数值计算基础

严克明 欧志英 刘树群 编著

甘肃人民出版社

数值计算基础

严克明 欧志英 刘树群 编著

甘肃人民出版社

图书在版编目（CIP）数据

数值计算基础/严克明，欧志英，刘树群编著—兰州：甘肃人民出版社，2006.3
ISBN 7-226-03392-5

Ⅰ.数... Ⅱ①严...②欧...③刘... Ⅲ.数值计算 Ⅳ.0241

中国版本图书馆 CIP 数据核字（2006）第 011091 号

责任编辑：高文波
封面设计：王小蕾

数值计算基础

严克明　欧志英　刘树群 编

甘肃人民出版社出版发行

（730000 兰州市南滨河东路 520 号）

西北师范大学印刷厂印刷

开本 787×1092 毫米 1/16 印张 13.75 插页 1 字数 350 千

2006 年 2 月第 1 版　2006 年 2 月第 1 次印刷

印数 1—3,000

ISBN 7-226-03392-5　定价：22.00 元

前　言

继实验方法、理论方法之后，科学计算已成为人们进行科学活动的第三种方法。因此，适用于计算机的数值计算方法课程已经是理工科大学生一门应用性很强的必修基础课..目前，国内外有关数值计算的教材很多，变化也很大，除了内容的变化和发展外，也出现了一些适应各种对象和教学时数不同类型的教材。对于少学时用的教材，许多担任该课程的教师，希望有一本不只是简单构造、描述算法，而且有一些必要的基础理论，便于教学和学生能进一步通过自学对算法概念及理论有更深了解，同时又便于数值方法在计算机上实现的教材，本书就是为此而编写的。

数值计算是一门基础课，它有严密的科学系统，但它又是一门应用性很强的课程，希望使学生能够用本学科的基础理论和基本方法，初步掌握在计算机上进行有关的科学与工程计算。本书在编写过程中，力求注意将一些比较新的、成熟的、重要的数值计算方法写进去；在基础理论方面，既考虑到它的系统性、严密性，又以够用即可为原则，取材繁简适度。本教材十分重视数值方法在计算机上的实现，书中所涉及的算法均采用 C 语言进行描述，以便能对算法中出现的各种问题有比较清晰而准确地把握，也便于直接在计算机上使用。

为了充分利用信息技术和网络资源，我们编写了大量的算法程序（网址为：http://www.lut.cn/kecheng/math1/shiyan/jsff.htm）供读者下载使用，为教学提供方便。也希望给学生在课后复习、训练带来方便，使得通过反复练习加深对算法的理解。

编者

2005 年 10 月

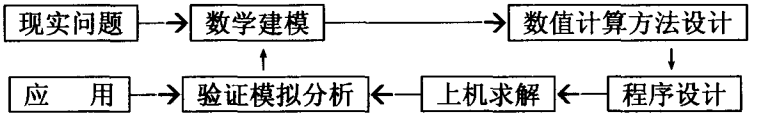
目　　录

第一章　数值计算方法的基本概念

§1 数值方法的对象和特点

计算机是二十世纪对科学、工程技术和人类社会生活影响最深刻的高新技术之一，而且在新世纪中必将发挥更重要的作用。它对科学技术的影响，莫过于使得科学计算与科学理论、实验研究并列，成为人类探索未知科学领域与工程设计的现代科学方法的第三种方法和手段。具有一定的科学计算能力和应用计算机解决工程、社会经济中的实际问题的能力是新世纪人才的基本素质。科学计算并不是计算机本身的自然产物，而是数学与计算机有机结合的结果，它的核心内容是以数学模型为基础，以计算机及数学软件为工具进行模拟研究。

为了能具体地说明数值计算的研究对象和特点，可考察科学计算解决实际问题的过程：



根据建立的数学模型，提出数值计算方法、编程并上机算出结果，这一过程是数值计算的任务，也是数值分析研究的对象。数值计算方法（计算方法、计算数学）是数学的一个分枝。从有数学的时候就有了数值计算方法，算盘、手摇计算器都是计算工具，现代则是电子计算机。数值计算方法研究的主要内容和特点：

1. 设计根据计算机特点，使计算机能顺利执行的可行和有效的算法。算法一般包括加、减、乘、除算术运算和逻辑运算，常用数学函数，条件和循环控制等。

2. 建立可靠的分析理论，研究所构造的算法的收敛性、数值稳定性，并对误差进行分析。我们知道许多数学问题的解，要经过无限次算术运算才能计算出结果来。而我们在计算上只能进行有限次运算，所以必须把无限次运算过程截断成有限次运算求解。另外，有些数学问题的解，尽管可以经过有限次运算计算出结果，但计算机只能使用有限位数字表示进行的运算，其余的数字必须取舍，从而产生舍入误差。有时还需把连续变量问题（如微分方程初值问题）转化为离散变量的问题（差分方程）求解。截断、舍入、离散化实质都是近似代替，因而要分析算法是否收敛，计算出的解是否可靠，误差有多大等？

3. 算法的有效性。对于一个可用计算机求解的问题可能不只一个算法。这里的问题是哪个算法运算量少？占用内存也少？这就是算法的有效性问题。

4. 数值实验。一个算法既要求理论上严密，还要通过数值实验来验证解是否行之有效。

总之，数值计算方法突出的特点是面向计算机，特别强调算法的构造、收敛性、数值的稳定性，进行误差估计和数值实验。

下面通过一个例子来说明学习数值计算方法的重要性。

例 1.1 对方程组

$$\begin{cases} x_1+\dfrac{1}{2}x_2+\dfrac{1}{3}x_3=\dfrac{11}{6} \\ \dfrac{1}{2}x_1+\dfrac{1}{3}x_2+\dfrac{1}{4}x_3=\dfrac{13}{12} \\ \dfrac{1}{3}x_1+\dfrac{1}{4}x_2+\dfrac{1}{5}x_3=\dfrac{47}{60} \end{cases}$$

求解时，先将系数舍入成两位有效数字的数，变为

$$\begin{cases} x_1+0.5x_2+0.33x_3=1.8 \\ 0.50x_1+0.33x_2+0.25x_3=1.1 \\ 0.33x_1+0.25x_2+0.20x_3=0.78 \end{cases}$$

用列主元消元法，将后两个方程中消去 x_1，得

$$\begin{cases} 0.08x_2+0.08x_3=0.20 \\ 0.08x_2+0.09x_3=0.19 \end{cases}$$

消去 x_2，则

$$x_3=-1.0$$

回代，于是

$$x_2=3.5 \quad , \quad x_1=0.38$$

与原方程精确解 $x_1=x_2=x_3=1$ 相比较，计算出的近似解完全不可靠。这是由于参数的舍入产生的微小变化引起解的剧烈变化，也就是所谓的病态问题。

有了计算机，也不是给出一个算法它就能有效地求解。例如，求解一个上述的 n 阶线性方程组，可用 *Cramer* 法则求解，它需要 $(n+1)!+n$ 次的乘除法，很容易计算出来，使用每秒进行十亿次乘除运算的计算机，当 $n=30$ 时大约需要二十六亿亿年！这显然是不可能的。若用列主元消元法需要 $\dfrac{n^3}{3}+n^2-\dfrac{n}{3}$ 次乘除法，则只需要不到 1 秒的时间。

从上面的例子可以看到研究算法的重要性。数值计算方法它既有严密与高度抽象性的特点，又有应用的广泛性的特点，是一门与计算机技术密切结合的实用性很强的课程。学习时我们首先要注意掌握方法的基本原理和思想，同时通过实例，学习针对具体问题筛选、构造和比较算法、编写程序、上机调试计算、分析解释计算结果，努力提高应用计算机解决实际问题的能力。本书以讲授算法构造的基本思想和应用为主，算法的分析则只作一些简略的介绍。

§2 误差来源与误差的基本概念

2.1 误差的来源及分类

在数值计算中，误差是不可避免的。引起误差的因素很多，主要的原因有以下几种：

1. **模型误差** 用数学方法解决实际问题，首先要建立数学模型，即将实际问题经过抽象合理简化，略去一些次要因素。因而它只是对所提出的问题的一种近似描述，包含有误差，这种误差称为模型误差。

2. **观测误差**　在数学模型中总含有一些来源于实验的参数，如温度、长度、电压等，它们的值往往是由观测得到的，它们不依赖于数的表示方法，而因观察仪器精度的限制，以及观测者能力的差别，必然会产生误差。这类称之为观测误差。

3. **截断误差**　数学问题常常难以求解或数学上难以表达，因此要简化为较易求解的问题或近似表达问题的解，由此引起的解的误差称为截断误差（或方法误差）。

如求一个收敛的无穷级数之和，这是一个无限的计算过程，通常总是用它的部分和作为近似值，例如

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + \frac{(-1)^n}{(2n)!}x^{2n} + \cdots$$

用台劳（*Taylor*）多项式

$$P_{2n}(x) = 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + \frac{(-1)^n}{(2n)!}x^{2n}$$

近似代替，则数值方法的截断误差是

$$R_{2n}(x) = \cos x - P_{2n}(x) = \frac{(-1)^{n+1}\cos\theta x}{(2n+2)!}x^{2n+2} \qquad (0 < \theta < 1)$$

在数值计算中，我们常常使用一种近似的方法求一个问题的解。例如，求曲边梯形（图 1.1）的面积

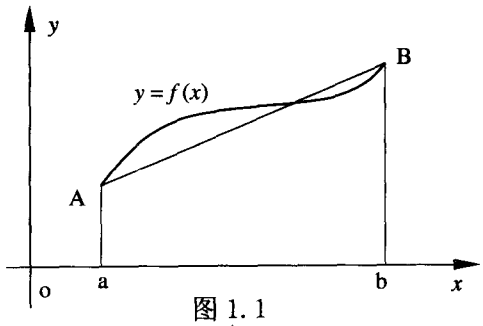


图 1.1

$$S = \int_a^b f(x)dx$$

若用梯形的面积

$$A = \frac{b-a}{2}[f(a) + f(b)]$$

作为它的近似值，所产生的误差 $S - A$ 也称为截断误差或方法误差，有时也将这种把连续型问题离散化而产生的误差称为离散误差。

4. **舍入误差**　在计算过程中往往要对数字进行舍入。如受计算机字长的研制，无穷小数和位数很多的数必须舍入成一定的位数的数。例如，用 3.14159 近似代替 π 所产出的误差，计算机的浮点运算使数值解形成的误差等。这类误差统称为舍入误差。舍入误差，单从一次计算来看对结果的影响不会很大，但在大量的计算中，误差的叠加及误差传播是逐步扩大的，因而对计算结果会产生较大的影响。此外在表示实数时，由于计算机内部多采用二进制形式的浮点表示，而使一些看起来规整的数据而在计算机内部只能近似表示。如短实数零

$$0 = 00000000\ 00000000\ 00000000\ 00000000\ b = \pm 2^{-127} \approx \pm 10^{-38}$$

算不上精确表示，只不过内部能把这个数始终按 0 对待而已。

本书主要讨论算法中的截断误差与舍入误差，对舍入误差通常只做一些定性的分析。

2.2　绝对误差和相对误差

表示误差大小的方法很多，较常用的有两种方法。

假设一个量的准确值为 x，其近似值为 x^*，称

$$e = x - x^* \tag{2.1}$$

为近似值 x^* 的绝对误差（记为 $e(x^*)$），简称误差。

一般情况下，我们只能知道近似值 x^*，而很难得到准确值 x，但可以根据测量工具或计算的情况，对绝对误差的大小范围作出估计，即可以给出一个正数 ε，使得

$$|e| = |x - x^*| \le \varepsilon \tag{2.2}$$

我们称 ε 为近似值 x^* 的一个绝对误差限。显然，绝对误差限不是唯一的。有了误差限及近似值，就可以给出准确值的范围

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

这个不等式也常记作

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

容易看出，经过的四舍五入得到的数，其误差必定不超过被保留的最后数位上的半个单位。例如，取 $\pi = 3.14$，则

$$|\pi - 3.14| \le \frac{1}{2} \times 10^{-2} = 0.005$$

有时，绝对误差不足以刻画近似值的精确程度，例如测量百米轨道和黑板长度时的两个量：$x = 10000 \pm 10$，$y = 200 \pm 1$（单位为 cm），从表面上看 $\varepsilon_1 = 10\varepsilon_2$，但前者每厘米长度最多只产生 0.001 厘米的误差，而后者则可能产生 0.005 厘米的误差。因此，要决定一个近似值的精确程度，除了要看绝对误差的大小，还必须考虑量本身的大小。我们定义

$$e_r = \frac{x - x^*}{x} \tag{2.3}$$

为 x^* 的相对误差. 因为计算过程中准确值 x 往往不知道, 所以常常将 x^* 的相对误差 e_r 定义为

$$e_r \approx \frac{x - x^*}{x^*} \tag{2.4}$$

同样的原因，我们无法准确计算出相对误差，然而可象绝对误差一样，估计出它的大小范围，即给出一个正数 δ_r，使得

$$|e_r| \approx \left|\frac{e}{x^*}\right| \le \delta_r \tag{2.5}$$

则称 δ_r 为 x^* 的一个相对误差限。

可以证明，当 $|e_r|$ 很小时 $\frac{e}{x} - \frac{e}{x^*}$ 是 e_r 的高阶无穷小，所以取 $\frac{e}{x^*}$ 为相对误差是合理的。x^* 与绝对误差、绝对误差限有相同的量纲，而 x^* 的相对误差、相对误差限是无量纲的，常用百分数来表示。在实践常取 ε 中“最小者”作为 $e(x^*)$。

2.3 有效数字

当准确值 x 有多位或无穷多位数时，常常用四舍五入的原则得到 x 的前几位的近似值。表示一个数的近似值时，常用到有效数字的概念，它既能表示近似值的大小，又能表示其精确程

度。一般，一个实数的表示是无限的，例如

$$\pi = 3014159265\cdots$$
$$e = 2071828182\cdots$$
$$\frac{1}{3} = 0.33333333\cdots$$
$$\sqrt{2} = 1.41421356\cdots$$

按四舍五入取四位小数，可得 $\sqrt{2}^* = 1.4142$，前面已经提到，该数的绝对误差不超过末位数字的半个单位，即

$$\left|\sqrt{2} - 1.4142\right| \le \frac{1}{2}\times 10^{-4} = 0.00005$$

定义 2.1 设 x 的近似值

$$x^* = \pm 0.a_1a_2\cdots a_n \times 10^m \quad (a_1 \neq 0) \tag{2.6}$$

如果

$$\left|x - x^*\right| \le \frac{1}{2}\times 10^{m-n} \tag{2.7}$$

则称近似值 x^* 有 n 位有效数字。

例 2.1 π 的近似值 $\pi^* = 3.1416$ 是具有 5 位有效数字近似值。

在实际中，式(2.6)又常以等价的形式出现

$$x^* = a_1a_2\cdots a_m.a_{m+1}\cdots a_n \tag{2.8}$$

其中 $a_i(i = 1,2,3,\cdots,n)$ 均为 0,1,2,$\cdots$,9 中的一个数字，$m \neq 0$ 时，$a_1 \neq 0$。若 x^* 的绝对误差满足(2.7)式，且 a_s 是 x^* 自左向右第一位非零数字，则自 a_s 起到最右边的数字为止，所有的数字都叫 x^* 的有效数字。

例 2.2 按四舍五入的原则，写出下列各数具有 5 位有效数字的近似值：187.9325，0.00654216，8.000023，5.000024×10^3。

解 根据定义和对（2.8）式的分析，以上各数具有 5 位有效数字的近似值分别是

187.93，　0.0065422，　8.0000，　5000.0

例 2.3 30.4 与 30.40 是有区别的，前者只有 3 位有效数字，而后者有 4 位有效数字。

关于有效数字和相对误差的关系，我们有下面的定理。

定理 1.1 若形如式（2.6）的近似值 x^* 具有 n 位有效数字，则其相对误差有估计式

$$\left|\frac{x - x^*}{x^*}\right| \le \frac{1}{2a_1}\times 10^{1-n} \tag{2.9}$$

反之，若 x^* 的相对误差满足

$$\left|\frac{x - x^*}{x^*}\right| \le \frac{1}{2(a_1+1)}\times 10^{1-n} \tag{2.10}$$

则 x^* 至少有 n 位有效数字。

证明 由式（2.6）及式（2.7）可得

$$a_1\times 10^{m-1} \le \left|x^*\right| \le (a_1+1)\times 10^{m-1} \tag{2.11}$$

$$|x-x^*|\le\frac{1}{2}\times10^{m-n}$$

所以，

$$\left|\frac{x-x^*}{x^*}\right|\le\frac{\frac{1}{2}\times10^{m-n}}{a_1\times10^{m-1}}=\frac{1}{2a_1}\times10^{1-n}$$

反之，由已知式（2.10）成立，并根据式（2.11），于是

$$|x-x^*|\le\frac{|x^*|}{2(a_1+1)}\times10^{1-n}\le\frac{(a_1+1)\times10^{m-1}}{2(a_1+1)}\times10^{1-n}=\frac{1}{2}\times10^{m-n}$$

则 x^* 至少有 n 位有效数字。

该定理表明，一个近似值有效数字愈多，其相对误差愈小。根据定理，例 2.3 中 30.40 的相对误差限为 0.017%，而 30.4 只能得到 0.17%的相对误差限。

例 2.4 要使 $\sqrt{20}$ 的近似值的相对误差小于 0.1%，问至少要取几位有效数字？

解 由于 $\sqrt{20}=4.4721\cdots$，所以 $a_1=4$。由定理可知，要使

$$\left|\frac{x-x^*}{x^*}\right|\le\frac{1}{2a_1}\times10^{1-n}=\frac{1}{8}\times10^{1-n}<0.1\%$$

取 $n=4$，即 $\sqrt{20}\approx4.472$ 就能满足要求。

§3 误差在数据计算中的传播

3.1 基本运算中的误差估计

本节讨论的基本运算主要是指四则运算与一些常用函数的计算。

在数值计算中，参与运算的数据往往都是近似值，带有误差，现在我们分析一下原始数据误差在基本运算时的传播规律。对于一组参量 $x_1,x_2,\cdots,x_n$，经运算 $y=f(x_1,x_2,\cdots,x_n)$，则数值运算的误差传播可用高等数学中的全微分进行分析。假设 $f(x_1,x_2,\cdots,x_n)$ 可微，$x_1^*,x_2^*,\cdots,x_n^*$ 分别是初始数据 $x_1,x_2,\cdots,x_n$ 的近似值，即

$$x_1=x_1^*+e(x_1^*)，\quad x_2=x_2^*+e(x_2^*)，\quad\cdots，\quad x_n=x_n^*+e(x_n^*)$$

其中，$e(x_1^*)$，$e(x_2^*)$，$\cdots$，$e(x_n^*)$ 分别是 $x_1^*,x_2^*,\cdots,x_n^*$ 的绝对误差。我们考察用 $x_1^*,x_2^*,\cdots,x_n^*$ 分别代替 $x_1,x_2,\cdots,x_n$ 时计算函数值时产生的误差，即 $y^*=f(x_1^*,x_2^*,\cdots,x_n^*)$ 的误差。假定 $|e(x_1^*)|$，$|e(x_2^*)|$，$\cdots$，$|e(x_n^*)|$ 都很小，则 y^* 的误差

$$e(y^*)=y-y^*=f(x_1,x_2)-f(x_1^*,x_2^*)$$

可近似地表示成

$$\begin{aligned}e(y^*)&\approx df(x_1^*,x_2^*,\cdots,x_n^*)\\&=\sum_{i=1}^{n}\frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_i}\cdot(x_i-x_i^*)\quad=\sum_{i=1}^{n}\frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_i}\cdot e(x_i^*)\end{aligned}\tag{3.1}$$

而且，其相对误差

$$
\begin{aligned}
e_r(y^*)&=\frac{e(y^*)}{y^*}\approx d(\ln f(x_1^*,x_2^*,\cdots,x_n^*))\\
&=\sum_{i=1}^{n}\frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_i}\cdot\frac{(x_i-x_i^*)}{f(x_1^*,x_2^*,\cdots,x_n^*)}\\
&=\sum_{i=1}^{n}\frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_i}\cdot\frac{x_i^*}{f(x_1^*,x_2^*,\cdots,x_n^*)}\cdot e_r(x_i^*)
\end{aligned}
\tag{3.2}
$$

从（3.1）式和（3.2）式容易得到，进行四则运算时，初始数据误差和计算结果产生的误差之间有下面的关系：

（1） $f(x_1,x_2)=x_1\pm x_2$

$$
\begin{cases}
e(x_1^*\pm x_2^*)=e(x_1^*)\pm e(x_2^*)\\
e_r(x_1^*\pm x_2^*)=\dfrac{x_1^*}{x_1^*\pm x_2^*}e_r(x_1^*)+\dfrac{x_2^*}{x_1^*\pm x_2^*}e_r(x_2^*)
\end{cases}
\tag{3.3}
$$

（2） $f(x_1,x_2)=x_1x_2$

$$
\begin{cases}
e(x_1^*x_2^*)=x_2^*e(x_1^*)+x_1^*e(x_2^*)\\
e_r(x_1^*x_2^*)=e_r(x_1^*)+e_r(x_2^*)
\end{cases}
\tag{3.4}
$$

（3） $f(x_1,x_2)=\dfrac{x_1}{x_2}$

$$
\begin{cases}
e(\dfrac{x_1^*}{x_2^*})=\dfrac{1}{x_2^*}e(x_1^*)-\dfrac{x_1^*}{(x_2^*)^2}e(x_2^*)\\
e_r(\dfrac{x_1^*}{x_2^*})=e_r(x_1^*)-e_r(x_2^*)
\end{cases}
\tag{3.5}
$$

（3.1）、（3.2）式中的

$$
\frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_i}\ \text{和}\ \frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_i}\cdot\frac{x_i^*}{f(x_1^*,x_2^*,\cdots,x_n^*)},\qquad (i=1,2,\cdots,n)
$$

分别表示了数值计算的绝对误差和相对误差相对于初始数据误差经过传播后放大或缩小的倍数，也常常称作增长因子。从（3.3）式中我们看到，两个相近的数相减时，会将计算结果的相对误差变的很大。又从（3.5）式看到，作除法运算时，若分母 x_2^* 的绝对值相对很小时，其计算结果的绝对误差变得很大。

例 3.1 设 $y=x^n$，求 y 的相对误差与 x 的相对误差之间的关系。

解 由(3.2)式

$$
\begin{aligned}
e_r(y^*)&=\frac{x^*}{y^*}(\frac{d(y^*)}{dx})e_r(x^*)=\frac{x^*n(x^*)^{n-1}}{(x^*)^n}e_r(x^*)\\
&=ne_r(x^*)
\end{aligned}
$$

所以，y 的相对误差是 x 的相对误差的 n 倍。

例 3.2 已测得某矩形场地长 $l^*=110$ 米，宽 $m^*=80$ 米，$l=110\pm0.2$，$d=80\pm0.1$ 试求面

积 $s=lm$ 的绝对误差和相对误差。

解 因为

$$e(l^*)=0.2,\quad e(m^*)=0.1,\quad e_r(l^*)=\frac{0.2}{110},\quad e_r(m^*)=\frac{0.1}{80}$$

由（3.4）式可知

$$e(s^*)=m^*\cdot e(l^*)+l^*\cdot e(m^*)=80\times0.2+110\times0.1=27\ \text{（平方米）}$$

$$e_r(s^*)=e_r(l^*)+e_r(m^*)=\frac{0.2}{110}+\frac{0.1}{80}=\frac{27}{8800}=0.31\%$$

3.2 舍入误差对浮点运算的影响

当在计算机上执行算法时，由于计算机的有限字长和计数的方式不同，如整数的补码表示、实数的定点表示、实数的浮点表示等多种表示数的方法。假设计算机数码按十进制表示的字长为 t，我们提供给计算机的数 x 的最好近似值在用四舍五入（有时用只舍不入的断位）方法，用精度为 t 的浮点数可表示为

$$\tau(x)=\pm(0.a_1a_2\cdots a_t)\times10^s$$

其中，s 为整数（$|s|\le p$），称为 x 的阶码，$a_i(i=1,2,\cdots,t)$ 是 $0,1,\cdots,9$ 中的一个数，$a_1\neq0$。根据 §2 定理 2.1，它的相对误差满足

$$|e_r(\tau(x))|\le\frac{1}{2a_1}\times10^{1-t}\le\frac{1}{2}\times10^{1-t}=5\times10^{-t}$$

令 $\varepsilon=\dfrac{\tau(x)-x}{x}$，则有

$$\tau(x)=x(1+\varepsilon)\quad,\quad |\varepsilon|\le\frac{1}{2}\times10^{1-t}$$

再设计算机有 t 位的累加器，x、y 是精度为 t 的浮点数，则采用四舍五入法时，分句浮点运算过程，我们有

$$\tau(x+y)=(x+y)(1+\varepsilon_1)$$
$$\tau(x-y)=(x-y)(1+\varepsilon_2)$$
$$\tau(xy)=xy(1+\varepsilon_3)$$
$$\tau(\frac{x}{y})=\frac{x}{y}(1+\varepsilon_4)$$

其中相对误差 $|\varepsilon_i|\le\dfrac{1}{2}\times10^{1-t}$，$i=1,2,3,4$。这些结果表明，在有 t 位累加器的计算机上，浮点运算的结果有相对误差

$$|\varepsilon_i|\le\frac{1}{2\times10}\times10^{1-(t-1)}$$

由 §2 定理 2.1 可知，结果至少有 $t-1$ 位有效数字，即结果的有效数字可能损失一位。而在一般的断位计算机上情形更坏。

例 3.3 假设在 $t=4$ 的断位计算机上求 $x=0.12378$ 与 $y=0.12362$ 的差。

解 $\tau(x)-\tau(y)=0.1237\times10^{0}-0.1236\times10^{0}=0.1000\times10^{-3}$

$\tau(x)$，$\tau(y)$ 的相对误差分别为

$$\left|\frac{x-\tau(x)}{\tau(x)}\right|=0.6467\times10^{-3}\quad,\quad\left|\frac{y-f(y)}{f(y)}\right|=0.1618*10^{-3}$$

但是 $x-y=0.00016$，$\tau(x)-\tau(y)=0.0001$ 则

$$e_r[\tau(x)-\tau(y)]=\frac{0.00006}{0.0001}=0.6$$

$\tau(x)-\tau(y)=0.0001$ 只有 1 位有效数字，产生了相减相消，而相对误差是参与运算的近似数相对误差的 10^3 倍。

舍入误差对浮点运算的影响，这个问题比较复杂，但只要满足一定条件时，可将初始数据的实际浮点运算归结为初始数据的精确数学运算，例如

$$\tau(x+y)=(x+y)(1+\varepsilon)=x(1+\varepsilon)+y(1+\varepsilon).$$

3.3 算法的数值稳定性

计算一个数学问题，需要预先设计好由已知数据计算问题结果的运算顺序，这就是算法。自然，必须要求它能在计算机上迅速实现并且准确计算出结果。选用不同的算法，结果往往不相同，这主要是由于初始数据的误差和计数过程中的舍入误差，因不同的计算方法和不同的传播过程所形成的差异。我们将运算过程中舍入误差不增长的算法称作数值稳定的，否则就是数值不稳定的。

例 3.4 计算 $I_n=\int_1^e\ln(x)^n dx$，$(n=0,1,2,\cdots,11)$。

解 由分部积分可得 I_n 的递推公式

$$I_n=x(\ln x)^n\Big|_1^e-nI_{n-1}=e-nI_{n-1}\qquad(3.6)$$

$$I_0=\int_1^e dx=e-1$$

可设计如下两种算法：

算法 I $\begin{cases}I_0=1.718\\I_n=2.718-nI_{n-1}\end{cases}$ （$n=0,1,2,\cdots,11$）

计算结果见表 1-1 的 I_n^* 列。

显然，$0<\ln x<1$，$x\in(1,e)$，于是 $0<I_n<I_{n-1}$，这样

$$I_n=\frac{e-I_{n+1}}{n+1}<\frac{e}{n+1}\quad,\quad\frac{e}{n+1}<\frac{e-\frac{e}{n+1}}{n}<I_{n-1}=\frac{e-I_n}{n}<\frac{e}{n}$$

当 $n=12$ 时，取 I_n 的上、下界 $\frac{12}{e}$、$\frac{13}{e}$ 的平均值

$$I_n^*=0.2178$$