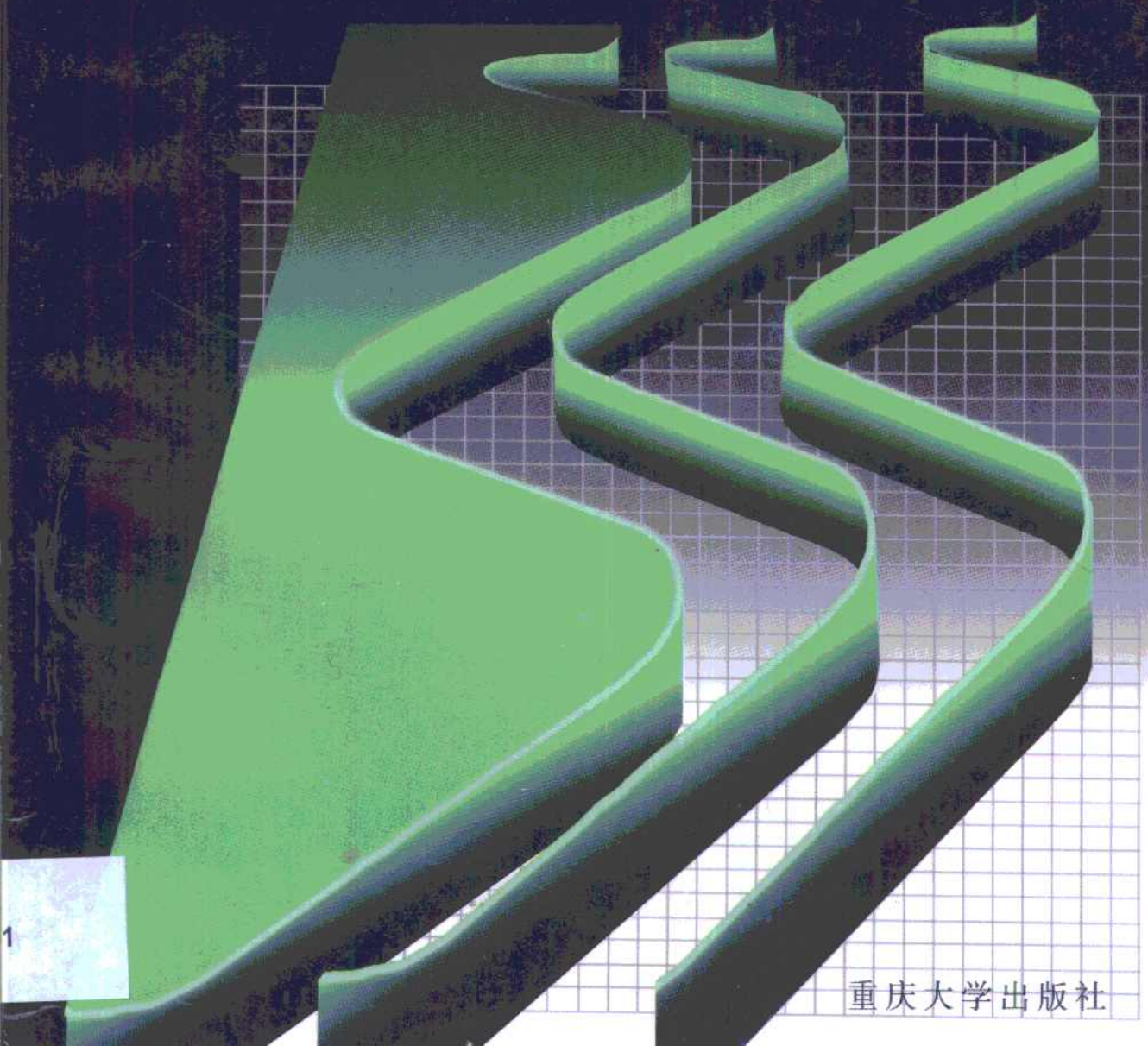
工程硕士研究生系列教材

实用数值分析

SHIYONG SHUZHI FENXI

杨大地　谈骏渝　编著



重庆大学出版社

工程硕士研究生系列教材

实用数值分析

杨大地　谈骏渝　　编著

重庆大学出版社

内 容 提 要

本书系统地介绍了数值计算的基本概念、常用算法及有关的理论分析，重点讨论了工程上实用的行之有效的具体算法的理论分析和实际应用。全书共分九章，主要内容包含了数值计算的基本问题：算法和误差、线性方程组数值解、矩阵特征值和特征向量算法、非线性方程数值解、插值方法、数据拟合和函数逼近、数值积分和数值微分、常微分方程数值解法等。本书基本概念清楚，理论分析简明，文字叙述深入浅出，循序渐进，注重算法的实际应用。各章都给出了典型例题并配有一定数量的习题。对大部分算法都给出了详细的计算步骤，有的还给出了流程图或算法语言描述，便于读者更好地掌握课程的基本内容。

本书是为高等理工科院校的工程硕士研究生编写的教科书，也可作为理工科大学生及研究生的数值分析课程的教材，或可为工程技术人员参考使用。

图书在版编目(CIP)数据

实用数值分析/杨大地，谈骏渝编著.—重庆：重庆大学出版社，2000.2
工程硕士研究生系列教材
ISBN 7-5624-2095-5

Ⅰ.实… Ⅱ.①杨… ②谈… Ⅲ.计算方法-研究生-教材
Ⅳ.0241

中国版本图书馆 CIP 数据核字(2000)第 12920 号

工程硕士研究生系列教材
实用数值分析
杨大地　谈骏渝　编著
责任编辑　肖顺杰
*
重庆大学出版社出版发行
新　华　书　店　经　销
重庆建筑大学印刷厂印刷
*
开本：787×1092　1/16　印张：12.75　字数：318 千
2000年2月第1版　　2000年2月第1次印刷
印数：1－6 000
ISBN 7-5624-2095-5/O·179　定价：20.00元

前　言

随着现代科学技术以及电子计算机的迅速发展，数值计算方法已得到了越来越普遍的发展和应用。数值分析课程也已成为理工科院校的重要课程。由于没有适合工程硕士研究生使用的教材，过去我们使用的仍然是工学硕士研究生的《数值分析》教材。为适应工程硕士研究生教学的需要，我们特编写了此教材。

在科学研究和工程实践中碰到的函数是各种各样的，有的表述式很复杂，有的甚至给不出精确的数学式子。对于许多实际问题，用传统的经典方法很难加以解决。有的从理论上讲，尽管问题的解存在，但是却难以找到它的精确解或解析解。这样，就提出了在一定的条件下，如何去寻求或构造这些函数的近似表达式，或者寻求这些问题的近似解。这些问题都要求进行大量而复杂的数值计算，依靠计算机去进行计算才能胜任和解决。我们把用电子计算机进行这种科学技术计算的工作，称之为科学计算，或简称电算。

本书的编写，考虑到工程硕士生的特点，注意突出了实用性。所介绍的内容都是科学技术或工程实际中常用的数值计算方法。掌握这些计算方法，对于解决实际问题无疑是十分重要的。另一方面，本书也介绍了必要的基础理论或结论。由于主要侧重于算法的叙述和算例分析，而省略了某些数学上的繁琐证明过程。尽管如此，理解和掌握这些基本的理论和知识，有助于加深对算法的认识和理解，更好地应用这些算法。作为数值分析的一个重要方面是误差和误差分析，本书所介绍的每种算法都对这一方面作了介绍。数值计算实际上都是近似计算，由于各种因素的影响必然会产生误差，实际问题所需要的数值计算结果，都要满足一定的精度要求，因此，在学习算法的同时，还要知道如何算才能达到预定的精度要求。也就是还要求知道计算结果的误差，同时会进行误差分析。这些对于工程和科学计算都是十分重要的。

学习本门课程，应该注意到本门课程的以下几个特点：解决一个数值计算问题的主要途径，是用有穷和去近似代替无穷和，或者是将非线性问题逐次线性化，或者将连续的过程离散化等。也就是用数列，函数序列，向量序列等去逼近问题的真解。通过这些途径，应用具体的算法将问题化为有限步，由计算机去实现。

本书在叙述上，一般是按问题的背景描述一问题的提出一计算方法及公式一算法描述一数值算例这样的方式来叙述。在学习过程中，抓住这一主线，有助于掌握书中所介绍的内容。在算法的描述上，我们是以数学公式加以必要的文字说明为主，有的给出了流程框图，对一些算法还给出了算法语言描述。这样，有助于对算法有更深刻，更直观的理解，同时也有利于将算法编写成程序。

本书是作者在近几年对理工科本科生、研究生，尤其是对工程硕士研究生讲授《数值分析》课的基础上，结合我们的教学经验和体会，参考和借鉴了国内外有关的教材而编写的。全书设计讲授时数为 50 ~ 72 学时，针对不同的专业，学时可以根据课时和实际需要选取其中的一些章节。对目录中带 * 的章节可以少讲或不讲。本书编写时已注意到各章节的独立性，删掉带 * 的章节不致于影响其他章节的学习。

全书共分九章。第一章主要论述了算法及误差的基本概念、特点以及基本设计方法和原则，并介绍了一种算法描述语言。第二章到第九章介绍了工程上常用的数值计算方法。对于其中的主要算法进行了详细的讨论，并给出了算法框图和算法描述。所介绍的这些基本算法虽然不可能解决工程实际中所遇到的全部问题，但只要读者理解并实践了这些常用算法，就可以方便地在计算机上应用这些算法处理通常所遇到的数学问题。各章后均附有习题，在有条件的情况下，学生可配合习题编写计算机程序，在计算机上检验算法的正确性。

作者在编著的过程中自始至终得到我校研究生部和理学院领导的关心和支持，许多同志对本书的编写也给予了热情帮助，这里一并表示衷心的感谢。

由于计算数学发展迅速，我们的水平有限，本书的编写又比较仓促，肯定会存在许多不足之处，我们热情地希望读者批评指正并提出宝贵的意见和建议。

编著者

2000 年元月

第一章　绪　论

运用数学方法解决科学研究或工程技术问题，一般按如下途径进行：

实际问题→模型设计→算法设计→程序设计→上机计算→问题的解

其中算法设计是数值分析课程的主要内容．

数值分析课程研究常见的基本数学问题的数值解法．包含了数值代数(线性方程组的解法、非线性方程的解法、矩阵求逆、矩阵特征值计算等)、数值逼近、数值微分与数值积分、常微分方程及偏微分方程的数值解法等．它的基本理论和研究方法建立在数学理论基础之上，研究对象是数学问题，因此它是数学的分支之一．

但它又与计算机科学有密切的关系．我们在考虑算法时，往往要同时考虑计算机的特性，如计算速度、存贮量、字长等技术指标，考虑程序设计时的可行性和复杂性．如果已经具备了一定的计算机基础知识和程序设计方法，学习数值分析的理论和方法就会更深刻、更实际，选择或设计的算法也会更合理、更实用．

在科学研究、工程实践和经济管理等工作中，存在大量的科学计算、数据处理等问题．应用计算机解决数值计算问题是理工科学生应当具备的基本能力．

第一节　算　法

解决某类数学问题的数值方法称为算法，它是解题过程的完整而准确的描述．为使算法能在计算机上实现，它必须将一个数学问题分解为有限次的 $+$、$-$、$\times$、$\div$ 运算和一些简单的基本函数运算．

一、算法的表述形式

算法的表述形式可以用多种方式．

1)用数学公式和文字说明描述，这种方式符合人们的理解习惯，和算法的推证相衔接，易于学习接受，但离上机应用距离较大．

2)用框图描述，这种方式描述计算过程流向清楚，易于编制程序，但对初学者有一个习惯过程．此外框图描述格式不很统一，详略也不尽相同．

3)算法描述语言，它是表述算法的一种通用的语言，有特定的表述程序和语句．它可以很容易地转换某种实用的计算机高级语言，同时也具有一定的可读性．在本节三中我们将作介

绍．

4)算法程序，即用计算机语言描述的算法．它是面对计算机的算法，可以是印刷文本，也可以是磁介质上存贮的文件．实际上我们以后讨论的算法，都有现成的程序文本和软件可资利用．一个合格的计算工作者，应当能熟练地应用这些已有的软件工具．但从学习算法的角度看，这种描述方式并不有利．

必须指出，上面介绍的前三种算法描述方式，都只是面向读者，不能直接用于计算机，但它们又是编制计算机算法程序并进行计算的前提和基础．本教材将介绍适用于计算机进行计算的数值计算方法以及误差分析等知识．在此基础上，将分别采用前三种方式表述各种算法，以便对算法的各种表述方式有所了解和掌握．应该说，能否正确地制定和使用算法是科学计算成败的关键．

二、算法的基本特点

算法要准确而全面地描述整个解题过程，它具有动态特征，其基本特点是：

1)算法常表现为一个无穷过程的截断

例 1 计算 $\sin x$ 的值，$x \in \left(0, \frac{\pi}{4}\right)$

解 据泰勒公式：

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \cdots \tag{1-1}$$

这是一个无穷级数，只能在适当的地方“截断”，使计算量不太大，而精度又能满足要求．

如计算 $\sin 0.5$，取 $n=3$

$$\sin 0.5 \approx 0.5 - \frac{0.5^3}{3!} + \frac{0.5^5}{5!} - \frac{0.5^7}{7!} = 0.479\,426$$

据泰勒余项公式，它的误差应为

$$R = (-1)^4 \frac{\xi^9}{9!} \quad \xi \in \left(0, \frac{\pi}{4}\right) \tag{1-2}$$

$$|R| \leqslant \frac{\left(\frac{\pi}{4}\right)^9}{362\,880} = 3.13 \times 10^{-7}$$

可见结果是相当精确的．实际上，结果中的 6 位数字都是正确的．

2)算法常表现为一个连续过程的离散化

例 2 计算积分值

$$I = \int_0^1 \frac{1}{1+x} \mathrm{d}x$$

解 见图 1-1，将[0,1]分为 4 等分，分别计算 4 个小曲边梯形的面积的近似值，然后加起来作为积分的近似值．记被积函数为 $f(x)$，即

$$f(x) = \frac{1}{1+x}$$

取

$$h = \frac{1}{4}, \text{有 } x_i = ih,\ i = 0,1,2,3,4$$

$$T_i = \frac{f(x_i) + f(x_{i+1})}{2}h, \quad i = 0,1,2,3.$$

所以有

$$I \approx \sum_{i=0}^{3} T_i \tag{1-3}$$

计算有：$I \approx 0.697\ 024$ 与精确值0.693 147比较，可知结果不够精确，如进一步细分区间，精度可以提高．

3)算法常表现为“迭代”形式．迭代是指某一简单算法的多次重复，后一次使用前一次的结果．这种形式易于在计算程序中实现，在程序中表现为“循环”过程．

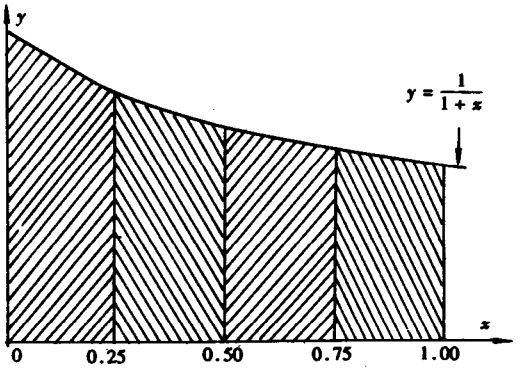


图 1-1

例 3 多项式求值

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \tag{1-4}$$

解 用 t_k 表示 x^k，u_k 表示(1-4)式前 $k+1$ 项之和．作为初值，令：

$$\begin{cases} t_0 = 1 \\ u_0 = a_0 \end{cases} \tag{1-5}$$

对 $k = 1,2,\cdots,n$，反复执行：

$$\begin{cases} t_k = x t_{k-1} \\ u_k = u_{k-1} + a_k t_k \end{cases} \tag{1-6}$$

显然 $P_n(x) = u_n$，而 1-6 式是一种简单算法的多次循环．

对此问题还有一种更好的迭代算法．

$$\begin{aligned} P_n(x) &= a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \\ &= (a_n x^{n-1} + a_{n-1} x^{n-2} + \cdots + a_1)x + a_0 \\ &= ((a_n x^{n-2} + a_{n-1} x^{n-3} + \cdots + a_2)x + a_1)x + a_0 \\ &= (\cdots(a_n x + a_{n-1})x + \cdots + a_1)x + a_0 \end{aligned}$$

令

$$\begin{cases} v_0 = a_0 \\ v_k = x v_{k-1} + a_{n-k} \end{cases} \qquad k = 1,2,3,\cdots,n \tag{1-7}$$

显然 $P_n(x) = v_n$．

这两种算法都是将 n 次多项式化为 n 个一次多项式来计算，这种化繁为简的方法在数值分析中经常使用．

下面估计一下以上两种算法的计算量：

第一法　执行 n 次(1-6)式，每次 2 次乘法，一次加法，共计 $2n$ 次乘法，n 次加法；

第二法　执行 n 次(1-7)式，每次 1 次乘法，一次加法，共计 n 次乘法，n 次加法．

显然第二种方法运算量小，它是我国宋代数学家秦九韶最先提出的，被称为“秦九韶算法”．

例 4 不用开平方计算$\sqrt{a}$ $(a>0)$的值．

解　假定 x_0 是$\sqrt{a}$的一个近似值，$x_0>0$，则$\frac{a}{x_0}$也是$\sqrt{a}$的一个近似值，且 x_0 和$\frac{a}{x_0}$两个近似值必有一个大于$\sqrt{a}$，另一个小于$\sqrt{a}$，可以设想它们的平均值应为$\sqrt{a}$的更好的近似值，于是设计一种算法：

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{a}{x_k}\right)\qquad(k=0,1,2,\cdots)\tag{1-8}$$

如计算$\sqrt{3}$，取 $x_0=2$，有

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{3}{x_k}\right)\qquad(k=0,1,2,\cdots)$$

计算有：

$$x_0=2$$
$$x_1=1.75$$
$$x_2=1.732\ 142\ 9$$
$$x_3=1.732\ 050\ 8$$
$$\cdots$$

可见此法收敛速度很快，只算三次得到 8 位精确数字．

迭代法应用时要考虑是否收敛、收敛条件及收敛速度等问题，今后课程将进一步讨论．

三、算法描述语言

前面指出，描述算法可以用多种方式，算法描述语言是表示算法的一种工具．在本书中，部分算法将采用本节所介绍的算法描述语言来描述．

在用算法描述语言描述一个算法时，如果被描述的算法可以用某一过程来实现，则一开始总是先要对输入与输出的参数作必要的说明．并且第一行总是为：

PROCEDURE　过程名(输入输出参数表)

在算法正文中，必要时对算法中的每一行用自然数顺序依次编上行号，以便在叙述中对算法予以说明．

(1)符号与表达式

符号是以字母开头的字母和数字的有限串，用以表示变量名、数组名等，必要时也用来表示语句标号．

在语句标号后应跟随一个冒号，然后是语句，例如：　$loop:i\leftarrow i+1$

在算法中，变量或数组的类型通常可以从上下文看出，因此一般不予说明，除非在特殊情况下才作必要说明．

有时，算法中的某些指令或某个子功能直接用叙述的方式给出，以便使算法更简洁．例如，“设 x 是 A 中的最大项”(其中 A 是一个数组)，“将 x 插入 L 中”(其中 L 是某个数组)等等．

算术运算符通常用 +，−，*，/和↑(乘方)，(其中 * 和↑一般可省略，而沿用通常的数学表示法．

关系运算符通常用 =，≠，<，>，≤，和≥来表示；

逻辑运算符用 and(与)，or(或)和 not(非)来表示，

(2)赋值语句

赋值语句的形式为　　$a \leftarrow e$

其中 a 表示变量名或数组元素，e 表示数学表达式或逻辑表达式．

如果 a 和 b 都是变量名或数组元素，则可用记号 $a \leftrightarrow b$ 表示 a 与 b 的内容互换；而用记号 $a \leftarrow b \leftarrow e$ 则表示将 e 的内容同时赋给 a 和 b．

(3)控制转移语句

无条件转移语句用 GOTO 〈标号〉，它导致转到具有指定标号的语句去执行．

条件控制语句有以下两种形式

IF c THEN s

或

IF c THEN $s1$
　　ELSE $s2$

其中表示 c 是一个逻辑表达式；s，$s1$，$s2$ 可以是单一的语句，也可以是用一对花括符{}括起来的语句组．如果 c 为“真”，则 s 或 $s1$ 被执行一次；如果 c 为“假”，则在第一种形式，IF 语句不执行任何语句，即告完成；而在第二种形式中，将执行 $s2$ 一次．

一般情况下，IF 语句执行完后，控制将进行到下一句．除非 s，$s1$ 或 $s2$ 中有其它的控制转移到别的地方．

(4)循环语句

循环语句有两种形式：一是 WHILE 语句；一是 FOR 语句．

WHILE 语句的形式为

WHILE c DO s

其中 c 是逻辑表达式；s 可以是单一的语句，也可以是用一对花括符{}括起来的语句组．如果 c 为“真”，则执行 s，且在每次执行 s 之后都要重新检查 c；如果 c 为“假”，控制就转到紧跟在 WHILE 语句之后的语句．当控制第一次到达 WHILE 时 c 为“假”，则 s 一次也不执行．

WHILE 语句的功能相当于如下的 IF 语句：

```
loop:  IF c  THEN
           {s;
           GOTO loop
           }
```

FOR 语句的形式为

FOR i = *init* TO *limit* BY *step* DO s

其中 i 是循环控制变量，*init*，*limit* 和 *step* 分别表示循环的初值，终值和步长，它们都可以是算术表达式；s 可以是单一的语句，也可以是用一对花括符{}括起来的语句组．

当 *step* > 0 时，FOR 语句的功能等价于如下的 IF 语句：

```
       i = init
loop:  IF i ≤ limit THEN
       {s;
       i ← i + step;
       GOTO loop
       }
```

当 *step* < 0 时，FOR 语句的功能等价于如下的 IF 语句：

$i = init$

loop: IF $i \geqslant limit$ THEN

{ s;

$i \leftarrow i + step$;

GOTO *loop*

}

在 FOR 语句中如果 *step* = 1，则 BY *step* 可以省略不写，变为：

FOR $i = init$ TO *limit* DO s

(5)其它语句

EXIT 语句通常用于退出某个循环，使控制转到包含 EXIT 语句的最内层的 WHILE 或 FOR 循环后面的一个语句．

RETURN 语句用于结束算法的执行．如果算法是在最后一行结束，则有时省略最后的 RETURN 语句．并且，在 RETURN 语句后面还允许使用用引号括起来的注释信息．

READ(或 INPUT)和 OUTPUT(或 WRITE)语句用于输入和输出．

在算法描述过程中，还可能用到其它一些语句，读者完全可以了解它们的意义．最后还要说明：算法中的注释总是用一对方括符[]括起来；几个短语句可以写在一行上，彼此之间用分号隔开；复合语句总是用一对花括符{}括起来作为语句组．

最后，我们用算法的三种描述方式分别描述求解一元二次方程：

$$x^2 + 2bx + c = 0$$

实根的全过程．

(1)用公式加文字说明

1)输入 b, c

2)计算 $d = b^2 - c$

3)判断

①若 $d < 0$，无实数根；

②若 $d = 0$，输出实数根 $x_1 = x_2 = -b$；

③若 $d > 0$，用公式 $x_{1,2} = -b \pm \sqrt{d}$ 得到两个互异的实根并输出 x_1, x_2.

(2)用算法描述语言

PROCEDURE *pros*1 (b, c, x_1, x_2)

INPUT b, c

$d \leftarrow b^2 - c$

IF $d = 0$ THEN $\{x_1 \leftarrow x_2 \leftarrow -b$; OUTPUT $x_1, x_2\}$

ELSE {IF $d > 0$ THEN $\{x_1 \leftarrow -b + \sqrt{d}; x_2 \leftarrow -2b - x_1$; OUTPUT $x_1, x_2\}$

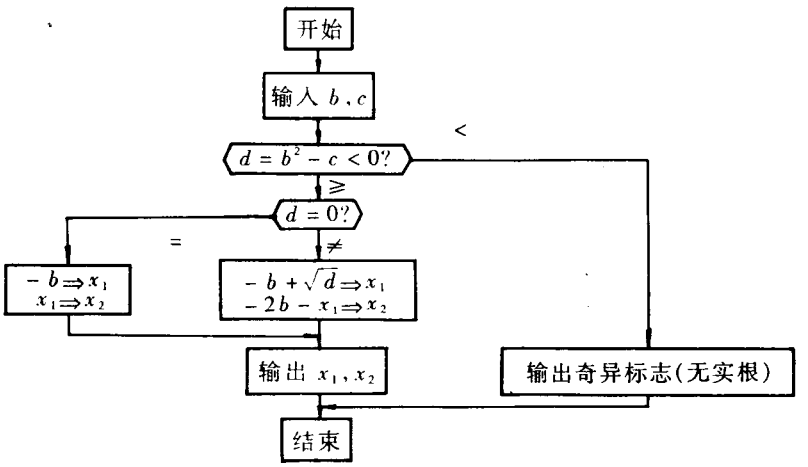
ELSE OUTPUT 无实根信息}

RETURN

(3)用框图

其中矩形框称叙述框，计算公式就填在里面，它也用来表示算法的开始或结束；圆边框称

为检验框，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，由出口处标注的条件决定．另外，用箭头“→”指明各框执行的顺序，而用“⇒”表示赋值语句．



第二节　误　差

一、误差的来源

在运用数学方法解决实际问题的过程中，每一步都可能带来误差．

1)模型误差　在建立数学模型时，往往要忽视很多次要因素，把模型“简单化”、“理想化”，这时模型就与真实背景有了差距，即带入了误差．

2)测量误差　数学模型中的已知参数，多数是通过测量得到．而测量过程受工具、方法、观察者的主观因素、不可预料的随机干扰等影响必然带入误差．

3)截断误差　数学模型常难于直接求解，往往要近似替代，简化为易于求解的问题，这种简化带入的误差称为方法误差或截断误差．

4)舍入误差　计算机只能处理有限数位的小数运算，初始参数或中间结果都必须进行四舍五入运算，这必然产生舍入误差．

在数值分析课程中不分析讨论模型误差；截断误差是数值分析课程的主要讨论对象，它往往是计算中误差的主要部分，在讲到各种算法时，通过数学方法可推导出截断误差限的公式(如(1-2)式)；舍入误差的产生往往带有很大的随机性，讨论比较困难，在问题本身呈病态或算法稳定性不好时，它可能成为计算中误差的主要部分；至于测量误差，我们把它作为初始的舍入误差看待．

误差分析是一门比较艰深的专门学科．在数值分析中主要讨论截断误差及舍入误差．但一个训练有素的计算工作者，当发现计算结果与实际不符时，应当能诊断出误差的来源，并采取相应的措施加以改进，直至建议对模型进行修改．

二、误差的基本概念

1)绝对误差与误差限

定义 1.1 设 x^* 是准确值，x 是它的一个近似值，称 $e = x - x^*$ 为近似值 x 的绝对误差，简称误差．

误差是有量纲的量，量纲同 x，它可正可负．

误差一般无法准确计算，只能根据测量或计算情况估计出它的绝对值的一个上限，这个上界称为近似值 x 的误差限，记为 ε，即 $|x - x^*| \leqslant \varepsilon$，其意义是：$x - \varepsilon \leqslant x^* \leqslant x + \varepsilon$

在工程中常记为：

$$x^* = x \pm \varepsilon$$

如 $l = 10.2 \pm 0.05\text{mm}$

$R = 1500 + 100\Omega$

2)相对误差与相对误差限：

误差不能完全刻画近似值的精度．如测量百米跑道产生 10cm 的误差与测量一个课桌长度产生 1cm 的误差，我们不能简单地认为后者更精确，还应考虑被测值的大小．下面给出定义：

定义 1.2 误差与精确值的比值

$$\frac{e}{x^*} = \frac{x - x^*}{x^*}$$

称为 x 的相对误差，记为 e_r.

相对误差是无量纲的量，常用百分比表示，它也可正可负．

相对误差也常不能准确计算，而是用相对误差限来估计．

相对误差限：

$$\varepsilon_r = \frac{\varepsilon}{|x^*|} \geqslant \frac{|x - x^*|}{|x^*|} = |e_r|$$

实际上由于 x^* 不知道，用上式无法确定 ε_r，常用 x 代 x^* 作分母，以后就用

$$\varepsilon_r = \frac{\varepsilon}{|x|}$$

表示相对误差限．

例 5 在刚才测量的例子中，若测得跑道长为 $100 \pm 0.1\text{m}$，课桌长为 $120 \pm 1\text{cm}$，则

$$\varepsilon_r^{(1)} = \frac{0.1}{100} = 0.1\%$$

$$\varepsilon_r^{(2)} = \frac{1}{120} = 0.83\%$$

显然后者比前者相对误差大．

三、有效数字

定义 1.3 如果近似值 x 的误差限 ε 是它某一数位的半个单位，我们就说 x 准确到该位，从这一位起直到前面第一个非零数字为止的所有数字称为 x 的有效数字．

如 $x = \pm 0.a_1 a_2 \cdots a_n \times 10^m$，其中 $a_1, a_2, \cdots, a_n$ 是 0～9 之中的整数，且 $a_1 \neq 0$，如 $e = |x - x^*| \leqslant \varepsilon = 0.5 \times 10^{m-l}$，$1 \leqslant l \leqslant n$，则称 x 有 l 位有效数字．

如 $\pi = 3.141\ 592\ 65\cdots$

则 3.14 和 3.141 6 分别有 3 位和 5 位有效数字．而 3.143 相对于 π 也只能有 3 位有效数

字．

在更多的情况，不知道准确值 x^*．如果认为计算结果各数位可靠，将它四舍五入到某一位，这时从这一位起到前面第一个非零数字共 l 位，由于四舍五入的原因，它与计算结果之差必不超过该位的半个单位．习惯上说将计算结果保留 l 位有效数字．如计算机上得到方程 $x^3 - x - 1 = 0$的一个正根为 1.324 72，保留 4 位有效数字的结果为 1.325，保留 5 位有效数字的结果为 1.324 7．

相对误差与有效数位的关系十分密切．定性地讲，相对误差越小，有效数位越多，反之亦正确．定量地讲，有如下两个定理．

定理 1.1 设近似值 $x = 0.a_1a_2\cdots a_n \times 10^m$，有 n 位有效数字，则其相对误差限

$$\varepsilon_r \leqslant \frac{1}{2a_1} \times 10^{-n+1}$$

此定理的证明不难，可作为习题完成．

定理 1.2 设近似值 $x = \pm 0.a_1a_2\cdots a_n\cdots \times 10^m$ 的相对误差限不大于$\frac{1}{2(a_1+1)} \times 10^{-n+1}$，则它至少有 n 位有效数字．

证 $|x| \leqslant (a_1+1) \times 10^{m-1}$

$$|x - x^*| = \frac{|x - x^*|}{|x|} \times |x| \leqslant \frac{1}{2(a_1+1)} \times 10^{-n+1} \times (a_1+1) \times 10^{m-1} = 0.5 \times 10^{m-n}$$

由定义 1.3 知 x 有 n 位有效数字．

例 6 计算 sin1.2，问要取几位有效数字才能保证相对误差限不大于 0.01%

解 $\sin 1.2 = 0.93\cdots$，故 $a_1 = 9, m = 0$

$$\varepsilon_r = \frac{1}{2a_1} \times 10^{-n+1} \leqslant 0.01\% = 10^{-4}$$

解关于 n 的不等式，得

$$10^{-n} \leqslant 18 \times 10^{-5} = 1.8 \times 10^{-4}$$

所以取 $n = 4$，即可满足要求．

对有效数字的观察比估计相对误差容易得多，故监视有效数字是否损失，常可发现相对误差的突然扩大．

例 7 计算$\frac{1}{759} - \frac{1}{760}$，视已知数为精确值，用 4 位浮点数计算

解 原式 $= 0.131\,8 \times 10^{-2} - 0.131\,6 \times 10^{-2} = 0.2 \times 10^{-5}$

结果只剩一位有效数字，有效数字大量损失，造成相对误差的扩大．若通分后再计算：

$$\frac{1}{759} - \frac{1}{760} = \frac{1}{759 \times 760} = \frac{1}{0.576\,8 \times 10^6} = 0.173\,4 \times 10^{-5}$$

就得到 4 位有效数字的结果．下文将会提到相近数字相减会扩大相对误差．

第三节 设计算法时应注意的原则

一、数值运算时误差的传播

当参与运算的数值带有误差时，结果也必然带有误差，问题是结果的误差与原始误差相比

是否扩大．

1)对函数 $f(x)$的计算:设 x 是 x^* 的近似值,则结果误差

$$e(f(x)) = f(x) - f(x^*)$$

用泰勒展式分析

$$f(x^*) = f(x) + f'(x)(x^* - x) + f''(\xi)\frac{(x^* - x)^2}{2}$$

$$e(f(x)) = f'(x)(x - x^*) - \frac{f''(\xi)}{2}(x - x^*)^2 \quad \xi \in (x, x^*)$$

$$| e(f(x)) | \leqslant | f'(x) | \varepsilon(x) + \left|\frac{f''(\xi)}{2}\right| \varepsilon^2(x)$$

忽略第二项高阶无穷小之后,可得函数 $f(x)$的误差限估计式

$$\varepsilon(f(x)) \approx | f'(x) | \varepsilon(x) \tag{1-9}$$

2)对多元函数 $f(x_1^*, x_2^*, \cdots, x_n^*) = A^*$,若 $x_1^*, x_2^*, \cdots, x_n^*$ 的近似值分别是 $x_1, x_2, \cdots, x_n$,则 $A = f(x_1, x_2, \cdots, x_n)$是结果的近似值．

$$e(A) = A - A^* = f(x_1, x_2, \cdots, x_n) - f(x_1^*, x_2^*, \cdots, x_n^*)$$

$$| A - A^* | = | f(x_1^*, x_2^*, \cdots, x_n^*) - f(x_1, x_2, \cdots, x_n) |$$

$$\leqslant \sum_{k=1}^{n} \left|\frac{\partial f(x_1, \cdots, x_n)}{\partial x_k}\right| | x_k - x_k^* | + O((\Delta x)^2)$$

其中 $\Delta x = \max\limits_{1 \leqslant k \leqslant n} | x_k - x_k^* |$,略去高阶项后

$$\varepsilon(A) \approx \sum_{k=1}^{n} \left|\frac{\partial f(x_1^*, \cdots, x_n^*)}{\partial x_k}\right| \varepsilon(x_k) \tag{1-10}$$

3)四则运算中误差的传播:按公式(1-10)易得近似数作四则运算后的误差限公式:

$$\varepsilon(x_1 \pm x_2) = \varepsilon(x_1) + \varepsilon(x_2) \tag{1-11}$$

$$\varepsilon(x_1 x_2) \approx | x_1 | \varepsilon(x_2) + | x_2 | \varepsilon(x_1) \tag{1-12}$$

$$\varepsilon\left(\frac{x_1}{x_2}\right) \approx \frac{| x_1 | \varepsilon(x_2) + | x_2 | \varepsilon(x_1)}{| x_2 |^2} \quad (x_2 \neq 0) \tag{1-13}$$

其中公式(1-11)取等号,是因为作为多元函数,加减运算是一次函数,泰勒展开式没有二次余项．

例 8 若电压 $V = 220 \pm 5$V,电阻 $R = 300 \pm 10\Omega$,求电流 I 并计算其误差限及相对误差限．

解 $I \approx \dfrac{220}{300} = 0.733\ 3$ A

$$\varepsilon(I) = \frac{| V | \varepsilon(R) + | R | \varepsilon(V)}{R^2}$$

$$= \frac{220 \times 10 + 300 \times 5}{90\ 000}$$

$$\approx 0.041\ 1$$

所以 $I = 0.733\ 3 \pm 0.041\ 1$A

$$\varepsilon_r(I) = \frac{0.041\ 1}{0.733\ 3} \approx 0.6\%$$

二、算法中应避免的问题

1)避免相近数相减．由公式(1-11)

$$\varepsilon(x_1 - x_2) = \varepsilon(x_1) + \varepsilon(x_2)$$

推出

$$\begin{aligned}\varepsilon_r(x_1 - x_2) &= \frac{\varepsilon(x_1 - x_2)}{|x_1 - x_2|} \\ &= \frac{|x_1|}{|x_1 - x_2|} \times \frac{\varepsilon(x_1)}{|x_1|} + \frac{|x_2|}{|x_1 - x_2|} \times \frac{\varepsilon(x_2)}{|x_2|} \\ &= \frac{|x_1|}{|x_1 - x_2|} \times \varepsilon_r(x_1) + \frac{|x_2|}{|x_1 - x_2|} \times \varepsilon_r(x_2)\end{aligned}$$

当 x_1, x_2 十分相近时，$|x_1 - x_2|$接近零，$\frac{|x_1|}{|x_1 - x_2|}$和$\frac{|x_2|}{|x_1 - x_2|}$将很大，所以 $\varepsilon_r(x_1 - x_2)$将比 $\varepsilon_r(x_1)$和 $\varepsilon_r(x_2)$大很多，即相对误差将显著扩大．

从直观上看，相近数相减会造成有效数位的减少，本章例 7 就是一个例子．有时，通过改变算法可以避免相近数相减．

例 9 解方程 $x^2 - 18x + 1 = 0$，假定用 4 位浮点计算．

解 用公式解法

$$x_1 = \frac{18 + \sqrt{18^2 - 4}}{2} = 9 + \sqrt{80} \approx 17.94$$

$$x_2 = 9 - \sqrt{80} \approx 9.000 - 8.944 = 0.056$$

可见第二个根只有两位有效数字，精度较差．若第二个根改为用韦达定理计算

$$x_2 = \frac{1}{x_1} \approx 0.055\ 74$$

可以得到较好的结果．

如$\sqrt{x+1} - \sqrt{x}$ $(x \gg 1)$可改为$\frac{1}{\sqrt{x+1} + \sqrt{x}}$

如 $1 - \cos x$ $(|x| \ll 1)$可改为 $2\sin^2\left(\frac{x}{2}\right)$等等，都可以得到比直接计算好的结果．

2)避免除法中除数的数量级远小于被除数．由公式(1-13)

$$\varepsilon\left(\frac{x_1}{x_2}\right) \approx \frac{|x_1|}{|x_2|^2}\varepsilon(x_2) + \frac{1}{|x_2|}\varepsilon(x_1)$$

若$|x_2| \ll |x_1|$，则$\frac{|x_1|}{|x_2|^2} \gg 1$，这时 $\varepsilon\left(\frac{x_1}{x_2}\right)$将比 $\varepsilon(x_2)$扩大很多．

3)防止小数被大数“吃掉”．在大量数据的累加运算中，由于加法必须进行对位，有可能出现小数被大数“吃掉”．

如用六位浮点数计算某市的工业总产值，原始数据是各企业的工业产值，当加法进行到一定程度，部分和超过 100 亿元(0.1×10^{11})，再加产值不足 10 万元的小企业产值，将再也加不进去．而这部分企业可能为数不少，合计产值相当大．这种情况应将小数先分别加成大数，然后相加，结果才比较正确．

这个例子说明，在计算机数系中，加法的交换律和结合律可能不成立，这是在大规模数据处理时应注意的问题．

4)注意运算步骤的简化，减少算术运算的次数以减少误差的积累效应；同时参与运算的数

字的精度应尽量保持一致，否则那些较高精度的量的精度没有太大意义．

最后指出，解决一个问题，可能有不同的算法．除了遵循以上原则外，算法的工作量，即算法的时间代价，算法所需的内存空间，以及算法的稳定性都是要考虑的问题．

习 题 一

1.用例 4 的算法计算$\sqrt{10}$，迭代 3 次，计算结果保留 4 位有效数字．

2.推导开平方运算的误差限公式，并说明什么情况下结果误差不大于自变量误差．

3.以下各数都是对准确值进行四舍五入得到的近似数，指出它们的有效数位、误差限和相对误差限．

$x_1 = 0.304\,0$，　$x_2 = 5.1 \times 10^9$，　$x_3 = 400$，　$x_4 = 0.003\,346$，　$x_5 = 0.875 \times 10^{-5}$

4.证明本章之定理 1.1.

5.为使球体积 V 的相对误差小于 0.1%，要求它的半径 R 的相对误差为多少？

6.若跑道长的测量有 0.1% 的误差，对 400m 成绩为 60s 的运动员的成绩将会带来多大的误差和相对误差．

7.为使$\sqrt{20}$的近似数相对误差小于 0.05%，试问该保留几位有效数字．

8.将多项式的“秦九韶算法”用框图描述出来．

9.将例 4 分别用框图和算法描述语言描述出来(先设定误差界 ε，当两次相邻的迭代值之差的绝对值小于 ε 时退出计算)．

10.下列各式应如何改进，使计算更准确：

(1) $y = \dfrac{1}{1+2x} - \dfrac{1-x}{1+x}$　$(|x| \ll 1)$

(2) $y = \sqrt{x + \dfrac{1}{x}} - \sqrt{x - \dfrac{1}{x}}$　$(x \gg 1)$

(3) $y = \dfrac{1 - \cos 2x}{x}$　$(|x| \ll 1)$

(4) $y = \sqrt{p^2 + q^2} - p$；　$p > 0, q > 0, p \gg q$