目　　录

绪　论

第一章　解线性代数方程组的直接法 …… (4)

1.1　引言 …… (4)

1.2　高斯(Gauss)消去法 …… (4)

1.3　矩阵的三角分解 …… (10)

1.4　解三对角方程组的追赶法 …… (29)

第二章　解线性代数方程组的迭代法 …… (35)

2.1　引言 …… (35)

2.2　基本迭代法 …… (35)

2.3　范数及方程组的性态、条件数 …… (42)

2.4　收敛性分析 …… (48)

2.5　共轭梯度法 …… (58)

第三章　解非线性方程的迭代法 …… (66)

3.1　引言 …… (66)

3.2　二分法 …… (67)

3.3　迭代法 …… (68)

3.4　迭代过程的加速 …… (72)

3.5　牛顿法 …… (74)

3.6　弦截法 …… (80)

3.7　解非线性方程组的迭代法简介 …… (82)

第四章　矩阵特征值与特征向量的计算 …… (94)

4.1　引言 …… (94)

4.2　乘幂法 …… (96)

4.3　反幂法 …… (104)

4.4 雅可比(Jacobi)方法 …… (106)
4.5 QR 方法 …… (112)
第五章 代数插值 …… (120)
5.1 引言 …… (120)
5.2 拉格朗日(Lagrange)插值 …… (122)
5.3 差商与牛顿(Newton)插值 …… (129)
5.4 差分与等距节点插值公式 …… (136)
5.5 埃尔米特(Hermite)插值 …… (140)
5.6 样条插值函数 …… (146)
5.7 B-样条函数 …… (159)
第六章 函数逼近 …… (167)
6.1 引言 …… (167)
6.2 正交多项式 …… (168)
6.3 最佳一致逼近 …… (173)
6.4 最佳平方逼近 …… (185)
6.5 曲线拟合的最小二乘法 …… (195)
6.6 有理函数逼近 …… (205)
第七章 数值积分和数值微分 …… (212)
7.1 引言 …… (212)
7.2 牛顿-柯特斯(Newton-Cotes)求积公式 …… (216)
7.3 龙贝格(Romberg)求积算法 …… (230)
7.4 高斯(Gauss)求积公式 …… (237)
7.5 多重积分的求积公式 …… (249)
7.6 数值微分 …… (257)
7.7 外推法在数值微分中的应用 …… (266)
第八章 常微分方程初值问题的数值解 …… (272)
8.1 引言 …… (272)
8.2 尤拉(Euler)方法及其改进 …… (273)
8.3 龙格-库塔(Runge-Kutta)方法 …… (279)

8.4 线性多步法 …… (291)
8.5 收敛性与稳定性 …… (302)
8.6 一阶方程组和高阶方程 …… (307)
附录 部分算法的数值计算程序 …… (313)
参考书目 …… (350)

绪　论

随着现代科学技术的迅速发展，从实际问题中建立起来的数学模型越来越复杂，由于这些数学模型往往不能简便地求出精确解，于是人们就局限于讨论问题的特殊情形或简化了的模型，但这样做往往不能满足精度要求．因此，常用的处理方法就是对较少简化的数学模型运用数值计算方法在计算机上进行数值计算，计算时间的多少将依赖于数值计算的计算量以及所用计算机的运行速度，而计算量的大小又依赖于所用的数值计算方法与所要达到的精度．因此，要更好更快地解决实际问题中的数学模型，必须提高人类的计算能力，即计算工具的性能与计算方法效率的总和．计算能力的提高有赖于双方，我们不但要改进计算机的硬件设备，提高计算机的运行速度，而且还必须提出具有高精度、低计算量的数值计算方法．对同一问题来讲，采用不同的数值计算方法，其计算工作量有时相差非常之大，比如：对一个 n 阶的线代数方程组，当 $n=20$ 时，若用 Cramer 法则求解，其乘除法运算次数约需 9.7×10^{20}次，即使用每秒运算 1 亿次的计算机也要算 30 多万年，而用 Gauss 消去法大约只需 2660 次乘除法运算，并且，n 愈大，相差就愈大．这个例子表明算法的好坏对计算能力的提高起着重要的作用．在 1955～1975 年的 20 年间，计算机的计算速度提高了数千倍，而同一时间解决一定规模的椭圆型偏微分方程的数值计算方法的效率提高了约 100 万倍．所以，研究和选择好的计算方法是非常重要的．

由于我们讨论的数值计算方法是以计算机作为计算工具的，因此还必须考虑到有限字长给运算所带来的误差．若用计算机来解决一个实际问题，计算机算出的值与实际问题的值往往存在着

差异，两者之差称为**误差**，而引起误差的原因是多方面的，为了解决一个实际问题，首先必须对这个问题建立合理的数学模型，由于建模时要忽略某些次要因素，对原始问题进行近似，这个过程中产生的误差称之为**模型误差**，而数学模型中包含的若干参量往往是通过观测得到的，这种观测也难免产生误差，称它为**观测误差**. 然后还要把数学模型运用数值计算方法转换成数值问题，这个过程引起的误差称为**截断误差**或**方法误差**. 最后，用计算机对数值问题进行求解时，由于计算机的有限字长还会产生**舍入误差**. 在上面的所有这些误差中，模型误差和观测误差不属于本课程的讨论范围，本课程主要研究从数学模型到数值问题之间的误差，即方法误差，舍入误差也是我们要研究的，另外，数值计算方法的稳定与否还会影响到舍入误差的增长，我们可以来看这样的一个例子：对 $n=0,1,\cdots,8$，计算积分

$$S_n=\int_0^1\frac{x^n}{x+5}\mathrm{d}x,$$

由于

$$S_n+5S_{n-1}=\int_0^1\frac{x^n+5x^{n-1}}{x+5}\mathrm{d}x=\int_0^1 x^{n-1}\mathrm{d}x=\frac{1}{n},$$

$$S_0=\int_0^1\frac{1}{x+5}\mathrm{d}x=\ln 6-\ln 5=\ln(1.2).$$

若取 $S_0=\ln(1.2)\approx 0.182$，用公式

$$\bar{S}_n=\frac{1}{n}-5\bar{S}_{n-1}\quad(n=1,2,\cdots,8)$$

进行计算（精确到小数点后 3 位），我们得到

$$\bar{S}_1\approx 0.090,\quad \bar{S}_2\approx 0.050,\quad \bar{S}_3\approx 0.083,\quad \bar{S}_4\approx -0.165,$$

$$\bar{S}_5\approx 1.025,\quad \bar{S}_6\approx -4.958,\quad \bar{S}_7\approx 24.933,\quad \bar{S}_8\approx -124.540,$$

若记 $\varepsilon_n=S_n-\bar{S}_n$，则

$$\varepsilon_n=-5\varepsilon_{n-1},$$

显然，如果从 $\bar{S}_{n-1}$ 计算 $\bar{S}_n$，误差将以每步 5 倍的速度增长，这种计

算是不稳定的，反之，如果从 $\bar{S}_n$ 计算 $\bar{S}_{n-1}$，即

$$\bar{S}_{n-1}=\frac{1}{5n}-\frac{1}{5}\bar{S}_n \quad (n=8,7,\cdots,1)$$

则误差的增长速度为每步 20%，即误差逐步减少，计算稳定，所以我们可以这样确定 $\bar{S}_n$，设 $\bar{S}_8\approx\bar{S}_9$，从

$$\bar{S}_8=\frac{1}{5\times 9}-\frac{1}{5}\bar{S}_9$$

可得 $\bar{S}_8\approx\bar{S}_9\approx 0.019$，再由逐次计算求得(精确到小数点后 3 位)

$$S_7\approx 0.021,\quad S_6\approx 0.024,\quad S_5\approx 0.029,\quad S_4\approx 0.034,$$
$$\bar{S}_3\approx 0.043,\quad \bar{S}_2\approx 0.058,\quad \bar{S}_1\approx 0.088,\quad ,\bar{S}_0\approx 0.182$$

它与实际结果相近.

因此，在讨论数值计算方法时还必须考虑算法的稳定性，数值不稳定的算法是不能使用的.

本书针对工科研究生的实际情况，不仅对经典的数值计算方法作了较为详细的介绍，而且还在此基础上介绍了近代发展起来的一些有效的、常用的算法，如预条件共轭梯度法等. 最后还对部分算法编制了数值计算程序.

通过本书的学习，不仅可对基本的数值计算方法有充分的了解，而且还可提高学生运用不同的方法来有效地解决不同问题的能力.

第一章　解线性代数方程组的直接法

1.1　引　言

在自然科学与社会科学的研究中，常常需要求解线性代数方程组．如实验数据的曲线、曲面的拟合和用差分法或有限元法解偏微分方程等都要用到线性代数方程组的求解．由于从不同的问题导出的线性代数方程组的系数矩阵不同，比如：矩阵阶数的大小、矩阵中的非零元稠密情况等，粗略地，系数矩阵可以分为低阶稠密矩阵和大型稀疏矩阵．关于线性代数方程组的求解，主要分为直接法和迭代法两大类．在理论上，用直接法可以通过有限步的计算得到精确解，而迭代法是通过逐次迭代逼近来求得近似解．实际上，由于舍入误差的影响，由直接法得到的解也不精确，因此，在某些需要高精度解的问题中，常常把由直接法得到的解再运用迭代法迭代若干步，以提高解的精度．一般地说，对于低阶稠密的线性代数方程组以及大型带形方程组的求解，采用直接法比较有效，而对于大型稀疏（非带形）方程组则用迭代法求解比较有利．当然，采用直接法，还是迭代法，还是直接法与迭代法交替运用，要根据具体情况确定．

本章主要讨论一些最基本的直接法，并在此基础上讨论它的变形情况，对于求解线性代数方程组的迭代法，我们将在下一章中介绍．

1.2　高斯(Gauss)消去法

考虑 n 阶线性代数方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \quad\vdots \qquad\quad \vdots \qquad\quad \cdots \qquad\quad \vdots \qquad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \tag{1.1}$$

采用矩阵和向量记号，我们可以把式(1.1)改写成

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1.2}$$

其中

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

现在，我们给出用高斯消去法求解式(1.2)的计算过程.

算法 1.1 求解线代数方程组(1.2)的高斯消去法

我们先分别记矩阵 $\boldsymbol{A}^{(1)} = \boldsymbol{A}$，向量 $\boldsymbol{b}^{(1)} = \boldsymbol{b}$，它们的元素分别为

$$a_{ij}^{(1)} = a_{ij} \qquad (i,j = 1,2,\cdots,n),$$
$$b_i^{(1)} = b_i \qquad (i = 1,2,\cdots,n).$$

(1) 消元过程

第一步：若 $a_{11}^{(1)} \neq 0$，可对 $i = 2,3,\cdots,n$ 进行如下计算，用 $m_{i1} = -a_{i1}^{(1)}/a_{11}^{(1)}$ 乘方程(1.2)两边的第一行加到第 i 行中去，得到

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(2)} \\ \vdots \\ b_n^{(2)} \end{bmatrix}, \tag{1.3}$$

这里，

$$a_{ij}^{(2)} = a_{ij}^{(1)} + m_{i1} \cdot a_{1j}^{(1)}, \qquad (i,j = 2,3,\cdots,n),$$
$$b_i^{(2)} = b_i^{(1)} + m_{i1} \cdot b_1^{(1)}, \qquad (i = 2,3,\cdots,n).$$

第二步：若 $a_{22}^{(2)} \neq 0$，可对 $i = 3,\cdots,n$ 进行如下计算，用 $m_{i2} = -a_{i2}^{(2)}/a_{22}^{(2)}$ 乘方程组(1.3)两边的第二行加到第 i 行中去，得到

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(3)} \\ \vdots \\ b_n^{(3)} \end{bmatrix},$$

这里，

$$a_{ij}^{(3)} = a_{ij}^{(2)} + m_{i2} \cdot a_{2j}^{(2)}, \qquad (i,j = 3,\cdots,n),$$
$$b_i^{(3)} = b_i^{(2)} + m_{i2} \cdot b_2^{(2)}, \qquad (i = 3,\cdots,n).$$

依次下去，……，一直做到第 $(n-1)$ 步，即：

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,n-1}^{(1)} & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2,n-1}^{(2)} & a_{2n}^{(2)} \\ \vdots & & \ddots & \vdots & \vdots \\ \vdots & & \ddots & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ 0 & & \cdots & 0 & a_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_{n-1}^{(n-1)} \\ b_n^{(n)} \end{bmatrix}. \tag{1.4}$$

(2) 回代过程

若 $a_{nn}^{(n)} \neq 0$，我们可以从式(1.4)逐次回代计算出方程(1.2)的解：

$$\begin{cases} x_n = b_n^{(n)}/a_{nn}^{(n)} \\ x_k = \left(b_k^{(k)} - \sum_{s=k+1}^{n} a_{ks}^{(k)} x_s\right)/a_{kk}^{(k)}, \quad (k = n-1,\cdots,2,1). \end{cases}$$

如果在消元过程的第一步中，$a_{11}^{(1)} = 0$，由于矩阵 A 非奇，所以

在 $A^{(1)}$ 的第一列中至少有一个元素 $a_{i_1 1}\neq 0$，于是可以通过对方程两边进行第一行与第 i_1 行的行交换，将 $a_{i_1 1}$ 调到(1,1)位置，然后进行消元计算. 同理，对以后各步都可以进行这样的行交换，以使 $a_{22}^{(2)}, a_{33}^{(3)}, \cdots, a_{n-1,n-1}^{(n-1)}$ 均不为零. 最后，由 A 的非奇异性得到 $a_{nn}^{(n)}$ 亦不为零.

纵观算法 1.1，我们得到：

定理 1.1 如果方程组(1.2)的系数矩阵 A 为非奇，则它可以通过高斯消去法(即算法 1.1)进行求解.

下面，我们分析在消元过程中不需进行行交换的高斯消去法的乘除运算工作量.

(1) 消元过程的计算量. 在第 k 步的消元中，对矩阵消元需作 $(n-k)$ 次除法运算和 $(n-k)^2$ 次乘法运算，对右端需作 $(n-k)$ 次乘法运算，于是，总的消元过程乘除法运算量为

$$\sum_{k=1}^{n-1}(n-k)^3+\sum_{k=1}^{n-1}(n-k)^2+\sum_{k=1}^{n-1}(n-k)=\frac{1}{3}n^3+\frac{1}{2}n^2-\frac{5}{6}n.$$

(2) 回代过程的计算量. 计算每个 x_k 需作 $(n-k)$ 次乘法运算，于是总的回代过程乘除法运算量为

$$\sum_{k=1}^{n-1}(n-k)=\frac{1}{2}n(n+1).$$

因此，用不需进行行交换的高斯消去法计算线代数方程组(1.2)所需的总的乘除运算次数为

$$\frac{1}{3}n^3+\frac{1}{2}n^2-\frac{5}{6}n+\frac{1}{2}n(n+1)=\frac{1}{3}n^3+n^2-\frac{1}{3}n.$$

如果在消元过程中考虑进行行交换，则第 k 个消元步至多需要 $(n-k)$ 次的非零判断以及 $(n-k)$ 次两个元素的对换. 于是，总的非零判断次数最多为

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

总的两个元素的对换次数最多为

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

另外，如果在算法 1.1 的消元过程中，对每一个当前消元步，比如第 k 步，都进行这样的判断，必要时，还进行行交换，以使得 $|a_{kk}^{(k)}|\geqslant|a_{ik}^{(k)}|$，从而 $|m_{ik}|\leqslant 1,(i=k+1,\cdots,n)$. 这样，在消元过程和回代过程中，就避免了绝对值较小的数作为除数的情况，于是，计算过程稳定. 我们称经这种修改后的算法 1.1 为列选主元高斯消去法.

下面我们给出不选主元和列选主元高斯消去法计算线代数方程组的数值计算例子.

例 1.1 试用高斯消去法计算线代数方程组

$$\begin{bmatrix}1&1&1&1&1\\2&1&1&1&1\\3&2&1&1&1\\4&3&2&1&1\\5&4&3&2&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix}=\begin{bmatrix}5\\6\\8\\11\\15\end{bmatrix}$$

解 首先进行消元过程，对方程组两边第 1 行分别乘(－2)加到第 2 行、乘(－3)加到第 3 行、乘(－4)加到第 4 行、乘(－5)加到第 5 行，得到

$$\begin{bmatrix}1&1&1&1&1\\0&-1&-1&-1&-1\\0&-1&-2&-2&-2\\0&-1&-2&-3&-3\\0&-1&-2&-3&-4\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix}=\begin{bmatrix}5\\-4\\-7\\-9\\-10\end{bmatrix}$$

对方程组两边第 2 行乘(－1)分别加到第 3 行、第 4 行、第 5 行，得到

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & -1 & -2 & -2 \\ 0 & 0 & -1 & -2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 5 \\ -4 \\ -3 \\ -5 \\ -6 \end{bmatrix}.$$

对方程组两边第 3 行乘(-1)分别加到第 4 行、第 5 行,得到

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 5 \\ -4 \\ -3 \\ -2 \\ -3 \end{bmatrix}$$

对方程组两边第 4 行乘(-1)加到第 5 行,得到

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 5 \\ -4 \\ -3 \\ -2 \\ -1 \end{bmatrix}$$

然后,回代计算出

$$x_5 = 1, \quad x_4 = 1, \quad x_3 = 1, \quad x_2 = 1, \quad x_1 = 1.$$

例 1.2 采用四位有效数字,分别用不选主元高斯消去法和列选主元高斯消去法计算线性代数方程组

$$\begin{bmatrix} 0.003000 & 59.14 \\ 5.291 & -6.130 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 59.17 \\ 46.78 \end{bmatrix}, \tag{1.5}$$

(此方程组的精确解为:$x_1 = 10.00, x_2 = 1.000$).

解 首先用不选主元法高斯消去法计算式(1.5).

消元计算(考虑四位有效数字):

$$m_{21} = -\frac{5.291}{0.0030000} = -1763.66 \approx -1764,$$

对方程组消元，于是得到

$$\begin{bmatrix} 0.0030000 & 59.14 \\ 0 & -104300 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 59.17 \\ -104400 \end{bmatrix}.$$

回代计算：

$$x_2 = 1.001, \quad x_1 = -10.00$$

由此看出，在计算 x_2 时产生的误差 0.001，导致了求解 x_1 时产生更大的误差，即误差扩大了 20000 倍.

下面采用列选主元高斯消去法计算式(1.5)，交换方程组(1.5)的第 1 行与第 2 行得

$$\begin{bmatrix} 5.291 & -6.130 \\ 0.003000 & 59.14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 46.78 \\ 59.17 \end{bmatrix}.$$

消元计算(考虑四位有效数字)：

$$m_{21} = -\frac{0.0030000}{5.291} = -0.0005670$$

方程组消元后得

$$\begin{bmatrix} 5.291 & -6.130 \\ 0 & 59.14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 46.78 \\ 59.14 \end{bmatrix}.$$

回代计算：

$$x_2 = 1.000, \quad x_1 = 10.00.$$

与不选主元的高斯消去法相比，显然用列选主元高斯消去法可以得到好的结果.

1.3 矩阵的三角分解

在本节中，我们将讨论基于高斯消去法思想的另一种求解线性代数方程组的直接法——矩阵的三角分解.

回顾§1.2节中的算法1.1,当消元过程不需作行交换以调整对角线上非零元时,算法1.1的消元过程与下面等价:

第一步等价于 $a_{11}^{(1)}\neq 0$,并用矩阵

$$L_1=\begin{bmatrix} 1 & & & \\ m_{21} & 1 & & \\ m_{31} & & 1 & \\ \vdots & & & \ddots \\ m_{n1} & & & & 1 \end{bmatrix},\quad m_{i1}=-\frac{a_{i1}^{(1)}}{a_{11}^{(1)}}$$

乘方程(1.2)的两边,左边得到

$$L_1A^{(1)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}=:A^{(2)},\qquad (1.6)$$

第二步等价于 $a_{22}^{(2)}\neq 0$,并用矩阵

$$L_2=\begin{bmatrix} 1 & & & \\ & 1 & & \\ & m_{32} & 1 & \\ & \vdots & & \ddots \\ & m_{n2} & & & 1 \end{bmatrix},\quad m_{i2}=-\frac{a_{i2}^{(2)}}{a_{22}^{(2)}}$$

乘方程(1.3)的两边,左边得到

$$L_2A^{(2)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{bmatrix}=:A^{(3)},$$

依次下去,……,一直进行到第 $n-1$ 步,方程左边为

$$L_{n-1}A^{(n-1)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,n-1}^{(1)} & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2,n-1}^{(2)} & a_{2n}^{(2)} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ & & & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ 0 & \cdots & \cdots & 0 & a_{nn}^{(n)} \end{bmatrix}=:A^{(n)},$$

这里

$$L_{n-1}=\begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & m_{n,n-1} & 1 \end{bmatrix},\quad m_{n,n-1}=-\frac{a_{n,n-1}^{(n-1)}}{a_{n-1,n-1}^{(n-1)}}$$

综合上述 $n-1$ 步，我们得到

$$A=A^{(1)}=L_1^{-1}A^{(2)}=\cdots=L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}A^{(n)}. \tag{1.7}$$

由 L_i 的定义和简单计算知

$$L_i^{-1}=\begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{i+1,i} & 1 & \\ & & \cdots & & \ddots \\ & & -m_{n,i} & & 1 \end{bmatrix},$$

$$L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}=\begin{bmatrix} 1 & & & \\ -m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ -m_{n-1,1} & -m_{n-1,2} & \cdots & 1 \\ -m_{n1} & -m_{n,1} & \cdots & -m_{n,n-1} & 1 \end{bmatrix}$$

记

$$L=L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1},\quad U=A^{(n)}.$$

对照式(1.7)，矩阵 A 可以分解成单位下三角阵 L 和上三角阵 U

的乘积. 这种分解过程称为矩阵的杜利脱尔(Doolittle)分解.

上述消元过程也可以通过下列直接紧凑格式分解完成:

算法 1.2 矩阵 $A \in R^{n\times n}$的杜利脱尔分解.

考虑分解

$$A=\begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix}\cdot\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix}=:LU.$$

第一步:用 L 的第 1 行乘 U 的第 $j(1\leqslant j\leqslant n)$列,再比较两边得

$$a_{1j}=u_{1j},\quad (j=1,2,\cdots,n).$$

用 L 的第 $i(2\leqslant i\leqslant n)$行乘 U 的第 1 列,再比较两边得

$$a_{i1}=l_{i1}\cdot u_{11},$$

于是

$$l_{i1}=a_{i1}/u_{11},\quad (i=2,3,\cdots,n).$$

第二步:用 L 的第 2 行乘 U 的第 $j(2\leqslant j\leqslant n)$列,再比较两边得

$$a_{2j}=l_{21}\cdot u_{1j}+u_{2j}.$$

于是

$$u_{2j}=a_{2j}-l_{21}\cdot u_{1j},\quad (j=2,3,\cdots,n).$$

用 L 的第 $i(3\leqslant i\leqslant n)$行乘 U 的第 2 列,再比较两边得

$$a_{i2}=l_{i1}\cdot u_{12}+l_{i2}\cdot u_{22},$$

于是

$$l_{i2}=(a_{i2}-l_{i1}\cdot u_{12})/u_{22},\quad (i=3,\cdots,n).$$

依次下去,……,一直进行到第 $k-1$ 步,即已求出 U 的前 $k-1$ 行和 L 的前 $k-1$ 列,那么,有

第 k 步:用 L 的第 k 行乘 U 的第 $j(k\leqslant j\leqslant n)$列,再比较两边

得

$$a_{kj} = l_{k1} \cdot u_{1j} + \cdots + l_{k,k-1} \cdot u_{k-1,j} + u_{kj}.$$

于是

$$u_{kj} = a_{kj} - (l_{k1} \cdot u_{1j} + \cdots + l_{k,k-1} \cdot u_{k-1,j}), \quad (j = k, k+1, \cdots, n).$$

用 L 的第 $i(k+1 \leqslant i \leqslant n)$行乘 L 的第 k 列，再比较两边得

$$a_{ik} = l_{i1} \cdot u_{1k} + \cdots + l_{i,k-1} \cdot u_{k-1,k} + l_{ik} \cdot u_{kk}.$$

于是

$$l_{ik} = (a_{ik} - l_{i1} \cdot u_{1k} - \cdots - l_{i,k-1} \cdot u_{k-1,k}) / u_{kk}, \quad (i = k+1, \cdots, n).$$

……

第 n 步：用 L 的第 n 行乘 U 的第 n 列，再比较两边得

$$a_{nn} = l_{n1} u_{1n} + \cdots + l_{n,n-1} u_{n-1,n} + u_{nn}.$$

于是

$$u_{nn} = a_{nn} - (l_{n1} u_{1n} + \cdots + l_{n,n-1} u_{n-1,n}).$$

上述 n 步计算过程可以综合改写成如下的计算公式：

对 $k = 1, 2, \cdots, n$ 计算：

$$\begin{cases} u_{kj} = a_{kj} - \sum_{s=1}^{k-1} l_{ks} \cdot u_{sj} & (j = k, k+1, \cdots, n); \\ l_{ik} = (a_{ik} - \sum_{s=1}^{k-1} l_{is} \cdot u_{sk}) / u_{kk}, & (i = k+1, \cdots, n), \end{cases}$$

在上式和本书的以后各处，都规定 s 从 1 到 0 求和 $\sum_{s=1}^{0}$ 为零.

如果一个矩阵 A 可以分解成一个单位下三角阵 L 与一个上三角阵 U 的乘积，那么求方程组

$$Ax = b$$

的解就可以通过求解两个三角方程组

$$\begin{cases} Ly = b, \\ Ux = y \end{cases}$$

来得到.

例 1.3 用杜利脱尔分解法求解方程组

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 5 \\ 1 & 3 & 6 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 5 \end{bmatrix}.$$

解 先用算法 1.2 进行矩阵的三角分解,即

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 5 \\ 1 & 3 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} =: \boldsymbol{LU}.$$

第一步可以算出

$$u_{11} = 1, \quad u_{12} = 2, \quad u_{13} = 3,$$
$$l_{21} = 1, \quad l_{31} = 1.$$

第二步可以算出

$$u_{22} = 1, \quad u_{23} = 2,$$
$$l_{32} = 1.$$

第三步可以算出

$$u_{33} = 1.$$

于是

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}.$$

求解

$$\boldsymbol{L}y = b, \quad \text{得到} \quad y = [2,2,1]^{\mathrm{T}},$$
$$\boldsymbol{U}x = y, \quad \text{得到} \quad x = [-1,0,1]^{\mathrm{T}}.$$

下面的结果给出了算法 1.2 可以分解到底的条件:

定理 1.2 如果非奇异矩阵 A 满足下列三个条件之一:

(1) 各阶顺序主子式

$$\Delta_k = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{vmatrix} \neq 0, \quad (k = 1, 2, \cdots, n);$$

(2)
$$\begin{cases} |a_{11}| > \sum_{j=2}^{n} |a_{1j}|, \\ |a_{ii}| \geqslant \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad (i = 2, \cdots, n); \end{cases}$$

(3) 对称正定，

则矩阵 A 可以进行杜利脱尔分解：

$$A = LU,$$

并且这种分解是唯一的，其中 L 为单位下三角阵，U 为上三角阵.

证明 首先证明杜利脱尔分解的存在性. 从算法 1.1 和算法 1.2 知，当 $a_{11}^{(1)}, a_{22}^{(2)}, \cdots, a_{nn}^{(n)}$ 均不为零时，杜利脱尔分解的计算过程可以进行到底，于是，我们仅需证明如果条件(1)、(2)、(3)之一成立，就有 $a_{kk}^{(k)} \neq 0 (k = 1, 2, \cdots, n)$ 即可.

(1) 类似于算法 1.1 的消元过程，我们计算

$$\Delta_k = \begin{vmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2k}^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ a_{k1}^{(1)} & a_{k2}^{(1)} & \cdots & a_{kk}^{(1)} \end{vmatrix} = \begin{vmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & a_{k2}^{(2)} & \cdots & a_{kk}^{(2)} \end{vmatrix}$$

$$= \cdots = \begin{vmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,k-1}^{(1)} & a_{1k}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2,k-1}^{(2)} & a_{2k}^{(2)} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ & & & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} \\ 0 & & \cdots & 0 & a_{kk}^{(k)} \end{vmatrix}$$

$$= a_{11}^{(1)} a_{22}^{(2)} \cdots a_{kk}^{(k)}.$$

因此，当 $\Delta_k \neq 0(k=1,2,\cdots,n)$ 时，有 $a_{kk}^{(k)} \neq 0(k=1,2,\cdots,n)$.

(2) 由假定条件知 $|a_{11}^{(1)}| = |a_{11}| > \sum_{j=2}^{n} |a_{1j}| \geqslant 0$，即 $|a_{11}^{(1)}| \neq 0$.

经过一步消元过程后，得到的矩阵形式为

$$\begin{bmatrix} a_{11} & \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{O} & \boldsymbol{A}_2 \end{bmatrix},$$

由简单计算知 $\boldsymbol{A}_2$ 中元素为

$$a_{ij}^{(2)} = a_{ij}^{(1)} + m_{i1} a_{1j}^{(1)} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j},$$

并且，对角线元满足下列关系：

$$\begin{aligned} \sum_{\substack{j=2 \\ j\neq i}}^{n} |a_{ij}^{(2)}| &\leqslant \sum_{\substack{j=2 \\ j\neq i}}^{n} \left(|a_{ij}| + \left| \frac{a_{i1}}{a_{11}} a_{1j} \right| \right) \\ &\leqslant \sum_{\substack{j=2 \\ j\neq i}}^{n} |a_{ij}| + \left| \frac{a_{i1}}{a_{11}} \right| \sum_{\substack{j=2 \\ j\neq i}}^{n} (|a_{1j}| + |a_{1i}| - |a_{1i}|) \\ &< \sum_{\substack{j=2 \\ j\neq i}}^{n} |a_{ij}| + |a_{i1}| - \left| \frac{a_{i1}}{a_{11}} \right| \cdot |a_{1i}| \\ &\leqslant \left| a_{ii} - \frac{a_{i1}}{a_{11}} a_{1i} \right| \leqslant |a_{ii}^{(2)}|. \end{aligned}$$

于是有 $a_{22}^{(2)} \neq 0$. 同理可以证明 $a_{kk}^{(k)} \neq 0, (k=3,\cdots,n)$.

(3) 假如存在某个顺序主子式 $\Delta_k = 0$，则存在 $[z_1,\cdots,z_k]^{\mathrm{T}} \neq 0$ 满足

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix} = 0.$$

于是，有

$$\begin{bmatrix} z_1 \\ \vdots \\ z_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} a_{11} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ a_{k1} & \cdots & a_{kk} & a_{k,k+1} & \cdots & a_{kn} \\ a_{k+1,1} & \cdots & a_{k+1,k} & a_{k+1,k+1} & \cdots & a_{k+1,n} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ a_{n1} & \cdots & a_{nk} & a_{n,k+1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} = 0,$$

这与 $\boldsymbol{A}$ 为正定阵矛盾，因此，$\boldsymbol{A}$ 的各阶顺序主子式非零．由式(1)知，$a_{kk}^{(k)} \neq 0, (k = 1, 2, \cdots, n)$．

现在证明杜利脱尔分解的唯一性．假定 $\boldsymbol{A}$ 有两个杜利脱尔分解：

$$\boldsymbol{A} = \boldsymbol{L}_1 \boldsymbol{U}_1 = \boldsymbol{L}_2 \boldsymbol{U}_2,$$

其中，$\boldsymbol{L}_1, \boldsymbol{L}_2$ 为单位下三角阵，$\boldsymbol{U}_1, \boldsymbol{U}_2$ 为上三角阵．由于 $\boldsymbol{A}$ 非奇，则 $\boldsymbol{U}_1$ 亦非奇，于是

$$\boldsymbol{L}_2^{-1} \boldsymbol{L}_1 = \boldsymbol{U}_2 \boldsymbol{U}_1^{-1},$$

上式右边为上三角阵，左边为单位下三角阵，所以

$$\boldsymbol{L}_2^{-1} \boldsymbol{L}_1 = \boldsymbol{U}_2 \boldsymbol{U}_1^{-1} = \boldsymbol{I},$$

即

$$\boldsymbol{L}_1 = \boldsymbol{L}_2, \quad \boldsymbol{U}_1 = \boldsymbol{U}_2.$$

唯一性得证．

矩阵的三角分解除了杜利脱尔分解之外，还有其他一些变形情况．

定理 1.3 如果非奇异矩阵 $\boldsymbol{A}$ 满足定理 1.2 中的三个条件之一，则

(1) $\boldsymbol{A}$ 有唯一的三角分解：

$$\boldsymbol{A} = \boldsymbol{LDR},$$

其中，$\boldsymbol{L}$ 为单位下三角阵，$\boldsymbol{D}$ 为对角阵，$\boldsymbol{R}$ 为单位上三角阵．

(2) $\boldsymbol{A}$ 有唯一的克洛脱(Crout)分解

$$\boldsymbol{A} = \bar{\boldsymbol{L}} \cdot \bar{\boldsymbol{U}},$$

其中，$\bar{\boldsymbol{L}}$ 为下三角阵，$\bar{\boldsymbol{U}}$ 为单位上三角阵．

证明 （留作习题）.

关于计算矩阵 A 的 LDR 分解和克洛脱分解有下列的计算公式：

算法 1.3 矩阵 A 的 LDR 分解

考虑分解

$$A = LDR,$$

即

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & a_{n,n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & d_{n-1} & 0 \\ 0 & \cdots & 0 & d_n \end{bmatrix} \begin{bmatrix} 1 & r_{12} & \cdots & r_{1n} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & r_{n-1,n} \\ 0 & \cdots & 0 & 1 \end{bmatrix}.$$

可以通过比较上式两边的系数得到矩阵 L, D, R 中元素的计算公式：

对 $k = 1, 2, \cdots, n$，计算

$$\begin{cases} d_k = a_{kk} - \sum_{s=1}^{k-1} l_{ks} d_s r_{sk}, \\ r_{kj} = \left(a_{kj} - \sum_{s=1}^{k-1} l_{ks} d_s r_{sj} \right) / d_k \quad (j = k+1, \cdots, n), \\ l_{ik} = \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} d_s r_{sk} \right) / d_k \quad (i = k+1, \cdots, n). \end{cases}$$

算法 1.4 矩阵 A 的克洛脱分解.

考虑分解

$$A = \bar{L} \cdot \bar{U},$$

即

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

$$= \begin{bmatrix} \bar{l}_{11} & 0 & \cdots & 0 \\ \bar{l}_{21} & \bar{l}_{22} & \ddots & \vdots \\ \vdots & \ddots & \bar{l}_{n-1,n-1} & 0 \\ \bar{l}_{n1} & \cdots & \bar{l}_{n,n-1} & \bar{l}_{nn} \end{bmatrix} \begin{bmatrix} 1 & \bar{u}_{12} & \cdots & \bar{u}_{1n} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \bar{u}_{n-1,n} \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

可以通过比较上式两边的系数得到矩阵 $\bar{L}, \bar{U}$ 中元素的计算公式：

对 $k = 1, 2, \cdots, n$ 计算

$$\begin{cases} \bar{l}_{jk} = a_{jk} - \sum_{s=1}^{k-1} \bar{l}_{js} \cdot \bar{u}_{sk}, & (j = k+1, \cdots, n), \\ \bar{u}_{kj} = \left(a_{kj} - \sum_{s=1}^{k-1} \bar{l}_{ks} \cdot \bar{u}_{sj} \right) / \bar{l}_{kk}, & (j = k, k+1, \cdots, n), \end{cases}$$

当矩阵 $\boldsymbol{A}$ 为对称正定时，我们还有下面更好的计算方法：

算法 1.5 给定对称正定矩阵 $\boldsymbol{A}$，用乔列斯基(Cholesky)分解求解线性代数方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 的平方根法.

(1) 先考虑考乔列斯基分解：

$$\boldsymbol{A} = \begin{bmatrix} \hat{l}_{11} & 0 & \cdots & 0 \\ \hat{l}_{21} & \hat{l}_{22} & \cdots & \vdots \\ \vdots & \vdots & \vdots & 0 \\ \hat{l}_{n1} & \hat{l}_{n2} & \cdots & \hat{l}_{nn} \end{bmatrix} \cdot \begin{bmatrix} \hat{l}_{11} & \hat{l}_{12} & \cdots & \hat{l}_{n1} \\ 0 & \hat{l}_{22} & \cdots & \hat{l}_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \hat{l}_{nn} \end{bmatrix} = : \hat{\boldsymbol{L}} \hat{\boldsymbol{L}}^{\mathrm{T}}.$$

第一步：对 $i = 1, \cdots, n$，用 $\boldsymbol{L}$ 的第 i 行乘 $\boldsymbol{L}^{\mathrm{T}}$ 的第 1 列，再比较两边得

$$a_{11} = \hat{l}_{11} \hat{l}_{11},$$

$$a_{i1} = \hat{l}_{11}\hat{l}_{i1} \quad (i = 2, \cdots, n).$$

从而

$$\hat{l}_{11} = \sqrt{a_{11}},$$

$$\hat{l}_{i1} = a_{i1} / \hat{l}_{11} \quad (i = 2, \cdots, n).$$

第二步：对 $i = 2, \cdots, n$，用 $\boldsymbol{L}$ 的第 i 行乘 $\boldsymbol{L}^{\mathrm{T}}$ 的第 2 列，再比较两边得

$$a_{22} = \hat{l}_{21}^2 + \hat{l}_{22}^2,$$

$$a_{i2} = \hat{l}_{i1}\hat{l}_{21} + \hat{l}_{i2}\hat{l}_{22} \quad (i = 3, \cdots, n).$$

从而

$$\hat{l}_{22} = \sqrt{a_{22} - \hat{l}_{21}^2},$$

$$\hat{l}_{i2} = (a_{i2} - \hat{l}_{i1}\hat{l}_{21}) / \hat{l}_{22} \quad (i = 3, \cdots, n).$$

依次下去，……，一直进行到第 $(n-1)$ 步.

第 n 步：对 $i = n$，用 $\hat{\boldsymbol{L}}$ 的第 n 行乘 $\hat{\boldsymbol{L}}^{\mathrm{T}}$ 的第 n 列，再比较两边得

$$a_{nn} = \hat{l}_{n1}^2 + \hat{l}_{n2}^2 + \cdots + \hat{l}_{n,n-1}^2 + \hat{l}_{nn}^2$$

从而

$$\hat{l}_{nn} = \sqrt{a_{nn} - \sum_{s=1}^{n-1} \hat{l}_{ns}^2}$$

上述的 n 步计算格式可以写成如下的计算公式：

对 $i = 1, 2, \cdots, n$ 计算

$$\begin{cases} \hat{l}_{ii} = \sqrt{a_{ii} - \sum_{s=1}^{k-1} \hat{l}_{is}^2} \\ \hat{l}_{ki} = \left(a_{ki} - \sum_{s=1}^{i-1} \hat{l}_{is} \cdot \hat{l}_{ks} \right) / \hat{l}_{ii}, \quad (k = i+1, i+2, \cdots, n). \end{cases}$$

(2) 回代过程：令

$$\begin{cases} \hat{\boldsymbol{L}}\boldsymbol{y} = \boldsymbol{b}, \\ \hat{\boldsymbol{L}}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{y}. \end{cases}$$

逐次回代计算方程组的解：

$$\begin{cases} y_1 = b_1 / \hat{l}_{11}, \\ y_k = \left(b_k - \sum_{s=1}^{k-1} \hat{l}_{ks} y_s \right) / \hat{l}_{kk}, \quad (k = 2,3,\cdots,n), \end{cases}$$

$$\begin{cases} x_n = y_n / \hat{l}_{nn}, \\ x_k = \left(y_k - \sum_{s=k+1}^{n} \hat{l}_{sk} x_s \right) / \hat{l}_{kk}, \quad (k = n-1,\cdots,2,1). \end{cases}$$

由于解对称正定方程组的平方根法在进行乔列斯基分解时需要将对角元作开方运算，为了避免开方运算，可采用对称正定矩阵的分解式：

$$\boldsymbol{A} = \boldsymbol{L}\,\boldsymbol{D}\,\boldsymbol{L}^{\mathrm{T}}.$$

下面我们给出求解对称正定方程组的改进平方根法：

算法 1.6 给定对称正定矩阵 $\boldsymbol{A}$，求解线性代数方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 的改进平方根法.

(1) 先考虑 $\boldsymbol{L}\,\boldsymbol{D}\,\boldsymbol{L}^{\mathrm{T}}$ 分解：

$$\boldsymbol{A} = \boldsymbol{L}\,\boldsymbol{D}\,\boldsymbol{L}^{\mathrm{T}}.$$

即

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & d_{n-1} & 0 \\ 0 & \cdots & 0 & d_n \end{bmatrix} \begin{bmatrix} 1 & l_{21} & \cdots & l_{n1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & l_{n,n-1} \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

通过比较上式两边的系数得到矩阵 $\boldsymbol{L}, \boldsymbol{D}$ 中元素的计算公式：

对 $k = 1,2,\cdots,n$ 计算

$$\begin{cases} d_k = a_{kk} - \sum_{s=1}^{k-1} d_s l_{ks}^2, \\ l_{ik} = \left(a_{kj} - \sum_{s=1}^{k-1} l_{ks} d_s l_{is} \right) / d_k \quad (i = k+1, \cdots, n) \end{cases}$$

(2) 回代过程:令

$$\begin{cases} \boldsymbol{LY} = \boldsymbol{b}, \\ \boldsymbol{L}^{\mathrm{T}} \boldsymbol{X} = \boldsymbol{D}^{-1} \boldsymbol{Y}, \end{cases}$$

逐次回代计算方程组的解:

$$y_k = b_k - \sum_{s=1}^{k-1} l_{ks} y_s, \quad (k = 1, 2, \cdots, n),$$

$$x_k = y_k / d_k - \sum_{s=k+1}^{n} l_{sk} x_s, \quad (k = n-1, \cdots, 2, 1).$$

对一般的非奇异矩阵的三角分解,只要在每个分解步中考虑行交换就可以了. 下面我们仅对一般的非奇异矩阵给出杜利脱尔分解结果,即带行交换的杜利脱尔分解,至于克洛脱分解和 ***LDR*** 分解也可以进行相应的讨论.

定理 1.4 如果矩阵 $\boldsymbol{A} \in R^{n \times n}$ 非奇,则存在 n 阶置换阵 $\boldsymbol{P} = \prod_{k=1}^{n-1} \boldsymbol{P}_{ki_k}$,其中,$k \leqslant i_k$.

$$\boldsymbol{P}_{ki_k} = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & 0 & & & 1 & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & 1 & & & 0 & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \text{第 } k \text{ 行} \\ \\ \\ \\ \text{第 } i_k \text{ 行} \\ \\ \\ \end{matrix}$$

为初等置换阵($k = i_k$ 时为恒等阵),成立

$$PA = LU,$$

这里，L 为单位下三角阵，U 为上三角阵.

证明　由算法 1.1 及矩阵的杜利脱尔分解可以得到

$$L_{n-1}P_{n-1}\cdots L_2P_2L_1P_1A = U, \tag{1.8}$$

其中，$P_k = P_{ki_k}(k \leqslant i_k)$为初等置换阵，$U$ 为上三角阵.

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & m_{n,k} & & & 1 \end{bmatrix},$$

由 P_k 的定义及式(1.8)得

$$\begin{aligned} &L_{n-1}P_{n-1}\cdots L_2P_2L_1P_1 \\ &\quad = L_{n-1}\cdot(P_{n-1}L_2P_{n-1})\cdots(P_{n-1}\cdots P_3L_2P_3\cdots P_{n-1}) \\ &\qquad \cdot(P_{n-1}\cdots P_2L_1P_2\cdots P_{n-1})\cdot(P_{n-1}\cdots P_2P_1) \\ &= L_{n-1}\bar{L}_{n-2}\cdots\bar{L}_1P, \end{aligned}$$

这里

$$\bar{L}_k = P_{n-1}\cdots P_{k+1}L_kP_{k+1}\cdots P_{n-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & \bar{m}_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & \bar{m}_{n,k} & & & 1 \end{bmatrix},$$

$$k = 1,2,\cdots,n-2;$$

$P = P_{n-1}\cdots P_2P_1$.

由简单计算知

$$L := \bar{L}_1^{-1}\cdots\bar{L}_{n-2}^{-1}L_{n-1}^{-1}$$

为单位上三角阵. 因此

$$\boldsymbol{PA} = \bar{\boldsymbol{L}}_1^{-1}\cdots\bar{\boldsymbol{L}}_{n-2}^{-1}\boldsymbol{L}_{n-1}^{-1}\boldsymbol{U} = \boldsymbol{LU}.$$

例 1.4 用杜利脱尔分解计算下列 4 个线性代数方程组：

$$\boldsymbol{Ax}_i = \boldsymbol{b}_i, \quad (i = 1,2,3,4) \tag{1.9}$$

这里

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 3 & 2 \\ 2 & 3 & 4 \end{bmatrix}, \boldsymbol{b}_1 = \begin{bmatrix} 4 \\ 7 \\ 9 \end{bmatrix}, \boldsymbol{b}_2 = \boldsymbol{x}_1, \boldsymbol{b}_3 = \boldsymbol{x}_2, \boldsymbol{b}_4 = \boldsymbol{x}_3.$$

解 首先用杜利脱尔分解对矩阵 $\boldsymbol{A}$ 进行三角分解：

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} =: \boldsymbol{LU}.$$

比较两边系数得

$$\begin{cases} u_{11} = 2, & u_{12} = 1, & u_{13} = 1; \\ l_{21} = 1, & l_{31} = 1; \\ u_{22} = 2, & u_{23} = 1; \\ l_{32} = 1; \\ u_{33} = 2. \end{cases}$$

于是

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}.$$

可以把方程组(1.9)分解为

$$\begin{cases} \boldsymbol{Ly}_i = \boldsymbol{b}_i, \\ \boldsymbol{Ux}_i = \boldsymbol{y}_i, \end{cases} \tag{1.10}$$

对 $i = 1$，可以计算出

$$\boldsymbol{y}_1 = \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix}, \quad \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

由 $\boldsymbol{b}_2 = \boldsymbol{x}_1$，可以从式(1.10)计算出

$$\boldsymbol{y}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 0.5 \\ 0 \\ 0 \end{bmatrix},$$

由 $\boldsymbol{b}_3 = \boldsymbol{x}_2$，可以从式(1.10)计算出

$$\boldsymbol{y}_3 = \frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_3 = \frac{1}{8}\begin{bmatrix} 3 \\ -2 \\ 0 \end{bmatrix},$$

由 $\boldsymbol{b}_4 = \boldsymbol{x}_3$，可以从式(1.10)计算出

$$\boldsymbol{y}_4 = \frac{1}{8}\begin{bmatrix} 3 \\ -5 \\ -2 \end{bmatrix}, \quad \boldsymbol{x}_4 = \frac{1}{16}\begin{bmatrix} 5 \\ -6 \\ 2 \end{bmatrix}.$$

在例 1.3 中，由于后一方程组的右端依赖于前一方程组的解，对第 1 个方程组的计算，三角分解法与高斯消去法的工作量相同，但是，当计算第 2,3,4 个方程组时，采用三角分解法仅需回代计算，计算工作量比高斯消去法大大节省. 这是三角分解法的优点.

例 1.5 用带行交换的杜利脱尔分解计算线性代数方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1.11}$$

这里

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}.$$

解 先进行杜利脱尔分解. 设

$$\boldsymbol{P}_1 = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, \quad \boldsymbol{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix},$$

于是

$$L_1P_1A=\begin{bmatrix}1&1&0\\0&0&1\\0&1&1\end{bmatrix}=A^{(1)}.$$

设

$$P_2=\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix},\quad L_2=\begin{bmatrix}1&&\\&1&\\&&1\end{bmatrix},$$

于是

$$L_2P_2A^{(1)}=\begin{bmatrix}1&1&0\\0&1&1\\0&0&1\end{bmatrix}=A^{(2)}=:U$$

即

$$U=L_2P_2L_1P_1A=P_2L_1A=P_2L_1P_2^{\mathrm{T}}P_2A.$$

令

$$P=P_2,\quad L^{-1}=P_2L_1P_2^{\mathrm{T}},$$

从而

$$L=P_2L_1^{-1}P_2^{\mathrm{T}}=\begin{bmatrix}1&0&0\\1&1&0\\1&0&1\end{bmatrix}.$$

这样,我们得到 A 的杜利脱尔分解

$$PA=LU.$$

由上式,我们可以把方程组(1.11)分解为

$$\begin{cases}Ly=Pb,\\Ux=y.\end{cases}$$

回代计算得到

$$y=\begin{bmatrix}5\\7\\4\end{bmatrix},\quad x=\begin{bmatrix}2\\3\\4\end{bmatrix}.$$

例 1.6 用乔列斯基分解计算线性代数方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1.12}$$

这里

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 5 \\ 9 \\ 12 \\ 14 \\ 15 \end{bmatrix}.$$

解 首先用乔列斯基分解对矩阵 $\boldsymbol{A}$ 进行三角分解：

$$\boldsymbol{A} = \begin{bmatrix} l_{11} & 0 & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} & 0 \\ l_{51} & l_{52} & l_{53} & l_{54} & l_{55} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} & l_{41} & l_{51} \\ 0 & l_{22} & l_{32} & l_{42} & l_{52} \\ 0 & 0 & l_{33} & l_{43} & l_{53} \\ 0 & 0 & 0 & l_{44} & l_{54} \\ 0 & 0 & 0 & 0 & l_{55} \end{bmatrix} =: \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}.$$

比较两边系数得

$$\begin{cases} l_{11} = 1, & l_{21} = 1, & l_{31} = 1, & l_{41} = 1, & l_{51} = 1; \\ l_{22} = 1, & l_{32} = 1, & l_{42} = 1, & l_{52} = 1; \\ l_{33} = 1, & l_{43} = 1, & l_{53} = 1; \\ l_{44} = 1, & l_{54} = 1; \\ l_{55} = 1. \end{cases}$$

于是

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

可以把方程组(1.12)分解为

$$\begin{cases} \boldsymbol{L}\boldsymbol{y} = \boldsymbol{b} \\ \boldsymbol{L}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{y}. \end{cases}$$

回代计算得到

$$\boldsymbol{y} = \begin{bmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

1.4 解三对角方程组的追赶法

从有些实际问题导出的线性代数方程组的系数矩阵是三对角的,也即方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1.13}$$

有形式

$$\boldsymbol{A} = \begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix},$$

并且,满足条件:

(1) $c_i \neq 0, a_i, b_i \neq 0 \quad (i = 2,3,\cdots,n-1), a_n \neq 0;$

(2) $|b_1| > |c_1|, |b_i| > |a_i| + |c_i| (i = 2,\cdots,n-1), |b_n| > |a_n|.$

条件(1)表示方程组(1.13)不能降阶,事实上,若 $a_i = 0$,则方程组(1.13)可以分解为两个低阶的方程组;条件(2)保证了对这个三对角矩阵的三角分解可以进行到底.

下面,我们给出求解三对角方程组(1.13)的追赶法的计算过

程.

算法 1.7 求解三对角方程组的追赶法.

(1) 三角分解过程. 考虑三角分解

$$\boldsymbol{A} = \bar{\boldsymbol{L}} \cdot \bar{\boldsymbol{U}}. \tag{1.14}$$

即

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix}$$

$$= \begin{bmatrix} l_1 & & & & \\ m_2 & l_2 & & & \\ & \ddots & \ddots & & \\ & & m_{n-1} & l_{n-1} & \\ & & & m_n & l_n \end{bmatrix} \cdot \begin{bmatrix} 1 & u_1 & & & \\ & 1 & u_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & u_{n-1} \\ & & & & 1 \end{bmatrix}.$$

比较上式两边的对应元素,可得 m_i, l_i, u_i 的计算公式:

$$\begin{cases} l_1 = b_1, u_1 = \dfrac{c_1}{l_1}; \\ m_i = a_i, l_i = b_i - m_i u_{i-1}, u_i = \dfrac{c_i}{l_i}, (i = 2, 3, \cdots, n-1); \\ m_n = a_n, l_n = b_n - m_n u_{n-1}. \end{cases}$$

(2) 回代过程. 利用三角分解式(1.14)可以把方程组(1.13)化为如下的等价形式:

$$\begin{cases} \bar{\boldsymbol{L}}\boldsymbol{y} = \boldsymbol{d} \\ \bar{\boldsymbol{U}}\boldsymbol{x} = \boldsymbol{y}. \end{cases}$$

这两个方程组都是三角方程组,因此,可以递推计算:

$$\begin{cases} y_1 = \dfrac{d_1}{l_1} \\ y_i = \dfrac{1}{l_i}(d_i - m_i y_{i-1}), \quad (i = 2,3,\cdots,n) \end{cases}$$

$$\begin{cases} x_n = y_n \\ x_i = y_i - u_i x_{i+1}, \quad (i = n-1,\cdots,2,1) \end{cases}$$

例 1.7 用追赶法求解三对角线性代数方程组

$$\boldsymbol{Ax} = \boldsymbol{b},$$

其中

$$A = \begin{bmatrix} 2 & 2 & 0 & 0 & 0 \\ -1 & 1 & 2 & 0 & 0 \\ 0 & -1 & 1 & 2 & 0 \\ 0 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 6 \\ 7 \\ 9 \\ 11 \\ 1 \end{bmatrix}.$$

解 首先考虑追赶法的三角分解

$$\boldsymbol{A} = \begin{bmatrix} l_1 & 0 & 0 & 0 & 0 \\ m_2 & l_2 & 0 & 0 & 0 \\ 0 & m_3 & l_3 & 0 & 0 \\ 0 & 0 & m_4 & l_4 & 0 \\ 0 & 0 & 0 & m_5 & l_5 \end{bmatrix} \begin{bmatrix} 1 & u_1 & 0 & 0 & 0 \\ 0 & 1 & u_2 & 0 & 0 \\ 0 & 0 & 1 & u_3 & 0 \\ 0 & 0 & 0 & 1 & u_4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} =: \bar{\boldsymbol{L}} \cdot \bar{\boldsymbol{U}},$$

比较两边系数得

$$\begin{cases} & l_1 = 2, & u_1 = 1; \\ m_2 = -1, & l_2 = 2, & u_2 = 1; \\ m_3 = -1, & l_3 = 2, & u_3 = 1; \\ m_4 = -1, & l_4 = 2, & u_4 = 1; \\ m_5 = -1, & l_5 = 2. & \end{cases}$$

于是

$$\bar{\boldsymbol{L}}=\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \quad \bar{\boldsymbol{U}}=\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

令

$$\begin{cases} \bar{\boldsymbol{L}}\boldsymbol{y}=\boldsymbol{b}, \\ \bar{\boldsymbol{U}}\boldsymbol{x}=\boldsymbol{y}. \end{cases}$$

回代计算得到

$$\boldsymbol{y}=\begin{bmatrix} 3 \\ 5 \\ 7 \\ 9 \\ 5 \end{bmatrix}, \quad \boldsymbol{x}=\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}.$$

习 题 1

1. 分别用不带行交换的高斯消去法和列选主元高斯消去法，解线性代数方程组

$$\begin{bmatrix} 0.2641 & 0.1735 & 0.8642 \\ 0.9411 & -0.0175 & 0.1463 \\ -0.8641 & -0.4243 & 0.0711 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} -0.7521 \\ 0.6310 \\ 0.2501 \end{bmatrix}.$$

2. 用杜利脱尔分解计算线性代数方程组

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 3 & 1 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 14 \\ 18 \\ 20 \end{bmatrix}.$$

3. 用乔列斯基分解计算线性代数方程组

$$\begin{bmatrix} 4 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \\ 200 \\ 200 \\ 100 \end{bmatrix}.$$

4. 设 A 为 $n \times n$ 对称正定矩阵，试证明

(1) A 的对角线元素 $a_{ii} > 0 (i = 2,3,\cdots,n)$；

(2) 设 L 为非奇异矩阵，则 LAL^{T} 是对称正定矩阵；

(3) 经过不带行交换的高斯消去法消元一步后，A 约化为

$$\begin{bmatrix} a_{11} & \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{O} & \boldsymbol{A}_2 \end{bmatrix},$$

求证：A_2 为对称正定矩阵；

(4) $a_{ii}^{(2)} \leqslant a_{ii} (i = 2,3,\cdots,n)$.

5. 设 A 为 n 阶非奇异矩阵，且有分解式 $A = LU$，其中 L 为单位下三角矩阵，U 为上三角矩阵，求证：A 的所有顺序主子式均不为零.

6. 试证定理 1.3.

7. 证明：

(1) 初等下三角阵

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -l_{nk} & & 1 \end{bmatrix},$$

的逆矩阵为

$$\boldsymbol{L}_k^{-1}=\begin{bmatrix}1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & l_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & l_{n,k} & & 1\end{bmatrix},$$

即只是非对角元差一符号；

(2) $\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1}=\begin{bmatrix}1 & & & \\ l_{21} & 1 & & \\ & & \ddots & \\ \vdots & \ddots & & 1 \\ l_{n1} & \cdots & l_{n,n-1} & 1\end{bmatrix}$；

(3) 两个下三角阵之积仍为下三角阵.

8. 用追赶法求解三对角线性代数方程组

$$\begin{bmatrix}2 & 2 & 0 & 0\\ 1 & 3 & 2 & 0\\ 0 & 1 & 3 & 2\\ 0 & 0 & 1 & 3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}=\begin{bmatrix}0\\ 0\\ 0\\ -2\end{bmatrix}.$$

第二章　解线性代数方程组的迭代法

2.1　引　言

在许多实际问题中，常常需要求解这样的线性代数方程组，它的系数矩阵阶数很高，但非零元素很少，人们称其为大型稀疏线性代数方程组，对于这类方程组，如果它又不具有带状性，那么，再用直接法求解就不太有效，因为用直接法进行消元或矩阵的三角分解时，没有考虑到系数矩阵的稀疏性，破坏了系数矩阵的形状，导致了计算量的增加和存储单元的浪费，于是，人们常用迭代法求解大型稀疏线性代数方程组．迭代法只需要存储系数矩阵的非零元素，这样，占用内存单元较少，能解高阶线性代数方程组．由于迭代法是通过逐次迭代来逼近方程组的解，因此，收敛性和收敛速度是构造迭代法时要注意的问题．那么，是否可以构造一种适用于一般情况的迭代法呢？回答是否定的，这是因为不同的系数矩阵具有不同的性态，一般地，每一种迭代法都具有一定的适用范围，在本章的学习中将会看到，有时，某种方法对一类方程组迭代收敛，而对另一类方程组进行迭代时就会发散．因此，我们应该学会针对具有不同性质的线性代数方程组，构造合适的迭代方法．

本章主要介绍一些基本的迭代法，并在一定的范围内讨论其中几种方法的收敛性．

2.2　基本迭代法

考虑线性代数方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \quad\vdots \qquad\quad \vdots \qquad\quad\; \vdots \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \tag{2.1}$$

采用矩阵和向量记号,我们可以把(2.1)式写成

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{2.2}$$

其中,$\boldsymbol{A} \in R^{n\times n}$为非奇异矩阵,设 $a_{ii} \neq 0 (i = 1,2,\cdots,n)$.

下面我们介绍雅可比(Jacobi)迭代,高斯-塞德尔(Gauss-Seidel)迭代与 SOR 迭代以及 SSOR 迭代的基本思想和算法. 为了方便地给出矩阵表示式,我们引进下列矩阵分裂:

$$\boldsymbol{A} = -\boldsymbol{L} + \boldsymbol{D} - \boldsymbol{U}, \tag{2.3}$$

其中

$$\boldsymbol{L} = \begin{bmatrix} 0 & & & \\ -a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix},$$

$$\boldsymbol{U} = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ & 0 & \ddots & \vdots \\ & & \ddots & -a_{n-1,n} \\ & & & 0 \end{bmatrix}.$$

(1) 雅可比迭代的基本思想

从式(2.1)的第 i 个方程中解出 $x_i (i = 1,2,\cdots,n)$:

$$\begin{cases} x_1 = \dfrac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 \cdots - a_{1n}x_n) \\ \vdots \quad \vdots \qquad\qquad \vdots \\ x_i = \dfrac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_j\right) \\ \vdots \quad \vdots \qquad\qquad \vdots \\ x_n = \dfrac{1}{a_{nn}}(b_n - a_{n1}x_1 - a_{n2}x_2 \cdots - a_{n,n-1}x_{n-1}). \end{cases}$$

我们把迭代前的值代入上式右边，由计算得到等式左边的值作为一次迭代的新值，然后再把这个新值代入右边，再从左边得到一个新值，如此反复，就得到了雅可比迭代公式.

算法 2.1 雅可比迭代法.

选定初值 $x^{(0)} \in R^n$，对 $m = 1,2,\cdots$，计算

$$\begin{cases} x_1^{(m)} = \dfrac{1}{a_{11}}(b_1 - a_{12}x_2^{(m-1)} \cdots - a_{1n}x_n^{(m-1)}) \\ \vdots \quad \vdots \qquad\qquad \vdots \\ x_i^{(m)} = \dfrac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(m-1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(m-1)}\right) \\ \vdots \quad \vdots \qquad\qquad \vdots \\ x_n^{(m)} = \dfrac{1}{a_{nn}}(b_n - a_{n1}x_1^{(m-1)} \cdots - a_{n,n-1}x_{n-1}^{(m-1)}). \end{cases} \tag{2.4}$$

由式(2.4)及采用矩阵 A 的分裂记号式(2.3)，可以得到

$$Dx^{(m)} = b + Lx^{(m-1)} + Ux^{(m-1)},$$

于是得到雅可比迭代的矩阵表示形式：

$$x^{(m)} = D^{-1}(L + U)x^{(m-1)} + D^{-1}b. \tag{2.5}$$

(2) 高斯-塞德尔迭代的基本思想

在用雅可比迭代式(2.4)计算第 i 个新分量 $x_i^{(m)}$ 时，前 $i-1$ 个分量已经更新，与雅可比迭代法相比，高斯-塞德尔迭代是把更新过的分量 $x_j^{(m)}$ 代替式(2.4)第 i 个方程右端中的 $x_j^{(m-1)}(j = 1,$

$\cdots, i-1)$，于是，得到高斯-塞德尔迭代公式：

算法 2.2 高斯-塞德尔迭代法.

选定初值 $\boldsymbol{x}^{(0)} \in R^n$，对 $m = 1, 2, \cdots$，计算

$$\begin{cases} x_1^{(m)} = \dfrac{1}{a_{11}}(b_1 - a_{12}x_2^{(m-1)} \cdots - a_{1n}x_n^{(m-1)}) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ x_i^{(m)} = \dfrac{1}{a_{ii}}(b_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j^{(m)} - \sum\limits_{j=i+1}^{n} a_{ij}x_j^{(m-1)}) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ x_n^{(m)} = \dfrac{1}{a_{nn}}(b_n - a_{n1}x_1^{(m)} \cdots - a_{n,n-1}x_{n-1}^{(m)}). \end{cases} \tag{2.6}$$

由式(2.6)及采用矩阵 $\boldsymbol{A}$ 的分裂记号式(2.3)，可以得到

$$\boldsymbol{D}\boldsymbol{x}^{(m)} = (\boldsymbol{b} + \boldsymbol{L}\boldsymbol{x}^{(m)} + \boldsymbol{U}\boldsymbol{x}^{(m)}),$$

于是得到高斯-塞德尔迭代的矩阵表示形式：

$$\boldsymbol{x}^{(m)} = (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{U}\boldsymbol{x}^{(m-1)} + (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{b}. \tag{2.7}$$

(3) SOR 迭代的基本思想

在高斯-塞德尔迭代式(2.6)中，第 i 个迭代分量可以改写成

$$x_i^{(m)} = x_i^{(m-1)} + \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(m)} - \sum_{j=i}^{n} a_{ij}x_j^{(m-1)}).$$

上式等号右端的第二项可以看成是校正量，高斯-塞德尔迭代法相比，SOR 迭代是把这个校正量乘上一个因子 ω，于是得到 SOR 迭代公式.

算法 2.3 SOR 迭代法.

选定初值 $\boldsymbol{x}^{(0)}$，对 $m = 1, 2, \cdots$，计算

$$x_i^{(m)} = x_i^{(m-1)} + \frac{\omega}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(m)} - \sum_{j=i}^{n} a_{ij}x_j^{(m-1)}),$$

$$(i = 1, 2, \cdots, n).$$

由上式及采用矩阵 $\boldsymbol{A}$ 的分裂记号式(2.3)，可以得到

$$\boldsymbol{x}^{(m)} = \boldsymbol{x}^{(m-1)} + \omega\boldsymbol{D}^{-1}(\boldsymbol{b} + \boldsymbol{L}\boldsymbol{x}^{(m)} + \boldsymbol{U}\boldsymbol{x}^{(m-1)} - \boldsymbol{D}\boldsymbol{x}^{(m-1)}),$$

于是得到 SOR 迭代的矩阵表示形式：

$$\boldsymbol{x}^{(m)} = (\boldsymbol{D} - \omega\boldsymbol{L})^{-1}\{(1-\omega)\boldsymbol{D} + \omega\boldsymbol{U}\}\boldsymbol{x}^{(m-1)} + \omega(\boldsymbol{D} - \omega\boldsymbol{L})^{-1}\boldsymbol{b}. \tag{2.8}$$

称 ω 为松弛因子.

(4) SSOR 迭代的基本思想

在 SOR 迭代过程中，新向量的分量计算依次从第 1 到第 n 逐个进行，这个次序也可以倒过来，即

$$x_i^{(m)} = x_i^{(m-1)} + \frac{\omega}{a_{ii}}\Big(b_i - \sum_{j=1}^{i} a_{ij}x_j^{(m-1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(m)}\Big),$$
$$(i = n, n-1, \cdots, 2, 1).$$

如果两种次序的 SOR 迭代过程交替使用，就可以得到 SSOR 迭代公式.

算法 2.4 SSOR 迭代法.

选定初值 $\boldsymbol{x}^{(0)}$，对 $m = 1, 2, \cdots$，计算

$$\begin{cases} x_i^{\left(m-\frac{1}{2}\right)} = x_i^{(m-1)} + \dfrac{\omega}{a_{ii}}\Big(b_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j^{\left(m-\frac{1}{2}\right)} - \sum\limits_{j=i}^{n} a_{ij}x_j^{(m-1)}\Big) \\ \qquad (i = 1, 2, \cdots, n-1, n), \\ x_i^{(m)} = x_i^{\left(m-\frac{1}{2}\right)} + \dfrac{\omega}{a_{ii}}\Big(b_i - \sum\limits_{j=1}^{i} a_{ij}x_j^{\left(m-\frac{1}{2}\right)} - \sum\limits_{j=i+1}^{n} a_{ij}x_j^{(m)}\Big) \\ \qquad (i = n, n-1, \cdots, 2, 1). \end{cases}$$

由上式及采用矩阵 $\boldsymbol{A}$ 的分裂记号式(2.3)，可以得到

$$\begin{cases} \boldsymbol{x}^{\left(m-\frac{1}{2}\right)} = \boldsymbol{x}^{(m-1)} + \omega\boldsymbol{D}^{-1}\big[\boldsymbol{b} + \boldsymbol{L}\boldsymbol{x}^{\left(m-\frac{1}{2}\right)} + (\boldsymbol{U} - \boldsymbol{D})\boldsymbol{x}^{(m-1)}\big] \\ \boldsymbol{x}^{(m)} = \boldsymbol{x}^{\left(m-\frac{1}{2}\right)} + \omega\boldsymbol{D}^{-1}\big[\boldsymbol{b} + (\boldsymbol{L} - \boldsymbol{D})\boldsymbol{x}^{\left(m-\frac{1}{2}\right)} + \boldsymbol{U}\boldsymbol{x}^{(m)}\big] \end{cases}$$

从上面的第一式解出 $\boldsymbol{x}^{\left(m-\frac{1}{2}\right)}$ 后，再化简可得

$$\begin{aligned} \boldsymbol{x}^{(m)} = &\left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{U}\right)^{-1}\boldsymbol{D}\left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{L}\right)^{-1} \\ &\cdot\left(\frac{\omega-1}{\omega}\boldsymbol{D} - \boldsymbol{L}\right)\boldsymbol{D}^{-1}\left(\frac{\omega-1}{\omega}\boldsymbol{D} - \boldsymbol{U}\right)\boldsymbol{x}^{(m-1)} \\ &+ \omega(2-\omega)(\boldsymbol{I} - \omega\boldsymbol{D}^{-1}\boldsymbol{L})^{-1}(\boldsymbol{I} - \omega\boldsymbol{D}^{-1}\boldsymbol{U})^{-1}\boldsymbol{D}^{-1}\boldsymbol{b}. \end{aligned}$$

例 2.1 试用雅可比迭代法、高斯-塞德尔迭代法、SOR 迭代法以及 SSOR 迭代法计算线代数方程组

$$\begin{bmatrix} -4 & 1 & 1 & 1 \\ 1 & -4 & 1 & 1 \\ 1 & 1 & -4 & 1 \\ 1 & 1 & 1 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

取初值为 $x^{(0)} = 0$, 精确到 10^{-5}(精确解 $x^* = (-1, -1, -1, -1)^T$).

解 构造雅可比迭代:对 $k = 1, 2, \cdots$, 计算

$$\begin{cases} x_1^{(m)} = -\frac{1}{4}(1 - x_2^{(m-1)} - x_3^{(m-1)} - x_4^{(m-1)}) \\ x_2^{(m)} = -\frac{1}{4}(1 - x_1^{(m-1)} - x_3^{(m-1)} - x_4^{(m-1)}) \\ x_3^{(m)} = -\frac{1}{4}(1 - x_1^{(m-1)} - x_2^{(m-1)} - x_4^{(m-1)}) \\ x_4^{(m)} = -\frac{1}{4}(1 - x_1^{(m-1)} - x_2^{(m-1)} - x_3^{(m-1)}). \end{cases}$$

迭代到第 43 次结果为

$$x^{(43)} = (-1.00000, -1.00000, -1.00000, -1.00000)^T.$$

构造高斯-塞德尔迭代:对 $k = 1, 2, \cdots$, 计算

$$\begin{cases} x_1^{(m)} = -\frac{1}{4}(1 - x_2^{(m-1)} - x_3^{(m-1)} - x_4^{(m-1)}) \\ x_2^{(m)} = -\frac{1}{4}(1 - x_1^{(m)} - x_3^{(m-1)} - x_4^{(m-1)}) \\ x_3^{(m)} = -\frac{1}{4}(1 - x_1^{(m)} - x_2^{(m)} - x_4^{(m-1)}) \\ x_4^{(m)} = -\frac{1}{4}(1 - x_1^{(m)} - x_2^{(m)} - x_3^{(m)}). \end{cases}$$

迭代到第 22 次结果为

$$x^{(22)} = (-0.99999, -0.99999, -1.00000, -1.00000)^T.$$

构造 SOR 迭代:对 $k=1,2,\cdots$,计算

$$\begin{cases} x_1^{(m)} = x_1^{(m-1)} - \dfrac{\omega}{4}(1+4x_1^{(m-1)} - x_2^{(m-1)} - x_3^{(m-1)} - x_4^{(m-1)}) \\ x_2^{(m)} = x_2^{(m-1)} - \dfrac{\omega}{4}(1 - x_1^{(m)} + 4x_2^{(m-1)} - x_3^{(m-1)} - x_4^{(m-1)}) \\ x_3^{(m)} = x_3^{(m-1)} - \dfrac{\omega}{4}(1 - x_1^{(m)} - x_2^{(m)} + 4x_3^{(m-1)} - x_4^{(m-1)}) \\ x_4^{(m)} = x_4^{(m-1)} - \dfrac{\omega}{4}(1 - x_1^{(m)} - x_2^{(m)} - x_3^{(m)} + 4x_4^{(m-1)}). \end{cases}$$

当 ω 取不同值时的迭代结果为

$\omega=1.0, x^{(21)}=(-0.99999,-0.99999,-1.00000,-1.00000)^T$;

$\omega=1.1, x^{(16)}=(-0.99999,-1.00000,-1.00000,-1.00000)^T$;

$\omega=1.2, x^{(11)}=(-1.00000,-1.00000,-1.00000,-1.00000)^T$;

$\omega=1.3, x^{(10)}=(-1.00000,-1.00000,-1.00000,-1.00000)^T$;

$\omega=1.4, x^{(13)}=(-1.00000,-1.00000,-1.00000,-1.00000)^T$;

$\omega=1.5, x^{(16)}=(-0.99999,-1.00000,-1.00000,-1.00000)^T$;

$\omega=1.6, x^{(22)}=(-1.00000,-1.00000,-1.00000,-1.00000)^T$;

$\omega=1.7, x^{(32)}=(-1.00000,-0.99999,-1.00000,-1.00000)^T$;

$\omega=1.8, x^{(52)}=(-0.99999,-1.00000,-1.00000,-0.99999)^T$;

$\omega=1.9, x^{(108)}=(-1.00000,-1.00000,-0.99999,-1.00000)^T$.

构造 SSOR 迭代:对 $k=1,2,\cdots$,计算

$$\begin{cases} x_1^{\left(m-\frac{1}{2}\right)} = x_1^{(m-1)} - \dfrac{\omega}{4}(1+4x_1^{(m-1)} - x_2^{(m-1)} - x_3^{(m-1)} - x_4^{(m-1)}) \\ x_2^{\left(m-\frac{1}{2}\right)} = x_2^{(m-1)} - \dfrac{\omega}{4}(1 - x_1^{\left(m-\frac{1}{2}\right)} + 4x_2^{(m-1)} - x_3^{(m-1)} - x_4^{(m-1)}) \\ x_3^{\left(m-\frac{1}{2}\right)} = x_3^{(m-1)} - \dfrac{\omega}{4}(1 - x_1^{\left(m-\frac{1}{2}\right)} - x_2^{\left(m-\frac{1}{2}\right)} + 4x_3^{(m-1)} - x_4^{(m-1)}) \\ x_4^{\left(m-\frac{1}{2}\right)} = x_4^{(m-1)} - \dfrac{\omega}{4}(1 - x_1^{\left(m-\frac{1}{2}\right)} - x_2^{\left(m-\frac{1}{2}\right)} - x_3^{\left(m-\frac{1}{2}\right)} + 4x_4^{(m-1)}). \end{cases}$$

$$\begin{cases} x_4^{(m)} = x_4^{\left(m-\frac{1}{2}\right)} - \dfrac{\omega}{4}\left(1 - x_1^{\left(m-\frac{1}{2}\right)} - x_2^{\left(m-\frac{1}{2}\right)} - x_3^{\left(m-\frac{1}{2}\right)} + 4x_4^{\left(m-\frac{1}{2}\right)}\right) \\ x_3^{(m)} = x_3^{\left(m-\frac{1}{2}\right)} - \dfrac{\omega}{4}\left(1 - x_1^{\left(m-\frac{1}{2}\right)} - x_2^{\left(m-\frac{1}{2}\right)} + 4x_3^{\left(m-\frac{1}{2}\right)} - x_4^{(m)}\right) \\ x_2^{(m)} = x_2^{\left(m-\frac{1}{2}\right)} - \dfrac{\omega}{4}\left(1 - x_1^{\left(m-\frac{1}{2}\right)} + 4x_2^{\left(m-\frac{1}{2}\right)} - x_3^{(m)} - x_4^{(m)}\right) \\ x_1^{(m)} = x_1^{\left(m-\frac{1}{2}\right)} - \dfrac{\omega}{4}\left(1 + 4x_1^{\left(m-\frac{1}{2}\right)} - x_2^{(m)} - x_3^{(m)} - x_4^{(m)}\right). \end{cases}$$

当 ω 取不同值时的迭代结果为

$\omega = 1.0, x^{(16)} = (-1.00000, -1.00000, -1.00000, -0.99999)^{\mathrm{T}}$;

$\omega = 1.1, x^{(16)} = (-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$;

$\omega = 1.2, x^{(16)} = (-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$;

$\omega = 1.3, x^{(17)} = (-1.00000, -1.00000, -1.00000, -0.99999)^{\mathrm{T}}$;

$\omega = 1.4, x^{(20)} = (-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$;

$\omega = 1.5, x^{(24)} = (-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$;

$\omega = 1.6, x^{(31)} = (-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$;

$\omega = 1.7, x^{(42)} = (-1.00000, -1.00000, -0.99999, -0.99999)^{\mathrm{T}}$;

$\omega = 1.8, x^{(66)} = (-1.00000, -1.00000, -0.99999, -0.99999)^{\mathrm{T}}$;

$\omega = 1.9, x^{(139)} = (-1.00000, -1.00000, -0.99999, -0.99999)^{\mathrm{T}}$.

2.3 范数及方程组的性态、条件数

由于迭代法是通过逐次迭代来逼近精确解的，于是近似解向量与精确解向量之差趋于零的收敛速度就是迭代法最为关心的问题，为了讨论迭代法的收敛性，我们需要引进衡量向量和矩阵大小的度量概念——向量范数和矩阵范数.

首先给出向量范数的定义：

定义 2.1 对任意的向量 $x \in R^n$，若对应一个非负实值函数 $\|x\|$，且满足：

(1) 正定性：$\|x\| \geqslant 0$，等号当且仅当 $x = 0$ 时成立；

(2) 齐次性:对任意实数 α, $\| \alpha x \| = |\alpha| \cdot \| x \|$;

(3) 三角不等式: $\| x + y \| \leqslant \| x \| + \| y \|$, $\forall x, y \in R^n$,

则称 $\| x \|$ 为向量 x 的**范数或模**.

在下面的例子中,给出了三个常用的向量范数:

例 2.2 设 $x = (x_1, x_2, \cdots, x_n)^T$,试证

$$\| x \|_2 := \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2},$$

$$\| x \|_1 := |x_1| + |x_2| + \cdots + |x_n|,$$

$$\| x \|_\infty := \max_{1 \leqslant i \leqslant n} |x_i|,$$

满足向量范数定义中的三个条件,它们分别称为向量 2 范数、1 范数和无穷范数.

证明 我们仅对向量 2 范数进行证明,向量 1 范数和向量无穷范数的证明由读者自己完成.

显然, $\| \cdot \|_2$ 满足定义 2.1 中条件(1)、(2),现在证明条件(3). 对任意的 $x, y \in R^n$,由柯西(Cauchy)不等式得

$$\begin{aligned} \| x + y \|_2^2 &= \sum_{i=1}^n (x_i + y_i)^2 \\ &= \sum_{i=1}^n x_i^2 + \sum_{i=1}^n y_i^2 + 2\sum_{i=1}^n x_i y_i \\ &\leqslant \sum_{i=1}^n x_i^2 + \sum_{i=1}^n y_i^2 + 2\sqrt{\sum_{i=1}^n x_i^2} \cdot \sqrt{\sum_{i=1}^n y_i^2} \\ &= (\| x \|_2 + \| y \|_2)^2, \end{aligned}$$

即条件(3)成立,于是 $\| \cdot \|_2$ 是 R^n 中的向量范数.

现在给出矩阵范数的定义:

定义 2.2 对任意的 $A \in R^{n \times n}$,若对应一个非负实值函数 $\| A \|$,且满足:

(1) 正定性: $\| A \| \geqslant 0$,等号当且仅当 $A = 0$ 时成立;

(2) 齐次性:对任意实数 α, $\| \alpha A \| = |\alpha| \cdot \| A \|$;

(3) 三角不等式: $\| A + B \| \leqslant \| A \| + \| B \|$, $\forall A, B \in R^{n \times n}$;

(4) $\| AB \| \leqslant \| A \| \cdot \| B \|, \forall A, B \in R^{n\times n}$,

则称 $\| A \|$ 为矩阵 A 的范数.

进一步,若对给定的矩阵范数 $\| \cdot \|_M$,它与某个向量范数 $\| \cdot \|_V$满足条件:

(5) $\| Ax \|_V \leqslant \| A \|_M \cdot \| x \|_V, \forall x \in R^n$,

则称矩阵范数 $\| \cdot \|_M$ 与向量范数 $\| \cdot \|_V$ 相容.

对 $A = \{a_{ij}\}_{n\times n} \in R^{n\times n}$,常用的矩阵范数有

$$\| A \|_1 := \max_{1\leqslant j\leqslant n} \sum_{i=1}^{n} |a_{ij}|,$$

$$\| A \|_\infty := \max_{1\leqslant i\leqslant n} \sum_{j=1}^{n} |a_{ij}|,$$

$$\| A \|_2 := \sqrt{\text{矩阵 } A^{\mathrm{T}}A \text{ 的最大特征值}},$$

$$\| A \|_F := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2},$$

它们分别称为矩阵 1 范数、无穷范数、2 范数和 F-范数. 可以证明它们都满足定义 2.2 矩阵范数的四个条件,并且它们分别与向量范数有下列的相容关系:

$$\| Ax \|_1 \leqslant \| A \|_1 \| x \|_1,$$

$$\| Ax \|_\infty \leqslant \| A \|_\infty \| x \|_\infty,$$

$$\| Ax \|_2 \leqslant \| A \|_2 \| x \|_2,$$

$$\| Ax \|_2 \leqslant \| A \|_F \| x \|_2.$$

在用计算机求解线性代数方程组时,常会发现这样的情况,当用同一方法求解不同的线性代数方程组时,有时产生不同的效果,这主要涉及到所解的线性代数方程组的性态. 下面,我们给出线性代数方程组性态"好"、"坏"的概念,并给出一种衡量标准.

考虑线性代数方程组

$$\begin{cases} 12x_1 + 35x_2 = 59 \\ 12x_1 + 35.000001x_2 = 59.000001, \end{cases} \tag{2.9}$$

由简单计算知方程组(2.9)的解为 $x_1 = 2, x_2 = 1$. 当式(2.9)的第

二个方程有一个小扰动时，例如

$$\begin{cases} 12x_1 + 35x_2 = 59 \\ 12x_1 + 34.999999x_2 = 59.000002, \end{cases} \tag{2.10}$$

则方程组(2.10)的解为 $x_1 = 10.75, x_2 = -2$.

比较方程组(2.9)与方程组(2.10)的解可以看出，由于系数矩阵和右端的小扰动，导致它们的解完全不同. 下面我们研究一般的线性代数方程组

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{2.11}$$

的系数矩阵 $\boldsymbol{A} \in R^{n \times n}$ 和右端 $\boldsymbol{b} \in R^n$ 有小扰动时，对方程组(2.11)的解 $\boldsymbol{x}$ 所产生的影响.

ⅰ) 如果(2.11)式中 $\boldsymbol{b}$ 有一小扰动 $\delta \boldsymbol{b}$，则解 $\boldsymbol{x}$ 产生一个扰动 $\delta \boldsymbol{x}$，即

$$\boldsymbol{A}(\boldsymbol{x} + \delta \boldsymbol{x}) = \boldsymbol{b} + \delta \boldsymbol{b}.$$

于是

$$\delta \boldsymbol{x} = \boldsymbol{A}^{-1} \cdot \delta \boldsymbol{b},$$

从而

$$\| \delta \boldsymbol{x} \| \leqslant \| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{b} \|, \tag{2.12}$$

另一方面，由式(2.11)知

$$\| \boldsymbol{b} \| \leqslant \| \boldsymbol{A} \| \cdot \| \boldsymbol{x} \|. \tag{2.13}$$

由式(2.13)和式(2.12)即得

$$\frac{\| \delta \boldsymbol{x} \|}{\| \boldsymbol{x} \|} \leqslant \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \| \cdot \frac{\| \delta \boldsymbol{b} \|}{\| \boldsymbol{b} \|}. \tag{2.14}$$

ⅱ) 如果式(2.11)中 $\boldsymbol{A}$ 有一小扰动 $\delta \boldsymbol{A}$，则解 $\boldsymbol{x}$ 产生一个扰动 $\delta \boldsymbol{x}$，即

$$(\boldsymbol{A} + \delta \boldsymbol{A})(\boldsymbol{x} + \delta \boldsymbol{x}) = \boldsymbol{b}.$$

于是

$$\delta \boldsymbol{x} = -\boldsymbol{A}^{-1} \cdot \delta \boldsymbol{A} \cdot (\boldsymbol{x} + \delta \boldsymbol{x}),$$

从而

$$\| \delta \boldsymbol{x} \| \leqslant \| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{A} \| \cdot \| \boldsymbol{x} + \delta \boldsymbol{x} \|.$$

所以

$$\frac{\| \delta \boldsymbol{x} \|}{\| \boldsymbol{x} + \delta \boldsymbol{x} \|} \leqslant \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \| \cdot \frac{\| \delta \boldsymbol{A} \|}{\| \boldsymbol{A} \|}. \tag{2.15}$$

ⅲ）如果式(2.11)中 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 都有小扰动时，则 $\boldsymbol{x}$ 的扰动满足：

$$(\boldsymbol{A} + \delta \boldsymbol{A})(\boldsymbol{x} + \delta \boldsymbol{x}) = \boldsymbol{b} + \delta \boldsymbol{b}.$$

于是

$$(\boldsymbol{A} + \delta \boldsymbol{A}) \cdot \delta \boldsymbol{x} = \delta \boldsymbol{b} - \delta \boldsymbol{A} \cdot \boldsymbol{x}.$$

即

$$\delta \boldsymbol{x} + \boldsymbol{A}^{-1} \cdot \delta \boldsymbol{A} \cdot \delta \boldsymbol{x} = \boldsymbol{A}^{-1} \cdot \delta \boldsymbol{b} - \boldsymbol{A}^{-1} \cdot \delta \boldsymbol{A} \cdot \boldsymbol{x}.$$

从而

$$\| \delta \boldsymbol{x} \| - \| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{A} \| \cdot \| \delta \boldsymbol{x} \|$$
$$\leqslant \| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{b} \| + \| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{A} \| \cdot \| \boldsymbol{x} \|.$$

两边除以 $\| \boldsymbol{x} \|$，并考虑式(2.13)得到

$$(1 - \| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{A} \|) \cdot \frac{\| \delta \boldsymbol{x} \|}{\| \boldsymbol{x} \|}$$
$$\leqslant \frac{\| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{b} \|}{\| \boldsymbol{x} \|} + \| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{A} \|$$
$$= \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \| \left(\frac{\| \delta \boldsymbol{b} \|}{\| \boldsymbol{A} \| \cdot \| \boldsymbol{x} \|} + \frac{\| \delta \boldsymbol{A} \|}{\| \boldsymbol{A} \|} \right)$$
$$\leqslant \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \| \left(\frac{\| \delta \boldsymbol{b} \|}{\| \boldsymbol{b} \|} + \frac{\| \delta \boldsymbol{A} \|}{\| \boldsymbol{A} \|} \right).$$

当 $\| \boldsymbol{A}^{-1} \| \cdot \| \delta \boldsymbol{A} \| < 1$ 时，由上式可以推出

$$\frac{\| \delta \boldsymbol{x} \|}{\| \boldsymbol{x} \|} \leqslant \frac{\| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \|}{1 - \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \| \cdot \frac{\| \delta \boldsymbol{A} \|}{\| \boldsymbol{A} \|}} \left(\frac{\| \delta \boldsymbol{b} \|}{\| \boldsymbol{b} \|} + \frac{\| \delta \boldsymbol{A} \|}{\| \boldsymbol{A} \|} \right). \tag{2.16}$$

观察式(2.14)、式(2.15)和式(2.16)可以知道，无论方程组(2.11)中的系数矩阵 $\boldsymbol{A}$ 有扰动，还是右端 $\boldsymbol{b}$ 有扰动，或者两者都有扰动，解 $\boldsymbol{x}$ 的相对误差除了受相应扰动的相对误差的影响外，还与 $\| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \|$ 的大小有关，因此，研究 $\| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \|$ 的值对估计解的相对误差有着重要的意义.

定义 2.3 设 $A\in R^{n\times n}$ 为非奇异矩阵，称数

$$\mathrm{Cond}(A)_v = \|A\|_v \|A^{-1}\|_v$$

为矩阵 A 的条件数，其中 $\|\cdot\|_v$ 为 $R^{n\times n}$ 中的某种矩阵范数.

常用的条件数有

$$\mathrm{Cond}(A)_1 = \|A\|_1 \|A^{-1}\|_1,$$

$$\mathrm{Cond}(A)_\infty = \|A\|_\infty \|A^{-1}\|_\infty,$$

$$\mathrm{Cond}(A)_2 = \|A\|_2 \|A^{-1}\|_2 = \sqrt{\frac{\lambda_{\max}(A^{\mathrm{T}}A)}{\lambda_{\min}(A^{\mathrm{T}}A)}}.$$

如果 A 对称正定，则

$$\mathrm{Cond}(A)_2 = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

条件数有下列性质：

(1) 对任意的非奇异矩阵 $A\in R^{n\times n}$，有 $\mathrm{Cond}(A)_v \geqslant 1$，其中 $\|\cdot\|_v$ 为 A 的某种矩阵范数.

(2) 对任意的非奇异矩阵 $A\in R^{n\times n}$，$c\neq 0$ 为常数，则

$$\mathrm{Cond}(cA)_v = \mathrm{Cond}(A)_v.$$

(3) 对任意的正交矩阵 A，则 $\mathrm{Cond}(A)_2 = 1$.

(4) 设 A 为非奇异矩阵，P 为正交矩阵，则

$$\mathrm{Cond}(PA)_2 = \mathrm{Cond}(AP)_2 = \mathrm{Cond}(A)_2.$$

矩阵的条件数是方程组 $Ax = b$ 的解 x 对系数矩阵 A、右端 b 中数据有微小扰动时敏感性的一种度量，或者说是对方程组 $Ax = b$ 是否病态的一种度量.

定义 2.4 设 $A\in R^{n\times n}$ 为非奇异矩阵，对线性代数方程组

$$Ax = b, \tag{2.17}$$

(1) 如果条件数 $\mathrm{Cond}(A)_v$ 很大时（$\mathrm{Cond}(A)\gg 1$），则称式(2.17)为**病态方程组**(或 A **为病态的**)；

(2) 如果条件数 $\mathrm{Cond}(A)_v$ 相对较小时，则称式(2.17)为**良态方程组**(或 A **为良态的**).

n 阶希尔伯特(Hilbert)矩阵

$$\boldsymbol{H}_n=\begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix}$$

是常见的病态矩阵. 它的条件数有下列结果:

$$\mathrm{Cond}(\boldsymbol{H}_3)_\infty = 748,$$

$$\mathrm{Cond}(\boldsymbol{H}_5)_\infty = 4.77\times 10^5,$$

$$\mathrm{Cond}(\boldsymbol{H}_7)_\infty = 4.75\times 10^8.$$

2.4 收敛性分析

在 2.1 中给出的三种迭代公式,雅可比迭代、高斯-塞德尔迭代和 SOR 迭代,都可以看成是把原方程式(2.1)改写成某种等价形式:

$$\boldsymbol{x}=\boldsymbol{B}\boldsymbol{x}+\boldsymbol{f},$$

再选定一个初值 $\boldsymbol{x}^{(0)}$,进行迭代

$$\boldsymbol{x}^{(m)}=\boldsymbol{B}\boldsymbol{x}^{(m-1)}+\boldsymbol{f} \qquad (m=1,2,\cdots).$$

参照式(2.3)的矩阵分裂,则在雅可比迭代、高斯-塞德尔迭代、SOR 迭代和 SSOR 迭代时的 $\boldsymbol{B}$ 和 $\boldsymbol{f}$ 分别为

$$\begin{cases} \boldsymbol{B}_J=\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U}), \\ \boldsymbol{f}_J=\boldsymbol{D}^{-1}\boldsymbol{b}, \end{cases}$$

$$\begin{cases} \boldsymbol{B}_G=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}, \\ \boldsymbol{f}_G=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{b}, \end{cases}$$

$$\begin{cases} \boldsymbol{B}_S = (\boldsymbol{D} - \omega\boldsymbol{L})^{-1}\{(1-\omega)\boldsymbol{D} + \omega\boldsymbol{U}\}, \\ \boldsymbol{f}_S = \omega(\boldsymbol{D} - \omega\boldsymbol{L})^{-1}\boldsymbol{b}, \end{cases}$$

$$\begin{cases} \boldsymbol{B}_{SS} = \left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{U}\right)^{-1}\boldsymbol{D}\left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{L}\right)^{-1} \\ \qquad \cdot\left(\frac{\omega-1}{\omega}\boldsymbol{D} - \boldsymbol{L}\right)\boldsymbol{D}^{-1}\left(\frac{\omega-1}{\omega}\boldsymbol{D} - \boldsymbol{U}\right), \\ \boldsymbol{f}_{ss} = \omega(2-\omega)(\boldsymbol{I} - \omega\boldsymbol{D}^{-1}\boldsymbol{L})^{-1}(\boldsymbol{I} - \omega\boldsymbol{D}^{-1}\boldsymbol{U})^{-1}\boldsymbol{D}^{-1}\boldsymbol{b}. \end{cases}$$

为了分析迭代法的收敛性，我们需要引进一些特殊矩阵，并讨论它们的某些性质.

定义 2.5 如果矩阵 $\boldsymbol{A} = \{a_{ij}\}_{n\times n} \in R^{n\times n}$ 中的元素满足

(1) $$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \qquad (i = 1,2,\cdots,n),$$

$$(\text{或} \quad |a_{jj}| > \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}| \qquad (j = 1,2,\cdots,n)),$$

则称 $\boldsymbol{A}$ 为按行(或按列)**严格对角占优**；

(2) $$|a_{ii}| \geqslant \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \qquad (i = 1,2,\cdots,n),$$

$$(\text{或} \quad |a_{jj}| \geqslant \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}| \qquad (j = 1,2,\cdots,n)),$$

且上式至少有一个不等式是严格成立，则称 $\boldsymbol{A}$ 为按行(或按列)**弱对角占优**.

定义 2.6 设矩阵 $\boldsymbol{A} = \{a_{ij}\}_{n\times n} \in R^{n\times n}(n \geqslant 2)$，如果存在置换阵 $\boldsymbol{P}$ 使得

$$\boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ 0 & \boldsymbol{A}_{22} \end{bmatrix}, \tag{2.18}$$

其中 $\boldsymbol{A}_{11}$ 为 r 阶方阵，$\boldsymbol{A}_{22}$ 为 $n-r$ 阶方阵$(1 \leqslant r \leqslant n)$，则称 $\boldsymbol{A}$ 为可约矩阵；否则，如果不存在使得式(2.18)成立的置换阵 $\boldsymbol{P}$，则称 $\boldsymbol{A}$ 为不可约矩阵.

引理 2.1 如果 $\boldsymbol{A} \in R^{n\times n}$ 为按行(或按列)严格对角占优矩

阵,或者为按行(或按列)弱对角占优且为不可约矩阵,则 A 为非奇异矩阵.

证明 仅对第一个结论中的按行情况证明,其他情况的证明留作习题.

假定按行严格对角占优阵 A 为奇异矩阵,则存在 $0 \neq z \in R^n$ 满足

$$Az = 0.$$

于是存在指标 i_0,使得 $|z_{i_0}| = \max_{1 \leqslant i \leqslant n} |z_i| > 0$,并且

$$\sum_{j=0}^{n} a_{i_0 j} z_j = 0.$$

从而 z 的第 i_0 个分量满足

$$|a_{i_0 i_0}| \leqslant \sum_{\substack{j=1 \\ j \neq i_0}}^{n} |a_{i_0 j} z_j| \Big/ |z_{i_0}| \sum_{\substack{j=1 \\ j \neq i_0}}^{n} |a_{ij}|.$$

这与 A 为按行严格对角占优矛盾,因此 A 为奇异矩阵.

引理 2.2 矩阵 $A \in R^{n \times n}$,则下列三个条件等价:

(1) 对任意的 $z \in R^n$ 成立: $\lim_{m \to \infty} A^m z = 0$;

(2) $\lim_{m \to \infty} A^m = 0$;

(3) A 的谱半径 $\rho(A) < 1$.

证明 (留作习题).

引理 2.3 如果矩阵 $A \in R^{n \times n}$ 的谱半径 $\rho(A) < 1$,则 $(I - A)$ 为非奇异矩阵.

证明 若 $(I - A)$ 为奇异矩阵,则

$$\det(I - A) = 0,$$

于是矩阵 A 有一个特征值为 1,与 $\rho(A) < 1$ 矛盾,所以 $(I - A)$ 为非奇异矩阵.

下面我们研究一般迭代法的基本收敛结果.

定理 2.1 给定迭代格式

$$x^{(m)} = Bx^{(m-1)} + f, \tag{2.19}$$

则对任意的 $x^{(0)} \in R^n$,有下列的收敛性结果:

(1) 迭代格式(2.19)收敛的充要条件为谱半径 $\rho(B) < 1$,

(2) 迭代格式(2.19)收敛的充分条件为范数 $\| B \| < 1$.

证明 (1) 若迭代格式(2.19)收敛,不妨设收敛于 $x^* \in R^n$,于是

$$x^* = Bx^* + f.$$

把上式与式(2.19)相减得到

$$x^* - x^{(m)} = B(x^* - x^{(m-1)}) = \cdots = B^m(x^* - x^{(0)}).$$

在上式中令 $m \to \infty$,于是有

$$0 = \lim_{m \to \infty}(x^* - x^{(m)}) = \lim_{m \to \infty} B^m(x^* - x^{(m)}).$$

再由引理 2.2 及$(x^* - x^{(0)})$的任意性即得 $\rho(B) < 1$.

反之,若 $\rho(B) < 1$,则由引理 2.3 知$(I - B)$为非奇异矩阵,从而$(I - B)^{-1} f$ 存在,不妨设为 x',即 $x' = (I - B)^{-1} f$,于是

$$x' = Bx' + f.$$

把上式与式(2.19)相减得到

$$x^{(m)} - x' = B(x^{(m-1)} - x') = \cdots = B^m(x^{(0)} - x').$$

由假设条件 $\rho(B) < 1$ 及引理 2.2 可以推出

$$\lim_{m \to \infty}(x^{(m)} - x') = \lim_{m \to \infty} B^m(x^{(0)} - x') = 0,$$

即迭代格式(2.19)收敛.

(2) 设 λ 为 B 的任一特征值,$0 \neq z \in R^n$ 为其对应的特征向量,即

$$Bz = \lambda z.$$

于是

$$|\lambda| = \frac{\| Bz \|}{\| z \|} \leqslant \sup_{y \in R^n} \frac{\| By \|}{\| y \|} = \| B \| < 1.$$

从而 $\rho(B) < 1$,由(1)知迭代格式(2.19)收敛.

定理 2.2 用雅可比迭代或高斯-塞德尔迭代计算线性代数方程组 $Ax = b$,有下列的收敛性结果:

(1) 若 A 为按行(或按列)严格对角占优或者 A 为按行(或按

列)弱对角占优且为不可约矩阵,则雅可比迭代和高斯-塞德尔迭代收敛;

(2) 若 $\boldsymbol{A}$ 对称正定,则高斯-塞德尔迭代收敛;

(3) 若 $\boldsymbol{A}$ 对称正定,则雅可比迭代收敛的充要条件为矩阵 $2\boldsymbol{D}-\boldsymbol{A}$ 也对称正定.

证明 (1) 我们仅对 $\boldsymbol{A}$ 为按行严格对角占优或者 $\boldsymbol{A}$ 为按行弱对角占优且为不可约矩阵时,证明雅可比迭代和高斯-塞德尔迭代的收敛性,按列情况的收敛性证明可完全类似地进行.

先证高斯-塞德尔迭代的收敛性. 用反证法,若不收敛,那么,由定理 2.1 中的结论(1)知高斯-塞德尔迭代法的迭代矩阵的谱半径 $\rho(\boldsymbol{B}_G)\geqslant 1$,参考矩阵分裂式(2.3),于是存在 $\boldsymbol{B}_G$ 的某一特征值 λ,$|\lambda|\geqslant 1$,满足

$$\begin{aligned}0 &= \det(\lambda\boldsymbol{I}-\boldsymbol{B}_G)\\ &= \det[\lambda\boldsymbol{I}-(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}]\\ &= \det[(\boldsymbol{D}-\boldsymbol{L})^{-1}]\cdot\det[\lambda(\boldsymbol{D}-\boldsymbol{L})-\boldsymbol{U}]\end{aligned}$$

由于 $\det[(\boldsymbol{D}-\boldsymbol{L})^{-1}]\neq 0$,因此

$$\det[\lambda(\boldsymbol{D}-\boldsymbol{L})-\boldsymbol{U}]=0. \tag{2.20}$$

另一方面,考虑矩阵

$$\lambda(\boldsymbol{D}-\boldsymbol{L})-\boldsymbol{U}=\begin{bmatrix}\lambda a_{11} & -a_{12} & \cdots & -a_{1,n-1} & -a_{1n}\\ -\lambda a_{21} & \lambda a_{22} & \cdots & \vdots & \vdots\\ \vdots & \vdots & \cdots & \lambda a_{n-1,n-1} & -a_{n-1,n}\\ -\lambda a_{n1} & -\lambda a_{n2} & \cdots & -\lambda a_{n,n-1} & \lambda a_{nn}\end{bmatrix}.$$

ⅰ) 当 $\boldsymbol{A}$ 为按行严格对角占优时,由假定 $|\lambda|>1$ 即得

$$|\lambda|\cdot|a_{ii}|>|\lambda|\Big(\sum_{j=1}^{i-1}|a_{ij}|+\sum_{j=i+1}^{n}|a_{ij}|\Big)\geqslant|\lambda|\sum_{j=1}^{i-1}|a_{ij}|+\sum_{j=i+1}^{n}|a_{ij}|,$$

于是矩阵$[\lambda(\boldsymbol{D}-\boldsymbol{L})-\boldsymbol{U}]$为行严格对角占优,再由引理 2.1 知它为非奇异矩阵,与式(2.20)矛盾,从而谱半径 $\rho(\boldsymbol{B}_G)<1$;

ⅱ) 当 $\boldsymbol{A}$ 为按行弱对角占优且为不可约矩阵时,由类似于ⅰ)的推导可得矩阵$[\lambda(\boldsymbol{D}-\boldsymbol{L})-\boldsymbol{U}]$为按行弱对角占优且为不可

约矩阵,再由引理 2.1 知它为非奇异矩阵,与式(2.20)矛盾,从而谱半径 $\rho(\boldsymbol{B}_G)<1$.

由 ⅰ)、ⅱ)及定理 2.1 即得高斯-塞德尔迭代收敛.

类似地,当 $\boldsymbol{A}$ 为按行严格对角占优或按行弱对角占优且为不可约矩阵时,可以证明雅可比迭代法的迭代矩阵的谱半径 $\rho(\boldsymbol{B}_J)<1$,从而雅可比迭代收敛.

(2) 若 $\boldsymbol{A}$ 对称正定,于是高斯-塞德尔迭代的迭代矩阵为

$$\boldsymbol{B}_G=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}$$

设 λ 为 $\boldsymbol{B}_G$ 的任一特征值,$0\neq z\in R^n$ 为其对应的特征向量,即

$$(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}z=\lambda z,$$

这里 λ 和 z 分别可能是复特征值和复特征向量. 设 z^{H} 为 z 的转置复共轭向量,于是由上式得到

$$\boldsymbol{L}^{\mathrm{T}}z=\lambda(\boldsymbol{D}-\boldsymbol{L})z,$$

设 $z^{\mathrm{H}}\boldsymbol{L}z=\alpha+\mathrm{i}\beta$, $z^{\mathrm{H}}\boldsymbol{D}z=\sigma>0$,于是

$$\lambda=\frac{z^{\mathrm{H}}\boldsymbol{L}^{\mathrm{T}}z}{z^{\mathrm{H}}(\boldsymbol{D}-\boldsymbol{L})z}=\frac{\alpha-\mathrm{i}\beta}{\sigma-\alpha-\mathrm{i}\beta}. \tag{2.21}$$

另一方面,由 $\boldsymbol{A}$ 对称正定得

$$0<z^{\mathrm{H}}\boldsymbol{A}z=z^{\mathrm{H}}\boldsymbol{D}z-z^{\mathrm{H}}\boldsymbol{L}z-z^{\mathrm{H}}\boldsymbol{L}^{\mathrm{T}}z=\sigma-2\alpha.$$

由上式、式(2.21)以及 $\sigma>0$ 可以得到

$$|\lambda|^2=\frac{\alpha^2+\beta^2}{(\sigma-\alpha)^2+\beta^2}=\frac{\alpha^2+\beta^2}{\alpha^2+\beta^2+(\sigma-2\alpha)\sigma}<1.$$

即 $|\lambda|<1$,从而谱半径 $\rho(\boldsymbol{B}_G)<1$,所以高斯-塞德尔迭代收敛.

(3) 由 $\boldsymbol{A}$ 对称正定,于是 $\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D}-\boldsymbol{A})\boldsymbol{D}^{-\frac{1}{2}}$ 对称,并有完全标准正交特征向量系,不妨设为 $\{\boldsymbol{y}_k\}_{k=1}^n$,其对应的 n 个特征值 $\lambda_k\in R$ 为实数,并有

$$\begin{cases}\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D}-\boldsymbol{A})\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{y}_k=\lambda\boldsymbol{y}_k,\\ \boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{y}_l=\delta_{kl}.\end{cases}$$

设 $z_k = D^{-\frac{1}{2}} y_k$，并由简单计算得

$$\begin{cases} B_J z_k = D^{-1}(D - A) D^{-\frac{1}{2}} y_k = D^{-\frac{1}{2}} \lambda_k y_k = \lambda z_k, \\ z_k^{\mathrm{T}} D z_l = \delta_{kl}. \end{cases}$$

从而，$\{z_k\}_{k=1}^n$ 构成 B_J 的完全特征向量系，$\{\lambda_k\}_{k=1}^n$ 为其对应的所有特征值.

若雅可比迭代收敛，那么由定理 2.1 知谱半径 $\rho(B_J) < 1$，于是对 $k = 1, 2, \cdots, n$，有

$$\begin{aligned} z_k^{\mathrm{T}}(2D - A) z_l &= z_k^{\mathrm{T}}[D + DD^{-1}(D - A)] z_l \\ &= z_k^{\mathrm{T}} D z_l + z_k^{\mathrm{T}} D \lambda_l z_l = (1 + \lambda_l)\delta_{kl}. \end{aligned}$$

所以，对任意的 $z \in R^n$，有 $z = \sum_{k=1}^n c_k z_k$，并且

$$z^{\mathrm{T}}(2D - A) z = \sum_{k,l=1}^{n} c_k c_l z_k^{\mathrm{T}}(2D - A) z_l = \sum_{k=1}^{n} c_k^2 (1 + \lambda_k) > 0,$$

即 $(2D - A)$ 对称正定.

反之，若 $(2D - A)$ 对称正定，设 λ 为 B_J 的任一特征值，其对应的特征向量为 z，即

$$B_J z = D^{-1}(D - A) z = \lambda z. \tag{2.22}$$

设 $\sigma = z^{\mathrm{T}} D z > 0$，$\alpha = z^{\mathrm{T}}(D - A) z$，于是由式(2.22)得到

$$\lambda = \frac{z^{\mathrm{T}}(D - A) z}{z^{\mathrm{T}} D z} = \frac{\alpha}{\sigma}.$$

又因 $2D - A$，A 对称正定，即有

$$\begin{cases} 0 < z^{\mathrm{T}}(2D - A) z = \sigma + \alpha, \\ 0 < z^{\mathrm{T}} A z = z^{\mathrm{T}}(A - D + D) z = -\alpha + \sigma. \end{cases}$$

由上式可得 $-\sigma < \alpha < \sigma$，从而 $|\alpha| < \sigma$，因此

$$|\lambda| = \left| \frac{\alpha}{\sigma} \right| < 1.$$

于是谱半径 $\rho(B_J) < 1$，所以雅可比迭代收敛.

定理 2.3 若 SOR 迭代收敛，则松弛因子满足 $\omega \in (0, 2)$.

证明 设 SOR 方法的迭代矩阵为 B_S，$\lambda_1, \lambda_2, \cdots, \lambda_n$ 为 B_S 的 n

个特征值，于是

$$\begin{aligned}(\rho(\boldsymbol{B}_S))^n &\geqslant |\lambda_1\lambda_2\cdots\lambda_n| = |\det \boldsymbol{B}_S| \\ &= |\det[(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}-\omega\boldsymbol{U})]| \\ &= |\det[(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}]\det[(1-\omega)\boldsymbol{D}-\omega\boldsymbol{U}]| \\ &= |\det(\boldsymbol{D}^{-1})\cdot(1-\omega)^n\cdot\det(\boldsymbol{D})| \\ &= |(1-\omega)^n|.\end{aligned}$$

从而

$$|1-\omega| \leqslant \rho(\boldsymbol{B}_S) < 1.$$

所以

$$0 < \omega < 2.$$

定理 2.4 对线性代数方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$，

(1) 若 $\boldsymbol{A}$ 为对称正定矩阵，则当 $0 < \omega < 2$ 时，SOR 迭代收敛；

(2) 若 $\boldsymbol{A}$ 为按行(或按列)严格对角占优矩阵，则当 $0 < \omega \leqslant 1$ 时，SOR 迭代收敛.

证明 (1) 考虑 SOR 迭代矩阵

$$\boldsymbol{B}_S = (\boldsymbol{D}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}).$$

由 $\boldsymbol{A}$ 对称正定知 $\boldsymbol{U} = \boldsymbol{L}^{\mathrm{T}}$，于是

$$\boldsymbol{B}_S = (\boldsymbol{D}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}^{\mathrm{T}}).$$

设 λ 为 $\boldsymbol{B}_S$ 的任一特征值，$z \neq 0$ 为对应的特征向量，即

$$(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}^{\mathrm{T}})z = \lambda z,$$

这里 λ 和 z 可能是复特征值和复特征向量. 设 z^{H} 为 z 的转置复共轭向量，于是由上式得到

$$z^{\mathrm{H}}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}^{\mathrm{T}})z = \lambda z^{\mathrm{H}}(\boldsymbol{D}-\omega\boldsymbol{L})z,$$

设 $z^{\mathrm{H}}\boldsymbol{L}z = \alpha + \mathrm{i}\beta$，$z^{\mathrm{H}}\boldsymbol{D}z = \sigma > 0$，于是

$$\begin{aligned}\lambda &= \frac{z^{\mathrm{H}}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}^{\mathrm{T}})z}{z^{\mathrm{H}}(\boldsymbol{D}-\omega\boldsymbol{L})z} \\ &= \frac{(1-\omega)\sigma+\omega(\alpha-\mathrm{i}\beta)}{\sigma-\omega(\alpha+\mathrm{i}\beta)}\end{aligned}$$

$$= \frac{(\sigma - \omega\sigma + \omega\alpha) - \mathrm{i}\omega\beta}{(\sigma - \omega\alpha) - \mathrm{i}\omega\beta},$$

其模的平方为

$$|\lambda|^2 = \frac{(\sigma - \omega\sigma + \omega\alpha)^2 + (\omega\beta)^2}{(\sigma - \omega\alpha)^2 + (\omega\beta)^2}. \tag{2.23}$$

另一方面,由简单计算知

$$0 < z^{\mathrm{H}}\boldsymbol{A}z = -z^{\mathrm{H}}\boldsymbol{L}z + z^{\mathrm{H}}\boldsymbol{D}z - z^{\mathrm{H}}\boldsymbol{L}^{\mathrm{T}}z = \sigma - 2\alpha.$$

由上式及 $\sigma > 0$、$0 < \omega < 2$ 得

$$(\sigma - \omega\sigma + \omega\alpha)^2 - (\sigma - \omega\alpha)^2 = -\sigma\omega(2 - \omega)(\sigma - 2\alpha) < 0.$$

所以,由上式和式(2.23)可得 $|\lambda|^2 < 1$,即 $|\lambda| < 1$,从而谱半径 $\rho(\boldsymbol{B}_S) < 1$,根据定理 2.1 知 SOR 迭代收敛.

(2) 我们仅对 $\boldsymbol{A}$ 为按行严格对角占优矩阵时证明定理结果,按列情况的证明也可类似进行.

用反证法,若不收敛,则 SOR 迭代法的迭代矩阵的谱半径 $\rho(\boldsymbol{B}_S) \geqslant 1$,于是存在 $\boldsymbol{B}_S$ 的某一特征值 λ,$|\lambda| \geqslant 1$,满足

$$\begin{aligned} 0 &= \det(\lambda\boldsymbol{I} - \boldsymbol{B}_S) \\ &= \det[\lambda\boldsymbol{I} - (\boldsymbol{D} - \omega\boldsymbol{L})^{-1}((1 - \omega)\boldsymbol{D} + \omega\boldsymbol{U})] \\ &= \det[(\boldsymbol{D} - \omega\boldsymbol{L})^{-1}] \cdot \det[\lambda(\boldsymbol{D} - \omega\boldsymbol{L}) - ((1 - \omega)\boldsymbol{D} + \omega\boldsymbol{U})]. \end{aligned}$$

由于 $\det[(\boldsymbol{D} - \omega\boldsymbol{L})^{-1}] \neq 0$,因此

$$\det[(\lambda - 1 + \omega)\boldsymbol{D} - \lambda\omega\boldsymbol{L} - \omega\boldsymbol{U}] = 0. \tag{2.24}$$

设 $\lambda = \alpha + \mathrm{i}\beta$,由 $|\lambda| \geqslant 1$,$0 < \omega \leqslant 1$ 可以得到

$$\begin{aligned} |\lambda - 1 + \omega|^2 - |\lambda\omega|^2 &= [(\alpha - 1 + \omega)^2 + \beta^2] - [\omega^2(\alpha^2 + \beta^2)] \\ &= (1 - \omega)[\omega(\alpha^2 + \beta^2 - 1) + (\alpha - 1)^2 + \beta^2] \\ &\geqslant 0, \end{aligned}$$

于是

$$|\lambda - 1 + \omega| \geqslant |\lambda\omega|.$$

由上式,$\boldsymbol{A}$ 严格对角占优以及 $|\lambda| \geqslant 1$,可以推出矩阵

$$\boldsymbol{C} = [(\lambda - 1 + \omega)\boldsymbol{D} - \lambda\omega\boldsymbol{L} - \omega\boldsymbol{U}]$$

中的元素满足

$$
\begin{aligned}
|c_{ii}| &= |(\lambda - 1 + \omega)a_{ii}| \\
&\geqslant |\lambda\omega| \cdot |a_{ii}| \\
&> |\lambda\omega| \cdot \Big(\sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^{n} |a_{ij}|\Big) \\
&\geqslant \sum_{j=1}^{i-1} |\lambda\omega a_{ij}| + \sum_{j=i+1}^{n} |\omega a_{ij}| \\
&= \sum_{\substack{j=1 \\ j\neq i}}^{n} |c_{ij}| .
\end{aligned}
$$

即矩阵$[(\lambda - 1 + \omega)\boldsymbol{D} - \lambda\omega\boldsymbol{L} - \omega\boldsymbol{U}]$严格对角占优,其行列式非零,与式(2.24)矛盾,从而 $\rho(\boldsymbol{B}_S) < 1$,即 SOR 迭代收敛.

定理 2.5 对线性代数方程组 $\boldsymbol{Ax} = \boldsymbol{b}$,若 $\boldsymbol{A}$ 为对称正定矩阵,则当 $0 < \omega < 2$ 时,SSOR 迭代收敛.

证明:考虑 SSOR 迭代矩阵

$$
\begin{aligned}
\boldsymbol{B}_{SS} = &\left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{U}\right)^{-1} \boldsymbol{D} \left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{L}\right)^{-1} \\
&\cdot \left(\frac{\omega - 1}{\omega}\boldsymbol{D} - \boldsymbol{L}\right) \boldsymbol{D}^{-1} \left(\frac{\omega - 1}{\omega}\boldsymbol{D} - \boldsymbol{U}\right)
\end{aligned}
$$

由于 $\boldsymbol{A}$ 为对称正定矩阵,于是 $\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{L}$,取

$$
\boldsymbol{W} = \boldsymbol{D}^{-\frac{1}{2}} \left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{U}\right),
$$

$$
\boldsymbol{P} = \boldsymbol{D}^{\frac{1}{2}} \left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{L}\right)^{-1} \left(\frac{\omega - 1}{\omega}\boldsymbol{D} - \boldsymbol{L}\right) \boldsymbol{D}^{-\frac{1}{2}},
$$

可以计算得到

$$
\begin{aligned}
\boldsymbol{W}\boldsymbol{B}_{SS}\boldsymbol{W}^{-1} = &\left[\boldsymbol{D}^{\frac{1}{2}} \left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{L}\right)^{-1} \left(\frac{\omega - 1}{\omega}\boldsymbol{D} - \boldsymbol{L}\right) \boldsymbol{D}^{-\frac{1}{2}}\right] \\
&\cdot \left[\boldsymbol{D}^{-\frac{1}{2}} \left(\frac{\omega - 1}{\omega}\boldsymbol{D} - \boldsymbol{L}^{\mathrm{T}}\right) \left(\frac{\boldsymbol{D}}{\omega} - \boldsymbol{L}^{\mathrm{T}}\right)^{-1} \boldsymbol{D}^{\frac{1}{2}}\right], \\
= &\boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}
\end{aligned}
$$

$$
\boldsymbol{W}(\boldsymbol{I} - \boldsymbol{B}_{SS})\boldsymbol{W}^{-1} = \frac{2 - \omega}{\omega}(\boldsymbol{W}^{-1})^{\mathrm{T}} \boldsymbol{A} (\boldsymbol{W}^{-1}).
$$

从而 $\boldsymbol{B}_{SS}$, $\boldsymbol{I} - \boldsymbol{B}_{SS}$都为对称正定矩阵,因此 $\rho(\boldsymbol{B}_{SS}) < 1$,即 SSOR 迭代收敛.

2.5 共轭梯度法

在本节中，我们将对线性代数方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

当矩阵 $\boldsymbol{A} \in R^{n\times n}$ 为对称正定时，引进一种迭代方法——共轭梯度法，并讨论基于这种方法发展起来的一类加速方法——预条件共轭梯度法.

算法 2.5 给定的对称正定矩阵 $\boldsymbol{A} \in R^{n\times n}$，求解线性代数方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 的共轭梯度法.

选定初值 $\boldsymbol{x}^{(0)} \in R^n$，设 $\boldsymbol{r}^{(0)} = \boldsymbol{d}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$，对 $k = 1,2,\cdots$，计算

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k, \quad \left(\text{其中 } \alpha_k = \frac{(\boldsymbol{r}_k, \boldsymbol{r}_k)}{(\boldsymbol{d}_k, \boldsymbol{A}\boldsymbol{d}_k)}\right);$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{d}_k;$$

$$\boldsymbol{d}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{d}_k, \quad \left(\text{其中 } \beta_k = \frac{(\boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1})}{(\boldsymbol{r}_k, \boldsymbol{r}_k)}\right),$$

其中，$(\cdot,\cdot)$ 为 R^n 中的内积，即 $(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$.

引理 2.4 在算法 2.5 中，如果对 $1 \leqslant k \leqslant n-1$，$\boldsymbol{r}_0, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_k$ 均不为零，则 $\boldsymbol{d}_0, \boldsymbol{d}_1, \cdots, \boldsymbol{d}_k$ 亦不为零，从而 $\alpha_0, \alpha_1, \cdots, \alpha_k$ 与 $\beta_0, \beta_1, \cdots, \beta_{k-1}$ 都不为零，并且

$$(\boldsymbol{r}_i, \boldsymbol{r}_j) = 0, \quad (\boldsymbol{r}_i, \boldsymbol{d}_j) = 0, \quad (\boldsymbol{d}_i, \boldsymbol{A}\boldsymbol{d}_j) = 0, \quad 0 \leqslant j < i \leqslant k.$$

证明 用归纳法证明. 当 $k = 1$ 时，由算法 2.5 知

$$\begin{aligned}(\boldsymbol{r}_1, \boldsymbol{r}_0) &= (\boldsymbol{r}_1 - \boldsymbol{r}_0, \boldsymbol{r}_0) + (\boldsymbol{r}_0, \boldsymbol{r}_0)\\ &= (-\alpha_0 \boldsymbol{A}\boldsymbol{d}_0, \boldsymbol{r}_0) + (\boldsymbol{r}_0, \boldsymbol{r}_0)\\ &= -\frac{(\boldsymbol{r}_0, \boldsymbol{r}_0)}{(\boldsymbol{d}_0, \boldsymbol{A}\boldsymbol{d}_0)}(\boldsymbol{A}\boldsymbol{d}_0, \boldsymbol{d}_0) + (\boldsymbol{r}_0, \boldsymbol{r}_0)\\ &= 0,\end{aligned}$$

$$(\boldsymbol{r}_1, \boldsymbol{d}_0) = (\boldsymbol{r}_1, \boldsymbol{r}_0) = 0.$$

于是

$$(\boldsymbol{d}_1, \boldsymbol{d}_1) = (\boldsymbol{r}_1 + \boldsymbol{\beta}_0 \boldsymbol{d}_0, \boldsymbol{r}_1 + \boldsymbol{\beta}_0 \boldsymbol{d}_0) = (\boldsymbol{r}_1, \boldsymbol{r}_1) + \boldsymbol{\beta}_0^2 (\boldsymbol{d}_0, \boldsymbol{d}_0) > 0,$$

即 $\boldsymbol{d}_1 \neq 0$, 又 $\boldsymbol{d}_0 = \boldsymbol{r}_0 \neq 0$, 从而 $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1$ 和 $\boldsymbol{\beta}_0$ 都不为零, 并有

$$\begin{aligned}(\boldsymbol{d}_1, \boldsymbol{A}\boldsymbol{d}_0) &= \left(\boldsymbol{r}_1 + \boldsymbol{\beta}_0 \boldsymbol{d}_0, \frac{1}{\boldsymbol{\alpha}_0}(\boldsymbol{r}_0 - \boldsymbol{r}_1)\right) \\ &= \frac{1}{\boldsymbol{\alpha}_0}[-(\boldsymbol{r}_1, \boldsymbol{r}_1) + \boldsymbol{\beta}_0 (\boldsymbol{r}_0, \boldsymbol{r}_0)] \\ &= 0.\end{aligned}$$

假设当 $k = m$ 时结论成立, 那么, 当 $k = m + 1$ 时, 对 $j = 1, 2, \cdots, m - 1$, 由算法 2.5 可得

$$(\boldsymbol{r}_{m+1}, \boldsymbol{r}_j) = (\boldsymbol{r}_{m+1} - \boldsymbol{r}_m, \boldsymbol{r}_j) = -\boldsymbol{\alpha}_m (\boldsymbol{A}\boldsymbol{d}_m, \boldsymbol{d}_j - \boldsymbol{\beta}_{j-1} \boldsymbol{d}_{j-1}) = 0,$$

$$(\boldsymbol{r}_{m+1}, \boldsymbol{d}_j) = (\boldsymbol{r}_m - \boldsymbol{\alpha}_m \boldsymbol{A}\boldsymbol{d}_m, \boldsymbol{d}_j) = 0,$$

而且

$$\begin{aligned}(\boldsymbol{r}_{m+1}, \boldsymbol{r}_m) &= (\boldsymbol{r}_m, \boldsymbol{r}_m) - \boldsymbol{\alpha}_m (\boldsymbol{A}\boldsymbol{d}_m, \boldsymbol{r}_m) \\ &= (\boldsymbol{r}_m, \boldsymbol{r}_m) - \frac{(\boldsymbol{r}_m, \boldsymbol{r}_m)}{(\boldsymbol{d}_m, \boldsymbol{A}\boldsymbol{d}_m)} (\boldsymbol{A}\boldsymbol{d}_m, \boldsymbol{d}_m - \boldsymbol{\beta}_{m-1} \boldsymbol{d}_{m-1}) \\ &= 0\end{aligned}$$

$$(\boldsymbol{r}_{m+1}, \boldsymbol{d}_m) = (\boldsymbol{r}_{m+1}, \boldsymbol{r}_m + \boldsymbol{\beta}_{m-1} \boldsymbol{d}_{m-1}) = 0.$$

于是

$$\begin{aligned}(\boldsymbol{d}_{m+1}, \boldsymbol{d}_{m+1}) &= (\boldsymbol{r}_{m+1} + \boldsymbol{\beta}_m \boldsymbol{d}_m, \boldsymbol{r}_{m+1} + \boldsymbol{\beta}_m \boldsymbol{d}_m) \\ &= (\boldsymbol{r}_{m+1}, \boldsymbol{r}_{m+1}) + \boldsymbol{\beta}_m^2 (\boldsymbol{d}_m, \boldsymbol{d}_m) \\ &> 0,\end{aligned}$$

即 $\boldsymbol{d}_{m+1} \neq 0$, 从而 $\boldsymbol{\alpha}_{m+1}, \boldsymbol{\beta}_m$ 都不为零. 再对 $j = 1, 2, \cdots, m - 1$ 有

$$\begin{aligned}(\boldsymbol{d}_{m+1}, \boldsymbol{A}\boldsymbol{d}_j) &= (\boldsymbol{d}_{m+1} - \boldsymbol{\beta}_m \boldsymbol{d}_m, \boldsymbol{A}\boldsymbol{d}_j) \\ &= \left(\boldsymbol{r}_{m+1}, \frac{1}{\alpha_j}(\boldsymbol{r}_j - \boldsymbol{r}_{j+1})\right) \\ &= 0,\end{aligned}$$

而且

$$
\begin{aligned}
&(\boldsymbol{d}_{m+1}, \boldsymbol{A}\boldsymbol{d}_m) \\
&\quad = (\boldsymbol{r}_{m+1}, \boldsymbol{A}\boldsymbol{d}_m) + \boldsymbol{\beta}_m(\boldsymbol{d}_m, \boldsymbol{A}\boldsymbol{d}_m) \\
&\quad = \left(\boldsymbol{r}_{m+1}, \frac{1}{\boldsymbol{\alpha}_m}(\boldsymbol{r}_m - \boldsymbol{r}_{m+1})\right) + \frac{(\boldsymbol{r}_{m+1}, \boldsymbol{r}_{m+1})}{(\boldsymbol{r}_m, \boldsymbol{r}_m)}(\boldsymbol{d}_m, \boldsymbol{A}\boldsymbol{d}_m) \\
&\quad = -\frac{(\boldsymbol{d}_m, \boldsymbol{A}\boldsymbol{d}_m)}{(\boldsymbol{r}_m, \boldsymbol{r}_m)}(\boldsymbol{r}_{m+1}, \boldsymbol{r}_{m+1}) + \frac{(\boldsymbol{r}_{m+1}, \boldsymbol{r}_{m+1})}{(\boldsymbol{r}_m, \boldsymbol{r}_m)}(\boldsymbol{d}_m, \boldsymbol{A}\boldsymbol{d}_m) \\
&\quad = 0.
\end{aligned}
$$

引理 2.4 得证.

若用算法 2.5 计算线性代数方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$，其中 $\boldsymbol{A} \in R^{n\times n}$ 是对称正定矩阵，我们有下列结论：

定理 2.6 设 $\boldsymbol{A} \in R^{n\times n}$ 为对称正定矩阵，假定 $\boldsymbol{x}_k$ 为采用算法 2.5 计算线性代数方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 时产生的迭代序列，并取

$$
\kappa = \frac{\lambda_{\max}(\boldsymbol{A})}{\lambda_{\min}(\boldsymbol{A})},
$$

它为矩阵 $\boldsymbol{A}$ 的条件数 $\mathrm{Cond}_2(\boldsymbol{A})$，那么

(1) 用不超过 n 次的算法 2.5 的迭代(如果计算是精确的)就可求得精确解；

(2) 对于每步的迭代值 $\boldsymbol{x}_k$，有如下的误差估计：

$$
\| \boldsymbol{x}_k - \boldsymbol{x}^* \|_{\boldsymbol{A}} \leqslant 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k \| \boldsymbol{x}_0 - \boldsymbol{x}^* \|_{\boldsymbol{A}},
$$

其中，$\boldsymbol{x}^*$ 为方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 的精确解，范数 $\| \boldsymbol{x} \|_{\boldsymbol{A}} = (\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})$.

证明 (1) 由算法 2.5 知

$$
\boldsymbol{A}(\boldsymbol{x}_j - \boldsymbol{x}_{j-1}) = \boldsymbol{A}\boldsymbol{\alpha}_{j-1}\boldsymbol{d}_{j-1} = \boldsymbol{r}_{j-1} - \boldsymbol{r}_j.
$$

于是

$$
\boldsymbol{A}(\boldsymbol{x}_k - \boldsymbol{x}_0) = \sum_{j=1}^{k} \boldsymbol{A}(\boldsymbol{x}_j - \boldsymbol{x}_{j-1}) = \sum_{j=1}^{k} (\boldsymbol{r}_{j-1} - \boldsymbol{r}_j) = \boldsymbol{r}_0 - \boldsymbol{r}_k.
$$

从而

$$
\boldsymbol{r}_k = \boldsymbol{r}_0 - \boldsymbol{A}(\boldsymbol{x}_k - \boldsymbol{x}_0) = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k.
$$

显然，如果对 $k \leqslant n-1$，有 $\boldsymbol{r}_k = 0$，那么，$\boldsymbol{x}_k$ 为方程组的精确解. 否

则，如果 $\boldsymbol{r}_k \neq 0 (0 \leqslant k \leqslant n-1)$，由引理 2.4 知，这 n 个向量正交，从而构成 R^n 中的一组基，再由引理 2.4 可得，对 $j = 1, 2, \cdots, k-1$，成立

$$(\boldsymbol{r}_n, \boldsymbol{r}_j) = (\boldsymbol{r}_{n-1} - \alpha_{n-1} \boldsymbol{A} \boldsymbol{d}_{n-1}, \boldsymbol{r}_j) = 0,$$

而且

$$\begin{aligned}(\boldsymbol{r}_n, \boldsymbol{r}_{n-1}) &= (\boldsymbol{r}_{n-1} - \alpha_{n-1} \boldsymbol{A} \boldsymbol{d}_{n-1}, \boldsymbol{r}_{n-1}) \\ &= (\boldsymbol{r}_{n-1}, \boldsymbol{r}_{n-1}) - \alpha_{n-1} (\boldsymbol{A} \boldsymbol{d}_{n-1}, \boldsymbol{d}_{n-1} - \beta_{n-2} \boldsymbol{d}_{n-2}) \\ &= 0.\end{aligned}$$

即 $\boldsymbol{r}_n$ 与 $\boldsymbol{r}_k (0 \leqslant k \leqslant n-1)$ 正交，从而 $\boldsymbol{b} - \boldsymbol{A} \boldsymbol{x}_n = \boldsymbol{r}_n = 0$，于是 $\boldsymbol{x}_n$ 为方程组 $\boldsymbol{A} \boldsymbol{x} = \boldsymbol{b}$ 的精确解.

(2) (留作习题).

由定理 2.6 可知，当 $\boldsymbol{A} \boldsymbol{x} = \boldsymbol{b}$ 为病态方程组时，$\kappa(\boldsymbol{A})$ 很大，算法 2.5 收敛缓慢. 因此，希望能找到一个适合下列两个条件的对称正定矩阵 $\boldsymbol{M}$:

1) 方程组 $\boldsymbol{M} \boldsymbol{h} = \boldsymbol{r}$ 容易求解;

2) 设 $\boldsymbol{M} = \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W}$ ($\boldsymbol{W} \in R^{n \times n}$ 为非奇异矩阵)，矩阵

$$\hat{\boldsymbol{A}} = (\boldsymbol{W}^{-1})^{\mathrm{T}} \boldsymbol{A} \boldsymbol{W}^{-1}$$

的条件数 $\kappa(\hat{\boldsymbol{A}}) \ll \kappa(\boldsymbol{A})$.

于是，求解方程组 $\boldsymbol{A} \boldsymbol{x} = \boldsymbol{b}$ 就等价于求解方程组

$$\hat{\boldsymbol{A}} \hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}, \tag{2.25}$$

这里

$$\hat{\boldsymbol{A}} = (\boldsymbol{W}^{-1})^{\mathrm{T}} \boldsymbol{A} \boldsymbol{W}^{-1}, \quad \hat{\boldsymbol{x}} = \boldsymbol{W} \boldsymbol{x}, \quad \hat{\boldsymbol{b}} = (\boldsymbol{W}^{-1})^{\mathrm{T}} \boldsymbol{b},$$

假定再采用共轭梯度法计算式(2.25)：选定初值 $\hat{\boldsymbol{x}}^{(0)} \in R^n$，设 $\hat{\boldsymbol{r}}^{(0)} = \hat{\boldsymbol{d}}^{(0)} = \hat{\boldsymbol{b}} - \hat{\boldsymbol{A}} \hat{\boldsymbol{x}}_0$，对 $k = 1, 2, \cdots$，计算

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \hat{\alpha}_k \hat{\boldsymbol{d}}_k, \quad \left(\text{其中 } \hat{\alpha}_k = \frac{(\hat{\boldsymbol{r}}_k, \hat{\boldsymbol{r}}_k)}{(\hat{\boldsymbol{d}}_k, \hat{\boldsymbol{A}} \hat{\boldsymbol{d}}_k)} \right);$$

$$\hat{\boldsymbol{r}}_{k+1} = \hat{\boldsymbol{r}}_k - \hat{\alpha}_k \hat{\boldsymbol{A}} \hat{\boldsymbol{d}}_k;$$

$$\hat{\boldsymbol{d}}_{k+1} = \hat{\boldsymbol{r}}_{k+1} + \hat{\beta}_k \hat{\boldsymbol{d}}_k, \quad \left(\text{其中 } \hat{\beta}_k = \frac{(\hat{\boldsymbol{r}}_{k+1}, \hat{\boldsymbol{r}}_{k+1})}{(\boldsymbol{r}_k, \boldsymbol{r}_k)} \right).$$

取

$$\bar{x}_k = W^{-1}\hat{x}_k, \quad \bar{r}_k = W^{\mathrm{T}}\hat{r}_k, \quad \bar{d}_k = W^{-1}\hat{d}_k, \quad \bar{h}_k = M^{-1}\hat{r}_k,$$

$$\bar{\alpha}_k = \frac{(\bar{h}_k, \bar{r}_k)}{(\bar{d}_k, A\bar{d}_k)}, \qquad \bar{\beta}_k = \frac{(\bar{h}_{k+1}, \bar{r}_{k+1})}{(\bar{h}_k, \bar{d}_k)},$$

并代入前面的计算公式,即得预条件共轭梯度法的计算过程.

算法 2.6 给定的对称正定矩阵 $A \in R^{n\times n}$,求解线性代数方程组 $Ax = b$ 的预条件共轭梯度法.

选定初值 $\bar{x}^{(0)} \in R^n$,设 $\bar{r}^{(0)} = b - A\bar{x}_0$,求解 $M\bar{h}^{(0)} = \bar{r}^{(0)}$,$\bar{d}_0 = \bar{h}_0$,对 $k = 1, 2, \cdots$,计算

$$\bar{x}_{k+1} = x_k + \bar{\alpha}_k\bar{d}_k, \quad \left(\text{其中 } \bar{\alpha}_k = \frac{(\bar{h}_k, \bar{r}_k)}{(\bar{d}_k, A\bar{d}_k)}\right);$$

$$\bar{r}_{k+1} = \bar{r}_k - \bar{\alpha}_k A\bar{d}_k;$$

求解:$M\bar{h}_{k+1} = \bar{r}_{k+1}$;

$$\bar{d}_{k+1} = \bar{h}_{k+1} + \bar{\beta}_k\bar{d}_k, \quad \left(\text{其中 } \bar{\beta}_k = \frac{(\bar{h}_{k+1}, \bar{r}_{k+1})}{(\bar{h}_k, \bar{r}_k)}\right).$$

在算法 2.6 中,每步迭代都需要解一个线性代数方程组 $Mh = r$,再由定理 2.6 知,算法 2.6 的收敛率为

$$\kappa = \frac{\lambda_{\max}(\hat{A})}{\lambda_{\min}(\hat{A})},$$

因此,选取 M 的好坏将直接影响到算法的计算速度和收敛速度,并称 M 为 A 的预条件矩阵.

下面我们介绍对称正定矩阵 $A \in R^{n\times n}$ 的几种预条件矩阵 M 的选取方法:

(1) 当选取

$$M = D, \quad W = D^{\frac{1}{2}},$$

那么,预条件共轭梯度法就是雅可比-cg 加速方法;

(2) 当选取

$$M = \frac{\omega}{2-\omega}\left(\frac{1}{\omega}D - L\right)D^{-1}\left(\frac{1}{\omega}D - U\right),$$

$$\boldsymbol{W}=\frac{1}{\sqrt{\omega(2-\omega)}}(\boldsymbol{D}-\omega\boldsymbol{L})\boldsymbol{D}^{-\frac{1}{2}},$$

这里，$0<\omega<2$，预条件共轭梯度法就是 SSOR-cg 加速方法；

(3) 设 $\boldsymbol{A}$ 有不完全三角分解(即不完全乔列斯基分解)

$$\boldsymbol{A}=\tilde{\boldsymbol{L}}^{\mathrm{T}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{L}}^{\mathrm{T}}-\tilde{\boldsymbol{R}},$$

且 $\tilde{\boldsymbol{L}}$ 保持 $\boldsymbol{A}$ 的稀疏性，$\tilde{\boldsymbol{D}}$ 元素为正，如果选取 $\boldsymbol{M}=\tilde{\boldsymbol{L}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{L}}^{\mathrm{T}}$，那么，预条件共轭梯度法就是不完全乔列斯基分解的 cg 加速方法(Iccg).

习 题 2

1. 用迭代法求解线代数方程组

$$\begin{bmatrix}10 & 4 & 4\\ 4 & 10 & 8\\ 4 & 8 & 10\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}13\\ 11\\ 25\end{bmatrix}.$$

(1) 分别写出雅可比迭代、高斯-塞德尔迭代、SOR 迭代(取 $\omega=1.35$)和 SSOR 迭代(取 $\omega=1.35$)的计算式；

(2) 对任意初值，迭代式是否收敛？为什么？

2. 设线性代数方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},$$

讨论

$$(1)\ \boldsymbol{A}=\begin{bmatrix}1 & 0 & 1\\ -1 & 1 & 0\\ 1 & 2 & -3\end{bmatrix},\qquad (2)\ \boldsymbol{A}=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & 1 & \frac{1}{2}\\ \frac{1}{2} & \frac{1}{2} & 1\end{bmatrix}$$

时用雅可比迭代、高斯-塞德尔迭代求解的收敛性.

3. 对于线代数方程组

(1) $\begin{bmatrix} 3 & 1 & 2 \\ 0 & 4 & 1 \\ 1 & 0 & 2 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 2 \end{bmatrix}$;

(2) $\begin{bmatrix} 2 & 1 & 3 \\ 1 & 4 & 0 \\ 2 & 0 & 1 \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 2 \end{bmatrix}$,

讨论用高斯-塞德尔迭代法求解时的收敛性,并说明高斯-塞德尔迭代法的收敛情况可能由方程组中方程或未知元的改变而改变.

4. 用雅可比迭代法、高斯-塞德尔迭代法求解下列线性代数方程组(如果可能),且比较收敛速度,要求当 $\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \|_2 \leqslant 0.01$ 时迭代终止.

(1) $\begin{bmatrix} 2 & -1 & 10 & 0 \\ 0 & 3 & -1 & 8 \\ 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -11 \\ -11 \\ 6 \\ 25 \end{bmatrix}$;

(2) $\begin{bmatrix} 10 & -1 & 0 \\ -1 & 10 & -2 \\ 0 & -4 & 10 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 7 \\ 6 \end{bmatrix}$;

(3) $\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 6 \\ -2 \\ 6 \end{bmatrix}$;

(4) 试重排方程组(1),再解之,以改善结果.

5. 用 SOR 迭代法和 SSOR 迭代法求解下列线代数方程组(分别取 $\omega = 1.03, \omega = 1.00, \omega = 1.10$)

$$\begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ -3 \end{bmatrix},$$

要求当 $\| x^* - x^{(k)} \|_\infty < 5 \times 10^{-6}$ 时迭代终止，其中精确解 $x^* = (0.5, 1.0, -0.5)^T$，并对每个 ω 值确定迭代次数.

6. 设线性代数方程组 $Ax = b$，其中 $A \in R^{n \times n}$ 为对称正定阵（且设 A 的特征值满足：$0 < \alpha \leqslant \lambda(A) \leqslant \beta$），又设有迭代法

$$x^{(k+1)} = x^{(k)} + \omega(b - Ax^{(k)}) \quad (k = 0, 1, \cdots),$$

证明：当 $0 < \omega < \frac{2}{\beta}$ 时，上述迭代法收敛.

7. 试证矩阵

$$A = \begin{bmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{bmatrix}$$

对于 $-0.5 < a < 1$ 时是正定的，当 $-0.5 < a < 0.5$ 时，用雅可比迭代法求解 $Ax = b$ 是收敛的.

8. 用共轭梯度法、雅可比-cg 加速方法和 SSOR-cg 加速方法求解线性代数方程组

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 2 \end{bmatrix}.$$

9. 证明定理 2.6 中的(2).

第三章　解非线性方程的迭代法

3.1　引　言

在科学和工程技术的研究中，常常需要求解非线性方程

$$f(x)=0 \tag{3.1}$$

的问题．式(3.1)的解通常称为方程的根，或称为函数 $f(x)$ 的零点．当 $f(x)$ 为多项式时，即

$$f(x)=a_0+a_1x+\cdots+a_nx^n \qquad (a_n\neq 0),$$

则称式(3.1)为代数方程，否则称为超越方程．例如

$$x^2-5x+1=0 \tag{3.2}$$

$$2^x-5x+2=0 \tag{3.3}$$

式(3.2)是代数方程，式(3.3)是超越方程．如果 $f(x)$ 可以分解为

$$f(x)=(x-x^*)^m g(x),$$

其中，$0<|g(x^*)|<\infty$，m 为正整数，则称 x^* 为 $f(x)$ 的 m 重零点，当 $m=1$ 时，称 x^* 为 $f(x)$ 的单重零点．零点 x^* 可能是实数，也可能是复数．

对于高次代数方程，由代数学基本定理知，其根(包括实根或复根)的个数与代数方程的次数相同，对于超越方程，可能没有零点，也可能有一个或若干个零点，甚至无穷多个零点．求解非线性方程零点的问题，没有一般的求根公式，而实际问题也并不一定需要得到精确解，只要求得满足精度要求的根的近似值就可以了．求解非线性方程的根的问题大致可分为下面三个方面：

(1) 根的存在性．即方程有没有根？如果有根，有几个根？

(2) 根的分布．即求出根的区间．

(3) 根的精确化．即在已知一个根的近似值后，设法逐步把根精确化，直到满足精度为止．

本章我们将对求解非线性方程(3.1)问题介绍几种常用的、有效的数值计算方法．

3.2 二分法

假设 $f(x)$是定义在某区域内的连续函数，在某区间$[a,b]$内有且仅有一个单根 x^*，则逐步搜索法的步骤如下：

(1) 判断 $f(a)$的符号．若 $f(a)=0$，则 $a=x^*$，若 $f(a)\neq 0$，则不妨假设 $f(a)>0$.

(2) 选择适当的步长 $h=\dfrac{b-a}{n}$，搜索一步，看 $f(a+h)$的符号，若 $f(a+h)=0$，则 $x^*=a+h$ 已找到．若 $f(a+h)<0$ 则可知 $x^*\in(a,a+h)$，这时，可取 a 或 $a+h$ 作为 x^* 的近似值．若 $f(a+h)>0$，则继续往前搜索一步，看 $f(a+2h)$的符号，直到 $f(a+kh)$与 $f(a+(k-1)h)$异号，则可知 $x^*\in(x_{k-1},x_k)$．这里 $x_{k-1}=a+(k-1)h$，$x_k=a+kh$，这时，可取 x_k 或 x_{k-1}作为 x^* 的近似值．

这样搜索，步长 h 的选择很难恰到好处，若 h 取得较大，则精度较差，若 h 足够小，精度提高了，但计算量增加了许多．因此，如果精度要求较高的话，此方法不太经济．

二分法是上述方法的一种改进．条件如上．

设 $f(a)>0$，记 $x_0=\dfrac{a+b}{2}$，若 $f(x_0)>0$，则 $x^*\in(x_0,b)$，若 $f(x_0)=0$，则 $x^*=x_0$，若 $f(x_0)<0$，则 $x^*\in(a,x_0)$．并记新的有根区间为(a_1,b_1)，继续判别 $f\left(\dfrac{a_1+b_1}{2}\right)$的符号，记 $x_1=\dfrac{a_1+b_1}{2}$．类似，又可得到新的有根区间(a_2,b_2)，…，继续此过程，得到一系

列有根区间:$(a,b),(a_1,b_1),(a_2,b_2),\cdots,(a_k,b_k),\cdots$. 这里 $b_k-a_k=\frac{1}{2^k}(b-a)\to 0(k\to\infty)$. 所有有根区间的中点 $x_1,x_2,\cdots,x_k,\cdots$,形成了一个序列,此序列必以 x^* 为极限,即 $\lim\limits_{k\to\infty}x_k=x^*$.

但实际过程不可能无限重复,事实上也无此必要. 因为数值计算总允许有误差,当误差小于预先给定的精度 ε,就可停止计算,即当

$$|x^*-x_k|\leqslant\frac{1}{2}(b_k-a_k)=b_{k+1}-a_{k+1}<\varepsilon$$

时,就可认为 x_k 是满足方程 $f(x)=0$ 的根.

二分法是求单根的行之有效的方法. 其优点是算法简单,收敛性可得到保证. 但它的缺点是,在求根过程中,只用到了函数的符号,而未用到计算出的函数值,这总有点浪费.

3.3 迭代法

迭代法是一种逐步逼近根的方法. 它使用某个固定公式,反复校正根的近似值,使之达到精度要求.

首先,把方程 $f(x)=0$ 化为等价的形式:

$$x=g(x) \tag{3.4}$$

由于 $x=g(x)$ 是隐式,因此不可能直接求出根,需要先给出一个初值 x_0(猜测值),代入式(3.4)右端,得到 $x_1=g(x_0)$,进而可有

$$\begin{aligned} x_2&=g(x_1)\\ &\cdots\\ x_{k+1}&=g(x_k)\end{aligned} \tag{3.5}$$

若按格式 $x_{k+1}=g(x_k)$ 得到的序列 $\{x_k\}$ 有极限,即 $\lim\limits_{k\to\infty}x_k=x^*$,则称迭代过程式(3.2)收敛,这时,对式(3.5)取极限,可得:$x^*=g(x^*)$.(这里 $g(x)$ 是定义在区间上的连续函数),即 x^* 是

$f(x)=0$ 的根 .

这里显然有这样的问题:由 $f(x)=0$ 化为等价的 $x=g(x)$, 可以有多种不同的化法 . 例如 $x^3-x-1=0$

可以化为 $$x_{k+1}=\sqrt[3]{x_k+1},$$

也可化为 $$x_{k+1}=x_k^3-1.$$

这里,前一种迭代格式收敛,而后一种迭代格式发散 . 由此可见,必须研究迭代法的收敛性和收敛速度 .

定理 3.1 设与 $f(x)=0$ 等价的迭代方程 $x=g(x)$在$[a,b]$上具有连续的一阶导数,且

(1) 任意 $x\in[a,b]$,总有 $g(x)\in[a,b]$;

(2) 存在 $0\leqslant L<1$,使对任意 $x\in[a,b]$,有$|g(x)|\leqslant L<1$,

则

1) 方程在$[a,b]$内有唯一的根 x^*;

2) 对任意初值 x_0,迭代格式(3.5)均收敛于根 x^*;

3) $$|x^*-x_k|\leqslant\frac{1}{1-L}|x_{k+1}-x_k|,$$

$$|x^*-x_k|\leqslant\frac{L^k}{1-L}|x_1-x_0|.$$

证明 1) 先证有根 x^*. 令 $f(x)=x-g(x)$,则 $f(x)$在$[a,b]$上连续,且 $f(a)=a-g(a)\leqslant 0$. $f(b)=b-g(b)\geqslant 0$,则必有根 x^*,使 $f(x^*)=0$,即 $x^*=g(x^*)$.

设方程在$[a,b]$内还有一根 y^*: $y^*=g(y^*)$,则

$$x^*-y^*=g(x^*)-g(y^*)=g'(\zeta)(x^*-y^*) \qquad (3.6)$$

ζ 是介于 x^* 与 y^* 之间的某点 . 由于 $x^*\neq y^*$,则由式(3.6)可知 $g'(\zeta)=1$,这与已知$|g'(x)|\leqslant L<1$ 矛盾 .

2) 由条件(1)可知 $x_{k+1}=g(x_k)\in[a,b]$,故

$$\begin{aligned}|x_{k+1}-x^*|&=|g(x_k)-g(x^*)|\\&=|g'(\zeta)|\cdot|x_k-x^*|\\&\leqslant L|x_k-x^*|\end{aligned}$$

$$\leqslant L^2 |x_{k-1} - x^*|$$
$$\leqslant \cdots \leqslant L^{k+1} |x_0 - x^*| \to 0 \qquad (k \to \infty)$$

3)(略)

定理 3.1 中要求对任意 $x \in [a, b]$,均有 $|g'(x)| \leqslant L < 1$,这个条件的验证是比较困难的,通常用局部收敛性来刻画格式的收敛性 .

定义 3.1 对于 $x^* = g(x^*)$,若存在邻域 $U(x^*, \delta)$,迭代过程对于任意 $x_0 \in U(x^*, \delta)$均收敛,则称迭代过程 $x_{k+1} = g(x_k)$在 x^* 的邻域 $U(x^*, \delta)$内局部收敛 .

定理 3.2 设 $g(x)$在 $x = g(x)$的根 x^* 邻近有连续的一阶导数,且 $|g'(x^*)| < 1$,则 $x_{k+1} = g(x_k)$具有局部收敛性 .

证明 由于 $|g'(x^*)| < 1$,且 $g'(x)$在 x^* 的邻近连续,所以存在充分小的领域 $U(x^*, \delta)$. 在此邻域内恒有 $|g'(x)| \leqslant L < 1$,又,对于任意 $x \in U(x^*, \delta)$,有

$$\begin{aligned} |g(x) - x^*| &= |g(x) - g(x^*)| \\ &= |g'(\zeta)| |x - x^*| \\ &< |x - x^*| < \delta \end{aligned}$$

即 $g(x)$也落在 $U(x^*, \delta)$内,这样,就满足了定理 3.1 中的两个条件,可知迭代格式 $x_{k+1} = g(x_k)$对于任意初值 $x_0 \in U(x^*, \delta)$均收敛,即具有局部收敛性 .

例 3.1 求方程 $x = e^{-x}$在 $x = 0.5$ 附近的一个根,要求精度达到 10^{-5}.

解 设 $f(x) = x - e^{-x}$, $f(0.5) < 0$, $f(0.6) > 0$, 可知 $x^* \in (0.5, 0.6)$. 对于 $x \in (0.5, 0.6)$, $|g'(x)| = e^{-x} \leqslant e^{-0.5} \approx 0.607 < 1$,则由定理 3.2 知:$x_{k+1} = e^{-x_k}$必收敛 . 迭代 18 次,可得 $x^* \approx 0.56714$. 具体过程如下表所列:

k	x_k	k	x_k
0	0.500 000	10	0.566 907
1	0.606 531	11	0.567 277
2	0.545 239	12	0.567 067
3	0.579 703	13	0.567 186
4	0.560 065	14	0.567 119
5	0.571 172	15	0.567 157
6	0.564 867	16	0.567 135
7	0.568 438	17	0.567 148
8	0.566 409	18	0.567 141
9	0.567 560		

现在我们分析迭代过程的收敛速度 .

定义 3.2 记迭代误差为 $e_k = x^* - x_k$，若 $\lim\limits_{k\to\infty}\dfrac{e_{k+1}}{e_k^p} = c$，$c$ 是不等于 0 的常数，则称迭代过程为 p 阶收敛 . 特别当 $p = 1$ 时，为线性收敛；$p \approx 2$ 时，称为平方收敛 . 也就是说，收敛速度是指在接近收敛时，迭代误差的下降速度 .

定理 3.3 设 $g(x)$在 $x = g(x)$的根 x^*邻近有连续的二阶导数，且 $|g'(x^*)| < 1$；则当 $g'(x^*) \neq 0$ 时，线性收敛；当 $g'(x^*) = 0, g''(x^*) \neq 0$ 时，平方收敛 .

证明 从已知条件可知，迭代格式 $x_{k+1} = g(x_k)$具有局部收敛性 . 由于 $x^* - x_{k+1} = g(x^*) - g(x_k) = g'(\zeta)(x^* - x_k)$，$\zeta$ 介于 x_k 与 x^* 之间，故 $\dfrac{e_{k+1}}{e_k} = g'(\zeta)$，即有 $\lim\limits_{k\to\infty}\dfrac{e_{k+1}}{e_k} = g'(x^*)$，则当 $g'(x^*) \neq 0$ 时，是线性收敛 .

若 $g'(x^*) = 0$，则令 $g(x_k)$在 x^* 处泰勒(Taylor)展开：

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

即 $x_{k+1} = x^* + \frac{g''(\eta)}{2}(x_k - x^*)^2$，$\eta$ 介于 x_k 与 x^* 之间

即有 $$\frac{e_{k+1}}{e_k^2} = \frac{g''(\eta)}{2} \longrightarrow \frac{g''(x^*)}{2} \neq 0 \quad (k \to \infty)$$

即迭代过程为平方收敛．

3.4 迭代过程的加速

对于收敛的迭代过程，从理论上说，只要迭代足够多次，总可得到满意的精度．但有时迭代过程过于缓慢，计算量极大，实际上得不到满意的计算结果．因此，迭代过程的加速便成了一个重要的研究课题．

设 x_0 是精确解 x^* 的某个预测值，用迭代公式校正一次，得到

$$\bar{x}_1 = g(x_0).$$

假设 $g(x)$ 在 $[a, b]$ 内可微，且 $g'(x)$ 在求根范围内改变不大，近似地记为常数 L，则有

$$\bar{x}_1 - x^* = g(x_0) - g(x^*) = g'(\zeta)(x_0 - x^*) = L(x_0 - x^*)$$

解出 x^*：

$$x^* = \frac{1}{1-L}\bar{x}_1 - \frac{L}{1-L}x_0 = \bar{x}_1 + \frac{L}{1-L}(\bar{x}_1 - x_0)$$

由此可见，若把误差值 $\frac{L}{1-L}(\bar{x}_1 - x_0)$ 作为计算结果 $\bar{x}_1$ 的一种补偿，即记

$$x_1 = \bar{x}_1 + \frac{L}{1-L}(\bar{x}_1 - x_0)$$

那么，x_1 应比 $\bar{x}_1$ 近似得更好．这样，对于每一步的加速迭代方案可以表述如下：

迭代： $\bar{x}_{k+1} = g(x_k)$

改进： $x_{k+1} = \bar{x}_{k+1} + \frac{L}{1-L}(\bar{x}_{k+1} - x_k)$

$$= \frac{1}{1-L}\bar{x}_{k+1} - \frac{L}{1-L}x_k$$

上面的改进步可以看作是迭代值 $\bar{x}_{k+1}$ 和 x_k 的加权平均．这个加速过程，效果是比较显著的，我们可从下面例子中看出．

例 3.2 题同例 3.1.

解 $g'(x) = -e^{-x}$，取 $L = -0.6$，则加速公式的表述为

$$x_{k+1} = \frac{1}{1.6}(e^{-x_k} + 0.6x_k)$$

迭代方程：

k	x_k
0	0.50000
1	0.56658
2	0.56712
3	0.56714

即只要迭代三次便可得到结果．这与前面例 3.1 需要迭代 18 次相比，加速效果是明显的．

在前面的加速方案中，要用到 $g'(x) \approx L$，对于实际计算，导数的处理是比较麻烦的．因此，要设法把这个“L”去掉，仍设 x_0 为 x^* 的某个预测值，校正一次，得

$$\bar{x}_1 = g(x_0)$$

再校正一次，得 $\bar{\bar{x}}_1 = g(\bar{x}_1)$

由于 $$\bar{x}_1 - x^* = g(x_0) - g(x^*) \approx L(x_0 - x^*)$$

$$\bar{\bar{x}}_1 - x^* = g(\bar{x}_1) - g(x^*) \approx L(\bar{x}_1 - x^*)$$

上面两式相除，得

$$\frac{\bar{x}_1 - x^*}{\bar{\bar{x}}_1 - x^*} \approx \frac{x_0 - x^*}{\bar{x}_1 - x^*}$$

解出 x^*：$$x^* = \frac{x_0\bar{\bar{x}}_1 - \bar{x}_1^2}{x_0 - 2\bar{x}_1 + \bar{\bar{x}}_1} = \bar{x}_1 - \frac{(\bar{\bar{x}}_1 - \bar{x}_1)^2}{x_0 - 2\bar{x}_1 + \bar{\bar{x}}_1},$$

这就是说，把误差值 $-\dfrac{(\bar{\bar{x}}_1 - \bar{x}_1)^2}{x_0 - 2\bar{x}_1 + \bar{\bar{x}}_1}$ 作为 $\bar{\bar{x}}_1$ 的一种补偿，补偿后的结果应比 $\bar{x}_1$ 近似得更好一些．作为一般步骤，它的具体算式如下：

迭代：$\bar{x}_{k+1} = g(x_k)$，

再迭代：$\bar{\bar{x}}_{k+1} = g(\bar{x}_{k+1})$，

改进：$$x_{k+1} = \bar{\bar{x}}_{k+1} - \frac{(\bar{\bar{x}}_{k+1} - \bar{x}_{k+1})^2}{x_k - 2\bar{x}_{k+1} + \bar{\bar{x}}_{k+1}},$$

这种方法，称为埃特金(Aitken)加速方法．

注　上面公式中的分母：$x_k - 2\bar{x}_{k+1} + \bar{\bar{x}}_{k+1} = (x_k - \bar{x}_{k+1}) + (\bar{\bar{x}}_{k+1} - \bar{x}_{k+1})$，由于收敛慢(才需要加速)，因此右端的二项中，均相差不大，因此要注意有效数字的损失．因此，这种方法一般最好在双倍位字长的机器上使用．

3.5　牛顿法

首先我们导出牛顿法的计算公式．对于方程 $f(x) = 0$，为了使用迭代法，应先改写成 $x = g(x)$的形式，这个迭代函数 $g(x)$可以有各种不同的形式．譬如，显然可以令 $g(x) = x + f(x)$．这时相应的迭代公式就是 $x_{k+1} = x_k + f(x_k)$，但这种格式不一定收敛，或者收敛速度很慢．

我们将 3.3 节中的加速技巧用于 $g(x) = x + f(x)$，则加速公式为

$$\begin{cases} \bar{x}_{k+1} = x_k + f(x_k) \\ x_{k+1} = \bar{x}_{k+1} + \dfrac{L}{1-L}(\bar{x}_{k+1} - x_k) \end{cases}$$

记 $M = L - 1$,则上两式可以合并为

$$x_{k+1} = x_k - \frac{f(x_k)}{M} \tag{3.7}$$

此式称为简化的 Newton 公式,而其相应的迭代函数为

$$g(x) = x - \frac{f(x)}{M}$$

这里,L 是 $g'(x)$的估计值,而 $g(x) = f(x) + x$,故 $M = L - 1$ 实际上是 $f'(x)$的估计值．如果用 $f'(x)$代替式(3.7)中的 M,就得到了如下形式的迭代函数:

$$g(x) = x - \frac{f(x)}{f'(x)}$$

其相应的迭代公式为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

此即著名的 Newton(牛顿)公式．

也可以从另一角度导出牛顿公式．对于方程 $f(x) = 0$,设它的近似根为 x_k,函数 $f(x)$可以在 x_k 附近作 Taylor(泰勒)展开:

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \cdots$$

利用其线性部分作为函数 $f(x)$的近似,并求其根,求出的根记为 x_{k+1},即

$$f(x_k) + f'(x_k)(x - x_k) = 0$$

$$x = x_k - \frac{f(x_k)}{f'(x_k)} \xrightarrow{\text{记为}} x_{k+1}$$

可见相应的迭代函数就是

$$g(x) = x - \frac{f(x)}{f'(x)}$$

即牛顿公式．

下面我们来看牛顿法的几何意义.

方程 $f(x)=0$ 的根 x^*,即为曲线 $y=f(x)$与 x 轴的交点的横坐标.设 x_k 是 x^* 的某个近似值,过曲线 $y=f(x)$上相应的点 P_k $(x_k,f(x_k))$作切线.并将该切线与 x 轴的交点的横坐标 x_{k+1}作为 x^* 的新的近似值,注意到此切线方程即为

$$y=f(x_k)+f'(x_k)(x-x_k)$$

它与 x 轴的交点的横坐标(即令 $y=0$,解出 x)正是牛顿公式计算的结果.因此,牛顿法也称为切线图(见图 3-1).它的基本思想就是将非线性方程 $f(x)=0$ 一次次地用切线方程来求解.

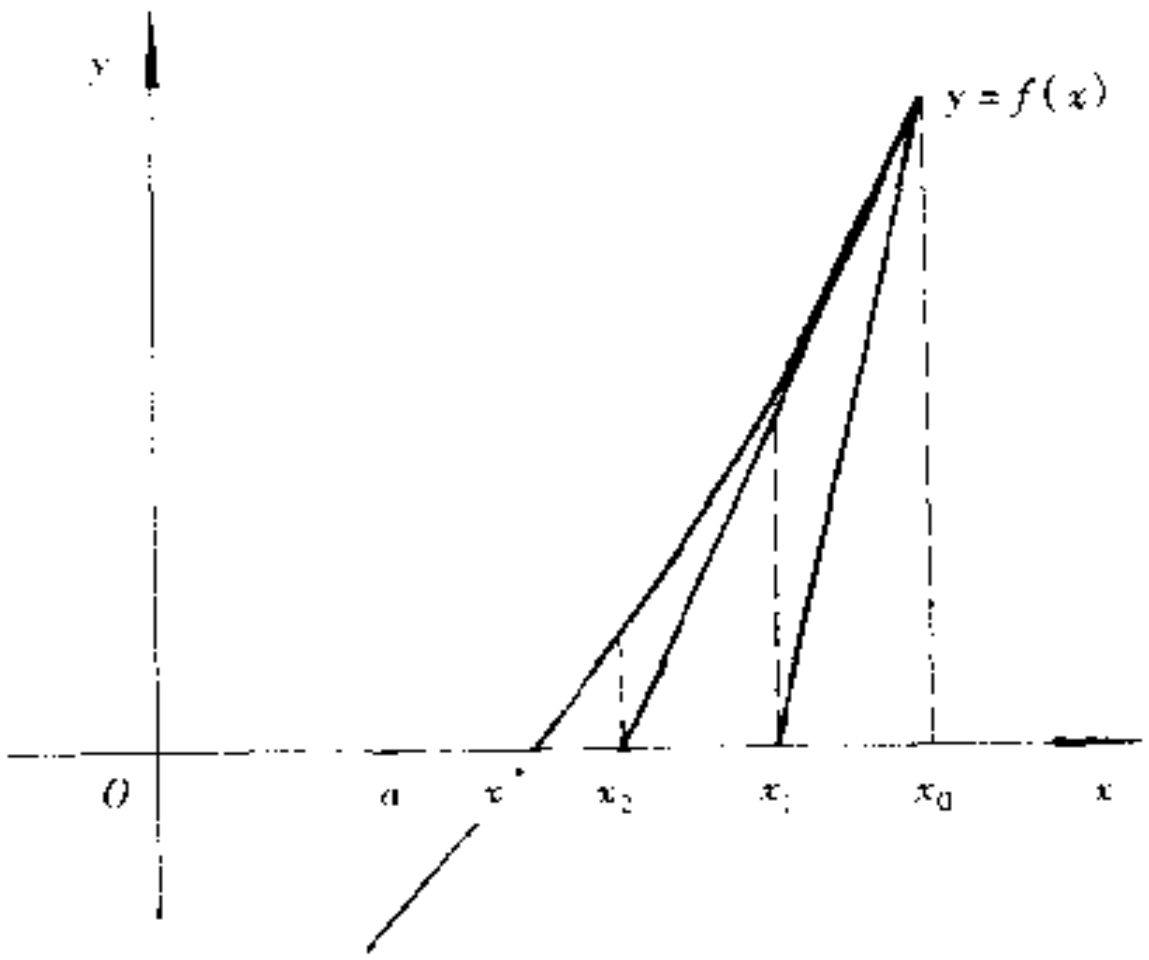


图 3-1

牛顿法的局部收敛性:

由迭代函数 $g(x)=x-\dfrac{f(x)}{f'(x)}$,故有 $g'(x)=\dfrac{f(x)f''(x)}{[f'(x)]^2}$,可见:

(1) 牛顿法局部收敛条件为

$$|g'(x^*)|=\left|\frac{f(x^*)f''(x^*)}{[f'(x^*)]^2}\right|<1$$

(2) 若 x^* 是 $f(x)$的一个单根,即 $f(x^*)=0, f'(x^*)\neq 0$,则 $g'(x^*)=0$,若 $f''(x^*)=0$,则 $g''(x^*)\neq 0$,这时,由定理 3.3 可知,牛顿法生成的序列 $\{x_k\}$ 在 x^* 附近平方收敛,这说明牛顿法收敛很快 .

(3) 若 x^* 是 $f(x)=0$ 的 $m(m\geqslant 2)$重根时,则可证明牛顿法不是平方收敛的,这时可将迭代过程改为

$$x_{k+1}=x_k-m\frac{f(x_k)}{f'(x_k)}$$

这时,仍能使其保持平方收敛 . 这些,就不再展开讨论了 .

(4) 牛顿法的收敛是指局部收敛 . 这就是说,它对初始值的选取有一定的要求,即要 x_0 充分靠近 x^*,才能保证有平方收敛 .

例 3.3 对于给定的正数 c,应用牛顿法解二次方程 $x^2-c=0$.

解 令 $f(x)=x^2-c$,即 $f'(x)=2x$. 牛顿迭代法为

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$$

$$=x_k-\frac{x_k^2-c}{2x_k}=\frac{1}{2}\left(x_k+\frac{c}{x_k}\right)$$

这可以这样理解:x_k 是$\sqrt{c}$的近似值,$\frac{c}{\sqrt{x_k}}$也是$\sqrt{c}$的近似值,它们两者的算术平均将是$\sqrt{c}$的更好的近似 .

例 3.4 例 3.3 中的迭代公式对于任意初值皆平方收敛 .

证明 对于 $x_{k+1}=\frac{1}{2}\left(x_k+\frac{c}{x_k}\right)$

右端配方可得

$$x_{k+1}-\sqrt{c}=\frac{1}{2x_k}(x_k-\sqrt{c})^2 \tag{3.8}$$

以及

$$x_{k+1}+\sqrt{c}=\frac{1}{2x_k}(x_k+\sqrt{c})^2 \tag{3.9}$$

两式相除,得

$$\frac{x_{k+1}-\sqrt{c}}{x_{k+1}+\sqrt{c}}=\left(\frac{x_k-\sqrt{c}}{x_k+\sqrt{c}}\right)^2$$

递推可得

$$\frac{x_k-\sqrt{c}}{x_k+\sqrt{c}}=\left(\frac{x_0-\sqrt{c}}{x_0+\sqrt{c}}\right)^{2^k}$$

记 $q=\dfrac{x_0-\sqrt{c}}{x_0+\sqrt{c}}$，则 $x_k-\sqrt{c}=(x_k+\sqrt{c})q^{2^k}$

即有 $$x_k=\frac{1+q^{2^k}}{1-q^{2^k}}\sqrt{c}$$

任意初值 $x_0>0$，总有 $|q|<1$，故 $x_k\to\sqrt{c}\,(k\to\infty)$

记 $e_k=x_k-\sqrt{c}$，则由式(3.8)，可得

$$\frac{e_{k+1}}{e_k^2}=\frac{1}{2x_k}\longrightarrow\frac{1}{2\sqrt{c}}$$，即为平方收敛

例 3.5 对于给定的正数 c，应用牛顿法于方程 $\dfrac{1}{x}-c=0$，并证明：当初值 x_0 满足：$0<x_0<\dfrac{2}{c}$ 时收敛.

解 $f(x)=\dfrac{1}{x}-c$，$f'(x)=-\dfrac{1}{x^2}$.

牛顿迭代公式为

$$x_{k+1}=x_k-\frac{\frac{1}{x_k}-c}{-\frac{1}{x_k^2}}=x_k(2-cx_k)$$

记第 k 步误差为 $r_k=1-cx_k$，则

$$\begin{aligned}r_{k+1}&=1-cx_{k+1}\\&=1-cx_k(2-cx_k)\\&=1-cx_k(1+r_k)\end{aligned}$$

$$= r_k - cx_k r_k$$

$$= r_k(1 - cx_k) = r_k^2$$

迭代可得 $$r_k = r_0^{2^k}$$

若 $0 < x_0 < \frac{2}{c}$，则 $-1 < r_0 = 1 - cx_0 < 1$

即 $|r_0| < 1$，则有 $r_0^{2^k} \longrightarrow 0 \quad (k \to \infty)$

即 $r_k \to 0$

迭代收敛．

现在我们讨论牛顿法的一种重要变形——牛顿下山法．牛顿法的收敛与初值 x_0 的选取有很大关系．若 x_0 偏离 x^* 较远，就有可能发散．

例如，对于 $x^3 - x - 1 = 0$，牛顿迭代法为 $x_{k+1} = x_k - \frac{x_k^3 - x_k - 1}{3x_k^2 - 1}$，若选 $x_0 = 1.5$，则 $x_1 = 1.34783$，$x_2 = 1.32520$，$x_3 = 1.32472$．迭代三次，便已有六位有效数字，但若选初值 $x_0 = 0.6$，则 $x_1 = 17.9$，离 x^* 更远，发散了．

为了防止迭代发散，我们对迭代过程再附加一个要求，即具有单调(下降)性：$|f(x_{k+1})| < |f(x_k)|$．这是因为，我们最终目标是使 $f(x^*) = 0$，故若一直保持上述下降要求，就可保证收敛．满足这个要求的算法称为下山法．我们将牛顿法与之结合使用，即在下山法保证函数值单调下降的前提下，用牛顿法加快收敛速度．为此，将牛顿法计算结果：$\bar{x}_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$，与前一步的近似值 x_k 适当加权平均，作为新的改进值：

$$x_{k+1} = \lambda \bar{x}_{k+1} + (1 - \lambda) x_k$$

$$= \lambda \left(x_k - \frac{f(x_k)}{f'(x_k)} \right) + (1 - \lambda) x_k$$

$$= x_k - \lambda \frac{f(x_k)}{f'(x_k)}, \qquad 0 < \lambda \leqslant 1$$

希望挑选下山因子 λ，使得单调性条件 $|f(x_{k+1})| < |f(x_k)|$ 成立．一般情况，从 $\lambda = 1$ 开始，反复将 λ 减半进行试算，一旦单调下降条件成立，则称下山成功．若在上述过程中，找不到使单调条件成立的下山因子，则称下山失败．此时，须另选初值 x_0．再以本例说明此方法．

$x_0 = 0.6, \bar{x}_1 = 17.9$，一直选到 $\lambda = \frac{1}{2^5} = \frac{1}{32}$ 时，得到 $x_1 = 17.9 \times \frac{1}{32} + \left(1 - \frac{1}{32}\right) \times 0.6 = 1.140625$．这时纠正了 $\bar{x}_1$ 的严重偏差．

3.6 弦截法

牛顿法突出的优点是收敛速度快，但它也有明显的缺点：公式中带有导数，当 $f(x)$ 较复杂时，使用牛顿法不太方便．代替导数的常用方法是用差商．

若用 $\frac{f(x_k) - f(x_0)}{x_k - x_0} \approx f'(x_k)$，则牛顿迭代即为

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_0)}(x_k - x_0) \tag{3.10}$$

这里，x_{k+1} 即为弦 $\overline{P_0 P_k}$ 与 x 轴交点的横坐标．因为 $\overline{P_0 P_k}$ 的方程为 $y - f(x_k) = \frac{f(x_k) - f(x_0)}{x_k - x_0}(x - x_k)$，令 $y = 0$，即得式(3.10)．

迭代格式(3.10)在迭代时，x_0 固定不变，x_k 变化：$x_k, x_{k+1}, \cdots, \to x^*$，故又称单点弦截法．

单点弦截法的收敛性：考察迭代函数

$$g(x) = x - \frac{f(x)}{f(x) - f(x_0)}(x - x_0)$$

$$g'(x^*) = 1 - \frac{f'(x^*)[f(x^*) - f(x_0)] - 0}{[f(x^*) - f(x_0)]^2}(x^* - x_0) - 0$$

$$= 1 + \frac{f'(x^*)}{f(x_0)}(x^* - x_0)$$

$$= 1 - \frac{f'(x^*)}{\dfrac{f(x^*) - f(x_0)}{x^* - x_0}}.$$

当 x_0 充分接近 x^* 时，$\dfrac{f(x^*) - f(x_0)}{x^* - x_0}$ 充分接近 $f'(x^*)$，分子与分母这两项同号，且比值 $\to 1$（当 x_0 充分接近 x^* 时），故有 $0 < |g'(x^*)| < 1$，由 3.2 节定理 3.3，可知为线性收敛 .

为了使收敛速度加快，我们用差商 $\dfrac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$ 近似代替公式中的 $f'(x_k)$，则得到

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1})$$

这个 x_{k+1} 即为弦 $\overline{P_{k-1}P_k}$ 与 x 轴交点的横坐标，故也称双点弦截法 .可以证明，它的收敛速度是超线性的 .

注 （1）双点弦截法与牛顿法虽然都是线性化方法，但两者有本质区别 . 切线法是单步法，而双点弦截法使用时，必须给出两个初始值 x_0, x_1 .

（2）我们也可从另一角度导出双点弦截法，即用插值多项式的方法：设 x_{k-1}, x_k 是 $f(x) = 0$ 的近似根，则利用 $f(x_{k-1}), f(x_k)$ 可以构造一次插值多项式（牛顿插值型）：

$$P_1(x) = f(x_k) + f[x_k, x_{k-1}](x - x_k)$$

$$= f(x_k) + \frac{f(x_k) - f(x_{k-1})}{x_k - x_k - 1}(x - x_k) \qquad (3.11)$$

把 $P_1(x)$ 作为 $f(x)$ 的近似，即把 $P_1(x) = 0$ 的根作为 $f(x) = 0$ 的近似根，令式（3.11）右面 $= 0$，解出 λ，记为 x_{k+1}，即得双点弦截法

的公式．

(3) 用同样的思想，若把 x_k, x_{k-1}, x_{k-2} 作为 $f(x)=0$ 的近似根．则由三点及其函数值，可作二次插值多项式 $P_2(x)$．并令其 $=0$，解出 x，作为 $f(x)=0$ 的近似根，这样得到的迭代格式称为抛物线法．

3.7 解非线性方程组的迭代法简介

对于形如

$$\begin{cases} f_1(x_1, x_2, \cdots, x_n) = 0 \\ f_2(x_1, x_2, \cdots, x_n) = 0 \\ \cdots\cdots \\ f_n(x_1, x_2, \cdots, x_n) = 0. \end{cases} \tag{3.12}$$

的非线性方程组，其中 $f_i(i=1,2,\cdots,n)$ 是定义在域 $D \subset R^n$ 上的 n 个自变量 $x_1, \cdots, x_n$ 的实值函数．f_i 中至少有一个是非线性的，$n \geqslant 2$．为了讨论方便，引进向量记号．令

$$\boldsymbol{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \boldsymbol{F}(\boldsymbol{X}) = \begin{pmatrix} f_1(\boldsymbol{X}) \\ \vdots \\ f_n(\boldsymbol{X}) \end{pmatrix}, \quad \boldsymbol{O} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

则方程组(3.12)就可改写为

$$\boldsymbol{F}(\boldsymbol{X}) = \boldsymbol{O} \tag{3.13}$$

这里，$\boldsymbol{F}(\boldsymbol{X})$ 是定义在某区域 $D \subset R^n$ 上的向量值函数．如果存在 $\boldsymbol{X}^* \in D$，使 $\boldsymbol{F}(\boldsymbol{X}^*) = \boldsymbol{O}$，则称 $\boldsymbol{X}^*$ 为非线性方程组(3.13)的解．

先讨论解非线性方程组的一般迭代法．类似于求解非线性方程，首先将方程组(3.12)化为等价的便于迭代的方程组：

$$\begin{cases} x_1 = g_1(x_1, x_2, \cdots, x_n) \\ x_2 = g_2(x_1, x_2, \cdots, x_n) \\ \cdots\cdots \\ x_n = g_n(x_1, x_2, \cdots, x_n). \end{cases} \tag{3.14}$$

向量形式为

$$\boldsymbol{X} = \boldsymbol{g}(\boldsymbol{X}) \tag{3.15}$$

以解的某个初始近似值 $\boldsymbol{X}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)})^{\mathrm{T}} \in D$ 出发,构造向量序列$\{\boldsymbol{X}^{(k)}\}$:

$$\boldsymbol{X}^{(k+1)} = \boldsymbol{g}(\boldsymbol{X}^{(k)}) \qquad k = 0, 1, \cdots, n$$

类似于本章定理 3.1(非线性方程迭代法收敛定理),可有如下收敛性定理.

定理 3.4 令 $D = \{(x_1, \cdots, x_n) \mid a_i \leqslant x_i \leqslant b_i, i = 1, 2, \cdots, n, a_i, b_i$ 均为常数$\}$. 假设 $\boldsymbol{g}(\boldsymbol{X})$在 D 上具有连续的一阶偏导数,且

(1) 任意 $\boldsymbol{X} \in D$;总有 $\boldsymbol{g}(\boldsymbol{X}) \in D$

(2) 对每个 $g_i(i = 1, 2, \cdots, n)$均存在常数 $L < 1$,使得

$$\left| \frac{\partial g_i(\boldsymbol{X})}{\partial x_j} \right| \leqslant \frac{L}{n}, \ \boldsymbol{X} \in D; \ i, j = 1, 2, \cdots, n.$$

则对任何初值 $\boldsymbol{X}^{(0)}$,迭代向量序列收敛于唯一解 $\boldsymbol{X}^*$,且有估计式

$$\| \boldsymbol{X}^{(k)} - \boldsymbol{X}^* \|_\infty \leqslant \frac{L^K}{1 - L} \| \boldsymbol{X}^{(1)} - \boldsymbol{X}^{(0)} \|_\infty$$

例 3.6 用迭代法解非线性方程组

$$\begin{cases} 3x_1 - \cos(x_2 x_3) - \dfrac{1}{2} = 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0 \\ \mathrm{e}^{-x_1 x_2} + 20x_3 + \dfrac{10\pi - 3}{3} = 0. \end{cases}$$

解 从第 i 个方程中解出 $x_i(i = 1, 2, 3)$. 得到等价的方程组

$$\begin{cases} x_1 = \dfrac{1}{3}\cos(x_1 x_2) + \dfrac{1}{6} = g_1(x_1, x_2, x_3) \\ x_2 = \dfrac{1}{9}\sqrt{x_1^2 + \sin x_3 + 1.06} - 0.1 = g_2(x_1, x_2, x_3) \\ x_3 = -\dfrac{1}{20}e^{-x_1 x_2} - \dfrac{10\pi - 3}{60} = g_3(x_1, x_2, x_3) \end{cases}$$

可以验证，在域 $D = \{(x_1, x_2, x_3) \mid -1 \leqslant x_i \leqslant 1, i = 1,2,3\}$ 上，有

$$\left|\frac{\partial g_i(x)}{\partial x_j}\right| \leqslant 0.281,\ i, j = 1,2,3 \quad (\text{即 } L = 0.843).$$

根据定理，我们选取 $X^{(0)} = (0.1, 0.1, -0.1)^T \in D$，由上述迭代格式 $X^{(k+1)} = g(X^{(k)})$ 产生的向量序列 $\{X^{(k)}\}$ 收敛于非线性方程组 $X = g(x)$ 的解．已经求得使 $\| X^{(k+1)} - x^{(k)} \|_\infty < 10^{-5}$ 的结果列于下表：

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\| X^{(k)} - X^{(k-1)} \|_\infty$
0	0.1	0.1	−0.1	
1	0.49998333	0.00944115	−0.52310127	0.423
2	0.49999593	0.00002557	−0.52336331	9.4×10^{-3}
3	0.50000000	0.00001234	−0.52359814	2.3×10^{-4}
4	0.50000000	0.00000003	−0.52359847	1.2×10^{-5}
5	0.50000000	0.00000002	−0.52359877	3.1×10^{-7}

实际上，方程组的精确解为

$$X^* = \left(0.5, 0, -\frac{\pi}{6}\right)^T \approx (0.5, 0, -0.523598775)^T$$

真正误差是

$$\| X^{(5)} - X^* \|_\infty \leqslant 2 \times 10^{-8}$$

注 类似于解线性方程组的 GS 方法，在解非线性方程组 X

$= g(X)$的迭代过程中．如果用已经算出的最新分量 $x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}$代替 $x_1^{(k)}, \cdots, x_{i-1}^{(k)}$来计算 $x_i^{(k+1)}$，就有可能加速收敛．

像一个方程的牛顿迭代法一样．对于方程组 $F(X) = O$，我们也采用逐次线性化的方法．

将函数 $F(X)$在某个近似根 $X^{(k)}$处泰勒展开到一次项，而有

$$F(X) \approx F(X^{(k)}) + F(X^{(k)})(X - X^{(k)})$$

令右端项等于 O 来近似代替方程组 $F(X) = O$

$$F(X^{(k)}) + F'(X^{(k)})(X - X^{(k)}) = O \tag{3.16}$$

式中
$$F'(X^{(k)}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}_{|X = X^{(k)}}$$

称为雅可比矩阵．若此矩阵非奇异，则方程组(3.16)有唯一解，记为

$$X^{(k+1)} = X^{(k)} - [F'(X^{(k)})]^{-1} \cdot F(X^{(k)}) \tag{3.17}$$

这里，$F(X) = O$ 化为等价的迭代形式，$X = g(X)$，其中

$$g(X) = X - [F'(X)]^{-1} \cdot F(X).$$

与本章定理 3.3 相类似，有以下定理：

定理 3.5 设 X^* 为 $X = g(X)$的解，如果存在常数 $\delta > 0$，使

(1) 在 X^* 的邻域 $N_\delta = \{X \mid \| X - X^* \| < \delta\}$ 上 $\frac{\partial_i(x)}{\partial x_j}(1 \leqslant i, j \leqslant n)$是连续的；

(2) $\frac{\partial^2 g_i(x)}{\partial x_j \partial x_k}$连续，且 $\left| \frac{\partial^2 g_i(x)}{\partial x_j \partial x_k} \right| \leqslant M, X \in N_\delta, 1 \leqslant i, j, k \leqslant n.$

常数 $M > 0$.

(3) $$\frac{\partial g_i(\boldsymbol{X}^*)}{\partial x_j}=0,\quad 1\leqslant i,j\leqslant n.$$

则对任何初值 $\boldsymbol{X}^{(0)}\in N_R$，由式(3.16)产生的向量序列$\{\boldsymbol{X}^{(k)}\}$平方收敛于 $\boldsymbol{X}^*$，且有

$$\|\boldsymbol{X}^{(k+1)}-\boldsymbol{X}^*\|_\infty\leqslant\frac{n^2M}{2}\|\boldsymbol{X}^{(k)}-\boldsymbol{X}^{(*)}\|_\infty^2$$

此定理表明，解非线性方程组的牛顿迭代法式(3.17)收敛很快，但需要有较好的初值，且$[\boldsymbol{F}'(\boldsymbol{X})]^{-1}$存在，否则可能不收敛.

例 3.7 用牛顿迭代法求方程组

$$\begin{cases}f_1(x_1,x_2)=x_1^2+x_2^2-5=0\\ f_2(x_1,x_2)=(x_1+1)x_2-(3x_1+1)=0\end{cases}$$

在(1,1)附近的解：

解 直接可算得雅可比矩阵

$$\boldsymbol{F}'(\boldsymbol{X})=\begin{pmatrix}\dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2}\\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2}\end{pmatrix}=\begin{pmatrix}2x_1 & 2x_2\\ x_2-3 & x_1+1\end{pmatrix}$$

$$[\boldsymbol{F}'(\boldsymbol{X})]^{-1}=\frac{1}{|\boldsymbol{F}'(\boldsymbol{X})|}\cdot\begin{pmatrix}\dfrac{\partial f_2}{\partial x_2} & -\dfrac{\partial f_1}{\partial x_2}\\ -\dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_1}{\partial x_1}\end{pmatrix}$$

$$=\frac{1}{2(x_1^2-x_2^2)+2(x_1+3x_2)}\begin{pmatrix}x_1+1 & -2x_2\\ 3-x_2 & 2x_1\end{pmatrix}$$

这时牛顿迭代公式为

$$\begin{pmatrix}x_1^{(k+1)}\\ x_2^{(k+1)}\end{pmatrix}=\begin{pmatrix}x_1^{(k)}\\ x_2^{(k)}\end{pmatrix}-\frac{1}{2[(x_1^{(k)})^2-(x_2^{(k)})^2]+2(x_1^{(k)}+3x_2^{(k)})}$$

$$\cdot\begin{pmatrix} x_1^{(k)}+1 & -2x_2^{(k)} \\ 3-x_2^{(k)} & 2x_1^{(k)} \end{pmatrix}$$

$$\cdot\begin{pmatrix} (x_1^{(k)})^2+(x_2^{(k)})^2-5 \\ (x_1^{(k)}+1)x_2^{(k)}-3(x_1^{(k)}+1) \end{pmatrix} \quad k=0,1,2,\cdots$$

取初值 $X^{(0)}=(1,1)^{\mathrm{T}}$,利用迭代格式相继算出

$$X^{(1)}=\begin{pmatrix}1.25\\2.25\end{pmatrix} \qquad X^{(2)}=\begin{pmatrix}1.0000\\2.0278\end{pmatrix}$$

原方程组精确解为 $X^*=(1,2)^{\mathrm{T}}$,可见近似解的误差向量分别为

$$e^{(0)}=(0,1)^{\mathrm{T}},\quad e^{(1)}=(-0.25,-0.25)^{\mathrm{T}},$$

$$e^{(2)}=(0,0.0278)^{\mathrm{T}}$$

且 $\quad \|e^{(0)}\|_\infty=1,\quad \|e^{(1)}\|_\infty=0.25,\quad \|e^{(2)}\|_\infty=0.0278.$

由此可见,误差减小得很快.

我们知道牛顿法平方收敛,但在具体实施时,每一步要计算雅可比矩阵 $F'(X^{(k)})$,并需求解与这个矩阵相关的 n 阶线性方程组,计算量很大,而且对初值 $X^{(0)}$也有一定要求.

为了减少计算量而引进了各种牛顿方法的变形算法,其本质就是在某种近似意义下,以矩阵 A_k 近似代替 $F'(X^{(k)})$,而 A_k 非奇异且比计算 $F'(X^{(k)})$更方便,概括为如下形式的迭代程序.

$$X^{(k+1)}=X^{(k)}-A_k^{-1}\cdot F(X^{(k)}) \quad k=1,2,\cdots \qquad (3.18)$$

式(3.18)称为拟牛顿程序.不同的 A_k 就得到了不同的变形牛顿程序.

下面简单介绍 Broyden(波罗登)算法.设非线性方程组 $F(X)=O$ 的解 X^* 的初始近似为 $X^{(0)}$,像牛顿法那样,用相同的算法计算下一个近似 $X^{(1)}$. 由 $F(X)$的泰勒展开式有

$$F(X^{(0)})\approx F(X^{(1)})+F'(X^{(1)})(X^{(0)}-X^{(1)})$$

且当 $\|X^{(0)}-X^{(1)}\|_2$ 越小,近似程度越好.因此,若以 A_1 表示 $F'(X^{(1)})$的近似,则只需取

$$\boldsymbol{A}_1(\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)})=\boldsymbol{F}(\boldsymbol{X}^{(1)})-\boldsymbol{F}(\boldsymbol{X}^{(0)}) \tag{3.19}$$

常称式(3.19)为拟牛顿方程．它是各种拟牛顿方法的出发点．注意到这样的 $\boldsymbol{A}_1$ 不能完全确定．为确定 $\boldsymbol{A}_1$，需附加条件，令 $\boldsymbol{A}_1$ 满足

$$\boldsymbol{A}_1\boldsymbol{Z}=\boldsymbol{F}'(\boldsymbol{X}^{(0)})\boldsymbol{Z},\quad \text{其中}(\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)})^{\mathrm{T}}\cdot\boldsymbol{Z}=\boldsymbol{O} \tag{3.20}$$

由此解得

$$\boldsymbol{A}_1=\boldsymbol{F}'(\boldsymbol{X}^{(0)})+\frac{[\boldsymbol{F}(\boldsymbol{X}^{(1)})-\boldsymbol{F}(\boldsymbol{X}^{(0)})-\boldsymbol{F}(\boldsymbol{X})^{(0)}(\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)})](\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)})^{\mathrm{T}}}{\|\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)}\|_2^2}$$

这里要求 $\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)}\neq\boldsymbol{O}$；若 $\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)}=\boldsymbol{O}$，则取 $\boldsymbol{A}_1=\boldsymbol{F}'(\boldsymbol{X}^{(0)})$.

利用 $\boldsymbol{A}_1$ 代替 $\boldsymbol{F}'(\boldsymbol{X}^{(1)})$，从而确定 $\boldsymbol{X}^{(2)}$ 为

$$\boldsymbol{X}^{(2)}=\boldsymbol{X}^{(1)}-\boldsymbol{A}_1^{-1}\cdot\boldsymbol{F}(\boldsymbol{X}^{(1)})$$

仿此，利用 $\boldsymbol{A}_1$ 代替 $\boldsymbol{A}_0=\boldsymbol{F}'(\boldsymbol{X}^{(0)})$，$\boldsymbol{X}^{(2)}$，$\boldsymbol{X}^{(1)}$ 分别代替 $\boldsymbol{X}^{(1)}$，$\boldsymbol{X}^{(0)}$，这个算法能确定 $\boldsymbol{X}^{(3)}$．一般情况，由 $\boldsymbol{X}^{(k)}$ 可以确定 $\boldsymbol{X}^{(k+1)}$.

$$\begin{cases}\boldsymbol{A}_k=\boldsymbol{A}_{k-1}+\dfrac{(\boldsymbol{y}_k-\boldsymbol{A}_{k-1}\boldsymbol{S}_k)\boldsymbol{S}_k^{\mathrm{T}}}{\boldsymbol{S}_k^{\mathrm{T}}\boldsymbol{S}_k} & (3.21)\\ \boldsymbol{X}^{(k+1)}=\boldsymbol{X}^{(k)}-\boldsymbol{A}_K^{-1}\cdot\boldsymbol{F}(\boldsymbol{X}^{(k)})\quad k=1,2,\cdots & (3.22)\end{cases}$$

其中，$\boldsymbol{y}_k=\boldsymbol{F}(\boldsymbol{X}^{(k)})-\boldsymbol{F}(\boldsymbol{X}^{(k-1)})$，$\boldsymbol{S}_k=\boldsymbol{X}^{(k)}-\boldsymbol{X}^{(k-1)}$．按照式(3.21)和式(3.22)算得迭代向量序列$\{\boldsymbol{X}^{(k)}\}$，这就是波罗登算法．这个算法需计算 n 个函数值(即计算 $\boldsymbol{F}(\boldsymbol{X}^{(k)})$)，比原来需计算 n^2+n 个函数值大大减少了．但仍需解 n 阶线性方程组 $\boldsymbol{A}_k(\boldsymbol{X}-\boldsymbol{X}^{(k)})=-\boldsymbol{F}(\boldsymbol{X}^{(k)})$．一个可供改进的格式是使用 Sherman(希曼)和 Morrison(莫利松)的矩阵求逆公式．这个结果可以表述如下：如果 $\boldsymbol{A}$ 是非奇异阵且 $\boldsymbol{X}$ 是向量，当 $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{X}\neq-1$ 时，则 $\boldsymbol{A}+\boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}}$ 是非奇异阵，且逆阵为

$$(\boldsymbol{A}+\boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}})^{-1}=\boldsymbol{A}^{-1}-\frac{\boldsymbol{A}^{-1}\boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}^{-1}}{1+\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}} \tag{3.23}$$

利用式(3.23)可以直接从 A_{k-1}^{-1} 计算 A_k^{-1}. 取 $A = A_{k-1}$, $X = (Y_k - A_{k-1}S_k)/\| S_k \|_2^2$ 以及 $Y = S_k$, 则有

$$A_k^{-1} = A_{k-1}^{-1} - \frac{(A_{k-1}^{-1}Y_k - S_k)S_k^{\mathrm{T}}A_{k-1}^{-1}}{S_k^{\mathrm{T}}A_{k-1}^{-1}Y_k}$$

因此,用波罗登方法,式(3.21)和式(3.22)可以改写成

$$\begin{cases} A_0 = F(X^{(0)}) \\ A_k^{-1} = A_{k-1}^{-1} + \dfrac{(S_k - A_k^{-1}Y_k)S_k^{\mathrm{T}}A_{k-1}^{-1}}{S_k^{\mathrm{T}}A_{k-1}^{-1}Y_k} \\ X^{(k+1)} = X^{(k)} - A_k^{-1} \cdot F(X^{(k)}) \quad k = 1,2,\cdots \end{cases} \tag{3.24}$$

每重复一步,计算逆阵约需 $O(n^2)$ 次算术运算. 且避免了计算矩阵 A_k, 仅需计算 n 个函数值.

例 3.8 利用波罗登方法解非线性方程组

$$\begin{cases} 3x_1 - \cos(x_2x_3) - \dfrac{1}{2} = 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0 \\ e^{-x_1x_2} + 20x_3 + \dfrac{10\pi - 3}{3} = 0 \end{cases}$$

解 雅可比矩阵为

$$F'(X) = \begin{pmatrix} 3 & x_3\sin(x_2x_3) & x_2\sin(x_2x_3) \\ 2x_1 & -162(x_2 + 0.1) & \cos x_3 \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{pmatrix}$$

取 $X^{(0)} = (0.1, 0.1, -0.1)^{\mathrm{T}}$, 有

$$F'(X^{(0)}) = \begin{pmatrix} 3 & 9.999836 \times 10^{-4} & -9.999836 \times 10^{-4} \\ 0.2 & -323.9999 & 0.995041 \\ -9.900498 \times 10^{-2} & -9.900498 \times 10^{-2} & 20 \end{pmatrix}$$

$$A_0^{-1} = [F'(X^{(0)})]^{-1}$$

$$= \begin{pmatrix} 0.3333331 & 1.023852\times10^{-5} & 1.615703\times10^{-5} \\ 2.108606\times10^{-3} & -3.086882\times10^{-2} & 1.535838\times10^{-3} \\ 1.660552\times10^{-3} & 1.527579\times10^{-4} & 5.000774\times10^{-2} \end{pmatrix}$$

$$\boldsymbol{X}^{(1)} = \boldsymbol{X}^{(0)} - \boldsymbol{A}_0^{-1}\boldsymbol{F}(\boldsymbol{X}^{(0)}) = \begin{pmatrix} 0.4998693 \\ 1.946693\times10^{-2} \\ -0.5215209 \end{pmatrix}$$

$$\boldsymbol{F}(X^{(1)}) = \begin{pmatrix} -3.404021\times10^{-4} \\ -0.3443899 \\ 3.18737\times10^{-2} \end{pmatrix}$$

$$\boldsymbol{Y}_1 = \boldsymbol{F}(\boldsymbol{X}^{(1)}) - \boldsymbol{F}(\boldsymbol{X}^{(0)})$$

$$= \begin{pmatrix} 1.999608 \\ 1.925442 \\ -8.430152 \end{pmatrix}$$

$$\boldsymbol{S}_1^{\mathrm{T}}\boldsymbol{A}_0^{-1}\boldsymbol{Y}_1 = 0.3424604$$

$$\boldsymbol{A}_1^{-1} = \boldsymbol{A}_0^{-1} + (1/0.3424604)[(\boldsymbol{S}_1 - \boldsymbol{A}_0^{-1}\boldsymbol{Y}_1)\boldsymbol{S}_1^{\mathrm{T}}\boldsymbol{A}_0^{-1}]$$

$$= \begin{pmatrix} 0.3333781 & 1.11077\times10^{-5} & 8.944584\times10^{-4} \\ -2.021271\times10^{-3} & -3.094847\times10^{-2} & 2.196909\times10^{-3} \\ 1.022381\times10^{-3} & -1.650679\times10^{-4} & 5.010987\times10^{-2} \end{pmatrix}$$

且

$$\boldsymbol{X}^{(2)} = \boldsymbol{X}^{(1)} - \boldsymbol{A}_1^{-1}\cdot\boldsymbol{F}(\boldsymbol{X}^{(1)}) = \begin{pmatrix} 0.4999863 \\ 8.737888\times10^{-3} \\ -0.5231746 \end{pmatrix}$$

其他结果见下表：

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\| X^{(k)} - X^{(k-1)} \|_2$
3	0.5000066	8.672215×10^{-4}	−0.5236918	7.88×10^{-3}
4	0.5000005	6.087473×10^{-5}	−0.5235954	8.12×10^{-4}
5	0.5000002	-1.445223×10^{-6}	−0.5235989	6.24×10^{-5}

习 题 3

1. 用二分法求方程 $x^2 - x - 1 = 0$ 的正根，要求准确到小数点后第一位 .

2. 为求方程 $x^3 - x^2 - 1 = 0$ 在 $x_0 = 1.5$ 附近的一个根，设将方程改写成下列等价形式，并建立相应的迭代公式：

(1) $x = 1 + \frac{1}{x^2}$　迭代公式　$x_{k+1} = 1 + \frac{1}{x_k^2}$

(2) $x^3 = 1 + x^2$　迭代公式　$x_{k+1} = \sqrt[3]{1 + x_k^2}$

(3) $x^2 = \frac{1}{x - 1}$　迭代公式　$x_{k+1} = \frac{1}{\sqrt{x_k - 1}}$

试分析每种迭代公式的收敛性，并选取一种公式求出具有四位有效数字的近似值 .

3. 给定函数 $f(x)$，设对一切 x，$f'(x)$存在且 $0 < m \leqslant f'(x) \leqslant M$，证明对于任意 $\lambda \in \left(0, \frac{2}{M}\right)$，迭代格式 $x_{k+1} = x_k - \lambda f(x_k)$均收敛，并确定使迭代收敛得最快的值 λ.

4. 设 $\varphi(x)$在$[a, b]$上连续可微，且 $0 \leqslant \varphi'(x) < 1$，$x = \varphi(x)$在$[a, b]$上有根 α，$x_0 \in [a, b]$，但 $x_0 \neq \alpha$，证明由 $x_{k+1} = \varphi(x_k)$产生的迭代序列$\{x_k\}$单调收敛于 α；

5. 已知 $x = \varphi(x)$在$[a, b]$内只有一根，而当 $a < x < b$ 时，

$|\varphi'(x)| \geqslant k > 1$,试问如何将 $x = \varphi(x)$化为适合迭代的形式? 将 $x = \tan x$ 化为适合迭代的形式,并求 $x = 4.5$(弧度)附近的根 .

6. 能否用迭代法解下列各方程,如果不能时,试将方程改写成能用迭代法求解的形式:

(1) $x = \dfrac{1}{4}(\cos x + \sin x)$;

(2) $x = 4 - 2^x$.

7. 用牛顿法解下列方程(算到 x_4 即可).

(1) $x^3 - 3x^2 - x + 9 = 0$,取 $x_0 = -2$;

(2) $x^3 + 2x^2 + 10x - 20 = 0$,取 $x_0 = 1$.

8. 就下列函数讨论牛顿法的收敛性和收敛速度:

(1) $f(x) = \begin{cases} \sqrt{x}, & x \geqslant 0, \\ -\sqrt{-x}, & x < 0; \end{cases}$

(2) $f(x) = \begin{cases} \sqrt[3]{x^2}, & x \geqslant 0, \\ -\sqrt[3]{x^2}, & x < 0. \end{cases}$

9. 证明迭代公式 $x_{k+1} = \dfrac{x_k(x_k^2 + 3a)}{3x_k^2 + a}$是计算$\sqrt{a}$的三阶方法;假定初值 x_0 充分靠近根 x^*,求 $\lim\limits_{k \to \infty} \dfrac{\sqrt{a} - x_{k+1})}{(\sqrt{a} - x_k)^3}$.

10. 分别用牛顿法,单点弦截法,双点弦截法求方程 $4\cos x = e^x$ 的根,要求精度 $\varepsilon = 10^{-4}$.

11. 用简单迭代法解方程组:

$$\begin{cases} x = \sin(x + y) \\ y = \cos(x - y) \end{cases}$$

取 $x_0 = y_0 = 0$.

12. 用牛顿法解下列方程组:

(1) $\begin{cases} x^2 + y^2 = 4 \\ x^2 - y^2 = 1 \end{cases}$ 在(1.6,1.2)附近的根;

(2) $\begin{cases} 4x^2 + y^2 - 4 = 0 \\ x + y - \sin(x - y) = 0 \end{cases}$ 在(1.0,0.0)附近的根 .

13. 用牛顿法求解方程组

$$\begin{cases} x = \frac{1}{2}\cos y \\ y = \frac{1}{2}\sin x \end{cases}$$

取初值 $x_0 = y_0 = 0$. 证明对于所有的 x, y, $[F']^{-1}$存在,其中 F' 为雅可比矩阵 .

14. 求函数 $g(X) = X - [F'(X)]^{-1} \cdot F(x)$的雅可比矩阵 . 其中 $X, F(X) \in R^n$. $F'(X)$为 $F(X)$的雅可比矩阵 .

15. 用波罗登方法的式(3.23)求方程组

$$\begin{cases} 4x_1^2 + x_2^2 + 2x_1x_2 - x_2 - 2 = 0 \\ 2x_1^2 + 3x_1x_2 + x_2^2 - 3 = 0 \end{cases}$$

的解,取 $X^{(0)} = (0.4, 0.9)^T$,迭代到 $\| X^{(k)} - X^{(k-1)} \| \leqslant \frac{1}{2} \times 10^{-5}$为止 .

第四章　矩阵特征值与特征向量的计算

4.1　引　言

在实践中，有很多问题（如振动问题），物理学中某些临界值的确定以及理论物理中的一些问题，归结为数学问题就是：

（1）已知矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}$，要求代数方程 $\varphi(\lambda)=\det(\lambda\boldsymbol{I}-\boldsymbol{A})=0$ 的根，$\varphi(\lambda)$称为 $\boldsymbol{A}$ 的特征多项式，$\varphi(\lambda)$的根就称为 $\boldsymbol{A}$ 的特征值，即求 $\boldsymbol{A}$ 的特征值．

（2）设 λ 是$\boldsymbol{A}$ 的特征值，要求齐次线性方程组$(\lambda\boldsymbol{I}-\boldsymbol{A})\boldsymbol{X}=\boldsymbol{O}$ 的非零解 $\boldsymbol{X}$，此非零解 $\boldsymbol{X}$ 就称为$\boldsymbol{A}$ 的对应于λ 的特征向量．

有些实际问题只需求出按模最大的特征值和相应的特征向量．有的要求出全部特征值和相应的特征向量，对于不同的要求，我们将选择不同的方法．

叙述一些与特征值有关的结论．

定理 4.1　若 $\lambda_i(i=1,2,\cdots,n)$是矩阵 $\boldsymbol{A}$ 的特征值，则有

（1）$\sum\limits_{i=1}^{n}\lambda_i=\sum\limits_{i=1}^{n}a_{ii}=\mathrm{tr}(\boldsymbol{A})$

（2）$\det(\boldsymbol{A})=\lambda_1\lambda_2\cdots\lambda_n$．

定理 4.2　若 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 相似，则 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 有相同的特征值．

定理 4.3　（圆盘定理）设 $\boldsymbol{A}=(a_{ij})_n$，则 $\boldsymbol{A}$ 的每一个特征值要属于下述某个圆盘之中：$|\lambda-a_{ii}|\leqslant\sum\limits_{j=1}^{n}|a_{ij}|(i=1,2,\cdots,n)$．

证明　设 λ 是$\boldsymbol{A}$ 的任一特征值，$\boldsymbol{X}$ 为对应的特征向量，即

$$(\lambda\boldsymbol{I}-\boldsymbol{A})\boldsymbol{X}=\boldsymbol{O}. \tag{4.1}$$

因为 $\boldsymbol{X}$ 是特征向量，故 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\neq 0$.

记 $|x_i| = \max_j |x_j| \neq 0$，则方程组式(4.1)中的第 i 个方程为

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + (\lambda - a_{ii})x_i + a_{i\,i+1}x_{i+1} + \cdots + a_{in}x_n = 0$$

即 $(\lambda - a_{ii})x_i = -(a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{i\,i-1}x_{i-1} + a_{i\,i+1}x_{i+1} + \cdots + a_{in}x_n)$

即有 $|\lambda - a_{ii}| \cdot |x_i| \leqslant |a_{i1}| \cdot |x_1| + \cdots + |a_{i\,i-1}| \cdot |x_{i-1}| + |a_{i\,i+1}| \cdot |x_{i+1}| + \cdots + |a_{in}| \cdot |x_n|$

又由 $$|x_j| \leqslant |x_i| \qquad (j \neq i)$$

则得 $$|\lambda - a_{ii}| \leqslant \sum_{j \neq i} |a_{ij}| \cdot \left|\frac{x_j}{x_i}\right| \leqslant \sum_{j \neq i} |a_{ij}|$$

这就是说，特征值 λ 一定落在某个圆盘中．那个圆盘就是特征向量 $\boldsymbol{X}$ 中的分量按模最大的下标 i 所对应的那个圆盘，这个圆盘是以复平面上以 a_{ii} 为圆心，以 $\sum_{j \neq i} |a_{ij}|$ 为半径．

定义 4.1 瑞雷(Rayleigh)商　设 $\boldsymbol{A}$ 是 n 阶实对称阵，对于任一非零向量 $\boldsymbol{x}$，称 $R(\boldsymbol{x}) = \dfrac{(\boldsymbol{Ax}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})}$ 为关于向量 $\boldsymbol{x}$ 的瑞雷商．

定理 4.4 设 $\boldsymbol{A}$ 为 n 阶实对称阵，对于任一非零向量 $\boldsymbol{x}$，则

$$\lambda_1 \geqslant \frac{(\boldsymbol{Ax}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \geqslant \lambda_n,\ \lambda_1 = \max_{\substack{x \in R^n \\ x \neq 0}} \frac{(\boldsymbol{Ax}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})},\quad \lambda_n = \min_{\substack{x \in R^n \\ x \neq 0}} \frac{(\boldsymbol{Ax}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})}.$$

其中：$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$ 为 $\boldsymbol{A}$ 的特征值．

证明　因为 $\boldsymbol{A}$ 实对称，故有完全的特征向量系．设 $x_1, x_2, \cdots, x_n$ 是 $\boldsymbol{A}$ 的规范化正交组，即 $(x_i, x_j) = \delta_{ij}$，则任一非零向量 $\boldsymbol{x}$ 皆能表示为 $\boldsymbol{x} = \sum_{i=1}^{n} \alpha_i x_i$，则

$$(\boldsymbol{x}, \boldsymbol{x}) = \left(\sum_{i=1}^{n} \alpha_i x_i,\ \sum_{i=1}^{n} \alpha_i x_i\right) = \sum_{i=1}^{n} \alpha_i^2$$

$$(\boldsymbol{Ax}, \boldsymbol{x}) = \left(\sum_{i=1}^{n} \boldsymbol{A}\alpha_i x_i,\ \sum_{i=1}^{n} \alpha_i x_i\right)$$

$$= \left(\sum_{i=1}^{n} \alpha_i \lambda_i x_i,\ \sum_{i=1}^{n} \alpha_i x_i\right)$$

$$= \sum_{i=1}^{n} \lambda_i \alpha_i^2$$

故 $$\frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}=\frac{\sum_{i=1}^{n}\lambda_i\alpha_i^2}{\sum_{i=1}^{n}\alpha_i^2}\leqslant\frac{\lambda_1\sum_{i=1}^{n}\alpha_i^2}{\sum_{i=1}^{n}\alpha_i^2}=\lambda_1$$

同理可证 $$\frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}\geqslant\lambda_n$$

在瑞需商中，分别取 $\boldsymbol{x}=x_1$ 和 $\boldsymbol{x}=x_n$，则可达到瑞雷商的最大值 λ_1 和最小值 λ_n，故

$$\lambda_1=\max_{x\in R^n,\,x\neq0}\frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})},\quad \lambda_n=\min_{x\in R^n,\,x\neq0}\frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}$$

4.2 乘幂法

乘幂法是一种计算实矩阵 $\boldsymbol{A}$ 的按模最大的特征值及其对应的特征向量的方法 .

为了便于分析，我们假设：

(1) $\boldsymbol{A}$ 有 n 个线性无关的特征向量 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$；

(2) $\boldsymbol{Ax}_i=\lambda_i\boldsymbol{x}_i,i=1,2,\cdots,n$. 其特征值满足 $|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$，且 λ_i 是实特征值 .

由于 $\boldsymbol{A}$ 具有完全的特征向量系，即 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$ 构成了 n 维向量空间 R^n 的基底 . 故对任意 $\boldsymbol{v}_0\in R^n$，均可表示成

$$\boldsymbol{v}_0=\alpha_1\boldsymbol{x}_1+\alpha_2\boldsymbol{x}_2+\cdots+\alpha_n\boldsymbol{x}_n\quad(\text{设 }\alpha_1\neq0)$$

则 $$\boldsymbol{v}_1=\boldsymbol{Av}_0=\alpha_1\lambda_1\boldsymbol{x}_1+\alpha_2\lambda_2\boldsymbol{x}_2+\cdots+\alpha_n\lambda_n\boldsymbol{x}_n$$

$$\boldsymbol{v}_2=\boldsymbol{Av}_1=\alpha_1\lambda_1^2\boldsymbol{x}_1+\alpha_2\lambda_2^2\boldsymbol{x}_2+\cdots+\alpha_n\lambda_n^2\boldsymbol{x}_n$$

一般有 $$\boldsymbol{v}_k=\boldsymbol{Av}_{k-1}=\alpha_1\lambda_1^k\boldsymbol{x}_1+\alpha_2\lambda_2^k\boldsymbol{x}_2+\cdots+\alpha_n\lambda_n^k\boldsymbol{x}_n$$

$$=\lambda_1^k\left[\alpha_1\boldsymbol{x}_1+\alpha_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\boldsymbol{x}_2+\cdots+\alpha_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\boldsymbol{x}_n\right]$$

当 k 足够大时，由于 $\left|\frac{\lambda_i}{\lambda_1}\right|<1,i=2,3,\cdots,n$

故 $$\boldsymbol{v}_k\approx\lambda_1^k\alpha_1\boldsymbol{x}_1,$$

$$v_{k+1} \approx \lambda_1^{k+1} \alpha_1 x_1 \approx \lambda_1 v_k$$

而 $$v_{k+1} = Av_k$$

故 v_k 可以近似地作为对应于 λ_1 的特征向量 .

现在,我们讨论特征值 λ_1 的计算 .

我们用 $(v_k)_j$ 表示向量 v_k 的第 j 个分量,当 k 足够大时,

$$\frac{(v_{k+1})_j}{(v_k)_j} \approx \frac{\lambda_1^{k+1}\alpha_1(x_1)_j}{\lambda_1^k \alpha_1 (x_1)_j} = \lambda_1, \ (j = 1, 2, \cdots, n)$$

考虑到有时可能由于 j 的不同,所得的 λ_1 也不同,故也可用平均值 $\frac{1}{n}\sum_{j=1}^{n} \frac{(v_{k+1})_j}{(v_k)_j}$ 来作为 λ_1 的近似值 .

例 4.1 求 $A = \begin{pmatrix} \frac{1}{4} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{6} \end{pmatrix}$ 的按模最大的特征值 .

解 取初始向量 $v_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$,计算:$v_k = Av_{k-1}$,其结果的分量表示见下表:

k	0	1	2	3	4
$(v_k)_1$	1	0.25	0.10250	0.042292	0.017451
$(v_k)_2$	0	0.20	0.08333	0.034389	0.014190

可见:$\frac{(v_2)_1}{(v_1)_1} = 0.4 \quad \frac{(v_2)_2}{(v_1)_2} = 0.41666 \quad \frac{(v_3)_1}{(v_2)_1} = 0.41260$

$$\frac{(v_3)_2}{(v_2)_2} = 0.41259 \quad \frac{(v_4)_1}{(v_3)_1} = 0.41263 \quad \frac{(v_4)_2}{(v_3)_2} = 0.41263$$

相应分量的比值很快趋于同一值,故可取 $\lambda_1 \approx 0.41263$. 而相应的特征向量可取为 $x_1 = v_4 = (0.017451, 0.014190)^T$,$\lambda_1$ 的真值为 0.41262,可见已准确到小数点以后五位 .

问题：由 $\boldsymbol{v}_k \approx \lambda_1^k \alpha_1 \boldsymbol{x}_1$. 当 $|\lambda_1| < 1$ 时，$\boldsymbol{v}_k$ 的分量随 k 的增大而趋于零；当 $|\lambda_1| > 1$ 时，各分量要趋于无穷，这在计算机里即所谓“下溢”和“上溢”。为了避免这种情况，我们对向量 $\boldsymbol{v}_k$ 采取“规范化”措施，即对于 $\boldsymbol{v}_k$，令

$$\boldsymbol{u}_k = \frac{\boldsymbol{v}_k}{\max(\boldsymbol{v}_k)}, \quad k = 0,1,2,\cdots$$

这里，$\max(\boldsymbol{v}_k)$表示 $\boldsymbol{v}_k$ 中取绝对值最大的分量．通常取 $\boldsymbol{v}_0 = \boldsymbol{u}_0 \neq 0$，此时 $\boldsymbol{v}_0$ 已是规范化的，即它的分量的最大模不超过 1（常取 $\boldsymbol{v}_0 = (1,1,\cdots,1)^{\mathrm{T}}$）.

$$\boldsymbol{v}_1 = \boldsymbol{A}\boldsymbol{u}_0$$

$$\boldsymbol{u}_1 = \frac{\boldsymbol{v}_1}{\max(\boldsymbol{v}_1)} = \frac{\boldsymbol{A}\boldsymbol{u}_0}{\max(\boldsymbol{A}\boldsymbol{u}_0)},$$

$$\boldsymbol{v}_2 = \boldsymbol{A}\boldsymbol{u}_1 = \frac{\boldsymbol{A}^2\boldsymbol{u}_0}{\max(\boldsymbol{A}\boldsymbol{u}_0)},$$

$$\boldsymbol{u}_2 = \frac{\boldsymbol{v}_2}{\max(\boldsymbol{v}_2)} = \frac{\boldsymbol{A}^2\boldsymbol{u}_0}{\max(\boldsymbol{A}^2\boldsymbol{u}_0)},$$

$$\cdots,$$

$$\boldsymbol{v}_k = \boldsymbol{A}\boldsymbol{u}_{k-1} = \frac{\boldsymbol{A}^k\boldsymbol{u}_0}{\max(\boldsymbol{A}^{k-1}\boldsymbol{u}_0)},$$

$$\boldsymbol{u}_k = \frac{\boldsymbol{A}^k\boldsymbol{u}_0}{\max(\boldsymbol{A}^k\boldsymbol{u}_0)} = \frac{\boldsymbol{v}_k}{\max(\boldsymbol{v}_k)}$$

由于 $\boldsymbol{A}^k\boldsymbol{v}_0 = \boldsymbol{v}_k = \lambda_1^k\left(\alpha_1\boldsymbol{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k \alpha_2\boldsymbol{x}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k \alpha_n\boldsymbol{x}_n\right)$

故 $$\boldsymbol{u}_k = \frac{\lambda_1^k\left(\alpha_1\boldsymbol{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k \alpha_2\boldsymbol{x}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k \alpha_n\boldsymbol{x}_n\right)}{\max\left[\lambda^k\left(\alpha_1\boldsymbol{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k \alpha_2\boldsymbol{x}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k \alpha_n\boldsymbol{x}_n\right)\right]}$$

$\lim\limits_{k\to 0}\boldsymbol{u}_k = \dfrac{\boldsymbol{x}_1}{\max(\boldsymbol{x}_1)}$，表明 $\boldsymbol{u}_k$ 可近似作为对应于 λ_1 的特征向量．

另一方面，

$$v_k = Au_{k-1} = \frac{A^k u_0}{\max(A^{k-1}u_0)} = \frac{A^k v_0}{\max(A^{k-1}v_0)}$$

$$= \lambda_1 \cdot \frac{\alpha_1 x_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i}{\max\left[\alpha_1 x_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^{k-1} x_i\right]}$$

于是

$$\mu_k = \max(v_k) = \lambda_1 \frac{\max\left[\alpha_1 x_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i\right]}{\max\left[\alpha_1 x_1 + \sum_{k=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^{k-1} x_i\right]} \tag{4.2}$$

且 $$\lim_{k\to 0} \mu_k = \lambda_1 .$$

可见，当 k 足够大时，可取特征值为 $\lambda_1 = \mu_k = \max(v_k)$ 相应的特征向量为

$$x_1 = u_k = \frac{v_k}{\max(v_k)} .$$

例 4.2 用乘幂法求矩阵 A 的按模最大特征值和相应的特征向量

$$A = \begin{pmatrix} -4 & 14 & 0 \\ -5 & 13 & 0 \\ -1 & 0 & 2 \end{pmatrix}$$

解 设 $u_0 = v_0 = (1,1,1)^T$，则

$$v_1 = Au_0 = (10,8,1)^T$$

$$\mu_1 = \max(v_1) = 10$$

于是规范化向量

$$u_1 = \frac{v_1}{\mu_1} = (1,0.8,0.1)^T$$

其余计算见下表：

k	$\mu_k=\max(y_k)$	$u_k^{\mathrm{T}}=\frac{y_k^{\mathrm{T}}}{\mu_k}$	$\bar{\mu}_k$
0		(1,1,1)	
1	10	(1,0.8,0.1)	
2	7.2	(1,0.75,−0.111)	6.002473
3	6.5	(1,0.730769,−0.188034)	6.015054
4	6.230769	(1,0.722222,−0.220850)	6.004202
5	6.111	(1,0.718182,−0.235915)	6.000855
6	6.054546	(1,0.716216,−0.243095)	6.000240
7	6.027027	(1,0.715247,−0.246588)	6.000058
8	6.013453	(1,0.714765,−0.248306)	6.000017
9	6.006711	(1,0.714525,−0.249157)	6.000003
10	6.003352	(1,0.714405,−0.249579)	6.000000
11	6.001675	(1,0.714346,−0.249790)	6.000000
12	6.000837	(1,0.714316,−0.249895)	

结果表明：A 的按模最大特征值 λ_1 的近似值 $\mu_{12}=6.000837$，相应的特征向量为 $u_{12}=(1,0.714316,-0.249895)^{\mathrm{T}}$. 实际上，$A$ 的特征值分别为 6,3,2.

表中的 $\bar{\mu}_k$ 是下面所述的埃特金加速算法式(4.3)和计算结果.

乘幂法的收敛速度取决于比值 $\left|\frac{\lambda_2}{\lambda_1}\right|$，我们将着眼于减小 $\left|\frac{\lambda_2}{\lambda_1}\right|$ 的值和提高 $\left|\frac{\lambda_2}{\lambda_1}\right|^k$ 中的 k 的方次. 为减小 $\left|\frac{\lambda_2}{\lambda_1}\right|$ 的值，可采用原点移位法，为提高 k 的方次，可采用埃特金加速法和瑞雷商加速. 下面，详细介绍原点移位法.

1. 原点位移法

设矩阵 $\boldsymbol{A}$ 的特征值为 $\lambda_1 > \lambda_2 \geqslant \cdots \geqslant \lambda_n$，并有完全的特征向量系：$\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n$. 则矩阵 $\boldsymbol{B} = \boldsymbol{A} - p\boldsymbol{I}$ 的特征值就是 $\lambda_1 - p, \lambda_2 - p, \cdots, \lambda_n - p$. 而特征向量不变，仍为 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$. 我们将设法取到数 p，使 $\left|\frac{\lambda_2 - p}{\lambda_1 - p}\right| < \left|\frac{\lambda_2}{\lambda_1}\right|$，然后，对矩阵 $\boldsymbol{B} = \boldsymbol{A} - p\boldsymbol{I}$ 来做乘幂法，求得 $\boldsymbol{B}$ 的特征值 $\lambda_1(\boldsymbol{B})$，则 $\boldsymbol{A}$ 的特征值即为 $\lambda_1 = \lambda_1(\boldsymbol{B}) + p$.

如何取这个 p 呢？对于 $\boldsymbol{A}$ 的特征值 $\lambda_1 > \lambda_2 \geqslant \cdots \geqslant \lambda_n$ 来说，它的主特征值(即按模最大特征值)可能是 λ_1，也可能是 λ_n，那么，$\boldsymbol{B}$ 的主特征值可能是 $\lambda_1 - p$，也可能是 $\lambda_n - p$. 现假设 $|\lambda_1 - p| > |\lambda_n - p|$，那么，在诸数：

$$\left|\frac{\lambda_2 - p}{\lambda_1 - p}\right|, \quad \left|\frac{\lambda_3 - p}{\lambda_1 - p}\right|, \cdots, \left|\frac{\lambda_n - p}{\lambda_1 - p}\right|$$

中，只可能是 $\left|\frac{\lambda_2 - p}{\lambda_1 - p}\right|$ 或 $\left|\frac{\lambda_n - p}{\lambda_1 - p}\right|$ 最大. 我们选 p. 要使数 $\omega = \max\left\{\left|\frac{\lambda_2 - p}{\lambda_1 - p}\right|, \left|\frac{\lambda_n - p}{\lambda_1 - p}\right|\right\}$ 尽可能的小(即 = min). 显见，当 $\lambda_2 - p = -(\lambda_n - p)$ 时，即 $p = \frac{\lambda_2 + \lambda_n}{2}$ 时，数 ω 取值最小，此时，矩阵 $\boldsymbol{B}$ 按乘幂法迭代最快.

下面再说明：在 $|\lambda_1| > |\lambda_n|$ 的条件下，$0 < \frac{\lambda_2 - p}{\lambda_1 - p} < \frac{\lambda_2}{\lambda_1}$. 由于 $|\lambda_1| > |\lambda_n|$，可知 $\lambda_1 > 0$，而且此数选择的 $p = \frac{\lambda_2 + \lambda_n}{2}$ 满足 $0 < p < \lambda_n$(因为 λ_2 与 λ_1 很靠近，否则，$\left|\frac{\lambda_2}{\lambda_1}\right|$ 就已足够小，不必再找 p). 因此，

$$\frac{\lambda_2 - p}{\lambda_1 - p} - \frac{\lambda_2}{\lambda_1} = \frac{\lambda_1\lambda_2 - \lambda_2 p - \lambda_1\lambda_2 + \lambda_2 p}{\lambda_1(\lambda_1 - p)}$$

$$= \frac{(\lambda_2 - \lambda_1)p}{\lambda_1(\lambda_1 - p)} < 0$$

这种算法简单,而且对矩阵原有的稀疏性等也不破坏．但它致命的弱点是 λ_2, λ_n 一般并不知道．因为若知道了 λ_2, λ_n,通常也就知道了 λ_1,也就不必再去求 p. 因此,一般总是对 λ_2 和 λ_n 大约有个估计,粗略估计一个值 p,在计算机上试算,看迭代加速的效果如何．

2. 埃特金加速方法

由乘幂法的迭代公式及式(4.2),有

$$\mu_k = \max(y_k) = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

即当 k 充分大时,存在常数 $c > 0$,使得

$$|\mu_k - \lambda_1| \approx c\left|\frac{\lambda_2}{\lambda_1}\right|^k$$

$$|\mu_{k+1} - \lambda_1| \approx c\left|\frac{\lambda_2}{\lambda_1}\right|^{k+1}$$

于是

$$\frac{|\mu_{k+1} - \lambda_1|}{|\mu_k - \lambda_1|} \approx c\left|\frac{\lambda_2}{\lambda_1}\right|$$

即序列 $\{\mu_k\}$ 线性收敛于 λ_1,此时,也有如下近似关系式:

$$\frac{\mu_{k+2} - \lambda_1}{\mu_{k+1} - \lambda_1} \approx \frac{\mu_{k+1} - \lambda_1}{\mu_k - \lambda_1}$$

从而

$$\lambda_1 \approx \frac{\mu_k\mu_{k+2} - \mu_{k+1}^2}{\mu_k - 2\mu_{k+1} + \mu_{k+2}} = \bar{\mu}_k \tag{4.3}$$

即由已知的三个迭代值 $\mu_k, \mu_{k+1}, \mu_{k+2}$ 可以算出新的特征值估计值 $\bar{\mu}_k$,这种加速方法称为 Aitken 方法．加速效果可以在例 4.2 的表格中看到．

3. 瑞雷商加速

定理 4.5 设 A 为 n 阶实对称阵，$|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|$，对应的特征向量 $x_x, x_2, \cdots, x_n$，满足 $(x_i, x_j) = \delta_{ij}$，当应用乘幂法公式：

$$\begin{cases} v_0 = u_0 \neq 0 \\ v_k = Au_{k-1} \\ u_k = \dfrac{v_k}{\max(v_k)} \end{cases} \quad (k = 1, 2, \cdots)$$

计算 A 的主特征值 λ_1 时，规范化向量 u_k 的瑞雷商给出 λ_1 的较好的近似：

$$\frac{(Au_k, u_k)}{(u_k, u_k)} = \lambda_1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right).$$

证明

$$v_{k+1} = Au_k = \frac{A^{k+1}u_0}{\max(A^{k+1}u_0)}$$

$$= \frac{A^{k+1} \cdot (\sum_{i=1}^{n} \alpha_i x_i)}{\max(A^{k+1}u_0)} = \frac{\sum_{i=1}^{n} \alpha_i \lambda_i^{k+1} x_i}{\max(A^{k+1}u_0)}$$

$$u_k = \frac{A^k u_0}{\max(A^k u_0)} = \frac{\sum_{i=1}^{n} \alpha_i \lambda_i^k x_i}{\max(A^k u_0)},$$

故

$$\frac{(Au_k, u_k)}{(u_k, u_k)} = \frac{(\sum_{i=1}^{n} \alpha_i \lambda_i^{k+1} x_i, \sum_{i=1}^{n} \alpha_i \lambda_i^k x_i)}{(\sum_{i=1}^{n} \alpha_i \lambda_i^k x_i, \sum_{i=1}^{n} \alpha_i \lambda_i^k x_i)}$$

$$= \frac{\sum_{i=1}^{n} \alpha_i^2 \lambda_i^{2k+1}}{\sum_{i=1}^{n} \alpha_i^2 \lambda_i^{2k}} = \lambda_1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)$$

4.3 反幂法

反幂法也称为逆迭代，它用来计算非奇异阵的按模最小的特征值及相应的特征向量．

设$|\boldsymbol{A}|\neq 0$，$\boldsymbol{A}$ 的特征值$|\lambda_1|\geqslant|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$，相应的特征向量为 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$ 则$\boldsymbol{A}^{-1}$的特征值即为：$\left|\frac{1}{\lambda_n}\right|\geqslant\left|\frac{1}{\lambda_{n-1}}\right|\geqslant\cdots\geqslant\left|\frac{1}{\lambda_1}\right|$，相应的特征向量为 $\boldsymbol{x}_n,\boldsymbol{x}_{n-1},\cdots\boldsymbol{x}_1$．因此，对 $\boldsymbol{A}^{-1}$使用乘幂法，得到按模最大的特征值，也就是得到了 $\boldsymbol{A}$ 的按模最小的特征值．

任取 $\boldsymbol{v}_0=\boldsymbol{u}_0\neq 0$，计算向量序列：

$$\begin{cases}\boldsymbol{v}_k=\boldsymbol{A}^{-1}\boldsymbol{u}_{k-1}\\ \boldsymbol{u}_k=\dfrac{\boldsymbol{v}_k}{\max(\boldsymbol{v}_k)}\end{cases}\quad(k=1,2,\cdots)$$

由乘幂法的介绍可知，$\max(\boldsymbol{v}_k)\approx\frac{1}{\lambda_n}$，$\boldsymbol{x}_n\approx\boldsymbol{u}_k=\frac{\boldsymbol{v}_k}{\max(\boldsymbol{v}_k)}$，但这里必须求得 $\boldsymbol{A}^{-1}$，当 n 较大时，求 A^{-1}往往比较困难，因此，通常是用到主元消元法解线性方程组的方法来求 $\boldsymbol{v}_k$，即通过

$$\boldsymbol{A}\boldsymbol{u}_k=\boldsymbol{u}_{k-1}\tag{4.4}$$

来求 $\boldsymbol{u}_k$．由于系数矩阵 $\boldsymbol{A}$ 不变，也常用三角分解：$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$（$\boldsymbol{L}$ 为下三角，$\boldsymbol{U}$ 为上三角）来求 $\boldsymbol{v}_k$．即式(4.4)等价于求解两个三角形方程组：

$$\begin{cases}\boldsymbol{L}\boldsymbol{y}_k=\boldsymbol{u}_{k-1}\\ \boldsymbol{U}\boldsymbol{v}_k=\boldsymbol{Y}_k\end{cases}$$

从而得到

$$\boldsymbol{u}_k=\frac{\boldsymbol{v}_k}{\max(\boldsymbol{v}_k)}\tag{4.5}$$

重复式(4.4)和式(4.5)迭代,直到满足精度要求为止,此时可确定:

$$\max(\boldsymbol{v}_k) = \frac{1}{\lambda_n}$$

$$\boldsymbol{u}_k = \frac{\boldsymbol{v}_k}{\max(\boldsymbol{v}_k)} \approx \boldsymbol{x}_n .$$

显然,这时的迭代精度取决于比值$\left|\frac{\lambda_n}{\lambda_{n-1}}\right|$,倘若$\left|\frac{\lambda_n}{\lambda_{n-1}}\right| \approx 1$,则采用原点位移法,即以 $\boldsymbol{A} - p\boldsymbol{I}$ 代替 $\boldsymbol{A}$,使得 $\boldsymbol{A} - p\boldsymbol{I}$ 的反幂法有较快的收敛速度.这里的位移值 p 通常取为 λ_n 的一个近似值$\bar{\lambda}_n$,这时,带原点位移的反幂法计算公式如下:

任取初始向量 $\boldsymbol{v}_0 = \boldsymbol{u}_0 \neq 0$,依次计算:

$$\begin{cases} (\boldsymbol{A} - \bar{\lambda}_n \boldsymbol{I})\boldsymbol{v}_k = \boldsymbol{u}_{k-1} \\ \boldsymbol{u}_k = \dfrac{\boldsymbol{v}_k}{\max(\boldsymbol{v}_k)} \qquad (k = 1, 2, \cdots) \end{cases} \tag{4.6}$$

此时,收敛速度由比值$\left|\frac{\lambda_n - \bar{\lambda}_n}{\lambda_j - \lambda_n}\right|$确定,$\bar{\lambda}_n$ 越精确,收敛越快.实际计算时,反幂法常用于已知矩阵的近似特征值而去求矩阵的特征向量.譬如,已用其他方法求得其特征值 λ_i 的近似值$\bar{\lambda}_i$,常用带原点位移法的反幂法式(4.6)求相应于 λ_i 的特征向量.当$|\lambda_i - \bar{\lambda}_i| < |\lambda_j - \bar{\lambda}_i|\ (j \neq i$ 时),$\lambda_i - \bar{\lambda}_i$ 为 $\boldsymbol{A} - \bar{\lambda}_i \boldsymbol{I}$ 的按模最小的特征值,从而在式(4.6)中,用 $\bar{\lambda}_i$ 代替$\bar{\lambda}_n$ 后,迭代格式(4.6)便给出了 $\boldsymbol{A}$ 的相应于λ_i 的特征向量 $\boldsymbol{x}_i$,只要近似值 $\bar{\lambda}_i$ 足够好,则收敛是相当快的.

例 4.3 利用反幂法计算矩阵 $\boldsymbol{A}$ 的对应于特征值$\bar{\lambda}_3 = 1.2679$(精确特征值比为 $\lambda_3 = 3 - \sqrt{3}$)的特征向量 $\boldsymbol{x}_3$.

$$\boldsymbol{A} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 4 \end{pmatrix}$$

解 根据原点位移反幂法，先分解矩阵

$$A = LU$$

$$= \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 1.3659 & 1.0 & 0.0 \\ 0.0 & 2.7310 & 1.0 \end{pmatrix} \begin{pmatrix} 0.7321 & 1.0 & 0.0 \\ 0.0 & 0.3662 & 1.0 \\ 0.0 & 0.0 & 0.0011 \end{pmatrix}$$

取初始向量 $$\boldsymbol{v}_0 = \boldsymbol{u}_0 = (1,1,1)^{\mathrm{T}}$$

解方程 $$\boldsymbol{LUv}_1 = \boldsymbol{u}_0$$

得 $$\boldsymbol{v}_1 = (3385.2,\ -2477.3,\ 908.20)^{\mathrm{T}}$$

且 $$\boldsymbol{u}_1 = \frac{\boldsymbol{v}_1}{\max(\boldsymbol{v}_1)} = (1.0000,\ -0.73180,\ 0.26828)^{\mathrm{T}}$$

再解方程组 $$\boldsymbol{LUv}_2 = \boldsymbol{u}_1$$

得 $$\boldsymbol{v}_2 = (20345,\ -14894,\ 5451.9)^{\mathrm{T}}$$

且 $$\boldsymbol{u}_2 = (1.0000,\ -0.73207,\ 0.26979)^{\mathrm{T}}$$

再解方程组 $$\boldsymbol{LUv}_3 = \boldsymbol{u}_2$$

得 $$\boldsymbol{v}_3 = \boldsymbol{v}_2$$

实际上的精确向量是

$$\boldsymbol{x}_3 = (1,\ 1-\sqrt{3},\ 2-\sqrt{3})^{\mathrm{T}}$$

$$= (1.0000,\ -0.73205,\ 0.26795)^{\mathrm{T}}$$

由此可见，$\boldsymbol{v}_2$ 是 $\boldsymbol{x}_3$ 的足够精确的近似向量，而改进的特征值为 $\lambda_3 \approx \bar{\lambda}_3 + \frac{1}{\max(\boldsymbol{v}_2)} = 1.2679492$，已准确到 $\lambda_3 = 3-\sqrt{3}$ 的小数点后7位．

4.4 雅可比(Jacobi)方法

雅可比方法用来求实对称阵 $\boldsymbol{A}$ 的全部特征值和特征向量．因为 $\boldsymbol{A}$ 实对称，故必存在正交阵 $\boldsymbol{P}$，使 $\boldsymbol{P}^{-1}\boldsymbol{AP} = \boldsymbol{D}$，这里 $\boldsymbol{D}$ 是对角阵，其对角线上元素即为 $\boldsymbol{A}$ 的特征值．雅可比方法就是用一系列正交相似变换，逐步构造出一个近似的正交矩阵 $\boldsymbol{P}$．

定义 4.2 称 n 阶矩阵

$$\boldsymbol{P}(i,j,\theta)=\begin{pmatrix}
1 & & & & & & & \\
& \ddots & & & & & & \\
& & 1 & & & & & \\
& & & \cos\theta & \cdots & \sin\theta & & \\
& & & \vdots & 1 & & & \\
& & & & & \ddots & & \\
& & & & & & 1 & \\
& & & -\sin\theta & \cdots & \cos\theta & & \\
& & & & & & 1 & \\
& & & & & & & \ddots \\
& & & & & & & & 1
\end{pmatrix}\begin{matrix}\\ \\ \\ \\ \\ \\ \\ (i\text{ 行})\\ \\ (j\text{ 行})\\ \\ \end{matrix}$$

为 R^n 中 x_i, x_j 平面内的一个平面旋转矩阵 .

平面旋转矩阵 $\boldsymbol{P}$ 的性质:

(1) $\boldsymbol{P}$ 是正交阵,即 $\boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}=\boldsymbol{I}$;

(2) $\boldsymbol{B}=\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$ 还是实对称,且 $\boldsymbol{B}$ 与 $\boldsymbol{A}$ 有相同的特征值 .

(3) $\|\boldsymbol{B}\|_F=\|\boldsymbol{A}\|_F$.

证明
$$\|\boldsymbol{A}\|_F^2=\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}^2=\mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$$
$$=\mathrm{tr}(\boldsymbol{A}^2)=\sum_{i=1}^{n}\lambda_i^2(\boldsymbol{A}),$$

又
$$\|\boldsymbol{B}\|_F^2=\mathrm{tr}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})=\mathrm{tr}\boldsymbol{B}^2=\sum_{i=1}^{n}\lambda_i^2(\boldsymbol{B}).$$

由性质(2),可知 $\|\boldsymbol{B}\|_F=\|\boldsymbol{A}\|_F$

对于 $\boldsymbol{B}=\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$,记 $\boldsymbol{B}=(b_{ij})_{n\times n}$, $\boldsymbol{A}=(a_{ij})_{n\times n}$,它们的元素之间有如下关系:

$$(1)\begin{cases}
b_{ii}=a_{ii}\cos^2\theta-2a_{ij}\sin\theta\cos\theta+a_{jj}\sin^2\theta\\
b_{jj}=a_{ii}\sin^2\theta+2a_{ij}\sin\theta\cos\theta+a_{jj}\cos^2\theta\\
b_{ij}=\dfrac{1}{2}(a_{ii}-a_{jj})\sin2\theta+a_{ij}\cos2\theta=b_{ji}
\end{cases}$$

$$(2)\ \begin{cases} b_{ik} = a_{ik}\cos\theta - a_{jk}\sin\theta \\ b_{jk} = a_{ik}\sin\theta + a_{jk}\cos\theta \end{cases} \qquad (k \neq i, j)$$

$$(3)\ \begin{cases} b_{ri} = a_{ri}\cos\theta - a_{rj}\sin\theta \\ b_{rj} = a_{ri}\sin\theta + a_{rj}\cos\theta \end{cases} \qquad (r \neq i, j)$$

$$(4)\quad b_{rk} = a_{rk} \qquad (r, k \neq i, j)$$

由上面这些关系式，可得如下结论：

(1) 要使 $b_{ij} = b_{ji} = 0$，应选择 θ，使

$$\tan 2\theta = \frac{-2a_{ij}}{a_{ii} - a_{jj}},$$

(这里为保证单值，限定 $|\theta| \leqslant \frac{\pi}{4}$).

$$(2)\ \begin{cases} b_{ii}^2 + b_{jj}^2 + 2b_{ij}^2 = a_{ii}^2 + a_{jj}^2 + 2a_{ij}^2 \\ b_{ri}^2 + b_{rj}^2 = a_{ri}^2 + a_{rj}^2 \qquad (r \neq i, j) \\ b_{ik}^2 + b_{jk}^2 = a_{ik}^2 + a_{jk}^2 \qquad (k \neq i, j) \end{cases}$$

进而可得：
$$\sum_{r,k} b_{rk}^2 = \sum_{r,k} a_{rk}^2$$

即
$$\| \boldsymbol{B} \|_F^2 = \| \boldsymbol{A} \|_F^2$$

记 $\sigma_A = \| \boldsymbol{A} \|_F^2$，$\sigma_B = \| \boldsymbol{B} \|_F^2$，上式即 $\sigma_B = \sigma_A$

$$(3)\ \sum_{r=1}^{n} b_{rr}^2 = \sum_{r \neq i,j} a_{rr}^2 + b_{ii}^2 + b_{jj}^2$$

$$= \sum_{r \neq i,j} a_{rr}^2 + (a_{ii}^2 + a_{jj}^2 + 2a_{ij}^2 - 2b_{ij}^2)$$

$$= \sum_{r=1}^{n} (a_{rr}^2 + 2a_{ij}^2 - 2b_{ij}^2)$$

$$(4)\ \tau_B = \sum_{\substack{r,k=1 \\ r \neq k}}^{n} b_{rk}^2 = \sigma_B - \sum_{r=1}^{n} b_{rr}^2$$

$$= \sigma_A - (\sum_{r=1}^{n} a_{rr}^2 + 2a_{ij}^2 - 2b_{ij}^2)$$

$$= \tau_A - 2a_{ij}^2 + 2b_{ij}^2$$

这里，τ_A，τ_B 分别表示非对角线上的元素的平方和.

由(3)与(4)可以看出，若选择 θ，使(1)成立，那么经过一次这

样的旋转变换后，$\boldsymbol{B}$ 的非对角线元素的平方和减少了 $2a_{ij}^2$，而对角线元素的平方和增加了 $2a_{ij}^2$.

(5) 若 a_{ij}是$\boldsymbol{A}$ 的按模最大的非对角线元素，则

$$a_{ij}^2 \geqslant \frac{1}{n(n-1)} \sum_{r \neq k} a_{rk}^2$$

从而，由(4)及(1)，得到

$$\sum_{r \neq k} b_{rk}^2 \leqslant \left(1 - \frac{2}{n(n-1)}\right) \sum_{r \neq k} a_{rk}^2$$

也即

$$\tau_B \leqslant \left(1 - \frac{2}{n(n-1)}\right) \tau_A \tag{4.7}$$

式(4.7)表明非对角线元素的平方和不超过原来的 $\left(1 - \frac{2}{n(n-1)}\right)$ 倍，这就是选择平面旋转变换 $\boldsymbol{P}(i,j,\theta)$的道理 .

一般，并不需要求出 θ，而只需求出 $\sin\theta$ 和 $\cos\theta$ 即可 .

(1) 若 $a_{ii} = a_{jj}$，则 $\tan 2\theta = \infty$，$\theta = \frac{\pi}{4}\operatorname{sgn}(a_{ij})$

(2) 若 $a_{ii} \neq a_{jj}$，则令

$$\frac{-2a_{ij}}{a_{ii} - a_{jj}} = \tan 2\theta = \frac{2\tan\theta}{1 - \tan^2\theta} = \frac{1}{c}$$

则有

$$\tan^2\theta + 2c\tan\theta - 1 = 0$$

$$\tan\theta = \frac{1}{c \pm \sqrt{c^2 + 1}} \tag{4.8}$$

由于限定 $|\theta| \leqslant \frac{\pi}{4}$，故 $|\tan\theta| \leqslant 1$，可选取

$$\tan\theta = \frac{\operatorname{sgn}(c)}{|c| + \sqrt{c^2 + 1}} \triangleq t \tag{4.9}$$

则

$$\begin{cases} \cos\theta = \dfrac{1}{\sqrt{1 + \tan^2\theta}} = \dfrac{1}{\sqrt{1 + t^2}} \\ \sin\theta = t \cdot \cos\theta = \dfrac{t}{\sqrt{1 + t^2}} \end{cases} \tag{4.10}$$

下面讨论雅可比方法的收敛性．由上面讨论可知，对 $\boldsymbol{A}_0=\boldsymbol{A}$ 施行一系列平面旋转变换：

$$\boldsymbol{A}_1=\boldsymbol{P}_1^{\mathrm{T}}\boldsymbol{A}_0\boldsymbol{P}_1$$

$$\boldsymbol{A}_2=\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{A}_1\boldsymbol{P}_2=\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{P}_1^{\mathrm{T}}\boldsymbol{A}_0\boldsymbol{P}_1\boldsymbol{P}_2$$

$$\cdots$$

$$\boldsymbol{A}_k=\boldsymbol{P}_k^{\mathrm{T}}\boldsymbol{A}_{k-1}\boldsymbol{P}_k$$

$$\cdots$$

每一步变换选择 $\boldsymbol{A}_{k-1}=(a_{ij}^{(k-1)})_{n\times n}$ 中的非对角元素按模最大者 $a_{ij}^{(k-1)}$（称为主元素）作为歼灭对象，同时，利用式(4.8)，式(4.9)和式(4.10)确定 $\cos\theta$ 和 $\sin\theta$，从而可得到旋转阵 $\boldsymbol{P}(i,j,\theta)$，使变换后，得到 $\boldsymbol{A}_k=(a_{ij}^{(k)})_{n\times n}$，而根据式(4.7)，有

$$\tau_{A_k}\leqslant\left(1-\frac{2}{n(n-1)}\right)\tau_{A_{k-1}},$$

递推可得

$$\tau_{A_k}\leqslant\left(1-\frac{2}{n(n-1)}\right)^k\tau_A$$

易见矩阵序列 $\{\boldsymbol{A}_k\}$ 的非对角元素的平方和趋于零，即

$$\lim_{k\to\infty}\boldsymbol{A}_k=\boldsymbol{D}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n),$$

这说明雅可比方法收敛．

另外，据 $\boldsymbol{A}_k=\boldsymbol{P}_k^{\mathrm{T}}\boldsymbol{P}_{k-1}^{\mathrm{T}}\cdots\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{P}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P}_1\boldsymbol{P}_2\cdots\boldsymbol{P}_{k-1}\boldsymbol{P}_k\xrightarrow{\Delta}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P}$，可知 $\boldsymbol{P}$ 的列向量 $\boldsymbol{x}_j(j=1,2,\cdots,n)$ 可作为相应的特征向量．

例 4.4 用雅可比方法计算矩阵 $\boldsymbol{A}$ 的特征值 λ_i 和相应的特征向量 $\boldsymbol{x}_i(i=1,2,3)$.

$$\boldsymbol{A}=\begin{pmatrix}3.5 & -6 & 5\\ -6 & 8.5 & -9\\ 5 & -9 & 8.5\end{pmatrix}$$

解 计算过程如下表所列，表中给出的特征值为

$$\lambda_1\approx 0.4851,\ \lambda_2\approx=-0.9520,\ \lambda_3=20.9669.$$

正交矩阵为

$$\boldsymbol{P} \approx \boldsymbol{P}_1\boldsymbol{P}_2\boldsymbol{P}_3\boldsymbol{P}_4\boldsymbol{P}_5 = \begin{pmatrix} 0.7998 & -0.3139 & 0.5117 \\ -0.2257 & 0.6325 & 0.7408 \\ -0.6159 & -0.7147 & 0.4351 \end{pmatrix}$$

故 $\lambda_1, \lambda_2, \lambda_3$ 对应的一组单位特征向量分别为

$$\boldsymbol{x}_1 = \begin{pmatrix} 0.7998 \\ -0.2257 \\ -0.6159 \end{pmatrix}, \ \boldsymbol{x}_2 = \begin{pmatrix} -0.3139 \\ 0.6325 \\ -0.7147 \end{pmatrix}, \ \boldsymbol{x}_3 = \begin{pmatrix} 0.5117 \\ 0.7408 \\ 0.4351 \end{pmatrix}$$

k	矩阵 A_k	$a_{ij}^{(k)}$	$\cos\theta_k$	$\sin\theta_k$
0	$A_0 = \begin{pmatrix} 3.5 & -6 & 5 \\ -6 & 8.5 & -9 \\ 5 & -9 & 8.5 \end{pmatrix}$	$a_{23}^{(0)} = -9$	0.7071	0.7071
1	$A_1 = \begin{pmatrix} 3.5 & -7.7782 & -0.7071 \\ -7.7782 & 17.5 & 0 \\ -0.7071 & - & -0.5 \end{pmatrix}$	$a_{12}^{(1)} = -7.7782$	0.9135	0.4069
2	$A_2 = \begin{pmatrix} 0.0358 & 0 & -0.6459 \\ 0 & 20.9653 & 0.2877 \\ -0.6459 & 0.2877 & -0.5 \end{pmatrix}$	$a_{13}^{(2)} = -0.6459$	0.8316	-0.5554
3	$A_3 = \begin{pmatrix} 0.4672 & -0.1598 & 0 \\ -0.1598 & 20.9653 & 0.2392 \\ 0 & 0.2392 & -0.9312 \end{pmatrix}$	$a_{23}^{(3)} = 0.2392$	0.9999	0.0107

续表

k	矩阵 A_k	$a_{ij}^{(k)}$	$\cos\theta_k$	$\sin\theta_k$
4	$A_4=\begin{pmatrix}0.4672 & -0.1598 & -0.0017\\ -0.1598 & -0.9340 & 0\\ -0.0017 & 0 & 20.9669\end{pmatrix}$	$a_{12}^{(4)}=-0.1598$	0.9937	-0.1119
5	$A_5=\begin{pmatrix}0.4851 & 0 & 0.0017\\ 0 & -0.9520 & -0.0002\\ 0.0017 & -0.0002 & 20.9669\end{pmatrix}$			

4.5 QR 方法

定义 4.3 设 W 是实的 n 维列向量,且 $W^{\mathrm{T}}W=1$,则 n 阶矩阵 $H=I-2WW^{\mathrm{T}}$ 称为 Housholder(豪斯豪尔特)矩阵或镜像反射矩阵,也称为初等反射阵.

若记 $W=(w_1,w_2,\cdots,w_n)^{\mathrm{T}}$ 则

$$H(W)=\begin{pmatrix}1-2w_1^2 & -2w_1w_2 & \cdots & -2w_1w_n\\ -2w_2w_1 & 1-2w_2^2 & \cdots & -2w_2w_n\\ \vdots & \vdots & \cdots & \vdots\\ -2w_nw_1 & -2w_nw_2 & \cdots & 1-2w_n^2\end{pmatrix}$$

初等反射阵 H 的性质:

(1) $H^{\mathrm{T}}=H$(是对称阵);

(2) $H^TH=I$(是正交阵);

(3) $H^2=I$(是对合阵).

任意给定一个非零向量 u,则

$$H(u)=I-2\frac{uu^{\mathrm{T}}}{\|u\|_2^2}$$是一个初等反射阵 .

任意给定非零向量 x,令 $y=Hx$,则 $\|x\|_2=\|y\|_2$,这是因为正交变换是保模变换:

$$\|y\|_2=\sqrt{(y,y)}=\sqrt{(Hx,Hx)}$$

$$=\sqrt{(H^{\mathrm{T}}Hx,x)}=\sqrt{x,x)}=\|x\|_2$$

定理 4.6 设 x,y 是给定的两个不相等的 n 维实向量,若 $\|x\|_2=\|y\|_2$,则一定存在初等反射阵 H,使 $Hx=Y$,即总可找到 H 阵,把 x 向量经过镜面反射变换到 y 方向,而保持长度不变 .

证明 当 $x\neq y$ 时,可取 $W=\dfrac{(x-y)}{\|x-y\|_2}$,显然 $W^{\mathrm{T}}W=1$ 构造

$$H=I-2\frac{(x-y)(x-y)^{\mathrm{T}}}{\|x-y\|_2^2},\text{则}$$

$$Hx=Ix-\frac{2(x-y)(x-y)^{\mathrm{T}}\cdot x}{\|x-y\|_2^2}$$

$$=x-2\frac{(x-y)(x^{\mathrm{T}}x-y^{\mathrm{T}}x)}{\|x-y\|_2^2}$$

而
$$\|x-y\|_2^2=(x-y)^{\mathrm{T}}(x-y)$$

$$=2(x^{\mathrm{T}}x-y^{\mathrm{T}}x),$$

故
$$Hx=x-(x-y)=y.$$

由定理6可知:对任何一个非零向量 x,可通过 Housholder 变换,变换到向量 σe_1,这里 $e_1=(1,0,\cdots,0)^T$,$|\sigma|=\|x\|_2$,即 $Hx=\sigma e_1$,且

$$W=\frac{x-\sigma e_1}{\|x-\sigma e_1\|_2}=\frac{1}{\sqrt{2\sigma(\sigma-x_1)}}(x_1-\sigma,\ x_2,\cdots,x_n)^T$$

$$H=I-\frac{2(x-\sigma e_1)(x-\sigma e_1)^T}{\|x-\sigma e_1\|_2^2}$$

$$= \boldsymbol{I} - \frac{(\boldsymbol{x} - \sigma \boldsymbol{e}_1)(\boldsymbol{x} - \sigma \boldsymbol{e}_1)^{\mathrm{T}}}{\sigma(\sigma - \boldsymbol{x}_1)}$$

$$\boldsymbol{HX} = \sigma \boldsymbol{e}_1 = (\sigma, 0, \cdots, 0)^{\mathrm{T}}$$

为了防止 σ 与 $\boldsymbol{x}_1$ 相互抵消，以利于算法稳定，可选取 σ 与 $\boldsymbol{x}_1$ 异号，即 $\sigma = -\operatorname{sgn}(\boldsymbol{x}_1) \| \boldsymbol{x} \|_2$.

若记 $\boldsymbol{u} = \boldsymbol{x} - \sigma \boldsymbol{e}_1$，$\alpha = \sigma(\sigma - \boldsymbol{x}_1)$，则上述 $\boldsymbol{H}$ 阵表示为 $\boldsymbol{H} = \boldsymbol{I} - \frac{1}{\alpha}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$.

推论 对任意非零向量 $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)^{\mathrm{T}} \in R^n$ 及单位长度向量 $\boldsymbol{e}_1 = (1, 0, \cdots, 0)^{\mathrm{T}}$，总存在一个初等反射阵 $\boldsymbol{H} = \boldsymbol{I} - \frac{1}{\alpha}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$，使 $\boldsymbol{H}\boldsymbol{x} = \sigma \boldsymbol{e}_1$. 其中 $\sigma = -\operatorname{sgn}(\boldsymbol{x}_1) \| \boldsymbol{x} \|_2$，$\alpha = \sigma(\sigma - \boldsymbol{x}_1)$，$\boldsymbol{u} = \boldsymbol{x} - \sigma \boldsymbol{e}_1 = (\boldsymbol{x}_1 - \alpha, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)^{\mathrm{T}}$.

例 4.5 给定向量 $\boldsymbol{x} = (2, 2, 1)^{\mathrm{T}}$，确定一个初等反射阵 $\boldsymbol{H}$，使 $\boldsymbol{HX}$ 的后两个分量为零.

解

$$\sigma = -\operatorname{sgn}(2) \| \boldsymbol{x} \|_2 = -3,$$

$$\alpha = \sigma(\sigma - \boldsymbol{x}_1) = 15,$$

$$\boldsymbol{u} = (5, 2, 1)^{\mathrm{T}}$$

则

$$\boldsymbol{H} = \boldsymbol{I} - \frac{1}{15}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$$

$$= \begin{pmatrix} -\frac{2}{3} & -\frac{2}{3} & -\frac{1}{3} \\ -\frac{2}{3} & \frac{11}{15} & -\frac{2}{15} \\ -\frac{1}{3} & -\frac{2}{15} & \frac{14}{15} \end{pmatrix}$$

且

$$\boldsymbol{H}\boldsymbol{x} = (-3, 0, 0)^{\mathrm{T}}$$

定义 4.4 若 n 阶矩阵 $\boldsymbol{A}$ 有分解式：$\boldsymbol{A} = \boldsymbol{QR}$，其中 $\boldsymbol{Q}$ 为 n 阶正交阵，$\boldsymbol{R}$ 为上三角阵，则称为 $\boldsymbol{A}$ 的 $\boldsymbol{QR}$ 分解或称为 $\boldsymbol{A}$ 的正交三角分解.

定理 4.7 设矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}$非奇异,则存在正交三角分解 $\boldsymbol{A}=\boldsymbol{QR}$,其中 $\boldsymbol{Q}$ 是 n 阶正交阵,$\boldsymbol{R}$ 是非奇异的上三角阵,且若限定 $\boldsymbol{R}$ 的对角元素均为正数,则此种分解唯一 .

证明 第一步:第 $\boldsymbol{A}^{(1)}=\boldsymbol{A}=[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n]$,$\boldsymbol{a}_j$ 表示$\boldsymbol{A}$ 的第j列向量 . 由 $\boldsymbol{A}$ 非奇异,可知 $\boldsymbol{a}_1\neq\boldsymbol{O}$,故由定理4.6之推论知:对 $\boldsymbol{a}_1$ 可确定 σ_1,α_1 和 $\boldsymbol{u}_1$,使 $\boldsymbol{H}_1\boldsymbol{a}_1=\left(\boldsymbol{I}-\frac{1}{\alpha_1}\boldsymbol{u}_1\boldsymbol{u}_1^{\mathrm{T}}\right)\boldsymbol{a}_1=\sigma_1\boldsymbol{e}_1$.

令 $\boldsymbol{A}^{(2)}=\boldsymbol{H}_1\boldsymbol{A}^{(1)}=\boldsymbol{H}[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n]$

$$=\begin{pmatrix}\sigma_1 & \boldsymbol{r}_2 & \boldsymbol{B}_2\\ & \boldsymbol{C}_2 & \boldsymbol{D}_2\end{pmatrix}$$

$$=\left(\begin{array}{c|ccc}\sigma_1 & a_{12}^{(2)} & \cdots & a_{1n}^{(2)}\\ \hline 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)}\\ \vdots & \vdots & \cdots & \vdots\\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)}\end{array}\right)$$

其中 $\boldsymbol{C}_2=\begin{pmatrix}a_{22}^{(2)}\\ \vdots\\ a_{n2}^{(2)}\end{pmatrix}$且 $\boldsymbol{A}^{(2)}$非奇异,可知 $\boldsymbol{C}_2\neq\boldsymbol{O}$.

第二步,对向量 $\boldsymbol{C}_2$,可确定 σ_2,α_2 和 $\boldsymbol{u}_2$,作 $\tilde{\boldsymbol{H}}_2=\left(\boldsymbol{I}-\frac{1}{\alpha_2}\boldsymbol{u}_2\boldsymbol{u}_2^{\mathrm{T}}\right)$,$\tilde{\boldsymbol{H}}_2\boldsymbol{C}_2=\sigma_2\boldsymbol{e}_1^{(n-1)}$,这里 $\boldsymbol{e}_1^{(n-1)}$ 表示 $n-1$ 维单位向量 . 再令

$$\boldsymbol{H}_2=\begin{pmatrix}1 & \boldsymbol{O}\\ \boldsymbol{O} & \tilde{\boldsymbol{H}}_2\end{pmatrix}$$

则
$$\boldsymbol{A}^{(3)}=\boldsymbol{H}_2\boldsymbol{A}^{(2)}=\begin{pmatrix}\sigma_1 & \boldsymbol{r}_2 & \boldsymbol{B}_2\\ & \sigma_2\boldsymbol{e}_1^{(n-1)} & \tilde{\boldsymbol{H}}_2\boldsymbol{D}\end{pmatrix}$$

$$=\begin{pmatrix}\boldsymbol{R}_3 & \boldsymbol{r}_3 & \boldsymbol{B}_3\\ & \boldsymbol{C}_3 & \boldsymbol{D}_3\end{pmatrix}$$

$$= \left(\begin{array}{cc:ccc} \sigma_1 & a_{12}^{(2)} & a_{13}^{(3)} & \cdots & a_{1n}^{(3)} \\ & \sigma_2 & a_{23}^{(3)} & \cdots & a_{2n}^{(3)} \\ \hdashline & & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ & & \vdots & \cdots & \vdots \\ & & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{array}\right)$$

这里，$\boldsymbol{R}_3$ 即左上角的二阶上三角阵，$\boldsymbol{C}_3$ 是 $n-2$ 维非零向量．

重复上述步骤，一般第 k 步为

$$\boldsymbol{A}^{(k)} = \begin{pmatrix} \boldsymbol{R}_k & \boldsymbol{r}_k & \boldsymbol{B}_k \\ & \boldsymbol{C}_k & \boldsymbol{D}_k \end{pmatrix}$$

其中，$\boldsymbol{R}_k$ 是一个 $k-1$ 阶上三角阵，$\boldsymbol{C}_k$ 是 $n-k+1$ 维非零向量，由定理 4.6 推论，可确定 σ_k, α_k 和 $\boldsymbol{u}_k$，使

$$\tilde{\boldsymbol{H}}_k \boldsymbol{C}_k = \left(\boldsymbol{I} - \frac{1}{\alpha_k}\boldsymbol{u}_k \boldsymbol{u}_k^{\mathrm{T}}\right)\boldsymbol{C}_k = \sigma_k \boldsymbol{e}_1^{(n-k+1)},$$

这里 $\boldsymbol{e}_1^{(n-k+1)}$ 表示 $n-k+1$ 维单位向量．再令

$$\boldsymbol{H}_k = \begin{pmatrix} \boldsymbol{I}_{k-1} & \boldsymbol{O}^{\mathrm{T}} \\ \boldsymbol{O} & \tilde{\boldsymbol{H}}_k \end{pmatrix}$$

则

$$\boldsymbol{A}^{(k+1)} = \boldsymbol{H}_k \boldsymbol{A}^{(k)} = \begin{pmatrix} \boldsymbol{R}_k & \boldsymbol{r}_k & \boldsymbol{B}_k \\ & \sigma_k \boldsymbol{e}_1^{(n-k+1)} & \boldsymbol{H}_k \boldsymbol{D}_k \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{R}_{k+1} & \boldsymbol{r}_{k+1} & \boldsymbol{B}_{k+1} \\ & \boldsymbol{C}_{k+1} & \boldsymbol{D}_{k+1} \end{pmatrix}$$

这样就完成了第 k 步的三角化过程，其中

$$\boldsymbol{R}_{k+1} = \begin{pmatrix} \boldsymbol{R}_k & \boldsymbol{r}_k \\ & \sigma_k \end{pmatrix}$$

是 k 阶上三角阵．经过 $n-1$ 步，逐次得到初等反射阵 $\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_{n-1}$，使得 $\boldsymbol{R} = \boldsymbol{A}^{(n)} = \boldsymbol{H}_{n-1}\boldsymbol{H}_{n-2}\cdots\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A}$ 为上三角阵．令 $\tilde{\boldsymbol{Q}} = \boldsymbol{H}_{n-1}\boldsymbol{H}_{n-2}\cdots\boldsymbol{H}_2\boldsymbol{H}_1$，则 $\tilde{\boldsymbol{Q}}$ 仍为正交阵，$\tilde{\boldsymbol{Q}}^{-1}$ 仍为正交阵．则有

$A=\tilde{Q}^{-1}R\xrightarrow{\Delta}QR$. 即利用 $n-1$ 个初等反射阵,实现了 A 的 QR 分解 .

下面给出 QR 方法计算过程:据定理 4.7,对于给定的可逆实矩阵 A,令 $A=A^{(1)}=Q_1R_1Q_1$ 为正交阵,R_1 为可逆上三角阵,交换乘法次序 . 得 $R_1Q_1=A^{(2)}$,对 $A^{(2)}$再分解:$A^{(2)}=Q_2R_2$,再交换乘法次序,$R_2Q_2=A^{(3)}$. 对 $A^{(3)}$再分解:$A^{(3)}=Q_3R_3,\cdots,$

一般地,令　　$A^{(k)}=Q_kR_k$

交换乘法次序:　$R_kQ_k=A^{(k+1)}$

再分解　　$A^{(k+1)}=Q_{k+1}R_{k+1},$

这样可得矩阵序列$\{A^{(k)}\}$. 注意到 $R_k=Q_k^{-1}A^{(k)}$. 则

$$A^{(k+1)}=R_kQ_k=Q_k^{-1}A^{(k)}Q_k\quad(k=1,2,\cdots)$$

表明 $A^{(k+1)}$是对 $A^{(k)}$做正交相似变换的结果 . 递推上式,便有

$$\begin{aligned}A^{(k+1)}&=Q_k^{-1}A^{(k)}Q_k\\&=Q_k^{-1}Q_{k-1}^{-1}A^{(k-1)}Q_{k-1}Q_k\\&=\cdots=Q_k^{-1}\cdots Q_1^{-1}AQ_1\cdots Q_k\\&\xrightarrow{\Delta}Q^{-1}AQ.\end{aligned}$$

表明 $A^{(k+1)}$与 A 有相同的特征值 . 可以证明,这样得到的矩阵序列$\{A^{(k)}\}$,基本收敛于某上三角阵 R_A,而 R_A 的对角线元素就是 A 的诸特征值 .

习　题　4

1. 用幂法求下列矩阵的按模最大特征值及相应的特征向量 .

(1) $\begin{pmatrix}2&3&2\\10&3&4\\3&6&1\end{pmatrix}$;　(2) $\begin{pmatrix}3&-4&3\\-4&6&3\\3&3&1\end{pmatrix}$

当特征值有三位小数稳定时迭代终止 .

2. 设 A 是非奇异对称矩阵，且有相异特征值，矩阵 M 由 A 的特征向量组成．证明 $M^{\mathrm{T}}M$ 是对角矩阵．

3. 直接应用乘幂法，且选择适当的原点位移计算矩阵

$$A=\begin{pmatrix}-1 & 2 & 1\\ 2 & -4 & 1\\ 1 & 1 & 6\end{pmatrix}$$

的按模最大特征值，精确到四位小数．

4. 用幂法求矩阵 A 的按模最大特征值，取初值 $x^{(0)}=(1,1,1,1)^{\mathrm{T}}$ 其中

$$A=\begin{pmatrix}2 & 1 & 3 & 4\\ 1 & -3 & 1 & 5\\ 3 & 1 & 6 & -2\\ 4 & 5 & -2 & -1\end{pmatrix}$$

这是一个收敛很慢的例子．(迭代 1200 次后，λ_1 仅取到精确四位小数)

5. 用反幂法求矩阵 A 的按模最小的特征值．其中

$$A=\begin{pmatrix}2 & 0 & 0\\ 2 & 2 & 1\\ 1 & 1 & 2\end{pmatrix}$$

6. 用雅可比方法求下列矩阵的所有特征值：

(1) $A=\begin{pmatrix}7 & 3 & -2\\ 3 & 4 & -1\\ -2 & -1 & 3\end{pmatrix}$；　(2) $A=\begin{pmatrix}3 & -4 & 3\\ -4 & 6 & 3\\ 3 & 3 & 1\end{pmatrix}$.

7. 求 Householder 矩阵 $I-2WW^{\mathrm{T}}(W^{\mathrm{T}}W=1)$ 的特征值和特征向量．

8. 求一平面反射变换 H，使 $HX=Y=\pm\|x\|_2\cdot e_1$，其中

(1) $x=(1,0,1)^{\mathrm{T}}$；

(2) $x=(5,1,0,0)^{\mathrm{T}}$.

9. (1) 设 A 是对称矩阵，λ 和 $x(\|x\|_2=1)$ 是 A 的一个特征

值及相应的特征向量，又设 $\boldsymbol{P}$ 为一个正交阵，使

$$\boldsymbol{PX} = \boldsymbol{e}_1 = (1,0,\cdots,0)^{\mathrm{T}}$$

证明：$\boldsymbol{B} = \boldsymbol{PAP}^{\mathrm{T}}$ 的第一行和第一列除了 λ 外其余元素均为零．

(2) 对于矩阵

$$\boldsymbol{A} = \begin{pmatrix} 2 & 10 & 2 \\ 10 & 5 & -8 \\ 2 & -8 & 11 \end{pmatrix},$$

$\lambda = 9$ 是其特征值，$\boldsymbol{x} = \left(\frac{2}{3}, \frac{1}{3}, \frac{2}{3}\right)^{\mathrm{T}}$ 是相应于 9 的特征向量，试求一初等反射阵 $\boldsymbol{P}$，使 $\boldsymbol{PX} = \boldsymbol{e}_1$，并计算 $\boldsymbol{B} = \boldsymbol{PAP}^{\mathrm{T}}$．

10. 对矩阵

$$\boldsymbol{A} = \begin{pmatrix} 3 & 3 & 4 \\ 1 & 1 & 2 \\ 4 & 2 & 1 \end{pmatrix}$$

进行 QR 分解．

第五章　代数插值

5.1　引　言

我们知道，许多实际问题都可用函数 $y=f(x)$来表示某种内在规律的数量关系，但在很多应用领域，往往只能通过实验或观测等手段得到 $y=f(x)$在某$[a,b]$区间上的有限个互异点 x_i 处对应的函数值 $y_i=f(x_i)$，$(i=0,1,2,\cdots,n)$，也即已知一个数据表．为了研究函数的变化规律，必须将其公式化，因此，我们希望根据给定的数据表作一个既能反映函数 $f(x)$的特性、又便于计算的简单函数 $p(x)$来近似 $f(x)$，如果要求 $p(x_i)=f(x_i)(i=0,1,2,\cdots,n)$，这就是最基本的插值问题．$p(x)$称为插值函数，插值函数的选择，取决于使用上的需要，可以是代数多项式，也可以是三角多项式或有理函数；可以是区间上任意光滑函数，也可以是分段光滑函数．当然，对于同一个问题来讲，在不同的插值函数类中求得的插值函数 $p(x)$逼近 $f(x)$的效果是不同的，当选择 $p(x)$为代数多项式时就称为代数插值．本章重点讨论代数插值．

定义 5.1　设函数 $y=f(x)$在区间$[a,b]$上有定义，且已知它在点 $a\leqslant x_0<x_1<\cdots<x_n\leqslant b$ 上的函数值 $y_0,y_1,\cdots,y_n$，若存在一个次数不超过 n 次的多项式

$$p(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n \quad (\text{其中 } a_i \text{ 为实数})$$

满足条件

$$p(x_i)=f(x_i)=y_i \quad (i=0,1,2,\cdots,n) \tag{5.1}$$

则称 $p(x)$为函数 $f(x)$的 n 次代数插值多项式．

相应的插值问题称为 n 次代数插值问题，求插值函数 $p(x)$

的方法称为**插值法**．点 $x_0, x_1, \cdots, x_n$ 称为插值节点，函数 $f(x)$ 叫做**被插值函数**，式(5.1)叫做**插值条件**，区间 $[a, b]$ 叫做**插值区间**，点 x 叫做**插值点**．插值点在插值区间内的叫做**内插**，否则叫做**外插**．

代数插值有明确的几何意义，即就是通过平面上给定的 $n+1$ 个互异点 $(x_i, y_i)(i=0,1,2,\cdots,n)$，作一条 n 次代数曲线 $y=p(x)$ 近似地表示曲线 $y=f(x)$．见图 5-1．

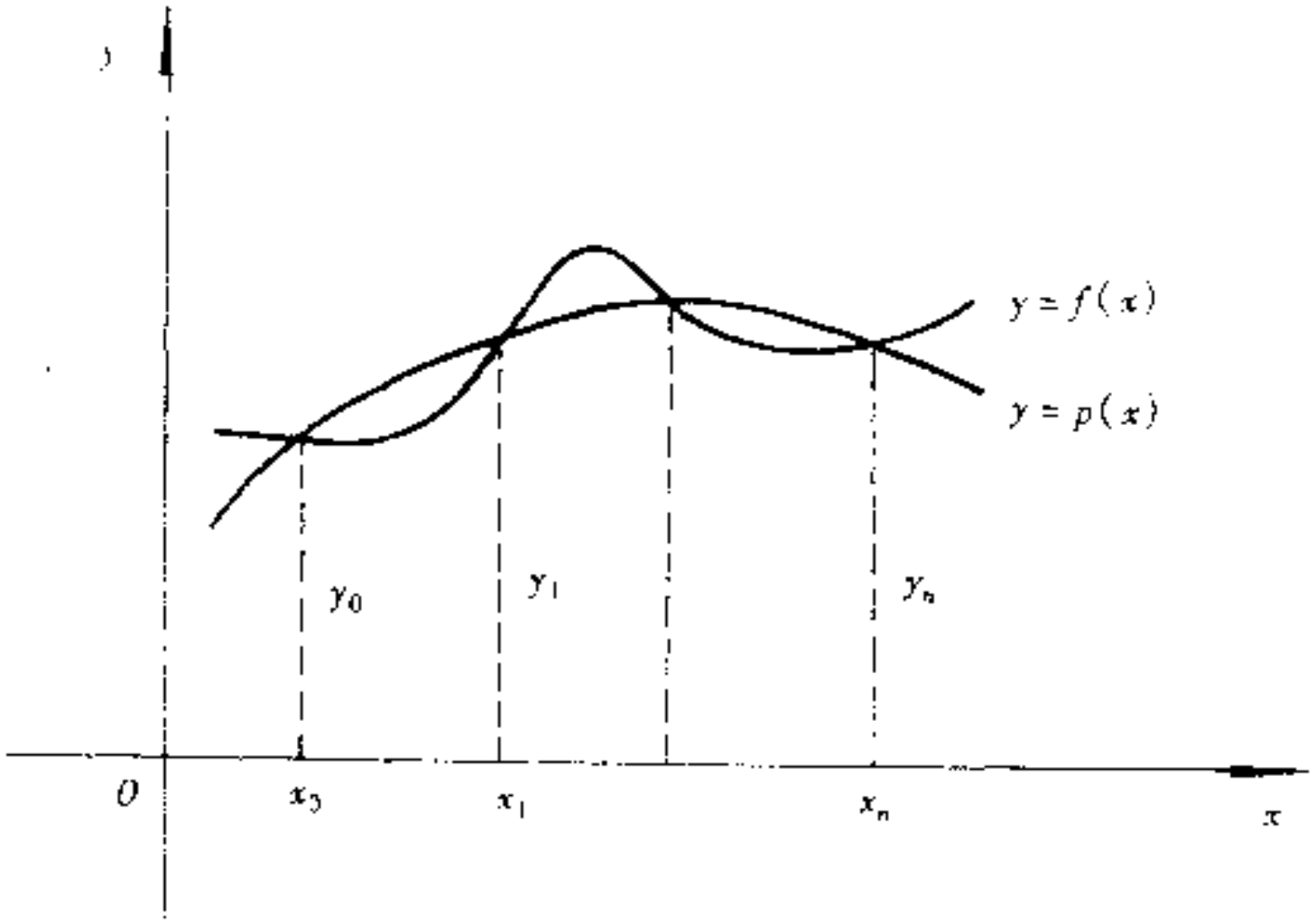


图 5-1

定理 5.1　n 次代数插值问题的解是存在且唯一的．

证明　设 n 次多项式

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

是函数 $y=f(x)$ 在 $[a, b]$ 上的 $n+1$ 个互异的节点 $x_i(i=0,1,2,\cdots,n)$ 上的插值多项式．则求插值多项式 $p(x)$ 的问题就可归结为求它的系数 $a_i(i=0,1,2,\cdots,n)$．

由插值条件：$p(x_i)=y_i(i=0,1,2,\cdots,n)$，可得关于系数 $a_0, a_1, \cdots, a_n$ 为未知元的 $n+1$ 阶线性代数方程组

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n = y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n = y_1 \\ \cdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n = y_n \end{cases} \tag{5.2}$$

由于方程组(5.2)的系数行列式是范德蒙德(Vandermonde)行列式

$$V(x_0, x_1, \cdots, x_n) = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix} = \prod_{i=1}^{n} \prod_{j=0}^{i-1} (x_i - x_j).$$

而 x_i 又是互不相同的，所以上式不为零，根据解线性方程组的克莱姆(Cramer)法则，方程组的解 a_i 存在唯一，从而 $p(x)$ 被唯一确定，这就证得了 n 次代数插值问题的解是存在且唯一的 .

唯一性说明，不论用什么方法来构造，也不论用什么形式来表示插值多项式 $p(x)$，只要满足同样的插值条件(5.1)，其结果都是互相恒等的 . 下面介绍几种常用的方法 .

5.2 拉格朗日(Lagrange)插值

一、基本插值多项式

为了构造满足插值条件 $p(x_i) = y_i (i = 0, 1, 2, \cdots, n)$ 的便于使用的插值多项式 $p(x)$，我们首先考虑一个简单而特殊的插值问题 .

求一个 n 次基本插值多项式 $l_k(x)$，使它在各节点 $x_i (i = 0, 1, 2, \cdots n)$ 上的值为

$$l_k(x_i) = \begin{cases} 0, i \neq k \\ 1, i = k \end{cases} \quad (i, k = 0, 1, 2, \cdots, n) \tag{5.3}$$

显然,这个问题容易解决.事实上,由条件 $l_k(x_i)=0(i\neq k)$知,$x_0,x_1,\cdots,x_{k-1},x_{k+1},\cdots,x_n$ 都是 n 次多项式 $l_k(x)$的零点,故可设

$$l_k(x)=A_k(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n),$$

其中,A_k 为待定系数.由条件 $l_k(x_k)=1$ 可求得 A_k,从而得

$$l_k(x)=\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{x-x_j}{x_k-x_j} \tag{5.4}$$

取 $k=0,1,2,\cdots,n$,就得到满足条件(5.3)的 $n+1$ 个 n 次基本插值多项式 $l_0(x),l_1(x),\cdots,l_n(x)$. 这些多项式称为在 $n+1$ 个节点 $x_0,x_1,\cdots,x_n$ 上的 n 次基本插值多项式或 n 次拉格朗日插值基函数.

二、拉格朗日插值多项式

以 $n+1$ 个 n 次基本插值多项式 $l_k(x)(k=0,1,2,\cdots,n)$为基础,就能直接写出满足插值条件(5.1)的 n 次代数插值多项式.

定理 5.2 n 次代数插值问题的解可表为

$$p(x)=\sum_{k=0}^{n}l_k(x)y_k, \tag{5.5}$$

证明 由于 $l_k(x)(k=0,1,2,\cdots,n)$都是 n 次多项式,所以它们的线性组合 $p(x)=\sum_{k=0}^{n}l_k(x)y_k$ 是次数不超过 n 次的多项式,且由

$$l_k(x_i)=\delta_{ki}=\begin{cases}1, & i=k\\ 0, & i\neq k\end{cases}$$

可得

$$p(x_i)=\sum_{k=0}^{n}l_k(x_i)y_k=\sum_{k=0}^{n}\delta_{ki}y_k=y_i,(i=0,1,\cdots,n)$$

因而 $p(x)$满足插值条件.所以公式(5.5)就是要求的 n 次代数插值多项式.

我们称形如式(5.5)的插值多项式为 n 次拉格朗日插值多项式,并记为 $L_n(x)$,即

$$L_n(x)=\sum_{k=0}^{n} l_k(x)y_k \tag{5.6}$$

由定理 5.1 知,式(5.6)必定与解方程组(5.2)所确定的插值多项式相同 . 因而,它只是插值多项式的另一种表示形式 .

例 5.1 给定数据表:

x_i	0	1	2	3
y_i	0	1	5	14

求三次拉格朗日插值多项式 $L_3(x)$.

解 在公式(5.6)中令 $n=3$ 展开,并代入数据表中的值得

$$L_3(x)=0\cdot l_0(x)+1\cdot l_1(x)+5\cdot l_2(x)+14\cdot l_3(x)$$

$$=0+1\cdot\frac{x(x-2)(x-3)}{1\cdot(-1)\cdot(-2)}+5\cdot\frac{x(x-1)(x-3)}{2\times1\times(-1)}$$

$$+14\cdot\frac{x(x-1)(x-2)}{3\times2\times1}$$

$$=\frac{1}{6}(2x^3+3x^2+x)$$

$$=\frac{1}{6}x(x+1)(2x+1)$$

三、线性插值与抛物线插值

当 $n=1$ 时,拉格朗日插值多项式(5.6)为

$$L_1(x)=l_0(x)y_0+l_1(x)y_1$$

即

$$L_1(x)=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1 \tag{5.7}$$

用 $L_1(x)$近似代替 $f(x)$称为线性插值 . 公式(5.7)称为线性

插值公式．在利用各种函数表计算函数值时，常常用到线性插值．

在几何上，用 $L_1(x)$近似代替 $f(x)$就是用通过两点(x_0, y_0)与(x_1, y_1)的直线 $y = L_1(x)$近似地代替曲线 $y = f(x)$，如图 5-2 所示．当 $n = 2$ 时，拉格朗日插值多项式(5.6)为

$$L_2(x) = l_0(x)y_0 + l_1(x)y_1 + l_2(x)y_2$$

即
$$L_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}y_2 \tag{5.8}$$

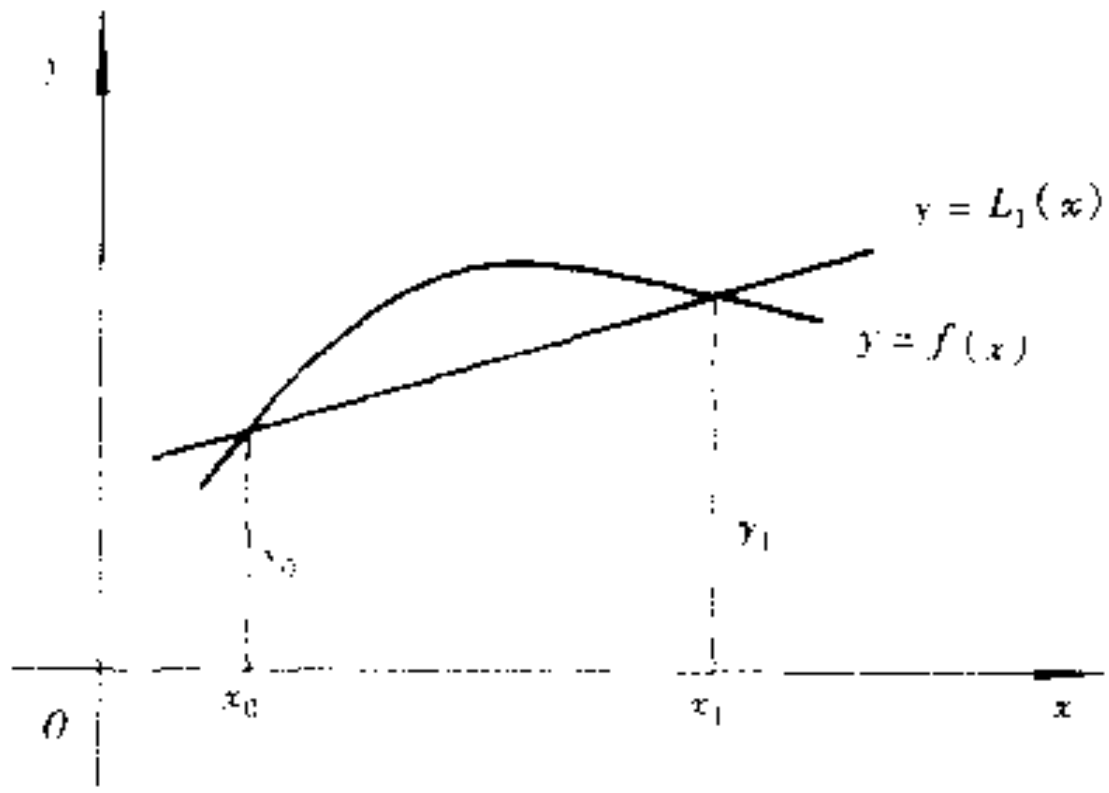


图 5-2

用 $L_2(x)$近似代替 $f(x)$称为二次插值．在几何上就是通过三点(x_0, y_0)，(x_1, y_1)与(x_2, y_2)的抛物线 $y = L_2(x)$近似地代替曲线 $y = f(x)$，如图 5-3 所示．因此二次插值又称抛物线插值．称式(5.8)为抛物线插值公式，抛物线插值是工程技术中广泛应用的一种插值方法．

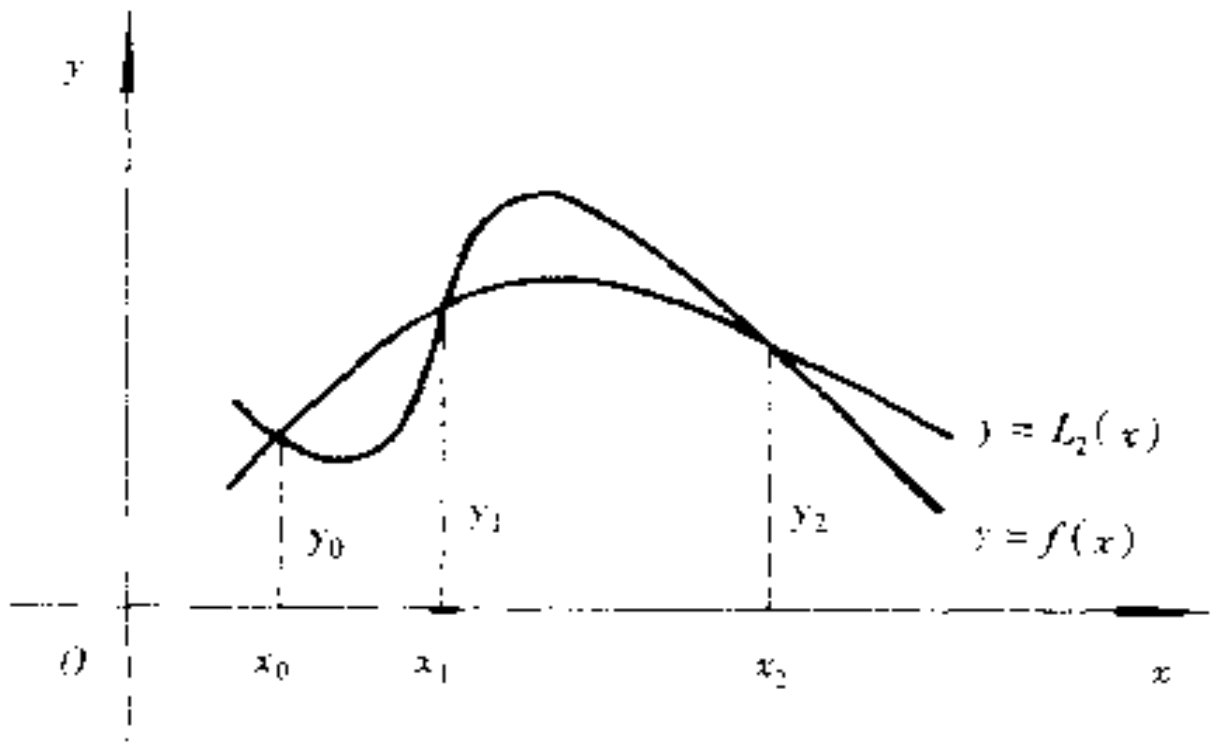


图 5-3

四、插值余项

我们称 $R_n(x) = f(x) - L_n(x)$为拉格朗日插值多项式的插值余项．关于插值余项估计有如下的拉格朗日余项定理．

定理 5.3 设 $f^{(n)}(x)$在$[a,b]$上连续，$f^{(n+1)}(x)$在(a,b)内存在，$L_n(x)$是函数 $f(x)$在$[a,b]$上的 $n+1$ 个互异节点 $x_0, x_1, \cdots, x_n$ 上的插值多项式，则对任意给定的 $x \in [a,b]$，插值余项为

$$R_n(x) = f(x) - L_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x - x_j) \qquad (5.9)$$

其中 $\xi \in (a,b)$，且依赖于 x．

证明 当 $x = x_i$ 时，式(5.9)的两边均为零，结论显然成立．现设 $x \neq x_i (i = 0, 1, \cdots, n)$，且令 x 是在$[a,b]$中任一固定的数．作辅助函数

$$g(t) = f(t) - L_n(t) - K(x)\pi(t), t \in [a,b],$$

其中 $\pi(t) = (t - x_0)(t - x_1)\cdots(t - x_n)$，$K(x)$满足

$$f(x) - L_n(x) - K(x)\pi(x) = 0, \qquad (5.10)$$

这里，由于 $x \neq x_i$，故 $\pi(x) \neq 0$，所以，这样的 $K(x)$是存在的．我们可以看出 $g(t)$具有下列性质：

(1) $g^{(n)}(t)$在$[a,b]$上连续，$g^{(n+1)}(t)$在(a,b)中存在，且

$$g^{(n+1)}(t)=f^{(n+1)}(t)-K(x)\cdot(n+1)!, \qquad (5.11)$$

(2) 由插值条件$f(x_i)=L_n(x_i)$知，当$t=x_i$时，有$g(x_i)=0$$(i=0,1,\cdots,n)$，且当$t=x$时，由(5.10)式可得

$$g(x)=f(x)-L_n(x)-K(x)\pi(x)=0.$$

于是，$g(t)$在$[a,b]$上至少有$n+2$个互异的零点：

$$t=x,x_0,\ x_1,\cdots,x_n.$$

根据罗尔(Rolle)定理，$g'(t)$在$g(t)$的每两个零点之间至少有一个零点，故$g'(t)$在(a,b)中至少有$n+1$个互异的零点$\zeta_i^{(1)}$，使$g'(\xi_i^{(1)})=0(i=0,1,\cdots,n)$，对$g'(t)$再应用罗尔定理，知$g''(t)$在$(a,b)$内至少有$n$个互异的零点$\xi_i^{(2)}$，使$g''(\xi_i^{(2)})=0(i=0,1,2,\cdots,n-1)$，依次类推，反复应用罗尔定理$n+1$次，可知$g^{(n+1)}(t)$在$(a,b)$由至少有一个零点，不妨记为$\xi$，使$g^{(n+1)}(\xi)=0$，于是由式(5.11)得

$$f^{(n+1)}(\xi)-K(x)\cdot(n+1)!\ =0$$

从而得

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

将它代入式(5.10)，即得余项的表达式

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j),\ \xi(x)\in(a,b).$$

在具体应用时，由于ξ在(a,b)内不能具体给出，计算误差$R_n(x)$仍然困难，对于某些函数$f(x)$．若在区间$[a,b]$上有$|f^{(n+1)}_{(x)}|\leqslant M_{n+1}$，则可用不等式$|R_n(x)|\leqslant\frac{M_{n+1}}{(n+1)!}|\prod_{j=0}^{n}(x-x_j)|$来估计插值的误差上界．

由插值余项公式(5.9)知，当函数$f(x)\in p_n$时，有$L_n(x)\equiv f(x)$，特别地，当令$f(x)\equiv 1$时，可得$\sum_{i=0}^{n}l_i(x)\equiv 1$.

但应当注意:插值多项式可以有不同的表达形式,如果不限次数,插值多项式并不唯一,事实上,若设 $\alpha(x)$是任意多项式,则多项式 $p(x)=L_n(x)+\alpha(x)(x-x_0)(x-x_1)\cdots(x-x_n)$都是满足插值条件的插值多项式.

由于抛物插值的精度比线性插值好,人们容易产生一种错觉,认为插值多项式的次数 n 越高精度越好,但实际并非如此.事实上,当代数插值的次数较高时,常有数值不稳定的现象,随着节点数的增多,插值多项式不一定收敛到被插值函数.本世纪初(1901年)德国数学家龙格(Runge)就给出了一个等距节点插值多项式 $L_n(x)$不收敛于 $f(x)$的例子:若取 $f(x)=\dfrac{1}{1+x^2}$在$[-5,5]$上的 n 次等距节点插值多项式 $L_n(x)$,则当 $n\to\infty$时,这一插值多项式是发散的.人们常把这种现象称为龙格现象.著名的俄国数学家伯恩斯坦(Бернштейн),也给出了一个十分重要的例子,对于定义在区间$[-1,1]$上的函数 $f(x)=|x|$,按等距节点:$-1=x_0<x_1<\cdots<x_n=1$ 构造 $f(x)$的插值多项式序列$\{L_n\}$,当 $n\to\infty$时,只有在三个点 $x=-1,0,1$ 上收敛于 $f(x)$,其余都不收敛.所以,这些事实都告诉我们,对于一般的连续函数,作插值多项式时,是需要小心的,不能认为插值节点取得越多总能保证逼近得越好.同时,在节点越多的情况下,相应的高次插值多项式会发生激烈振荡,这亦是很不理想的,特别在构造插值多项式时,所用到的节点上的函数值,往往都带有误差,因此,随着高次插值多项式的不良性态,这种误差可能会越来越扩大.也即高次插值多项式是不稳定的,因此从收敛性与稳定性考虑,使用高次插值多项式是不可取的.所以应用中,高次($n>7$)插值一般不用,但在解决实际问题时,我们总希望将所得到的数据点都用上,且所用数据点越多越好,解决这一矛盾的办法是改用分段低次插值或样条插值.

所谓**分段低次插值**,即用分段多项式来代替单个多项式作插值.具体说,就是在实际进行插值时,先把整个插值区间分成若干

个小区间，然后在每个小区间上分别用低次插值多项式进行插值．例如可用线性插值或抛物线插值等，然后再将每个小区间上的插值函数拼接在一起，作为整个插值区间上的插值函数，这样一种插值方法就叫做分段低次插值．显然，分段插值具有公式简单，节省运算量，稳定性好，收敛性有保证等优点，只要每一个小区间的长度取得足够小，分段低次插值总能满足所要求的精度．由于分段低次插值多项式对函数通常有较好的逼近性态，因而近年来有着十分广泛的应用．

例 5.2 设函数 $f(x)=e^x, x\in[0,1]$．试给出建立在对区间 $[0,1]$ 作 n 等分基础上的分段线性插值多项式，并在 $[0,1]$ 区间上估计误差：

解 设 $x_j=jh$，这里 $h=1/n, j=0,1,\cdots,n$，在任一子区间 $[x_j,x_{j+1}]$ 有

$$p(x)=e^{x_j}\cdot\frac{x_{j+1}-x}{h}+e^{x_{j+1}}\frac{x-x_j}{h}, x\in[x_j,x_{j+1}],\ (j=0,1,\cdots,n)$$

在任一子区间 $[x_j,x_{j+1}]$ 上进行误差估计：

$$f(x)-p(x)=\frac{f''(\xi)}{2!}(x-x_j)(x-x_{j+1})$$

$$|f(x)-p(x)|\leqslant\frac{|f''(\xi)|}{2!}|(x-jh)(x-(j+1)h)|$$

$$|f(x)-p(x)|\leqslant\frac{1}{2}\max_{\xi\in[0,1]}e^{\xi}\max_{x\in[x_j,x_{j+1}]}|(x-jh)(x-(j+1)h)|$$

$$\leqslant\frac{1}{2}e\max_{x\in[x_j,x_{j+1}]}\left[\frac{((j+1)h-x)+(x-jh)}{2}\right]^2$$

所以

$$|f(x)-p(x)|\leqslant\frac{1}{8}eh^2.$$

5.3 差商与牛顿(Newton)插值

拉格朗日插值公式含义直观，构造容易，很适合在电子计算机

上使用．但是它的一个主要缺点是，当我们用已知的 $n+1$ 个数据点求出拉格朗日插值多项式后，若又获得了新的数据点，要用它连同原有的 $n+1$ 个数据点一起求出高一次的插值多项式时，整个计算工作必须从头开始，不仅原来的每一项都要改变，还要增加一项计算．为克服这一缺点，本节将介绍一种能灵活增加节点的牛顿插值公式．由于在构造牛顿值公式时，要用到差商，下面我们先介绍差商的概念．

一、差商的概念

我们已知，对于连续变量的函数，在高等数学中常运用微商来研究函数的性质，对于由表格形式给出的离散变量的函数，则需要引进新的数学工具．数值分析的重要任务之一是将连续形式的函数离散化，使之便于上机计算．而差商正是函数微商形式离散化的重要工具．

设给定一个函数表：

x_i	x_0	x_1	$\cdots$	x_i	$\cdots$	x_j	$\cdots$	x_n
$f(x_i)$	$f(x_0)$	$f(x_1)$	$\cdots$	$f(x_i)$	$\cdots$	$f(x_j)$	$\cdots$	$f(x_n)$

$x_0, x_1, \cdots, x_n$ 是 $[a,b]$ 上互不相同的节点．

各阶差商定义如下：

$f[x_i]=f(x_i)$，称为函数 $f(x)$ 在 x_i 点的零阶差商．

$$f[x_i,x_j]=\frac{f[x_i]-f[x_j]}{x_i-x_j},\ (i\neq j) \tag{5.12}$$

称为函数 $f(x)$ 在 x_i 及 x_j 两点的一阶差商．显然式(5.12)表示函数 $f(x)$ 在 $[x_i,x_j]$ 上的平均变化率．

一阶差商的差商称为二阶差商．$f(x)$ 在 x_i, x_j, x_k 点处的二阶差商定义为

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k} \tag{5.13}$$

依次类推，一般地，有了 $n-1$ 阶差商之后，可以递推地来定义 $n-1$ 阶差商的差商为 n 阶差商 .

例如，函数 $f(x)$在 $x_0, x_1, \cdots, x_n$ 点处的 n 阶差商定义为

$$f[x_0, x_1, \cdots, x_n] = \frac{f[x_0, x_1, \cdots, x_{n-1}] - f[x_1, x_2, \cdots, x_n]}{x_0 - x_n} \tag{5.14}$$

也就是说，高阶差商是由比它低一阶的两个差商组合而得到 .

二、差商的性质

差商具有如下一些重要性质：

(1) 差商关于所含的节点是对称的。

例如，在 k 阶差商 $f[x_0, x_1, \cdots, x_k]$ 中，任意调换节点 x_i 与 x_j 的次序，其值不变 . 即

$$f[x_0, x_1, \cdots, x_i, \cdots, x_j, \cdots, x_k] = f[x_0, x_1, \cdots, x_j, \cdots, x_i, \cdots, x_k] \tag{5.15}$$

(2) 差商可以表示为函数值的线性组合。

例如，函数 $f(x)$的 k 阶差商 $f[x_0, x_1, \cdots, x_k]$ 可由函数值 $f(x_0), f(x_1), f(x_2), \cdots, f(x_k)$的线性组合表示，即

$$f[x_0, x_1, \cdots, x_k] = \sum_{j=0}^{k} \frac{f(x_j)}{(x_j - x_0)(x_j - x_1)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_k)}, \tag{5.16}$$

(3) 如果 $f(x)$的 k 阶差商 $f[x, x_0, \cdots, x_{k-1}]$是 x 的 m 次多项式 . 则 $f(x)$的 $k+1$ 阶差商 $f[x, x_0, \cdots, x_{k-1}, x_k]$是 x 的 $m-1$ 次多项式 .

事实上，由差商定义

$$f[x,x_0,\cdots,x_{k-1},x_k]=\frac{f[x,x_0,\cdots,x_{k-1}]-f[x_0,\cdots,x_{k-1},x_k]}{x-x_k}$$

由已知,上式右端分子是 x 的 m 次多项式,在 $x=x_k$ 时,

$$\begin{aligned}&f[x,x_0,\cdots,x_{k-1}]-f[x_0,\cdots,x_{k-1},x_k]\\&=f[x_k,x_0,\cdots,x_{k-1}]-f[x_0,\cdots,x_{k-1},x_k]=0,\end{aligned}$$

故分子中必含有因子$(x-x_k)$,所以分子与分母可约去公因子$(x-x_k)$,故上式右端应是 $m-1$ 次多项式 .

推论 n 次多项式$f(x)$的 k 阶差商$f[x,x_0,x_1,\cdots,x_{k-1}]$,当 $k\leqslant n$ 时是一个 $n-k$ 次多项式;而当 $k>n$ 时等于零 .

(4) 设 $f(x)$在含有 $x_0,x_1,\cdots,x_n$ 的区间$[a,b]$上具有 n 阶导数,则在这一区间内至少有一点 ζ,使

$$f[x_0,x_1,\cdots,x_n]=\frac{f^{(n)}(\zeta)}{n!},\ \zeta\in(a,b)\qquad(5.17)$$

式(5.17)揭示了差商与导数之间的关系 .

三、牛顿插值公式

下面我们从差商的定义来构造 n 次代数插值多项式的另一表达形式——牛顿插值多项式 .

由 $f(x)$在 x,x_0 上的一阶差商的定义

$$f[x,x_0]=\frac{f[x]-f[x_0]}{x-x_0},$$

可得 $f[x]=f[x_0]+(x-x_0)f[x,x_0]$,

由 $f(x)$在 x,x_0,x_1 上的二阶差商的定义

$$f[x,x_0,x_1]=\frac{f[x,x_0]-f[x_0,x_1]}{x-x_1},$$

可得 $f[x,x_0]=f[x_0,x_1]+(x-x_1)f[x,x_0,x_1]$,依次类推,继续由各阶差商的定义可得一组式子:

$$\begin{cases} f[x] = f[x_0] + (x - x_0) f[x, x_0] \\ f[x, x_0] = f[x_0, x_1] \\ \qquad\qquad + (x - x_1) f[x, x_0, x_1] \\ f[x, x_0, x_1] = f[x_0, x_1, x_2] \\ \qquad\qquad + (x - x_2) f[x, x_0, x_1, x_2] \\ \cdots\cdots \\ f[x, x_0, x_1, \cdots, x_{n-1}] = f[x_0, x_1, \cdots, x_n] \\ \qquad\qquad + (x - x_n) f[x, x_0, \cdots, x_n] \end{cases} \tag{5.18}$$

将式(5.18)中的第二式代入第一式的右端得

$$f[x] = f[x_0] + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x, x_0, x_1] \tag{5.19}$$

再将式(5.18)中的第三式代入式(5.19),依次由下向上逐步代入,最后可得

$$\begin{aligned} f(x) = {} & f(x_0) + (x - x_0) f[x_0, x_1] \\ & + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + \cdots \\ & + (x - x_0)(x - x_1) \cdots (x - x_{n-1}) f[x_0, x_1, \cdots, x_n] \\ & + (x - x_0)(x - x_1) \cdots (x - x_n) f[x, x_0, x_1, \cdots, x_n], \end{aligned} \tag{5.20}$$

记 $N_n(x) = f(x_0) + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) \cdots (x - x_{n-1}) f[x_0, x_1, \cdots, x_n]$ (5.21)

$$E_n(x) = (x - x_0)(x - x_1) \cdots (x - x_n) f[x, x_0, x_1, \cdots, x_n] \tag{5.22}$$

于是有 $$f(x) = N_n(x) + E_n(x).$$

可以证明 $N_n(x)$也是满足插值条件 $N_n(x_i) = f(x_i), (i = 0, 1, \cdots, n)$的函数 $f(x)$的 n 次代数插值多项式.(留给读者自己证明).

公式(5.21)称为 n 次牛顿插值多项式,公式(5.22)称为牛顿插值多项式的余项.

由于 $L_n(x) = N_n(x)$，所以 $f(x) - L_n(x) = f(x) - N_n(x)$，即 $R_n(x) = E_n(x)$.

这说明牛顿插值多项式的余项与拉格朗日插值多项式的余项也是相同的，只是书写形式不同而已 .

所以，当 $f(x)$在$[a, b]$上具有 $n+1$ 阶导数时，比较余项公式 $R_n(x)$和 $E_n(x)$可知

$$f[x, x_0, x_1, \cdots, x_n] = \frac{1}{(n+1)!} f^{(n+1)}(\xi), \quad \xi \in [a, b]$$

即证得差商性质(4).

牛顿插值公式(5.21)是 n 次插值多项式的又一种构造形式，但它克服了拉格朗日插值多项式的缺点，它的一个明显的优点是，每增加一个插值节点，只要在原牛顿插值公式中增添一项便可形成高一次的插值公式 .

例如由公式(5.21)

得
$$N_1(x) = f(x_0) + (x - x_0) f[x_0, x_1],$$
$$N_2(x) = f(x_0) + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2],$$

显然，$N_2(x) = N_1(x) + (x - x_0)(x - x_2) f[x_0, x_1, x_2]$,

上式指出了在求得一次插值 $N_1(x)$后，如果想再增加一个节点 x_2 计算 $N_2(x)$，则只要在 $N_1(x)$中增添项 $(x - x_0)(x - x_1) f[x_0, x_1, x_2]$就可以了 .

一般地，有递推关系式

$$N_{k+1}(x) = N_k(x) + (x - x_0)(x - x_1) \cdots (x - x_k) f[x_0, x_1, \cdots, x_{k+1}] \qquad (5.23)$$

式(5.23)中 $N_k(x)$为 $f(x)$在节点 $x_0, x_1, \cdots, x_k$ 上的 k 次牛顿插值公式，$N_{k+1}(x)$为 $f(x)$在节点 $x_0, x_1, \cdots, x_k, x_{k+1}$上的 $k+1$ 次牛顿插值公式 . 由于牛顿插值公式具有灵活增加节点的优点，故应用非常方便 .

从公式(5.21)中还可看到，牛顿插值公式中各项的系数就是

函数 $f(x)$的各阶差商，$f[x_0]$，$f[x_0,x_1]$，$\cdots$，$f[x_0,x_1,\cdots,x_n]$，因此，在构造牛顿插值公式时，常常先把差商列成一个表，此表称为差商表 .

四、差商表

我们可以把差商列成如下形状的表格，即差商表 .

差 商 表

k	x_k	$f(x_k)$	一阶差商	二阶差商	三阶差商	$\cdots$
0	x_0	$f(x_0)$				
			$f[x_0,x_1]$			
1	x_1	$f(x_1)$		$f[x_0,x_1,x_2]$		
			$f[x_1,x_2]$		$f[x_0,x_1,x_2,x_3]$	
2	x_2	$f(x_2)$		$f[x_1,x_2,x_3]$		$\cdots$
			$f[x_2,x_3]$		$f[x_1,x_2,x_3,x_4]$	
3	x_3	$f(x_3)$		$f[x_2,x_3,x_4]$		
			$f[x_3,x_4]$			
4	x_4	$f(x_4)$				
⋮	⋮	⋮	⋮	⋮	⋮	

在差商表中，前两列是数据表给出的值，从第三列起依次是各阶差商，计算可从左至右逐列进行 .

显然，$N_n(x)$的表达式(5.21)中各项的系数就是差商表中斜线上所对应的各阶差商，很有规律，按箭头所示，就容易写出各阶牛顿插值多项式 . 而且，由差商的性质 3 知，k 次多项式 $f(x)$的 k 阶差商为常数，由此可知，如果一个列表函数在差商表中的 k 阶差商已接近于常数，那么用 k 次多项式去近似它是恰当的 .

例 5.3 已知当 $x_i=1,2,3,4,5$ 时，对应的函数值 $f(x_i)=1,4,7,8,6$，求四次牛顿插值多项式 .

解 由给定的数据先作差商表

差 商 表

k	x_k	$f(x_k)$	一阶差商	二阶差商	三阶差商	四阶差商
0	1	1				
			3			
1	2	4		0		
			3		$-1/3$	
2	3	7		-1		$1/24$
			1		$-1/6$	
3	4	8		$-3/2$		
			-2			
4	5	6				

以表格中斜线上对应的数字为系数，照公式(5.21)写出所求的牛顿插值多项式为

$$N_4(x) = 1 + (x-1)\times 3 + (x-1)(x-2)\times 0$$
$$+ (x-1)(x-2)(x-3)\times\left(-\frac{1}{3}\right)$$
$$+ (x-1)(x-2)(x-3)(x-4)\times\frac{1}{24}$$
$$= \frac{1}{24}x^4 - \frac{9}{12}x^3 + \frac{83}{24}x^2 - \frac{33}{12}x + 1.$$

5.4 差分与等距节点插值公式

在实际应用中，经常会遇到插值节点是等距分布的情况，这时，牛顿插值公式可以进一步简化，得到所谓等距节点的插值公式.

一、差分的概念与差分表

设已知函数 $y=f(x)$在等距节点 $x_i=x_0+ih(i=0,1,2,\cdots,n)$上的函数值为 $f(x_i)=f_i$,式中 h 称为步长 .

定义 5.2 $\Delta f_i=f(x_i+h)-f(x_i)=f_{i+1}-f_i$ 称为函数 $f(x)$在点 x_i 处步长为 h 的**一阶差分** .

一阶差分的差分

$$\Delta^2 f_i=\Delta(\Delta f_i)=\Delta f_{i+1}-\Delta f_i=f_{i+2}-2f_{i+1}+f_i,$$

称为函数 $f(x)$在点 x_i 处的**二阶差分** .

一般地,把 $n-1$ 阶差分的差分定义为 n 阶差分,记为

$$\Delta^n f_i=\Delta^{n-1}f_{i+1}-\Delta^{n-1}f_i,\ (n=2,3,\cdots) \tag{5.24}$$

并规定 $\Delta^0 f_i=f_i$ 为 $f(x)$在 x_i 处的**零阶差分** .

以上定义的各阶差分也统称为**向前差分**,符号 Δ 叫做**向前差分算子** .

还可定义向后差分:

称 $\nabla f_i=f(x_i)-f(x_i-h)=f_i-f_{i-1}$ 为 $f(x)$在 x_i 处步长为 h 的**一阶向后差分** .

称 $\nabla^2 f_i=\nabla(\nabla f_i)=\nabla f_i-\nabla f_{i-1}=f_i-2f_{i-1}+f_{i-2}$ 为 $f(x)$在 x_i 处的**二阶向后差分** .

一般地,n 阶向后差分定义为

$$\nabla^n f_i=\nabla^{n-1}f_i-\nabla^{n-1}f_{i-1},\ (n=2,3,\cdots)$$

向前差分与向后差分只是记写方式上的不同,其实,容易看出它们之间存在关系:

$$\nabla^k f_i=\Delta^k f_{i-k},\ (k\geqslant 1)$$

例如,当 $k=2$ 时,$\nabla^2 f_i=f_i-2f_{i-1}+f_{i-2}$, $\Delta^2 f_{i-2}=f_i-2f_{i-1}+f_{i-2}$是相等的 .

计算差分时,也可以像计算差商一样列成一个表格,称为差分表 .

向前差分表

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$	$\cdots$
x_0	f_0					
		Δf_0				
x_1	f_1		$\Delta^2 f_0$			
		Δf_1		$\Delta^3 f_0$		
x_2	f_2		$\Delta^2 f_1$		$\Delta^4 f_0$	
		Δf_2		$\Delta^3 f_1$		
x_3	f_3		$\Delta^2 f_2$			
		Δf_3				
x_4	f_4					

斜线上是 $f(x)$在 x_0 处的各阶向前差分 $\Delta f_0, \Delta^2 f_0, \Delta^3 f_0, \cdots$，差分表比差商表的优点是不需作除法运算 .

二、差分与差商的关系

为了推导等距节点的插值公式，需要知道差分与差商的关系 . 现介绍如下 .

在 $x_i = x_0 + ih(i = 0, 1, 2, \cdots)$时，由差分与差商的定义，很容易得到各阶差分与差商之间存在以下关系：

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{\Delta f_0}{h} = \frac{\Delta f_0}{1! \cdot h},$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

$$= \frac{\frac{\Delta f_1}{h} - \frac{\Delta f_0}{h}}{2h} = \frac{\Delta^2 f_0}{2! \cdot h^2}$$

一般地，k 阶差商与 k 阶向前差分的关系为

$$f[x_0, x_1, \cdots, x_k] = \frac{\Delta^k f_0}{k!\ h^k}, \ (k = 1, 2, \cdots, n) \qquad (5.25)$$

同理，k 阶差商与 k 阶向后差分之间也有如下关系：

$$f[x_n, x_{n-1}, \cdots, x_{n-k}] = \frac{\nabla^k f_n}{k!\ \cdot h^k}, \ (k = 1, 2, \cdots, n) \quad (5.26)$$

三、等距节点的插值公式

当节点的分布是等距时，利用上述差分与差商的关系，就可得到牛顿插值多项式的几种变形，其计算更为简单 .

1. 牛顿向前插值公式

令 $x_i = x_0 + ih (i = 0, 1, 2, \cdots, n)$ 为等距插值节点，则牛顿插值公式可写为

$$N_n(x) = f_0 + \frac{x - x_0}{1!\ h}\Delta f_0 + \frac{(x - x_0)(x - x_1)}{2!\ h^2}\Delta^2 f_0 + \cdots + \frac{(x - x_0)(x - x_1)\cdots(x - x_{n-1})}{n!\ h^n}\Delta^n f_0, \qquad (5.27)$$

当 $x_0 \leqslant x \leqslant x_1$ 时，令 $x = x_0 + th \quad (0 \leqslant t \leqslant 1)$

即得

$$N_n(x) = N_n(x_0 + th) = f_0 + \frac{t}{1!}\Delta f_0 + \frac{t(t-1)}{2!}\Delta^2 f_0 + \cdots + \frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n f_0, \qquad (5.28)$$

这个用各阶向前差分表示的插值公式(5.28)就称为牛顿向前插值公式 .

其余项可由拉格朗日余项推导得

$$R_n(x) = R_n(x_0 + th)$$

$$= \frac{t(t-1)(t-2)\cdots(t-n)}{(n+1)!}h^{n+1} \cdot f^{(n+1)}(\zeta), \quad \zeta \in (x_0, x_n) \qquad (5.29)$$

2. 牛顿向后插值公式

当 $x_{n-1} \leqslant x \leqslant x_n$ 时，令 $x = x_n + th(-1 \leqslant t \leqslant 0)$，利用向后差分与差商的关系代入牛顿插值公式可推得用各阶向后差分表示的牛顿向后插值公式：

$$\begin{aligned} N_n(x) &= N_n(x_n + th) \\ &= f_n + \frac{t}{1!}\nabla f_n + \frac{t(t+1)}{2!}\nabla^2 f_n + \cdots \\ &\quad + \frac{t(t+1)\cdots(t+n-1)}{n!}\nabla^n f_n, \end{aligned} \tag{5.30}$$

其余项为

$$\begin{aligned} R_n(x) &= R_n(x_n + th) \\ &= \frac{h^{n+1}}{(n+1)!}t(t+1)\cdots(t+n)f^{(n+1)}(\zeta),\ \zeta \in (x_0, x_n) \end{aligned} \tag{5.31}$$

运用牛顿向前、向后插值公式进行计算时，可先构造差分表 .

5.5 埃尔米特(Hermite)插值

前面介绍的代数插值，只要求插值多项式 $p(x)$ 在插值节点处与被插值函数 $f(x)$ 有相同的函数值 $p(x_i) = f(x_i)(i = 0, 1, 2, \cdots, n)$，但是这种插值多项式往往还不能全面反映被插值函数 $f(x)$ 的性态，许多实际问题不但要求插值函数与 $f(x)$ 在各节点的函数值相同，而且还要求插值多项式在某些节点或全部节点上与 $f(x)$ 的导数值也相等，甚至要求高阶导数值也相等，能满足这种要求的插值问题就称为**埃尔米特插值问题** .

在本节中，我们不去讨论一般的埃尔米特插值问题，而只是讨论下面这种最简单的也是最常用的情形 .

定义 5.3 设已知函数 $f(x)$ 在插值区间 $[a, b]$ 上 $n+1$ 个互异的节点 $x_i(i = 0, 1, \cdots, n)$ 处的函数值 $f(x_i) = f_i$ 及一阶导数值 $f'(x_i) = f'_i(i = 0, 1, 2, \cdots, n)$，若存在函数 $H(x)$ 满足条件：

(1) $H(x)$是一个次数不超过 $2n+1$ 次的多项式;

(2) $H(x_i)=f(x_i)$, $H'(x_i)=f'(x_i)$ $(i=0,1,2,\cdots n)$.

则称 $H(x)$为 $f(x)$在 $n+1$ 个节点 x_i 上的**埃尔米特插值多项式**.

这是最简单的埃尔米特插值问题,为了讨论方便,下面简称为埃尔米特插值问题(1),(2).

这个问题的几何意义是十分明显的,就是要求代数曲线 $y=H(x)$与函数曲线 $y=f(x)$不仅在 $n+1$ 个互异的节点 x_i 处完全重合,而且还要求在 x_i 处有公切线.

定理 5.4 埃尔米特插值问题(1)(2)的解 $H(x)$是存在且唯一的.

证明 Ⅰ存在性

(先分析)我们从构造拉格朗日插值多项式得到启发,它是先构造 $n+1$ 个特殊的 n 次多项式 $l_k(x)(k=1,2,\cdots,n)$(拉格朗日基函数)满足条件:$l_k(x_i)=\delta_{ki}=\begin{cases}0, k\neq i\\ 1, k=i\end{cases}$,然后就很容易得到拉格朗日插值多项式 $L_n(x)=\sum\limits_{k=0}^{n}l_k(x)f_k$.

如果现在也能构造两组次数都是 $2n+1$ 次的特殊多项式,记为

$$h_k(x)\text{和 }\bar{h}_k(x)\quad (k=0,1,2,\cdots,n)$$

使它们分别满足条件:

$$\begin{aligned}&h_k(x_i)=\delta_{ki},\ h_k'(x_i)=0,\\ &\bar{h}_k(x_i)=0,\ \bar{h}_k'(x_i)=\delta_{ki},\ (i,k=0,1,2,\cdots,n)\end{aligned}\tag{5.32}$$

这里,$h_k(x)$, $\bar{h}_k(x)$也称为插值基函数,共有 $2n+2$ 个.

$$\text{显然,}H(x)=\sum_{k=0}^{n}h_k(x)\cdot f_k+\sum_{k=0}^{n}\bar{h}_k(x)\cdot f_k',\tag{5.33}$$

就是满足插值条件(1),(2)的埃尔米特插值问题(1),(2)的解,也即满足插值问题条件(1),(2)的埃尔米特插值多项式 $H(x)$可写

成上述这种用插值基函数表示的形式(5.33).

这是因为 $h_k(x)$, $\bar{h}_k(x)$都是 $2n+1$ 次多项式,则由它们的线性组合得到的 $H(x)$的表达式(5.33)必是一个次数不超过 $2n+1$ 次多项式函数 . 所以由式(5.33)确定的 $H(x)$满足埃尔米特插值问题条件(1). 再由条件(5.32),显然有

$$H(x_i)=\sum_{k=0}^{n}h_k(x_i)f_k+\sum_{k=0}^{n}\bar{h}_k(x_i)f'_k$$

$$=\sum_{k=0}^{n}\delta_{ki}\cdot f_k+0=f_i,$$

$$H'(x_i)=\sum_{k=0}^{n}h'_k(x_i)\cdot f_k+\sum_{k=0}^{n}\bar{h}'_k(x_i)f'_k$$

$$=0+\sum_{k=0}^{n}\delta_{ki}\cdot f'_k=f'_i,$$

即由式(5.33)确定的 $H(x)$满足埃尔米特插值问题条件(2),因此下面要解决的问题归结为如何构造满足条件(5.32)的 $2n+2$ 个插值基函数 $h_k(x)$和 $\bar{h}_k(x)(k=0,1,2,\cdots,n)$.

(一) 构造 $h_k(x)(k=0,1,2,\cdots,n)$

由条件 $h_k(x_i)=h'_k(x_i)=0(i\neq k)$知, x_i(当 $i\neq k$ 时)都是 $h_k(x)$的两重零点,而已知拉格朗日插值基函数 $l_k(x)$是 n 次多项式,且具有性质 $l_k(x_i)=\delta_{ki}$,则 $2n$ 次多项式$[l_k(x)]^2$ 也具有性质 $[l_k(x_i)]^2=\delta_{ki}$,而 $[l_k(x)]^2$ 的一阶导数在 $x_i(i\neq k)$处的值 $([l_k(x_i)]^2)'=2l_k(x_i)l'_k(x_i)=0$,所以当 $i\neq k$ 时, x_i 也都是 $[l_k(x)]^2$的两重零点,注意到 $h_k(x)$是 $2n+1$ 次多项式,而 $[l_k(x)]^2$是 $2n$ 次多项式,因此可设 $h_k(x)=(ax+b)l_k^2(x)(k=0,1,2,\cdots,n)$,其中 a,b 为待定常数 . 显然,由上述讨论知,当 $i\neq k$ 时, $h_k(x)$已满足条件(5.32)中的 $h_k(x_i)=h'_k(x_i)=0$,现在只要适当选取常数 a,b,使当 $i=k$ 时, $h_k(x)$也满足条件(5.32),即满足

$$\begin{cases}h_k(x_k)=1\\ h'_k(x_k)=0\end{cases}$$

由此两个条件立即可得到确定 a,b 的两个方程：

$$\begin{cases} h_k(x_k) = (ax_k + b)l_k^2(x_k) = ax_k + b = 1 \\ h_k'(x_k) = 2l_k(x_k) \cdot l_k'(x_k)(ax_k + b) \\ \qquad\qquad + al_k^2(x_k) = 2l_k'(x_k) + a = 0 \end{cases}$$

即

$$\begin{cases} ax_k + b = 1 \\ a + 2l_k'(x_k) = 0 \end{cases}$$

解出

$$\begin{cases} a = -2l_k'(x_k) \\ b = 1 + 2l_k'(x_k) \cdot x_k \end{cases}$$

于是

$$h_k(x) = [1 - 2l_k'(x_k)(x - x_k)]l_k^2(x).$$

（二）构造 $\bar{h}_k(x)(k = 0,1,2,\cdots,n)$

类似于上述做法，由条件 $\bar{h}_k(x_i) = \bar{h}_k'(x_i) = 0(i \neq k)$ 知 $x_i(i \neq k)$ 是 $\bar{h}_k(x)$ 的两重零点，因此可设 $\bar{h}_k(x)$ 也含有因式 $l_k^2(x)$，又 $\bar{h}_k(x_k) = 0$，所以 x_k 是 $\bar{h}_k(x)$ 的零点，即 $\bar{h}_k(x)$ 还含有因式 $(x - x_k)$，因此设 $\bar{h}_k(x) = A(x - x_k)l_k^2(x)$，其中 A 为待定常数．

显然，$\bar{h}_k(x)$ 是 $2n + 1$ 次多项式，且当 $i \neq k$ 时已满足条件 (5.32)，由现条件 $\bar{h}_k'(x_k) = 1$ 可确定 A 如下：

$$\begin{aligned} \bar{h}_k'(x_k) &= Al_k^2(x_k) + A(x_k - x_k) \\ &\quad \cdot 2l_k(x_k) \cdot l_k'(x_k) = A = 1, \end{aligned}$$

所以

$$\bar{h}_k(x) = (x - x_k)l_k^2(x).$$

到此为止，已构造出 $2n + 2$ 个插值基函数 $h_k(x)$ 和 $\bar{h}_k(x)$，则将其代入 $H(x)$ 的表达式 (5.33)，即得满足插值条件 (1)，(2) 的埃尔米特插值问题的解：

$$\begin{aligned} H(x) = &\sum_{k=0}^{n} [1 - 2l_k'(x_k)(x - x_k)]l_k^2(x)f_k \\ &+ \sum_{k=0}^{n} (x - x_k)l_k^2(x)f_k', \end{aligned} \tag{5.34}$$

以上是用构造性的方法证明了埃尔米特插值问题 (1)，(2) 的解

$H(x)$的存在性，且具体地给出了满足条件(1)，(2)的插值多项式$H(x)$的求法．下面用反证法证明埃尔米特插值问题(1)，(2)的解是唯一的．

Ⅱ　唯一性：

假设另有一个$2n+1$次多项式$\bar{H}(x)$也满足插值条件(1)，(2)，则有

$$H(x_i)=\bar{H}(x_i)=f_i,$$

$$H'(x_i)=\bar{H}'(x_i)=f_i',(i=0,1,2,3,\cdots,n)$$

于是　$\varphi(x)=H(x)-\bar{H}(x)$也是次数不超过$2n+1$次的多项式

且　$\varphi(x_i)=H(x_i)-\bar{H}(x_i)=f_i-f_i=0,$

$$\varphi'(x_i)=H'(x_i)-\bar{H}'(x_i)$$

$$=f_i'-f_i'=0,(i=0,1,2,\cdots,n)$$

即$\varphi(x)$在每个节点x_i上都有两重根，从而$\varphi(x)$至少有$2n+2$个根(包括重根)，但$\varphi(x)$是不超过$2n+1$次的多项式，要使它有$2n+2$个根，除非其恒为零，即$\varphi(x)\equiv 0$，则$H(x)\equiv\bar{H}(x)$．唯一性得证．

用埃尔米特插值多项式$H(x)$近似代替函数$f(x)$，在非节点处，一般说来也是有误差的．类似于拉格朗日插值余项定理的证明，可以证得下述定理：

定理 5.5　若$f(x)$在插值区间$[a,b]$内有$2n+2$阶导数$f^{(2n+2)}(x)$，则对于任何$x\in[a,b]$，埃尔米特插值问题(1)，(2)的余项为

$$R(x)=f(x)-H(x)=\frac{f^{(2n+2)}(\xi)}{(2n+2)!}[\pi(x)]^2,\qquad(5.35)$$

其中　$\pi(x)=\prod_{j=0}^{n}(x-x_j)$，$\xi\in(a,b)$且$\xi$与$x$有关．

实际中使用最广泛的是三次埃尔米特插值多项式，即在式(5.34)中令$n=1$，得

$$H_3(x)=\left(1+2\frac{x-x_0}{x_1-x_0}\right)\left(\frac{x-x_1}{x_0-x_1}\right)^2 f_0$$

$$+\left(1+2\frac{x-x_1}{x_0-x_1}\right)\left(\frac{x-x_0}{x_1-x_0}\right)^2 f_1$$
$$+(x-x_0)\left(\frac{x-x_1}{x_0-x_1}\right)^2 f_0'$$
$$+(x-x_1)\left(\frac{x-x_0}{x_1-x_0}\right)^2 f_1', \qquad (5.36)$$

其误差为

$$R_3(x)=f(x)-H_3(x)$$
$$=\frac{1}{4!}f^{(4)}(\xi)(x-x_0)^2(x-x_1)^2,\ \xi\in(x_0,x_1) \qquad (5.37)$$

还须指出,埃尔米特插值问题解的形式可以是多种形式的,插值条件也可由多种形式给出,下面举例说明 .

例 5.4 若 $f(x)$在$[a,b]$上有三阶连续导数,且已知 $f(x)$在$[a,b]$上两个互异的点 x_0,x_1 上的函数值 $f(x_0),f(x_1)$和一阶导数值 $f'(x_0)$. 试求满足条件

$$H(x_0)=f(x_0),\ H(x_1)=f(x_1),\ H'(x_0)=f'(x_0)$$

的插值多项式,并估计其误差 .

解 由给定的 3 个插值条件,显然可确定一个次数不超过 2 次的埃尔米特插值多项式 $H(x)$. 又由 $H(x)$应满足插值条件 $H(x_i)=f(x_i)(i=0,1)$,而节点 x_0,x_1 上的线性插值函数 $N_1(x)$也满足插值条件 $N_1(x_i)=f(x_i)(i=0,1)$,故可设

$$H(x)-N_1(x)=A(x-x_0)(x-x_1),$$

其中 A 为待定常数,上式又可记为

$$H(x)=N_1(x)+A(x-x_0)(x-x_1)$$
$$=f(x_0)+(x-x_0)f[x_0,x_1]+A(x-x_0)(x-x_1), \qquad (5.38)$$

为了确定常数 A,对上式求导,得

$$H'(x)=f[x_0,x_1]+A[(x-x_1)+(x-x_0)],$$

令 $x = x_0$ 代入，且注意插值条件 $H'(x_0) = f'(x_0)$，得

$$H'(x_0) = f[x_0, x_1] + A(x_0 - x_1) = f'(x_0)$$

于是有

$$A = \frac{f[x_0, x_1] - f'(x_0)}{x_1 - x_0},$$

将上式代入式(5.38)，即得所求的插值多项式 $H(x)$ 为

$$H(x) = f(x_0) + (x - x_0)f[x_0, x_1] + \frac{f[x_0, x_1] - f'(x_0)}{x_1 - x_0}(x - x_0)(x - x_1),$$

因为上式是采用牛顿插值多项式构造的，所以也称为牛顿-埃尔米特插值多项式 .

当然也可先采用拉格朗日多项式构造，得到满足相同插值条件的 $H(x)$ 的另一种形式：

$$H(x) = \left(\frac{x - 2x_0 + x_1}{x_1 - x_0}\right)\left(\frac{x_1 - x}{x_1 - x_0}\right)f(x_0) + \left(\frac{x - x_0}{x_1 - x_0}\right)^2 f(x_1) + \left(\frac{x_1 - x}{x_1 - x_0}\right)(x - x_0)f'(x_0)$$

上式也称为拉格朗日-埃尔米特插值多项式 .

类同于埃尔米特插值的误差估计推导过程可作辅助函数

$$g(t) = f(t) - H(t) - K(x)(t - x_0)^2(t - x_1),$$

则 $g^{(3)}(t) = f^{(3)}(t) - 6K(x)$，且 $g(t)$ 在 $[a, b]$ 上有互异的零点 x_0, x_1(两重)，x，反复运用罗尔定理，知在 (a, b) 内至少有一点使 $g^{(3)}(\xi) = 0$，则 $K(x) = \frac{f^{(3)}(\xi)}{6}$，于是得余项 $R(x) = \frac{1}{6}(x - x_0)^2(x - x_1)f^{(3)}(\xi)$.

5.6 样条插值函数

我们已经知道，当选取的插值节点较多时，一般不用高次($n > 7$)插值，而常改用分段低次插值 . 但是分段低次插值，只能

保证插值曲线在连接点上的连续性，而不能保证整条曲线在连接点处的光滑性，即导数不一定连续，如分段线性插值，其图形是锯齿形的折线，它虽然连续，但一阶导数在点 $x_i(i=1,2,\cdots,n-1)$ 处都不存在，因而极不光滑，这在实用上，往往不能满足某些工程技术的高精度要求．例如在船体、飞机等外形曲线的设计中，不仅要求曲线连续，而且还要求曲线的曲率连续．这就要求分段插值函数在整个区间上具有连续的二阶导数，因此有必要引进新的插值方法．近年来，分段多项式逼近在理论研究和工程应用中获得重大的进展，其中在具有收敛性与稳定性的插值函数中最常用和最重要的是样条函数插值．

一、样条插值函数的概念

样条(Spline)这个词本来是指在飞机和船舶的制造过程中为了描绘出光滑的外形曲线所用的一种绘图工具，它是一种富有弹性的细长条，使用时，用压铁固定在一些给定的点(称为样点)上，在其他地方让它自由弯曲，然后依样画下光滑的曲线，就称为样条曲线，从数学上看，在小挠度的假定下，它实际上是由一段一段的三次多项式曲线拼接而成的曲线，在拼接处，不仅函数自身是连续的，而且它的一阶导数和二阶导数也是连续的(可是三阶导数一般却不是连续的)．所以样条曲线具有非常好的光滑性．从数学上加以概括就得到样条函数这一概念．

自 1946 年舒恩柏格(I. J. Schoenberg)首次提出样条函数这一名称到现在已将近有 50 年历史了．但一开始样条函数并没有得到应有的重视，只是在电子计算机的迅速发展和样条函数也找到了很多重要的应用以后，它才得到了蓬勃发展，建立了完备的理论．自60 年代以来，样条函数越来越流行，它不仅是现代函数逼近的一个活跃的分支，而且也是现代数学计算中一个十分重要的数学工具．它已以各种方式应用到逼近论、数据拟合、数值微分、数值积分、微分方程和积分方程等数值求解中．尽管样条函数内

容十分丰富，应用非常广泛，但在这里我们只介绍最简单的，然而也是最常用的“三次样条插值函数”.

定义 5.4 对于给定的函数表

x_i	x_0	x_1	$\cdots$	x_n
$y_i=f(x_i)$	y_0	y_1	$\cdots$	y_n

其中 $$a=x_0<x_1<x_2<\cdots<x_n=b$$

若存在函数 $s(x)$满足条件：

(1) $s(x)$在每一个子区间$[x_{j-1},x_j](j=1,2,\cdots,n)$上是一个不超过三次多项式 .

(2) $s(x)$在每一个内接点 $x_j(j=1,2,\cdots,n-1)$上具有直到二阶的连续导数，即 $s(x)\in C^2[a,b]$，则称 $s(x)$为节点 $x_0,x_1,\cdots,x_n$ 上的三次样条函数 .

若 $s(x)$在所有节点上还满足插值条件：

(3) $s(x_j)=y_j(j=0,1,2,\cdots,n)$

则称 $s(x)$为三次样条插值函数 .

由此可见，三次样条插值函数就是全部通过样点的二阶连续可微的分段三次多项式函数 . 但要注意，$s(x)$在整个$[a,b]$上不是一个三次多项式函数 . 下面的问题是要解决满足上述定义的样条插值函数是否存在 .

从样条函数的定义 5.4 的条件(1)知，$s(x)$在每一个小区间$[x_{j-1},x_j]$上是一个三次多项式，若记为 $s_j(x)$，则可设

$$s_j(x)=a_jx^3+b_jx^2+c_jx+d_j,\ (j=1,2,\cdots,n)$$

从而有

$$s(x)=\begin{cases}s_1(x)\\ s_2(x)\\ \vdots\\ s_n(x)\end{cases}$$

这是一个分段函数 .

显然,$s(x)$由 a_j, b_j, c_j, d_j 四个系数唯一确定,因此要确定整个$[a, b]$上的三次样条插值函数 $s(x)$,必须确定 $4n$ 个未知系数 $\{a_j, b_j, c_j, d_j\}(j = 1,2,\cdots, n)$,现在我们来分析一下这 $4n$ 个未知系数是否能由已知条件来确定 .

首先,由定义 5.4 的条件(2)表明:$s(x), s'(x), s''(x)$在内节点 $x_1, x_2, \cdots, x_{n-1}$上是连续的,于是有

$$s(x_j - 0) = s(x_j + 0),\ s'(x_j - 0) = s'(x_j + 0),$$
$$s''(x_j - 0) = s''(x_j + 0)\ (j = 1,2,\cdots, n-1).$$

由于这样的等式共有 $3(n-1)$个,从而提供了包含未知系数 a_j, b_j, c_j, d_j 的 $3n-3$ 个方程,而定义 5.4 中的条件(3)——$s(x_j) = y_j$ $(j = 0,1,2,\cdots, n)$则提供了 $n+1$ 个方程,这样,一共可以得到 $4n$ -2 个方程,与要确定的 $4n$ 个未知系数相比较,还差两个条件,所以对于给定的插值函数表,按定义 5.4 所得到的样条插值函数 $s(x)$中含有两个自由度 . 为了确定一个特定的样条插值函数,还需要增加两个条件,这两个条件通常是在区间$[a, b]$的两个端点处给出,我们把它叫做边界(或端点)条件 . 从力学角度考虑,附加边界条件相当于在细梁的两端加上约束,是有意义的 .

边界条件应根据实际问题的要求来确定,其类型很多,常见的边界条件类型如下:

(1) 第一种边界条件:给定端点处的一阶导数值,即

$$s'(x_0) = y'(x_0),\ s'(x_n) = y'(x_n)$$

(2) 第二种边界条件:给定端点处的二阶导数值,即

$$s''(x_0) = y''(x_0),\ s''(x_n) = y''(x_n)$$

作为特例,$s''(x_0) = s''(x_n) = 0$ 称为自然边界条件,满足自然边界条件的样条函数叫做**自然样条函数**,它是通过所有数据点的插值函数中,是曲率最小的唯一函数,因此自然三次样条是插值所有数据点的最光滑的函数 .

(3) 第三种边界条件：当 $y=f(x)$ 是以 $b-a$ 为周期的周期函数时，则要求 $s(x)$ 也是周期函数，这时的边界条件可写成

$$s(x_0+0)=s(x_n-0),\ s'(x_0+0)=s'(x_n-0),$$

$$s''(x_0+0)=s''(x_n-0)$$

注意：由于 $y=f(x)$ 以 $b-a$ 为周期，故 $f(a)=f(b)$，即

$$y_0=y_n$$

从而必有 $$s(x_0-0)=s(x_n-0)$$

因此，在第三种边界条件中，真正起作用的是后面的两个等式．

补充了两个端点条件即增加了两个方程，也即得到了关于未知系数 $\{a_j,b_j,c_j,d_j\}$ 的 $4n$ 个方程，一般可定出 $4n$ 个未知参数．可以证明，由以上推得的含有 $4n$ 个未知元 $\{a_j,b_j,c_j,d_j\}$ 的线性代数方程组存在唯一解．所以插值函数 $s(x)$ 是唯一确定的．理论上，只要求解 $4n$ 阶线性代数方程组即可得到 $s(x)$ 在各个小区间上的表达式 $s_j(x)$ 的各项系数，从而得到 $s(x)$．但实际上，这种做法的工作量相当大，且上述方程组一般是病态的，所以，上述这种将样条插值函数的求解归结为求解 $4n$ 个方程组成的线性代数方程组的方法在实际计算中很少应用．下面我们将介绍一种计算工作量小得多的切实可行的构造 $s(x)$ 的有效方法．

二、三次样条插值函数的求法

求三次样条插值函数 $s(x)$ 有多种方法，常用的有三弯矩法和三转角法．下面重点介绍求解三次样条插值函数的三弯矩法．

设 $s(x)$ 在节点 $x_j(j=0,1,2,\cdots,n)$ 处的二阶导数为

$$s''(x_j)=M_j(j=0,1,2,\cdots,n),$$

令 $h_j=x_j-x_{j-1}$，由于 $s(x)$ 在每个子区间 $[x_{j-1},x_j]$ 上是一个三次多项式 $s_j(x)$，故 $s_j''(x)$ 在区间 $[x_{j-1},x_j]$ 上是 x 的线性函数，且因为 $s_j''(x_{j-1})=M_{j-1}$，$s_j''(x_j)=M_j$ 用线性插值，可知其表达式为

$$s_j''(x)=\frac{x_j-x}{h_j}M_{j-1}+\frac{x-x_{j-1}}{h_j}M_j$$

经两次积分得

$$s_j'(x) = -\frac{(x_j - x)^2}{2h_j}M_{j-1} + \frac{(x - x_{j-1})^2}{2h_j}M_j + c_1$$

$$s_j(x) = \frac{(x_j - x)^3}{6h_j}M_{j-1} + \frac{(x - x_{j-1})^3}{6h_j}M_j + c_1 x + c_2,$$

由插值条件 $s_j(x_{j-1}) = y_{j-1}, s_j(x_j) = y_j$ 确定两个积分常数 c_1 和 c_2,从而得到 $s(x)$在小区间$[x_{j-1}, x_j]$上的表达式为

$$\begin{aligned} s_j(x) = &\frac{(x_j - x)^3}{6h_j}M_{j-1} + \frac{(x - x_{j-1})^3}{6h_j}M_j \\ &+ \left(y_{j-1} - \frac{M_{j-1}h_j^2}{6}\right)\frac{x_j - x}{h_j} \\ &+ \left(y_j - \frac{M_j h_j^2}{6}\right)\frac{x - x_{j-1}}{h_j}, (j = 1, 2, \cdots, n) \end{aligned} \tag{5.39}$$

由此可知,只要确定 $M_0, M_1, \cdots, M_n$ 这 $n+1$ 个待定参数,就可以定出三次样条插值函数 $s(x)$. 并由以上构造过程知,这时的 $s(x)$保证了逐段三次插值,即 $s(x)$的连续性,并保证了 $s''(x)$在节点的连续性 .

为了确定待定参数 $M_0, M_1, \cdots, M_n$,我们利用函数 $s(x)$在插值区间$[a, b]$上各内节点 $x_j (j = 1, 2, \cdots, n-1)$处有一阶连续导数的条件:$s'(x_j - 0) = s'(x_j + 0)$,为此,对 $s(x)$求导一次,得在区间$[x_{j-1}, x_j]$上的表达式为

$$s_j'(x) = -\frac{(x_j - x)^2}{2h_j}M_{j-1} + \frac{(x - x_{j-1})^2}{2h_j}M_j + \frac{y_j - y_{j-1}}{h_j} - \frac{M_j - M_{j-1}}{6}h_j$$

从而在右端点 x_j 上有

$$s_j'(x_j - 0) = \frac{h_j}{2}M_j - \frac{h_j}{6}(M_j - M_{j-1}) + \frac{y_j - y_{j-1}}{h_j}$$

$$= \frac{h_j}{6} M_{j-1} + \frac{h_j}{3} M_j + \frac{y_j - y_{j-1}}{h_j}, \tag{5.40}$$

在左端点 x_{j-1} 上有

$$s_j'(x_{j-1}+0) = -\frac{h_j}{2} M_{j-1} - \frac{h_j}{6}(M_j - M_{j-1}) + \frac{y_j - y_{j-1}}{h_j}$$

$$= -\frac{h_j}{3} M_{j-1} - \frac{h_j}{6} M_j + \frac{y_j - y_{j-1}}{h_j}$$

则将上式中的 $j-1$ 改为 j，即得

$$s_{j+1}'(x_j+0) = -\frac{h_{j+1}}{3} M_j - \frac{h_{j+1}}{6} M_{j+1} + \frac{y_{j+1} - y_j}{h_{j+1}}, \tag{5.41}$$

利用 $s'(x)$ 在内接点的连续性，即 $s_j'(x_j - 0) = s_{j+1}'(x_j + 0)$ 就得到关于参数 M_{j-1}, M_j, M_{j+1} 的一个方程：

$$\frac{h_j}{6} M_{j-1} + \frac{h_j + h_{j+1}}{3} M_j + \frac{h_{j+1}}{6} M_{j+1} = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j},$$

$$(j = 1, 2, \cdots, n-1). \tag{5.42}$$

这就是在 $n-1$ 个内结点 $x_j (j = 1, 2, \cdots, n-1)$ 处关于 $n+1$ 个待定参数 $M_j (j = 0, 1, 2, \cdots, n)$ 的 $n-1$ 个方程式，为了进一步化简，对式(5.42)两边乘上 $6/(h_j + h_{j+1})$ 并由差商

$$f[x_{j-1}, x_j, x_{j+1}] = \frac{1}{h_j + h_{j+1}} \left\{ \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j} \right\}$$

可将式(5.42)化为

$$\frac{h_j}{h_j + h_{j+1}} M_{j-1} + 2M_j + \frac{h_{j+1}}{h_j + h_{j+1}} M_{j+1} = 6f[x_{j-1}, x_j, x_{j+1}],$$

$$(j = 1, 2, \cdots, n-1). \tag{5.43}$$

令 $\lambda_j = \dfrac{h_j}{h_j + h_{j+1}}$，$\mu_j = 1 - \lambda_j = \dfrac{h_j}{h_j + h_{j+1}}$，$d_j = 6f[x_{j-1}, x_j, x_{j+1}]$，

则 $n-1$ 个方程(5.43)可化为

$$\mu_j M_{j-1} + 2M_j + \lambda_j M_{j+1} = d_j, (j = 1,2,\cdots,n-1) \quad (5.44)$$

由于力学上将 M_j 解释为梁在截面 x_j 处的弯矩,故式(5.44)又称为三弯矩方程 . $M_j(j=0,1,\cdots,n)$称为 s 的矩 . 因为式(5.44)是关于 $n+1$ 个待定参数 $M_0, M_1, \cdots, M_n$ 的 $n-1$ 方程,所以有无穷多组解 . 但实际问题只能选取特定的一个解,所以要完全确定 M_j $(j=0,1,2,\cdots,n)$的值还需补充两个条件,这两个条件通常根据实际问题的要求,根据插值区间$[a,b]$的两个端点处的边界条件来补充 .

例如(1)第一种边界条件:即已知插值区间两端的一阶导数值:$s'(x_0)=y_0', s'(x_n)=y_n'$

我们可令 $j=0$,由式(5.41)得

$$s_1'(x_0+0) = s'(x_0) = -\frac{h_1}{3}M_0 - \frac{h_1}{6}M_1 + \frac{y_1-y_0}{h_1} = y_0'$$

两边乘$\left(-\frac{6}{h_1}\right)$,整理得

$$2M_0 + M_1 = \frac{6}{h_1}\left(\frac{y_1-y_0}{h_1} - y_0'\right) \xlongequal{\text{记}} d_0 \quad (5.45)$$

同样由式(5.40)中令 $j=n$ 得

$$s_n'(x_n-0) = s'(x_n) = \frac{h_n}{6}M_{n-1} + \frac{h_n}{3}M_n + \frac{y_n-y_{n-1}}{h_n} = y_n',$$

两边乘$\left(\frac{6}{h_n}\right)$,整理得

$$M_{n-1} + 2M_n = \frac{6}{h_n}\left(y_n' - \frac{y_n-y_{n-1}}{h_n}\right) \xlongequal{\text{记}} d_n \quad (5.46)$$

将式(5.44)、式(5.45)和式(5.46)合在一起,即得确定 $n+1$

个参数 $M_0, M_1, \cdots, M_n$ 的 $n+1$ 阶线性代数方程组为

$$\begin{bmatrix} 2 & 1 & & & & & \\ \mu_1 & 2 & \lambda_1 & & & & \\ & \mu_2 & 2 & \lambda_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} & \\ & & & & 1 & 2 & \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \tag{5.47}$$

由于方程组的系数矩阵的非零元素都集中在三条对角线上，所以方程组(5.47)被称为三对角线方程组，这种三对角方程组的系数矩阵元素 $\mu_j + \lambda_j = 1$，且 $\lambda_j \geqslant 0, \mu_j \geqslant 0$，故它是严格占优的．可用第一章介绍的"追赶法"求解．求得 M_j 后，再由式(5.39)就可求得 $s_j(x)$，从而求得 $s(x)$.

(2) 第二种边界条件：即已知插值区间两端的二阶导数

$$s''(x_0) = y''(x_0), \quad s''(x_n) = y''(x_n),$$

由于在区间端点处二阶导数 $M_0 = y_0''$ 与 $M_n = y_n''$ 已知，所以方程组(5.47)中实际上只包含有 $n-1$ 个未知数 $M_1, M_2, \cdots, M_{n-1}$，从而得方程组

$$\begin{bmatrix} 2 & \lambda_1 & & & & \\ \mu_2 & 2 & \lambda_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \mu_{n-2} & 2 & \lambda_{n-2} & \\ & & & \mu_{n-1} & 2 & \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \begin{bmatrix} d_1 - \mu_1 y_0'' \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} - \lambda_{n-1} y_n'' \end{bmatrix} \tag{5.48}$$

特别当 $M_0 = M_n = 0$ 时为自然边界条件，方程组(5.48)的右端成为 $[d_1, d_2, \cdots, d_{n-1}]^{\mathrm{T}}$，其形式特别简单．

(3) 第三种边界条件

由 $s''(x_0+0)=s''(x_n-0)$与 $s'(x_0+0)=s'(x_n-0)$

可以得到方程 $M_0=M_n$

和
$$\lambda_n M_1+\mu_n M_{n-1}+2M_n=d_n$$

其中
$$\lambda_n=\frac{h_1}{h_1+h_n},\ \mu_n=1-\lambda_n=\frac{h_n}{h_1+h_n}$$

$$d_n=\frac{6}{h_1+h_n}\left(\frac{y_1-y_0}{h_1}-\frac{y_n-y_{n-1}}{h_n}\right)$$

与式(5.44)合在一起,即得确定 $M_1,M_2,\cdots,M_n$ 的线性代数方程组:

$$\begin{bmatrix} 2 & \lambda_1 & & & & \mu_1 \\ \mu_2 & 2 & \lambda_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} & \\ \lambda_n & & & \mu_n & 2 & \end{bmatrix}\begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix}=\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \tag{5.49}$$

因此,只要根据给定数据和边界条件算出 λ_j,μ_j 和 d_j,就能分别求出方程(5.47)、(5.48)、(5.49),并由此解出 $s''(x)$在 x_j 上的值 M_j,将 M_j 代入式(5.39)就得到了 $s(x)$在各小区间上的表达式 $s_j(x)$从而求得在整个$[a,b]$区间上的三次样条插值函数 $s(x)$的表达式.

类似地,也可取 $s(x)$在 x_j 上的一阶导数 $s'(x)=m_j(j=0,1,2,\cdots,n)$作为待定参数,用埃尔米特插值构造出含有 m_j 的 $s(x)$的表达式,再利用 $s(x)$在内节点的二阶导数连续的条件,同样可确定 m_j,从而求得 $s(x)$.

综上所述,得到对于给定的函数表

x_i	x_0	x_1	$\cdots$	x_n
$f(x_i)$	y_0	y_1	$\cdots$	y_n

满足第一(或第二,第三)种边界条件的三次样条插值函数是存在且唯一的．其求解过程可归纳如下：

(1) 根据给定数据(x_i, y_i)及相应边界条件一(或条件二、三)建立方程组(5.47)、(5.48)或方程组(5.49).

即先由给定的数表列出二阶差商表,从而求得$f[x_{j-1}, x_j, x_{j+1}]$再算出h_j, λ_j, μ_j,就能建立三对角线方程组(5.47)、(5.48)、(5.49).

(2) 解方程组(5.47)、(5.48)或方程组(5.49),求出$M_0, M_1, \cdots, M_n$.

(3) 用求得的各$M_j(j=0,1,2,\cdots,n)$分别代入$s_j(x)$的表达式(5.39)求得$s(x)$在每一个小区间$[x_{j-1}, x_j]$上的分段表达式$s_j(x)$.

(4) 整个区间$[a,b]$上的三次样条插值函数即可表示为分段函数：

$$s(x)=\begin{cases} s_1(x) \\ s_2(x) \\ \vdots \\ s_n(x) \end{cases}$$

(5) 验算所求得的$s(x)$必须满足条件：在每一个小区间$[x_{j-1}, x_j]$上是一个三次多项式,在各节点上具有直到二阶的连续导数．

例 5.5 已知函数表

x_i	1	2	4	5
$y_i=f(x_i)$	1	3	4	2

在区间[1,5]上求满足上述函数表所给出的插值条件的三次自然样条插值函数.

解 因为这是在第二种边界条件下的插值问题,故确定 M_0, M_1, M_2, M_3 的方程组形如式(5.48),由已知自然边界条件,有 $M_0 = M_3 = 0$,则得求解 M_1, M_2 的方程组为

$$\begin{bmatrix} 2 & \lambda_1 \\ \mu_2 & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{5.50}$$

为确定方程组中系数,先作差商表

i	x_i	$f(x_i)$	一阶差商	二阶差商
0	1	1	2	
				$-\frac{1}{2}$
1	2	3	$\frac{1}{2}$	
				$-\frac{5}{6}$
2	4	4	-2	
3	5	2		

由 $x_0 = 1, x_1 = 2, x_2 = 4, x_3 = 5$ 得

$$h_1 = x_1 - x_0 = 1,\ h_2 = x_2 - x_1 = 2,\ h_3 = x_3 - x_2 = 1,$$

所以 $\lambda_1 = \dfrac{h_2}{h_1 + h_2} = \dfrac{2}{1+2} = \dfrac{2}{3},\ \mu_2 = \dfrac{h_2}{h_2 + h_3} = \dfrac{2}{2+1} = \dfrac{2}{3},$

$$d_1 = 6f[x_0, x_1, x_2] = 6 \times \left(-\frac{1}{2}\right) = -3,$$

$$d_2 = 6f[x_1, x_2, x_3] = 6 \times \frac{-5}{6} = -5,$$

代入方程组(5.50)得

$$\begin{cases} 2M_1 + \dfrac{2}{3}M_2 = -3 \\ \dfrac{2}{3}M_1 + 2M_2 = -5 \end{cases}$$

解得 $M_1 = -\dfrac{3}{4},\ M_2 = -\dfrac{9}{4}$, 又 $M_3 = M_0 = 0$ 代入公式(5.39),得

$s(x)$在$[x_0, x_1]$上的表达式为

$$s_1(x) = \frac{(x_1 - x)^3}{6h_1} M_0 + \frac{(x - x_0)^3}{6h_1} M_1 + \left(y_0 - \frac{M_0 h_2}{6}\right) \frac{x_1 - x}{h_1} + \left(y_1 - \frac{M_1 h_1^2}{6}\right) \frac{x - x_0}{h_1},$$

将 $x_0 = 1$，$x_1 = 2$，$y_0 = 1$，$y_1 = 3$，$h_1 = 1$，$M_0 = 0$，$M_1 = -\frac{3}{4}$代入上式，化简后得：

$$s_1(x) = -\frac{1}{8}x^3 + \frac{3}{8}x^2 + \frac{7}{4}x - 1,$$

同理可得 $s(x)$在$[x_1, x_2]$上的表达式为

$$s_2(x) = -\frac{1}{8}x^3 + \frac{3}{8}x^2 + \frac{7}{4}x - 1,$$

$s(x)$在$[x_2, x_3]$上的表达式为

$$s_3(x) = \frac{3}{8}x^3 - \frac{45}{8}x^2 + \frac{103}{4}x - 33$$

故所求的三次自然样条插值函数 $s(x)$在区间$[1,5]$上的表达式为

$$s(x) = \begin{cases} -\frac{1}{8}x^3 + \frac{3}{8}x^2 + \frac{7}{4}x - 1 & (1 \leqslant x \leqslant 2) \\ -\frac{1}{8}x^3 + \frac{3}{8}x^2 + \frac{7}{4}x - 1 & (2 \leqslant x \leqslant 4) \\ \frac{3}{8}x^3 - \frac{45}{8}x^2 + \frac{103}{4}x - 33 & (4 \leqslant x \leqslant 5) \end{cases}$$

用三次样条插值函数 $s(x)$逼近 $f(x)$是收敛的，并且也是数值稳定的，但其误差估计与收敛定理的证明都较复杂，下面我们只给出结论．

定理 5.6 设 $f(x)$是$[a, b]$上二次连续可微函数，在$[a, b]$上，以 $a = x_0 < x_1 < \cdots < x_n = b$ 为节点的三次样条插值函数 $s(x)$满足 $|f(x) - s(x)| \leqslant \frac{M_2}{2} \max_i |x_i - x_{i-1}|$，其中

$$M_2 = \max_{a \leqslant x \leqslant b} |f''(x)|.$$

证明(略)

定理 5.6 的结论告诉我们,样条插值有分段一次、二次插值同样的优点,即当增加插值点使区间减小时,误差会随插值点的增加而减小 .

应当指出,样条函数不一定必须是逐段三次多项式,也可以是逐段的简单函数,且连接点保持足够光滑 . 但因三次多项式计算简单,且能满足一般实际问题的要求,故三次样条函数用得最多 .

5.7 B-样条函数

我们首先引入磨光函数概念,由此可导出一类基本的样条函数即 B-样条函数 .

对任意给定的函数 $f(x) \in [a, b]$,我们总可以按某种方式将它的定义域延拓到整个实轴$(-\infty, +\infty)$上,并仍记为 $f(x)$. 我们将记 $\delta_h f(x)$为如下的对称差分:

$$\delta_h f(x) = f(x+h) - f(x-h)$$

定义 5.5 将函数 $f(x)$作一次不定积分后,接着进行一次以 h 为步长的对称差商运算,得到新函数:

$$f_1(x) = \frac{\delta_h}{2h}\int_0^x f(t)\mathrm{d}t = \frac{1}{2h}\int_{x-h}^{x+h} f(t)\mathrm{d}t \tag{5.51}$$

我们称 $f_1(x)$为 $f(x)$的**一次磨光函数**,而 $k-1$ 次磨光函数的一次磨光函数,则称为 $f(x)$的 k **次磨光函数** .

定义 5.6

$$x_+^m = \begin{cases} x^m, & x \geqslant 0 \\ 0, & x < 0 \end{cases} \tag{5.52}$$

称为 m **次截幂函数** . 对 $m=0$ 须特别定义 x_+^0 在原点等于$\frac{1}{2}$.

从磨光函数的定义可知,它可以使间断函数变成光滑函数 .

例 5.6 设 $f(x)=x_+^0$,求它的一次磨光函数 $f_1(x)$.

解

$$f_1(x)=\frac{1}{2h}\int_{x-h}^{x+h}t_+^0\,\mathrm{d}t$$

$$=\begin{cases}1, & x\geqslant h\\ (x+h)\dfrac{1}{2h}, & -h\leqslant x\leqslant h\\ 0, & x\leqslant -h\end{cases}$$

很明显,这时 $f_1(x)$是整个实轴上的连续函数,而原来的 $f(x)=x_+^0$ 在原点是间断的 . 另外,随着参数 h 的缩小,$f_1(x)$逼近 $f(x)$的精度随着提高 .

现在,我们转向讨论如何定义 B-样条函数 . 对 x_+^0 作一阶对称差分,所得函数 $\delta_{\frac{1}{2}}x_+^0$ 为一单位方波函数:

$$\delta_{\frac{1}{2}}x_+^0=\begin{cases}0, & x<-\dfrac{1}{2}\\ \dfrac{1}{2}, & x=-\dfrac{1}{2}\\ 1, & -\dfrac{1}{2}<x<\dfrac{1}{2}\\ \dfrac{1}{2}, & x=\dfrac{1}{2}\\ 0, & x<\dfrac{1}{2}\end{cases}$$

我们取 $h=\frac{1}{2}$,研究 $\delta_{\frac{1}{2}}x_+^0$ 的各次磨光函数 . 由于这一套函数重要,我们用专门的符号来表示它们 . 首先,记函数 $\delta_{\frac{1}{2}}x_+^0$ 为 $B_0(x)$,其 k 次磨光函数用 $B_k(x)$表示,称为 k 次 B-样条函数 . 下面我们来推导 $B_k(x)$的具体表达式 .

对任意非负整数 k,有结果:

$$B_k(x) = \delta_{\frac{1}{2}}^{k+1}\{x_+^k/k!\} \tag{5.53}$$

这就是说，$B_0(x)$的 k 次磨光函数乃是 $x_+^k/k!$ 的 $k+1$ 阶对称差分．当 $k=0$ 时式(5.33)是成立的，现证明 $k=1$ 时它也成立．事实上，根据磨光函数的定义的式(5.51)及 $h=\frac{1}{2}$，我们有

$$\begin{aligned}
B_1(x) &= \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} B_0(t)\mathrm{d}t \\
&= \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} \left\{\left(t+\frac{1}{2}\right)_+^0 - \left(1-\frac{1}{2}\right)_+^0\right\}\mathrm{d}t \\
&= \left\{\left(t+\frac{1}{2}\right)_+ - \left(t-\frac{1}{2}\right)_+\right\}\Big|_{x-\frac{1}{2}}^{x+\frac{1}{2}} \\
&= (x+1)_+ - x_+ - x_+ + (x-1)_+ \\
&= (x+1)_+ - 2x_+ + (x-1)_+ \\
&= \delta_{\frac{1}{2}}^2 x_+,
\end{aligned}$$

故 $k=1$ 时，式(5.53) 成立．

设对 $k-1$ 公式(5.52) 成立，我们来证对 k 也同样成立．事实上，由 $B_k(x)$ 的定义和归纳法的假定，有

$$B_{k-1}(x) = \delta_{\frac{1}{2}}^k\{x_+^{k-1}/(k-1)!\},$$

$$\begin{aligned}
B_k(x) &= \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} B_{k-1}(t)\mathrm{d}t \\
&= \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} \delta_{\frac{1}{2}}^k\{t_+^{k-1}/(k-1)!\}\mathrm{d}t \\
&= \delta_{\frac{1}{2}}^k \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} \{t_+^{k-1}/(k-1)!\}\mathrm{d}t
\end{aligned}$$

$$= \delta_{\frac{1}{2}}^{k}\left\{t_{+}^{k}/k!\Big|_{x-\frac{1}{2}}^{x+\frac{1}{2}}\right\}$$

$$= \delta_{\frac{1}{2}}^{k}\left\{\delta_{\frac{1}{2}}(x_{+}^{k}/k!)\right\}$$

$$= \delta_{\frac{1}{2}}^{k+1}\left\{x_{+}^{k}/k!\right\},$$

因此式(5.53)得证.

由于高阶差分可由函数表示,故我们进一步有

$$B_k(x) = \sum_{j=0}^{k+1}(-1)^j\binom{k+1}{j}\left(x+\frac{k+1}{2}-1\right)_{+}^{k}/k! \qquad (5.54)$$

从式(5.54)可看出,$B_k(x)$是一个分段 k 次多项式,k 次导数的间断点为

$$x_j = j - \frac{k+1}{2},\ j=0,1,\cdots,k+1.$$

也就是说,它是对应于$(-\infty,+\infty)$的一个分划

$$\Delta: -\infty < x_0 < \cdots < x_k < x_{k+1} < +\infty$$

的 k 次样条函数,x_j 为它的样条结点. 在研究样条函数中,不管是理论分析还是实际数值计算,函数 $B_k(x)$有它的独特作用,称为基本样条函数,简称为 k 次 B 样条.

当 $x < -\frac{k+1}{2}$时,有

$$x+\frac{k+1}{2}-j<0,\ j=0,1,\cdots,k+1,$$

即有

$$\left(x+\frac{k+1}{2}-j\right)_{+}^{k}=0,$$

从而

$$B_k(x)=0.$$

而当 $x > \frac{k+1}{2}$时,有

$$x+\frac{k+1}{2}-j>0,\ j=0,1,\cdots,k+1,$$

于是

$$\left(x+\frac{k+1}{2}-j\right)_+^k=\left(x+\frac{k+1}{2}-j\right)^k,$$

从而有

$$B_k(x)=\delta^{k+1}\{x^k/k!\}=0.$$

因而，函数 $B_k(x)$在区间$\left[-\frac{k+1}{2},\ \frac{k+1}{2}\right]$之外恒为零．运用定义和式(5.54)容易写出最初几个 B 样条函数：

$$B_0(x)=\begin{cases}0, & \text{当}|x|>\frac{1}{2}\\ 1, & \text{当}-\frac{1}{2}<x<\frac{1}{2}\\ \frac{1}{2}, & \text{当}\ x=\pm\frac{1}{2}\end{cases}$$

$$B_1(x)=\begin{cases}0, & \text{当}|x|\geqslant 1\\ 1-|x|, & \text{当}|x|<1\end{cases}$$

$$B_2(x)=\begin{cases}0, & \text{当}|x|\geqslant\frac{3}{2}\\ -x^2+\frac{3}{4}, & \text{当}|x|<\frac{1}{2}\\ \frac{1}{2}x^2-\frac{3}{2}|x|+\frac{9}{8}, & \text{当}\frac{1}{2}\leqslant|x|\leqslant\frac{3}{2}\end{cases}$$

$$B_3(x)=\begin{cases}0, & \text{当}|x|\geqslant 2\\ \frac{1}{2}|x|^3-x^2+\frac{2}{3}, & \text{当}|x|\leqslant 1\\ -\frac{1}{6}|x|^3+x^2-2|x|, & \text{当}\ 1<|x|<2\end{cases}$$

下面分别给出它们的图形：

由图 5-4 中可看到，从单位方波函数出发进行磨光，其图形越来越光滑，随着样条次数的增加，其结点越来越多．

引入 B-样条函数的好处是，对各次样条函数进行了统一的处理，并且在用样条求解许多问题时，如插值、最小二乘数据拟合、数

值微分、积分，求解常微分方程等问题，显得方便多了．

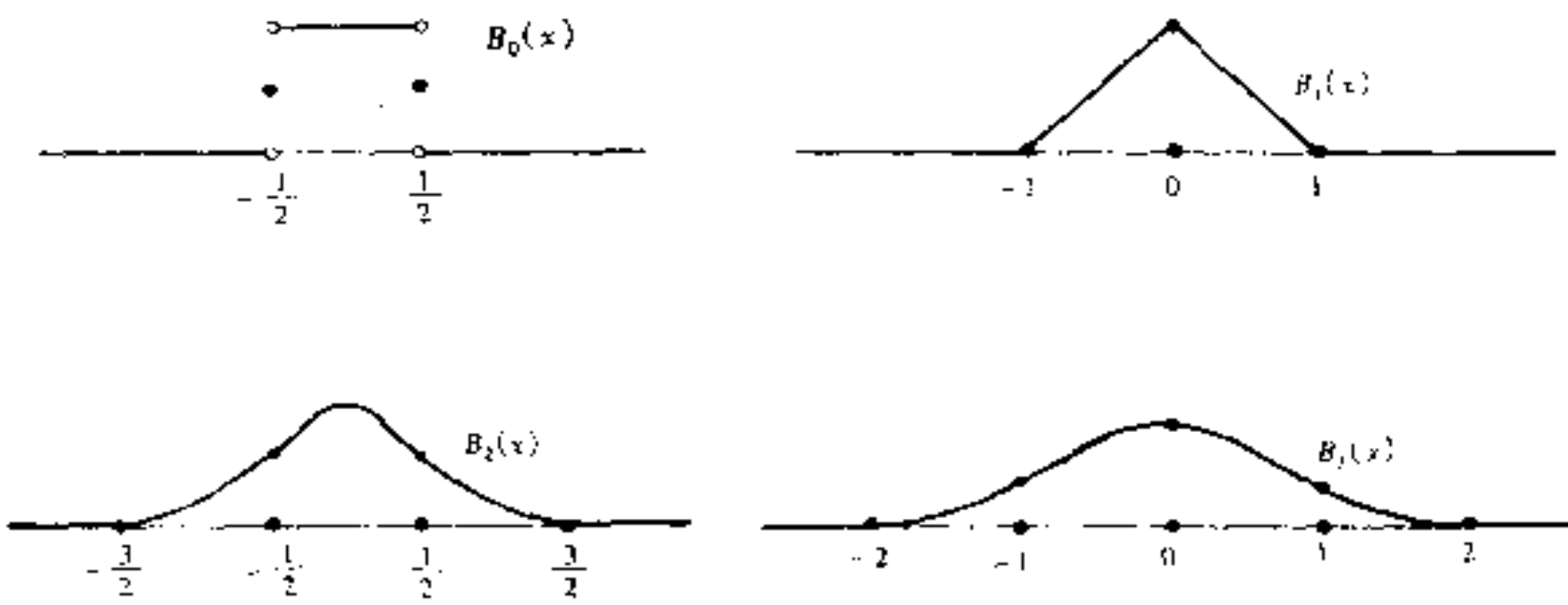


图 5-4

习 题 5

1. 已知 $f(0)=1$，$f(1)=2$，$f(2)=4$，求 $f(x)$的二次插值多项式．

2. 已知下列数据表：

x	10	11	12	13
$\ln x$	2.3026	2.3979	2.4849	2.5649

试分别用线性插值与二次插值计算 ln11.75 的近似值，并估计误差．

3. 设 $l_1(x)$，$l_1(x)$，$\cdots$，$l_n(x)$是以 x_0，x_1，$\cdots$，x_n 为节点的 n 次基本插值多项式．证明下述恒等式：

$$\sum_{i=0}^{n} x_i^k l_i(x) = x^k,\ (k=0,1,\cdots,n).$$

4. 假设对函数 $f(x)$ 在步长为 h 的等距点上造表，且有 $|f''(x)| \leqslant M$，证明：在表中任意相邻两点间做线性插值，误差不超过 $\frac{1}{8}Mh^2$. 设 $f(x)=\sin x$，问 h 应取多大才能保证线性插值的误差不大于 $\frac{1}{2}\times 10^{-6}$.

5. 已知 $f(x)=x^7+x^4+3x+1$，求 $f[2^0, 2^1, \cdots, 2^7]$ 及 $f[2^0, 2^1, \cdots, 2^7, 2^8]$ 的值 .

6. 根据下列函数表，作 4 次牛顿插值多项式，并求 $x=1.05$ 时 $\ln x$ 的近似值

x	1.00	1.02	1.06	1.08	1.09
$\ln x$	0.00000	0.01980	0.05827	0.07696	0.08618

7. 试用两种不同的方法求一个三次多项式 $p(x)$，使在节点 $x_0=0$, $x_1=1$ 上满足条件

$$p(0)=f(0)=0, \qquad p(1)=f(1)=1,$$

$$p'(0)=f'(0)=-3, p'(1)=f'(1)=9.$$

其中 $p'(x), f'(x)$ 是 $p(x)$ 和 $f(x)$ 的一阶导数，并估计余项 .

8. 记 $\pi(x)=\prod_{j=0}^{n}(x-x_j)$，证明 $l_i(x)=\dfrac{\pi(x)}{(x-x_i)\pi'(x_i)}$.

9. 已知函数 $y=f(x)$ 的函数表如下：

x	0.00	0.15	0.30	0.45	0.60
$f(x)$	1.00000	0.97800	0.91743	0.83160	0.73529

在区间[0,0.6]上求三次样条插值函数，使满足边界条件

$$s'(0)=0.00000, s'(0.60)=-0.64879.$$

10. 若 $f(x)$在$[a,b]$上有三阶连续导数,且已知 $f(x)$在$[a,b]$上两个互异节点 x_0, x_1 上的函数值 $f(x_0), f(x_1)$和一阶导数值 $f'(x_0)$. 试利用插值,导出 $f(x)$的下述表达式:

$$f(x)=-\frac{(x-x_1)(x-2x_0+x_1)}{(x_1-x_0)^2}f(x_0)$$

$$+\frac{(x-x_0)(x-x_1)}{x_0-x_1}f'(x_0)$$

$$+\frac{(x-x_0)^2}{(x_1-x_0)^2}f(x_1)+R(x)$$

其中 $R(x)=\frac{1}{6}(x-x_0)^2(x-x_1)f'''(\zeta)$, $\zeta\in(a,b)$.

第六章　函数逼近

6.1　引　言

用简单函数$p(x)$近似地代替函数$f(x)$,是“数值计算”中最基本的概念和方法之一.**近似代替又称为逼近,函数$f(x)$称为被逼近的函数,$p(x)$称为逼近函数,两者之差称为逼近的误差.**

函数逼近问题的具体提法是:对于函数类 A 中给定的函数$f(x)$,要求在另一类较简单的函数类$B(\subset A)$中找一个函数$p(x)$,使$p(x)$与$f(x)$之差在某种度量意义下达到最小.

一般,最常见的函数类 A 是区间$[a,b]$上的连续函数,最常用的函数类 B 有代数多项式、三角多项式、分段多项式以及有理分式函数类等容易计算的函数.

最常用的度量标准有两种:

1. 一致逼近

以函数$f(x)$和$p(x)$的最大误差

$$\max_{a\leqslant x\leqslant b}|f(x)-p(x)|$$

作为度量误差$f(x)-p(x)$的“大小”标准,在这种意义下的函数逼近称为**一致逼近或均匀逼近**.

2. 平方逼近

用积分

$$\int_a^b[f(x)-p(x)]^2\mathrm{d}x$$

作为度量误差$f(x)-p(x)$的“大小”标准,在这种意义下的函数逼近称为**平方逼近或均方逼近**.

本章主要介绍在这两种度量意义下用代数多项式$p(x)$去逼近$[a,b]$上的连续函数,也就是介绍函数的最佳一致逼近多项式

和最佳平方逼近多项式.

由于正交多项式是函数逼近的重要工具,因此,下面先介绍几种常见的正交多项式.

6.2 正交多项式

一、正交函数系的概念

1. 权函数的概念

定义 6.1 设$\rho(x)$是区间$[a,b]$上非负函数,如果满足条件:

(1) 积分$\int_b^a |x|^n \rho(x)\mathrm{d}x$存在,$(n = 0,1,2,\cdots)$;

(2) 对非负的连续函数$g(x)$,若

$$\int_a^b g(x)\,\rho(x)\mathrm{d}x = 0,$$

则在(a,b)上$g(x) \equiv 0$.

我们就称$\rho(x)$为$[a,b]$上的**权函数**.

2. 内积的概念

定义 6.2 设$f(x), g(x) \in c[a,b]$,$\rho(x)$是$[a,b]$上的权函数,则称

$$(f,g) = \int_a^b \rho(x)\,f(x)\,g(x)\mathrm{d}x$$

为$f(x)$与$g(x)$在$[a,b]$上以$\rho(x)$为**权函数的内积**.

3. 正交性的概念

定义 6.3 设$f(x), g(x) \in c[a,b]$,$\rho(x)$是$[a,b]$上的权函数,若

$$(f,g) = \int_a^b \rho(x)\,f(x)\,g(x)\mathrm{d}x = 0$$

则称$f(x)$与$g(x)$在$[a,b]$上**带权$\rho(x)$正交**.

定义 6.4 设在$[a,b]$上给定函数系$\{\varphi_0(x), \varphi_1(x), \cdots,$

$\varphi_n(x),\cdots\}$,若满足条件:

$$(\varphi_i(x),\varphi_k(x)) = \begin{cases} 0, & i \neq k \\ A_k > 0, & i = k(i,k = 0,1,2,\cdots) \end{cases}$$

则称函数系$\{\varphi_k(x)\}$是$[a,b]$上**带权$\rho(x)$的正交函数系**,特别地,当$A_k \equiv 1$时,则称该函数系为**标准正交函数系**.

若定义6.4中的函数系为多项式函数系$\{p_0(x),p_1(x),\cdots\}$,则称$\{p_k(x)\}$为以$\rho(x)$为权的在$[a,b]$上的正交多项式系.并称$p_n(x)$是$[a,b]$上带权$\rho(x)$的n次正交多项式.容易证明,在$[a,b]$上带权$\rho(x)$的正交函数系,一定是在$[a,b]$上线性无关的函数系.下面我们介绍几个常用的正交多项式.

二、常用的正交多项式

1. 切比雪夫(чебышев)多项式

切比雪夫多项式具有很多重要性质,是函数逼近的重要工具,并且有广泛的应用.

定义6.5 称多项式

$$T_n(x) = \cos(n\arccos x) \quad (-1 \leqslant x \leqslant 1, n = 0,1,2,\cdots) \tag{6.1}$$

为n次的**切比雪夫多项式(第一类)**.

切比雪夫多项式$T_n(x)$具有许多重要性质,下面简述几条:

(1) 正交性

切比雪夫多项式系$\{T_n(x)\}(n = 0,1,2,\cdots)$是在区间$[-1,1]$上带权$\rho(x) = \dfrac{1}{\sqrt{1-x^2}}$的正交多项式系.且

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} T_m(x)T_n(x)\mathrm{d}x = \begin{cases} 0, & m \neq n \\ \pi/2, & m = n \neq 0 \\ \pi, & m = n = 0 \end{cases} \tag{6.2}$$

证明 因为 $T_n(x) = \cos(n\arccos x)$,令$x = \cos\theta$,

则　$T_n(x) = \cos n\theta$，于是

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} T_m(x) T_n(x) \mathrm{d}x = \int_{\pi}^{0} \frac{1}{\sin\theta} \cos m\theta \cos n\theta (-\sin\theta) \mathrm{d}\theta$$

$$= \int_{0}^{\pi} \cos m\theta \cos n\theta \mathrm{d}\theta$$

$$= \begin{cases} 0, & m \neq n \\ \pi/2, & m = n \neq 0 \\ \pi, & m = n = 0. \end{cases}$$

(2) 递推关系

相邻的三个切比雪夫多项式具有三项递推关系式：

$$\begin{cases} T_0(x) = 1, \ T_1(x) = x, \\ T_{n+1}(x) = 2x \cdot T_n(x) - T_{n-1}(x), \ (n = 1, 2, \cdots) \end{cases} \tag{6.3}$$

证明　显然，$n = 0$ 时，$T_0(x) = 1$；$n = 1$ 时，$T_1(x) = x$；当 $n \geqslant 1$ 时，令 $x = \cos\theta$，则 $T_n(x) = \cos n\theta$.

由三角恒等式　$\cos(n+1)\theta + \cos(n-1)\theta = 2\cos n\theta \cos\theta$

即得　$T_{n+1}(x) + T_{n-1}(x) = 2x \cdot T_n(x)$

移项就得上述递推关系式(6.3).

(3) 奇偶性

切比雪夫多项式 $T_n(x)$ 当 n 为奇数时为奇函数；n 为偶数时为偶函数.

(4) $T_n(x)$ 在区间 $[-1,1]$ 上有 n 个不同的零点

$$x_k = \cos\frac{(2k-1)\pi}{2n}, \quad (k = 1, 2, \cdots, n).$$

显然，$T_n(x)$ 在 $[-1,1]$ 中的零点都是实的，互异的，且全部在 $[-1,1]$ 内.

(5) $T_n(x)$ 在 $[-1,1]$ 上有 $n+1$ 个不同的极值点

$$x_k' = \cos k\frac{\pi}{n}, (k = 0, 1, 2, \cdots, n).$$

使 $T_n(x)$ 轮流取得最大值 1 和最小值 (-1).

$\{x_k'\}$ $(k = 0,1,2,\cdots,n)$ 也称为 $T_n(x)$ 的交错点组.

(6) $T_n(x)$ 的最高次幂 x^n 的系数为 2^{n-1},$(n \geqslant 1)$.

2. 勒让德(Legendre) 多项式

定义 6.6 称多项式

$$p_n(x) = \frac{1}{2^n \cdot n!} \cdot \frac{\mathrm{d}^n}{\mathrm{d}x^n}[(x^2 - 1)^n] \quad (n = 0,1,2,\cdots) \tag{6.4}$$

为 n 次勒让德多项式.

勒让德多项式具有以下一些重要性质:

(1) 正交性

勒让德多项式序列 $\{P_n(x)\}$ 是在 $[-1,1]$ 上带权 $\rho(x) = 1$ 的正交多项式序列,即

$$\int_{-1}^{1} p_m(x)\, p_n(x)\mathrm{d}x = \begin{cases} 0, & m \neq n \\ \dfrac{2}{2n+1}, & m = n \end{cases} \tag{6.5}$$

(2) 递推关系

相邻的三个勒让德多项式具有三项递推关系式:

$$\begin{cases} p_0(x) = 1,\ p_1(x) = x, \\ p_{n+1}(x) = \dfrac{2n+1}{n+1} x\, p_n(x) - \dfrac{n}{n+1} p_{n-1}(x), (n = 1,2,\cdots) \end{cases} \tag{6.6}$$

由此依次可得

$$p_2(x) = \frac{1}{2}(3x^2 - 1),$$

$$p_3(x) = \frac{1}{2}(5x^3 - 3x), \cdots\cdots$$

(3) 奇偶性

当 n 为偶数时,$p_n(x)$ 为偶函数;当 n 为奇数时,$p_n(x)$ 为奇函数.

(4) $p_n(x)$ 的 n 个零点都是实的,互异的,且全部在区间 $[-1,1]$ 内部.

3. 其他常用的正交多项式

一般而言,如果给定的区间$[a,b]$不同且权函数不同,正交多项式也就不同,我们再介绍几个常用的正交多项式.

(1) 第二类切比雪夫多项式

定义 6.7 称多项式

$$u_n(x)=\frac{\sin[(n+1)\arccos x]}{\sqrt{1-x^2}}\quad(n=0,1,2,\cdots)\tag{6.7}$$

为第二类切比雪夫多项式.

可以证明

① $\{u_n(x)\}$ 是在区间$[-1,1]$上带权函数$\rho(x)=\sqrt{1-x^2}$的正交多项式序列.

即
$$\int_{-1}^{1}\sqrt{1-x^2}\,u_n(x)\cdot u_m(x)\mathrm{d}x=\begin{cases}0, & m\neq n\\ \pi/2, & m=n\end{cases}\tag{6.8}$$

② 相邻的三项具有递推关系式:

$$\begin{cases}u_0(x)=1,\ u_1(x)=2x,\\ u_{n+1}(x)=2xu_n(x)-u_{n-1}(x)\quad(n=1,2,\cdots)\end{cases}\tag{6.9}$$

(2) 拉盖尔(Laguerre)多项式

定义 6.8 称多项式

$$L_n(x)=\mathrm{e}^x\frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^n\mathrm{e}^{-x}),(0\leqslant x<+\infty)(n=0,1,2,\cdots)\tag{6.10}$$

为拉盖尔多项式.

可以证明

① $\{L_n(x)\}$ 是在区间$[0,+\infty)$上带权$\rho(x)=e^{-x}$的正交多项式序列.即

$$\int_0^{+\infty}\mathrm{e}^{-x}L_m(x)\cdot L_n(x)\mathrm{d}x=\begin{cases}0, & m\neq n\\ (n!)^2, & m=n\end{cases}.\tag{6.11}$$

② 相邻的三项具有递推关系式:

$$\begin{cases} L_0(x) = 1,\ L_1(x) = 1 - x, \\ L_{n+1}(x) = (1 + 2n - x)L_n(x) - n^2 L_{n-1}(x), (n = 1,2,\cdots) \end{cases} \tag{6.12}$$

(3) 埃尔米特(Hermite) 多项式

定义 6.9 称多项式

$$H_n(x) = (-1)^n \cdot e^{x^2} \cdot \frac{d^n}{dx^n}(e^{-x^2}),$$

$$(-\infty < x < +\infty)\ (n = 0,1,2,\cdots) \tag{6.13}$$

为埃尔米特多项式.

可以证明

① $\{H_n(x)\}$ 是在区间$(-\infty, +\infty)$上带权$\rho(x) = e^{-x^2}$的正交多项式序列.即

$$\int_{-\infty}^{+\infty} e^{-x^2} \cdot H_m(x) \cdot H_n(x)dx = \begin{cases} 0, & m \neq n \\ 2^n \cdot n! \cdot \sqrt{n}, & m = n \end{cases} \tag{6.14}$$

② 相邻的三项具有递推关系式:

$$\begin{cases} H_0(x) = 1,\ H_1(x) = 2x \\ H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x),\ (n = 1,2,\cdots) \end{cases} \tag{6.15}$$

6.3 最佳一致逼近

一、最佳一致逼近的概念

定义 6.10 设函数$f(x)$是区间$[a,b]$上的连续函数,对于任意给定的$\varepsilon > 0$,如果存在多项式$p(x)$,使不等式

$$\max_{a \leqslant x \leqslant b} |f(x) - p(x)| < \varepsilon$$

成立,则称多项式$p(x)$在区间$[a,b]$上一致逼近于函数$f(x)$.

那么,对于在区间$[a,b]$上的连续函数$f(x)$,是否存在多项

式$p(x)$一致逼近$f(x)$呢?这个问题许多人研究过.德国数学家维尔斯特拉斯(Weierstrass)在1885年曾给出下述著名定理.

维尔斯特拉斯定理 若函数$f(x)$是区间$[a,b]$上的连续函数,则对于任意$\varepsilon > 0$,总存在多项式$p(x)$,使对一切$a \leqslant x \leqslant b$,有$|f(x) - p(x)| < \varepsilon$成立.

但维尔斯特拉斯定理只在理论上肯定了闭区间上的连续函数$f(x)$可以用多项式$p(x)$以任意精确度来逼近,而并没有给出确定逼近得最快的多项式的方法.事实上,如果精确度要求较高,则用来逼近的多项式的次数一般也很高,这就增加了计算工作量,因此,用这个定理作为实际计算的依据是十分困难的.在实际计算时,人们总希望在一定的精确度要求下,逼近多项式的次数越低越好.

切比雪夫从这样的观点去研究一致逼近问题,他不让逼近多项式的次数 n 趋于无穷大,而是先把 n 加以固定,对于给定的$f(x) \in C[a,b]$,他提出在次数不超过 n 的多项式的集合p_n中去寻找一个多项式$p_n^*(x)$,使它在$[a,b]$上"最佳地逼近"$f(x)$,这里"最佳逼近"的意思是指$p_n^*(x)$对$f(x)$的偏差:

$\max\limits_{a\leqslant x\leqslant b}|f(x) - p_n^*(x)|$和其他任一$p(x) \in p_n$对$f(x)$的偏差

$\max\limits_{a\leqslant x\leqslant b}|f(x) - p(x)|$比较时是最小的,也就是

$$\max_{a\leqslant x\leqslant b}|f(x) - p_n^*(x)| = \min_{p(x)\in p_n}\left\{\max_{a\leqslant x\leqslant b}|f(x) - p(x)|\right\} \tag{6.16}$$

成立.这就是通常所谓的最佳一致逼近问题,也称为切比雪夫逼近问题.若这样的$p_n^*(x)$存在,则称多项式$p_n^*(x)$是函数$f(x)$在区间$[a,b]$上的 n 次最佳一致逼近多项式,简称为最佳逼近多项式.

现在进一步要讨论,最佳逼近多项式$p_n^*(x)$是否存在?是否唯一?如何构造?

我们不妨设 n 次多项式

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

显然 $\max\limits_{a \leqslant x \leqslant b} | f(x) - p(x) |$ 与 $p(x)$ 的系数 $a_0, a_1, \cdots, a_n$ 有关.

若记 $\varphi(a_0, a_1, \cdots, a_n) = \max\limits_{a \leqslant x \leqslant b} | f(x) - p(x) |$,

则 φ 应是 $n+1$ 个系数 $a_0, a_1, \cdots, a_n$ 的正值连续函数.我们称多元函数 $\varphi(a_0, a_1, \cdots, a_n)$ 的最小值

$$\min\limits_{a_k} \varphi(a_0, a_1, \cdots, a_n) = \min\limits_{a_k} \{ \max\limits_{a \leqslant x \leqslant b} | f(x) - p(x) | \}$$

$$(k = 0, 1, 2, \cdots, n) \quad (6.17)$$

为 $f(x)$ 与 $p(x)$ 在 $[a, b]$ 上的最小偏差.

由上述讨论可知,寻求函数 $f(x)$ 在 $[a, b]$ 上的 n 次最佳一致逼近多项式 $p_n^*(x)$ 的问题就归结为求多元函数 $\varphi(a_0, a_1, \cdots, a_n)$ 的最小值问题.

可以证明,存在唯一的 $(a_0^*, a_1^*, \cdots, a_n^*)$ 能使

$$\varphi(a_0^*, a_1^*, \cdots, a_n^*) = \min\limits_{a_k} \{ \max\limits_{a \leqslant x \leqslant b} | f(x) - p(x) | \}$$

成立,也即存在唯一的 $p_{(x)}^* = a_0^* + a_1^* x + \cdots + a_n^* x^n$ 满足关系式:

$$\max\limits_{a \leqslant x \leqslant b} | f(x) - p_n^*(x) | = \min\limits_{p(x) \in p_n} \{ \max\limits_{a \leqslant x \leqslant b} | f(x) - p(x) | \}$$

这就说明,对于 $[a, b]$ 上的任意连续函数 $f(x)$,其 n 次最佳逼近多项式 $p_n^*(x)$ 是存在且唯一的,因证明较繁,这里我们不予讨论,下面我们主要介绍 $p_n^*(x)$ 的构造.

二、最佳一致逼近多项式的求法

按照定义去求最佳一致逼近多项式是很困难的,因此需要进一步研究最佳一致逼近多项式的特征.

定义 6.11 设 $f(x) \in C[a, b]$, $p(x) \in p_n$, 若存在 $x_0 \in [a, b]$ 使

$$| f(x_0) - p(x_0) | = \max\limits_{a \leqslant x \leqslant b} | f(x) - p(x) | = \mu,$$

则称点 x_0 是$p(x)$ 关于$f(x)$ 的偏差点.

并在　$p(x_0)-f(x_0)=\mu$ 时,称 x_0 为正偏差点,

$p(x_0)-f(x_0)=-\mu$ 时,称 x_0 为负偏差点.

由于连续函数在$[a,b]$ 上有最大值,故$p(x)$ 关于$f(x)$ 的偏差点总是存在的.更进一步还有以下重要定理.

定理6.1 (切比雪夫定理)　$p^*(x)\in p_n$ 是区间$[a,b]$上连续函数$f(x)$ 的 n 次最佳一致逼近多项式的充分必要条件是$p^*(x)$ 在区间$[a,b]$上至少有 $n+2$ 个点依次轮流为正、负的偏差点,即有 $n+2$ 个点 $a\leqslant x_1<x_2<\cdots<x_{n+2}\leqslant b$,使

$$f(x_k)-p^*(x_k)=(-1)^k\cdot\sigma\cdot\mu \tag{6.18}$$

成立.其中 $\mu=\max\limits_{a\leqslant x\leqslant b}|f(x)-p(x)|,\sigma=\pm 1,k=1,2,\cdots,n+2$.(证明从略)

点集$\{x_1,x_2,\cdots,x_{n+2}\}$ 称为切比雪夫交错点组,其中每一个 $x_k(k=1,2,\cdots,n+2)$ 称为交错点.

定理6.1表明　若用最佳一致逼近多项式 $p^*(x)$ 来近似代替$f(x)$ 时,其误差$R(x)=f(x)-p^*(x)$ 在$[a,b]$上的分布是十分均匀的(见图6-1).

定理6.1就是函数逼近理论中非常著名的切比雪夫定理,由此定理可推得下面两个推论.

推论1　设$f(x)\in c[a,b]$,则$f(x)$ 在 p_n 中的最佳一致逼近多项式,就是$f(x)$ 在$[a,b]$上的某个 n 次拉格朗日插值多项式.

推论1表明:如果能适当选择插值节点,求出使偏差 $\max\limits_{a\leqslant x\leqslant b}|f(x)-p(x)|$ 为最小的 n 次拉格朗日插值多项式,那么就能求得$f(x)$ 的 n 次最佳一致逼近多项式$p^*(x)$.

推论2　如果函数$f(x)$ 在区间$[a,b]$上有 $n+1$ 阶导数,且 $f^{(n+1)}(x)$ 在$[a,b]$上恒为正(或负),那么区间$[a,b]$的端点 a 和 b 都属于$f(x)-p^*(x)$ 的交错点组.

切比雪夫定理不仅给出了最佳一致逼近多项式的特性,并从

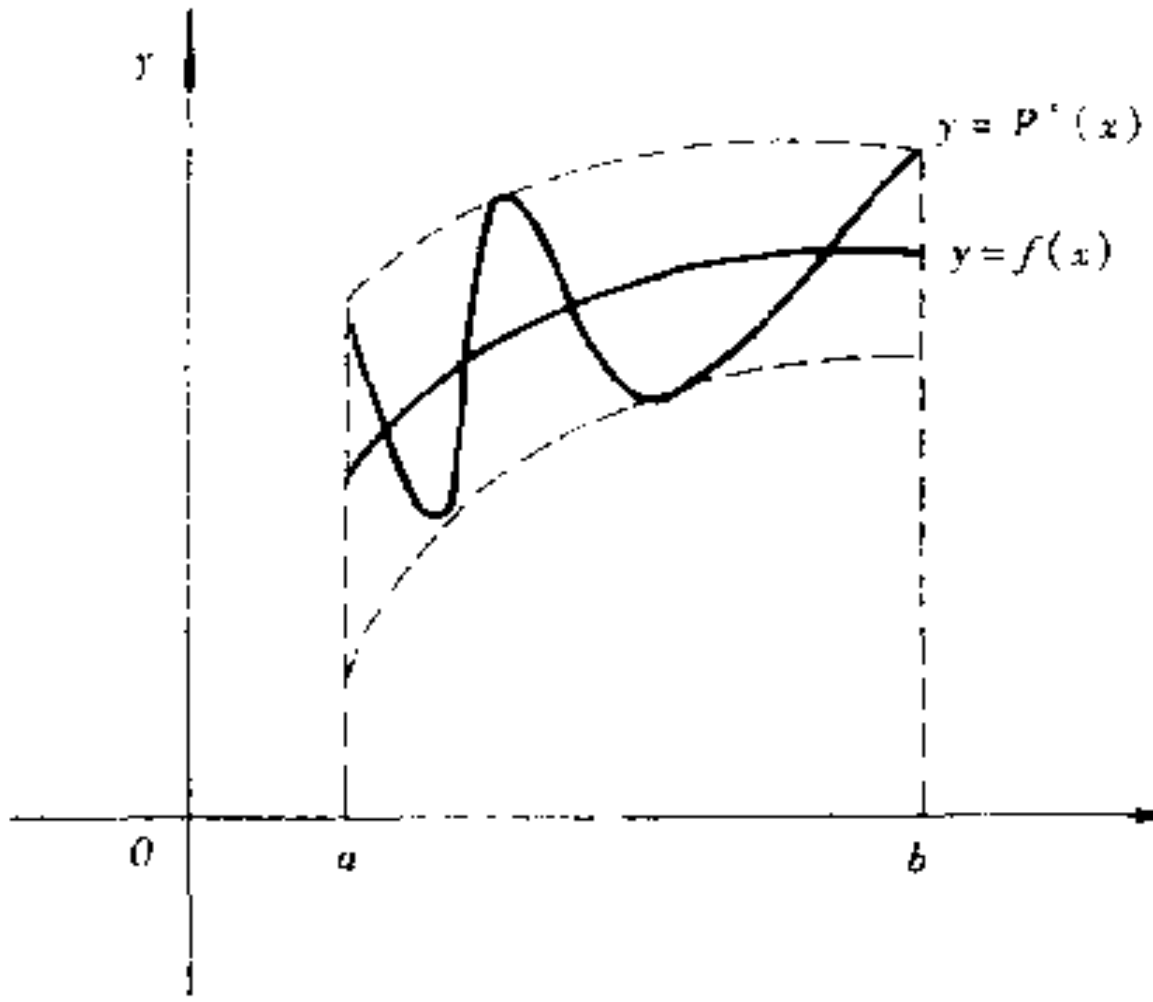


图 6-1

理论上给出了寻找最佳一致逼近多项式的方法.但事实上,要为某一连续函数求得最佳一致逼近多项式往往是十分困难的.下面我们仅介绍比较简单的线性最佳一致逼近多项式的求法以及切比雪夫多项式在函数逼近中的应用.

1. 线性最佳一致逼近多项式 $p_1^*(x)$ 的求法

下面讨论 $n = 1$ 时,求一次最佳一致逼近多项式 $p_1^*(x) = a_0 + a_1 x$ 的方法.

设函数 $f(x)$ 在 $[a,b]$ 上有二阶导数,且 $f''(x)$ 在 $[a,b]$ 上不变号,则根据切比雪夫定理,存在点 $a \leqslant x_1 < x_2 < x_3 \leqslant b$,使

$$f(x_k) - p_1^*(x_k) = (-1)^k \cdot \sigma \cdot \max_{a \leqslant x \leqslant b} | f(x) - p_1^*(x) | \tag{6.19}$$

其中,$\sigma = \pm 1, k = 1,2,3$.

由于 $f''(x)$ 在 $[a,b]$ 上不变号,根据推论 2 可知,区间 $[a,b]$ 的两个端点 a,b 都属于 $f(x) - p_1^*(x)$ 的交错点组,即有 $x_1 = a$, $x_3 = b$,而另一交错点 x_2 必位于 $[a,b]$ 的内部,且它是

$f(x)-p_1^*(x)$ 的极值点，所以 $f'(x_2)-p_1^{*'}(x_2)=0$，而 $p_1^{*'}(x_2)=a_1$，从而 $f'(x_2)=p_1^{*'}(x_2)=a_1$，又因为 $f''(x)$ 在 $[a,b]$ 上不变号，所以 $f'(x)$ 在 $[a,b]$ 上严格单调，从而 $f(x)-p_1^*(x)$ 在 (a,b) 内有，且只有一个极值点 x_2，由此可得

$$f(a)-p_1^*(a)=-[f(x_2)-p_1^*(x_2)]=f(b)-p_1^*(b)$$

或

$$\begin{cases} f(a)-(a_0+a_1a)=f(b)-(a_0+a_1b) \\ f(a)-(a_0+a_1a)=-[f(x_2)-(a_0+a_1x_2)] \end{cases}$$

解此方程组得

$$\begin{cases} a_1=\dfrac{f(b)-f(a)}{b-a} \\ a_0=\dfrac{1}{2}[f(a)+f(x_2)]-\dfrac{a+x_2}{2}\dfrac{f(b)-f(a)}{b-a} \end{cases} \tag{6.20}$$

(6.21)

其中，x_2 由 $f'(x_2)=0$ 得到，由此则得 $f(x)$ 在 $[a,b]$ 上线性最佳一致逼近多项式

$$p_1^*(x)=\frac{f(a)+f(x_2)}{2}+\frac{f(b)-f(a)}{b-a}\left(x-\frac{a+x_2}{2}\right) \tag{6.22}$$

线性最佳一致逼近的几何意义是：直线 $y=p_1^*(x)$ 与弦 AB 平行且过线段 AC 的中点 D，其方程为

$$y=\frac{1}{2}[f(a)+f(x_2)]+a_1\left(x-\frac{a+x_2}{2}\right)$$

见图 6-2，其中 $\begin{cases} d_1=d_3 \\ d_1=d_2 \end{cases}$.

例 6.1 求函数 $f(x)=\sqrt{x}$ 在 $\left[\frac{1}{4},1\right]$ 上的线性最佳一致逼近多项式 $p_1^*(x)$.

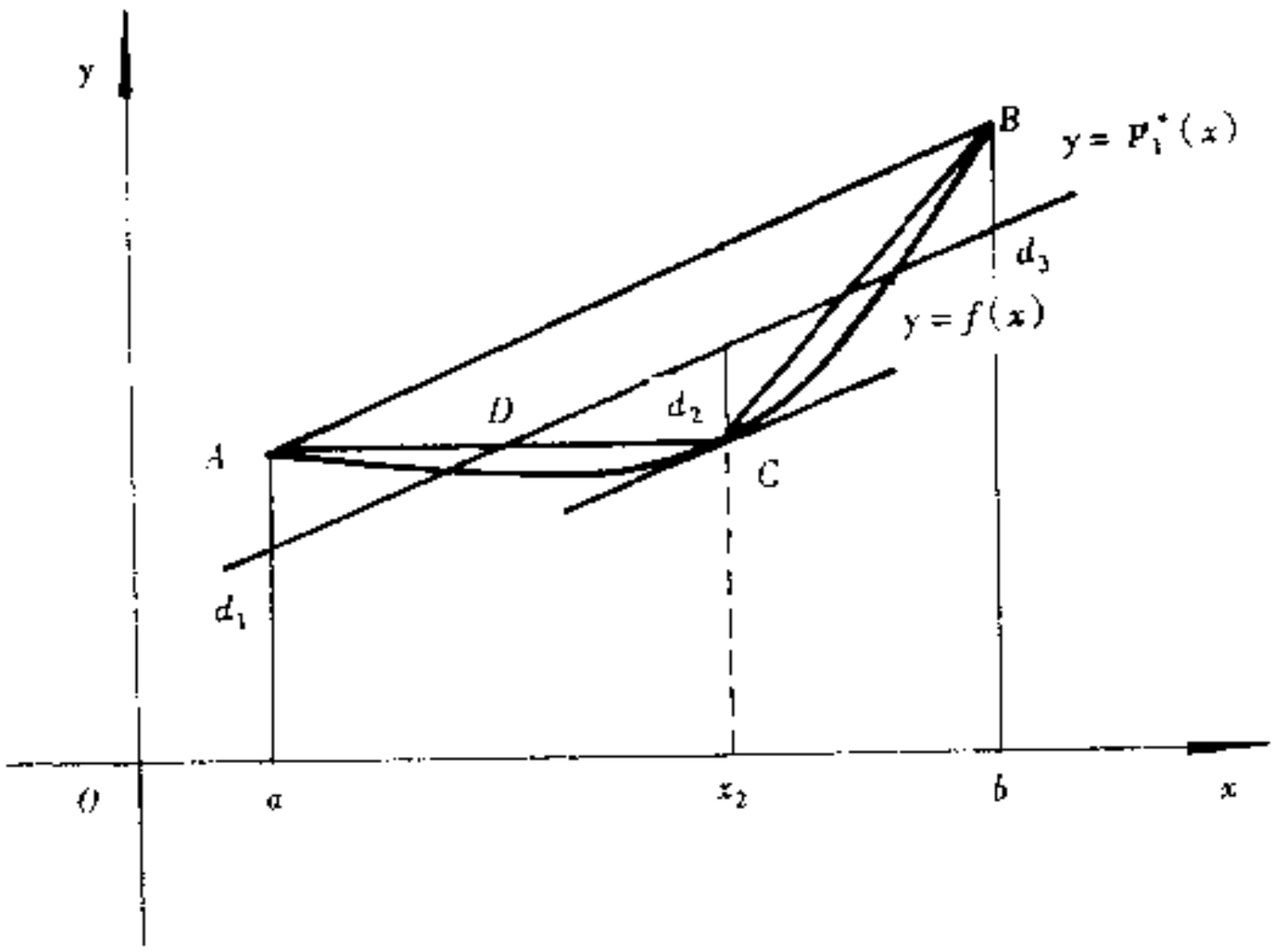


图 6-2

解　显然 $f''(x)$ 在 $\left[\frac{1}{4},1\right]$ 上不变号，$(f''(x) < 0)$，故 $x_1 = \frac{1}{4}, x_3 = 1$，由式(6.20)得

$$a_1 = \frac{\sqrt{1} - \sqrt{1/4}}{1 - 1/4} = \frac{2}{3}$$

由　$f'(x) = \frac{1}{2\sqrt{x}}$ 及 $f'(x_2) = a_1$，得

$$\frac{1}{2\sqrt{x_2}} = \frac{2}{3}, \text{所以 } x_2 = \frac{9}{16},$$

再由式(6.21)得

$$a_0 = \frac{1}{2}\left[\frac{1}{2} + \frac{3}{4} - \frac{2}{3}\left(\frac{1}{4} + \frac{9}{16}\right)\right] = \frac{17}{48},$$

所以，所求的线性最佳一致逼近多项式为

$$p_1^*(x) = \frac{2}{3}x + \frac{17}{48}.$$

2. 切比雪夫多项式在函数逼近中的应用

由切比雪夫多项式的三项递推关系式容易验证 $T_n(x)$ 的最高次幂 x^n 的系数为 $2^{n-1}(n = 1,2,\cdots)$. 则 $\tilde{T}_n(x) = \frac{1}{2^{n-1}}T_n(x)$ 是一个首项系数为 1 的 n 次多项式，又由 $T_n(x)$ 的性质(5)知 $\tilde{T}_n(x)$ 在点 $x_k' = \cos k\frac{\pi}{n}(k = 0,1,2,\cdots,n)$ 上依次达到它在 $[-1,1]$ 上的极值 $\pm 1/2^{n-1}$，即 $\tilde{T}_n(x_k') = \frac{(-1)^k}{2^{n-1}}$.

若用 $H_n(x)$ 表示首项系数为 1 的 n 次多项式的全体，则有

定理 6.2 在区间 $[-1,1]$ 上，在首项系数为 1 的一切 n 次多项式 $H_n(x)$ 中，$\tilde{T}_n(x) = \frac{1}{2^{n-1}}T_n(x)$ 与零的偏差最小，且其偏差为 $\frac{1}{2^{n-1}}$，即对于任何 $p(x) \in H_n(x)$，有

$$\frac{1}{2^{n-1}} = \max_{-1\leqslant x\leqslant 1} |\tilde{T}_n(x) - 0| \leqslant \max_{-1\leqslant x\leqslant 1} |p(x) - 0|$$

证明 (用反证法)

假若存在首项系数为 1 的另一个 n 次多项式 $p(x) \neq \tilde{T}_n(x)$，它对零的偏差比 $\tilde{T}_n(x)$ 与零的偏差还小，即有

$$\max_{-1\leqslant x\leqslant 1} |p(x) - 0| < \max_{-1\leqslant x\leqslant 1} |\tilde{T}_n(x) - 0| = \frac{1}{2^{n-1}}$$

令 $$q(x) = \tilde{T}_n(x) - p(x),$$

由于 $\tilde{T}_n(x), p(x)$ 均属于 $H_n(x)$，则 $q(x)$ 是一个次数不超过 $n-1$ 次的多项式.

在 $T_n(x)$ 的交错点组 $x_k' = \cos\frac{k\pi}{n}$ $(k = 0,1,2,\cdots,n)$ 处，由于 $|p(x_k')| \leqslant \max_{-1\leqslant x\leqslant 1}|p(x)| < \frac{1}{2^{n-1}}$ 即 $-\frac{1}{2^{n-1}} < p(x_k') < \frac{1}{2^{n-1}}$，$\tilde{T}_n(x)$ 在 $x_k'(k = 0,1,2,\cdots,n)$ 处轮流取到 $(-1)^k\frac{1}{2^{n-1}}$，因而

$q(x_k') = \frac{(-1)^k}{2^{n-1}} - p(x_k')$，显然，$q(x)$ 在 $n+1$ 个交错点处轮流取正负值，由连续函数的性质知，$q(x)$ 应当具有 n 个零点，但已知 $q(x)$ 至多是 $n-1$ 次多项式，因而 $q(x) \equiv 0$，所以 $p(x) \equiv \tilde{T}_n(x)$，这个结果与假设矛盾，从而定理得证.

从这个定理知，所有首项系数为 1 的 n 次多项式在区间 $[-1,1]$ 上的最大值满足 $\max\limits_{-1\leqslant x\leqslant 1} |p(x)| \geqslant \frac{1}{2^{n-1}}$.

定理 6.2 称为切比雪夫多项式的极性，这种极性也是切比雪夫多项式的重要性质，正因为如此，$\tilde{T}_n(x)$ 成为在区间 $[-1,1]$ 上逼近其他函数的一种重要工具，例如这一性质在求函数的近似最佳逼近多项式中有广泛的应用.

作为应用，下面介绍一下多项式插值余项的极小化.

设在 $[-1,1]$ 上给定 $n+1$ 个互异的节点 $x_0, x_1, \cdots, x_n$，函数 $f(x)$ 在 $[-1,1]$ 上具有 $n+1$ 阶连续导数 $f^{(n+1)}(x)$，对 $f(x)$ 作多项式插值时，由拉格朗日插值的余项表示式

$$R_n(x) = f(x) - L_n(x)$$

$$= \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x - x_j)，\text{其中 } \xi \in (-1,1).$$

若
$$M_{n+1} = \max_{-1\leqslant x\leqslant 1} |f^{(n+1)}(x)|,$$

则有
$$|R_n(x)| \leqslant \frac{M_{n+1}}{(n+1)!}\left|\prod_{j=0}^{n}(x - x_j)\right|$$

显然，余项的大小，取决于因子 $\prod\limits_{j=0}^{n}|x - x_j|$ 的大小.

现在我们提出一个问题：怎样选取节点 $x_j (j = 0,1,2,\cdots,n)$ 能使 $\max\limits_{-1\leqslant x\leqslant 1}|(x - x_0)(x - x_1)\cdots(x - x_n)|$ 尽可能小？

由于 $\prod\limits_{j=0}^{n}(x - x_j)$ 是一个最高次项系数为 1 的 $n+1$ 次多项式，由 $T_n(x)$ 的极性讨论知，当 x_j 满足

$$(x - x_0)(x - x_1)\cdots(x - x_n) = \frac{1}{2^n}T_{n+1}(x)$$

时，$\max\limits_{-1\leqslant x\leqslant 1}|(x-x_0)(x-x_1)\cdots(x-x_n)|=\dfrac{1}{2^n}$ 取得极小，亦即只要插值节点 x_k 取成 $n+1$ 次切比雪夫多项式的零点

$$x_k=\cos(2k+1)\frac{\pi}{2(n+1)}(k=0,1,2,\cdots,n),$$

则插值公式的余项在全区间$[-1,1]$上的最大绝对值为极小，此时，有

$$\max|R_n(x)|\leqslant\frac{M_{n+1}}{(n+1)!}\max_{-1\leqslant x\leqslant 1}|(x-x_0)(x-x_1)\cdots(x-x_n)|$$

$$=\frac{M_{n+1}}{(n+1)!}\max_{-1\leqslant x\leqslant 1}\frac{|T_{n+1}(x)|}{2^n}=\frac{M_{n+1}}{(n+1)!\cdot 2^n}.$$

这表明用切比雪夫多项式$T_{n+1}(x)$的零点

$$x_k=\cos\frac{2k+1}{2(n+1)}\pi(k=0,1,2,\cdots,n)$$

作插值点的插值多项式$L_n(x)$的余项具有极小化性质，因此$L_n(x)$可作为$f(x)$的近似的最佳一致逼近多项式.

如果插值区间是$[a,b]$，而不是$[-1,1]$，则可通过变量置换

$$x=\frac{a+b}{2}+\frac{b-a}{2}t$$

把函数变换成

$$f(x)=f\left(\frac{a+b}{2}+\frac{b-a}{2}t\right)=g(t)$$

其中$-1\leqslant t\leqslant 1$，即将定义在区间$[a,b]$上的函数$f(x)$化为定义在$[-1,1]$上的函数 $g(t)$. 因此，应选取

$$t_k=\cos\frac{2k+1}{2(n+1)}\pi(k=0,1,2,\cdots,n),$$

即取插值节点为

$$x_k=\frac{a+b}{2}+\frac{b-a}{2}\cdot\cos\frac{2k+1}{2(n+1)}\pi(k=0,1,2,\cdots,n)$$

能使$\max\limits_{a\leqslant x\leqslant b}|R_n(x)|$达到极小，并且有

$$\max_{a\leqslant x\leqslant b}|R_n(x)|\leqslant\frac{M_{n+1}}{(n+1)!}\frac{(b-a)^{n+1}}{2^{n+1}}\max_{-1\leqslant t\leqslant 1}\left|\frac{1}{2^n}T_{n+1}(t)\right|$$

$$=\frac{M_{n+1}}{(n+1)!}\cdot\frac{(b-a)^{n+1}}{2^{2n+1}}\tag{6.23}$$

例6.2 用多项式插值余项的极小化方法求函数$f(x)=e^{-x}$在$[0,1]$上的近似最佳一致逼近多项式，要求逼近误差不超过0.5×10^{-3}.

解 由式(6.23)得$\max\limits_{0\leqslant x\leqslant 1}|R_n(x)|\leqslant\dfrac{M_{n+1}}{(n+1)!2^{2n+1}}$

而 $M_{n+1}=\max\limits_{0\leqslant x\leqslant 1}|f^{(n+1)}(x)|=\max\limits_{0\leqslant x\leqslant 1}|(-1)^{n+1}e^{-x}|=1$

当$n=3$时，有

$$\max_{0\leqslant x\leqslant 1}|R_3(x)|\leqslant\frac{1}{2^7\cdot 4!}\approx 0.0003255<0.5\times10^{-3}$$

因此，取插值节点为$x_k=\dfrac{1}{2}\left(1+\cos\dfrac{2k+1}{8}\pi\right)$ $(k=0,1,2,3)$，得到的三次拉格朗日插值多项式为

$$L_3(x)\approx 0.99977-0.99290x+0.46323x^2-0.10240x^3,$$

这可作为$f(x)=e^{-x}$在$[0,1]$上的近似最佳一致逼近多项式.

由定理6.2，我们还可以利用切比雪夫多项式来降低逼近多项式的次数.例如，我们常要在一定区间上求一个函数在一定误差范围内的逼近多项式，自然，我们希望此逼近多项式的次数越低越好.

由于$\{x^k\}$也可用切比雪夫多项式$\{T_n(x)\}$表示，例如

$$1=T_0,\ x=T_1(x),\ x^2=\frac{1}{2}(T_0+T_2),\cdots\tag{6.24}$$

若把普通n次多项式$P_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n$中的所有$x_k(k=0,1,2,\cdots,n)$用上述式(6.24)中的切比雪夫多项式去代替，则$p_n(x)$可改写为

$$p_n(x)=b_0+b_1T_1(x)+b_2T_2(x)+\cdots+b_nT_n(x),$$

在满足误差要求的情况下，可利用切比雪夫多项式的极值性质，把 $p_n(x)$ 的高次幂的项缩减下来，使它成为 $m(m \leqslant n-1)$ 次多项式，这个 m 次多项式可以作为在已给精度下的 $f(x)$ 在 $[-1,1]$ 上的近似的最佳一致逼近多项式.

下面通过例子来说明这种方法.

例 6.3 设在区间 $[-1,1]$ 上，要计算 $f(x)=e^x$，欲找一个低次多项式 $p(x)$ 近似代替 e^x，使误差 $\epsilon < 0.01$.

解 先对 $f(x)=e^x$ 在 $x=0$ 处作泰勒展开，有

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^5}{4!} + \cdots$$

若取前六项之和 $P_5(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!}$ 作为 e^x 的近似，这时其截断误差 $R_5(x)$ 满足

$$|R_5(x)| = \left|\frac{1}{6!}e^{\xi}\cdot x^6\right| < \frac{1}{6!}e = \frac{1}{720}e < 0.0038 < 0.01$$

即误差已满足所提要求，但注意 $p_5(x)$ 是一个 5 次多项式. 若利用 $\{x^k\}$ 和 $|T_k(x)|$ 的关系式，$p_5(x)$ 也可表示为

$$p_5(x) = \frac{81}{64}T_0 + \frac{217}{192}T_1 + \frac{13}{48}T_2 + \frac{17}{384}T_3 + \frac{1}{192}T_4 + \frac{1}{1920}T_5$$

这里，我们看到 T_k 的足标 k 越大，T_k 的系数就越小，由于在 $|x| \leqslant 1$ 上，$T_k(x) \leqslant 1(k=0,1,2,\cdots)$，故可以略去次数高的 $T_k(x)$ 项，而这就意味着逼近多项式的次数降低，这正是我们希望达到的. 此题中，若略去其最后的两项，则所增添的误差为

$$\frac{1}{192} + \frac{1}{1920} < 0.0058,$$

从而在 $|x| \leqslant 1$ 上，若用

$$p(x) = \frac{81}{64}T_0 + \frac{217}{192}T_1 + \frac{13}{48}T_2 + \frac{17}{384}T_3$$

近似代替 e^x 的总误差不超过 $0.0038 + 0.0058 = 0.0096 < 0.01$，亦即用 $p(x)$ 代替 e^x 也满足所给误差要求.

再用$\{T_k(x)\}$和$\{x^k\}$之间的互换关系式(6.24),把$p(x)$重写为通常的x的多项式,即

$$p(x) = \frac{1}{384}(382 + 383x + 208x^2 + 68x^3),$$

显然,这个$p(x)$是一个3次多项式,比原来的$p_5(x)$降低了2次,从而用$p(x)$求值的计算工作量比用$p_5(x)$少.这个方法的处理过程就称为**缩减幂级数法**.

上述方法既使逼近多项式的次数降低,而且还可以使误差分布更为均匀,因此$p(x)$也可以作为$f(x)$的近似最佳一致逼近多项式.

对于任一有限区间$[a,b]$上的逼近问题,仍可以通过变量替换$x = \frac{a+b}{2} + \frac{b-a}{2}t$把$[a,b]$区间转化为$[-1,1]$上的逼近问题类似讨论.

6.4 最佳平方逼近

一般而言,在$[a,b]$上对给定的函数求它的一致逼近函数比较困难,下面我们介绍在$[a,b]$上较容易计算的另一种逼近方法——**最佳平方逼近**.

一、函数的最佳平方逼近

定义6.12 对于给定的函数$f(x) \in C[a,b]$,若n次多项式$s^*(x) = \sum_{j=0}^{n} a_j^* x^j$满足关系式

$$\int_a^b [f(x) - s^*(x)]^2 dx = \min_{s(x) \in P_n} \int_a^b [f(x) - s(x)]^2 dx \tag{6.25}$$

则称$s^*(x)$为$f(x)$在区间$[a,b]$上的n次最佳平方逼近多项式.

实用上,为了使问题的讨论更具有一般性,我们可以把定义进

行更一般的推广，若把 x^j 推广为一般的连续函数$\varphi_j(x)$，且要求 $\varphi_0, \varphi_1, \cdots, \varphi_n$ 线性无关，则 $s_j(x) = \sum_{j=0}^{n} a_j \varphi_j(x)$，并且积分可带权函数，则定义 6.12 可改为如下定义.

定义 6.13 对于给定的函数 $f(x) \in c[a, b]$，如果存在 $s^*(x) \in \Phi = \text{span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$ 使

$$\int_a^b \rho(x)[f(x) - s^*(x)]^2 \mathrm{d}x = \min_{s(x) \in \Phi} \int_a^b \rho(x)[f(x) - s(x)]^2 \mathrm{d}x \tag{6.26}$$

则称 $s^*(x)$ 是$f(x)$ 在集合 Φ 中的最佳平方逼近函数.

显然，定义 6.12 是定义 6.13 中取$\rho(x) = 1, \varphi_j(x) = x^j$ 的特殊情况，作这样的推广，使我们可以考虑更一般的，像

$$\int_{-1}^{1} \frac{1}{\sqrt{1 - x^2}}[f(x) - s(x)]^2 \mathrm{d}x, \quad \int_0^{+\infty} \mathrm{e}^{-x}[f(x) - s(x)]^2 \mathrm{d}x$$

之类在实际问题中有用的带权函数的平方逼近问题.

显然，求最佳平方逼近函数 $s^*(x) = \sum_{j=0}^{n} a_j^* \cdot \varphi_j(x)$的问题可归结为求它的系数 $a_0^*, a_1^*, \cdots, a_n^*$，使多元函数

$$I(a_0, a_1, \cdots, a_n) = \int_a^b \rho(x)\left[f(x) - \sum_{j=0}^{n} a_j \varphi_j(x)\right]^2 \mathrm{d}x$$

取得极小值，也即点$(a_0^*, a_1^*, \cdots, a_n^*)$ 是 $I(a_0, a_1, \cdots, a_n)$ 的极值点.由于 $I(a_0, a_1, \cdots, a_n)$ 是关于 $a_0, a_1, \cdots, a_n$ 的二次函数，利用多元函数取得极值的必要条件：$\dfrac{\partial I}{\partial a_k} = 0 (k = 0, 1, 2, \cdots, n)$，即

$$\frac{\partial I}{\partial a_k} = 2\int_a^b \rho(x)\left[f(x) - \sum_{j=0}^{n} a_j \varphi_j(x)\right][-\varphi_k(x)]\mathrm{d}x = 0 \tag{6.27}$$

得方程组

$$\sum_{j=0}^{n} a_j \int_a^b \rho(x)\varphi_k(x)\varphi_j(x)\mathrm{d}x = \int_a^b \rho(x) f(x)\varphi_k(x)\mathrm{d}x,$$

$$(k = 0,1,2,\cdots,n), \quad (6.28)$$

如果采用内积记号

$$(\varphi_k, \varphi_j) = \int_a^b \rho(x)\varphi_k(x)\varphi_j(x)\mathrm{d}x,$$

$$(f, \varphi_k) = \int_a^b \rho(x) f(x)\varphi_k(x)\mathrm{d}x,$$

那么,方程组(6.28)可简写为

$$\sum_{j=0}^{n}(\varphi_k, \varphi_j)a_j = (f, \varphi) \quad (k = 0,1,2,\cdots,n) \qquad (6.29)$$

这是一个包含 $n+1$ 个未知元 $a_0, a_1, \cdots, a_n$ 的 $n+1$ 阶线性代数方程组,写成矩阵形式为

$$\begin{bmatrix} (\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n) \\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n) \\ \vdots & \vdots & \cdots & \vdots \\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (f,\varphi_0) \\ (f,\varphi_1) \\ \vdots \\ (f,\varphi_n) \end{bmatrix} \qquad (6.30)$$

此方程组叫做求 $a_j(j = 0,1,2,\cdots,n)$ 的**法方程**.

显然,其系数行列式就是克莱姆行列式

$$G_n = G_n(\varphi_0, \varphi_1, \cdots, \varphi_n),$$

由于 $\varphi_0, \varphi_1, \cdots, \varphi_n$ 线性无关,故 $G_n \neq 0$,于是上述方程组存在唯一解 $a_k = a_k^*(k = 0,1,2,\cdots,n)$,从而肯定了函数 $f(x)$ 在 Φ 中如果存在最佳平方逼近函数,则必是

$$s^*(x) = \sum_{j=0}^{n} a_j^* \varphi_j(x)$$

此处的 a_j^* 由法方程(6.30)求得.

下面进一步证明,由上述方法确定的 $s^*(x)$ 确是函数 $f(x)$ 在 Φ 中的最佳平方逼近函数.这是因为:对任何的 $s(x) \in \Phi$,均有

$$\int_a^b \rho(x)[f(x) - s^*(x)]^2\mathrm{d}x \leqslant \int_a^b \rho(x)[f(x) - s(x)]^2\mathrm{d}x$$

为此只要考虑

$$\int_a^b \rho(x)[f(x)-s(x)]^2\mathrm{d}x-\int_a^b \rho(x)[f(x)-s^*(x)]^2\mathrm{d}x$$

$$=\int_a^b \rho(x)s^2(x)\mathrm{d}x-2\int_a^b \rho(x)f(x)s(x)\mathrm{d}x$$

$$+2\int_a^b \rho(x)f(x)s^*(x)\mathrm{d}x-\int_a^b \rho(x)s^{*2}(x)\mathrm{d}x$$

$$=\int_a^b \rho(x)[s(x)-s^*(x)]^2\mathrm{d}x$$

$$+2\int_a^b \rho(x)[s^*(x)-s(x)]\cdot[f(x)-s^*(x)]\mathrm{d}x.$$

其中 $s^*(x)$ 的系数 a_k^* 满足方程组(6.27),因此

$$\int_a^b \rho(x)[f(x)-s^*(x)]\varphi_k(x)\mathrm{d}x=0 \quad (k=0,1,2,\cdots,n)$$

而 $$s^*(x)-s(x)=\sum_{k=0}^{n}(a_k^*-a_k)\varphi_k(x),$$

所以 $$\int_a^b \rho(x)[s^*(x)-s(x)]\cdot[f(x)-s^*(x)]\mathrm{d}x=0$$

于是 $$\int_a^b \rho(x)[f(x)-s(x)]^2\mathrm{d}x-\int_a^b \rho(x)[f(x)-s^*(x)]^2\mathrm{d}x$$

$$=\int_a^b \rho(x)[s(x)-s^*(x)]^2\mathrm{d}x\geqslant 0$$

即 $$\int_a^b \rho(x)[f(x)-s^*(x)]^2\mathrm{d}x\leqslant\int_a^b \rho(x)[f(x)-s(x)]^2\mathrm{d}x.$$

这就证明了 $s^*(x)$ 确是 $f(x)$ 在 Φ 中的最佳平方逼近函数.

以上不仅证明了函数 $f(x)$ 在 Φ 中的最佳平方逼近函数 $s^*(x)$ 的存在且唯一,同时,也给出了求最佳平方逼近函数 $s^*(x)$ 的构造性方法.

例 6.4 在区间 $\left[\frac{1}{4},1\right]$ 上给定函数 $f(x)=\sqrt{x}$,求其在 $\Phi=\mathrm{span}\{1,x\}$ 中关于 $\rho(x)=1$ 的最佳平方逼近多项式.

解　已知 $\varphi_0 = 1$, $\varphi_1 = x$. 设所求的最佳平方逼近多项式形如 $p_1(x) = a_0 + a_1 x$, 则有

$$\begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} (f, \varphi_0) \\ (f, \varphi_1) \end{bmatrix}$$

先计算六个内积:

$$(\varphi_0, \varphi_0) = \int_{\frac{1}{4}}^{1} 1^2 \mathrm{d}x = \frac{3}{4},$$

$$(\varphi_0, \varphi_1) = (\varphi_1, \varphi_0) = \int_{\frac{1}{4}}^{1} x \mathrm{d}x = \frac{15}{32},$$

$$(\varphi_1, \varphi_1) = \int_{\frac{1}{4}}^{1} x^2 \mathrm{d}x = \frac{21}{64},$$

$$(f, \varphi_0) = \int_{\frac{1}{4}}^{1} \sqrt{x} \mathrm{d}x = \frac{7}{12},$$

$$(f, \varphi_1) = \int_{\frac{1}{4}}^{1} x\sqrt{x} \mathrm{d}x = \frac{31}{80}.$$

则得 a_1, a_0 应满足方程组

$$\begin{cases} \frac{3}{4} a_0 + \frac{15}{32} a_1 = \frac{7}{12} \\ \frac{15}{32} a_0 + \frac{21}{64} a_1 = \frac{31}{80} \end{cases}$$

解得

$$\begin{cases} a_0 = \frac{10}{27} \\ a_1 = \frac{88}{135} \end{cases}$$

即线性函数 $p_1(x) = \frac{88}{135} x + \frac{10}{27}$ 为 $f(x) = \sqrt{x}$ 在区间 $\left[\frac{1}{4}, 1\right]$ 上关于 $\rho(x) = 1$ 的一次最佳平方逼近多项式. 与例6.1比较, 平方逼近的算法比较简单.

再例如, 在上述一般性讨论中, 若取权函数 $\rho(x) \equiv 1$, 求函数 $f(x) \in [0,1]$ 在 $\Phi = \mathrm{span}\{1, x, x^2, \cdots, x^n\}$ 中的 n 次最佳平方逼

近多项式 $s^*(x) = a_0^* + a_1^* x + a_2^* x^2 + \cdots + a_n^* x^n$

此时，有

$$(\varphi_j, \varphi_k) = \int_0^1 x^{j+k} \mathrm{d}x = \frac{1}{j+k+1}, \quad (j,k = 0,1,2,\cdots,n)$$

记 $\quad (f, \varphi_k) = \int_0^1 x^k \cdot f(x) \mathrm{d}x = \mathrm{d}_k \quad (k = 0,1,2,\cdots,n)$

代入式(6.30)，可得法方程的系数矩阵 $\boldsymbol{G}$ 为

$$\boldsymbol{G} = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n+1} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix} \tag{6.31}$$

称 $\boldsymbol{G}$ 为 $n+1$ 阶希尔伯特(Hilbert)矩阵.

再记 $\quad \boldsymbol{d} = (d_0, d_1, \cdots, d_n)^{\mathrm{T}}, \boldsymbol{a} = (a_0, a_1, \cdots, a_n)^{\mathrm{T}}$

于是，法方程为 $\quad \boldsymbol{G}\boldsymbol{a} = \boldsymbol{d}$

解此法方程得

$$a_j = a_j^* (j = 0,1,2,\cdots,n)$$

即为 $s^*(x)$ 的系数，从而求得 $s^*(x)$.

虽然上述求最佳平方逼近的方法比较简单，但方程组(6.30)的系数矩阵往往是病态的，特别当系数矩阵是希尔伯特矩阵时，它是著名的病态矩阵.也即当 n 增大时，解的精确度将由于舍入误差的影响而迅速恶化，一般，当 $n \geqslant 4$，就不能令人满意了，为了克服这个缺点，常用正交多项式系 $\{\psi_k(x)\}(k = 0,1,2,\cdots,n)$ 来构造逼近函数，这就是所谓正交化方法.

二、用正交多项式作函数的平方逼近

设 $f(x) \in c[a,b]$，$\{\psi_k(x)\}$ 是区间 $[a,b]$ 上带权 $\rho(x)$ 的正交函数系，即

$$(\psi_k(x),\ \psi_j(x)) = \begin{cases} 0, & j \neq k \\ A_k > 0, & j = k \end{cases}$$

则在集合 $\psi = \mathrm{span}\{\psi_0, \psi_1, \cdots, \psi_n\}$ 中求解最佳平方逼近函数

$$s^*(x) = \sum_{j=0}^{n} a_j^* \psi_j(x)$$

的问题，可先求法方程(6.30)，此处，因 $\{\psi_k(x)\}$ 是正交函数系，于是法方程是如下的系数矩阵为对角阵的方程组：

$$\begin{cases} (\psi_0, \psi_0) a_0^* & = (f, \psi_0) \\ (\psi_1, \psi_1) a_1^* & = (f, \psi_1) \\ (\psi_n, \psi_n) a_n^* & = (f, \psi_n) \end{cases}$$

因此

$$a_j^* = (f, \psi_j)/(\psi_j, \psi_j) \quad (j = 0, 1, 2, \cdots, n)$$

于是

$$s^*(x) = \sum_{j=0}^{n} \frac{(f, \psi_j)}{(\psi_j, \psi_j)} \psi_j(x) \tag{6.32}$$

若 $\{\psi_k(x)\}$ 还是标准正交的，即 $(\psi_j, \psi_j) = 1$，则 $a_j^* = (f, \psi_j)$，这一表示式的形式更简单.

例如，当所讨论的区间为 $[-1, 1]$，且权函数 $\rho(x) = 1$ 时，可以利用勒让德正交多项式 $p_n(x)$ 在 $[-1, 1]$ 上的正交性，取

$$\varphi_0(x) = p_0(x),\ \varphi_1(x) = p_2(x), \cdots, \varphi_n(x) = p_n(x),$$

此时，法方程(6.30)简化为

$$\begin{bmatrix} (p_0, p_0) & & & \\ & (p_1, p_1) & & \\ & & \ddots & \\ & & & (p_n, p_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (f, p_0) \\ (f, p_1) \\ \vdots \\ (f, p_n) \end{bmatrix}$$

容易得到，上述方程组的解是

$$a_j^* = \frac{(f, p_j)}{(p_j, p_j)} = \frac{2j+1}{2}(f, p_j) \quad (j = 0, 1, 2, \cdots, n) \tag{6.33}$$

就可以求得最佳平方逼近多项式为

$$s^*(x)=\sum_{j=0}^{n}a_j^*p_j(x)=\sum_{j=0}^{n}\frac{2j+1}{2}(f,p_j)p_j(x).$$

这样的计算简化,且避免了求解一个病态方程组.

例 6.5 求$f(x)=e^x$在$[-1,1]$上的三次最佳平方逼近多项式.

解 先计算内积

$$(f,p_0)=\int_{-1}^{1}e^x\mathrm{d}x\approx 2.3504,$$

$$(f,p_1)=\int_{-1}^{1}xe^x\mathrm{d}x\approx 0.7358,$$

$$(f,p_2)=\int_{-1}^{1}\left(\frac{3}{2}x^2-\frac{1}{2}\right)e^x\mathrm{d}x\approx 0.1431,$$

$$(f,p_3)=\int_{-1}^{1}\left(\frac{5}{2}x^3-\frac{3}{2}x\right)e^x\mathrm{d}x\approx 0.02013.$$

代入计算式(6.33) 即得

$$a_0^*=\frac{1}{2}(f,p_0)\approx 1.1752,\ a_1^*=\frac{3}{2}(f,p_1)\approx 1.1036,$$

$$a_2^*=\frac{5}{2}(f,p_2)\approx 0.3578,\ a_3^*=\frac{7}{2}(f,p_3)\approx 0.07046,$$

于是,所求的三次平方逼近多项式为

$$s^*(x)=1.1752p_0+1.1036p_1+0.3578p_2+0.07046p_3$$

如果用x的乘幂来重写上式中的勒让德多项式p_n,则得

$$s^*(x)=0.9963+0.099796x+0.53672x^2+0.17614x^3.$$

如果所给的区间不是$[-1,1]$,而是一般的有限区间$[a,b]$,那么,可以通过变量置换$x=\frac{a+b}{2}+\frac{b-a}{2}t$将它转化为区间$-1\leqslant t\leqslant 1$上的情况来处理.

例 6.6 求函数$f(x)=\sqrt{x}$在$[0,1]$上的一次最佳平方逼近多项式.

解 先令变换 $x=\frac{1}{2}(1+t)$

则 $$f(x)=\frac{1}{\sqrt{2}}\sqrt{1+t}=g(t),\ (-1\leqslant t\leqslant 1)$$

求 $g(t)$ 在区间$[-1,1]$上的一次最佳平方逼近多项式 $q_1(t)$

由 $$a_0^*=\frac{1}{2}(g,p_0)=\frac{1}{2}\int_{-1}^{1}\frac{1}{\sqrt{2}}\sqrt{1+t}\,\mathrm{d}t=\frac{2}{3},$$

$$a_1^*=\frac{3}{2}(g,p_1)=\frac{3}{2}\int_{-1}^{1}\frac{t}{\sqrt{2}}\sqrt{1+t}\,\mathrm{d}t=\frac{6}{15},$$

可知 $$q_1(t)=\frac{2}{3}p_0+\frac{6}{15}p_1=\frac{2}{3}+\frac{6}{15}t,\ (-1\leqslant t\leqslant 1)$$

再把 $t=2x-1$ 代入 $q_1(t)$ 就得到$f(x)=\sqrt{x}$ 在区间$[0,1]$上的一次最佳平方逼近多项式为

$$s^*(x)=\frac{2}{3}+\frac{6}{15}(2x-1)=\frac{4}{15}+\frac{12}{15}x.$$

三、函数展开为切比雪夫级数

由上面的讨论还可得到关于函数的广义傅立叶级数.

定义 6.14 对于函数$f(x)\in C[a,b]$,若在$[a,b]$上对某一正交函数系$\{\psi_k(x)\}(k=0,1,2,\cdots)$写出形如$\sum\limits_{j=0}^{\infty}a_j\psi_j(x)$的级数,其中 a_j 由 $a_j=(f,\psi_j)/(\psi_j,\psi_j)$表示,则称$\sum\limits_{j=0}^{\infty}a_j\psi_j(x)$为函数$f(x)$的广义傅立叶级数.

广义傅立叶级数可看成傅立叶级数的直接推广.可以证明,有关傅立叶级数的主要结论对广义傅立叶级数都成立.

显然,当$\{\psi_k(x)\}$是三角函数系:$1,\cos x,\sin x,\cos 2x,\sin 2x,\cdots$时,上述广义傅立叶级数就是普通的傅立叶级数.

设函数$f(x)$在$[-1,1]$上连续,当正交函数系取为切比雪夫正交多项式函数系$\{T_n(x)\}$时,广义傅立叶级数$\sum\limits_{j=0}^{\infty}a_jT_j(x)$称为函数$f(x)$的切比雪夫级数.

根据上述定义,其中,系数为

$$a_0 = (f, T_0)/(T_0, T_0)$$

$$= \frac{1}{\pi}(f, T_0) = \frac{1}{\pi}\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x,$$

$$a_j = (f, T_j)/(T_j, T_j)$$

$$= \frac{2}{\pi}(f, T_j) = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_j(x)}{\sqrt{1-x^2}}\mathrm{d}x \quad (j = 1,2,\cdots,n)$$

若系数统一表示为

$$a_j = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_j(x)}{\sqrt{1-x^2}}\mathrm{d}x \quad (j = 0,1,2,\cdots,n) \tag{6.34}$$

则函数$f(x)$的切比雪夫级数为

$$f(x) \sim \frac{a_0}{2}T_0 + \sum_{j=0}^{\infty} a_j T_j(x) \tag{6.35}$$

在一般情况下,切比雪夫级数与$f(x)$仅只有上述这样的对应关系,但和傅立叶级数理论相仿,在$f(x)$满足一定条件的情况下,式(6.35)中的相关记号"~"写成为等式,即右端级数是收敛的并以$f(x)$为和,譬如我们能证明:只要函数$f(x)$在$[-1,1]$上的导数$f'(x)$存在且有界,由式(6.35)所确定的级数一致收敛于函数$f(x)$,上述级数也称为$f(x)$的切比雪夫展开式,此时

$$f(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} a_j T_j(x)$$

如取式(6.35)级数的部分和

$$s_n(x) = \frac{a_0}{2}T_0 + \sum_{j=1}^{n} a_j T_j(x)$$

则 $s_n(x)$实际上就是函数$f(x)$在集合

$$T = \mathrm{span}\{T_0(x), T_1(x), \cdots, T_n(x)\}$$

中的最佳平方逼近多项式.

例 6.7 将函数$f(x) = \arcsin x$ 在$[-1,1]$上展开为切比雪夫级数.

解 因$f(x) = \arcsin x$ 是奇函数,由式(6.34)所确定的系数

$$a_{2j} = 0,\ a_{2j+1} = \frac{2}{\pi}\int_{-1}^{1} \arcsin x \cdot \cos[(2j+1)\operatorname{arcsoc} x]\frac{dx}{\sqrt{1-x^2}},$$

令 $x = \cos\theta$，即有

$$a_{2j+1} = \frac{2}{\pi}\int_0^{\pi}\left(\frac{\pi}{2} - \theta\right)\cos[(2j+1)\theta]d\theta$$

$$= \frac{4}{\pi}\frac{1}{(2j+1)^2}$$

所以

$$\arcsin x = \frac{4}{\pi}\left[T_1(x) + \frac{1}{9}T_3(x) + \frac{1}{25}T_5(x) + \frac{1}{49}T_7(x) + \cdots\right]$$

若取 $\arcsin x \approx p_7(x)$

$$= \frac{4}{\pi}\left[T_1(x) + \frac{1}{9}T_3(x) + \frac{1}{25}T_5(x) + \frac{1}{49}T_7(x)\right]$$

把 $T_j(x)(j = 1,3,5,7)$ 的表达式代入上式右端，得

$$p_7(x) = \frac{4}{\pi}\left(\frac{76}{105}x + \frac{248}{315}x^3 - \frac{288}{175}x^5 + \frac{64}{49}x^7\right), \quad -1 \leqslant x \leqslant 1.$$

设 $q_7(x)$ 是函数 $\arcsin x$ 的麦克劳林级数 $n = 7$ 的部分和，那么，在区间$[-1,1]$ 上用例6.7所得的 $p_7(x)$ 近似代替$\arcsin x$ 的精确度比用 $q_7(x)$ 高得多，原因是麦克劳林级数 $n = 7$ 的部分和逼近 $\arcsin x$ 只在 $x = 0$ 的近旁才有良好的精确度，而切比雪夫级数的部分和却是在区间$[-1,1]$ 上 $f(x)$ 的最佳平方逼近，其最佳逼近是对整个区间$[-1,1]$ 而言的，用切比雪夫级数的部分和来逼近函数 $f(x)$ 计算方便，精确度高. 因此函数的切比雪夫展开式常常用作函数在整个区间的近似计算，有最经济展开的称号.

6.5 曲线拟合的最小二乘法

一、拟合问题

曲线拟合问题，亦即测量数据的拟合，这是一个既古老，但又非常实用的问题，特别在现代计算机迅速发展的时期，则拟合问题

就更显得重要了.

曲线拟合问题是这样提出的,设我们已经获得一组看上去杂乱无章的实验数据(x_i, y_i),$(i = 1,2,\cdots,m \gg n)$,我们希望从中找出规律来,也即希望构造一个近似函数 $s(x)$ 去逼近所求函数 $y = f(x)$.虽然第五章中介绍的插值法是处理数据近似的一种数值方法,但是用来解决这里提出的一类问题,是有明显缺陷的.

首先,这类问题的数据(x_i, y_i) 是由实验提供的,本身往往带有测试误差,如果要求所得的近似函数曲线精确无误地通过所有的点(x_i, y_i),就会使曲线保留着一切测试误差,这是我们所不希望的.

其次,由实验提供的数据个数往往很多,如果采用多项式插值,必然得到次数较高的插值多项式,这样,不但计算麻烦,而且插值多项式的收敛性、稳定性一般不好,逼近效果往往较差;如果用样条函数插值,虽然有很好的数值性质,且计算量也不大,但存放参数 M_i 的量很大,且没有一个统一的数学公式来表示,也带来不便.因此,怎样从给定的数据表出发,寻找一个简单合理的函数近似表达式来拟合给定的一组数据,正是本节要讨论的主要内容.这里所说的"拟合",即不要求所作的曲线完全通过所有的数据点,只要求所得的近似曲线能反映数据的基本趋势.数据拟合在实际中有广泛的应用,它的实质是离散情况下的最佳平方逼近,其基本思想和处理方法也具有相似性.其几何解释是:求一条曲线,使数据点均在离此曲线的上方或下方不远处,如图 6-3 所画的曲线称为拟合曲线.

二、曲线拟合的最小二乘法

下面讨论这样一个问题:对于给定的一组数据$(x_i, y_i)(i = 1, 2,\cdots, m)$,如何在给定的函数类 Φ 中寻找一个"最好"的函数 $s(x)$,并用 $s(x)$ 近似代替$f(x)$.

实际应用得比较多的是线性最小二乘法,它的一般提法如下:

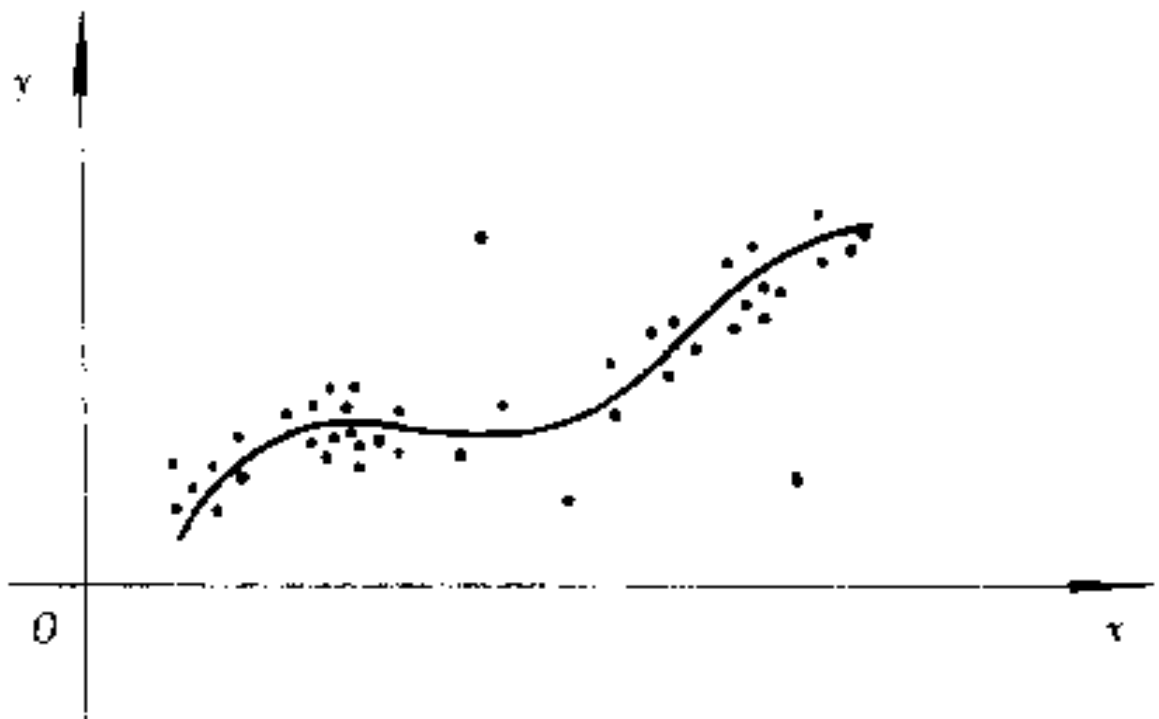


图 6-3

对于给定的数据$(x_i, y_i)(i = 1,2,\cdots,m)$,要求在给定的函数类 $\Phi = \text{span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$ 中找一个函数

$$s^*(x) = a_0^* \varphi_0 + a_1^* \varphi_1 + \cdots + a_n^* \varphi_n = \sum_{k=0}^{n} a_k^* \varphi_k(x) \quad (n < m)$$

使 $s^*(x)$满足

$$\sum_{i=1}^{m} \delta_i^2 = \sum_{i=1}^{m} [s^*(x_i) - f(x_i)]^2$$

$$= \min_{s(x) \in \Phi} \sum_{i=1}^{m} [s(x_i) - f(x_i)]^2 \tag{6.36}$$

这里 $$s(x) = a_0\varphi_0 + a_1\varphi_1 + \cdots + a_n\varphi_n$$

这种求逼近函数 $s^*(x)$的方法就称为曲线拟合的最小二乘法.这里,函数类 Φ 一般可取比较低次的多项式集合或其他较简单的函数类.

实用中,为了使问题提法更具有一般性,常对最小二乘法中 $\sum_{i=1}^{m} \delta_i^2$ 考虑加权平方,即

$$\sum_{i=1}^{m} \omega(x_i)[s(x_i) - f(x_i)]^2$$

其中,$\omega(x)(\geqslant 0)$是$[a, b]$上给定的权函数,它表示不同的点(x_i, y_i)地位的强弱,例如点(x_i, y_i)处的权 $\omega(x_i)$可以用来表示数

据(x_i, y_i)在实验中重复出现的次数，也可以用来表示数 y_i 的准确度，y_i 越准确，它的地位越重要，从而权 $\omega(x_i)$ 也越大.

满足关系式(6.36)的函数 $s^*(x)$ 称为上述最小二乘问题的最小二乘解.

三、最小二乘解的求法

曲线拟合的最小二乘解实际上与前面介绍的函数平方逼近相当，实际上它是最佳平方逼近的离散化情形. 为了求满足式(6.36)的最小二乘解

$$s^*(x) = a_0^* \varphi_0 + a_1^* \varphi_1 + \cdots + a_n^* \varphi_n,$$

就要求多元函数

$$I(a_0, a_1, \cdots, a_n) = \sum_{i=1}^{m} \omega(x_i)\left[\sum_{k=0}^{n} a_k \varphi_k(x_i) - f(x_i)\right]^2$$

的极小值点$(a_0^*, a_1^*, \cdots, a_n^*)$，这与 6.3 节讨论的问题完全类似.

由多元函数取极值的必要条件：$\dfrac{\partial I}{\partial a_j} = 0 \quad (j = 0, 1, 2, \cdots, n)$

即 $$\frac{\partial I}{\partial a_j} = 2\sum_{i=1}^{m} \omega(x_i)\left\{\sum_{k=0}^{n} a_k \varphi_k(x_i) - f(x_i)\right\}\varphi_j(x_i) = 0$$

若引进离散情况下函数的内积记号

$$(\varphi_k, \varphi_j) = \sum_{i=1}^{m} \omega(x_i)\varphi_k(x_i)\cdot\varphi_j(x_i),$$

$$(f, \varphi_j) = \sum_{i=1}^{m} \omega(x_i) f(x_i)\cdot\varphi_j(x_i),$$

则得方程组 $\sum\limits_{k=0}^{n}(\varphi_k, \varphi_j)a_k = (f, \varphi_j) \quad (j = 1, 2, \cdots, n)$

写成矩阵形式为

$$\begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \cdots & \vdots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \cdots & (\varphi_n, \varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (f, \varphi_0) \\ (f, \varphi_1) \\ \vdots \\ (f, \varphi_n) \end{bmatrix} \tag{6.37}$$

这个方程组叫做 $a_j(j=0,1,2,\cdots,n)$的法方程，它的未知元是 $a_0,a_1,\cdots,a_n$，它的系数行列式是 $\varphi_0,\varphi_1,\cdots,\varphi_n$ 的克莱姆行列式，记作 G，由于 $\varphi_0,\varphi_1,\cdots,\varphi_n$ 线性无关，故 $G\neq 0$，所以方程组(6.37)有唯一解 $a_k=a_k^*(k=0,1,2,\cdots,n)$，从而得到函数$f(x)$的最小二乘解为

$$s^*(x)=a_0^*\varphi_0+a_1^*\varphi_1+\cdots+a_n^*\varphi_n$$

并且，类似于6.3节相应内容，可以证明这样得到的 $s^*(x)$确是所求的$f(x)$的最小二乘解，即对于函数类 Φ 中的任何函数 $s(x)$，都有

$$\sum_{i=1}^{m}\omega(x_i)[s^*(x_i)-f(x_i)]^2\leqslant\sum_{i=1}^{m}\omega(x_i)[s(x_i)-f(x_i)]^2.$$

由上述讨论可以得到如下结论：

(1) 对于给定的函数表(x_i,y_i) $(i=1,2,\cdots,m)$，在函数类 $\Phi=\mathrm{span}\{\varphi_0,\varphi_1,\cdots,\varphi_n\}$（$\varphi_0,\varphi_1,\cdots,\varphi_n$ 线性无关）中存在唯一的函数

$$s^*(x)=a_0^*\varphi_0+a_1^*\varphi_1+\cdots+a_n^*\varphi_n,$$

使得关系式(6.36)成立.

(2) 最小二乘解的系数 $a_0^*,a_1^*,\cdots,a_n^*$ 可以通过解法方程(6.37)来获得.

作为曲线拟合的一种常用的情况，如果讨论的是代数多项式拟合，即取

$$\{\varphi_0,\varphi_1,\cdots,\varphi_n\}=\{1,x,x^2,\cdots,x^n\},$$

那么，相应的法方程式(6.37)就是

$$\begin{bmatrix}\sum\omega_i & \sum\omega_i x_i & \cdots & \sum\omega_i x_i^n\\ \sum\omega_i x_i & \sum\omega_i x_i^2 & \cdots & \sum\omega_i x_i^{n+1}\\ \vdots & \vdots & \cdots & \vdots\\ \sum\omega_i x_i^n & \sum\omega_i x_i^{n+1} & \cdots & \sum\omega_i x_i^{2n}\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_n\end{bmatrix}=\begin{bmatrix}\sum\omega_i y_i\\ \sum\omega_i x_i y_i\\ \vdots\\ \sum\omega_i x_i^n y_i\end{bmatrix}\tag{6.38}$$

其中 $\omega_i = \omega(x_i)$，并且将“$\sum_{i=1}^{m}$”简写成“$\sum$”.

此时，$s^*(x) = \sum_{k=0}^{n} a_k^* x^k$，称它为数据拟合多项式，上述拟合称为多项式拟合.

例 6.8 已知函数 $y = f(x)$的实测数据组如表 6-1 所示，试用最小二乘法求多项式曲线与此数据组拟合.

表 6-1

x_i	1	2	3	4	6	7	8
y_i	2	3	6	7	5	3	2

解 (1) 先描点；把表 6-1 中的数据(x_i, y_i)描在坐标纸上，如图 6-4 所示.

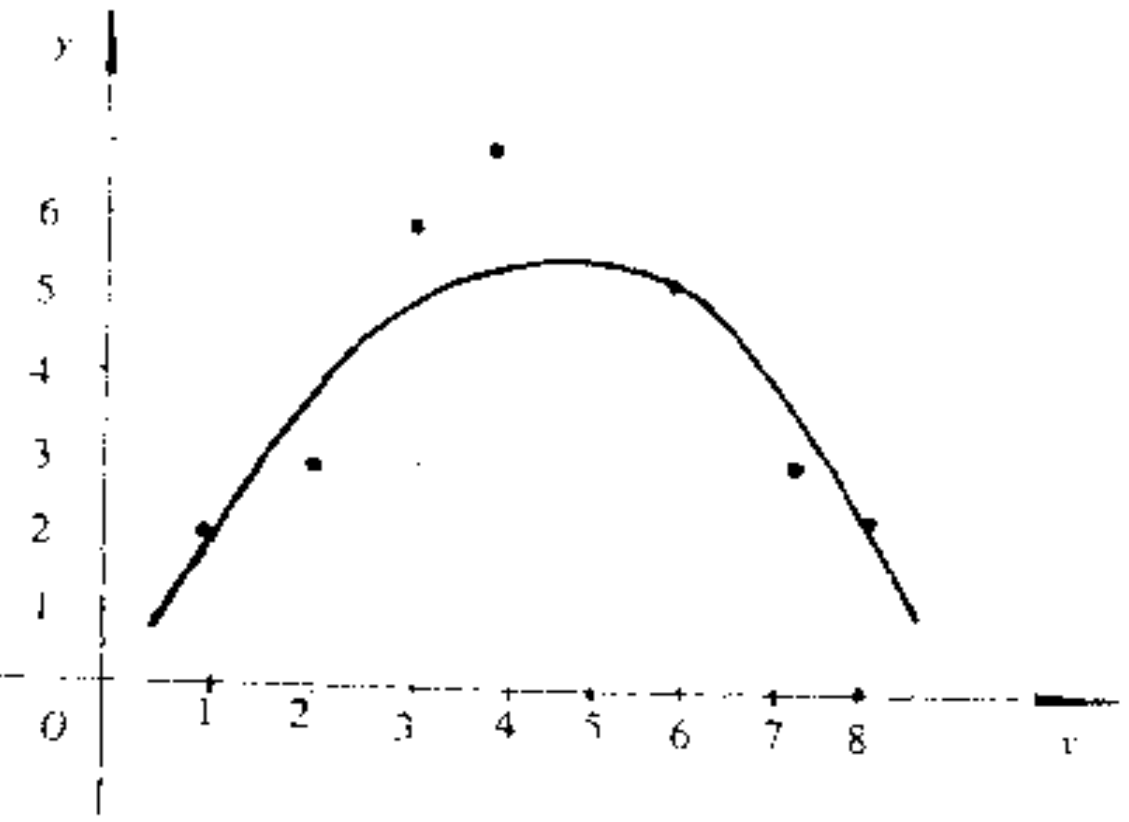


图 6-4

从图中可以看出，这些点位于一条抛物线的附近，因此可以取二次曲线 $y = a_0 + a_1 x + a_2 x^2$ 作为已给函数 $y = f(x)$的拟合函数.

(2) 建立法方程(取 $\omega(x_i) = 1$)

$$\begin{bmatrix} \sum_{i=1}^{7} 1 & \sum_{i=1}^{7} x_i & \sum_{i=1}^{7} x_i^2 \\ \sum_{i=1}^{7} x_i & \sum_{i=1}^{7} x_i^2 & \sum_{i=1}^{7} x_i^3 \\ \sum_{i=1}^{7} x_i^2 & \sum_{i=1}^{7} x_i^3 & \sum_{i=1}^{7} x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{7} y_i \\ \sum_{i=1}^{7} x_i y_i \\ \sum_{i=1}^{7} x_i^2 y_i \end{bmatrix}$$

由已知数据计算出 $\sum_{i=1}^{7} x_i$, $\sum_{i=1}^{7} x_i^2$, $\sum_{i=1}^{7} x_i^3$, $\sum_{i=1}^{7} x_i^4$, $\sum_{i=1}^{7} y_i$, $\sum_{i=1}^{7} x_i y_i$, $\sum_{i=1}^{7} x_i^2 y_i$ 的值.

然后将算得的结果代入法方程,即得 a_0, a_1, a_2 应满足的线代数方程组

$$\begin{cases} 7a_0 + 31a_1 + 179a_2 = 28 \\ 31a_0 + 179a_1 + 1171a_2 = 121 \\ 179a_0 + 1171a_1 + 8147a_2 = 635 \end{cases}$$

(3) 求法方程得 $a_0 = -1.3185, a_1 = 3.4321, a_2 = -0.3864$, 故所求的二次拟合曲线为 $y = -1.3185 + 3.4321x - 0.3864x^2$.

在最小二乘逼近中如何选择数学模型是很重要的,即如何根据所给定的(x_i, y_i)来选择 Φ,一般常用的方法是取

$$\Phi = \text{span}\{1, x, x^2, \cdots, x^n\},$$

但这样取的结果,当 n 较大时,法方程病态,需另作处理,而 n 较小有时并不合适.通常根据物理意义或$(x_i, y_i)(i = 1, 2, \cdots, m)$数据分布的大致图形选择相应的数学模型.但必须指出,我们以上讨论的是线性最小二乘问题,即 $s(x)$是形如 $s(x) = \sum_{j=0}^{n} a_j \varphi_j(x)$的线性组合,有的数学模型表面上不是线性模型,但可通过变换化为线性模型,同样可以使用.例如当取指数函数 $y = ae^{bx}$(a, b 为待定常数)作为拟合函数时,因为 y 是一个关于 a, b 的非线性模型,因此首先应通过适当的变换把 $y = ae^{bx}$化为线性模型,然后再利用最小二乘法来求解.

为此,对函数 $y = ae^{bx}$两边取对数得 $\ln y = \ln a + bx$.

若令　　$u = \ln y$，$A = \ln a$，$B = b$，

则就得到了线性模型：　　$u = A + Bx$，

它关于 A，B 是线性的，于是 A，B 所满足的方程组为

$$\begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} (u, \varphi_0) \\ (u, \varphi_1) \end{bmatrix}$$

此处 $\varphi_0 = 1$，$\varphi_1 = x$，由方程组解出 A^*，B^* 后再由 $a^* = \mathrm{e}^{A^*}$，$b^* = B^*$ 求得 $s^*(x) = a^* \mathrm{e}^{b^* x}$.

例 6.9　设已知一组实验数据如表 6-2 所示.

表 6-2

i	1	2	3	4	5	6
x_i	0	0.5	1	1.5	2	2.5
y_i	2.0	1.0	0.9	0.6	0.4	0.3

试求它的最小二乘拟合曲线　(取 $\omega(x) \equiv 1$).

解　先将已给数据点 $(x_i, y_i)(i = 1, 2, \cdots, 6)$ 描在 xOy 坐标系中，可见这些点接近一指数曲线，因而可取指数函数 $y = a\mathrm{e}^{bx}$ 作为拟合函数.

按上述讨论知，可作变换，令 $A = \ln a$，$B = b$，得 $u = A + Bx$.

相应的　$(\varphi_0, \varphi_0) = \sum_{i=1}^{6} 1 = 6$，

$$(\varphi_0, \varphi_1) = (\varphi_1, \varphi_0) = \sum_{i=1}^{6} x_i = 7.5,$$

$$(\varphi_1, \varphi_1) = \sum_{i=1}^{6} x_i^2 = 13.75,$$

$$(u, \varphi_0) = \sum_{i=1}^{6} \ln y_i = -2.043302,$$

$$(u, \varphi_1) = \sum_{i=1}^{6} x_i \ln y_i = -5.714112.$$

由此得法方程

$$\begin{cases} 6A + 7.5B = -2.043302 \\ 7.5A + 13.75B = -5.714112 \end{cases}$$

解得 $$\begin{cases} A = 0.562302 \\ B = -0.722282 \end{cases}$$

从而 $a^* = e^A = 1.754708$, $b^* = B = -0.722282$

因此,所求拟合曲线为

$$y = 1.754708e^{-0.722282x}.$$

上面的讨论已从原则上解决了最小二乘意义下曲线的拟合问题,由于方程比较简单,实际中应用广泛,特别是,因为任何连续函数至少在一个较小的邻域内可以用多项式任意逼近,因此有多项式作数据拟合,有它的特殊重要性.从而,在许多实际问题中,不论具体函数关系如何,都可用多项式作近似拟合,但我们也已知,在用多项式拟合时,当 n 较大时($n \geqslant 7$),其法方程的系数矩阵的条件数一般较大,所以往往是病态的,因而给求解工作带来了困难.近年来,产生一些直接解线性最小二乘问题的新方法,例如正交三角化方法.此外,常用改变函数类 Φ 的基底 $\varphi_0, \varphi_1, \cdots, \varphi_n$ 来改善方程状态的做法,例如常用的方法是把 $\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$ 正交化.

四、用正交函数作最小二乘拟合

如果函数系 $\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$ 是关于点集 $\{x_i\}(i = 1, 2, \cdots, m)$ 带权 $\omega(x_i)(i = 1, 2, \cdots, m)$ 的正交函数系.

即
$$(\varphi_j, \varphi_k) = \sum_{i=1}^{m} \omega(x_i)\varphi_j(x_i)\varphi_k(x_i) = \begin{cases} 0, & j \neq k \\ A_k > 0, & j = k \end{cases}.$$

则法方程(6.37)化为

$$\begin{bmatrix} (\varphi_0, \varphi_0) & & & \\ & (\varphi_1, \varphi_1) & & \\ & & \ddots & \\ & & & (\varphi_n, \varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (f, \varphi_0) \\ (f, \varphi_1) \\ \vdots \\ (f, \varphi_n) \end{bmatrix}$$

从而解得

$$a_k^* = \frac{(f,\varphi_k)}{(\varphi_k,\varphi_k)} = \frac{\sum_{i=1}^{m}\omega(x_i)f(x_i)\varphi_k(x_i)}{\sum_{i=1}^{m}\omega(x_i)\varphi_k^2(x_i)} \quad (k=0,1,2,\cdots,n),$$

所以
$$s^*(x) = \sum_{k=0}^{n}\frac{(f,\varphi_k)}{(\varphi_k,\varphi_k)}\varphi_k(x).$$

在实际应用中，常用首项系数为 1 的正交多项式系 $\{p_0(x),p_1(x),\cdots,p_n(x)\}$ 代替函数系 $\{\varphi_0(x),\varphi_1(x),\cdots,\varphi_n(x)\}$，从而有

$$s^*(x) = \sum_{k=0}^{n}\frac{(f,p_k)}{(p_k,p_k)}p_k(x) \tag{6.39}$$

下面我们根据给定的节点 $x_1, x_2, \cdots, x_m$ 以及权函数 $\omega(x)>0$，构造出带权 $\omega(x)$ 的正交多项式 $\{p_k(x)\}$ $(k=0,1,2,\cdots,n)$.

首项系数为 1 的正交多项式系 $\{p_0(x),p_1(x),\cdots,p_n(x)\}$ 可用下述递推公式表示：

$$\begin{cases} p_0(x) = 1 \\ p_1(x) = x - \alpha_1 \\ p_{k+1}(x) = (x-\alpha_{k+1})p_k(x) - \beta_k p_{k-1}(x) \end{cases} \tag{6.40}$$

$$(k=1,2,\cdots,n-1) \quad (n\leqslant m)$$

这里的 $p_k(x)$ 是首项系数为 1 的 k 次多项式.

根据 $p_k(x)$ 的正交性，可得

$$\alpha_{k+1} = \frac{(xp_k(x),p_k(x))}{(p_k(x),p_k(x))} = \frac{\sum_{i=1}^{m}\omega(x_i)x_i p_k^2(x_i)}{\sum_{i=1}^{m}\omega(x_i)p_k^2(x_i)} \tag{6.41}$$

$$\beta_k = \frac{(p_k(x),p_k(x))}{(p_{k-1}(x),p_{k-1}(x))} = \frac{\sum_{i=1}^{m}\omega(x_i)p_k^2(x_i)}{\sum_{i=1}^{m}\omega(x_i)p_{k-1}^2(x_i)} \tag{6.42}$$

$$(k=1,2,\cdots,n-1)$$

则计算拟合曲线 $s^*(x)$的步骤可归纳如下：

(1) 利用公式(6.40),(6.41)和式(6.42)依次构造出正交多项式函数系$\{p_0(x),p_1(x),\cdots,p_n(x)\}$.

(2) 求出系数 $a_k^*=\dfrac{(f,p_k)}{(p_k,p_k)},(k=0,1,2,\cdots,n)$

(3) 写出拟合曲线函数

$$s^*(x)=a_0^*p_0(x)+a_1^*p_1(x)+\cdots+a_n^*p_n(x).$$

这里，n 可事先给定或在计算过程中根据误差确定.用这一方法编程序，不用解方程组，只用递推公式，并且当逼近次数增加一次时，只要把程序中循环数加 1,其余不用改变.

例 6.10 已知一组实验数据如下：

x_i	-2	-1	0	1	2
y_i	-1	-1	0	1	1

并设 $\omega(x)\equiv 1$ 求函数 $y=f(x)$的三次拟合曲线.

解 由 $\omega(x_i)\equiv 1$, $p_0(x)=1$ 利用公式(6.40),(6.41)和式(6.42)依次构造出 $p_1(x),p_2(x),p_3(x)$如下：

$$\alpha_1=0,\ p_1(x)=x$$

$$\alpha_2=0,\ \beta_1=2,\ p_2(x)=x^2-2$$

$$\alpha_3=0,\ \beta_2=\frac{7}{5},\ p_3(x)=x^3-\frac{17}{5}x$$

从而求出 $a_0^*=0,\ a_1^*=\dfrac{3}{5},\ a_2^*=0,\ a_3^*=-\dfrac{1}{6}$

所以，所求的三次拟合曲线为 $s^*(x)=\dfrac{1}{6}(7x-x^3)$.

6.6 有理函数逼近

大家都知道，代数多项式有许多优越性：在闭区间上可任意逼近连续函数，不论是函数值的计算、微分、积分等都能方便地运算.

因此，在某种意义下，代数多项式是逼近光滑函数的重要工具.但是代数多项式也有一些弱点，它具有振荡的倾向.这种倾向经常引起多项式逼近的误差界大大超过了它的平均逼近误差.为了寻求一些能够减少逼近误差界的技术，我们将主要考虑那些能够把逼近误差比较均匀地分布在整个逼近区间上的方法.为此，我们需要引进一类新的逼近函数——有理函数.

一个次数为 N 的有理函数 r 是具有下述形式的一个函数

$$r(x)=\frac{p(x)}{q(x)}$$

其中 p 和 q 为多项式，且两个多项式的次数之和为 N.

因为每一个多项式都是一个有理函数，所以有理函数逼近的误差界将不会超过多项式逼近的误差界.事实上，那些分子和分母具有相同或差不多相同次数的有理函数一般会得到远优于那些需要相同计算工作量的多项式函数所得到的结果.另外，有理函数还能够有效地逼近那些在逼近区间外具有无穷间断性质的函数，而多项式逼近一般不能够适应这种情况.

假设 r 是一个 $N=n+m$ 次有理函数，具有下述形式：

$$r(x)=\frac{p(x)}{q(x)}=\frac{p_0+p_1x+\cdots+p_nx^n}{q_0+q_1x+\cdots q_mx^m} \tag{6.43}$$

式(6.43)将被用来逼近一个在包含原点的闭区间上的函数.不失一般性，我们可假设式(6.43)中的 $q_0=1$.因此，式(6.43)在逼近函数 f 时，有 $N+1$ 个参数 $q_1,q_2,\cdots,q_m;p_0,p_1,\cdots,p_n$ 可供选择.

如果我们选择这些 $N+1$ 个参数，使得 $f^{(k)}(0)=r^{(k)}(0)$，对每个 $k=0,1,\cdots,N$，则我们称这个过程为 Padé 逼近.实际上，Padé 逼近是泰勒多项式逼近在有理函数中的推广.当 $n=N$ 和 $m=0$ 时，Padé 逼近是在零点展开的 N 次泰勒多项式，也就是 N 次麦克劳林多项式.

考虑函数

$$f(x)-r(x)=f(x)-\frac{p(x)}{q(x)}$$

$$=\frac{f(x)q(x)-p(x)}{q(x)}=\frac{f(x)\sum_{i=0}^{m}q_ix^i-\sum_{i=0}^{n}p_ix^i}{q(x)};$$

且假设函数 f 有麦克劳林展开 $f(x)=\sum_{i=0}^{\infty}a_ix^i$. 则有

$$f(x)-r(x)=\frac{\sum_{i=0}^{\infty}a_ix^i\sum_{i=0}^{m}q_ix^i-\sum_{i=0}^{n}p_ix^i}{q(x)} \tag{6.44}$$

我们的目标是选择参数 $q_1,q_2,\cdots,q_m$ 和 $p_0,p_1,\cdots,p_n$ 满足

$$f^{(k)}(0)-r^{(k)}(0)=0,\text{对每个 } k=0,1,\cdots,N.$$

容易证明,这等价于函数 $f-r$ 在原点具有 $N+1$ 重根. 因此,我们想选择 $q_1,q_2,\cdots,q_m$ 和 $p_0,p_1,\cdots,p_n$ 满足式(6.44)右端的分子

$$(a_0+a_1x+\cdots)(1+q_1x+\cdots+q_mx^m)-(p_0+p_1x+\cdots+p_nx^n) \tag{6.45}$$

没有小于或者等于 N 次的幂. 为了简化记号,定义 $p_{n+1}=p_{n+2}=\cdots p_N=0$和 $q_{m+1}=q_{m+2}=\cdots=q_N=0$. 则我们能够表示在式(6.45)中 x^k 前面的系数为

$$\sum_{i=0}^{k}a_iq_{k-i}-p_k.$$

故有理函数的 Padé 逼近可从下述线性方程组的解导出

$$\sum_{i=0}^{k}a_iq_{k-i}-p_k=0,\ k=0,1,\cdots,N$$

这里有 $N+1$ 个未知量 $q_1,q_2,\cdots,q_m;p_0,p_1,\cdots,p_n$ 和 $N+1$ 个方程.

例 6.11 求函数 $f(x)=e^{-x}$ 的分子为三次多项式,分母为二次多项式的 Padé 逼近.

解 函数 $f(x)=e^{-x}$ 的麦克劳林级数展开为

$$e^{-x}=\sum_{i=0}^{\infty}\frac{(-1)^i}{i!}x^i.$$

按照题意,我们要选择参数 p_0,p_1,p_2,p_3 及 q_1 和 q_2 使得下述表达

式没有低于或等于 5 次的幂：

$$\left(1-x+\frac{x^2}{2}-\frac{x^3}{6}+\cdots\right)(1+q_1x+q_2x^2)-(p_0+p_1x+p_2x^2+p_3x^3).$$

展开及合并同类项，可得

$$x^5:-\frac{1}{120}+\frac{1}{24}q_1-\frac{1}{6}q_2=0,\qquad x^2:\frac{1}{2}-q_1+q_2=p_2,$$

$$x^4:\ \frac{1}{24}-\frac{1}{6}q_1+\frac{1}{2}q_2=0,\qquad x^1:\ -1+q_1=p_1,$$

$$x^3:-\frac{1}{6}+\frac{1}{2}q_1-q_2=p_3,\qquad x^0:\ 1=p_0.$$

这个方程组的解为

$$p_0=1,\ p_1=-\frac{3}{5},\ p_2=\frac{3}{20},\ p_3=-\frac{1}{60},\ q_1=\frac{2}{5},\ \text{和}\ q_2=\frac{1}{20}.$$

故函数 e^{-x} 的 Padé 逼近为

$$r(x)=\frac{1-\frac{3}{5}x+\frac{3}{20}x^2-\frac{1}{60}x^3}{1+\frac{2}{5}x+\frac{1}{20}x^2}.$$

表 6-3 给出了 5 次 Padé 逼近有理函数和 5 次麦克劳林多项式 $p_5(x)$ 在一些节点上的函数值及对应的截断误差. 很明显在这个例子中，Padé 逼近的结果是优于麦克劳林多项式给出的结果.

表 6-3　　有理函数值的截断误差表

x	$p_5(x)$	$\lvert e^{-x}-p_5(x)\rvert$	$r(x)$	$\lvert e^{-x}-r(x)\rvert$
0.2	0.81873067	8.64×10^{-8}	0.81873075	7.55×10^{-9}
0.4	0.67031467	5.38×10^{-6}	0.67031963	4.11×10^{-7}
0.6	0.54875200	5.96×10^{-5}	0.54880763	4.00×10^{-6}
0.8	0.44900267	3.26×10^{-4}	0.44930966	1.93×10^{-5}
1.0	0.36666667	1.21×10^{-3}	0.36781609	6.33×10^{-5}

我们还是来考察上述例 6.11. 分别研究用 $p_5(x)$ 和 $r(x)$ 来计算函数值时所需要的算术运算的次数. 利用乘法的嵌套形式，

$p_5(x)$可表成

$$p_5(x)=1-x\left(1-x\left(\frac{1}{2}-x\left[\frac{1}{6}-x\left(\frac{1}{24}-\frac{1}{120}\right)\right]\right)\right). \tag{6.46}$$

假设所有的系数都已经被表示成十进制小数，则上述嵌套形式式(6.45)中的 $p_5(x)$的计算过程需要 5 次乘法和 5 次加法或减法.利用嵌套形式的乘法，$r(x)$可表成

$$r(x)=\frac{1-x\left(\frac{3}{5}-x\left(\frac{3}{20}-\frac{1}{60}x\right)\right)}{1+x\left(\frac{2}{5}+\frac{1}{20}x\right)}$$

则上述 $r(x)$的计算过程需要 5 次乘法，5 次加或法减法和 1 次除法.因此，从计算工作量来看，多项式逼近较为有利.但是，我们能够通过 n 次连续的除法，改写 $r(x)$为

$$\begin{aligned} r(x) &= -\frac{1}{3}x+\frac{17}{3}+\frac{\left(-\frac{152}{3}x-\frac{280}{3}\right)}{(x^2+8x+20)} \\ &= -\frac{1}{3}x+\frac{17}{3}+\frac{-\frac{152}{3}}{\left(\frac{x^2+8x+20}{x+\frac{35}{19}}\right)} \\ &= -\frac{1}{3}x+\frac{17}{3}+\frac{-\frac{152}{3}}{\left(x+\frac{117}{19}+\frac{\frac{3125}{361}}{\left(x+\frac{35}{19}\right)}\right)} \end{aligned} \tag{6.47}$$

在式(6.47)中，$r(x)$的计算过程需要 1 次乘法，5 次加法或减法和 2 次除法.显然，这时用式(6.46)计算 $p_5(x)$比利用式(6.47)计算 $r(x)$多用了 2 次乘法或除法.

把有理函数逼近表达成式(6.47)的形式被称为**连分式逼近**.由于连分式表示的计算有效性，近来这类经典的逼近技术越来越引起大家的重视.

习 题 6

1. 设 $T_k(x)$是 k 次切比雪夫多项式,证明

(1) $T_m[T_n(x)] = T_{mn}(x)$;

(2) $T_{m+n}(x) + T_{m-n}(x) = 2T_m(x)T_n(x)$;

(3) $T_n(2x^2 - 1) = T_{2n}(x)$.

2. 证明函数系$\{\varphi_j\}_{j=0}^n$线性无关的充要条件是其 Gram 行列式 $G(\varphi_0, \varphi_1, \cdots, \varphi_n) \neq 0$.

3. 证明一组正交函数必为线性无关组.

4. 求下列函数在区间$[-1,1]$上的线性最佳一致逼近多项式:

(1) $f(x) = x^2 + 3x - 5$;

(2) $f(x) = e^x$.

5. 令 M 和 m 分别代表连续函数$f(x)$在区间$[a,b]$上的最大值和最小值.证明$f(x)$在$[a,b]$上的零次最佳一致逼近多项式是 $p(x) = \frac{1}{2}(M + m)$.

6. 试分别求函数$f(x) = \sqrt{1 + x^2}$在区间$[0,1]$上的一次最佳一致逼近多项式和一次最佳平方逼近多项式.

7. 求函数$f(x) = \sin(\pi x)$在$[0,1]$上的二次最佳平方逼近多项式.

8. 设有一组实验数据如下:

x_i	0.00	0.25	0.50	0.75	1.00
$f(x_i)$	1.0000	1.2840	1.6487	2.1170	2.7183

试在函数类 $\Phi = \text{span}\{1, x, x^2\}$中,求它的拟合曲线.

9. 给定实验数据

x_i	1.00	1.25	1.50	1.75	2.00
$f(x_i)$	5.10	5.79	6.53	7.45	8.46

试求经验公式 $y = be^{ax}$.

10. 将函数 $f(x) = \arccos x$ 在 $[-1, 1]$ 上展开为切比雪夫级数.

11. 设在 $0 \leqslant x \leqslant 1$ 上给定函数 $f(x) = 1 - x + x^2 - x^3 + x^4$, 试在容许误差 $\epsilon = 0.008$ 的要求下降低 $f(x)$ 的次数.

12. 求函数 $f(x) = \sin x$ 的分子和分母都是三次多项式的 Padé 逼近.

第七章　数值积分和数值微分

7.1　引　言

本章主要介绍定积分 $I[f]=\int_a^b f(x)\mathrm{d}x$ 的数值计算方法，这是科学计算中经常用到的计算问题.

一、求积公式和代数精确度

由定积分定义知，定积分$\int_a^b f(x)\mathrm{d}x$ 是和的极限，但从数值逼近的观点看，只要能找到一个逼近$f(x)$ 足够好，又易于求积分的简单函数 $p(x)$ 近似代替$f(x)$，就有计算式

$$\int_a^b f(x)\mathrm{d}x \approx \int_a^b p(x)\mathrm{d}x.$$

数值积分就是将定积分$\int_a^b f(x)\mathrm{d}x$ 的计算用有限和形式近似，它的一般做法是将区间$[a,b]$ 划分为 $a \leqslant x_0 < x_1 < x_2 < \cdots < x_n \leqslant b$，在这些节点上被积函数$f(x)$ 的值为 $f(x_i)=f_i(i=0,1,2,\cdots,n)$，则用数值积分方法求积分$\int_a^b f(x)\mathrm{d}x$ 的值，就是用$f(x)$ 在节点 x_i 上的函数值$f(x_i)$ 的某种线性组合来近似逼近它，即

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n}\omega_i f_i \tag{7.1}$$

或者写成

$$\int_a^b f(x)\mathrm{d}x = \sum_{i=0}^{n}\omega_i f_i + R[f] \tag{7.2}$$

式(7.1)与式(7.2)都叫做**数值求积公式**，其中 ω_i 称为**求积系数**，

它是与被积函数$f(x)$无关的常数，公式中的和 $Q[f]=\sum_{i=0}^{n}\omega_i f(x_i)$ 就叫做**求积算式**，而 $R[f]$表示**截断误差**，称为**求积公式的余项**.

数值积分的特点是直接用一些离散节点上的函数值 $f(x_i)$的线性组合计算定积分的近似值，从而将定积分的计算归结为函数值的计算，这就避开了牛顿-莱布尼兹公式中需要寻求原函数的困难，并为用计算机求积分提供了可行性，这类方法常称为机械求积法.

为了构造形如 $\int_a^b f(x)\mathrm{d}x\approx\sum_{i=0}^{n}\omega_i f_i$ 的求积公式，原则上是确定参数 x_i 和 ω_i 的代数问题，使用求积公式时，计算的工作量通常主要花在函数求值上.但要注意，不应把求积公式看作是针对某一特定的函数$f(x)$，而是要把它看作是在$[a,b]$区间上函数整体(一般只考虑连续函数)来讲的.所以求积公式常记为

$$I[f]=Q[f]+R[f]$$

数值求积方法是近似方法，为了保证精度，使一个求积公式具有较好的实际意义，就应该要求它对尽可能多的被积函数$f(x)$都精确成立，在数学上常用代数精确度这个概念来说明.

定义 7.1 若一个求积公式$\int_a^b f(x)\mathrm{d}x\approx\sum_{i=0}^{n}\omega f(x_i)$对于所有次数不超过 m 的代数多项式，等式都精确成立，而对于某一个$m+1$次多项式不能精确成立，则称该求积公式具有 m 次代数精度.

显然，求积公式的代数精度越高，它就越能使更多的被积函数$f(x)$相对应的余项 $R[f]$"很小"，从而具有更好的实际计算意义.求积公式的代数精确度概念是衡量公式逼近好坏的标准之一.

一般而言，用上述定义来直接判断求积公式的代数精度比较麻烦，而由多项式和定积分的性质，显然容易证明上述定义等价于

$$\begin{cases}R[x^k]=I[x^k]-Q[x^k]=0 & (k=0,1,2,\cdots,m)\\ R[x^{m+1}]\neq 0\end{cases}$$

也即，若依次用$f(x)=x^0,x^1,x^2,\cdots,x^m$代入求积公式(7.1)，均有$I[f]=Q[f]$，而用$f(x)=x^{m+1}$代入求积公式有$I[f]\neq Q[f]$，则称该求积公式具有$m$次代数精度.

例 7.1 设有求积公式

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \omega_0 f(-1)+\omega_1 f(0)+\omega_2 f(1)$$

试确定系数$\omega_0,\omega_1,\omega_2$，使上述求积公式的代数精度尽量高，并指出该求积公式所具有的代数精度.

解 令求积公式依次对$f(x)=1,x,x^2$都精确成立，即系数$\omega_0,\omega_1,\omega_2$应满足方程组

$$\begin{cases} \omega_0 + \omega_1 + \omega_2 = \int_{-1}^{1} 1\mathrm{d}x = 2 \\ -\omega_0 + \omega_2 = \int_{-1}^{1} x\mathrm{d}x = 0 \\ \omega_0 + \omega_2 = \int_{-1}^{1} x^2\mathrm{d}x = \dfrac{2}{3} \end{cases}$$

解得 $$\omega_0=\frac{1}{3},\ \omega_1=\frac{4}{3},\ \omega_2=\frac{1}{3}$$

因此，该求积公式应为

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \frac{1}{3}f(-1)+\frac{4}{3}f(0)+\frac{1}{3}f(1).$$

又容易验证，该求积公式对于$f(x)=x^3$也精确成立，但对$f(x)=x^4$，求积公式不能精确成立，因此，该求积公式具有三次代数精度.

二、插值型求积公式

构造数值求积公式的方法很多，常用的方法是利用插值多项式来构造数值求积公式.

具体做法是：对于定积分$I[f]=\int_a^b f(x)\mathrm{d}x$，如果已知被积函

数$f(x)$在区间$[a,b]$上的一组节点x_i上的函数值为$f(x_i)$,构造$L_n(x)$,满足$L_n(x_i)=f(x_i)$,由于多项式的积分很容易计算,因此用插值多项式近似代替$f(x)$求积是一条可行的途径.

取$\int_a^b f(x)\mathrm{d}x \approx \int_a^b L_n(x)\mathrm{d}x = \sum_{i=0}^{n} f(x_i)\cdot\int_a^b l_i(x)\mathrm{d}x$,这样就得到了一个数值求积公式

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n}\omega_i f(x_i), \tag{7.3}$$

其中

$$\omega_i = \int_a^b l_i(x)\mathrm{d}x, \tag{7.4}$$

我们称求积系数由式(7.4)所确定的求积公式(7.3)为插值型求积公式.其余项是

$$\begin{aligned} R[f] &= \int_a^b [f(x) - L_n(x)]\mathrm{d}x \\ &= \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j)\mathrm{d}x, \end{aligned} \tag{7.5}$$

显然,插值型求积公式(7.3)对一切次数不超过n次的多项式都能精确成立,所以含有$n+1$个节点$x_i(i=0,1,2,\cdots,n)$的插值型求积公式至少具有n次代数精度.

特别的,当$f(x)\equiv 1$时,公式$I[f]=\sum_{i=0}^{n}\omega_i f_i$显然成立.故有$\sum_{i=0}^{n}\omega_i = b-a$,即求积系数之和等于积分区间长度.

反之,也容易证明,如果一个求积公式$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n}\omega_i f_i$的代数精度至少是$n$次,那么,它必然可利用插值多项式推导出来的.这只要证明该求积公式的求积系数可表示为$\omega_i=\int_a^b l_i(x)\mathrm{d}x$,其中$l_i(x)$是$n$次插值基函数.事实上,在该求积公式中取$f(x)=l_i(x)$,则有$\int_a^b l_i(x)\mathrm{d}x = \sum_{j=0}^{n}\omega_j l_i(x_j)$,注意到$l_i(x_j)=\delta_{ij}$,则上式右端$\sum_{j=0}^{n}\omega_j l_i(x_j)=\omega_i$,因而有$\omega_i=\int_a^b l_i(x)\mathrm{d}x$成立.综上所

述，我们得到以下结论.

定理7.1 形如$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f_i$的求积公式至少有$n$次代数精度的充分必要条件是它是插值型求积公式.

7.2 牛顿-柯特斯(Newton-Cotes)求积公式

为了避免在插值型求积公式中，遇到计算求积系数ω_i的困难，使求积公式的形式简单，下面我们先讨论等距节点下的插值型求积公式.

一、牛顿-柯特斯求积公式的导出

设将积分区间$[a,b]$分成为n等分，记步长$h=\frac{b-a}{n}$，则求积节点x_i为等距节点：$x_i = a + ih(i=0,1,2,\cdots,n)$，对应函数值$f(x_i)=f_i$为已知，以这些等距节点为插值节点所导出的插值型求积公式就称为牛顿-柯特斯求积公式，简记为N-C公式.

为了便于计算，在等距节点下，将求积公式

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i)$$

改写成

$$\int_a^b f(x)\mathrm{d}x \approx (b-a)\sum_{i=0}^{n} C_i^{(n)} f(x_i), \tag{7.6}$$

由于$\omega_i = \int_a^b l_i(x)\mathrm{d}x$，则式(7.6)中

$$C_i^{(n)} = \frac{\omega_i}{b-a} = \frac{1}{b-a}\int_a^b l_i(x)\mathrm{d}x$$

$$= \frac{1}{b-a}\int_a^b \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}\mathrm{d}x.$$

为了进一步简化，作变换

$$x = a + sh,$$

则 $\mathrm{d}x = h\mathrm{d}s,$

$$x - x_j = (a + sh) - (a + jh) = (s - j)h,$$

$$x_i - x_j = (i - j)h$$

从而得

$$C_i^{(n)} = \frac{(-1)^{n-i}}{n \cdot i! \cdot (n-i)!}\int_0^n \prod_{j=0}^{n}(s-j)\mathrm{d}s \quad (i = 0,1,2,\cdots,n) \tag{7.7}$$

称 $C_i^{(n)}$ 为柯特斯系数，从式(7.7)中容易看出 $C_i^{(n)}(i = 0,1,2,\cdots,n)$ 只依赖于 n，而与积分区间和被积函数都无关，因此，只要给出 n，就能算出柯特斯系数.并且，在求出 $C_i^{(n)}$ 后将适用于任意的$f(x)$与任意的$[a,b]$，代入公式(7.6)，即得到相应的牛顿 - 柯特斯公式.此类 N-C 公式也称为闭型的 N-C 公式.

例如，当 $n = 1$ 时，柯特斯系数为 $C_0^{(1)} = \frac{1}{2}, C_1^{(2)} = \frac{1}{2}$，此时得到的求积公式称为梯形公式，即

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}[f(a) + f(b)] \tag{7.8}$$

常记为 $$T = \frac{b-a}{2}[f(a) + f(b)].$$

当 $n = 2$ 时，柯特斯系数为 $C_0^{(2)} = C_2^{(2)} = \frac{1}{6}, C_1^{(2)} = \frac{4}{6}$，此时得到的求积公式称为抛物线求积公式，或辛普生(Simpson)公式，即

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \tag{7.9}$$

为了便于应用，把部分柯特斯系数列在表 7-1 中，利用这张柯特斯系数表，可以很快地写出各种等距节点下的N-C 求积公式.

表 7-1　　柯特斯系数表

n	$C_i^{(n)}$								
1	$\frac{1}{2}$	$\frac{1}{2}$							
2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$						
3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$					
4	$\frac{7}{90}$	$\frac{16}{45}$	$\frac{2}{15}$	$\frac{16}{45}$	$\frac{7}{90}$				
5	$\frac{19}{288}$	$\frac{25}{96}$	$\frac{25}{144}$	$\frac{25}{144}$	$\frac{25}{96}$	$\frac{19}{288}$			
6	$\frac{41}{840}$	$\frac{9}{35}$	$\frac{9}{280}$	$\frac{34}{105}$	$\frac{9}{280}$	$\frac{9}{35}$	$\frac{41}{840}$		
7	$\frac{751}{17280}$	$\frac{3577}{17280}$	$\frac{1323}{17280}$	$\frac{2989}{17280}$	$\frac{2989}{17280}$	$\frac{1323}{17280}$	$\frac{3577}{17280}$	$\frac{751}{17280}$	
8	$\frac{989}{28350}$	$\frac{5888}{28350}$	$\frac{-928}{28350}$	$\frac{10496}{28350}$	$\frac{-4540}{28350}$	$\frac{10496}{28350}$	$\frac{-928}{28350}$	$\frac{5888}{28350}$	$\frac{989}{28350}$

例如　$n = 4$ 时，有

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{90}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] \quad (7.10)$$

其中　$x_i = a + i \cdot \frac{b-a}{4}$　$(i = 0,1,2,3,4)$.

公式(7.10) 称为柯特斯公式.

牛顿 - 柯特斯公式是求积节点为等距情况下的插值型求积公式，因此，至少具有 n 次代数精度，但可证明，当 n 为偶数时，N-C 公式至少具有 $n+1$ 次代数精度.

由此，梯形公式只有一次代数精度，而抛物线公式却具有三次代数精度，柯特斯公式具有五次代数精度. 所以，在使用N-C 公式时，为了既能保证精度，又能节省时间，应尽量选用 n 是偶数的求积公式.

二、牛顿 - 柯特斯公式的余项

我们已知,当函数$f(x)$具有 $n+1$ 阶导数时,N - C公式的余项可表为

$$R[f] = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x - x_j)\mathrm{d}x \tag{7.11}$$

其中,$\xi \in (a,b)$ 且依赖于 x.

下面给出几个常用的低阶N-C公式的余项,即公式的截断误差.

1. 梯形求积公式的余项

定理 7.2 若 $f''(x)$ 在$[a,b]$上连续,则梯形求积公式的余项为

$$R_1[f] = \int_a^b f(x)\mathrm{d}x - \frac{b-a}{2}[f(a) + f(b)]$$

$$= -\frac{(b-a)^3}{12}f''(\eta),\ \eta \in [a,b] \tag{7.12}$$

证明 在N-C公式的余项公式(7.11)中,令 $n=1$,得梯形公式的余项为

$$R_1[f] = \int_a^b \frac{f''(\xi)}{2}(x-a)(x-b)\mathrm{d}x, \xi \in [a,b],$$

且依赖于 x.

由于被积函数$(x-a)(x-b)$在$[a,b]$上不变号,而已知$f''(x)$在$[a,b]$上连续,由积分中值定理知,在$[a,b]$内存在一点 η,使

$$R_1[f] = \frac{f''(\eta)}{2}\int_a^b (x-a)(x-b)\mathrm{d}x = -\frac{(b-a)^3}{12}f''(\eta).$$

2. 抛物线求积公式的余项

定理 7.3 若 $f^{(4)}(x)$ 在$[a,b]$上连续,则抛物线求积公式的余项为

$$R_2[f] = \int_a^b f(x)\mathrm{d}x - \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

$$= -\frac{1}{2880}(b-a)^5 f^{(4)}(\eta) \quad \eta \in [a,b]$$

证明 同样由N-C公式的余项公式(7.11)得

$$R_2[f] = \int_a^b f(x)\mathrm{d}x - \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

$$= \int_a^b \frac{f'''(\xi)}{3!}(x-a)\left(x-\frac{a+b}{2}\right)(x-b)\mathrm{d}x.$$

但此时,因为被积函数$(x-a)\left(x-\frac{a+b}{2}\right)(x-b)$在$[a,b]$上不再保号,故不可能直接应用积分中值定理导出抛物线求积公式的余项表达式,因此必须采用其他方法,其中比较简便的方法是利用带导数的埃尔米特插值多项式.

构造次数不高于三次的埃尔米特插值多项式 $H_3(x)$,要求满足条件:

$$H_3(a) = f(a),\ H_3(b) = f(b),$$

$$H_3\left(\frac{a+b}{2}\right) = f\left(\frac{a+b}{2}\right),\ H_3'\left(\frac{a+b}{2}\right) = f'\left(\frac{a+b}{2}\right),$$

由第五章的讨论可知,

$$f(x) - H_3(x) = \frac{1}{4!}f^{(4)}(\xi)(x-a)\left(x-\frac{a+b}{2}\right)^2(x-b),$$

$$\xi \in [a,b].$$

另一方面,因抛物线求积公式具有三次代数精度,故它对 $H_3(x)$ 能精确成立,即有

$$\int_a^b H_3(x)\mathrm{d}x = \frac{b-a}{6}\left[H_3(a) + 4H_3\left(\frac{a+b}{2}\right) + H_3(b)\right]$$

$$= \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{4}\right) + f(b)\right]$$

故抛物线求积公式的余项为

$$R_2[f] = \int_a^b f(x)\mathrm{d}x - \int_a^b H_3(x)\mathrm{d}x$$

$$= \int_a^b [f(x) - H_3(x)]\mathrm{d}x$$

$$= \int_a^b \frac{f^{(4)}(\xi)}{4!}(x-a)\left(x-\frac{a+b}{2}\right)^2(x-b)\mathrm{d}x.$$

此时，被积函数$(x-a)\left(x-\frac{a+b}{2}\right)^2(x-b)$在$[a,b]$上不变号，故当$f^{(4)}(x)$在$[a,b]$上连续时，由积分中值定理得

$$R_2[f] = \frac{f^{(4)}(\eta)}{4!}\int_a^b (x-a)\left(x-\frac{a+b}{2}\right)^2(x-b)\mathrm{d}x$$

$$= \frac{f^{(4)}(\eta)}{4!}\left[-\frac{(b-a)^5}{120}\right]$$

$$= -\frac{(b-a)^5}{2880}f^{(4)}(\eta) \quad \eta \in [a,b]$$

类似地，由更复杂的计算，可得柯特斯求积公式的余项.

定理7.4 若$f^{(6)}(x)$在$[a,b]$上连续，则柯特斯求积公式的余项为

$$R_c[f] = -\frac{2(b-a)}{945}\left(\frac{b-a}{4}\right)^6 f^{(6)}(\eta), \quad \eta \in [a,b].$$

三、牛顿－柯特斯公式的稳定性和收敛性

实际使用任何求积公式时，除截断误差外，还有舍入误差. 事实上，初始数据$f(x_i)$一般不可避免地有舍入误差，所谓求积公式的数值稳定性问题就是指初始数据$f(x_i)$的舍入误差对计算结果的影响大小问题.

设$f(x_i)$有误差ε_i，即$f(x_i)-\bar{f}(x_i)=\varepsilon_i \quad (i=0,1,2,\cdots,n)$，且设$\varepsilon = \max\limits_{0\leqslant i\leqslant n}|\varepsilon_i|$，那么用N-C公式计算积分近似值时产生的误差的绝对值为

$$|R| = \left|(b-a)\sum_{i=0}^{n} C_i^{(n)} f(x_i) - (b-a)\sum_{i=0}^{n} C_i^{(n)} \bar{f}(x_i)\right|$$

$$\leqslant (b-a)\sum_{i=0}^{n}|C_i^{(n)}|\cdot|\varepsilon_i|$$

$$\leqslant (b-a)\cdot\varepsilon\sum_{i=0}^{n}|C_i^{(n)}|.$$

当 $n \leqslant 7$ 时，$C_i^{(n)}$ 都是正数(见表 7-1).

$\sum_{i=0}^{n} |C_i^{(n)}| = \sum_{i=0}^{n} C_i^{(n)} = 1$，则有

$$|R| \leqslant (b-a) \max_{1 \leqslant i \leqslant n} |f(x_i) - \bar{f}(x_i)| \leqslant (b-a)\varepsilon.$$

可见，只要使 $f(x_i)$ 取得足够精确，则初始数据的误差对计算结果的影响被常数 $(b-a)$ 控制住，对结果影响不大，因此，求积是数值稳定的. 但当 $n \geqslant 8$ 时，因为 $C_i^{(n)}$ 可正可负，因此，随着 n 的增大，初始数据的误差会引起结果误差的成倍增加，这时该求积公式是数值不稳定的.

另外，可以证明，并非一切连续函数当 $n \to \infty$ 时，都有 $R[f] \to 0$，即N-C公式的收敛性也没有保证. 因此，在实际计算中不使用高阶的牛顿-柯特斯公式.

四、复合求积法

在实际计算时，高阶N-C公式一般不用，这是因为，首先，当 n 很大时，柯特斯系数 $C_i^{(n)}$ 多且复杂，其次在 n 较大时，该数值积分公式的舍入误差可能有很大增加，从而导致计算的 $\sum_{i=0}^{n} C_i^{(n)} f(x_i)$ 是数值不稳定的，但若在积分区间较大的情况下，单独用一个低阶的N-C公式来计算积分的近似值，它的精确度又不好，为了提高数值积分的精确度，可利用积分对区间的可加性，很容易地解决这个问题，这就是通常采用的复合求积法.

所谓复合求积法，就是先将积分区间分成若干个小区间，并在每个小区间上采用低阶的N-C公式计算积分的近似值，然后相加得出新的求积公式，这些公式就称为复合求积公式.

1. 复合梯形求积公式

把区间 $[a,b]$ n 等分为 n 个小区间 $[x_i, x_{i+1}]$，其中

$$x_i = a + ih \left(h = \frac{b-a}{n}, i = 0,1,2,\cdots,n-1 \right),$$

并在每个小区间 $[x_i, x_{i+1}]$ 上应用梯形公式，就可导出复合梯形公

式：

$$\int_a^b f(x)\mathrm{d}x = \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} f(x)\mathrm{d}x \approx \sum_{i=0}^{n-1}\frac{h}{2}[f(x_i)+f(x_{i+1})]$$

$$= \frac{h}{2}\Big[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\Big],$$

记

$$T_n = \frac{h}{2}\Big[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\Big] \tag{7.13}$$

式(7.13)称为复合梯形公式.

设 $f''(x)$ 在 $[a,b]$ 上连续，由梯形公式的余项公式可得复合梯形公式 T_n 的余项是

$$R = \int_a^b f(x)\mathrm{d}x - T_n = \sum_{i=1}^{n}\Big[-\frac{1}{12}h^3 f''(\eta_i)\Big],\ \eta_i\in[x_i,x_{i+1}],$$

由连续函数性质知，在 $[a,b]$ 内必有一点 η，使得

$$f''(\eta) = \frac{1}{n}\sum_{i=1}^{n}f''(\eta_i),$$

于是复合梯形公式的误差为

$$R = -\frac{n}{12}h^3 f''(\eta) = -\frac{b-a}{12}h^2 f''(\eta),\quad \eta\in[a,b]. \tag{7.14}$$

完全类似地可推出复合抛物线求积公式及其余项.

2. 复合抛物线求积公式

记 $[x_i,x_{i+1}]$ 的中点为 $x_{i+\frac{1}{2}}$，在每个小区间 $[x_i,x_{i+1}]$ 上应用抛物线求积公式

$$\int_{x_i}^{x_{i+1}} f(x)\mathrm{d}x \approx \frac{h}{6}\Big[f(x_i)+4f\Big(x_{i+\frac{1}{2}}\Big)+f(x_{i+1})\Big],$$

就可导出复合抛物线求积公式：

$$S_n = \sum_{i=0}^{n-1}\frac{h}{6}\Big[f(x_i)+4f\Big(x_{i+\frac{1}{2}}\Big)+f(x_{i+1})\Big]$$

或

$$S_n = \frac{h}{6}\Big[f(a)+4\sum_{i=0}^{n-1}f\Big(x_{i+\frac{1}{2}}\Big)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\Big] \tag{7.15}$$

同理，可得其余项是

$$R = \int_a^b f(x)\mathrm{d}x - S_n = -\frac{b-a}{2880}h^4 f^{(4)}(\eta), \eta \in [a,b] \tag{7.16}$$

复合抛物线求积公式不但容易编排程序上机计算，而且精度也比较高，是一个较好的数值积分方法，应用较广泛.

同理可得，复合柯特斯公式为

$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{90}\left[7f(a) + 32\sum_{i=0}^{n-1} f\left(x_i + \frac{1}{4}\right) + 12\sum_{i=0}^{n-1} f\left(x_{i+\frac{1}{2}}\right)\right.$$
$$\left. + 32\sum_{i=0}^{n-1} f\left(x_i + \frac{3}{4}\right) + 14\sum_{i=1}^{n-1} f(x_i) + 7(b)\right] = C_n \tag{7.17}$$

其中：$h = \frac{b-a}{n}$，$x_{i+\frac{1}{4}} = x_i + \frac{1}{4}h$，$x_{i+\frac{1}{2}} = x_i + \frac{h}{2}$，$x_{i+\frac{3}{4}} = x_i + \frac{3}{4}h$.

余项是

$$R = I[f] - C_n = -\frac{2(b-a)}{945}\left(\frac{h}{4}\right)^6 \cdot f^{(6)}(\eta), \eta \in [a,b] \tag{7.18}$$

其他的N-C公式亦可用类似的方法进行复合，显然，复合的N-C公式仍然是形如 $\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i)$ 的机械求积公式.

例7.2 试分别利用复合求积公式，根据表 7-2 计算积分 $I = \int_0^1 \frac{\sin x}{x}\mathrm{d}x$ 的值.

表 7-2

x	0	1/8	1/4	3/8	1/2
$f(x)$	1	0.9973978	0.9896158	0.9767267	0.9538510
x	5/8	3/4	7/8	1	
$f(x)$	0.9361556	0.9088516	0.8771925	0.8414709	

解 (1) 将积分区间[0,1]八等分，用复合梯形求积公式(7.13)计算得

$$T_8 = \frac{1}{8}\left[\frac{f(0)}{2} + f\left(\frac{1}{8}\right) + f\left(\frac{1}{4}\right) + f\left(\frac{3}{8}\right) + f\left(\frac{1}{2}\right) + f\left(\frac{5}{8}\right)\right.$$

$$\left. + f\left(\frac{3}{4}\right) + f\left(\frac{7}{8}\right) + \frac{f(1)}{2}\right] \approx 0.9456909.$$

(2) 将积分区间[0,1]四等分,用复合抛物线求积公式(7.15)计算得

$$S_4 = \frac{1}{4\times 6}\left\{f(0) + 4\left[f\left(\frac{1}{8}\right) + f\left(\frac{3}{8}\right) + f\left(\frac{5}{8}\right) + f\left(\frac{7}{8}\right)\right]\right.$$

$$\left. + 2\left[f\left(\frac{1}{4}\right) + f\left(\frac{1}{2}\right) + f\left(\frac{3}{4}\right) + f(1)\right]\right\} \approx 0.9460832.$$

(3) 将积分区间[0,1]两等分,用复合柯特斯公式(7.17)计算得

$$C_2 = \frac{1}{2\times 90}\left\{7f(0) + 32\left[f\left(\frac{1}{8}\right) + f\left(\frac{3}{8}\right) + f\left(\frac{5}{8}\right) + f\left(\frac{7}{8}\right)\right]\right.$$

$$\left. + 12\left[f\left(\frac{1}{4}\right) + f\left(\frac{3}{4}\right)\right] + 14f\left(\frac{1}{2}\right) + 7f(1)\right\} \approx 0.9460829.$$

比较上述计算结果,虽然它们都分别用到9个点上的函数值,计算工作量基本相同,然而精度却差别很大,与积分的准确值 $I = 0.9460831$ 比较,显然,在对积分区间作同样分割或利用同样个数的函数值的条件下,复合抛物线求积公式比复合梯形公式的计算结果好,故在实际计算时,复合抛物线求积公式得到较多的应用.

此外,根据定积分的存在定理,容易证明,只要被积函数 $f(x)$ 在区间 $[a,b]$ 上连续,那么当步长 $h \to 0(n \to \infty)$ 时,复合求积值 T_n, S_n, C_n 都收敛于 $\int_a^b f(x)\mathrm{d}x$,而且收敛速度一个比一个快.也容易证明,以上的复合求积法的计算都是数值稳定的.

由于复合求积公式在整个计算过程中的步长是固定不变的,所以,这种求积公式也常叫做定步长的求积公式.

在实际计算时,我们一般可事先根据余项满足精度的要求来确定步长.但是,因为余项中往往含有被积函数 $f(x)$ 的高阶导

数，而估计高阶导数的最大值是困难的，而且用这种“事前估计误差”方法所估计的误差界往往又偏大，这也是定步长求积公式的缺点.为了改变这一状况，而又能适当地确定节点数或等分数，在实际应用中，常采用“事后估计误差”的方法.

所谓事后估计误差的基本思想是，在求数值积分时，将区间逐次对分，利用前后两次的计算结果进行误差估计.

例如，对复合梯形法，其积分值 T_n 的截断误差为

$$I - T_n = -\frac{b-a}{12}h^2 f''(\eta_1),$$

它与 h^2 成正比，若把区间再对分，则 T_{2n} 的截断误差为

$$I - T_{2n} = -\frac{b-a}{12}\left(\frac{h}{2}\right)^2 f''(\eta_2).$$

如果 $f''(x)$ 在$[a,b]$上变化不大，即有 $f''(\eta_1) \approx f''(\eta_2)$.

由上述两个误差式可得

$$\frac{I - T_n}{I - T_{2n}} \approx 4, \text{或 } I - T_{2n} \approx \frac{1}{3}(T_{2n} - T_n),$$

这说明，以 T_{2n} 作为积分的近似值，其误差近似于$\frac{1}{3}(T_{2n} - T_n)$.由此可见，只要区间二分前后算得的两个积分值 T_n 与 T_{2n} 相当接近，就可以保证计算结果 T_{2n} 的误差很小.

同理，对于复合抛物线求积公式，如果 $f^{(4)}(x)$ 在$[a,b]$上变化不大，则有 $I - S_{2n} \approx \frac{1}{4^2-1}(S_{2n} - S_n)$，其中，$s_n$ 及 s_{2n} 分别表示区间为 n 和 $2n$ 等分时按复合抛物线求积公式所求得的积分近似值.

这种直接用计算结果来估计误差的方法，通常称作误差的“事后估计法”.这种事后估计误差的方法在计算机上很容易实现，因而达到了自动选择步长的要求.

五、变步长的求积法

上面介绍的复合求积法对提高积分精度是行之有效的，但在

使用求积公式前，必须给出合适的步长，步长取得太大，精度难以保证；步长太小，则会导致计算量的增加，而事先给出一个恰当的步长又往往是困难的，因此，在实际计算中常常采用所谓的“变步长求积法”.

“变步长的求积法”其基本思想是：在步长逐次折半的过程中，反复用复合求积公式进行计算，直到二分前后的两次积分计算结果之差的绝对值 $|I_{2n}-I_n|$ 小于允许的精度 ε 为止，并取 I_{2n} 作为所求的积分的近似值.采用这种方法计算时，步长 h 不是固定不变的，因此叫做变步长求积法.

我们现在从变步长的求积过程中探讨梯形法的递推计算规律，也即介绍所谓变步长的梯形法则.

设将求积区间 $[a,b]$ 分成 n 等分，则一共有 $n+1$ 个分点，使用复合梯形公式

$$T_n=\frac{h}{2}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right],$$

计算积分值 T_n，需要提供 $n+1$ 个函数值.若将区间分为 $2n$ 等分，则分点增至 $2n+1$ 个，若仍直接使用复合梯形公式计算二分后的积分值 T_{2n}，即

$$T_{2n}=\frac{1}{2}\left(\frac{h}{2}\right)\left[f(a)+2\sum_{i=1}^{2n-1}f(x_i)+f(b)\right]$$

就需要重新提供 $2n+1$ 个函数值.

但注意到在 T_{2n} 的全部分点中，有 $n+1$ 个是二分前的原有分点，重复计算这些“老分点”上的函数值，显然是个浪费.为了避免这种浪费，我们将二分前后两个积分值联系起来加以考虑.注意到每个子区间 $[x_i,x_{i+1}]$ 经过二分后只增加了一个分点

$$x_{i+\frac{1}{2}}=\frac{1}{2}(x_i+x_{i+1}),$$

用复合梯形求积公式求得该子区间上的积分值为

$$\frac{1}{2}\left(\frac{h}{2}\right)\left[f(x_i)+2f(x_{i+\frac{1}{2}})+f(x_{i+1})\right],\quad (i=0,1,2,\cdots,n-1).$$

这里，$h=\frac{b-a}{n}$代表二分前的步长，将每个子区间上的积分值相加，得

$$T_{2n}=\sum_{i=0}^{n-1}\frac{1}{2}\left(\frac{h}{2}\right)\left[f(x_i)+2f(x_{i+\frac{1}{2}})+f(x_{i+1})\right]$$

$$=\frac{1}{2}\sum_{i=0}^{n-1}\frac{h}{2}\left[f(x_i)+f(x_{i+1})\right]+\frac{h}{2}\sum_{i=0}^{n-1}f(x_{i-\frac{1}{2}}),$$

于是，得到梯形法的递推公式为

$$T_{2n}=\frac{1}{2}T_n+\frac{h}{2}\sum_{i=0}^{n-1}f(x_{i+\frac{1}{2}}) \tag{7.19}$$

式(7.19)表明，在计算 T_{2n}时，我们已经算出的结果 T_n 可作为已知数据使用，从而避免了老分点上的函数值的重复计算，而使计算量节省了一半.

完全类似地可推导出其他变步长的求积公式.其中，变步长的复合抛物线求积法是最为常用的一种方法，因为它在计算机上容易实现.如果相邻两次的积分值 S_n 和 S_{2n}在误差范围内的有效数字数值相等，便以 S_{2n}作为所求的积分近似值，否则，再继续将区间折半，直到满足精度要求为止.

例 7.3 用变步长的梯形法求积分 $\int_0^1\frac{\sin x}{x}dx$ 的近似值.

解 先在积分区间[0,1]上使用梯形公式，并令$f(0)=1$，而 $f(1)=0.8414709$，根据梯形求积公式计算得

$$T_1=\frac{1}{2}[f(0)+f(1)]=0.9207355.$$

将积分区间[0,1]二等分，再计算分点 $x=\frac{1}{2}$ 处的函数值 $f\left(\frac{1}{2}\right)=0.9588510$，然后由递推公式(7.19)得

$$T_2=\frac{1}{2}T_1+\frac{1}{2}f\left(\frac{1}{2}\right)=0.9397933,$$

进一步将积分区间[0,1]四等分，再利用递推式(7.19)有

$$T_4 = \frac{1}{2}T_2 + \frac{1}{4}\left[f\left(\frac{1}{4}\right) + f\left(\frac{3}{4}\right)\right] = 0.9445135,$$

这样不断二分下去，计算结果列成表7-3(表中 k 代表二分次数，区间等分数 $n = 2^k$).

表 7-3

k	0	1	2	3	4	5
T_k	0.9207355	0.9397933	0.9445135	0.9456909	0.9459850	0.9460586
k	6	7	8	9	10	11
T_k	0.9460815	0.9460827	0.9460827	0.9460830	0.9460830	0.9460831

梯形法的算法虽然简单，但精度较差，收敛速度缓慢，如何提高收敛速度以节省计算量，自然是一个应该重视的问题.

六、样条函数求积

变步长的求积法虽然能由计算机自动计算出满足精度要求的积分近似值，但只能用于被积函数能以数学公式给出的情况，而对于列表函数是不适用的.对于列表函数，我们可先构造样条插值函数 $s(x)$ 满足所给函数表，然后用 $s(x)$ 代表被积函数，也可以得到一个计算积分的数值公式.例如设已知函数 $y = f(x)$ 在 $n+1$ 个互异点 x_i 上的函数值 $y_i = f(x_i)(i = 0,1,2,\cdots,n)$，构造满足插值条件 $s(x_i) = y_i$ 的三次样条函数 $s(x)$，以 $s(x)$ 代替 $f(x)$ 在 $[a,b]$ 上积分得

$$\int_a^b f(x)\mathrm{d}x \approx \int_a^b s(x)\mathrm{d}x = \sum_{j=1}^{n}\int s_j(x)\mathrm{d}x$$

在区间 $[x_{j-1}, x_j]$ 上，$s_j(x)$ 由前面已推导得的样条函数公式(6.39)给出，则有

$$\int_{x_{j-1}}^{x_j} s_j(x)\mathrm{d}x = \int_{x_{j-1}}^{x_j}\left\{\frac{(x_j - x)^3}{6h_j}M_{j-1} + \frac{(x - x_{j-1})^3}{6h_j}M_j\right.$$

$$+\left(y_{j-1}-\frac{M_{j-1}h_j^2}{6}\right)\frac{x_j-x}{h_j}+\left(y_j-\frac{M_jh_j^2}{6}\right)\frac{x-x_{j-1}}{h_j}\Big\}\mathrm{d}x$$

$$=\frac{h_j}{2}(y_{j-1}+y_j)-\frac{h_j^3}{24}(M_{j-1}+M_j)$$

从而 $$\int_a^b f(x)\mathrm{d}x\approx\sum_{j=1}^{n}\frac{h_j}{2}(y_{j-1}+y_j)-\sum_{j=1}^{n}\frac{h_j^3}{24}(M_{j-1}+M_j)\quad(7.20)$$

上述求积公式只用到被积函数在已知节点 x_i 上的函数值 $f(x_i)$和样条插值函数的二阶导数值 M_j，而 $M_j(j=0,1,2,\cdots,n)$ 的计算，我们在样条插值函数的构造中已介绍，所以积分的近似值能计算出.

并且可以证明，以上用三次样条插值函数构造的数值求积公式(7.20)具有与抛物线求积公式同样的代数精度.

另外，机器上还常采用自适应的求积方法.这种方法当被积函数在整个区间$[a,b]$上变化不均匀，即在有的区间急剧变化，而其余地方变化较平稳，这时为了达到计算精度而又节省工作量，在变化平稳部分可用求积节点较少的公式，而对变化剧烈部分则增加节点，这种根据函数变化决定求积节点的方法就称为自适应方法.它比根据精度要求统一确定节点数目的保守作法要节省较多的计算工作量.

7.3 龙贝格(Romberg)求积算法

一、数值方法中的加速收敛技巧——外推算法

在数值方法中，常常利用一序列 $F_1,F_2,\cdots$，去逼近准确解 F，然后在理论上给出序列$\{F_k\}$收敛于 F 的误差估计，即估计截断误差 $F-F_k$，一个有趣的问题是：能否在截断误差估计的基础上，通过简易的办法，在序列$\{F_k\}$的基础上产生新的序列$\{\tilde{F}_k\}$，而使$\{\tilde{F}_k\}$比$\{F_k\}$更快地收敛于准确解 F.这就是数值方法的加速收敛技巧.让我们通过例子来说明.

对于给定的函数$f(x)$在区间$[a,b]$,由复合梯形公式

$$T_n=\frac{h}{2}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right] \tag{7.21}$$

计算的结果 T_n 可看作为步长 h 的函数 $T_n=T(h)$,容易证明式(7.21)对区间$[a,b]$上的任何可积函数$f(x)$,总有

$$\lim_{h\to 0}T(h)=\int_a^b f(x)\mathrm{d}x=T(0),$$

显然,h 越小,计算越精确,但计算量相当大.在实际问题中,有时 $T(0)$ 是无法求得的,只能得到一系列函数值 $T(h),T\left(\frac{h}{2}\right),\cdots,T\left(\frac{h}{2^k}\right)$,而且,$h$ 越小,计算的难度就越大.于是提出这样一个问题,能否通过 $T(h),T\left(\frac{h}{2}\right),\cdots$,构造出一个新序列,使它更快地收敛于 $T(0)$ 呢?推广到一般情况,即研究能否通过序列 $f(h),f\left(\frac{h}{2}\right),\cdots$,构造出一个新序列,使它更快地收敛于 $f(0)$ 呢?在某种条件下,这是亦得到的,如利用泰勒展开式

$$f(h)=f(0)+hf'(0)+\frac{h^2}{2!}f''(0)+\frac{h^3}{3!}f'''(0)+\cdots \tag{7.22}$$

$$f\left(\frac{h}{2}\right)=f(0)+\frac{h}{2}f'(0)+\frac{1}{2!}\left(\frac{h}{2}\right)^2f''(0)+\frac{1}{3!}\left(\frac{h}{2}\right)^3f'''(0)+\cdots \tag{7.23}$$

如果 $f'(0)\neq 0$,那么 $f(h),f(h/2)$ 逼近 $f(0)$ 的阶都是 $O(h)$,现在将式(7.23)乘以 2,减去式(7.22),就得

$$2f\left(\frac{h}{2}\right)-f(h)=f(0)-\frac{h^2}{4}f''(0)-\frac{h^3}{8}f'''(0)+\cdots,$$

如果 $f''(0)\neq 0$,那么 $2f\left(\frac{h}{2}\right)-f(h)$ 逼近 $f(0)$ 的误差阶为 $O(h^2)$,也就是说,通过简单计算,新序列 $f_1(h)=2f\left(\frac{h}{2}\right)-f(h)$

有可能比老序列 $f(h)$ 更快地收敛于 $f(0)$，既然 $f_1(h)$ 容易获得，一个有趣的问题发生了：是否能从序列 $f_1(h)$ 再产生出新的序列 $f_2(h)$，使它更快收敛于 $f(0)$ 呢？由上式有

$$f_1(h) = f(0) - \frac{h^2}{4}f''(0) - \frac{h^3}{8}f'''(0) + \cdots$$

$$f_1(h/2) = f(0) - \frac{h^2}{16}f''(0) - \frac{h^3}{64}f'''(0) + \cdots$$

后一式子乘以 4 减去前式并除以 3，有

$$\frac{4f_1\left(\frac{h}{2}\right) - f_1(h)}{3} = f(0) + \frac{h^3}{48}f'''(0) + \cdots,$$

如果 $f'''(0) \neq 0$，记

$$f_2(h) = \frac{4f_1\left(\frac{h}{2}\right) - f_1(h)}{3} = \frac{8f\left(\frac{h}{4}\right) - 6f\left(\frac{h}{2}\right) + f(h)}{3}.$$

那么序列 $\{f_2(h)\}$ 逼近 $f(0)$ 的误差阶为 $O(h^3)$，这种利用若干已经算出的近似值作适当组合以求更精确的近似值的加速收敛的方法就称为外推算法.

数值方法中有许多外推算法，下面介绍数值积分中的一种常用的外推算法——龙贝格求积算法.

二、龙贝格求积算法

从变步长梯形法则的计算值可以看到，因梯形公式的精度较差，因此计算得到的积分近似值序列 $\{T_n\}$ 收敛于积分精确值的速度较慢. 但前面在介绍事后误差估计中，根据梯形法的误差公式推得估计积分近似值 T_{2n} 的误差关系式为

$$I - T_{2n} \approx \frac{1}{3}(T_{2n} - T_n),$$

因此，如果用这个误差值作为 T_{2n} 的一种补偿，可以期望所得到

$$\bar{T} = T_{2n} + \frac{1}{3}(T_{2n} - T_n) = \frac{4}{3}T_{2n} - \frac{1}{3}T_n$$

可能是积分精确值 I 的更好的近似结果. 如当 $n = 1$ 时,直接计算有

$$\bar{T} = T_2 + \frac{1}{3}(T_2 - T_1) = \frac{4}{3}T_2 - \frac{1}{3}T_1$$

$$= \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right].$$

这正是在$[a,b]$上应用抛物线求积公式的结果,也即

$$S_1 = \frac{4}{3}T_2 - \frac{1}{3}T_1,$$

为了写出便于记忆的统一公式,我们把此式改写为

$$S_1 = \frac{4}{4-1}T_2 - \frac{1}{4-1}T_1,$$

类似地可以验证,

$$S_2 = \frac{4}{4-1}T_4 - \frac{1}{4-1}T_2,$$

恰好是将$[a,b]$二等分后,在每个小区间上应用抛物线求积公式的结果.

一般地,有

$$S_n = \frac{4}{4-1}T_{2n} - \frac{1}{4-1}T_n$$

S_n 是区间$[a,b]n$ 等分后,在每个小区间上应用抛物线求积公式的结果. 这就是说,用复合梯形法二分前后的两个积分近似值 T_n 与 T_{2n} 按上式作这样简单的线性组合,就可得精度较高的抛物线法的积分近似值 S_n,从而加速了逼近的效果,这种由 T_n 与 T_{2n} 导出 S_n 的过程实际上就是线性外推.

$$S_n = \frac{4T_{2n} - T_n}{4-1} \tag{7.24}$$

式(7.24) 称为梯形加速公式. 类似地可以证明,由复合抛物线公式的误差公式 $R = -\frac{b-a}{180}h^4 \cdot f^{(4)}(\eta)$ 知,其截断误差大致与 h^4 成正比,因此若 $f^{(4)}(x)$ 在$[a,b]$上变化不大,则将步长折半时,误差

将减至原有误差的 1/16，即有$\frac{I - S_{2n}}{I - S_n} \approx \frac{1}{16}$，由此得到

$$I \approx \frac{16}{15}S_{2n} - \frac{1}{15}S_n.$$

不难验证，上式右端的值，其实是积分区间 n 等分后，在每个小区间上用柯特斯公式得到的具有更高精度的复合柯特斯公式的积分值 C_n，即有

$$C_n = \frac{4^2 S_{2n} - S_n}{4^2 - 1} \tag{7.25}$$

式(7.25) 称为抛物线加速公式.

重复同样的手续，根据复合柯特斯公式的误差公式，可进一步导出公式

$$R_n = \frac{4^3 C_{2n} - C_n}{4^3 - 1} \tag{7.26}$$

由式(7.26) 算出的序列 $R_1, R_2, \cdots, R_n \cdots$，计算精度又迅速得到了提高.

按照上述规律，还可以构造出新的求积公式，其线性组合的两个系数分别为$\frac{4^m}{4^m - 1}$ 及$\frac{1}{4^m - 1}$，但当 $m \geqslant 4$ 时，第一个系数接近于 1，第二个系数绝对值很小，因此，这样组合的新公式与前一个公式计算结果差别不大，反而增加了计算的工作量，故计算时只用到式(7.26) 为止，通常称上述这种在变步长的求积过程中运用三个加速公式(7.24)、式(7.25) 和式(7.26) 将变步长的梯形法则得到的粗糙的积分近似值迅速加工成精度较高的积分近似值的求积方法为龙贝格求积算法，为了方便起见，称公式(7.26) 为龙贝格求积公式. 应当注意，龙贝格求积公式(7.26) 已经不属于N-C 公式的范围.

由上述讨论知，龙贝格求积算法是一种区间逐次分半的加速收敛方法，它是在复化梯形求积公式的基础上，应用外推的思想构

造的一种外推算法，是一种重要而实用的数值积分方法.

三、计算步骤及数值例子

在计算机上应用龙贝格积分方法求积分$\int_a^b f(x)\mathrm{d}x$的计算步骤如下：

(1) 准备初值：先用梯形公式计算积分近似值

$$T_1 = \frac{b-a}{2}[f(a)+f(b)].$$

(2) 按变步长梯形法则计算积分近似值

将区间逐次分半，令区间长度$h = \frac{b-a}{2^i}(i = 0,1,2,\cdots)$，计算

$$T_{2n} = \frac{1}{2}T_n + \frac{h}{2}\sum_{i=0}^{n-1} f\left(x_i + \frac{1}{2}\right) \quad (n = 2^i).$$

(3) 按加速公式求加速值：

梯形加速公式：　$S_n = T_{2n} + (T_{2n} - T_n)/3$

抛物线加速公式：　$C_n = S_{2n} + (S_{2n} - S_n)/15$

龙贝格求积公式：　$R_n = C_{2n} + (C_{2n} - C_n)/63.$

(4) 精度控制：直到相邻两次积分值R_{2n}和R_n满足如下关系：

当$|R_{2n}| \leqslant 1$，满足(绝对误差)$|R_{2n} - R_n| < \varepsilon$；

当$|R_{2n}| > 1$，满足(相对误差)$\left|\frac{R_{2n} - R_n}{R_{2n}}\right| < \varepsilon$，

(其中ε为允许的误差限)

则终止计算并取R_{2n}作为积分$\int_a^b f(x)\mathrm{d}x$的近似值，否则将区间再对分，重复(2)，(3)，(4)的计算，直到满足精度为止.

例 7.4　用龙贝格求积算法求积分值$I = \int_0^1 \frac{\sin x}{x}\mathrm{d}x$.

解　按上述步骤计算结果列表于 7-4.

表 7-4

k	T_2^k	S_2^{k-1}	C_2^{k-2}	R_2^{k-3}
0	0.9207355			
1	0.9397933	0.9461459		
2	0.9445135	0.9460869	0.9400830	
3	0.9456909	0.9460833	0.9460831	0.9460831

表 7-4 所列的计算结果表明，对于只运用了三次二分的复合梯形法计算的值，尽管这些近似值的精确度都不高，只有二位准确数字，但经过三次利用加速公式所得到的龙贝格积分值 $R_1 = 0.9460831$，它的每一位数字都是准确数字，可见加速的效果是很明显的.

为了便于编制程序上机计算，人们常引进记号 $T_m^{(k)}$：以 $T_0^{(k)}$ 表示二分 k 次后求得的梯形值，以 $T_m^{(k)}$ 表示序列 $\{T_0^{(m)}\}$ 的 m 次加速值，则按递推公式

$$T_m^{(k)} = \frac{4^m}{4^m - 1} T_{m-1}^{(k+1)} - \frac{1}{4^m - 1} T_{m-1}^{(k)} \quad (m = 1,2,3)$$

可以逐行构造出下面的所谓 T 数表：

表 7-5

k	2^k	$T_0^{(k)}$	$T_1^{(k)}$	$T_2^{(k)}$	$T_3^{(k)}$
0	1	$T_0^{(0)}$			
1	2	$T_0^{(1)}$	$T_1^{(0)}$		
2	4	$T_0^{(2)}$	$T_1^{(1)}$	$T_2^{(0)}$	
3	8	$T_0^{(3)}$	$T_1^{(2)}$	$T_2^{(1)}$	$T_3^{(0)}$
⋮	⋮	⋮	⋮	⋮	⋮

这里，$T_m^{(k)}$ 中的上标 k 和下标 m 分别指出了近似值 $T_m^{(k)}$ 在表中位置的行数与列数(均从 0 数起).

实际计算时，并不是作出 $T_0^{(k)}$ 序列后再作 $T_m^{(k)}$ 序列，而是在算出 $T_0^{(0)}$，$T_0^{(1)}$ 后马上计算 $T_1^{(0)}$；在计算出 $T_1^{(0)}$，$T_1^{(1)}$ 后马上计算 $T_2^{(0)}$；在算出 $T_2^{(0)}$，$T_2^{(1)}$ 后计算 $T_3^{(0)}$，以下类推，其计算顺序如 T 数表所示，是按行计算的.

可以证明，如果$f(x)$ 充分光滑，那么 T 数表中每一列的元素及对角线元素均收敛到所求的积分值 $I=\int_a^b f(x)\mathrm{d}x$，即

$$\lim_{k\to\infty} T_m^{(k)} = I,\quad \lim_{m\to\infty} T_m^{(k)} = I.$$

计算机上实现龙贝格算法就是在区间逐次二分过程中逐步形成 T 数表的具体方法.

应用龙贝格算法，系数有规律，不需要存贮求积系数，占用存贮单元少，精确度较高，因此很适合在计算机上应用.

7.4 高斯(Gauss)求积公式

一、高斯积分问题的提出

在前面建立牛顿 - 柯特斯公式时，为了简化计算，对插值公式中的节点 x_i 限定为等分的节点，然后再定求节系数 ω_i，这种方法虽然简便，但求积公式的精度受到限制. 现在讨论怎样导出具有最高代数精度的求积公式.

由本章7.2已知，过 $n+1$ 个节点的插值型求积公式至少具有 n 次代数精度，那么，最高的代数精度能达到多少呢?

定理 7.5 形如

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n}\omega f(x_i) \tag{7.27}$$

的插值型求积公式的代数精确度最高不超过 $2n+1$ 次.

证明 由代数精度定义，只要找到一个 $2n+2$ 次的多项式，使求积公式(7.27)不能精确成立即可.

因为 $\pi(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$是 $n+1$ 次多项式，则取$f(x)=[\pi(x)]^2$ 时，$f(x)$为 $2n+2$ 次多项式，且只在 $x_0, x_1, \cdots, x_n$，处 $f(x)$ 为零，故 $\int_a^b f(x)\mathrm{d}x = \int_a^b [\pi(x)]^2\mathrm{d}x > 0$，而 $\sum_{i=0}^{n}\omega_i \cdot f(x_i)=0$，因此，当取$f(x)=[\pi(x)]^2$ 时，求积公式(7.27)不能精确成立.

所以，求积公式(7.27)的代数精度不能超过 $2n+1$.

上述定理给出了插值型求积公式的最高代数精确度的一个上界 $2n+1$，那么，最高代数精度为 $2n+1$ 的形如式(7.27)的求积公式存在吗？回答是肯定的，只要适当选择求积节点，可使插值型求积公式的代数精度达到最高，这就是本节要介绍的高斯求积公式.

定义 7.2 若一组节点 $x_0, x_1, \cdots, x_n \in [a,b]$，使插值型求积公式(7.27)具有 $2n+1$ 次代数精度，则称此组节点为高斯点，并称相应的求积公式(7.27)为**高斯求积公式**.

下面我们将阐明，要使求积公式(7.27)的代数精度达到$2n+1$次是完全可能的，并介绍相应的节点 x_i 和系数 ω_i 的求法，由于我们考虑的求积公式是插值型的，只要节点 x_i 确定了，ω_i 也随之确定，因此解决问题的关键在于节点 x_i 的选择，也即构造高斯求积公式的关键是求高斯点.初看起来，似乎我们可以从代数精度的定义出发，通过解一个方程组来同时求取求积系数和节点.但是，在 n 较大的情况下，这种求法是很困难的，所以，一般不采用解方程组而是利用正交多项式的特性来构造高斯求积公式.

二、不带权的高斯求积公式

1. 高斯求积公式存在的条件

定理 7.6 插值型求积公式(7.27)中，节点 $x_i(i=0,1,2,\cdots,n)$是高斯点的充分必要条件是：在区间$[a,b]$上，以 x_i 为零点的 $n+1$次多项式 $\pi(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$与所有次数不超

过 n 的多项式 $p(x)$ 都正交. 即

$$\int_a^b p(x)\pi(x)\mathrm{d}x = 0 \tag{7.28}$$

证明　必要性

设 $p(x)$ 是任意一个次数不超过 n 的多项式. 则 $p(x)\pi(x)$ 是次数不超过 $2n+1$ 的多项式. 因此, 如果 $x_0, x_1, \cdots, x_n$ 是插值型求积公式 $\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i)$ 的高斯点, 则求积公式(7.27)对于 $p(x)\pi(x)$ 能精确成立, 即有

$$\int_a^b p(x)\pi(x)\mathrm{d}x = \sum_{i=0}^{n} \omega_i p(x_i)\pi(x_i),$$

但 $\pi(x_i) = 0 (i = 0, 1, 2, \cdots, n)$

所以 $\int_a^b p(x)\pi(x)\mathrm{d}x = 0$ 成立.

充分性　设 $f(x)$ 是任意给定的次数不超过 $2n+1$ 次的多项式, 用 $\pi(x)$ 除 $f(x)$, 记商为 $p(x)$, 余项为 $q(x)$, 即

$$f(x) = p(x) \cdot \pi(x) + q(x),$$

其中 $p(x)$ 和 $q(x)$ 均为次数不超过 n 次的多项式. 于是

$$\int_a^b f(x)\mathrm{d}x = \int_a^b p(x)\pi(x)\mathrm{d}x + \int_a^b q(x)\mathrm{d}x,$$

由定理假设得 $\int_a^b p(x)\pi(x)\mathrm{d}x = 0$, 故

$$\int_a^b f(x)\mathrm{d}x = \int_a^b q(x)\mathrm{d}x,$$

由于所给的求积公式是插值型的求积公式, 则它至少具有 n 次代数精度, 从而对次数不超过 n 的多项式 $q(x)$ 能精确成立, 即

$$\int_a^b q(x)\mathrm{d}x = \sum_{i=0}^{n} \omega_i q(x_i),$$

再注意到 $\pi(x_i) = 0 (i = 0, 1, 2, \cdots, n)$　所以

$$f(x_i) = p(x_i)\pi(x_i) + q(x_i) = q(x_i)$$

从而有 $\int_a^b q(x)\mathrm{d}x = \sum_{i=0}^{n}\omega_i f(x_i)$，于是

$$\int_a^b f(x)\mathrm{d}x = \sum_{i=0}^{n}\omega_i f(x_i).$$

可见求积公式(7.27)对于一切次数不超过 $2n+1$ 的多项式都能准确成立，所以它是高斯求积公式，其节点 $x_0, x_1, \cdots, x_n$ 就是高斯点.

从以上讨论可知，高斯求积公式具有特点：

(1) 代数精度达到最高 $2n+1$；

(2) 节点是 $[a,b]$ 上的 $n+1$ 次正交多项式的 $n+1$ 个零点.

2. 高斯求积公式的构造

高斯求积公式存在条件的讨论其实已提供了构造高斯求积公式的方法，即只要去找 $[a,b]$ 上的 $n+1$ 次正交多项式的 $n+1$ 个零点，由于正交多项式具有性质：在 $[a,b]$ 上的 $n+1$ 次正交多项式一定有 $n+1$ 个不同的零点，且全部位于 $[a,b]$ 内，所以只要将此 $n+1$ 个零点作为 $n+1$ 次插值多项式的节点，构造出的插值求积公式即为高斯求积公式. 但因为求一般 $[a,b]$ 区间上的 $n+1$ 次正交多项式比较困难，而我们已讨论过 $[-1,1]$ 区间上有带权 $\rho(x)=1$ 的勒让德正交多项式，所以我们先考虑积分区间为 $[-1,1]$ 上的高斯求积公式的构造.

为此只要求出 $n+1$ 次的勒让德正交多项式 $p_{n+1}(x)$ 的 $n+1$ 个零点，就能构造出相应的高斯求积公式. 我们称由勒让德正交多项式的零点构造出来的高斯求积公式为高斯-勒让德求积公式.

下面我们具体地构造在 $[-1,1]$ 区间上各阶高斯-勒让德求积公式.

(1) 一个节点时：因一次勒让德正交多项式 $p_1(x)=x$，其零点为 $x_0=0$，以 $x_0=0$ 为节点构造形如 $\int_{-1}^{1} f(x)\mathrm{d}x \approx \omega_0 f(0)$ 的高斯求积公式，可求得 $\omega_0=2$，这样构造出来的高斯-勒让德求积公

式为$\int_{-1}^{1} f(x)\mathrm{d}x \approx 2f(0)$,这就是大家熟悉的中矩形公式.

(2) 二个节点时:因两次勒让德正交多项式

$$p_2(x) = \frac{1}{2}(3x^2 - 1),$$

其零点为 $x_0 = -\frac{1}{\sqrt{3}}, x_1 = \frac{1}{\sqrt{3}}$,以 $x_0 = \frac{-1}{\sqrt{3}}, x_1 = \frac{1}{\sqrt{3}}$ 为节点构造形如$\int_{-1}^{1} f(x)\mathrm{d}x \approx \omega_0 f\left(-\frac{1}{\sqrt{3}}\right) + \omega_1 f\left(\frac{1}{\sqrt{3}}\right)$的高斯求积公式,由于它的代数精度为 3,所以它对$f(x) = 1, x$,都精确成立,可得

$$\begin{cases} \omega_0 + \omega_1 = 2 \\ \omega_0\left(-\frac{1}{\sqrt{3}}\right) + \omega_1\left(\frac{1}{\sqrt{3}}\right) = 0, \end{cases}$$

由此求得

$$\begin{cases} \omega_0 = 1 \\ \omega_1 = 1 \end{cases}$$

从而得到两点的高斯 - 勒让德求积公式为

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \tag{7.29}$$

完全类似地可求出三点的高斯 - 勒让德求积公式为:

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \frac{5}{9}f\left(-\frac{\sqrt{15}}{5}\right) + \frac{8}{9}f(0) + \frac{5}{9}f\left(\frac{\sqrt{15}}{5}\right) \tag{7.30}$$

由于求积节点 x_i 和系数ω_i 与$f(x)$ 无关,只与勒让德正交多项式有关,所以为了便于应用,对于不同的 n,高斯 - 勒让德求积公式中的节点和系数可制成表,使用时,只要查表就可以方便地写出相应的高斯 - 勒让德求积公式.下面的表7-6给出了 1 ~ 6 个节点的高斯 - 勒让德求积公式的求积节点和系数.

表 7-6

节点数 n	节点 x_n	系数 ω_n
1	0.0000000	2.0000000
2	± 0.5773503	1.0000000
3	± 0.7745967 0.0000000	0.5555556 0.8888889
4	± 0.8611363 ± 0.3399810	0.3478548 0.6521452
5	± 0.9061798 ± 0.5384693 0.0000000	0.2369269 0.4786287 0.5688889
6	± 0.93246951 ± 0.66120939 ± 0.23861919	0.17132449 0.36076157 0.46791393

对于一般的$[a,b]$区间上的积分,可先用变量置换

$$x = \frac{a+b}{2} + \frac{b-a}{2}t$$

将积分区间转化为$[-1,1]$,即

$$\int_b^a f(x)\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{a+b}{2} + \frac{b-a}{2}t\right)\mathrm{d}t,$$

然后用相应的高斯 - 勒让德求积公式来计算.

例如,$n+1$个节点的高斯公式为

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}\sum_{i=0}^{n}\omega_i f\left(\frac{a+b}{2} + \frac{b-a}{2}t_i\right).$$

其中ω_i和t_i可在事先制成的高斯-勒让德求积公式的节点和系数表中查取.

例 7.5 用高斯-勒让德求积公式计算$I = \int_0^1 \frac{\sin x}{x}\mathrm{d}x$.

解　因为积分区间是[0,1]，先作变换 $x = \frac{1}{2}(t+1)$，把[0,1]区间化为[-1,1]区间上的积分，有

$$\int_0^1 \frac{\sin x}{x}\mathrm{d}x = \int_{-1}^1 \frac{\sin\frac{1}{2}(t+1)}{t+1}\mathrm{d}t,$$

用两个节点的高斯-勒让德求积公式，有

$$I \approx \frac{\sin\frac{1}{2}(-0.5773503+1)}{-0.5773503+1} + \frac{\sin\frac{1}{2}(0.5773503+1)}{0.5773503+1}$$
$$= 0.9460411.$$

用三个节点的高斯-勒让德求积公式，有

$$I \approx 0.5555556 \times \frac{\sin\frac{1}{2}(-0.7745967+1)}{-0.7745967+1} + 0.8888889 \times \frac{\sin\frac{1}{2}}{0+1}$$
$$+ 0.5555556 \times \frac{\sin\frac{1}{2}(0.7745967+1)}{0.7745967+1} = 0.9460831.$$

与例7.3和例7.4的求解结果作对比，可见例7.3用复合梯形公式计算，对积分区间[0,1]二分了11次，用2049个函数值，才取得7位准确数字的结果；例7.4用龙贝格求积算法，对区间二分3次，用了9个函数值，得到了相同的结果；本例仅用了3个函数值，也得到七位准确数字的相同结果，这说明高斯求积公式是高精度的.

3. 高斯求积公式的余项

定理7.7　若 $f^{(2n+2)}(x)$ 在[a,b]上连续，则高斯求积公式的余项

$$R[f] = \int_a^b f(x)\mathrm{d}x - \sum_{i=0}^{n}\omega_i f(x_i)$$
$$= \frac{f^{(2n+2)}(\eta)}{(2n+2)!}\int_a^b [\pi(x)]^2\mathrm{d}x \quad \eta \in [a,b] \qquad (7.31)$$

证明

以 $x_0, x_1, \cdots, x_n$ 为节点构造 $f(x)$ 的次数不超过 $2n+1$ 次的埃尔米特插值多项式 $H(x)$，使满足条件

$$H(x_i) = f(x_i), \quad H'(x_i) = f'(x_i), \quad (i = 0,1,2,\cdots,n)$$

则由埃尔米特插值余项公式，有

$$f(x) - H(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}[\pi(x)]^2, \text{ 其中 } \xi \in [a,b].$$

再由高斯求积公式 $\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i)$ 具有 $2n+1$ 次代数精度，所以它对于 $2n+1$ 次的埃尔米特插值多项式 $H(x)$ 能精确成立，即有 $\int_a^b H(x)\mathrm{d}x = \sum_{i=0}^{n} \omega_i H(x_i)$，又由插值条件：$H(x_i) = f(x_i)$，有 $\int_a^b H(x)\mathrm{d}x = \sum_{i=0}^{n} \omega_i f(x_i)$，则高斯求积公式的余项为

$$R[f] = \int_a^b f(x)\mathrm{d}x - \sum_{i=0}^{n} \omega_i f_i = \int_a^b f(x)\mathrm{d}x - \int_a^b H(x)\mathrm{d}x$$

$$= \int_a^b [f(x) - H(x)]\mathrm{d}x = \int_a^b \frac{f^{(2n+2)}(\xi)}{(2n+2)!}[\pi(x)]^2\mathrm{d}x,$$

因为 $f^{(2n+2)}(x)$ 在 $[a,b]$ 上连续，$[\pi(x)]^2 \geqslant 0$，由积分中值定理知，必有 $\eta \in [a,b]$ 使 $R[f] = \frac{f^{(2n+2)}(\eta)}{(2n+2)!}\int_a^b [\pi(x)]^2\mathrm{d}x$.

4. 高斯求积公式的稳定性和收敛性

由于高斯点 x_i 往往是无理数，数据 $f(x_i)$ 必然有计算误差，故人们非常关心高斯求积公式的稳定性. 事实上，与N-C公式相比，高斯求积公式不但具有高精度，而且是数值稳定的求积公式，高斯求积公式的稳定性之所以能够得到保证，是由于它的求积系数具有非负性.

定理 7.8 高斯求积公式中的求积系数 $\omega_i (i = 0,1,2,\cdots,n)$ 全是正的. 且有

$$\omega_i = \int_a^b [l_i(x)]^2\mathrm{d}x \quad (i = 0,1,2,\cdots,n) \tag{7.32}$$

其中 $l_i(x)$ 是 n 次拉格朗日插值基函数.

证明 因为 $l_i(x)$ 是 n 次多项式,所以 $[l_i(x)]^2$ 是 $2n$ 次多项式.而高斯求积公式 $\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i)$ 具有 $2n+1$ 次代数精度,则有

$$\int_a^b [l_i(x)]^2 \mathrm{d}x = \sum_{k=0}^{n} \omega_k \cdot l_i^2(x_k) = \omega_i, \text{即 } \omega_i > 0 \quad (i = 0,1,2,\cdots,n).$$

由于高斯求积公式的系数 ω_i 具有以下特点:

① $\sum_{i=0}^{n} \omega_i = b - a$; ② $\omega_i (i = 0,1,2,\cdots,n)$ 全是正的.

所以,若 $f(x)$ 在 x_i 处有误差

$$f(x_i) - \tilde{f}(x_i) = \varepsilon_i \quad (i = 0,1,2,\cdots,n),$$

并记 $\varepsilon = \max\limits_{0 \leqslant i \leqslant n} |\varepsilon_i|$,则由此引起的结果误差为

$$|\delta| = \left| \sum_{i=0}^{n} \omega_i f(x_i) - \sum_{i=0}^{n} \omega_i \tilde{f}(x_i) \right| = \left| \sum_{i=0}^{n} \omega_i [f(x_i) - \tilde{f}(x_i)] \right|$$

$$\leqslant \sum_{i=0}^{n} |\omega_i| \cdot |f(x_i) - \tilde{f}(x_i)| \leqslant \varepsilon \sum_{i=0}^{n} \omega_i = \varepsilon(b-a).$$

因此,只要在各节点 x_i 处计算 $f(x_i)$ 的初始误差足够小,就能保证求积结果的误差也足够小,由此可以断言,高斯求积公式是稳定的.

关于高斯求积公式的收敛性,有结论:若 $f(x)$ 在区间 $[a,b]$ 上连续,那么当 $n \to \infty$ 时,高斯求积公式 $\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f_i$ 收敛,且收敛到积分值 $\int_a^b f(x)\mathrm{d}x$,即 $\lim\limits_{n \to \infty} \sum_{i=0}^{n} \omega_i f_i = \int_a^b f(x)\mathrm{d}x$.

三、带权的高斯求积公式

若被积函数有某种坏的性态,例如在某一点函数值为无穷大,对这些积分,我们常常考虑如下形式的所谓带权的积分

$$I = \int_a^b \rho(x) f(x)\mathrm{d}x,$$

.其中 $\rho(x) \geqslant 0$ 是 $[a,b]$ 上的权函数. 当 $\rho(x) \equiv 1$ 时,就称为不带权的积分.

设:

$$\int_a^b \rho(x) f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i) \tag{7.33}$$

$$\int_a^b \rho(x) f(x)\mathrm{d}x = \sum_{i=0}^{n} \omega_i f(x_i) + R \tag{7.34}$$

式(7.33)和式(7.34)分别称为不带余项和带余项的带权求积公式,其中 ω_i 是与 $f(x)$ 无关而只与节点和权函数 $\rho(x)$ 有关的常数,构造这种求积公式的意义是将被积函数写成 $[\rho(x)f(x)]$,其中 $f(x)$ 无奇性,$\rho(x)$ 可能有奇性,也即假设坏性态置于 $\rho(x)$ 中,而函数 $f(x)$ 的性态良好,这样求积公式右端的求和部分在计算过程中不会发生奇性.

可以仿照处理普通积分的方法讨论带权的积分. 例如,若求积公式(7.33)对于所有次数不超过 $2n+1$ 次的多项式均能精确成立,则称此求积公式为带权的高斯求积公式,相应的求积节点 $x_i(i=0,1,2,\cdots,n)$ 仍称为高斯点. 且不带权的高斯求积公式的一些结论在这里也基本适用,只需要在前面讨论的积分号下加一个权函数 $\rho(x)$,并把"正交"改成"对权函数 $\rho(x)$ 正交"即可. 因而也有下述类似定理:

定理 7.9 求积公式 $\int_a^b \rho(x) f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i)$ 中,$x_i(i=0,1,2,\cdots,n)$ 是高斯点的充分必要条件是:在区间 $[a,b]$ 上,$\pi(x) = \prod_{j=0}^{n}(x-x_j)$ 是关于权函数 $\rho(x)$ 的 $n+1$ 次正交多项式.

本定理的证明方法与不带权的高斯求积公式中定理 7.6 的证明类似,这里从略.

考虑这种带权积分的推广可对更多定积分构造出高精度的求积公式. 由于正交多项式随权函数不同而异,所以有各种各样的高斯型求积公式.

例如，当 $a=-1, b=1$，且取权函数为 $\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$，则所建立的带权的高斯型求积公式

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\sum_{i=0}^{n}\omega_i f(x_i) \tag{7.35}$$

称作高斯-切比雪夫求积公式. 由于在 $[-1,1]$ 上关于权函数 $\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$ 的正交多项式是切比雪夫多项式，因此该求积公式的高斯点是 $n+1$ 次切比雪夫正交多项式的 $(n+1)$ 个零点，即 $x_i=\cos\left(\dfrac{2i+1}{2(n+1)}\pi\right)$，$(i=0,1,2,\cdots,n)$，由此，通过较复杂的计算可得 $\omega_i=\dfrac{\pi}{n+1}(i=0,1,2,\cdots,n)$，于是得到 n 阶高斯-切比雪夫求积公式为

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\frac{\pi}{n+1}\sum_{i=0}^{n}f\left(\cos\left(\frac{2i+1}{2(n+1)}\pi\right)\right).$$

类似地，在权函数 $\rho(x)=\mathrm{e}^{-x}$，积分区间为 $[0,+\infty]$ 时，可用拉盖尔正交多项式的零点建立求积公式：

$$\int_{0}^{+\infty}\mathrm{e}^{-x}f(x)\mathrm{d}x\approx\sum_{i=0}^{n}\omega_i\cdot f(x_i) \tag{7.36}$$

称式(7.36)为高斯-拉盖尔求积公式.

在权函数 $\rho(x)=\mathrm{e}^{-x^2}$，积分区间为 $(-\infty,+\infty)$ 时，用埃尔米特正交多项式建立的求积公式

$$\int_{-\infty}^{+\infty}\mathrm{e}^{-x^2}\cdot f(x)\mathrm{d}x\approx\sum_{i=0}^{n}\omega_i\cdot f(x_i) \tag{7.37}$$

称为高斯-埃尔米特求积公式.

例 7.6 求形如 $\int_{-1}^{1}\dfrac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\omega_0 f(x_0)+\omega_1 f(x_1)$ 的两点高斯型求积公式.

解　由于求积节点 x_0, x_1 必是 $[-1,1]$ 上带权 $\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$ 的二次切比雪夫正交多项式的零点. 为此，令 $T_2(x)=2x^2-1=0$，得 $x_0=-\dfrac{\sqrt{2}}{2}, x_1=\dfrac{\sqrt{2}}{2}$，则所求高斯型求积公式形如

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\omega_0 f\left(-\frac{\sqrt{2}}{2}\right)+\omega_1 f\left(\frac{\sqrt{2}}{2}\right),$$

由于该求积公式对 $f(x)=1, x$ 都精确成立，可得 ω_0, ω_1 应满足方程组

$$\begin{cases}\omega_0+\omega_1=\pi\\ -\dfrac{\sqrt{2}}{2}\omega_0+\dfrac{\sqrt{2}}{2}\omega_1=0\end{cases}$$

解得

$$\omega_0=\omega_1=\frac{\pi}{2}.$$

则所求的高斯型求积公式为

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\frac{\pi}{2}\left[f\left(-\frac{\sqrt{2}}{2}\right)+f\left(\frac{\sqrt{2}}{2}\right)\right].$$

值得指出的是，利用正交多项式的零点构造高斯型求积公式，这种方法只是针对某些特殊的区间和特殊的权函数才有效，而对于一般的权函数，要构造正交多项式是不容易的，即使有了表达式，要求它的根也比较困难，因此构造一般的高斯型求积公式常常还是从最基本的代数精度的定义出发来构造，但这往往要求解一个非线性的方程组，计算较麻烦.

由上面的讨论可知，高斯求积方法的优点是：积分精度高，精度不够时可以复化，计算上数值稳定，当被积函数 $f(x)$ 连续时积分总是收敛的，且收敛速度较快. 在应用中，用有限元法解工程问题时常采用高斯型求积公式. 高斯积分还可以较好地处理一些近似于奇异积分的问题，例如计算旁义积分和解一些积分方程. 高斯

积分的主要缺点是节点无规律，且当积分精度不满足要求而需增加节点时，所用数据都要重新计算.

7.5 多重积分的求积公式

本章前面几节的求积公式通过适当的改造可应用于对多重积分的近似计算.本节首先考虑如下类型二重积分的近似计算

$$\iint_R f(x,y)\mathrm{d}A$$

这里，R 是平面上的一个长方形区域，就是

$$R = \{(x,y) \mid a \leqslant x \leqslant b,\ c \leqslant y \leqslant d\}.$$

这里，a,b,c 和 d 都是常数.

我们将直接推导出二维的复化辛普生公式.使用相同的技巧，可把其他N-C 公式推广至二维的情况.

假设 x 方向的步长为 $h = (b-a)/2n$，y 方向的步长为 $k = (d-c)/2m$，这里，n 和 m 是用来选择步长大小的正整数.写二重积分为下述累次积分：

$$\iint_R f(x,y)\mathrm{d}A = \int_a^b \left(\int_c^d f(x,y)\mathrm{d}y \right) \mathrm{d}x,$$

我们首先利用一维的复化辛普生公式来求下述积分的值：

$$\int_c^d f(x,y)\mathrm{d}y,$$

视 x 为一个常数.让 $y_0 = c$ 和 $y_j = c + jk, j = 1,2,\cdots,2m$，有

$$\int_c^d f(x,y)\mathrm{d}y = \frac{k}{3}\left[f(x,y_0) + 2\sum_{j=1}^{m-1} f(x,y_{2j}) + 4\sum_{j=1}^{m} f(x,y_{2j-1}) + f(x,y_{2m}) \right] - \frac{(d-c)k^4}{180}\frac{\partial 4 f(x,\mu)}{\partial y^4}$$

其中 $\mu \in [c,d]$.因此有

$$\int_a^b \int_c^d f(x,y)\mathrm{d}y\mathrm{d}x = \frac{k}{3}\int_a^b f(x,y_0)\mathrm{d}x + \frac{2k}{3}\sum_{j=1}^{m-1}\int_a^b f(x,y_{2j})\mathrm{d}x$$

$$+\frac{4k}{3}\sum_{j=1}^{m}\int_a^b f(x,y_{2j-1})\mathrm{d}x+\frac{k}{3}\int_a^b f(x,y_{2m})\mathrm{d}x$$

$$-\frac{(d-c)}{180}k^4\int_a^b\frac{\partial^4 f(x,\mu)}{\partial y^4}\mathrm{d}x \tag{7.38}$$

对上述式(7.38)右端的每一个积分使用一维复化辛普生公式.让 $x_0=a$ 和 $x_i=a+ih$, $i=1,2,\cdots,2n$,则对 $j=0,1,\cdots,2m$,有

$$\int_a^b f(x,y_j)\mathrm{d}x=\frac{h}{3}[f(x_0,y_j)+2\sum_{i=1}^{n-1}f(x_{2i},y_i)+4\sum_{i=1}^{n}f(x_{2i-1},y_j)$$

$$+f(x_{2n},y_j)]-\frac{(b-a)}{180}h^4\frac{\partial^4 f}{\partial x^4}(\xi_j,y_j) \tag{7.39}$$

其中,$\xi_j\in(a,b)$.则可导出下述二维复化辛普生公式:

$$\int_a^b\int_c^d f(x,y)\mathrm{d}y\mathrm{d}x\approx\frac{hk}{9}[f(x_0,y_0)+2\sum_{i=1}^{n-1}f(x_{2i},y_0)+4\sum_{i=1}^{n}f(x_{2i-1},y_0)$$

$$+f(x_{2n},y_0)+2\sum_{j=1}^{m-1}f(x_0,y_{2j})+4\sum_{j=1}^{m-1}\sum_{i=1}^{n-1}f(x_{2i},y_{2j})+8\sum_{j=1}^{m-1}\sum_{i=1}^{n}f(x_{2i-1},y_{2j})$$

$$+2\sum_{j=1}^{m-1}f(x_{2n},y_{2j})+4\sum_{j=1}^{m}f(x_0,y_{2j-1})+8\sum_{j=1}^{m}\sum_{i=1}^{n-1}f(x_{2i},y_{2j-1})$$

$$+16\sum_{j=1}^{m}\sum_{i=1}^{n}f(x_{2i-1},y_{2j-1})+4\sum_{j=1}^{m}f(x_{2n},y_{2j-1})+f(x_0,y_{2m})$$

$$+2\sum_{i=1}^{n-1}f(x_{2i},y_{2m})+4\sum_{i=1}^{n}f(x_{2i-1},y_{2m})+f(x_{2n},y_{2m})] \tag{7.40}$$

误差余项是

$$E=\frac{-k(b-a)h^4}{540}\left[\frac{\partial^4 f(\xi_0,y_0)}{\partial x^4}+2\sum_{j=1}^{m-1}\frac{\partial^4 f(\xi_{2j},y_{2j})}{\partial x^4}\right.$$

$$\left.+4\sum_{j=1}^{m}\frac{\partial^4 f(\xi_{2j-1},y_{2j-1})}{\partial x^4}+\frac{\partial^4 f(\xi_{2m},y_{2m})}{\partial x^4}\right]$$

$$-\frac{(d-c)k^4}{180}\int_a^b\frac{\partial^4 f(x,\mu)}{\partial y^4}\mathrm{d}x. \tag{7.41}$$

如果 $\partial^4 f/\partial x^4$ 和 $\partial^4 f/\partial y^4$ 都是在区域 R 上的连续函数,则误差项可化成

$$E = \frac{-(d-c)(b-a)}{180}\left[h^4 \frac{\partial^4 f}{\partial x^4}(\bar{\eta},\bar{\mu}) + k^4 \frac{\partial^4 f}{\partial y^4}(\hat{\eta},\hat{\mu})\right] \tag{7.42}$$

其中点$(\bar{\eta},\bar{\mu})$和$(\hat{\eta},\hat{\mu})$都在区域 R 中.上述结论可从积分学中值定理及介值定理的二维形式导出.

例 7.7 用二维复化辛普生公式近似计算下述积分(这里使用 $n = 2$ 和 $m = 1$),

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\,dy\,dx.$$

解 这时,步长分别为 $h = 0.15$ 和 $k = 0.25$. 区域剖分成(见图 7-1).

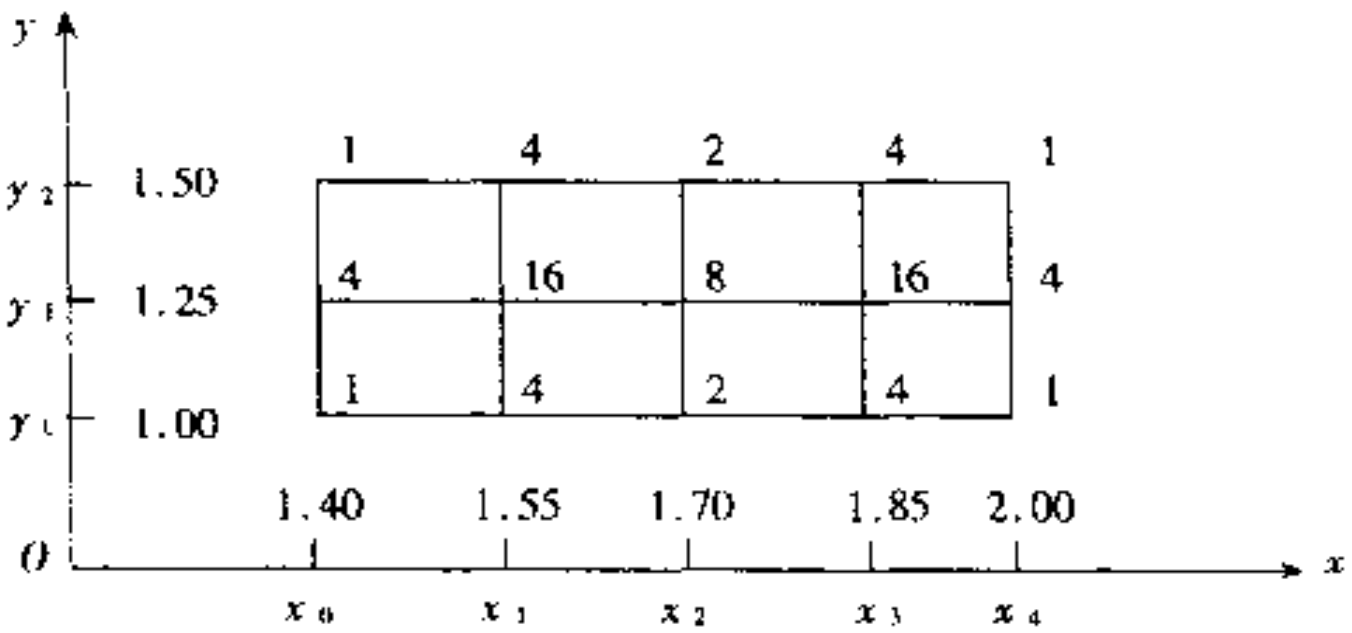


图 7-1

这里节点坐标为(x_i, y_j), $i = 0,1,2,3,4$ 及 $j = 0,1,2$. 设 $w_{i,j}$ 为每一节点对应的函数值在求积公式中的权数. 每一节点的权数已在上述剖分图中直接算出. 则有

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\,dy\,dx \approx \frac{(0.15)(0.25)}{9}\sum_{i=0}^{4}\sum_{j=0}^{2} w_{i,j}\ln(x_i+2y_j)$$

$$= 0.429552438\,7.$$

因为 $\partial^4 f/\partial x^4 = -6/(x+2y)^4$ 和 $\partial^4 f/\partial y^4 = -96/(x+2y)^4$, 则误差估计如下:

$$|E| \leqslant \frac{(0.5)(0.6)}{180}\left[(0.15)^4 \max_R \frac{6}{(x+2y)^4}\right.$$

$$+(0.25)^4 \max_K \frac{96}{(x+2y)^4}\Big] \leqslant 4.72\times 10^{-6}.$$

这个二重积分的精确值为(这儿保留小数点后 10 位)

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\mathrm{d}y\mathrm{d}x = 0.4295545265;$$

故我们的近似积分值的实际误差为 2.1×10^{-6}.

上述方法同样可用于近似三重积分及多重积分.显然,在近似计算多重积分的时候,随着重数的增加需要计算的函数值成倍增加.为了减少计算工作量,我们可以使用更有效率的数值求积法,如高斯公式、龙格积分等.下述例子说明了在二维情况下如何使用九点高斯公式.

例 7.8 考虑使用二维九点高斯公式近似计算二重积分

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\mathrm{d}y\mathrm{d}x.$$

解 首先我们需要把求积区域$[1.4,2.0]\times[1.0,1.5]$通过一个线性变换,变成标准区域$[-1,1]\times[-1,1]$.可检验下述线性变换满足我们的要求:

$$u = 6(2x-3.4),\quad v = 5(2y-2.5)$$

应用上述变量代换,原积分可变成标准区域上的积分:

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\mathrm{d}y\mathrm{d}x = 0.075\int_{-1}^{1}\int_{-1}^{1} \ln(0.3u+0.5v+4.2)\mathrm{d}v\mathrm{d}u.$$

在标准区域上,分别对 v 及 u 使用三点高斯公式.这时,节点坐标为 $u_1 = v_1 = 0, u_0 = v_0 = -0.7745966692$, 和 $u_2 = v_2 = 0.7745966692$.对应的权数为 $w_1 = 0.888888889$ 和 $w_0 = w_2 = 0.5555555556$.因此有

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\mathrm{d}y\mathrm{d}x \approx \sum_{i=0}^{2}\sum_{j=0}^{2} w_i w_j \ln(3u_i+5v_j+4.2)$$

$$= 0.4295545313.$$

尽管上述结果的得到只用了 9 个节点函数值,但这个结果的误差只有 4.8×10^{-9}.同例 7.1 比较,利用 15 个节点的复化辛普生公式

得出的结果的误差比较大，达到 2.1×10^{-6}. 所以，在多维的情况下，使用高斯公式是高效率的数值方法.

对二重积分近似方法的使用并不局限在仅带长方形积分区域的那些积分上面. 上述讨论过的方法通过调整，能用来处理下述形式的二重积分：

$$\int_a^b\int_{c(x)}^{d(x)} f(x,y)\mathrm{d}y\mathrm{d}x \tag{7.43}$$

或

$$\int_c^d\int_{a(y)}^{b(y)} f(x,y)\mathrm{d}x\mathrm{d}y. \tag{7.44}$$

其他区域形状的情况可通过对该区域作适当的剖分，使剖分后的子区域具有上述形状.

下面我们将使用辛普生公式处理上述第一型积分式(7.43)：

$$\int_a^b\int_{c(x)}^{d(x)} f(x,y)\mathrm{d}y\mathrm{d}x$$

使用 x 方向的步长为 $h=(b-a)/2$，y 方向的步长为 x 的函数 $k(x)=(d(x)-c(x))/2$. 则可导出二维的辛普生公式：

$$\int_a^b\int_{c(x)}^{d(x)} f(x,y)\mathrm{d}y\mathrm{d}x \approx \int_a^b \frac{k(x)}{3}[f(x,c(x))+4f(x,c(x)+k(x))$$

$$+f(x,d(x))]\mathrm{d}x \approx \frac{h}{3}\left\{\frac{k(a)}{3}[f(a,c(a))+4f(a,c(a)+k(a))\right.$$

$$+f(a,d(a))]+\frac{4k(a+h)}{3}[f(a+h,c(a+h))$$

$$+4f(a+h,c(a+h)+k(a+h))+f(a+h,d(a+h))]$$

$$\left.+\frac{k(b)}{3}[f(b,c(b))+4f(b,c(b)+k(b))+f(b,d(b))]\right\}$$

基于上述求积公式，通过对曲边矩形的进一步剖分，可导出对应的复化辛普生公式.

例 7.9 对区域 $\{(x,y)\mid 0.1\leqslant x\leqslant 0.5, x^3\leqslant y\leqslant x^2\}$ 的 x 方向 5 等分，y 方向也 5 等分. 然而使用曲边矩形上的复化辛普生

公式近似计算二重积分$\int_{0.1}^{0.5}\int_{x^3}^{x^2}e^{y/x}dydx$.

解

$$\int_{0.1}^{0.5}\int_{x^3}^{x^2}e^{y/x}dydx = \sum_{i,j=0}^{4}\int_{0.1+0.08i}^{0.1+0.08(i+1)}\int_{x^3+0.2(x^2-x^3)j}^{x^3+0.2(x^2-x^3)(j+1)}e^{y/x}dydx$$

$$\approx 0.0333054.$$

上述二重积分的精确值(保留小数点后7位)是0.0333056.

本节例7.9介绍了如何在二维长方形区域上使用高斯求积公式.这类公式在处理复杂区域上的多重积分有局限性.为此,我们有必要介绍一些三角形区域上的高斯求积公式.由于三角形剖分可以较好地逼近复杂的区域边界,故三角形上的高斯求积公式在许多实际的数值计算过程中得到了广泛的应用.首先,我们介绍一下三角形上的面积坐标及有关变换公式.

设$\Delta(i,j,k)$是以i,j,k为顶点的任意三角形,面积为R.我们规定顶点i,j,k的次序按逆时针方向排列.在$\Delta(i,j,k)$内部任取一点$p(x,y)$,坐标为(x,y).通过p点作与三个顶点的联线,把$\Delta(i,j,k)$分成三个子三角形(见图7-2):$\Delta(i,j,p)$,$\Delta(j,k,p)$,和$\Delta(k,i,p)$,其面积分别记为R_k,R_i,R_j.显然$R_i+R_j+R_k=R$.

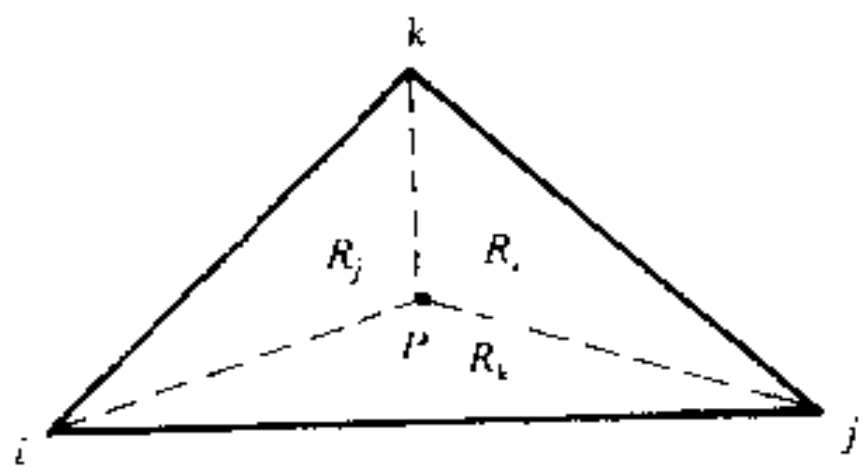


图7-2

令

$$\mathscr{L}_i = R_i/R,\ \mathscr{L}_j = R_j/R,\ \mathscr{L}_k = R_k/R. \qquad (7.45)$$

则有 $\mathscr{S}_i,\mathscr{S}_j,\mathscr{S}_k \geqslant 0,\mathscr{S}_i+\mathscr{S}_j+\mathscr{S}_k=1$. 对给定的一点 $P(x,y)$，有唯一确定的一组数$(\mathscr{S}_i,\mathscr{S}_j,\mathscr{S}_k)$与之对应. 反之，任给一组数$(\mathscr{S}_i,\mathscr{S}_j,\mathscr{S}_k),\mathscr{S}_i,\mathscr{S}_j,\mathscr{S}_k \geqslant 0$，且有 $\mathscr{S}_i+\mathscr{S}_j+\mathscr{S}_k=1$，则按关系式(7.45)也唯一确定一点 $p(x,y)$. 所以同一点 p，既可用直角坐标(x,y)来表示，也可用$(\mathscr{S}_i,\mathscr{S}_j,\mathscr{S}_k)$来表示. 我们称$(\mathscr{S}_i,\mathscr{S}_j,\mathscr{S}_k)$为点 p 的面积坐标. 因为三角形的面积与参考坐标系的选择无关，所以面积坐标也与参考坐标系无关，这是采用面积坐标的一个优点. 面积坐标的这一优点给我们叙述和构造一般三角形区域上的高斯公式带来了很大的方便.

我们知道在解析几何中，任意三角形的面积都能用其对应的三个顶点的直角坐标来表示，即有

$$\begin{cases} R=\dfrac{1}{2}\begin{vmatrix}1 & x_i & y_i\\ 1 & x_j & y_j\\ 1 & x_k & y_k\end{vmatrix}, \quad R_i=\dfrac{1}{2}\begin{vmatrix}1 & x & y\\ 1 & x_j & y_j\\ 1 & x_k & y_k\end{vmatrix}, \\ R_j=\dfrac{1}{2}\begin{vmatrix}1 & x_i & y_i\\ 1 & x & y\\ 1 & x_k & y_k\end{vmatrix}, \quad R_k=\dfrac{1}{2}\begin{vmatrix}1 & x_i & y_i\\ 1 & x_j & y_j\\ 1 & x & y\end{vmatrix}. \end{cases} \tag{7.46}$$

由此可以建立基于某个一般三角形上的面积坐标与直角坐标之间的下述转换关系：

$$\begin{cases} \mathscr{S}_i=\dfrac{1}{2R}[(x_jy_k-x_ky_j)+(y_j-y_k)x+(x_k-x_j)y], \\ \mathscr{S}_j=\dfrac{1}{2R}[(x_ky_i-x_iy_k)+(y_k-y_i)x+(x_i-x_k)y], \\ \mathscr{S}_k=\dfrac{1}{2R}[(x_iy_j-x_jy_i)+(y_i-y_j)x+(x_j-x_i)y]. \end{cases} \tag{7.47}$$

$$\begin{cases} x=x_i\mathscr{S}_i+x_j\mathscr{S}_j+x_k\mathscr{S}_k, \\ y=y_i\mathscr{S}_i+y_j\mathscr{S}_j+y_k\mathscr{S}_k. \end{cases} \tag{7.48}$$

利用式(7.48),我们可把在某个三角形 $\Delta(i,j,k)$ 上的二元函数 $f(x,y)$ 转化成依赖于面积坐标 $\mathscr{L}_i$,$\mathscr{L}_j$ 和 $\mathscr{L}_k$ 的二元函数,这里,$\mathscr{L}_i+\mathscr{L}_j+\mathscr{L}_k=1$.下面我们直接给出两个三点高斯公式和一个四点高斯公式.为简单方便起见,不失一般性,我们设 i,j,k 分别为 1,2,3.

$$\iint_{\Delta} f(x,y)\mathrm{d}x\mathrm{d}y \approx R\sum_{l=1}^{3} W_l f(\mathscr{L}_1^{(l)},\mathscr{L}_2^{(l)},\mathscr{L}_3^{(l)}) \quad (7.49)$$

这里,$W_1=W_2=W_3=0.3333333333$. R 为 $\Delta(1,2,3)$的面积.

$$\mathscr{L}_1^{(1)}=0.6666666667,\mathscr{L}_2^{(1)}=\mathscr{L}_3^{(1)}=0.1666666667.$$
$$\mathscr{L}_2^{(2)}=0.6666666667,\mathscr{L}_1^{(2)}=\mathscr{L}_3^{(2)}=0.1666666667.$$
$$\mathscr{L}_3^{(3)}=0.6666666667,\mathscr{L}_1^{(3)}=\mathscr{L}_2^{(3)}=0.1666666667.$$

这是一个三点高斯公式,其代数精度为二阶.

$$\iint_{\Delta} f(x,y)\mathrm{d}x\mathrm{d}y \approx R\sum_{l=1}^{3} W_l f(\mathscr{L}_1^{(l)},\mathscr{L}_2^{(l)},\mathscr{L}_3^{(l)}) \quad (7.50)$$

这里,$W_1=W_2=W_3=0.3333333333$. R 为 $\Delta(1,2,3)$的面积.

$$\mathscr{L}_1^{(1)}=0.0000000000,\mathscr{L}_2^{(1)}=\mathscr{L}_3^{(1)}=0.5000000000.$$
$$\mathscr{L}_2^{(2)}=0.0000000000,\mathscr{L}_1^{(2)}=\mathscr{L}_3^{(2)}=0.5000000000.$$
$$\mathscr{L}_3^{(3)}=0.0000000000,\mathscr{L}_1^{(3)}=\mathscr{L}_2^{(3)}=0.5000000000.$$

这也是一个三点高斯公式,其代数精度同样也是二阶.

$$\iint_{\Delta} f(x,y)\mathrm{d}x\mathrm{d}y \approx R\sum_{l=1}^{4} W_l f(\mathscr{L}_1^{(l)},\mathscr{L}_2^{(l)},\mathscr{L}_3^{(l)}) \quad (7.51)$$

这里,$W_1=-0.5625000000$, $W_2=W_3=W_4=0.5208333333$.

$$\mathscr{L}_1^{(1)}=\mathscr{L}_2^{(1)}=\mathscr{L}_3^{(1)}=0.3333333333.$$
$$\mathscr{L}_1^{(2)}=0.6000000000,\mathscr{L}_2^{(2)}=\mathscr{L}_3^{(2)}=0.2000000000.$$
$$\mathscr{L}_2^{(3)}=0.6000000000,\mathscr{L}_1^{(3)}=\mathscr{L}_3^{(3)}=0.2000000000.$$
$$\mathscr{L}_3^{(4)}=0.6000000000,\mathscr{L}_1^{(4)}=\mathscr{L}_2^{(4)}=0.2000000000.$$

这是一个四点高斯公式,其代数精度为三阶.

上述三个高斯求积公式的求积节点都是用其对应的面积坐标

给出.在二维情况下,所谓的代数精度是这样定义的:如果某个求积公式对所有的单项式 $x^iy^j(i+j\leqslant n)$都精确成立,且至少在某个单项式 $x^py^q(p+q=n+1)$使得求积公式不能精确成立,则称该求积公式具有 n 次代数精度.

在表 7-7 中,列出了另外六个高斯公式的求积系数和节点坐标.从前面三个已经给出的高斯公式中可以看出公式关于面积坐标$(\mathscr{L}_1,\mathscr{L}_2,\mathscr{L}_3)$是对称分布的.这种对称分布的特点对其他高斯公式也成立.若表中出现的某一个求积节点的面积坐标 $\mathscr{L}_1,\mathscr{L}_2,\mathscr{L}_3$ 互异,则经过置换可生成六个不同的求积节点.若表中出现的 $\mathscr{L}_1,\mathscr{L}_2,\mathscr{L}_3$ 有两个互异,则经过置换后可生成三个不同的求积节点.点(1/3,1/3,1/3)是三角形的形心,若取作求积节点,则只出现一次.

表 7-7

	W_i	$\mathscr{L}_1$	$\mathscr{L}_2$	$\mathscr{L}_3$	重数
六点公式	0.1099517437	0.8168475730	0.0915762135	0.0915762135	3
四阶精度	0.2233815897	0.1081030182	0.4459484909	0.4459484909	3
七点公式	0.3750000000	0.3333333333	0.3333333333	0.3333333333	1
四阶精度	0.1041666667	0.7367124990	0.2379323665	0.0253551346	6
七点公式	0.2250300003	0.3333333333	0.3333333333	0.3333333333	1
五阶精度	0.1250391805	0.7974269854	0.1012865073	0.1012865073	3
	0.1323941528	0.4701420641	0.4701420641	0.0597158718	3
九点公式	0.2059505048	0.1249495032	0.4375252484	0.4375252484	3
五阶精度	0.0636914143	0.7971126519	0.1654099274	0.0374774208	6
12 点公式	0.0508449064	0.8738219710	0.0630890145	0.0630890145	3
六阶精度	0.1167862757	0.5014265097	0.2492867452	0.2492867452	3
	0.0828510756	0.6365024991	0.3103524510	0.0531450498	6
13 点公式	-0.1495700445	0.3333333333	0.3333333333	0.3333333333	1
七阶精度	0.1756152574	0.4793080678	0.2603459661	0.2603459661	3
	0.0533472356	0.8697397942	0.0651301029	0.0651301029	3
	0.0771137609	0.6384441886	0.3128654960	0.4869031543	6

7.6 数值微分

在微积分中,当函数可用初等函数有限次的复合及四则运算

来表示时，该函数可用导数定义或求导法则来求其对应的导函数. 然后，当函数$f(x)$用表格形式给出时，通常只能用近似方法求其节点上的导数值，这一类方法我们称为数值微分方法. 另外，这一节推导出的数值微分公式对我们今后进一步研究微分方程数值解是非常有帮助的.

假定函数 $f \in C^2[a,b]$ 和任一节点 $x_0 \in [a,b]$. 利用 $x_1 = x_0 + h \in [a,b]$ 和 x_0，计算其对应的一次插值函数 $P_{0,1}(x)$. 则有

$$\begin{aligned} f(x) &= P_{0,1}(x) + \frac{(x-x_0)(x-x_1)}{2!} f''(\xi(x)) \\ &= \frac{f(x_0)(x-x_0-h)}{-h} + \frac{f(x_0+h)(x-x_0)}{h} \\ &\quad + \frac{(x-x_0)(x-x_0-h)}{2} f''(\xi(x)) \end{aligned}$$

其中，$\xi(x) \in [a,b]$. 上式两边求导可得

$$\begin{aligned} f'(x) &= \frac{f(x_0+h)-f(x_0)}{h} + \mathrm{D}_x\left[\frac{(x-x_0)(x-x_0-h)}{2} f''(\xi(x))\right] \\ &= \frac{f(x_0+h)-f(x_0)}{h} + \frac{2(x-x_0)-h}{2} f''(\xi(x)) \\ &\quad + \frac{(x-x_0)(x-x_0-h)}{2} \mathrm{D}_x(f''(\xi(x))) \end{aligned} \tag{7.52}$$

这个在任意点 x 上逼近$f'(x)$的公式的难点在于没有任何关于函数 $\mathrm{D}_x f''(\xi(x)) = f'''(\xi(x)) \cdot \xi'(x)$的信息，所以无法估计截断误差. 然后，当 x 是 x_0 时，该未知项前面系数是零且公式简化成

$$f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} - \frac{h}{2} f''(\xi) \tag{7.53}$$

对于小的 h 值，差商 $[f(x_0+h)-f(x_0)]/h$ 能够被用来近似$f'(x_0)$，它的截断误差的上界为$(h/2)M$，这里 $|f''(x)| \leqslant M$，对 $x \in [a,b]$. 当 $h>0$ 时，这个公式被称为向前差商公式；当 $h<0$ 时，这个公式被称为向后差商公式.

例 7.10 设函数$f(x)=\ln x$ 和 $x_0=1.8$,试利用式(7.53),分别对 $h=0.1,0.01$ 和 0.001 近似计算在 x_0 处的导数值并估计对应的误差.

解 $f'(x_0)\approx\dfrac{f(1.8+h)-f(1.8)}{h}$, $h>0$,

其对应的截断误差为

$$\frac{|hf''(\xi)|}{2}=\frac{h}{2\xi^2}\leqslant\frac{h}{2(1.8)^2}$$

这里,$1.8<\xi<1.8+h$.具体计算及估计结果见表 7-8.

表 7-8

h	0.1	0.01	0.001
$f'(x_0)$	0.5406722	0.5540180	0.5554013
E	0.0154321	0.0015432	0.0001543

由于 $f'(x)=1/x$,$f'(1.8)$的精确值为 $0.55\dot{5}$,故表 7-8 中的截断误差界的估计值是恰当的.

假设$\{x_0,x_1,\cdots,x_n\}$是 $n+1$ 在某个区间 I 中的互不相同的节点且$f\in C^{n+1}(I)$.应用拉格朗日插值公式及插值余项有

$$f(x)=\sum_{k=0}^{n}f(x_k)\mathscr{L}_k(x)+\frac{(x-x_0)\cdots(x-x_n)}{(n+1)!}f^{(n+1)}(\xi(x))$$

对某个 $\xi(x)\in I$.对上述表达式两边分别求导,则有

$$f'(x)=\sum_{k=0}^{n}f(x_k)\mathscr{L}'_k(x)+\mathrm{D}_k\left[\frac{(x-x_0)\cdots(x-x_n)}{(n+1)!}\right]f^{(n+1)}(\xi(x))$$

$$+\frac{(x-x_0)\cdots(x-x_n)}{(n+1)!}\mathrm{D}_x[f^{n+1}(\xi(x))] \tag{7.54}$$

同样,除非 x 是节点之一,我们在估计截断误差时会遇到问题.故让 $x=x_k$,这时式(7.54)变成

$$f'(x_k)=\sum_{j=0}^{n}f(x_j)\mathscr{L}'_j(x_k)+\frac{f^{(n+1)}(\xi(x))}{(n+1)!}\prod_{\substack{j=0\\j\neq k}}^{n}(x_k-x_j)$$

(7.55)

方程(7.55)被称为 $n+1$ 个节点的近似导数 $f'(x_k)$ 的一个数值微分公式.

一般来说,在方程(7.55)中使用较多的节点会取得较高的精度,虽然函数的求值个数及舍入误差的增长在某种程度上会阻止我们这样做.最常用的是三点和五点的数值微分公式.

我们将首先推导一些有用的三点公式且考虑它们的舍入误差方面的因素.由于

$$\mathscr{L}_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)},\quad \mathscr{L}_0'(x)=\frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)}.$$

同理有

$$\mathscr{L}_1'(x)=\frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)}\text{ 和 }\mathscr{L}_2'(x)=\frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)}.$$

因此,从方程(7.48)可导出

$$\begin{aligned}f'(x_j)=&f(x_0)\frac{2x_j-x_1-x_2}{(x_0-x_1)(x_0-x_2)}+f(x_1)\frac{2x_j-x_0-x_2}{(x_1-x_0)(x_1-x_2)}\\&+f(x_2)\frac{2x_j-x_0-x_1}{(x_2-x_0)(x_2-x_1)}+\frac{1}{6}f^{(3)}(\xi_j)\prod_{\substack{i=0\\i\neq j}}^{2}(x_j-x_i)\end{aligned}\tag{7.56}$$

这里,$j=0,1,2$,ξ_j 依赖于 x_j.

如果节点是等距的,也就是,当 $x_1=x_0+h$,$x_2=x_0+2h$,对某个 $h\neq 0$ 时,从方程(7.56)导出的三个公式变得特别有用.我们将假设这一节余下的部分中的所有节点都是等距的.利用方程(7.49)及 $x_j=x_0$,$x_1=x_0+h$ 和 $x_2=x_0+2h$ 可分别导出

$$f'(x_0)=\frac{1}{h}\left[-\frac{3}{2}f(x_0)+2f(x_1)-\frac{1}{2}f(x_2)\right]+\frac{h^2}{3}f^{(3)}(\xi_0),$$

$$f'(x_1)=\frac{1}{h}\left[-\frac{1}{2}f(x_0)+\frac{1}{2}f(x_2)\right]-\frac{h^2}{6}f^{(3)}(\xi_1),$$

$$f'(x_2)=\frac{1}{h}\left[\frac{1}{2}f(x_0)-2f(x_1)+\frac{3}{2}f(x_2)\right]+\frac{h^2}{3}f^{(3)}(\xi_2).$$

由于 $x_1 = x_0 + h$ 和 $x_2 = x_0 + 2h$，故上述这些公式也可写成

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_0 + h) - \frac{1}{2}f(x_0 + 2h)\right] + \frac{h^2}{3}f^{(3)}(\xi_0), \tag{7.57}$$

$$f'(x_0 + h) = \frac{1}{h}\left[-\frac{1}{2}f(x_0) + \frac{1}{2}f(x_0 + 2h)\right] - \frac{h^2}{6}f^{(3)}(\xi_1), \tag{7.58}$$

$$f'(x_0 + 2h) = \frac{1}{h}\left[\frac{1}{2}f(x_0) - 2f(x_0 + h) + \frac{3}{2}f(x_0 + 2h)\right] + \frac{h^2}{3}f^{(3)}(\xi_2). \tag{7.59}$$

为方便起见，在方程(7.58)中用 x_0 中用 x_0 替换 $x_0 + h$. 同理，在方程(7.59)中用 x_0 替换 $x_0 + 2h$. 这时，上述三公式变成了都是近似 $f'(x_0)$的三个数值微分公式：

$$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3}f^{(3)}(\xi_0), \tag{7.60}$$

$$f'(x_0) = \frac{1}{2h}[-f(x_0 - h) + f(x_0 + h)] - \frac{h^2}{6}f^{(3)}(\xi_1), \tag{7.61}$$

$$f'(x_0) = \frac{1}{2h}[f(x_0 - 2h) - 4f(x_0 - h) + 3f(x_0)] + \frac{h^2}{3}f^{(3)}(\xi_2). \tag{7.62}$$

最后，注意到由于方程(7.62)可通过在方程(7.60)中用 $-h$ 替换 h 得到，故实际上只有两个公式：

$$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3}f^{(3)}(\xi_0), \tag{7.63}$$

这里，ξ_0 落在 x_0 和 $x_0 + 2h$ 之间，和

$$f'(x_0) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)] - \frac{h^2}{6}f^{(3)}(\xi_1), \tag{7.64}$$

这里，ξ_1 落在(x_0-h)和(x_0+h)之间.

方程(7.64)的误差近似地是方程(7.63)的误差的一半.因为在方程(7.64)中的数据关于 x_0 是对称分布的，而在方程(7.63)中的数据都在 x_0 的某一边，所以，这个结果是合理的.方程(7.63)中的逼近在区间 I 的两个端点附近是非常有用的，因为在区间 I 外有关函数 f 的信息不一定能够得到.

在方程(7.63)和方程(7.64)中给出的数值微分法都称为三点公式，尽管在方程(7.64)中，函数值 $f(x_0)$ 并没有出现.类似地，还有一些被称作五点公式的数值微分方法.例如

$$f'(x_0)=\frac{1}{12h}[f(x_0-2h)-8f(x_0-h)+8f(x_0+h)$$

$$-f(x_0+2h)]+\frac{h^4}{30}f^{(5)}(\xi), \tag{7.65}$$

方程(7.65)将在本章 7.7 节中导出.

例 7.11 给定函数 $f(x)=xe^x$ 的函数表 7-9，试利用三点及五点公式求 $f'(2.0)$的近似值，并与精确值进行比较.

表 7-9

x	1.8	1.9	2.0	2.1	2.2
$f(x)$	10.889365	12.703199	14.778112	17.148957	19.855030

解 由于 $f'(x)=(x+1)e^x$，故 $f'(2.0)=22.167168$.利用不同的三点及五点公式去逼近 $f'(2.0)$的结果如下：

三点公式：

利用式(7.56)及 $h=0.1$：

$$\frac{1}{0.2}[-3f(2.0)+4f(2.1)-f(2.2)]=22.032310,$$

利用式(7.56)及 $h=-0.1$：

$$\frac{1}{-0.2}[-3f(2.0)+4f(1.9)-f(1.8)]=22.054525,$$

利用式(7.57)及 $h=0.1$:

$$\frac{1}{0.2}[f(2.1)-f(1.9)]=22.228790,$$

利用式(7.57)及 $h=0.2$:

$$\frac{1}{0.4}[f(2.2)-f(1.8)]=22.414163.$$

五点公式:

利用式(7.58)及 $h=0.1$:

$$\frac{1}{1.2}[f(1.8)-8f(1.9)+8f(2.1)-f(2.2)]=22.166999.$$

上述逼近的误差分别是: 1.35×10^{-1}, 1.13×10^{-1}, -6.16×10^{-2}, -2.47×10^{-1}和 1.69×10^{-4}. 很明显,五点公式给出了最好的结果. 我们也可以注意到利用式(7.57)及 $h=0.1$ 时的误差大约是利用式(7.56)及 $h=\pm0.1$ 时的误差的一半. 这一现象正好与其对应的误差余项的大小相一致.

对于逼近高阶导数的数值微分公式,为简单起见,我们这儿仅给出一个逼近二阶导数的三点公式.

在 x_0 点用三次泰勒多项式展开函数 $f(x)$ 且在 x_0+h 与 x_0-h 这两点上分别求值,我们可得

$$f(x_0+h)=f(x_0)+f'(x_0)h+\frac{1}{2}f''(x_0)h^2+\frac{1}{6}f'''(x_0)h^3+\frac{1}{24}f^{(4)}(\xi_1)h^4, \tag{7.66}$$

$$f(x_0-h)=f(x_0)-f'(x_0)h+\frac{1}{2}f''(x_0)h^2-\frac{1}{6}f'''(x_0)h^3+\frac{1}{24}f^{(4)}(\xi_{-1})h^4, \tag{7.67}$$

这里, $x_0-h<\xi_{-1}<x_0<\xi_1<x_0+h$.

如果我们把方程(7.66)和方程(7.67)相加,可得

$$f(x_0+h)+f(x_0-h)=2f(x_0)+f''(x_0)h^2+\frac{1}{24}[f^{(4)}(\xi_1)+f^{(4)}(\xi_{-1})]h^4 \tag{7.68}$$

即

$$f''(x_0)=\frac{1}{h^2}[f(x_0-h)-2f(x_0)+f(x_0+h)]$$

$$-\frac{h^2}{24}[f^{(4)}(\xi_1)+f^{(4)}(\xi_{-1})]. \tag{7.69}$$

如果$f^{(4)}(x)$在区间$[x_0-h,x_0+h]$上连续,则连续函数的介值定理允许我们把式(7.69)改写成

$$f''(x_0)=\frac{1}{h^2}[f(x_0-h)-2f(x_0)+f(x_0+h)]-\frac{h^2}{12}f^{(4)}(\xi) \tag{7.70}$$

这里,$\xi\in[x_0-h,x_0+h]$.

例7.12 利用在例7.12中给出的关于函数$f(x)=xe^x$的数据,试求$f''(2.0)$的近似值并分析误差.

解 由于$f''(x)=(x+2)e^x$,故$f''(2.0)$的精确值是29.556224.利用式(7.70)及$h=0.1$:

$$f''(2.0)\approx\frac{1}{0.01}[f(1.9)-2f(2.0)+f(2.1)]=29.593200.$$

利用式(7.70)及$h=0.2$:

$$f''(2.0)\approx\frac{1}{0.04}[f(1.8)-2f(2.0)+f(2.2)]=29.704275.$$

其对应的误差分别为-3.70×10^{-2}和-1.48×10^{-1}.

在数值微分中,一个特别重要的问题是舍入误差的影响.让我们来检验方程(7.64):

$$f'(x_0)=\frac{1}{2h}[f(x_0+h)-f(x_0-h)]-\frac{h^2}{6}f^{(3)}(\xi_1).$$

假设,在求$f(x_0+h)$和$f(x_0-h)$值的时候,我们已经碰到了舍入误差$e(x_0+h)$和$e(x_0-h)$;此时,我们被用来计算的值$\tilde{f}(x_0+h)$和$\tilde{f}(x_0-h)$与真值$f(x_0+h)$和$f(x_0-h)$有下述关系:

$$f(x_0+h)=\tilde{f}(x_0+h)+e(x_0+h)$$

和 $$f(x_0-h)=\tilde{f}(x_0-h)+e(x_0-h).$$

在这种情况下,在近似过程中的误差是

$$f'(x_0)-\frac{\tilde{f}(x_0+h)-\tilde{f}(x_0-h)}{2h}=\frac{e(x_0+h)-e(x_0-h)}{2h}-\frac{h^2}{6}f^{(3)}(\xi_1),$$

这里,一部分误差是由舍入误差产生的,另一部分误差是由截断误差产生的.如果我们假设$|e(x_0\pm h)|\leqslant\varepsilon$和$|f^{(3)}(x)|\leqslant M$,则

$$\left|f'(x_0)-\frac{\tilde{f}(x_0+h)-\tilde{f}(x_0-h)}{2h}\right|\leqslant\frac{\varepsilon}{h}+\frac{h^2}{6}M.$$

如果h很小,由舍入误差引起的误差会很大.在实践中,由于舍入误差的影响,使用太小的步长h会徒劳无益的.

例 7.13 函数$f(x)=\sin x$的函数值见表 7-10.在考虑舍入误差影响的基础上,试利用数值微分公式(7.57)及选择适当的步长h求$f'(0.900)$的近似值,并与精确解比较.

表 7-10

x	0.850	0.880	0.890	0.895	0.898	0.899
$\sin x$	0.75128	0.77074	0.77707	0.78021	0.78208	0.78270
x	0.901	0.902	0.905	0.910	0.920	0.950
$\sin x$	0.78395	0.78457	0.78643	0.78950	0.79560	0.81342

解 $$f'(0.900)=\cos(0.900)=0.62161$$

$$f'(0.900)\approx\frac{f(0.900+h)-f(0.900-h)}{2h}$$

总的误差为

$$\left|f(x_0)-\frac{\tilde{f}(x_0+h)-\tilde{f}(x_0-h)}{2h}\right|\leqslant\frac{\varepsilon}{h}+\frac{h^2}{6}M$$

这里,误差界为$e(h)=\frac{\varepsilon}{h}+\frac{h^2}{6}M$,当$h=\sqrt[3]{3\varepsilon/M}$时,$e(h)$达到最小值,其中

$$M=\max_{[0.850,\,0.950]}|f'''(x)|=\max_{[0.850,\,0.950]}|\cos x|\approx 0.65966.$$

因为给定的关于函数 f 的值只保留小数点后面五位，故假定 $\varepsilon = 0.000005$ 是合理的. 所以，步长 h 的最佳选择是

$$h = \sqrt[3]{\frac{3(0.000005)}{0.65966}} \approx 0.02833.$$

按照函数表 7-8，自然最佳的步长选择为 $h_1 = 0.020$，另外我们仍考察 $h_2 = 0.050$ 和 $h_3 = 0.001$ 的情况. 计算结果分别如下：

$$f'(0.900) \approx \frac{f(0.900 + h_1) - f(0.900 - h_1)}{2h_1} = 0.62150,$$

$$f'(0.900) \approx \frac{f(0.900 + h_2) - f(0.900 - h_2)}{2h_2} = 0.62140,$$

和

$$f'(0.900) \approx \frac{f(0.900 + h_3) - f(0.900 - h_3)}{2h_3} = 0.62500.$$

与精确值 $f'(0.900) = 0.62161$ 相比较，$h_1 = 0.020$ 是最佳的选择，而函数表所允许的最小步长选择 $h_3 = 0.001$ 所对应的结果并不理想.

7.7 外推法在数值微分中的应用

我们已在本章 7.3 节中使用外推方法导出了龙贝格算法，一种重要且实用的数值积分方法. 同理，我们可以建立数值微分公式的误差表达式，运用缩小步长的外推算法，精密地求得导数值.

例如，我们运用中心差分公式来计算导数值：

$$f'(x_0) \approx D(h) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} \tag{7.71}$$

下面分析式(7.71)的截断误差. 由泰勒展开式，有

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{1}{2!}h^2 f''(x_0) + \frac{1}{3!}h^3 f'''(x_0) + \frac{1}{4!}h^4 f^{(4)}(x_0) + \cdots$$

$$f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{1}{2!}h^2 f''(x_0)$$

$$-\frac{1}{3!}h^3f'''(x_0)+\frac{1}{4!}h^4f^{(4)}(x_0)+\cdots$$

上述两式相减并整理,有

$$f'(x_0)-D(h)=-\frac{1}{3!}f'''(x_0)\cdot h^2-\frac{1}{5!}f^{(5)}(x_0)\cdot h^4$$

$$-\frac{1}{7!}f^{(7)}(x_0)\cdot h^6-\cdots, \tag{7.72}$$

这里,系数与 h 无关,故误差估计式符合李查逊外推算法,因而有

$$\begin{cases} D_1(h)=D(h) \\ D_{m+1}(h)=\dfrac{D_m\left(\dfrac{h}{2}\right)-\left(\dfrac{1}{2}\right)^{2m}D_m(h)}{1-\left(\dfrac{1}{2}\right)^{2m}},\ m=1,2,\cdots,. \end{cases} \tag{7.73}$$

$D_{m+1}(h)$逼近 $f'(x_0)$的误差为

$$f'(x_0)-D_{m+1}(h)=C_{m+1}^{(m+1)}\cdot h^{2(m+1)}+\cdots. \tag{7.74}$$

式(7.73)的计算过程可列表(见表 7-11),从式(7.73)看,m 越大越好,但这时需要计算 $D(h/2^m)$,而太小的步长 $h/2^m$ 将会引起计算的不稳定.因此,式(7.73)取到一定的 m 后就必须停止计算.

表 7-11

① $D_1(h)$			
② $D_1\left(\frac{h}{2}\right)$	③ $D_2(h)$		
④ $D_1\left(\frac{h}{4}\right)$	⑤ $D_2\left(\frac{h}{2}\right)$	⑥ $D_3(h)$	
⑦ $D_1\left(\frac{h}{8}\right)$	⑧ $D_2\left(\frac{h}{4}\right)$	⑨ $D_3\left(\frac{h}{2}\right)$	⑩ $D_4(h)$

例 7.14 给定函数$f(x)=xe^x$,试利用关于中心差商的外推算法及初始步长 $h=0.2$ 来计算 $f'(2.0)$的值.计算过程中,保留小数点后六位小数.

解 由于 $f'(x)=(x+1)e^x$,故 $f'(2.0)=22.167168$.中心差商公式为 $f'(2.0)\approx D(h)=[f(2.0+h)-f(2.0-h)]/2h$,这里,步长 h 分别取 0.2,0.1 和 0.05.则外推法的计算见表 7-12.

表 7-12

22.414160		
22.228786	22.166995	
22.182564	22.167157	22.167168

通过比较可知,最后的计算结果与精确值完全一致.

下面将利用外推技巧来推导本章 7.6 节中给出的五点公式(7.65).我们可重写式(7.72)为

$$f'(x_0)=\frac{1}{2h}[f(x_0+h)-f(x_0-h)]-\frac{h^2}{6}f'''(x_0)-\frac{h^4}{120}f^{(5)}(\xi_0) \tag{7.75}$$

这里,ξ_0 落于区间$[x_0-h,x_0+h]$中.

在式(7.75)中用 $2h$ 替换 h 可得

$$f'(x_0)=\frac{1}{4h}[f(x_0+2h)-f(x_0-2h)]$$

$$-\frac{4h^2}{6}f'''(x_0)-\frac{16h^4}{120}f^{(5)}(\xi_1), \tag{7.76}$$

这里,ξ_1 落于区间$[x_0-2h,\ x_0+2h]$中.

方程(7.75)乘 4,然后减去方程(7.76)可得

$$3f'(x_0)=\frac{2}{h}[f(x_0+h)-f(x_0-h)]-\frac{1}{4h}[f(x_0+2h)$$

$$-f(x_0-2h)]-\frac{h^4}{30}f^{(5)}(\xi_1)+\frac{2h^4}{15}f^{(5)}(\xi_2) \tag{7.77}$$

如果 $f^{(5)}(x)$ 是在区间 $[x_0-2h,\ x_0+2h]$ 上的连续函数，则能够证明关于 $f^{(5)}(\xi_1)$ 和 $f^{(5)}(\xi_2)$ 的项可被关于 $f^{(5)}(\xi)$ 的项所代替. 利用这个结果及上式两边除以 3 可得五点公式：

$$f'(x_0)=\frac{1}{12h}[f(x_0-2h)-8f(x_0-h)]+8f(x_0+h)$$

$$-f(x_0-2h)]+\frac{h^4}{30}f^{(5)}(\xi).$$

习　题　7

1. 推导下列积分公式：

(1) 矩形公式 $\int_{x_0}^{x_1}f(x)\mathrm{d}x=hf(x_0)+\frac{h^2f'(\xi)}{2}$；

(2) 中点公式 $\int_{x_0}^{x_1}f(x)\mathrm{d}x=hf\left(\frac{x_0+x_1}{2}\right)+\frac{h^3f''(\xi)}{24}$，

其中，$h=x_1-x_0$，$\xi\in(x_0,x_1)$.

2. 分别用复合梯形公式和复合抛物线公式计算下列积分，并比较结果.

(1) $\int_0^1\frac{x}{4+x^2}\mathrm{d}x\quad(n=8)$；(2) $\int_1^9\sqrt{x}\mathrm{d}x\quad(n=4)$.

3. 若 $f''(x)>0$，证明用梯形公式计算积分 $\int_a^b f(x)\mathrm{d}x$ 所得的结果比准确值大，并说明其几何意义.

4. 用代数精度定义直接验证抛物线求积公式具有三次代数精度.

5. 确定下列求积公式中的待定参数，使其代数精度尽量高. 并指出其所具有的代数精度.

(1) $\int_1^2 f(x)\mathrm{d}x\approx\omega_0f(0)+\omega_1f(1)+\omega_2f(2)$；

(2) $\int_{-2h}^{2h} f(x)\mathrm{d}x \approx \omega_{-1}f(-h) + \omega_0 f(0) + \omega_1 f(h)$;

(3) $\int_0^h f(x)\mathrm{d}x \approx \frac{h}{2}[f(0) + f(h)] + ah^2[f'(0) - f'(h)]$.

6. 利用埃尔米特插值公式,导出带有导数值的求积公式:

$$\int_a^b f(x)\mathrm{d}x = \frac{b-a}{2}[f(a) + f(b)]$$

$$-\frac{(b-a)^2}{12}[f'(b) - f'(a)] + \frac{(b-a)^5}{720}f^{(4)}(\eta),$$

其中,$a \leqslant \eta \leqslant b$.

7. 用龙贝格求积算法,计算积分 $I = \int_0^{\pi} \mathrm{e}^x \cos x\mathrm{d}x$.

8. 试用下述算法计算积分 $I = \int_1^3 y^{-1}\mathrm{d}y$.

(1) 利用龙贝格求积公式;

(2) 利用三点及五点的高斯求积公式;

(3) 将积分区间分成四等分,在每一段上用两点高斯求积公式,然后累加得 I 的近似值.

9. 试确定高斯型求积公式

$$\int_{-1}^{1}(1 + x^2) f(x)\mathrm{d}x \approx A_1 f(x_1) + A_2 f(x_2)$$

中的求积系数 A_1, A_2 和节点 x_1, x_2.

10. 对积分区域的 x 方向和 y 方向分别六等分,然后应用复化辛普生公式求下述二重积分的近似值,并与其精确值比较.

(1) $\int_{2.1}^{2.2}\int_{1.3}^{1.4} xy^2\mathrm{d}y\mathrm{d}x$; (2) $\int_{1.3}^{1.5}\int_{-0.1}^{0.1} \sqrt{x}y^2\mathrm{d}y\mathrm{d}x$;

(3) $\int_{-0.1}^{0.1}\int_{0}^{0.1} xy\mathrm{e}^{x^2+y^2}\mathrm{d}x\mathrm{d}y$; (4) $\int_{1.0}^{1.1}\int_{0}^{x} (x^2 + \sqrt{y})\mathrm{d}y\mathrm{d}x$.

11. 利用三点公式近似计算下述表格中各节点上一阶导数的近似值

x	1.1	1.2	1.3	1.4
$f(x)$	0.48603	0.86160	1.59751	3.76155

12. 证明下述近似二阶导数的数值微分公式的截断误差是 $o(h^2)$.

$$f''(x_0) = \frac{1}{h^2}[f(x_0 + h) - 2f(x_0) + f(x_0 - h)].$$

第八章　常微分方程初值问题的数值解

8.1　引　言

科学研究和工程技术中的许多实际问题都需要求解常微分方程或常微分方程组.然而,我们从常微分方程课程中已经知道,除了少数特殊类型的常微分方程能用解析方法求得其精确解外,多数情况要找出解的解析表达式是极其困难的,甚至是不可能的.因此,只能用近似方法求得其数值解.利用计算机解常微分方程主要用数值方法.

本章主要介绍一阶常微分方程的初值问题

$$\begin{cases} \dfrac{dy}{dx} = f(x,y) & (8.1) \\ y(a) = y_0 & (8.2) \end{cases}$$

在区间 $a \leqslant x \leqslant b$ 上的数值解法.

我们在常微分方程课程中已经知道,如果方程(8.1)的右端函数 $f(x,y)$ 在带形区域 $R:\{a \leqslant x \leqslant b, -\infty < y < +\infty\}$ 中为 x,y 的连续函数,并且关于 y 满足李普希兹(Lipschitz)条件:$|f(x,y_1)-f(x,y_2)| \leqslant L|y_1-y_2|$,其中$(x,y_1),(x,y_2) \in R$,$L>0$ 为李普希兹常数,则理论上就可保证初值问题式(8.1)~(8.2)的解 $y=y(x)$ 在$[a,b]$上存在且唯一.今后如无特殊说明,我们总假定方程式(8.1)~(8.2)的解 $y(x)$ 存在、唯一且足够光滑,此外,还假定方程本身是稳定的,即精确解 $y(x)$ 连续依赖于初始值及右端函数.

所谓初值问题的数值解法,就是能算出精确解 $y(x)$ 在自变量 x 的一系列离散节点 $a=x_0<x_1<x_2<\cdots<x_{n-1}<x_n<\cdots$ 上

的近似值 $y_1, y_2, \cdots, y_{n-1}, y_n, \cdots$ 的方法. 这里, 把 $y_k (k = 1, 2, \cdots)$ 叫做初值问题在点列 x_k 上的数值解. 也即数值解法的基本思想是通过某种离散化方法, 将微分方程转化为差分方程(代数方程) 来求解.

最常用的离散方法是数值积分方法和泰勒展开方法.

相邻两个节点间的距离 $h = x_{k+1} - x_k$ 称作步长, 为了讨论方便, 下面如不加说明, 一般都把 h 取为定步长, 这时, 节点可表为 $x_k = x_0 + kh (k = 0, 1, 2, \cdots)$. 初值问题式(8.1)、(8.2) 的数值解法有个基本特点, 它们都是采取"步进式", 即求解过程可顺着节点排列的次序一步一步地向前推进, 描述这类算法只要给出已知结果 $y_n, y_{n-1}, y_{n-2}, \cdots$, 计算 y_{n+1} 的递推公式.

本章讨论常微分方程初值问题的几种常用的数值解法, 如尤拉(Euler) 方法, 龙格 - 库塔(Runge-Kutta) 方法及一些常用的线性多步法, 重点是怎样构造这些数值方法, 并引进一些重要的概念(如收敛性、稳定性等) 及与之相关的理论问题.

8.2 尤拉(Euler) 方法及其改进

一、尤拉公式

尤拉方法是解初值问题式(8.1)、式(8.2) 的最简单、最古老的一种数值解法. 因为它不够精确, 在实际计算时很少被采用, 但由于它的公式简单, 又具有明显的几何意义, 且在某种程度上反映了数值方法的基本思路, 因此, 我们还是从介绍尤拉方法开始, 同时引进一些常用的术语和概念.

我们看到式(8.1), 方程 $\dfrac{dy}{dx} = f(x, y)$ 中含有导数项 $y'(x)$, 这是微分方程的本质特征, 也是它难以求解的困难所在. 数值解法的关键, 在于设法消去其导数项, 这一工作常称为"离散化". 例如由于差商是微商的近似运算, 因此可用来实现其离散化.

用差商代替式(8.1)中的微商,即 $y'(x) \approx \dfrac{y(x_{n+1}) - y(x_n)}{x_{n+1} - x_n}$,并以 y_n, y_{n+1} 分别代替 $y(x_n)$ 及 $y(x_{n+1})$,就得到

$$y_{n+1} = y_n + hf(x_n, y_n) \quad (n = 0,1,2,\cdots). \tag{8.3}$$

式(8.3)就是著名的尤拉公式,它是一个显式的差分方程,很容易求解,只需取定初值 y_0,依式(8.3)就可以递推地求出初值问题式(8.1)和式(8.2)在节点 $x_1, x_2, \cdots$ 处的数值解 $y_1, y_2, \cdots$.

尤拉方法的几何意义是十分清楚的,它是用一条自点 $P_0(x_0, y_0)$ 出发的折线 $\overline{P_0P_1\cdots P_n}$ 去逼近积分曲线 $y = f(x)$,如图 8-1 所示.所以尤拉方法又称为尤拉折线法.显然,尤拉折线法简单地取折线的端点 P_n 作为数值解,精度非常差.

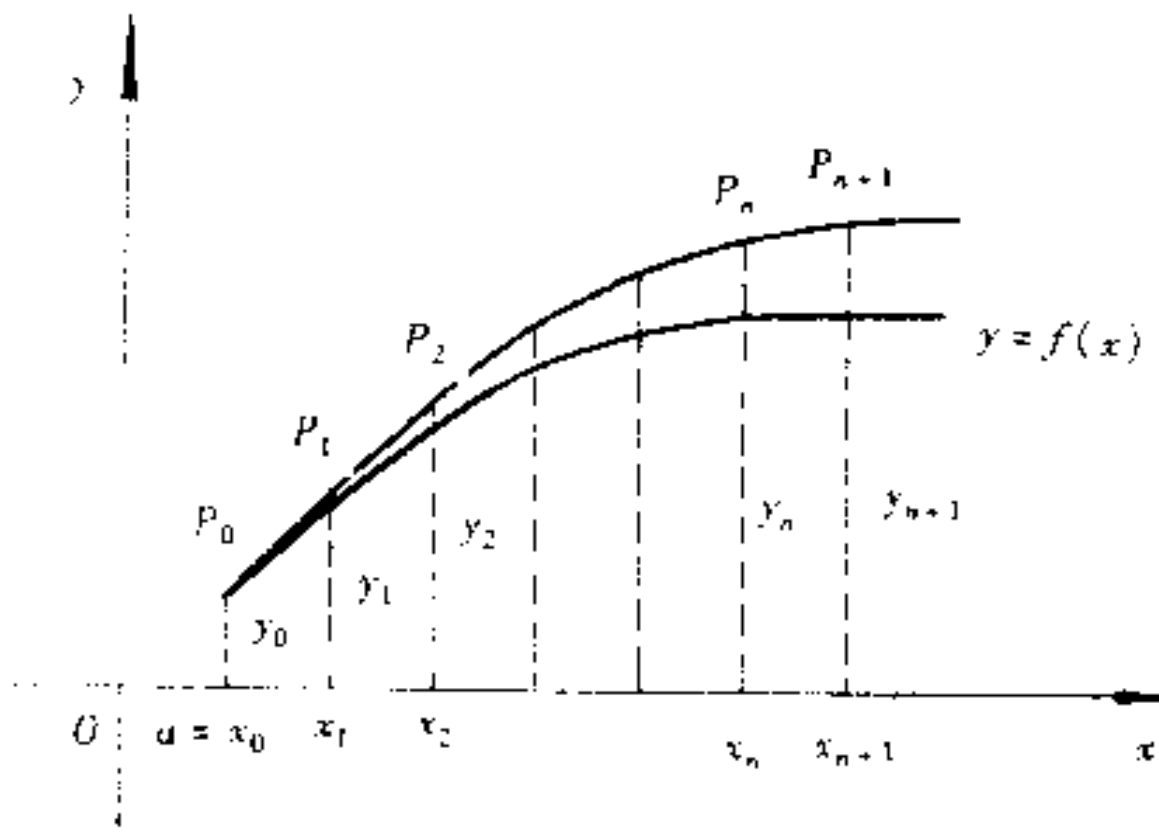


图 8-1

二、梯形公式

为了改进尤拉方法的精度,我们从另一角度来考察初值问题式(8.1)和式(8.2).事实上,方程式(8.1)可改写为 $\mathrm{d}y(x) = \mathrm{f}(x, y)\mathrm{d}x$,两边对 x 作从 x_n 到 x_{n+1} 的积分得 $y(x_{n+1}) = y(x_n) +$

$\int_{x_n}^{x_{n+1}} f(x, y(x))\mathrm{d}x$，这里 $y(x)$ 是初值问题式(8.1)和式(8.2)的精确解.

要得到 $y(x_{n+1})$ 的值，就必须计算右端的积分 $\int_{x_n}^{x_{n+1}} f(x, y(x))\mathrm{d}x$，但积分中含有未知函数 $y(x)$，无法直接计算，若我们采用数值积分方法计算积分 $\int_{x_n}^{x_{n+1}} f(x, y(x))\mathrm{d}x$ 的近似值，就可获得 $y(x_{n+1})$ 的近似值. 这种将初值问题式(8.1)和式(8.2)离散化的方法就称为求初值问题的数值积分方法. 当我们选用不同的数值积分公式时，就会得到不同的计算公式. 例如，用最简单的左矩形公式计算积分得 $\int_{x_n}^{x_{n+1}} f(x, y(x))\mathrm{d}x \approx hf(x_n, y(x_n))$，于是就得到 $y(x)$ 在 $x = x_{n+1}$ 处的近似值 $y(x_{n+1}) \approx y(x_n) + hf(x_n, y(x_n))$，此式中，如用 y_n 代替右端的 $y(x_n)$，并将算出的右端值作为 $y(x_{n+1})$ 的近似值 y_{n+1}，这样建立起来的计算公式，也就是前述的尤拉公式：$y_{n+1} = y_n + hf(x_n, y_n)$，$(n = 0, 1, 2, \cdots)$.

因矩形求积公式的精度较低，因此所导出的尤拉公式的精度也较低. 为了提高精度，可改用梯形公式在 $[x_n, x_{n+1}]$ 上计算右端的积分得：

$$\int_{x_n}^{x_{n+1}} f(x, y(x))\mathrm{d}x \approx \frac{h}{2}[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))],$$

于是得：

$$y(x_{n+1}) \approx y(x_n) + \frac{h}{2}[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))],$$

以 y_n, y_{n+1} 分别代替 $y(x_n), y(x_{n+1})$，则得差分方程：

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \quad (n = 0, 1, 2, \cdots) \tag{8.4}$$

式(8.4)称为梯形公式，由于数值积分的梯形公式比矩形公式的

精度高，因此，梯形公式(8.4)的精度比尤拉公式高.

必须指出，尤拉公式与梯形公式在计算上有明显的区别，尤拉公式的特点是可以由 y_n 直接计算 y_{n+1}，是一个直接计算的公式，通常称这类公式为显式；而梯形公式中因它的右端也含有未知的 y_{n+1}，故不能直接得到 y_{n+1}，必须通过解方程才能求得 y_{n+1}，这类公式称为隐式. 显式和隐式两类公式各有特点，考虑到数值稳定性等因素，人们有时要用隐式公式，但从算法的角度看，显式则远比隐式方便. 隐式方程通常用迭代法求解.

三、截断误差

一般说，不论用什么数值方法，即使在初始值是准确的且计算过程没有舍入误差的情况下，所得的数值解往往并不与初值问题的准确解完全符合，记 $e_n = y(x_n) - y_n$ 表示数值方法在节点 x_n 处的截断误差，因为它纯碎是由于微分方程的离散化产生的，所以又叫做离散误差或方法误差.

人们常以泰勒展开为工具分析计算公式的误差. 为简化分析，有时假定 y_n 为准确，即在 $y_n = y(x_n)$ 的前提下估计误差 $y(x_{n+1}) - y_{n+1}$，这种误差称为局部截断误差. 如图 8-2 所示.

在图8-2上点 P_n 落在积分曲线 $y = y(x)$ 上，计算由 x_n 到 x_{n+1} 这一步方法引起的误差 ε_{n+1} 为局部截断误差. 为了便于区别，又将 e_n 称为整体误差. 在以后，我们将会看到，局部截断误差与整体截断误差有着密切的联系，也在一定程度上刻画了数值方法的精度.

定义 8.1 若某种微分方程数值解公式的局部截断误差为 $\varepsilon_{n+1} = o(h^{p+1})$，则称这种方法具有 p 阶精度或称为 p 阶方法. 这里，p 为正整数，通常 p 越大，h 越小，则截断误差越小，数值方法越精确.

例如，可以证明：尤拉方法是一阶方法.

设 $y_n = y(x_n)$，把 $y(x_{n+1})$ 在 x_n 处展开成泰勒级数，即

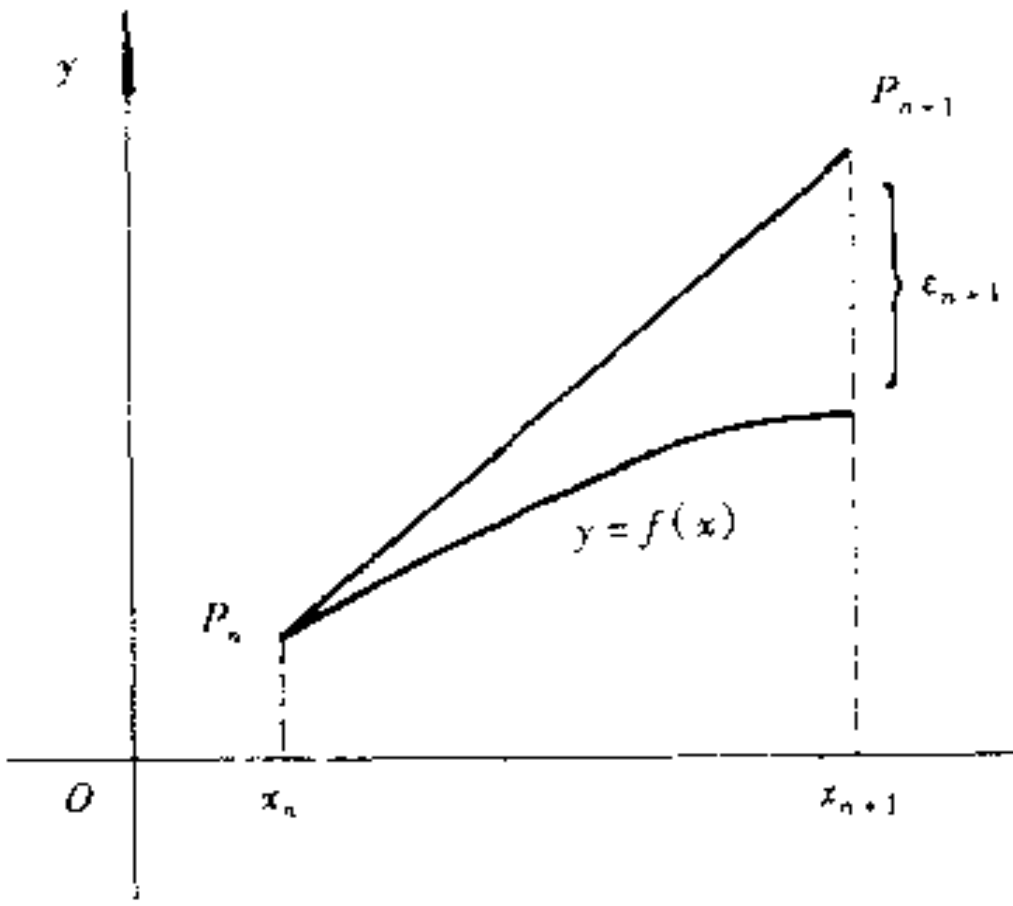


图 8-2

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(\xi), \quad (x_n < \xi < x_{n+1}),$$

用尤拉公式有

$$y_{n+1} = y_n + hf(x_n, y_n)$$
$$= y(x_n) + hf(x_n, y(x_n)) = y(x_n) + hy'(x_n)$$

两式相减，得尤拉公式的局部截断误差为

$$T_1 = y(x_{n+1}) - y_{n+1} = \frac{h^2}{2}y''(\xi),$$

假定 $y(x)$ 在其存在区间$[a,b]$上充分光滑，并令 $M = \max_{a\leqslant x\leqslant b} |y''(x)|$，

则 $|T_1| \leqslant M \cdot \frac{h^2}{2} = O(h^2)$，即尤拉方法的局部截断误差是 h^2 的同阶无穷小．所以尤拉方法具有一阶精度或称它是一阶方法．

也可以证明梯形公式(8.4)是二阶方法，比尤拉方法高一阶．

四、预估－校正法

我们看到，梯形方法相对于尤拉方法虽然精度提高了，但是每一步都要用迭代法解方程，每迭代一次，都要重新计算函数 $f(x)$

的值，而迭代又要反复进行若干次，计算量很大．且事先难以估计迭代次数．实际计算时，一种有效措施是采用所谓预估－校正法．当 h 较小时，迭代一次或二次就可转入下一步的计算，这就简化了算法．

例如，

$$\begin{cases} \text{预估} & \bar{y}_{n+1} = y_n + hf(x_n, y_n) \\ \text{校正} & y_{n+1} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})] \end{cases} \tag{8.5}$$

就是一个预估－校正格式．具体地说，我们先用尤拉公式求得一个初步的近似值 $\bar{y}_{n+1}$，称之为预估值，预估值 $\bar{y}_{n+1}$ 的精度可能很差，我们再用梯形公式将它校正一次，即按式(8.5)迭代一次得 y_{n+1}，这个结果称校正值，作为初值问题式(8.1)与式(8.2)在节点 x_{n+1} 处的数值解．而这样建立的预估－校正格式(8.5)也称为改进的尤拉公式．

我们再给出一个预估－校正格式：

$$\begin{cases} \text{预估} & \bar{y}_{n+1} = y_{n-1} + 2hf(x_n, y_n) \\ \text{校正} & y_{n+1} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})] \end{cases} \tag{8.6}$$

式(8.6)中的预估公式

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n) \tag{8.7}$$

很容易用数值积分的中矩形公式推得，它的精度比尤拉公式高．

对比尤拉公式，梯形公式及式(8.7)，我们发现，前面两个公式的共同点是，欲计算 y_{n+1}，只用到前一步的 y_n，这样的方法称为单步法．单步法的优点是可以“自开始”的，即只要已知初值 y_0，就可以按计算公式顺次求出数值解 $y_1, y_2, \cdots$．而后者则不然，欲计算 y_{n+1}，需要用到前两步的 y_{n-1} 及 y_n，我们称这样的方法为两步法．两步法乃至更多步的方法也称多步法．多步法的计算需要用单步法提供足够的初始值，然后才能按计算公式计算其数值解．例如一个 k 步算法，除了初值条件 y_0 外，必须由其他方法为它提供 y_1，

$y_2, \cdots, y_{k-1}$ 后才能开始运算，这些 $y_0, y_1, \cdots, y_{k-1}$ 通常称为“表头”，一般来讲，多步法具有较高的精度，当“表头”问题解决之后，并不增加多少计算函数值的工作量就能得到较高精度的近似解，因此是一种具有较高计算效率的算法.

8.3 龙格 - 库塔(Runge-Kutta) 方法

本节介绍求解初值问题式(8.1) 与式(8.2) 的一类高精度的单步法，即龙格 - 库塔方法. 由于这类方法与泰勒展开有密切联系，所以下面先简单地介绍一下泰勒展开方法.

一、泰勒展开方法

利用泰勒展开的方法也是微分方程离散化的一个重要途径.

设初值问题
$$\begin{cases} \dfrac{dy}{dx} = f(x, y) & (8.1) \\ y(a) = y_0 & (8.2) \end{cases}$$
在 $a \leqslant x \leqslant b$ 上有解 $y = y(x)$，将 $y(x_{n+1})$ 在 $x = x_n$ 处作泰勒展开有

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + \cdots + \frac{h^p}{p!}y^{(p)}(x_n) + \cdots.$$

注意到 $y' = f(x, y)$，式(8.1) 中 $y(x)$ 的各阶导数可以用函数 f 来表示，我们引进函数序列 $f^{(i)}(x, y)$，$(i = 0, 1, 2, \cdots)$ 来描述求导过程，则有

$$y(x_{n+1}) = y(x_n) + hf^{(0)}(x_n, y(x_n)) + \frac{h^2}{2!}f^{(1)}(x_n, y(x_n)) + \cdots + \frac{h^p}{p!}f^{(p-1)}(x_n, y(x_n)), \quad (8.8)$$

其中 $f^{(0)}(x_n, y(x_n)) = f(x_n, y(x_n)) = y'(x_n)$,

$$f^{(1)}(x_n, y(x_n)) = f_x(x_n, y(x_n)) + f_y(x_n, y(x_n)) \cdot f(x_n, y(x_n))$$
$$= y''(x_n),$$
$$f^{(2)}(x_n, y(x_n)) = \frac{\partial f^{(1)}}{\partial x} + f\frac{\partial f^{(1)}}{\partial y} = y'''(x_n)$$
$$\cdots$$

一般地有

$$f^{(j-1)}(x_n, y(x_n)) = \frac{\partial f^{(j-2)}}{\partial x} + f\frac{\partial f^{(j-2)}}{\partial y} = y^{(j)}(x_n), (j = 2,3,\cdots)$$

在展开式(8.8)的右端截取若干项，例如一般截取 $p+1$ 项，并以 y_n, y_{n+1} 分别代替 $y(x_n), y(x_{n-1})$，则得下列泰勒公式：

$$y_{n+1} = y_n + hf^{(0)}(x_n, y_n) + \frac{h^2}{2!}f^{(1)}(x_n, y_n)$$
$$+ \cdots + \frac{h^p}{p!}f^{(p-1)}(x_n, y_n), \tag{8.9}$$

当 $p = 1$ 时，有尤拉公式为

$$y_{n+1} = y_n + hf(x_n, y_n),$$

$p = 2$ 时，为

$$y_{n+1} = y_n + hf(x_n, y_n)$$
$$+ \frac{h^2}{2}[f_x(x_n, y_n) + f_y(x_n, y_n) \cdot f(x_n, y_n)]$$

显然，式(8.9)的局部截断误差为 $o(h^{p+1})$，它是由泰勒展开式得到的求解初值问题式(8.1)与式(8.2)的 p 阶方法，而提高泰勒公式的阶 p，即可提高计算的精度，从理论上讲，只要 $y(x)$ 充分光滑，泰勒展开方法可以构造任意有限阶的计算公式. 但事实上，具体构造这种公式往往是相当困难的. 因为复合函数 $f(x, y(x))$ 的高阶导数常常是很繁琐的，因此泰勒展开方法一般不直接使用.

例 8.1 取步长 $h = 0.1$，应用二阶及四阶的泰勒展开格式求解初值问题

$$\begin{cases} \dfrac{dy}{dx} = -y(x) + x + 1 \\ y(0) = 1 \end{cases}, \quad (0 \leqslant x \leqslant 1).$$

解　首先函数$f(x,y(x)) = -y(x) + x + 1$的一阶、二阶及三阶导数:$\dfrac{\mathrm{d}}{\mathrm{d}x}f(x,y) = y - x$,

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}f(x,y) = -y + x,$$

$$\frac{\mathrm{d}^3}{\mathrm{d}x^3}f(x,y) = y - x.$$

故可分别写出二阶及四阶泰勒格式

$$\begin{cases} y_0 = 1 \\ y_{n+1} = y_n + h\left[\left(1 - \dfrac{h}{2}\right)(x_n - y_n) + 1\right] \end{cases}$$

及

$$\begin{cases} y_0 = 1 \\ y_{n+1} = y_n + h\left[\left(1 - \dfrac{h}{2} + \dfrac{h^2}{6} - \dfrac{h^3}{24}\right)(x_n - y_n) + 1\right] \end{cases}$$

具体计算格式分别为

$$\begin{cases} y_0 = 1 \\ y_{n+1} = 0.905y_n + 0.0095n + 0.1, n = 0,1,\cdots,9. \end{cases}$$

及

$$\begin{cases} y_0 = 1 \\ y_{n+1} = 0.9048375y_n + 0.00951625n + 0.1, n = 0,1,\cdots,9. \end{cases}$$

计算结果为

x_i	0.1	0.2	0.3	0.4	0.5
二阶	1.005000	1.041218	1.041218	1.070802	1.107076
四阶	1.0048375000	1.0408184220	1.0408184220	1.0703202889	1.1065309344

x_i	0.6	0.7	0.8	0.9	1.0
二阶	1.149404	1.197211	1.249976	1.307228	1.368541
四阶	1.1488119344	1.1965856187	1.2493292897	1.3065699912	1.3678797744

由于原方程的精确解为 $y(x) = e^{-x} + x$. 通过比较,二阶方法求出的数值解的精度为 10^{-4} 阶的,而四阶方法求出的数值解的精度为 10^{-7} 阶的.

二、龙格 - 库塔方法的基本思想

由于一般不宜直接使用泰勒展开法,但我们可设法间接使用泰勒展开方法,以求得精度较高的数值方法. 下面我们介绍一种应用较广泛的高精度的单步法 —— 龙格 - 库塔方法,简记为 R-K 方法. 为了导出 R-K 方法,我们先对前面介绍的尤拉公式和预估 - 校正格式作进一步分析.

尤拉公式可改写成 $\begin{cases} y_{n+1} = y_n + k_1 \\ k_1 = hf(x_n, y_n) \end{cases}$,用它计算 y_{n+1},需计算一次 $f(x, y)$ 的值,若设 $y(x_n) = y_n$,则 y_{n+1} 的表达式与 $y(x_{n+1})$ 的泰勒展开式的前两项完全相同,即局部截断误差为 $O(h^2)$.

而预估 - 校正公式

$$\begin{cases} \bar{y}_{n+1} = y_n + hf(x_n, y_n), \\ y_{n+1} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})], \end{cases}$$

可改写为

$$\begin{cases} y_{n+1} = y_n + \dfrac{k_1}{2} + \dfrac{k_2}{2}, \\ k_1 = hf(x_n, y_n), \\ k_2 = hf(x_n + h, y_n + k_1). \end{cases}$$

用它计算 y_{n+1} 需计算两次 $f(x, y)$ 的值. 若设 $y(x_n) = y_n$,则 y_{n+1} 与 $y(x_{n+1})$ 的泰勒展开式的前三项完全相同,即局部截断误差为 $O(h^3)$.

上述两组公式在形式上有一个共同点:都是用 $f(x, y)$ 在某些点上的值的线性组合得出 $y(x_{n+1})$ 的近似值 y_{n+1},而且可以看出,

增加计算 $f(x,y)$ 的次数,可提高截断误差的阶.

类似地,龙格 - 库塔方法的基本思想也是设法计算 $f(x,y)$ 在某些点上的函数值,然后对这些函数值作线性组合,构造近似计算公式,再把近似公式和精确解的泰勒展开式相比较,使前面的若干项吻合,从而获得达到一定精度的数值计算公式.

具体构造时,先引进若干参数,例如一般的显示 R-K 方法的形式为

$$y_{n+1} = y_n + \sum_{i=1}^{r} \omega_i k_i \tag{8.10}$$

其中 $k_1 = hf(x_n, y_n)$,

$$k_i = hf\left(x_n + \alpha_i h, y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j\right) \quad (i = 2,3,\cdots,r)$$

其中参数 $\omega_i, \alpha_i, \beta_{ij}$ 是既与方程(8.1)的右端 $f(x,y)$ 无关,又与步数 n 无关的常数,选择它们的原则是要求式(8.10)的右端在 (x_n, y_n) 处作泰勒展开后,按 h 的幂次作升幂排列重新整理得式

$$y_{n+1} = y_n + r_1 h + \frac{1}{2!} r_2 h^2 + \frac{1}{3!} r_3 h^3 + \cdots \tag{8.11}$$

再与微分方程(8.1)的精确解 $y(x)$ 在点 $x = x_n$ 处的泰勒展开式

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{1}{2!} h^2 y''(x_n)$$

$$+ \cdots + \frac{h^p}{p!} y^{(p)}(x_n) + O(h^{p+1})$$

相比较,使其有尽可能多的项重合,例如要求

$$r_1 = f_n, \quad r_2 = f'_n, \quad r_3 = f''_n, \cdots, \quad r_p = f_n^{(p-1)}$$

就得到 p 个方程,从而定出参数 $\omega_i, \alpha_i, \beta_{ij}$,再代入 $k_1, k_2, \cdots, k_r$ 的表达式,就可得到计算微分方程初值问题式(8.1)与式(8.2)的数值计算公式

$$y_{n+1} = y_n + \sum_{i=1}^{r} \omega_i k_i \tag{8.11'}$$

式(8.11)称为 r 段的 R-K 方法的计算公式.

若式(8.11)与 $y(x_{n+1})$ 的泰勒展开式的前 $p+1$ 项完全一致，即，使局部截断误差达到 $O(h^{p+1})$，则称式(8.10)为 p 阶 r 段的 R-K 方法.

通常用得较多的是 r 取 2,3,4，这时，为了计算方便，可适当选取 $\alpha_i, \beta_{ij}, \omega_i$，使 $p=r$(即阶数 = 段数). 这样，问题就归结为怎样选取参数 $\omega_i, \alpha_i, \beta_{ij}$，使式(8.11)成为所需精度的计算方法.

三、龙格-库塔公式的推导

现在我们就按上述介绍的龙格-库塔方法的基本思想，具体地推导一些常用的龙格-库塔公式.

由于选取 $\omega_i, \alpha_i, \beta_{ij}$ 的推导涉及复合函数求导和二元函数的泰勒展开. 显然，公式的段数 r 越大，推导越繁琐，但基本思想方法是一样的. 下面我们仅以二阶二段的 R-K 方法为例，说明推导过程.

当 $r=2$ 时，龙格-库塔格式为

$$\begin{cases} y_{n+1} = y_n + \omega_1 k_1 + \omega_2 k_2 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \alpha_2 h, y_n + \beta_{21} k_1) \end{cases} \tag{8.12}$$

适当选取参数 $\omega_1, \omega_2, \alpha_2, \beta_{21}$ 的值，使得在 $y(x_n) = y_n$ 的假设下，局部截断误差 $y(x_{n+1}) - y_{n+1}$ 的阶尽可能高. 若要使 $y(x_{n+1}) - y_{n+1} = O(h^3)$，可将 $y_{n+1} = y_n + \omega_1 k_1 + \omega_2 k_2$ 在 (x_n, y_n) 处作泰勒展开，为此，把 k_1, k_2 的表达式代入式(8.12)得

$$y_{n+1} = y_n + \omega_1 hf(x_n, y_n) + \omega_2 hf(x_n + \alpha_2 h, y_n + \beta_{21} hf_n)$$

显然，上式中前二项不必展开，且记 $f(x_n, y_n) = f_n$，只要把 $f(x_n + \alpha_2 h, y_n + \beta_{21} hf_n)$ 在 (x_n, y_n) 处展开即可，也即

$$f(x_n + \alpha_2 h, y_n + \beta_{21} hf_n) = f(x_n, y_n) + \alpha_2 h \frac{\partial f}{\partial x} + \beta_{21} h \frac{\partial f}{\partial y} f_n + O(h^2),$$

其中$\frac{\partial f}{\partial x}$,$\frac{\partial f}{\partial y}$都是在$x_n$,$y_n$处取值,记为$\frac{\partial f_n}{\partial x}$,$\frac{\partial f_n}{\partial y}$,从而

$$y_{n+1} = y_n + \omega_1 k_1 + \omega_2 k_2$$

$$= y_n + \omega_1 h f_n + \omega_2 \left[h f_n + h^2 \left(\alpha_2 \frac{\partial f_n}{\partial x} + \beta_{21} \frac{\partial f_n}{\partial y} f_n \right) \right] + O(h^3),$$

再按 h 的幂次作升幂排列整理得

$$y_{n+1} = y_n + (\omega_1 f_n + \omega_2 f_n) h$$

$$+ \omega_2 \left(\alpha_2 \frac{\partial f_n}{\partial x} + \beta_{21} \frac{\partial f_n}{\partial y} f_n \right) h^2 + O(h^3),$$

与微分方程$\frac{dy}{dx} = f(x, y)$的精确解 $y = y(x)$在点 $x = x_n$ 处的泰勒展开式

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + y'(x_n) h + \frac{1}{2!} y''(x_n) h^2 + O(h^3)$$

$$= y_n + f_n h + \frac{1}{2} \left(\frac{\partial f_n}{\partial x} + \frac{\partial f_n}{\partial y} f_n \right) h^2 + O(h^3)$$

逐项进行比较,为使 $y(x_{n+1}) - y_{n+1} = O(h^3)$,则令 h,h^2 项的系数相同,便得到三个等式

$$\begin{cases} \omega_1 + \omega_2 = 1 \\ \omega_2 \alpha_2 = \frac{1}{2} \\ \omega_2 \beta_{21} = \frac{1}{2} \end{cases} \tag{8.13}$$

由这三个等式中确定的 $\omega_1, \omega_2, \alpha_2, \beta_{21}$ 代入 R-K 计算式(8.12)中,即得二阶二段的 R-K 方法的计算格式. 由于上述方程组式(8.13)是关于 4 个未知元的三个方程,因而有一个未知元可作为自由参数,这表明式(8.13)有无穷多组解,而且每一组解所构成的两段 R-K 方法的阶数都是是 2,所以都叫做二阶二段的 R-K 公式,因而 R-K 公式是一类公式,每特殊确定一组系数,就得到一个特殊的 R-K 公式.

例如,若取 $\alpha_2 = 1$,则从式(8.13)解得 $\omega_1 = \omega_2 = \frac{1}{2}$,$\beta_{21} = 1$,则

得

$$\begin{cases} y_{n+1} = y_n + \dfrac{1}{2}k_1 + \dfrac{1}{2}k_2 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + h, y_n + k_1) \end{cases}$$

这就是前面的预估-校正式(8.5).

若取 $\omega_1 = 0$,则 $\omega_2 = 1, \alpha_2 = \dfrac{1}{2}, \beta_{21} = \dfrac{1}{2}$,则得另一个二阶二段的 R-K 计算公式为

$$\begin{cases} y_{n+1} = y_n + k_2 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf\left(x_n + \dfrac{h}{2}, y_n + \dfrac{1}{2}k_1\right) \end{cases}$$

此即中点方法.

显然,有许多可能取法,相应的计算公式的局部截断误差都是 $O(h^3)$,并由此可见,上节所讨论的尤拉方法及其各种改进也可看成龙格-库塔方法的特殊情况.

可能有人会问,在上述二阶 R-K 计算格式中,若计算函数值的次数不增加,能否有其他选择参数的办法,使局部截断误差的阶再提高呢? 譬如,能否适当选择这四个参数,使近似公式的局部截断误差的阶达到 $O(h^4)$呢?

事实上,因为在 $y(x_{n+1})$的展开式中的 $f_x \cdot f_y + f_y^2 \cdot f$ 是不能通过选择 $\alpha_2, \beta_{21}, \omega_1, \omega_2$ 来消掉,所以不论四个参数如何选择都不能使截断误差的阶达到 $O(h^4)$,这说明在计算两次函数值的情况下,局部截断误差的阶最高是 $O(h^3)$,要再进一步提高方法的阶,就必须通过继续增加计算函数值的次数的途径来取得.

高阶龙格-库塔公式的推导方法与上面讨论的二阶 R-K 方法推导类似,只是随着阶数的增高,推导的工作量也随之增大.

例如,在 R-K 方法的一般式中,取 $r = 3$,则有

$$\begin{cases} y_{n+1} = y_n + \omega_1 k_1 + \omega_2 k_2 + \omega_3 k_3 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \alpha_2 h, y_n + \beta_{21} k_1) \\ k_3 = hf(x_n + \alpha_3 h, y_n + \beta_{31} k_1 + \beta_{32} k_2) \end{cases} \tag{8.14}$$

仿照二阶情况的推导，只要把 k_1, k_2, k_3 代入 y_{n+1} 的表达式中，在 (x_n, y_n) 处作泰勒展开，再与 $y(x_{n+1})$ 在点 x_n 处的泰勒展开式比较，使公式的局部截断误差 $y(x_{n+1}) - y_{n+1} = O(h^4)$，可得含有八个参数 $\omega_1, \omega_2, \omega_3, \alpha_2, \alpha_3, \beta_{21}, \beta_{31}, \beta_{32}$ 的六个方程组成的方程组：

$$\begin{cases} \omega_1 + \omega_2 + \omega_3 = 1 \\ \alpha_2 = \beta_{21} \\ \alpha_3 = \beta_{31} + \beta_{32} \\ \omega_2 \alpha_2 + \omega_3 \alpha_3 = \dfrac{1}{2} \\ \omega_2 \alpha_2^2 + \omega_3 \alpha_3^2 = \dfrac{1}{3} \\ \omega_2 \beta_{32} \alpha_2 = \dfrac{1}{6} \end{cases} \tag{8.15}$$

显然，有无穷多组解能满足方程组(8.15)，而且每组解所构成的三段 R-K 方法的阶数都是 3，例如很容易验证，当取

$$\omega_1 = \omega_3 = \frac{1}{6}, \omega_2 = \frac{4}{6}, \alpha_2 = \beta_{21} = \frac{1}{2}, \alpha_3 = 1, \beta_{31} = -1, \beta_{32} = 2$$

时，确实满足方程组(8.15)，从而得到一个重要的三阶 R-K 公式：

$$\begin{cases} y_{n+1} = y_n + \dfrac{1}{6}(k_1 + 4k_2 + k_3) \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{2}k_1) \\ k_3 = hf(x_n + h, y_n - k_1 + 2k_2) \end{cases} \tag{8.16}$$

式(8.16)称为库塔公式.

在实际应用中,最常用的是四阶龙格-库塔公式,但推导十分麻烦,要涉及到求解含有 13 个参数的 11 个方程组成的方程组,我们这里不加推导,仅给出两个常用的四阶 R-K 公式.

1. 标准四阶龙格-库塔公式

$$\begin{cases} y_{n+1} = y_n + \dfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = hf(x_n, y_n) \\ k_2 = hf\left(x_n + \dfrac{h}{2}, y_n + \dfrac{k_1}{2}\right) \\ k_3 = hf\left(x_n + \dfrac{h}{2}, y_n + \dfrac{k_2}{2}\right) \\ k_4 = hf(x_n + h, y_n + k_3) \end{cases} \tag{8.17}$$

2. 吉尔(Gill)公式

$$\begin{cases} y_{n+1} = y_n + \dfrac{1}{6}[k_1 + (2-\sqrt{2})k_2 + (2+\sqrt{2})k_3 + k_4] \\ k_1 = hf(x_n, y_n) \\ k_2 = hf\left(x_n + \dfrac{h}{2}, y_n + \dfrac{k_1}{2}\right) \\ k_3 = hf\left(x_n + \dfrac{h}{2}, y_n + \dfrac{\sqrt{2}-1}{2}k_1 + \dfrac{2-\sqrt{2}}{2}k_2\right) \\ k_4 = hf\left(x_n + h, y_n - \dfrac{\sqrt{2}}{2}k_2 + \dfrac{2+\sqrt{2}}{2}k_3\right) \end{cases} \tag{8.18}$$

其局部截断误差均为 $O(h^5)$.

从理论上说,我们可构造任意高阶的计算方法,但须注意精度的阶数与计算函数值的次数之间的关系并非等量增加的,它们间的关系如下表:

计算函数值次数 r	1	2	3	4	5	6	7	$r \geqslant 8$
精度的阶数	1	2	3	4	4	5	6	$r-2$

显然可见,四阶及四阶以下的龙格-库塔方法,每一步需调用

函数 $f(x,y)$的次数与阶数一致，如二阶公式需调用两次$f(x,y)$，而四阶公式需调用四次 $f(x,y)$，但对于更高阶的情况则不然，需要调用 $f(x,y)$的次数，远远大于方法的阶数，由于计算量较大，从而很少使用更高阶的 R-K 方法. 事实上，对于大量的实际问题，四阶的 R-K 方法已可满足对精度的要求.

例 8.2 取步长 $h=0.2$，从 $x=0$ 直到 $x=1$ 用标准四阶龙格-库塔方法式(8.17)求解初值问题

$$\begin{cases}\dfrac{dy}{dx}=y-\dfrac{2x}{y}\\ y(0)=1\end{cases}.$$

解 已知 $f(x,y)=y-\dfrac{2x}{y}$，$y(0)=1$，$h=0.2$. 在$[0,1]$上求 $x=0.2,0.4,0.6,0.8,1.0$ 上的数值解.

只要将 $f(x,y)$，h 代入标准的四阶龙格-库塔格式(8.17)中，得具体的计算公式为

$$\begin{cases}y_{n+1}=y_n+\dfrac{1}{6}(k_1+2k_2+2k_3+k_4)\\ k_1=0.2\left(y_n-\dfrac{2x_n}{y_n}\right)\\ k_2=0.2\left(y_n+\dfrac{k_1}{2}-\dfrac{2(x_n+0.1)}{y_n+\dfrac{k_1}{2}}\right)\\ k_3=0.2\left(y_n+\dfrac{k_2}{2}-\dfrac{2(x_n+0.1)}{y_n+\dfrac{k_2}{2}}\right)\\ k_4=0.2\left(y_n+k_3-\dfrac{2(x_n+0.2)}{y_n+k_3}\right)\end{cases}\tag{8.19}$$

$$(n=0,1,2,\cdots)$$

从 $n=0$ 开始，每算得四个 $k_i(i=1,2,3,4)$的值代入式(8.19)得 y_1，再由 $n=1$ 算得四个 k_i 的值再代入式(8.19)算得 y_2，依次类

推，可得方程的精确解在节点 $x=0.2,0.4,0.6,0.8,1.0$ 上的数值解 y_1,y_2,y_3,y_4,y_5，其计算结果见表 8-1.

表 8-1

n	0	1	2	3	4	5
x_n	0	0.2	0.4	0.6	0.8	1.0
y_n	1	1.18323	1.34167	1.48328	1.61251	1.73214

龙格-库塔方法在求解范围较大而精确度要求较高时是比较好的方法，它和尤拉方法和改进的尤拉方法一样可以直接从头算起，且为了达到同样精确度，它所用的步长可以比尤拉方法和改进的尤拉方法所用的步长大得多.但值得指出的是，由于龙格-库塔方法的推导基于泰勒展开方法，因而它要求所求的解具有较好的光滑性质.反之，如果解的光滑性差，那么，使用四阶龙格-库塔方法求得的数值解，其精度可能反而不如改进的尤拉方法，在实际计算时，我们应当针对问题的具体特点选择合适的算法.

四、步长的选择

怎样选取合适的步长，这在实际计算中是很重要的.从截断误差来看，步长 h 取得越小越好，但随着步长的缩小，要完成的步数就增加，这不但引起计算量的增大，而且步数的增加往往造成舍入误差的严重积累.上面介绍的龙格-库塔方法是对定步长而言的，但为了保证精确度，一种有效的措施是在计算过程中自动调整步长，即变步长技巧.我们以四阶 R-K 方法为例，讨论如何自动选择步长，使计算结果满足给定精度的要求.

设从节点 x_n 出发，先以 h 为步长，利用四阶 R-K 方法经过一步计算得 $y(x_{n+1})$ 的近似值，记为 $y_{n+1}^{(h)}$，由于公式的局部截断误差是 $O(h^5)$，故有

$$y(x_{n+1}) - y_{n+1}^{(h)} \approx ch^5 \tag{8.20}$$

当 h 不大时，c 可近似地看作常数. 然后将步长 h 折半，即取 $h/2$ 为步长，从 x_n 出发经过两步计算求得 $y(x_{n+1})$ 的近似值，记为 $y_{n+1}^{(h/2)}$，每一步计算的局部截断误差为 $c(h/2)^5$，于是有 $y(x_{n+1}) - y_{n+1}^{(h/2)} \approx 2c\left(\frac{h}{2}\right)^5$，把它与式(8.20)相比，经整理可得

$$y(x_{n+1}) - y_{n+1}^{(h/2)} \approx \frac{1}{15}[y_{n+1}^{(h/2)} - y_{n+1}^{(h)}]$$

这表明以 $y_{n+1}^{(h/2)}$ 作为 $y(x_{n+1})$ 的近似值，其误差可用先后两次计算结果之差来表示. 因而，只需考察 $|y_{n+1}^{(h/2)} - y_{n+1}^{(h)}| < \varepsilon$ 是否成立？若成立，则可将 $y_{n+1}^{(h/2)}$ 作为 $y(x_{n+1})$ 的近似值；若不成立，则将步长再次折半进行计算，直到不等式成立为止，并取最后的 $y_{n+1}^{(h/2)}$ 作为计算结果. 以上方法就是计算过程中自动选择步长的方法，也称为变步长方法. 从表面上看，为了选择适当的步长，每一步的计算量增加了，但从总体看，工作量还是减少的.

R-K方法的优点是精度高，通常能满足计算的要求，容易编制程序，每次计算 y_{n+1}，只用到前一步的计算结果 y_n，是单步法，而且计算过程中可随时改变步长. 它的缺点是每前进一步需要计算多次 $f(x,y)$ 的值，因此计算工作量较大，且其截断误差难以估计. 在实用上，一般当要求更高的精度时，采用的办法是缩小步长，而不是采用更高阶的公式，一般选用标准四阶龙格-库塔方法即可.

8.4 线性多步法

上述几节介绍的求解初值问题式(8.1)与式(8.2)的各种数值计算方法都有一个共同的特点，在计算 y_{n+1} 时只用到前面一步提供的信息 y_n，这种方法通称为单步法. 事实上，在计算 y_{n+1} 之前已经求得了一系列的近似值 $y_0, y_1, \cdots, y_n$ 及 $y_0', y_1', \cdots, y_n'$，如果充分

利用这些信息来计算 y_{n+1}，可望计算结果具有较高的精度，这就是线性多步法的基本思想.

解初值问题

$$\begin{cases}\dfrac{dy}{dx}=f(x,y),\\ y(a)=y_0. \quad (a\leqslant x\leqslant b)\end{cases}$$

的线性多步法一般计算公式形如

$$y_{n+1}=\alpha_0 y_n+\alpha_1 y_{n-1}+\cdots+\alpha_r y_{n-r}$$
$$+h(\beta_{-1}f_{n+1}+\beta_0 f_n+\beta_1 f_{n-1}+\cdots+\beta_r f_{n-r}), \quad (8.21)$$

这里，$y_n, y_{n-1}, \cdots, y_{n-r}$ 是已算出的值，$f_k=f(x_k,y_k)$，$(k=n-r, n-r+1, \cdots, n+1)$，其中，常数 $\alpha_0, \alpha_1, \cdots, \alpha_r, \beta_{-1}, \beta_0, \cdots, \beta_r$ 不仅与方程(8.1)的右端 $f(x,y)$ 无关而且也与步数 n 无关. 因此，线性多步法的实质是用若干节点处的函数值和导数值的线性组合来近似 $y(x_{n+1})$ 的值. 由于在计算 y_{n+1} 时需要先知道 $y_n, y_{n-1}, \cdots, y_{n-r}$ 等共 $r+1$ 个函数值，所以称式(8.21)所确定的方法为多步法. 注意到 $f_{n+1}=f(x_{n+1}, y_{n+1})$ 中含有要计算的 y_{n+1}，所以如 $\beta_{-1}\neq 0$，则式(8.21)是隐式；如 $\beta_{-1}=0$，则式(8.21)是显式.

构造多步法有多种途径，可以说所有形如式(8.21)类型的线性多步法的计算公式都可以利用泰勒展开方法构造出来，但其中有一些也可以用数值积分的方法构造出来，如阿达姆斯(Adams)外推法和内插法，下面分别介绍.

一、阿达姆斯外推法

将微分方程式(8.1)的两端从 x_n 到 x_{n+1} 积分，得等价的积分方程

$$y(x_{n+1})=y(x_n)+\int_{x_n}^{x_{n+1}} f(x,y(x))dx \quad (8.22)$$

为了从式(8.22)获得 $y(x_{n+1})$ 的近似值，只要近似地算出其

中的积分项$\int_{x_n}^{x_{n+1}} f(x,y(x))\mathrm{d}x$，而选用不同的数值方法计算这个积分项，就会导出不同的计算公式.

例如，前面已介绍过，用矩形法计算积分项即可导出尤拉公式；为了改善精度，我们改用梯形方法计算积分项就可导出梯形公式，梯形公式也可以理解为积分号下的被积函数 $f(x,y(x))$ 是用点 x_n 和 x_{n+1} 上的线性插值函数近似代替而得出的. 显然，根据插值原理，可以建立一系列的数值积分法，从而导出求解微分方程的一系列数值计算公式，这就是用数值积分方法来构造微分方程数值解的指导思想. 建立阿达姆斯方法的基本思想，就是用一个近似的插值多项式 $p_r(x)$ 来近似代替式(8.22)中的被积函数 $f(x,y)$，然后用

$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} p_r(x)\mathrm{d}x$，代替式(8.22)当选取不同的插值点作插值多项式 $p_r(x)$ 时，就会得出不同的数值解法，下面先推导阿达姆斯外推公式.

要使用插值法，关键在于选取合适的插值节点. 现取 $r+1$ 个等距节点 $x_n, x_{n-1}, x_{n-2}, \cdots, x_{n-r}$，设已经求得这 $r+1$ 个节点处的函数近似值为 $y_n, y_{n-1}, y_{n-2}, \cdots, y_{n-r}$，从而有 $r+1$ 组数据 (x_n, f_n)，$(x_{n-1}, f_{n-1}), \cdots, (x_{n-r}, f_{n-r})$，注意，这里的 $f_k = f(x_k, y_k)$，$(k = n-r, n-r+1, \cdots, n)$ 并不是函数 $f(x,y(x))$ 在点 $x_{n-r}, x_{n-r+1}, \cdots, x_n$ 上的精确值，而是它的近似值，用这 $r+1$ 组数据 (x_k, f_k) 来构造被积函数 $f(x,y(x))$ 的 r 次插值多项式 $p_r(x)$，用它近似代替式(8.22)中的被积函数 $f(x,y(x))$，并用 y_n 和 y_{n+1} 分别近似代替 $y(x_n)$ 和 $y(x_{n+1})$，则得

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} p_r(x)\mathrm{d}x \tag{8.23}$$

由于 $r+1$ 个节点 $x_n, x_{n-1}, \cdots, x_{n-r}$ 是等距节点，且 $x \in [x_n,$

x_{n+1}],用 r 次牛顿向后插值公式

$$p_r(x) = p_r(x_n + sh) = f_n + \frac{s}{1!}\nabla f_n + \frac{s(s+1)}{2!}\nabla^2 f_n + \cdots + \frac{s(s+1)\cdots(s+j-1)}{j!}\nabla^j f_n + \cdots + \frac{s(s+1)\cdots(s+r-1)}{r!}\nabla^r f_n$$

其中，$x = x_n + sh \quad (0 \leqslant s \leqslant 1)$，$\nabla^j$ 是 j 阶向后差分算子，而

$$\frac{s(s+1)\cdots(s+j-1)}{j!}$$

$$= (-1)^j \frac{(-s)(-s-1)\cdots(-s-(j-1))}{j!} = (-1)^j \binom{-s}{j}.$$

称 $\binom{-s}{j}$ 为广义的二项式系数.

将 r 次牛顿向后插值公式 $p_r(x) = \sum_{j=0}^{r}(-1)^j \binom{-s}{j} \nabla^j f_n$ 代入式(8.23)中的积分，并令 $x = x_n + sh$，得

$$\int_{x_n}^{x_{n+1}} p_r(x)\mathrm{d}x = \int_0^1 \sum_{j=0}^{r}(-1)^j \binom{-s}{j} \nabla^j f_n \cdot h\mathrm{d}s$$

$$= h \sum_{j=0}^{r}\left[(-1)^j \int_0^1 \binom{-s}{j}\mathrm{d}s\right]\nabla^j f_n = h \sum_{j=0}^{r} b_j \nabla^j f_n \tag{8.24}$$

其中，$b_j = (-1)^j \int_0^1 \binom{-s}{j}\mathrm{d}s$，不依赖于 n 和 r，b_j 的值可事先计算，表 8-2 给出了 b_j 的部分值.

表 8-2

j	0	1	2	3	4	5	…
b_j	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$	…

将式(8.24)代入式(8.23)即得到阿达姆斯外推公式($r+1$步):

$$y_{n+1} = y_n + h(b_0 f_n + b_1 \nabla f_n + \cdots + b_r \nabla^r f_n)$$

$$= y_n + h\sum_{j=0}^{r} b_j \nabla^j f_n \tag{8.25}$$

式(8.25)是显式的差分方程,故又称为阿达姆斯显式公式,因为插值点 $x \in [x_n, x_{n+1}]$,不包括在插值区间$[x_{n-r}, x_n]$内,故称为外推公式.

若将式(8.25)中的差分展开,写成用函数值表示的形式,则在计算机上使用更为方便,因

$$\nabla^j f_n = \sum_{k=0}^{j} (-1)^k \binom{j}{k} f_{n-k},$$

则有

$$y_{n+1} = y_n + h(\beta_{r0} f_n + \beta_{r1} f_{n-1} + \cdots + \beta_{rr} f_{n-r}), \tag{8.26}$$

其中 $$\beta_{rk} = (-1)^k \sum_{j=k}^{r} \binom{j}{k} b_j, \quad (k = 0, 1, 2, \cdots, r).$$

式(8.26)是阿达姆斯外推公式的常见形式,是 $r+1$ 步的显式多步法公式,注意系数 β_{rk} 与 r 的值有关,r 一旦选定后,对于 $k = 0, 1, 2, \cdots, r$ 便得到一组系数 β_{rk},其部分数值见表 8-3.

表 8-3

k	0	1	2	3
β_{0k}	1			
β_{1k}	3/2	-1/2		
β_{2k}	23/12	-16/12	5/12	
β_{3k}	55/24	-59/24	37/24	-9/24

例如,当 $r = 1$ 时(2步),阿达姆斯公式(8.25)为

$$y_{n+1} = y_n + h(b_0 f_n + b_1 \nabla f_n),$$

其中 $$b_0 = \int_0^1 1\mathrm{d}s = 1, \quad b_1 = \int_0^1 s\mathrm{d}s = \frac{1}{2},$$

所以

$$y_{n+1} = y_n + h(f_n + \frac{1}{2}(f_n - f_{n-1})) = y_n + h\left(\frac{3}{2}f_n - \frac{1}{2}f_{n-1}\right),$$

即 $$\beta_{10} = \frac{3}{2}, \qquad \beta_{11} = -\frac{1}{2}.$$

类似地，可得 $r = 3$ 时的阿达姆斯外推公式为

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}), \quad (8.27)$$

这是常用的阿达姆斯 4 步显式公式.

由牛顿插值多项式的余项表示式即可推得阿达姆斯外推公式 (8.25) 的局部截断误差

$$R_1 = (-1)^{r+1} \cdot h^{r+2} \cdot y^{(r+2)}(\xi_n)\int_0^1 \binom{-s}{r+1}\mathrm{d}s$$

$$= h^{r+2} \cdot b_{r+1} \cdot y^{(r+2)}(\xi_n) = O(h^{r+2}),$$

其中 $\xi_n \in (x_{n-r}, x_{n+1})$.

所以阿达姆斯外推方法是 $r+1$ 阶方法.

二、阿达姆斯内插法

上述阿达姆斯外推法通过 $r+1$ 个点 $(x_n, f_n), (x_{n-1}, f_{n-1}), \cdots, (x_{n-r}, f_{n-r})$ 构造 r 次插值多项式 $p_r(x)$，并用它在求积区间 $[x_n, x_{n+1}]$ 上逼近 $f(x, y(x))$，因为插值点 x 落在插值区间 $[x_{n-r}, x_n]$ 的外边，所以称为外推公式，精度较差. 由插值理论知道，插值节点的选择对精度有直接影响，为了改善逼近效果，我们变外推为内插，改用取 $x_{n+1}, x_n, \cdots, x_{n-r+1}$ 为插值节点，而通过数据点 $(x_{n+1}, f_{n+1}), (x_n, f_n), \cdots, (x_{n-r+1}, f_{n-r+1})$ 构造插值函数 $p_r(x)$，然后重复前一段的推导过程，可得到阿达姆斯内插法公式.

与外推法类似，也是取 $r+1$ 个节点，两者的区别是内插法从 x_{n+1} 开始，这样插值点 x 就落在插值区间 $[x_{n-r+1}, x_{n+1}]$ 内，仍用 r

次牛顿向后插值公式，对函数 $f(x, y(x))$ 作 r 次插值，得 r 次插值公式 $q_r(x) = \sum_{j=0}^{r}(-1)^j\binom{-s}{j}\nabla^j f_{n+1}$，其中 $s = \frac{x - x_{n+1}}{h}$，则 $-1 \leqslant s \leqslant 0$，则积分

$$\int_{x_n}^{x_{n+1}} q_r(x)\mathrm{d}x = h\sum_{j=0}^{r} c_j \nabla^j f_{n+1} \tag{8.28}$$

其中 $c_j = (-1)^j\int_{-1}^{0}\binom{-s}{j}\mathrm{d}s$，显然 c_j 不依赖于 r 和 n，也可事先制成表，表 8-4 给出了 c_j 的前几个值.

表 8-4

j	0	1	2	3	4	5	…
c_j	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$	

令 $y(x_n) = y_n, y(x_{n+1}) = y_{n+1}$，得到数值计算公式

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} q_r(x)\mathrm{d}x \tag{8.29}$$

将 $q_r(x)$ 的积分值式(8.28) 代入式(8.29)，即得阿达姆斯内插公式(r 步)：

$$y_{n+1} = y_n + h(c_0 f_{n+1} + c_1\nabla f_{n+1} + \cdots + c_r\nabla^r f_{n+1}), \tag{8.30}$$

同样，可将式(8.30) 各阶差分展开成函数值的组合并整理得

$$y_{n+1} = y_n + h(\beta_{r0}^* f_{n+1} + \beta_{r1}^* f_n + \cdots + \beta_{rr}^* f_{n-r+1}), \tag{8.31}$$

其中

$$\beta_{rk}^* = (-1)^k\sum_{j=k}^{r}\binom{j}{k}c_j,$$

显然，β_{rk}^* 与节点个数 r 有关，可事先算好制成表. 表 8-5 提供了 β_{rk}^* 的部分数值.

例如，当 $r = 1$ 时，$\beta_{10}^* = \beta_{11}^* = \frac{1}{2}$，则得 $y_{n+1} = y_n +$

$h\left(\frac{1}{2}f_{n+1}+\frac{1}{2}f_n\right)$，此即梯形公式.

表 8-5

k	0	1	2	3
β_{0k}^*	0			
β_{1k}^*	$\frac{1}{2}$	$\frac{1}{2}$		
β_{2k}^*	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$	
β_{3k}^*	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$

由于阿达姆斯内插公式中的 f_{n+1} 含有要求的 y_{n+1}，这是一个含有 y_{n+1} 的方程式，所以阿达姆斯内插公式是一个隐式公式. 常用的是 $r=3$ 的阿达姆斯内插公式，即

$$y_{n+1}=y_n+\frac{h}{24}(9f_{n+1}+19f_n-5f_{n-1}+f_{n-2}), \qquad (8.32)$$

此式是三步隐式公式，由前面三个点的数值，用迭代法来求出下一个点的数值.

同样可证阿达姆斯内插公式的局部截断误差是 $R_2=O(h^{r+2})$，它表示阿达姆斯内插公式是 $r+1$ 阶方法.

若把阿达姆斯外推法与内插法进行比较，可以看出：

(1) 从局部截断误差来看，若 h 的阶数一样，外推法比内插法多用一个预算值，如当 $R=O(h^5)$，外推法用预算值 $y_n, y_{n-1}, y_{n-2}, y_{n-3}$，而内插法只用 y_n, y_{n-1}, y_{n-2}.

(2) 内插法的系数 $|c_j|$ 比外推法的系数 $|b_j|$ 小，因而由计算 f_i 所产生的误差对结果的影响，内插公式比外推公式要小，从而精度高.

(3) 从计算上看，内插法一般要用迭代，计算工作量比外推法

大.

阿达姆斯方法与 R-K 方法相比,其优点是若要求达到同样的截断误差,它的计算工作量比 R-K 方法少.但它的缺点是,首先它不是自开始的方法,当采用阿达姆斯公式时,必须先用其他方法算出开头几个函数值,常用 R-K 方法或泰勒展开法求"表头",其次是计算过程中变动步长较麻烦.

实际计算时,常把阿达姆斯外推公式和内插公式联合使用,即先用外推公式求初始近似值,再用内插公式进行迭代,而通常只迭代一次,这就是又一种预估 - 校正格式,称之为阿达姆斯预估 - 校正系统,经常使用的是 $r = 3$ 的格式,

即

$$\begin{cases} y_{n+1}^{(0)} = y_n + \dfrac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \\ y_{n+1}^{(i+1)} = y_n + \dfrac{h}{24}(9f_{n+1}^{(i)} + 19f_n - 5f_{n-1} + f_{n-2}) \end{cases}, \tag{8.33}$$

$$(i = 0,1,2,\cdots)$$

此两式的局部截断误差均是 $O(h^5)$,因此是四阶四步方法.

三、利用泰勒展开的构造方法

从上面的讨论我们看到,用数值积分的方法可以构造出一系列求解常微分方程的数值计算公式,然而利用泰勒展开的方法对常微分方程离散化,更是常用的数值方法,可以这样说,一切线性多步法公式都可以用泰勒展开的方法构造出来.

利用泰勒展开的方法构造线性多步法的计算公式的基本思想类同于 R-K 方法的推导.

设一般的线性多步法公式为

$$y_{n+1} = \alpha_0 y_n + \alpha_1 y_{n-1} + \cdots + \alpha_r y_{n-r} + h(\beta_{-1} f_{n+1} + \beta_0 f_n + \beta_1 f_{n-1} + \cdots + \beta_r f_{n-r}),$$

即

$$y_{n+1} = \sum_{k=0}^{r} \alpha_k y_{n-k} + h \sum_{k=-1}^{r} \beta_k y'_{n-k}, \tag{8.34}$$

式中　　$y'_{n-k}=f(x_{n-k},y_{n-k})\quad(k=-1,0,1,2,\cdots,r)$,

设 $y_{n-k}=y(x_{n-k}),y'_{n-k}=y'(x_{n-k}),(k=0,1,2,\cdots,r)$,

$y'_{n+1}=y'(x_{n+1})$. 将 $y(x_{n-k}),y'(x_{n-k})$都在 $x=x_n$ 处作泰勒展开,有

$$y(x_{n-k})=y(x_n-kh)=\sum_{j=0}^{p}\frac{(-kh)^j}{j!}y_n^{(j)}+\frac{(-kh)^{p+1}}{(p+1)!}y_n^{(p+1)}+\cdots$$

$$y'(x_{n-k})=\sum_{j=1}^{p}\frac{(-kh)^{j-1}}{(j-1)!}y_n^{(j)}+\frac{(-kh)^p}{p!}y_n^{(p+1)}+\cdots$$

代入式(8.34),整理得

$$\begin{aligned}y_{n+1}=&\left(\sum_{k=0}^{r}a_k\right)y_n+\sum_{j=1}^{p}\frac{h^j}{j!}\left[\sum_{k=1}^{r}(-k)^j a_k+j\sum_{k=-1}^{r}(-k)^{j-1}\beta_k\right]\cdot y_n^{(j)}\\&+\frac{h^{p+1}}{(p+1)!}\left[\sum_{k=1}^{r}(-k)^{p+1}a_k+(p+1)\sum_{k=-1}^{r}(-k)^p\beta_k\right]\cdot y_n^{(p+1)}\\&+\cdots\end{aligned}\tag{8.35}$$

于是,欲使公式(8.34)成为 p 阶的,只要令展开式(8.35)与 $y(x_{n+1})$在 x_n 处的泰勒展开式:

$$y(x_{n+1})=\sum_{j=0}^{p}\frac{h^j}{j!}y_n^{(j)}+\frac{h^{p+1}}{(p+1)!}y_n^{(p+1)}+\cdots\tag{8.36}$$

相比较,能符合到 h^p 项,为此令

$$\begin{cases}\sum_{k=0}^{r}a_k=1\\ \sum_{k=1}^{r}(-k)^j a_k+j\sum_{k=1}^{r}(-k)^{j-1}\cdot\beta_k=1,(j=1,2,\cdots,p)\end{cases}\tag{8.37}$$

解出系数 α_k,β_k 代入式(8.34),即得 p 阶的线性多步法公式.

由式(8.36)与式(8.35)相减可得方法的局部截面断误差:

$$\begin{aligned}y(x_{n+1})-y_{n+1}=\frac{h^{p+1}}{(p+1)!}&\Big[1-\sum_{k=1}^{r}(-k)^{p+1}\cdot a_k\\&-(p+1)\sum_{k=-1}^{r}(-k)^p\cdot\beta_k\Big]y_n^{(p+1)}+\cdots\end{aligned}\tag{8.38}$$

例 8.3　构造一个以 $x_{n-2},x_{n-1},x_n,x_{n+1}$ 为节点的三步线性公式:

$$y_{n+1} = \alpha_0 y_n + \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + h(\beta_{-1} f_{n+1} + \beta_0 f_n + \beta_1 f_{n-1}) \tag{8.39}$$

解 显然,关键只要确定公式中的系数 $\alpha_0, \alpha_1, \alpha_2, \beta_{-1}, \beta_0, \beta_1$.

设
$$y_i = y(x_i) \quad (i = n-2, n-1, n),$$
$$y_i' = y'(x_i), \ (i = n-1, n, n+1),$$

然后将函数 $y(x_{n-2}), y(x_{n-1}), y'(x_{n-1}), y'(x_{n+1})$都在点 $x = x_n$ 处作泰勒展开后,代入式(8.39),再按 h 升幂排列整理得

$$\begin{aligned} y_{n+1} = {} & (\alpha_0 + \alpha_1 + \alpha_2) y(x_n) + (-\alpha_1 - 2\alpha_2 + \beta_{-1} + \beta_0 + \beta_1) y'(x_n) h \\ & + \left(\frac{1}{2}\alpha_1 + 2\alpha_2 + \beta_{-1} - \beta_1\right) y''(x_n) h^2 \\ & + \left(-\frac{1}{6}\alpha_1 - \frac{4}{3}\alpha_2 + \frac{1}{2}\beta_{-1} + \frac{1}{2}\beta_1\right) y'''(x_n) h^3 \\ & + \left(\frac{1}{24}\alpha_1 + \frac{2}{3}\alpha_2 + \frac{1}{6}\beta_{-1} - \frac{1}{6}\beta_1\right) y^{(4)}(x_n) h^4 \\ & + \left(-\frac{1}{120}\alpha_1 - \frac{32}{120}\alpha_2 + \frac{1}{24}\beta_{-1} + \frac{1}{24}\beta_1\right) y^{(5)}(x_n) h^5 + \cdots \end{aligned} \tag{8.40}$$

要使公式成为五阶,也即局部截断误差为 $y(x_{n+1}) - y_{n+1} = O(h^6)$,则要使公式(8.40)与方程的精确解 $y(x_{n+1})$在 x_n 处的泰勒展开式中的前六项关于 h 的同次幂系数完全相同,从而得关于未知元 $\alpha_0, \alpha_1, \alpha_2, \beta_{-1}, \beta_0, \beta_1$ 的六个方程,从中解得 $\alpha_0, \alpha_1, \alpha_2$ 和 $\beta_{-1}, \beta_0, \beta_1$ 后代入式(8.39)中,即得到一个五阶的数值计算公式.

当然也可先令 $\alpha_0 = 0, \alpha_2 = 0$,这时从前四个方程中解得 $\alpha_1 = 1, \beta_{-1} = \frac{1}{3}, \beta_0 = \frac{4}{3}, \beta_1 = \frac{1}{3}$代入式(8.39)得公式:

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}),$$

这个公式称为辛普生公式,此公式也可用数值积分得到,这是一个四阶数值计算公式.

8.5 收敛性与稳定性

收敛性与稳定性从不同角度描述了数值方法的可靠性.只有既收敛又稳定的方法,才能提供比较可靠的计算结果.为了简单起见,本节只讨论单步法的收敛性与稳定性.单步法是只要利用 x_n 及 $y(x)$在 x_n 上的近似值 y_n 就可以确定 $y(x)$在 x_{n+1}上的近似值 y_{n+1}的数值方法.尤拉方法、R-K 方法都是单步法的例子.一般的显式单步法的计算公式可以统一地写成

$$y_{n+1} = y_n + h\psi(x_n, y_n, h), \tag{8.41}$$

其中,h 为步长,$x_n = a + nh$, y_n 是方程的精确解 $y(x)$在点 x_n 处的近似值,而 $\psi(x, y, h)$称为增量函数,它依赖于 f,且仅仅是 x_n, y_n 和 h 的函数,它是关于所含变量的连续函数.

不同的单步法,对应于不同的增量函数.例如尤拉公式(8.3)其增量函数 $\psi(x, y, h) = f(x, y)$,改进的尤拉方法的增量函数

$$\psi(x, y, h) = \frac{1}{2}[f(x, y) + f(x + h, y + hf(x, y))].$$

一、单步法的收敛性

我们已知微分方程初值问题的数值解法的基本思想是通过某种离散化手续,将微分方程转化为差分方程来求解,用不同的离散方法,便得到不同的数值计算公式,这里,首先一个问题是,这种离散化是否合理,例如单步法:$y_{n+1} = y_n + h\psi(x_n, y_n, h)$的解当 $h \to 0$ 时(同时 $n \to \infty$),是否会收敛到微分方程的精确解.

定义 8.2 某数值方法对任意固定的节点 $x_n = x_0 + nh$,($n = 1, 2, \cdots$),当 $h \to 0$(同时 $n \to \infty$)时,有 $y_n \to y(x_n)$,则称该方法是收敛的.

我们是在初值问题式(8.1)与式(8.2)的右端 $f(x, y)$在区域 $a \leqslant x \leqslant b$, $-\infty < y < +\infty$ 上满足李普希兹条件,即 $|f(x, y_1) -$

$f(x,y_2)|\leqslant L|y_1-y_2|$(L为李氏常数)的前提下,讨论单步法 $y_{n+1}=y_n+h\psi(x_n,y_n,h)$的收敛性.

从上述收敛性定义可以看出,数值方法的收敛性并不涉及计算过程的舍入误差,而只与方法的截断误差有关,我们有下述关于估计方法的截断误差及判别单步法收敛的充分定理.

定理 8.1 若①单步法 $y_{n+1}=y_n+h\psi(x_n,y_n,h)$中的增量函数 $\psi(x,y,h)$在区域 $a\leqslant x\leqslant b,-\infty<y<+\infty,0\leqslant h\leqslant h_0$ 上连续,并且关于 y 满足李普希兹条件:$|\psi(x,y,h)-\psi(x,z,h)|\leqslant L|y-z|,(L>0)$;②方法的局部截断误差 $\varepsilon_n=O(h^{p+1}),(n=1,2,\cdots)$;③初始值是精确的,即 $y(a)=y_0$.

则 (1) 方法的整体截断误差为 $e_n=O(h^p)$;

(2) 当 $p\geqslant 1$ 时,方法是收敛的.

(证明从略).

以上定理表明,只要增量函数 $\psi(x,y,h)$满足李普希兹条件,当 $e_0=y_0-y(a)=O$,且阶数 $p\geqslant 1$ 时,数值方法式(8.41)是收敛的,且其局部截断误差 $\varepsilon_n=O(h^{p+1})$时,其整体截断误差 $e_n=O(h^p)$,由于整体截断误差比局部截断误差低一阶,这结论带有一般性,因此在构造高精度的计算方法时,只要设法提高方法的局部截断误差.

依据这一定理,可知在初始值精确的前提下,判断单步法的收敛性,只要验证它们的增量函数 $\psi(x,y,h)$ 能否满足李普希兹条件就可以了.

例如,对于尤拉方法:$y_{n+1}=y_n+hf(x_n,y_n)$,由于其增量函数 $\psi(x,y,h)$就是 $f(x,y)$,所以当初值问题式(8.1)的右端函数 $f(x,y)$满足李普希兹条件且初值 y_0 是精确时,它是收敛的.

类似地,不难验证其它单步法的收敛性.

例如,标准四阶 R-K 方法在初值问题式(8.1)的右端函数,$f(x,y)$关于 y 满足李普希兹条件时,应用定理 8.1 也可以证明其

是收敛的.

二、单步法的稳定性

前面讨论的收敛性是很重要的,如果常微分方程的数值解不收敛于它的精确解,那么,它个数值解法只能弃而不用,但前面关于收敛性的讨论有个前提,必须假设数值方法本身的计算是精确的,这样得到的数值解我们称之为初值问题式(8.1)与式(8.2)的精确数值解.在实际计算时,一方面,初始值 y_0 不一定是完全精确的,往往带有一定的误差,同时由于计算机字长有限,在计算过程中一般总会产生舍入误差,且这种误差都会传递下去,对以后的结果产生影响.所谓数值解的稳定性问题,就是指误差的积累是否受到控制的问题,粗略地说,如果计算结果对初始数据的误差及计算过程中的误差不敏感,也即舍入误差不增长,就称相应的计算方法是稳定的,否则就称为不稳定的.

数值稳定性有各种各样的定义,下面我们只简单地介绍"绝对稳定性"的概念.

设某数值方法在节点 x_n 处的初值问题式(8.1)与式(8.2)的数值解仍记为 y_n,而实际计算得到的近似值为 $\bar{y}_n$,差值 $\delta_n = \bar{y}_n - y_n$ 称为第 n 步数值解的扰动.假设 $\delta_n \neq 0$,即第 n 步确有扰动,则关于绝对稳定性定义如下.

定义 8.3 若一种数值方法在节点 x_n 处的数值解 y_n 处有大小为 δ_n 的扰动,而在以后各节点值 $y_m(m > n)$上产生的扰动按绝对值均不超过 $|\delta_n|$,即 $|\delta_m| \leqslant |\delta_n|$,$(m = n+1, n+2, \cdots)$,则称该数值方法是绝对稳定的.

数值稳定性的分析相当复杂,且总跟微分方程的右端 $f(x, y)$ 有关,所以,对一般的一阶常微分方程研究这个问题很困难,事实上,我们不可能也不必对每个不同的右端函数 $f(x, y)$进行讨论,为简单起见,通常只用如下的"典型方程"(也称为试验方程或模型方程):

$$\frac{dy}{dx}=\lambda y,(\text{其中 }\lambda\text{ 是复常数,且 }\mathrm{Re}(\lambda)<0)$$

来讨论数值方法的绝对稳定性,这种思想来源于如下判断:若一个数值方法对如此简单问题都不是绝对稳定的话,那么就难以用此数值方法来解一般的微分方程的初值问题,不言而喻,某一数值方法对试验方程是绝对稳定的,对一般微分方程不一定也是绝对稳定的,但试验方程在一定程度上还是反映了数值方法的特性.

在把某一数值方法应用于试验方程,为了保证数值稳定,步长 h 和 λ 都要受到一定的限制,它们所允许的范围,就称为相应方法的绝对稳定性区域,显然,绝对稳定性区域越大,我们便说这种解法的绝对稳定性越好.若某个方法的绝对稳定性区域包括 $h\lambda$ 复平面的左半平面,则称此数值方法是 A-稳定的.

例如,先考虑尤拉方法的绝对稳定性区域,把该方法用到试验方程 $\frac{dy}{dx}=\lambda y$ 上,得

$$y_{n+1}=y_n+hf(x_n,y_n)=y_n+\lambda hy_n=(1+\lambda h)y_n,$$

而实际计算得近似解满足:$\bar{y}_{n+1}=\bar{y}_n+\lambda h\bar{y}_n$,两式相减,并令 $\delta_n=y_n-\bar{y}_n$,得

$$\delta_{n+1}=y_{n+1}-\bar{y}_{n+1}=(1+\lambda h)(y_n-\bar{y}_n)=(1+\lambda h)\delta_n,$$

若 $|1+\lambda h|\leqslant 1$,显然误差就不增大,这时尤拉方法是绝对稳定的.

条件 $|1+h\lambda|<1$ 在 $h\lambda$ 的复平面上,这是以 1 为半径,以 -1 为中心的圆内部,所以我们说尤拉方法的绝对稳定性区域是圆域,如图 8-3 所示.

条件 $|1+h\lambda|<1$ 是使用尤拉方法时为了保证绝对稳定性对步长 h 的限制,如 $\lambda=-5$ 时,步长 h 应小于 0.4;$\lambda=-100$ 时,步长 h 应小于 0.02,显然,为了保证尤拉方法的稳定性,必须选取步长 h 满足 $|1+\lambda h|<1$,这表明尤拉方法是条件稳定的,上述稳定性条件可表为 $0<h<-\frac{2}{\lambda}$.

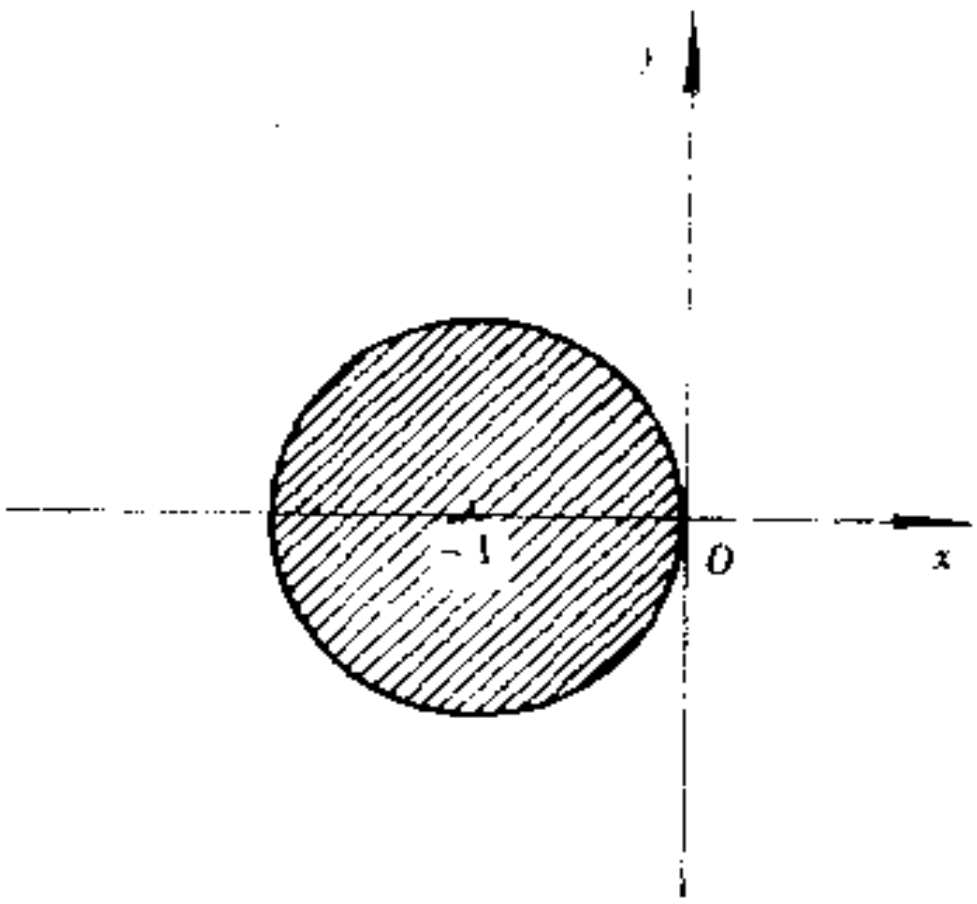


图 8-3　尤拉法的绝对稳定区域

对于后退的尤拉方法：

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}),$$

用试验方程$\frac{dy}{dx} = \lambda y$，得 $y_{n+1} = y_n + \lambda h y_{n+1}$，解出 y_{n+1}，有 $y_{n+1} = \frac{1}{1-\lambda h} y_n$，则对于后退的尤拉方法，$\delta_n$ 应满足 $\delta_{n+1} = \frac{1}{1-\lambda h}\delta_n$，可见它的绝对稳定性区域是 $|1 - \lambda h| > 1$.

在 λh 复平面上，这是以 1 为半径、以 1 为中心的圆外部，如图 8-4 所示，显然，该方法是 A-稳定的.

由图 8-3 和图 8-4 可见，后退尤拉方法的绝对稳定性区域比尤拉方法大得多，但可证明，这两种方法的阶数相同，只是尤拉方法是显式方法，而后退的尤拉方法是隐式方法，这说明，隐式方法的绝对稳定性一般比同阶的显式方法好得多.

用类似的方法对于“标准的四阶 R-K 方法”可推得其绝对稳定性区域为

$$\left|1 + \lambda h + \frac{1}{2}(\lambda h)^2 + \frac{1}{6}(\lambda h)^3 + \frac{1}{24}(\lambda h)^4\right| < 1$$

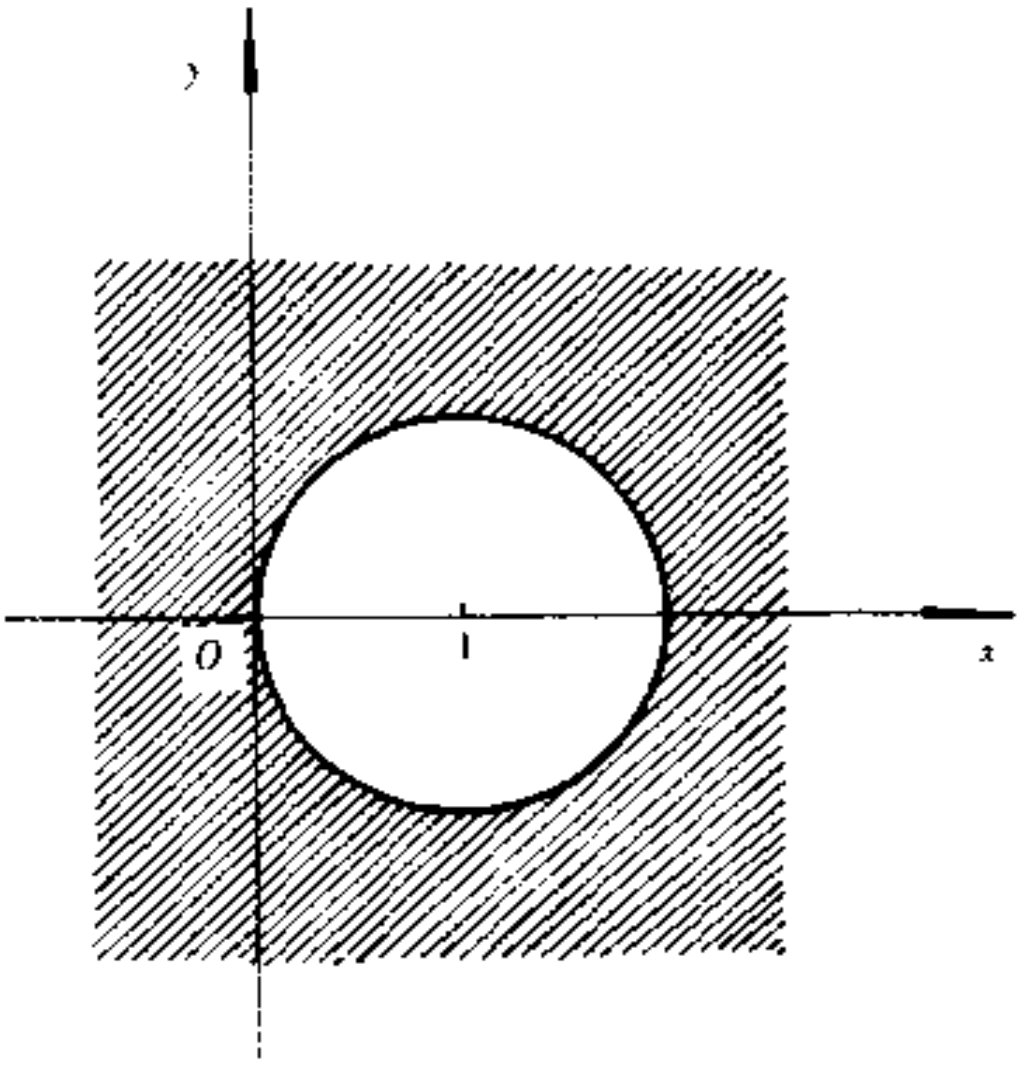


图 8-4　后退尤拉方法的绝对稳定区域

(若设 $\lambda<0$,由上式可近似地得到 $0<h\leqslant\dfrac{-2.78}{\lambda}$).

收敛性和稳定性是两个重要的概念,在数值分析的不同分枝,它们的含义可以不同,我们这里介绍的收敛性反映数值公式本身截断误差对计算结果的影响;稳定性是反映某一计算步骤中出现的舍入误差对计算结果的影响.稳定性是和步长 h 密切相关的,对于一种步长是稳定的数值公式,若改变步长为大一点的步长可能就不稳定,只有既收敛又稳定的数值方法才可以在实际计算中放心地使用.

8.6　一阶方程组和高阶方程

前面我们讨论了一阶常微分方程初值问题的数值解法.而在实际问题中,遇到的常常是微分方程组和高阶微分方程,但了解了单个方程 $y'=f(x,y)$ 的数值解法后,只要把 y 和 f 理解为向量,

那么,以上所介绍的各种数值计算格式都可平行地推广应用到一阶方程组的情况.

一、一阶方程组

设微分方程组的初值问题表为

$$\begin{cases} y_1'(x) = f_1(x, y_1, y_2, \cdots, y_m) \\ y_2'(x) = f_2(x, y_1, y_2, \cdots, y_m) \\ \cdots \\ y_m(x) = f_m(x, y_1, y_2, \cdots, y_m) \end{cases} \quad (a \leqslant x \leqslant b) \tag{8.42}$$

初值为

$$\begin{cases} y_1(a) = s_1 \\ y_2(a) = s_2 \\ \cdots \\ y_m(a) = s_m \end{cases} \tag{8.43}$$

为了求满足式(8.43)的式(8.42)数值解,引进向量记号:

$$\boldsymbol{y}'(x) = (y_1'(x), y_2'(x), \cdots, y_m'(x))^{\mathrm{T}},$$

$$\boldsymbol{f}(x, \boldsymbol{y}) = (f_1(x, \boldsymbol{y}), f_2(x, \boldsymbol{y}), \cdots, f_m(x, \boldsymbol{y}))^{\mathrm{T}}$$

$$\boldsymbol{s} = (s_1, s_2, \cdots, s_m)^{\mathrm{T}}, \quad \boldsymbol{y}(x) = (y_1(x), y_2(x), \cdots, y_m(x))^{\mathrm{T}}$$

由方程组式(8.42)和初值式(8.43)可改写为向量形式:

$$\begin{cases} \boldsymbol{y}'(x) = \boldsymbol{f}(x, \boldsymbol{y}(x)) \\ \boldsymbol{y}(a) = \boldsymbol{s} \end{cases} \tag{8.44}$$

它在形式上具有与一阶微分方程初值问题式(8.1)与式(8.2)类似的形式,只是函数变成了向量函数,从而,前述的各种数值方法都可以推广到式(8.44)上去,推广的方法是只要把函数换成向量函数就行了.

例如,尤拉公式应写为

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\boldsymbol{f}(x_n, \boldsymbol{y}_n) \tag{8.45}$$

其中，

$$\boldsymbol{y}_n=(y_{1n},y_{2n},\cdots,y_{mn})^{\mathrm{T}}\approx\boldsymbol{y}(x_n)=(y_1(x_n),y_2(x_n),\cdots,y_m(x_n))^{\mathrm{T}}$$

又如适用于式(8.44)的标准四阶龙格-库塔公式应写为

$$\begin{cases}\boldsymbol{y}_{n+1}=\boldsymbol{y}_n+\dfrac{1}{6}[\boldsymbol{k}_1+2\boldsymbol{k}_2+2\boldsymbol{k}_3+\boldsymbol{k}_4]\\ \boldsymbol{k}_1=h\boldsymbol{f}(x_n,\boldsymbol{y}_n)\\ \boldsymbol{k}_2=h\boldsymbol{f}\left(x_n+\dfrac{h}{2},\boldsymbol{y}_n+\dfrac{1}{2}\boldsymbol{k}_1\right)\\ \boldsymbol{k}_3=h\boldsymbol{f}\left(x_n+\dfrac{h}{2},\boldsymbol{y}_n+\dfrac{1}{2}\boldsymbol{k}_2\right)\\ \boldsymbol{k}_4=h\boldsymbol{f}(x_n+h,\boldsymbol{y}_n+\boldsymbol{k}_3)\end{cases}\tag{8.46}$$

其中 $\boldsymbol{y}_n$ 表示 $\boldsymbol{y}(x)$ 在 $x=x_n$ 处的近似向量，写成分量形式就是

$$\begin{cases}y_{i,n+1}=y_{in}+\dfrac{1}{6}(k_{i1}+2k_{i2}+2k_{i3}+k_{i4})\\ k_{i1}=hf_i(x_n,y_{1n},y_{2n},\cdots,y_{mn})\\ k_{i2}=hf_i\left(x_n+\dfrac{h}{2},y_{1n}+\dfrac{1}{2}k_{11},y_{2n}+\dfrac{1}{2}k_{21},\cdots,y_{mn}+\dfrac{1}{2}k_{m1}\right)\\ k_{i3}=hf_i\left(x_n+\dfrac{h}{2},y_{1n}+\dfrac{1}{2}k_{12},y_{2n}+\dfrac{1}{2}k_{22},\cdots,y_{mn}+\dfrac{1}{2}k_{m2}\right)\\ k_{i4}=hf_i(x_n+h,y_{1n}+k_{13},y_{2n}+k_{23},\cdots,y_{mn}+k_{m3})\end{cases}$$

$$(i=1,2,\cdots,m)\tag{8.47}$$

这里，y_{in} 是 $y_i(x)$ 在节点 $x_n=x_0+nh$ 处的近似值，h 是步长.

二、高阶微分方程的初值问题

关于高阶方程的初值问题，一般可引入变量代换，转化为一阶方程组初值问题的方法求解.具体做法如下：

设高阶方程的初值问题为

$$\begin{cases}y^{(m)}(x)=f(x,y,y',\cdots,y^{(m-1)})\\ y(a)=s_1,y'(a)=s_2,\cdots,y^{(m-1)}(a)=s_m\end{cases}\tag{8.48}$$

引进新变量 $y_1 = y, y_2 = y', \cdots, y_m = y^{(m-1)}$，则式(8.48)转化为一阶方程组的初值问题

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \cdots \\ y_{m-1}' = y_m \\ y_m' = f(x, y_1, y_2, \cdots, y_m) \end{cases}, \quad \text{初值} \begin{cases} y_1(a) = s_1 \\ y_2(a) = s_2 \\ \cdots \\ y_m(a) = s_m \end{cases}.$$

再用一阶方程组中介绍的方法解此方程组的初值问题.

例如 设

$$\begin{cases} \dfrac{d^2y}{dx^2} = f\left(x, y, \dfrac{dy}{dx}\right) \\ y(0) = y_0 \\ y'(0) = y_0' \end{cases}.$$

只要引进新的变量，如 $z = \dfrac{dy}{dx}$，即可将上述二阶方程转化为一阶方程组的初值问题：

$$\begin{cases} \dfrac{dz}{dx} = f(x, y, z) \\ \dfrac{dy}{dx} = z \end{cases}, \quad \text{其初值条件为} \begin{cases} y(0) = y_0 \\ z(0) = y_0' \end{cases}.$$

这时，其四阶龙格-库塔格式具有以下形式：

$$\begin{cases} z_{n+1} = z_n + \dfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ y_{n+1} = y_n + \dfrac{1}{6}(L_1 + 2L_2 + 2L_3 + L_4) \end{cases},$$

其中

$$\begin{cases} k_1 = hf(x_n, y_n, z_n) \\ k_2 = hf\left(x_n + \dfrac{h}{2}, y_n + \dfrac{1}{2}\mathrm{L}_1, z_n + \dfrac{1}{2}k_1\right) \\ k_3 = hf\left(x_n + \dfrac{h}{2}, y_n + \dfrac{1}{2}\mathrm{I}_2, z_n + \dfrac{1}{2}k_2\right) \\ k_4 = hf(x_n + h, y_n + \mathrm{L}_3, z_n + k_3) \end{cases},$$

$$\begin{cases} \mathrm{L}_1 = hz_n \\ \mathrm{L}_2 = hz_n + \dfrac{h}{2}k_1 \\ \mathrm{L}_3 = hz_n + \dfrac{h}{2}k_2 \\ \mathrm{L}_4 = hz_n + hk_3 \end{cases}$$

习 题 8

1. 用尤拉方法求解初值问题(取步长 $h = 0.1$)

$$\begin{cases} y' = -y + x + 1, \quad x \in (0,1] \\ y(0) = 1 \end{cases}.$$

2. 证明三段龙格-库塔方法

$$\begin{cases} y_{n+1} = y_n + \dfrac{1}{4}(k_1 + 3k_3) \\ k_1 = hf(x_n, y_n) \\ k_2 = hf\left(x_n + \dfrac{h}{3}, y_n + \dfrac{k_1}{3}\right) \\ k_3 = hf\left(x_n + \dfrac{2h}{3}, y_n + \dfrac{2k_2}{3}\right) \end{cases}$$

是三阶方法.

3. 取步长 $h = 0.2$,用标准四阶龙格-库塔方法求解初值问题.

$$\begin{cases} y' = x + y, & x \in (0,1] \\ y(0) = 1 \end{cases}.$$

4. 求系数 a, b, c 和 d，使 $y_{n+1} = ay_{n-1} + h(by'_{n+1} + cy'_n + dy'_{n-1})$ 有 $y(x_{n+1}) - y_{n+1} = O(h^5)$.

5. 验证公式 $y_{n+1} = y_n + h\psi(x_n, y_n, h)$ 是二阶龙格-库塔方法，其中，$\psi(x, y, h) = \frac{1}{2}[f(x, y), f(x + h, y + hf(x, y))]$. 若 L_0 是 f 关于 y 的李普希兹常数，验证对 $h \leqslant h_0$，$L = L_0\left(1 + \frac{h_0}{2}L_0\right)$ 是 ψ 关于 y 的李普希兹常数.

6. 对初值问题 $y' = -10y$，且 $y(x_0) = y_0$，由习题 8 第 5 题给出的推算公式，讨论绝对稳定性对步长 h 的限制.

7. 证明下述格式对任意参数 t 都是二阶的.

$$\begin{cases} y_{n+1} = y_n + \frac{1}{2}(k_2 + k_3) \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + th, y_n + tk_1) \\ k_3 = hf(x_n + (1 - t)h, y_n + (1 - t)k_1) \end{cases}$$

8. 取步长为 $h = 0.1$，分别用四阶 Adams 内插公式与外推公式求解下述初值问题的数值解：

$$\begin{cases} y' = -y + x + 1, x \in (0,1], \\ y(0) = 1 \end{cases}.$$

附录　部分算法的数值计算程序

本附录所有程序都是用 FORTRAN77 语言编制，并在 pc-486 微机上用 NDP 软件对典型的例子进行了调试. 读者可结合前面的理论及算法对此部分程序进行调试，以加深对所学理论和算法的理解. 更主要的希望藉此以提高读者解决具体问题的兴趣。

一、列选主元高斯消去法

本方法由两个子程序组成：GAUSS 和 SOLVE. 子程序 GAUSS 的作用是对数组 A(IA,N)执行列选主元高斯消元；子程序 SOLVE 是处理右端项数组 B 和执行回代过程.

1. 子程序 GAUSS(N,A,IA,L,S)中变量说明

A：待消元的数组；

IA：数组 A 的行维数；

N：数组 A 的列维数；

L：记录消元过程中主元位置的指标数组；

S：记录 A 的每列绝对值最大的元素，即 $S(I)=\underset{1\leqslant J\leqslant N}{MAX}|A(I,J)|$，$I=1,\cdots,IA$；

2. 子程序 GAUSS 的算法要点

(1) 给数组 L(IA)赋初值：

$$L(I)\leftarrow I;\quad (I=1,\cdots,IA)$$

计算数组 S(IA)的每个值

$$S(I)=\underset{1\leqslant J\leqslant N}{MAX}|A(I,J)|,\quad (I=1,\cdots,IA)$$

(2) 对列 $k=1,2,\cdots,N-1$ 选主元，主元方程的位置放在数组 L 的第 k 个元素上，计算消元乘子数：

$$XMULT = A(L(I),K)/A(LK,K)$$

消元：

$$A(L(I),J) = A(L(I),J) - XMULT * A(LK,J)$$

贮存乘子数：

$$A(L(I),K) \leftarrow XMULT$$

3. 子程序 SOLVE(N,A,IA,L,B,X)中的变量说明

A,N,IA,L:同 GAUSS 中的变量含义一样；

B:右端项数组 B(N)；

X:方程组的解；

4. 子程序 SOLVE 的算法要点

(1) 对数组 B 作消元修改；

(2) 回代产生解；

5. 运用列选主元的方法对方程组

(1) $$\mathbf{AX} = \mathbf{B}$$

其中

$$\mathbf{A} = \begin{bmatrix} 0.4096 & 0.1234 & 0.3678 & 0.2943 \\ 0.2246 & 0.3872 & 0.4015 & 0.1129 \\ 0.3345 & 0.1920 & 0.3781 & 0.0643 \\ 0.1784 & 0.4002 & 0.2786 & 0.3927 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0.4043 \\ 0.1550 \\ 0.4240 \\ 0.2557 \end{bmatrix}$$

求解得

$$\mathbf{X} = \begin{bmatrix} 3.4606 \\ 1.5610 \\ -2.9342 \\ -0.4301 \end{bmatrix}$$

(2) $$\mathbf{AX} = \mathbf{B}$$

其中 **A** 与(1)的系数矩阵 **A** 只在一个位置不同

$$\mathbf{A}=\begin{bmatrix}0.4096 & 0.1234 & 0.3678 & 0.2943\\ 0.2246 & 0.3872 & 0.4015 & 0.1129\\ 0.3645 & 0.1920 & 0.3781 & 0.0643\\ 0.1784 & 0.4002 & 0.2786 & 0.3927\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0.4043\\ 0.1550\\ 0.4240\\ 0.2557\end{bmatrix}$$

求解得

$$\mathbf{X}=\begin{bmatrix}6.7831\\ 3.5914\\ -6.4451\\ -1.5179\end{bmatrix}$$

想想看，为什么系数矩阵作一个小小的变化，解的变化很大？

6. 程序

```
      SUBROUTINE GAUSS(N,A,IA,L,S)
      DIMENSION A(IA,N),L(N),S(N)
      DO 3 I=1,IA
      L(I)=I
      SMAX=0.0
      DO 2 J=1,N
      SMAX=AMAX1(SMAX,ABS(A(I,J)))
2     CONTINUE
      IF(SMAX.LE.1.E-8)THEN
      WRITE(*,*)'DET(A)=0.0.EQUATIONS HAVE NO SLOVES'
      STOP
      END IF
      S(I)=SMAX
3     CONTINUE
      DO 7 K=1,N-1
      RMAX=0.0
        DO 4 I=K,N
        R=ABS(A(L(I),K))/S(L(I))
        IF(R.LE.RMAX)GO TO 4
        J=I
        RMAX=R
```

```
4         CONTINUE
          IF (RMAX.LE.1.E-8)THEN
          WRITE(*,*)'DET(A) = 0.0,EQUATIONS HAVE NO SLOVES'
          STOP
          END IF
          LK = L(J)
          L(J) = L(K)
          L(K) = LK
            DO 6 I = K + 1,N
            XMULT = A(L(I),K)/A(LK,K)
              DO 5 J = K + 1,N
              A(L(I),J) = A(L(I),J) - XMULT * A(LK,J)
5             CONTINUE
            A(L(I),K) = XMULT
6           CONTINUE
7         CONTINUE
          RETURN
          END

          SUBROUTINE SOLVE(N,A,IA,L,B,X)
          DIMENSION A(IA,N),L(N),B(N),X(N)
          DO 3 K = 1,N - 1
            DO 2 I = K + 1,N
            B(L(I)) = B(L(I)) - A(L(I),K) * B(L(K))
2           CONTINUE
3         CONTINUE
          X(N) = B(L(N))/A(L(N),N)
          DO 5 I = N - 1,1, - 1
          SUM = B(L(I))
            DO 4 J = I + 1,N
            SUM = SUM - A(L(I),J) * X(J)
4           CONTINUE
          X(I) = SUM/A(L(I),I)
5         CONTINUE
```

```
      RETURN
      END

      DIMENSION A(4,4),L(4),B(4),X(4)
      DATA A/0.4096,0.2246,0.3345,0.1784,0.1234,0.3872,0.1920,0.4002
     *      0.3678,0.4015,0.3781,0.2786,0.2943,0.1129,0.0643,0.3927/,
     *      B/0.4043,0.1550,0.4240,0.2557/
      OPEN(10,FILE='OU2.DAT')
      WRITE(10,*)'----------A----------'
      DO 6 I=1,4
6     WRITE(10,11)(A(I,J),J=1,4)
11    FORMAT(5X,4E18.8)
      WRITE(10,*)'----------B----------'
      WRITE(10,11)(B(J),J=1,4)
      CALL GAUSS(4,A,4,L,X)
      CALL SOLVE(4,A,4,L,B,X)
      WRITE(10,*)'----------X----------'
      WRITE(10,11)(X(J),J=1,4)
      CLOSE(10)
      STOP
      END
----------A----------
     0.40959999E+00  0.12340000E+00  0.36780000E+00  0.29429999E+00
     0.22460000E+00  0.38720000E+00  0.40149999E+00  0.11290000E+00
     0.33450001E+00  0.19200000E+00  0.37810001E+00  0.64300001E-01
     0.17839999E+00  0.40020001E+00  0.27860001E+00  0.39269999E+00
----------B----------
     0.40430000E+00  0.15500000E+00  0.42399999E+00  0.25569999E+00
----------X----------
     0.67830997E+01  0.35914426E+01 -0.64451241E+01 -0.15179240E+01

----------A----------
     0.40959999E+00  0.12340000E+00  0.36780000E+00  0.29429999E+00
     0.22460000E+00  0.38720000E+00  0.40149999E+00  0.11290000E+00
```

0.36449999E+00　0.19200000E+00　0.37810001E+00　0.64300001E-01

0.17839999E+00　0.40020001E+00　0.27860001E+00　0.39269999E+00

----------B----------

0.40430000E-00　0.15500000E+00　0.42399999E+00　0.25569999E+00

----------X----------

0.34605875E+01　0.15609535E+01　-0.29342353E+01　-0.43005961E+00

二、ROMBERG 方法求定积分的值

1. 程序要点

ROMBERG 算法产生一个三角数组，这个数组用记号表示如下：

$$\begin{array}{llllll} R(1,1) & & & & & \\ R(2,1) & R(2,2) & & & & \\ R(3,1) & R(3,2) & R(3,3) & & & \\ R(4,1) & R(4,2) & R(4,3) & R(4,4) & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ R(N,1) & R(N,2) & R(N,3) & R(N,4) & \cdots & R(N,N) \end{array}$$

其中的每一个数都是对定程分 $\int_a^b f(x)dx$ 的数值估计。

第 1 列 R(i,1)是 2^i 个相等子区间运用梯形法则的结果

$$R(1,1)=\frac{1}{2}(b-a)(f(a)+f(b))$$

$$R(i,1)=\frac{1}{2}R(i-1,1)+h\sum_{k=1}^{2^{i-1}-1}f(a+(2k-1)h)$$

$$(h=(b-a)/2^i,\quad i>1)$$

第 2 列以及后面的各列是由外推公式产生：

$$R(i,j)=R(i,j-1)+\frac{1}{4^{j-1}-1}(R(i,j-1)-R(i-1,j-1))$$

本程序是逐行产生 ROMBERG 数组.

2. 子程序 ROMBERG(F,A,B,N,R,ER)变量说明

F:被积函数,以函数子程序给出;

A,B:积分的下上限;

R:ROMBERG 数组;

N:计算步数;

ER:计算误差;

3. 运用本程序对定积分 $\int_0^{\pi} \sin x dx$ 进行了计算。

4. 程序

```
C
C
C
C     THIS PROGRAM IS ROMBERG METHOD. IT CONTINUES ONE SUBROUTINE
C     AND ONE FUNCTION AND ONE MAIN PROGRAM.
C          F——FUNCTION;
C          A,B——INTERNAL NODE;
C          N——CALCULATE STEPS;
C          R——VALUE;
C          ER——CONTRALED ERROR;
C
C
      SUBROUTINE ROMBRG(F,A,B,N,R,ER)
      DIMENSION R(N,N)
      H=B-A
      R(1,1)=0.5*H*(F(A)+F(B))
      OPEN(10,FILE='OUT1.DAT')
      WRITE(10,5)R(1,1)
      L=1
      DO 4 I=2,N
      H=0.5*H
      L=L+L
      SUM=0.
      DO 2 K=1,L-1,2
      SUM=SUM+F(A+REAL(K)*H)
```

```
2       CONTINUE
        R(I,1) = 0.5 * R(I-1,1) + H * SUM
        M = 1
        DO 3 J = 2,I
        M = 4 * M
        R(I,J) = R(I,J-1) + (R(I,J-1) - R(I-1,J-1))/REAL(M-1)
3       CONTINUE
        WRITE(10,5)(R(I,J),J = 1,I)
        IF(ABS(R(I,J) - R(I,J-1)),LE,ER)RETURN
4       CONTINUE
        CLOSE(10)
5       FORMAT(5X,5E14.7)
        RETURN
        END

        FUNCTION F(X)
        F = SIN(X)
        RETURN
        END

        EXTERNAL F
        DIMENSION R(5,5)
        DATA ER,A,B,N/1.E-5,0.0,3.14159,5/
        CALL ROMBRG(F,A,B,N,R,ER)
        STOP
        END
```

```
0.3982251E-05
0.1570797E+01  0.2094395E+01
0.1896119E+01  0.2004560E+01  0.1998571E+01
0.1974232E+01  0.2000269E+01  0.1999983E+01  0.2000006E+01
0.1993570E+01  0.2000017E+01  0.2000000E+01  0.2000000E+01  0.2000000E+01
```

三、四阶龙格-库塔方法解常微分方程的初值问题

1. 程序要点

对初值问题

$$x' = F(x,t)$$
$$x(0) = S$$

四阶龙格-库塔方法：

$$x(t+h) = x(t) + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$

其中：

$$F_1 = hF(t,x)$$
$$F_2 = hF\left(t + \frac{h}{2}, x + \frac{1}{2}F_1\right)$$
$$F_3 = hF\left(t + \frac{h}{2}, x + \frac{1}{2}F_2\right)$$
$$F_4 = hF(t+h, x+F_3)$$

2. 子程序 RK4(F,T,X,H,NSTEP)中变量说明

F:初值问题的导函数,以函数子程序给出;

T:自变量的值;

X:T处的函数值;

H:步长;

NSTEP:计算步数;

3. 例子:计算

$$\begin{cases} x' = 2 + (x - t - 1)^2 \\ x(1) = 2 \end{cases}$$

在[1,1.5625]上的值,步长为 H = 1/128,步数为 NSTEP = 72

4. 程序

例子的结果是每两步打印一个值.

```
SUBROUTINE RK4(F,T,X,H,NSTEP)
OPEN(1,FILE = 'OUT.DAT')
WRITE(1,3)T,X
H2 = 0.5 * H
START = T
DO 2 K = 1,NSTEP
```

```
      F1 = H * F(T,X)
      F2 = H * F(T + H2, X + 0.5 * F1)
      F3 = H * F(T + H2, X + 0.5 * F2)
      F4 = H * F(T + H, X + F3)
      X = X + (F1 + F2 + F2 + F3 + F3 + F4)/6.0
      T = START + H * REAL(K)
      IF(MOD(K,2),EQ,0)THEN
      WRITE(1,3)T,X
      END IF
2     CONTINUE
      CLOSE(1)
3     FORMAT(5X,2E22.14)
      RETURN
      END

      EXTERNAL F
      DATA T/1.0/,X/2.0/,H/7.8125E-3/,NSTEP/72/
      CALL RK4(F,T,X,H,NSTEP)
      STOP
      END

      FUNCTION F(T,X)
      F = 2.0 + (X - T - 1.0) * * 2
      RETURN
      END
```

```
0.10000000000000E+01    0.20000000000000E+01
0.10156250000000E+01    0.20312514305115E+01
0.10312500000000E+01    0.20625104904175E+01
0.10468750000000E+01    0.20937848091125E+01
0.10625000000000E+01    0.21250820159912E+01
0.10781250000000E+01    0.21564097404480E+01
0.10937500000000E+01    0.21877760887146E+01
```

0.1109375000000E + 01	0.2219188690185 5E + 01
0.1125000000000E + 01	0.22506554126740E + 01
0.1140625000000E + 01	0.2282184600830 1E + 01
0.1156250000000E + 01	0.23137843608856E + 01
0.1171875000000E + 01	0.23454627990723E + 01
0.1187500000000E + 01	0.23772287368774E + 01
0.1203125000000E + 01	0.24090907573700E + 01
0.1218750000000E + 01	0.24410574436188E + 01
0.1234375000000E + 01	0.24731380939484E + 01
0.1250000000000E + 01	0.25053420066833E + 01
0.1265625000000E + 01	0.25376787185669E + 01
0.1281250000000E + 01	0.25701582431793E + 01
0.1296875000000E + 01	0.26027905941010E + 01
0.1312500000000E + 01	0.26355862617493E + 01
0.1328125000000E + 01	0.26685562133789E + 01
0.1343750000000E + 01	0.27017116546631E + 01
0.1359375000000E + 01	0.27350642681122E + 01
0.1375000000000E + 01	0.27686264514923E + 01
0.1390625000000E + 01	0.28024106025696E + 01
0.1406250000000E + 01	0.28364300727844E + 01
0.1421875000000E + 01	0.28706984519958E + 01
0.1437500000000E + 01	0.29052300453186E + 01
0.1453125000000E + 01	0.29400401115417E + 01
0.1468750000000E + 01	0.29751441478729E + 01
0.1484375000000E + 01	0.30105590820313E + 01
0.1500000000000E + 01	0.30463023185730E + 01
0.1515625000000E + 01	0.30823919773102E + 01
0.1531250000000E + 01	0.31188471317291E + 01
0.1546875000000E + 01	0.31556885242462E + 01
0.1562500000000E + 01	0.31929373741150E + 01

四、三次样条

1. 程序要点

本程序的目的是利用曲线上有限个点的坐标，构造三次插值

样条的光滑曲线去拟合这些数据点。采用的边界条件是第二边界条件。有三个子程序构成：子程序 XISU；子程序 TRI；子程序 SPL3.

(1) 子程序 XISU 形成系数，其要点为：

(a) 对 i = I,N,计算

$$h_i = X(i) - X(i-1)$$

$$b_i = (Y(i) - Y(i-1))/h_i$$

(b) 对 i = 1,N - 1,计算

$$\lambda_i = h_{i+1}/(h_i + h_{i+1})$$

$$\mu_i = h_i/(h_i + h_{i+1})$$

$$d_i = 6(b_{i+1} - b_i)$$

(c) 处理边界：

$$d_1 = d_1 - \mu_1 \cdot y_0''$$

$$d_{N-1} = d_{N-1} - \lambda_{N-1} \cdot y_N''$$

即得三对角方程组的系数：

$$\begin{bmatrix} 2 & \lambda_1 & & & \\ \mu_2 & 2 & \lambda_2 & & \\ & \ddots & \ddots & \ddots & \\ & \mu_{n-2} & 2 & \lambda_{n-2} & \\ & & \mu_{n-1} & 2 & \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \begin{bmatrix} d_1 - \mu_1 y_0'' \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} - \lambda_{n-1} y_n'' \end{bmatrix}$$

(2) 子程序 TRI 解三对角方程组，其要点为：

对三对角方程组

$$\begin{bmatrix} d_1 & c_1 & & & \\ a_1 & d_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-2} & d_{n-1} & c_{n-1} \\ & & & a_{n-1} & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

(a) 消元过程

$$\begin{cases} d_2 \leftarrow d_2 - \left(\dfrac{a_1}{d_1}\right) c_1 \\ b_2 \leftarrow b_2 - \left(\dfrac{a_1}{d_1}\right) b_1 \end{cases}$$

一般
$$\begin{cases} d_i \leftarrow d_i - \left(\dfrac{a_{i-1}}{d_{i-1}}\right) c_{i-1} \\ b_i \leftarrow b_i - \left(\dfrac{a_{i-1}}{d_{i-1}}\right) b_{i-1} \end{cases}, \quad 2 \leqslant i \leqslant n$$

得方程组：

$$\begin{bmatrix} d_1 & c_1 & & & & & \\ & d_2 & c_2 & & & & \\ & & \ddots & \ddots & & & \\ & & & d_i & c_i & & \\ & & & & \ddots & \ddots & \\ & & & & & d_{n-1} & c_{n-1} \\ & & & & & & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

(b) 回代过程：

$$x_n \leftarrow \frac{b_n}{d_n}$$

$$x_{n-1} \leftarrow \frac{b_{n-1} - c_{n-1}x_n}{d_{n-1}}$$

一般
$$x_i \leftarrow \frac{b_i - c_i x_{i+1}}{d_i}, \quad i = n-1, n-2, \cdots, 1$$

(3) 子程序 SPL3 是用来对给定的 x_0 值求三次样条 $S(x_0)$ 的值，其要点为：

(a) 确定包含 x_0 的区间$[x_i, x_{i+1}]$；

(b) 利用三次样条的公式

$$S_i(x_0) = y(i) + (x - x(i))(B_i + (x - x(i))$$

$$\times\left(M_i/2+\frac{1}{6h_i}(x-x_i)(M_{i+1}-M_i)\right)$$

计算 $S_i(x_0)$

2. 例子

取 SINX 在区间[0,1.6875]中 10 个等矩节点,作此 10 个点上的自然三次样条.

3. 程序

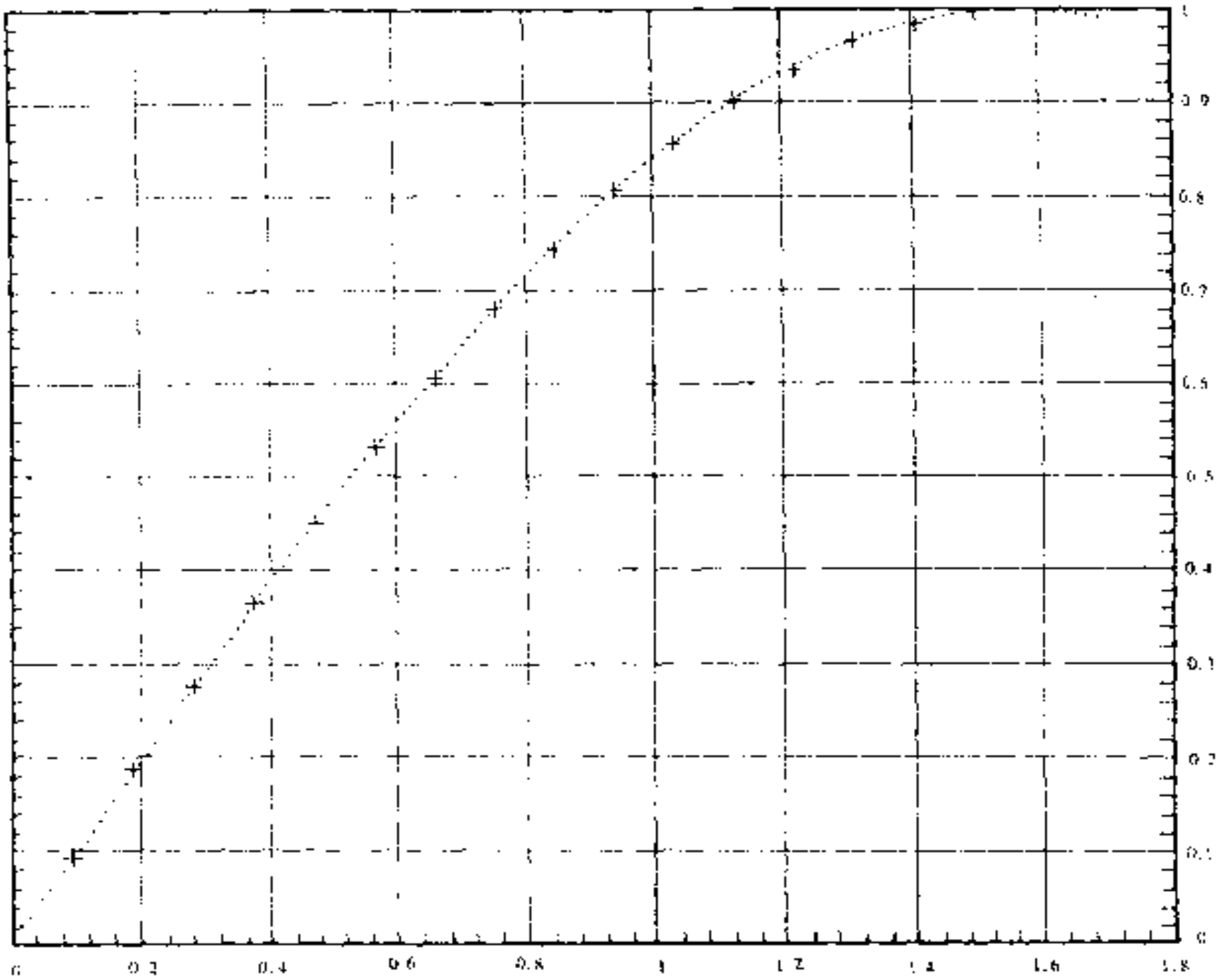
```
      SUBROUTINE TRI(N,A,D,C,B,X)
      DIMENSION A(N),D(N),C(N),B(N),X(N)
      DO 2 I=2,N
      XMULT=A(I-1)/D(I-1)
      D(I)=D(I)-XMULT*C(I-1)
      B(I)=B(I)-XMULT*B(I-1)
2     CONTINUE
      X(N)=B(N)/D(N)
      DO 3 I=N-1,1,-1
      X(I)=(B(I)-C(I)*X(I+1))/D(I)
3     CONTINUE
      RETURN
      END

      SUBROUTINE XISU(N,X,Y,H,B,RMTA,RMU,D,SMO,SMN)
      DIMENSION X(0:N),Y(0:N),H(N),B(N),RMTA(N-1),RMU(N-1),D(N-1)
      DO 2 I=1,N
      H(I)=X(I)-X(I-1)
      B(I)=(Y(I)-Y(I-1))/H(I)
2     CONTINUE
      DO 3 I=1,N-1
      RMTA(I)=H(I+1)/(H(I)+H(I+1))
      RMU(I)=H(I)/(H(I)+H(I+1))
      D(I)=6.*(B(I+1)-B(I))
3     CONTINUE
      D(1)=D(1)-RMU(1)*SMO
```

```
      D(N-1)=D(N-1)-RMTA(N-1)*SMN
      RETURN
      END

      FUNCTION SPL3(N,X,Y,SM,XO)
      DIMENSION X(0:N),Y(0:N),SM(0:N)
      DO 2 I=N-1,1,-1
      DIFF=XO-X(I)
      IF(DIFF.GE.0.0)GO TO 3
    2 CONTINUE
      I=0
      DIFF=XO-X(0)
    3 H=X(I+1)-X(I)
      B=(Y(I+1)-Y(I))/H-H*(SM(I+1)+2.0*SM(I))/6.
      P=0.5*SM(I)+DIFF*(SM(I+1)-SM(I))/(6.*H)
      P=B+DIFF*P
      SPL3=Y(I)+DIFF*P
      RETURN
      END

      PARAMETER (N=4)
      DIMENSION X(0:N),Y(0:N),U(N),V(N),RMTA(N-1),RMU(N-1),D(N-1),
     *U1(N-1),V1(N-1),SM(0:N)
      H=1.6875/REAL(N)
      DO 2 I=0,N
      X(I)=REAL(I)*H
      Y(I)=SIN(X(I))
      Y(I)=SIN(X(I))
    2 CONTINUE
      CALL XISU(N,X,Y,U,V,RMTA,RMU,D,0.0,0.0)
      DO 3 I=1,N-2
    3 RMU(I)=RMU(I+1)
      DO 4 I=1,N-1
    4 U1(I)=2.0
```

```
      CALL TRI(N-1,RMU,U1,RMTA,D,V1)
      DO 8 I=1,N-1
8     SM(I)=V1(1)
      SM(O)=0.0
      SM(N)=0.0
      I=4*(N+1)-3
      J=2
      OPEN(10,FILE='OU1.DAT')
      WRITE(10,11)I,J
      DO 5 I=1.4*(N+1)-3
      XO=REAL(I-1)*H*0.25
      YO=SPL3(N,X,Y,SM,XO)
      WRITE(10,6)XO,YO
5     CONTINUE
6     FORMAT(2X,2F22.14)
11    FORMAT(2X.2I10)
      CLOSE(10)
      STOP
      END
```

```
  17          2
0.0000000000000    0.0000000000000
0.10546875000000   0.10491132736206
0.21093750000000   0.20880529284477
0.31640625000000   0.31066456437111
0.42187500000000   0.40947178006172
0.52734375000000   0.50147551298141
0.63281250000000   0.58881694078445
0.73828125000000   0.67090302705765
0.84375000000000   0.74714088439941
0.94921875000000   0.81170284748077
1.05468750000000   0.86852902173996
1.16015625000000   0.91632473468781
1.26562500000000   0.95379549264908
```

1.37109375000000	0.97381323575974
1.47656250000000	0.98511630296707
1.58203125000000	0.99060952663422
1.68750000000000	0.99319785833359



+一自然三次样条曲线上的点;……一要拟合的曲线

五、牛顿法求根(用双精度编程)

1. 程序要点

(1) 置初值 x;

(2) 记录:$x_0 \leftarrow x$

迭代:$x \leftarrow x - F(x)/F'(x)$

(3) 若 $|x - x_0| < \varepsilon$(ε:要求的精度),则程序结束;否则转(2).

2. 例子

(1) 求 $F(x)=x^3-x-1=0$ 的根

对初值 $\begin{cases}x=1.5\\x=0.6\\x=20\\x=120\end{cases}$，进行了验算，均在 17 步内迅速收敛到精确解.

(2) 求 $F(x)=\arctan x=0$ 的根

取初值 $x_0=1.5$，则 $x_1=-1.6941, x_2=2.3211, x_3=-5.1141, x_4=32.2957, x_5=-1575.3, \cdots$

高精确解 $x=0$ 越来越远，牛顿法发散.

3. 程序

```
      DOUBLE PRECISION X,F,G,FX,XO
      F(X)=X*(X*X-1.0D0)-1.0D0
      G(X)=3.0D0*X*X-1.0D0
      X=1.5D0
      OPEN(10,FILE='OU3.DAT')
      WRITE(10,3)
      FX=F(X)
      I=0
2     WRITE(10,4)I,X,FX
      XO=X
      X=X-FX/G(X)
      FX=F(X)
      IF(I.EQ.20)THEN
      GO TO 5
      END IF
      IF(DABS(X-XO).GE.1.0D-20)THEN
      I=I+1
      GO TO 2
      END IF
3     FORMAT(//9X,'N',24X,'X',26X,'F(X)',//)
4     FORMAT(5X,I5,5X,D36.28,5X,E10.3)
```

```
5       CLOSE(10)
        STOP
        END
```

N	X	F(X)
0	0.150000000000000000000000000D+01	0.875E+00
1	0.134782608695652171969413757 3D+01	0.101E+00
2	0.132520039895090691745281219 5D+01	0.206E-02
3	0.132471817399905368685722351 1D+01	0.924E-06
4	0.132471795724478978663682937 6D+01	0.187E-12
5	0.132471795724474605172872543 3D+01	0.138E-15

N	X	F(X)
0	0.600000000000000000000000000D+00	-0.138E+01
1	0.179000000000001639127731323D+02	0.572E+04
2	0.119468023286087809503078460 7D+02	0.169E+04
3	0.798552035193621993064880371 1D+01	0.500E+03
4	0.535690931479546606540679931 6D+01	0.147E+03
5	0.362499603294610127806663513 2D+01	0.430E+02
6	0.250558919010663494467735290 5D+01	0.122E+02
7	0.182012942231947079300880432 1D+01	0.321E+01
8	0.146104410988768272101879119 9D+01	0.658E+00
9	0.133932322426252596080303192 1D+01	0.631E-01
10	0.132491286771865591406822204 6D+01	0.831E-03
11	0.132471799263781476765871048 0D+01	0.151E-06
12	0.132471795724474716931581497 2D+01	0.487E-14
13	0.132471795724474605172872543 3D+01	0.138E-15

N	X	F(X)
0	0.200000000000000000000000000D+02	0.798E+04
1	0.133452877397831528633832931 5D+02	0.236E+04
2	0.891541660672349318861961364 7D+01	0.699E+03
3	0.597285299693780303001403808 6D+01	0.206E+03
4	0.402889002733688083942979574 2D+01	0.604E+02

5	0.276320648093052595853805542OD + 01	0.173E + 02
6	0.197188049673904225230216980OD + 01	0.470E + 01
7	0.153161469929110057651996612 5D + 01	0.106E + 01
8	0.135582862665749236941337585 4D + 01	0.137E + 00
9	0.132558326460723698139190673 8D + 01	0.369E - 02
10	0.132471865418062996119260788 0D + 01	0.297E - 05
11	0.132471795724519859999418258 7D + 01	0.193E - 11
12	0.132471795724474605172872543 3D + 01	0.138E - 15
13	0.132471795724474605172872543 3D + 01	0.138E - 15

N	X	F(X)
0	0.120000000000000000000000000D + 03	0.173E + 07
1	0.800018750434037855255883187 1D + 02	0.512E + 06
2	0.533374133032206217199563980 1D + 02	0.152E + 06
3	0.355625595549530473351478576 7D + 02	0.449E + 05
4	0.237148870906279931962490081 8D + 02	0.133E + 05
5	0.158198939171302139759063720 7D + 02	0.394E + 04
6	0.105619953661150436103343963 6D + 02	0.117E + 04
7	0.706543009362754226662218570 7D + 01	0.345E + 03
8	0.474867239663472771644592285 2D + 01	0.101E + 03
9	0.322828425809644058346748352 1D + 01	0.294E + 02
10	0.225634089577474039047956466 7D + 01	0.823E + 01
11	0.167967658696745708584785461 4D + 01	0.206E + 01
12	0.140378778672362193465232849 1D + 01	0.363E + 00
13	0.132997772717675276100635528 6D + 01	0.225E - 01
14	0.132474355486142147332429885 9D + 01	0.109E - 03
15	0.132471795785533208400011062 6D + 01	0.260E - 08
16	0.132471795724474605172872543 3D + 01	0.138E - 15

```
      DOUBLE PRECISION X, F, G, FX, XO
      F(X) = DATAN(X)
      G(X) = 1.0D0/(1.0D0 + X * X)
      X = 1.5D0
      OPEN(10, FILE = 'OU3.DAT')
      WRITE(10,3)
      FX = F(X)
```

```
      I = 0
2     WRITE(10,4)I, X, FX
      XO = X
      X = X - FX/G(X)
      FX = F(X)
      IF(I.EQ.11)THEN
      GO TO 5
      END IF
      IF(DABS(X - XO).GE.1.0D - 20)THEN
      I = I + 1
      GO TO 2
      END IF
3     FORMAT(//9X,'N',24X,'X',26X,'F(X)',//)
4     FORMAT(5X,I5,5X,D36.28,5X,E10.3)
5     CLOSE(10)
      STOP
      END
```

N	X	F(X)
0	0.150000000000000000000000000D+01	0.983E+00
1	-0.169407960055381938815116882 3D+01	-0.104E+01
2	0.232112696143838804215192794 8D+01	0.116E+01
3	-0.511408783677751354873180389 4D+01	-0.138E+01
4	0.322956839142100165039300918 6D+02	0.154E+01
5	-0.157531695082120450023561716 1D+04	-0.157E+01
6	0.389497600776088517159223556 5D+07	0.157E+01
7	-0.238302889735521450638771057 1D+14	-0.157E+01
8	0.892028016112383157014846801 8D+27	0.157E+01
9	-0.124990459936571344733238220 2D+55	-0.157E+01
11	0.245399463749850466847419738 8+109	0.157E+01
12	-0.945947635034227669239044189 5+217	-0.157E+01

六、牛顿下山法(用双精度编程)

1. 程序要点

(1) 置初值 x;记录:x_0←x;I←0;

(2) λ←1;

(3) 迭代 x←x − λF(x)/F′(x)　(λ:下山因子)

(4) 若|F(x)|≤|F(x_0)|,则

I←I + 1;

若 I < N(N:最大迭代次数),则 x_0←x,转(2)继续;

若 i = N,则迭代结束

(5) 若|F(x)| > |F(x_0)|,则

λ←λ/2;

x←x_0;

转(3)继续.

2. 例子

(1) 求 F(x) = arctanx = 0 的根.

取初值 x_0 = 1.5,在 4 步内收敛到精确解,而牛顿法对初值 x_0 = 1.5 是发散的.

(2) 求 $F(x) = x^3 - x - 1 = 0$ 的根.

取初值 x_0 = 0.6,在 6 步内收敛,比牛顿法收敛速度要快.

3. 程序

```
DOUBLE PRECISION X,F,G,FX,RMIA,XO,FXO
F(X) = DATAN(X)
G(X) = 1.0D0/(1.0D0 + X * X)
X = 1.5D0
I = 0
RMIA = 1.0D0
OPEN(10,FILE = 'OU4.DAT')
WRITE(10,3)
FX = F(X)
WRITE(10,4)I,X,FX,RMIA
FXO = FX
XO = X
DO 5 I = 1,20
```

```
      RMTA = 1.0D0
2     X = X - FX/G(X) * RMTA
      FX = F(X)
      IF(DABS(FX).GT.DABS(FXO))THEN
      RMTA = RMTA/2.0D0
      X = XO
      FX = FXO
      GO TO 2
      END IF
      WRITE(10,4)I,X,FX,RMTA
      XO = X
      FXO = FX
5     CONTINUE
      CLOSE(10)
3     FORMAT(//3X,'N',24X,'X',26X,'F(X)',7X,'RMTA',//)
4     FORMAT(1X,I3,5X,D36.28,5X,E10.3,5X,E8.3)
      STOP
      END
```

N	X	F(X)	RMTA
0	0.150000000000000000000000000D+01	0.983E+00	.100E+01
1	-0.970398002769097089767456054 7D-01	-0.967E-01	.500E+00
2	0.608055212247787229716777801 5D-03	0.608E-03	.100E+01
3	-0.149877953954222433269023895 3D-09	-0.150E-09	.100E+01
4	0.000000000000000000000000000D+00	0.000E+00	.100E+01
5	0.000000000000000000000000000D+00	0.000E+00	.100E+01
6	0.000000000000000000000000000D+00	0.000E+00	.100E+01
7	0.000000000000000000000000000D+00	0.000E-00	.100E+01
8	0.000000000000000000000000000D+00	0.000E+00	.100E+01
9	0.000000000000000000000000000D+00	0.000E+00	.100E+01
10	0.000000000000000000000000000D+00	0.000E+00	.100E+01
11	0.000000000000000000000000000D+00	0.000E+00	.100E+01
12	0.000000000000000000000000000D+00	0.000E+00	.100E+01
13	0.000000000000000000000000000D+00	0.000E+00	.100E+01

14	0.000000000000000000000000D+00	0.000E+00	.100E+01
15	0.000000000000000000000000D+00	0.000E+00	.100E+01
16	0.000000000000000000000000D+00	0.000E+00	.100E+01
17	0.000000000000000000000000D+00	0.000E+00	.100E+01
18	0.000000000000000000000000D+00	0.000E+00	.100E+01
19	0.000000000000000000000000D+00	0.000E+00	.100E+01
20	0.000000000000000000000000D+00	0.000E+00	.100E+01

N	X	F(X)	RMTA
0	0.600000000000000000000000D+00	−0.138E+01	.100E+01
1	0.114062500000000047034835815D+01	−0.657E+00	.313E−01
2	0.136681366159280106425285339 4D+01	0.187E+00	.100E+01
3	0.132627980400831989943981170 7D+01	0.667E−02	.100E+01
4	0.132472022563605953800678253D+01	0.967E−05	.100E+01
5	0.132471795724954109638929367 1D+01	0.204E−10	.100E+01
6	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
7	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
8	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
9	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
10	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
11	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
12	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
13	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
14	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
15	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
16	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
17	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
18	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
19	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01
20	0.132471795724474605172872543 3D+01	0.138E−15	.100E+01

七、超松弛迭代算法解线性方程组

本算法用 SOR 方法解方程组 AX = B. 数组 X 为一组工作单元,开始存放初始向量,接着存放中间数据,最后存放结果. 用

$|p_0| = \max\limits_{1\leqslant i\leqslant n} |\Delta x_i| = \max\limits_{1\leqslant i\leqslant n} |x_i^{(k+1)} - x_i^{(k)}| < eps$(精度要求)控制迭代终止,k 表示迭代次数.

1. 程序要点

(1) $k \leftarrow 0$;

(2) $x_i \leftarrow 0(i = 1,2,\cdots,n)$;

(3) $k \leftarrow k + 1$;

(4) $p_0 \leftarrow 0$;

(5) 对于 $i = 1,2,\cdots,n$

(a) $p \leftarrow \Delta x_i = \omega\left(b_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j - \sum\limits_{j=i}^{n} a_{ij}x_j\right)/a_{ii}$;

(b) 若 $|p| > |p_0|$,则 $p_0 \leftarrow p$;

(c) $x_i \leftarrow x_i + p$;

(6) 若 $|p_0| > eps$,则转(3);

(7) 输出结果 x,k.

2. 子程序名 SOR(A,B,X,N,OMGA,EPS,K)

3. 例子

求方程组 $AX = B$ 的解. 其中

$$A = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 4 \\ 3 \end{bmatrix}$$

$\omega = 1.03$,迭代次数 $k = 6$

$\omega = 1$,迭代次数 $k = 7$

$\omega = 1.5$,迭代次数 $k = 21$

由此,选取合适的松弛因子,在同样精度下,可减少迭代次数.

4. 程序

```
SUBROUTINE SOR(A,B,X,N,OMCA,EPS,K)
DIMENSION A(N,N),B(N),X(N)
K=0
DO 1 I=1,N
X(I)=0.0
```

```
1       CONTINUE
2       PO = 0.0
        K = K + 1
        DO 4 I = 1, N
        S = 0.0
        DO 3 J = 1, N
        S = S + A(I, J) * X(J)
3       CONTINUE
        P = OMGA * (B(I) - S)/A(I, J)
        IF(ABS(P) . GT. ABS(PO))THEN
        PO = P
        END IF
        X(I) = X(I) + P
4       CONTINUE
        IF(ABS(PO) . GT. EPS)THEN
        GO TO 2
        ENDIF
        RETURN
        END

        DIMENSION A(3,3), B(3), X(3)
        DATA  A/4.0, - 1.0,0.0, - 1.0,4.0, - 1.0,0.0, - 1.0,4.0/, B/1.0,4.0, - 3.0/
        DATA OMGA/1.03/, EPS/5.0E - 6/
        CALL SOR(A, B, X,3, OMGA, EPS, K)
        OPEN(10, FILE = 'OU5 . DAT')
        WRITE(10, * )'----------A----------'
        DO 2 I = 1,3
2       WRITE(10M3)(A(I, J), J = 1, 3)
        WRITE(10, * )'----------B----------'
        WRITE(10,3)(B(J), J = 1,3)
        WRITE(10, * )'----------X----------'
        WRITE(10,3)(X(J), J = 1,3)
        WRITE(10,4)K
        WRITE(10,5)OMGA
```

```
3       FORMAT(5X,E14.8,5X,E14.8,5X,E14.8)
4       FORMAT(5X,'K = ',5I)
5       FORMAT(5X,'OMGA = ',E14.8)
        CLOSE(10)
        STOP
        END
```

```
----------A----------
     0.40000000E+01    -.10000000E+01     0.00000000E+00
    -.10000000E+01     0.40000000E+01    -.10000000E+01
     0.00000000E+00    -.10000000E+01     0.40000000E+01
----------B----------
     0.10000000E+01     0.40000000E+01    -.30000000E+01
----------X----------
     0.50000024E+00     0.10000001E+01    -.49999997E+00
     K =          6
    OMGA = 0.10300000E+01

----------A----------
     0.40000000E+01    -.10000000E+01     0.00000000E+00
    -.10000000E+01     0.40000000E+01    -.10000000E+01
     0.00000000E+00    -.10000000E+01     0.40000000E+01
----------B----------
     0.10000000E+01     0.40000000E+01    -.30000000E+01
----------X----------
     0.50000048E+00     0.10000002E+01    -.49999994E+00
     K =          7
    OMGA = 0.10000000E+01

----------A----------
     0.40000000E+01    -.10000000E+01     0.00000000E+00
    -.10000000E+01     0.40000000E+01    -.10000000E+01
     0.00000000E+00    -.10000000E+01     0.40000000E+01
----------B----------
     0.10000000E+01     0.40000000E+01    -.30000000E+01
```

----------X----------

0.49999937E+00　　0.99999928E+00　　-.50000024E+00

K=　　21

OMGA=0.15000000E+01

八、雅可比方法求实对称矩阵的所有特征值和特征向量

1. 程序要点

雅可比方法是求实对称矩阵 A 的所有特征值和特征向量的方法. 开始时 A 存放实对称矩阵,结束时 A 的对角线元素即为特征值. R 开始时存单位矩阵 E,结束时 R 的列向量存放特征向量. 以 $s=\sum_{i\neq j}a_{ij}^2<EPS$(精度要求)控制求解结束.

(1) R←E;

(2) k=0;

(3) $s=\sum_{i\neq j}a_{ij}^2$;

$am=\max_{i\neq j}|a_{ij}|=|a(i_0,j_0)|$;

(4) 若 s>eps,则

(a) $c\leftarrow(a_{i_0i_0}-a_{j_0j_0})/(2a_{i_0j_0})$;

(b) 若 $|c|<10^{-10}$,则

$c_1\leftarrow\cos(\pi/4)$;

$s_1\leftarrow\sin(\pi/4)\cdot sgn(a(i_0,j_0))$;

若 $|c|>10^{-10}$,则

$t\leftarrow sgn(c)/(|c|+\sqrt{c_2+1})$;

$c_1\leftarrow 1/\sqrt{t^2+1}$;

$s_1\leftarrow c_1*t$;

(c) $a_1\leftarrow a(i_0,i_0)$;

$a_2\leftarrow a(j_0,j_0)$;

$a(i_0,i_0)\leftarrow a_1c_1^2+a_2s_1^2+2a(i_0,j_0)c_1s_1$;

$a(j_0,j_0)\leftarrow a_1s_1^2+a_2c_1^2-2a(i_0,j_0)c_1s_1$;

$a(i_0,j_0)\leftarrow 0$;

$a(j_0,i_0)\leftarrow 0$;

(d) $j\neq i_0, j\neq j_0$, 则

$a_1\leftarrow a(i_0,j)$;

$a(i_0,j)\leftarrow a_1c_1+a(j_0,j)s_1$;

$a(j_0,j)\leftarrow a_1s_1+a(j_0,j)c_1$;

(e) 对 $i=1,\cdots,N$,

$r_1\leftarrow r(i,i_0)$;

$r_2\leftarrow r(i,j_0)$;

$r(i,i_0)\leftarrow r_1c_1+r_2s_1$;

$r(i,j_0)\leftarrow -r_1s_1+r_2c_1$;

(f) $k=k+1$, 转(4);

(5) 若 $s\leqslant eps$, 则结束.

2. 例子

求下列矩阵的特征值及特征向量:

(1) $$\mathbf{A}=\begin{bmatrix}3.5 & -6 & 5\\ -6 & 8.5 & -9\\ 5 & -9 & 8.5\end{bmatrix}$$

(2) $$\mathbf{A}=\begin{bmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 2\end{bmatrix}$$

(3) $$\mathbf{A}=\begin{bmatrix}1 & 1 & 0.5\\ 1 & 1 & 0.25\\ 0.5 & 0.25 & 2\end{bmatrix}$$

3. 程序

```
SUBROUTINE JACB(A,N,R,EPS,K)
DIMENSION A(N,N),R(N,N)
DO 1 I=1,N
DO 1 J=1,N
R(I,J)=0.0
```

```
1       CONTINUE
        DO 2 I = 1, N
        R(I,I) = 1.0
2       CONTINUE
        K = 0
7       S = 0.0
        AM = 0.0
        DO 3 I = 1, N
        DO 3 J = 1, N
        IF(I.NE.J)THEN
        S = S + A(I,J) * A(I,J)
        IF(ABS(A(I,J)).GT.AM)THEN
          AM = ABS(A(I,J))
          I0 = I
          J0 = J
        END IF
        END IF
3       CONTINUE
        IF(S.GT.EPS)THEN
        C = (A(I0,I0) - A(J0,J0))/(2.0 * A(I0,J0))
        IF(ABS(C).LT.1.0E - 10)THEN
        C1 = COS(ATAN(1.0))
        S1 = SIGN(C1, A(I0,J0))
        ELSE
        T = SIGN(1./ABS(C) + SQRT(C * * 2 + 1.0)), C)
        C1 = 1.0/SQRT(1.0 + T * * 2)
        S1 = C1 * T
        END IF
        A1 = A(I0,I0)
        A2 = A(J0,J0)
        A(I0,I0) = A1 * C1 * C1 + A2 * S1 * S1 + 2.0 * A(I0,J0) * C1 * S1
        A(J0,J0) = A1 * S1 * S1 + A2 * C1 * C1 - 2.0 * A(I0,J0) * C1 * S1
        A(I0,J0) = 0.0
        A(J0,I0) = 0.0
```

```
      DO 4 J = 1, N
      IF((J.NE.I0).AND.(J.NE.J0))THEN
      A1 = A(I0,J)
      A(I0,J) = A1 * C1 + A(J0,J) * S1
      A(J0,J) = A(J0,J) * C1 - A1 * S1
      ENDIF
4     CONTINUE
      DO 5 I = 1, N
      IF((I.NE.I0).AND.(I.NE.J0))THEN
      A(I,I0) = A(I0,I)
      A(I,J0) = A(J0,I)
      ENDIF
5     CONTINUE
      DO 6 I = 1, N
      R1 = R(I,I0)
      R2 = R(I,J0)
      R(I,I0) = R1 * C1 + R2 * S1
      R(I,J0) = - R1 * S1 + R2 * C1
6     CONTINUE
      K = K + 1
      GO TO 7
      END IF
      RETURN
      END

      DIMENSION A (3,3), R(3,3)
C     DATA A/2.0, -1.,0.0, -1.0,2.0, -1.0,0.0, -1.0,2.0/
      DATA A/3.5, -6.0,5.0, -6.0,8.5, -9.0,5.0, -9.0,8.5/
C     DATA A/1.0,1.0,0.5,1.0,1.0,0.25,0.5,0.25,2.0/
C     DATA A/1.0,2.0,0.0,2., -1.,1.,0.0,1.,3./
      DATA EPS/1.E-10/
      OPEN(10,FILE='OU8.DAT')
      WRITE(10,*)'----------A----------'
      DO 2 I = 1,3
```

```
      WRITE(10,3)(A(I,J),J=1,3)
2     CONTINUE
      CALL JACB(A,3,R,EPS,K)
      WRITE(10,*)'-------EIGENVALUE-------'
      WRITE(10,3)(A(I,I),I=1,3)
      WRITE(10,*)'-------EIGENVECTOR-------'
      DO 4 I=1,3
      WRITE(10,3)(R,(I,J),J=1,3)
4     CONTINUE
      WRITE(10,*)'-------ITRATION STEPS------'
      WRITE(10,5)K
3     FORMAT(2X,3(F14.8,5X))
5     FORMAT(2X,I5)
      CLOSE(10)
      STOP
      END
```

```
-------------A-------------
     3.50000000    -6.00000000     5.00000000
    -6.00000000     8.50000000    -9.00000000
     5.00000000    -9.00000000     8.50000000
-------EIGENVALUE-------
     0.46593031    20.96808243    -0.93401372
-------EIGENVECTOR------
     0.75714153    -0.40719166     0.51081485
    -0.14745581     0.65524483     0.74088526
    -0.63639110    -0.63627762     0.43607029
-------ITRATION STEPS------
     7

-------------A-------------
     2.00000000    -1.00000000     0.00000000
    -1.00000000     2.00000000    -1.00000000
     0.00000000    -1.00000000     2.00000000
```

```
-------EIGENVALUE-------
          3.41421390          0.58578640          2.00000024
-------EIGENVECTOR-------
          0.50000006          0.50000006         -0.70710683
         -0.70710683          0.70710677          0.00000005
          0.50000000          0.50000000          0.70710689
-------ITRATION STEPS------
          8
--------------A--------------
          1.00000000          1.00000000          0.50000000
          1.00000000          1.00000000          0.25000000
          0.50000000          0.25000000          2.00000000
-------EIGENVALUE-------
          1.48012125         -0.01664728          2.53652596
------EIGENVECTOR-------
          0.44428247         -0.72120625          0.53148341
          0.56210798          0.68635041          0.46147329
         -0.69760120          0.09372657          0.71032941
-------ITRATION STEPS------
          6
```

九、最小二乘法拟合数据

最小二乘法是在函数类确定的前提下求最小二乘解. 我们的程序是在一个具体例子下,先求出最二乘解的法方程,然后利用列选主元 GAUSS 消去法,求解法方程,得最小二乘解.

例子:用形如 $a\ln x + b\cos x + ce^x$ 的函数在最小二乘的意义下拟合数据表:

x	0.24	0.65	0.95	1.24	1.73	2.01	2.23	2.52	2.77	2.99
y	0.23	-0.26	-1.10	-0.45	0.27	0.10	-0.29	0.24	0.56	1.00

此问题的法方程:

$$\begin{cases} a\sum_{k=1}^{10}(\ln x_k)^2 + b\sum_{k=1}^{10}(\ln x_k)\cos x_k + c\sum_{k=1}^{10}(\ln x_k)e^{x_k} = \sum_{k=1}^{10} y_k \ln x_k \\ a\sum_{k=1}^{10}(\ln x_k)\cos x_k + b\sum_{k=1}^{10}(\cos x_k)^2 + c\sum_{k=1}^{10}(\cos x_k)e^{x_k} = \sum_{k=1}^{10} y_k \cos x_k \\ a\sum_{k=1}^{10}(\ln x_k)e^{x_k} + b\sum_{k=1}^{10}(\cos x_k)e^{x_k} + c\sum_{k=1}^{10}(e^{x_k})^2 = \sum_{k=1}^{10} y_k e^{x_k} \end{cases}$$

由子程序 XISU(X, Y, N, A, B)形成该方程的系数. 再由列选主元 GAUSS 迭代法子程序 GAUSS 及 SOLVE 求解该法方程,得最小二乘解

$$y = -1.04103\ln x - 1.26132\cos x + 0.03073e^x$$

```
      SUBROUTINE GAUSS(N,A,IA,L,S)
      DIMENSION A(IA,N),L(N),S(N)
      DO 3 I=1,IA
      L(I)=I
      SMAX=0.0
      DO 2 J=1,N
      SMAX=AMAXI(SMAX,ABS(A(I,J)))
2     CONTINUE
      IF(SMAX.LE.1.E-8)THEN
      WRITE(*,*)'DET(A)=0.0,EQUATIONS HAVE NO SLOVES'
      STOP
      END IF
      S(I)=SMAX
3     CONTINUE
      DO 7 K=1,N-1
      RMAX=0.0
        DO 4 I=K,N
        R=ABS(A(L(I),K))/S(L(I))
        IF(R.LE.RMAX)GO TO 4
        J=I
        RMAX=R
4       CONTINUE
      IF (RMAX.LE.1.E-8)THEN
      WRITE(*,*)'DET(A)=0.0,EQUATIONS HAVE NO SLOVES'
```

```
      STOP
      END IF
      LK = L(K)
      L(J) = L(K)
      L(K) = LK
        DO 6 I = K + 1, N
        XMULT = A(L(I), K)/A(LK, K)
          DO 5 J = K + 1, N
          A(L(I), J) = A(L(I), J) - XMULT * A(LK, J)
5         CONTINUE
          A(L(I), K) = XMULT
6         CONTINUE
7       CONTINUE
        RETURN
        END

        SUBROUTINE SOLVE(N, A, IA, L, B, X)
        DIMENSION A (IA, N), L(N), B(N), X(N)
        DO 3 K = 1, N - 1
          DO 2 I = K + 1, N
          B(L(I)) = B(L(I)) - A(L(I), K) * B(L(K))
2         CONTINUE
3       CONTINUE
        X(N) = B(L(N))/A(L(N), N)
        DO 5 I = N - 1, 1, - 1
        SUM = B(L(I))
          DO 4 J = I + 1, N
          SUM = SUM - A(L(I), J) * X(J)
4       CONTINUE
        X(I) = SUM/A(L(I), I)
5       CONTINUE
        RETURN
        END
```

```
      SUBROUTINE XISU(X,Y,N,A,B)
      DIMENSION X(N),Y(N),A(3,3),B(3)
      DO 1 I=1,3
      DO 1 J=1,3
1     A(I,J)=0.0
      DO 2 I=1,3
2     B(I)=0.0
      DO 3 I=1,10
      A(1,1)=A(1,1)+ALOG(X(I))*ALOG(X(I))
      A(1,2)=A(1,2)+ALOG(X(I))*COS(X(I))
      A(1,3)=A(1,3)+ALOG(X(I))*EXP(X(I))
      A(2,2)=A(2,2)+COS(X(I))*COS(X(I))
      A(2,3)=A(2,3)+COS(X(I))*EXP(X(I))
      A(3,3)=A(3,3)+EXP(X(I))*EXP(X(I))
      B(1)=B(1)+Y(I)*ALOG(X(I))
      B(2)=B(2)+Y(I)*COS(X(I))
      B(3)=B(3)+Y(I)*EXP(X(I))
3     CONTINUE
      A(2,1)=A(1,2)
      A(3,1)=A(1,3)
      A(3,2)=A(3,2)
      RETURN
      END
      DIMENSION A(3,3),B(3),X(10),Y(10),L(3),S(3)
      DATA X/0.24,0.65,0.95,1.24,1.73,2.01,2.23,2.52,2.77,2.99/
      DATA Y/0.23,-0.26,-1.1,-0.45,0.27,0.1,-0.29,0.24,0.56,1.0/
      CALL XISU(X,Y,10,A,B)
      OPEN(10,FILE='OU9.DAT')
      WRITE(10,*)'-----COEFICIENT MATRIX-----'
      DO 11 I=1,3
      WRITE(10,6)(A(I,J),J=1,3)
11    CONTINUE
      WRITE(10,*)'-----RIGHT ITEMS-------'
```

```
      WRITE(10,6)(B(I),I=1,3)
      CALL GAUSS(3,A,3,L,S)
      CALL SOLVE(3,A,3,L,B,S)
      FI=0.0
      DO 1 I=1,10
      FI=FI+(S(1)*ALOG(X(I))+S(2)*COS(X(I))+S(3)*EXP(X(I))
     -Y(I))**2
1     CONTINUE
      WRITE(10,2)(X(I),I=1,10)
      WRITE(10,3)(Y(I),I=1,10)
      WRITE(10,4)(S(I),I=1,3)
      WRITE(10,5)FI
2     FORMAT(//,2X,'X',2X,10(F5.2,2X))
3     FORMAT(2X,'Y',2X,10(F5.2,2X),//)
4     FORMAT(2X,'Y=',F8.5,'LN(X)',F8.5,'COS(X)+',F8.5,'EXP(X)')
5     FORMAT(2X,'ERR=',F14.5)
6     FORMAT(2X,3(F14.5,2X))
      CLOSE(10)
      STOP
      END
```

```
-----COEFFICIENT MATRIX-----
        6.79410      -5.34749      63.25889
       -5.34749       5.10842     -49.00859
       63.25889     -49.00859    1002.50647
-----RIGHT ITEMS-------
        1.61627      -2.38271      26.77277
```

x	0.24	0.65	0.95	1.24	1.73	2.01	2.23	2.52	2.77	2.99
y	0.23	-0.26	-1.10	-0.45	0.27	0.10	-0.29	0.24	0.56	1.00

Y = -1.04103LN(X) - 1.26132COS(X) + 0.03073EXP(X)

ERR= 0.92557

参考书目

1. 李庆扬,王能超,易大义编. 数值分析. 武汉:华中工学院出版,1982 年
2. 武汉大学,山东大学计算数学教研室编. 计算数学. 北京:人民教育出版,1979 年
3. 冯康等. 数值分析. 北京:国防工业出版社,1978 年
4. 杨风翔,陆君良. 数值分析. 天津:天津大学出版社,1985 年
5. 易大义,蒋叔豪,李有法. 数值方法. 杭州:浙江科学技术出版社,1984 年
6. K.E.阿特金森著. 匡蛟勋等译. 数值分析引论. 上海:上海科学技术出版社,1986 年
7. 李岳生,黄友谦. 数值逼近. 北京:人民教育出版社,1978 年
8. 李荣华,冯果忱. 微分方程数值解法. 北京:人民教育出版社,1980 年
9. 孙庆新等. 数值分析(上册、下册). 沈阳:东北工学院出版社,
10. 李庆扬,王能超,易大义编. 现代数值分析. 北京:高等教育出版社,1995 年