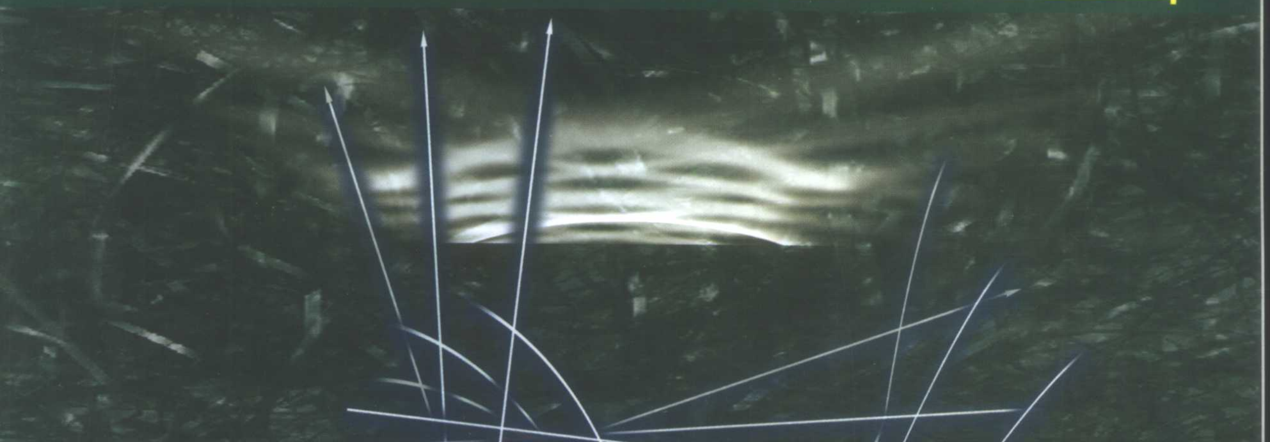
新世纪本科生系列教材

数值计算方法

李乃成　邓建中



西安交通大学出版社

新世纪本科生系列教材

数值计算方法

李乃成　　邓建中

西安交通大学出版社

内容简介

本书是根据国家教委工科数学课程指导委员会制定的《数值计算方法课程教学基本要求》编写的教材。内容包括计算机上常用的几种数值计算方法，如插值法、最小二乘法、数值微积分、方程求根法、线性方程组的直接解法与迭代解法、常微分方程初值问题的数值解法。书中还包含矩阵特征值和特征向量求法、非线性方程组解法、迭代序列加速收敛法、以及同类书中未见到的一些内容，如广义皮亚诺定理、逐次插值求根法等。这些内容可供要求更高的专业选用。

本书源于邓建中等编写的研究生教材《计算方法》，该书曾获国家教委优秀教材二等奖。本书保留了原书特色，又针对本专科学生的需要作了增删修改。

本书适于作理工科专业本专科生 32～54 学时课程的教材，也可供研究生和科技工作者参考。

图书在版编目(CIP)数据

数值计算方法/李乃成，邓建中编．—西安：西安交通大学出版社，2002.11
ISBN 7-5605-1577-0

Ⅰ．数...　Ⅱ．①李...②邓...　Ⅲ．数值计算—计算方法—高等学校—教材　Ⅳ．O241

中国版本图书馆 CIP 数据核字(2002)第 072584 号

*

西安交通大学出版社出版发行
(西安市兴庆南路 25 号　邮政编码：710049　电话：(029)2668315)
西安东江印务有限公司印装
各地新华书店经销

*

开本：727 mm×960 mm　1/16　印张：10.75　字数：196 千字
2002 年 11 月第 1 版　2002 年 11 月第 1 次印刷
印数：0 001－4 000　定价 15.00 元

发行科电话：(029)2668357，2667874

前 言

在现代科学研究与工程设计中，计算机已成为不可缺少的有力工具。科教兴国，要求现代理工科大学生，掌握计算机常用数值计算方法，具有使用计算机解决实际问题的能力。因此国家教委工科数学课程指导委员会建议工科各专业把数值计算方法列为本科生必修课，并制定了教学基本要求(见附录)。本书是按这要求编写的教材，适合 32～36 学时教学需要。附加少量用 * 标识的章节，可用于有更高要求的各专业。

本书以邓建中、葛仁杰、程正兴合编的《计算方法》为基础。该书由西安交通大学出版社于 1985 年出版，1988 年获国家教委优秀教材二等奖，主要供研究生使用。该书也在本专科学生中使用，但作为代用教材，多有不便。本书为适应本专科学生需要，对该书作了增删修改。删去部分，包括差分法、有限元法两章和矩阵求逆、B 样条、正交多项式、最优一致逼近等节。增添部分，一是补足主要算法的数值例题，二是添加典型方法的过程描述，三是增配上机实习题，目的是加强培养应用能力。修改部分，主要是降起点，补基础，追直观，求简化，使读者便于理解。例如矩阵三角分解的推导，原书和其他教材均采用消列变换，本书则改用线性代数必学的初等行变换，使学生易懂并减少篇幅。又如因式定理，书中多处应用，但一般高等数学教材却不提及，本书便作了完整叙述和证明。

本书保留了原书的体系，保证了内容的循序渐进。本书注意保持和发扬原书的优点：重点突出，强调算法的构造和应用；推导详细，力求简单、统一，统一于近似替代、待定系数法和问题转换三种基本手段；讲述由浅入深、由简到繁，贴近学生水平；每章开头有概述，末尾有小结，各节有复习题，书末还附基本要求，便于学生抓住要领；配备例题、习题较多，利于学生理解和应用。

本书还包含了同类书中未见的一些内容，如广义皮亚诺定理、逐次插值求根法、收敛阶数确定法、插值加速收敛法等。这些内容来自作者的科研成果，有助于简化推导、提高计算效率。其中广义皮亚诺定理，在 1990 年工科研究生数值分析课程教学研讨会上，已被会议审定列入研究生课程基本要求。

本书还注意反映了中国古代数学家对计算数学的贡献。

由于水平有限，缺点与错误在所难免，恳请专家和读者批评指正。

编　者

2002.8.10

目 录

第1章

数值计算方法的一般概念

本章是本书的绪论，介绍两个基本概念：算法与误差，使读者大体了解本课程的任务和学习方法，掌握设计算法的一般原则与注意事项。

1.1 算法

随着计算机的发展和普及，数值计算已成为工程设计与科学研究的重要手段。掌握数值计算方法，会用计算机解决科学与工程实际提出的数值计算问题，已成为科技人员必须具有的能力。本书的编写，就是为了适应理工科院校本专科普遍开设数值计算方法课程的需要。

所谓数值计算方法，又称**计算数学**或**数值数学**，就是研究怎样利用算盘、算尺、计算器、计算机等工具，来求出数学问题数值解答的学问。本书介绍计算机上常用的数值计算方法。

计算机实质上只会做加减乘除等算术运算和一些逻辑运算。由这些基本运算及运算顺序的规定构成的完整的解题步骤，称 ε 为**算法**。它可用框图、算法语言、数学语言或自然语言描述。用计算机算法语言描述的算法称为计算机程序。数值计算方法的任务，便是为各种数学问题的数值解答，提供最有效的算法。

最有效的算法，应当适用范围广，运算工作量少，需用存储单元少，逻辑结构简单，便于编写计算机程序，而且计算结果可靠。由于计算机计算通常是近似的，因而一般要求算法能估计误差；计算过程中误差能得到的控制，各步误差对计算结果不致产生过大影响（即具**稳定性**）；通过增加计算量，能使近似计算解充分接近理论解（即具**收敛性**）。

遗憾的是没有一种算法处处都最有效；而且各种算法在计算过程中往往都会出现某些问题。因此，为用计算机解决实际问题，除正确提出数学问题（数学

模型)外,必须针对具体问题选择或改造适当的算法,清楚计算过程中可能发生的问题。这就要求读者了解算法的设计原理,充分熟悉算法的运算过程,知道算法的长处和短处。为熟悉算法的运算过程,需要通过手算和上机实习,解决几个具体问题。分析算法的优劣,特别是算法的稳定性、收敛性和误差估计,需要较多的数学推导,一般专业的大学生不必深究,但需知道结论。

复习题

1. 何谓算法?评判算法优劣的标准有哪些?
2. 本课程的主要任务是什么?重点在哪里?

1.2 误　差

1.2.1 误差的来源与分类

科学与工程计算中的数通常都是近似的。近似值与真正值之差称为误差。误差的来源或分类有四种。从实际问题提炼出数学问题时往往忽略了许多次要因素,因而即使数学问题能求出准确解,也与实际问题的真正解不同。它们之差称为**模型误差**。一般数学问题包含若干参量,它们的值需要通过观测得到,难免有误差。这种误差称为**数据误差**或**观测误差**、**参量误差**。一般数学问题难以准确求解,往往要通过近似替代,简化为较易求解的问题后再求解。这样引起的误差称为**截断误差**或**方法误差**。计算机只能对有限位的数进行运算,一般数必须进行舍入。此时产生的误差称为**计算误差**或**舍入误差**。计算结果的误差是四种误差累积影响的误差。本课程不讨论数学模型的建立,所以只讨论后三种误差的影响。

1.2.2 误差与准确数字

设 $\tilde{x}$ 是真值 x 的近似值,则两者之差或其绝对值

$$\Delta x = x - \tilde{x} \text{ 或 } |\Delta x|$$

称为近似值 $\tilde{x}$ 的**绝对误差**。真值 x 一般无法知道,绝对误差自然也无法知道。通常用绝对误差不超过的数 ε,即满足 $|\Delta x|\leqslant\varepsilon$ 的数 ε,来表示绝对误差的大小,称 ε 为**绝对误差界**或**绝对误差限**,并记为

$$x = \tilde{x} \pm \varepsilon \text{ 或 } \tilde{x}(\pm \varepsilon)$$

绝对误差或绝对误差界常常简称误差。

绝对值很大或很小的近似值 $\tilde{x}$,不宜用绝对误差而应当用相对误差表示其

近似程度。$\delta x = \Delta x / x$ 或 $\Delta x / \tilde{x}$ 或 $|\delta x|$ 称为 $\tilde{x}$ 的**相对误差**。$|\delta x| \leqslant \varepsilon_r$ 时 ε_r 称为 $\tilde{x}$ 的**相对误差界**或**相对误差限**，记为

$$x = \tilde{x}(1 \pm \varepsilon_r)$$

相对误差界常用百分数表示，有时也称为相对误差。

近似值的准确数字与误差界密切相关。设 $x_1 > 0$,

$$\tilde{x} = \pm x_1 x_2 \cdots x_m x_{m+1} \cdots x_{m+n} \cdots$$

若

$$|x - \tilde{x}| \leqslant 0.\underbrace{00\cdots0}_{n\text{个}0}5 = \frac{1}{2} \times 10^{-n}$$

则称 $\tilde{x}$ 准确到 n 位小数，并称 x_{m+n} 及其以前非零数字为**准确数字**。例如 $\pi =$ 3.141 592 653 589 793 238 46…的近似值 $\pi_1 = 3.141\ 6$, $\pi_2 = \frac{22}{7} = 3.142\ 8\cdots$, $\pi_3 = \frac{355}{113} = 3.141\ 592\ 9\cdots$，由于

$$|\pi - \pi_1| = 0.000\ 007\ 3\cdots < 0.000\ 05$$

$$|\pi - \pi_2| = 0.001\ 2 < 0.005$$

$$|\pi - \pi_3| = 0.000\ 000\ 26\cdots < 0.000\ 000\ 5$$

便知 π_1 准确到 4 位小数，具有 5 位准确数字；π_2 准确到 2 位小数，具有 3 位准确数字；π_3 准确到 6 位小数，具有 7 位准确数字。一般，设 $x_1 > 0$,

$$\tilde{x} = \pm x_1 x_2 \cdots x_m x_{m+1} \cdots x_{m+n} = \pm 10^m \times 0.x_1 x_2 \cdots$$

或

$$\tilde{x} = \pm 0.\underbrace{0\cdots0}_{m\text{个}} x_1 x_2 \cdots x_{n-m} = \pm 10^{-m} \times 0.x_1 x_2 \cdots$$

如果绝对误差界

$$|x - \tilde{x}| \leqslant \frac{1}{2} \times 10^{-n}$$

则 $\tilde{x}$ 准确到 n 位小数，具有 $n \pm m$ 位准确数字。

像 π_1 那样各位数字皆准确的近似数称为**有效数**。此时各准确数字也称为**有效数字**。从准确值按四舍五入原则截取得到的近似数都是有效数。计算机算出的近似值一般不是有效数，只具有若干准确数字。

近似值 $\tilde{x} = \pm 10^m \times 0.x_1 x_2 \cdots$ 时，$x_1 > 0$，如果知道相对误差界是 $\frac{1}{2} \times 10^{-t}$，则绝对误差

$$|x - \tilde{x}| = |\tilde{x}|\,|\delta x| \leqslant 10^m \times 0.x_1 x_2 \cdots \times \frac{1}{2} \times 10^{-t} < \frac{1}{2} \times 10^{-(t-m)}$$

即绝对误差界不超过 $\frac{1}{2} \times 10^{-(t-m)}$，可见 $\tilde{x}$ 至少具有 $(t-m)+m = t$ 位准确数字。

总之，近似值 $\tilde{x}$ 的绝对误差界为 0.5×10^{-t} 时，$\tilde{x}$ 准确到 t 位小数；$\tilde{x}$ 的相

对误差界为 0.5×10^{-t} 时，$\bar{x}$ 至少具 t 位准确数字。

1.2.3 数据误差影响的估计

任何数学问题的解 y 总与某些参量 $x_1,x_2,\cdots,x_n$ 有关：

$$y=\varphi(x_1,x_2,\cdots,x_n)$$

参量的值(即数据)若有误差，解也一定有误差。设 $x_1,x_2,\cdots,x_n$ 的近似值为 $\bar{x}_1,\bar{x}_2,\cdots\bar{x}_n$，相应解为 $\bar{y}$，则近似解 $\bar{y}$ 的绝对误差

$$\Delta y=y-\bar{y}=\varphi(x_1,x_2,\cdots,x_n)-\varphi(\bar{x}_1,\bar{x}_2,\cdots,\bar{x}_n)$$

相对误差

$$\delta y=\Delta y/y=\Delta y/\varphi(x_1,x_2,\cdots,x_n)$$

当数据误差较小时，由于函数增量近似等于其微分，得误差估计式

$$\Delta y\approx\sum_{i=1}^{n}\frac{\partial\varphi(x_1,x_2,\cdots,x_n)}{\partial x_i}\Delta x_i \tag{1-1}$$

$$\delta y\approx\sum_{i=1}^{n}\frac{\partial\varphi(x_1,x_2,\cdots,x_n)}{\partial x_i}\frac{x_i}{\varphi}\delta x_i \tag{1-2}$$

在这两个公式中，系数绝对值

$$\left|\frac{\partial\varphi}{\partial x_i}\right| \text{或} \left|\frac{x_i}{\varphi}\frac{\partial\varphi}{\partial x_i}\right| \tag{1-3}$$

表示解的误差相对参量 x_i 的误差的放大或缩小“倍数”。它们的值如果很大，则 $\bar{x}_i$ 的小误差可能导致解 $\bar{y}$ 很大的误差。因此，这些系数绝对值称为求 y 问题的**条件数**，其值很大的问题称为**坏条件问题**或**病态问题**。对病态问题最好采用高精度计算，避免数据误差给解带来很大影响。

$y=\varphi(x_1,x_2,\cdots,x_n)\approx0$ 时相对误差条件数 $\left|\frac{x_i}{\varphi}\frac{\partial\varphi}{\partial x_i}\right|$ 一般很大。因此，凡计算结果按近于零的问题往往都是病态问题。例如相近的数相减，因差接近于零，就是病态问题。设计算法时应尽量避免相近数相减。例如 $x\gg1$ 时，为计算

$$y=\ln\left(x+\frac{1}{x}\right)-\ln\left(x-\frac{1}{x}\right),\qquad z=\arctan(x+1)-\arctan x$$

最好采用公式

$$y=\ln\frac{x^2+1}{x^2-1},\qquad z=\arctan\frac{1}{1+x(x+1)}$$

又如 $b^2\gg4ac$ 时，为解二次方程 $ax^2+bx+c=0$，不宜采用公式

$$x_1=(-b+\sqrt{b^2-4ac})/2a,\qquad x_2=(-b-\sqrt{b^2-4ac})/2a$$

最好采用公式

$$x_1=[-b-\mathrm{sgn}(b)\sqrt{b^2-4ac}]/2a,\qquad x_2=c/ax_1$$

其中 $\mathrm{sgn}(x)$ 是 x 的符号函数，即 $b>0$ 时 $\mathrm{sgn}(b)=1$，$b<0$ 时 $\mathrm{sgn}(b)=-1$，$\mathrm{sgn}(0)=0$。

由误差估计式(1-1)易知

$$\Delta(x_1x_2)\approx x_2\Delta x_1+x_1\Delta x_2$$

$$\Delta\left(\frac{x_1}{x_2}\right)\approx\frac{\Delta x_1}{x_2}-\frac{x_1}{x_2^2}\Delta x_2$$

由此可见，前式中 $|x_1|$ 很大或后式中 $x_2\approx 0$ 时，计算结果的绝对误差可能很大。因此，实际计算中应当尽量避免用绝对值很大的数作乘数或用接近于零的数作除数。简言之，避免小除数和大乘数。

1.2.4 机器数与舍入误差

在计算机中，一般实数 x 均按舍入原则表示成

$$\mathrm{fl}(x)=\bar{x}=\pm b^m(0.x_1x_2\cdots x_t)$$

并称为 b 进制**浮点数**。正整数 b 称为基数，一般取为 2；但为照顾习惯和书写方便，通常将 2 进制数化为 10 进制数输入或输出。整数 m 称为阶码。小数

$$0.x_1x_2\cdots x_t=x_1b^{-1}+x_2b^{-2}+\cdots+x_tb^{-t}$$

称为尾数或数码，其中 $x_1,x_2,\cdots,x_t$ 只能是 $0,1,\cdots,b-1$ 中数字。$x_1>0$ 时 $\mathrm{fl}(x)$ 称为规格化的浮点数。一定型号的计算机，尾数的位数 t 是固定的，称为计算机的位数。阶码 m 也有一定的取值范围：$L\leqslant m\leqslant U$。因此，计算机存储或运算的数 $|x|$ 不能过大，否则计算机无法表示，只好"溢出"停止计算；$|x|$ 也不能过小，否则计算机自动令 $\mathrm{fl}(x)=0$，得出难以预料的结果。

b,t,L,U 固定的浮点数只能准确表示有限个数，其余实数都只能按舍入原则近似表示。规格化的浮点数 $\mathrm{fl}(x)$，其末位数字 x_t 可能有半位误差，即绝对误差

$$|x-\mathrm{fl}(x)|\leqslant\frac{1}{2}b^{-t}\times b^m=\frac{1}{2}b^{m-t}$$

相对误差

$$|x-\mathrm{fl}(x)|/|x|\leqslant\frac{1}{2}b^{m-t}/b^m(b^{-1})=\frac{1}{2}b^{-(t-1)}$$

其误差限 $\frac{1}{2}b^{-(t-1)}$ 只同计算机基数 b 与位数 t 有关，称为计算机的相对精度。

在计算机中，规格化的浮点数运算之后仍化为规格化的浮点数存储。因此计算机的每次运算都可能产生舍入误差。例如在有双倍位运算器的 4 位十进制计算机上两数相加：

$$10^4\times 0.199\,5+10^{-1}\times 0.427\,0=1\,995+0.042\,7$$

$$= 1\,995.042\,7 = 10^4 \times 0.199\,5$$

结果小数被大数“吃掉”,小数对和数不起作用,和数自然产生误差。

为减少舍入误差,也为节省计算机时间,实际计算时应当设法减少运算次数。例如要算 2/3,在 4 位机上最好直接计算

$$2.000/3.000 = 0.666\,7$$

而不宜先算 1/3

$$2 \times (1.000/3.000) = 2.000 \times 0.333\,3 = 0.666\,6$$

又如要算多项式的值

$$0.062\,5x^4 + 0.425\,0x^3 + 1.215x^2 + 1.192x + 2.129$$

宜改用下式

$$(((0.062\,5x + 0.425\,0)x + 1.215)x + 1.192)x + 2.129$$

这种算法称为**秦九韶算法**①。它只需 4 次乘法和 4 次加法,而按前式需 10 次乘法和 4 次加法。一般,为计算多项式

$$p(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_{n-1}x + a_n$$

宜改写为

$$p(x) = (\cdots(((a_0)x + a_1)x + a_2)x + \cdots + a_{n-1})x + a_n$$

记第 k 个内括号的值为 b_k,则秦九韶算法为:

令 $b_0 \Leftarrow a_0$
对 $k = 1 \sim n$
令 $b_k \Leftarrow b_{k-1}x + a_k$

计算结果 $p(x) = b_n$。

为减少舍入误差,实际计算时还需注意运算顺序,避免大数吃掉小数。例如在 4 位计算机上计算 3 个数的和:31.97+2.456+0.135 2,先加前二数得

$$(31.97 + 2.456) + 0.135\,2 = 34.43 + 0.135\,2 = 34.57$$

先加后二数得

$$31.97 + (2.456 + 0.135\,2) = 31.97 + 2.591 = 34.56$$

由于真正和是 34.561 2,显然后者比较准确。一般来说,若干数相加,最好先加绝对值较小的数。

1.2.5 算法的稳定性

计算机计算过程中,原始数据可能有误差,每次运算又可能产生舍入误差,

---

① 秦九韶(约 1202~1261),南宋四川人,著有《数书九章》,对同余式、高次方程和线性方程组的求解有独创性研究,国外称秦九韶算法为 Horner 算法,其实 Ruffini 与 Horner 分别于 1804、1819 年提出,比秦九韶晚了 500 多年。

误差积累起来，很可能淹没真正解，使计算结果根本不可靠。例如要在4位计算机上计算8个积分。

$$I_i = e^{-1}\int_0^1 x^i e^x dx, \qquad i = 0,1,\cdots,7$$

利用分部积分法可得递推关系 $I_i = 1 - iI_{i-1}$。注意 $I_0 = 1 - e^{-1}$ 及 $I_i = \int_0^1 x^i e^{-(1-x)}dx < \int_0^1 x^i dx = \frac{1}{i+1} \to 0$（当 $i \to \infty$ 时），可得两种算法：

（Ⅰ）令 $I_0 = 0.632\,1 \approx 1 - e^{-1}$，再算 $I_i = 1 - iI_{i-1}$，　$i = 1,2,\cdots,7$.

（Ⅱ）令 $I_{11} = 0$，再算 $I_{i-1} = (1 - I_i)/i$，$i = 11,10,\cdots,1$. 按算法（Ⅰ）得8个积分的近似值为

0.632 1，　0.367 9，　0.264 2，　0.207 4

0.170 4，　0.148 0，　0.112 0，　0.216 0

按算法（Ⅱ）则得

0.632 1，　0.367 9，　0.264 2，　0.207 3

0.170 9，　0.145 5，　0.126 8，　0.112 4

后者均准确到4位小数，可见算法（Ⅰ）最后结果没有一位数字准确。

可靠的算法，各步误差不应对计算结果产生过大影响，即具有稳定性。研究算法是否稳定，理应考察每步误差的影响。但这相当繁琐。为简单起见，通常只考虑一步（如运算开始时）误差的影响。这实质上是把算法稳定性的研究，转化为初始数据误差对算法影响的分析。可以设想，一步误差影响大，多步误差影响更大；一步误差影响逐步削弱，多步误差影响也削弱。因此简化研究得出的结论具有指导意义。

现用此法分析上述两种算法的稳定性。对算法（I），设 $\tilde{I}_0 \approx I_0$ 有误差，此后计算无误差，则

$$\begin{cases} I_i = 1 - iI_{i-1}, \\ \tilde{I}_i = 1 - i\tilde{I}_{i-1} \end{cases} \qquad i = 1,2,\cdots,7$$

两式相减得　$I_i - \tilde{I}_i = (-i)(I_{i-1} - \tilde{I}_{i-1})$，　$i = 1,2,\cdots,7$

由此递推，得

$$I_7 - \tilde{I}_7 = -7!(I_0 - \tilde{I}_0) = -5\,040(I_0 - \tilde{I}_0)$$

同理，对算法（Ⅱ）可得

$$I_0 - \tilde{I}_0 = -(I_7 - \tilde{I}_7)/5\,040$$

由此可见，按算法（Ⅰ），最后结果误差是初始误差的5 040倍；按算法（Ⅱ），最后结果误差是初始误差的5 040分之一。这说明算法（Ⅱ）的稳定性较好。

在计算机上应当采用稳定性好的算法，尤其是对病态问题。

复习题

1. 何谓绝对误差、相对误差、准确数字？它们关系如何？
2. 何谓病态问题？怎样判断？
3. 计算机中数怎样表示？为减少计算误差，应采取哪些措施？

小　结

本章介绍了算法、误差及相关概念，说明了本课程的基本任务，减少计算误差的措施。

本书各节安排了复习题，目的是帮助读者掌握主要内容，建议读者认真思考、总结。例如减少计算误差的措施，可总结为：减少运算次数，避免相近数相减，避免小除数、大乘数和“溢出”，注意运算顺序（如若干数相加时先加绝对值较小的数，避免大数吃小数），防止误差影响扩大（采用稳定算法）。

本书各章安排了习题和上机实习题，目的是帮助读者熟悉算法过程，理解基本概念，掌握数值算法设计与分析的常用技巧。建议读者选作。

习　题

1-1　已知$\sqrt{2}=1.414\ 213\ 562\ 373\cdots$，写出它的3～6位近似有效数；它的近似分数

$$17/12 = 1.416\ 666\ 6\cdots, \qquad 41/29 = 1.413\ 793\ 1\cdots$$
$$239/169 = 1.414\ 201\ 1\cdots, \qquad 577/408 = 1.414\ 215\ 68\cdots$$

各有几位数字准确？利用四舍五入原则，写出这些分数相应位数的近似有效数。它们是否$\sqrt{2}$的近似有效数？

1-2　求下列近似有效数的绝对误差界、相对误差界及有效数字位数：
(1) 3 450；(2) 0.005 67；(3) $0.234\ 0\times10^5$；(4) $4\ 567\times10^{-3}$；
(5) 123.45×10^1。

1-3　求$\sqrt{20}$的近似有效数，(1)使绝对误差不超过0.01；(2) 使相对误差不超过0.01。

1-4　测得某地纬度 $\varphi=45°56'23''$，误差不超过半秒。问计算 $\sin\varphi$ 的误差界是多少？算出 $\sin\varphi=0.718\ 608\ 590$ 有几位数字准确？

1-5　正方形的边长约为10 cm，问量测边长的误差界多大，才能保证面积误差不超过0.1 cm^2。

1-6 证明下述命题：

(1) 设函数 $y=f(x)$ 可导，$f(\alpha)=0$，则在 $x\approx\alpha$ 时计算 $f(x)$ 的值，按相对误差是病态问题。

*(2) 设 α 是方程 $P(x)=a_0x^n+a_1x^{n-1}+\cdots+a_n=0$ 的根，则当 α 接近重根时，即 $p'(\alpha)\approx 0$ 时，按照绝对误差，求根 α 的问题是病态问题。

1-7 已知 $|x|\ll 1$，下列计算 y 的公式哪个算得准？

(1) (A) $y=\dfrac{1}{1+2x}-\dfrac{1-x}{1+x}$，　　(B) $y=\dfrac{2x^2}{(1+2x)(1+x)}$

(2) (A) $y=\dfrac{2|x|}{\sqrt{\dfrac{1}{|x|}+|x|}+\sqrt{\dfrac{1}{|x|}-|x|}}$

(B) $y=\sqrt{|x|+\dfrac{1}{|x|}}-\sqrt{\dfrac{1}{|x|}-|x|}$

(C) $y=\dfrac{2|x|^{3/2}}{\sqrt{1+x^2}+\sqrt{1-x^2}}$

(D) $y=\dfrac{\sqrt{1+x^2}-\sqrt{1-x^2}}{\sqrt{|x|}}$

(3) (A) $y=\dfrac{2\sin^2 x}{x}$，　　(B) $y=\dfrac{1-\cos 2x}{x}$

(4) (A) $y=\ln\dfrac{1-\sqrt{1-x^2}}{|x|}$，　　(B) $y=\ln|x|-\ln(1+\sqrt{1-x^2})$

(C) $y=\ln\dfrac{|x|}{1+\sqrt{1-x^2}}$

1-8 已知数列 x_n 收敛于方程 $f(x)=0$ 的根 α，问计算数列 y_n 和 z_n 宜采用哪个算式？为什么？

(1) (A) $y_n=x_{n+1}-\dfrac{(x_{n+1}-x_n)^2}{x_{n+1}-2x_n+x_{n-1}}$

(B) $y_n=\dfrac{x_{n+1}x_{n-1}-x_n^2}{x_{n+1}-2x_n+x_{n-1}}$

(2) (A) $z_n=x_n-\dfrac{(x_n-x_{n-1})f(x_n)}{f(x_n)-f(x_{n-1})}$

(B) $z_n=x_n-\dfrac{x_n-x_{n-1}}{f(x_n)/f(x_{n-1})-1}\cdot\dfrac{f(x_n)}{f(x_{n-1})}$

1-9 化简或改写下列算式，减少运算次数：

(1) $(x-5)^4+9(x-5)^3+7(x-5)^2+6(x-5)+4$

(2) $1+x+x^2/2!+x^3/3!+\cdots+x^n/n!$

(3) $\dfrac{1}{1\times 3}+\dfrac{1}{2\times 4}+\dfrac{1}{3\times 5}+\cdots+\dfrac{1}{99\times 101}$