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之类简单的计算工具）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、加工并求得天气图像，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.

§1　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，这是否意味着计算机上的算法我们可以随意选择呢？

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，为此总共需要做 $n!(n-1)(n+1)$ 次乘法.当 n 充分大时，这个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，这项计算即使用百万次每秒的电子计算机去做，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（参看本书第五章与第六章），譬如用众所周知的消元法，一个 20 阶的方程组即使用计算器也能很快地解出来.这个简单的例子说明，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　据式(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.

同传统意义的近似计算方法不同,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和完整的描述.

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,设要用公式(3)求解二次方程(1),则需依判别式 $d = b^2 - c$ 的符号区分下列三种情况:

1° $d < 0$,无实根;

2° $d = 0$,有重根 $x_1 = x_2 = -b$;

3° $d > 0$,可用公式(3)求得两个互异实根 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

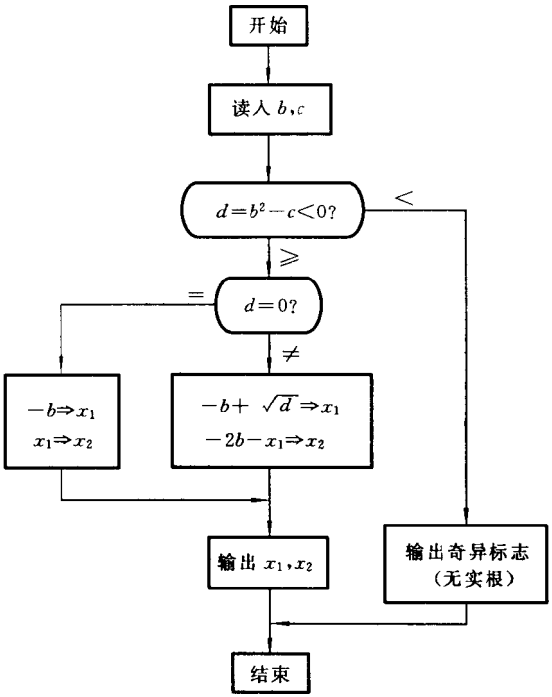

图 0-1

有两个出口，究竟选择哪个出口，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.

下面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 表示式(4)右端前 $k+1$ 项的部分和：

$$t_k=x^k$$

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

则

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

作为初值,令

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

利用初值(6)对 $k = 1,2,\cdots$,直到 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.加减操作的机器运行时间比乘除操作少得多,在统计计算量时,我们可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)按降幂的顺序重排为

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

从它的前两项提出 x^{n-1},则有

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

经过这个手续,如果算出括号内的值,则问题归结为计算一个 $n-1$ 次多项式(注意降了一次).再施行同样的手续,则进一步有

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

这样每做一步,所归结出的多项式就降低一次,最终可将所给计算公式(4)加工成如下嵌套形式

$$p(x) = (\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_1)x + a_0 \tag{7}$$

我们可利用式(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}:

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

作为初值,令

$$v_0 = a_n \tag{9}$$

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,它是我国宋代大数学家秦九韶最先

提出的①.

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，而将式(8)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_0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.

[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.

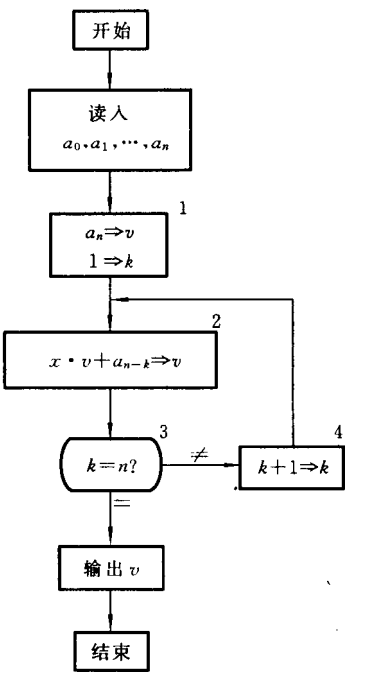

图 0-2

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.下述方程求根算法就是这类算法的一个范例.

设函数$f(x)$在$[a,b]$上连续,$f(a)f(b) < 0$，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.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：若确系同号，说明所求的根x^*在x_0的右侧,这

①外国文献称这一算法为霍纳(Horner)算法,其实霍纳的工作比秦九韶晚了五百多年.

时令 $a_1 = x_0, b_1 = b$;否则 x^* 必在 x_0 的左侧，这时令 $a_1 = a, b_1 = x_0$（图 0-3）.不管出现哪一种情形，新的有根区间 $[a_1, b_1]$ 的长度仅为 $[a, b]$ 的一半.

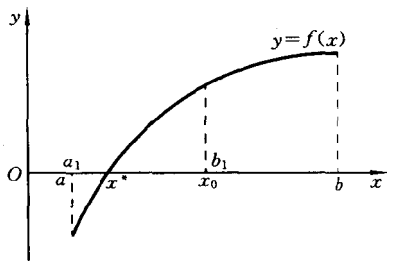

图 0-3

对于压缩了的有根区间 $[a_1, b_1]$ 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 的哪一侧，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

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

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.

第 k 次二分后，设取有根区间 $[a_k, b_k]$ 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 $x_0, x_1, x_2, \cdots$，该序列以根 x^* 为极限.

不过在实际计算时，我们不可能也没有必要完成这种无穷过程，因为计算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.

上述求根方法称为二分法，它是电子计算机上一种常用算法.我们给出其算法框图（图 0-4），图中 a, b 表示有根区间的

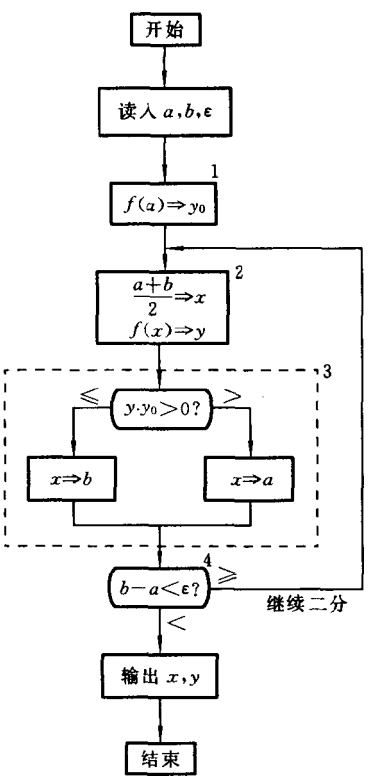

图 0-4

左、右端点；x 表示近似根.

图 0-4 中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] 判定二分后生成的有根区间 $[a,b]$.

[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	1.000 0	1.500 0	1.250 0
1	1.250 0		1.375 0
2		1.375 0	1.312 5
3	1.312 5		1.343 8
4		1.343 8	1.328 1
5		1.328 1	1.320 3
6	1.320 3		1.324 2

§2 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.

例 3 用中心差商公式求 $f(x)=\sqrt{x}$ 在 $x=2$ 的导数值：

$$f'(2)\approx\frac{\sqrt{2+h}-\sqrt{2-h}}{2h}\tag{10}$$

从理论上说，步长 h 愈小，计算结果愈准确. 上机计算的实际情况将会怎样呢？

解 我们知道，在计算机上数的表示受机器字长的限制，设取五位数字计算，若令 $h=0.1$，得

$$f'(2) \approx \frac{1.4491 - 1.3784}{0.2} = 0.35350$$

与导数的精确值 $f'(2) = 0.353553\cdots$ 比较，这项计算还是可取的．但是，如果缩小步长取 $h = 0.0001$，则得

$$f'(2) \approx \frac{1.4142 - 1.4142}{0.0002} = 0$$

算出的结果反而毫无价值．

这个例子告诫我们：两个相近的数相减，会造成有效数字的严重损失．实际计算中要尽量避免这种情况发生．

例 4 求解方程 $x^2 - (10^5 + 1)x + 10^5 = 0$.

解 仍取五位数字进行计算，并用“$\triangleq$”标记对阶舍入的计算过程．具体求根利用式 (3)，这里 $c = 10^5$，而

$$b = -\frac{1}{2} \times (10^5 + 1) \triangleq -\frac{1}{2} \times 10^5$$

$$\sqrt{b^2 - c} = \sqrt{\left[-\frac{1}{2}(10^5 + 1)\right]^2 - 10^5} \triangleq \frac{1}{2} \times 10^5$$

故有

$$x_1 = -b + \sqrt{b^2 - c} \triangleq 10^5$$

$$x_2 = -b - \sqrt{b^2 - c} \triangleq 0$$

原方程的精确解显然是 $x_1 = 10^5, x_2 = 1$，可见上面求出的结果严重失真

在计算机上，加减运算之前先要“对阶”，例 4 说明，对阶手续会造成大数“吃掉”小数的后果，因而实际计算时不宜用相差悬殊的两个数作加减运算．

例 5 考察方程组

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{11}{6} \\ \frac{13}{12} \\ \frac{47}{60} \end{bmatrix}$$

其解为

$$x_1 = x_2 = x_3 = 1$$

解 这类方程组的求解有很大困难．事实上，如果把系数舍入成三位小数：

$$\begin{bmatrix} 1.000 & 0.500 & 0.333 \\ 0.500 & 0.333 & 0.250 \\ 0.333 & 0.250 & 0.200 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.830 \\ 1.080 \\ 0.783 \end{bmatrix}$$

则其解变成

$$x_1 = 1.09, \quad x_2 = 0.488, \quad x_3 = 1.49$$

由此看出，尽管系数改变不大，所求出的解却有很大出入，这类问题称作是病态的（参看第六章 §4）.

2. 误差的来源

提起数值分析，往往给人以不严格、不准确以至于不够完善的感觉.其实,数值计算中的近似是正常的，计算误差是不可避免的.

我们知道，为要进行数值计算，首先必须将实际问题归结为数学问题，建立起合适的数学模型.在建立数学模型时，通常总要加上许多限制，总要忽略一些次要因素,这样建立起的‘理想化’的数学模型,虽然具有‘精确’而‘完美’的外表,其实只是客观现象的一种近似而粗糙的描述.这种数学描述上的近似必然会产生误差.

另外，在数值计算的过程中不可避免地还会产生其他各种各样的误差，本书主要考察以下两种：

(1)截断误差

我们知道，许多数学运算（诸如微分、积分及无穷级数求和等）是通过极限过程来定义的，然而计算机上只能完成有限次的算术运算和逻辑运算，因此需要将解题方案加工成算术运算与逻辑运算的有限序列.这种加工常常表现为无穷过程的截断，由此产生的误差通常称作截断误差.

譬如,指数函数 e^x 可展开为幂级数形式：

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

但用计算机求值时，我们不能得出右端无穷多项的和，而只能截取有限项计算

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

这样计算部分和 $S_n(x)$ 作为 e^x 的值必然会有误差,据泰勒(Taylor)余项定理,其截断误差为

$$e^x - S_n(x) = \frac{x^{n+1}}{(n+1)!}e^{\theta x}, \qquad 0 < \theta < 1$$

(2) 舍入误差

计算过程中所用的数据可能位数很多，甚至是无穷小数，然而受机器字长的限制，用机器代码表示的数据必须舍入成一定的位数，这就会引起舍入误差.每一步的舍入误差是微不足道的，但经过计算过程的传播和积累，舍入误差甚至可能会“淹没”所要的真解.

3. 误差限和有效数字

误差虽然不可避免，但人们总是希望计算结果能足够准确，这就需要估计误

差. 设以 x 代表数 x^* 的近似值,误差 $x-x^*$ 的具体数值通常是无法确定的，只能根据测量工具或计算过程设法估算出它的取值范围，即误差绝对值的一个上界：

$$|x-x^*|\leqslant\varepsilon$$

这种上界 ε 称作近似值 x 的绝对误差限，简称误差限，或称精度. 精度为 ε 的近似值 x,可以认为和真值 x^* 关于允许误差 ε 是“重合”的,或者说,结果 x 关于精度 ε 是“准确”的.

写出一个近似值后，我们当然希望指明它的准确程度，为此需要引进有效数字的概念.

我们知道，要将一个位数很多的数表示成一定的位数，通常用四舍五入的办法,如

$$\pi=3.14159265\cdots$$

可表为 3.14,3.1416 等，这种表示方法的特点是，近似数的误差限为其最末一位的半个单位.

如果近似值 x 的误差限是它的某一位的半个单位，我们就说它“准确”到这一位，并且从这一位起直到前面第一个非零数字为止的所有数字均称有效数字. 具体地说，对于 x^* 的近似值（规格化形式）

$$x=\pm\,0.a_1a_2\cdots a_n\times 10^m$$

(其中，$a_1,a_2,\cdots,a_n$ 是 0 到 9 之间的自然数，$a_1\neq 0$),如果误差

$$|x-x^*|\leqslant\frac{1}{2}\times 10^{m-l},\qquad 1\leqslant l\leqslant n$$

则称近似值 x 有 l 位有效数字，或称“准确”到第 l 位.

按照这种说法,π 的近似值 3.14 和 3.1416 分别有 3 位和 5 位有效数字.

4. 相对误差与有效数字的联系

近似值 x 的绝对误差 $|x-x^*|$ 还不足以刻画它的精度，譬如，测量 1000 m 的长度时发生了 1cm 的误差，同测量 1m 的长度时发生了 1cm 的误差，两者的含义是大有区别的，可见要刻画近似值的精度，除了要看绝对误差的大小之外，还应当考察这个值本身的大小，这就需要进一步引进相对误差的概念.

仍以 x 代表 x^* 的近似值,若

$$\frac{|x-x^*|}{|x|}\leqslant\varepsilon$$

则称 ε 为近似数 x 的相对误差限.

设数 x,y 分别是数 x^*,y^* 的近似值,考察 $x-y$ 的相对误差

$$\frac{|(x-y)-(x^*-y^*)|}{|x-y|}\leqslant\frac{|x-x^*|+|y-y^*|}{|x-y|}$$

$$= \left|\frac{x-x^*}{x}\right| \cdot \left|\frac{x}{x-y}\right| + \left|\frac{y-y^*}{y}\right| \cdot \left|\frac{y}{x-y}\right|$$

当 x 与 y 相近时，$\left|\frac{x}{x-y}\right|$ 和 $\left|\frac{y}{x-y}\right|$ 都很大，这时 $x-y$ 的相对误差可能比 x 与 y 的相对误差大得多.

由此可见，两个值相近的近似数相减，可能会造成有效数字的严重损失，在实际计算时常加工计算公式，以避免这种情况发生.

再考察例 3，当 h 很小时，式 (10) 的分子为两个相近的数 $\sqrt{2+h}$ 和 $\sqrt{2-h}$ 相减，为避免有效数字的严重损失，将式 (10) 的分子和分母同乘以 $\sqrt{2+h}+\sqrt{2-h}$，而将它加工成

$$f'(2) \approx \frac{1}{\sqrt{2+h}+\sqrt{2-h}}$$

按这一公式求导，仍取 $h=0.0001$ 得 $f'(2) \approx 0.35356$，这个结果有 4 位有效数字.

习 题 0

1. 用二分法求方程 $x^3-x-1=0$ 在 $[1,2]$ 内的近似根，要求误差不超过 10^{-3}.

2. 证明方程 $f(x)=e^x+10x-2=0$ 在 $[0,1]$ 内有唯一实根；用二分法求这一实根，要求误差不超过 $\frac{1}{2}\times 10^{-2}$.

3. 利用秦九韶算法求多项式

$$p(x)=x^5-3x^4+4x^2-x+1$$

在 $x=3$ 时的值.

4. 试证：对任给初值 x_0，求开方值 $\sqrt{a}\,(a>0)$ 的牛顿迭代公式

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

恒成立下列关系式：

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

2) $x_k \geqslant \sqrt{a}, \qquad k=1,2,\cdots$

5. 试用题 4 的迭代公式求 $\sqrt{8}$，要求准确到 10^{-3}.

6. 设用题 4 的迭代公式求 $\sqrt{8}$，证明，若迭代值 x_k 有 n 位有效数字，则 x_{k+1} 必有 $2n$ 位有效数字.

7. 测量一木条长为 542cm,若其绝对误差不超过 0.5cm,问测量的相对误差是多少？

8. 已知 $e = 2.71828\cdots$,试问其近似值 $x_1 = 2.7, x_2 = 2.71, x_3 = 2.718$ 各有几位有效数字？并给出它们的相对误差限.

9. 设 $x_1 = -2.72$, $x_2 = 2.71828$, $x_3 = 0.0718$ 均为经过四舍五入得出的近似值，试指明它们的绝对误差与相对误差.

10. 已知近似值 $x_1 = 1.42$, $x_2 = -0.0184$, $x_3 = 184 \times 10^{-4}$ 的绝对误差限均为 0.5×10^{-2}，问它们各有几位有效数字？

11. 用 $\frac{22}{7}$ 和 $\frac{355}{113}$ 作为 $\pi = 3.14159265\cdots$ 的近似值，问它们各有几位有效数字？

12. 为尽量避免有效数字的严重损失，当 $|x| << 1$ 时应如何加工下列计算公式：

1) $\frac{1}{1+2x} - \frac{1-x}{1+x}$　　2) $1 - \cos x$

3) $e^x - 1$

13. 设 $|x| >> 1$，加工下列计算公式以尽量避免有效数字的严重损失：

1) $\sqrt{x + \frac{1}{x}} - \sqrt{x - \frac{1}{x}}$　　2) $\int_x^{x+1} \frac{1}{1+t^2} dt$

3) $\ln\left(x - \sqrt{x^2 - 1}\right)$

第一章　插值方法

实际问题中碰到的函数 $f(x)$ 是各种各样的，有的表达式很复杂，有的甚至给不出数学式子，只提供了一些离散数据，譬如某些点上的函数值和导数值．由于问题的复杂性，直接研究函数 $f(x)$ 可能很困难．面对这种情况，一个很自然的想法是，设法将所考察的函数 $f(x)$“简单化”，就是说，构造某个简单函数 $p(x)$ 作为 $f(x)$ 的近似函数，然后通过处理 $p(x)$ 获得关于 $f(x)$ 的结果．如果要求近似函数 $p(x)$ 取给定的离散数据，则称之为 $f(x)$ 的插值函数．

选用不同类型的插值函数，逼近的效果不同．由于代数多项式的结构简单，数值计算和理论分析都很方便，实用上常取代数多项式作为插值函数，这就是所谓的代数插值．

本章先讨论代数插值，然后在此基础上进一步研究所谓的样条插值．

§1　问题的提法

1. 泰勒插值

我们所熟悉的泰勒展开方法其实就是一种插值方法．温故而知新，首先回顾一下这种方法是有益的．

已知泰勒多项式

$$p_n(x)=f(x_0)+f'(x_0)(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n \tag{1}$$

与 $f(x)$ 在点 x_0 具有相同的导数值：

$$p_n^{(k)}(x_0)=f^{(k)}(x_0),\qquad k=0,1,\cdots,n$$

因此 $p_n(x)$ 在点 x_0 邻近会很好地逼近 $f(x)$．下述泰勒余项定理则是众所周知的．

定理 1　假设 $f(x)$ 在含有点 x_0 的区间 $[a,b]$ 内有直到 $n+1$ 阶导数，则当 $x\in[a,b]$ 时，对于由式(1)给出的 $p_n(x)$，成立

$$f(x)-p_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}$$

式中 ξ 界于 x_0 与 x 之间，因而 $\xi\in[a,b]$.

所谓泰勒插值是指下述插值问题：

问题 1　求作 n 次多项式 $p_n(x)$①，使满足条件：

$$p_n^{(k)}(x_0) = y_0^{(k)}, \qquad k = 0, 1, \cdots, n$$

这里 $y_0^{(k)}(k = 0,1,\cdots,n)$ 为一组已给数据.

对于给定的函数 $f(x)$，若导数值 $f^{(k)}(x_0) = y_0^{(k)}, k = 0,1,\cdots,n$ 已给，则上述泰勒插值问题的解就是泰勒多项式(1).

例 1　求作 $f(x) = \sqrt{x}$ 在 $x_0 = 100$ 的一次和二次泰勒多项式，利用它们计算 115 的近似值并估计误差.

解　由于 $x_0 = 100$，而

$$f(x) = \sqrt{x}, \quad f'(x) = \frac{1}{2\sqrt{x}}, \quad f''(x) = -\frac{1}{4x\sqrt{x}},$$

$$f(x_0) = 10, \quad f'(x_0) = \frac{1}{20}, \quad f''(x_0) = -\frac{1}{4000}$$

$f(x)$ 在 x_0 的一次泰勒多项式是

$$p_1(x) = f(x_0) + f'(x_0)(x - x_0) = 5 + 0.05x$$

用 $p_1(x)$ 作为 $f(x)$ 的近似表达式，容易求出当 $\bar{x} = 115$ 时

$$\sqrt{115} = f(\bar{x}) \approx p_1(\bar{x}) = 10.75$$

据定理 1 可估计出误差

$$0 > f(\bar{x}) - p_1(\bar{x}) = \frac{f''(\xi)}{2}(\bar{x} - x_0)^2$$

$$> \frac{f''(x_0)}{2}(\bar{x} - x_0)^2 = -0.028125$$

$\sqrt{115}$ 的精确值为 $10.723805\cdots$，与精确值相比较，近似值 10.75 的误差大约等于 -0.026，因而它有 3 位有效数字.

修正 $p_1(x)$ 可进一步得出二次泰勒多项式

$$p_2(x) = p_1(x) + \frac{f''(x_0)}{2}(x - x_0)^2$$

据此可得到新的近似值

$$\sqrt{115} = f(\bar{x}) \approx p_2(\bar{x}) = 10.75 - 0.028125 = 10.721875$$

这个结果有 4 位有效数字.

2. 拉格朗日插值

上述泰勒插值要求提供 $f(x)$ 在点 x_0 处的各阶导数值，这项要求很苛刻，函数

①在这一章，所谓“n 次多项式”常常泛指次数不超过 n 的多项式.

$f(x)$ 的表达式必须相当简单才行. 如果仅仅给出一系列节点上的函数值 $f(x_i) = y_i, i = 0,1,\cdots,n$ ，则插值问题可表述如下：

问题 2 求作 n 次多项式 $p_n(x)$，使满足条件

$$p_n(x_i) = y_i, \qquad i = 0,1,\cdots,n \tag{2}$$

这就是所谓拉格朗日（Lagrange）插值. 点 x_i（它们互不相同）称为插值节点. 用几何语言来表述这类插值，就是，通过曲线 $y = f(x)$ 上给定的 $n+1$ 个点 $(x_i, y_i), i = 0,1,\cdots,n$，求作一条 n 次代数曲线 $y = p_n(x)$ 作为 $y = f(x)$ 的近似.

首先讨论插值问题的可解性. 设所求的插值多项式为

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

则插值条件（2）具体写出来是关于系数 $a_0, a_1, \cdots, a_n$ 的线性方程组

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n = y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n = y_1 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n = y_n \end{cases}$$

其系数行列式是所谓范德蒙（Vandermonde）行列式

$$V = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix}$$

可以证明，如果节点 $x_0, x_1, \cdots, x_n$ 互不相同，则行列式 V 值必异于 0，据此可以断定：

定理 2 问题 2 的解 $p_n(x)$ 存在并且是唯一的.

以上关于插值问题可解性的论证是构造性的，通过求解线性方程组即可确定插值函数 $p_n(x)$，问题在于这种算法的计算量大，不便于实际应用.

插值多项式的构造能否回避求解线性方程组呢？回答是肯定的.

下面提供插值多项式的显示表达式.

§2 拉格朗日插值公式

1. 线性插值

首先考察线性插值的简单情形.

问题 3 求作一次式 $p_1(x)$①，使满足条件

①再强调一遍，严格地讲，这里 $p_1(x)$ 是不高于一次的多项式，譬如，若 $y_1 = y_0$，则插值多项式 $p_1(x) = y_0$ 是个零次式.

$$p_1(x_0)=y_0,\quad p_1(x_1)=y_1$$

从几何图形上看，$y=p_1(x)$ 表示通过两点 $(x_0,y_0),(x_1,y_1)$ 的直线，因此一次插值亦称线性插值.

上述简单的线性插值是我们所熟悉的，它的解 $p_1(x)$ 可表为下列点斜式

$$p_1(x)=y_0+\frac{y_1-y_0}{x_1-x_0}(x-x_0) \tag{3}$$

例 2 已知 $\sqrt{100}=10$，$\sqrt{121}=11$，求 $y=\sqrt{115}$.

解 这里 $x_0=100, y_0=10$，$x_1=121$，$y_1=11$，令 $x=115$ 代入式 (3) 求得 $y=10.71428$，这个结果有 3 位有效数字（试与例 1 的结果相比较）.

我们知道，线性插值公式(3) 亦可表为下列对称式

$$p_1(x)=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1 \tag{4}$$

若令

$$l_0(x)=\frac{x-x_1}{x_0-x_1},\quad l_1(x)=\frac{x-x_0}{x_1-x_0}$$

则有

$$p_1(x)=y_0\,l_0(x)+y_1\,l_1(x) \tag{5}$$

注意，这里的 $l_0(x)$ 和 $l_1(x)$ 分别可以看作是满足条件

$$l_0(x_0)=1,\quad l_0(x_1)=0$$

$$l_1(x_1)=1,\quad l_1(x_0)=0$$

的插值多项式. 这两个特殊的插值多项式称作问题 3 的插值基函数(参看图 1-1、1-2).

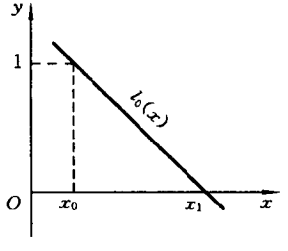

图 1-1

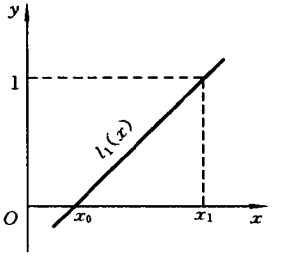

图 1-2

式(5)表明，插值问题 3 的解 $p_1(x)$ 可以通过插值基函数 $l_0(x)$ 和 $l_1(x)$ 组合得出，且组合系数恰为所给数据 y_0,y_1.

2. 抛物插值

线性插值仅仅利用了两个节点上的信息，精度自然很低，为了提高精度，我们进一步考察下述二次插值：

问题 4　求作二次式 $p_2(x)$，使满足条件

$$p_2(x_0)=y_0,\quad p_2(x_1)=y_1,\quad p_2(x_2)=y_2 \tag{6}$$

二次插值的几何解释是，用通过三点 $(x_0,y_0),(x_1,y_1),(x_2,y_2)$ 的抛物线 $y=p_2(x)$ 来近似所考察的曲线 $y=f(x)$，因此这类插值亦称抛物插值.

为了得出插值多项式 $p_2(x)$，我们先解决一个特殊的二次插值问题：求作二次式 $l_0(x)$，使满足条件

$$l_0(x_0)=1,\quad l_0(x_1)=l_0(x_2)=0 \tag{7}$$

这个问题是容易求解的，事实上，由式（7）的后两个条件知，x_1,x_2 是 $l_0(x)$ 的两个零点，因而

$$l_0(x)=c(x-x_1)(x-x_2)$$

再利用式(7)剩下的一个条件 $l_0(x_0)=1$ 确定系数 c，结果得出

$$l_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}$$

类似地可以构造出满足条件

$$l_1(x_1)=1,\quad l_1(x_0)=l_1(x_2)=0$$

$$l_2(x_2)=1,\quad l_2(x_0)=l_2(x_1)=0$$

的插值多项式 $l_1(x)$ 与 $l_2(x)$，其表达式分别为

$$l_1(x)=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}$$

$$l_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

这样构造出的 $l_0(x),l_1(x)$ 和 $l_2(x)$ 称作问题 4 的插值基函数.

设取已知数据 y_0,y_1,y_2 作为组合系数，将插值基函数 $l_0(x),l_1(x),l_2(x)$ 组合得

$$p_2(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2 \tag{8}$$

容易看出，这样构造出的 $p_2(x)$ 满足条件(6)，因而它就是问题 4 的解.

例 3　利用 100,121 和 144 的开方值求 $\sqrt{115}$.

解　$x_0=100,y_0=10,x_1=121,y_1=11,x_2=144,y_2=12$，令 $x=115$ 代入式(8)求得 $\sqrt{115}$ 近似值为 10.7228. 同精确值比较，这里得到