计算方法

黄友谦 主编

高等教育出版社

计 算 方 法

黄友谦　主编

高等教育出版社

(京)112号

内容提要

本书介绍的计算方法，主要是叙述常用数值算法的基础理论，并将算法的建立与算法的机器实现紧密联系在一起．本书强调用数值例子说明算法的可靠性，同时对某些算法的复杂性也做了较详尽的分析．

本书在传统的计算方法基础上，引进了一些较新的算法，全书内容较为完备，可以适应不同专业学习计算方法课程的需要。每章配有适量的习题。

计算方法
黄友谦 主编

高等教育出版社 出版
新华书店总店科技发行所发行
北京市通县觅子店印刷厂印装

开本 850×1168 1/32 印张 8.875 字数 210 000
1994年11月第1版 1995年10月第2次印刷
印数 4 076—6 783
ISBN 7-04-004878-7/TP·131
定价 7.20 元

序 言

本书的大部份是编者在中山大学计算机科学系讲授"计算方法"的讲课笔记．它也反映了我们多年来在中山大学数学、力学等专业讲这门课的教学体会和经验，借鉴了国内外有关的教材，仔细地进行了取舍，在实践的基础上经反复修改讨论写成的．书中对传统的计算方法作了锤炼并适当注入一些新的算法以适应不同专业的需要．书稿写成后编者又在中山大学计算机科学系讲授了一遍，作了若干修改和补充．

解决一个复杂的数值计算问题的主要途径是用有穷和代替无穷和或将非线性问题逐步线性化．换句话说，我们常用数列、函数序列、向量序列去逼近真解．具体的实施方案则是将问题化为有穷步，由计算机去计算．在编写这本书时，重点突出：算法是如何构思的；从非线性到线性是如何逐步转化的；算法的效率及机器实现，并用典型的数值例子去说明算法的正确性．为使学生能运用计算机的高级语言编写计算方法的程序，书中对算法的叙述尽量做到条理分明、格式规范、下标变量的功能明确，总之一目了然．使学生根据算法编写的程序容易阅读、容易使用、容易包装、容易维护．并让学生自己体会到，通过剖析典型的数值例子和在计算机上做数值试验，能更牢固地掌握计算方法．为此，对某些有代表性的算法，书中列出了容易阅读的计算流程．

为适应不同专业的需要并使学生在约68学时内学完本书的主要内容，我们建议，对数学、应用数学、力学、物理、气象等专业，可选择第一至第八章、第十章中的部份内容进行讲授，而对工科类如计

算机软件、计算机应用、成人教育的有关专业，则建议删去其中的常微分方程初值问题数值解和第四章矩阵特征值的计算的部份或全部内容，增加第九章线性规划或第十一章贝齐尔和B样条曲线的部份内容.我们还将一些较难或适应面较窄的章节，或某些章节中的定理、例题和习题用*号示明.特别是，正文中略去了矩阵的范数，将它列入本书末尾部分的附录1.这样做并不影响讲授的系统性和完整性.因为，对于那些已掌握微积分和线性代数的学生来讲，能自学好书中标上*号的内容.我们的观点是，对某些专业或专业方向，教师可选择部份标有*号的内容作为专题讲座或毕业论文的参考阅读资料.例如，快速付里叶变换FFT、贝齐尔和B样条曲线对计算机专业是有用的.软件专业的学生学线性规划甚至学一点非线性规划是有好处的.自由曲线设计一节，讲授时要突出保直、保凸、保尖概念，并用大量实例启发学生在计算机屏幕上画出五彩缤纷的艺术性图案，如再向学生介绍一点分形几何知识，学生设计出的图案将更丰富多彩.

必须明确指出，为使学生掌握这门课的基本知识，教师应选择每章后面三至五道习题作为学生的课外作业，同时本课程还应安排学生完成二至三次计算机实习.例如，解一元非线性代数方程；解线性代数方程组的高斯消元法，LU分解、Cholesky分解或迭代法，共轭梯度法；矩阵求逆；求特征值的幂法或QR方法；曲线拟合的最小二乘法；样条插值法；龙贝格求积分方法；解常微分方程的预报校正法；在计算机屏幕上作自由曲线设计等都是实习的好题目，教师可选择其中二至三个题目让学生将算法编成程序，通过上机，写出实习报告.例如，当线性代数方程组的系数矩阵是Hilbert矩阵时，学生通过实习会发现计算是严重病态的.又例如，对被积函数的不同选择(参见第七章习题4)在龙贝格求积分方法的上机实践中将体会到软件包装和维护的重要性.对于一个算法来讲，如

何选定停机的条件呢？只能通过上机做数值试验去摸索，在实践中完善软件的包装.

潘震泰副教授参加了线性和非线性规划、自由曲线设计等内容的讨论和这些内容的部份编写工作，他还仔细地阅读了本书的初稿，提出了好的建议. 书中所有算法的流程设计和算法的复杂性分析是由黄东斌高级程序员编写的. 常微分歧解一节是吴湘辉副教授编写的. 我的几位研究生曾参加本教材初期的建设工作，提供个别内容的素材或算例. 中山大学计算机科学系90级和92级本科生提供了许多生动的上机实习报告，我十分珍惜这些充满活力的实例并将它写入书中. 作者感谢湘潭大学傅凯新教授对本书提出的十分具体而又中肯的宝贵意见.

作者从中山大学李岳生教授的教学和学术讲演中得到许多教益. 中山大学岭南(大学)学院王屏山院长十分关心并支持本书的编写工作. 本书付梓前，北京大学数学系应隆安教授审阅了全部书稿. 谨此表示深深的感谢.

受水平的限制，书中的错漏和不妥之处敬请兄弟院校同行和读者指正.

黄友谦

于中山大学岭南(大学)

学院计算机科学系

1993.5.22

目　　录

引论……………………………………………………………1
习题……………………………………………………………6
第一章　一元非线性代数方程的求解……………………………8
§1　对半分法……………………………………………………8
§2　一般迭代法…………………………………………………10
§3　牛顿法………………………………………………………15
§4　正割法………………………………………………………19
习题……………………………………………………………22
第二章　解线性代数方程组的直接法……………………………24
§1　高斯(Gauss)消元法…………………………………………24
§2　矩阵的 LU 分解……………………………………………35
§3　乔列夫斯基(Cholesky)方法…………………………………40
§4　向量的范数…………………………………………………43
习题……………………………………………………………45
第三章　解线性代数方程组的迭代法……………………………48
§1　迭代法的基本概念…………………………………………48
§2　雅可比(Jacobi)和高斯-塞德尔(Gauss-Seidel)迭代………52
§3　松弛(SOR)迭代……………………………………………56
§4　共轭梯度法(Conjugate Gradient Methods)………………59
习题……………………………………………………………65
第四章　矩阵特征值的计算………………………………………67
§1　幂法和反幂法………………………………………………67
§2　矩阵的 QR 分解及其应用…………………………………71
*§3　Hessenberg 矩阵及其应用…………………………………81
*§4　求对称矩阵特征值的雅可比(Jacobi)方法…………………88
习题……………………………………………………………92

第五章 最小二乘法……95
§1 线性最小二乘法问题……95
*§2 矩阵的奇异值分解……102
§3 积分意义下的最小二乘法……105
§4 函数按正交函数系展开……107
习题……115
第六章 插值方法……117
§1 代数插值的拉格朗日(Lagrange)公式……117
§2 代数插值的牛顿(Newton)公式……120
§3 埃尔米特(Hermite) 插值……127
§4 三次样条插值……129
*§5 张力样条……135
*§6 复指数插值与快速付里叶变换(FFT)……137
习题……144
第七章 数值积分和数值微分……149
§1 中矩形、梯形和辛浦生(Simpson) 求积公式……149
§2 复化求积公式……153
§3 龙贝格(Romberg) 求积方法……156
§4 高斯型求积公式……160
§5 数值微分……165
习题……169
第八章 常微分方程初值问题数值解……171
§1 解常微初值问题的单步法……171
§2 单步法的理论分析……178
§3 预报-校正方法……185
*§4 分歧解……192
习题……196
第九章 线性规划……198
§1 问题的提出……198
§2 解的代数性质……201
§3 单纯形方法……206

*§4　对偶问题……215
习题……219
第十章　非线性代数方程组求解……222
§1　求解非线性代数方程组的迭代法……222
§2　牛顿法……226
*§3　非线性最小二乘法……228
*§4　非线性规划初步……233
习题……241
第十一章　贝齐尔(Bézier)和B样条曲线……243
*§1　伯恩斯坦(Bernstein)多项式……243
*§2　贝齐尔曲线……246
§3　B样条函数……248
§4　B样条曲线……252
§5　自由曲线设计……255
习题……261
***附录1　矩阵范数及其应用**……264
***附录2　贝努里(Bernoulli)多项式及其应用**……269

引　论

计算机对科学技术的冲击是十分深远的，对于科学研究和工程设计来说，计算的手段已成为一种重要的因素，甚至成为一种基本方式．计算机能够解决那些至今人们还不知道如何解决的许多问题．随着计算技术和系统软件的不断更新，计算机解题的范围越来越广，既有数值的，也有非数值的．人工智能理论的提出和逐趋完善使数值和非数值的算法紧密结合，相互渗透．从应用角度看人们常将计算机算法分成两大类．一类称为"计算方法"，主要解决数值计算问题，如求解方程组、曲线拟合、积分计算等，主要算法有迭代法、分解法、投影法、降维法、插值法、蒙特卡罗方法、有限付里叶分析法等．另一类称为"组合算法"，解决搜索、排序、离散变量（一个排列或一个图）的最优化问题，常用算法有动态规划法、回溯法和广泛应用于智能领域的启发式搜索算法．本书要介绍的是"计算方法"，叙述常用数值算法的基本理论，将算法的建立与算法的机器实现紧密联系在一起，强调用数值例子去说明算法的可靠性，对某些算法的复杂性也作了较详尽的分析，举例说明如下．

例 1　设计一个算法求代数多项式 $P_n(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$ 的值和一阶，二阶导数值．

解　记 $P_0(x)=a_0$，对 $k=1,2,\cdots,n$ 引进记号

$$P_k(x)=a_0x^k+a_1x^{k-1}+\cdots+a_{k-1}x+a_k,$$

有

$$P_k(x)=xP_{k-1}(x)+a_k,\ P_0(x)=a_0.$$

求

$$P'_k(x)=xP'_{k-1}(x)+P_{k-1}(x),\ P'_0(x)=0,$$

$$P''_k(x)=xP''_{k-1}(x)+2P'_{k-1}(x),\ P''_0(x)=0.$$

分别用三个单元 b_0, b_1, b_2 来存储 $P_n(x), P'_n(x), P''_n(x)$的值，上述算法(又称秦九韶算法)过程写成

$$
\begin{aligned}
&\textbf{\textit{iuput}}\ a_0, a_1, \cdots, a_n, x\\
&b_0 \leftarrow a_0, b_1 \leftarrow 0, b_2 \leftarrow 0\\
&\textbf{\textit{for}}\ k=1,2,\cdots,n\ \textbf{do}\\
&\quad b_2 \leftarrow x b_2 + 2b_1\\
&\quad b_1 \leftarrow x b_1 + b_0\\
&\quad b_0 \leftarrow x b_0 + a_k\\
&\quad \textbf{end}\\
&\quad \textbf{output}\ b_0, b_1, b_2
\end{aligned}
$$

对一个复杂的问题，只是给出一个算法是不够的，我们必须研究算法在计算机上实现的可能性，也即计算机能否用可能利用的存储空间和适度的时间来完成解题的全部运算.为此，需要引入算法复杂性度量，它包含**时间复杂度**和**空间复杂度**两个概念.

时间复杂度是算法需耗费时间的度量，即分析计算量，也称计算的复杂度.它依赖于问题的大小，例如矩阵计算的复杂度是用矩阵的阶 n 来度量的. 一个问题计算量的大小往往可以用一个整数或多个整数来描述，计算的复杂度则可以用这些整数的数量级来估计.由于乘除法与加减法相比花费的机时多，通常将算法所需要的乘除法的总次数作为计算量大小的尺度. 例如两个 n 阶矩阵相乘的乘法次数是 n^3，称两个 n 阶矩阵相乘的计算复杂度为 $O(n^3)$. 用克莱姆法求 n 阶行列式的值的计算复杂度是 $(n-1)n!$，而用消元法计算的复杂度为 $n^3/3$，当 n很大时，前值远远超过后值。例如当 $n=100$ 时，

$$99\cdot 100! \approx 10^{160},\ (100)^3/3 = 10^6/3,$$

即使是每秒 1 亿次的计算机也无法完成 10^{160} 量级的计算量.

如果计算复杂度对于问题的大小来说呈多项式级(例如 n^k,

$n\log n$ 等)，则称问题是可计算的，否则称为不可计算的，即 NP 问题. 一个问题可能有多种算法，达到算法复杂性下界的算法称为最佳算法.

算法的**空间复杂度**是指算法需占用存储空间的量度，它也依赖问题的大小.

无疑，算法的建立和算法复杂性分析是十分重要的. 还有，由于参加运算的是数，每一步计算都会有舍入误差，误差的积累是否对计算结果的正确性有影响呢?

例 2 试建立一个算法来计算积分

$$I_n=\int_0^1\frac{x^n}{x+999}\mathrm{d}x,\ n=0,1,\cdots.$$

解 注意到

$$(x^n+999x^{n-1})/(x+999)=x^{n-1}$$

两边在区间[0,1]上积分有

$$I_n+999I_{n-1}=\int_0^1 x^{n-1}\mathrm{d}x=1/n,$$

即

$$I_n=-999I_{n-1}+1/n,\ n=1,2,\cdots,$$

$$I_0=\ln(1000/999)=1.000500333\times10^{-3}.$$

按照所建立的递推关系作计算

n	I_n
0	1.000500333×10^{-3}
1	5.0016734×10^{-4}
2	3.3282734×10^{-4}
3	8.3882067×10^{-4}
4	-0.58798184
5	587.5938675
6	-587006.106

计算结果可靠吗?我们知道当 $0\leqslant x\leqslant 1$ 时 $x^n\leqslant x^{n-1}$,由此推知

$$I_n<I_{n-1},$$

又

$$0<\int_0^1\frac{x^n}{x+999}\mathrm{d}x<\int_0^1 x^{n-1}\mathrm{d}x=\frac{1}{n}.$$

故 $\{I_n\}$ 是严格单调递减收敛于 0 的数列. 但从计算结果看 $I_3>I_2$, $I_4<0$. 即递推式从 $n=3$ 开始便产生不正确的数值结果. 从理论上讲 $0<I_6<1/6$, 但计算结果 I_6 等于 -587006.106, 计算过程中,由于舍入误差的影响,将真解完全掩盖了. 因而,我们所建立的算法是坏的. 产生计算谬误的原因是,若 I_0 有 ε 误差,则计算 I_1 时有 999 倍 ε 的误差, 计算 I_2 有 $(999)^2$ 倍 ε 的误差, 误差的传播象洪水冲破闸门一样泛滥成灾了.

如果某算法在运算的过程中舍入误差的积累对最后计算结果的影响不大,则这种算法是稳定算法,否则就是不稳定算法. 如果算法对输入数值的微小误差反应敏感, 即输入数值的微小变化引起了解的很大变化,则称算法是**病态**的,否则称算法是**良性**的.

我们来叙述近似数的绝对误差,相对误差和有效数字的概念.

定义 1 记 x^* 为 x 的近似数,称 $E(x)=x-x^*$ 为近似数 x^* 的绝对误差. 如果 $|x-x^*|\leqslant\eta$,称 η 为近似数 x^* 的**绝对误差限**.

只用绝对误差还不能刻画数的近似程度, 例如测量 100 米距离和 10 米距离的最大误差都为 1 厘米,那么它们的绝对误差限都是 0.01 米,但前者比后者测量得更准确,因为前者平均 10 米才有 0.1 厘米的测量误差. 这启发我们, 在考察近似数的误差时,除了要看绝对误差的大小外,还必须顾及量本身的大小.

定义 2 称 $E_r(x)=(x-x^*)/x$ 为近似数 x^* 的**相对误差**. 实际运算时,也将 $E_r^*(x)=(x-x^*)/x^*$ 称为 x^* 的相对误差. 如果 $|E_r(x)\leqslant\delta$ 或 $|E_r^*(x)|\leqslant\delta$, 则称 δ 为近似数 x^* 的相对误差限.

大家知道,十进制数在运算时常常要进行“四舍五入”,即尾数

四以下则舍，尾数五以上则进 1，如果尾数恰是五，则作如下规定：若其前一位数字是偶数则舍去，是奇数则进一．我们将近似数 x^* 写成

$$x^* = \pm 10^{m+1} \sum_{j=1}^{n} a_j 10^{-j}, m \text{ 为整数}, a_1 \neq 0, \tag{1.1}$$

若 x^* 的最末一位数 a_n 是经过"四舍五入"而得到的，则

$$|x - x^*| \leqslant \frac{1}{2} 10^{m-n+1} \tag{1.2}$$

定义 3 若 x 的近似数 x^* 写成(1.1)的形式，且其绝对误差限满足

$$|x - x^*| \leqslant \frac{1}{2} 10^{m-n+1} \quad ,$$

则称 x^* 有 n 位**有效数字**.

π 的近似值 3.1416 具有五位有效数字．将 $x = 0.00934$ 舍入为 $x^* = 0.0093$ 具有二位有效数字，不能说它具有四位有效数字．

绝对误差、相对误差、有效数字，它们之间有什么关系呢？由定义知，有效数字位越多，绝对误差限便越小；反而若近似数 x^* 有 n 位有效数字，则其相对误差限

$$|E_r^*(x)| \leqslant 10^{-(n-1)}/2a_1. \tag{1.3}$$

一般讲，在计算机上进行算术运算时

$$(a+b)+c \neq a+(b+c), (a+b)\cdot c \neq a\cdot c + b\cdot c,$$

发生误差的根本原因，是计算机只能对有限位数进行运算．一个工程或科学问题，其数值运算的次数以亿万次计，如果我们对每一步的计算都去分析计算的误差，那是办不到的．我们在这里提出近似数的算术运算中应注意的若干事项．

(1) 两个相近的数进行减法运算时有效数字会严重损失．例如 $\cos 2° \approx 0.9994$ 有四位有效数字，而 $1-\cos 2° \approx 1-0.9994=$

0.0006 却可能只具有一位有效数字.

(2) 要避免用绝对值很小的数作除数.

(3) 要防止大数"吃掉"小数，阶码相差很大的两个浮点数相加减时，由于对阶大数会"吃掉"小数，调整运算的次序往往可以避免这种情况.

(4) 要尽量简化计算步骤，这不但可以节省解题的计算时间，还能减少舍入误差.

(5) 参与运算的各数的精度尽量保持一致，如 π 在参加运算时取到小数后第五位，$\sqrt{2}$ 也取到小数后第五位.

例 3 设 $b\approx 10$, $a\approx c=10^{12}$ 有 $(a-c)+b\approx 10$，而 $(a+b)-c\approx 0$. 在计算 $a+b$ 时由于对阶 a 将 b"吃掉"了，$a+b\approx 10^{12}$.

例 4 在 8位十进制的计算机上解方程

$$x^2+(a+b)x+10^9=0, a=-10^9, b=-1$$

利用求根公式

$$x_1=(-(a+b)+\sqrt{(a+b)^2-4\cdot 10^9})/2\approx 10^9,$$

$$x_2=(-(a+b)+\sqrt{(a+b)^2-4\cdot 10^9})/2\approx 0,$$

而方程的一个根是 10^9，另一个根是1. 故 x_2 的计算结果是谬误的，这因为 $\sqrt{(a+b)^2-4\cdot 10^9}\approx|a+b|$，故在 x_2 的计算式子中，分子的计算碰到两个相近的数相减，结果精度降低. 为了避免两个相近的数相减，利用求根的韦达公式便有 $x_2=10^9/x_1=1$ 与真解相同.

习　题

1. 设计一个算法求方程 $x^2+c_1x+c_2=0$ 的解，算法中要避免出现两个接近的数相减.

2. 改变下列表达式使计算结果较可靠

(a) $f(x)=(x-\sin x)/\mathrm{tg}\,x,\ x=0.01$;

(b) $f(x)=x-\sqrt{x^2-100}$, $x=100000$;

(c) $f(x)=(\sqrt{x}-1)^{10}$, $x=2, \sqrt{2}\approx1.414$.

3. 说明用递推关系 $I_n=1-nI_{n-1}$ 来计算积分 $I_n=\int_0^1 x^n e^{x-1}\mathrm{d}x$, $n=0, 1, 2, \cdots$ 是病态的.

4. 给定

$$f(t)=(\sqrt{1+t}-1)/t, \quad g(t)=1/(1+\sqrt{1+t}), t\neq0,$$

易知 $f(t)\equiv g(t)$, 且当 $t\to0$ 时, $f(t), g(t)\to1/2$, 记 $t_i=10^{-8}(1-i/10)$, $i=1, 2\cdots, 9$, 在计算机上求 $f(t_i), g(t_i)$, $i=1, 2, \cdots, 9$, 将结果与极限值1/2作比较.

5. 记 $x_n=\sum_{k=1}^{n}(-1)^{k-1}/k$, 有 $|\ln2-x_n|<1/(n+1)$, 取 $n=500000$, 有 $|x_{500000}-\ln2|<0.5\times10^{-5}$, 即 x_{500000} 具有 5 位有效数字. 在计算机上计算 x_n, 由于舍入误差影响, 设计算结果为 $\bar{x}_{500000}$, 问 $\bar{x}_{500000}$ 有几位有效数字.

第一章　一元非线性代数方程的求解

在自然科学的许多领域中常会遇到这样的问题，寻找某一函数 $f(x)$ 的零点，即确定方程 $f(x)=0$ 的根. 例如在光的绕射中要求解方程 $x-\mathrm{tg}\,x=0$；为了确定行星的轨道，必须求解开普勒(Kepler) 方程 $x-a\sin x=b$，这里 a、b 是已知常数. 我们仅限于讨论实函数的零点，即如何确定方程 $f(x)=0$ 的实根.

求方程 $f(x)=0$ 实根的几何意义是寻找曲线 $y=f(x)$ 与 x 轴的交点. 通过 $y=f(x)$ 的图象可粗略地确定方程 $f(x)=0$ 的根的初始近似，有时通过 $f(x)$ 的物理意义也能找到方程 $f(x)=0$ 的根所落在的区间.

§1　对 半 分 法

假定函数 f 在区间 $[a, b]$ 上连续并在区间的端点异号即 $f(a)f(b)<0$. 由连续函数的性质知道，在 $[a, b]$ 上至少有一个 f 的零点，即在 $[a,b]$ 上至少有方程 $f(x)=0$ 的一个根. 用对半分法寻找 $f(x)=0$ 在 $[a,b]$ 上一个根的方法如下.

记区间 $[a,b]$ 的中点为 c，即 $c=(a+b)/2$，对于 $f(x)$ 会出现下面三种情况：若 $f(a)f(c)<0$，则 f 在区间 $[a,c]$ 有零点，将 c 看成新的 b 仍记新区间为 $[a,b]$，这个新区间的长度是原区间长度的一半；若 $f(a)f(c)>0$，则 $f(c)f(b)<0$，将 c 看成新的 a 仍记新区间为 $[a,b]$，f 在这个新区间有零点，这个新区间的长度也是原区间长度的一半；若 $f(a)f(c)=0$，则由于 $f(a)\neq 0$，有 $f(c)=0$，即 c 是