数值计算方法理论与典型例题选讲

雷金贵　蒋　勇　陈文兵 编

科学出版社

数值计算方法理论与典型例题选讲

雷金贵　蒋　勇　陈文兵　编

科学出版社

北京

内 容 简 介

本书是为理工科大学本科课程“数值分析”和“计算方法”编写的教材与课外自学指导两用书，主要内容包括引言、插值法、线性方程组的直接解法与迭代法、方程求根、数据拟合与函数逼近、数值积分与数值微分、常微分方程数值解法、矩阵特征值与特征向量的计算. 此外，为了兼顾学生能力的培养和考试技能的提高，并帮助其合理掌握学习重点，本书附录包括上机实习、理工科专业“计算方法”模拟题 6 套、数学专业“数值分析”模拟题 2 套和数学专业硕士研究生入学考试“数值分析”模拟题 2 套，并附有答案.

本书可作为理工科大学数学及相关专业本科和研究生 “(数值)计算方法”课程的教材，也可供从事科学与工程计算的科技工作者参考.

图书在版编目(CIP)数据

数值计算方法理论与典型例题选讲/雷金贵，蒋勇，陈文兵编. —北京：科学出版社，2012

ISBN 978-7-03-035019-0

Ⅰ. ①数… Ⅱ. ①雷… ②蒋… ③陈… Ⅲ. ①数值计算－计算法－高等学校－教学参考资料 Ⅳ. ①O241

中国版本图书馆 CIP 数据核字(2012) 第 133201 号

责任编辑：伍宏发 顾 艳 胡 凯／责任校对：张凤琴

责任印制：赵德静／封面设计：许 瑞

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

新科印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2012 年 7 月第 一 版 开本：787×1092 1/16

2012 年 7 月第一次印刷 印张：21 1/4

字数：486 000

定价：48.00 元

(如有印装质量问题，我社负责调换)

前　言

数值算法是将数值计算理论、方法与计算机相结合, 并在计算机上编写、运行相应的计算软件以实现具体计算过程, 进而得出结果的方法统称. 而数值计算方法就是一门研究数值算法理论与应用的学科.

利用计算机进行数值计算工作的核心是: ① 研究建立数学模型; ② 研究数值算法的构造方法, 并探索算法的特点与规律, 如计算速度、收敛性和稳定性等问题; ③ 理论联系实际, 研究如何编制相应的算法软件, 以及如何降低软件的计算复杂度、时间复杂度, 减少内存的占用, 降低算法的结构复杂性, 增加算法的稳定性等问题. 这些也是计算方法的研究内容.

因此, 通过数值计算方法课程的学习, 学习者不但能够掌握数值计算方法的理论与方法, 同时也能提高发现问题、分析问题、解决问题的能力, 从而提高综合运用数学知识的技巧和能力. 然而, 在学习数值计算方法的时候, 总有些人会遇到一些困难, 比如由于基础知识欠缺, 有些人虽然有兴趣但是苦于繁杂的推导与计算, 学习的激情无法持久; 有些人理解理论有难度, 需要老师长期耐心的指导, 学习才能继续; 有些人则无法理论联系实际; 也有些人学习总是很难抓住重点等. 身为教育工作者, 有责任给正在学习数值计算方法课程的学习者们提供一本理论推导详尽、重点突出、思路清晰的读物, 以帮助他们鼓起勇气战胜困难并树立继续学习的决心.

本书是为理工科大学本科课程 “数值分析” 和 “计算方法” 编写的教材与课外自学指导两用书. 本书强调 “科学、清晰的算法思路大于一切” 的理念, 坚持 “能力培养与考试技能相结合” 的指导原则, 注重理论联系实际. 因此, 本书将重点放在了算法思想的梳理与证明及重要知识点的习题讲解上, 并特别介绍了一些与考试相关的内容, 如结业考试的重点、难点等, 并在最后附了 10 套题目, 其中, 6 套理工科专业 “计算方法” 模拟题, 2 套数学专业 “数值分析” 模拟题, 再加 2 套数学专业硕士研究生入学考试 “数值分析” 模拟题, 每套题目都附有参考答案. 全书采用模块化结构, 章节相对独立, 同时, 为便于读者查阅, 本书有针对性地将每个章节的基本理论讲解和典型例题分析划分为两部分; 全书思路清晰、语言简洁、论证严谨、分析深入浅出, 便于教师根据学生专业特色灵活安排教学内容以及初学者根据自身情况合理安排学习进度.

我们希望读者在通读本书之前, 能够掌握微积分学、代数学以及常微分方程等学科基础知识, 另外, 希望读者至少学过一门程序设计语言, 如 C、Fortran90、Matlab 等.

本书是编者多年教学实践和经验的积累, 由南京信息工程大学数学与统计学院蒋勇教授策划, 历时一年多整理而成. 在成书过程中, 编者主要参考了林成森、李庆杨、孙志忠和戴嘉尊等教授的著作, 以及南京信息工程大学信息与计算科学系杨建伟、刘文军、张

天良、朱建等老师的教学课件. 对以上提到的诸位老师, 编者致以衷心的感谢. 由于编者能力所限, 本书错误之处在所难免, 欢迎读者批评指正.

编　　者

2011 年 10 月于南京

目　　录

第1章 引 言

作为数值计算工作的一部分, 误差估计占有重要地位. 如果数值计算得到的结果没有进行相应的误差估计, 那么, 一般来说, 该结果就不能保证是十分可靠或可用的, 因此一定要尽可能地进行误差估计这项工作. 那么, 什么是误差? 什么是误差估计? 怎么进行误差估计呢? 本章将对这些看似简单的问题做一个汇总, 以方便后面章节的叙述.

本章的重点是误差、有效数字与机器数系的概念与相关知识, 以及数值计算陷阱的预防措施. 在 1.3 节介绍应当掌握的典型例题.

1.1 误差、有效数字与机器数系

利用计算机进行数值计算可以按照图 1.1 所示的流程进行操作.

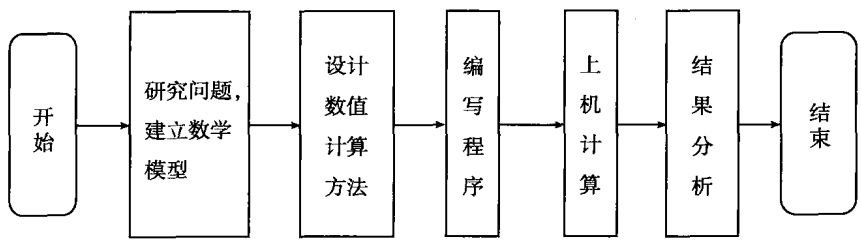

图 1.1 计算机数值计算流程图

上述流程图是用计算机解决实际问题的一般步骤, 虽然不同的书中有不同的画法, 但本质上区别不大. 图 1.1 表明, 数值计算的第一步是建立数学模型, 该步骤在一般情况下可以省略, 因为很多问题都早已建立了较为成熟的数学模型; 第二步是算法设计, 其实就某些具体问题来说, 该步骤也不需要每个细节都由自己完成, 科技工作者只有学会 "站在巨人的肩膀上", 才能事半功倍, 但是作为初学者, 应该关注如何设计算法, 毕竟这是我们学习这门课程的根本目的; 第三步、第四步要求将算法在计算机上实现, 严格来说这也是数值计算的重要环节, 但本书的重点不在于此, 只是在有些地方稍作论述; 最后一项是结果分析, 即分析结果的有效性和正确性, 一般来说是指误差估计以及算法的稳定性和收敛速度的分析等.

1.1.1 误差的来源与概念

根据图 1.1, 可以确定如下误差来源, 即**模型误差**、**观测误差**、**截断误差**和**舍入误差**.

那么到底什么是模型误差? 例如要研究大气运动问题, 如果研究人员研究的是有黏性、可压缩、斜压、干气体的深厚系统, 则需要建立如下 Navier-Stokes 方程组 (简称 N-S 方程组)

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = -\dfrac{1}{\rho}\nabla p - 2\boldsymbol{\Omega}\times\boldsymbol{v} + \boldsymbol{g} + \boldsymbol{F} \\ \dfrac{\mathrm{d}\rho}{\mathrm{d}t} + \rho\nabla\cdot\boldsymbol{v} = 0 \\ c_p\dfrac{\mathrm{d}T}{\mathrm{d}t} - \dfrac{ART}{p}\dfrac{\mathrm{d}p}{\mathrm{d}t} = Q \\ p = \rho RT \end{cases}$$

上述 N-S 方程组就是天气预报问题的数学模型, 该模型考虑了气体的动力学特征, 如压强 p、速度矢量 $\boldsymbol{v}$ 和密度 ρ 以及热力学特征如温度 T、热源 (或汇)Q, 还考虑了地球引力 $\boldsymbol{g}$、摩擦力 $\boldsymbol{F}$ 以及地转偏向力 $-2\boldsymbol{\Omega}\times\boldsymbol{v}$ 等因素的影响, 但没有考虑大气化学特征和大气电场等诸多因素对气体状态的影响. 因此, 即使能够求出上述方程组的精确解, 这个所谓的 "精确解" 也注定不是真实大气的真实状态. 这种建立数学模型时由于舍弃化学特性和电场等影响因素而产生的误差称为**模型误差**. 用数学方法解决任何实际问题都会产生模型误差, 也就是说, 模型误差是不可避免的.

观测误差就比较好理解, 例如, 今天南京的实际气温是 25℃, 但是测量人员用仪器测量得到的温度测量值是 25.3℃, 这 0.3℃的差值就是今天南京大气温度的测量误差. 测量仪器总是会产生误差, 况且观测员在读数据的时候也会产生误差, 因此测量误差也是不可避免的. 由仪器和人为测量产生的误差通称为**测量误差**或**观测误差**.

一般来说, **舍入误差**产生在存储环节, 比如用二进制表示十进制数 0.01, 然后再输出十进制表示后的结果时, 将会产生截断误差, 这种误差就是舍入误差. 事实上, 如果我们用 20 个比特存储十进制数的小数部分, 然后再转化成十进制输出, 计算机输出的结果将是 0.009 999 275 而不是 0.01. 因此原来的十进制数 0.01 与计算机输出结果 0.009 999 275 之间的差值就是舍入误差. 舍入过程可参见图 1.2.

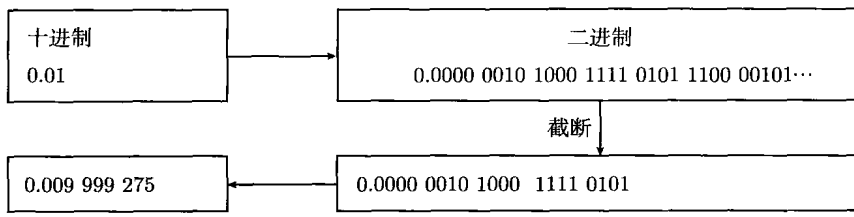

图 1.2　十进制数 0.01 的二进制表示和输出结果

同理, 用计算机存储数值 1/3 时, 在内存中的实际值是小数, 例如, 0.333 333 33. 由于计算机总是用有限位二进制单元来存储整数和实数, 因而舍入误差也是不可避免的.

截断误差可以这样理解, 例如 N-S 方程组由于无法直接求得它的精确解, 只能通过数值方法来求解, 这种数值解一般是通过一种称为模式或差分格式的算法程序求得, 如当前气象学界使用的 WRF 模式, 则模式的 "精确解" 与 N-S 方程组的 "精确解" 之间的误差称为**截断误差**. 有关截断误差的讨论, 只有大家学习了微分方程数值解或数值积分、插值法等相关章节后才能有一个比较深刻的理解, 读者在此不必深究.

总之, 在用数值方法求解实际问题的时候, 各种误差都不可避免, 并且最终的 "数值解" 与物理问题 "精确解" 的误差总是以上面提到的各种误差的**总和**形式出现. 况且如果研究的问题很复杂, 则很难得到物理问题的 "精确解", 所以误差估计会较难进行. 所以,

青年学生在工作之前首先应该学会工作方法, 否则将来的工作将难以开展.

可能也有人认为既然数值计算很困难, 那为什么不能求出物理问题的精确解呢?

对于这个问题, 可以用上述天气预报的例子来说明. 理论上, 现在老百姓每天从电视台、网络或者报纸中获取的天气预报信息基本上是中央气象台用数值预报模式计算出来的 N-S 方程组的数值解. 由于 N-S 方程组是一个非线性的偏微分方程组, 按照人类目前的科学技术水平并不能求出其显式的精确解, 只能求其数值解. 每天, 气象台的科学家们就是用全国各个气象观测台站的地面常规观测资料、卫星遥感资料以及雷达探空资料等诸多数据构造的大气初始状态为初始值, 用数值预报模式计算出数值结果; 该结果经过气象台站的软件加工处理后就是我们得到的天气预报. 天气预报有时很准确, 有时不准确, 这说明对预报结果的误差估计有时很小, 有时很大.

学习数值计算方法课程, 首先要掌握两个定义: **绝对误差**和**相对误差**. 所谓**绝对误差**e, 就是指两个数的差

$$e = x^* - x \tag{1.1}$$

其中, x^* 是某个数学量或物理量 (可以是函数的自变量或因变量) 的准确值, x 是 x^* 的测量值或近似值.

两个数的**相对误差**定义为

$$e_r = \frac{x^* - x}{x^*} \tag{1.2}$$

由于准确值 x^* 通常无法获得, 因此常用公式 $\dfrac{x^* - x}{x}$ 代替 $\dfrac{x^* - x}{x^*}$ 来求近似相对误差, 有时也称 $\dfrac{x^* - x}{x}$ 是相对误差, 实际上当 x 非常接近 x^* 时,

$$\frac{x^* - x}{x} - \frac{x^* - x}{x^*} \approx o\left((x^* - x)^2\right)$$

因此当 $\dfrac{x^* - x}{x^*}$ 无法计算时, 并不需要估计表达式 $\dfrac{x^* - x}{x} - \dfrac{x^* - x}{x^*}$ 的大小, 我们直接就用 $\dfrac{x^* - x}{x}$ 计算相对误差 e_r 的大小.

当存在常数 δ, 使得

$$|e| = |x^* - x| \leqslant \delta$$

时, 则称 δ 为**绝对误差限**; 而当存在常数 γ 使得

$$|e_r| = \left|\frac{x^* - x}{x^*}\right| \leqslant \gamma$$

时, 称 γ 为**相对误差限**. 绝对误差限可以用来估计精确值, 即

$$x - \delta \leqslant x^* \leqslant x + \delta$$

通常, 要衡量一个近似值是否与其对应的精确值足够接近, 不能简单看绝对误差是否小, 而是要看相对误差的大小. 例如, 在测量小行星的大小时, 行星半径的绝对误差是几公里, 则可以认为测量已经相当精确, 但如果测量的是一个普通建筑物的高度, 相差即便只有几米, 误差也已经很大了. 所以通常情况下, 不能仅靠绝对误差就断定一个近似值的好与坏.

1.1.2 误差的传播

下面以二元函数 $y=f(x_1,x_2)$ 为例, 解释自变量的误差是怎样传递给因变量的.

设 x_1,x_2 与 x_1^*,x_2^* 分别是二元函数 $y=f(x_1,x_2)$ 自变量的近似值与精确值, y 与 y^* 是因变量的近似值与精确值, 则函数值的绝对误差为

$$e(y)=y^*-y=f(x_1^*,x_2^*)-f(x_1,x_2)$$

将 $f(x_1^*,x_2^*)$ 在 (x_1,x_2) 处 Taylor 展开到一阶导数项, 略去高阶项, 得

$$\begin{aligned} e(y)=y^*-y &\approx f(x_1,x_2)+\frac{\partial f(x_1,x_2)}{\partial x_1}(x_1^*-x_1)+\frac{\partial f(x_1,x_2)}{\partial x_2}(x_2^*-x_2)-f(x_1,x_2)\\ &=\frac{\partial f(x_1,x_2)}{\partial x_1}e(x_1)+\frac{\partial f(x_1,x_2)}{\partial x_2}e(x_2) \end{aligned} \tag{1.3}$$

上式说明自变量 x_i 的绝对误差 $e(x_i)$ 对 $e(y)$ 的贡献是 $\dfrac{\partial f(x_1,x_2)}{\partial x_i}e(x_i)$, 其中, 偏导数 $\dfrac{\partial f(x_1,x_2)}{\partial x_i}$ 是这个方向的**放大系数**, $i=1,2$.

二元函数的相对误差为

$$\begin{aligned} e_r(y)=\frac{e(y)}{y} &\approx \frac{\partial f(x_1,x_2)}{\partial x_1}\frac{e(x_1)}{y}+\frac{\partial f(x_1,x_2)}{\partial x_2}\frac{e(x_2)}{y}\\ &=\frac{\partial f(x_1,x_2)}{\partial x_1}\frac{x_1}{y}e_r(x_1)+\frac{\partial f(x_1,x_2)}{\partial x_2}\frac{x_2}{y}e_r(x_2) \end{aligned} \tag{1.4}$$

利用式 (1.3) 和式 (1.4) 可得到一些简单二元函数, 如两个自变量 x_1 与 x_2 的和、差、积、商的绝对误差和相对误差的估计式

$$\begin{aligned} &e(x_1+x_2)\approx e_1+e_2\\ &e(x_1-x_2)\approx e_1-e_2\\ &e(x_1x_2)\approx x_2e_1+x_1e_2\\ &e(x_1/x_2)\approx e_1/x_2-e_2x_1/x_2^2,\quad x_2\neq 0\\ &e_r(x_1+x_2)\approx \frac{x_1}{x_1+x_2}e_{1r}+\frac{x_2}{x_1+x_2}e_{2r},\quad x_1+x_2\neq 0\\ &e_r(x_1-x_2)\approx \frac{x_1}{x_1-x_2}e_{1r}-\frac{x_2}{x_1-x_2}e_{2r},\quad x_1-x_2\neq 0\\ &e_r(x_1x_2)\approx e_{1r}+e_{2r}\\ &e_r(x_1/x_2)\approx e_{1r}-e_{2r},\quad x_2\neq 0 \end{aligned}$$

1.1.3 有效数字

若实数 x^* 的非零近似值为 x, x 的十进制**规格化浮点数**的标准形式为

$$x=\pm 0.x_1x_2\cdots x_n\cdots x_p\cdot 10^m \tag{1.5}$$

其中, $x_1\neq 0$, 如果有 $|x^*-x|=|e(x)|<\dfrac{1}{2}\times 10^{m-n}$, 则称 x**有n位有效数字**.

有效数字的定义, 也可以解释为: 从左边第一位非零数字开始数, 到第 n 位数字截止, 如果绝对误差小于第 n 位数字的数值的 0.5 倍, 则该近似数的有效数字位数为 n.

关于有效数字, 需要记住的是: ① 用四舍五入得到的数都是有效数字; ② 有效数字越多, 截断误差越小, 计算结果越精确.

1.1.4　机器数系

计算机内部能够处理的数据除了整型数据和字符型数据之外, 最常用的是浮点数, 也就是人们常用的实数类型.

实数类型在 C 语言中称为**浮点数**(用 float 定义单精度或用 double 定义双精度), 在 Fortran 语言中称为**实型数据**(用关键字 real*m 来定义实型数据, 其中, m 为一数字, 表示实型数据的种数, 详细解释请参见 Fortran 语言相关文献).

总体来看, 浮点数是计算机表示范围最广、最为常用的一种数据类型, 表示为式 (1.5) 形式的数称为**规格化浮点数**, 式 (1.5) 只是十进制形式浮点数的表示形式. 计算机能够处理的数不仅仅是十进制的, 它有多种形式, 最常见的是二进制、八进制、十六进制三种, 这与计算机用二进制存储数据有关. 1985 年, 美国电气与电子工程师学会 IEEE(Institute of Electrical and Electronics Engineers) 颁布了名为二进制浮点数标准的报告 (编号: 754-1985; 2008 年, 该标准更新为 IEEE 754-2008). 该报告为二进制和十进制浮点数、数据交换格式、舍入误差算法和异常处理操作等提供了一系列的标准. 标准中还规定了单精度、双精度以及扩展精度数据类型的格式, 这些格式被所有的计算机厂商广泛遵守, 并做成硬件模块集成到计算机设备中. 在 2008 年的新标准中, 实数用 64 位二进制数表示, 其中, 从左边起第一个比特是符号位 (sign indicator), 接下来的 11 个比特是指数部分 (characteristic), 剩余的 52 个比特表示小数部分 (mantissa)[1].

一般情况下, 一个 β 进制的规格化浮点数可以表示为

$$x = \pm 0.a_1a_2\cdots a_t \cdot \beta^p \tag{1.6}$$

其中, $a_1 \neq 0$, $0 \leqslant a_j \leqslant \beta - 1, j = 1, 2, \cdots, t, L \leqslant p \leqslant U, a_j$ 和 p 都是整数. 如果把计算机中所有浮点数的集合加上 "机器零" 记为 F, 则 F 被如下 4 个参数所描述, 分别是基数 β、字长 t、阶码范围 $[L, U]$. 集合 F 称为**机器数系**. 应该说, 机器数系 F 是一个分布不均匀的、有限个离散数据的集合, 阶码越小的地方表示的数越稠密, 阶码越大的地方表示的数越稀疏 (图 1.3).

不难证明, 集合 F 仅含有 $1 + 2(\beta - 1)\beta^{t-1}(U - L + 1)$ 个数.

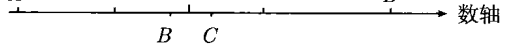

图 1.3　机器数系

A: 绝对值最大负实数, D: 绝对值最大正实数; B: 绝对值最小负实数, C: 绝对值最小正实数

图 1.3 把机器数系能够表示的绝对值最大和绝对值最小的 4 个数标在数轴上, 当一个数的值大于数 D 或者小于 A 时, 则为**上溢**; 当一个非零浮点数的值介于数轴上的 B 和 C 之间时, 则产生**下溢**, 该数被计算机当做数值 0 来处理.

因此当一个实数 $x = \pm 0.\ a_1a_2\cdots a_ta_{t+1} \cdot \beta^p (a_1 \neq 0)$ 被存入计算机后, 称之为机器数, 记为 $fl(x)$. 当前计算机有两种方式产生机器数 $fl(x)$, 一种是直接截取 x 的前 t 位数, 得到 $fl(x) = \pm 0.\ a_1a_2\cdots a_t \cdot \beta^p$, 称这种计算机为**截断机**; 还有一种计算机, 对数 x 的第 $t+1$ 位数字四舍五入得到 $\tilde{a}_t$, 即得到 $fl(x) = \pm 0.\ a_1a_2\cdots \tilde{a}_t \cdot \beta^p$, 称这种计算机为**舍入机**. 一般来说, 用截断机处理数值计算所得结果的误差要大于等于舍入机的误差.