# 应用数值分析

王韶华
黄瑞霖 编著

西南交通大学出版社

## 内 容 提 要

本书内容包括插值、数值微分与数值积分、线性方程组的解法、常微分方程的初值问题和边值问题的解法、本征值问题以及解各类偏微分方程的有限差分法。本书对数值分析中的舍入误差、截断误差、计算结果的精度、病态矩阵、差分格式的相容性、稳定性、收敛性等都进行了一些讨论。每章末都附有用 FORTRAN 语言编写的计算机程序，全书共 39 个。本书着眼于实用，书中附有大量例题及计算机程序以阐明解法；叙述力求清晰、深入浅出以便于自学；对于数值分析的理论内容，只是在它们能够加深了解或能激发读者对所讲方法进行研究的兴趣时，才酌情列入。

本书可作为理工科大学各专业的本科生或研究生的教材或参考书；对需要大量使用计算机的科技工作者，也具有实用价值。

**应 用 数 值 分 析**
**YINGYONG SHUZHI FENXI**
王韶华　黄瑞霖　编著
*
西南交通大学出版社出版
（四川　峨眉）
西南交通大学出版社印刷厂印刷
*
开本：787×1092　1/16　印张：22.375
1987 年 3 月第一版　1987 年 3 月第一次印刷
字数：573 千字　印数：1—1000 册
ISBN 7—81022—000—4
O　001
统一书号：13478 · 03　定价：4.00元

# 前　言

随着电子计算机的出现和迅速发展，熟练地运用计算机进行科学计算已成为科技工作者不可缺少的技能，而数值计算方法是进行科学计算的必备工具。为读者提供进行数值计算的基本知识和基本技能，正是编写本书的目的。

本书是在王韶华编写的《应用数值分析》讲义（西南交大印刷厂，1983年）的基础上经过补充、修改重新编写的。它保留了原讲义着眼于实用、深入浅出、便于自学、便于使用计算机等特点。

本书按照数值分析中的基本手段的逻辑分类编写。在内容上，由于目前用计算机解各类常微分方程、偏微分方程问题已具有广泛的实用意义，所以用较多的篇幅介绍了各类微分方程，特别是偏微分方程的有限差分解法。

书中所附的计算机程序，绝大部分都用 WANG 2200 VS/80 计算机进行了调试。如果读者使用其他型号的计算机，除需对输入、输出的设备号按所用机型的规定加以变更外，还可能要根据该机编译系统的一些特殊要求稍加改动。

本书前三章由黄瑞霖编写，其他各章由王韶华编写。限于编者水平，错误和不妥之处在所难免，恳请读者批评指正。

王韶华　
黄瑞霖　1986年10月

# 目 录

# 第1章
# 引　论

## 1.1　数值分析

数值分析是一门实用的学科。它研究寻求数学问题近似解的方法和为此而必须遵循的过程。

数值分析同含有数字的问题有关，虽然目前它已使用了最先进的计算工具——数字计算机，但本质上与它有关的仍只是加、减、乘、除四则算术运算。数值分析是数学最古老的分支之一，起源于史前时期。古代的数学家，曾用这种方法计算过圆周率 $\pi$ 和 $\sqrt{2}$ 的近似值；而对数表和三角函数表，更是数值分析工作中早期的杰作。

微积分学的出现，大大促进了自然科学与工程技术的发展。对于一个实际问题，可以利用已被认识到的某些基本规律，建立起它的数学模型，即某种数学方程或方程组（如代数方程、常微分方程、偏微分方程、积分方程、微分积分方程等等），然后结合特定的条件（如该问题的边界条件、起始条件等）求出方程的解答；再通过实践来检验解的正确性。在有些情况下，通过此种工作，甚至还可以预见到新的科学规律。

随着科学技术的发展，人们对自然界的认识日益深入，于是提出的各种数学模型，也日益复杂。例如，方程所含的变量数目大大增加，有时还会出现随机量；一些偏微分方程的边界条件十分复杂等等。这时，已难以按数学分析寻求理论解，即用数学表达式给出的解析解。过去，每当遇到这类问题，人们往往不得不对模型进行简化或采用近似方法（如微扰法、变分法等）来推求近似解。事实上，往往虽有方程，而得不到解答。

数字计算机的诞生和惊人的发展速度，越来越显示出它是大型数值计算的强有力的工具。现代数字计算机具有巨大的信息贮存量和极高的运算速度，因而能按照事先编制的计算程序自动地、迅速地完成极其繁杂的计算过程并提供计算结果。从根据数学模型提出求解的数值计算方法，直到编制程序并由计算机算出数值解的整个过程，构成了当今数值分析工作的主要内容。

科学技术的发展向数值分析课程提出了更高的要求。受到这种要求的推动，加之以计算机功能的进一步完善，数值分析的内容，包括理论研究和算法创建，也不断地得到充实。限于篇幅，本书将只介绍一些成熟的、有实用意义的计算方法，和某些有关的理论。后者主要是误差分析、收敛性和稳定性的分析。此外，也适当地注意了编制程序的某些技巧。

## 1.2　连续变量的离散表示

离散变量与离散化，是数值计算中最基本的概念。在数值计算时，由于数字计算机只能执行算术的和逻辑的操作，因此，任何涉及连续变量的计算问题，都需要经过离散化以后才能运算。例如，一个一维坐标中的连续函数 $f(x)$，就可以离散化为在有限个离散点上的有限个函数值的集合。考虑独立的连续变量 $x$，它处于下述域中：

$$X=[X_1,\ X_2]，\text{即}\quad X_1\leqslant x\leqslant X_2。$$

把域分为$J$个基元$\Delta x_i$，$i=1, 2, 3, \cdots, J$（可以等分，也可以不等分），则域$X$就是$J$个基元$\Delta x_i$的集合。$J+1$个离散点称为节点（结点），如图1.1所示。对于给定的$J$，矢量$\{x_i\}$可以由连续变量$x$在各个节点上的值组成。可以看出

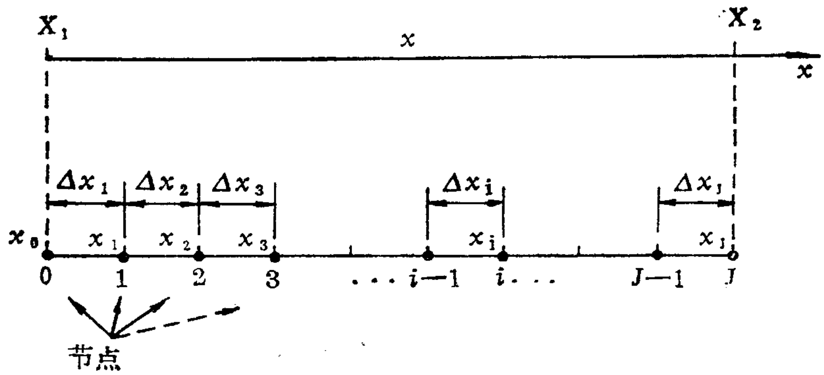


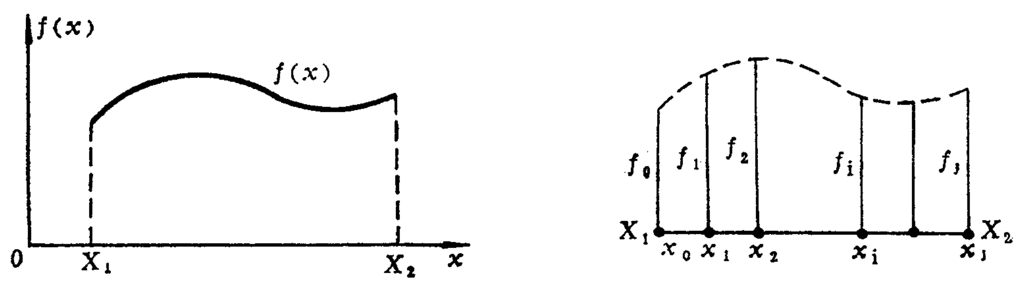

图 1.1

$$x_i = X_1 + \sum_{k=1}^{i} \Delta x_k = x_0 + \sum_{k=1}^{i} \Delta x_k, \tag{1.1}$$

这样，连续函数$f(x)$就可离散化为矢量$\{f_i\}$，且

$$f_i = f(x_i) \quad (i=0, 1, 2, \cdots, J)。 \tag{1.2}$$

图1.2($a$)的曲线表达了连续函数$f(x)$；($b$)给出了$f(x)$被离散化后的$J+1$个函数值。这里，$x_i$和与之对应的$f_i$，都不再是连续变量。式(1.1)中的$\Delta x_i$，称为步长。




($a$) ($b$)

图 1.2

对于被离散化的函数，我们不再能用极限概念来定义其微分和积分；而是用有限差分代替微分，用有限差商代替微商，用离散点上函数值的累加代替积分，这就是数值微分及数值积分的方法。当然，这种计算方法是近似的。

与变量的离散化相反，也可以通过插值方法（或最小二乘法），把离散变量的诸函数值“联成”连续变量的函数（函数的逼近）。实际上，在物理实验或其他科学实验中测量出来的数据，就是以离散变量的形式给出的。我们可以通过上述方法求出已知离散点附近另一点上相应的近似函数值。这类问题的详细讨论是下一章的主要内容。

## 1.3 误差 有效数字

前面提到，数值计算是一种近似方法。但在解决实际问题时，近似解的价值绝不低于准确解。特别是利用高速、大贮存量的计算机，可以将步长取得较小，从而大大提高了近似解的准确度。为了使近似解达到一定的准确度，就必须研究“误差”。有关误差的概念定义如下：

$$\text{误差}=\text{精确值（真值）}-\text{近似值},$$

$$\text{相对误差}=\left|\frac{\text{误差}}{\text{近似值}}\right|。$$

误差的来源有以下几方面：

模型误差 物理学或其他科学中的许多数学模型，都是在大量观察、实验的基础上，抓住主要矛盾，抛开次要因素，经过抽象、简化而得到的，因而，它们总是近似的。这种误差难以用数量表示。通常假定数学模型是合理的，因此，这种误差在“数值分析”中不予讨论。

原始数据误差 在数学模型中往往包含一些由观测得到的物理量，如温度、比重、电阻率、电压等等，这些参量本身就带有一定的误差（称为观测误差）；在解微分方程时，初值条件或边值条件不够精确，或必须将无理数表示成有尽小数，都给计算结果带来误差。以上原因带来的误差统称为原始数据误差。在“数值分析”中，不研究观测误差。

另外，在数值计算工作中，不可避免地还存在着以下两类误差。

舍入误差 尽管用计算机运算时，数的位数可以很多，但总因受到机器字长的限制而不得不取有限位数，对超过一定位数的数字进行舍入，这就产生了舍入误差。运算次数较少时，其舍入误差是微不足道的，但若用计算机进行千百万次计算，舍入误差的累积将会大大影响结果的精度。

截断误差（方法误差） 在数值计算中，常需利用极限或无穷过程来求值，但在具体计算时，又只能进行有限次运算，因而只好用有限的步骤来求近似值。比如，当物理问题的数学描述为微分方程时，要用有限差分格式进行计算；各种函数用无穷级数求值时，把截断的级数做为近似式进行计算。由此产生的精确解与近似解之间的误差，称为截断误差（方法误差）。例如，在计算 $e^h(h>0)$ 的值时，由马克劳林（Maclaurin）展开式有

$$e^h=1+h+\frac{h^2}{2!}+\frac{h^3}{3!}+\cdots+\frac{h^n}{n!}+\cdots$$

若取前面 $(n+1)$ 项进行计算，$e^h$ 的近似式为

$$e^h\approx 1+h+\frac{h^2}{2!}+\frac{h^3}{3!}+\cdots+\frac{h^n}{n!}, \tag{1.3}$$

则截断误差（即余项）为

$$R_n=\frac{h^{n+1}}{(n+1)!}+\frac{h^{n+2}}{(n+2)!}+\cdots=\frac{e^{\theta h}}{(n+1)!}\cdot h^{n+1}\quad(0<\theta<1)。 \tag{1.4}$$

我们总可以找到一个正数 $M$，使 $R_n<M\cdot h^{n+1}$。例如，可取 $M=e^h/(n+1)!$，这时截断误差可写成 $R_n=O(h^{n+1})$。一般说来，如果某个近似式的误差小于 $h^k$ 乘以一个任意正数的积，则称此误差是 $h^k$ 阶，简称 $k$ 阶，记作 $O(h^k)$。这里指的是无穷小量的阶（order），可以用它来衡量误差趋于零的快慢程度。符号 $f(x)=O(g(x))$ 表示：当 $x\to 0$ 时，$f(x)$ 的

量阶不低于 $g(x)$ 的量阶。在上例中，$R_n=O(h^{n+1})$ 表示：当 $h\to 0$ 时，$R_n$ 是比 $h^n$ 高阶的无穷小量。

以上说明，在数值计算过程中，会出现各种各样的误差。有时，误差会严重"泛滥"，竟"湮没"了欲求的结果。因此，进行任何计算，事先都必须考虑误差的来源和大小，以保证达到预期的精度。

为了表示近似值的精度，还要引进有效数字的概念。

设 $I$ 为精确值，$I^*$ 为其近似值，可以将 $I$ 写成如下形式：

$$I=0.a_1a_2\cdots a_na_{n+1}\cdots\times 10^k \quad (k\text{ 可以是 }0,\ \pm 1,\ \pm 2,\ \cdots),$$

其中 $a_i(i=1,\ 2,\ 3,\ \cdots)$ 取自十进制数 0 至 9 中任意一个，且 $a_1\neq 0$。近似值 $I^*$ 可写为

$$I^*=0.a_1a_2\cdots a_na'_{n+1}\cdots\times 10^k。$$

若误差的绝对值

$$|I-I^*|\leqslant 0.\underbrace{00\,0\cdots 0\,0}_{n\text{ 个 }0}5\times 10^k=0.5\times 10^{k-n},$$

则称 $I^*$ 为具有 $n$ 位有效数字的有效数。$a_1$、$a_2$、…、$a_n$ 称为有效数字，$n$ 称为 $I^*$ 的有效位数。有效位数用来描述近似值 $I^*$ 的精度，$n$ 越大，绝对误差越小。

例如，对于 $I=\pi=3.1415926\cdots=0.31415926\cdots\times 10$，若取近似值 $I_1^*=3.14$，绝对误差为 $0.0001526\cdots\times 10$，可见 $I_1^*$ 有 3 位有效数字；同理，若取 $I_2^*=3.1416$，有 5 位有效数字；若取 $I^*=3.1415$，则只有 4 位有效数字了。

又如，$I=9.80665$ 米 $=0.980665\times 10$ 米，取 $I^*=9.81$ 米，有 3 位有效数字；若写成 $I=0.00980665$ 千米，取 $I^*=0.00981$ 千米，仍有 3 位有效数字。可见，近似值的有效位数只与有效数字的位数有关，与小数点后有多少位数无关。

考虑到有效位数，数字 8.0000 与 8，二者意义不同。前者有 5 位有效数字，而后者只有一位有效数字。

## 1.4 缩小误差危害的若干原则

在数值运算中，误差分析是至关重要而又非常复杂的问题。一个实际问题往往要进行成千上万次计算，每一步计算都有误差，因而每一步都分析误差是不可能的。为了保证数值运算结果的可靠性，一方面要分门别类研究误差的规律，如数值微分与数值积分、矩阵计算、差分方程……等；另一方面，由于计算方法不同，误差的累积有时会增加，有时可以互相抵消而减少。因此，我们针对普遍性的问题，提出缩小误差危害的若干注意事项。

(1) 要使用稳定的计算过程

我们把运算过程中舍入误差不增长的计算过程称为稳定的，否则是不稳定的。对于不稳定的计算过程，如计算次数太多，就有可能得出错误结果，因此要尽量避免使用不稳定的公式，如要使用也必须特别小心。因此在数值分析中，对很多计算方法都要分析计算过程的稳定性而不具体估计舍入误差的大小。

例如，计算 $I_n=e^{-1}\int_0^1 x^n e^x dx,\quad n=0,\ 1,\ 2,\ \cdots$，并分析误差大小。

$$I_0=e^{-1}\int_0^1 e^x dx=1-e^{-1},$$

取 4 位有效数字，$I_0\approx 0.6321=I_0^*$，误差 $\varepsilon_0<10^{-4}$。

由分部积分可得计算 $I_n$ 的递推公式

$$I_n = 1 - nI_{n-1}, \qquad n = 1, 2, 3, \cdots, \tag{1.5}$$

得到 $I_n^* = 1 - nI_{n-1}^*$。

将 $n = 1, 2, 3, \cdots$ 代入上式分别计算得到：

| | |
|---|---|
| $I_0^* = 0.6321$ | $I_5^* = 0.1480$ |
| $I_1^* = 0.3679$ | $I_6^* = 0.1120$ |
| $I_2^* = 0.2642$ | $I_7^* = 0.2160$ |
| $I_3^* = 0.2074$ | $I_8^* = -0.728$ |
| $I_4^* = 0.1704$ | $I_9^* = 7.552$ |

由上结果看到 $I_8^* < 0$，与 $I_n = e^{-1}\int_0^1 x^n e^x dx > 0$ 发生矛盾。我们所用的计算公式与每一步计算都是正确的，为什么计算结果是错误的呢？主要原因就是初值 $I_0^*$ 有截断误差 $\varepsilon_0 = I_0 - I_0^*$。根据 $\varepsilon_n = I_n - I_n^*$，可得

$$\varepsilon_n = -n\varepsilon_{n-1} = (-1)^n (n!)\varepsilon_0。 \tag{1.6}$$

上式说明 $I_0^*$ 有误差 $\varepsilon_0$，则 $I_n^*$ 的误差 $\varepsilon_n$ 就是 $\varepsilon_0$ 的（$n!$）倍！误差的累积使正确的结果被湮没了。

现在我们将计算过程倒过来，先求 $I_9$ 的近似值。由积分估值得

$$\frac{e^{-1}}{n+1} = e^{-1}\left(\min_{0\leqslant x\leqslant 1} e^x\right)\int_0^1 x^n dx < I_n < e^{-1}\left(\max_{0\leqslant x\leqslant 1} e^x\right)\int_0^1 x^n dx = \frac{1}{n+1}。$$

取 $n = 9$，可得

$$\frac{e^{-1}}{10} < I_9 < \frac{1}{10},$$

粗略地取 $I_9^{**} = \frac{1}{2}\left(\frac{e^{-1}}{10} + \frac{1}{10}\right) = 0.0684$。然后将公式（1.5）倒过来算，得到

$$I_{n-1}^* = \frac{1}{n}(1 - I_n^*)。$$

将 $n = 8, 7, 6, \cdots 0$ 分别代入上式计算得到结果如下：

| | |
|---|---|
| $I_9^{**} = 0.0684$ | $I_4^{**} = 0.1708$ |
| $I_8^{**} = 0.1035$ | $I_3^{**} = 0.2073$ |
| $I_7^{**} = 0.1121$ | $I_2^{**} = 0.2643$ |
| $I_6^{**} = 0.1268$ | $I_1^{**} = 0.3679$ |
| $I_5^{**} = 0.1455$ | $I_0^{**} = 0.6321$ |

这第二个计算过程使初值 $I_9^{**}$ 的误差逐渐减小，$\varepsilon_0$ 比 $\varepsilon_n$ 缩小到 $\frac{1}{n!}$。

本例说明，第一个计算过程是不稳定的，第二个计算过程是稳定的。如果不注意计算过程的稳定性分析，就会“差之毫厘，失之千里”。

(2) 要防止重要的物理量被"吃掉"

由于计算机的字长有限，当一个比较小的物理量与一个数量级很大的物理量在机内进行加、减运算时，小数可能被大数"吃掉"而引起结果的失真。

例如，使用四位数字浮点计算机(假设为十进制)计算

$$
\begin{aligned}
&0.1986\times10^{3}+0.2478\times10^{-5}\\
\rightarrow\ &10^{3}(0.1986)+10^{3}(0.0000) \qquad \text{(对阶)}\\
\rightarrow\ &10^{3}(0.1986) \qquad \text{(规格化)}
\end{aligned}
$$

其结果大数"吃掉"了小数。当两数的数量级相差没有上例大时，加、减的结果可能部分"吃掉"小数。

又如，已知 $a=5\times10^{10}$，$b=8$，$c=-a$，求 $a$、$b$、$c$ 之和。如果按 $(a+b)+c$ 的次序来编程序，$a$"吃掉"了 $b$，而 $a$ 与 $c$ 又互相抵消，因而结果接近于零；若按 $(a+c)+b$ 来编程序，其结果则为 8。由此可见，如果事先估计计算方案中各量的数量级，编程序时加以合理安排，重要的物理量就可避免被"吃掉"。

(3) 要避免两个接近相等的数相减

两个相近的数相减，有效位数会严重损失。因此，在实际计算时，要尽可能避免出现这种情况。

例如，计算 $x=\sqrt{1001}-\sqrt{1000}$

取 4 位有效数字，$\sqrt{1001}=31.64$，$\sqrt{1000}=31.62$，两者相减得到 $x^*=0.02$，与精确值 $x=0.015807437\cdots$ 比较，只剩下一位有效数字了。

若将原式先变换成

$$x=\sqrt{1001}-\sqrt{1000}=\frac{1}{\sqrt{1001}+\sqrt{1000}},$$

$$x^*=\frac{1}{31.64+31.62}=0.01581,$$

则结果有 4 位有效数字。这是由于避免了两个接近相等的数相减而引起的严重的舍入误差。

类似地，例如当 $x_1$ 很接近于 $x_2$ 时，变换

$$\lg x_1-\lg x_2=\lg\frac{x_1}{x_2},$$

当 $x$ 接近于零时，变换

$$1-\cos x=2\sin^2\frac{x}{2}。$$

当 $f(x^*)$ 很接近于 $f(x)$ 时，用泰勒(Taylor)级数展开

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

$$+\cdots+\frac{(x-x^*)^n}{n!}f^n(x^*)+\cdots$$

如此等等。

若某一问题无法通过变换公式改变算法，可以增加有效数字位数进行运算，使用计算机时则可采用双倍字长运算，以保证结果有一定的有效位数。

（4）要注意简化计算步骤，减少算术运算的次数

对于一个数值计算问题，如果能设法减少运算次数，不但可以节省计算机时间，而且还可以减少累积的舍入误差。

例如，计算多项式

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

的值，若先计算各项的值再求和，要进行$[n+(n-1)+\cdots+1]$次，即$\frac{1}{2}n(n+1)$ 次乘法和 $n$ 次加法。若按下列递推公式

$$\begin{cases} u_n=a_n \\ u_i=xu_{i+1}+a_i \quad (i=n-1,\ n-2,\ \cdots,\ 0) \end{cases}$$

进行计算，

$$\begin{aligned} u_{n-1}&=x\,u_n+a_{n-1}=x\,a_n+a_{n-1}, \\ u_{n-2}&=x\,u_{n-2}+a_{n-2}=x(x\,a_n+a_{n-1})+a_{n-2}, \\ &\vdots \\ u_0&=x\,u_1+a_0=x(x^{n-1}a_n+x^{n-2}a_{n-1}+\cdots+a_1)+a_0=P_n(x) \end{aligned}$$

则只需进行 $n$ 次乘法和 $n$ 次加法运算就可以了。

由上例看出，对于同一个问题，采取不同的计算方法，算术运算的次数可以有较大的差别。设法减少运算次数是数值计算中必须考虑的原则，也是“数值分析”要研究的重要问题之一。

## 习　　题

1.1　当准确数 $I$ 有很多位数时，常常按“四舍五入”的原则得到 $I$ 的近似数 $I^*$。四以下舍，五以上入是肯定的，若刚好是五，则作如下规定：若五前一位是偶数，则将五舍去；若前一位是奇数，则将五进一。按此原则，将下列数舍入成五位数。

913.95873，93.18222，14.32250，17.36750，8.000033，4.280657。

1.2　为了使$\sqrt{30}$的近似数的相对误差小于0.1%，需要取多少位有效数字？

1.3　试说明绝对误差与小数点后的位数有关，相对误差与有效数字的位数有关。

1.4　建立积分 $I_n=\int_0^1\frac{x^n}{x+5}dx$，$n=0,1,\cdots,20$ 的递推关系式，并在计算机上计算之。

1.5　求方程 $x^2-(\alpha+\beta)x+10^9=0$ 的根。这里 $\alpha=10^9$，$\beta=1$。

1.6　求方程 $x^2-56x+1=0$ 的两个根，使它们至少具有四位有效数字。
（$\sqrt{783}=27.983$）

1.7　给定函数 $f(x)=10^7(1-\cos x)$，试由四位数学用表计算 $f(2^\circ)$ 的近似值，并与准确值 6091.73 相比较。

1.8 当 $n$ 充分大时，怎样计算 $\int_n^{n+1} \frac{1}{1+x^2} dx$？

1.9 举例说明在计算机上 $(a+b)c \neq ac+bc$。

1.10 给定 $f(x)=\frac{1-\cos x}{\sin x}$，计算 $x=1^\circ$ 时 $f(x)$ 的值。

# 第2章 插 值

## 2.1 引 言

插值法是一种古老而实用的数值方法。早在一千多年前我国的科学家对插值法就有了研究，并应用于天文学中。例如，人们不能每时每刻连续观测“日月五星”的方位，怎样通过有限次观测的数据得到任一时刻“日月五星”的位置呢？这就需要用到插值的方法。

在许多实际问题中，有相当一部分函数是通过观测或实验得到的。尽管函数 $f(x)$ 在某个区间$[X_1, X_2]$上可能是连续的，但观测或实验只能得到区间内一系列离散点 $x_i$ 上的离散的函数值 $f(x_i)$，$i=0, 1, 2, \cdots, n$，也就是说给出了一个函数表；还有一些问题，函数虽有解析表达式，但由于计算复杂，使用不便，通常也造一个函数表，如三角函数表、对数表、平方根和立方根表等。为了得到不在表上的函数值，我们希望寻求一个便于计算的简单函数 $P(x)$ 去逼近 $f(x)$。若要求 $P(x_i)=f(x_i)$，在其他地方让 $P(x)$近似地代替 $f(x)$，则这类问题就称为**插值问题**。其中 $P(x)$ 称为 $f(x)$ 的**插值函数**，离散点 $x_0, x_1, x_2, \cdots, x_n$ 称为**插值节点**，包含插值节点的区间$[X_1, X_2]$称为**插值区间**。

从几何上看，插值法就是寻求曲线 $y=P(x)$，使其通过给定的 $n+1$ 个点 $(x_i, f_i)$，$i=0, 1, 2, \cdots, n$，并用 $P(x)$ 近似 $f(x)$，如图 2.1(a) 所示。

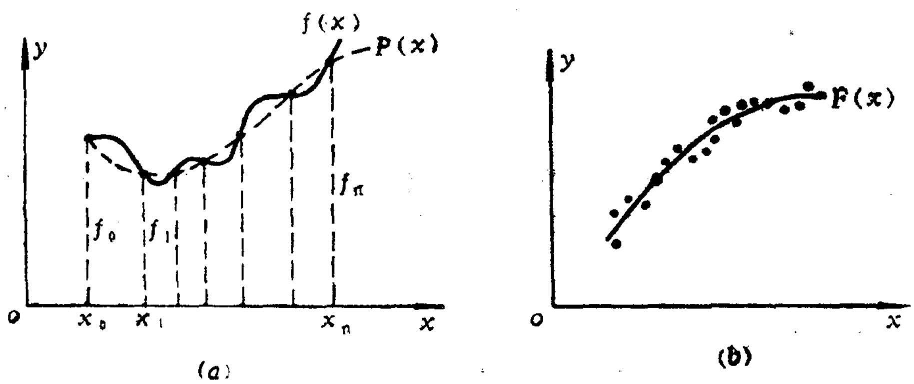



图 2.1

另一类最优拟合问题，即当原始数据含有不可避免的误差（“噪音”）时，寻求一个初等函数 $F(x)$，不是严格地通过已知点而是要求总的偏差为最小（如图 2.1(b) 所示），这不是本章要讨论的内容。

插值理论在微积分产生以后逐步完善起来，其应用也日益增多，插值法在理论上和实践中日益重要。尤其是近十几年发展起来的样条（spline）插值，更获得了广泛的应用。

本章主要研究最常用的一些插值函数、插值方法及误差估计等。

## 2.2 拉格朗日（Lagrange）插值

可供选择的插值函数有各种类型，但最简单而又最常用的是代数多项式。这是由于多项式具有各阶导数，同时求值又非常方便。设原来函数 $f(x)$ 在区间 $[X_1, X_2]$ 上有 $n+1$个

插值节点，则插值函数一般是 $n$ 次多项式，记作

$$P_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n, \tag{2.1}$$

对于这个插值多项式，应满足插值条件

$$P_n(x_i)=f_i, \quad i=0,\ 1,\ 2,\ \cdots,\ n。 \tag{2.2}$$

由上式，得到下述 $n+1$ 个方程，

$$\begin{cases} a_0+a_1x_0+a_2x_0^2+\cdots+a_nx_0^n=f_0, \\ a_0+a_1x_1+a_2x_1^2+\cdots+a_nx_1^n=f_1, \\ \vdots \\ a_0+a_1x_n+a_2x_n^2+\cdots+a_nx_n^n=f_n。 \end{cases} \tag{2.3}$$

解上面线性方程组（2.3），可以求出插值多项式的 $n+1$ 个系数 $a_0$，$a_1$，…，$a_n$。但这个方程组求解是很麻烦的，因而常用另一方法来建立多项式。

我们先建立 $n+1$ 个与每一插值节点有联系的 $n$ 次多项式

$$l_j(x)=\frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)(x_j-x_1)(x_j-x_2)\cdots(x_j-x_{i-1})(x_j-x_{j+1})\cdots(x_j-x_n)}$$

$$=\prod_{\substack{i=0\\i\neq j}}^{n}\frac{x-x_i}{x_j-x_i}, \qquad j=0,\ 1,\ 2,\ \cdots,\ n。$$

多项式 $l_j(x)$ 有如下性质

$$l_j(x_k)=\begin{cases}0, & k\neq j, \\ 1, & k=j。\end{cases}$$

即每一 $n$ 次多项式，除了在 $k=j$ 的节点 $x_k$ 处之外，在其余各节点处，多项式的值均为零。我们称这 $n+1$ 个多项式 $l_0(x)$，$l_1(x)$，…，$l_n(x)$ 为节点 $x_0$，$x_1$，…，$x_n$ 上的**n次插值基函数**。

我们再建立 $l_j(x)$ 的线性组合，为

$$P_n(x)=\sum_{j=0}^{n}f_jl_j(x)=\sum_{j=0}^{n}\prod_{\substack{i=0\\i\neq j}}^{n}\left(\frac{x-x_i}{x_j-x_i}\right)f_j。 \tag{2.4}$$

显然它也是一个 $n$ 次多项式，而且满足插值条件

$$P_n(x_i)=f_i。$$

所以，（2.4）式就是所求的插值多项式，称为**拉格朗日插值公式**。可以证明，它具有存在唯一性。

对给定点 $x$（称为**插值点**），用（2.4）式计算出 $P_n(x)$ 的值，作为 $f(x)$ 在点 $x$ 处的近似值，这个过程叫**插值**。插值多项式的次数叫**插值的阶**。若插值点 $x$ 在插值区间内，这种插值过程称为**内插**，否则叫**外推**（或**外插**）。

下面讨论两个特例。

### 2.2.1 线性插值

利用两个插值节点（$x_0$，$f_0$）及（$x_1$，$f_1$）进行插值，多项式 $P_n(x)$ 为一次，即 $P_1(x)$。

对于 $n=1$，将插值基函数

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

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

代入插值公式（2.4），得

$$P_1(x)=\frac{(x-x_1)}{(x_0-x_1)}f_0+\frac{(x-x_0)}{(x_1-x_0)}f_1$$

$$=f_0+\frac{f_1-f_0}{x_1-x_0}(x-x_0)。\qquad (2.5)$$

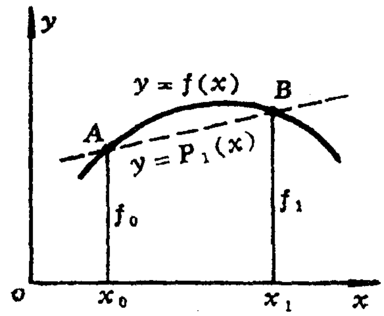



图 2.2

从几何上看，$y=P_1(x)$ 表示通过 $A(x_0, f_0)$ 和 $P(x_1, f_1)$ 的直线（图 2.2），因此两点插值又称为线性插值。

### 2.2.2 抛物插值

与前面类似，利用三个插值节点进行插值，多项式为二次，即 $P_2(x)$。

将 $n=2$ 代入（2.4）式，得到

$$P_2(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f_0+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f_1$$

$$+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f_2。\qquad (2.6)$$

从几何上看，$y=P_2(x)$ 是通过 $A(x_0, f_0)$、$B(x_1, f_1)$ 及 $C(x_2, f_2)$ 三点的抛物线（如图 2.3），因此三点插值又称为抛物插值。

### 2.2.3 误差分析

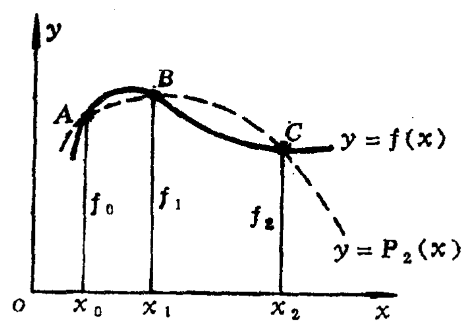



图 2.3

对任意插值点 $x$ 来说，将存在插值误差 $E_n(x)=f(x)-P_n(x)$，它又叫插值余项。

令 $$E_n(x)=\prod_{i=0}^{n}(x-x_i)g(x), \qquad (2.7)$$

$g(x)$ 是用来衡量在插值点处误差的、与 $x$ 及 $n$ 有关的待定函数。由于插值多项式 $P_n(x)$ 与未知函数 $f(x)$ 在各插值节点 $x_0$，$x_1$，…，$x_n$ 处具有相同的函数值（即满足插值条件），所以（2.7）式在 $n+1$ 个节点处满足