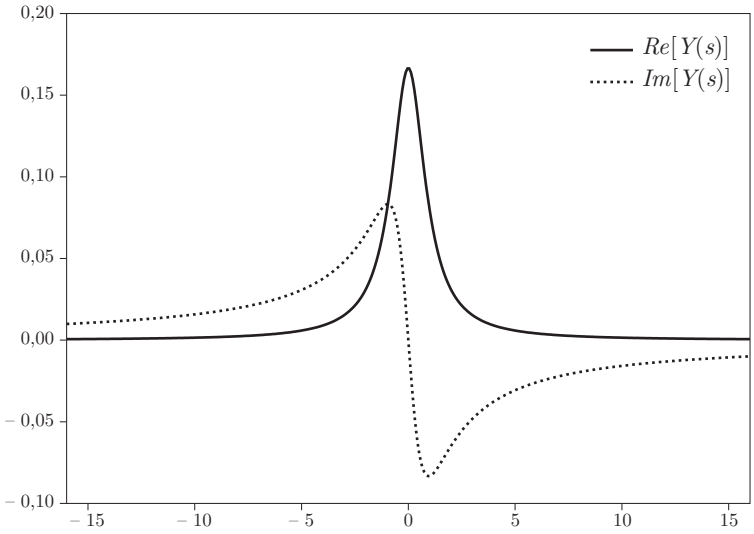


Figure 9.5 – Transformée de Fourier d'une exponentielle : partie réelle en trait plein, partie imaginaire en pointillé.

Remarque : Vous avez peut-être remarqué que la fonction y n'a pas été définie en $x = 0$. Or, nous savons que

$$y(0) = \int_{-\infty}^{\infty} Y(s)ds = \int_{-\infty}^{\infty} \Re[Y(s)]ds = \frac{1}{2}.$$

Par souci de cohérence, nous poserons donc $y(0) = \frac{1}{2}$.

Appliquons maintenant la TFD. Nous créons d'abord le vecteur $\boldsymbol{y}$, à l'aide d'instructions comme celles-ci :

```
N = 32;                                                    1
x = linspace(0,3.1,N)'; xa = (0:N-1)';                     2
alfa = input("coeff. exp. :");                             3
y = exp(-alfa*x)                                           4
```

Nous avons réparti 32 échantillons sur l'intervalle $[0 \ldots 3,1]$. Nous calculons ensuite la TFR par `Y = fft(y)` (ou `Y = dft(y)`). Il reste enfin à représenter graphiquement le résultat, en utilisant les éléments de `xa` comme abscisses (comme dans l'algorithme). Nous ne relions pas les points par des segments, pour souligner le caractère discret des « fonctions » considérées.

Les données (suite `y(n)`) apparaissent sur la figure 9.6. Il est important de se souvenir que la TFD suppose implicitement une fonction de départ y périodique ; vous ne voyez qu'une période sur la figure, de longueur 3,2. La période d'échantillonnage est de $\tau = 0,1$, la fréquence d'échantillonnage est de 10, la fréquence de Nyquist vaut 5 (nous utilisons des grandeurs sans dimension). Le premier point (quelle que soit la façon de le numéroter) correspond à la valeur $x = 0$; ceci fait que la TFR est particulièrement bien adaptée au traitement d'un signal « causal » ou nul pour $x < 0$.

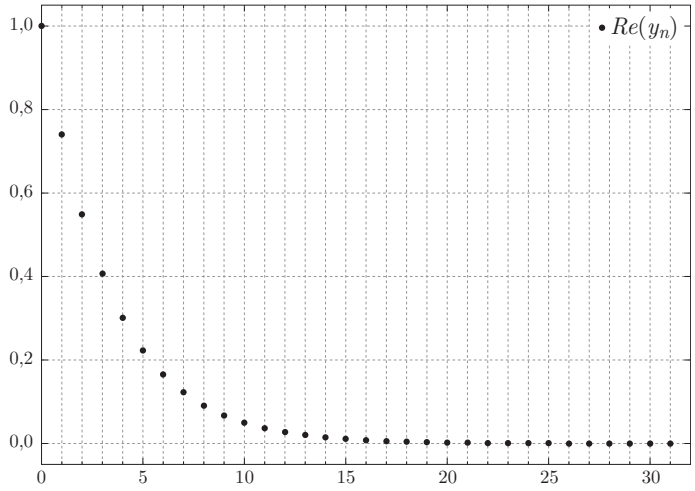


Figure 9.6 – Échantillons d'une fonction exponentielle.

Tournons-nous à présent vers le résultat fourni par `fft`, figure 9.7. Les Y_k constituent une suite infinie périodique dont une seule période a été conservée pour le tracé ; les points successifs sont encore numérotés de 0 à $N-1$.

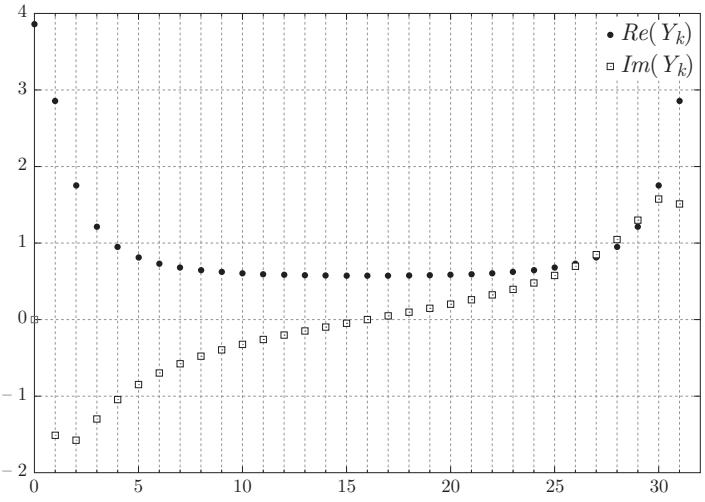


Figure 9.7 – TFD d'une fonction exponentielle.

Le premier point de l'espace des s correspond à la valeur $s = 0$. La valeur maximale de s est déterminée par le théorème de Shannon : c'est la moitié de la fréquence d'échantillonnage, soit $s_{Ny} = 5$ ici. Cette valeur est représentée par le point de rang $k = 15$. La distance entre points successifs de l'axe des s est donc de $\delta s = 5/16 = 0,3125$. Vous constatez que δs est l'inverse de la période dans l'espace des x, soit 3,2. Les valeurs suivantes de k correspondent à ce que nous appelons habituellement des fréquences négatives. Il est facile de s'en convaincre.

Explicitons Y_{N-1} :

$$Y_{N-1} = \sum_{n=0}^{N-1} y_n e^{-2i\pi(N-1)n/N} = \sum_{n=0}^{N-1} y_n e^{-2i\pi n} e^{2i\pi n/N} = \sum_{n=0}^{N-1} y_n e^{-2i\pi(-1)n/N} = Y_{-1}.$$

On aurait plus généralement $Y_{N/2+p} = Y_{p-N/2}$ avec le cas particulier $Y_{N/2} = Y_{-N/2}$. Cette présentation est souvent gênante ; c'est pourquoi Scilab vous offre la fonction `Z = fftshift(Y)` qui réorganise les éléments de Y pour que $s = 0$ revienne au centre de l'intervalle : si Y est un vecteur de taille N, Z vaut $Y([N/2+1 : N, 1 : N/2])$. La même fonction peut aussi s'appliquer à un tableau à deux dimensions. Pour certaines applications (traitement de signal et d'image en particulier), on fait des allers et retours entre l'espace des x et celui des s, ce dernier ne servant que d'intermédiaire. Le recours à `fftshift` est alors inutile.

Contrairement au résultat analytique, $\Re(Y_k)$ ne semble pas tendre vers zéro lorsque k augmente. Une partie de cette différence est due à ce que nous avons défini de façon peu soigneuse la suite y_n. Elle présente, comme la fonction y, une discontinuité en $n = 0$; il faut en fait poser $y_0 = \frac{1}{2}$. Cette remarque est à rapprocher du fait qu'une série de Fourier converge, à la hauteur d'une discontinuité, vers la moyenne des valeurs limites à droite et à gauche.

Il est banal de tenir compte des remarques précédentes dans le programme de démonstration. Il suffit d'insérer `y(0) = 0.5*y(0);` après la définition des y_n et d'utiliser `Y = fftshift(fft(y));`. La figure 9.8 montre que nous nous sommes rapprochés du résultat analytique.

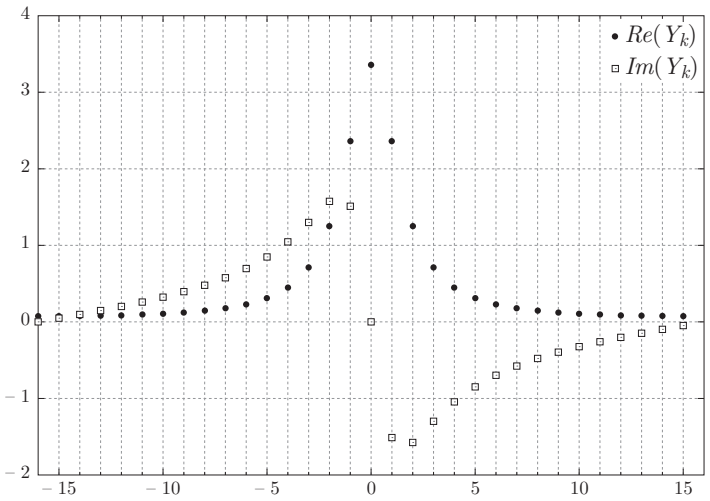


Figure 9.8 – TFD d'une fonction exponentielle : présentation améliorée.

Vous remarquez que le profil défini par les points Y_k est encore un peu différent de la courbe analytique. Une théorie plus détaillée montrerait que les Y_k peuvent être considérés comme obtenus en numérisant Y (avec un pas δs) et en répliquant cette séquence une infinité de fois, avec un décalage égal à $1/\tau$ entre deux répliques. Si la

période d'échantillonnage n'est pas assez courte, les différentes répliques empiètent les unes sur les autres, ce qui provoque la distorsion observée.

Considérons maintenant la fonction $y = e^{-\alpha x + 2i\pi f x}$ si $x > 0, y = 0$ si $x < 0$. La TF de cette fonction se déduit de la précédente par une translation de f. La figure 9.9 représente la TFD correspondante, pour les mêmes conditions de numérisation et $f = 2$. La même translation affecte la TFD, mais elle est moins reconnaissable à cause de la discrétisation des abscisses.

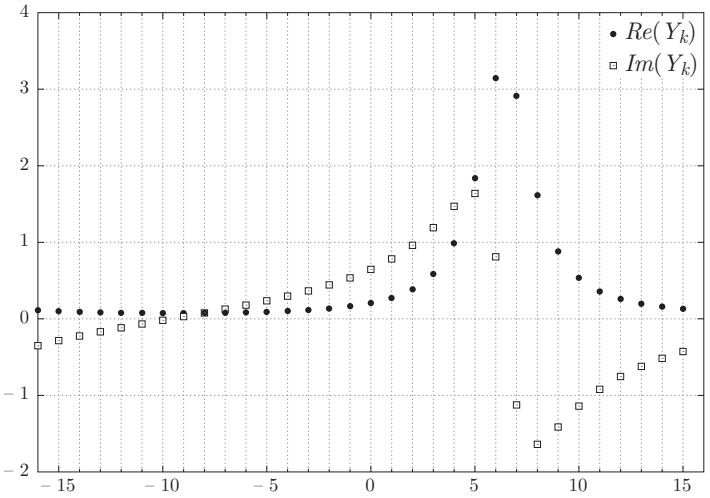


Figure 9.9 – TFD de la fonction $y = e^{-3x+4i\pi x}$.

Que se passe-t-il lorsque la fréquence de la sinusoïde approche ou dépasse la fréquence de Nyquist ? C'est ce que montre la figure 9.10, obtenue pour $f = 6$.

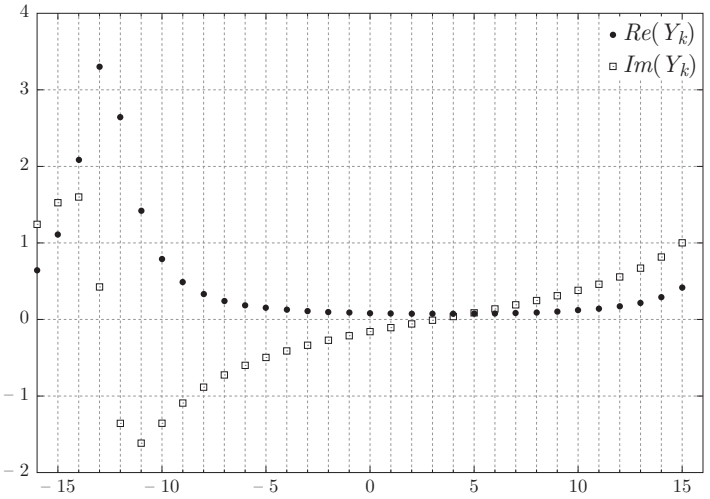


Figure 9.10 – TFD de la fonction $y = e^{-3x+12i\pi x}$.

Surprise ! La TFD étant périodique (de période 10 ici), elle présente un maximum à l'abscisse $s = 6 - 10 = -4$. Toute fonction y de fréquence extérieure à l'intervalle $[-5 : 5]$ sera traitée de la même façon : sa TFD subira des translations de ± 10 en nombre suffisant pour apparaître sur le segment $[-5 : 5]$. Ce comportement est appelé « aliasing » en anglais : une fréquence supérieure en valeur absolue à s_{Ny} acquiert un « pseudonyme » qui la représente. Ce déguisement de fréquence se comprend plus facilement dans l'espace des x, où il apparaît comme une conséquence directe de la numérisation. La figure 9.11 illustre ce propos. Elle représente les échantillons d'une sinusoïde amortie de fréquence 6, comme précédemment. Nous avons superposé à ces points une courbe en trait plein qui représente la fonction $y = \Re[e^{-x+12i\pi x}]$ et une courbe en pointillé d'équation $y = \Re[e^{-x-8i\pi x}]$. Comme vous le voyez, ces deux fonctions produisent la même séquence de valeurs de y_n. Vous constatez aussi que la fonction représentée en trait plein ne respecte pas les conditions du théorème de Shannon : elle admet moins de deux échantillons par période ; c'est le contraire pour la fonction tracée en pointillé. Encore une fois, une suite de valeurs discrètes ne détermine pas complètement la fonction continue dont elle est issue.

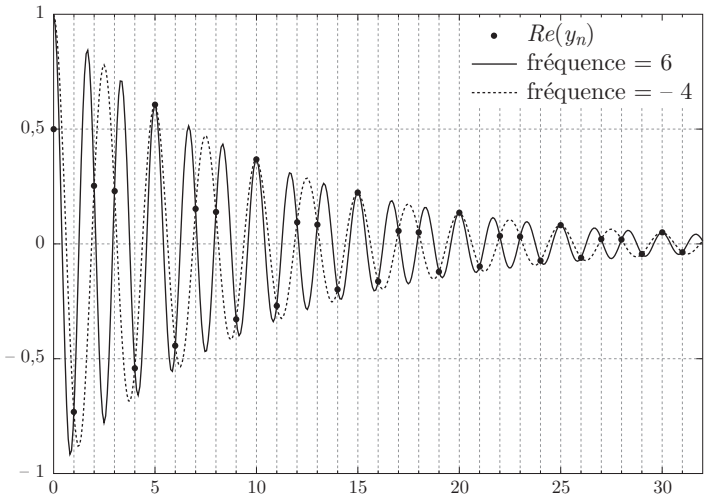


Figure 9.11 – Deux fonctions ayant la même suite d'échantillons.

Dans la pratique, on doit souvent calculer la TFD de fonctions compliquées présentant un spectre très large. Il faut alors adopter une période d'échantillonnage très courte et donc accumuler énormément d'échantillons ou exclure par filtrage, avant numérisation, les composantes rapidement variables.

Vous trouverez dans les ouvrages spécialisés, les énoncés de nombreuses autres propriétés de la TFD. Elles sont en général des transpositions des théorèmes correspondants qui s'appliquent à la TF.

9.5. Pour en savoir plus

Les livres (et les sites Internet) sur le traitement du signal et des images, sur la spectroscopie expérimentale constituent une mine inépuisable de présentations de la transformée de Fourier et de ses propriétés numériques. Il existe aussi d'autres méthodes d'analyse spectrale, comme la prédiction linéaire et les méthodes d'entropie maximale.

- E. Oran Brigham : *The fast Fourier transform*, (Prentice-Hall, 1974).
- B. Picinbono : *Théorie des signaux et des systèmes avec problèmes résolus*, (Dunod, Paris, 1989).
- W.H. Press, S.A. Teukolsky, W.T. Vettering, B.P. Flannery : *Numerical recipes, the art of scientific programming*, ch. 12 (Cambridge University Press, Cambridge, 2007).
- M. Schatzman : *Analyse numérique, une approche mathématique*, ch. 7 (Dunod, Paris, 2001).
- Sur le site `http://www.script.univ-paris-diderot.fr` : cours de J.P. Gazeau : Transformation de Fourier (ch. 6).
- `http://cristallo.epfl.ch/exercices/exercices_schiltz/2007-2008/BioDiff_Notes4a.pdf`
- `http://www.polytech.unice.fr/~leroux/presentationfourier/presentationfourier.html`
- `http://www.fftw.org/` (« the Fastest Fourier Transform in the West »).

9.6. Exercices

Exercice 1

Calculer la TFD de la suite $[1, 0, 0, 1]$, puis la TFD inverse de $[2, -1 - i, 0, -1 + i]$.

Exercice 2

Si f_n et F_k sont deux suites de N valeurs liées par une transformation de Fourier discrète, démontrer les relations

$$\sum_{n=0}^{N-1} f_n = F_0,$$

$$f_0 = \frac{1}{N} \sum_{k=0}^{N-1} F_k.$$

Exercice 3

f_n, F_k étant deux suites liées par une TFD, démontrer les relations

$$f_{n-\ell} \Longleftrightarrow e^{-2i\pi k\ell/N} F_k$$

$$e^{2i\pi \ell n/N} f_n \Longleftrightarrow F_{n-\ell}$$

Exercice 4

Soient $f_n \Longleftrightarrow F_k$ et $g_n \Longleftrightarrow G_k$ deux couples de séquences liées par une TFD. On constitue la nouvelle suite

$$H_k \equiv \frac{1}{N} \sum_{\ell=0}^{N-1} F_\ell G_{k-\ell},$$

analogue discret d'un produit de convolution. Montrer que les H_k sont liés par une TFD aux $f_n g_n$:

$$f_n g_n \Longleftrightarrow \frac{1}{N} \sum_{\ell=0}^{N-1} F_\ell G_{k-\ell}.$$

Exercice 5

Avec les définitions de l'exercice précédent, vérifier que

$$\sum_{\ell=0}^{N-1} f_\ell g_{n+\ell} \Longleftrightarrow F_k^* \times G_k$$

(corrélation discrète).

Exercice 6

En calculant le produit de convolution de facteurs bien choisis, démontrer l'analogue discret du théorème de Parseval-Plancherel pour une suite f_n réelle

$$\sum_{n=0}^{N-1} f_n^2 = \frac{1}{N} \sum_{k=0}^{N-1} |F_k|^2$$

Exercice 7

La plus haute fréquence présente dans la musique jouée par un orchestre est de 21 kHz. Quelle est la fréquence d'échantillonnage minimale que l'on doit utiliser pour numériser convenablement ce son ? Combien faut-il d'échantillons pour enregistrer un clip de trois minutes ? Chaque échantillon est un nombre binaire de 12 bits ; quelle est la taille du fichier correspondant ?

Exercice 8

Le signal capté par un spectromètre de résonance magnétique nucléaire est une fonction du temps qui peut être assimilée à une superposition de sinusoïdes amorties exponentiellement. Ce signal est numérisé et on en calcule la transformée de Fourier pour obtenir ce que l'on appelle un spectre. Plus précisément, on s'intéresse à l'amplitude de la partie réelle de la TF comme fonction de la fréquence. Par rapport à une origine arbitraire, les fréquences présentes dans le signal s'étendent de -3000 à 6000 Hz.

a) Quelle période d'échantillonnage maximale faut-il choisir pour pouvoir reconstruire correctement le spectre ? On conserve cette période pour la suite de l'exercice.

b) On choisit de collecter $2^{14} = 16384$ échantillons. Quelle sera la durée de l'accumulation ?

c) A cause de l'amortissement, le signal disparaît au bout d'une seconde. Quelle est la fraction d'échantillons qui contiennent de l'information ? Pourrait-on rendre la numérisation plus efficace en changeant d'origine des fréquences ?

d) On sait que le spectre contient deux composantes dont les fréquences diffèrent de 2 Hz. Pourra-t-on les discerner dans les conditions de l'exercice ?

9.7. Projet

Diffraction de Fraunhofer (diffraction à l'infini)

La lumière que traverse une ouverture pratiquée dans un écran subit une diffraction. En lumière monochromatique, lorsque l'ouverture est éclairée par une onde plane et que l'observation a lieu assez loin pour que l'on puisse considérer comme planes les ondes diffractées (conditions de Fraunhofer), on montre que « l'amplitude complexe » de l'onde diffractée est la transformée de Fourier du facteur de transmission de l'ouverture (plus exactement la « fonction de transmittance » ou la « transparence pupillaire »). L'intensité diffractée est le carré du module de l'amplitude complexe. La transformée de Fourier inverse de l'intensité est appelée la fonction de transfert de l'ouverture ; elle est égale au produit de convolution de la transmittance par la fonction complexe conjuguée.

a) Diffraction par une fente

Une fente, de largeur faible devant sa hauteur, diffracte la lumière essentiellement dans la direction perpendiculaire à sa plus grande dimension et on peut limiter l'étude à cette seule direction. L'écran sera représenté par un vecteur $\boldsymbol{x}$ à $N = 2^p$ éléments, tous égaux à zéro. Pour simuler une fente de largeur L, on posera $x_n = 1$, $(N-L)/2 \leq n \leq (N+L)/2$. Calculer la TFR $\boldsymbol{X}$ de $\boldsymbol{x}$, puis l'intensité $\boldsymbol{E}$. Examiner l'effet d'un changement de valeur de L. Est-il utile de poser, comme indiqué dans le texte à propos de l'exponentielle, $x_{(N-L)/2} = x_{(N+L)/2} = 0,5$? Quel est le rôle du paramètre p ?

b) Diffraction par plusieurs fentes

Il est facile de traiter le cas de la diffraction par deux ou plusieurs fentes, en modifiant le vecteur $\boldsymbol{x}$. Examiner l'influence sur l'intensité diffractée de la distance entre fentes et du nombre de fentes.

c) Diffraction par une ouverture bidimensionnelle

Les objets à deux dimensions se traitent tout aussi facilement à l'aide de la fonction `fft2` de Scilab, associée à `fftshift`. Retrouver les figures classiques de la diffraction à l'infini par une ouverture rectangulaire ou circulaire.

d) Autres formes géométriques et assemblage de formes

Vous pouvez encore examiner les figures de diffraction produites par un triangle ou un hexagone. Il est instructif de comparer la symétrie de l'ouverture avec celle du diagramme de diffraction. Un « cristal » à deux dimensions peut être simulé en pratiquant dans l'écran plusieurs ouvertures régulièrement espacées.

Chapitre 10

Valeurs propres, vecteurs propres

La physique fournit un grand nombre de problèmes mathématiquement équivalents à la diagonalisation d'une matrice symétrique ou hermitique. C'est le cas de la recherche, par le calcul, des fréquences de vibration d'un système mécanique ou électrique ou encore des niveaux d'énergie d'un système quantique. En algèbre linéaire, on parle plutôt de détermination des « vecteurs propres » et des « valeurs propres » d'une matrice. C'est un sujet aussi bien connu que la résolution des systèmes linéaires. Nous nous limiterons dans ce chapitre essentiellement au cas des matrices réelles symétriques. Rappelons quelques propriétés des « éléments propres » d'une telle matrice.

Nous appelons valeurs propres de la matrice carrée symétrique $\boldsymbol{A}$ les solutions de l'équation en λ :

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0 \tag{10.1}$$

où $\boldsymbol{I}$ est la matrice identité. Si $\boldsymbol{A}$ et $\boldsymbol{I}$ sont de taille $n \times n$, les valeurs propres sont donc solutions de l'équation polynômiale de degré n :

$$(-1)^n(\lambda^n - C_{n-1}\lambda^{n-1} + \cdots - C_1\lambda) + C_0 = 0 \tag{10.2}$$

appelée « équation caractéristique » (le membre de droite est le « polynôme caractéristique »). Dans le cas d'une matrice symétrique $n \times n$, l'équation (10.2) admet n racines réelles, distinctes ou non. Comme il n'existe pas d'expression analytique des racines d'un polynôme de degré supérieur à 4, la recherche des valeurs propres est toujours un processus itératif. Un certain nombre de méthodes anciennes (Leverrier, Krylov) forment le polynôme caractéristique et déterminent ses zéros. Ces méthodes sont abandonnées, car, en l'absence d'hypothèses supplémentaires sur $\boldsymbol{A}$, elles sont lentes (calcul des C_i) et instables (recherche des racines). Par contre, ces deux opérations sont rapides pour une matrice tridiagonale. Vous verrez que l'une des méthodes de diagonalisation les plus performantes consiste à « tridiagonaliser » $\boldsymbol{A}$ en conservant ses valeurs propres, puis à calculer les racines de son polynôme caractéristique.

Étant donné une matrice $\boldsymbol{M}$ orthogonale, nous dirons que la matrice $\boldsymbol{A'}$ est semblable à $\boldsymbol{A}$ si $\boldsymbol{A'} = \boldsymbol{M}^T\boldsymbol{A}\boldsymbol{M}$; la transformation $\boldsymbol{A} \longrightarrow \boldsymbol{A'}$ s'appelle une similitude. On a

le théorème : $\boldsymbol{A'}$ et $\boldsymbol{A}$ ont mêmes valeurs propres. Le vecteur propre $\boldsymbol{u}$ associé à la valeur propre λ est défini par la relation :

$$\boldsymbol{Au} = \lambda \boldsymbol{u} \tag{10.3}$$

Dans le cas particulier d'une matrice symétrique, les vecteurs propres (au nombre de n) peuvent être choisis orthonormés : ils forment une base complète de $\mathbb{R}^n$. Soit $\boldsymbol{U}$ la matrice dont les colonnes sont les vecteurs propres. Vous pourrez vérifier que l'ensemble des n équations telles que (10.3) peut s'écrire :

$$\boldsymbol{AU} = \boldsymbol{U\Lambda} \tag{10.4}$$

si $\boldsymbol{\Lambda}$ désigne une matrice diagonale dont les éléments diagonaux sont les valeurs propres. De par sa définition, $\boldsymbol{U}$ est orthogonale, $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$. En effet, l'élément i, j du produit $\boldsymbol{U}^T\boldsymbol{U}$ est le produit scalaire de la ligne i de $\boldsymbol{U}^T$, soit le vecteur ligne $\boldsymbol{u}_i^T$, par la colonne j de $\boldsymbol{U}$, soit $\boldsymbol{u}_j$. Comme les vecteurs propres sont orthonormés, nous avons $(\boldsymbol{U}^T\boldsymbol{U})_{i,j} = \delta_{i,j}$; autrement dit, ce produit de matrices se réduit à la matrice unité. Finalement :

$$\boldsymbol{\Lambda} = \boldsymbol{U}^T\boldsymbol{AU} \tag{10.5}$$

ce qui traduit le théorème suivant. Une matrice symétrique peut toujours être diagonalisée au moyen d'une similitude ; la matrice de transformation est orthogonale et ses colonnes forment un jeu complet de vecteurs propres.

Remarque : Comme ce chapitre comporte beaucoup de résultats numériques fournis par Scilab, nous avons conservé le point décimal plutôt que la virgule.

10.1. Les éléments propres sans peine

Dans la suite, nous utilisons la matrice

$$\boldsymbol{F} = \begin{bmatrix} 1.0 & 1. & 0.5 \\ 1.0 & 1. & 0.25 \\ 0.5 & 0.25 & 2.0 \end{bmatrix}$$

comme exemple pour illustrer les algorithmes que nous allons présenter. Nous pouvons bien sûr demander à Scilab de faire les calculs ; ce logiciel propose deux fonctions. La première fournit uniquement les valeurs propres

```
-->spec(F)
  ans  =  - 0.0166473
            1.4801214
            2.5365259
```

alors que la deuxième

```
-->[D,X] = bdiag(F)
  X  =  !    .7212071  -  .5314834  -  .4442811 !
        ! -  .6863493  -  .4614734  -  .5621094 !
        ! -  .0937280  -  .7103293     .6976011 !

  D  =  ! -  .0166473    0.           0.         !
        !    0.          2.5365259    0.         !
        !    0.          0.           1.4801214 !
```

renvoie les vecteurs propres et les valeurs propres.

Remarque : L'ordre dans lequel apparaissent les valeurs propres (pour `spec`) ou les éléments propres (pour `bdiag`) est arbitraire (mais la correspondance entre la valeur propre et le vecteur propre est toujours conservée). De plus, Scilab traite les matrices hermitiennes à coefficients complexes par les mêmes instructions.

Maple vous propose, de façon analogue, les fonctions `eigenvalues` et `eigenvectors`, qui se trouvent dans la bibliothèque `linalg`.

10.2. Méthode de la puissance n-ième et méthodes dérivées

10.2.1. Puissance n-ième

Il arrive parfois que l'on recherche uniquement la plus grande des valeurs propres. La méthode de la puissance n-ième répond bien à ce besoin. L'algorithme est très simple. Ayant fait choix d'un vecteur initial $\boldsymbol{x}^{(0)}$, nous formons par récurrence :

$$\boldsymbol{x}^{(n)} = \boldsymbol{A}\boldsymbol{x}^{(n-1)} = \boldsymbol{A}^n\boldsymbol{x}^{(0)}. \tag{10.6}$$

Le vecteur $\boldsymbol{x}^{(n)}$ tend vers un vecteur propre associé à la plus grande valeur propre (en module) ; appelons-la λ_1. Le rapport de deux composantes homologues de $\boldsymbol{x}^{(n)}$ et $\boldsymbol{x}^{(n-1)}$, soit $x_k^{(n)}/x_k^{(n-1)}$, tend vers λ_1. La démonstration est facile dans le cas d'une matrice symétrique, bien que la méthode qui vient d'être exposée soit plus générale. Soient $\boldsymbol{u}^{(i)}, \lambda_i$ les éléments propres de $\boldsymbol{A}$. Développons $\boldsymbol{x}^{(0)}$ sur la base des $\boldsymbol{u}^{(i)}$:

$$\boldsymbol{x}^{(0)} = \sum_i c_i \boldsymbol{u}^{(i)}$$

Appliquons $\boldsymbol{A}$: $$\boldsymbol{x}^{(1)} = \boldsymbol{A}\boldsymbol{x}^{(0)} = \sum_i c_i \boldsymbol{A}\boldsymbol{u}^{(i)} = \sum_i c_i \lambda_i \boldsymbol{u}^{(i)}$$

puis $\boldsymbol{A}^2$: $$\boldsymbol{x}^{(2)} = \boldsymbol{A}\boldsymbol{x}^{(1)} = \sum_i c_i \lambda_i^2 \boldsymbol{u}^{(i)}.$$

Il est facile d'écrire le cas général si l'on se souvient de ce que $\boldsymbol{A}^k$ admet les mêmes vecteurs propres que $\boldsymbol{A}$ et les λ_i^k comme valeurs propres :

$$\boldsymbol{x}^{(k)} = \sum_i c_i \lambda_i^k \boldsymbol{u}^{(i)} = \lambda_1^k c_1 \boldsymbol{u}^{(1)} + \lambda_1^k \sum_{i=2} c_i (\lambda_i/\lambda_1)^k \boldsymbol{u}^{(i)}.$$

Comme $|\lambda_i/\lambda_1| < 1$, la somme du second membre tend vers zéro quand k croît : la propriété est démontrée.

En pratique, il arrive souvent que certaines composantes de $\boldsymbol{x}^{(k)}$ deviennent très grandes au cours du calcul. Pour éviter tout problème de dépassement de capacité, on normalise $\boldsymbol{x}^{(k)}$ avant de calculer $\boldsymbol{x}^{(k+1)}$. Une façon de faire consiste à imposer que $\max_m |x_m^{(k)}| = 1$, (normalisation au sens de la norme infinie). Pour l'exemple qui suit, nous avons choisi la norme euclidienne, simple à programmer. Dans ce cas, si $\boldsymbol{x}^{(k)} \simeq u^{(1)}$, alors $\boldsymbol{x}^{(k)T}\boldsymbol{A}\boldsymbol{x}^{(k)} \simeq \lambda_1 \boldsymbol{x}^{(k)}\boldsymbol{x}^{(k)} \simeq \lambda$.

Exemple – Cherchons la plus grande valeur propre de $\boldsymbol{F}$. Le programme comporte les instructions

```
x0 = [1,0,0]';                                    1
x1 = F*x0; x1 = x1/norm(x1);                      2
\\cinq itérations supplémentaires                 3
lambda6 = x6'*F*x6                                4
```

et produit les résultats suivants

```
x0        x1          x2          x3          x4          x5          x6
 1   0.6666667   0.6328463   0.5970940   0.5718624   0.5557210   0.5458476
 0   0.6666667   0.5976882   0.5473361   0.5135451   0.4924756   0.4797619
 0   0.3333333   0.4922138   0.5864316   0.6397224   0.6698074   0.6869344

 lambda6  =  2.5353767
```

La figure 10.1 représente la suite des vecteurs normalisés $\boldsymbol{x}^{(i)}$.

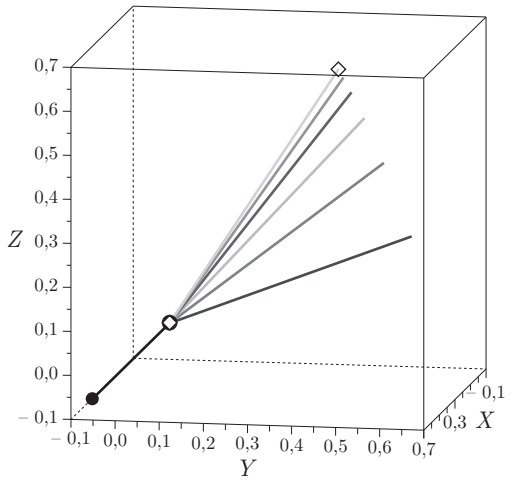


Figure 10.1 – Méthode de la puissance n-ième : vecteurs itérés à partir de $[1{,}0{,}0]^T$ (point noir).

10.2.2. Puissance n-ième avec décalage

La rapidité de convergence de l'algorithme précédent dépend des rapports $|\lambda_i/\lambda_1|$. La convergence sera lente s'il existe une autre valeur propre proche de λ_1 en module. Il est possible d'accélérer la convergence, en augmentant les différences entre valeurs propres. Il suffit de remarquer que $\boldsymbol{A}$ et $\boldsymbol{A'} = \boldsymbol{A} - s\boldsymbol{I}$ ont mêmes vecteurs propres (ces matrices commutent) et que leurs valeurs propres diffèrent par une translation. Si $\boldsymbol{u}$ et λ sont éléments propres de $\boldsymbol{A}$, alors :

$$(\boldsymbol{A} - s\boldsymbol{I})\boldsymbol{u} = \boldsymbol{A}\boldsymbol{u} - s\boldsymbol{I}\boldsymbol{u} = (\lambda - s)\boldsymbol{u}$$

Ce procédé est utile dans le cas $\lambda_1 \cong -\lambda_2$; après changement d'origine, $\lambda'_1 = \lambda_1 - s$ pourra être très différent en module de $\lambda'_2 = \lambda_2 - s$.

10.2.3. Puissance n-ième de l'inverse

Nous pouvons, en principe, obtenir une approximation de la plus petite valeur propre de $\boldsymbol{A}$, en module, en multipliant n fois le vecteur $\boldsymbol{x}^{(0)}$ par la matrice $\boldsymbol{A}^{-1}$. En effet, soit $\boldsymbol{u}$ l'un des vecteurs propres de $\boldsymbol{A}$; il obéit à la relation $\boldsymbol{A}\boldsymbol{u} = \lambda\boldsymbol{u}$, ou encore $\boldsymbol{u} = \lambda\boldsymbol{A}^{-1}\boldsymbol{u}$, ce qui montre que $1/\lambda$ est valeur propre de $\boldsymbol{A}^{-1}$. Dans la pratique, on ne calcule pas $\boldsymbol{x}^{(n)} = \boldsymbol{A}^{-1}\boldsymbol{x}^{(n-1)}$, mais on résout le système linéaire $\boldsymbol{A}\boldsymbol{x}^{(n)} = \boldsymbol{x}^{(n-1)}$, ce que l'on peut faire rapidement en formant une fois pour toute la décomposition $\boldsymbol{LU}$ de $\boldsymbol{A}$. Cette méthode a pour mérite principal de servir d'introduction à l'algorithme suivant, plus utile.

10.2.4. Puissance n-ième de l'inverse avec décalage

Considérons la matrice $\boldsymbol{B} = (\boldsymbol{A} - s\boldsymbol{I})^{-1}$; quels sont ses éléments propres ? Si s ne coïncide pas avec l'une des valeurs propres de $\boldsymbol{A}$, $\boldsymbol{B}$ est régulière et commute avec $\boldsymbol{A}$. Elle admet donc les mêmes vecteurs propres. En combinant les deux résultats précédents (éléments propres de $\boldsymbol{A} - s\boldsymbol{I}$ et de $\boldsymbol{A}^{-1}$), nous voyons que $1/(\lambda - s)$ est valeur propre de $\boldsymbol{B}$. Étant donné un vecteur initial $\boldsymbol{x}^{(0)}$, nous formons par récurrence :

$$\boldsymbol{x}^{(k)} = \boldsymbol{B}\boldsymbol{x}^{(k-1)} = \boldsymbol{B}^k\boldsymbol{x}^{(0)} = (\boldsymbol{A} - s\boldsymbol{I})^{-k}\boldsymbol{x}^{(0)}. \tag{10.7}$$

Développons $\boldsymbol{x}^{(0)}$ selon les vecteurs propres de $\boldsymbol{A}$, en appelant λ_m la valeur propre la plus proche de s :

$$\boldsymbol{x}^{(k)} = \sum_i \frac{c_i}{(\lambda_i - s)^k}\boldsymbol{u}_i = \frac{c_m}{(\lambda_m - s)^k}\left[\boldsymbol{u}_m + \sum_{i\neq m} c_i \left(\frac{\lambda_m - s}{\lambda_i - s}\right)^k \boldsymbol{u}_i\right]$$

Comme $|\lambda_m - s| < |\lambda_i - s|$, le vecteur $\boldsymbol{x}^{(k)}$ converge vers un vecteur proportionnel à $\boldsymbol{u}_m$. Ici encore, il ne faut pas calculer les inverses des matrices, mais résoudre des systèmes linéaires, comme $(\boldsymbol{A} - s\boldsymbol{I})\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)}$.

Cet algorithme peut nous permettre de calculer toutes les valeurs propres de $\boldsymbol{A}$. Il suffit de balayer, à l'aide du paramètre s, l'intervalle où doivent se trouver les valeurs

propres. Pour chaque valeur de s, l'itération sur $\boldsymbol{x}$ converge vers le vecteur propre appartenant à la valeur propre la plus proche de s ; par ailleurs, le rapport de deux coordonnées permet le calcul de cette valeur propre : $x_j^{(k)}/x_j^{(k-1)} = 1/(\lambda_j - s)$.

En fait, la méthode s'avère peu pratique pour la recherche systématique des valeurs propres ; elle est au contraire très commode pour déterminer les vecteurs propres, connaissant des valeurs approchées des λ_i. Supposons en effet qu'une autre méthode, différente, nous ait fourni un nombre λ', valeur approchée au millième de λ_m. Si nous choisissons $s = \lambda'$, l'algorithme précédent va converger en quelques itérations vers le vecteur propre associé à λ_m, à condition que toutes les autres valeurs propres diffèrent de λ_m par plus de quelques millièmes.

Exemple – La matrice $\boldsymbol{F}$ possède une valeur propre voisine de 1,5 ; cherchons le vecteur propre correspondant. La partie principale du programme ressemble à celle-ci

```
E = eye(F);                                          1
s = input("valeur du décalage: ");                   2
F1 = F - s*E                                         3
x1 = F1\x0; x1 = x1/norm(x1);                        4
```

et voici le résultat de quatre itérations avec $s = 1,5$

```
x0        x1           x2           x3           x4
1   - 0.4472136    0.4447160  - 0.4442792    0.4442812
0   - 0.5366563    0.5621210  - 0.5621030    0.5621095
0     0.7155418  - 0.6973146    0.6976075  - 0.6976010
```

Vous constatez que le vecteur propre associé à la valeur propre 1,48012 est obtenu avec une très bonne précision.

10.2.5. Quotient de Rayleigh

Étant donné une matrice $\boldsymbol{A}$ symétrique d'ordre n et un vecteur donné $\boldsymbol{x}$, le nombre

$$R(\boldsymbol{x}) \equiv \frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \tag{10.8}$$

est appelée le quotient de Rayleigh de $\boldsymbol{x}$. Vous voyez que si $\boldsymbol{x}$ coïncide avec un vecteur propre de $\boldsymbol{A}$, disons $\boldsymbol{u}_k$, alors R est égal à la valeur propre correspondante, soit λ_k. Montrons que, quel que soit $\boldsymbol{x}$, la quantité

$$D^2 \equiv \| \, (\boldsymbol{A} - \lambda \boldsymbol{I}) \boldsymbol{x} \, \|^2$$

est minimale quand $\lambda = R$. En développant, nous trouvons

$$D^2 = \lambda^2 \boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{x} - 2\lambda \boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{A} \boldsymbol{x} + \boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{A} \boldsymbol{x}.$$

Lorsque λ varie, D^2 présente bien le minimum annoncé. En conséquence, si $\boldsymbol{x}$ est proche d'un vecteur propre de $\boldsymbol{A}$, $R(\boldsymbol{x})$ est une bonne approximation de la valeur propre associée. Nous pouvons alors combiner cette propriété avec l'algorithme de la puissance n-ième de l'inverse avec décalage. Le fragment de programme du paragraphe précédent devient

```
E = eye(F);
x0 = [1,0,0]';
s0 = x0'*F*x0;
F1 = F - s0*E
x1 = F1\x0; x1 = x1/norm(x1);
s1 = x1'*F*x1;
F2 = F - s1*E;
```

Les premières itérations donnent

$s0$	$s1$	$s2$	$s3$	$s4$
1	1.056338	1.3666975	1.4793132	1.4801214

La convergence est rapide, bien que l'approximation de départ soit médiocre.

10.3. Méthode de Jacobi

Contrairement aux algorithmes précédents, la méthode de Jacobi permet le calcul de tous les éléments propres simultanément. Il s'agit d'effectuer une série de similitudes qui vont amener la matrice carrée symétrique $\boldsymbol{A}$ progressivement à la forme diagonale, tout en préservant ses valeurs propres.

10.3.1. Principe

Le principe de la méthode est facile à comprendre sur le cas à deux dimensions, et c'est par là que nous commençons. Soit la matrice :

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix}$$

que nous souhaitons diagonaliser par une similitude $\boldsymbol{A'} = \boldsymbol{R}^T\boldsymbol{A}\boldsymbol{R}$, où $\boldsymbol{R}$ est une matrice orthogonale qui s'écrit :

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}.$$

Un calcul élémentaire montre que :

$$\boldsymbol{A'} = \begin{bmatrix} a_{11}c^2 - 2a_{12}sc + a_{22}s^2 & a_{12}(c^2 - s^2) + (a_{11} - a_{22})cs \\ a_{12}(c^2 - s^2) + (a_{11} - a_{22})cs & a_{11}s^2 + 2a_{12}cs + a_{22}c^2 \end{bmatrix}.$$

Nous rendons $\boldsymbol{A'}$ diagonale en choisissant θ selon la relation :

$$\frac{cs}{c^2 - s^2} = \frac{a_{12}}{a_{22} - a_{11}}$$

ou encore

$$\operatorname{tg} 2\theta = \frac{2a_{12}}{a_{22} - a_{11}}.$$

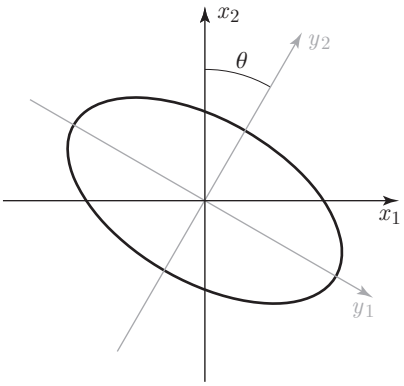


Figure 10.2 – Rotation des axes pour « ramener l'ellipse à ses axes principaux ».

Si $a_{22} - a_{11} = 0$, on choisit $\theta = \pi/4$.

Ces petits calculs admettent une interprétation géométrique simple. Soit $\boldsymbol{x} = (x_1, x_2)$ un vecteur de $\mathbb{R}^2$. Á la matrice $\boldsymbol{A}$, nous associons la forme quadratique $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ et l'équation :

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = 1 = a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2$$

qui représente une conique (ellipse, parabole ou hyperbole). Nous supposons, pour fixer les idées, que cette courbe est une ellipse (il faut pour cela que $\det \boldsymbol{A} > 0$). Effectuons le changement de repère défini par :

$$\boldsymbol{x} = \boldsymbol{R}\boldsymbol{y}.$$

Il s'agit d'une rotation des axes d'angle θ. Effectuons le changement de variables dans l'équation de l'ellipse :

$$(\boldsymbol{R}\boldsymbol{y})^T\boldsymbol{A}\boldsymbol{R}\boldsymbol{y} = \boldsymbol{y}\boldsymbol{R}^T\boldsymbol{A}\boldsymbol{R}\boldsymbol{y} = \boldsymbol{y}\boldsymbol{A}'\boldsymbol{y} = 1$$

La nouvelle équation de la conique est définie par la nouvelle matrice. Si $\boldsymbol{A}'$ est diagonale parce que nous avons fait le bon choix pour θ, l'équation s'écrit :

$$a'_{11}y_1^2 + a'_{22}y_2^2 = 1.$$

On dit que la conique a été ramenée à ses « axes principaux » ; les axes principaux sont aussi les axes de symétrie de l'ellipse. Les termes « croisés » ou « rectangles » (en x_1x_2) ont disparu. La généralisation de ce procédé à n dimensions constitue l'algorithme de Jacobi.

10.3.2. Mise en œuvre

Nous allons appliquer à la matrice $\boldsymbol{A}$, symétrique, réelle, une suite de similitudes $\boldsymbol{A} \equiv \boldsymbol{A}^{(0)} \to \boldsymbol{A}^{(1)} \to \boldsymbol{A}^{(2)} \to \cdots \to \boldsymbol{A}^{(n)}$, avec $\boldsymbol{A}^{(i)} = \boldsymbol{R}^{(i)T}\boldsymbol{A}^{(i-1)}\boldsymbol{R}^{(i)}$ de telle manière que $\boldsymbol{A}^{(n)}$ tende vers une matrice diagonale $\boldsymbol{D}$ d'éléments diagonaux λ_i. Il est facile de comprendre pourquoi la diagonalisation n'est qu'approchée. A l'aide d'une

rotation des axes du plan (x_1, x_2), nous pouvons annuler l'élément a_{12} comme au paragraphe précédent. Mais, lorsque nous appliquons une nouvelle similitude pour faire disparaître a_{13}, tous les éléments de la ligne 1, en particulier le zéro que nous venons de créer en a_{12}, sont modifiés. Ce n'est que progressivement que la "substance" de la matrice va se concentrer sur la diagonale.

Le passage de $\boldsymbol{A}^{(i)}$ (éléments notés a_{rs} pour alléger l'écriture) à $\boldsymbol{A}^{(i+1)}$ (d'éléments notés a'_{rs}) fait intervenir la matrice orthogonale $\boldsymbol{R}_{jk}$ qui vaut (en posant $C = \cos\theta_{jk}$ et $S = \sin\theta_{jk}$) :

$$\boldsymbol{R}_{j,k} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & & & & \cdots & & 0 \\ 0 & & 1 & & & & & \vdots \\ \vdots & & & C & & S & & \vdots \\ \vdots & & & & 1 & & & \vdots \\ \vdots & & & -S & & C & & \vdots \\ 0 & & & & & & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 \end{bmatrix}.$$

$\boldsymbol{R}_{jk}$ représente une rotation des axes dans le sous-espace (j, k). On emploie souvent les termes de « rotation de Jacobi » ou de « rotation de Givens » pour désigner cette transformation. Seules les colonnes j et k et les lignes j et k de $\boldsymbol{A}^{(i+1)}$ diffèrent de celles de $\boldsymbol{A}^{(i)}$. En calculant d'abord $\boldsymbol{AR}$, puis $\boldsymbol{R}^T\boldsymbol{AR}$, nous obtenons les équations de transformation suivantes

$$\begin{aligned} &a'_{rs} = a_{rs}, \quad r, s \neq j, k, \\ &a'_{rj} = a'_{jr} = Ca_{rj} - Sa_{rk}, \quad r \neq j, k, \\ &a'_{rk} = a'_{kr} = Sa_{rj} + Ca_{rk}, \quad r \neq j, k, \\ &a'_{jj} = C^2 a_{jj} + S^2 a_{kk} - 2CSa_{jk}, \\ &a'_{jk} = a'_{kj} = CS(a_{jj} - a_{kk}) + (C^2 - S^2)a_{jk}, \\ &a'_{kk} = S^2 a_{jj} + C^2 a_{kk} + 2CSa_{jk}. \end{aligned} \tag{10.9}$$

L'angle de rotation est défini par la condition $a'_{jk} = 0$ soit

$$\operatorname{tg} 2\theta_{jk} = \frac{2a_{jk}}{a_{kk} - a_{jj}}. \tag{10.10}$$

L'indétermination sur θ est levée par les conditions

$$a_{kk} \neq a_{jj} : \quad |\theta| \leq \frac{\pi}{4} \quad ; \quad a_{kk} = a_{jj} : \quad \theta = \frac{\pi}{4}.$$

En pratique, on effectue les calculs sur une moitié de la matrice, puisque la similitude respecte la symétrie. On ne calcule pas non plus a'_{jk}, dont on sait qu'il doit être nul. Enfin, on gagne beaucoup de temps en ne calculant pas explicitement l'angle θ.

Nous pouvons donner une indication sur la convergence de la méthode. Définissons la norme de Frobenius de $\boldsymbol{A}$:

$$\| \boldsymbol{A} \|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2$$

(et de même pour $\boldsymbol{A'}$), puis les « normes extradiagonales » des mêmes matrices par les relations :

$$S^2 = \sum_{i \neq j}^{n} |a_{ij}|^2; \qquad S'^2 = \sum_{i \neq j}^{n} |a'_{ij}|^2.$$

Nous admettrons qu'une transformation orthogonale générale, telle que $\boldsymbol{B} = \boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{Q}$ avec $\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}$, préserve la norme de Frobenius, $\| \boldsymbol{B} \| = \| \boldsymbol{A} \|$. Le calcul explicite de ces normes dans le cas des matrices $\boldsymbol{A}$ et $\boldsymbol{A'}$, en utilisant les formules (10.9), montre que

$$a_{jj}^2 + a_{kk}^2 + 2a_{jk}^2 = {a'_{jj}}^2 + {a'_{kk}}^2 + 2{a'_{jk}}^2 = {a'_{jj}}^2 + {a'_{kk}}^2, \tag{10.11}$$

puisque a'_{jk} est nul. D'après la définition des normes, nous savons que

$$\| \boldsymbol{A'} \|^2 = S'^2 + \sum_{i=1}^{n} {a'_{ii}}^2 \text{ et } \| \boldsymbol{A} \|^2 = S^2 + \sum_{i=1}^{n} {a_{ii}}^2.$$

Les deux sommes sur i contiennent des termes identiques, sauf pour $i = j$ ou $i = k$. En simplifiant, il vient

$$S'^2 + a'^2_{jj} + a'^2_{kk} = S^2 + a^2_{jj} + a^2_{kk}$$

et, d'après (10.11), :

$$S'^2 = S^2 - 2a^2_{jk}. \tag{10.12}$$

Cette équation prouve que $S' < S$: chaque similitude (passage de $(\boldsymbol{A}^{(i)})$ à $(\boldsymbol{A}^{(i+1)})$) diminue la norme extradiagonale. La suite des S (qui correspondent à la suite des $(\boldsymbol{A}^{(i)})$), positive décroissante, converge.

Dans le cas de l'algorithme dit de « Jacobi classique », on choisit les indices j et k tels que $|a_{jk}|$ soit maximal. Nous pouvons alors affirmer que $S^2 \leq n(n-1)a^2_{jk}$, d'où

$$S'^2 \leq \left(1 - \frac{2}{n(n-1)}\right) S^2.$$

En itérant cette relation entre les étapes 1 et i, nous trouvons que

$$S^{(i)^2} \leq \left(1 - \frac{2}{n(n-1)}\right)^i S^{(0)^2}$$

ce qui représente une convergence linéaire vers zéro.

L'algorithme « classique » que nous venons de présenter est assez lent à cause de la recherche du plus grand élément extradiagonal avant chaque rotation de Jacobi. Il se trouve que l'algorithme converge aussi si les indices j, k sont choisis systématiquement,

dans l'ordre « lexicographique », (1,2), (1,3), ..., (1,n), (2,3), (2,4), ..., même si la démonstration, plus difficile, n'est pas abordée ici.

Lorsque la convergence est atteinte, la matrice est réputée diagonale et $\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{D}$, avec $\boldsymbol{R}^{(1)}\boldsymbol{R}^{(2)}\cdots\boldsymbol{R}^{(n)} = \boldsymbol{Q}$. Les colonnes de $\boldsymbol{Q}$ sont les vecteurs propres et nous intéressent. Il serait ruineux de conserver les matrices $\boldsymbol{R}^{(i)}$ pour les multiplier à la fin. On procède en fait par récurrence, en posant

$$\boldsymbol{Q}^{(i)} = \boldsymbol{R}^{(1)}\boldsymbol{R}^{(2)}\cdots\boldsymbol{R}^{(i)}, i \leq n$$

et

$$\boldsymbol{Q}^{(i+1)} = \boldsymbol{Q}^{(i)}\boldsymbol{R}^{(i+1)}.$$

Chaque fois que nous définissons les éléments d'une matrice $\boldsymbol{R}$, nous en profitons pour former la matrice $\boldsymbol{Q}^{(i+1)}$ (éléments q'_{rs}) à partir de $\boldsymbol{Q}^{(i)}$ (éléments q_{rs}) :

$$\begin{aligned} q'_{rs} &= q_{rs}, \quad r,s \neq j,k, \\ q'_{rj} &= cq_{rj} + sq_{rk}, \quad r \neq j,k, \\ q'_{rk} &= -sq_{rj} + cq_{rk}, \quad r \neq j,k; \end{aligned}$$

Dès que l'ordre de la matrice dépasse une dizaine, la méthode de Jacobi est moins rapide que la réduction de la matrice à la forme tridiagonale, suivie d'un calcul de valeurs propres. Cette remarque est d'autant plus pertinente que $\boldsymbol{A}$ a la forme d'une matrice bande plus ramassée : la méthode de Jacobi détruit cette structure.

Exemple – Nous nous proposons de diagonaliser la matrice des exemples précédents.

$$\boldsymbol{F} = \boldsymbol{F}^{(0)} = \begin{bmatrix} 1.0 & 1. & 0.5 \\ 1.0 & 1. & 0.25 \\ 0.5 & 0.25 & 2.0 \end{bmatrix}$$

Nous calculons d'abord le carré de la norme complète `-->sum(F.*F) = 8.625` et le carré de la norme diagonale de $\boldsymbol{F}$: `-->trace(F.*F) = 6.0` Pour annuler l'élément $f_{12}^{(0)}$, il faut effectuer une première rotation dans le sous-espace (1,2). Comme $f_{22}^{(0)} = f_{11}^{(0)}$, nous avons $2\theta = \pi/2, \theta = \pi/4$, si bien que

$$\boldsymbol{R}^{(1)} = \begin{bmatrix} .7071068 & .7071068 & 0. \\ -.7071068 & .7071068 & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

et

$$\boldsymbol{F}^{(1)} = \boldsymbol{R}^{(1)T}\boldsymbol{F}\boldsymbol{R}^{(1)} = \begin{bmatrix} 0. & 0. & .1767767 \\ 0. & 2. & .5303301 \\ .1767767 & .5303301 & 2. \end{bmatrix}$$

La norme au carré reste constante, `sum(F1.*F1) = 8.625`, alors que la somme des carrés des éléments diagonaux augmente : `trace(F1.*F1) = 8.` La deuxième rotation, destinée à annuler le plus grand élément extradiagonal, soit $f_{23}^{(1)}$, est encore

particulière, avec $\theta = \pi/4$ et

$$\boldsymbol{R}^{(2)} = \begin{bmatrix} 1. & 0. & 0. \\ 0. & .7071068 & .7071068 \\ 0. & -.7071068 & .7071068 \end{bmatrix},$$

$$\boldsymbol{F}^{(2)} = \boldsymbol{R}^{(2)T}\boldsymbol{F}^{(1)}\boldsymbol{R}^{(2)} = \begin{bmatrix} 0. & -.125 & .125 \\ -.125 & 1.4696699 & 2.220E-16 \\ .125 & 2.220E-16 & 2.5303301 \end{bmatrix}$$

Les tous petits nombres $f_{23}^{(2)} = f_{32}^{(2)}$ sont dus à des erreurs d'arrondi ; faisons le ménage pour les faire disparaître

```
F2 = clean(F2) = !   0.      -  .125          .125     !
                 ! -  .125     1.4696699     0.        !
                 !    .125     0.            2.5303301 !
```

Nous surveillons le carré de la norme complète, `sum(F2.*F2) = 8.625` et le carré de la norme diagonale, `trace(F2.*F2) = 8.5625`. Vous remarquez que les éléments précédemment annulés ont tendance à « repousser », bien que moins vigoureusement.

La troisième rotation, dans le sous-espace (1,2), vise à éliminer $f_{12}^{(2)}$. Nous trouvons $\operatorname{tg} 2\theta = 2f_{12}^{(2)}/(f_{22}^{(2)} - f_{11}^{(2)}) = -0.1701062$, d'où la matrice de rotation

$$\boldsymbol{R}^{(3)} = \begin{bmatrix} .9964533 & -.0841471 & 0. \\ .0841471 & .9964533 & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

et la matrice $\boldsymbol{F}^{(3)}$, débarrassée d'éléments inférieurs à 10^{-16} :

$$\boldsymbol{F}^{(3)} = \begin{bmatrix} -.0105558 & 0. & .1245567 \\ 0. & 1.4802257 & -.0105184 \\ .1245567 & -.0105184 & 2.5303301 \end{bmatrix}$$

Les éléments diagonaux se renforcent : `trace(F3.*F3) = 8.59375`.

La quatrième rotation aura lieu dans le sous-espace (1,3), avec $\operatorname{tg} 2\theta = 0.0980419$ ou $\theta = 0.0488648$ rd. La matrice de rotation s'écrit

$$\boldsymbol{R}^{(4)} = \begin{bmatrix} .9988064 & 0. & .0488453 \\ 0. & 1. & 0. \\ -.0488453 & 0. & .9988064 \end{bmatrix}$$

et permet d'obtenir

$$\boldsymbol{F}^{(4)} = \begin{bmatrix} -.0166471 & .0005138 & 0. \\ .0005138 & 1.4802257 & -.0105058 \\ 0. & -.0105058 & 2.5364214 \end{bmatrix}$$

Le critère de convergence est atteint : nous arrêtons l'itération. Le carré de la norme diagonale vaut maintenant 8.6247788 (à $3\,10^{-4}$ de la norme de Frobenius).

Les vecteurs propres sont les colonnes de la matrice de similitude globale ; comme nous n'avons à manipuler que quatre petites matrices, nous formons $\boldsymbol{Q}$ directement

$$\boldsymbol{R} = \boldsymbol{R}^{(1)}\boldsymbol{R}^{(2)}\boldsymbol{R}^{(3)}\boldsymbol{R}^{(4)} = \begin{bmatrix} .7213585 & .4387257 & .5358747 \\ -.6861572 & .5577276 & .4670419 \\ -.0939688 & -.7045989 & .7033564 \end{bmatrix}$$

Cette matrice est orthogonale aux erreurs d'arrondi près, comme vous le vérifierez en formant $\boldsymbol{R}^T\boldsymbol{R}$. Nous vous recommandons aussi d'effectuer le produit de $\boldsymbol{F}$ par une colonne de $\boldsymbol{R}$, pour vérifier que l'on obtient bien un vecteur proportionnel à cette colonne et à la valeur propre correspondante.

L'erreur encourue sur les valeurs propres après quatre itérations de l'algorithme de Jacobi n'excède pas 10^{-4}. Les vecteurs propres sont nettement moins précis. C'est toujours le cas pour l'algorithme de Jacobi.

10.4. Transformation de Householder

L'algorithme de Householder se comprend facilement à partir de sa représentation géométrique dans l'espace à deux ou trois dimensions. Soient un point M et un plan (P), contenant l'origine (mais pas M) et normal au vecteur unitaire $\hat{\boldsymbol{n}}$ (cf. figure 10.3). Nous cherchons à construire le point M' qui se déduit de M par une symétrie orthogonale par rapport à (P). La droite passant par M et perpendiculaire à (P) perce ce plan en K ; elle passe aussi par M'. Le vecteur $\overrightarrow{KM}$ est parallèle à $\hat{\boldsymbol{n}}$; sa longueur est celle de la projection orthogonale de $\overrightarrow{OM}$ sur $\hat{\boldsymbol{n}}$; autrement dit, $\overrightarrow{KM} = (\hat{\boldsymbol{n}}\cdot\overrightarrow{OM})\hat{\boldsymbol{n}}$. Par symétrie, $\overrightarrow{KM'}$ est l'opposé de $\overrightarrow{KM}$. Nous pouvons relier M' à M par $\overrightarrow{OM'} = \overrightarrow{OM} + \overrightarrow{MM'} = \overrightarrow{OM} - 2\overrightarrow{KM} = \overrightarrow{OM} - 2(\hat{\boldsymbol{n}}\cdot\overrightarrow{OM})\hat{\boldsymbol{n}}$. Il est clair que $OM' = OM$, la symétrie préservant les longueurs.

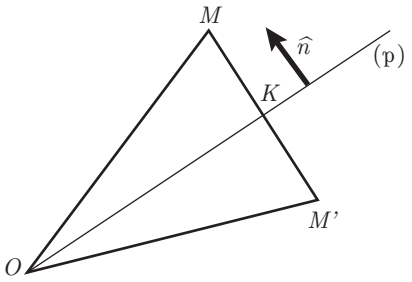


Figure 10.3 – Construction du symétrique d'un point par rapport à un plan. Le plan (P) coupe le plan de figure selon la droite (p).

Lorsque nous représentons les vecteurs par des matrices colonne, les relations précédentes deviennent $\overrightarrow{OM} = \boldsymbol{x}$, $\overrightarrow{OM'} = \boldsymbol{x'}$ et $\boldsymbol{x'} = \boldsymbol{x} - 2(\hat{\boldsymbol{n}}^T\boldsymbol{x})\hat{\boldsymbol{n}} = \boldsymbol{x} - 2\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T\boldsymbol{x} = (I - 2\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T)\boldsymbol{x}$, les dernières formes étant rendues possibles par l'associativité des produits.

Ce raisonnement se généralise sans peine à l'espace à n dimensions, même si la représentation géométrique devient plus difficile. Soit $\boldsymbol{v}$ un vecteur unitaire de $\mathbb{R}^n$: $\boldsymbol{v}^T\boldsymbol{v} = 1$. Nous définissons une matrice :

$$\boldsymbol{P} = \boldsymbol{I} - 2\boldsymbol{v}\boldsymbol{v}^T. \tag{10.13}$$

Cette matrice est symétrique : $\boldsymbol{P}^T = \boldsymbol{P}$ parce que le transposé d'un produit est égal au produit des transposées dans l'ordre inverse. Elle est aussi orthogonale :

$$\begin{aligned}\boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{P}\boldsymbol{P} &= (\boldsymbol{I} - 2\boldsymbol{v}\boldsymbol{v}^T)(\boldsymbol{I} - 2\boldsymbol{v}\boldsymbol{v}^T)\\ &= \boldsymbol{I} - 2\boldsymbol{v}\boldsymbol{v}^T - 2\boldsymbol{v}\boldsymbol{v}^T + 4\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{v}\boldsymbol{v}^T = \boldsymbol{I}\end{aligned}$$

et donc idempotente $\boldsymbol{P}^2 = \boldsymbol{I}$. Si $\boldsymbol{y}$ se déduit de $\boldsymbol{x}$ par la relation :

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x} = \boldsymbol{x} - 2\boldsymbol{v}(\boldsymbol{v}^T\boldsymbol{x}),$$

alors :

$$\boldsymbol{y}^T\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{x},$$

les vecteurs symétriques ont même longueur.

Nous nous intéressons maintenant à une réciproque du cas précédent : nous souhaitons déterminer un vecteur $\boldsymbol{v}$ (et donc une matrice $\boldsymbol{P}$) tel qu'un $\boldsymbol{x}$ donné soit transformé par $\boldsymbol{P}$ en un multiple du premier vecteur de la base canonique

$$\boldsymbol{P}\boldsymbol{x} = k\boldsymbol{e}_1$$

ou, de façon imagée,

$$\begin{bmatrix} \times \\ \times \\ \times \\ \times \end{bmatrix} \longrightarrow \begin{bmatrix} k \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

D'après la conservation de la norme, $|k|^2 = \| \boldsymbol{x} \|^2 = \boldsymbol{x}^T\boldsymbol{x}$ et $k = \pm \| \boldsymbol{x} \|$.

Le principe du calcul est représenté fig. 10.4, dans le cas à deux dimensions. Le point M_+, symétrique de M par rapport à p_+, est à l'abscisse $\| x \|$, le point M_-, symétrique de M par rapport à p_-, est à l'abscisse $- \| x \|$. On montre que, pour diminuer les erreurs d'arrondi, il faut choisir, pour k, le signe correspondant au plus long des vecteurs $\overrightarrow{MM_+}$ et $\overrightarrow{MM_-}$, soit, ici, le signe moins (M et M_- de part et d'autre de l'axe vertical). On écrit dans le cas général

$$k = -\text{sign}(x_1) \| \boldsymbol{x} \| .$$

D'après les relations précédentes :

$$\boldsymbol{x} - k\boldsymbol{e}_1 = \boldsymbol{x} - \boldsymbol{P}\boldsymbol{x} = 2(\boldsymbol{v}^T\boldsymbol{x})\boldsymbol{v},$$

ce qui montre que le vecteur $\boldsymbol{v}$ est proportionnel au vecteur $\boldsymbol{x} - k\boldsymbol{e}_1$; nous pouvons donc l'écrire

$$\boldsymbol{v} = \frac{\boldsymbol{x} + \text{sign}(x_1) \| \boldsymbol{x} \| \boldsymbol{e}_1}{\| \boldsymbol{x} + \text{sign}(x_1) \| \boldsymbol{x} \| \boldsymbol{e}_1 \|}. \tag{10.14}$$

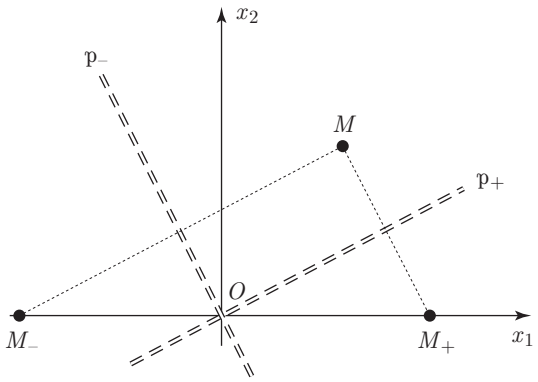


Figure 10.4 – Construction des points symétriques de M sur l'axe 1.

Le fragment de programme ci-dessous exécute ces calculs.

Listing 10.1 – Construction de la matrice de Householder

```
//...lecture de N et x...                                    1
u = zeros(x);                                                2
e1 = zeros(x);                                               3
e1(1) = 1;                                                   4
IN = eye(N,N);                                               5
v = sign(x(1))*norm(x)*e1 + x;                               6
v = v/norm(v);                                               7
P = IN - 2*v*v'                                              8
P*x                                                          9
```

Les lignes 6,7 et 8 traduisent les équations (10.13) et (10.14).

Exemple – Soit le vecteur $\boldsymbol{x} = [1, -2, 3, 1, -1]^T$ dont la norme est 4. Le vecteur $\boldsymbol{x} - k\boldsymbol{e}_1$ vaut $[5, -2, 3, 1, -1]^T$. On trouve alors :

```
P  =   - 0.25    0.5  - 0.75  - 0.25    0.25
         0.5     0.8    0.3     0.1   - 0.1
       - 0.75    0.3    0.55  - 0.15    0.15
       - 0.25    0.1  - 0.15    0.95    0.05
         0.25  - 0.1    0.15    0.05    0.95
```

et on vérifie que $\boldsymbol{Px} = [-4, 0, 0, 0, 0]^T$. Vous pouvez changer le signe de x_1 et observer que le programme construit le « bon » symétrique.

Remarque : On peut aussi écrire, sans normaliser le vecteur $\boldsymbol{v}$ et quel que soit le signe de x_1 :

$$\boldsymbol{P} = \boldsymbol{I} - \beta \boldsymbol{u}\boldsymbol{u}^T$$

avec : $\boldsymbol{u} = \boldsymbol{x} + \parallel x \parallel \boldsymbol{e}_1 = [x_1 + \parallel x \parallel, x_2, \ldots, x_n]^T, \beta = 1/ \parallel x \parallel (\parallel x \parallel + x_1)$.

10.5. Factorisation QR et algorithme QR

10.5.1. Factorisation QR

Nous allons utiliser les matrices de Householder pour former une nouvelle factorisation de la matrice $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$$

où $\boldsymbol{Q}$ est une matrice orthogonale et $\boldsymbol{R}$ une matrice triangulaire supérieure ($R_{i,j} = 0$ si $i > j$). Pour simplifier l'exposé, nous décrivons cet algorithme dans le cas d'une matrice $\boldsymbol{A}$ carrée $n \times n$ régulière, une hypothèse qu'il est facile de lever.

Soit $\boldsymbol{a}_1$ la première colonne de $\boldsymbol{A}$. Nous venons de voir comment former une matrice $\boldsymbol{Q}^{(1)}$ telle que $\boldsymbol{Q}^{(1)}\boldsymbol{a}_1 = k\boldsymbol{e}_1$. Posons $\boldsymbol{A}^{(1)} = \boldsymbol{Q}^{(1)}\boldsymbol{A}$; la première colonne de cette matrice comporte $n-1$ zéros. Nous créons maintenant $\boldsymbol{P}^{(2)}$ $(n-1 \times n-1)$ que nous insérons dans une matrice $n \times n$ $\boldsymbol{Q}^{(2)}$ comme ceci

$$\boldsymbol{Q}^{(2)} = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \boldsymbol{P}^{(2)} \end{array}\right]. \tag{10.15}$$

Cette matrice a pour mission de créer $n-2$ zéros dans la deuxième colonne du produit $\boldsymbol{A}^{(2)} = \boldsymbol{Q}^{(2)}\boldsymbol{A}^{(1)}$. Étant donné la structure particulière en blocs de $\boldsymbol{Q}^{(2)}$, les zéros de la première colonne de $\boldsymbol{A}^{(1)}$ se retrouvent dans $\boldsymbol{A}^{(2)}$. Ce procédé est itéré jusqu'à l'avant-dernière colonne de $\boldsymbol{A}$. La matrice $\boldsymbol{A}^{(n-1)} = \boldsymbol{Q}^{(n-1)}\boldsymbol{A}^{(n-2)}$ est alors triangulaire supérieure.

L'ensemble du calcul peut être résumé comme suit

$$\boldsymbol{R} \equiv \boldsymbol{A}_{n-1} = \boldsymbol{Q}^{(n-1)} \cdots \boldsymbol{Q}^{(2)}\boldsymbol{Q}^{(1)}\boldsymbol{A}.$$

Posons

$$\boldsymbol{Q}^T \equiv \boldsymbol{Q}^{(n-1)} \cdots \boldsymbol{Q}^{(2)}\boldsymbol{Q}^{(1)} \tag{10.16}$$

La matrice $\boldsymbol{Q}^T$ est aussi orthogonale et son inverse est égal à sa transposée, laquelle vaut

$$\boldsymbol{Q} = \boldsymbol{Q}^{(1)T}\boldsymbol{Q}^{(2)T} \cdots \boldsymbol{Q}^{(n-1)T} = \boldsymbol{Q}^{(1)}\boldsymbol{Q}^{(2)} \cdots \boldsymbol{Q}^{(n-1)}, \tag{10.17}$$

car tous les facteurs sont symétriques. Nous avons ainsi obtenu une décomposition de $\boldsymbol{A}$ en un produit de deux matrices, $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$, la première orthogonale, la deuxième triangulaire supérieure.

Dans beaucoup d'applications, il n'est pas nécessaire de connaître explicitement la matrice $\boldsymbol{Q}$; il suffit de savoir calculer des produits tels que $\boldsymbol{Q}\boldsymbol{y}$ ou $\boldsymbol{Q}^T\boldsymbol{y}$ ($\boldsymbol{y}$ est un vecteur quelconque). Ces évaluations se font par itération, en utilisant les définitions (10.13) et (10.15) ainsi que (10.16) ou (10.17). Il faut bien sûr avoir pris soin de conserver les vecteurs $\boldsymbol{v}$ successifs. On peut de même former $\boldsymbol{Q}$ (ou $\boldsymbol{Q}^T$) par une itération :

$$\boldsymbol{Q} = \boldsymbol{I} \quad ; \quad \boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}_k, \quad k = 1, \ldots, n.$$

Le listing ci-contre montre un programme simple, mais peu économe en mémoire, qui réalise ces calculs.

Listing 10.2 – Factorisation QR

```
//  ...  lecture de N et de A  ...                                  1
for k = 1:N-1                                                       2
    x = A(k:N,k);                                                   3
    e1 = zeros(x);                                                  4
    e1(1) = 1;                                                      5
    v = sign(x(1))*norm(x)*e1+x;                                    6
    v = v/norm(v);                                                  7
    V(k:N,k) = v;                                                   8
    A(k:N,k:N) = A(k:N,k:N) - 2*v*(v'*A(k:N,k:N));                  9
end                                                                 10
for k = 1:N                                                         11
   Q = Q - 2*(Q*V(1:N,k))*V(1:N,k)';                                12
end                                                                 13
```

Dans la boucle sur k, la ligne 3 prélève les $N-k+1$ dernières composantes de la colonne k ; les lignes 4 à 7 forment le vecteur $\boldsymbol{v}_k$ comme précédemment ; celui-ci est rangé dans la matrice $\boldsymbol{V}$; le gros du travail est fait ligne 9 ; celle-ci « ronge » le coin inférieur gauche de $\boldsymbol{A}$, colonne par colonne. Enfin, une deuxième boucle forme la matrice $\boldsymbol{Q}$ à partir des vecteurs $\boldsymbol{v}_k$.

Pour économiser de la mémoire, on peut recouvrir progressivement le triangle supérieur de $\boldsymbol{A}$ par le triangle supérieur de $\boldsymbol{R}$ et ranger les éléments non nuls des vecteurs $\boldsymbol{v}_k$ dans le triangle inférieur de $\boldsymbol{A}$. Le nombre d'opérations est équivalent à $\frac{4}{3}n^3$.

Une première application de la factorisation QR concerne la résolution des systèmes linéaires. Pour résoudre $\boldsymbol{Ax} = \boldsymbol{b}$, connaissant la décomposition de $\boldsymbol{A}$, il suffit de résoudre successivement les systèmes $\boldsymbol{Qy} = \boldsymbol{b}$ puis $\boldsymbol{Rx} = \boldsymbol{y}$. Le solution du premier système implique la transposition de $\boldsymbol{Q}$, suivie d'un produit matrice × vecteur, celle du second est rapide puisqu'il s'agit d'un système triangulaire (voir le § 6.3.3). Ce formalisme est particulièrement bien adapté au cas des systèmes surdéterminés, comme ceux associés à la méthode des moindres carrés (§ 6.7 et 14.6), à condition de l'étendre aux matrices rectangulaires.

Le calcul des valeurs propres d'une matrice constitue une deuxième application de la factorisation QR ; c'est le sujet de la section suivante.

10.5.2. Algorithme QR

Cet algorithme est, dans sa version de base, d'une simplicité étonnante. Soit $\boldsymbol{A}$ une matrice réelle symétrique. L'algorithme QR consiste en l'itération

poser $\boldsymbol{A}^{(0)} = \boldsymbol{A}$, $\boldsymbol{S}^{(0)} = \boldsymbol{I}$;
pour $k = 1, 2, ...$
$\quad \boldsymbol{Q}^{(k)}\boldsymbol{R}^{(k)} = \boldsymbol{A}^{(k-1)}$; // factorisation
$\quad \boldsymbol{A}^{(k)} = \boldsymbol{R}^{(k)}\boldsymbol{Q}^{(k)}$; // produit dans l'ordre inverse
$\quad \boldsymbol{S}^{(k)} = \boldsymbol{S}^{(k-1)}\boldsymbol{Q}^{(k)}$;

Vous remarquez que chaque itérée $\boldsymbol{A}^{(k)}$ se déduit de la précédente par une similitude

$$\boldsymbol{A}^{(k)} = (\boldsymbol{Q}^{(k)})^{-1}\boldsymbol{A}^{(k-1)}\boldsymbol{Q}^{(k)} \tag{10.18}$$

qui préserve les valeurs propres. Les conditions de convergence s'énoncent ainsi :

> **Théorème** – Soit $\boldsymbol{A}$ une matrice réelle symétrique dont les valeurs propres sont toutes différentes en module et soit $\boldsymbol{S}$ la matrice formée par ses vecteurs propres. Si tous les sous-déterminants principaux de $\boldsymbol{S}$ sont non-nuls, alors $\boldsymbol{A}^{(k)}$ converge vers une matrice diagonale et $\boldsymbol{S}^{(k)}$ converge vers $\boldsymbol{S}$.

Vous pourrez lire la démonstration, assez compliquée, dans les ouvrages cités.

Exemple – Le listing ci-dessous vous montre comment utiliser la fonction `qr` de Scilab pour diagonaliser la matrice $\boldsymbol{F}$ définie au § 10.1.

Listing 10.3 – Diagonalisation par QR

```
F0 = [1,1,0.5;1,1,0.25;0.5,0.25,2]
F = F0; S = eye(F);
for i = 1:20
  [Q,R] = qr(F);
  F = R*Q;
  S = S*Q;
end
F,R,Q,S
```

Nous avons obtenu, après 20 itérations,

```
F  =   2.5365259  - 0.0000185  - 2.997D-16
     - 0.0000185    1.4801214  - 2.813D-17
       2.206D-43  - 1.169D-38  - 0.0166473

S  =   0.5314912    0.4442718  - 0.7212071
       0.4614832    0.5621013    0.6863493
       0.7103171  - 0.6976136    0.0937280
```

Les valeurs propres sont précises (six chiffres significatifs), bien qu'il subsiste dans $\boldsymbol{F}$ un petit élément extradiagonal. Le même calcul peut être conduit avec Maple et la fonction `QRdecomp`.

Chaque itération requiert un nombre d'opération équivalent à n^3 ; de plus, la convergence paraît lente. Vous pouvez à bon droit estimer que cet algorithme est peu performant. Deux améliorations le rendent très efficace. Il faut, d'une part, l'appliquer à une matrice tridiagonale (voir le paragraphe suivant). Vous pouvez vérifier facilement que cette structure est préservée par la similitude (10.18). Il faut, d'autre part, opérer des décalages, d'une manière analogue à ce que nous avons expliqué au § 10.2.2. Ce dernier point ne sera pas abordé ici.

10.6. Réduction à la forme tridiagonale et calcul des valeurs propres

Il est avantageux d'abandonner l'objectif de l'algorithme de Jacobi (diagonaliser une matrice symétrique par une suite convergente de similitudes) pour un objectif plus restreint : amener la matrice réelle symétrique $\boldsymbol{A}$ à la forme tridiagonale. En effet, la détermination des éléments propres d'une matrice tridiagonale est très rapide. D'autre part, la transformation de $\boldsymbol{A}$ en matrice tridiagonale nécessite, grâce à l'algorithme de Householder, un nombre fini d'opérations. Enfin, la réduction à la forme tridiagonale est une étape pratiquement obligée dans la mise en oeuvre de l'algorithme QR.

10.6.1. Tridiagonalisation

Nous allons utiliser des matrices de Householder pour amener la matrice réelle symétrique $\boldsymbol{A}$ à la forme tridiagonale, à l'aide d'une série de similitudes :

$$\boldsymbol{A}^{(i)} = \boldsymbol{Q}^{(i)T} \boldsymbol{A}^{(i-1)} \boldsymbol{Q}^{(i)}.$$

Le premier produit (prémultiplication, qui agit sur les colonnes), $\boldsymbol{Q}^T \boldsymbol{A}$, doit laisser invariant a_{11} et doit transformer le vecteur $[a_{21}, a_{31}, \ldots, a_{n1}]^T$ en $[k, 0, 0, \ldots]^T$, un multiple du premier vecteur de la base canonique de $\mathbb{R}^{n-1}$. Il faut donc faire agir une matrice de Householder $\boldsymbol{P^{(1)}}$ de dimension $n-1 \times n-1$; celle-ci sera insérée dans une matrice d'ordre n ayant la structure

$$\boldsymbol{Q}^{(1)} = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \boldsymbol{P^{(1)}} \end{array}\right]$$

Le deuxième produit $(\boldsymbol{Q}^{(1)T} \boldsymbol{A}) \boldsymbol{Q}^{(1)}$ (postmultiplication agissant sur les lignes) fera de même pour la première ligne de $\boldsymbol{A}$: a_{11} invariant et $[a_{12}, a_{13}, \ldots, a_{1n}]$ transformé en $[k, 0, \ldots, 0]$.

Vous imaginez sans peine la suite de l'itération : à l'aide d'une matrice de Householder $n-2 \times n-2$, nous formerons une nouvelle matrice $\boldsymbol{Q}^{(2)}$, d'ordre $n-1$, dont la mission sera d'annuler les éléments $a_{42}, \ldots, a_{n2}$ et $a_{24}, \ldots, a_{2n}$ (ce sont les mêmes). Au bout de $n-2$ similitudes, la matrice sera exactement sous forme tridiagonale : cet algorithme est plus compliqué que celui de Jacobi, mais il aboutit au résultat cherché en un nombre fini d'opérations. Le nombre d'opérations nécessaire est équivalent à $\frac{4}{3}n^3$. Il faudra, en principe, garder les matrices $\boldsymbol{Q}^{(i)}$ pour le calcul des vecteurs propres. Si $\boldsymbol{y}$ est un vecteur propre de $\boldsymbol{A}^{(n-2)}$, $\boldsymbol{x} = \boldsymbol{Q}^{(1)} \boldsymbol{Q}^{(2)} \cdots \boldsymbol{Q}^{(n-2)} \boldsymbol{y}$ est vecteur propre de $\boldsymbol{A}$. En pratique, le produit des $\boldsymbol{Q}^{(i)}$ est accumulé au fur et à mesure.

Exemple – Appliquons cet algorithme à la matrice qui nous sert d'exemple.

$$\boldsymbol{F} = \begin{bmatrix} 1. & 1. & .5 \\ 1. & 1. & .25 \\ .5 & .25 & 2. \end{bmatrix}$$

Le bas de la première colonne constitue le vecteur à traiter

$$\texttt{x = F(2:3,1)} = \begin{bmatrix} 1. \\ .5 \end{bmatrix}.$$

Nous calculons successivement

`k = sqrt(x'*x) = 1.118034`,

puis

$$\texttt{v = x - k* [1, 0]' =} \begin{bmatrix} -.1180340 \\ .5 \end{bmatrix}$$

que nous normalisons

$$\texttt{v = v/sqrt(v'*v) =} \begin{bmatrix} -.2297529 \\ .9732490 \end{bmatrix}.$$

Nous en déduisons

$$\texttt{P = eye(2,2) - 2*v*v' =} \begin{bmatrix} .8944272 & .4472136 \\ .4472136 & -.8944272 \end{bmatrix}.$$

Vérifions ce premier résultat

$$\texttt{P*F(2:3,1) = Px =} \begin{bmatrix} 1.118034 \\ 2.776E-16 \end{bmatrix}.$$

Construisons la matrice $\boldsymbol{Q}$ avant de tridiagonaliser $\boldsymbol{F}$:

`Q = [1 0 0;0 P(1,1) P(1,2);0 P(2,1) P(2,2)] =`

$$\begin{bmatrix} 1. & 0. & 0. \\ 0. & .8944272 & .4472136 \\ 0. & .4472136 & -.8944272 \end{bmatrix}.$$

Pour une matrice 3×3, il n'y a qu'une étape de tridiagonalisation :

$$\texttt{F1 = clean(Q*F*Q) =} \begin{bmatrix} 1. & 1.118034 & 0. \\ 1.118034 & 1.4 & -.55 \\ 0. & -.55 & 1.6 \end{bmatrix}.$$

Quelques vérifications élémentaires :

`trace(F1) = 4. = trace(F)`

et

`det(F1) = -.0625 = det(F)`.

Vous pourrez refaire ces calculs plus rapidement en utilisant la fonction `householder` de Scilab :

$$\texttt{u = householder(x,[1,0]') =} \begin{bmatrix} -0.2297529 \\ 0.9732490 \end{bmatrix}$$

10.6.2. Calcul des valeurs propres

Soit $\boldsymbol{J}$ une matrice tridiagonale symétrique réelle d'ordre n :

$$\boldsymbol{J} = \begin{bmatrix} \delta_1 & \gamma_2 & 0 & \cdots & 0 \\ \gamma_2 & \delta_2 & \gamma_3 & \cdots & 0 \\ 0 & \gamma_3 & \delta_3 & \gamma_4 & \cdots \\ \vdots & & \ddots & \ddots & \gamma_n \\ & & & \gamma_n & \delta_n \end{bmatrix}.$$

Nous supposons les γ_i tous différents de zéro. Nous pouvons former facilement le polynôme caractéristique de $\boldsymbol{J}$, en procédant par récurrence. Définissons en effet la matrice partielle :

$$\boldsymbol{J}_i = \begin{bmatrix} \delta_1 & \gamma_2 & 0 & \cdots \\ \gamma_2 & \delta_2 & \gamma_3 & \cdots \\ \cdots & & & \cdots \\ & & & \gamma_i \\ & & \gamma_i & \delta_i \end{bmatrix}$$

et le déterminant

$$p_i(x) = \det[\boldsymbol{J}_i - x\boldsymbol{I}].$$

En développant ce déterminant en fonction des éléments de la dernière colonne, nous trouvons les relations :

$$\begin{aligned} &p_0(x) = 1; \quad p_1(x) = \delta_1 - x; \\ &p_i(x) = (\delta_i - x)p_{i-1}(x) - \gamma_i^2 p_{i-2}(x), i = 2, 3, \ldots, n, \end{aligned} \tag{10.19}$$

qui permettent de calculer facilement $p_n(x) = \det \boldsymbol{J}$.

Toute méthode efficace de recherche des zéros d'un polynôme est maintenant applicable au calcul des valeurs propres (les racines de p_n) ; la méthode de bissection est stable et commode. On peut aussi utiliser l'algorithme de Newton, en calculant la dérivée de p_n à l'aide de la récurrence :

$$\begin{aligned} &p_0'(x) = 0; \quad p_1'(x) = -1, \\ &p_i'(x) = -p_{i-1}(x) + (\delta_i - x)p_{i-1}'(x) - \gamma_i^2 p_{i-2}'(x). \end{aligned}$$

Il est recommandé de localiser ces racines avant de faire appel à un programme qui convergera alors plus sûrement. Nous avons vu (§ 5.7.3) que les suites de Sturm étaient un moyen puissant de localisation des racines d'un polynôme. Nous allons appliquer ce formalisme à la matrice `F1` formée au paragraphe précédent, en utilisant les outils fournis par Scilab.

Nous pouvons former le polynôme caractéristique de `F1`, comme ceci :

```
-->p = poly(F1,"s")
                      2   3
    0.0625 + 3.6875s - 4s + s
```

et appliquer l'algorithme de Sturm. Il est plus rapide d'utiliser le théorème suivant.

> **Théorème** – Les polynômes définis par la récurrence (10.19) forment une suite de Sturm. Si $w(a)$ désigne le nombre de changements de signe dans la suite $p_0(a), p_1(a), \ldots, p_n(a)$, alors $w(a)$ est égal au nombre de valeurs propres de $\boldsymbol{J}$ plus petites que a. Par convention, le signe de $p_i(a)$ est pris opposé à celui de $p_{i-1}(a)$ au cas où $p_i(a) = 0$.
> De plus, les zéros de p_{i-1} séparent ceux de p_i.

Dans le cas simple de la matrice 3×3 `F1`, nous utilisons les capacités en calcul algébrique de Scilab pour former les p_i (attention à l'inversion de numérotation par rapport au chapitre 5) :

```
-->x = poly(0,"x");
-->p0 = 1;
-->p1 = 1-x
-->p2 = (1.4-x)*p1-F1(1,2)^2
                2
    0.15 - 2.4x + x
-->p3 = (1.6-x)*p2 -F1(2,3)*F1(3,2)*p1
                      2   3
  - 0.0625 - 3.6875x + 4x - x
```

Il nous reste à calculer les valeurs de ces polynômes en quelques points (fonction `horner(p,a)`). Nous avons trouvé

a	0	1	3
p_0	+	+	+
p_1	+	$(0)-$	−
p_2	+	−	+
p_3	−	−	−
$w(a)$	1	1	3

ce qui montre que l'une des valeurs propres est négative et que les deux autres sont comprises entre 1 et 3.

En pratique, on calculera numériquement $p_n(a)$ à l'aide des relations (10.19).

10.7. Matrices hermitiennes

On rencontre souvent, en physique, des matrices hermitiennes. Une matrice $\boldsymbol{A}$ à éléments complexes est dit hermitienne si la complexe conjuguée de $\boldsymbol{A}$ coïncide avec la transposée de $\boldsymbol{A}$: $\boldsymbol{A}^* = \boldsymbol{A}^T$. Comment chercher les éléments propres d'une telle matrice ? La méthode de Householder peut s'étendre à des matrices complexes. La programmation en Pascal est pénible, puisque ce langage ignore le type complexe et que l'on est obligé de réécrire toutes les opérations arithmétiques. Au contraire, FORTRAN, C++, Java, Scilab ou Maple traitent sans problème les nombres complexes. Une autre méthode consiste à séparer partie réelle et partie imaginaire du problème de valeurs propres. Nous cherchons des solutions de :

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x},$$

où $\boldsymbol{A}$ est hermitienne et λ est réel. Posons $\boldsymbol{A} = \boldsymbol{A'} + i\boldsymbol{A''}$, $\boldsymbol{x} = \boldsymbol{x'} + i\boldsymbol{x''}$. L'hermiticité de $\boldsymbol{A}$ impose la symétrie de $\boldsymbol{A'}$ et l'antisymétrie de $\boldsymbol{A''}$ ($\boldsymbol{A''}^T = -\boldsymbol{A''}$). En séparant partie réelle et partie imaginaire :

$$\boldsymbol{A'}\boldsymbol{x'} - \boldsymbol{A''}\boldsymbol{x''} = \lambda \boldsymbol{x'},$$
$$\boldsymbol{A''}\boldsymbol{x'} + \boldsymbol{A'}\boldsymbol{x''} = \lambda \boldsymbol{x''}.$$

Ces relations peuvent être considérées comme le développement par blocs de l'équation :

$$\begin{bmatrix} \boldsymbol{A'} & -\boldsymbol{A''} \\ \boldsymbol{A''} & \boldsymbol{A'} \end{bmatrix} \begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{x''} \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{x''} \end{bmatrix}.$$

Nous avons ainsi remplacé un problème $n \times n$ complexe par un problème $2n \times 2n$ réel. Le nombre de valeurs propres a-t-il doublé pour autant ?

10.8. Pour en savoir plus

– M. Schatzman : *Analyse numérique, une approche mathématique*, ch. 6 (Dunod, Paris, 2001).

– P. Lascaux, R. Théodor : *Analyse numérique matricielle appliqué à l'art de l'ingénieur*, ch. 6,10,11 (Masson, Paris, 1993).

– G. Allaire, S.M. Kaber : *Algèbre linéaire numérique, cours et exercices* (Ellipses, Paris, 2002).

– G. Allaire, S.M. Kaber : *Introduction à Scilab, Exercices pratiques corrigés d'algèbre linéaire* (Ellipses, Paris, 2002).

– W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery : *Numerical Recipes, The Art of Scientific Computing*, ch. 11 (Cambridge University Press, Cambridge, 2007).

– Polycopiés des cours d'analyse numérique de MM. E. Hairer et G. Wanner : ch. 5, valeurs et vecteurs propres :
`http://www.unige.ch/~hairer/polycop.html`

Les matrices rectangulaires admettent une « décomposition en valeurs singulières » (Singular Value Decomposition, SVD) que l'on peut considérer comme une généralisation de la diagonalisation des matrices carrées ; elle est en particulier utile pour le traitement des problèmes de moindres carrés mal conditionnés.

10.9. Exercices

Exercice 1

On donne la matrice

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

Utiliser la méthode de la puissance n-ième pour calculer la plus grande valeur propre λ_1 de $\boldsymbol{A}$ et le vecteur propre associé $\boldsymbol{v}_1$, en utilisant le vecteur initial $[1, 0]^T$. On arrêtera le calcul lorsque deux approximations successives de λ_1 différeront de moins de 0,01.

Exercice 2

Chercher la plus petite valeur propre de la matrice définie à l'exercice précédent par la méthode de la puissance n-ième de l'inverse.

Exercice 3

La droite (D) du plan xOy fait l'angle $\theta/2$ avec l'axe Ox. Exprimer, en fonction de $\cos\theta$ et de $\sin\theta$, la matrice qui représente une symétrie orthogonale par rapport à (D).

Exercice 4

Étant donné un vecteur réel non nul $\boldsymbol{u}$ et une matrice réelle symétrique d'ordre n $\boldsymbol{A}$, on sait que le quotient de Rayleigh $Q(\boldsymbol{u})$ s'écrit

$$Q(\boldsymbol{u}) = \frac{\boldsymbol{u}^T \boldsymbol{A} \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{u}}$$

a) Si $\boldsymbol{v}_i$ est un vecteur propre de $\boldsymbol{A}$, associé à la valeur propre λ_i, calculer $Q(\boldsymbol{v}_i)$ $(1 \leq i \leq n)$.

b) Soient θ_i $(i = 1, 2, ...n)$ les coefficients du développement de $\boldsymbol{u}$ sur la base des $\boldsymbol{v}_i$. Exprimer $Q(\boldsymbol{u})$ en fonction des θ_i et des λ_i. On suppose que les valeurs propres λ_i ont été numérotées en ordre décroissant : $\lambda_1 > \lambda_2 \cdots > \lambda_n$. Montrer que, quel que soit $\boldsymbol{u} \neq 0$, on a :

$$\lambda_1 \geq Q(\boldsymbol{u}) \geq \lambda_n.$$

c) On donne la matrice $\boldsymbol{B}$ et le vecteur $\boldsymbol{u}$:

$$\boldsymbol{B} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}; \boldsymbol{u} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

Former $Q(\boldsymbol{u}) = Q(\theta)$ et en déduire les valeurs propres de $\boldsymbol{B}$.

Exercice 5

Montrer comment on peut utiliser le quotient de Rayleigh pour accélérer le calcul de la valeur propre dominante par la méthode de la puissance n-ième. Appliquer à la matrice de l'exercice 1, avec le même vecteur initial.

Exercice 6

On reprend les définitions de l'exercice 3 et on cherche à déterminer la valeur propre immédiatement inférieure à la plus grande, soit λ_2, et le vecteur propre correspondant $\boldsymbol{v}_2$.

On définit pour cela la matrice :

$$\boldsymbol{A'} = \boldsymbol{A} - \lambda_1 \boldsymbol{v}_1 \boldsymbol{v}_1^T$$

où $\boldsymbol{v}_1$ est supposé normalisé à l'unité : $\boldsymbol{v}_1^T \boldsymbol{v}_1 = 1$.

a) Quels sont les valeurs propres et les vecteurs propres de $\boldsymbol{A'}$? Quel élément propre peut-on déterminer en appliquant la méthode de la puissance n-ième à cette matrice ?

b) On sait que la matrice particulière :

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

admet la valeur propre $2 + \sqrt{2}$ et le vecteur propre $[0,5; -1/\sqrt{2}; 0,5]$. Former $\boldsymbol{A'}$ et estimer numériquement un autre élément propre, à l'aide d'une seule itération de la méthode de la puissance n-ième associée au quotient de Rayleigh, pour le vecteur initial $\boldsymbol{x}_0 = [1; 0,5; -0,5]$.

Exercice 7

Appliquer l'algorithme QR au calcul des éléments propres de la matrice tridiagonale $F1$ obtenue au § 10.6.1. Combien faut-il d'itérations pour obtenir la même précision sur les valeurs propres qu'au § 10.5.2 ?

Exercice 8

On examine ici une méthode de réduction de l'ordre d'une matrice utile pour la recherche d'une valeur propre. Les hypothèses et les notations sont celles de l'exercice 3. Le vecteur propre $\boldsymbol{v}_1$ est supposé normalisé pour que sa composante la plus grande soit égale à 1 ; soit s l'ordre de cette composante. On suppose aussi que tous les autres vecteurs propres sont normalisés de la même manière : leur composante d'ordre s vaut l'unité. Enfin, on appelle $\boldsymbol{a}_s^T$ la s-ième ligne de $\boldsymbol{A}$ et on pose :

$$\boldsymbol{A'} = \boldsymbol{A} - \boldsymbol{v}_1 \boldsymbol{a}_s^T.$$

a) En utilisant la définition de $\boldsymbol{v}_1$, calculer le produit scalaire $\boldsymbol{a}_s^T \boldsymbol{v}_1$.

b) Calculer les éléments de la ligne s de $\boldsymbol{A'}$.

c) Montrer que $\boldsymbol{v}_1$ est vecteur propre de $\boldsymbol{A'}$; quelle est la valeur propre correspondante ?

d) Calculer le produit scalaire $\boldsymbol{a}_s^T \boldsymbol{v}_i$.

e) Montrer que $\boldsymbol{v}_i - \boldsymbol{v}_1$ est vecteur propre de $\boldsymbol{A'}$ et trouver la valeur propre associée.

f) En examinant la structure de $\boldsymbol{A'}$ et de ses vecteurs propres, expliquer l'intérêt du procédé décrit ci-dessus.

Exercice 9

$\boldsymbol{A}$ étant une matrice carrée $n \times n$, on pose :

$$r_i = \sum_{j \neq i}^{n} |a_{i,j}|$$

(r_i est la somme des modules des éléments non diagonaux de la ligne i) et on appelle d_i le disque de centre $a_{i,i}$ et de rayon r_i.

a) On se propose de démontrer que toute valeur propre λ de $\boldsymbol{A}$ appartient à l'un des disques d_i. Soit $\boldsymbol{x} = [x_1, x_2, ..., x_n]^T$ le vecteur propre associé à λ. À quelles relations obéissent les coordonnées x_i ? Soit k un entier pour lequel la relation $|x_k| = \sup |x_i|$ est vérifiée. Trouver une borne supérieure de $|\lambda - a_{k,k}|$ et en déduire la propriété annoncée.

b) Soit $\boldsymbol{D}$ la matrice diagonale dont les éléments diagonaux coïncident avec ceux de la diagonale de $\boldsymbol{A}$ ($d_{i,i} = a_{i,i}$ et $d_{i,j} = 0$ si $i \neq j$). On pose encore $\boldsymbol{E} = \boldsymbol{A} - \boldsymbol{D}$ et

$$\boldsymbol{A}(\epsilon) = \boldsymbol{D} + \epsilon \boldsymbol{E}.$$

Quelles sont les matrices $\boldsymbol{A}(0)$ et $\boldsymbol{A}(1)$? On note $\lambda_i(\epsilon)$ les valeurs propres de $\boldsymbol{A}(\epsilon)$ et on admet qu'elles sont des fonctions continues de ϵ. Dans quelle région du plan complexe trouve-t-on les images des nombres $\lambda_i(\epsilon)$? On suppose que p disques d_i forment une région connexe disjointe des $n - p$ autres disques. Utiliser les résultats précédents pour prouver que la région formée par la réunion des p premiers disques contient p valeurs propres.

c) Application. Localiser le mieux possible les valeurs propres des deux matrices suivantes :

$$\boldsymbol{A} = \begin{bmatrix} 4 & 1 & 0 \\ 1 & 0 & -1 \\ 1 & 1 & -4 \end{bmatrix}; \quad \boldsymbol{B} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 4 \end{bmatrix}$$

10.10. PROJETS

Projet 1. Vibrations d'une « molécule » linéaire

Un modèle simpliste de molécule est constitué d'une file de masses capables de glisser sans frottement le long d'un axe Ox. Chaque masse est liée à ses deux voisines par un ressort. Nous commençons par une « molécule » triatomique, pour laquelle les masses ont pour valeurs $m_1 = m_3 \equiv m$ et $m_2 \equiv \theta m$. Les constantes de raideur des ressorts sont toutes égales à k.

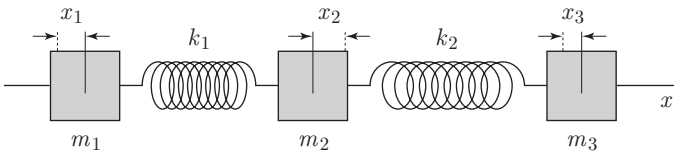


Figure 10.5 – Schéma d'une molécule triatomique linéaire.

En notant x_1, x_2, x_3 les déplacements des trois masses à partir de leurs positions d'équilibre (où les ressorts n'exercent aucune force), nous trouvons que les équations différentielles du mouvement sont

$$\begin{aligned} m\ddot{x}_1 &= -k(x_1 - x_2), \\ \alpha m\ddot{x}_2 &= -k(x_2 - x_1) - k(x_2 - x_3), \\ m\ddot{x}_3 &= -k(x_3 - x_2). \end{aligned}$$

Nous faisons l'hypothèse que les masses sont animées d'un mouvement sinusoïdal, de même pulsation ω, $x_i = X_i e^{i\omega t}$. Substituant dans les équations du mouvement, nous trouvons, après simplification par $e^{i\omega t}$:

$$\begin{aligned} m\omega^2 X_1 &= k(X_1 - X_2), \\ \alpha m\omega^2 X_2 &= k(X_2 - X_1) + k(X_2 - X_3), \\ m\omega^2 X_3 &= k(X_3 - X_2). \end{aligned}$$

Il est commode de dédimensionnaliser ces équations en posant $\omega_0^2 = k/m$ puis $\beta^2 = (\omega/\omega_0)^2$. Nous introduisons encore le vecteur $\boldsymbol{x} = [X_1, X_2, X_3]^T$. Le système s'écrit commodément sous forme matricielle

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \boldsymbol{x} = \beta^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}$$

ou encore

$$\boldsymbol{K}\boldsymbol{x} = \beta^2 \boldsymbol{M}\boldsymbol{x}.$$

La matrice $\boldsymbol{K}$ représente les effets des ressorts (matrice de raideur), tandis que $\boldsymbol{M}$ représente l'effet des masses (matrice d'inertie).

Nous sommes en présence d'un problème de valeurs propres « généralisé », qu'il est facile de ramener au cas habituel. Définissons pour cela une matrice $\boldsymbol{M}^{1/2}$, telle que $[\boldsymbol{M}^{1/2}]^2 = \boldsymbol{M}$; c'est élémentaire ici, parce que $\boldsymbol{M}$ est diagonale et que ses éléments diagonaux sont strictement positifs. En appelant $\boldsymbol{M}^{-1/2}$ l'inverse de $\boldsymbol{M}^{1/2}$, nous faisons le changement d'inconnues $\boldsymbol{x} = \boldsymbol{M}^{-1/2}\boldsymbol{y}$. En insérant cette définition dans l'équation aux valeurs propres, on trouve

$$\boldsymbol{M}^{-1/2}\boldsymbol{K}\boldsymbol{M}^{-1/2}\boldsymbol{y} = \beta^2 \boldsymbol{y},$$

ce qui constitue un problème aux valeurs propres pour la matrice symétrique

$$\boldsymbol{M}^{-1/2}\boldsymbol{K}\boldsymbol{M}^{-1/2}.$$

Écrire un programme pour déterminer les fréquences et les amplitudes relatives de vibrations des trois masses.

Projet 2. Modèle de Hückel

Nous nous intéressons ici aux propriétés électroniques des molécules d'hydrocarbures planes comportant des doubles liaisons, comme l'éthylène (éthène), le butadiène, le

benzène, la pyridine. Ces propriétés sont essentiellement celles des électrons les moins liés, appelés « électrons π », qui résident dans des orbitales $2p_z$ orientées perpendiculairement au plan de la molécule. Les autres particules (noyaux et électrons) ne font que créer un potentiel électrostatique qui agit sur les électrons considérés. Les électrons π sont considérés comme indépendants les uns des autres.

La fonction d'onde de chaque électron est représentée comme une combinaison linéaire d'orbitales atomiques $2p_z$ (approximation « LCAO » selon l'expression anglaise) :

$$\Psi_k = \sum_1^n c_{ki}\varphi_i$$

si n est le nombre d'atomes participant aux doubles liaisons ($n = 2$ pour l'éthylène, $n = 6$ pour le benzène ou la pyridine). $\varphi_i(\boldsymbol{r}) \equiv \varphi(\boldsymbol{r} - \boldsymbol{r}_i)$ est l'orbitale $2p_z$ portée par l'atome i, situé au point $\boldsymbol{r}_i$. L'indice k numérote les différentes orbitales moléculaires (OM) que l'on peut construire à partir des orbitales atomiques (OA). Il y aura autant d'OM que d'OA, $1 \leq k \leq n$.

Nous déterminons les coefficients inconnus c_{ki} par la méthode variationnelle. Les meilleures valeurs des c_{ki} sont celles qui minimisent l'énergie de l'OM k :

$$\langle E_k \rangle \equiv \frac{\langle \Psi_k | \mathcal{H} | \Psi_k \rangle}{\langle \Psi_k | \Psi_k \rangle}.$$

Les quantités entre crochets $\langle \ldots \rangle$ sont des valeurs moyennes au sens de la mécanique quantique. Développons cette expression :

$$\langle E_k \rangle = \frac{\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ki}^* c_{kj} \langle \varphi_i | \mathcal{H} | \varphi_j \rangle}{\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ki}^* c_{kj} \langle \varphi_i | \varphi_j \rangle}$$

Dans le modèle de Hückel, on n'explicite pas l'opérateur hamiltonien $\mathcal{H}$. Celui-ci est défini implicitement par la donnée de ses « éléments de matrice » sur la base des OA, les $\langle \varphi_i | \mathcal{H} | \varphi_j \rangle$. Pour une molécule ne comportant que des atomes de carbone (les hydrogènes ne jouent aucun rôle), les éléments de matrice du hamiltonien ont pour valeurs

$$\langle \varphi_i | \mathcal{H} | \varphi_i \rangle = \alpha,$$

$$\langle \varphi_i | \mathcal{H} | \varphi_j \rangle = \begin{cases} \beta & \text{si et seulement si les atomes i et j sont séparés par une liaison,} \\ 0 & \text{autrement.} \end{cases}$$

Comme le hamiltonien est un opérateur quantique hermitien,

$$\langle \varphi_i | \mathcal{H} | \varphi_j \rangle = \langle \varphi_j | \mathcal{H} | \varphi_i \rangle.$$

Toutes ces quantités sont des nombres réels. Au voisinage du minimum,$\langle E_k \rangle$ ne doit pas varier, quelles que soient les petites variations δc_{ki} des coefficients. En différenciant $\langle E_k \rangle$, on trouve la condition

$$(\boldsymbol{H} - E_k \boldsymbol{I})\boldsymbol{c}_k = 0.$$

En d'autres termes, l'énergie de l'OM k est une valeur propre de la matrice $\boldsymbol{H}$, tandis que le vecteur des coefficients $\boldsymbol{c}_k$ est le vecteur propre correspondant. Les éléments de $\boldsymbol{H}$ se déduisent des règles précédentes :

$$H_{ii} = \alpha; \quad H_{ij} = H_{ji} = \beta$$

si les atomes i et j sont liés. La recherche des OM et de leurs énergies est ainsi ramenée à la résolution d'un problème de valeurs propres et de vecteurs propres.

Traitons le cas de l'éthylène pour montrer la démarche. La formule de l'éthène est CH_2=CH_2. Il y a deux « atomes lourds » engagés dans une double liaison, les deux carbones, numérotés 1 et 2. Ils portent chacun une OA $2p_z$, notée φ_1 et φ_2. Les OM seront de la forme

$$\Psi_1 = c_{11}\varphi_1 + c_{12}\varphi_2 \quad ; \quad \Psi_2 = c_{21}\varphi_1 + c_{22}\varphi_2.$$

La matrice représentative du hamiltonien a pour éléments diagonaux $H_{11} = H_{22} = \alpha$ et comme éléments extradiagonaux $H_{12} = H_{21} = \beta$. Les éléments propres de $\boldsymbol{H}$ vérifient

$$\begin{bmatrix} \alpha - E_k & \beta \\ \beta & \alpha - E_k \end{bmatrix} \begin{bmatrix} c_{k1} \\ c_{k2} \end{bmatrix} = 0.$$

Comme α représente, à l'approximation de Hückel, l'énergie d'un électron π dans un atome isolé (aussi appelée « intégrale de Coulomb »), il est commode de faire une translation d'origine en posant $x' = \alpha - E$. De plus, nous adimensionnons l'équation en divisant chaque ligne par β (souvent appelée « l'intégrale de résonance ») et nous posons

$$x \equiv \frac{x'}{\beta} = \frac{\alpha - E}{\beta}.$$

L'équation aux valeurs propres devient

$$\begin{vmatrix} x & 1 \\ 1 & x \end{vmatrix} = 0.$$

Les solutions sont $x = \pm 1$ ou encore $E = \alpha \pm \beta$. Les vecteurs propres sont

$$\boldsymbol{c}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad ; \quad \boldsymbol{c}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

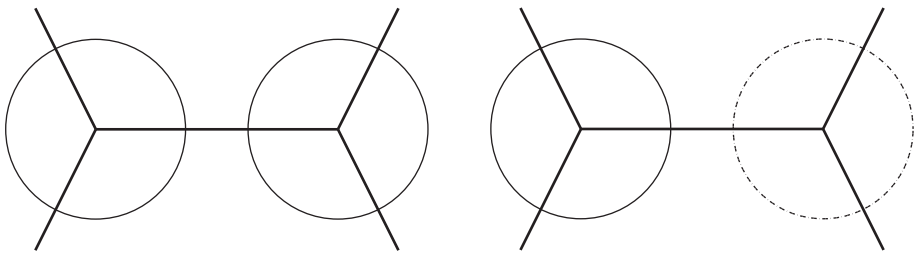


Figure 10.6 – Représentation schématique des orbitales moléculaires.
orbitale liante (à gauche) ; orbitale antiliante (à droite).
Les tirets indiquent une région où la fonction d'onde est négative.

Pour interpréter ces résultats, il faut encore savoir que α et β sont négatifs. Le niveau d'énergie le plus bas (le plus stable) est $E_1 \equiv \alpha + \beta$, associé à l'OM $\Psi_1 = \varphi_1 + \varphi_2$. Chaque atome de carbone possède quatre électrons de valence, dont 3 sont engagés dans les liaisons C-H ou C-C de type σ (« hybridation sp^3 ») et qui ne sont pas pris en compte par le modèle de Hückel. Il reste un électron $2p$ par carbone et c'est cette paire électronique qui constitue la liaison π. Vous voyez que l'OM Ψ_1 a une énergie plus basse qu'une OA isolée ($\alpha + \beta$ au lieu de α). Autrement dit, lorsque l'éthylène se forme, l'énergie des deux électrons $2p$ passe de 2α à $2(\alpha + \beta)$. Cette orbitale est dite « liante ». Au contraire nous dirons que Ψ_2, dont l'énergie est supérieure à celle d'une OA isolée ($\alpha - \beta$ au lieu de α), est « antiliante ».

L'OM $\Psi_2 = \varphi_1 - \varphi_2$ a l'énergie $E_2 \equiv \alpha - \beta$; elle est normalement vide. Il est possible de « promouvoir » un électron de Ψ_1 à Ψ_2. C'est ce qui arrive lorsque la molécule absorbe un photon d'énergie $E_2 - E_1 = -2\beta$.

Utiliser le modèle de Hückel pour étudier les énergies et les orbitales moléculaires d'hydrocarbures simples : butadiène, benzène, naphtalène, azulène. À quelle fréquence (ou à quelle longueur d'onde) se situe la première bande d'absorption de ces espèces ?

Chapitre 11

Problèmes différentiels à conditions initiales

Un très grand nombre de questions scientifiques ou techniques admettent des modèles mathématiques qui impliquent des équations différentielles. Songez à la dynamique des corps matériels ou à la cinétique des réactions chimiques. Les cours de mathématiques sont riches de méthodes analytiques de résolution des équations différentielles : on pourrait croire que l'analyse numérique est inutile dans cette partie des mathématiques. La réalité est tout autre : seule une infime minorité des équations différentielles est analytiquement soluble, particulièrement les équations linéaires. A mesure que l'on s'écarte des problèmes scolaires pour s'approcher des questions réelles, les équations différentielles deviennent « de moins en moins » linéaires et de « moins en moins » solubles. Dans ce chapitre, nous décrivons quelques familles d'algorithmes de résolution de problèmes différentiels et nous énonçons qualitativement leurs propriétés, leurs qualités et leurs défauts.

Il est important de bien faire la différence entre équation différentielle et problème différentiel. La première admet une solution dépendant de paramètres inconnus (n constantes arbitraires pour une équation d'ordre n). Pour cette raison, on ne peut jamais obtenir de solution numérique d'une équation différentielle. Une équation différentielle associée à des conditions initiales ou à des conditions aux limites constitue un problème différentiel. Un tel problème admet généralement une solution unique et c'est l'objet de ce chapitre que d'expliquer comment on peut l'approcher numériquement.

Comme nous venons de le suggérer, on distingue deux types de problèmes différentiels, où l'inconnue est toujours la fonction y. Les problèmes à condition(s) initiale(s) (ou « de Cauchy »), dont un exemple s'écrit :

$$y' = f[x, y(x)] \quad ; \quad y(a) = A \quad ; \quad x \geq a, \tag{11.1}$$

et les problèmes aux limites, comme par exemple :

$$y'' = g[x, y(x), y'(x)] \quad ; \quad y(a) = A \quad ; \quad y(b) = B \quad ; \quad a \leq x \leq b. \tag{11.2}$$

Dans ce chapitre, nous ne considérons que la première catégorie, les problèmes à condition(s) initiale(s). Nous supposerons que l'équation différentielle considérée est résolue par rapport à la dérivée d'ordre le plus élevé, comme c'est le cas dans les deux formules précédentes. Il peut arriver en pratique que cela ne soit pas vrai : on parle alors d'équation algébrodifférentielle ; ce type d'équation ne sera pas traité ici. Le « second membre » $f(x, y)$ est une fonction continue de ses deux arguments à l'intérieur d'un domaine D de $\mathbb{R}^2$; le point $[a, A]$ appartient à D. Désignons par Y la solution exacte du problème différentiel considéré. Nous dirons que cette fonction est effectivement solution de (11.1) sur $[a, b]$ si, pour tout x de cet intervalle, le point $(x, Y(x))$ appartient à D, la condition initiale est vérifiée : $Y(x_0) = A$ et Y' existe et vérifie $Y'(x) = f(x, Y(x))$. Vous trouverez dans les cours d'analyse une étude détaillée des conditions que doit remplir la fonction f pour que (11.1) ait une solution. Nous pouvons résumer cette discussion comme suit.

> **Théorème** – Si la fonction continue $f(x, y)$ obéit à la condition de Lipschitz
>
> $$|f(x, y_1) - f(x, y_2)| \leq K|y_1 - y_2|, \quad a < x \leq b$$
>
> quels que soient y_1 et y_2 pour un réel $K \geq 0$, indépendant de y_1 et y_2, alors (11.1) admet une solution unique sur $[a, b]$.

La condition de Lipschitz est vérifiée par toute fonction f telle que $\partial f(x, y)/\partial y$ existe et est bornée dans le domaine considéré. Pour expliquer les algorithmes de résolution, nous supposerons presque toujours que l'équation différentielle à résoudre est du premier ordre (comme dans (11.1)). L'hypothèse d'un problème du premier ordre peut paraître fort éloignée de la réalité ; c'est tout à fait vrai, mais ce n'est pas bien grave, comme vous allez le voir. Toute équation différentielle d'ordre supérieur à un est équivalente à un système différentiel d'ordre un. Soit, par exemple, l'équation du second ordre qui décrit exactement le mouvement d'un pendule simple, en absence de frottement, avec des variables sans dimensions :

$$y'' = -k^2 \sin y.$$

Introduisons la variable auxiliaire $z = y'$; nous pouvons maintenant écrire l'équation précédente sous la forme

$$\begin{cases} y' = z, \\ z' = -k^2 \sin y. \end{cases} \tag{11.3}$$

Nous voila confrontés à un système d'équations différentielles (ou système différentiel) du premier ordre. La forme générale d'un système différentiel du premier ordre à deux fonctions inconnues est :

$$\begin{cases} y' = h[x, y(x), z(x)], \\ z' = g[x, y(x), z(x)]. \end{cases} \tag{11.4}$$

Dans l'exemple du pendule, la fonction h se réduisait à la variable z. Si nous définissons deux vecteurs : $\boldsymbol{r} = (y, z)$ et $\boldsymbol{s} = (h, g)$, nous pouvons mettre (11.4) sous la forme :

$$\boldsymbol{r'} = \boldsymbol{s}(x, \boldsymbol{r}). \tag{11.5}$$

Cette écriture est commode car elle permet d'obtenir sans peine un algorithme de résolution d'un système différentiel : il suffit de transposer en notation vectorielle

l'algorithme valable pour une équation différentielle (du premier ordre). Les ouvrages classiques d'analyse numérique développent largement les théories relatives à la résolution des problèmes différentiels à une fonction inconnue, mais sont discrets quant aux systèmes différentiels. Nous suivrons cet exemple et nous admettrons que les algorithmes développés pour une fonction inconnue sont encore valables, avec des notations adaptées, pour plusieurs fonctions.

Toutes les méthodes de résolution numérique des problèmes différentiels utilisent une discrétisation. Ayant fait le choix d'un pas h, on va approcher la solution y pour les valeurs kh de la variable indépendante, obtenant ainsi une suite de valeurs y_k avec $y_0 = y(a)$. Les divers algorithmes se distinguent par les informations qu'ils utilisent pour calculer y_k : certains emploient y_{k-1} uniquement (on parle de méthode ou de schéma à pas séparés ou à un pas), d'autres utilisent un certain nombre de valeurs précédentes de y, dont y_{k-1} (ce sont les méthodes/schémas à pas liés ou à pas multiples). On pourrait tout aussi bien considérer que y_k est déterminé par une relation de récurrence à deux ou plusieurs termes.

Cependant, avant d'aborder les algorithmes purement numériques, nous allons passer en revue quelques méthodes analytiques qui peuvent servir à des calculs numériques.

11.1. Méthodes analytiques

11.1.1. Développement de Taylor

Nous cherchons la solution du problème différentiel classique (11.1). Le théorème de Taylor offre, au moins en principe, une solution

$$y(x) = y(a) + (x-a)y'(a) + \frac{1}{2}(x-a)^2 y''(a) + \frac{1}{6}(x-a)^3 y'''(a) + \dots$$

Nous connaissons $y(a)$ (condition initiale), $y'(a)$ (par substitution dans l'équation différentielle), $y''(a)$ et toutes les dérivées d'ordre supérieur (par dérivation et substitution dans l'équation différentielle !). Ainsi :

$$y'' = [f(x,y)]' = f_x + f f_y, \tag{11.6}$$

où f_x, f_y représentent les dérivées partielles de f par rapport aux variables indiquées. Ensuite :

$$y''' = f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_x f_y + f f_y^2.$$

La méthode est simple mais devient rapidement laborieuse. De plus, nous avons montré (§ 2.3) que le développement de Taylor pouvait converger très lentement. En conséquence, cette méthode ne s'emploie que « localement » : pour calculer y_{k+1} à partir de y_k ou pour calculer, à l'aide des conditions initiales, les premières valeurs de y dont certains algorithmes ont besoin pour démarrer et calculer la suite des y_k.

11.1.2. Méthode des coefficients indéterminés (Frobenius)

Nous faisons l'hypothèse que y peut s'écrire :

$$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \dots$$

et donc que :

$$y' = c_1 + 2c_2 x + 3c_3 x^2 + \dots$$

En substituant ces développements dans l'équation différentielle, nous obtenons des relations entre coefficients c_i, que nous résolvons de proche en proche, à partir de $c_0 = A$. Ayant ces coefficients, il est simple d'approcher y, à une certaine précision. Cette méthode est aussi assez laborieuse.

11.1.3. Méthode de Picard, ou d'approximations successives

Nous nous proposons toujours de résoudre le problème de Cauchy (11.1). Nous intégrons les deux membres de l'équation différentielle, entre a et x (en renommant u la variable indépendante) :

$$\int_a^x y' du = y(x) - y(a) = \int_a^x f(u, y) du$$

Nous ne savons bien sûr pas calculer l'intégrale du second membre (ce qui reviendrait à savoir résoudre l'équation différentielle), aussi allons-nous procéder à une approximation brutale : nous remplaçons y par une « approximation d'ordre zéro », $y^{[0]}$, que nous prenons égale à y_0 ; l'intégrale est alors calculable en principe ; le résultat nous donne l'approximation suivante de y, soit $y^{[1]}$:

$$y^{[1]}(x) = y(a) + \int_a^x f(u, y^{[0]}(u)) du$$

Nous itérons ce procédé, jusqu'à ce qu'un critère de convergence convenable soit atteint :

$$y^{[n]}(x) = y(a) + \int_a^x f(u, y^{[n-1]}(u)) du.$$

Cette méthode est peut-être encore plus lente que les précédentes !

Depuis quelques années, on trouve dans le commerce des programmes de manipulation algébrique (*Derive, Maple, Mathematica, Macsyma, Mupad, etc.*) fonctionnant sur microordinateurs et capables de faire seuls les calculs algébriques des trois méthodes précédentes. Il est à prévoir que ces algorithmes vont connaître de ce fait une nouvelle jeunesse.

11.2. Méthodes d'Euler et de Taylor

La méthode d'Euler n'est jamais employée en pratique (sauf peut-être comme constituant d'une méthode de résolution d'équations aux dérivées partielles), mais c'est un excellent exemple introductif, généralisable selon plusieurs directions intéressantes. L'algorithme proposé par Euler pour résoudre (11.1) s'écrit :

$$y_{n+1} = y_n + hf(x_n, y_n) = y_n + hf_n. \tag{11.7}$$

Nous allons donner trois interprétations de cette formule.

- Approximation de la dérivée. La dérivée de y au point x_n est à peu près $y'(x_n) \simeq (y_{n+1} - y_n)/h$; en écrivant qu'elle vaut f_n, nous retrouvons la méthode d'Euler.
- Approximation d'une intégrale. Nous intégrons les deux membres de l'équation (11.1) :

  $$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f[t, y(t)]dt$$

  Nous remplaçons l'intégrale du second membre par son approximation la plus simple, l'aire d'un rectangle de base h et de hauteur f_n (méthode du « point gauche » ou du « rectangle à droite », § 8.5), ce qui donne :

  $$y_{n+1} - y_n = hf_n.$$

- Développement de Taylor. Supposant connu y_n, nous pouvons approcher y_{n+1} à l'aide du théorème de Taylor :

  $$y(x_n + h) = y(x_n) + hy'_n + \mathcal{O}(h^2).$$

  En négligeant le terme d'erreur et en identifiant $y(x_n + h)$ et y_{n+1}, nous retrouvons la formule d'Euler.

Résumons les caractéristiques de la méthode d'Euler. Pour calculer y_{n+1}, nous avons utilisé la valeur de y_n et pas celles de $y_{n-1}, y_{n-2}, \ldots$ Il s'agit donc d'une méthode à un pas (ou à pas séparés). Pour passer de y_n à y_{n+1}, nous ne calculons f qu'une fois (les méthodes de Runge–Kutta qui seront exposées plus loin utilisent plusieurs valeurs de f pour améliorer la précision sur y'). L'algorithme ne fait appel qu'à $f = y'$ et pas à ses dérivées d'ordre supérieur (Certains algorithmes s'appuient sur un développement de Taylor, et donc sur $y^{[n]}$, qui dépend des dérivées de f). Enfin, comme nous le verrons plus en détail plus tard, le calcul de y_{n+1} s'accompagne d'une « erreur de troncation » proportionnelle à h^2 et c'est pourquoi ce schéma est dit d'ordre un.

La programmation de l'algorithme d'Euler est des plus simples. Voici une ébauche de programme.

Lire h, x_0, y_0, x_{max}
Initialiser x à x_0, y à y_0
Tant que $x < x_{max}$ faire :
    calculer $f = f(x, y)$
    $y = y + hf$
    $x = x + h$
    imprimer x, y

La généralisation à un système différentiel est immédiate. La fonction inconnue, sa valeur initiale et le second membre sont maintenant remplacés par des vecteurs de $\mathbb{R}^n$:

Lire $h, x_0, x_{max}, \boldsymbol{y}_0$
Initialiser x à x_0, $\boldsymbol{y}$ à $\boldsymbol{y}_0$
Tant que $x < x_{max}$ faire :
    calculer :
        $\boldsymbol{f} = \boldsymbol{f}(x, \boldsymbol{y})$
        $\boldsymbol{y} = \boldsymbol{y} + h\boldsymbol{f}$
        $x = x + h$
    imprimer $x, \boldsymbol{y}$

Il faut traiter sur un pied d'égalité toutes les coordonnées de chaque vecteur.

Exemple – Le programme ci-dessous résout, par la méthode d'Euler, l'équation différentielle du pendule simple : $y'' + \sin y = 0$. Comme nous l'avons déjà dit, on doit en fait résoudre le système du premier ordre équivalent

$$\begin{cases} y' = z, \\ z' = -\sin y. \end{cases}$$

Les deux seconds membres apparaissent dans le programme sous forme de `function` ; la `function euler` calcule la solution en `tf=ti+h`. Ces trois sous-programmes sont inutilement compliqués : ils contiennent par exemple la variable indépendante (`t`), sans objet ici, mais ils peuvent s'adapter facilement au cas d'un pendule soumis à un couple dépendant du temps. Le programme principal ne fait que gérer les éléments précédents et construire pas à pas les vecteurs solutions.

Listing 11.1 – Mouvement du pendule traité par l'algorithme d'Euler

```
function F = smy(t,y,z)                                          1
  F = z;                                                         2
endfunction                                                      3
function F = smz(t,y,z)                                          4
  F = -sin(y);                                                   5
endfunction                                                      6
function [yf,zf,tf] = euler(h,ti,yi,zi)                          7
  yf = yi + h*smy(ti,yi,zi); zf = zi + h*smz(ti,yi,zi);          8
  tf = ti + h;                                                   9
endfunction                                                      10
```

```
np = 600;
t = zeros(np,1);
y = zeros(t); z = zeros(t);
y0 = input('position initiale: ');
z0 = input('vitesse initiale: ');
pas = input('longueur du pas: ');
t(1) = 0; y(1) = y0; z(1) = z0;
for i = 2:np do
   [y(i),z(i),t(i)] = euler(pas,t(i-1),y(i-1),z(i-1));
end
plot2d(t,y)
fprintfMat("D:\an_008\pdl_euler.dta",[t,y]);
```

La figure 11.1 montre le résultat de l'exécution. Comme vous pouvez le constater, la qualité de la méthode laisse à désirer : l'amplitude augmente nettement avec le temps.

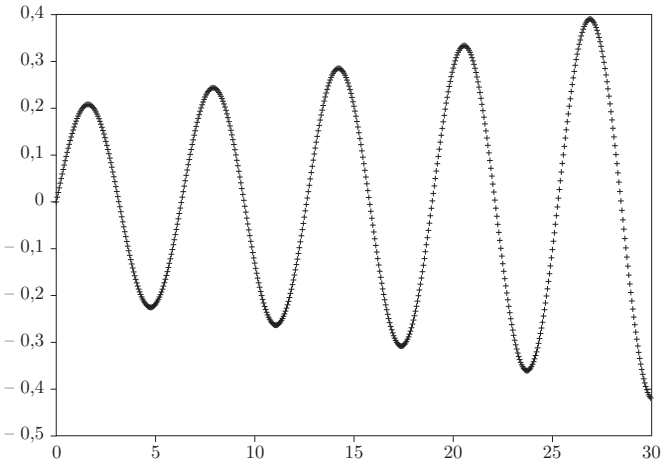


Figure11.1 – Mouvement du pendule simple obtenu par l'algorithme d'Euler. Le pas était de 0,05.

Comme nous l'avons indiqué au § 11.2, la méthode d'Euler peut être considérée comme une application, à l'ordre 1, du développement de Taylor. Nous avons aussi montré comment l'on pouvait utiliser un développement d'ordre supérieur. Rappelons qu'il faut calculer, jusqu'à l'ordre souhaité, les dérivées de y, par les relations de récurrence :

$$y' = f(x,y) \quad ; \quad y'' = f_x(x,y) + f_y(x,y)f \equiv f^{(1)},$$
$$f^{(k)} = f_x^{(k-1)} + f_y^{(k-1)} f,$$

puis déduire y_{i+1} par :

$$y_{i+1} = y_i + h\left[f(x_i,y_i) + \ldots + \frac{h^{p-1}}{p!} f^{(p-1)}(x_i,y_i)\right]$$

Cet algorithme n'est intéressant que si les $f^{(k)}$ se calculent facilement et sont suffisamment régulières sur l'intervalle considéré.

11.3. Méthodes de Runge–Kutta

Les méthodes que nous allons décrire dans ce paragraphe peuvent être considérées comme des généralisations de celle d'Euler, où l'on calcule f (le second membre) plusieurs fois par pas, pour réduire l'erreur de troncation. Ce sont aussi probablement les méthodes les plus employées dans la pratique. Nous exposerons la théorie pour une méthode d'ordre 2, bien que les applications fassent généralement appel à des méthodes d'ordre 4 ou plus (pour une définition de l'ordre d'un schéma numérique, voir ci-dessous et le § 4).

11.3.1. Méthodes d'ordre 2

Nous cherchons toujours la solution du problème représenté par (11.1) en supposant connue la solution numérique approchée y_n au point $x_n = x_0 + nh$. Pour cela, nous allons construire une « fonction incrément » $\Phi(x_n, y_n, h)$ telle que :

$$y_{n+1} - y_n = h\Phi(x_n, y_n, h), \quad 0 \leq n < N. \tag{11.8}$$

Cette fonction dépend aussi du second membre f. Soit d'autre part la solution exacte $z(t)$ du problème différentiel :

$$z'(t) = f[t, z(t)] \quad ; \quad z(x) = y.$$

En termes imagés, $z(t)$ est la solution de l'équation différentielle « qui passe par le point de coordonnées x et y » ; x et y sont ici considérés comme arbitraires mais constants. L'incrément exact est défini par :

$$z(x+h) - z(x) = h\Delta(x, y, h). \tag{11.9}$$

Dans le cas du schéma d'Euler, on avait $\Phi = f$. Nous allons chercher, pour obtenir une méthode d'ordre 2, une expression de Φ qui coïncide avec Δ jusqu'aux termes en h compris (un schéma est dit d'ordre p si la quantité $\Delta - \Phi$ est $\mathcal{O}(h^p)$, comme expliqué au § 4). Nous supposons (comme l'ont fait Runge et Kutta il y a plus d'un siècle) que Φ est de la forme :

$$\Phi(x, y, h) = b_1 f(x, y) + b_2 f[x + p_1 h, y + p_2 h f(x, y)] \tag{11.10}$$

où les quantités b_1, b_2, p_1 et p_2 sont des constantes à déterminer. Il nous suffit d'imposer que le développement de Taylor de Φ, au premier ordre en h, coïncide avec celui de Δ au même ordre. Ils s'écrivent

$$\Delta = f(x, y) + \frac{1}{2}h(f_x + f f_y) + \mathcal{O}(h^2),$$

$$\Phi = (b_1 + b_2)f + h b_2 p_1 f_x + h b_2 p_2 f f_y + \mathcal{O}(h^2).$$

D'où trois conditions à remplir pour que $\Delta \equiv \Phi$:

$$\begin{cases} b_1 + b_2 = 1, \\ b_2 p_1 = 1/2, \\ b_2 p_2 = 1/2. \end{cases} \tag{11.11}$$

Nous disposons de quatre paramètres inconnus ; il est d'usage de conserver un paramètre libre et de déterminer les trois autres à l'aide des relations ci-dessus ; la dernière quantité sera ajustée après pour obtenir une propriété désirable de l'algorithme (bonne stabilité, erreur d'arrondi faible...). Nous choisissons de conserver b_2, rebaptisé β pour la circonstance. (11.11) devient :

$$\begin{cases} b_1 = 1 - \beta, \\ b_2 = \beta, \\ p_1 = p_2 = \frac{1}{2\beta}. \end{cases}$$

Nous obtenons ainsi une méthode de Runge–Kutta d'ordre 2 caractérisée par la fonction incrément :

$$\Phi(x, y, h) = (1 - \beta) f(x, y) + \beta f[x + \frac{h}{2\beta}, y + \frac{h}{2\beta} f(x, y)].$$

Parmi toutes les valeurs possibles de β, deux cas sont restés dans l'histoire : $\beta = 1/2$ (on parle alors de méthode de Heun ou d'Euler améliorée) et $\beta = 1$ (méthode d'Euler modifiée). Vous vérifiez facilement que ces algorithmes s'écrivent :

$$\beta = 1/2 \quad ; \quad y_{n+1} = y_n + \frac{h}{2} \{f(x_n, y_n) + f[x_n + h, y_n + h f(x_n, y_n)]\} \tag{11.12}$$

et

$$\beta = 1 \quad ; \quad y_{n+1} = y_n + h f[x_n + \frac{h}{2}, y_n + \frac{h}{2} f(x_n, y_n)]. \tag{11.13}$$

Exemple – Reprenons le problème du pendule simple. Il nous suffit de remplacer, dans le programme précédent, la `function euler` par une `function rk2` que voici.

```
function [yf,zf,tf] = rk2(h,ti,yi,zi)                                    1
  yf = yi + 0.5*h*(smy(ti,yi,zi)                                         2
          + smy(ti+h,yi+h*smy(ti,yi,zi),zi+h*smz(ti,yi,zi)));            3
  zf = zi + 0.5*h*(smz(ti,yi,zi)                                         4
          + smz(ti+h,yi+h*smy(ti,yi,zi),zi+h*smz(ti,yi,zi)));            5
  tf = ti + h;                                                           6
endfunction                                                              7
```

Comme la variable `ti` n'est pas utilisée dans les fonctions, la valeur de cet argument au moment de l'appel est sans importance ; elle est néanmoins conforme à la définition de l'algorithme. La figure 11.2 montre la loi du mouvement produite par ce programme. Vous pouvez constater que, cette fois, l'amplitude reste constante, malgré une valeur initiale très grande (quasiment égale à π). Dans ces conditions, le pendule est un système mécanique non-linéaire, dont le mouvement n'est plus sinusoïdal ni isochrone.

11.3.2. Méthode d'ordre d'ordre 4

Nous n'allons pas établir ici les formules de Runge–Kutta d'ordre supérieur à 2 ; en effet, la longueur des calculs croît très rapidement avec l'ordre. Nous nous contenterons de quelques remarques générales.

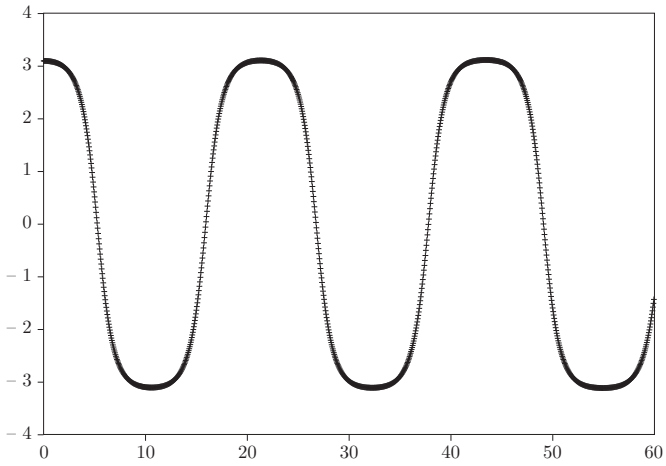


Figure 11.2 – Mouvement du pendule calculé par la méthode RK2. Le pas était de 0,05.

Un schéma de Runge–Kutta à s « étages » est défini par les formules

$$\begin{aligned}
&k_1 = f(x_n, y_n),\\
&k_2 = f(x_n + c_2 h, y_n + h a_{2,1} k_1),\\
&k_3 = f(x_n + c_3 h, y_n + h(a_{3,1} k_1 + a_{3,2} k_2)),\\
&\cdots\\
&k_s = f(x_n + c_s h, y_n + h(a_{s,1} k_1 + \cdots + a_{s,s-1} k_{s-1})),\\
&y_{n+1} = y_n + h(b_1 k_1 + \cdots + b_s k_s),\\
&x_{n+1} = x_n + h.
\end{aligned} \tag{11.14}$$

Les valeurs des coefficients k_i dépendent de x et sont donc différentes pour chaque valeur de n. Pour des méthodes explicites, on a $c_1 = 0$ et $a_{s,j} = 0$ pour $j \geq s$. On résume souvent cette méthode par le tableau

$$\begin{array}{c|ccccc}
c_1 & 0 & 0 & \cdots & 0 & 0\\
c_2 & a_{21} & 0 & \cdots & 0 & 0\\
c_3 & a_{31} & a_{32} & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss-1} & 0\\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Avec ces notations, la méthode d'Euler modifiée est symbolisée comme

$$\begin{array}{c|cc}
0 & 0 & 0\\
1/2 & 1/2 & 0\\
\hline
 & 0 & 1
\end{array}$$

Quel est le tableau associé à l'algorithme d'Euler amélioré ? Que se passerait-il si la matrice des $a_{s,j}$ n'était pas strictement triangulaire inférieure ?

La méthode de Runge–Kutta « classique » comporte quatre étages et on démontre qu'elle est également d'ordre 4. Elle est définie par le tableau

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
	1/6	2/6	2/6	1/6

ou par les formules

$$\begin{aligned} k_1 &= f(x_n, y_n), \\ k_2 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1), \\ k_3 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2), \\ k_4 &= f(x_n + h, y_n + hk_3), \\ y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\ x_{n+1} &= x_n + h. \end{aligned} \tag{11.15}$$

En pratique, on traite le plus souvent un système différentiel. La « fonction » inconnue y est alors un vecteur, de même que la « fonction » du second membre f. L'expérience montre que la mise en oeuvre de la méthode « RK-4 » est alors assez déroutante au début ; il vaut mieux expliciter en détail toutes les formules, quitte à les simplifier par la suite. C'est ce que nous avons fait dans le programme qui suit.

Listing 11.2 – Mouvement du pendule traité par l'algorithme de Runge–Kutta d'ordre 4

```
function F = smy(t,y,z)
  F = z;
endfunction
function F = smz(t,y,z)
  F = -sin(y);
endfunction
function [yf,zf,tf] = rk4(h,ti,yi,zi)
  k1y = smy(ti,yi,zi);
  k1z = smz(ti,yi,zi);
  k2y = smy(ti+h/2,yi+h*k1y/2,zi+h*k1z/2);
  k2z = smz(ti+h/2,yi+h*k1y/2,zi+h*k1z/2);
  k3y = smy(ti+h/2,yi+h*k2y/2,zi+h*k2z/2);
  k3z = smz(ti+h/2,yi+h*k2y/2,zi+h*k2z/2);
  k4y = smy(ti+h,yi+h*k3y,zi+h*k3z);
  k4z = smz(ti+h,yi+h*k3y,zi+h*k3z);
  yf = yi + (h/6)*(k1y + 2*k2y + 2*k3y + k4y);
  zf = zi + (h/6)*(k1z + 2*k2z + 2*k3z + k4z);
  tf = ti+h;
endfunction
```

```
np = 600;                                                          20
t = zeros(np,1);                                                   21
y = zeros(t); z = zeros(t);                                        22
y0 = input('position initiale: ');                                 23
z0 = input('vitesse initiale: ');                                  24
pas = input('longueur du pas: ');                                  25
t(1) = 0; y(1) = y0; z(1) = z0;                                    26
for i = 2:np do                                                    27
   [y(i),z(i),t(i)] = rk4(pas,t(i-1),y(i-1),z(i-1));               28
end                                                                29
plot2d(t,y)                                                        30
```

L'architecture générale est la même que pour les méthodes d'Euler ou de Runge–Kutta d'ordre 2 ; la fonction `rk4` fait avancer la solution du point `ti` au point `ti+h`. Lorsque nous serons sûrs que ce programme fonctionne, nous pourrons le simplifier notablement. Dans le cas particulier qui nous occupe, la fonction `smy` est identique à son troisième argument. Nous pourrons donc écrire les lignes de code suivantes

```
    k1y = zi;                                                      1
    k2y = zi+h*k1z/2;                                              2
    k3y = zi+h*k2z/2;                                              3
    k4y = zi+h*k3z;                                                4
```

Vous pourrez simplifier aussi les quantités relatives à z en utilisant le fait que `smz` n'est autre que l'opposé du sinus de son deuxième argument. Il faut néanmoins respecter rigoureusement l'ordre des définitions ; par exemple `k3y` dépend de `k2z` (ligne 3) et doit donc être calculé après.

11.3.3. Avantages et inconvénients des méthodes de Runge–Kutta

La méthode RK4 se distingue par sa simplicité de mise en oeuvre ; nous la recommandons d'ailleurs pour tous les problèmes simples. Elle ne requiert que quatre évaluations de f par pas. On peut améliorer aisément la précision des calculs (si le langage et le compilateur le permettent) en calculant $\Phi = \sum a_j k_j$ en précision multiple, car c'est là que se produisent la plupart des erreurs d'arrondi.

L'algorithme de Runge–Kutta souffre de deux inconvénients liés. On ne dispose d'aucune indication sur l'erreur de troncation en cours de calcul. D'autre part, il y a des problèmes de stabilité dès que le second membre est grand ou rapidement variable. Une méthode simple (mais coûteuse) de surveillance de la précision consiste à conduire deux calculs simultanément, l'un avec le pas h, l'autre avec le pas $h/2$. Tant que la « distance » entre les deux résultats reste inférieure à un certain seuil, on admet que le résultat est valable. Si le seuil est franchi, il faut rejeter la (ou les) dernière(s) valeur(s) calculée(s) et repartir avec un pas plus petit.

Il existe des algorithmes plus raffinés (qui portent les noms de Runge–Kutta–Fehlberg ou méthodes de Runge–Kutta emboîtées). Dans ce type de méthode, on calcule, par

exemple, six fois f par pas ; on combinant « bien » les résultats, on obtient une estimation précise de y_{n+1} ; une autre combinaison des mêmes valeurs fournit une estimation de l'erreur. La seule difficulté réside dans la forme compliquée des coefficients qui remplacent les a_j et b_j, mais l'ordinateur n'en a cure. Dormand et Prince ont publié, vers 1980, des versions performantes de ces méthodes ; vous les trouverez par exemple sur le site du Professeur Hairer.

11.3.4. Organisation d'un programme

Il est commode de construire le programme de résolution sous une forme bien hiérarchisée, qui permette de changer commodément d'algorithme et d'équation. Le sous-programme de niveau le plus bas fait avancer la solution de x_n à x_{n+1} ; c'est lui qui renferme les détails de l'algorithme. Ce « noyau » de niveau zéro est englobé dans un sous-programme qui gère la surveillance de l'erreur (quand elle existe). Avec un pas h, on calcule y_{n+1} et une estimation τ_{n+1} de l'erreur locale de troncation ; si celle-ci est inférieure au seuil, le pas est accepté ; sinon, on rejette le dernier résultat, on diminue le pas et on recommence. Cet ensemble est à son tour inclus dans un sous-programme qui conduit l'intégration de x_{min} à x_{max} et prend en charge le stockage des résultats intermédiaires. Celui-ci est un peu délicat si l'on a affaire à des pas de longueur variable : comment concilier cette démarche avec l'enregistrement des résultats pour des abscisses à peu près régulièrement réparties ? Il est fréquent, quel que soit l'algorithme d'intégration, que l'on doive utiliser un pas assez petit, ce qui conduit à calculer de nombreuses valeurs de x_n, alors qu'on ne peut ou ne veut conserver qu'un petit nombre de valeurs (pour les tracés par exemple). On peut concilier ces deux exigences en ne conservant qu'une valeur de x toutes les p. La couche extérieure du programme est spécifique de l'application ; elle est destinée à la lecture des paramètres et des valeurs initiales, à l'affichage des résultats. Les seconds membres sont en général définis dans un sous-programme spécial.

11.4. Ordre, stabilité et convergence des méthodes à un pas

Nous n'avons pas la prétention de construire la solution exacte du problème différentiel quel que soit l'algorithme utilisé ; le résultat du calcul pratique est en réalité entaché d'erreurs dont il est intéressant de comprendre la nature. L'analyse générale des schémas numériques de résolution des problèmes différentiels est beaucoup trop longue pour être abordée ici ; nous nous contenterons d'évoquer les principales questions. Nous cherchons la solution dans le domaine $a = x_0 \leq x \leq x_N = b$ et nous supposons le pas h identique pour chacun de N intervalles, si bien que $h = (b-a)/N$. Nous supposons que la valeur initiale ($y_0 = A$) est connue exactement.

Supposons d'abord que les calculs sont effectués avec un très grand nombre de chiffres significatifs. Le résultat d'une étape de la méthode (y_{n+1}) ne sera pas pour autant exact : il a en effet été obtenu à l'aide d'une approximation. Les grandeurs qui interviennent sont indiquées sur la figure 11.3, dans le cas particulier du schéma d'Euler.

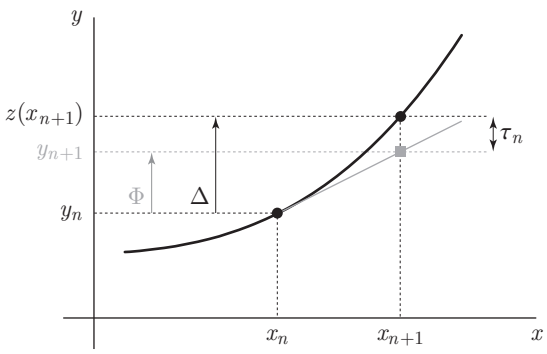


Figure 11.3 – Erreur de troncation pour l'algorithme d'Euler.

Le point de coordonnées (x_n, y_n) est le dernier point calculé. La courbe en noir représente la solution du problème différentiel

$$z' = f(x, z(x)); \quad z(x_n) = y_n.$$

On peut dire que c'est la solution de l'équation différentielle « qui passe par le point (x_n, y_n) » (et sans doute pas par (x_0, A) comme prévu par (11.1)). Au paragraphe précédent, nous avons défini les deux incréments Δ et Φ et l'erreur locale de troncation $\tau_n = h(\Delta - \Phi)$. La méthode d'Euler consiste à construire la tangente à la courbe au point d'abscisse x_n et à la prolonger jusqu'à l'abscisse $x_n + h$. Nous obtenons ainsi le point d'ordonnée y_{n+1}. L'erreur locale de troncation (ou erreur de méthode) est

$$\tau_n = z(x_{n+1}) - y_{n+1} = \frac{h^2}{2} z''(x_n) + \mathcal{O}(h^3) = \frac{h^2}{2} \left[f'_x + f'_z f\right]_n + \mathcal{O}(h^3). \tag{11.16}$$

La deuxième expression résulte de l'application du théorème de Taylor et la troisième du calcul de la dérivée première de f comme au paragraphe précédent. On dit que le schéma d'Euler est d'ordre 1 (erreur équivalente à h^2). Plus généralement, une méthode est dite d'ordre p si l'erreur pour un pas vérifie (avec les notations du § 11.3)

$$\tau_n = z(x_{n+1}) - y_{n+1} = h\left[\Delta(x, y, h) - \Phi(x, y, h)\right] = \mathcal{O}(h^{p+1}) \text{ quand } h \to 0. \tag{11.17}$$

En annulant cette différence jusqu'au terme en h^2 compris, nous avons, au paragraphe précédent, construit l'incrément Φ d'un schéma de Runge–Kutta d'ordre 2.

L'analyse théorique fait appel à une notion plus globale, la cohérence (plus souvent appelée consistance, comme en anglais). Un schéma tel que 11.8 est dit consistant si

$$\lim_{h \to 0} \sum_{0 \le n < N} |\tau_n| = 0. \tag{11.18}$$

Une autre propriété importante d'un algorithme de résolution d'équations différentielles est la stabilité. Nous allons illustrer ce concept par le cas particulier suivant. Considérons deux problèmes différentiels presque identiques et qui ne diffèrent que par une petite modification de la condition initiale (qui pourrait être due à une erreur d'arrondi ou une erreur de mesure).

$$\begin{cases} y' = f(x, y), \\ y(a) = A. \end{cases} \qquad \begin{cases} y' = f(x, y), \\ y(a) = A + \delta. \end{cases}$$

Appelons y la solution du premier système, $\hat{y}$ la solution du système perturbé. L'algorithme de résolution est dit stable si

$$\max_{0 \leq n \leq N} |y_n - \hat{y}_n| \leq S|\delta|. \tag{11.19}$$

Le nombre réel positif S est appelé « constante de stabilité ». On peut démontrer la condition suffisante de stabilité énoncée ci-dessous.

> **Théorème : condition suffisante de stabilité d'un schéma à un pas** – Pour que la méthode soit stable, il suffit que la fonction incrément Φ vérifie une condition de Lipschitz par rapport à y. En d'autres termes, il doit exister une constante $L \geq 0$ telle que, pour $x \in [a, b]$, $y_1, y_2 \in \mathbb{R}^2$ et quel que soit h, on ait
>
> $$|\Phi(x, y_1, h) - \Phi(x, y_2, h)| \leq L|y_1 - y_2|.$$
>
> La « constante de stabilité » vaut alors $S = e^{L(b-a)}$.

Dans le cas de l'algorithme d'Euler on a $\Phi = f$: ce schéma est donc stable partout où la solution de l'équation différentielle existe. Vous avez sans doute remarqué que la théorie ne dit rien des valeurs de L ou de S. En fait, il est facile de trouver des cas où ces constantes sont si grandes que l'algorithme devient inutilisable.

Nous allons décrire un procédé simple d'étude de la stabilité mais qui fournit souvent un critère plus précis que le théorème précédent. Il s'agit d'étudier un « problème modèle » dont la solution analytique et la solution algorithmique sont également faciles à construire. Soit le problème différentiel :

$$y' = cy; \quad y(0) = 1.$$

Ici, $f \equiv cy$ et l'algorithme d'Euler s'énonce :

$$y_{n+1} = y_n + hcy_n = (1 + hc)y_n.$$

Il s'agit d'une relation de récurrence (on dit aussi une équation aux différences) pour y_{n+1}. Comme $y_0 = 1$, nous trouvons :

$$y_n = (1 + hc)^n.$$

Imaginez que la condition initiale devienne $y(0) = 1 + \delta$. Que pourrez vous dire de la différence $|y_n - \hat{y}_n|$? Nous allons appliquer un raisonnement différent pour montrer que l'algorithme est convergent, c'est à dire que $y_n/y(x_n) \to 1$ pour $h \to 0$. Pour cela, il suffit de remarquer que $x_n = nh$ et donc que :

$$y_n = [(1 + ch)^{1/h}]^x \to e^{cx} \text{ quand } h \to 0.$$

Si $c < 0$, on a encore $y_n \to e^{-|c|x}$. Mais y_n ne sera uniformément décroissant (comme la solution exacte) que si $0 < |1 + hc| < 1$. Comme $c = -|c|$, cela implique $h < 2/|c|$. On dit que la méthode d'Euler est conditionnellement stable

Ce procédé (étude d'un problème modèle) s'applique assez facilement à tous les algorithmes de résolution du problème de Cauchy.

L'utilisateur s'intéresse plutôt à l'erreur globale, qui est la différence entre la solution numérique et la solution exacte, par exemple à l'extrémité de l'intervalle considéré ($n = N$). On espère que l'erreur globale tend vers zéro avec h ; si cette condition est remplie, on dit que l'algorithme est convergent. En fait, la convergence est une condition impérative pour que l'algorithme soit utilisable. On se doute que l'erreur globale dépend d'une certaine manière de la somme des erreurs locales. En réalité, la théorie est assez compliquée car on doit tenir compte de la propagation des erreurs jusqu'à la dernière valeur y_N. On démontre qu'un schéma cohérent et stable est aussi convergent ; c'est ce qu'exprime le théorème suivant.

Théorème de convergence des méthodes de Runge–Kutta – Soit y la solution exacte de (11.1) sur $[a, b]$. Si la méthode est d'ordre p, ou encore si

$$\| y(x+h) - y(x) - h\Phi(x, y, h) \| \leq Ch^{p+1},$$

si la fonction Φ vérifie une condition de Lipschitz

$$\| \Phi(x, y, h) - \Phi(x, z, h) \| \leq L \| y - z \|$$

lorsque les points (x, y) et (x, z) sont proches de la solution exacte, alors l'erreur globale est bornée et satisfait à

$$\| y(x_N) - y_N \| \leq h^p \frac{C}{L} \left(e^{L(b-a)} - 1 \right)$$

Imaginons maintenant que nous disposons d'un algorithme exact capable de calculer y_n sans erreur de troncation. Le résultat a cependant été obtenu par une succession d'opérations arithmétiques, sur une machine comportant un nombre limité de chiffres significatifs : il présente donc une certaine erreur d'arrondi :

$$e_n = y_n - y(x_n).$$

En particulier, les erreurs d'arrondi perturbent aussi bien les valeurs calculées y_n que la valeur initiale $y_0 = A$. Heureusement, ces erreurs ont un signe aléatoire et s'ajoutent donc de façon incohérente. Nous pouvons donc espérer que la précision du résultat restera convenable.

En pratique, les deux sources d'erreur (troncation et arrondi) coexistent, ce qui conduit à l'existence d'un pas d'intégration optimal. En effet, si h est très petit, les erreurs de troncation seront faibles mais comme le nombre de pas N sera grand, les erreurs d'arrondi deviendront importantes. À l'opposé, si nous choisissons un h grand, le nombre de pas sera limité et les erreurs d'arrondi seront négligeables ; par contre les erreurs de méthode deviendront insupportables.

Le problème traité fournit parfois lui-même le moyen de contrôler la qualité de l'algorithme. C'est le cas en particulier pour les problèmes de mécanique sans frottement, pour lesquels l'énergie est conservée. Il suffit alors de calculer périodiquement l'énergie du système : si la variation d'énergie dépasse le seuil prévu, on rejette les dernières itérations et on reprend le calcul avec un pas plus petit.

11.5. Méthodes à pas multiples

Les méthodes de Runge–Kutta posent parfois des problèmes de stabilité ou de précision. Nous allons maintenant décrire des méthodes plus stables et généralement plus précises ; le prix à payer pour ces améliorations sera une structure plus compliquée de l'algorithme.

11.5.1. Schémas explicites (ouverts)

Nous allons détailler un algorithme élémentaire à deux pas. Nous cherchons la solution du problème différentiel (11.1). La variable indépendante x sera discrétisée avec un pas h. En reprenant l'une des interprétations de la méthode d'Euler, nous remplaçons y' par une forme approchée, mais plus précise que celle utilisée au § 11.2 :

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

soit, en reportant dans l'équation différentielle :

$$y_{n+1} = y_{n-1} + 2h f_n + \frac{h^3}{3} y'''(\xi).$$

À partir de cette relation exacte, nous obtenons une méthode d'intégration en négligeant le terme d'erreur (qui fournira l'erreur locale de troncation) :

$$y_{n+1} = y_{n-1} + 2h f_n. \tag{11.20}$$

Par ailleurs, vous pourrez montrer sans peine que si l'on intègre terme à terme l'équation différentielle et que l'on remplace l'intégrale de $f(x, y)$ par son approximation dite du point milieu, on retrouve le même algorithme. L'amorçage du calcul suppose connus y_0 et y_1, que l'on obtient par Taylor ou Runge–Kutta d'ordre 2 ou encore par Euler. La figure 11.4 illustre cet algorithme.

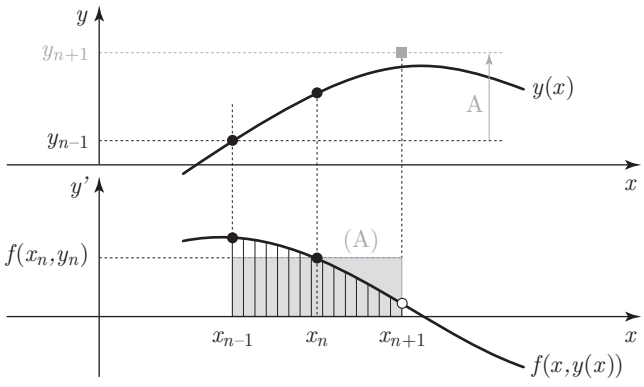


Figure 11.4 – Schéma d'Adams ouvert. L'aire du rectangle (A) (en gris) est une approximation de l'aire hachurée ; on l'ajoute à y_{n-1} pour obtenir y_{n+1}.

Pour obtenir des méthodes plus précises, nous allons encore intégrer « formellement » l'équation différentielle

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(u, y(u))du$$

et remplacer, dans l'intégrale du second membre, la fonction f par un polynôme d'interpolation $p(x)$ de degré k. Soit (x_n, y_n) le dernier point calculé. Le polynôme p s'appuiera sur les noeuds $n, n-1, n-2, \ldots, n-j, \ldots, n-k$ (au nombre de $k+1$). Si nous choisissons la forme de Lagrange de ce polynôme, nous pourrons écrire

$$p(x) = \sum_{j=0}^{j=k} \ell_j(x) f_{n-j},$$

en posant $f_{n-j} = f[(x_{n-j}, y(x_{n-j})]$. Nous estimons la nouvelle valeur de y par

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} p(u)du.$$

Nous obtenons ainsi un algorithme pour chaque valeur de k. Lorsque les pivots sont équidistants, on peut calculer une fois pour toutes l'intégrale de p, comme nous l'avons expliqué dans le cas de la méthode d'intégration de Newton–Cotes (§ 8.6.1) et comme l'ont fait Adams et Bashforth vers 1880. On trouve alors

$$y_{n+1} = y_n + h \sum_{j=0}^{j=k} \alpha_j f_{n-j}.$$

Les valeurs des α_j (différentes pour chaque choix de k) sont données dans le tableau ci-dessous.

k	α_j			
	$j =$ 0	1	2	3
0	1			
1	3/2	−1/2		
2	23/12	−16/12	5/12	
3	55/24	−59/24	37/24	−9/24

Pour $k = 3$ on a ainsi

$$y_{n+1} = y_n + \frac{h}{12}\left[23f(x_n, y_n) - 16f(x_{n-1}, y_{n-1}) + 5f(x_{n-2}, y_{n-2})\right].$$

Pour démarrer l'un de ces algorithmes, il faut se procurer (développement de Taylor, méthode à un pas...) les $k+1$ premières valeurs de y. Comment s'appelle le schéma correspondant à $k = 0$?

11.5.2. Schémas implicites (fermés)

Comme Adams, vous avez du remarquer que les algorithmes de la section précédente utilisent un polynôme d'interpolation en dehors de l'intervalle défini par les pivots et vous savez que cela peut entraîner de grosses erreurs. Cet auteur a donc proposé de construire un polynôme d'interpolation sur les pivots $n+1, n, \ldots, n-k+1$. Commençons par le cas le plus simple, l'interpolation linéaire. Nous intégrons (de façon approchée) l'équation différentielle entre x_n et x_{n+1} par la méthode des trapèzes et nous trouvons

$$y_{n+1} = y_n + \frac{h}{2}(f_n + f_{n+1}) - \frac{h^3}{12}y'''(\xi).$$

Nous négligeons le terme d'erreur pour obtenir une méthode d'intégration du problème différentiel :

$$y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n). \tag{11.21}$$

Cette méthode est stable et précise. Elle souffre cependant d'un petit défaut : elle est inapplicable en l'état. En effet, la relation ci-dessus est une équation (implicite) en y_{n+1}, puisque cette quantité figure dans f_{n+1}. Il serait dommage d'échouer si près du but ; en réalité, nous pouvons très bien résoudre l'équation en y_{n+1} par approximations successives, à condition de connaître une valeur initiale $y^{[0]}$. Il suffit d'appliquer la méthode du point fixe

$$y_{n+1}^{[k+1]} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{[k]})]$$

jusqu'à ce qu'un critère de convergence convenable soit atteint. La dernière valeur calculée est retenue comme valeur de y_{n+1}. La figure 11.5 est une représentation graphique d'une itération de ce calcul.

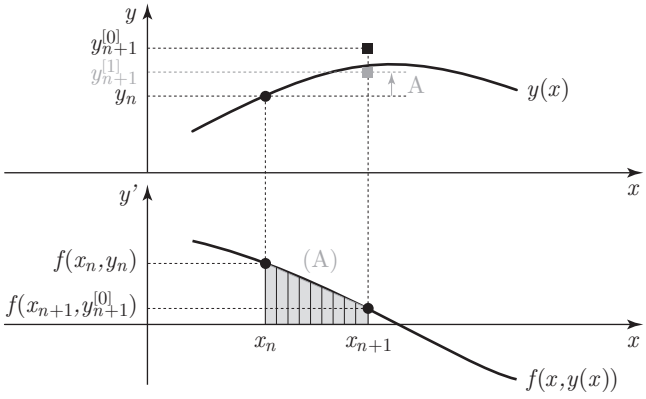


Figure 11.5 – Schéma d'Adams ouvert. L'aire du trapèze (A) (en gris) est une approximation de l'aire hachurée ; on l'ajoute à y_{n-1} pour obtenir y_{n+1}.

Pour plus de précision, nous pouvons avoir recours à une interpolation d'ordre plus élevé, comme pour les schémas explicites. Nous construisons maintenant le polynôme de degré k, qui s'appuie sur les $k+1$ pivots $n+1, n, \ldots, n-j, \ldots, n-k+1$ et

nous l'intégrons terme à terme entre les abscisses x_n et x_{n+1}. Dans le cas de noeuds équidistants, Adams et Moulton ont établi le schéma défini par les formules suivantes :

$$y_{n+1} = y_n + h \sum_{j=-1}^{j=k-1} \alpha_j^* f_{n-j}$$

avec

k	α_j^*			
	$j=$ -1	0	1	2
0	1			
1	1/2	1/2		
2	5/12	8/12	−1/12	
3	9/24	19/24	−5/24	1/24

Chaque ligne de ce tableau représente une équation non linéaire de la forme $y_{n+1} = A_n + hBf(x_{n+1}, y_{n+1})$ comme, par exemple, pour $k = 2$, $A_n = y_n + (h/12)(8f_n - f_{n-1})$, $B = 5/12$. Quel nom pouvez-vous proposer pour la méthode correspondant à $k = 0$?

11.5.3. Méthodes de prédiction-correction

Comment allons-nous nous procurer la valeur $y_{n+1}^{[0]}$ nécessaire au démarrage de l'itération qui conduira à y_{n+1} ? Vous l'avez deviné : par une méthode explicite. Celle-ci fournit une valeur approchée de la solution, que nous appelons maintenant la valeur « prédite », $y_{n+1}^{[P]}$ et cette valeur prédite servira à amorcer l'itération du schéma implicite, que l'on appelle aussi le correcteur. L'algorithme complet, utilisant la méthode du point milieu explicite et la méthode des trapèzes implicite, peut donc s'écrire, en détaillant les étapes :

$$\begin{array}{ll} y_{n+1}^{[P]} = y_{n-1} + 2hf_n, & \text{prédiction : } \mathbf{P} \\ \text{évaluation de } f(x_{n+1}, y_{n+1}^{[P]}), & \mathbf{E} \\ y_{n+1}^{[C]} = y_n + (h/2)[f_n + f(x_{n+1}, y_{n+1}^{[P]})], & \text{correction : } \mathbf{C} \\ \text{évaluation de } f(x_{n+1}, y_{n+1}^{[C]}), & \mathbf{E} \end{array} \tag{11.22}$$

Cette méthode est parfois appelée PECE. L'itération du correcteur peut, en principe, être répétée plusieurs fois (comme PECECE). Cela s'avère inutile en général : en cas de difficultés, il vaut mieux diminuer le pas. Dans tous les cas, $y_{n+1}^{[C]}$ devient la valeur définitive y_{n+1}. On montre que si l'équation différentielle obéit à une condition de Lipschitz pour y, avec une constante K et si h vérifie

$$Kh/2 < 1,$$

l'itération du correcteur converge.

Exemple – Nous résolvons une fois de plus le problème du pendule simple, à l'aide de l'algorithme de prédiction-correction. Les fonctions `smx, smy` sont inchangées, mais nous avons créé trois fonctions `pred, corct, pc` qui remplacent `rk2` et dont les codes sont présentés dans le listing 11-3.

Listing 11.3 – « Prédicteur » et « correcteur » d'ordre 2 pour le pendule

```
function [yp,zp] = pred(h,ynm1,znm1,tn,yn,zn)
    yp = ynm1 + 2*h*smy(tn,yn,zn)
    zp = znm1 + 2*h*smz(tn,yn,zn)
endfunction
function [yc,zc] = corct(h,tn,yn,zn,yp,zp)
    yc = yn + (h/2)*(smy(tn,yn,zn)+smy(tn+h,yp,zp))
    zc = zn + (h/2)*(smz(tn,yn,zn)+smz(tn+h,yp,zp))
endfunction
function [tnp1,ynp1,znp1] = pc(h,ynm1,znm1,tn,yn,zn)
    [yp,zp] = pred(h,ynm1,znm1,tn,yn,zn)
    [ynp1,znp1] =  corct(h,tn,yn,zn,yp,zp)
    tnp1 = tn + h
endfunction
```

Comme précédemment, on pourrait beaucoup simplifier ces instructions en tirant parti de la forme particulièrement simple des seconds membres. Il faut aussi modifier le programme principal pour utiliser trois valeurs de y. Les conditions initiales fournissent `y(1)` et `z(1)`. Les valeurs correspondantes pour le deuxième point sont obtenues par un développement de Taylor à l'ordre 2. Nous démarrons ensuite une boucle interne (compteur `j`), où nous calculons `yf` (soit y_{n+1}) en fonction de `yder` (y_n) et `yavder` (y_{n-1}) ; `zder` et `tder` sont traités de la même façon. Tous les `prd` pas, nous rangeons les valeurs de `tder,yder,zder` dans les vecteurs `t,y,z`. Cela nous permet d'utiliser un petit pas sans accumuler un trop grand nombre de valeurs. Voici le fragment de programme en question

Listing 11.4 – Programme de prédiction-correction pour le pendule

```
pas = 0.01; pp = pas*pas;
y(1) = %pi - 0.01; z(1) = 0; t(1) = 0;
tavder = t(1); yavder = y(1); zavder = z(1);
tder = t(1) + pas; yder = y(1) + pas*z(1) - (pp/2)*sin(y(1));
zder = z(1) - pas*sin(y(1))- (pp/2)*z(1)*cos(y(1));
for i = 2:n do
    for j = 1:prd do
        [tf,yf,zf] = pc(pas,yavder,zavder,tder,yder,zder);
        tavder = tder; yavder = yder; zavder = zder;
        tder = tf; yder = yf; zder = zf;
    end
    t(i) = tf; y(i) = yf; z(i) = zf;
end
```

11.5.4. Surveillance de l'erreur

Une qualité importante des méthodes de prédiction-correction est qu'elles permettent, sans calculs supplémentaires, une surveillance de l'erreur de troncation. La solution exacte Y_{n+1} de l'équation différentielle en x_{n+1} est reliée à la valeur prédite par la relation :

$$Y_{n+1} = y_{n+1}^{[P]} + \frac{h^3}{3} y'''(\xi_p).$$

Si le correcteur converge, il diffère de la solution exacte par son erreur de troncation :

$$Y_{n+1} = y_{n+1}^{[C]} - \frac{h^3}{12} y'''(\xi_c).$$

Nous disposons ainsi d'un encadrement de Y et d'une majoration de l'erreur. Si nous supposons en effet que $\xi_p \cong \xi_c$, nous pouvons écrire $y''' \cong (12/5h^3)(y^{[C]} - y^{[P]})_{n+1}$, d'où une estimation de l'erreur de troncation au point n (pour le calcul de y_{n+1}) :

$$e_n = \frac{1}{5}(y^{[P]} - y^{[C]})_{n+1}$$

L'utilisation de ce résultat est évidente. A chaque pas, nous calculons $|e_n|$; si cette quantité est supérieure à un seuil fixé à l'avance, nous rejetons ce pas, nous diminuons h et nous reprenons le calcul. Nous pouvons aussi augmenter h si $|e_n|$ est « très petit ».

11.5.5. Formules d'ordre 4

Nous donnerons plus loin une définition précise de l'ordre d'une méthode multipas. En pratique, on utilise le schéma explicite d'Adams–Bashforth à 4 pas associé au schéma implicite d'Adams–Moulton pour $k = 3$, le tout constituant une méthode d'ordre 4. Détaillons les formules correspondantes.

$$y_{n+1}^{[P]} = y_n + (h/24)(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}), \tag{11.23}$$

Le terme d'erreur est :

$$+\frac{251}{720} h^5 y^{(5)} \tag{11.24}$$

À cette équation de prédiction, nous associons le correcteur d'Adams-Moulton :

$$y_{n+1}^{[C]} = y_n + (h/24)(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}). \tag{11.25}$$

dont l'erreur de troncation s'écrit :

$$-\frac{19}{720} h^5 y^{(5)}. \tag{11.26}$$

Il est encore possible d'estimer l'erreur de troncation, et donc de la surveiller, à partir de la différence $y^{[P]} - y^{[C]}$, qui vaut, d'après (11.24) et (11.26) :

$$y_{n+1}^{[P]} - y_{n+1}^{[C]} = \frac{3}{8} h^5 y^{(5)}$$

La modification du pas h demande un peu d'attention. Si nous nous apercevons que l'erreur de troncation lors du calcul de y_{n+1} est trop grande, nous rejetons cette valeur et nous divisons h par 2. Le calcul peut-il repartir pour autant ? Pas tout de suite, car l'évaluation de y_{n+1} demande la connaissance des quatre valeurs précédentes de y ; or, nous ne les connaissons que pour le pas h et non pour $h/2$. Il faut donc interpoler entre valeurs de y pour trouver les précurseurs de y_{n+1} avec le pas $h/2$.

11.5.6. Avantages et inconvénients des méthodes à pas multiples

Les méthodes de prédiction-correction sont très appréciées des spécialistes, qui en ont poussé très loin l'analyse théorique (voir la section suivante). Leur principal avantage est qu'elles permettent une estimation aisée de l'erreur de troncation. Leur stabilité est aussi renommée. Les inconvénients des méthodes « PECE » sont dus à la complexité de l'algorithme. Une méthode d'ordre 4 requiert, pour démarrer, les 4 premières valeurs de y, lesquelles doivent être obtenues sans « trop » d'erreur, par exemple par Runge–Kutta d'ordre 4.

11.6. Ordre, stabilité et convergence des méthodes multi-pas

L'analyse théorique des méthodes à pas multiples est rendue compliquée justement par l'existence de plusieurs pas. La définition précise de l'erreur locale de troncation est plus subtile que nous ne l'avons laissé entendre au paragraphe précédent. Nous supposons connaître les valeurs de la solution exacte $y(x_i)$ pour $i = n, n+1, \ldots, n+k-1$ et nous employons ces valeurs dans un schéma multipas pour obtenir y_{n+k}. L'erreur de troncation est alors

$$y(x_{n+k}) - y_{n+k}.$$

On dit que la méthode est d'ordre p si l'erreur de troncation est $\mathcal{O}(h^{p+1})$ et on démontre que les schémas explicites d'Adams-Bashforth à k pas sont d'ordre k tandis que les méthodes implicites à k pas d'Adams-Moulton sont d'ordre $k+1$.

La définition de la stabilité d'une méthode multipas est analogue à celle utilisée pour les méthodes de Runge–Kutta : une petite perturbation des conditions initiales doit impliquer une erreur bornée à l'étape n. Les méthodes implicites, comme celles d'Adams sont en général stables. Plutôt que le cas général, examinons le cas particulier de la méthode d'Euler implicite. Intégrons 11.1 entre les abscisses x_n et x_{n+1}, en utilisant la méthode approchée dite du « rectangle à gauche » (§ 8.5) ; nous obtenons

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}).$$

Appliquons ce schéma à la résolution du problème modèle

$$y' = cy; \quad y(0) = 1.$$

Nous obtenons la relation de récurrence

$$y_{n+1} = y_n + hcy_{n+1}$$

ou encore

$$y_{n+1} = \frac{y_n}{1 - hc},$$

dont la solution est

$$y_n = \frac{1}{(1 - hc)^n}.$$

Lorsque $c < 0$, y_n décroît régulièrement vers 0 lorsque n croît, quelle que soit la valeur de $|c|$; à la différence de la variante explicite, le schéma d'Euler implicite est stable.

La convergence concerne l'erreur globale $|y(x_N) - y_N|$. On a le théorème suivant pour une méthode à k pas.

Théorème : condition de convergence d'une méthode multi-pas
Supposons que les k premières valeurs vérifient

$$|y(x_i) - y_i| \leq Ch^p, \quad i = 0, 1, \ldots, k-1.$$

Si, de plus, la méthode est stable et d'ordre p, alors elle est convergente d'ordre p ou encore

$$|y(x_n) - y_n| \leq C'h^p, \quad a = x_0 \leq x \leq x_N = b.$$

11.7. Méthodes pour les équations du second ordre

La physique fournit souvent des équations différentielles du second ordre où y' ne figure pas. C'est le cas en mécanique classique (au moins pour les systèmes à un seul degré de liberté) en l'absence de frottement ; c'est aussi le cas en mécanique quantique pour les problèmes à une dimension. Il existe plusieurs algorithmes spécialisés et bien adaptés à ce type de problème. Nous en décrirons deux.

11.7.1. Algorithme de Verlet ou de saute-mouton

Notons $y(t)$ la fonction inconnue, solution de l'équation différentielle particulière :

$$y'' = f(t, y) \tag{11.27}$$

où la variable indépendante peut être le temps. La méthode résulte du remplacement de la dérivée seconde par une approximation bien connue

$$y''|_n \cong \frac{1}{h^2}[y_{n+1} - 2y_n + y_{n-1}],$$

où l'erreur de troncation est $\mathcal{O}(h^2)$. L'algorithme de Stoermer–Verlet s'écrit donc

$$y_{n+1} = 2y_n - y_{n-1} + h^2 f(t_n, y_n). \tag{11.28}$$

Par suite de la symétrie de cette formule, les vitesses (et tous les termes impairs en h) ont disparu. En cas de besoin, nous pourrons estimer les vitesses au moyen de la relation

$$y'_n = v_n = \frac{1}{2h}[y_{n+1} - y_{n-1}].$$

Le schéma passe « par dessus » le point (t_n, y_n) d'où le nom de « saute-mouton » (« leapfrog » en anglais). Cet algorithme jouit de plusieurs avantages. Il est rapide, la seule étape coûteuse en temps étant le calcul du second membre, qui intervient une fois par pas. Le terme d'erreur est petit et une analyse plus détaillée montrerait que l'énergie totale est conservée à chaque pas, jusqu'aux termes en h^2. Il est de ce fait presque universellement employé pour les calculs de dynamique moléculaire où l'on doit suivre les trajectoires de milliers, voire de millions, d'atomes simultanément. Les bonnes performances de la méthode de Verlet en mécanique classique tiennent à la forme particulière des équations de la dynamique. Celles-ci s'écrivent, sous la forme donnée par Hamilton

$$\dot{p} = -\frac{\partial H}{\partial q} \quad ; \quad \dot{q} = \frac{\partial H}{\partial p}.$$

(q est une coordonnée généralisée et p une quantité de mouvement généralisée). Ce système différentiel est particulier : c'est la même fonction (l'énergie totale ou le hamiltonien H) qui figure au second membre. L'algorithme de Verlet respecte bien cette structure et on le qualifie de « symplectique » (entrelacé en grec). L'algorithme de Verlet est aussi un exemple « d'intégration géométrique ».

11.7.2. Algorithme de Numerov

C'est à un astronome russe (publiant vers 1935) que l'on doit une méthode d'intégration spécialement adaptée aux équations de la mécanique céleste et qui porte son nom. Nous cherchons à résoudre l'équation (11.27), mais avec une précision bien supérieure à celle permise par la méthode de Verlet.

L'algorithme de Numerov est une méthode de prédiction-correction. Intéressons nous tout d'abord à la partie « correction », que nous allons traiter par la méthode des coefficients indéterminés. Nous faisons l'hypothèse que y_{n+1} peut s'exprimer comme

$$y_{n+1} = a_0 y_n + a_1 y_{n-1} + a_2 y_{n-2} + h^2(b_{-1} f_{n+1} + b_0 f_n + b_1 f_{n-1} + b_2 f_{n-2}).$$

Le coefficient h^2 est là pour que les b_i soient des nombres sans dimensions. La présence d'un coefficient b_{-1} non nul indique qu'il s'agit d'une méthode implicite (y_{n+1} est donné en fonction de f_{n+1} qui dépend elle-même de y_{n+1}). Nous disposons de sept paramètres inconnus $\{a_i, b_i\}$ entre lesquels nous allons imposer six relations ; nous utiliserons à la fin les résultats de Numerov pour lever l'indétermination. Nous imposons donc que la formule précédente soit exacte pour $y = 1, x, x^2, x^3, x^4$ et x^5. Comme d'habitude, nous pouvons choisir $x = 0$, puisque les relations obtenues doivent être

vérifiées quel que soit x. Il vient :

$$\begin{cases} 1 = a_0 + a_1 + a_2, \\ h = -h(a_1 + 2a_2), \\ h^2 = h^2(a_1 + 4a_2) + 2h^2(b_{-1} + b_0 + b_1 + b_2), \\ h^3 = -h^3(a_1 + 8a_2) + 6h^3(b_{-1} - b_1 - 2b_2), \\ h^4 = h^4(a_1 + 16a_2) + 12h^4(b_{-1} + b_1 + 4b_2), \\ h^5 = -h^5(a_1 + 32a_2) + 20h^5(b_{-1} - b_1 - 8b_2). \end{cases}$$

Exprimons les coefficients en fonction de a_2

$$\begin{cases} a_0 = 2 + a_2 \quad , \quad a_1 = -1 - 2a_2, \\ b_{-1} = 1/12 \quad , \quad b_0 = (10 - a_2)/12, \\ b_1 = (1 - 10a_2)/12 \quad , \quad b_2 = -a_2/12. \end{cases}$$

Selon Numerov, le choix $a_2 = 0$ jouit de qualités intéressantes : c'est celui que nous retiendrons. La formule définitive (avec le terme d'erreur que nous pourrions obtenir à partir d'un développement de Taylor des y_i) s'écrit :

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{12}(f_{n+1} + 10f_n + f_{n-1}) - \frac{h^6 y^{(6)}(\xi_c)}{240}. \tag{11.29}$$

Pour être complet, citons la formule de prédiction établie par Numerov, que vous pouvez démontrer par la même méthode des coefficients indéterminés :

$$y_{n+1} = 2y_{n-1} - y_{n-3} + \frac{4h^2}{3}(f_n + f_{n-1} + f_{n-2}) + \frac{16h^6 y^{(6)}(\xi_p)}{240}$$

Vous pourrez aussi trouver l'erreur de troncation en fonction de $y^{[P]}$ et de $y^{[C]}$. La formule de correction est la plus précise et la plus stable des formules de Numerov ; aussi, nous la particulariserons pour le cas d'un problème linéaire, où l'équation différentielle devient :

$$y'' = A(x)y + B(x).$$

Cette hypothèse entraîne une simplification considérable, car le second membre est alors soluble en y et l'équation de correction n'est plus implicite. Nous pouvons donc abandonner l'étape de prédiction pour écrire :

$$y_{n+1} = \frac{1}{1 - \frac{h^2}{12}A_{n+1}}\{2y_n - y_{n-1} + \frac{h^2}{12}[B_{n+1} + 10(A_n y_n + B_n) + A_{n-1}y_{n-1} + B_{n-1}]\}. \tag{11.30}$$

Cet algorithme est facile à programmer et donne de très bons résultats.

11.8. ÉQUATIONS « RAIDES »

Avant de terminer ce chapitre, mentionnons une difficulté que l'on rencontre aussi bien en cinétique chimique qu'en dynamique des structures, sans offrir de solution concrète.

Certains problèmes de cinétique chimique ont pour expression mathématique un système différentiel de la forme :

$$\boldsymbol{y'} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{f}(t)$$

où $\boldsymbol{y}$ est un vecteur dont les n coordonnées représentent des concentrations, $\boldsymbol{A}$ une matrice carrée constante d'ordre n et $\boldsymbol{f}$ le vecteur de seconds membres. Nous supposons que les valeurs propres λ_k de $\boldsymbol{A}$ sont réelles et distinctes ; il leur correspond des vecteurs propres notés $\boldsymbol{z}_k$. Nous connaissons alors la forme de la solution :

$$\boldsymbol{y} = \sum c_k \exp(\lambda_k t)\boldsymbol{z}_k + \boldsymbol{\varphi}(t).$$

Du point de vue numérique, il se posera un problème de stabilité chaque fois qu'une valeur propre sera négative et le problème sera d'autant plus aigu que les λ_k seront plus différents les uns des autres. En termes qualitatifs, il nous faudra choisir un pas très petit pour traiter correctement les décroissances les plus rapides (grand λ) mais, avec ce même pas, il faudra un nombre immense d'itérations pour voir évoluer les décroissances les plus lentes. On dit que le système d'équations différentielles est "raide".

Un problème analogue se rencontre pour des systèmes du second ordre

$$\boldsymbol{y''} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{f}(t)$$

comme on en trouve en mécanique des vibrations. Ici, l'échelle de temps est en principe définie par la période du mouvement, mais il y a plusieurs périodes, puisque la matrice $\boldsymbol{A}$ a plusieurs valeurs propres, lesquelles peuvent être très différentes. Nous serons encore une fois obligés de choisir un pas très petit pour suivre le mouvement le plus rapide et, simultanément, de prolonger le calcul pour décrire le mouvement le plus lent.

Dans la pratique, la difficulté est souvent accrue par la non-linéarité des seconds membres. Seules des méthodes implicites sont assez stables pour traiter ce type de problèmes. La convergence de la méthode d'itération décrite au § 11.5.3 n'est pas assurée dans ce cas et il faut résoudre l'équation implicite du « correcteur » par une méthode de Newton.

11.9. Résoudre une équation différentielle en dormant

Les logiciels de « haut niveau » comme Scilab ou Maple proposent des programmes « tous faits » de résolution des équations ou des systèmes différentiels. Commençons par décrire la fonction `ode` de Scilab appliquée au cas du pendule mathématique amorti et soumis à un couple extérieur. Comme nous tous, Scilab ne connaît en principe que les systèmes du premier ordre ; il faut donc commencer par transformer l'équation en posant $y' = z$ pour obtenir

$$\begin{cases} y' = z, \\ z' = -\sin y - rz + g\cos\Omega t. \end{cases}$$

La fonction `ode`, dans sa version de base, demande quatre arguments : un vecteur de conditions initiales, une valeur initiale de la variable indépendante (le temps ici), les valeurs du temps pour lesquelles on veut connaître le déplacement y et une fonction calculant les seconds membres. Nous choisissons la notation $y = u(1), z = u(2), y(0) = u0(1), z(0) = u0(2)$. Le squelette du programme est alors

```
function [uprime] = pdl(t,u)
uprime = [u(2),-sin(u(1))- r*u(2) + g*cos(omg*t)]
endfunction
//
//lecture des paramètres et des conditions initiales
//
t0 = 0; t = 0:0.05:tmax;
u = ode(u0, t0, t,  pdl);
plot2d(u(1,:),u(2,:))
```

Le résultat apparaît dans la matrice `u` dont la première ligne contient le déplacement et la deuxième la vitesse angulaire du pendule. Nous avons choisi de représenter la trajectoire dans le « plan de phase » de coordonnées (position, vitesse) (figure 11.6).

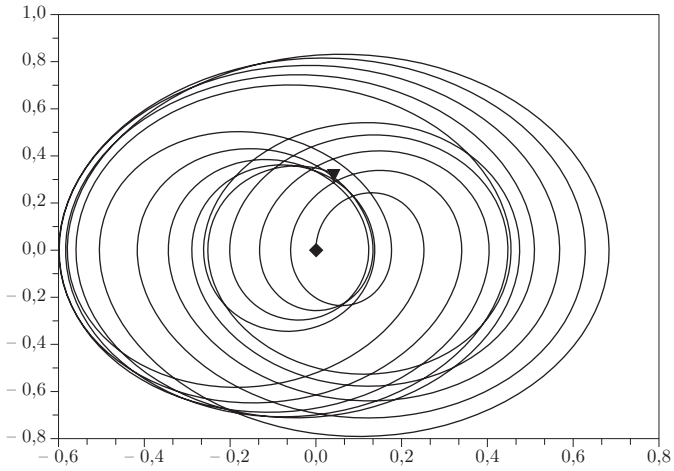


Figure 11.6 – Mouvement du pendule entretenu ; le mouvement propre disparaît au profit du mouvement forcé. Les conditions initiales étaient $y_0 = z_0 = 0$.

`ode` utilise « par défaut » le code `lsoda` de la bibliothèque ODEPACK (disponible aussi sur Netlib). Ce programme perfectionné choisit automatiquement un schéma de prédiction-correction pour les cas « simples » et un schéma de « différenciation vers l'arrière » pour les équations raides. Vous pouvez aussi demander l'utilisation d'une méthode de Runge–Kutta ou de Runge–Kutta–Fehlberg en ajoutant, comme cinquième paramètre, la chaîne de caractères `"rk"` ou `"rkf"`.

Essayons maintenant de parvenir au même résultat à l'aide de Maple. Ce logiciel connaît les équations différentielles d'ordre quelconque. On peut lui demander une

solution analytique ou une solution numérique; il faut alors accepter une démarche un peu tortueuse pour afficher les résultats.

```
> omega := 3.0; Te := 2*Pi/omega;
```

$$\omega := 3.0$$
$$Te := 0.6666666666\,\pi$$

```
> Q := 10;
```

$$Q := 10$$

```
> A := 22.5;
```

$$A := 22.5$$

```
> pdl := diff(y(t),t$2)+(1/Q)*diff(y(t),t)+sin(y(t)) =
> A*sin(omega*t);
```

$$pdl := (\frac{d^2}{dt^2}\,\mathrm{y}(t)) + \frac{1}{10}\,(\frac{d}{dt}\,\mathrm{y}(t)) + \sin(\mathrm{y}(t)) = 22.5\sin(3.0\,t)$$

```
> ci := y(0) = 3.1, D(y)(0) = -7.1;}
```

$$ci := \mathrm{y}(0) = 3.1,\ \mathrm{D}(y)(0) = -7.1$$

```
> sol := dsolve(\{pdl,ci\},y(t),type=numeric);
```

$$sol := \mathbf{proc}(x\_rkf45)\ \ldots\ \mathbf{end\ proc}$$

Par défaut, Maple utilise l'algorithme de Runge–Kutta–Fehlberg d'ordre 4-5. Maple renvoie une « procedure », dont le résultat est une liste : valeur de t, valeur de y, valeur de y' :

```
> sol(0);
```

$$[t = 0.,\ \mathrm{y}(t) = 3.1000000000000,\ \frac{d}{dt}\,\mathrm{y}(t) = -7.1000000000000]$$

```
> sol(1.23);
```

$$[t = 1.23,\ \mathrm{y}(t) = 4.47563587637383442,\ \frac{d}{dt}\,\mathrm{y}(t) = 5.99137941724169742]$$

On transforme le second membre (`rhs`) de sol en une fonction ; en fait, c'est le second opérande (y) qui est intéressant

```
> Y := t -> rhs( op(2,sol(t)) );
```

$$Y := t \to \mathrm{rhs}(\mathrm{op}(2,\ \mathrm{sol}(t)))$$

On fait la même chose pour le troisième opérande (y') :

```
> YP := t -> rhs( op(3,sol(t)) );
```

$$YP := t \to \mathrm{rhs}(\mathrm{op}(3,\ \mathrm{sol}(t)))$$

On crée une structure (`pl1`) contenant une première courbe, la position du pendule au cours du temps ; les «deux points» empèchent l'affichage à l'écran d'une myriade de coordonnées.

```
> pl1 := plot(Y/evalf(Pi),0..30, thickness=2):
```

La deuxième structure (`pl2`) représentera l'excitation, normalisée.

```
> pl2 := plot(sin(omega*t)/4,t=0..30,linestyle = DASH,color=grey):
```

On appelle la bibliothèque graphique :

```
> with(plots):
```

et on affiche les deux courbes

```
> display(pl1,pl2);
```

Quelle est la position moyenne du pendule au cours de cette simulation (voir figure 11.7) ?

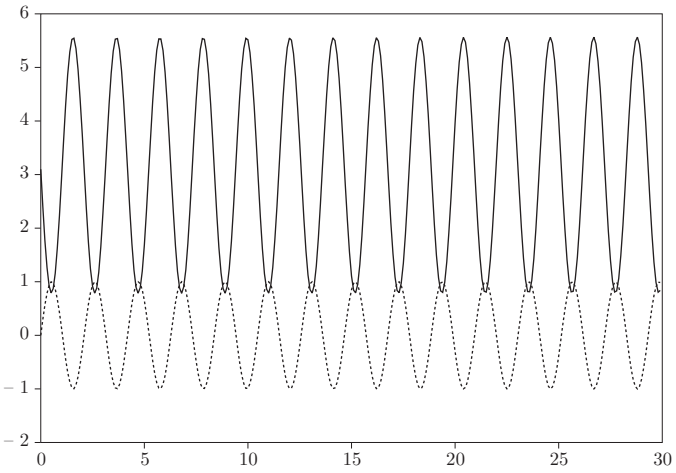


Figure 11.7 – Mouvement du pendule entretenu autour d'un « attracteur ». Le couple appliqué est représenté en tirets.

Si le schéma utilisé n'est pas à votre goût, vous pouvez en choisir un autre (parmi la vingtaine de méthodes disponibles) ; il suffit d'ajouter à la liste d'arguments la chaîne de caractères `'method=classical[rk2]'` par exemple.

Il ne faut pas croire que le recours à un programme « tout fait » vous met automatiquement à l'abri des erreurs d'arrondi ou des instabilités. Les explications très claires et les nombreux exemples instructifs que vous trouverez sur le site du Professeur Sallet vous prouveront le contraire.

11.10. Pour en savoir plus

– R. Théodor : *Initiation à l'analyse numérique*, ch. 7 (Masson, Paris, 1994).

– M. Schatzman : *Analyse numérique, une approche mathématique*, ch. 16 (Dunod, Paris, 2001).

– Polycopiés des cours d'analyse numérique de MM. E. Hairer et G. Wanner : ch. 3, équations différentielles ordinaires :
`http://www.unige.ch/~hairer/polycop.html`

– `http://www.math.univ-metz.fr` : voir page personnelle de G. Sallet, en particulier son cours sur la résolution des équations différentielles ordinaires avec Scilab.

– J.P. Demailly : *Analyse numérique et équations différentielles*, ch. 5–11 (EDP, Grenoble Sciences, 2006).

– W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery : *Numerical recipes, the art of scientific computing*, ch. 16 (Cambridge University Press, Cambridge, 2007).

– S. Guerre–Delabrière et M. Postel : *Méthodes d'approximation – Équations différentielles – Applications Scilab – Niveau L3* (Ellipses, Paris, 2004).

11.11. Exercices

Exercice 1

Pour résoudre le problème de Cauchy (11.1), il existe une catégorie de méthodes qui s'écrivent en général

$$y_{n+1} = a_0 y_n + a_1 y_{n-1} + h[b_0 f(x_n, y_n) + b_1 f(x_{n-1}, y_{n-1})] \tag{11.31}$$

On impose que cette relation soit exacte pour $y = 1, x$ et x^2. Quelles doivent être alors les expressions de a_0, a_1 et b_1 en fonction de b_0 ? À quelle valeur de b_0 correspond la méthode du point milieu ?

Exercice 2

On suppose que le second membre de l'équation différentielle type $y' = f[x, y(x)]$ ne dépend pas de y ; elle s'écrit donc $y' = f(x)$.

a) Écrire, dans ce cas particulier, les deux algorithmes de Runge et Kutta d'ordre 2 (Euler amélioré et Euler modifié).

b) L'équation différentielle peut être intégrée terme à terme entre les abscisses x_n et $x_{n+1} = x_n + h$:

$$y(x_n + h) - y(x_n) = \int_{x_n}^{x_n+h} f(u, y(u))du.$$

On ne sait pas, en général, calculer l'intégrale du second membre mais on peut en trouver une valeur approchée à l'aide de l'une des méthodes proposées au chapitre

8. Montrer que, dans le cas particulier où f ne dépend pas de y, les algorithmes écrits en (a) sont identiques à des méthodes d'intégration classiques. Lesquelles ?

Exercice 3

Pour étudier la stabilité de la méthode du point milieu, on l'applique au problème modèle

$$y' = Ay \quad \text{avec} \quad y(0) = 1,$$

où A est une constante.

a) Montrer que l'on obtient alors une relation de récurrence linéaire entre trois valeurs successives de y.

b) Montrer que $y_n = r^n$ satisfait cette relation, à condition que r vérifie une « équation caractéristique » que l'on précisera. On note r', r'' les valeurs possibles de r.

c) La solution la plus générale de l'équation de récurrence, notée encore y_n, dépend de deux constantes arbitraires, c' et c''

$$y_n = c'r'^n + c''r''^n.$$

Exprimer c', c'' en fonction des valeurs initiales y_0 et y_1.

d) On suppose que y_0 et y_1 coïncident avec les valeurs exactes 1 et e^{Ah}. Trouver des expressions approchées de c' et c'' valables jusqu'aux termes en h^2 compris. Quelle est la partie de la solution générale qui correspond le mieux à la solution exacte du problème différentiel ?

e) Résoudre numériquement, par la méthode du point milieu, le problème différentiel pour $A = -1$, $h = 0,3$ et $0 \leq x \leq 2,1$. y_1 sera calculé par l'algorithme d'Euler. Comparer avec la solution exacte.

Exercice 4

a) Adapter l'algorithme d'Euler au cas d'une équation différentielle du second ordre. Donner en particulier les équations nécessaires pour résoudre numériquement le problème du mouvement d'un pendule réel soumis à un couple dépendant du temps :

$$x'' + k^2 \sin x = g(t).$$

b) On donne $g(t) = \sin 3t, k = 2, x(0) = 0, x'(0) = 1$ et $h = 0,2$. Calculer numériquement x_1, x_2 et x_3.

Exercice 5

Pour analyser les performances de l'algorithme d'Euler, on l'applique au problème modèle

$$x'' + \omega^2 x = 0.$$

a) Vérifier que cette équation décrit le mouvement d'un oscillateur d'énergie cinétique $T = \frac{1}{2}x'^2$ et d'énergie potentielle $V = \frac{1}{2}\omega^2 x^2$.

b) Calculer l'énergie totale E_{n+1} de l'oscillateur, donnée par l'algorithme d'Euler à la date t_{n+1}. Comparer le résultat à l'énergie E_n au temps t_n. Peut-on trouver une valeur du pas h telle qu'il y ait conservation de l'énergie calculée ?

Exercice 6

Un pendule rigide est constitué d'une masse ponctuelle liée par une tige sans masse à un pivot situé à l'origine O. L'ensemble masse + tige peut tourner sans frottement dans un plan vertical. On appelle x l'angle que fait le pendule avec la verticale descendante. Le système est caractérisé par son énergie cinétique $E_c = x'^2/2$ et son énergie potentielle $E_p = 1 - \cos x$, ou par son équation du mouvement

$$x'' + \sin x = 0.$$

On a pris égales à l'unité la longueur, la masse et l'accélération de la pesanteur.

a) Écrire un programme pour résoudre numériquement l'équation du mouvement, à l'aide de l'algorithme

$$x_{n+1} = 2x_n - x_{n-1} + h^2 \sin x_n.$$

b) Tracer $x(t)$ et $v(t) = x'(t)$ sur un même graphique. Représenter aussi $v(t)$ comme fonction de $x(t)$ sur un autre graphique.

c) Calculer, pour chaque instant, l'énergie cinétique E_c, l'énergie potentielle E_p et l'énergie totale $E_t = E_c + E_p$. Représenter ces trois grandeurs simultanément en fonction du temps.

Exercice 7

Une autre façon d'étudier l'algorithme d'Euler consiste à analyser l'évolution des valeurs de x_n et de x'_n pour le problème modèle

$$x'' + \omega^2 x = 0.$$

a) Montrer le vecteur $\vec{r}_n = [x_n, x'_n]^T$ se déduit de $\vec{r}_{n-1} = [x_{n-1}, x'_{n-1}]^T$ par la relation :

$$\vec{r}_n = \boldsymbol{M}\vec{r}_{n-1}$$

où $\mathbf{M}$ est une matrice dont on précisera les éléments.

b) Utiliser les valeurs propres de $\mathbf{M}$ pour prédire la stabilité de l'algorithme.

Exercice 8

Dans le cas où l'équation différentielle à résoudre est du second ordre mais ne fait pas intervenir de dérivée première, on propose l'algorithme suivant

$$\begin{aligned} v_{n+1/2} &= v_{n-1/2} + hf(x_n, t_n), \\ x_{n+1} &= x_n + hv_{n+1/2}, \end{aligned} \qquad \text{(L)}$$

où on a posé $x' = v$. La première valeur de la vitesse, $v_{1/2} = x'_{1/2}$ est calculée par l'algorithme d'Euler : $v_{1/2} = v_0 + (h/2)f(x_0, t_0)$.

a) Montrer que ces relations sont équivalentes à l'algorithme :

$$x_{n+1} = 2x_n - x_{n-1} + h^2 f(x_n, t_n).$$

b) Justifier directement cette dernière formule.

c) Utiliser l'algorithme (L) pour résoudre le problème différentiel de l'exercice 4. On calculera à nouveau x_1, x_2 et x_3.

Exercice 9

On revient au problème modèle

$$x'' + \omega^2 x = 0$$

abordé par l'algorithme (L) de l'exercice 7. On définit une quantité analogue à une énergie :

$$\varepsilon' = \frac{1}{2}(v_{n-1/2}^2 + \omega^2 x_{n-1} x_n).$$

a) Montrer que cette quantité se conserve lors d'une itération.

b) Même question pour

$$\varepsilon'' = \frac{1}{2}(v_{n-3/2} v_{n-1/2} + \omega^2 x_{n-1}^2).$$

c) Quelle relation existe-t-il entre ε' et ε'' ?

d) À chaque itération de l'algorithme, on peut associer un vecteur $\vec{r}_n = [x_n, v_{n-1/2}]^T$. Montrer que, dans le cas du problème modèle, les équations (L) se mettent sous la forme :

$$\vec{r}_n = \boldsymbol{N} r_{n-1}$$

et préciser les éléments de $\mathbf{N}$. Utiliser les valeurs propres de $\mathbf{N}$ pour étudier la stabilité de l'algorithme.

Exercice 10

Les intégrales de Fresnel

$$C(x) = \int_0^x \cos\left(\frac{\pi t^2}{2}\right) dt \qquad ; \qquad S(x) = \int_0^x \sin\left(\frac{\pi t^2}{2}\right) dt$$

interviennent dans la théorie de la diffraction.

a) Montrer que le calcul de ces intégrales indéfinies peut être remplacé par la résolution d'un problème différentiel à condition initiale que l'on détaillera.

b) pour résoudre numériquement le problème de forme générale

$$y' = f(x, y) \quad ; \quad y(a) = A,$$

on peut faire appel à l'algorithme

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n-1} + 4f_n + f_{n+1}).$$

Justifier cette formule et expliquer comment cette méthode peut servir à calculer $C(x)$ et $S(x)$.

c) Trouver les valeurs nécessaires au démarrage par un développement de Taylor à l'ordre 4.

d) Calculer, avec un pas $h = 0,2$, $C(0{,}6)$ et $S(0{,}6)$.

e) Rédiger un programme destiné à calculer S et C pour toute valeur de l'argument. Tracer la courbe d'équations paramétriques $x = C(t), y = S(t), -10 \leq t \leq 10$ (spirale de Cornu).

Exercice 11

On s'intéresse au mouvement d'une planète autour d'un centre fixe. Celui-ci est décrit par le système différentiel

$$\boldsymbol{r''} = \frac{k}{r^3}\boldsymbol{r}.$$

Le vecteur position (sans dimensions) $\mathbf{r}$ et sa dérivée, le vecteur vitesse $\boldsymbol{r'}$, ont respectivement comme coordonnées (x, y) et (u, v) ; on a posé $r = |\boldsymbol{r}|$.

a) Écrire les équations qui permettent de calculer les coordonnées de la planète à l'aide de l'algorithme d'Euler.

b) Écrire les équations qui déterminent le mouvement selon la méthode du point milieu.

11.12. Projets

Projet 1. Pendule élastique

Un pendule élastique est constitué d'une masse m suspendue à un ressort à spires non jointives (c'est-à-dire capable d'exercer une force proportionnelle à la variation de sa longueur à l'allongement aussi bien qu'à la compression) de raideur k. La masse peut osciller autour de sa position d'équilibre dans un plan vertical. Il y a deux types de mouvement particulièrement simples : l'oscillation le long de la verticale et un mouvement pendulaire de part et d'autre de la verticale. On étudie dans ce projet le mouvement général.

1. Écrire les équations différentielles du mouvement, en utilisant les coordonnées polaires r (longueur du ressort) et θ (écart angulaire par rapport à la verticale descendante) et le formalisme de Lagrange ou celui de Hamilton. On note r_0 la longueur du ressort isolé.

2. Rédiger un programme pour résoudre numériquement les équations du mouvement par une méthode d'ordre 4. Représenter graphiquement les variations de $r(t), \theta(t)$

en fonction du temps, la trajectoire (r fonction de θ ou y fonction de x) ainsi que le comportement des énergies potentielle, cinétique et totale.

3. Vérifier que le programme donne des résultats plausibles pour $\dot{r}(0) = \dot{\theta}(0) = 0$ (mouvement vertical) et pour k très grand et $\dot{r}(0) = 0$ (mouvement pendulaire).

4. Faire fonctionner le programme pour divers choix de paramètres et de conditions initiales. Les valeurs suivantes donnent lieu à une résonance intéressante entre mouvement vertical et mouvement pendulaire :
$$k = 17{,}8 \text{ N/m}, \ m = 0{,}2 \text{ kg}, \ r_0 = 0{,}44 \text{ m}, \ \text{g} = 9{,}8 \text{ m/s}^2$$
avec les conditions initiales
$$r(0) = 0{,}66 \text{ m}, \ \theta(0) = 0{,}03 \text{ rd}, \text{ vitesses initiales nulles.}$$

5. Généraliser le programme pour traiter des mouvements à trois dimensions

Projet 2. Pendule double

Un pendule double est constitué d'une masse m_1 liée à un pivot O par une tige rigide de longueur ℓ_1, de masse nulle ; une masse m_2 est suspendue à m_1 par une tige rigide sans masse de longueur ℓ_2. L'ensemble peut osciller dans un plan vertical. On suppose que les pivots sont construits de façon à permettre à la masse 1 de faire le tour de O et à la masse 2 de faire le tour de m_1. Il n'y a aucun frottement. On appelle θ_1 et θ_2 respectivement l'angle que fait chaque tige avec la verticale descendante, qui sera prise comme axe des x ; l'axe des y est horizontal, orienté vers la droite.

1. Mise en équations.

   a) Écrire les coordonnées x_i, y_i, $i = 1, 2$ de chaque masse en fonction des angles θ_i.

   b) En déduire les coordonnées des vitesses de chaque masse, $\dot{x}_i, \dot{y}_i$, qui feront intervenir les vitesses angulaires $\dot{\theta}_i$. On choisit les θ_i comme coordonnées généralisées.

   c) Exprimer, en fonction des θ_i et des $\dot{\theta}_i$, l'énergie cinétique T, l'énergie potentielle V et la fonction de Lagrange du système.

   d) Calculer les dérivées
$$\frac{\partial L}{\partial \theta_1}, \quad \frac{\partial L}{\partial \theta_2}, \quad \frac{\partial L}{\partial \dot{\theta}_1}, \quad \frac{\partial L}{\partial \dot{\theta}_2}.$$

   e) Vérifier que les équation de Lagrange de ce système s'écrivent
$$\begin{cases} (m_1+m_2)\ell_1^2\ddot{\theta}_1 + m_2\ell_1\ell_2\ddot{\theta}_2\cos(\theta_1-\theta_2) + m_2\ell_1\ell_2\dot{\theta}_2^2\sin(\theta_1-\theta_2) \\ \qquad\qquad +(m_1+m_2)g\ell_1\sin\theta_1 = 0 \\ m_2\ell_2^2\ddot{\theta}_2 + m_2\ell_1\ell_2\ddot{\theta}_1\cos(\theta_1-\theta_2) - m_2\ell_1\ell_2\dot{\theta}_1^2\sin(\theta_1-\theta_2) + m_2g\ell_2\sin\theta_2 = 0 \end{cases}$$

2. On demande d'écrire un programme pour résoudre numériquement les équations du mouvement. Comme ce système est très sensible aux erreurs numériques, on emploiera la méthode RK4. Pour pouvoir faire une étude complète, il faut une simulation assez longue. Pour cela, il est commode d'écrire les résultats sur disque. Pour la même raison, on est amené à ne retenir qu'un résultat tout les p (avec une période d'échantillonnage $p \simeq 10$). Vérifiez la conservation de l'énergie.

3. On suppose encore que les masses sont égales, que $\ell_1 = \ell_2 = 0,2$m et que $g = 9,8$ m/s^2. Représenter graphiquement, au choix de l'utilisateur, les valeurs des angles en fonction du temps, l'énergie potentielle de chaque masse, la trajectoire, dans l'espace x_2, y_2 de la masse 2.

4. Il est aussi très parlant de créer une « section de Poincaré » ; il s'agit de représenter, dans le plan $(\theta_2, \dot{\theta}_2)$, la position et la vitesse angulaire du pendule 2 quand le pendule 1 passe à la verticale (ou réciproquement). Les temps de passage doivent être calculés par interpolation entre deux positions encadrant l'origine ; de même, les positions du deuxième pendule doivent être interpolées.

Projet 3. Trajectoires de particules chargées dans le champ magnétique terrestre

Le champ magnétique de la terre peut être assimilé à celui d'un dipole situé au centre de la planète et aligné avec l'axe nord-sud. On se propose de calculer les trajectoires de particules chargées émises par le soleil dans le champ magnétique terrestre, particules qui sont responsables des aurores boréales. Le champ créé par un dipole magnétique peut s'écrire

$$\vec{B} = \frac{\mu_0}{4\pi} \frac{3(\vec{r} \cdot \vec{m})\vec{r} - r^2 \vec{m}}{r^5} \tag{11.32}$$

où $\vec{m}$ désigne le moment magnétique du dipôle et $\vec{r}$ le vecteur qui joint le dipôle au point d'observation.

En France, $\| B \| = 46600\gamma = 46600 \times 10^{-9}$ Tesla, à une distance de $R_e = 6380$ km du centre de la terre. Que vaut m_z ?

1. Écrire les équations du mouvement d'une charge q dans le champ du dipôle.

2. Rédiger un programme pour résoudre numériquement ces équations par une méthode d'ordre 4. Calculer l'énergie de la particule et vérifier qu'elle reste approximativement constante.

3. Les particules, des électrons de masse $m_e = 9,11 \times 10^{-31}$ kg et de charge $q_e = -1,6 \times 10^{-19}$ C, ont des énergies comprises entre 100 et 1000 MeV. Leur vitesse initiale et leur masse doivent donc être calculées à partir de formules relativistes. Représenter graphiquement quelques trajectoires. Les conditions initiales $x_0 = z_0 = 0, y_0 = 0,2 \times 10^8$ m, $v_{x0} = 0, v_{y0} = -10^8$ m/s conduisent à des trajectoires intéressantes (v_{z0} est calculée à partir de l'énergie et de v_{y0}).

4. Qu'arrive-t-il si la vitesse initiale de l'électron est dirigée selon Oz, selon Oy ou parallèlement à l'équateur ?

5. Quelle énergie totale faut-il pour qu'un électron s'échappe du champ magnétique terrestre ? Et un proton ?

This page intentionally left blank

Chapitre 12

Problèmes à conditions aux limites et problèmes aux valeurs propres

La physique fournit de nombreux exemples de problèmes différentiels qui prennent un aspect différent du problème de Cauchy (ou « à conditions initiales ») examiné dans le chapitre 11. Il s'agit soit de problèmes à « conditions aux limites », soit de problèmes « aux valeurs propres ». Les méthodes employées pour résoudre ces deux types de questions sont assez proches et certaines se déduisent des algorithmes utilisés pour résoudre le problème de Cauchy.

Voici un exemple du premier type. Une poutre horizontale de longueur L, soumise à une charge uniforme $q(N/m)$ et à une tension S à chaque extrémité, se déforme verticalement ; appelons $w(x)$ le déplacement d'un point x de la poutre à partir de l'horizontale. w obéit à l'équation différentielle :

$$w'' - \frac{S}{EI}w = \frac{1}{2EI}qx(x-L),$$

où I est le moment d'inertie de la section droite de la poutre et E le module d'élasticité. Si la poutre repose sur des supports fixes à chaque extrémité, w doit aussi satisfaire aux conditions aux limites :

$$w(0) = w(L) = 0.$$

La résolution analytique est facile si les paramètres sont constants, mais une méthode numérique devient indispensable dès lors que la charge q ou les propriétés (E, I) de la poutre varient avec x.

Remarque : Vous avez remarqué que l'exemple qui vient d'être décrit ne constitue pas vraiment un problème à une dimension : la poutre se déforme dans la direction perpendiculaire à sa longueur. À une certaine approximation, on peut néanmoins considérer que les grandeurs physiques qui interviennent ne dépendent que de l'abscisse. Cet exemple a une portée générale, car nous vivons dans un univers à trois dimensions et nous sommes toujours tentés de réduire le nombre de dimensions pour simplifier le problème. Une corde vibrante ou un tuyau sonore sont eux aussi décrits, de façon approchée, comme des systèmes à une seule dimension d'espace.

Il est facile de généraliser le modèle précédent. Nous supposons avoir affaire à une équation résolue en y'' (ou résolue par rapport à la dérivée d'ordre le plus élevé), avec des conditions aux limites « séparées » sur y (le cas de conditions sur y' est très semblable) :

$$y'' = f(x, y, y'); \quad y(a) = A; \quad y(b) = B; \quad a \leq x \leq b.$$

On rencontre plus rarement des conditions aux limites portant sur y **et** y' :

$$c_0 y(a) + c_1 y'(a) = A; \quad d_0 y(b) + d_1 y'(b) = B.$$

Il peut arriver qu'une condition concerne les deux extrémités simultanément : $uy(a) + vy(b) = g$ avec, comme cas particulier, la « condition aux limites périodique » $y(a) = y(b)$. Nous n'examinerons pas ces derniers cas.

Vous savez qu'une équation différentielle peut être homogène, comme celle-ci

$$A(x)y'' + B(x)y' + C(x)y = 0$$

ou inhomogène comme l'équation

$$A(x)y'' + B(x)y' + C(x)y = D(x).$$

De même, les conditions aux limites peuvent être homogènes ($y(0) = 0, y'(1) = 0$) ou inhomogènes ($y(0) = y_0, y'(1) = y_1$). Un problème avec conditions aux limites est dit homogène si l'équation différentielle **et** les conditions aux limites le sont ; autrement, il est qualifié d'inhomogène. Les problèmes à conditions aux limites n'ont pas toujours de solution ; ils peuvent aussi en avoir plusieurs, un comportement que l'équation différentielle très simple

$$w'' + w = 0$$

illustre bien. La solution générale s'écrit $w(x) = c_1 \cos(x) + c_2 \sin(x)$. Selon les conditions aux limites choisies, le problème différentiel a une, une infinité ou zéro solution :

$w(0) = 0$	$w(\pi/2) = 1$	$w(x) = \sin x$	une solution
$w(0) = 0$	$w(\pi) = 0$	$w(x) = c_1 \sin x$	infinité de solutions
$w(0) = 0$	$w(\pi) = 1$		aucune solution

L'étude des mouvements périodiques d'une corde vibrante de masse linéique $\mu(x)$, soumise à une tension $T(x)$ fait apparaître un problème de valeurs propres. Après séparation des variables x et t, la déformation de la corde, $y(x)$, obéit à l'équation différentielle :

$$[T(x)y']' + (2\pi\nu)^2 \mu(x) y = 0, \tag{12.1}$$

où ν est la fréquence. y doit aussi respecter des conditions aux limites comme par exemple

$$y(0) = y(L) = 0$$

(extrémités fixes) ou $\partial y/\partial x|_{x=0} = 0$ (extrémité libre). Il y a deux inconnues : la fréquence ν (on parle souvent de fréquence propre) et la forme de la corde, $y(x)$. Un couple fréquence/forme constitue un mode de vibration.

Nous pouvons facilement imaginer une formulation plus générale de ce type de problème. Il faut résoudre une équation différentielle du second ordre (peut-être d'ordre plus élevé, mais presque toujours d'ordre pair), résolue en y'' et dépendant d'un paramètre, tout en respectant deux conditions aux limites

$$y'' = f(x, y, y', \lambda); \quad y(a) = y(b) = 0; \quad a \leq x \leq b.$$

Bien souvent, et c'est le seul cas que nous considérerons, les conditions aux limites sont homogènes, de même que l'équation différentielle. De plus, l'équation différentielle peut être linéaire en y (c'est le cas le plus fréquent en physique) ou non-linéaire. Le cas linéaire homogène général peut s'écrire

$$\begin{aligned} &y'' + p(x, \lambda)y' + q(x, \lambda)y = 0, \\ &a_0 y(0) + b_0 y'(0) = 0, \quad a_1 y(1) + b_1 y'(1) = 0. \end{aligned}$$

Cette forme est encore trop générale pour que l'on puisse se prononcer sur l'existence de solutions ; heureusement, la plupart des problèmes aux valeurs propres rencontrés en pratique font intervenir linéairement la valeur propre ; on parle alors de problème de Sturm-Liouville lequel s'écrit conventionnellement

$$[p(x)y']' - q(x)y + \lambda r(x)y = 0, \tag{12.2}$$
$$a_0 y(0) + b_0 y'(0) = 0, \quad a_1 y(1) + b_1 y'(1) = 0. \tag{12.3}$$

L'équation de la corde vibrante (12.1) est de cette forme, comme souvent après séparation des variables. Le problème ainsi posé n'a de solution que pour certaines valeurs de λ (les valeurs propres) ; les fonctions y correspondantes sont les fonctions propres. Pour toute autre valeur de λ, il n'y a que la solution « banale » $y = 0$ (on dit aussi « triviale », comme en anglais), sans intérêt physique.

Voyons cela sur l'exemple très simple

$$y'' + \lambda^2 y = 0, \quad y(0) = y(1) = 0.$$

Les solutions sont de la forme

$$y_p = \sin \lambda x, \qquad \text{si } \lambda = p\pi.$$

Il y a une infinité dénombrable de solutions correspondant à tous les entiers naturels $p \neq 0$.

12.1. La méthode du tir

12.1.1. Problème aux limites

Soit à résoudre le problème différentiel à conditions aux limites :

$$y'' = f(x, y, y'); \quad y(a) = A; \quad y(b) = B. \tag{12.4}$$

Nous associons à ce premier problème un problème à valeurs initiales :

$$w'' = f(x, w, w'); \quad w(a) = A; \quad w'(a) = s \tag{12.5}$$

qui admet en général une solution unique, dépendante du paramètre s, $w(x,s)$. Par hypothèse, ce problème n'est pas soluble analytiquement et nous devons donc en déterminer numériquement la solution pour une valeur donnée de s, par l'une des méthodes exposées au chapitre 11. Comme il s'agit de résoudre une équation différentielle du second ordre, il faudra sans doute la remplacer par un système différentiel équivalent. Pour résoudre ensuite le problème aux limites, connaissant $w(x,s)$, il nous suffit de trouver s tel que :

$$w(b,s) = B$$

ce qui revient à résoudre une équation non linéaire en s. Pour les personnes qui n'auraient eu l'avantage de recevoir un formation militaire, nous allons développer l'analogie avec le comportement d'un artilleur. A l'aide d'un canon situé en $x = a$, celui-ci veut atteindre une cible en $x = b$; il ne dispose pour cela que d'un seul réglage, la hausse (angle que fait l'axe du canon avec l'horizontale). Il procède par approximations successives : un coup court, un coup long (il encadre l'objectif) et, idéalement (?), un coup au but. Pour la résolution pratique de problèmes aux limites, il est recommandé de s'inspirer de l'artilleur, en traçant diverses fonctions $w(x,s)$ pour différentes valeurs de s et en observant leur comportement. Ce n'est qu'une fois acquise une certaine expérience du problème que l'on se tournera vers des méthodes plus systématiques.

Nous avons présenté, au chapitre 5, de nombreuses méthodes de résolution d'une équation non-linéaire ; la plus simple est la méthode de bissection (ou dichotomie). Sa mise en oeuvre suppose que nous disposons de deux valeurs de s telles que $w(b,s_1)$ et $w(b,s_2)$ encadrent B ou encore que $F(s_1) \equiv w(b,s_1) - B$ et $F(s_2) \equiv w(b,s_2) - B$ soient de signes opposés. Nous calculons alors $F(s_m) \equiv w(b,s_m)$ où $s_m = (s_1+s_2)/2$; selon le signe de $F(s_m)$, s_m remplace s_1 ou s_2 et l'algorithme se poursuit jusqu'à convergence. D'après un théorème connu sur les équations différentielles, la solution $w(x,s)$ est une fonction continûment dérivable de s. Nous pouvons donc aussi utiliser la méthode de Newton. Rappelons que, à partir d'une valeur initiale $s^{(0)}$, nous calculons, de façon itérative, des valeurs $s^{(i)}$ à l'aide de la formule :

$$s^{(i+1)} = s^{(i)} - \frac{F[s^{(i)}]}{F'[s^{(i)}]} \quad ; \quad F(s) = w(b,s) - B.$$

Nous obtenons $F[s^{(i)}]$ en résolvant le problème différentiel :

$$w'' = f(x,w,w'); \quad w(a) = A; \quad w'(a) = s^{(i)}.$$

La méthode la plus simple pour calculer F' consiste à l'approcher par le quotient :

$$F'[s^{(i)}] \cong \{F[s^{(i)} + h] - F[s^{(i)} - h]\}/(2h).$$

Nous avons supposé que l'algorithme qui détermine w partait de a et progressait vers les x croissants jusqu'en b. Ce n'est pas obligatoire et c'est même suicidaire si w a un point singulier en $x = a$; dans ce cas, on peut très bien partir de b, avec le même algorithme et un pas négatif. On peut encore être amené à construire un morceau de solution à partir de a, un autre à partir de b : il faut alors imposer la continuité de w ou de w' au point de rencontre quelque part entre a et b. Il est souvent commode d'imposer que la quantité w'/w (la « dérivée logarithmique ») soit continue ; cette condition a l'avantage d'être insensible à la normalisation de w.

Le cas d'une équation différentielle linéaire, fréquent en physique, est nettement plus simple. Il suffit en effet de déterminer deux solutions $w(x, s_1)$ et $w(x, s_2)$ du problème à conditions initiales associé. La bonne valeur de s, soit s_0, pour laquelle $w(b, s_0) = B$, s'obtient par interpolation linéaire. En effet, la solution du problème différentiel est une fonction linéaire des conditions initiales.

Exemple – Soit le problème différentiel :

$$y'' + y = x; \quad y(0) = y(1) = 0.$$

L'équation différentielle du second ordre est équivalente au système :

$$y' = z; \quad z' = -y + x$$

que nous avons résolu par la méthode RK2, avec les conditions initiales $y(0) = 0, y'(0) = s$. Les deux premières exécutions ont donné, pour $s_1 = -0,3$, $y(1) = -0,093915$ et pour $s_2 = 0,1$, $y(1) = 0,242674$. Une interpolation inverse linéaire nous donne alors $s_0 \simeq -0,1884$, qui conduit, comme le montre la figure 12.1, à $y(1) = 0$. La solution analytique est $y = x - \sin(x)/\sin(1)$; la dérivée à l'origine est $y'(0) = -0,188395$, très proche de la solution numérique.

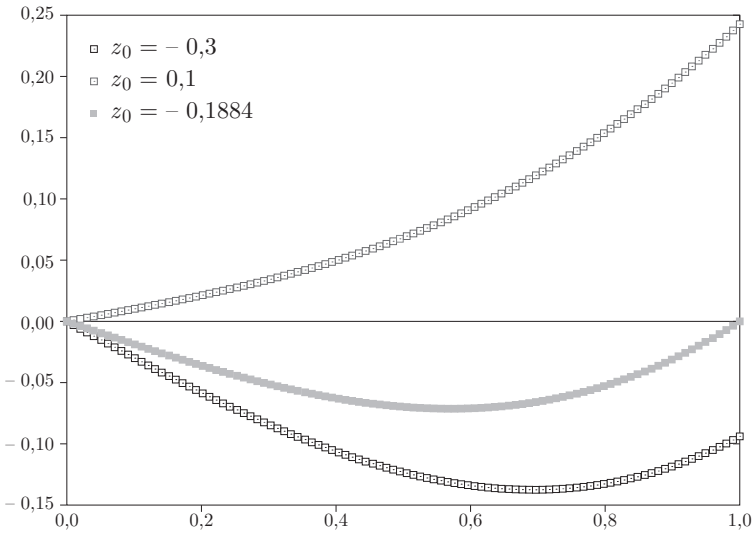


Figure 12.1 – La méthode du tir, associée au schéma RK2.

Nous ne dirons rien de la théorie associée à l'algorithme du tir ; elle se trouve dans les ouvrages cités.

12.1.2. Problèmes de valeurs propres

Un problème aux valeurs propres comme

$$y'' = f(x, y, y', \lambda); \quad y(a) = y(b) = 0 \tag{12.6}$$

peut se ramener à un problème à conditions initiales en introduisant une fonction inconnue auxiliaire $w \equiv \lambda$ et l'équation auxiliaire $w' = 0$ ou encore en considérant la valeur propre comme un paramètre ajustable, ce qui est d'une mise en oeuvre plus simple. Nous nous limitons à une équation différentielle linéaire, écrite sous la forme

$$y'' + p(x)y' + \lambda q(x)y = 0; \quad y(a) = y(b) = 0; \quad a \leq x \leq b. \tag{12.7}$$

Comme l'équation (12.7) et ses conditions aux limites sont homogènes, ky est solution si y l'est. Nous associons à ce problème de valeurs propres un problème auxiliaire à conditions initiales :

$$v'' + p(x)v' + \lambda q(x)v = 0; \quad v(a) = 0; v'(a) = p,$$

où p est une constante arbitraire, la pente initiale. La solution est de la forme $v = v(x, \lambda, p)$ et nous savons la trouver numériquement à l'aide de l'une des méthodes du chapitre 11. Pour déterminer la solution du problème aux valeurs propres, il nous faudra déterminer λ tel que :

$$v(b, \lambda, p) = 0.$$

Il s'agit encore d'une équation algébrique, mais ici l'inconnue est λ. Le paramètre p ne sert qu'à fixer la normalisation de y. Il peut être choisi empiriquement avant le début du calcul pour obtenir des valeurs commodes de y. La méthode de dichotomie se révèle très pratique pour trouver λ.

Exemple – La masse linéique d'une corde de longueur L varie selon la loi $\mu = \mu_0(1 + \alpha x)$. Lorsque cette corde effectue des oscillations sinusoïdales, sa forme est définie par l'équation :

$$y'' + (k_0 L)^2(1 + \alpha r)y = 0; \quad y(0) = y(1) = 0$$

où $r = x/L, k_0 = 2\pi\nu/c_0$ et $c_0 = \sqrt{T/\mu_0}$, T étant la tension. Les fréquences de vibration sont les valeurs de ν telles que le problème différentiel admette une solution. Nous les avons cherchées par la méthode du tir et l'algorithme de Runge–Kutta d'ordre 2. La figure 12.2 montre le résultat de trois exécutions du programme, pour des fréquences proches du second mode ($(k_0 L)^2 = 10\,;\, 11{,}532\,;\, 13$).

12.2. Méthodes des différences finies

12.2.1. Problème aux limites

Dans ce type d'algorithme, nous remplaçons, dans l'équation différentielle, les dérivées $y', y'', \ldots$ par des approximations formées à partir de différences latérales. Nous nous restreignons aux problèmes linéaires, comme le problème du second ordre à conditions aux limites

$$y'' + q(x)y = g(x); \quad y(a) = A; \quad y(b) = B. \tag{12.8}$$

Pour discrétiser ce problème, nous subdivisons $[a, b]$ en $n + 1$ sous-intervalles égaux, de longueur $h = (b - a)/(n + 1)$, séparés par des pivots d'abscisses $x_i = a + ih$, $x_0 = a, x_{n+1} = b$ et nous remplaçons $y''(x_i)$ par $[u_{i+1} - 2u_i + u_{i-1}]/h^2$, en posant $u_i \simeq y(x_i)$, $q_i = q(x_i)$, $g_i = g(x_i)$.

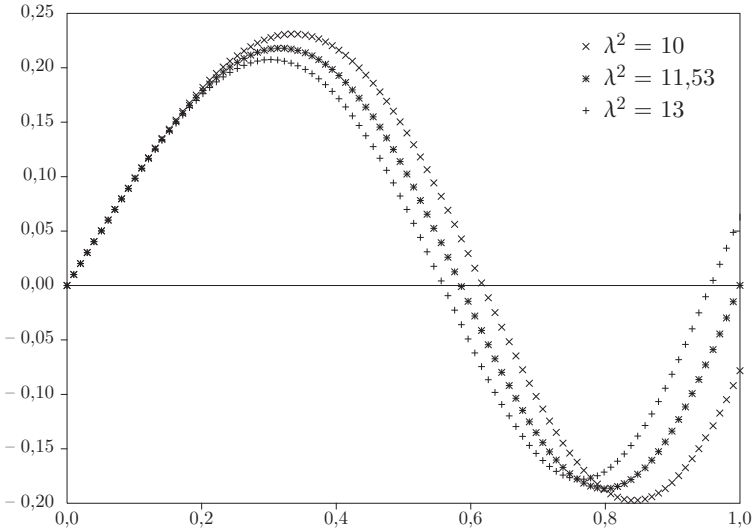


Figure 12.2 – Recherche du 2e mode d'une corde de densité variable. Le paramètre α vaut 5. La valeur propre est $k_0 L = 3,396$ ou $(k_0 L)^2 = 11,53$.

Ce faisant, nous commettons une erreur de troncation que le théorème de Taylor appliqué à y_{i+1} et y_{i-1} permet d'évaluer à $(h^2/12)y^{(4)}(x_i + \theta h), -1 < \theta < 1$. Il y a $n+2$ valeurs de u_i, dont deux nous sont déjà connues : $u_0 = A$, $u_{n+1} = B$; les autres sont solutions d'un système linéaire qui s'écrit, sous forme matricielle :

$$\boldsymbol{M}\boldsymbol{u} = \boldsymbol{c}$$

avec les définitions $\boldsymbol{u} = [u_1, u_2, \ldots, u_n]^T$, $\boldsymbol{c} = [h^2 g_1 - A, h^2 g_2, \ldots, h^2 g_{n-1}, h^2 g_n - B]^T$ et

$$\boldsymbol{M} = \begin{bmatrix} -2 + q_1 h^2 & 1 & 0 & \ldots & 0 \\ 1 & -2 + q_2 h^2 & 1 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & 1 & -2 + q_n h^2 \end{bmatrix}$$

On a le théorème suivant. Si $q_i \geq 0$ pour $i = 1, 2, ..., n$, alors $\boldsymbol{M}$ est définie positive : la solution du système linéaire précédent est alors facile, les méthodes de Gauss ou de Cholesky s'appliquent sans permutation de lignes (voir le chapitre 6). On montre aussi que l'erreur de troncation est partout d'ordre h^2.

Cet algorithme s'étend facilement aux problèmes contenant la dérivée première de y. Il est recommandé de représenter y' par une expression qui respecte la symétrie de la matrice $\boldsymbol{M}$.

Exemple – Reprenons le problème déjà résolu par la méthode du tir

$$y'' + y = x \; ; \quad y(0) = y(1) = 0.$$

Nous allons utiliser 20 points, donc 18 valeurs inconnues $\{u_k\}$. Les éléments de la matrice $\boldsymbol{M}$ sont $M_{i,i} = h^2 - 2, M_{i,i+1} = M_{i+1,i} = 1$ tandis que les coordonnées de $\boldsymbol{c}$ sont $c_i = h^2 x_i$. Voici le programme pour Scilab, qui produit le tracé de la figure 12.3. Nous avons prolongé, aux deux bouts, les vecteurs `xx` et `yy` par les valeurs connues.

Listing 12.1 – Résolution d'un problème aux limites par discrétisation

```
xmin = 0; xmax = 1;
npt = input("nombre total de points: ");
nint = npt-1; nx = npt-2;
M = zeros(nx,nx);
h = (xmax-xmin)/nint;
sousdiag = ones(nx-1,1);
M = diag(sousdiag,-1)+diag(sousdiag,1)
                    +(h*h-2)*diag(ones(nx,1),0);
x = linspace(xmin+h,xmax-h,nx)';
c = h*h*x
y = M\c;
yy = [0;y;0]; xx = [xmin;x;xmax];
xsegs([xmin,xmax],[0,0])
plot2d(xx,yy)
```

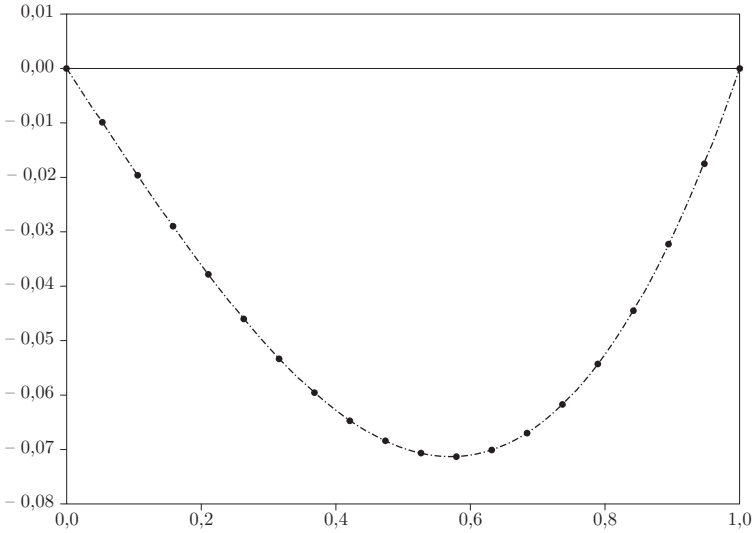


Figure 12.3 – Solution d'un problème avec conditions aux limites. La fonction inconnue est « discrétisée » sur 20 points. On a superposé la courbe représentant la solution analytique $y = x - \sin(x)/\sin(1)$.

12.2.2. Problème de valeurs propres

L'algorithme de discrétisation s'applique aussi, en principe, aux problèmes de valeurs propres linéaires ; cependant, ce n'est que pour des cas assez particuliers que l'on aboutit à une formulation algébrique facile à résoudre et stable. Considérons les problèmes ayant la forme conventionnelle (dite de Liouville)

$$-y'' + q(x)y = \lambda y; \quad y(A) = y(B) = 0. \tag{12.9}$$

Comme à la section précédente, nous subdivisons $[a, b]$ en $n+1$ sous-intervalles égaux, de longueur $h = (b-a)/(n+1)$, séparés par des pivots d'abscisses $x_i = a + ih$, $x_0 = a, x_{n+1} = b$ et nous remplaçons $y''(x_i)$ par $[u_{i+1} - 2u_i + u_{i-1}]/h^2$, en posant $u_i \simeq y(x_i)$. L'équation (12.9) est remplacée par son approximation discrète

$$\boldsymbol{M}\boldsymbol{u} = \lambda \boldsymbol{u}$$

avec les mêmes définitions de $\boldsymbol{M}$ et $\boldsymbol{u}$ qu'au paragraphe précédent. Comme $\boldsymbol{M}$ est d'ordre n, l'équation précédente admet n solutions comprenant chacune une valeur propre λ_k et un vecteur propre $\boldsymbol{u}^{(k)}$ (voir chapitre 10).

Dès que l'on s'écarte de cette structure simple (comme dans l'exemple de la corde de masse linéique variable), soit en compliquant les conditions aux limites, soit en généralisant l'équation différentielle, on aboutit à un problème « aux valeurs propres généralisé » de la forme

$$\boldsymbol{M}\boldsymbol{u} = \lambda \boldsymbol{K}\boldsymbol{u}$$

dont l'étude sort du cadre de cet ouvrage.

Les deux approches que nous venons de décrire (tir et discrétisation) sont en gros équivalentes, du point de vue de la commodité comme du point de vue de la qualité des résultats, pour tous les problèmes simples. L'équation de Schrödinger indépendante du temps, avec ses conditions aux limites, constitue un problème de valeurs propres. Ces conditions aux limites sont souvent « qualitatives ». On impose, par exemple, que la fonction d'onde $\psi(x)$, solution de l'équation de Schrödinger, tende vers zéro quand $|x| \to \infty$, « assez vite » pour que $|\psi(x)|^2$ soit intégrable sur l'intervalle $[-\infty, \infty]$. Comme l'ordinateur ne peut pas atteindre l'infini, il faut faire une approximation ; on imposera que $\psi(\pm L) = 0$.

Exemple – L'oscillateur harmonique quantique. Dans ce problème bien connu, une particule quantique de masse m se déplace selon l'axe des x sous l'effet du potentiel $V(x) = \frac{1}{2}kx^2$. Les fonctions d'ondes et les énergies des états stationnaires sont solutions de l'équation (variables sans dimension, comme au chapitre 3)

$$-\psi'' + x^2\psi = \lambda\psi. \tag{12.10}$$

Listing 12.2 – Résolution de l'équation de Schrödinger par discrétisation

```
N = 20;   L = 4;                                                    1
Nt = 2*N+1; Ni = 2*N; Nx = 2*N-1;                                   2
h = 2*L/Ni;                                                         3
x = linspace(-L+h,L-h,Nx);                                          4
y = zeros(x);                                                       5
xxhh = h*h*x.*x + 2;                                                6
sousdiag = ones(Nx-1,1);                                            7
M = diag(xxhh) - diag(sousdiag,1) - diag(sousdiag,-1);              8
[Psi,En] = spec(M);                                                 9
xx = [-L,x,L];                                                      10
psi1 = [0;Psi(:,1);0];                                              11
psi2 = [0;Psi(:,2);0];                                              12
psi3 = [0;Psi(:,3);0];                                              13
```

```
psi4 = [0;Psi(:,4);0];                                    14
plot2d(xx,[psi1,psi2,psi3,psi4]);                         15
ee = diag(En)/(h*h)                                       16
```

Tous les livres de mécanique quantique décrivent la résolution analytique de ce problème. La conclusion principale en est que les valeurs propres s'expriment comme $\lambda = 2n+1, n$ entier ≥ 0 (λ est le double de l'énergie). La fonction ψ est définie sur tout l'axe réel, mais nous ne cherchons une solution numérique que sur le segment $[-L, L]$. Celui-ci est divisé en `Ni` intervalles par `Nx = Ni - 1` noeuds où nous voulons connaître les valeurs des u_i. Celles-ci doivent vérifier

$$-u_{i-1} + (2 + h^2 x_i^2)u_i - u_{i+1} = h^2 \lambda u_i, \quad 1 \leq i \leq Nx. \tag{12.11}$$

Le programme ci-contre construit la matrice $\boldsymbol{M}$ en utilisant la fonction `diag`; celle-ci crée la diagonale principale ou des diagonales secondaires à partir de vecteurs de taille convenable. Nous obtenons ensuite les éléments propres grâce à la fonction `spec` de Scilab. La matrice `Psi` contient les vecteurs propres, `En` est la matrice diagonale, dont nous extrayons les valeurs propres, rangées dans le vecteur `ee`. Pour tracer les fonctions d'onde approchées de $-L$ à L, nous ajoutons un élément nul au début et à la fin de chaque vecteur propre et nous allongeons aussi le vecteur des abscisses (lignes 10 à 14). Les valeurs propres correspondantes sont 0,9974949 ; 2,987475 ; 4,967689 et 6,9401858 à comparer aux valeurs théoriques 1, 3, 5 et 7. Les solutions suivantes sont de moins en moins précises. On démontre que l'erreur sur les valeurs propres est $\mathcal{O}(h^2)$. On peut donc augmenter sensiblement la précision, sans beaucoup de calculs, par une extrapolation de Richardson.

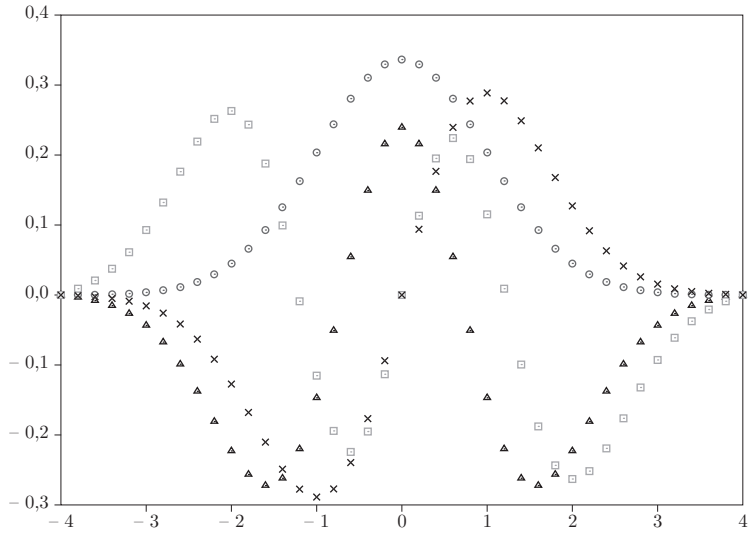


Figure 12.4 – Les quatre premières fonctions propres de l'oscillateur harmonique quantique.

12.3. Les boîtes noires

Un programme qui fonctionne et résout un problème numérique sans que l'utilisateur sache comment est souvent appelé une « boîte noire » (à ne pas confondre avec un programme dont on n'a pas lu la documentation). Du point de vue de l'utilisateur, Maple ne fait pas de différence entre problème à valeurs initiales et problème à conditions aux limites. Voici la solution de l'exemple du § 12.1 avec Maple.

```
> ode := diff(y(x),x,x)+y(x) = x;
```

$$ode := (\frac{d^2}{dx^2}\,\mathrm{y}(x)) + \mathrm{y}(x) = x$$

Solution analytique.

```
> dsolve(\{ode,y(0) = 0,y(1) = 0\},y(x));
```

$$\mathrm{y}(x) = -\frac{\sin(x)}{\sin(1)} + x$$

Solution numérique : on crée une " procedure " qui renvoie une liste de valeurs. Le deuxième élément de la liste est ensuite transformé en fonction de x.

```
> sln := dsolve(\{ode,y(0) = 0,y(1) = 0\},y(x),type = numeric);
```

$$sln := \mathbf{proc}(x\_bvp) \ldots \mathbf{end\ proc}$$

```
> sln(0.5);
```

$$[x = 0.5,\ \mathrm{y}(x) = -0.0697469634754012274,\ \frac{d}{dx}\,\mathrm{y}(x) = -0.0429148216718923322]$$

```
> Y := x -> rhs(op(2,sln(x)));
```

$$Y := x \rightarrow \mathrm{rhs}(\mathrm{op}(2,\ \mathrm{sln}(x)))$$

```
> plot(Y,0..1);
```

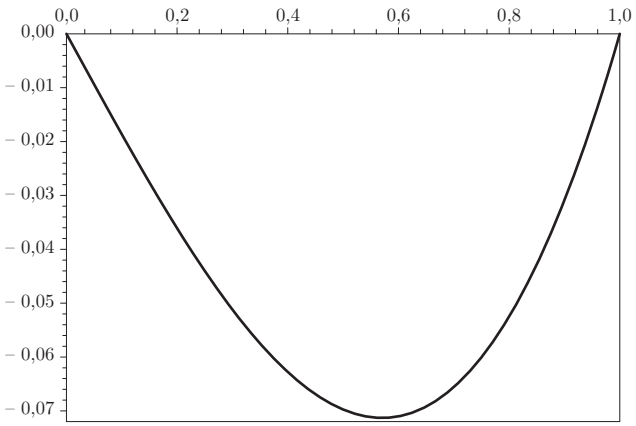


Figure 12.5 – Résolution d'un problème aux limites à l'aide de Maple.

Remarque : En réalité, il est possible de savoir ce que fait Maple, en modifiant la valeur du paramètre `verboseproc`, comme vous l'apprendra l'aide en ligne.

Scilab comporte aussi une boîte noire destinée à résoudre aussi bien les problèmes à conditions aux limites que les problèmes aux valeurs propres, `bvode` et sa version « simplifiée » `bvodeS`. La mise en oeuvre de ces logiciels constitue un excellent projet d'analyse numérique.

12.4. Pour en savoir plus

La résolution numérique des problèmes différentiels avec conditions aux limites a inspiré de nombreux auteurs, d'autant plus que certains de ces algorithmes peuvent être étendus aux équations aux dérivées partielles. Citons, parmi les sujets non traités dans le texte : la méthode de collocation (ou méthode pseudo-spectrale), la méthode spectrale et la méthode des éléments finis.

- W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery : *Numerical recipes, The art of scientific computing*, ch. 17 (Cambridge University Press, Cambridge, 2007).
- J.D. Pryce : *Numerical solution of Sturm-Liouville problems* (Clarendon Press, Oxford, 1993).
- J. P. Boyd : *Chebyshev and Fourier Spectral Methods* (Dover, Mineola, New-York, 2000) ; ce livre est aussi accessible à partir de la page personnelle de professeur Boyd : `http://www-personal.umich.edu/~jpboyd/`
- Site du programme SLEIGN2 (en Fortran) : `www.math.niu.edu/SL2/`
- Sur le site : `http://www.librecours.org/` :
  - L. Champaney : *Méthodes d'Approximation de Solution pour les Problèmes de Physique.*
  - É. Gonçalves : *Résolution numérique et discrétisation des EDP/EDO.*

12.5. Exercices

Exercice 1

Soit le problème différentiel avec conditions aux limites

$$y'' = y + 1; \qquad y(0) = 1; \qquad y(1) = 2. \tag{I}$$

a) Trouver la solution exacte.

b) Retrouver la solution exacte à l'aide de la méthode du tir, en résolvant analytiquement le problème à conditions initiales associé.

c) On considère maintenant les deux problèmes à conditions initiales

$$\begin{cases} u'' = u + 1 \\ u(0) = 1; \quad u'(0) = 0 \end{cases} \qquad \begin{cases} v'' = v + 1 \\ v(0) = 1; \quad v'(0) = 1 \end{cases}$$

Trouver u et v. Montrer que $w = u + \lambda(v - u)$ est une solution de l'équation différentielle (I) et qu'elle respecte la condition $w(0) = 1$. Comment choisir λ pour que w vérifie aussi $w(1) = 2$?

d) Résoudre (I) par la méthode du tir, associée à l'algorithme d'Euler, pour un pas $h = 1/3$.

e) Résoudre (I) par la méthode de discrétisation ; on choisira $h = 1/3$.

Exercice 2

On considère le problème aux valeurs propres :

$$y'' + \lambda y = 0; \quad y(0) = y(1) = 1.$$

a) Chercher la solution exacte.

b) Résoudre par la méthode du tir, en résolvant analytiquement le problème à conditions initiales associé.

c) Chercher une solution numérique, par la méthode du tir et l'algorithme d'Euler, avec un pas $h = 1/3$.

d) Résoudre par la méthode des différences finies (ou de discrétisation), pour le même pas.

Exercice 3

Appliquer deux fois la méthode des différences finies, avec les pas $h_1 = 0,5$ et $h_2 = 0,25$, pour résoudre le problème aux limites

$$y'' = y; \quad y(0) = 0; \quad y(1) = 1.$$

Améliorer les résultats par extrapolation de Richardson et comparer à la solution exacte.

Exercice 4

On applique la méthode de discrétisation à la résolution du problème

$$y'' + y = 0; \quad y(0) = y(1) = 0$$

avec un pas h, si bien que $x_i = ih$ et $y(x_i) = y_i$.

a) Quelle est l'équation vérifiée par les y_i ?

b) Montrer que la solution discrète converge vers la solution exacte lorsque $h \to 0$. Répondre aux mêmes questions pour le problème aux valeurs propres :

$$y'' + \lambda y = 0; \quad y(0) = y(1) = 0.$$

12.6. Projets

Projet 1. Flexion d'une poutre

Une poutre de longueur L est encastrée dans un mur ; elle est horizontale en l'absence de déformation. La poutre va fléchir sous l'effet de son poids et sous l'effet d'une force horizontale appliquée à l'extrémité libre. En première approximation, la forme de la poutre en charge est solution de l'équation différentielle

$$EIy'' = Ty - \mu g \frac{x^2}{2} \tag{12.12}$$

avec les notations suivantes
– E : module d'Young du matériau ;

– I : moment d'inertie de la section droite de la poutre ;

– μ : masse linéique de la poutre ;

– g : accélération de la pesanteur ;

– T : force appliquée

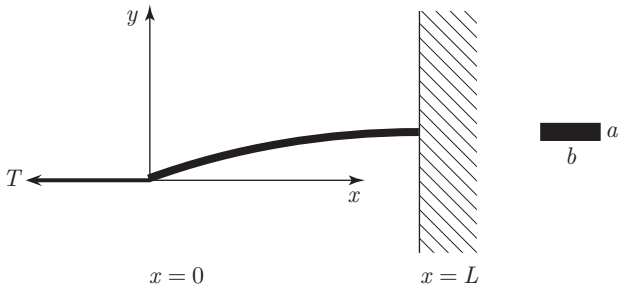


Figure 12.6 – Poutre en flexion.

On considère une poutre de section rectangulaire $a \times b$. Le moment d'inertie est de la forme $I = ba^3$, si b est le côté horizontal et a le côté vertical ; les rôles de a et b sont inversés si la poutre est tournée de 90°.

1. Étude analytique. Cette équation différentielle à coefficients constants peut se résoudre exactement. On ne considère ici que le cas particulier où $T = 0$. Trouver la fonction $y_0(x)$ satisfaisant à l'équation différentielle et aux conditions aux limites $y_0(0) = 0, y_0'(L) = 0$. Quelles sont alors les valeurs de $y_0'(0)$ et de la « flèche » $y_0(L)$?

2. Étude numérique. L'extrémité libre est choisie comme origine ; en ce point, la déformation verticale est $y(0) = 0$. La poutre reste horizontale à l'extrémité encastrée : $y'(L) = 0$.

3. Écrire le programme correspondant à la méthode qui vient d'être exposée. Données numériques pour une poutre en bois :
$$E \simeq 10^{10}\text{N/m}^2\,;\ \mu = 4{,}2\ \text{kg/m}\,;\ a = 5\ \text{cm}\,;\ b = 10\ \text{cm}\,;\ L = 3\text{m}.$$

a) Vérifier que les résultats, en l'absence de force, concordent avec ceux du 1°.

b) La force appliquée est une tension de 5000 N.

4. Comparer les déformations de la poutre lorsque le petit côté est horizontal et lorsque le grand côté est horizontal.

5. Examiner l'effet d'une force de compression de même module.

Remarque : Pour éviter d'avoir à calculer sur des nombres très petits ou très grands, il est recommandé de changer d'unités. Ainsi, on peut utiliser, pour les abscisses, la variable $X = x/L$ et pour les ordonnées, la variable $Y = y/y_0(L)$, où $y_0(L)$ est la flèche en l'absence de charge trouvée en 1°.

Projet 2. Tuyau sonore

La propagation du son dans un tuyau présentant une symétrie de révolution et une section variable $S(x)$ est régie par deux équations : la relation fondamentale de la dynamique appliquée à une tranche d'air et l'équation de continuité. En supposant que la pression et le débit sont constants dans un plan de section droite, celles-ci s'écrivent :

$$\frac{\partial p}{\partial x} = -i\frac{\omega\rho_0}{S}U; \quad \frac{\partial U}{\partial x} = -i\frac{\omega S}{\gamma P_0}p,$$

où p est la pression acoustique qui s'ajoute à la pression atmosphérique constante et uniforme P_0, ρ_0 est la masse volumique moyenne du fluide, U est le débit (m^3/s) à l'abscisse x et $\gamma = C_p/C_v$. On a supposé que p et U étaient des fonctions sinusoïdales du temps, de pulsation ω, ce qui permet d'éliminer la variable temps. Ces équations sont tout à fait analogues aux relations entre courant et tension le long d'une ligne électrique (la pression remplaçant la tension et le débit prenant la place du courant). La constante $i\omega\rho_0/S$ est une impédance acoustique série par unité de longueur, alors que $i\omega S/\gamma P_0$ est une admittance acoustique parallèle par unité de longueur.

1. Écrire un programme pour résoudre le système différentiel par la « méthode du tir », associée à la méthode Runge–Kutta d'ordre 4. On rappelle que cet algorithme revient à choisir ω, des valeurs de p et U à une extrémité du tuyau ($x = 0$ par exemple) compatibles avec les conditions aux limites, à calculer p et U le long du tuyau, jusqu'à l'autre extrémité ($x = L$) à l'aide de l'algorithme de Runge–Kutta et enfin à s'assurer que les conditions aux limites en $x = L$ sont vérifiées. Si ce n'est pas le cas, on choisit une nouvelle valeur de ω et on recommence. Une recherche par dichotomie des valeurs de ω est assez efficace. Chaque fréquence propre doit être déterminée à une précision meilleure que le demi-ton, soit 0.5 %.

   Les conditions aux limites à une extrémité fermée sont $U = 0$, $p \neq 0$ (la valeur de p fixe l'amplitude arbitraire des vibrations acoustiques de l'onde stationnaire). A une extrémité ouverte, on a $p = 0, U \neq 0$ (c'est U qui fixe alors l'amplitude).

   Le programme peut ne comporter que des quantités réelles, (en séparant partie réelle et partie imaginaire de U et p) mais on peut aussi, sous Scilab, manipuler directement des variables complexes , pour pouvoir incorporer divers perfectionnements (dissipation en particulier).

2. Retrouver les modes de tuyaux cylindriques de section constante ouvert-ouvert ou ouvert-fermé ($L = 0{,}8$ m).

3. En réalité, l'impédance acoustique d'une extrémité ouverte ($Z = p/U$) n'est pas nulle, parce que l'énergie acoustique est rayonnée sous forme d'ondes sphériques. Si le rayon du tuyau (a) est très petit devant la longueur d'onde, on a :

$$Z_{ouv} = \frac{p}{U} = i\frac{0,6133}{\pi}\frac{\rho_0\omega}{a}.$$

On peut en déduire la règle approchée suivante : les corrections d'extrémité font que la longueur du tuyau est augmentée de $0,6a$. Incorporer cette correction dans le programme ($a = 0{,}012$ m).

4. Tuyau conique. Beaucoup d'instruments à vent réels ont un profil presque rigoureusement conique. Cependant, comme on souffle dans l'extrémité étroite, le cône n'est pas tout à fait complet (S n'est pas nulle). Trouver les modes d'un tuyau conique, ouvert-ouvert ou fermé-ouvert. On interrompra le cône à une distance du sommet égale au pas d'intégration. On s'attend à ce que les fréquences propres soient peu sensibles à la nature ouverte ou fermée de l'extrémité étroite (puisqu'elle est presque fermée de toute manière). Cette hypothèse est-elle vérifiée ?

5. L'ensemble gorge-bouche peut être modélisé comme un tuyau de profil variable selon les sons à émettre. Pour la lettre "a", des clichés aux rayons X ont permis d'établir les dimensions données dans la table (section droite en fonction de la distance à partir de fond de la gorge). Trouver les trois premières fréquences ("partiels") composant ce son. Les valeurs numériques de la fonction $S(x)$ seront calculées par interpolation linéaire dans la table

$d(cm)$	$A(cm^2)$	d	A	d	A	d	A	d	A
0.0	2.6	3.5	1.0	7.0	2.0	10.5	6.5	14.0	9.0
0.5	1.6	4.0	0.65	7.5	2.6	11.0	8.0	14.5	8.0
1.0	1.3	4.5	0.65	8.0	2.6	11.5	8.0	15.0	6.5
1.5	1.0	5.0	0.65	8.5	1.6	12.0	8.0	15.5	5.0
2.0	4.0	5.5	1.0	9.0	3.2	12.5	8.0	16.0	5.0
2.5	2.6	6.0	1.3	9.5	4.0	13.0	8.0	16.5	5.0
3.0	1.6	6.5	1.6	10.0	5.0	13.5	8.0	17.0	5.0

Projet 3. Modèle de Kronig–Penney

1. Introduction

Un électron dans un cristal se déplace dans le potentiel périodique créé par les noyaux, disposés aux noeuds d'un réseau. Bloch (1928) a montré que, dans un milieu infini, les fonctions d'onde étaient des ondes planes progressives, modulées par une fonction de même période que le réseau. Pour certains intervalles d'énergie, le vecteur de propagation est complexe, les solutions n'ont pas de sens physique : on parle de bandes interdites (ou « gaps »).

Le modèle des liaisons fortes aboutit à la même conclusion. Dans ce modèle, on s'aperçoit que les niveaux d'énergie discrets d'un électron dans un puits éclatent en plusieurs niveaux en présence de puits voisins ; il apparaît des bandes quasi-continues d'états permis séparées par des bandes interdites.

Kronig et Penney (1930) ont étudié une rangée de puits rectangulaires en nombre fini. En utilisant des conditions aux limites périodiques, ils retrouvent les résultats de Bloch ; les énergies permises sont solutions d'une équation transcendante.

Depuis 1950 et avec l'apparition de semi-conducteurs dopés, on s'intéresse aux réseaux irréguliers. Ils présentent des niveaux d'énergie permis au sein des bandes interdites, et Anderson (1958) a prouvé que les fonctions d'ondes correspondantes étaient localisées au voisinage des défauts. Ce projet consiste à étudier numériquement le modèle de Kronig–Penney, avec un nombre de puits compris entre 1 et environ 20.

2. Modèle Physique

La figure ci-dessous représente le potentiel de Kronig et Penney. $a_0 = 0,529\,\text{Å}$ est le rayon de Bohr. Le potentiel devient très grand aux bords de la région considérée, si bien que la fonction d'onde s'annule en ces points. L'échelle verticale est plausible, sans plus ; on peut perfectionner le modèle en donnant une valeur explicite au potentiel à l'extérieur (cf. le travail d'extraction) et en modifiant les conditions aux limites en conséquence.

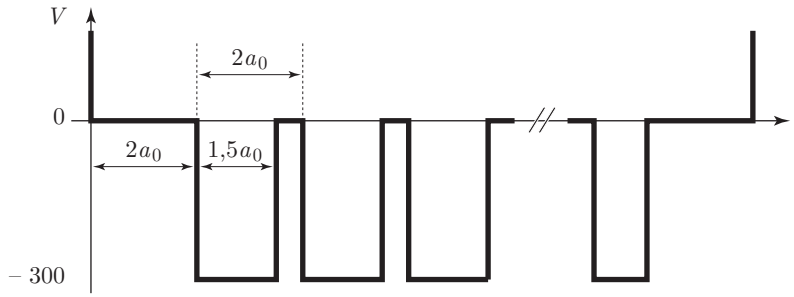


Figure 12.7 – Potentiel de Kronig–Penney.

L'équation de Schrödinger s'écrit, en unités atomiques

$$\frac{1}{2}\psi''(x) + [E - V(x)]\psi(x) = 0.$$

3. Programmation

Écrire un programme pour déterminer les énergies permises et les fonctions d'ondes par la méthode du tir, pour un nombre de puits choisi par l'utilisateur. Il arrive que les sauts de potentiel posent problème. On peut dans ce cas poser $V = -150$ eV en chaque point de discontinuité. Il est recommandé de faire fonctionner l'algorithme « à la main » au début ; lorsque tout fonctionne très bien, on peut chercher les énergies automatiquement par dichotomie. Il est souvent nécessaire de renormaliser la fonction d'onde en cours de route, pour éviter des valeurs trop grandes. Prévoir de représenter graphiquement la fonction potentiel et les énergies permises, d'une

part, et la fonction potentiel et une fonction d'onde déterminée d'autre part. Vérifier le bon fonctionnement pour un puits unique.

4. Applications

   a) Pour 10 puits, on doit trouver 25 niveaux répartis en 3 « bandes » permises (le plus lié est à -266.7709 eV, le moins lié à -3.98506 eV). Vérifier qu'un regroupement analogue persiste pour d'autres nombres de puits.

   b) Examiner les fonctions d'onde. Elles devraient avoir une forme d'onde de Bloch « stationnaire » : un produit d'une fonction ayant la périodicité du réseau (et un nombre de noeuds correct) par une onde stationnaire caractérisant un électron libre (dans un grand puits unique).

   c) Calculer la densité de probabilité due à toutes les fonctions d'une bande et la représenter.

   d) Mimer la présence d'une impureté, par exemple en réduisant la largeur d'un puits de 25%. Quelle est la nouvelle structure de bande ? Voit-on apparaître un niveau d'impureté ? Que deviennent les densités de probabilité ?

   e) Simuler un matériau amorphe en faisant fluctuer aléatoirement la distance entre puits. Que devient la structure de bandes ?

   f) Effet d'un champ électrique. En superposant à la fonction $V(x)$ en créneau une fonction $V_1(x)$ variant linéairement entre -10 et +10 eV, on simule l'effet d'un champ électrique. Examiner à nouveau les fonctions d'onde et la densité électronique globale pour chaque bande.

Projet 4. Vibrations d'une membrane circulaire

Les vibrations transversales d'une membrane circulaire, de tension uniforme T et de masse surfacique uniforme μ, sont décrites par l'équation aux dérivées partielles :

$$\nabla^2 u = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \varphi^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2}$$

où $u(r, \varphi, t)$ est la déformation de la membrane et $c = \sqrt{T/\mu}$ est la vitesse de propagation des ondes. On se propose de chercher les mouvements sinusoïdaux d'une membrane circulaire de rayon a.

1. On fait l'hypothèse que $u = R(r)\Phi(\varphi)e^{-2i\pi\nu t}$ et on applique la méthode de séparation des variables. Montrer que la fonction R(r) doit être solution de l'équation :

$$R''(r) + \frac{1}{r}R'(r) + \left(k^2 - \frac{m^2}{r^2}\right)R(r) = 0 \tag{12.13}$$

   où on a posé $k = 2\pi\nu/c$ (le nombre d'onde). Déterminer la fonction $\Phi(\varphi)$ et les valeurs possibles de la constante de séparation m. Faire le changement de variable $\rho = r/a$; quelle forme prend l'équation (12.13) ? On note $S(\rho)$ la nouvelle fonction inconnue.

2. L'équation (12.13) est associée à deux conditions aux limites : $R(a) = S(1) = 0$ et $R(0) = S(0)$ finie. On demande de déterminer, par la méthode du tir, les valeurs

permises de la fréquence. On admet de plus que les solutions acceptables sont de deux types : celles pour lesquelles $S(0) = 0$ (et $S'(0) \neq 0$) et celles qui correspondent à $S(0) = 1$ (et $S'(0) = 0$). Pour une valeur de m donnée, on notera ν_p et k_p les valeurs permises de la fréquence et du nombre d'onde. On se limitera à p et m de l'ordre de 3. Les fréquences obtenues sont-elles les harmoniques d'une fréquence fondamentale ?

3. Représenter graphiquement la fonction u_{mp} pour quelques valeurs de m et p.

This page intentionally left blank

Chapitre 13

Équations aux dérivées partielles

Bien des phénomènes physiques ou physico-chimiques peuvent être décrits par des équations aux dérivées partielles. La plupart portent le nom d'un savant des siècles précédents. C'est le cas en électrostatique (Laplace, Poisson), en électrodynamique (Maxwell), pour la théorie de la chaleur (Fourier), pour la diffusion (Fick), en mécanique des fluides (Euler, Navier–Stokes) comme pour la mécanique quantique (Schrödinger). La résolution des équations aux dérivées partielles est donc un sujet important. C'est aussi un domaine très travaillé et encore en plein développement. Autant dire que, dans ce chapitre, nous ne ferons qu'explorer une toute petite partie de la question. Nous nous limitons aux équations linéaires, à coefficients constants, à deux (ou trois) variables indépendantes, en ne considérant que des problèmes simples, solubles par des méthodes élémentaires de discrétisation. Comme dans le cas des problèmes différentiels à conditions aux limites (chapitre 12), ce formalisme conduit à un système d'équations linéaires. Les développements qui vont suivre peuvent donc être considérés comme des applications des algorithmes décrits dans le chapitre 6.

13.1. Approximation des dérivées par des différences finies

Comme nous l'avons détaillé dans le chapitre 8, nous pouvons approcher les dérivées d'une fonction u par les expressions suivantes :

$$u'(x) = \frac{1}{2h}[u(x+h) - u(x-h)] + O(h^2),$$
$$u''(x) = \frac{1}{h^2}[u(x+h) - 2u(x) + u(x-h)] + O(h^2).$$

Ces relations symétriques utilisent la parité des termes de développements de Taylor pour minimiser l'erreur de troncation. Nous connaissons aussi des formules dissymétriques inspirées des différences latérales :

$$u'(x) = \frac{1}{h}[u(x+h) - u(x)] + O(h) \quad ; \quad u'(x) = \frac{1}{h}[u(x) - u(x-h)] + O(h),$$

qui peuvent être utiles aux bords d'un domaine. On a plus rarement recours à des formules d'ordre plus élevé ; il est souvent préférable de diminuer h. Dans le cas de fonctions de plusieurs variables, nous appliquons ces relations indépendamment à chaque variable. Il faut alors définir un incrément pour chacune.

13.2. Équations de Laplace et Poisson

L'équation de Poisson (13.1), qui inclut l'équation de Laplace comme cas particulier lorsque $f = 0$, apparaît dans bien des problèmes d'électrostatique. Si nous choisissons les coordonnées cartésiennes, x et y comme variables indépendantes et si nous notons u la fonction inconnue, elle s'écrit :

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y); \quad (x, y) \in \mathcal{D} \tag{13.1}$$

où $f(x, y)$ est une fonction connue, suffisamment régulière dans le domaine $\mathcal{D}$ du plan x, y, domaine que nous supposons borné et simplement connexe. Lorsque l'on cherche à résoudre un problème concret, cette équation n'intervient pas seule, mais elle est associée à des « conditions aux limites ». Classiquement, on distingue deux catégories de conditions. On parle de « problème de Dirichlet » si u doit vérifier, en plus de l'équation de Poisson, la condition :

$$u(x, y) = g(x, y); \quad (x, y) \in \mathcal{C} \tag{13.2}$$

où g est une fonction régulière et $\mathcal{C}$ est la frontière de $\mathcal{D}$. On désigne par « problème de von Neumann » le cas où, u étant solution de l'équation (13.1) la dérivée de u dans la direction normale à la frontière de $\mathcal{D}$ est imposée :

$$\partial u / \partial n|_{x,y} = g(x, y); \quad (x, y) \in \mathcal{C}. \tag{13.3}$$

Une équation comme (13.1) est dite « elliptique » (ou de type elliptique) par analogie avec l'équation d'une ellipse en coordonnées cartésiennes : $x^2/a^2 + y^2/b^2 = 1$ où les termes en x^2 et en y^2 ont le même signe. Remarquez qu'il faut trouver u **à l'intérieur d'un domaine fermé**. D'autre part, bien que u représente une grandeur physique telle que le potentiel électrique, toute référence à des unités ou à des paramètres physiques a disparu : le problème a été « dédimensionnalisé ».

Abordons la discrétisation de l'équation de Poisson. Pour simplifier l'exposé, nous cherchons la solution à l'intérieur d'une région rectangulaire du plan (x, y), les côtés du rectangle étant parallèles aux axes de coordonnées. Nous superposons au plan (x, y) un quadrillage de pas h en x et k en y ; dans la suite, nous nous intéresserons aux valeurs de u aux points de coordonnées $x_i = a + ih, y_j = c + jk$, avec $x_0 = a, x_m = b, y_0 = c, y_n = d$. Il y a m intervalles dans la direction x, donc $h = (b - a)/m$; de même, nous comptons n intervalles en y et $k = (d - c)/n$. Nous adoptons la notation $u(x_i, y_j) \simeq U_{ij}$ et ce sont ces quantités (des approximations de la fonction inconnue, au nombre de $(m-1)\times(n-1)$) que nous devons trouver. Nous remplaçons les dérivées partielles par leurs approximations en termes de différences finies ; ainsi, au point i, j :

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{ij} \simeq \frac{1}{h^2}[U_{i+1,j} - 2U_{i,j} + U_{i-1,j}] \tag{13.4}$$

avec une équation analogue en y. L'équation de Poisson se réduit donc à un système de $(m-1)(n-1)$ équations linéaires pour autant d'inconnues :

$$(1/h^2)(U_{i+1,j} - 2U_{i,j} + U_{i-1,j}) + (1/k^2)(U_{i,j+1} - 2U_{i,j} + U_{i,j-1}) = F_{i,j} \qquad (13.5)$$

où $F_{i,j} \equiv f(x_i, y_j)$. Nous supposons avoir affaire à un problème de Dirichlet, u est connue sur les frontières du domaine :

$$U_{0,j} = g(x_0, y_j); \quad U_{m,j} = g(x_m, y_j);$$
$$U_{i,0} = g(x_i, y_0); \quad U_{i,n} = g(x_i, y_n).$$

Toutes les méthodes de résolution des systèmes linéaires sont en principe applicables à ce cas. Cependant, il est raisonnable de tenir compte de la structure de ces équations, laquelle dépend à son tour de la façon dont nous allons renuméroter les inconnues. Pour l'instant, $U_{i,j}$ dépend de deux indices, ce qui est malcommode. Le plus simple consiste à numéroter les inconnues ligne par ligne, à l'aide d'un indice unique $k = i + (m-1)j$. Avec une numérotation convenable, la matrice des coefficients sera symétrique. Ce sera aussi une matrice tridiagonale ou presque, comportant beaucoup d'éléments nuls. De plus, cette matrice sera diagonalement dominante. Enfin, on peut prévoir que m et n seront assez grands. Toutes ces raisons font que les méthodes itératives sont très souvent utilisées pour résoudre l'équation de Poisson-Laplace.

Lorsque x et y jouent des rôles comparables, il est raisonnable de choisir $h = k$; le système d'équations se simplifie alors en :

$$U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4U_{ij} = h^2 g_{i,j} \qquad (13.6)$$

Vous remarquez que, dans le cas de l'équation de Laplace ($g = 0$), le potentiel au point (i, j) est la moyenne des potentiels existants aux points les plus proches au nord, au sud, à l'ouest et à l'est. Jusque dans les années 60, on résolvait un tel système linéaire sans ordinateur, à la main, par une méthode d'approximations successivess, dite méthode de relaxation, et qui n'était en fait qu'une variante de l'algorithme de Gauss-Seidel. Décrivons rapidement la mise en oeuvre de la relaxation. Nous nous donnons une approximation initiale $U_{i,j}^{(0)}$, chaque valeur étant reportée, au point convenable, sur un dessin à grande échelle. Nous explorons ensuite ligne par ligne (par exemple) le tableau des $U_{i,j}$ $(1 \leq i \leq m-1; 1 \leq j \leq n-1)$ et nous remplaçons chaque valeur par la moyenne des 4 valeurs qui l'entourent : $U_{i,j} \leftarrow (1/4)(U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1})$, en gommant l'ancienne valeur et en écrivant la nouvelle. Après un certain nombre de balayages, les valeurs des $U_{i,j}$ se stabilisent ; nous admettons que la convergence est atteinte quand la variation relative de **chaque** $U_{i,j}$ d'une itération à l'autre est inférieure à un seuil convenable. De nos jours, les tableurs (comme Excel ou Calc d'OpenOffice) peuvent remplacer le papier, le crayon et la gomme. Il suffit d'installer dans chaque cellule intérieure la formule précédente comme par exemple

$$D5 = 0.25 * (D4 + D6 + C5 + E5)$$

et les conditions aux limites invariables dans les cellules des bords. Il faut enfin permettre au logiciel de traiter les « références circulaires » et indiquer le nombre maximal d'itérations et le seuil de convergence. La figure 13.1 montre le résultat d'un tel calcul pour un condensateur plan, constitué de deux rubans parallèles infiniment longs vus en coupe.

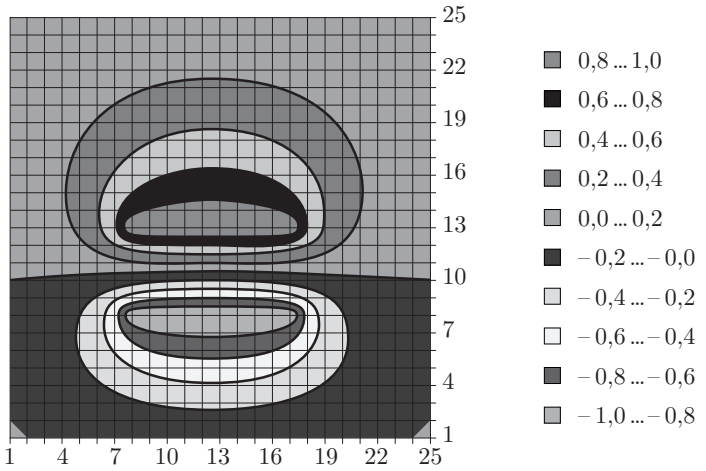


Figure 13.1 – Résolution de l'équation de Laplace à l'aide d'Excel.

Dans la réalité, il est rare que les frontières du domaine de définition de U coïncident avec des lignes de coordonnées. Il y a alors au moins deux façons de procéder. La plus simple consiste à adopter un pas en x et y assez petit pour que l'on puisse approcher la frontière par un escalier plus ou moins régulier. La seconde consiste à modifier les formules exprimant les dérivées seconde pour tenir compte de pivots non équidistants. Dans tous les cas, on espère que le programme « suivra » automatiquement le tracé de la frontière, ce qui n'est pas difficile, mais est fastidieux à réaliser.

13.3. Équation de la chaleur

L'équation $y = ax^2$ représente une parabole. Par analogie, on dit que l'équation de la chaleur (ou de la diffusion), qui s'écrit :

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}; \quad 0 \leq x \leq L; \quad t > 0 \tag{13.7}$$

est une équation de type parabolique. Le choix de la notation sous-entend que x est une variable d'espace, t est le temps, u est une température ou une concentration (moyennant un choix convenable d'unités pour que l'équation apparaisse sans dimension). Il existe des conditions aux limites ; pour fixer les idées, nous les supposons de Dirichlet :

$$u(0,t) = f_1(t); \quad u(L,t) = f_2(t)$$

et une « condition initiale » : $u(x,0) = g(x)$. Ici, nous cherchons u dans un domaine ouvert du plan x,t : la bande $t > 0$, $0 \leq x \leq L$. On peut également rencontrer des conditions aux limites portant sur les dérivées de u.

Nous partageons le segment $[0,L]$ en m intervalles de longueur h par les points $x_i, 1 \leq i \leq m-1$, $x_0 = 0, x_m = L$ et nous remplaçons la dérivée seconde par

son approximation sous forme de différence finie :

$$\frac{du_i}{dt} = \frac{1}{h^2}[u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)],$$

où la notation $u_i(t)$ représente la valeur de la fonction u au point d'abscisse discrète ih. L'équation précédente représente une composante d'un système de $m-1$ équations différentielles linéaires couplées, qui déterminent les $m-1$ fonctions $u_i(t)$. Toutes les méthodes exposées au chapitre 11 sont applicables en principe pour résoudre ce système. Cependant, il peut y avoir beaucoup d'inconnues à calculer sur un temps très long et les équations ont l'air très simples (coefficients constants). Par conséquent, nous essayons d'abord les méthodes les plus simples et les plus rapides. Nous commençons par remplacer la variable continue t par la variable discrète $t_j = j\tau$ et nous notons $u(x_i, t_j) \equiv U_i^j$.

Nous pouvons faire appel à la méthode d'Euler :

$$\frac{1}{\tau}(U_i^{j+1} - U_i^j) = \frac{1}{h^2}(U_{i+1}^j - 2U_i^j + U_{i-1}^j)$$

soit encore ($r \equiv \tau/h^2$)

$$U_i^{j+1} = (1-2r)U_i^j + r(U_{i+1}^j + U_{i-1}^j), \tag{13.8}$$

ce que nous pouvons aussi écrire $\boldsymbol{u}^{(j+1)} = \boldsymbol{A}\boldsymbol{u}^{(j)}$ en posant $u_i^{(j)} = U_i^j, 0 \leq i \leq m-1$. Ces relations ont le mérite de la simplicité : le vecteur $\boldsymbol{u}^{(j+1)}$ se calcule à partir du vecteur $\boldsymbol{u}^{(j)}$ par une simple multiplication par la matrice tridiagonale $\boldsymbol{A}$ et ce calcul est itéré, à partir de $\boldsymbol{u}^{(0)} = \{U(x_i, 0)\}$, autant de fois que nécessaire. Cependant, une étude plus approfondie montre que l'algorithme est instable dès que $r = \tau/h^2 > 1/2$, comme vous pouvez le constater en comparant les figures 13.2 et 13.3.

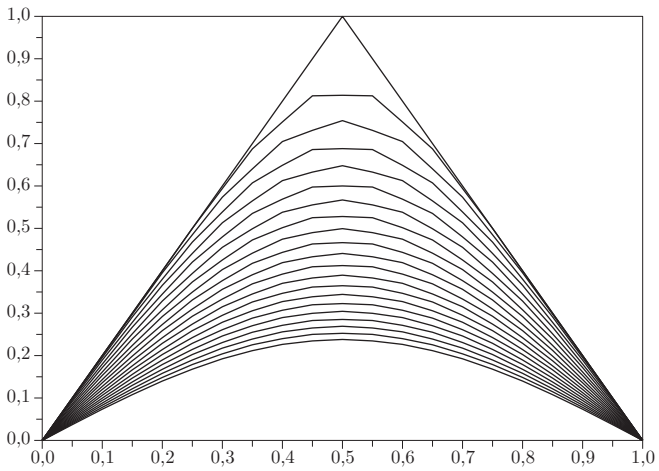


Figure 13.2 – Résolution de l'équation de la chaleur par une méthode explicite. Les valeurs des paramètres sont : $L = 1$, $m = 21$, $r = 0{,}496$. On a représenté une courbe sur 5, de $t = 0$ à $t = 0{,}13$.

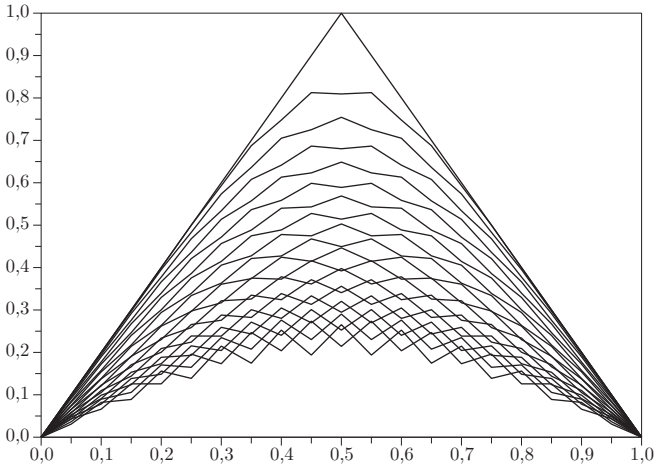


Figure 13.3 – Résolution de l'équation de la chaleur par la méthode de la figure 13.1, mais pour $r = 0{,}508$.

Exemple – Une barre conductrice, aux propriétés uniformes, est portée, à l'instant 0, à une température dépendant de l'abscisse :

$$0 \leq x \leq L/2 : u(x,0) = x; L/2 \leq x \leq L : u(x,0) = L - x$$

et nous cherchons la répartition des températures aux époques ultérieures. Les figures montrent les résultats de deux calculs, l'un pour $r = 0,495$, l'autre pour $r = 0,515$.

Nous pouvons essayer une autre approximation de $\partial u/\partial t$ qui conduit à

$$\frac{1}{\tau}(U_i^{j+1} - U_i^j) = \frac{1}{h^2}(U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1}) \tag{13.9}$$

qui n'est autre que la transposition de l'algorithme d'Euler implicite (voir §11.5.2 et §11.6). Cet algorithme est stable, mais il est plus compliqué à mettre en oeuvre que le précédent. En effet, une relation relie plusieurs U_i^{j+1} inconnus à l'un des U_i^j : l'équation aux dérivées partielles est devenue un système d'équations linéaires que nous pouvons résumer en $\boldsymbol{A}\boldsymbol{u}^{(j+1)} = \boldsymbol{u}^{(j)}$. Il faut donc résoudre ce système pour chaque pas en temps. C'est une bonne occasion d'appliquer la factorisation $\boldsymbol{LU}$ à $\boldsymbol{A}$; il nous suffit alors de résoudre deux systèmes triangulaires à chaque pas en t.

L'algorithme précédent est rarement utilisé, car on connaît une variante plus performante, la méthode de Crank et Nicolson. Ces auteurs ont proposé de travailler avec la moyenne des deux formules (13.8) (Euler explicite) et (13.9) (Euler implicite) :

$$\frac{2}{\tau}(U_i^{j+1} - U_i^j) = \frac{1}{h^2}(U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1}) + \frac{1}{h^2}(U_{i+1}^j - 2U_i^j + U_{i-1}^j).$$

Posons encore $r = \tau/h^2$, il vient :

$$-rU_{i-1}^{j+1} + (2+2r)U_i^{j+1} - rU_{i+1}^{j+1} = rU_{i-1}^j + (2-2r)U_i^j + rU_{i+1}^j \tag{13.10}$$

que nous symbolisons en $\boldsymbol{A}\boldsymbol{u}^{(j+1)} = \boldsymbol{B}\boldsymbol{u}^{(j)}$. Il est encore commode de factoriser $\boldsymbol{A}$, en tenant compte de ce que le système est tridiagonal.

Exemple – Nous traitons à nouveau le problème de la barre homogène conductrice. Le programme utilisé met en oeuvre mécaniquement les formules (6.19) et (6.20) qui permettent de résoudre un système linéaire tridiagonal. La seule difficulté réside dans le choix de la numérotation des composantes des différents vecteurs. Si les abscisses sont numérotées de 1 à $N+2$, les températures inconnues seront notées $T_2, \ldots, T_{N+1}$, la matrice $\boldsymbol{A}$ sera d'ordre N et les sous-diagonales comporteront chacune $N-1$ éléments. Il est alors commode, au vu des formules (6.19), (6.20) de définir des vecteurs $\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{\ell}, \boldsymbol{u}$ de longueur N avec $b(1) = \ell(1) = c(N) = u(N) \equiv 0$.

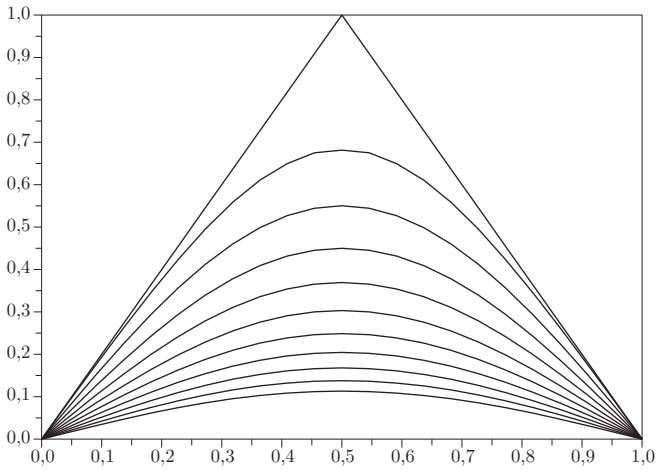


Figure 13.4 – Résolution de l'équation de la chaleur par la méthode implicite de Crank et Nicolson. Le paramètre r est pris égal à 1,94.

13.4. Équation des ondes

La dernière section conique est l'hyperbole, d'équation : $x^2/a^2 - y^2/b^2 = cte$ (signes opposés pour x et y), dont l'analogue est :

$$c^2 \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial t^2} = 0 \tag{13.11}$$

Nous avons choisi le cas particulier de l'équation des ondes, où x est une variable d'espace (généralement confinée au segment $0, L$) et t est le temps. Comme précédemment, u (ou l'une de ses dérivées) nous est donnée en $x = 0$ et en $x = L$, pour tout t :

$$u(0,t) = a(t) \quad ; \quad u(L,t) = b(t)$$

et nous cherchons $u(x,t)$ pour $t > 0$, sachant que $u(x,0) = f(x)$ et que $|\partial u/\partial t|_{t=0} = g(x)$. Nous avons encore affaire à un domaine de définition ouvert. Ce type d'équation est généralement plus difficile à résoudre numériquement que les cas traités précédemment. En effet, cette équation décrit la propagation d'une onde ; vous pouvez imaginer que le schéma numérique correspondant décrira, non seulement la propagation de l'onde physique mais aussi celle d'une erreur (de troncation ou d'arrondi) commise en un point du domaine.

Décrivons l'algorithme le plus simple applicable à l'équation des ondes. Nous discrétisons le domaine avec un pas h en x (indice i, $1 \leq i \leq m-1$) et un pas τ en t (indice j) et nous remplaçons chaque dérivée seconde par une différence finie :

$$U_i^{j+1} - 2U_i^j + U_i^{j-1} = r[U_{i+1}^j - 2U_i^j + U_{i-1}^j]$$

avec $r = (c\tau/h)^2$, ou encore $\boldsymbol{u}^{(j+1)} = \boldsymbol{A}\boldsymbol{u}^{(j)} - \boldsymbol{u}^{(j-1)}$, méthode explicite dont le démarrage demande la connaissance de $\boldsymbol{u}^{(0)}$ ($\equiv f$) et de $\boldsymbol{u}^{(1)}$ (obtenu à partir de f et g) :

$$U_i^1 \simeq f(ih) + \tau g(ih), 1 \leq i \leq m-1.$$

Ici aussi, nous rencontrons une condition de stabilité, connue sous le nom de condition de Courant, Friedrichs et Levy (« condition CFL »). Les incréments h et τ doivent vérifier

$$\frac{h}{c\tau} \geq 1.$$

La quantité h/τ a les dimensions d'une vitesse. La condition CFL peut donc s'exprimer ainsi : la pseudo-vitesse numérique doit être plus grande que la vitesse des ondes physiques, ou encore : τ doit être inférieur au temps h/c mis par l'onde pour parcourir la distance h. En pratique, la quantité c est imposée par la physique du problème, le pas h est choisi en fonction de la résolution souhaitée et on déduit τ pour que l'algorithme soit stable.

13.5. Pour en savoir plus

La résolution des équations aux dérivées partielles est un domaine immense et en plein développement, dont nous n'avons présenté que les aspects les plus élémentaires. La méthode de loin la plus utilisée est celle des éléments finis ; le logiciel gratuit et d'installation facile sur toute plate-forme `Freefem++` permet une initiation très commode à la pratique de cette méthode, mais devra être complété par un texte théorique. Il faut citer aussi la méthode multigrille et les méthodes de Fourier.

– P. Lascaux, R. Théodor : *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, ch. 7,9 (Masson, 1993).

– K.W. Morton, D.F. Mayers : *Numerical solution of partial differential equations* (Cambridge University Press, 2005).

– J.W. Thomas : *Numerical partial differential equations. Finite difference methods* (Springer, 1998).

– B. Lucquin, O. Pironneau : *Introduction au calcul scientifique* (Masson, 1996).

– W.H. Press, S.A. Teukolsky, W.T. Vettering, B.P. Flannery : *Numerical recipes, the art of scientific computing*, ch. 19 (Cambridge University Press, Cambridge, 2007).

– Sur le site `http://www.librecours.org/` :
  – L. Champaney : *Méthodes Numériques pour la Mécanique.*
  – É. Gonçalves : *Résolution numérique et discrétisation des EDP/EDO.*

– `http://www.freefem.org/`

13.6. Exercices

Exercice 1

a) On considère un système de n équations linéaires à n inconnues de la forme :

$$\begin{aligned} b_1U_1 - c_1U_2 &= d_1,\\ -a_2U_1 + b_2U_2 - c_2U_3 &= d_2,\\ &\cdots\\ -a_jU_{j-1} + b_jU_j - c_jU_{j+1} &= d_j,\\ &\cdots\\ -a_{J-1}U_{J-2} + b_{J-1}U_{J-1} &= d_{J-1}. \end{aligned}$$

On voit que l'équation numéro j ne fait intervenir que les trois inconnues numérotées $j-1, j$, et $j+1$. On a imposé de plus que $U_0 = U_J = 0$.

Pour appliquer la méthode de résolution de Gauss, il faut éliminer successivement chaque inconnue U_{j-1} dans chaque équation (ce qui rend le système « triangulaire »). Lorsque les k premières inconnues ont été éliminées, les k premières équations ont pris la forme

$$U_j - e_jU_{j+1} = f_j, \quad j = 1, 2, \ldots, k.$$

Le cas particulier $j = 0$ (ou $U_0 = 0$) impose que

$$e_0 = f_0 = 0.$$

Montrez que les nombres e_j et f_j peuvent se calculer à partir de e_{j-1} et f_{j-1} par les relations de récurrence

$$\begin{aligned} e_j &= \frac{c_j}{b_j - a_je_{j-1}};\\ f_j &= \frac{d_j + a_jf_{j-1}}{b_j - a_jc_{j-1}},\\ j &= 1, 2, \ldots, J-1. \end{aligned}$$

Une fois les e_j et f_j calculés, vérifiez que l'équation

$$U_j - e_jU_{j+1} = f_j, \quad j = J-1, J-2, \ldots, 1$$

permet de trouver les U_j, à partir de $U_J = 0$.

b) Écrire un programme pour résoudre un système linéaire tridiagonal. Vérifier le programme sur le système

$$\begin{aligned} 2x_1 + x_2 &= 1,\\ x_1 + 2x_2 + x_3 &= 2,\\ x_2 + 2x_3 &= 3, \end{aligned}$$

c) Utiliser cet algorithme pour résoudre l'équation de la chaleur dans le cas de la barre homogène exposé dans le texte.

Exercice 2

Programmer la résolution de l'équation des ondes selon l'algorithme du § 13.4 avec $c = 1$. On prendra comme conditions initiales

$$u(x,0) = f(x) = \exp[-k * (x - x_0)^2],$$
$$\left.\frac{\partial u}{\partial t}\right|_{t=0} = g(x) = 0,$$
$$0 \leq x \leq 1$$

et comme conditions aux limites

$$u(0,t) = u(1,t) = 0.$$

Le pas h, de l'ordre de 0,01, est constant, alors que le pas en temps, τ, pourra être choisi par l'utilisateur. Représenter graphiquement la solution U_i^j pour des valeurs de j variant de 10 en 10. Quelle sorte d'onde simule ce programme ? Quel est l'effet de la valeur de τ sur les résultats ?

13.7. PROJET

Équation de Laplace avec symétrie cylindrique

L'équation de Laplace peut être résolue numériquement par la méthode des différences finies (ou de discrétisation). L'application de ce procédé en coordonnées cylindriques est un peu plus subtile qu'en coordonnées cartésiennes. Dans tout le projet, on se limite à des problèmes à symétrie axiale (le potentiel est invariant par rotation autour de l'axe Oz). Dans ce cas, l'équation de Laplace s'écrit

$$\nabla^2\Phi = \frac{\partial^2\Phi}{\partial z^2} + \frac{\partial^2\Phi}{\partial r^2} + \frac{1}{r}\frac{\partial\Phi}{\partial r} = 0. \tag{13.12}$$

Cette équation est valable en dehors de l'axe ($r \neq 0$). On remplace maintenant chaque dérivée par une approximation d'ordre deux, en notant h le pas en r et ℓ le pas en z ; on pose encore $\beta = h/\ell$. L'équation (13.12) devient

$$\left(1 + \frac{h}{2r}\right)\Phi(r+h,z) + \left(1 - \frac{h}{2r}\right)\Phi(r-h,z) + \beta^2\Phi(r,z+\ell) + \beta^2\Phi(r,z-\ell)$$
$$= 2(1+\beta^2)\Phi(r,z). \tag{13.13}$$

Sur l'axe, les solutions physiquement acceptables et symétriques satisfont à la condition

$$\left.\frac{\partial\Phi}{\partial r}\right|_{r=0} = 0. \tag{13.14}$$

En appliquant la règle de l'Hospital, on peut alors écrire le terme apparemment divergent de l'équation (13.12) comme

$$\left.\frac{1}{r}\frac{\partial\Phi}{\partial r}\right|_{r=0} = \left.\frac{\partial^2\Phi}{\partial r^2}\right|_{r=0} \tag{13.15}$$

et l'équation de Laplace devient

$$2\frac{\partial^2\Phi}{\partial r^2} + \frac{\partial^2\Phi}{\partial z^2} = 0, \text{ si } r = 0. \tag{13.16}$$

On en déduit que, sur l'axe, l'équivalent discrétisé de (13.16) est

$$4\Phi(h,z) + \beta^2[\Phi(0,z+\ell) + \Phi(0,z-\ell)] = (4+2\beta^2)\Phi(0,z). \tag{13.17}$$

1. Justifier les affirmations contenues dans les équations (13.12) à (13.17).

2. Écrire un programme pour résoudre l'équation de Laplace en coordonnées cylindriques et en symétrie axiale par la méthode de relaxation, très voisine de l'algorithme de Gauss–Seidel. Pour cela on remplace la fonction continue $\Phi(r,z)$ par une fonction définie sur des points, $\phi_{i,j}$, l'indice i repérant les valeurs successives de r, l'indice j celles de z. En principe, les inconnues $\phi_{i,j}$ d'un système linéaire devraient être numérotées par un seul indice, ce qui conduit à des changements d'indice laborieux. Cette opération n'est pas nécessaire ici parce que, dans l'algorithme de relaxation, on utilise à chaque itération les valeurs les plus récentes des potentiels, quand elles sont disponibles ; il suffit de balayer régulièrement et toujours de la même façon les points (i,j). À chaque passage, on calcule $\phi_{i,j}^{(n+1)}$ grâce à l'aide des formules (13.13) et (13.17) et des valeurs $\phi_{i+1,j}, \phi_{i-1,j}, \phi_{i,j+1}, \phi_{i,j-1}$.

3. Essayer votre programme en calculant le potentiel entre deux cylindres coaxiaux infiniment longs ; le cylindre intérieur a un rayon a, il est au potentiel $V = 1$. Le cylindre extérieur a un rayon b, il est au potentiel 0. On peut simuler un cylindre infini en imposant que $\Phi(r,0)$ et $\Phi(r,L)$ soient identiques à la solution analytique. On peut supposer qu'au départ le potentiel varie linéairement entre les deux électrodes.

4. Modifier votre programme pour introduire la surrelaxation. On sait que la valeur « sur-relaxée » de ϕ est donnée par

$$\phi_{i,j}^{sr} = (1-\omega)\phi_{i,j}^{(n)} + \omega\phi_{i,j}^{(n+1)}, \tag{13.18}$$

   $\phi_{i,j}^{(n+1)}$ étant obtenu comme précédemment et ω de l'ordre de 1,7. Ensuite, on remplace $\phi_{i,j}^{(n+1)}$ par $\phi_{i,j}^{sr}$. Comment varie la vitesse de convergence avec ω ?

5. Traiter les deux cas représentés par les figures 13.5 a et b, qui représentent des coupes d'électrodes à symétrie cylindrique. Le conducteur extérieur est encore à $V = 0$, le conducteur intérieur à $V = 1$. Les dimensions sont données à titre indicatif ; elles pourront figurer en paramètres dans le programme. Prévoir de représenter les surfaces équipotentielles dans le plan (r,z).

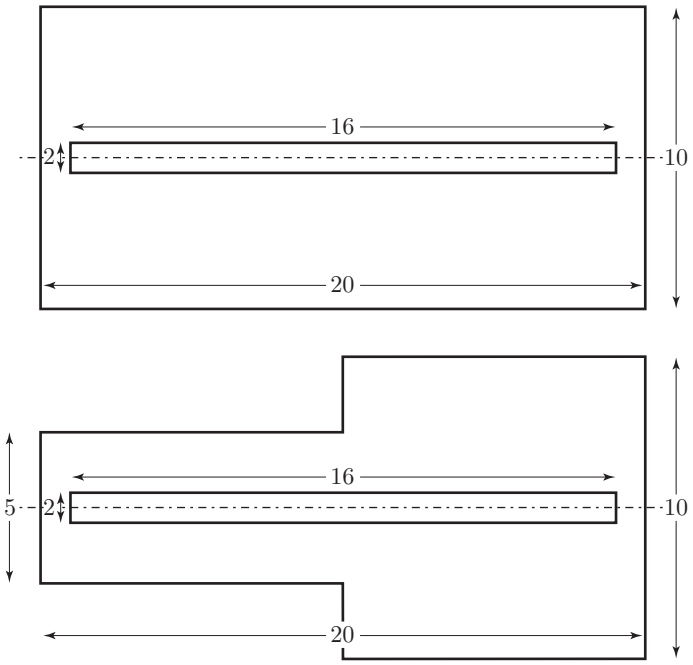


Figure 13.5 – Sections de deux systèmes de conducteurs cylindriques.

Chapitre 14

Probabilités et erreurs

Le hasard intervient souvent dans la vie courante et il en est de même en science. Cette intervention peut prendre des formes diverses. Nous pouvons être amenés à étudier un phénomène régi par le hasard (on dit souvent « aléatoire »), comme le désintégration d'un noyau radioactif ou la trajectoire d'une molécule d'azote dans l'atmosphère. Il peut arriver qu'un processus aléatoire vienne « parasiter » un phénomène régulier (ou « déterministe »). Ainsi, le signal très faible reçu d'une sonde interplanétaire est accompagné d'un « bruit » qui peut rendre sa compréhension difficile. Sur terre, toute mesure d'une grandeur physique ou chimique est entachée d'erreurs. Le fait d'avoir affaire au hasard n'empêche pas de raisonner de façon quantitative, mais il faut manipuler des probabilités et des valeurs moyennes plutôt que des données certaines. Dans ce chapitre, nous vous proposons un résumé de résultats utiles de la théorie des probabilités avant de les appliquer à l'estimation des erreurs de mesures et de leurs conséquences.

14.1. Probabilité

Nous supposons que les lecteurs ont des connaissances élémentaires en théorie des probabilités et nous ne ferons que rappeler quelques propriétés. Imaginons une expérience (on dit aussi un tirage, une épreuve) qui peut avoir un nombre fini de résultats distincts ; l'apparition (ou occurrence) de l'un de ces résultats (ou événement) est le fruit du hasard. C'est le cas du lancement d'une pièce de monnaie ou d'un dé. Les résultats possibles sont repérés par l'indice $i, 1 \leq i \leq n$. Soit p_i la probabilité du résultat i, avec $0 \leq p_i \leq 1$. Ces quantités sont normalisées, en ce sens que

$$\sum_{1}^{n} p_i = 1. \tag{14.1}$$

À chacun des résultats possibles du tirage, nous associons un nombre x_i (par exemple la valeur qui apparaît sur la face supérieure du dé). Nous pouvons encore imaginer une variable X qui prend la valeur x_i lorsque l'événement i se produit. X est une « variable aléatoire discrète ».

Remarque : Le terme traditionnel de « variable aléatoire » est mal choisi ; X est en fait une fonction du résultat de l'épreuve aléatoire.

On écrit souvent

$$p_i \equiv \text{Proba}(X = x_i).$$

Il peut être commode de considérer la probabilité cumulée ou fonction de répartition

$$P_i = \text{Proba}(X \leq x_i) = \sum_{j=1}^{i} p_j.$$

C'est une fonction croissante de zéro à un. L'espérance (ou valeur moyenne) de la grandeur aléatoire X est notée $\langle X \rangle$ (ou $\bar{X}$ ou μ_X) ; elle est définie comme

$$\langle X \rangle \equiv \sum_{1}^{n} p_i X_i. \tag{14.2}$$

C'est une indication, en général, de l'ordre de grandeur des valeurs possibles de X. Un autre paramètre important est l'écart-type σ tel que :

$$\sigma^2 \equiv \langle (X - \langle X \rangle)^2 \rangle; \tag{14.3}$$

σ^2 est appelée la variance de X. L'écart-type permet d'estimer la dispersion des valeurs prises par X.

Plus généralement, l'espérance d'une fonction de X est définie comme

$$\langle f(X) \rangle \equiv \sum_{1}^{n} p_i f(X_i). \tag{14.4}$$

Nous pouvons très bien imaginer des événements complexes (ou composés). Ainsi, lors du lancement d'un dé, le résultat « obtenir un nombre premier » est réalisé si le dé affiche 2 **ou** 3 **ou** 5. La probabilité de l'événement complexe est alors la somme des probabilités des résultats élémentaires qui le composent. Si le dé est équilibré, la probabilité d'apparition de chaque face est de $1/6$ et la probabilité de l'événement composé « nombre premier » est $1/2$. Lorsque les événements élémentaires sont « équiprobables », la probabilité d'un résultat complexe est proportionnelle au nombre d'événements élémentaires qui le compose (c'est le cas de l'exemple, $3 \times \frac{1}{6} = \frac{1}{2}$). On traduit souvent ce fait en disant que la probabilité d'un événement composé est le rapport du nombre de cas « favorables » au nombre de cas total ($3/6$ ici).

Ces définitions prennent une forme un peu différente lorsque la quantité X est une grandeur réelle, continue, définie dans un intervalle fini ou non. Il est commode d'introduire la fonction de répartition de X, qui est la probabilité d'observer une valeur de X inférieure ou égale à x :

$$P(x) \equiv \text{Proba}(X \leq x) \tag{14.5}$$

et la densité de probabilité de X qui est la dérivée de la fonction précédente :

$$p(x) \equiv P'(x). \tag{14.6}$$

On interprète p comme proportionnelle à la probabilité de trouver X dans un petit intervalle :

$$p(x)dx = \text{Proba}(x \leq X \leq x + dx). \tag{14.7}$$

La figure 14.1 montre les deux fonctions P et $P' = p$ et la représentation géométrique de la probabilité de trouver X dans l'intervalle [5,3 ... 6,4]. La densité de probabilité est normalisée :

$$\int p(x)dx = 1,$$

ce qui implique que la fonction de répartition P croît de 0 à 1 d'une extrémité à l'autre de l'intervalle de définition.

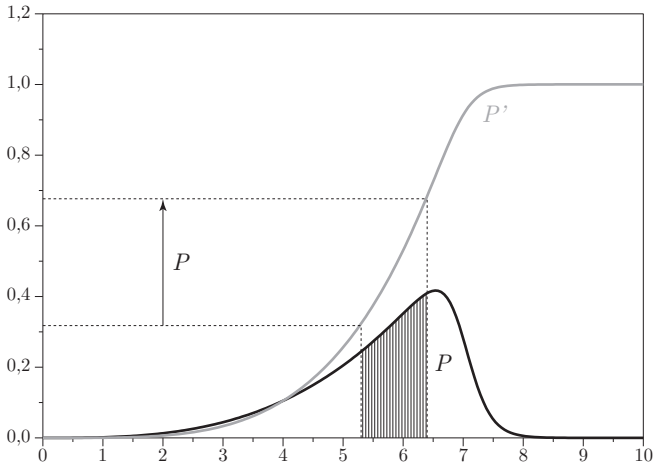


Figure 14.1 – Fonction de répartition et fonction densité de probabilité.

La valeur moyenne et l'écart-type sont alors définis comme :

$$\langle X \rangle \equiv \int xp(x)dx, \tag{14.8}$$

$$\sigma^2 \equiv \int (x - \langle X \rangle)^2 p(x)dx, \tag{14.9}$$

les intégrales étant étendues à tout le domaine permis de X. La valeur moyenne d'une fonction de X est

$$\langle f(X) \rangle \equiv \int f(x)p(x)dx. \tag{14.10}$$

14.2. Lois de probabilité

14.2.1. Loi binomiale

Soit une épreuve aléatoire n'ayant que deux résultats possibles : A et B ; si A ne se réalise pas, il y aura réalisation de l'événement complémentaire B (événements

mutuellement exclusifs). Soit p la probabilité de A, et donc $q = 1 - p$ celle de B. Procédons à n épreuves consécutives et indépendantes. Nous pouvons définir une variable aléatoire à valeurs entières X égale au nombre d'apparitions du résultat A sur l'ensemble des n épreuves. La probabilité pour que A se réalise exactement k fois (et donc B $n - k$ fois), sans tenir compte de l'ordre des événements, s'écrit :

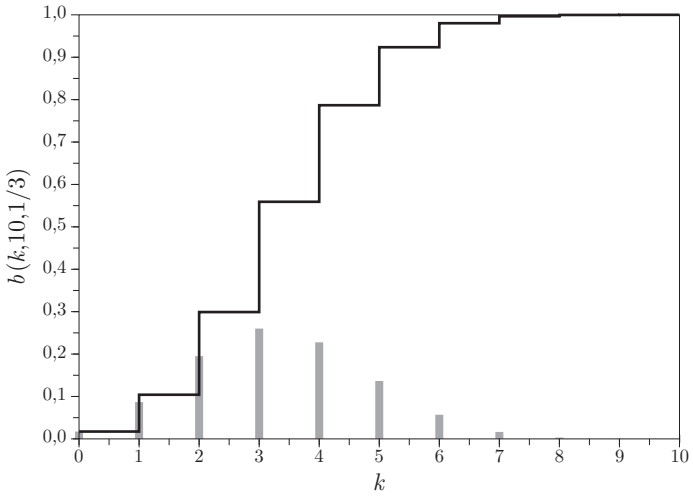


Figure 14.2 – La loi de probabilités $b(k, 10, 1/3)$ et la fonction de répartition correspondante.

$$\text{Proba}(X = k) \equiv b(k; n, p) = C_n^k p^k q^{n-k}. \tag{14.11}$$

Cette loi de probabilité est appelée loi binomiale, ou de Bernoulli. Le maximum de $b(k; n, p)$ est atteint approximativement pour $k = np \equiv k_m$. On montre aussi que pour la loi binomiale :

$$\langle X \rangle = np \quad ; \quad \sigma_X = \sqrt{npq}. \tag{14.12}$$

14.2.2. Loi de Poisson

Lorsque la probabilité $p \to 0$ et que $n \to \infty$, de telle sorte que $np \to \lambda$, la loi binomiale tend vers une forme limite, connue sous le nom de loi de Poisson, laquelle s'écrit en général

$$\text{Proba}(X = k) \equiv p(k; \lambda) \equiv e^{-\lambda} \frac{\lambda^k}{k!}. \tag{14.13}$$

À partir de cette définition, vous pourrez vérifier que les $p(k; \lambda)$ sont normalisées et que

$$\langle X \rangle = \lambda \; ; \quad \sigma_X = \sqrt{\lambda}. \tag{14.14}$$

Cette loi de probabilité intervient dans tous les problèmes de comptage en radioactivité comme en microbiologie.

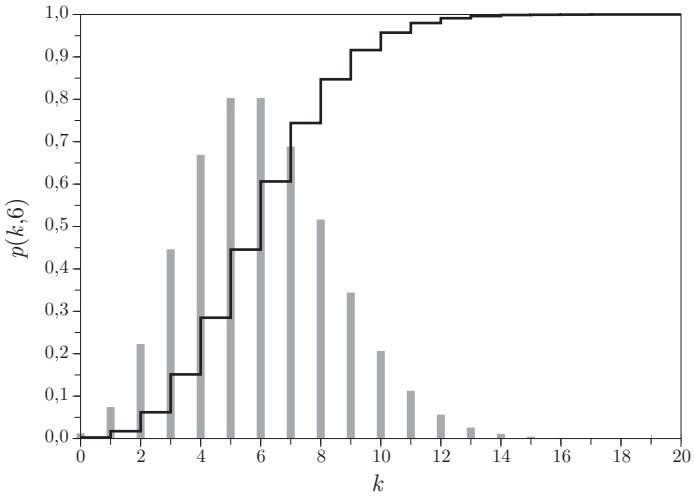


Figure 14.3 – Probabilités de Poisson $p(k, 6)$ et fonction de répartition correspondante. Les valeurs des $p(k)$ ont été multipliées par 5 pour le tracé.

14.2.3. Loi uniforme

Soit une variable aléatoire X qui ne peut prendre que des valeurs réelles comprises dans l'intervalle fini $[a, b], a < b$; nous supposons de plus que toutes les valeurs permises sont équiprobables. Comme la densité de probabilité doit être constante dans l'intervalle et normalisée à un, nous déduisons que

$$p(x) = \frac{1}{b-a} \; ; \quad P(x) = \frac{x-a}{b-a} \; ; \quad x \in [a, b].$$

On montre que

$$\langle X \rangle = \frac{a+b}{2} \; ; \quad \sigma = \frac{b-a}{\sqrt{12}}. \tag{14.15}$$

14.2.4. Loi normale ou de Gauss

Lorsque le nombre d'épreuves n et le nombre de « succès » k deviennent grands (en pratique supérieurs à 10), la loi binomiale tend vers une limite appelée loi de Gauss ou loi normale. Celle-ci est définie par

$$N(x; \mu, \sigma) \equiv \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left[-\frac{(x'-\mu)^2}{2\sigma^2}\right] dx'. \tag{14.16}$$

La probabilité pour que X soit compris entre x et $x + dx$ est proportionnelle à la densité de probabilité $n(x) = N'(x)$:

$$n(x)dx = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] dx. \tag{14.17}$$

Il est commode d'introduire la loi normale réduite (ou standard) pour laquelle $\sigma = 1$ et $\langle x \rangle = 0$, représentée figure 14.4.

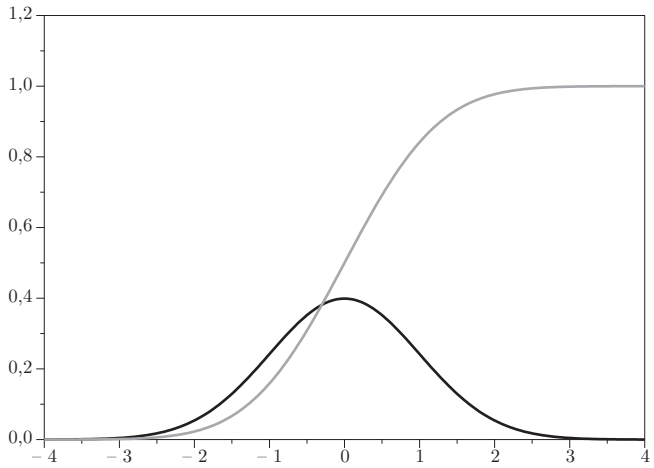


Figure 14.4 – Fonction de répartition et densité de probabilité de la loi normale réduite.

Vous pourrez montrer directement que, comme on pouvait s'y attendre, la moyenne de X est μ et que son écart-type est σ.

14.2.5. Loi du χ^2 ou de Pearson

Si $X_1, X_2, \ldots, X_n$ sont n variables aléatoires indépendantes, chacune distribuée selon la loi normale standard (réduite) de moyenne 0 et d'écart-type 1, alors la quantité $\sum X_i^2$ est distribuée selon une loi dite « du χ^2 » (prononcez qui-deux ou qui-carré) « à n degrés de liberté ».

La densité de probabilité correspondante est :

$$p(x,n) = \frac{x^{n/2-1}e^{-x/2}}{2^{n/2}\Gamma(n/2)}. \tag{14.18}$$

La « fonction gamma », $\Gamma(x)$, est l'intégrale :

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt.$$

Elle obéit à la relation de récurrence

$$\Gamma(x+1) = x\Gamma(x)$$

et, de plus,

$$\Gamma(1/2) = \sqrt{\pi}.$$

Cette fonction est contenue dans Scilab, sous le nom de `gamma(x)`. Nous l'utilisons dans le programme ci-contre (les lignes 7 et 12 ne servent qu'à mettre en place la légende).

Listing 14.1 – Calcul de le densité de probabilité pour le χ^2

```
deff("y = chi2(x,n)",
     "y = x.^(n/2-1).*exp(-x/2)/(2^(n/2)*gamma(n/2))")

xmax = 20; npt = 200;
x = linspace(0,xmax,npt);

a = gca();
plot2d(x,chi2(x,5),rect = [0,0,xmax,0.2]);
plot2d(x,chi2(x,9),2,"000")
plot2d(x,chi2(x,16),5,"000")

h1 = legend(['n = 5';'n = 9';'n = 16'])
```

Vous pourrez montrer sans calcul que $\mu_X = n$ et que $\sigma_X^2 = 2n$. La figure 14.5 montre trois exemples de la loi du χ^2.

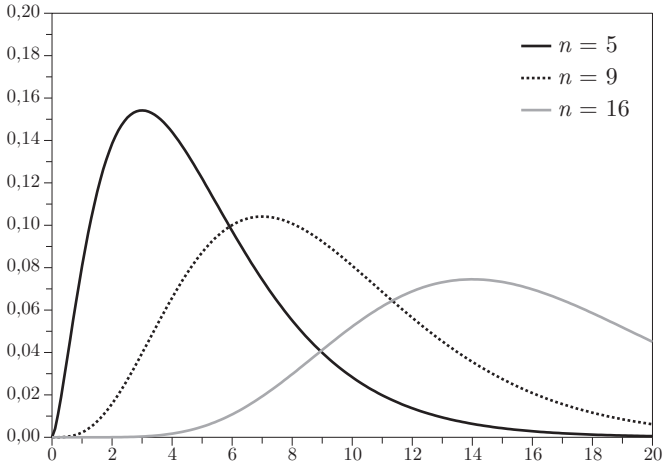


Figure 14.5 – Densité de probabilité pour la loi du χ^2 à 5, 9 ou 16 degrés de liberté.

On utilise beaucoup plus souvent la fonction de répartition $P(\chi^2, n) \equiv \text{Proba}(X \leq \chi^2)$ ou la fonction complémentaire $Q(\chi^2, n) \equiv \text{Proba}(X \geq \chi^2) = 1 - P$, que l'on trouve tabulées dans les ouvrages de statistique.

14.2.6. Paramètres de la loi de probabilité et paramètres de l'échantillon

Dans la pratique, nous disposons d'un certain nombre n d'observations d'une grandeur Y, réparties au hasard. L'ensemble des y_i constitue un échantillon extrait de la « population » de toutes les valeurs possibles de Y. La variable aléatoire Y obéit à une certaine loi de probabilité $p(y)$, à laquelle sont associées une moyenne μ et un écart-type σ. Nous n'aurons jamais connaissance de ces paramètres, mais nous

pourrons essayer de les estimer à partir de propriétés de l'échantillon. Il importe de ne pas confondre paramètres de la population et paramètres de l'échantillon. Cette distinction est bien connue dans le cas des sondages. Les instituts de sondage s'efforcent de sélectionner un « échantillon représentatif » de telle sorte que la réponse de l'échantillon soit aussi proche que possible de la réponse (inconnue) de la population entière. En fait, on a les résultats suivants :

- La meilleure estimation de la moyenne μ de la population parente est la moyenne m de l'échantillon ($m = (1/n)\sum y_i$).
- La meilleure estimation de l'écart-type σ de la population parente est égale à l'écart-type s de l'échantillon, défini par :

$$s^2 = \frac{1}{n-1}\sum_1^n (y_i - m)^2. \tag{14.19}$$

Le facteur $n-1$ (au lieu de n que l'on pourrait attendre) tient compte de ce qu'une partie de l'information contenue dans l'échantillon a été « consommée » pour calculer m. On dit que les nombres m et s sont respectivement des estimateurs de μ et σ. Ce sont des variables aléatoires puisque ce sont des combinaisons de valeurs de la variable aléatoire Y et un autre échantillon donnerait d'autres valeurs de m et s. On montre que les estimateurs m et s^2 sont sans biais, c'est-à-dire que $\langle m \rangle = \mu$ et que $\langle s^2 \rangle = \sigma^2$.

14.2.7. Vérification d'une loi de probabilité

Nous disposons d'un échantillon d'une variable aléatoire Y : sommes-nous capables de déterminer la loi de probabilité à laquelle obéit Y ? Non, pas plus que nous ne pourrions trouver rigoureusement la moyenne ou l'écart-type de Y. La seule approche possible est la suivante. Nous choisissons une loi de probabilité $p(y)$ plausible et nous essayons de vérifier l'hypothèse (dite hypothèse nulle ou H_0) : « les résultats expérimentaux sont compatibles avec la loi p, au seuil de confiance α ». Nous ne considérons ici qu'un exemple élémentaire. Soit une épreuve conduisant à l'un ou l'autre de r résultats mutuellement exclusifs dont les probabilités (inconnues) sont $p_1, p_2, \ldots, p_r$. Par ailleurs, nous désignons par q_i notre estimation de la probabilité du résultat i. Nous allons donc essayer de vérifier l'hypothèse $q_i = p_i, 1 \leq i \leq r$. Nous répétons n fois l'épreuve en appelant f_i le nombre d'apparitions du résultat i. La quantité f_i/n est proche de p_i (loi des grands nombres) et, si l'hypothèse est valable, elle est aussi voisine de q_i. La théorie des probabilités nous fournit le théorème suivant. La variable S, définie comme

$$S = \sum_1^r \frac{(Y_i - np_i)^2}{np_i}$$

obéit à la loi de Pearson à $r-1$ degrés de liberté, à condition que les nombres np_i soient assez grands ($np_i \geq 10$ en pratique). Comme nous ignorons les p_i, nous allons employer, à la place de S, l'estimateur

$$T = \sum_1^r \frac{(f_i - nq_i)^2}{nq_i}, \tag{14.20}$$

qui obéit à peu près à la même loi de répartition. Il est clair que T sera petit si l'hypothèse est vérifiée (il obéit approximativement à une loi de Pearson à $r-1$ degrés de liberté). Choisissons un seuil α ; T' étant une variable distribuée selon le chi-deux à $r-1$ degrés de liberté, nous déterminons un nombre χ^2 vérifiant

$$\text{Proba}(T' \geq \chi^2) \leq \alpha.$$

Nous rejetons l'hypothèse si $T \geq \chi^2$. En d'autres termes, si l'écart entre fréquences observées et fréquences calculées nous paraît trop grand pour être dû au hasard, nous devrons conclure que notre hypothèse est fausse.

Exemple – Nous avons lancé 120 fois un dé, pour obtenir les résultats consignés ci-dessous. Nous faisons l'hypothèse que le dé est équilibré, ou encore que la probabilité de chaque face est 1/6.

Rang i	nombres observés	nombres calculés	$(y_i - nq_i)^2/nq_i$
1	25	20	1,25
2	21	20	0,05
3	16	20	0,8
4	28	20	3,2
5	14	20	1,8
6	16	20	0,8
	total : 120	total : 120	$T = 7{,}9$

Nous constatons, à la lecture d'une table du χ^2, que la probabilité pour que T (avec 5 degrés de liberté) soit au moins égal à 7,9 est comprise entre 0,1 et 0,2 (cf. figure 14.6). L'hypothèse est considérée comme plausible et elle est acceptée.

Le tableur Excel (entre autres) permet d'obtenir ce résultat en quelques pressions de touches, une fois que les données ont été saisies. La fonction « LOI.CHI-DEUX(A2 ;n) » renvoie la probabilité d'apparition d'une valeur de T' au moins égale au contenu de la cellule « A2 » pour un nombre de degrés de liberté égal à n ; le résultat est ici 0,1618. La fonction « TEST.CHI-DEUX » effectue automatiquement l'ensemble des calculs, à condition de fournir en arguments la plage de cellules contenant les résultats observés et la plage destinée à contenir les résultats attendus.

14.3. Erreurs

Lorsque nous cherchons à mesurer pratiquement une grandeur, taille d'une vitre, poids d'un sac de pommes de terre, concentration d'une solution, brillance d'une lampe ou autre, nous obtenons un résultat plus ou moins éloigné de la « vraie » valeur.

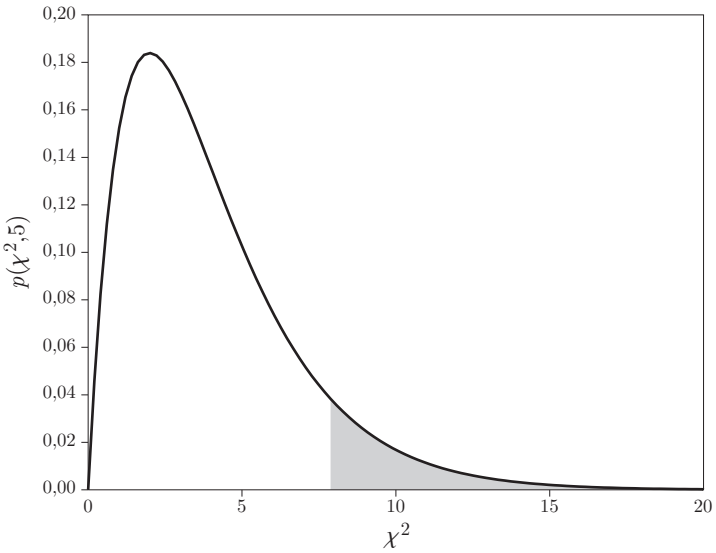


Figure 14.6 – Cas particulier de loi de χ^2. L'aire en gris représente la probabilité pour que, sous l'effet du hasard, la quantité χ^2 soit supérieure à 7,9.

En répétant la mesure ou l'expérience, en améliorant la technique, la méthode et les instruments de mesure, nous avons l'espoir (souvent vérifié) que la part d'erreur va diminuer et que les résultats vont s'approcher asymptotiquement d'une estimation raisonnable et sûre. Des considérations de ce genre s'appliquent à toute activité et en particulier aux mesures physiques, qui sont toujours entachées d'erreurs et d'incertitudes, lesquelles doivent être réduites par des perfectionnements de la technique et aussi estimées pour établir la validité des résultats. Il est commode d'établir une classification des erreurs qui guettent la personne qui effectue des mesures. On distingue ainsi les erreurs systématiques, qui découlent par exemple d'un mauvais étalonnage d'un instrument ou d'un biais de l'observateur. Si, pour prévoir l'issue d'un prochain scrutin présidentiel, nous n'interrogions que des personnes de sexe masculin, d'âge supérieur à 50 ans et dont les revenus sont supérieurs à 40 000 euros/an, nous aboutirions à une prédiction « biaisée » , entachée d'une erreur de méthode. De même la mesure des dimensions d'une pièce avec une règle tordue, la mesure précise d'une masse sans tenir compte de la poussée d'Archimède de l'air... D'autre part, malgré la bonne volonté de la personne effectuant les mesures, il peut s'introduire des erreurs non reproductibles, aléatoires. C'est par exemple le cas lorsque la sensibilité de l'appareil est insuffisante et que le signal cherché est accompagné de « bruit » ou de fluctuations. Les erreurs aléatoires affectent la précision de la mesure : plus elles sont importantes, plus les mesures sont dispersées. Les erreurs systématiques affectent la justesse du résultat.

Si la dispersion se détecte aisément, le biais est moins visible ; en toute rigueur, il faudrait, pour le mettre en évidence, disposer d'une mesure juste et indépendante de la quantité en question. On augmente la précision d'une expérience en répétant celle-ci plusieurs fois et en retenant la moyenne des résultats individuels. On améliore la justesse en analysant avec soin la méthode et en cherchant toutes les causes d'erreurs possibles.

La figure 14.7 illustre les notions de justesse (absence de biais) et de précision (dispersion ou bruit faible). Nous imaginons que nous avons fait de nombreuses (100) mesures d'une même grandeur et nous avons représenté nos résultats sous forme d'histogramme. Nous supposons aussi que nous connaissons la valeur exacte de la quantité à mesurer (prise ici égale à zéro) : il s'agit bien sûr d'un artifice pédagogique.

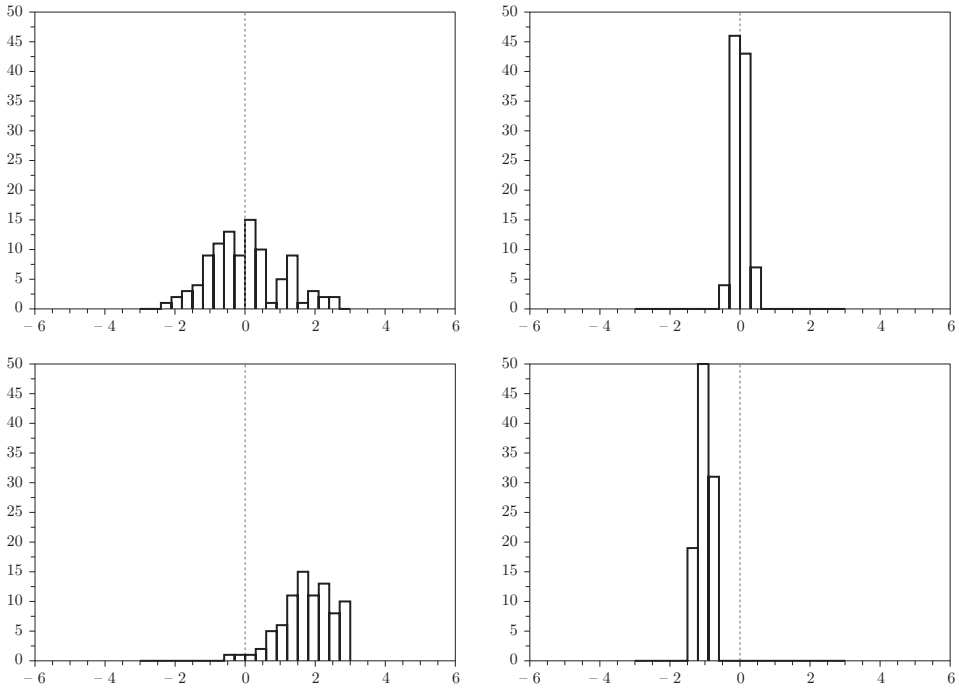


Figure 14.7 – Justesse et précision d'une série de mesures. De haut en bas et de gauche à droite : mesures justes et imprécises, mesures justes et précises, mesures biaisées et imprécises, mesures biaisées et précises.

Dans les paragraphes qui suivent, nous considérerons a priori que nous disposons de mesures parfaitement justes, seulement affectées d'erreurs aléatoires. Cette situation permet de faire des prédictions assez simplement. Le résultat d'une mesure est considéré comme le résultat du tirage d'une variable aléatoire, superposition de la valeur « vraie » (non fluctuante ou certaine) et de l'erreur ; nous supposerons que les erreurs sont additives, c'est-à-dire qu'un résultat Y est la somme de la valeur vraie inconnue y et d'une erreur ϵ : $Y = y + \epsilon$. Ce n'est pas toujours le cas. Lorsque nous mesurons l'absorbance d'une solution à l'aide d'un spectromètre dont la source lumineuse a une fréquence fluctuante, nous sommes confrontés à une erreur non additive ; la théorie correspondante ne sera pas abordée ici.

14.4. Propagation des erreurs

Nous nous intéressons maintenant à une grandeur physique G qui est déterminée indirectement à partir des mesures de deux grandeurs U et V. Sachant que les valeurs de U et de V sont entachées d'erreurs aléatoires, nous nous demandons quelle peut être la dispersion de $G = g(U, V)$. La probabilité d'obtenir, lors d'une mesure particulière, le résultat (u, v) dépend de la « densité de probabilité conjointe » $f(u, v)$. Plus précisément, la probabilité pour que U soit dans l'intervalle $[u, u+du]$ **et** V dans $[v, v + dv]$ vérifie :

$$\text{Proba}(u \leq U \leq u + du; v \leq V \leq v + dv) = p_2(u, v)dudv.$$

Il faut encore généraliser la notion de valeur moyenne :

$$\langle g(U, V)\rangle = \int\int g(u, v)p_2(u, v)dudv.$$

La probabilité pour que U soit dans l'intervalle $[u, u + du]$, cela **quel que soit** V, s'écrit :

$$\text{Proba}(u \leq U \leq u + du; \forall V) = p_1(u)du = \left[\int p_2(u, v)dv\right] du.$$

Remarque : On peut rendre ce résultat plausible par le « raisonnement » approché suivant. Le fait de mesurer U dans l'intervalle $[u, u + du]$ quel que soit V est un événement composé qui est la réunion des événements élémentaires trouver u, v dans $[u, u + du; v_0, v_0 + dv]$, ou dans $[u, u + du; v_0 + dv, v_0 + 2dv]$, ou dans $[u, u + du; v_0 + 2dv, v_0+3dv]$, etc. . . . D'autre part, la probabilité de l'événement composé est la somme des probabilités des événements élémentaires et cette somme devient une intégrale lorsque l'intervalle dv tend vers zéro.

À partir de la densité p_1, nous pouvons calculer la valeur moyenne $< U >$ et l'écart-type σ_u de U, quelque soit V ; des formules symétriques existent pour la variable V.

Nous introduisons ensuite les « moments » de la distribution p_2 :

$$m_{p,q} \equiv \langle U^p V^q\rangle = \int\int p_2(u, v)u^p v^q dudv,$$

puis les « moments centrés » $\mu_{p,q} = \langle(U - \langle U\rangle)^p(V - \langle V\rangle)^q\rangle$. Nous trouvons en particulier :

$$\mu_{2,0} = \sigma_u^2; \qquad \mu_{0,2} = \sigma_v^2.$$

Le moment centré $\mu_{1,1} \equiv \langle(U - \langle U\rangle)(V - \langle V\rangle)\rangle$ est souvent appelé la covariance de U et V. Les variables aléatoires U et V sont dites corrélées si $\mu_{1,1} \neq 0$.

Bien entendu, nous ne disposons pas des « vraies » valeurs de U et V ; nous ne connaissons que des estimateurs u^* et v^* (qui pourraient être les moyennes de mesures de U et V). Ces estimateurs sont supposés sans biais : $u^* = \langle U\rangle$ et $v^* = \langle V\rangle$. Il est raisonnable de choisir $G^* = g(u^*, v^*)$ comme estimateur de G. Nous allons estimer la moyenne et la variance de G^*. Faisons un développement limité de $g(U, V)$ autour du point u^*, v^* :

$$g(U, V) = g(u^*, v^*) + (U - u^*)g_u + (V - v^*)g_v + \ldots$$

Dans cette expression, les dérivées ($g_U = \partial g/\partial U, g_V = \partial g/\partial V$) sont calculées en u^*, v^*. La valeur moyenne de cette expression vaut, sachant que u^* et v^* sont sans biais,

$$\langle g(U,V)\rangle = g(u^*, v^*)$$

en négligeant les termes d'ordre supérieur. G^* est donc lui-même, à cette approximation, un estimateur non biaisé.

Cherchons maintenant la variance de G^*

$$\sigma^2_{G*} = \langle [G(u^*, v^*) - G(U,V)]^2\rangle = \langle [(u^* - U)g_U + (v^* - V)g_V]^2\rangle$$

et, en développant la valeur moyenne du carré :

$$\sigma^2_{G^*} = \mu_{2,0} g_U^2 + 2\mu_{1,1} g_U g_V + \mu_{0,2} g_V^2.$$

Cette formule donne l'écart-type sur G^* connaissant les moments centrés de U et V ; on l'écrit en général en fonction des écarts types :

$$\sigma^2_{G^*} = \sigma_u^2 g_U^2 + 2\sigma_{u,v} g_U g_V + \sigma_v^2 g_V^2 \tag{14.21}$$

où nous avons introduit la notation habituelle $\sigma_{uv} \equiv \mu_{1,1}$ pour la covariance. En pratique, on ne connaît pas les $\mu_{i,j}$ mais seulement des estimateurs construits à partir de résultats expérimentaux. Si les observations de U et V ne sont pas corrélées, alors $\sigma_{uv} = 0$ et la formule précédente se simplifie :

$$\sigma^2_{G^*} = \sigma_u^2 \left(\frac{\partial g}{\partial U}\right)^2 + \sigma_v^2 \left(\frac{\partial g}{\partial V}\right)^2. \tag{14.22}$$

Exemple – Considérons les fonctions $x = uv$ et $y = u/v$; les variances correspondantes sont :

$$\sigma_x^2 = \sigma_u^2 v^2 + \sigma_v^2 u2 + 2\sigma_{uv}^2 uv \quad ; \quad \sigma_y^2 = \sigma_u^2/v^2 + \sigma_v^2 u^2/v^4 - 2\sigma_{uv}^2 u/v^3$$

ce que l'on écrit en général de façon plus symétrique ($z = uv$ ou $z = u/v$) :

$$\left(\frac{\sigma_z}{z}\right)^2 = \left(\frac{\sigma_u}{u}\right)^2 + \left(\frac{\sigma_v}{v}\right)^2 \pm 2\left(\frac{\sigma_{uv}}{uv}\right)^2 \tag{14.23}$$

Dans le cas de variables non corrélées, on retrouve un résultat qui évoque l'addition des erreurs relatives que l'on enseigne dans les cours élémentaires.

14.5. Méthode du maximum de vraisemblance

Un noyau radioactif a une certaine probabilité de désintégration par unité de temps, notée $1/\tau$; en conséquence, pour un grand nombre de noyaux, le nombre de désintégrations par unité de temps, dN/dt, (la fréquence de l'événement) est proportionnel au nombre de noyaux :

$$dN/dt = -N(t)/\tau,$$

équation différentielle dont la solution s'écrit $N(t) = N_0 e^{-t/\tau}$. La probabilité de désintégration d'un noyau entre les instants t et $t + dt$ est $p(t) = (1/\tau)e^{-t/\tau}$. Comment déterminer la « durée de vie » τ ? Nous pouvons (en principe au moins) mesurer les dates $t_1, t_2, \ldots, t_N$ auxquelles chacun de ces N noyaux disparaît. Si nous recommençons l'expérience avec une nouvelle famille de N noyaux, nous obtiendrons des valeurs des t_i différentes des précédentes : ces fluctuations sont dues à la nature aléatoire du processus radioactif et existent même si les erreurs de mesure sont négligeables. La probabilité $V(\tau)$ d'observer effectivement les désintégrations aux époques $t_1, t_2, \ldots$ s'écrit (événements indépendants) :

$$V(\tau) = \prod_i^N p(t_i) = \exp\left\{-\frac{1}{\tau}\sum_1^N t_i - N\ln\tau\right\}.$$

Le meilleur choix de τ est celui qui maximise V, c'est à dire celui qui rend le plus vraisemblable possible le résultat effectivement observé. Le maximum de V est atteint lorsque $-\ln V$ est minimal, soit pour la valeur :

$$\tau^* = \frac{1}{N}\sum_1^N t_i.$$

Autrement dit, une valeur de τ égale à la moyenne des t_i calculée sur l'échantillon rend maximale la probabilité d'apparition de cet ensemble de valeurs. On dit que V est une fonction de vraisemblance et que τ^* a été choisi selon un critère de maximum de vraisemblance.

L'exemple qui précède est particulier : si les mesures ne sont pas reproductibles, c'est à cause de la nature aléatoire du phénomène. De plus, le nombre de désintégrations par seconde est donné par une loi exponentielle. Dans beaucoup d'autres cas, les mesures vont être entachées d'erreurs expérimentales. Examinons un autre exemple simple d'application du principe de maximum de vraisemblance.

Nous supposons avoir réalisé m déterminations de la grandeur X, que les conditions sont un peu différentes d'une mesure à l'autre, si bien qu'à chaque mesure est associée une incertitude (écart-type) σ_i particulière. La probabilité d'apparition de la valeur x_i est, comme très souvent pour des erreurs expérimentales, donnée par une loi de Gauss, de valeur moyenne μ et de variance σ_i^2 :

$$p(x_i) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x_i-\mu)^2}{2\sigma_i^2}\right)$$

Nous nous demandons quelle est la « meilleure » valeur de μ, en supposant, pour simplifier le raisonnement, que les σ_i sont connus. Pour répondre, nous construisons une fonction de vraisemblance $V(\mu)$, égale au produit des $p(x_i)$, et qui représente la probabilité d'observer l'ensemble des résultats x_i. Comme au paragraphe précédent, nous allons chercher la valeur de μ qui rend maximale la probabilité V, ce qui revient à chercher le maximum de $\ln V$ (ou le minimum de l'argument de l'exponentielle). Il vient :

$$V(\mu) = \left[\prod_1^m \frac{1}{\sigma_i\sqrt{2\pi}}\right]\exp\left[-\frac{1}{2}\sum_1^m\left(\frac{x_i-\mu}{\sigma_i}\right)^2\right].$$

En annulant la dérivée par rapport à μ de l'exposant, nous trouvons

$$\mu^* = \sum_1^m \frac{x_i}{\sigma_i^2} \Bigg/ \sum_1^m \frac{1}{\sigma_i^2}$$

La valeur la plus vraisemblable de μ est la moyenne pondérée des valeurs observées x_i, les poids étant les inverses des variances relatives à chaque mesure. Nous pouvons même déterminer l'incertitude sur μ : cette quantité dépend en effet des observations x_i et obéit à la loi de propagation des erreurs (nous supposons les mesures indépendantes, donc les corrélations nulles) :

$$\sigma_\mu^2 = \sum_1^m \left(\frac{\partial \mu}{\partial x_i}\right)^2 \sigma_i^2.$$

Un calcul simple montre que

$$\sigma_\mu^2 = \frac{1}{\displaystyle\sum_1^m \frac{1}{\sigma_i^2}}. \tag{14.24}$$

Lorsque toutes les incertitudes sont égales à σ, cette relation se réduit à

$$\sigma_\mu = \frac{\sigma}{\sqrt{m}}. \tag{14.25}$$

> L'incertitude sur la valeur moyenne d'une série de mesures équivalentes décroît comme l'inverse de la racine carrée du nombre de mesures indépendantes.

14.6. Méthode des moindres carrés

Nous allons maintenant appliquer la méthode du maximum de vraisemblance à un problème plus général.

Remarque : Il est permis de considérer la méthode (on dit souvent le principe) du maximum de vraisemblance comme compliquée et peu convaincante. C'est pourquoi bien des auteurs, suivant en cela Gauss, préfèrent se référer plus directement à un « principe des moindres carrés ». Les deux démarches sont très généralement équivalentes.

Nous imaginons que nous avons mesuré la variation d'une grandeur physique y en fonction d'une autre grandeur indépendante x : par exemple la susceptibilité magnétique d'un matériau en fonction de la température. Nous disposons donc d'une série de résultats expérimentaux $\{x_i, y_i\}, i = 1, 2, \ldots, m$. Les valeurs $\{x_i\}$ de la variable indépendante sont supposées parfaitement exactes, alors que les mesures de y sont entachées d'erreurs aléatoires. D'autre part, nous avons des raisons de penser que y est lié à x par une loi physique de la forme :

$$y = f(x, a_1, a_2, \ldots, a_n)$$

où les a_k sont des paramètres constants. Autrement dit, nous avons un modèle du phénomène physique ; ce modèle comporte n paramètres qu'il s'agit de déterminer. Nous supposons que les erreurs qui affectent chaque mesure y_i sont additives, indépendantes et réparties selon une loi normale. Nous allons donc utiliser le modèle

$$Y = f(x, a_1, a_2, \ldots, a_n) + \epsilon.$$

Nous supposons encore que ce modèle est juste ou non biaisé, c'est-à-dire que $\langle \epsilon \rangle = 0$ ou encore que la moyenne de la distribution $p(y_i)$ est $f(x_i, a_1, \ldots) = f_i = \langle Y_i \rangle$.

Nous voulons trouver les valeurs des a_k telles que la loi précédente représente au mieux l'ensemble des résultats expérimentaux $\{x_i, y_i\}, i = 1, \ldots, m$. Avec les hypothèses faites, nous savons écrire la probabilité d'apparition conjointe des événements $y_1, y_2, \ldots, y_m$:

$$V(a_1, a_2, \ldots, a_n) = \prod_{i=1}^{m} \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left[-\frac{(y_i - f_i)^2}{2\sigma_i^2}\right]$$

ou encore

$$V = \exp\left(-\frac{\mathcal{S}}{2}\right) \prod_{1}^{m} \frac{1}{\sigma_i \sqrt{2\pi}} \; ; \quad \mathcal{S} \equiv \sum_{1}^{m} \frac{(y_i - f_i)^2}{\sigma_i^2}$$

La quantité $\mathcal{S}$ joue un grand rôle dans ce formalisme : c'est la somme des carrés des écarts entre valeurs observées (y_i) et valeurs calculées (f_i), chaque terme étant pondéré par l'inverse du carré de l'écart-type. Le maximum de V (la vraisemblance) est atteint lorsque l'exposant est minimal. On rejoint ici le « principe des moindres carrés » qui stipule que les meilleures valeurs des paramètres sont celles qui minimisent la somme (pondérée) des carrés des écarts entre valeurs expérimentales et valeurs théoriques.

Le maximum de V et le minimum de $\mathcal{S}$ sont atteints quand

$$\frac{\partial \mathcal{S}}{\partial a_k} = -2 \sum_{i=1}^{m} \frac{y_i - f_i}{\sigma_i^2} \frac{\partial f_i}{\partial a_k} = 0, \quad k = 1, 2, \ldots, n.$$

14.6.1. Ajustement sur une fonction affine

Pour commencer, nous considérons un modèle dépendant linéairement de deux paramètres (et linéairement de x, bien que cette hypothèse ne soit pas nécessaire). Nous dirons que nous ajustons les paramètres d'un modèle linéaire ou encore que nous « lissons » les données à l'aide d'une droite. Le modèle s'écrit

$$Y = f(x, a, b) + \epsilon = ax + b + \epsilon$$

Comme $\partial f / \partial a = x$ et que $\partial f / \partial b = 1$, les conditions du maximum de vraisemblance sont alors

$$\sum_{i=1}^{n} \frac{y_i - f_i}{\sigma_i^2} = 0 \; ; \quad \sum_{i=1}^{n} x_i \frac{y_i - f_i}{\sigma_i^2} = 0.$$

Ces relations constituent en fait un système de deux équations linéaires à deux inconnues (a et b). Il est commode de poser

$$S = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \; ; \quad S_x = \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2} \; ; \quad S_{xx} = \sum_{i=1}^{n} \frac{x_i^2}{\sigma_i^2} \; ; \quad S_y = \sum_{i=1}^{n} \frac{y_i}{\sigma_i^2} \; ; \quad S_{xy} = \sum_{i=1}^{n} \frac{x_i y_i}{\sigma_i^2}.$$

Le système mentionné ci-dessus s'écrit explicitement :

$$\left\{ \begin{array}{lcl} aS_{xx} + bS_x & = & S_{xy}, \\ aS_x + bS & = & S_y. \end{array} \right. \tag{14.26}$$

Ces équations sont appelées « équations normales » ou « équations de Gauss ». Le tableau des coefficients est symétrique et les valeurs expérimentales (y_i) n'apparaissent qu'au second membre. La solution est élémentaire ; en posant $\Delta = SS_{xx} - S_x^2$, nous trouvons

$$a^* = \frac{1}{\Delta}(SS_{xy} - S_x S_y) \; ; \quad b^* = \frac{1}{\Delta}(S_{xx} S_y - S_x S_{xy}). \tag{14.27}$$

Les coefficients a^* et b^* sont des fonctions des variables aléatoires y_i : ce sont donc elles-mêmes des variables aléatoires qui nous servent à estimer les « vraies » a et b. Pour apprécier la précision (ou l'incertitude) de a^* et de b^*, il faut calculer l'écart-type de ces paramètres, au moyen de la formule de propagation des erreurs (14.22). Pour $u = a^*$ ou b^*, nous savons que

$$\sigma_u^2 = \sum_1^m \left(\frac{\partial u}{\partial y_i} \right)^2 \sigma_i^2.$$

Nous calculons les dérivées partielles à l'aide des formules (14.27) :

$$\frac{\partial a}{\partial y_i} = \frac{x_i S - S_x}{\sigma_i^2 \Delta} \; ; \frac{\partial b}{\partial y_i} = \frac{S_{xx} - x_i S_x}{\sigma_i^2 \Delta}.$$

Nous obtenons ensuite

$$\sigma_{a*}^2 = \frac{S}{\Delta} \; ; \quad \sigma_{b*}^2 = \frac{S_{xx}}{\Delta}. \tag{14.28}$$

Cette méthode élémentaire ne permet pas de trouver la covariance de a^* et b^* ; citons simplement le résultat

$$\sigma_{a*b*} = -S_x/\Delta. \tag{14.29}$$

Les formules (14.27), (14.28) et (14.29) se trouvent assez couramment programmées sur les calculettes.

D'autre part, le système linéaire des équations normales définissant a^* et b^* fait intervenir la matrice :

$$\boldsymbol{M} = \left[\begin{array}{cc} S_{xx} & S_x \\ S_x & S \end{array} \right],$$

Nous savons déjà que $\det \boldsymbol{M} = \Delta$ et nous calculons $\boldsymbol{M}^{-1}$:

$$\boldsymbol{M}^{-1} = \frac{1}{\Delta} \left[\begin{array}{cc} S & -S_x \\ -S_x & S_{xx} \end{array} \right] = \left[\begin{array}{cc} \sigma_{a*}^2 & \sigma_{a*b*} \\ \sigma_{a*b*} & \sigma_{b*}^2 \end{array} \right].$$

Vous voyez que l'inverse de la matrice des équations normales est la matrice dite des « variances-covariances ». C'est un cas où le calcul de l'inverse d'une matrice s'avère fructueux.

14.6.2. Linéarisation

Bien des fonctions peuvent, par un changement de variable approprié, se ramener au modèle linéaire du paragraphe précédent. Nous allons encore traiter un cas particulier. L'étude théorique de la cinétique d'une réaction chimique montre que la concentration d'un réactif dépend du temps selon le modèle :

$$y = y_0 e^{-\alpha t}.$$

L'expérience nous a fourni des valeurs y_i relevées aux instants t_i . Pour déterminer y_0 et α, nous « linéarisons » ce modèle en posant $z = \ln y$:

$$z = \ln y_0 - \alpha t \equiv \beta - \alpha t.$$

Dans un calcul de moindres carrés habituel, chaque terme $y_i - f_i$ est pondéré par l'inverse de l'écart-type, σ_i. Quel est l'écart-type sur $\ln y_i$? Il se déduit de la loi de propagation des erreurs :

$$\sigma_z = \frac{\partial z}{\partial y}\sigma_y = \frac{\sigma_y}{y}$$

en supposant que chaque mesure souffre de la même incertitude. L'algorithme des moindres carrés va fournir la meilleure valeur de β, avec son écart-type, σ_β, mais c'est y_0 qui a un sens physique ; nous utiliserons encore la relation (14.22) pour trouver cette fois : $\sigma_{y0} = y_0\sigma_\beta$.

14.7. Qualité de l'ajustement

Après avoir déterminé les meilleures valeurs des paramètres a et b caractérisant un modèle linéaire, nous devons nous demander si l'expérience vérifie ce modèle. On dispose pour cela de divers tests statistiques. Nous allons décrire le plus courant, le test dit du « χ^2 ».

Nous avons expliqué comment la méthode du maximum de vraisemblance conduisait aux valeurs les plus probables de a et b en rendant minimale la quantité

$$\mathcal{S}^2 = \sum_{i=1}^{m} \left(\frac{y_i - f_i}{\sigma_i} \right)^2$$

où y_i est un résultat de mesure et f_i la valeur de y calculée par le modèle pour la même valeur de la variable indépendante x_i. Si le modèle était exact et les fluctuations absentes, $\mathcal{S}^2$ serait nul. Ce ne sera pas le cas en pratique, à cause des erreurs de mesure d'une part, et des défauts du modèle d'autre part. Nous espérons néanmoins que $\mathcal{S}^2$ est « petit ». Comme il s'agit d'une somme sur toutes les mesures (m en tout), $\mathcal{S}^2$ augmente avec m. On peut raisonnablement s'attendre à ce que $y_i - f_i$ soit de l'ordre de σ_i et $\mathcal{S}^2$ soit de l'ordre de m. Pour nous affranchir de l'effet du nombre de mesures, nous pourrions être tentés de considérer la quantité $\mathcal{S}^2/m$. Ce n'est pas tout à fait le bon choix, comme le montre l'exemple de $m = 2$. Nous pouvons toujours faire passer une droite exactement par deux points, mais la perfection de cet ajustement n'est pas très convaincante. Il faut en fait rapporter $\mathcal{S}^2$ au « nombre de degrés de liberté » de

l'expérience (ν), défini comme le nombre de résultats indépendants (m) diminué du nombre de contraintes ou du nombre de paramètres ajustables (k en général, deux pour l'exemple d'une fonction affine), $\nu = m - k$. Une meilleure mesure de la qualité de l'ajustement est donc fournie par le « chi-deux réduit »

$$\chi^2_\nu \equiv \frac{\mathcal{S}^2}{\nu} = \frac{1}{m-k} \sum_{i=1}^{m} \left(\frac{y_i - f_i}{\sigma_i} \right)^2 .$$

D'après les considérations précédentes, χ^2_ν est une fonction aléatoire dont la valeur moyenne est proche de 1. Aussi, une valeur de χ^2_ν voisine de 1 signale-t-elle un bon modèle. De façon un peu plus précise, nous voyons que la quantité $\mathcal{S}^2$ est une somme de carrés de variables aléatoires gaussiennes ; elle obéit donc à la loi du χ^2. Nous allons devoir décider si, et avec quelle probabilité, la valeur de χ^2_ν déduite de nos mesures (et proportionnelle à $\mathcal{S}^2$) peut être le fait du hasard. Nous faisons donc l'hypothèse que notre modèle est juste, nous choisissons un seuil α et, munis d'une table de la fonction de répartition de la loi de Pearson, nous cherchons un nombre u tel que pour une variable T' (distribuée selon la loi du chi-carré), Proba$(T' > u) = \alpha$. Si $\chi^2/\nu > u$, nous concluons que l'écart théorie-expérience est très improbable, ou que notre hypothèse est fausse.

Il est malheureusement assez fréquent que nous ne connaissions pas individuellement l'incertitude affectant les y_i : c'est le cas lorsque n'avons pas les moyens ou le temps de répéter les expériences nécessaires. Nous sommes alors contraints de faire le raisonnement simplifié suivant. Nous supposons que chaque mesure a le même écart-type, $\sigma_i = \sigma$ pour l'instant inconnu. Nous déterminerons les meilleures valeurs de a et b, puis nous recalculerons σ comme

$$\sigma^2 = \frac{1}{m-2} \sum_{i=1}^{m} (y_i - a^* x_i - b^*)^2.$$

Les écarts-types sur a^* et b^* peuvent alors être calculés par les formules (14.28) multipliées par le facteur $\sqrt{\mathcal{S}/(m-2)}$.

Exemple – Nous avons relevé les couples de valeurs suivantes :

x	1	2	4	5	8	9	10	12	15
y	$-0,1$	1,7	3,3	4,9	7,6	8,4	9,1	10,7	12,3

et nous voulons lisser les valeurs de y à l'aide du modèle $y = ax + b$, autrement dit, ajuster au sens des moindres carrés les paramètres a et b pour que l'équation proposée représente au mieux les résultats expérimentaux. Vous pouvez vous prêter au jeu suivant : reporter ces données sur papier millimétré et déterminer « à l'oeil » la meilleure droite et les valeurs correspondantes de a et b, avant de faire les calculs.

- 1er cas (fréquent) : nous ignorons les valeurs des erreurs commises sur les y, mais nous avons de bonnes raisons de penser qu'elles sont toutes aléatoires et distribuées de façon identique. Nous posons alors $\sigma_i = \sigma = 1$. Comme toutes les formules qui donnent a^* et b^* sont homogènes en σ, la valeur exacte importe peu. Le calcul se fait facilement avec une calculette ou avec Scilab. Dans ce logiciel, nous définissons deux vecteurs `x` et `y` contenant les données. Nous exécutons ensuite les instructions

```
S = length(x); Sx = sum(x); Sxx = sum(x.*x);
Sy = sum(y); Sxy = sum(x.*y);
Delta = S*Sxx-Sx*Sx;
a = (S*Sxy - Sx*Sy)/Delta, b = (Sxx*Sy - Sx*Sxy)/Delta;
siga = sqrt(S/Delta); sigb = sqrt(Sxx/Delta)
```

Les résultats sont $a^* = 0,8903$, $\sigma_{a*} = 0,0754$, $b^* = -0,0958$, $\sigma_{b*} = 0,645$; les points expérimentaux et la droite sont représentés sur la figure 14.8. Si la pente est bien déterminée, l'ordonnée à l'origine est assez imprécise.

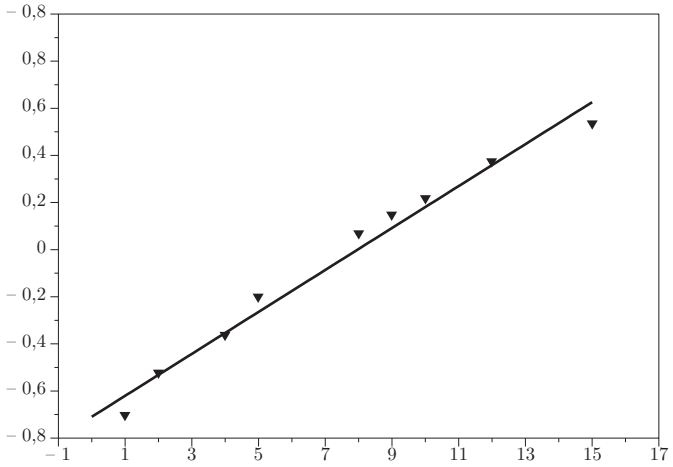


Figure 14.8 – Points expérimentaux et droite des moindres carrés.

Avec ces valeurs de a^* et b^*, nous recalculons σ : nous admettons que tout écart entre valeur expérimentale et valeur calculée est dû à une erreur aléatoire ; nous retenons la moyenne du carré de ces écarts comme valeur de σ^2 (comme indiqué plus haut, nous considérons qu'il n'y a que $n - 2 = 7$ mesures). Nous trouvons que $\sigma = 0,62$, d'où nous déduisons les « bonnes » valeurs $\sigma_a = 0,047$, $\sigma_b = 0,401$. Les amateurs d'Excel verront que ce logiciel, grâce à la fonction « DROITEREG », peut faire automatiquement les calculs précédents.

- 2ème cas (idéal) : Nous savons que les valeurs de y sont entachées d'erreurs telles que l'écart-type sur les 4 premières valeurs de y est de 0,4 et qu'il est de 0,6 sur les 5 dernières. Nous obtenons alors :

$$S = 38,89 \quad S_x = 225,0 \quad S_y = 194,86 \quad S_{xx} = 1993,06 \quad S_{xy} = 1757,08$$

d'où nous tirons $a = 0,911, \sigma_a = 0,038, b = -0,259, \sigma_b = 0,272$. La quantité $\mathcal{S}^2$ vaut 11,04 soit un χ^2_ν réduit (7 degrés de liberté) de 1,577. Les tables donnent une probabilité d'environ 0,14 d'obtenir par hasard une valeur au moins aussi élevée. Le modèle est donc accepté.

14.8. Coefficient de corrélation

Une suite d'observations nous a fourni des couples de valeurs (x_i, y_i), comme par exemple le nombre de taches solaires observées pendant l'année i et le cours moyen du bourgogne à la vente des Hospices de Beaune la même année. Nous nous posons la question de savoir s'il existe une relation causale entre ces deux variables, en d'autres termes, nous cherchons à savoir si x est corrélé à y. Nous allons examiner le cas le plus simple, celui d'une corrélation linéaire. Si y dépend de x selon la loi linéaire $y = ax+b$, les résultats du paragraphe précédent nous permettent d'estimer le coefficient a. Si x et y sont indépendants, y ne doit, en moyenne, ni croître ni décroître quand x augmente, donc $a = 0$. Il est aussi permis d'estimer les paramètres de la loi $x = a'y + b'$. Ils sont différents des précédents, mais a et a' sont liés si x et y sont corrélées. Si la corrélation entre x et y est parfaite, les deux lois sont décrites par des fonctions inverses l'une de l'autre et $a = 1/a'$ ou $aa' = 1$. Nous définissons le coefficient empirique de corrélation par la relation $r = \sqrt{aa'}$. En supposant tous les écarts types égaux (une hypothèse qu'il est facile de lever), on trouve, pour m observations, la formule ci-dessous.

$$r \equiv \frac{\sum_1^m (x_i - \langle x\rangle)(y_i - \langle y\rangle)}{\sqrt{\sum_1^m (x_i - \langle x\rangle)^2 \sum_1^m (y_i - \langle y\rangle)^2}}. \tag{14.30}$$

$|r|$ est compris entre 0 (pas de corrélation) et 1 (corrélation complète) : on considère que x et y sont fortement corrélés lorsque que $|r|$ est voisin de 1, pratiquement indépendants si r est proche de 0, mais il est difficile de donner un critère vraiment quantitatif en l'absence d'hypothèses plus précises sur la nature des erreurs ou des fluctuations entachant x et y.

14.9. Ajustement sur une fonction linéaire de plusieurs paramètres

Nous généralisons hardiment le raisonnement des paragraphes précédents. Nous disposons de m observations (x_i, y_i), $1 \leq i \leq m$. Les valeurs de x sont précises, alors que celles de y sont entachées d'erreurs aléatoires. Nous avons des raisons de penser que la quantité y dépend de plusieurs paramètres. Nous considérons alors le modèle

$$Y = \sum_{k=1}^{n} a_k \varphi_k(x) + \varepsilon \equiv f(x, \boldsymbol{a}) + \varepsilon \tag{14.31}$$

qui dépend **linéairement** des n paramètres a_k, alors que les fonctions φ_k qui définissent la loi en x peuvent être quelconques. Les paramètres a_k sont considérés comme les coordonnées d'un vecteur $\boldsymbol{a}$. Posons encore

$$f_i = f(x_i, \boldsymbol{a}) = \sum_{k=1}^{n} a_k \varphi_k(x_i).$$

Nous souhaitons de nouveau rendre minimale la somme des carrés des écarts :

$$\mathcal{S}^2 = \sum_{i=1}^{m} \frac{1}{\sigma_i^2}(y_i - f_i)^2.$$

Les conditions pour que $\mathcal{S}^2$ soit minimale s'écrivent :

$$0 = \frac{\partial \mathcal{S}}{\partial a_\ell} = -2\sum_{i=1}^{m} \frac{1}{\sigma_i^2}(y_i - f_i)\frac{\partial f_i}{\partial a_\ell}, \quad 1 \leq \ell \leq n. \tag{14.32}$$

Or $\partial f_i / \partial a_\ell = \varphi_\ell(x_i)$, si bien que la relation 14.32 devient

$$\sum_{i=1}^{m} \frac{1}{\sigma_i^2} f_i \varphi_\ell(x_i) = \sum_{i=1}^{m} \frac{1}{\sigma_i^2} y_i \varphi_\ell(x_i).$$

Remplaçons, au premier membre, f_i par son expression $\sum_{k=1}^{n} a_k \varphi_k(x_i)$; nous trouvons

$$\sum_{k=1}^{n} a_k \left[\sum_{i=1}^{m} \frac{1}{\sigma_i^2} \varphi_\ell(x_i)\varphi_k(x_i) \right] = \sum_{i=1}^{m} \frac{1}{\sigma_i^2} y_i \varphi_\ell(x_i),$$

après avoir inversé l'ordre des sommes du premier membre. Posons maintenant

$$\left[\sum_{i=1}^{m} \frac{1}{\sigma_i^2} \varphi_\ell(x_i)\varphi_k(x_i) \right] \equiv M_{\ell,k}$$

et

$$\sum_{i=1}^{m} \frac{1}{\sigma_i^2} y_i \varphi_\ell(x_i) \equiv b_\ell.$$

Vous constatez que les conditions 14.32 peuvent s'écrire, avec ces notations,

$$\sum_{k=1}^{n} M_{\ell,k} a_k = b_\ell, \quad 1 \leq \ell \leq n, \tag{14.33}$$

ou encore

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{b}.$$

Ce sont les équations normales (ou de Gauss) du problème d'ajustement. La matrice $\boldsymbol{M}$ est symétrique définie positive et le système linéaire se résout facilement par la méthode de Cholesky. La solution formelle est

$$\boldsymbol{a} = \boldsymbol{M}^{-1}\boldsymbol{b}.$$

Comme dans le cas à 2 inconnues, l'inverse de $\boldsymbol{M}$ présente un intérêt : l'élément diagonal $[\boldsymbol{M}^{-1}]_{ii}$ est la variance du paramètre a_i, alors que l'élément extra-diagonal $[\boldsymbol{M}^{-1}]_{ik}$ est la covariance entre a_i et a_k.

Les équations normales peuvent être obtenues en suivant un autre raisonnement. Supposons un instant que toutes les mesures soient également précises et donc que $\sigma_i = \sigma$. L'ajustement sur le modèle (14.31) se traduit par le système linéaire

$$\sum_{k=1}^{n} \varphi_k(x_i) a_k = y_i, \quad 1 \leq i \leq m. \tag{14.34}$$

Définissons une matrice rectangulaire $\boldsymbol{A}$ par ses éléments

$$A_{ik} = \varphi_k(x_i), \quad 1 \leq i \leq m, 1 \leq k \leq n$$

et un vecteur $\boldsymbol{y}$ de composantes y_i, $1 \leq i \leq m$. Avec ces notations, le système surdéterminé (14.34) s'écrit

$$\boldsymbol{Aa} = \boldsymbol{y}.$$

Au § 6.7, nous avons expliqué comment en former une solution, au sens des moindres carrés. On est amené à résoudre

$$\boldsymbol{A}^T \boldsymbol{Aa} = \boldsymbol{A}^T \boldsymbol{y},$$

un système équivalent à (14.33), après disparition du facteur commun $1/\sigma^2$. La matrice $\boldsymbol{M} = \boldsymbol{A}^T \boldsymbol{A}$ est évidemment symétrique ; elle est aussi définie positive comme le montre la forme quadratique $\boldsymbol{b}^T \boldsymbol{A}^T \boldsymbol{Ab} = \| \boldsymbol{Ab} \|^2$.

Lorsque les mesures sont de précisions inégales, il faut pondérer différemment les équations (14.34) ; cela revient à poser

$$A_{ik} = \frac{1}{\sigma_i} \varphi_k(x_i)$$

et à remplacer y_i par y_i/σ_i.

14.10. Pour en savoir plus

– D. Taupin : *Probabilities, data reduction and error analysis in the physical sciences* (Éditions de Physique, Paris, 1988).

– P. Jaffard : *Initiation aux méthodes de la statistique et du calcul des probabilités* (Masson, Paris, 1990).

– P.R. Bevington, D.K. Robinson : *Data reduction and error analysis for the physical sciences* (McGraw-Hill, New York 1992).

– K. Protassov : *Probabilités et incertitudes* (Grenoble Sciences, 2000).

– Cours de J. Collot : *Erreur, Probabilité et Statistique*, `http://lpsc.in2p3.fr/collot`

– sur le site `http://www.librecours.org/` :
  – J. Harthong : *Calcul des probabilités*
  – C. Aslangul : *Éléments de théorie des probabilités.*

– Cours de T. Dudok de Witt : *Outils statistiques et numériques pour la mesure et la simulation*, `http://lpce.cnrs-orleans.fr`, voir la page personnelle de T. Dudok de Witt, section enseignement.

14.11. EXERCICES

Exercice 1

K est une variable distribuée selon la loi binomiale.

a) Vérifier que les probabilités $b(k;n,p)$ données par (14.11) sont normalisées, c'est-à-dire que

$$\sum_{k=1}^{k=n} b(k;n,p) = 1.$$

b) Calculer $\mu = \langle K \rangle$ et σ_K, pour retrouver les formules (14.12). Pour établir l'expression de l'écart-type, il est commode de calculer la valeur de $\langle K(K-1) \rangle$.

c) On définit la fonction génératrice $g(t) = \langle t^K \rangle$. Calculer $g(t)$ en fonction de p, q et n, puis $g'(t)$ et $g''(t)$. Utiliser ces expressions pour retrouver les valeurs de μ et de σ.

Exercice 2

K est maintenant une variable aléatoire répartie selon la loi de Poisson.

a) Vérifier que les probabilités définies par la formule (14.13) sont convenablement normalisées.

b) En employant les mêmes méthodes que pour l'exercice précédent, calculer la moyenne et la variance de K, pour retrouver les formules (14.14).

Exercice 3

Une variable aléatoire prend certainement ses valeurs dans l'intervalle $[a,b]$, avec une probabilité uniforme.

a) Calculer la valeur moyenne et l'écart-type de cette variable, comme donnés par les formules (14.15).

b) Application : une longueur X est mesurée au millimètre près ; quel est l'écart-type de X dû aux erreurs d'arrondi ?

Exercice 4

Une variable aléatoire X obéit à la loi normale de moyenne nulle et d'écart-type unité (loi normale standard). Quelle est la densité de probabilité pour la variable $Z = aX + b$?

Exercice 5

L'observation de 12 familles comportant quatre enfants a fait apparaître les nombres de filles suivants : 3, 2, 2, 0, 1, 3, 1, 1, 1, 2, 3, 1. En utilisant la méthode du maximum de vraisemblance, calculer la meilleure estimation de la probabilité d'avoir une fille, en supposant qu'il n'y a aucune corrélation entre les sexes des enfants d'une même famille.

Exercice 6

Une promotion d'étudiants a mesuré la chaleur molaire de neutralisation de la soude par l'acide chlorhydrique, à 0,1 M et 20°C ; les résultats (valeurs absolues en kJ mol^{-1}) sont consignés ci-après.

56,9 59,2 56,3 58,0 56,9 53,8 55,4 58,0 59,6 55,5
58,4 55,0 55,7 56,6 57,2 58,0 56,4 57,6 57,5 55,0
57,7 58,5 58,9 57,8 57,4 54,8 56,4 55,2 60,3 57,1
57,1 60,4 58,9 55,5 54,7 58,6 57,8 58,0 55,5 55,6
58,9 59,8 60,0 57,1 56,4 59,5 57,7 60,0 57,6 56,8
57,2 58,2 57,4 55,7 59,1 55,4 56,1 57,7 56,9 59,2
55,1 56,8 55,7 61,6 58,3

a) Calculer la moyenne $\langle \Delta H \rangle$ et l'écart-type s de cet échantillon.

b) Les tables donnent la valeur ΔH = -56,40 kJ/mol. Les résultats précédents sont-ils compatibles avec cette donnée ? On sait que la moyenne $\langle X \rangle$ d'un grand nombre de variables X_i, réparties identiquement, est distribuée selon une loi normale, quelle que soit la densité de probabilité $p(x)$ des termes de la somme. Plus précisément, la quantité

$$T = \frac{\langle X \rangle - \mu}{s/\sqrt{n}}$$

est répartie à peu près selon la loi normale standard. On sait aussi que 95% de l'aire sous la courbe de Gauss standard est comprise entre les abscisses -2 et 2. Les résultats des étudiants sont-ils significativement différents de ceux des livres ?

c) On cherche maintenant à vérifier si les résultats sont bien répartis selon une loi normale. Pour avoir un nombre de valeurs significatif dans chaque intervalle, on regroupe les données selon les classes suivantes : $[-\infty \ldots 55,05]$, $[55,05 \ldots 56,05]$, $[56,05 \ldots 57,05]$, $[57,05 \ldots 58,05]$, $[58,05 \ldots 59,05]$ et $[59,05 \ldots \infty]$. On forme l'estimateur

$$T = \sum \frac{(f_i^{obs} - f_i^{calc})^2}{f_i^{calc}},$$

les f_i étant les fréquences, observées ou calculées. T obéit à la loi du χ^2 à $n-3$ degrés de libertés. Les chaleurs de neutralisation sont-elles gaussiennes ?

Exercice 7

On mesure l'angle d'incidence θ_1 et l'angle de réfraction θ_2 d'un rayon à la surface de séparation de deux milieux d'indices n_1 et n_2 et on trouve (en degrés)

$$\theta_1 = 22,03 \; ; \quad \theta_2 = 14,45.$$

D'autre part, on estime que l'écart-type sur de telles mesures d'angle vaut 0,2 degré. Déterminer n_2 et son incertitude, sachant que n_1 vaut exactement 1,000.

Exercice 8

Une étudiante a procédé à des mesures répétées de la tension d'une pile étalon ; pour 40 mesures, elle trouve $\langle v_1 \rangle$ = 1,022 V avec un écart-type de 0,01 V. Sur les conseils

d'un enseignant, elle améliore son montage, ce qui réduit l'incertitude d'un facteur 2,5. Elle refait une série de 10 mesures, pour trouver $v_2 = 1{,}018$ V. Quelle valeur de la tension faut-il retenir ?

Exercice 9

On a déterminé 3 fois, indépendamment, les angles (x, y, z) entre faces d'un prisme triangulaire. Toutes les mesures d'angles ont la même précision ($\pm 0,5$ minute d'arc). Les résultats sont les suivants :

x	y	z
$89^\circ 55'$	$45^\circ 5'$	$44^\circ 57'$
$89^\circ 59'$	$45^\circ 6'$	$44^\circ 55'$
$89^\circ 57'$	$45^\circ 5'$	$44^\circ 58'$

On peut choisir comme valeurs définitives les moyennes arithmétiques de ces résultats. On s'aperçoit alors que la condition $x + y + z = 180^c irc$ n'est pas exactement respectée. Déterminer, par la méthode des moindres carrés, les meilleures valeurs des trois angles, compatibles avec les données expérimentales et respectant la contrainte.

Exercice 10

On a mesuré la température en différentes positions le long d'une barre métallique dont les extrémités sont maintenues en contact avec des thermostats réglés sur les températures de 0 et 100 ° C. Les résultats sont les suivants :

x (cm) :	1	2	3	4	5	6	7	8	9
t (° C) :	15,6	17,5	36,6	43,8	58,2	61,6	64,2	70,4	98,8

On pense que la température est fonction linéaire de l'abscisse. Déterminer les coefficients de cette relation linéaire (1) graphiquement et (2) par la méthode des moindres carrés. Quelle est la précision sur les résultats ? On pourra envisager successivement les deux cas : (a) on sait, par l'expérience acquise, que la précision sur une température est de 0,1° C et (b) on ignore la précision des mesures.

Exercice 11

On a déterminé expérimentalement la résistance électrique R d'un échantillon semiconducteur en fonction de la température absolue T. les résultats sont consignés dans le tableau ci-dessous.

T	10	20	50	100	150	200
$R(\Omega)$	0,63	1,05	1,38	2,22	2,70	2,79

L'auteur de ces mesures estime l'écart-type sur chaque valeur de R à $\sigma_R = 0,5\Omega$. Il a d'autre part des raisons de penser que ses résultats pourraient être décrits par le modèle théorique

$$R = aT^b.$$

a) Montrer qu'un changement de variable simple permet de mettre le modèle sous la forme

$$y = \alpha + \beta x.$$

Calculer la variance de chaque valeur de y.

b) Utiliser la méthode des moindres carrés pour déterminer les meilleures valeurs de α et de β. En déduire des estimations de a et de b.

c) Calculer l'écart-type sur les paramètres α et β, puis sur les valeurs finales a et b.

Exercice 12

Les atomes d'un gaz ont été portés dans leur état excité par absorption de lumière. L'éclairage étant interrompu soudainement en $t = 0$, les atomes retournent dans leur état fondamental en émettant de la lumière (fluorescence). On cherche à déterminer la cinétique du phénomène. Pour cela, on enregistre l'énergie émise (ou le nombre de photons émis) dans chaque intervalle de 1 ns suivant la fin de l'excitation, avec les résultats suivants :

t_n	1	2	3	4	5	6	7	8
I_n	415	232	162	91	62	20	35	9

Ainsi, 415 photons ont été émis entre 0 et 1 nanoseconde. On sait que les valeurs mesurées sont entachées d'une erreur aléatoire gaussienne dont on estime l'écart-type à 30 unités. La théorie laisse prévoir que la lumière émise décroît exponentiellement :

$$I(t) = I_0 e^{-at}$$

Pour déterminer, par la méthode des moindres carrés, les valeurs des paramètres I_0 et a qui représentent le mieux les résultats, on linéarise le modèle en posant :

$$y(t) \equiv \ln I(t) = \alpha t + \beta$$

et on va ajuster α et β pour reproduire au mieux les valeurs « expérimentales » $y_n = \ln I_n$.

a) Calculer les y_n avec leurs écarts-types.

b) Trouver les meilleures valeurs de α et β, au sens des moindres carrés.

c) Déterminer la valeur de a et celle de la durée de vie $\tau = 1/a$, ainsi que leurs écarts-types.

This page intentionally left blank

Chapitre 15

Méthodes de Monte Carlo

Pourquoi choisir comme titre de chapitre le nom d'une station balnéaire et d'un haut-lieu des jeux de hasard ? Un semblant d'explication vous sera donné plus loin dans ce chapitre. Pour l'instant, contentons-nous d'affirmer que les « méthodes de Monte Carlo » jouent un rôle important dans de nombreux domaines, de la physique numérique aux mathématiques financières. Dans la suite, nous qualifions « d'aléatoire » (on d'indéterministe ou parfois de « stochastique ») toute grandeur physique, toute variable, tout phénomène régi par le hasard. Si, au contraire, le hasard n'intervient pas, nous employons les termes de certain ou de « déterministe ».

Comme nous l'avons déjà dit au début du chapitre 14, le hasard peut intervenir directement de plusieurs façons dans un travail scientifique. Il peut arriver que le phénomène étudié soit par nature aléatoire (mouvement brownien par exemple) ; en d'autres occasions, un phénomène déterministe sera « parasité » du bruit ou des erreurs de mesures elles-mêmes aléatoires. Il peut être instructif ou utile de simuler sur ordinateur un système physique de ce genre. Dans ce cas, on parle de méthode de Monte Carlo indéterministe.

À partir de 1945, des physiciens théoriciens, suivis par des spécialistes de bien d'autres disciplines, se sont aperçus que le hasard pouvait leur servir à calculer des quantités certaines, comme une intégrale ou la solution d'une équation aux dérivées partielles. On range ce type d'activité dans la catégorie des méthodes de Monte Carlo déterministes.

Vous pouvez à bon droit vous demander comment on peut obtenir d'un ordinateur un comportement aléatoire. Tout ce que nous savons des ordinateurs et des algorithmes nous conduit à prévoir un comportement déterministe et certain, même si nous sommes parfois incapables de prévoir le résultat fourni par la machine. C'est pourquoi nous allons consacrer le premier paragraphe à la fabrication (« génération » dans le jargon de l'analyse numérique) de nombres aléatoires ou prétendus tels.

15.1. Générateurs de nombres aléatoires

La résolution du paradoxe énoncé ci-dessus est simple : nous allons calculer, par itération, une suite de nombres périodique mais caractérisée par une très longue période. Localement (à une échelle inférieure à la période) ces nombres auront l'apparence de nombres tout à fait aléatoires. Nous nous limitons, dans cette section, à des nombres aléatoires répartis de façon uniforme sur un intervalle fini. Le programme correspondant s'appellera un « générateur de nombres aléatoires » ou « GNA » (Random Number Generator, RNG, en anglais).

15.1.1. Principe

Le premier algorithme proposé l'a été par von Neumann vers 1946 ; il est connu sous le nom d'algorithme du « milieu du carré ». Partant d'un nombre entier (disons à six chiffres) considéré comme le premier nombre de la suite, nous l'élevons au carré ; nous supprimons les trois chiffres de tête et les trois chiffres de queue pour conserver les six chiffres du milieu, qui forment le deuxième nombre pseudo-aléatoire de la suite. Ce procédé est itéré autant que nécessaire. Voici un petit exemple :

$$\begin{array}{rl} 738497 & u_0 = 738497 \\ 545\mathbf{377819}009 & u_1 = 377819 \\ 142\mathbf{747196}761 & u_2 = 747196 \end{array}$$

Simple à mettre en oeuvre et d'exécution rapide, ce générateur est en fait très mauvais d'un point de vue probabiliste, comme l'ont montré les analyses ultérieures.

La plupart des générateurs actuels utilisent une itération comportant une « congruence linéaire »

$$\begin{array}{lll} u_{n+1} = au_n & \mod m & \text{ou} \\ u_{n+1} = au_n + b & \mod m. & \end{array}$$

Rappelons que cette notation signifie que u_{n+1} est le reste de la division de au_n (ou de $au_n + b$) par m, comme $2 = 23 \mod 7$. La plupart des langages de programmation comportent l'opérateur `mod` ou un équivalent. En Scilab, on écrit `modulo(23,7)` pour obtenir 2. Vous voyez que u_{n+1} est certainement au plus égal à $m-1$ et que la période de ces itérations est au plus égale à m. Un choix convenable du multiplicateur a et de l'incrément b conduira à une période proche de cette borne supérieure. Par exemple, le choix $a = 57, c = 1, m = 256$ produit une séquence de période 256 (mais de qualité statistique déplorable). Au contraire, les valeurs $a = 16807, c = 0, m = 2^{31} - 1 = 2147483647$ produit une suite de nombres « très » aléatoires et de période $m - 1$.

La construction d'un « bon » générateur de nombres aléatoires est affaire de spécialistes et il est extrêmement facile de rédiger de mauvais algorithmes. On sait maintenant que la fonction `randu` fournie avec les compilateurs IBM des années 60 relève de cette dernière catégorie. La manipulation de grands entiers (comme a et m) et le calcul exact de leur produit est une des difficultés pratiques qu'il faut résoudre.

Scilab propose deux fonctions, `rand()` et `grand()`. La première renvoie un nombre fractionnaire dont la loi de probabilité est soit uniforme sur [0,1], soit normale réduite

(moyenne nulle, écart-type égal à un). La seconde offre plus de possibilités : elle engendre des entiers ou des réels, selon une loi de probabilité choisie par l'utilisateur. Les compilateurs (FORTRAN, C ou autres) comportent également des fonctions rendant les mêmes services. Comme toute itération, les formules précédentes utilisent une valeur initiale, la « graine » (« seed » en anglais). Les fonctions que nous venons d'évoquer sont toutes accompagnées d'une variante qui permet à l'utilisateur de choisir la graine. Pendant la mise au point d'un programme, on conserve la même graine, et donc la même séquence pseudo-aléatoire. Lorsque le programme est utilisé pour un long calcul, il est raisonnable de changer la graine de temps en temps.

15.1.2. Vérification d'un GNA

Tous les générateurs, ou presque, sont bons lorsqu'il s'agit de fournir quelques dizaines de nombres aléatoires ; il n'en va pas de même pour des calculs « réels » qui utilisent des centaines de millions de valeurs aléatoires. Il faut que ces nombres ne soient pas corrélés. Il est recommandé de vérifier avant usage le comportement d'un GNA inconnu.

Une première approche consiste a construire l'histogramme des valeurs réputées aléatoires pour voir s'il est compatible avec la loi de probabilité annoncée. Si un GNA crée des entiers répartis uniformément sur [1,10], nous pouvons dénombrer les apparitions des chiffres 1, 2,...,9 et ces valeurs doivent être approximativement égales. Plus précisément, nous pouvons appliquer le test du χ^2 pour vérifier la conformité à la loi uniforme.

Une méthode graphique simple consiste à reporter dans le plan (x, y), les points de coordonnées $x_i = r_i, y_i = r_{i+1}$. Un « bon » GNA produira un nuage de points sans structure évidente, alors qu'un mauvais GNA conduira à des points organisés en bandes. Les figures 15.1 montrent ces deux possibilités. Nous avons utilisé d'une part le générateur incorporé dans Scilab, d'autre part le générateur déjà mentionné plus haut ($a = 57, c = 1, m = 256$).

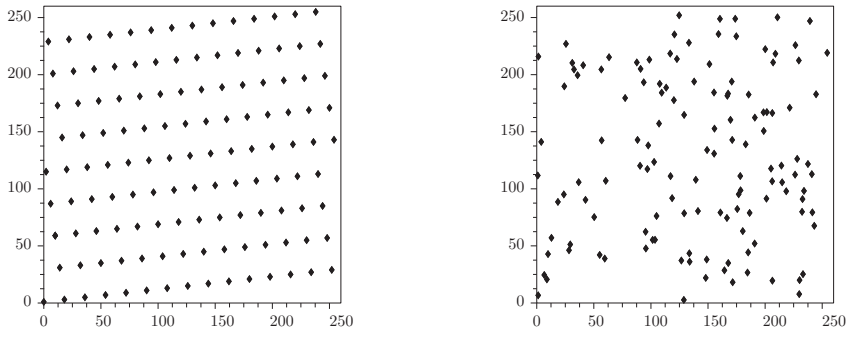


Figure 15.1 – Nombres aléatoires produits par un mauvais générateur (à gauche) et par un bon générateur (à droite).

Vous trouverez en fin de chapitre des références à des tests modernes et performants.

15.1.3. Validation d'un GNA à l'aide d'une marche aléatoire

Un mobile effectue des déplacements aléatoires dans le plan, occupant une suite de positions $(0,0),(x_1,y_1),(x_2,y_2),\ldots,(x_k,y_k),\ldots,(x_N,y_N)$. Chaque déplacement (ou « pas ») est indépendant du déplacement précédent (non corrélé). Ce type de mouvement est souvent utilisé comme modèle mathématique du phénomène de diffusion ou encore pour simuler la conformation dans l'espace d'une molécule de polymère. On peut très bien envisager des marches aléatoires à une, deux, trois ou n dimensions. D'autre part, on dispose de nombreuses options pour la réalisation de chaque pas. On peut tirer au hasard deux nombres $\delta x_k, \delta y_k$ et poser $x_{k+1} = x_k + \delta x_k, y_{k+1} = y_k + \delta y_k$, on peut engendrer un nombre aléatoire $\alpha \in [0, 2\pi]$ et poser $\delta x = \ell \cos\alpha, \delta y = \ell \sin\alpha$. Le mobile fictif peut encore se déplacer sur un réseau. Dans le cas simple du réseau carré plan, le point $k+1$ se déduit du point k par un « saut » de longueur ℓ vers la droite, la gauche, le haut ou le bas, toujours au hasard. La figure 15.2 montre quelques réalisations d'une telle marche.

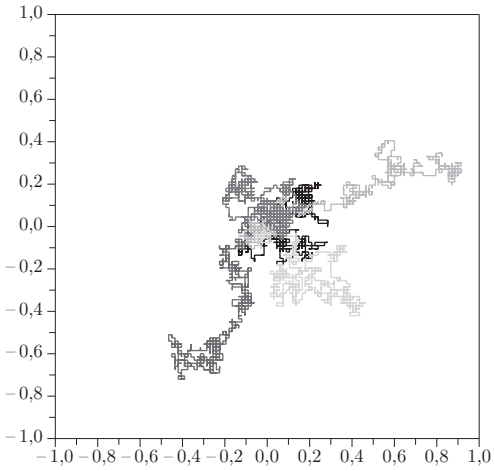


Figure 15.2 – Marches aléatoires sur un réseau plan carré.
Chaque trajectoire comporte 1000 pas de longueur 0,015.

Si nous abordons la marche aléatoire à ce moment du développement, c'est que ce processus permet aussi de vérifier commodément la qualité du GNA qui est à l'origine des déplacements. Si vous affichez à l'écran un certain nombre de marches aléatoires, vous devrez observer qu'aucune région n'est particulièrement « favorisée » ; les extrémités (x_N, y_N) des trajectoires doivent être réparties dans toutes les directions. Dans le cas contraire, on peut soupçonner un défaut du GNA.

Plus précisément, on peut montrer que la variable aléatoire $r_N = \sqrt{x_N^2 + y_N^2}$ (la distance à l'origine en fin de parcours) vérifie

$$\langle r_N \rangle = \sqrt{N}\rho,$$

où ρ est la « valeur quadratique moyenne » des déplacements individuels $(\delta x_k, \delta y_k)$

$$\rho^2 \equiv \frac{1}{N}[\delta x_1^2 + \delta y_1^2 + \delta x_2^2 + \delta y_2^2 + \cdots + \delta x_N^2 + \delta y_N^2].$$

Si tous les pas ont même longueur ℓ, on a aussi $\rho = \ell$ et

$$\langle r_N \rangle = \ell\sqrt{N}. \tag{15.1}$$

Attention, cette relation ne s'applique qu'en moyenne. Pour vérifier la qualité d'un GNA, il faut disposer d'un certains nombre de réalisations de r_N, en faire la moyenne et refaire tout le calcul pour diverses valeurs de N. On sait aussi que l'écart-type de r_N varie comme $N^{1/4}$. Voici le squelette d'un programme pour réaliser ces opérations.

Listing 15.1 – Analyse de marches aléatoires

```
//boucle sur le nombre de pas
for np = NPMIN:NPMAX
   NR = sqrt(np);
   //boucle sur les répétitions
   for ir= 1:NR
      x = 0; y = 0;
      //une marche aléatoire
      for i = 1:np
        dx = rand()-0.5; dy = rand()-0.5;
        x = x+dx; y = y+dy;
      end
      d(ir) = sqrt(x*x+y*y);
   end
//calcul de la valeur quadratique moyenne et de l'écart-type
//du déplacement total
   dm(np) = sum(d)/NR;
   ddm = sum(d.*d)/NR;
   sigma_d(np) = sqrt(ddm - dm(np)*dm(np));
end
```

Pour vérifier « à l'oeil » la relation (15.1), il est commode de tracer $\langle r_N \rangle$ en fonction de $\sqrt{N}$; c'est ce que nous avons fait, comme le montre la figure 15.3, avec la fonction `rand()` de Scilab. On n'observe pas d'écart significatif avec la théorie.

15.2. Nombres aléatoires à répartition non-uniforme

Il existe de nombreux phénomènes physiques stochastiques associés à des densités de probabilité non constantes. Le cas le plus courant est la répartition gaussienne. On a inventé de nombreux algorithmes pour produire des suites de nombres aléatoires répondant à une loi de probabilité donnée. Nous allons en décrire quelques uns.

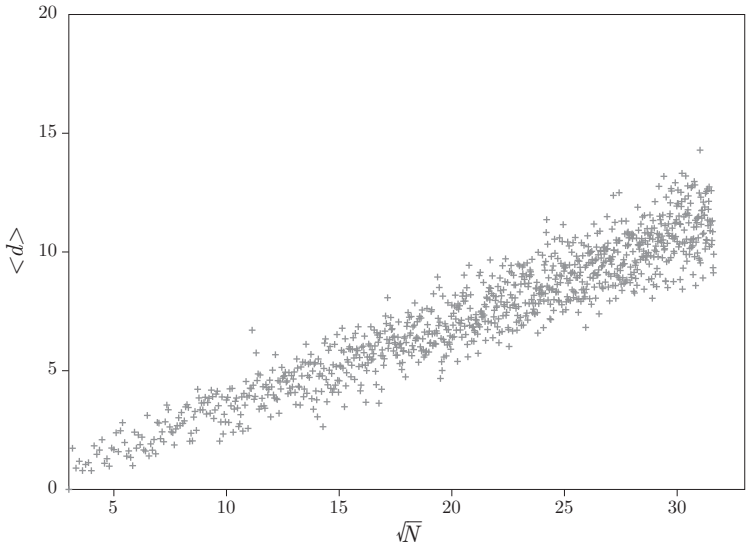


Figure 15.3 – Le déplacement moyen en fonction de la racine carrée du nombre de pas.

15.2.1. Fonction d'une variable aléatoire

Posons-nous le problème suivant : soit un nombre aléatoire X de répartition connue (uniforme, gaussienne, etc.) ; quelle est la loi de répartition de la quantité $g(X)$? L'électronique nous fournit de nombreux exemples de problèmes de ce genre ; en voici un. Une tension de bruit aléatoire et gaussienne $e(t)$ est appliquée aux bornes d'une résistance R. Quelles sont les caractéristiques de la puissance dépensée, e^2/R ?

Plus précisément, nous supposons que la fonction g est continue, que son domaine de définition inclut le domaine de variation de X et que l'équation $g(x) = y$ n'a qu'une solution dans le domaine considéré (cette dernière restriction étant facile à lever). Ces conditions sont remplies par la fonction représentée par la figure 15.4.

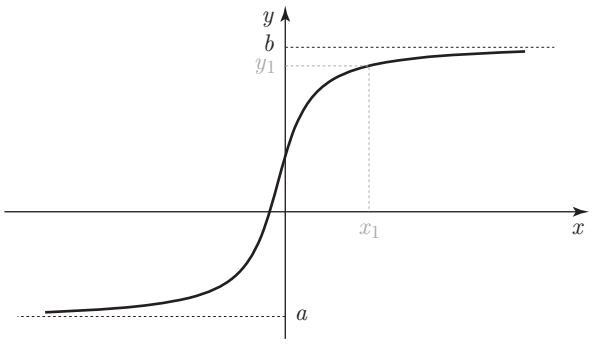


Figure 15.4 – Fonction d'une variable aléatoire : cas de la fonction de répartition.

Nous notons $P(x)$ la fonction de répartition de X et $Q(y)$ celle de $Y = g(X)$. Comme vous pouvez le voir, si $y < a$, il n'existe pas de valeur de x telle $g(x) \leq y$ et donc

Proba$(Y \leq y) = 0$. Si $y \geq b$, $g(x) < y$ quel que soit x, si bien que Proba$(Y < y) = 1$. En résumé

$$Q(y) = \begin{cases} 0 & y < a, \\ 1 & y \geq b. \end{cases}$$

Lorsque x_1 et $y_1 = g(x_1)$ sont disposés comme sur la figure 15.4, $g(x) < y_1$ si $x < x_1$, puisque nous avons choisi une fonction croissante ; par conséquent

$$Q(y_1) = \text{Proba}(x \leq x_1) = P(x_1). \tag{15.2}$$

Il est souvent plus utile de savoir déduire la densité de probabilité de Y, soit $q(y)$, à partir de celle de X, soit $p(x)$. Comme le montre la figure 15.5, l'intervalle $[x_1, x_1 + \delta x]$ correspond, par l'application bijective $x \mapsto g(x)$, à l'intervalle $[y_1, y_1 + \delta y]$. En termes de probabilités, cela se traduit par

$$\text{Proba}(y_1 < y < y_1 + \delta y) = \text{Proba}(x_1 < x < x_1 + \delta x).$$

Nous savons que

$$\text{Proba}(x_1 < x < x_1 + \delta x) = p(x_1)dx$$

et que $\delta x = \delta y / g'(x_1)$. Nous en déduisons que

$$q(y_1) = \frac{p(x_1)}{|g'(x_1)|}, \tag{15.3}$$

où la valeur absolue tient compte de ce que g' peut être négative. Dans le cas où la fonction g est strictement croissante ou décroissante, la formule (15.3) se déduit de (15.2) par simple dérivation.

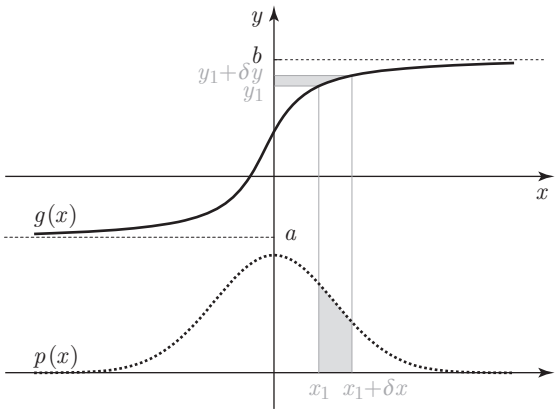


Figure 15.5 – Fonction d'une variable aléatoire : cas de la densité de probabilité.

15.2.2. Méthode de la fonction réciproque ou du changement de variable

Soit U une variable aléatoire à densité de probabilité constante sur l'intervalle $[a, b]$ $(a < b)$. Cette densité de probabilité s'écrit

$$p(u) = \frac{1}{b-a}, u \in [a, b], \quad p(u) = 0 \quad \text{autrement.}$$

Il lui correspond la fonction de répartition

$$P(u) = \frac{u-a}{b-a}, u \in [a,b], \quad P(u) = 0 \text{ si } u < a, P(u) = 1 \text{ si } u > b.$$

Vous vous souvenez (voir chapitre 14) que $P(u)$ est la probabilité pour que la relation $U \leq u$ soit vraie et que $p = P'$. Nous disposons, grâce aux considérations du paragraphe précédent, d'un algorithme ou d'un programme capable de fournir une suite de réalisations de la variable aléatoire U, soit $U_1, U_2, \ldots, U_n, \ldots$.

À partir de cette suite, nous voulons engendrer une nouvelle suite, $X_1, X_2, \ldots, X_n, \ldots$, qui soit caractérisée par la fonction de répartition $F(x)$ (ou par la densité de probabilité $F'(x) = f(x)$). Nous supposons que F est continue et strictement croissante (ce qui exclut les variables aléatoires discrètes). Sous ces conditions, F admet une fonction inverse (ou réciproque) notée F^{-1}. Nous pouvons écrire l'équivalence

$$y = F(x) \Longleftrightarrow x = F^{-1}(y),$$

à condition que $0 < y < 1$. Après ces préliminaires, nous pouvons énoncer la propriété suivante. La variable X définie comme

$$X = F^{-1}(U) \tag{15.4}$$

admet F comme fonction de répartition. En effet, la probabilité pour que $X \leq x$ est aussi la probabilité pour que $F^{-1}(U) \leq x$ et, d'après l'équivalence énoncée plus haut, c'est encore la probabilité que $U \leq F(x)$. Comme U est caractérisée par une répartition uniforme, cette dernière probabilité vaut $F(x)$.

Notre but est donc atteint en principe. Concrètement, par contre, la méthode qui vient d'être décrite ne sera applicable que si la fonction F^{-1} est facile à calculer. C'est le cas pour la distribution exponentielle dont la fonction de répartition est

$$F(x) = 1 - e^{-x}.$$

Si $y = F(x) = 1 - e^{-x}$, alors $x = F^{-1}(y) = -\log(1-y)$. U étant une variable aléatoire uniforme, la quantité $-\log(1-U)$ est distribuée selon une loi exponentielle. De plus, comme $1-U$ est uniforme si U l'est, $-\log U$ est aussi une variable aléatoire, de densité de probabilité exponentielle.

15.2.3. La méthode du rejet de von Neumann

Nous notons encore U un nombre aléatoire de densité de probabilité uniforme sur [0,1] et X un nombre aléatoire de densité de probabilité $p(x)$ et de fonction de répartition $P(x)$. Il faut que p soit à support borné : $f(x) = 0$ si $x \notin [a,b]$. Nous supposons encore qu'il existe un réel p_0 tel que $p(x) \leq p_0$ si $x \in [a,b]$, comme représenté sur la figure 15.6.

On tire des paires de valeurs aléatoires x_k, y_k, avec $a \leq x_k \leq b$ et $0 \leq y_k \leq p_0$. Chaque paire définit un point à l'intérieur du rectangle gris clair de la figure 15.6. Cette paire est « acceptée » si le point correspondant tombe entre la courbe représentative de p et l'axe horizontal (en gris foncé). La variable aléatoire x_k est alors distribuée selon la densité de probabilité $p(x)$.

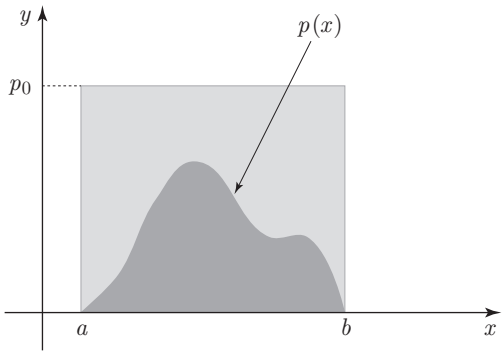


Figure 15.6 – La méthode du rejet : l'abscisse des points appartenant à la région gris foncé est distribuée selon la densité de probabilité $p(x)$.

15.2.4. La distribution normale

De nombreux algorithmes ont été proposés pour engendrer des nombres distribués selon cette loi qui apparaît si souvent dans les phénomènes naturels. Nous en décrirons deux.

La première méthode fait appel au « théorème central limite » : la somme de N variables aléatoires indépendantes et de lois identiques admet une loi de répartition qui tend vers la loi normale lorsque N devient « grand ». En pratique, $N = 12$ est déjà grand. Le programme correspondant est des plus simples, la seule petite difficulté consistant à ajuster la moyenne et l'écart-type de la variable somme (voir exercices).

La méthode de Box–Muller (parfois appelée méthode polaire) produit deux variables aléatoires gaussiennes X_1, X_2 à partir de deux nombres U_1, U_2 répartis uniformément sur [0,1]. Le pseudo-code correspondant est le suivant.

1. Engendrer U_1, U_2 et calculer $V_1 = 2U_1 - 1, V_2 = 2U_2 - 1$.
2. Calculer $S = V_1^2 + V_2^2$.
3. Si $S \leq 1$, former $X_1 = V_1\sqrt{\dfrac{-2\log S}{S}}$; $\quad X_2 = V_2\sqrt{\dfrac{-2\log S}{S}}$.
4. Sinon, recommencer au début.

La preuve de ce résultat est un peu semblable à la méthode classique de calcul de l'intégrale $\int \exp(-x^2)dx$. Le point $\boldsymbol{V}$ de coordonnées (V_1, V_2) est un point aléatoire distribué uniformément dans le carré centré sur l'origine et de coté 2 ; si $S < 1$ ce point est aussi à l'intérieur du disque de même centre et de rayon unité (toujours avec un répartition uniforme). Introduisons les coordonnées polaires de ce point par les relations : $V_1 = R\cos\Theta$, $V_2 = R\sin\Theta$. Alors $|OV|^2 = S = R^2$.

D'après l'algorithme, $\overrightarrow{OX}$ est proportionnel à $\overrightarrow{OV}$ et ses coordonnées peuvent s'écrire $X_1 = \sqrt{-2\log S}\cos\Theta$, $X_2 = \sqrt{-2\log S}\sin\Theta$. Les coordonnées polaires du point $\boldsymbol{X}$ sont évidemment $R' = \sqrt{-2\log S}$, $\Theta' = \Theta$.

Les variables aléatoires R et Θ sont indépendantes entre elles, de même que R' et Θ'. Les variables Θ et Θ' sont distribuées uniformément sur $[0, 2\pi]$, avec une densité de probabilité $1/2\pi$ (puisque $\boldsymbol{V}$ est réparti uniformément autour de l'origine).

La fonction de répartition de S, $P(s)$, est la probabilité pour que $S \leq s$; c'est le rapport de la surface du disque de rayon $\sqrt{s}$ à celle du disque de rayon 1, soit $P(s) = s$. La densité de probabilité correspondante est constante et égale à 1, tant que $s \leq 1$.

La probabilité pour que $R' \leq r$ est identique à la probabilité que $-2\log S \leq r^2$ ou encore que $S \geq \exp(-r^2/2)$. Comme S est uniforme sur [0,1], cette dernière probabilité vaut $1-\exp(-r^2/2)$. La densité de probabilité de R' est la dérivée de cette expression, soit $r\exp(-r^2/2)$.

Cherchons maintenant la probabilité pour que $X_1 \leq x_1$ **et** $X_2 \leq x_2$. Elle est donnée par l'intégrale

$$J = \int\limits_{\mathcal{D}} \frac{1}{2\pi} e^{-r^2/2} r dr d\theta,$$

où la région d'intégration $\mathcal{D}$ est définie par $r\cos\theta \leq x_1$, $r\sin\theta \leq x_2$. Revenant à des variables cartésiennes dans l'intégrale, nous avons

$$J = \frac{1}{2\pi} \int\limits_{x\leq x_1, y\leq y_1} e^{-(x^2+y^2)/2} dx dy, = \sqrt{\frac{1}{2\pi}} \int\limits_{-\infty}^{x_1} e^{-x^2/2} dx \times \sqrt{\frac{1}{2\pi}} \int\limits_{-\infty}^{x_2} e^{-y^2/2} dy.$$

La dernière forme est un produit de fonctions de répartition gaussiennes pour X_1 et X_2, ce qui montre que ces quantités sont indépendantes et obéissent à la loi normale. L'algorithme produit donc deux nombres aléatoires gaussiens.

15.3. Simulation de phénomènes aléatoires

Fermi et ses collaborateurs ont été les premiers à utiliser la méthode de Monte Carlo pour résoudre un problème de physique, la diffusion des neutrons dans un réacteur nucléaire. On sait en principe traiter la diffusion au niveau macroscopique en résolvant une équation aux dérivées partielles, mais cette approche devient très malcommode lorsque le milieu où se produit la diffusion est compliqué, comme c'est le cas pour un empilement de briques de graphite et de billes d'oxyde d'uranium, parcouru par un fluide de refroidissement. Il vaut mieux alors adopter un point de vue microscopique et considérer le mouvement d'un neutron comme une marche au hasard dans un milieu hétérogène. Il faudra bien sûr étudier assez de trajectoires pour avoir un échantillon représentatif. Cette démarche est d'autant plus adaptée que les propriétés du milieu sont décrites de façon probabiliste. Par exemple, on connaît la probabilité d'absorption d'un neutron par un noyau de carbone, de même que sa probabilité de diffusion. La petite histoire veut que l'un des collaborateurs de Fermi, S. Ulam, ait eu un oncle passionné de roulette et qui aurait été l'inspirateur du terme Monte Carlo.

Nous allons traiter deux exemples, la décroissance radioactive et l'agrégation limitée par la diffusion. Nous pensons que ce domaine présente peu de difficultés et que vous pourrez facilement traiter d'autres cas, en vous inspirant de ces exemples.

15.3.1. La radioactivité

Nous disposons d'un échantillon de substance radioactive comportant, à l'instant zéro, $n_{A0} = n_A(0)$ noyaux instables, appartenant à l'espèce A. Chaque noyau peut, de façon tout à fait aléatoire, subir une désintégration qui le transforme en un noyau stable, de l'espèce B. À l'instant t, l'échantillon contient $n_A(t)$ noyaux A et $n_B(t)$ noyaux de la substance « fille ». Il s'agit de déterminer ces deux fonctions.

La solution analytique est fort simple. Soit α la probabilité de désintégration par unité de temps pour un noyau. Entre les instants t et $t+\delta t$, nous observerons donc $n_A(t)\alpha\delta t$ désintégrations et, pendant cet intervalle de temps, la « population » de A variera de $\delta n_A = -\alpha n_A(t)\delta t$. Nous avons donc

$$\frac{\delta n_A}{\delta t} \simeq n'_A = -\alpha n_A(t).$$

La résolution de cette équation différentielle (linéaire, du premier ordre, à coefficients constants) est immédiate :

$$n_A(t) = n_{A0}\exp(-\alpha t).$$

La simulation de la décroissance par la méthode de Monte Carlo n'est pas plus difficile. L'échantillon sera représenté en machine par un vecteur à n_{A0} éléments, baptisé `tab`. Nous devrons choisir un code pour repérer les deux substances : les lettres A et B ou les entiers 0 et 1 par exemple (ce dernier choix étant le plus commode). Voici une ébauche du programme de simulation.

Listing 15.2 – Simulation d'une décroissance radioactive : principe

```
N = 1000; P = 0.1;                                          1
NR = 40; nB = zeros(NR+1,1);                                2
tab = zeros(1,N);                                           3
for ir = 1:NR                                               4
  for i = 1:N                                               5
      x = rand();                                           6
      if x <= P & tab(i) == 0 then tab(i) = 1; end          7
  end                                                       8
  nB(ir+1) = sum(tab);                                      9
end                                                         10
nA = N-nB;                                                  11
t = 1:NR+1;                                                 12
```

Les lignes 1 et 2 définissent les paramètres du problème, la ligne 3 initialise à 0 (A) le contenu de l'échantillon. Le tableau sera parcouru `NR` fois. Pour chaque case, nous produisons un nombre aléatoire ; s'il est inférieur à `P`, ce qui se produit avec une probabilité P et si nous sommes en présence d'un noyau A, nous le désintégrons pour le transformer en B. À tout moment de la simulation, le nombre de 1 dans `tab` est égal au nombre de noyaux B. En vue du tracé des résultats, nous définissons un « temps » conventionnel, égal au nombre de balayages effectués. La figure 15.7 montre une réalisation ; nous avons superposé la courbe de décroissance analytique.

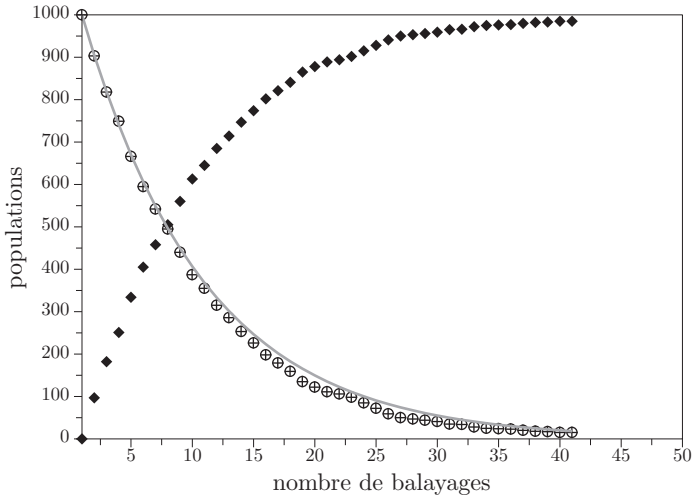


Figure 15.7 – Décroissance d'une espèce radioactive et croissance de l'espèce « fille ». La courbe est la solution analytique.

15.3.2. L'AGRÉGATION

Comment se forment les particules de suie dans une cheminée, les dépôts de métal lors d'une électrolyse ou encore les flocons de neige ? À partir d'un germe ou d'une amorce, par « capture » d'éléments supplémentaires (atomes, molécules ou petites particules). Il peut se trouver que le seul mécanisme de transport vers le voisinage de l'agrégat en formation soit la diffusion ; on parle alors d'agrégation limitée par la diffusion.

Il existe de nombreux modèles de ces phénomènes. Comme pour la diffusion proprement dite, qui peut être représentée par une marche au hasard, nous pouvons imaginer que le déplacement des éléments qui vont constituer l'agrégat se fait sur un réseau ou dans un espace continu, à deux ou à trois dimensions, avec des pas de longueur constante ou aléatoire.

Il faut aussi préciser comment se fait la capture. Lorsqu'une particule, dans sa marche aléatoire, parvient au contact de la surface de l'agrégat, elle peut être instantanément capturée, ou avoir une certaine probabilité de rebondir ; cette probabilité peut, à son tour, dépendre du nombre de particules voisines déjà immobilisées.

Quel que soit le modèle, les grandes lignes de l'algorithme seront les suivantes.

– Choisir un germe.

– Lancer, à partir d'une origine quelconque mais assez éloignée, un mobile.

– Faire décrire à ce mobile une marche aléatoire jusqu'à ce qu'il soit adsorbé ou perdu loin de l'agrégat.

– Recommencer un lancement, à partir d'une autre origine. À mesure que l'agrégat grossit, éloigner le point de départ.

La figure 15.8 représente le résultat d'une simulation, avec les règles suivantes. Le « milieu » est représenté par une matrice A d'entiers, à M lignes et N colonnes. Les particules fictives se déplacent sur un réseau carré, d'une unité vers la droite, la gauche, le haut ou le bas. Les éléments de A sont initialisés à zéro, sauf pour le germe ($A(M/2, N/2) = 3$) et pour ceux de la première et dernière ligne et de la première et dernière colonne, qui valent -1, ce qui permet de détecter facilement la sortie du « milieu ». Le lancement se fait en un point aléatoire d'un cercle (en gris pâle, code 2) et le rayon de ce cercle double dès que l'agrégat s'en approche. Les points noirs (code 0) n'ont jamais été atteints par les particules, alors que les cases grises (code 1) ont été « visitées » au moins une fois. L'agrégat lui-même est en gris foncé (code 3).

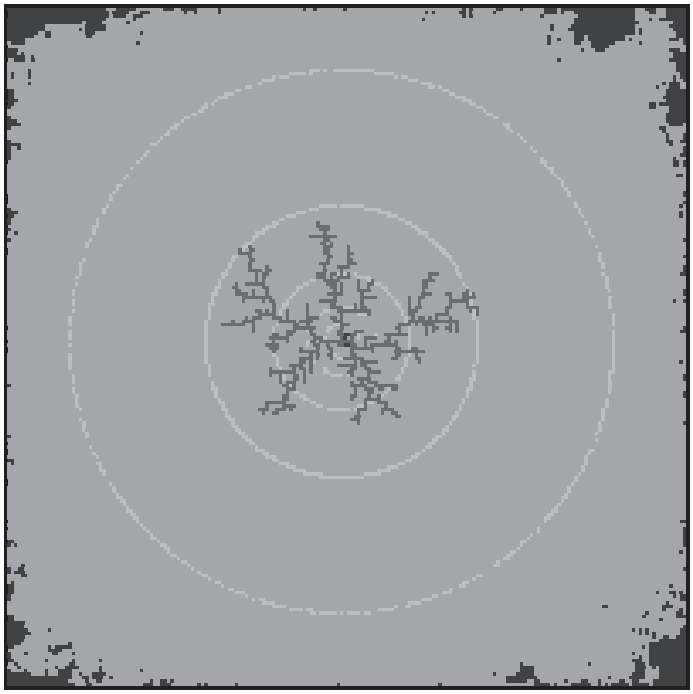


Figure 15.8 – Agrégation limitée par la diffusion sur un réseau plan carré.

15.4. Méthodes de Monte Carlo déterministes : calcul d'intégrales

Il s'agit maintenant d'utiliser une méthode stochastique pour calculer ou plutôt estimer une quantité certaine. Nous commençons par un exemple très classique, le calcul de π.

15.4.1. Calcul de π

Soit un carré de coté a, dont le coin inférieur gauche coïncide avec l'origine (fig. 15.9).

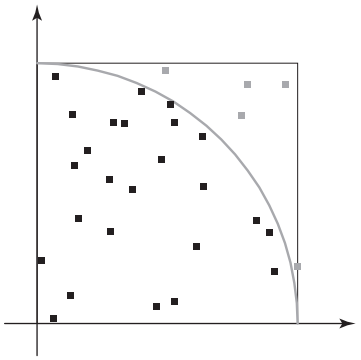


Figure 15.9 – Calcul de π.

Un quart de cercle, de centre O, est inscrit dans ce carré. Le rapport (aire du quart de cercle)/(aire du carré) vaut $\pi/4$. Nous pouvons estimer ce rapport de la façon suivante. Nous choisissons au hasard N points dans le carré (en fabriquant deux coordonnées aléatoires par point). Parmi ceux-ci, N_i tombent à l'intérieur du quart de cercle. L'événement « un point, pris au hasard dans le carré, est aussi dans le quart de cercle » a une probabilité égale au rapport des surfaces de ces deux figures. La fréquence de cet événement, rapportée au nombre de tirages (N_i/N) est un estimateur de cette probabilité. Pour notre exemple (figure 15.9), $N = 30$, $N_i = 25$ et $\pi \simeq 3,3$. La variable N_i obéit à une loi binomiale d'écart-type $\sigma_{Ni} = \sqrt{Np(1-p)} \simeq 2,25$, ce qui nous conduit au résultat final $\pi = 3,3 \pm 0,07$.

15.4.2. Avantages et inconvénients des méthodes stochastiques pour le calcul d'intégrales

Les aires des deux figures géométriques de la section précédente peuvent être considérées comme des intégrales, $\mathcal{I} = \int_0^1 dx$ pour le carré et $\mathcal{J} = \int_0^1 \sqrt{1-x^2}dx$ pour le quart de cercle. Sans nous en apercevoir, nous venons donc de calculer des intégrales par une méthode de Monte Carlo. Nous pourrions même nous vanter d'avoir calculé des intégrales doubles comme $\iint_{\mathcal{D}} dxdy$, où $\mathcal{D}$ est l'un des deux domaines précités.

Nous avons décrit (chapitre 8) de nombreuses méthodes numériques d'approximation d'une intégrale. Elles se caractérisent toutes en pratique par le fait que l'intervalle d'intégration est partagé en N sous-intervalles de longueur h, que l'on doit calculer l'intégrant N fois et que l'erreur de troncation varie comme h^k ou comme $1/N^k$. En principe, ces algorithmes s'appliquent tout aussi bien aux intégrales multiples. Cependant, pour calculer une intégrale à d dimensions, il nous faudra ou bien évaluer l'intégrant N^d fois ou bien diminuer N et la précision. Le calcul de la répulsion électrostatique entre deux électrons d'un atome fait intervenir une intégrale à 6 dimensions, une tâche trop lourde pour un algorithme classique. Le calcul direct des propriétés thermodynamiques d'un fluide à partir des positions et des vitesses des particules qui le constituent est un cas extrême : il faudrait calculer des intégrales portant sur $\simeq 10^{23}$ variables.

Comme le suggère l'exemple du calcul de π et comme nous allons le voir plus en détail, l'erreur de méthode dans le cas d'un calcul de Monte Carlo dépend essentiellement de $\sqrt{N}$, quelle que soit la dimensionnalité du domaine d'intégration. Un autre avantage des méthodes de Monte Carlo est qu'il est en général beaucoup plus facile de prendre en compte la forme du domaine d'intégration lorsque celle-ci est compliquée.

Ces raisons nous conduisent à exposer plusieurs variantes de la méthode de Monte Carlo appliquée au calcul intégral. Cependant, pour simplifier l'exposé, nous ne calculerons que des intégrales à une dimension.

Vous avez dû remarquer que notre méthode d'estimation de π était très semblable à la méthode du rejet de von Neumann. Nous traitons le cas général dans le paragraphe suivant.

15.4.3. Intégrales par la méthode du rejet

La figure 15.6 peut commodément illustrer la méthode. Nous voulons calculer

$$\mathcal{J} = \int_a^b p(x)dx,$$

où $p(x)$ n'est plus interprétée comme une densité de probabilité, mais comme la fonction à intégrer ; elle reste néanmoins strictement positive et intégrable sur $[a, b]$. Nous tirons des points au hasard dans le rectangle gris claie de base (a, b) et de hauteur $p_0 > p(x)$, dont l'aire est $A = (b - a)p_0$. La probabilité pour que l'un de ces points tombe dans l'aire gris foncé est $\theta = \mathcal{J}/A$. Nous estimons cette probabilité par le rapport $\theta^* =$ (nombre de points entre la courbe et l'axe)/(nombre total de points) et nous en déduisons une estimation $\mathcal{J}^*$ de $\mathcal{J}$.

Exemple – Cherchons la valeur de $\mathcal{J} = \int_0^\pi \sin x dx$. Nous connaissons bien sûr le résultat exact, $\mathcal{J} = 2$. Le rectangle choisi a une largeur de π, une hauteur $p_0 \equiv 1$ et une aire $A = \pi$. Nous avons trouvé que, sur $N = 1000$ points, $N_i = 634$ tombaient entre la courbe et l'axe horizontal, d'où les estimations $\theta^* = 0.634$ et $\mathcal{J}^* = 1,9918$.

Quelle peut être la précision de ce résultat et, plus généralement, celle de la méthode du rejet ? Dans le cas particulier de cet exemple, nous pouvons donner deux réponses : théorique et empirique. La méthode du rejet est en fait une « épreuve de Bernoulli » dont les résultats se répartissent selon une loi binomiale (§ 14.2.1), de probabilité θ. Nous savons que pour N tirages

$$\langle N_i \rangle = \theta N, \quad \sigma_{N_i} = \sqrt{\theta(1-\theta)N}.$$

Nous en déduisons que l'estimateur de l'intégrale, $\mathcal{J}^* = \pi N_i/N$ a pour écart-type

$$\sigma_{\mathcal{J}^*} = \pi\sqrt{\frac{\theta(1-\theta)}{N}}.$$

Comme $\theta = 2/\pi$ et $N = 1000$, il vient $\sigma_{\mathcal{J}^*} = 0,0477$.

Pour vérifier numériquement les résultats précédents, nous devons disposer de plusieurs résultats indépendants. Au lieu d'engendrer 1000 points, nous allons tirer dix

fois cent points ; nous obtiendrons ainsi dix réalisations d'une épreuve de Bernoulli ce qui nous permettra d'estimer l'écart-type. Nous pourrons enfin faire la moyenne de ces dix résultats partiels pour obtenir une précision identique au tirage de 1000 points. Nous avons trouvé des valeurs de $\mathcal{J}^*$ comprises entre 1,7 et 2,3, ce qui est compatible avec la valeur prédite de l'écart-type (0,151 pour $N = 100$) et avec sa valeur estimée : 0,162.

La démarche précédente est souvent utilisée dans le domaine des méthodes stochastiques. Plutôt que de faire une simulation portant sur N valeurs aléatoires, il est plus profitable de faire k simulations indépendantes portant sur N/k valeurs, afin de pouvoir contrôler les propriétés statistiques des résultats.

15.4.4. Intégrales par la valeur moyenne

Soit $f(x)$ une fonction définie sur $[a, b]$ dont nous cherchons l'intégrale $\mathcal{J} = \int_a^b f(x)dx$. Si X est une variable aléatoire répartie selon une loi uniforme sur le même intervalle (avec une densité de probabilité $1/(b-a)$), $f(X)$ est aussi un nombre aléatoire, dont la moyenne est

$$\langle f \rangle = \frac{1}{b-a} \int_a^b f(x)dx = \frac{\mathcal{J}}{b-a}$$

et la variance

$$\sigma^2 = \frac{1}{b-a} \int_a^b [f(x) - \langle f \rangle]^2 dx = \langle f^2 \rangle - \langle f \rangle^2.$$

Si $x_1, x_2, \ldots, x_n$ sont n réalisations de X, les quantités $f_i = f(x_i)$ constituent un échantillon de la variable aléatoire $f(X)$. La moyenne d'échantillon est un estimateur sans biais de la moyenne de la population, soit

$$\bar{f} = \frac{1}{n} \sum_{i=1}^{n} f_i \simeq \langle f \rangle.$$

L'écart-type de $\bar{f}$ est

$$\sigma_{\bar{f}} = \frac{\sigma}{\sqrt{n}}.$$

En revenant à $\mathcal{J}$, nous avons

$$\mathcal{J}^* = \frac{b-a}{n} \sum_{i=1}^{n} f_i, \quad \sigma_{\mathcal{J}^*} = (b-a)\sigma_{\bar{f}} = \frac{b-a}{\sqrt{n}}\sigma. \tag{15.5}$$

Cette formule d'intégration ressemble beaucoup à la méthode des rectangles (chapitre 8, où le choix des abscisses (pivots) aurait été laissé au hasard et où les longueurs des intervalles h_i auraient été remplacées par leur valeur moyenne $(b-a)/n$. Vous remarquez aussi la similitude avec la formule de la moyenne énoncée au début du même chapitre.

En général, nous ne savons pas calculer analytiquement les intégrales de f ou de f^2. La variance de l'échantillon est un estimateur de σ^2 :

$$s^2 = \frac{1}{n-1}\sum_1^n (f_i - \bar{f})^2.$$

Exemple – Reprenons le calcul de l'intégrale de $\sin x$. Nous avons les valeurs théoriques

$$\langle f\rangle = \frac{1}{\pi}\int_0^\pi \sin x dx = 0,6366\ ; \quad \langle f^2\rangle = \frac{1}{\pi}\int_0^\pi \sin^2 x dx = \frac{1}{2},$$

et donc $\sigma = \sqrt{\langle f^2\rangle - \langle f\rangle^2} = 0,3078$.

Si notre programme utilise $n = 100$ valeurs aléatoires, nous devons nous attendre à trouver $\mathcal{J}^* = \pi\bar{f}$ avec un écart-type de $\pi\sigma/10$. Nous avons exécuté un programme qui calcule dix fois $\mathcal{J}^*$ à partir de cent points ; les résultats se sont répartis entre 1,938 et 2,138, avec une moyenne de 2,045. Les valeurs de σ fluctuaient entre 0,25 et 0,32, ce qui conduit à un écart-type pour $\mathcal{J}^*$ de l'ordre de 0,1.

La dispersion est inférieure de 50% à ce qu'elle était avec la méthode du rejet. Ce résultat est général : on peut montrer que le calcul d'une intégrale utilisant la valeur moyenne de la fonction est toujours plus précis (à nombre d'évaluations de f constant) que celui faisant appel au rejet.

15.5. Pour en savoir plus

– D. Knuth : *The art of computer programming, volume 2 : seminumerical algorithms*, ch. 3 (Addison-Wesley, Reading, Mass., 1969).

– TestU01 : A C Library for Empirical Testing of Random Number Generators
P. L'Écuyer et R. Simard, ACM Transactions on Mathematical Software, **33** (4) Article 22, août 2007.

– The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness, `http://www.stat.fsu.edu/pub/diehard`

– B. Gaujal : *La simulation en science*, INRIA et ID-IMAG,séminaire, Septembre 2006.

– `http://stp.clarku.edu/simulations/` : de nombreuses applets illustrant tous les aspects de la physique statistique.

– R.H. Landau, M.J. Paez : *Computational Physics, problem solving with computers*, ch. 6,7 (Wiley, New York, 1997).

– H. Gould, J. Tobochnik, W. Christian : *An introduction to computer simulation methods* (Addison-Wesley, Reading, Mass., 2006).

– Sur le site `http://www.phytem.ens-cachan.fr/coursenligne1A.htm` :
modèle d'Ising.

15.6. EXERCICES

Exercice 1

Simuler une décroissance radioactive faisant intervenir une espèce intermédiaire :

$$A \longrightarrow B \longrightarrow C$$

la probabilité de désintégration de B en C étant deux fois inférieure à celle de la transformation de A en B.

Exercice 2

L'histoire a retenu une autre méthode d'estimation de π, sous le nom de problème de Buffon. On dispose d'un grand nombre d'aiguilles identiques de longueur a. On jette ces aiguilles au hasard sur un parquet dont les lattes, toutes parallèles, ont une largeur $\ell \geq a$; la largeur des rainures est négligeable. On demande la probabilité pour qu'une aiguille tombe à cheval sur deux lattes, en coupant une rainure.

Il est commode d'introduire les paramètres géométriques de la figure 15.10 : y est la distance du milieu de l'aiguille à la rainure, θ est l'angle aigu entre l'aiguille et la direction des lattes.

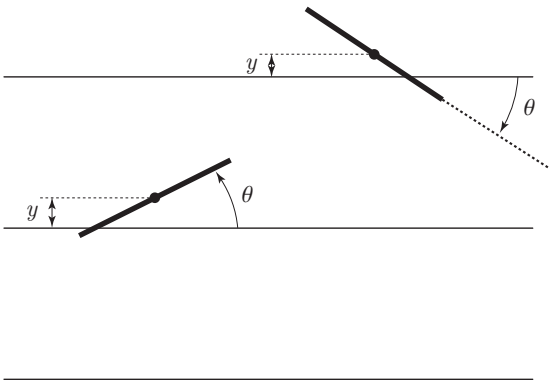


Figure 15.10 – Le problème de Buffon : les deux cas d'intersection.

a) Trouver une relation entre y et θ pour que l'aiguille coupe une rainure.

b) y et θ sont deux variables aléatoires indépendantes. Exprimer leur densité de probabilité et leur fonction de répartition.

c) La position d'une aiguille peut être représentée par un point du plan (θ, y). Quelles sont les régions du plan qui correspondent à une intersection ? Quelle est la probabilité d'y trouver un point figuratif ?

d) Simuler le lancement d'aiguilles pour estimer π (la convergence est lente !).

Exercice 3

a) Un GNA fournit des réels répartis uniformément dans $[0, a]$. Quelle est la moyenne et l'écart-type de cette variable aléatoire ?

b) On utilise maintenant la somme de p nombres aléatoires fournis par le générateur précédent pour approcher une variable aléatoire gaussienne, notée Y. Calculer la moyenne $\langle Y \rangle$ et l'écart-type σ_Y.

c) Réciproquement, comment faut-il ajuster la somme pour obtenir une variable gaussienne de moyenne μ et d'écart-type σ ?

Exercice 4

Nous considérons dans cet exercice une variante de la méthode de Box–Muller. Nous souhaitons engendrer des nombres aléatoires répartis selon la loi normale générale

$$p(x)dx = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} dx.$$

Nous faisons pour cela un détour par une distribution gaussienne à deux variables indépendantes et de mêmes paramètres

$$p(x)p(y)dxdy = \frac{1}{2\pi\sigma^2} e^{-[(x-\mu)^2+(y-\mu)^2]/2\sigma^2} dxdy.$$

a) Trouver le changement de variables qui permet d'écrire cette densité de probabilité sous la forme

$$p_\rho(\rho)p_\theta(\theta)d\rho d\theta = \frac{1}{2\pi\sigma^2} e^{-\rho^2/2\sigma^2} \rho d\rho d\theta.$$

b) Nous introduisons un nouveau changement de variables $u = \rho^2/2\sigma^2$, $v = \theta/2\pi$. Quelles sont les densités de probabilité pour u et v ? Comment s'écrit la densité de probabilité conjointe $p_u p_v du dv$?

c) En utilisant l'algorithme du texte pour engendrer une variable aléatoire à densité exponentielle, écrire un programme pour former, à partir de deux nombres aléatoires u, v répartis uniformément sur [0,1], deux nombres x, y à répartition gaussienne.

d) Comparer l'efficacité de cet algorithme avec celle de la méthode de Box–Muller.

Exercice 5

Les nombres $x_i, i = 1 \ldots n$, sont issus d'une distribution uniforme sur [0,1]. Montrer que la quantité

$$\frac{1}{n} \sum_1^n f(x_i)$$

est un estimateur sans biais de

$$\int_0^1 f(x)dx$$

pour une fonction f intégrable sur cet intervalle.

Exercice 6

On connaît la fonction de répartition $F(i)$ d'une variable aléatoire discrète i à valeur positive ou nulle. Montrer que la fonction ci-dessous engendre des entiers aléatoires tirés de cette distribution et compris entre $n1$ et $n2$.

```
function y = aleaF(n1,n2)
    r = rand();
    for i = n1:n2
        if r < F(i) then break end
    end
    y = i;
endfunction
```

15.7. PROJET

Le modèle d'Ising

Ce projet est consacré au modèle d'Ising, une représentation très simplifiée d'un milieu magnétique. On ne connait de solution analytique que pour le cas à une dimension et à deux dimensions en champ nul. Les méthodes de Monte Carlo sont bien adaptées à la simulation de l'équilibre thermique du modèle d'Ising. Nous vous proposons de mettre en oeuvre un algorithme particulier appelé algorithme de Metropolis.

Un échantillon est constitué d'un arrangement régulier de N atomes immobiles (en file ou sur un réseau à 2, 3 ou d dimensions). Chaque atome porte un moment magnétique. Ces vecteurs n'ont qu'une composante non nulle, disons m_z, et celle-ci est quantifiée : elle ne peut prendre que l'une des deux valeurs ± 1. Dans la suite, nous ne considérerons que les propriétés associées à ces moments magnétiques. De ce point de vue, l'état de l'échantillon peut être décrit en détail par l'énumération des valeurs des m_{iz} : $m_{1z} = 1, m_{2z} = -1, \ldots, m_{Nz} = -1$. On dit qu'on a spécifié un « micro-état » ou une « configuration ».

Le système ne présente qu'une seule forme d'énergie, une interaction magnétique entre moments voisins. Pour une chaîne linéaire (file) d'atomes, la contribution de l'atome i s'écrit $E_i = -Jm_{iz}m_{i+1z}$ et l'énergie de la chaîne entière vaut

$$E = -J \sum_{1}^{N-1} m_{iz}m_{i+1z}. \tag{15.6}$$

Dans ces expressions, la constante J décrit la force de l'interaction et le signe moins est conventionnel. On écrit des expressions analogues, faisant intervenir les plus proches voisins du moment i, dans les cas à deux ou plusieurs dimensions.

Une simulation réaliste devra faire intervenir des milliers d'atomes. Or, il y a 2^N micro-états possibles et il serait impossible de seulement les énumérer tous. On a donc recours à la mécanique statistique et, comme le problème posé n'a, à ce jour, reçu de solution analytique que pour le cas à une dimension et à deux dimensions en champ nul, à la simulation numérique.

Un mécanisme non précisé maintient le système de moments magnétiques à l'équilibre thermique à la température T. Appelons c_j une configuration particulière du système de moments magnétiques. Sa probabilité d'apparition est donnée par l'exponentielle de Boltzmann

$$p(c_j) = \frac{1}{Z} \exp\left[-\frac{E(c_j)}{kT}\right], \quad Z = \sum_{c_j} \exp\left[-\frac{E(c_j)}{kT}\right].$$

La prescription suivante permet de choisir les configurations probables.

1. Choisir une configuration initiale, c_k.
2. Engendrer la configuration suivante, c_{k+1}, selon la règle
   a) choisir une particule, i, au hasard
   b) retourner son moment magnétique pour créer une configuration d'essai, c_e ($m^e_{iz} = -m^k_{iz}$).
   c) calculer $E(c_e)$ et poser $\delta E = E(c_e) - E(c_k)$.
   d) si $\delta E < 0$, accepter cette configuration : $c_{k+1} = c_e$.
   e) si $\delta E > 0$, accepter cette configuration avec la probabilité $p = \exp(-\delta E/kT)$. Pour cela
   – engendrer r de loi uniforme sur [0,1]
   – poser $c_{k+1} = c_e$ si $p \geq r$ (acceptation) et $c_{k+1} = c_k$ si $p < r$ (rejet).

Quelle que soit la géométrie du système, il ne peut apparaître que quelques valeurs de δE et de $\exp(-\delta E/kT)$; celles-ci peuvent être calculées à l'avance. On dit qu'on a accompli un « balayage de Metropolis » lorsque chaque moment magnétique a été choisi **en moyenne** une fois. Metropolis a démontré que la suite de configurations créées de cette façon correspondait à celles que l'on attend à l'équilibre thermique.

1. Écrire un programme pour appliquer l'algorithme de Metropolis au problème d'Ising à une dimension. Choisir $N \simeq 20$ au début, $N \simeq 1000$ quand le programme est au point. Les résultats ne dépendent que de $x \equiv kT/J$ et du signe de J ; on peut choisir $J = 1$ (milieu « ferromagnétique ») ou $J = -1$ (échantillon anti-ferromagnétique). Pour diminuer les effets de la petite taille de l'échantillon, on rend les données périodiques : $m_{Nz} \equiv m_{1z}$. Une méthode consiste à surveiller chaque indice avant utilisation : `if j > N then j = j - N` et `if j < 1 then j = j+ N` La configuration initiale peut être choisie au hasard (si la température est élevée), ou tous les m_i de même signe (arrangement ferromagnétique) si la température est basse et $J > 0$ ou encore tous les m_{iz} de signes alternés, si $J < 0$ et la température faible.
2. L'énergie d'une configuration vaut

$$E(c_k) = -J \sum_{1}^{N-1} m_{iz} m_{i+1z},$$

   Introduire cette expression dans le programme. À chaque température, il faut laisser passer un nombre de tirages au moins égal à N pour que le système parvienne à

l'équilibre thermodynamique, avant de calculer l'énergie et de faire la moyenne sur un certain nombre de configurations. Lorsque le tirage au sort a conduit à un rejet de la configuration d'essai, la nouvelle configuration (c_{k+1}) est identique à l'ancienne (c_k) ; elle doit quand même être prise en compte dans le calcul de la moyenne. Représenter les variations de $\langle E \rangle$ avec T. Le résultat théorique s'écrit

$$\langle E \rangle = -NJ \tanh \frac{J}{kT}.$$

3. Modifier le programme pour faire intervenir un champ magnétique extérieur, qui apporte à l'énergie de chaque moment la contribution $E_i = -m_{iz}B$. Déterminer l'aimantation du milieu en présence du champ :

$$M = \sum_{1}^{N} m_{iz}$$

et comparer à la valeur théorique

$$\langle M \rangle = \frac{Ne^{J/kT}\mathrm{sh}(B/kT)}{\sqrt{e^{2J/kT}\mathrm{sh}^2(B/kT) + e^{-2J/kT}}}.$$

Index

agrégation limitée par la diffusion, 348
ajustement, 324, 327
algorithme, 23
    de Crout, 126
    de Cooley–Tukey, 194
    de Doolittle, 126
    de Horner, 24, 97
    de Metropolis, 356
    de Neville, 84
    de Numerov, 263
    de Remes, 32
    de Verlet, 262
    du saute-mouton, 262
    symplectique, 263
aliasing, 204
analyse dimensionnelle, 49
approximant de Padé, 28
approximation de fonction, 31
aurore boréale, 275
axes principaux d'une ellipse, 216

`bdiag`, 211
biais, 318

Calc, 9
coefficient de corrélation, 329
cohérence d'un schéma numérique, 252
condition
    de Courant, Friedrichs et Levy, 304
    de Lipschitz, 240, 253, 254
conditionnement, 111
conditions aux limites, 277–279
congruence linéaire, 338
consistance d'un schéma numérique, 252
constante de stabilité, 253
convergence d'un schéma numérique, 254, 261–262
courbes de Bézier, 83

décomposition LU, 117–119, 302
densité de probabilité, 310
dérivée
    d'une fonction analytique, 160–164
    d'une fonction empirique, 164
déterminant, 118, 141
    de van der Monde, 59
développement
    asymptotique, 33
    limité, 26, 241
`dft`, 199–204
`diary`, 116
différence latérale, 68, 282–287, 298–304
différences divisées, 63
dimension, 49–53
division des polynômes, 96–98
`DROITEREG`, 328
`dsolve`, 268, 288

écart-type, 310
échantillonnage, 189
`eigenvalues`, 211
`eigenvectors`, 211
entrelacement
    en fréquence, 199
    en temps, 194
équation
    de Schrödinger, 285
    de van der Waals, 107
    différentielle raide, 264
    elliptique, 298
    hyperbolique, 303
    parabolique, 300
équations normales, 325, 330
erreur
    aléatoire, 318
    d'arrondi, 27, 36, 113
    d'interpolation, 66–67
    de troncation, 27, 283

locale de troncation, 252
systématique, 318
espérance, 310
estimateur, 316
Excel, 299, 317, 328

factorisation
de Cholesky, 127
directe, 125–127
`fft`, 199–205
`fftshift`, 202
fonction
d'Airy, 38
d'une variable aléatoire, 342
de Bessel, 20, 38
de poids, 145, 149, 175
de Riemann, 43
de répartition, 310
de vraisemblance, 322
$\Gamma(x)$, 314
génératrice, 149
incrément, 246
forme barycentrique de l'interpolation de Lagrange, 82
formule
de la moyenne, 160, 352
de Newton, 69–71
de Richardson, 163, 172, 180
de Rodrigues, 150
de Taylor, 160
formules composites, 171–172
fraction
continue, 40
rationnelle, 25
fréquence de Nyquist, 190, 203
`fsolve`, 95

générateur de nombres aléatoires, 338–341
Gnu Scientific Library, 30
gnuplot, 12
graine, 339
`grand()`, 339

identité
de Darboux–Christofel, 150
`ifft`, 200
image, 110, 140
`int`, 179
intégrale
elliptique, 45
généralisée, 177
multiple, 177
`integrate`, 178
intégration de Gauss–Laguerre, 176
intégration numérique, 165–179
`interp`, 77
interpolant
spline complet, 77
spline naturel, 82
spline not-a-knot, 77
interpolation
inverse, 72
linéaire, 59
par intervalle, 73
spline, 74–78
à deux dimensions, 78
intervalle
fermé, 168
ouvert, 170
tabulaire, 25, 69
`intg`, 178

Java, 10
justesse, 318

lissage, 324
loi
binomiale, 311
de Gauss, 313
de Poisson, 312
de probabilité, 311–317
du χ^2, 314
normale, 313
normale réduite, 314
uniforme, 313
`LOI.CHI-DEUX`, 317
`lu`, 119
`LUdecomp`, 119

Maple, 13
marche aléatoire, 340
Matplotlib, 11
matrice
à diagonale dominante, 127
bande, 128

de Frobenius, 119, 137
des variances-covariances, 325, 330
hermitienne, 230
inverse, 113, 125, 141
régulière, 110, 141
semblable, 210
singulière, 110, 209
symétrique définie positive, 127
méthode
d'Adams–Bashforth, 256
d'Adams–Moulton, 258
d'intégration de Gauss, 174–176
de bissection, 86–87, 229, 280
de Box–Muller, 345
de Cholesky, 127, 283, 330
de Cramer, 112
de Crank et Nicolson, 302
de diagonalisation de Jacobi, 215
de Gauss, 113–125, 283
de Gauss–Jordan, 117, 125
de Gauss–Seidel, 132
de Jacobi, 130
de la puissance n-ième, 211–215
de la puissance n-ième de l'inverse, 213
de la sécante, 93
de Newton, 90–93, , 229, 280
de Picard, 242
de prédiction-correction, 258–261
de relaxation, 299
de Romberg, 172
de surrelaxation, 133
des coefficients indéterminés, 59–60, 162, , 169, 242
des différences finies, 282, 297
des moindres carrés, 323–326
des parties proportionnelles, 87–88
du maximum de vraisemblance, 321–331
du point fixe, 88–90
du point milieu, 167
du rejet, 344, 351
du tir, 279–283
du trapèze, 168
PECE, 258
QR, 224–226
RK4, 249
méthodes
de Newton–Cotes, 168–172
itératives pour les systèmes linéaires, 130–135
modèle
d'Ising, 356
de Kronig–Penney, 292

nombre
d'opérations pour la méthode de Gauss, 124
flottant, 35
norme
de Frobenius, 142, 218
extradiagonale, 218
induite, 142
matricielle, 142
vectorielle, 141
noyau, 110, 140
Numpy, 11

`ode`, 265
OpenOffice, 9
opération sur des blocs, 143
ordre du schéma RK2, 246
ordre, stabilité, convergence
des schémas multipas, 261–262
des schémas à un pas, 251–254
oscillateur harmonique quantique, 285

papillon, 196
pendule double, 274
pendule élastique, 273
permutation de lignes, 121–123
perturbation
des coefficients, 111
des seconds membres, 111
pgplot, 16
phénomène de Runge, 67, 78
pivot, 114
pivot maximal, 121
Plotutils, 16
points de Tschebychef, 67
polynôme, 24
caractéristique, 209
d'interpolation de Hermite, 71–72
de Hermite, 151
de Jacobi, 154

de Lagrange, 60–61
de Laguerre, 152
de Laguerre généralisé, 154
de Legendre, 25, 150, 175
de Newton, 62–65
de Tschebychef, 153
de Wilkinson, 103
minimax, 32
précision, 318
problème
aux valeurs propres généralisé, 285
de Buffon, 354
de Cauchy, 240, 277
de Dirichlet, 298
de Sturm-Liouville, 279
de von Neumann, 298
différentiel homogène, 278
modèle, 253, 270
à condition initiale, 240
procédé de Gram–Schmidt, 146
produit scalaire de fonctions, 145
programme, 23
propagation des erreurs, 320–321
PtPlot, 10
Python, 11

`qr`, 226
`QRdecomp`, 226
quadrature numérique, 165

racines des polynômes, 96–104
`rand()`, 339
rang, 140
`read`, 16
`readdata`, 13
réduction à la forme tridiagonale, 227–230
règle de Descartes, 98
relation de récurrence, 25
renversement binaire, 197
rotation
de Givens, 217
de Jacobi, 217

`Scanner`, 11
schéma
d'Euler, 243–245
d'Euler amélioré, 247
d'Euler implicite, 261, 302
d'Euler modifié, 247
de Heun, 247
de Runge–Kutta, 246–251, 282
de Runge–Kutta–Fehlberg, 250
explicite, 255
implicite, 257
à pas multiples, 255–261
Scilab, 14
Scipy, 11
série de Fourier, 187
similitude, 210
`spec`, 210
`splin`, 77
stabilité d'un schéma numérique, 252
suites de Sturm, 99–101, 148, 229
symbole de Kronecker, 60
système
d'équations non-linéaires, 93–96
différentiel, 240
surdéterminé, 135–136
tridiagonal, 129, 302

tableur, 9
`TEST.CHI-DEUX`, 317
théorème
central limite, 345
de Buckingham, 52
de Shannon, 190
de Sturm, 99
de Weierstrass, 31
des accroissements finis, 159
des valeurs intermédiaires, 85
transformation
conforme, 46
de Fourier, 188
de Fourier discrète, TFD, 191
de Fourier rapide, TFR, 193
de Householder, 221–223, 227
de Joukovsky, 47
Tschebychef, 32
tuyau sonore, 291

valeur moyenne, 310
de la Vallée–Poussin, 32
variable aléatoire, 309
variance, 310

xmgrace, 16

Table des matières

Avant-propos 5

Chapitre 1. Représentation graphique de fonctions 9

1.1. Les tableurs 9

1.2. Java et PtPlot 10

1.3. Python et Matplotlib 11

1.4. Gnuplot 12

1.5. Maple 13

1.6. Scilab 14

1.7. Grace 16

1.8. Pour en savoir plus 16

1.9. Exercices 18

Chapitre 2. Calcul et approximation de fonctions 23

2.1. Polynômes et fractions rationnelles 24

2.2. Relations de récurrence 25

2.3. Développement limité 26

2.4. Approximant de Padé 28

2.5. Utilisation de bibliothèques de programmes 30

2.6. Approximation de fonctions 30

2.7. Développement asymptotique 32

2.8. Représentation des nombres en machine 34

2.8.1. Les nombres entiers 34

2.8.2. Les nombres fractionnaires 35

2.9. Pour en savoir plus 36

2.10. Exercices 37

2.11. Projets 43

Chapitre 3. Représentation des grandeurs physiques 49
3.1. Une méthode simple de « dédimensionnement » 50
3.2. Construction systématique de variables sans dimension 51
3.3. Pour en savoir plus 53
3.4. Exercices 53

Chapitre 4. L'interpolation 57
4.1. Définition de l'interpolation 58
4.2. Méthode des coefficients indéterminés 59
4.3. Le polynôme d'interpolation de Lagrange 60
4.4. Le polynôme de Newton 62
4.4.1. Interpolation linéaire 62
4.4.2. Les différences divisées 63
4.4.3. La formule de Newton 63
4.5. L'erreur d'interpolation 66
4.6. Interpolation entre pivots équidistants 68
4.6.1. Les différences finies latérales 68
4.6.2. La formule d'interpolation de Newton 69
4.7. Le polynôme d'interpolation de Hermite 71
4.8. L'interpolation inverse 72
4.9. L'interpolation par intervalle 73
4.10. L'interpolation « spline » 74
4.11. Interpolation à deux ou plusieurs dimensions 78
4.12. Pour en savoir plus 79
4.13. Exercices 79
4.14. Projets 82

Chapitre 5. Résolution d'équations non linéaires 85
5.1. Méthode de bissection ou de dichotomie 86
5.2. Méthode « Regula falsi » ou des parties proportionnelles 87
5.3. Méthode du point fixe ou d'itération 88
5.4. Méthode de Newton 90

5.5. Méthode de la sécante 93
5.6. Résolution de systèmes d'équations 93
5.7. Racines des polynômes 96
5.7.1. Division des polynômes 96
5.7.2. Séparation des racines 98
5.7.3. Suites de Sturm 99
5.7.4. La méthode de Newton pour les polynômes 101
5.7.5. Scilab et les polynômes 102
5.7.6. Condition du problème 103
5.8. Pour en savoir plus 104
5.9. Exercices 104

Chapitre 6. Résolution de systèmes d'équations linéaires 109
6.1. Le « conditionnement » 111
6.2. Orientation 112
6.3. Méthode de Gauss 113
6.3.1. Algorithme 113
6.3.2. Méthode de Gauss–Jordan 117
6.3.3. Décomposition ***LU*** 117
6.3.4. Représentation matricielle de l'élimination 119
6.3.5. Permutation de lignes 121
6.3.6. Nombre d'opérations 124
6.3.7. Calcul de l'inverse de A 125
6.4. Factorisation directe 125
6.4.1. Variantes 126
6.5. Matrices particulières 127
6.5.1. Matrice à diagonale dominante 127
6.5.2. Matrice symétrique définie positive 127
6.5.3. Matrice bande 128
6.5.4. Système tridiagonal 129
6.6. Méthodes itératives de résolution des systèmes linéaires 130
6.6.1. Méthode de Jacobi 130
6.6.2. Méthode de Gauss–Seidel 132

6.6.3. Méthode de surrelaxation 133
6.6.4. Convergence des méthodes itératives 134
6.7. Système surdéterminé 135
6.8. Pour en savoir plus 136
6.9. Exercices 137
6.10. Projet 139
6.11. Annexe : rappels d'algèbre linéaire 140
6.11.1. Base et sous-espace 140
6.11.2. Image, noyau et rang 140
6.11.3. Inverse et déterminant 141
6.11.4. Normes vectorielles 141
6.11.5. Normes de matrices 142
6.11.6. Opérations sur des blocs 143

Chapitre 7. Polynômes orthogonaux 145
7.1. Définition, existence 145
7.2. Relation avec les polynômes habituels 147
7.3. Propriétés des zéros 147
7.4. Relation de récurrence 148
7.5. Équation différentielle 149
7.6. Fonction génératrice 149
7.7. Formule de Rodrigues 150
7.8. Identité de Darboux–Christofel 150
7.9. Polynômes particuliers 150
7.9.1. Legendre 150
7.9.2. Hermite 151
7.9.3. Laguerre 152
7.9.4. Tschebychef 153
7.10. Autres polynômes classiques 154
7.10.1. Jacobi 154
7.10.2. Laguerre généralisé 154
7.11. Pour en savoir plus 154
7.12. Exercices 155

Chapitre 8. Dérivation et intégration numériques 159
8.1. Rappels d'analyse 159
8.2. Dérivée d'une fonction analytique 160
8.2.1. Développements limités 160
8.2.2. Méthode des coefficients indéterminés 162
8.2.3. Dérivée du polynôme d'interpolation 163
8.2.4. Accélération de la convergence 163
8.3. Dérivée d'une fonction empirique 164
8.4. Généralités sur l'intégration numérique 165
8.5. Méthodes élémentaires d'intégration 166
8.6. Méthodes de Newton–Cotes 168
8.6.1. Intervalle fermé 168
8.6.2. Intervalle ouvert 170
8.6.3. Formules composites 171
8.7. Méthode de Romberg 172
8.8. Intégration de Gauss 174
8.9. Généralisations de la méthode de Gauss 176
8.10. Les intégrales généralisées 177
8.11. Les intégrales multiples 177
8.12. L'intégrale sans peine 178
8.13. Pour en savoir plus 179
8.14. Exercices 179
8.15. Projet 184

Chapitre 9. Analyse spectrale, transformation de Fourier numérique 187
9.1. Les méthodes de Fourier 187
9.1.1. Série de Fourier 187
9.1.2. Intégrale ou transformée de Fourier (TF) 188
9.1.3. Vocabulaire et notations 189
9.1.4. Échantillonnage 189
9.1.5. Transformée de Fourier d'une fonction échantillonnée (TFTD) .. 190
9.2. Transformée de Fourier discrète (TFD) 191
9.2.1. Définition 191

9.2.2. La TFD comme approximation de l'intégrale de Fourier 191
9.2.3. Notation matricielle pour la TFD 192
9.3. Transformée de Fourier rapide (TFR) 193
9.3.1. Algorithme de Cooley–Tukey ou « entrelacement en temps » ... 194
9.3.2. Le renversement binaire 197
9.3.3. Factorisation de la matrice **V** et variantes de l'algorithme TFR . 197
9.4. Propriétés de la transformée de Fourier discrète 199
9.5. Pour en savoir plus 205
9.6. Exercices 205
9.7. Projet 207

Chapitre 10. Valeurs propres, vecteurs propres 209
10.1. Les éléments propres sans peine 210
10.2. Méthode de la puissance n-ième et méthodes dérivées 211
10.2.1. Puissance n-ième 211
10.2.2. Puissance n-ième avec décalage 213
10.2.3. Puissance n-ième de l'inverse 213
10.2.4. Puissance n-ième de l'inverse avec décalage 213
10.2.5. Quotient de Rayleigh 214
10.3. Méthode de Jacobi 215
10.3.1. Principe 215
10.3.2. Mise en œuvre 216
10.4. Transformation de Householder 221
10.5. Factorisation QR et algorithme QR 224
10.5.1. Factorisation QR 224
10.5.2. Algorithme QR 225
10.6. Réduction à la forme tridiagonale et calcul des valeurs propres 227
10.6.1. Tridiagonalisation 227
10.6.2. Calcul des valeurs propres 228
10.7. Matrices hermitiennes 230
10.8. Pour en savoir plus 231
10.9. Exercices 231
10.10. Projets 234

Chapitre 11. Problèmes différentiels à conditions initiales 239
11.1. Méthodes analytiques 241
11.1.1. Développement de Taylor 241
11.1.2. Méthode des coefficients indéterminés (Frobenius) 242
11.1.3. Méthode de Picard, ou d'approximations successives 242
11.2. Méthodes d'Euler et de Taylor 243
11.3. Méthodes de Runge–Kutta 246
11.3.1. Méthodes d'ordre 2 246
11.3.2. Méthode d'ordre d'ordre 4 247
11.3.3. Avantages et inconvénients des méthodes de Runge–Kutta 250
11.3.4. Organisation d'un programme 251
11.4. Ordre, stabilité et convergence des méthodes à un pas 251
11.5. Méthodes à pas multiples 255
11.5.1. Schémas explicites (ouverts) 255
11.5.2. Schémas implicites (fermés) 257
11.5.3. Méthodes de prédiction-correction 258
11.5.4. Surveillance de l'erreur 260
11.5.5. Formules d'ordre 4 260
11.5.6. Avantages et inconvénients des méthodes à pas multiples 261
11.6. Ordre, stabilité et convergence des méthodes multi-pas 261
11.7. Méthodes pour les équations du second ordre 262
11.7.1. Algorithme de Verlet ou de saute-mouton 262
11.7.2. Algorithme de Numerov 263
11.8. Équations « raides » 264
11.9. Résoudre une équation différentielle en dormant 265
11.10. Pour en savoir plus 269
11.11. Exercices 269
11.12. Projets 273

Chapitre 12. Problèmes à conditions aux limites et problèmes aux valeurs propres 277
12.1. La méthode du tir 279
12.1.1. Problème aux limites 279

12.1.2. Problèmes de valeurs propres 281
12.2. Méthodes des différences finies 282
12.2.1. Problème aux limites 282
12.2.2. Problème de valeurs propres 284
12.3. Les boîtes noires 287
12.4. Pour en savoir plus 288
12.5. Exercices 288
12.6. Projets 290

Chapitre 13. Équations aux dérivées partielles 297
13.1. Approximation des dérivées par des différences finies 297
13.2. Équations de Laplace et Poisson 298
13.3. Équation de la chaleur 300
13.4. Équation des ondes 303
13.5. Pour en savoir plus 304
13.6. Exercices 305
13.7. Projet 306

Chapitre 14. Probabilités et erreurs 309
14.1. Probabilité 309
14.2. Lois de probabilité 311
14.2.1. Loi binomiale 311
14.2.2. Loi de Poisson 312
14.2.3. Loi uniforme 313
14.2.4. Loi normale ou de Gauss 313
14.2.5. Loi du χ^2 ou de Pearson 314
14.2.6. Paramètres de la loi de probabilité et paramètres de l'échantillon 315
14.2.7. Vérification d'une loi de probabilité 316
14.3. Erreurs 317
14.4. Propagation des erreurs 320
14.5. Méthode du maximum de vraisemblance 321
14.6. Méthode des moindres carrés 323
14.6.1. Ajustement sur une fonction affine 324

14.6.2. Linéarisation 326
14.7. Qualité de l'ajustement 326
14.8. Coefficient de corrélation 329
14.9. Ajustement sur une fonction linéaire de plusieurs paramètres 329
14.10. Pour en savoir plus 331
14.11. Exercices 332

Chapitre 15. Méthodes de Monte Carlo 337
15.1. Générateurs de nombres aléatoires 338
15.1.1. Principe 338
15.1.2. Vérification d'un GNA 339
15.1.3. Validation d'un GNA à l'aide d'une marche aléatoire 340
15.2. Nombres aléatoires à répartition non-uniforme 341
15.2.1. Fonction d'une variable aléatoire 342
15.2.2. Méthode de la fonction réciproque ou du changement de variable 343
15.2.3. La méthode du rejet de von Neumann 344
15.2.4. La distribution normale 345
15.3. Simulation de phénomènes aléatoires 346
15.3.1. La radioactivité 347
15.3.2. L'agrégation 348
15.4. Méthodes de Monte Carlo déterministes : calcul d'intégrales 349
15.4.1. Calcul de π 349
15.4.2. Avantages et inconvénients des méthodes stochastiques pour le calcul d'intégrales 350
15.4.3. Intégrales par la méthode du rejet 351
15.4.4. Intégrales par la valeur moyenne 352
15.5. Pour en savoir plus 353
15.6. Exercices 354
15.7. Projet 356

Index 359

Table des matières 363

This page intentionally left blank