

地球物理数据处理基础

Fundamentals of Geophysical Data Processing

吕玉增 熊 彬 薛霆琥 编



地质出版社

地球物理数据处理基础

吕玉增 熊 彬 薛霆琥 编

地 质 出 版 社

· 北 京 ·

内 容 提 要

本书针对地球物理学、勘查技术与工程专业本科教学课程要求,系统地阐述了地学领域中常用的数值计算方法及其原理。主要内容包括:误差分析、向量范数和条件数、解线性方程组的直接法和迭代法、插值法、数值积分与微分、数字信号处理、数值模拟方法与反演等。

本书注重数据处理的基本思想与基本方法,强调可行性和实用性,书后配有若干基于 MATLAB 的源程序,便于理解和使用。

本书可作为高等地质院校地球物理学、勘查技术与工程专业本科生,地球探测与信息技术、固体地球物理学、应用地球物理学专业研究生的教材或参考书,也可供从事科学与工程计算的科技人员学习参考。

图书在版编目(CIP)数据

地球物理数据处理基础 / 吕玉增等编. —北京:
地质出版社, 2011. 8
ISBN 978-7-116-07209-1

I. ①地… II. ①吕… III. ①地球物理数据—数据处
理 IV. ①P631

中国版本图书馆 CIP 数据核字 (2011) 第 080386 号

DIQIU WULI SHUJU CHULI JICHU

责任编辑:王春庆

责任校对:黄苏晔

出版发行:地质出版社

社址邮编:北京海淀区学院路 31 号, 100083

电 话:(010) 82324508 (邮购部); (010) 82324514 (编辑部)

网 址: <http://www.gph.com.cn>

电子邮箱: zbs@gph.com.cn

传 真:(010) 82324340

印 刷:北京纪元彩艺印刷有限公司

开 本:787mm×1092mm 1/16

印 张:8.25

字 数:200 千字

印 数:1—1000 册

版 次:2011 年 8 月北京第 1 版

印 次:2011 年 8 月北京第 1 次印刷

定 价:16.00 元

书 号:ISBN 978-7-116-07209-1

(如对本书有建议或意见,敬请致电本社;如本书有印装问题,本社负责调换)

前 言

众所周知，地球物理勘探工作中存在大量的数据处理工作，在一定程度上来说，数据处理方法决定着数据处理的结果。

为了更好地了解和掌握地球物理数据处理的方法和计算技术，保证地球物理数据处理结果的可靠性，提高地球物理数据处理的解释精度，本书在专业课程讲授的基础上，对地球物理数据处理的各种方法进行了较为系统的介绍，包括地球物理数据处理方法的基本原理、计算技术及适用条件等。

《地球物理数据处理基础》是在已故的桂林理工大学原副校长阮百尧教授的亲自关怀和支持下，在桂林理工大学地球科学学院地球物理教研室阮百尧、梁锦文、沈云发、王有学等编写的《物探数据处理》讲稿的基础上，几经修改，由吕玉增、熊彬、薛霆虢等老师进行了充实和完善，最后由王有学教授主审完成。本书基本上涵盖了近几十年来地球物理数据处理与解释中用到的所有数值方法，紧密结合地球物理实际问题，比较系统地介绍了各种数值计算方法在地球物理问题中的具体应用，既注重普遍性，又不失针对性。基于地球物理数据处理的需要，本书注重算法的实用性，而且通俗易懂，可作为理工院校地球物理等相关专业学生的教材或参考书，也可供从事科学与工程计算的科技人员参考使用。

在本书的编写过程中，研究生李静和同学对书中介绍的算法在 MATLAB 平台上进行了编程及实例计算；研究生王海燕、杨强及王顺国同学对全书进行了统一和认真的校对，在此表示衷心的感谢！

由于编者水平有限，书中错误和缺点在所难免，恳请读者批评指正。

谨以此书献给英年早逝的阮百尧教授，以表达我们对他的深切怀念！

编 者

2011年4月于桂林

目 录

前 言	
第一章 地球物理数据处理的研究对象和特点	(1)
第二章 地球物理数据处理中的误差分析	(2)
第一节 误差的基本概念	(2)
一、误差与误差限	(2)
二、相对误差与相对误差限	(2)
三、有效数字	(3)
四、数值运算的误差估计	(3)
第二节 误差分析的重要性	(5)
第三节 数值计算中控制误差的若干原则	(6)
第三章 向量范数、矩阵范数和条件数	(9)
第一节 向量范数	(9)
一、范数的定义	(9)
二、向量范数的性质	(9)
第二节 矩阵范数	(10)
一、矩阵范数的定义	(10)
二、矩阵范数的性质	(11)
第三节 矩阵的条件数	(11)
第四章 解线性方程组的分解法(直接法)	(14)
第一节 直接三角分解法	(14)
一、基本思想	(14)
二、矩阵 A 的 LU 分解	(15)
三、三角分解的存在性和唯一性	(18)
四、数值稳定性	(19)
五、特点与应用	(20)
第二节 列主元 LU 分解法	(20)
一、基本思想	(20)
二、算法	(21)
第三节 解对称方程组的平方根法和改进的平方根法	(23)
第四节 带状矩阵分解及追赶法	(26)
第五章 解线性方程组的迭代法	(29)
第一节 雅可比(Jacobi)迭代法	(29)
一、雅可比迭代法的计算公式	(30)

二、迭代矩阵与收敛定理	(31)
第二节 高斯-赛德尔 (Gauss-Seidel) 迭代法	(32)
一、G-S 迭代法的计算公式	(32)
二、迭代矩阵收敛定理	(33)
第三节 松弛迭代法	(33)
一、迭代计算式	(33)
二、迭代矩阵及收敛定理	(34)
第六章 插值法	(36)
第一节 拉格朗日插值	(36)
一、线性插值	(36)
二、二次插值	(37)
三、 n 次 Lagrange 插值	(38)
四、Lagrange 多项式插值误差及其稳定性	(40)
第二节 三次样条插值	(41)
一、基本思想	(41)
二、边界条件	(42)
三、样条函数的构造	(43)
四、三次样条插值计算步骤	(46)
第三节 二维插值	(46)
一、线性插值	(46)
二、二次插值	(48)
三、三次插值	(48)
四、应用举例	(49)
第七章 数值积分与微分	(51)
第一节 数值积分的基本概念	(51)
一、数值积分的基本思想	(51)
二、数值积分的精度及其稳定性	(52)
第二节 复化求积	(53)
一、复化梯形求积	(53)
二、复化辛卜生求积	(55)
第三节 插值求导和样条求导	(57)
一、插值求导法	(57)
二、辛卜生积分方程求导法	(58)
三、三次样条插值求导法	(59)
第八章 数字信号处理	(60)
第一节 快速傅里叶变换	(60)
一、傅里叶变换及其性质	(60)
二、一维复数序列的快速傅里叶变换 (FFT)	(61)
三、一维实序列的快速傅里叶变换 (FFT)	(66)

四、二维实序列的快速傅里叶变换 (FFT)	(70)
第二节 褶积、相关和 z 变换	(75)
一、褶积	(75)
二、相关	(81)
三、 z 变换和逆 z 变换	(83)
四、反褶积	(87)
第三节 拉普拉斯变换	(90)
一、问题的提出	(90)
二、拉普拉斯变换	(91)
三、拉普拉斯变换的性质	(91)
第九章 数值模拟方法与反演	(93)
第一节 有限差分法	(93)
一、差分方程的导出	(93)
二、边界条件的处理	(95)
第二节 有限单元法	(95)
一、直流线源二维地电断面的边值问题	(96)
二、直流线源二维地电断面的变分问题	(96)
三、有限单元法	(96)
第三节 边界单元法	(98)
一、二维位场延拓边值问题	(99)
二、二维位场延拓问题的基本解	(99)
三、二维延拓问题的边界积分方程	(100)
四、边界积分方程	(100)
五、边界单元法	(100)
六、延拓计算	(103)
第四节 地球物理反演方法	(103)
一、“阮氏”直接反演	(103)
二、最小二乘最优化反演方法	(106)
参考文献	(110)
附 录	(111)

第一章 地球物理数据处理的研究对象和特点

地球物理勘查数据处理就是利用计算机技术分析、处理地球物理数据。在地球物理资料的数据处理中,要解决比较复杂的理论和技术问题时,通常要进行微积分运算和微分方程求解等数学过程。这些数学问题在计算机上无法用经典微积分和普通代数等方法解决,只能通过数值方法解决。因此,数值计算方法是数据处理中的主要内容,是利用数学手段解决地球物理数据处理的桥梁,是一门十分重要的地球物理专业基础课。

一般来说,地球物理数据处理流程如图 1-1 所示。

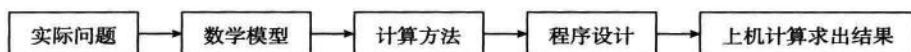


图 1-1 地球物理数据处理流程

由实际问题的提出,利用专业理论和数学理论相结合的方法建立数学模型,这就是建模过程;由建立的数学模型出发,选择最佳计算方法,这就是数值分析方法;根据计算方法编写程序,利用计算机计算求出结果,这就是数据处理过程。对于地球物理专业来说,它是重力、磁法、电法和地震勘探等方法的资料解释的基础。这就要求我们熟练掌握数值计算方法的理论和方法,充分利用电子计算机技术,为地球物理资料的解释提供可靠的数值依据。

在数值计算过程中,如积分、微分与无穷级数求和等,是无穷过程的运算,属于精确运算。然而在计算机中,只能进行有限项数和有限位数的四则运算,属于近似计算。

数值方法就是为解决各类数学运算而设计的,并能达到所需精度的算法。与经典数学只研究数学本身的理论不同,数值方法还有如下特点:

第一,计算机能直接处理的数值解法。把连续变量问题转化为离散数学问题,并根据计算机的特点,使算法的计算量尽可能地少,从而节省机时。

第二,要有可靠性分析。近似计算不可避免地存在误差,关键是如何把它控制在允许的误差范围之内,因此还要研究与解决算法的收敛性、稳定性及误差估计等问题。

第三,要具体问题具体分析。尽管各类数值方法已很丰富,但它们都不是万能的,不是在任何情况下都是最优的。因此,在选用解决具体问题的算法时,通常要进行数值试验。

第二章 地球物理数据处理中的误差分析

第一节 误差的基本概念

一、误差与误差限

记 x^* 为准确值, x 为 x^* 的一个近似值, 称 $e^* = x - x^*$ 为近似值的绝对误差, 简称误差。 $-e^*$ 为 x 的修正值。故 $x^* = x - e^*$ 表示近似值去掉它的误差, 或加上它的修正值, 就是准确值。

通常我们不能算出准确值 x^* , 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 称为近似值的误差限, 它满足

$$|e^*| = |x - x^*| \leq \epsilon \quad (2-1)$$

一般取 ϵ 为某位数字的半个单位。

常用 $x^* = x \pm \epsilon$ 表示近似值 x 的精确度。如果用毫米刻度的米尺量得桌面长为 1235mm, 因为所用米尺测量误差不超过半毫米, 所以桌面实际长度为 $x = (1235 \pm 0.5) \text{ mm}$ 。

二、相对误差与相对误差限

绝对误差看起来很直观, 但有时不能确切地反映出某种测量的精确程度。例如, 第一次测量 100m 的距离时, 产生的绝对误差为 1m, 第二次测量 1000m 的距离时, 产生的绝对误差为 5m, 那么, 这两次测量哪一次比较精确呢? 比较科学的对比方法, 就是要考察在两次测量当中, 平均每测量 1m 能产生多大的误差。很显然, 前后两次测量中每米产生的误差分别为 1% 与 0.5%, 可见, 第二次测量绝对误差是前次的 5 倍, 但相对误差仅是前次的一半, 相比还是第二次的测量结果比较精确。

在理论上, 绝对误差与准确值之比称为相对误差, 记为 e_r^* , 有

$$e_r^* = \frac{e^*}{x^*} = \frac{(x - x^*)}{x^*} \quad (2-2)$$

在实际计算中, 真值 x^* 总是不知道的, 故通常取

$$e_r^* = \frac{e^*}{x} = \frac{(x - x^*)}{x} \quad (2-3)$$

作为 x 的相对误差, 条件是 e_r^* 较小。

相对误差可正可负, 它的绝对值上界称为相对误差限, 记作

$$\epsilon_r = \frac{\epsilon}{|x|} \quad (2-4)$$

三、有效数字

当准确值 x^* 有多位数时, 常按四舍五入的原则得到 x^* 的有效近似值 x , 例如: $x^* = \pi = 3.14159265\dots$, 取 3 位有效数字, $x_3 = 3.14$, $\epsilon_3 \leq 0.002$; 取 5 位有效数字, $x_5 = 3.1416$, $\epsilon_5 \leq 0.000008$; 它们的误差都不超过末位数字的半个单位, 即 $|\pi - 3.14| \leq \frac{1}{2} \times 10^{-2}$, $|\pi - 3.1416| \leq \frac{1}{2} \times 10^{-4}$ 。

若近似值 x 的误差限是末位的半个单位, 该位到 x 的第一位非零数字共有 n 位, 我们就说 x 有 n 位有效数字。如取 $x = 3.14$ 作 π 的近似值, x 就有 3 位有效数字; 取 $x = 3.1416$ 作 π 近似值, x 就有 5 位有效数字。

x 有 n 位有效数字可写成标准形式:

$$x = \pm 10^m \times [a_1 + a_2 \times 10^{-1} + \dots + a_n \times 10^{-(n-1)}] \quad (2-5)$$

其中: a_1 是 1 到 9 中的一个整数; a_2, \dots, a_n 是 0 到 9 中的一个整数; m 为整数, 且

$$|x - x^*| \leq \frac{1}{2} \times 10^{m-n+1} \quad (2-6)$$

显然, 其绝对误差限为

$$\epsilon = \frac{1}{2} \times 10^{m-n+1} \quad (2-7)$$

在 m 相同的情况下, n 越大, 则 10^{m-n+1} 越小, 故有效位数越多, 绝对误差限越小。

至于有效数字与相对误差限的关系, 有

$$\epsilon_r \leq \frac{1}{2a_1} \times 10^{-(n-1)} \quad (2-8)$$

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

要使 $\sqrt{20}$ 的近似值的相对误差小于 0.1%, 要取几位有效数字? 根据上述原理, $\epsilon_r \leq \frac{1}{2a_1} \times 10^{-n+1}$, 由于 $\sqrt{20} = 4.4\dots$, 知 $a_1 = 4$, 故只要取 $n = 4$, 就有: $\epsilon_r \leq 0.125 \times 10^{-3} < 10^{-3} = 0.1\%$, 即只要对 $\sqrt{20}$ 的近似值取 4 位有效数字, 其相对误差限就小于 0.1%, 此时由开方表查得: $\sqrt{20} \approx 4.472$ 。

四、数值运算的误差估计

两个近似数 x_1 与 x_2 , 其误差限分别为 $\epsilon(x_1)$ 及 $\epsilon(x_2)$ 。它们进行加、减、乘、除运算得到的误差限分别为

$$\epsilon(x_1 \pm x_2) = \epsilon(x_1) \pm \epsilon(x_2) \quad (2-9)$$

$$\epsilon(x_1 \cdot x_2) \approx |x_1| \cdot \epsilon(x_2) + |x_2| \cdot \epsilon(x_1) \quad (2-10)$$

$$\epsilon\left(\frac{x_1}{x_2}\right) \approx \frac{|x_2| \cdot \epsilon(x_1) + |x_1| \cdot \epsilon(x_2)}{|x_2|^2} \quad (x_2 \neq 0) \quad (2-11)$$

更一般情况是, 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泰勒级数展开式进行估计。设 $f(x^*)$ 是一元函数, x^* 的近似值为 x , 以 $f(x)$ 近似 $f(x^*)$, 其误差限记作 $\epsilon[f(x)]$, 可用泰勒级数展开, 即

$$f(x^*) - f(x) = f'(x)(x^* - x) + \frac{f''(\xi)}{2}(x^* - x)^2 \quad (2-12)$$

ξ 介于 x, x^* 之间, 对上式取绝对值得

$$|f(x^*) - f(x)| \leq |f'(x)| \cdot \epsilon(x) + \frac{|f''(\xi)|}{2} \cdot \epsilon^2(x) \quad (2-13)$$

若 $\epsilon(x)$ 值很小, 可忽略 $\epsilon(x)$ 的高阶项, 于是可得到计算函数的误差限

$$\epsilon[f(x)] \approx |f'(x)| \cdot \epsilon(x) \quad (2-14)$$

当 f 为多元函数时, 如计算 $A^* = f(x_1^*, x_2^*, \dots, x_n^*)$, 如果 $x_1^*, x_2^*, \dots, x_n^*$ 的近似值为 x_1, x_2, \dots, x_n , 则 A^* 的近似值为 $A = f(x_1, x_2, \dots, x_n)$, 于是函数值 A 的误差 $e(A)$, 由泰勒级数展开得

$$\begin{aligned} e(A) &= A - A^* = f(x_1, x_2, \dots, x_n) - f(x_1^*, x_2^*, \dots, x_n^*) \\ &\approx \sum_{k=1}^n \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_k} \cdot (x_k - x_k^*) \\ &\approx \sum_{k=1}^n \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_k} \cdot e(x_k) \end{aligned} \quad (2-15)$$

于是误差限

$$\epsilon(A) = \sum_{k=1}^n \left| \frac{\partial f}{\partial x_k} \right| \cdot \epsilon(x_k) \quad (2-16)$$

而 A 的相对误差限为

$$\epsilon_r(A) = \frac{\epsilon(A)}{A} \approx \sum_{k=1}^n \left| \frac{\partial f}{\partial x_k} \right| \cdot \frac{\epsilon(x_k)}{|A|} \quad (2-17)$$

这里举一个简单的例子, 例如测量某场地长 $l=110\text{m}$, 宽 $d=80\text{m}$, 已知 $|l-l^*| \leq 0.2\text{m}$, $|d-d^*| \leq 0.1\text{m}$, 试求面积 $S=ld$ 的绝对误差限和相对误差限。

解: 因 $S=ld$, $\frac{\partial S}{\partial l} = d$, $\frac{\partial S}{\partial d} = l$ 由式(2-16)可得

$$\epsilon(S) \approx \left| \frac{\partial S}{\partial l} \right| \epsilon(l) + \left| \frac{\partial S}{\partial d} \right| \epsilon(d) = d \cdot \epsilon(l) + l \cdot \epsilon(d)$$

其中, $\left| \frac{\partial S}{\partial l} \right| = d = 80\text{m}$, $\left| \frac{\partial S}{\partial d} \right| = l = 110\text{m}$, 而 $\epsilon(l) = 0.2\text{m}$, $\epsilon(d) = 0.1\text{m}$, 于是

绝对误差限为: $\epsilon(S) \approx 80 \times 0.2 + 110 \times 0.1 = 27\text{m}^2$,

相对误差限为: $\epsilon_r(S) = \frac{\epsilon(S)}{|S|} = \frac{\epsilon(S)}{l \cdot d} \approx \frac{27}{8800} = 0.31\%$ 。

第二节 误差分析的重要性

在数值分析中,一般不讨论“模型误差”和“观测误差”。数值分析只研究用数学方法求解数学模型产生的误差,主要包括算法的截断误差和舍入误差。

当数学模型不能得到精确解时,通常要用数值方法求它的近似解,其近似解与精确解之间的误差称为截断误差或方法误差。如果用泰勒多项式 $P_n(x)$ 近似代替函数 $f(x)$, 即

$$P_n(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \cdots + \frac{f^{(n)}(\xi)}{n!}x^n \quad (2-18)$$

则数值方法的截断误差是

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}x^{n+1} \quad (2-19)$$

可见截断误差与算法的收敛性有关。如用泰勒多项式 $P_n(x)$ 代替函数 $f(x)$, 计算时要确定 n 。若 n 越大误差越小,则算法是收敛的;反之,若随着 n 的增大,截断误差随之增大,则称算法是发散的。

在用计算机进行数值计算时,由于计算机内存的字长有限,原始数据在计算机内部的储存会产生误差,计算过程又可能产生新的误差,这种误差称为舍入误差。例如,用 3.14159 近似代替 π , 产生的误差 $R = \pi - 3.14159 = 0.000026\cdots$, R 就是舍入误差。在大型计算中运算次数太多,舍入误差一般很难估计,为此引入算法稳定性的概念。我们把运算过程舍入误差不增长的计算公式称为数值稳定的。这样,对于数值稳定的计算过程,可以不去估计具体的舍入误差,只是分析计算过程的数值稳定性,因而对运算过程中的误差积累问题进行定性分析是有重要意义的。

由此可见,在数值分析中除了研究数学问题的算法外,还要研究计算结果的误差是否满足精度要求,这就是误差估计问题。

下面举例说明误差分析的重要性。

[例] 计算积分 $I_n = e^{-1} \int_0^1 x^n e^x dx, n = 0, 1, \cdots$, 并估计误差。

解: 由分部积分法可得计算 I_n 的递推公式:

$$I_n = 1 - nI_{n-1} \quad (2-20)$$

(1) 算法 A (顺序递推法)

首先算出 $I_0 = e^{-1} \int_0^1 e^x dx = 1 - e^{-1}$, 即要算出 e^{-1} , 可用泰勒多项式展开, 取 $n=7$ 以前各项的和, 并且精确到 4 位小数, 计算得到 $e^{-1} \approx 0.3679$ 。

截断误差 $R_7 = |e^{-1} - 0.3679| \leq \frac{1}{8!} < \frac{1}{4} \times 10^{-4}$ 。计算过程中小数点后的第 5 位数字

按四舍五入原则取舍, 由此产生的舍入误差不作讨论。当取初值为 $I_0 \approx 0.6321 = \bar{I}_0$ 时, 递推公式为

$$\begin{cases} \bar{I}_0 = 0.6321 \\ \bar{I}_n = 1 - n\bar{I}_{n-1} \end{cases} \quad (n = 1, 2, \cdots) \quad (2-21)$$

计算结果如表 2-1 的 \bar{I}_n 列, 即

表 2-1 两种算法计算结果

n	\bar{I}_n	I_n	n	\bar{I}_n	I_n
0	0.6321	0.6321	5	0.1460	0.1455
1	0.3679	0.3679	6	0.1120	0.1268
2	0.2642	0.2643	7	0.2160	0.1121
3	0.2074	0.2073	8	-0.728	0.1035
4	0.1704	0.1708	9	7.552	0.0684

利用积分估计可知 $\frac{e^{-1}}{n+1} < I_n < \frac{1}{n+1}$, 而表 2-1 中 \bar{I}_8 出现负值, 显然是不正确的。这里计算公式与每步计算都是正确的, 那么什么原因使计算结果错误呢? 主要是初值 \bar{I}_0 有误差 $e_0 = \bar{I}_0 - I_0$, 由此引起以后各步计算的误差为

$$|e_n| = |\bar{I}_n - I_n| = |(1 - n\bar{I}_{n-1}) - (1 - nI_{n-1})| = n |\bar{I}_{n-1} - nI_{n-1}| = n |e_{n-1}|$$

容易推出

$$|e_n| = n! |e_0| \quad (2-22)$$

这说明 \bar{I}_0 有误差 e_0 , 则 \bar{I}_n 就有 e_0 的 $n!$ 倍误差。例如 $n=8$, 若 $|e_0| = \frac{1}{2} \times 10^{-4}$, 则 $|e_8| = 8! \times |e_0| > 2$, 这就说明 \bar{I}_8 完全不能近似 I_8 了。

(2) 算法 B (逆序递推法)

现在换一种计算方案, 取 $n=9$, 得

$$\frac{e^{-1}}{10} < I_9 < \frac{1}{10} \quad (2-23)$$

我们粗略取 $I_9 \approx \frac{1}{2} \left(\frac{e^{-1}}{10} + \frac{1}{10} \right) = 0.0684 = \bar{I}_9$, 然后得倒序递推公式

$$\begin{cases} \bar{I}_9 = 0.0684 \\ \bar{I}_{n-1} = \frac{(1 - \bar{I}_n)}{n} \end{cases} \quad (n = 9, 8, \dots, 1) \quad (2-24)$$

计算结果见表 2-1 的 \bar{I}_n 列。我们发现 \bar{I}_0 与 I_0 的误差不超过 10^{-4} 。由于 $|\bar{e}| = \frac{1}{n!} |\bar{e}_n|$, \bar{e}_0 比 \bar{e}_n 缩小了 $n!$ 倍。因此, 尽管 \bar{e}_9 较大, 但由于误差逐步缩小, 故可用 \bar{I}_n 近似 I_n , 此例说明我们应重视计算过程中的误差分析。

第三节 数值计算中控制误差的若干原则

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进行除法运算时, 舍入误差会增大。如计算 x/y 时, 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] 线性方程组

$$\begin{cases} 0.00001x_1 + x_2 = 1 \\ 2x_1 + x_2 = 2 \end{cases} \quad (2-25)$$

其准确解为

$$\begin{aligned} x_1 &= \frac{200000}{399999} = 0.50000125 \\ x_2 &= \frac{199998}{199999} = 0.999995 \end{aligned}$$

在四位浮点十进制数（仿机器实际计算）下，用消去法求解，上述方程可写成

$$\begin{cases} 10^{-4} \times 0.1000x_1 + 10^1 \times 0.1000x_2 = 10^1 \times 0.1000 \\ 10^1 \times 0.2000x_1 + 10^1 \times 0.1000x_2 = 10^1 \times 0.2000 \end{cases} \quad (2-26)$$

若用 $\frac{1}{2}$ ($10^{-4} \times 0.1000$) 除第一方程减第二方程，则出现用小的数除大的数，得到

$$\begin{cases} 10^{-4} \times 0.1000x_1 + 10^1 \times 0.1000x_2 = 10^1 \times 0.1000 \\ 10^6 \times 0.2000x_2 = 10^6 \times 0.2000 \end{cases} \quad (2-27)$$

由此解出

$$x_1 = 0, x_2 = 10^1 \times 0.1000 = 1, \text{ 显然结果严重失真。}$$

若反过来用第二个方程消去第一个方程中含 x_1 的项，则避免了小除数、大乘数，得到

$$\begin{cases} 10^6 \times 0.2000x_2 = 10^6 \times 0.2000 \\ 10^1 \times 0.2000x_1 + 10^1 \times 0.1000x_2 = 10^1 \times 0.2000 \end{cases} \quad (2-28)$$

由此求得相当好的近似解

$$x_1 = 0.5000, x_2 = 10^1 \times 0.1000 = 1$$

2. 要避免两相近数相减

在数值计算中两个相近的数相减有效数字会严重损失，例如 $X=532.65$, $Y=532.52$ 都是有五位有效数字，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。现举例说明。

[例] 计算 $A=10^7 [1-\cos 2^\circ]$ （四位数学表）

由于 $\cos 2^\circ = 0.9994$ ，直接计算则有

$$A = 10^7 [1 - \cos 2^\circ] = 10^7 (1 - 0.9994) = 6 \times 10^3 \quad (2-29)$$

其结果只有一位有效数字；若用 $1 - \cos x = 2 \sin^2 \frac{x}{2}$ ， $\sin 1^\circ = 0.0175$ ，则

$$A = 10^7 [1 - \cos 2^\circ] = 10^7 \times 2 \sin^2 1^\circ = 6.13 \times 10^3 \quad (2-30)$$

其结果则具有三位有效数字。此例说明，可通过改变计算公式避免或减少有效数字的损失。

类似的，如果 x_1 和 x_2 很接近时，则 $\lg x_1 - \lg x_2 = \lg \left(\frac{x_1}{x_2} \right)$ ，用右边算式，有效数字就不会损失。

当 x 很大时:

$$\begin{aligned}\sqrt{x+1}-\sqrt{x} &= \frac{1}{\sqrt{x+1}+\sqrt{x}} \\ \arctan(x+1)-\arctan x &= \frac{1}{1+x(x+1)}\end{aligned}\quad (2-31)$$

都用右端算式代替左端。如果无法改变算式,则采用双字长运算,但这会增加计算时间和内存占有量。

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,如不注意就会出现大数“吃掉”小数。

[例] 求解 $x^2 - (10^{12} + 1)x + 10^{12} = 0$

解:由因式分解易知准确解为 $x_1 = 10^{12}$, $x_2 = 1$ 。若用 10 位有效数字的计算机,按下式计算,即

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2-32)$$

计算即得 $x_1 = 10^{12}$, $x_2 = 0$ 。错误原因反映在公式中 $-b = 10^{12} + 1 \triangleq 10^{12}$, 其中“ \triangleq ”表示计算机中的相等。这是因为在计算机内计算时要对阶,即

$$10^{12} + 1 = 0.1000000000 \times 10^{13} + 0.0000000000001 \times 10^{13} \triangleq 10^{12} \quad (2-33)$$

把小数吃了。

同理,因为 $b^2 - 4ac \triangleq b^2$, $\sqrt{b^2 - 4ac} \triangleq |b|$, 故

$$x_1 \triangleq \frac{10^{12} + 10^{12}}{2} = 10^{12}, \quad x_2 \triangleq \frac{10^{12} - 10^{12}}{2} = 0$$

4. 注意简化计算步骤,减少运算次数

简化计算步骤,减少运算次数不但可节省计算时间,而且还能减少舍入误差。这是数值计算必须遵从的原则。

[例] 计算 x^{255} 的值,若逐个相乘,需用 254 次乘法,若按各次平方计算,即

$$x^{255} = x \cdot x^2 \cdot x^4 \cdot x^8 \cdot x^{16} \cdot x^{32} \cdot x^{64} \cdot x^{128}$$

只要做 14 次乘法运算即可。

第三章 向量范数、矩阵范数和条件数

在各种计算方法中讨论误差时，例如在研究线性方程组近似解的误差估计和迭代法的收敛性时，我们需要对 R^n (n 维向量空间) 中向量或 $R^{n \times n}$ 中矩阵的“大小”引进某种度量——向量范数或矩阵范数。向量范数概念是三维欧氏空间中向量长度概念的推广，在数值分析中起着重要作用。

第一节 向量范数

一、范数的定义

一般来说，向量的范数定义为

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}} \quad (p \geq 1) \quad (3-1)$$

称为向量 x 的 p 范数，简称范数。式中 p 为不小于 1 的任意实数。常用的有 $p=1, 2, \infty$ 三种范数。

$$1. \|x\|_1 = \sum_{i=1}^n |x_i| \quad (3-2)$$

称为“1-范数”、“ L_1 范数”或“和范数”。

$$2. \|x\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}} = (x, x)^{\frac{1}{2}} \quad (3-3)$$

称为“2-范数”、“ L_2 范数”或“欧氏范数”，也就是所谓的向量长度。上式中 (x, x) 表示向量 x 的内积。

$$3. \|x\|_\infty = \max_{1 \leq i \leq n} |x_i| \quad (3-4)$$

称为“ ∞ 范数”、“最大值范数”或“切贝舍夫范数”。这里“最大值”的含义应理解为指向 x 的绝对值最大分量或 p 最大，而不是范数最大，事实上

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \quad (3-5)$$

二、向量范数的性质

1. 非负性（正定性）

$x \neq 0$ （零向量）时， $\|x\| > 0$

2. 齐次性

$$\|ax\| = |a| \cdot \|x\| \quad (a \text{ 为任意复数})$$

3. 三角不等式

$$\|x+y\| \leq \|x\| + \|y\|, \quad \|x-y\| \geq \|x\| - \|y\|$$

[例] 计算向量 $x = (1, -2, 3)^T$ 的范数

解: $\|x\|_1 = 1 + |-2| + 3 = 6$

$$\|x\|_2 = \sqrt{1^2 + (-2)^2 + 3^2} = \sqrt{14} = 3.7417$$

$$\|x\|_\infty = 3$$

由此可见, 向量的范数是一个具体的数, 不是一个向量, 它们是从不同角度反映向量的大小和长度, 通常可用符号 $N(x)$ 或 $\text{Norm}(x)$ 表示 x 的范数。

第二节 矩阵范数

一、矩阵范数的定义

1. 第一种定义—— p 范数

设 $\{x: \|x\|_p = 1\}$, 则对于给定的 n 阶矩阵 A 存在相应的向量集合 $\{Ax: \|x\|_p = 1\}$, 定义

$$\|A\|_p = \max_{\|x\|_p=1} \|Ax\|_p \quad (3-6)$$

为从属于某种向量范数的矩阵范数, 简称从属范数。因为是通过向量 p 范数定义的矩阵范数, 也称 p 范数或算子范数。

由定义可知, $\|x\|_p$ 的含义是向量集合 $\{Ax: \|x\|_p = 1\}$ 中各向量都有一个对应的范数, 其中最大的就是 $\|x\|_p$ 。可见, 这是具体的定义方法, 其优点可以把矩阵范数与向量范数用一个公式联系起来, 使两者相容。由于单位矩阵 $\|I\|_p = \max_{\|x\|_p=1} \|Ix\|_p = \max_{\|x\|_p=1} \|x\|_p = 1$, 所以 $\|I\|_p = 1$ 是从属范数的一个特征。

矩阵范数有分别从属于 $\|x\|_1$, $\|x\|_2$ 和 $\|x\|_\infty$ 的三种具体形式。

$$(1) \|A\|_1 = \max_{\|x\|_1=1} \|Ax\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| \quad (3-7)$$

称为列和范数 (矩阵各列向量的“和范数”中最大者)。

$$(2) \|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \sqrt{\lambda_1(A^T A)} \quad (3-8)$$

称为 A 的 2-范数。式中 $A^T A$ 称为格兰姆矩阵, $\lambda_1(A^T A)$ 表示 $A^T A$ 的最大特征值。因为 $\lambda_1(A^T A)$ 就是 $A^T A$ 的谱半径, 又称 $\|A\|_2$ 是 A 的谱范数。

$$(3) \|A\|_\infty = \max_{\|x\|_\infty=1} \|Ax\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| \quad (3-9)$$

称为行和范数 (矩阵各行向量“和范数”中的最大者)。

2. 第二种定义—— F 范数

定义:

$$\|A\|_F = \left(\sum_{i,j=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}} \quad (3-10)$$

为 A 的 F 范数 (Frobenius), 这是将向量 2-范数概念直接推广到 $R^{n \times n}$ 空间中的范数, 也可以把这种范数记为 $\|A\|_E$, 称为欧氏范数。

注意 F 范数不从属于任何一种向量范数, 它与 $\|x\|_2$ 相容, 但并不从属于 $\|x\|_2$ 。这时, 单位阵 I 的任一种从属范数 $\|I\|_p = 1$, 但 $\|I\|_F = \left(\sum_{i,j=1}^n 1 \right)^{\frac{1}{2}} = \sqrt{n}$ 。