计算方法

田天海　许松林　编

武汉工业大学出版社

计 算 方 法

田天海　许松林　编

武汉工业大学出版社

图书在版编目(CIP)数据

计算方法/田天海,许松林编.—武汉:武汉工业大学出版社,1997.5

ISBN 7-5629-1279-3

Ⅰ.计… Ⅱ.①田… ②许… Ⅲ.计算方法-高等学校-教材 Ⅳ.O241

内容提要

本书是根据国家教委批准的高等工科院校《工科数学课程教学基本要求》(计算方法部分)编写的,着重介绍了电子计算机上常用的各种数值计算方法。其内容有:误差分析,非线性方程求根,线性代数方程组的直接解法与迭代解法,插值与拟合,数值微分与数值积分,常微分方程的数值解法,同时还给出了五个常用算法的BASIC程序。本书内容丰富,取材精练、实用,内容安排由浅入深,循序渐进,便于教学。

本书可作为高等工科院校高年级学生教材,也可供从事数值计算的科技工作者阅读参考。

武汉工业大学出版社出版发行

(武昌珞狮路14号 邮编:430070)

武汉市皇冠彩印厂印刷

*

开本:787×1092 1/32 印张:7.125 字数:153千字

1997年5月第1版 1997年5月第1次印刷

印数:1～2000册

定价:11.50元

前　言

随着科学技术迅速发展和计算机日益普及的需要，计算方法已是各理工科大学中普遍开设的一门课程，本书是根据国家教委批准的高等工科院校《工科数学课程教学基本要求》（计算方法部分），为工科院校的高年级学生学习计算方法课程而编写的教材，本教材着重介绍现代工程技术、科学研究和计算机上常用的数值方法和理论，取材力求精练、实用，编排上尽量由浅入深、循序渐进，读者只需具有高等数学和线性代数的知识，就能顺利地学习本书。本教材曾作为讲义在我院使用过多年，并作过一次修订，此次修改后正式出版。

本教材共分八章：第一章为数值计算中的误差，介绍了绝对误差、相对误差和算法的稳定性等概念；第二章为非线性方程求根，介绍了一元非线性方程的解法；第三章为线性代数方程组的直接解法，以高斯消元法及其变形为主线；第四章为线性代数方程组的迭代解法；第五章为插值与拟合，用代数多项式来近似地代替较复杂的函数或由表格给出的函数；第六章为数值微分与数值积分；第七章为常微分方程的数值解法，主要讨论一阶常微分方程初值问题的几个数值解法；第八章为计算方法实习，给出了具有典型意义的五个算法的 BASIC 程序。本书参考学时为 40 学时。

本书的第一至第四章、第七章由田天海编写，第五、六、八章由许松林编写，全书由田天海统稿。在编写过程中，得到了华中理工大学林化夷教授和王能超教授的指导和帮助，武汉工业大学出版社、湖北工学院教务处和基础课部对本书的编写和出版给予了大力支持，在此我们表示衷心地感谢。

限于作者的水平和经验，书中缺点乃至错误在所难免，望读者批评指正。

编　者

1997年元月

目　录

第一章　数值计算中的误差

用数值方法来求解科学研究和工程技术中的实际问题时，误差是不可避免的，这里既包含数学问题本身所蕴含的误差，也有由于数值方法本身所产生的误差。我们自然希望这些误差对于计算结果不会产生较大的影响，因此能否控制误差对计算结果产生的影响就成为衡量算法优劣的一个重要因素。在研究数值方法时，必须注重误差分析，分析误差的来源、误差的传播情况以及对计算结果给出合理的误差估计。本章的目的就在于认识误差并介绍几种控制计算误差的基本方法。

§1　误差的来源和定义

利用数值方法来求解实际问题，其误差的来源是多方面的，但主要有以下四个方面：

模型误差：在构造数学模型，即将实际问题转化为数学问题时，我们必须对实际问题进行一定的简化，加上一定的限制条件，忽略一些次要因素，以便于分析和计算，这就不可避免地要产生误差，我们把这种数学模型的解与实际问题的解之间出现的误差称为模型误差。由于这类误差很难作定性分析，因此我们总是假定数学模型是准确的，即在计算方法中不讨论模型误差。

观测误差：数学模型中一般都含有观测(或实验)数据，如时间、距离、速度、压力和电压等等，但由于观测手段的限制或一些偶然因素的影响，这些数据都不同程度地带有误差，这些误差称为观测误差。在计算方法中我们不研究这些误差。

截断误差：在计算中常常遇到只有通过无限过程才能得到的结果，但在实际计算时只能采用有限过程，于是产生了有限过程代替无限过程的误差，称为截断误差。

例如在讨论函数的计算时，通常用泰勒展开式的部分和来近似原始函数，对于指数函数

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots$$

我们用前 $n+1$ 项的部分和

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

来近似替代 e^x，则其余项

$$e^x - p_n(x) = \frac{x^{n+1}}{(n+1)!}e^{\theta x} \qquad (0 < \theta < 1)$$

称为截断误差。

由于截断误差的大小直接影响到计算结果，在计算方法中必须讨论截断误差。

舍入误差：由于计算机只能表示有限位数字，我们将计算机表示的数字与实际数字之差称为舍入误差。例如 $\pi = 3.1415926\cdots$，若计算机是 6 位的，则只能表示 $\pi \approx 3.14159$。通常计算机对不能表示的第一位数采用四舍五入。对于每个数据而言，其舍入误差是很小的，但当计算量很大时，其舍入误差是不可忽略的，因此计算方法必须研究舍入误差，对于舍入误差的研究是计算方法中一个基础而又重要的课题。

在研究误差时，必须对误差有一个合适的度量。如果对于精确值 u 有近似值 u^*，一个自然的想法是讨论它们的偏差 $u-u^*$，若偏差值较大则误差较大。但是偏差这种度量不一定能反映出误差对计算结果的影响程度。例如 $u_1=0.1, u_2=1000$，如果它们近似值分别为 $u_1^*=0.2, u_2^*=1000.1$，虽然它们的偏差均为 0.1，但这个偏差对 u_2 的影响不大，而对 u_1 的影响却是很大的，因为此时

$$\left|\frac{u_1-u_1^*}{u_1}\right|=100\% \qquad \left|\frac{u_2-u_2^*}{u_2}\right|=0.1\%$$

因此仅凭偏差 $u-u^*$ 不能完全反映误差的影响程度，故引入如下的绝对误差和相对误差的定义：

绝对误差：设 x^* 是准确值 x 的近似，

(1)称 $e(x)=x-x^*$ 为近似值 x^* 的误差；

(2)称 $|e(x)|=|x-x^*|$ 为近似值 x^* 的绝对误差，有时也简称误差。

通常 x 不能精确算出，也就无法准确算出误差 $e(x)$，但是实际问题往往可以估计出绝对误差 $|e|$ 不会超出某个数 ε，也就是绝对误差的一个上界，我们称 $\varepsilon(x)$ 为近似值的绝对误差限。例如用刻有毫米的尺测量某物的长度时，得近似值 125 毫米，即 $x^*=125$ 为 x 的近似。由于此时误差不应超出半毫米，即 $|x^*-x|\leqslant 0.5=\varepsilon$，虽然不能确定 x 到底为多少，但可以确定 x 在区间[124.5, 125.5]内。

相对误差：设 x^* 为准确值 x 的近似，称

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

为近似值 x^* 的相对误差。

若有 ε_r 存在，使得 $|e_r| \leqslant \varepsilon_r$，称 ε_r 为相对误差限，由于 x 不能准确算出，通常取

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

来作为 x^* 的相对误差。

由于计算机只能表示有限位数字，当准确值有多位数时按四舍五入的原则得到有限位数字。例如 $\pi = 3.14159265$，若取 3 位数，则 $x_1 = 3.14$，第四位 1 舍去，若取 4 位数，则 $x_2 = 3.1412$，第五位的 5 进 1。因此引入有效数字的定义。

有效数字：若近似值 x^* 的误差限是某一位的半个单位，从该位到 x^* 的第一位非零数字共有 n 位，则称 x^* 有 n 位有效数字。

若将 x 的近似值 x^* 写成如下的指数形式

$$x^* = \pm a_1 a_2 a_3 \cdots a_n \times 10^m$$

其中 a_1 是 1 到 9 中的一个数字，$a_2, \cdots, a_n$ 分别是 0 到 9 中的一个数字，m 为整数，若 x^* 的绝对误差限为

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

则称近似值 x^* 具有 n 个有效数字，$a_1, a_2, \cdots, a_n$ 是 x^* 的 n 位有效数字。

显然，在 m 相同的情况下，n 越大，则 10^{m-n+1} 越小，故有效位数越多，绝对误差越小。完全类似，当有效位数越多时，相对误差也越小。而且只要知道了有效数字位数，就容易写出它的绝对误差限和相对误差限。

例如 π 的近似值 $x_1 = 3.14$ 有 3 位有效数字，因为此时

$$|\pi - 3.14| < \frac{1}{2} \times 10^{-2}$$

$x_2=3.1416$ 则有 5 位有效数字，

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

而 $x_3=3.1415$ 只有 4 位有效数字，

$$|\pi - 3.1415| > \frac{1}{2} \times 10^{-4}$$

因此若某个近似数的最后一位是通过四舍五入得到的，则以最后一位到第一个非零数字的位数就是该近似数的有效位数。由于 3.1416 中的 6 是通过四舍五入得到的，所以计入有效位，而 3.1415 中的 5 是通过舍去 9 得到的，因此不能计入有效位。

§2 四则运算的误差估计

数值运算的误差估计一般可通过 Taylor 展开的方法来得到。若要计算 $A=f(x_1,x_2,\cdots,x_n)$，设 $x_1^*,x_2^*,\cdots,x_n^*$ 为 $x_1,x_2,\cdots,x_n$ 的近似值，记 $A^*=f(x_1^*,\cdots,x_n^*)$，则 A^* 的误差可通过 Taylor 展开的方法得到

$$e(A)=A-A^*=f(x_1,\cdots,x_n)-f(x_1^*,\cdots,x_n^*)$$

$$\approx \sum_{i=1}^{n}\frac{\partial f}{\partial x_i}(x_i-x_i^*)=\sum_{i=1}^{n}\frac{\partial f}{\partial x_i}e(x_i)$$

其绝对误差限 $\varepsilon(A)$ 和相对误差限 $\varepsilon_r(A)$ 分别为

$$|e(A)| \leqslant \sum_{i=1}^{n}\left|\frac{\partial f}{\partial x_i}\right||e(x_i)|$$

$$\leqslant \sum_{i=1}^{n}\left|\frac{\partial f}{\partial x_i}\right|\varepsilon(x_i)=\varepsilon(A) \qquad (1.1)$$

$$|e_r(A)| \leqslant \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i}\right| \frac{|x_i|\ |\varepsilon_r(x_i)|}{|A|} = \varepsilon_r(A)$$

考虑+、−、×、÷等四则运算，有如下绝对误差估计

$$\begin{aligned} e(x \pm y) &= |(x^* \pm y^*) - (x \pm y)| \\ &\leqslant |x^* - x| + |y^* - y| \\ &= e(x) + e(y) \leqslant \varepsilon(x) + \varepsilon(y) = \varepsilon(x \pm y) \end{aligned}$$

$$e(xy) = |x^* y^* - xy| \leqslant |x|\varepsilon(y) + |y|\varepsilon(x) = \varepsilon(xy)$$

$$e\left(\frac{x}{y}\right) = \left|\frac{x^*}{y^*} - \frac{x}{y}\right| \leqslant \frac{1}{|y|}\varepsilon(x) + \left|\frac{x}{y^2}\right|\varepsilon(y) = \varepsilon\left(\frac{x}{y}\right)$$

和如下相对误差估计

$$\varepsilon_r(x \pm y) = \left|\frac{x}{x+y}\right|\varepsilon_r(x) + \left|\frac{y}{x+y}\right|\varepsilon_r(y)$$

$$\varepsilon_r(xy) = \varepsilon_r(x) + \varepsilon_r(y)$$

$$\varepsilon_r\left(\frac{x}{y}\right) = \varepsilon_r(x) + \varepsilon_r(y)$$

由上可以看到，绝对误差估计比较适用于加减运算，而相对误差估计比较适用于乘除运算。

一般说来用上述方法所得到的误差限是比较保守的，即误差限比实际误差要大，但若误差限较小，可得到计算结果是可信的结论。

§3 设计数值算法的几个基本原则

本节通过几个具体的数值例子来说明设计数值算法的几个基本原则。

一、要使用数值稳定的算法

例 1.1 计算 $I_n = e^{-1}\int_0^1 x^n e^x dx$ $(n=0,1,2,\cdots)$并估计

误差。

解 由分部积分公式可以算出 I_n 的递推式

$$\begin{cases} I_0 = e^{-1}\int_0^1 e^x dx = 1 - e^{-1} \\ I_n = 1 - nI_{n-1} \qquad (n = 1,2,\cdots) \end{cases}$$

取 $I_0=0.632105$，并采用 8 位有效数字进行计算，将第 9 位进行四舍五入，有如下计算公式

算法 A：
$$\begin{cases} \bar{I}_0 = 0.632105 \\ \bar{I}_n = 1 - n\bar{I}_{n-1} \end{cases}$$

将计算结果列入表 1.1。从 I_n 的积分表达式可以看到 $I_n \geqslant 0$，但从 $n=15$ 开始其计算结果已是正负相间，因此计算结果误差较大。

由于当 $0 \leqslant x \leqslant 1$ 时，$1 \leqslant e^x \leqslant e$，故

$$\frac{e^{-1}}{n+1} = e^{-1}\int_0^1 x^n dx \leqslant e^{-1}\int_0^1 x^n e^x dx \leqslant e^{-1}\int_0^1 x^n e dx = \frac{1}{n+1}$$

将 I_n 近似表示为

$$I_n \approx \frac{1}{2}\left(\frac{e^{-1}}{n+1} + \frac{1}{n+1}\right) \doteq \bar{I}_n$$

取 $n=19$，则有如下的递推式：

算法 B：
$$\begin{cases} \bar{I}_{19} = \frac{1}{2}\left(\frac{e^{-1}}{20} + \frac{1}{20}\right) \\ \bar{I}_{n-1} = \frac{1}{n}(1 - \bar{I}_n) \qquad (n = 19,18,\cdots,1) \end{cases}$$

其计算结果见表 1.1 所示。

从计算结果来看算法 B 的计算结果是精确的。

下面来分析两种计算方法的误差传播情况，若记 $E_n = I_n - \bar{I}_n$，对于算法 A 有

$$E_n = (-1)nE_{n-1} = (-1)^2 n(n-1)E_{n-2} = \cdots = (-1)^n n!E_0$$

即 I_n 的误差是 I_0 的 $n!$ 倍，当 n 较大时，I_n 已无任何精度可言（见上表 $n=19$ 的计算结果）。而对于算法 B，有

$$E_{n-1} = \frac{1}{n}E_n$$

$$E_0 = \frac{1}{n!}E_n$$

因此即使 I_n 有一定误差，但计算结果可以使得误差逐步变小，因此其计算结果是可以接受的。

表 1.1 例 1.1 的计算数表

n	I_n(算法 A)	I_n(算法 B)	n	I_n(算法 A)	I_n(算法 B)
0	0.632105	0.632105	13	0.244777	0.0669477
2	0.22642411	0.2642411	14	⋯2.4268774	0.0627322
4	0.1708934	0.1708934	15	37.403116	0.0590174
7	0.1123836	0.1123835	18	−182836.88	0.0508317
10	0.093773	0.0837707	19	3473901.7	0.03419678

由上例可以看到，在数值计算中如不注意误差分析，选用了不稳定的数值算法就会算出精度很差的数值结果。对于这类问题，一个判别的标准是尽可能地使所采用的算法中(1.1)和(1.2)式中的 $\left|\frac{\partial f}{\partial x_i}\right|$ 或 $\left|\frac{\partial f}{\partial x_i}\right|\left|\frac{x_i}{A}\right|$ 的值较小。

二、防止有效位数的损失

两个相近的数相减会导致有效位数的损失，例如计算

$$A = 2/(\sqrt{201} - \sqrt{200})$$

采用 6 位有效数字进行计算，精确值为 $A=56.639165$。若直接计算上式可得 $\bar{A}=56.6572$，$\bar{A}$ 具有 4 位有效数字，若采用

下式计算

$$A = 2(\sqrt{201} + \sqrt{200})$$

可得计算结果 $A^* = 56.639$，A^* 具有五位有效数字，很明显 A^* 精度较高。

因此在实际计算时应尽可能地利用恒等变换来避免出现两相近数相减，例如

$$1 - \cos x = 2\sin^2 \frac{x}{2} \qquad (|x| \text{ 较小})$$

$$\lg x_2 - \lg x_1 = \lg \frac{x_2}{x_1} \qquad (x_1, x_2 \text{ 相近})$$

$$\operatorname{arctg}(x+1) - \operatorname{arctg} x = \operatorname{arctg} \frac{1}{1 + x(x+1)} \qquad (x \text{ 充分大})$$

另外也可以采用双精度进行计算。

三、防止大数"吃掉"小数

计算机在进行加法运算时都要进行对阶，即将二数均写成绝对值小于 1 而阶码相同的数，此时阶码小的数的阶码要升到和阶码大的数的一样，例如 $x_1 = 10^9$，$x_2 = 1$，设计算机能表示到小数点后的 8 位，则计算机运算结果为

$$\begin{aligned} x_1 + x_2 &= 10^9 + 1 = 0.1 \times 10^{10} + 0.1 \times 10 \\ &= 0.1 \times 10^{10} + 0.0000000001 \times 10^{10} \text{(对阶)} \\ &= 0.1 \times 10^{10} = 0.1 \times 10^{10} \end{aligned}$$

由于计算机只能表示小数点后 8 位，因此将 0.0000000001 按第 9 位四舍五入得到零。这样较小的数 1 被较大的数 10^9"吃掉"了。因此在设计计算方法时，应先计算同一数量级上的数，例如

$$A = 12345 + 0.1 + 0.3 + 0.7 + 0.9$$

则应先计算 0.1+0.3+0.7+0.9=2，再将 2 和 12345 相加。

四、简化计算步骤、减少运算次数

减少运算次数，不仅可以减少计算机的计算时间，还可以减少舍入误差的积累。例如计算多项式

$$p_n(x) = a_n x^n + \cdots + a_1 x + a_0$$

一种算法为先直接计算 $a_i x^i$，再逐项相加，采用这种算法共需做 $\frac{(n+1)n}{2}$ 次乘法和 n 次加法。下面将多项式变形，将前 n 项提出 x

$$\begin{aligned} p_n(x) &= a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \\ &= (a_n x^{n-1} + a_{n-1}x^{n-2} + \cdots + a_1)x + a_0 \end{aligned}$$

括号内为 $n-1$ 次多项式，对它再施行同样手续，反复下去，最后得到如下嵌套结构

$$p_n(x) = [\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_1]x + a_0$$

利用上式结构上的特点，从里往外一层层地计算，就得到如下秦九韶算法

$$\begin{cases} s_n = a_n \\ s_i = x s_{i+1} + a_i \qquad (i = n-1, \cdots, 2, 1, 0) \\ s_0 = p_n(x) \end{cases}$$

易见秦九韶算法的计算量要小得多，仅需 n 次乘法和 n 次加法运算。

习　题　一

1. 按四舍五入的原则，将下列各数舍入成五位有效数字：

816.9567　　6.000015　　17.32250