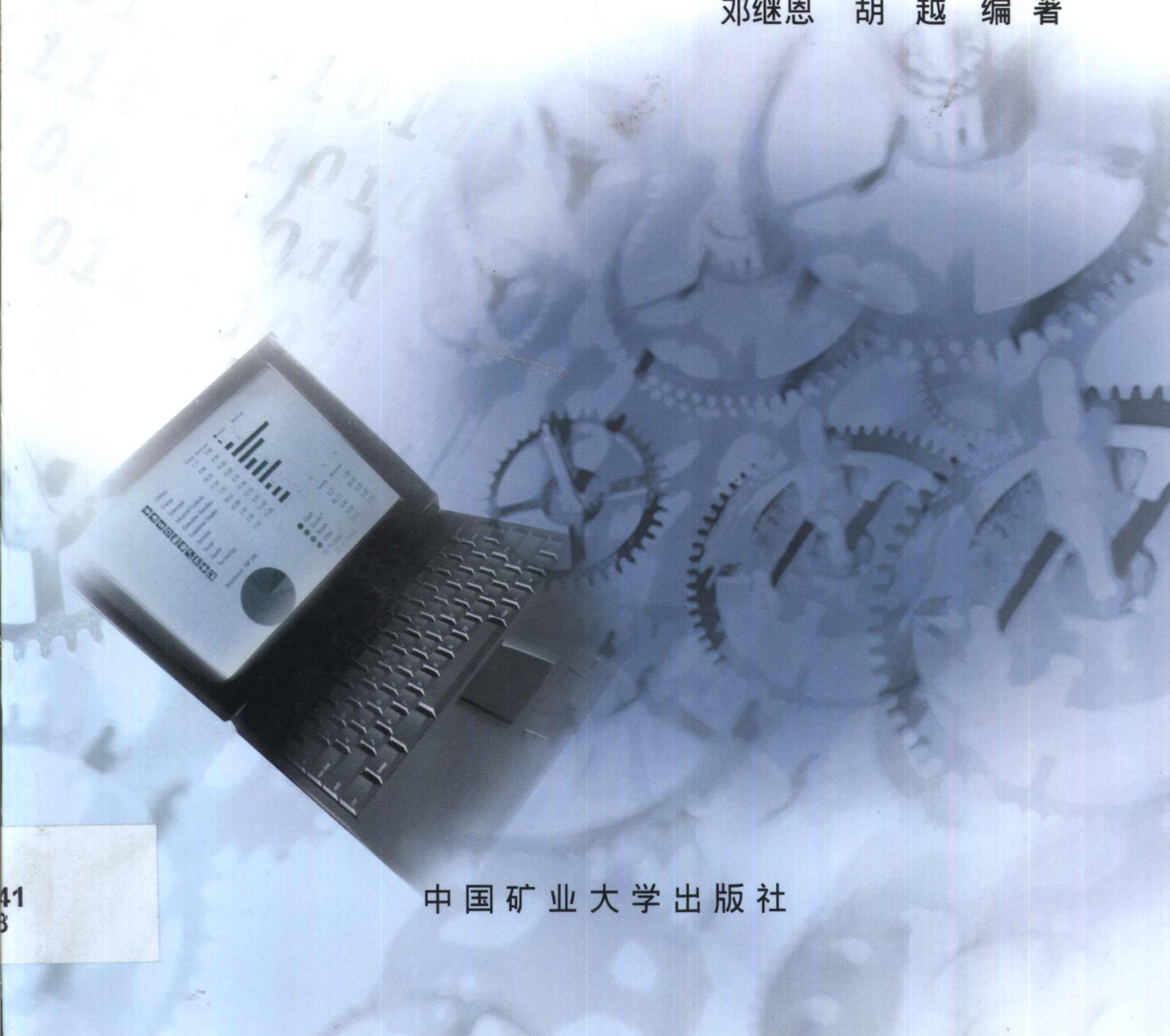
高等学校教学用书

科学与工程计算方法

邓继恩　胡　越　编著



中国矿业大学出版社

高等学校教学用书

科学与工程计算方法

邓继恩　胡　越　编著

中国矿业大学出版社

内 容 提 要

本书详细地介绍了计算机中常用的数值计算方法，主要内容包括：误差分析、解线性代数方程组的消去法和迭代法、插值法、数值积分、非线性方程求解、常微分方程数值解等。本书取材少而精，深浅适度，实用性强，可作为一般理工科院校本科教材和相关专业的研究生教材，也可供广大科技人员及其他人员自学时参考。

责任编辑：刘社育

图书在版编目（CIP）数据

科学与工程计算方法/邓继恩，胡越编著 .—徐州：中国矿业大学出版社，2001.8

ISBN 7-81070-354-4

Ⅰ.科… Ⅱ.①邓… ②胡… Ⅲ.电子计算机-计算方法-高等学校-教材 Ⅳ.TP301.6

中国版本图书馆 CIP 数据核字（2001）第 049330 号

中国矿业大学出版社出版发行

（江苏徐州市 邮政编码 221008）

出版人 解京选

北京科技印刷厂印刷 新华书店经销

开本 787×1092 1/16 印张 9.125 字数 225 千字

2001 年 8 月第 1 版 2001 年 8 月第 1 次印刷

印数 1～3500 册 定价 16.50 元

前　言

随着计算机科学的发展和计算机的广泛应用，科学计算已成为与理论、实验并列的三种科学手段之一。计算的方法已经渗透到自然科学和社会科学的大多数领域之中。它将许多领域的科学研究工作由定性阶段迅速地推向定量阶段。计算机中常用的数值计算方法及有关理论已经成为工程、技术、医学、经济、人文等领域的研究人员所必备的知识。因此，我们在广泛调研的基础上，结合多年的教学和科研经验，在现有的硕士生及本科生讲义的基础上，经过补充、修改、整理而成本书。书中着重介绍最基本、应用最广泛、最有效的数值计算方法。每种方法都配有伪程序，便于编程和上机计算，旨在坚实基础，易学实用。

阅读本书只需具备“高等数学”和“线性代数”的基础知识，因而很适合一般理工科院校的学生和广大科研人员使用。本书另配有“计算方法辅助教学系统”软件一套和上机实验指导书一册。

在本书的编写过程中，中国工程物理研究院北京应用与计算数学研究所研究员苗长兴博士提出了大量的宝贵意见和建议；黎祖庆、张辉、杨世华和梁永权等同志在本书的编排和程序调试中付出了大量辛勤的劳动，在此一并致谢。

由于作者水平有限，书中难免有不妥之处，恳请读者批评指正。

编　者
2001年6月

目　录

第一章 绪 论

第一节 数值计算

自第一台计算机问世以来，计算机在科学技术各个领域中广泛使用所取得的成就无可辩驳地说明，科学方法的第三个分支——计算的方法，已经建立起来了。今天，除了传统的理论方法和实验方法以外，计算的方法已渗透到自然科学和社会科学的大多数领域中，研究工作和工程设计更加离不开计算的方法。它将许多领域的科学研究工作由定性阶段迅速地推向定量阶段。

计算机的高速计算能力和一定的逻辑功能使它成为数学应用于各领域最有力的工具。计算技术和计算数学的发展，使得科学研究和工程实际中非常复杂的问题求数值解，例如从适当的计算结构设计和算法分析到物理、化学、生物、工程乃至社会科学中的许多问题，都可以得到满意的解答。

用计算机解决科学计算问题时经历下面几个过程：实际问题—建立数学模型—数值计算方法—程序设计—上机计算出结果。

利用计算机解决实际问题，通常按以下步骤进行。

一、建立数学模型

数学模型用来描述研究对象的某些性质，通常用数学方程来表示。为方便起见，模型力求简单，但所有有关因素必须以无二义性的方式加以考虑，所以要作相应的假定。例如，在研究天体运行时，假定行星为质点；在研究经济问题时，模型中因素的依赖关系常假定为线性的。为使所确定的模型有意义，描述模型的数学问题必须是适定的。模型的构造及问题的适定性的证明一般可用数学推导方式解决。

二、计算问题的解

对应用来讲，只证明了问题的适定性是远远不够的，必须知道具体的解是什么。例如，在土木工程中经常需要绘制等高线，但若在理论上指出等高线存在而未绘制出来，这显然是不够的。

三、实验验证

在计算结果应用于实践之前，必须验证所得结果是否符合客观规律，这一步是科学实践的一个重要目的。若计算结果与实验不相符，必须重新考察前面两步的工作，适当修正模型、重新计算(甚至改进算法)，直到获得满意的解答为止。

计算数学的任务主要是用计算机计算出数学问题的数值解。从这种意义上说，计算数学是一门辅助性的学科，它是为各具体学科服务的。它的着眼点不在于纯数学的、公理化

的、独立于具体事物的抽象结论，而在于(通常来自于其他领域的)具体问题的数值解。例如，在“线性代数”中只介绍了解线性代数组的存在惟一性及有关理论和精确解法，用这些理论和方法还不能在计算机上解上百个未知数的方程组，更不用说求解十几万个未知数的方程组了。要求这类问题还应根据方程的特点，研究适合计算机使用的、满足精度要求的、节省计算时间的有效算法及其相关的理论。在实现这些算法时往往还要根据计算机容量、字长、速度等指标，研究具体求解步骤和程序设计技巧。有的方法在理论上还不够严格，但通过实际计算、对比分析等手段，证明它是行之有效的方法时，也可采用。

计算数学的任务决定了它的研究内容，这就是寻找求解问题的数值方法，并对它们的数值性质进行研究，这种研究称为数值分析。数值方法的构造、解的可靠性和计算效率是结合在一起的，两者是紧密相联不可分割的。

构造数值方法主要借助于数学推导，因此计算数学同基础数学的关系十分密切。下面的例子给出了如何用数学推导来构造数值方法。

例 1-1　用数学方法推导出求线性方程组

$$\begin{cases}\alpha_{11}x_1+\alpha_{12}x_2=\beta_1\\ \alpha_{21}x_1+\alpha_{22}x_2=\beta_2\end{cases}\tag{1-1}$$

解的方法。

解：将线性方程组

$$\begin{cases}\alpha_{11}x_1+\alpha_{12}x_2=\beta_1\\ \alpha_{21}x_1+\alpha_{22}x_2=\beta_2\end{cases}$$

两个方程分别乘以 α_{22}和 α_{12}，然后相减得

$$(\alpha_{11}\alpha_{22}-\alpha_{12}\alpha_{21})x_1=a_{22}\beta_1-a_{12}\beta_2$$

同理可得

$$(a_{11}a_{22}-a_{12}a_{21})x_2=\alpha_{11}\beta_2-a_{21}\beta_1$$

因此，当 $d=\alpha_{11}\alpha_{22}-\alpha_{12}\alpha_{21}\neq0$ 时，方程组有惟一解

$$\begin{cases}x_1=(\alpha_{22}\beta_1-a_{12}\beta_2)/d\\ x_2=(\alpha_{11}\beta_{11}-\alpha_{21}\beta_2)/d\end{cases}\tag{1-2}$$

当 $d=0$ 时，可能有无穷多解，也可能无解。

上面的例子给出了用数学推导来构造求解数学问题的方法，并在有解的情况下给出解的公式。但并不是一切问题的解都能用一个解析式表示出来。例如，Galois 证明了五次以上的代数方程不可能用公式求解。事实上，在以后各章将会看到，大多数问题的解不可能用公式来表示。例如，计算函数 $f(x)$的积分

$$I=\int_\alpha^\beta f(x)\mathrm{d}x\tag{1-3}$$

只有在极少数情况下才能用 Newton-Leibniz 公式计算，而在一般情况下，取

$$h=(b-a)/n,x_k=a+kh\qquad(k=0,1,\cdots,n)$$

用

$$S=\sum_{k=1}^{n}f(x_k)h\tag{1-4}$$

计算。无论 n 多么大，都只能获得定积分式(1－3)的近似值，用式(1－4)进行计算就是求定积分式(1－3)的一种(近似)数值方法，这类方法称为近似替代法，它是数值计算中常用的方法，常常用在一些有关复杂的函数的计算问题中。

上例说明，如何用数学手段构造求解问题的方法(或公式)，前后所获得的两个方法是不同的，前者是精确的，后者是近似的。

第二节 误 差

提起数值分析，往往会给人以不严格、不准确以至于不够完美的感觉。其实，近似是正常的，误差是不可避免的，根本不存在绝对的严格和精确。

一、误差的来源与误差分析的重要性

用计算机解决科学计算问题首先要建立数学模型。数学模型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。只有实际问题提法正确，建立数学模型时又抽象、简化得合理，才能得到好的结果。由于这种误差难于用数量表示，通常都假定数学模型是合理的，这种误差可忽略不计，在“数值分析”中不予讨论。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，这些参量也包含误差，这种由观测产生的误差称为观测误差，在“数值分析”中也不讨论这种误差。数值分析只研究用数值方法求解数学模型产生的误差。

我们知道，许多数学运算(如微分、积分及无穷级数求和等)是通过极限过程来定义的，然而计算机只能完成有限次的算术运算及逻辑运算，因此需将解题方案加工成算术运算与逻辑运算的有限序列。这种加工常常表现为某种无穷过程的“截断”，由此产生的误差通常称作截断误差。

例如，指数函数 e^x 可展开成下列幂级数形式

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots \tag{1-5}$$

但在实际计算时，我们就无法得到其右端无穷多项的和，而只能截取有限项求出

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

这种计算部分和 $S_n(x)$ 作为 e^x 的值的方法必然会有误差。根据 Taylor 定理，$S_n(x)$ 关于 e^x 的截断误差是

$$R_n(x) = e^x - S_n(x) = \frac{x^{n+1}}{(n+1)!}e^{\theta x} \qquad (0 < \theta < 1) \tag{1-7}$$

例 1－2 假定 $|x| \leqslant 1$，用 Taylor 多项式(1－6)计算 e^x 的值，要求截断误差不超过 0.005，问公式(1－6)该取多少项?

解：利用余项公式(1－7)，注意到 $e^{\theta x} < e < 3$，知

$$|R_n(x)| < \frac{3}{(n+1)!} \tag{1-8}$$

于是，欲使 $|R_n(x)| \leqslant 0.005$，只要取 $n = 5$，即采用下列近似公式

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^2}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{150} \qquad (1-9)$$

进行计算就可以达到要求。

另外，计算当中遇到的数据可能位数很多，甚至会是非常小的数，然而受机器字长的限制，用机器代码表示的数据必须舍入成一定的位数，这又会产生舍入误差。少量的运算的舍入误差一般是微不足道的，但在计算机上完成千百万次运算之后，舍入误差的积累就可能很惊人了。

以上说明，在数值计算的过程中出现各种各样的误差。有时误差会严重“泛滥”，甚至完全“淹没”了所要的真值。因此，进行任何一项计算，首先必须保证精度。尽管数值计算中估计误差比较困难，但仍应重视计算过程中的误差分析，应当在满足精度要求的前提下，挑选和设计好的算法。

二、误差与误差限

设 z^* 为准确值，z 是 z^* 的近似值，称 $e^* = z - z^*$ 为近似值的绝对误差，简称误差。

通常我们不能算出准确值 z^*，也不能算出误差 e^* 的准确值，只能根据测量工具或计算情况估计出误差的绝对值不超过某正数 ε^*，也就是误差绝对值的一个上界。ε^* 叫做近似值的误差限，它总是正数。例如，用毫米刻度的米尺测量一长度 z^*，读出和该长度接近的刻度 z，z 是近似值，z 与 z^* 的误差限是 0.5 mm，于是 $|z - z^*| \leqslant 0.5$；如读出的长度为 765 mm，则有 $|765 - z^*| \leqslant 0.5$，我们从这个不等式仍不知道准确的 z^* 是多少，但知道，$764.5 \leqslant z^* \leqslant 765.5$，说明 z^* 在区间 $[764.5, 765.5]$ 内。

对于一般情形 $|z - z^*| \leqslant \varepsilon$，则

$$z - \varepsilon^* \leqslant z^* \leqslant z + \varepsilon^*$$

这个不等式有时也表示为

$$z^* = z \pm \varepsilon^*$$

三、相对误差与相对误差限

误差限的大小还不能完全表示近似值的好坏。例如两个量 $x^* = 10 \pm 1$，$y^* = 1\,000 \pm 5$，则

$$x = 10, \varepsilon^*(x) = 1; y = 1\,000, \varepsilon^*(y) = 5$$

其中，$\varepsilon^*(x)$ 与 $\varepsilon^*(y)$ 分别为 x 与 y 的误差限。虽然 $\varepsilon^*(x)$ 比 $\varepsilon^*(y)$ 大 4 倍，但 $\varepsilon^*(y)/y = 0.5\ \%$，比 $\varepsilon^*(x)/x = 10\ \%$ 要小得多，这说明 y 近似 y^* 的程度比 x 近似 x^* 的程度要好得多。所以考虑误差时，除考虑误差限的大小外，还应考虑准确值本身的大小，我们把近似值 z 的误差 e^* 与准确值 z^* 的比值

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

称为近似值 z 的相对误差，记作 e_r^*。

在实际计算中，由于真值 z^* 总是不知道的，通常取

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

作为 z 的相对误差。

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 $\varepsilon_r^* = \frac{\varepsilon^*}{|z|}$。

根据定义，上述例子中$\frac{\varepsilon^*(x)}{|x|}=10\%$和$\frac{\varepsilon^*(y)}{|y|}=0.5\%$分别为 x 和 y 的相对误差限，可见 y 近似y^*的程度比 x 近似x^*的程度好。

两个近似数 x_1 与 x_2，其误差限分别为 $\varepsilon^*(x_1)$及 $\varepsilon^*(x_2)$，它们进行代数运算得到的一些误差限分别为

$$\varepsilon^*(x_1\pm x_2)\approx\varepsilon^*(x_1)+\varepsilon^*(x_2)$$

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

$$\varepsilon^*(x_1/x_2)\approx\frac{|x_1|\varepsilon^*(x_2)+|x_2|\varepsilon^*(x_1)}{|x_2|^2}\quad(x_2\neq 0)$$

更有一般情况是，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 Taylor 展开式进行估计。设 $f(x)$是一元函数，x^*的近似值为 x，以 $f(x)$近似代替 $f(x^*)$，其误差界记作 $\varepsilon^*(f(x^*))$，可用 Taylor 展开式得

$$f(x^*)-f(x)=f'(x^*-x)+\frac{f(\xi)}{2}(x^*-x)^2$$

$$(\xi\text{介于}x\text{与}x^*)$$

取绝对值得

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

假定$|f''(x)|$变化不大，可忽略高阶无穷小，于是

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

对于多元函数情况，这里不再赘述。

四、有效数字

对于某个准确值给出一个近似值后，我们当然希望说明它的准确程度。为此再引入有效数字的概念。

如果近似值 z 的误差限是某一位上的半个单位，且该位直到 z 的第一位非零数字一共有 n 位，我们就说近似值有 n 位有效数字。

例如，对于 $z^*=\pi=3.141\,592\,6\cdots$，近似值 $z_1=3.14$ 的误差为 $-0.001\,592\,6\cdots$，可见它有 3 位有效数字；又近似值 $z_2=3.141\,6$ 的误差为 $0.000\,007\,4\cdots$，它有五位有效数字；然而对于近似值 $z_3=3.141\,5$，由于误差为 $-0.000\,092\,6\cdots$，故 z_3 仅有四位有效数字。

z 有 n 位有效数字可写成标准形式

$$z=\pm 10^m\times(a_1+a_2\times 10^{-1}+\cdots+a_n\times 10^{-(n-1)})\qquad(1-10)$$

其中，a_1 是 1 到 9 的一个数字，a_2，$\cdots$，a_n 是 0 到 9 的一个数字，m 为整数，且

$$|z-z^*|\leqslant\frac{1}{2}\times 10^{m-n+1}\qquad(1-11)$$

至于有效数字与相对误差限的关系，这里不加证明给出一个结论。

定理 1－1 用式(1－10)表示的近似数 z，若 z 有 n 位有效数字，则其相对误差限为

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

反之，若 x 的相对误差限为 $\epsilon_r^* \leqslant \frac{1}{2a_1} \times 10^{-(n-1)}$，则 x 至少具有 n 位有效数字。

定理说明，有效位数越多，相对误差限越小。

五、数值运算中误差分析的原则

数值运算中的误差分析是个很重要而复杂的问题。这里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

用绝对值小的数作除数舍入误差会增大。如计算 $\frac{x}{y}$，若 $0<|y|\ll|x|$，则可能对计算结果带来严重影响，应尽量避免。

例 1－3　线性方程组

$$\begin{cases} 0.000\,01x_1 + x_2 = 1 \\ 2x_1 + x_2 = 2 \end{cases}$$

的准确解为

$$x_1 = \frac{200\,000}{399\,999} = 0.500\,001\,25$$

$$x_2 = \frac{199\,998}{199\,999} = 0.999\,995$$

现在在四位浮点十进制数（仿计算机实际计算）下用消去法求解，上述方程写成

$$\begin{cases} 0.100\,0 \times 10^{-4}x_1 + 0.100\,0 \times 10^1 x_2 = 0.100\,0 \times 10^1 \\ 0.200\,0 \times 10^1 x_1 + 0.100\,0 \times 10^1 x_2 = 0.200\,0 \times 10^1 \end{cases}$$

若用 $\frac{1}{2}$（$0.100\,0\times10^{-4}$）除第一方程减第二方程，则出现用小的数除大的数，得到

$$\begin{cases} 0.100\,0 \times 10^{-4}x_1 + 0.100\,0 \times 10^1 x_2 = 0.100\,0 \times 10^1 \\ 0.200\,0 \times 10^6 x_2 = 0.200\,0 \times 10^6 \end{cases}$$

由此解出

$$x_1 = 0, x_2 = 0.100\,0 \times 10^1 = 1$$

显然严重失真。

若反过来用第二个方程消去第一个方程中含 x_1 的项，则避免了大数被小数除，得到

$$\begin{cases} 0.100\,0 \times 10^5 x_2 = 0.100\,0 \times 10^6 \\ 0.200\,0 \times 10^1 x_1 + 0.100\,0 \times 10^1 x_2 = 0.200\,0 \times 10^1 \end{cases}$$

由此求得相当好的近似解

$$x_1 = 0.500\,0, x_2 = 0.100\,0 \times 10^1$$

用小数除大数，有时甚至会出现上溢，它使计算无法进行下去。类似地，用很大的数除小数可能发生下溢。一般情况下，当下溢发生时，计算机令结果为零，不给出任何指示而继续计算，但我们并不能察知这一情况。因此，应控制计算过程的量，使其不产生上溢和下溢。

2. 要避免两相近数相减

在数值运算中两相近数相减有效数字会严重损失。例如 $x=532.65$，$y=532.52$ 都具

有五位有效数字，但 $x-y=0.13$ 只有两位有效数字。这说明必须尽量避免出现这类运算。最好是改变计算方法，防止这种现象产生。

例 1－4 设要对 $a=1\ 000$ 取四位数字计算。

$$x=\sqrt{a+1}-\sqrt{a}$$

开方得到了 $\sqrt{a+1}=31.64$，$\sqrt{a}=31.62$，两者直接相减得 $x=0.02$，这个结果只有一位有效数字（x 的实际值为 0.015 81）。但是，如果先加工，计算公式成为

$$x=\sqrt{a+1}-\sqrt{a}=\frac{1}{\sqrt{a+1}+\sqrt{a}}$$

则按后一公式求得的结果 $x=0.015\ 81$ 有四位有效数字。可见，计算公式写成怎样的形式，对舍入误差的影响可能是很大的。如果无法改变算式，则采用增加有效位数进行计算，在计算机上则采用双倍字长运算，但这要增加计算机计算时间和多占内存单元。

3. 要防止大数“吃掉”小数

在数值运算中参加运算的数有时数量级相差很大，而计算机位数有限，如不注意运算次序就可能出现大数“吃掉”小数的现象，影响计算结果的可靠性。

例 1－5 在五位计算机上计算

$$A=52\ 492+\sum_{i=1}^{1\ 000}\delta_i$$

其中 $0.1<\delta_i<0.9$。把运算的数写成规格化形式

$$A=0.524\ 92\times 10^5+\sum_{i=1}^{1\ 000}\delta_i$$

由于在计算机内计算时要对阶，若取 $\delta_i=0.9$，对阶时 $\delta_i=0.000\ 009\times 10^5$，在五位的计算机中表示为机器 0，因此

$$A=0.524\ 92\times 10^5+0.000\ 009\times 10^5+\cdots+0.000\ 009\times 10^5$$
$$\underline{\underline{\Delta}}\,0.524\ 92\times 10^5(\text{符号}\underline{\underline{\Delta}}\text{ 表示机器中相等})$$

结果显然不可靠，这是由于运算中出现了大数 52 492“吃掉”小数 δ_i 造成的。如果计算时先把数量级相同的一千个 δ_i 相加，最后再加 52 492，就不会出现大数“吃掉”小数现象。这时

$$0.1\times 10^3\leqslant\sum_{i=1}^{1\ 000}\delta_i\leqslant 0.9\times 10^3$$

于是

$$0.001\times 10^5+0.524\ 92\times 10^5\leqslant A\leqslant 0.009\times 10^5+0.524\ 92\times 10^5$$
$$52\ 592\leqslant A\leqslant 53\ 392$$

4. 注意简化计算步骤，减少运算次数

同样一个计算问题，如果能减少运算次数，不但可节省计算机的计算时间，还能减少舍入误差，这是数值计算必须遵从的原则，也是“数值分析”要研究的重要内容。

例如，计算多项式

$$P(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0 \tag{1-12}$$

的值，若直接计算 a_kx^k 再逐项相加，一共需做

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

次乘法和 n 次加法。若将式(1 - 12)改写为

$$p_n(x) = (\cdots((a_n x + a_{n-1})x + a_{n-1})x + \cdots + a_1)x + a_0 \qquad (1-13)$$

采用 Horner 算法

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

只要 n 次乘法和 n 次加法就可算出 $P_n(x)$ 的值。

第三节　问题的性态

在上一节中，我们看到理论计算中用到了实数系对四则运算的封闭性，没有舍入误差。而计算机中的浮点运算不具有这种封闭性，所以就产生了舍入误差。可以运用数值计算中应注意的一些原则，尽可能防止和减少误差的危害，得到稳定的数值方法，获得一个问题的满意解答。但也有一些问题，无论用任何方法都不可能得到一个好的解。因为问题本身对数据中小的误差和计算过程中的舍入误差十分敏感。

例 1 - 6　求多项式 $P(x) = x^2 + x - 1\ 150$ 在 $x = 100/3$ 处的值。

解：

$$P(\frac{100}{3}) = -\frac{50}{9} \approx -5.6$$

而

$$P(33) = -28$$

数据变化仅 1 %，结果的变化却有 400 %。

这种数据相对小的变化引起解相对大的变化的问题称为病态问题；否则称为良态问题。例 1 - 4 的问题就是良态问题。

设一个问题的已知数据只有一个，用 x 表示，结果可以看成是 x 的函数 $f(x)$。若有两个不同的数据 x 和 $\bar{x}$，则可得到两个不同的结果 $f(x)$ 和 $f(\bar{x})$。设 $x \neq 0$ 时，$f(x) \neq 0$，那么，$\delta = |x - \bar{x}| / |x|$ 和 $R = |f(x) - f(\bar{x})| / |f(\bar{x})|$ 分别表示数据和结果的相对误差。若能找到一个数 m，满足

$$\frac{|f(x) - f(\bar{x})|}{|f(x)|} \leqslant m \frac{|x - \bar{x}|}{|x|}$$

则 m 可以解释为结果的相对误差对数据的相对误差的一个放大倍数，称这个数 m 为该问题的条件数，以符号 Cond(f)记之。

若函数 $f(x)$ 在包含 x 和 $\bar{x}$ 的某个区间上连续可微，则由微分中值定理可以得到

$$f(x) - f(\bar{x}) = f'(\xi)(x - \bar{x})$$

这里，ξ 是 x 和 $\bar{x}$ 之间的某个值。当 $|x - \bar{x}|$ 足够小时，由 $f'(x)$ 的连续性，知

$$f'(\xi) \approx f'(x)$$

于是

$$f(x) - f(\bar{x}) \approx f'(x)(x - \bar{x})$$

从而

$$\frac{f(x) - f(\bar{x})}{f(x)} \approx x\left[\frac{f'(x)}{f(x)}\right]\frac{x - \bar{x}}{x}$$

故

$$\frac{|f(x)-f(\bar{x})|}{|f(x)|} \leqslant \left| x\frac{f'(x)}{f(x)} \right| \frac{|x-\bar{x}|}{|x|}$$

这表明，可以近似地取

$$\mathrm{Cond}(f) \approx \left| x\frac{f'(x)}{f(x)} \right| \tag{1-15}$$

作为 $f(x)$ 求值问题的条件数。

将式(1－15)应用于例 1－6 中的求值问题，得 $f(x)$在 $x=100/3$ 的条件数为

$$\mathrm{Cond}(p) = 406$$

这说明，在 $x=100/3$ 处求值 $p(x)$，数据的相对误差要引起结果产生约 400 倍的相对误差。这样的问题是病态的。

将式(1－15) 应用于例 1－4 中的求 $f(a)=\sqrt{a+1}-\sqrt{a}$ 值问题，得条件数为

$$\mathrm{Cond}(f) \approx 0.5$$

在 $a=1\,000$ 处不大，故这是一个良态问题，说明用好的方法可以得到满意的解。

综上所述，在用计算机解一个数学问题时，首先要考虑问题本身性态，然后要构造(或选择)一个稳定的(好的)方法。只有用稳定的方法去解良态问题，才能获得满意的结果。更详细了解关于线性方程组解的稳定性和常微分方程解的稳定性，可参看其他文献。

第四节 数值算法及其描述

一、算法

从事任何一项计算，事先总要拟订计算方案和规划计算步骤。不过人工手算时，解题步骤不一定要书写成文字的形式，可以留在人们的脑子里酝酿。在解题过程中，人们常常随时处理各种“意外”的情况，充实或修改预定的方案。

计算机的一个重要特点是运算的高速度，这是人工手算无法比拟的。要利用计算机处理问题，需要编写出使计算机按人们意愿工作的计算机程序。我们知道，计算机只能机械地执行人的指示和命令，因此，交给机器执行的解题方案中的每个细节都必须准确地加以定义，并且全部解题过程应当完整地描述出来。本课程的一个主要目的是将各种问题的计算方法以算法的形式表示出来，也就是将所构造出的方法变成算法。

算法是由有限个无二义性的法则组成的一个计算过程，这些法则明确规定了一串运算，以产生一个问题或者一类问题的解。

一个问题的算法可能有多个，虽说计算机的功能很强，但是若算法选择不当，计算机的利用效率就得不到充分发挥，所以不能忽略对算法的研究。判定算法优劣的标准一般有三个：计算量大小；占内存多少；逻辑结构是否简单。

按算法的定义，一个数学公式不是一个算法。因为数学公式可以利用运算定律(如交换律、结合律等) 按不同的顺序计算，一个数学公式也不可能说明当除数为零时应如何处理等，因此，为描述算法只有数学公式是不够的。本课程中，描述算法主要有两个目的：一是帮助读者理解并掌握按某种方法计算的全过程；二是帮助读者将所学的方法在计算机上实现。为此，需要一种语言。通常的程序设计语言，如 Fortran，Basic，C 语言等，是

描述算法的精确语言，可直接在计算机上使用，但这类语言太详细，反而使方法的计算思路不易表示出来，因而不宜采用。我们用一种比较易懂的非形式语言表示算法。这种语言就是伪代码。伪代码是介于自然语言和计算机语言之间的用文字和符号来描述算法的一种语言。它如同一篇文章一样，自上而下地写下来，每一行(或几行)表示一个基本操作。它不用图形符号，因此书写方便、格式紧凑、易懂，也便于向计算机语言算法(即程序)过渡。

例如，用伪代码描述求解方程组(1-1)的算法如下：

算法 LES(A，b，X)

[本算法用于求二阶线性方程组 $Ax=b$ 的解系数矩阵 A 存在一个 2×2 数组中，右端向量单独放在数组 b 中。]

(1)$\alpha_{12}\alpha_{22}-\alpha_{12}\alpha_{21}\Rightarrow d$；

(2)If $d=0$ then 给出信息“无解或无穷多解”;stop;

(3)$(\alpha_{22}\beta_1-\alpha_{12}\beta_2)/d\Rightarrow x_1$；

(4)$(\alpha_{11}\beta_2-\alpha_{21}\beta_1)/d\Rightarrow x_2$。

上面方括号中给出算法的一个简要说明，其他符号的意义不难理解。另外，按式(1-3)的 Horner 算法给出计算式(1-12)多项式的值的算法。

算法 Horner(A，n，x，p)

[本算法求多项式(1-12)在给定 x 处的值，其系数 a_i 存放于数组 A 中]

(1)$a_n\Rightarrow p$；

(2)For $k=n-1,n-2,\cdots,1,0$;

(3)$xp+a_n\Rightarrow p$。

习 题

1-1 试改造下列各式，使能比较准确地计算出 y 的值。

(1) 当 $|x|<1$ 时，$y=\dfrac{1}{1+2x}-\dfrac{1-x}{1+x}$；

(2) 当 $|x|>1$ 时，$y=\sqrt{x+\dfrac{1}{x}}-\sqrt{x-\dfrac{1}{x}}$；

(3) 当 $|x|\ll 1$ 时，$y=1-\cos 2x$；

(4) 当 $p>0,q>0$ 且 $p\gg q$ 时，$y=-p+\sqrt{p^2+q^2}$。

1-2 设 $s=\dfrac{1}{2}gt^2$，假定 g 是准确的，而对 t 的测量有 ±0.1 s 的误差，证明当 t 增加时 s 的绝对误差增加，而相对误差却减少。

1-3 试用消元法解方程组

$$\begin{cases}x_1+10^{10}x_2=10^{10}\\x_1+x_2=2\end{cases}$$

假定只用三位有效数字计算，问结果是否可靠？

1-4 表达式 $(\sqrt{2}-1)^6$ 与下列诸式是等价的

$$\frac{1}{(\sqrt{2}+1)^6}\qquad(3-2\sqrt{2})^3\qquad\frac{1}{(3+2\sqrt{2})^3}\qquad 99-70\sqrt{2}$$

取$\sqrt{2}\approx1.4$，利用上述各等式计算，哪一个得到的结果最好？

1－5 已知三角形面积为 $S=\frac{1}{2}ab\sin\theta$，其中 θ 为弧度，且测量 a，b，θ 的误差分别为Δa、Δb、$\Delta\theta$，证明面积的误差 ΔS 满足

$$\left|\frac{\Delta S}{S}\right|\leqslant\left|\frac{\Delta a}{a}\right|+\left|\frac{\Delta b}{b}\right|+\left|\frac{\Delta\theta}{\theta}\right|$$

1－6 正方形的边长约为 100 cm，问测量时误差应多大，才能保证面积误差不超过 1 cm^2？