

清华大学

计算机系列教材

喻文健 编著

# 数值分析与算法

(第2版)



清华大学出版社

清华大学 计算机系列教材

喻文健 编著

# 数值分析与算法

(第2版)

清华大学出版社  
北京

## 内 容 简 介

本书是针对“数值分析”、“计算方法”、“数值分析与算法”等课程编写的教材,主要面向理工科大学信息与科学技术各专业,以及信息与计算科学专业的本科生。本书内容包括数值计算基础、非线性方程的数值解法、线性方程组的直接解法与迭代解法、矩阵特征值与特征向量的计算、数值逼近与插值、数值积分方法、常微分方程初值问题的解法,以及数值算法与应用的知识。本书涵盖数值分析、矩阵计算领域最基本、最常用的一些知识与方法,而且在算法及应用方面增加了一些较新的内容。在叙述上既注重理论的严谨性,又强调方法的应用背景、算法设计,以及不同方法的对比。为了增加实用性与可扩展性,每章配备了应用实例、算法背后的历史、评述等子栏目,书末附有算法、术语索引。在附录中还包括 MATLAB 软件的简介,便于读者快速掌握并进行编程实验。

本书适合作为高年级本科生或研究生的教材,也可供从事科学与工程计算的科研人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

数值分析与算法/喻文健编著. —2版. —北京:清华大学出版社,2015

清华大学计算机系列教材

ISBN 978-7-302-40982-3

I. ①数… II. ①喻… III. ①数值计算—高等学校—教材 IV. ①O241

中国版本图书馆 CIP 数据核字(2015)第 166771 号

责任编辑:白立军

封面设计:常雪影

责任校对:白蕾

责任印制:何羊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:22.75

字 数:553千字

版 次:2012年1月第1版 2015年9月第2版

印 次:2015年9月第1次印刷

印 数:1~2000

定 价:39.50元

产品编号:061720-01

## 序

“清华大学计算机系列教材”已经出版发行了 30 余种,包括计算机科学与技术专业的基础数学、专业技术基础和专业等课程的教材,覆盖了计算机科学与技术专业本科生和研究生的主要教学内容。这是一批至今发行数量很大并赢得广大读者赞誉的书籍,是近年来出版的大学计算机专业教材中影响比较大的一批精品。

本系列教材的作者都是我熟悉的教授与同事,他们长期在第一线担任相关课程的教学工作,是一批很受本科生和研究生欢迎的任课教师。编写高质量的计算机专业本科生(和研究生)教材,不仅需要作者具备丰富的教学经验和科研实践,还需要对相关领域科技发展前沿的正确把握和了解。正因为本系列教材的作者们具备了这些条件,才有了这批高质量优秀教材的产生。可以说,教材是他们长期辛勤工作的结晶。本系列教材出版发行以来,从其发行的数量、读者的反映、已经获得的国家级与省部级的奖励,以及在各个高等院校教学中所发挥的作用上,都可以看出本系列教材所产生的社会影响与效益。

计算机学科发展异常迅速,内容更新很快。作为教材,一方面要反映本领域基础性、普遍性的知识,保持内容的相对稳定性;另一方面,又需要紧跟科技的发展,及时地调整和更新内容。本系列教材都能按照自身的需要及时地做到这一点。如王爱英教授等编著的《计算机组成与结构》、戴梅萼教授等编著的《微型计算机技术及应用》都已经出版了第四版,严蔚敏教授的《数据结构》也出版了三版,使教材既保持了稳定性,又达到了先进性的要求。

本系列教材内容丰富,体系结构严谨,概念清晰,易学易懂,符合学生的认知规律,适合教学与自学,深受广大读者的欢迎。系列教材中多数配有丰富的习题集、习题解答、上机及实验指导和电子教案,便于学生理论联系实际地学习相关课程。

随着我国进一步的开放,我们需要扩大国际交流,加强学习国外的先进经验。在大学教材建设上,我们也应该注意学习和引进国外的先进教材。但是,“清华大学计算机系列教材”的出版发行实践以及它所取得的效果告诉我们,在当前形势下,编写符合国情的具有自主版权的高质量教材仍具有重大意义和价值。它与国外原版教材不仅不矛盾,而且是相辅相成的。本系列教材的出版还表明,针对某一学科培养的要求,在教育部等上级部门的指导下,有计划地组织任课教师编写系列教材,还能促进对该学科科学、合理的教学体系和内容的研究。

我希望今后有更多、更好的我国优秀教材出版。

清华大学计算机系教授,中国科学院院士

张钹

## 第2版前言

“数值分析”或“计算方法”是理工科大学各专业普遍开设的一门课程,其内容主要包括有关数值计算(numerical computing)的理论与方法。数值计算是计算数学、计算机科学与其他工程学科相结合的产物,随着计算技术的发展与普及,它正变得越来越重要。

本书的主要内容与一般的“数值分析”教材基本一致,但还具有如下特点。

(1) 对数学理论的介绍简明扼要。尽量用形象的方式解释数学中的一些概念与理论,通过定理总结重要的结论。在不失严谨性的前提下,省略部分定理的证明,取而代之的是进行直观的解释、验证,并说明其意义与用途。

(2) 强调算法的实际应用与分析比较。对大多数算法,采用程序伪码的形式加以描述,同时分析其计算复杂度。说明算法应用中的细节问题,对几个较新的算法还给出了 MATLAB 源程序。通过“应用实例”和相关 MATLAB 命令,更详细地介绍算法的应用。

(3) 具有较强的可读性与实用性。尽量用图、表等形象的方式对概念、现象进行解释。每章编写了“算法背后的历史”子栏目,增强阅读的趣味性。书末附有算法、术语索引,便于查阅。为了便于读者动手实践,对 MATLAB 软件的相关功能做了介绍。

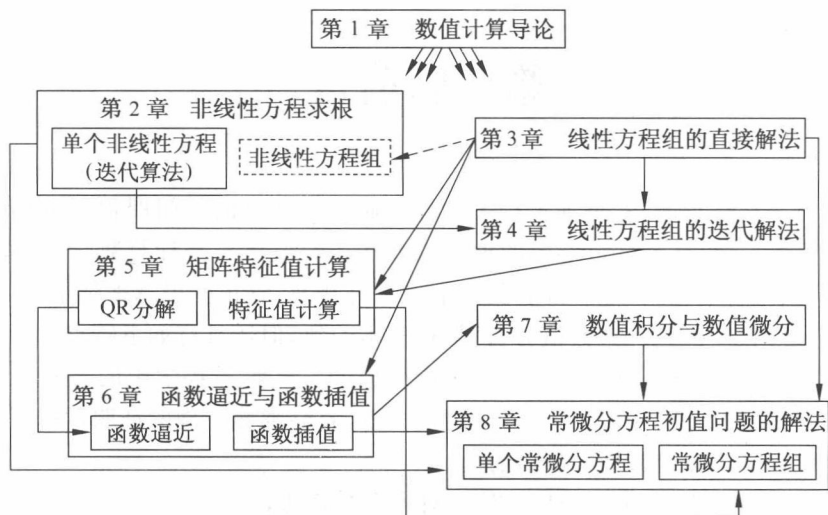
(4) 在内容编排上有利于教学。依据教学规律安排各章的顺序;在每章的“评述”部分列出主要知识点,除练习题外还提供了上机实验题,附录中给出了部分习题的答案。

学习数值分析与算法,应重视通过计算机编程加深理解相关理论与算法。本书提倡使用 MATLAB 软件来进行编程实验,基于以下理由:① MATLAB 编程语言易于学习、代码简洁,可节省编程实验时间。② MATLAB 是功能强大的科学计算集成环境,便于程序调试和形象直观地展示程序运行结果。③ MATLAB 具有丰富、先进的数值计算能力,已被广泛用于科学与工程实践中,掌握 MATLAB 中使用的技术可作为课程学习的扩展与提高。

本书第1版于2012年出版后,即作为“数值分析”课的教材投入使用,收到了较满意的效果。但通过教学实践也发现了书中的一些问题与纰漏之处,有必要进行更正与修订。本书第2版对第1版中幂法使用条件、对角占优矩阵 LU 分解稳定性等处进行了更正,修改了第1章部分定理的证明,更新了第2章和第7章,以及附录 B 有关 MATLAB 软件的内容,共计修订文字、图、表两百余处。在排版方面也做了一些改进,力求呈现出更高的品质。

本书体现了作者过去十年的教学工作积累,参考、借鉴了十几种较新的国内外优秀教材,力争在理论与实践相结合、反映学科发展前沿,以及适应时代发展对学生培养的新要求等方面取得好的效果。本书内容由误差分析、非线性方程求根、数值线性代数、函数插值、数值积分、常微分方程数值解法等部分组成,包括了数值计算领域中最经典、应用最广泛的一些内容,它们也为学习数学规划、大数据分析、机器学习等较新领域中的一些高级算法提供基础。使用本教材时,可用 48 学时讲授主要的内容,几乎每章也包含一些简介性质或与 MATLAB 软件有关的内容,供感兴趣的学生选学或课后阅读。

下图显示了各章主要内容的知识依赖关系。总体上,建议教师按照从第1章到第8章的顺序开展教学,只是第2.7节依赖于线性方程组的有关知识,需在第3章讲完后介绍。



白如冰、朱臻垚参加了本书第1版部分内容的编写,选修作者讲授的“数值分析”课的广大同学指出了第1版中的很多错误,提供了积极反馈,在此致以诚挚的谢意!此外,还要感谢清华大学王泽毅、殷人昆、边计年、蔡懿慈等教授给予的指导与帮助,以及清华大学出版社的编辑在出版本书过程中付出的辛勤劳动。

据不完全统计,本书已被20多所大学选作教材,使用的专业包括计算机专业、软件工程专业、电子信息专业、自动化专业等,在清华大学使用本教材的也包括物理、经管、工业工程,以及其他一些工科专业的学生。在此,作者对广大读者的支持表示诚挚的感谢!也希望广大读者提出宝贵的意见与建议。

喻文健  
2015年7月



# 目 录

第 1 章 数值计算导论	1
1.1 概述	1
1.1.1 数值计算与数值算法	1
1.1.2 数值计算的问题与策略	2
1.1.3 数值计算软件	4
1.2 误差分析基础	6
1.2.1 数值计算的近似	6
1.2.2 误差及其分类	7
1.2.3 问题的敏感性与数据传递误差估算	11
1.2.4 算法的稳定性	13
1.3 计算机浮点数系统与舍入误差	15
1.3.1 计算机浮点数系统	15
1.3.2 舍入与机器精度	18
1.3.3 浮点运算的舍入误差	19
1.3.4 抵消现象	21
1.4 保证数值计算的准确性	22
1.4.1 减少舍入误差的几条建议	22
1.4.2 影响结果准确性的主要因素	24
评注	25
算法背后的历史: 浮点运算的先驱——威廉·卡亨	26
练习题	28
上机题	29
第 2 章 非线性方程求根	30
2.1 引言	30
2.1.1 非线性方程的解	30
2.1.2 问题的敏感性	31
2.2 二分法	31
2.2.1 方法原理	31
2.2.2 算法稳定性和结果准确度	33
2.3 不动点迭代法	35
2.3.1 基本原理	35
2.3.2 全局收敛的充分条件	36
2.3.3 局部收敛性	38
2.3.4 稳定性与收敛阶	38

2.4	牛顿迭代法	40
2.4.1	方法原理	40
2.4.2	重根的情况	42
2.4.3	判停准则	43
2.4.4	牛顿法的问题	43
2.5	割线法与抛物线法	44
2.5.1	割线法	44
2.5.2	抛物线法	46
2.6	实用的方程求根技术	46
2.6.1	阻尼牛顿法	46
2.6.2	多项式方程求根	47
2.6.3	通用求根算法 zeroin	48
	应用实例:城市水管应埋于地下多深?	50
2.7	非线性方程组和有关数值软件	52
2.7.1	非线性方程组	52
2.7.2	非线性方程求根的相关软件	54
	评述	55
	算法背后的历史:牛顿与牛顿法	56
	练习题	57
	上机题	58
<b>第3章</b>	<b>线性方程组的直接解法</b>	<b>59</b>
3.1	基本概念与问题的敏感性	59
3.1.1	线性代数中的有关概念	59
3.1.2	向量范数与矩阵范数	62
3.1.3	问题的敏感性与矩阵条件数	65
3.2	高斯消去法	69
3.2.1	基本的高斯消去法	69
3.2.2	高斯-约当消去法	72
3.3	矩阵的LU分解	75
3.3.1	高斯消去过程的矩阵形式	75
3.3.2	矩阵的直接LU分解算法	79
3.3.3	LU分解的用途	82
3.4	选主元技术与算法稳定性	83
3.4.1	为什么要选主元	83
3.4.2	使用部分主元技术的LU分解	85
3.4.3	其他选主元技术	89
3.4.4	算法的稳定性	90
3.5	对称正定矩阵与带状矩阵的解法	91
3.5.1	对称正定矩阵的Cholesky分解	91



3.5.2 带状线性方程组的解法 .....	95
应用实例: 稳态电路的求解 .....	97
3.6 有关稀疏线性方程组的实用技术 .....	99
3.6.1 稀疏矩阵基本概念 .....	99
3.6.2 MATLAB 中的相关功能 .....	102
3.7 有关数值软件 .....	104
评述 .....	106
算法背后的历史: 威尔金森与数值分析 .....	107
练习题 .....	108
上机题 .....	110
<b>第 4 章 线性方程组的迭代解法</b> .....	<b>112</b>
4.1 迭代解法的基本理论 .....	112
4.1.1 基本概念 .....	112
4.1.2 1 阶定常迭代法的收敛性 .....	113
4.1.3 收敛阶与收敛速度 .....	116
4.2 经典迭代法 .....	118
4.2.1 雅可比迭代法 .....	118
4.2.2 高斯-赛德尔迭代法 .....	119
4.2.3 逐次超松弛迭代法 .....	121
4.2.4 三种迭代法的收敛条件 .....	123
应用实例: 桁架结构的应力分析 .....	126
4.3 共轭梯度法简介 .....	128
4.3.1 最速下降法 .....	128
4.3.2 共轭梯度法 .....	131
4.4 各种方法的比较 .....	135
4.4.1 迭代法之间的比较 .....	135
4.4.2 直接法与迭代法的对比 .....	138
4.5 有关数值软件 .....	139
评述 .....	140
算法背后的历史: 雅可比 .....	142
练习题 .....	143
上机题 .....	144
<b>第 5 章 矩阵特征值计算</b> .....	<b>146</b>
5.1 基本概念与特征值分布 .....	146
5.1.1 基本概念与性质 .....	146
5.1.2 特征值分布范围的估计 .....	150
5.2 幂法与反幂法 .....	152
5.2.1 幂法 .....	152
5.2.2 加速收敛的方法 .....	156

5.2.3	反幂法	158
	应用实例: Google 的 PageRank 算法	160
5.3	矩阵的正交三角化	162
5.3.1	Householder 变换	163
5.3.2	Givens 旋转变换	165
5.3.3	矩阵的 QR 分解	166
5.4	所有特征值的计算与 QR 算法	170
5.4.1	收缩技术	170
5.4.2	基本 QR 算法	171
5.4.3	实用 QR 算法的有关技术	173
5.5	有关数值软件	177
	评述	178
	算法背后的历史: A. Householder 与矩阵分解	179
	练习题	180
	上机题	183
<b>第 6 章</b>	<b>函数逼近与函数插值</b>	<b>185</b>
6.1	函数逼近的基本概念	185
6.1.1	函数空间	185
6.1.2	函数逼近的不同类型	188
6.2	连续函数的最佳平方逼近	190
6.2.1	一般的法方程方法	190
6.2.2	用正交函数族进行逼近	194
6.3	曲线拟合的最小二乘法	197
6.3.1	问题的矩阵形式与法方程法	198
6.3.2	用正交化方法求解最小二乘问题	202
	应用实例: 原子弹爆炸的能量估计	206
6.4	函数插值与拉格朗日插值法	207
6.4.1	插值的基本概念	207
6.4.2	拉格朗日插值法	208
6.4.3	多项式插值的误差估计	211
6.5	牛顿插值法	213
6.5.1	基本思想	213
6.5.2	差商与牛顿插值公式	214
6.6	分段多项式插值	219
6.6.1	高次多项式插值的病态性质	219
6.6.2	分段线性插值	220
6.6.3	分段埃尔米特插值	221
6.6.4	保形分段插值	224
6.7	样条插值函数	226

6.7.1	三次样条插值	226
6.7.2	三次样条插值函数的构造	227
6.7.3	B-样条函数	229
	评述	232
	算法背后的历史：拉格朗日与插值法	233
	练习题	234
	上机题	236
<b>第7章</b>	<b>数值积分与数值微分</b>	<b>238</b>
7.1	数值积分概论	238
7.1.1	基本思想	238
7.1.2	求积公式的积分余项与代数精度	240
7.1.3	求积公式的收敛性与稳定性	241
7.2	牛顿-柯特斯公式	242
7.2.1	柯特斯系数与几个低阶公式	242
7.2.2	牛顿-柯特斯公式的代数精度	244
7.2.3	几个低阶公式的余项	245
7.3	复合求积公式	246
7.3.1	复合梯形公式	246
7.3.2	复合辛普森公式	247
7.3.3	步长折半的复合求积公式计算	249
7.4	Romberg 积分算法	250
7.4.1	复合梯形公式的余项展开式	250
7.4.2	理查森外推法	251
7.4.3	Romberg 算法	252
7.5	自适应积分算法	254
7.5.1	自适应积分的原理	255
7.5.2	一个具体的自适应积分算法	255
7.6	高斯求积公式	258
7.6.1	一般理论	258
7.6.2	高斯-勒让德积分公式及其他	261
	应用实例：探月卫星轨道长度计算	263
7.7	数值微分	264
7.7.1	基本的有限差分公式	265
7.7.2	插值型求导公式	266
7.7.3	数值微分的外推算法	268
	评述	269
	算法背后的历史：“数学王子”高斯	271
	练习题	272
	上机题	273

第8章 常微分方程初值问题的解法	275
8.1 引言	275
8.1.1 问题分类与可解性	275
8.1.2 问题的敏感性	276
8.2 简单的数值解法与有关概念	278
8.2.1 欧拉法	278
8.2.2 数值解法的稳定性与准确度	280
8.2.3 向后欧拉法与梯形法	282
8.3 龙格-库塔方法	284
8.3.1 基本思想	284
8.3.2 几种显式 R-K 公式	285
8.3.3 显式 R-K 公式的稳定性与收敛性	289
8.3.4 自动变步长的 R-K 方法	290
8.4 多步法	292
8.4.1 多步法公式的推导	292
8.4.2 Adams 公式	295
8.4.3 更多讨论	298
8.5 常微分方程组与实用技术	299
8.5.1 1 阶常微分方程组	299
8.5.2 MATLAB 中的实用 ODE 求解器	302
应用实例: 洛伦兹吸引子	306
评述	308
算法背后的历史: “数学家之英雄”欧拉	309
练习题	310
上机题	312
附录 A 有关数学记号的说明	314
附录 B MATLAB 简介	316
附录 C 部分习题答案	336
算法索引	339
术语索引	341
参考文献	349

# 第 1 章 数值计算导论

本章先简单介绍数值计算的背景,再讨论数值计算误差的有关概念,以及影响结果准确度的各种因素,然后结合计算机浮点运算系统分析舍入误差现象,最后给出减少舍入误差的若干建议.

## 1.1 概 述

### 1.1.1 数值计算与数值算法

数值计算(numerical computing)是在理工类各学科专业中广泛运用的一项技术,多年来在我国高等教育培养体系中受到重视,讲授数值计算有关知识的“数值分析”、“计算方法”课程也逐渐成为各专业的必修课或重要选修课.近些年来,计算机科学与技术发展迅速,计算机已成为日常工作、生活中不可缺少的工具.在这种情况下,数值计算与计算机的联系变得更为密切,其应用也日益广泛.

数值计算是横跨计算数学与计算机学科的交叉学科.图 1-1 显示了多个学科经过发展、融合,形成“数值计算”方向的过程.为了突出数值计算在各种科学与工程问题中的应用以及它作为计算机学科一部分的重要性,近年来也将它称为科学计算(scientific computing).

与数值计算联系紧密的一个研究方向是高性能计算,它的研究对象是高性能的计算机硬件体系结构及其应用,包括并行计算机、并行算法等内容.事实上,在高性能计算的应用中,数值计算占有相当重要的地位.

与计算机学科的其他方向不同,数值计算中的问题和方法有其鲜明的特点.它主要处理连续物理量,例如时间、距离、速度、温度、密度、电压、压强、应力等,而不是离散量.同时,数值计算涉及的问题很多都是连续数学问题(例如涉及求导数、积分或非线性方程等),理论上不可能通过有限步计算出准确的结果,因此求解过程往往需要做近似,并通过有限的迭代步得到“充分接近准确解”的近似解.由于计算机不能精确表示所有的实数,数值计算的每一步几乎都存在近似,因此估计、分析计算结果的准确度非常重要.可以将数值计算的研究目标归纳为:寻找一个能迅速完成的(迭代)算法,同时估计计算结果的准确度.

数值计算研究的核心内容是数值算法的设计与分析.在计算机界有句名言:“计算机程序=数据结构+算法”,从中可见算法的重要性.算法又可分为“数值算法”和“非数值算法”,两者有着明显的区别.数值算法用途广泛,发展迅速,具有跨学科的特点,而“非数值算法”的研究主要限于计算机科学的范围内.自从计算机问世以来,算法对科学与工程发展的

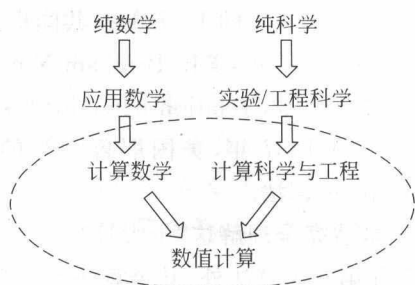


图 1-1 多个学科经过发展、融合,形成数值计算研究方向

推动作用有目共睹。IEEE<sup>①</sup>主办的《科学与工程中的计算》杂志(*Computing in Science & Engineering*)开展了一次评选,选出在20世纪对科学和工程的发展与实践影响最大的10个算法,它们按时间顺序依次如下。

(1) 1946年,美国 Los Alamos 国家实验室的 John Von Neumann(冯·诺依曼)、S. Ulam 和 N. Metropolis 编写的 Metropolis 算法,即蒙特卡洛(Monte Carlo)方法。

(2) 1947年,美国兰德(RAND)公司的 G. Dantzig 创造的解线性规划问题的单纯型算法(Simplex 算法)。

(3) 1950年,美国国家标准局数值分析研究所的 M. Hestenes、E. Stiefel 和 C. Lanczos 开创的 Krylov(克雷洛夫)子空间迭代法。

(4) 1951年,美国 Oak Ridge(橡树岭)国家实验室的 A. Householder(豪斯霍尔德)形式化的矩阵计算的分解方法。

(5) 1957年,美国 IBM 公司的 J. Backus 领导开发的 FORTRAN 最优编译器算法。

(6) 1959—1961年,英国伦敦 Ferranti 公司的 J. Francis 发现的计算矩阵特征值的稳定算法,即 QR 算法。

(7) 1962年,英国伦敦 Elliott Brothers 公司的 T. Hoare 提出快速排序算法,即 Quicksort 算法。

(8) 1965年,美国 IBM 公司 Watson 研究中心的 J. Cooley 和普林斯顿大学及 AT&T 公司贝尔实验室的 J. Tukey 共同提出的快速傅里叶变换算法,即 FFT 算法。

(9) 1977年,美国 Brigham Young 大学的 H. Ferguson 和 R. Forcade 提出的整数关系检测(integer relation detection)算法。

(10) 1987年,美国耶鲁大学的 L. Greengard 和 V. Rokhlin 发明的快速多极(fast multipole)算法。

虽然完全理解这“十大算法”不是件容易的事情,但粗略看看可以发现,除了第5个、第7个和第9个算法外,其余算法都与数值计算关系密切。由此可见数值算法的重要性和普遍性。本书将讨论最基础、应用最广的一些数值算法,部分内容也涉及这十大算法。

### 1.1.2 数值计算的问题与策略

数值计算的问题来自各个科学和工程分支,可归纳为下述3种情况。

(1) 没有解析解的数学问题。一个简单的例子是五次或更高次一元代数方程的求解,例如:

$$x^5 + 2x^4 - 3x^3 + 4x^2 + 5x - 6 = 0$$

根据阿贝尔定理<sup>②</sup>我们知道,五次以及更高次代数方程没有通用的求根公式。因此只能采用数值算法得到其近似解。

(2) 有解析解的数学问题,但解析公式的计算很复杂。容易想到的例子是涉及  $e^x$ 、 $\sin x$  等函数的计算问题<sup>③</sup>,因为将这些函数的计算转化为加、减、乘、除运算时涉及无穷级数,因

① the Institute of Electrical and Electronics Engineers: 电器与电子工程师学会,是世界上最大的专业学术组织。

② 1824年,Abel(阿贝尔)证明了次数大于四次的多项式并不都有用初等运算及系数表示的求根公式。

③ 在计算机高级编程语言中,已将常用函数的计算编制成库函数提供给用户,因此一般用户往往忽视了其内部的技术细节。



此须采用数值计算中的近似、截断等技术加以处理。

(3) 在科学与工程研究中模拟难以形成的实验条件。这是最常见,也是最贴近应用的情况,被称为数值模拟(simulation)或计算机仿真。例如,在天体物理研究中,有些过程是不能直接或者通过实验再现的,因此只能用计算机建立有关物理方程的模型,然后通过数值求解这些方程进行模拟实验。除了处理这种用其他手段无法解决的问题外,数值计算还被大量用于解决“常规”实验可以解决的问题,其好处是降低时间和金钱的成本,而且更安全。例如,用数值计算模拟汽车碰撞实验、集成电路芯片投产前的性能模拟等。

通过数值计算解决问题的过程通常包括以下 6 步。

- (1) 根据相关学科的背景知识建立数学模型,通常是某种类型的方程。
- (2) 研究数值求解这个方程的算法。
- (3) 通过计算机程序或软件实现这个算法。
- (4) 在计算机上运行软件进行数值模拟。
- (5) 将计算结果用较直观的方式输出,例如使用计算机可视化技术。
- (6) 解释并确认计算结果,如果需要,调整参数后重复上面的某些或全部步骤。

上述步骤中的第(2)、(3)两步是数值计算研究的主要内容,而数值计算的有关知识对设计好其他各步都有帮助。应该强调的是,上述各步骤相互间紧密关联,最终计算结果的准确性和效率受这些步骤整体的影响。此外,问题的实际背景和要求也左右着各步骤中方法的选择。

数值计算处理的问题还应当是适定的。如果一个数学问题的解存在、唯一,且连续地依赖于问题的数据,则称这个问题是适定的(well-posed, well-defined)。这里“连续地依赖于问题的数据”指问题数据的微小改变不会造成问题解的剧烈变化。这个条件对于数值计算是极其重要的,因为数值计算以有限字长的计算机为工具,计算过程中数据的扰动总不可避免。当然,绝大多数科学与工程问题也满足适定条件,只有少数例外情况,例如地震问题的计算模型。在各种适定的数值计算问题中,仍有一些问题的解对数据扰动是比较敏感的。关于问题敏感性的定义和讨论,将在后面的 1.2.3 节详细介绍。

求解数值计算问题的一般策略是将复杂或困难的问题用解相同或相近的简单问题代替,这种近似过程通常包括下述 7 种。

- (1) 用有限维空间代替无限维空间。
- (2) 用有限的过程代替无限的过程。
- (3) 用代数方程代替微分方程。
- (4) 用线性问题代替非线性问题。
- (5) 用低阶系统代替高阶系统。
- (6) 用简单函数代替复杂函数。
- (7) 用简单结构的矩阵代替一般的矩阵。

例如,在函数逼近问题中就是用有限维子空间近似无限维的函数空间;在计算无穷级数或积分时,就是将无限的过程近似为有限的过程;求解微分方程的基本方法是将其转化为代数方程;求解非线性方程通常将它转化为一组线性方程的求解;非线性方程的线性化也可看成是将高阶系统变为低阶系统;任何复杂函数的计算都需近似为简单函数,因为计算机最终只能进行加、减、乘、除等基本运算;用简单结构的矩阵代替复杂结构的矩阵往往能以很小的精度损失换取较大的计算效率提升。在本书后续章节的内容中,对各种数值方法的介绍

将充分体现这些策略。

实际的求解过程通常包括两步:在保证问题的解不变或变化不大的前提下将给定问题转化为另一个容易求解的问题,以及对简化后得到的问题进行求解。因此,好的数值算法应具备两方面特点:一方面要计算效率高(计算时间短、占用内存少等);另一方面还要尽可能地准确、可靠,也就是说,在出现各种近似的前提下还能得到尽可能准确的结果。

### 1.1.3 数值计算软件

近几十年来,随着计算机软件、互联网等技术的发展,已涌现出一些高质量的数学软件或程序包,其中一些还可免费获得源代码。有效地借助这些软件、程序,可方便地求解一些典型问题,推进具体的科学与工程研究工作。使用数学软件的重要性已得到了广泛认同,基于此种考虑,目前理工科大职业院校基本上都开设了“数学实验”课程。本书在介绍数值计算软件的基础算法的同时,还希望读者重视数值算法程序的编写和数值计算软件的使用,加深对方法的理解,了解最新的进展,真正获得实践能力。

在介绍一些具体的数值软件之前,先列出高质量数学软件应具有的特点<sup>[1]</sup>。

- (1) 可靠。对一般的问题总能正确运行。
- (2) 准确。根据问题和输入数据产生精确的结果,并能对达到的准确度进行评估。
- (3) 高效率。求解问题所用的时间和存储空间尽可能地小。
- (4) 方便使用。具有方便、友好的用户界面。
- (5) 可移植。在各种计算机平台下都(或经少量修改后)能使用。
- (6) 可维护。程序易于理解和修改(开放源代码的软件)。
- (7) 适用面广。可求解的问题广泛。
- (8) 鲁棒。能解决大部分问题,偶尔失败的情况下能输出提示信息,及时地终止。

事实上,满足所有这些特点的软件几乎是不存在的。它们只是作为使用者挑选软件考虑的因素,同时也为数值算法软件开发者提出了努力的目标。

表 1-1 列出了一些重要的数值计算软件、程序包的来源,其中很多能通过互联网免费获得。这些免费程序代码通常由 FORTRAN 语言编写,也有的使用的是 C 语言、C++ 语言。CMLIB、Netlib 和 NR 是 3 个重要资源,它们都汇集了来自不同渠道的各种程序包,这些程序源代码可以被用户编写的程序调用,或者嵌入到用户自己开发的应用程序中。另外,CMLIB 还具有方便的检索、查找功能。更多的数值软件资源参见文献[2],或通过互联网进行搜索。

表 1-1 重要的数值计算软件、程序包的网络资源

名称	内容说明	商业/免费	网址
CMLIB	美国国家标准技术协会(NIST)的数学与统计软件虚拟仓库和检索系统	免费	gams. nist. gov
FMM	<i>Computer Methods for Mathematical Computations</i> 一书(参见文献[3])所附软件	免费	www. netlib. org
HSL	英国 Science and Technology Facilities Council 提供的科学计算程序包	免费	www. hsl. rl. ac. uk
IMSL	Visual Numerics 公司提供的数学与统计程序软件库	商业	www. vni. com

续表

名称	内容说明	商业/免费	网址
NAG	NAG公司提供的数值算法程序集	商业	www.nag.com
NAPACK	<i>Applied Numerical Linear Algebra</i> 一书(参见文献[4])所附软件	免费	www.netlib.org
Netlib	汇集各种免费数值计算软件的网站	免费	www.netlib.org
NR	<i>Numerical Recipes</i> 系列书(参见文献[5])所附软件	部分免费	www.nr.com
Numeralgo	期刊 <i>Numerical Algorithms</i> 中的软件	免费	www.netlib.org
MATLAB	MathWorks公司出品的数学软件	商业	www.matlab.com
PORT	贝尔实验室开发的软件	部分免费	www.netlib.org
SLATEC	美国政府实验室联合汇编的软件	免费	www.netlib.org
SOL	美国斯坦福大学系统优化实验室开发的优化及相关问题的软件	免费	www.stanford.edu/group/SOL/
TOMS	期刊 <i>ACM Transactions on Mathematical Software</i> 中的算法程序,用名字和数字编号两种方式标识	免费	www.netlib.org

除了包含源代码的程序包,另一种数值计算软件的形式是交互式集成环境,它能提供强大、丰富的数学工具,同时有较好的图形/图像显示功能和高级编程语言,使得在它的基础上能快速地开发出用户所需的软件原型。当前应用最广泛的数值计算软件是 MathWorks 公司的 MATLAB 软件。MATLAB 是一个交互式系统,汇集了大量的数值算法,尤其是处理线性代数、矩阵计算问题的能力很强。MATLAB 本身也定义了一种高级编程语言,语法比较简洁,有利于快速地编写出程序,同时附带的其他功能形成了一个很好的集成开发环境。表 1-2 将 MATLAB 与其他高级编程语言进行了对比。

表 1-2 MATLAB 与其他高级编程语言的比较

比较项目	MATLAB(作为编程语言)	C、C++、FORTRAN
	第四代编程语言	第三代编程语言
编译方式	解释器,或 JIT 加速器(v 6.5 以后版本)	编译器
是否声明变量	不需要	需要
开发时间	较快	较慢
运行时间	较慢	较快
开发环境	集成环境(编辑器、调试器、命令历史、变量空间、profiler、编译器)	—

应说明的是,由于 MATLAB 中包含大量的内部算法函数,如果编程时主要使用的是这些内部函数,MATLAB 程序的运行速度并不会比其他高级语言实现的相同算法慢。此外,较新的 MATLAB 版本具有一些加速运行的机制,因此只要稍加注意,用 MATLAB 进行编程实验是非常便捷的。

除了集成了大量先进的数值算法,MATLAB 还提供了强大的计算可视化功能。通过 MATLAB 中的命令能很方便地分析计算的数据,以及将计算结果以图形的形式加以直观