高等学校试用教材

数值分析简明教程

王能超 编

高等教育出版

高等学校试用教材

数值分析简明教程

王能超 编

高等教育出版社

内 容 提 要

本书是以《工程数学——算法语言·计算方法》(高等教育出版社1978年版)一书中的计算方法部分为基础，经过补充、修改编写而成.本书取材适当，写得深入浅出，通俗易懂，主要内容有：插值方法，数值积分，常微分方程的差分方法，方程求根的迭代法，线性方程组的解法等．在编排上本书注意贯穿了计算方法的思想，并附有算法框图和习题．全书讲授约45～50学时．本书是一本适合于一般工科专业学生使用的教材，也可供工程技术人员以及其他科技人员阅读参考．

高等学校试用教材

数值分析简明教程

王能超 编

*

高等教育出版社出版

新华书店北京发行所发行

北京印刷一厂印装

*

开本850×1168 1/32 印张 7 字数 160,000

1984年10月第1版 1985年4月第1次印刷

印数 00,001—16,200

书号 13010·01048 定价 1.55 元

前　言

人类社会正迈进电子计算机时代．在今天，熟练地运用计算机进行科学计算，已经成为广大科技工作者一项基本技能，这就需要向高等工科院校的学生普及有关计算方法的知识．本书正是为适应这一形势而编写的．

要提高运用计算机进行科学计算的能力，关键在于加强数学修养．不应当将计算方法片面地理解为各种算法的简单罗列和堆积，同数学分析一样，它也是一门内容丰富、思想方法深刻而有着自身的理论体系的数学学科．本书取名为数值分析正是基于这一认识．

本书是以《工程数学——算法语言·计算方法》(人民教育出版社1978年版）一书中的计算方法部分为基础，经过补充修改编写而成．

在教育部直属工科院校计算数学教材讨论会上（1983年，武汉)，曾对本书原稿进行了审议．参加审议的有清华大学、浙江大学、西安交通大学、大连工学院、南京工学院、天津大学、重庆大学和华侨大学等院校的老师．由西安交通大学游兆永教授负责主审．参加审议的同志在推荐本书出版的同时，还提出了许多宝贵的意见和建议，编者对此表示深切的谢意．

王　能　超

1984年12月25日，于华中工学院

目　录

引论…… 1
第一章　插值方法……18
§ 1　问题的提法 ……18
§ 2　拉格朗日插值公式 ……22
§ 3　插值余项 ……29
§ 4　埃特金算法 ……33
§ 5　牛顿插值公式 ……37
§ 6　埃尔米特插值 ……42
§ 7　分段插值法 ……47
§ 8　样条函数 ……52
§ 9　曲线拟合的最小二乘法 ……57
第二章　数值积分……66
§ 1　机械求积 ……66
§ 2　牛顿-柯特斯公式……71
§ 3　龙贝格算法 ……81
§ 4　高斯公式 ……89
§ 5　数值微分 ……97
第三章　常微分方程的差分方法…… 105
§ 1　尤拉方法 …… 105
§ 2　改进的尤拉方法 …… 110
§ 3　龙格-库塔方法…… 114
§ 4　亚当姆斯方法 …… 124
§ 5　收敛性与稳定性 …… 131

§ 6 方程组与高阶方程的情形 …… 134
§ 7 边值问题 …… 137
第四章 方程求根的迭代法 …… 140
§ 1 迭代原理 …… 140
§ 2 迭代过程的加速 …… 151
§ 3 牛顿法 …… 155
§ 4 弦截法 …… 162
第五章 线性方程组的解法 …… 166
§ 1 迭代公式的建立 …… 166
§ 2 向量和矩阵的范数 …… 176
§ 3 迭代过程的收敛性 …… 181
§ 4 消去法 …… 186
§ 5 追赶法 …… 201
§ 6 平方根法 …… 208
§ 7 误差分析 …… 213

引　论

科学技术的发展提出大量复杂的数值计算问题，这些问题的解决不是人工手算（包括使用算盘以及计算器一类简单的计算工具）所能胜任的，必须依靠电子计算机．用电子计算机进行这种科学技术计算的工作，称为科学计算，或简称电算．

科学计算的应用范围非常广泛，国防尖端的一些科研项目，如核武器的研制，导弹的发射等等，始终是科学计算最为活跃的领域．今天，科学计算在工农业生产的各个部门也正在发挥日益重要的作用．

例如，气象资料的汇总、加工并求得天气图象，这方面工作的计算量大而且时间性强，要求电子计算机作高速或超高速运算，以对天气作出短期及中期预报．

又如，将所设计的船型型体数值表转换成初始数据输入电子计算机，经过计算即可求出外板和肋骨的展开数据．在造船工业中用这种方法进行数学放样，既节省了人力物力，又缩短了设计周期．

本门课程将着重介绍进行科学计算所必须掌握的一些最基本、最常用的算法．

一、算　法

1．研究算法的意义

电子计算机的运算速度高，可以承担大运算量的工作，但这是否意味着计算机上的算法可以随意选择呢？

我们知道，行列式解法的克莱姆（Cramer）法则原则上可用

来求解线性方程组，用这种方法求解一个 n 阶方程组，要算 $n+1$ 个 n 阶行列式的值，总共需要 $n!(n-1)(n+1)$ 次乘法．当 n 充分大时，计算量是相当惊人的．譬如一个20阶不算太大的方程组，大约要做 10^{21} 次乘法，这项计算即使用每秒百万次的电子计算机去做，也得要连续工作千百万年才能完成．当然这是完全没有实际意义的．其实，解线性方程组有许多实用的算法(参看第五章)，譬如用众所周知的消元法，一个20阶的方程组即使用一台小型计算器也能很快地解出来．这个简单的例子告诉我们，能否正确地制定算法是科学计算成败的关键．

2．什么是算法

针对一个具体的数学问题，可以给出多种解法．

例 1　证明二次方程

$$x^2+2bx+c=0 \tag{1}$$

至多有两个不同的实根．

解　下面提供三种解法．

1) **反证法**　假定方程(1)有三个互异的实根 x_1，x_2 和 x_3，则有

$$x_1^2+2bx_1+c=0$$
$$x_2^2+2bx_2+c=0$$
$$x_3^2+2bx_3+c=0$$

以上式子两两相减得

$$x_2+x_1+2b=0$$
$$x_3+x_2+2b=0$$

从而有　$x_1=x_3$，这与原设矛盾．证毕．

2) **图解法**　方程(1) 配方得

$$(x+b)^2+c-b^2=0 \tag{2}$$

在坐标纸上描出抛物线 $y=(x+b)^2+c-b^2$，它与 x 轴的交点

(横坐标)即为所求的实根，而交点至多只有两个.

3) 公式法　据(2)可导出直接的求根公式

$$x_{1,2}=-b\pm\sqrt{b^2-c} \tag{3}$$

上述三种方法,反证法不是构造性的；作图法虽是构造性的,但不是数值的. 我们所说的算法，必须是构造性的数值方法，即不但要论证问题的可解性，而且解的构造是通过数值演算过程来完成的.

我们所要研究的算法是为电子计算机提供的，因此，解题方案当中的每个细节都必须准确地加以定义，并且要完整地描述整个解题过程. 我们所说的“算法”,不仅仅是单纯的数学公式，而是指解题方案的准确而完整的描述.

描述算法可以用多种方式. 本书常用框图直观地显示算法的全貌.

譬如，设要求解二次方程(1).我们将依判别式 $d=b^2-c$ 的符号区分下列三种情况：

1) 若 $d<0$，这时无实根；

2) 若 $d=0$，这时有重根 $x_1=x_2=-b$；

3) 若 $d>0$,这时可用公式(3)获得两互异实根 x_1，x_2.

图0-1形象地描述了上面的算法.

这里我们使用了两种形式的框. 一种是矩形框 [　　]，称叙述框. 计算公式就填在这种框内. 另一种是圆边框(　　)，称检查框，表示算法的判断检查部分. 检查框有两个出口，究竟选择那个出口，要看框内的检查条件是否成立来决定.

今后所有的框图均以 [开始] 框标志计算过程开始启动，而用 [结束] 框表示计算过程的最终结束. 另外,我们将用箭头“⟶”指明各框执行的顺序.

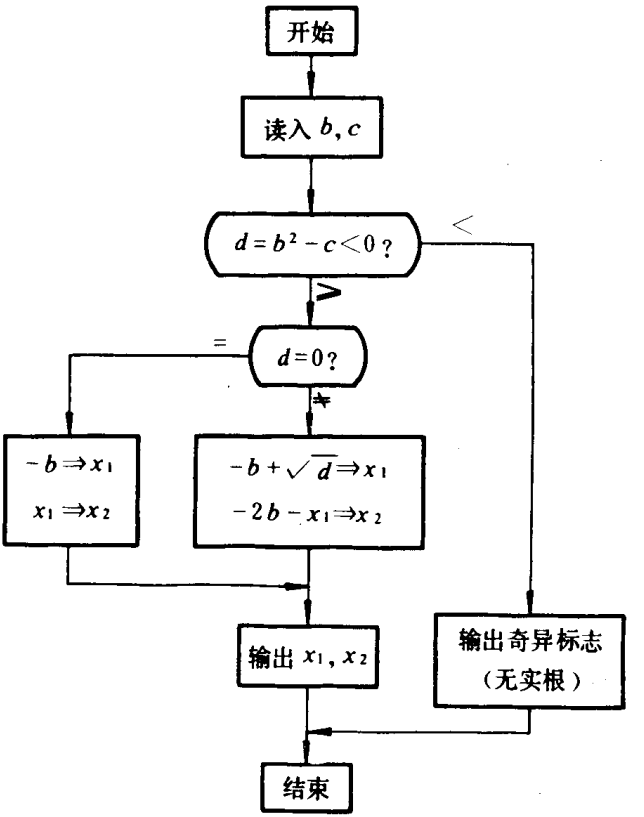

图 0-1

下面我们剖析两个常用算法来阐述算法的基本特征.

3. 多项式求值的秦九韶方法

计算公式通常是算法的核心部分. 计算机上使用的算法，其计算公式常采取递推化形式. 递推化的基本思想是将一个复杂的计算过程归结为简单过程的多次重复，这种重复在算法上表现为

循环，描述是容易的.

譬如，设要对给定的x求下列多项式的值

$$p(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n \tag{4}$$

一种看起来很“自然”的算法是直接逐项求和. 我们用t_k表示x的k次幂，u_k表示(4)式右端前$k+1$项的部分和：

$$t_k=x^k$$

$$u_k=a_0+a_1x+\cdots+a_kx^k$$

则

$$\begin{cases} t_k=x\cdot t_{k-1}, \\ u_k=u_{k-1}+a_kt_k, \end{cases} \quad (k=1,2,\cdots,n) \tag{5}$$

作为初值，令

$$\begin{cases} t_0=1 \\ u_0=a_0 \end{cases} \tag{6}$$

利用初值(6)对$k=1,2,\cdots,n$，反复执行算式(5)，最终得出的u_n就是所求的值$p(x)$.

统计上述算法的计算量. 由于加减操作的机器运行时间比乘除操作少得多，这里在统计计算量时，我们忽略加减法而只统计乘除法的次数. 递推公式(5)的每一步需做两次乘法，因此总的计算量为$2n$次乘法.

下面再介绍一种求值方案. 为此首先加工计算公式，设将式(4)按降幂的顺序重排：

$$p(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$$

从它的前n项提出x，则有

$$p(x)=(a_nx^{n-1}+a_{n-1}x^{n-2}+\cdots+a_1)x+a_0$$

经过这个手续，括号内得到的是一个$n-1$次多项式(注意降了一次). 如果对括号内再施行同样的手续，进一步有

$$p(x)=((a_nx^{n-2}+a_{n-1}x^{n-3}+\cdots+a_2)x+a_1)x+a_0$$

这样每做一步，最内层的多项式降低一次，最终可加工成如下嵌套形式

$$p(x)=(\cdots(a_n x+a_{n-1})x+\cdots+a_1)x+a_0 \qquad (7)$$

我们利用形式(7)结构上的特点，从里往外一层一层地计算．设用v_k表示第k层(从里面数起)的值：

$$v_k=(\cdots(a_n x+a_{n-1})x+\cdots+a_{n-k+1})x+a_{n-k}$$

那么，第k层的结果v_k显然等于第$k-1$层的结果v_{k-1}乘上x再加上系数a_{n-k}：

$$v_k=x\ v_{k-1}+a_{n-k}, \qquad k=1,2,\cdots,n \qquad (8)$$

作为初值，这里令

$$v_0=a_n \qquad (9)$$

比较(5)—(6)和(8)—(9)两种算法，后一种不但逻辑结构简单，而且计算量节约了一半．

多项式求值的这种算法称作秦九韶算法，它是我国宋代一位数学家秦九韶最先提出的．秦九韶算法的特点在于，它通过一次式的反复计算，逐步得出高次多项式的值．具体地说，它将一个n次多项式(4)的求值问题，归结为重复计算n个一次式(8)来实现．这种化繁为简的处理方法在数值分析中是屡见不鲜的．

现在考虑秦九韶方法的计算程序．

按式(8)计算，每求出一个“新值”v_k以后，“老值”v_{k-1}便失去继续保存的价值，因此我们可以将新值v_k存放在老值v_{k-1}所占用的单元内．这样，我们只要设置一个单元v进行累算，而将(8)式表为下列动态形式

$$x\cdot v+a_{n-k}\Rightarrow v, \qquad k=1,2,\cdots,n$$

执行这组算式之前，应先送初值a_n到单元v中

$$a_n\Rightarrow v$$

图0-2描述了秦九韶算法，其中：

[框 1] **准备部分**. 单元 v 中送初值 a_n，单元 k 中送计数值 1.

[框 2] **计算部分**. 每循环一次，单元 v 中的老值 v_{k-1} 为新值 v_k 所替换.

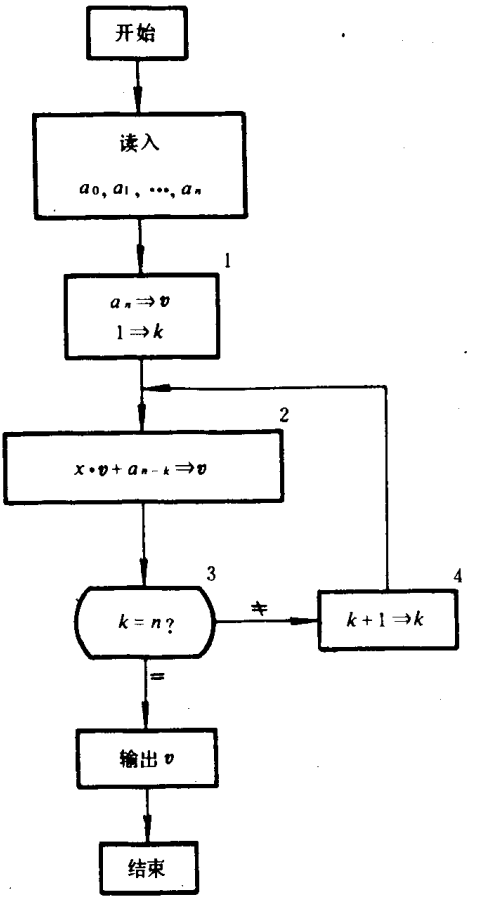

图 0-2

[框 3] **控制部分**. 检查单元 k 中计数值以判断循环应否结束. 当计数值为 n 时输出 v 中结果，否则转框 4.

[框 4] **修改部分**. 修改单元 k 中计数值，然后转框 2 再作下一步的计算.

4. 方程求根的二分法

许多实际算法表现为某种无穷递推过程的截断，实现这类算法，不但需要建立计算公式，还需要解决精度控制问题.

设函数 $f(x)$ 在 $[a, b]$ 上连续，且 $f(a)f(b)<0$，根据连续函数的性质，$f(x)$ 在 $[a, b]$ 内一定有实的零点，即方程 $f(x)=0$ 在 $[a, b]$ 内一定有实根. 我们这里假定它在 $[a, b]$ 内有唯一的单实根 x^*.

考察有根区间 $[a, b]$，取中点 $x_0=(a+b)/2$ 将它分为两半，然后进行根的搜索，即检查 $f(x_0)$ 与 $f(a)$ 是否同号：如果确系同号，说明所求的根 x^* 在 x_0 的右侧，这时令 $a_1=x_0$，$b_1=b$；否则

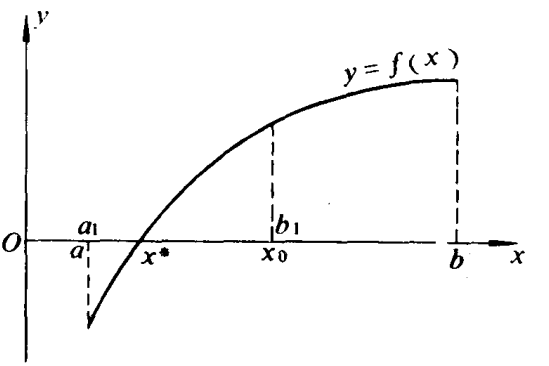

图 0-3

x^*必在x_0的左侧，这时令$a_1=a$，$b_1=x_0$(图0-3).不管出现那一种情形，新的有根区间$[a_1,b_1]$的长度仅为$[a,b]$的一半.

对压缩了的有根区间$[a_1,\ b_1]$又可施行同样的手续，即用中点$x_1=(a_1+b_1)/2$将区间$[a_1,\ b_1]$再分为两半，然后通过根的搜索判定所求的根在x_1的那一侧，从而又确定一个新的有根区间$[a_2,\ b_2]$，其长度是$[a_1,\ b_1]$的一半.

如此反复二分下去，即可得出一系列有根区间

$$[a,b]\supset[a_1,b_1]\supset[a_2,b_2]\supset\cdots\supset[a_k,\ b_k]\supset\cdots$$

其中每个区间都是前一个区间的一半，因此二分k次后的有根区间$[a_k,\ b_k]$的长度

$$b_k-a_k=\frac{1}{2^k}(b-a)$$

可见，如果二分过程无限地继续下去，这些有根区间最终必收缩于一点x^*，该点显然就是所求的根.

每次二分后，设取有根区间的中点

$$x_k=\frac{1}{2}(a_k+b_k)$$

作为根的近似值，则在二分过程中可以获得一个近似根的序列x_0，x_1，x_2，…，该序列以根x^*为极限.

不过在实际计算时，我们不可能完成这种无穷过程，其实也没有这种必要，因为数值分析的结果允许带有一定的误差，由于

$$|x^*-x_k|\leqslant\frac{1}{2}(b_k-a_k)=\frac{1}{2^{k+1}}(b-a)$$

只要二分足够多次(即k充分大)，便有

$$|x^*-x_k|<\varepsilon$$

这里ε为预定精度.

上述求根方法称二分法，它是电子计算机上一种常用算法.

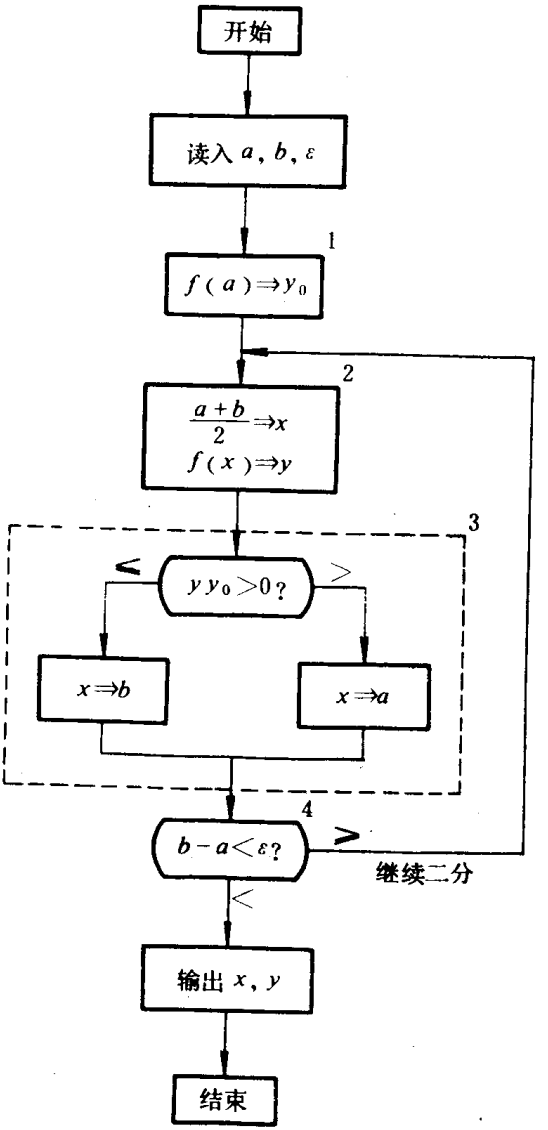

图 0-4

我们给出其算法框图．在图0-4中，a，b表示有根区间的左右端点；x表示近似根．

各框的具体含义如下：

[框 1] 从所给区间(a, b)着手二分．

[框 2] 取有根区间(a, b)的中点x作为近似根．

[框 3] 确定二分后新的有根区间(a, b)．

[框 4] 检查近似根x是否满足精度要求．

例 2　用二分法求方程$x^3-x-1=0$在区间$[1, 1.5]$内的一个实根，要求误差不超过0.005．

解　首先预估所要二分的次数．按误差估计式

$$|x^*-x_k| \leqslant \frac{1}{2^{k+1}}(b-a)$$

只要二分 6 次，便能达到所要求的精度．

二分法的计算结果如下表：

表 0-1

k	a_k	b_k	x_k
0	1.0000	1.5000	1.2500
1	1.2500		1.3750
2		1.3750	1.3125
3	1.3125		1.3438
4		1.3438	1.3281
5		1.3281	1.3203
6	1.3203		1.3242

二、误　差

1. 误差分析不容忽视

在研究算法的同时，必须注重误差分析，否则，一个合理的算法也可能得出错误的结果．