研究生数学基础课程系列教材

曾金平 主编

数值计算方法

ShuZhi Jisuan Fangfa

ShuZhi Jisuan Fangfa

湖南大学出版社

研究生数学基础课程系列教材

数 值 计 算 方 法

主　编　曾金平
副主编　李郴良

湖南大学出版社
2004年·长沙

内 容 简 介

本书可作为大学数学、力学和计算机等专业的“计算方法”教材以及理工科硕士研究生的“数值分析”教材.本教材介绍计算机上常用的数值计算方法,主要包括非线性方程求根、线性代数方程组直接法和迭代法、插值逼近、拟合逼近、数值微积分和常微分方程数值解等内容.全书深入浅出,层次分明,部分理论证明和全书内容独立,便于根据不同学时和要求进行取材和教学.

图书在版编目(CIP)数据

数值计算方法/曾金平主编.—长沙:湖南大学出版社,
2004.8
ISBN 7-81053-827-6
Ⅰ.数… Ⅱ.曾… Ⅲ.数值计算—计算方法
—研究生—教材 Ⅳ.O241
中国版本图书馆 CIP 数据核字(2004)第 079026 号

数值计算方法

Shuzhi Jisuan Fangfa

主　　编:曾金平
责任编辑:厉　亚
特邀编辑:彭亚新
封面设计:张　毅
出版发行:湖南大学出版社
社　　址:湖南·长沙·岳麓山　　**邮　　编**:410082
电　　话:0731-8821691(发行部),8649149(编辑室),8821006(出版部)
传　　真:0731-8649312(发行部),8822264(总编室)
电子邮箱:press@hnu.net.cn
网　　址:http://press.hnu.net.cn
印　　装:长沙环境保护学校印刷厂
总 经 销:湖南省新华书店
开本:720×960　16 开　　**印张**:18　　**字数**:314 千
版次:2004 年 8 月第 1 版　**印次**:2004 年 8 月第 1 次印刷　**印数**:1～5 000 册
书号:ISBN 7-81053-827-6/O·54
定价:30.00 元

前　言

随着计算机技术的迅速发展与广泛应用，大大促进了计算数学的飞跃发展.同时，随计算数学和计算机技术的发展而日益兴起的计算科学(计算数学与仿真、图像处理、统计分析等)已经深入渗透于自然科学、工程技术、经济管理以至人文科学各个领域，并成为继牛顿与伽利略创立的理论研究与科学实验两大科学方法后的第三种科学方法.

为适应新的形势，各高等院校对数学、力学和计算机等理科本科专业的学生开设了“计算方法”课程，对理工科硕士研究生开设了“数值分析”课程.

本书介绍了计算机上常用的数值计算方法，主要包括非线性方程求根、线性代数方程组直接法和迭代法、插值逼近、拟合逼近、数值微积分和常微分方程数值解等内容.全书深入浅出，层次分明，部分理论证明和全书内容独立，便于根据不同对象、学时和要求进行取材和教学.

本书适合大学数学、力学和计算机等理科专业的本科生，以及理工科各个专业的硕士研究生使用，也可供从事科学计算的科技工作者参考.

本书是根据我们在湖南大学多年的教学经验编写的.在教材的编写过程中，信息与计算科学专业赵旭鹰、李庆娜等学生和我们一起讨论，提出了很好的意见并对全书进行了仔细的校对工作，在此我们深表感谢.本书难免会有不妥之处，希望使用本书的教师和学生提出宝贵意见，指出其错误和不足之处，以便进一步改进.

作　者

2004 年 5 月

目　次

第一章　引　言 …… 1
第一节　数值计算方法及其主要内容 …… 1
第二节　误差及误差分析 …… 5
第三节　算法的稳定性 …… 14
习题一 …… 18
第二章　非线性方程求根 …… 20
第一节　二分法 …… 20
第二节　简单迭代法 …… 23
第三节　牛顿迭代法及其简单变形 …… 31
习题二 …… 43
第三章　求解线性代数方程组的数值方法 …… 45
第一节　高斯(Gauss)消元法 …… 46
第二节　矩阵的三角分解及其在解线性代数方程组中的应用 …… 54
第三节　线性代数方程组的性态与误差分析 …… 67
第四节　迭代法 …… 80
第五节　共轭梯度法 …… 98
习题三 …… 110
第四章　插值逼近 …… 114
第一节　Lagrange 插值 …… 115
第二节　差商、差分与 Newton 插值 …… 127
第三节　Hermite 插值 …… 135
第四节　分段多项式插值 …… 140
习题四 …… 152
第五章　最佳逼近 …… 155
第一节　离散最小二乘逼近 …… 155
第二节　最佳平方逼近 …… 172
第三节　最优一致逼近 …… 184
第四节　其它类逼近问题 …… 197
习题五 …… 218

第六章 数值微积分…… 221
第一节 数值微分…… 221
第二节 Newton-Cotes 求积公式 …… 229
第三节 龙贝格(Romberg)求积法…… 246
第四节 高斯(Gauss)型求积公式 …… 248
第五节 奇异积分的计算…… 256
习题六…… 265
第七章 常微分方程数值解初步…… 267
第一节 常微分方程初值问题数值解法…… 267
第二节 解常微分方程边值问题的差分法…… 273
习题七…… 275

习题答案…… 276
参考文献…… 281

第一章 引 言

第一节 数值计算方法及其主要内容

数学是研究数与形的科学，而计算数学是数学的一个分支，通常也称为数值分析或数值计算方法，它是研究如何利用计算工具（如计算器、计算机等）求出数学问题的数值解答（如数据、表格、图形等）的学问. 由此可见，计算数学的前身可追溯到人类文明萌芽时期的算学和测绘学，是数学中最古老的一部分. 但是，由于计算工具的笨拙和数值计算的繁杂，长期制约了计算数学的发展. 随着 20 世纪 40 年代电子计算机的出现以及现代科学与工程中大规模科学计算的迫切需求，计算数学获得了前所未有的发展. 半个多世纪以来，计算数学已经发展成为现代意义下的计算科学，成为继牛顿-伽利略（Isaac Newton，英国，1642～1727；Galileo Galilei，意大利，1564～1642）理论研究和科学实验两大科学方法之后的第三大科学方法，并深入到各个学科领域的方方面面，扮演着越来越重要的角色.

许多科学计算问题来源于科学与工程，其目的在于理解一些自然现象、解决工程实际问题或进行最优设计等. 在计算科学中，通过计算机重造或模拟物理系统或过程，即计算仿真，能够大大增强人们对物理系统或过程的认知力，尤其是那些通过理论、观测、实验等手段难于观察到的系统或过程. 例如，在天文学中，由于过程的复杂性，很难用理论描述两个碰撞黑洞的具体形态，也不可能直接观察到，更不可能在实验室中进行重造，而通过计算仿真可达到我们的目的. 它所需要的只是一个适当的数学模型（如广义相对论中的 Einstein 方程组）、求解该数学模型的数值计算方法和用来实现相应数值计算方法的计算速度足够快且内存足够大的计算机. 由此可见，计算数学不同于数学学科的其它分支，它不能单独存在而必须依托现代的计算手段并随着计算工具的发展而发展. 同时，它也区别于计算机科学的各个分支. 这是因为它所涉及的对象不仅有离散变量，而且更多的有连续变量，如时间、位移、速度等.

一个计算过程主要包括如下几个环节：

（a）建立一个数学模型，即数学建模；

(b) 设计求解数学模型的算法，即算法设计；

(c) 设计计算机上实现算法的软件；

(d) 上机运行，数值模拟物理过程；

(e) 计算结果再表示，如图像的可视化等；

(f) 分析计算结果的可靠性，必要时重复上述过程.

其中算法设计是本书的核心内容. 本书将针对来源于科学与工程中的数学模型问题，介绍计算机上常用的数值计算方法的算法设计思想并进行算法分析. 具体包括非线性方程求根、线性代数方程组求解、函数逼近(插值逼近和拟合逼近)、数值微积分和常微分方程数值解等内容.

设计一个算法的第一步是将问题可算化. 一般地，可通过如下手段将问题可算化.

(a) 用有限维空间代替无限维空间，如多项式逼近连续函数等；

(b) 用有限过程代替无限过程，如积分和无穷级数用有限项和代替，导数用差分代替等；

(c) 用代数方程代替微分方程，线性问题代替非线性问题，低阶系统代替高阶系统，简单函数代替复杂函数等. 简言之，用简单问题替代复杂问题.

例如，在求解一个非线性微分方程组时，首先将问题化为非线性代数系统，即非线性方程组，然后再用一系列线性代数方程组逼近该非线性方程组. 而在求解线性代数方程组时，通常可将问题化为易于求解的具有特殊系数矩阵(如三角形矩阵)的线性代数方程组.

在上述每个环节中，我们不仅希望问题得到简化，而且需要验证转化后的替代问题的解是否和原问题的解保持一致或者在容许的误差范围之内，也即说，我们还需做到

(d) 替代问题的解和原问题的解在某种意义上保持一致.

从理论的角度上看，人们自然希望利用等解变换将问题可算化. 而在实际上，这种等解变换往往不可能. 这意味着在原问题和变换后的替代问题的解之间存在着扰动，因而需要我们对解的扰动进行分析，即估计逼近解(变换后的替代问题的解)与精确解(原数学模型的解)之间的误差或精度.

例 1.1.1 计算 $\sin x$ 的值，其中已知角 x 的弧度在 0 和 π 之间.

解 电子计算机实质上只不过是一个减法器，即只会做加减乘除等基本运算. 因此，需将问题可算化. 根据微分学的泰勒(Brook Taylor，英国，1685～1731)公式，函数值 $\sin x$ 的计算问题可化为无穷级数求和问题

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + R_{2n+1}(x).$$

这是一个等解变换. 变换后的级数求和问题包含有无穷多次的加减乘除运算. 由于计算机只能进行有限次运算，故用有限和替代无穷级数，即

$$\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} = p_{2n+1}(x). \tag{1.1.1}$$

多项式 $p_{2n+1}(x)$仅包含有限次的加减乘除运算，可以上机计算. 当 n 充分大时，余项(通常称为截断误差或公式误差)

$$R_{2n+1}(x) = \sin x - p_{2n+1}(x)$$

的绝对值会很小. 故当 n 充分大时，可用多项式 $p_{2n+1}(x)$的值近似函数 $\sin x$ 的值，即三角函数 $\sin x$ 的计算问题转化为计算机可计算的多项式计算问题.

式(1.1.1)通常称为泰勒多项式逼近. 在第三章和第四章，我们将介绍更为有效的函数逼近方法.

例 1.1.2　计算正实数 c 的开平方值$\sqrt{c}$.

解　下面介绍求解此问题的一种迭代算法，即从给定初值出发，通过某种方式，产生一个逐步逼近$\sqrt{c}$的序列的方法. 这种迭代方法早在 4000 多年以前就已被古巴比伦人所知. 设给定初值 $x_0>0$. 显然，根值$\sqrt{c}$在正数 x_0 和$\frac{c}{x_0}$之间. 取 x_0 和$\frac{c}{x_0}$的中间数 $x_1=\frac{1}{2}(x_0+\frac{c}{x_0})$，则 x_1 较数 x_0 或$\frac{c}{x_0}$更接近根值$\sqrt{c}$. 如此可得到点列 $x_0, x_1, \cdots, x_k, \cdots$(称为迭代序列)，其计算公式(称为迭代格式)为

$$x_{k+1} = \frac{1}{2}(x_k + \frac{c}{x_k}). \tag{1.1.2}$$

如果$\{x_k\}$收敛，即$\lim\limits_{k\to\infty} x_k = x^*$存在，则在式(1.1.2)中令 k 趋于无穷，得

$$x^* = \frac{1}{2}(x^* + \frac{c}{x^*}),$$

即 $x^*=\sqrt{c}$. 故当 k 充分大时，可用 x_k 的值近似根值$\sqrt{c}$，即$\sqrt{c}$的计算问题转化为有限次迭代计算 $x_0, x_1, \cdots, x_k$.

定义 1.1.1　称通过已知点 x_0，按照某种规则产生一系列点 $x_0, x_1, \cdots, x_k, \cdots$的方法为迭代法，而点列 $x_0, x_1, \cdots, x_k \cdots$称为迭代序列.

对于迭代法，通常我们需考虑迭代序列的收敛性和收敛效率. 式(1.1.2)又称为求解非线性方程 $x^2-c=0$ 的牛顿迭代. 牛顿迭代是非线性方程求根的

一种非常有效的数值迭代算法.有关内容我们将在第二章详细介绍.

一个好的数值计算方法应具备适用范围广、运算量少、存储单元省、逻辑结构简单且计算结果可靠等特点.

例 1.1.3 计算多项式 $p(x)=5+6x+7x^2+8x^3+9x^4$ 并分析计算量.

解 如果直接计算各项并累加需要 10 次乘除法和 4 次加减法运算.但若改用下式计算

$$p(x)=5+x(6+x(7+x(8+9x))),$$

则仅需要 4 次乘除法和 4 次加减法运算.

对于一般多项式的计算,通常采用第二种方法.这种方法称为秦九韶(中国,1202～1261)算法或 Horner 算法.设多项式

$$p(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n.$$

如果直接计算,需要$\frac{n(n+1)}{2}$次乘除法和 n 次加减法运算.而若采用秦九韶算法,即

$$\begin{cases} b_0=a_n, \\ b_k=a_{n-k}+b_{k-1}x, \quad k=1,2,\cdots,n, \\ p(x)=b_n, \end{cases}$$

则仅需要 n 次乘除法和 n 次加减法运算.因此,当多项式的次数很高时,秦九韶算法有明显的优越性.

对于小型问题,计算量的减少或存储单元的节省似乎不太重要,但对于大规模问题,有时却有着决定性的意义.

例 1.1.4 计算行列式 $D_n=\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$ 并分析计算量.

解 如果按照行列式的定义

$$D_n=\sum_{i_1i_2\cdots i_n}(-1)^{\tau(i_1i_2\cdots i_n)}a_{i_11}a_{i_22}\cdots a_{i_nn}$$

计算,需计算 $n!$ 项 n 个因子的乘积,故需 $M_n=n!$ $(n-1)$次乘除法运算.当 $n=20$ 时,$M_{20}\approx 4.6\times 10^{19}$. 做这么多的乘除法运算,即使在每秒做万亿次乘除法的计算机上,也需运行一年之久.因此,在计算行列式时,人们通常采用通过行或列变换将行列式化为上三角形行列式或下三角形行列式的途径计算.这样的计算方法仅需做

$$\begin{aligned}\overline{M}_n &= (n-1)^2+(n-2)^2+\cdots+1^2+(n-1)\\ &= \frac{n(n-1)(2n-1)}{6}+(n-1)\end{aligned}$$

次乘除法运算. 当 $n=20$ 时，$\overline{M}_{20}\approx 2\,489$. 此计算量较前者要少得多.

第二节　误差及误差分析

为了理解算法的可靠性和稳定性的概念，我们首先引入误差的概念.

一、误差及其来源

误差在数值计算中无处不在. 通常可分为模型误差、观测误差、截断误差和舍入误差等. 在数值计算时，首先需将一个物理系统或过程用一个数学模型描述. 由于数学建模时往往忽略了许多次要因素，使得数学模型问题和真实的物理系统或过程存在差异，称这种差异为模型误差；在数学模型中存在各种参数，它们的值往往通过观测、测量或实验等手段获得，称由此产生的误差为观测误差(也称为数据误差或前计算误差)；在计算过程中，为了求得数学模型问题的解，往往通过近似替代，将问题可算化，称由此产生的误差为截断误差(也称为离散误差、公式误差或方法误差)；由于计算机只能对数的有限位进行储存和计算，因而往往会进行舍入，称这种由舍入引起的误差为舍入误差或计算误差.

例 1.2.1　计算地球的表面积并进行误差分析.

解　我们可按照公式 $A=4\pi r^2$ 计算地球的表面积，其中地球半径取为 $r=6370$ km. 根据无穷乘积公式

$$\frac{\pi}{2}=\frac{2}{\sqrt{2}}\cdot\frac{2}{\sqrt{2+\sqrt{2}}}\cdot\frac{2}{\sqrt{2+\sqrt{2+\sqrt{2}}}}\cdot\frac{2}{\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2}}}}}\cdots,\tag{1.2.1}$$

可用有限项乘积近似圆周率 π 的值，即

$$\pi\approx p_n=2\cdot\frac{2}{q_1}\cdot\frac{2}{q_2}\cdot\cdots\cdot\frac{2}{q_n},$$

其中

$$\begin{cases}q_1=\sqrt{2},\\ q_{k+1}=\sqrt{2+q_k}, \quad k=1,2,\cdots,n-1.\end{cases}$$

比如，取 $n=8$，得

$$\pi\approx p_8=3.141\,587\,725\cdots\approx 3.141\,6.$$

因此，

$$A = 4\pi r^2 \approx 4 \times 3.141\ 6 \times 6\ 370^2$$
$$= 5.099\ 055\ 561\ 6 \times 10^8 \approx 5.099\ 1 \times 10^8 (\mathrm{km}^2).$$

在上述计算过程中，存在着各种误差：

模型误差：将地球视为球体而产生的误差；

观测误差：地球半径取估值 $r=6\ 370$ km 而产生的误差；

截断误差：π 的值由一个无穷乘积给出，计算时只截取有限项乘积计算，故存在截断误差；

舍入误差：计算过程会因舍入而产生相应的舍入误差，如 p_8 用 3.141 6 近似等.

二、误差与误差限

定义 1.2.1 设 x 为准确值，$\tilde{x}$ 为 x 的一个近似值. 称 $\Delta(x)=\tilde{x}-x$ 为近似值 $\tilde{x}$ 的绝对误差，简称误差.

由于通常情况下，准确值 x 的值未知，绝对误差 $\Delta(x)$ 的值也不能算出. 但是，我们往往可以给出绝对误差 $\Delta(x)$ 的绝对值的一个上界值. 例如，用有毫米刻度的计量工具测量某人身高 x 时，通常读出和该身高最接近的刻度 $\tilde{x}$. 此时，

$$|\Delta(x)| = |\tilde{x}-x| \leqslant 0.5(\mathrm{mm}).$$

再如，用一般的电子跑表读取时间时（读取到秒），绝对误差的绝对值不会超过 1 秒. 因此，我们给出如下绝对误差限的定义. 它的值为正值且不惟一.

定义 1.2.2 设 x 为准确值，$\tilde{x}$ 为 x 的一个近似值. 若正数 ε 满足

$$|\Delta(x)| = |\tilde{x}-x| \leqslant \varepsilon, \tag{1.2.2}$$

则称 ε 为近似值 $\tilde{x}$ 的一个绝对误差限，简称误差限.

不等式(1.2.2)也即 $\tilde{x}-\varepsilon \leqslant x \leqslant \tilde{x}+\varepsilon$，有时也表示为

$$x = \tilde{x} \pm \varepsilon.$$

误差和误差限的大小还不能完全表达近似值的精确程度. 例如，有两个量 $x=10\pm1$ 和 $y=1000\pm1$ 的绝对误差限都等于 1，但后者的精度显然比前者高. 因此，为了更好地反映近似值的近似程度，必须考虑误差与准确值的比值，即相对误差.

定义 1.2.3 设 $x\neq0$ 为准确值，$\tilde{x}$ 为 x 的一个近似值. 称

$$\delta(x) = \frac{\Delta(x)}{x} = \frac{\tilde{x}-x}{x}$$

为近似值$\tilde{x}$的相对误差.

若$\frac{\Delta(x)}{x}$的值较小，由于

$$\frac{\Delta(x)}{x}-\frac{\Delta(x)}{\tilde{x}}=\frac{(\tilde{x}-x)\Delta(x)}{x\tilde{x}}=\frac{(\Delta(x))^2}{x(x+\Delta(x))}=\frac{\left(\frac{\Delta(x)}{x}\right)^2}{1+\frac{\Delta(x)}{x}}$$

为$\frac{\Delta(x)}{x}$的近似平方，可忽略不计，故在实际计算时，也可取

$$\tilde{\delta}(x)=\frac{\Delta(x)}{\tilde{x}}=\frac{\tilde{x}-x}{\tilde{x}}$$

作为近似值$\tilde{x}$的相对误差.

定义 1.2.4　设 x 为准确值，$\tilde{x}$为 x 的一个近似值. 若正数 ε_r 满足

$$|\delta(x)|=\left|\frac{\tilde{x}-x}{x}\right|\leqslant\varepsilon_r,$$

则称 ε_r 为近似值$\tilde{x}$的一个相对误差限.

同样，若$\frac{\Delta(x)}{x}$的值较小，可用易于计算的 $\tilde{\varepsilon}_r$ 值代替 ε_r，其中 $\tilde{\varepsilon}_r$ 满足

$$|\tilde{\delta}(x)|=\left|\frac{\tilde{x}-x}{\tilde{x}}\right|\leqslant\tilde{\varepsilon}_r.$$

对于前述的两个量 $x=10\pm1$ 和 $y=1\,000\pm1$，由于

$$\varepsilon_r(x)=\frac{1}{10}=0.1,\varepsilon_r(y)=\frac{1}{1\,000}=0.001,$$

有 $\varepsilon_r(x)\gg\varepsilon_r(y)$，后者的相对误差限要小得多.

三、浮点数与有效数字

按照四舍五入原则近似下面三个值

$$x=3.141\,592\,65\cdots,\quad y=314.159\,265\cdots,\quad z=0.031\,415\,926\,5\cdots,$$

得到准确到小数点后两位的近似值：

$$\tilde{x}=3.14,\quad \tilde{y}=314.16,\quad \tilde{z}=0.03.$$

它们的绝对误差限都等于 0.005. 但是，它们的近似程度是不一样的. 比较它们的相对误差限不难判断 $\tilde{y}$ 的精度最高，$\tilde{z}$ 的精度最低.

习惯上，在数的表示中，数中的小数点的位置均固定在个位数后，这种数称为定点数. 计算机中可表示的数称为机器数. 为了提高精度，机器数通常是用浮点数表示的. 比如在一个 β 进制的字长为 t 的计算机中，非零的机器数可表示为如下浮点数的形式：

$$fl(x) = \pm 0.a_1a_2\cdots a_t \times \beta^m,$$

其中 $1 \leqslant a_1 < \beta, 0 \leqslant a_2, a_3, \cdots, a_t < \beta, m_1 \leqslant m \leqslant m_2$，而 m_1 和 m_2 为整数且 $m_1 < m_2$.

不难看出，最大和最小的正的机器数分别为

$$x_{\max}^{+} = 0.(\beta-1)(\beta-1)\cdots(\beta-1) \times \beta^{m_2}, \quad x_{\min}^{+} = 0.10\cdots0 \times \beta^{m_1};$$

而最小和最大的负的机器数分别为

$$x_{\min}^{-} = -0.(\beta-1)(\beta-1)\cdots(\beta-1) \times \beta^{m_2}, \quad x_{\max}^{-} = -0.10\cdots0 \times \beta^{m_1}.$$

数零在计算机中通常表示为

$$0.00\cdots0 \times \beta^0.$$

它在有的计算机中用最小的正的机器数，即 $0.10\cdots0 \times \beta^{m_1}$ 表示. 显然，计算机仅能表示有限个数. 下面的例子还可看出，机器数之间不仅是离散的，而且不等距.

例 1.2.2 试表示出所有的机器数，其中 $\beta=2, t=3, m_1=0, m_2=2$.

解 非零机器数可表示为

$$fl(x) = \pm (0.a_1a_2a_3)_2 \times 2^m = \pm \left(\frac{a_1}{2} + \frac{a_2}{2^2} + \frac{a_3}{2^3}\right) \times 2^m,$$

其中 $a_1=1, a_2=0,1, a_3=0,1, 0 \leqslant m \leqslant 2$. 它们可分为如下六组

$$\pm\begin{cases}0.5,\\0.625\\0.75,\\0.875;\end{cases} \quad \pm\begin{cases}1,\\1.25,\\1.5,\\1.75;\end{cases} \quad \pm\begin{cases}2,\\2.5,\\3,\\3.5.\end{cases}$$

在各组中，机器数是等距的. 因此，在一个计算机系统中，机器数的全体仅表示实轴上的有限个点，这些点分段等距(如图 1.2.1 所示).

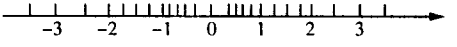

图 1.2.1

一个数在输入计算机时，如果不出现上溢(如大于最大机器数的数)，都将以机器数(如浮点数)的形式表示. 由于计算机只能表示有限个数，故通常利用某种舍入的规则(如四舍五入，截断舍入等)，将数进行浮点化. 这样势必产生舍入误差. 下面通过引入有效数字的概念分析舍入误差. 为习惯起见，我们在下面的讨论中采用十进制且假定按照四舍五入规则进行舍入.

定义 1.2.5 设 x 为准确值，如果 $\frac{1}{2} \times 10^{-n}$ 为近似数 $\tilde{x}$ 的一个绝对误差

限，即

$$|\Delta(x)|=|\tilde{x}-x|\leqslant\frac{1}{2}\times10^{-n},$$

则称$\tilde{x}$准确到小数点后n位，并称$\tilde{x}$的第一个非零数字位到第n位的全部数字为$\tilde{x}$的有效数字.

显然，如果将x经四舍五入保留到小数点后n位，则得到准确到小数点后n位的近似数. 例如：

$$\tilde{x}=3.14\approx x=3.141\ 592\ 65\cdots,$$
$$\tilde{y}=314.16\approx y=314.159\ 265\cdots,$$
$$\tilde{z}=0.03\approx z=0.031\ 415\ 926\ 5\cdots$$

的有效数字分别为3位，5位和1位.

对于一般情形，设$\tilde{x}$准确到小数点后n位. 如果$\tilde{x}$的形式为

$$\tilde{x}=\pm a_1a_2\cdots a_m.b_1b_2\cdots b_n\cdots\quad(a_1\neq0),$$

则$\tilde{x}$具有$n+m$位有效数字，且

$$|\delta(x)|=\frac{|\tilde{x}-x|}{|x|}\leqslant\frac{0.5\times10^{-n}}{0.a_1\times10^{m}}\leqslant5\times10^{-(n+m)};\qquad(1.2.3)$$

如果$\tilde{x}$的形式为

$$\tilde{x}=\pm0.00\cdots0b_{m+1}b_{m+2}\cdots b_n\cdots\quad(b_{m+1}\neq0),$$

则$\tilde{x}$具有$n-m$位有效数字，且

$$|\delta(x)|=\frac{|\tilde{x}-x|}{|x|}\leqslant\frac{0.5\times10^{-n}}{0.b_{m+1}\times10^{-m}}\leqslant5\times10^{-(n-m)}.\qquad(1.2.4)$$

由式(1.2.3)和式(1.2.4)可以看出，有效数字越多，相对误差就越小，近似数的精确程度也就越高.

在字长为t的十进制计算机中，设x经四舍五入得到机器数，即浮点数$fl(x)$，且$fl(x)$的浮点表示形式为

$$fl(x)=\pm0.a_1a_2\cdots a_t\times10^{m}\quad(a_1\neq0),$$

则$a_1,a_2,\cdots,a_t$为$fl(x)$的t位有效数字，且$fl(x)$的相对误差满足

$$|\delta(t)|=\frac{|fl(x)-x|}{|x|}\leqslant\frac{0.5\times10^{-t}}{0.a_1}\leqslant5\times10^{-t},\qquad(1.2.5)$$

即5×10^{-t}为$fl(x)$的一个相对误差限. 因此，数在转化成机器数的过程中，因舍入导致的相对误差限仅与机器的字长有关. 通常称相对误差限5×10^{-t}为计算机的相对精度.

四、误差的传播

例 1.2.3　考虑$\triangle ABC$，其中边长$a=100$，$b=101$以及夹角$C=1°$. 由余

弦定理,第三边

$$c=\sqrt{a^2+b^2-2ab\cos C}=\sqrt{(b-a)^2+4ab\sin^2\frac{C}{2}}.$$

假设边长 a 存在误差:$a\approx\tilde{a}=100.1$,试分析利用上述公式计算出来的第三边的误差有多大?

解 在表 1.2.1 中列出了各个步骤的计算结果以及相应的误差.

表 1.2.1

计算公式	真 值	近似值	相对误差(%)
a	100	100.1	0.1
$b-a$	1	0.9	−10
$(b-a)^2$	1	0.81	−19
$4ab\sin^2\frac{C}{2}$	3.076 5	3.079 6…	0.1
$(b-a)^2+4ab\sin^2\frac{C}{2}$	4.076 5	3.889 6…	−4.6
c	2.019 0	2.972 2…	−2.3

由表 1.2.1 可以看出,由于边长 a 存在 0.1%的误差,导致计算结果 c 产生 2.3%的误差,是数据误差的 23 倍.

数据误差对不同计算方法的计算结果的影响程度是不一样的.例如在例 1.2.3 中,减法运算 $b-a$ 使计算结果的误差扩大到计算前的 100 倍,而开方运算$\sqrt{(b-a)^2+4ab\sin^2\frac{C}{2}}$则使计算结果的误差缩小至计算前的一半.设有两个数 x_1,x_2 的近似数$\tilde{x}_1,\tilde{x}_2$,将它们进行简单的四则运算,可得

$$\begin{cases}\Delta(x_1\pm x_2)=\tilde{x}_1\pm\tilde{x}_2-(x_1\pm x_2)=\Delta(x_1)\pm\Delta(x_2),\\ \delta(x_1\pm x_2)=\dfrac{\Delta(x_1\pm x_2)}{x_1\pm x_2}=\dfrac{x_1}{x_1\pm x_2}\delta(x_1)\pm\dfrac{x_2}{x_1\pm x_2}\delta(x_2);\end{cases}\tag{1.2.6}$$

$$\begin{cases}\Delta(x_1x_2)=\tilde{x}_1\tilde{x}_2-x_1x_2\approx x_2\Delta(x_1)+x_1\Delta(x_2),\\ \delta(x_1x_2)=\dfrac{\Delta(x_1x_2)}{x_1x_2}\approx\delta(x_1)+\delta(x_2);\end{cases}\tag{1.2.7}$$

$$\begin{cases}\Delta(x_1/x_2)=\tilde{x}_1/\tilde{x}_2-x_1/x_2\approx\dfrac{\Delta(x_1)}{x_2}-\dfrac{x_1}{x_2^2}\Delta(x_2),\\ \delta(x_1/x_2)=\dfrac{\Delta(x_1/x_2)}{x_1/x_2}\approx\delta(x_1)-\delta(x_2).\end{cases}\tag{1.2.8}$$

由上面各公式可以看出，当两个同(异)号相近数相减(加)时，相对误差可能很大，会严重丧失有效数字；而当两个数相乘(除)时，大因子(小除数)可能使积(商)的绝对误差增大许多. 因此，在设计算法时，应尽量避免上述情况发生.

当计算函数值 $f(x)$时，计算结果也会因自变量的误差而产生误差：

$$\Delta(f(x))=f(\tilde{x})-f(x)=f'(x)\Delta(x)+\frac{f''(\xi)}{2}(\Delta(x))^2\approx f'(x)\Delta(x),$$

$$\delta(f(x))=\frac{\Delta(f(x))}{f(x)}\approx\frac{x}{f(x)}f'(x)\delta(x).$$

对于更一般的多元函数 $f(x_1,x_2,\cdots,x_n)$，有

$$\Delta(f(x_1,x_2,\cdots,x_n))\approx\sum_{i=1}^{n}\frac{\partial f(x_1,x_2,\cdots,x_n)}{\partial x_i}\Delta(x_i),\tag{1.2.9}$$

$$\delta(f(x_1,x_2,\cdots,x_n))\approx\sum_{i=1}^{n}\frac{x_i}{f(x_1,x_2,\cdots,x_n)}\frac{\partial f(x_1,x_2,\cdots,x_n)}{\partial x_i}\delta(x_i).\tag{1.2.10}$$

因此，在计算函数值时，如果

$$\frac{\partial f(x_1,x_2,\cdots,x_n)}{\partial x_i}\ \text{或}\ \frac{x_i}{f(x_1,x_2,\cdots,x_n)}\frac{\partial f(x_1,x_2,\cdots,x_n)}{\partial x_i}\tag{1.2.11}$$

大，则可能引起计算结果产生较大的绝对误差或相对误差. 式(1.2.11)中的量通常称为问题 f 的条件数，其绝对值的大小可反映函数值对数据的敏感程度.

在例 1.2.3 中，仅仅考虑了初始数据的误差给问题带来的影响. 由于计算过程中步步都可能有舍入，还应同时考虑舍入给计算结果带来的影响. 例如，在计算函数值 $f(x)$时，由于自变量有误差 $\Delta(x)=\tilde{x}-x$ 以及计算过程中的舍入误差，实际产生的误差为

$$\tilde{f}(\tilde{x})-f(x)=[\tilde{f}(\tilde{x})-f(\tilde{x})]+[f(\tilde{x})-f(x)]=\Delta_1+\Delta_2,$$

其中 Δ_1 由舍入误差引起，称为计算误差；Δ_2 由数据误差引起，称为数据误差或前计算误差(参见图 1.2.2).

例 1.2.4　利用十进制六位浮点数计算 $f(x)=x(\sqrt{x+1}-\sqrt{x})$在 $x=500$ 处的值.

解　利用浮点计算