数值计算方法

吕同富　康兆敏　方秀男　编著

清华大学出版社

北　京

内 容 简 介

本书介绍了数值计算方法．内容涉及数值计算方法的数学基础、数值计算方法在工程、科学和数学问题中的应用以及所有数值方法的 MATLAB 程序等，涵盖了经典数值分析的全部内容．包括：非线性方程的数值解法；线性方程组的数值解法；矩阵特征值与特征向量的数值算法；插值方法；函数最佳逼近；数值积分；数值微分；常微分方程数值解法等．基于 MATLAB 是本书的特色，对书中所有的数值方法都给出了 MATLAB 程序，有大量详实的应用实例可供参考，有相当数量的习题可供练习．

本书取材新颖、阐述严谨、内容丰富、重点突出、推导详尽、思路清晰、深入浅出、富有启发性，便于教学与自学．

本书可作为理工科本科生、研究生“数值计算方法”课程的教材或参考书，也可作为科技人员使用数值计算方法和 MATLAB 的参考手册．

图书在版编目（CIP）数据

数值计算方法/吕同富，康兆敏，方秀男编著．—北京：清华大学出版社, 2008.9
ISBN 978–7–302–18238–2

Ⅰ. 数…　Ⅱ. ①吕… ②康… ③方…　Ⅲ. 数值计算–计算方法　Ⅳ. O241

中国版本图书馆 CIP 数据核字 (2008) 第 112188 号

责任编辑：佟丽霞　王海燕
封面设计：
责任校对：刘玉霞
责任印制：

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010–62770175　　**邮　　购**：010–62786544
投稿与读者服务：010–62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010–62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185×230　　**印　张**：20.5　　**字　数**：442 千字
版　　次：2008年9月第1版　　**印　次**：2008 年 9 月第 1 次印刷
印　　数：
定　　价：0.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010–62770177 转3103　　产品编号：

前　　言

数值计算方法与计算机相结合是本书的特点，也是科学计算发展的需要. 随着计算机的不断发展和进步，优秀的数学软件 MATLAB 应运而生，MATLAB 一问世就以它强大的功能，被广大科技工作者公认为科学计算最好的软件之一. 为使数值分析与 MATLAB 更好地结合，我们以最新版 MATLAB 为平台，编写了新版《数值计算方法》，这也是数值计算方法教材发展进步的必然结果.

本书介绍了数值计算方法. 内容涉及数值计算方法的数学基础、数值计算方法在工程、科学和数学问题中的应用以及所有数值方法的 MATLAB 程序等. 涵盖了经典数值分析的全部内容，包括非线性方程的数值解法；线性方程组的数值解法；矩阵特征值与特征向量的数值算法；插值方法；函数最佳逼近；数值积分；数值微分；常微分方程数值解法等. 重点讲述数值分析方法的思想和原理，尽可能避免过深的数学理论和过于繁杂的算法细节. 基于 MATLAB 是本书的特色. 数值计算方法与科学计算软件 MATLAB 相结合，有助于读者更有效地利用 MATLAB 的超强功能，来处理科学计算问题，有助于避免那种学过数值计算方法但不能上机解决实际问题的现象发生.

在编写过程中，参考了国内已出版的同类教材 (参考文献 [1~23])，吸收了其中许多精华，在题材的选取上作了一些变动，适当地增加了一些新内容，对书中所有的数值方法都给出了 MATLAB 程序，有大量详实的应用实例可供参考，有相当数量的习题可供练习.

本书取材新颖、阐述严谨、内容丰富、重点突出、推导详尽、思路清晰、深入浅出、富有启发性、便于教学与自学.

全书内容由吕同富教授主持编写. 方秀男编写第 1 章和第 2 章；康兆敏编写第 3 章；吕同富编写第 4 章至第 9 章. 吉林大学马富明教授、周蕴时教授，哈尔滨工业大学吴勃英教授，认真地阅读了本书，纠正了书中很多错误，并提出了许多保贵的修改意见. 这里向他们及本书所列参考文献的作者们，清华大学出版社的佟丽霞、王海燕，以及为本书出版给予热心支持和帮助的朋友们，表示衷心的感谢.

本书可作为理工科本科生、研究生“数值计算方法”课程的教材或参考书，也可作为科技人员使用数值计算方法和 MATLAB 的参考手册.

出好书，使千百万莘莘学子受益，一直是作者追求的目标. 但由于水平所限，尽管作了很大努力，可能还会有很多不妥甚至错误，望广大读者给予批评指正.

吕同富

2008 年 3 月

目　　录

第 1 章　序　　论

1.1　科学计算的一般过程

科学计算是人类从事科学研究和工程技术活动不可缺少的手段之一，在科学计算与计算机技术飞速发展的今天，为使计算机能更好地应用于科学研究和工程技术领域，必须按照下面的步骤进行：实际问题 → 数学模型 → 数值方法 → 程序设计 → 上机计算 → 分析结果.

1.1.1　对实际工程问题进行数学建模

应用有关学科的知识和数学理论，将实际工程问题，用精练准确的数学语言对其核心部分进行描述并给出数学模型，这一过程常称为数学建模. 一个好的数学模型需符合以下两方面要求：一是数学模型要能真实而准确地反映实际工程问题的本质；二是数学模型所用的数学算法能在计算机上实现，这两者缺一不可. 工程中的数学模型，按数学性质，可分为确定型与随机型；按表达形式，可分为连续型与离散型. 这些数学模型，有的能用确定的数学解析式描述，有的不能用确定的数学解析式描述，数值计算方法，主要讨论能用确定的数学解析式描述的实际工程计算问题.

1.1.2　对数学问题给出数值计算方法

计算机无论如何先进，它所能执行的计算也不过是简单的算术运算和逻辑运算，要想使计算机能够解决科学和工程计算问题，需把从科学和工程实际问题中建立的数学模型数值化，也就是根据不同的数学问题，寻求不同的数值计算方法. 数值计算方法只能用算术运算和逻辑运算，否则计算机将无法计算，这将直接关系到能否把计算机用于实际问题. 可见，数值计算方法在科学研究和工程技术计算中具有重要地位. 数值计算方法的优劣，显然速度和精度是两个重要的指标，一个好的数值计算方法不仅精度高而且速度快. 速度快，虽然就适当规模的问题而言，这一优势因计算机的能力而被削弱殆尽，但对于规模大的问题，速度仍是重要的因素，慢的数值计算方法由于不实用而被淘汰.

1.1.3　对数值计算方法进行程序设计

一个好的数值计算方法要通过程序设计才能在计算机上实现. 程序设计要求用最简练的计算机语言、最快的速度、最少的存储空间来实现某种要求的计算结果. 要达到这样的要求，程序设计者不仅要掌握数值计算方法，而且要熟悉并能熟练使用计算机语言，准确无误地描述每一个算法，并能以最快的速度发现和解决计算过程中出现的各种问题.

1.1.4 上机计算并分析结果

前面三个阶段工作的结果如何? 还需上机实验后才能得出结论. 上机计算的结果是否与工程实际相符合? 所作研究是否具有推广价值? 都是必须关注的问题. 若与工程实际不相符合, 则需找出原因, 回到前面三个阶段, 继续研究, 直到得出正确结论为止.

1.2 数值计算方法的研究内容与特点

1.2.1 数值计算方法的研究内容

科学技术发展到今天, 计算机的应用已渗透到社会生活的各个领域. 而数值计算方法是计算机处理实际问题的一种重要手段, 从宏观天体运动学到微观分子细胞学, 从工程系统到非工程系统, 无一能离开数值计算方法. 数值计算方法这门学科的诞生, 使科学发展产生了巨大飞跃, 它使各学科领域从定性分析阶段走向定量分析阶段, 从粗糙走向精密.

数值计算方法是数学的一个分支, 它以计算机为工具, 以数值代数 (线性方程组、矩阵特征值特征向量、非线性方程与方程组的数值解法), 数值逼近 (各种函数逼近问题数值解、数值积分、数值微分), 常微分方程数值解, 偏微分方程数值解, 最优化理论与方法等为研究内容.

1.2.2 数值计算方法的特点

先来看几个实例.

例 1.1 线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的行列式解法 —— Cramer 法则, 理论上可用来求解线性方程组. 用这种方法解一个 n 元线性方程组, 要计算 $n+1$ 个 n 阶行列式的值, 按 Laplace 展开法计算 n 阶行列式, 总共要计算

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

次乘法, 不计加法, 解一个 n 元线性方程组, 需计算 $n!(n+1)$ 次乘法. 当 n 充分大时, 计算量是很大的, 如一个 20 元的线性方程组, 大约要算 $21!\approx 5.1\times 10^{19}$ 次以上的乘法. 每秒可做百万次乘法的计算机, 每年可做 $365\times 24\times 3600\times 10^6\approx 3.15\times 10^{13}$ 次乘法, 所以, 在每秒可做上百万次乘法的计算机上, 用 Cramer 法则解 20 元的线性方程组, 所需的计算时间是 $(5.11\times 10^{19})\div(3.15\times 10^{13})=1.62\times 10^6\approx 162$ 万年, 这当然是没有实际意义的. 其实解线性方程组有很多方法, 如 Gauss 消去法, 一个 20 元的线性方程组乘法次数不超过 3000 次, 即使用一台微型计算机, 只需几秒钟就能完成计算. 这个例子说明研究数值计算方法很有必要, 因为数值计算方法所研究的正是在计算效率上最佳的或近似最佳的方法, 而不是像 Cramer 法则这样的方法.

例 1.2 计算积分 $I_n = \int_0^1 \frac{x^n}{x+5}\mathrm{d}x$. 通过直接计算可产生递推公式

$$I_n = -5I_{n-1} + \frac{1}{n}, \quad I_0 = \ln\frac{6}{5} \approx 0.182322, \tag{1.1}$$

且由经典积分知识可推得 I_n 具有如下性质：

① $I_n > 0$；

② I_n 单调递减；

③ $\lim\limits_{n\to\infty} I_n = 0$；

④ $\frac{1}{6n} < I_n < \frac{1}{5n}(n > 1)$.

下面用两种算法计算 I_n.

算法 A 递推关系 $I_n = -5I_{n-1} + \frac{1}{n}, I_0 = \ln\frac{6}{5} \approx 0.182322$.

MATLAB 程序

```
x=0.182322
for n=1:20
    n
    x=-5*x+1/n
end
```

自 $n = 1$ 计算到 $n = 20$ 产生如下计算结果 (见表 1.1).

表 1.1 算法 A 计算结果

n	I_n	n	I_n	n	I_n	n	I_n
1	0.0884	6	0.0344	11	−31.3925	16	9.8145e+4
2	0.5810	7	−0.0290	12	157.0457	17	−4.9073e+5
3	0.0431	8	0.2701	13	−785.1516	18	2.4536e+6
4	0.3470	9	−1.2393	14	3.9258e+3	19	−1.2268e+7
5	0.0265	10	0.2967	15	−1.9629e+4	20	6.1341e+7

由表 1.1 可见，该算法产生的数值解自 $n = 7$ 开始出现负值，且绝对值逐渐增加，这显然与 I_n 固有的性质相矛盾，因此本算法所得的数值解不符合问题的要求. 究其原因，在构造算法时未能充分考虑原积分模型的性态，即由公式 (1.1)，其计算从 I_{n-1} 到 I_n 每向前推进一步，其计算值的舍入误差 (见 1.3 节内容) 便增长 5 倍，误差由此积蓄传播导致最终数值解与原问题相悖的结果. 为了克服这一缺点改进算法 A 为算法 B.

算法 B 递推关系 $I_{n-1} = -\frac{1}{5}I_n + \frac{1}{5n}, I_{20} \approx \frac{\frac{1}{6\times 21} + \frac{1}{5\times 21}}{2} = 0.00873016$.

MATLAB 程序

```
x=0.00873016
```

```
for n=20:-1:1
    n-1
    x=-(1/5)*x+1/(5*n)
end
```

第二步用递推公式

$$I_{n-1} = -\frac{1}{5}I_n + \frac{1}{5n} \tag{1.2}$$

自 $n = 20$ 计算到 $n = 1$. 由于该算法每向后推一步，其舍入误差便减少 5 倍，因此获得符合原积分模型性态的数值结果 (见表 1.2).

表 1.2　算法 B 计算结果

n	I_n	n	I_n	n	I_n	n	I_n
19	0.0083	14	0.0112	9	0.0169	4	0.0343
18	0.0089	13	0.0120	8	0.0188	3	0.0431
17	0.0093	12	0.0130	7	0.0212	2	0.0580
16	0.0099	11	0.0141	6	0.0243	1	0.0884
15	0.0105	10	0.0154	5	0.0285	0	0.1823

对例 1.2 采用的是由原模型解的递推关系来实现计算机求解，这种方法称为**直接法**. 在大多数情况下，只能获得原模型的近似关系，即将连续系统离散化，这种解法称为**离散变量法**.

上面两个例子表明，数值计算方法与纯数学有明显不同，这种不同主要是由于数值计算方法是纯数学与科学工程实际和计算机相结合而形成的一门数学分支. 它既有纯数学高度抽象性、严密科学性的特点，又有应用的广泛性与实际实验的高度技术性的特点，是一门与计算机使用密切结合的实用性很强的学科. 一个好的数值计算方法，概括起来有以下特点：

第一，面向计算机数值计算方法理论的发展，与计算机技术的发展密切相关. 要根据计算机的特点，提供行之有效的数值计算方法. 算法只能包括算术运算和逻辑运算，这些运算是计算机能直接处理的运算. 在数值计算方法中，评价一种算法会随着计算机技术的发展而改变. 比如，人们普遍认为解线性方程组的超松弛迭代法 (SOR) 优于 Jacobi 迭代，这是因为 SOR 法有更高的收敛速度. 但在并行计算发展起来之后，人们发现 Jacobi 迭代法有很好的并行性，而 SOR 法却不具备可并行性，从而在使用并行计算机解大规模线性方程组时，经典的 SOR 法不及 Jacobi 法优越.

第二，数值计算方法的理论分析. 所设计的数值计算方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，这些都是建立在相应的数学理论基础之上.

第三，一个好的数值算法不仅要节省运算时间，而且要节省计算机的存储空间. 这是建立数值算法所要研究的内容，也是数值算法在计算机上实现必须满足的条件. 也可以称为数

值计算方法的可行性.

有时，所研究的问题，数学上有明确的求解方法，像前面提到的用 Cramer 法则解线性方程组的例子，Cramer 法则不仅给出了解的存在性，而且给出了求解的数学公式，似乎只要在计算机上实现这个公式即可得到方程组的解，但实际上，即使一个规模不大的线性方程组，用 Cramer 法则求解，其计算量仍然是大得惊人，使用一般的计算机在人们可以接受的时间内，几乎不可能得到方程组的解. 因此，如何使用合理的计算量，求解一个线性方程组的解，就成了数值计算方法的一个重要课题. 除此之外还有方法的稳定性问题. 有些数学方法对于计算过程中误差太敏感，使其无法在计算机上实现. 这些问题的研究是数值计算方法理论区别一般数学理论的重要特征.

第四，实际验证数值计算方法. 任何一个好的数值计算方法除了理论上要满足上述三点之外，还要通过数值计算实际验证，看是否行之有效.

根据数值计算方法的上述特点，在学习数值计算方法课程时，首先要掌握构造方法的原理、思想，注意算法的技巧与计算机的实现相结合，也要注重数值计算方法基础和数学理论的学习，其次要重视实践，通过实例和动手计算，学会怎样使用数值计算方法在计算机上解决各类科学工程技术计算问题，避免那种学过数值计算方法，但不能上机解决实际问题的现象发生. 为了掌握本课程的内容，需要做一定量的习题和上机计算题.

另外，由于本课程内容涉及微积分、线性代数、常微分方程、偏微分方程、泛函分析等内容，因此需读者了解这几门课程的基本内容.

1.3 计算过程中的误差及其控制

1.3.1 误差的来源与分类

从科学研究和实际工程技术问题计算的全过程看，误差的来源主要有四个方面.

(1) 模型误差

对实际工程技术问题建立数学模型时，总是在一定条件下抓主要因素，忽略次要因素，这样得到的数学模型是理想化的数学模型，它包含了对实际问题进行近似的数学描述时所引起的误差，这种误差称为**模型误差**.

(2) 观测误差

在数学模型或计算公式中包含着一些已知数据 (称为原始数据)，这些数据往往是由观测实验得到，它们和实际的数据大小之间有误差，这种误差称为**观测误差**.

(3) 截断误差

许多数学运算理论上的精确值往往要用无限的过程才能求出，如微分、积分、无穷级数求和等都是通过无限的极限过程来定义的. 然而, 实际问题的计算在计算机上只能用有限次的算术运算和逻辑运算来完成，因此需要将问题的解决方案加工成算术运算和逻辑运算的

有限序列，这种加工常常表现为无穷过程的截断，由此产生的误差称为**截断误差**.

例 1.3 计算函数 e^x 在某点的值时，由于 e^x 的幂级数展开式为

$$\mathrm{e}^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots, \tag{1.3}$$

但是用计算机求解时，不能直接得出无穷项的和，只能截取有限项，求出

$$S_n(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}, \tag{1.4}$$

用 $S_n(x)$ 作为 e^x 的值必然会有误差，根据 taylor 展开式得其截断误差为

$$\mathrm{e}^x - S_n(x) = \frac{x^{n+1}}{n+1!}\mathrm{e}^{\theta x}, \quad 0 < \theta < 1. \tag{1.5}$$

(4) 舍入误差

由于计算机数系是离散的有限集，计算机在接收和运算数据时，总是将位数较多的数舍入成一定位数的机器数，这样产生的误差称为**舍入误差**. 每一步的舍入误差是微不足道的，但是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，如例 1.2 就是这种情况.

模型误差和观测误差也称**固有误差**，一般来讲不是计算工作者所能独立解决的; 截断误差和舍入误差也称**计算误差**，是数值计算方法要讨论的内容.

1.3.2 误差与有效数字

在数值计算中，误差是不可避免的，但人们总希望计算结果能足够准确，这就需要对误差进行估计，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概念.

定义 1.1 设 x^* 是某量的精确值，x 是 x^* 的近似值，则称差 $e = x^* - x$ 为近似值 x 的**绝对误差**，简称**误差**.

由于精确值 x^* 在实际中未知，因而误差 e 通常是无法确定的，人们只能通过测量工具或计算过程，设法估计出它们的取值范围，即误差绝对值的一个上界.

定义 1.2 设存在一个 $\varepsilon > 0$ 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**.

若近似值 x 的误差限为 ε ，则 $x - \varepsilon \leqslant x^* \leqslant x + \varepsilon$，这表明 x^* 落在 $[x - \varepsilon, x + \varepsilon]$ 上，在实际应用中常采用 $x = x^* \pm \varepsilon$ 的写法，来表示 x 近似的精度或精确值 x^* 所在的范围. 例如，$x = 15 \pm 2, y = 1000 \pm 5$.

绝对误差限的大小不能完全刻画近似值的精确程度. 例如，某量的精确值为 $x^* = 1000$，其近似值为 $x_1 = 999$，另一个量的精确值为 $x^* = 10$，相应的近似值为 $x_2 = 9$，这两个量的

绝对误差限都是 $\varepsilon = 1$，显然 x_1 的精度比 x_2 的精度好，为反映这种近似程度，引入相对误差的概念.

定义 1.3 称 $e_r = \dfrac{e}{x^*} = \dfrac{x^* - x}{x^*}$ 为近似值 x 的**相对误差**.

相对误差是一个无量纲量，通常用百分数表示，相对误差的绝对值越小，近似值的精度越高. 例如，前面两个量 x_1 和 x_2，它们的相对误差限分别为 $e_r(x_1) = 0.1\%$ 和 $e_r(x_2) = 10\%$，所以近似值 x_1 的精度比 x_2 的精度好. 同样由于精确值 x^* 通常是未知的，一般不能给出 e_r 的精确值，只能估计它的大小范围.

定义 1.4 相对误差可正可负，它的绝对值上界叫做**相对误差限**，记作 ε_r，即

$$|e_r| = \left|\frac{x^* - x}{x^*}\right| = \left|\frac{e}{x^*}\right| \leqslant \varepsilon_r. \tag{1.7}$$

相对误差限 ε_r 不如绝对误差限 ε 容易得到，在实际计算应用中常用式 $\varepsilon_r = \left|\dfrac{\varepsilon}{x}\right|$ 计算相对误差限.

为了给出一种近似数的表示方法，使之既能表示其大小又能表示其精确程度，引进有效数字的概念. 例如，设

$$x^* = \pi = 3.1415926\cdots,$$

经四舍五入取其三位近似值得 $\pi \approx 3.14, \pi \pm 0.002$；若取其五位近似值得 $\pi \approx 3.1416, \pi \pm 0.000008$. 它们的绝对误差限都不超过末位数字的半个单位，即

$$|\pi - 3.14| \leqslant \frac{1}{2} \times 10^{-2}, \quad |\pi - 3.1416| \leqslant \frac{1}{2} \times 10^{-4},$$

称它们精确到了末位.

定义 1.5 设精确值 x^* 的近似值 $x = \pm 0.a_1a_2\cdots a_n \times 10^m$，其中 $a_1 \neq 0$, 诸 $a_i \in \{0, 1, 2, \cdots, 9\}, m$ 为整数，如果

$$|e| = |x^* - x| < \frac{1}{2} \times 10^{m-n}, \tag{1.8}$$

则称近似值 x 具有 n 位有效数字或称 x 精确到 10^{m-n} 位，其中，$a_1, a_2, \cdots, a_n$ 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 知 $\pi = 3.14$ 和 $\pi = 3.1416$ 分别有 3 位和 5 位有效数字. 由式 (1.8) 知，有效数字越多，绝对误差越小，至于有效数字与相对误差的关系有如下结论.

定理 1.1 设近似值 $x = \pm 0.a_1a_2\cdots a_n \times 10^m$ (其中 $a_1 \neq 0$)，有 n 位有效数字，则 x 的相对误差限为 $\varepsilon_r \leqslant \dfrac{1}{2a_1} \times 10^{-n+1}$.

证明 由 x 有 n 位有效数字知 $|e| = |x^* - x| < \dfrac{1}{2} \times 10^{m-n}$，而 $|x| > a_1 \times 10^{m-1}$，故有

$$|e_r| = \left|\frac{x^* - x}{x}\right| \leqslant \frac{\frac{1}{2} \times 10^{m-n}}{a_1 \times 10^{m-1}} = \frac{1}{2a_1} \times 10^{-n+1} = \varepsilon_r,$$

即相对误差限为 $\varepsilon_r = \dfrac{1}{2a_1} \times 10^{-n+1}$. 证毕.

定理 1.2　设近似值 $x = \pm 0.a_1a_2\cdots a_n \times 10^m$(其中 $a_1 \neq 0$), 相对误差限为 $\varepsilon_r = \dfrac{1}{2(a_1+1)} \times 10^{-n+1}$, 则 x 至少有 n 位有效数字.

证明　由于 $\varepsilon = |x|\varepsilon_r$, 而 $|x| \leqslant (a_1+1) \times 10^{m-1}$, 所以,

$$\varepsilon \leqslant (a_1+1) \times 10^{m-1} \times \frac{1}{2(a_1+1)} \times 10^{-n+1} = \frac{1}{2} \times 10^{m-n}.$$

因此, x 至少有 n 位有效数字. 证毕.

例 1.4　设 $x = 2.72$ 表示 e 具有 3 位有效数字的近似值, 求此近似值的相对误差限.

解　$x = 2.72 = 0.272 \times 10^1, a_1 = 2, n = 3$, 由定理 1.1 有

$$\begin{aligned}\varepsilon_r &\leqslant \frac{1}{2a_1} \times 10^{-(n-1)} \\ &= \frac{1}{2 \times 2} \times 10^{-(3-1)} = 0.25 \times 10^{-2}.\end{aligned}$$

例 1.5　要使 $\sqrt{20}$ 的近似值的相对误差限小于 0.1%, 要取几位有效数字?

解　因 $4 < \sqrt{20} < 5$, 故可在定义 1.5 中取 $a_1 = 4$. 若相对误差限满足 $\varepsilon_r < 0.001$, 由定理 1.2 有

$$\varepsilon_r \leqslant \frac{1}{2(a_1+1)} \times 10^{-n+1},$$

可见 n 位有效数字应满足

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

由此解出 $n = 4$, 即应取 4 位有效数字.

1.3.3　误差的传播

在科学研究和工程计算中每步都可能产生误差, 而一个问题的解决往往要经过成千上万次的运算, 不可能每一步都加以分析. 只能通过对误差的某些传播规律进行分析, 指出在数值计算中应遵循的几条原则, 这将有助于鉴别计算结果的可靠性并防止误差危害现象的产生.

(1) 误差分析的重要性

在例 1.2 中, 算法 A 用精确的计算公式却产生了一个错误的结果, 分析原因是因为: 初值 I_0 有误差 $e(I_0)$, 由此引起以后各步计算的误差 $e(I_n)$, 满足关系

$$e(I_n) = -5e(I_{n-1}), \quad n = 1, 2, \cdots,$$

从而有

$$e(I_n) = (-5)^n e(I_0), \quad n = 1, 2, \cdots,$$

这说明 I_0 有误差 $e(I_0)$，则 $e(I_n)$ 就有误差 $e(I_0)$ 的 $(-5)^n$ 倍.

而例 1.2 算法 B 是将递推公式倒过来使用，由式 (1.2) 得 $e(I_{n-1}) = -\frac{1}{5}e(I_n)(n = 20, 19, \cdots, 1)$, 尽管初值 $I_{20} = 0.0087$ 误差 $e(I_{20})$ 很大，但因为误差传播逐渐缩小，I_n 的误差为 $e(I_n)$，则 I_0 的误差是 $e(I_{20})$ 的 $\left(\frac{1}{5}\right)^{20}$ 倍. 也就是每计算一步，误差就会缩小前一步的 $-\frac{1}{5}$，故计算结果可靠. 此例说明，在数值计算中不注意误差分析，用了类似于例 1.2 算法 A 的公式，就会出现“差之毫厘，谬之千里”的错误结果.

(2) 误差的传播

计算机的数值运算主要是加、减、乘、除四则运算，带有误差的数据经过四则运算后误差怎样变化，用微分可以描述. 由于精确值与近似值通常很接近，其差可以认为是较小的增量，即可以把误差看作微分，由此可得误差的微分近似关系：

$$e = x^* - x = \mathrm{d}x,$$

$$e_r = \frac{e}{x^*} = \frac{\mathrm{d}x}{x} = \mathrm{d}\ln x,$$

即 x 的微分表示 x 的绝对误差，$\ln x$ 的微分表示它的相对误差. 利用这两个关系式及微分运算可以得到一系列有关四则运算的误差结果，例如：

由 $\mathrm{d}(x \pm y) = \mathrm{d}x \pm \mathrm{d}y$ 可得两数之和 (差) 的误差等于两数的误差之和 (差)；

由 $\mathrm{d}(\ln xy) = \mathrm{d}\ln x + \mathrm{d}\ln y$ 可得两数之积的相对误差等于两数相对误差之和；

由 $\mathrm{d}\left(\ln\frac{x}{y}\right) = \mathrm{d}\ln x - \mathrm{d}\ln y$ 可得两数商的相对误差等于两数相对误差之差.

一般地，设变量 u 由变量 $x_1, x_2, \cdots, x_n$ 经某种运算得到，可设 $u = f(x_1, x_2, \cdots, x_n)$, 则绝对误差为

$$\mathrm{d}u = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}\mathrm{d}x.$$

要得到更准确的误差估计，一般可利用函数的 Taylor 展开式进行估计.

1.3.4 误差的控制

(1) 简化计算步骤，减少运算次数

同一个计算问题，如果能减少运算次数不但可节省计算时间，提高计算速度，而且还能减少误差的积累.

计算 x^{255} 的值，如果逐个乘要做 254 次乘法，但若写成

$$x^{255} = x \cdot x^2 \cdot x^4 \cdot x^8 \cdot x^{16} \cdot x^{32} \cdot x^{64} \cdot x^{128},$$

只要做 14 次乘法运算即可. 又如计算多项式

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

的值，若直接计算 a_kx^k 再逐次相加，一共要做

$$n+(n-1)+\cdots+1=\frac{1}{2}n(n+1)$$

次乘法和 n 次加法，若采用秦九韶算法

$$\begin{cases} S_n=a_n, \\ S_k=xS_{k+1}+a_k, \quad k=n-1,n-2,\cdots,1,0, \\ P_n(x)=S_0, \end{cases}$$

只要 n 次乘法和 n 次加法即可算出 $P_n(x)$ 的值.

(2) 避免两相近数相减

在数值计算中两相近数相减有效数字会严重损失.

例如，$x=618.45$ 和 $y=618.32$ 都是 5 位有效数字，但 $x-y=0.13$ 只有两位有效数字，所以最好改变计算方法，避免这类运算的发生.

如当 x_1 和 x_2 较接近时，则

$$\ln x_1-\ln x_2=\ln\frac{x_1}{x_2},$$

右端算式有效数字不损失. 当 x 很大时，按

$$\sqrt{x+a}-\sqrt{x}=\frac{a}{\sqrt{x+a}+\sqrt{x}}$$

计算结果较好. 当计算 $f(x)-f(x_0)$ 的近似值时，可用 Taylor 展开式

$$f(x)-f(x_0)=f'(x_0)(x-x_0)+\frac{1}{2}f''(x_0)(x-x_0)^2+\cdots$$

取右端有限项近似左端. 如果无法改变算式，则采用增加有效位数进行运算.

(3) 防止大数吃掉小数

在数值运算中有时数量级相差很大，而计算机字长有限，如不注意运算次序就有可能出现大数吃掉小数的现象，影响计算结果的可靠性. 例如，在 8 位十进制计算机上计算 $x=54272401+0.6$，由于在计算机内计算时，要写成浮点形式，且要先对阶，对阶时 $x=54272401=0.54272401\times10^8$，$0.6=0.000000006\times10^8$ 在 8 位机上表示 0，因此

$$\begin{aligned} x=&54272401+0.6=0.54272401\times10^8+0.00000000\times10^8 \\ =&0.54272401\times10^8=54272401. \end{aligned}$$

(4) 绝对值太小的数不宜作除数

绝对值很小的数作除数也会影响数值计算结果的精度，由

$$\mathrm{d}\left(\frac{x}{y}\right)=\frac{y\mathrm{d}x-x\mathrm{d}y}{y^2}$$