电子·教育

高职高专计算机系列教材

中国计算机学会高职高专教育学会推荐出版

计算方法

（第三版）

吴筑筑 谭信民 邓秀勤 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: http://www.phei.com.cn

高职高专计算机系列教材

计 算 方 法

（第三版）

吴筑筑　谭信民　邓秀勤　编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书根据计算机专业(高职高专)教学大纲编写,着重介绍电子计算机上常用的数值计算方法。全书分6章,内容包括误差、一元非线性方程的解法、线性代数方程组的解法、插值法和曲线拟合、数值积分、常微分方程数值解法等方面的基础知识。常用算法给出计算步骤或计算框图,并有用C语言编写的参考程序,便于上机应用。各章有较多例题和习题,附录中给出习题答案以及用数学软件 Mathcad 7.0 解决常用数值计算问题的例子。全书叙述由浅入深,文字通俗流畅,便于自学。

本书适合作为高职高专院校开设数值计算方法课程的教材,也适合工程技术人员自学或参考。

图书在版编目(CIP)数据

计算方法/吴筑筑,谭信民,邓秀勤编著. -3版. 北京:电子工业出版社,2001.7

高等专科计算机系列教材

ISBN 7-5053-6709-9

Ⅰ.计… Ⅱ.①吴…②谭…③邓… Ⅲ.电子计算机-计算方法-高等学校-教材 Ⅳ.TP301.6

中国版本图书馆 CIP 数据核字(2001)第035171号

丛 书 名:高职高专计算机系列教材

书 名:**计算方法(第三版)**

编 著 者:吴筑筑 谭信民 邓秀勤

策 划:张孟玮

责任编辑:赵文博

排版制作:电子工业出版社计算机排版室

印 刷 者:北京牛山世兴印刷厂

装 订 者:三河市路通装订厂

出版发行:电子工业出版社 URL:http://www.phei.com.cn

北京市海淀区万寿路173信箱 邮编100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:8.5 字数:216千字

版 次:2001年7月第3版 2001年7月第1次印刷

书 号:ISBN 7-5053-6709-9/TP·3748

印 数:8 000册 定价:11.00元

出版说明

高职高专的计算机专业面临着两方面的巨大变化，一是计算机技术的飞速发展，另一方面是高职高专教育本身的改革和重组。

当前，计算机技术正经历着高速度、多媒体网络化的发展，计算机教育特别是计算机专业的教材建设必须适应这种日新月异的形势，才能培养出不同层次的合格的计算机技术专业人才。为了适应这种变化，国内外都在对计算机教育进行深入的研究和改革。美国 IEEE 和 ACM 在推出了《Computing Curricula 2000》之后，立即又推出了《Computing Curricula 2001》。全国高校计算机专业教学指导委员会和中国计算机学会教育委员会在 1999 年 9 月也提出了高等院校《计算机学科教学计划 2000》(征求意见稿)。目前，国内许多院校老师、专家正在研究《Computing Curricula 2001》，着手 21 世纪的中国计算机教育的改革。

高专层次和本科层次的计算机教育既有联系又有区别，高职高专的计算机教育旨在培养应用型人才。自 20 世纪 70 年代末高等专科学校计算机专业相继成立以来，高等专科学校积极探索具有自己特色的教学计划和配套教材。1985 年，在原电子工业部的支持下，由全国数十所高等专科学校参加成立了中国计算机学会教育委员会大专教育学会，之后又成立了大专计算机教材编委会。从 1986 年到 1999 年，在各校老师的共同努力下，已相继完成了三轮高等专科计算机教材的规划与出版工作，共出版了 78 种必修课、选修课、实验课教材，较好地解决了高专层次计算机专业的教材需求。

为了适应计算机技术的飞速发展以及高职高专计算机教育形势发展的需要，中国计算机学会教育委员会高职高专教育学会和高职高专计算机教材编委会于 2000 年 7 月开始，又组织了一批本科高校、高等专科学校、高等职业技术院校和成人高等院校的有教学经验的老师，学习研究参考了高等院校《计算机学科教学计划 2000》(征求意见稿)，提出了按照新的计算机教育计划和教学改革的要求，编写高专、高职、成人高等教育三教统筹的第四轮教材。

第四轮教材的编写工作采取了以招标的方式征求每门课程的编写大纲和主编，要求投标老师详细说明课程改革的思路、本课程和相关课程的联系、重点和难点的处理等。在第四轮教材的编写过程中，编委会强调加强实践环节、强调三教统筹、强调理论够用为度的原则，要求教学计划、教学内容适应高等教育发展的新形势。本套教材的编者均为各院校具有丰富教学实践经验的教师。因此，第四轮教材的特点是体系结构比较合理、内容新颖、概念清晰、通俗易懂、理论联系实际、实用性强。

竭诚希望广大师生对本套教材提出批评建议。

中国计算机学会教育委员会高职高专教育学会

2001 年 1 月

先后参加中国计算机学会教育委员会高职高专教育学会和高职高专计算机教材编委会学术活动的部分学校名单

山西师范大学
河北师范大学
承德石油高等专科学校
河北大学
保定职业技术学院
北京科技大学
北京市机械工业管理局职工大学
北方工业大学
北京船舶工业管理干部学院
海淀走读大学
北京信息工程学院
中国人民大学
北京师范大学
沈阳电力高等专科学校
辽宁交通高等专科学校
吉林大学
吉林职业师范学院
黑龙江大学
哈尔滨工业大学
哈尔滨师范大学
上海理工大学
上海第二工业大学
上海交通大学
上海商业职业技术学院
上海电机技术高等专科学校
上海旅游高等专科学校
金陵职业大学
南京建筑工程学院
南京工程学院
南京师范大学
常州工学院
无锡职业技术学院
苏州市职工大学
空军后勤学院
连云港化工高等专科学校
泰州职业技术学院
潍坊高等专科学校
青岛化工学院
天津轻工业学院
浙江大学
宁波高等专科学校
福州大学
重庆电子职业技术学院
湖南大学
湖南计算机高等专科学校
中国保险管理干部学院
湖南税务高等专科学校
长沙大学
湖南财经高等专科学校
邵阳高等专科学校
江汉大学
中国地质大学
武汉职业技术学院
河南职业技术学院
平原大学
安阳大学
开封大学
洛阳大学
河南大学
广州市财贸管理干部学院
广东轻工职业技术学院
广州航海高等专科学校
韶关大学
佛山科技学技术学院
南宁职业技术学院
广西水利电力职业技术学院
桂林电子工业学院
柳州职业技术学院
成都电子机械高等专科学校
电子科技大学
成都师范高等专科学校
四川师范学院
云南财贸学院
西安电子科技大学
兰州石化职业技术学院
兰州师范高等专科学校

前　言

本书由中国计算机学会高职高专教育学会、高职高专计算机专业教材编审委员会负责征稿、审定、推荐出版。本书适合高职高专院校作为数值计算方法课程的教材。

电子计算机的应用日益广泛，为工程技术和科学实验中进行科学计算提供了强有力的工具，面对种类繁多的数值计算问题，如何选择合适的计算方法，并正确地在计算机上实现以及如何估计结果的可靠程度，这都是科技工作者需要掌握的基本知识。

本教材是根据新时期高等专科和高等职业技术教育发展的需要，在作者 1998 年 1 月出版的《计算方法》一书的基础上修订而成。此次修订进一步注意在知识面上既有一定的广度又深浅适合，符合高职高专学生的培养目标和基础知识水平。在内容的安排上突出思维方法的培养，强调各种数值算法的构造思想及用法，数学推导过程注意加强启发性，一些较难的理论过程作了压缩或简化，文字叙述更注意通俗流畅，增强了可读性。各章都增加了一些例题和习题，并在附录 B 中给出习题答案，便于学习时参考。本书注意突出应用技能的训练，在实验内容的安排上增加了一些常用算法的参考程序，一些原有程序的流程设计补充了较为详细的说明，更有利于学生理解数值算法的编程特点，提高应用技能。注意到现代数学软件的快速发展，在附录 A 中介绍了如何应用国际流行的数学软件 Mathcad 7.0 解决本教材涉及的数值计算问题的各种例子，适合学生通过上机实验掌握该软件的基本用法。希望这有助于学生扩展计算机软件的知识面，增加解决数值计算问题的能力。

本书共分 6 章及两个附录。第 1 章介绍误差的有关知识；第 2 章介绍一元非线性方程的几种基本解法；第 3 章介绍线性代数方程组的直接解法和迭代解法；第 4 章为插值法和曲线拟合；第 5 章为数值积分；第 6 章为常微分方程初值问题的数值解法。各章有较多的习题和上机实验题，可供选用。所有程序都用 Turbo C 语言编写并上机运行通过。附录 A 给出 Mathcad 7.0 基本用法介绍，并给出用 Mathcad 7.0 解决数值计算问题的例子，可作为课外上机实验的内容。附录 B 为习题答案。对于学时数安排较少的专业可根据情况自行选择。

本书由吴筑筑主编。第 1,2 章及附录 B 由邓秀勤编写，第 3,4 章及附录 A 由吴筑筑编写，第 5,6 章由谭信民编写。

本书由季夜眉副教授主审。本书的编写得到韶关学院计算机系骆耀祖、叶宇风、严廷栋的热情支持和关心。编写时收到一些兄弟院校同行教师提出的中肯建议并参考了一些相关教材。在资料整理中得到于江明的协助。曾浦华为本书软盘稿做了许多输入及制图工作并参与了资料整理，邓广雄、梁清玲协助输入 3 部分文稿。在此一并表示诚挚的谢意。

由于我们水平有限，错误和不妥之处在所难免，敬请读者批评指正。

编　者

2001 年春

目　录

第1章 误 差

科学实验方法、科学理论方法和科学计算方法是现代社会的三类科学方法。本书的目的是介绍一些常用的、基本的科学计算方法，也就是求解科学和技术领域中常见的各种数学问题的数值计算方法。这些计算方法所给出的答案一般是所求真解的某些近似值，所以必须了解并估计近似值与真解的准确值之间的差异，即误差。由于科学计算的主要工具是数字电子计算机，因此，还应当了解在电子计算机上如何实施数值计算，它与严格的数学计算有什么区别。为此，本章简要介绍误差的基本理论以及算法的数值稳定性概念，作为学习以后各章的准备。

1.1 科学计算中误差的来源

1.1.1 计算机中数的表示

为了讨论近似数在运算过程中的变化和误差的有关概念，先给出数的一种一般表示形式。假定提供给计算机的数 x 只是有限位小数，这样，数 x 可以表示成

$$x = \pm 0.d_1 d_2 \cdots d_t \times 10^n \tag{1.1}$$

其中 n 为一整数，t 为一正整数，$d_1, d_2, \cdots, d_t$ 为 0 到 9 中任一数字。x 也可以写成

$$x = \pm 10^n (d_1 \times 10^{-1} + d_2 \times 10^{-2} + \cdots + d_t \times 10^{-t}) = \pm 10^n \sum_{k=1}^{t} d_k 10^{-k} \tag{1.2}$$

例如

$$536 = 0.536 \times 10^3 = 10^3 \times (5 \times 10^{-1} + 3 \times 10^{-2} + 6 \times 10^{-3})$$
$$-62.4 = -0.624 \times 10^2 = -10^2 \times (6 \times 10^{-1} + 2 \times 10^{-2} + 4 \times 10^{-3})$$
$$1.256 = 0.1256 \times 10^1 = 10^1 \times (1 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3} + 6 \times 10^{-4})$$
$$0.0023 = 0.23 \times 10^{-2} = 10^{-2} \times (2 \times 10^{-1} + 3 \times 10^{-2})$$

式 (1.1)或(1.2)是通常的数的十进制系统计数法，其中 10 称为十进制系统的**基底**。

在计算机中，还广泛采用二进制，八进制和十六进制系统表示数的方法，它们的基底分别为 2、8 和 16。

一般地，在计算机上一个 p 进制数 x 可以表示成

$$x = \pm 0.d_1 d_2 \cdots d_t \times p^n \tag{1.3}$$

或

$$x = \pm p^n \sum_{k=1}^{t} d_k p^{-k} \tag{1.4}$$

其中 $d_1, d_2, \cdots d_t$ 都是 0，1，$\cdots p-1$ 中的一个数字，$0.d_1 d_2 \cdots d_t$ 称为数 x 的**尾数**。自然数 t 称为计算机的**字长**，它表示数 x 的尾数的位数。n 是整数，称为数 x 的**阶**，它用来确定该数的小数点的位置。

在各种计算机中，有各自规定的字长 t，以及阶 n 的范围：$L \leqslant n \leqslant U$，$L$，$U$ 的大小表明计算机所能表示的数的范围大小。

由(1.3)或(1.4)式表示的数，小数点的位置决定于数 x 的阶 n，这种允许小数点位置浮动的表示方法，称为**数的浮点表示法**，用浮点表示法表示的数称为**浮点数**。一个无符号的浮点数由尾数和阶两部分确定。

一个数可以有不同的浮点表示。例如 3650 可以表示成 0.3650×10^4，也可以表示成 0.0365×10^5。

为了避免发生这种现象，以保证浮点数表示式的惟一性，当数 $x\neq0$ 时，规定在浮点数表示式中尾数的第一位数字 $d_1\neq0$，这样的浮点数称为**规格化浮点数**。

数的浮点表示方法是现代数字电子计算机通用的表示法，也是我们研究数值方法的基础。按式(1.3)或(1.4)规定的浮点数的全体组成的集合记作 F，称为**浮点数系**。再假定当 $x\neq0$时 $d_1\neq0$，则称 F 为**规格化浮点数系**。

计算机的字长总是有限的，所以浮点数系 F 是一个离散的有限集合，在利用计算机进行计算时，初始数据和中间结果都可能不在 F 中，于是便发生用 F 中的数来近似地表示相应数据的问题。设实数 x 不属于 F，计算机会用 F 中最接近 x 的一个浮点数作为 x 的近似值，记这个浮点数为 $fl(x)$，它一般用“舍入”法来确定。

例如，设 $t=4,p=10$，则

$$fl(0.204\ 56\times10^{12})=0.204\ 6\times10^{12}$$

$$fl(15.732)=0.157\ 3\times10^{2}$$

一般说来，若将非零实数 x 写成

$$x=\pm0.\alpha_1\alpha_2\cdots\alpha_t\alpha_{t+1}\cdots\times10^n,0\leqslant\alpha_i\leqslant9,\alpha_1>0 \tag{1.5}$$

则

$$fl(x)=\begin{cases}\pm0.\alpha_1\alpha_2\cdots\alpha_t\times10^n,\text{若 }0\leqslant\alpha_{t+1}\leqslant4\\ \pm(0.\alpha_1\alpha_2\cdots\alpha_t+10^{-t})\times10^n,\text{若 }\alpha_{t+1}\geqslant5\end{cases} \tag{1.6}$$

其中 $fl(x)$ 与 x 符号相同。

也就是说，对于十进制数，一般还是采用“四舍五入”的方法，对于二进制数，一般采用“零舍一入”的方法，其余类似。

数 0 在计算机中用尾数为 0 的浮点数表示，其阶可以是允许范围内的任意值。

当实数 x 大于 F 中的最大数，或小于 F 中的最小非零数时，F 中找不到一个浮点数等于 $fl(x)$，这时计算机就不能继续进行下去，这种现象就是“溢出”(上溢或下溢)。

1.1.2 浮点数的运算特点

设 x,y 都是规格化的浮点数，即 $x,y\in F$。它们的算术运算的精确结果不一定是 F 中的浮点数，计算机自动把运算结果用 F 中的规格化浮点数表示出来，称这个过程为“**规格化**”。此外，当两个数量级不同的数相加减时需要**对阶**，将阶码统一为较大者，然后才能将尾数相加减。

例 1.1 设 $t=4,p=10,x=0.312\ 7\times10^{-6},y=0.415\ 3\times10^{-4}$

则

$$x+y\approx0.003\ 1\times10^{-4}+0.415\ 3\times10^{-4}\quad\text{(对阶)}$$

$$=0.418\ 4\times10^{-4}\quad\text{(规格化)}$$

而 $x+y$ 的精确结果是 $0.418\ 427\times10^{-4}$，它不在 F 中。

例 1.2 设 $t=5,p=10,x=0.375\ 69\times10^{4},y=0.963\ 31\times10^{-5}$

则

$$x+y\approx0.375\ 69\times10^{4}+0.000\ 00\times10^{4}\quad\text{(对阶)}$$

$$=0.375\ 69\times10^{4}\quad\text{(规格化)}$$

其结果大数“吃掉”了小数。

例 1.3 在 3 位十进制计算机上对数 0.043 8,0.039 7,13.2 做加法运算，则

$$(0.438\times10^{-1}+0.397\times10^{-1})+0.132\times10^{2}$$
$$\approx0.835\times10^{-1}+0.132\times10^{2}\approx0.133\times10^{2}$$
$$0.438\times10^{-1}+(0.397\times10^{-1}+0.132\times10^{2})$$
$$\approx0.438\times10^{-1}+0.132\times10^{2}\approx0.132\times10^{2}$$

而这三个数的准确和为 13.283 5。

由例 1.3 可见，在计算机中进行浮点数的运算时，通常实数加法的结合律不成立。另外，易证对于浮点运算，乘法对加法的分配律也不成立。由于浮点运算具有种种与实数运算不同的特点，每一步运算都可能产生误差，因此在设计算法和实际计算中必须注意对误差的估计。

1.1.3 误差的来源与分类

数值计算的过程首先需要建立科研和工程设计中所提出的实际问题的数学模型，再用数值方法来求解相应数学问题，并以某种计算机能理解的计算机语言来描述相应算法，上机运算并求出计算结果，最后还要验证结果的正确性，其过程如图 1.1 所示。

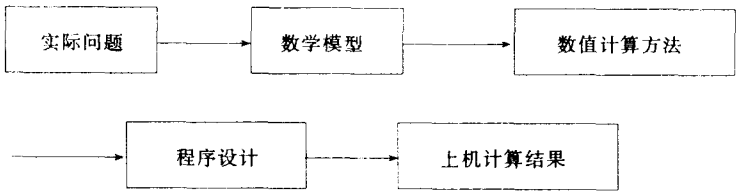

图 1.1 数值计算过程

因此误差按照它们的来源可分为以下四类。

1.模型误差

在将实际问题转化为数学模型的过程中，为了使数学模型尽量简单，以便于分析或计算，往往要忽略一些次要的因素，进行合理的简化。这样，实际问题与数学模型之间就产生了误差，这种误差称为**模型误差**。由于这类误差难于做定量分析，所以在计算方法中，总是假定所研究的数学模型是合理的，对模型误差不做深入的讨论。

2.观测误差

在数学模型中，一般都含有从观测(或实验)得到的数据，如温度、时间、速度、距离、电流、电压等等。但由于仪器本身的精度有限或某些偶然的客观因素，会引入一定的误差，这类误差叫做**观测误差**。通常根据测量工具或仪器本身的精度，可以知道这类误差的上限值，所以无须在计算方法中做过多的研究。

3.截断误差(方法误差)

有许多数学问题的解，不可能经过有限次算术运算计算出来。当数学模型得不到精确解时，要用数值计算方法求它的近似解，其近似解与精确解之间的误差称为**截断误差**或**方法误**

差。比如在数值计算中，常用收敛的无穷级数的前几项来代替无穷级数进行计算，即抛弃了无穷级数的后段，这样就产生了截断误差。例如

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots, \quad -\infty < x < +\infty$$

当$|x|$很小时，常用 x 代替 $\sin x$，其截断误差 $\sin x - x$ 的绝对值大约为$\frac{1}{6}|x|^3$。截断误差的大小，直接影响数值计算的精度，所以它是数值计算中必须十分重视的一类误差。

4.舍入误差

由于计算机字长有限，原始数据的输入及浮点运算的过程中都可能产生误差。而事实上，无论用电子计算机、计算器计算还是笔算，都只能用有限位小数来代替无穷小数或用位数较少的小数来代替位数较多的有限小数，这样产生的误差叫做**舍入误差**。

例如：用 1.414 2 近似代替$\sqrt{2}$，产生的误差 $R = \sqrt{2} - 1.414\ 2 = 0.000\ 013\ 5\cdots$就是舍入误差。

用 2.718 28 近似代替 e，舍入误差 = e − 2.718 28 = 0.000 001 8…。

在数值计算中，往往要进行成千上万次四则运算，因而就会有成千上万个舍入误差产生，这些误差一经叠加或传递，对精度可能有较大的影响。所以，做数值计算时，对舍入误差应予以足够的重视。

显然，上述四类误差都会影响计算结果的准确性，但模型误差和观测误差往往需要与各有关学科的科学工作者共同研究，因此在计算方法课程中，主要研究截断误差和舍入误差(包括初始数据的误差)对计算结果的影响。

1.2 误差的基本估计方式

1.2.1 绝对误差和绝对误差限

定义 1.1 假设某一量的准确值为 x，近似值为 x^*，则 x 与 x^* 之差叫做近似值 x^* 的**绝对误差**(简称**误差**)，记为 $\varepsilon(x)$，即

$$\varepsilon(x) = x - x^* \tag{1.7}$$

$|\varepsilon(x)|$的大小在一定程度上标志着 x^* 的精确度。一般地，在同一量的不同近似值中，$|\varepsilon(x)|$越小，x^* 的精确度越高。当$|\varepsilon(x)|$较小时，由微分和增量的关系知 x^* 的绝对误差 $\varepsilon(x) \approx \mathrm{d}x$，因此可以利用微分估计误差。

由于准确值 x 一般不能得到，于是误差 $\varepsilon(x)$的准确值也无法求得，但在实际测量计算时，可根据具体情况估计出它的大小范围。也就是指定一个适当小的正数 ξ，使

$$|\varepsilon(x)| = |x - x^*| \leqslant \xi \tag{1.8}$$

称 ξ 为近似值 x^* 的**绝对误差限**。

值得注意的是 ξ 不是惟一的，一般 ξ 越小，x^* 精度越高。有时也用

$$x = x^* \pm \xi \tag{1.9}$$

表示近似值的精度或准确值的所在范围。

例 1.4 设 $x = \frac{2}{3} = 0.666\ 6\cdots$，取 $x^* = 0.667$，则

$$|\varepsilon(x)| = |x - x^*| = 0.000\,333\cdots < 0.000\,5$$

可以估计 x^* 的绝对误差限为 0.000 5。

在实际问题中，绝对误差一般是有量纲的。例如，测得某一物体的长度为 5m，其误差限为 0.01m，通常将准确长度 s 记为

$$s = 5 \pm 0.01$$

即准确值在 5m 左右，但不超过 0.01m 的误差限。

1.2.2 相对误差和相对误差限

对不同大小的量，单凭近似值的绝对误差的大小并不能确定近似程度的好坏。

例 1.5 设 $x = 100$(厘米)，$x^* = 99$(厘米)，则 $|\varepsilon(x)| = 1$(厘米)；

$y = 10\,000$(厘米)，$y^* = 9\,950$(厘米)，则 $|\varepsilon(y)| = 50$(厘米)。

从表面上看，后者的绝对误差是前者的 50 倍。但是，前者平均每厘米长度产生了 0.01 厘米的误差，而后者每厘米长度只产生了 0.005 厘米的误差。可见，决定一个量的近似值的精确度除了要看绝对误差的大小外，还要考虑到该量本身的大小。据此，引进相对误差的概念。

定义 1.2 绝对误差与准确值之比

$$\varepsilon_r(x) = \frac{\varepsilon(x)}{x} = \frac{x - x^*}{x},\quad x \neq 0 \tag{1.10}$$

称为 x^* 的**相对误差**。

由于准确值 x 往往是不知道的，因此在实际问题中，当 $|\varepsilon_r(x)|$ 较小时，常取

$$\varepsilon_r(x) = \frac{\varepsilon(x)}{x^*}$$

一般地，在同一量或不同量的几个近似值中，$|\varepsilon_r(x)|$ 越小，x 精确度高。相对误差是一个无量纲量。

在实际计算中，由于 $\varepsilon(x)$ 与 x 都不能准确地求得，因此相对误差 $\varepsilon_r(x)$ 也不可能准确地得到，只能估计它的大小范围。即指定一个适当小的正数 η，使

$$|\varepsilon_r(x)| = \frac{|\varepsilon(x)|}{|x|} \leqslant \eta \tag{1.11}$$

称 η 为近似值 x^* 的**相对误差限**。

注意，η 是不惟一的，一般 η 越小，x^* 精度越高。当 $|\varepsilon_r(x)|$ 较小时，可以用下式来计算 η：

$$\eta = \frac{\xi}{|x^*|} \tag{1.12}$$

例 1.5 中 x^* 的相对误差 $\varepsilon_r(x) = \frac{1}{100} = 0.01$，$y^*$ 的相对误差 $\varepsilon_r(x) = \frac{50}{10\,000} = 0.005$，由于 $|\varepsilon_r(x)| > |\varepsilon_r(y)|$，所以 y^* 近似 y 的程度比 x^* 近似 x 的程度好。

1.2.3 有效数字

为了衡量一个近似值的准确程度，引入有效数字的概念。

定义 1.3 若近似值 x^* 的绝对误差限是某一位上的半个单位，则说 x^* **精确到该位**，若从该位到 x^* 的左面第一位非零数字一共有 n 位，则称近似值 x^* 有 n 位**有效数字**。

准确数有无穷多位有效数字，因为准确数的绝对误差为 0。

例 1.6 设 $x=\pi=3.141\,592\,6\cdots$，分别取 $x_1^*=3, x_2^*=3.14, x_3^*=3.141\,6$ 作为 x 的近似值，试求它们各有多少位有效数字？

解 $x_1^*=3$，$|\varepsilon_1(x)|=0.141\,5\cdots\leqslant 0.5\times 10^0$，其绝对误差限是 10^0 位(即个位)上的半个单位，所以 x_1^* 精确到个位，它有 1 位有效数字。

$x_2^*=3.14$，$|\varepsilon_2(x)|=0.001\,59\cdots\leqslant 0.5\times 10^{-2}$，所以 x_2^* 精确到 10^{-2}，从这一位到左面第一位非零数字 3 一共有 3 位，因此 x_2^* 有 3 位有效数字。

$x_3^*=3.141\,6$，$|\varepsilon_3(x)|=0.000\,007\,4\cdots\leqslant 0.5\times 10^{-4}$，所以 x_3^* 精确到 10^{-4}，它有 5 位有效数字。

实际上，用四舍五入法从准确值 x 的左面第一位非零数字起取前 n 位作为 x 的近似值 x^* 时，x^* 有 n 位有效数字。这是因为绝对误差限不超过 x^* 末位上的半个单位。

例 1.7 设 $x=4.269\,72$，按四舍五入原则，分别取 $x_1^*=4.3$，$x_2^*=4.27$，$x_3^*=4.270$，则 x_1^* 有 2 位有效数字，x_2^* 有 3 位有效数字，x_3^* 有 4 位有效数字。

值得注意的是，近似值后面的零不能随便省去。如例 1.7 中 4.27 和 4.270，前者精确到 0.01，其有 3 位有效数字；而后者精确到 0.001，其有 4 位有效数字。可见，它们的近似程度完全不同。

例 1.8 如果用 $\frac{22}{7}=3.142\,857\cdots$ 近似 π，因为 $|\pi-\frac{22}{7}|=0.001\,264\cdots<0.5\times 10^{-2}$，所以它精确到 10^{-2}，按定义 1.3，它有 3 位有效数字。

定义 1.3 的一种等价说法是定义 1.3′。

定义 1.3′ 设 x 的近似值 x^* 表示成

$$x^*=\pm 0.\alpha_1\alpha_2\cdots\alpha_n\cdots\times 10^p \tag{1.13}$$

若其绝对误差限为 $0.5\times 10^{p-n}$，即

$$|\varepsilon(x)|=|x-x^*|\leqslant 0.5\times 10^{p-n} \tag{1.14}$$

则称近似数 x^* 具有 n 位**有效数字**。这里 p 为整数，α_1，α_2，…，α_n 是 0 到 9 中的一个数字且 $\alpha_1\neq 0$。

事实上，若(1.14)成立，则 x^* 的绝对误差限不超过 10^{p-n} 这个数位上的半个单位，而根据(1.13)，x^* 的 10^{p-n} 位正是 α_n 所在的数位，由于 $\alpha_1\neq 0$，根据定义 1.3，显然 x^* 具有 n 位有效数字。

必须注意，有效数字位数与小数点位置无关，只有经过规格化以后，小数点后的数字位数才能反映其有效数字位数多少。

例 1.9 已知 $x=31.203\,6$ 的绝对误差 $\varepsilon(x)=0.5\times 10^{-3}$，问 x 有多少位有效数字？

解法一 根据 $|\varepsilon(x)|$ 为 0.5×10^{-3} 可知 x 精确到 10^{-3}，因此 x 有 5 位有效数字。

解法二 ∵ $x=0.312\,036\times 10^2$

∴ $p=2, 2-n=-3$ ∴ $n=5$

∴ x 有 5 位有效数字。

例 1.10 求近似数 56.000 0 的绝对误差限。

解法一 因为该近似数最末位有效数字在 10^{-4} 数位上，所以它的绝对误差限为 0.5×10^{-4}。

解法二 由 $56.000\,0=0.560\,000\times 10^2$ 得 $p=2$

已知该数有 6 位有效数字，$n=6$

所以绝对误差限为 $0.5\times10^{2-6}=0.5\times10^{-4}$

从式(1.14)可知，具有 n 位有效数字的近似数 x^* 的绝对误差限为 $0.5\times10^{p-n}$，在 p 相同的情况下，n 越大，10^{p-n} 的值越小，所以**对同一个量的不同近似值，有效数字位越多，则绝对误差限越小**。我们还可证明：**有效数字位越多，相对误差限也越小，反之亦然。**

由此可见，**有效数字位数可确切反映近似数的精确度，相对误差限与有效数字的位数有关**。

1.2.4 算术运算的误差

1. 利用微分估计误差

数值运算的误差估计情况较复杂，通常可利用微分估计误差。设数学问题的解 y 与变量 x_1,x_2 有关，$y=f(x_1,x_2)$。若 x_1,x_2 的近似值为 x_1^*,x_2^*，相应解为 y^*，则当数据误差较小时解的绝对误差

$$\begin{aligned}\varepsilon(y)&=y-y^*=f(x_1,x_2)-f(x_1^*,x_2^*)\\&\approx \mathrm{d}y=\frac{\partial f(x_1,x_2)}{\partial x_1}\varepsilon(x_1)+\frac{\partial f(x_1,x_2)}{\partial x_2}\varepsilon(x_2)\end{aligned}\tag{1.15}$$

设 $y=f(x)$为一元函数，则计算函数值的误差为

$$\varepsilon(y)\approx \mathrm{d}y=f'(x)\mathrm{d}x\approx f'(x)\varepsilon(x)$$

$$\varepsilon_r(y)\approx\frac{\mathrm{d}y}{y}$$

利用微分可得到两数和、差、积、商的误差估计。

2. 算术运算的误差估计

利用式(1.15)可推出两个近似数进行加、减、乘、除运算得到的绝对误差分别为

$$\varepsilon(x_1\pm x_2)\approx\varepsilon(x_1)\pm\varepsilon(x_2)\tag{1.16}$$

$$\varepsilon(x_1x_2)\approx d(x_1x_2)\approx x_2\varepsilon(x_1)+x_1\varepsilon(x_2)\tag{1.17}$$

$$\varepsilon\left(\frac{x_1}{x_2}\right)\approx d\left(\frac{x_1}{x_2}\right)\approx\frac{x_2\varepsilon(x_1)-x_1\varepsilon(x_2)}{x_2^2},(x_2\neq0)\tag{1.18}$$

由于 $|\varepsilon(x_1+x_2)|=|\varepsilon(x_1)+\varepsilon(x_2)|\leqslant|\varepsilon(x_1)|+|\varepsilon(x_2)|$，因此，**任何两个数之和的绝对误差限为这两个数的绝对误差限之和，且可推广到有限多个数相加的情形**。所以做大量加减运算后的绝对误差限是绝不可以忽视的。

两个近似数进行加、减、乘、除运算得到的相对误差分别为

$$\varepsilon_r(x_1\pm x_2)=\frac{\varepsilon(x_1\pm x_2)}{x_1\pm x_2}\approx\frac{x_1}{x_1\pm x_2}\varepsilon_r(x_1)+\frac{x_2}{x_1\pm x_2}\varepsilon_r(x_2)\tag{1.19}$$

$$\varepsilon_r(x_1x_2)=\frac{\varepsilon(x_1x_2)}{x_1x_2}\approx\varepsilon_r(x_1)+\varepsilon_r(x_2)\tag{1.20}$$

$$\varepsilon_r\left(\frac{x_1}{x_2}\right)=\varepsilon\left(\frac{x_1}{x_2}\right)\cdot\frac{x_2}{x_1}\approx\varepsilon_r(x_1)-\varepsilon_r(x_2),(x_2\neq0)\tag{1.21}$$

由(1.19)可知，当 x_1 和 x_2 相当接近时，$x_1-x_2\approx0$，$\left|\frac{x_1}{x_1-x_2}\right|$ 和 $\left|\frac{x_2}{x_1-x_2}\right|$ 都将很大，所以相

近两数之差的相对误差将很大，即原始数据的误差会对计算结果产生很大的影响。

由(1.20)，(1.21)可得：**两数乘积的相对误差，可看作是各乘数的相对误差之和；两数商的相对误差，可看作是被除数与除数的相对误差之差**。

以上所述的误差积累规律，对多个近似数的运算也是成立的。一般地，**任意多次连乘连除所得的结果的相对误差限，可看作是各乘数和除数的相对误差限之和**。

例 1.11 用四位有效数字计算 $y=\sqrt{1001}-\sqrt{1000}$的值。

解 如果直接计算，则

$$y=\sqrt{1001}-\sqrt{1000}=31.64-31.62=0.02$$

由于 y 的准确值是 0.0158074374…，可见直接计算所得的近似值仅有一位有效数字，其相对误差大于 26%。有时若做适当变形后计算，可以避免相近两数相减的计算，如

$$y=\sqrt{1001}-\sqrt{1000}=\frac{1}{\sqrt{1001}+\sqrt{1000}}=\frac{1}{63.26}\approx 0.015\ 81$$

所得结果与准确值比较可知具有四位有效数字，其相对误差不超过 0.02%。所以在数值计算中，必须避免相近两数相减，以免损失有效数字的位数。

1.3 算法的数值稳定性

1.3.1 算法的数值稳定性概念

所谓**算法**，是指对一些数据按某种规定的顺序进行运算的一个运算序列。现代电子计算机常用浮点法表示数，由于计算机计算的近似性，在实际计算中，对于同一问题我们选用不同的算法，所得的结果的精度往往大不相同。这是因为初始数据的误差或计算中的舍入误差在计算过程中的传播，因算法不同而异，于是就产生了算法的数值稳定性问题。一个算法，如果计算结果受误差的影响小，就称这个算法具有较好的**数值稳定性**。否则，就称这个算法的数值稳定性不好。例如在第 1.2 节的例 1.11 中，第一种算法数值是不稳定的，而第二种算法数值是稳定的。为了说明数值稳定性这个概念，下面再举一个例子。

例 1.12 一元二次方程

$$x^2+2px+q=0 \tag{1.22}$$

的两个根分别为 $x_1=-p+\sqrt{p^2-q}$ 和 $x_2=-p-\sqrt{p^2-q}$。当 $|p|\gg|q|$ 时，$\sqrt{p^2-q}\approx|p|$，用上述公式计算 x_1 和 x_2，其中之一会造成相近两数相减，从而损失有效数字。

例如，当 $p=-0.5\times10^5$，$q=1$ 时，方程(1.22)的两个取 11 位有效数字的根为

$$x_1=99\ 999.999\ 990,\ x_2=0.000\ 010\ 000\ 000\ 001$$

而在 8 位字长的计算机($\beta=10,t=8,L=-50,U=50$)上直接用上述求根公式计算的结果为

$$x_1=100\ 000.00,\ x_2=0$$

可见，结果 x_1 很好，但 x_2 很不理想。这说明直接用上述公式计算第二个根是不稳定的，其原因在于 $|p|\gg|q|$，在计算 x_2 时造成相近两数相减，从而使有效数字严重损失。但是，根据根与系数的关系可知

$$x_1x_2=q=1$$

所以

$$x_2=1/x_1 \tag{1.23}$$

因此，如果仍用前述方法算出 x_1，然后用公式(1.23)计算 x_2 可得