21世纪高等院校教材

数值计算方法

黄明游　刘　播　徐　涛　编

科学出版社
www.sciencep.com

21世纪高等院校教材

数值计算方法

黄明游　刘　播　徐　涛　编

科学出版社

北京

内 容 简 介

本书旨在介绍科学与工程计算中一些基本数学问题的实用计算方法，主要内容包括：线性代数方程组的直接解法和迭代法，矩阵特征值与特征向量的计算，非线性方程组和最优化问题的计算方法，函数插值与曲线拟合方法，数值积分，离散傅里叶变换快速算法，常微分方程初值问题的数值积分法，解偏微分方程的差分法和有限元法.

本书可作为理工科各专业本科生、研究生的数值计算方法课程教材，也可作为科学与工程技术人员学习、应用科学计算方法的参考书.

图书在版编目(CIP)数据

数值计算方法/黄明游等编. —北京：科学出版社，2005
(21 世纪高等院校教材)
ISBN 7-03-015763-X

I. 数… II. 黄… 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(2005) 第 067132 号

责任编辑：李鹏奇 吕 虹 祖翠娥 / 责任校对：宋玲玲
责任印制：安春生 / 封面设计：陈 敬

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
源海印刷有限责任公司印刷
科学出版社发行 各地新华书店经销
*
2005 年 8 月第 一 版 开本：B5(720×1000)
2005 年 8 月第一次印刷 印张：15 1/4
印数：1—5 000 字数：292 000
定价：22.00 元
(如有印装质量问题，我社负责调换〈路通〉)

前　言

由于计算机的发展和普及，科学计算已成为解决各类科学技术问题的重要手段. 因此，掌握科学计算的基本原理和方法是当今科学技术工作者不可缺少的本领和技能之一.

本书为适应理、工科大学有关专业开设“数值计算方法”课程的需要而编写. 此课程的内容极为广泛，目前计算数学(或信息与计算科学)专业一般分三门课程(数值代数、数值逼近、微分方程数值解法)讲授其内容，需要花费近 200 个学时. 然而，对于其他有关专业上述作法是难于做到的，同时也并非必要. 编写这本教材的目的，是为便于在一个学期(约用 68 个学时)之内讲授科学与工程计算中经常遇到的一些基本问题的数值计算方法，并通过选讲一些典型、通用的数值方法来阐明构造方法的基本原理和技巧.

这本教材是编者及同事在吉林大学为理科(应用数学、力学、计算机与软件等专业)和工科(机械、汽车、电讯等专业)的本科生和研究生讲授《数值计算方法》课的实践基础上编写的. 内容基本上覆盖了数值代数、数值逼近和常(偏)微分方程数值解法的内容. 我们本着实用和精练的原则选择题材，着重于介绍构造计算方法的基本思想和算法设计的技巧，使学生能够举一反三，具有一定的构造方法和设计算法的能力. 其次，深入浅出、简要地介绍数值方法中的一些基本概念和理论结果，使学生具有分析计算方法性能和按实际情况选用方法的能力. 另外，各章配有适量的算例和习题，帮助学生掌握课程的内容和锻炼实际计算的能力.

在编写过程中，我们参考了国内已出版的同类教材(参考文献[1]~[6])，吸取了它们的许多优点，但在题材的取舍和内容的阐述方面有较大的变化. 另外，本着推陈出新和与时俱进的精神，本书也适度地增添了一些新内容，如最优化问题的计算方法和解偏微分方程的有限元方法等. 全书主体内容为绪论和第 1~8 章. 讲授本书全部内容约需 68 个学时，如果舍去 1~8 章中带“*”号的内容，可在 51 个学时内讲授完此课程. 此外，本书亦可作为科学技术人员学习、应用科学计算方法的参考书.

此书的第 1、2、5 章由刘播教授编写，第 3、4、6 章由徐涛教授编写，绪论、第 7、8 章由黄明游教授编写. 另外，黄明游教授主持了本书的编撰和修改、定稿工作.

由于编者水平、经验所限，时间比较匆忙，一定有些疏漏乃至错误的地方，欢迎读者批评指正.

编　者

2005 年 5 月 29 日

目　录

绪　论

0.1　数值计算方法的内容、特点与学习方法

科学计算是人类从事科学活动和解决科学技术问题不可缺少的手段. 在计算技术与计算机得到迅猛发展的今天, 人们有了快速数字电子计算机的工具, 科学计算被推向科学活动的前沿, 上升为一种重要的科学方法. 用数字电子计算机进行科学计算, 解决科学技术问题, 大体上经历如图 0.1 所示的几个步骤.

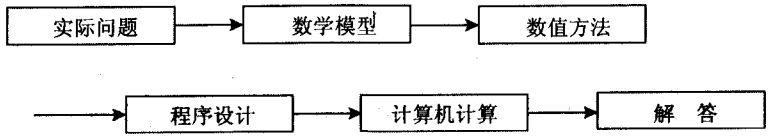

图 0.1　科学计算的流程图

将科学技术中的实际问题转化为数学问题, 即根据有关的科学理论知识和数学理论、方法建立数学模型, 这是进行科学计算的前题或先决条件. 然而, 如何运用计算机获得这些数学模型 (问题) 的满足精度要求的解答尤为重要, 它是使实际问题得以解决的关键. 事实上, 许多数学问题 (高次代数方程的根、许多函数的积分值和大多微分方程的解) 是没有办法求出其精确解的, 因此只好通过数值方法求其近似解 (值). 另外, 很多复杂的计算问题 (如大型线性方程组的求解) 是人工难以胜任的, 只有利用快速电子计算机才能完成大规模、复杂的科学计算工作.

现代数字电子计算机无论它如何先进, 但它所能执行的只不过是简单的四则算术运算和逻辑运算, 用它直接解决不了数学问题. 用数字电子计算机进行科学计算的一个关键, 就是要将数学模型 (问题) 的求解归结、转化为计算机所能够执行的运算和操作, 即需采用恰当的数值计算方法, 此乃本课程所要讨论的内容.

数值计算方法(又称 **数值分析**) 是数学的一个分支, 它以数字计算机求解数学问题的方法与理论为研究对象, 其内容包括: 函数插值, 数值微分与积分, 线性方程组的解法, 矩阵特征值与特征向量的计算, 非线性方程 (组) 的解法与最优化问题的计算方法, 常微与偏微分方程的数值解法等, 此外还包括有关计算方法可靠性的理论研究, 如方法的收敛性和稳定性分析与误差估计等.

数值计算方法的应用极为广泛, 无论是日常工农业生产还是国防尖端科学技术的研究, 如大、中型机电产品的优化设计、重大工程项目的设计、地质勘探与油田开发、气象预报与地震预测、新型尖端武器的研制和航空与航天的发展等都离不开它, 近年来还被应用到医学、生物学及经济管理、金融和社会学等领域. 另外,

它作为一种科学方法渗透到不同的科学领域，形成了一些诸如计算力学、计算物理、计算化学、数字图像处理、计量经济学等交叉学科方向. 随着科学技术的突飞猛进，计算技术和数值计算方法将有更加广阔的发展前景.

数值计算方法是一门与计算机应用紧密结合、实用性很强的数学课程，它所涉及的数学问题面很广、内容非常丰富，亦有其自身的体系. 它既有数学的高度概括(抽象)性和严密的科学性，又非常讲究实用性并具有高度的技巧性. 本课程所具有的特点，概括起来有以下几个方面：

第一，面向计算机，重点研究数字计算机上使用的计算方法. 如所提供的求解算法最终只能包括四则算术运算和逻辑运算，这样才能由计算机来实现. 另外，计算机的水平 (如速度容量、串行与并行等) 也会对数值计算方法的选择乃至研究产生重要影响.

第二，注重实用性和计算效率. 在纯数学中，往常只介绍问题解的存在性和唯一性，至于如何求解和计算很少过问. 例如，在线性代数里对线性方程组何时存在唯一解有明确的结论，同时也给出了解的精确表达式 (Cramer 公式)，后面将说明用此公式求解是不现实的. 计算效率包括计算时间、所需存储量和编制计算机程序的难易程度等.

第三，讲究算法的技巧性. 在求解算法 (包括计算公式和计算步骤) 的设计中，必须讲究技巧. 因为算法上的区别可能会对误差的传播和计算结果的精度产生重要的影响，后面会用例子来说明这一点.

第四，重视方法的理论研究. 为了确保可获得任意指定精度的解，要求所提供的计算方法具有收敛的性质，相应的算法能够抑制舍入误差的干扰，即算法是稳定的，同时应给出近似解相对于精确解的误差界限. 这些都属于有关数值计算方法可靠性的理论研究内容.

根据数值计算方法的上述特点，在学习此课程时，我们首先要掌握构造方法的原理、思想，注意设计算法的技巧并要与计算机的实际密切结合，也要重视有关计算方法基础知识和数学理论的学习. 其次，要重视实践，通过算例和动手计算，学会怎样使用数值方法在计算机上解决各类数学计算问题. 为了掌握本课程的内容，需要做一定数量的习题和计算实习题. 另外，由于课程内容包括微积分、线性代数和常偏微分方程的数值计算方法，因此需要读者初步了解这几门课程的基本内容.

0.2　计算机的算术运算、若干计算例题

介绍计算机的算术运算，需从数的计算机表示开始. 目前，数字电子计算机大多数采用二进制浮点系统，每个实数 x 被表示为

$$x = \pm 0.d_1 d_2 \cdots d_k \times 2^p. \tag{0.2.1}$$

这里 $d_1=1$, d_i 为 0 或 $1, i=2,3\cdots,k$, 整数 p 满足 $-m\leqslant p\leqslant M$. 零的浮点表示可能是不同的, 通常用 $0=\pm 0.00\cdots 0\times 2^{-m}$ 表示. 因此, 一部给定的二进制浮点计算机, 只能表示所有形如 (0.2.1) 式的有限数集 $S=S(k,m,M)$, 这是实数轴上的不等距有限点集. 为了说明问题, 我们考虑一个特殊的计算机, 它的 $k=3,m=1,M=2$, 那么这个计算机所能表示的浮点数的集合 $S=S(3,1,2)$ 只有如图 0.2 所示的 33 个点.

图 0.2 数集 $S(3,1,2)$

一个实数 x, 在计算机上只能按一定规则用 S 中最接近 x 的数来近似表示, 这就带来了数的 **表示误差**. 在我们这个小计算机上, $|x|>3.5$ 就溢出了 (上溢出), 而 $|x|<0.25$ 为机器零 (下溢出).

计算机只能对浮点数集 S 中的数做加、减、乘、除四则运算. 由于用二进制数作四则运算我们不习惯, 所以这里不妨用十进制的 4 位浮点数的四则运算来说明, 因为它们的情况是一样的. 设 S_0 是所有形如 $\pm 0.d_1d_2d_3d_4\times 10^p$ 的 4 位十进制浮点数的集合, 其中 $1\leqslant d_1\leqslant 9, 0\leqslant d_i\leqslant 9, i=2,3,4$, 整数 p 满足 $-9\leqslant p\leqslant 10$. 下面举例说明 S_0 上的算术运算, " $\rightarrow$ " 表示在运算器上的运算过程和结果:

(1) $0.1984\times 10^4+0.2008\times 10^2$
$\rightarrow 0.1984\times 10^4+0.0020\times 10^4$
$\rightarrow 0.2004\times 10^4$;

(2) $0.1984\times 10^4+0.9876\times 10^{-1}$
$\rightarrow 0.01984\times 10^4+0.0000\times 10^4$
$\rightarrow 0.1984\times 10^4$;

(3) $0.1984\times 10^{-6}-0.1976\times 10^{-6}$
$\rightarrow 0.0008\times 10^{-6}$
$\rightarrow 0.8000\times 10^{-9}$;

(4) $(0.5678\times 10^3)\times(0.6789\times 10^{-5})$
$\rightarrow (0.5678\times 0.6789)\times 10^{-2}$
$\rightarrow 0.38547942\times 10^{-2}$
$\rightarrow 0.3855\times 10^{-2}$;

(5) $(0.5678\times 10^4)\div(0.4567\times 10^{-5})$
$\rightarrow 0.5678\times 10^9\div 0.4567$
$\rightarrow 0.12432669\times 10^{10}$

$\rightarrow 0.1243 \times 10^{10}$.

由这些算例可以看出，计算机上的算术运算不可避免地带来舍入误差. 值得注意，应尽量避免像 (2), (3), (5) 那样的情形发生：绝对值相差悬殊的两个数做加、减，很接近的两数相减，相对被除数来说绝对值很小的数做除数. 这些是数值计算中防止出现严重运算误差的基本原则. 在 (5) 中，当除数为 $0.456\,7 \times 10^{-6}$ 时，商 $0.124\,3 \times 10^{10}$ 已不属于数集 S_0, 这就是所谓“溢出”.

了解了计算机的算术运算，就不难理解为什么要讨论计算方法. 下面，我们通过几个简单的计算问题，进一步说明计算方法和算法设计的重要性.

例 0.1　在前面所述 4 位十进制浮点计算机 (数集 S_0) 上求解如下一元二次方程

$$x^2 - 18x + 1 = 0.$$

按求根公式，此方程的两个根是

$$x_1 = 9 - \sqrt{80}, \qquad x_2 = 9 + \sqrt{80}.$$

在所用的计算机上，$\sqrt{80} = 0.8944 \times 10^1$. 按求根公式计算

$$x_1 = 0.5600 \times 10^{-1}, \qquad x_2 = 0.1794 \times 10^2.$$

另外，若换一公式计算 x_1, 即利用 $x_1 x_2 = 1$, 可得

$$x_1 = \frac{1}{9 + \sqrt{80}} = 0.5574 \times 10^{-1},$$

它的误差不超过 $0.0001 \times 10^{-1} \times 0.5 = 0.0005 \times 10^{-2}$. 而前面按求根公式算出的 x_1, 其误差不超过 $0.01 \times 10^{-1} \times 0.5 = 0.0005$. 显然，在数学上公式 $x_1 = 9 - \sqrt{80}$ 和 $x_1 = 1/(9 + \sqrt{80})$ 是等价的，但在计算上它们却不相同. 从计算结果看到，后者优于前者，原因是在前一公式中出现了接近的两个数相减.

例 0.2　考虑对任意给定的 x, 计算代数多项式

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n \tag{0.2.2}$$

的值的问题. 显然，(0.2.2) 式等价于

$$P_n(x) = (((a_0 x + a_1)x + a_2)x + \cdots + a_{n-1})x + a_n. \tag{0.2.3}$$

但从计算角度看，两个公式却有很大的区别. 按公式 (0.2.2), 需先算出 $x^2, x^3, \cdots, x^n$, 共需 $n-1$ 次乘法，还需将它们保存起来，这要额外占用 $n-1$ 个存储单元，然后再相乘 (n 个乘法) 、相加 (n 个加法), 合起来共需 $2n-1$ 个乘法和 n 个加法. 然而，按公式 (0.2.3) 计算仅需 n 个乘法和 n 个加法，并无须增加存储单元. 公式 (0.2.3)

和 (0.2.2) 的差别还不止这些. 譬如, 在使用前述计算机的情形, 当 $x=10$ 时按公式 (0.2.2) 计算就进行不下去, 因为在计算 10^{10} 时已经溢出.

例 0.3 考虑线性代数方程组

$$Ax=b \tag{0.2.4}$$

的求解计算问题. 设系数矩阵 A 是 $n\times n$ 方阵, 其行列式 $D=\det(A)\neq 0$. 由线性代数知, 此方程组存在唯一解并由 Cramer 公式给出:

$$x_i=\frac{D_i}{D},\qquad i=1,2,\cdots,n. \tag{0.2.5}$$

用此公式求解, 共需计算 $n+1$ 个 n 阶行列式. 按 Laplace 展开法计算 n 阶行列式, 需作

$$n!\left(1+\frac{1}{2!}+\frac{1}{3!}+\cdots+\frac{1}{(n-1)!}\right)$$

个乘法, 姑且算作 $n!$ 个乘法. 这样, 不计加法, 用公式解 n 阶线性方程组, 共需 $(n+1)n!=(n+1)!$ 次以上的乘法运算. 对于一个 20 阶以上的线性方程组, 就需要 $(21)!\doteq 5.11\times 10^{19}$ 次以上的乘法运算. 设用每秒可做百万次乘法的计算机, 它每年可做 $365\times 24\times 3\,600\times 10^6\doteq 3.15\times 10^{13}$ 次乘法. 所以, 在每秒做百万次乘法的计算机上, 用 Cramer 公式解 20 阶的线性代数方程组, 所需要的计算时间在 $(5.11\times 10^{19})\div(3.15\times 10^{13})=1.62\times 10^6\doteq 162$ 万年以上!

例 0.4 考虑积分

$$I_n=\int_0^1 x^n\mathrm{e}^{x-1}\mathrm{d}x \tag{0.2.6}$$

的近似计算. 此积分满足递推关系式

$$I_n=1-nI_{n-1}, \tag{0.2.7}$$

我们首先算出 I_0 的近似值 $\overline{I}_0$, 再利用递推关系式 (0.2.7) 依次地算出.

$\overline{I}_0$	$\overline{I}_1$	$\overline{I}_2$	$\overline{I}_3$	$\overline{I}_4$	$\overline{I}_5$	$\overline{I}_6$	$\overline{I}_7$
0.632 1	0.368 0	0.264 0	0.208 0	0.168 0	0.160 0	0.040 0	0.720 0

容易看出, I_7 的计算结果 $\overline{I}_7$ 是不对的. 实际上, 当 n 增加时 $\{I_n\}$ 是下降的 (但这里 $\overline{I}_7>\overline{I}_6$). 另从

$$I_n<\mathrm{e}^{-1}\Big(\max_{0\leqslant x\leqslant 1}\mathrm{e}^x\Big)\int_0^1 x^n\mathrm{d}x=\frac{1}{(n+1)}$$

知道

$$I_7<\frac{1}{8}=0.1250.$$

为什么会产生这样一个面目全非的错误结果呢？这是由于 $\bar{I}_0$ 的误差 (表示舍入误差) 在按 (0.2.7) 式的递推过程中被逐次地乘以因子 $2,3,\cdots,7$, 致使误差急剧地增长. 然而, 若将 (0.2.7) 式改写为

$$I_{n-1}=(1-I_n)/n, \tag{0.2.8}$$

先计算出 I_7 的近似值 $\bar{I}_7=0.1124$, 再从 $\bar{I}_7$ 开始按 (0.2.8) 式递推, 可得如下值.

$\bar{I}_7$	$\bar{I}_6$	$\bar{I}_5$	$\bar{I}_4$	$\bar{I}_3$	$\bar{I}_2$	$\bar{I}_1$	$\bar{I}_0$
0.112 4	0.126 9	0.145 5	0.170 8	0.207 3	0.264 3	0.368 0	0.632 0
0.112 4	0.126 8	0.145 6	0.170 9	0.207 3	0.264 2	0.367 9	0.632 1

表中末行是精确值 I_n 的舍入结果. 我们看到按 (0.2.8) 递推计算出的 $\bar{I}_n(n=6,5,\cdots,0)$ 是令人满意的. 有趣的是, 即便从 $\bar{I}_7=0$ 开始按 (0.2.8) 递推, 也可得到 $\bar{I}_0=0.6320$. 这是由于原始误差每步依次被乘以因子 $\frac{1}{7},\frac{1}{6},\cdots,1$, 使得误差逐步地缩小.

以上例题说明, 即便数学模型的解具有理论公式, 仍然存在能否在计算机上应用和如何计算的问题. 因此, 我们必须研究计算方法, 并提供可行、有效的算法.

0.3　误差的来源和有关误差的基本概念

用数值方法在计算机上求解数学问题, 不可避免地会出现误差, 得到的一般是问题的近似解. 因此, 误差分析和误差的估计成为数值计算法研究的一项重要内容.

首先讨论一下误差的来源和分类. 一般来说, 数学模型仅是实际问题的一个近似, 它们之间的误差称为 **模型误差**. 另外, 模型中所含数据大都由实验或观测得到, 受条件限制也会有误差, 称此为 **观测误差**. 这里, 我们专门讨论数值计算中的误差, 不考虑上述两类误差, 即假定所利用的模型和数据是恰当、合理的. 这样一来, 数值计算出现误差的主要原因有以下两个方面:

(1) 在数值方法中, 通常是用近似公式 (方程) 代替精确公式 (方程), 由此所得解自然是问题的近似解, 这种原因造成的近似解与精确解之间的误差称为 **方法误差**. 例如, 为了计算函数值 $\mathrm{e}^x,|x|<1$, 我们用有限 Taylor 展式

$$P_n(x)=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}$$

近似代替 e^x, 此时的方法误差 (又称 **截断误差**) 为

$$R_n(x)=\mathrm{e}^x-P_n(x)=\frac{\mathrm{e}^\xi}{(n+1)!},\qquad |\xi|<1.$$

(2) 因计算机的字长有限，存在数的表示误差，运算中也会发生误差，称之为**舍入误差**. 这些误差在计算过程中的传播和积累，将会影响计算结果的精度，甚至得出面目全非、毫无意义的计算结果. 0.2 节中的例 0.4 表明，对于不同的计算公式和算法，舍入误差的传播影响可能截然不同. 由此看到，恰当地选择计算方法和合理地设计算法是非常重要的. 在讨论计算机的算术运算时，曾介绍过防范运算中舍入误差的若干基本原则.

下面，我们介绍有关误差的一些基本概念.

定义 0.1 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**, 或简称**误差**.

准确值往往是未知的，故无法给出所得近似值的绝对误差. 通常只能根据估算或测量给出其绝对误差的一个界限.

定义 0.2 设 $\varepsilon^* > 0$, 并满足

$$|e^*| = |x^* - x| \leqslant \varepsilon^*, \tag{0.3.1}$$

则称 ε^* 为近似值 x^* 的**绝对误差限**, 或简称**误差限**.

设 ε^* 是近似值 x^* 的误差限，由 (0.3.1) 式可知

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

换言之，

$$x \in [x^* - \varepsilon^*, x^* + \varepsilon^*].$$

有时，除考虑误差大小之外，还应考虑准确值本身的大小. 为此引进如下相对误差和相对误差限的概念.

定义 0.3 设 x^* 是 x 的一个近似值，则称比值

$$\frac{e^*}{x} = \frac{x^* - x}{x} \tag{0.3.2}$$

为近似值 x^* 的**相对误差**, 记作 e_r^*(实际应用中，常用 x^* 代替上式分母中的 x).

定义 0.4 设 ε^* 是近似值 x^* 的误差限，则称

$$\varepsilon_r^* = \frac{\varepsilon^*}{|x^*|}$$

为近似值 x^* 的**相对误差限**. 此时，有

$$\frac{|x^* - x|}{|x^*|} \leqslant \frac{\varepsilon^*}{|x^*|} = \varepsilon_r^*. \tag{0.3.3}$$

例 0.5 设 $x_1^* = 10$ 和 $x_2^* = 1000$ 分别都是近似值，它们相应的精确值 x_1 和 x_2 未知，但已知它们的误差限都是 1, 试比较这两个近似值的准确程度.

解　由定义 0.1, 定义 0.2, 有

$$x_1 \in (10-1,\ 10+1), \qquad x_2 \in (1000-1,\ 1000+1).$$

另外，根据定义 0.4, 近似值 x_1^* 和 x_2^* 的相对误差限分别为

$$\frac{|x_1^* - x_1|}{|x_1^*|} \leqslant \frac{1}{10} = 0.1$$

和

$$\frac{|x_2^* - x_2|}{|x_2^*|} \leqslant \frac{1}{1000} = 0.001.$$

可见，x_2^* 的准确度比 x_1^* 的准确度要高得多.

有效数字也是被用来表示近似值准确程度的一个常用概念，其准确定义如下.

定义 0.5　如果

$$|e^*| = |x^* - x| \leqslant \frac{1}{2} \times 10^{-n}, \tag{0.3.4}$$

则说 x^* 近似表示 x 准确到第 n 位，并从第 n 位起直到最左边的非零数字之间的一切数字都称为 **有效数字**, 并把有效数字的位数称为 **有效位数**.

例 0.6　取 π 的近似值为 $x^* = 3.14$, 则

$$|3.14 - \pi| \leqslant 0.0015926\cdots \leqslant \frac{1}{2} \times 10^{-2},$$

此时 x^* 具有 3 位有效数字. 若取 $x^* = 3.1416$ 作为近似，则

$$|3.1416 - \pi| \leqslant \frac{1}{2} \times 10^{-4},$$

这时 x^* 具有 5 位有效数字.

例 0.7　若 $x^* = 3587.64$ 是 x 的具有 6 位有效数字的近似值，试求 x^* 的误差限.

解　将 x^* 写成 (0.3.5) 式的形式

$$x^* = 3578.64 = 0.357864 \times 10^4,$$

则 $m = 4, n = 6$, 从而，由有效数字的定义，有

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

定义 0.5 表明，近似值的有效数字位数越多，那么其近似程度就越好. 下面是有效数字的一个等价定义：若将 x 近似值 x^* 表示成十进制浮点数的标准形式

$$x^* = 0.\alpha_1\alpha_2\cdots\alpha_n \times 10^m \qquad (\alpha_i : 0 \sim 9, \alpha_1 \neq 0), \tag{0.3.5}$$

如果

$$|x^* - x| \leqslant \frac{1}{2} \times 10^{m-n}, \tag{0.3.6}$$

则说近似值 x^***具有 n 位有效数字**. 这里 n 为正整数， m 为整数.

如下定理给出了有效数字与相对误差的关系.

定理 0.1 *若近似值 x^* 具有 n 位有效数字，则其相对误差满足*

$$e_r^* \leqslant \frac{1}{2\alpha_1} \times 10^{-(n-1)}; \tag{0.3.7}$$

反之，若 x^* 的相对误差 e_r^* 满足

$$e_r^* \leqslant \frac{1}{2(\alpha_1+1)} \times 10^{-(n-1)}, \tag{0.3.8}$$

则 x^* 至少具有 n 位有效数字.

证明 若 $x^* = 0.\alpha_1\alpha_2\cdots\alpha_n \times 10^m$ 具有 n 位有效数字，由定义

$$|x^* - x| \leqslant \frac{1}{2} \times 10^{m-n},$$

故相对误差满足

$$|e_r^*| = \frac{|x^* - x|}{|x^*|} \leqslant \frac{1}{2} \times 10^{m-n} \times \frac{1}{|x^*|}.$$

又因

$$|x^*| = |\alpha_1\alpha_2\cdots\alpha_n \times 10^{m-1}| \geqslant \alpha_1 \times 10^{m-1},$$

所以

$$|e_r^*| \leqslant \frac{1}{2\alpha_1} \times 10^{-(n-1)}.$$

反之，若 (0.3.8) 式成立，利用

$$|x^*| \leqslant (\alpha_1+1) \times 10^{m-1},$$

得

$$\begin{aligned} |x^* - x| &= |e_r^*| \cdot |x^*| \\ &\leqslant \frac{1}{2(\alpha_1+1)} \times 10^{-(n-1)} \cdot (\alpha_1+1) \times 10^{m-1} \\ &= \frac{1}{2} \times 10^{m-n}. \end{aligned}$$

于是，根据定义 x^* 具有 n 位有效数字. 定理证毕.

例 0.8 用 $x^* = 2.72$ 表示 e 具有 3 位有效数字的近似值，给出此近似值的相对误差限.

解　因 $x^* = 2.72 = 0.272 \times 10^1, \alpha_1 = 2, n = 3$, 由定理 0.1 的前半部分，有

$$|e_r^*| \leqslant \frac{1}{2\alpha_1} \times 10^{-(n-1)}$$
$$= \frac{1}{2 \times 2} \times 10^{-(3-1)} = 0.25 \times 10^{-2}.$$

例 0.9　要使 $\sqrt{20}$ 的近似值 x^* 的相对误差小于 0.001, 那么应取几位有效数字？

解　因 $4 < \sqrt{20} < 5$, 故可确定其表达式 (0.3.5) 中 $\alpha_1 = 4$. 若相对误差限满足

$$\varepsilon_r^* < 0.001,$$

由定理 0.1 的后半部分，有

$$|e_r^*| \leqslant \frac{1}{2(\alpha_1 + 1)} \times 10^{-(n-1)} = \varepsilon_r^*,$$

可见 n(有效数字位数) 应满足

$$\frac{1}{2(4+1)} \times 10^{-(n-1)} < 0.001,$$

由此解出 $n = 4$, 即应取 4 位有效数字.

习　题

1. 下列各数都是经过四舍五入得到的近似数，试分别指出它们的有效数字的位数，并给出它们的相对误差限

$$x_1^* = 1.1021; \qquad x_2^* = 0.031;$$
$$x_3^* = 56.430; \qquad x_4^* = 7 \times 10^5.$$

2. 求方程 $x^2 - 56x + 1 = 0$ 的两个实根，要求结果至少具有 4 位有效数字.

3. 设 $A = (1 - \cos 2^\circ) \times 10^7$, 先按此公式计算 A 的近似值 ($\cos 2^\circ = 0.9994$), 然后再用等价公式 $A = 2 \times \sin 1^\circ \times 10^7$ 计算 ($\sin 1^\circ = 0.0175$), 比较两次计算所得近似值的精确度.

4. 采用恰当公式计算

$$\sqrt{101.1} - \sqrt{101},$$

使结果至少具有 4 位有效数字.

5. 序列 $\{y_n\}$ 满足递推关系式

$$y_n = 10y_{n-1} - 1, \qquad n = 1, 2, \cdots.$$

若 $y_0 = \sqrt{2}$ 用近似值 $y_0^* = 1.41$ 代替，通过实际计算考察一下初始误差 $e_0^* = y_0 - y_0^*$ 在递推过程中的传播情况.

6. 在十进制 4 位浮点计算机上求解线性方程组：

$$\begin{aligned} 10^{-5}x_1 + x_2 &= 1, \\ 2x_1 + x_2 &= 2. \end{aligned}$$

按避免用绝对值很小的数作除数的法则，设计一个消元算法，使得计算结果具有较好的精确度 $(x_1 \doteq 0.5000, x_2 \doteq 0.1000 \times 10^1)$.