清華大學 计算机系列教材

喻文健 编著

# 数值分析与算法

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

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科学与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.本书涵盖了数值分析、矩阵计算领域最基本、最常用的一些知识与方法，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.每章配备了应用实例、算法背后的历史、评述等子栏目，书末附有术语索引.对常用算法给出了简明的算法伪码描述，在附录中还包括了 MATLAB 软件的简介，便于读者进行上机编程实验.

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.



图书在版编目（CIP）数据

数值分析与算法/喻文健编著. —北京：清华大学出版社，2012.1
（清华大学计算机系列教材）
ISBN 978-7-302-26645-7

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(2011)第 179152 号

**责任编辑**：白立军 王冰飞
**责任校对**：焦丽丽
**责任印制**：何 芊

**出版发行**：清华大学出版社　　**地　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编**：100084
**社 总 机**：010-62770175　　**邮　购**：010-62786544
**投稿与读者服务**：010-62795954，jsjjc@tup.tsinghua.edu.cn
**质 量 反 馈**：010-62772015，zhiliang@tup.tsinghua.edu.cn
**印 刷 者**：清华大学印刷厂
**装 订 者**：三河市新茂装订有限公司
**经　销**：全国新华书店
**开　本**：185×260　**印　张**：22.25　**字　数**：559 千字
**版　次**：2012 年 1 月第 1 版　**印　次**：2012 年 1 月第 1 次印刷
**印　数**：1～3000
**定　价**：35.00 元

产品编号：036930-01

# 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，近些年来也被称为科学计算(scientific computing)，是当今科学研究的三种基本手段之一，它是计算数学、计算机科学与其他工程学科相结合的产物.随着计算机技术的发展与普及日益受到人们的重视，尤其是近十年来，科学技术逐渐发展进入“纳米时代”、“互联网时代”，各种高科技领域产生出大量高复杂度的计算问题，使得发展、推广数值计算变得空前重要.

本书的主要内容与一般的“数值分析”教材基本一致，但比较突出算法设计与实际应用，强调通过理论联系实际提高学生的实践能力.为此，本书在数值软件与程序资源方面做了较多介绍，同时结合 MATLAB 软件对一些较新的算法与实用技巧进行讨论.由于主要面向信息科学与技术有关专业的本科生，本书在编写细节上力求做到以下几点：

1. 对数学理论的介绍尽量简明扼要.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.同时，阐明有关数学理论的意义与用途.

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，同时分析其时间、空间复杂度.说明算法实际应用中的细节问题，对几个较新的实用算法还结合 MATLAB 源程序加以介绍.通过“应用实例”子栏目以及对 MATLAB 相关命令的介绍，突出算法的实际应用.

3. 从读者的角度出发增强可读性与实用性.尽量用图、表等形象的方式对概念、现象进行解释，书末附有术语索引，便于查阅.每章编写了“算法背后的历史”子栏目，增强阅读的趣味性.通过附录介绍 MATLAB 的基本知识，在正文中的“应用实例”和部分例题中也给出了 MATLAB 源程序，读者可根据它们动手实践.

4. 在内容编排上有利于教学.根据知识的相关性安排各章，使得数值线性代数(矩阵计算)的内容集中出现在第 3 章到第 6 章的前半部分，这样的安排也能适合从第 1 章到第 8 章的学习顺序.在每章的“评述”部分列出主要知识点，便于学生复习，而其他评述内容有助于感兴趣的读者深入学习有关知识.每章给出上机实习题，附录中提供部分习题的答案.

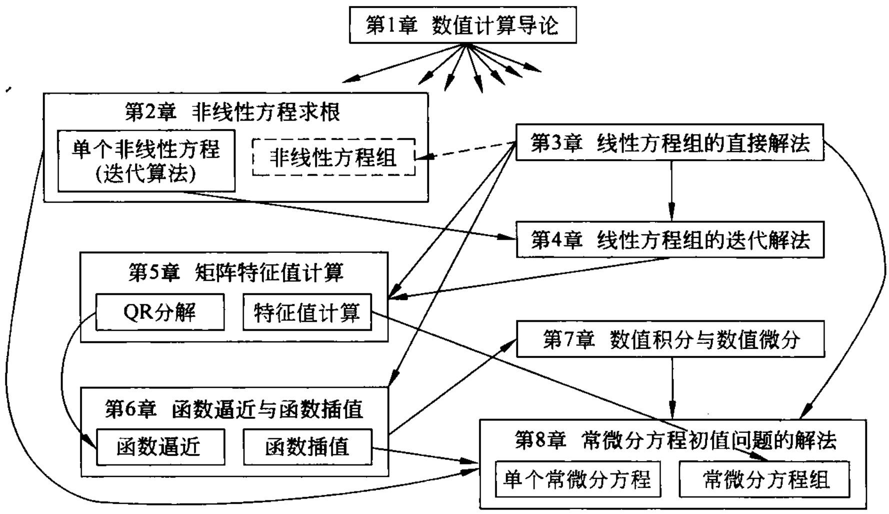
学习数值分析与算法，不但要掌握基本的理论与方法，还必须通过计算机编程实践来进一步理解有关算法及其理论.本书提倡使用 MATLAB 软件来实现算法、解决实际应用问题，主要基于如下理由：(1)MATLAB 本身是一种科学计算环境，其数值计算功能强大，已被广泛用于求解科学工程中的计算问题；(2)MATLAB 具有丰富的数学函数，在学习理解有关教学内容时可直接使用它们，例如在 MATLAB 中执行命令 x=A\b 可方便地求解最小二乘问题或线性方程组；(3)MATLAB 体现了当前科学计算的发展趋势，例如尽量使用向量运算而不是用循环，同时提供丰富的图形工具，易于对计算结果进行可视化；(4)MATLAB 编程语言语法简洁、易于掌握，可节省编程实验时间.

本书作者十多年来一直从事数值算法与软件的科研工作，自 2005 年开始在清华大学计算机系主讲数值分析与算法的有关课程十余次，本书是对近几年教学工作的一个总结.在编

写本书的过程中，作者认真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合中国大学教育等方面取得好的效果，从而适应时代发展对学生培养提出的新要求.

本书适合于 48 个学时或者更多学时的教学安排. 为了方便教学内容的取舍，对于难度较大或者超出传统教学大纲的较新内容，在目录中以“ * ”号进行了标记. 每章最后包含“有关数值软件”等扩展性的内容，可供感兴趣的学生课后阅读.

下图显示了各章主要内容的依赖关系，其中第 1 章是基础，后续各章都会用到有关知识. 对于后续各章，授课教师可根据实际情况适当调整顺序. 应注意，第 2 章中“非线性方程组”的内容依赖于第 3 章的部分知识，而其他部分的依赖关系基本上与章节前后顺序一致.




在编写本教材的过程中，得到了许多本科生、研究生的支持. 白如冰编写了附录 B 的初稿，朱臻垚编写了部分“算法背后的历史”栏目，程康、翟匡亚、汤启明、张青青帮助做了部分文字输入和格式修改工作，袁仲达与何鸥多次担任课程助教，对教学工作的改进做了很大贡献. 本书稿在 2011 年春季学期清华大学的“数值分析”课上进行了试用，收到了选课学生的诸多反馈，在此一并表示感谢. 在作者讲授“数值分析”课程之初，使用了王泽毅教授的课程讲义，其中部分思想也融入了本书之中，在此表示特别的感谢. 另外，还要感谢殷人昆、边计年、蔡懿慈等几位教授给予的帮助与支持，以及清华大学出版社的编辑在出版本书过程中提出的宝贵意见和付出的辛勤劳动.

数值计算领域宽广、博大精深，编写本书是一个小小的尝试，希望有助于刚刚接触这个领域的读者打好基础、产生兴趣，起到“抛砖引玉”的作用. 由于作者水平有限，书中难免还存在不妥与错误之处，诚恳地希望读者提出宝贵意见.

喻文健

yu-wj@tsinghua. edu. cn

清华大学计算机科学与技术系

2011 年 8 月于清华园

# 目　录

# 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).

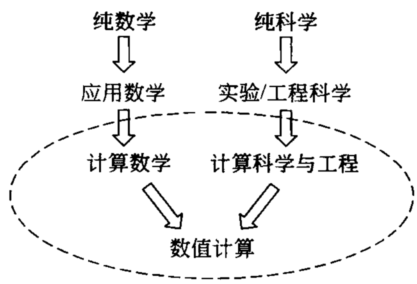



图 1-1　多个学科经过发展、融合，形成数值计算研究方向

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

推动作用有目共睹. IEEE[1] 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、第 7 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,可归纳为下述三种情况:

(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,如:

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

根据阿贝尔定理[2]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[3],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,因此须采用数值计算中的近似、截断等技术加以处理.

---

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.

(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几步:

(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小节详细介绍.

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几种:

(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系列线性方程的求解;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，目前理工科大专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，真正获得实践应用的能力.

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[1]：

(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 是三个重要资源，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 | *Computer Methods for Mathematical Computations* 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 | *Applied Numerical Linear Algebra* 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 | *Numerical Recipes* 系列书(参见文献[5])所附软件 | 部分免费 | www.nr.com |
| Numeralgo | 期刊 *Numerical Algorithms* 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 *ACM Transactions on Mathematical Software* 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 与第三代编程语言的比较**

| | 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 |
| 是否申明变量 | 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

显示. MATLAB 中还包含了很多选项工具箱(Toolbox),为信号处理、图像处理、控制、系统辨别、优化、统计、金融、通信等各种专业应用提供专门的工具. MATLAB 已成为广大科学与工程研发人员首选的工具软件,其正式用户数目已超过了一百万,很多大学也采用 MATLAB 作为数值计算相关课程的教学辅助软件.

MATLAB 是跨平台的软件,无论在 Windows 操作系统下,还是 UNIX/Linux 操作系统下都能使用. 与 MATLAB 功能类似的还有一些免费软件,其中比较著名的一个是 octave. 可从互联网上免费下载,它在 UNIX/Linux 操作系统下运行,界面和功能与 MATLAB 基本一致. 另外两种类似于 MATLAB 的免费软件是 RLaB 和 Scilab.

本书讨论的算法基本上都能在表 1-1 中的数值软件资源中找到实现代码,或在 MATLAB 中找到相应的程序命令,在后面介绍具体内容时将提及.

## 1.2 误差分析基础

### 1.2.1 数值计算的近似

前面提到,数值计算主要研究求解连续数学问题的算法和程序实现,但这并不意味着我们只需关心数值算法执行过程中的误差. 为了对最终计算结果的准确度进行评价,我们必须知道一些在数值计算开始之前就存在的误差或近似. 它们主要有以下三种情况:

(1) 建立数学模型时做的近似. 这个过程中可能简化或忽略系统的一些不重要的物理特性(例如摩擦、空气阻力等).

(2) 经验或测量造成的数据误差. 有些计算中用的数据(例如重要的物理常数、普朗克常数、万有引力常数等)都是通过实验测量得到的,或多或少存在一些误差.

(3) 输入数据来自以前计算的结果. 一个实际的科学或工程问题通常被分解为多个前后衔接的数值计算问题,因此当前问题的输入数据包括问题的计算结果,所以存在误差.

通过分析计算前主要的近似来源我们知道,为了使最终结果准确,应采取建立更精确数学物理模型、采用更准的测量值,以及改变前一步计算方案以减少输入数据误差等措施.

计算前的误差有时会超出我们的控制范围,而计算过程中的近似往往可以控制,采用不同的算法和编程实现技巧对其影响很大. 计算过程中的近似主要有:

(1) *截断误差或方法误差*. 求解一个数学方程往往可采用多种数值方法,各种数值方法往往都进行了一些简化处理,例如,函数 $f(x)$ 用其 $n$ 阶泰勒展开 $f(0)+\frac{f'(0)}{1!}x+\frac{f''(0)}{2!}x^2+\cdots+\frac{f^{(n)}(0)}{n!}x^n$ 代替,则截断误差为 $\frac{f^{(n+1)}(\xi)}{(n+1)!}x^{(n+1)}$,$\xi$ 在 0 和 $x$ 之间.

(2) *舍入误差*. 无论是手工计算、利用计算器,还是数字计算机,输入与结果数据都只能用有限位数字表示. 即数据的“四舍五入”会产生误差. 此外,常用的十进制数据输入计算机后需变成二进制数据,这个转换过程产生的误差通常也归为舍入误差.

由于计算过程中舍入误差总是存在,研究数值算法抵抗这种扰动的稳定性非常重要. 关于算法稳定性的概念见 1.2.4 节. 另外,在各种误差来源中,通常只有几个方面的误差起主要作用,应对其重点考虑和分析.

**例 1.1(数值计算的近似)**: 为了计算地球的表面积,可以采用球体表面积的计算公式: