数值分析引论

[德]J. Stoer　R. Bulirsch　著
孙文瑜　等　译
何旭初　等　校

南京大学出版社

数值分析引论

〔德〕J. Stoer R. Bulirsch 著
孙文瑜 何炳生 赵金熙
盛松柏 欧阳梓祥 译
何旭初 张元继 黄开斌 审校

南京大学出版社
1995·南京

(苏)新登字 011 号

J. Stoer, R. Bulirsch
Introduction to Numerical Analysis

1980,1993 by Springer -Verlag New York Inc.
Printed in the United States of America

数值分析引论

〔德〕 J. Stoer R.Bulirsch 著

孙文瑜 等译

何旭初 等校

*

南京大学出版社出版
(南京大学校内 邮政编码：210093)
江苏省新华书店发行 丹阳市新华印刷厂印刷

*

开本 850×1168 1/32 印张 22.75 字数 591 千
1995 年 9 月第 1 版 1995 年 9 月第 1 次印刷
印数 1—1000
ISBN 7-305-02610-7/O·186
定价：28.50元

中译本前言

这本数值分析引论是作者最初为德国高等学校需要而编写的，现在也已经在非德语国家被广泛使用。这个中译本是以最新的德文版本为基础，同时也在某些章节参考英文版本翻译而成的。

如果这个译本能够第一次被欧美以外的广大中国读者使用，作者将因为他们的著作受到尊重而感到莫大荣幸。

南京大学数学系何旭初教授和他的同事们为翻译这部书做了许多工作。除何旭初先生以外，我们要特别感谢孙文瑜博士、何炳生博士和所有他们的同事在翻译工作的各个方面给予的有力支持。我们对他们的劳动成果表示祝贺，并希望这部中译本能够受到广大中国读者的青睐。

最后，作者对南京大学出版社在出版过程中提供的合作表示感谢。

约·斯托尔

罗·波里希

于德国维尔茨堡和慕尼黑

1989年12月

译 者 的 话

Stoer教授和Bulirsch教授的著作《数值分析引论》是国际上计算数学方面颇有影响的重要著作，是各国大学计算数学专业广为采用的教材和教学参考书，已被译成七八种文字。这本《数值分析引论》中文版是根据该书英文版(1980,1993)和最新德文第五版(1990年版)译出的。其中，第一、二章由盛松柏翻译，第三、四章由孙文瑜翻译，第五、八章由欧阳梓祥翻译，第六章由赵金熙翻译，第七章由何炳生翻译。全书由孙文瑜统一组织和润色加工。已故何旭初教授对本书翻译给予了热情支持，Stoer教授对翻译工作给予了很大鼓励和关心。黄开斌教授和张元继教授作了一丝不苟的审核。南京大学出版社秦涛副社长等同志为本书及时出版作出了很大努力。译者仅在此表示感谢。

译 者

1993.5.1

英译本前言

这本书是根据作者在德国和美国几所大学讲授数值分析引论(一学年课程)的材料编写的。作者的注意力集中于那些在计算机上可以算出来的方法。对于重要的课题，我们或多或少以ALGOL 60语言正式地描述了算法，并详尽而简明地论述了理论基础。对于求解某一个问题，我们提供了若干种方法，并对这些方法的适应性和限制性作了比较。每一个比较都是根据以下几个方面作出的：即运算量、理论性质(例如收敛速度)、以及代表算法可靠性与不可靠性的内在数值性质(这一点是更重要的)。

就上下文关系而言，开头一章误差分析起着特殊作用，因为它明确描述了一些基本概念，例如算法的数值稳定性，这在详细研究数值问题中是不可缺少的。

其余七章致力于描述各种数值方法，它们除了包括一般课题外，还包含在通常数值分析的引论性教材中不介绍的一些特殊课题。在讨论插值法的第二章中，我们介绍了现代快速Fourier变换方法。在第三章中为了加速离散方法的收敛性，与Romberg迭代相联系的外推法我们详细阐述了。

在解线性方程组的第四章中，我们描述了解线性规划单纯形法的一个数值稳定的实现方法。第五章研究解非线性方程组，也进而讨论了解无约束极小化问题的算法。

第六章介绍矩阵特征值问题。第七章考虑解常微分方程的方法。这一章广泛讨论了解两点边值问题的现代多重打靶法。相反，解偏微分方程的方法没有系统研究，目的仅仅是指出解常微分方程的某些方法的类似，例如差分方法和变分方法。最后一章致力于讨论解大稀疏线性方程组的特殊方法，这些方程组主要来自应

用差分方法或有限元方法解偏微分方程。除了迭代法以外，我们还描述了 Hestenes-Stiefel 共轭梯度法，Buneman 算法(它提供了解离散Poisson 问题的一个现代直接法的例子)。

在每一章内，我们给出了若干例子和习题用以解释各个方法的数值和理论性质，每一章最后都安排了详尽的参考文献。

作者感谢所有那些为本书出版作出贡献的人们。首先，作者非常感激 F. L.Bauer 的早期讲义对我们写作的深刻影响；许多同事帮助我们仔细阅读手稿，并提出了很多有用的建议。在其余的人中间，我们想感谢 C.Reinsch, M.B. Spijker，特别，对于这本英文版的译者R.Bartels, W.Gautschi 和 C.Witzgall 的不倦工作深表谢意。和我们一起工作的 K.Butendeich, G. Schuller, J.Zowe和I.Brugger 帮助我们准备了本书原先的德文版。最后，但并非最不重要的，我们向 Springer-Verlag 出版社在过去几年中提供的良好合作表示诚挚的感谢。

约·斯托尔
罗·波里希

目　录

第一章 误 差 分 析

确定计算结果的精度是数值分析的首要目标。我们先来区分可能限制这种精度的几种误差类型：

(1) 输入数据误差，

(2) 舍入误差，

(3) 近似误差。

输入或数据误差是计算过程不能控制的，例如，它们可能是由于物理度量的固有不准确性而引起的。在通常情况，如果用限定有限位数字表示的数来计算就会出现舍入误差。

至于第三种误差，它会使许多方法得不到给定问题 P 的精确解，即使执行的计算没有舍入误差，仍然只能得到另一个较简单问题 $\tilde{P}$ 的解，而 $\tilde{P}$ 仅是 P 的一个近似。例如无穷级数求和问题 P，譬如

$$e=1+\frac{1}{1!}+\frac{1}{2!}+\frac{1}{3!}+\cdots,$$

可用仅求此级数的前有限项和的更简单的求和问题 $\tilde{P}$ 来代替。这样产生的近似误差通常称为**截断误差**(然而，这个术语也被用来表示将一个数的最后几位数字删去之后出现的舍入误差)。

P 的许多近似问题是通过“离散化”原问题 P 而获得的：用有限和来近似定积分，用差商来近似微商，等等。在这些情形，近似误差常常称为**离散化误差**。有些作者将术语“截断误差”推广到包含离散化误差。

在这一章，我们将研究输入误差和舍入误差对于计算结果的一般影响。近似误差将在后面诸章中涉及到各种方法时讨论。关于浮点计算中舍入误差的全面论述可以参看Sterbenz(1974)。

1.1 数的表示

根据表示数的基本不同方式，可将计算机分成两类：

(1) 模拟计算机，

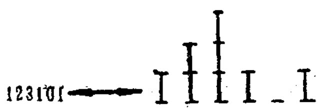
(2) 数字计算机。

模拟计算机的例子有计算尺和机械积分仪，以及电子模拟计算机。当使用这些装置时，我们可以用物理量，例如一根棒的长度或电压的强度，来表示数，并且可以用物理问题来模拟数学问题，前者用测量方法求解，从而产生原数学问题的解。例如计算尺的刻度表示数 x 是利用长为 $k\ln x$ 的线段，模拟乘法是通过邻接有关线段并测量综合长度而得到的。

显然，模拟装置的精度是直接受所用物理测量方法限制的。

数字计算机是用一列离散的物理量来表示一个数的每一位数字。典型的例子是台式计算器和电子数字计算机。

例



每一个数字可以用一个特殊的物理量表示。因为只有少数几个不同数字必须被代码化，例如在十进制数系中只有十个数字，所以在数字计算机中数字的表示并不需要像模拟计算机中表示数字那样精密。因此，当希望用 8 伏特来表示数字 8 时，我们允许电压在一定的范围内，譬如说在 7.8 到 8 伏特之间。

因而数字计算机的精确性并不直接受物理测量精密性的限制。

由于技术原因，大多数当代电子数字计算机在内部表示数是用**二进制**而非十进制。在数 x 的表示中，利用 2 的幂将它分解成

$$x=\pm(\alpha_n 2^n+\alpha_{n-1}2^{n-1}+\cdots+\alpha_0 2^0+\alpha_{-1}2^{-1}+\alpha_{-2}2^{-2}+\cdots)$$

$$\alpha_i = 0 \text{ 或 } 1,$$

其中的系数或二进位数α_i就起着数字的作用。为了不混淆数的十进制和二进制表示，我们将二进制数表示中的二进制位数分别用 **O** 和 **L** 表示。

例 数 $x=18.5$ 允许分解成

$$18.5 = 1\times 2^4 + 0\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 + 1\times 2^{-1},$$

因而其二进制表示是

LOOLO.L

我们将主要使用十进制，凡与手工检验有关时，就指出两种数制之间的差异。

正如 $3.999\cdots=4$ 指出的那样，一个十进制数的表示可能不是唯一的，二进制表示同样不是唯一的。为了排除这种歧义性，除非另作说明，我们始终采用有限位表示法。

一般来说，数字计算机内表示一个数必定用一固定的有限位数位，我们称其为**字长**。字长 n 是由机器决定的，虽然有些机器内部另有装置，在需要的时候可以扩充到 n 的整数倍 $2n$, $3n$, …(即双倍位字长、三倍位字长…)以提供更高的精度。一个 n 位的字长可以用几种不同方式来表示一个数。

定点表示是在十进制(或二进制)小数点前指定 n_1 位数，而在小数点后指定 n_2 位数，并使

$$n = n_1 + n_2 \quad (\text{通常 } n_1 = 0 \text{ 或 } n_1 = n)。$$

例 对于 $n=10$, $n_1=4$, $n_2=6$

$$\begin{array}{rl}
30.421 \longrightarrow & \boxed{0\;0\;3\;0 \mid 4\;2\;1\;0\;0\;0} \\
0.0437 \longrightarrow & \boxed{0\;0\;0\;0 \mid 0\;4\;3\;7\;0\;0} \\
 & \;\underbrace{}_{n_1}\;\;\underbrace{}_{n_2}
\end{array}$$

在此表示法中，十进制(二进制)小数点的位置是固定的。少数简单的数字装置，主要用于计数，仍然限用定点表示。更重要

的，特别对科学计算是以具有数的**浮点表示**为特色的数字计算机。在这种表示法中，十进制（二进制）小数点并不固定在开头，而其相对于第一位数字的位置是由每个数各自指定的，这可以通过规定一个被称为**阶**的数来实现。换句话说，每一个实数可以表示成如下形式

(1.1.1)　　$x=a\times 10^{b}(x=a\times 2^{b})$，$|a|<1$，$b$ 为整数。

（例如30.421可以表示为 0.30421×10^{2}），其中阶 b 指出了十进制小数点相对于尾数 a 的位置。Rutishauser 提出了以下的“半对数”表示法，它在下标位置上显示出数制的基，并将阶移动到尾数之后：

$$0.30421_{10}2$$

类似地，

$$\mathbf{0.LOOLOL_{2}LOL}$$

表示在二进制中的18.5。当然，在任何数字计算机上，只需固定的有限位数字 t 和 e，$n=t+e$，用以分别表示尾数和阶。

例　对于 $t=4$，$e=2$，十进制数 5420 有下面的浮点表示形式

$$0\boxed{5\ 4\ 2\ 0}_{10}\boxed{0\ 4}\quad \text{或更简洁地表示为}\quad \boxed{5\ 4\ 2\ 0\,|\,0\ 4}$$

一个数的浮点表示未必是唯一的，因为

$$5420=0.542_{10}4=0.0542_{10}5,$$

我们也可以用浮点表示

$$0\boxed{0\ 5\ 4\ 2}_{10}\boxed{0\ 5}\quad \text{或}\quad \boxed{0\ 5\ 4\ 2\,|\,0\ 5}$$

来代替上例所书浮点表示。

浮点表示是**规格化的**，如果尾数的第一位数字不是0（对二进制，则不是O）。那么在(1.1.1)中必有 $|a|\geqslant 10^{-1}$ $(|a|\geqslant 2^{-1})$。一个数的**有效数字**（包括二进制）是尾数中不计开头几位 0 的数字。

以后，我们将只考虑规格化浮点表示和相应的浮点运算。数 t 和 e——连同数表示法中的基 $B=10$ 或 $B=2$ 一起——决定了给定机器内能被精确表示出的实数集 $A\subseteq \mathbf{R}$。A 中的元素称为机