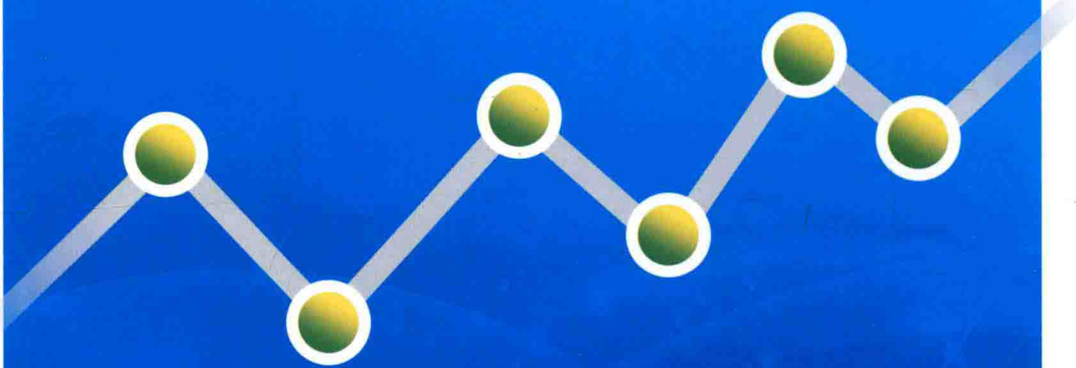
数值分析

Numerical Analysis

许　峰　主编



中国科学技术大学出版社

数值分析

Numerical Analysis

许 峰 主编

中国科学技术大学出版社

内 容 简 介

本书主要介绍了基本的、常用的数值计算方法及其理论，内容包括插值与逼近、数值微分与积分、线性方程组的数值求解、非线性方程和方程组的数值解法、常微分方程的数值解法和特征值的数值计算等. 书中对各种计算方法的构造思想做了较详细的阐述，对稳定性、收敛性、误差估计及算法的优缺点等也做了适当的讨论. 本书结构严谨，条理清晰，语言通俗易懂，论述简明扼要，且配有较丰富的复习思考题和习题.

本书可作为工科专业研究生的教材或教学参考书，也可以供从事科学与工程计算的科技工作者阅读参考.

图书在版编目(CIP)数据

数值分析/许峰主编. —合肥：中国科学技术大学出版社，2017.8
ISBN 978-7-312-04319-2

Ⅰ. 数…　Ⅱ. 许…　Ⅲ. 数值分析—研究生—教材　Ⅳ. O241

中国版本图书馆 CIP 数据核字(2017)第 211283 号

出版　中国科学技术大学出版社
安徽省合肥市金寨路 96 号，230026
http://press.ustc.edu.cn
https://zgkxjsdxcbs.tmall.com
印刷　合肥华苑印刷包装有限公司
发行　中国科学技术大学出版社
经销　全国新华书店
开本　710 mm×1000 mm　1/16
印张　19.75
字数　409 千
版次　2017 年 8 月第 1 版
印次　2017 年 8 月第 1 次印刷
定价　39.00 元

前　言

随着计算机的广泛应用和科学技术的迅速发展，使用计算机进行科学计算已成为科学研究、工程应用中越来越不可或缺的一个环节．因此，科学计算的核心——“数值分析”已被许多工科专业列为硕士研究生的学位课程．

安徽理工大学自20世纪80年代末开始在工科研究生中开设数值分析课程．2015年，数值分析被列为我校首批研究生核心课程．作为课程建设的初步成果，在多年教学实践和教学研究的基础上，课程组编写了本书的初稿．本书着重介绍了基本数值计算方法的构造，对误差估计、算法的收敛性、数值稳定性、适用范围及优缺点等做了适当分析，并配备了较丰富的例题和习题．本书具体内容包括插值与逼近、数值微分与积分、线性方程组的数值求解、非线性方程和方程组的数值解法、常微分方程的数值解法和特征值的数值计算等，完全符合工科研究生数学课程指导委员会制定的工科硕士研究生数值分析课程教学基本要求．

需要指出的是，与其他所有教材编写者一样，作者在编写本教材时，在诸如内容编排、定理的论述、例题和习题的选择等方面参考、借鉴了多种优秀数值分析教材．在此，向这些教材的作者表示感谢和敬意．

这本教材从开始编写到完成只有几个月时间，而且是在繁重的工作之余编写的．因此，教材中存在的问题甚至错误是在所难免的．作者期待着广大读者特别是各位同行提出批评意见和建议．

作　者

2017年6月

目　　录

第 1 章　引　　论

1.1　数值分析及其特点

1.1.1　数值分析及其研究对象

数值分析是数学学科中的一个分支，主要研究用计算机求解各种数学问题的数值计算方法及其理论．因此，数值分析也称计算方法或科学与工程计算．

许多数学问题往往不能用解析的方法来求解．例如，除了少数几类典型方程外，大部分微分方程没有解析解；五次及以上的代数方程没有解析解法；高阶线性方程组虽然可以根据 Cramer① 法则进行求解，但计算量太大，在实际中需要寻求高效的数值解法；由于被积函数过于复杂或其原函数不是初等函数等原因，很多定积分也需要用数值方法计算．

用数值方法求解科学技术问题通常有以下步骤：

(1) 根据实际问题建立数学模型；

(2) 针对具体模型选择或设计数值计算方法；

(3) 根据计算方法编程或利用数学软件在计算机上求出结果．

第(1)步建立数学模型主要是应用数学的任务，而第(2)和第(3)步就是数值分析的任务，即数值分析研究的对象，内容主要包括：算法设计、误差分析、收敛性、稳定性、计算量、存储量、自适应性等．本课程只介绍最基本、最常用的数值计算方法及其理论，包括插值与逼近、数值微分与积分、线性方程组的数值求解、非线性方程和方程组的数值解法、常微分方程的数值解法和特征值的数值计算等．

1.1.2　数值分析的特点

数值分析有下列三大特点：

---

① 克拉默(Gabriel Cramer，1704～1752)是瑞士数学家，主要著作有《代数曲线的分析引论》．

(1) 首先要有可靠的理论分析，以确保算法在理论上的收敛性和数值稳定性；

(2) 其次要对计算结果进行误差估计，以确定其是否满足精度；

(3) 还要考虑算法的运行效率，即算法的计算量与存储量.

与其他数学分支学科类似，数值分析也有一套完整的理论体系，其作用是对设计的算法进行理论上的分析，研究算法的收敛性和数值稳定性. 与解析计算不同，由于数值计算中的误差不可避免，计算结果通常为近似值，所以计算完成后，往往还要对计算结果进行误差分析和估计，以判定其是否满足精度要求. 因为数值方法是面向计算机的，必须要考虑算法的计算量和存储量. 例如，离散傅里叶变换(DFT)的计算量为 N^2，而快速傅里叶变换(FFT)的计算量为 $2N\log_2 N$，即使采样点数 N 仅为 32K，FFT 的计算效率也提升了近 1000 倍.

例 1.1 分析用 Cramer 法则解一个 n 阶线性方程组的计算量.

解 计算机的计算量主要取决于乘除法的次数. 用 Cramer 法则解一个 n 阶线性方程组需计算 $n+1$ 个 n 阶行列式，而用定义计算 n 阶行列式需 $n!(n-1)$ 次乘法，故总共需 $(n+1)n!(n-1)=(n+1)!(n-1)$ 次. 此外，还需 n 次除法.

当 $n=20$ 时，计算量约为 $(n+1)!(n-1)=9.7\times10^{20}$ 次乘法. 即使用每秒百亿次乘法的超级计算机，也需计算 3000 多年才能完成.

可见，Cramer 法则仅仅是理论上的，而不是面向计算机的.

1.2 误差的基本概念

1.2.1 误差的来源与分类

绝大多数的数值计算结果都会有误差. 引起误差的原因是多方面的，可以大致分为如下四类：

(1) **模型误差**

将实际问题转化为数学问题即建立数学模型时，通常要对实际问题进行抽象和简化. 因此，数学模型只是实际问题的一种近似、粗糙的描述. 这种数学模型与实际问题之间的误差称为模型误差.

(2) **观测误差**

在建立的数学模型中往往涉及根据观测得到的物理量，如电压、温度、长度等，而观测不可避免地会带有误差. 这种误差称为观测误差.

(3) **截断误差**

在计算机中常常会遇到只有通过无限过程才能得到最终结果的情形，但实际

计算时只能采用有限过程，如无穷级数求和，只能取前面有限项之和来近似代替，这种以有限过程代替无限过程的误差称为截断误差. 在数值计算时，有时要用容易计算的简化问题代替不易计算的原问题，这种方法误差通常也归为截断误差. 例如，用差商作为导数的近似值，这种离散化的误差也被视为截断误差.

(4) **舍入误差**

在计算中遇到的数据位数可能很多，也可能是无穷小数. 但计算时只能对有限位进行运算，一般采用四舍五入的截尾方法. 这类误差称为舍入误差，也称计算误差.

例 1.2 根据 Taylor① 展式 $e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + R_n(x)$计算 e^{-1}，要求误差小于 0.01.

解 因为 $5! = 120 > 100$，所以只需展开到 5 次项：

$$e^{-1} = 1 + (-1) + \frac{(-1)^2}{2!} + \frac{(-1)^3}{3!} + \frac{(-1)^4}{4!} + \frac{(-1)^5}{5!} + R_5(x).$$

略去余项后得

$$e^{-1} \approx \frac{1}{2} - \frac{1}{6} + \frac{1}{24} - \frac{1}{120}.$$

这一过程产生的误差即为截断误差.

取 4 位有效数字，得 $e^{-1} \approx 0.3667$，由此产生的是舍入误差.

少量的舍入误差是微不足道的，但在计算机上做了成千上万次运算后，舍入误差的积累有时可能是十分惊人的.

根据误差来源的分析可以得到如下结论：误差是不可避免的，在数值计算中只要能求出满足精度的近似解即可. 在数值计算中，要设法减少误差，提高精度. 在四种误差中，前两种误差是客观存在的，而后两种误差是由计算方法所引起的. 本课程主要研究数学问题的数值解法，因此只涉及截断误差和舍入误差.

1.2.2 绝对误差、相对误差和有效数字

定义 1.1 设 x 是准确值，x^* 为 x 的一个近似值，则称 $e = |x^* - x|$ 为近似值 x^* 的**绝对误差**，简称**误差**. 若 $x \neq 0$，则称

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

为近似值 x^* 的**相对误差**.

由于精确值 x 往往是未知的，所以绝对误差或相对误差很难得到. 在实际计算中，通常用误差的某个适当的上限来代替误差.

---

① 泰勒(Brook Taylor，1685～1731)是英国数学家，他主要以泰勒公式和泰勒级数而著名.

定义 1.2 设 x^* 是准确值 x 的一个近似值.若存在 $\varepsilon>0$,使得 $|x^*-x|\leqslant\varepsilon$,则称 ε 为近似值 x^* 的**绝对误差限**.若存在 $\varepsilon_r>0$,使得 $e_r\leqslant\varepsilon_r$,则称 ε_r 为近似值 x^* 的**相对误差限**.

显然,误差限不是唯一的,而最小的误差限是很难求得的.一般地,只要根据测量工具或计算方法求得一个适当的上限即可.

例如,用毫米刻度的卷尺做测量,在正常情况下,可知测量值的误差限为 0.5 mm,即卷尺的最小刻度的一半.

与测量类似,若在十进制运算中采用四舍五入,不难证明,近似值的绝对误差限可以取为被保留的最后数位上的半个单位.由此可引入有效数字的概念.

定义 1.3 设 x^* 为准确值 x 的一个近似值,

$$x^* = \pm 10^k \times 0.d_1 d_2 \cdots d_i \cdots,$$

x^* 是有限小数或无限小数,其中 $d_i\ (i=1,2,\cdots)$ 是 $0,1,\cdots,9$ 中的一个数字,且 $d_i \neq 0$,k 为整数.如果 n 为满足

$$|x-x^*| \leqslant 0.5 \times 10^{k-n}$$

的最大非负整数,则 x^* 称为 x 的具有 n 位**十进制有效数字**的近似值.

显然,近似值的有效数字越多,其相对误差限就越小.

1.2.3 函数求值的误差估计

设 $f(x)$ 的二阶导数连续,x^* 为 x 的一个近似值,$f(x^*)$ 为 $f(x)$ 的近似值,x^* 的绝对误差限为 $\varepsilon(x^*)$.根据 Taylor 公式,

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

$$|f(x)-f(x^*)| \leqslant |f'(x^*)|\,|x-x^*| + \frac{|f''(\xi)|}{2!}(x-x^*)^2.$$

略去二阶项后,即可得 $f(x^*)$ 的一个绝对误差限为

$$\varepsilon(f(x^*)) \leqslant |f'(x^*)|\,\varepsilon(x^*).$$

对 n 元函数 $A=f(x_1,x_2,\cdots,x_n)$,设 $x_1^*,x_2^*,\cdots,x_n^*$ 为 $x_1,x_2,\cdots,x_n$ 的近似值,x_k^* 的绝对误差限为 $\varepsilon(x_k^*)$,则根据多元函数的一阶 Taylor 展式有

$$\begin{aligned} A^*-A &= f(x_1^*,x_2^*,\cdots,x_n^*) - f(x_1,x_2,\cdots,x_n) \\ &= \sum_{k=1}^{n}\left[\frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_k}\right](x_k^*-x_k) + R_1 \\ &\approx \sum_{k=1}^{n}\left[\frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_k}\right](x_k^*-x_k). \end{aligned}$$

从而可得函数值 A 的一个近似绝对误差限

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

例 1.3 已测得某场地长 l 的近似值 $l^*=110(\mathrm{m})$，宽 d 的近似值 $d^*=80(\mathrm{m})$. 已知 $|l-l^*|\leqslant 0.2(\mathrm{m})$，$|d-d^*|\leqslant 0.1(\mathrm{m})$，试求面积 $S=ld$ 的绝对误差限和相对误差限.

解 因为 $S=ld$，$\dfrac{\partial S}{\partial l}=d$，$\dfrac{\partial S}{\partial d}=l$，所以

$$\varepsilon(S^*)\approx\left|\left(\frac{\partial S}{\partial l}\right)^*\right|\varepsilon(l^*)+\left|\left(\frac{\partial S}{\partial d}\right)^*\right|\varepsilon(d^*),$$

其中

$$\left(\frac{\partial S}{\partial l}\right)^*=d^*=80,\quad \left(\frac{\partial S}{\partial d}\right)^*=l^*=110,$$

$$\varepsilon(l^*)=0.2,\quad \varepsilon(d^*)=0.1.$$

从而，绝对误差限

$$\varepsilon(S^*)\approx 80\times 0.2+110\times 0.1=27(\mathrm{m}^2),$$

相对误差限

$$\varepsilon_r(S^*)=\frac{\varepsilon(S^*)}{|S^*|}\approx 0.31\%.$$

1.3 算法的数值稳定性与病态问题

在数值计算中，初始数据可能有初始误差，几乎每一步运算都会产生舍入误差，研究这些误差对计算结果的影响是数值分析中的重要问题. 本节首先讨论在不同算法下计算结果受误差的影响程度，然后研究问题对误差的敏感程度.

1.3.1 算法的数值稳定性

首先看一个积分计算的例子.

例 1.4 建立递推公式，计算下列积分

$$I_n=\int_0^1\frac{x^n}{x+5}\mathrm{d}x\quad(n=0,1,2,\cdots,6),$$

并研究计算过程中的误差传播.

解 由

$$\begin{aligned}I_n&=\int_0^1\frac{x^n+5x^{n-1}-5x^{n-1}}{x+5}\mathrm{d}x\\&=\int_0^1 x^{n-1}\mathrm{d}x-5\int_0^1\frac{x^{n-1}}{x+5}\mathrm{d}x\\&=\frac{1}{n}-5I_{n-1},\end{aligned}$$

$$I_0 = \int_0^1 \frac{\mathrm{d}x}{x+5} = \ln\frac{6}{5} \approx 0.1823 \xlongequal{\text{记为}} I_0^*,$$

得 I_n 的递推公式

$$\begin{cases} I_0^* = 0.1823, \\ I_n^* = \dfrac{1}{n} - 5I_{n-1}^* \quad (n = 1,2,\cdots,6). \end{cases}$$

也可以按另一个次序计算积分. 因为

$$\frac{x^n}{6} < \frac{x^n}{x+5} < \frac{x^n}{5},$$

两边积分得

$$\frac{1}{6(n+1)} < I_n < \frac{1}{5(n+1)}.$$

取

$$I_6 \approx \frac{1}{2}\left(\frac{1}{6\times 7} + \frac{1}{5\times 7}\right) \approx 0.02619 \xlongequal{\text{记为}} I_6^*,$$

从而得 I_n 的另一个递推公式

$$\begin{cases} I_6^* = 0.02619, \\ I_{n-1}^* = \dfrac{\dfrac{1}{n} - I_n^*}{5} \quad (n = 6,5,\cdots,1). \end{cases}$$

上述两种递推公式计算的结果及积分的精确值见表 1.1.

表 1.1

n	0	1	2	3	4	5	6
方法 1	0.1823	0.0885	0.0575	0.0458	0.0208	0.0958	−0.3125
方法 2	0.1823	0.0884	0.0580	0.0431	0.0281	0.0281	0.0262
精确值	0.1823	0.0884	0.0580	0.0431	0.0343	0.0285	0.0243

表 1.1 中的结果显示,第一个递推公式刚开始的计算结果的误差较小,但随着 n 的增加,误差也逐渐增大;第二个递推公式恰好与前述情况相反,计算结果的误差随着 n 的增加而逐渐减小. 下面讨论两个递推过程中误差的传播情况.

第一个递推公式第 n 步的计算误差

$$\begin{aligned} E_n &= |I_n - I_n^*| = \left|\left(\frac{1}{n} - 5I_{n-1}\right) - \left(\frac{1}{n} - 5I_{n-1}^*\right)\right| \\ &= 5|I_{n-1} - I_{n-1}^*| = \cdots = 5^n|I_0 - I_0^*| = 5^n E_0, \end{aligned}$$

即第一个递推公式第 n 步将初始误差放大了 5^n 倍.

上述结果也可以写为

$$E_0 = \frac{1}{5^n}E_n,$$

即第二个递推公式第 n 步将初始误差缩小到 $1/5^n$.

可见,对于同一个问题,有的算法受误差的影响较小,可以得到精度较高的近似结果;而另一些算法则受误差的影响较大,得不到精度较高的近似结果.为了衡量算法受误差的影响程度,下面给出算法的数值稳定性的概念.

定义 1.4 若某算法受初始误差或计算过程中产生的舍入误差的影响较小,则称之为**数值稳定**的,反之称之为不稳定算法.

显然,例 1.4 中的第一个递推公式数值不稳定,而第二个递推公式数值稳定.

1.3.2 病态问题

前面讨论的数值稳定性是对算法而言的.对于某些问题,可以找到数值稳定的算法,从而可得精度较高的近似结果.但存在这样的问题,其受误差的影响巨大,无论用什么样的算法都得不到理想的计算结果.

例 1.5 将代数方程 $p(x)=(x-1)(x-2)\cdots(x-20)=0$ 即 $x^{20}-210x^{19}+\cdots+20!=0$ 改为摄动方程 $x^{20}-(210+\varepsilon)x^{19}+\cdots+20!=0$ 即 $p(x)-\varepsilon x^{19}=0$,其中 $\varepsilon=2^{-23}\approx10^{-7}$,讨论这一摄动对解的影响.

解 20 世纪 60 年代,数值分析和数值计算的开拓者和奠基人 Wilkinson① 为了检验计算机的精度,将一个 20 次代数方程

$$p(x)=(x-1)(x-2)\cdots(x-20)=0$$

即

$$x^{20}-210x^{19}+\cdots+20!=0$$

改为摄动方程

$$x^{20}-(210+\varepsilon)x^{19}+\cdots+20!=0,$$

其中 $\varepsilon=2^{-23}\approx10^{-7}$是当时的计算机中最小的浮点数.

通过编程计算,他惊讶地发现摄动方程的根为

$$\begin{aligned}
&1.000000000,\ 2.000000000,\ 3.000000000,\\
&4.000000000,\ 4.999999928,\ 6.000006944,\\
&6.999697234,\ 8.007267603,\ 8.917250249,\\
&10.095266145\pm0.643500904\mathrm{i},\\
&11.793633881\pm1.652329728\mathrm{i},\\
&13.992358137\pm2.518830070\mathrm{i},\\
&16.730737466\pm2.812624894\mathrm{i},
\end{aligned}$$

① 威尔金森(James Hardy Wilkinson, 1919~1986)是英国数学家和计算机学家.他在数值分析领域做出了重要贡献.

$$19.502439400 \pm 1.940330347\mathrm{i},$$

$$20.846908101,$$

其中有 10 个根变成了复根，且有两个根偏离实根超过 2.81 个单位.

Wilkinson 起初认为问题可能出在程序或计算机上，但后来排除了这种可能性，最终怀疑问题在于方程本身. 下面再现 Wilkinson 研究这个问题的过程.

令 $p(x,\varepsilon)=x^{20}-(210+\varepsilon)x^{19}+\cdots+20!$，其根为 $x_i(\varepsilon)(i=1,2,\cdots,20)$，则当 $\varepsilon\to 0$ 时，$x_i(\varepsilon)\to i$. 现在的问题归结为要研究初始数据的微小摄动对根的影响程度，即根 $x_i(\varepsilon)$ 对 ε 的变化率，亦即

$$\left.\frac{\mathrm{d}x_i(\varepsilon)}{\mathrm{d}\varepsilon}\right|_{\varepsilon=0},$$

称之为问题的条件数.

因

$$p[x_i(\varepsilon),\varepsilon]\equiv 0,$$

故

$$\left.\frac{\mathrm{d}x_i(\varepsilon)}{\mathrm{d}\varepsilon}\right|_{\varepsilon=0}=-\left.\left(\frac{\dfrac{\partial p}{\partial \varepsilon}}{\dfrac{\partial p}{\partial x_i}}\right)\right|_{\varepsilon=0}=\frac{x_i^{19}}{\sum\limits_{k=1}^{20}\prod\limits_{\substack{j=1\\ j\neq i}}^{20}(x_i-j)}=\frac{i^{19}}{\prod\limits_{\substack{j=1\\ j\neq i}}^{20}(x_i-j)}.$$

条件数的具体数值见表 1.2.

表 1.2

n	条件数	n	条件数
1	8.2×10^{-18}	11	4.6×10^{7}
2	-8.2×10^{-11}	12	-2.0×10^{8}
3	1.6×10^{-6}	13	6.1×10^{8}
4	-2.2×10^{-3}	14	-1.3×10^{9}
5	6.1×10^{-1}	15	2.1×10^{9}
6	-5.8×10^{1}	16	-2.4×10^{9}
7	2.5×10^{3}	17	1.9×10^{9}
8	-6.0×10^{4}	18	-1.0×10^{9}
9	8.3×10^{5}	19	3.1×10^{8}
10	7.6×10^{6}	20	-4.3×10^{7}

由微分概念

$$x_i(\varepsilon)-x_i(0)\approx\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon}(\varepsilon-0),$$

得

$$|x_i(\varepsilon)-x_i(0)|\approx\left|\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon}\right||\varepsilon|>10^6\varepsilon\quad(i=10,11,\cdots,20).$$

这表明 x^{19} 的系数的微小摄动 ε 将会引起方程许多根的巨大变化.为了描述这类问题,下面给出病态问题的概念.

定义 1.5　若初始数据的微小误差都会对最终的计算结果产生极大的影响,则称这种问题为病态问题或坏条件问题,反之称其为良态问题或好条件问题.

显然,例 1.5 即为一个病态问题.下面再给出一个病态线性方程组.

例 1.6　线性方程组

$$\begin{pmatrix}10&7&8&7\\7&5&6&5\\8&6&10&9\\7&5&9&10\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}=\begin{pmatrix}32\\23\\33\\31\end{pmatrix}$$

的精确解为 $\boldsymbol{x}=(1,1,1,1)^{\mathrm{T}}$.现将其右端向量和系数矩阵中数据做一个扰动,具体数据分别为

$$\begin{pmatrix}10&7&8&7\\7&5&6&5\\8&6&10&9\\7&5&9&10\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}=\begin{pmatrix}32.1\\22.9\\33.1\\30.9\end{pmatrix},$$

$$\begin{pmatrix}10&7&8.1&7.2\\7.08&5.04&6&5\\8&5.98&9.89&9\\6.99&4.99&9&9.98\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}=\begin{pmatrix}32\\23\\33\\31\end{pmatrix}.$$

上述方程组的精确解为 $\boldsymbol{x}=(9.2,-12.6,4.5,-1.1)^{\mathrm{T}}$ 和 $\boldsymbol{x}=(-81,137,-34,22)^{\mathrm{T}}$,与原方程解相比发生了很大的变化.因此,可以判定此方程组为病态方程组.

1.4　数值计算的原则与技术

在数值计算中,不仅算法的选择会对计算产生很大的影响,而且算法的设计也会影响计算结果的精度和运行效率.本节介绍设计数值计算方法时要注意的一些原则和技术.

1.4.1　避免误差危害

如果使用数值不稳定的方法,会由于误差的增长而出现误差危害现象,主要表

现为有效数字的损失. 因此,在实际计算中要注意避免.

例 1.7 当 $a\neq 0$ 时,二次方程 $ax^2+2bx+c=0$ 的两个根的公式通常写成

$$x_1=\frac{-b+\sqrt{b^2-ac}}{a},\quad x_2=\frac{-b-\sqrt{b^2-ac}}{a}.$$

如果 $b^2\gg|ac|$,则 $\sqrt{b^2-ac}\approx|b|$,用上述公式计算,其中之一会出现相近数相减,从而损失有效数字.

例如,方程 $x^2-16x+1=0$ 的根为 $x_1=8+\sqrt{63}$, $x_2=8-\sqrt{63}$. 如果用 3 位十进制数字计算, $\sqrt{63}\approx 7.94$,则 $x_1\approx 15.9$,有 3 位有效数字. 但是 $x_2\approx 0.06$,只有 1 位有效数字. 若用 $x_2=1/x_1$ 计算,得 $x_2\approx 0.0629$,有 3 位有效数字.

一般地,为了避免由于相近数相减导致的有效数字损失,二次方程 $ax^2+2bx+c=0$ 的根可以用公式

$$x_1=\frac{-b-\operatorname{sgn}(b)\cdot\sqrt{b^2-ac}}{a},\quad x_2=\frac{c}{ax_1}$$

来计算.

类似地,在计算 $f(x)=\sqrt{1+x^2}-1$ 时,若 $|x|\ll 1$,则可考虑将公式改写为

$$f(x)=\frac{1}{\sqrt{1+x^2}+1}.$$

有时,在有限位数数值计算过程中会出现"大数"和"小数"相加减的情形,"小数"可能由于舍入而消失,即"大数"吃"小数",这也是一种误差危害. 通常可以通过调整计算次序来避免这种危害.

例 1.8 用 3 位十进制数字计算

$$x=101+\delta_1+\delta_2+\cdots+\delta_{100},$$

其中 $\delta_i\in[0.1,0.4]$($i=1,2,\cdots,100$). 如果按照上式自左至右的顺序逐个相加,则所有的 δ_i 都将在加的过程中被舍掉,得到的结果是 $x\approx 101$. 但是如果把所有的 δ_i 先加起来,则有

$$101+100\times 0.1\leqslant x\leqslant 101+100\times 0.4,$$

即 $x\in[111,141]$,这显然是一个比较准确的结果.

1.4.2 减少运算次数

对于同一个计算问题,如果能减少运算次数,不仅可以节省计算时间,而且还能减少舍入误差,这是数值计算通常遵循的原则,也是数值分析的一个重要研究内容.

例 1.9 给定 x,求五次多项式

$$p_5(x)=a_5x^5+a_4x^4+\cdots+a_1x+a_0$$

的值. 如果先求 a_kx^k($k=1,2,\cdots,5$)再相加,则要做 15 次乘法和 5 次加法. 如果按照