在计算机上解题的计算方法

黄友谦　吴炯平　编著

在计算机上解题的

计 算 方 法

黄友谦　吴炯平编著

广东高等教育出版社
·广　州·

图书在版编目（CIP）数据

在计算机上解题的计算方法/黄友谦，吴炯平编著．—广州：广东高等教育出版社，1998.4
ISBN 7-5361-2095-8

Ⅰ．在…　Ⅱ．①黄…　②吴…　Ⅲ．电子计算机-计算方法
Ⅳ．TP30

中国版本图书馆 CIP 数据核字（98）第 00249 号

广东高等教育出版社出版发行
广东省茂名广发印刷有限公司印刷
850 毫米×1168 毫米　1/32　13.625 印张　342 千字
1998 年 4 月第 1 版　2001 年 1 月第 3 次印刷
印数：4001～7000 册
定价：20.00 元

前　言

电子计算机的诞生是本世纪重大科学成就之一．半个世纪来，几乎所有的重要学科都在利用计算机的成就来加速本学科的发展．从宇宙飞行到核电站的设计，从家电设计到石油勘探，从人体内脏的计算机仿真到生物工程，从计算机辅助设计到三维动画，从自然科学到社会科学，科学家和工程师们越来越重视通过数学建模和寻求模型的可编程算法（数值或非数值），借助计算机求解，实现高技术的高精度、高效益．计算机促进了许多新的、特别是交叉学科的发展，全面推动了社会的进步．今天，从科学方法论看，物理实验、理论研究和计算机的“计算”手段共同组成了人类认识自然的基本途径．

本书从一个侧面阐述计算机解题的一类方法，即计算方法或称数值方法．牛顿(Newton)、欧拉（Euler）和高斯（Gauss）是计算方法这一学科的奠基人．他们十分重视通过近似计算来解决物理和数学中的许多问题．计算机出现后，他们提出的算法得到了实现．回顾半个世纪以来科学和技术迅猛的发展，人们十分深切地感到，科学巨匠们的辛勤劳动得到

了巨大的回报. 当计算机经过几亿次甚至几千亿次的算术运算, 最后获得一个科学或工程设计问题的答案时, 人们的心情是何等的兴奋!

人类总是不断进步的. 科学家和工程师们在计算机解题的实践过程中, 逐步明确并形成了 SOR, CGM, FFT, QR, Romberg 等新的算法, 出现了样条函数 (Spline Function) 的新概念, 稀疏矩阵的存储技巧, 加速收敛的外推手段, 三维动画的造型方法, 分形、分叉、混沌、小波的计算方法等, 本书将选择其中一些作介绍.

本书主要介绍: 在计算机上解线性和非线性代数方程的数值方法; 求矩阵特征值的计算机方法; 代数插值方法和样条插值方法; 在计算机上求积分的 Romberg 方法和 Gauss 方法; 实验数据处理的 Gauss 最小二乘法; 自由曲线; 解常微分方程初值问题的数值方法等. 阐述这些常用的数值方法的基本理论, 将算法的建立与算法的机器实现紧密联系在一起, 对某些算法的计算复杂性也作了较详尽的分析. 这里必须强调, 只有通过上机实验, 才能体会算法的可靠性与有效性, 书中列举的 Hilbert 矩阵便是一个例子.

在编写本书时, 重点突出: 如何根据计算机高级语言的特点去构思解题的计算方法; 对计算机舍入误差的积累引起计算严重失真的某些现象给予充分的重视和揭露; 强调方法与流程设计的一致性与可读性; 某些非线性问题如何逐步线性化, 以至用计算机求出它的答案的全过程. 这是本书的一个难点, 也是近代计算机方法的精髓.

书中对某些适应面较窄或较难的章节或定理用"*"号示明. 如果选作教材, 略去"*"号部分, 可在 68 学时内讲授完毕. 建议教师指导学生完成解非线性代数方程、Gauss 消元法解线性代数方程组、Jacobi 和 Gauss-Seidel 迭代法解线性代数方程组,

Romberg 方法求积分和自由曲线等 5 个上机实习题目．通过对方法的理解，输入输出格式和停机条件的设置，人机界面和方法流程图的设计，程序注释和上机调试，并以各种典型的例子在计算机上求解，去说明方法的可靠性与有效性．学生应将上机结果写成实验报告，包括：题目，方法的描述，流程设计，程序注释和程序清单，典型例子的选择，数值结果的报告和分析．通过上机，学生将体会到，在理论上已经证明是正确的计算方法，在计算机的具体计算中，有时会得到不可思议的错误结果．由此可见，计算机与其他事物一样，也有弱点和短处．通过上机，学生还将体会软件包装的重要性，并理解人机界面对监控计算机计算过程中出现异端情况的意义．

21 世纪即将到来了，如何在计算机上建立高精度、高效率的算法，如何去编写结构优良的程序使更多学科的研究和工程设计工作均被明白无误地表现为软件，更富有挑战性的课题是，如何利用计算机去探索与解释人类至今尚未知道的蕴藏在众多学科中的各种非线性的现象，这是明明白白地落在青年科技工作者的肩上了．

我们在从事计算方法的教学和研究中，积累了点滴体会与经验，愿与对计算机解题算法有兴趣的读者、青年科技工作者、老师与学生们一起切磋计算机解题的计算方法，求得共同的进步．

本书的编写分工为：第一、四、九、十章及附录由黄友谦编写；第二、三、五、六、七章由吴炯平编写；第八章由黄东斌编写．另外，中山大学计算机科学系 90～94 级学生的一些学习实践成果在本书中有所反映．

受水平限制，书中错漏之处，敬请读者批评指正．

1997 年 10 月

目　　录

第一章

§1 计算方法及其流程设计

在科学和技术领域里，许多问题都不能求出解的准确数值答案，例如解超越方程 $x=e^{-x}$. 我们必须寻求一个近似的方法也称为计算方法或数值方法来得到问题的解答. 解决一个复杂的数值计算问题的思想是**以直代曲，化繁为简**，这是计算方法的精髓. 可以理解为，用简单代替复杂，用直线逼近曲线，用有穷代替无穷，将非线性问题逐步线性化. 具体的实施方案是**近似求解，逐步求精**，换句话说，我们常用**数列**、**函数序列**、**向量序列**去逼近真解，其目的是将不可编程变为可编程，将“不可解”变为“可解”，将近似的方法归结为有穷步，由计算机去完成计算.

我们说，对某一数值问题建立了一个**计算机的算法**，它包含两个方面的内容：

（1）找到了解决数值问题的一个近

似方法，也称计算方法或数值方法；分析了方法的截断误差、收敛性、计算的稳定性和计算的复杂性.

(2) 设计了方法的计算流程和程序. 流程既是计算方法的准确和完整的描述，又能处理动态计算过程中出现的异端状态，如计算溢出，收敛太慢或方法病态等. **方法、流程、程序应尽量保持一致性、可读性、可维护性**. 一个友好的人机界面是一个好程序的重要标志.

例 1 给定 n 次代数多项式

$$P_n(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$$

设计一个算法求 $P_n(x),P_n'(x),P_n''(x)$的值.

解 **方案**(Ⅰ) 记

$$P_n(x)=\sum_{i=0}^{n}a_ix^{n-i}$$

有

$$P_n'(x)=\sum_{i=0}^{n-1}a_i(n-i)x^{n-i-1}$$

$$P_n''(x)=\sum_{i=0}^{n-2}a_i(n-i)(n-i-1)x^{n-i-2}$$

如果直接按照这三个式子来设计算法，要设置单元来存储 x，x^2，$\cdots x^n$，共需作 $n-1$ 次乘法，此外，在求 $P_n(x)$时还要增加 n 次乘法；在求 $P_n'(x)$时，要设置单元存储$(n-i)a_i$，$i=0$，1，…，$n-2$，这要作 $n-1$ 次乘法；进一步计算 $P_n'(x)$时还要增加 $n-1$ 次乘法；计算 $P_n''(x)$需作 $2(n-2)$次乘法. 故一共要作 $6n-7$ 次乘法，并且流程的设计将比较麻烦.

方案(Ⅱ) 我们来建立计算 $P_n(x),P_n'(x),P_n''(x)$的递推关系式，引进记号

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

有

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

对上面式子分别求一阶和二阶导数有

$$P_k'(x) = xP_{k-1}'(x) + P_{k-1}(x), P_0'(x) = 0$$
$$P_k''(x) = xp_{k-1}''(x) + 2P_{k-1}'(x), P_0''(x) = 0$$

分别设置三个单元 b_0, b_1, b_2 来存储 $P_n(x), P_n'(x), P_n''(x)$，用上面的递推计算方法计算（我国宋朝秦九韶曾提出求多项式 $P_n(x)$ 值的递推方法）：

秦九韶方法求 $P_n(x)$、$P_n'(x)$、$P_n''(x)$ 的流程

> **input** $a_0, a_1, \cdots, a_n, x$
>
> $b_0 \leftarrow a_0, b_1 \leftarrow 0, b_2 \leftarrow 0$
>
> **for** $k = 1, 2, \cdots, n$ **do**
>
> $\quad b_2 \leftarrow xb_2 + 2b_1$
>
> $\quad b_1 \leftarrow xb_1 + b_0$
>
> $\quad b_0 \leftarrow xb_0 + a_k$
>
> **end**
>
> **output** b_0, b_1, b_2

$P_n(x), P_n'(x), P_n''(x)$ 的值分别在单元 b_0，b_1，b_2 给出．我们来分析计算的复杂性，即分析乘法和除法的次数．对于每一个固定的 k，计算 b_2 要作 2 次乘法，计算 b_1，b_0 要分别作 1 次乘法，故共要作 4 次乘法，所以秦九韶方法求 $P_n(x), P_n'(x), P_n''(x)$ 共要作 $4n$ 次乘法，与方案（Ⅰ）相比，少做了 $2n-7$ 次乘法．更重要的是方案（Ⅱ）的流程简单，结构严谨，容易编程，可读性强．方法与流程保持一致，且容易推广到更高阶的导数．

例 2 记 n 阶上三角阵

$$A = \begin{bmatrix} u_1 & c_1 & & & \\ & u_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & u_{n-1} & c_{n-1} \\ & & & & u_n \end{bmatrix}$$

这个上三角阵除了主和次对角线外，其余元素都等于 0．假设它是

非奇异的，即 $u_i \neq 0$，$i=1, 2, \cdots, n$. 试设计一个算法求 A 的逆矩阵.

解 根据假设 A 是上三角阵，故 A^{-1} 也是上三角阵. 对 $\forall\ \boldsymbol{x} \in \mathbf{R}^n$，引进辅助向量

$$\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x} \Rightarrow \boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{y}$$

为了求 A^{-1}，仅需将关系式 $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$ 右端的 $x_1, x_2, \cdots, x_n$ 和左端的 $y_1, y_2, \cdots, y_n$ 调换，写成

$$y \leftrightarrow x$$

所以求 A^{-1} 等价于二个数组 $\boldsymbol{y}$ 和 $\boldsymbol{x}$ 的交换. 记

$$\boldsymbol{A}^{-1}=(u_{ij})_{n \times n}$$

其中，当 $j<i$ 时 $u_{ij}=0$. 将 $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$，$\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{y}$ 分别写成分量形式，记 $c_n=0$，有

$$y_i=u_i x_i+c_i x_{i+1}, \quad i=1, 2, \cdots, n \tag{1.1}$$

$$x_i=\sum_{j=i}^{n} u_{ij} y_j, \quad i=1, 2, \cdots, n \tag{1.2}$$

如何求出 A 的逆矩阵的元素 u_{ij} 呢？我们利用比较系数法. 由 (1.2) 式有

$$u_i x_i=u_i u_{ii} y_i+\sum_{j=i+1}^{n} u_i u_{ij} y_j$$

又
$$c_i x_{i+1}=\sum_{j=i+1}^{n} c_i u_{i+1,j} y_j$$

将这两个式子相加并利用 (1.1) 式对 $i=n, n-1, \cdots, 1$ 有

$$y_i=u_i u_{ii} y_i+\sum_{j=i+1}^{n}(u_i u_{ij}+c_i u_{i+1,j}) y_j \tag{1.3}$$

比较 (1.3) 式两端 y_i 和 y_j 的系数，当 $i=n$ 时有

$$y_n=u_n u_{nn} y_n \Rightarrow u_{nn}=1/u_n \tag{1.4}$$

对于 $i=n-1, n-2, \cdots, 1$ 有

$$u_i u_{ii}=1$$

$$u_i u_{ij}+c_i u_{i+1,j}=0, \quad j=i+1, i+2, \cdots, n$$

求得

$$u_{ii}=\frac{1}{u_i} \tag{1.4$'$}$$

$$u_{i,j}=-\frac{c_i}{u_i}u_{i+1,j},\quad j=i+1,\ i+2,\ \cdots,\ n \tag{1.4$''$}$$

(1.4) ～ (1.4)″式构成了求 A^{-1}的计算方法.

求具有主和次对角线的上三角阵的逆阵 $(\boldsymbol{u}_{ij})_{n\times n}$的流程

```
input u_1, …, u_n, c_1, …, c_n
u_nn ← 1/u_n
for i = n-1, n-2, …, 1 do
    u_ii ← 1/u_i
    l ← -c_i/u_i
    for j = i+1, i+2, …, n do
        u_ij ← l · u_{i+1,j}
    end
end
output u_ij, j = i, i+1, …, n; i = 1, 2, …, n
```

从流程中看出求 u_{ij}要作

$$\sum_{i=1}^{n-1}(n-i)=\frac{n(n-1)}{2}$$

次乘法. 求 u_{ii}及 l 要作 $2n-1$ 次乘除法，故求 A^{-1}共要求作

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

次乘除法的运算. 例如当 $n=4$ 又

$$\boldsymbol{A}=\begin{bmatrix} -2 & 1 & & \\ & -\frac{3}{2} & 1 & \\ & & -\frac{4}{3} & 1 \\ & & & -\frac{5}{4} \end{bmatrix}$$

时按上述流程计算如下：

$$u_{44}=\frac{1}{u_4}=-\frac{4}{5}$$

$i=3$ 时求得

$$u_{33}=\frac{1}{u_3}=-\frac{3}{4},\quad l=-\frac{c_3}{u_3}=\frac{3}{4}$$

$$u_{34}=l\cdot u_{44}=-\frac{3}{5}$$

$i=2$ 时求得

$$u_{22}=\frac{1}{u_2}=-\frac{2}{3},\quad l=-\frac{c_2}{u_2}=\frac{2}{3}$$

$$u_{23}=l\cdot u_{33}=-\frac{1}{2},\quad u_{24}=l\cdot u_{34}=-\frac{2}{5}$$

$i=1$ 时求得

$$u_{11}=\frac{1}{u_1}=-\frac{1}{2},\quad l=-\frac{c_1}{u_1}=\frac{1}{2}$$

$$u_{12}=l\cdot u_{22}=-\frac{1}{3},\quad u_{13}=l\cdot u_{23}=-\frac{1}{4}$$

$$u_{14}=lu_{24}=-\frac{1}{5}$$

共做 13 次乘除法，求得

$$\boldsymbol{A}^{-1}=\begin{bmatrix} -\frac{1}{2} & -\frac{1}{3} & -\frac{1}{4} & -\frac{1}{5} \\ & -\frac{2}{3} & -\frac{1}{2} & -\frac{2}{5} \\ & & -\frac{3}{4} & -\frac{3}{5} \\ & & & -\frac{4}{5} \end{bmatrix}$$

例 2 的计算方法与流程完全一致．流程简单，结构严谨，容易编程．例 2 中，我们没有用代数余子式的方法去求逆矩阵，而是根据计算机高级语言的特点，将求逆矩阵理解为数组的互相交换，用最简单的比较系数法，设计了一个具有递推关系、结构严

谨的计算方法.

这里，必须郑重指出，对一个问题有一个解法，如求行列式的 Gramer 法则，并不等于给出了计算机上求解的计算方法. **在设计计算机上解题的计算方法时应充分考虑高级语言的特点；计算机计算的舍入误差的积累；计算复杂性（即计算时间和存储空间）等因素的影响**. 纯数学的方法与计算机解题的计算方法是有区别的.

例 3 给定函数 $f(x)=\sqrt{x}$，$1\leqslant x\leqslant 2$. 试构造一个只包含四则运算的简单函数 $g(x)$ 来计算 $\sqrt{x}$，使 $|f(x)-g(x)|\leqslant\frac{1}{2}\times 10^{-9}$.

解 记

$$\varepsilon=\sqrt{x}-1,\quad 1\leqslant x\leqslant 2$$

易知 $0\leqslant\varepsilon<\sqrt{2}-1<1$

于是，当 n 充分大时 $(\sqrt{x}-1)^n=\varepsilon^n$ 便接近于 0. 取 $n=15$，将 $(\sqrt{x}-1)^n$ 作二项式展开，写为

$$(\sqrt{x}-1)^n=q(x)\sqrt{x}-p(x)=\varepsilon^{15}\tag{1.5}$$

其中

$$p(x)=15x^7+455x^6+3\,003x^5+6\,435x^4+5\,005x^3+1\,365x^2+105x+1$$

$$q(x)=x^7+105x^6+1\,365x^5+5\,005x^4+6\,435x^3+3\,003x^2+455x+15$$

由(1.5)式求得

$$\sqrt{x}=\frac{p(x)}{q(x)}+\frac{\varepsilon^{15}}{q(x)}\tag{1.6}$$

记 $g(x)=\frac{p(x)}{q(x)}$

这是一个有理函数，仅含四则运算，用它去逼近 $\sqrt{x}$ 的误差是

$$0<\sqrt{x}-g(x)=\frac{\varepsilon^{15}}{q(x)}\leqslant\frac{(\sqrt{2}-1)^{15}}{16\ 384}\leqslant\frac{1}{2}\times 10^{-9}$$

取若干点，验证如下：

x	$g(x)=\frac{p(x)}{q(x)}$	$f(x)=\sqrt{x}$
1	1.000000000	1.000000000
1.25	1.118033989	1.118033989
1.5	1.224744871	1.224744871
1.75	1.322875656	1.322875656
2	1.414213562	1.414213562

从（1.6）式看出，我们舍去了截断误差 $\varepsilon^{15}/q(x)$，它是一个无理函数，但按绝对值小于 $\frac{1}{2}\times 10^{-9}$，**化繁为简**，一个有理函数 $g(x)=p(x)/q(x)$ 便在允许的精度范围内逼近无理函数 $\sqrt{x}$.

请读者自己验明，若用 Taylor 展开式

$$(1+t)^{\alpha}=1+\alpha t+\frac{\alpha(\alpha-1)}{2!}t^2+\cdots+\frac{\alpha(\alpha-1)\cdots(\alpha-n+1)}{n!}t^n+\cdots$$

（这里 $\alpha=\frac{1}{2}$，又 $0\leqslant t\leqslant 1$）去计算 $\sqrt{x}$，要达到 $\frac{1}{2}\times 10^{-9}$ 的计算精度，n 要取多大.

对一个复杂的数值计算问题，只是给出一个方法是不够的，我们必须研究该方法在计算机上实现的可能性. 由于计算机表示一个数的字长是有限的，因此计算的每一步都会产生舍入误差，误差的积累是否影响计算结果的正确性呢？这便是方法的**计算稳定性**问题. 我们将在下一节讨论这个问题. 还有，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，这是**计算的复杂性问题**，它包含**时间复杂度**和**空间复杂度**两个概念.

时间的复杂度是方法需耗费时间的度量，即估计四则运算的