高等教育精品教材

数值分析

SHUZHI FENXI

曾繁慧 主编

中国矿业大学出版社
China University of Mining and Technology Press

数 值 分 析

主编 曾繁慧

中国矿业大学出版社

内容简介

本书介绍了科学和工程实际中常用的数值计算方法及其相关的理论，内容包括非线性方程(组)的数值解法、线性方程组的数值解法、插值法、函数逼近、数值积分与数值微分、常微分方程数值解法、矩阵特征问题的数值计算。每章都有相关的MATLAB应用函数、主要数值方法的MATLAB程序，并配有应用例题、数值计算习题和实验题。为便于自学，数值计算习题附有答案。

本书注重实际应用和科学计算能力的训练，由案例引入数值计算方法，起点低，跨度较大。本书可作为理工科大学各专业以及研究生的“数值分析”课程教材，并可供从事科学与工程计算的科技工作者参考。

图书在版编目(CIP)数据

数值分析/曾繁慧主编. —徐州：中国矿业大学出版社，2009.9

ISBN 978-7-5646-0421-9

Ⅰ.数… Ⅱ.曾… Ⅲ.数值计算 Ⅳ.O241

中国版本图书馆CIP数据核字(2009)第137614号

书　　名 数值分析
主　　编 曾繁慧
责任编辑 周　红　潘俊成
责任校对 李　敬
出版发行 中国矿业大学出版社
(江苏省徐州市中国矿业大学内　邮编 221008)
网　　址 http://www.cumtp.com　**E-mail**：cumtpvip@cumtp.com
排　　版 中国矿业大学出版社排版中心
印　　刷 江苏徐州新华印刷厂
经　　销 新华书店
开　　本 787×1092　1/16　**印张** 18.5　**字数** 462千字
版次印次 2009年9月第1版　2009年9月第1次印刷
定　　价 30.00元

前　言

在计算机应用已广泛普及的信息时代，应用数值计算方法求解数学问题已成为数学学科的重要内容。“数值分析”是研究各种数学问题求解的数值计算理论及其方法的一门课程。在理工科数学类科目教学体系中，“数值分析”是高等理工科院校的重要基础课程，起着承上启下的作用。承上是使微积分、代数与几何、随机数学中的原理得以应用，方法得以实现，启下是为后续课程中数学问题的建模和求解提供思路，激发学生进一步学习数学、应用数学的意识和能力，同时也具有培养学生创新思维、创新能力的作用。

“数值分析”是高等院校数学、力学及计算机等专业的一门主要基础课程，也是理工科各专业本科及研究生的数学基础课程。“数值分析”是数学建模与数学实验之间的桥梁与媒介，本书编写的目的是促使读者掌握用数值计算方法处理各种数学问题的基本理论与技术；通过利用数学软件平台与程序设计的实验环节提高读者的科学计算与数值计算能力，为读者使用计算机解决科学与工程中的实际问题打下良好的理论基础和扎实的应用基础。在信息技术高速发展的时代，掌握这种能力是至关重要的。

培养实践能力强、具备科学研究素质的应用创新型人才对适应社会经济发展的需要和建设创新型国家的需要具有十分重要的战略意义。为配合培养应用创新型人才的需要，本书在作者多年“数值分析”精品课程建设的基础上形成，基于国际上流行通用的MATLAB计算语言，具有新时代的气息。与国内同类教材相比，本书在介绍数值计算理论的基础上更加注重实际应用和科学计算能力的训练。每章由简单的工程应用案例引入，然后由浅入深地介绍实用的数值方法，并配有相关的MATLAB工具箱函数，编写了各种数值算法的函数程序，利用这些函数可以方便、快捷地求解工程实际中的复杂问题。每章在数值计算习题后都配有实验题，这些实验习题需要借助于计算机，利用本书介绍的MATLAB函数及程序来完成。认真准备与完成实验对读者掌握该章数值计算基本理论非常重要，可作为考查其掌握知识的依据。通过本书实验习题环节，培养读者熟练运用计算机与数值计算软件、掌握各种数值计算算法的实现技能及对数值结果进行分析的能力。通过本书的学习，可以促进读者学习如何提出

问题和解决问题，提高读者自身动手能力和独立思考能力，有效培养读者面向工程实际问题的算法设计与实现能力，进而培养读者成为具备科学计算能力、拥有科学研究素质的创新人才。作为教材，本书尽可能保证全书的系统性，既讲述数值方法的原理，又给出了算法公式的推导和描述。为适应各专业大学生和研究生的需要，本书利用简单的应用案例引入数值计算方法，内容按低起点、较大跨度的原则选取。为便于自学，数值计算习题附有答案。本书可作为理工科大学各专业以及研究生的“数值分析”课程教材，并可供从事科学与工程计算的科技工作者参考。

本书介绍了科学和工程实际中常用的数值计算方法及其相关的理论，内容包括非线性方程(组)的数值解法、线性方程组的直接解法和迭代解法、插值法、函数逼近、数值积分与数值微分、常微分方程直接解法和迭代解法、矩阵特征问题的数值计算。

在本书编写中，郭嗣琮教授提出了许多建设性的意见，李付举副教授负责全书的校对工作，本书也借鉴了许多作者的成果，在此一并向他们表示衷心的感谢。

由于作者水平有限，在内容的取材、结构的编排以及叙述方式上难免有不当之处，敬请读者和专家指正。

编　者

2009 年 2 月

目 录

第1章 绪 论

1.1 数值分析的研究对象与特点

数值分析(Numerical Analysis)这门课程的主要内容是研究用计算机求解各种数学问题的数值计算方法,对求得的解的精度进行评估,以及如何在计算机上实现求解等.数值分析是计算数学的一个主要部分,计算数学是数学科学的一个分支.数学学科十分广泛,这里只涉及工程和科学实验中常见的数学问题,如非线性方程(组)、函数的插值与逼近、微积分、线性方程(组)、常微分方程、矩阵的特征值等问题,它们是其他数学的基础.

我们知道,无论是自然科学、社会科学还是其他学科,其研究领域涉及大量数学问题的求解,其解包括解析解和数值解.解析解固然很重要,但不是任何时候都能获得的,即使能得到解析解,有时也不一定实用.例如定积分 $I=\int_a^b \mathrm{e}^{-x^2}\,\mathrm{d}x$,其中的被积函数 $f(x)=\mathrm{e}^{-x^2}$ 没有有限形式的原函数 $F(x)$,因此也就不能用牛顿—莱布尼茨公式 $I=F(b)-F(a)$ 求积分值.又如,工程中常用的常微分方程模型 $y'=f(x,y),y(x_0)=y_0$ 也没有有限形式的解析解,而从应用的角度看能得到数值解也就够了.至于一般的非线性方程(组)、线性方程(组)和矩阵的特征值等问题,几乎无法用解析方法求解.由此可见,数值方法是不可或缺的途径和手段.特别是在计算机高度发展的今天,应用数值方法,不仅可以求解常规的数学问题,而且可以求解大规模的复杂数学问题.

数值分析是一门内容丰富,研究方法深刻,有自身理论体系的课程.概括起来,数值分析的特点为:

① 它是建立在严格的数学理论基础上的一门实用性很强的课程;

② 它面向计算机,根据计算机特点提供实际可行的且计算复杂性好的有效算法;

③ 它具有可靠的理论分析与数值试验,以保证算法的计算结果达到要求的精度,验证一个算法是行之有效的.

根据数值分析课程的特点,学习时我们首先要注意掌握方法的基本原理和思想,要注意方法处理的技巧及其与计算机的结合,要重视误差分析、收敛性及稳定性的基本理论;其次,要通过实际问题的解决,学习使用各种数值方法;为了掌握课程内容,还应做一定数量的理论分析和计算练习,以及结合数学软件 MATLAB 的实验题.由于本课程包括了微积分、线性代数、常微分方程的数值方法,这就要求我们必须掌握这几门课的基本内容才能学好这门

课程.

1.2 数值计算的误差

1.2.1 误差的来源

应用数学方法研究工程或科学问题,一般只能得到问题的近似解. 误差(Error)的产生主要有以下几方面.

(1) **模型误差**. 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. 我们把模型与实际问题之间出现的这种误差称为模型误差.

(2) **观测误差**(测量误差). 建模时,实验、量测等数据误差称为观测误差.

(3) **方法误差**(截断误差,Truncation Error). 由于计算机本身的特性,要求算法必须在有限步内完成,这就要求把数学模型用数值分析方法导出一个计算公式来近似,由此而产生的误差称为方法误差.

例 1-1 由 Taylor 公式求 e^x 的近似值,由于

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

取 n 次多项式近似则有 $e^x \approx 1 + x + \frac{x^2}{2} + \cdots + \frac{x^n}{n!}$,截断误差为 $\frac{e^{\theta x}}{(n+1)!}x^{n+1}$.

(4) **舍入误差**(Roundoff Error). 由于计算机字长有限,参加运算的数据只能截取有限位,由此而产生的误差称为舍入误差.

例 1-2 用 0.333 3 近似代替 1/3,产生的误差为

$$1/3 - 0.333\ 3 = 0.000\ 033\cdots$$

【注】 在数值分析中,我们主要关心方法误差和舍入误差.

1.2.2 误差的概念

精确值是指理论意义上的一个客观量. 设 x^* 为精确值,a 为 x^* 的一个近似数.

(1) **绝对误差**(Absolute Error)

定义 1 称 $E(a) = x^* - a$ 为近似数 a 的**绝对误差**,简称为**误差**.

如果 $|E(a)| \leqslant \delta$,则称 δ 为近似数 a 的**绝对误差限**,简称为**误差限**(Error Bound).

例 1-3 绝对误差的局限性例子. 若测量光速误差为 4 km/s,而运动员的跑速误差为 0.01 km/s. 从数量大小上看,后者误差小.

但是,0.01 km/s=10 m/s,已经接近运动员的真实跑速. 二者比较可知,光速的测量更准确.

一般来说,绝对误差(或误差限)的大小不能充分说明近似数的精确程度.

(2) **相对误差**(Relative Error)

定义 2 称 $E_r(a) = (x^* - a)/x^*$ 为近似数 a 的**相对误差**.

如果 $|E_r(a)| \leqslant \delta_r$,则称 δ_r 为近似数 a 的**相对误差限**.

实际运算时,由于精确值总是不知道的,通常取 $E_r(a) = (x^* - a)/a$,$\delta_r = \delta/|a|$.

例 1-4 $a = 3.14$ 是 π 的近似值,则误差 $|E(a)| = |\pi - 3.14| < 0.002$,误差限

$\delta = 0.002$. 相对误差 $|E_r(a)| \leqslant \dfrac{0.002}{\pi} \approx \dfrac{0.002}{3.14} = 6.369\ 42 \times 10^{-4}$，相对误差限 $\delta_r = 6.369\ 42 \times 10^{-4}$.

(3) 有效数字(Significant Digits or Significant Figures)

当精确值 x^* 有多位数时，常常按四舍五入的原则得到前几位近似值. 例如

$$\pi = 3.141\ 592\ 65\cdots$$

取 3 位，$a = 3.14$，$\delta \leqslant 0.002$；取 5 位，$a = 3.141\ 6$，$\delta \leqslant 0.000\ 008$.

它们的误差都不超过末位数字的半个单位，即

$$|\pi - 3.14| \leqslant \frac{1}{2} \times 10^{-2}, |\pi - 3.141\ 6| \leqslant \frac{1}{2} \times 10^{-4}.$$

一般地，经过四舍五入后得到的近似数，从第一位非零数开始直到最末位，有几位就称该近似数有几位有效数字.

定义 3 设 x^* 的近似值 a 可表示为

$$a = \pm 10^m \times 0.a_1a_2\cdots a_n,$$

其中 a_1 是 1 到 9 中的一个整数，$a_2, \cdots, a_n$ 为 0 到 9 中的任意整数，m 为整数，若使

$$|E(a)| = |x^* - a| \leqslant \frac{1}{2} \times 10^{m-n}$$

成立，则称 a 近似 x^* 有 n 位**有效数字**.

例 1-5 设 $x^* = 0.002\ 567$，$a = 0.002\ 56 = 10^{-2} \times 0.256$，则

$$|x^* - a| \leqslant 0.000\ 05 = \frac{1}{2} \times 10^{-4}.$$

因为 $m = -2$，所以 $n = 2$，即 a 有 2 位有效数字.

例 1-6 设 $x^* = 8.000\ 01$，则 $a = 8.000\ 0$ 具有五位有效数字.

【注】 近似数的有效数字不但给出了近似值的大小，而且还指出了它的绝对误差限.

(4) 有效数字与相对误差的关系

近似值的有效数字与相对误差密切相关. 粗略地说，有效数字的位数是 n，则相当于相对误差约为 10^{-n}，反之亦然.

确切地说，有如下两个定理.

定理 1 设 x^* 的近似数为 $a = \pm 10^m \times 0.a_1a_2\cdots a_n$，其中 $a_1 \neq 0$，如果 a 具有 n 位有效数字，则 a 的相对误差限为

$$|\delta_r| = \left|\frac{a - x^*}{a}\right| \leqslant \frac{1}{2a_1} \cdot 10^{-(n-1)} = \frac{5}{a_1} \cdot 10^{-n}.$$

证明 显然有

$$0.a_1 \times 10^m \leqslant |a| < (0.a_1 + 0.1) \times 10^m \text{ 或 } a_1 \times 10^{m-1} \leqslant |a| < (a_1 + 1) \times 10^{m-1},$$

于是，a 的相对误差

$$|\delta_r| = \left|\frac{a - x^*}{a}\right| \leqslant \frac{1}{2} \times 10^{m-n} \cdot \frac{1}{a_1 \times 10^{m-1}} = \frac{1}{2a_1} \cdot 10^{-(n-1)} = \frac{5}{a_1} \cdot 10^{-n}.$$

【注】 如果 a 的有效数字位数越多，则 a 的相对误差就越小.

反过来，可以从近似数 a 的相对误差限来估计 a 有效数字的位数.

定理 2 设 x^* 的近似数为 $a = \pm(0.a_1a_2\cdots a_n) \times 10^m (a_1 \neq 0)$，如果 a 的相对误差满足

$|\delta_r| = \left|\frac{a-x^*}{a}\right| \leqslant \frac{1}{2(a_1+1)} \times 10^{-(n-1)} = \frac{5}{(a_1+1)} \times 10^{-n}$，则 a 至少具有 n 位有效数字.

证明 由于

$$|a-x^*| = |a|\left|\frac{a-x^*}{a}\right| \leqslant (a_1+1) \times 10^{m-1} \times \frac{1}{2(a_1+1)} \times 10^{-(n-1)} \leqslant \frac{1}{2} \times 10^{m-n},$$

故 a 至少具有 n 位有效数字.

推论 设 $a = \pm(0.a_1a_2\cdots a_n) \times 10^m (a_1 \neq 0)$，如果 a 的相对误差满足

$$|\delta_r| = \left|\frac{a-x^*}{a}\right| \leqslant \frac{1}{2} \times 10^{-n},$$

则 a 至少具有 n 位有效数字.

证明 由于

$$|a-x^*| = |a|\left|\frac{a-x^*}{a}\right| \leqslant (a_1+1) \times 10^{m-1} \times \frac{1}{2} \times 10^{-n}$$

$$\leqslant \frac{(a_1+1)}{10} \cdot \frac{1}{2} \times 10^{m-n} \leqslant \frac{1}{2} \times 10^{m-n},$$

故 a 至少具有 n 位有效数字.

1.2.3 函数值的误差估计

设 a、b 分别为精确值 x、y 的近似值；δa、δb 分别为 a、b 的误差限. 对于一元函数 $f(x)$ 和二元函数 $f(x,y)$，讨论它们的近似值 $f(a)$ 和 $f(a,b)$ 的误差估计问题. 分析方法采用一阶 Taylor 展开，变量更多的函数与此类同.

(1) 一元函数 $f(x)$ 的误差估计

$f(a)$ 为 $f(x)$ 的近似函数值. 设函数 $f(x)$ 在 a 的邻域上连续可微，由一阶近似 Taylor 展开

$$f(x) \approx f(a) + f'(a)(x-a),$$

所以

$$|f(x)-f(a)| \approx |f'(a)(x-a)| \leqslant |f'(a)|\delta a.$$

则近似函数值 $f(a)$ 的误差限、相对误差限估计式：

$$\begin{cases} \delta f(a) \approx |f'(a)|\delta a, \\ \delta_r f(a) = \dfrac{\delta f(a)}{|f(a)|} \approx \left|\dfrac{f'(a)}{f(a)}\right|\delta a. \end{cases}$$

(2) 二元函数 $f(x,y)$ 的误差估计

$f(a,b)$ 为 $f(x,y)$ 的近似函数值. 设函数 $f(x,y)$ 在 (a,b) 的邻域上连续可微，由一阶近似 Taylor 展开

$$f(x,y) \approx f(a,b) + \frac{\partial f(a,b)}{\partial x}(x-a) + \frac{\partial f(a,b)}{\partial y}(x-b),$$

所以

$$|f(x,y)-f(a,b)| \approx \left|\frac{\partial f(a,b)}{\partial x}(x-a) + \frac{\partial f(a,b)}{\partial y}(x-b)\right|$$

$$\leqslant \left|\frac{\partial f(a,b)}{\partial x}\right|\delta a + \left|\frac{\partial f(a,b)}{\partial y}\right|\delta b.$$

则近似函数值 $f(a,b)$ 的误差限、相对误差限估计式为：

$$\begin{cases}\delta f(a,b) \approx \left|\dfrac{\partial f(a,b)}{\partial x}\right|\delta a + \left|\dfrac{\partial f(a,b)}{\partial y}\right|\delta b, \\ \delta_r f(a,b) = \dfrac{\delta f(a,b)}{|f(a,b)|}. \end{cases}$$

(3) 简单算术运算的误差限和相对误差限

用计算机进行数值运算时，由于所有的函数都必须化成算术运算，因此在函数值的误差分析中最基本的是算术运算. 根据二元函数的误差估计，两个数的加、减、乘、除算术运算得到的误差限和相对误差限分别为：

$$\delta(a \pm b) \approx \delta a + \delta b, \delta_r(a \pm b) \approx \frac{\delta a + \delta b}{|a \pm b|},$$

$$\delta(ab) \approx |b|\delta a + |a|\delta b, \delta_r(ab) \approx \frac{\delta a}{|a|} + \frac{\delta b}{|b|} = \delta_r a + \delta_r b,$$

$$\delta\left(\frac{a}{b}\right) \approx \frac{1}{|b|}\delta a + \left|\frac{a}{b^2}\right|\delta b, \delta_r\left(\frac{a}{b}\right) \approx \frac{\delta a}{|a|} + \frac{\delta b}{|b|} = \delta_r a + \delta_r b.$$

【注】 在作加减法时，应尽量避免接近的两个数相减，否则会使相对误差增大，导致有效数字损失. 显然，加减运算结果的精度不会比原始数据的高. 在作乘除法时，计算结果的相对误差是原始数据的相对误差之和，因此，计算结果的精度也不会比原始数据的高. 由于高精度数据的运算需要更多的时间和空间，所以应避免用高精度数据和低精度数据作混合运算，否则是不经济的.

1.3 误差定性分析与避免误差危害

数值运算中的误差分析是个很重要而复杂的问题，上节讨论了不精确数据运算结果的误差限，它只适用于简单情形，然而一个工程或科学计算问题往往要运算千万次，由于每步运算都有误差，所以每步都做误差分析是不可能的，也不科学. 采用有效的方法对误差做出定量估计是很难的. 为了确保数值计算结果的正确性，首先需对数值计算问题做定性分析，为此本节讨论以下三个问题.

1.3.1 算法的数值稳定性

用一个算法进行计算，由于初始数据误差(由舍入误差造成的)在计算中传播使计算结果误差增长很快，则称该算法是数值不稳定的，否则是数值稳定的.

例 1-7 计算积分 $E_n = \int_0^1 \frac{x^n}{x+10}\mathrm{d}x, \quad n = 0,1,\cdots,8.$

解 由

$$E_n + 10E_{n-1} = \int_0^1 \frac{x^n + 10x^{n-1}}{x+10}\mathrm{d}x = \int_0^1 x^{n-1}\mathrm{d}x = \frac{1}{n},$$

可得两个递推算法.

算法 1：$E_n = \frac{1}{n} - 10E_{n-1}, \quad n = 1,2,\cdots,8.$

算法 2：$E_{n-1} = \frac{1}{10}\left(\frac{1}{n} - E_n\right), \quad n = 8,7,\cdots,1.$

利用 MATLAB 的库函数 quadl 计算数值积分，得到算法 1 的初始值 $E_0 =$

0.095 310 179…，算法 2 的初始值 $E_8=0.010\ 194\ 390\cdots$. 利用 MATLAB 程序递推计算，结果见表 1-1.

表 1-1　　两算法计算值与精确值比较

n	E_n(算法 1)	E_n(算法 2)	E_n^*（精确值）
0	0.095 31	0.095 3	0.095 310 18
1	0.046 9	0.046 9	0.046 898 20
2	0.031 0	0.031 0	0.031 017 98
3	0.023 3	0.023 2	0.023 153 53
4	0.016 7	0.018 5	0.018 464 71
5	0.033 3	0.015 4	0.015 352 90
6	−0.166 7	0.013 1	0.013 137 66
7	1.809 5	0.011 5	0.011 480 56
8	−17.970 2	0.010 2	0.010 194 39

算法 1 的 MATLAB 计算程序如下：

```
E0=0.09531;                          %算法 1 的初始值
E(1)=1-10*E0;
for  n=2:8
  E(n)=1/n-10*E(n-1);                %递推计算
end
E8=quadl('x.^8./(x+10)',0,1)         %利用 MATLAB 的 quadl 计算积分值 E8
```

算法 2 的计算程序省略.

由表 1-1 可知，算法 1 是不稳定的，算法 2 是稳定的.

由于串行运算的舍入误差是逐步传递的，所以可用“向后误差”分析方法，即把舍入误差的影响看成是初值误差即输入数据误差的传播. 因此，有下述定义：

定义 4　对于某个算法，若输入数据的误差在计算过程中迅速增长而得不到控制，则称该算法是**数值不稳定**的(Unstable Algorithm)，否则是**数值稳定**的(Stable Algorithm).

对于例 1-7，当仅考虑初始值有误差时，对于算法 1，由

$$E_n^*=\frac{1}{n}-10E_{n-1}^*,E_n=\frac{1}{n}-10E_{n-1},$$

可知误差 $\varepsilon_n=E_n^*-E_n$ 满足：

$$\varepsilon_n=-10\varepsilon_{n-1}=(-10)^n\varepsilon_0,\quad n=1,2,\cdots,$$

因此算法 1 是不稳定的.

对于算法 2，同理可知误差 $\varepsilon_i=E_i^*-E_i$ 满足：

$$\varepsilon_{i-1}=-\frac{1}{10}\varepsilon_i,\quad i=n,n-1,\cdots,1,$$

所以 $\varepsilon_0=\left(-\frac{1}{10}\right)^n\varepsilon_n$，因此算法 2 是稳定的.

蝴蝶效应 什么是蝴蝶效应(Butterfly Effect)?先从美国麻省理工学院气象学家爱德华·洛伦兹(Lorenz,混沌学创始人)的发现谈起.1961年冬天,为了预报天气,洛伦兹用计算机求解仿真地球大气的13个方程式.为了更细致地考察结果,他把一个中间解取出,提高精度再送回.而当他喝了杯咖啡以后回来再看时竟大吃一惊:本来很小的差异,结果却偏离了十万八千里!计算机没有毛病,于是,洛伦兹认定,他发现了新的现象:"对初始值的极端不稳定性".

这个发现非同小可,以致科学家都不理解,几家科学杂志也都拒登他的文章,认为"违背常理":相近的初值代入确定的方程,结果也应相近才对,怎么能大大远离呢!

18年后,洛伦兹在华盛顿的美国科学促进会的一次演讲中提出:在大气运动过程中,即使各种误差和不确定性很小,也有可能在过程中将结果积累起来,经过逐级放大,形成巨大的大气运动.接着,他举出了一个后来闻名于世的例子:一只蝴蝶在巴西扇动翅膀,有可能会在美国的得克萨斯引起一场龙卷风.他的演讲和结论给人们留下了极其深刻的印象.从此以后,所谓"蝴蝶效应"之说就不胫而走,名声远扬了.

1.3.2 病态数学问题与条件数

上面讨论的数值稳定性是对算法而言的,这里的病态则是数学问题即数学模型本身的性质,与算法无关.所谓病态数学问题是指这样的问题:当输入数据(如参数、初始值等)有微小摄动时,会引起解的大扰动;相反的问题为良态数学问题.因为实现算法时总有舍入误差,所以对于病态的数学问题,用任何算法求数值解都是不稳定的.但是,良态数学问题的算法未必是数值稳定的.

病态和良态是相对的,界线比较模糊,病态越严重,对算法稳定性的影响越大.通常用条件数(Condition Number)来衡量问题的病态程度,条件数越大病态可能越严重.不同的数学问题条件数的定义是不同的,例如线性代数方程组的条件数定义可参见第3章.这里举一个简单的例子.

例 1-8 函数求值问题 $y=f(x)$ 的条件数定义为:

$$C(x)=\operatorname{cond}(f(x))=\frac{|xf'(x)|}{|f(x)|}.$$

这是因为

$$\delta_r f(a)=\frac{|f(x)-f(a)|}{|f(x)|}\approx\frac{|f'(x)(x-a)|}{|f(x)|}=\frac{|xf'(x)|}{|f(x)|}\cdot\frac{|x-a|}{|x|}=\operatorname{cond}(f(x))\cdot\delta_r a,$$

若取对数函数 $y=\ln(x)$,则其条件数为 $C(x)=1/|\ln(x)|$,它的值与 x 有关.如

$$C(0.1)\approx 0.43, C(0.9)\approx 0.95, C(0.99)\approx 99.5,$$

$$C(1.01)\approx 100.5, C(10)\approx 0.43, C(100)\approx 0.22,$$

可见,x 越接近1,条件数越大,从而求对数函数值的相对误差越大.

1.3.3 避免误差危害的若干原则

数值计算中首先要分清问题是否病态和算法是否数值稳定,计算时还应尽量避免误差危害,防止有效数字的损失,下面给出若干原则.

(1) 避免两相近的数相减

在数值计算中两相近数相减有效数字会严重损失.

例 1-9 计算 $1-\cos 2^\circ$(用四位函数表示三角函数值).

$$1-\cos 2^\circ \approx 1-0.999\ 4 = 0.000\ 6$$

具有四位有效数字的近似数 0.999 4,经过运算,只有一位有效数字.

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.对于上式,若利用三角恒等式 $1-\cos x = 2\sin^2\frac{x}{2}$ 进行公式变换,则

$$1-\cos 2^\circ = 2\sin^2 1^\circ = 0.000\ 613$$

具有三位有效数字.

也可将 $\cos x$ 展开成泰勒(Taylor)级数后,按

$$1-\cos x = \frac{x^2}{2!}-\frac{x^4}{4!}+\cdots$$

来进行计算.这两种算法都避开了两个相近数相减的不利情况.

又如,当 x 很大时,

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

用右端算式代替左端,可减少有效数字的损失.

如果无法改变算式,则采用增加有效位数进行运算;在计算机上则采用双倍字长的高精度运算.

(2) 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 1-10 计算多项式 $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ 的值.

若直接计算 $a_k x^k$ 再逐项相加,一共需做 $n+(n-1)+\cdots+2+1 = \frac{n(n+1)}{2}$ 次乘法和 n 次加法.

若采用秦九韶算法

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

只要 n 次乘法和 n 次加法就可算出 $P_n(x)$ 的值.

思考 秦九韶算法为什么减少了运算次数?

对于 4 次代数多项式 $P(x) = 2x^4 + 4x^3 - 6x^2 + 5x - 1$,计算 $x = 2$ 时的多项式值 $P(2)$. 秦九韶算法的 MATLAB 计算程序如下:

```
a=[-1  5  -6  4  2];                %4 次多项式的系数
x=2;n=length(a);P=a(n);
for  k=n-1:-1:1
    P=P*x+a(k);                     %秦九韶算法的递推计算
end
P                                   %输出多项式的值
```

MATLAB 内嵌多项式值的计算函数为 polyval,下面数组 a 的元素为多项式的系数,由高次幂至低次幂排列.利用 MATLAB 函数计算多项式值:

```
a=[2  4  -6  5  -1];P=polyval(a,2)
```

秦九韶算法 秦九韶算法是我国宋代的一位数学家秦九韶提出的.国外文献通常称这种算法为霍纳(Horner)算法,其实 Horner 的工作比秦九韶晚了五个多世纪.

秦九韶(1202—1261),字道古,四川安岳人,我国南宋数学家,1247 年写成《数学九章》.霍纳(W. G. Horner,1789—1837),英国人.

例 1-11 用克莱姆(Cramer)法则求解线性方程组 $Ax=b$.

用克莱姆法则求解 n 阶线性方程组,当 $n=20$ 时,需要计算 21 个 20 阶行列式.按行列式定义计算,每个行列式需要计算(20−1)×20! 次乘法,如果加减法的计算次数忽略不计,则总的计算量约为 21×19×20!.假设计算机每秒可作 1 亿次乘除法运算,则求解这样一个线性方程组需用的时间是多少?

利用 MATLAB 程序解算如下:

```
n=1:20;
zjl=21*19*prod(n);          %总计算量
ms=zjl/100000000;           %占用 CPU 的时间(秒)
ns=ms/60/60/24/365          %转换为年数
```

程序运行,得到需用的时间约为 307 816 年.

天啊,30 多万年的时间!

如果改用第 3 章的高斯(Gauss)消元法只需 2 000 多次乘除运算,并且 n 越大两种算法的计算量差别越大.这说明当我们把一个数学模型转化为计算机程序时,对算法的研究是十分重要的.

【注】 一个算法的计算代价也就是完成计算需要支付的金额.算法由算术运算组成,用计算机来实现.计算机计算量的主要依据有两项:一是使用中央处理器(CPU)的时间,主要由算术运算的次数决定;二是占用存储器的空间,主要由使用数据的数量决定.

算法的计算代价称为**算法的复杂度**,其中,算法的计算量称为**时间复杂度**;算法需占用存储空间的量度称为**空间复杂度**.

(3) 要避免用绝对值很小的数作除数

用绝对值很小的数作除数,舍入误差会增大,甚至会在计算机中造成“溢出”错误.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,使数值不稳定,应尽量避免.

又如,计算 $\frac{1}{\sqrt{x+1}-\sqrt{x}}$,当 $x\gg 0$ 时,分母的绝对值很小,此算法是数值不稳定的.

(4) 两数相加要防止大数“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