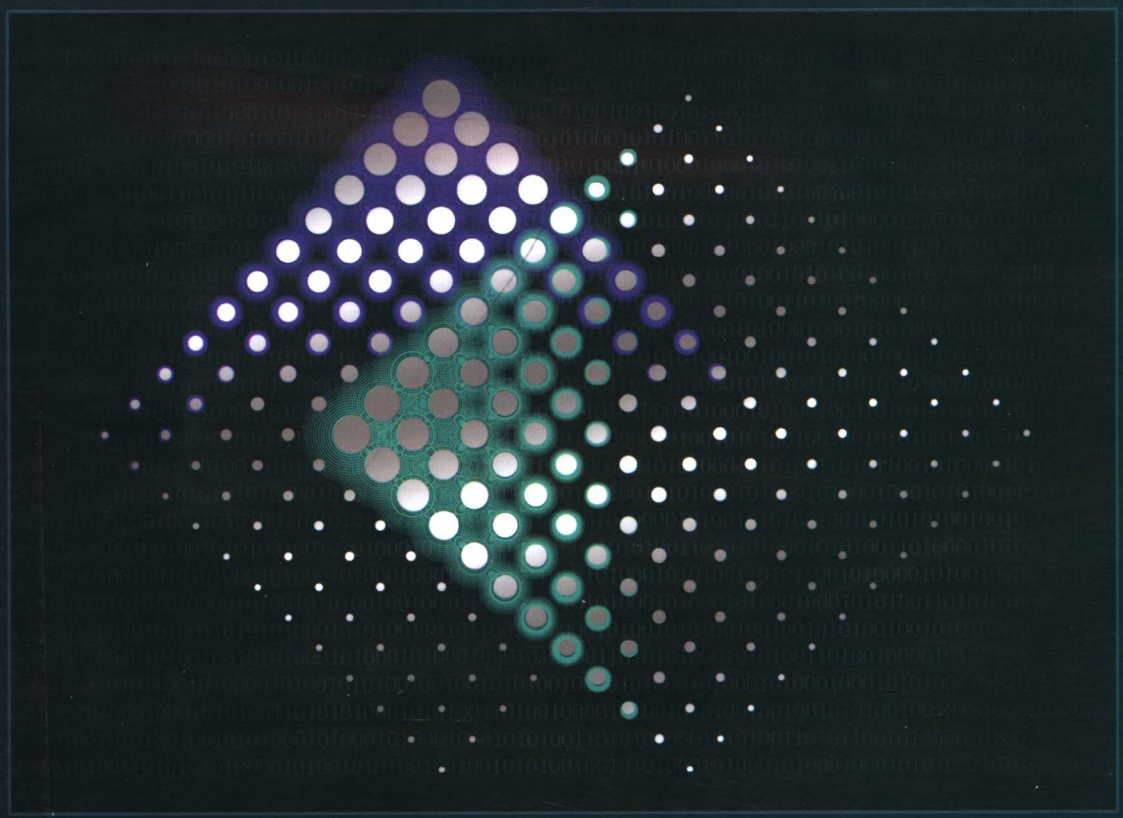
新编计算机类本科规划教材

数值计算方法

薛 莲 编著 江金生 审校



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

新编计算机类本科规划教材

数值计算方法

薛　莲　编著

江金生　审校

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

内 容 简 介

本书介绍了进行科学计算所必须掌握的一些最基本、最常用的数值计算方法及其 MATLAB 软件的应用，主要的内容包括误差知识、一元非线性方程的解法、线性方程组的解法、插值与拟合、数值积分与数值微分、常微分方程数值解法等。作者将数值计算的“分析”和“计算”放在了并重的地位，不仅仅强调“方法”的使用，并且对“方法”的研究和创造也进行了深入的阐述。此外，在每章的最后均给出一段 MATLAB 软件评注，主要介绍了相关算法的 MATLAB 程序和函数、工具箱等。

本书可作为一般高等学校理工类专业计算方法课程的教材，也可面向选读数学实验和数学建模课程的学生，同时适用从事科学计算的科技工作者。

图书在版编目（CIP）数据

数值计算方法 / 薛莲编著. —北京：电子工业出版社，2007.10
（新编计算机类本科规划教材）
ISBN 978-7-121-05098-5

Ⅰ. 数… Ⅱ. 薛… Ⅲ. 数值计算－计算方法－高等学校－教材 Ⅳ. 0241

中国版本图书馆 CIP 数据核字（2007）第 148948 号

责任编辑：冯小贝
印　　刷：北京市通州大中印刷厂
装　　订：三河市鹏成印业有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1092　1/16　印张：18　字数：461 千字
印　　次：2007 年 10 月第 1 次印刷
定　　价：25.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

随着科学技术的迅猛发展和生产实践的不断丰富，有越来越多的数值计算问题亟待人们去解决。而计算机技术的日益丰富和提高及人们对计算软件的深入研究和发展，使得这些问题的解决变得相对容易。因此，一般高等学校的绝大多数理工类专业也都相继开设了“计算方法”或“数值分析”这门课，本书正是为了满足这一广泛需要而编写的。

本书着重介绍了进行科学计算所必须掌握的一些最基本、最常用的算法及其处理技术，并加入了算法在 MATLAB 软件上的应用知识。全书内容覆盖面广、信息量大，对于科学计算的各个分支都进行了一定深度的介绍和讨论。本书可作为一般高等学校理工类专业计算方法课程的教材，同时也可作为初步接触科学计算的技术人员的参考书。

本书从一些最基本、最常用的数值计算方法及其 MATLAB 软件的应用出发，介绍了误差知识、一元非线性方程的解法、线性方程组的解法、插值与拟合、数值积分与数值微分、常微分方程数值解法等内容。主要以科学计算的实际过程为主线组织编排，突出数值计算的实用性。每一章内容均以实际问题引出，然后介绍解决同类问题的一些最具代表性的典型方法。对算法的处理重点集中在构造算法的基本思想和基本原理上，突出相关概念和内容的联系与衔接，同时对算法的误差估计、收敛性、稳定性等理论问题也进行了适当的讨论，此外，在每章的最后均给出一节 MATLAB 软件点评，内容包括相关算法的 MATLAB 程序和函数、工具箱等的介绍。每章配有一定数量的例题和习题，并对常用算法给出了详细的计算步骤，再以相应的 MATLAB 程序和相关函数、具体问题应用作为结束。此外，本书还以附录的形式简单介绍了易学易用的 MATLAB 软件。

全书由浙江大学城市学院的薛莲编著，并得到校重点课程和重点规划教材的资助。在撰写全书的过程中，浙江大学数学系的江金生教授和程晓良教授对该书进行了精心审校，并提出了许多宝贵的修改意见，对此作者深表感谢。同时感谢浙江大学城市学院信息与计算科学系的全体老师及吴明晖、张琦、郑辉、沈雷赟、马耀华、徐晋、刘振华、郑书泉、封佳栋、崔良峰等同志对本教材出版的关心和支持。

由于编者水平有限，不妥或错误之处在所难免，恳请广大读者、同行和有关专家对本书进行批评指正，以便于今后进一步修改。

编者

目　录

第1章 误 差

1.1 数值计算的基本概念

由计算器或计算机所完成的算术运算不同于代数和微积分课程中的算术运算。我们把用计算机进行各种科学技术计算的工作，称为**科学计算**。科学计算与科学实验及理论研究是现代科学的三大组成部分，而数值计算是科学计算的关键环节。下面给出用计算机解决科学计算问题时所经历的几个环节，如图 1.1 所示。

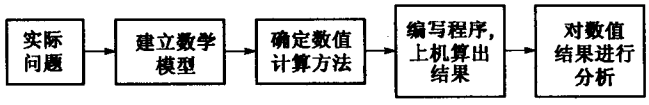


图 1.1 计算机解决科学计算问题的几个环节

从实际问题出发，应用相关的科学知识和数学理论建立数学模型的这一过程，通常作为应用数学的任务。而根据数学模型提出求解的数值计算方法直到编写程序、上机算出结果，这一过程则是数值计算的任务。并且对数值结果进行分析这一过程又是二者共同关注的问题。数值计算的研究对象即求解各种数学问题的数值方法的设计、分析，以及有关的数学理论与软件实现。所得到的数值方法又称为算法，应用算法得到的解为数值解，而数值解对精确解的“近似”程度可用误差来衡量。因此，误差成为算法研究的核心问题。

数值计算是用计算机进行数学计算的，并且计算机的运算速度高，可以承担各种计算工作。因此，很多人认为只要把涉及到的一些数学公式用一种计算机语言正确编程，计算机就一定能给出正确的结果，但事实上是这样的吗？

例1.1 我们知道，行列式解法的 Cramer 法则原则上可用来求解线性方程组。使用这种方法解一个 n 元方程组，需要 $n+1$ 个 n 阶行列式的值，总共需要 $n!(n+1)(n-1)$ 次乘法。当 n 充分大时，计算量是相当惊人的。例如一个 20 元的不算太大的方程组，大约要做 10^{21} 次乘法，这项计算即使用每秒千亿次的计算机来实现，也需要连续工作上百年才能完成。当然这是完全没有实际意义的。其实，求解线性方程组有许多实用的算法。例如在第 5 章介绍的消元法，对于一个 20 元的方程组，利用一台小型计算机就能很快地求解出来。

这个例子告诉我们，在数值计算中要注意计算量的分析。另外，计算机的内存也是有限的。因此，在设计算法时，也要尽量节省存储空间。

例 1.2 一元二次方程 $x^2+(\alpha+\beta)x+10^9=0$（其中 $\alpha=10^{-9}$，$\beta=-1$）有两个互异实根：$x_1=10^9$，$x_2=1$。但是若直接引进求根公式：

$$x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

在尾数八位的浮点计算机上进行运算，则得 $x_1=10^9$，$x_2=0$，其中一个根很明显是错误的。

出现这一错误的原因是受机器字长的限制所引起的误差。因此，在设计算法时，也要注意算法的误差分析。

例 1.3 积分：

$$I_n=\mathrm{e}^{-1}\int_0^1 x^n\mathrm{e}^x\mathrm{d}x \qquad (n=0,\ 1,\ 2,\cdots)$$

的值必定落在区间[0, 1]中，而且随着 n 的增大而减小。用分部积分法易得递推关系式：

$$I_n=1-nI_{n-1} \qquad (n=1,\ 2,\cdots) \tag{1.1}$$

若在尾数八位的浮点计算机上先计算出 $I_0=1-\mathrm{e}^{-1}$ 的近似值（具有八位有效数字），然后利用递推式(1.1)依次算出 I_1, I_2, I_3,$\cdots$ 的近似值，那么所得结果见表 1.1。I_{13} 的近似值小于 0，显然是错误的。此后，随着 n 的增大，错误越来越严重。

表 1.1 积分 I_n 的近似值

n	I_n 近似值	n	I_n 近似值
0	0.63212056	8	0.10097920
1	0.36787944	9	0.091187200
2	0.26424112	10	0.088128000
3	0.20727664	11	0.030592000
4	0.17089344	12	0.63289600
5	0.14553280	13	−7.2276480
6	0.12680320	14	102.18707
7	0.11237760	⋮	

出现这一错误，是由于受机器字长的限制所引起的误差在计算过程中的传播造成的。一个好的算法应能控制误差的传播，即应该是所谓的数值稳定的算法。

通过上面几个例子，可以初步看出，计算数学与纯数学有着明显的不同。而数值计算更是一门与计算机使用密切结合的、实用性很强的数学课程。概括起来，数值计算有以下四个特点：

- **第一，面向计算机**。要根据计算机特点提供切实可行的有效算法，即算法只能包括加、减、乘、除运算和简单的逻辑运算，这些运算是计算机能直接处理的运算。
- **第二，有可靠的理论分析**。设计的算法应能任意逼近并达到精度要求，对近似算法要保证收敛性和数值稳定性，还要对误差进行分析。这些都建立在相应的数学理论的基础上。
- **第三，算法要尽量节省存储空间，减少计算工作量**。这关系到算法能否在计算机上实现。
- **第四，任何算法都要有数值实验**。即任何一个算法除了从理论上要满足上述三点之外，还要通过数值实验证明是行之有效的。有的方法在理论上虽不够严格，但通过实际计算、对比分析等手段，证明是行之有效的方法，那么也应该采用。

本门课程将着重介绍进行科学计算所必须掌握的一些最基本、最常用的数值算法，其内容涉及了插值法、数值微分与积分、一元非线性方程、线性方程组及常微分方程初值问题。

1.2 计算机中的浮点数

1.2.1 浮点数的基本概念

要理解数值计算的基本原理，我们必须深入了解一下计算机是如何进行数学计算的，这有助于构造和分析各种数值分析。

数学运算主要是实数运算，我们都知道，任一实数可表示为

$$x=\pm 10^{s}\times 0.d_1d_2\cdots \tag{1.2}$$

其中 $d_i\in\{0,1,2,3,\cdots,9\}(i=1,2,\cdots)$，$s$ 为整数，由式(1.2)表示的 x 称为**十进制浮点数**。同样，可定义 β **进制浮点数**为

$$x=\pm\beta^{s}\times 0.d_1d_2\cdots d_t \tag{1.3}$$

其中 $d_i\in\{0,1,2,3,\cdots,\beta-1\}(i=1,2,\cdots,t)$。这里 t 为正整数，是计算机的**字长**；β 叫做这个数的**基**；s 是**阶**，是一个整数，取值正、负或零。并且满足 $L\leqslant s\leqslant U$，L 和 U 为固定整数，对于不同的计算机，t、L 和 U 是不同的。$d_1,d_2,\cdots,d_t$ 是尾数，由 t 位小数构造。若 $d_1\neq 0$，则称该浮点数为**规格化浮点数**。由式(1.3)表示的数 x 称为 t 位 β 进制浮点数，这样一些数的全体：

$$F(\beta,t,L,U)=\{\pm\beta^{s}\times 0.d_1d_2\cdots d_t,\ 0\leqslant d_i\leqslant\beta-1,\ d_1\neq 0,\ L\leqslant S\leqslant U\}\cup\{0\}$$

称为**机器数系**，它是计算机进行实数运算所用的数系。一般 β 取 2、8、10 和 16。集合 F 可用 β、t、L、U 四个参数来刻画。对于不同的机器，这四个值不一定相同，最常见的有(2, 56, –64, 64)。它表示一个二进制数集合，每个数有 56 位有效小数，阶码由–64 到 64。

“数”在今天的计算机是用二进制表示的，一个非零的二进制数的一般描述形式为

$$\pm 2^{s}\times 0.d_1d_2\cdots d_t$$

对于一个特定的机器来说，尾数的位数 t 是固定的，也称其机器精度有十个 β 进位数字。浮点数中阶的上界为 U，下界为 L。不难验证 F 中任意不为零的数 f，有

$$m\leqslant|f|\leqslant M$$

其中 $m=2^{L-1}$，$M=2^{U}(1-2^{-t})$。所以计算机上的数值运算会有“溢出”的现象。当运算的结果超过集合 F 的上界时称为“**上溢**”。当运算的结果超过集合 F 的下界时称为“**下溢**”。例如在数系 $F(2,4,-99,99)$ 中，$M=2^{99}\times 0.9999$，$m=2^{-99}\times 0.0001$。在上溢时，由计算机中断程序进行处理；在下溢时，计算机将此数用零表示继续执行程序。无论是上溢还是下溢，都称为溢出错误。通常，计算机把尾数为 0 且阶数最小的数表示为数零。

1.2.2 实数在计算机中的转换

设非零实数 x 是计算机接收的数，则计算机对其进行的处理方法是

- 若 $x\in F(\beta,t,L,U)$，则原样接收 x。
- 若 $x\notin F(\beta,t,L,U)$，但 $m\leqslant|x|\leqslant M$，则用 $F(\beta,t,L,U)$ 中最接近 x 的数 $fl(x)$ 表示并记录 x，以便后续处理。

计算机对接收的数只能做加、减、乘、除四则运算，其运算方式是

- 加减法：先向上对阶，后运算，再舍入。
- 乘除法：先运算，再舍入。

例如，某一计算机中的数系为 $F(10,\ 4,\ -90,\ 90)$，如下所示的

$$fl(x_1)=0.2337\times10^{-1},\quad fl(x_2)=0.3364\times10^{2}$$

是计算机接收到的两个实数，则有

$$\begin{aligned}fl(x_1+x_2)&=fl(0.2337\times10^{-1}+0.3364\times10^{2})\\&\underline{\underline{\text{对阶}}}\ fl(0.0002337\times10^{2}+0.3364\times10^{2})\\&\underline{\underline{\text{运算}}}\ fl(0.3366337\times10^{2})\\&\underline{\underline{\text{舍入}}}\ 0.3366\times10^{2}\end{aligned}$$

$$\begin{aligned}fl(x_1x_2)&=fl(0.2337\times10^{-1}\times0.3364\times10^{2})\\&\underline{\underline{\text{运算}}}\ fl(0.7861668\times10^{0})\\&\underline{\underline{\text{舍入}}}\ 0.7862\times10^{0}=0.7862\end{aligned}$$

由于计算机对接收到的数进行转换，往往使一些计算公式经过上机编程后得不到正确的结果。但只要我们注意到计算机的这些特点，就可以使用科学的计算方法解决这一问题。

1.2.3 MATLAB 中的浮点数

MATLAB 软件使用的是 IEEE 国际通用标准的双精度二进制数。使用单精度数固然可以节省存储空间，但是在现代的计算机上并不能提高运行速度。IEEE 双精度二进制数使用 64 个位存储一个数。每个位上的电器元件有高和低两个状态，低电位代表 0，高电位代表 1。其中位的分配如下：

尾数符号	尾数	阶码（包括符号）
1	52	11

IEEE 标准的双精度二进制数采用的形式是

$$x=\pm2^{e}\times(1+f)$$

其中的尾数是满足

$$0 \leqslant f < 1$$

的二进制小数。也就是 $2^{52} \cdot f$ 为正整数且满足

$$0 \leqslant 2^{52} \cdot f < 2^{52}$$

指数满足

$$-1022 \leqslant e \leqslant 1023$$

指数部分的存储形式是 $e+1023$，这样可以同时记录指数的符号。

1.3　数值计算的误差

在研究算法时，必须注重误差分析，否则，一个合理的算法也可能得出错误的结果。只要我们能对误差进行合理的处理和控制，就可以有效地解决问题。

1.3.1　误差的来源

从图 1.1 可以看出，每个环节都会产生误差。误差来源主要有以下四个方面：

- 模型误差（描述误差）
- 观测误差
- 截断误差
- 舍入误差

下面我们一一进行分析。

1. 模型误差（描述误差）

在对实际问题进行抽象与向量化并建立数学模型时，总是在一定条件下抓住主要因素，忽略次要因素。这样得到的模型是一种理想化的数学描述，它与实际问题之间总存在误差。这样的误差就称为**模型误差**或描述误差。

例 1.4　通常用

$$S(t) = \frac{1}{2} \cdot gt^2, \qquad g \approx 9.81\ \mathrm{m/s^2}$$

来描述自由落体下落时距离和时间的关系。设自由落体在时间 t 的实际下落距离为 $\tilde{S}$，则把 $\tilde{S} - S$ 叫做“模型误差”。

2. 观测误差

在数学模型或各种计算公式中包含着一些已知数量（称为原始数据），这些数量往往是由观测或实验得到的，例如温度、时间、电压等，它们和实际测量结果之间有误差，这种误差称为**观测误差**。

例 1.5 设一根铝棒在温度 t 时的实际长度为 L_t，在 $t=0$ 时的实际长度为 L_0。用 l_t 来表示铝棒在温度为 t 时的长度计算值，并建立一个数学模型 $l_t = L_0(1+\alpha t)$，其中 α 是由实验观察到的常数，$\alpha = (0.0000238 \pm 0.0000001)/℃$。则称 $L_t - l_t$ 为“模型误差”，0.0000001/℃ 是 α 的“观测误差”。

3. 截断误差

根据实际问题建立的数学模型，在很多情况下很难得到准确解，这就需要选用适当的数值计算求其近似解。数值计算方法所得到的近似解与实际问题准确解之间的这种误差，称为**截断误差**或方法误差。

例 1.6 有一元函数 $f: R \to R$，则 f 在 x_0 的导数定义为

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0+h) - f(x_0)}{h}$$

所以在 x_0 的导数值可以用算法：

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h} \tag{1.4}$$

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h} \tag{1.5}$$

来计算。但这样的结果与实际解是有误差的，由泰勒（Taylor）公式有

$$f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(x_0) + O(h^3)$$

所以有

$$\frac{f(x_0+h) - f(x_0)}{h} = f'(x_0) + \frac{h}{2} f''(x_0) + O(h^2)$$

$$T_1 = \frac{f(x_0+h) - f(x_0)}{h} - f'(x_0) = \frac{h}{2} f''(x_0) + O(h^2)$$

T_1 称为算法(1.4)的截断误差，它来源于算法中有限的差分替代了无限的极限过程。类似地，可以分析算法(1.5)的截断误差，其结果为

$$T_2 = \frac{h^2}{3!} f'''(x_0) + O(h^3)$$

上述截断误差的分析表明算法(1.5)是比算法(1.4)更好的算法，因为对同样的步长 h（$\ll 1$），算法(1.5)更接近于 $f'(x_0)$。

计算方法的截断误差是数值计算中误差的重要来源，然而并不是唯一的。如果在实验中确定已将 h 取到足够小，特别是在高阶导数的计算中，就会发现当 h 小到一定程度之后，数值计算结果的误差不但不再减小，反而会变大，请见图 1.2。事实上，当步长 h 过小时，计算结果的误差变大则是由舍入误差引起的。

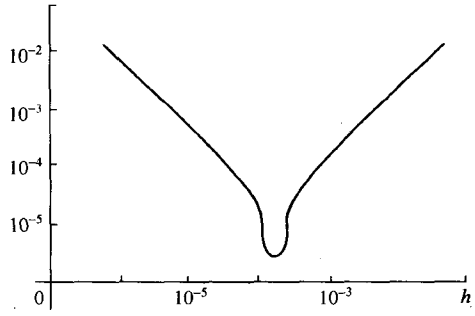


图 1.2 最佳步长

4. 舍入误差

由于计算机数系 $F(\beta,t,L,U)$ 是离散的有限集，计算机接收和运算数据时总是将位数较多的数舍入成一定位数的机器数，这样产生的误差就是**舍入误差**。例如，在尾数四位的浮点计算机上用 0.3333 表示1/3，产生的误差：

$$R=1/3-0.3333=0.000033\cdots$$

就是舍入误差。

每一步的舍入误差都是微不足道的，但经过计算过程的传播和积累，舍入误差甚至可能会“淹没”所要的真解。

虽然我们了解了以上几种误差，并且这些知识对于数值计算都是有帮助的，但是前两种误差往往不是计算工作所能独立解决的。因此，在数值计算过程中通常只能讨论截断误差和舍入误差。

1.3.2 绝对误差、相对误差、有效数字

在数值计算中，误差虽然不可避免，但我们总是希望计算结果能足够精确，这就需要估计误差。为了从不同的侧面表示近似数的精确程度，通常运用绝对误差、相对误差和有效数字的位数来描述。

定义 1.1 设 x^* 为准确值，x 是 x^* 的一个近似值，称

$$e=x^*-x$$

为近似值 x 的**绝对误差**，简称误差。

由于准确值 x^* 未知，因而误差 e 通常是无法确定的，人们只能根据测量工具或计算过程，事先估计出误差的取值范围，即误差绝对值的一个上界。

定义 1.2 设存在一个正数 ε，使

$$|e|=|x^*-x|\leqslant\varepsilon \tag{1.6}$$

则称 ε 是近似值 x 的**绝对误差限**，简称误差限或精度。

因为在任何情况下都有

$$|x^* - x| \leqslant \varepsilon$$

即

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

这就表明 x 在 $[x^* - \varepsilon, x^* + \varepsilon]$ 这个区间内，因此用

$$x = x^* \pm \varepsilon$$

来表示近似值 x^* 的精确度或准确值所在的范围。

例 1.7 用一把有毫米（mm）刻度的米尺来测量桌子的长度。读出来的长度 $x^* = 1235$ mm，它是桌子实际长度 x 的一个近似值。由米尺的精度知道，这个近似值的误差不会超过半个毫米，则有

$$|x^* - x| = |1235 - x| \leqslant \frac{1}{2} \text{ (mm)}$$

即

$$1234.5 \leqslant x \leqslant 1235.5$$

这表明 x 在 $[1234.5, 1235.5]$ 这个区间内，可写成

$$x = 1235 \pm 0.5 \text{ (mm)}$$

这个例子说明绝对误差是有量纲单位的。例如，工人甲平均每生产一百个零件有一个次品，而工人乙平均每生产五百个零件有一个次品。他们的次品都是一个，但显然乙的技术水品要比甲高。这就启发人们除了要看次品的多少之外，还必须注意到产品的合格率，甲的次品率是百分之一，而乙的次品率是五百分之一。显然乙产品的质量要比甲好，为反映这种近似程度，我们接着再引入如下的相对误差的概念。

定义 1.3 称

$$e_r = \frac{e}{x^*} = \frac{x^* - x}{x^*}$$

为近似值 x 的**相对误差**。在实际运算中，由于准确值 x^* 总是不知道的，所以也把 $e_r = \dfrac{e}{x} = \dfrac{x^* - x}{x}$ 记为近似值 x 的相对误差，条件是 e_r 比较小。

相对误差是一个无量纲量，通常可用百分数表示，相对误差的绝对值越小，近似程度越高。例如前面所述，甲生产的产品的相对误差为 $e_r(\text{甲}) = 1\%$，乙生产的产品的相对误差为 $e_r(\text{乙}) = 0.2\%$，所以乙产品的质量比甲好。同样，由于准确值 x^* 通常是未知的，一般我们不能确定出 e_r 的准确值，而只能估计它的大小范围。

定义 1.4 如果存在正数 ε_r，使

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

则称正数ε_r为x的**相对误差限**。

相对误差限不如绝对误差限容易得到，在实际计算中常借助绝对误差来求之，并取分母中的准确值x^*为近似值x，即取$\varepsilon_r = \dfrac{\varepsilon}{|x|}$。

为了给出一种近似数的表示方法，使之既能表示数的大小，又能表示其精确程度，可以引入有效数字的概念。例如，近似值x可以写成图 1.3 所示的形式。若x某位数的半个单位是它的误差限，而且从该位数字到x最左边的那个非零数字共有n位，那么我们把这n位数字称为有效数字。并且说近似值x具有n位有效数字。

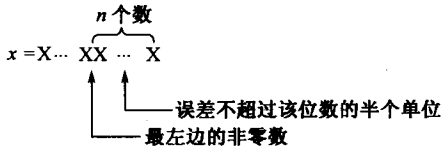


图 1.3 有效数字

定义 1.5 设准确值为x^*，近似值$x = \pm 0.d_1 d_2 \cdots d_n \times 10^m$，其中$d_i \in \{0, 1, 2, \cdots, 9\}$（$i = 1, 2, \cdots, n$），$d_1 \neq 0$，$m$为整数，如果

$$|e| = |x^* - x| \leqslant \frac{1}{2} \times 10^{m-n} \tag{1.7}$$

则称近似值x有n位**有效数字**，其中$d_1, d_2, \cdots, d_n$都是x的有效数字，也称x为有n位有效数字的近似值。

由定义 1.5 可知，π的近似值 3.14 和 3.1416 分别有 3 位和 5 位有效数字。由式(1.7)可知，有效数字越多，绝对误差越小。而有效数字与相对误差的关系，则有如下的结论。

定理 1.1 设近似值$x = \pm 0.d_1 d_2 \cdots d_n \times 10^m$有$n$位有效数字，则其相对误差限为

$$\varepsilon_r = \frac{1}{2d_1} \times 10^{-n+1} \tag{1.8}$$

证：由x有n位有效数字可知

$$|e| = |x^* - x| \leqslant \frac{1}{2} \times 10^{m-n}，\ 而 |x| \geqslant a_1 \times 10^{m-1}$$

故有

$$|e_r| = \left|\frac{x^* - x}{x}\right| = \frac{\frac{1}{2} \times 10^{m-n}}{d_1 \times 10^{m-1}} = \frac{1}{2d_1} \times 10^{-n+1}$$