



重点院校推荐教材

# 数值计算方法

SHUZHI JISUAN FANGFA

刘玲 葛福生 编著



科学出版社

[www.sciencep.com](http://www.sciencep.com)

重点院校推荐教材

# 数值计算方法

刘 玲 葛福生 编著

科学出版社

北 京

## 内 容 简 介

本书是一本全面讲述数值计算方法的教材。全书共分七章,内容包括数值方法的研究及误差分析、非线性方程(组)的数值解、线性方程组的直接方法和迭代方法、函数逼近的插值与曲线拟合法、数值积分与数值微分、常微分方程初值问题与边值问题的数值解、矩阵特征值与特征向量的数值解等。本书概念清晰,理论分析严谨,语言叙述通俗易懂,并注重实用性,所有的算法均配有伪程序和算法框图。各章都附有一定数量的习题,以供读者学习时进行练习。

本书可作为高等院校计算机应用专业等非数学专业理工科本科生的教材,以及工科研究生的参考教材,也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

数值计算方法 / 刘玲, 葛福生编著. —北京: 科学出版社, 2005  
(重点院校推荐教材)

ISBN 7-03-015964-0

I. 数… II. ①刘…②葛… 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(2005)第 081941 号

责任编辑:鞠丽娜 刘亚军 / 责任校对:刘彦妮

责任印制:吕春珉 / 封面设计:三函设计

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

\*

2005年8月第一版 开本: B5(720×1000)

2005年8月第一次印刷 印张: 16 1/4

印数: 1—4 000 字数: 316 000

定价: 23.00 元

(如有印装质量问题, 我社负责调换(环伟))

销售部电话 010-62136131 编辑部电话 010-62138978-8002(HI06)

# 前 言

随着科学技术的飞速发展,科学计算愈来愈显示出其重要性。科学计算的应用之广已遍及各行各业,如气象资料的图像分析,飞机、汽车及轮船的外形设计,高科技研究等都离不开科学计算。因此,作为科学计算的数学工具——数值计算方法,已成为各高等院校数学、物理和计算机应用专业等理工科本科生的专业基础课,也是工科硕士研究生的学位必修课。

本书比较全面地介绍了现代科学技术中常用的数值计算方法,并对这些数值计算方法的计算效果、稳定性、收敛性、适用范围以及优劣性与特点做了简要地分析。全书共分七章,内容包括数值方法的研究及误差分析、非线性方程(组)的数值解、线性方程组的直接方法和迭代方法、函数逼近的插值与曲线拟合法、数值积分与数值微分、常微分方程初值问题和边值问题的数值解及矩阵特征值与特征向量的数值解等。

为了适应愈来愈多的非数学专业学生的学习,本教材在阐述数学理论的同时注重对算法的实现,并特别注重以下几方面。

第一,理论证明重思维方法,论述简明扼要,所用的知识基本限于数学分析和线性代数等基本内容之内。

第二,强化各种算法的计算机实现。书中所有算法均配有算法框图和伪程序,可使学生加深理解各种算法的逻辑结构,并可对照算法快速入门。

第三,在习题设置方面,分为A、B两类,A类为基本算法题和基础理论题,B类为加深拓宽题。本科学生只要完成A类习题即可,B类习题为有兴趣读者进一步学习之用,一时做不出来也不影响对教材基本内容的掌握。

第四,在内容安排上,尽可能符合认识论,即由浅入深,由简单到复杂、由特殊到一般、由具体到抽象等原则。其目的在于使学生从普遍感到本门课程“繁”、“难”等思想中解放出来,使之能尝到甜头,学出兴趣。

本书可作为非数学专业理工科在职申请硕士学位、全日制非数学专业理工科本科生的教材,也可供广大科技工作者参考学习。

本书全部讲完需要80课时左右,如要求60学时,可对带星号的内容不作要求,也可根据学习者的实际情况及实际学时,有选择地讲解部分内容。

为便于多媒体教学,本教材配有电子教案,有教学需求的教师可到科学出版社网站上下载(<http://www.sciencep.com>)。

本书在编写过程中得到了南京大学林应举、包雪松等教授的指导,研究生蒋智凯、崔隽同学也做了大量的工作,在此深表谢意。由于编者水平有限,书中谬误和不妥之处在所难免,热忱希望广大读者批评指正。

# 目 录

<b>第 1 章 绪论</b> .....	1
1.1 数学问题的数值解法实例 .....	2
1.2 误差概念和有效数字 .....	4
1.2.1 误差概述 .....	4
1.2.2 误差和有效数字 .....	6
1.2.3 函数值的误差估计 .....	7
1.3 算法的优化 .....	8
习题 .....	10
<b>第 2 章 非线性方程与方程组的数值解法</b> .....	13
2.1 二分法 .....	13
2.2 一般迭代法 .....	17
2.2.1 迭代法及收敛性 .....	17
2.2.2 Steffensen 加速收敛方法 .....	19
2.3 Newton 切线法 .....	24
2.3.1 Newton 迭代法和收敛性 .....	24
2.3.2 代数方程的 Newton 迭代法 .....	28
2.4 弦截法 .....	30
<sup>*</sup> 2.5 非线性方程组的数值解法 .....	33
2.5.1 一般迭代法 .....	33
2.5.2 Newton 迭代法 .....	35
2.5.3 拟 Newton 法 .....	36
习题 .....	40
<b>第 3 章 线性方程组的数值解法</b> .....	44
3.1 Gauss 消元法 .....	45
3.1.1 Gauss 顺序消元法 .....	45
3.1.2 Gauss 主元素消元法 .....	48
3.2 矩阵的三角分解法 .....	54
3.2.1 Gauss 消元法矩阵形式 .....	54
3.2.2 Doolittle 分解 .....	56
3.2.3 对称矩阵的 Cholesky 分解 .....	58
3.2.4 三对角方程组求解的追赶法 .....	62

3.3	矩阵求逆 .....	64
3.4	向量和矩阵的范数 .....	68
3.4.1	向量范数 .....	68
3.4.2	矩阵范数 .....	69
3.4.3	矩阵的谱半径和矩阵序列收敛性 .....	72
3.5	病态方程组与矩阵条件数 .....	73
3.5.1	病态方程组与扰动方程组的误差分析 .....	73
3.5.2	矩阵条件数 .....	75
3.6	线性方程组的迭代方法 .....	77
3.6.1	线性方程组迭代法概述 .....	77
3.6.2	Jacobi 迭代法和 Gauss-Seidel 迭代法 .....	77
3.6.3	线性方程组迭代法收敛条件 .....	81
*	3.6.4 分块迭代法简介 .....	84
*	3.6.5 改善精度的迭代校正法 .....	85
	习题 .....	86
<b>第 4 章</b>	<b>函数逼近的插值法与曲线拟合法</b> .....	<b>92</b>
4.1	Lagrange 插值法 .....	92
4.2	Newton 插值法 .....	97
4.2.1	差商及其性质 .....	97
4.2.2	Newton 插值公式 .....	98
4.2.3	等距节点 Newton 插值公式 .....	101
4.3	Hermite 插值 .....	103
4.4	三次样条插值 .....	108
4.4.1	分段插值 .....	108
4.4.2	三次样条插值 .....	108
4.5	曲线拟合的最小二乘法 .....	112
4.5.1	最佳平方逼近 .....	112
4.5.2	对离散数据的曲线拟合最小二乘法 .....	116
4.5.3	矛盾方程组的最小二乘解 .....	119
	习题 .....	121
<b>第 5 章</b>	<b>数值积分</b> .....	<b>126</b>
5.1	Newton-Cotes 求积公式 .....	126
5.1.1	Cotes 系数 .....	126
5.1.2	Newton-Cotes 公式截断误差及代数精度 .....	129
5.2	复化求积公式 .....	131
5.2.1	定步长复化求积公式 .....	132

5.2.2	变步长求积公式	133
5.3	Romberg 求积公式	137
5.3.1	外推法基本思想	137
5.3.2	Romberg 求积算法	138
* 5.4	Gauss 求积公式	141
5.4.1	正交多项式	142
5.4.2	Gauss 型求积公式一般理论	146
5.4.3	Gauss-Legendre 求积公式	148
5.4.4	Gauss-Chebyshev 求积公式	150
5.4.5	一般权函数下 Gauss 型求积公式	151
5.5	数值微分	153
5.5.1	Taylor 展开式方法	153
5.5.2	数值微分的插值方法	155
5.5.3	数值微分的隐式格式	157
	习题	160
<b>第 6 章</b>	<b>常微分方程数值解法</b>	<b>164</b>
6.1	初值问题的 Euler 方法	164
6.1.1	Euler 方法	164
6.1.2	误差概述	166
6.1.3	数值稳定性分析	168
6.2	Runge-Kutta 方法	170
6.2.1	二阶 R-K 方法	171
6.2.2	四阶 R-K 方法	172
6.2.3	R-K 法的稳定性	174
* 6.2.4	一般显式单步法的收敛性	175
6.2.5	隐式 R-K 法	177
6.3	线性多步法	178
6.3.1	基于数值积分的方法	179
6.3.2	基于 Taylor 展开式的方法	182
6.4	一阶常微分方程组数值解法	185
6.4.1	解一阶常微分方程组的 R-K 方法	186
6.4.2	刚性方程组	190
6.5	常微分方程边值问题的数值解法	195
6.5.1	差分方程的建立	195
* 6.5.2	差分方程的可解性与收敛性	198
* 6.5.3	打靶法	201

习题 .....	204
<b>第 7 章 矩阵特征值和特征向量的数值解法 .....</b>	<b>208</b>
7.1 幂法 .....	208
7.1.1 幂法原理及实用幂法 .....	208
7.1.2 幂法的加速收敛方法 .....	213
7.1.3 逆幂法 .....	217
7.2 Jacobi 法 .....	220
7.2.1 古典 Jacobi 方法 .....	220
7.2.2 Jacobi 法的改进 .....	225
7.3 QR 算法 .....	227
7.3.1 Householder 变换 .....	227
7.3.2 矩阵的 QR 分解 .....	230
7.3.3 QR 算法 .....	231
习题 .....	248
<b>主要参考文献 .....</b>	<b>251</b>



# 第 1 章 绪 论

数值分析或数值计算方法主要研究如何运用计算机去获得数学问题的数值解的理论和算法。对那些在经典数学中,用解析方法在理论上已作出解的存在,但要求出它的解析解又十分困难,甚至不可能的这一类数学问题,数值解法就显得不可缺少,同时又十分有效。一个经典的例子就是高次代数方程根的求解问题。尽管代数基本定理已明确指出: $n$ 次代数方程

$$a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n = 0 \quad (a_0 \neq 0)$$

在复数域内有且仅有  $n$  个根,然而求解这  $n$  个根并非是一件容易的事,一般解析方法很难奏效。又如线性方程组  $Ax = b$ , 在矩阵  $A$  非奇异条件下,著名的 Cramer 法则给出它的解:  $x_i = \frac{D_i}{D} (i = 1, 2, \cdots, n)$ 。其中,  $D = \det(A)$ ,  $D_i = \det(A_i)$ ,  $A_i$  为  $b$  代替  $A$  中第  $i$  列所得矩阵。尽管这个解析解在理论上有着重大意义,然而,要实际算出这些解却计算量过大,因此探讨数值方法求解是十分必要的。还有一些问题,几乎无法给出解析方法去求解。例如,矩阵的特征值与特征向量计算,规划问题的最优解,大量实验数据的曲线与曲面拟合,非线性方程和方程组的求解等都是经典数学解析方法难以胜任的。正是这些矛盾的存在推动着数学的发展,数值方法不仅弥补了解析方法的空白和缺憾,而且本身也在日新月异地获得迅速的发展,特别是随着计算机的快速更新换代,软件系统的高效完善,为数值算法提供了极为有利的工具。近几十年,许多算法十分成功地解决了科学和工程中所出现的极为困难和复杂的问题,推动了整个世界的科技进步。今天,一个国家科学数值计算能力的高低,已成为衡量该国科技水平的重要标志之一。因此,学习和掌握数值分析对于每个科技工作者,都是十分重要和有意义的。

数值数学除实用性较强外,理论上也是十分严谨的。它要求读者具备一定的数学理论基础,主要有微积分、线性代数、微分方程等。此外,还要有计算机算法语言等知识。值得注意的是,从数值分析所讨论的一系列算法,到计算机的实际实现,直至正确地算出结果,这中间还有一个不小的距离。因此重视算法的编程及上机调试是学习本课程必不可少的环节,使用近十年来较为流行的 C 语言, C++ 进行编程,均能取得较好的效果。目前,更有一些专家推荐在美国十分流行的 Matlab 语言或其他功能更强、实现更为方便的软件系统。对初学数值分析的读者而言,只要能正确、熟练地掌握一种语言进行编程,实现算法,我们认为已经达到了目的。在这个基础上再去进一步地学习和掌握不断出现的更新的语言,一般不会出现原则性障碍。

## 1.1 数学问题的数值解法实例

我们先通过观察几个例子,初步了解数学问题的数值解题思想方法和效果。

**例 1.1.1** 试求函数方程  $x = \cos x$  在区间  $(0, \frac{\pi}{2})$  内的一个根。

**解** 令  $f(x) = x - \cos x$ , 易知  $f(x)$  在  $[0, \frac{\pi}{2}]$  上是连续函数, 且

$$f(0)f(\frac{\pi}{2}) = (-1) \times \frac{\pi}{2} < 0$$

由零点定理知, 方程  $f(x) = 0$  在  $(0, \frac{\pi}{2})$  内至少有一个零点。又由

$$f'(x) = 1 + \sin x > 0, x \in (0, \frac{\pi}{2})$$

知上述零点唯一。本题用解析方法求解较为困难。若用图解法, 可大致判定此零点位置。作图像

$$\begin{cases} y = x \\ y = \cos x \end{cases}$$

取两曲线交点  $p^*$  的横坐标  $x^*$  为所求方程的解。从图 1.1.1 可以看出,  $x^*$  大致位于  $\frac{\pi}{4}$  附近。

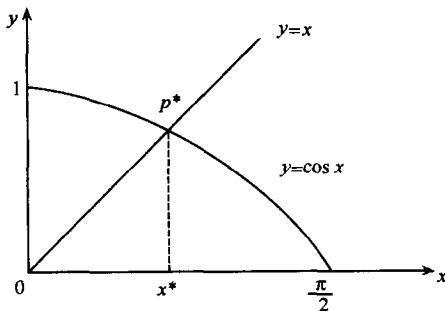


图 1.1.1 方程  $x = \cos x$  根的位置示意图

用数值方法求解本题有多种算法, 我们选用简单迭代法和 Newton 法分别求解。简单迭代法格式如下:

$$x_{n+1} = \cos x_n \quad (n = 0, 1, 2, \dots) \quad (1.1.1)$$

取初始值  $x_0 = 0.75$ , 按算式(1.1.1)可算得

$$x_1 = 0.731\ 688\ 868$$

$$x_2 = 0.744\ 047\ 084$$

$$x_3 = 0.735\ 733\ 618$$

.....

$$x_{41} = 0.739\ 085\ 134$$

$$x_{42} = 0.739\ 085\ 133$$

$$x_{43} = 0.739\ 085\ 133$$

由上可知:  $x^* \approx x_{43} = 0.739\ 085\ 133$ 。

Newton 法迭代格式如下:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n - \cos x_n}{1 + \sin x_n} \quad (n = 0, 1, 2, \dots) \quad (1.1.2)$$

仍取初始值  $x_0 = 0.75$ , 按算式(1.1.2)可算得

$$x_1 = 0.739\ 111\ 138,$$

$$x_2 = 0.739\ 085\ 133,$$

$$x_3 = 0.739\ 085\ 133,$$

由此可知:  $x^* \approx x_3 = 0.739\ 085\ 133$ 。

我们看到, 采用上述两种数值解法都获得了方程具有九位相同数字的近似解  $x^*$ 。不同的是, 简单迭代法要迭代到  $x_{43}$ , 而 Newton 法只需迭代到  $x_3$ 。

**例 1.1.2** 计算定积分。

$$(1) I_1 = \int_0^1 \frac{4}{1+x^2} dx \qquad (2) I_2 = \int_0^1 e^{-x^2} dx$$

**解** (1) 由 Newton-Leibniz 公式有

$$I_1 = 4 \arctan x \Big|_0^1 = \pi$$

数值解法有多种算法, 如选择  $n = 2$ ,  $h = \frac{1}{2}$ , 被积函数  $f(x) = \frac{4}{1+x^2}$  的复化 Simpson 公式有

$$I_1 \approx \frac{h}{6} [f(0) + 4f(\frac{1}{4}) + 2f(\frac{1}{2}) + 4f(\frac{3}{4}) + f(1)] = 3.141\ 568\ 627$$

这一解答与精确解  $\pi$  相比较有五位数字相同。

(2) 本题被积函数  $f(x) = e^{-x^2}$ , 它不存在有限形式的初等函数的原函数, 故 Newton-Leibniz 公式失效。

我们仍选择  $n = 2$ ,  $h = \frac{1}{2}$  的复化 Simpson 公式进行数值求解有

$$I_2 \approx 0.746\ 855\ 379$$

依据被积函数的性态, 由(1)的结果看来, 本题的数值解与精确解相比较, 大约会有四到五位数字相同。

例 1.1.3 求解初值问题。

$$\begin{cases} \frac{dy}{dx} = y - \frac{2x}{y} \\ y(0) = 1 \end{cases} \quad (1.1.3)$$

解 式(1.1.3)是 Bernoulli 方程, 令  $u = y^2$  可得解析解  $y = \sqrt{2x+1}$ 。

本题数值解法算式很多, 如我们选择经典的四阶 Runge-Kutta 算式

$$\begin{cases} y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 = hf(x_n + h, y_n + k_3) \quad (n = 0, 1, 2, \dots) \end{cases} \quad (1.1.4)$$

这里,  $f(x, y) = y - \frac{2x}{y}$ ;  $h$  为步长。现取  $h = 0.05$ , 经上机计算, 其部分结果如表 1.1.1 所示。

表 1.1.1 四阶 Runge-Kutta 法求解结果与精确解对照表

$x_n$	$y_n$	$y = \sqrt{2x+1}$	$x_n$	$y_n$	$y = \sqrt{2x+1}$
0	1.000 00	1.000 00	1.2	1.849 31	1.849 31
0.2	1.183 22	1.183 22	1.4	1.943 96	1.943 96
0.4	1.341 64	1.341 64	1.6	2.049 39	2.049 39
0.6	1.483 24	1.483 24	1.8	2.144 76	2.144 76
0.8	1.612 45	1.612 45	2.0	2.236 07	2.236 07
1.0	1.732 05	1.732 05	...	...	...

从表 1.1.1 可以看出, 用四阶 Runge-Kutta 法所得到的数值解与精确的解析解 (这里的开方计算之结果也不是精确的) 相比较有六位数字相吻合, 在实际运用中已是够精确的了, 因此可以说数值方法求解毫不逊色。与例 1.1.2(2) 相似, 实际问题中有相当多的微分方程其解析解是无法求出的, 此时数值解法就显示出了它的优越性。从这个意义上, 数值方法不仅是解析方法的重要补充, 而且是数学方法的一个重要发展。

## 1.2 误差概念和有效数字

### 1.2.1 误差概述

首先, 我们观察一例, 了解误差产生的原因及种类, 进而认识任何科学和工程

中所出现的问题,其解的精确性总是相对的,而误差则是绝对的。

**例 1.2.1** 试求摆长为  $l$  的单摆运动的周期  $T$ 。

**解** 如图 1.2.1 所示,设质点质量为  $m$ ,如果忽略空气阻力和摆轴在  $O$  点的摩擦力,由牛顿第二定律有

$$mg \sin \theta = -ml \frac{d^2 \theta}{dt^2} \quad (1.2.1)$$

当  $\theta$  较小时,由  $\sin \theta \approx \theta$ ,则有

$$\frac{d^2 \theta}{dt^2} + \omega^2 \theta = 0 \quad (1.2.2)$$

其中,  $\omega = \sqrt{\frac{g}{l}}$ 。解式(1.2.2)得

$$\theta(t) = A \sin(\omega t + \varphi) \quad (1.2.3)$$

其中,  $A, \varphi$  均为常数。由式(1.2.3)知,摆长为  $l$  的单摆运动周期为

$$T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{l}{g}} \quad (1.2.4)$$

式(1.2.4)就是著名的单摆运动周期计算公式。现在我们来分析一下上述求解过程的误差情况。

首先,方程(1.2.1)是在忽略了空气阻力和在  $O$  点的摩擦力条件下得到的,因此式(1.2.1)只是对问题作了近似的描述。这里所建立的数学模型(1.2.1)是忽略了上述因素而产生的误差,称为模型误差。人们在研究问题时,常常是抓住主要因素,略去次要因素,这不仅有利于数学模型的建立,同时也不失其对问题最本质部分的描述。

其次,方程(1.2.2)由  $\sin \theta \approx \theta$  得到,显然略去了  $\sin \theta$  的 Taylor 展开式中  $-\frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots$  的项,我们称这种误差为截断误差。

第三,在解答式(1.2.4)中,已知量  $g, l$  通过观测才能获得。受观测方式、仪器精度及外部观测条件等多种因素影响,  $g$  和  $l$  只能是近似值,不可能获得精确值,这种误差称为观测误差。

最后,利用式(1.2.4)在计算周期  $T$  时,对原始数据及中间过程的数据经常进行舍入。在计算机上受字长限制,这种舍入不可避免,由此产生的误差称为舍入误差。

综上所述,一个科学或工程问题在求解过程中,常见有四种类型误差,即模型误差、截断误差、观测误差和舍入误差。由此可知,几乎没有一个问题的求解是不包含误差的。问题在于我们在讨论问题时,如何分析和控制这些误差,使结果达到预期的精度。在数值分析里,主要研究截断误差和舍入误差,而数学模型误差和数

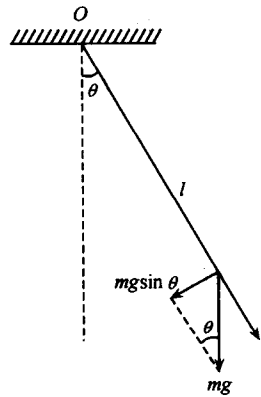


图 1.2.1 数学摆示意图

据观测误差则常常是通过其他方法加以改善。

### 1.2.2 误差和有效数字

**定义 1.2.1** 设  $x^*$  为准确数  $x$  的一个近似数, 称

$$e(x^*) = x^* - x$$

和

$$e_r(x^*) = \frac{e(x^*)}{x} \quad (x \neq 0) \quad (1.2.5)$$

为近似数  $x^*$  的绝对误差(简称误差)和相对误差。

近似数  $x^*$  的绝对误差的绝对值  $|e(x^*)|$  的大小, 作为衡量  $x^*$  的精度高低比较直观, 而  $x^*$  的相对误差  $e_r(x^*)$  作为衡量  $x^*$  精度的好坏更为合理一些, 因为我们不仅要注意  $x^*$  的绝对误差的大小, 更要注意  $e(x^*)$  是在多大的  $x$  情况下发生的。由于准确值  $x$  在一般情况下是未知的, 因此  $e(x^*)$  和  $e_r(x^*)$  常常无法算出, 但有可能予以估计。

**定义 1.2.2** 设  $x^*$  是精确数  $x$  的一个近似数, 若有正数  $\epsilon$  和  $\epsilon_r$ , 满足

$$\begin{aligned} |e(x^*)| &= |x^* - x| < \epsilon \\ |e_r(x^*)| &= \frac{|x^* - x|}{|x|} < \epsilon_r \end{aligned} \quad (1.2.6)$$

则称  $\epsilon$  和  $\epsilon_r$  为近似数  $x^*$  的绝对误差界和相对误差界。

在实际计算绝对误差界和相对误差界时, 由于准确数  $x$  未知, 式(1.2.5)和式(1.2.6)常用

$$e_r(x^*) = \frac{e(x^*)}{x^*}$$

来代替。例如, 测得光速  $C = (2.997\ 925 \pm 0.000\ 001) \times 10^{10}$  cm/s, 则有绝对误差界  $\epsilon = 10^4$  cm/s, 而

$$|e_r(x^*)| \leq \frac{\epsilon}{C^*} = \frac{0.000\ 001}{2.997\ 925} < 4 \times 10^{-7}$$

即有相对误差界  $\epsilon_r = 4 \times 10^{-7}$ 。

**定义 1.2.3** 设近似数  $x^*$  有规格化形式

$$x^* = \pm 10^m \times 0.a_1 a_2 a_3 \cdots a_n \cdots$$

其中,  $m$  和  $a_i$  ( $i = 1, 2, \cdots, n, \cdots$ ) 是整数且  $a_1 \neq 0, 0 \leq a_i \leq 9$ 。如果  $x^*$  的绝对误差满足

$$|e(x^*)| = |x^* - x| \leq \frac{1}{2} \times 10^{m-n}$$

则称  $x^*$  为  $x$  的具有  $n$  位有效数字的近似数。

容易看出, 任何一个数  $x$  经过四舍五入到第  $n$  位所得的近似数  $x^*$ , 由于绝对

误差不超过第  $n$  位的半个单位, 即有  $\epsilon \leq \frac{1}{2} \times 10^{m-n}$ , 因此由定义 1.2.3 知, 凡四舍五入所得的具有  $n$  位的近似数  $x^*$  均为有效数字。

绝对误差、相对误差和有效数字是度量近似数精度常用的三种方式。实际算题时, 最终结果均以有效数字给出, 同时也就隐含地给出了绝对误差界和相对误差界。例如, 给出  $x = \sqrt{2}$  具有五位有效数字的近似数  $x^* = 1.4142$ , 则  $x^*$  的绝对误差界是总可取  $\epsilon = \frac{1}{2} \times 10^{-4}$ 。而相对误差界估计为

$$|e_r(x^*)| \leq \frac{\epsilon}{x^*} = \frac{\frac{1}{2} \times 10^{-4}}{1.4142} < 4 \times 10^{-5}$$

即有  $\epsilon_r = 4 \times 10^{-5}$ 。

### 1.2.3 函数值的误差估计

设函数  $y = f(x)$ , 当  $x$  是用近似数  $x^*$  代替计算函数值  $f(x^*)$  时, 则误差为

$$\begin{aligned} e(f) &= f(x^*) - f(x) \\ &\approx df(x^*) = f'(x^*)(x^* - x) = f'(x^*)e(x^*) \end{aligned}$$

或

$$e_r(f) \approx \frac{x^* f'(x^*)}{f(x^*)} e_r(x^*)$$

若记  $C = |f'(x^*)|$ ,  $C_r = \left| \frac{x^* f'(x^*)}{f(x^*)} \right|$ , 当  $C \leq 1$  及  $C_r \leq 1$  时, 有

$$|e(f)| \leq |e(x^*)| \quad (1.2.7)$$

$$|e_r(f)| \leq |e_r(x^*)| \quad (1.2.8)$$

式(1.2.7)及式(1.2.8)表明, 当  $C \leq 1$  及  $C_r \leq 1$  时, 函数值的绝对误差和相对误差不超过原始数据自变量的绝对误差和相对误差, 即函数值的误差可控制, 或称是稳定的。若  $C$  或  $C_r$  很大, 视不同问题而定, 函数值的误差常常得不到控制, 则称为病态或坏条件。由于数值  $C$  或  $C_r$  刻画了函数  $f(x)$  在点  $x$  邻域的数值性态是否稳定, 一般分别称  $C$ 、 $C_r$  为  $f(x)$  在绝对意义下和相对意义下的条件数。

#### 例 1.2.2 讨论二次函数

$$f(x) = x^2 + x - 10100$$

在其正根附近的性态。

**解** 二次函数正根  $x_1 = 100$ , 由于  $C = |f'(100)| = 201$ , 这个条件数较大, 因此该二次函数在正根附近是病态的。下面计算两具体数据可更清楚地看出这一点。

取  $x_1^* = 99$ , 则  $f(x_1^*) = f(99) = -200$ , 这与  $f(x_1) = 0$  相差甚远; 再取  $x_1^* =$

99.9, 则  $f(x_1^*) = f(99.9) = -20.09$ , 同样与  $f(x_1) = 0$  相差很大。也就是说, 自变量  $x_1$  仅发生微小变化, 函数值变化极大, 这是明显病态情况。

对于多元函数  $y = f(x_1, x_2, \dots, x_n) = f(\mathbf{x})$  的计算也有类似的讨论。设点  $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)^T$  为点  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  的近似值。用  $f(\mathbf{x}^*)$  代替  $f(\mathbf{x})$ , 其绝对误差为

$$e(f) = f(\mathbf{x}^*) - f(\mathbf{x}) \approx df(\mathbf{x}^*) = \sum_{i=1}^n \frac{\partial f(\mathbf{x}^*)}{\partial x_i} e(x_i^*)$$

因此, 绝对误差界有如下估计:

$$|e(f)| \leq \sum_{i=1}^n \left| \frac{\partial f(\mathbf{x}^*)}{\partial x_i} \right| |e(x_i^*)|$$

类似地, 相对误差和相对误差界为

$$e_r(f) \approx \sum_{i=1}^n \frac{\partial f(\mathbf{x}^*)}{\partial x_i} \frac{x_i^*}{f(\mathbf{x}^*)} e_r(x_i^*)$$

$$|e_r(f)| \leq \sum_{i=1}^n \left| \frac{\partial f(\mathbf{x}^*)}{\partial x_i} \frac{x_i^*}{f(\mathbf{x}^*)} \right| |e_r(x_i^*)|$$

**例 1.2.3** 设  $\triangle ABC$  观测数据为  $b = (100 \pm 0.10)\text{m}$ ,  $c = (120 \pm 0.10)\text{m}$ ,  $A = (60 \pm 0.02)^\circ$ , 试估计  $\triangle ABC$  面积  $S$  的绝对误差和相对误差。

**解** 由  $S = \frac{1}{2}bc \sin A$  得

$$|e(S)| \leq \left| \frac{\partial S}{\partial b} \right| |e(b^*)| + \left| \frac{\partial S}{\partial c} \right| |e(c^*)| + \left| \frac{\partial S}{\partial A} \right| |e(A^*)|$$

$$= \frac{1}{2} c^* \sin A^* \times 0.10 + \frac{1}{2} b^* \sin A^* \times 0.10 + \frac{1}{2} b^* c^* \cos A^* \times 0.02 \times \frac{\pi}{180}$$

$$= 10.57 \text{ (m}^2\text{)}$$

$$|e_r(S)| \leq \frac{|e(S)|}{S} = \frac{10.57}{\frac{1}{2} ab \sin A} = 2.035 \times 10^{-3}$$

### 1.3 算法的优化

同一个问题, 可以用几种不同的算法实现, 评价算法的优劣大致有以下标准。

第一, 从截断误差观点看, 一个算法必须是截断误差小, 收敛速度快, 即运算量小, 机器用时少。

第二, 从舍入误差观点看, 舍入误差在计算过程中要能控制, 即数值要稳定。

第三, 从实现算法观点看, 一个算法逻辑结构不宜太复杂, 要便于程序编写和上机实现。

下面举例加以说明。



**例 1.3.1** 计算  $\ln 2$  的值。

**解** 算法一:由 Taylor 展开式有

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots + (-1)^{n-1} \frac{x^n}{n} + \cdots$$

令  $x=1$  有

$$\ln 2 = 1 - \frac{1}{2} + \frac{1}{3} - \cdots + (-1)^{n-1} \frac{1}{n} + \cdots$$

若要截断误差不超过  $\epsilon = 10^{-5}$ , 则在上式中要取 10 万项以上的和。显然计算量较大, 敛速过慢, 再加上中间计算结果不可避免的舍入误差, 因此最终结果难以保证达到预期精度。

算法二: 由于

$$\ln \frac{1+x}{1-x} = 2x \left( 1 + \frac{1}{3}x^2 + \frac{1}{5}x^4 + \cdots + \frac{1}{2n+1}x^{2n} + \cdots \right) \quad (|x| < 1)$$

令  $x = \frac{1}{3}$ , 并取  $n=10$  有

$$\ln 2 \approx \frac{2}{3} \left( 1 + \frac{1}{3} \times \frac{1}{9} + \frac{1}{5} \times \frac{1}{9^2} + \cdots + \frac{1}{21} \times \frac{1}{9^{10}} \right)$$

其截断误差为

$$\begin{aligned} T &= \frac{2}{3} \left( \frac{1}{23} \times \frac{1}{9^{11}} + \frac{1}{25} \times \frac{1}{9^{12}} + \cdots \right) \\ &< \frac{2}{3} \times \frac{1}{23} \times \frac{1}{9^{11}} \left( 1 + \frac{1}{9} + \frac{1}{9^2} + \cdots \right) = 10^{-12} \end{aligned}$$

由此可见, 算法二仅取 10 项计算, 精度比算法一高得多, 敛速也极大提高。本例表明, 在理论上都是正确的算法, 在计算机上的有效性不尽相同。有许多问题, 减少计算量, 提高收敛速度, 降低舍入误差, 并不单纯是缩短机时, 甚至关系到算法能否真正在计算机上得到实现。

**例 1.3.2** 计算圆周率  $\pi$  的值。

**解** 算法一: 由定积分

$$\frac{\pi}{4} = \int_0^1 \frac{1}{1+x^2} dx = 1 - \frac{1}{3} + \frac{1}{5} - \cdots + (-1)^{n-1} \frac{1}{2n-1} + \cdots$$

类似地, 当要求  $\epsilon = 10^{-5}$  时, 要从上式取 5 万项以上, 缺点与例 1.3.1 算法一相同。

算法二: 取  $n=2, h = \frac{1}{2}$  的 Simpson 公式, 并记  $f(x) = \frac{1}{1+x^2}$  有

$$S_2 = \frac{1/2}{6} [f(0) + 4f(\frac{1}{4}) + 2f(\frac{1}{2}) + 4f(\frac{3}{4}) + f(1)] = 0.785\ 392\ 156$$

$$\pi^* = 4S_2 = 3.141\ 568\ 627$$

算法二表明, 仅用不多的五次函数值计算, 已获得  $\pi$  的具有五位有效数字的近