高等学校规划教材

数值计算方法(第二版)

主编　朝伦巴根　贾德彬

中国水利水电出版社
www.waterpub.com.cn

高等学校规划教材

数值计算方法(第二版)

主编　朝伦巴根　贾德彬

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书系统介绍了常用的数值计算方法，包括线性代数方程组、非线性方程（组）的数值解法、矩阵特征值及特征向量的求法、插值法、曲线拟合、数值逼近、数值积分、常微分方程（组）的数值解法、最优化方法和人工智能计算方法。各种计算方法均给出了用 FORTRAN—90 编写的计算程序。

本书系统性较强，重点突出，阐述简明易懂。书中提供的程序符合设计规范，具有很强的应用性。

本书可作为高等学校水文与水资源工程、农业水利工程、水利水电工程、环境工程及工科其他专业教材，并可供工程技术人员参考。

图书在版编目（CIP）数据

数值计算方法 / 朝伦巴根，贾德彬主编. -- 2版
. -- 北京 : 中国水利水电出版社，2011.5
高等学校规划教材
ISBN 978-7-5084-8604-8

Ⅰ. ①数… Ⅱ. ①朝… ②贾… Ⅲ. ①数值计算－计算方法－高等学校－教材 Ⅳ. ①O241

中国版本图书馆CIP数据核字(2011)第085453号

书　　名	高等学校规划教材 **数值计算方法**（第二版）
作　　者	主编　朝伦巴根　贾德彬
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路1号D座　100038） 网址：www.waterpub.com.cn E-mail：sales@waterpub.com.cn 电话：（010）68367658（营销中心）
经　　售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643 全国各地新华书店和相关出版物销售网点
排　　版	中国水利水电出版社微机排版中心
印　　刷	北京市兴怀印刷厂
规　　格	184mm×260mm　16开本　19.25印张　456千字
版　　次	2007年1月第1版 2011年5月第2版　2011年5月第2次印刷
印　　数	5101—7100册
定　　价	**39.00**元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

第二版前言

随着工程技术领域的计算过程日益复杂和要求快速化，计算机已成为解决工程问题不可缺少的有力工具，与计算机应用密切相关的“数值计算方法”成为工科学生的必修课程。本教材正是为适应这种变化，并遵照水文与水资源工程、农业水利工程本科专业教学规范中提出的构造知识体系框架原则及水文与水资源工程专业教学组拟定的“《数值计算方法》教材编写大纲”编写的。

本教材在满足水文与水资源工程、农业水利工程等专业基本需要的基础上，以算法的构造和应用为重点。在内容安排上，力求介绍各类数值问题常用的数值计算方法，为适应水文与水资源工程专业本科和研究生教学的需要，设置了常用的最优化方法和人工智能计算方法，为提高学生分析数据和解决问题的能力，书中还加强了插值、拟合、平滑及逼近等内容；为适应大型水利工程计算，在线性代数方程组数值方法中，介绍了关于特殊矩阵方程组的存储与求解的内容。各章节的阐述方式力求概念明确、简练易懂、由浅入深、自成体系。有关数值计算方法的理论证明、误差分析及收敛性、稳定性的讨论，只在十分必要时才予以涉及。为了便于学生了解算法的原理和执行过程，大部分算法都附有程序框图和程序代码，学生通过对算法程序代码的编译和运行，能够加深对算法的理解，从而有效地培养了学生计算机高级语言表达算法的能力。

本书第一、五、六、七、八章由高新科编写，第二章由高新科、朝伦巴根编写，第三、四、九章由朝伦巴根、王丽萍、刘霞编写，第十章由高瑞忠、刘霞编写，第一、四、九、十章的程序由高瑞忠、李凤玲、刘霞编制，第二、五、六、七章程序由贾德彬、刘霞编制，第三、八章程序由刘廷玺、王丽萍编制。书中程序由贾德彬调试，全书由朝伦巴根统稿。

本书由裴喜春教授主审，李畅游教授校阅了书稿，他们在审核过程中提出了许多宝贵意见，对提高本书质量大有帮助，在此表示衷心感谢。

本书由国家自然科学基金项目（50969003）和科技部国际科技合作项目（2010DFA71460）联合资助出版。

由于我们水平有限，错误和不妥之处在所难免，恳请读者批评指正。

编者

2011 年 3 月

第一版前言

随着工程技术领域的计算过程日益复杂和要求快速化，计算机已成为解决工程问题不可缺少的有力工具，与计算机应用密切相关的“数值计算方法”成为工科学生的必修课程。本教材正是为适应这种变化，并遵照水文与水资源工程、农业水利工程本科专业教学规范中提出的构造知识体系框架原则及水文与水资源工程专业教学组拟定的“《数值计算方法》教材编写大纲”编写的。

本教材在满足水文与水资源工程、农业水利工程等专业基本需要的基础上，以算法的构造和应用为重点。在内容安排上，力求介绍各类数值问题常用的数值计算方法，为适应水文与水资源工程专业本科和研究生教学的需要，设置了常用的最优化方法和人工智能计算方法，为提高学生分析数据和解决问题的能力，书中还加强了插值、拟合、平滑及逼近等内容；为适应大型水利工程计算，在线性代数方程组数值方法中，介绍了关于特殊矩阵方程组的存储与求解的内容。各章节的阐述方式力求概念明确、简练易懂、由浅入深、自成体系。为了便于学生了解算法的原理和执行过程，大部分算法都附有程序框图和程序代码，学生通过对算法程序代码的编译和运行，能够加深对算法的理解，从而有效地培养学生计算机高级语言表达算法的能力。

本书第一、五、六、七、八章由高新科编写，第二章由高新科、朝伦巴根编写，第三、四、九章由朝伦巴根编写，第十章由高瑞忠编写。第一、四、九章的程序由朝伦巴根、李凤玲编制，第二、五、六、七章程序由贾德彬、李凤玲编制，第三、八章程序由刘廷玺编制，第十章程序由高瑞忠编制。书中所有程序由贾德彬调试，全书由朝伦巴根统稿。

本书由裴喜春教授主审，李畅游教授校阅了书稿，他们在审核过程中提出了许多宝贵意见，对提高本书质量大有帮助，在此表示衷心感谢。

本书由国家自然科学基金重点项目（50139040）和内蒙古自然科学基金重大项目（200408020301）资助。

由于我们水平有限，错误和不妥之处在所难免，恳请读者批评指正。

编者

2006 年 9 月

目　录

第一章　引　　论

工程技术人员在解决实际问题时，经常需要建立一个与之等价的数学模型，并通过计算工具求解这一模型。过去利用手算或计算器，由于受到计算能力的限制，只能解决一些简单而规模小的问题。随着计算机的迅速发展和普及，许多复杂的工程问题得以圆满解决。这是因为在计算机上可以采用更为精确的模型和更为有效的算法，能够处理多达数千个、数万个变量的大系统。目前计算机已经成为进行工程问题数值分析的强有力的工具。

计算机的广泛应用不仅激起工程技术人员对数值方法及计算机数值分析的巨大兴趣，也促进了数值分析本身的发展。作为导论，本章介绍计算机数值分析的特点与过程、数值方法的概念以及数在计算机中的近似表示。最后简要讨论算法的稳定性和数值问题的适定性。

第一节　数值分析与计算机

数值分析是应用数学的一个重要组成部分，也是计算机应用的基础学科。它主要研究数学问题的数值解法（即数值方法），包括方法的推导、描述和精度评价；研究数值方法在实际问题中的应用及整个求解过程。因此，数值分析贯穿于求解过程的始终。

数值分析的基础是数学模型，即通过数学模型，对实际系统或工程技术问题进行数学描述，将实际问题转变为数值计算问题。其目的就是运用数学方法，以一定的计算工具，研究和处理数值问题。

长久以来，人们以简单的计算手段进行数值分析，从而使用的数学方法是简化了的，处理的对象和研究的范围也是很有限的。电子计算机作为一种高速、自动的信息处理系统，出现于 20 世纪 40 年代末。经历了 60 多年的迅猛发展，对整个科学技术产生了变革性的影响，特别是对数值分析给予强有力的冲击和推动。它使新的数值方法层出不穷，也使过去因计算能力所限而埋没了的方法重新复活。广泛应用于各类工程问题的有限单元法的问世和完善与计算机的出现和发展是分不开的。通过数学模型借助计算机进行数值分析的领域日益扩大（如应用于核反应堆、空间探索、海洋油田的开发、长期天气预报等），而且已成为工程设计和运行管理的有效手段。所以，掌握完整的数值分析的知识，并将其应用于所处理的问题中，将会有效地解决问题。

计算机数值分析有哪些特点呢？首先，因为计算机本身只能执行简单的离散运算（加、减、乘、除）和基本的按位逻辑操作（逻辑加、乘、非和移位等），因此在使用计算机之前，必须实现问题的离散化，即使模型方程及求解区域离散化（如用有限差分法或有限单元法把无限多个自由度的连续体模型，变换为有限个自由度的离散模型，即数值模

型）和使初始数据离散化（如以有限个离散点上的函数值或采样值，代替连续函数或光滑曲线所给定的值）。这样，计算机才有能力处理。从某种意义上讲，从连续性、光滑性的概念过渡到离散性，是计算机数值分析的一个基本特征。实现模型方程、求解区域及解法的离散化，为问题数值求解提供了必要的前提。

计算机数值分析的第二个特点就是实现算法的代码化，即完成程序设计。程序设计是把计算过程变换为一系列能使计算机识别并予自动执行的代码的过程。对工程师来说，使用高级程序设计语言（如 FORTRAN、COBOL、PASCAL、BASIC 和 C 语言等）是一种简便的编程手段。凭借这些语言，用户可以用代数公式、英语形式的逻辑判断以及循环、输入、输出等基本语句编写程序。用户的程序将由计算机所配置的编译系统或解释系统转换为机器可以执行的基本代码（即机器代码）。

对计算结果的分析验证是数值分析的另一个关键步骤。首先，必须保证数据的正确性与精度。俗话说："送进去的是垃圾，出来的仍是垃圾"。在排除了程序的语法错误和逻辑错误之后，需要判断是否得到了正确结果。由于问题的正确解答不会预先知道，对结果的分析验证通常是间接的，如查看极限状态下的解，或对比计算值与实测值等。这些对比性试算所花费的工时、机时和费用通常是编程阶段所花费的好多倍，但这是值得的。对结果的分析验证往往引起对模型、算法和程序的反复修正和完善，直到获得正确结果为止。计算机数值分析的基本过程见图 1－1。

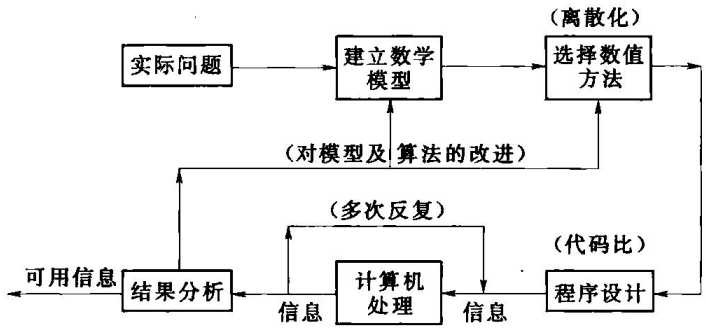

图 1－1 计算机数值分析的基本过程

第二节 数值方法的概念

数值方法是在解决数值问题的长期过程中形成的。它的产生和应用比较早，有许多方法在现代数学的萌芽期就形成了。计算机的出现才使其面目一新，有了更迅猛的发展。

众所周知，多数数学问题是不能用解析方法精确求解的，除非问题比较简单或是简化了的。如对方程

$$ax^2 + bx + c = 0$$

有精确解

$$x_1 = (-b + \sqrt{b^2 - 4ac})/2a$$

$$x_2 = (-b - \sqrt{b^2 - 4ac})/2a$$

但对形式上略作变化的方程

$$a\sin^2 x + bx + c = 0$$

就没有可用的精确公式了。这就需要研究和使用近似解法。给出问题的数值近似解的算法就是数值方法（或计算方法）。数值方法大多是为应用计算机而建立的，因此也称之为计算机数值方法。数值方法的特点如下：

（1）数值方法得到的解总是精确解的近似值，即数值解。

（2）数值方法是用基本算术运算给出的。

（3）它们都能直接用于计算机。

数值分析的成败或解的精确度在很大程度上取决于所用数值方法的优劣。应尽可能好的选择数值方法。数值方法的选择一般应遵从以下原则：

（1）方法应有较高精度。

（2）方法的有效性好，即为达到给定精度所需要的算术运算的次数要少。这样，计算时间就短。

（3）方法应是稳定的。实际问题中的数据总是有误差的，计算中的舍入误差也是难免的。不稳定算法将使舍入误差严重积累，导致错误结果。

（4）方法应节省计算机资源。这里主要指占用计算机的内存容量要少。

尽管数值方法与计算机结合，形成了强有力的数值分析工具，但并非对任何问题都有效。有的问题难于建立数学模型；有的问题过于复杂，超越了当前的计算技术水平。数值方法对这两类问题均无能为力。近年来，为解决各种工程问题，已出现了大量功能齐全的程序和软件包，随着计算机和软件技术的日益发展，这种软件产品会越来越多。也许有人以为，只要从这些软件产品中收集到自己需要的程序，送入数据就能从计算机得到结果，学习数值方法似无必要。这是一种误解。事实上，程序在运行中，会出现各种意想不到的问题。有相当一部分问题是由于程序选择不当和数值方法运用有误引起的，致使计算结果错误或全然没有结果。为了高质量、高效率地运用数值分析解决工程技术问题，学习数值方法和掌握数值分析的技巧，对工程师来说无疑是十分必要的。在学习了数值方法的基本知识后，就能自如地为具体问题选择或改进数值方法，就能从现成的软件包中选择适用的程序，甚至自编程序。有不少人会成为软件开发的能手。事实上，当今不少优质的大型应用软件并不是出自计算机科学家或数值分析专家之手，而是由其它专业的工程技术人员做出的。

第三节　误差及数在计算机内的近似表示

一、误差的有关概念

（一）绝对误差与相对误差

数值分析是利用计算机对数据信息加工处理的过程。一般说，数据总是有误差的，计算结果也是有误差的。设 x 为准确值，x^* 为 x 的近似值，则称

$$E(x^*) = x - x^* \tag{1-1}$$

为近似值 x^* 的绝对误差，简称误差。通常 $E(x^*)$ 的准确值很难求出，但常可估计出 $E(x^*)$ 的绝对值的某个上界 ϵ，即有

$$|E(x^*)|=|x-x^*|\leqslant\varepsilon \tag{1-2}$$

称 ε 为绝对误差限。这样，真值 x 的范围就是

$$x^*-\varepsilon\leqslant x\leqslant x^*+\varepsilon \tag{1-3}$$

亦即

$$x=x^*\pm\varepsilon \tag{1-4}$$

例1 用天平称物重，读数是 155.094g，天平的精度是 0.001g，问物重是多少？

已知物重 W 的近似值 $W^*=155.094\text{g}$，天平的精度 $\varepsilon=0.001\text{g}$，由式（1-4），则物重

$$W=W^*\pm\varepsilon=155.094\text{g}\pm0.001\text{g}$$

误差限自然越小越好，但绝对误差不足以说明近似值的准确程度。因为 1000g 重物体的 $E(x^*)$ 为 0.001g 与 10g 重物体的 $E(x^*)$ 也是 0.001g 就不能相提并论。为弥补这个缺陷，需用相对误差。称

$$R(x^*)=E(x^*)/x=(x-x^*)/x \tag{1-5}$$

为 x^* 的相对误差。设其绝对值上界为 ε_r，则有

$$|R(x^*)|=|x-x^*|/|x|\leqslant\varepsilon_r \tag{1-6}$$

式中 ε_r 为 x^* 的相对误差限。因 x 的真值难于知道，当 ε_r 很小时，可取

$$R(x^*)=\frac{E(x^*)}{x^*}=\frac{x-x^*}{x^*} \tag{1-7}$$

可以看到，相对误差说明 x^* 的绝对误差与 x^* 本身相比所占的比例，进一步刻画了 x^* 的准确程度，而且，绝对误差通常是有量纲的，相对误差则没有。

（二）有效数字

我们也可以用有效数字说明近似值的准确程度。设 x^* 是 x 的十进制近似值，如其误差绝对值不超过 x 的第 K 位小数的半个单位，亦即

$$E(x^*)=|x-x^*|\leqslant0.5\times10^{-K} \tag{1-8}$$

则称 x^* 准确到小数点后第 K 位。从第 K 位数字到最左边非零数字间的所有数字都叫有效数字。

例2 若真值 $x=0.98632$ 的近似值 $x^*=0.9864$，由于 $|x-x^*|=0.0008<0.5\times10^{-3}$，因此，$x^*$ 具有三位有效数字：9，8，6。又如 $x=-0.00460918$ 的近似值 $x^*=-0.00460929$，则 $|x-x^*|=0.00000011<0.5\times10^{-6}$，所以 x^* 有四位有效数字：4，6，0，9。

需要提及的是，如果 x^* 是由 x 用通常的四舍五入原则得到的近似值，根据四舍五入的定义，则从被保留的最后一位起直到 x^* 最左边非零数字之间的所有数字都是有效数字。

例3 用四舍五入原则将 $\pi=3.14159265\cdots$ 舍入成 3.14，3.142，3.1416，3.14159 等近似值。容易看出，舍入误差的绝对值均不超过被保留的最后数位的半个单位，例如，$|\pi-3.14|<0.5\times10^{-2}$，因此，各个近似值的所有数字都是有效数字。

又如，按四舍五入原则写出下述各数的有五位有效数字的近似数：237.4523，0.03974412，5.000046，1.6282768。

按定义，则上述各数的具有五位有效数字近似数分别是：237.45，0.039744，

5.0000，1.6283，注意 5.000046 的五位有效数字近似值是 5.0000 而不是 5，因为 5 只有一位有效数字。

在计算中，应尽可能多地保留数据中的有效数字，因为有效数字越多，相对误差就越小，计算就越准确。

造成误差的原因是多方面的。有数学模型的近似造成的模型误差；有参数测定不准确产生的观测误差；有近似算法引起的截断误差及运算中出现的舍入误差。在讨论数值方法时，尽管上述四种误差可能同时存在，相互影响，但为了分清主次，简化讨论，主要考虑截断误差与舍入误差。

（三）截断误差与舍入误差

如欲计算 e^{-x} 的值，用其泰勒（Taylor）级数

$$e^{-x}=1-x+\frac{x^2}{2!}-\frac{x^3}{3!}+\cdots \tag{1-9}$$

计算时，若取其前四项，即为

$$e^{-x}=1-x+\frac{x^2}{2!}-\frac{x^3}{3!} \tag{1-10}$$

可以证明，由于舍弃其它的项而引起的截断误差为

$$E(x)=\frac{1}{24}e^{-\xi}x^4,\xi\in(0,x) \tag{1-11}$$

$E(x)$ 反映了泰勒三次多项式对 e^{-x} 的逼近程度，其量值是一个与 x^4 成正比的常数，记为 $E(x)=O(x^4)$。

计算机字长是有限的。任何实数都得取字长以内的位数参与运算，从而产生舍入误差。

设有无穷十进小数

$$x=x_1 10^{-1}+x_2 10^{-2}+\cdots+x_s 10^{-s}+\cdots \tag{1-12}$$

其中 x_s（$s=1$，2，…）是 0 到 9 中的任一个数。欲将 x 舍入为 s 位小数，一种方法是将 x 的 $s+1$ 位及其后的所有位均舍弃，得到

$$x^*=x_1 10^{-1}+x_2 10^{-2}+\cdots+x_s 10^{-s} \tag{1-13}$$

其绝对（舍入）误差限显然是

$$|x-x^*|<10^{-s} \tag{1-14}$$

第二种办法是通常的四舍五入原则，即

$$\left.\begin{aligned} x^*&=x_1 10^{-1}+x_2 10^{-2}+\cdots+x_s 10^{-s}, && \text{当 } x_{s+1}<5 \text{ 时}\\ x^*&=x_1 10^{-1}+x_2 10^{-2}+\cdots+(x_s+1)10^{-s}, && \text{当 } x_{s+1}\geqslant 5 \text{ 时}\end{aligned}\right\} \tag{1-15}$$

其绝对（舍入）误差限显然是

$$|x-x^*|\leqslant 0.5\times 10^{-s} \tag{1-16}$$

以后提到舍入误差，就是指第二种舍入方法。

二、数在计算机内的近似表示

几乎所有计算机都是用浮点数近似表示任一实数的。实数 x 的浮点数定义为

$$fL(x)=\pm\left(\frac{d_1}{\beta}+\frac{d_2}{\beta^2}+\cdots+\frac{d_t}{\beta^t}\right)\beta^e \tag{1-17}$$

其中整数 d_i（$i=1, 2, \cdots, t$）满足

$$0 \leqslant d_i \leqslant \beta-1$$

整数 e 满足 $L \leqslant e \leqslant U$，这里 L、U 分别为负、正整数。

数的这种表示形式，在工程和自然科学中是常见的。为了表示很小或很大的十进制数，常将该数写成一个小数乘上以 10 为底的幂的形式。例如，将 15460000 写为 $(0.1546)\times10^8$，将 -0.00001546 写为 $-(0.1546)\times10^{-4}$，只是这里 $\beta=10$，$t=4$，$e=8$ 及 -4，$d_i=1, 5, 4, 6$。

在式（1-17）中 β 称为数基（$\beta=2$，是二进制；$\beta=8$，是八进制；$\beta=10$，为十进制等），e 为阶数，L 和 U 是 e 的上、下界。显然，$fL(x)$ 的小数点的位置随 e 值的变化而浮动，故称 $fL(x)$ 是 x 的浮点数。$f=\pm\left(\frac{d_1}{\beta}+\frac{d_2}{\beta^2}+\cdots+\frac{d_t}{\beta^t}\right)=\pm 0.d_1d_2\cdots d_t$ 是 $fL(x)$ 的尾数。对非零的 x，如 $d_1\neq0$，称 $fL(x)$ 为 x 的规格化浮点数。t 是浮点数的有效位数，也是计算机保存尾数的最大位数，称为计算机的字长。数在计算机内被自动转变为式（1-17）的形式。其阶数 e 及尾数 f 被分别存储起来。

四个参数 β，t，L，U 决定了计算机上的浮点数集合 $\boldsymbol{F}$（β，t，L，U）。不同的计算机对浮点数参数有不同的规定，例如，PDP-11 计算机的 $\beta=2$，$t=24$，$L=-128$，$U=127$；IBM360/370 计算机的 $\beta=16$，$t=14$，$L=-64$，$U=63$；Texas Instruments SR-SX 计算机的 $\beta=10$，$t=12$，$L=-98$，$U=100$ 等。这些参数取决于计算机的系统设计。但就某一计算机而言，只有有限集合 $\boldsymbol{F}$（β，t，L，U）中的数才能被机器所接受和处理。凡不属于 $\boldsymbol{F}$ 的数，不管是初始数据还是运算结果，均舍入为 $\boldsymbol{F}$ 中的最接近该数的浮点数 $fL(x)$。如数 x 的绝对值大于 $\boldsymbol{F}$ 中的最大正数，机器就将 x 按无穷大处理，终止运行（上溢）；如 x 的绝对值小于 $\boldsymbol{F}$ 中的最小正数，机器就令 x 为零参与运算（下溢）。

综上所述，近似数 $fL(x)$ 实际上就是计算机对 x 舍入的结果。其误差限为最后保留数位的半个单位，即

$$|x-fL(x)|\leqslant\frac{1}{2}\beta^{-t}\beta^e=\frac{1}{2}\beta^{(e-t)} \tag{1-18}$$

其相对误差则是

$$\frac{|x-fL(x)|}{|x|}\leqslant\frac{0.5\beta^{e-t}}{0.1\beta^e}=\frac{1}{2}\beta^{(1-t)} \tag{1-19}$$

可见绝对误差限与 e、t 有关。固定 t，则阶数 e 越大，其近似值 $fL(x)$ 的误差限越大。反之，阶数 e 越小，其近似值 $fL(x)$ 的误差限越小。但相对误差限仅与机器字长 t 有关。为方便计，可令 $\varepsilon=\frac{1}{2}\beta^{(1-t)}$，则对实数 x，有

$$fL(x)=x(1+\eta) \tag{1-20}$$

其中 $|\eta|\leqslant\varepsilon$，通常称 ε 为计算机的相对精度。显然，t 越大，机器精度越高。

[redacted] 试说明浮点数集合 $\boldsymbol{F}$（10，6，-98，100）的组成，并判断下列各数 $x_1=0.457281\times10^{-5}$，$x_2=-0.722864\times10^{99}$，$x_3=0.1122334\times10^0$，$x_4=0.458230\times10^{101}$，$x_5=0.\overline{5}\,\overline{4}5201\times16^3$ 及 $x_6=999999\times10^{98}$ 是否属于集合 $\boldsymbol{F}$？计算机如何处理它们（注：x_5

中的$\bar{5}$代表 15，$\bar{4}$代表 14)?

解 已知 F 对应的参数是$\beta=10$，$t=6$，$L=-98$，$U=100$，根据浮点数的定义，易知 F 是由下面三个集合中的数组成：$\{-0.999999\times10^{100}, -0.000001\times10^{-98}\}$，$\{0\}$ 及 $\{0.000001\times10^{-98}, 0.999999\times10^{100}\}$。根据集合 F 的组成，容易判断

$x_1=0.457281\times10^{-5}\in F$（机器接受并予处理）。

$x_2=-0.7228674\times10^{99}\in F$（机器接受并予处理）。

$x_3=0.1122334\times10^{0}\notin F$［尾数位数大于 6，按 $fL(x_3)=0.112233\times10^{0}$ 存储和处理］。

$x_4=0.458230\times10^{101}\notin F$（阶数大于 100，机器按无穷大处理，出现“上溢”停机）。

$x_5=0.\bar{5}\,\bar{4}5201\times16^{3}\notin F$（数基 $\beta\neq10$，故机器不予接受，按出错处理）。

$x_6=999999\times10^{98}\notin F$(数值大于 F 中的最大正数,机器按无穷大处理,出现“上溢”停机)。

例 5 试写出 $x=100.1$ 的十进制（$\beta=10$，$t=6$），二进制（$\beta=2$，$t=16$）及八进制（$\beta=8$，$t=6$）的浮点数及相对误差。

解 (1) $\beta=10$，$t=6$，则

$$fL(x)=\left(\frac{1}{10}+\frac{0}{10^2}+\frac{0}{10^3}+\frac{1}{10^4}+\frac{0}{10^5}+\frac{0}{10^6}\right)\times10^3$$
$$=0.100100\times10^3$$

相对误差 $R\leqslant\frac{1}{2}\times10^{-5}$

(2) $\beta=2$，$t=16$，则

$$fL(x)=\left(\frac{1}{2}+\frac{1}{2^2}+\frac{0}{2^3}+\frac{0}{2^4}+\cdots+\frac{1}{2^{15}}+\frac{1}{2^{16}}\right)\times2^7$$
$$=0.1100100000110011\times2^7$$

相对误差 $R\leqslant\frac{1}{2}\times2^{-15}$

(3) $\beta=8$，$t=6$，则

$$fL(x)=\left(\frac{1}{8}+\frac{4}{8^2}+\frac{4}{8^3}+\frac{0}{8^4}+\frac{6}{8^5}+\frac{3}{8^6}\right)\times8^3$$
$$=0.144063\times8^3$$

相对误差 $R\leqslant\frac{1}{2}\times8^{-5}$

第四节 算 法 的 稳 定 性

前面把稳定性作为选择算法的标准之一。算法可定义为数值问题解法的一个完备而确切的描述。在本书中，算法与数值方法是两个等价的概念。为了保证计算结果的可靠性，必须使用舍入误差影响小的算法，即**稳定的算法**。

下面的例子说明一个不稳定的算法怎样得出令人失望的结果；改变算法又如何得到满意的解答。

例 1 试计算 $e^{-5.5}$ 的值。

解 对任意实数 x，e^x 可按泰勒公式表示为

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \tag{1-21}$$

在上式中取 $x=-5.5$，并以浮点数的五位有效数字（即 $\beta=10$，$t=5$）计算，有

$$\begin{aligned} e^{-5.5} = {} & 0.10000 \times 10^1 - 0.55000 \times 10^1 + 0.15125 \times 10^2 \\ & - 0.27730 \times 10^2 + 0.38129 \times 10^2 - 0.41942 \times 10^2 \\ & + 0.38446 \times 10^2 - 0.30208 \times 10^2 + 0.20768 \times 10^2 \\ & - 0.12692 \times 10^2 + 0.69803 \times 10^1 - 0.34902 \times 10^1 \\ & + 0.15997 \times 10^1 + \cdots + 0.94623 \times 10^{-6} \\ = {} & 0.26363 \times 10^{-2} = 0.0026363 \end{aligned} \tag{1-22}$$

结果是在计算到第 25 项后得到的。以后的项对五位有效数字的项已无影响。实际上，$e^{-5.5}$ 的真值是 0.00408677。因此，式（1-22）给出的是错误结果。

为什么会出错呢？可以看到，和式中的一些项及中间和的量级比最终结果的量级大得多。数据的有效位数偏少，使数据中包含了较大的误差。例如，第五个数据项 38.129 的舍入误差限是 0.005，同最终结果的量级差不多。另外，在计算中有效数字严重丢失（通常称之为“灾难性抵消”），致使计算结果同准确结果差之甚远。当然，把数据项的有效位数取多些，将会使结果得到改善，但最根本的办法是改变算法，如按

$$\begin{aligned} e^{-5.5} &= \frac{1}{e^{5.5}} \\ &= \frac{1}{1 + 5.5 + 15.125 + \cdots} \\ &= 0.0040865 \end{aligned} \tag{1-23}$$

进行计算，结果就会相当准确，其相对误差

$$\begin{aligned} R &= \frac{0.00408677 - 0.0040865}{0.0040865} \\ &= 0.000066 \end{aligned}$$

算式（1-23）利用同样的数据，却使结果的相对误差减少到 0.007%。

例 2 计算下述积分的值

$$E_n = \int_0^1 x^n e^{x-1}\,\mathrm{d}x,\quad n = 1,2,\cdots \tag{1-24}$$

解 由分部积分法得

$$\int_0^1 x^n e^{x-1}\,\mathrm{d}x = x^n e^{x-1}\,\Big|_0^1 - \int_0^1 n x^{n-1} e^{x-1}\,\mathrm{d}x$$

或

$$E_n = 1 - nE_{n-1},\quad n = 2,3,\cdots \tag{1-25}$$

其中 $E_1 = \frac{1}{e}$，且任何 $E_n > 0$。

现以浮点数六位有效数字（即 $\beta=10$，$t=6$）按式（1-25）计算，得到

$$E_1 \approx 0.367879,\quad E_2 \approx 0.264242,$$

$$E_3 \approx 0.207274,\quad E_4 \approx 0.170904,$$
$$E_5 \approx 0.145480,\quad E_6 \approx 0.127120,$$
$$E_7 \approx 0.110160,\quad E_8 \approx 0.118720,$$
$$E_9 \approx -0.0684800。$$

在这里 E_9 的计算值为负，与任何 $E_n>0$ 相矛盾。什么原因引起这样大的误差？由分部积分得到的递推公式是精确的，如不计中间计算过程的误差，唯一的原因就是 E_1 按 6 位有效数字舍入，在计算中引入了初始误差 $\Delta E_1 \approx 0.3678794412-0.367879=4.412\times 10^{-7}$。由式（1-25），$E_9$ 的误差 $\Delta E_9=(-2)\times(-3)\cdots\times(-9)\times\Delta E_1=9!\times 4.412\times 10^{-7}\approx 0.1601$。而 E_9 的准确值应是 $E_9+\Delta E_9=-0.06848+0.1601=0.0916$。舍入误差这样严重的被积累、扩大，显然是由算法［式（1-25）］本身引起的。

需要说明的是，前面假定了只在计算开始时才有舍入误差 ΔE_1，以后各步计算误差均不计入。这是为了简化讨论。可以设想，如果第一步舍入引起的初始误差对计算产生了很大影响，那么每步的舍入将造成更大的影响；反之，如果第一步的舍入误差在计算中得到控制，或逐步减少，则以后各步的舍入误差也理应会得到控制或削弱。

如何构造另一个算法，来避免这种数值不稳定性呢？若将递推公式改写为

$$E_{n-1}=(1-E_n)/n,\qquad n=\cdots,3,2 \tag{1-26}$$

在每步计算中，E_n 的误差以 $\frac{1}{n}$ 的因子减少，因此，从某个 E_n（$n\geqslant 1$）开始，沿 n 减小的方向计算，初始（舍入）误差就会受到控制，逐次减少。显然，式（1-26）是一种稳定算法。考虑到

$$E_n=\int_0^1 x^n e^{x-1}\mathrm{d}x \leqslant \int_0^1 x^n \mathrm{d}x \tag{1-27}$$

这是因为在［0，1］上，$e^{x-1}\leqslant 1$。由式（1-27），则

$$E_n \leqslant \frac{x^{n+1}}{n+1}\Big|_0^1=\frac{1}{n+1} \tag{1-28}$$

故当 $n\to\infty$ 时，$E_n\to 0$，如取 $E_{20}\approx 0$，用此值作为按式（1-26）计算的开始值，则引入的初始舍入误差 ΔE_{20} 充其量为 1/21，即 0.0476。进行上述计算，得到如下结果。

$$E_{20}\approx 0.0,\qquad E_{14}\approx 0.0627322,$$
$$E_{19}\approx 0.0500000,\qquad E_{13}\approx 0.0669477,$$
$$E_{18}\approx 0.0500000,\qquad E_{12}\approx 0.0717733,$$
$$E_{17}\approx 0.0527778,\qquad E_{11}\approx 0.0773523,$$
$$E_{16}\approx 0.0557190,\qquad E_{10}\approx 0.0838771,$$
$$E_{15}\approx 0.0590176,\qquad E_9\approx 0.0916123。$$

由式（1-26）及式（1-27），E_{19} 的误差 $\Delta E_{19}\leqslant\frac{1}{20}\times\frac{1}{21}\approx 0.0024$，计算到 E_{15} 时，初始误差已减少到 4×10^{-8}，已远远小于六位有效数字的舍入误差。所以，从 E_{15} 到 E_9 都是具有六位有效数字的正确结果，其可能的舍入误差不超过最后一个数字位的半个单位。

上述两例说明了不同算法对舍入误差的灵敏程度不一。计算中舍入误差的增长能得以控制，对计算结果影响不大的算法，称为**稳定的算法**，例如式（1-23）和式（1-26）；

否则便是**不稳定的算法**，如式（1－21）及式（1－25）。

尽管本书给出的都是稳定的算法，但在实际应用中，不同问题会产生不同情况。当对所用算法的稳定性知之甚少时，或计算中出现了不稳定的迹象时，可用试验比较的方法进行处理。例如，用不同的数据组合或不同的算法试算，比较这些结果，以了解舍入误差的影响，确定结果的可靠性。在一般情况下，这种实用方法是可行而有效的。

第五节 数值问题的适定性

有一类数值问题不管使用何种算法，都得不到正确结果。这种性质与所用的算法和浮点数舍入运算均无关，而是由于问题本身对数据的微小变化非常敏感所致。为了说明适定性的概念，现考虑下面两个例子。

例 1 考查二次方程

$$(x-2)^2 = 10^{-6} \tag{1-29}$$

求根问题对数据变化的反映情况。

解 方程的根应是 2 ± 10^{-3}。可以看到，方程右端常数项 10^{-6} 的一个微小变化，将引起根的 10^{-3} 量级的变化。两者在量级上有很大差异。所以说，该式对数据变化的反应相当灵敏。即使没有用受扰动的常数参加运算，仅由舍入而产生的误差也使问题得不到正确的解答。

例 2 考虑如下多项式的零点。

$$\begin{aligned} p(x) &= (x-1)(x-2)\cdots(x-19)(x-20) \\ &= x^{20} - 210x^{19} + \cdots \end{aligned} \tag{1-30}$$

解 $p(x)$ 的根或零点应是 1，2，…，19，20。但如使 x^{19} 的系数稍作变化，例如由 -210 变为 $-210+2^{-23}$，多项式的零点会有多大变化呢？这相当于求解方程 $p(x)+2^{-23}x^{19}=0$ 的根。有人（Wilkinson，1963）用 $\beta=2$，$t=90$ 在高精度计算机上作了仔细的验算，得到了如下结果：

1.00000 0000，　　2.00000 0000，
3.00000 0000，　　4.00000 0000，
4.99999 9928，　　6.00000 6944，
6.99969 7234，　　8.00726 7603，
8.91725 0249，　　20.84690 8101，
$10.09526\ 6145 \pm 0.64350\ 0904i$，
$11.79363\ 3881 \pm 1.65232\ 9728i$，
$13.99235\ 8137 \pm 2.51883\ 0070i$，
$16.73073\ 7466 \pm 2.81262\ 4894i$，
$19.50243\ 9400 \pm 1.94033\ 0347i$。

注意到，x^{19} 系数的微小变化已使方程第 10～第 20 根发生剧烈变化：第 10 根增大一倍多，第 11～第 20 根由实数变为复数。上述结果并非舍入误差及算法所致，而是由于问