Christian GUILPIN

MANUEL DE CALCUL NUMÉRIQUE APPLIQUÉ

EDP SCIENCES

MANUEL DE CALCUL NUMÉRIQUE APPLIQUÉ

Christian Guilpin
Maître de Conférence, responsable de l'enseignement des mathématiques appliquées en maîtrise de Physique et Applications à l'université Paris VII-Denis Diderot

EDP
SCIENCES

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

ISBN : 2-86883-406-X

Avant-propos

Ce livre de calcul appliqué trouve son origine dans un cours dispensé aux étudiants de maîtrise de physique et applications (MPA) à l'université Paris VII depuis 1984, ainsi que dans quelques préoccupations de recherche au laboratoire.

Ce manuel ne constitue pas à proprement parler un cours au sens usuel du mot, et si certains chapitres ont des liens évidents et s'enchaînent naturellement, d'autres, plus isolés, ont été réunis sous forme d'annexes pour préserver au mieux l'unité, mais leur importance n'est pas secondaire.

L'organisation de cet ouvrage vise à une présentation suffisamment concise et assimilable des algorithmes numériques fondamentaux, développés jusqu'à leur mise en œuvre, de telle sorte qu'ils soient susceptibles d'aider l'étudiant, le chercheur et l'ingénieur dans l'exercice quotidien de leur art : il s'agit de pouvoir obtenir des résultats numériques convenables chaque fois qu'une méthode analytique fait défaut.

Pour ce qui concerne certains théorèmes très importants, nous nous sommes parfois borné à les énoncer sans les démontrer, le contraire eut risqué de nous éloigner de notre préoccupation majeure : le résultat numérique; cependant, les indications bibliographiques permettent d'obtenir aisément ces démonstrations qui sont classiques.

Les objectifs poursuivis se situent sur deux plans que l'on a coutume de séparer mais qui sont indissociables de notre point de vue : l'acquisition d'algorithmes numériques indispensable à la résolution de problèmes usuels et la maîtrise du traitement des données expérimentales selon la méthode statistique. Traiter les données de l'expérience impose l'usage de techniques numériques appropriées, et l'examen des résultats entachés d'erreur et d'incertitude impose l'usage de la statistique. La propagation des erreurs à travers les algorithmes relève d'une analyse subtile qui est éternellement omise tant elle est délicate. Nous avons tenté de l'effleurer et c'est une des raisons qui nous a poussé à développer l'étude des lois de distribution ainsi que leurs fondements dans une partie qui est davantage dévolue aux statistiques.

De même qu'il est impensable de vouloir apprendre à jouer du piano la veille de donner un concert, de même il est impensable de vouloir apprendre l'algorithmique numérique le jour où le besoin s'impose. Dans les deux cas, il convient de recourir aux gammes afin d'acquérir une solide expérience. En calcul numérique il n'y a pas de voie royale, et aucun algorithme n'est capable de fournir de résultats corrects quelles que soient les données fournies. Il est toujours possible de mettre en défaut une procédure et d'obtenir des résultats abérrants pourvu que l'on s'en donne la peine... Un très bel exemple est étudié à l'occasion de la résolution des systèmes linéaires dépendant d'une matrice de Hilbert.

L'expérience pratique prend alors toute sa valeur, et c'est ainsi que notre enseignement comporte une séance hebdomadaire de trois heures sur calculateur arithmétique. Peu importe le langage et la manière, seul le « résultat correct » compte et ce n'est pas une mince affaire que de se faire une opinion sur les erreurs qui entachent les résultats finals. Ensuite viendront éventuellement se greffer les problèmes d'élégance et d'optimisation.

En aucun cas, cet enseignement n'a pour but d'explorer et de recenser tous les algorithmes ayant trait à un type de problèmes. Nous avons voulu présenter ceux qui se sont montrés paradigmatiques soit sous l'angle de la simplicité soit sous l'angle de l'efficacité. Il s'agit de construire des programmes que nous aurons soigneusement testés, dont nous connaîtrons les limites et qui rempliront peu à peu notre boîte à outils.

Pour simplifier, nous dirons que ce livre peut se subdiviser en trois parties à savoir :

1. Études d'algorithmes numériques et leur mise en oeuvre.
2. Analyse statistique des résultats d'expériences.
3. Annexes, problèmes et corrigés.

Nous l'avons déjà dit, les deux premières parties interfèrent partiellement, et c'est une des raisons pour laquelle nous avons renoncé à présenter un ouvrage où tout ce qui est étudié dans un chapitre s'appuie nécessairement sur ce qui a été établi précédemment. Par souci d'unité nous avons préféré regrouper les titres par centre d'intérêt. Ainsi, il nous est apparu plus intéressant d'avoir rassemblé l'étude des polynômes orthogonaux plutôt que d'avoir dispersé l'information dans différents chapitres concernant l'interpolation et l'intégration numérique.

Il aurait été dommage de ne pas avoir abordé, ne serait-ce que rapidement, les méthodes de Monte-Carlo d'une part, et les problèmes mal posés d'autre part. Ces domaines illustrent bien la synthèse des deux premières parties, d'autant plus qu'ils s'intègrent remarquablement dans les préoccupations des chercheurs et des ingénieurs. Qui, en physique, n'a pas eu à résoudre numériquement une équation de convolution? Qui n'a pas tenté la résolution d'un problème au moyen d'une simulation?

Pour terminer nous proposons un avant-dernier chapitre constitué d'un ensemble de problèmes et d'exercices qui illustrent quelques usages des méthodes qui ont été présentées ; ils servent également à éclairer quelques points de théorie qui seraient venus alourdir le cours s'ils avaient été intégrés dans les divers chapitres : on montre par exemple que le coefficient de conformité de Pearson obéit bien à une loi du χ^2. Le dernier chapitre donne les solutions des problèmes présentés.

La plupart des chapitres font l'objet d'une illustration et se terminent par des programmes écrits dans le langage C : il s'agit du langage de base qui assure la portabilité. Ce point de vue s'explique par la facilité qu'il y a à changer de langage : Fortran, Pascal, etc., sans avoir grand chose à modifier dans le programme source. On n'est pas obligé de partager ces vues, mais il est très facile de modifier les programmes proposés pour qu'ils apparaissent moins « archaïques ».

Pour en finir avec les algorithmes choisis et les programmes présentés, nous dirons qu'ils sont fournis **sans garantie** d'aucune sorte malgré le grand soin porté à ce travail. Ils peuvent comporter des imprécisions voire des imperfections, à ceci s'ajoute le fait qu'aucun algorithme n'est irréprochable dans la mesure où il est toujours possible de trouver des valeurs numériques qui le mette en défaut.

Bien sûr, nous formons le vœu que cet ouvrage puisse apporter une aide solide aux étudiants, ingénieurs et chercheurs pour lesquels il constituera un outil dont le rôle favorise la réalisation de sa propre boîte à outils.

La rédaction d'un ouvrage ne se réalise jamais dans l'isolement, et il m'a fallu bien des oreilles attentives, bien des lecteurs vigilants, bien des conseillers éclairés. L'instant est venu de remercier tous ceux qui, à quelque titre que ce soit, m'ont apporté une aide inconditionnelle, je citerai par ordre alphabétique : Claude Bardos, Jean Bornarel, Jacques Gacougnolle, Patricia Guilpin,

Michel Jacques, Claude Marti, Yvan Simon ainsi que l'équipe de physique théorique de Chaouqui Misbah.

Pour terminer, j'ajouterai une mention particulière à EDP Sciences qui m'a offert un contexte de travail optimum afin d'obtenir la meilleure réalisation possible.

Christian GUILPIN

PROGRAMMES SOURCES ACCOMPAGNANT CE MANUEL

Les programmes sources cités dans ce livre sont disponibles sur le site Web d'EDP Sciences (adresse : `http://www.edpsciences.com/guilpin/`).

Chapitre 2
aitken_0.c
epsilon_0.c
richard.h
richar_0.c
epsilon_1.c
epsilon_2.c
epsilon_3.c
epsilon_4.c
aitken_2.c
kacmarz1.c
fredholm.c
newton_1.c

Chapitre 4
dichot0.c
itera.c
newton1d.c
newtonpp.c
kacmarz.c
newton2d.c
bairstow.c
bairstow.h
racine.h

Chapitre 5
systlin.c
triangle.c
trianlin.c
hilbert.c
trianinv.c
leverier.c
givens.c
rutisacc.c
danilev.c
dsyslin.h

Chapitre 6
lagpoly.c
ascend.c
descend.c
lispline.c
spline.h
sudeter.c
multma.h
transpos.h
dsyslin.h
invers.h

Chapitre 7
legendre.c
decomleg.c
r_legend.c
getname.h

Chapitre 9
laguerre.c
r_laguer.c

Chapitre 10
hermite.c
r_hermit.c

Chapitre 11
rn_x.txt
sn_x.txt
un_x.txt
vn_x.txt

Chapitre 12
argent.c
cotes_1.c
cotes_2.c

Chapitre 13
epsilon.h
combina.h
retrogra.c
runge.c
pendule.c
moulton.c
bashfort.c
taylor.c

Chapitre 14
triode.c
chaleur.c
corde.c

Chapitre 15
echantil.c
gibbs.c
dirac.c
filtre.c

Chapitre 16
dfft0.h
dfftinv.h
tf_image.c
inv_imag.c

Chapitre 17
fredh_1.c
gaussien.h
dconvol.c
dconvol.h
intmonte.h
xaleamen.h

Chapitre 18
calculpi.c
matmonte.c
intmonte.c
recuit.c

Chapitre 20
gaussleg.h

Chapitre 22
khi2.h
student.h

Chapitre 24
histogra.c
kolmogor.c

Chapitre 25
teststat.c
varian_1.c
varian_2.c

Chapitre 26
regres.c

Annexe A
sturm.c

Annexe F
bessel1n.c
bessel1f.c
bessel2n.c
bessel2f.c
besseljf.h
besseljn.h
bessel2f.h

Annexe I
dzeta0.c
dzeta1.c
grosys1.c
jacobi0.c
jacobi1.c
pendule0.c
predcor0.c
souriau0.c
sudeter.c
syslinit.c
tangent2.c
gradconj.c
refrig1.c
mathieu9.c
vanderp0.c
card0.c

Sommaire

3. LES DÉVELOPPEMENTS ASYMPTOTIQUES — 43

4. RÉSOLUTION DES ÉQUATIONS NUMÉRIQUES — 51

5. ÉLÉMENTS DE CALCUL MATRICIEL — 69

6. L'INTERPOLATION — 89

7. LES POLYNÔMES DE LEGENDRE. MÉTHODE D'INTÉGRATION DE GAUSS-LEGENDRE — 113

8. LES POLYNÔMES DE TCHEBYCHEFF. APPLICATION À LA MÉTHODE DE GAUSS-TCHEBYCHEFF — 133

9. LES POLYNÔMES DE LAGUERRE. MÉTHODE D'INTÉGRATION DE GAUSS-LAGUERRE — 141

10. LES POLYNÔMES D'HERMITE. LA MÉTHODE D'INTÉGRATION DE GAUSS-HERMITE — 153

15. LES SÉRIES DE FOURIER 229

16. LES TRANSFORMÉES DE FOURIER 249

17. INITIATION AUX PROBLÈMES MAL POSÉS : ÉQUATIONS INTÉGRALES, SYSTÈMES LINÉAIRES MAL CONDITIONNÉS ET ÉQUATIONS DE CONVOLUTION 273

B. POLYNÔMES ORTHOGONAUX RELATIVEMENT À UNE FONCTION POIDS. GÉNÉRALISATION DE LA MÉTHODE DE GAUSS 389

C. LES FRACTIONS CONTINUES 397

D. LES APPROXIMANTS DE PADÉ ET DE MAEHLY 405

E. CALCUL DES FONCTIONS DE BIBLIOTHÈQUE ÉLÉMENTAIRES 421

F. CALCUL NUMÉRIQUE DES FONCTIONS DE BESSEL 429

G. ÉLÉMENTS SUCCINCTS SUR LE TRAITEMENT DU SIGNAL 433

H. PROBLÈMES ET EXERCICES 443

I. CORRIGÉS DES PROBLÈMES ET EXERCICES 497

INDEX 567

1 Généralités sur le calcul numérique

Le calcul numérique est une branche des mathématiques appliquées qui étudie les méthodes pratiques destinées à fournir des solutions numériques aux problèmes formalisés dans le langage des mathématiques pures. La plupart du temps, ce sont les ingénieurs et les chercheurs qui se trouvent concernés par l'usage de ces méthodes pratiques, car ce sont en définitive les nombres qui vont retenir leur attention. Face à un jeu d'équations plus ou moins complexes, ils devront obtenir un ensemble de valeurs numériques qui pourra servir par exemple soit à la réalisation d'un édifice ou d'un prototype, soit à la confrontation des résultats expérimentaux et théoriques etc.

1. La notion d'algorithme en calcul numérique

Un algorithme est un procédé de calcul qui ne met en œuvre que des opérations arithmétiques et logiques. L'étymologie de ce mot est arabe et c'est une altération du nom du mathématicien Al-Khwârizmi (†812?), probablement sous l'influence du mot grec (repris par les latins) ό αριθμός, le nombre.

Quoi qu'il en soit, c'est une dénomination commode qui sert à désigner l'ensemble des opérations qui interviennent au cours d'une démonstration conduisant à l'énoncé d'un théorème. Cependant sa portée ne dépasse pas celle de la « prose que chacun fait sans le savoir », et en aucun cas cette définition ne permet de donner une manière de construction des algorithmes. Il s'agit donc d'un concept commode mais peu fécond qui sert à désigner un certain type d'organisation de propositions à caractère mathématique. Sans que rien ne soit changé, il est tout à fait possible de remplacer ce mot par procédé de calcul, technique de calcul, procédure...

Pour illustrer ce concept, on peut évoquer, par exemple, la technique de résolution des équations du deuxième degré à coefficients réels à condition toutefois d'admettre que l'on dispose outre les quatre opérations fondamentales (addition, soustraction, multiplication et division) de l'opération racine carrée. L'examen des mathématiques montre que le nombre d'opérations proposées dans un algorithme peut être infini (mais dénombrable), c'est le cas des développements en série de fonctions : série entière, série de Fourier, etc.

Le concept d'algorithme en calcul numérique est quelque peu plus restrictif : le nombre d'opérations élémentaires arithmétiques et logiques est obligatoirement fini. En outre, cela implique que l'on ne peut manipuler que des nombres admettant une représentation finie ce qui conduit à effectuer des troncatures et des arrondis au cours des opérations successives. Cette remarque en apparence triviale doit pourtant être présente à l'esprit lorsque l'on fait usage d'une machine arithmétique (mais aussi du calcul manuel...) : la première division venue, la première racine carrée venue ont toutes les chances d'introduire un nombre dont la représentation impose une infinité de chiffres significatifs, il faudra les tronquer...

La connaissance de la manière dont les nombres sont représentés dans la machine utilisée est fondamentale en ce sens qu'elle permet l'estimation de la précision attachée aux résultats d'un calcul. Dès lors, on comprend très bien que la recherche d'une grande précision imposera une grande taille de mots-machine et par conséquent un grand temps de calcul. En revanche, une faible précision n'imposera qu'une petite taille du mot-machine ainsi qu'un faible temps de calcul. Le choix retenu est en général un compromis entre ces deux situations extrêmes.

Cette dernière étape intervient lors de l'évaluation de l'incertitude qui entache les résultats d'un calcul, mais ce n'est pas la seule source. Le problème général de l'évaluation des incertitudes et erreurs est délicat qui plus est lorsque l'on fait usage d'une machine ; nous l'aborderons un peu plus en détail dans quelques paragraphes.

2. Le calcul numérique ne concerne que les nombres entiers

Ce n'est pas une boutade, et il faut bien admettre que l'on ne peut manipuler (au sens étymologique) que des représentations finies. La représentation dite en virgule flottante éclaire ce point de vue car en fait elle traite de nombres ayant une certaine quantité fixe de chiffres significatifs (agrémenté d'un facteur de cadrage appelé exposant) et dans la réalité, à chaque opération élémentaire sur les nombres il y a une opération de cadrage suivie d'une opération arithmétique sur les nombres entiers, elle-même suivie éventuellement d'une opération de troncature ou d'arrondi.

Cette conception n'est pas tellement restrictive en soi, car les seules opérations arithmétiques pratiques que l'Homme est susceptible de réaliser ne peuvent que concerner les nombres admettant une représentation finie de symboles (chiffres significatifs) donc en fait des entiers. Pour ce qui concerne les machines arithmétiques, les deux représentations, entière et flottante, ne doivent pas masquer la réalité et il faut voir là uniquement deux représentations commodes. En fin de chapitre, on trouvera un paragraphe traitant de quelques représentations assez générales.

3. Le calcul numérique traite du problème pratique de l'approximation de fonctions explicites ou implicites

Un problème d'analyse se trouvant modélisé dans le langage des mathématiques pures, la première préoccupation consiste à rechercher la solution sous forme littérale à l'aide des fonctions connues qui sont les polynômes et les fonctions transcendantes élémentaires. On peut se poser la question de savoir si cette façon de concevoir est toujours possible.

Tous les problèmes ne sont pas algorithmiquement solubles, et force est de répondre non à la question posée. Pour s'en convaincre, il suffit de considérer un exemple, sous l'angle de l'histoire, qui illustre cette difficulté : la recherche des racines d'un polynôme de degré quelconque à coefficients réels.

Si l'équation du deuxième degré est connue depuis l'Antiquité, on peut dire qu'il revient à Al-Khwârizmi (fin VIIIe, début IXe) et à Luca Pacioli (1445?–1514?) le fait d'avoir raffiné les solutions.

L'école italienne de Bologne s'attaque à l'équation du troisième degré ; on retiendra à son propos les noms de Tartaglia (1500?–1557), de Cardan (1501–1576) qui parvinrent à la solution au travers de défis et de provocations qui semblaient être coutumiers à cette époque. Signalons toutefois que les bases de l'étude sont dues à Del Fero (1465?–1526).

Ensuite l'équation du quatrième degré fut abordée et résolue par Ferrari (1522–1565) qui fut un élève de Cardan. Plus tard, les mathématiciens s'intéressèrent tout naturellement à l'équation du cinquième degré et les recherches furent très riches d'enseignements dans la mesure où, selon toute vraisemblance pour la première fois dans l'histoire des sciences, le travail conduisit à l'énoncé d'un résultat négatif.

1. En 1608, Rothe (?–1617) émit une proposition selon laquelle toute équation algébrique de degré n possédait n racines réelles ou complexes, et Gauss (1777–1855) en effectua la démonstration rigoureuse en 1799.
2. Cette même année Ruffini (1765–1822) affirma l'impossibilité formelle d'obtenir la résolution des équations algébriques de degré supérieur à quatre. Il fallut attendre la publication des travaux d'Abel (1802–1829) pour obtenir la démonstration définitive de la proposition de Ruffini (1826).

Ce résumé de l'histoire des équations algébriques nous mène à deux types de remarques :

Remarque 1 **:** Le fait de poser un problème dont la solution existe n'est pas suffisant pour permettre d'exhiber effectivement ladite solution selon des moyens donnés. On peut se rappeler à ce propos le problème de la trisection de l'angle qui n'est pas algorithmiquement soluble si les moyens de construction autorisés sont la règle et le compas.

Aujourd'hui, on sait que les problèmes que l'on peut se poser sont de trois types à savoir :

a. les problèmes algorithmiquement solubles, l'équation du deuxième degré par exemple ;
b. les problèmes algorithmiquement non solubles, par exemple la quadrature du cercle à l'aide de la règle et du compas ;
c. les problèmes indécidables pour lesquels on ne peut rien démontrer. Du reste, dans une axiomatique donnée, on sait qu'il existe des propositions vraies que l'on ne peut pas démontrer, à condition de considérer toutefois que les axiomes de l'arithmétique ne sont pas contradictoires (*cf.* Gödel (1906–1978)).

Remarque 2 **:** Ce n'est pas parce qu'un problème est algorithmiquement insoluble que l'on n'est pas en mesure de proposer une solution approchée — généralement avec une précision fixée à l'avance — et c'est justement la raison d'être du calcul numérique que de fournir de telles solutions ; par exemple, on verra comment calculer les racines d'un polynôme de degré quelconque à coefficients réels.

En terminant ce paragraphe, nous remarquerons que les problèmes de calcul numérique sont uniquement des problèmes d'approximation de fonctions au sens le plus large. Souvenons-nous d'une des premières équations différentielles que nous avons rencontrée en Physique, il s'agit de l'équation du pendule pesant assujetti à osciller dans un plan. On écrit traditionnellement l'équation à laquelle obéit son mouvement :

$$\frac{\mathrm{d}^2\theta(t)}{\mathrm{d}t^2} + \omega^2 \sin\theta(t) = 0,$$

où $\theta(t)$ est l'angle que le pendule fait avec la verticale à un instant donné, ω étant une constante dépendant de la longueur du pendule et de l'accélération. Alors, on apprend que l'on ne peut pas obtenir la solution sous forme littérale $(t,\, \theta_0,\, \theta_0',\, \omega^2)$ à l'aide des transcendances usuelles (θ_0 et θ_0' sont les conditions initiales sur l'angle et la vitesse angulaire). Pourtant, le théorème de Cauchy-Lipschitz nous permet d'affirmer l'existence et l'unicité d'une telle solution qui est de surcroît une fonction analytique.

La tradition veut également que l'on ne s'intéresse qu'au cas des petits angles pour lesquels on peut écrire le développement de $\sin\theta(t)$ au premier ordre. Alors, dans ce cas particulier, on sait