计算机数值方法

明日报出版社

中小学教师参考丛书

计算机数值方法

张石铭 编著
瞿连林 审校

光明日报出版社

(京)新登字101号

内容提要

本书系科普性读物，主要介绍电子计算机上常用的数值方法，如最小二乘法、插值法、数值微积分、方程求根、线性方程组数值解以及常微分方程数值解简介等。全书共分八章，每章开始均有概述，章末附有小结和习题，书末附有答案或提示，以便自学。

本书取材精炼，内容全面，重点突出，推导详细，深入浅出，通俗易懂，凡具有中等以上文化程度者均可阅读。本书可作为教师进修、成人自学以及各类成人高等教育的函授教材，也可作为师范专科学校、中等专业学校的数学教学参考书。

计算机数值方法
张石铭　编著
☆
光明日报出版社出版发行
(北京永安路106号)
邮政编码：100050
电话：3017733-225
新华书店北京发行所经销
[illegible]城新[illegible]印刷厂印刷
*
787×1092　1/32　　印张10.5字数 230 千字
1992 年 7 月第 1 版　　1992 年 7 月第 1 次印刷
印数：1—3000册
ISBN 7-80091-332-5/G·555

定　价：5.80 元

前　言

当今世界早已进入到电子计算机时代，为适应我国“四化建设”，在广大职工中推广和普及电子计算机上常用的算法知识已成为必不可少。正是基于这种考虑，近年来在成人教学的实践基础上，按照成人自学规律编写了这本书。

本书的主导思想是：从实用角度出发，强调计算机上行之有效的常用算法，重点介绍数值方法的基本概念，基本理论、数值计算的基本方法及其应用，因此，每种算法都附框图，典型算法还配有计算机上的通用程序，以供读者选用。

编者在内容安排上力求做到重点突出、条理清楚、知识完整；在文字叙述上力争做到由浅入深，通俗易懂，便于自学；在讲授顺序上尽量体现系统性原则。编写时是以数值计算采用的基本手段——“近似逼近”为主线，纵贯全书。全书共分八章，第一章除概述本门课程的主要内容、特点及误差理论外，突出介绍了计算机数系。第二章是介绍在计算机上处理数据的常用方法——最小二乘法，以曲线拟合，回归分析的两个具体内容为例，它们的共同特点是通过一组数据点寻找一个多项式来近似逼近所要研究的函数，使它们的偏差达到最小．若在所有的数据点上的偏差均为零，则此多项式就是插值多项式，这就是第三章的任务——插值法。第四章是直接利用插值理论来建立数值微积分的基本公式。第五章则是利用数值微积分来简介常微分方程的数值解。第六章是一元函数方程求根的迭代法。从形式上看它与前几章没有明

显的内在联系，但若从“逼近”的角度考虑，迭代法则是“逼近”的另一形式，只不过这里是找一数列逼近方程的根。第七章是线性方程组的迭代法，它也是通过找一向量序列来逼近其解向量。第八章则是线性方程组的直接解法，当然求出的结果仍然是近似解。

本书在编写过程中曾得到河北教育学院数学系领导以及计算机教研室同志们的大力支持；河北师院数学系主任王旭东副教授、计算数学教研室张国梁、董瑞山副教授在选材、取材以及编写等方面给予了很多帮助，特别董瑞山副教授对原稿进行了认真的阅读并提出了很多宝贵意见；保定师专荆霜雁同志为核对习题答案做了很多工作；特别是翟连林先生为本书出版提了很多建设性意见；卢承箴先生为出版本书付出了艰苦的劳动，在此一并向上述各位表示衷心地感谢。

由于编写时间仓促，再加水平有限，选材不当及错漏之处在所难免，恳切希望读者给予批评指正。

编者

1992年2月

目 录

第一章 预篇

为学好这门课程，本章将对什么是计算机数值方法，它的内容和方法，它所在的数系——计算机数系以及误差概念和其理论等作一概括介绍.

§1 计算方法的意义、内容和方法

(一) 什么叫计算机数值方法

研究如何利用手指、算尺、算盘、计算器、计算机等工具来求解数学问题数值解的学问叫计算数学或数值数学，又叫计算机数值方法或称计算方法、数值方法、数值分析等. 虽然名称不同但其含义一样.

计算方法这门科学，是在解决实际问题的长期过程中逐渐形成的. 它是数学中最古老的部分，是专门研究数值问题的一个学科. 所谓数值问题泛指是由一组原始数据，按确定的运算规律，最后计算出结果。由于数值计算繁难，长期来计算工具的笨拙，人们总是回避复杂计算，因而致使这门学科发展很缓慢.

1946 年电子数字计算机的问世，是计算数学史上一个里程碑. 由于有了先进的自动化高速度的计算工具，从而促进了计算方法的迅速发展和更新. 目前它的理论和应用仍在继续向深度和广度方面发展，一些边缘科学如计算物理、计算化学、计算生物学、计算经济学等相继出现. 为了更有效地利用计算机求解各类数学问题，学习计算机上常用的数值

方法已成为必不可少。

（二）计算方法的任务

计算方法的根本任务是研究算法。所谓算法是指：研究怎样通过计算机能够执行的运算（加、减、乘、除四则运算和逻辑运算）来求各类数学问题数值解时，所构成的完整的解题过程。当然算法的核心是计算公式。

例1 n 阶线性方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ \cdots\cdots\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n. \end{cases}$$

若用克莱姆(Cramer)规则来解，则需要计算 $n+1$ 个 n 阶行列式和做 n 次除法。而每一个 n 阶行列式有 $n!$ 个乘积项，每个这样的乘积项需做 $n-1$ 次乘法。因此，用克莱姆规则解 n 阶线性方程共需要做

$$A_n = (n+1)n!\,(n-1) + n$$

次乘除法。当 $n=20$ 时，大约需要做

$$A_{20} = 21 \times 20! \times 19 + 20 \approx 9.7 \times 10^{20}$$

次乘除，这么多次的乘除法，既使在每秒做一亿次乘除法的计算机上，也需 20 多万年才能完成。可见理论上很漂亮的克莱姆规则，在计算机上并不适用。由此可见研究算法的重要性。

那么一个算法的好坏，应该用什么样的标准来衡量呢？请看下面例子。

例2 求多项式

$$y = 0.0625x^4 + 0.425x^3 + 1.215x^2 + 1.912x + 2.1296$$

的值，可有下列三种作法：

（1） 直接逐项求和：需做10次乘法，4次加法。

（2） 秦九韶算法

$$y=(((0.0625x+0.425)x+1.215)x+1.912)x+2.1296$$

仅须做4次乘法，4次加法。这种算法具有一般性，它把计算高次多项式的复杂问题归结到了计算一次式简单问题的多次重复，此为计算机上常用的递推化手段，递推性在算法上表现为循环。

（3） 第三种方法是将多项式变形为

$$v=[(0.5\times+0.6)^2+0.5x+0.7]\times[(0.5x+0.6)^2+0.8]+0.9$$

只需做3次乘法，5次加法。表面上看，比秦九韶算法减少了一次乘法，但却增加了一次加法，似乎意义不大，但对经常要计算的多项式来说（比如取作计算机的标准函数），这样做还是值得的，因为计算机做加减法要比做乘除法快的多。

此例告诉人们，在选择计算机上算法时，应注意运算速度的快慢以及所占内存的大小。

近似代替是计算机数值计算的的基本方法，那么当进行近似计算时，应注意些什么呢？请看下例：

例3 按指数函数 e^x 的台劳展开式

$$e^x=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots$$

求值时，计算机不可能将右端无穷项相加，而只能截取有限项

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

作为 e^x 的近似值。在使用近似代替方法时，应该注意的是

收敛速度的快慢和误差大小的分析这两个问题。

例 4　线性方程组

$$\begin{cases} x_1 + \dfrac{1}{2}x_2 + \dfrac{1}{3}x_3 = \dfrac{1}{6}, \\ \dfrac{1}{2}x_1 + \dfrac{1}{3}x_2 + \dfrac{1}{4}x_3 = \dfrac{13}{12}, \\ \dfrac{1}{3}x_1 + \dfrac{1}{4}x_2 + \dfrac{1}{5}x_3 = \dfrac{47}{60}. \end{cases}$$

的真正解是 $x_1 = x_2 = x_3 = 1$. 而用计算机解此方程组时，须先将各系数取为近似数，比方取成两位数字

$$\begin{cases} x_1 + 0.5x_2 + 0.33x_3 = 1.8, \\ 0.5x_1 + 0.33x_2 + 0.25x_3 = 1.1, \\ 0.33x_1 + 0.25x_2 + 0.20x_3 = 0.78. \end{cases}$$

然后求解得 $x_1 = 6.222\cdots$, $x_2 = 38.25\cdots$, $x_3 = -33.65\cdots$ 此与真正解面目皆非了。数据的微小变化却引起了解的剧烈变化，这样的问题称病态问题。用计算机近似求解这类问题，十分冒险，所得结果很可能根本不可靠，处理这类问题应十分小心，一般应采用高精度计算，比如取各系数为3位数字，则解为 $x_1 = 1.09$，$x_2 = 0.488$，$x_3 = 1.49$ 此与真正解的差别和 2 位数字相比小多了。

既使对非病态的问题，算法选择不当也会出现大的误差。比如两个相近的近似数相减就属于这种情况，其原因请参看本章 §3 定理 4。

总之，由于近似计算不可避免地会出现舍入误差，这就影响了精度，有的影响大，有的影响小，人们称舍入误差影响小的算法为数值稳定的算法。显然计算机上应采用这种算法。

综合上述各例，可以看出研究算法应从构造算法和分析算法这两个方面着手．在构造算法时，不仅要注意算法的可计算性，还应注意选择计算过程简单，计算量小，所占内存小且具有递推性的算法（如上面例2中的第二种解法秦九韶算法）；在分析算法时，应着重分析算法的稳定性，收敛性以及算法的精度——误差的大小等．

（三）计算方法的内容和方法

分析数学与计算数学的主要区别在于：前者是研究运算规则及其表现形式，而后者着重于运算结果，而数值计算又常采用近似代替．由递推公式求近似解的算法是逐次逼近又称迭代法，它可用来求线性及非线性方程的数值解；由函数逼近求近似解的算法是插值法、数值积分、微分方程数值解等．计算方法涉及的面很广，而且内容很杂，本书所讨论的范围仅限于下述三个方面：

（1）数值逼近法．就其具体内容有：最小二乘法、插值法、数值积分等．

（2）数值代数．就其具体内容有：一元函数方程的数值解，多元线性函数方程的数值解等等．

（3）微分方程数值解——仅限于常微分方程的数值解简介．

§2 计算机数系

由于我们所讨论的数值方法是在数字电子计算机上实现的，因此对计算机中存有哪些数，怎样运算以及如何表示等，在学数值方法时应有所了解．

（一）定点数

数字电子计算机进行运算时，必须按一定方法确定小数

点的位置，有一种方法是将小数点永远固定在指定的位置上，称它为定点表示法。

例如：一个n位的r进制数

$$x = a_{l-1}a_{l-2}\cdots a_1 a_0 . a_{-1}a_{-2}\cdots a_{-m},$$

其中a_i为0，1，2，…，$r-1$中的某一个。数x还可表示成：

$$x = a_{l-1}r^{l-1} + a_{l-2}r^{l-2} + \cdots + a_1 r^1 + a_0 r^0 + a_{-1}r^{-1} + \cdots + a_{-m}r^{-m}.$$

在机器内数x是用依次排列的n个单元来表示的，前l个单元是整数部分，后m个单元是小数部分。每个单元可产生r种不同的物理信息。这里的n称字长。r为基数，它可取作10，2，8，16等。当$r=10$，$l=4$，$m=4$时，下列数

109.312，0.4375，4236

分别表示成

0109.3120，0000.4375，4236.0000

这种把小数点永远固定在指定位置上且位数有限的数，称定点数。采用定点数的计算机称定点机。当$l=m=4$，而$r=10$时，在8位定点非零数中，绝对值最小和绝对值最大的数分别为

±0000.0001，±9999.9999

由此可见，定点数所能表示的范围是很小的。

（二）浮点数

在数的表示中，还有一种方法，那就是允许小数点的位置可以变动，称它为浮点表示法

1. 十进制浮点数

一个很大的数或很小的数用浮点表示是极为方便的。

如我国十亿零七千万人口可表示成

$$1070000000=107\times10^{7}=1.07\times10^{9},$$

再如一个电子的质量可表示成 9.11×10^{-31}。

一般，设 p 为十进整数（正、负或零），s 为十进小数，即

$$-1<s<1 \quad ①$$

若数 x 可表示成形如

$$x=s\times10^{p} \quad ②$$

则称数 x 为满足条件①的十进制浮点数。s 和 p 分别称浮点数 x 的尾数和阶码。若尾数 s 的小数位数是 t 位，则 x 称 t 位浮点数。采用浮点数的数字电子计算机称浮点机。浮点机比定点机表示数的范围要大，使用也方便，但结构复杂，所用器材较多，运算速度也慢。浮点机中的浮点数的位数是有限的，若为 t 位则 t 称作计算机的字长。

如果要让尾数 s 小数点后第一位数字不等于零，也就是要求尾数 s 满足

$$10^{-1}\leqslant|s|<1 \quad ③$$

那么，形如②且满足条件③的浮点数称作十进制规格化浮点数。如下列数

$$0.0004012,\ 31.27,\ 3450$$

四位规格化的浮点形式为

$$0.4012\times10^{-3},\ 0.3127\times10^{2},\ 0.3450\times10^{4}.$$

由此可见，只要数 $x\neq0$，它就一定能表示成规格化的浮点数。

除十进制浮点外，还有二进制、八进制、十六进制等浮点数，为避免重复，仅对一般数制加以讨论。

2. r 进制浮点数

设基数 r 为大于 1 的整数，s 是 r 进制小数，p 为 r 进制

整数（正、负或零），则称形如

$$x=s\times r^{p} \qquad ④$$

且满足条件①的数 x 为 r 进制浮点数。其中 s 和 p 分别为 r 进制浮点数 x 的尾数和阶码。

如果要让尾数 s 的小数点后第一位数字不为零，即满足

$$r^{-1}\leqslant|s|<1 \qquad ⑤$$

则称形如④且满足条件⑤的 r 进制数 x 为 r 进制规格化的浮点数。如二进数

$$111.1000,\ 0.0111$$

4 位规格化的浮点形式分别为

$$0.1111\times 2^{11},\ 0.1110\times 2^{-1}.$$

由此不难发现，只要 $x\neq 0$ 就一定能化为规格化的浮点数。

综合上述，不论何种数制的浮点数，均系由尾数和阶码两部分构成，阶码的大小决定了小数点的位置，每一部分各自有自已的符号位，不同的计算机，尾数和阶码是不同的。

（三）计算机数系

在数学分析中，已经知道数系是一个无穷稠密的连续集合，在这个数系上建立了严格的极限理论，从而使微积分学才有了可靠的逻辑基础。

那么计算机数系又是什么样的呢？上面介绍的数的浮点表示法为现代数字电子计算机所通用，是我们讨论数值方法的基础。任一台计算机只能用有限的位数来表示浮点数的尾数和阶码，设进位制为 r，阶码 p 满足条件

$$-m\leqslant p\leqslant M \qquad ⑥$$

其中 m，M 为正整数。它们主要由计算机用多少位数来表示阶码而决定。如果尾数 s 的小数位数为 t（t 一般比 m，M

的位数大若干倍)，则计算机中的数系是：由一切阶码满足条件⑥的 t 位 r 进制浮点数所组成的集合 F。

为了说明计算机数系的特点，分析几个例子。

例 1　设 $r=10$，$t=4$，$m=M=99$ 在此条件下的计算机数系为集合

$$F=\{s\times 10^{p}\mid \text{一切满足} -99\leqslant p\leqslant 99 \text{ 的 4 位浮点数}\}$$

显然，F 中绝对值最小的非零数是 $\pm 0.0001\times 10^{-99}$，而 F 中的最小数与最大数分别为

$$-0.9999\times 10^{99} \text{ 和 } 0.9999\times 10^{99}.$$

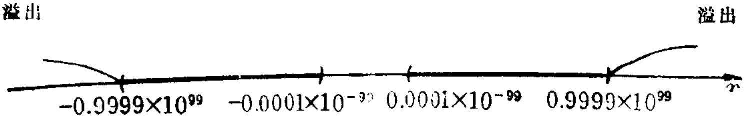

图 1-1

若计算的中间结果超出上述范围(如图1-1)，则称溢出。

下面我们将指出这个数系中的数是离散的，比方说二数

$$0.2345\times 10^{15} \text{ 与 } 0.2346\times 10^{15}$$

在 F 中是紧挨着的，因为它们中间的任何有理数都不是 4 位浮点数，因而都不是这个数系中的数。如数 0.23454×10^{15} 虽然满足

$$0.2345\times 10^{15}<0.23454\times 10^{15}<0.2346\times 10^{15}$$

但它却是 5 ($t=5$) 位浮点数。由此不难理解二数

$$s\times 10^{p} \text{ 与 } (s+10^{-4})\times 10^{p}$$

在 F 中是紧挨着的，也就是说它们中间的任何有理数都不是 4 位浮点数，从而都不是这个数系中的数。

此例说明了计算机数系是一个有限的，离散的有理数集合并且数的分布也极不均匀，这可从下例看出。