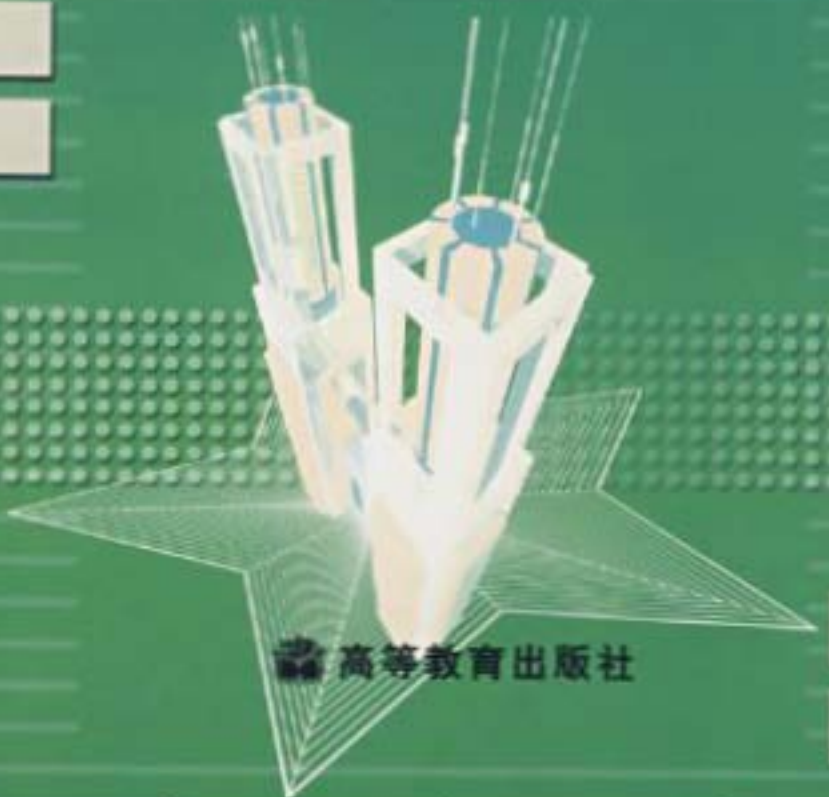
高等学校教材

工程数学

计算方法

华中科技大学数学系　　张诚坚　何南忠



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, 内容涉及数值算法概论、非线性方程数值解法、线性方程组数值解法、插值方法、数值积分及常微分方程初值问题的数值解法。该书以介绍通用数值算法为基础, 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, 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。本书正是为顺应这一知识需求而编写的。

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。在数值方法的基本概念与理论方面, 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。

全书共分六章, 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、插值方法、数值积分、常微分方程初值问题的数值解法。此外, 每章配备了一定量的习题, 以使读者通过练习, 进一步掌握其教材内容。

在编写本书过程中, 我们得到了上海大学高健副教授及华中科技大学博士生王志勇、谷伟、李东方、陈浩、牛原玲等的大力帮助, 此外也得到了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

2007 年 7 月于武汉

目 录

第一章 绪论 ····· (1)
§1.1 数值算法概论 ····· (1)
§1.2 预备知识 ····· (4)
§1.3 误差 ····· (11)
习题一 ····· (15)
第二章 非线性方程的数值解法 ····· (18)
§2.1 二分法 ····· (18)
§2.2 Jacobi 迭代法 ····· (19)
§2.3 Newton 迭代法 ····· (24)
§2.4 加速迭代方法 ····· (27)
习题二 ····· (29)
第三章 线性方程组的数值解法 ····· (31)
§3.1 Jacobi 迭代法 ····· (31)
§3.2 Gauss-Seidel 迭代法 ····· (33)
§3.3 超松弛迭代法 ····· (34)
§3.4 迭代法的收敛性 ····· (36)
§3.5 Gauss 消元法 ····· (38)
§3.6 三角分解法 ····· (43)
§3.7 追赶法 ····· (47)
§3.8 误差分析 ····· (48)
习题三 ····· (50)
第四章 插值方法 ····· (52)
§4.1 多项式插值问题 ····· (52)
§4.2 Lagrange 插值公式 ····· (54)
§4.3 差商与差分 ····· (60)
§4.4 Newton 插值公式 ····· (62)
§4.5 分段插值公式 ····· (65)
§4.6 三次样条插值 ····· (68)
§4.7 最小二乘法 ····· (72)
习题四 ····· (75)
第五章 数值积分 ····· (76)
§5.1 机械求积公式 ····· (76)
§5.2 Newton-Cotes 公式 ····· (79)

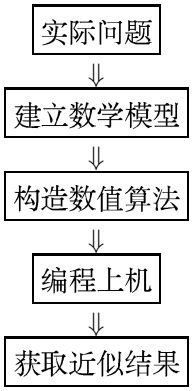
§5.3 变步长求积公式 ····· (83)
§5.4 Gauss 求积公式 ····· (87)
习题五 ····· (92)
第六章 常微分方程初值问题的数值解法 ····· (93)
§6.1 基本离散方法 ····· (93)
§6.2 Runge-Kutta 方法 ····· (98)
§6.3 数值算法理论 ····· (105)
§6.4 数值方法的有效实现 ····· (110)
§6.5 微分方程组的数值处理 ····· (115)
习题六 ····· (117)
参考文献 ····· (119)
习题答案 ····· (120)

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社会生活的各个领域。其中，数值计算是电子计算机处理实际问题的一种关键手段，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。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，数值计算方法是当今每一位从事科学研究与应用的人不可缺少的知识。本章主要介绍数值算法的预备知识及其基本思想。

§1.1　数值算法概论

一个实际问题当采用计算机来求解时，主要分下面几个步骤：



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

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

且由经典微积分知识可推得 I_n 具如下性质：

(1) $I_n > 0$,

(2) I_n 单调递减,

(3) $\lim\limits_{n\to\infty} I_n = 0$,

(4) $\frac{1}{6n} < I_{n-1} < \frac{1}{5n}$ $(n > 1)$.

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, 产生如下计算结果 (见表 1.1):

表 1.1

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

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 (4), 取

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

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数值结果 (见表 1.2).

表 1.2

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，这种数值求解方法称为**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**离散变量法**。如在后继章节中将要学习的代数方程 (组) 的迭代法、微分方程 (组) 的数值积分法及差分法等均属这类方法。

作为离散变量方法的实例，我们考察结构力学、热传导问题中经常出现的数学定解问题 —— 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] 离散化，即将 [a,b]N 等分，所得节点为

$$x_i = a + ih \quad (i = 0, 1, 2, ..., N;\ x_0 = a,\ x_N = b),$$

其中 $h = \frac{b-a}{N}$ 称为方法的步长;

(2) 将问题 (1.3) 离散化。由于

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

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

故可略去上两式中的余项 $\mathcal{O}(h^2)$, 并取 $y_i\approx y(x_i)$, 即得

$$y'(x_i)\approx\frac{y_{i+1}-y_{i-1}}{2h},\quad\text{(一阶中心差商)}\tag{1.4}$$

$$y''(x_i)\approx\frac{y_{i+1}-2y_i+y_{i-1}}{h^2}\quad\text{(二阶中心差商)}\tag{1.5}$$

将 (1.4)、(1.5) 代入 (1.3) 中得差分格式

$$\begin{cases}\dfrac{y_{i+1}-2y_i+y_{i-1}}{h^2}+p_i\dfrac{y_{i+1}-y_{i-1}}{2h}+q_iy_i=f_i, & i=1,2,\ldots,N-1,\\ y_0=\alpha,\quad y_N=\beta,\end{cases}\tag{1.6}$$

其中 $p_i=p(x_i)$, $q_i=q(x_i)$ 及 $f_i=f(x_i)$. (1.6) 实质是含 $N-1$ 个未知数 $y_1,y_2,\cdots,y_{N-1}$ 的线性方程组, 由此可解得数值解 $y_1,y_2,\cdots,y_{N-1}$.

§1.2 预备知识

在相继算法理论的学习中, 我们将涉及数值解的误差估计、稳定性及收敛性等, 为此本章引入一些相关的基础知识。

§1.2.1 范数

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

(1) $\|x\|=0$ 当且仅当 $x=0$ ($x\in R^n$),

(2) $\|\alpha x\|=|\alpha|\|x\|$, $\forall\alpha\in R$ 及 $\forall x\in R^n$,

(3) $\|x+y\|\leqslant\|x\|+\|y\|$, $\forall x,y\in R^n$.

对一维实空间 R 而言, $\|x\|$ 即为绝对值 $|x|$。下面我们将主要涉及 $l_p(p=1,2,\cdots)$ 范数:

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

特别, l_∞ 范数即为

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

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，则存在正常数 $C_2 \geqslant C_1$ 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

$$\lim_{k\to\infty} \|x^{(k)} - x\| = 0.$$

证 一方面，若序列 $\{x^{(k)}\}$ 收敛于向量 x，则

$$\lim_{k\to\infty} \|x^{(k)} - x\|_\infty = 0;$$

另一方面，若存在范数 $\|\cdot\|$ 使得

$$\lim_{k\to\infty} \|x^{(k)} - x\| = 0,$$

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.

定义 1.3 设 A 为 n 级方阵，$\|\cdot\|$ 为 R^n 中的某范数，则称

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

为矩阵 A 的从属于该向量范数的范数，记为 $\|A\|$.

利用定义 1.3 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) 对任意两个 n 级方阵 A, B，有

$$\|A + B\| \leqslant \|A\| + \|B\|, \quad \|AB\| \leqslant \|A\|\|B\|;$$

(4) 对 $\forall x \in R^n$ 及任意 n 级方阵 A，有

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

由矩阵范数的定义及其性质可知，矩阵范数与向量范数之间存在着一定的对应关系，特别地，性质 (4) 称为两者之间的**相容性**.

定理 1．3 设有 n 级实方阵 $A=(a_{ij})$，则与 l_1、l_2、l_∞ 范数相容的矩阵范数分别为

$$\|A\|_1 = \max_{1\leqslant j\leqslant n}\sum_{i=1}^{n}|a_{ij}|, \tag{1.7}$$

$$\|A\|_2 = \sqrt{\rho(A^TA)}, \tag{1.8}$$

$$\|A\|_\infty = \max_{1\leqslant i\leqslant n}\sum_{j=1}^{n}|a_{ij}|, \tag{1.9}$$

其中 $\rho(\cdot)$ 为矩阵的谱半径，其满足

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

λ_i^B 为方阵 B 的特征值.

证 此处仅证 (1.8)，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 维正交方阵 H，使得

$$A^TA = H^Tdiag(\lambda_i)H,$$

其中

$$diag(\lambda_i) = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}.$$

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

$$\|y\|_2^2 = x^TH^THx = \|x\|_2^2 = 1,$$

及

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

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 $\|A\|_2 \geqslant \sqrt{\rho(A^TA)}$. 故式 (1.8) 得证。

定理 1.4 设 A 为 n 级方阵，则对任意矩阵范数 $\|\cdot\|$ 有 $\rho(A) \leqslant \|A\|$.

证 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 1.5 $\forall \varepsilon > 0$, 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，在矩阵范数的概念下，我们可讨论矩阵序列的收敛性。

定义 1.4 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 1.6 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

$$\lim_{k\to\infty} \|A^{(k)} - A\| = 0.$$

进一步有：

定理 1.7 $\forall A \in R^{n\times n}$, $\lim\limits_{m\to\infty} A^m = 0$ 的充要条件是 $\rho(A) < 1$.

证 根据定理 1.6，本定理仅需证明：对某矩阵范数 $\|\cdot\|$, $\lim\limits_{m\to\infty} \|A^m\| = 0$ 的充要条件是 $\rho(A) < 1$. 事实上，一方面由定理 1.4，有

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

从而 $\rho(A) < 1$. 另一方面，由于 $\rho(A) < 1$，则存在 $\varepsilon > 0$，使得

$$\rho(A) + \varepsilon < 1.$$

进一步，由定理 1.5 存在 $R^{n\times n}$ 中某范数 $\|\cdot\|$ 使得

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

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

定理 1.8 设有 n 级方阵 A, 若存在矩阵范数 $\|\cdot\|$, 使得 $\|A\|<1$, 则 $I-A$ 非奇异, 且有

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

其中 I 为 n 级单位阵.

证 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

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

即

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

令 $k\to\infty$, 即得

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

由此得

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

§1.2.2 差分方程

形如

$$F(t;x(t),x(t+1),\cdots,x(t+k))=0,\tag{1.10}$$

且 $x(t)$、$x(t+k)$ 均含于 (1.10) 中的方程称为k **阶差分方程**. 在今后数值算法的学习中, 我们常常遇见的是 (1.10) 中 t 取有理数或整数情形的差分方程。在这里我们重点介绍线性差分方程

$$\sum_{j=0}^{k}a_j(n)x_{n+j}=b_n,\quad n=0,1,2,\cdots,\tag{1.11}$$

其中系数 $a_j(n)$、$b_n\in C$, 且 $a_k(n)a_0(n)\neq0$. 若给定其 k 个初始值 $x_0,x_1,\cdots,x_{k-1}$, 则由 (1.11) 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。

定理 1.9 若 (1.11) 为齐次差分方程, $x_n^{(1)}, x_n^{(2)}, \cdots, x_n^{(m)}$ 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 1.10 若 (1.11) 为齐次差分方程, $x_n^{(1)}, x_n^{(2)}, \cdots, x_n^{(k)}$ 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 1.11 若 (1.11) 为齐次差分方程, 则其解 $x_n^{(1)}, x_n^{(2)}, \cdots, x_n^{(k)}$ 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 1.12 非齐次差分方程 (1.11) 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 (1.11) 中系数 $a_i(n)$ 均与 n 无关, 则得 k 阶常系数差分方程

$$\sum_{j=0}^{k} a_j x_{n+j} = b_n, \quad n = 0, 1, 2, \cdots, \tag{1.12}$$

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

$$\sum_{j=0}^{k} a_j x_{n+j} = 0, \quad n = 0, 1, 2, \cdots, \tag{1.13}$$

设 (1.13) 有形如 $x_n = y^n$ 的解 $(y \neq 0)$, 则将其代入 (1.13) 即得代数方程

$$\sum_{j=0}^{k} a_j y^j = 0, \tag{1.14}$$

我们称 (1.14) 为方程 (1.13) 的**特征方程**, 而其根称为**特征根.** 因此我们有

定理 1.13 当且仅当 y 为特征根时, $x_n = y^n (n = 0, 1, 2, \cdots)$ 为齐次方程 (1.13) 的解。

定理 1.14 若特征方程 (1.14) 有 k 个互异根 $y_1, y_2, \cdots, y_k$, 则 (1.13) 的通解可表为 $x_n = \sum\limits_{i=1}^{k} C_i y_i^n$.

定理 1.15 若特征方程 (1.14) 的全体互异根为 $y_1, y_2, \cdots, y_p$ $(p \leqslant k)$, 其中 y_i 的重数为 m_i, 则

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

为 (1.13) 的基本解组, 从而此时 (1.13) 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 1.16 在定理 1.15 的假设条件下, 非齐次差分方程 (1.12) 的通解可表为

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

其中 x_n^* 为 (1.12) 的某个特解。

(1.12) 的特解可取为以 $x_0 = x_1 = \cdots = x_{k-1} = 0$ 为初值的解

$$x_n^* = \sum_{q=0}^{n-k} b_q \tilde{x}_{n-q-1}, \tag{1.15}$$

其中 $\tilde{x}_n$ 为相应常系数齐次差分方程初值问题

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

的解, 特别, 若诸 $b_n \equiv$ 常数 b, 且 $\sum\limits_{j=0}^{k} a_j \neq 0$, 则 (1.12) 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

$$y^4 + 2y^3 + 3y^2 + 2y + 1 = 0,$$

即

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

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

$$\begin{aligned}x_n &= (\hat{c}_1+\hat{c}_2n)(\cos\frac{2\,\pi}{3}+i\sin\frac{2\,\pi}{3})^n+(\hat{c}_3+\hat{c}_4n)(\cos\frac{2\,\pi}{3}-i\sin\frac{2\,\pi}{3})^n\\ &= (c_1+c_2n)\cos\frac{2\,\pi\,n}{3}+(c_3+c_4n)\sin\frac{2\,\pi\,n}{3},\end{aligned}$$

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

$$x_n=(c_1+c_2n)\cos\frac{2\,\pi\,n}{3}+(c_3+c_4n)\sin\frac{2\,\pi\,n}{3}+1.$$

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

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

故其初值问题的解为

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

§1.3　误　差

利用数值方法求得的数值解是一个近似结果, 因此总给人们一个不严格的感觉。其实, 现实世界各种问题的求解过程中, 误差产生是绝对的, 而精确存在是相对的。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, 有可能忽略问题的一些次要因素, 因而产生模型误差。此外, 模型的原始数据是通过观测获得的, 因而又有观测误差。在本课程的学习中, 我们仅考虑数值计算带来的误差。

§1.3.1 绝对误差与相对误差

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“截断误差”，如: 函数 $f(x)=ln(1+x)$ 有 Taylor 展开式

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

若去掉上式中的余项，利用

$$S_n(x)=\sum_{i=1}^{n}\frac{(-1)^{i-1}}{i}x^i$$

来近似计算 $f(x)$，则会产生截断误差

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

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，因而产生所谓“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 1.5 设 x^* 是某量的精确值，x 是其近似值，则称差

$$e=x^*-x$$

为 x 的绝对误差。

在定义 1.5 中, $e>0$ 意味着 x 为 x^* 的不足近似值，$e<0$ 意味着 x 为 x^* 的过量近似值。由此可见 $|e|$ 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 e 通常是困难的。实际应用中常用满足 $|e|\leqslant\varepsilon$ 的较小正数 ε 来表征绝对误差，而称 ε 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 ε，则记 $x=x^*\pm\varepsilon$.

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另一尺度。

定义 1.6 在定义 1.5 的假设条件下，称比值

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

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，$|e_r|$ 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 $|e_r| \leqslant \varepsilon_r$ 的**相对误差限** ε_r 来表征相对误差，且在实际应用中常取 $\varepsilon_r = \frac{\varepsilon}{|x|}$. 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 1.7　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

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

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 1.17　若近似值 x 具有定义 1.7 中的形式，且有 p 位有效数字，则其相对误差限

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

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.**

§1.3.2　误差的 Richardson 外推估计法

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 $y(t)$ 的近似值。设 $y(t, h)$

关于 h 可展开为

$$y(t,h)=y(t)+\sum_{i=1}^{\infty}A_ih^i, \tag{1.16}$$

若将方法的步长减半为 $\frac{1}{2}$, 则有

$$y(t,\frac{h}{2})=y(t)+\sum_{i=1}^{\infty}\frac{A_i}{2^i}h^i. \tag{1.17}$$

取 (1.16)、(1.17) 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 $y(t)$ 的近似值时, 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

$$y(t,h)=y(t)+\sum_{i=p}^{\infty}A_ih^i, \tag{1.18}$$

$$y(t,\frac{h}{2})=y(t)+\sum_{i=p}^{\infty}\frac{A_i}{2^i}h^i. \tag{1.19}$$

由上两式可得

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

从而此时得 $y(t)$ 的近似值

$$\tilde{y}(t)=\frac{1}{2^p-1}[2^py(t,\frac{h}{2})-y(t,h)], \tag{1.20}$$

其截断误差为 $\mathcal{O}(h^{p+1})$.

上述方法即为**Richardson 外推法,** 利用此方法也可获得其计算结果的误差估计。设所用方法的截断误差为 $\mathcal{O}(h^p)$, 并记

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

由 (1.18) 及 (1.19) 得

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

将上式代入 $\varepsilon_h=y(t)-y(t,h)$ 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

$$\tilde{\varepsilon}_{\frac{h}{2}} = \frac{1}{2^p - 1}[y(t, \frac{h}{2}) - y(t, h)]. \tag{1.21}$$

由于 (1.21) 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

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

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　什么叫数值方法？数值方法的基本思想及其优劣的评价标准如何？

1.2　试证明

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

及

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

1.3　验证 (1.15) 为 (1.12) 的特解。

1.4　试求解差分方程初值问题

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

1.5　古代数学家祖冲之曾以 $\frac{355}{113}$ 作为圆周率 π 的近似值，问此近似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

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

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方程

$$f(x) = 0,$$

这里 $f(x)$ 可以是代数多项式，也可以是超越函数。而非线性方程的精确求解一般均非常困难，如对于高于 4 次的代数方程，由于其不存在求根公式，因此其精确求解一般是不可行的，为此本章将介绍非线性方程的各种数值解法。

§2.1　二分法

设函数 $f(x)$ 在区间 $[a,b]$ 上连续，且 $f(a)\cdot f(b) < 0$，根据连续函数的性质可知方程 $f(x) = 0$ 在 $[a,b]$ 内一定有实根，并称 $[a,b]$ 为方程 $f(x) = 0$ 的有根区间。为明确起见，不妨假定它在 $[a,b]$ 内有唯一的实根 x^*.

把区间 $[a,b]$ 二等分，分点为 $x_0 = \frac{1}{2}(a+b)$，计算函数值 $f(\frac{a+b}{2})$，如果

$$f(\frac{a+b}{2}) = 0,$$

则求得实根

$$x^* = \frac{a+b}{2},$$

否则 $f(\frac{a+b}{2})$ 或者与 $f(a)$ 异号，或者与 $f(b)$ 异号。若

$$f(a)f(\frac{a+b}{2}) < 0,$$

说明根在区间 $[a,\frac{a+b}{2}]$ 内，这时取

$$a_1 = a,\quad b_1 = \frac{a+b}{2};$$

若

$$f(\frac{a+b}{2})f(b) < 0,$$

说明根在区间 $[\frac{a+b}{2},b]$ 内，这时取

$$a_1 = \frac{a+b}{2},\quad b_1 = b.$$

不管出现哪一种情形，新的有根区间 $[a_1,b_1]$ 的长度仅为原有根区间 $[a,b]$ 的一半。

在新的有根区间 $[a_1, b_1]$ 上重复上述二分步骤，得下列有根区间序列

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

其中每个区间仅为前一个区间的一半，二分 k 次以后得有根区间 $[a_k, b_k]$，其长度是

$$b_k - a_k = \frac{1}{2^k}(b - a). \tag{2.1}$$

由此可见，如果二分过程无限地进行下去 $(k \to \infty)$，则有根区间必定缩为一点 x^*，该点显然就是所求的根。

在实际应用中，只要能获得满足预定精度的近似值就行了。我们没必要也不可能去完成这种无穷过程。如果令有根区间 $[a_k, b_k]$ 的中点 $x_k = \frac{1}{2}(a_k + b_k)$ 为 x^* 的近似值，则在二分过程中，得到下列以 x^* 为极限的近似根序列

$$x_0, x_1, \cdots, x_k, \cdots,$$

由于

$$|x^* - x_k| \leqslant \frac{1}{2}(b_k - a_k) = \frac{1}{2^{k+1}}(b - a), \tag{2.2}$$

对于预先给定的精度 $\varepsilon > 0$，只要

$$k > \frac{ln(b - a) - ln2\varepsilon}{ln2}, \tag{2.3}$$

便有

$$|x^* - x_k| < \varepsilon,$$

这时 x_k 就是满足精度要求的近似值。

上述二分方法是方程求根问题的一种直接搜索方法，其特点是算法简单直观，收敛性总能得到保证，并且非常实用。局限性是事先要确定一个有根区间。一般地，二分法是一种高效算法，但这里的计算速度不算很快。

例 2.1 用二分法求方程 $f(x) = x^3 - x - 1 = 0$ 在区间 $[1, 1.5]$ 内的一个实根，要求误差不超过 0.005.

解 首先按公式 (2.3) 估计所要二分的次数

$$k > \frac{ln(1.5 - 1) - ln0.01}{ln2} = 5.644.$$

只要二分 6 次，便能达到所要求的精度 (这里 $f(1) < 0, f(1.5) > 0$). 具体计算结果见表 2.1.

$$x_6 = 1.3242 \approx x^*,$$
$$|x^* - x_6| = \frac{1.5 - 1}{2^7} \approx 0.0039 \leqslant 0.005.$$

表 2.1

k	a_k	b_k	x_k	$f(x_k)$ 的符号
0	1.0	1.5	1.25	-
1	1.25	1.5	1.375	+
2	1.25	1.375	1.3125	-
3	1.3125	1.375	1.3438	+
4	1.3125	1.3438	1.3281	+
5	1.3125	1.3281	1.3203	-
6	1.3203	1.3281	1.3242	-

§2.2 Jacobi 迭代法

§2.2.1 Jacobi 迭代的基本原理

迭代是一种逐步逼近的方法，已知方程 $f(x)=0$ 的一个近似根后，通常使用某个固定公式反复校正根的近似值，使之逐步精确化，一直到满足给定的精度要求为止。

具体作法是，把给定方程 $f(x)=0$ 改写成等价形式

$$x=\varphi(x), \tag{2.4}$$

在根 x^* 的附近任取一点 x_0 作为 x^* 的预测值，把 x_0 代入式 (2.4) 的右端，计算得到

$$x_1=\varphi(x_0).$$

一般说来 $x_1\neq x_0$(如果 $x_1=x_0$, 则 $x_1=x^*$)，再把 x_1 作为根 x^* 的新的预测值代入式 (2.4) 得 $x_2=\varphi(x_1)$. 重复上述步骤，则有如下迭代公式

$$x_{k+1}=\varphi(x_k)\ \ (k=0,1,2,\cdots), \tag{2.5}$$

其中 $\varphi(x)$ 称为迭代函数，并有如下迭代序列

$$x_0,x_1,\cdots,x_k,\cdots,$$

如果迭代序列 $\{x_k\}$ 的极限存在，则称迭代过程收敛，由式 (2.5), 显然有

$$x^*=\lim_{k\to\infty}x_k;$$

如果迭代序列 $\{x_k\}$ 的极限不存在，则称迭代过程发散。

方程 $x=\varphi(x)$ 的求根问题在几何上就是确定曲线 $y=\varphi(x)$ 与直线 $y=x$ 交点 p^* 的横坐标 x^*. 设迭代初值为 x_0, 曲线 $y=\varphi(x)$ 上以 x_0 为横坐标的点为 P_0，显然 $\varphi(x_0)$ 为点 P_0 的纵坐标，过 P_0 点引平行 x 轴的直线，它与直线 $y=x$ 相交于 P_0', 其横坐标为 $x_1=\varphi(x_0)$. 然后过 P_0' 点引平行 y 轴的直线，它与曲线 $y=\varphi(x)$ 的交点记作 P_1. 重复上述过程可得点列 $P_1,P_2,\cdots,P_k,\cdots$, 它们的横坐标依次由迭代公式 $x_{k+1}=\varphi(x_k)$ 所确定。如果点列 $P_1,P_2,\cdots,P_k,\cdots$ 逐步逼近 P^*，则迭代过程收敛 (见图 2.1)，否则迭代过程发散 (见图 2.2)。

图 2.1 图 2.2

例 2.2 求方程

$$f(x)=x^3-x-1=0 \tag{2.6}$$

在 x=1.5 附近的根 x^* 的近似值.

解 设将方程 (2.6) 改写成下列形式

$$x=(x+1)^{\frac{1}{3}},$$

由此得迭代公式

$$x_{k+1}=(x_k+1)^{\frac{1}{3}},\quad k=0,1,2,\cdots.$$

迭代初值取 $x_0=1.5$, 计算值用 6 位有效数字表示，迭代结果如表 2.2 所示。

从表 2.2 中可看到 x_7 与 x_8 完全相同，这时可认为 x_8 已满足方程 (2.6)，x_8 即为所求根的近似值。

$$x^*\approx x_8=1.32472.$$

上述迭代过程是收敛的。

如果将方程 (2.6) 改写成如下等价形式

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

表 2.2

k	x_k	k	x_k
0	1.5	5	1.32476
1	1.35721	6	1.32473
2	1.33086	7	1.32472
3	1.32588	8	1.32472
4	1.32494		

据此有迭代公式

$$x_{k+1} = x_k^3 - 1,$$

迭代初值仍取 $x_0 = 1.5$, 则有

$$x_1 = 2.357, x_2 = 12.39, \cdots .$$

当 k 增大时, x_k 随之增大而不趋于任何极限, 此时迭代过程发散。

通过此例说明, 迭代过程只有在一定条件下才可能收敛。一个发散的过程是没有任何实际意义的。下面我们将给出两个迭代收敛的充分性定理。

§2.2.2 迭代过程的收敛性

定理 2.1 假设迭代函数 $\varphi(x)$ 满足下列两项条件

(1) 对任意 $x \in [a, b]$, 有

$$a \leqslant \varphi(x) \leqslant b. \tag{2.7}$$

(2) 存在正数 $L < 1$, 使对任意 $x \in [a, b]$, 有

$$|\varphi\prime(x)| \leqslant L < 1. \tag{2.8}$$

则迭代过程 $x_{k+1} = \varphi(x_k)$ 对于任意初值 $x_0 \in [a, b]$ 均收敛于方程 $x = \varphi(x)$ 的根 x^*, 且有如下误差事后估计式

$$|x^* - x_k| \leqslant \frac{1}{1-L}|x_{k+1} - x_k|. \tag{2.9}$$

证 由微分中值定理

$$\begin{aligned} |x^* - x_k| &= |\varphi(x^*) - \varphi(x_{k-1})| \\ &= |\varphi'(\xi)(x^* - x_{k-1})| \leqslant L|x^* - x_{k-1}|, \end{aligned}$$

式中 ξ 是 x^* 与 x_{k-1} 之间的一点。

据此反复递推

$$\begin{aligned}|x^*-x_k| &\leqslant L|x^*-x_{k-1}| \leqslant L^2|x^*-x_{k-2}| \\ &\leqslant \cdots \leqslant L^k|x^*-x_0|.\end{aligned}$$

由此可知，当 $k\to\infty$ 时，$x_k\to x^*$，迭代序列 $\{x_k\}$ 收敛到所求根 x^*. 在上述证明中，为确保收敛，应保证对所有的迭代值 x_k 全部落在区间 $[a,b]$ 内，为此要求对任意 $x\in[a,b]$，均有 $\varphi(x)\in[a,b]$，即条件 (1)。

下面证明误差估计式 (2.9)，对任意正整数 p 有

$$\begin{aligned}|x_{k+p}-x_k| &\leqslant |x_{k+p}-x_{k+p-1}|+|x_{k+p-1}-x_{k+p-2}|+ \\ &\quad \cdots+|x_{k+1}-x_k| \\ &\leqslant L^{p-1}|x_{k+1}-x_k|+L^{p-2}|x_{k+1}-x_k|+ \\ &\quad \cdots+|x_{k+1}-x_k| \\ &= (L^{p-1}+L^{p-2}+\cdots+1)|x_{k+1}-x_k|,\end{aligned}$$

在上式中固定 k 并令 $p\to\infty$，则有

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

由此可见，只要前后两次迭代值的差值足够小，就可使近似值 x_{k+1} 达到任意的精度。在实际计算中，一般用条件 $|x_{k+1}-x_k|<\varepsilon$ 来控制迭代过程结束，其中 ε 为所要求的精度。

一般说来，定理 2.1 中的条件在较大的有根区间上是很难保证的，为此我们通常在根 x^* 的附近考察其收敛性。

定义 2.1 称迭代过程在根 x^* 的附近具有局部收敛性，指的是如果存在领域 $\triangle:|x-x^*|\leqslant\delta$，迭代过程 $x_{k+1}=\varphi(x_k)$ 对任意初值 $x_0\in\triangle$ 收敛。

定理 2.2 设 $\varphi(x)$ 在方程 $x=\varphi(x)$ 根 x^* 的附近有连续的一阶导数，且

$$|\varphi'(x^*)|<1,$$

则迭代过程 $x_{k+1}=\varphi(x_k)$ 具有局部收敛性 (作为习题，请读者证明)。

由此可见，迭代过程的收敛性通常依赖于迭代初值 x_0 的选取。

例 2.3 求方程

$$x=e^{-x}$$

在 $x=0.5$ 附近的一个根，要求精度 $\varepsilon=10^{-5}$.

解 过 $x=0.5$ 以 $h=0.1$ 为步长搜索一次，可发现所求根在区间 $[0.5,0.6]$ 内，且

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

据定理 2.1，迭代公式 $x_{k+1}=e^{-x_k}$ 对于初值 $x_0=0.5$ 是收敛的。

经过 18 次迭代后，得 $x_{18}=0.56714$, 它满足所规定的精度要求。该方程取 5 位有效数字的准确根为 $x^*=0.56714$.

§2.2.3 迭代法的收敛速度

衡量一个迭代算法的实用价值，除了需要保证它是收敛的，还要考察它的收敛速度，所谓收敛速度是指接近收敛时迭代误差的下降速度。记 $e_k=|x^*-x_k|$(即第 k 步迭代误差)，为此作如下定义

定义 2.2 当 $k\to\infty$ 时，有

$$\frac{|e_{k+1}|}{|e_k|^p}\to C \quad (C\neq 0, \text{且为常数}), \tag{2.10}$$

则称迭代过程是 p 阶收敛的。特别地，当 $p=1, 0<C<1$ 时，称作线性收敛；$p=2$ 称作平方收敛。对收敛速度有如下定理,

定理 2.3 设 $\varphi(x)$ 在 $x=\varphi(x)$ 的根 x^* 附近有连续的 p 阶导数，且

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

则迭代过程 $x_{k+1}=\varphi(x_k)$ 是 p 阶收敛的。

证 由于 $\varphi'(x^*)=0$, 根据定理 2.2 迭代过程 $x_{k+1}=\varphi(x_k)$ 具有局部收敛性。

将 $\varphi(x_k)$ 在 x^* 处作泰勒展开，得

$$\begin{aligned}\varphi(x_k) &= \varphi(x^*)+\sum_{i=1}^{p-1}\frac{\varphi^{(i)}(x^*)}{i\ !}(x_k-x^*)^i+\frac{\varphi^{(p)}(\xi)}{p\ !}(x_k-x^*)^p\\ &= \varphi(x^*)+\frac{\varphi^{(p)}(\xi)}{p\ !}(x_k-x^*)^p,\end{aligned}$$

亦即

$$\varphi(x_k)-\varphi(x^*)=\frac{\varphi^{(p)}(\xi)}{p\ !}(x_k-x^*)^p,$$

其中 ξ 是 x_k 与 x^* 之间的某一点。

由于 $\varphi(x_k)=x_{k+1},\ \varphi(x^*)=x^*$, 于是

$$|x_{k+1}-x^*|=\left|\frac{\varphi^{(p)}(\xi)}{p\ !}\right||x_k-x^*|^p,$$

由此得

$$\frac{|e_{k+1}|}{|e_k|^p} = \left|\frac{\varphi^{(p)}(\xi)}{p\,!}\right| \quad \xrightarrow{k\to\infty} \quad \left|\frac{\varphi^{(p)}(x^*)}{p\,!}\right| \neq 0.$$

所以迭代过程 $x_{k+1} = \varphi(x_k)$ 是 p 阶收敛的。

§2.3 Newton 迭代法

§2.3.1 Newton 迭代的基本思想

Newton 迭代是方程求根问题的一个极其基本的、十分重要的算法，它的基本思路是将非线性方程 $f(x) = 0$ 逐步线性化而形成迭代公式。

设已知方程 $f(x) = 0$ 的近似根 x_k，将函数 $f(x)$ 在点 x_k 处做一阶泰勒展开，则有

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k), \tag{2.11}$$

于是方程 $f(x) = 0$ 可近似地表为

$$f(x_k) + f'(x_k)(x - x_k) = 0. \tag{2.12}$$

式 (2.12) 是个线性方程，记其根为 x_{k+1}，则 x_{k+1} 就可理解为方程 $f(x) = 0$ 的一个新的近似根，即

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \tag{2.13}$$

这就是著名的 Newton 迭代公式，相应的迭代函数是

$$\varphi(x) = x - \frac{f(x)}{f'(x)} \tag{2.14}$$

Newton 迭代方法是在根 x^* 附近以 x_k 作为第 k 次迭代值，然后以 $y = f(x)$ 在点 $(x_k, f(x_k))$ 的切线与 x 轴的交点 x_{k+1} 作为方程 $f(x) = 0$ 的根 x^* 的第 $k+1$ 次迭代近似值，再作 $y = f(x)$ 在点 $(x_{k+1}, f(x_{k+1}))$ 的切线与 x 轴的交点 x_{k+2}，并以它作方程 $f(x) = 0$ 的根 x^* 的第 k+2 次迭代近似值，如此反复下去，逐步逼近方程 $f(x) = 0$ 的根 x^*(见图 2.3)。上述切线与 x 轴交点的横坐标正是按 Newton 迭代公式 (2.13) 求得的近似根。基于上述几何解释，Newton 迭代法又称作切线法。

图 2.3

§2.3.2 Newton 迭代的收敛速度

Newton 迭代可以看成是关于方程

$$x = \varphi(x), \quad \varphi(x) = x - \frac{f(x)}{f'(x)}$$

的迭代公式。如果 x^* 为方程 $f(x) = 0$ 的一个单根，则有

$$f(x^*) = 0, \quad f'(x^*) \neq 0,$$

而

$$\varphi'(x^*) = \frac{f(x^*)f''(x^*)}{[f'(x^*)]^2} = 0$$

由上节定理 2.3 的证明可知，Newton 迭代在根 x^* 的邻近是平方收敛的。因此用牛顿法求单根的收敛速度是较快的。

例 2.5 用 Newton 迭代解方程

$$x = e^{-x}$$

在 $x = 0.5$ 附近的根。

解 首先将方程 $x = e^{-x}$ 改写为

$$xe^x - 1 = 0,$$

于是

$$f(x) = xe^x - 1,$$

相应的 Newton 迭代公式为

$$x_{k+1} = x_k - \frac{x_k - e^{-x_k}}{1 + x_k},$$

取 $x_0 = 0.5$ 作为迭代初值。迭代结果列于表 2.3 中。

表 2.3

k	0	1	2	3
x_k	0.5	0.57102	0.56716	0.56714

经过 3 次迭代后得 $x_3 = 0.56714$，比较例 2.3 和例 2.5 可发现，Newton 迭代法的收敛速度是相当快的。

例 2.6 用 Newton 迭代法求 $\sqrt{115}$ 的近似值，精度 $\varepsilon = 10^{-6}$.

解 该问题可转化为应用 Newton 迭代解下列二次方程

$$x^2 - 115 = 0$$

的正根，相应的 Newton 迭代公式是

$$x_{k+1} = \frac{1}{2}(x_k + \frac{115}{x_k}).$$

取初值 $x_0 = 10$, 经 4 次迭代便得所要求的近似值

$$\sqrt{115} \approx 10.723805.$$

§2.3.3 Newton 下山法

前面我们讨论了 Newton 迭代的局部收敛性，一般而言，Newton 迭代的收敛性依赖于初值的选取，如果初值选得离所求根 x^* 较远，则可能导致迭代过程发散。

例 2.7 设有方程

$$x^3 - x - 1 = 0,$$

求该方程在 $x = 1.5$ 附近的根。

解 设初值 $x_0 = 1.5$，Newton 迭代公式是

$$x_{k+1} = x_k - \frac{x_k^3 - x_k - 1}{3x_k^2 - 1}. \tag{2.15}$$

迭代结果如下

$$x_1 = 1.34783, \quad x_2 = 1.32520, \quad x_3 = 1.32472.$$

其中第 3 步迭代值 x_3 已具有 6 位有效数字 (参考例 2.2)，迭代过程收敛。

如果选取初值 $x_0 = 0.6$(离所求根 x^* 较远)，按 Newton 迭代公式 (2.15) 求得第一步迭代值是

$$x_1 = 17.9,$$

这个结果比 x_0 更远离所求根 x^*，这样继续计算下去，迭代过程是发散的。

在实际应用 Newton 迭代时，往往很难给出一个较好的迭代初值 x_0，以保证迭代过程收敛。一般而言，对于一个收敛的迭代过程来说，在根 x^* 的附近区域内，越接近 x^* 的迭代值 x_k，其函数值的绝对值 $|f(x_k)|$ 越小。基于这点，我们可在每步迭代过程中附加一项 $|f(x)|$ 函数值单调下降的条件，即

$$|f(x_k)| > |f(x_{k+1})|, \tag{2.16}$$

强制使迭代过程收敛。满足条件 (2.16) 的算法称作下山法。

具体作法是，把 Newton 迭代的计算结果 (记作 $\bar{x}_{k+1}$)

$$\bar{x}_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \tag{2.17}$$

与前一步的迭代近似值 x_k 适当加权平均作为迭代改进值 x_{k+1}，即

$$x_{k+1} = \lambda \bar{x}_{k+1} + (1-\lambda)x_k, \tag{2.18}$$

将式 (2.17) 代入式 (2.18) 中，便得下列 Newton 下山公式

$$x_{k+1} = x_k - \lambda \frac{f(x_k)}{f'(x_k)}. \tag{2.19}$$

其中 $0 < \lambda \leqslant 1$ 称作下山因子，为保证迭代过程中下山成功，即使式 (2.16) 成立，必须选取适当的下山因子 λ.

下山因子的选取是个搜索试探过程，譬如，首先从 $\lambda = 1$ 开始试探条件 (2.16) 成立，若不成立，则将 λ 反复减半进行试算，亦即在集合

$$1, \frac{1}{2}, \frac{1}{4}, \cdots, \frac{1}{2^n}, \cdots$$

中依次挑选下山因子，直至找到某个使单调条件 (2.16) 成立的下山因子。如果在上述过程中挑选的 λ 已非常的小，但仍无法使单调条件 (2.16) 成立，这时应考虑重新选取初值 x_0，然后进行另一轮的迭代。

现在我们回到例 2.7 中来，仍取初值 $x_0 = 0.6$ 经过几次试算后，可找到 $\lambda = \frac{1}{32}$，且由式 (2.19) 算得 $x_1 = 1.140625$，这时 $|f(x_1)| < |f(x_0)|$(下山成功)。显然，$x_1 = 1.140625$ 比 $x_0 = 0.6$ 更接近于根 $x^* = 1.32472$, 因此迭代过程收敛了。

§2.4 加速迭代方法

§2.4.1 加速方法的构造

对于收敛的迭代过程，如果不论迭代次数，总是可以经过若干次迭代得到满足一定精度要求的近似值。但在实际应用中，如果收敛速度太慢则是很难接受的。因此有必要讨论如何改进迭代方法，使迭代收敛过程加速进行。

设 x_k 是第 k 次迭代近似值，利用迭代公式计算得到第 $k+1$ 次迭代近似值，这里记作 $\bar{x}_{k+1}$, 即

$$\bar{x}_{k+1} = \varphi(x_k), \tag{2.20}$$

假设 $\varphi'(x)$ 在根 x^* 的附近变化不大，估计其值大约为 L，由微分中值定理，得

$$x^* - \bar{x}_{k+1} = \varphi'(\xi)(x^* - x_k) \approx L(x^* - x_k). \tag{2.21}$$

整理后得 $\bar{x}_{k+1}$ 的事后估计式

$$x^* - \bar{x}_{k+1} \approx \frac{L}{1-L}(\bar{x}_{k+1} - x_k), \tag{2.22}$$

记

$$\begin{aligned} x_{k+1} &= \bar{x}_{k+1} + \tfrac{L}{1-L}(\bar{x}_{k+1} - x_k) \\ &= \tfrac{1}{1-L}(\bar{x}_{k+1} - Lx_k) \\ &= \tfrac{1}{1-L}(\varphi(x_k) - Lx_k). \end{aligned} \tag{2.23}$$

式 (2.23) 称作迭代加速公式。一般而言，x_{k+1} 将比 $\bar{x}_{k+1}$ 具有更高的精度。

例 2.8 用迭代加速公式求方程

$$x = e^{-x}$$

在 $x = 0.5$ 附近的根。

解 取迭代初值 $x_0 = 0.5$. 这里迭代函数为

$$\varphi(x) = e^{-x},$$

而 $\varphi'(x) = -e^{-x}$, 其在 0.5 附近的估计值为

$$\varphi'(x) \approx L = -0.6, \quad x \in [0.5, 0.6].$$

由迭代加速公式 (2.23)，得

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

经过 3 次迭代得近似根 $x_3 = 0.56714$. 比较例 2.3 和例 2.8 可见加速公式的效果是十分明显的。

§2.4.2 Aitken 加速方法

上述迭代加速公式在实际应用中常常会遇到 $\varphi'(x)$ 估计不太容易等困难，为避免对导数 $\varphi'(x)$ 的估计，可采用下述改进方法进行迭代加速。仍记 $\bar{x}_{k+1} = \varphi(x_k)$，对 $\bar{x}_{k+1}$ 再进行一次迭代，得

$$\tilde{x}_{k+1} = \varphi(\bar{x}_{k+1}),$$

且有

$$x^* - \tilde{x}_{k+1} \approx L(x^* - \bar{x}_{k+1}), \tag{2.24}$$

将式 (2.21) 与式 (2.24) 联立，消去未知常数 L，有

$$\frac{x^* - \bar{x}_{k+1}}{x^* - \tilde{x}_{k+1}} \approx \frac{x^* - x_k}{x^* - \bar{x}_{k+1}}.$$

整理后，得近似值 $\tilde{x}_{k+1}$ 的事后估计式

$$x^* - \tilde{x}_{k+1} \approx -\frac{(\tilde{x}_{k+1} - \bar{x}_{k+1})^2}{\tilde{x}_{k+1} - 2\bar{x}_{k+1} + x_k}. \tag{2.25}$$

由此得下列 Aitken 迭代加速公式：

$$\begin{cases} \text{校正}: \quad \bar{x}_{k+1} = \varphi(x_k) \\ \text{再校正}: \quad \tilde{x}_{k+1} = \varphi(\bar{x}_{k+1}) \\ \text{改进}: \quad x_{k+1} = \tilde{x}_{k+1} - \dfrac{(\tilde{x}_{k+1} - \bar{x}_{k+1})^2}{\tilde{x}_{k+1} - 2\bar{x}_{k+1} + x_k} \end{cases}$$

例 2.9 用 Aitken 方法求解方程

$$x^3 - x - 1 = 0$$

在 $x = 1.5$ 附近的根。

解 在前面例 2.2 中我们曾指出，求解这个方程的下述迭代公式

$$x_{k+1} = x_k^3 - 1.$$

是发散的。

现以该迭代公式为基础得到的 Aitken 迭代公式为

$$\begin{cases} \bar{x}_{k+1} = x_k^3 - 1 \\ \tilde{x}_{k+1} = \bar{x}_{k+1}^3 - 1 \\ x_{k+1} = \tilde{x}_{k+1} - \dfrac{(\tilde{x}_{k+1} - \bar{x}_{k+1})^2}{\tilde{x}_{k+1} - 2\bar{x}_{k+1} + x_k} \end{cases}$$

仍取初值 $x_0 = 1.5$, 经过 5 次迭代得

$$x_5 = 1.32472.$$

由此例可看到 Aitken 方法将一个原本发散的迭代改造成了一个收敛的迭代。

习题二

2.1 使用二分法求方程 $x = \frac{1}{2^x}$ 在 $[0, 1]$ 内的根, 精确到 10^{-5}.

2.2 试构造收敛的迭代公式求解下列方程：

(1) $x = \frac{cosx+sinx}{4}$; (2) $x = 4 - 2^x$.

2.3 方程 $x^3 - x^2 - 1 = 0$ 在 $x = 1.5$ 附近有根，把方程写成三种不同的等价形式：

(1) $x = 1 + \frac{1}{x^2}$, 对应迭代公式 $x_{k+1} = 1 + \frac{1}{x_k^2}$;

(2) $x^3 = 1 + x^2$, 对应迭代公式 $x_{k+1} = (1 + x_k^2)^{\frac{1}{3}}$;

(3) $x^2 = \frac{1}{x-1}$, 对应迭代公式 $x_{k+1} = \sqrt{\frac{1}{x_k - 1}}$.

判断以上三种迭代公式在 $x_0 = 1.5$ 的收敛性, 选一种收敛公式求出 $x_0 = 1.5$ 附近的根到 4 位有效数字。

2.4 已知 $x = \varphi(x)$ 在 $[a, b]$ 内有一根 x^*, $\varphi(x)$ 在 $[a, b]$ 上一阶可微, 且 $\forall x \in [a, b], |\varphi'(x) - 3| < 1$, 试构造一个局部收敛于 x^* 的迭代公式。

2.5 设 $\varphi(x)$ 在方程 $x = \varphi(x)$ 根 x^* 的邻近有连续的一阶导数, 且 $|\varphi'(x^*)| < 1$, 证明迭代公式 $x_{k+1} = \varphi(x_k)$ 具有局部收敛性。

2.6 用 Newton 迭代法求方程 $f(x) = x^3 - 2x^2 - 4x - 7 = 0$ 在 $[3, 4]$ 中的根的近似值 (精确到小数点后两位)。

2.7 试证用 Newton 迭代法求方程 $(x-2)^2(x+3) = 0$ 在 $[1, 3]$ 内的根 $x^* = 2$ 是线性收敛的。

2.8 应用 Newton 迭代法于方程 $x^3 - a = 0$, 导出求立方根 $a^{\frac{1}{3}}$ 的迭代公式, 并讨论其收敛性。

第三章　线性方程组的数值解法

在工程技术的科学计算中，常常需要求解线性方程组。因此，线性方程组的解法在计算数学中占有极其重要的地位。线性方程组的解法大致分为直接法与迭代法两大类。

§3.1　Jacobi 迭代法

首先用一个具体例子来说明 Jacobi 迭代法的基本思路。

例 3.1　求解方程组

$$\begin{cases} 10x_1 - x_2 - 2x_3 = 7.2 \\ -x_1 + 10x_2 - 2x_3 = 8.3 \\ -x_1 - x_2 + 5x_3 = 4.2 \end{cases} \tag{3.1}$$

首先从式 (3.1) 中的三个方程分离出变量 x_1, x_2, x_3, 由此将方程组 (3.1) 改写成便于迭代且等价的形式

$$\begin{cases} x_1 = 0.1x_2 + 0.2x_3 + 0.72 \\ x_2 = 0.1x_1 + 0.2x_3 + 0.83 \\ x_3 = 0.2x_1 + 0.2x_2 + 0.84 \end{cases}$$

据此建立迭代公式

$$\begin{cases} x_1^{(k+1)} = 0.1x_2^{(k)} + 0.2x_3^{(k)} + 0.72 \\ x_2^{(k+1)} = 0.1x_1^{(k)} + 0.2x_3^{(k)} + 0.83 \\ x_3^{(k+1)} = 0.2x_1^{(k)} + 0.2x_2^{(k)} + 0.84 \end{cases} \tag{3.2}$$

取迭代值 $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$, 迭代结果列于表 3.1 中。容易验证方程组 (3.1) 的准确解为 $x_1^* = 1.1, x_2^* = 1.2, x_3^* = 1.3$. 从表 3.1 中可看到，当迭代次数增加时，迭代结果越来越逼近准确解。这种迭代过程是收敛的，其迭代序列 $(x_1^{(k)}, x_2^{(k)}, x_3^{(k)})$ 以 (x_1^*, x_2^*, x_3^*) 为极限。这种迭代方法称作 Jacobi 迭代法。

从这个例子可看到，雅可比迭代法的基本思想是将方程组的求解问题，转化为重复计算一族彼此独立的线性表达式。

下面我们就一般情形下的方程组建立 Jacobi 迭代公式。

设有方程组

$$\sum_{j=1}^{m} a_{ij}x_j = b_i, a_{ii} \neq 0 \quad (i = 1, 2, \cdots, n), \tag{3.3}$$

表 3.1

.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.00000	0.00000	0.00000
1	0.72000	0.83000	0.84000
2	0.97100	1.07000	1.15000
3	1.05700	1.15710	1.24820
4	1.08535	1.18534	1.28282
5	1.09510	1.19510	1.29414
6	1.09834	1.19834	1.29504
7	1.09944	1.19981	1.29934
8	1.09981	1.19941	1.29978
9	1.09994	1.19994	1.29992

从式 (3.3) 中的第 i 个方程分离出变量 $x_i(i=1,2,\cdots,n)$, 将它改写成

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

据此建立 Jacobi 迭代公式

$$x_i^{(k+1)}=\frac{1}{a_{ii}}(b_i-\sum_{j=1,j\neq i}^{n}a_{ij}x_j^{(k)})\ \ (i=1,2,\cdots,n),\tag{3.4}$$

上述公式是 Jacobi 迭代公式的分量表达形式。我们也可以用矩阵形式来表示雅可比迭代公式。

设有方程组

$$AX=b,\tag{3.5}$$

其中 $A=(a_{ij})_n$ 为非奇异阵, $X=(x_1,x_2,\cdots,x_n)^T, b=(b_1,b_2,\cdots,b_n)^T$, 将系数 A 作如下分解

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

其中

$$U=\begin{bmatrix}0 & a_{12} & \cdots & a_{1n}\\ & 0 & \cdots & a_{2n}\\ & & \ddots & \vdots\\ \bigcirc & & & 0\end{bmatrix}$$

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

$$U=\begin{bmatrix} 0 & & & \bigcirc \\ a_{21} & 0 & & \\ \vdots & & \ddots & \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}$$

于是式 (3.5) 可写为

$$(U+D+L)X=b.$$

由此得

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

据此得矩阵形式的 Jacobi 迭代公式

$$X^{(k+1)}=D^{-1}b-D^{-1}(U+L)X^{(k)}. \tag{3.6}$$

记

$$B=-D^{-1}(U+L),\quad f=D^{-1}b,$$

便有

$$X^{(k+1)}=BX^{(k)}+f. \tag{3.7}$$

矩阵 B 称作 Jacobi 方法的迭代矩阵。

§3.2 Gauss-Seidel 迭代法

对 Jacobi 迭代公式稍加改进，就可得到实用上更为有效的计算公式。由上节讨论可知，Jacobi 迭代的每一步的迭代新值

$$X^{(k+1)}=(x_1^{(k+1)},x_2^{(k+1)},\cdots,x_n^{(k+1)})^T$$

都是用

$$X^{(k)}=(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^T$$

的全部分量计算出来的。一般说来，对于一个收敛的迭代过程，新值 $x_i^{(k+1)}(i=1,2,\cdots,n)$ 将比老的值 $x_i^{(k)}(i=1,2,\cdots,n)$ 更准确些。雅可比法在计算第 i 个分量 $x_i^{(k+1)}$ 时，已经计算出

$$x_1^{(k+1)},x_2^{(k+1)},\cdots,x_{i-1}^{(k+1)},$$

这前 $i-1$ 个新的迭代值，但却没有用在计算 $x_i^{(k+1)}$ 上。如果将这些分量利用起来，则可能得到一个收敛更快的迭代公式。

考察式 (3.4)，将公式右端前 $i-1$ 个分量的上标由 k 换成 $k+1$，则得下列 Gauss-Seidel 迭代公式：

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}) \quad (i = 1, 2, \cdots, n), \tag{3.8}$$

其矩阵形式是

$$DX^{(k+1)} = b - LX^{(k+1)} - UX^{(k)},$$

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

如果 $(D+L)^{-1}$ 存在，则有

$$X^{(k+1)} = -(D+L)^{-1}UX^{(k)} + (D+L)^{-1}b, \tag{3.9}$$

记 $B = -(D+L)^{-1}U, f = (D+L)^{-1}b$，便有

$$X^{(k+1)} = BX^{(k)} + f, \tag{3.10}$$

称矩阵 B 为 Gauss-Seidel 法的迭代矩阵。

例 3.2 用 Gauss-Seidel 迭代法解例 3.1 中的方程组。

解 Gauss-Seidel 迭代法迭代公式为

$$\begin{cases} x_1^{(k+1)} = 0.1x_2^{(k)} + 0.2x_3^{(k)} + 0.72 \\ x_2^{(k+1)} = 0.1x_1^{(k+1)} + 0.2x_3^{(k)} + 0.83 \\ x_3^{(k+1)} = 0.2x_1^{(k+1)} + 0.2x_2^{(k+1)} + 0.84 \end{cases} \tag{3.11}$$

仍取初值 $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$，按公式 (3.11) 的计算结果见表 3.2，与表 3.1 的计算结果比较，可明显看出，迭代公式 (3.11) 的效果比公式 (3.2) 的好。应当指出，一般而言 Gauss-Seidel 迭代法的收敛速度比 Jacobi 迭代法快，但这两种迭代法的收敛范围并不完全重合，而只是部分相交，有的时候 Jacobi 法可能比 Gauss-Seidel 法的收敛速度更快，甚至可举出 Jacobi 法收敛而 Gauss-Seidel 法发散的例子。

§3.3 超松弛迭代法

超松弛法是迭代方法的一种加速方法，其计算公式简单，但需要选择合适的松弛因子，以保证迭代过程有较快的收敛速度。

设有方程组

$$AX = b, \tag{3.12}$$

表 3.2

.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.00000	0.00000	0.00000
1	0.72000	0.90200	1.16440
2	1.04308	1.16719	1.28205
3	1.09313	1.19572	1.29778
4	1.09913	1.19947	1.29972
5	1.09989	1.19993	1.29996
6	1.09999	1.19999	1.30000

其中 $A=(a_{ij})_n$ 为非奇异阵，$X=(x_1,x_2,\cdots,x_n)^T, b=(b_1,b_2,\cdots,b_n)^T$, 记 $X^{(k)}$ 为第 k 步迭代近似值，则

$$r^{(k)}=b-AX^{(k)}$$

表示近似解 $X^{(k)}$ 的残余误差，于是我们可引进如下形式的加速迭代公式

$$X^{(k+1)}=X^{(k)}+w(b-AX^{(k)}), \tag{3.13}$$

其中 w 称作松弛因子，上式的分量形式为

$$x_i^{(k+1)}=x_i^{(k)}+w(b_i-\sum_{j=1}^{n}a_{ij}x_j^{(k)})\ \ (i=1,2,\cdots,n), \tag{3.14}$$

选择适当的松弛因子，可期望获得较快的收敛速度。如果在计算分量 $x_i^{(k+1)}$ 时，考虑利用已经计算出来的分量 $x_1^{(k+1)},x_2^{(k+1)},\cdots,x_{i-1}^{(k+1)}$, 又可得到一个新的迭代公式

$$x_i^{(k+1)}=x_i^{(k)}+w(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i}^{n}a_{ij}x_j^{(k)})\ \ (i=1,2,\cdots,n), \tag{3.15}$$

特别当 $a_{ii}\neq 0$ 时 $(i=1,2,\cdots,n)$, 将迭代公式 (3.15) 应用于方程组

$$\sum_{j=1}^{n}\frac{a_{ij}}{a_{ii}}x_j=\frac{b_i}{a_{ii}}\ \ (i=1,2,\cdots,n),$$

由此得下列超松弛迭代公式 (称作 SOR 方法)

$$x_i^{(k+1)}=x_i^{(k)}+\frac{w}{a_{ii}}(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i}^{n}a_{ij}x_j^{(k)})\ \ (i=1,2,\cdots,n) \tag{3.16}$$

显然当取 $w=1$ 时, 式 (3.16) 就是 Gauss-Seidel 迭代公式。可以证明为保证迭代过程收敛, 必须要求 $0<w<2$. 当 $w<1$ 时称作低松弛法, 当 $w>1$ 时称作超松弛法。

§3.4 迭代法的收敛性

迭代法有着算法简单, 程序设计容易以及可节省计算机存贮单元等优点。但是迭代法也存在着收敛性和收敛速度等方面的问题。因此, 弄清楚迭代方法在什么样的条件下收敛是至关重要的。

定理 3.1(迭代法基本定理) 设有方程组 $X=BX+f$, 对于任意初始向量 $X^{(0)}$ 及任意 f, 迭代公式 $X^{(k+1)}=BX^{(k)}+f$ 收敛的充要条件是迭代矩阵 B 的谱半径

$$\rho(B)<1.$$

证 设 X^* 为方程组 $X=BX+f$ 的准确解, 即

$$X^*=BX^*+f.$$

对任意初值 $X^{(0)}$ 和任意 f, 迭代公式为

$$X^{(k+1)}=BX^{(k)}+f,$$

于是

$$\begin{aligned} X^{(k)}-X^* &= BX^{(k-1)}+f-(BX^*+f)\\ &= B(X^{(k-1)}-X^*)\\ &= B^2(X^{(k-2)}-X^*)\\ &\vdots\\ &= B^k(X^{(0)}-X^*). \end{aligned}$$

根据第一章定理 1.7 的结论即可得证。

例 3.3 考察用迭代法解下列方程组

$$X^{(k+1)}=BX^{(k)}+f \tag{3.17}$$

的收敛性, 其中

$$\begin{bmatrix} 0 & \frac{3}{8} & -\frac{2}{8}\\ -\frac{4}{11} & 0 & \frac{1}{11}\\ -\frac{6}{12} & -\frac{3}{12} & 0 \end{bmatrix}.$$

解 计算 B 的特征值, 由

$$det(\lambda E-B)=\lambda^3+0.034090909\lambda+0.039772727=0$$

解得

$$\lambda_1 = -0.3082, \quad \lambda_2 = 0.1541 + 0.3245i, \quad \lambda_3 = 0.1541 - 0.3245i,$$

从而

$$|\lambda_1| = 0.3082 < 1, \quad |\lambda_2| = |\lambda_3| = 0.3592 < 1.$$

即 $\rho(B) < 1$, 故迭代收敛。定理 3.1 中的 $\rho(B) < 1$ 涉及到求矩阵的特征值, 一般很难验证。为此, 我们利用第一章中的定理 1.4 直接推得下列收敛性充分条件。

定理 3.2(迭代法收敛的一个充分条件) 设有迭代公式

$$X^{(k+1)} = BX^{(k)} + f,$$

如果 $\|B\| < 1$, 则对任意初始向量 $X^{(0)}$ 和任意 f, 这一迭代公式均是收敛的。

在实际问题中时常遇到这样一些方程组, 其系数矩阵是对角占优阵, 即设 $A = (a_{ij})_n$.

定义 3.1 如果

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

即主对角线上的元素绝对值大于同行其他元素的绝对值之和, 那么称矩阵 A 是对角占优阵。

定理 3.3(迭代法收敛的另一充分条件) 如果线性方程组 $AX = b$ 的系数矩阵 A 是对角占优阵, 则求解这一方程组的 Jacobi 迭代法和 Gauss-Seidel 迭代法均收敛。

例 3.4 对下列方程组

$$\begin{cases} 11x_1 - 3x_2 - 33x_3 = 1 \\ -22x_1 + 11x_2 + x_3 = 0 \\ x_1 - 4x_2 + 2x_3 = 1 \end{cases} \tag{3.18}$$

建立收敛的迭代公式。

解 通过观察可发现这一方程组的系数矩阵不是对角占优的。但经行交换后可得下列等价形式

$$\begin{cases} -22x_1 + 11x_2 + x_3 = 0 \\ x_1 - 4x_2 + 2x_3 = 1 \\ 11x_1 - 3x_2 - 33x_3 = 1 \end{cases} \tag{3.19}$$

方程组 (3.19) 的系数矩阵是对角占优阵, 据此建立的 Jacobi 迭代公式和 Gauss-Seidel 迭代公式是收敛的。

§3.5 Gauss 消元法

所谓 Gauss 消元法 (或叫消元法) 即顺序消元法, 是一个古老的求解线性方程组的直接方法, 其主要计算过程分为消元和回代两个步骤。

§3.5.1 消元过程

设有线性方程组

$$AX = b, \tag{3.20}$$

其中

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

$$X = (x_1, x_2, \cdots, x_n)^T,$$

$$b = (b_1, b_2, \cdots, b_n)^T.$$

且设 A 为 n 阶非奇异矩阵。

为讨论方便, 将式 (3.20) 记为

$$A^{(1)}X = b^{(1)},$$

其中

$$A^{(1)} = (a_{ij}^{(1)})_n = (a_{ij})_n = A, \quad b^{(1)} = b.$$

1. 假设 $a_{11}^{(1)} \neq 0$, 则先计算

$$m_{i1} = a_{i1}^{(1)}/a_{11}^{(1)} \quad (i = 2, 3, \cdots, n),$$

再用 m_{i1} 乘上方程组 (3.20) 的第 1 个方程后对应加到第 i 个方程 $(i = 2, 3, \cdots, n)$ 上, 即可消去第 2 个方程到第 n 个方程中的变量 x_1, 由此得下列等价方程组

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{bmatrix}, \tag{3.21}$$

简记为

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

其中

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

2. 假设 $a_{22}^{(2)} \neq 0$, 则分别计算

$$m_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}} \quad (i = 3,4,\cdots,n),$$

再用 m_{i2} 乘上方程 (3.21) 的第 2 个方程后对应加到第 i 个方程 $(i = 3,4,\cdots,n)$ 上, 则可消去第 3 个方程到第 n 个方程中的变量 x_2, 得下列等价方程组

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

简记为

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

其中

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

3. 重复上述步骤, 经 $n-1$ 次消元后, 得下列等价方程组

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(n)} \end{bmatrix}. \tag{3.22}$$

方程组 (3.22) 是一个与式 (3.20) 同解的上三角方程组, 其求解是十分容易的。

§3.5.2 回代过程

从方程组 (3.22) 的第 n 个方程开始, 从下往上依次解出变量 $x_n, x_{n-1}, \cdots, x_1$ 的这一过程称作回代过程, 具体计算公式是

$$x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}},$$

$$x_i = (b_i^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)}x_j)/a_{ii}^{(i)} \quad (i = n-1, n-2, \cdots, 1). \tag{3.23}$$

上述解法就是一般的 Gauss 消元法，或称沿系数矩阵主对角元素的顺序消元法。这种顺序消元法，在经过前 $k-1$ 次消元后，总是用系数矩阵中位于第 k 行第 k 列的元素 $a_{kk}^{(kk)}$ 作除数除以第 k 行的所有元素，然后依次将第 $k+1$ 行到第 n 行、第 k 列上的元素全部化为零。如果此时 $a_{kk}^{(kk)}$ 为零或者其绝对值很小，则消元过程无法进行下去，或者将严重地影响计算精度。这就是 Gauss 顺序消元法的一大缺陷。

§3.5.3 Gauss 选主元消元法

由上节讨论可知，在消元过程中可能出现 $a_{kk}^{(kk)}=0$ 的情形，致使消元过程无法继续下去。即使 $a_{kk}^{(kk)}\neq 0$ 但其绝对值很小时，用它作除数，可导致其他元素数量级的严重增长和舍入误差的扩散，使计算结果不可靠。

例 3.5 求解方程组

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

用 4 位有效数字计算。

解 为讨论问题方便，我们用下列增广矩阵的形式来表示方程组

$$(A\ \ b) = \begin{bmatrix} 0.001 & 2.000 & 3.000 & 1.000 \\ -1.000 & 3.712 & 4.623 & 2.000 \\ -2.000 & 1.070 & 5.643 & 3.000 \end{bmatrix} \begin{matrix} \\ m_{21}=-1000 \\ m_{31}=-2000 \end{matrix}$$

$$\to \begin{bmatrix} 0.001 & 2.000 & 3.000 & 1.000 \\ 0 & 2004 & 3005 & 1002 \\ 0 & 4001 & 6006 & 2003 \end{bmatrix} \quad m_{32}=1.997$$

$$\to \begin{bmatrix} 0.001 & 2.000 & 3.000 & 1.000 \\ 0 & 2004 & 3005 & 1002 \\ 0 & 0 & 5.000 & 2.000 \end{bmatrix},$$

回代后得计算解

$$X=(0.000,-0.09980,0.4000)^T.$$

而该方程组舍入到 4 位有效数字的准确解是

$$X^*=(-0.4904,-0.05104,0.3675)^T.$$

显然计算解是一个不可靠的解，误差太大。原因是在作消元时，用了小主元 0.001 作除数，致使其他元素的数量级大大地增加，在计算位数有限的情形下，会导致较大的舍入误差。如元素 3.712,4.623,1.072, 5.643 经第 1 次消元后分别变成

2004,3005,4001 和 6006，且小数点后的数全部被舍掉了，舍入误差的扩散将准确解淹没掉了。

为使计算结果可靠，在消元过程中应避免使用绝对值小的主元素 $a_{kk}^{(k)}$，以减少舍入误差对计算结果的影响。我们可以采用选主元素的方法来克服 Gauss 顺序消元法的这一缺点。Gauss 选主元法一般分为完全主元素消元法和列主元素消元法两种。

§3.5.4 完全选主元素消元法

设有方程组 $AX=b$，其中 $A=(a_{ij})_n$ 为非奇异矩阵，$X=(x_1,x_2,\cdots,x_n)^T$，$b=(b_1,b_2,\cdots,b_n)^T$，经过 k 次选主元消元后，设得下列等价方程组

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2k} & a_{2,k+1} & \cdots & a_{2n} \\ & & \ddots & \vdots & \vdots & & \vdots \\ & & & a_{kk} & a_{k,k+1} & \cdots & a_{kn} \\ & 0 & & & a_{k+1,k+1} & \cdots & a_{k+1,n} \\ & & & & \vdots & & \vdots \\ & & & & a_{n,k+1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_k^{(k)} \\ x_{k+1}^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \\ f_{k+1} \\ \vdots \\ f_n \end{bmatrix}, \tag{3.24}$$

其中向量 $(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^T$ 是向量 $X=(x_1,x_2,\cdots,x_n)^T$ 的重新排序。

在式 (3.24) 的系数矩阵的子块

$$\begin{bmatrix} a_{k+1,k+1} & \cdots & a_{k+1,n} \\ \vdots & & \vdots \\ a_{n,k+1} & \cdots & a_{nn} \end{bmatrix} \tag{3.25}$$

中找出绝对值最大的元素，作第 $k+1$ 次消元。假设

$$|a_{pq}|=\max_{k+1\leqslant i,j\leqslant n}|a_{ij}|,$$

于是第 $k+1$ 行与第 p 行互换，第 $k+1$ 列与第 q 列互换。同时，右端项 f_{k+1} 与 f_p 互换，变量 $x_{k+1}^{(k)}$ 与 $x_p^{(k)}$ 互换 (因为系数矩阵作了列交换，故各变量之间的位置也相应地发生了改变)。通过上述行交换和列交换 a_{pq} 移到子块 (3.25) 的左上角位置。以 a_{pq} 作除数，作第 $k+1$ 次消元后，将式 (3.24) 第 $k+2$ 行到第 n 行、第 $k+1$ 列上的元素全部变为零。

重复上述步骤，经 $n-1$ 次选完全主元消元后，则可将原方程组转化为一个上三角的同解方程组，再由回代过程求出各变量。注意，由于变量之间的位置在消元过程中 (列变换) 发生了改变，所以在计算过程中应记住各变量 $x_1,x_2,\cdots,x_n$ 的具体位置，最后按变量的原始顺序 $(x_1,x_2,\cdots,x_n)^T$ 输出计算结果。

§3.5.5 选列主元消元法

完全主元素消元法在计算过程中花费了大量的时间用于寻找主元。同时，各变量间的位置在消元过程中也可能会发生变化。下面介绍的列主元消元法仅按列选主元，在消元过程中只作行交换，节省了主元搜寻时间。

设有方程组

$$AX = b,$$

其中 $A = (a_{ij})_n$ 为非奇异矩阵，$X = (x_1, x_2, \cdots, x_n)^T, b = (b_1, b_2, \cdots, b_n)^T$，经过 k 次选列主元消元后，可得下列形式的等价方程组

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2k} & a_{2,k+1} & \cdots & a_{2n} \\ \vdots & \ddots & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & a_{kk} & a_{k,k+1} & \cdots & a_{kn} \\ 0 & 0 & 0 & 0 & a_{k+1,k+1} & \cdots & a_{k+1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & a_{n,k+1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \\ f_{k+1} \\ \vdots \\ f_n \end{bmatrix}, \quad (3.26)$$

在式 (3.26) 系数矩阵的第 $k+1$ 列、第 $k+1$ 行到第 n 行上，找出绝对值最大的元素。假设 $|a_{p,k+1}| = \max\limits_{k+1 \leqslant i \leqslant n} |a_{i,k+1}|$，于是交换第 $k+1$ 行与第 p 行，f_{k+1} 与 f_p 互换，主元 $a_{p,k+1}$ 移到位置 $(k+1, k+1)$ 上来。以 $a_{p,k+1}$ 作除数，作第 $k+1$ 次消元，将方程组 (3.26) 系数矩阵第 $k+1$ 列、第 $k+2$ 行到第 n 行上的全部元素变为零。

重复上述步骤，经 $n-1$ 次消元后，可得原方程的一个同解的上三角形式的方程组，通过回代即可求得方程组的解。

例 3.6 用列主元法求解例 3.5。

解

$$(A \quad b) \rightarrow \begin{bmatrix} -2.000 & 1.070 & 5.643 & 3.000 \\ -1.000 & 3.712 & 4.623 & 2.000 \\ 0.001 & 2.000 & 3.000 & 1.000 \end{bmatrix} \quad \begin{matrix} m_{21} = -0.5000 \\ m_{31} = -0.0005 \end{matrix}$$

$$\rightarrow \begin{bmatrix} -2.000 & 1.070 & 5.643 & 3.000 \\ 0 & 3.176 & 1.801 & 0.5000 \\ 0 & 2.001 & 3.003 & 1.002 \end{bmatrix} \quad m_{32} = 0.6300$$

$$\rightarrow \begin{bmatrix} -2.000 & 1.070 & 5.643 & 3.000 \\ 0 & 3.176 & 1.801 & 0.5000 \\ 0 & 0 & 1.868 & 0.6870 \end{bmatrix},$$

得计算解为

$$X = (-0.4900, -0.05113, 0.3678)^T.$$

显然这个解的误差较小 (参考例 3.5), 是一个可靠的近似解。

§3.5.6 其它应用

Gauss 消元法可用于计算行列式的值。根据行列式的性质，行列式的任一行 (列) 乘以同一个数后，再加到另一行 (列) 对应元素上，其行列式的值不变；任意交换两行 (或列) 的位置，其值反号；三角阵的行列式之值等于主对角线上元素的乘积。利用上述性质，通过消元法可计算得到行列式的值。

此外, Gauss 消元法也可用于逆矩阵计算。设 $A = (a_{ij})_n$ 是满秩矩阵，且令 $A^{-1} = X = (x_{ij})_n$. 因为

$$AA^{-1} = AX = E,$$

从而，由分块矩阵性质可知，计算 A^{-1} 的问题等价于求解下列 n 个线性方程组

$$A\begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad A\begin{bmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \cdots, A\begin{bmatrix} x_{1n} \\ x_{2n} \\ \vdots \\ x_{nn} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

求解上述方程组即可求得 A^{-1} 的 n 个列向量，也就求得 A^{-1}. 由于这 n 个方程组的系数矩阵相同，故可采用三角分解法进行计算，这样可节省计算工作量。

§3.6 三角分解法

Gauss 消元法是通过逐步消元过程，将方程组的系数矩阵 A 转变为一个上三角的矩阵。这实际上相当于用一系列初等矩阵左乘 A.

记 $A^{(1)} = A, b^{(1)} = b$, 作第 1 次消元 (设 $a_{11}^{(1)} \neq 0$), 相当于用一个下三角初等矩阵左乘 $A^{(1)}$，由此得 $A^{(2)}$，即

$$L_1 A^{(1)} = A^{(2)}, \quad L_1 b^{(1)} = b^{(2)},$$

其中

$$L_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & 0 \\ -m_{31} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ -m_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

作第 2 次消元 (设 $a_{22}^{(2)} \neq 0$), 相当于用一个下三角初等矩阵左乘 $A^{(2)}$, 由此得 $A^{(3)}$, 即

$$L_2 A^{(2)} = A^{(3)}, \quad L_2 b^{(2)} = b^{(3)},$$

其中

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & -m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & -m_{n2} & 0 & \cdots & 1 \end{bmatrix}.$$

作第 k 次消元 (设 $a_{kk}^{(k)} \neq 0$), 即有

$$L_k A^{(k)} = A^{(k+1)}, \quad L_k b^{(k)} = b^{(k+1)},$$

其中

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

重复上述过程 (假设 $a_{kk}^{(k)} \neq 0, k = 1, 2, \cdots, n-1$), 经 $n-1$ 次消元后, 得

$$\begin{aligned} L_{n-1} L_{n-2} \cdots L_2 L_1 A^{(1)} &= A^{(n)}, \\ L_{n-1} L_{n-2} \cdots L_2 L_1 b^{(1)} &= b^{(n)}. \end{aligned} \tag{3.27}$$

其中

$$A^{(1)} = A, \quad b^{(1)} = b.$$

由式 (3.27) 得

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

这里 $L_1^{-1}, L_2^{-1}, \cdots, L_{n-2}^{-1}, L_{n-1}^{-1}$ 仍分别为下三角矩阵, 它们的乘积也是一个下三角矩阵, 记作 U. 于是有

$$A = LU, \tag{3.28}$$

其中

$$L = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & 0 \\ m_{31} & m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & \cdots & 1 \end{bmatrix}.$$

由上述讨论可知，Gauss 消元法实质上产生了一个将系数矩阵 A 分解为上三角阵与下三角阵相乘的因式分解。从矩阵理论来讲，只要 A 的各顺序主子式不为零，则有 $A = LU$。即当

$$a_{11} \neq 0, \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0, \quad \cdots, \quad det(A) \neq 0,$$

则 A 可分解为一个单位下三角矩阵 L 和一个上三角矩阵 U，且这种分解是唯一的。

设有方程组 $AX = b$，并设 $A = LU$，于是

$$AX = LUX = b, \tag{3.29}$$

其中

$$L = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}.$$

令

$$UX = Y,$$

则由式 (3.29) 得

$$LY = b.$$

于是求解 $AX = b$ 的问题等价于求解两个方程组 $UX = Y$ 和 $LY = b$. 具体解法是：

1. 利用前推过程解方程组 $LY = b$，求出变量 $Y = (y_1, y_2, \cdots, y_n)^T$，计算公式是

$$y_i = b_i - \sum_{j=1}^{i-1} l_{ij}y_j \quad (i = 1, 2, \cdots, n). \tag{3.30}$$

2. 利用回代过程解方程组 $UX = Y$，求出变量 $X = (x_1, x_2, \cdots, x_n)^T$，计算公式是

$$x_i = (y_i - \sum_{j=i+1}^{n} u_{ij}x_j)/u_{ii}. \tag{3.31}$$

上述方法称作求解线性方程组的三角直接分解法。这种方法实际上是 Gauss 消元法的一种变形。实际计算时，L 和 U 的各元素可简单地由矩阵乘积的定义求得，下面的例子说明这一方法。

例 3.7 用三角分解法解

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

解 设系数矩阵作了如下三角分解

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

根据矩阵乘法可得

$$\begin{aligned} 1 \cdot u_{11} = 1 \quad &\Rightarrow \quad u_{11} = 1, \\ 1 \cdot u_{12} = 2 \quad &\Rightarrow \quad u_{12} = 2, \\ 1 \cdot u_{13} = 3 \quad &\Rightarrow \quad u_{13} = 3, \\ l_{21}u_{11} = 2 \quad &\Rightarrow \quad l_{21} = 2, \\ l_{31}u_{11} = 3 \quad &\Rightarrow \quad l_{31} = 3, \\ l_{21}u_{12} + u_{22} = 5 \quad &\Rightarrow \quad u_{22} = 1, \\ l_{21}u_{13} + u_{23} = 2 \quad &\Rightarrow \quad u_{23} = -4, \\ l_{31}u_{12} + l_{32}u_{22} = 1 \quad &\Rightarrow \quad l_{32} = -5, \\ l_{31}u_{13} + l_{32}u_{23} + u_{33} = 5 \quad &\Rightarrow \quad u_{33} = -24. \end{aligned}$$

注意上述分解时元素的求解顺序。于是原方程组可表为

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & -5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & -24 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 18 \\ 20 \end{bmatrix}.$$

求解

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & -5 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 18 \\ 20 \end{bmatrix}.$$

得

$$y = (14, -10, -72)^T.$$

求解

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & -24 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ -10 \\ -72 \end{bmatrix}.$$

得

$$x = (1, 2, 3)^T$$

为使三角分解能顺利进行下去，或者为避免较大的舍入误差，可将三角分解法与选主元法结合起来，这种方法称作选主元的三角分解法。(读者可参考有关书籍。)

§3.7 追赶法

在一些实际问题中，例如解常微分方程边值问题，解热传导方程以及船体数学放样中所建立的三次样条函数等，都会要求解如下形式的三对角方程组

$$\begin{bmatrix} b_1 & c_1 & & & & & \\ a_2 & b_2 & c_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_i & b_i & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}, \tag{3.32}$$

其系数矩阵是一种带状的稀疏矩阵，非零元素集中分布在主对角线及其相邻两条次对角线上。且系数矩阵为对角占优阵，即有下列关系式成立

$$\begin{aligned} &|b_1| > |c_1| \\ &|b_i| > |a_i| + |c_i|, \quad i = 2, 3, \cdots, n-1. \\ &|b_n| > |a_n| \end{aligned}$$

利用 Gauss 顺序消元法（只是在每次消元时先用主元除以主元所在行使得主元变成 1 后再进行消元），通过 $n-1$ 次消元后，可得式 (3.32) 的同解方程组

$$\begin{bmatrix} 1 & u_1 & & & \\ & 1 & u_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & u_{n-1} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{n-1} \\ q_n \end{bmatrix}, \tag{3.33}$$

其中

$$\begin{aligned} &u_1 = c_1/b_1, \quad q_1 = d_1/b_1, \\ &u_i = c_i/(b_i - u_{i-1}a_i) \quad (i = 2, \cdots, n-1), \\ &q_i = (d_i - x_{i-1}a_i)/(b_i - u_{i-1}a_i) \quad (i = 2, \cdots, n). \end{aligned} \tag{3.34}$$

利用回代过程依次求出方程组 (3.33) 各变量

$$\begin{aligned} &x_n = q_n, \\ &x_i = q_i - u_i x_{i+1} \quad (i = n-1, n-2, \cdots, 1). \end{aligned} \tag{3.35}$$

上述方法就是解三对角方程组的追赶方法。这里所谓的追 (消元过程)，指的就是按公式 (3.34) 顺序计算出 $u_1, u_2, \cdots, u_{n-1}$ 和 $q_1, q_2, \cdots, q_n$; 所谓的赶 (回代过程)，指的是按公式 (3.35) 逆序求出变量 $x_n, x_{n-1}, \cdots, x_1$.

在追赶法的计算过程中不会出现小主元，因此不会引起舍入误差的严重扩散。同时，由于系数矩阵中含有大量的零元素，实际计算时可将这些零元素撇开，从而大大地节省了计算量。

§3.8 误差分析

在方程组的建立过程中，由于各种因素的影响，方程组的系数往往含有误差 (或称作扰动)。在这种情形下，弄清楚方程组解的性态是十分重要的。

例 3.8 设有方程组

$$\begin{cases} x_1 + x_2 = 2 \\ x_1 + 1.0001x_2 = 2.0001 \end{cases}$$

其准确解为 $x_1 = 1, x_2 = 1$.

假定右端项发生了微小变化，比如

$$\begin{cases} x_1 + x_2 = 2 \\ x_1 + 1.0001x_2 = 2 \end{cases}$$

这时，准确解为 $x_1 = 2, x_2 = 0$.

通过上例可看到，虽然右端项仅发生了微小的改变，但引起方程组的解发生了很大的变化。这类方程组称为" 病态" 方程组。

设有方程组

$$AX = b \tag{3.36}$$

其中 A 为非奇异矩阵，X 为方程组 (3.36) 的准确解。

1. 设方程组 (3.36) 的右端项 b 有扰动 δb, 则相应的解为 $X+\delta X$, 即

$$A(X+\delta X)=b+\delta b,$$

于是

$$\delta X=A^{-1}\delta b.$$

则

$$\|\delta X\|\leqslant\|A^{-1}\|\ \ \|\delta b\|.$$

由 (3.36) 又有

$$\|b\|\leqslant\|A\|\ \ \|X\|,$$

所以

$$\frac{\|\delta X\|}{\|X\|}\leqslant\|A^{-1}\|\ \ \|A\|\frac{\|\delta b\|}{\|b\|}. \tag{3.37}$$

式 (3.37) 表明右端项的相对误差 $\frac{\|\delta b\|}{\|b\|}$ 在解中可能放大了 $\|A^{-1}\|\ \ \|A\|$ 倍。

2. 设方程组 (3.36) 的系数矩阵 A 有扰动 δA, 则相应的解为 $X+\delta X$, 即

$$(A+\delta A)(X+\delta X)=b,$$

于是

$$\delta X=-A^{-1}\delta A(X+\delta X).$$

则

$$\|\delta X\|=\|A^{-1}\|\ \ \|\delta A\|(\|X\|+\|\delta X\|).$$

假设 $\|\delta A\|$ 足够小, 并使下式成立

$$\|A^{-1}\|\ \ \|\delta A\|<1,$$

则有

$$\frac{\|\delta X\|}{\|X\|}\leqslant\frac{\|A^{-1}\|\ \ \|\delta A\|}{1-\|A^{-1}\|\ \ \|\delta A\|}=\frac{\|A^{-1}\|\ \ \|A\|\frac{\|\delta A\|}{\|A\|}}{1-\|A^{-1}\|\ \ \|A\|\frac{\|\delta A\|}{\|A\|}}. \tag{3.38}$$

式 (3.38) 表明, 当 $\|\delta A\|$ 充分小时, 矩阵 A 的相对误差在解中可能放大 $\|A^{-1}\|\ \ \|A\|$ 倍。

由上述讨论可知, 数 $\|A^{-1}\|\|A\|$ 反映了方程组的" 病态" 程度, 即刻画了解对原始数据扰动的灵敏程度。鉴此, 我们给出如下概念.

定义 3.2 称数 $\|A^{-1}\|\|A\|$ 为矩阵 A 的条件数, 并记为 $cond(A)$。

通常使用的条件数有

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

$$cond(A)_2 = \|A^{-1}\|_2 \|A\|_2 = \sqrt{\frac{\lambda_{max}(A^TA)}{\lambda_{min}(A^TA)}}.$$

特别, 当 A 为对称矩阵时

$$cond(A)_2 = \frac{|\lambda_{max}|}{|\lambda_{min}|},$$

这里 λ_{max} 与 λ_{min} 分别是对称阵 A 的特征值的最大模与最小模。

当 $cond(A) \gg 1$, 则方程组是 "病态" 的; 当 $cond(A)$ 较小时, 则方程组是 "良态" 的。

习题三

3.1 设有方程组

$$\begin{cases} 5x_1 + 2x_2 + x_3 = -12 \\ -x_1 + 4x_2 + 2x_3 = 20 \\ 2x_1 - 3x_2 + 10x_3 = 3 \end{cases}$$

(1) 考察用 Jacobi 迭代法, Gauss-Seidel 迭代法解此方程组的收敛性;

(2) 用 Jacobi 迭代法及 Gauss-Seidel 迭代法解此方程组, 要求当 $\|X^{(k+1)} - X^{(k)}\|_\infty < 10^{-4}$ 时迭代终止。

3.2 设有方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases} \quad (a_{11}, a_{22} \neq 0),$$

迭代公式

$$\begin{cases} x_1^{(k)} = \dfrac{1}{a_{11}}(b_1 - a_{12}x_2^{(k-1)}) \\ x_2^{(k)} = \dfrac{1}{a_{22}}(b_2 - a_{21}x_1^{(k-1)}) \end{cases} \quad (k = 1, 2, \cdots),$$

求证: 由上述迭代公式产生的向量序列 $\{X^{(k)}\}$ 收敛的充要条件是

$$\gamma = \left|\frac{a_{12}a_{21}}{a_{11}a_{22}}\right| < 1.$$

3.3 用 SOR 方法解下列方程组 (取松弛因子 $w = 1.2$)，要求 $\|X^{(k+1)} - X^{(k)}\|_\infty < 10^{-4}$.

$$\begin{cases} 2x_1 + x_2 = 1 \\ x_1 - 4x_2 = 5 \end{cases}$$

3.4 设

$$A = \begin{bmatrix} 1 & 10^4 \\ 1 & 1 \end{bmatrix},$$

计算 $cond(A)_\infty$.

3.5 用选列主元 Gauss 消元法求解方程组

$$\begin{cases} 3x_1 - x_2 + 4x_3 = 7 \\ -x_1 + 2x_2 - 2x_3 = -1 \\ 2x_1 - 3x_2 + 2x_3 = 0 \end{cases}$$

3.6 用追赶法解三对角方程组

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

3.7 用三角分解法解方程组

$$\begin{bmatrix} -2 & 4 & 8 \\ -4 & 18 & -16 \\ -6 & 2 & -20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 7 \end{bmatrix}.$$

3.8 用选主元消元法计算下列行列式的值

$$\begin{bmatrix} 1 & 2 & 6 \\ 3 & 2 & 4 \\ 9 & 5 & 1 \end{bmatrix}.$$

第四章　插值方法

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，为进一步分析问题的性质和变化规律，自然希望找到一种能近似描述函数 $f(x)$ 变化规律、又便于处理的简单函数 $P(x)$ 作为 $f(x)$ 的近似。

近似函数 $P(x)$ 选自不同类型的函数，则近似 $f(x)$ 的效果不同，这里 $P(x)$ 可以是一个代数多项式或三角多项式，也可以是有理分式；$P(x)$ 即可以是光滑函数也可以是分段光滑函数。由于代数多项式结构简单，便于计算和分析，实际上 $P(x)$ 常取作代数多项式。如果要求近似函数 $P(x)$ 取给定的离散数据，则称之为 $f(x)$ 的插值函数。

本章将主要研究代数插值多项式，同时介绍作为插值法和常微分方程数值解法的重要工具：差商与差分及其性质。

§4.1　多项式插值问题

设函数 $f(x)$ 在区间 $[a,b]$ 上有定义，且已知在点 $x_i \in [a,b]$ 上的函数值 $f(x_i)(i = p_0, p_1, \cdots, p_n)$ 和点 $x_j \in [a,b]$ 上的导数值 $f^{K_j}(x_j)(j = q_0, q_1, \cdots, q_m)$，其中 K_j 为小于或等于 $n+m+1$ 的任意正整数。求作一次数不超过 $n+m+1$ 的代数多项式

$$P(x) = a_0 + a_1 x + \cdots + a_{n+m+1} x^{n+m+1}$$

使

$$\begin{aligned} P(x_i) &= f(x_i)(i = p_0, p_1, \cdots, p_n), \\ P^{K_j} &= f^{K_j}(x_j)(j = q_0, q_1, \cdots, q_m) \end{aligned}$$

成立，则称 $P(x)$ 为 $f(x)$ 的插值函数，$x_i(i = p_0, p_1, \cdots, p_n)$ 和 $x_j(j = q_0, q_1, \cdots, q_m)$ 称之插值节点，区间 $[a,b]$ 称之插值区间。

上述问题称作为代数多项式插值问题，从计算的观点看，插值问题就是用一简单函数 $P(x)$ 在某种误差范围内近似的代替 $f(x)$，由于 $P(x)$ 的表达式简单，从而使插值问题有广泛的应用。例如当 $f(x)$ 的表达式很复杂或者是由一张函数表给出时，其对 $f(x)$ 的积分或微分可近似的认为

$$\int_a^b f(x)dx \approx \int_a^b P(x)dx, \qquad \frac{d}{dx}(f(x)) \approx \frac{d}{dx}(P(x)),$$

对于外形设计及微分方程的数值解，插值法也有重要应用。

例 4.1 已知函数 $f(x)$ 的如下数据

x	0	1	2
y	0	1	1
y'	0	1	

求 $f(x)$ 的插值多项式 $P(x)$.

解 由于已知函数 $f(x)$ 的 3 个点上的函数值和 2 个点上的导数值, 故可构造一个 4 次多项式 (含 5 个待定系数)

$$P(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4,$$

使

$$\begin{aligned}
&P(0) = a_0 = 0,\\
&P(1) = a_0 + a_1 + a_2 + a_3 + a_4 = 1,\\
&P(2) = a_0 + 2a_1 + 4a_2 + 8a_3 + 16a_4 = 1,\\
&P'(0) = a_1 = 0 = f'(0),\\
&P'(1) = a_1 + 2a_2 + 3a_3 + 4a_4 = 1.
\end{aligned}$$

联立上式得

$$a_0 = 0, a_1 = 0, a_2 = \frac{9}{4}, a_3 = -\frac{3}{2}, a_4 = \frac{1}{4}.$$

于是

$$P(x) = \frac{9}{4}x^2 - \frac{3}{2}x^3 + \frac{1}{4}x^4.$$

例 4.2 已知函数 $f(x)$ 在 x_0 处的函数值 $f(x_0)$ 和点 x_0 处的一阶到 n 阶导数值 $f^k(x_0)(k = 1, 2, \cdots, n)$, 试构造一插值多项式 $P(x)$.

解 设

$$P(x) = a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0)^n,$$

令

$$\begin{aligned}
P(x_0) &= a_0 = f(x_0),\\
P'(x_0) &= a_1 = f'(x_0),\\
P''(x_0) &= 2!a_2 = f''(x_0),\\
&\vdots\\
P^{(n)}(x_0) &= n!a_n = f^{(n)}(x_0),
\end{aligned}$$

联立上式得

$$a_0 = f(x_0), a_1 = f'(x_0), a_2 = \frac{1}{2!}f''(x_0), \cdots, a_n = \frac{1}{n!}f^{(n)}(x_0),$$

于是

$$P(x) = f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{1}{n!}f^{(n)}(x_0)(x - x_0)^n.$$

这个式子就是我们所熟悉的 Taylor 展开式的前 $n+1$ 项。

§4.2 Lagrange 插值公式

§4.2.1 Lagrange 插值及其存在唯一性

设函数 $y = f(x)$ 在区间 $[a, b]$ 上有定义，且已知在点 $a \leqslant x_0 < x_1 < x_2 < \cdots < x_n \leqslant b$ 上的函数值 $y_0, y_1, \cdots, y_n$，求一个次数不高于 n 的插值多项式

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

使

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

成立，则称式 (4.1) 为满足插值条件 (4.2) 的 Lagrange 插值。

根据条件 (4.2) 有

$$\begin{cases} a_0 + a_1 x_0 + \cdots + a_n x_0^n = y_0 \\ a_0 + a_1 x_1 + \cdots + a_n x_1^n = y_1 \\ \qquad\qquad \vdots \\ a_0 + a_1 x_n + \cdots + a_n x_n^n = y_0 \end{cases} \tag{4.3}$$

这是一个关于 $a_0, a_1, \cdots, a_n$ 的 $n+1$ 元线性方程组，其系数矩阵的行列式为

$$V_n(x_0, x_1, \cdots, x_n) = \begin{vmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{vmatrix}$$

这个行列式称作 Vandermonde 行列式，如果 $x_i \neq x_j (i \neq j)$，则

$$V_n(x_0, x_1, \cdots, x_n) \neq 0.$$

故方程组 (4.3) 有唯一解 $a_0, a_1, \cdots, a_n$. 于是满足条件 (4.2) 的插值多项式 (4.1) 存在且是唯一的。

§4.2.2 Lagrange 插值的基函数构造法

从上述讨论可知, 通过求解方程组 (4.3) 即可确定 Lagrange 插值多项式 $L_n(x)$ 。但这种作法的计算量大, 不便于实际应用。

首先讨论 $n = 1$ 的情形, 设已知函数 $f(x)$ 在区间 $[x_k, x_{k+1}]$ 端点处的函数值 $f(x_k)$ 和 $f(x_{k+1})$, 求一个线性插值多项式 $L_1(x)$ 使

$$\begin{aligned} &L_1(x_k) = y_k = f(x_k), \\ &L_1(x_{k+1}) = y_{k+1} = f(x_{k+1}) \end{aligned}$$

成立。

由点斜式公式很容易求得 $L_1(x)$ 的表达式

$$L_1(x) = y_k + \frac{y_{k+1} - y_k}{x_{k+1} - x_k}(x - x_k). \tag{4.4}$$

式 (4.4) 也可写成下列两点式的形式

$$L_1(x) = \frac{x - x_{k+1}}{x_k - x_{k+1}} y_k + \frac{x - x_k}{x_{k+1} - x_k} y_{k+1}. \tag{4.5}$$

由 (4.5) 看出, $L_1(x)$ 是两个线性函数

$$l_k(x) = \frac{x - x_{k+1}}{x_k - x_{k+1}}, \quad l_{k+1}(x) = \frac{x - x_k}{x_{k+1} - x_k}$$

的线性组合, 即

$$L_1(x) = y_k l_k(x) + y_{k+1} l_{k+1}(x).$$

这里 $l_k(x)$ 和 $l_{k+1}(x)$ 称作线性插值基函数, 并满足条件

$$\begin{aligned} &l_k(x_k) = 1, \quad l_k(x_{k+1}) = 0, \\ &l_{k+1}(x_{k+1}) = 1, \quad l_{k+1}(x_k) = 0 \end{aligned}$$

成立。

再讨论 $n = 2$ 的情形, 设已知函数 $f(x)$ 在点 x_{k-1}, x_k, x_{k+1} 上的函数值 $f(x_{k-1}), f(x_k), f(x_{k+1})$, 求一个二次插值（抛物插值）多项式 $L_2(x)$, 使

$$\begin{aligned} &L_2(x_{k-1}) = y_{k-1} = f(x_{k-1}), \\ &L_2(x_k) = y_k = f(x_k), \\ &L_2(x_{k+1}) = y_{k+1} = f(x_{k+1}) \end{aligned} \tag{4.6}$$

成立。

仿照线性插值的基函数构造方法，可令

$$L_2(x) = y_{k-1}l_{k-1}(x) + y_k l_k(x) + y_{k+1}l_{k+1}(x), \tag{4.7}$$

其中 $l_{k-1}(x), l_k(x), l_{k+1}(x)$ 是二次函数称作二次插值基函数，且满足

$$\begin{array}{lll} l_{k-1}(x_{k-1}) = 1, & l_k(x_{k-1}) = 0, & l_{k+1}(x_{k-1}) = 0, \\ l_{k-1}(x_k) = 0, & l_k(x_k) = 1, & l_{k+1}(x_k) = 0, \\ l_{k-1}(x_{k+1}) = 0, & l_k(x_{k+1}) = 0, & l_{k+1}(x_{k+1}) = 1 \end{array} \tag{4.8}$$

容易验证式 (4.7) 满足插值条件 (4.6) 。由前述 Lagrange 插值函数存在唯一性的讨论可知，式 (4.7) 就是所求的抛物插值多项式。

二次基函数 $l_{k-1}(x), l_k(x)$ 和 $l_{k+1}(x)$ 的求解是容易的，比如求 $l_k(x)$ ，根据条件 $l_k(x_{k-1}) = 0, l_k(x_{k+1}) = 0$ ，令

$$l_k(x) = C(x - x_{k-1})(x - x_{k+1}) \quad (C\text{为待定系数}),$$

再由条件 $l_k(x_k) = 1$，解得

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

于是

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

同理，得

$$\begin{aligned} l_{k-1}(x) &= \frac{(x - x_k)(x - x_{k+1})}{(x_{k-1} - x_k)(x_{k-1} - x_{k+1})}, \\ l_{k+1}(x) &= \frac{(x - x_{k-1})(x - x_k)}{(x_{k+1} - x_{k-1})(x_{k+1} - x_k)}, \end{aligned}$$

所以

$$\begin{aligned} L_2(x) = \quad & \frac{(x - x_k)(x - x_{k+1})}{(x_{k-1} - x_k)(x_{k-1} - x_{k+1})} y_{k-1} \\ & + \frac{(x - x_{k-1})(x - x_{k+1})}{(x_k - x_{k-1})(x_k - x_{k+1})} y_k \\ & + \frac{(x - x_{k-1})(x - x_k)}{(x_{k+1} - x_{k-1})(x_{k+1} - x_k)} y_{k+1}. \end{aligned} \tag{4.9}$$

例 4.3 利用 100, 121 的开方求 $\sqrt{115}$.

解 这里 $x_0 = 100, y_0 = 10, x_1 = 121, y_1 = 11$, 由式 (4.5) 得

$$L_1(x) = \frac{x - 121}{100 - 121} \times 10 + \frac{x - 100}{121 - 100} \times 11,$$

于是

$$\begin{aligned}\sqrt{115} \approx L_1(115) &= \frac{115-121}{100-121} \times 10 + \frac{115-100}{121-100} \times 11 \\ &= 10.71428.\end{aligned}$$

$\sqrt{115}$ 的准确值是 $10.723805\cdots$，因此近似值 10.71428 有 3 位有效数字。

如果对线性插值的精度不满意，则可增加一个插值节点 $x_2 = 144(y_2 = 12)$，按式 (4.9) 构造抛物插值 $L_2(x)$．于是

$$\sqrt{115} \approx L_2(115) = 10.7228,$$

这个近似值具有 4 位有效数字。

现在讨论一般情形下的基函数的构造方法。

设已知函数 $f(x)$ 在 $n+1$ 个插值节点 $a \leqslant x_0 < x_1 < x_2 < \cdots < x_n \leqslant b$ 上的函数值 $f(x_0), f(x_1), \cdots, f(x_n)$，求一次数不超过 n 的插值多项式 $L_n(x)$，使

$$L_n(x_i) = y_i = f(x_i) \quad (i = 0, 1, 2, \cdots, n) \tag{4.10}$$

成立。

仿照前面线性插值和抛物插值的基函数构造方法，为此令

$$L_n(x) = l_0(x)y_0 + l_1(x)y_1 + \cdots + l_n(x)y_n,$$

其中 $l_i(x)(i = 0, 1, \cdots, n)$ 为 n 次多项式，称作 n 次插值基函数，且满足

$$l_i(x_j) = \begin{cases} 1, & x_j = x_i \\ 0, & x_j \neq x_i \end{cases} \qquad (i = 0, 1, 2, \cdots, n) \tag{4.11}$$

据条件 (4.11) 容易求得

$$\begin{aligned} l_i(x) &= \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} \\ &= \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}. \end{aligned}$$

于是

$$L_n(x) = \sum_{i=0}^{n} y_i l_i(x) = \sum_{i=0}^{n} \Big(\prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j} \Big) y_i. \tag{4.12}$$

§4.2.3 插值余项

插值方法是在区间 $[a,b]$ 上用 $y=L_n(x)$ 近似 $f(x)$，在插值节点 x_i 上有 $f(x_i)=L_n(x_i)$，而在 $[a,b]$ 上的其它点 $x(x\neq x_i,i=0,1,\cdots,n)$ 上就会有误差，令

$$R_n(x)=f(x)-L_n(x),$$

$R_n(x)$ 称作插值多项式的余项，它表示用 $L_n(x)$ 近似 $f(x)$ 的截断误差的大小。

据插值条件 (4.10) 可知

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

于是令

$$R_n(x)=k(x)(x-x_0)(x-x_1)\cdots(x-x_n)=k(x)\omega_{n+1}(x),$$

这里 $\omega_{n+1}(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$，插值节点 $x_i(i=0,1,2,\cdots,n)$ 都是 $\omega_{n+1}(x)$ 的零点。

作辅助函数

$$\varphi(t)=f(t)-L_n(t)-k(x)(t-x_0)(t-x_1)\cdots(t-x_n),$$

其中 x 为 $[a,b]$ 上的某一固定点，显然 $\varphi(t)$ 在点 $x_0,x_1,x_2,\cdots,x_n$ 和 x 上均为零，故 $\varphi(t)$ 在 $[a,b]$ 上有 $n+2$ 个零点，据 Rolle 定理，$\varphi'(t)$ 在 $\varphi(t)$ 的两个零点之间至少有一个零点，故 $\varphi'(t)$ 在 $[a,b]$ 上至少有 $n+1$ 个零点，依次类推 $\varphi^{(n+1)}(t)$ 在 (a,b) 内至少有一个零点 ξ，使

$$\varphi^{(n+1)}(\xi)=f^{(n+1)}(\xi)-(n+1)!k(x)=0,$$

于是

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

其中 $\xi\in(a,b)$，且依赖于 x。所以

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x), \tag{4.13}$$

记 $M=\max\limits_{a\leqslant x\leqslant b}|f^{(n+1)}(x)|$，则

$$R_n(x)\leqslant\frac{M}{(n+1)!}|\omega_{n+1}(x)|.$$

注意，我们在上述余项表达式的推导中默认 $f(x)$ 的高阶导数是存在的，因此余项表达式 (4.13) 只有在肯定 $f(x)$ 的高阶导数是存在的方可使用，同时，由于余项中含有因子

$$\omega_{n+1}(x) = (x - x_0)(x - x_1)\cdots(x - x_n)$$

如果被插值点 x 离插值节点 x_i 较远，则插值误差较大，当 x 位于插值区间 $[a, b]$ 以外（称作外插，反之为内插）一般会引起较大的误差，因此外插的效果是不理想的，实际使用中应尽量避免外插。

例 4.4 设 $f(x) = e^{-x}$ 在 $x_0 = 0.10, x_1 = 0.15, x_2 = 0.25, x_3 = 0.30$ 的值分别是 $0.904837, 0.860708, 0.778801, 0.740818$，试构造 Lagrange 插值多项式，求 e^{-x} 在 $x = 0.20$ 的近似值并估计误差。

解 依题意，取 $n = 3$

$$L_3(x) = l_0(x)y_0 + l_1(x)y_1 + l_2(x)y_2 + l_3(x)y_3,$$

于是

$$\begin{aligned} L_3(0.20) &= 0.904837l_0(0.20) + 0.860708l_1(0.20) + 0.778801l_2(0.20) + 0.740818l_3(0.20) \\ &= 0.904837 \times (-\frac{1}{6}) + 0.860708 \times \frac{2}{3} + 0.778801 \times \frac{2}{3} + 0.740818 \times (-\frac{1}{6}) \\ &= 0.818730, \end{aligned}$$

由式 (4.13)

$$|R_3(0.20)| = \left|\frac{f^{(4)}(\xi)}{4!}(0.20 - 0.10)(0.20 - 0.15)(0.20 - 0.25)(0.20 - 0.30)\right|$$

由于

$$f(x) = e^{-x},$$

故

$$|f^{(4)}(\xi)| \leqslant \max_{0.1 \leqslant x \leqslant 0.3} |e^{-x}| \leqslant 0.91,$$

所以

$$\begin{aligned} |R_3(0.20)| &= \frac{0.91}{4!}|(0.10)(0.05)(-0.05)(-0.10)| \\ &< 0.95 \times 10^{-6}. \end{aligned}$$

实际上 $e^{-0.20} = 0.8187307\cdots$，误差估计与实际计算结果相符。

§4.3 差商与差分

定义 4.1 称

$$f(x_0,x_1):=\frac{f(x_1)-f(x_0)}{x_1-x_0}$$

为函数 $f(x)$ 的一阶差商; 称

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

为 $f(x)$ 的二阶差商; 一般称

$$f(x_0,x_1,\cdots,x_n)=\frac{f(x_1,\cdots,x_n)-f(x_0,\cdots,x_{n-1})}{x_n-x_0}$$

为 $f(x)$ 的 n 阶差商。此外, 为统一起见, 补充定义 $f(x_0)$ 为 $f(x)$ 的零阶差商。

由差商定义, 显然有

$$\begin{aligned}f(x_0,x_1)&=\frac{f(x_0)}{x_0-x_1}+\frac{f(x_1)}{x_1-x_0}\\ f(x_0,x_1,x_2)&=\frac{f(x_1,x_2)-f(x_0,x_1)}{x_2-x_0}\\ &=\frac{f(x_0)}{(x_0-x_1)(x_0-x_2)}+\frac{f(x_1)}{(x_1-x_0)(x_1-x_2)}\\ &\quad+\frac{f(x_2)}{(x_2-x_0)(x_2-x_1)},\end{aligned}$$

如此类推, 可以证明

$$\begin{aligned}&f(x_0,x_1,\cdots,x_n)\\ =&\sum_{j=0}^{n}\frac{f(x_j)}{(x_j-x_0)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)}.\end{aligned}$$

由此看出, 差商的值与节点 $x_0,x_1,x_2,\cdots,x_n$ 的排列次序无关, 即

$$f(x_0,x_1,\cdots,x_n)=f(x_1,x_0,\cdots,x_n)=\cdots=f(x_1,x_2,\cdots,x_n,x_0).$$

这种性质称为差商对称性。

定义 4.1 设函数 $y=f(x)$ 在等距节点 $x_i=x_0+ih(i=0,1,\cdots,n)$ 上的函数值为 $f(x_i)=f_i$, 其中 h 为常数, 称作步长。称

$$\begin{aligned}\triangle f_i&=f_{i+1}-f_i,\\ \nabla f_i&=f_i-f_{i+1},\\ \delta f_i&=f(x_i+h/2)-f(x_i-h/2)=f_{i+\frac{1}{2}}-f_{i-\frac{1}{2}}\end{aligned}$$

分别为 $f(x)$ 在 x_i 处以 h 为步长的一阶向前差分，一阶向后差分和一阶中心差分。符号 $\triangle$ 、∇、δ 分别称为向前差分算子，向后差分算子和中心差分算子。

由一阶差分的定义出发，可定义二阶差分

$$\begin{aligned}\triangle^2 f_i &= \triangle f_{i+1} - \triangle f_i = f_{i+2} - 2f_{i+1} + f_i,\\ \nabla^2 f_i &= \nabla f_i - \nabla f_{i+1} = f_i - 2f_{i-1} + f_{i-2},\\ \delta^2 f_i &= \delta f_{i+\frac{1}{2}} - \delta f_{i-\frac{1}{2}} = f_{i+1} - 2f_i + f_{i-1}\end{aligned}$$

一般地可定义 n 阶差分为

$$\begin{aligned}\triangle^n f_i &= \triangle^{n-1} f_{i+1} - \triangle^{n-1} f_i,\\ \nabla^n f_i &= \nabla^{n-1} f_i - \nabla^{n-1} f_{i+1},\\ \delta^n f_i &= \delta^{n-1} f_{i+\frac{1}{2}} - \delta^{n-1} f_{i-\frac{1}{2}}\end{aligned}$$

如果令 $If_i = f_i$，$Ef_i = f_{i+1}$ 则称 I 为不变算子，E 为移位算子，由于

$$\triangle f_i = f_{i+1} - f_i = Ef_i - If_i = (E - I)f_i,$$

于是

$$\triangle = E - I.$$

同理可得

$$\begin{aligned}\nabla &= I - E^{-1},\\ \delta &= E^{1/2} - E^{-1/2}.\end{aligned}$$

例 4.5 证明函数值与差分可互相线性表示。

证 因为

$$\begin{aligned} f_{i+n} = \; & E^n f_i = (I + \triangle)^n f_i = \Big[\sum_{j=0}^{n} \binom{n}{j} \triangle^j\Big] f_i \\ = \; & \sum_{j=0}^{n} \binom{n}{j} \triangle^j f_i. \end{aligned} \tag{4.14}$$

另一方面

$$\begin{aligned} \triangle^n f_i = \; & (E - I)^n f_i = \sum_{j=0}^{n} (-1)^j \binom{n}{j} E^{n-j} f_i \\ = \; & \sum_{j=0}^{n} (-1)^j \binom{n}{j} f_{n+i-j}. \end{aligned} \tag{4.15}$$

例 4.5 证明当节点 x_i 是等距离 $(x_i = x_0 + ih, y_i = f(x_i))$ 时，差分与差商存在着关系

$$f(x_0, x_1, \cdots, x_n) = \frac{\triangle^n y_0}{n! h^n}. \tag{4.16}$$

证 事实上，当 $n = 1$ 时有

$$f(x_0, x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{y_1 - y_0}{h} = \frac{\triangle y_0}{h}.$$

现假设 $n = m - 1$ 时式 (4.16) 成立，则有

$$f(x_1, x_2, \cdots, x_m) = \frac{\triangle^{m-1} y_0}{(m-1)! h^{m-1}},$$
$$f(x_0, x_1, \cdots, x_{m-1}) = \frac{\triangle^{m-1} y_1}{(m-1)! h^{m-1}}.$$

于是

$$\begin{aligned} f(x_0, x_1, \cdots, x_m) &= \frac{f(x_1, \cdots, x_m) - f(x_0, \cdots, x_{m-1})}{x_m - x_0} \\ &= \left[\frac{\triangle^{m-1} y_1}{(m-1)! h^{m-1}} - \frac{\triangle^{m-1} y_0}{(m-1)! h^{m-1}}\right] / mh \\ &= \frac{1}{m! h^m}(\triangle^{m-1} y_1 - \triangle^{m-1} y_0) = \frac{\triangle^m y_0}{m! h^m} \end{aligned} \tag{4.17}$$

证毕。

同理可证明

$$f(x_0, x_1, \cdots, x_n) = \frac{\nabla^n y_n}{n! h^n}. \tag{4.18}$$

§4.4 Newton 插值公式

有了差商的概念，前面介绍的线性插值与抛物插值可表示为

$$N_1(x) = f(x_k) + f(x_k, x_{k+1})(x - x_k),$$
$$N_2(x) = f(x_{k-1}) + f(x_{k-1}, x_k)(x - x_{k-1}) + f(x_{k-1}, x_k, x_{k+1})(x - x_{k-1})(x - x_k),$$

事实上，按差商定义

$$f(x) = f(x_0) + f(x_0, x)(x - x_0),$$
$$f(x_0, x) = f(x_0, x_1) + f(x_0, x_1, x)(x - x_1),$$
$$f(x_0, x_1, x) = f(x_0, x_1, x_2) + f(x_0, x_1, x_2, x)(x - x_2)$$
$$\vdots$$
$$f(x_0, x_1, \cdots, x_{n-1}, x) = f(x_0, x_1, \cdots, x_n) + f(x_0, x_1, \cdots, x_n, x)(x - x_n),$$

反复用后一个式子代入前面的式子，可得

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

记

$$\begin{aligned} N_n(x) &= f(x_0) + f(x_0, x_1)(x - x_0) \\ &\quad \cdots + f(x_0, x_1, \cdots, x_n)(x - x_0)(x - x_1)\cdots(x - x_{n-1}), \qquad (4.19) \\ R_n(x) &= f(x_0, x_1, \cdots, x_n, x)(x - x_0)(x - x_1)\cdots(x - x_n). \qquad (4.20) \end{aligned}$$

于是

$$f(x) = N_n(x) + R_n(x).$$

由于 $R_n(x_i) = 0 (i = 0, 1, 2, \cdots, n)$，故必有

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

所以式 (4.19) 为 n 次插值多项式，由插值多项式的唯一性讨论（见 §4.2.1）可知

$$N_n(x) \equiv L_n(x),$$

且有

$$f(x_0, x_1, \cdots, x_n, x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

式 (4.19) 称作 Newton 基本插值公式，它的各项系数就是函数的各阶差商，每增加一个插值节点，只需在原来的基础上多计算一项，这一性质称作 Newton 插值公式的承袭性。

例 4.7 已知函数 $f(x) = shx$ 的函数表如下，构造 4 次 Newton 插值多项式并计算 $f(0.596) = sh0.596$ 的值。

k	0	1	2	3	4	5
x_k	0.40	0.55	0.65	0.80	0.90	1.05
$f(x_k)$	0.41075	0.57815	0.69675	0.88811	1.02652	1.25386

解 首先根据给定函数表造出差商表

k	x_k	$f(x_k)$	$f(x_0,x_k)$	$f(x_0,x_1,x_k)$	$f(x_0,x_1,x_2,x_k)$	$f(x_0,x_1,x_2,x_3,x_k)$
0	0.40	0.41075				
1	0.55	0.57815	1.11600			
2	0.65	0.69675	1.14400	0.28000		
3	0.80	0.88811	1.19340	0.30960	0.19733	
4	0.90	1.02652	1.23154	0.33011	0.20044	0.03110
5	1.05	1.25382	1.29703	0.36206	0.20515	0.03128

从差商表求得 4 次插值项式

$$\begin{aligned} N_4(x) =& 0.41075 \\ &+ 1.116(x-0.4) \\ &+ 0.28000(x-0.4)(x-0.55) \\ &+ 0.19733(x-0.4)(x-0.55)(x-0.65) \\ &+ 0.03110(x-0.4)(x-0.55)(x-0.65)(x-0.8), \end{aligned}$$

于是

$$sh(0.596) \approx N_4(0.596) = 0.63192$$

如果插值节点为等距节点 $x_i = x_0 + ih, (i = 0, 1, \cdots, n)$，并要计算 x_0 附近点 $x(x_0 < x < x_1)$ 的函数值，令

$$x = x_0 + th, \quad 0 < t < 1,$$

于是

$$\omega_{i+1}(x) = \prod_{j=0}^{i}(x - x_j) = t(t-1)\cdots(t-i)h^{i+1}.$$

将式 (4.16) 代入式 (4.19) 中，得

$$\begin{aligned} N_n(x_0+th) &= y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \cdots \\ &\quad + \frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n y_0. \end{aligned} \tag{4.21}$$

公式 (4.21) 称作 Newton 前插公式，其余项由 (4.13) 得

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

如果要计算 x_n 附近点 $x(x_{n-1} < x < x_n)$ 的函数值，可令 $x = x_n + th(-1 < t < 0)$，插值节点按 $x_n, x_{n-1}, \cdots, x_0$ 的次序排列，于是

$$\begin{aligned} N_n(x) =& f(x_n) + f(x_n, x_{n-1})(x - x_n) + \\ & \cdots + f(x_n, x_{n-1}, \cdots, x_0)(x - x_n)\cdots(x - x_1), \end{aligned}$$

将式 (4.18) 代入上式，得

$$\begin{aligned} N_n(x_n + th) &= y_n + t\nabla y_n + \frac{t(t+1)}{2!}\nabla^2 y_n + \cdots \\ &\quad + \frac{t(t+1)\cdots(t+n-1)}{n!}\nabla^n y_n. \end{aligned} \tag{4.22}$$

公式 (4.22) 称作 Newton 后插公式，其余项是

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

§4.5 分段插值公式

实际应用中，高次插值（例如七、八次以上）很少被采用。这是因为高次插值的逼近效果往往是不理想的。节点的增多固然使插值函数 $P(x)$ 在更多地方与 $f(x)$ 相等，但是另一方面在两个插值节点间 $P(x)$ 不一定能更好地逼近 $f(x)$，有时差异很大。

例 4.8 考察函数

$$f(x) = \frac{1}{1 + 25x^2}, \qquad -1 \leqslant x \leqslant 1.$$

设将区间 $[-1, 1]$ 分为 n 等份，以 $P_n(x)$ 表示取 $n+1$ 个等份点作节点的插值多项式。图 4.1 给出了 $P_{10}(x)$ 的图象。从图中可看出，虽然 $P_{10}(x)$ 在 11 个插值节点上取与所逼近函数 $f(x)$ 相同的值，但整体逼近效果是很差的，越靠近端点逼近的效果就越差。这种现象称为 Runge 现象。

图 4.1

Runge 现象说明，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能很好地逼近 $f(x)$。如果在每个小区间 $[x_k, x_{k+1}]$ 内应用低次插值，例如一次插值或二次插值，则可避免 Runge 现象的发生。容易明白，如果每个小区间的两个端点取为插值节点的话，那么相邻区间的两个插值函数在节点处将保持连续，也就是说，$P(x)$ 的全体是一个连续函数，但在节点处一阶二阶导数并不一定连续。实践证明，用一分段光滑的低次多项式去逼近 $f(x)$ 的效果要优于用一任意光滑的高次多项式去逼近 $f(x)$。

§4.5.1 分段线性插值

分段线性插值从几何上看，就是在每一个小区间 $[x_k, x_{k+1}](k = 0, 1, \cdots, n)$ 上作连续插值点 (x_k, y_k) 与 (x_{k+1}, y_{k+1}) 的直线。记 $S_1(x)$ 为分段线性插值函数，则 $S_1(x)$ 显然满足下列条件。

1. $S_1(x)$ 在每个小区间 $[x_k, x_{k+1}]$ 上是线性函数，
2. $S_1(x_i) = y_i(i = 0, 1, \cdots, n)$,
3. $S_1(x)$ 在插值区间 $[a, b]$ 上为连续函数。

函数 $S_1(x)$ 在区间 $[x_{k-1}, x_k]$ 上的表达式是

$$\begin{aligned} S_1(x) &= \frac{x - x_k}{x_{k-1} - x_k} y_{k-1} + \frac{x - x_{k-1}}{x_k - x_{k-1}} y_k \\ &= l_{k-1}(x) y_{k-1} + l_k(x) y_k \quad (x_{k-1} \leqslant x \leqslant x_k), \end{aligned}$$

在区间 $[x_k, x_{k+1}]$ 上的表达式是

$$\begin{aligned} S_1(x) &= \frac{x - x_{k+1}}{x_k - x_{k+1}} y_k + \frac{x - x_k}{x_{k+1} - x_k} y_{k+1} \\ &= l_k(x) y_k + l_{k+1}(x) y_{k+1} \quad (x_k \leqslant x \leqslant x_{k+1}). \end{aligned}$$

如果用插值基函数表示表示，则 $S_1(x)$ 在插值区间 $[a, b]$ 上的表达式是

$$S_1(x) = \sum_{k=0}^{n} l_k(x) y_k \tag{4.23}$$

其中

$$l_k(x) = \begin{cases} \dfrac{x - x_{k-1}}{x_k - x_{k-1}}, & x_{k-1} \leqslant x \leqslant x_k (k = 0\text{略去}) \\ \dfrac{x - x_{k+1}}{x_k - x_{k+1}}, & x_k \leqslant x \leqslant x_{k+1} (k = n\text{略去}) \\ 0, & x \in [a, b], x \overline{\in} [x_k, x_{k+1}] \end{cases} \tag{4.24}$$

分段线性插值基函数 $l_k(x)$ 只在 x_k 附近不为零，在其它地方均为零，这种性质称为局部非零性质。当插值点有误差时，这种局部非零性质将误差控制在一个局部区域内。

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。

§4.5.2 分段三次 Hermite 插值

不少实际问题不但要求插值函数 $P(x)$ 在节点处的值与 $f(x)$ 在相应节点的值相等，而且要求 $P(x)$、$f(x)$ 在节点处的导数值也相等，这就导致下面的 Hermite 插值。

设函数 $f(x)$ 在 $n+1$ 个互异节点 $x_i(i=0,1,\cdots,n)$ 上的函数值和导数值为

$$y_0, y_1, \cdots, y_n; \quad y_0', y_1', \cdots, y_n'.$$

由于已知每个小区间 $[x_k, x_{k+1}]$ 两端点处的函数值 y_k, y_{k+1} 和 y_k', y_{k+1}'，故可在 $[x_k, x_{k+1}]$ 上构造三次插值多项式

$$P_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3,$$

使

$$\begin{aligned} &P_3(x_k) = y_k, \quad P_3'(x_k) = y_k', \\ &P_3(x_{k+1}) = y_{k+1}, \quad P_3'(x_{k+1}) = y_{k+1}'. \end{aligned}$$

记 $S_3(x)$ 为插值区间 $[a,b]$ 上的分段三次插值函数，则 $S_3(x)$ 满足

1. $S_3(x)$ 在每个小区间 $[x_k, x_{k+1}]$ 上是三次多项式，
2. $S_3(x_i) = y_i, S_3'(x_i) = y_i'(i=0,1,\cdots,n)$,
3. $S_3(x)$ 在插值区间 $[a,b]$ 上为一阶导数连续的函数。

仿照 Lagrange 插值多项式基函数构造法，可令分段三次插值函数 $S_3(x)$ 在 $[a,b]$ 上的表达式为

$$S_3(x) = \sum_{k=0}^{n} [y_k \alpha_k(x) + y_k' \beta_k(x)]. \tag{4.25}$$

在区间 $[x_k, x_{k+1}]$ 上 $S_3(x)$ 的表达式是

$$S_3(x) = y_k \alpha_k(x) + y_{k+1} \alpha_{k+1}(x) + y_k' \beta_k(x) + y_{k+1}' \beta_{k+1}(x). \tag{4.26}$$

并满足

$$S_3(x_k)=y_k,\quad S_3(x_{k+1})=y_{k+1},$$
$$S_3'(x_k)=y_k',\quad S_3'(x_{k+1})=y_{k+1}'.$$

由此可见，$\alpha_k(x),\alpha_{k+1}(x),\beta_k(x),\beta_{k+1}(x)$ 为满足下列条件的三次式

$$\begin{array}{cccc}\alpha_k(x_k)=1 & \alpha_{k+1}(x_k)=0 & \beta_k(x_k)=0 & \beta_{k+1}(x_k)=0\\ \alpha_k(x_{k+1})=0 & \alpha_{k+1}(x_{k+1})=1 & \beta_k(x_{k+1})=0 & \beta_{k+1}(x_{k+1})=0\\ \alpha_k'(x_k)=0 & \alpha_{k+1}'(x_k)=0 & \beta_k'(x_k)=1 & \beta_{k+1}'(x_k)=0\\ \alpha_k'(x_{k+1})=0 & \alpha_{k+1}'(x_{k+1})=0 & \beta_k'(x_{k+1})=0 & \beta_{k+1}'(x_{k+1})=1\end{array}$$

根据上述条件可分别求得 $\alpha_k(x),\alpha_{k+1}(x),\beta_k(x)$ 和 $\beta_{k+1}(x)$ 。比如求 $\alpha_k(x)$ ，由于

$$\alpha_k(x_{k+1})=\alpha_k'(x_{k+1})=0,$$

于是令

$$\alpha_k(x)=(ax+b)(x-x_{k+1})^2,$$

利用条件 $\alpha_k(x_k)=1$ 和 $\alpha_k'(x_k)=0$ 求得待定系数 a,b，于是得

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

同理，可求得 $S_3(x)$ 在 $[x_{k-1},x_k]$ 上的表达式。于是

$$\alpha_k(x)=\begin{cases}(\frac{x-x_{k-1}}{x_k-x_{k-1}})^2(1+2\frac{x-x_k}{x_{k-1}-x_k})^2, & x_{k-1}\leqslant x\leqslant x_k(k=0\text{略去})\\ (\frac{x-x_{k+1}}{x_k-x_{k+1}})^2(1+2\frac{x-x_k}{x_{k+1}-x_k})^2, & x_k\leqslant x\leqslant x_{k+1}(k=n\text{略去})\\ 0, & x\in[a,b],x\overline{\in}[x_k,x_{k+1}]\end{cases}\tag{4.27}$$

$$\beta_k(x)=\begin{cases}(\frac{x-x_{k-1}}{x_k-x_{k-1}})^2(x-x_k), & x_{k-1}\leqslant x\leqslant x_k(k=0\text{略去})\\ (\frac{x-x_{k+1}}{x_k-x_{k+1}})^2(x-x_k), & x_k\leqslant x\leqslant x_{k+1}(k=n\text{略去})\\ 0, & x\in[a,b],x\overline{\in}[x_k,x_{k+1}]\end{cases}\tag{4.28}$$

分段三次 (Hermite) 插值函数具有一阶光滑度，与分段线性插值相比，算法复杂了一些，但却改善了精度。

§4.6 三次样条插值

我们知道，在插值区间上作高次插值多项式，可以保证曲线的光滑性，但会出现较大的计算误差。分段低次插值虽可避免龙格现象，但光滑性较差，这往往不满

足某些工程设计上的要求，对于飞机的机翼型线，船体放样型值线等往往要求有二阶光滑度，即有二阶连续导数。这就导致了所谓三次样条插值的提出。

样条的概念来源于生产实践，“样条”是绘制曲线的一种绘图工具，它是富有弹性的细长条，绘图时用压铁使样条通过指定的型值点（样点），并调整样条使其具有满意的形状，然后沿样条画出曲线。这种曲线称作样条曲线。它实际上是由分段三次曲线“装配”起来的，在型值点处具有二阶连续系数。从数学上加以抽象就得到三次样条函数这一概念。

定义 4.3 设函数 $S(x)$ 在区间 $[a,b]$ 上的三次样条函数，$a=x_0<x_1<\cdots<x_n=b$ 是给定节点，则 $S(x)$ 满足

1. $S(x)$ 在每个小区间 $[x_k,x_{k+1}]$ 上是三次多项式，
2. $S(x_i)=y_i(i=0,1,\cdots,n)$，
3. $S(x)$ 在插值区间 $[a,b]$ 上为二阶导数连续的函数。

由于 $S(x)$ 在每个小区间上为三次多项式，故总计有 $4n$ 个待定系数，但根据 $S(x)$ 在 $[a,b]$ 上二阶导数连续，那么在节点 $x_i(i=0,1,\cdots,n-1)$ 处应满足连续性条件

$$\begin{cases} S(x_i-0)=S(x_i+0) \\ S'(x_i-0)=S'(x_i+0) \\ S''(x_i-0)=S''(x_i+0) \quad (i=0,1,\cdots,n-1) \end{cases}$$

共有 $3n-3$ 个条件，加上插值条件 $S(x_i)=y_i(i=0,1,\cdots,n)$ 共有 $4n-2$ 个条件，因此还差 2 个条件就可确定 $S(x)$ 。通常在区间 $[a,b]$ 两个端点上各加一个边界条件，视具体情况而定。

§4.6.1 三转角方程

假定已知 $S(x)$ 在节点 $x_i(i=0,1,\cdots,n)$ 处的一阶导数值，即设

$$S'(x_i)=m_i \quad (i=0,1,\cdots,n),$$

由分段三次 Hermite 插值 (4.25) 得

$$S(x)=\sum_{i=0}^{n}[y_i\alpha_i(x)+m_i\beta_i(x)], \tag{4.29}$$

其中 $\alpha_i(x)$ 和 $\beta_i(x)$ 是 Hermite 插值基函数，表达式由 (4.27) 和 (4.28) 给出。式 (4.29) 中的 m_i 实际上是未知的，可由连续性条件 $S''(x_i-0)=S''(x_i+0)(i=0,1,\cdots,n-1)$ 来确定。

$S(x)$ 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

$$\begin{aligned} S''(x) &= y_k\alpha_k''(x) + y_{k+1}\alpha_{k+1}''(x) + m_k\beta_k''(x) + m_{k+1}\beta_{k+1}''(x) \\ &= \frac{6x - 2x_k - 4x_{k+1}}{h_k^2}m_k + \frac{6x - 4x_k - 2x_{k+1}}{h_k^2}m_{k+1} \\ &\quad + \frac{6(x_k + x_{k+1} - 2x)}{h_k^3}(y_{k+1} - y_k), \end{aligned} \tag{4.30}$$

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

$$\begin{aligned} S''(x) = &\frac{6x - 2x_{k-1} - 4x_k}{h_{k-1}^2}m_{k-1} + \frac{6x - 4x_{k-1} - 2x_k}{h_{k-1}^2}m_k \\ &+ \frac{6(x_{k-1} + x_k - 2x)}{h_{k-1}^2}(y_k - y_{k-1}), \end{aligned} \tag{4.31}$$

其中 $h_{k-1} = x_k - x_{k-1}$。

由连续性条件 $S''(x_i - 0) = S''(x_i + 0)$ ，得

$$\begin{aligned} &\frac{1}{h_{k-1}}m_{k-1} + 2(\frac{1}{h_{k-1}} + \frac{1}{h_k})m_k + \frac{1}{h_k}m_{k+1} \\ =&3\Big(\frac{y_{k+1} - y_k}{h_k^2} + \frac{y_k - y_{k-1}}{h_{k-1}^2}\Big) \quad (k = 1, 2, \cdots, n-1), \end{aligned} \tag{4.32}$$

整理后，式 (4.32) 可表示为

$$(1 - a_k)m_{k-1} + 2m_k + a_km_{k+1} = b_k \quad (k = 1, 2, \cdots, n-1), \tag{4.33}$$

其中

$$\begin{aligned} a_k &= \frac{h_{k-1}}{h_{k-1} + h_k}, \\ b_k &= 3\big[(1 - a_k)\frac{y_k - y_{k-1}}{h_{k-1}} + a_k\frac{y_{k+1} - y_k}{h_k}\big]. \end{aligned}$$

这是关于未知数 $m_0, m_1, \cdots, m_n$ 的 $n-1$ 个方程，如果补充边界条件

$$S'(x_0) = m_0 = y_0', S'(x_n) = m_n = y_n',$$

则可得关于未知数 $m_1,\cdots,m_n$ 的方程组

$$\begin{pmatrix} 2 & a_1 & & & \\ 1-a_1 & 2 & a_2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1-a_{n-2} & 2 & a_{n-2} \\ & & & 1-a_{n-1} & 2 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-2} \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} b_1-(1-a_1)y_0' \\ b_2 \\ \vdots \\ b_{n-2} \\ b_{n-1}-a_{n-1}y'n \end{pmatrix}. \tag{4.34}$$

方程组 (4.34) 为三对角方程，系数矩阵为对角占优阵，可用追赶法求解。

§4.6.2 三弯矩方程

假设已知 $S(x)$ 在节点 $x_i(i=0,1,\cdots,n)$ 处的二阶导数值，即设

$$S''(x_i)=M_i \quad (i=0,1,\cdots,n).$$

由于 $S(x)$ 在区间 $[x_k,x_{k+1}]$ 上是三次多项式，故 $S''(x)$ 在 $[x_k,x_{k+1}]$ 是线性函数，可表示为

$$S''(x)=M_k\frac{x_{k+1}-x}{h_k}+M_{k+1}\frac{x-x_k}{h_k}, \tag{4.35}$$

其中

$$h_k=x_{k+1}-x_k.$$

对式 (4.35) 作两次积分并利用 $S(x_k)=y_k$ 和 $S(x_{k+1})=y_{k+1}$ 定出积分常数，得

$$\begin{aligned} S(x)=&M_k\frac{(x_{k+1}-x)^3}{6h_k}+M_{k+1}\frac{(x-x_k)^3}{h_k} \\ &+\big(y_k-\frac{M_kh_k^2}{6}\big)\frac{x_{k+1}-x}{h_k}+\big(y_{k+1}-\frac{M_{k+1}h_k^2}{6}\big)\frac{x-x_k}{h_k}, \end{aligned}$$

对 $S(x)$ 求导得

$$\begin{aligned} S'(x)=&-M_k\frac{(x_{k+1}-x)^2}{2h_k}+M_{k+1}\frac{(x-x_k)^2}{2h_k} \\ &+\frac{y_{k+1}-y_k}{h_k}-\frac{M_{k+1}-M_k}{6}h_k. \end{aligned} \tag{4.36}$$

同理，可得 $S'(x)$ 在 $[x_{k-1},x_k]$ 上的表达式

$$\begin{aligned} S'(x)=&-M_{k-1}\frac{(x_k-x)^2}{2h_{k-1}}+M_k\frac{(x-x_{k-1})^2}{2h_{k-1}} \\ &+\frac{y_k-y_{k-1}}{h_{k-1}}-\frac{M_k-M_{k-1}}{6}h_{k-1}. \end{aligned} \tag{4.37}$$

由连续性条件 $S'(x_k-0)=S'(x_k+0)$，得

$$(1-a_k)M_{k-1}+2M_k+a_kM_{k+1}=b_k\ \ (k=1,2,\cdots,n-1), \tag{4.38}$$

其中

$$a_k=\frac{h_k}{h_{k-1}+h_k},\quad b_k=\frac{6}{h_k+h_{k-1}}\Big(\frac{y_{k+1}-y_k}{h_k}-\frac{y_k-y_{k-1}}{h_{k-1}}\Big).$$

如果补充两边界条件 $S''(x_0)=M_0=y_0''$ 和 $S''(x_n)=M_n=y_n''$，则可用追赶法解下列三对角方程，求得未知数 $M_1,M_2,\cdots,M_{n-1}$ 。

$$\begin{pmatrix} 2 & a_1 & & & \\ 1-a_1 & 2 & a_2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1-a_{n-2} & 2 & a_{n-2} \\ & & & 1-a_{n-1} & 2 \end{pmatrix}\begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{pmatrix}=\begin{pmatrix} b_1-(1-a_1)y_0'' \\ b_2 \\ \vdots \\ b_{n-2} \\ b_{n-1}-a_{n-1}y''n \end{pmatrix}. \tag{4.39}$$

§4.7 最小二乘法

在各种科学实验中，常常需要通过一组实验数据 $(x_i,y_i)(i=0,1,\cdots,n)$ 来估计函数 $f(x)=y$ 。插值方法就是处理这类问题的一种方法，但插值法严格要求插值曲线通过各个数据点，而实验数据经常带有误差。因此插值函数将保留一切测试误差，导致插值函数不能很好地反映数据集 $(x_i,y_i)(i=0,1,\cdots,n)$ 的总体趋势。

为克服上述缺点，人们常采用曲线拟合方法来进行数据处理。所谓曲线拟合就是从数据集 $(x_i,y_i)(i=0,1,\cdots,n)$ 中找出总体规律性，并构造一条能较好反映这种规律的曲线 $P(x)$。这里不要求曲线 $P(x)$ 点点通过数据点，但希望曲线 $P(x)$ 能尽量地靠近数据点，这就是误差 $\delta_i=P(x_i)-y_i(i=0,1,\cdots,n)$ 按某种标准达到最小。我们可采用下列三种标准来度量误差的大小：

$$\|\delta\|_1=\sum_{i=0}^{n}|\delta_i|,\quad \|\delta\|_2=(\sum_{i=0}^{n}\delta_i^2)^{1/2},\quad \|\delta\|_\infty=\max_{0\leqslant i\leqslant n}|\delta_i|.$$

由于 2- 范数在计算上较方便，通常采用 2- 范数作为总体误差的度量标准。我们称范数 $\|\delta\|_2$ 达到最小的曲线拟合方法为曲线拟合的最小二乘法。

具体地说，就是对给定的一组数据

$$(x_i,y_i)\qquad (i=0,1,\cdots,n)$$

在函数类 $\Phi=\{\varphi_0(x),\varphi_1(x),\cdots,\varphi_m(x)\}$ 中寻找一个函数 $P(x)$，使误差平方和

$$\|\delta\|_2^2=\sum_{i=0}^{n}\delta_i^2=\sum_{i=0}^{n}(P(x_i)-y_i)^2 \tag{4.40}$$

达到最小。这里 $\varphi_0(x),\varphi_1(x),\cdots,\varphi_m(x)$ 是 Φ 的一组线性无关的基函数，$P(x)$ 为 $\{\varphi_i(x)\}(i=0,1,\cdots,n)$ 的线性式，即

$$P(x)=a_0\varphi_0(x)+a_1\varphi_1(x)+\cdots+a_m\varphi_m(x)\ \ (m<n). \tag{4.41}$$

将式 (4.41) 代入式 (4.40) 中后可知，使误差平方和 $\|\delta\|_2^2$ 取最小值问题可转化为求下列多元函数

$$F(a_0,a_1,\cdots,a_m)=\sum_{i=0}^{n}\big(\sum_{j=0}^{m}a_j\varphi_j(x_i)-y_i\big)^2$$

的极小点 $(a_0^*,a_1^*,\cdots,a_m^*)$，即令

$$\frac{\partial F}{\partial a_k}=0\quad\ (k=0,1,\cdots,m),$$

由此得

$$\sum_{i=0}^{n}\big(\sum_{j=0}^{m}a_j\varphi_j(x_i)-y_i\big)\varphi_k(x_i)=0\ \ (k=0,1,\cdots,m). \tag{4.42}$$

记

$$(\varphi_j,\varphi_k)=\sum_{i=0}^{n}\varphi_j(x_i)\varphi_k(x_i),\qquad\qquad (f,\varphi_k)=\sum_{i=0}^{n}y_i\varphi_k(x_i)=d_k,$$

则式 (4.42) 可写为

$$\sum_{j=0}^{m}(\varphi_k,\varphi_j)a_j=d_k\quad (k=0,1,\cdots,m).$$

把它改写成矩阵形式为

$$\begin{pmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_m)\\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_m)\\ \vdots & \vdots & & \vdots\\ (\varphi_m,\varphi_0) & (\varphi_m,\varphi_1) & \cdots & (\varphi_m,\varphi_m)\end{pmatrix}\begin{pmatrix}a_0\\ a_1\\ \vdots\\ a_m\end{pmatrix}=\begin{pmatrix}d_0\\ d_1\\ \vdots\\ d_m\end{pmatrix}. \tag{4.43}$$

这是关于系数 $a_j(j=0,1,\cdots,m)$ 的线性方程组，也称为正规方程。由于 $\varphi_0,\varphi_1,\cdots,\varphi_m$ 线性无关，故方程组 (4.43) 的系数矩阵行列式不为零，因此方程组 (4.43) 有唯一解 $(a_0^*,a_1^*,\cdots,a_m^*)$。

例 4.9 已知一组实验数据如下：

i	1	2	3	4	5
x_i	165	123	150	123	141
y_i	187	126	172	125	148

求它的拟合曲线。

解 将给出数据标在坐标纸上，我们将看到各点在一条直线附近，故设拟合曲线为 $y = a_0 + a_1 x$，这里

$$n = 4, m = 1, \varphi_0(x) = 1, \varphi_1(x) = x$$

故

$$\begin{aligned}
(\varphi_0, \varphi_0) &= \sum_{i=0}^{4} \varphi_0(x_i)\varphi_0(x_i) = 5, \\
(\varphi_0, \varphi_1) &= \sum_{i=0}^{4} \varphi_0(x_i)\varphi_1(x_i) = \sum_{i=0}^{4} x_i = 702, \\
(\varphi_1, \varphi_0) &= \sum_{i=0}^{4} \varphi_1(x_i)\varphi_0(x_i) = 702, \\
d_0 &= \sum_{i=0}^{4} y_i\varphi_0(x_i) = \sum_{i=0}^{4} y_i = 758, \\
d_1 &= \sum_{i=0}^{4} y_i\varphi_1(x_i) = \sum_{i=0}^{4} y_i x_i = 108396.
\end{aligned}$$

于是正则方程为

$$\begin{cases} 5a_0 + 702a_1 = 758 \\ 702a_0 + 99864a_1 = 108396 \end{cases}$$

解得

$$a_0 = -60.9392, \quad a_1 = 1.5138.$$

于是所求拟合曲线为

$$y = -60.9392 + 1.5138x.$$

习 题 四

4.1 给出概率积分

$$f(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-x^2}dx$$

的数据表：

x	0.46	0.47	0.48	0.49
$f(x)$	0.4846555	0.4937542	0.5027498	0.5116683

试用二次插值计算 $f(0.472)$.

4.2 已知 $y=\sin(x)$ 的函数表

x	1.5	1.6	1.7
$\sin(x)$	0.99749	0.99957	0.99166

试构造出差商表，利用二次 Newton 插值公式计算 $\sin(1.609)$（保留 5 位小数），并估计其误差。

4.3 设 $x_j(j=0,1,\cdots,n)$ 为互异节点，求证

1. $\sum\limits_{j=0}^{n} x_j^k l_j(x)\equiv x^k \quad (k=0,1,\cdots,n)$;

2. $\sum\limits_{j=0}^{n}(x_j-x)^k l_j(x)\equiv 0 \quad (k=0,1,\cdots,n)$.

4.4 若 $y_n=2^n$ ，求 $\Delta^2 y_n$ 及 $\delta^4 y_n$。

4.5 求不高于 4 次的多项式 $H(x)$, 使它满足 $H(1)=-2,\ H'(1)=4,\ H(2)=H'(2)=0,\ H(3)=2$, 并写出余项表达式。

4.6 证明两点三次 Hermite 插值余项是

$$R_3(x)=\frac{1}{4!}f^{(4)}(\xi)(x-x_k)^2(x-x_{k+1})^2,\quad \xi\in(x_k,x_{k+1}).$$

4.7 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

4.8 用最小二乘法求一个形如 $y=a+bx^2$ 的经验公式，使与下列数据相拟合

x	19	25	31	38	41
y	19.0	32.3	49.0	73.3	97.8

第五章　数值积分

本章主要涉及定积分 $\int_a^b f(x)dx$ 的数值计算，也许有人会问，定积分的计算不是有了 Newton-Leibniz 公式吗？事实上，实际问题并非如此简单，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 $\int_a^b e^{-x^2}dx$、$\int_a^b \sin(x^2)dx$ 等。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，本章要考虑定积分的数值计算。

§5.1　机械求积公式

§5.1.1　数值积分的基本方法

求定积分 $\int_a^b f(x)dx$ 的关键困难在于被积函数 $f(x)$ 的复杂性。为此，将 $f(x)$ 用简单函数近似替代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 $\int_a^b f(x)dx$ 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

$$y = f[\theta a + (1-\theta)b], \quad \theta \in [0,1]$$

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，$\theta = 0,\ \frac{1}{2},\ 1$ 时分别称之为左矩公式，中矩公式及右矩公式。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

$$y = px^2 + qx + r, \quad a \leqslant x \leqslant b,$$

其中 p,q,r 由方程组

$$\begin{cases} pa^2 + qa + r = f(a), \\ p(\dfrac{b+a}{2})^2 + q(\dfrac{b+a}{2}) + r = f(\dfrac{b+a}{2}), \\ pb^2 + qb + r = f(b), \end{cases}$$

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

$$\int_a^b f(x)dx \approx \frac{b-a}{6}[f(a)+4f(\frac{a+b}{2})+f(b)]. \tag{5.3}$$

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

$$\int_a^b f(x)dx \approx \sum_{n=0}^{N} A_n f(x_n), \tag{5.4}$$

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**.

§5.1.2 代数精度法

由 Taylor 展开定理可知，任一充分可微函数 $f(x)$ 均能展开为一个关于 x 的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 能准确成立。为此引入

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。

由定义 5.1 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可获得构造求积公式的一种方法，称之为**代数精度法**。如果令公式 (5.4) 对 $f(x)=x^i$ $(i=0,1,\cdots,N)$ 准确成立，那么得线性方程组

$$\sum_{n=0}^{N} A_n x_n^i = \frac{b^{i+1}-a^{i+1}}{i+1}, \quad i=0,1,\cdots,N. \tag{5.5}$$

当节点 $x_n(n=0,1,\cdots,N)$ 给定且互异时，诸系数 A_n 即可由 (5.5) 唯一确定。

例 5.1 试确定一个具有 3 次代数精度的公式

$$\int_0^3 f(x)dx \approx A_0 f(x_0)+A_1 f(x_1)+A_2 f(x_2)+A_3 f(x_3).$$

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

$$\int_0^3 f(x)dx \approx \frac{3}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)],$$

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。

§5.1.3 插值求积法

本段我们以插值思想来构造求积公式。即根据已知节点处的函数值，构造一个插值多项式 $P_N(x)$ 代替被积函数 $f(x)$，然后用 $\int_a^b P_N(x)dx$ 作为积分 $\int_a^b f(x)dx$ 的近似值。这样获得的求积公式称为**插值型求积公式**.

对于积分 $\int_a^b f(x)dx$，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 $\int_a^b f(x)dx$ 中得

$$\int_a^b f(x)dx = \sum_{n=0}^{N} A_n f(x_n) + R_N(f), \tag{5.7}$$

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

$$\int_a^b f(x)dx \approx \sum_{n=0}^{N} A_n f(x_n). \tag{5.9}$$

若有 $\max\limits_{x\in[a,b]} |f^{(N+1)}(x)| = M_{N+1}$，则得其余项 $R_n(f)$ 的估计式

$$|R_n(f)| \leqslant \frac{M_{N+1}}{(N+1)!}\int_a^b |\omega_{N+1}(x)|dx. \tag{5.10}$$

至此，我们获得了又一个确定求积公式的方法。前述两种方法具有如下的关系。

定理 5.1 $N+1$ 个节点的求积公式为插值型的充要条件是该公式至少有 N 次代数精度。

证 先证必要性：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。

再证充分性：设公式 (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

$$\int_a^b l_n(x)dx=\sum_{j=0}^{N}A_j l_n(x_j).$$

而 $l_n(x_j)=\delta_{nj}$，因此

$$A_n=\int_a^b l_n(x)dx.$$

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。

§5.2 Newton-Cotes 公式

本节给出具等距节点的插值型求积公式— Newton-Cotes 求积公式，并具体讨论其二种特殊形式：梯形公式及 Simpson 公式。

§5.2.1 公式的一般形式

当公式 (5.9) 取等距节点

$$x_n=a+nh,\quad n=0,1,2,\cdots,N;\quad h=\frac{b-a}{N}$$

且记 $x = a + th$ 时，其系数 A_n 由 (5.8) 得

$$A_n = \frac{(-1)^{N-n}}{n!\ (N-n)!} \int_0^N \prod_{i=0, i\neq n}^{N} (t-i)dt, \quad n = 0, 1, \cdots, N.$$

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：

$$B_n = B_{N-n}, \quad n = 0, 1, \cdots, N.$$

理论分析表明，当公式 (5.11) 的阶数 N 较大时，其稳定性会产生较大的误差积累，因此 Newton-Cotes 公式中有实用价值的往往是一些低阶公式。

§5.2.2 两种低阶公式及其余项

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 $f(x) \in C^{(2)}((a, b))$，则其余项由 (5.8) 及积分中值定理知

$$\begin{aligned} R_1(f) &= \int_a^b \frac{f^{(2)}(\xi)}{2!}(x-a)(x-b)dx \\ &= \frac{f^{(2)}(\tilde{\xi})}{2!} \int_a^b (x-a)(x-b)dx \\ &= -\frac{(b-a)^3}{12} f^{(2)}(\tilde{\xi}), \quad \tilde{\xi} \in (a, b). \end{aligned} \tag{5.13}$$

在公式 (5.11) 中取 $N = 2$，则得 Simpson 公式

$$\int_a^b f(x)dx \approx \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)]. \tag{5.14}$$

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 $H(x)$ 。利用微分中值定理可证得

$$f(x)=H(x)+\frac{f^{(4)}(\xi)}{4!}(x-a)(x-\frac{a+b}{2})^2(x-b), \quad \xi\in(a,b).$$

由于 Simpson 公式具有三次代数精度，则根据 (5.15) 及积分中值定理得

$$\begin{aligned} \int_a^b f(x)dx &= \int_a^b H(x)dx+\frac{1}{4!}\int_a^b f^{(4)}(\xi)(x-a)(x-\frac{a+b}{2})^2(x-b)dx \\ &= \frac{b-a}{6}\left[H(a)+4H(\frac{a+b}{2})+H(b)\right] \\ &\quad +\frac{f^{(4)}(\xi)}{4!}\int_a^b (x-a)(x-\frac{a+b}{2})^2(x-b)dx \\ &= \frac{b-a}{6}\left[f(a)+4f(\frac{a+b}{2})+f(b)\right]-\frac{(b-a)^5}{2880}f^{(4)}(\xi), \quad \xi\in(a,b). \end{aligned}$$

故公式 (5.14) 的余项为

$$R_2(f)=-\frac{(b-a)^5}{2880}f^{(4)}(\xi), \quad \xi\in(a,b). \tag{5.16}$$

§5.2.3 复合求积公式

上述低阶公式要真正做到实用，其精度仍有待提高。为此，本段将把积分区间 $[a,b]$ 等分成若干子区间 $[x_n,x_{n+1}]$ $(n=0,1,2,\cdots,N)$，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。这种公式称为**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

$$\int_{x_n}^{x_{n+1}} f(x)dx=\frac{h}{2}[f(x_n)+f(x_{n+1})]-\frac{h^3}{12}f^{(2)}(\eta_n), \quad \eta_n\in(x_n,x_{n+1})$$

求和得

$$\begin{aligned}\int_a^b f(x)dx &= \frac{h}{2}\sum_{n=0}^{N-1}[f(x_n)+f(x_{n+1})] - \frac{h^3}{12}\sum_{n=0}^{N-1} f^{(2)}(\eta_n) \\ &= \frac{h}{2}\left[f(a)+2\sum_{n=1}^{N-1} f(x_n)+f(b)\right] - \frac{h^2}{12}\sum_{n=0}^{N-1}[f^{(2)}(\eta_n)h]\end{aligned}$$

故得复合梯形公式

$$\int_a^b f(x)dx \approx \frac{h}{2}\left[f(a)+2\sum_{n=1}^{N-1} f(x_n)+f(b)\right], \tag{5.17}$$

其余项

$$\begin{aligned}\tilde{R}_1(f) &= -\frac{h^2}{12}\sum_{n=0}^{N-1}[f^{(2)}(\eta_n)h] \\ &\approx -\frac{h^2}{12}\int_a^b f^{(2)}(x)dx \quad (\text{根据定积分定义}) \\ &= -\frac{h^2}{12}[f'(b)-f'(a)].\end{aligned} \tag{5.18}$$

若在 $[x_n, x_{n+1}]$ 上使用带余项的 Simpson 公式 (记 $x_{n+\frac{1}{2}} = \frac{x_n+x_{n+1}}{2}$)

$$\int_{x_n}^{x_{n+1}} f(x)dx = \frac{h}{6}[f(x_n)+4f(x_{n+\frac{1}{2}})+f(x_{n+1})] - \frac{h^5}{2880}f^{(4)}(\theta_n), \quad \theta_n \in (x_n, x_{n+1})$$

求和得

$$\begin{aligned}\int_a^b f(x)dx &= \frac{h}{6}\sum_{n=0}^{N-1}[f(x_n)+4f(x_{n+\frac{1}{2}})+f(x_{n+1})] - \frac{h^5}{2880}\sum_{n=0}^{N-1} f^{(4)}(\theta_n) \\ &= \frac{h}{6}\left[f(a)+4\sum_{n=0}^{N-1} f(x_{n+\frac{1}{2}})+2\sum_{n=1}^{N-1} f(x_n)+f(b)\right] - \frac{h^4}{2880}\sum_{n=0}^{N-1}[f^{(4)}(\theta_n)h]\end{aligned}$$

故得复合 Simpson 公式

$$\int_a^b f(x)dx \approx \frac{h}{6}\left[f(a)+4\sum_{n=0}^{N-1} f(x_{n+\frac{1}{2}})+2\sum_{n=1}^{N-1} f(x_n)+f(b)\right], \tag{5.19}$$

其余项

$$\begin{aligned}\tilde{R}_2(f) &= -\frac{h^4}{2880}\sum_{n=0}^{N-1}[f^{(4)}(\theta_n)h] \\ &\approx -\frac{h^4}{2880}\int_a^b f^{(4)}(x)dx \quad (\text{根据定积分定义}) \\ &= -\frac{h^4}{2880}[f^{(3)}(b)-f^{(3)}(a)].\end{aligned} \tag{5.20}$$

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 (5.19) 写成

$$\int_a^b f(x)dx \approx \frac{\tilde{h}}{3}\left\{f(a)-f(b)+2\sum_{n=1}^{N}[2f(\tilde{x}_{2n-1})+f(\tilde{x}_{2n})]\right\}, \tag{5.21}$$

其中 $\tilde{h}=\frac{b-a}{2N},\ \tilde{x}_n=a+n\tilde{h},\quad n=1,2,\cdots,2N.$

例 5.2 取步长 $h=\frac{1}{8}$，分别用公式 (5.17) 及 (5.21) 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 (5.17) 得

$$I\approx\frac{1}{16}\left[f(0)+2\sum_{n=1}^{7}f(\frac{n}{8})+f(1)\right]=9.456908635827014e-001.$$

由公式 (5.21) 得

$$I\approx\frac{1}{24}\left\{f(0)-f(1)+2\sum_{n=1}^{4}\left[2f(\frac{2n-1}{8})+f(\frac{n}{4})\right]\right\}=9.460833108884721e-001.$$

将上述两个近似值与 I 的准确值 $0.9460831\cdots$ 相比较可知: 复合梯形公式仅有两位有效数字，而 Simpson 公式有六位有效数字。故由此可见，在计算量基本相同的情况下, 后者精度高于前者。

§5.3 变步长求积公式

梯形公式与 Simpson 公式的复合使其精度获得改善，但两者均属于定步长公式，若要求达到某个计算精度，其步长的选择则成为一件困难的事情。为此，本节介绍一种在计算机上自动选择步长的变步长方法，同时也将涉及其加速收敛技巧。

§5.3.1 变步长梯形求积公式

在区间 $[a,b]$ 上使用梯形公式及其复合形式逐次计算积分 $\int_a^b f(x)dx$ 。首先，利用梯形公式可得积分近似值

$$T_0 \approx \frac{b-a}{2}[f(a)+f(b)],$$

此时步长 $h_0 = b-a$; 再将 $[a,b]$ 二等分，即取步长 $h_1 = \frac{b-a}{2}$，使用 $N=2$ 时的复合梯形公式得积分近似值

$$\begin{aligned} T_1 &= \frac{b-a}{4}[f(a)+2f(\frac{a+b}{2})+f(b)] \\ &= \frac{1}{2}T_0 + \frac{b-a}{2}f(a+\frac{b-a}{2}); \end{aligned}$$

将 $[a,b]$ 四等分，即取步长 $h_2 = \frac{b-a}{2^2}$ ，使用 $N=4$ 时的复合梯形公式得积分近似值

$$\begin{aligned} T_2 &= \frac{b-a}{2^3}\{f(a)+2\sum_{n=1}^{3} f[a+\frac{n(b-a)}{2^2}]+f(b)\} \\ &= \frac{1}{2}T_1 + \frac{b-a}{2^2}\sum_{n=1}^{2} f[a+\frac{2n-1}{2^2}(b-a)]; \end{aligned}$$

进而将区间八等分，即取步长 $h_3 = \frac{b-a}{2^3}$，使用 $N=8$ 时的复合梯形公式得积分近似值

$$\begin{aligned} T_3 &= \frac{b-a}{2^4}\{f(a)+2\sum_{n=1}^{7} f[a+\frac{n(b-a)}{2^3}]+f(b)\} \\ &= \frac{1}{2}T_2 + \frac{b-a}{2^3}\sum_{n=1}^{2^2} f[a+\frac{2n-1}{2^3}(b-a)]; \end{aligned}$$

依次类推可得变步长梯形公式

$$T_K = \frac{1}{2}T_{K-1} + \frac{b-a}{2^K}\sum_{n=1}^{2^{K-1}} f[a+\frac{2n-1}{2^K}(b-a)], \quad K=1,2,\cdots. \tag{5.22}$$

进一步，我们推导其事后误差估计式，以获得其计算过程 (5.22) 的终止准则。记积分精确值 $\int_a^b f(x)dx := I$，则由 (5.18) 得

$$\begin{cases} I - T_K \approx -\dfrac{h^2}{12}[f'(b)-f'(a)], \\ I - T_{K+1} \approx -\dfrac{h^2}{48}[f'(b)-f'(a)], \end{cases} \tag{5.23}$$

其中步长 $h = \frac{b-a}{2^K}$ 。由 (5.23) 即得其事后误差估计

$$I - T_{K+1} \approx \frac{1}{3}(T_{K+1} - T_k). \tag{5.24}$$

鉴于上式是一个近似估计，因此实际计算中往往保守地以 $|T_{K+1} - T_k|$ 作为当前步近似值 T_{K+1} 的误差。即若预定精度为 ϵ，则以式 (5.22) 计算积分近似值，直至 $|T_{K+1} - T_k| < \epsilon$ 终止计算，并以当前值 T_{K+1} 作为欲求近似值。我们不难看出，上述步骤均能在计算机上自动实现。此外，对于 Simpson 公式，也可类似构造出相应的变步长公式。

§5.3.2 Romberg 算法

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。但这一切均是以提高计算量为代价的。为此，本节介绍一种加速收敛技巧，其基本思想是采用 Richardson 外推。

记 $T_0^{(K)}$ 为二分 K 次积分区间 $[a,b]$ 后利用复合梯形公式所得积分近似值（称为**梯形值**），$T_m^{(K)}$ 为将序列 $\{T_0^{(K)}\}$ 经 m 次外推后所得积分近似值。其加速步骤如下：

(1) 计算初始值

$$T_0^{(0)} = \frac{b-a}{2}[f(a) + f(b)]; \tag{5.25}$$

(2) 将积分区间 $[a,b]$ 二分一次计算梯形值

$$T_0^{(1)} = \frac{1}{2}T_0^{(0)} + \frac{b-a}{2}f(a + \frac{b-a}{2});$$

(3) 鉴于理论分析表明梯形值 $T(h)$ 具有如下误差渐近展开式

$$T(h) - I = \sum_{i=1}^{\infty} A_i h^{2i}, \quad \text{其中 诸}A_i\text{与}h\text{无关.}$$

因此, 据习题 1.6 可将步骤 (1)、(2) 中的梯形值外推一次得逼近值

$$T_1^{(0)} = \frac{4T_0^{(1)} - T_0^{(0)}}{4-1}, \quad \text{其截断误差为}\mathcal{O}(h^4);$$

(4) 再将区间 $[a,b]$ 二分二次计算梯形值

$$T_0^{(2)} = \frac{1}{2}T_0^{(1)} + \frac{b-a}{2^2}\sum_{n=1}^{2} f\left[a + \frac{2n-1}{2^2}(b-a)\right];$$

(5) 又由习题 1.5 可将 $T_0^{(1)}$、$T_0^{(2)}$ 外推一次得逼近值

$$T_1^{(1)} = \frac{4T_0^{(2)} - T_0^{(1)}}{4-1}, \quad \text{其截断误差为}\mathcal{O}(h^4);$$

(6) 进一步将 $T_1^{(0)}$、$T_1^{(1)}$ 外推一次得逼近值

$$T_2^{(0)} = \frac{4^2 T_1^{(1)} - T_1^{(0)}}{4^2 - 1}, \quad \text{其截断误差为} \mathcal{O}(h^6);$$

(7) 依据公式

$$\begin{cases} T_0^{(K)} = \dfrac{1}{2} T_0^{(K-1)} + \dfrac{b-a}{2^K} \displaystyle\sum_{n=1}^{2^{K-1}} f\left[a + \frac{2n-1}{2^K}(b-a)\right], \quad K = 1,\ 2,\ \cdots \\ T_m^{(l)} = \dfrac{4^m T_{m-1}^{(l+1)} - T_{m-1}^{(l)}}{4^m - 1}, \quad l = 0,\ 1,\ \cdots;\quad m = 1,\ 2,\ \cdots \end{cases}$$

重复上述步骤即可获得系列逼近值：

$$\begin{array}{ccccccccc} T_0^{(0)} & & & & & & & & \\ \downarrow & \searrow & & & & & & & \\ T_0^{(1)} & \rightarrow & T_1^{(0)} & & & & & & \\ \downarrow & \searrow & & \searrow & & & & & \\ T_0^{(2)} & \rightarrow & T_1^{(1)} & \rightarrow & T_2^{(0)} & & & & \\ \downarrow & \searrow & & \searrow & & \searrow & & & \\ T_0^{(3)} & \rightarrow & T_1^{(2)} & \rightarrow & T_2^{(1)} & \rightarrow & T_3^{(0)} & & \\ \vdots & & & \ddots & & \ddots & & \ddots & \end{array}$$

(8) 精度控制（假设预定精度为 ϵ）：若 $|T_m^{(0)} - T_{m-1}^{(0)}| < \epsilon$，则停止计算，取积分近似值为当前逼近值 $T_m^{(0)}$；否则转入 (7) 继续计算直至 $|T_m^{(0)} - T_{m-1}^{(0)}| < \epsilon$.

上述计算方法称为**Romberg 算法**.

例 5.3 用 Romberg 算法计算积分（精确到 10^{-4}）

$$I = \int_0^1 \frac{dx}{1+x^2}.$$

解 据 Romberg 算法依次计算得

$$\begin{aligned} &T_0^{(0)} = 7.5000e-001, \\ &T_0^{(1)} = 7.7500e-001, \quad T_1^{(0)} = 7.8333e-001, \\ &T_0^{(2)} = 7.8279e-001, \quad T_1^{(1)} = 7.8539e-001, \quad T_2^{(0)} = 7.8553e-001, \\ &T_0^{(3)} = 7.8475e-001, \quad T_1^{(2)} = 7.8540e-001, \quad T_2^{(1)} = 7.8540e-001, \quad T_3^{(0)} = 7.8540e-001, \end{aligned}$$

由于 $T_3^{(0)}$ 已达到预定精度，故取 $I \approx T_3^{(0)} = 7.8540e-001$.

§5.4 Gauss 求积公式

据 §5.1 我们已经知道，通常插值型求积公式的代数精度与节点个数有关，要提高其精度必然以增加节点个数为代价。但是我们也知道，节点的无限增加将导致其稳定性能减弱，且当 $N \to \infty$ 时 $\sum\limits_{n=0}^{N} A_n f(x_n)$ 不一定收敛于原积分 $\int_a^b f(x)dx$ 的值。为在一定程度上克服这些缺陷，我们于本节引入 Gauss 公式。

§5.4.1 公式的构造

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 $x_0, x_1, \cdots, x_N$ 称为 Gauss 点。

从定义 5.2 可知，Gauss 型求积公式比通常同级插值型公式的精度高。为公式构造的简单起见，下面我们假定公式 (5.9) 的积分限为 $a=-1,\ b=1$，而对于更一般情形则可作变换

$$x = \frac{2}{b-a}(t - \frac{a+b}{2}), \tag{5.26}$$

使积分

$$\int_a^b f(t)dt = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right)dx.$$

定理 5.2 $x_0, x_1, \cdots, x_N$ 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

$$\int_{-1}^{1} \omega_{N+1}(x)Q(x)dx = 0. \tag{5.27}$$

证 先证必要性：设 x_n $(n=0,1,2,\cdots,N)$ 是 Gauss 点，则公式 (5.9) 对任意 $2N+1$ 次多项式都精确成立。而多项式 $Q(x)\omega_{N+1}(x)$ 的次数至多为 $2N+1$，则

$$\int_{-1}^{1} Q(x)\omega_{N+1}(x)dx = \sum_{n=0}^{N} A_n Q(x_n)\omega_{N+1}(x_n) = 0.$$

再证充分性：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

$$f(x) = Q(x)\omega_{N+1}(x) + r(x). \tag{5.28}$$

积分得

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} Q(x)\omega_{N+1}(x)dx + \int_{-1}^{1} r(x)dx = \sum_{n=0}^{N} A_n r(x_n).$$

进一步由 (5.28) 得 $f(x_n)=r(x_n)$ $(n=0,1,2,\cdots,N)$，因此

$$\int_{-1}^{1} f(x)dx=\sum_{n=0}^{N} A_n f(x_n). \tag{5.29}$$

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 (5.29) 对 $2N+2$ 次多项式不精确成立，故 (5.29) 为 Gauss 型求积公式，即 $x_0,x_1,\cdots,x_N$ 为 Gauss 点。

由定理 5.2 可知，若能找到满足 (5.27) 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，为解决这一问题，我们引入 Legendre 多项式及其相关结论。

定义 5.3 一个仅以区间 $[-1,1]$ 上的 Gauss 点 x_n $(n=0,1,2,\cdots,N)$ 为零点的 $N+1$ 次多项式称为 Legendre 多项式。

定理 5.3 首项系数为 1 的 Legendre 多项式可唯一地表示为

$$\omega_{N+1}(x)=\frac{(N+1)!}{(2N+2)!}\frac{d^{N+1}[(x^2-1)^{N+1}]}{dx^{N+1}},\quad N=0,1,\cdots.$$

证 考虑 $2N+2$ 次多项式

$$u(x)=\underbrace{\int_{-1}^{x}\int_{-1}^{x}\cdots\int_{-1}^{x}}_{N+1}\omega_{N+1}(x)dxdx\cdots dx,$$

其满足

$$u^{(N+1)}(x)=\omega_{N+1}(x),\quad u^{(i)}(-1)=0\quad (i=0,1,2,\cdots,N). \tag{5.30}$$

设 $v(x)$ 为任意 N 次多项式, 则由 (5.30) 得

$$\begin{aligned}
&\int_{-1}^{1} v(x)\omega_{N+1}(x)dx = \int_{-1}^{1} v(x)u^{(N+1)}(x)dx \\
=&[v(x)u^{(N)}(x)]_{-1}^{1} - \int_{-1}^{1} u^{(N)}(x)v'(x)dx \\
=&v(1)u^{(N)}(1) - \left\{[v'(x)u^{(N-1)}(x)]_{-1}^{1} - \int_{-1}^{1} u^{(N-1)}(x)v''(x)dx\right\} \\
&\qquad\qquad\qquad\qquad \vdots \quad \text{(继续逐次分部积分)} \\
=&\sum_{i=0}^{N}(-1)^i v^{(i)}(1)u^{(N-i)}(1) + \int_{-1}^{1} u(x)v^{(N+1)}(x)dx \\
=&\sum_{i=0}^{N}(-1)^i v^{(i)}(1)u^{(N-i)}(1).
\end{aligned}$$

而由定理 5.2 知

$$\int_{-1}^{1} v(x)\omega_{N+1}(x)dx = 0,$$

则

$$\sum_{i=0}^{N}(-1)^i v^{(i)}(1)u^{(N-i)}(1) = 0.$$

进一步据 $v(x)$ 的任意性得

$$u^{(N-i)}(1) = 0 \quad (i = 0, 1, 2, \cdots, N). \tag{5.31}$$

由 (5.30) 和 (5.31) 即知 $x = \pm 1$ 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 $\omega_{N+1}(x)$ 的首项系数为 1, 则要 $c = \frac{(N+1)!}{(2N+2)!}$ 。因此由 (5.30) 第一式有

$$\omega_{N+1}(x) = \frac{(N+1)!}{(2N+2)!}\frac{d^{N+1}[(x^2-1)^{N+1}]}{dx^{N+1}}.$$

至此, 定理 5.3 即解决了 Gauss 点的确定问题。

例 5.4 试构造 Gauss 型求积公式

$$\int_{-1}^{1} f(x)dx \approx A_0 f(x_0) + A_1 f(x_1) + A_2 f(x_2),$$

并由此计算

$$\int_0^1 \frac{\sqrt{t}}{(1+t)^2}dt \quad (\text{精确到}10^{-4}).$$

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

$$x_0 = -\sqrt{\frac{3}{5}}, \quad x_1 = 0, \quad x_2 = \sqrt{\frac{3}{5}}.$$

其公式的系数由 (5.8) 第一式得

$$A_0 = \frac{5}{9}, \quad A_1 = \frac{8}{9}, \quad A_2 = \frac{5}{9}.$$

故所求公式为

$$\int_{-1}^1 f(x)dx \approx \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}f(0) + \frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right). \tag{5.32}$$

据 (5.26) 作变换

$$x = 2(t - \frac{1}{2}),$$

且由公式 (5.32) 得

$$\begin{aligned}
&\int_0^1 \frac{\sqrt{t}}{(1+t)^2}dt = \sqrt{2}\int_{-1}^1 \frac{\sqrt{x+1}}{(x+3)^2}dx \\
\approx&\sqrt{2}\left\{\frac{5}{9}\frac{\sqrt{-\sqrt{\frac{3}{5}}+1}}{(-\sqrt{\frac{3}{5}}+3)^2} + \frac{8}{9}\frac{1}{3^2} + \frac{5}{9}\frac{\sqrt{\sqrt{\frac{3}{5}}+1}}{(\sqrt{\frac{3}{5}}+3)^2}\right\} \\
=&2.8845e-001.
\end{aligned}$$

§5.4.2 Gauss 求积公式的特征

定理 5.4 Gauss 求积公式的全体系数 $A_n > 0$, 且

$$A_n = \int_{-1}^1 \left[\frac{\omega_{N+1}(x)}{(x-x_n)\omega'_{N+1}(x)}\right]^2 dx, \quad n = 0, 1, 2, \cdots, N.$$

证 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

$$\int_{-1}^{1}\left[\frac{\omega_{N+1}(x)}{(x-x_n)\omega'_{N+1}(x)}\right]^2dx=\sum_{j=0}^{n}A_jl_n(x_j)=A_n,\quad n=0,1,2,\cdots,N.$$

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

$$\varepsilon=\sum_{n=0}^{N}A_n\varepsilon_n,$$

由此有误差估计

$$|\varepsilon|\leqslant(\sum_{n=0}^{N}|A_n|)\max_{0\leqslant n\leqslant N}|\varepsilon_n|. \tag{5.33}$$

对某些插值型求积公式，当 N 增大时 $\sum\limits_{n=0}^{N}|A_n|$ 将无界。此即说明 N 增大到一定程度时这些求积公式已失去其实用价值。而对于 Gauss 求积公式，一方面由于取 $f(x)=1$ 时有

$$\sum_{n=0}^{N}A_n=\int_{-1}^{1}1dx=2,$$

另一方面根据定理 5.4 诸 $A_n>0$，则由 (5.33) 有

$$|\varepsilon|\leqslant 2\max_{0\leqslant n\leqslant N}|\varepsilon_n|.$$

即 Gauss 公式的误差是可控制的，即具较好的稳定性能。此外，我们可以证明：若 $f(x)\in C([-1,1])$，则对 Gauss 求积公式有

$$\lim_{N\to\infty}\sum_{n=0}^{N}A_nf(x_n)=\int_{-1}^{1}f(x)dx.$$

Gauss 求积公式在计算机上的实现技术完全可采用类似于前述低阶 Newton-Cotes 公式的处理手段。

习 题 五

5.1 试确定下面求积公式

$$\int_{-1}^{1} f(x)dx \approx C[f(x_0) + f(x_1) + f(x_2)],$$

使其具三次代数精度。

5.2 在区间 $[a, b]$ 上导出含五个节点的 Newton-Cotes 公式, 并指出其余项及代数精度。

5.3 取步长 $h = \frac{1}{6}$, 分别用复合梯形公式及复合 Simpson 公式计算

$$\int_{1}^{2} \frac{x}{\ln(x+1)}dx.$$

5.4 用变步长梯形求积公式计算 (精确到 10^{-4})

$$\int_{0}^{1} e^{-x^2}dx.$$

5.5 用 Romberg 算法计算积分 (精确到 10^{-4})

$$\int_{0}^{\frac{\pi}{4}} \sin(x^2)dx.$$

5.6 试构造两点 Gauss 公式

$$\int_{-1}^{1} f(x)dx \approx A_0 f(x_0) + A_1 f(x_1),$$

并由此计算积分 (精确到 10^{-4})

$$\int_{0}^{1} \sqrt{1+2x}dx.$$

5.7 证明: 若求积公式

$$\int_{-1}^{1} f(x)dx \approx \sum_{n=0}^{N} A_n f(x_n)$$

为 Gauss 型的, 且被积函数 $f(x) \in C([-1, 1])$, 则有

$$\lim_{N\to\infty} \sum_{n=0}^{N} A_n f(x_n) = \int_{-1}^{1} f(x)dx.$$

第六章　常微分方程初值问题的数值解法

对于一阶常微分方程初值问题

$$\begin{cases} \dfrac{dy}{dt} = f(t, y(t)), & t \in [\,a, b\,] \\ y(a) = y_0, \end{cases} \tag{6.1}$$

我们已知当其右函数 $f(t, y) \in C([a, b] \times R)$ 且满足经典 Lipschitz 条件

$$|f(t, y_1) - f(t, y_2)| \leqslant L|y_1 - y_2|, \qquad t \in [a, b], \quad y_1, y_2 \in R \tag{6.2}$$

时，其连续可微解 $y(t)$ 在 $[a, b]$ 上存在且唯一。理论上虽已证明在适当条件下其解的存在性，但实际求解此类问题时，仍十分困难，甚至对一些非常简单的问题，如: 对于初值问题

$$\begin{cases} y'(t) = \exp(-t^2), & t \in [\,0, 1\,] \\ y(0) = 1, \end{cases}$$

我们易求得其解析解

$$y(t) = 1 + \int_0^t \exp(-x^2)dx, \quad t \in [0, 1],$$

但由于其解中积分项 $\int_0^t \exp(-x^2)dx$ 的值无法精确得到，因此解析方法并不能确切告诉我们 $y(t)$ 的值。级数解法，逐次逼近法等一些近似方法虽能求解部分初值问题，但仍存在着极大的局限性。为此本章将介绍求解问题 (6.1) 的通用方法 – 数值解法，其基本思想是将问题 (6.1) 离散化为差分方程，然后通过解差分方程而获得其数值解。

§6.1　基本离散方法

本节将介绍常微分方程初值问题的三种基本离散方法, 即差商逼近法, 数值积分法和 Taylor 展开法. 在下列算法构造中，我们恒记节点 $t_n = a + nh$ $(n = 0, 1, 2, \cdots, N)$，步长 $h \leqslant \frac{b-a}{N}$，$y_n \approx y(t_n)$ 及 $f_n := f(t_n, y_n) \approx y'(t_n)$.

§6.1.1　差商逼近法

差商逼近法即是用适当差商逼近导数值而使系统 (6.1) 离散化的一种方法. 如: 当 h 充分小时, 我们有

$$y'(t_n) \approx \frac{y(t_{n+1}) - y(t_n)}{h}, \qquad y'(t_{n+1}) \approx \frac{y(t_n) - y(t_{n+1})}{-h}, \tag{6.3}$$

若引入参数 $\theta \in [0,1]$, 则由上两式得

$$\theta y'(t_n) + (1-\theta) y'(t_{n+1}) \approx \frac{y(t_{n+1}) - y(t_n)}{h},$$

即

$$y(t_{n+1}) \approx y(t_n) + h[\theta y'(t_n) + (1-\theta) y'(t_{n+1})]. \tag{6.4}$$

采用先前约定符号即得求解系统 (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, 仅需要先前一步的信息 y_n, f_n, 因此我们称这种方法为**单步法**. 特别, 若 $\theta = 1$ 则得**显式 Euler 法**

$$y_{n+1} = y_n + h f_n; \tag{6.6}$$

若 $\theta = 0$, 则得**隐式 Euler 法**

$$y_{n+1} = y_n + h f_{n+1}; \tag{6.7}$$

若 $\theta = \frac{1}{2}$, 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} 的隐式方程才可获得当前数值解 y_{n+1}. 隐式方法的若干实现技巧可详见 §6.4.

若引入中间逼近值

$$Y_{1,\,n} = y_n, \quad Y_{2,\,n} = y_n + h[\theta f(t_n, Y_{1,\,n}) + (1-\theta) f(t_n + h, Y_{2,\,n})],$$

则方法 (6.5) 也可等价地写成二级 Runge-Kutta 方法形式

$$\begin{cases} Y_{1,\,n} = y_n, \\ Y_{2,\,n} = y_n + h[\theta f(t_n, Y_{1,\,n}) + (1-\theta) f(t_n + h, Y_{2,\,n})], \\ y_{n+1} = y_n + h[\theta f(t_n, Y_{1,\,n}) + (1-\theta) f(t_n + h, Y_{2,\,n})]. \end{cases} \tag{6.9}$$

§6.1.2 数值积分法

数值积分法的基本思想是先将问题 (6.1) 转化为积分方程

$$y(t_m) - y(t_n) = \int_{t_n}^{t_m} f(t, y(t))\,dt, \qquad m > n, \tag{6.10}$$

然后采用第五章介绍的数值积分公式将上式右端离散化，从而获得原初值问题的差分格式。如取 $m = n + 1$，应用公式 (5.1) 于 (6.10) 中的积分项，则得单支 $\theta-$ 方法

$$y_{n+1} = y_n + hf[\theta t_n + (1-\theta)t_{n+1}, \theta y_n + (1-\theta)y_{n+1}], \qquad \theta \in [0,1]. \tag{6.11}$$

特别 $\theta = \frac{1}{2}$ 时, 称之为**隐式中点公式**.

此外，若引入中间逼近值

$$Y_{1,\,n} = y_n + h(1-\theta)f[t_n + (1-\theta)h, Y_{1,\,n}],$$

则方法 (6.11) 也可以写为单级 Runge-Kutta 方法形式

$$\begin{cases} Y_{1,\,n} = y_n + h(1-\theta)f[t_n + (1-\theta)h, Y_{1,\,n}], \\ y_{n+1} = y_n + hf[t_n + (1-\theta)h, Y_{1,\,n}]. \end{cases} \tag{6.12}$$

在公式 (6.10) 中取 $m = n + k$ (k 为某给定正整数)，并将 Newton-Cotes 公式代入其中，则可得一类线性 k 步公式

$$y_{n+k} = y_n + h\sum_{i=0}^{k} \beta_i f_{n+i}, \qquad n = 0, 1, 2, \cdots, N-k, \tag{6.13}$$

其中,

$$\beta_i = \frac{(-1)^{k-i}}{i!(k-i)!}\int_0^k \prod_{j=0, j\neq i}^{k} (t-j)dt.$$

特别, 当取 $k = 2$ 时, 可得 Milne 公式

$$y_{n+2} = y_n + \frac{h}{3}(f_{n+2} + 4f_{n+1} + f_n). \tag{6.14}$$

利用公式 (6.13) 计算当前步 y_{n+k} 时, 其需要前 k 步逼近值 $y_n, y_{n+1}, \cdots, y_{n+k-1}$ 的有关消息，故称之为 k 步公式。关于这类公式更广泛的形式将在下段继续讨论。

§6.1.3 Taylor 展开法

Taylor 展开法的基本思想是首先构造一个关于真解及其有关信息的含参算子，将算子中诸项在某点处按 Taylor 展式展开，合并该展式中的同类项并截去余项，然后令诸同类项系数为零，由此即可确定出原算子中的全部或部分参数，从而获得一个或一类关于数值解的差分格式。

如定义算子

$$L[y(t), h] = \sum_{i=0}^{k} [\alpha_i y(t+ih) - h\beta_i y'(t+ih)], \tag{6.15}$$

其中 α_i, β_i 为待定系数。若 $y(t)$ 有 $p+2$ 阶连续导数，则上式右端诸项在 t 点可按 Taylor 展式展开为

$$L[y(t),h] = \sum_{j=0}^{p} C_j h^j y^{(j)}(t) + C_{p+1}h^{p+1}y^{(p+1)}(t) + \mathcal{O}(h^{p+2}), \tag{6.16}$$

其中

$$\begin{cases} C_0 = \sum_{i=0}^{k} \alpha_i, \\ C_1 = \sum_{i=1}^{k} (i\alpha_i - \beta_i) - \beta_0, \\ C_j = \dfrac{1}{j!}\sum_{i=1}^{k} i^{j-1}(i\alpha_i - j\beta_i), \quad j = 2,3,\cdots,p+1. \end{cases} \tag{6.17}$$

令

$$C_0 = C_1 = \cdots = C_p = 0, \tag{6.18}$$

且 $C_{p+1} \neq 0$，则得诸参数 α_i，β_i 使得算子

$$L[y(t),h] = C_{p+1}h^{p+1}y^{(p+1)}(t) + \mathcal{O}(h^{p+2})$$

的条件，由上式及 (6.16) 得

$$\sum_{i=0}^{k}[\alpha_i y(t+ih) - h\beta_i y'(t+ih)] = C_{p+1}h^{p+1}y^{(p+1)}(t) + \mathcal{O}(h^{p+2}). \tag{6.19}$$

在上式中取

$$t = t_n, \quad y_{n+i} \approx y(t_{n+i}), \quad y'(t_{n+i}) \approx f_{n+i},$$

并去掉余项，则得**线性 k 步方法**

$$\sum_{i=0}^{k}\alpha_i y_{n+i} = h\sum_{i=0}^{k}\beta_i f_{n+i}, \qquad \text{其中} \ \ \alpha_k \neq 0, \ \ \alpha_0^2 + \beta_0^2 \neq 0. \tag{6.20}$$

若 (6.18) 成立, 且 $C_{p+1} \neq 0$，则称方法 (6.20) 为p **阶相容的**(或简称为 p 阶的)，此时称式 (6.19) 的右端为该方法的**局部截断误差**, 而称 $C_{p+1}h^{p+1}y^{(p+1)}(t_n)$ 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** .

上述分析过程表明

定理 6.1 线性 k 步法 (6.20) 为 p 阶相容的充要条件是 (6.18) 成立, 且 $C_{p+1} \neq 0$.

在 (6.18) 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

$$y_{n+2}-(1+\alpha_0)y_{n+1}+\alpha_0 y_n=\frac{h}{12}[(5+\alpha_0)f_{n+2}+8(1-\alpha_0)f_{n+1}-(1+5\alpha_0)f_n], \tag{6.21}$$

且此时

$$C_4=-\frac{1}{24}(1+\alpha_0),\qquad C_5=-\frac{1}{360}(17+13\alpha_0).$$

因此, 当 $\alpha_0\neq -1$ 时, $C_4\neq 0$, 此时方法 (6.21) 为三阶的. 特别, 若取 $\alpha_0=-5$, 则得二步显式方法

$$y_{n+2}+4y_{n+1}-5y_n=2h(2f_{n+1}+f_n); \tag{6.22}$$

若取 $\alpha_0=0$, 则得三阶 Adams-Monlton 方法

$$y_{n+2}-y_{n+1}=\frac{h}{12}(5f_{n+2}+8f_{n+1}-f_n); \tag{6.23}$$

若取 $\alpha_0=-1$, 则 $C_4=0$, 但 $C_5=-\frac{1}{90}\neq 0$, 此时方法 (6.21) 即为四阶 Milne 方法

$$y_{n+2}-y_n=\frac{h}{3}(f_{n+2}+4f_{n+1}+f_n). \tag{6.24}$$

类似地, 若在 (6.18) 中取 $k=p=3$, $\alpha_3=-\alpha_2=1$, $\alpha_1=\alpha_0$, 则得三步 Adams 方法

$$\begin{aligned} y_{n+3} \;=\; & y_{n+2}+\frac{h}{12}\big[(5-12\beta_0)f_{n+3}+(8+36\beta_0)f_{n+2} \\ & -(1+36\beta_0)f_{n+1}+12\beta_0 f_n\big], \end{aligned} \tag{6.25}$$

且此时

$$C_4=\beta_0-\frac{1}{24},\qquad C_5=-\frac{4}{45}+\frac{3}{2}\beta_0.$$

因此, 当 $\beta_0\neq\frac{1}{24}$ 时, 方法 (6.25) 为三阶的; 当 $\beta_0=\frac{1}{24}$ 时, $C_4=0$, 但 $C_5\neq 0$, 此时该公式即为四阶 Adams 方法

$$y_{n+3}=y_{n+2}+\frac{h}{24}[9f_{n+3}+19f_{n+2}-5f_{n+1}+f_n]. \tag{6.26}$$

§6.2 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 $y_1, y_1, \ldots, y_{k-1}$. 而一般单步方法则可自起始完成计算. 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 (6.5) 与单支 $\theta-$ 方法 (6.11) 的局部截断误差均为

$$R_n := (\theta - \frac{1}{2})h^2 y''(t_n) + \mathcal{O}(h^3),$$

则当 $\theta \neq \frac{1}{2}$ 时, 其方法的相容阶仅等于 1; 当 $\theta = \frac{1}{2}$ 时, 其方法即分别为梯形法与隐式中点法, 其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计算方案, 这就形成了我们本节将要介绍的 Runge-Kutta 方法. 值得注意的是: 在上节, 我们已指出线性 $\theta-$ 方法与单支 $\theta-$ 方法可写成 Runge-Kutta 方法形式, 但其属低阶方法, 本节介绍的 Runge-Kutta 方法将涵盖高阶方法.

§6.2.1 方法的一般形式

Runge-Kutta 方法的一般形式为

$$\begin{cases} Y_{i,n} = y_n + h\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{j,n}), & i = 1, 2, \cdots, s, \\ y_{n+1} = y_n + h\sum_{j=1}^{s} b_j f(t_n + c_j h, Y_{j,n}), & n \geqslant 0, \end{cases} \tag{6.27}$$

其中诸系数 a_{ij}, b_j 及横标 c_j 为实数, $h > 0$ 为步长, $t_n = a + nh$, $Y_{i,n} \approx y(t_n + c_i h)$, $y_n \approx y(t_n)$。

对于方法 (6.27), 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 $A = (a_{ij}) \in R^{s\times s}, c = (c_1, c_2, \cdots, c_s)^T, b = (b_1, b_2, \cdots, b_s)^T \in R^s$. 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**

$$d_{n+1} := y(t_{n+1}) - \tilde{y}_{n+1} = \mathcal{O}(h^{p+1}), \qquad h \to 0, \tag{6.29}$$

其中 $\tilde{y}_{n+1}$ 由

$$\begin{cases} y_{i,n} = y(t_n) + h\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, y_{j,n}), \qquad i = 1, 2, \cdots, s, \\ \tilde{y}_{n+1} = y(t_n) + h\sum_{j=1}^{s} b_j f(t_n + c_j h, y_{j,n}) \end{cases} \tag{6.30}$$

确定. Runge-Kutta 方法的相容阶概念是我们构造具体方法的基本点.

§6.2.2 显式方法

显式 Runge-Kutta 方法与隐式 Runge-Kutta 方法相比较, 由于前者在每步无须解 s 个隐式方程, 因此其具有计算简单的特点. 具体显式 Runge-Kutta 方法的推导可采用 Taylor 展开法获得. 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

$$\begin{aligned} y(t_n + h) &= y(t_n) + \sum_{i=1}^{3} \frac{1}{i!} h^i y^{(i)}(t_n) + \mathcal{O}(h^4) \\ &= y(t_n) + h\hat{f}_n + \frac{1}{2}h^2 F_n + \frac{1}{6}h^3(F_n \hat{f}'_n + G_n) + \mathcal{O}(h^4) \end{aligned} \tag{6.31}$$

其中

$$\hat{f}_n = f(t_n, y(t_n)),\ \hat{f}'_n = \frac{\partial f(t_n, y(t_n))}{\partial y},\ F_n = \frac{\partial f(t_n, y(t_n))}{\partial t} + \hat{f}_n \cdot \hat{f}'_n,$$

$$G_n = \frac{\partial^2 f(t_n, y(t_n))}{\partial t^2} + 2\hat{f}_n \frac{\partial^2 f(t_n, y(t_n))}{\partial t \partial y} + \hat{f}_n^2 \frac{\partial^2 f(t_n, y(t_n))}{\partial y^2}.$$

以下恒设三级显式 Runge-Kutta 法满足

$$c_1 = 0, \qquad c_i = \sum_{j=1}^{i-1} a_{ij}, \quad i = 2, 3. \tag{6.32}$$

若方法自精确值 $y(t_n)$ 出发计算一步, 则有

$$\begin{cases} y_{1,n} = f(t_n, y(t_n)), \\ y_{2,n} = f(t_n + c_2 h, y(t_n) + h a_{21} y_{1,n}), \\ y_{3,n} = f(t_n + c_3 h, y(t_n) + h a_{31} y_{1,n} + h a_{32} y_{2,n}), \\ \tilde{y}_{n+1} = y(t_n) + h\sum_{j=1}^{3} b_j f(t_n + c_j h, y_{j,n}). \end{cases} \tag{6.33}$$

将 $y_{2,n},\ y_{3,n}$ 在 $(t_n, y(t_n))$ 处展开，并注意到条件 (6.32) 得

$$\begin{cases} y_{2,n} = \hat{f}_n + hc_2F_n + \dfrac{1}{2}h^2c_2^2G_n + \mathcal{O}(h^3), \\ y_{3,n} = \hat{f}_n + hc_3F_n + h^2(c_2a_{32}F_n\hat{f}'_n + \dfrac{1}{2}c_3^2G_n) + \mathcal{O}(h^3). \end{cases} \tag{6.34}$$

将上述所获 $y_{i,n}\ (i = 1, 2, 3)$ 的表达式代入 (6.33) 的最后一式得

$$\begin{aligned} \tilde{y}_{n+1} =& y(t_n) + h(b_1 + b_2 + b_3)\hat{f}_n + h^2(b_2c_2 + b_3c_3)F_n \\ &+ \frac{1}{2}h^3[2b_3c_2a_{32}F_n\hat{f}'_n + (b_2c_2^2 + b_3c_3^2)G_n] + \mathcal{O}(h^4). \end{aligned} \tag{6.35}$$

由相容阶定义及式 (6.31) 与 (6.35)，若要方法有三阶，则必有 (6.32) 及下列条件成立:

$$\begin{cases} b_1 + b_2 + b_3 = 1, \\ b_2c_2 + b_3c_3 = \dfrac{1}{2}, \\ b_2c_2^2 + b_3c_3^2 = \dfrac{1}{3}, \\ b_3c_2a_{32} = \dfrac{1}{6}. \end{cases} \tag{6.36}$$

取 {(6.32), (6.36)}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

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \tag{6.37}$$

取 {(6.32), (6.36)}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

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \tag{6.38}$$

仿上, 我们还可获得如下四级四阶方法

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{\sqrt{2}-1}{2} & \frac{2-\sqrt{2}}{2} & 0 & 0 \\ 1 & 0 & -\frac{\sqrt{2}}{2} & \frac{2+\sqrt{2}}{2} & 0 \\ \hline & \frac{1}{6} & \frac{2-\sqrt{2}}{6} & \frac{2+\sqrt{2}}{6} & \frac{1}{6} \end{array} \tag{6.39}$$

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 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.

§6.2.3 隐式方法

由上节可见, 随着方法的级数与阶数的增加, Runge-Kutta 方法利用 Taylor 展式来获得愈来愈加困难，不言而喻，直接利用 Taylor 展式来获取隐式方法将更加困难。为克服该困难, Butcher 于 1964 年引入了下列简化条件

$$\begin{aligned} B(p):&\quad \sum_{i=1}^{s} b_i c_i^{l-1} = \frac{1}{l}, \quad l = 1, 2, \cdots, p \\ C(\eta):&\quad \sum_{j=1}^{s} a_{ij} c_j^{l-1} = \frac{c_i^l}{l}, \quad i = 1, 2, \cdots, s; \ \ l = 1, 2, \cdots, \eta \\ D(\xi):&\quad \sum_{i=1}^{s} a_{ij} b_i c_i^{l-1} = \frac{1}{l} b_j (1 - c_j^l), \quad j = 1, 2, \cdots, s; \ \ l = 1, 2, \cdots, \xi, \end{aligned}$$

这里, $B(p)$, $C(\eta)$ 分别意味着系统 (6.1) 的真解 $y(t)$ 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 给出了 Runge-Kutta 方法的相容阶判据。

定理 6.3 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	$B(2s-2),\ \ C(s-1),\ \ D(s-1)$	$2s-2$

利用阶简化条件及定理 6.3, 我们可构造隐式 Runge-Kutta 方法. 如: 对于二级 Runge-Kutta 方法, 若有 $B(3)$, $C(1)$, $D(1)$ 成立, 即

$$\begin{cases} \displaystyle\sum_{i=1}^{2} b_i c_i^{l-1} = \frac{1}{l}, \quad l = 1,2,3, \\ \displaystyle\sum_{j=1}^{2} a_{ij} = c_i, \quad i = 1,2, \\ \displaystyle\sum_{i=1}^{2} a_{ij} b_i = b_j(1-c_j), \quad j = 1,2, \end{cases} \tag{6.40}$$

则由定理 6.3 可知这类二级方法至少是 3 阶相容的. 特别, 在 (6.40) 中若取 $c_1 = 0$, $a_{11} = \frac{1}{4}$, 则可解得

$$c_2 = \frac{3}{2}, \ \ b_1 = \frac{1}{4}, \ \ b_2 = \frac{3}{4}$$
$$a_{11} = \frac{1}{4}, \ \ a_{12} = -\frac{1}{4}, \ \ a_{21} = \frac{1}{4}, \ \ a_{22} = \frac{5}{12},$$

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 (6.40) 中取 $c_1 = \frac{1}{3}, a_{11} = \frac{5}{12}$, 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, 依据阶简化条件, 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 1, 2, 3 级各类方法列表如下:

表 6.2 2, 4, 6 阶 Gauss 方法

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{c|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{1}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{1}{2}+\frac{\sqrt{15}}{10} & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

表 6.3 1, 3, 5 阶 RadauIA 方法

$$\begin{array}{c|cc} 0 & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

$$\begin{array}{c|ccc} 0 & \frac{1}{9} & \frac{-1-\sqrt{6}}{18} & \frac{-1+\sqrt{6}}{18} \\ \frac{6-\sqrt{6}}{10} & \frac{1}{9} & \frac{88+7\sqrt{6}}{360} & \frac{88-43\sqrt{6}}{360} \\ \frac{6+\sqrt{6}}{10} & \frac{1}{9} & \frac{88+43\sqrt{6}}{360} & \frac{88-7\sqrt{6}}{360} \\ \hline & \frac{1}{9} & \frac{16+\sqrt{6}}{36} & \frac{16-\sqrt{6}}{36} \end{array}$$

表 6.4 1, 3, 5 阶 RadauIIA 方法

$$\begin{array}{c|cc} 1 & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}$$

表 6.5　2, 4, 6 阶 LobattoIIIA 方法

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

0	0	0	0
$\frac{1}{2}$	$\frac{5}{24}$	$\frac{1}{3}$	$-\frac{1}{24}$
1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

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	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{5}{12}$	$\frac{1}{12}$
	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{5}{12}$	$\frac{1}{12}$

表 6.6　2, 4, 6 阶 LobattoIIIB 型方法

0	$\frac{1}{2}$	0
1	$\frac{1}{2}$	0
	$\frac{1}{2}$	$\frac{1}{2}$

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
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

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
	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{5}{12}$	$\frac{1}{12}$

表 6.7　2, 4, 6 阶 LobattoIIIC 方法

0	$\frac{1}{2}$	$\frac{1}{2}$
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

0	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{6}$
$\frac{1}{2}$	$\frac{1}{6}$	$\frac{5}{12}$	$-\frac{1}{12}$
1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

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	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{5}{12}$	$\frac{1}{12}$
	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{5}{12}$	$\frac{1}{12}$

§6.3 数值算法理论

为讨论问题的广泛性，我们将 §6.1, §6.2 中各类算法写成统一的一般多步法形式

$$\sum_{i=0}^{k}\alpha_i y_{n+i}=h\varphi_f(t_n;y_n,y_{n+1},\cdots,y_{n+k};h),\quad n=0,1,2,\cdots,N-k, \tag{6.41}$$

其中 $h>0$ 为积分步长，$k\geqslant 1$ 是方法的步数，诸 α_i 为实常数, 且 $\alpha_k\neq 0$, $t_n=a+nh,\ Nh\leqslant b-a,\ y_n\approx y(t_n),\ \varphi_f:D_H\to R$ 是依赖于问题 (6.1) 的右函数 f 的映射，其中

$$D_H \ = \ \{(t_n;y_n,y_{n+1},\cdots,y_{n+k};h)|\ 0<h\leqslant H,\ a\leqslant t_n\leqslant b-kh,\ y_{n+i}\in R\},$$

$H>0$ 是适当选取的常数。恒设

1. 当 $f\equiv 0$ 时，$\varphi_f\equiv 0$；
2. 当 f 满足 (6.2) 时，φ_f 满足 Lipschitz 条件

$$\begin{aligned}&|\varphi_f(t_n;y_n,y_{n+1},\cdots,y_{n+k};h)-\varphi_f(t_n;z_n,z_{n+1},\cdots,z_{n+k};h)|\\ \leqslant\ & L_\varphi\sum_{i=0}^{k}|y_{n+i}-z_{n+i}|,\qquad 0<h\leqslant h_\varphi,\end{aligned} \tag{6.42}$$

这里 h_φ 和 L_φ 通常依赖于 (6.2) 中的 Lipschitz 常数 L.

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 $y_0,y_1,\cdots,y_{k-1}$ 出发，方程 (6.41) 可唯一地确定问题 (6.1) 的一个数值解序列 $\{y_n\}_{n=0}^N$. 方法 (6.41) 是一类非常广泛的方法，它几乎包含当今各种常用的各种常微分方程数值方法。如取 $\varphi_f=\sum\limits_{i=0}^{k}\beta_i f(t_n+ih,y_{n+i})$，则得线性多步法 (6.20)；若取 $\varphi_f=\sum\limits_{j=1}^{s}b_j f(t_n+c_jh,Y_{j,n})$，则得 s 级 Runge-Kutta 方法 (6.27). 此外，(6.41) 也包括 §6.4 中将要介绍的预校算法及迄今为止尚待开发的各种方法。本节将涉及该类方法的相容性、稳定性及收敛性。

§6.3.1 相容性

§6.1 和 §6.2 中局部截断误差及相容阶概念可统一定义如下：

定义 6.1 设 f 满足 (6.2)，若问题 (6.1) 的真解 $y(t)$ 满足

$$\sum_{i=0}^{k}\alpha_i y(t_{n+i})=h\varphi_f(t_n;y(t_n),y(t_{n+1}),\cdots,y(t_{n+k});h)+T_{n+k},$$
$$n=0,1,2,\cdots,N-k, \tag{6.43}$$

则称其余项 T_{n+k} 为方法 (6.41) 在点 t_{n+k} 处的局部截断误差。进一步，若 p 为满足

$$\max_{0\leqslant n\leqslant N-k}|T_{n+k}| = \mathcal{O}(h^{p+1}), \qquad h\to 0 \tag{6.44}$$

的最大正数，则称方法 (6.41) 是 p 阶相容的。特别若

$$\frac{1}{h}\max_{0\leqslant n\leqslant N-k}|T_{n+k}| \to 0, \qquad h\to 0, \tag{6.45}$$

则称方法 (6.41) 为相容的。

定理 6.4 方法 (6.41) 相容的充要条件是 $\rho(1)=0$，且下列等式关于 n 一致成立

$$\lim_{h\to 0}\varphi_f(t_n; y(t_n), y(t_{n+1}), \cdots, y(t_{n+k}); h) = \rho'(1)y'(t_n), \tag{6.46}$$

其中 $\rho(\xi)=\sum\limits_{i=0}^{k}\alpha_i\xi^i$ $(\xi\in C)$。

证 先证充分性：由于 f 满足 Lipschitz 条件及 $y(t)$ 在 $[a,b]$ 上连续可微，则由 (6.43) 及 Taylor 展式得

$$\begin{aligned} T_{n+k} &= \rho(1)y(t_n) + h[\rho'(1)y'(t_n) - \\ &\quad \varphi_f(t_n; y(t_n), y(t_{n+1}), \cdots, y(t_{n+k}); h)] + hR_{n,h}, \end{aligned} \tag{6.47}$$

其中

$$R_{n,h} = \sum_{i=0}^{k} i\alpha_i \int_0^1 [y'(t_n+i\theta h) - y'(t_n)]d\theta \to 0 \quad (h\to 0).$$

进而由 $\rho(1)=0$ 及 (6.47) 可知下式

$$\begin{aligned} \frac{1}{h}|T_{n+k}| &= |[\rho'(1)y'(t_n) - \varphi_f(t_n; y(t_n), y(t_{n+1}), \cdots, y(t_{n+k}); h)] \\ &\quad + R_{n,h}| \to 0 \ \ (h\to 0) \end{aligned}$$

关于 n 一致成立，即方法 (6.41) 是相容的。

必要性：若方法 (6.41) 是相容的，则由 (6.47)，当 $h\to 0$ 时有

$$\begin{aligned} \frac{1}{h}|T_{n+k}| &= |\frac{1}{h}\rho(1)y(t_n) + [\rho'(1)y'(t_n) \\ &\quad -\varphi_f(t_n; y(t_n), y(t_{n+1}), \cdots, y(t_{n+k}); h)]| \to 0 \end{aligned} \tag{6.48}$$

关于 n 一致成立。当方法 (6.41) 应用于问题

$$\begin{cases} y'(t) = 0, \qquad t\in[0,1], \\ y(0) = 1, \end{cases} \tag{6.49}$$

时，鉴于由 $f \equiv 0$ 有 $\varphi_f \equiv 0$，且问题 (6.49) 的真解为 $y(t) \equiv 1$，将其代入 (6.48) 得 $\rho(1) = 0$. 由 $\rho(1) = 0$ 及 (6.48) 即知 (6.46) 成立。

由定理 6.4 可直接推得

推论 6.1 线性多步法 (6.20) 相容的充要条件是

$$\rho(1) = 0, \qquad \rho'(1) = \sigma(1),$$

其中 $\sigma(\xi) = \sum_{i=0}^{k} \beta_i \xi^i$ $(\xi \in C)$.

推论 6.2 Runge-Kutta 方法 (6.27) 相容的充要条件是 $\sum_{j=1}^{s} b_j = 1$.

§6.3.2 稳定性

鉴于误差传播对计算影响的严重性，下面我们将讨论方法 (6.41) 的稳定性。

定义 6.2 方法 (6.41) 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，方法 (6.41) 的任一解序列 $\{y_n\}$ 与相应的扰动问题（设诸 ε_n 为任给扰动）

$$\begin{cases} \sum_{i=0}^{k} \alpha_i z_{n+i} = h[\varphi_f(t_n; z_n, \cdots, z_{n+k}; h) + \varepsilon_{n+k}], \quad n = 0, 1, \cdots, N-k, \\ z_j = y_j + \varepsilon_j, \quad j = 0, 1, \cdots, k-1, \end{cases}$$

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扰动对数值解的影响。显然，依据定义 6.2 来判断零稳定是困难的，下述结论可在一定程度上简化其判定。

定理 6.5 方法 (6.41) 为零稳定的充要条件是多项式 $\rho(\xi)$ 的每个根的模不超过 1，且模为 1 的根是单根。

方法 (6.41) 满足上述充要条件也称其符合**根条件**。由于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，即是零稳定的。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 给出了如下阶障碍结果。

定理 6.6 零稳定的线性 k 步法 (6.20) 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, 下面引入绝对稳定性概念。

定义 6.3 方法 (6.41) 称为是绝对稳定的, 若该方法应用于线性标量试验方程

$$y' = \lambda y, \quad \lambda \in C \tag{6.50}$$

时, 其解满足 $\lim\limits_{n\to\infty} y_n = 0$.

相应地, 称集合

$$\mathbb{S} = \{h\lambda \in C| \text{ 方法 (6.41) 应用于方程 (6.50) 时满足} \lim_{n\to\infty} y_n = 0\}$$

为方法 (6.41) 的绝对稳定域。

作为范例, 考虑线性多步法 (6.20) 和 Runge-Kutta 方法 (6.27), 当其应用于 (6.50) 时, 分别产生下列差分方程

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = \hat{h} \sum_{i=0}^{k} \beta_i y_{n+i} \tag{6.51}$$

及

$$y_{n+1} = [1 + \hat{h} b^T (I - \hat{h} A)^{-1} e] y_n, \tag{6.52}$$

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, 将其代入则分别得其特征方程为

$$\rho(\xi) = \hat{h}\sigma(\xi) \tag{6.53}$$

及

$$\xi = 1 + \hat{h} b^T (I - \hat{h} A)^{-1} e. \tag{6.54}$$

由此可知, 线性多步法 (6.20) 和 Runge-Kutta 方法 (6.27) 的绝对稳定域分别为

$$\mathbb{S}_{LM} = \{\hat{h} \in C| \text{ 方程 } \rho(\xi) = \hat{h}\sigma(\xi) \text{ 的根模满足} |\xi| < 1\}$$

和

$$\mathbb{S}_{RK} = \{\hat{h} \in C| \ |1 + \hat{h} b^T (I - \hat{h} A)^{-1} e| < 1\}.$$

§6.3.3 收敛性

前面我们仅讨论了数值方法每步产生的误差情况，即局部截断误差。本节将讨论方法的整体误差。当以方法 (6.41) 求解满足 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

$$\varepsilon_n := y(t_n) - y_n, \quad n = 0, 1, \cdots, N$$

为方法 (6.41) 的**整体截断误差**。

定义 6.5 方法 (6.41) 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

$$\max_{0 \leqslant j \leqslant k-1} |y(t_j) - y_j| = \mathcal{O}(h^p), \quad h \to 0,$$

则有

$$\varepsilon_n = \mathcal{O}(h^p), \quad h \to 0.$$

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

$$y_n \to y(t), \quad \text{当 } h \to 0,\ a + nh \to t,\ y_i \to y_0\ (0 \leqslant i \leqslant k-1).$$

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 (6.41) 相容，则其零稳定性与收敛性等价。

证 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

$$\max_{0 \leqslant n \leqslant N} |\varepsilon_n| = C\left(\max_{0 \leqslant j \leqslant k-1} |\varepsilon_j| + \frac{1}{h} \max_{0 \leqslant n \leqslant N-k} |T_{n+k}|\right), \quad 0 < h \leqslant h_0. \tag{6.55}$$

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 (6.45) 知方法是收敛的。

类似可证

定理 6.8 若方法 (6.41) 是 p 阶相容且零稳定的，则该方法是 p 阶收敛的。

由推论 6.1, 定理 6.1, 6.5, 6.7 及定理 6.8 可分别推得

推论 6.3 若线性多步法 (6.20) 满足根条件及条件

$$\rho(1)=0, \qquad \rho'(1)=\sigma(1),$$

则该方法是收敛的。

推论 6.4 若线性 k 步法 (6.20) 满足根条件及 (6.18), 且 $C_{p+1}\neq 0$，则该方法是 p 阶收敛的。

由于 Runge-Kutta 方法属单步法, 则必零稳定。据此, 推论 6.2 及定理 6.3, 6.7, 6.8 可分别推得

推论 6.3 若 Runge-Kutta 方法 (6.27) 满足 $\sum\limits_{j=1}^{s} b_j=1$, 则该方法是收敛的。

推论 6.4 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 阶收敛的。

§6.4 数值方法的有效实现

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量。对于方法的阶，我们一般要求其不超过原问题真解的可微次数。但是，我们也必须注意到：高阶方法的稳定性能通常较差，因此阶数很高的方法一般不适用。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。通常可利用误差主项来确定步长，即要求所取步长使方法的误差主项不超过预定精度 ε 且接近 ε，据此在计算中调整步长。与此同时，我们也要求步长符合稳定性要求。数值方法主要分为显式方法与隐式方法，从计算实现来说，显式方法易于处理。为此，本节主要介绍隐式方法的有效实现，其主要倚重二类技巧，一是迭代技巧，二是预估 - 校正技巧。以下，我们将应用这二种技巧于隐式线性多步法和隐式 Runge-Kutta 方法，从而实现隐式方法有效求解常微分方程初值问题。

§6.4.1 迭代技巧

隐式方法的实现问题实质上是非线性方程的求解问题，因此我们在第二章所学的迭代方法一般均可用于隐式方法有效实现。本节以 Jacobi 迭代法和 Newton 迭代法为例来说明隐式方法的实现。

对于隐式线性多步法

$$y_{n+k} = h\beta_k f(t_{n+k}, y_{n+k}) + \omega_n, \tag{6.56}$$

其中 $\beta_k \neq 0, \omega_n = \sum_{i=0}^{k-1} (h\beta_i f_{n+i} - \alpha_i y_{n+i})$，我们可采用如下 Jacobi 迭代格式

$$y_{n+k}^{(r+1)} = h\beta_k f(t_{n+k}, y_{n+k}^{(r)}) + \omega_n, \quad r = 0, 1, 2, \cdots, \tag{6.57}$$

其中迭代初值 $y_{n+k}^{(0)}$ 及计算启动值 $y_0, y_1, \cdots, y_{k-1}$ 可通过与该方法同阶的显式单步方法获得。若欲求解的问题 (6.1) 满足 Lipschitz 条件 (6.2)，则由定理 2.1 可知: 当步长满足

$$h < \frac{1}{|\beta_k| L}$$

时，其迭代格式 (6.57) 是收敛的, 且有事后误差估计

$$|y_{n+k}^{(r+1)} - y_{n+k}| \leqslant \frac{1}{1 - h\beta_k L} |y_{n+k}^{(r+1)} - y_{n+k}^{(r)}|.$$

因此，当步长足够小时，我们可用

$$|y_{n+k}^{(r+1)} - y_{n+k}^{(r)}| < \varepsilon$$

作为计算终止准则, 其中 ε 为预定精度. 当上准则成立时, 则计算终止, 且取当前步的数值解为 $y_{n+k} \approx y_{n+k}^{(r+1)}$。

隐式线性多步法 (6.56) 也可采用 Newton 迭代法来处理, 应用 Newton 迭代法于 (6.56) 有

$$y_{n+k}^{(r+1)} = y_{n+k}^{(r)} - \left[1 - h\beta_k \frac{\partial f(t_{n+k}, y_{n+k}^{(r)})}{\partial y}\right]^{-1} \left[y_{n+k}^{(r)} - h\beta_k f(t_{n+k}, y_{n+k}^{(r)}) - \omega_n\right], \quad r = 0, 1, 2, \cdots, \tag{6.58}$$

其计算终止准则可取为

$$|y_{n+k}^{(r+1)} - y_{n+k}^{(r)}| < \varepsilon \quad \text{或} \quad \left|y_{n+k}^{(r+1)} - h\beta_k f(t_{n+k}, y_{n+k}^{(r+1)}) - \omega_n\right| < \eta,$$

其中 ε, η 为预定精度. 当上准则之一成立时, 则计算终止, 且取当前步的数值解为 $y_{n+k} \approx y_{n+k}^{(r+1)}$。

对于隐式 Runge-Kutta 方法 (6.27), 其实现的关键是每步需解其 s 个内部级值

$$Y_{i,n} = y_n + h\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{j,n}), \qquad i = 1, 2, \cdots, s. \tag{6.59}$$

若采用 Jacobi 迭代法, 则得

$$Y_{i,n}^{(r+1)} = y_n + h\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{j,n}^{(r)}), \quad i = 1, 2, \cdots, s; \quad r = 0, 1, 2, \cdots, \tag{6.60}$$

其每步迭代初值可取为 $Y_{j,n}^{(0)} \approx y_n + c_j h f_n$ $(j = 1, 2, \cdots, s)$。又一次由定理 2.1 及 Lipschitz 条件可知: 当步长 h 满足

$$h < \frac{1}{L\sum\limits_{j=1}^{s} |a_{ij}|} \tag{6.61}$$

时, 迭代格式 (6.60) 收敛, 且有事后误差估计

$$|Y_{i,\,n}^{(r+1)} - Y_{i,\,n}| \leqslant \frac{1}{1 - hL\sum\limits_{j=1}^{s} |a_{ij}|} |Y_{i,\,n}^{(r+1)} - Y_{i,\,n}^{(r)}|, \qquad i = 1, 2, \cdots, s.$$

因此, 当步长足够小时, 我们可用

$$|Y_{i,\,n}^{(r+1)} - Y_{i,\,n}^{(r)}| < \varepsilon$$

作为计算终止准则, 其中 ε 为预定精度. 当上准则成立时, 则取当前步的级值为 $Y_{i,\,n} \approx Y_{i,\,n}^{(r+1)}$ $(i = 1, 2, \cdots, s)$, 将其代入 (6.27) 的第二式即获当前步的数值解 y_{n+1}。

若应用 Newton 迭代法到 (6.59), 则得

$$Y_{i,\,n}^{(r+1)} = Y_{i,\,n}^{(r)} - \left[1 - ha_{ii}\frac{\partial f(t_n + c_i h,\, Y_{i,\,n}^{(r)})}{\partial y}\right]^{-1} \left[Y_{i,\,n}^{(r)} - y_n - h\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{j,\,n}^{(r)})\right],$$
$$i = 1, 2, \cdots, s; \quad r = 0, 1, 2, \cdots, \tag{6.62}$$

其每步迭代初值取为 $Y_{j,n}^{(0)} \approx y_n + c_j h f_n$ $(j = 1, 2, \cdots, s)$, 计算终止准则可取为

$$|Y_{i,\,n}^{(r+1)} - Y_{i,\,n}^{(r)}| < \varepsilon \ \text{ 或 } \ \left|Y_{i,\,n}^{(r)} - y_n - h\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_{j,\,n}^{(r)})\right| < \eta,$$

其中 ε, η 为预定精度. 当上准则之一成立时, 则取当前步的级值为 $Y_{i,\,n} \approx Y_{i,\,n}^{(r+1)}$ $(i = 1, 2, \cdots, s)$, 将其代入 (6.27) 的第二式即获当前步的数值解 y_{n+1}。

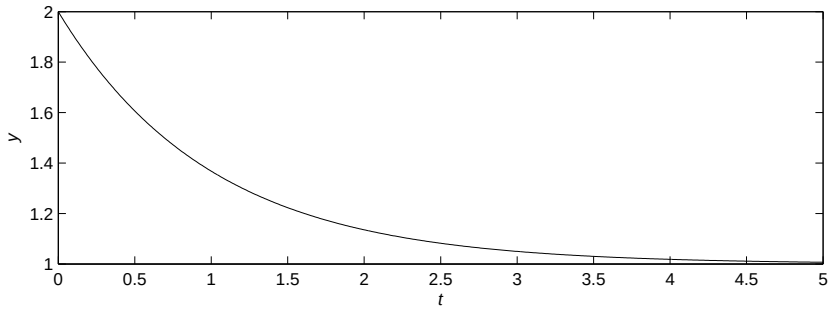


图 6.1 四阶 Gauss 方法应用于初值问题 (6.63) 的数值解.

例 6.1 试取步长 $h = 0.01$, 应用四阶 Gauss 方法计算初值问题

$$\begin{cases} y'(t) = y(t)\ln[1 + y(t)] - \exp(-t)[1 + (1 + \exp(t))\ln(2 + \exp(-t))], \quad t \in [0, 5], \\ y(0) = 2, \end{cases} \tag{6.63}$$

并给出所获数值解的最大整体误差 $\max\limits_{0\leqslant n\leqslant 500} |y(t_n) - y_n|$, 其中初值问题 (6.63) 的精确解为 $y(t) = 1 + \exp(-t)$.

解 取步长 $h = 0.01$, 应用四阶 Gauss 方法于初值问题 (6.63). 计算过程采用 Newton 迭代技巧, 每步迭代初值取为 $Y_{j,n}^{(0)} \approx y_n + c_j h f_n$ $(j = 1, 2)$, 其中 $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$, $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$。据此得该初值问题的数值解 (见图 6.1), 其最大整体误差 $\max\limits_{0\leqslant n\leqslant 500} |y(t_n) - y_n| = 7.8440e - 008$。

§6.4.2 预估 - 校正技巧

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, 我们可用与迭代公式同阶的显式公式先计算迭代初值 (并称此显式公式为预估公式), 然后用迭代公式 (称之为校正式) 修正一次, 即可获得其数值解。此方法称为**预估 - 校正方法**。如我们以二步三阶显式公式 (6.22) 作为预估式, 而以三阶 Adams-Moulton 公式 (6.23) 作为校正式, 则可得预校算法

$$\begin{cases} \text{预估}: \tilde{y}_{n+2} = -4y_{n+1} + 5y_n + 2h[2f_{n+1} + f_n], \\ \text{校正}: y_{n+2} = y_{n+1} + \dfrac{h}{12}[5f(t_{n+2}, \tilde{y}_{n+2}) + 8f_{n+1} - f_n]. \end{cases} \tag{6.64}$$

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 (6.64) 中，若取 $y_{n+1} \approx y(t_{n+1}), y_n \approx y(t_n)$ ，则根据 (6.19) 有

$$y(t_{n+2}) - \tilde{y}_{n+2} \approx \frac{1}{6}h^4 y^{(4)}(t_n),$$
$$y(t_{n+2}) - y_{n+2} \approx -\frac{1}{24}h^4 y^{(4)}(t_n).$$

由上两式可得其事后误差估计式

$$y(t_{n+2}) - \tilde{y}_{n+2} \approx \frac{1}{6}(y_{n+2} - \tilde{y}_{n+2}), \tag{6.65}$$

$$y(t_{n+2}) - y_{n+2} \approx -\frac{1}{24}(y_{n+2} - \tilde{y}_{n+2}), \tag{6.66}$$

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 (6.65) 及 (6.66) 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

$$\begin{cases} \text{预估：} & p_{n+2} = -4y_{n+1} + 5y_n + 2h(2f_{n+1} + f_n), \\ \text{改进：} & m_{n+2} = p_{n+2} + \dfrac{1}{6}(c_{n+1} - p_{n+1}), \\ & F_{n+2} = f(t_{n+2}, m_{n+2}), \\ \text{校正：} & c_{n+2} = y_{n+1} + \dfrac{h}{12}(5F_{n+2} + 8f_{n+1} - f_n), \\ \text{改进：} & y_{n+2} = c_{n+2} - \dfrac{1}{24}(c_{n+2} - p_{n+2}), \\ & f_{n+2} = f(t_{n+2}, y_{n+2}). \end{cases} \tag{6.67}$$

上述方案中，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，在启动方案 (6.67) 时，我们可取 $c_1 - p_1 = 0$。

例 6.2 试取步长 $h = 0.01$, 应用预估 - 改进 - 校正方法 (6.67) 计算初值问题 (6.63), 并给出其最大整体误差 $\varepsilon := \max\limits_{0 \leqslant n \leqslant 500} |y(t_n) - y_n|$.

解 取步长 $h = 0.01$, 应用预估 - 改进 - 校正方法 (6.67) 于初值问题 (6.63), 计算初值由三阶 Kutta 方法 (6.38) 给出。据此得该初值问题的数值解（见图 6.2），其最大整体误差 $\max\limits_{0 \leqslant n \leqslant 500} |y(t_n) - y_n| = 6.3041e-006$。

在上面，我们仅以线性多步法为例来说明预估 - 校正算法的构造。事实上，我们也可以用 Runge-Kutta 方法来构造预估 - 校正算法。

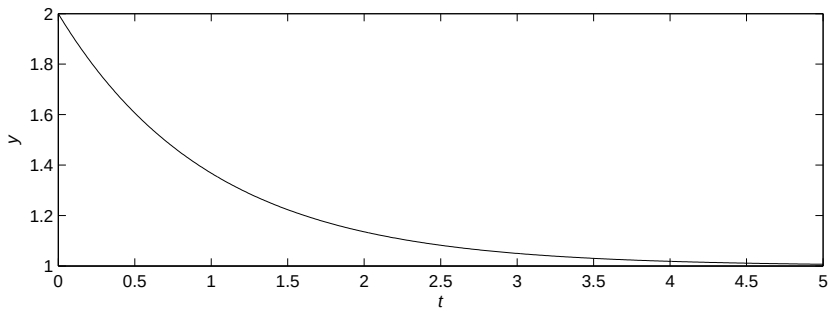


图 6.2 预估 - 改进 - 校正方法 (6.67) 应用于初值问题 (6.63) 的数值解.

§6.5 微分方程组的数值处理

§6.5.1 一阶微分方程组

在实际工程技术与科学研究工作中，我们常常会遇到高阶微分方程（组）初值问题。这些问题一般可化为一阶 m 维微分方程组初值问题

$$\begin{cases} y'(t) = f(t, y(t)), \quad t \in [a, b], \\ y(a) = y_0, \quad y_0 \in R^m, \end{cases} \tag{6.68}$$

其中

$$y(t) = [y_1(t),\ y_2(t),\ \cdots,\ y_m(t)]^T,\ \ y'(t) = \left[y_1'(t),\ y_2'(t),\ \cdots,\ y_m'(t)\right]^T,$$

$$f(t, y(t)) = [f_1(t, y(t)),\ f_2(t, y(t)),\ \cdots,\ f_m(t, y(t))]^T \in R^m.$$

如: 对与 m 阶非线性标量微分方程初值问题

$$\begin{cases} x^{(m)}(t) = F(t;\ x^{(m-1)}(t),\ \cdots, x'(t),\ x(t)) \quad t \in [a, b], \\ x^{(i-1)}(a) = x_{i-1}, \quad i = 1, 2, \ldots, m, \end{cases} \tag{6.69}$$

我们通过做变换

$$x^{(i)}(t) = y_{i+1}(t), \quad i = 0, 1, \cdots, m-1,$$

可将 (6.69) 化为一阶 m 维微分方程组问题 (6.68)，这里

$$f(t, y(t)) = \begin{pmatrix} y_2(t) \\ y_3(t) \\ \vdots \\ y_m(t) \\ F(t;\ y_m(t),\ y_{m-1}(t),\ \cdots,\ y_1(t)) \end{pmatrix}, \quad y_0 = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{m-1} \end{pmatrix}$$

微分方程组 (6.68) 的数值处理直接应用先前介绍的各种数值方法到该方程组的每个方程即可。此外，先前的数值分析理论也完全可拓广到方程组情形。

例 6.3 试用四阶经典 Runge-Kutta 方法 (6.41) 求解初值问题

$$\begin{cases} \dfrac{du}{dt} = f_1(t, u(t), v(t)), \quad t \in [a, b], \\ \dfrac{dv}{dt} = f_2(t, u(t), v(t)), \quad t \in [a, b], \\ u(t_0) = u_0, \quad v(t_0) = v_0. \end{cases} \tag{6.70}$$

解 将四阶经典 Runge-Kutta 方法应用到本问题得

$$\begin{aligned} u_{n+1} =& u_n + \frac{h}{6}[f_1(t_n, U_{1,\,n}, V_{1,\,n}) + 2f_1(t_n + \frac{h}{2}, U_{2,\,n}, V_{2,\,n}) \\ &+ 2f_1(t_n + \frac{h}{2}, U_{3,\,n}, V_{3,\,n}) + f_1(t_n + h, U_{4,\,n}, V_{4,\,n})], \end{aligned} \tag{6.71}$$

$$\begin{aligned} v_{n+1} =& v_n + \frac{h}{6}[f_2(t_n, U_{1,\,n}, V_{1,\,n}) + 2f_2(t_n + \frac{h}{2}, U_{2,\,n}, V_{2,\,n}) \\ &+ 2f_2(t_n + \frac{h}{2}, U_{3,\,n}, V_{3,\,n}) + f_2(t_n + h, U_{4,\,n}, V_{4,\,n})], \end{aligned} \tag{6.72}$$

其中

$$\begin{cases} U_{1,\,n} = u_n, \quad V_{1,\,n} = v_n, \\ U_{2,\,n} = u_n + \dfrac{h}{2} f_1(t_n, U_{1,\,n}, V_{1,\,n}), \quad V_{2,\,n} = v_n + \dfrac{h}{2} f_2(t_n, U_{1,\,n}, V_{1,\,n}), \\ U_{3,\,n} = u_n + \dfrac{h}{2} f_1(t_n + \dfrac{h}{2}, U_{2,\,n}, V_{2,\,n}), \quad V_{3,\,n} = v_n + \dfrac{h}{2} f_2(t_n + \dfrac{h}{2}, U_{2,\,n}, V_{2,\,n}), \\ U_{4,\,n} = u_n + \dfrac{h}{2} f_1(t_n + \dfrac{h}{2}, U_{3,\,n}, V_{3,\,n}), \quad V_{4,\,n} = v_n + \dfrac{h}{2} f_2(t_n + \dfrac{h}{2}, U_{3,\,n}, V_{3,\,n}). \end{cases} \tag{6.73}$$

上述公式可按如下步骤完成：首先将原问题初值代入 (6.73) 解出当前步的 $U_{i,\,n}$, $V_{i,\,n}$ $(i = 1, 2, 3, 4)$，然后将其代入 (6.71),(6.72) 可分别求出 u_1, v_1 ；以 u_1, v_1 作为第 2 个计算步的初值重复上述步骤可求出 u_2, v_2；依次类推即可求出原问题的相继数值解序列 $\{(u_n, v_n)\}$.

若引入记号

$$y(t) = [u(t),\ v(t)]^T, \quad f(t, y(t)) = [f_1(t, u(t), v(t)),\ f_2(t, u(t), v(t))]^T,$$

$$y_n = [u_n,\ v_n]^T, \quad Y_{i,\,n} = [U_{i,\,n},\ V_{i,\,n}]^T, \quad i = 1, 2, 3, 4,$$

$$f(t, Y_{i,n}) = [f_1(t, U_{i,\,n}, V_{i,\,n}),\ f_2(t, U_{i,\,n}, V_{i,\,n})]^T, \quad i = 1, 2, 3, 4,$$

则初值问题 (6.70) 可写为

$$\begin{cases} y'(t) = f(t, y(t)), \quad t \in [a, b], \\ y(a) = y_0, \end{cases} \tag{6.74}$$

其求解方法可写为

$$\begin{cases} Y_{1,\,n} &= y_n, \\ Y_{2,\,n} &= y_n + \frac{h}{2} f(t_n, Y_{1,\,n}), \\ Y_{3,n} &= y_n + \frac{h}{2} f(t_n + \frac{h}{2}, Y_{2,\,n}), \\ Y_{4,\,n} &= y_n + \frac{h}{2} f(t_n + \frac{h}{2}, Y_{3,\,n}), \\ y_{n+1} &= y_n + \frac{h}{6}[f(t_n, Y_{1,\,n}) + 2f(t_n + \frac{h}{2}, Y_{2,\,n}) \\ & \quad +2f(t_n + \frac{h}{2}, Y_{3,\,n}) + f(t_n + h, Y_{4,\,n})], \end{cases} \tag{6.75}$$

由此可见, 问题 (6.70) 的求解方法 (6.71)-(6.73) 也可视为四阶经典 Runge-Kutta 方法直接应用于向量方程 (6.74).

习 题 六

6.1 试用三种方法导出线性二步方法

$$y_{n+2} = y_n + 2hf_{n+1}.$$

6.2 用 Taylor 展开法求三步四阶方法类, 并确定三步四阶显式方法。

6.3 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, 并给出其截断误差主项。

6.4 试用显式 Euler 法及改进的 Euler 法

$$y_{n+1} = y_n + \frac{h}{2}[f(t_n, y_n), f(t_{n+1}, y_n + hf_n)]$$

计算初值问题 (取步长 $h = 0.2$)

$$\begin{cases} y'(t) = y(t) - \dfrac{2t}{y(t)}, \quad t \in [0, 1] \\ y(0) = 1, \end{cases}$$

并比较两者误差。

6.5 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。

6.6 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 $B(2), C(2)$ 导出一个二阶 Runge-Kutta 方法, 并指出该方法的相容阶。

6.7 给出题 6.5 中 $\alpha = 1$ 时的公式的绝对稳定域。

6.8 试以方法类 (6.25) 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。

6.9 指出 Heun 方法

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}$$

的相容阶，并给出由该方法以步长 h 计算初值问题 (6.70) 的步骤。

6.10 试取步长 h=0.01, 分别应用 Newton 迭代技巧于三阶 Adams-Monlton 方法和三阶 Radau IIA 方法, 计算初值问题

$$\begin{cases} y'(t) = \dfrac{y^2(t) + y(t)}{t}, \quad t \in [1, 5], \\ y(1) = -2, \end{cases}$$

并比较其最大整体误差 $\varepsilon := \max\limits_{0 \leqslant n \leqslant 400} |y(t_n) - y_n|$.

参考文献

[1] 李岳生, 黄友谦. 数值逼近. 北京：人民教育出版社, 1978

[2] 李寿佛. 刚性微分方程算法理论. 长沙：湖南科技出版社, 1997

[3] 李荣华, 冯果忱. 微分方程数值解法. 北京：高等教育出版社, 1989

[4] 王能超. 数值分析简明教程. 北京：高等教育出版社, 1985

[5] 李庆扬, 王能超, 易大义. 数值分析. 武汉：华中理工大学出版社, 1981

[6] Hairer E, Nϕsett S P, Wanner G. Solving Ordinary Differential Equations I: Nonstiff Problems. Heidelberg：Spriner-Verlag, 1993

[7] Hairer E, Wanner G. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Heidelberg：Spriner-Verlag, 1996

习题答案

习题一

1.2　提示：利用不等式

$$\max_{1\leqslant i\leqslant n}|x_i|\leqslant\|x\|_p\leqslant\sqrt[p]{p}\max_{1\leqslant i\leqslant n}|x_i|$$

可证 $\|x\|_\infty=\max\limits_{1\leqslant i\leqslant n}|x_i|$. 此外，若记

$$\mu=\max_{1\leqslant i\leqslant n}\sum_{j=1}^{n}|a_{i,j}|=\sum_{j=1}^{n}|a_{i_0,j}|\quad\text{及}\quad x^{(0)}=\left(x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)}\right)^T,\quad\text{其中}$$

$$x_n^{(0)}=\begin{cases}|a_{i_0,j}|/a_{i_0,j}, & a_{i_0,j}\neq 0,\\ 1, & a_{i_0,j}=0,\end{cases}$$

则由

$$\|Ax^{(0)}\|_\infty\leqslant\|A\|_\infty\leqslant\mu$$

可证 $\|A\|_\infty=\mu$。

1.4　其解为 $y_n=8[(-1)^{n-1}+\frac{1}{2^n}]-n$. 提示: 其方程有特解 $y_n^*=-n$.

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　$x\approx 0.64118$

2.2　(1) $x_{k+1}=\frac{\cos x_k+\sin x_k}{4}$; (2) $x_{k+1}=\frac{1}{\ln 2}\ln(4-x_k)$.

2.3　(1) 收敛;　(2) 收敛;　(3) 发散。

根的近似值 $x^*=1.4655$.

2.4　$x_{k+1}=-\frac{1}{2}(\varphi(x_k)-3x_k)$.

2.5　仿定理 2.1 的证明方法。

2.6　3.63。

2.7　$\lim\limits_{k\to\infty}\frac{|x_{k+1}-2|}{|x_k-2|}=\frac{1}{2}$。

2.8　$x_{k+1}=x_k-\frac{x_{k+1}^2-a}{3x_k^2}$,　平方收敛。

习题三

3.1 (1) 系数矩阵强对角占优，故 Jacobi 迭代法和 Gauss-Seidel 迭代法解此方程组收敛；

(2) 初值 $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$,

Jacobi 迭代法： $x_1^{(18)} = -3.999996$, $x_2^{(18)} = 2.999974$, $x_3^{(18)} = 2.0$,

Gauss-Seidel 迭代法： $x_1^{(8)} = -4.000033$, $x_2^{(8)} = 2.999983$, $x_3^{(8)} = 2.000002$.

3.2 迭代矩阵 $B = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 1 \end{bmatrix}$, 当 $|\frac{a_{12}a_{21}}{a_{11}a_{22}}| < 1$ 时, $\rho(B) < 1$.

3.3 取初值 $x_1^{(0)} = x_2^{(0)} = 0$ 得 $x_1^{(16)} = 1.000017$, $x_2^{(16)} = -0.9999913$.

3.4 $\frac{(1+10^4)^2}{10^4-1} \approx 10^4$.

3.5 $x_1 = 2$, $x_2 = 1$, $x_3 = 0.5$.

3.6

$$x_1 = 0.8333333, \quad x_2 = 0.6666666,$$

$$x_3 = 0.4999999, \quad x_4 = 0.3333333, \quad x_5 = 0.1666666.$$

3.7
$$\begin{bmatrix} 1 & & \\ 2 & 1 & \\ 3 & -1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 4 & 8 \\ & 10 & -32 \\ & & -76 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_n \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 7 \end{bmatrix}.$$

3.8 30.

习题四

4.1 $f(0.472) \approx 0.4955616$ （取插值节点 0.46, 0.47, 0.48）。

4.2 $\sin(1.609) \approx 0.99927$, $|R_2(1.609)| \leqslant 0.000199$.

4.3 (1) $f(x) = x^k$, $R_{n+1}(x) \equiv 0$; (2) $f(t) = (t-x)^k$, 利用 (1) 即可得证。

4.4 $\Delta^2 y_n = 2^n$, $\delta^4 y_n = 2^{n-2}$.

4.5 $H(x) = (x-2)^2(x^2-2x-1)$, $R_5(x) = \frac{f^{(5)}(\xi)}{5\,!}(x-1)^2(x-2)^2(x-3)$.

4.6 令 $R_3(x) = k(x)(x-x_k)^2(x-x_{k+1})^2$.

4.7 $s(x) = x^3 + x + 1$.

4.8 $y = 0.973 + 0.050x^2$.

习题五

5.1 $\int_{-1}^{1} f(x)ds \approx \frac{2}{3}\left[f(0)+f(\frac{\sqrt{2}}{2})+f(-\frac{\sqrt{2}}{2})\right]$.

5.2 $\int_{b}^{a} f(x)ds \approx \frac{2h}{45}\left[7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)\right]$, 其中

$$h=\frac{b-a}{4},\quad x_i=a+ih,\quad i=0,1,2,3,4.$$

该公式具有 5 次代数精度, 且其余项为

$$R(f)=\frac{1}{5!}\int_{b}^{a} f^{(5)}(\xi)\prod_{i=0}^{4}(x-x_i)dx,\quad \xi\in(a,b).$$

5.3 $I_T\approx 1.6355,\quad I_S\approx 1.6360.$

5.4 $I\approx 0.74632.$

5.5 $I\approx 0.1571.$

5.6 $\int_{-1}^{1} f(x)ds \approx f(-\frac{1}{\sqrt{3}})+f(\frac{1}{\sqrt{3}}),\quad I\approx 1.3987.$

习题六

6.2 三步四阶方法的系数为:

$$\begin{aligned}
&\alpha_0=-b,\quad \beta_0=\frac{1}{24}(1+a+9b),\\
&\alpha_1=a+b,\quad \beta_1=\frac{1}{24}(-5-13a+19b),\\
&\alpha_2=-(1+a),\quad \beta_2=\frac{1}{24}(19-13a-5b),\\
&\alpha_3=1,\quad \beta_3=\frac{1}{24}(9+a+b),
\end{aligned}$$

其中 $C_{p+1}=-\frac{1}{720}(19+11a+19b)$, 且 $a+b=-9$ 时可得其显式公式。

6.3 $11y_{n+3}-18y_{n+2}+9y_{n+1}-2y_n=6hf_{n+3}$, 其截断误差主项为 $R_n=-\frac{3}{2}h^4y^{(4)}(t_n)$.

6.4 其计算结果如下 (问题精确解为 $y(t)=\sqrt{2t+1}$.)

t_n	数值解		误差 $\lvert y(t_n)-y_n\rvert$	
	Euler 法	改进的 Euler 法	Euler 法	改进的 Euler 法
0.2	1.2000	1.1867	0.168×10^{-1}	0.35×10^{-2}
0.4	1.3733	1.3484	0.317×10^{-1}	0.68×10^{-2}
0.6	1.5315	1.4938	0.483×10^{-1}	0.106×10^{-1}
0.8	1.6811	1.6279	0.687×10^{-1}	0.155×10^{-1}
1.0	1.8270	1.7542	0.950×10^{-1}	0.222×10^{-1}

6.5 其零稳定的条件为 $|\alpha| < 1$.

6.6

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

且通过验证 $B(4), C(2), D(2)$ 成立, 由此可推知其相容阶为 4。

6.7 $S = \left\{ \bar{h} \in C \ : \ \left| \frac{1+\bar{h}}{1-\bar{h}} \right| < 1 \right\}$.

6.8

$$\begin{cases} \text{预估} & P_{n+3} = y_{n+2} + \dfrac{h}{12}(23f_{n+2} - 16f_{n+1} + 5f_n) \\ \text{改进} & m_{n+3} = P_{n+3} + \dfrac{3}{8}(C_{n+2} - P_{n+2}) \\ & F_{n+3} = f(t_{n+3}, m_{n+3}) \\ \text{校正} & C_{n+3} = \dfrac{1}{11}(18y_{n+2} - 9y_{n+1} + 2y_n) + \dfrac{6h}{11}F_{n+3} \\ \text{改进} & y_{n+3} = C_{n+3} - \dfrac{3}{22}(C_{n+3} - P_{n+3}) \\ & f_{n+3} = f(t_{n+3}, y_{n+3}). \end{cases}$$

6.9 相容阶为 3； 提示：参照例 6.3。

6.10 三阶 Adams-Monlton 方法和三阶 Radau IIA 方法的最大整体误差分别为

$$\varepsilon_{AM} = 1.0879e-006, \quad \varepsilon_{RIIA} = 5.7191e-008.$$