“十二五”普通高等教育本科国家级规划教材

工程数学

计算方法

（第二版）

张诚坚　何南忠　覃婷婷

高等教育出版社

内容提要

本书是为大学本科生开设计算方法课程而专门编写的一本教科书, 全书共分六章, 内容涉及数值算法的基础知识、非线性方程的数值解法、线性方程组的数值解法、插值与曲线拟合方法、数值积分及常微分方程初值问题的数值解法. 该书以介绍经典数值算法为基础, 同时也适当地引入了现代算法的内容. 书中既注重算法理论的严谨性, 又突出算法设计的原始思想与实现技巧, 并给出了所有常用算法的Matlab程序代码, 从而使算法理论与算法实现形成一体化. 此外, 该书还配备了一定量的习题, 其中有些是算法理论分析题, 有些是上机实验题, 学生完成这些习题有利于其对书本知识的巩固和理解.

本书取材适当, 用语深入浅出, 通俗易懂, 除适合于学生作为教材外, 也可供科技人员和工程技术人员作为参考书.

前　言

随着科学技术的高速发展, 大量复杂的科学计算问题呈现在人们面前, 要完成这些人自身所不能及的工作, 必须借助于计算机这一人类有史以来最伟大的科技发明, 而使计算机有效解决科学计算问题的关键技术是计算方法. 鉴此, 数值计算方法是每一位科研人员和工程技术人员所必备的知识, 也是每一位理工科大学生必修的重要课程.

数值计算方法包含十分丰富的内容, 但是作为一门基础课教材, 不可能也不必要面面俱到, 重要的是使读者通过一些典型、通用的数值方法掌握其方法构造的基本思想及其实现技巧, 从而达到触类旁通的功效. 因此，在本书编写过程中，在数值方法理论方面, 我们力求其严谨性, 使读者通读完全书后具备初步的算法分析能力. 在数值方法的有效实现方面，我们着眼于算法构造与算法实现的一体化，进而使学生具备一定的算法开发与应用能力.

本书是在作者编撰的1999年版和2008年版的同名书《计算方法》基础上修订而成的，其目前已入选“十二五普通高等教育本科国家级规划教材”. 修订后的该书在算法理论上加强了其严谨性，在算法实现上补充了所有常用算法的Matlab程序，在算法知识的巩固与提高上更新了每章习题，且特别配备了重要算法的实验题. 全书保留了原书的基本架构，共分六章, 其内容分别为绪论、非线性方程的数值解法、线性方程组的数值解法、插值与曲线拟合方法、数值积分、常微分方程初值问题的数值解法.

在编写本书过程中, 我们得到了高等教育出版社、华中科技大学及校内外许多同行的支持与鼓励, 编者对此深表感谢. 由于编者水平所限, 仓促付梓, 书中必有疏漏之处, 诚望读者指正.

张诚坚　何南忠

2015年9月于武汉华中科技大学

目 录

第一章 绪论 ····· (1)
§1.1 数值算法概论 ····· (1)
§1.2 向量范数 ····· (3)
§1.3 矩阵范数 ····· (5)
§1.4 差分方程 ····· (8)
§1.5 误差 ····· (11)
§1.6 Richardson外推法 ····· (12)
习题一 ····· (14)
第二章 非线性方程的数值解法 ····· (15)
§2.1 二分法 ····· (15)
§2.2 弦截法 ····· (16)
§2.3 Picard迭代法 ····· (19)
§2.4 Aitken加速迭代法 ····· (21)
§2.5 Newton迭代法 ····· (22)
§2.6 Newton迭代法的推广与改进 ····· (24)
§2.7 迭代法的收敛阶 ····· (25)
习题二 ····· (28)
第三章 线性方程组的数值解法 ····· (29)
§3.1 Gauss消元法 ····· (29)
§3.2 Doolitle分解法 ····· (32)
§3.3 Cholesky分解法 ····· (35)
§3.4 追赶法 ····· (38)
§3.5 扰动分析 ····· (40)
§3.6 一般单步迭代法 ····· (42)
§3.7 Jacobi迭代法 ····· (44)
§3.8 Gauss-Seidel迭代法 ····· (46)
§3.9 JOR迭代法 ····· (46)
§3.10 SOR迭代法 ····· (48)
习题三 ····· (50)
第四章 插值与曲线拟合方法 ····· (52)
§4.1 Lagrange插值 ····· (52)
§4.2 分段线性插值 ····· (55)
§4.3 Newton插值公式 ····· (57)
§4.4 Hermite插值公式 ····· (61)
§4.5 样条插值 ····· (65)
§4.6 曲线拟合方法 ····· (68)

习题四 ····· (72)
第五章 数值积分 ····· (74)
§5.1 机械求积公式 ····· (74)
§5.2 代数精度法 ····· (75)
§5.3 插值求积法 ····· (76)
§5.4 Newton-Cotes公式及其复合求积法 ····· (78)
§5.5 变步长求积公式 ····· (81)
§5.6 Gauss求积公式 ····· (84)
习题五 ····· (87)
第六章 常微分方程初值问题的数值解法 ····· (89)
§6.1 θ−方法 ····· (89)
§6.2 线性多步法 ····· (90)
§6.3 一般Runge-Kutta方法 ····· (92)
§6.4 显式Runge-Kutta方法 ····· (94)
§6.5 隐式Runge-Kutta方法 ····· (98)
§6.6 隐式方法的有效实现 ····· (101)
§6.7 一般多步法 ····· (106)
§6.8 刚性问题的数值处理 ····· (111)
习题六 ····· (115)
参考文献 ····· (117)
习题答案 ····· (118)

第一章　绪　论

科学技术发展到今天，电子计算机的应用已渗透到自然科学、社会科学、工程技术及日常生活等各个领域.其中，数值计算方法是电子计算机处理实际问题的一种关键手段，其与理论分析、科学实验一起成为当今探索现实世界的三大工具，高效的计算方法与高性能计算机的研究成为同等重要. 从宏观天体运动学到微观分子细胞学说，从工程系统到非工程系统，无一能离开数值计算方法.数值计算这门学科的诞生，使科学发展产生了巨大飞跃，它使各科学领域从定性分析阶段走向定量分析阶段，从粗糙走向精密. 由此可见，数值计算方法是现代每一位从事科学研究与应用的人不可缺少的知识. 本章将主要介绍数值算法的预备知识及其基本思想，为相继章节的学习奠定一个良好的基础.

§1.1　数值算法概论

当利用计算机求解一个实际问题时，其主要步骤如下：

$$\boxed{\text{实际问题}} \Rightarrow \boxed{\text{建立数学模型}} \Rightarrow \boxed{\text{构造数值算法}} \Rightarrow \boxed{\text{编程上机}} \Rightarrow \boxed{\text{获取近似结果}}.$$

由此可知，**数值算法**是利用计算机求解数学问题近似解的方法.其中，所获近似解也称为原问题的**数值解**或**逼近解**.当构造一个数值算法时，它既要面向数学模型，使算法能尽可能地仿真原问题；同时，它也要面向计算机及其程序设计，要求算法具递推性、简洁性及必要的准确性，使其能借助于计算机最终在尽可能少的时间内获得合符原问题精度要求的数值解.

例1.1　计算积分

$$I_n = \int_0^1 \frac{x^n}{x+5}dx, \quad n = 0, 1, 2, ..., 30.$$

解　通过直接计算可产生递推关系

$$I_n = -5I_{n-1} + \frac{1}{n}, \quad I_0 = \ln\frac{6}{5} \approx 1.8232e-001. \tag{1.1}$$

由经典微积分知识可推得I_n具如下性质：

(1) $I_n > 0$且单调递减;　(2) $\lim\limits_{n\to\infty} I_n = 0$;　(3) $\frac{1}{6(n+1)} < I_n < \frac{1}{5(n+1)}$ $(n \geqslant 0)$.

下面我们用两种算法计算I_n:

算法A: 按公式(1.1)自n=1计算到n=30，其产生的计算结果见表1.1.

由上表可见，该算法产生的数值解自$n = 18$开始并未呈现单调递减性质且出现负值和大于1的数，这显然与I_n的固有性质相矛盾，因此本算法所得数值解不符合原问题要求. 究其原因，我们在构造算法时未能充分考虑原积分模型的性态，即由公式(1.1)，其计算从I_{n-1}到I_n 每向前推进一步，其计算值的舍入误差便增长5倍，误差由此积累传播导致最终数值解与原问题真解相悖的结果.为克服这一缺陷，我们改进算法A为：

算法B: 第1步，由性质(3)取

$$I_{30} \approx \frac{1}{2}\left(\frac{1}{6\times 31} + \frac{1}{5\times 31}\right) = 5.9140e-003,$$

表 1.1 自n=1计算到n=30的I_n值

n	1	2	3	4	5
I_n	8.8392e-002	5.8039e-002	4.3139e-002	3.4306e-002	2.8468e-002
n	6	7	8	9	10
I_n	2.4325e-002	2.1233e-002	1.8837e-002	1.6926e-002	1.5368e-002
n	11	12	13	14	15
I_n	1.4071e-002	1.2977e-002	1.2040e-002	1.1229e-002	1.0522e-002
n	16	17	18	19	20
I_n	9.8903e-003	9.3719e-003	8.6960e-003	9.1515e-003	4.2426e-003
n	21	22	23	24	25
I_n	2.6406e-002	-8.6575e-002	4.7635e-001	-2.3401e+000	1.1740e+001
n	26	27	28	29	30
I_n	-5.8664e+001	2.9336e+002	-1.4667e+003	7.3338e+003	-3.6669e+004

第2步，用递推公式

$$I_{n-1} = -\frac{I_n}{5} + \frac{1}{5n}, \tag{1.2}$$

自n=30计算到n=1.由于该算法每向后推进一步，其舍入误差便减少5倍，因此获得合符原积分模型性态的计算结果, 其可参见表1.2.

表 1.2 自n=30计算到n=1的I_n值

n	29	28	27	26	25
I_n	5.4839e-003	5.7998e-003	5.9829e-003	6.2108e-003	6.4501e-003
n	24	23	22	21	20
I_n	6.7100e-003	6.9913e-003	7.2974e-003	7.6314e-003	7.9975e-003
n	19	18	17	16	15
I_n	8.4005e-003	8.8462e-003	9.3419e-003	9.8963e-003	1.0521e-002
n	14	13	12	11	10
I_n	1.1229e-002	1.2040e-002	1.2977e-002	1.4071e-002	1.5368e-002
n	9	8	7	6	5
I_n	1.6926e-002	1.8837e-002	2.1233e-002	2.4325e-002	2.8468e-002
n	4	3	2	1	0
I_n	3.4306e-002	4.3139e-002	5.8039e-002	8.8392e-002	1.8232e-001

对上述例子，我们采用的是由原模型精确解的递推关系来实现计算机求解的，这种求解方

法称为**直接法**.

在大多数情况下，我们只能获得原模型解的近似递推关系，即将连续系统离散化，这种求解方法称为**数值方法**. 作为数值方法的实例，我们考察结构力学、热传导问题中经常出现的两点边值问题 :

$$\begin{cases} y'' + p(x)y' + q(x)y = f(x), \quad x \in (a,b) \\ y(a) = \alpha, \quad y(b) = \beta, \end{cases} \tag{1.3}$$

的数值解法，其中$p(x)$、$q(x)$及$f(x)$是(a,b)上的给定函数，α、β为已知常数，且设问题(1.3)在$[a,b]$上恒有唯一解$y(x)$.其解法步骤如下：

(1) 将区间[a,b]剖分成N等分： 其等分节点为$x_i = a + ih$ $(i = 0, 1, 2, ..., N)$, 步长$h = \frac{b-a}{N}$;

(2) 将问题(1.3)离散化： 由Taylor展式可得

$$\frac{y(x_{i+1}) - y(x_{i-1})}{2h} = y'(x_i) + \mathcal{O}(h^2), \qquad \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} = y''(x_i) + \mathcal{O}(h^2).$$

略去上两式中的余项$\mathcal{O}(h^2)$, 并取$y_i \approx y(x_i)$，则可分别获得下列一阶和二阶中心差商逼近公式:

$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}, \qquad y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}. \tag{1.4}$$

将(1.4)代入(1.3)中得逼近格式

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i \ (i = 1, 2, \ldots, N-1); \quad y_0 = \alpha, \ \ y_N = \beta. \tag{1.5}$$

其中$p_i = p(x_i)$, $q_i = q(x_i)$及$f_i = f(x_i)$. (1.5)实质上是含$N-1$个未知数$y_1, y_2, \cdots, y_{N-1}$ 的线性方程组，解此线性方程组即得数值解$y_1, y_2, \cdots, y_{N-1}$.

§1.2 向量范数

为探讨各种高维问题的算法理论, 本节引入向量范数概念及其相关性质.

定义1.1 称n维实空间R^n上的一个非负函数$\|\cdot\|$为范数，若其满足 :

$$\|x\| = 0 \text{ 当且仅当 } x = 0; \qquad \|\alpha x\| = |\alpha|\|x\|; \qquad \|x + y\| \leqslant \|x\| + \|y\|,$$

其中$\alpha \in R$, $x, y \in R^n$.

下面我们将主要涉及l_p $(p = 1, 2, \cdots)$范数:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}, \qquad x = (x_1, x_2, \cdots, x_n)^T \in R^n.$$

特别，l_∞范数为

$$\|x\|_\infty \triangleq \lim_{p \to \infty} \|x\|_p = \max_{1 \leqslant i \leqslant n} |x_i|.$$

对于R^n上的任意两种范数有如下等价性定理.

定理1.1 若$\|\cdot\|$与$\|\cdot\|'$为 R^n上的任意两种范数，则存在常数$C_2 \geqslant C_1 > 0$使得

$$C_1\|x\| \leqslant \|x\|' \leqslant C_2\|x\|, \quad \forall x \in R^n.$$

在范数概念下，我们即可讨论向量序列的收敛性问题.

定义1.2 设有向量序列$\{x^{(k)} \in R^n |\ x^{(k)} = (x_1^{(k)}, x_2^{(k)}, \cdots, x_n^{(k)})^T\}$, 若

$$\lim_{k\to\infty} x_i^{(k)} = x_i, \quad i = 1, 2, \cdots, n,$$

则称序列$\{x^{(k)}\}$收敛于向量$x = (x_1, x_2, \cdots, x_n)^T$.

定理1.2 在空间R^n中，序列$\{x^{(k)}\}$收敛于向量x 的充要条件是存在范数$\|\cdot\|$，使得$\lim\limits_{k\to\infty} \|x^{(k)} - x\| = 0$.

证明 一方面，若序列$\{x^{(k)}\}$收敛于向量x，则$\lim\limits_{k\to\infty} \|x^{(k)} - x\|_\infty = 0$. 另一方面，若存在范数$\|\cdot\|$ 使得$\lim\limits_{k\to\infty} \|x^{(k)} - x\| = 0$, 则由定理1.1，存在常数$C_2 \geqslant C_1 > 0$，使得

$$C_1\|x^{(k)} - x\| \leqslant \|x^{(k)} - x\|_\infty \leqslant C_2\|x^{(k)} - x\|.$$

因此, 由夹逼定理知$\lim\limits_{k\to\infty} \|x^{(k)} - x\|_\infty = 0$, 故$\{x^{(k)}\}$收敛于$x$. ■

设有映射$\Phi:\ D \subset R^n \to R^n$. 在相继章节, 我们常常需要判断非线性方程

$$x = \Phi(x), \quad x \in D \subset R^n \tag{1.6}$$

的解的存在唯一性. 为此, 作为向量范数的一个应用, 我们引入不动点原理. 方程(1.6)的解也称为**映射Φ的不动点**, 其存在性往往可通过其压缩性保障.

定义 1.3 设存在向量空间R^n中的范数$\|\cdot\|$及常数$q \in [0, 1)$，使得映射$\Phi:\ D \subset R^n \to R^n$在集$D_0 \subset D$ 上满足

$$\|\Phi(x) - \Phi(y)\| \leqslant q\|x - y\|, \quad \forall x, y \in D_0, \tag{1.7}$$

则称Φ 为集$D_0 \subset D$ 上的具压缩系数q 的压缩映射.

定理 1.3 (Banach 压缩映照原理) 设$\Phi:\ D \subset R^n \to R^n$ 是闭集$D_0 \subset D$ 上的压缩映射，且$\Phi(D_0) \subset D_0$，则Φ 于D_0 上有唯一不动点x^*.

证明 构造序列

$$x_{k+1} = \Phi(x_k), \quad k = 0, 1, 2, \cdots; \ \ x_0 \in D_0. \tag{1.8}$$

由已知条件$\Phi(D_0) \subset D_0$ 可知：序列$\{x_k\} \subset D_0$.由于Φ 是闭集D_0 上的压缩映射，则对于 R^n 中的任意向量范数$\|\cdot\|$，存在常数$q \in [0, 1)$ 使得

$$\|x_{k+1} - x_k\| = \|\Phi(x_k) - \Phi(x_{k-1})\| \leqslant q\|x_k - x_{k-1}\|$$

$$= q\|\Phi(x_{k-1}) - \Phi(x_{k-2})\| \leqslant q^2\|x_{k-1} - x_{k-2}\| \leqslant \cdots \leqslant q^k\|x_1 - x_0\|.$$

因此，对任意自然数p 有

$$\|x_{k+p} - x_k\| \leqslant \sum_{i=1}^{p} \|x_{k+i} - x_{k+i-1}\| \leqslant \sum_{i=1}^{p} q^{k+i-1}\|x_1 - x_0\| \leqslant \frac{q^k}{1-q}\|x_1 - x_0\|. \tag{1.9}$$

由此及$q \in [0, 1)$可知序列$\{x_k\}$ 是一个 Cauchy 序列.又由$\{x_k\} \subset D_0$ 及D_0 为闭集可知：存在$x^* \in D_0$，使得$\lim\limits_{k\to\infty} x_k = x^*$.既然$\Phi$ 是压缩映射，因此Φ 也是连续映射，从而

$$x^* = \lim_{k\to\infty} x_k = \lim_{k\to\infty} \Phi(x_{k-1}) = \Phi(x^*),$$

即x^* 为Φ 在D_0 上的不动点.若Φ 在D_0 上存在其它不动点y^*，则由压缩性有

$$\|x^*-y^*\|=\|\Phi(x^*)-\Phi(y^*)\|\leqslant q\|x^*-y^*\|<\|x^*-y^*\|.$$

矛盾! 因此，必有$x^*=y^*$.故定理获证. ■

该证明导出了压缩映射Φ 的不动点的一个逼近序列$\{x_k:|\ x_k=\Phi(x_{k-1}),k=1,2,\cdots\}$，其误差估计可由下述定理给出.

定理 1.4 设$\Phi:\ D\subset\ R^n\to\ R^n$ 是闭集$D_0\subset D$ 上具压缩系数$q\in[0,1)$ 的压缩映射，且$\Phi(D_0)\subset D_0$，则迭代序列(1.8) 收敛于Φ 在D_0 上的不动点x^*，且有先验误差估计

$$\|x^*-x_k\|\leqslant\frac{q^k}{1-q}\|x_1-x_0\|,\ \ k=1,2,\cdots \tag{1.10}$$

及后验误差估计

$$\|x^*-x_k\|\leqslant\frac{q}{1-q}\|x_k-x_{k-1}\|,\ \ k=1,2,\cdots. \tag{1.11}$$

证明 由定理1.3 的证明过程可知迭代序列(1.8) 满足$\lim\limits_{k\to\infty}x_k=x^*$，且在(1.9) 中令$p\to+\infty$即得先验误差估计(1.10). 以下证明后验误差估计式.事实上，对于任意自然数p，我们有

$$\begin{aligned}\|x_{k+p}-x_k\|&\leqslant\sum_{i=1}^{p}\|x_{k+i}-x_{k+i-1}\|=\sum_{i=1}^{p}\|\Phi(x_{k+i-1})-\Phi(x_{k+i-2})\|\\ &\leqslant\ \sum_{i=1}^{p}q\|x_{k+i-1}-x_{k+i-2}\|\leqslant\cdots\leqslant\sum_{i=1}^{p}q^i\|x_k-x_{k-1}\|\leqslant\tfrac{q}{1-q}\|x_k-x_{k-1}\|.\end{aligned} \tag{1.12}$$

在上式中令$p\to+\infty$ 即得后验误差估计(1.11). ■

§1.3 矩阵范数

定义1.4 设A为n级方阵，$\|\cdot\|$为 R^n中的某范数，则称

$$\max_{\|x\|=1}\|Ax\|\ \ (x\in\ R^n)$$

为矩阵A的从属于该向量范数的范数，记为$\|A\|$.

利用定义1.4可直接推得其矩阵范数具有如下性质:

(1) 对任意n级方阵A有$\|A\|\geqslant 0$; 且$\|A\|=0$当且仅当$A=0$;

(2) 对任意实数k及任意n级方阵A，有$\|kA\|=|k|\|A\|$;

(3) 对$\forall x\in\ R^n$及任意n级方阵A，有$\|Ax\|\leqslant\|A\|\|x\|$.

(4) 对任意两个n级方阵A,B，有

$$\|A+B\|\leqslant\|A\|+\|B\|,\ \ \|AB\|\leqslant\|A\|\|B\|.$$

性质(3)称为矩阵范数与向量范数之间的**相容性**. 由定义1.4可知，矩阵范数与向量范数之间存在着从属关系. 特别对于从属于l_1、l_2、l_∞范数的矩阵范数，我们可以给出其具体表达式.

定理1.5 设有n级实方阵$A=(a_{ij})$，则从属于l_1、l_2、l_∞范数的矩阵范数分别为:

$$\|A\|_1=\max_{1\leqslant j\leqslant n}\sum_{i=1}^{n}|a_{ij}|,\quad \|A\|_2=\sqrt{\rho(A^TA)},\quad \|A\|_\infty=\max_{1\leqslant i\leqslant n}\sum_{j=1}^{n}|a_{ij}|,$$

其中$\rho(\cdot)$为矩阵的谱半径，其满足$\rho(B)=\max\limits_{1\leqslant i\leqslant n}|\lambda_i^B|$, λ_i^B为方阵B的第i个特征值.

证明 此处仅证l_2范数情形，其余两式类似可证. 由于A^TA为对称非负定阵，则其特征值λ_i $(i=1,2,\cdots,n)$非负，且存在n维实正交阵H，使得$A^TA=H^T\mathrm{diag}(\lambda_i)H$, 其中

$$\mathrm{diag}(\lambda_i)=\begin{bmatrix}\lambda_1 & 0 & \cdots & \cdots & 0\\ 0 & \lambda_2 & 0 & \dots & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & \lambda_{n-1} & 0\\ 0 & \cdots & \cdots & 0 & \lambda_n\end{bmatrix}.$$

$\forall x\in\{x|\ \|x\|_2=1\}$, 若记$y=Hx=(y_1,y_2,\cdots,y_n)^T$, 则有$\|y\|_2^2=x^TH^THx=\|x\|_2^2=1$, 且

$$\|Ax\|_2^2=x^TA^TAx=(Hx)^T\mathrm{diag}(\lambda_i)(Hx)=\sum_{i=1}^{n}\lambda_iy_i^2\leqslant(\max_{1\leqslant i\leqslant n}\lambda_i)\|y\|_2^2.$$

从而$\|Ax\|_2\leqslant\sqrt{\rho(A^TA)}$. 另一方面，若$\rho(A^TA)$对应矩阵$A^TA$的单位特征向量为$\tilde{x}$，则

$$\|A\|_2^2\geqslant\|A\tilde{x}\|_2^2=\tilde{x}^TA^TA\tilde{x}=\tilde{x}^T\rho(A^TA)\tilde{x}=\rho(A^TA)\|\tilde{x}\|_2^2=\rho(A^TA),$$

即$\|A\|_2\geqslant\sqrt{\rho(A^TA)}$. 故$\|A\|_2=\sqrt{\rho(A^TA)}$. ■

定理1.6 设A为n级方阵，则对任意矩阵范数$\|\cdot\|$有$\rho(A)\leqslant\|A\|$.

证明 设λ为阵A的任一特征值，x为其相应的特征向量，则

$$|\lambda|\|x\|=\|\lambda x\|=\|Ax\|\leqslant\|A\|\|x\|,$$

即$|\lambda|\leqslant\|A\|$. 故$\rho(A)\leqslant\|A\|$. ■

此外矩阵范数与谱半径之间还存在如下关系.

定理1.7 $\forall\varepsilon>0$,必存在 $R^{n\times n}$ 中的某范数$\|\cdot\|$，使得

$$\|A\|\leqslant\rho(A)+\varepsilon,\quad A\text{为任意}n\text{级方阵}.$$

记 $R^{n\times m}$ 为全体实$n\times m$级矩阵的集合. 以下，我们讨论矩阵序列的收敛性.

定义1.5 设有矩阵序列$\{A^{(k)}|\ A^{(k)}=(a_{ij}^{(k)})\}\subset R^{n\times m}$，若

$$\lim_{k\to\infty}a_{ij}^{(k)}=a_{ij},\quad i=1,2,\cdots,n;\ j=1,2,\cdots,m,$$

则称矩阵序列$\{A^{(k)}\}$收敛于矩阵$A=(a_{ij})$, 记为$\lim\limits_{k\to\infty}A^{(k)}=A$.

矩阵序列有类似于向量序列的收敛性结果.

定理1.8 设有矩阵序列$\{A^{(k)}\}\subset R^{n\times n}$, 则$\lim\limits_{k\to\infty}A^{(k)}=A$的充要条件是存在矩阵范数$\|\cdot\|$, 使得$\lim_{k\to\infty}\|A^{(k)}-A\|=0$.

进一步, 我们有

定理1.9 $\forall A \in R^{n\times n}$, $\lim\limits_{m\to\infty} A^m = 0$的充要条件是$\rho(A) < 1$.

证明 根据定理1.8，本定理仅需证明：$\lim\limits_{m\to\infty} \|A^m\| = 0$的充要条件是$\rho(A) < 1$. 事实上，一方面, 由定理1.6有

$$[\rho(A)]^m = \rho(A^m) \leqslant \|A^m\| \to 0 \ \ (m \to \infty).$$

从而$\rho(A) < 1$. 另一方面，由于$\rho(A) < 1$，则存在$\varepsilon > 0$，使得$\rho(A) + \varepsilon < 1$. 故由定理1.7, 存在 $R^{n\times n}$中某范数$\|\cdot\|$使得

$$\|A^m\| \leqslant \|A\|^m \leqslant (\rho(A) + \varepsilon)^m \to 0 \ \ (m \to \infty).$$

由此得 $\lim\limits_{m\to\infty} \|A^m\| = 0$. ■

定理1.10 设有n级方阵A及矩阵范数$\|\cdot\|$使得$\|A\| < 1$，则$I - A$非奇异，且有

$$\frac{1}{1 + \|A\|} \leqslant \|(I - A)^{-1}\| \leqslant \frac{1}{1 - \|A\|},$$

其中I为n级单位阵.

证明 由于

$$\rho(A) = \max_{1\leqslant i\leqslant n} |\lambda_i^A| \leqslant \|A\| < 1,$$

则$I - A$非奇异，且$\lim\limits_{m\to\infty} A^m = 0$. 又

$$I + A + A^2 + \cdots + A^k = (I - A)^{-1}(I - A^{k+1}),$$

令$k \to \infty$得

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k.$$

因此

$$\|(I - A)^{-1}\| \leqslant \sum_{k=0}^{\infty} \|A\|^k = \frac{1}{1 - \|A\|}.$$

此外，我们有

$$\|(I - A)^{-1}\| = \|I + A(I - A)^{-1}\| \geqslant 1 - \|A\|\|(I - A)^{-1}\|.$$

由此得

$$\|(I - A)^{-1}\| \geqslant \frac{1}{1 + \|A\|}.$$

故本定理获证. ■

§1.4 差分方程

差分方程可视为微分方程的离散形式，微分方程数值方法主要用差分方程描述. 因此，本节将简要介绍差分方程的部分理论.

设$x:\ [a,b]\to C^d$ 为给定函数，记$t_i=a+ih\in[a,b],\ x_i=x(t_i),$ 则分别称

$$\Delta x_i=x_{i+1}-x_i,\qquad \nabla x_i=x_i-x_{i-1}$$

为$x(t)$ 在节点t_i 处以h 为步长的**一阶向前差分**和**一阶向后差分**, 而符号Δ,∇分别称为**向前差分算子**和**向后差分算子**. 在相继分析中，我们也将用到**恒等算子**$I:\ Ix_i=x_i$ 及**位移算子**$E:\ Ex_i=x_{i+1}$. 由其定义易验证，算子Δ,∇,I,E 均是线性算子且可两两交换. 因此, 这些算子的和的幂可按二项式展开. 此外，各算子之间还存在着一些相互关系，例如：由

$$\Delta x_i=x_{i+1}-x_i=Ex_i-Ix_i=(E-I)x_i,$$

我们有

$$\Delta=E-I. \tag{1.13}$$

类似地，我们可导出

$$\nabla=I-E^{-1}. \tag{1.14}$$

自一阶差分出发，我们可依次定义高阶差分，一般有

$$\triangle^m x_i=\triangle(\triangle^{m-1}x_i),\qquad \nabla^m x_i=\nabla(\nabla^{m-1}x_i).$$

根据该定义及算子关系(1.13) 和(1.14), 我们进一步有

$$\triangle^m x_i=(E-I)^m x_i=\sum_{j=0}^{m}(-1)^j\binom{m}{j}E^{m-j}x_i=\sum_{j=0}^{m}(-1)^j\binom{m}{j}x_{m+i-j}, \tag{1.15}$$

$$\nabla^m x_i=(I-E^{-1})^m x_i=\sum_{j=0}^{m}(-1)^{m-j}\binom{m}{j}E^{-m+j}x_i=\sum_{j=0}^{m}(-1)^{m-j}\binom{m}{j}x_{i+j-m}, \tag{1.16}$$

其中$\binom{m}{j}=\frac{m!}{j!\ (m-j)!}$.

形如

$$F(n;x_n,\Delta x_n,\cdots,\Delta^k x_n)=0,\quad x_n,x_{n+1},\cdots,x_{n+k}\in C^d \tag{1.17}$$

且x_n、x_{n+k} 均显含于(1.17) 中的方程称为k **阶差分方程**. 由(1.15) 可知，(1.17) 也可等价地化为如下方程

$$G(n;x_n,x_{n+1},\cdots,x_{n+k})=0,\quad x_n,x_{n+1},\cdots,x_{n+k}\in C^d. \tag{1.18}$$

在今后数值算法的学习中，我们将主要涉及如下**线性**k**阶差分方程**

$$\sum_{j=0}^{k}a_j(n)x_{n+j}=b_n,\quad n=0,1,2,\cdots, \tag{1.19}$$

其中系数$a_j(n)$、$b_n\in C$，且$a_k(n)a_0(n)\neq 0$. 若给定其k个初始值$x_0,x_1,\cdots,x_{k-1}$，则由(1.19)即可求出其解序列$\{x_n\}$. 特别，若$b_n=0(n=0,1,2,\cdots)$, 则称之为**齐次的**，否则称为**非齐次的**.

线性差分方程具有与线性常微分方程相类似的性质, 以下我们仅列出其主要性质, 其更细致的论述及其它性质可进一步参见文献[4, 5, 6, 7, 8, 9].

定理1.11 若(1.19)为齐次差分方程，$\{x_n^{(i)}\}_{i=1}^m$为该方程的一组特解，则其任意线性组合$\sum\limits_{i=1}^m c_i x_n^{(i)}$仍为该方程的解，其中$c_i$为任意常数.

定理1.12 若(1.19)为齐次差分方程，$\{x_n^{(i)}\}_{i=1}^k$为其线性无关的特解(称为**基本解组**)，则$\sum\limits_{i=1}^k c_i x_n^{(i)}$为该方程的通解.

定理1.13 若(1.19)为齐次差分方程，则其解$\{x_n^{(i)}\}_{i=1}^k$ 线性无关的充要条件是相应的Wronski行列式$\det(W)\neq 0$，其中

$$W=\begin{bmatrix} x_0^{(1)} & x_0^{(2)} & \cdots & x_0^{(k)} \\ x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(k)} \\ \vdots & \vdots & & \vdots \\ x_{k-1}^{(1)} & x_{k-1}^{(2)} & \cdots & x_{k-1}^{(k)} \end{bmatrix}.$$

定理1.14 非齐次差分方程(1.19)的通解可表示为它的任一特解与相应的齐次方程的通解之和.

以上性质可由差分方程解的定义直接获证. 若(1.19)中系数$a_i(n)$均与n无关，则得k阶常系数差分方程

$$\sum_{j=0}^k a_j x_{n+j}=b_n,\quad n=0,1,2,\cdots, \tag{1.20}$$

其中$a_k a_0\neq 0$. 特别取$b_n=0$ $(n=0,1,2,\cdots)$, 即得相应齐次差分方程

$$\sum_{j=0}^k a_j x_{n+j}=0,\quad n=0,1,2,\cdots. \tag{1.21}$$

设(1.21)有形如$x_n=y^n$的解$(y\neq 0)$, 则将其代入(1.21)即得代数方程

$$\sum_{j=0}^k a_j y^j=0. \tag{1.22}$$

我们称(1.22)为方程(1.21)的**特征方程**，而其根称为**特征根.** 因此我们有

定理1.15 $x_n=y^n(n=0,1,2,\cdots)$为齐次方程(1.21)的解当且仅当y为特征根.

定理1.16 若特征方程(1.22)有k个互异根$\{y_i\}_{i=1}^k$, 则(1.21)的通解可表为$x_n=\sum\limits_{i=1}^k c_i y_i^n$, 其中诸$c_i$为任意常数.

定理1.17 若特征方程(1.22)的全体互异根为$\{y_i\}_{i=1}^p$ $(p\leqslant k)$, 其中y_i的重数为m_i，则

$$y_i^n, ny_i^n, \cdots, n^{m_i-1}y_i^n,\quad i=1,2,\cdots,p$$

为(1.21)的基本解组，从而此时(1.21)的通解可表为

$$x_n=\sum_{i=1}^p\sum_{j=0}^{m_i-1} c_{ij}n^j y_i^n,$$

其中诸c_{ij}为任意常数.

定理1.18 在定理1.17的假设条件下，若非齐次差分方程(1.20)有特解x_n^*, 则其通解为

$$x_n = \sum_{i=1}^{p} \sum_{j=0}^{m_i-1} c_{ij} n^j y_i^n + x_n^*.$$

若常系数齐次差分方程初值问题

$$\begin{cases} \sum\limits_{j=0}^{k} a_j x_{n+j} = 0, \quad n = 0, 1, 2, \cdots, \\ x_0 = x_1 = \cdots = x_{k-2} = 0, \quad x_{k-1} = \dfrac{1}{a_k} \end{cases}$$

有解$\tilde{x}_n$，则当取初值$x_0 = x_1 = \cdots = x_{k-1} = 0$时, 相应的非齐次方程(1.20)有特解

$$x_n^* = \sum_{j=0}^{n-k} b_j \tilde{x}_{n-j-1}. \tag{1.23}$$

特别，若诸$b_n \equiv b$，且$\sum\limits_{j=0}^{k} a_j \neq 0$, 则(1.20)的特解也可取为$x_n^* = \frac{b}{\sum\limits_{j=0}^{k} a_j}$.

例1.2 试求解差分方程初值问题

$$\begin{cases} x_{n+4} + 2x_{n+3} + 3x_{n+1} + 2x_{n+1} + x_n = 9, \\ x_0 = x_1 = x_3 = 0, \quad x_2 = -1. \end{cases}$$

解 其差分方程的特征方程为

$$(y^2 + y + 1)^2 = 0,$$

解之得

$$y_{1,\,2} = \cos\frac{2\,\pi}{3} - i\sin\frac{2\,\pi}{3}, \qquad y_{3,\,4} = \cos\frac{2\,\pi}{3} - i\sin\frac{2\,\pi}{3}.$$

因此, 其差分方程对应的齐次方程的通解为

$$\begin{aligned} x_n &= (\hat{c}_1 + \hat{c}_2 n)(\cos\frac{2\,\pi}{3} + i\sin\frac{2\,\pi}{3})^n + (\hat{c}_3 + \hat{c}_4 n)(\cos\frac{2\,\pi}{3} - i\sin\frac{2\,\pi}{3})^n \\ &= (c_1 + c_2 n)\cos\frac{2\,\pi\, n}{3} + (c_3 + c_4 n)\sin\frac{2\,\pi\, n}{3}, \end{aligned}$$

这里, 诸$\hat{c_j}$, c_j 为任意常数. 而原差分方程有特解$x_n = 1$, 则该方程的通解为

$$x_n = (c_1 + c_2 n)\cos\frac{2\,\pi\, n}{3} + (c_3 + c_4 n)\sin\frac{2\,\pi\, n}{3} + 1.$$

将初始条件代入其中得

$$\begin{cases} c_1 = -1, \\ (c_1 + c_2)\cos\frac{2\,\pi}{3} + (c_3 + c_4)\sin\frac{2\,\pi}{3} = -1 \\ c_1 + 3c_2 = -1, \\ (c_1 + 2c_2)\cos\frac{4\,\pi}{3} + (c_3 + 2c_4)\sin\frac{4\,\pi}{3} = -2. \end{cases}$$

解之得

$$c_1 = -1,\ \ c_2 = 0,\ \ c_3 = -\frac{11\sqrt{3}}{3},\ \ c_4 = \frac{8\sqrt{3}}{3}.$$

故其初值问题的解为

$$x_n = -\cos\frac{2\,\pi\, n}{3} - \frac{\sqrt{3}}{3}(11-8n)\sin\frac{2\,\pi\, n}{3} + 1. \quad \blacksquare$$

§1.5 误　差

利用数值方法求得的数值解是一个近似结果，因此总给人们一种不严格的感觉. 其实，在现实世界各种问题的求解过程中，误差产生是绝对的，而所谓的精确是相对的. 当然，数值解的近似程度必须是原问题所容许的、合理的，否则，其计算结果将毫无意义. 误差产生的原因是多样的，就数学建模而言，模型的原始数据是通过观测获得的，因而有观测误差，数学建模时有可能忽略问题的一些次要因素，因而产生模型误差. 就模型计算而言，其算法往往是通过截断局部误差而得，因而有离散误差. 此外，算法在计算机上运行过程中又产生舍入误差. 为简单见，在本课程的学习中，我们仅考虑数值计算带来的误差.

在数值计算中，首先由于对模型离散化而产生一个近似公式，因而有所谓的截断误差，如: 函数$f(x)=\ln(1+x)$有Taylor展开式

$$f(x) = \sum_{i=1}^{n}\frac{(-1)^{i-1}}{i}x^i + \frac{(-1)^n x^{n+1}}{(n+1)(1+\theta x)^{n+1}},\ \ 0<\theta<1.$$

若去掉上式中的余项，利用$S_n(x) \triangleq \sum\limits_{i=1}^{n}\frac{(-1)^{i-1}}{i}x^i$ 来近似计算$f(x)$，则会产生截断误差

$$R_n(x) = \frac{(-1)^n x^{n+1}}{(n+1)(1+\theta x)^{n+1}},\ \ 0<\theta<1.$$

其次，由于计算机的字长有限，计算过程中超过界定位的尾数将按四舍五入原则舍入，因而产生舍入误差，算法的这两种误差在每步计算中也许微不足道，但其在整个计算过程中会不断积累和传播，因此有必要考虑其数值结果中的最终误差，即绝对误差和相对误差.

定义1.6 设x^*是某量的精确值，x是其近似值，则称差$\varepsilon \triangleq x^* - x$ 为x的绝对误差.

在定义1.6中, $\varepsilon > 0$意味着x为x^*的不足近似值，$\varepsilon < 0$意味着x为x^*的过量近似值.由此可见$|\varepsilon|$ 标志着x的精确程度.由于问题的精确解往往是未知的，因此要直接确定ε通常是困难的.实际应用中常用满足$|\varepsilon| \leqslant \eta$的较小正数$\eta$来表征绝对误差，而称$\eta$为**绝对误差限.** 若某实际问题中精确解$x^*$的近似值$x$有绝对误差限$\eta$，则记$x = x^* \pm \eta$.

值得注意的是绝对误差概念用于比较不同条件下的近似值精度仍有不足之处，如有甲、乙两个学生分别做满分为100分和150分的试题，甲得90分，乙得139分，显然若从绝对误差角度来衡量两者成绩优劣，则会导致乙比甲成绩差的不合理结果.为什么这种比较不合理呢？原因是乙做150分题才错11分，而甲做100分题就错了10分.人们自然在考虑绝对误差的同时也会将其与原量的精确值大小进行比较. 为此我们引入衡量近似值精度的另一尺度.

定义1.7 在定义1.6的假设条件下，称比值$\varepsilon_r \triangleq \frac{\varepsilon}{x^*}$ 为近似值的相对误差.

一般在同一量或不同量的n个近似值中，$|\varepsilon_r|$小者，x的精度就高.如上例中甲的相对误差为10%, 乙的相对误差约为7.3%，因此乙的成绩优于甲.同样，我们通常用满足不等式$|\varepsilon_r| \leqslant \eta_r$的**相对误差限** η_r来表征相对误差，且在实际应用中常取$\eta_r = \frac{\eta}{|x|}$. 如某商品标注其重量为$35 \pm 0.2kg$, 则由此可知商品重量的绝对误差(限)为0.2kg，相对误差(限)为$\frac{0.2}{35}(\approx 0.57\%)$.

实际工作中，人们也总结出用近似值的有效数字位数来表征其精度，即若其近似值x的绝对误差限是它的某一位的半个单位，则说该近似值准确到这一位，且这一位直到前面第一个非零数字为止的所有数字称为**有效数字**. 如设有圆周率 π 的近似数为$x = 3.142$，由于其绝对误差

$$|\pi - x| = 0.000407\cdots < 0.0005 = \frac{1}{2} \times 10^{-3},$$

则知近似数x有4位有效数字.一般有

定义1.8 若x^*的近似值$x = \pm 0.x_1x_2\cdots x_n \times 10^m$, 其中$x_1 \neq 0$, 诸$x_i \in \{0, 1, 2, \cdots, 9\}, m$为整数，且

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

则称近似值x有p位有效数字或称x准确到10^{m-p}位.

此外，相对误差与有效数字具有下述关系.

定理1.19 若近似值x具有定义1.8中的形式，且有p位有效数字，则其相对误差限

$$\eta_r = \frac{1}{2x_1} \times 10^{-p+1}.$$

一个算法的优劣除需考虑其截断误差、舍入误差及达到预定精度的计算量外，其误差传播也是不容忽视的.关于这点，我们从§1.1中例1.1可见一斑.此外在后面算法的学习中，我们将在线性方程组的数值求解中遇到所谓“病态”问题，在常微分方程数值解法中遇到所谓“刚性”(stiff)”问题，当利用常规算法求解这些问题时，其误差迅速传播，而最终导致数值解严重失真，甚至使计算失败.刻画误差传播状况的概念是数值稳定性，目前针对各种算法的数值稳定性概念形式上呈多样性，但其实质均是指误差传播对计算结果的影响程度，若误差传播是可控的，则称之为**数值稳定**的，否则称之为**数值不稳定.**

§1.6 Richardson外推法

Richardson外推法是实际数值计算中提高数值解精度及其误差估计的常用方法. 记$y(t, h)$为某数值方法以步长h求解某定解问题真解$y(t)$的逼近值.设$y(t, h)$关于h可展开为

$$y(t, h) = y(t) + \sum_{i=1}^{\infty} A_i h^i. \tag{1.24}$$

若将方法的步长减半为$\frac{h}{2}$, 则有

$$y(t, \frac{h}{2}) = y(t) + \sum_{i=1}^{\infty} \frac{A_i}{2^i} h^i. \tag{1.25}$$

取(1.24)、(1.25)的一个线性组合使其右端关于h的一次项系数为零，得

$$2y(t,\frac{h}{2})-y(t,h)=y(t)-\frac{1}{2}A_2h^2-(1-\frac{1}{2^2})A_3h^3+\cdots.$$

由此可见，当取$\tilde{y}(t)=2y(t,\frac{h}{2})-y(t,h)$作为$y(t)$的逼近值时，其误差量级减少为$\mathcal{O}(h^2)$, 即此时截断误差为$\mathcal{O}(h^2)$. 进一步，若$y(t,h)$的截断误差为$\mathcal{O}(h^p)$，则有

$$y(t,h)=y(t)+\sum_{i=p}^{\infty}A_ih^i,\quad y(t,\frac{h}{2})=y(t)+\sum_{i=p}^{\infty}\frac{A_i}{2^i}h^i. \tag{1.26}$$

由上两式可得

$$2^py(t,\frac{h}{2})-y(t,h)=(2^p-1)y(t)-\frac{1}{2}A_{p+1}h^{p+1}+\cdots.$$

从而得$y(t)$的新的逼近值

$$\tilde{y}(t)=\frac{1}{2^p-1}[2^py(t,\frac{h}{2})-y(t,h)], \tag{1.27}$$

其截断误差为$\mathcal{O}(h^{p+1})$. 由此可见, 通过组合先前步的逼近值, 新逼近值$\tilde{y}(t)$的计算精度被提高一阶. 这种方法称为**Richardson外推法**.

利用Richardson外推法也可获得逼近值的误差估计.设所用方法的截断误差为$\mathcal{O}(h^p)$, 并记

$$\varepsilon_h=y(t)-y(t,h),\quad \varepsilon_{\frac{h}{2}}=y(t)-y(t,\frac{h}{2}).$$

据(1.26)有$\varepsilon_{\frac{h}{2}}=\frac{1}{2^p}\varepsilon_h+\mathcal{O}(h^{p+1})$, 将该式代入$\varepsilon_h=y(t)-y(t,h)$得

$$y(t)=y(t,h)+2^p\varepsilon_{\frac{h}{2}}+\mathcal{O}(h^{p+1}).$$

再将上式代入$\varepsilon_{\frac{h}{2}}=y(t)-y(t,\frac{h}{2})$得

$$\varepsilon_{\frac{h}{2}}=\frac{1}{2^p-1}[y(t,\frac{h}{2})-y(t,h)]+\mathcal{O}(h^{p+1}).$$

由此即得方法的截断误差主项

$$\tilde{\varepsilon}_{\frac{h}{2}}=\frac{1}{2^p-1}[y(t,\frac{h}{2})-y(t,h)]. \tag{1.28}$$

由于(1.28)仅是方法的截断误差主项，因此实际估计误差时, 人们保守地以

$$\triangle\ \triangleq\ |y(t,\frac{h}{2})-y(t,h)|$$

作为误差.利用该量可获得实际计算的终止准则：设$\varepsilon>0$为预定精度.若$\triangle>\varepsilon$，则反复将步长折半进行计算，直至$\triangle<\varepsilon$，这时取步长最终折半后的计算结果作为欲求数值解；若$\triangle<\varepsilon$，则反复将步长加倍计算，直至$\triangle>\varepsilon$ 为止，这时取步长最终加倍前的计算结果作为欲求数值解.

习 题 一

1.1 试证明:

(1) $\|x\|_\infty = \max\limits_{1\leqslant i\leqslant n} |x_i|, \quad x = (x_1, x_2, \cdots, x_n)^T \in R^n$;

(2) $\|A\|_\infty = \max\limits_{1\leqslant i\leqslant n} \sum\limits_{j=1}^{n} |a_{ij}|, \quad A = (a_{ij}) \in R^{n\times n}$.

1.2 设$\Phi: D \subset R^n \to R^n$ 是闭集$D_0 \triangleq \{x| \ \|x - x_0\| \leqslant r\} \subset D$ 上的压缩映射，且

$$\|x_0 - \Phi(x_0)\| \leqslant (1 - q)r,$$

其中q 为Φ 的压缩系数. 试证明Φ 于D_0 上有唯一不动点x^*.

1.3 设矩阵$A, B \in R^{n\times n}$，其中A 可逆，且存在矩阵范数$\|\cdot\|$ 使得

$$\|A^{-1}\| \leqslant \alpha, \ \ \|A - B\| \leqslant \beta, \ \ \alpha\beta < 1.$$

试证明矩阵B 也可逆，且其逆矩阵满足估计

$$\|B^{-1}\| \leqslant \frac{\alpha}{1 - \alpha\beta}.$$

1.4 求解差分方程初值问题

$$\begin{cases} 2x_{n+4} - 5x_{n+3} + 5x_{n+1} - 2x_n = 2, \\ x_0 = 0, \ \ x_1 = 11, \ \ x_2 = -8, \ \ x_3 = 6. \end{cases}$$

1.5 若以$\frac{355}{113}$作为圆周率 π 的逼近值，问此逼近值具有多少位有效数字?

1.6 若$T(h)$逼近其精确值T的截断误差为

$$R(T) \triangleq T(h) - T = \sum_{i=1}^{\infty} A_i h^{2i},$$

其中，系数A_i与h无关. 试证明由

$$\begin{cases} T_0(h) = T(h), \\ T_m(h) = \dfrac{4^m T_{m-1}(\frac{h}{2}) - T_{m-1}(h)}{4^m - 1}, \quad m = 1, 2, \cdots \end{cases}$$

所定义的T的逼近序列$\{T_m(h)\}$的误差为

$$T_m(h) - T = \sum_{i=1}^{\infty} A_i^{(m)} h^{2(m+i)},$$

其中诸$A_i^{(m)}$是与h无关的常数.

第二章　非线性方程的数值解法

在许多实际问题中，问题的解决常常归结为解非线性代数方程组

$$F(X)=0, \tag{2.1}$$

这里$X=(x_1,x_2,\cdots,x_n)^T$ $(x_i\in[a_i,b_i])$, $F(X)=(f_1(X),f_2(X),\cdots,f_n(X))^T\in R^n$. 该方程组的标量情形可简化为

$$f(x)=0,\quad x\in[a,b]. \tag{2.2}$$

一般而言, 非线性方程的精确求解是非常困难. 因此, 人们常常需要借助于各种数值方法对其进行求解. 为简单见, 在§ 2.1-§2.5节, 我们将首先介绍标量方程(2.2)的典型求解方法. 而后, 在§2.6节, 我们将标量方程的Newton迭代法拓展应用于向量型方程(2.1).

§2.1　二分法

设函数$f(x)$满足

$$f(x)\in C([a,b]),\quad f(a)f(b)<0, \tag{2.3}$$

则据闭区间上的连续函数性质可知方程(2.2)在$[a,b]$ 内一定有根. 此时, 我们称$[a,b]$为方程(2.2)的有根区间. 进一步, 若方程(2.2)在$[a,b]$内有唯一根, 则称$[a,b]$为方程(2.2)的隔离区间. 为简单见，以下我们不妨假定方程(2.2)在$[a,b]$内有唯一实根x^*.

用分点$a_0=a$, $x_0=\frac{1}{2}(a+b)$, $b_0=b$将区间$[a,b]$二等分，并计算函数值$f(x_0)$. 若$f(x_0)=0$, 则求得实根$x^*=x_0$; 否则, $f(x_0)$与$f(a)$或$f(b)$异号. 若$f(a)f(x_0)<0$, 则根在区间$[a,x_0]$内，取$a_1=a$, $b_1=x_0$; 若$f(x_0)f(b)<0$, 则根在区间$[x_0,b]$内，取$a_1=x_0$, $b_1=b$.

不断重复上述二分步骤，则可得系列隔离区间:

$$[a_0,b_0]\supset[a_1,b_1]\supset[a_2,b_2]\supset\cdots\supset[a_k,b_k]\supset\cdots,$$

其中二分k次后的隔离区间$[a_k,b_k]$的长度为

$$b_k-a_k=\frac{1}{2^k}(b-a). \tag{2.4}$$

若记隔离区间$[a_k,b_k]$的中点$x_k=\frac{1}{2}(a_k+b_k)$, 则由(2.4)有

$$|x^*-x_k|\leqslant\frac{1}{2}(b_k-a_k)=\frac{1}{2^{k+1}}(b-a),\quad k=0,1,2,\ldots. \tag{2.5}$$

由此可知, 随着二分次数k的增加, 序列x_k愈来愈逼近精确解x^*. 因此, 我们可取序列x_k作为精确解x^*的逼近解序列. 若预定精度要求为$\varepsilon>0:\ |x^*-x_k|<\varepsilon$, 则由(2.5)可知, 我们只需

$$k>\frac{\ln(b-a)-\ln(2\varepsilon)}{\ln 2}. \tag{2.6}$$

以上步骤即构成求解方程(2.2)的**二分法**, 其计算代码如下:

算法 2.1 二分法

```
function x=half(a,b,tol)
c=(a+b)/2; k=1;
m=1+round((log(b-a)-log(2*tol))/log(2));
while k<=m
  if f(c)==0
    c=c;
  return;
  elseif f(a)*f(c)<0
    b=(a+b)/2;
  else
    a=(a+b)/2;
  end
 c=(a+b)/2; k=k+1;
end
k
c
```

例2.1 应用二分法求解方程$x^3 - x - 1 = 0$在区间$[1, 1.5]$内的数值解x_k，要求绝对误差小于10^{-8}.

解 记$f(x) = x^3 - x - 1$. 易知$f(x) \in C([1, 1.5])$，且其满足

$$f(1) = -1 < 0, \quad f(1.5) = \frac{7}{8} > 0.$$

此外，$\forall x \in [1, 1.5]$有$f'(x) = 3x^2 - 1 > 0$, 即$f(x)$ 在区间$[1, 1.5]$上单调递增. 因此，原方程在区间$[1, 1.5]$ 内有唯一根. 运行 Matlab命令$half(1, 1.5, 10^{-8})$得满足精度要求的数值解$x = 1.324717957526445$, 其需迭代27 次. ■

二分方法是方程求根问题的一种直接搜索方法，其优点是算法简单直观，数值解的精度易于判别. 该算法的局限性是仅适用于标量方程，且事先要确定方程的根的一个隔离区间，当隔离区间较长时其计算速度较慢.

§2.2 弦截法

为引入比二分法计算速度更快的弦截法, 我们设方程(2.2) 中函数$f(x)$ 在隔离区间$[a, b]$ 上除满足条件(2.3)外还满足如下条件：存在精确解x^* 的一个邻域$S(x^*, \delta) = \{x : |x - x^*| \leqslant \delta\} \subseteq [a, b]$ 使得$f(x)$ 在该邻域内二阶连续可微且

$$f'(x) \neq 0, \quad Q \triangleq \frac{\max\limits_{x \in S(x^*, \delta)} |f''(x)|\ \delta}{2 \min\limits_{x \in S(x^*, \delta)} |f'(x)|} < 1. \tag{2.7}$$

图 2.1 弦截法的几何图示

今在邻域$S(x^*,\delta)$内任取二点x_0,x_1, 作以$(x_0,f(x_0))$ 和$(x_1,f(x_1))$ 为端点的弦

$$l_1:\quad y=f(x_1)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_1),$$

其与x 轴相交于点

$$x_2=x_1-\frac{x_1-x_0}{f(x_1)-f(x_0)}f(x_1);$$

进一步，作以$(x_1,f(x_1))$ 和$(x_2,f(x_2))$ 为端点的弦

$$l_2:\quad y=f(x_2)+\frac{f(x_2)-f(x_1)}{x_2-x_1}(x-x_2),$$

其与x 轴相交于点

$$x_3=x_2-\frac{x_2-x_1}{f(x_2)-f(x_1)}f(x_2);$$

如此循环往复，可得一列逼近x^* 的点$x_0,x_1,\cdots,x_k,\cdots$(见图 2.1)，其一般表达式为

$$x_{k+1}=x_k-\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}f(x_k),\quad k=1,2,\cdots. \tag{2.8}$$

该公式所表征的求解方法称为**弦截法**或**割线法**. 其计算代码如下：

算法 2.2 弦截法

```
function x=secant(x0,x1,tol)
x2=x1-(x1-x0)*f(x1)/(f(x1)-f(x0)); k=1;
while abs(x2-x1)>=tol
x0=x1; x1=x2;
```

```
x2=x1-(x1-x0)*f(x1)/(f(x1)-f(x0)); k=k+1;
end
x2
k
```

理论分析表明：当条件(2.3)和(2.7) 成立时，由弦截法(2.8) 产生的序列$\{x_k\}$ 收敛于精确解x^*, 且有如下误差估计

$$|x_k - x^*| \leqslant \frac{2\min\limits_{x\in S(x^*,\delta)}|f'(x)|}{\max\limits_{x\in S(x^*,\delta)}|f''(x)|} Q^{\frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^k}. \tag{2.9}$$

例 2.2 应用弦截法求解方程$x^3 - x - 1 = 0$在区间$[1, 1.5]$内的数值解x_k，要求$|x_k - x_{k-1}| < 10^{-8}$.

解 记$f(x) = x^3 - x - 1$. 例 2.1 的求解过程已表明方程$f(x) = 0$ 在区间$[1, 1.5]$内有唯一根. 今取初始迭代点$x_0 = 1$, $x_1 = 1.5$，应用弦截法(2.8) 到该方程得满足精度要求的数值解$x = 1.324717957244753$, 其仅需迭代 6 次. ■

将例2.1 与例2.2相比较，显见弦截法的收敛速度要快于二分法的收敛速度. 弦截法是一种二步迭代方法，即其每个计算步均需前二步的迭代值. 下面，我们设法将其改造为单步迭代法, 其每个计算步仅需前一步的迭代值. 若弦截法产生的迭代序列x_k充分接近于精确解x^*，则$x_k - x_{k-1}$ 与$f(x_k)$ 均充分接近于零. 因此，我们可近似地置$x_k - x_{k-1} \approx f(x_k)$，从而弦截法可改造为**Steffensen 方法**

$$x_{k+1} = x_k - \frac{f^2(x_k)}{f(x_k) - f(x_k - f(x_k))}, \quad k = 1, 2, \cdots. \tag{2.10}$$

该方法是一种单步迭代方法，其计算代码如下：

算法 2.3 Steffensen 方法

```
function x=steffensen(x0,tol)
y0=f(x0); x1=x0-y0^2/(y0-f(x0-y0)); k=1;
while abs(x1-x0)>=tol
x0=x1; y0=f(x0);
x1=x0-y0^2/(y0-f(x0-y0)); k=k+1;
end
x1
k
```

一般而言，Steffensen方法的收敛速度要快于弦截法，但此与初始点x_0的选取有关. 例如：我们若取初始迭代点$x_0 = 1$，应用Steffensen方法到方程$x^3 - x - 1 = 0$，则经7次迭代后可得其在区间$[1, 1.5]$内满足精度要求$|x_k - x_{k-1}| < 10^{-8}$的数值解$x = 1.324717957244746$, 此时Steffensen 方法

的收敛速度要稍慢于弦截法. 若取初始迭代点$x_0 = 1.05$，应用Steffensen方法到方程$x^3-x-1=0$，则经6次迭代后可得其满足同样精度的数值解$x = 1.324717957244746$. 此时，Steffensen 方法的收敛速度要快于弦截法.

§2.3 Picard 迭代法

在本节，我们给出非线性方程的 Picard 迭代法. 为简单见，我们首先考虑标量方程的 Picard 迭代法. 设非线性标量方程(2.2) 可等价地写成

$$x = \varphi(x), \quad x \in [a, b], \tag{2.11}$$

其中$\varphi(x)$为$[a, b]$上的连续函数. 该方程的求根问题在几何上可视为求曲线$y = \varphi(x)$ 与直线$y = x$ 的交点P^* 的横坐标x^*. 因此，我们可从这一几何观点出发来构造求解方程(2.11) 的数值算法.设有根x^* 的初始逼近值x_0, 过曲线$y = \varphi(x)$ 上的点$P_0(x_0, \varphi(x_0))$ 引平行 x 轴的直线，它与直线$y = x$ 相交于点$P_0'(\varphi(x_0), \varphi(x_0))$, 记其横坐标$x_1 = \varphi(x_0)$; 然后过点$P_0'$ 引平行 y 轴的直线，它与曲线$y = \varphi(x)$ 相交于点$P_1(x_1, \varphi(x_1))$; 又过点P_1 引平行 x 轴的直线，它与直线$y = x$ 相交于点$P_1'(\varphi(x_1), \varphi(x_1))$, 记其横坐标$x_2 = \varphi(x_1)$; 进而过点$P_1'$ 引平行 y 轴的直线，它与曲线$y = \varphi(x)$ 相交于点$P_2(x_2, \varphi(x_2))$; 不断重复上述过程, 可得点列$P_1, P_2, \cdots, P_k, \cdots$（见图 2.2），其横坐标由公式

$$x_{k+1} = \varphi(x_k), \quad k = 0, 1, 2, \cdots \tag{2.12}$$

迭代获得. 该迭代过程表明，若迭代函数φ 及初始逼近值x_0 选择恰当，则点列$\{P_k\}_{i=k}^{\infty}$ 逐步逼近P^*，即迭代序列x_k 收敛于x^*(见图 2.2 中左图)；否则，迭代过程发散（见图 2.2 中右图）.由公式(2.12) 确定的方法称为**Picard 迭代法**.

例 2.3 应用 Picard 迭代法求方程$x^3 - x - 1 = 0$在 x=1 附近的数值解x_k，并使其满足$|x_k - x_{k-1}| < 10^{-8}$.

图 2.2 Picard 迭代法的几何图示

解 其方程可写成下列等价形式

$$x = \sqrt[3]{x+1},$$

由此得 Picard 迭代公式

$$x_{k+1} = \sqrt[3]{x_k+1}, \quad k = 0, 1, 2, \cdots, \ x_0 = 1.$$

根据该迭代公式，经 12 次迭代后，可得其方程在 x=1 附近且满足精度要求的近似根x_{11} = 1.324717956483471.

若取其方程的另一等价形式

$$x = x^3 - 1,$$

则有迭代格式

$$x_{k+1} = x_k^3 - 1, \quad k = 0, 1, 2, \cdots, \ x_0 = 1.$$

根据该迭代公式，经 11次迭代后，计算机发生溢出，无法获得满足题设精度要求的近似根. ■

此例表明，尽管一个非线性方程可以有多种 Picard 迭代格式，但是只有那些收敛的迭代格式才是实际计算的有效格式. 为判别其格式的有效性, 我们给出Picard 迭代法的收敛性准则.

定理2.1 设Picard迭代格式(2.12)中的迭代函数$\varphi(x)$满足下列条件 :

(1) 对任意$x \in [a,b]$ 有$a \leqslant \varphi(x) \leqslant b$,

(2) 存在正数$q < 1$, 使对任意$x \in [a,b]$ 有$|\varphi'(x)| \leqslant q < 1$,

则该迭代格式自任意初值$x_0 \in [a,b]$出发均收敛于方程(2.11)的根x^*，且有如下误差估计

$$|x^* - x_k| \leqslant \frac{q}{1-q}|x_k - x_{k-1}|. \tag{2.13}$$

证明 由已知条件及定理1.4, 我们仅需证明函数φ为闭区间$[a,b]$上的压缩映照. 事实上, 据微分中值定理, 对于任意$x, y \in [a,b]$, 存在介于x与y之间的点ξ使得

$$|\varphi(x) - \varphi(y)| = |\varphi'(\xi)(x-y)| \leqslant q|x-y|.$$

因此, 由$q \in [0,1)$可知φ为闭区间$[a,b]$上的压缩映照. 故定理获证. ■

由此可见，在条件(1)-(2)下, 只要前后两次迭代值的差的绝对值足够小, 就可使当前迭代值x_{k+1}达到足够的高精度. 在实际计算中，一般用条件$|x_{k+1} - x_k| < \varepsilon$作为迭代终止准则, 其中$\varepsilon$为方程求解的预定精度. 定理2.1是一个全局收敛性判别准则, 其要求收敛性条件在整个有根区间上成立, 而这对于具体计算格式往往难以保证. 为此, 我们常常代之以探讨计算格式的局部收敛性.

定义2.1 若存在方程(2.11)的精确解x^*的某邻域$U \triangleq \{x| \ |x - x^*| < \delta\}$, 当取初始迭代点$x_0 \in U$时, 其迭代格式(2.12)的逼近序列$x_k$收敛于$x^*$, 则称迭代格式(2.12)是局部收敛的.

由定理2.1可导出迭代格式(2.12)的如下局部收敛性准则.

定理2.2 设迭代函数$\varphi(x)$在方程(2.11)的精确解x^*的某邻域内连续可微，且$|\varphi'(x^*)| < 1$, 则迭代格式(2.12)是局部收敛的.

例2.4 求方程$x = \exp(-x)$在$x = 0.5$附近的数值解x_k，要求精度满足$|x_{k+1} - x_k| < 10^{-5}$.

解 记$f(x)=x-\exp(-x)$, $\varphi(x)=\exp(-x)$. 易验证$f(x)$和$\varphi(x)$均为区间$(-\infty,\infty)$上的无穷次连续可微函数, 且有

$$f(0.5)\approx -0.1065<0,\quad f(0.6)\approx 0.0512>0,\quad f'(x)=1+\exp(-x)>0.$$

因此, 其方程在区间$[0.5,0.6]$内有唯一解x^*. 又

$$|\varphi'(x^*)|\leqslant \max_{0.5\leqslant x\leqslant 0.6}|(e^{-x})'|=\exp(-0.5)<1,$$

则据定理2.2，迭代格式$x_{k+1}=\exp(-x_k)$关于初值$x_0=0.5$是收敛的. 自初值$x_0=0.5$出发, 运行该迭代格式, 经过18次迭代后得满足精度要求的数值解$x_{18}=5.671407632698067e-001$. ■

§2.4 Aitken加速迭代法

Picard迭代法计算格式简单, 但其收敛速度一般较慢. 为提高其收敛速度, 本节考虑改进Picard迭代法. 设x_k为第k次迭代逼近值，迭代函数$\varphi(x)$ 在方程(2.11)的精确解x^*的某邻域内连续可微, 且其导数值变化不大, 记其近似值为l, 则由 Taylor 展开式有

$$x^*-\varphi(x_k)=\varphi(x^*)-\varphi(x_k)=\int_0^1\varphi'(x_k+\theta(x^*-x_k))(x^*-x_k)d\theta\approx l(x^*-x_k).$$

由此得

$$x^*\approx(1-l)^{-1}[\varphi(x_k)-lx_k].$$

故得**加速迭代格式**

$$x_{k+1}=(1-l)^{-1}[\varphi(x_k)-lx_k],\quad k=0,1,\cdots. \tag{2.14}$$

例2.5 应用加速迭代法求方程$x^3-x-1=0$在 x=1 附近的数值解x_k，并使其满足$|x_k-x_{k-1}|<10^{-8}$.

解 记$\varphi(x)=\sqrt[3]{x+1}$, 则方程$x^3-x-1=0$可等价地写成$x=\varphi(x)$. 由例 2.1可知, 该方程 在区间$[1,1.5]$内有唯一根. 取

$$x_0=1,\quad l=\varphi'\left(\frac{1+1.5}{2}\right)=\frac{1}{3\sqrt[3]{(x+1)^2}}\Big|_{x=\frac{5}{4}}\approx 0.2,$$

应用加速迭代公式(2.14) 到其方程, 则迭代 5次后可得满足精度要求的数值解

$$x=1.324717957249321.\quad ■$$

加速迭代格式可加快Picard迭代法的收敛速度，但该方法的缺陷是需要确定参数l，而这对于某些方程而言是十分困难的. 为克服该困难，我们引入如下Aitken 加速迭代法. 记

$$\tilde{x}_{k+1}=\varphi(x_k),\quad \hat{x}_{k+1}=\varphi(\tilde{x}_{k+1}),$$

则由Taylor 展开定理近似地有

$$x^*-\tilde{x}_{k+1}\approx l(x^*-x_k),\quad x^*-\hat{x}_{k+1}\approx l(x^*-\tilde{x}_{k+1}).$$

由上二式消去未知参数l 得

$$x^* \approx \hat{x}_{k+1} - \frac{(\hat{x}_{k+1} - \tilde{x}_{k+1})^2}{\hat{x}_{k+1} - 2\tilde{x}_{k+1} + x_k}.$$

故得**Aitken 加速迭代格式**

$$x_{k+1} = \hat{x}_{k+1} - \frac{(\hat{x}_{k+1} - \tilde{x}_{k+1})^2}{\hat{x}_{k+1} - 2\tilde{x}_{k+1} + x_k}. \tag{2.15}$$

该格式的计算程序如下：

算法 2.4 Aitken 加速迭代法

```
function x=aitken(x0,tol)
x1=g(x0); x2=g(x1);
x3=x2-(x2-x1)^2/(x2-2*x1+x0); k=1;
while abs(x3-x0)>=tol
x0=x3; x1=g(x0); x2=g(x1);
x3=x2-(x2-x1)^2/(x2-2*x1+x0); k=k+1;
end
x3
k
```

例 2.6 应用 Aitken 加速迭代法求方程$x^3 - x - 1 = 0$在 x=1 附近的数值解x_k，并使其满足$|x_k - x_{k-1}| < 10^{-8}$.

解 记$\varphi(x) = \sqrt[3]{x+1}$, 则方程$x^3 - x - 1 = 0$可等价地写成$x = \varphi(x)$. 取$x_0 = 1$, 应用 Aitken 加速迭代格式(2.15)于该方程，则经 3 次迭代后可得满足精度要求的数值解$x_3 = 1.324717957244746$. ■

将例2.3与例2.6 比较，可知Aitken 迭代法进一步加速了 Picard 迭代公式的收敛速度，而且克服了加速迭代格式需要确定参数l的缺陷.

§2.5 Newton 迭代法

Newton 迭代法是一种求解非线性方程的高效方法, 其通过在隔离区间$[a, b]$上不断作曲线$y = f(x)$的切线而获得解的逼近序列. 构造该方法的具体步骤如下: 在方程(2.2) 的解的隔离区间$[a, b]$上选取适当迭代初值x_0，过曲线$y = f(x)$ 的点$(x_0, f(x_0))$ 引切线

$$l_1: \quad y = f(x_0) + f'(x_0)(x - x_0),$$

其与x 轴相交于点

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)};$$

图 2.3 Newton 迭代法的几何图示

进一步，过曲线$y = f(x)$ 的点$(x_1, f(x_1))$ 引切线

$$l_2 : \quad y = f(x_1) + f'(x_1)(x - x_1),$$

其与x 轴相交于点

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)};$$

如此循环往复，可得一列逼近方程(2.2)精确解x^* 的点$x_0, x_1, \cdots, x_k, \cdots$(见图 2.3)，其一般表达式为

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})}, \quad k = 1, 2, \cdots. \tag{2.16}$$

该公式所表述的求解方法称为**Newton 迭代法**或**切线法**, 其计算程序如下:

算法 2.5 Newton 迭代法

```
function x=newton_iteration(x0,tol)
x1=x0-f(x0)/f1(x0); k=1;
while abs(x1-x0)>=tol
x0=x1; x1=x0-f(x0)/f1(x0); k=k+1;
end
x1
k
```

例 2.7 应用Newton 迭代法求方程$x^3 - x - 1 = 0$在 x=1 附近的数值解x_k，并使其满足$|x_k - x_{k-1}| < 10^{-8}$.

解 记$f(x) = x^3 - x - 1$. 由例 2.1 可知方程$f(x) = 0$ 在区间$[1, 1.5]$内有唯一根. 今取初始迭代点$x_0 = 1$，应用Newton 迭代法(2.16) 于该方程, 则迭代 6次后可得满足精度要求的数值解$x_6 = 1.324717957244746$. ■

由该例可见，Newton 迭代法是一种的收敛速度较快的方法.

§2.6 Newton 迭代法的推广与改进

标量方程的 Newton 迭代法也可推广应用于非线性方程组(2.1). 事实上，若记方程组(2.1)的精确解为X^*，其第k 次逼近值为X_k，则由一阶 Taylor 展开式有

$$F(X^*) \approx F(X_k) + F'(X_k)(X^* - X_k).$$

由$F(X^*) = 0$ 及上式得

$$X^* \approx X_k - [F'(X_k)]^{-1}F(X_k).$$

由此获得求解方程组(2.1)的**向量型Newton 迭代公式**

$$X_{k+1} = X_k - [F'(X_k)]^{-1}F(X_k), \quad k = 0, 1, 2, \cdots, \tag{2.17}$$

这里$F'(X)$表示向量函数$F(X)$的Jacobi阵, 即

$$F'(X) = \begin{bmatrix} \frac{\partial f_1(X)}{\partial x_1} & \frac{\partial f_1(X)}{\partial x_2} & \cdots & \frac{\partial f_1(X)}{\partial x_n} \\ \frac{\partial f_2(X)}{\partial x_1} & \frac{\partial f_2(X)}{\partial x_2} & \cdots & \frac{\partial f_2(X)}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n(X)}{\partial x_1} & \frac{\partial f_n(X)}{\partial x_2} & \cdots & \frac{\partial f_n(X)}{\partial x_n} \end{bmatrix}.$$

为简单见, 在下面我们仍称由向量型Newton 迭代公式所表征的方法为**Newton 迭代法**.

在实际计算过程中, 为避免每步计算其Jacobi阵的逆矩阵, 我们常常将公式(2.17)分如下二步运行 :

Step 1 : 求解关于ΔX_k线性方程组$F'(X_k)\Delta X_k = -F(X_k)$;

Step 2 : 计算当前数值解$X_{k+1} = X_k + \Delta X_k$.

向量型Newton 迭代法的计算程序如下 :

算法 2.6 向量型Newton 迭代法

```
function x=newton_iteration(x0,tol)
dx=-f1(x0)\f(x0); x1=x0+dx; k=1;
while norm(x1-x0)>=tol
x0=x1; dx=-f1(x0)\f(x0); x1=x0+dx; k=k+1;
end
x1
k
```

此外, 为节省每步计算 Jacobi 阵$F'(X_k)$ 的时间，我们有时也可采用如下**简化的 Newton 迭代公式**

$$X_{k+1} = X_k - [F'(X_0)]^{-1}F(X_k), \quad k = 0, 1, 2, \cdots. \tag{2.18}$$

该迭代公式的计算实现仅需对算法 2.6稍作修改即可获得. 简化的 Newton 迭代法与标准Newton 迭代法相比较, 尽管其减少了每步的计算量, 但由于前者本身所固有的收敛速度要低于后者的收敛速度, 就实际计算过程而言, 简化的 Newton 迭代法仍往往比标准Newton 迭代法慢.

例 2.8 取迭代初始向量$X_0 = [0,\ 0]^T$，试分别应用 Newton 迭代法及简化的 Newton法求解方程组

$$\begin{cases} \cos x - \sin y = 2x, \\ \sin x + \cos y = 2y, \end{cases} \tag{2.19}$$

并要求所得数值解X_k满足精度要求: $\|X_k - X_{k-1}\|_2 < 10^{-8}$.

解 记

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, \quad F(X) = \frac{1}{2}\begin{bmatrix} 2x - \cos x + \sin y \\ 2y - \sin x - \cos y \end{bmatrix}.$$

自初始点$[0,0]^T$出发, 应用 Newton 迭代法(2.17) 计算方程组(2.19)，经 5 次迭代后获得满足精度要求的数值解

$$X_5 = [0.2290592672028649,\ 0.5418967160206241]^T.$$

当自初始点$[0,0]^T$出发应用简化的Newton 迭代法(2.18) 计算方程组(2.19)时，则需11次迭代方可获得满足精度要求的数值解

$$X_5 = [0.2290592675184943,\ 0.5418967163496685]^T. \quad \blacksquare$$

例2.8 表明，Newton 迭代法是一种收敛速度较快的的方法. 但是, 该方法与先前诸方法有一个共同的缺点，即其要求选择一个恰当的迭代初值X_0 方可保障其计算快速成功运行. 为尽量避免因初值X_0 选择不当而导致其计算过程缓慢收敛或发散，我们在 Newton 迭代公式中引入一个**下山因子**$\lambda:\ 0 < \lambda \leqslant 1$，从而产生下述**Newton 下山法**

$$X_{k+1} = X_k - \lambda[F'(X_k)]^{-1}F(X_k), \quad k = 0, 1, 2, \cdots. \tag{2.20}$$

在实际计算中，λ 可依次取为 $1,\ \frac{1}{2},\ \frac{1}{2^2},\ \cdots,\ \frac{1}{2^k},\ \cdots$，并且采用双精度控制：

$$\|X_{k+1} - X_k\|_2 < \varepsilon_1 \ \text{ 或 } \ \|F(X_k)\|_2 < \varepsilon_2,$$

这里$\varepsilon_1,\ \varepsilon_2$ 为预定精度.

§2.7 迭代法的收敛阶

如前所见, 求解非线性方程有多种迭代方法，但其计算速度却各不相同. 因此, 我们需给出评价其计算速度的指标及其判据. 下面，我们讨论迭代法的收敛阶. 为简单见, 我们仅介绍非线性标量方程迭代法的收敛阶结果.

定义 2.2 设有求解非线性方程(2.2) 的迭代方法，其迭代序列$\{x_k\}$收敛于方程的精确解x^*. 若存在常数$c,\ p > 0$，使得

$$|x^* - x_{k+1}| \leqslant c\,|x^* - x_k|^p, \quad k = 0, 1, \cdots,$$

则称该迭代方法是p 阶收敛的. 特别，当$p=1$ 时，称其为线性收敛的；当$1<p<2$ 时，称其为**超线性收敛的**；当$p=2$ 时，称其为平方收敛的.

对于Picard 迭代法，我们则有如下收敛阶判据.

定理 2.3 设$\varphi(x)$ 在其不动点x^* 的某邻域内p 阶连续可微，且

$$\varphi^{(i)}(x^*)=0 \quad (i=1,2,\cdots,p-1), \qquad \varphi^{(p)}(x^*)\neq 0,$$

则Picard 迭代法(2.12)是p 阶收敛的.

证明 由于φ 在其不动点x^*的某邻域内连续可微，且$\varphi'(x^*)=0$，则据定理2.2，存在x^* 的某邻域$S(x^*,\delta)$，使得迭代格式(2.12) 自任意迭代初值$x_0\in S(x^*,\delta)$ 出发收敛于不动点x^*. 又据Taylor 展开定理及已知条件，存在x^* 的某邻域$S(x^*,\hat{\delta})\supset S(x^*,\delta)$，使得所有$x_k\in S(x^*,\hat{\delta})$，且有

$$\varphi(x_k)=\varphi(x^*)+\sum_{i=1}^{p-1}\frac{\varphi^{(i)}(x^*)}{i!}(x_k-x^*)^i+\int_0^1\frac{(1-\theta)^{p-1}}{(p-1)!}\varphi^{(p)}(x^*+\theta(x_k-x^*))(x_k-x^*)^p d\theta$$

$$= \varphi(x^*)+\int_0^1\frac{(1-\theta)^{p-1}}{(p-1)!}\varphi^{(p)}(x^*+\theta(x_k-x^*))(x_k-x^*)^p d\theta.$$

因此，

$$|x_{k+1}-x^*|=|\varphi(x_k)-\varphi(x^*)|\leqslant\frac{1}{p!}\sup_{x\in S(x^*,\hat{\delta})}|\varphi^{(p)}(x)||x_k-x^*|^p, \quad k=0,1,\cdots.$$

故迭代法(2.12) 是p 阶收敛的. ■

为导出Newton迭代法的收敛阶，我们引入如下Kantorovich 收敛性定理(参见[11]).

定理 2.4 若$f(x)$ 于开区间(a,b) 上连续可微，且存在常数$\alpha,\beta,\gamma>0$及$x_0\in(a,b)$使得对于一切$x,y\in(a,b)$有

$$|f'(x)-f'(y)|\leqslant\alpha|x-y|, \quad |[f'(x)]^{-1}|\leqslant\beta, \quad \eta \triangleq \alpha\beta\gamma<\frac{1}{2}, \quad \gamma \triangleq \left|\frac{f(x_0)}{f'(x_0)}\right|,$$

且$S[x_0,2\gamma] \triangleq \{x|\ |x-x_0|\leqslant 2\gamma\}\subset(a,b)$, 则方程(2.2) 在$S[x_0,2\gamma]$ 内有唯一解x^*，Newton 迭代法(2.16)的解序列x_k收敛于x^*，且其迭代解x_k满足误差估计

$$|x_k-x^*|\leqslant 2\gamma\eta^{2^k-1}, \quad k=0,1,\cdots. \tag{2.21}$$

定理 2.5 在定理2.4 的条件下，Newton迭代法(2.16)是平方收敛的.

证明 据定理2.4, 在$S[x_0,2\gamma]$内, Newton 迭代法(2.16)的解序列x_k 收敛于方程(2.2)的精确解x^*. 此外, 我们有

$$f(x^*)-f(x_k)-f'(x_k)(x^*-x_k)=\int_0^1[f'(x_k+\theta(x^*-x_k))-f'(x_k)](x^*-x_k)d\theta.$$

因此, 由上式及已知条件得

$$|x^*-x_{k+1}|=|x^*-x_k+[f'(x_k)]^{-1}f(x_k)|$$

$$= |[f'(x_k)]^{-1}||f(x^*)-f(x_k)-f'(x_k)(x^*-x_k)|$$

$$\leqslant \alpha\beta\textstyle\int_0^1\theta|x^*-x_k|^2d\theta=\frac{1}{2}\alpha\beta|x^*-x_k|^2.$$

故Newton 迭代法(2.16) 是平方收敛的. ■

对于简化的 Newton 迭代法(2.18)，我们有

定理 2.6 在定理2.4的条件下，简化的Newton迭代法(2.18)是线性收敛的.

证明 记$\psi(x) = x - [f'(x_0)]^{-1}f(x)$, 则方程组$x = \psi(x)$ 与方程组$f(x) = 0$ 等价，且简化的 Newton 迭代法(2.18) 可视为如下 Picard 迭代法

$$x_{k+1} = \psi(x_k), \quad k = 0, 1, \cdots .$$

此外, 对于任意$x, y \in S[x_0, 2\gamma]$有

$$\begin{aligned} & |f(y) - f(x) - f'(x_0)(y - x)| \\ = \ & |\int_0^1 [f'((1-\theta)x + \theta y)) - f'(x_0)](y - x)d\theta| \\ \leqslant \ & \alpha \int_0^1 |x - x_0 + \theta(y - x)|d\theta|x - y| \\ \leqslant \ & \alpha \int_0^1 [(1-\theta)|x - x_0| + \theta|y - x_0|]d\theta|x - y| \\ = \ & \tfrac{1}{2}\alpha(|x - x_0| + |y - x_0|)|x - y|. \end{aligned} \tag{2.22}$$

利用上式及已知条件得

$$\begin{aligned} & |\psi(x) - \psi(y)| = |[f'(x_0)]^{-1}[f(y) - f(x) - f'(x_0)(y - x)]| \\ \leqslant \ & |[f'(x_0)]^{-1}||f(y) - f(x) - f'(x_0)(y - x)| \\ \leqslant \ & 2\alpha\beta\gamma|x - y| = 2\eta|x - y|, \quad \forall x, y \in S[x_0, 2\gamma]; \end{aligned}$$

且有

$$\begin{aligned} & |\psi(x) - x_0| = |[f'(x_0)]^{-1}[f(x) - f(x_0) - f'(x_0)(x - x_0) + f(x_0)]| \\ \leqslant \ & |[f'(x_0)]^{-1}||f(x) - f(x_0) - f'(x_0)(x - x_0)| + |[f'(x_0)]^{-1}f(x_0)| \\ \leqslant \ & \tfrac{1}{2}\alpha\beta|x - x_0|^2 + \gamma \leqslant (2\alpha\beta\gamma + 1)\gamma \leqslant 2\gamma \quad \forall x \in S[x_0, 2\gamma]. \end{aligned}$$

因此，ψ 是闭集$S[x_0, 2\gamma]$ 上的压缩映射，且$\psi(S[x_0, 2\gamma]) \subset S[x_0, 2\gamma]$.故据定理 1.3 及定理 1.4 可知：在$x_0$ 的闭邻域$S[x_0, 2\gamma]$ 内, 方程$x = \psi(x)$ (即$f(x) = 0$) 存在唯一解x^*，且简化的 Newton 迭代法(2.18) 收敛于x^*，其满足

$$|x_{k+1} - x^*| \leqslant 2\eta||x_k - x^*|.$$

因此, 简化的 Newton 迭代法是线性收敛的. ■

类似地，我们还可证明: 在适当条件下, 弦截法(2.8) 是超线性收敛的，Steffensen 方法(2.10) 是平方收敛的. 此外, 上述Picard迭代方法及Newton迭代方法的收敛阶结果也可推广到相应的向量情型(参见[5, 11, 18]).

习 题 二

2.1 使用二分法求方程$x = 2^{-x}$在$[0,1]$内的根, 精确到10^{-8}.

2.2 分别利用弦截法和 Steffensen 方法求方程$2\cos x = 1 + \sin x$ 在$[0, \frac{\pi}{4}]$ 内的近似根, 要求精确到10^{-8}, 并比较二者的计算效率.

2.3 设$\varphi(x)$在闭区间$[a,b]$上一阶连续可微，方程$x = \varphi(x)$ 在$[a,b]$ 内有一根x^*, 且

$$|\varphi'(x) - 3| < 1, \quad \forall x \in [a,b],$$

试构造一个局部收敛于x^* 的迭代公式.

2.4 设$\varphi(x)$在方程$x = \varphi(x)$的根x^*的某邻域$S(x^*,\delta)$内一阶连续可微，且$|\varphi'(x^*)| < 1$, 证明迭代公式$x_{k+1} = \varphi(x_k)$具有局部收敛性.

2.5 已知方程$x + \sin x - 1 = 0$ 在$x_0 = 1/2$ 附近有唯一根，试选择常数a 使得迭代格式

$$x_{k+1} = \frac{ax_k - \sin x_k + 1}{1 + a}$$

在求解其方程时能快速收敛，并用该迭代格式求其方程的根，要求精确到10^{-8}.

2.6 应用 Newton 迭代法求解方程$x = 2\sin(x + \frac{\pi}{3})$ 最小正根，要求精确到10^{-8}.

2.7 设$f(x) = 0$ 有根x^*，其中$f(x)$ 在x^* 的某邻域$S(x^*,\delta)$内二阶连续可微, 且$f'(x^*) \neq 0$，证明 Steffensen 方法是平方收敛的.

2.8 (实验题) 设有非线性方程组

$$\begin{cases} 6x - 2\cos(yz) - 1 = 0 \\ \sqrt{x^2 + \sin z + 1.06} - 9(y + 0.1) = 0 \\ 3\exp(-xy) + 60z + 10\pi - 3 = 0. \end{cases}$$

试分别应用 Picard 迭代法及Newton 迭代法求解该方程组，要求精确到10^{-8}, 并比较二者的计算效率.

第三章　线性方程组的数值解法

考虑线性方程组

$$AX = b, \tag{3.1}$$

其中$A \in R^{n\times n}$为非奇异矩阵，且

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

在科学工程计算中，我们常常需要求解这种线性方程组. 因此，其解法在计算数学中占有举足轻重的地位, 其大致可分为直接法与迭代法两大类. **直接法**是理论上可求得精确解的方法, 如本节将介绍的Gauss 消元法、Doolitle 分解法、Cholesky 分解法等. **迭代法**则是采取逐次逼近的方法，从一个或多个初始量出发，按照一定计算格式获得方程组数值解的方法. 显然，上节介绍的向量型Picard迭代法和Newton迭代法可应用于(3.1), 但基于线性方程组本身的特性，其具有专门的迭代求解方法，如：Jacobi迭代法、JOR迭代法、Gauss-Seidel迭代法、SOR迭代法等.

§3.1　Gauss 消元法

Gauss顺序消元法是一种经典的求解线性方程组的直接方法，该算法的主要思想是首先将方程组(3.1) 化为一个系数阵为下三角阵 (或上三角阵) 的方程组，然后采用前推 (或回代) 方法求得其线性方程组的解. 为利用计算机实现Gauss顺序消元，我们将计算过程中出现的矩阵及其元素进行编号，其计算步骤如下：

记方程组(3.1) 为$A^{(1)}X = b^{(1)}$, 其中

$$A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix} = A, \quad b^{(1)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{bmatrix} = b.$$

假设$a_{11}^{(1)} \neq 0$, 取$m_{i1} = a_{i1}^{(1)}/a_{11}^{(1)}$ $(i = 2, 3, \cdots, n)$, 用该数的负值乘方程组(3.1) 的第 1 个方程，然后将其加到第i 个方程$(i = 2, 3, \cdots, n)$ 上，则依次可消去自第 2 个方程到第n 个方程中的变量x_1，由此得以下等价方程组

$$A^{(2)}X = b^{(2)}, \tag{3.2}$$

其中

$$A^{(2)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}, \quad b^{(2)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{bmatrix},$$

$$a_{ij}^{(2)}=a_{ij}^{(1)}-m_{i1}a_{1j}^{(1)},\quad b_i^{(2)}=b_i^{(1)}-m_{i1}b_1^{(1)},\quad i,j=2,3,\cdots,n;$$

假设$a_{22}^{(2)}\neq 0$，取$m_{i2}=a_{i2}^{(2)}/a_{22}^{(2)}$ $(i=3,4,\cdots,n)$，再用m_{i2} 的负值乘上方程(3.2) 的第2 个方程，然后将其加到第i 个方程$(i=3,4,\cdots,n)$ 上，则依次可消去自第3 个方程到第n 个方程中的变量x_2，得以下等价方程组

$$A^{(3)}X=b^{(3)},\tag{3.3}$$

其中

$$A^{(3)}=\begin{bmatrix}a_{11}^{(1)}&a_{12}^{(1)}&a_{13}^{(1)}&\cdots&a_{1n}^{(1)}\\0&a_{22}^{(2)}&a_{23}^{(2)}&\cdots&a_{2n}^{(2)}\\0&0&a_{33}^{(3)}&\cdots&a_{3n}^{(3)}\\\vdots&\vdots&\vdots&&\vdots\\0&0&a_{n3}^{(3)}&\cdots&a_{nn}^{(3)}\end{bmatrix},\quad b^{(3)}=\begin{bmatrix}b_1^{(1)}\\b_2^{(2)}\\b_3^{(3)}\\\vdots\\b_n^{(2)}\end{bmatrix},$$

$$a_{ij}^{(3)}=a_{ij}^{(2)}-m_{i2}a_{2j}^{(2)},\quad b_i^{(3)}=b_i^{(2)}-m_{i2}b_2^{(2)},\quad i,j=3,4,\cdots,n;$$

重复上述步骤，经$n-1$ 次消元后得以下系数阵为上三角阵的方程组

$$A^{(n)}X=b^{(n)},\tag{3.4}$$

其中

$$A^{(n)}=\begin{bmatrix}a_{11}^{(1)}&a_{12}^{(1)}&\cdots&a_{1n}^{(1)}\\&a_{22}^{(2)}&\cdots&a_{2n}^{(2)}\\&&\ddots&\vdots\\&&&a_{nn}^{(n)}\end{bmatrix},\quad b^{(n)}=\begin{bmatrix}b_1^{(1)}\\b_2^{(2)}\\\vdots\\b_n^{(n)}\end{bmatrix},$$

$$a_{nn}^{(n)}=a_{nn}^{(n-1)}-m_{n,n-1}a_{n-1,n}^{(n-1)},\quad b_n^{(n)}=b_n^{(n-1)}-m_{n,n-1}b_{n-1}^{(n-1)},$$

$$m_{n,n-1}=a_{n,n-1}^{(n-1)}/a_{n-1,n-1}^{(n-1)}.$$

上述过程称为**前推过程**.

在前推过程完成后，我们从方程组(3.4) 的第n 个方程开始，自下而上依次解出$x_n,x_{n-1},\cdots,x_1$，该过程称为**回代过程**，其计算公式如下：

$$x_n=b_n^{(n)}/a_{nn}^{(n)},\quad x_i=\Big(b_i^{(i)}-\sum_{j=i+1}^{n}a_{ij}^{(i)}x_j\Big)/a_{ii}^{(i)},\quad i=n-1,n-2,\cdots,1.\tag{3.5}$$

上述方法即为**Gauss顺序消元法**. 这种消元法的计算可行性前提条件是其消元过程的所有**主元素** $a_{kk}^{(k)}\neq 0$，否则，将导致计算过程无法进行或计算结果严重失真.

例 3.1 利用 Gauss 顺序消元法求解线性方程组

$$\begin{bmatrix}0.001&2.000&3.000\\-1.000&3.712&4.623\\-2.000&1.070&5.643\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1.000\\2.000\\3.000\end{bmatrix}.$$

解 对方程组的增广矩阵进行行变换得

$$(A|b) = \left[\begin{array}{ccc|c} 0.001 & 2.000 & 3.000 & 1.000 \\ -1.000 & 3.712 & 4.623 & 2.000 \\ -2.000 & 1.070 & 5.643 & 3.000 \end{array}\right]$$

$$\Rightarrow \left[\begin{array}{ccc|c} 0.001 & 2.000 & 3.000 & 1.000 \\ 0 & 2004 & 3005 & 1002 \\ 0 & 4001 & 6006 & 2003 \end{array}\right]$$

$$\Rightarrow \left[\begin{array}{ccc|c} 0.001 & 2.000 & 3.000 & 1.000 \\ 0 & 2004 & 3005 & 1002 \\ 0 & 0 & 5.000 & 2.000 \end{array}\right].$$

采用回代法计算上述最后一个矩阵对应的方程组

$$\left[\begin{array}{ccc} 0.001 & 2.000 & 3.000 \\ 0 & 2004 & 3005 \\ 0 & 0 & 5.000 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right] = \left[\begin{array}{c} 1.000 \\ 1002 \\ 2.000 \end{array}\right],$$

得解$X = [0.000,\ -0.09980,\ 0.4000]^T$, 其与具4 位有效数字的解$X^* = [-0.4904,\ -0.05104,\ 0.3675]^T$相比较, 误差为

$$err \triangleq \|X - X^*\|_2 \approx 0.4939.$$

相对于原方程组系数阵中的元素而言，解X 的误差过大. 其原因是在作消元时，用了小主元0.001 作除数，致使后续计算过程产生的元素严重失真，在计算机字长有限的情况下，导致了较大的舍入误差. ■

为避免消元过程中出现零主元或绝对值非常小的主元，下面介绍二种改进的 Gauss 消元法，即**完全选主元法**和**选列主元法**.

设线性方程组(3.1) 经过k 次消元后的方程组对应的增广矩阵为

$$\left[\begin{array}{ccccccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & a_{1,k+1}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & a_{2,k+1}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & \vdots & \vdots & & \vdots & \vdots \\ & & & a_{kk}^{(k)} & a_{k,k+1}^{(k)} & \cdots & a_{k,n}^{(k)} & b_k^{(k)} \\ & & & & a_{k+1,k+1}^{(k+1)} & \cdots & a_{k+1,n}^{(k+1)} & b_{k+1}^{(k+1)} \\ & & & & \vdots & & \vdots & \vdots \\ & & & & a_{n,k+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} & b_n^{(k+1)} \end{array}\right], \tag{3.6}$$

在该矩阵的子块

$$\left[\begin{array}{ccc} a_{k+1,k+1}^{(k+1)} & \cdots & a_{k+1,n}^{(k+1)} \\ \vdots & & \vdots \\ a_{n,k+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} \end{array}\right]$$

中找出绝对值最大的元素，若$|a_{pq}^{(k+1)}| = \max_{k+1\leqslant i,j\leqslant n}|a_{ij}^{(k+1)}|$, 则交换矩阵(3.6) 中的第$k+1$ 行与第p 行、第$k+1$ 列与第q 列，从而将$a_{pq}^{(k+1)}$ 移到主元位置，该过程称为**完全选主元**.以$a_{pq}^{(k+1)}$作为新的主元，消去矩阵(3.6)中的位于$(k+2,k+1),(k+3,k+1),\cdots,(n,k+1)$处的元素，即完成第$k+1$次消元. 重复上述步骤，经$n-1$ 次完全选主元消元后，增广矩阵$(A|b)$ 中的阵A 可化为一个上三角阵，解其对应的方程组即获得原方程组的解.

完全选主元法在计算过程中花费了大量的时间用于寻找主元.事实上，对于某些线性方程组，我们可采用如下选列主元的方法完成消元过程，其将大大地节省搜寻主元的时间.在线性方程组(3.1) 经过k次消元获得增广矩阵(3.6)的基础上，我们在阵(3.6)的子块

$$\begin{bmatrix} a_{k+1,k+1}^{(k+1)} \\ \\ a_{k+2,k+1}^{(k+1)} \\ \vdots \\ a_{n,k+1}^{(k+1)} \end{bmatrix}$$

中找出绝对值最大的元素$a_{p,k+1}^{(k+1)}$：$|a_{p,k+1}^{(k+1)}| = \max_{k+1\leqslant i\leqslant n}|a_{i,k+1}^{(k+1)}|$，然后交换矩阵(3.6) 中的第$k+1$行与第$p$ 行，从而将$a_{p,k+1}^{(k+1)}$ 移到主元位置，该过程称为**选列主元**.以$a_{p,k+1}^{(k+1)}$ 作为新的主元，消去矩阵(3.6) 中的位于$(k+2,k+1),(k+3,k+1),\cdots,(n,k+1)$ 处的元素，即完成第$k+1$ 次消元.重复上述步骤，经$n-1$ 次选列主元消元后，增广矩阵$(A|b)$ 中的阵A 可化为一个上三角阵，解其对应的方程组即获得原方程组的解，此即为**选列主元消元法**.

例 3.2 用选列主元法求解例 2.1 中的线性方程组.

解 对方程组的增广矩阵进行选列主元消元得

$$(A|b)\Rightarrow \left[\begin{array}{ccc|c} -2.000 & 1.070 & 5.643 & 3.000 \\ -1.000 & 3.712 & 4.623 & 2.000 \\ 0.001 & 2.000 & 3.000 & 1.000 \end{array}\right] \Rightarrow$$

$$\left[\begin{array}{ccc|c} -2.000 & 1.070 & 5.643 & 3.000 \\ 0 & 3.176 & 1.801 & 0.5000 \\ 0 & 2.001 & 3.003 & 1.002 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|c} -2.000 & 1.070 & 5.643 & 3.000 \\ 0 & 3.176 & 1.801 & 0.5000 \\ 0 & 0 & 1.868 & 0.6870 \end{array}\right].$$

解上述最后一个矩阵对应的方程组得原方程组的解$X = [-0.4900,\ -0.05113,\ 0.3678]^T$, 其与精确解$X^* = [-0.4904,\ -0.05104,\ 0.3675]^T$ 比较，误差为$\|X - X^*\|_2 \approx 5.0804e-004$. ■

由此可见，选主元法有效地改进了Gauss 顺序消元法.

§3.2 Doolitle 分解法

当 Gauss 消元过程无零主元时，线性方程组(3.1) 的求解过程也可变换为求解二个系数阵分别为上三角阵和下三角阵的线性方程组，这将大大地节省 Gauss 消元过程所需时间和存储空间，其方法称为**Doolitle 分解法**、**三角分解法**或**LU 分解法**.

记$m_{ij}=a_{ij}^{(j)}/a_{jj}^{(j)}$，且引入Gauss变换

$$L_k=\begin{bmatrix}1&0&\cdots&\cdots&\cdots&\cdots&\cdots&0\\0&1&0&\cdots&\cdots&\cdots&\cdots&\vdots\\\vdots&0&\ddots&0&\cdots&\cdots&\cdots&\vdots\\0&\cdots&0&1&0&\cdots&\cdots&0\\0&\cdots&0&-m_{k+1,k}&1&0&\cdots&0\\\vdots&\cdots&0&\vdots&0&1&0&\vdots\\\vdots&\cdots&0&\vdots&0&0&\ddots&\vdots\\0&\cdots&0&-m_{n,k}&0&\cdots&0&1\end{bmatrix},$$

则 Gauss 消元过程：$(A^{(k)}|b^{(k)})\to(A^{(k+1)}|b^{(k+1)})$ 可等价地视为

$$L_kA^{(k)}=A^{(k+1)},\ \ L_kb^{(k)}=b^{(k+1)},\quad k=1,2,\cdots,n-1. \tag{3.7}$$

据此, 我们可导出Gauss顺序消元过程中产生的主元素 $a_{kk}^{(k)}\neq0$的判据.

定理 3.1 Gauss顺序消元过程中的主元素$a_{kk}^{(k)}$ $(k=1,2,\cdots,m)$均不为零的充要条件是原方程组(3.1)的系数阵A 的k阶顺序主子阵$\det(A_k)$ $(k=1,2,\cdots,m)$均非奇异.

证明 利用数学归纳法证明. 当$k=1$时, $a_{11}^{(1)}=\det(A_1)$, 此时结论成立. 设$1\leqslant k\leqslant m-1$时结论均成立. 以下仅需证: 当$A_1,A_2,\ldots,A_{m-1}$非奇异时, A_m非奇异当且仅当$a_{mm}^{(m)}\neq0$. 由归纳假设可知$a_{kk}^{(k)}\neq0$ $(k=1,2,\cdots,m-1)$. 因此, Gauss顺序消元过程至少可进行到$m-1$步, 且由式(3.7)有

$$L_{m-1}\cdots L_2L_1A=A^{(m)}=\begin{bmatrix}A_{11}^{(m)}&A_{12}^{(m)}\\0&A_{22}^{(m)}\end{bmatrix}, \tag{3.8}$$

其中$A_{11}^{(m)}$是以诸$a_{ii}^{(i)}$ $(i=1,2,\ldots,m-1)$为对角元的上三角阵. 若记L_i, $A^{(m)}$的m阶顺序主子阵分别为$L_{i,m}$及

$$\begin{bmatrix}A_{11}^{(m)}&\tilde{A}_{12}^{(m)}\\0&a_{mm}^{(m)}\end{bmatrix},$$

这里$\tilde{A}_{12}^{(m)}$为$A_{12}^{(m)}$的第一列元素, 则由式(3.8)有

$$L_{m-1,m}\cdots L_{2,m}L_{1,m}A_m=\begin{bmatrix}A_{11}^{(m)}&\tilde{A}_{12}^{(m)}\\0&a_{mm}^{(m)}\end{bmatrix}.$$

由此得

$$\det(A_m)=a_{mm}^{(m)}\det(A_{11}^{(m)}).$$

而$\det(A_{11}^{(m)})=\prod\limits_{i=1}^{m-1}a_{ii}^{(i)}\neq0$. 故$a_{mm}^{(m)}\neq0$当且仅当$\det(A_m)\neq0$. ■

进一步, 由式(3.7)递推得

$$A^{(n)}=L_{n-1}L_{n-2}\cdots L_2L_1A^{(1)},\quad b^{(n)}=L_{n-1}L_{n-2}\cdots L_2L_1b^{(1)}. \tag{3.9}$$

注意到$A^{(1)}=A$, $b^{(1)}=b$, 则由(3.9) 有

$$A=L_1^{-1}L_2^{-1}\cdots L_{n-2}^{-1}L_{n-1}^{-1}A^{(n)},\quad b=L_1^{-1}L_2^{-1}\cdots L_{n-2}^{-1}L_{n-1}^{-1}b^{(n)}. \tag{3.10}$$

由于诸L_k^{-1}为下三角矩阵，则其乘积$L \triangleq L_1^{-1}L_2^{-1}\cdots L_{n-2}^{-1}L_{n-1}^{-1}$ 仍是一个下三角矩阵. 若记

$$L=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & \cdots & 1 \end{bmatrix}, \quad U=A^{(n)}=\begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & u_{nn} \end{bmatrix},$$

则有

$$A=LU, \tag{3.11}$$

该分解称为矩阵A的**Doolitle 分解、三角分解**或LU **分解**. 形如L 的矩阵称为**单位下三角阵**，其转置阵称为**单位上三角阵**. 综合上述论据并应用定理 3.1，我们有如下结论.

定理 3.2 若阵$A\in R^{n\times n}$ 的顺序主子式$\det(A_i)\neq 0$ $(i=1,2,\cdots,n)$，则存在唯一的单位下三角阵L 及上三角阵U 使得(3.11) 成立.

由定理 3.2，方程组(3.1) 的求解过程可分解为下列子过程：

$$LY=b \;\Rightarrow\; Y=[y_1,\ y_2,\ \cdots,\ y_n]^T \;\Rightarrow\; UX=Y \;\Rightarrow\; X.$$

其具体步骤如下：

Step 1. 据(3.11) 计算$L=(l_{ij}), U=(u_{ij})$ 中的元素：

$$\begin{cases} u_{1i}=a_{1i}, \quad i=1,2,\cdots,n, \\ l_{i1}=a_{i1}/u_{11}, \quad i=2,3,\cdots,n, \\ u_{ri}=a_{ri}-\sum\limits_{k=1}^{r-1} l_{rk}u_{ki}, \quad i=r,r+1,\cdots,n; \quad r=2,3,\cdots,n, \\ l_{ir}=\big(a_{ir}-\sum\limits_{k=1}^{r-1} l_{ik}u_{kr}\big)/u_{rr}, \quad i=r,r+1,\cdots,n; \quad r=2,3,\cdots,n. \end{cases} \tag{3.12}$$

Step 3. 利用前推过程解方程组$LY=b$:

$$\begin{cases} y_1=b_1, \\ y_i=b_i-\sum\limits_{k=1}^{i-1} l_{ik}y_k, \quad i=2,3,\cdots,n. \end{cases} \tag{3.13}$$

Step 3. 利用回代过程解方程组$UX=Y$:

$$\begin{cases} x_n=y_n/u_{nn}, \\ x_i=\big(y_i-\sum\limits_{k=i+1}^{n} u_{ik}x_k\big)/u_{ii}, \quad i=n-1,n-2,\cdots,1. \end{cases} \tag{3.14}$$

该算法的 Matlab 程序如下:

算法 3.1 Doolitle 分解法

```
function x=lux(A,b)
 [n,n]=size(A); L=zeros(n); U=zeros(n);
 x=zeros(n,1); y=zeros(n,1);
for r=1:n
    for i=r:n
      U(r,i)=A(r,i)-sum(L(r,1:r-1).*U(1:r-1,i)');
      L(i,r)=(A(i,r)-sum(L(i,1:r-1).*U(1:r-1,r)'))/U(r,r);
    end
end;
 L, U
for i=1:n
    y(i)=b(i)-sum(L(i,1:i-1).*y(1:i-1)');
end
for j=n:-1:1
    x(j)=(y(j)-sum(U(j,j+1:n).*x(j+1:n)'))/U(j,j);
end
```

例 3.3 用Doolitle 分解法求解线性方程组

$$\begin{bmatrix} 1 & -2 & 3 & -1 \\ 4 & -1 & -2 & 2 \\ 3 & 2 & -1 & 1 \\ 2 & 5 & 2 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 8 \\ 10 \end{bmatrix}.$$

解 运行算法 3.1，可计算出系数阵A 的Doolitle 分解: $A = LU$, 其中

$$L = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 4.0000 & 1.0000 & 0 & 0 \\ 3.0000 & 1.1429 & 1.0000 & 0 \\ 2.0000 & 1.2857 & 2.3333 & 1.0000 \end{bmatrix}, \quad U = \begin{bmatrix} 1.0000 & -2.0000 & 3.0000 & -1.0000 \\ 0 & 7.0000 & -14.0000 & 6.0000 \\ 0 & 0 & 6.0000 & -2.8571 \\ 0 & 0 & 0 & -1.0476 \end{bmatrix},$$

且可得方程组的解$X = [x_1,\ x_2,\ x_3,\ x_4]^T = [1,\ 2,\ 3,\ 4]^T$. ■

§3.3 Cholesky 分解法

若线性方程组(3.1)的系数阵A 为对称正定阵，则其求解可采取一种特殊的方法: **Cholesky 分解法**，也称为**平方根法**. 该方法主要基于下述 Cholesky 分解定理.

定理 3.3 若矩阵A 为对称正定阵，则存在一个对角元均为正数的下三角阵$L \in R^{n\times n}$ 使得

$$A = LL^T, \tag{3.15}$$

其中L 称为A 的**Cholesky 因子**.

证明 由于矩阵A 为对称正定阵，则其所有主子阵均对称正定.由定理 3.2 存在唯一的单位下三角阵$\tilde{L}=(l_{ij})$ 及上三角阵$U=(u_{ij})$ 使得$A=\tilde{L}U$.记

$$D=\mathrm{diag}(u_{11},u_{22},\cdots,u_{nn}),\quad \tilde{U}=D^{-1}U,$$

则有

$$\tilde{U}^TD\tilde{L}^T=(\tilde{L}U)^T=A^T=A=\tilde{L}D\tilde{U},$$

即

$$\tilde{L}^T\tilde{U}^{-1}=D^{-1}(\tilde{U}^{-1})^T\tilde{L}D.$$

而上式左边是一个单位上三角阵，右边是一个单位下三角阵，故其二边均等于单位矩阵.因此$\tilde{U}=\tilde{L}^T$，从而$A=\tilde{L}D\tilde{L}^T$.又由于A 为正定阵，则D 的对角元均为正数.记

$$L=\tilde{L}\mathrm{diag}(\sqrt{u_{11}},\sqrt{u_{22}},\cdots,\sqrt{u_{nn}}),$$

即有$A=LL^T$，且L 的对角元为$l_{ii}=\sqrt{u_{ii}},\ i=1,2,\cdots,n.$ ■

通过比较方程(3.15) 二端的对应元素，可得

$$a_{ij}=\sum_{p=1}^{j}l_{ip}l_{jp},\quad 1\leqslant j\leqslant i\leqslant n. \tag{3.16}$$

由该式可按列依次计算$L=(l_{ij})$ 的元素，其计算步骤如下：

Step 1. 计算L 的第 1 列元素

$$l_{11}=\sqrt{a_{11}},\ \ l_{i1}=a_{i1}/l_{11},\quad i=2,3,\cdots,n;$$

Step 2. 若L 的前$k-1$ 列元素已计算，则进一步计算L 的第k 列元素

$$l_{kk}=\sqrt{a_{kk}-\sum_{p=1}^{k-1}l_{kp}^2},\ \ l_{ik}=\left(a_{ik}-\sum_{p=1}^{k-1}l_{ip}l_{kp}\right)/l_{kk},\quad i=k+1,k+2,\cdots,n.$$

在完成Cholesky分解后，我们可分别求解以下系数阵为下三角型和上三角型的方程组

$$LY=b,\quad L^TX=Y,$$

从而获得原方程组(3.1) 的解X.

我们注意到上述线性方程组的解法含有开方运算，其在计算过程中占用大量的运行时间.为避免开方运算，在下面我们介绍一个**改进的Cholesky 分解法**，或称为**改进的平方根法**. 由定理 3.3 的证明过程可知，对称正定阵A 也有如下分解：

$$A=LDL^T, \tag{3.17}$$

其中$L=(l_{ij})$为单位下三角阵,$D=\mathrm{diag}(d_1,d_2,\cdots,d_n)>0$. 由(3.17) 有

$$a_{ij}=\sum_{k=1}^{j-1}l_{ik}d_kl_{jk}+l_{ij}d_j,\quad 1\leqslant j\leqslant i\leqslant n. \tag{3.18}$$

由该式可得L, D 的如下计算步骤：

Step 1. 计算d_1, L 的第 1 列元素

$$d_1 = a_{11}, \quad l_{j1} = a_{j1}/d_1, \quad j = 2, 3, \cdots, n;$$

Step 2. 若D, L 的前$j-1$ 列元素已计算，则计算D, L 的第j 列元素

$$d_j = a_{jj} - \sum_{k=1}^{j-1} l_{jk} v_{jk}, \quad v_{jk} = l_{jk} d_k,$$

$$l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} v_{jk} \right) / d_j, \quad i = j+1, j+2, \cdots, n.$$

在完成形如(3.17)的分解后，我们可分别求解下列系数阵为下三角型和上三角型的方程组

$$LY = b, \quad DL^T X = Y.$$

从而获得原方程组(3.1) 的解X. 该算法程序如下：

算法 3.2 改进的Cholesky分解法

```
function x=cholesky(a,b)
 n=length(b); v=zeros(n); x=zeros(n,1); y=zeros(n,1);
 for j=1:n
      for i=1:j-1
        v(j,i)=a(j,i)*a(i,i);
      end
    a(j,j)=a(j,j)-a(j,1:j-1)*v(j,1:j-1)';
    a(j+1:n,j)=(a(j+1:n,j)-a(j+1:n,1:j-1)*v(j,1:j-1)')/a(j,j);
 end
    L=tril(a,-1)+eye(n), U=diag(diag(a))*L'
for i=1:n
  y(i)=b(i)-L(i,1:i-1)*y(1:i-1);
end
for j=n:-1:1
     x(j)=(y(j)-U(j,j+1:n)*x(j+1:n))/U(j,j);
end
```

在算法 3.2 中，为节省存储单元，我们将矩阵L 的主对角线以下的元素存储到了矩阵A 的相应位置，而将矩阵D 的对角元存储到了矩阵A 的对角元相应位置.

例 3.4 用改进的 Cholesky 分解法求解方程组

$$\begin{bmatrix} 4 & -2 & 4 & 2 \\ -2 & 10 & -2 & -7 \\ 4 & -2 & 8 & 4 \\ 2 & -7 & 4 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ 16 \\ 6 \end{bmatrix}.$$

解 记

$$a=\begin{bmatrix}4&-2&4&2\\-2&10&-2&-7\\4&-2&8&4\\2&-7&4&7\end{bmatrix},\quad b=\begin{bmatrix}8\\2\\16\\6\end{bmatrix}.$$

据算法3.2，运行Matlab命令Cholesky(a,b)得其方程组的解$X=[1,\ 2,\ 1,\ 2]^T$. ■

§3.4 追赶法

微分方程的有限元方法、有限差分法以及三次样条函数逼近法等常常产生如下形式的三对角稀疏型方程组

$$\begin{bmatrix}b_1&c_1&&&&&\\a_2&b_2&c_2&&&&\\&\ddots&\ddots&\ddots&&&\\&&a_i&b_i&c_i&&\\&&&\ddots&\ddots&\ddots&\\&&&&a_{n-1}&b_{n-1}&c_{n-1}\\&&&&&a_n&b_n\end{bmatrix}\begin{bmatrix}x_1\\x_2\\\vdots\\x_i\\\vdots\\x_{n-1}\\x_n\end{bmatrix}=\begin{bmatrix}d_1\\d_2\\\vdots\\d_i\\\vdots\\d_{n-1}\\d_n\end{bmatrix}.\tag{3.19}$$

例如：二阶边值问题(1.3)的差分格式(1.5) 可写成$(N-1)\times(N-1)$阶三对角型线性方程组:

$$\begin{bmatrix}-2+h^2q_1&1+\frac{h}{2}p_1&&&&\\1-\frac{h}{2}p_2&-2+h^2q_2&1+\frac{h}{2}p_2&&&\\\ddots&\ddots&\ddots&&&\\&\ddots&\ddots&\ddots&&\\&&1-\frac{h}{2}p_{N-2}&-2+h^2q_{N-2}&1+\frac{h}{2}p_{N-2}\\&&&1-\frac{h}{2}p_{N-1}&-2+h^2q_{N-1}\end{bmatrix}\begin{bmatrix}y_1\\y_2\\\vdots\\\vdots\\y_{N-2}\\y_{N-1}\end{bmatrix}$$

$$=\Big[h^2f_1-\Big(1-\frac{h}{2}p_1\Big)\alpha,\ h^2f_2,\ \cdots,\ h^2f_{N-2},\ h^2f_{N-1}-\Big(1+\frac{h}{2}p_{N-1}\Big)\beta\Big]^T.\tag{3.20}$$

方程组(3.19)的系数阵的非零元素集中分布在主对角线及其相邻两条次对角线上，依据该特点, 我们首先采用Gauss消元将其化为系数阵为单位上三角阵的方程组, 而后导出其求解公式, 并利用该求解公式编程计算求得其解. 为保证Gauss消元过程中的主元素不为零, 我们要求方程组(3.19)的系数阵的元素满足:

$$|b_1|>|c_1|>0;\ |b_i|\geqslant|a_i|+|c_i|,\ a_ic_i\neq0\ (i=2,3,\ldots,n-1);\ |b_n|\geqslant|a_n|>0.\tag{3.21}$$

方程组(3.19) 经$n-1$ 次消元后，可化为同解的上三角型方程组

$$\begin{bmatrix}1&\beta_1&&&\\&1&\beta_2&&\\&&\ddots&\ddots&\\&&&1&\beta_{n-1}\\&&&&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\\vdots\\x_{n-1}\\x_n\end{bmatrix}=\begin{bmatrix}\gamma_1\\\gamma_2\\\vdots\\\gamma_{n-1}\\\gamma_n\end{bmatrix},\tag{3.22}$$

其中

$$\begin{cases} \beta_1 = c_1/b_1, \quad \gamma_1 = d_1/b_1, \\ \beta_i = c_i/(b_i - \beta_{i-1}a_i), \quad i = 2, 3, \cdots, n-1, \\ \gamma_i = (d_i - \gamma_{i-1}a_i)/(b_i - \beta_{i-1}a_i), \quad i = 2, 3, \cdots, n. \end{cases} \tag{3.23}$$

利用回代过程即可求得方程组(3.19) 的解

$$x_n = \gamma_n, \quad x_i = \gamma_i - \beta_i x_{i+1}, \quad i = n-1, n-2, \cdots, 2. \tag{3.24}$$

该算法程序如下：

算法 3.3 追赶法

```
function x=chase(a,b,c,d)
  n=length(b); f(1)=c(1)/b(1); g(1)=d(1)/b(1);
 for i=2:n-1
  h(i)=b(i)-f(i-1)*a(i-1); f(i)=c(i)/h(i);
  g(i)=(d(i)-g(i-1)*a(i-1))/h(i);
 end
  g(n)=(d(n)-g(n-1)*a(n-1))/(b(n)-f(n-1)*a(n-1)); x(n)=g(n);
 for i=n-1:-1:1
  x(i)=g(i)-f(i)*x(i+1);
 end
```

上述前推过程称之为 **"追"**, 回代过程称之为 **"赶"**, 故称该方法为**追赶法**. 追赶法的优点在于可直接用公式 (3.23)-(3.24) 计算方程组(3.19)，其与 Gauss 消元法相比较, 不但节省了大量的存储单元，而且加快了计算速度.

例 3.5 用追赶法求解三对角型稀疏型方程组

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

解 记

$$a = [2,\ 2,\ 2,\ 2]^T, \quad b = [6,\ 6,\ 6,\ 6,\ 6]^T, \quad c = [3,\ 3,\ 3,\ 3]^T, \quad d = [1,\ 2,\ 3,\ 4,\ 5]^T$$

据算法 3.3，运行Matlab 命令chase(a, b, c, d) 得其方程组的解

$$X = [0.1194,\ 0.0944,\ 0.3981,\ 0.1407,\ 0.7864]^T. \quad \blacksquare$$

追赶法是一类求解带型稀疏方程组的方法. 在实际应用中还存在着各种其它类型的稀疏方程组, 其数值处理方法可参见专著[1, 12].

§3.5 扰动分析

线性方程组的各种直接方法理论上可计算出方程组的精确解，可是当针对某些方程组上机运行其算法时，其计算结果严重失真，甚至产生计算机溢出. 其主要原因是由于计算过程中的舍入误差使得输入的已知数据往往带有微小扰动，某些方程组对这种小扰动异常敏感，而导致其计算解产生大的偏差，我们称该类方程组为**病态方程组**. 下面，我们给出一个典型的病态方程组示例.

例 3.6 考虑4阶Hilbert型线性方程组

$$\begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix} X = \begin{bmatrix} 25/12 \\ 77/60 \\ 19/20 \\ 319/420 \end{bmatrix},$$

其精确解为$X = [1,\ 1,\ 1,\ 1]^T$. 当将该方程组的已知数据仅保留5位有效数字时，其方程组成为

$$\begin{bmatrix} 1.0000 & 0.5000 & 0.3333 & 0.2500 \\ 0.5000 & 0.3333 & 0.2500 & 0.2000 \\ 0.3333 & 0.2500 & 0.2000 & 0.1667 \\ 0.2500 & 0.2000 & 0.1667 & 0.1429 \end{bmatrix} \hat{X} = \begin{bmatrix} 2.0833 \\ 1.2833 \\ 0.9500 \\ 0.7595 \end{bmatrix}.$$

经计算，该方程组保留5位有效数字的解为$\hat{X} = [1.0185,\ 0.7832,\ 1.5355,\ 0.6457]^T$, 其相对误差

$$\frac{\|X - \hat{X}\|_\infty}{\|X\|_\infty} \approx 0.5355. \quad \blacksquare$$

在下面，为研究线性方程组的病态现象，我们通过分析其误差导出衡量线性方程组是否病态的一个评价指标.

定理3.4 设线性方程组(3.1) 的系数阵A 及右端向量b 分别带有微小扰动δ_A 和δ_b，其导出的扰动解$X + \delta_X$ 满足

$$(A + \delta_A)(X + \delta_X) = b + \delta_b. \tag{3.25}$$

则其解具有如下局部相对误差估计：

$$\frac{\|\delta_X\|}{\|X\|} \leqslant \frac{\|A^{-1}\|\|A\|}{1 - \|A^{-1}\|\|\delta_A\|} \left(\frac{\|\delta_b\|}{\|b\|} + \frac{\|\delta_A\|}{\|A\|} \right). \tag{3.26}$$

证明 将(3.1) 代入(3.25) 得

$$\delta_X = (A + \delta_A)^{-1}(\delta_b - \delta_A X) = (I + A^{-1}\delta_A)^{-1} A^{-1}(\delta_b - \delta_A X). \tag{3.27}$$

不妨设$\|A^{-1}\delta_A\| < 1$，则由上式及定理 1.10 得

$$\|\delta_X\| \leqslant \frac{\|A^{-1}\|}{1 - \|A^{-1}\delta_A\|}(\|\delta_b\| + \|\delta_A\|\|X\|) \leqslant \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\delta_A\|}(\|\delta_b\| + \|\delta_A\|\|X\|).$$

从而

$$\frac{\|\delta_X\|}{\|X\|} \leqslant \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\delta_A\|} \left(\frac{\|\delta_b\|}{\|X\|} + \|\delta_A\| \right). \tag{3.28}$$

又由$\|b\| = \|AX\| \leqslant \|A\|\|X\|$ 有

$$\frac{1}{\|X\|} \leqslant \frac{\|A\|}{\|b\|}. \tag{3.29}$$

代入(3.29)到(3.28)即得误差估计(3.26). ■

当扰动$\|\delta_A\|$充分小时，我们有如下近似关系：

$$\frac{\|A^{-1}\|\|A\|}{1 - \|A^{-1}\|\|\delta_A\|} \approx \|A^{-1}\|\|A\|. \tag{3.30}$$

定理3.4及上述近似关系表明：$\|A^{-1}\|\|A\|$ 表征着方程组(3.1)的解X 对其系数阵A 及右端向量b变化的敏度，故称之为方程组(3.1) 的**条件数**，并记为$\mathrm{Cond}(A)$. 当$\mathrm{Cond}(A) >> 1$ 时，方程组(3.1)视为是**病态的**. 常用条件数有

$$\mathrm{Cond}_1(A) \triangleq \|A^{-1}\|_1\|A\|_1, \quad \mathrm{Cond}_\infty(A) \triangleq \|A^{-1}\|_\infty\|A\|_\infty,$$

$$\mathrm{Cond}_2(A) \triangleq \|A^{-1}\|_2\|A\|_2 = \sqrt{\frac{\max_{1\leqslant i\leqslant n}\{\lambda_i^{A^TA}\}}{\min_{1\leqslant i\leqslant n}\{\lambda_i^{A^TA}\}}}.$$

这里$\lambda_i^{A^TA}$表示矩阵A^TA的第i个特征值. 据此，我们可计算出例3.6的三种条件数为：

$$\mathrm{Cond}_1(A) \approx 36250, \quad \mathrm{Cond}_2(A) \approx 19808, \quad \mathrm{Cond}_\infty(A) \approx 36250.$$

由此可见, 例3.6中的Hilbert型线性方程组是病态方程组. 因此, 其扰动解$\hat{X}$存在着较大的偏差. 此外，当矩阵A为对称阵时，其基于l_2范数的条件数可写成：

$$\mathrm{Cond}_2(A) = \frac{\max_{1\leqslant i\leqslant n}\{|\lambda_i^A|\}}{\min_{1\leqslant i\leqslant n}\{|\lambda_i^A|\}}.$$

§3.6 一般单步迭代法

一般而言，迭代法比直接法更适合于现代大规模科学工程计算，其具有所需存储单元少、程序简单、计算速度快等优点. 若迭代法每步计算仅需其前面一步的迭代值, 则称该迭代法为**单步迭代法**, 否则称之为**多步迭代法**. 以下, 我们将主要介绍单步迭代法. 设线性方程组(3.1)有如下迭代格式

$$X^{(k+1)} = BX^{(k)} + F, \quad k = 0, 1, 2, \cdots \tag{3.31}$$

其中$X^{(0)}$为给定的初始向量，$X^{(k)} = [x_1^{(k)},\ x_2^{(k)},\ \cdots,\ x_n^{(k)}]^T \in R^n$，$B = (b_{ij}) \in R^{n\times n}$，$F = [f_1,\ f_2,\ \cdots,\ f_n]^T \in R^n$，矩阵$B$ 称为其迭代格式的**迭代矩阵**. 若有

$$\lim_{k\to+\infty} X^{(k)} = X^*,$$

则称其迭代格式是**收敛的**. 在迭代格式收敛的前提下, 由(3.31) 二端取极限得

$$X^* = BX^* + F. \tag{3.32}$$

等式(3.32)中的X^*未必就是方程组(3.1)的精确解. 为此, 我们需要引入相容性概念. 称方程组(3.32) 与(3.1)是**相容的**，若存在可逆阵$G \in R^{n\times n}$ 使得

$$G(I-B)=A, \quad GF=b. \tag{3.33}$$

当方程组(3.32)与(3.1)相容时，其二者是等价的，即X^* 必为线性方程组(3.1) 的精确解. 为简单见, 以下我们恒设方程组(3.32) 与(3.1)是相容的. 对于一个迭代算法，只有当其迭代序列收敛于方程组的精确解时，其才是一个有效算法. 此时, 我们可将迭代收敛序列$\{X^{(k)}\}$ 的某项$X^{(k)}$ 作为精确解X^* 的近似值. 鉴此，本节将对线性方程组的一般单步迭代法的收敛性进行分析，并导出其收敛性判据.

定理 3.5 当给定初始向量$X^{(0)}$ 时，迭代格式(3.31) 收敛的充要条件是其迭代矩阵B 的谱半径$\rho(B)<1$.

证明 由(3.31)-(3.32)有

$$X^{(k)}-X^*=B(X^{(k-1)}-X^*)=\cdots=B^k(X^{(0)}-X^*), \tag{3.34}$$

则$\lim\limits_{k\to+\infty}X^{(k)}=X^*$当且仅当$\lim\limits_{k\to+\infty}B^k=0$，从而应用定理 1.9即得欲证结论. ■

事实上，当迭代法应用于高阶线性方程组时，利用其迭代阵的谱半径来判别收敛性是十分困难的，因此有必要给出方便易行的收敛性判别准则. 下面我们给出一个利用矩阵范数来判别收敛性的准则，其不但给出了收敛性判据，而且给出了迭代解的先验误差估计与后验误差估计.

定理 3.6 若存在从属于某向量范数$\|\cdot\|$ 的矩阵范数$\|\cdot\|$ 使得迭代法(3.31) 的迭代阵B 满足$\|B\|<1$, 则该迭代法是收敛的，且满足先验误差估计

$$\|X^{(k)}-X^*\|\leqslant\frac{\|B\|^k}{1-\|B\|}\|X^{(1)}-X^{(0)}\|, \quad k=1,2,\cdots \tag{3.35}$$

及后验误差估计

$$\|X^{(k)}-X^*\|\leqslant\frac{\|B\|}{1-\|B\|}\|X^{(k)}-X^{(k-1)}\|, \quad k=1,2,\cdots \tag{3.36}$$

其中X^* 为方程组(3.1) 的精确解.

证明 据(3.34) 有

$$\|X^{(k)}-X^*\|\leqslant\|B\|^k\|X^{(0)}-X^*\|. \tag{3.37}$$

由$\|B\|<1$ 可知$I-B$ 可逆，则方程(3.32) 等价于$X^*=(I-B)^{-1}F$, 且由此有

$$X^{(0)}-X^*=(I-B)^{-1}[(I-B)X^{(0)}-F]=(I-B)^{-1}(X^{(0)}-X^{(1)}).$$

因此

$$\|X^{(0)}-X^*\|\leqslant\|(I-B)^{-1}\|\|X^{(0)}-X^{(1)}\|\leqslant\frac{1}{1-\|B\|}\|X^{(0)}-X^{(1)}\|.$$

将该式代入(3.37) 即得先验误差估计(3.35). 由该误差估计及$\|B\|<1$可知迭代法(3.31)是收敛的. 又据(3.31) 和(3.32) 有

$$\begin{aligned}X^{(k)}-X^*&=B(X^{(k-1)}-X^*)=B[X^{(k-1)}-(I-B)^{-1}F]\\ &=B(I-B)^{-1}[(I-B)X^{(k-1)}-F]=B(I-B)^{-1}(X^{(k-1)}-X^{(k)}).\end{aligned}$$

由此得

$$\|X^{(k)}-X^*\|\leqslant\|B\|\|(I-B)^{-1}\|\|X^{(k-1)}-X^{(k)}\|\leqslant\frac{\|B\|}{1-\|B\|}\|X^{(k)}-X^{(k-1)}\|.\quad\blacksquare$$

由后验误差估计(3.36)可得迭代法的计算终止准则，即当用户要求数值解的精度满足：$\|X^{(k)}-X^*\|<\varepsilon$时，我们仅需在计算程序中设置终止准则：$\|X^{(k)}-X^{(k-1)}\|<\varepsilon$.

求解线性方程组有着多种收敛的迭代算法，但其计算效率有时存在着较大差异. 评价一个算法计算效率的重要标志是其收敛速度. 对于单步迭代法(3.31)，由递推关系(3.34)有

$$\|X^{(k)}-X^*\|\leqslant\|B^k\|\|X^{(0)}-X^*\|.$$

因此，人们常常用

$$R_k(B)\triangleq-\ln\sqrt[k]{\|B^k\|}$$

来刻画迭代法的**收敛速度**. 特别，当B为对称阵、Hermite阵或正规阵时，则有$\|B^k\|_2=[\rho(B)]^k$，且因此有$R_k(B)=-\ln[\rho(B)]$. 然而, 对于分别具迭代阵G, H的二个迭代法而言，在迭代过程中$R_k(G),R_k(H)$常常无固定大小之分，为此人们以

$$R_\infty(B)\triangleq\lim_{k\to\infty}R_k(B)$$

来比较二者的收敛速度，并称之为**渐近收敛速度**.

定理 3.7 设单步迭代法(3.31)的迭代阵为B，则其渐近收敛速度$R_\infty(B)=-\ln[\rho(B)]$.

证明 本定理仅需证$\lim\limits_{k\to\infty}\sqrt[k]{\|B^k\|}=\rho(B)$. 事实上，由$[\rho(B)]^k=\rho(B^k)\leqslant\|B^k\|$得$\rho(B)\leqslant\sqrt[k]{\|B^k\|}$. 另一方面，$\forall\varepsilon>0$，作矩阵$B_\varepsilon=\frac{1}{\rho(B)+\varepsilon}B$，其满足

$$\rho(B_\varepsilon)=\frac{\rho(B)}{\rho(B)+\varepsilon}<1.$$

因此，由定理1.9有$\lim\limits_{k\to\infty}B_\varepsilon^k=0$. 从而，存在自然数$N$使当$k\geqslant N$时，$\|B_\varepsilon^k\|<1$, 即$\|B^k\|<[\rho(B)+\varepsilon]^k$. 至此，我们已证：$\forall\varepsilon>0$，存在自然数$N$使当$k\geqslant N$时，

$$\rho(B)\leqslant\sqrt[k]{\|B^k\|}\leqslant\rho(B)+\varepsilon.$$

故$\lim\limits_{k\to\infty}\sqrt[k]{\|B^k\|}=\rho(B)$. $\blacksquare$

§3.7 Jacobi迭代法

将线性方程组(3.1)的系数阵A 分解为

$$A=L+D+U,$$

其中$D=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$,

$$L=\begin{bmatrix}0&0&\cdots&0&0\\a_{21}&0&\cdots&0&0\\a_{31}&a_{32}&\cdots&0&0\\\vdots&\vdots&\ddots&\ddots&\vdots\\a_{n1}&a_{n2}&\cdots&a_{n,n-1}&0\end{bmatrix},\quad U=\begin{bmatrix}0&a_{12}&a_{13}&\cdots&a_{1n}\\0&0&a_{23}&\cdots&a_{2n}\\\vdots&\vdots&\ddots&\ddots&\vdots\\0&0&\cdots&0&a_{n-1,n}\\0&0&\cdots&0&0\end{bmatrix}.$$

当对角阵D 可逆时，方程组(3.1) 可等价地写为

$$X = -D^{-1}(L+U)X + D^{-1}b. \tag{3.38}$$

据此得**Jacobi 迭代公式**

$$X^{(k+1)} = -D^{-1}(L+U)X^{(k)} + D^{-1}b, \quad k = 0, 1, \cdots, \tag{3.39}$$

其中$X^{(k)} = [x_1^{(k)},\ x_2^{(k)},\ \cdots,\ x_n^{(k)}]^T \in R^n$. 该迭代公式也可写成如下分量形式：

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, 2, \cdots, n. \tag{3.40}$$

为导出Jacobi迭代法更为简单的收敛性判据, 我们需要引入下列概念及相关预备性结论.

定义 3.1 设矩阵$A = (a_{ij}) \in R^{n\times n}$, 若其满足

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

则称矩阵A 是对角占优的；若上述不等式对所有i 有严格不等号成立，则称矩阵A 是严格对角占优的；若上述不等式至少对某个i 有严格不等号成立，则称矩阵A 是弱严格对角占优的.

定义 3.2 若存在n阶排列阵P，使得

$$PAP^T = \begin{bmatrix} A_{11} & 0 \\ A_{12} & A_{22} \end{bmatrix},$$

其中A_{11}是r $(0 < r < n)$阶方阵，A_{22}是$n-r$阶方阵，则称矩阵A是可约的；若不存在这样的排列阵P，则称A是不可约的；若矩阵A不可约且弱严格对角占优，则称A是不可约对角占优的.

引理 3.1 (参见[4, 5, 7]) 若矩阵A 严格对角占优或不可约对角占优，则A 非奇异. 进一步, 若矩阵A是具正对角元的对称阵，则A 为正定阵.

基于上述引理, 我们有下列收敛性判据.

定理 3.8 若线性方程组(3.1) 的系数矩阵A 严格对角占优或不可约对角占优，则 Jacobi 迭代法是收敛的.

证明 由A 严格对角占优或不可约对角占优有$|a_{ii}| > 0$ $(1 \leqslant i \leqslant n)$，因此对角阵$D = \mathrm{diag}(a_{11}, a_{22}, \cdots, a_{nn})$ 可逆.若 Jacobi 迭代阵$B_J = -D^{-1}(L+U)$ 有特征值$\tilde{\lambda}$ 满足$|\tilde{\lambda}| \geqslant 1$, 则由$A$ 严格对角占优或不可约对角占优可推得$\tilde{\lambda}D + L + U$ 也严格对角占优或不可约对角占优，从而据引理 3.1有

$$\det(\tilde{\lambda}I - B_J) = \det[\tilde{\lambda}I + D^{-1}(L+U)] = \det(D^{-1})\det(\tilde{\lambda}D + L + U) \neq 0.$$

因此 Jacobi 迭代阵的谱半径$\rho(B_J) < 1$. 故由定理3.5推得Jacobi 迭代法是收敛的. ■

例 3.7 试利用 Jacobi 迭代公式求解方程组

$$\begin{bmatrix} 5 & -1 & -1 & -1 \\ -1 & 10 & -1 & -1 \\ -1 & -1 & 5 & -1 \\ -1 & -1 & -1 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -4 \\ 12 \\ 8 \\ 34 \end{bmatrix}, \tag{3.41}$$

要求数值解$X^{(k)}$ 满足：$\|X - X^{(k)}\|_2 \leqslant 10^{-4}$，其中$X = [1,\ 2,\ 3,\ 4]^T$ 为方程组的精确解.

解 易验证, 方程组(3.41)的系数阵是严格对角占优的. 因此, 据定理3.8, Jacobi 迭代公式关于方程组(3.41)是收敛的. 自$X^{(0)} = 0$ 开始，应用 Jacobi 迭代公式(3.39) 计算其方程组，经 14 次迭代后可得数值解

$$X^{(14)} \approx \begin{bmatrix} 0.99996955406204 \\ 1.999981943318820 \\ 2.999969553734361 \\ 3.999981943318800 \end{bmatrix},$$

其计算精度$\|X - X^{(14)}\|_2 \approx 5.006014083879933e - 005 < 10^{-4}$. ■

§3.8 Gauss-Seidel 迭代法

方程组(3.1) 也可等价地写为

$$DX = -LX - UX + b. \tag{3.42}$$

若记$X^{(k)} = [x_1^{(k)},\ x_2^{(k)},\ \cdots,\ x_n^{(k)}]^T \in R^n$，则由上式可构造迭代格式

$$DX^{(k+1)} = -LX^{(k+1)} - UX^{(k)} + b, \quad k = 0, 1, 2, \cdots.$$

当对角阵D 可逆时，上式等价于

$$X^{(k+1)} = -(D + L)^{-1}UX^{(k)} + (D + L)^{-1}b. \tag{3.43}$$

此即为**Gauss-Seidel 迭代公式**, 其分量形式为

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, 2, \cdots, n. \tag{3.44}$$

由§3.10节中的定理3.12可直接获得Gauss-Seidel迭代法的如下收敛性判定准则.

定理 3.9 若线性方程组(3.1) 的系数矩阵A 严格对角占优或不可约对角占优，则Gauss-Seidel 迭代法是收敛的.

例 3.8 试利用 Gauss-Seidel 迭代公式求解方程组(3.41)，并使数值解$X^{(k)}$ 满足精度要求：$\|X - X^{(k)}\|_2 \leqslant 10^{-4}$，其中$X = [1,\ 2,\ 3,\ 4]^T$ 为方程组的精确解.

解 易验证, 方程组(3.41)的系数阵是严格对角占优的. 因此, 据定理3.9, Gauss-Seidel 迭代公式关于方程组(3.41)是收敛的. 自$X^{(0)} = 0$ 开始，应用 Gauss-Seidel 迭代公式(3.43) 计算其方程组，经 8 次迭代后可得数值解

$$X^{(8)} \approx \begin{bmatrix} 0.999954785345266 \\ 1.999981611948102 \\ 2.999979261443341 \\ 3.999991565873671 \end{bmatrix},$$

且其满足计算精度$\|X - X^{(8)}\|_2 \approx 5.370016456711980e - 005 < 10^{-4}$. ■

§3.9 JOR迭代法

逐次超松弛迭代是加速经典迭代格式收敛速度的一种技术，其通过引入适当的松弛因子，使得改进的迭代格式具有较快的收敛速度. 本节以改进的Jacobi迭代公式为例来阐明这种加速收敛技术.

在矩阵D可逆条件下, Jacobi迭代公式有下列等价形式

$$X^{(k+1)} = X^{(k)} - D^{-1}(AX^{(k)} - b).$$

今在上式右端引入松弛因子$\omega > 0$，则得

$$X^{(k+1)} = X^{(k)} - \omega D^{-1}(AX^{(k)} - b),$$

即

$$X^{(k+1)} = (I - \omega D^{-1}A)X^{(k)} + \omega D^{-1}b. \tag{3.45}$$

方法(3.45)被称为**JOR迭代方法**, 其有分量形式

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, 2, \cdots, n. \tag{3.46}$$

当$\omega = 1$时，JOR迭代方法即为 Jacobi迭代方法.

理论分析表明(参见文献[5])：当Jacobi迭代方法的迭代阵B_J的特征值为实数, 且B_J的谱半径$\rho(B_J) < 1$时, JOR迭代方法有最佳松弛因子

$$\omega_{opt} = \frac{2}{2 - \lambda_{\max}^{B_J} - \lambda_{\min}^{B_J}},$$

其中$\lambda_{\max}^{B_J}$, $\lambda_{\min}^{B_J}$ 分别表示Jacobi迭代阵$B_J \triangleq -D^{-1}(L+U)$的最大和最小特征值. 此外，当$\lambda_{\min}^{B_J} \neq -\lambda_{\max}^{B_J}$时，JOR迭代方法的收敛速度比相应Jacobi迭代法的收敛速度快. 此外, JOR迭代方法具有如下收敛性准则.

定理 3.10 若Jacobi迭代法关于线性方程组(3.1)收敛，则松弛因子$\omega \in (0, 1]$ 的JOR迭代方法是收敛的.

证明 设λ是JOR迭代阵B_{JOR}的任一特征值, E为相应特征向量, 则有

$$\lambda E = B_{JOR}E = [I - \omega(I - B_J)]E.$$

由此得$B_J E = \frac{1}{\omega}(\lambda - 1 + \omega)E$, 此表明$\frac{1}{\omega}(\lambda - 1 + \omega)$为$B_J$的特征值. 由Jacobi迭代法收敛性有

$$\left|\frac{1}{\omega}(\lambda - 1 + \omega)\right| < 1.$$

据此

$$|\lambda| - |1 - \omega| \leqslant |\lambda - 1 + \omega| < |\omega|,$$

且由此及$0 < \omega \leqslant 1$得

$$|\lambda| < |1 - \omega| + |\omega| = 1.$$

故JOR迭代方法收敛. ■

综合定理3.8 和定理3.10, 我们可进一步获得以下收敛性准则.

定理 3.11 若线性方程组(3.1) 的系数矩阵A 严格对角占优或不可约对角占优，则松弛因子$\omega \in (0,1]$ 的 JOR迭代方法是收敛的.

例 3.9 试采用最佳松弛因子ω_{opt} 构造一类JOR迭代方法求解方程组(3.41)，并使数值解$X^{(k)}$满足精度要求：$\|X - X^{(k)}\|_2 \leqslant 10^{-4}$，其中$X = [1,\ 2,\ 3,\ 4]^T$ 为方程组的精确解.

解 由于Jacobi迭代公式关于方程组(3.41)的迭代阵有实特征值, 且其谱半径$\rho(B_J) \approx 0.4372 < 1$, 因此JOR迭代方法关于方程组(3.41) 具最佳松弛因子：

$$\omega_{opt} = \frac{2}{2 - \lambda_{\min}^{B_J} - \lambda_{\max}^{B_J}} \approx 1.1346,$$

且相应JOR迭代阵

$$B_{JOR} \triangleq \ I - \omega D^{-1}A \approx \begin{bmatrix} -0.1346 & 0.2269 & 0.2269 & 0.2269 \\ 0.1135 & -0.1346 & 0.1135 & 0.1135 \\ 0.2269 & 0.2269 & -0.1346 & 0.2269 \\ 0.1135 & 0.1135 & 0.1135 & -0.1346 \end{bmatrix}.$$

经计算, 迭代阵的谱半径$\rho(B_{JOR}) \approx 0.3615 < 1$. 由此可知, JOR迭代方法关于方程组(3.41)是收敛的. 自$X^{(0)} = 0$ 开始，应用JOR迭代方法(3.45) 计算其方程组，经11 次迭代后可得数值解

$$X^{(11)} \approx \begin{bmatrix} 0.999939720704236 \\ 1.999974436999159 \\ 2.999967270420092 \\ 3.999974874180734 \end{bmatrix},$$

且其满足计算精度$\|X - X^{(11)}\|_2 \approx 7.739245896476788e-005 < 10^{-4}$. ■

由上例可见, JOR迭代方法加快了 Jacobi迭代法的收敛速度.

§3.10 SOR迭代法

以下，我们将Gauss-Seidel迭代法改造为一类逐次超松弛迭代格式. Gauss-Seidel迭代公式有下列等价形式

$$DX^{(k+1)} = DX^{(k)} + [b - LX^{(k+1)} - (D+U)X^{(k)}].$$

今在上式右端引入松弛因子$\omega > 0$，则得

$$DX^{(k+1)} = DX^{(k)} + \omega[b - LX^{(k+1)} - (D+U)X^{(k)}],$$

即

$$(D + \omega L)X^{(k+1)} = [(1-\omega)D - \omega U]X^{(k)} + \omega b.$$

若阵D 可逆，则有

$$X^{(k+1)} = (D + \omega L)^{-1}\{[(1-\omega)D - \omega U]X^{(k)} + \omega b\}. \tag{3.47}$$

该方法被称为**SOR迭代方法**, 其分量形式为

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, 2, \cdots, n. \tag{3.48}$$

当$\omega = 1$ 时，SOR迭代方法方法即为 Gauss-Seidel 迭代方法.

理论分析表明(参见文献[5, 15])： SOR迭代方法收敛的必要条件是$0 < \omega < 2$. 特别，当Jacobi迭代方法的迭代阵B_J的特征值为实数且其谱半径$\rho(B_J) < 1$时, SOR迭代方法有最佳松弛因子

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2[-D^{-1}(L+U)]}}.$$

此外, 对于系数阵为严格对角占优阵或不可约对角占优阵的线性方程组，其SOR 迭代方法有下列更细致的收敛性定理.

定理 3.12 若线性方程组(3.1) 的系数矩阵A 严格对角占优或不可约对角占优，则松弛因子$\omega \in (0, 1]$ 的SOR 迭代方法是收敛的.

证明 若SOR 方法的迭代阵

$$B_{SOR} = (D + \omega L)^{-1}[(1-\omega)D - \omega U], \quad \omega \in (0, 1]$$

有特征值$\tilde{\lambda}$ 满足$|\tilde{\lambda}| \geqslant 1$, 则由$A$严格对角占优或不可约对角占优可推得$(\tilde{\lambda} - 1 + \omega)D + \omega(\tilde{\lambda}L + U)$ 也严格对角占优或不可约对角占优，从而

$$\begin{aligned} & \det[\tilde{\lambda}I - B_{SOR}] = \det[\tilde{\lambda}I - (D + \omega L)^{-1}((1-\omega)D - \omega U)] \\ = \quad & \det[(D + \omega L)^{-1}] \det[(\tilde{\lambda} - 1 + \omega)D + \omega(\tilde{\lambda}L + U)] \neq 0. \end{aligned}$$

因此SOR迭代阵的谱半径$\rho(B_{SOR}) < 1$. 故由定理 3.5，SOR 迭代法是收敛的. ■

例 3.10 试采用最佳松弛因子ω_{opt} 构造一类SOR迭代方法求解方程组(3.41)，并使数值解$X^{(k)}$ 满足精度要求：$\|X - X^{(k)}\|_2 \leqslant 10^{-4}$，其中$X = [1,\ 2,\ 3,\ 4]^T$ 为方程组的精确解.

解 由前已知: Jacobi迭代公式关于方程组(3.41)的迭代阵有实特征值，且其谱半径$\rho(B_J) \approx 0.4372 < 1$. 因此, SOR迭代方法关于方程组(3.47) 有最佳松弛因子：

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2[-D^{-1}(L+U)]}} \approx 1.0530.$$

经计算，具上述最佳松弛因子的SOR迭代阵

$$\begin{aligned} B_{SOR} \triangleq \quad & (D + \omega L)^{-1}[(1-\omega)D - \omega U] \\ \approx \quad & \begin{bmatrix} -0.0530 & 0.2106 & 0.2106 & 0.2106 \\ -0.0056 & -0.0308 & 0.1275 & 0.1275 \\ -0.0123 & 0.0379 & 0.0182 & 0.2818 \\ -0.0075 & 0.0229 & 0.0375 & 0.0123 \end{bmatrix}, \end{aligned}$$

其谱半径$\rho(B_{SOR}) \approx 0.1229 < 1$. 因此，此时SOR迭代方法关于方程组(3.41) 是收敛的. 自$X^{(0)} = 0$ 开始，应用该SOR迭代方法于方程组，经6次迭代后可得数值解

$$X^{(6)} \approx \begin{bmatrix} 0.999929047746394 \\ 1.999968030776484 \\ 2.999974285302719 \\ 3.999990727549335 \end{bmatrix},$$

且其满足计算精度$\|X - X^{(6)}\|_2 \approx 8.248319551014242e-005 < 10^{-4}$. ■

由上例可见, SOR迭代方法加快了Gauss-Seidel 迭代法的收敛速度.

习 题 三

3.1 设矩阵$A = (a_{ij}) \in R^{n\times n}$满足$a_{11} \neq 0$，且其经一步Gauss消元后成为

$$\begin{bmatrix} a_{11} & a_1^T \\ 0 & A_2 \end{bmatrix}.$$

证明:（1）若A为对称阵，则A_2也是对称阵;（2）若A为正定阵，则A_2也是正定阵;（3）若A为严格对角占优阵，则A_2也是严格对角占优阵.

3.2 用Doolitle 分解法求解方程组

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

3.3 用改进的 Cholesky 分解法求解方程组

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

3.4 用追赶法求解三对角型方程组

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

3.5 设有线性方程组$AX = b$, 其中

$$A = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix}, \quad b = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}.$$

试解答下列问题：

(1) 给出该方程组的精确解X;

(2) 计算条件数$\mathrm{Cond}_\infty(A)$, 并由此判定方程组是否病态;

(3) 若给系数阵A以扰动: $\delta A = 0.01A$, 给向量b以扰动: $\delta b = [0.01,\ -0.01,\ 0.01,\ -0.01]^T$, 试求解扰动方程

$$(A + \delta A)\hat{X} = b + \delta b,$$

并计算其相对误差$\hat{e} \triangleq \|X - \hat{X}\|_\infty / \|X\|_\infty$.

3.6 设$B \in R^{n\times n}$的谱半径$\rho(B) = 0$. 证明：对任意给定的X_0, $G \in R^n$, 迭代格式

$$X^{k+1} = BX^k + G, \quad k = 1, 2, \ldots$$

最多迭代n次即可获得方程组$(I - B)X = G$的精确解.

3.7 设线性方程组(3.1)的系数矩阵A是具正对角元的对称阵. 证明：JOR迭代法收敛的充要条件是A和$2\omega^{-1}D - A$均正定.

3.8 设线性方程组(3.1) 的系数矩阵A为实对称正定阵. 证明：SOR 迭代方法收敛的充要条件是其松弛因子$\omega \in (0, 2)$.

3.9 设线性方程组$AX = b$的实系数矩阵为

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

证明： 当$|a| < \frac{1}{2}$时, 求解该方程组的Jacobi迭代法收敛; 当$\frac{1}{2} < a < 1$时, 求解该方程组的Gauss-Seidel迭代法收敛.

3.10 (实验题) 给定线性方程组

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ -2 & 3 & 4 & 5 & 6 \\ -3 & -4 & 5 & 6 & 7 \\ -4 & -5 & -6 & 7 & 8 \\ -5 & -6 & -7 & -8 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 55 \\ 66 \\ 63 \\ 36 \\ -25 \end{bmatrix}.$$

试以$\|X^{(k+1)} - X^{(k)}\|_\infty < 10^{-5}$作为终止准则，分别利用Jacobi迭代法、Gauss-Seidel迭代法及具最佳松弛因子的JOR迭代方法和SOR迭代法求解该方程组, 并比较这些方法的计算时间.

第四章　插值与曲线拟合方法

实际问题中遇到的函数$f(x)$是多种多样的，有的表达式很复杂，有的甚至没有给出表达式，只提供了一些离散点上的函数值或导数值，为洞察实际问题中复杂函数的性质和变化规律，人门自然希望能找到一个充分逼近$f(x)$的简单函数$P(x)$. 当然, 逼近函数$P(x)$的选择不同, 其逼近$f(x)$的效果也不同. 通常, $P(x)$ 可以是一个代数多项式、三角多项式、有理分式或某些分段光滑函数. 为寻求这些逼近函数，本章将主要介绍插值与曲线拟合方法.

§4.1　Lagrange 插值

代数多项式函数类是最简单的函数类，为此本节首先考虑在该函数类中的插值问题. 设函数$f(x)$ 在区间$[a,b]$ 上有定义，其在$n+1$ 个互异点$x_i \in [a,b]$ 处的函数值$y_i = f(x_i)(i = 0, 1, \cdots, n)$ 为已知，今求一个次数不超过n 的代数多项式

$$P_n(x) = a_0 + a_1 x + \cdots + a_n x^n, \tag{4.1}$$

使其满足

$$P_n(x_i) = y_i, \quad i = 0, 1, 2, \cdots, n. \tag{4.2}$$

该问题称为**代数插值问题**, 诸x_i 称为**插值节点**，所求得的多项式$P_n(x)$ 称为**Lagrange 插值多项式**. 下述定理表明该代数插值问题的解$P_n(x)$ 存在且唯一.

定理 4.1　代数插值问题(4.1)-(4.2) 的解$P_n(x)$ 存在且唯一.

证明　由条件(4.2) 有

$$\begin{cases} a_0 + a_1 x_0 + \cdots + a_n x_0^n &= y_0 \\ a_0 + a_1 x_1 + \cdots + a_n x_1^n &= y_1 \\ \cdots \quad \cdots & \\ a_0 + a_1 x_n + \cdots + a_n x_n^n &= y_n. \end{cases} \tag{4.3}$$

注意到方程组(4.3) 的系数行列式是 Vandermonde 行列式, 且有

$$\begin{vmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{vmatrix} = \prod_{i=1}^{n} \prod_{j=0}^{i-1} (x_i - x_j) \neq 0.$$

因此, 方程组(4.3) 有唯一解$a_0, a_1, \cdots, a_n$, 即$P_n(x)$ 存在且唯一. ■

通过直接验证，我们有

定理 4.2　多项式

$$l_j(x) = \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i} \tag{4.4}$$

满足

$$l_j(x_i)=\delta_{ij}\triangleq\begin{cases}1, & i=j,\\ 0, & i\neq j,\end{cases}\qquad i,\ j=0,1,2,\cdots,n.$$

由定理 4.1 及定理 4.2 即可推得 Lagrange 插值多项式.

定理 4.3 代数插值问题(4.1)-(4.2) 的解

$$P_n(x)=\sum_{j=0}^{n}y_jl_j(x),\quad n>0, \tag{4.5}$$

其中$l_j(x)$ 由(4.4) 给出.

证明 因$l_j(x)$ 是一个n 次多项式，则(4.5) 的右端也是一个n 次多项式，且由定理 4.2 有

$$P_n(x_i)=\sum_{j=0}^{n}y_jl_j(x_i)=\sum_{j=0}^{n}y_j\delta_{ij}=y_i,\quad i=0,1,2,\cdots,n.$$

故由定理 4.1 可知$P_n(x)$ 为所求. ■

在 Lagrange 插值公式(4.5) 中，当n 取 1 或 2，我们分别称之为**线性插值公式**和**抛物插值公式**. 此外，Lagrange 插值多项式$P_n(x)$ 也可等价地写为：

$$P_n(x)=\sum_{j=0}^{n}y_j\frac{\omega_{n+1}(x)}{(x-x_j)\omega'_{n+1}(x_j)},\quad n>0, \tag{4.6}$$

其中$\omega_{n+1}(x)=\prod_{i=0}^{n}(x-x_i)$. 公式(4.5) 的计算程序见算法 4.1.

算法 4.1 Lagrange 插值算法

```
function y=lagrange(x0,y0,x)
 n=length(x0); m=length(x);
 for i=1:m
     z=x(i); s=0.0;
     for k=1:n
         p=1.0;
         for j=1:n
             if j~=k
                 p=p*(z-x0(j))/(x0(k)-x0(j));
             end
         end
         s=p*y0(k)+s;
     end
     y(i)=s;
 end
end
```

Lagrange 插值多项式$P_n(x)$ 是区间$[a,b]$ 上函数$f(x)$ 的一种逼近，除在插值节点x_i 处有$P_n(x_i)=f(x_i)$ 外，在其它点处$P_n(x)$ 与函数$f(x)$ 之间一般均存在着误差，该误差可由微积分原理分析导出.

定理 4.4 设$f(x)\in C^{(n)}([a,b])$, $f^{(n+1)}(x)$ 在(a,b) 内存在，$\{x_i\}_{i=0}^n\subseteq[a,b]$为Lagrange多项式$P_n(x)$的插值节点，则$P_n(x)$ 在$[a,b]$ 上的截断误差为

$$R_n(x)\triangleq\ f(x)-P_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x),\quad x\in[a,b],\tag{4.7}$$

其中ξ 在诸x_i 与x 之间.

证明 $\forall x\in[a,b]$, 若x 是插值节点，则(4.7) 显然成立；若x 是非插值节点，则作辅助函数

$$\Phi(t)=f(t)-P_n(t)-\frac{\omega_{n+1}(t)}{\omega_{n+1}(x)}[f(x)-P_n(x)],\quad t\in[a,b].$$

可验证：

$$\Phi(x)=\Phi(x_0)=\Phi(x_1)=\cdots=\Phi(x_n)=0.$$

因此，由 Rolle 定理可得$\Phi'(t)$ 在(a,b) 内至少有$n+1$ 个互异零点. 当依次反复应用 Rolle 定理时，我们可得$\Phi^{(n+1)}(t)$ 在区间$[\min\limits_{0\leqslant i\leqslant n}\{x_i,x\},\max\limits_{0\leqslant i\leqslant n}\{x_i,x\}]$ 上至少有一个零点ξ, 即

$$\Phi^{(n+1)}(\xi)=f^{(n+1)}(\xi)-\frac{(n+1)!}{\omega_{n+1}(x)}[f(x)-P_n(x)]=0.$$

由此即得(4.7) 式. ■

若$f(x)\in C^{(n+1)}([a,b])$，则由(4.7) 式进一步有$P_n(x)$ 的误差估计

$$|R_n(x)|\leqslant\frac{M}{(n+1)!}|\omega_{n+1}(x)|,\quad \text{其中}\ \ M=\max_{a\leqslant x\leqslant b}|f^{(n+1)}(x)|.\tag{4.8}$$

在误差表达式(4.7) 中, 当函数$f(x)$ 给定时, 因子$\omega_{n+1}(x)=\prod\limits_{i=0}^{n}(x-x_i)$ 直接影响到误差$R_n(x)$ 的大小. 一般而言，在插值区间中部使用插值，则插值精度较高；在插值区间两端或之外使用插值，则插值精度较低.

例 4.1 给定函数$f(x)=x(1+\cos x)$及插值节点

$$x_0=0,\ \ x_1=\frac{\pi}{8},\ \ x_2=\frac{\pi}{4},\ \ x_3=\frac{3\,\pi}{8},\quad\ x_4=\frac{\pi}{2}.$$

试构造Lagrange 插值多项式，给出其误差估计, 并由此计算$f\big(\frac{3\,\pi}{16}\big)$及其误差.

解 依题意可取$n=4$，函数$f(x)$ 在插值节点x_0,x_1,x_2,x_3,x_4 处的值依次为

$$y_0=0,\quad y_1=0.755505725716153,\quad y_2=1.340758530667244,$$

$$y_3=1.628935542509432,\quad y_4=1.570796326794897.$$

因此，其Lagrange 插值多项式为

$$P_4(x)=\sum_{j=0}^{4}y_jl_j(x),\quad\ \text{其中}\ \ l_j(x)=\prod_{i=0,i\neq j}^{4}\frac{x-x_i}{x_j-x_i}.\tag{4.9}$$

既然

$$f^{(6)}(x) = -6(\sin x + x\cos x) \leqslant 0, \quad x \in [0, \frac{\pi}{2}].$$

则$f^{(5)}(x)$在插值区间$[0, \frac{\pi}{2}]$ 上单调递减, 且

$$\max_{0\leqslant x\leqslant \pi/2} |f^{(5)}(x)| = \max\{|f^{(5)}(0)|, |f^{(5)}(\frac{\pi}{2})|\} = 5.$$

由式(4.8)可得误差估计

$$|R_4(x)| \leqslant \frac{1}{4!}\left|x\left(x-\frac{\pi}{8}\right)\left(x-\frac{\pi}{4}\right)\left(x-\frac{3\pi}{8}\right)\left(x-\frac{\pi}{2}\right)\right|. \tag{4.10}$$

应用算法4.1于公式(4.9)可得

$$f(\frac{3\pi}{16}) \approx P_4(\frac{3\pi}{16}) = 1.07915851119701,$$

且其误差

$$\left|R_4\left(\frac{3\pi}{16}\right)\right| = \left|f\left(\frac{3\pi}{16}\right) - P_4\left(\frac{3\pi}{16}\right)\right| \approx 3.338588315180413e-004.$$

若根据(4.10)估计其误差, 则有

$$\left|R_4\left(\frac{3\pi}{16}\right)\right| \leqslant \frac{1}{4!}\left|\frac{3\pi}{16}\left(\frac{\pi}{16}\right)\left(-\frac{\pi}{16}\right)\left(-\frac{3\pi}{16}\right)\left(-\frac{5\pi}{16}\right)\right| \approx 5.472058381771113e-004. \quad \blacksquare$$

§4.2 分段线性插值

在使用 Lagrange 插值公式(4.5) 时，其插值多项式的次数n 必须保持适度大小, n过大或过小均有可能导致大的插值误差.

例 4.2 考察函数

$$f(x) = \frac{\cos x}{\sqrt{1+(\sin x)^2}}, \quad x \in [-5, 5].$$

剖分区间$[-5, 5]$ 为10 等分，取其等分点为插值节点, 构造插值多项式$P_{10}(x)$, 函数$P_{10}(x)$和$f(x)$的图像见图4.1. 该图表明，虽然$P_{10}(x)$与被逼近函数$f(x)$ 在插值节点处取相同值，但整体逼近效果很差，且越靠近端点其逼近效果越差. 这种由于采用高次插值而产生较大误差的现象称为**Runge 现象**. $\blacksquare$

Runge 现象表明: 节点加密或大范围内使用高次插值不一定能保证插值函数精确逼近被插值函数. 为了保证所需插值精度, 一个可行办法是采用**分段低次插值**, 即首先在插值区间$[a, b]$ 上引入适当节点:

$$a = x_0 < x_1 < \cdots < x_n = b,$$

将$[a, b]$ 分为n 个小区间$[x_k, x_{k+1}]$, 然后在每个小区间$[x_k, x_{k+1}]$ 上使用低次多项式插值. 其中，诸点(x_k, y_k) 称为**型值点**，这里$y_k = f(x_k)$.

图 4.1 Runge 现象

以下，我们探讨分段线性插值，即在每个小区间$[x_k, x_{k+1}]$ 上使用一次多项式插值. 从几何观点来看，即是以过型值点(x_k, y_k) $(k = 0, 1, \cdots, n)$ 的折线$L_h(x)$ 去逼近曲线$f(x)$. 记$h_k = x_{k+1} - x_k$, $h = \max_{0\leqslant k\leqslant n-1}\{h_k\}$, 今求一折线函数$L_h(x)$，其满足下列条件：

1) $L_h(x) \in C([a, b])$，

2) $L_h(x_k) = y_k$, $k = 0, 1, \cdots, n$,

3) $L_h(x)$ 在每个插值区间$[x_k, x_{k+1}]$ 上为一次函数.

易知，函数$L_h(x)$ 在区间$[x_k, x_{k+1}]$ 上的表达式为

$$L_h(x) = \frac{x - x_{k+1}}{x_k - x_{k+1}} y_k + \frac{x - x_k}{x_{k+1} - x_k} y_{k+1}, \quad x \in [x_k, x_{k+1}], \tag{4.11}$$

在整个插值区间$[a, b]$ 上的表达式为

$$L_h(x) = \sum_{k=0}^{n} l_k(x) y_k, \quad x \in [a, b], \tag{4.12}$$

其中$l_k(x)$ 为$L_h(x)$ 在$[a, b]$ 上的基函数，其表达式为

$$l_k(x) = \begin{cases} \dfrac{x - x_{k-1}}{x_k - x_{k-1}}, & x_{k-1} \leqslant x \leqslant x_k \ (k \neq 0), \\ \dfrac{x - x_{k+1}}{x_k - x_{k+1}}, & x_k \leqslant x \leqslant x_{k+1} \ (k \neq n), \\ 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

由插值基函数$l_k(x)$ 的表达式可知

$$l_k(x_j) = \delta_{kj} \triangleq \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases}$$

上式表明$l_k(x)$ 具有**局部非零性**, 即$l_k(x)$ 只在x_k 附近不为零，在其它地方均为零. 这种局部非零性质使得插值点处的误差可控制在一个局部区域内, 而 Lagrange 插值公式中的基函数$l_k(x)$ 不

图 4.2 分段线性插值$L_1(x)$

具有该性质. 此外, 分段线性插值计算简单，适用于光滑性要求不高的插值问题. 由文献[13]可知，当我们采用 MATLAB 实现分段线性插值计算时, 可直接采用命令$interp1(x, y, \tilde{x}, 'linear')$或$interp1(x, y, \tilde{x})$, 其中$x$ 为节点向量, y 为对应的向量函数值, $\tilde{x}$ 为插值点. 在例 4.2 同样的插值条件下，利用分段线性插值$L_1(x)$ 逼近函数$f(x) = \frac{\cos x}{\sqrt{1+(\sin x)^2}}$，其二者对应的曲线见图 4.2. 该图表明, $L_1(x)$ 的逼近效果要优于$P_{10}(x)$.

§4.3 Newton 插值公式

为提高插值精度，人们在计算过程中往往需要增加或减少插值节点，而Lagrange插值公式在上机计算时不易变动节点. 因此，本章将通过对Lagrange插值公式实施恒等变形，而引入Newton插值公式. 为此，首先我们需要介绍差商与差分的概念及其相关性质.

定义 4.1 设$x_0, x_1, \cdots, x_n$ 为区间$[a, b]$ 上的互异节点, 则称$f(x_i)$ 为$f(x)$ 在x_i 处的零阶差商；称

$$\frac{f(x_i) - f(x_j)}{x_i - x_j}$$

为函数$f(x)$ 在x_i, x_j 处的一阶差商, 记为$f[x_i, x_j]$; 一般称

$$f[x_0, x_1, \cdots, x_n] \triangleq \frac{f[x_0, x_1, \cdots, x_{n-1}] - f[x_1, x_2, \cdots, x_n]}{x_0 - x_n}$$

为$f(x)$ 在$x_0, x_1, \cdots, x_n$ 处的n 阶差商.

利用差商的定义及数学归纳法可以直接获得n 阶差商的表达式.

定理 4.5 n 阶差商

$$f[x_0, x_1, \cdots, x_n] = \sum_{j=0}^{n} \frac{f(x_j)}{\omega'_{n+1}(x_j)}, \quad \text{这里 } \omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i) \tag{4.14}$$

表达式(4.14)表明差商的值与节点$\{x_i\}_{i=0}^n$ 的排序无关，即具对称性. 此外, 由差商定义有

$$\begin{aligned}
f(x) &= f(x_0) + f[x_0, x](x - x_0),\\
f[x_0, x] &= f[x_0, x_1] + f[x_0, x_1, x](x - x_1),\\
f[x_0, x_1, x] &= f[x_0, x_1, x_2] + f[x_0, x_1, x_2, x](x - x_2)\\
&\vdots\\
f[x_0, x_1, \cdots, x_{n-1}, x] &= f[x_0, x_1, \cdots, x_n] + f[x_0, x_1, \cdots, x_n, x](x - x_n).
\end{aligned}$$

从上面最后一个式子逐次往上代入得

$$f(x) = N_n(x) + R_n(x), \tag{4.15}$$

其中

$$N_n(x) = f(x_0) + f[x_0, x_1](x - x_0) + \cdots + f[x_0, x_1, \cdots, x_n]\omega_n(x), \tag{4.16}$$

$$R_n(x) = f[x_0, x_1, \cdots, x_n, x]\omega_{n+1}(x), \quad \omega_n(x) = \prod_{i=0}^{n-1}(x - x_i), \tag{4.17}$$

由于

$$f(x_i) - N_n(x_i) = R_n(x_i) = 0, \quad i = 0, 1, 2, \cdots, n,$$

且$N_n(x)$ 为n 次多项式，则由定理4.1 可知

$$N_n(x) \equiv P_n(x), \quad \forall x \in [a, b]. \tag{4.18}$$

因此, $N_n(x)$ 可作为$f(x)$的n 次插值多项式，称之为n **次Newton 插值公式**. 由(4.18) 及定理4.4，其截断误差

$$R_n(x) = f(x) - N_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x), \quad x \in [a, b], \tag{4.19}$$

其中ξ 在诸x_i 与x 之间.

比较式(4.17)与式(4.19)可直接推得n 阶差商与n 阶导数的如下等价关系.

定理 4.6 若$f(x)$ 在$[a, b]$ 上n 阶可导，且诸节点$\{x_i\}_{i=0}^n \subseteq [a, b]$, 则

$$f[x_0, x_1, \cdots, x_n] = \frac{f^{(n)}(\xi)}{n!}, \quad \xi\text{在诸}x_i\text{之间}. \tag{4.20}$$

由式(4.16)可知Newton 插值公式具有如下**承袭性**：

$$N_n(x) = N_{n-1}(x) + f[x_0, x_1, \cdots, x_n]\prod_{i=0}^{n-1}(x - x_i), \quad x \in [a, b]. \tag{4.21}$$

因此, 在实际应用Newton 插值公式(4.16) 进行计算时, 其每增加或减少一个插值节点，只需在上一步计算结果的基础上加上或减去一个形如$f[x_0, x_1, \cdots, x_n]\prod_{i=0}^{n-1}(x - x_i)$的相应项, 且预先按照如下箭头所示顺序计算其差商:

$$
\begin{array}{lllllll}
f(x_0) & & & & & & \\
 & \searrow & & & & & \\
f(x_1) & \rightarrow & f[x_0,x_1] & & & & \\
 & \searrow & & \searrow & & & \\
f(x_2) & \rightarrow & f[x_1,x_2] & \rightarrow & f[x_0,x_1,x_2] & & \\
 & \searrow & & \searrow & & \searrow & \\
f(x_3) & \rightarrow & f[x_2,x_3] & \rightarrow & f[x_1,x_2,x_3] & \rightarrow & f[x_0,x_1,x_2,x_3] \\
 & \ddots & & \ddots & & \ddots & & \ddots
\end{array}
$$

尔后, 将上表中处于对角线位置的差商值代入公式(4.16) 计算出插值, 其计算程序见算法4.2.

算法 4.2 Newton 插值算法

```
function np=newton_interpolation(x0,y0,x)
 n=length(x0); a(1)=y0(1);
for k=1:n-1
    d(k,1)=(y0(k+1)-y0(k))/(x0(k+1)-x0(k));
end
for j=2:n-1
    for k=1:n-j
        d(k,j)=(d(k+1,j-1)-d(k,j-1))/(x0(k+j)-x0(k));
    end
end
d
for j=2:n
    a(j)=d(1,j-1);
end
 b(1)=1; c(1)=a(1);
for j=2:n
    b(j)=(x-x0(j-1)).*b(j-1);
    c(j)=a(j).*b(j);
end
np=sum(c)
```

例 4.3 已知$f(x)=\sqrt{1+ch^2x}$. 试取插值节点

$$x_0=0.35,\quad x_1=0.50,\quad x_2=0.65,\quad x_3=0.80,\quad x_4=0.95,$$

构造4 次Newton 插值多项式, 由此计算$f(0.7)$的逼近值, 并指出其绝对误差.

解 其插值节点x_k所对应的函数值$y_k=f(x_k)$依次为:

$$y_0=1.458624181485920,\quad y_1=1.507163002932205,\quad y_2=1.576533258502121,$$

$$y_3 = 1.669949770381565, \quad y_4 = 1.791330724733843.$$

据此及算法4.2 可得差商值

一阶差商	二阶差商	三阶差商	四阶差商
0.3235921429752330			
0.4624683704661070	0.4629207583029130		
0.6227767458629592	0.5343612513228406	0.1587566511553948	
0.8092063623485214	0.6214320549518743	0.1934906747311859	0.05789003929298514

因此, 其4 次Newton 插值项式为

$$\begin{aligned} N_4(x) &= 1.458624181485920 \\ &+ 0.3235921429752330(x-0.35) \\ &+ 0.4629207583029130(x-0.35)(x-0.50) \\ &+ 0.1587566511553948(x-0.35)(x-0.50)(x-0.65) \\ &+ 0.05789003929298514(x-0.35)(x-0.50)(x-0.65)(x-0.80), \end{aligned}$$

且$f(0.7)$ 的逼近值$N_4(0.7) \approx 1.604821271373747$, 其绝对误差为

$$|R_4(0.7)| = |f(0.7) - N_4(0.7)| \approx 6.481586261042338e-007. \quad \blacksquare$$

当插值节点取为等距节点时，Newton插值公式可被简化. 为此，我们需要下列预备性结果.

定理 4.7 当取等距离节点$x_i = x_0 + ih$ $(i = 0, 1, \cdots, n)$ 时，差分与差商存在着下列关系：

$$f[x_0, x_1, \cdots, x_n] = \frac{\triangle^n f(x_0)}{n! h^n}, \tag{4.22}$$

$$f[x_0, x_1, \cdots, x_n] = \frac{\nabla^n f(x_n)}{n! h^n}. \tag{4.23}$$

证明 我们利用数学归纳法证明(4.22). 事实上，当$n = 1$ 时，有

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{\triangle f(x_0)}{h}.$$

假设$n = m - 1$ 时, 式(4.22) 成立，则有

$$f[x_0, x_1, \cdots, x_{m-1}] = \frac{\triangle^{m-1} f(x_0)}{(m-1)! h^{m-1}}, \quad f[x_1, x_2, \cdots, x_m] = \frac{\triangle^{m-1} f(x_1)}{(m-1)! h^{m-1}}.$$

于是

$$f[x_0, x_1, \cdots, x_m] = \frac{f[x_1, \cdots, x_m] - f[x_0, \cdots, x_{m-1}]}{x_m - x_0} = \frac{\triangle^{m-1} f(x_1) - \triangle^{m-1} f(x_0)}{m! h^m} = \frac{\triangle^m f(x_0)}{m! h^m}.$$

因此，等式(4.22) 获证. 类似地, 等式(4.23) 也可被证. $\blacksquare$

若插值点$x \in (x_0, x_1)$，则可令$x = x_0 + th$ $(0 < t < 1)$, 应用该式及(4.22) 到(4.16) 得

$$N_n(x_0 + th) = f(x_0) + t\Delta f(x_0) + \frac{t(t-1)}{2!}\Delta^2 f(x_0) + \cdots + \frac{\prod_{i=0}^{n-1}(t-i)}{n!}\Delta^n f(x_0). \tag{4.24}$$

该公式称为**Newton 前插公式**，其余项为

$$R_n(x) = \frac{t(t-1)\cdots(t-n)}{(n+1)!}h^{n+1}f^{(n+1)}(\xi), \quad \xi \in (x_0, x_n).$$

若插值点$x \in (x_{n-1}, x_n)$，则可令$x = x_n + th$ $(-1 < t < 0)$ ，应用该式及(4.23) 到(4.16) 得

$$N_n(x_n + th) = f(x_n) + t\nabla f(x_n) + \frac{t(t+1)}{2!}\nabla^2 f(x_n) + \cdots + \frac{\prod_{i=0}^{n-1}(t+i)}{n!}\nabla^n f(x_n). \tag{4.25}$$

该公式称为**Newton 后插公式**，其余项为

$$R_n(x) = \frac{t(t+1)\cdots(t+n)}{(n+1)!}h^{n+1}f^{(n+1)}(\xi), \quad \xi \in (x_0, x_n).$$

§4.4 Hermite 插值公式

为在不增加插值节点的前提下提高插值精度, 一种可行的方法是: 不但要求插值函数$H(x)$与被插值函数$f(x)$ 在节点$x_0, x_1, \cdots, x_n$ 处相等, 而且要求导数值也在其节点处相等, 即

$$H^{(j)}(x_i) = f^{(j)}(x_i), \quad j = 0, 1, \cdots, m_i; \quad i = 0, 1, \cdots, n. \tag{4.26}$$

这种方法称为**Hermite 插值法**. 若其诸节点x_i 互异, 且插值函数$H(x)$ 取为代数多项式$\Phi(x)$, 则确定多项式$\Phi(x)$ 共需$n + \sum_{i=0}^{n} m_i + 1$ 个插值条件. 为简单见，下面我们将仅讨论$2n+1$ 次Hermite 代数插值问题

$$H(x_i) = f(x_i), \quad H'(x_i) = f'(x_i), \quad i = 0, 1, \cdots, n. \tag{4.27}$$

记$y_j = f(x_j)$, $y'_j = f'(x_j)$, 则下列结果给出了$2n+1$ 次Hermite 代数插值问题的解.

定理 4.8 $2n+1$ 次 Hermite 插值问题(4.27) 的解$H_{2n+1}(x)$ 存在且唯一，其表达式为

$$H_{2n+1}(x) = \sum_{j=0}^{n}[y_j + (x - x_j)(y'_j - 2y_j l'_j(x_j))]l_j^2(x), \tag{4.28}$$

其中$l_j(x) = \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i}$.

证明 由插值条件(4.27) 可设

$$H_{2n+1}(x) = \sum_{j=0}^{n}[y_j\hat{l}_j(x) + y'_j\tilde{l}_j(x)], \tag{4.29}$$

其中

$$\hat{l}_j(x_i) = \tilde{l}'_j(x_i) = \delta_{ij}, \quad \hat{l}'_j(x_i) = \tilde{l}_j(x_i) = 0, \quad i, j = 0, 1, \cdots, n.$$

以下确定$\hat{l}_j(x)$ 和$\tilde{l}_j(x)$. 令$\hat{l}_j(x)=(ax+b)l_j^2(x)$, 易验证其为$2n+1$ 次多项式，且满足

$$\hat{l}_j(x_i)=\hat{l}_j'(x_i)=0,\quad i\neq j.$$

因此只须选择常数a,b 使得

$$\begin{cases}(ax_j+b)l_j^2(x_j)=1,\\ l_j(x_j)[al_j(x_j)+2(ax_j+b)l_j'(x_j)]=0,\end{cases}$$

即

$$\begin{cases}ax_j+b=1,\\ a+2(ax_j+b)l_j'(x_j)=0,\end{cases}$$

解之得$a=-2l_j'(x_j),\ b=1+2x_jl_j'(x_j)$. 类似地，若令$\tilde{l}_j(x)=(cx+d)l_j^2(x)$, 由已知条件可导出$c=1,\ d=-x_j$. 将已求得的$a,b,c,d$值代入(4.29) 即得(4.28) 式. ■

Hermite 插值公式(4.28) 的误差由下列命题给出.

定理 4.9 设$f(x)\in C^{(2n+1)}([a,b])$, 且$f^{(2n+2)}(x)$ 在(a,b) 内存在，$\{x_i\}_{i=0}^n\subseteq[a,b]$为Hermite 插值多项式$H_{2n+1}(x)$ 的互异节点，则$H_{2n+1}(x)$ 在$[a,b]$ 上的截断误差为

$$R_{2n+1}(x)\triangleq\ f(x)-H_{2n+1}(x)=\frac{f^{(2n+2)}(\xi)}{(2n+2)!}\omega_{n+1}^2(x),\quad x\in[a,b],\tag{4.30}$$

其中ξ 在诸x_i 与x 之间, $\omega_{n+1}(x)=\prod\limits_{i=0}^{n}(x-x_i)$.

证明 $\forall x\in[a,b]$, 若x 是插值节点，则(4.30) 显然成立；若x 是非插值节点，则作辅助函数

$$\Psi(t)=f(t)-H_{2n+1}(t)-\frac{\omega_{n+1}^2(t)}{\omega_{n+1}^2(x)}[f(x)-H_{2n+1}(x)],\quad t\in[a,b].$$

可验证：

$$\Psi(x)=0,\quad \Psi(x_i)=0,\quad \Psi'(x_i)=0,\quad i=0,1,2,\cdots,n.$$

因此，由 Rolle 定理可知$\Psi'(t)$ 在节点$x_0,x_1,\cdots,x_n$ 之间至少有$n+1$ 个零点, 且诸节点本身仍为$\Psi'(t)$ 的零点. 故$\Psi'(t)$ 在$[\min\limits_{0\leqslant i\leqslant n}\{x_i,x\},\max\limits_{0\leqslant i\leqslant n}\{x_i,x\}]$ 内共有$2n+2$ 个互异零点. 同理，$\Psi''(t)$ 在$[\min\limits_{0\leqslant i\leqslant n}\{x_i,x\},\max\limits_{0\leqslant i\leqslant n}\{x_i,x\}]$ 内共有$2n+1$ 个互异零点. 通过反复应用 Rolle 定理，我们可知$\Psi^{(2n+2)}(t)$ 在区间$[\min\limits_{0\leqslant i\leqslant n}\{x_i,x\},\max\limits_{0\leqslant i\leqslant n}\{x_i,x\}]$ 上至少存在一个零点ξ, 使得

$$\Psi^{(2n+2)}(\xi)=f^{(2n+2)}(\xi)-\frac{(2n+2)!}{\omega_{n+1}^2(x)}[f(x)-H_{2n+1}(x)]=0.$$

由此即得(4.30) 式. ■

记

$$a_j(x)=l_j^2(x),\quad b_j=l_j'(x_j)=\sum_{i=0,i\neq j}^{n}\frac{1}{x_j-x_i}.$$

Hermite 插值算法的计算程序见算法 4.3.

算法 4.3 Hermite 插值算法

```
function y=hermite_interpolation(x0,y0,y1,x)
 n=length(x0); m=length(x);
for k=1:m
    yk=0;
  for i=1:n
    a=1; b=0;
     for j=1:n
       if j~=i
         a=a*((x(k)-x0(j))/(x0(i)-x0(j)))^2; b=1/(x0(i)-x0(j))+b;
       end
     end
     yk=yk+a*((x0(i)-x(k))*(2*b*y0(i)-y1(i))+y0(i));
  end
  y(k)=yk;
end
```

例 4.4 设函数$f(x)=x\ln(1+x)$. 试取插值节点$x_0=1,\ x_1=1.2,\ x_2=1.4$构造5 次Hermite 插值多项式，求$f(1.3)$ 的逼近值，并给出其绝对误差.

解 函数$f(x)$及$f'(x)$在插值节点x_0,x_1,x_2 处的值为

j	0	1	2
y_j	0.6931471805599453	0.9461488324371243	1.225656232295460
y'_j	1.193147180559945	1.333911905818816	1.458802070687233

因此，其 5 次 Hermite 插值多项式为

$$H_5(x)=\sum_{j=0}^{2}[y_j+(x-x_j)(y'_j-2y_jl'_j(x_j))]l_j^2(x),\quad \text{其中 } l_j(x)=\prod_{i=0,i\neq j}^{2}\frac{x-x_i}{x_j-x_i}.$$

由此$f(1.3)$ 的逼近值$H_5(1.3)=1.082781841233770$, 其绝对误差为

$$|R_5(1.3)|\triangleq\ |f(1.3)-H_5(1.3)|\approx 1.858186493564062e-008.\quad\blacksquare$$

在小范围内采用低次 Hermite 插值是恰当的，但在大范围内则不然，此时试图增加节点引入高次插值其逼近效果同样也不理想. 为此，本节将又一次采用分段插值的思想来解决这一问题，我们将以分段三次 Hermite 插值为例来阐明其算法.

在插值区间$[a,b]$ 上引入适当节点: $a=x_0<x_1<\cdots<x_n=b$, 将$[a,b]$ 分为n 个小区间$[x_k,x_{k+1}]$, 然后在每个小区间$[x_k,x_{k+1}]$ 上使用三次插值$H_3(x)$ 逼近函数$f(x)$，并要求其满足：

1) $H_3(x)\in C^{(1)}([a,b])$，

2) $H_3(x_k)=y_k,\ \ H'_3(x_k)=y'_k,\quad k=0,1,\cdots,n,$

3) $H_3(x)$ 在每个插值区间$[x_k, x_{k+1}]$ 上为三次多项式函数.
由(4.28), 区间$[x_k, x_{k+1}]$ 上的三次 Hermite 插值为

$$H_3(x) = y_k\alpha_k(x) + y_{k+1}\beta_k(x) + y'_k\gamma_k(x) + y'_{k+1}\eta_k(x), \quad x \in [x_k, x_{k+1}], \tag{4.31}$$

其中

$$\alpha_k(x) = \left(1 + 2\frac{x - x_k}{x_{k+1} - x_k}\right)\left(\frac{x - x_{k+1}}{x_k - x_{k+1}}\right)^2, \quad \gamma_k(x) = (x - x_k)\left(\frac{x - x_{k+1}}{x_k - x_{k+1}}\right)^2,$$

$$\beta_k(x) = \left(1 + 2\frac{x - x_{k+1}}{x_k - x_{k+1}}\right)\left(\frac{x - x_k}{x_{k+1} - x_k}\right)^2, \quad \eta_k(x) = (x - x_{k+1})\left(\frac{x - x_k}{x_{k+1} - x_k}\right)^2.$$

由此，整个插值区间$[a, b]$ 上的分段三次 Hermite 插值的表达式为

$$H_3(x) = \sum_{k=0}^{n}[y_k\tilde{\alpha}_k(x) + y'_k\tilde{\beta}_k(x)], \quad x \in [a, b]. \tag{4.32}$$

其中

$$\tilde{\alpha}_k(x) = \begin{cases} \beta_{k-1}(x), & x_{k-1} \leqslant x \leqslant x_k \ (k \neq 0), \\ \alpha_k(x), & x_k \leqslant x \leqslant x_{k+1} \ (k \neq n), \\ 0, & otherwise, \end{cases}$$

$$\tilde{\beta}_k(x) = \begin{cases} \eta_{k-1}(x), & x_{k-1} \leqslant x \leqslant x_k \ (k \neq 0), \\ \gamma_k(x), & x_k \leqslant x \leqslant x_{k+1} \ (k \neq n), \\ 0, & otherwise. \end{cases}$$

分段三次Hermite插值具有分段线性插值类似的局部非零性，但前者比后者具有更高的光滑度和计算精度. 参照文献[13]，当采用MATLAB 实现分段三次Hermite 插值时, 我们可直接使用MATLAB 命令$interp1(x, y, \tilde{x},\ 'cubic')$ 或$interp1(x, y, \tilde{x},\ 'pchip')$, 其中$x$ 为节点向量, y 为对应的节点向量函数值, $\tilde{x}$为插值点.

例 4.5 给定函数$f(x) = \frac{x}{\sqrt{1+\cos^2 x}}$及插值节点$x_i = i\,\pi\,/10\ (i = 0, 1, \cdots, 5)$. 试分别利用11次Hermite 插值多项式和分段3 次Hermite 插值多项式计算$f(\,\pi\,/3)$ 的逼近值.

解 函数$f(x), f'(x)$ 在诸插值节点x_i 处的值为

i	y_i	y'_i
0	0	0.7071067811865474
1	0.2276451569741872	0.7327198693970755
2	0.4884784033206227	0.8027921877339462
3	0.8125136162744258	0.8965076700460836
4	1.200619402274652	0.9780537826528916
5	1.570796326794897	1

因此, 利用11 次Hermite 插值多项式可得$f(\pi/3)$ 的逼近值$\tilde{H}_{11}(\pi/3) \approx 0.9129817259566481$, 其绝对误差为

$$|\tilde{R}_{11}(\pi/3) = |f(\pi/3) - \tilde{H}_{11}(\pi/3)| \approx 2.366023818211527e-002.$$

利用分段3次Hermite插值多项式并直接运行MATLAB命令$interp1(x, y, \tilde{x}, 'cubic')$ 可得$f(\pi/3)$ 的逼近值$H_3(\pi/3) \approx 0.9373889857742555$, 其绝对误差为

$$|R_3(\pi/3)| = |f(\pi/3) - H_3(\pi/3)| \approx 7.470216354921178e-003.$$

此时，分段3次Hermite插值的计算精度要高于11 次Hermite 插值的计算精度. ■

§4.5 样条插值

分段低次插值虽可克服高次插值的龙格现象，但其光滑度往往不能满足工程技术中的外形设计要求. 为此，本节介绍基于三次样条函数的插值技术.

定义 4.2 若函数$S(x) \in C^{(2)}([a,b])$, 且对于$[a,b]$上的一个分划:

$$\Pi: \quad a = x_0 < x_1 < \cdots < x_n = b,$$

$S(x)$在每个小区间$[x_k, x_{k+1}]$ 上为三次多项式，则称$S(x)$ 是基于分划Π 的三次多项式样条函数，其中$x_1, x_2, \cdots, x_{n-1}$ 称为内节点，x_0, x_n 称为边界节点.

以下, 我们利用区间$[a,b]$上的三次多项式样条函数$S(x)$逼近函数$f(x)$. 记

$$y_k = f(x_k), \quad y'_k = f'(x_k), \quad y''_k = f''(x_k), \quad k = 0, 1, \cdots, n,$$

并设

$$S(x_k) = y_k, \quad k = 0, 1, \cdots, n. \tag{4.33}$$

由于$S(x)$ 在每个小区间$[x_k, x_{k+1}]$ 上为三次多项式，故要确定$[a,b]$ 上的三次样条插值函数总计需$4n$ 个条件. 插值条件(4.33) 给出了$n+1$ 个条件. 此外, 在插值内节点x_k处, 既然$S(x) \in C^{(2)}([a,b])$, 因此有$3(n-1)$ 个条件成立：

$$S(x_k-0) = S(x_k+0), \quad S'(x_k-0) = S'(x_k+0), \quad S''(x_k-0) = S''(x_k+0), \quad k = 1, \cdots, n-1. \tag{4.34}$$

至此, 共获得$4n-2$ 个方程, 剩余2 个条件可由下列边界条件获得：

$$S'(x_0+0) = y'_0, \quad S'(x_n-0) = y'_n \tag{4.35}$$

或

$$S''(x_0+0) = y''_0, \quad S''(x_n-0) = y''_n \tag{4.36}$$

据此，我们将插值问题分为I型样条插值问题$\{(4.33), (4.34), (4.35)\}$ 及II 型样条插值问题$\{(4.33), (4.34), (4.36)\}$.

首先, 我们考虑I型样条插值问题. 记$S'(x_k) = m_k$, 则据分段三次Hermite 插值(4.32)有

$$S(x) = \sum_{k=0}^{n} [y_k \tilde{\alpha}_k(x) + m_k \tilde{\beta}_k(x)], \tag{4.37}$$

其中m_k由条件(4.35) 及$S''(x_k-0)=S''(x_k+0)$ $(k=1,\cdots,n-1)$ 确定. 由(4.31)，$S(x)$ 在区间$[x_k,x_{k+1}]$ 上的表达式为

$$S(x)=y_k\alpha_k(x)+y_{k+1}\alpha_{k+1}(x)+m_k\beta_k(x)+m_{k+1}\beta_{k+1}(x).$$

对$S(x)$ 求二阶导数得

$$\begin{aligned}&S''(x)=y_k\alpha_k''(x)+y_{k+1}\alpha_{k+1}''(x)+m_k\beta_k''(x)+m_{k+1}\beta_{k+1}''(x)\\&=\frac{6x-2x_k-4x_{k+1}}{h_k^2}m_k+\frac{6x-4x_k-2x_{k+1}}{h_k^2}m_{k+1}+\frac{6(x_k+x_{k+1}-2x)}{h_k^3}(y_{k+1}-y_k),\end{aligned}\tag{4.38}$$

其中$h_k=x_{k+1}-x_k$. 同理，可得$S''(x)$ 在区间$[x_{k-1},x_k]$ 上的表达式

$$S''(x)=\frac{6x-2x_{k-1}-4x_k}{h_{k-1}^2}m_{k-1}+\frac{6x-4x_{k-1}-2x_k}{h_{k-1}^2}m_k+\frac{6(x_{k-1}+x_k-2x)}{h_{k-1}^2}(y_k-y_{k-1}).\tag{4.39}$$

由连续性条件: $S''(x_k-0)=S''(x_k+0)$ 有

$$\frac{1}{h_{k-1}}m_{k-1}+2\Big(\frac{1}{h_{k-1}}+\frac{1}{h_k}\Big)m_k+\frac{1}{h_k}m_{k+1}=3\Big(\frac{y_{k+1}-y_k}{h_k^2}+\frac{y_k-y_{k-1}}{h_{k-1}^2}\Big).\tag{4.40}$$

记

$$a_k=\frac{h_{k-1}}{h_{k-1}+h_k},\quad b_k=3\left[(1-a_k)\frac{y_k-y_{k-1}}{h_{k-1}}+a_k\frac{y_{k+1}-y_k}{h_k}\right],$$

则(4.40)可写成

$$(1-a_k)m_{k-1}+2m_k+a_km_{k+1}=b_k,\quad k=1,2,\cdots,n-1,\tag{4.41}$$

该方程组称为**三转角方程**. 联立(4.35) 和(4.41) 得方程组

$$\begin{pmatrix}2&a_1&&&&\\1-a_2&2&a_2&&&\\&\ddots&\ddots&\ddots&&\\&&1-a_{n-2}&2&a_{n-2}\\&&&1-a_{n-1}&2\end{pmatrix}\begin{pmatrix}m_1\\m_2\\\vdots\\m_{n-2}\\m_{n-1}\end{pmatrix}=\begin{pmatrix}b_1-(1-a_1)y_0'\\b_2\\\vdots\\b_{n-2}\\b_{n-1}-a_{n-1}y_n'\end{pmatrix}.\tag{4.42}$$

该方程组为三对角型方程组, 因此可利用追赶法求解，所求得的诸m_i 代回到(4.37) 即得 I 型样条插值问题的解. 若插值问题未给出边界值y_0', y_n'，则可采用如下近似外推公式

$$y_0'\approx\left(1+\frac{h_0}{h_1}\right)m_1-\frac{h_0}{h_1}m_2,\quad y_n'\approx\left(1+\frac{h_{n-1}}{h_{n-2}}\right)m_{n-1}-\frac{h_{n-1}}{h_{n-2}}m_{n-2}$$

分别替代(4.42)中的y_0', y_n'.

下面我们继续考虑II型插值问题. 由于$S(x)$ 在区间$[x_k,x_{k+1}]$ 上是三次多项式，因此$S''(x)$ 在$[x_k,x_{k+1}]$ 上是线性函数. 若记$S''(x_k)=M_k$, 则有

$$S''(x)=M_k\frac{x_{k+1}-x}{h_k}+M_{k+1}\frac{x-x_k}{h_k},\quad x\in[x_k,x_{k+1}].\tag{4.43}$$

对式(4.43) 积分两次, 并利用$S(x_i)=y_i\ (i=k,k+1)$ 可得

$$\begin{aligned}S(x)=&M_k\frac{(x_{k+1}-x)^3}{6h_k}+M_{k+1}\frac{(x-x_k)^3}{h_k}+\left(y_k-\frac{M_kh_k^2}{6}\right)\frac{x_{k+1}-x}{h_k}\\&+\left(y_{k+1}-\frac{M_{k+1}h_k^2}{6}\right)\frac{x-x_k}{h_k},\quad x\in[x_k,x_{k+1}],\ \ k=0,1,\cdots,n-1.\end{aligned}\tag{4.44}$$

对$S(x)$ 求导得

$$S'(x)=-M_k\frac{(x_{k+1}-x)^2}{2h_k}+M_{k+1}\frac{(x-x_k)^2}{2h_k}+\frac{y_{k+1}-y_k}{h_k}-\frac{M_{k+1}-M_k}{6}h_k,\ \ x\in[x_k,x_{k+1}].\tag{4.45}$$

同理可得$S'(x)$ 在$[x_{k-1},x_k]$ 上的表达式

$$S'(x)=-M_{k-1}\frac{(x_k-x)^2}{2h_{k-1}}+M_k\frac{(x-x_{k-1})^2}{2h_{k-1}}+\frac{y_k-y_{k-1}}{h_{k-1}}-\frac{M_k-M_{k-1}}{6}h_{k-1},\ \ x\in[x_{k-1},x_k].\tag{4.46}$$

记

$$a_k=\frac{h_k}{h_{k-1}+h_k},\quad b_k=\frac{6}{h_k+h_{k-1}}\left(\frac{y_{k+1}-y_k}{h_k}-\frac{y_k-y_{k-1}}{h_{k-1}}\right),$$

则由连续性条件：$S'(x_k-0)=S'(x_k+0)$得

$$(1-a_k)M_{k-1}+2M_k+a_kM_{k+1}=b_k,\ \ k=1,2,\cdots,n-1.\tag{4.47}$$

该方程组称为**三弯矩方程**. 联立(4.47) 与(4.36) 得方程组

$$\begin{pmatrix}2&a_1&&&&\\1-a_2&2&a_2&&&\\&\ddots&\ddots&\ddots&&\\&&1-a_{n-2}&2&a_{n-2}\\&&&1-a_{n-1}&2\end{pmatrix}\begin{pmatrix}M_1\\M_2\\\vdots\\M_{n-2}\\M_{n-1}\end{pmatrix}=\begin{pmatrix}b_1-(1-a_1)y_0''\\b_2\\\vdots\\b_{n-2}\\b_{n-1}-a_{n-1}y_n''\end{pmatrix}.\tag{4.48}$$

利用追赶法解该三对角方程组可求得诸M_i, 将其代回到(4.44) 即得 II 型样条插值问题的解. 若插值问题未给出边界值y_0'', y_n'', 则可采用如下近似外推公式

$$y_0''\approx\left(1+\frac{h_0}{h_1}\right)M_1-\frac{h_0}{h_1}M_2,\quad y_n''\approx\left(1+\frac{h_{n-1}}{h_{n-2}}\right)M_{n-1}-\frac{h_{n-1}}{h_{n-2}}M_{n-2}$$

分别替代(4.48)中的y_0'', y_n''.

根据上面对三次样条插值方法的陈述，人们可自行编制出其算法代码. 为简单见, 这里我们仅介绍几个常用的计算三次样条插值的Matlab 命令. 以下记$X0,Y0$ 为已知数据点，X 为插值点，Y 为相应的函数值，y_a',y_a'' 分别为左边界点处的一、二阶导数值，y_b',y_b''分别为右边界点处的一、二阶导数值. 如果三次样条插值没有给出边界条件, 则据[13]通常采用命令$interp1(X0,Y0,X,'spline')$或$spline(X0,Y0,X)$. 若给出了边界处的一阶导数值$y_a',y_b'$，则可采用命令$c1=csape(X0,Y0,'complete',[y_a',y_b'])$及$ppval(c1,X)$. 若给出了边界处的二阶导数值$y_a'',y_b''$, 则可采用命令$c2=csape(X0,Y0,'second',[y_a'',y_b''])$及$ppval(c2,X)$.

例 4.6 已知某轮船断面轮廓线的型值点座标如下:

x	0	1.5	3	4.5	6	7.5	9	10.5	12	13.5	15
y	0.28	1.75	3.43	3.29	4.48	6.00	7.37	8.48	9.27	9.60	9.92

试取步长$h = 0.1$，利用三次样条插值方法绘出其轮廓线.

解 利用三次样条插值的Matlab 命令$spline(X0, Y0, x)$可编制如下程序：

```
X0=[0   1.5   3   4.5   6   7.5   9   10.5   12   13.5   15]';
Y0=[0.28   1.75  3.43  3.29   4.48   6.0   7.37   8.48   9.27   9.60   9.92]';
x=0:0.1:15;
y=spline(X0,Y0,x);
plot(X0,Y0,'ro',x,y,'-.b');
xlabel('x'); ylabel('y');
```

运行上述程序得其船断面的轮廓线(见图4.3). ■

§4.6 曲线拟合方法

在数据统计分析中，我们常常需要知道几组数据间的精确函数关系. 由于主观或客观原因，这些数据往往带有误差，利用这些数据构造的插值公式因此保留了这些误差. 事实上, 在实际构造逼近函数时, 我们不必一定要求逼近函数的曲线过型值点, 而只需从“误差尽可能小”这一观点出发去寻求逼近曲线, 由这一观点所衍生的方法统称为**曲线拟合方法**. 目前, 人们已推出许多有效的曲线拟合方法, 本节将主要介绍正则化方法和最小二乘法.

设有函数$f(x)$ 及一组数据$\{(x_i, y_i)|\ y_i = f(x_i)\}_{i=0}^n$，今拟在由函数$\{\varphi_i(x)\}_{i=0}^m$所张成的函数类$\Phi \triangleq span\{\varphi_0(x), \varphi_1(x), \cdots, \varphi_m(x)\}$ 中寻找一个逼近$f(x)$的函数$\psi(x)$，使得

$$\sum_{i=0}^{n}[\psi(x_i) - y_i]^2 = \min_{\varrho\in\Phi}\sum_{i=0}^{n}[\varrho(x_i) - y_i]^2, \quad \text{其中} \quad \psi(x) \in \Phi. \tag{4.49}$$

图 4.3 轮船断面的轮廓线

若记

$$\varrho(x)=\sum_{i=0}^{m}a_i\varphi_i(x),\quad \psi(x)=\sum_{i=0}^{m}a_i^*\varphi_i(x),$$

$$a=(a_0,a_1,\cdots,a_m)^T,\quad a^*=(a_0^*,a_1^*,\cdots,a_m^*)^T,$$

$$A=\begin{pmatrix}\varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_m(x_0)\\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_m(x_1)\\ \vdots & \vdots & & \vdots\\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_m(x_n)\end{pmatrix},\quad b=\begin{pmatrix}y_0\\ y_1\\ \vdots\\ y_n\end{pmatrix},$$

则问题(4.49) 等价于: 求一个向量$a^*\in R^{m+1}$, 使得

$$\|Aa^*-b\|_2=\min_{a\in\in R^{m+1}}\|Aa-b\|_2. \tag{4.50}$$

由下述定理, 问题(4.50) 可进一步转化为线性方程组的求解问题.

定理 4.11 $a^*\in R^{m+1}$ 为问题(4.50) 的解当且仅当

$$A^TAa^*=A^Tb. \tag{4.51}$$

证明 若a^* 为方程组(4.51) 的解，$a\in R^{m+1}$ 为任意给定向量，$\Delta a=a-a^*$, 则有

$$\begin{aligned}&\|Aa-b\|_2^2=\|Aa^*-b+A\Delta a\|_2^2\\ =&\|Aa^*-b\|_2^2+\|A\Delta a\|_2^2+2(\Delta a)^TA^T(Aa^*-b)\\ =&\|Aa^*-b\|_2^2+\|A\Delta a\|_2^2\geqslant\|Aa^*-b\|_2^2.\end{aligned}$$

因此，$a^*\in R^{m+1}$ 为问题(4.50) 的解.

反之，若a^* 为问题(4.50) 的解，但其不满足(4.51), 即$z\triangleq A^T(Aa^*-b)\neq 0$, 那么，对于向量$a\triangleq a^*-\varepsilon z$(这里$\varepsilon$ 为任意正数) 有

$$\begin{aligned}&\|Aa-b\|_2^2=\|Aa^*-b-\varepsilon Az\|_2^2\\ =&\|Aa^*-b\|_2^2+\varepsilon^2\|Az\|_2^2-2\varepsilon z^TA^T(Aa^*-b)\\ =&\|Aa^*-b\|_2^2+\varepsilon^2\|Az\|_2^2-2\varepsilon\|z\|_2^2<\|Aa^*-b\|_2^2+\varepsilon^2\|Az\|_2^2.\end{aligned}$$

因此，必存在充分小的正数ε 使得$\|Aa-b\|_2<\|Aa^*-b\|_2$. 这与a^* 为问题(4.50) 的解相矛盾，故必有(4.51) 成立. ■

定理4.11表明: 通过解线性方程组(4.51)可获得问题(4.50) 的解. 故此, 我们获得了一种求解问题(4.50) 的曲线拟合方法，该方法称为**正则化方法**.

例 4.7 已知一组药物试验数据如下：

i	0	1	2	3	4	5	6	7	8	9
x_i	1.0	1.5	2.2	2.4	2.7	3.5	4.0	4.4	4.6	5.0
y_i	6.72	6.92	7.36	7.47	7.56	8.04	8.38	8.64	8.70	8.95

试由此确定该药物的疗效规律$\psi(x)=\alpha\exp(\beta x)$.

解 记$z=\ln\psi(x)$, 则$\ln\psi(x)=\ln\alpha+\beta x$. 从而, 原问题转化为直线$z=a_0+a_1x$的拟合问题, 这里$a_0=\ln\alpha,\ a_1=\beta$. 让$\varphi_0(x)=1,\ \varphi_1(x)=x$, 则由此及已知数据得

$$A=\begin{pmatrix}1 & 1.0\\ 1 & 1.5\\ 1 & 2.2\\ 1 & 2.4\\ 1 & 2.7\\ 1 & 3.5\\ 1 & 4.0\\ 1 & 4.4\\ 1 & 4.6\\ 1 & 5.0\end{pmatrix},\quad b=\begin{pmatrix}\log(6.72)\\ \log(6.92)\\ \log(7.36)\\ \log(7.47)\\ \log(7.56)\\ \log(8.04)\\ \log(8.38)\\ \log(8.64)\\ \log(8.70)\\ \log(8.95)\end{pmatrix}.$$

利用Matlab的矩阵乘除法命令得其拟合直线的系数阵

$$a\triangleq\ [a_0,a_1]^T=(A'*A)\setminus(A'*b)\approx[1.8313,\ 0.0728]^T.$$

因此, 所求拟合曲线为

$$\psi(x)=\exp(a_0+a_1x)\approx 6.242\exp(0.0728x),$$

该曲线图形见图 4.4. ■

图 4.4 拟合曲线$\psi(x)=6.242\exp(0.0728x)$

问题(4.50) 也可转化为求多元函数

$$F(a_0,a_1,\cdots,a_m)=\sum_{i=0}^{n}\left[\sum_{j=0}^{m}a_j\varphi_j(x_i)-y_i\right]^2$$

的极小点$(a_0^*, a_1^*, \cdots, a_m^*)$ 问题. 令

$$\frac{\partial F(a_0, a_1, \cdots, a_m)}{\partial a_k} = 0, \quad k = 0, 1, \cdots, m,$$

即

$$\sum_{i=0}^{n} \left[\sum_{j=0}^{m} a_j \varphi_j(x_i) - y_i \right] \varphi_k(x_i) = 0, \quad k = 0, 1, \cdots, m. \tag{4.52}$$

记

$$(\varphi_j, \varphi_k) = \sum_{i=0}^{n} \varphi_j(x_i)\varphi_k(x_i), \quad d_k = \sum_{i=0}^{n} y_i \varphi_k(x_i),$$

则式(4.52) 可写为

$$\sum_{j=0}^{m} (\varphi_k, \varphi_j) a_j = d_k, \quad k = 0, 1, \cdots, m. \tag{4.53}$$

该方程称为上述最优化问题的**法方程**或**正则方程**，其可写成矩阵形式

$$\begin{pmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_m) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_m) \\ \vdots & \vdots & & \vdots \\ (\varphi_m, \varphi_0) & (\varphi_m, \varphi_1) & \cdots & (\varphi_m, \varphi_m) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_m \end{pmatrix}. \tag{4.54}$$

设$\varphi_0, \varphi_1, \cdots, \varphi_m$ 线性无关，则方程组(4.54) 的系数矩阵行列式不为零，从而该方程组有唯一解$(a_0^*, a_1^*, \cdots, a_m^*)$. 故得最小二乘拟合解$\psi(x) = \sum\limits_{i=0}^{m} a_i^* \varphi_i(x)$.

上述曲线拟合方法称为**最小二乘法**, 类似于正则化方法, 我们可采用Matlab 的矩阵乘除法命令解线性方程组(4.54) 而获得拟合解. 当基函数$\varphi_0(x), \varphi_1(x), \cdots, \varphi_m(x)$ 在多项式函数类

$$\Phi \triangleq span\{1, x, x^2, \cdots, x^m\}$$

中选取时，我们也可使直接使用Matlab 命令$polyfit(x0, y0, m)$ 求得拟合解, 其中$x0, y0$ 为给定的数据组.

例 4.8 已知一组实验数据如下：

i	1	2	3	4	5	6	7	8	9	10	11	12
x_i	-2	-1.63	-1.49	-1.26	-0.89	-0.45	0	0.35	0.76	1.28	1.56	2
y_i	18.2	16.2	15.1	14.1	12.8	11.8	11.2	11.4	12.0	13.1	14.2	16.5

求它的拟合曲线$y = a_0 x^2 + a_1 x + a_2$.

解 记

$$x0 = [-2,\ -1.63,\ -1.49,\ -1.26,\ -0.89,\ -0.45,\ 0,\ 0.35,\ 0.76,\ 1.28,\ 1.56,\ 2]^T,$$

$$y0 = [18.2,\ 16.2,\ 15.1,\ 14.1,\ 12.8,\ 11.8,\ 11.2,\ 11.4,\ 12.0,\ 13.1,\ 14.2,\ 16.5]^T.$$

则利用 Matlab 命令得其拟合曲线的系数阵

$$a \triangleq [a_0, a_1, a_2]^T = polyfit(x0, y0, 2) = [1.5115,\ -0.4278,\ 11.2719]^T.$$

因此, 所求拟合曲线为

$$y = 1.5115x^2 - 0.4278x + 11.2719,$$

其图形见图4.5. ■

图 4.5 拟合曲线$y = 1.5115x^2 - 0.4278x + 11.2719$

习 题 四

4.1 给定函数$f(x) = x\exp(x)[1 + \exp(x)]$ 及插值节点$x_0 = 1.00,\ x_1 = 1.02,\ x_2 = 1.04,\ x_3 = 1.06$, 试构造3次Lagrange 插值多项式计算$f(1.03)$ 的逼近值，并导出其误差估计.

4.2 给定$f(x) = shx$及插值节点$x_0 = 0.40,\ x_1 = 0.55,\ x_2 = 0.70,\ x_3 = 0.85,\ x_4 = 1.00$, 试构造4 次Newton 插值多项式计算$f(0.596)$ 的逼近值, 并指出其绝对误差.

4.3 求不高于4 次的多项式$H(x)$, 使它满足$H(1) = -2,\ H'(1) = 4,\ H(2) = H'(2) = 0,\ H(3) = 2$，并写出其余项表达式.

4.4 证明两点三次Hermite 插值余项是

$$R_3(x) = \frac{1}{4!}f^{(4)}(\xi)(x - x_k)^2(x - x_{k+1})^2,\quad \xi \in (x_k, x_{k+1}).$$

4.5 设四次连续可微函数$f(x)$ 的三次插值多项式$P(x)$ 满足条件

$$P(x_0) = f(x_0),\quad P(x_1) = f(x_1),\quad P(x_2) = f(x_2),\quad P'(x_1) = f'(x_1),$$

试求插值多项式$P(x)$，并证明：在x 与诸x_i 之间存在ξ 使得

$$f(x) = P(x) + \frac{f^{(4)}(\xi)}{4!}(x - x_0)(x - x_1)^2(x - x_2).$$

4.6 构造适合下列数据表的三次样条插值函数$S(x)$

x	-1	0	1	3
y	-1	1	3	31
y'	4			28

4.7 已知一组实验数据如下：

i	0	1	2	3	4	5
x_i	0.1	0.3	0.5	0.6	0.7	0.9
y_i	0.61	0.92	1.12	1.52	1.47	2.04

试用正则化方法确定拟合曲线$\psi(x) = a_0 + a_1 x + a_2 \sin(x) + a_3 \exp(x)$.

4.8 给定下列实验数据

x	19	25	31	38	41
y	19.0	32.3	49.0	73.3	97.8

试用最小二乘法求一个形如$y = a + bx^2$ 的经验公式.

4.9 (实验题)已知微分方程初值问题

$$x' + ax = 0,\ x(0) = b$$

的解$x(t)$ 在其网格点$\{t_i\}$ 处的数值解$\{x_i\}$:

i	1	2	3	4	5
t_i	0.1	0.2	0.3	0.4	0.5
x_i	80.4	53.9	36.1	24.2	16.2

试用最小二乘法确定参数$a,\ b$.

第五章　数值积分

就理论上而言, Newton-Leibniz公式已彻底解决了定积分的计算问题. 但是, 实际计算并非如此，有些积分理论上可证明其原函数存在，但却无法用初等函数明显表出, 如积分

$$\int_a^b \exp(-x^2)dx,\quad \int_a^b \sin(x^2)dx,\quad \int_a^b \frac{1}{(1+k\sin^2 x)\sqrt{1-l^2\sin^2 x}}dx\quad (0<l<1).$$

此外, 有些函数关系是用图表表示的，对这种函数的积分上述公式也无能为力. 鉴此，本章将探讨定积分的数值计算.

§5.1　机械求积公式

定积分求值的困难性源于被积函数的复杂性. 因此，用简单函数逼近复杂被积函数是构造数值积分算法的基本思想. 众所周知，从几何观点来看定积分$\int_a^b f(x)dx$ 即为由曲线$y=f(x)$ ，直线$x=a$、$x=b$ 及x 轴所围平面图形面积的代数和. 因此，若用直线段$y=f[\theta a+(1-\theta)b]$ $(\theta\in[0,1])$ 近似代替曲线段$y=f(x)$ $(a\leqslant x\leqslant b)$ ，则可得**矩形积分公式**

$$\int_a^b f(x)dx\approx(b-a)f[\theta a+(1-\theta)b],\quad \theta\in[0,1]. \tag{5.1}$$

特别，当取$\theta=0,\ \frac{1}{2},\ 1$ 时，我们分别称之为**右矩公式，中矩公式和左矩公式**. 若以过点$A(a,f(a))$、$B(b,f(b))$的直线段

$$y=f(a)+\frac{f(b)-f(a)}{b-a}(x-a),\quad x\in[a,b]$$

近似代替曲线段$y=f(x)$，则得梯形公式

$$\int_a^b f(x)dx\approx\frac{b-a}{2}[f(a)+f(b)]. \tag{5.2}$$

考虑过点$A(a,f(a))$、$C(\frac{b+a}{2},f(\frac{b+a}{2}))$、$B(b,f(b))$ 的抛物线段

$$y=px^2+qx+r,\quad x\in[a,b],$$

其中p,q,r 由方程组

$$\begin{cases} pa^2+qa+r=f(a),\\ p\left(\frac{a+b}{2}\right)^2+q\left(\frac{a+b}{2}\right)+r=f\left(\frac{a+b}{2}\right),\\ pb^2+qb+r=f(b)\end{cases}$$

确定，用该抛物线段近似代替曲线$y=f(x)$ $(a\leqslant x\leqslant b)$ ，则得 **Simpson 公式**

$$\int_a^b f(x)dx\approx\frac{b-a}{6}[f(a)+4f(\frac{a+b}{2})+f(b)]. \tag{5.3}$$

例 5.1　试分别利用中矩公式、梯形公式及 Simpson 公式计算定积分

$$I=\int_0^{1/2}\frac{1}{(x+1)\sqrt{x^2+1}}dx,$$

并比较其计算精度.

解 为比较其数值积分的精度，我们首先计算该定积分的精确值. 令$\frac{1}{x+1}=t$，则有

$$I=\int_{2/3}^{1}\frac{1}{\sqrt{2t^2-2t+1}}dt=\frac{1}{\sqrt{2}}\ln(2t-1+\sqrt{4t^2-4t+2})\Big|_{2/3}^{1}=\frac{1}{\sqrt{2}}\ln\frac{3(1+\sqrt{2})}{1+\sqrt{10}}.$$

若利用中矩公式计算该定积分，则有

$$I_M\triangleq\frac{1/2}{(1/4+1)\sqrt{(1/4)^2+1}}=0.38805700005813,$$

其绝对误差为$e_M\triangleq|I-I_M|\approx 0.00362601834878$. 若利用梯形公式计算该定积分，则有

$$I_T\triangleq\frac{1}{4}\left(1+\frac{1}{(1/2+1)\sqrt{(1/2)^2+1}}\right)\approx 0.39907119849999,$$

其绝对误差为$e_T\triangleq|I-I_T|=0.00738818009308$. 若利用 Simpson 公式计算该定积分，则有

$$I_S\triangleq\frac{1}{12}\left(1+\frac{4}{(1/4+1)\sqrt{(1/4)^2+1}}+\frac{1}{(1/2+1)\sqrt{(1/2)^2+1}}\right)\approx 0.39172839953875,$$

其绝对误差为$e_S\triangleq|I-I_S|\approx 4.538113183999437e-005$. 由上可知 Simpson 公式的计算精度为最佳. ■

公式(5.1)-(5.3) 实质是采用$[a,b]$ 上若干节点x_n 处的函数值$f(x_n)$ 进行适当加权平均获得的，这类公式的一般形式为

$$\int_a^b f(x)dx\approx\sum_{n=0}^{N}A_nf(x_n),\tag{5.4}$$

其中x_n 称为**求积节点**，A_n 称为**求积系数**. 鉴于其系数A_n 仅与节点选择有关而与被积函数$f(x)$ 无关，因此求积公式(5.4) 具有通用性，且称之为**机械求积公式**.

§5.2 代数精度法

Taylor 展开定理表明，一个充分可微函数$f(x)$ 可展开为一个多项式与其余项的和. 因此，若要求积分$\int_a^b f(x)dx$ 的近似计算具有一定精度，则需公式(5.4) 对x^i 自$i=0$ 到足够大的正整数m能精确成立. 为此，我们引入代数精度概念.

定义 5.1 若一个求积公式(5.4) 对$f(x)=x^i$ $(i=0,1,\cdots,m)$ 能精确成立，但对$f(x)=x^{m+1}$ 不精确成立，则称该公式具m 次代数精度.

依据上述定义, 我们可直接验证矩形公式(5.1) 具有 0 次代数精度，梯形公式(5.2) 具 1 次代数精度，而 Simpson 公式(5.3) 具 3 次代数精度. 以代数精度作为标准构造求积公式的方法称为**代数精度法**. 若公式(5.4)对于$f(x)=x^i$ $(i=0,1,\cdots,N)$ 精确成立，则得线性方程组

$$\sum_{n=0}^{N}A_nx_n^i=\frac{b^{i+1}-a^{i+1}}{i+1},\qquad i=0,1,\cdots,N.\tag{5.5}$$

当给定的节点$x_n(n=0,1,\cdots,N)$ 互异时，诸系数A_n 即可由(5.5) 唯一确定.

例 5.2 试确定一个具有3 次代数精度的求积公式

$$\int_0^3 f(x)dx \approx A_0 f(0)+A_1 f(1)+A_2 f(2)+A_3 f(3),$$

并由该公式计算定积分

$$I=\int_0^3 \frac{x\exp(x)}{(x+1)^2}dx$$

及其绝对误差.

解 据(5.5)，要公式具 3 次代数精度，则必有

$$\begin{cases} A_0+A_1+A_2+A_3=3,\\ A_1+2A_2+3A_3=\dfrac{9}{2},\\ A_1+4A_2+9A_3=9,\\ A_1+8A_2+27A_3=\dfrac{81}{4}. \end{cases}$$

解之得

$$A_0=\frac{3}{8},\quad A_1=\frac{9}{8},\quad A_2=\frac{9}{8},\quad A_3=\frac{3}{8}.$$

由此即得求积公式

$$\int_0^3 f(x)dx \approx \frac{3}{8}[f(0)+3f(1)+3f(2)+f(3)],$$

且当将 $f(x)=x^4$ 代入上式时，其不能精确成立，故所得公式具 3 次代数精度. 应用该公式得

$$\tilde{I}\triangleq \frac{9e}{8}\left(\frac{1}{4}+\frac{2e}{9}+\frac{e^2}{16}\right).$$

又其积分的精确值为

$$I=\int_0^3 \frac{x\exp(x)}{(x+1)^2}dx=\frac{\exp(x)}{1+x}\Big|_0^3=\frac{e^3}{4}-1,$$

则$\tilde{I}$的绝对误差为$|I-\tilde{I}|=2.660873101484107e-003.$ ■

§5.3 插值求积法

数值求积公式也可通过插值方法获得. 在积分区间$[a,b]$ 上选取若干节点, 依此构造一个逼近被积函数$f(x)$的插值多项式$P_N(x)$, 进而获得数值求积公式

$$\int_a^b f(x)dx \approx \int_a^b P_N(x)dx.$$

这样获得的求积公式称为**插值型求积公式**, 其方法称为**插值求积法**

对于积分$\int_a^b f(x)dx$，我们考虑Lagrange 插值. 在区间$[a,b]$ 上任取$N+1$ 个互异点$x_0,x_1,\cdots,x_N$，构造$f(x)$ 的带余项的 Lagrange 插值公式

$$f(x)=\sum_{n=0}^{N}\frac{\omega_{N+1}(x)}{(x-x_n)\omega'_{N+1}(x_n)}f(x_n)+R_N(f,x), \tag{5.6}$$

其中

$$\omega_{N+1}(x)=\prod_{i=0}^{N}(x-x_i),\qquad R_N(f,x)=\frac{f^{(N+1)}(\xi)}{(N+1)!}\omega_{N+1}(x),\quad \xi\in(a,b).$$

将(5.6) 代入积分$\int_a^b f(x)dx$得

$$\int_a^b f(x)dx=\sum_{n=0}^{N}A_nf(x_n)+R_N(f),\tag{5.7}$$

其中

$$A_n=\int_a^b\frac{\omega_{N+1}(x)}{(x-x_n)\omega'_{N+1}(x_n)}dx,\quad R_N(f)=\int_a^b R_N(f,x)dx.\tag{5.8}$$

在(5.7) 略去余项$R_N(f)$ 即得插值型求积公式

$$\int_a^b f(x)dx\approx\sum_{n=0}^{N}A_nf(x_n).\tag{5.9}$$

若$\max\limits_{x\in[a,b]}|f^{(N+1)}(x)|=M_{N+1}$，则其余项$R_N(f)$有如下估计式

$$|R_N(f)|\leqslant\frac{M_{N+1}}{(N+1)!}\int_a^b|\omega_{N+1}(x)|dx.\tag{5.10}$$

例 5.3 取节点$x_n=\frac{n}{4}$ $(n=0,1,2,3,4)$, 试利用4 次插值型求积公式计算定积分

$$I=\int_0^1\sin(x^2)dx,$$

并估计其误差.

解 由(5.8) 可计算出求积公式的系数

$$A_0=\frac{7}{90},\ \ A_1=\frac{16}{45},\ \ A_2=\frac{2}{15},\ \ A_3=\frac{16}{45},\ \ A_4=\frac{7}{90}.$$

因此，利用4次插值型求积公式有

$$\int_0^1\sin(x^2)dx\approx\sum_{n=0}^{4}A_n\sin(x_n^2)=0.3102614236535374.$$

又

$$M_5\triangleq\max_{x\in[0,1]}\left|\tfrac{d^5(\sin x^2)}{dx^5}\right|=\max_{x\in[0,1]}\left|32x^5\cos(x^2)+160x^3\sin(x^2)-120x\cos(x^2)\right|$$

$$=\left|32x^5\cos(x^2)+160x^3\sin(x^2)-120x\cos(x^2)\right|_{x=1}\approx 8.7089e+001,$$

则据(5.10)，其误差估计为

$$|R_4(f)|\leqslant\frac{M_5}{5!}\int_0^1\prod_{n=0}^{4}|x-x_n|dx\approx 1.1222e-003.\quad\blacksquare$$

插值求积法与代数精度法具有如下的关系.

定理 5.1 求积公式(5.4)为插值型的充要条件是该公式至少有N 次代数精度.

证明 设公式(5.4) 属于插值型，即为公式(5.9). 因为对$f(x) = x^i$ $(i = 0, 1, \cdots, N)$ 均有$f^{(N+1)}(x) = 0$ ，从而此时$R_N(f) = 0$, 即公式(5.9) 对$f(x) = x^i$ $(i = 0, 1, \cdots, N)$ 均精确成立，故公式(5.9) 至少具N 次代数精度. 另一方面，若公式(5.4) 至少具有N 次代数精度，则其对N 次多项式

$$l_n(x) = \frac{\omega_{N+1}(x)}{(x - x_n)\omega'_{N+1}(x_n)}, \quad n = 0, 1, \cdots, N$$

精确成立，即

$$\int_a^b l_n(x)dx = \sum_{j=0}^{N} A_j l_n(x_j).$$

由$l_n(x_j) = \delta_{nj}$有

$$A_n = \int_a^b l_n(x)dx.$$

故公式(5.9) 成立，即(5.4) 为插值型的. ■

值得注意的是，定理 5.1 只表明$N + 1$ 个节点的插值型公式至少具N 次代数精度，但并不意味着此时公式仅有N 次代数精度. 如 Simpson 公式有 3 个节点，但其具 3 次代数精度.

§5.4 Newton-Cotes 公式及其复合求积法

下面，我们考虑具等距节点的插值型求积公式及其复合求积法. 记$x = a + th$. 当公式(5.9) 取等距节点

$$x_n = a + nh, \quad n = 0, 1, 2, \cdots, N, \quad h = \frac{b - a}{N}$$

时，其系数A_n 由(5.8) 得

$$A_n = \frac{(-1)^{N-n}h}{n!\ (N - n)!} \int_0^N \prod_{i=0, i\neq n}^{N} (t - i)dt, \quad n = 0, 1, \cdots, N.$$

引入 Cotes 系数

$$B_n = \frac{(-1)^{N-n}}{N[n!\ (N - n)!]} \int_0^N \prod_{i=0, i\neq n}^{N} (t - i)dt,$$

则由(5.9) 得**Newton-Cotes 求积公式**

$$\int_a^b f(x)dx \approx (b - a) \sum_{n=0}^{N} B_n f(a + nh). \tag{5.11}$$

在实际应用公式(5.11) 计算积分时，由于系数B_n 仅与n 及节点数N 有关，而与积分限a, b 无关，因此对不同的N 可事先将B_n 算出，且注意诸系数具对称性：

$$B_n = B_{N-n}, \quad n = 0, 1, \cdots, N.$$

理论分析及数值实验表明，当公式(5.11) 的阶数N 较大时，其稳定性会大大地降低，从而产生较大误差. 事实上, Newton-Cotes 公式中有实用价值的往往是一些改进的低阶公式. 为此, 我们首先考察两种低阶公式.

在公式(5.11) 中取$N=1$，则得梯形公式

$$\int_a^b f(x)dx \approx \frac{b-a}{2}[f(a)+f(b)]. \tag{5.12}$$

若$f(x)\in C^{(2)}([a,b])$ ，则由(5.8) 及积分中值定理可推得其余项

$$\begin{aligned} R_1(f) &= \int_a^b \frac{f''(\tilde{\xi})}{2!}(x-a)(x-b)dx \\ =&\frac{f''(\xi)}{2!}\int_a^b (x-a)(x-b)dx = -\frac{(b-a)^3}{12}f''(\xi), \quad \xi,\ \tilde{\xi}\in[a,b]. \end{aligned} \tag{5.13}$$

在公式(5.11) 中取$N=2$，则得 Simpson 公式

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left[f(a)+4f(\frac{a+b}{2})+f(b)\right]. \tag{5.14}$$

为研究(5.14) 的余项，设$f(x)\in C^{(4)}([a,b])$，并构造一个满足条件

$$\begin{cases} H(a)=f(a), \quad H(b)=f(b), \\ H(\dfrac{a+b}{2})=f(\dfrac{a+b}{2}), \quad H'(\dfrac{a+b}{2})=f'(\dfrac{a+b}{2}), \end{cases} \tag{5.15}$$

的三次多项式$H(x)$. 应用习题4.5 的结论得

$$f(x)=H(x)+\frac{f^{(4)}(\tilde{\xi})}{4!}(x-a)(x-\frac{a+b}{2})^2(x-b), \quad \xi\in[a,b]. \tag{5.16}$$

由于 Simpson 公式具有三次代数精度，则根据(5.16) 及积分中值定理得

$$\begin{aligned} \int_a^b f(x)dx &= \int_a^b H(x)dx+\frac{1}{4!}\int_a^b f^{(4)}(\tilde{\xi})(x-a)(x-\frac{a+b}{2})^2(x-b)dx \\ &= \frac{b-a}{6}\left[H(a)+4H(\frac{a+b}{2})+H(b)\right]+\frac{f^{(4)}(\xi)}{4!}\int_a^b (x-a)(x-\frac{a+b}{2})^2(x-b)dx \\ &= \frac{b-a}{6}\left[f(a)+4f(\frac{a+b}{2})+f(b)\right]-\frac{(b-a)^5}{2880}f^{(4)}(\xi), \quad \xi,\ \tilde{\xi}\in[a,b]. \end{aligned}$$

故公式(5.14) 的余项为

$$R_2(f)=-\frac{(b-a)^5}{2880}f^{(4)}(\xi), \quad \xi\in[a,b]. \tag{5.17}$$

上述低阶公式要真正做到实用，其精度仍有待提高. 为此，我们把将分区间$[a,b]$ 等分成N个子区间$[x_n,x_{n+1}]$ $(n=0,1,2,\cdots,N)$，其中

$$x_n=a+nh, \quad h=\frac{b-a}{N},$$

然后在每个子区间上使用低阶公式，再将计算结果累加起来. 所得公式称为**复合求积公式**.

首先，我们构造复合梯形公式. 在每个子区间$[x_n, x_{n+1}]$ 上使用带余项的梯形公式

$$\int_{x_n}^{x_{n+1}} f(x)dx = \frac{h}{2}[f(x_n) + f(x_{n+1})] - \frac{h^3}{12}f^{''}(\eta_n), \quad \eta_n \in [x_n, x_{n+1}],$$

求和得

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{n=1}^{N-1} f(x_n) + f(b)\right] - \frac{h^3}{12}\sum_{n=0}^{N-1} f^{''}(\eta_n).$$

略去其余项即得**复合梯形公式**

$$\int_a^b f(x)dx \approx \frac{h}{2}\left[f(a) + 2\sum_{n=1}^{N-1} f(x_n) + f(b)\right], \tag{5.18}$$

其余项

$$\tilde{R}_T(f) \triangleq -\frac{h^2}{12}\sum_{n=0}^{N-1}[f^{''}(\eta_n)h] \approx -\frac{h^2}{12}\int_a^b f^{''}(x)dx = -\frac{h^2}{12}[f'(b) - f'(a)]. \tag{5.19}$$

记$x_{n+\frac{1}{2}} = \frac{x_n + x_{n+1}}{2}$，则在$[x_n, x_{n+1}]$ 上有带余项的 Simpson 公式

$$\int_{x_n}^{x_{n+1}} f(x)dx = \frac{h}{6}[f(x_n) + 4f(x_{n+\frac{1}{2}}) + f(x_{n+1})] - \frac{h^5}{2880}f^{(4)}(\theta_n), \quad \theta_n \in [x_n, x_{n+1}].$$

求和得

$$\int_a^b f(x)dx = \frac{h}{6}\left[f(a) + 4\sum_{n=0}^{N-1} f(x_{n+\frac{1}{2}}) + 2\sum_{n=1}^{N-1} f(x_n) + f(b)\right] - \frac{h^5}{2880}\sum_{n=0}^{N-1} f^{(4)}(\theta_n).$$

略去其余项即得**复合 Simpson 公式**

$$\int_a^b f(x)dx \approx \frac{h}{6}\left[f(a) + 4\sum_{n=0}^{N-1} f(x_{n+\frac{1}{2}}) + 2\sum_{n=1}^{N-1} f(x_n) + f(b)\right], \tag{5.20}$$

其余项

$$\tilde{R}_S(f) \triangleq -\frac{h^5}{2880}\sum_{n=0}^{N-1} f^{(4)}(\theta_n) \approx -\frac{h^4}{2880}\int_a^b f^{(4)}(x)dx = -\frac{h^4}{2880}[f^{(3)}(b) - f^{(3)}(a)]. \tag{5.21}$$

出于计算机编程方面的考虑，我们记

$$\tilde{x}_{2n} = x_n, \quad \tilde{x}_{2n-1} = x_{n-\frac{1}{2}}, \qquad n = 1, 2, \cdots, N,$$

而将式(5.20) 写成

$$\int_a^b f(x)dx \approx \frac{\tilde{h}}{3}\left\{f(a) - f(b) + 2\sum_{n=1}^{N}[2f(\tilde{x}_{2n-1}) + f(\tilde{x}_{2n})]\right\}, \tag{5.22}$$

其中$\tilde{h} = \frac{b-a}{2N}$, $\quad \tilde{x}_n = a + n\tilde{h}, \qquad n = 1, 2, \cdots, 2N.$

例 5.4 取步长$h=\frac{1}{8}$，分别用公式(5.18) 及(5.22) 计算积分

$$I=\int_0^1 \frac{\sin x}{x}dx.$$

解 记$f(x)=\frac{\sin x}{x}$，并取$f(0)=\lim\limits_{x\to 0}\frac{\sin x}{x}=1$，则由公式(5.18) 得

$$I\approx\frac{1}{16}\left[f(0)+2\sum_{n=1}^{7}f(\frac{n}{8})+f(1)\right]=0.9456908635827014.$$

由公式(5.22) 得

$$I\approx\frac{1}{24}\left\{f(0)-f(1)+2\sum_{n=1}^{4}\left[2f(\frac{2n-1}{8})+f(\frac{n}{4})\right]\right\}=0.9460833108884721.$$

将上述两个近似值与I 的精确值$0.9460831\cdots$ 相比较, 复合梯形公式仅有2 位有效数字，而Simpson公式有6 位有效数字. 由此可见，在计算量基本相同的情况下, 后者精度高于前者. ■

§5.5 变步长求积法

利用复合求积技巧使得梯形公式与Simpson公式的计算精度得以改善，但两者均属于定步长公式，若积分计算要求达到某个精度，则其步长的选择成为一件非常困难的事情. 为此，本节探讨在计算机上自动选择步长的变步长方法及其加速收敛技术.

下面, 我们以变步长梯形求积法为例来说明如何自动选择步长并加速计算速度. 将求积区间$[a,b]$ 等分k 次时，则复合梯形公式的数值积分值为

$$T_k=\frac{b-a}{2^{k+1}}\left[f(a)+2\sum_{n=1}^{2^k-1}f\left(a+\frac{n(b-a)}{2^k}\right)+f(b)\right]. \tag{5.23}$$

由该表达式可得递推关系

$$T_k=\frac{1}{2}T_{k-1}+\frac{b-a}{2^k}\sum_{n=1}^{2^{k-1}}f\left[a+\frac{2n-1}{2^k}(b-a)\right],\quad k=1,2,\cdots, \tag{5.24}$$

此即为**变步长梯形求积公式**. 记$I=\int_a^b f(x)dx,\ h=\frac{b-a}{2^k}$, 则由(5.19) 有

$$I-T_k\approx-\frac{h^2}{12}[f'(b)-f'(a)],\quad I-T_{k+1}\approx-\frac{h^2}{48}[f'(b)-f'(a)]. \tag{5.25}$$

根据(5.25), 其事后误差估计

$$I-T_{k+1}\approx\frac{1}{3}(T_{k+1}-T_k). \tag{5.26}$$

鉴于上式是一个近似估计，因此我们可保守地以$|T_{k+1}-T_k|$ 作为当前步逼近值T_{k+1} 的误差. 若计算精度要求$|I-T_{k+1}|<\varepsilon$，则在实际运行变步长梯形求积公式时，不等式$|T_{k+1}-T_k|<\epsilon$ 可作为其计算终止准则，并以满足该终止准则的逼近值T_{k+1} 作为欲求的积分值. 变步长梯形求积法的计算程序如下：

算法 5.1　变步长梯形求积法

```
function z=vstrapezoid(a,b,tol)
  t0=(b-a)*(f(a)+f(b))/2;
  t1=t0/2+(b-a)*f(a+(b-a)/2)/2; k=1;
while abs(t1-t0)>=tol & k<=1000
    k=k+1;
   for n=1:2^(k-1)
     g(n)=f(a+(b-a)*(2*n-1)/2^k);
   end
    s=sum(g); t0=t1; t1=t0/2+s*(b-a)/2^k;
end
t1
k
```

例 5.5　应用变步长梯形求积法计算定积分

$$I=\int_0^3 \frac{x\exp(x)}{(x+1)^2}dx,$$

并要求其计算精度满足：$|T_k-T_{k-1}|<10^{-10}$.

解　取

$$a=0,\quad b=3,\quad tol=10^{-8},\quad f(x)=\frac{x\exp(x)}{(x+1)^2}.$$

运行算法5.1，经18次迭代后获得满足精度要求的积分逼近值$T_{18}=4.021384230820235$, 其所用计算时间为20.9秒. 该逼近值与精确值$I=\frac{e^3}{4}-1$相比较，误差为

$$|T_{18}-I|\approx 2.331645987396769e-011.\quad \blacksquare$$

变步长求积方法不仅提高了低阶公式的精度，而且能在计算机上自动实现, 但这一切均是以增加计算量为代价的. 为补偿这一缺陷，我们考虑采用Richardson 外推加速该算法收敛速度. 记$T_0^{(k)}$ 为二分k 次积分区间$[a,b]$ 后应用复合梯形公式所得积分逼近值，称之为**梯形值**，$T_m^{(k)}$ 为将梯形值序列$\{T_0^{(k)}\}$ 经m 次外推后所得积分逼近值，即有

$$\begin{cases} T_0^{(k)}=\dfrac{1}{2}T_0^{(k-1)}+\dfrac{b-a}{2^k}\displaystyle\sum_{n=1}^{2^{k-1}} f\left[a+\frac{2n-1}{2^k}(b-a)\right], & k=1,\ 2,\ \cdots \\ T_m^{(l)}=\dfrac{4^mT_{m-1}^{(l+1)}-T_{m-1}^{(l)}}{4^m-1}, & l=0,\ 1,\ \cdots;\quad m=1,\ 2,\ \cdots. \end{cases} \tag{5.27}$$

其计算步骤可按如下箭头所示方向依次进行：

$$
\begin{array}{ccccccccc}
T_0^{(0)} & & & & & & & & \\
\downarrow & \searrow & & & & & & & \\
T_0^{(1)} & \rightarrow & T_1^{(0)} & & & & & & \\
\downarrow & \searrow & & \searrow & & & & & \\
T_0^{(2)} & \rightarrow & T_1^{(1)} & \rightarrow & T_2^{(0)} & & & & \\
\downarrow & \searrow & & \searrow & & \searrow & & & \\
T_0^{(3)} & \rightarrow & T_1^{(2)} & \rightarrow & T_2^{(1)} & \rightarrow & T_3^{(0)} & & \\
\vdots & & & \ddots & & \ddots & & \ddots &
\end{array}
$$

若要求逼近精确积分值I 的计算精度满足$|T_m^{(0)} - I| < \varepsilon$，则可利用终止准则$|T_m^{(0)} - T_{m-1}^{(0)}| < \varepsilon$ 结束计算，并取积分逼近值为$T_m^{(0)}$. 该方法称为 **Romberg 算法** ，其计算程序如下：

算法 5.2 Romberg 算法

```
function z=romberg(a,b,tol)
  h=b-a; t(1,1)=h*(f(a)+f(b))/2; m=1; l=0; err=1;
while err>=tol
      l=l+1; h=h/2;
   for n=1:m
      g(n)=f(a+h*(2*n-1));
   end
  s=sum(g); t(l+1,1)=t(l,1)/2+s*h; m=2*m;
   for k=1:l
     t(l+1,k+1)=(4^k*t(l+1,k)-t(l,k))/(4^k-1);
   end
  err=abs(t(l+1,l+1)-t(l,l));
end
t(l+1,l+1)
```

例 5.6 用Romberg 算法计算积分

$$I = \int_0^3 \frac{x\exp(x)}{(x+1)^2}dx,$$

并要求其计算精度满足：$|T_m^{(0)} - T_{m-1}^{(0)}| < 10^{-10}$.

解 取

$$a = 0, \quad b = 3, \quad tol = 10^{-8}, \quad f(x) = \frac{x\exp(x)}{(x+1)^2}.$$

应用Romberg 算法5.2, 迭代8次后可得满足精度要求的积分逼近值$T_8^{(0)} = 4.021384230796916$, 其所用计算时间几乎为0秒. 该逼近值与精确值$I = \frac{e^3}{4} - 1$相比较，误差为

$$|T_8^{(0)} - I| \approx 2.664535259100376e - 015. \quad \blacksquare$$

该例表明, 在计算速度和精度方面, Romberg 算法(5.27)均优于变步长梯形求积法(5.24).

§5.6 Gauss 求积公式

插值型求积公式的精度通常与节点个数有关，要提高其精度必然以增加节点个数为代价. 但是，节点的无限增加将导致其稳定性能减弱，从而使得积分和$\sum\limits_{n=0}^{N} A_n f(x_n)$ 收敛缓慢，甚至不收敛于积分值$\int_a^b f(x)dx$. 为在一定程度上克服这些缺陷，本节引入Gauss型求积公式.

定义 5.2 具有$2N+1$ 次代数精度的插值型求积公式(5.9) 称为Gauss 型求积公式，其节点$\{x_i\}_{i=0}^N$ 称为Gauss 点.

从定义5.2 可知，Gauss 型求积公式比通常同阶插值型公式的精度高. 不失一般性，我们假设公式(5.9) 的积分限为$a=-1,\ b=1$，而对于更一般情形则可作变换

$$x=\frac{2}{b-a}\left(t-\frac{a+b}{2}\right), \tag{5.28}$$

其使得积分

$$\int_a^b f(t)dt=\frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}x+\frac{a+b}{2}\right)dx. \tag{5.29}$$

定理 5.2 $\{x_i\}_{i=0}^N$为Gauss 节点的充要条件是$N+1$ 次多项式$\omega_{N+1}(x)=\prod\limits_{n=0}^{N}(x-x_n)$ 与一切次数小于或等于N 的多项式$Q(x)$ 均正交，即

$$\int_{-1}^{1}\omega_{N+1}(x)Q(x)dx=0. \tag{5.30}$$

证明 设$\{x_i\}_{i=0}^N$是Gauss 点，则公式(5.9) 对任意次数不超过$2N+1$ 的多项式均精确成立. 而多项式$Q(x)\omega_{N+1}(x)$ 的次数至多为$2N+1$，则

$$\int_{-1}^{1}Q(x)\omega_{N+1}(x)dx=\sum_{n=0}^{N}A_nQ(x_n)\omega_{N+1}(x_n)=0.$$

另一方面，设$f(x)$ 是任意次数至多为$2N+1$ 的多项式，则由代数学理论，存在次数至多为N 的多项式$Q(x)$ ，$r(x)$ 使得

$$f(x)=Q(x)\omega_{N+1}(x)+r(x). \tag{5.31}$$

积分得

$$\int_{-1}^{1}f(x)dx=\int_{-1}^{1}Q(x)\omega_{N+1}(x)dx+\int_{-1}^{1}r(x)dx=\sum_{n=0}^{N}A_nr(x_n).$$

进一步由(5.31) 得$f(x_n)=r(x_n)$ $(n=0,1,2,\cdots,N)$ ，因此

$$\int_{-1}^{1}f(x)dx=\sum_{n=0}^{N}A_nf(x_n). \tag{5.32}$$

此外由

$$\sum_{n=0}^{N}A_n\omega_{N+1}^2(x_n)=0,\qquad \int_{-1}^{1}\omega_{N+1}^2(x)dx>0$$

知公式(5.32) 对$2N+2$ 次多项式不精确成立，故(5.32) 为Gauss 型求积公式，即$\{x_i\}_{i=0}^N$ 为Gauss 点. ■

定理5.2表明，若能找到满足(5.30) 的$N+1$ 次多项式$\omega_{N+1}(x)$，则公式的Gauss 点就确定了，从而确定了一个Gauss 型求积公式. 为此，我们引入Legendre 多项式. 一个仅以区间$[-1,1]$ 上的Gauss 点$\{x_i\}_{i=0}^N$为零点的$N+1$ 次多项式称为**Legendre 多项式**.

定理 5.3 首项系数等于1 的Legendre 多项式可唯一地表示为

$$\omega_{N+1}(x)=\frac{(N+1)!}{(2N+2)!}\frac{d^{N+1}[(x^2-1)^{N+1}]}{dx^{N+1}},\quad N=0,1,\cdots. \tag{5.33}$$

证明 考虑$2N+2$ 次多项式

$$u(x)=\underbrace{\int_{-1}^{x}\int_{-1}^{x}\cdots\int_{-1}^{x}}_{N+1}\omega_{N+1}(x)dxdx\cdots dx.$$

该多项式满足

$$u^{(N+1)}(x)=\omega_{N+1}(x),\quad u^{(i)}(-1)=0,\quad i=0,1,2,\cdots,N. \tag{5.34}$$

设$v(x)$ 为任意N 次多项式，则由(5.34) 得

$$\begin{aligned}&\int_{-1}^{1}v(x)\omega_{N+1}(x)dx=\int_{-1}^{1}v(x)u^{(N+1)}(x)dx=[v(x)u^{(N)}(x)]_{-1}^{1}-\int_{-1}^{1}u^{(N)}(x)v'(x)dx\\ =&v(1)u^{(N)}(1)-\left\{[v'(x)u^{(N-1)}(x)]_{-1}^{1}-\int_{-1}^{1}u^{(N-1)}(x)v''(x)dx\right\}\\ =&\cdots\cdots\cdots\cdots\quad\text{(反复逐次分部积分)}\\ =&\sum_{i=0}^{N}(-1)^iv^{(i)}(1)u^{(N-i)}(1)+\int_{-1}^{1}u(x)v^{(N+1)}(x)dx=\sum_{i=0}^{N}(-1)^iv^{(i)}(1)u^{(N-i)}(1).\end{aligned}$$

而据定理5.2有$\int_{-1}^{1}v(x)\omega_{N+1}(x)dx=0$. 从而$\sum\limits_{i=0}^{N}(-1)^iv^{(i)}(1)u^{(N-i)}(1)=0$. 由此及$v(x)$ 的任意性得

$$u^{(N-i)}(1)=0,\quad i=0,1,2,\cdots,N. \tag{5.35}$$

(5.34) 和(5.35) 表明$x=\pm1$ 均为$u(x)$ 的$N+1$ 重零点，故

$$u(x)=c(x^2-1)^{N+1},\quad c\text{为待定常数}.$$

而$\omega_{N+1}(x)$ 的首项系数等于1，则由(5.34) 的第一式有$c=\frac{(N+1)!}{(2N+2)!}$，且因此(5.33)成立. ■

例 5.7 试构造Gauss 型求积公式

$$\int_{-1}^{1}f(x)dx\approx A_0f(x_0)+A_1f(x_1)+A_2f(x_2),$$

并由此计算积分$\int_0^1\frac{\sqrt{t}}{(1+t)^2}dt$.

解 通过求三次 Legendre 多项式$p_3(x)=x^3-\frac{3}{5}x$ 的零点获 Gauss 点

$$x_0=-\sqrt{\frac{3}{5}},\quad x_1=0,\quad x_2=\sqrt{\frac{3}{5}}.$$

由(5.8)的第一式可得其Gauss 型求积公式的系数

$$A_0=\frac{5}{9},\quad A_1=\frac{8}{9},\quad A_2=\frac{5}{9}.$$

故所求公式为

$$\int_{-1}^{1}f(x)dx\approx\frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right)+\frac{8}{9}f(0)+\frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right). \tag{5.36}$$

据(5.28) 作变换$x=2(t-\frac{1}{2})$，则由公式(5.36)得

$$\begin{aligned}&\int_0^1\frac{\sqrt{t}}{(1+t)^2}dt=\sqrt{2}\int_{-1}^{1}\frac{\sqrt{x+1}}{(x+3)^2}dx\\ \approx&\sqrt{2}\left\{\frac{5}{9}\frac{\sqrt{-\sqrt{\frac{3}{5}}+1}}{\left(-\sqrt{\frac{3}{5}}+3\right)^2}+\frac{8}{9}\frac{1}{3^2}+\frac{5}{9}\frac{\sqrt{\sqrt{\frac{3}{5}}+1}}{\left(\sqrt{\frac{3}{5}}+3\right)^2}\right\}\approx 2.8845e-001.\quad\blacksquare\end{aligned}$$

Gauss 求积公式的系数具有下述特征.

定理 5.4 Gauss 求积公式的全体系数$A_n>0$，且

$$A_n=\int_{-1}^{1}\left[\frac{\omega_{N+1}(x)}{(x-x_n)\omega'_{N+1}(x_n)}\right]^2dx,\quad n=0,1,2,\cdots,N.$$

证明 由于Gauss 求积公式具$2N+1$ 次代数精度，且$2N$ 次多项式

$$l_n(x)=\left[\frac{\omega_{N+1}(x)}{(x-x_n)\omega'_{N+1}(x_n)}\right]^2,\quad n=0,1,2,\cdots,N$$

满足

$$l_n(x_j)=\begin{cases}1, & j=n\\ 0, & j\neq n\end{cases}$$

则

$$\int_{-1}^{1}\left[\frac{\omega_{N+1}(x)}{(x-x_n)\omega'_{N+1}(x_n)}\right]^2dx=\sum_{j=0}^{N}A_jl_n(x_j)=A_n,\quad n=0,1,2,\cdots,N.$$

由此即知诸系数$A_n>0$. ■

对于一般插值型求积公式(5.9)，若计算每个函数$f(x_n)$ 时产生舍入误差ε_n，则其整个积分计算中产生总的舍入误差为$\varepsilon=\sum\limits_{n=0}^{N}A_n\varepsilon_n$, 由此有误差估计

$$|\varepsilon|\leqslant\left(\sum_{n=0}^{N}|A_n|\right)\max_{0\leqslant n\leqslant N}|\varepsilon_n|. \tag{5.37}$$

对某些插值型求积公式，$\sum_{n=0}^{N}|A_n|$ 随着N的增大而增大. 此即说明N 增大到一定程度时，这些求积公式将失去其实用价值. 而对于Gauss 求积公式，当取$f(x)=1$ 时, $\sum_{n=0}^{N}A_n=\int_{-1}^{1}1dx=2$. 另一方面根据定理5.4 诸$A_n>0$. 因此, 由(5.37) 有$|\varepsilon|\leqslant 2\max\limits_{0\leqslant n\leqslant N}|\varepsilon_n|$. 由此可知Gauss公式的误差是可控的，即具较好的稳定性. 此外，我们可以证明：若$f(x)\in C([-1,1])$，则Gauss 求积公式必收敛，且有

$$\lim_{N\to\infty}\sum_{n=0}^{N}A_nf(x_n)=\int_{-1}^{1}f(x)dx.$$

习 题 五

5.1 试分别利用中矩公式、梯形公式及Simpson 公式计算定积分

$$I=\int_0^{\frac{1}{2}}\exp(3x)\cos 2xdx,$$

并比较其计算精度.

5.2 试确定下面求积公式

$$\int_{-1}^{1}f(x)dx\approx a[f(x_0)+f(x_1)+f(x_2)],$$

使其具三次代数精度，并由该公式计算定积分

$$I=\int_{-1}^{1}\frac{x\sin x}{\sqrt{1+x^2}}dx.$$

5.3 在区间$[a,b]$ 上导出具五个节点的Newton-Cotes 公式，并指出其余项及代数精度.

5.4 取步长$h=\frac{1}{6}$, 分别用复合梯形公式及复合Simpson 公式计算

$$\int_1^2\frac{x}{\ln(x+1)}dx.$$

5.5 试构造两点Gauss 公式

$$\int_{-1}^{1}f(x)dx\approx A_0f(x_0)+A_1f(x_1),$$

并由此计算积分

$$\int_0^1\sqrt{1+2x}dx.$$

5.6 证明: 若求积公式

$$\int_{-1}^{1}f(x)dx\approx\sum_{n=0}^{N}A_nf(x_n)$$

为Gauss 型的, 且被积函数$f(x)\in C([-1,1])$，则有

$$\lim_{N\to\infty}\sum_{n=0}^{N}A_nf(x_n)=\int_{-1}^{1}f(x)dx.$$

5.7 (实验题) 分别用变步长梯形求积公式和Romberg 算法计算椭园积分

$$\int_0^{\pi} \frac{\sqrt{2}}{(1+\sin^2 x)\sqrt{2-\sin^2 x}}dx,$$

要求其逼近值$T_k, T_k^{(0)}$的计算精度分别满足：$|T_k - T_{k-1}| < 10^{-12}$和$|T_k^{(0)} - T_{k-1}^{(0)}| < 10^{-12}$.

第六章　常微分方程初值问题的数值解法

考虑常微分方程d 维初值问题

$$\frac{dy}{dt} = f(t, y(t)), \quad t \in [a, b]; \quad y(a) = y_0. \tag{6.1}$$

若存在某个向量范数$\|\cdot\|$，使得右函数$f(t, y)$ 满足经典 Lipschitz 条件

$$\|f(t, y_1) - f(t, y_2)\| \leqslant L\|y_1 - y_2\|, \qquad t \in [a, b], \quad y_1, y_2 \in R^d, \tag{6.2}$$

则初值问题(6.1) 在$[a, b]$ 上存在唯一的连续解$y(t)$. 理论上虽已给出了其解的存在性唯一性条件, 但实际精确求解此类问题时仍非常困难. 为此, 本章将介绍一些数值解法. 以下, 我们恒记

$$t_n = a + nh, \quad h = \frac{b-a}{N}, \quad y_n \approx y(t_n), \quad f_n \triangleq f(t_n, y_n) \approx y'(t_n), \qquad n = 0, 1, 2, \cdots, N.$$

§6.1　$\theta-$方法

$\theta-$方法是常微分方程初值问题数值方法中最基本的数值方法，其分为线性$\theta-$方法和单支$\theta-$方法. 下面，我们分别通过差商逼近法和数值积分法获得这二类方法.

当h 充分小时, 若引入参数$\theta \in [0, 1]$，则由一阶向前、向后差商逼近公式：

$$y'(t_n) \approx \frac{y(t_{n+1}) - y(t_n)}{h}, \quad y'(t_{n+1}) \approx \frac{y(t_{n+1}) - y(t_n)}{h} \tag{6.3}$$

得

$$y(t_{n+1}) \approx y(t_n) + h[\theta y'(t_n) + (1-\theta)y'(t_{n+1})]. \tag{6.4}$$

由此得求解系统(6.1) 的**线性$\theta-$ 方法**

$$y_{n+1} = y_n + h[\theta f_n + (1-\theta)f_{n+1}]. \tag{6.5}$$

方法(6.5) 在求当前值y_{n+1} 时，仅需前一步的信息，因此这类方法属于单步法. 特别，若取$\theta = 1$，则得**显式 Euler 法**

$$y_{n+1} = y_n + hf_n; \tag{6.6}$$

若取$\theta = 0$, 则得**隐式 Euler 法**

$$y_{n+1} = y_n + hf_{n+1}; \tag{6.7}$$

若取$\theta = \frac{1}{2}$, 则得**梯形法**

$$y_{n+1} = y_n + \frac{h}{2}(f_n + f_{n+1}). \tag{6.8}$$

方法(6.6) 是一个显式方法，即其每个计算步: $y_n \to y_{n+1}$ 只需直接递推就可获得当前数值解y_{n+1}. 但方法(6.7) 及(6.8) 是隐式方法，其均含有非线性项f_{n+1}，因此它们的每个计算步: $y_n \to y_{n+1}$ 一般均需解一个关于y_{n+1} 的非线性方程才可获得当前数值解y_{n+1}.

例 6.1 试组合显式Euler 法和梯形法构造出一个改进的计算格式，并取步长$h = 0.1$分别应用显式Euler 法及所构造的新格式计算初值问题

$$y'(t) = \frac{2}{t+1}y(t) + (t+1)^2\exp(t), \quad t \in [0,1]; \quad y(0) = 1. \tag{6.9}$$

解 组合显式Euler 法及梯形法可得如下改进的Euler 法

$$y_{n+1} = y_n + \frac{h}{2}[f(t_n, y_n) + f(t_{n+1}, y_n + hf_n)]. \tag{6.10}$$

取步长$h = 0.1$，并分别应用方法(6.6),(6.10) 于初值问题(6.9) 可得如下数值结果

t_n	数值解		误 差	
	Euler 法	改进的 Euler 法	Euler 法	改进的 Euler 法
0.1	1.3000	1.3350	3.7257e-002	$2.2122e-003$
0.2	1.6701	1.7538	8.8731e-002	$4.9783e-003$
0.3	2.1243	2.2729	1.5694e-001	$8.3367e-003$
0.4	2.6793	2.9117	2.4471e-001	$1.2326e-002$
0.5	3.3544	3.6926	3.5521e-001	$1.6984e-002$
0.6	4.1726	4.6423	4.9199e-001	$2.2350e-002$
0.7	5.1607	5.7913	6.5907e-001	$2.8462e-002$
0.8	6.3498	7.1754	8.6097e-001	$3.5360e-002$
0.9	7.7764	8.8361	1.1028e+000	$4.3080e-002$
1.0	9.4829	10.821	1.3903e+000	$5.1663e-002$

其中初值问题(6.9)的精确解$y(t) = (t+1)^2\exp(t)$，网格点$t_n = nh$ 处的误差为$|y(t_n) - y_n|$. 由上表可见，改进的Euler 法的计算精度要优于显式Euler 法. ■

利用数值积分法可得另一类$\theta-$方法. 将问题(6.1)中的微分方程二端自t_n到t_{n+1}积分得

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f(t, y(t))dt. \tag{6.11}$$

应用矩形数值积分公式(5.1) 于上式右端即得**单支$\theta-$ 方法**

$$y_{n+1} = y_n + hf[\theta t_n + (1-\theta)t_{n+1}, \theta y_n + (1-\theta)y_{n+1}], \quad \theta \in [0,1]. \tag{6.12}$$

特别当$\theta = \frac{1}{2}$时, 称之为**隐式中点公式**.

§6.2 线性多步法

在常微分方程初值问题的数值方法中, 除先前介绍的单步方法外, 也有多步方法, 其在计算当前步数值解时需要前多步的信息. 如同单支步方法的构造一样, 多步方法也可由差商逼近法或数值积分法推得. 然而, 这二种推导方法存在着一定的局限性, 其往往只能得到某些类型的多步方法. 一个更为广泛的推导方法是Taylor 展开法.

定义算子

$$L[y(t),h]=\sum_{i=0}^{k}[\alpha_i y(t+ih)-h\beta_i y'(t+ih)], \tag{6.13}$$

其中α_i,β_i 为待定系数. 若$y(t)$ 有$p+2$ 阶连续导数，则上式右端诸项可按Taylor 展式展开为

$$L[y(t),h]=\sum_{j=0}^{p}C_j h^j y^{(j)}(t)+C_{p+1}h^{p+1}y^{(p+1)}(t)+\mathcal{O}(h^{p+2}), \tag{6.14}$$

其中

$$C_0=\sum_{i=0}^{k}\alpha_i,\quad C_1=\sum_{i=1}^{k}(i\alpha_i-\beta_i)-\beta_0,\quad C_j=\frac{1}{j!}\sum_{i=1}^{k}i^{j-1}(i\alpha_i-j\beta_i),\ j=2,3,\cdots,p+1. \tag{6.15}$$

置

$$C_0=C_1=\cdots=C_p=0,\quad C_{p+1}\neq 0 \tag{6.16}$$

则得诸参数α_i，β_i，其使得算子

$$L[y(t),h]=C_{p+1}h^{p+1}y^{(p+1)}(t)+\mathcal{O}(h^{p+2}).$$

由此及(6.14) 得

$$\sum_{i=0}^{k}[\alpha_i y(t+ih)-h\beta_i y'(t+ih)]=C_{p+1}h^{p+1}y^{(p+1)}(t)+\mathcal{O}(h^{p+2}). \tag{6.17}$$

在上式中取

$$t=t_n,\quad y_{n+i}\approx y(t_{n+i}),\quad f_{n+i}\approx y'(t_{n+i}),$$

并去掉余项，则得**线性k 步方法**

$$\sum_{i=0}^{k}\alpha_i y_{n+i}=h\sum_{i=0}^{k}\beta_i f_{n+i},\quad \text{其中}\ \ \alpha_k\neq 0,\ \ \alpha_0^2+\beta_0^2\neq 0. \tag{6.18}$$

若(6.16) 成立，则称方法(6.18) 为p **阶相容的**(或简称为p **阶的**)，此时称式(6.17) 的右端为该方法的**局部截断误差**, 而称$C_{p+1}h^{p+1}y^{(p+1)}(t_n)$ 称为该方法的**局部截断误差主项**. 特别，若$\beta_k\neq 0$，则称之为**隐式的**，否则称之为**显式的**. 由上述分析过程，我们有下述结论.

定理 6.1 线性k 步法(6.18) 为p 阶相容的充要条件是(6.16) 成立.

在(6.16) 中取$k=2$，$p=3$，$\alpha_2=1$ 得

$$\begin{cases}\alpha_0+\alpha_1+1=0,\\ \alpha_1+2-(\beta_0+\beta_1+\beta_2)=0,\\ \dfrac{1}{2!}(\alpha_1+4)-(\beta_1+2\beta_2)=0,\\ \dfrac{1}{3!}(\alpha_1+8)-\dfrac{1}{2!}(\beta_1+4\beta_2)=0,\end{cases}$$

解之得

$$\alpha_1 = -1 - \alpha_0, \quad \beta_0 = -\frac{1}{12}(1 + 5\alpha_0), \quad \beta_1 = \frac{2}{3}(1 - \alpha_0), \quad \beta_2 = \frac{1}{12}(5 + \alpha_0).$$

从而得二步方法

$$y_{n+2} - (1 + \alpha_0)y_{n+1} + \alpha_0 y_n = \frac{h}{12}[(5 + \alpha_0)f_{n+2} + 8(1 - \alpha_0)f_{n+1} - (1 + 5\alpha_0)f_n], \tag{6.19}$$

且有

$$C_4 = -\frac{1}{24}(1 + \alpha_0), \qquad C_5 = -\frac{1}{360}(17 + 13\alpha_0).$$

因此, 当$\alpha_0 \neq -1$ 时，$C_4 \neq 0$，此时方法为三阶的. 特别，若取$\alpha_0 = -5$，则得二步显式方法

$$y_{n+2} + 4y_{n+1} - 5y_n = 2h(2f_{n+1} + f_n); \tag{6.20}$$

若取$\alpha_0 = 0$，则得三阶Adams-Moulton 方法

$$y_{n+2} - y_{n+1} = \frac{h}{12}(5f_{n+2} + 8f_{n+1} - f_n); \tag{6.21}$$

若取$\alpha_0 = -1$，则$C_4 = 0$，但$C_5 = -\frac{1}{90} \neq 0$，此时方法(6.19)成为四阶Milne 方法

$$y_{n+2} - y_n = \frac{h}{3}(f_{n+2} + 4f_{n+1} + f_n). \tag{6.22}$$

类似地，若在(6.16) 中取$k = p = 3,\ \alpha_3 = -\alpha_2 = 1,\ \alpha_1 = \alpha_0$, 则得三步 Adams 方法

$$y_{n+3} = y_{n+2} + \frac{h}{12}[(5 - 12\beta_0)f_{n+3} + (8 + 36\beta_0)f_{n+2} - (1 + 36\beta_0)f_{n+1} + 12\beta_0 f_n], \tag{6.23}$$

且有

$$C_4 = \beta_0 - \frac{1}{24}, \qquad C_5 = -\frac{4}{45} + \frac{3}{2}\beta_0.$$

因此，当$\beta_0 \neq \frac{1}{24}$ 时，方法(6.23) 为三阶的；当$\beta_0 = \frac{1}{24}$ 时，$C_4 = 0$，但$C_5 \neq 0$，此时该方法即为四阶Adams 方法

$$y_{n+3} = y_{n+2} + \frac{h}{24}[9f_{n+3} + 19f_{n+2} - 5f_{n+1} + f_n]. \tag{6.24}$$

§6.3 一般Runge-Kutta 方法

一个k 步方法当用于计算初值问题(6.1) 时, 除需问题本身的初值y_0 外, 还需要$k - 1$ 个额外的计算启动值$y_1, y_1, \ldots, y_{k-1}$. 而单步法则可自起始计算. 但是, 如果单步方法的每步计算: $y_n \to y_{n+1}$ 仅由前一步的逼近值y_n直接导出, 则这样的单步方法往往计算精度不高. 如: 线性$\theta-$ 方法与单支$\theta-$ 方法的局部截断误差均为$(\theta - \frac{1}{2})h^2 y''(t_n) + \mathcal{O}(h^3)$, 因此当$\theta \neq \frac{1}{2}$ 时, 方法的相容阶仅等于1; 当$\theta = \frac{1}{2}$时, 方法的相容阶达到这类方法的最高阶2. 为提高单步法的精度，Runge与Kutta 分别提出了在由y_n 到y_{n+1}的计算过程中增加若干中间逼近值的数值方案, 这就形成了我们本节将要介绍的Runge-Kutta 方法. 作为特例, 单支$\theta-$方法与线性$\theta-$方法可分别写成如下一、二级Runge-Kutta 方法形式 :

$$\begin{cases} Y_{1,\,n} = y_n + h(1 - \theta)f[t_n + (1 - \theta)h, Y_{1,\,n}], \\ y_{n+1} = y_n + hf[t_n + (1 - \theta)h, Y_{1,\,n}]; \end{cases} \tag{6.25}$$

$$\begin{cases} Y_{1,\,n} = y_n, \\ Y_{2,\,n} = y_n + h[\theta f(t_n, Y_{1,\,n}) + (1-\theta) f(t_n + h, Y_{2,\,n})], \\ y_{n+1} = y_n + h[\theta f(t_n, Y_{1,\,n}) + (1-\theta) f(t_n + h, Y_{2,\,n})]. \end{cases} \tag{6.26}$$

Runge-Kutta 方法的一般形式为

$$\begin{cases} Y_{i,n} = y_n + h\sum\limits_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{j,n}), \qquad i = 1, 2, \cdots, s, \\ y_{n+1} = y_n + h\sum\limits_{j=1}^{s} b_j f(t_n + c_j h, Y_{j,n}), \qquad n \geqslant 0, \end{cases} \tag{6.27}$$

其中诸系数a_{ij}，b_j 及横标c_j 为实数，$h > 0$ 为步长，$t_n = a + nh$，$Y_{i,n} \approx y(t_n + c_i h)$，$y_n \approx y(t_n)$. 若$i \leqslant j$ 时均有$a_{ij} = 0$，则称之为**显式方法**，否则称之为是**隐式方法**.

为简化Runge-Kutta 方法的书写，我们经常采用所谓Butcher 表

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \tag{6.28}$$

来表征方法(6.27)，其中$A = (a_{ij}) \in R^{s\times s}, c = (c_1, c_2, \cdots, c_s)^T, b = (b_1, b_2, \cdots, b_s)^T \in R^s$. 此外, Runge-Kutta 方法也可通过引入矩阵的Kronecker积来表示, 其将应用于实际计算. 为此, 在给出Runge-Kutta方法的Kronecker积表示式之前, 我们需要简单介绍一下矩阵的Kronecker积定义及其基本性质. 设m_1, m_2, n_1, n_2为给定的正整数, 矩阵$P = (p_{ij}) \in R^{m_1\times m_2}, Q = (q_{ij}) \in R^{n_1\times n_2}$, 则称下列分块矩阵

$$\begin{bmatrix} p_{11}Q & p_{12}Q & \cdots & p_{1m}Q \\ p_{21}Q & p_{22}Q & \cdots & p_{2m}Q \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1}Q & p_{m2}Q & \cdots & p_{mm}Q \end{bmatrix}$$

为**矩阵P 与Q 的Kronecker 积**，记为$P \otimes Q$. 由Kronecker 积的定义可直接推得其如下性质：

$$(\lambda P) \otimes (\mu Q) = \lambda\mu(P \otimes Q)\ (\lambda, \mu \in R); \quad (P \otimes Q) \otimes S = P \otimes (Q \otimes S);$$

$$(P + Q) \otimes S = P \otimes S + Q \otimes S; \quad P \otimes (Q + S) = P \otimes Q + P \otimes S; \quad (P \otimes Q)^T = P^T \otimes Q^T;$$

$$(P \otimes Q)(S \otimes D) = (PS) \otimes (QD); \quad (P \otimes Q)^{-1} = P^{-1} \otimes Q^{-1}, \text{ 这里矩阵}P, Q\text{均为可逆方阵.}$$

Kronecker积除具有上述基本性质外，其它性质可进一步参见文献[4, 7].

借助于上述Kronecker积定义及下列记号：

$$e = [\underbrace{1, 1, \ldots, 1}_{s}]^T, \quad Y_n = [{Y_{1,n}}^T, {Y_{2,n}}^T, \ldots, {Y_{s,n}}^T]^T,$$

$$F(t_n, Y_n) = [f(t_n + c_1 h, Y_{1,n})^T, f(t_n + c_2 h, Y_{2,n})^T, \ldots, f(t_n + c_s h, Y_{s,n})^T]^T,$$

我们可将Runge-Kutta 方法(6.27)写成如下紧凑形式

$$\begin{cases} Y_n = (e \otimes I_d) y_n + h(A \otimes I_d) F(t_n, Y_n), \\ y_{n+1} = y_n + h(b^T \otimes I_d) F(t_n, Y_n), \quad n \geqslant 0, \end{cases} \tag{6.29}$$

在上面, 我们仅仅给出了Runge-Kutta 方法的一般形式, 其具体方法的构造将基于相容阶的概念. 一个Runge-Kutta 方法(6.27) 称为是 p **阶相容的**(或简称为p 阶的)，若其**局部离散误差**

$$d_{n+1} := y(t_{n+1}) - \tilde{y}_{n+1} = \mathcal{O}(h^{p+1}), \qquad h \to 0, \tag{6.30}$$

其中$\tilde{y}_{n+1}$ 为由精确值$y(t_n)$ 出发计算一步所得的数值解，即

$$\begin{cases} y_{i,n} = y(t_n) + h\sum\limits_{j=1}^{s} a_{ij} f(t_n + c_j h, y_{j,n}), \qquad i = 1, 2, \cdots, s, \\ \tilde{y}_{n+1} = y(t_n) + h\sum\limits_{j=1}^{s} b_j f(t_n + c_j h, y_{j,n}) \qquad n \geqslant 0. \end{cases} \tag{6.31}$$

基于p 阶相容概念, 以下我们将分节探讨显、隐式Runge-Kutta方法的构造.

§6.4 显式Runge-Kutta 方法

显式Runge-Kutta方法的s个中间逼近值$\{Y_{i,n}\}_{i=1}^{s}$可直接递推获得, 因此其拥有计算简单的特点. 但是, 该类方法的稳定性一般不如同级同阶的隐式Runge-Kutta方法. 显式Runge-Kutta方法可依据相容阶概念及Taylor 展开法构造. 下面, 我们以三级三阶显式Runge-Kutta 法为例来说明其导出过程. 设$y(t)$ 为系统(6.1) 的充分可微解，则由Taylor 展式有

$$\begin{aligned} y(t_n + h) &= y(t_n) + \sum_{i=1}^{3} \frac{1}{i!} h^i y^{(i)}(t_n) + \mathcal{O}(h^4) \\ &= y(t_n) + h\hat{f}_n + \frac{1}{2}h^2 F_n + \frac{1}{6}h^3(F_n \hat{f}'_n + G_n) + \mathcal{O}(h^4), \end{aligned} \tag{6.32}$$

其中

$$\hat{f}_n = f(t_n, y(t_n)), \quad \hat{f}'_n = \frac{\partial f(t_n, y(t_n))}{\partial y}, \quad F_n = \frac{\partial f(t_n, y(t_n))}{\partial t} + \hat{f}_n \cdot \hat{f}'_n,$$

$$G_n = \frac{\partial^2 f(t_n, y(t_n))}{\partial t^2} + 2\hat{f}_n \frac{\partial^2 f(t_n, y(t_n))}{\partial t \partial y} + \hat{f}_n^2 \frac{\partial^2 f(t_n, y(t_n))}{\partial y^2}.$$

以下恒设三级显式 Runge-Kutta 法满足

$$c_1 = 0, \qquad c_i = \sum_{j=1}^{i-1} a_{ij}, \quad i = 2, 3. \tag{6.33}$$

若方法自精确值$y(t_n)$ 出发计算一步，则有

$$\begin{cases} f_{1,n} = f(t_n, y(t_n)), \\ f_{2,n} = f(t_n + c_2 h, y(t_n) + h a_{21} f_{1,n}), \\ f_{3,n} = f(t_n + c_3 h, y(t_n) + h a_{31} f_{1,n} + h a_{32} f_{2,n}), \\ \tilde{y}_{n+1} = y(t_n) + h\sum\limits_{j=1}^{3} b_j f_{j,n}. \end{cases} \tag{6.34}$$

将$f_{2,n}$, $f_{3,n}$ 在$(t_n, y(t_n))$ 处展开，并注意到条件(6.33) 得

$$\begin{cases} f_{2,n} = \hat{f}_n + hc_2F_n + \dfrac{1}{2}h^2c_2^2G_n + \mathcal{O}(h^3), \\ f_{3,n} = \hat{f}_n + hc_3F_n + h^2(c_2a_{32}F_n\hat{f}'_n + \dfrac{1}{2}c_3^2G_n) + \mathcal{O}(h^3). \end{cases} \tag{6.35}$$

将上述所获$f_{i,n}$ $(i = 1, 2, 3)$ 的表达式代入(6.34) 的最后一式得

$$\begin{aligned} \tilde{y}_{n+1} =& y(t_n) + h(b_1 + b_2 + b_3)\hat{f}_n + h^2(b_2c_2 + b_3c_3)F_n \\ &+ \frac{1}{2}h^3[2b_3c_2a_{32}F_n\hat{f}'_n + (b_2c_2^2 + b_3c_3^2)G_n] + \mathcal{O}(h^4). \end{aligned} \tag{6.36}$$

由相容阶定义及式(6.32) 与(6.36)，若要方法有三阶，则必有(6.33) 及下列条件成立:

$$b_1 + b_2 + b_3 = 1, \quad b_2c_2 + b_3c_3 = \frac{1}{2}, \quad b_2c_2^2 + b_3c_3^2 = \frac{1}{3}, \quad b_3c_2a_{32} = \frac{1}{6}. \tag{6.37}$$

取 {(6.33), (6.37)}的一组解

$$b_1 = \frac{1}{4}, \ \ b_2 = 0, \ \ b_3 = \frac{3}{4}, \ \ c_1 = a_{31} = 0, \ \ c_2 = a_{21} = \frac{1}{3}, \ \ c_3 = a_{32} = \frac{2}{3},$$

可得三阶 Heun 方法

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \tag{6.38}$$

取 {(6.33), (6.37)}的另一组解

$$b_1 = b_3 = \frac{1}{6}, \ \ b_2 = \frac{2}{3}, \ \ c_1 = 0, \ \ c_2 = a_{21} = \frac{1}{2}, \ \ c_3 = -a_{31} = 1, \ \ a_{32} = 2,$$

得三阶 Kutta 方法

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \tag{6.39}$$

仿上, 我们还可获得如下四级四阶方法

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{\sqrt{2}-1}{2} & \frac{2-\sqrt{2}}{2} & 0 & 0 \\ 1 & 0 & -\frac{\sqrt{2}}{2} & \frac{2+\sqrt{2}}{2} & 0 \\ \hline & \frac{1}{6} & \frac{2-\sqrt{2}}{6} & \frac{2+\sqrt{2}}{6} & \frac{1}{6} \end{array} \tag{6.40}$$

上二方法分别称为**经典 Runge-Kutta 方法**与**Gill 方法**. 当构造显式 Runge-Kutta 方法时，下述由Butcher给出的最大可达阶定理是值得注意的.

定理 6.2 (参见[2, 6])一个s 级显式 Runge-Kutta 方法的相容阶不可能超过s，且不存在五级五阶显式 Runge-Kutta 法.

以下，我们以 Gill 方法为例来说明显式 Runge-Kutta 方法的计算实现.

图 6.1 例 6.2 示意图.

算法 6.1 Gill 方法

```
function z=gill(a,b,y0,d,h)
 A=[0 0 0 0; 1/2 0 0 0;(sqrt(2)-1)/2 (2-sqrt(2))/2 0 0;...
    0 -sqrt(2)/2 (2+sqrt(2))/2 0]; c=[0 1/2 1/2 1]';
 B=[1/6 (2-sqrt(2))/6 (2+sqrt(2))/6 1/6]; t0=a;
for n=a+h:h:b
 t1=t0+c(1)*h; t2=t0+c(2)*h; t3=t0+c(3)*h; t4=t0+c(4)*h;
 Y1=y0; Y2=y0+h*kron(A(2,1),eye(d))*f(t1,Y1);
 Y3=y0+h*kron(A(3,1:2),eye(d))*[f(t1,Y1);f(t2,Y2)];
 Y4=y0+h*kron(A(4,1:3),eye(d))*[f(t1,Y1);f(t2,Y2);f(t3,Y3)];
 y1=y0+h*kron(B(1:4),eye(d))*[f(t1,Y1);f(t2,Y2);f(t3,Y3);f(t4,Y4)];
 t0=t0+h; y0=y1;
end
```

例 6.2 设置在某边防岛屿的雷达发现正东方向25 海里处有一艘敌舰侵入我领海，其正以每小时60 海里的速度向正北方向行驶，驻岛部队立即派出一艘舰艇以每小时65 海里的速度追击. 试建立我舰艇的航海轨迹模型，描绘出其轨迹图，并问经多少时间我舰艇能追上敌舰？

解 以边防岛屿为原点$O(0,0)$，敌舰初始位置记为A 点，线段OA 所在直线为x 轴，过原点O 且垂直于x 轴的直线为y 轴，建立直角座标系如图(见图6.1). 设敌我二舰在t 时刻的位置分别为D, $B(x(t),y(t))$. 由于$B(x(t),y(t))$ 点的运动方向总是指向敌舰位置D，因此，向量$\overrightarrow{BD}$ 的方向即为我舰运动轨迹的切向，由图6.1 中直角三角形ΔBCD 的边角关系可得我舰的航海轨迹模

型

$$\begin{cases} x'(t) = \dfrac{65[25 - x(t)]}{\sqrt{[25 - x(t)]^2 + [60t - y(t)]^2}}, & t > 0, \\ y'(t) = \dfrac{65[60t - y(t)]}{\sqrt{[25 - x(t)]^2 + [60t - y(t)]^2}}, & t > 0, \\ x(0) = 0, \quad y(0) = 0. \end{cases} \tag{6.41}$$

记

$$f(x, y) = \left[\frac{65(25 - x)}{\sqrt{(25 - x)^2 + (60t - y)^2}}, \quad \frac{65(60t - y)}{\sqrt{(25 - x)^2 + (60t - y)^2}} \right]^T,$$

对算法 6.1 进行适当改造，并应用于初值问题(6.41)，即可获得描绘我舰艇运动的轨迹图及追上敌舰的时间. 其计算程序如下：

```
function z=gillex(a,y0,d,h)
 t0=a; A=[0 0 0 0; 1/2 0 0 0; (sqrt(2)-1)/2...
 (2-sqrt(2))/2 0 0; 0 -sqrt(2)/2 (2+sqrt(2))/2 0];
  B=[1/6 (2-sqrt(2))/6 (2+sqrt(2))/6 1/6]; c=[0 1/2 1/2 1]'; k=25;  dist=25;
 while dist>=0.001
   t1=t0+c(1)*h; t2=t0+c(2)*h; t3=t0+c(3)*h; t4=t0+c(4)*h;
   Y1=y0; Y2=y0+h.*kron(A(2,1),eye(d))*f(t1,Y1);
   Y3=y0+h*kron(A(3,1:2),eye(d))*[f(t1,Y1);f(t2,Y2)];
   Y4=y0+h*kron(A(4,1:3),eye(d))*[f(t1,Y1);f(t2,Y2);f(t3,Y3)];
   y1=y0+h*kron(B(1:4),eye(d))*[f(t1,Y1);f(t2,Y2);f(t3,Y3);f(t4,Y4)];
   t0=t0+h; y0=y1;
   dist=sqrt((25-y0(1))^2+(60*t0-y0(2))^2);
 hold on;
   plot(y0(1),y0(2),'b.')
 hold off;
   xlabel('x');  ylabel('y');
  end
  t0
```

这里，我们考虑到计算机的舍入误差，没有取程序终止准则为dist= 0，而是将其设置为dist< 0.001(“dist”为敌我二舰之距离). 采用命令gillex(0, [0 0]′, 2, 0.001) 即得我舰艇运动的轨迹图(见图6.2), 其追上敌舰的时间$T \approx 2.62$ 小时. ■

§6.5 隐式Runge-Kutta 方法

随着方法的级数与阶数的增加，直接利用Taylor 展式来构造显式Runge-Kutta 方法已愈来愈加困难, 试图利用其方法来构造隐式Runge-Kutta 方法则更加困难. 为克服该困难，获得具体

图 6.2 Gill 方法解初值问题(6.41) 的解曲线图.

的高效隐式Runge-Kutta 方法, Butcher 于1964 年引入了下列阶简化条件 :

$$
\begin{aligned}
B(p):&\quad \sum_{i=1}^{s} b_i c_i^{l-1} = \frac{1}{l}, \quad l = 1, 2, \cdots, p \\
C(\eta):&\quad \sum_{j=1}^{s} a_{ij} c_j^{l-1} = \frac{c_i^l}{l}, \quad i = 1, 2, \cdots, s; \;\; l = 1, 2, \cdots, \eta \\
D(\xi):&\quad \sum_{i=1}^{s} a_{ij} b_i c_i^{l-1} = \frac{1}{l} b_j (1 - c_j^l), \quad j = 1, 2, \cdots, s; \;\; l = 1, 2, \cdots, \xi,
\end{aligned}
$$

这里，$B(p)$, $C(\eta)$ 分别意味着系统(6.1) 的精确解$y(t)$ 满足

$$
y(t_n + h) = y(t_n) + h \sum_{j=1}^{s} b_j y'(t_n + c_j h) + \mathcal{O}(h^{p+1}),
$$

$$
y(t_n + c_i h) = y(t_n) + h \sum_{j=1}^{s} a_{ij} y'(t_n + c_j h) + \mathcal{O}(h^{\eta+1}).
$$

基于上述阶简化条件, Butcher进一步给出了Runge-Kutta方法的相容阶判据.

定理 6.3 (参见[3, 9]) 若Runge-Kutta 方法(6.27) 满足阶简化条件$C(\eta)$, $D(\xi)$，且p 为满足$B(p)$ 及不等式: $p \leqslant \min\{\eta + \xi + 1, 2\eta + 2\}$ 的最大正整数, 则该方法为p 阶相容的.

依据阶简化条件，人们将常用隐式Runge-Kutta 方法分为六类，详见下表 :

表 6.1 六类隐式 Runge-Kutta 方法

方 法	阶 条 件	相 容 阶
Gauss	$B(2s),\ C(s),\ D(s)$	$2s$
Radau IA	$B(2s-1),\ C(s-1),\ D(s)$	$2s-1$
Radau IIA	$B(2s-1),\ C(s),\ D(s-1)$	$2s-1$
Lobatto IIIA	$B(2s-2),\ C(s),\ D(s-2)$	$2s-2$
Lobatto IIIB	$B(2s-2),\ C(s-2),\ D(s)$	$2s-2$
Lobatto IIIC	$B(2s-2),\ C(s-1),\ D(s-1)$	$2s-2$

利用上述理论, 我们可构造隐式Runge-Kutta 方法. 考虑二级Runge-Kutta 方法, 若有$B(3)$, $C(1)$, $D(1)$成立, 即

$$\begin{cases} \sum\limits_{i=1}^{2} b_i c_i^{l-1} = \dfrac{1}{l}, \quad l = 1,2,3, \\ \sum\limits_{j=1}^{2} a_{ij} = c_i, \quad i = 1,2, \\ \sum\limits_{i=1}^{2} a_{ij} b_i = b_j(1-c_j), \quad j = 1,2, \end{cases} \tag{6.42}$$

则由定理6.3 可知方法至少是3阶的. 特别, 在(6.42) 中取$c_1 = 0,\ a_{11} = \frac{1}{4}$, 则可解得

$$c_2 = \frac{2}{3},\ \ b_1 = \frac{1}{4},\ \ b_2 = \frac{3}{4},\ \ a_{12} = -\frac{1}{4},\ \ a_{21} = \frac{1}{4},\ \ a_{22} = \frac{5}{12},$$

从而得二级三阶 Radau IA 方法

$$\begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

在(6.42) 中取$c_1 = \frac{1}{3}, a_{11} = \frac{5}{12}$, 则可解得

$$c_2 = 1,\ \ b_1 = \frac{3}{4},\ \ b_2 = \frac{1}{4},\ \ a_{12} = -\frac{1}{12},\ \ a_{21} = \frac{3}{4},\ \ a_{22} = \frac{1}{4},$$

从而获得二级三阶 Radau IIA 方法

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

在Runge-Kutta 方法(6.27) 中取$s = 3, c_1 = 0, c_2 = \frac{1}{2}, c_3 = 1$, 并使$B(4), C(2), D(2)$ 成立, 则可得三级四阶 Lobatto IIIC 方法

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

类似地, 我们还可得到其它类型的隐式 Runge-Kutta 方法. 为今后使用方便起见, 现将文献[3, 9]中所给1-3 级各类方法列表如下:

表 6.2　2, 4, 6 阶 Gauss 方法

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{1}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{1}{2}+\frac{\sqrt{15}}{10} & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

表 6.3　1, 3, 5 阶 RadauIA 方法

$$\begin{array}{c|c} 0 & 1 \\ \hline & 1 \end{array}$$

$$\begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

$$\begin{array}{c|ccc} 0 & \frac{1}{9} & \frac{-1-\sqrt{6}}{18} & \frac{-1+\sqrt{6}}{18} \\ \frac{6-\sqrt{6}}{10} & \frac{1}{9} & \frac{88+7\sqrt{6}}{360} & \frac{88-43\sqrt{6}}{360} \\ \frac{6+\sqrt{6}}{10} & \frac{1}{9} & \frac{88+43\sqrt{6}}{360} & \frac{88-7\sqrt{6}}{360} \\ \hline & \frac{1}{9} & \frac{16+\sqrt{6}}{36} & \frac{16-\sqrt{6}}{36} \end{array}$$

表 6.4　1, 3, 5 阶 RadauIIA 方法

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}$$

表 6.5　2, 4, 6 阶 LobattoIIIA 方法

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{5-\sqrt{5}}{10} & \frac{11+\sqrt{5}}{120} & \frac{25-\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & \frac{-1+\sqrt{5}}{120} \\ \frac{5+\sqrt{5}}{10} & \frac{11-\sqrt{5}}{120} & \frac{25+13\sqrt{5}}{120} & \frac{25+\sqrt{5}}{120} & \frac{-1-\sqrt{5}}{120} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

表 6.6　2, 4, 6 阶 LobattoIIIB 型方法

$$\begin{array}{c|cc} 0 & \frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ 1 & \frac{1}{6} & \frac{5}{6} & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$$\begin{array}{c|cccc} 0 & \frac{1}{12} & \frac{-1-\sqrt{5}}{24} & \frac{-1+\sqrt{5}}{24} & 0 \\ \frac{5-\sqrt{5}}{10} & \frac{1}{12} & \frac{25+\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & 0 \\ \frac{5+\sqrt{5}}{10} & \frac{1}{12} & \frac{25+13\sqrt{5}}{120} & \frac{25-\sqrt{5}}{120} & 0 \\ 1 & \frac{1}{12} & \frac{11-\sqrt{5}}{24} & \frac{11+\sqrt{5}}{24} & 0 \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

表 6.7　2, 4, 6 阶 LobattoIIIC 方法

$$\begin{array}{c|cc} 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$$\begin{array}{c|cccc} 0 & \frac{1}{12} & -\frac{\sqrt{5}}{12} & \frac{\sqrt{5}}{12} & -\frac{1}{12} \\ \frac{5-\sqrt{5}}{10} & \frac{1}{12} & \frac{1}{4} & \frac{10-7\sqrt{5}}{60} & \frac{\sqrt{5}}{60} \\ \frac{5+\sqrt{5}}{10} & \frac{1}{12} & \frac{10+7\sqrt{5}}{60} & \frac{1}{4} & -\frac{\sqrt{5}}{60} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

§6.6 隐式方法的有效实现

一个微分方程数值方法构造出来后，要想真正在计算机上有效实现，求出合乎原问题要求的数值解，还需克服许多困难，如误差如何估计，步长及计算初值如何选取，隐式方法及多步方法如何处理等. 在这些问题中，方法的阶与步长的选择是数值求解微分方程的首要问题，它直接影响计算精度和计算效率. 对于方法的阶，我们一般要求其不超过原问题精确解的可微次数. 但是，高阶显式方法的稳定性能通常较差，因此阶数很高的显式方法一般没有实用价值. 对于方法步长的选取，从减少截断误差的角度来看，应采用小步长，但是步长取得过小，计算量将增大，而引起大的舍入误差. 为此在步长选择时，我们必须二者兼顾，合理选取. 从计算实现来说，显式方法易于处理, 而隐式方法的处理较为困难. 但是, 就同类型同阶的数值方法而言, 隐式方法的稳定性比显式方法更佳. 为此，本节主要探讨隐式方法的有效实现，重点介绍迭代技巧和预估- 校正技巧.

隐式方法的实现问题实质上是非线性方程的求解问题，因此第二章所介绍的迭代方法一般均可用于隐式方法的有效实现. 鉴于Newton 迭代法在计算方面的优势, 下面我们主要以其为例来说明隐式方法的实现.

对于隐式线性多步法(6.18)，不妨置$\alpha_k = 1$，并记$\omega_n = \sum\limits_{i=0}^{k-1}(h\beta_i f_{n+i} - \alpha_i y_{n+i})$, 则该方法可写为

$$y_{n+k} = h\beta_k f(t_{n+k}, y_{n+k}) + \omega_n, \quad \beta_k \neq 0. \tag{6.43}$$

应用Newton 迭代法于(6.43)得

$$y_{n+k}^{(r+1)} = y_{n+k}^{(r)} - \left[I_d - h\beta_k \frac{\partial f(t_{n+k}, y_{n+k}^{(r)})}{\partial y}\right]^{-1} \left[y_{n+k}^{(r)} - h\beta_k f(t_{n+k}, y_{n+k}^{(r)}) - \omega_n\right], \quad r = 0, 1, 2, \cdots. \tag{6.44}$$

计算格式(6.44)的终止准则可取为

$$\|y_{n+k}^{(r+1)} - y_{n+k}^{(r)}\| < \varepsilon \quad \text{或} \quad \left\|y_{n+k}^{(r+1)} - h\beta_k f(t_{n+k}, y_{n+k}^{(r+1)}) - \omega_n\right\| < \eta,$$

其中ε, η为预定精度. 当上准则之一成立时, 则计算终止, 且取当前步数值解$y_{n+k} \approx y_{n+k}^{(r+1)}$. 以下是三阶Adams-Moulton 方法(6.21) 求解初值问题(6.1) 的计算程序，其中计算启动值y_0, y_1 及每步迭代初值$y_{n+k}^{(0)}$ 均由三阶Heun 方法(6.38) 获得，且取迭代控制精度$\varepsilon = \eta = 10^{-12}$.

算法 6.2 三阶 Adams-Moulton 方法

```
function z=adams(a,b,y0,d,h)
 t0=a; alpha=[0 -1 1]'; beta=[-1 8 5]'/12;
 A=[0 0 0;1/3 0 0;0 2/3 0]; B=[1/4 0 3/4]; c=[0 1/3 2/3]';
 tc1=t0+c(1)*h; tc2=t0+c(2)*h; tc3=t0+c(3)*h;
 Y1=y0; Y2=y0+h*kron(A(2,1),eye(d))*f(tc1,Y1);
 Y3=y0+h*kron(A(3,1:2),eye(d))*[f(tc1,Y1);f(tc2,Y2)];
 y1=y0+h*kron(B(1:3),eye(d))*[f(tc1,Y1);f(tc2,Y2);f(tc3,Y3)];
 for n=a+2*h:h:b
```

```
    t1=t0+h; t2=t0+2*h; tc1=t0+(c(1)+1)*h;
    tc2=t0+(c(2)+1)*h; tc3=t0+(c(3)+1)*h;
    Y1=y1; Y2=y1+h*kron(A(2,1),eye(d))*f(tc1,Y1);
    Y3=y1+h*kron(A(3,1:2),eye(d))*[f(tc1,Y1);f(tc2,Y2)];
    y20=y1+h*kron(B(1:3),eye(d))*[f(tc1,Y1);f(tc2,Y2);f(tc3,Y3)];
    w=h*(beta(1)*f(t0,y0)+beta(2)*f(t1,y1))-(alpha(1)*y0+alpha(2)*y1);
    err1=1; err2=1;
      while err1>=10^(-12) &  err2>=10^(-12)
          r=y20-h*beta(3)*f(t2,y20)-w;
          y21=y20-inv(eye(d)-h*beta(3)*df(t2,y20))*r;
          err1=norm(y21-y20); err2=norm(r);
          y20=y21;
      end
  y2=y20; t0=t0+h; y0=y1; y1=y2;
end
```

例 6.3 设有初值问题

$$\begin{cases} x'(t) = \dfrac{x(t)+y(t)}{1+\sqrt{x^2(t)+y^2(t)}} + f(t), \quad t \in [0, 8\,\pi\,], \\ y'(t) = \dfrac{x(t)-y(t)}{1+\sqrt{x^2(t)+y^2(t)}} + g(t), \quad t \in [0, 8\,\pi\,], \\ x(0) = \pi\,, \quad y(0) = 0, \end{cases} \tag{6.45}$$

其中$f(t),\ g(t)$ 为使得该初值问题有唯一解$x(t) = (\,\pi\, + t)\cos t,\ y(t) = (\,\pi\, + t)\sin t$的已知函数. 试取步长$h = \pi\,/100$应用三阶Adams-Moulton 方法(6.21) 计算该初值问题，并作出其数值解的曲线图及整体误差图.

解 应用算法6.2 可获得初值问题(6.45) 的数值解，其整体误差$err \triangleq \|X_n - X(t_n)\|_2$ 及解曲线见图6.3，这里

$$X_n \triangleq [x_n, y_n]^T \approx X(t_n) \triangleq [x(t_n), y(t_n)]^T, \quad t_n = nh.$$

由其整体误差图可知，该Adams-Moulton 方法关于初值问题(6.45) 在求解区间$[0, 8\,\pi\,]$ 内的计算是有效的，其精度达10^{-5} 量级. ■

对于隐式Runge-Kutta方法(6.29)，其计算实现的关键是每步需解其级值向量

$$Y_n = e \otimes y_n + h(A \otimes I_d)F(t_n, Y_n). \tag{6.46}$$

应用Newton 迭代法到(6.46)得

$$Y_n^{(r+1)} = Y_n^{(r)} - \left[I_{sd} - h(A \otimes I_d)\frac{\partial F(t_n,\ Y_n^{(r)})}{\partial Y}\right]^{-1}\left[Y_n^{(r)} - e \otimes y_n - h(A \otimes I_d)F(t_n, Y_n^{(r)})\right], \quad r = 0, 1, 2, \cdots, \tag{6.47}$$

图 6.3 三阶Adams-Moulton 方法解初值问题(6.45) 的整体误差图及解曲线图.

其每步迭代初值$Y_n^{(0)}$ 可由同阶显式Runge-Kutta 方法给出, 计算终止准则可取为

$$\|Y_n^{(r+1)} - Y_n^{(r)}\| < \varepsilon \quad 或 \quad \left\|Y_n^{(r)} - e \otimes y_n - h(A \otimes I_d)F(t_n, Y_n^{(r)})\right\| < \eta,$$

其中ε, η 为预定精度. 当上准则之一成立时, 则取当前步的级值$Y_n \approx Y_n^{(r+1)}$, 将其代入(6.27) 的第二式即获当前步的数值解y_{n+1}.

下面，我们以二级 Gauss方法为例给出其计算程序，其中每步迭代初值$Y_n^{(0)}$ 均由三阶Heun方法(6.38) 获得，且取迭代控制精度$\varepsilon = \eta = 10^{-12}$.

算法 6.3 二级 Gauss 方法

```
function z=Gauss(a,b,y0,d,h)
 A=[1/4 1/4-sqrt(3)/6;1/4+sqrt(3)/6 1/4]; B=[1/2 1/2];
 c=[1/2-sqrt(3)/6;1/2+sqrt(3)/6]; A1=[0 0 0;1/3 0 0;0 2/3 0];
 B1=[1/4 0 3/4]; t0=a; s=2;
for n=a+h:h:b
  err1=1; err2=1; t=kron(ones(s,1),t0)+h*c;
  tc11=t0; tc12=t0+c(1)*h/3; tc13=t0+2*c(1)*h/3;
  Y11=y0; Y12=y0+h*c(1)*kron(A1(2,1),eye(d))*f(tc11,Y11);
  Y13=y0+h*c(1)*kron(A1(3,1:2),eye(d))*[f(tc11,Y11);f(tc12,Y12)];
  Y10=y0+h*c(1)*kron(B1(1:3),eye(d))*[f(tc11,Y11);f(tc12,Y12);f(tc13,Y13)];
  tc21=t0; tc22=t0+c(2)*h/3; tc23=t0+2*c(2)*h/3;
  Y21=y0; Y22=y0+h*c(2)*kron(A1(2,1),eye(d))*f(tc21,Y21);
  Y23=y0+h*c(2)*kron(A1(3,1:2),eye(d))*[f(tc21,Y21);f(tc22,Y22)];
  Y20=y0+h*c(2)*kron(B1(1:3),eye(d))*[f(tc21,Y21);f(tc22,Y22);f(tc23,Y23)];
  Y0=[Y10;Y20];
 while err1>=10^(-12) &  err2>=10^(-12)
```

图 6.4 二级Gauss 方法解初值问题(6.45) 的整体误差图及解曲线图.

```
    F=[f(t(1),[Y0(1);Y0(2)]);f(t(2),[Y0(3);Y0(4)])];
    r=Y0-kron(ones(s,1),y0)-h*kron(A,eye(d))*F;
    dF=[df(t(1),[Y0(1);Y0(2)]) zeros(2,2);zeros(2,2) df(t(2),[Y0(3);Y0(4)])];
    Y1=Y0-inv(eye(s*d)-h*kron(A,eye(d))*dF)*r;
    err1=norm(Y1-Y0); err2=norm(r); Y0=Y1;
 end
    y1=y0+h*kron(B,eye(d))*[f(t(1),[Y0(1);Y0(2)]);f(t(2),[Y0(3);Y0(4)])];
    t0=t0+h; y0=y1;
end
```

例 6.4 试取步长$h = \pi/100$，应用二级Gauss 方法计算初值问题(6.45)，作出其数值解的曲线图及整体误差图.

解 应用算法6.3可获得初值问题(6.45)的数值解，其整体误差$err \triangleq \|X_n - X(t_n)\|_2$ 及解曲线见图6.4，这里

$$X_n \triangleq [x_n, y_n]^T \approx X(t_n) \triangleq [x(t_n), y(t_n)]^T, \quad t_n = nh.$$

由其整体误差图可知，二级Gauss 方法关于初值问题(6.45) 在求解区间$[0, 8\pi]$内的计算是有效的，其精度达10^{-8}量级. ■

隐式方法的另一个有效处理途径是预估-校正方法. 理论分析表明，迭代法中的初始迭代值若以与迭代公式同阶或低一阶的公式计算，则至多只需一次迭代，其迭代值精度即可与迭代公式本身的固有精度同阶. 为此，我们可先用显式公式计算迭代初值，然后用隐式公式修正一次，即可获得满足精度要求的数值解. 这里, 前者称为**预估式**, 后者称为**校正式**, 其所组合的方法称为**预估- 校正方法**或**预校方法**. 下面, 我们以二步三阶显式公式(6.20) 作为预估式，以三

阶Adams-Moulton 公式(6.21) 作为校正式，组合成如下预校算法：

$$\begin{cases} \text{预估}: \tilde{y}_{n+2} = -4y_{n+1} + 5y_n + 2h[2f_{n+1} + f_n], \\ \text{校正}: y_{n+2} = y_{n+1} + \dfrac{h}{12}[5f(t_{n+2}, \tilde{y}_{n+2}) + 8f_{n+1} - f_n]. \end{cases} \tag{6.48}$$

其中方法的启动值y_0, y_1 可分别由原问题初值及同阶单步方法获得.

为进一步提高预校算法的精度，我们通常以预估式及校正式的误差估计量来分别修正预估值和校正值. 如在式(6.48) 中，若取$y_{n+1} \approx y(t_{n+1})$, $y_n \approx y(t_n)$ ，则根据(6.17)有

$$y(t_{n+2}) - \tilde{y}_{n+2} \approx \frac{1}{6}h^4 y^{(4)}(t_n), \quad y(t_{n+2}) - y_{n+2} \approx -\frac{1}{24}h^4 y^{(4)}(t_n).$$

由上两式可得其事后误差估计式

$$y(t_{n+2}) - \tilde{y}_{n+2} \approx \frac{4}{5}(y_{n+2} - \tilde{y}_{n+2}), \tag{6.49}$$

$$y(t_{n+2}) - y_{n+2} \approx -\frac{1}{5}(y_{n+2} - \tilde{y}_{n+2}), \tag{6.50}$$

记$p_{n+2}, m_{n+2}, c_{n+2}$ 分别为第$n+1$ 个计算步的预估值，修正值及校正值，且分别以式(6.49)及(6.50) 的右端作为预估值与校正值的修正量，则可得如下**预估- 改进-校正方法**：

$$\begin{cases} \text{预估}: & p_{n+2} = -4y_{n+1} + 5y_n + 2h(2f_{n+1} + f_n), \\ \text{改进}: & m_{n+2} = p_{n+2} + \dfrac{4}{5}(c_{n+1} - p_{n+1}), \\ & F_{n+2} = f(t_{n+2}, m_{n+2}), \\ \text{校正}: & c_{n+2} = y_{n+1} + \dfrac{h}{12}(5F_{n+2} + 8f_{n+1} - f_n), \\ \text{改进}: & y_{n+2} = c_{n+2} - \dfrac{1}{5}(c_{n+2} - p_{n+2}), \\ & f_{n+2} = f(t_{n+2}, y_{n+2}). \end{cases} \tag{6.51}$$

上述数值格式中，计算预估式的改进值m_{n+2} 时采用的是前一步的修正量$\frac{4}{5}(c_{n+1} - p_{n+1})$，其原因是当前步的校正值$c_{n+2}$ 此时还未算出. 此外，在启动格式(6.51)时，我们用三阶Heun 方法计算其启动值y_1，并取$c_1 - p_1 = 0$. 该算法计算程序如下：

算法 6.4 预估- 改进- 校正算法

```
function z=pmcm(a,b,y0,d,h)
 t0=a; c1=[0 0]'; p1=[0 0]';
 A=[0 0 0;1/3 0 0;0 2/3 0]; B=[1/4 0 3/4]; c=[0 1/3 2/3]';
 tc1=t0+c(1)*h; tc2=t0+c(2)*h; tc3=t0+c(3)*h;
 Y1=y0; Y2=y0+h*kron(A(2,1),eye(d))*f(tc1,Y1);
 Y3=y0+h*kron(A(3,1:2),eye(d))*[f(tc1,Y1);f(tc2,Y2)];
 y1=y0+h*kron(B(1:3),eye(d))*[f(tc1,Y1);f(tc2,Y2);f(tc3,Y3)];
 for n=a+2*h:h:b
  t1=t0+h; t2=t0+2*h;
  p2=(-4)*y1+5*y0+2*h*(2*f(t1,y1)+f(t0,y0));
```

```
  m2=p2+4*(c1-p1)/5; F2=f(t2,m2);
  c2=y1+h*(5*F2+8*f(t1,y1)-f(t0,y0))/12;
  y2=c2-(c2-p2)/5; f2=f(t2,y2);
  t0=t1; y0=y1; y1=y2;  c1=c2; p1=p2;
end
```

例 6.5 取步长$h=\frac{\pi}{200}$, 应用预估- 改进- 校正方法(6.51) 计算二阶初值问题

$$\begin{cases} t^2x''(t)-2tx'(t)+x(t)=(1-t^2)\sin t-t(1+2\cos t), & t\in[\,\pi\,,10\,\pi\,],\\ x(\,\pi\,)=\,\pi\,,\ \ x'(\,\pi\,)=0, \end{cases} \tag{6.52}$$

试作出其数值解的曲线图，并求出其全局误差$err\triangleq\max\limits_{0<n\leqslant 1800}|x_n-x(t_n)|$，这里$x_n\approx x(t_n)$.

解 记$y(t)=x'(t)$，则初值问题(6.52)可化为

$$\begin{cases} x'(t)=y(t), & t\in[\,\pi\,,10\,\pi\,],\\ y'(t)=\frac{2}{t}y(t)-\frac{1}{t^2}x(t)+\left(\frac{1}{t^2}-1\right)\sin t-\frac{1}{t}(1+2\cos t), & t\in[\,\pi\,,10\,\pi\,],\\ x(\,\pi\,)=\,\pi\,,\ \ y(\,\pi\,)=0, \end{cases} \tag{6.53}$$

其精确解为

$$x(t)=t+\sin t,\quad y(t)=1+\cos t.$$

取步长$h=\frac{\pi}{200}$, 应用算法(6.51) 于初值问题(6.53) 可得其数值解，该数值解x_n 的曲线图见图6.5，其全局误差为

$$err\ \triangleq\ \max_{0<n\leqslant 1800}|x_n-x(t_n)|\approx 7.594438322655606e-007.$$

当用预校算法(6.48) 计算初值问题(6.52) 时，其数值解x_n 的全局误差为

$$err\ \triangleq\ \max_{0<n\leqslant 1800}|x_n-x(t_n)|\approx 2.233246545557677e-005.$$

此时，预校算法的精度低于相应的预估- 改进- 校正方法的精度. ■

§6.7 一般多步法

一般多步法是一种抽象意义上的方法, 它几乎包含当今常用的各种常微分方程初值问题数值方法, 其一般形式为：

$$\sum_{i=0}^{k}\alpha_i y_{n+i}=h\varphi_f(t_n;y_n,y_{n+1},\cdots,y_{n+k};h),\quad n=0,1,\cdots,N-k, \tag{6.54}$$

其中$h>0$ 为计算步长，$k\geqslant 1$ 是方法的步数，诸α_i为实常数, 且$\alpha_k\neq 0$，$t_n=a+nh$, $h=(b-a)/N$, $y_n\approx y(t_n)$, $\varphi_f:\ D_H\to R^d$ 是一个依赖于问题(6.1) 右函数f 的映射，这里

$$D_H=\left\{(t_n;y_n,y_{n+1},\cdots,y_{n+k};h)|\ 0<h\leqslant H,\ a<t_n\leqslant b-kh,\ y_{n+i}\in R^d\right\}.$$

图 6.5 初值问题(6.52) 的数值解.

以下恒设： 当$f \equiv 0$ 时，$\varphi_f \equiv 0$；当f 满足(6.2)时，存在着依赖于其Lipschitz常数L的常数$h_\varphi,\ L_\varphi > 0$使得

$$\|\varphi_f(t_n; y_n, y_{n+1}, \cdots, y_{n+k}; h) - \varphi_f(t_n; z_n, z_{n+1}, \cdots, z_{n+k}; h)\| \leqslant L_\varphi \sum_{i=0}^{k} \|y_{n+i} - z_{n+i}\|, \quad 0 < h \leqslant h_\varphi. \tag{6.55}$$

理论分析表明：只要$h < \min\left\{\frac{|\alpha_k|}{L_\varphi}, h_\varphi\right\}$，则从任意初值$\{y_i\}_{i=0}^{k-1}$ 出发，方法(6.54) 可唯一地确定问题(6.1) 的一个数值解序列$\{y_n\}_{n=0}^{N}$.

如前所述, 一般多步方法(6.54)是一种非常广泛的数值方法类. 若取$\varphi_f = \sum\limits_{i=0}^{k} \beta_i f(t_n + ih, y_{n+i})$，则得线性$k$步方法(6.18)；若取$\varphi_f = \sum\limits_{j=1}^{s} b_j f(t_n + c_j h, Y_{j,n})$，则得$s$级Runge-Kutta 方法(6.27). 此外，方法(6.54) 也包括预校算法及迄今为止尚待开发的各种方法. 下面, 我们将涉及一般多步方法的收敛性理论, 其必然可应用于线性多步方法、Runge-Kutta 方法、预校算法等具体方法.

定义 6.1 若问题(6.1) 满足Lipschitz 条件(6.2), 且其精确解$y(t)$ 满足

$$\sum_{i=0}^{k} \alpha_i y(t_{n+i}) = h\varphi_f(t_n; y(t_n), y(t_{n+1}), \cdots, y(t_{n+k}); h) + T_{n+k}, \quad n = 0, 1, \cdots, N-k, \tag{6.56}$$

则称其余项T_{n+k} 为方法(6.54) 在点t_{n+k} 处的局部截断误差. 进一步，若p 为满足

$$\max_{0 \leqslant n \leqslant N-k} \|T_{n+k}\| = \mathcal{O}(h^{p+1}), \qquad h \to 0 \tag{6.57}$$

的最大正数，则称方法(6.54) 是p 阶相容的. 特别，若

$$\frac{1}{h} \max_{0 \leqslant n \leqslant N-k} \|T_{n+k}\| \to 0, \qquad h \to 0, \tag{6.58}$$

则称方法(6.54) 为相容的.

定理 6.4 方法(6.54) 相容的充要条件是多项式$\rho(\xi)=\sum\limits_{i=0}^{k}\alpha_i\xi^i$满足$\rho(1)=0$，且下列极限关于$n$一致成立：

$$\lim_{h\to 0}\varphi_f(t_n;y(t_n),y(t_{n+1}),\cdots,y(t_{n+k});h)=\rho'(1)y'(t_n). \tag{6.59}$$

证明 由于f 满足Lipschitz 条件且$y(t)$ 连续可微，则由(6.56) 及Taylor 展式得

$$T_{n+k}=\rho(1)y(t_n)+h[\rho'(1)y'(t_n)-\varphi_f(t_n;y(t_n),y(t_{n+1}),\cdots,y(t_{n+k});h)]+hR_{n,h}, \tag{6.60}$$

其中$R_{n,h}=\sum\limits_{i=0}^{k}i\alpha_i\int_0^1[y'(t_n+i\theta h)-y'(t_n)]d\theta\to 0\ (h\to 0)$. 进而，由$\rho(1)=0$ 及(6.60) 可知下列极限关于n 一致成立：

$$\frac{\|T_{n+k}\|}{h}=\|[\rho'(1)y'(t_n)-\varphi_f(t_n;y(t_n),y(t_{n+1}),\cdots,y(t_{n+k});h)]+R_{n,h}\|\to 0,\quad h\to 0.$$

因此, 方法(6.54) 是相容的.

另一方面，若方法(6.54) 是相容的，则(6.60)意味着下列极限关于n 一致成立：

$$\frac{\|T_{n+k}\|}{h}=\left\|\frac{\rho(1)y(t_n)}{h}+[\rho'(1)y'(t_n)-\varphi_f(t_n;y(t_n),y(t_{n+1}),\cdots,y(t_{n+k});h)]\right\|\to 0,\quad h\to 0. \tag{6.61}$$

当方法(6.54) 应用于问题

$$y'(t)=0,\quad t\in[0,1],\quad y(0)=1 \tag{6.62}$$

时，鉴于由$f\equiv 0$ 有$\varphi_f\equiv 0$，且问题(6.62) 的精确解为$y(t)\equiv 1$，由此及(6.61) 得$\rho(1)=0$. 故由$\rho(1)=0$ 及(6.61)即知(6.59) 成立. ■

记$\rho(\xi)=\sum\limits_{i=0}^{k}\alpha_i\xi^i,\ \sigma(\xi)=\sum\limits_{i=0}^{k}\beta_i$, 则由定理6.4可直接推得下列结果.

推论 6.1 线性多步法(6.18) 相容的充要条件是$\rho(1)=0,\ \rho'(1)=\sigma(1)$.

推论 6.2 Runge-Kutta 方法(6.27) 相容的充要条件是$\sum\limits_{j=1}^{s}b_j=1$.

一个数值方法的计算精度除受其本身的相容阶制约外，还与其稳定性相关，后者反映了方法的误差传播特性. 下面, 我们将讨论方法(6.54) 的稳定性. 为此, 我们在方法(6.54)中引入局部扰动ε_n, 并设其相应的扰动解序列$\{z_n\}$满足

$$\begin{cases}\sum\limits_{i=0}^{k}\alpha_i z_{n+i}=h[\varphi_f(t_n;z_n,\cdots,z_{n+k};h)+\varepsilon_{n+k}],\quad n=0,1,\cdots,N-k,\\ z_j=y_j+\varepsilon_j,\quad j=0,1,\cdots,k-1.\end{cases} \tag{6.63}$$

定义 6.2 方法(6.54) 称为是零稳定的，若对任意满足Lipschitz 条件的问题(6.1), 存在正常数$C,h_0>0$ 使当$0<h\leqslant h_0$ 时，使得方法(6.54) 的解序列$\{y_n\}$ 与其扰动问题(6.63) 的解序列$\{z_n\}$满足

$$\max_{0\leqslant n\leqslant N}\|z_n-y_n\|\leqslant C\max_{0\leqslant n\leqslant N}\|\varepsilon_n\|.$$

上述稳定性概念充分刻画了当步长h充分小时计算过程中局部扰动对数值解的影响. 显然，依据定义6.2直接判断零稳定是困难的. 为此, 我们引入根条件概念. 当多项式$\rho(\xi)$ 的每个根的

模不超过1，且模为1 的根为单根时, 我们称$\rho(\xi)$ 满足**根条件**. 下述基于根条件的结论可在一定程度上克服判定零稳定的困难性.

定理 6.5 方法(6.54) 为零稳定的充要条件是多项式$\rho(\xi)$ 满足根条件.

既然对于任何单步方法有$\rho(\xi) = \xi - 1$, 因此单步方法必满足根条件. 进而可知，Runge-Kutta方法均是零稳定. 此外，对于零稳定的线性多步法，Dahlquist 给出了如下阶限制结果.

定理 6.6 零稳定的线性k 步法(6.18) 的相容阶p 满足

$$p \leqslant \begin{cases} k+2, & \text{若}k \text{ 是偶数;} \\ k+1, & \text{若}k \text{ 是奇数;} \\ k, & \text{若}\frac{\beta_k}{\alpha_k} \leqslant 0. \end{cases}$$

由定理6.6 可知, 零稳定的显式线性k 步法的相容阶不可能超过k 阶. 零稳定描述的是$h \to 0$ 时的方法的稳定性，然而实际计算中，步长总是固定的，为刻画这种情况的稳定性态，我们引入绝对稳定性概念.

定义 6.3 方法(6.54) 称为是**绝对稳定的**，若该方法应用于标量试验方程

$$y' = \lambda y, \quad \lambda \in C \tag{6.64}$$

时，其解满足$\lim\limits_{n\to\infty} y_n = 0$. 相应地，称集合

$$S = \{h\lambda \in C| \text{ 方法(6.54) 应用于方程(6.64) 时满足} \lim_{n\to\infty} y_n = 0\}$$

为方法(6.54) 的绝对稳定域或稳定域.

例 6.6 试给出线性多步法(6.18) 及Runge-Kutta 方法(6.27) 的稳定域.

解 当方法(6.18) 和(6.27) 应用于问题(6.64) 时，分别产生下列差分方程

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = \hat{h} \sum_{i=0}^{k} \beta_i y_{n+i}, \quad y_{n+1} = [1 + \hat{h} b^T (I - \hat{h} A)^{-1} e] y_n, \tag{6.65}$$

其中$\hat{h} = h\lambda$, I 为s 级单位阵，$e = (1, 1, \ldots, 1)^T \in R^s$. 若设上述差分方程有形如$y_n = \xi^n$ $(\xi \neq 0)$ 的解，将其代入(6.65)，则分别得其特征方程

$$\rho(\xi) = \hat{h}\sigma(\xi), \quad \xi = 1 + \hat{h} b^T (I - \hat{h} A)^{-1} e. \tag{6.66}$$

由此可知，线性多步法(6.18) 和Runge-Kutta 方法(6.27)的稳定域分别为：

$$S_{LM} = \{\hat{h} \in C| \text{ 方程 } \rho(\xi) = \hat{h}\sigma(\xi) \text{ 的根模满足} |\xi| < 1\},$$

$$S_{RK} = \{\hat{h} \in C| \ |1 + \hat{h} b^T (I - \hat{h} A)^{-1} e| < 1\}. \quad \blacksquare$$

前述数值方法的截断误差仅反映了方法每步产生误差的局部情况，并未体现方法的全局误差情况. 当方法(6.54)应用于满足Lipschitz 条件的问题(6.1)时，若产生唯一的逼近序列$\{y_n\}$，则称误差

$$\varepsilon_n \triangleq y(t_n) - y_n, \quad n = 0, 1, \cdots, N$$

为方法(6.54) 的**整体误差**，而称$\max\limits_{0\leqslant n\leqslant N-1}\|\varepsilon_n\|$为方法关于范数$\|\cdot\|$的**全局误差**. 进一步，基于整体误差，我们可给出收敛阶的概念，其表征着数值方法的收敛速度.

定义 6.4 方法(6.54) 称为p 阶收敛的，若以该方法求解任意满足Lipschitz 条件的问题(6.1)时，只要f 充分连续可微，且$\max\limits_{0\leqslant j\leqslant k-1}\|y(t_j)-y_j\|=\mathcal{O}(h^p)$ $(h\to 0)$，则有$\varepsilon_n=\mathcal{O}(h^p)$ $(h\to 0)$. 特别，方法称为是收敛的，若以该方法求解上述问题时，$\forall t\in[a,b]$ 有

$$y_n\to y(t),\quad \text{当 } h\to 0,\ t_n\to t,\ y_i\to y_0\ (0\leqslant i\leqslant k-1).$$

在方法的相容性前提下，零稳定与收敛性具有下列等价关系.

定理 6.7 若方法(6.54) 相容，则其零稳定性与收敛性等价.

证明 命题的“收敛性$\Rightarrow$ 零稳定”部分可采用反证法及差分方程理论可证得. 此处仅证“相容性$+$ 零稳定$\Rightarrow$ 收敛性”. 事实上，由于序列$\{y(t_n)\}$ 满足

$$\begin{cases}\displaystyle\sum_{i=0}^{k}\alpha_i y(t_{n+i})=h\left[\varphi_f(t_n;y(t_n),\cdots,y(t_{n+k});h)+\frac{T_{n+k}}{h}\right],\\ y(t_j)=y_j+\varepsilon_j,\quad j=0,1,\cdots,k-1,\end{cases}$$

且方法为零稳定的，则存在常数$C,\ h_0>0$ 使得

$$\max_{0\leqslant n\leqslant N}\|\varepsilon_n\|=C\left(\max_{0\leqslant j\leqslant k-1}\|\varepsilon_j\|+\frac{1}{h}\max_{0\leqslant n\leqslant N-k}\|T_{n+k}\|\right),\quad 0<h\leqslant h_0. \tag{6.67}$$

又当$h\to 0;\ y_0,\ y_1,\ \cdots,\ y_{k-1}\to y_0$ 时有

$$\varepsilon_j=y(a+jh)-y_j\to y(a)-y_0=0,\quad 0\leqslant j\leqslant k-1,$$

且方法是相容的. 因此当$h\to 0$ 时，由(6.58) 知方法是收敛的. ■

类似可证

定理 6.8 p 阶相容且零稳定的方法(6.54)是p 阶收敛的.

由推论6.1及定理6.1, 6.5，6.7, 6.8 可分别推得

推论 6.3 满足根条件及条件：$\rho(1)=0,\ \rho'(1)=\sigma(1)$的线性多步法(6.18)是收敛的.

推论 6.4 满足根条件和条件(6.16)的线性多步法(6.18)是p 阶收敛的.

由于Runge-Kutta 方法均是零稳定的, 则由推论6.2 及定理6.3，6.7，6.8 可进一步推得

推论 6.5 满足$\sum\limits_{j=1}^{s}b_j=1$的Runge-Kutta 方法(6.27)是收敛的.

推论 6.6 满足定理6.3中的条件的Runge-Kutta 方法(6.27)是p 阶收敛的.

§6.8 刚性问题的数值处理

刚性问题广泛呈现于化学工程、电子电路、自动控制及热力学等工程领域，与通常的常微分初值问题相比较，其求解更为困难. 为介绍刚性问题，我们首先考虑如下由 Lambert[6]给出的常微分方程初值问题：

问题 I:

$$\begin{cases} \begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \begin{pmatrix} 2\sin t \\ 2(\cos t - \sin t) \end{pmatrix}, \quad t \in [0, 10], \\ \begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}; \end{cases}$$

问题 II:

$$\begin{cases} \begin{pmatrix} y_3'(t) \\ y_4'(t) \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 998 & -999 \end{pmatrix} \begin{pmatrix} y_3(t) \\ y_4(t) \end{pmatrix} + \begin{pmatrix} 2\sin t \\ 999(\cos t - \sin t) \end{pmatrix}, \quad t \in [0, 10], \\ \begin{pmatrix} y_3(0) \\ y_4(t) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}. \end{cases}$$

问题I 与问题II 有同样的精确解

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} y_3(t) \\ y_4(t) \end{pmatrix} = 2\exp(-t) \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} \sin t \\ \cos t \end{pmatrix}, \quad t \in [0, 10].$$

取步长$h = 1/10 * 2^i$ $(i = 0, 1, 2, 3, 4, 5, 6)$, 利用显式Euler 法分别计算问题I，II, 其末点处数值解$\tilde{y}_j(10)$的整体误差$\varepsilon_j = |y_j(10) - \tilde{y}_j(10)|$ $(j = 1, 2, 3, 4)$ 见表6.8.

表 6.8 显式Euler 法求解问题I，II 的整体误差

h \ ε_j	问题I		问题II	
	ε_1	ε_2	ε_3	ε_4
0.1	8.7014e-003	1.9646e-002	1.7714e+192	1.7679e+195
0.05	4.1652e-003	9.7393e-003	——	——
0.025	2.0377e-003	4.8492e-003	——	——
0.0125	1.0078e-003	2.4196e-003	——	——
0.00625	5.0118e-004	1.2085e-003	——	——
0.003125	2.4991e-004	6.0395e-004	——	——
0.0015625	1.2478e-004	3.0190e-004	1.1388e-004	1.1365e-004

表6.8 表明：显式Euler 法求解问题I是有效的. 但该方法用于求解问题II 时; 若取步长$h = 0.1$，则产生巨大误差; 若取步长$h = 1/10 * 2^i$ $(i = 1, 2, 3, 4, 5)$, 则产生溢出; 仅在步长逐步减半至非常小步长$h = 0.0015625$ 时，算法才产生有效数值结果. 对于上述数值实验，我们自然要问：同一数值方法求解形式上类似且具相同精确解的二个初值问题时，其数值结果为何产生如此大的差异? 显然从方法的相容阶角度我们无法解释上述现象，而只能从方法的稳定性来诠释其差异. 为此，我们引入文献[9] 中的一个稳定性结果.

定理 6.9 设方法(6.54) 的稳定域为S，d 维线性问题

$$y'(t) = Ay(t) + G(t), \quad t \in [a, b]; \quad y(a) = y_0 \tag{6.68}$$

的系数矩阵A具特征值$\{\lambda_i\}_{i=1}^d$, 方法(6.54) 求解具不同初值的问题(6.68) 所得数值解分别为$\{y_n\}$和$\{z_n\}$, 则 $\lim\limits_{n\to\infty}(y_n - z_n) = 0$ 当且仅当

$$h\lambda_i \in S, \quad i = 1, 2, \ldots, d. \tag{6.69}$$

该定理表明, 在条件(6.69)下数值误差在传播过程中保持有界, 且随n 无限增大而趋于零. 按此意义我们视计算过程是数值稳定的. 记显式Euler法的稳定域为$S_{EE} = \{z \in C| \ |z + 1| < 1\}$. 由定理6.9，若要显式Euler 法分别关于问题I，II 数值稳定的, 则依次应有

$$h\lambda_i^I \in S_{EE}, \ \ h\lambda_i^{II} \in S_{EE}, \quad i = 1, 2, \tag{6.70}$$

这里

$$\lambda_1^I = -1, \ \lambda_2^I = -3, \ \lambda_1^{II} = -1, \ \lambda_2^{II} = -1000 \tag{6.71}$$

为问题I，II 的系数矩阵的特征值. 将(6.71)分别代入(6.70)可解得

$$0 < h < \frac{2}{3} \approx 0.667 \ \ (\text{关于问题 I}), \quad 0 < h < 0.002 \ \ (\text{关于问题 II}). \tag{6.72}$$

就问题I而言，表6.8中所取步长均满足方法的数值稳定条件，因此其计算是成功的. 就问题II而言，表6.8中所取步长除$h = 0.0015625$外其余均不满足数值稳定条件，因此其计算仅在取步长$h = 0.0015625$为有效.

当一个数值方法被应用于求解某常微分方程初值问题时，若由于数值稳定性的苛刻要求而使得计算过程仅在方法取极小步长时才成功，否则计算失败，则称这类现象被称为**刚性现象**，而所求解的问题称为是一个**刚性问题**. 记$Re(\lambda_M) = \max\limits_{1\leqslant j\leqslant d}|Re(\lambda_j)|$, $Re(\lambda_m) = \min\limits_{1\leqslant j\leqslant d}|Re(\lambda_j)|$. 通过对问题(6.68)的理论分析, 人们发现刚性的引起是因为该问题具大的$Re(\lambda_M)$ 及小的$Re(\lambda_m)$. 因此, Lambert (1973)引入了下列刚性问题定义.

定义 6.5 线性问题(6.68) 称为是刚性的，若其系数阵A 的特征值$\{\lambda_j\}_{j=1}^d$ 满足

$$\max_{1\leqslant j\leqslant d} Re(\lambda_j) < 0, \quad r \triangleq |Re(\lambda_M)|/|Re(\lambda_m)| \gg 1.$$

定义6.5 中r被称为线性问题(6.68) 的**刚性比**. 一般，当$r \geqslant 10$ 时，问题即视为刚性的，如前述问题II是典型的刚性问题, 其刚性比$r = 10^3$. 工程领域中的许多问题其刚性比往往高大10^6以上量级，此时问题被视为是强刚性的. 上述刚性定义事实上并未完全反映刚性问题的实质，一则其没有涵盖标量刚性问题及特征值取小正数或零的刚性问题；二则问题的刚性往往并非发生在整个求解区间，而仅发生解的慢变区间；三则刚性比大的问题并不见得就是刚性问题，如Lambert 在文献[6] 中给出了如下反例：

$$\begin{cases} \begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ -1.999 & -0.999 \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \begin{pmatrix} 2\sin t \\ 0.999(\sin t - \cos t) \end{pmatrix}, \quad t \in [0, 10], \\ \begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}. \end{cases} \tag{6.73}$$

其系数阵的特征值为$\lambda_M = -1,\ \lambda_m = -0.001$，从而刚性比$r = 1000$. 但由于其无模非常大的特征值，因此方法的稳定性对其步长并无苛刻限制，若采用适当显式方法对问题(6.73) 进行数值求解，刚性现象将不会发生. 鉴此，Shampine 与 Gear 提出了另一个刚性问题的定义.

定义 6.6 线性问题(6.68) 称为是刚性的，若其满足该问题的精确解$y(t)$ 变化缓慢, 阵A至少有一个特征值其实部是模非常巨大的负数，且其所有特征值的实部都不取大的正数.

上述刚性定义仅涉及线性问题，为将该概念引伸到非线性情形，今考虑d 维非线性问题

$$y'(t) = f(t, y(t)),\quad t \in [a, b];\quad y(a) = y_0. \tag{6.74}$$

在精确解$y(t)$ 的适当邻域

$$U \triangleq \{(t, \tilde{y}) \in [a, b] \times C^d |\ \|\tilde{y} - y(t)\| < \varepsilon\}$$

内, 由Taylor 展式有

$$\tilde{y}'(t) = J(t)\tilde{y}(t) + [y'(t) - J(t)y(t)],$$

其中$J(t) = \int_0^1 \frac{\partial f}{\partial y}[t, y(t) + \theta(\tilde{y}(t) - y(t))]d\theta$. 若设$\frac{\partial f}{\partial y}$ 在邻域U 内变化缓慢，则有$J(t) \approx \frac{\partial f(t,y(t))}{\partial y}$，从而(6.74) 中微分方程可由下列线性摄动方程来逼近:

$$\tilde{y}'(t) = \frac{\partial f(t, y(t))}{\partial y}\tilde{y}(t) + \left[y'(t) - \frac{\partial f(t, y(t))}{\partial y}y(t)\right] \tag{6.75}$$

因此，问题(6.74)是否呈刚性可依据其问题的Jacobi 阵$\frac{\partial f}{\partial y}$ 的特征值的性态来检验. 鉴此，李寿佛[9] 在定义6.6 的基础上引入了非线性刚性问题的定义.

定义 6.7 非线性问题(6.74) 称为是刚性的，若问题的精确解$y(t)$ 在区间$[a, b]$ 上变化缓慢, 其Jacobi 阵$\frac{\partial f}{\partial y}$在每一点$(t, y(t))$至少有一个特征值其实部是模很大的负数，且所有特征值的实部都不取大的正值.

由矩阵谱半径性质及定义6.7 中条件II，对于任意$(t, y) \in U$，我们有

$$\|\tfrac{\partial f(t,y)}{\partial y}\| \geqslant \rho(\tfrac{\partial f(t,y)}{\partial y}) = \max_{1\leqslant j\leqslant d}|\lambda_j(\tfrac{\partial f(t,y)}{\partial y})| \geqslant \max_{1\leqslant j\leqslant d}|Re[\lambda_j(\tfrac{\partial f(t,y)}{\partial y})]|,$$

其中$\rho(\cdot),\ \lambda_j(\cdot)$ 分别为矩阵的谱半径与特征值. 因此，刚性问题(6.74)的Lipschitz 常数L 满足

$$L \geqslant L_{\min} = \sup_{(t,y)\in D}\|\tfrac{\partial f(t,y)}{\partial y}\| \geqslant \sup_{(t,y)\in D}\{\max_{1\leqslant j\leqslant d}|Re[\lambda_j(\tfrac{\partial f(t,y)}{\partial y})]|\} \gg 1.$$

由此可见, 非线性刚性问题的特征之一是其具有异常巨大的Lipschitz 常数.

上述刚性问题定义事实上存在不同程度的局限性，定义6.5未涵盖全体线性刚性问题，定义6.6 仅涉及线性刚性问题，定义6.7是在局部线性化假设条件下产生的，因此可以说至今尚未有一个完整而全面的刚性问题定义. 但人们可以断言：刚性是初值问题本身所固有的，其实质是数值求解慢变解时，存在快速衰减的扰动，这种扰动使得慢变解的计算复杂化. 以上分析也表明，数值求解刚性问题要克服的困难关键在于解除数值稳定性对方法步长的限制. 因此，我们自然希望数值方法的稳定域尽可能大，最好能覆盖复左半$h\lambda-$ 平面中的某些无限区域. 鉴此，Dahlquist 于1963 年提出了数值方法的A-稳定概念.

定义 6.8 方法(6.54) 称为是A-稳定的, 若其稳定域$S \supset C^- \triangleq \{z \in C|\ Re(z) < 0\}$.

由于任何显式Runge-Kutta方法及显式线性多步法的稳定域均为有界域，因此其不可能包含复半平面C^-, 故知这些方法均不是A-稳定的. 对于隐式Runge-Kutta 方法与隐式线性多步法, 由定义6.8可直接推得其如下A-稳定性判据.

定理 6.10 s 级 Runge-Kutta 方法(6.27) 为A- 稳定的充要条件是其稳定函数$R(z)$ 在C^- 上解析，且$\forall z \in C^-$ 有$|R(z)| < 1$.

定理 6.11 线性多步法(6.18) 为A- 稳定的充要条件是$\forall z \in C^-$ 有

$$\rho(\xi) - z\sigma(\xi) = 0 \quad \Rightarrow |\xi| < 1.$$

定理6.11 牵涉到多项式的根模小于1 的判定，一个根模均小于1 的多项式称为**Schur 多项式**，Schur 多项式的判定可采用如下Schur 准则.

定理 6.12 （Schur 准则）设有复系数多项式

$$P(\xi) = \sum_{i=0}^{k} c_i \xi^i, \quad Q(\xi) = \sum_{i=0}^{k} c_i^* \xi^{k-i}, \quad \xi \in C,$$

其中c_i^* 为c_i的共轭复数，则$P(\xi)$为$Schur$ 多项式当且仅当$|Q(0)| > |P(0)|$且函数

$$R(\xi) \triangleq \frac{P(\xi)Q(0) - P(0)Q(\xi)}{\xi}$$

为$Schur$多项式.

例 6.7 考虑二步BDF 方法

$$3y_{n+2} - 4y_{n+1} + y_n = 2hf_{n+2}, \tag{6.76}$$

其特征多项式为$P(\xi, z) = (3 - 2z)\xi^2 - 4\xi + 1$. 根据Schur 准则, $P(\xi, z)$关于ξ的二根之模均小于1当且仅当

$$8|z - 1| < |2z - 3|^2 - 1. \tag{6.77}$$

记$z = x + iy$ $(x, y \in R)$, 则(6.77) 等价于

$$x < (\sqrt{(x-1)^2 + y^2} - 1)^2. \tag{6.78}$$

而当$x < 0$ 时，(6.78) 恒成立，因此方法(6.76) 是A- 稳定的. ■

值得注意的是线性多步法的A- 稳定性是一个非常强的条件，Dalquist(1963) 指出：任何A-稳定的线性多步法的相容阶不可能超过2. 此表明不存在高阶A- 稳定的线性多步法. 但是，对于A-稳定的Runge-Kutta 方法，则有高阶方法存在，其结论及其证明可参见文献[9, 3].

定理 6.13 s 级Gauss 方法、Radau IA 方法、Radau IIA 方法、Lobatto IIIA 方法、Lobatto IIIB 方法、Lobatto IIIC 方法均是A- 稳定的.

由该定理可知，Gauss 方法、Radau IA 方法、Radau IIA 方法、Lobatto IIIA 方法、Lobatto IIIB 方法及Lobatto IIIC 方法适用于刚性问题的计算.

图 6.6 二级Gauss 方法($h = 0.01$) 解刚性问题II 的整体误差图及解曲线图.

例 6.8 试取步长$h = 0.01$，应用二级Gauss 方法求解刚性问题II，并作出其数值解的曲线图及整体误差图.

解 取步长0.01，应用算法6.3 可获得刚性问题 II 的数值解，其整体误差$err \triangleq \|X_n - X(t_n)\|_2$ 及解曲线分别见图6.6，其中

$$X_n \triangleq [x_n, y_n]^T \approx X(t_n) \triangleq [x(t_n), y(t_n)]^T, \quad t_n = nh.$$

由其整体误差图可知，其精度达10^{-8} 量级, 因此该二级Gauss 方法关于刚性问题II 的计算是有效的. ■

习 题 六

6.1 形如

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = h\beta_k f_{n+k}$$

的k 阶方法称为Gear 方法，试确定一个三步Gear 方法，并给出其截断误差.

6.2 给出线性多步法

$$y_{n+2} + (\alpha - 1)y_{n+1} - \alpha y_n = \frac{h}{4}[(\alpha + 3)f_{n+2} + (3\alpha + 1)f_n]$$

为零稳定的条件，并证明该方法为零稳定时是二阶收敛的.

6.3 若取$c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$，试由阶条件$B(2), C(2)$ 导出一个二级Runge-Kutta 方法，并指出其方法的相容阶.

6.4 给出题6.2中$\alpha = 1$ 时的公式的绝对稳定域.

6.5 试以方法类(6.23) 的显式公式为预估式，而以三步Gear 方法作为校正式构造一个预估-改进- 校正方法.

6.6 证明Kutta 方法

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

是3阶相容的，并取步长$h = 0.01$应用该方法计算初值问题(6.52)，作出其数值解的曲线图及整体误差图.

6.7 证明线性$\theta-$方法(6.5)为A- 稳定的充要条件是$0 \leqslant \theta \leqslant \frac{1}{2}$.

6.8 (实验题) 设有初值问题

$$\begin{cases} y_1'(t) = 2ty_1(t)\ln[\max(y_2(t), 10^{-3})], & t \in [0, 10], \\ y_2'(t) = -2ty_2(t)\ln[\max(y_1(t), 10^{-3})], & t \in [0, 10], \\ y(0) = 1, \quad y_2(0) = e, \end{cases}$$

其精确解为$y_1(t) = \exp[\sin(t^2)]$, $y_2(t) = \exp[\cos(t^2)]$. 试取步长h=0.01，分别应用三阶Adams-Monlton 方法和三阶Radau IIA 方法计算该初值问题，作出其数值解的曲线图及整体误差图.

参考文献

[1] G. H. Golub, C. F. Van Loan, Matrix Computations, London：Johns Hopkins University Press, 1996.

[2] E. Hairer, S. P. Nϕsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Heidelberg：Spriner-Verlag, 1993.

[3] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Heidelberg：Spriner-Verlag, 1996.

[4] R. A. Horn and C. R. Johnson, Topics in Matrix Analysis, Cambridge: Cambridge University Press, 1991.

[5] R. Kress, Numerical Ananlysis, Heidelberg: Springer-Verlag, 1998.

[6] J. D. Lambert, Numerical Methods for Ordinary Differential Equations, Chichester: John Wiley & Sons, 1991.

[7] P. Lancaster, M. Tismenetsky, The Theory of Matrices, Orlando: Academic Press, 1985.

[8] 李荣华，冯果忱, 微分方程数值解法, 北京：高等教育出版社, 1989.

[9] 李寿佛, 刚性微分方程算法理论, 长沙：湖南科技出版社, 1997.

[10] 李岳生，黄友谦，数值逼近，北京：高等教育出版社, 1987.

[11] J. M. Ortega, W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, New York: Academic Press, 1970.

[12] Y. Saad, Iterative Methods for Sparse Linear Systems, Philadelphia: SIAM, 2003.

[13] 王沫然，Matlab 6.0 与科学计算，北京：电子工业出版社，2001.

[14] 王能超, 数值分析简明教程, 北京：高等教育出版社, 1985.

[15] 徐树方，高立，张平文，数值线性代数，北京：北京大学出版社，2013.

[16] 战同胜，数值方法简明算法习题选集，大连：大连理工大学出版社，1994.

[17] 张诚坚，何南忠, 计算方法, 北京：高等教育出版社，2008.

[18] 张诚坚，覃婷婷, 科学计算引论, 北京：科学出版社，2011.

习 题 答 案

习 题 一

1.1 提示：利用不等式$\max\limits_{1\leqslant i\leqslant n}|x_i|\leqslant\|x\|_p\leqslant\sqrt[p]{p}\max\limits_{1\leqslant i\leqslant n}|x_i|$及夹逼定理可证$\|x\|_\infty=\max\limits_{1\leqslant i\leqslant n}|x_i|$.
若记$\mu=\max\limits_{1\leqslant i\leqslant n}\sum\limits_{j=1}^{n}|a_{i,j}|=\sum\limits_{j=1}^{n}|a_{i_0,j}|$, $x^{(0)}=\left(x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)}\right)^T$, 其中

$$x_n^{(0)}=\begin{cases}|a_{i_0,j}|/a_{i_0,j}, & a_{i_0,j}\neq 0,\\ 1, & a_{i_0,j}=0,\end{cases}$$

则由$\|Ax^{(0)}\|_\infty\leqslant\|A\|_\infty\leqslant\mu$ 可证$\|A\|_\infty=\mu$.

1.2 提示：据定理 1.3证明：当$\|x-x_0\|\leqslant r$ 时，$\|\Phi(x)-x_0\|\leqslant r$.

1.3 提示：考虑到等式$B^{-1}=[I-(I-A^{-1}B)]^{-1}A^{-1}$, 然后用定理1.10证明.

1.4 其解为$x_n=8[(-1)^{n-1}+\frac{1}{2^n}]-n$ (提示: 其差分方程有特解$x_n^*=-n$).

1.5 有7位有效数字.

1.6 提示：关于m用数学归纳法证之.

习 题 二

2.1 $x\approx 6.41185742$

2.2 弦截法: 迭代5次获得$x\approx 0.64350110879328$; Steffensen方法: 迭代4次获得$x\approx 0.64350110879328$.

2.3 $x_{k+1}=\frac{1}{2}[3x_k-\varphi(x_k)]$.

2.4 提示：应用定理2.1证明.

2.5 $a=\cos(1/2)$, $x\approx 0.51097342938822$.

2.6 $x\approx 1.35204421150006$.

2.7 提示：利用Taylor展式及收敛阶定义证明.

2.8 Picard 迭代解：

$$\begin{cases}x\approx 0.50000000000000\\ y\approx 0.00000000003320\\ z\approx -0.52359877559830.\end{cases}$$

Newton迭代解：

$$\begin{cases}x\approx 0.50000000000000\\ y\approx -0.00000000000007\\ z\approx -0.52359877559960.\end{cases}$$

习 题 三

3.1 提示：注意到阵A_2的i行j列元素为

$$a_{ij}^{(1)}=a_{ij}-\frac{a_{i1}a_{1j}}{a_{11}},\quad i,j=2,3,\cdots,n.$$

3.2 $X \approx [-0.59091,\ -1.3409,\ 4.5000,\ 3.4773]^T$.

3.3 $X \approx [1.0000,\ 1.0000,\ 1.0000,\ 1.0000]^T$.

3.4 $X \approx [0.83333,\ 0.66667,\ 0.50000,\ 0.16667]^T$.

3.5 (1) 解$X = [1,\ 1,\ 1,\ 1]^T$; (2) 由$\mathrm{Cond}_\infty(A) = 4488$可知方程组是病态的; (3)$\hat{X} = [1.8020,\ -0.3564,\ 1.3366,\ 0.7822]^T$, $\hat{e} = 0.6782$.

3.6 提示：利用递推关系

$$X^{k+1} - X^* = B(X^{k+1} - X^*)$$

及B的Jordan标准型证明.

3.7 提示：应用定理3.5证明.

3.8 提示：应用定理3.5证明.

3.9 提示：分别在习题3.7和习题3.8中取$\omega = 1$, 然后用其结论证明证之.

3.10 提示：方程组的精确解为$X = [1,\ 2,\ 3,\ 4,\ 5]^T$.

习 题 四

4.1 $f(1.03) \approx 10.967$， 误差界$|R_3(1.03)| \leqslant 1.5994e-006$.

4.2 $f(0.596) \approx 0.63191$，绝对误差为$|f(0.596) - N_4(0.596)| \approx 10^{-5}$.

4.3 $H(x) = (x-2)^2(x^2-2x-1)$, $R_4(x) = \frac{f^{(5)}(\xi)}{5\ !}(x-1)^2(x-2)^2(x-3)$.

4.4 提示：令$R_3(x) = k(x)(x-x_k)^2(x-x_{k+1})^2$.

4.5 提示：令

$$\begin{aligned} P(x) = \quad & f(x_0) + (x-x_0)f[x_0,x_1] + (x-x_0)(x-x_1)f[x_0,x_1,x_2] \\ & + A(x-x_0)(x-x_1)(x-x_2), \end{aligned}$$

其中A 待定，据已知条件求出A; 然后，作辅助函数

$$\Phi(t) = f(t) - P(t) - \frac{(t-x_0)(t-x_1)^2(t-x_2)}{(x-x_0)(x-x_1)^2(x-x_2)}[f(x) - P(x)],$$

反复应用 Rolle 定理证明之.

4.6 $S(x) = x^3 + x + 1$.

4.7 $\psi(x) = 4.1976 + 22.1870x - 16.4188\sin(x) - 3.7710\exp(x)$.

4.8 $y = 0.973 + 0.050x^2$.

4.9 a=4.00, b=120.

习 题 五

5.1 $I_M = 0.9289$, $I_T = 0.8554$, $I_S = 0.9044$.

5.2 $\int_{-1}^{1} f(x)ds \approx \frac{2}{3}\left[f(0) + f(\frac{\sqrt{2}}{2}) + f(-\frac{\sqrt{2}}{2})\right]$, $I \approx 0.5001$.

5.3 $\int_b^a f(x)ds \approx \frac{2h}{45}\left[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)\right]$, 其中$h = \frac{b-a}{4}$, $x_i = a + ih$, $(i = 0,1,2,3,4)$. 该公式具有5 次代数精度，且其余项为

$$R(f) = -\frac{1}{5!}\int_a^b f^{(5)}(\xi)\prod_{i=0}^{4}(x-x_i)dx, \quad \xi \in (a,b).$$

5.4 $I_T \approx 1.6355,\ \ I_S \approx 1.6360.$

5.5 $\int_{-1}^{1} f(x)ds \approx f(-\frac{1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}}),\ \ I \approx 1.3987.$

5.6 提示：由$f(x) \in C([-1,1])$可知: $\forall \varepsilon > 0$, 存在着$[-1,1]$上的m次多项式$p(x)$使得$|f(x) - p(x)| < \varepsilon$. 若记$Q_N(f) = \sum_{n=0}^{N} A_n f(x_n)$, 则有下列不等式成立:

$$\left|\int_{-1}^{1} f(x)dx - Q_N(f)\right| \leqslant \int_{-1}^{1} |f(x) - p(x)|dx + \left|\int_{-1}^{1} p(x)dx - Q_N(p)\right| + |Q_N(p) - Q_N(f)|.$$

利用该不等式、Gauss型求积公式性质及极限定义即可证之.

5.7 逼近值: $T_6 = 2.546254733499366,\ T_8^{(0)} = 2.546254733499361.$

计算时间: $t_V \approx 0.02$秒, $t_R \approx 0.$秒.

习 题 六

6.1 三步Gear 方法

$$11y_{n+3} - 18y_{n+2} + 9y_{n+1} - 2y_n = 6hf_{n+3},$$

其截断误差为$R_n = -\frac{3}{2}h^4 y^{(4)}(t_n) + \mathcal{O}(h^5)$.

6.2 其零稳定的条件为$|\alpha| < 1$.

6.3 二级四阶 Runge-Kutta 方法

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

6.4 绝对稳定域$S = \left\{\bar{h} \in C \mid \left|\frac{1+\bar{h}}{1-\bar{h}}\right| < 1\right\}$.

6.5 预估- 改进- 校正方法

$$\begin{cases} \text{预估} & P_{n+3} = y_{n+2} + \dfrac{h}{12}(23f_{n+2} - 16f_{n+1} + 5f_n) \\ \text{改进} & m_{n+3} = P_{n+3} + \dfrac{11}{15}(C_{n+2} - P_{n+2}) \\ & F_{n+3} = f(t_{n+3}, m_{n+3}) \\ \text{校正} & C_{n+3} = \dfrac{1}{11}(18y_{n+2} - 9y_{n+1} + 2y_n) + \dfrac{6h}{11}F_{n+3} \\ \text{改进} & y_{n+3} = C_{n+3} - \dfrac{4}{15}(C_{n+3} - P_{n+3}) \\ & f_{n+3} = f(t_{n+3}, y_{n+3}). \end{cases}$$

6.6 提示：利用定理6.3 证明，其计算参考算法6.1.

6.7 提示：利用A-稳定的定义.

6.8 提示：参照例6.4 及例6.5.