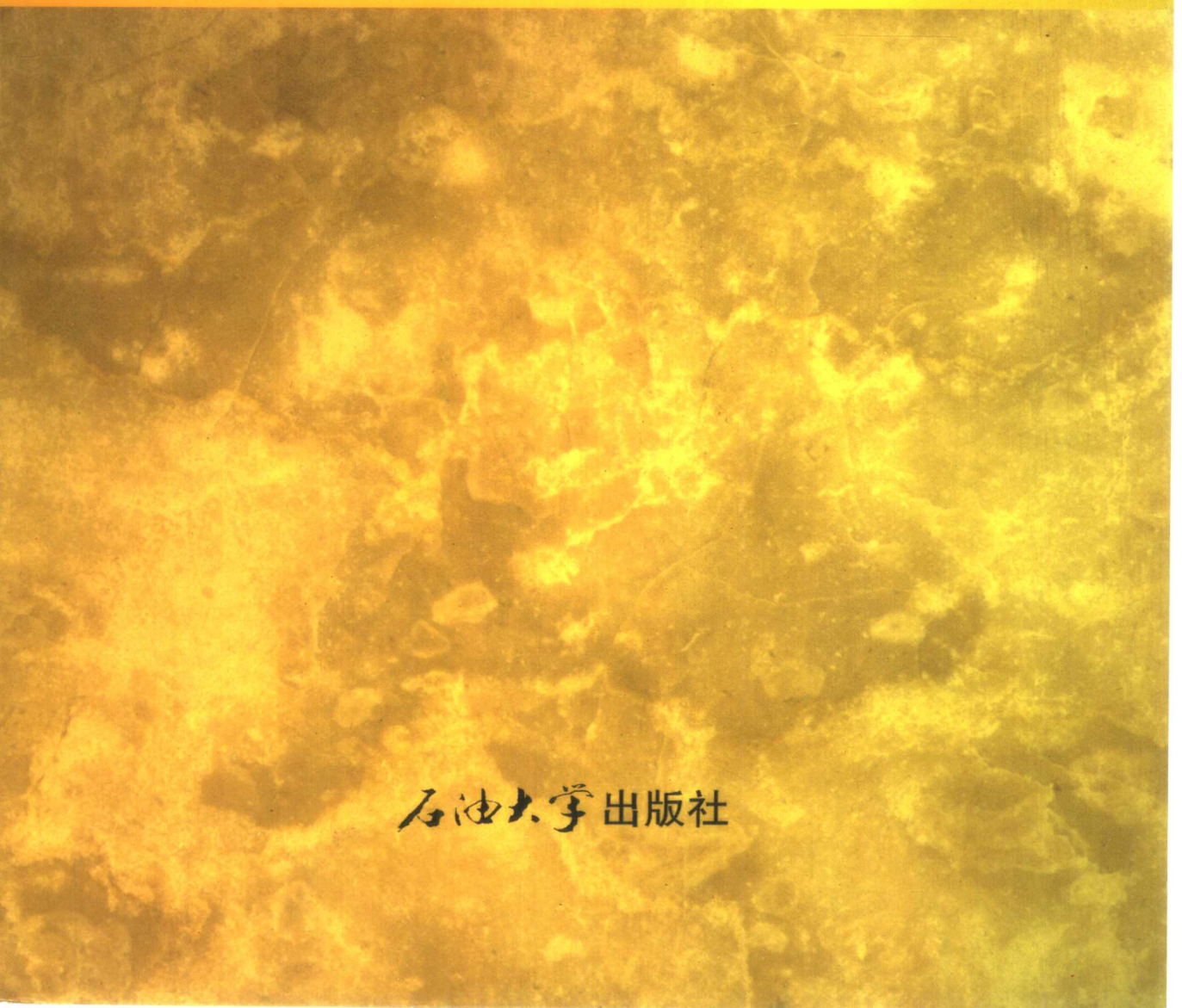
数值计算方法

李维国 黄炳家 同登科 王子亭 编著



石油大学出版社

数值计算方法

李维国　黄炳家　同登科　王子亭　编著

石油大学出版社

内容简介

本书是高等学校信息与计算科学专业本科生的《数值计算方法》或《数值分析》的一本面向21世纪的新教材。全书共分十章。第一章:绪论;第二章:非线性方程(组)的数值解法;第三章:数值逼近;第四章:数值积分与微分;第五章:线性方程组的直接解法;第六章:线性方程组的迭代解法;第七章:线性最小二乘问题;第八章:特征值问题的计算方法;第九章:常微分方程初值问题的数值解法;第十章:常微分方程边值问题的数值解法。每章均附有数值实验习题(包括实验目的、问题提出与实验要求)。本书是针对近年来本专业学生接受计算机知识越来越早,已有初步的信息处理与计算能力,对一些数值软件有一点了解,然而缺乏系统的数值计算理论和计算方法的设计思想而编写的。可作为理工科大学、高等师范院校信息与计算科学专业及其他相关专业的教材或教学参考用书,或该专业《数值逼近》与《数值代数》课程的相应教材。也可供从事科学与工程计算的科技人员参考。

图书在版编目(CIP)数据

数值计算方法/李维国主编 .—东营:石油大学出版社,2004.3

ISBN 7-5636-1933-X

Ⅰ.数… Ⅱ.李… Ⅲ.数值计算-计算方法 Ⅳ.0241

中国版本图书馆 CIP 数据核字(2004)第 024860 号

书　　名:数值计算方法
作　　者:李维国　黄炳家　同登科　王子亭　编著
出 版 者:石油大学出版社(山东 东营,邮编 257061)
网　　址:http://sunctr.hdpu.edu.cn
电子信箱:upcpress@mail.hdpu.edu.cn
排 版 者:石油大学出版社排版中心
印 刷 者:东营市新华印刷厂
发 行 者:石油大学出版社(电话　0546—8395977)
开　　本:787×1092　印张:19.25　字数:491 千字
版　　次:2004 年 8 月第 1 版第 1 次印刷
印　　数:1~3000 册
定　　价:27.00 元

前　言

Foreword

几年以来，我系从事计算数学与应用数学教学与科研的教师就筹划编写一本信息与计算科学专业学生使用的较适中的教材。经过我们编写组全体同仁的艰苦努力，今天终于与读者见面了。

由于电子计算机的迅速发展，近年来本专业学生接受计算机知识越来越早，且已有初步的信息处理与计算能力，对一些数值软件有一点了解，然而缺乏系统的数值计算理论和计算方法的设计思想。考虑到21世纪本专业教学改革的特点和对学生在数值计算方面的要求，根据我们多年的教学实践和系统性与科学性相结合的原则，并注意到我们在使用国内外同类教材时发现的一些问题，在深度和广度上做恰当的处理。力求得到一本既有理论深度和较系统的专业基础知识，又不使其过于专门化而包罗万象的教材。本教材旨在通过对一些典型问题和典型算法的剖析，使学生循序渐进地掌握本课程的基本理论和分析解决问题的基本思路与技巧。

全书分三部分，共十章。第一部分为数值分析(第一章至第四章)；第二部分为数值代数部分(第五章至第八章)；第三部分为常微分方程数值解法(第九章至第十章)。每章还附有数值实验(包括实验目的、问题提出与实验要求)，它是本书的又一特色，给教授本课程的教师和学生提供一些训练的素材。全书讲授时数为80～96个学时(也可增加16个学时的实验课)。

本书第一章和第八章由李维国博士编写；第二章、第三章和第四章由同登科博士编写；第五章、第六章和第七章由黄炳家副教授编写；第九章、第十章及各章的数值实验由王子亭博士编写。全书最后由李维国统稿。

学习本书所需要的数学基础是微积分和线性代数，以及常微分方程的基本概念。本书还附有一定数量的习题，通过这些习题可以加深对各章内容的理解，掌握必要的解题技巧。本书可作为理工科大学、高等师范院校信息与计算科学专业及其他相关专业的教材或教学参考用书，或该专业《数值逼近》与《数值代数》课程的相应教材。也可供从事科学与工程计算的科技人员参考。

我们特别感谢南京大学沈祖和教授，他仔细阅读了全书并提出了宝贵意见。我系研究生鲍文娣、陈金海、王磊、阮宗利、石丽娜等也做了部分工作，在此一并表示感谢。承蒙石油大学出版社为本书的出版给予了大力的支持。我们希望使用本书的老师、同学及广大读者对本书提出批评指正。

编　者

2004年3月

目 录

Contents

第一章　绪　论

数值计算方法(Numerical Computational Method)又称**数值分析**(Numerical Analysis),是研究适合计算机求解的各种数学问题的近似方法及其理论。它的内容包括函数逼近、数值微分与积分、非线性方程(组)的数值解、数值代数、常微分与偏微分方程数值解等。

自1946年第一台电子计算机问世以来,经过半个多世纪的发展,计算机对科学技术的冲击是极其深远的,其主要原因当然是由于计算机已经并将继续大大地扩展问题的可解范围。为此也使得科学与工程计算成为上个世纪最重要的科学进步之一。大型科学与工程计算是现代科学、工程和技术发展的重要组成部分,特别是涉及国防、能源、航天和气象等技术型密集的行业更是如此。当今高度复杂的科学与工程问题的求解只有一小部分能够用解析的方法解决,其他大部分要通过物理实验、数值计算来揭示其内在的规律。而数值计算包括了从适当的计算机结构设计出发,对相应的数学模型进行合理的算法设计,然后进行充分的数值试验、分析直至研制出相应的数值软件。另外,由于现代科学与工程计算的复杂性,数值实验还能代替某些物理实验无法做到的事情。如今**数值计算**已与**理论研究**及**物理实验**并列成为当今世界科学活动的三种主要方式。为众多的科学与工程问题提供计算方法、提高计算的可靠性、有效性和精确性,便是数值计算方法这门课程的主要研究内容。

§1　误　差

1.1 误差的来源

用计算机解决科学计算问题首先要建立数学模型,它是对被描述的实际问题进行抽象、简化而得到的,因而是近似的。我们把数学模型与实际问题之间出现的这种误差称为**模型误差**。在数学模型中往往还有一些根据观测得到的物理量,如温度、长度、电压等等,这些参量显然也包含误差,这种由观测产生的误差称为**观测误差**。这两种误差不属于本书的讨论范围。

当数学模型得不到精确解时,通常要用数值方法求它的近似解,或者说用数值算法模拟数学模型。此时产生的误差称为**方法误差**或**截断误差**。例如,求 $I=\int_0^1 \frac{\sin x}{x}\mathrm{d}x$ 的解,采用所谓梯形公式 $\hat{I}=[\sin 1+1]/2$ 来近似,这时 $\hat{I}$ 与 I 之间的误差称为梯形公式求解定积分 I 的截断误差。

有了求解数学模型或数学问题的算法以后,用计算机进行数值计算时,由于计算机的字长有限,原始数据在计算机上表示会产生误差,计算过程又可能产生新的误差,这种用计算机模拟或实现算法的误差称为**舍入误差**。例如,用5位有效数字计算 $\hat{I}$ 得到

$$\hat{I}\approx[0.841\,47+1.000\,0]/2.000\,0\approx 0.920\,74=\tilde{I}\ ,$$

$\hat{I}$ 与 $\tilde{I}$ 之间的误差称为舍入误差。

截断误差与舍入误差是用数值方法求解数学问题产生的误差，是数值计算中主要讨论的误差。

1.2 误差分析的基本概念

本书除了研究数学问题的算法外，还要研究计算解(数值解)与真解(精确解)相差多少，这就是计算的精确度问题，需要对误差做出估计和分析。

• **定义 1.1** 设 x 为真值(精确值)，x^* 为 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。

显然，误差 e^* 可正可负，且常常是无限位的，有时我们不能也没有必要求得它的一个精确结果，只需知道它的绝对值的一个上界 ε^*，这个上界称为**绝对误差限**。如取 $\pi^* = 3.141\ 59$，则

$$|\pi^* - \pi| \leqslant \frac{1}{2} \times 0.000\ 01,$$

即 $\varepsilon^* = 0.000\ 005$，或 $3.141\ 585 \leqslant \pi \leqslant 3.141\ 595$，有时记作 $\pi = 3.141\ 59 \pm 0.000\ 005$。

误差的大小还不能完全表示近似值的好坏。若 x 的近似值 x^* 是由 x 按“四舍五入”规则得到的，则，

$$|x^* - x| \leqslant \frac{1}{2}\alpha,$$

这里 α 是 x^* 的最后一位的一个单位。

譬如测量一段路程，其长为 1 000 km，知道有误差 20 m；另外测一条 400 m 的跑道，也有 20 m 误差，容易想像后者的精度不如前者。这就是相对误差的概念。

• **定义 1.2** 近似值 x^* 的误差 e^* 与准确值 x 的比值

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

称为近似值 x^* 的相对误差，记作 e_r^*。在实际计算时，由于真值常常是未知的，通常取

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

作为 x^* 的相对误差，条件是 $e_r^* = \frac{e^*}{x^*}$ 较小，此时

$$\frac{e^*}{x} - \frac{e^*}{x^*} = \frac{e^*(x^* - x)}{x^* x} = \frac{(e^*)^2}{x^*(x^* - e^*)} = \frac{(e^*/x^*)^2}{1 - e^*/x^*}。 \tag{1-1}$$

相对误差也可正可负，它的绝对值的上界称为**相对误差限**，记做 ε_r^*，$\varepsilon_r^* = e^*/x^*$。由此知道测量 1 000 km 的路程有 20 m 误差，其相对误差为 $20/10^6 = 2 \times 10^{-5}$，而测量 400 m 的跑道有 20 m 的误差，它的相对误差为 $20/400 = 5\%$。由此可见前者的相对误差比后者小多了。

在误差分析中，还常常用到有效数字的概念。

• **定义 1.3** 若近似值 x^* 与准确值的误差不超过某一位的半个单位，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。

例 1.1 ① 取 $x^* = 3.14$ 作 π 的近似值，x^* 就有 3 位有效数字；

② 取 $x^*=3.141\ 592$ 作 π 的近似值，x^* 就有 6 位有效数字，其中 2 不是有效数字；

③ 取 $x^*=3.141\ 593$ 作 π 的近似值，x^* 就有 7 位有效数字；

④ 取 $x^*=0.030\ 141\ 60$ 作 $x=0.030\ 141\ 594$ 的近似值，x^* 就有 7 位有效数字；

用数学的语言，即若 x 的近似值写作：

$$x^*=\pm 10^m\times(a_1\times 10^{-1}+a_2\times 10^{-2}+\cdots+a_n\times 10^{-n}+\cdots+a_k\times 10^{-k}+\cdots) \tag{1-2}$$

其中 m 是整数 $a_1\neq 0, a_1, a_2, \cdots, a_k$ 是 0 到 9 中的一个数字，若 $|x^*-x|\leqslant\frac{1}{2}\times 10^{m-n}$，则 x^* 至少具有 n 位有效数字，即 $a_1, a_2, \cdots, a_n$ 为有效数字，而 $a_{n+1}, \cdots, a_k, \cdots$ 等不一定是有效数字。

若 x^* 的每一位都是有效数字，那么称 x^* 为有效数。显然，若 x^* 是由 x 经“四舍五入”而来，则 x^* 是**有效数**。有效数字与相对误差之间有密切关系，我们叙述如下定理，证明参见[1]。

定理 1.1　将 x 的近似值 x^* 表示成式(1-2)，若 x_k 是有效数字，那么相对误差不超过 $\frac{1}{2}\times 10^{-(k-1)}$；反之，如果已知相对误差 r，且有 $|r|\leqslant\frac{1}{2}\times 10^{-k}$，那么 x_k 必为有效数字。

1.3 数值算法与算法的数值稳定性

数值问题是指输入数据(即问题中的自变量与原始数据)与输出数据(结果)之间函数关系的一个确定而无歧义的描述。输入输出数据可用有限维向量表示。根据这种定义，“数学问题”不一定是“数值问题”，但它往往可用“数值问题”来逼近。例如，解常微分方程 $\frac{dy}{dx}=x^2+y^2, y(0)=0$，它不是数值问题，因为输出不是数据而是连续函数 $y=y(x)$，但只要规定输出数据是 $y(x)$ 在 $x=h, 2h, \cdots, nh$ 处的近似值，这就是一个数值问题，它可用 Euler 折线法或其他数值方法(见第九章)求解，这些数值方法就是算法。

计算的基本单位称为**算法元**，它由算子、输入元和输出元组成，算子可以是简单操作如算术运算 +、−、×、÷，逻辑运算，也可以是宏操作如向量运算、数据传输、函数求值等。输入元和输出元分别可视为若干变量或向量。由一个或多个算法元组成一个**进程**，它是算法元的有限序列。一个数值问题的**算法**是指按规定顺序执行一个或多个完整的进程。通过它们将输入元变成一个输出元。面向计算机的算法可分为串行算法与并行算法两类。只有一个进程的算法适用于串行计算机，称为**串行算法**；两个与两个以上进程的算法适合于并行计算机，称为**并行算法**。对于一个给定的数值问题可以有许多不同的算法，它们都可以给出近似答案，但所需计算量和得到的精度可能相差很大。一个面向计算机，计算复杂性好，又有可靠理论分析的算法就是一个好算法。所谓**计算复杂性**包含时间复杂性和空间复杂性两方面，在同一精度下，计算时间少的为时间复杂性好，而占用内存空间少的为空间复杂性好。

例 1.2　计算下列多项式的值。

$$p(x)=a_0x^n+\cdots+a_{n-1}x+a_n$$

这是一个数值问题，输入数据为 $a_0, \cdots, a_n$ 及 x，输出数据为 $p(x)$，若直接由 x 算出 $x^2, \cdots, x^n$，再乘相应的系数 $a_{n-1}, a_{n-2}, \cdots, a_0$ 并相加，则要做 $\frac{n(n+1)}{2}$ 次乘法和 n 次加法，占用

$2n+1$个存储单元。若将 $p(x)$改写为

$$p(x) = (\cdots(a_0x + a_1)x + \cdots + a_{n-1})x + a_n$$

用递推公式表示为

$$b_0 = a_0, \quad b_i = a_i + b_{i-1}x, \quad i = 1,2,\cdots,n, \quad b_n = p_n(x)$$

它只用 n 次乘法和 n 次加法，并占用 $n+2$ 个存储单元，故这是一个好的串行算法，它称为**秦九韶方法**，也称为 **Horner 算法**。

对于大型计算问题，不同的计算复杂性差别就更大。例如解线性方程组，当 $n=20$ 时，用 Cramer 法则，仅乘除法的运算次数就需约 9.7×10^{20}，用每秒 1 亿次的计算机也要算 30 多万年，而用 Gauss 消去法只需 2660 次乘除运算。并且 n 愈大相差就愈大。这个例子既表明算法研究的重要性，又说明只提高计算机速度而不改进和选用好的算法也是不行的。人类的计算能力是计算工具的性能和计算效率的总和，因此，计算能力的提高有赖于两方面的提高。例如，1955 年至 1975 年的 20 年间，计算机速度提高数千倍，而同一时间解决一定规模的椭圆型偏微分方程计算方法效率提高约 100 万倍，这说明研究和选择好的算法对提高计算速度，在某种意义上说比提高计算机速度更重要，因为算法研究所需代价要小得多。当然，选择好算法的前提是保证计算结果的可靠性。这就要求有可靠的理论分析，使计算结果满足精度要求。一个算法是否可靠与舍入误差是否增长密切相关。

• **定义 1.4** 一个算法如果输入数据有扰动(即误差)，而计算过程中舍入误差不增长则称此算法是数值稳定的，否则此算法就称为数值不稳定的。

例 1.3 对 $n=0,1,\cdots,8$，计算积分 $\int_0^1 \frac{x^n}{x+5}\mathrm{d}x$。

解 由于

$$y_n + 5y_{n-1} = \int_0^1 \frac{x^n + 5x^{n-1}}{x+5}\mathrm{d}x = \frac{1}{n}$$

于是可得到计算积分 y_n 的递推公式

$$y_n = \frac{1}{n} - 5y_{n-1}, \quad n = 1,2,\cdots,8 \tag{1-3}$$

其中

$$y_0 = \int_0^1 \frac{1}{x+5}\mathrm{d}x = \ln(x+5)\Big|_0^1 = \ln\frac{6}{5} \approx 0.182 = \tilde{y}_0$$

利用公式(1-3)计算 y_n，计算取到小数点后 3 位，由于初值 y_0 用 $\tilde{y}_0$ 近似，实际计算结果为

$$\tilde{y}_n = \frac{1}{n} - 5\tilde{y}_{n-1}, \quad n = 1,2,\cdots,8。 \tag{1-4}$$

若用其他方法求 y_n 的精确解(精确到小数点后 6 位)，计算结果见下表。

表 1-1

n	y_n	$\tilde{y}_n$	$\bar{y}_n$
0	0.182 322	0.182	0.182
1	0.088 392	0.090	0.088

续表 1-1

n	y_n	$\tilde{y}_n$	$\bar{y}_n$
2	0.058 039	0.050	0.058
3	0.043 139	0.083	0.043
4	0.034 306	-0.165	0.034
5	0.028 468	1.025	0.028
6	0.024 325	-4.958	0.025
7	0.021 231	24.933	0.021
8	0.018 846	-124.540	0.019

从表中可以看到，$\tilde{y}_n$ 的结果误差很大，$n \geqslant 4$ 时已经完全失真。实际上，容易估计出 $\frac{1}{6(n+1)} < y_n < \frac{1}{5(n+1)}$），而 $\tilde{y}_4 < 0$ 是严重失真。这里计算公式(1-4)是精确的，误差都是由于 y_0 有微小误差 $\varepsilon_0 = y_0 - \tilde{y}_0 < \frac{1}{2} \times 10^{-3}$ 引起的，记 $\varepsilon_n = y_n - \tilde{y}_n$，由式(1-3)减式(1-4)得

$$\varepsilon_n = -5\varepsilon_{n-1} = \cdots = (-1)^n 5^n \varepsilon_0$$

这表明误差 ε_n 增长很快，故算法式(1-3)不稳定。

现在从另一方向使用这一公式，由式(1-3)得

$$y_{n-1} = \frac{1}{5}\left(\frac{1}{n} - y_n\right), \quad n = 8,7,\cdots,1 \tag{1-5}$$

若取 $\bar{y}_8 = 0.019$，$|\varepsilon_8| = |y_8 - \bar{y}_8| < \frac{1}{2} \times 10^{-3}$，由式(1-5)对 $n = 8,7,\cdots,1$ 算出 $\bar{y}_{n-1} = \frac{1}{5}\left(\frac{1}{n} - \bar{y}_n\right)$，此时所有 $\bar{y}_7,\cdots,\bar{y}_0$ 与 $y_7,\cdots,y_0$ 比较都有 3 位有效数字，见表 1-1，此时 $\varepsilon_{n-1} = (-1)\left(\frac{1}{5}\right)\varepsilon_n$，$\varepsilon_{8-k} = (-1)^k\left(\frac{1}{5}\right)^k \varepsilon_8$，$k = 1,2,\cdots,8$，当 k 增大时，$|\varepsilon_{8-k}|$ 是减少的，故它是数值稳定的。

数值不稳定的算法是不能使用的，有时即便使用双精度进行运算也是不行的。

对算法而言，有稳定性问题，对数学问题本身如果输入数据有微小扰动，引起输出数据(即问题真解)的很大扰动，这就是**病态问题**。它是数学问题本身性质所决定的，与算法无关，也就是说对病态问题，用任何算法(或方法)直接计算都将产生不稳定性。如下例。

例 1.4　求方程组

$$\begin{cases} x + ay = 1 \\ ax + y = 0 \end{cases}$$

的解。

当 $a = 1$ 时，系数矩阵奇异，解不存在。当 $a \neq 1$ 时，解为 $x = (1-a^2)^{-1}$，$y = -a(a-a^2)^{-1}$。例如，当 $a = 0.99$ 时，$x \approx 50.25$；若 $\hat{a} = 0.991$，则解 $\tilde{x} \approx 55.81$。误差 $\tilde{x} - x = 5.56$，由此看出问题是病态的。而当 $a \ll 1$ 时问题就是良态的。这时，a 的小变化就不会引起

解的大变化。

病态与良态是一个定性的概念，是相对的，没有严格的界限，通常判断问题是否病态，可通过问题的某些特征来衡量。对病态问题，其计算结果一般是不可靠的，通常应改变问题提法以改善条件数，或采用特殊处理方法，或用高精度运算以减少舍入误差等。

§2 误差分析的方法与原则

数值运算中的误差分析是个重要而复杂的问题，对于一些简单情形，可以利用误差限，随着计算过程逐步向前进行分析，直至估计出最后的结果，这种方法称为**向前误差分析法**。例如，两个近似数 x_1^* 与 x_2^*，其误差限分别为 $\varepsilon(x_1^*)$ 和 $\varepsilon(x_2^*)$，它们进行四则运算得到的误差限分别为：

$$\varepsilon(x_1^* \pm x_2^*) = \varepsilon(x_1^*) + \varepsilon(x_2^*);$$

$$\varepsilon(x_1^* \times x_2^*) \approx |x_1^*|\varepsilon(x_2^*) + |x_2^*|\varepsilon(x_1^*);$$

$$\varepsilon\left(\frac{x_1^*}{x_2^*}\right) \approx \frac{|x_1^*|\varepsilon(x_2^*) + |x_2^*|\varepsilon(x_1^*)}{|x_2^*|^2};$$

更一般的情况是，当自变量有误差时计算函数值也产生误差，其误差限可利用函数的展开式进行估计。设 $f(x)$ 是一元函数时，x 的近似值为 x^*，以 $f(x^*)$ 近似 $f(x)$，其误差界记作 $e(f(x^*))$，可利用 Tayler 展开

$$f(x) - f(x^*) = f'(x^*)(x - x^*) + \frac{f''(\xi)}{2}(x - x^*)^2, \xi \text{ 介于 } x \text{ 与 } x^* \text{ 之间},$$

取绝对值得

$$|f(x) - f(x^*)| \leqslant |f'(x^*)|\varepsilon(x^*) + \frac{|f''(\xi)|}{2}\varepsilon^2(x^*)。$$

假定 $f'(x^*)$ 与 $f''(x^*)$ 的比值不太大，可忽略 $\varepsilon(x^*)$ 的高阶项，于是可得计算函数的误差限

$$\varepsilon(f(x^*)) \approx |f'(x^*)|\varepsilon(x^*)。$$

当 f 为多元函数时，如计算 $A = f(x_1, x_2, \cdots, x_n)$。如果 $x_1, x_2, \cdots, x_n$ 的近似值为 $x_1^*, x_2^*, \cdots, x_n^*$ 则 A 的近似值分别为 $A^* = f(x_1^*, x_2^*, \cdots, x_n^*)$，于是函数值 A^* 的误差 $e(A^*)$ 由 Taylor 展开得

$$\begin{aligned} e(A^*) = A^* - A &= f(x_1^*, x_2^*, \cdots, x_n^*) - f(x_1, x_2, \cdots, x_n) \\ &\approx \sum_{k=1}^{n}\left(\frac{\partial f(x_1^*, x_2^*, \cdots, x_n^*)}{\partial x_k}\right)(x - x_k) \\ &= \sum_{k=1}^{n}\left(\frac{\partial f}{\partial x_k}\right)e_k^*, \end{aligned}$$

于是得误差限

$$\varepsilon(A^*) \approx \sum_{k=1}^{n}\left|\left(\frac{\partial f}{\partial x_k}\right)\right|\varepsilon(x_k^*)。$$

而 A^* 的相对误差限为

$$\varepsilon_r^* = \varepsilon_r(A^*) = \frac{\varepsilon(A^*)}{|A^*|} \approx \sum_{k=1}^{n} \left|\left(\frac{\partial f}{\partial x_k}\right)\right| \frac{\varepsilon(x_k^*)}{|A^*|}。$$

20世纪60年代，数值专家Givens与Wilkinson等人提出了所谓**向后误差分析法**，其基本思想是把舍入误差的累积与导出 A^* 的已知量 $x_1, x_2, \cdots, x_n$ 的某种摄动(微小误差)等价起来，即对某个 x_i 引进某个摄动量 ε_i，使得精确地成立等式

$$A^* = f(x_1 + \varepsilon_1, x_2 + \varepsilon_2, \cdots, x_n + \varepsilon_n),$$

并推出这些 ε_i 的界(并非要得出 ε_i 的具体值，ε_i 不是惟一的)，然后利用摄动理论估计最后的舍入误差界。Wilkinson将这种方法应用于数值代数(矩阵运算)的误差分析，取得较好的效果。

区间分析法是把参加运算的数 $x, y, z, \cdots$ 都看成区间量 $X, Y, Z, \cdots$，根据区间运算规则求得最后结果的近似值及误差限。例如，x, y 的近似值分别为 x^*, y^*，由于 $|x - x^*| \leqslant \varepsilon(x^*)$，$|y - y^*| \leqslant \varepsilon(y^*)$，则

$$x \in [x^* - \varepsilon(x^*), x^* + \varepsilon(x^*)] = X,$$
$$y \in [y^* - \varepsilon(y^*), y^* + \varepsilon(y^*)] = Y。$$

若计算 $z = xy$，由 $Z = XY = [\underline{z}, \bar{z}] = [z - \varepsilon(z), z + \varepsilon(z)]$，则 z 为所求近似值，而 $\varepsilon(z)$ 则为误差限。

然而一个科学与工程计算问题往往要运算千万次，由于每步运算都可能有误差，如果每步都做误差分析是非常繁琐的，况且误差积累有正有负，绝对值有大有小，都按最坏情况估计误差限得到的结果比实际误差大得多，这种保守的误差估计不反映实际误差积累。考虑到误差分布的随机性，有人用概率统计方法，将数据和运算中的舍入误差视为适合某种分布的随机变量，然后确定计算结果的误差分布，这样得到的误差估计更接近实际，这种方法称为**概率分析法**。

上述四种误差分析的方法，是常见的误差定量分析方法，由于数值问题的复杂性，除非十分必要，我们一般只做定性分析。在本书中对各计算过程都只研究它的数值稳定性，而不具体估计舍入误差。这里只提出数值运算中应注意的若干原则，它有助于鉴别计算结果的可靠性并防止误差危害的现象产生。

1. 要尽量避免除数绝对值远远小于被除数绝对值的除法

用绝对值小的数作为除数，舍入误差会增大，如计算 $\frac{x}{y}$，若 $0 < |y| \ll |x|$，则当 y 有很小的舍入误差时，就可能对计算结果带来严重影响。应尽量避免这种现象产生。

2. 要尽量避免两相近数相减

在数值计算中两相近数相减会大大损失有效数字。故在近似计算中，常常改变计算方法，以提高计算精度。现举例说明如下。

例 1.5　计算 $A = 10^7(1 - \cos 2°)$(用四位数学用表).

由于 $\cos 2° = 0.9994$，直接计算

$$A = 10^7(1 - \cos 2°) = 10^7(1 - 0.9994) = 6 \times 10^3$$

只有一位有效数字，若利用 $1 - \cos x = 2\sin^2\frac{x}{2}$，则

$$A = 10^7(1 - \cos 2°) = 2 \times (\sin 1°)^2 \times 10^7 = 6.13 \times 10^3$$