

21 世纪计算机科学与技术系列教材（本科）

数值分析

主 编 韩国强

副主编 林伟健 陈大正

华南理工大学出版社
• 广州 •

图书在版编目 (CIP) 数据

数值分析/韩国强主编. —广州: 华南理工大学出版社, 2005. 3
(21 世纪计算机科学与技术系列教材 (本科))

ISBN 7- 5623- 2182- 5

I. 数… II. 韩… III. 数值计算 - 高等学校 - 教材 IV. O241

中国版本图书馆 CIP 数据核字 (2004) 第 136326 号

总发行: 华南理工大学出版社(广州五山华南理工大学 17 号楼, 邮编 510640)

发行部电话: 020-87113487 87111048(传真)

Email: scut202@scut.edu.cn

<http://www.scutpress.com>

责任编辑: 欧建岸(ouja2@163.com)

印刷者: 湛江日报社印刷厂

开本: 787×960 1/16 印张: 13 字数: 262 千

版次: 2005 年 3 月第 1 版第 1 次印刷

印数: 1—3000 册

定价: 21.50 元

版权所有 盗版必究

前 言

在科学技术特别是计算机科学技术的发展过程中,遇到了大量的数值计算问题.数值分析就是研究解决这些数值计算问题的学科,也是一门实用性很强,内容丰富,有自身理论体系的课程.

根据多年从事《数值分析》教学的经验,按照学生学习的认知规律,我们精心构造了本教材的体系.在叙述本课程的内容时,采用由简单到复杂,由特殊到一般的叙述方法.在介绍数值分析基础理论的同时,也给出了数值计算的实例.为了方便学生在计算机上进行一些数值计算实验,我们对每一种数值计算方法都给出了算法描述.

本书分为9章.第1章介绍了数值计算中误差分析的基本理论,目的在于告诫读者在进行数值计算时要重视误差分析.第2章介绍了代数插值的理论和方法.第3章介绍了最小二乘法和一些数据拟合法.第4章介绍了数值积分的常用方法.第5章介绍了解线性代数方程组的直接法和对方程组求解的误差分析理论知识.第6章介绍了解线性代数方程组的迭代法.第7章介绍了解非线性代数方程和非线性代数方程组的方法.第8章介绍了矩阵的特征值和特征向量的一些求法.第9章介绍了常微分方程初值问题的数值解法.本书讲授的课时大约为56学时,实验时数为8学时.

我们希望读者在学习本课程时,不仅要注意掌握各种数值计算方法的基本思想、推导过程、人工计算过程和在计算机上如何实现,而且也要重视误差分析、收敛性的判别和稳定性分析的基本理论.为了更好地掌握本课程的内容,应做一定数量的数值计算和理论分析的练习,还应在计算机上进行一些数值计算实验.

本书由华南理工大学计算机科学与工程学院副院长、博士生导师韩国强教授主编,由华南理工大学林伟健副教授和中山大学陈大正副教授任副主编.第1~4章由林伟健编写,第5~6章由韩国强编写,第7~9章由陈大正编写.

本书可以作为高等学校本科生数值分析或计算方法课程教材,也可以作为研究生教材和教师、工程技术人员参考书.

在本书编写的过程中,得到华南理工大学出版社和广东省计算机学会的大力支持,在此表示感谢.

限于时间和编者的水平,书中难免有不妥之处,敬请同行专家和读者批评指正.

编著者
2005年2月

目 录

| | |
|---|-----|
| 1 误差 | () |
| 1.1 误差的来源 | () |
| 1.2 误差、误差限和有效数字 | () |
| 1.3 相对误差和相对误差限 | () |
| 1.4 数值运算中的误差估计 | () |
| 1.5 数值计算中应注意的一些问题 | () |
| 2 代数插值与数值微分 | () |
| 2.1 线性插值与二次插值 | () |
| 2.1.1 线性插值 | () |
| 2.1.2 二次插值 | () |
| 2.2 n 次插值的 Lagrange 形式和 Newton 形式 | () |
| 2.2.1 n 次插值函数的 Lagrange 形式 | () |
| 2.2.2 n 次插值函数的 Newton 形式 | () |
| 2.3 分段线性插值 | () |
| 2.4 Hermite 插值 | () |
| 2.4.1 三次 Hermite 插值 | () |
| 2.4.2 $2n+1$ 次 Hermite 插值 | () |
| 2.5 三次样条插值 | () |
| 2.6 数值微分 | () |
| 2.6.1 使用 n 次插值函数求导数 | () |
| 2.6.2 使用三次样条插值函数求导数 | () |
| 3 数据拟合 | () |
| 3.1 单变量数据拟合及最小二乘法 | () |
| 3.2 多变量数据拟合 | () |
| 3.3 非线性数据线性化 | () |
| 3.4 正交多项式拟合 | () |
| 4 数值积分 | () |

| | | |
|-------|---|-----|
| 4.1 | 梯形求积公式、抛物线求积公式和 Newton - Cotes 求积公式 | () |
| 4.1.1 | 梯形求积公式 | () |
| 4.1.2 | Simpson 求积公式 | () |
| 4.1.3 | Newton - Cotes 求积公式 | () |
| 4.2 | 求积公式的代数精确度 | () |
| 4.3 | 梯形求积公式和 Simpson 求积公式的误差估计 | () |
| 4.4 | 复化求积公式 | () |
| 4.4.1 | 复化梯形求积公式及其误差估计 | () |
| 4.4.2 | 复化 Simpson 求积公式及其误差估计 | () |
| 4.5 | 自动选取步长梯形法 | () |
| 4.6 | 数值方法中的加速收敛技巧——Richardson 外推算法 | () |
| 4.7 | Romberg 求积法 | () |
| 4.7.1 | Romberg 序列的推导 | () |
| 4.7.2 | Romberg 求积法的计算过程 | () |
| 4.8 | Gauss 型求积公式 | () |
| 4.8.1 | Gauss - Legendre(勒让得)求积公式 | () |
| 4.8.2 | Gauss - Leguerre(拉盖尔)求积公式 | () |
| 4.8.3 | Gauss - Hermite 求积公式 | () |
| 5 | 解线性代数方程组的直接法 | () |
| 5.1 | 高斯消去法 | () |
| 5.1.1 | 顺序高斯消去法 | () |
| 5.1.2 | 列主元高斯消去法 | () |
| 5.1.3 | 全主元高斯消去法 | () |
| 5.2 | LU 分解法 | () |
| 5.2.1 | 直接 LU 分解法 | () |
| 5.2.2 | 列主元 LU 分解法 | () |
| 5.3 | 对称正定矩阵的平方根法和 LDL ^T 分解法 | () |
| 5.4 | 向量与矩阵范数 | () |
| 5.4.1 | 向量范数 | () |
| 5.4.2 | 矩阵范数 | () |
| 5.4.3 | 谱半径 | () |
| 5.4.4 | 方程右端误差对解的影响 | () |
| 5.4.5 | 系数矩阵误差对解的影响 | () |

| | | |
|-------|----------------------------|-----|
| 6 | 解线性代数方程组的迭代法 | () |
| 6.1 | 几种常用的迭代格式 | () |
| 6.1.1 | 简单迭代法(Jacobi 迭代) | () |
| 6.1.2 | Seidel 迭代法 | () |
| 6.1.3 | 松弛法(SOR 迭代) | () |
| 6.2 | 迭代法收敛性理论 | () |
| 6.2.1 | 三种迭代法迭代矩阵的谱半径与系数矩阵 A 的关系 | () |
| 7 | 非线性方程和非性方程组的数值解 | () |
| 7.1 | 对分法 | () |
| 7.2 | 迭代法 | () |
| 7.2.1 | 迭代法 | () |
| 7.2.2 | 迭代法的几何意义 | () |
| 7.2.3 | 迭代法收敛条件 | () |
| 7.2.4 | 迭代法的加速 | () |
| 7.3 | 牛顿(Newton)法 | () |
| 7.3.1 | 牛顿公式 | () |
| 7.3.2 | 牛顿法的收敛速度 | () |
| 7.4 | 割线法 | () |
| 7.5 | 解非线性方程组的迭代法和牛顿法 | () |
| 7.5.1 | 迭代法 | () |
| 7.5.2 | 牛顿法 | () |
| 8 | 矩阵特征值和特征向量的数值解法 | () |
| 8.1 | 幂法 | () |
| 8.1.1 | 幂法概述 | () |
| 8.1.2 | 幂法的几点说明 | () |
| 8.2 | 反幂法 | () |
| 8.2.1 | 反幂法概述 | () |
| 8.2.2 | 原点平移加速 | () |
| 8.2.3 | 求已知特征值的特征向量 | () |
| 8.3 | 雅可比(Jacobi)方法 | () |
| 8.3.1 | 平面旋转矩阵 | () |
| 8.3.2 | 雅可比方法 | () |
| 8.3.3 | 过关雅可比方法 | () |

| | | |
|-------|---------------------------|-----|
| 8.4 | QR 算法 | () |
| 8.4.1 | 豪斯豪德尔 (Householder) 矩阵 | () |
| 8.4.2 | 化一般矩阵为拟上三角矩阵 | () |
| 8.4.3 | 矩阵的正交三角分解 | () |
| 8.4.4 | QR 算法 | () |
| 8.4.5 | QR 算法的收敛性 | () |
| 9 | 常微分方程初值问题的数值解法 | () |
| 9.1 | 欧拉 (Euler) 法 | () |
| 9.1.1 | 欧拉法的导出 | () |
| 9.1.2 | 欧拉隐式公式和欧拉中点公式 | () |
| 9.1.3 | 局部截断误差和方法的阶 | () |
| 9.1.4 | 梯形公式及其预报校正法 | () |
| 9.2 | 龙格 - 库塔 (Runge - Kutta) 法 | () |
| 9.2.1 | 二阶 R - K 法 | () |
| 9.2.2 | 四阶 R - K 法 | () |
| 9.3 | 线性多步法 | () |
| 9.3.1 | 用待定系数法构造线性多步法 | () |
| 9.3.2 | 用数值积分法构造线性多步法公式 | () |

1 误差

1.1 误差的来源

科学研究、科学实验和工程技术中的实际问题经常要用数学工具解决,而用数学工具解决实际问题往往会产生误差.按误差产生的原因可以把误差分为四种:模型误差、观测误差、方法误差和舍入误差.

在使用数学工具解决实际问题的过程中,往往需要通过分析问题,抓住主要矛盾,抛开次要因素,建立起量与量之间的数学模型.数学模型是关于部分现实世界和为一种特殊目的而作的一个抽象的、简化的结构.它是近似的,必然会带来误差,这种误差称为模型误差.

在数学模型中通常包含一些观测数据,这些数据受观测仪器精度的限制,误差在所难免,这种误差称为观测误差.

建立了数学模型之后,还要用数学方法求解数学模型.一般来说,数学模型是比较复杂的,不容易求出它的准确解,需要运用一套行之有效的近似方法来求解.用近似方法得到的解与数学模型的准确解之间必然存在误差,这种误差称为方法误差,有时也称为截断误差.

有些数要用无穷多位小数来表示,例如 π , $\sqrt{2}$, $\frac{1}{3}$, 但只能保留有限位小数进行运算,也会产生误差,这种误差称为舍入误差.

本书只讨论方法误差和舍入误差.

1.2 误差、误差限和有效数字

1.2.0.1 误差

定义 1.1 设准确值 x 的近似值为 x^* , 则近似值与准确值之间的差称为误差. 这个误差通常记为 e^* , 即

$$e^* = x^* - x \quad (1.1)$$

有时也记为 $e(x^*)$. 误差有时也称为绝对误差.

从(1.1)式可以看到,误差可能是正值,也可能是负值.当误差为正值时,说明近似值偏大,称为强近似;当误差为负值时,说明近似值偏小,称为弱近似.

1.2.0.2 误差限

一般情况下,准确值 x 是未知的,所以误差 e^* 的准确值求不出来.但有时可以根据具体测量或计算估计出误差的绝对值不可能超过某个正数,这个正数通常称为误差限.

定义 1.2 若 $|e^*| = |x^* - x| \leq \varepsilon^*$, 则 ε^* 称为近似值 x^* 的误差限. 近似值 x^* 的误差限也记为 $\varepsilon(x^*)$.

由于 $|x^* - x| \leq \varepsilon^*$, 所以 $x^* - \varepsilon^* \leq x \leq x^* + \varepsilon^*$, 即 $x \in [x^* - \varepsilon^*, x^* + \varepsilon^*]$, 也可以表示为 $x = x^* \pm \varepsilon^*$. 因此, 在日常生活中, 通常用 $x = x^* \pm \varepsilon^*$ 表示准确值 x 所在的范围和近似值 x^* 的近似程度.

例 1-1 设用米尺测量某物体的长度,测得其长度为 835 mm. 试求出该物体实际长度的范围.

解 设该物体的实际长度为 x . 由米尺的精度可知, 近似值 x^* 与准确值 x 之差的绝对值不会超过半个毫米, 即有

$$|x^* - x| = |835 - x| \leq \frac{1}{2} \text{ mm}$$

亦即 $834.5 \text{ mm} \leq x \leq 835.5 \text{ mm}$ 或 $x = 835 \pm 0.5 \text{ mm}$.

1.2.0.3 有效数字

在引入有效数字的定义前, 先看一下对数值进行四舍五入以后有什么规律.

当一个数值有很多位小数时, 常常按四舍五入的原则取有限位来表示.

例如 $\pi = 3.141\,592\,65\cdots$

取 3 位数字得 $x_3^* = 3.14$:

$$e_3^* = x_3^* - \pi \approx -0.001\,6$$

取 5 位数字得 $x_5^* = 3.141\,6$:

$$e_5^* = x_5^* - \pi \approx 0.000\,007$$

取 6 位数字得 $x_6^* = 3.141\,59$:

$$e_6^* = x_6^* - \pi \approx 0.000\,002\,6$$

下面分析一下近似值的误差, 看看能否发现有规律的东西.

$$|3.14 - \pi| \approx 0.001\,6 < 0.005 = \frac{1}{2} \times 10^{-2}$$

同理, 可以得到

$$|3.141\,6 - \pi| < \frac{1}{2} \times 10^{-4}$$

$$|3.141\,59 - \pi| < \frac{1}{2} \times 10^{-5}$$

从上面三个式子发现: π 这个数经过四舍五入之后所得到的近似值, 它的误差

限是它末位的半个单位.

可以证明:对任何数值经过四舍五入之后所得到的近似值,它的误差限是它末位的半个单位.

在数值分析中,为了更好地描述近似值的这种性质,特别引入了有效数字的定义.

定义 1.3 若近似值 x^* 的误差限为该值的某一位的半个单位,且从该位开始往左数到 x^* 的第一位非 0 数字共有 n 位,则称近似值 x^* 具有 n 位有效数字.

例如, $x_3^* = 3.14$ 具有 3 位有效数字.这是因为

$$|3.14 - \pi| < \frac{1}{2} \times 10^{-2}$$

即,近似值 $x_3^* = 3.14$ 与准确值 π 的误差限为小数点后第二位的半个单位,且从小数点第二位开始往左数到第一位非 0 数字“3”共有 3 位.

同理, $x_5^* = 3.1416$ 具有 5 位有效数字, $x_6^* = 3.14159$ 具有 6 位有效数字.

一般地,有这样的规律:凡是经过四舍五入得到的近似值,它的有效数字位等于从该近似值的末位开始往左数起到第一位非 0 数字的位数.

例如,对 0.045 678 小数点后第五位进行四舍五入后得 0.045 7,而从末位开始往左数到第一位非 0 数字共有 3 位,故 0.045 7 具有 3 位有效数字.

又如,对 8.000 5 的小数点后第三位进行四舍五入得到 8.00,而从末位开始往左数到第一位非 0 数字共有 3 位,故 8.00 有 3 位有效数字.

例 1-2 若 π 的近似值为 $\pi^* = 3.1415$,则 π^* 有多少位有效数字?

解 因为

$$|3.1415 - 3.14159265\cdots| = 0.00009265\cdots < 0.0005 = \frac{1}{2} \times 10^{-3}$$

所以,近似值 $\pi^* = 3.1415$ 的误差限为该值小数点后第三位的半个单位.由有效数字的定义可知, π^* 具有 4 位有效数字.

顺便指出,通常认为准确值具有无穷多位有效数字.

1.2.0.4 有效数字与误差限的关系

从上面的例子可以粗略地知道,有效数字与误差限一样,也可以表达近似值的近似程度.那么,它们之间有没有关系?

设 x^* 为准确值 x 的近似值,且将 x^* 表示为

$$x^* = \pm 0.\alpha_1\alpha_2\cdots\alpha_m \times 10^p$$

p 为整数; $\alpha_1, \alpha_2, \cdots, \alpha_m$ 为 0~9 之间的数字,且 $\alpha_1 \neq 0$,若有

$$|x^* - x| \leq \frac{1}{2} \times 10^{p-n} \quad (1.2)$$

则近似值 x^* 具有 n 位有效数字.

实际上,可以把这段话看成有效数字的一个等价定义.按照这个定义,如果知道近似值的误差限,就可以知道它有多少位有效数字;反过来,如果知道近似值有多少位有效数字,就可以知道它的误差限是多少.

例 1-3 假设 $x^* = 0.001\ 234\ 5$ 是准确值 x 的具有 5 位有效数字的近似值,则它的误差限为多少?

解 因为

$$x^* = 0.001\ 234\ 5 = 0.123\ 45 \times 10^{-2}$$

由此得到 $p = -2$. 所以有

$$|x^* - x| \leq \frac{1}{2} \times 10^{-2-5} = \frac{1}{2} \times 10^{-7}$$

即 x^* 的误差限为 $\frac{1}{2} \times 10^{-7}$.

例 1-4 利用有效数字与误差限的关系求解例 1-2.

解 由于

$$|3.141\ 5 - \pi| = 0.000\ 092\ 65\cdots < 0.000\ 5 = \frac{1}{2} \times 10^{-3}$$

从而得到 $p - n = -3$. 而

$$3.141\ 5 = 0.314\ 15 \times 10^1 \Rightarrow p = 1$$

因此 $n = 4$. 即 3.141 5 有 4 位有效数字.

1.3 相对误差和相对误差限

1.3.0.1 相对误差

误差和误差限在某些时候还不能完全反映近似值的近似程度.例如,某工厂有两个工人 A 和 B,工人 A 每生产 1 000 个零件就有一个次品,工人 B 每生产 2 000 个零件也有一个次品.所以,这两个工人的误差都是 1.那能不能说明这两个工人的技术水平一样高?显然是不能的,这是因为不能只看到次品的多少,还要看次品率.这两个工人的次品率分别为

$$A: \quad \frac{1\ 000 - 999}{1\ 000} = \frac{1}{1\ 000}$$

$$B: \quad \frac{2\ 000 - 1999}{2\ 000} = \frac{1}{2\ 000}$$

工人 B 的次品率比工人 A 的次品率低,说明工人 B 的技术水平高些.这启发我们在某些时候除了要看误差有多少外,还要考虑误差与准确值的比值的大小.误差与准确值的比值通常称为相对误差.

定义 1.4 若记 $e_r^* = \frac{x^* - x}{x}$, 则 e_r^* 称为 x^* 的相对误差. 近似值 x^* 的相对误差有时也记为 $e_r(x^*)$.

在实际计算中, 准确值 x 一般是不知道的, 分母中的 x 通常用近似值 x^* 代替, 即相对误差也定义为

$$e_r^* = \frac{x^* - x}{x^*} \quad (1.3)$$

1.3.0.2 相对误差限

从相对误差的定义可以看到, 由于准确值 x 一般是无法求出来的, 所以不能直接计算相对误差的值. 但有时近似值 x^* 的误差限可以求出来, 从而可以利用误差限来定义相对误差限.

对 (1.3) 式两边取绝对值后得

$$|e_r^*| = \left| \frac{x^* - x}{x^*} \right| \leq \frac{\varepsilon^*}{|x^*|}$$

故相对误差限的定义如下.

定义 1.5 若记 $\varepsilon_r^* = \frac{\varepsilon^*}{|x^*|}$, 则 ε_r^* 称为近似值 x^* 的相对误差限. 近似值 x^* 的相对误差限有时也记为 $\varepsilon_r(x^*)$.

1.3.0.3 相对误差限与有效数字的关系

相对误差限是利用误差限来定义的, 而误差限与有效数字有一定关系, 所以相对误差限也与有效数字有一定关系.

定理 1.1 若近似数 $x^* = \pm 0. \alpha_1 \alpha_2 \cdots \alpha_m \times 10^p$ 具有 n 位有效数字 ($n \leq m$), 则其相对误差限为

$$\varepsilon_r^* = \frac{\varepsilon^*}{|x^*|} \leq \frac{1}{2\alpha_1} \times 10^{-(n-1)} \quad (1.4)$$

证明 据题意, x^* 具有 n 位有效数字, 所以按有效数字的等价定义有

$$|x^* - x| \leq \frac{1}{2} \times 10^{p-n}$$

$$\text{从而 } |e_r^*| = \frac{|x^* - x|}{|x^*|} \leq \frac{\frac{1}{2} \times 10^{p-n}}{|x^*|} = \varepsilon_r^*$$

另一方面, 对 x^* 求绝对值得

$$\begin{aligned} |x^*| &= |\pm 0. \alpha_1 \alpha_2 \cdots \alpha_n \times 10^p| = \alpha_1 \cdot \alpha_2 \cdots \alpha_n \times 10^{p-1} \\ &= (\alpha_1 + 0. \alpha_2 \cdots \alpha_n) \times 10^{p-1} \geq \alpha_1 \times 10^{p-1} \end{aligned}$$

$$\text{于是 } \varepsilon_r^* = \frac{\frac{1}{2} \times 10^{p-n}}{|x^*|} \leq \frac{\frac{1}{2} \times 10^{p-n}}{\alpha_1 \times 10^{p-1}} = \frac{1}{2\alpha_1} \times 10^{-(n-1)}$$

从这个定理可以看到,有效数字位越多,相对误差限就会越小.因此,在计算过程中应该尽量避免有效数字的损失.

如果已知近似值的有效数字位,可以使用公式(1.4)求出它的相对误差限.

例 1-5 已知用 $e^* = 2.718$ 来表示 $e = 2.7182\cdots$ 具有 4 位有效数字,求 e^* 的相对误差限.

解 因为 $n=4$,由公式(1.4)得

$$\varepsilon_r^* \leq \frac{1}{2 \times 2} \times 10^{-(4-1)} = \frac{1}{4} \times 10^{-3}$$

所以 e^* 的相对误差限为 $\varepsilon_r^* = \frac{1}{4} \times 10^{-3}$.

定理 1.2 若近似数 $x^* = \pm 0.\alpha_1\alpha_2\cdots\alpha_m \times 10^p$,且相对误差限满足关系式

$$\varepsilon_r^* \leq \frac{1}{2(\alpha_1 + 1)} \times 10^{-(n-1)}$$

则 x^* 具有 n 位有效数字.

证明 据有效数字的等价定义,要证明 x^* 具有 n 位有效数字,只需证明

$$|x^* - x| \leq \frac{1}{2} \times 10^{p-n} \text{ 即可.}$$

$$\begin{aligned} \text{因为} \quad |x^*| &= |\pm 0.\alpha_1\alpha_2\cdots\alpha_m \times 10^p| = \alpha_1.\alpha_2\cdots\alpha_m \times 10^{p-1} \\ &= (\alpha_1 + 0.\alpha_2\cdots\alpha_m) \times 10^{p-1} \leq (\alpha_1 + 1) \times 10^{p-1} \end{aligned}$$

$$\begin{aligned} \text{所以} \quad |x^* - x| &= \frac{|x^*| |x^* - x|}{|x^*|} \leq |x^*| \frac{\varepsilon_r^*}{|x^*|} = |x^*| \varepsilon_r^* \\ &\leq |x^*| \times \frac{1}{2(\alpha_1 + 1)} \times 10^{-(n-1)} \leq \frac{1}{2} \times 10^{p-n} \end{aligned}$$

从而证得 x^* 具有 n 位有效数字.

1.4 数值运算中的误差估计

在计算一元函数 $f(x)$ 的值时,若用近似值 x^* 代替准确值 x 进行计算,则 $f(x)$ 的值必定存在误差.

函数 $f(x)$ 在 x^* 附近按泰勒公式展开得

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

ξ 在 x 和 x^* 之间.从而有

$$f(x) - f(x^*) = f'(x^*)(x - x^*) + \frac{f''(\xi)}{2!}(x - x^*)^2 \quad (1.5)$$

因为 $(x - x^*)$ 相对其它量是一个高阶小量,舍去右边第二项得

$$f(x) - f(x^*) \approx f'(x^*)(x - x^*) \quad (1.6)$$

即 $f(x^*)$ 的绝对误差

$$e(f(x^*)) \approx f'(x^*)e(x^*)$$

由 (1.6) 式可以得到 $f(x^*)$ 的相对误差

$$e_r(f(x^*)) \approx \frac{f'(x^*)}{f(x^*)}e(x^*) \quad (1.7)$$

对 (1.5) 式两边取绝对值得

$$\begin{aligned} |f(x) - f(x^*)| &= \left| f'(x^*)(x - x^*) + \frac{f''(\xi)}{2!}(x - x^*)^2 \right| \\ &\approx |f'(x^*)(x - x^*)| \leq |f'(x^*)| \varepsilon(x^*) \end{aligned}$$

故 $f(x^*)$ 的误差限

$$\varepsilon(f(x^*)) \approx |f'(x^*)| \varepsilon(x^*) \quad (1.8)$$

而 $f(x^*)$ 的相对误差限

$$\varepsilon_r(f(x^*)) \approx \left| \frac{f'(x^*)}{f(x^*)} \right| \varepsilon(x^*) \quad (1.9)$$

例 1-6 在计算球的体积时,为了使相对误差限为 1%,问测量半径 r 允许的相对误差限为多少?

解 计算球的体积公式为

$$V = f(r) = \frac{4}{3}\pi r^3$$

设体积 V 的近似值为 V^* , 半径 r 的近似值为 r^* , 则由 (1.9) 式得到相对误差限估计式为

$$\begin{aligned} \varepsilon_r(V^*) &\approx \left| \frac{f'(r^*)}{f(r^*)} \right| \varepsilon(r^*) \\ &= 3\varepsilon_r(r^*) \leq 1\% \end{aligned}$$

从而有 $\varepsilon_r(r^*) \leq \frac{1}{300}$.

这说明,测量半径 r 时允许的相对误差限为 $\frac{1}{300}$.

在计算二元函数 $f(x, y)$ 的值时,如果用近似值 x^* 代替准确值 x , 用近似值 y^* 代替准确值 y 进行计算, 则 $f(x, y)$ 的值也必然存在误差.

利用二元函数泰勒公式可以得到如下两个数的和、差、积与商的误差估计式:

$$\begin{aligned}
e(x^* \pm y^*) &\approx e(x^*) \pm e(y^*) \\
e(x^* \cdot y^*) &\approx y^* e(x^*) + x^* e(y^*) \\
e\left(\frac{x^*}{y^*}\right) &\approx \frac{1}{y^*} e(x^*) - \frac{x^*}{(y^*)^2} e(y^*) \quad (y^* \neq 0) \\
\varepsilon(x^* \pm y^*) &\leq \varepsilon(x^*) + \varepsilon(y^*) \\
\varepsilon(x^* y^*) &\leq |y^*| \varepsilon(x^*) + |x^*| \varepsilon(y^*) \\
\varepsilon\left(\frac{x^*}{y^*}\right) &\leq \frac{|y^*| \varepsilon(x^*) + |x^*| \varepsilon(y^*)}{(y^*)^2} \quad (y^* \neq 0) \\
e_r(x^* + y^*) &\approx \frac{x^*}{x^* + y^*} e_r(x^*) + \frac{y^*}{x^* + y^*} e_r(y^*) \\
e_r(x^* - y^*) &\approx \frac{x^*}{x^* - y^*} e_r(x^*) - \frac{y^*}{x^* - y^*} e_r(y^*) \\
e_r(x^* y^*) &\approx e_r(x^*) + e_r(y^*) \\
e_r\left(\frac{x^*}{y^*}\right) &\approx e_r(x^*) - e_r(y^*) \quad (y^* \neq 0) \\
\varepsilon_r(x^* \pm y^*) &\leq \frac{\varepsilon_r(x^*) + \varepsilon_r(y^*)}{|x^* \pm y^*|} \\
\varepsilon_r(x^* y^*) &\leq \varepsilon_r(x^*) + \varepsilon_r(y^*) \\
\varepsilon_r\left(\frac{x^*}{y^*}\right) &\leq \varepsilon_r(x^*) + \varepsilon_r(y^*) \quad (y^* \neq 0)
\end{aligned}$$

例 1-7 假定某长方形运动场的长为 x , 宽为 y , 并实地测得其长 $x^* = 100.30$ m, 宽 $y^* = 80.50$ m. 若 x^* 和 y^* 的误差限都是 0.005 m, 试求其面积 S 的近似值 S^* 的误差限和相对误差限.

解 据题意, $S = xy$, $|x - x^*| \leq 0.005$ m, $|y - y^*| \leq 0.005$ m. 由两个数的积的误差限估计式得

$$\begin{aligned}
\varepsilon(S^*) &= \varepsilon(x^* y^*) \leq |y^*| \varepsilon(x^*) + |x^*| \varepsilon(y^*) \\
&= 80.50 \times 0.005 + 100.30 \times 0.005 = 0.9040
\end{aligned}$$

由两个数的积的相对误差限估计式得

$$\begin{aligned}
\varepsilon_r(S^*) &= \varepsilon_r(x^* y^*) \leq \varepsilon_r(x^*) + \varepsilon_r(y^*) \\
&= \frac{0.005}{100.30} + \frac{0.005}{80.50} \approx 0.00011
\end{aligned}$$

假设多元函数 $y = f(x_1, x_2, \dots, x_n)$, y 的近似值为 y^* , x_1, x_2, \dots, x_n 的近似值分别为 $x_1^*, x_2^*, \dots, x_n^*$, 则由多元函数泰勒展开式可以得到

$$\begin{aligned}
 e(y^*) &\approx \sum_{i=1}^n \left[\frac{\partial f(x_1^*, x_2^*, \dots, x_n^*)}{\partial x_i} \right] e(x_i^*) \\
 e_r(y^*) &\approx \left[\sum_{i=1}^n \left[\frac{\partial f(x_1^*, x_2^*, \dots, x_n^*)}{\partial x_i} \right] e(x_i^*) \right] / y^* \\
 \varepsilon(y^*) &\approx \sum_{i=1}^n \left| \frac{\partial f(x_1^*, x_2^*, \dots, x_n^*)}{\partial x_i} \right| \varepsilon(x_i^*) \\
 \varepsilon_r(y^*) &\leq \left[\sum_{i=1}^n \left| \frac{\partial f(x_1^*, x_2^*, \dots, x_n^*)}{\partial x_i} \right| \varepsilon(x_i^*) \right] / |y^*|
 \end{aligned}$$

1.5 数值计算中应注意的一些问题

在利用计算机进行编程计算时,不要以为使用的计算公式或算法对了,编写的程序没有语法错误,就一定会得到正确的计算结果.因为在数值计算中几乎每一步计算都存在误差,有时由于某一步计算的误差很大,有可能造成计算结果错误;或者虽然每一步计算的误差不是很大,但由于误差会传播,会积累,也有可能造成计算结果严重不可靠.

在数值计算中,为了得到正确的计算结果,需要注意下面几个问题.

1.5.0.1 避免两个相近的数相减

由 1.4 节可知

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

从上式可以看到,如果两个相近的数相减,则 $|x^* - y^*|$ 较小,相对误差限就会较大,故有效数字位会大大减少.

因此,当遇到两个相近的数相减时,参与运算的数应当多保留几位有效数字,或者变换原来的公式以避免这种情况的发生.

例 1-8 给定 $f(x) = 1 - \cos x$,若使用计算器取 4 位有效数字计算 $f(2^\circ)$,应如何变换公式使有效数字位增加?

解 使用计算器计算取 4 位有效数字得 $\cos 2^\circ \approx 0.9994$,从而

$$f(2^\circ) = 1 - \cos 2^\circ \approx 0.0006$$

由于

$$f(x) = 1 - \cos x = 2 \sin^2 \frac{x}{2}$$

使用计算器取 4 位有效数字得 $\sin 1^\circ = 0.01745$,所以有

$$f(2^\circ) = 2 \sin^2 1^\circ \approx 0.0006090$$

若使用计算机计算,有

$$f(2^\circ) = 1 - \cos 2^\circ \approx 0.000609173$$

这说明变换公式后能使有效数字位由 1 位增加到 3 位.

1.5.0.2 要防止小数被大数“吃掉”而使有效数字位损失

在数值计算中,如果两个参与计算的数相差太大,则小数有可能被大数“吃掉”而使有效数字位损失,从而影响计算结果的可靠性.

例 1-9 求一元二次方程 $ax^2 + bx + c = 0$ 的根.

解 求一元二次方程的根可以使用公式

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

但是,如果 b^2 远远大于 $|4ac|$,则在计算机上计算有可能有

$$\sqrt{b^2 - 4ac} = |b|$$

b^2 “吃掉” $4ac$,从而使两个求根公式之一出现两个相近的数相减,使 x_1 或 x_2 损失有效数字位,从而使计算结果出现错误.

例如,在只有 7 位有效数字的计算机系统上使用求根公式解方程

$$x^2 - (10^8 + 1)x + 10^8 = 0$$

得到的两个根为 $x_1 = 10^8, x_2 = 0$. 但 x_2 显然不是方程的根.

为避免这种错误的发生,可以修改求根公式为

$$x_1 = \frac{-b - \text{sign}(b) \sqrt{b^2 - 4ac}}{2a} \quad x_2 = \frac{c}{ax_1}$$

其中函数 $\text{sign}(b)$ 用来求 b 的符号.

按新的求根公式解方程

$$x^2 - (10^8 + 1)x + 10^8 = 0$$

得方程的两个准确根为 $x_1 = 10^8, x_2 = 1$.

1.5.0.3 要注意减少运算的次数

对于一个计算问题,如果能减少运算次数,不仅能减少计算时间,提高运行速度,而且还可以减少误差的积累,提高计算结果准确性.

例 1-10 计算 n 次多项式

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

的值.

解 假如按公式直接计算每一项 $a_k x^k$ 后,再进行求和,就要进行

$$n + (n-1) + \cdots + 2 + 1 = \frac{n(n+1)}{2}$$

次乘法和 n 次加法计算. 如果把原式改写为

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

则计算 n 次多项式的算法可以是

$$m_n = a_n$$

$$m_k = xm_{k+1} + a_k \quad (k = n-1, n-2, \dots, 1, 0)$$

$$p_n(x) = m_0$$

则计算 n 次多项式的值只需 n 次乘法和 n 次加法.

这个算法就是著名的秦九韶算法.

1.5.0.4 避免做除数绝对值远远小于被除数绝对值的除法

很显然,用绝对值较小的数去除绝对值较大的数,得到的数一定会较大,有可能会产生溢出错误.如果不溢出,也有可能使舍入误差严重增大,导致最后结果不可靠.

例 1-11 求解方程组

$$\begin{cases} 0.0003x_1 + 3x_2 = 2.0001 \\ x_1 + x_2 = 1 \end{cases}$$

在计算过程中,按四舍五入原则取小数点后 4 位进行计算.

解 容易验证,方程组的准确解为 $x_1 = \frac{1}{3}, x_2 = \frac{2}{3}$. 为求方程组的解,把第一个方程乘上 $-1/0.0003$ 后加到第二个方程得

$$-9999x_2 = -6666$$

从而解得 $x_2 = 0.6666\cdots \approx 0.6667$. 把 x_2 代入第一个方程求得 $x_1 = 0$.

这说明,在计算过程中,若按四舍五入原则取小数点后 4 位进行计算,用消元法得到的近似解与准确解相差较大. 这是因为用较小的数作除数,使得在消元的过程中,方程组的一些元素的绝对值严重增长,增大了舍入误差的量级,在计算 x 时产生了较大的舍入误差,从而造成计算结果错误.

1.5.0.5 要选择数值稳定的计算公式

定义 1.6 一种数值方法,若原始数据有误差,而在计算的过程中,由于舍入误差的传播,使得近似计算结果与准确值相差很大,则称这种数值方法是不稳定的. 否则,在计算的过程中,若舍入误差得到控制,近似计算结果能逼近准确值,则称这种数值方法是稳定的.

例 1-12 给定 $I_n = \int_0^1 x^n e^{x-1} dx$, 试问用递推公式 $I_n = 1 - nI_{n-1}$ ($n = 1, 2, \dots$)

采用正向递推和逆向递推求 I_n 的值是否稳定?

解 设 I_n 的近似值为 I_n^* , 则

$$I_n^* = 1 - nI_{n-1}^*$$

从而有

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

对上式两边取绝对值得

$$|I_n - I_n^*| = n |I_{n-1} - I_{n-1}^*| = n(n-1) |I_{n-2} - I_{n-2}^*| = \cdots = n! |I_0 - I_0^*|$$

由此可知,按递推公式 $I_n = 1 - nI_{n-1}$ ($n = 1, 2, \dots$) 采用正向递推求 I_n 的值时,误差传播逐步增大, I_n 与 I_n^* 的误差是 I_0 与 I_0^* 的误差的 $n!$ 倍. 这说明,按给定的递推公式采用正向递推计算 I_n 的值是不稳定的.

由递推公式 $I_n = 1 - nI_{n-1}$ ($n = 1, 2, \dots$) 有

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

若已知 I_n 的近似值为 I_n^* , 则 I_{n-1} 的近似值

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

将上两式相减得

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

对上式两边取绝对值得

$$|I_{n-1} - I_{n-1}^*| = \frac{1}{n} |I_n - I_n^*|$$

这说明,按递推公式 $I_n = 1 - nI_{n-1}$ ($n = 1, 2, \dots$) 采用逆向递推求 I_n 的值时,误差传播逐步减少. 因此,按给定的递推公式采用逆向递推计算 I_n 的值是稳定的.

从例 1-12 可以看到,不稳定的数值方法会使误差变得越来越大. 在实践中,应当注意误差分析,避免使用不稳定的数值方法.

练习与思考

1. 下列各数是按四舍五入原则得到的近似值. 它们各自有多少位有效数字? 它们的误差限和相对误差限为多少?

(1) 123.45; (2) 0.835 00; (3) 0.015; (4) 10.008

2. 已知近似数 x^* 的相对误差限为 0.5%, 问 x^* 至少有多少位有效数字?

3. 给出某个正方形的边长约为 100 cm, 当边长的误差不超过多少厘米时, 才能保证其面积的误差不超过 1 cm^2 ?

4. 在真空中, 自由落体运动距离 s 与时间 t 的函数关系为

$$s = \frac{1}{2}gt^2$$

其中 g 为重力加速度. 假定 g 是准确的, 而 t 的测量有 $\pm 0.1 \text{ s}$ 的误差. 试证明: 当时间增加时, 距离的误差增加, 而相对误差却减少.

5. 假设准确值 x 和 y 的近似值为 $x^* = 1.004 8$ 和 $y^* = 0.253$, 它们是经过四舍五入得到的. 求 $x^* + y^*$, $x^* - y^*$, $x^* \times y^*$ 和 $\frac{x^*}{y^*}$ 的误差限和相对误差限.

6. 下面各数 $x_1 = 1.21$, $x_2 = 0.05$, $x_3 = 10.380$ 是按四舍五入原则得到的近似值. 试估计 $x_1 + x_2 + x_3$ 和 $x_1 x_2 x_3$ 的相对误差限.

7. 设有一个长方形的水池, 经测量得知它的长为 50 m, 宽为 25 m, 深为 20 m, 它们的误差限都为 0.01 m. 求该水池容积近似值的误差、误差限、相对误差和相对误差限.

8. 已知求三角形面积公式为

$$S = \frac{1}{2}absinc$$

其中 c 为弧度, $0 < c < \frac{\pi}{2}$. 而且假定测量 a, b, c 时的误差分别为 $e(a^*)$, $e(b^*)$ 和 $e(c^*)$. 证明: 三角形面积的误差 $e(S^*)$ 满足

$$\left| \frac{e(S^*)}{S} \right| \leq \left| \frac{e(a^*)}{a} \right| + \left| \frac{e(b^*)}{b} \right| + \left| \frac{e(c^*)}{c} \right|$$

9. 给定如下公式, 应如何变换公式使有效数字位增加?

$$(1) \frac{1 - \cos x}{\sin x} \quad (x \text{ 接近于 } 0) \qquad (2) \sqrt{1+x} - \sqrt{x} \quad (x \text{ 充分大})$$

$$(3) \lg x - \lg y \quad (x \text{ 接近于 } y)$$

10. 假定 $\{y_n\}$ 满足递推关系

$$y_n = 10y_{n-1} - 1 \quad (n = 1, 2, \dots)$$

如果取 $y_0 = \sqrt{2} \approx 1.41$, 试问用该递推公式求 y_n 的值是否稳定? 要求写出详细根据.

2 代数插值与数值微分

在工程技术中,经常会遇到只给定一个函数表,要求根据该函数表求出某些点上函数值的问题.

例如,某气象台经过气象探测得到高度与大气压的一些数据,这些数据如表 2-1 所示.

表 2-1 高度与大气压的探测数据

| | | | | | |
|----------------|---|-----|------|-----|------|
| x (高度) | 0 | 1.5 | 2.5 | 3 | 4.8 |
| $y=f(x)$ (大气压) | 1 | 0.9 | 0.85 | 0.7 | 0.67 |

表 2-2 函数表

| | | | | |
|----------|-------|-------|----------|-------|
| x | x_0 | x_1 | \cdots | x_n |
| $y=f(x)$ | y_0 | y_1 | \cdots | y_n |

表 2-1 表示了高度与大气压的函数关系.从这个表可以知道,当高度为 0 时,大气压为 1;当高度为 1.5 时,大气压为 0.9……当高度为 4.8 时,大气压为 0.67.但是,实际问题往往不仅仅要求出给定高度的大气压,而且还要求出任意高度的大气压,比如高度为 2.7 时的大气压.

上面提出的问题是说,给定一个函数的函数表(如表 2-2 所示),如何求出任意点的函数值.这个问题的其中一个办法是:根据表 2-2 按某些条件构造出一个函数 $\varphi(x)$ 来近似代替 $f(x)$.如果能构造函数 $\varphi(x)$ 近似代替 $f(x)$,那么只要给出任何一个点 x^* ,就可以求出 $\varphi(x^*)$,用 $\varphi(x^*)$ 近似代替 $f(x^*)$.现在的问题是, $\varphi(x)$ 选取什么样的函数. $\varphi(x)$ 的选取可以是多种多样的,它可以是代数多项式,也可以是三角多项式,还可以是有理分式.但是,为了便于计算函数值和研究函数的性质,一般选取代数多项式 $p_n(x)$ 作为 $\varphi(x)$.构造代数多项式 $p_n(x)$ 来近似代替 $f(x)$,称为代数插值.

本章主要讨论代数插值和函数的求导问题.

2.1 线性插值与二次插值

2.1.1 线性插值

线性插值是代数插值中最简单的一种情形.它的提法是:

给定 $y=f(x)$ 的函数表(如表 2-3 所示),构造函数 $p_1(x)$ 满足条件:

- ① $p_1(x)$ 是一个不超过一次的代数多项式;
- ② $p_1(x_0)=y_0, p_1(x_1)=y_1$.

其中, x_0 和 x_1 称为插值节点; $f(x)$ 称为被插值函数; $p_1(x)$ 称为线性插值函数; 条件①和②称为插值条件.

线性插值的目的是构造 $p_1(x)$ 来近似代替 $f(x)$. 当求某一点 x^* 的函数值 $f(x^*)$ 时, 可以用 $p_1(x^*)$ 来近似代替 $f(x^*)$.

因为 $p_1(x)$ 满足插值条件②, 所以线性插值的几何意义是用过 (x_0, y_0) 和 (x_1, y_1) 两点的直线 $y = p_1(x)$ 来近似代替曲线 $y = f(x)$, 如图 2-1 所示.

由初等几何知识可知, 过两点的直线存在且惟一. 因此, $p_1(x)$ 存在且惟一.

线性插值函数 $p_1(x)$ 有两种形式: Lagrange (拉格朗日) 形式和 Newton (牛顿) 形式.

2.1.1.1 线性插值函数的 Lagrange 形式

由直线方程的两点式可以直接得到

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

(2.1)

$p_1(x)$ 称为 Lagrange 型线性插值函数; 式 (2.1) 称为 Lagrange 型线性插值公式.

可以看到, $p_1(x)$ 实际上是两个线性函数 $\frac{x - x_1}{x_0 - x_1}$ 和 $\frac{x - x_0}{x_1 - x_0}$ 的线性组合. 如果把这两个线性函数分别记为 $l_0(x)$ 和 $l_1(x)$, 则有

$$p_1(x) = l_0(x)y_0 + l_1(x)y_1$$

其中 $l_0(x)$ 和 $l_1(x)$ 称为一次插值基函数, 它们在插值节点处的取值参见表 2-4.

2.1.1.2 线性插值函数的 Newton 形式

由于 Newton 形式的线性插值函数与一阶差商有关, 所以在推导 Newton 形式的线性插值函数之前, 先引入一阶差商的定义.

定义 2.1 若记 $f(x_i, x_j) = \frac{f(x_i) - f(x_j)}{x_i - x_j}$, 则称 $f(x_i, x_j)$ 为函数 $f(x)$ 在节点 x_i 和 x_j 处的一阶差商, 其中 x_i 与 x_j 互异.

特别地, 当 $i = 0$ 和 $i = 1$ 时, 有

表 2-3 线性插值函数表

| x | x_0 | x_1 |
|------------|-------|-------|
| $y = f(x)$ | y_0 | y_1 |

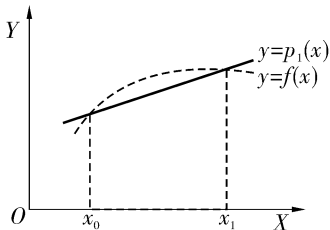


图 2-1 线性插值的几何意义

表 2-4 $l_0(x)$ 和 $l_1(x)$ 在节点处的取值

| 函数 \ 节点 | x_0 | x_1 |
|----------|-------|-------|
| $l_0(x)$ | 1 | 0 |
| $l_1(x)$ | 0 | 1 |

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

从一阶差商的定义可以看到

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

这就是说,一阶差商的值与节点的排列顺序无关. 这种性质称为对称性.

特别地,当 $i=0$ 和 $j=1$ 时,有

$$f(x_0, x_1) = f(x_1, x_0)$$

已知 $p_1(x) = a + bx$ 是一个直线方程且满足 $p_1(x_0) = y_0$ 和 $p_1(x_1) = y_1$, 即它经过 (x_0, y_0) 和 (x_1, y_1) 两点. 按点斜式直线方程 $y - y_0 = k(x - x_0)$, 其中 k 为斜率, 有

$$\begin{aligned} p_1(x) &= y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \\ &= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \\ &= f(x_0) + f(x_0, x_1)(x - x_0) \end{aligned} \quad (2.2)$$

式(2.2)称为 Newton 型线性插值公式, 对应的 $p_1(x)$ 称为 Newton 型线性插值函数.

从上面的讨论可以知道, 线性插值函数 $p_1(x)$ 有两种形式, 尽管形式不同, 但实质都一样, 都是代表同一条直线, 如图 2-1 所示.

前面已介绍过, 线性插值是按插值条件构造插值函数 $p_1(x)$ 来近似代替 $f(x)$. 很显然, $p_1(x)$ 与 $f(x)$ 之间必然存在误差, 那么它们之间的误差为多少?

定理 2.1 设 $p_1(x)$ 是过 (x_0, y_0) 和 (x_1, y_1) 两点的线性插值函数, $[a, b]$ 是包含 $[x_0, x_1]$ 的任一区间, 并设 $f(x) \in C^1[a, b]$, $f''(x)$ 在 $[a, b]$ 上存在, 则对任意给定的 $x \in [a, b]$, 总存在一点 $\xi \in (a, b)$, 使得

$$R(x) = f(x) - p_1(x) = \frac{f''(\xi)}{2!}(x - x_0)(x - x_1) \quad (2.3)$$

其中 ξ 依赖于 x .

证 对任一固定的 x , 引入辅助函数 $\varphi(t)$. 要求 $\varphi(x_0) = \varphi(x_1) = \varphi(x) = 0$, 不妨把 $\varphi(t)$ 表示为

$$\varphi(t) = f(t) - p_1(t) - k(t - x_0)(t - x_1) \quad (2.4)$$

其中 k 为待定常数.

显然, $\varphi(x_0) = \varphi(x_1) = 0$.

要使 $\varphi(x) = 0$, 由(2.4)式得

$$\varphi(x) = f(x) - p_1(x) - k(x - x_0)(x - x_1) = 0$$

即

$$R(x) - k(x - x_0)(x - x_1) = 0$$

由此得到

$$k = \frac{R(x)}{(x-x_0)(x-x_1)}$$

因而

$$\varphi(t) = f(t) - p_1(t) - \frac{R(x)}{(x-x_0)(x-x_1)}(t-x_0)(t-x_1)$$

有 3 个零点: x_0, x_1 和 x .

对于 $\varphi(t)$ 的 3 个零点 x_0, x_1 和 x , 它们可以构成两个子区间. 由 Rolle(洛尔)定理(若 $f(t)$ 在 $[a, b]$ 上连续, 在 (a, b) 上可导, 且有 $f(a) = f(b)$, 则在 (a, b) 内至少有一点 c , 使 $f'(c) = 0$) 可知 $\varphi'(t)$ 在每个子区间上至少存在一个零点, $\varphi'(t)$ 在 $[a, b]$ 上至少有两个零点, 可记为 ξ_1 和 ξ_2 , 即有 $\varphi'(\xi_1) = 0$ 和 $\varphi'(\xi_2) = 0$. 同理, $\varphi''(t)$ 在 (ξ_1, ξ_2) 上至少有一个零点, 可记为 ξ , 即有 $\varphi''(\xi) = 0$.

对 $\varphi(t)$ 求二次微商, 得

$$\varphi''(t) = f''(t) - p_1''(t) - 2! \frac{R(x)}{(x-x_0)(x-x_1)}$$

由于 $p_1(t)$ 是 t 的一次多项式, 所以有 $p_1''(t) = 0$. 于是得到

$$\varphi''(t) = f''(t) - 2! \frac{R(x)}{(x-x_0)(x-x_1)} \quad (2.5)$$

把 $t = \xi$ 代入(2.5)式, 并注意到 $\varphi''(\xi) = 0$, 得

$$\varphi''(\xi) = f''(\xi) - 2! \frac{R(x)}{(x-x_0)(x-x_1)} = 0$$

移项后有

$$R(x) = \frac{f''(\xi)}{2!} (x-x_0)(x-x_1) \quad (2.6)$$

$R(x)$ 就是 $f(x)$ 与 $p_1(x)$ 的误差, 称为线性插值余项.

对于线性插值余项, 还可以进一步证明

$$|R(x)| \leq \frac{(x_1 - x_0)^2}{8} \max_{x_0 \leq x \leq x_1} |f''(x)| \quad (2.7)$$

(2.7)式的证明留给读者作为习题.

例 2-1 给定如表 2-5 所示的函数表, 用 Lagrange 型线性插值公式求 $\ln 11.75$ 的近似值, 并估计误差.

表 2-5 函数表

| x | 11 | 12 |
|-------------|---------|---------|
| $y = \ln x$ | 2.397 9 | 2.484 9 |

解 根据 Lagrange 型线性插值公式(2.1), 有

$$p_1(x) = \frac{(x-12)}{(11-12)} \times 2.397\ 9 + \frac{(x-11)}{(12-11)} \times 2.484\ 9$$

则 $p_1(11.75) = \frac{11.75-12}{11-12} \times 2.397\ 9 + \frac{11.75-11}{12-11} \times 2.484\ 9 = 2.463\ 2$

从而得到 $\ln 11.75 \approx 2.463\ 2$. 又因为

$$\begin{aligned} f(x) - p_1(x) &= \frac{f''(\xi)}{2!} (x - x_0)(x - x_1) \\ &= -\frac{1}{2\xi^2} (x - 11)(x - 12) \end{aligned}$$

其中 $\xi \in [11, 12]$. 所以

$$\begin{aligned} |f(x) - p_1(x)| &= \frac{1}{2\xi^2} |(x - 11)(x - 12)| \\ &\leq \frac{1}{2} \max_{\xi \in [11, 12]} \frac{1}{\xi^2} |(x - 11)(x - 12)| \\ &= \frac{1}{2 \times 11^2} |(x - 11)(x - 12)| \end{aligned}$$

$$\begin{aligned} \text{于是} \quad |\ln 11.75 - p_1(11.75)| &\leq \frac{1}{2 \times 11^2} |(11.75 - 11)(11.75 - 12)| \\ &\approx 0.000\,774\,8 \end{aligned}$$

2.1.2 二次插值

二次插值的提法是:

表 2-6 二次插值函数表

给定函数表(如表 2-6 所示),构造函数 $p_2(x)$ 满足条件:

| x | x_0 | x_1 | x_2 |
|------------|-------|-------|-------|
| $y = f(x)$ | y_0 | y_1 | y_2 |

① $p_2(x)$ 是一个不超过二次的代数多项式;

② $p_2(x_0) = y_0, p_2(x_1) = y_1$ 和 $p_2(x_2) = y_2$.

其中 x_0, x_1 和 x_2 称为插值节点; $f(x)$ 称为被插值函数; $p_2(x)$ 称为二次插值函数; 条件①和②称为插值条件.

二次插值的目的是构造 $p_2(x)$ 来近似代替 $f(x)$. 当求某一点 x^* 的函数值 $f(x^*)$ 时, 可以用 $p_2(x^*)$ 来近似代替 $f(x^*)$.

根据二次插值的提法, $p_2(x)$ 是一个不超过二次的代数多项式, 所以不妨假设

$$p_2(x) = a_0 + a_1x + a_2x^2$$

其中 a_0, a_1 和 a_2 为待定常数.

为了确定这三个待定常数, 可以利用插值条件②得到三个等式, 把它们联立起来组成方程组

$$\begin{cases} p_2(x_0) = a_0 + a_1x_0 + a_2x_0^2 = y_0 \\ p_2(x_1) = a_0 + a_1x_1 + a_2x_1^2 = y_1 \\ p_2(x_2) = a_0 + a_1x_2 + a_2x_2^2 = y_2 \end{cases} \quad (2.8)$$

方程组(2.8)的系数行列式为

$$\begin{vmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{vmatrix} = (x_1 - x_0)(x_2 - x_0)(x_2 - x_1)$$

所以,只要 x_0, x_1 和 x_2 互不相等,方程组的解即 a_0, a_1 和 a_2 存在且惟一,这样就可以构造出惟一的 $p_2(x)$.

与线性插值函数一样,二次插值函数 $p_2(x)$ 也有两种形式:Lagrange 形式和 Newton 形式.

2.1.2.1 二次插值函数的 Lagrange 形式

设 $p_2(x) = l_0(x)y_0 + l_1(x)y_1 + l_2(x)y_2$, 其中 $l_0(x)$, $l_1(x)$ 和 $l_2(x)$ 为不超过二次的代数多项式,且满足表 2-7.

表 2-7 $l_0(x), l_1(x)$ 和 $l_2(x)$ 在节点处的取值

| 函数值 节点 函数 | x_0 | x_1 | x_2 |
|-----------------|-------|-------|-------|
| $l_0(x)$ | 1 | 0 | 0 |
| $l_1(x)$ | 0 | 1 | 0 |
| $l_2(x)$ | 0 | 0 | 1 |

容易验证, $p_2(x)$ 满足插值条件 ①和②. 因此,只要找到满足表 2-7 的 $l_0(x)$, $l_1(x)$ 和 $l_2(x)$, 就可以构造出 $p_2(x)$. 下面根据表 2-7 来构造

$l_0(x)$, $l_1(x)$ 和 $l_2(x)$.

首先构造 $l_0(x)$. 据表 2-7, $l_0(x_1) = 0$ 和 $l_0(x_2) = 0$, 所以 $l_0(x)$ 必有因子 $(x - x_1)(x - x_2)$. 又因 $l_0(x)$ 为不超过二次的代数多项式, 而 $(x - x_1)(x - x_2)$ 已是一个二次代数多项式, 故 $l_0(x)$ 只可能有一个常数因子, 从而有

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

其中 A 为待定常数.

为了确定待定常数 A , 利用 $l_0(x_0) = 1$ 得

$$l_0(x_0) = A(x_0 - x_1)(x_0 - x_2) = 1$$

由此推得

$$A = \frac{1}{(x_0 - x_1)(x_0 - x_2)}$$

因此

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

用同样的方法可以求得

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

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

因此

$$p_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}y_2 \quad (2.9)$$

其中 $l_0(x)$, $l_1(x)$ 和 $l_2(x)$ 称为二次插值基函数; $p_2(x)$ 称为 Lagrange 型二次插值函数; (2.9) 式称为 Lagrange 型二次插值公式.

2.1.2.2 二次插值函数的 Newton 形式

由于 Newton 形式的二次插值函数与二阶差商有关, 所以在推导 Newton 形式的二次插值函数之前, 先引入二阶差商的定义.

定义 2.2 若记 $f(x_i, x_j, x_k) = \frac{f(x_i, x_j) - f(x_j, x_k)}{x_i - x_k}$, 则称 $f(x_i, x_j, x_k)$ 为函数 $f(x)$ 在节点 x_i, x_j 和 x_k 处的二阶差商, 其中 x_i, x_j 和 x_k 互异.

特别地, 当 $i=0, j=1$ 和 $k=2$ 时, 有

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

与一阶差商一样, 二阶差商也具有对称性. 也就是说, 在求二阶差商时, 无论 x_i, x_j 和 x_k 如何排列, 它们的值都是一样的.

因为 $p_2(x)$ 为不超过二次代数多项式, 所以为了构造上的方便, 可以假设

$$p_2(x) = A + B(x - x_0) + C(x - x_0)(x - x_1) \quad (2.10)$$

其中 A, B 和 C 为待定常数.

确定三个待定常数 A, B 和 C , 需要利用插值条件②.

由插值条件 $p_2(x_0) = y_0$ 得

$$A = y_0 = f(x_0)$$

由插值条件 $p_2(x_1) = y_1$ 得

$$A + B(x_1 - x_0) = f(x_0) + B(x_1 - x_0) = y_1 = f(x_1)$$

故

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

由插值条件 $p_2(x_2) = y_2$ 得

$$\begin{aligned} & A + B(x_2 - x_0) + C(x_2 - x_0)(x_2 - x_1) \\ &= f(x_0) + f(x_0, x_1)(x_2 - x_0) + C(x_2 - x_0)(x_2 - x_1) = y_2 = f(x_2) \end{aligned}$$

从而有

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

于是, 由 (2.10) 式得

$$p_2(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) \quad (2.11)$$

可以验证, $p_2(x)$ 满足插值条件①和②, 它是一个二次插值函数. $p_2(x)$ 称为 Newton 型二次插值函数, (2.11) 式称为 Newton 型二次插值公式.

二次插值也称为抛物线插值. 抛物线插值的几何意义是用抛物线来代替曲线 $y=f(x)$, 如图2-2所示.

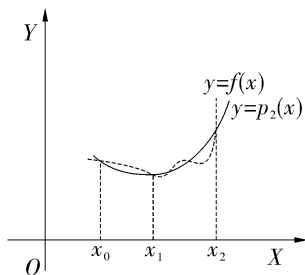


图2-2 抛物线插值的几何意义

定理 2.2 设 $p_2(x)$ 是过 (x_0, y_0) , (x_1, y_1) 和 (x_2, y_2) 三点的二次插值函数, $[a, b]$ 是包含 x_0, x_1, x_2 的任一区间, 并设 $f(x) \in C^2[a, b]$, $f'''(x)$ 在 $[a, b]$ 上存在, 则对任意给定的 $x \in [a, b]$, 总存在一点 $\xi \in (a, b)$, 使得

$$R(x) = f(x) - p_2(x) = \frac{f'''(\xi)}{3!} (x - x_0)(x - x_1)(x - x_2) \quad (2.12)$$

其中 ξ 依赖于 x .

证 对任一固定的 x , 引入辅助函数 $\varphi(t)$. 要使 $\varphi(x_0) = \varphi(x_1) = \varphi(x_2) = \varphi(x) = 0$, 不妨把 $\varphi(t)$ 表示为

$$\varphi(t) = f(t) - p_2(t) - k(t - x_0)(t - x_1)(t - x_2) \quad (2.13)$$

其中 k 为待定常数.

显然, $\varphi(x_0) = \varphi(x_1) = \varphi(x_2) = 0$.

要使 $\varphi(x) = 0$, 由 (2.13) 式得

$$\varphi(x) = f(x) - p_2(x) - k(x - x_0)(x - x_1)(x - x_2) = 0$$

即

$$R(x) - k(x - x_0)(x - x_1)(x - x_2) = 0$$

由此得到

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

因此

$$\varphi(t) = f(t) - p_2(t) - \frac{R(x)}{(x - x_0)(x - x_1)(x - x_2)}(t - x_0)(t - x_1)(t - x_2)$$

有 4 个零点: x_0, x_1, x_2 和 x .

$\varphi(t)$ 的 4 个零点 x_0, x_1, x_2 和 x 可以构成 3 个子区间. 由 Rolle 定理可知 $\varphi'(t)$ 在 $[a, b]$ 上至少有 3 个零点, 可记为 ξ_1, ξ_2 和 ξ_3 . 由 ξ_1, ξ_2 和 ξ_3 又可以构成两个子区间, 故 $\varphi''(t)$ 在 $[a, b]$ 上至少有 2 个零点, 可记为 η_1 和 η_2 . $\varphi'''(t)$ 在 (η_1, η_2) 上至少有 1 个零点, 把这个零点记为 ξ , 即有 $\varphi'''(\xi) = 0$.

现对 $\varphi(t)$ 求三次微商. 由于 $p_2(t)$ 是 t 的二次多项式, 所以求三次微商后, 它的值为 0. 于是

$$\varphi'''(t) = f'''(t) - 3! \frac{R(x)}{(x - x_0)(x - x_1)(x - x_2)} \quad (2.14)$$

把 ξ 代入 (2.14) 式并移项后得

$$R(x) = \frac{f'''(\xi)}{3!}(x-x_0)(x-x_1)(x-x_2)$$

例 2-2 给定如表 2-8 所示函数表, 试用 Newton 型二次插值公式求 $\ln 11.75$ 的近似值, 并估计误差.

表 2-8 函数表

| x | 11 | 12 | 13 |
|-------------|---------|---------|---------|
| $y = \ln x$ | 2.397 9 | 2.484 9 | 2.564 9 |

解 根据 Newton 型二次插值公式 (2.8) 有

$$p_2(x) = f(11) + f(11, 12)(x-11) + f(11, 12, 13)(x-11)(x-12)$$

而

$$f(11) = 2.397\ 9$$

$$f(11, 12) = \frac{f(11) - f(12)}{11 - 12} = 0.087$$

$$f(11, 12, 13) = \frac{f(11, 12) - f(12, 13)}{11 - 13} = -0.003\ 5$$

从而得到

$$p_2(x) = 2.397\ 9 + 0.087(x-11) - 0.003\ 5(x-11)(x-12)$$

$$p_2(11.75) = 2.463\ 75$$

于是得到 $\ln 11.75 \approx 2.463\ 75$. 又因为

$$f(x) - p_2(x) = \frac{f'''(\xi)}{3!}(x-x_0)(x-x_1)(x-x_2)$$

则

$$\ln x - p_2(x) = \frac{1}{3\xi^3}(x-11)(x-12)(x-13)$$

其中 $\xi \in [11, 13]$.

对上式两边取绝对值得

$$\begin{aligned} |\ln x - p_2(x)| &= \frac{1}{3\xi^3} |(x-11)(x-12)(x-13)| \\ &\leq \frac{1}{3} \max_{\xi \in [11, 13]} \frac{1}{\xi^3} |(x-11)(x-12)(x-13)| \\ &= \frac{1}{3 \times 11^3} |(x-11)(x-12)(x-13)| \end{aligned}$$

$$\begin{aligned} \text{于是 } |\ln 11.75 - p_2(11.75)| &\leq \frac{1}{3 \times 11^3} |(11.75-11)(11.75-12)(11.75-13)| \\ &\approx 0.000\ 058\ 7 \end{aligned}$$

2.2 n 次插值的 Lagrange 形式和 Newton 形式

2.1 节介绍了线性插值和二次插值, 从这两种插值可以推广到 n 次插值. n 次

插值的提法是:给定 $y=f(x)$ 的函数表(如表 2-2 所示),构造函数 $p_n(x)$ 满足条件:

① $p_n(x)$ 是一个不超过 n 次的代数多项式;

② $p_n(x_i) = y_i (i=0, 1, \dots, n)$.

其中 x_i 称为插值节点; $f(x)$ 称为被插值函数; $p_n(x)$ 称为 n 次插值函数.

n 次插值的目的是构造 $p_n(x)$ 来近似代替 $f(x)$. 当求某一点 x^* 的函数值 $f(x^*)$ 时,可以用 $p_n(x^*)$ 来近似代替 $f(x^*)$.

由于 $p_n(x)$ 是一个不超过 n 次的代数多项式,所以不妨假设

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

其中 a_0, a_1, \dots, a_n 为待定常数.

为了确定 a_0, a_1, \dots, a_n , 可以利用插值条件②得到 $n+1$ 个等式,把它们联立起来组成方程组

$$\begin{cases} p_n(x_0) = a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0 \\ p_n(x_1) = a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = y_1 \\ \vdots \\ p_n(x_n) = a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = y_n \end{cases} \quad (2.15)$$

方程组(2.15)的系数行列式为

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

这是著名的 Vandermonde(范德蒙)行列式,可以用归纳法证明它的值为

$$\begin{aligned} & (x_1 - x_0)(x_2 - x_0) \cdots (x_n - x_0)(x_2 - x_1)(x_3 - x_1) \cdots (x_n - x_1) \cdots (x_n - x_{n-1}) \\ &= \prod_{0 \leq j < i \leq n} (x_i - x_j) \end{aligned}$$

由于 x_0, x_1, \dots, x_n 互不相等,所以 Vandermonde 行列式的值不为 0,从而方程组的解 a_0, a_1, \dots, a_n 存在且惟一. 因此, $p_n(x)$ 存在且惟一.

n 次插值函数 $p_n(x)$ 有两种形式:Lagrange 形式和 Newton 形式.

2.2.1 n 次插值函数的 Lagrange 形式

设 $p_n(x) = l_0(x)y_0 + l_1(x)y_1 + \dots + l_n(x)y_n$

其中 $l_0(x), l_1(x), \dots, l_n(x)$ 为不超过 n 次的代数多项式,且满足表 2-9.

表 2-9 $l_0(x), l_1(x), \cdots, l_n(x)$ 在节点处的取值

| 函数值 节点 函数 | x_0 | x_1 | \cdots | x_n |
|-----------------|----------|----------|----------|----------|
| $l_0(x)$ | 1 | 0 | \cdots | 0 |
| $l_1(x)$ | 0 | 1 | \cdots | 0 |
| \vdots | \vdots | \vdots | | \vdots |
| $l_n(x)$ | 0 | 0 | \cdots | 1 |

容易验证, $p_n(x)$ 满足插值条件①和②. 因此, 只要找到满足表 2-9 的 $l_0(x), l_1(x), \cdots, l_n(x)$, 那么就可构造出 $p_n(x)$. 下面根据表 2-9 构造 $l_0(x), l_1(x), \cdots, l_n(x)$.

首先构造 $l_0(x)$. 根据表 2-9, $l_0(x_i) = 0 (i = 1, 2, \cdots, n)$, 所以 $l_0(x)$ 必有因子 $(x - x_1)(x - x_2) \cdots (x - x_n)$. 又因 $l_0(x)$ 为不超过 n 次的代数多项式, 而 $(x - x_1)(x - x_2) \cdots (x - x_n)$ 已是一个 n 次代数多项式, 故 $l_0(x)$ 只能还有一个常数因子, 从而得到

$$l_0(x) = A(x - x_1)(x - x_2) \cdots (x - x_n)$$

其中 A 为待定常数.

为了确定 A , 利用 $l_0(x_0) = 1$

$$l_0(x_0) = A(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n) = 1$$

得

$$A = \frac{1}{(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n)}$$

于是推得

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

用同样的方法可以求得

$$\begin{aligned} l_1(x) &= \frac{(x - x_0)(x - x_2) \cdots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \cdots (x_1 - x_n)} \\ &\vdots \\ l_n(x) &= \frac{(x - x_0)(x - x_1) \cdots (x - x_{n-1})}{(x_n - x_0)(x_n - x_1) \cdots (x_n - x_{n-1})} \end{aligned}$$

分析一下函数 $l_0(x), l_1(x), \cdots, l_n(x)$, 就会发现它们的变化是有规律的, 可以把它们写成统一形式

$$l_i(x) = \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} \quad (i = 0, 1, \cdots, n)$$

$l_0(x), l_1(x), \cdots, l_n(x)$ 称为插值基函数. 由于它们是不超过 n 次的代数多项

式,所以它们又称为 n 次插值基函数. 有了这 $n+1$ 个 n 次插值基函数后,就得到了满足插值条件的 Lagrange 型 n 次插值函数.

$$\begin{aligned} p_n(x) &= \sum_{i=0}^n l_i(x) y_i \\ &= \sum_{i=0}^n \frac{(x-x_0) \cdots (x-x_{i-1})(x-x_{i+1}) \cdots (x-x_n)}{(x_i-x_0) \cdots (x_i-x_{i-1})(x_i-x_{i+1}) \cdots (x_i-x_n)} y_i \end{aligned} \quad (2.16)$$

(2.16) 式称为 Lagrange 型 n 次插值公式. 特别地, 当 $n=1$ 时, (2.16) 式就是 Lagrange 型线性插值公式; 当 $n=2$ 时, (2.16) 式就是 Lagrange 型二次插值公式.

(2.16) 式还可以写成另一种形式

$$p_n(x) = \sum_{i=0}^n \frac{\omega(x)}{(x-x_i)\omega'(x_i)} y_i \quad (2.17)$$

其中

$$\omega(x) = (x-x_0)(x-x_1) \cdots (x-x_n)$$

$$\omega'(x_i) = (x_i-x_0) \cdots (x_i-x_{i-1})(x_i-x_{i+1}) \cdots (x_i-x_n)$$

对 (2.17) 式, 可以验证它的正确性. Lagrange 型 n 次插值函数的这种形式在数值分析的理论推导方面经常用到.

Lagrange 型 n 次插值算法

① 读入插值节点 $x_i, y_i (i=0, 1, \cdots, n)$ 和点 xx .

② $yy=0$.

③ 对 $i=0, 1, \cdots, n$ 做如下工作

$t=1$

对 $k=0, 1, \cdots, i-1, i+1, \cdots, n$ 做

$$t = t * \frac{xx - x_k}{x_i - x_k}$$

$$yy = yy + t * y_i.$$

④ 输出点 xx 相应的函数近似值 yy .

2.2.2 n 次插值函数的 Newton 形式

2.1 节引入了一阶差商和二阶差商的定义, 在此引入三阶差商的定义.

定义 2.3 若记 $f(x_i, x_j, x_k, x_l) = \frac{f(x_i, x_j, x_k) - f(x_j, x_k, x_l)}{x_i - x_l}$, 则 $f(x_i, x_j, x_k, x_l)$ 称

为函数 $f(x)$ 在节点 x_i, x_j, x_k 和 x_l 处的三阶差商. 其中 x_i, x_j, x_k 和 x_l 互异.

定义 2.4 一般地, n 阶差商的定义为

$$f(x_{i_0}, x_{i_1}, \cdots, x_{i_n}) = \frac{f(x_{i_0}, x_{i_1}, \cdots, x_{i_{n-1}}) - f(x_{i_1}, x_{i_2}, \cdots, x_{i_n})}{x_{i_0} - x_{i_n}}$$

特别地,当 $i_0=0, i_1=1, \cdots, i_n=n$ 时,有

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

从差商的定义出发,用归纳法可以证明

$$f(x_0, x_1, \cdots, x_k) = \sum_{i=0}^k \frac{f(x_i)}{\omega'_k(x_i)} \quad (2.18)$$

其中 $\omega'_k(x_i) = (x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_k)$, $k=1, 2, \cdots, n$.

由(2.18)式可知,无论 x_0, x_1, \cdots, x_n 如何排列,它们的值都是一样的.也就是说, n 阶差商具有对称性.

由一阶差商定义,有

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

从而得到

$$f(x) = f(x_0) + f(x, x_0)(x - x_0)$$

由二阶差商定义,有

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

可以得到

$$f(x, x_0) = f(x_0, x_1) + f(x, x_0, x_1)(x - x_1)$$

同样可以得到

$$f(x, x_0, x_1) = f(x_0, x_1, x_2) + f(x, x_0, x_1, x_2)(x - x_2)$$

\vdots

$$f(x, x_0, x_1, \cdots, x_{n-2}) = f(x_0, x_1, x_2, \cdots, x_{n-1}) + f(x, x_0, x_1, \cdots, x_{n-1})(x - x_{n-1})$$

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

把以上各式由下而上逐步回代可得

$$\begin{aligned} f(x) &= f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \cdots + \\ &\quad f(x_0, x_1, \cdots, x_n)(x - x_0)(x - x_1) \cdots (x - x_{n-1}) + \\ &\quad f(x_0, x_1, \cdots, x_n, x)(x - x_0)(x - x_1) \cdots (x - x_n) \\ &= N_n(x) + R_n(x) \end{aligned} \quad (2.19)$$

其中

$$\begin{aligned} R_n(x) &= f(x_0, x_1, \cdots, x_n, x)(x - x_0)(x - x_1) \cdots (x - x_n) \\ N_n(x) &= f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \cdots + \\ &\quad f(x_0, x_1, \cdots, x_n)(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned} \quad (2.20)$$

因为 $N_n(x)$ 是一个不超过 n 次的代数多项式,并且可以证明

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

故 $N_n(x)$ 满足插值条件①和②,也是一个 n 次插值函数. 这个 n 次插值函数称为 Newton 型 n 次插值函数;(2.20)式称为 Newton 型 n 次插值公式.

从(2.20)式可以看到,当 $n=1$ 时,(2.20)式就是 Newton 型线性插值公式;当

$n = 2$ 时, (2. 20) 式就是 Newton 型二次插值公式.

在实际问题中, 为了得到 Newton 型 n 次插值函数, 通常需要构造一个差商表 (如表 2 - 10 所示), 然后按照差商表写出 Newton 型 n 次插值函数.

表 2 - 10 差商表

| x_i | $y_i = f(x_i)$ | $f'(x_i, x_j)$ | $f(x_i, x_j, x_k)$ | \cdots | $f(x_0, x_1, \cdots, x_n)$ |
|-----------|----------------|-----------------------|--------------------------------|----------|----------------------------|
| x_0 | y_0 | | | | |
| x_1 | y_1 | $f(x_0, x_1)$ | | | |
| x_2 | y_2 | $f(x_1, x_2)$ | $f(x_0, x_1, x_2)$ | | |
| x_3 | y_3 | $f(x_2, x_3)$ | $f(x_1, x_2, x_3)$ | | |
| \vdots | \vdots | \vdots | \vdots | \cdots | |
| x_{n-1} | y_{n-1} | $f(x_{n-2}, x_{n-1})$ | $f(x_{n-3}, x_{n-2}, x_{n-1})$ | | |
| x_n | y_n | $f(x_{n-1}, x_n)$ | $f(x_{n-2}, x_{n-1}, x_n)$ | | $f(x_0, x_1, \cdots, x_n)$ |

例 2 - 3 已知函数表如表 2 - 11 所示. 求 Newton 型 n 次插值函数 $N_n(x)$, 并计算 $f(0.6)$ 的近似值.

表 2 - 11 函数表

| x | 0.4 | 0.55 | 0.65 | 0.8 | 0.9 |
|------------|---------|---------|---------|---------|---------|
| $y = f(x)$ | 0.41075 | 0.57815 | 0.69675 | 0.88811 | 1.02652 |

解 先作差商表如表 2 - 12 所示:

表 2 - 12 例 2 - 3 差商表

| x_i | $y_i = f(x_i)$ | $f(x_i, x_j)$ | $f(x_i, x_j, x_k)$ | $f(x_i, x_j, x_k, x_l)$ | $f(x_0, x_1, \cdots, x_4)$ |
|-------|----------------|---------------|--------------------|-------------------------|----------------------------|
| 0.4 | 0.41075 | | | | |
| 0.55 | 0.57815 | 1.116 | | | |
| 0.65 | 0.69675 | 1.186 | 0.28 | | |
| 0.8 | 0.88811 | 1.27573 | 0.35892 | 0.1973 | |
| 0.9 | 1.02652 | 1.3841 | 0.43348 | 0.21303 | 0.03146 |

根据 (2. 16) 式和差商表可以得到

$$\begin{aligned} N_4(x) &= f(x_0) + f(x_0, x_1)(x - x_0) + \\ &\quad f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \\ &\quad f(x_0, x_1, x_2, x_3)(x - x_0)(x - x_1)(x - x_2) + \\ &\quad f(x_0, x_1, x_2, x_3, x_4)(x - x_0)(x - x_1)(x - x_2)(x - x_3) \\ &= 0.41075 + 1.116(x - 0.4) + 0.28(x - 0.4)(x - 0.55) + \\ &\quad 0.1973(x - 0.4)(x - 0.55)(x - 0.65) + \end{aligned}$$

$$0.03146(x-0.4)(x-0.55)(x-0.65)(x-0.8)$$

于是

$$f(0.6) \approx N_4(0.6) = 0.63665$$

下面推导 Newton 型 n 次插值在计算机上实现的计算格式,并给出在计算机上实现的算法.

$$\text{设} \quad \omega_k = (x - x_0)(x - x_1) \cdots (x - x_{k-1}) \quad (k = 1, 2, \cdots, n)$$

$$\omega_0 = 1$$

$$N_0(x) = y_0 = f(x_0)$$

$$V_{0,k} = y_k = f(x_k) \quad (k = 0, 1, \cdots, n)$$

$$V_{i+1,k} = \frac{V_{i,i} - V_{i,k}}{x_i - x_k} \quad (k = 1, 2, \cdots, n; i = 0, 1, \cdots, k-1)$$

则有

$$\omega_k = (x - x_{k-1}) \omega_{k-1} \quad (k = 1, 2, \cdots, n) \quad (2.21)$$

$$V_{k,k} = f(x_0, x_1, \cdots, x_k) \quad (k = 1, 2, \cdots, n) \quad (2.22)$$

由(2.21)式和(2.22)式可以得到

$$N_k(x) = N_{k-1}(x) + \omega_k V_{k,k} \quad (k = 1, 2, \cdots, n)$$

综合上述各式得到 Newton 型 n 次插值的计算格式为

$$\omega_0 = 1$$

$$N_0(x) = y_0 = f(x_0)$$

$$V_{i+1,k} = \frac{V_{i,i} - V_{i,k}}{x_i - x_k} \quad (k = 1, 2, \cdots, n; i = 0, 1, \cdots, k-1)$$

$$\omega_k = (x - x_{k-1}) \omega_{k-1} \quad (k = 1, 2, \cdots, n)$$

$$N_k(x) = N_{k-1}(x) + \omega_k V_{k,k} \quad (k = 1, 2, \cdots, n)$$

Newton 型 n 次插值算法

① 读入插值节点 $x_i, y_i (i = 0, 1, \cdots, n)$ 和点 xx .

② $\omega = 1, V_0 = y_0, yy = y_0$.

③ 对 $k = 1, 2, \cdots, n$ 做如下工作:

$$V_k = y_k$$

对 $i = 0, 1, \cdots, k-1$ 做

$$V_k = \frac{V_i - V_k}{x_i - x_k}$$

$$\omega = \omega * (xx - x_{k-1})$$

$$yy = yy + \omega * V_k$$

④ 输出插值点 xx 相应的函数近似值 yy .

其中 ω, yy 和 V_k 分别用来存放 $\omega_k, N_k(x)$ 和 $V_{i+1,k}$ 的值.

类似于定理 2.2, 可以证明如下定理.

定理 2.3 设 $p_n(x)$ 是过 $(x_i, y_i) (i=0, 1, \dots, n)$ 的 n 次插值函数, $[a, b]$ 是包含 x_0, x_1, \dots, x_n 的任一区间, 并设 $f(x) \in C^n[a, b]$, $f^{(n+1)}(x)$ 在 $[a, b]$ 上存在, 则对任意给定的 $x \in [a, b]$, 总存在一点 $\xi \in (a, b)$, 使得

$$R(x) = f(x) - p_n(x) = \frac{p^{(n+1)}(\xi)}{(n+1)!} (x-x_0)(x-x_1)\cdots(x-x_n) \quad (2.23)$$

其中 ξ 依赖于 x .

2.3 分段线性插值

n 次插值是根据给定的 $y=f(x)$ 函数表构造代数多项式 $p_n(x)$, $p_n(x)$ 为 n 次插值函数, 用 $p_n(x)$ 近似代替 $f(x)$. 为了表明 $p_n(x)$ 是一个代数多项式, 有时亦称 $p_n(x)$ 为 n 次插值多项式. 按这种说法, 根据给定两个节点的函数表构造出来的插值多项式是一次的, 根据给定三个节点的函数表构造出来的插值多项式是二次的, \dots , 根据给定 $n+1$ 个节点的函数表构造出来的插值多项式是 n 次的.

由此可见, 对于 n 次插值, 随着插值节点的增加, 插值多项式的次数也会增加. 也许有些读者认为, 插值多项式的次数越高, 插值多项式 $p_n(x)$ 与 $f(x)$ 的误差就会越少. 这种想法是片面的.

例 2-4 给定函数

$$f(x) = \frac{1}{1+25x^2} \quad (-1 \leq x \leq 1)$$

取等距插值节点 $x_i = -1 + \frac{1}{5}i (i=0, 1, \dots, 10)$, 试建立插值多项式 $p_{10}(x)$, 并考察 $p_{10}(x)$ 与 $f(x)$ 的误差情况.

解 由 Lagrange 型 n 次插值公式得到

$$p_{10}(x) = \sum_{i=0}^{10} f(x_i) l_i(x)$$

其中

$$x_i = -1 + \frac{1}{5}i \quad (i=0, 1, \dots, n)$$

$$f(x_i) = \frac{1}{1+25x_i^2} \quad (i=0, 1, \dots, n)$$

$$l_i(x) = \frac{(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_{10})}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_{10})} \quad (i=0, 1, \dots, n)$$

画出函数 $f(x) = \frac{1}{1+25x^2}$ 和 $y=p_{10}(x)$ 的图形, 如图 2-3 所示. 从图上可以看到, 在原点附近 $p_{10}(x)$ 能较好地逼近 $f(x)$, 但在其他地方 $p_{10}(x)$ 与 $f(x)$ 的误差较

大,越靠近两个端点 -1 和 1 误差就越大,还出现激烈振荡现象.采用高次插值发生的这种现象称为 Runge(龙格)现象.

还可以证明,用 $p_n(x)$ 逼近 $f(x) = \frac{1}{1+25x^2}$,

当 $n \rightarrow \infty$ 时, $p_n(x)$ 只能在 $|x| < 0.72 \dots$ 内收敛,而在这个区间外是发散的.

从例 2-4 可以得出如下结论:高次插值多项式并不一定能很好地近似代替被插值函数.因此,在实际进行插值的时候,如果给定的插值节点很多,通常把插值区间分为若干段,然后在每个分段上使用低次插值多项式来近似代替 $f(x)$,即在每个分段上采用低次插值.这种做法称为分段插值.

如果在每个分段上使用不超过一次的代数多项式来近似代替被插值函数 $y=f(x)$,则这种插值称为分段线性插值.

分段线性插值的提法是:根据给定函数 $y=f(x)$ 的函数表(如表 2-2 所示),构造函数 $p(x)$ 满足条件:

- ① $p(x)$ 在 $[x_i, x_{i+1}]$ ($i=0, 1, \dots, n-1$) 上为不超过一次的代数多项式;
- ② $p(x_i) = y_i$ ($i=0, 1, \dots, n$).

其中 $p(x)$ 称为分段线性插值函数.

分段线性插值的目的是分段构造 $p(x)$ 来近似代替 $f(x)$.当要求某一点 x^* 的函数值 $f(x^*)$ 时,可以用 $p(x^*)$ 近似代替 $f(x^*)$.

根据分段线性插值的条件,可以把 $p(x)$ 直接构造出来:

$$p(x) = \begin{cases} \frac{x-x_1}{x_0-x_1}y_0 + \frac{x-x_0}{x_1-x_0}y_1 & (x_0 \leq x \leq x_1) \\ \frac{x-x_2}{x_1-x_2}y_1 + \frac{x-x_1}{x_2-x_1}y_2 & (x_1 < x \leq x_2) \\ \vdots \\ \frac{x-x_n}{x_{n-1}-x_n}y_{n-1} + \frac{x-x_{n-1}}{x_n-x_{n-1}}y_n & (x_{n-1} < x \leq x_n) \end{cases} \quad (2.24)$$

分段线性插值函数也可以由插值基函数组合而成.设 $l_0(x), l_1(x), \dots, l_n(x)$ 为不超过 n 次的代数多项式,且满足表 2-13.容易验证下面函数满足表 2-13.

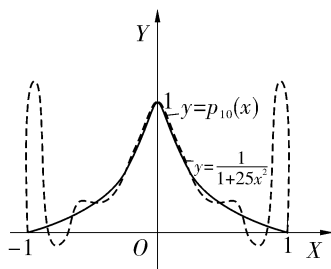


图 2-3 函数 $f(x) = \frac{1}{1+25x^2}$ 和 $y=p_{10}(x)$ 的图形

表 2-13 $l_0(x), l_1(x), \cdots, l_n(x)$ 在节点处的取值

| 函数值 函数 \ 节点 | x_0 | x_1 | \cdots | x_n |
|----------------|----------|----------|----------|----------|
| $l_0(x)$ | 1 | 0 | \cdots | 0 |
| $l_1(x)$ | 0 | 1 | \cdots | 0 |
| \vdots | \vdots | \vdots | | \vdots |
| $l_n(x)$ | 0 | 0 | \cdots | 1 |

$$l_0(x) = \begin{cases} \frac{x-x_1}{x_0-x_1} & (x_0 \leq x \leq x_1) \\ 0 & (x_1 < x \leq x_n) \end{cases}$$
$$l_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & (x_{i-1} \leq x \leq x_i) \\ \frac{x-x_{i+1}}{x_i-x_{i+1}} & (x_i < x \leq x_{i+1}) \\ 0 & [a,b] - [x_{i-1}, x_{i+1}] \quad (i=1,2,\cdots,n-1) \end{cases}$$
$$\vdots$$
$$l_n(x) = \begin{cases} \frac{x-x_{n-1}}{x_n-x_{n-1}} & (x_{n-1} \leq x \leq x_n) \\ 0 & (x_0 \leq x < x_{n-1}) \end{cases}$$

$l_0(x), l_1(x), \cdots, l_n(x)$ 通常称为分段线性插值基函数, 由这些分段线性插值基函数可以得到分段线性插值函数的另一种形式

$$p(x) = \sum_{i=0}^n y_i l_i(x)$$

分段线性插值的几何意义是用一条折线 $y=p(x)$ 来近似代替曲线 $y=f(x)$.

例 2-5 给定函数 $y=f(x)$ 的函数表如表 2-14 所示, 试用分段线性插值法求 $y=f(x)$ 在 $x=-0.9$ 处的近似值.

表 2-14 函数表

| | | | | | |
|----------|----------|----------|------|------|-----|
| x | -1 | -0.8 | -0.6 | -0.4 | 0 |
| $y=f(x)$ | 0.038 46 | 0.058 82 | 0.1 | 0.2 | 0.5 |

解

$$p(x) = \begin{cases} \frac{x+0.8}{-0.2} \times 0.03846 + \frac{x+1}{0.2} \times 0.05882 & (-1 \leq x \leq -0.8) \\ \frac{x+0.6}{-0.2} \times 0.05882 + \frac{x+0.8}{0.2} \times 0.1 & (-0.8 < x \leq -0.6) \\ \frac{x+0.4}{-0.2} \times 0.1 + \frac{x+0.6}{0.2} \times 0.2 & (-0.6 < x \leq -0.4) \\ \frac{x}{-0.4} \times 0.2 + \frac{x+0.4}{0.4} \times 0.5 & (-0.4 < x \leq 0) \end{cases}$$

$$f(-0.9) \approx p(-0.9) = 0.04864$$

分段线性插值算法

①读入插值节点 $x_i, y_i (i=0, 1, \dots, n)$ 和点 xx .②对 $i=1, 2, \dots, n$ 做:如果 $xx < x_i$ 则

$$yy = \frac{xx - x_{i-1}}{x_i - x_{i-1}} y_{i-1} + \frac{xx - x_{i-1}}{x_i - x_{i-1}} y_i$$

输出点 xx 相应的函数近似值 yy

转到③

③结束.

定理 2.4 设给定如表 2-2 所示的 $y=f(x)$ 的函数表, 令 $a=x_0, b=x_n$, $f(x) \in C^1[a, b]$, $f''(x)$ 在 $[a, b]$ 上存在, $p(x)$ 是 $f(x)$ 的分段线性插值函数, 则有

$$|R(x)| = |f(x) - p(x)| \leq \frac{h^2}{8} M \quad (2.25)$$

其中 $h = \max_{0 \leq i \leq n-1} |x_{i+1} - x_i|$; $M = \max_{a \leq x \leq b} |f''(x)|$.

证 因为 $p(x)$ 在每个小区间 $[x_i, x_{i+1}] (i=0, 1, \dots, n-1)$ 上都是线性函数, 所以由(2.7)式可知, 在 $[x_i, x_{i+1}]$ 上有

$$\begin{aligned} |R(x)| &\leq \frac{(x_{i+1} - x_i)^2}{8} \max_{x_i \leq x \leq x_{i+1}} |f''(x)| \\ &\leq \frac{\max_{0 \leq i \leq n-1} |x_{i+1} - x_i|^2}{8} \max_{a \leq x \leq b} |f''(x)| \end{aligned}$$

若记 $h = \max_{0 \leq i \leq n-1} |x_{i+1} - x_i|$; $M = \max_{a \leq x \leq b} |f''(x)|$, 则有

$$|R(x)| = |f(x) - p(x)| \leq \frac{h^2}{8} M$$

2.4 Hermite 插值

在某些实际问题中,为了保证插值函数更好地逼近被插值函数 $f(x)$,不仅要求插值函数在插值节点上的值与被插值函数 $f(x)$ 在插值节点上的值相等,而且还要求插值函数在插值节点上的导数值与被插值函数 $f(x)$ 在插值节点上的导数值相等. 解决这样的问题需要用到 Hermite 插值.

2.4.1 三次 Hermite 插值

三次 Hermite 插值的提法是:

根据给定的 $y = f(x)$ 的函数表(如表 2-15 所示),构造函数 $H(x)$ 满足条件:

① $H(x)$ 为不超过三次的代数多项式;

② $H(x_0) = y_0, H(x_1) = y_1; H'(x_0) = m_0$

和 $H'(x_1) = m_1$.

表 2-15 三次 Hermite 插值函数表

| x | x_0 | x_1 |
|--------------|-------|-------|
| $y = f(x)$ | y_0 | y_1 |
| $y' = f'(x)$ | m_0 | m_1 |

其中 $H(x)$ 称为三次 Hermite 插值函数.

三次 Hermite 插值的目的是构造 $H(x)$ 来近似代替 $f(x)$. 当求某一点 x^* 的函数值 $f(x^*)$ 时,可以用 $H(x^*)$ 近似代替 $f(x^*)$.

设 $H(x) = y_0 h_0(x) + y_1 h_1(x) + m_0 H_0(x) + m_1 H_1(x)$

其中 $h_0(x), h_1(x), H_0(x)$ 和 $H_1(x)$ 为不超过三次的代数多项式,且满足表 2-16.

表 2-16 $h_0(x), h_1(x), H_0(x)$ 和 $H_1(x)$ 在节点处的取值

| | 函数值 | | 导数值 | |
|----------|-------|-------|-------|-------|
| | x_0 | x_1 | x_0 | x_1 |
| $h_0(x)$ | 1 | 0 | 0 | 0 |
| $h_1(x)$ | 0 | 1 | 0 | 0 |
| $H_0(x)$ | 0 | 0 | 1 | 0 |
| $H_1(x)$ | 0 | 0 | 0 | 1 |

容易验证, $H(x)$ 就是满足插值条件①和②的 Hermite 插值函数. 下面要把 $h_0(x), h_1(x), H_0(x)$ 和 $H_1(x)$ 构造出来.

先构造 $h_0(x)$. 从表 2-16 可以看出,当 $x = x_1$ 时,函数 $h_0(x)$ 的值为 0,同时函数 $h_0(x)$ 的导数也为 0,所以它必有因子 $(x - x_1)^2$. 另外, $h_0(x)$ 是一个不超过三次的代数多项式,从而 $h_0(x)$ 可以表示为

$$h_0(x) = [a + b(x - x_0)] \left[\frac{x - x_1}{x_0 - x_1} \right]^2$$

其中 a 和 b 为待定常数.

由 $h_0(x_0) = 1$ 得 $a = 1$.

为了确定 b , 对 $h_0(x)$ 求微商, 得

$$h'_0(x) = b \left[\frac{x - x_1}{x_0 - x_1} \right]^2 + [1 + b(x - x_0)] \times 2 \times \left[\frac{x - x_1}{x_0 - x_1} \right] \times \frac{1}{x_0 - x_1}$$

再由 $h'_0(x_0) = 0$ 得到 $b = \frac{2}{x_1 - x_0}$.

于是

$$h_0(x) = \left[1 + 2 \frac{x - x_0}{x_1 - x_0} \right] \left[\frac{x - x_1}{x_0 - x_1} \right]^2 \quad (2.26)$$

用同样方法可以求得

$$h_1(x) = \left[1 + 2 \frac{x - x_1}{x_0 - x_1} \right] \left[\frac{x - x_0}{x_1 - x_0} \right]^2 \quad (2.27)$$

再构造 $H_0(x)$. 从表 2-16 可以看到, $H_0(x)$ 在 x_0 处的函数值为 0, 在 x_1 上不仅函数值为 0, 而且微商也为 0, 所以它必有因子 $(x - x_0)(x - x_1)^2$. 又因它是一个不超过三次的代数多项式, 而 $(x - x_0)(x - x_1)^2$ 已是一个三次代数多项式, 从而 $H_0(x)$ 可以表示为

$$H_0(x) = c(x - x_0) \left[\frac{x - x_1}{x_0 - x_1} \right]^2$$

其中 c 为待定常数.

为了确定 c , 对 $H_0(x)$ 求微商得

$$H'_0(x) = c \left[\left[\frac{x - x_1}{x_0 - x_1} \right]^2 + 2(x - x_0) \frac{x - x_1}{(x_0 - x_1)^2} \right]$$

由 $H'_0(x_0) = 1$, 得到 $c = 1$. 于是

$$H_0(x) = (x - x_0) \left[\frac{x - x_1}{x_0 - x_1} \right]^2 \quad (2.28)$$

用同样的方法可求得

$$H_1(x) = (x - x_1) \left[\frac{x - x_0}{x_1 - x_0} \right]^2 \quad (2.29)$$

因此

$$\begin{aligned} H(x) &= y_0 h_0(x) + y_1 h_1(x) + m_0 H_0(x) + m_1 H_1(x) \\ &= \left[1 + 2 \frac{x - x_0}{x_1 - x_0} \right] \left[\frac{x - x_1}{x_0 - x_1} \right]^2 y_0 + \left[1 + 2 \frac{x - x_1}{x_0 - x_1} \right] \left[\frac{x - x_0}{x_1 - x_0} \right]^2 y_1 + \end{aligned}$$

$$(x-x_0)\left[\frac{x-x_1}{x_0-x_1}\right]^2 m_0 + (x-x_1)\left[\frac{x-x_0}{x_1-x_0}\right]^2 m_1 \quad (2.30)$$

(2.30)式也称为三次 Hermite 插值公式.

例 2-6 给定函数 $y=f(x)$ 的函数表如表 2-17 所示. 构造一个三次 Hermite 插值函数, 并求出 $f\left[\frac{1}{2}\right]$ 的近似值.

表 2-17 函数表

| | | |
|------------|---|---|
| x | 0 | 1 |
| $y=f(x)$ | 0 | 1 |
| $y'=f'(x)$ | 3 | 9 |

解 由 (2.26) 式、(2.27) 式、(2.28) 式和 (2.29) 式得到

$$h_0(x) = (1+2x)(x-1)^2$$

$$h_1(x) = [1-2(x-1)]x^2$$

$$H_0(x) = x(x-1)^2$$

$$H_1(x) = (x-1)x^2$$

从而由 (2.30) 式得

$$\begin{aligned} H(x) &= y_0 h_0(x) + y_1 h_1(x) + m_0 H_0(x) + m_1 H_1(x) \\ &= [1-2(x-1)]x^2 + 3x(x-1)^2 + 9(x-1)x^2 \end{aligned}$$

因此

$$f\left[\frac{1}{2}\right] \approx H\left[\frac{1}{2}\right] = -\frac{1}{4}$$

定理 2.5 设 $H(x)$ 是 $f(x)$ 的三次 Hermite 插值函数, $[a, b]$ 是包含 x_0, x_1 的任一区间, 并设 $f(x) \in C^3[a, b]$, $f^{(4)}(x)$ 在 $[a, b]$ 上存在, 则对任意给定的 $x \in [a, b]$, 总存在一点 $\xi \in (a, b)$, 使得

$$R(x) = f(x) - H(x) = \frac{f^{(4)}(\xi)}{4!} (x-x_0)^2 (x-x_1)^2 \quad (2.31)$$

其中 ξ 依赖于 x .

证 设 x 是 $[a, b]$ 中的任一固定数, 若 $x = x_i (i=0, 1)$, 则 (2.31) 式成立. 下面考虑 x 不是插值节点的情况.

引入辅助函数

$$\varphi(t) = f(t) - H(t) - \frac{R(x)}{(x-x_0)^2(x-x_1)^2} (t-x_0)^2 (t-x_1)^2$$

易知 $\varphi(t)$ 在 $[a, b]$ 上具有四阶连续导数, 并有 x_0, x_1 和 x 3 个不同零点, 其中 x_0, x_1 为二重零点. 因此 $\varphi'(t)$ 在 $\varphi(t)$ 的每两个零点间至少存在一个零点, 记这两个零点为 η_0, η_1 , 从而 $\varphi'(t)$ 在 $[a, b]$ 上至少有 4 个零点 x_0, x_1, η_0, η_1 . 故 $\varphi''(t)$ 在 $[a, b]$ 上至少有 3 个零点. 按此类推, $\varphi^{(4)}(t)$ 在 $[a, b]$ 上至少有一个零点, 把该零点记为 ξ , 即有 $\varphi^{(4)}(\xi) = 0$.

对 $\varphi(t)$ 求 4 次微商, 并注意到 $H(t)$ 为不超过三次的代数多项式, 故

$$\varphi^{(4)}(t) = f^{(4)}(t) - 4! \frac{R(x)}{(x-x_0)^2(x-x_1)^2}$$

把 ξ 代入上式,并注意到 $\varphi^{(4)}(\xi) = 0$,移项后得

$$R(x) = \frac{f^{(4)}(\xi)}{4!}(x-x_0)^2(x-x_1)^2$$

2.4.2 $2n+1$ 次 Hermite 插值

$2n+1$ 次 Hermite 插值的提法是:

给定 $y=f(x)$ 的函数表(如表 2-18 所示),构造函数 $H(x)$ 满足条件:

- ① $H(x)$ 为不超过 $2n+1$ 次的代数多项式;
- ② $H(x_i) = y_i, H'(x_i) = m_i \quad (i=0,1,\cdots,n)$.

表 2-18 $2n+1$ 次 Hermite 插值函数表

| | | | | |
|------------|-------|-------|----------|-------|
| x | x_0 | x_1 | \cdots | x_n |
| $y=f(x)$ | y_0 | y_1 | \cdots | y_n |
| $y'=f'(x)$ | m_0 | m_1 | \cdots | m_n |

$H(x)$ 称为 $2n+1$ 次 Hermite 插值函数.

$2n+1$ 次 Hermite 插值的目的是构造 $H(x)$ 来近似代替 $f(x)$. 当求某一点 x^* 的函数值 $f(x^*)$ 时,可以用 $H(x^*)$ 近似代替 $f(x^*)$.

设
$$H(x) = \sum_{i=0}^n [y_i h_i(x) + m_i H_i(x)]$$

$i=0,1,\cdots,n$. 其中 $h_i(x)$ 和 $H_i(x)$ 为不超过 $2n+1$ 次的代数多项式,且满足表 2-19.

表 2-19 $h_i(x)$ 和 $H_i(x)$ 的取值

| | 函数值 | | | | 导数值 | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | x_0 | x_1 | \cdots | x_n | x_0 | x_1 | \cdots | x_n |
| $h_0(x)$ | 1 | 0 | \cdots | 0 | 0 | 0 | \cdots | 0 |
| $h_1(x)$ | 0 | 1 | \cdots | 0 | 0 | 0 | \cdots | 0 |
| \vdots | \vdots | \vdots | | \vdots | \vdots | \vdots | | \vdots |
| $h_n(x)$ | 0 | 0 | \cdots | 1 | 0 | 0 | \cdots | 0 |
| $H_0(x)$ | 0 | 0 | \cdots | 0 | 1 | 0 | \cdots | 0 |
| $H_1(x)$ | 0 | 0 | \cdots | 0 | 0 | 1 | \cdots | 0 |
| \vdots | \vdots | \vdots | | \vdots | \vdots | \vdots | | \vdots |
| $H_n(x)$ | 0 | 0 | \cdots | 0 | 0 | 0 | \cdots | 1 |

容易验证, $H(x)$ 满足插值的两个条件, 即 $H(x)$ 就是一个 $2n+1$ 次 Hermite 插值函数. 下面构造 $h_i(x)$ 和 $H_i(x)$.

先构造 $h_0(x)$. 因为 $h_0(x_i) = h'_0(x_i) = 0$, 所以 $h_0(x)$ 必有因子 $[(x-x_1)(x-x_2)\cdots(x-x_n)]^2$. 又因 $h_0(x)$ 为不超过 $2n+1$ 次的代数多项式, 故 $h_0(x)$ 可以表示为

$$h_0(x) = (ax+b)l_0^2(x)$$

$$\text{其中 } a \text{ 和 } b \text{ 为待定常数; } l_0(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}.$$

又因 $h_0(x_0) = 1$, 故有 $ax_0 + b = 1$, 即 $b = 1 - ax_0$.

对 $h_0(x)$ 求一阶导数得

$$h'_0(x) = al'_0(x) + 2(ax+b)l_0(x)l'_0(x)$$

由 $h'_0(x_0) = 0$, 故有

$$h'_0(x_0) = a + 2l'_0(x_0) = 0$$

从而得到

$$a = -2l'_0(x_0)$$

$$b = 1 + 2x_0l'_0(x_0)$$

$$h_0(x) = [1 - 2(x-x_0)l'_0(x_0)]l_0^2(x)$$

为求 $l'_0(x_0)$, 对 $l_0(x)$ 两边取对数得

$$\begin{aligned} \ln[l_0(x)] &= \ln(x-x_1) + \ln(x-x_2) + \cdots + \ln(x-x_n) - \\ &\quad [\ln(x_0-x_1) + \ln(x_0-x_2) + \cdots + \ln(x_0-x_n)] \end{aligned}$$

对上式两边求导数得

$$\frac{1}{l_0(x)}l'_0(x) = \sum_{j=1}^n \frac{1}{x-x_j}$$

从而有

$$l'_0(x) = l_0(x) \left[\sum_{j=1}^n \frac{1}{x-x_j} \right]$$

由此推得

$$l'_0(x_0) = \sum_{j=1}^n \frac{1}{x_0-x_j}$$

于是得到

$$h_0(x) = \left[1 - 2(x-x_0) \left[\sum_{j=1}^n \frac{1}{x_0-x_j} \right] \right] l_0^2(x) \quad (2.32)$$

用同样的方法可以得到

$$h_i(x) = \left[1 - 2(x-x_i) \left[\sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{x_i-x_j} \right] \right] l_i^2(x) \quad (i=1, 2, \cdots, n) \quad (2.33)$$

再来构造 $H_i(x)$, $i=0, 1, \cdots, n$. 先构造 $H_0(x)$. 因为 $H_0(x_0) = 0$ 和 $H_0(x_i) = H'_0(x_i) = 0$ ($i=1, 2, \cdots, n$), 所以 $H_0(x)$ 必有因子 $(x-x_0)[(x-x_1)\cdots(x-x_n)]^2$. 又因

$H_0(x)$ 为不超过 $2n+1$ 次的代数多项式, 故 $H_0(x)$ 可以表示为

$$H_0(x) = a(x - x_0)l_0^2(x)$$

其中 a 为待定常数.

对 $H_0(x)$ 求一阶导数得

$$H'_0(x) = a[l_0^2(x) + 2(x - x_0)l_0(x)l'_0(x)]$$

由 $H'_0(x_0) = 1$ 得 $a = 1$. 于是得到

$$H_0(x) = (x - x_0)l_0^2(x) \quad (2.34)$$

用同样的方法可以得到

$$H_i(x) = (x - x_i)l_i^2(x) \quad (i = 1, 2, \dots, n) \quad (2.35)$$

因此, $2n+1$ 次 Hermite 插值函数为

$$\begin{aligned} H(x) &= \sum_{i=0}^n [y_i h_i(x) + m_i H_i(x)] \\ &= \sum_{i=0}^n \left\{ y_i \left[1 - 2(x - x_i) \left[\sum_{j \neq i}^n \frac{1}{x_i - x_j} \right] \right] l_i^2(x) + m_i (x - x_i) l_i^2(x) \right\} \end{aligned} \quad (2.36)$$

类似于定理 2.5, 可以证明如下定理.

定理 2.6 设给定如表 2-18 所示的函数表, $H(x)$ 是 $2n+1$ 次 Hermite 插值函数, $[a, b]$ 是包含 x_0, x_1, \dots, x_n 的任一区间, 并设 $f(x) \in C^{2n+1}[a, b]$, $f^{(2n+2)}(x)$ 在 $[a, b]$ 上存在, 则对任意给定的 $x \in [a, b]$, 总存在一点 $\xi \in (a, b)$, 使得

$$R(x) = f(x) - H(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \omega^2(x) \quad (2.37)$$

其中 ξ 依赖于 x , $\omega(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$.

分段三次 Hermite 插值

在 2.3 中介绍的分段线性插值函数在节点处的导数是不连续的. 如果除了给定被插值函数在节点处的函数值外, 还给出被插值函数在节点处的导数值, 要想构造一个在节点处的导数也连续的插值函数可以使用分段三次 Hermite 插值.

分段三次 Hermite 插值的提法是:

给定函数 $y = f(x)$ 的函数表 (如表 2-12 所示), 构造函数 $q(x)$ 满足条件:

① $q(x)$ 在 $[x_i, x_{i+1}]$ ($i = 0, 1, \dots, n-1$) 上为三次代数多项式;

② $q(x_i) = y_i$ ($i = 0, 1, \dots, n$);

③ $q(x) \in C^1[a, b]$, 其中 $a = x_0, b = x_n$.

其中 $q(x)$ 称为分段三次 Hermite 插值函数.

根据分段三次 Hermite 插值条件和三次 Hermite 插值公式 (2.30), 可以把 $q(x)$ 直接构造出来. 在区间 $[x_i, x_{i+1}]$ 上有

$$q(x) = \left[1 + 2 \frac{x - x_i}{x_{i+1} - x_i} \right] \left[\frac{x - x_{i+1}}{x_i - x_{i+1}} \right]^2 y_i + \left[1 + 2 \frac{x - x_{i+1}}{x_i - x_{i+1}} \right] \left[\frac{x - x_i}{x_{i+1} - x_i} \right]^2 y_{i+1} + (x - x_i) \left[\frac{x - x_{i+1}}{x_i - x_{i+1}} \right]^2 m_i + (x - x_{i+1}) \left[\frac{x - x_i}{x_{i+1} - x_i} \right]^2 m_{i+1} \quad (2.38)$$

分段三次 Hermite 插值函数 $q(x)$ 也可以由插值基函数组合而成. 若在插值区间 $[a, b]$ 上定义一组分段三次 Hermite 插值基函数 $h_i(x)$ 和 $H_i(x)$ ($i=0, 1, \dots, n$), 则由 (2.38) 式得

$$q(x) = \sum_{i=0}^n [y_i h_i(x) + m_i H_i(x)] \quad (2.39)$$

其中 $h_i(x)$ 和 $H_i(x)$ 分别为:

$$h_0(x) = \begin{cases} \left[1 + 2 \frac{x - x_0}{x_1 - x_0} \right] \left[\frac{x - x_1}{x_0 - x_1} \right]^2 & (x_0 \leq x \leq x_1) \\ 0 & (x_1 < x \leq x_n) \end{cases}$$

$$h_i(x) = \begin{cases} \left[1 + 2 \frac{x - x_i}{x_{i-1} - x_i} \right] \left[\frac{x - x_{i-1}}{x_i - x_{i-1}} \right]^2 & (x_{i-1} \leq x \leq x_i) \\ \left[1 + 2 \frac{x - x_i}{x_{i+1} - x_i} \right] \left[\frac{x - x_{i+1}}{x_i - x_{i+1}} \right]^2 & (x_i < x \leq x_{i+1}) \\ 0 & [a, b] - [x_{i-1} - x_{i+1}] \end{cases}$$

$$(i = 1, 2, \dots, n-1)$$

$$h_n(x) = \begin{cases} \left[1 + 2 \frac{x - x_n}{x_{n-1} - x_n} \right] \left[\frac{x - x_{n-1}}{x_n - x_{n-1}} \right]^2 & (x_{n-1} \leq x \leq x_n) \\ 0 & (x_0 \leq x < x_{n-1}) \end{cases}$$

$$H_0(x) = \begin{cases} (x - x_0) \left[\frac{x - x_1}{x_0 - x_1} \right]^2 & (x_0 \leq x \leq x_1) \\ 0 & (x_1 < x \leq x_n) \end{cases}$$

$$H_i(x) = \begin{cases} (x - x_i) \left[\frac{x - x_{i-1}}{x_i - x_{i-1}} \right]^2 & (x_{i-1} \leq x \leq x_i) \\ (x - x_i) \left[\frac{x - x_{i+1}}{x_i - x_{i+1}} \right]^2 & (x_i < x \leq x_{i+1}) \\ 0 & [a, b] - [x_{i-1} - x_{i+1}] \end{cases}$$

$$(i = 1, 2, \dots, n-1)$$

$$H_n(x) = \begin{cases} (x - x_n) \left[\frac{x - x_{n-1}}{x_n - x_{n-1}} \right]^2 & (x_{n-1} \leq x \leq x_n) \\ 0 & (x_0 \leq x < x_{n-1}) \end{cases}$$

分段三次 Hermite 插值算法

①读入插值节点 x_i , 函数值 y_i , 导数值 $m_i (i=0, 1, \dots, n)$ 和点 xx

②对 $i=0, 1, \dots, n-1$, 如果 $xx < x_{i+1}$, 则

$$yy = \left[1 + 2 \frac{xx - x_i}{x_{i+1} - x_i} \right] \left[\frac{xx - x_{i+1}}{x_i - x_{i+1}} \right]^2 y_i + \left[1 + 2 \frac{xx - x_{i+1}}{x_i - x_{i+1}} \right] \left[\frac{xx - x_i}{x_{i+1} - x_i} \right]^2 y_{i+1} + \\ (xx - x_i) \left[\frac{xx - x_{i+1}}{x_i - x_{i+1}} \right]^2 m_i + (xx - x_{i+1}) \left[\frac{xx - x_i}{x_{i+1} - x_i} \right]^2 m_{i+1}$$

输出点 xx 相应的函数近似值 yy . 转到③

③结束

定理 2.7 设给定如表 2-13 所示的函数表, 令 $a = x_0, b = x_n, f(x) \in C^3[a, b], f^{(4)}(x)$ 在 $[a, b]$ 上存在, $q(x)$ 是 $f(x)$ 的分段三次 Hermite 插值函数, 则有

$$|R(x)| = |f(x) - q(x)| \leq \frac{h^4}{384} M \quad (2.40)$$

其中 $h = \max_{0 \leq i \leq n-1} |x_{i+1} - x_i|, M = \max_{a \leq x \leq b} |f^{(4)}(x)|$.

证明 由于 $q(x)$ 在每个小区间 $[x_i, x_{i+1}] (i=0, 1, \dots, n-1)$ 上都是三次 Hermite 插值函数, 所以由 (2.31) 式可知, 在 $[x_i, x_{i+1}] (i=0, 1, \dots, n-1)$ 上有

$$R(x) = f(x) - q(x) = \frac{f^{(4)}(\xi_i)}{4!} (x - x_i)^2 (x - x_{i+1})^2$$

其中 ξ_i 依赖于 x , 且 $x_i < \xi_i < x_{i+1}$.

因为
$$\max_{x_i \leq x \leq x_{i+1}} |(x - x_i)^2 (x - x_{i+1})^2| = \frac{(x_{i+1} - x_i)^4}{16}$$

故
$$\max_{x_i \leq x \leq x_{i+1}} |R(x)| \leq \frac{(x_{i+1} - x_i)^4}{384} \max_{x_i \leq x \leq x_{i+1}} |f^{(4)}(x)|$$

若记 $h = \max_{0 \leq i \leq n-1} |x_{i+1} - x_i|, M = \max_{a \leq x \leq b} |f^{(4)}(x)|$, 则在 $[a, b]$ 上有

$$|R(x)| = |f(x) - q(x)| \leq \frac{h^4}{384} M$$

2.5 三次样条插值

前面已经介绍过怎样根据给定函数表的节点构造插值多项式. 当函数表的节点很多时, 构造出来的是一个高次插值多项式, 而高次插值多项式与被插值函数 $f(x)$ 的误差有时比较大. 为了减少误差, 可以采用分段插值. 但分段插值也有缺点, 那就是插值函数在节点处不够光滑. 例如, 分段线性插值函数只保证在节点处连续, 它的导数在节点处就不连续了. 又如, 分段三次 Hermite 插值也只是一阶导数

连续,二阶导数就不连续了,而往往实际问题中有时很难给出导数条件.但在一些实际问题中,不仅要求构造的插值函数在节点处连续,一阶导数在节点处连续,而且还要求插值函数在节点处二阶导数也连续.换句话说,就是要求提高插值函数在节点处的光滑度,同时减少对节点处导数的要求.为了提高插值函数在节点处的光滑度,可以采用样条插值.

样条插值需要用到样条函数.什么叫样条函数?这要从工程中的样条曲线谈起.在工程实践中,比如造船、造飞机等,绘图员为了在钢板或铝板上将一些点连成一条光滑曲线,往往用细长的木条把这些点连接起来.对这条光滑曲线进行数学模拟就得到一个函数,由于这个函数与木条有联系,而木条又称为样条,所以该函数称为样条函数.

样条函数有一次、二次、三次或更高次的样条函数.三次样条函数是一个分段代数多项式,在每一个分段上它是一个不超过三次的代数多项式,它在节点上连续,其一阶导数和二阶导数在节点上也连续.因而,三次样条函数属于 $C^2[a, b]$,这是三次样条函数的一个突出优点.由于有这样的优点,三次样条函数在实际中得到广泛的应用.三次样条函数用在插值中就产生了一种插值方法,这种方法称为三次样条插值.

三次样条插值的提法是:

给定 $y=f(x)$ 的函数表(如表 2-2 所示),构造函数 $s(x)$ 满足条件:

①在 $[x_i, x_{i+1}]$ ($i=0, 1, \dots, n-1$) 上为不超过三次的代数多项式;

② $s(x_i) = y_i$ ($i=0, 1, \dots, n$);

③ $s(x) \in C^2[a, b]$, 其中 $a = x_0, b = x_n$.

其中 $s(x)$ 称为三次样条插值函数.

下面讨论如何构造三次样条插值函数 $s(x)$.

设 $s(x)$ 在点 x_i ($i=0, 1, \dots, n$) 处的微商为 m_i ($i=0, 1, \dots, n$), 则 $s(x)$ 在每个小区间 $[x_i, x_{i+1}]$ 上有

$$s(x_i) = y_i \quad s'(x_i) = m_i \quad s(x_{i+1}) = y_{i+1} \quad s'(x_{i+1}) = m_{i+1}$$

从而 $s(x)$ 在区间 $[x_i, x_{i+1}]$ 上满足三次 Hermite 插值的两个条件.按照三次 Hermite 插值公式,在区间 $[x_i, x_{i+1}]$ 上有

$$s(x) = \left[1 + 2 \frac{x - x_i}{x_{i+1} - x_i} \right] \left[\frac{x - x_{i+1}}{x_i - x_{i+1}} \right]^2 y_i + \left[1 + 2 \frac{x - x_{i+1}}{x_i - x_{i+1}} \right] \left[\frac{x - x_i}{x_{i+1} - x_i} \right]^2 y_{i+1} + (x - x_i) \left[\frac{x - x_{i+1}}{x_i - x_{i+1}} \right]^2 m_i + (x - x_{i+1}) \left[\frac{x - x_i}{x_{i+1} - x_i} \right]^2 m_{i+1} \quad (2.41)$$

在(2.41)式中, m_i 是未知的.为了求 m_i , 可以利用插值条件③,也就是利用 $s(x)$ 在 x_i ($i=0, 1, \dots, n$) 上具有连续的二阶微商的性质.

首先,令 $h_i = x_{i+1} - x_i$, 由(2.41)式得到

$$s(x) = \left[1 + 2 \frac{x - x_i}{h_i} \right] \frac{(x - x_{i+1})^2}{h_i^2} y_i + \left[1 - 2 \frac{x - x_{i+1}}{h_i} \right] \frac{(x - x_i)^2}{h_i^2} y_{i+1} + \\ (x - x_i) \frac{(x - x_{i+1})^2}{h_i^2} m_i + (x - x_{i+1}) \frac{(x - x_i)^2}{h_i^2} m_{i+1} \quad (2.42)$$

从而在 $[x_i, x_{i+1}]$ 上有

$$s''(x) = \left[\frac{6}{h_i^2} - \frac{12}{h_i^3} (x_{i+1} - x) \right] y_i + \left[\frac{6}{h_i^2} - \frac{12}{h_i^3} (x - x_i) \right] y_{i+1} + \\ \left[\frac{2}{h_i} - \frac{6}{h_i^2} (x_{i+1} - x) \right] m_i - \left[\frac{2}{h_i} - \frac{6}{h_i^2} (x - x_i) \right] m_{i+1} \quad (2.43)$$

分别把(2.42)式和(2.43)式中的下标 $i+1$ 改为 i , i 改为 $i-1$, 则在区间 $[x_{i-1}, x_i]$ 上有

$$s(x) = \left[1 + 2 \frac{x - x_{i-1}}{h_{i-1}} \right] \frac{(x - x_i)^2}{h_{i-1}^2} y_{i-1} + \left[1 - 2 \frac{x - x_i}{h_{i-1}} \right] \frac{(x - x_{i-1})^2}{h_{i-1}^2} y_i + \\ (x - x_{i-1}) \frac{(x - x_i)^2}{h_{i-1}^2} m_{i-1} + (x - x_i) \frac{(x - x_{i-1})^2}{h_{i-1}^2} m_i \quad (2.44)$$

$$s''(x) = \left[\frac{6}{h_{i-1}^2} - \frac{12}{h_{i-1}^3} (x_i - x) \right] y_{i-1} + \left[\frac{6}{h_{i-1}^2} - \frac{12}{h_{i-1}^3} (x - x_{i-1}) \right] y_i + \\ \left[\frac{2}{h_{i-1}} - \frac{6}{h_{i-1}^2} (x_i - x) \right] m_{i-1} - \left[\frac{2}{h_{i-1}} - \frac{6}{h_{i-1}^2} (x - x_{i-1}) \right] m_i \quad (2.45)$$

由(2.43)式和(2.45)式, 当 $i=1, 2, \dots, n-1$ 时, 有

$$s''(x_i^+) = -\frac{6}{h_i^2} y_i + \frac{6}{h_i^2} y_{i+1} - \frac{4}{h_i} m_i - \frac{2}{h_i} m_{i+1} \quad (2.46)$$

$$s''(x_i^-) = \frac{6}{h_{i-1}^2} y_{i-1} - \frac{6}{h_{i-1}^2} y_i + \frac{2}{h_{i-1}} m_{i-1} + \frac{4}{h_{i-1}} m_i \quad (2.47)$$

因为 $s(x)$ 在 x_i 上具有连续的二阶微商, 所以 $s''(x_i^-) = s''(x_i^+)$, 即

$$\frac{6}{h_{i-1}^2} y_{i-1} - \frac{6}{h_{i-1}^2} y_i + \frac{2}{h_{i-1}} m_{i-1} + \frac{4}{h_{i-1}} m_i = -\frac{6}{h_i^2} y_i + \frac{6}{h_i^2} y_{i+1} - \frac{4}{h_i} m_i - \frac{2}{h_i} m_{i+1}$$

将上式整理后得

$$\frac{m_{i-1}}{h_{i-1}} + 2 \frac{h_{i-1} + h_i}{h_{i-1} h_i} m_i + \frac{m_{i+1}}{h_i} = 3 \left[\frac{y_i - y_{i-1}}{h_{i-1}^2} + \frac{y_{i+1} - y_i}{h_i^2} \right]$$

上式两边乘 $\frac{h_{i-1} h_i}{h_{i-1} + h_i}$ 得

$$\frac{h_i}{h_{i-1} + h_i} m_{i-1} + 2 m_i + \frac{h_{i-1}}{h_{i-1} + h_i} m_{i+1} = 3 \left[\frac{h_i}{(h_{i-1} + h_i) h_{i-1}} (y_i - y_{i-1}) + \right. \\ \left. \frac{h_{i-1}}{(h_{i-1} + h_i) h_i} (y_{i+1} - y_i) \right]$$

令 $\alpha_i = \frac{h_{i-1}}{h_{i-1} + h_i}$, 上式变为

$$(1 - \alpha_i)m_{i-1} + 2m_i + \alpha_i m_{i+1} = 3 \left[\frac{1 - \alpha_i}{h_{i-1}}(y_i - y_{i-1}) + \frac{\alpha_i}{h_i}(y_{i+1} - y_i) \right]$$

再令 $\beta_i = 3 \left[\frac{1 - \alpha_i}{h_{i-1}}(y_i - y_{i-1}) + \frac{\alpha_i}{h_i}(y_{i+1} - y_i) \right]$, 由上式得

$$(1 - \alpha_i)m_{i-1} + 2m_i + \alpha_i m_{i+1} = \beta_i \quad (i = 1, 2, \dots, n-1)$$

分别令 $i = 1, 2, \dots, n-1$, 得到由 $n-1$ 个等式组成的方程组:

$$\begin{cases} (1 - \alpha_1)m_0 + 2m_1 + \alpha_1 m_2 & = \beta_1 \\ (1 - \alpha_2)m_1 + 2m_2 + \alpha_2 m_3 & = \beta_2 \\ & \vdots \\ (1 - \alpha_{n-1})m_{n-2} + 2m_{n-1} + \alpha_{n-1}m_n & = \beta_{n-1} \end{cases} \quad (2.48)$$

这是一个含有 $n+1$ 个未知数和 $n-1$ 个方程的线性方程组, 因而方程组有无穷多个解.

为了得到惟一解, 需要根据具体情况补充两个附加条件, 从而消去两个未知数或补充两个方程. 通常选取边界条件作为附加条件, 而边界条件一般有如下两种情形:

(1) 已知被插值函数 $y = f(x)$ 在两个端点 x_0 和 x_n 处的微商值, 即给定 $f'(x_0) = m_0$ 和 $f'(x_n) = m_n$, 相应要求 $s(x_0) = m_0$ 和 $s(x_n) = m_n$, 再把 m_0 和 m_n 代入方程组 (2.48), 方程组 (2.48) 就变成只有 $n-1$ 个未知数了.

(2) 已知被插值函数 $y = f(x)$ 在两个端点 x_0 和 x_n 处的二阶微商值为 0, 即给定 $f''(x_0) = 0$ 和 $f''(x_n) = 0$, 相应要求 $s''(x_0) = 0$ 和 $s''(x_n) = 0$.

由 (2.46) 式得

$$s''(x_0^+) = -\frac{6}{h_0^2}y_0 + \frac{6}{h_0^2}y_1 - \frac{4}{h_0}m_0 - \frac{2}{h_0}m_1 = 0$$

整理后得

$$2m_0 + m_1 = \frac{3}{h_0}(y_1 - y_0) \quad (2.49)$$

由 (2.47) 式得

$$s''(x_n^-) = \frac{6}{h_{n-1}^2}y_{n-1} - \frac{6}{h_{n-1}^2}y_n + \frac{2}{h_{n-1}}m_{n-1} + \frac{4}{h_{n-1}}m_n = 0$$

整理后得

$$m_{n-1} + 2m_n = 3 \frac{y_n - y_{n-1}}{h_{n-1}} \quad (2.50)$$

把方程 (2.49)、方程 (2.50) 和方程组 (2.48) 联立起来, 就得到一个有 $n+1$ 个

未知数和 $n+1$ 个方程的线性代数方程组.

从前面的讨论可知, 补充了边界条件之后就可以求线性代数方程组的解 m_0, m_1, \dots, m_n 了. 将 m_0, m_1, \dots, m_n 代入 $s(x)$ 的各个子区间上的表达式 (2.41) 就可以得到三次样条插值函数 $s(x)$.

为了便于在计算机上实现三次样条插值, 把由上述两种边界条件推得的方程组写成统一形式:

$$\begin{cases} 2m_0 + \alpha_0 m_1 & = \beta_0 \\ (1 - \alpha_1)m_0 + 2m_1 + \alpha_1 m_2 & = \beta_1 \\ (1 - \alpha_2)m_1 + 2m_2 + \alpha_2 m_3 & = \beta_2 \\ & \vdots \\ (1 - \alpha_{n-1})m_{n-2} + 2m_{n-1} + \alpha_{n-1}m_n & = \beta_{n-1} \\ (1 - \alpha_n)m_{n-1} + 2m_n & = \beta_n \end{cases} \quad (2.51)$$

其中, 对第一种边界条件: $\alpha_0 = 0, \beta_0 = 2m_0, \alpha_n = 1, \beta_n = 2m_n$; 而对第二种边界条件:

$$\alpha_0 = 1, \beta_0 = \frac{3}{h_0}(y_1 - y_0), \alpha_n = 0, \beta_n = 3 \frac{y_n - y_{n-1}}{h_{n-1}}.$$

可以看到, 方程组 (2.51) 所对应的系数矩阵强对角占优. 而强对角占优的矩阵必定非奇异. 因此, 方程组 (2.51) 的解存在且惟一.

方程组 (2.51) 是一个三对角方程组, 可以推出求 m_0, m_1, \dots, m_n 的计算公式.

从方程组 (2.51) 的第一个方程得

$$m_0 = -\frac{\alpha_0}{2}m_1 + \frac{\beta_0}{2}$$

$$\text{令 } a_0 = -\frac{\alpha_0}{2}, b_0 = \frac{\beta_0}{2}, \text{ 则有}$$

$$m_0 = a_0 m_1 + b_0$$

把 m_0 代入方程组 (2.51) 的第二个方程, 整理后得

$$m_1 = -\frac{\alpha_1}{2 + (1 - \alpha_1)a_0}m_2 + \frac{\beta_1 - (1 - \alpha_1)b_0}{2 + (1 - \alpha_1)a_0}$$

$$\text{令 } a_1 = -\frac{\alpha_1}{2 + (1 - \alpha_1)a_0}, b_1 = \frac{\beta_1 - (1 - \alpha_1)b_0}{2 + (1 - \alpha_1)a_0}, \text{ 则有}$$

$$m_1 = a_1 m_2 + b_1$$

把 m_1 代入方程组 (2.51) 的第三个方程, 整理后得

$$m_2 = -\frac{\alpha_2}{2 + (1 - \alpha_2)a_1}m_3 + \frac{\beta_2 - (1 - \alpha_2)b_1}{2 + (1 - \alpha_2)a_1}$$

$$\text{令 } a_2 = -\frac{\alpha_2}{2 + (1 - \alpha_2)a_1}, b_2 = \frac{\beta_2 - (1 - \alpha_2)b_1}{2 + (1 - \alpha_2)a_1}, \text{ 则有}$$

$$m_2 = a_2 m_3 + b_2$$

.....

依此类推,最后把 m_{n-1} 代入第 $n+1$ 个方程并整理后得

$$m_{n-1} = a_{n-1}m_n + b_{n-1}$$

$$\text{其中, } a_{n-1} = -\frac{\alpha_{n-1}}{2 + (1 - \alpha_{n-1})a_{n-2}}; b_{n-1} = \frac{\beta_{n-1} - (1 - \alpha_{n-1})b_{n-2}}{2 + (1 - \alpha_{n-1})a_{n-2}}.$$

把上面式子归纳起来得到递推关系式

$$m_i = a_i m_{i+1} + b_i \quad (i=0, 1, \dots, n-1)$$

$$\text{其中, } a_0 = -\frac{\alpha_0}{2}; b_0 = \frac{\beta_0}{2}; a_i = -\frac{\alpha_i}{2 + (1 - \alpha_i)a_{i-1}}; b_i = \frac{\beta_i - (1 - \alpha_i)b_{i-1}}{2 + (1 - \alpha_i)a_{i-1}}.$$

从递推关系式可以得到

$$m_{n-1} = a_{n-1}m_n + b_{n-1}$$

$$m_{n-2} = a_{n-2}m_{n-1} + b_{n-2}$$

$$\vdots$$

$$m_0 = a_0 m_1 + b_0$$

很显然,只要求出 m_n ,就能求出 m_{n-1} ;求出 m_{n-1} ,就能求出 m_{n-2} ……依此类推,就能求得 m_{n-3}, \dots, m_1, m_0 .

为了求出 m_n ,把 $m_{n-1} = a_{n-1}m_n + b_{n-1}$ 代入方程组(2.51)最后一个方程得

$$(1 - \alpha_n)(a_{n-1}m_n + b_{n-1}) + 2m_n = \beta_n$$

从而得到

$$m_n = \frac{\beta_n - (1 - \alpha_n)b_{n-1}}{2 + (1 - \alpha_n)a_{n-1}} = b_n$$

令 $m_{n+1} = 0$,由 $m_n = a_n m_{n+1} + b_n$ 得

$$m_i = a_i m_{i+1} + b_i \quad (i=0, 1, \dots, n) \quad (2.52)$$

$$\text{其中, } a_0 = -\frac{\alpha_0}{2}; b_0 = \frac{\beta_0}{2}; a_i = -\frac{\alpha_i}{2 + (1 - \alpha_i)a_{i-1}}; b_i = \frac{\beta_i - (1 - \alpha_i)b_{i-1}}{2 + (1 - \alpha_i)a_{i-1}}.$$

因此,求方程组(2.51)的解可以归结为求递推公式(2.52)的值.这种方法通常称为追赶法.

三次样条插值算法

①读入插值节点 $x_i, y_i (i=0, 1, \dots, n)$ 和点 xx .

②对 $i=0, 1, \dots, n-1$ 计算 $h_i = x_{i+1} - x_i$.

③计算 α_i 和 $\beta_i (i=0, 1, \dots, n)$.

对第一种边界条件:

$$\alpha_0 = 0 \quad \beta_0 = 2m_0$$

$$\alpha_n = 1 \quad \beta_n = 2m_n$$

对第二种边界条件:

$$\alpha_0 = 1 \quad \beta_0 = \frac{3}{h_0}(y_1 - y_0)$$

$$\alpha_n = 0 \quad \beta_n = \frac{3}{h_{n-1}}(y_n - y_{n-1})$$

对 $i = 1, 2, \dots, n-1$, 做

$$\alpha_i = \frac{h_{i-1}}{h_{i-1} + h_i}$$

$$\beta_i = 3 \left[\frac{1 - \alpha_i}{h_{i-1}}(y_i - y_{i-1}) + \frac{\alpha_i}{h_i}(y_{i+1} - y_i) \right]$$

④计算 a_i 和 b_i ($i = 0, 1, \dots, n$).

$$a_0 = -\frac{\alpha_0}{2} \quad b_0 = \frac{\beta_0}{2}$$

对 $i = 1, 2, \dots, n$, 做

$$a_i = -\frac{\alpha_i}{2 + (1 - \alpha_i)a_{i-1}}$$

$$b_i = \frac{\beta_i - (1 - \alpha_i)b_{i-1}}{2 + (1 - \alpha_i)a_{i-1}}$$

⑤计算 m_i ($i = 0, 1, \dots, n+1$).

$$m_{n+1} = 0$$

对 $i = n, n-1, \dots, 0$, 做

$$m_i = a_i m_{i+1} + b_i$$

⑥判别点 xx 所在的区间 $[x_i, x_{i+1}]$ ($i = 0, 1, \dots, n-1$), 然后由 (2.41) 式求出 $s(x)$ 的值并输出.

例 2-7 给定函数表(见表 2-20)和边界条件 $s''(x_0) = 0, s''(x_n) = 0$, 求三次样条插值函数 $s(x)$, 并求 $f(3)$ 的近似值.

表 2-20 函数表

| | | | | |
|----------|---|---|---|---|
| x | 1 | 2 | 4 | 5 |
| $y=f(x)$ | 1 | 3 | 4 | 2 |

解 由 $h_i = x_{i+1} - x_i$ ($i = 0, 1, 2$) 得 $h_0 = 1, h_1 = 2, h_2 = 1$; 由 $\alpha_i = \frac{h_{i-1}}{h_{i-1} + h_i}$ ($i = 1, 2$) 得 $\alpha_1 = \frac{1}{3}, \alpha_2 = \frac{2}{3}$; 由 $\beta_i = 3 \left[\frac{1 - \alpha_i}{h_{i-1}}(y_i - y_{i-1}) + \frac{\alpha_i}{h_i}(y_{i+1} - y_i) \right]$ ($i = 1, 2$) 得 $\beta_1 = \frac{9}{2}, \beta_2 = -\frac{7}{2}$.

而 $\alpha_0 = 1, \beta_0 = \frac{3}{h_0}(y_1 - y_0) = 6, \alpha_3 = 0, \beta_3 = 3 \frac{y_3 - y_2}{h_2} = -6$, 由方程组 (2.51) 得到

$$\begin{cases} 2m_0 + \alpha_0 m_1 = \beta_0 \\ (1 - \alpha_1)m_0 + 2m_1 + \alpha_1 m_2 = \beta_1 \\ (1 - \alpha_2)m_1 + 2m_2 + \alpha_2 m_3 = \beta_2 \\ (1 - \alpha_3)m_2 + 2m_3 = \beta_3 \end{cases}$$

把 $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \beta_0, \beta_1, \beta_2$ 和 β_3 代入上面的方程组得

$$\begin{cases} 2m_0 + m_1 = 6 \\ \frac{2}{3}m_0 + 2m_1 + \frac{1}{3}m_2 = \frac{9}{2} \\ \frac{1}{3}m_1 + 2m_2 + \frac{2}{3}m_3 = -\frac{7}{2} \\ m_2 + 2m_3 = -6 \end{cases}$$

解这个方程组得 $m_0 = \frac{17}{8}, m_1 = \frac{7}{4}, m_2 = -\frac{5}{4}, m_3 = -\frac{19}{8}$.

于是, 由 (2.41) 式可以得到三次样条插值函数

$$s(x) = \begin{cases} -\frac{1}{8}x^3 + \frac{3}{8}x^2 + \frac{7}{4}x - 1 & (1 \leq x \leq 2) \\ -\frac{1}{8}x^3 + \frac{3}{8}x^2 + \frac{7}{4}x - 1 & (2 < x \leq 4) \\ \frac{3}{8}x^3 - \frac{45}{8}x^2 + \frac{103}{4}x - 33 & (4 < x \leq 5) \end{cases}$$

从而得到 $f(3) \approx s(3) = 4.25$.

对三次样条插值来说, 被插值函数 $f(x)$ 与三次样条插值函数 $s(x)$ 的误差估计的讨论是比较复杂的. 为方便误差估计的讨论, 给出如下定理, 证明从略.

定理 2.8 设给定如表 2-2 所示的函数表, $s(x)$ 是满足第一种边界条件的三次样条插值函数, $[a, b]$ 是包含 x_0, x_1, \dots, x_n 的任一区间, 并设 $f(x) \in C^4[a, b]$, 则对任意给定的 $x \in [a, b]$, 有误差估计式

$$\max_{a \leq x \leq b} |f^{(k)}(x) - s^{(k)}(x)| \leq c_k \max_{a \leq x \leq b} |f^{(4)}(x)| h^{4-k} \quad (k=0, 1, 2) \quad (2.53)$$

其中, $c_0 = \frac{5}{384}; c_1 = \frac{1}{24}; c_2 = \frac{1}{8}$ 和 $h = \max_{0 \leq i \leq n-1} |x_{i+1} - x_i|$.

2.6 数值微分

在微积分中, 函数的导数是由极限来定义的, 而且还推导出很多函数求导公

式. 若对由表达式表示的函数求导数, 使用函数求导公式容易把导数求出来. 但若函数用函数表格表示, 就不能用函数求导公式求函数导数, 这时只能使用近似方法求函数导数.

若函数 $f(x)$ 用函数表格表示, 则使用近似方法求 $f(x)$ 的导数称为数值微分. 常用的数值微分方法是使用插值函数.

2.7.1 使用 n 次插值函数求导数

在代数插值理论中, 对于给定函数 $f(x)$ 如表 2-2 所示的数据表, 可以构造一个 n 次插值函数 $p_n(x)$ 来近似代替 $f(x)$, 即有

$$f(x) \approx p_n(x)$$

对上式两边求导得

$$f'(x) \approx p'_n(x)$$

由定理 2.3 可知, $f(x)$ 与 $p_n(x)$ 的误差估计式为

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

其中, $\xi \in [a, b]$; $\omega(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$.

对上式两边求导得 $f'(x)$ 与 $p'_n(x)$ 的误差估计式为

$$R'(x) = f'(x) - p'_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega'(x) + \frac{\omega(x)}{(n+1)!} \frac{d}{dx} f^{(n+1)}(\xi) \quad (2.54)$$

在 (2.54) 式中, 由于 ξ 是 x 的未知函数, 所以无法对 $\frac{d}{dx} f^{(n+1)}(\xi)$ 作进一步的估计. 如果只考虑节点 x_k 处的导数, 并注意到 $\omega(x_k) = 0$, 则

$$R'(x_k) = f'(x_k) - p'_n(x_k) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega'(x_k) \quad (2.55)$$

由于上述原因, 下面只讨论节点 x_k 处的导数. 为了简单起见, 假定给出的节点是等距的, 并把这个距离记为 h .

2.7.1.1 两点公式

根据给定的两点 (x_0, y_0) 、 (x_1, y_1) 可以构造一个线性插值函数

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

近似代替 $f(x)$.

对上式两边求导数, 并注意到 $h = x_1 - x_0$, 则得

$$p'_1(x) = \frac{1}{h} (y_1 - y_0)$$

从而有下面求导公式

$$p'_1(x_0) = \frac{1}{h}(y_1 - y_0)$$

$$p'_1(x_1) = \frac{1}{h}(y_1 - y_0)$$

并有如下误差估计式

$$R'(x_0) = f'(x_0) - p'_1(x_0) = \frac{f''(\xi_0)}{2!}(x_0 - x_1) = -\frac{h}{2}f''(\xi_0) \quad (x_0 < \xi_0 < x_1)$$

$$R'(x_1) = f'(x_1) - p'_1(x_1) = \frac{f''(\xi_1)}{2!}(x_1 - x_0) = \frac{h}{2}f''(\xi_1) \quad (x_0 < \xi_1 < x_1)$$

2.7.1.2 三点公式

对于给定函数 $f(x)$ 如表 2-5 所示的数据表, 可以构造一个二次插值函数

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

近似代替 $f(x)$.

对上式两边求导得

$$p'_2(x) = \frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)}y_0 + \frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)}y_1 + \frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)}y_2$$

从而得到下面求导公式

$$p'_2(x_0) = \frac{1}{2h}(-3y_0 + 4y_1 - y_2)$$

$$p'_2(x_1) = \frac{1}{2h}(-y_0 + y_2)$$

$$p'_2(x_2) = \frac{1}{2h}(y_0 - 4y_1 + 3y_2)$$

并有如下误差估计式

$$R'(x_0) = f'(x_0) - p'_2(x_0) = \frac{f'''(\xi_0)}{3}h^2 \quad (x_0 < \xi_0 < x_2)$$

$$R'(x_1) = f'(x_1) - p'_2(x_1) = -\frac{f'''(\xi_1)}{6}h^2 \quad (x_0 < \xi_1 < x_2)$$

$$R'(x_2) = f'(x_2) - p'_2(x_2) = \frac{f'''(\xi_2)}{3}h^2 \quad (x_0 < \xi_2 < x_2)$$

从上面的数值微分公式及其误差估计式来看, 似乎 h 越小精度越高, 但在实际计算中并不是那么简单. 在实际计算中, 截断误差只是误差的一部分, 还有舍入误差, 而数值微分对舍入误差比较敏感, 有时候它会随着 h 的减小而增大. 因此, 在使用 n 次插值函数求导数时, 要注意对误差进行分析.

2.7.2 使用三次样条插值函数求导数

从前面的介绍可以知道,使用 n 次插值函数求导不便于计算非节点处的导数值,也不能保证有较小的误差. 这时最好使用三次样条插值函数求导数.

由三次样条插值知道,可以构造出函数 $s(x)$ 来近似代替 $f(x)$, 即有

$$f(x) \approx s(x) \quad (2.56)$$

而 $s(x)$ 在区间 $[x_i, x_{i+1}]$ ($i=0, 1, \dots, n-1$) 上的表达式为

$$s(x) = \left[1 + 2 \frac{x - x_i}{h_i} \right] \frac{(x - x_{i+1})^2}{h_i^2} y_i + \left[1 - 2 \frac{x - x_{i+1}}{h_i} \right] \frac{(x - x_i)^2}{h_i^2} y_{i+1} + \\ (x - x_i) \frac{(x - x_{i+1})^2}{h_i^2} m_i + (x - x_{i+1}) \frac{(x - x_i)^2}{h_i^2} m_{i+1}$$

对 $s(x)$ 求得

$$s'(x) = \frac{6}{h_i^2} \left[\frac{1}{h_i} (x - x_{i+1})^2 + (x - x_{i+1}) \right] y_i + \\ \frac{6}{h_i^2} \left[(x - x_i) - \frac{1}{h_i} (x - x_i)^2 \right] y_{i+1} + \\ \frac{1}{h_i} \left[\frac{3}{h_i} (x - x_{i+1})^2 + 2(x - x_{i+1}) \right] m_i - \\ \frac{1}{h_i} \left[2(x - x_i) - \frac{3}{h_i} (x - x_i)^2 \right] m_{i+1} \quad (2.57)$$

对 (2.56) 式两边求导就得到近似公式

$$f'(x) \approx s'(x)$$

若只求函数 $f(x)$ 在节点 x_i ($i=0, 1, \dots, n$) 处的导数, 则由 $f'(x_i) = m_i$ 可直接得到.

如果要求二阶导数 $f''(x)$, 则可对 (2.57) 式两边求导得到

$$s''(x) = \frac{6}{h_i^2} \left[1 + \frac{2}{h_i} (x - x_{i+1}) \right] y_i + \frac{6}{h_i^2} \left[1 - \frac{2}{h_i} (x - x_i) \right] y_{i+1} + \\ \frac{1}{h_i} \left[\frac{3}{h_i} (x - x_{i+1})^2 + 1 \right] m_i - \frac{2}{h_i} \left[1 - \frac{3}{h_i} (x - x_i) \right] m_{i+1}$$

此时, 又得到近似公式

$$f''(x) \approx s''(x)$$

三次样条插值函数 $s(x)$ 作为函数 $f(x)$ 的近似函数, 不仅函数值比较接近, 而且导数值也比较接近, 当 $h = \max_{0 \leq i \leq n-1} |x_{i+1} - x_i| \rightarrow 0$ 时, $s(x)$, $s'(x)$, $s''(x)$ 分别收敛于 $f(x)$, $f'(x)$, $f''(x)$. 但也有缺点, 要解方程组, 而当 h 较小时, 解方程组的运算量往往较大.

练习与思考

1. 给定函数 $y = \sin x$ 的函数表如表 2-21 所示, 请分别用 Lagrange 和 Newton 插值公式求出 $y = \sin x$ 在 $x = 1.75$ 处的函数近似值, 并进行误差估计.

表 2 - 21 函数表

| | | | |
|--------------|----------|----------|----------|
| x | 1. 74 | 1. 76 | 1. 78 |
| $y = \sin x$ | 0. 985 7 | 0. 982 2 | 0. 978 2 |

2. 由线性插值的误差估计式(2. 3),进一步证明:

$$|R(x)| \leqslant \frac{(x_1 - x_0)^2}{8} \max_{x_0 \leqslant x \leqslant x_1} |f''(x)|$$

3. 给定函数 $y=f(x)$ 的函数表如表 2 - 22 所示,分别用 Lagrange 插值和 Newton 插值构造三次插值多项式,并写出插值余项.

表 2 - 22 函数表

| | | | | |
|----------|---|---|---|---|
| x | 1 | 2 | 4 | 6 |
| $y=f(x)$ | 4 | 1 | 0 | 1 |

4. 假设 x_0, x_1, \cdots, x_n 是互不相等的插值节点, $l_i(x) (i = 0, 1, \cdots, n)$ 为 Lagrange 插值基函数,证明:

$$(1) \sum_{i=0}^n x_i^k l_i(x) = x^k \quad (k=0, 1, \cdots, n);$$

$$(2) \sum_{i=0}^n (x_i - x)^k l_i(x) = 0 \quad (k=1, 2, \cdots, n).$$

5. 证明 n 阶差商具有如下性质:

(1) 若 $F(x) = cf(x)$, 则

$$F(x_0, x_1, \cdots, x_n) = cf(x_0, x_1, \cdots, x_n);$$

(2) 若 $F(x) = f(x) + g(x)$, 则

$$F(x_0, x_1, \cdots, x_n) = f(x_0, x_1, \cdots, x_n) + g(x_0, x_1, \cdots, x_n)$$

6. 给定函数表如表 2 - 23 所示,求分段线性插值函数 $p(x)$, 并求 $f(2.5)$ 的近似值.

表 2 - 23 函数表

| | | | | |
|----------|---|---|---|---|
| x | 0 | 1 | 3 | 5 |
| $y=f(x)$ | 2 | 3 | 5 | 2 |

7. 给定函数 $y=f(x)$ 的函数表如表 2 - 24 所示,请构造一个三次 Hermite 插值函数,并求出 $f(1.5)$ 的近似值.

表 2-24 函数表

| | | |
|------------|---|----|
| x | 1 | 2 |
| $y=f(x)$ | 2 | 3 |
| $y'=f'(x)$ | 1 | -1 |

8. 求一个次数不高于 4 次的代数多项式 $p(x)$, 使它满足条件: $p(0) = p'(0) = 0$; $p(1) = p'(1) = 1$ 和 $p(2) = 1$.

9. 给定函数 $y=f(x)$ 的函数表(表 2-25)和边界条件 $s''(75) = 0, s''(80) = 0$, 求三次样条插值函数 $s(x)$, 并求 $f(78.3)$ 的近似值.

表 2-25 函数表

| | | | | | | |
|----------|-------|-------|-------|-------|-------|-------|
| x | 75 | 76 | 77 | 78 | 79 | 80 |
| $y=f(x)$ | 2.768 | 2.833 | 2.903 | 2.979 | 3.062 | 3.153 |

10. 给定函数 $y=f(x)$ 的函数表如表 2-26 所示, 边界条件 $s'(0.25) = 1, s'(0.53) = 0.6868$, 求三次样条插值函数 $s(x)$, 并求 $f(0.35)$ 的近似值.

表 2-26 函数表

| | | | | | |
|----------|------|--------|--------|--------|-------|
| x | 0.25 | 0.3 | 0.39 | 0.45 | 0.53 |
| $y=f(x)$ | 0.5 | 0.5477 | 0.6245 | 0.6708 | 0.728 |

11. 已知函数 $y=e^x$ 的数据表如表 2-27 所示, 试用两点和三点头数微分公式计算该函数在 $x=2.7$ 处的一阶和二阶导数.

表 2-27 函数表

| | | | | | |
|---------|-------|-------|-------|-------|-------|
| x | 2.5 | 2.6 | 2.7 | 2.8 | 2.9 |
| $y=e^x$ | 12.18 | 13.46 | 14.88 | 16.44 | 18.17 |

12. 已知函数 $y=f(x)$ 的函数表和边界条件与第 10 题相同, 试使用三次样条插值函数求 $y=f(x)$ 在 $x=0.35$ 处的一阶和二阶导数.

3 数据拟合

代数插值是根据给定的数据表(也称为函数表),按某些条件构造一个代数多项式 $p_n(x)$ 近似代替函数 $f(x)$,所要求的条件中有一个条件是 $p_n(x_i) = y_i (i = 0, 1, \dots, n)$,即要求函数 $p_n(x)$ 经过点 (x_i, y_i) .但是,由于数据表中给定的数据 x_i 和 y_i 是从实验或测量中得到的,难免有些误差,而且有个别点的误差可能比较大.用代数插值所得到的插值多项式 $p_n(x)$ 因为要经过点 (x_i, y_i) ,所以必然保留原来数据的一切误差.这是我们所不希望的.为了避免这种情况的发生,可以使用数据拟合法.

3.1 单变量数据拟合及最小二乘法

若给定的数据表表示的是一个量与另一个量的关系,则可以使用单变量数据拟合法寻找一个近似函数代替函数 $f(x)$.

单变量数据拟合法的一般过程是:先根据给定函数 $y = f(x)$ 的数据表(如表 3-1 所示),用几何描点法或凭经验选择一个近似函数 $F(x)$,以反映数据表中数据的一般趋势,然后使用最小二乘法确定 $F(x)$ 中的未知参数,从而得到 $f(x)$ 的近似函数 $F(x)$.

表 3-1 $y=f(x)$ 的数据表

| | | | | |
|----------|-------|-------|----------|-------|
| x | x_1 | x_2 | \cdots | x_n |
| $y=f(x)$ | y_1 | y_2 | \cdots | y_n |

通常 $F(x)$ 称为拟合函数, $f(x)$ 称为被拟合函数.

从单变量数据拟合法的一般过程可以看到,在使用数据拟合法求拟合函数 $F(x)$ 时,要用到最小二乘法.那么,什么是最小二乘法呢?

与插值法的目的-样,单变量数据拟合法也是要寻找一个近似函数 $F(x)$ 来近似代替 $f(x)$.但它与插值法又有些不同,它不要求近似函数 $F(x)$ 一定经过点 (x_i, y_i) .很自然,人们总是希望能找到一个最好的函数来近似代替 $f(x)$.现在的问题是,什么是“最好”的函数?“最好”的函数以什么标准来衡量?为了讨论这个问题,需要引入偏差的定义.

定义 3.1 若记 $\delta_i = f(x_i) - F(x_i), i = 1, 2, \dots, n$, 则称 δ_i 为 $f(x)$ 与 $F(x)$ 在 x_i 处的偏差.

一般情况下,使用单变量数据拟合法能找到一个近似函数 $F(x)$,使它与 $f(x)$ 的偏差 δ_i 的平方和最小,即使

$$\sum_{i=1}^n \delta_i^2 = \sum_{i=1}^n [f(x_i) - F(x_i)]^2$$

最小.能使偏差 δ_i 的平方和最小的函数就是“最好”的函数.因此,可以以偏差 δ_i 的平方和最小作为原则选择近似函数 $F(x)$.

定义 3.2 以“偏差的平方和最小”为原则选择近似函数的方法称为最小二乘法.

下面介绍一个呈线性关系的数据拟合的例子.

例 3-1 已知一组实验数据如表 3-2 所示,试用单变量数据拟合法求其拟合函数.

表 3-2 例 3-1 数据表

| | | | | | | | | |
|----------|----|---|---|---|---|---|---|----|
| x | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| $y=f(x)$ | 10 | 9 | 7 | 5 | 4 | 3 | 0 | -1 |

解 按照单变量数据拟合法的一般过程,根据给定的数据表(表 3-2),用几何描点法或凭经验选择近似函数,以反映数据表中数据的一般趋势.这里,用几何描点法选择近似函数,而用几何描点法选择近似函数要画出数据表中数据的散点图,散点图如图 3-1 所示.

从图 3-1 可以看到,点 (x_i, y_i) ($i=1, 2, \dots, 8$) 在一条直线附近,这些点基本满足直线方程.因此,可以选择线性函数来拟合这些数据,即可以选取

$$F(x) = a + bx$$

作为 $f(x)$ 的近似函数.其中 a 和 b 为待定参数.

拟合函数选定之后,还要确定拟合函数中的待定参数.确定拟合函数中的待定参数最常用的方法是最小二乘法.

先求出被拟合函数 $f(x)$ 与拟合函数 $F(x)$ 的偏差:

$$\delta_i = f(x_i) - F(x_i) = y_i - a - bx_i \quad (i=1, 2, \dots, 8)$$

按照最小二乘法,要使偏差的平方和最小,即需选择 a 和 b ,使

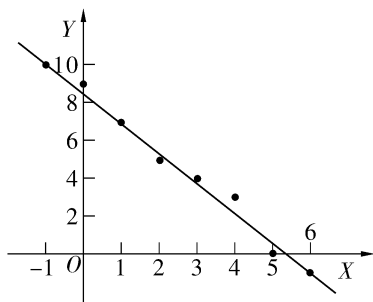


图 3-1 例 3-1 数据散点图

$$\sum_{i=1}^8 \delta_i^2 = \sum_{i=1}^8 (y_i - a - bx_i)^2 \quad (3.1)$$

最小.

显然, (3.1) 右边是关于未知参数 a 和 b 的函数, 所以可设

$$\sum_{i=1}^8 \delta_i^2 = \sum_{i=1}^8 (y_i - a - bx_i)^2 = \varphi(a, b)$$

这样一来, 选择 a 和 b 使偏差的平方和最小的问题就转化成选择 a 和 b 使二元函数 $\varphi(a, b)$ 最小的问题. 而选择 a 和 b 使二元函数 $\varphi(a, b)$ 最小的问题实际上就是求二元函数 $\varphi(a, b)$ 的极小值. 根据求多元函数极小值的方法, 先对 $\varphi(a, b)$ 分别求关于 a 和 b 的偏导数, 得

$$\frac{\partial \varphi(a, b)}{\partial a} = -2 \sum_{i=1}^8 (y_i - a - bx_i) \quad (3.2)$$

$$\frac{\partial \varphi(a, b)}{\partial b} = -2 \sum_{i=1}^8 (y_i - a - bx_i) x_i \quad (3.3)$$

令 (3.2) 式和 (3.3) 式等于 0, 整理后得

$$8a + \left[\sum_{i=1}^8 x_i \right] b = \sum_{i=1}^8 y_i \quad (3.4)$$

$$\left[\sum_{i=1}^8 x_i \right] a + \left[\sum_{i=1}^8 x_i^2 \right] b = \sum_{i=1}^8 x_i y_i \quad (3.5)$$

把 (3.4) 式和 (3.5) 式联立起来, 就得到含有两个未知参数 a 和 b 的有两个方程的线性代数方程组. 它通常称为正规方程组.

把 x_i 和 y_i 代入正规方程组得

$$\begin{cases} 8a + 20b = 37 \\ 20a + 92b = 25 \end{cases} \quad (3.6)$$

解方程组 (3.6) 得 $a = 8.6429, b = -1.6071$.

于是, 拟合函数

$$y = 8.6429 - 1.6071x$$

通过这个具体的例子, 可以把单变量数据拟合法的一般步骤归纳如下:

- ①按给定数据表画出散点图;
- ②分析散点图, 确定近似函数 $F(x)$ 的类型, 以反映给定数据的一般趋势;
- ③用最小二乘法确定近似函数 $F(x)$ 的未知参数, 从而得到最小二乘拟合函数

$F(x)$.

仿照例 3-1 的求解过程, 不难得到如下定理.

定理 3.1 给定 $y=f(x)$ 的数据表 (表 3-1), 若点 $(x_i, y_i) (i=1, 2, \dots, n)$ 大体上满足线性函数, 即最小二乘拟合函数为

$$F(x) = a + bx$$

则待定参数 a 和 b 是正规方程组

$$\begin{cases} na + \left[\sum_{i=1}^n x_i \right] b = \sum_{i=1}^n y_i \\ \left[\sum_{i=1}^n x_i \right] a + \left[\sum_{i=1}^n x_i^2 \right] b = \sum_{i=1}^n x_i y_i \end{cases}$$

(3.7)

的解。

单变量线性拟合法算法

①读入数据 x_i 和 $y_i (i = 1, 2, \cdots, n)$ 。

②计算

$$s_1 = \sum_{i=1}^n x_i \quad s_2 = \sum_{i=1}^n y_i \quad s_3 = \sum_{i=1}^n x_i^2 \quad s_4 = \sum_{i=1}^n x_i y_i$$

③解正规方程组

$$\begin{cases} na + s_1 b = s_2 \\ s_1 a + s_3 b = s_4 \end{cases}$$

即按下列公式求 a 和 b ：

$$a = \frac{s_2 s_3 - s_1 s_4}{n s_3 - s_1^2} \quad b = \frac{n s_4 - s_1 s_2}{n s_3 - s_1^2}$$

④输出 a 和 b 。

3.2 多变量数据拟合

在实际问题中,很多问题反映的不是一个量与一个量的关系,而是一个量与若干个量的关系. 具体来说,是一个量由若干个量确定. 这就是所谓的多元函数问题. 在数学上,这若干个量通常称为自变量,而由这些自变量确定的量通常称为因变量. 若假设这些自变量为 x_1, x_2, \cdots, x_k , 因变量为 y , 则每经过一次实验或测量就会得到一组数据 x_1, x_2, \cdots, x_k, y , 而经过 n 次实验或测量就会得到 n 组数据, 由这 n 组数据构成的数据表如表 3-3 所示.

表 3-3 多变量拟合数据表

| 实验或 测量次数 | x_1 | x_2 | \cdots | x_k | $y = f(x_1, x_2, \cdots, x_k)$ |
|-------------|----------|----------|----------|----------|--------------------------------|
| 1 | x_{11} | x_{12} | \cdots | x_{1k} | y_1 |
| 2 | x_{21} | x_{22} | \cdots | x_{2k} | y_2 |
| \vdots | \vdots | \vdots | | \vdots | \vdots |
| n | x_{n1} | x_{n2} | \cdots | x_{nk} | y_n |

根据表 3-3, 希望能找到一个函数来近似表达这些量的关系. 要做到这一点, 可以采用多变量数据拟合法.

多变量数据拟合法的一般过程是: 先根据表 3-3 选择变量 y 与变量 x_1, x_2, \dots, x_k 的一个近似函数 $F(x_1, x_2, \dots, x_k)$ 以反映 y 与变量 x_1, x_2, \dots, x_k 的函数关系, 然后使用最小二乘法确定近似函数 $F(x_1, x_2, \dots, x_k)$ 中的未知参数, 从而得到 $F(x_1, x_2, \dots, x_k)$.

通常 $F(x_1, x_2, \dots, x_k)$ 称为拟合函数, $y=f(x_1, x_2, \dots, x_k)$ 称为被拟合函数.

从多变量数据拟合法的一般过程可以看出, 若使用这种方法求拟合函数, 需根据表 3-3 选择一个近似函数 $F(x_1, x_2, \dots, x_k)$. 这是多变量数据拟合的关键. 这与两个变量的情况不同, 两个变量的情况可以使用几何描点法画出散点图辅助选择近似函数. 但对多变量的情况, 一般来说, 作图是困难的, 通常是凭经验或根据实际问题的物理背景和一些专业知识来找.

为了说明多变量数据拟合的一般过程, 现在讨论一种特殊情况. 假定表 3-3 中的数据呈线性关系, 这时选择线性函数

$$F(x_1, x_2, \dots, x_k) = a_0 + a_1 x_1 + a_2 x_2 + \dots + a_k x_k \quad (3.8)$$

来近似表达 y 与变量 x_1, x_2, \dots, x_k 的函数关系, 其中 a_0, a_1, \dots, a_k 为待定参数.

按数据拟合的方法, 要确定这些参数, 需使用最小二乘法. 现在使用最小二乘法来确定待定参数 a_0, a_1, \dots, a_k .

先把 $x_{m1}, x_{m2}, \dots, x_{mk} (m=1, 2, \dots, n)$ 代入 (3.8) 式得

$$F(x_{m1}, x_{m2}, \dots, x_{mk}) = a_0 + a_1 x_{m1} + a_2 x_{m2} + \dots + a_k x_{mk}$$

则 $F(x_1, x_2, \dots, x_k)$ 与 $y=f(x_1, x_2, \dots, x_k)$ 在 $x_{m1}, x_{m2}, \dots, x_{mk}$ 处的偏差为

$$\begin{aligned} \delta_m &= y_m - F(x_{m1}, x_{m2}, \dots, x_{mk}) \\ &= y_m - a_0 - a_1 x_{m1} - a_2 x_{m2} - \dots - a_k x_{mk} \end{aligned}$$

偏差的平方和为

$$\sum_{m=1}^n \delta_m^2 = \sum_{m=1}^n (y_m - a_0 - a_1 x_{m1} - a_2 x_{m2} - \dots - a_k x_{mk})^2 \xrightarrow{\text{记为}} \varphi(a_0, a_1, \dots, a_k)$$

根据最小二乘法, 要选择近似函数使 $\sum_{m=1}^n \delta_m^2$ 最小, 而 $\sum_{m=1}^n \delta_m^2$ 是关于 a_0, a_1, \dots, a_k 的函数, 所以可以转化为选择 a_0, a_1, \dots, a_k 使 $\varphi(a_0, a_1, \dots, a_k)$ 最小. 实际上就是求 $\varphi(a_0, a_1, \dots, a_k)$ 的极小值.

根据多元函数求极小值的方法, 对 $\varphi(a_0, a_1, \dots, a_k)$ 分别求关于 a_0, a_1, \dots, a_k 的偏导数并令其等于 0, 得

$$\begin{aligned} \frac{\partial \varphi}{\partial a_0} &= -2 \sum_{m=1}^n (y_m - a_0 - a_1 x_{m1} - a_2 x_{m2} - \dots - a_k x_{mk}) = 0 \\ \frac{\partial \varphi}{\partial a_1} &= -2 \sum_{m=1}^n (y_m - a_0 - a_1 x_{m1} - a_2 x_{m2} - \dots - a_k x_{mk}) x_{m1} = 0 \end{aligned}$$

$$\begin{aligned}\frac{\partial \varphi}{\partial a_2} &= -2 \sum_{m=1}^n (y_m - a_0 - a_1 x_{m1} - a_2 x_{m2} - \cdots - a_k x_{mk}) x_{m2} = 0 \\ &\vdots \\ \frac{\partial \varphi}{\partial a_k} &= -2 \sum_{m=1}^n (y_m - a_0 - a_1 x_{m1} - a_2 x_{m2} - \cdots - a_k x_{mk}) x_{mk} = 0\end{aligned}$$

整理化简后联立起来得方程组(3.9):

$$\begin{cases} na_0 & + a_1 \sum_{m=1}^n x_{m1} & + a_2 \sum_{m=1}^n x_{m2} & + \cdots + a_k \sum_{m=1}^n x_{mk} & = \sum_{m=1}^n y_m \\ a_0 \sum_{m=1}^n x_{m1} & + a_1 \sum_{m=1}^n x_{m1}^2 & + a_2 \sum_{m=1}^n x_{m2} x_{m1} & + \cdots + a_k \sum_{m=1}^n x_{mk} x_{m1} & = \sum_{m=1}^n y_m x_{m1} \\ a_0 \sum_{m=1}^n x_{m2} & + a_1 \sum_{m=1}^n x_{m1} x_{m2} & + a_2 \sum_{m=1}^n x_{m2}^2 & + \cdots + a_k \sum_{m=1}^n x_{mk} x_{m2} & = \sum_{m=1}^n y_m x_{m2} \\ a_0 \sum_{m=1}^n x_{mk} & + a_1 \sum_{m=1}^n x_{m1} x_{mk} & + a_2 \sum_{m=1}^n x_{m2} x_{mk} & + \cdots + a_k \sum_{m=1}^n x_{mk}^2 & = \sum_{m=1}^n y_m x_{mk} \end{cases}$$

方程组(3.9)是一个含有 $k+1$ 个方程和 $k+1$ 个未知数的线性代数方程组,也称为正规方程组.解这个方程组就能得到 a_0, a_1, \cdots, a_k .

综上所述,得到如下定理.

定理 3.2 给定 $y=f(x_1, x_2, \cdots, x_k)$ 的数据表(表3-3),若数据表中的数据呈线性关系,这时选取线性函数

$$F(x_1, x_2, \cdots, x_k) = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_k x_k$$

作为 y 与变量 x_1, x_2, \cdots, x_k 的最小二乘拟合函数,则待定参数 a_0, a_1, \cdots, a_k 是正规方程组(3.9)的解.

可以证明:当 $n > k$ 时,正规方程组(3.9)有惟一解.

例 3-2 已知一组测量数据如表 3-4 所示,求其线性拟合函数.

表 3-4 例 3-2 数据表

| 测量次数 | x_1 | x_2 | $y=f(x_1, x_2)$ |
|------|-------|-------|-----------------|
| 1 | 1 | 1 | 7 |
| 2 | 1 | 2 | 9 |
| 3 | 2 | 1 | 10 |
| 4 | 2 | 2 | 11 |
| 5 | 2 | 3 | 12 |

解 据题意,选择线性函数

$$F(x_1, x_2) = a_0 + a_1 x_1 + a_2 x_2 \quad (a_0, a_1, a_2 \text{ 为待定参数})$$

拟合给定数据表中的数据.

由定理 3.2 得到正规方程组

$$\begin{cases} na_0 + a_1 \sum_{m=1}^5 x_{m1} + a_2 \sum_{m=1}^5 x_{m2} = \sum_{m=1}^5 y_m \\ a_0 \sum_{m=1}^5 x_{m1} + a_1 \sum_{m=1}^5 x_{m1} x_{m1} + a_2 \sum_{m=1}^5 x_{m2} x_{m1} = \sum_{m=1}^5 y_m x_{m1} \\ a_0 \sum_{m=1}^5 x_{m2} + a_1 \sum_{m=1}^5 x_{m1} x_{m2} + a_2 \sum_{m=1}^5 x_{m2} x_{m2} = \sum_{m=1}^5 y_m x_{m2} \end{cases} \quad (3.10)$$

而 $\sum_{m=1}^5 x_{m1} = 8$, $\sum_{m=1}^5 x_{m2} = 9$, $\sum_{m=1}^5 x_{m1} x_{m1} = 14$, $\sum_{m=1}^5 x_{m1} x_{m2} = \sum_{m=1}^5 x_{m2} x_{m1} = 15$, $\sum_{m=1}^5 x_{m2} x_{m2} = 19$, $\sum_{m=1}^5 y_m = 49$, $\sum_{m=1}^5 y_m x_{m1} = 82$, $\sum_{m=1}^5 y_m x_{m2} = 93$, 把它们代入正规方程组 (3.10) 得

$$\begin{cases} 5a_0 + 8a_1 + 9a_2 = 49 \\ 8a_0 + 14a_1 + 15a_2 = 82 \\ 9a_0 + 15a_1 + 19a_2 = 93 \end{cases}$$

解方程组得 $a_0 = 3.8$, $a_1 = 2.4$, $a_2 = 1.2$.

于是, 所求的拟合函数为

$$y = F(x_1, x_2) = 3.8 + 2.4x_1 + 1.2x_2$$

多变量线性拟合法算法

①输入数据 x_{mi} 和 y_m ($m = 1, 2, \dots, n; i = 1, 2, \dots, k$).

②计算正规方程组的系数:

$$l_{00} = n \quad l_{0i} = l_{i0} = \sum_{m=1}^n x_{mi} \quad (i = 1, 2, \dots, k) \quad g_0 = \sum_{m=1}^n y_m$$

$$l_{ij} = \sum_{m=1}^n x_{mi} x_{mj} \quad (i, j = 1, 2, \dots, k) \quad g_i = \sum_{m=1}^n y_m x_{mi} \quad (i = 1, 2, \dots, k)$$

③解正规方程组

$$\begin{cases} l_{00}a_0 + l_{01}a_1 + \dots + l_{0k}a_k = g_0 \\ l_{10}a_0 + l_{11}a_1 + \dots + l_{1k}a_k = g_1 \\ \vdots \\ l_{k0}a_0 + l_{k1}a_1 + \dots + l_{kk}a_k = g_k \end{cases}$$

求出 a_0, a_1, \dots, a_k .

④输出 a_0, a_1, \dots, a_k .

3.3 非线性数据线性化

本章介绍的例 3-1 和例 3-2 都是线性拟合的例子. 在这两个例子中, 数据与数据之间的关系呈线性关系. 如果数据与数据之间的关系呈线性关系, 则可以按这两个例子所介绍的方法求拟合函数; 否则, 就不能直接使用这两个例子所介绍的方法求拟合函数. 但是, 在这类问题中有一些问题, 经过数据与数据之间的变换之后能得到线性关系, 这时就可以直接使用前面介绍过的方法.

例 3-3 某炼钢厂出钢时用的钢包(用来装钢水的容器)是用特殊耐火材料制成的, 在使用过程中, 由于钢水及炉渣对包衬耐火材料的侵蚀, 使其容量随着使用次数的增多而增大. 为了找出使用次数 x 与容量 y 之间的函数关系, 工程技术人员做了 15 次测试, 测试数据如表 3-5 所示. 试用数据拟合法找出使用次数 x 与容量 y 之间的函数关系.

表 3-5 例 3-3 数据表

| i | 使用次数 x_i | 容量 y_i | i | 使用次数 x_i | 容量 y_i | i | 使用次数 x_i | 容量 y_i |
|-----|---------------|-------------|-----|---------------|-------------|-----|---------------|-------------|
| 1 | 2 | 6.42 | 6 | 7 | 10.00 | 11 | 12 | 10.60 |
| 2 | 3 | 8.20 | 7 | 8 | 9.93 | 12 | 13 | 10.80 |
| 3 | 4 | 9.58 | 8 | 9 | 9.99 | 13 | 14 | 10.60 |
| 4 | 5 | 9.50 | 9 | 10 | 10.49 | 14 | 15 | 10.90 |
| 5 | 6 | 9.70 | 10 | 11 | 10.59 | 15 | 16 | 10.76 |

解 为了找出使用次数 x 和容量 y 之间的函数关系, 先画出散点图如图 3-2 所示.

按照图 3-2 中散点的趋势, 凭直观可以画出一条近似曲线, 使这些点或者落在这条曲线上, 或者落在曲线的两侧. 而这条近似曲线大致上像一条双曲线, 因而可以把使用次数 x 与容量 y 之间的关系近似表示为

$$\frac{1}{y} = a + b \frac{1}{x}$$

这是双曲线方程的一种形式, 其中 a 和 b 为待定参数.

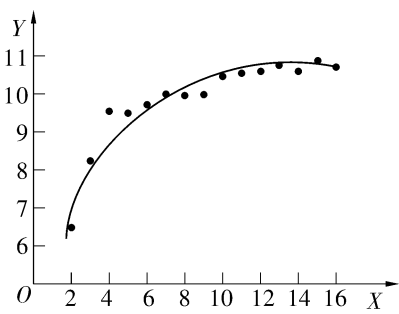


图 3-2 例 3-3 数据散点图

为了确定待定参数 a 和 b , 可以直接采用最小二乘法. 但这样得出的方程组为非线性方程组, 而非线性方程组一般是比较难求解的.

令 $X = \frac{1}{x}$ 和 $Y = \frac{1}{y}$, 得到 $Y = a + bX$. 这是一个线性函数, 现取它作为拟合函数.

由关系式 $X = \frac{1}{x}$ 和 $Y = \frac{1}{y}$ 得到 $X_i = \frac{1}{x_i} (i = 1, 2, \cdots, 15)$ 和 $Y_i = \frac{1}{y_i} (i = 1, 2, \cdots, 15)$, 从而得到一个新的数据表, 如表 3-6 所示.

表 3-6 例 3-3 新数据表

| i | x_i | y_i | $X_i = \frac{1}{x_i}$ | $Y_i = \frac{1}{y_i}$ | i | x_i | y_i | $X_i = \frac{1}{x_i}$ | $Y_i = \frac{1}{y_i}$ |
|-----|-------|-------|-----------------------|-----------------------|-----|-------|-------|-----------------------|-----------------------|
| 1 | 2 | 6.42 | 0.5000 | 0.1558 | 9 | 10 | 10.49 | 0.1000 | 0.0953 |
| 2 | 3 | 8.20 | 0.3333 | 0.1220 | 10 | 11 | 10.59 | 0.0909 | 0.0944 |
| 3 | 4 | 9.58 | 0.2500 | 0.1044 | 11 | 12 | 10.60 | 0.0833 | 0.0943 |
| 4 | 5 | 9.50 | 0.2000 | 0.1053 | 12 | 13 | 10.80 | 0.0769 | 0.0926 |
| 5 | 6 | 9.70 | 0.1667 | 0.1031 | 13 | 14 | 10.60 | 0.0714 | 0.0943 |
| 6 | 7 | 10.00 | 0.1429 | 0.1000 | 14 | 15 | 10.90 | 0.0667 | 0.0917 |
| 7 | 8 | 9.93 | 0.1250 | 0.1007 | 15 | 16 | 10.76 | 0.0625 | 0.0929 |
| 8 | 9 | 9.99 | 0.1111 | 0.1001 | | | | | |

有了新数据表和选定的拟合函数, 就可以按最小二乘法确定待定参数 a 和 b . 由定理 3.1 得到正规方程组

$$\begin{cases} 15a + \left[\sum_{i=1}^{15} X_i \right] b = \sum_{i=1}^{15} Y_i \\ \left[\sum_{i=1}^{15} X_i \right] a + \left[\sum_{i=1}^{15} X_i^2 \right] b = \sum_{i=1}^{15} X_i Y_i \end{cases} \tag{3.11}$$

而 $\sum_{i=1}^{15} X_i \approx 2.3807$, $\sum_{i=1}^{15} Y_i \approx 1.5469$, $\sum_{i=1}^{15} X_i^2 \approx 0.5842$, $\sum_{i=1}^{15} X_i Y_i \approx 0.2727$. 把它们代入正规方程组(3.11)得

$$\begin{cases} 15a + 2.3807b = 1.5469 \\ 2.3807a + 0.5842b = 0.2727 \end{cases}$$

解方程组得 $a \approx 0.0824$, $b \approx 0.1318$. 故

$$Y = 0.0824 + 0.1318X$$

即有

$$\frac{1}{y} = 0.0824 + 0.1318 \frac{1}{x}$$

于是,拟合函数

$$y = \frac{x}{0.0824x + 0.1318}$$

例 3-4 已知一组数据如表 3-7 所示,求一个经验函数,形如 $y = ae^{bx}$ (a, b 为常数),使它与表 3-7 中的数据相拟合.

表 3-7 例 3-4 数据表

| x | 0 | 1 | 2 | 3 | 4 |
|----------|-----|-----|-----|---|-----|
| $y=f(x)$ | 1.5 | 2.5 | 3.5 | 5 | 7.5 |

解 据题意,表 3-7 中的数据大体上满足函数关系 $y = ae^{bx}$. 所以,表 3-7 的拟合函数为 $y = ae^{bx}$. 为了便于确定 a 和 b ,可以通过变量变换把 $y = ae^{bx}$ 转换为线性函数,然后用最小二乘法来确定 a 和 b .

先对 $y = ae^{bx}$ 两边取自然对数得

$$\ln y = \ln a + bx$$

令 $Y = \ln y, A = \ln a, B = b$ 和 $X = x$,则有

$$Y = A + BX$$

这是一个线性函数,取它作为拟合函数.

由 $Y = \ln y$ 和 $X = x$ 得到 $Y_i = \ln y_i$ 和 $X_i = x_i$,从而得到新的数据表,如表 3-8 所示.

表 3-8 例 3-4 新数据表

| i | x_i | y_i | X_i | $Y_i = \ln y_i$ |
|-----|-------|-------|-------|-----------------|
| 1 | 0 | 1.5 | 0 | 0.405465 |
| 2 | 1 | 2.5 | 1 | 0.916291 |
| 3 | 2 | 3.5 | 2 | 1.252763 |
| 4 | 3 | 5.0 | 3 | 1.609438 |
| 5 | 4 | 7.5 | 4 | 2.014903 |

有了拟合函数,又有了新的数据表,就可以使用最小二乘法确定 a 和 b . 由定理 3.1 得到正规方程组

$$\begin{cases} 5A + \left[\sum_{i=1}^5 X_i \right] B = \sum_{i=1}^5 Y_i \\ \left[\sum_{i=1}^5 X_i \right] A + \left[\sum_{i=1}^5 X_i^2 \right] B = \sum_{i=1}^5 X_i Y_i \end{cases} \quad (3.12)$$

而 $\sum_{i=1}^5 X_i = 10$, $\sum_{i=1}^5 Y_i \approx 6.19886$, $\sum_{i=1}^5 X_i^2 = 30$, $\sum_{i=1}^5 X_i Y_i \approx 16.309743$, 把它们代入

正规方程组 (3.12) 得

$$\begin{cases} 5A + 10B = 6.19886 \\ 10A + 30B = 16.309743 \end{cases}$$

解上述方程组得 $A \approx 0.457367, B \approx 0.3912023$. 又由 $A = \ln a$ 得 $a \approx 1.579910, b = B \approx 0.3912023$.

于是, 所求的拟合函数为

$$y = 1.579910e^{0.3912023x}$$

在实践中, 有时要进行多项式拟合. 多项式拟合也可以采用非线性数据线性化的方法.

设有两个量 z 和 y 基本满足 m 次多项式函数, 经过实验或测量得到数据表 (表 3-9), 从而可以把

$$y = a_0 + a_1z + a_2z^2 + \cdots + a_mz^m \tag{3.13}$$

作为给定数据的拟合函数, 其中 a_0, a_1, \cdots, a_m 为待定参数.

表 3-9 多项式拟合数据表

| | | | | |
|----------|-------|-------|----------|-------|
| z | z_1 | z_2 | \cdots | z_n |
| $y=f(z)$ | y_1 | y_2 | \cdots | y_n |

令 $x_1 = z, x_2 = z^2, \cdots, x_m = z^m$, 则由 (3.13) 式有

$$y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_mx_m \overset{\text{记为}}{=} F(x_1, x_2, \cdots, x_m)$$

使它变成多变量线性拟合.

在数据拟合法中, 当选定拟合函数后, 非线性拟合函数能否线性化是首先要考虑的问题. 为了方便读者对非线性拟合函数进行线性化, 表 3-10 中给出了常用非线性拟合函数线性化的方法.

表 3-10 常用非线性拟合函数线性化的方法

| 非线性拟合函数的形式 | 对非线性拟合函数线性化 为 $Y = A + BX$ | 需做的变换 |
|-----------------------|-------------------------------|--|
| $y = a + b\ln x$ | $y = a + b\ln x$ | $X = \ln x, Y = y$ $A = a, B = b$ |
| $y = ae^{bx}$ | $\ln y = \ln a + bx$ | $X = x, Y = \ln y$ $A = \ln a, B = b$ |
| $y = a + \frac{b}{x}$ | $y = a + b \frac{1}{x}$ | $X = \frac{1}{x}, Y = y$ $A = a, B = b$ |

续上表

| 非线性拟合函数的形式 | 对非线性拟合函数线性化 为 $Y = A + BX$ | 需做的变换 |
|-----------------------------|---|---|
| $y = \frac{x}{a + bx}$ | $\frac{1}{y} = b + a \frac{1}{x}$ | $X = \frac{1}{x}, Y = \frac{1}{y}$ $A = b, B = a$ |
| $y = (a + bx)^{-2}$ | $y^{-\frac{1}{2}} = a + bx$ | $X = x, Y = y^{-\frac{1}{2}}$ $A = a, B = b$ |
| $y = axe^{-bx}$ | $\ln \frac{y}{x} = \ln a - bx$ | $X = x, Y = \ln \frac{y}{x}$ $A = \ln a, B = -b$ |
| $y = \frac{1}{1 + ae^{bx}}$ | $\ln \left[\frac{1}{y} - 1 \right] = \ln a + bx$ | $X = x, Y = \ln \left[\frac{1}{y} - 1 \right]$ $A = \ln a, B = b$ |

3.4 正交多项式拟合

多项式拟合的做法是:先把多项式拟合函数变成多变量拟合函数,然后按照多变量拟合法求出多项式的系数 a_0, a_1, \cdots, a_m . 在多变量拟合法中,要解一个正规方程组. 理论上可以证明该正规方程组有惟一解,但当多项式的次数比较高时,正规方程组就会变成病态方程组. 所谓病态方程组,就是如果方程组 $\mathbf{Ax} = \mathbf{b}$ 中的系数矩阵 \mathbf{A} 和常数项 \mathbf{b} 有微小变化,就会引起方程组的解很大变化,则方程组 $\mathbf{Ax} = \mathbf{b}$ 就是一个病态方程组. 为了克服这个缺点,把多项式拟合函数取为

$$y^* = \Psi_m(x) = a_0p_0(x) + a_1p_1(x) + \cdots + a_mp_m(x) \tag{3.14}$$
$$= \sum_{k=0}^m a_kp_k(x)$$

其中 a_0, a_1, \cdots, a_m 为待定参数, $p_k(x) (k=0, 1, \cdots, m)$ 是 k 次多项式.

很显然, $y^* = \Psi_m(x)$ 是一个 m 次多项式函数.

把 x_i 代入(3.14)式得到 y_i 的近似值 y_i^* , 则 y_i^* 与 y_i 就可能产生偏差. 记 y_i^* 与 y_i 的偏差为 δ_i , 有

$$\delta_i = y_i - y_i^* = y_i - \sum_{k=0}^m a_kp_k(x_i) \quad (i = 1, 2, \cdots, n)$$

为了求得 a_0, a_1, \cdots, a_m , 按最小二乘法选择 a_0, a_1, \cdots, a_m 使误差 δ_i 的平方和最小, 即选择 a_0, a_1, \cdots, a_m , 使 $\sum_{i=1}^n \delta_i^2$ 最小. 由于从实验或测量中得到的数据的精度

不同,通常在每一个 δ_i 的前面乘上一个表示数据精度的权数 α_i . 即使使

$$\sum_{i=1}^n (\alpha_i \delta_i)^2 = \sum_{i=1}^n \alpha_i^2 \delta_i^2 = \sum_{i=1}^n \omega_i \delta_i^2$$

最小. 其中 $\omega_i = \alpha_i^2$ 称为权因子. 而

$$\sum_{i=1}^n \omega_i \delta_i^2 = \sum_{i=1}^n \omega_i \left[y_i - \sum_{k=0}^m a_k p_k(x_i) \right]^2 \xrightarrow{\text{记为}} \varphi(a_0, a_1, \dots, a_m)$$

因此,可以把选择 a_0, a_1, \dots, a_m 使 $\sum_{i=1}^n \omega_i \delta_i^2$ 最小的问题转化为求函数 $\varphi(a_0, a_1, \dots, a_m)$ 极小值的问题. 按多元函数求极小值的方法,对函数 $\varphi(a_0, a_1, \dots, a_m)$ 分别求关于 a_0, a_1, \dots, a_m 的导数,令其为 0 并联立起来得到方程组

$$\frac{\partial \varphi}{\partial a_j} = -2 \sum_{i=1}^n \omega_i \left[y_i - \sum_{k=0}^m a_k p_k(x_i) \right] p_j(x_i) = 0 \quad (j=0, 1, \dots, m) \quad (3.15)$$

将方程组(3.15)整理后得

$$\sum_{i=1}^n \omega_i \left[\sum_{k=0}^m a_k p_k(x_i) p_j(x_i) \right] - \sum_{i=1}^n \omega_i y_i p_j(x_i) = 0$$

即

$$\sum_{k=0}^m a_k \left[\sum_{i=1}^n \omega_i p_k(x_i) p_j(x_i) \right] = \sum_{i=1}^n \omega_i y_i p_j(x_i) \quad (3.16)$$

令 $c_{jk} = \sum_{i=1}^n \omega_i p_k(x_i) p_j(x_i)$, $b_j = \sum_{i=1}^n \omega_i y_i p_j(x_i)$, 则方程组(3.16)可以写成

$$\sum_{k=0}^m c_{jk} a_k = b_j \quad (3.17)$$

方程组(3.17)也称为正规方程组.

显然,如果能找到 $p_k(x)$ 满足下列关系式:

$$c_{jk} = \sum_{i=1}^n \omega_i p_j(x_i) p_k(x_i) = 0 \quad (j \neq k) \quad (3.18)$$

$$c_{jj} = \sum_{i=1}^n \omega_i p_j^2(x_i) > 0 \quad (j, k=0, 1, \dots, m) \quad (3.19)$$

则正规方程组(3.17)变为

$$c_{kk} a_k = b_k \quad (3.20)$$

由(3.20)式可以求出

$$a_k = \frac{b_k}{c_{kk}} \quad (3.21)$$

定义 3.3 满足(3.18)式和(3.19)式的多项式 $p_k(x)$ ($k=0, 1, \dots, m$) 通常称为正交多项式簇,确切地说,称为对数据 x_i 和对应的权数 ω_i 的正交多项式簇.

常用的正交多项式簇有一些,在这里只介绍一个常用的等距节点正交多项式簇.

假设给定一组 $n+1$ 个等距节点 $\xi_i (i=0,1,\cdots,n)$, 它们的间隔为 h , 选取权数 $\omega_i = 1$, 引入变换

$$x = \frac{\xi - \xi_0}{h}$$

则 ξ_i 变为 $x_i = i$ 等 $n+1$ 个整数等距节点.

构造多项式

$$p_{m,n}(x) = \sum_{k=0}^m (-1)^k \begin{bmatrix} m \\ k \end{bmatrix} \begin{bmatrix} m+k \\ k \end{bmatrix} \frac{x^{(k)}}{n^{(k)}}$$

其中 $x^{(k)} = x(x-1)\cdots(x-k+1)$ 且 $x^{(0)} = 1$. 可以证明, $p_{m,n}(x)$ 是在 $x_i = i$ 等 $n+1$ 个整数等距节点上权数 $\omega_i = 1$ 的正交多项式簇, 其前 6 个多项式为

$$p_{0,n}(x) = 1$$

$$p_{1,n}(x) = 1 - 2 \frac{x}{n}$$

$$p_{2,n} = 1 - 6 \frac{x}{n} + 6 \frac{x(x-1)}{n(n-1)}$$

$$p_{3,n} = 1 - 12 \frac{x}{n} + 30 \frac{x(x-1)}{n(n-1)} - 20 \frac{x(x-1)(x-2)}{n(n-1)(n-2)}$$

$$p_{4,n} = 1 - 20 \frac{x}{n} + 90 \frac{x(x-1)}{n(n-1)} - 140 \frac{x(x-1)(x-2)}{n(n-1)(n-2)} + 70 \frac{x(x-1)(x-2)(x-3)}{n(n-1)(n-2)(n-3)}$$

$$p_{5,n} = 1 - 30 \frac{x}{n} + 210 \frac{x(x-1)}{n(n-1)} - 560 \frac{x(x-1)(x-2)}{n(n-1)(n-2)} + 630 \frac{x(x-1)(x-2)(x-3)}{n(n-1)(n-2)(n-3)} + 252 \frac{x(x-1)(x-2)(x-3)(x-4)}{n(n-1)(n-2)(n-3)(n-4)}$$

练习与思考

1. 求方程组

$$\begin{cases} x_1 + 2x_2 = 1 \\ x_1 + x_2 = 0 \\ x_1 - x_2 = 1 \end{cases}$$

的最小二乘解.

2. 给定数据表如表 3-11 所示, 求形如 $y = a + bx^2$ 的拟合函数.

表 3 - 11 数据表

| | | | | | |
|-----|----|------|----|------|------|
| x | 19 | 25 | 31 | 38 | 44 |
| y | 19 | 32.3 | 49 | 73.3 | 97.8 |

3. 用最小二乘法求一个形如 $y = \frac{1}{a + bx}$ 的经验公式,使之与如下数据(表 3 - 12)相拟合.

表 3 - 12 数据表

| | | | | | |
|-----|-------|-------|-------|-------|-------|
| x | 1 | 1.4 | 1.8 | 2.2 | 2.6 |
| y | 0.931 | 0.473 | 0.297 | 0.224 | 0.168 |

4. 在某个低温过程中,函数 y 依赖于温度 $\theta(^{\circ}\text{C})$ 的测试数据如表 3 - 13 所示,而且已知经验公式是 $y = a\theta + b\theta^2$,试用最小二乘法确定 a 和 b .

表 3 - 13 数据表

| | | | | |
|----------|-----|-----|-----|---|
| θ | 1 | 2 | 3 | 4 |
| y | 0.8 | 1.5 | 1.8 | 2 |

5. 给定数据表如表 3 - 14 所示,试用三次多项式拟合表中的数据.

表 3 - 14 数据表

| | | | | | |
|-----|------|-----|-----|-----|-----|
| x | -2 | -1 | 0 | 1 | 2 |
| y | -0.1 | 0.1 | 0.4 | 0.9 | 1.6 |

6. 在某一化学反应里,据实验得到分解生成物的浓度与时间的数据如表 3 - 15 所示,试使用正交多项式拟合法构造一个 5 次多项式求 y 的近似值.

表 3 - 15 数据表

| | | | | | | | | | | | |
|------------------------|---|------|------|------|------|------|------|------|------|------|------|
| 时间 $t(\text{min})$ | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 浓度 $y(\times 10^{-4})$ | 0 | 1.27 | 2.16 | 2.86 | 3.44 | 3.87 | 4.15 | 4.37 | 4.51 | 4.60 | 4.66 |

4 数值积分

在微积分中,计算连续函数 $f(x)$ 在区间 $[a, b]$ 上的积分通常采用求原函数的方法. 也就是说,利用积分的基本公式

$$\int_a^b f(x) dx = F(b) - F(a) \quad (4.1)$$

计算定积分的值. 这里的 $F(x)$ 为被积函数 $f(x)$ 的原函数. 但有时 $f(x)$ 的原函数无法用初等函数来表示. 例如,积分 $\int_0^1 \frac{\sin x}{x} dx$ 和 $\int_0^1 e^{-x^2} dx$ 中的被积函数 $\frac{\sin x}{x}$ 和 e^{-x^2} 的原函数就不能用初等函数表示出来.

在实际中,有些函数是用函数表来表示的,对这种被积函数的积分也不能通过求原函数的方法求得.

另外,计算机只能进行四则运算和逻辑运算,不能进行求原函数的运算,所以在计算机上也无法实现这种方法.

这说明用求原函数的方法求定积分的值有很大的局限性,有必要学习求定积分的近似方法. 下面介绍一些求定积分的近似方法.

4.1 梯形求积公式、Simpson 求积公式和 Newton - Cotes 求积公式

根据代数插值法,对于任一被积函数 $f(x)$,都可以构造一个插值多项式 $p(x)$ 近似代替它,即有

$$f(x) \approx p(x) \quad (4.2)$$

对(4.2)式两边求积分得

$$\int_a^b f(x) dx \approx \int_a^b p(x) dx$$

而 $p(x)$ 是一个代数多项式,它的定积分容易计算.

可以根据这种想法构造出几个近似求积公式.

4.1.1 梯形求积公式

由代数插值法知道,可以以 a 和 b 作为插值节点构造一个插值多项式

$$p_1(x) = \frac{x-b}{a-b}f(a) + \frac{x-a}{b-a}f(b)$$

近似代替 $f(x)$, 即有

$$f(x) \approx p_1(x) \quad (4.3)$$

对 (4.3) 式两边求积分得

$$\begin{aligned} \int_a^b f(x) dx &\approx \int_a^b p_1(x) dx \\ &= \int_a^b \left[\frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b) \right] dx \\ &= \frac{b-a}{2} [f(a) + f(b)] \end{aligned} \quad (4.4)$$

(4.4) 式是一个计算定积分的近似公式, 又称为梯形求积公式. 它的几何意义如图 4-1 所示.

从图 4-1 可以看到, 梯形求积公式的几何意义是用梯形区域的面积来代替曲边梯形区域的面积.

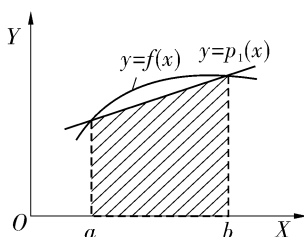


图 4-1 梯形求积公式的几何意义

4.1.2 Simpson 求积公式

把积分区间 $[a, b]$ 二等分, 得到三个分

点 $a, \frac{a+b}{2}$ 和 b . 根据代数插值法, 可以以这三个分点作为插值节点, 构造一个插值多项式

$$\begin{aligned} p_2(x) = & \left[\frac{x - \frac{a+b}{2}}{a - \frac{a+b}{2}} \right] \frac{(x-b)}{(a-b)} f(a) + \left[\frac{(x-a)(x-b)}{\left[\frac{a+b}{2} - a \right] \left[\frac{a+b}{2} - b \right]} \right] f\left[\frac{a+b}{2} \right] + \\ & \frac{(x-a) \left[x - \frac{a+b}{2} \right]}{(b-a) \left[b - \frac{a+b}{2} \right]} f(b) \end{aligned}$$

近似代替 $f(x)$, 即有

$$f(x) \approx p_2(x) \quad (4.5)$$

对 (4.5) 式两边求积分得

$$\begin{aligned} \int_a^b f(x) dx &\approx \int_a^b p_2(x) dx \\ &= \int_a^b \left[\left[\frac{x - \frac{a+b}{2}}{a - \frac{a+b}{2}} \right] \frac{(x-b)}{(a-b)} f(a) + \left[\frac{(x-a)(x-b)}{\left[\frac{a+b}{2} - a \right] \left[\frac{a+b}{2} - b \right]} \right] f\left[\frac{a+b}{2} \right] \right. \end{aligned}$$

$$\begin{aligned}
 & + \frac{(x-a)\left[x - \frac{a+b}{2}\right]}{(b-a)\left[b - \frac{a+b}{2}\right]} f(b) \Big] dx \\
 & = \frac{b-a}{6} \left[f(a) + 4f\left[\frac{a+b}{2}\right] + f(b) \right] \quad (4.6)
 \end{aligned}$$

(4.6)式也是一个计算定积分的近似公式,又称为 Simpson 或抛物线求积公式. 它的几何意义如图 4-2 所示.

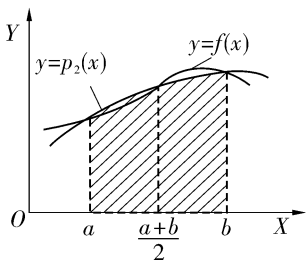


图 4-2 Simpson 求积公式的几何意义

4.1.3 Newton - Cotes 求积公式

把积分区间 $[a, b]$ n 等分, 得到 $n+1$ 个分点, 其分点记为 $x_i = a + ih (i=0, 1, \dots, n)$,

其中 $h = \frac{b-a}{n}$. 由代数插值法知道, 可以以这 $n+1$ 个分点作为插值节点, 构造一个插值多项式

$$p_n(x) = \sum_{i=0}^n \frac{\omega(x)}{(x-x_i)\omega'(x_i)} f(x_i)$$

近似代替 $f(x)$, 其中 $\omega(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$. 即有

$$f(x) \approx p_n(x) \quad (4.7)$$

对 (4.7) 式两边求积分得

$$\begin{aligned}
 \int_a^b f(x) dx & \approx \int_a^b p_n(x) dx = \int_a^b \frac{\omega(x)}{(x-x_i)\omega'(x_i)} f(x_i) dx \\
 & = \sum_{i=0}^n \left[\int_a^b \frac{\omega(x)}{(x-x_i)\omega'(x_i)} dx f(x_i) \right] \\
 & = \sum_{i=0}^n A_i f(x_i) \quad (4.8)
 \end{aligned}$$

其中 $A_i = \int_a^b \frac{\omega(x)}{(x-x_i)\omega'(x_i)} dx$.

现在的问题是如何求 A_i . 为了求 A_i , 作变换 $x = a + th$, 从而得

$$\begin{aligned}
 \omega(x) & = \omega(a+th) = (a+th-x_0)(a+th-x_1)\cdots(a+th-x_n) \\
 & = h^{n+1} t(t-1)\cdots(t-n)
 \end{aligned}$$

$$x - x_i = a + th - (a + ih) = (t - i)h$$

$$\begin{aligned}
 \omega'(x_i) & = (x_i - x_0)(x_i - x_1)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n) \\
 & = ih(i-1)h\cdots h(-h)(-2h)\cdots[-(n-i)]h \\
 & = (-1)^{n-i} h^n i! (n-i)!
 \end{aligned}$$

$$A_i = \int_a^b \frac{\omega(x)}{(x-x_i)\omega'(x_i)} dx$$

$$\stackrel{x=a+th}{=} \frac{(-1)^{n-i}(b-a)}{n \cdot i! (n-i)} \int_0^n \frac{t(t-1)\cdots(t-n)}{t-i} dt$$

$$\text{若记 } c_i^{(n)} = \frac{(-1)^{n-i}}{n \cdot i! (n-i)} \int_0^n \frac{t(t-1)\cdots(t-n)}{t-i} dt$$

则有 $A_i = (b-a)c_i^{(n)}$, 从而

$$\int_a^b f(x) dx \approx \int_a^b p_n(x) dx = \sum_{i=0}^n (b-a)c_i^{(n)} f(x_i) \quad (4.9)$$

(4.9) 式称为 Newton-Cotes (牛顿-柯特斯) 公式, $c_i^{(n)}$ 称为 Newton-Cotes 系数.

可以看出, $c_i^{(n)}$ 仅依赖于 n 和 i , 也就是说仅依赖于 n 和 x_i , 而不依赖于被积函数 $f(x)$ 和积分区间 $[a, b]$. 而 n 和 i 都是整数, 从而 $c_i^{(n)}$ 是一个常数, 这个常数可以预先计算出来构成 Newton-Cotes 系数表. 在计算时, 可以查 Newton-Cotes 系数表得到 Newton-Cotes 系数 $c_i^{(n)}$, 而不用求积分.

n 从 1 到 6 取值的 Newton-Cotes 系数见表 4-1.

表 4-1 n 从 1 到 6 的 Newton-Cotes 系数表

| n | $c_0^{(n)}$ | $c_1^{(n)}$ | $c_2^{(n)}$ | $c_3^{(n)}$ | $c_4^{(n)}$ | $c_5^{(n)}$ | $c_6^{(n)}$ |
|-----|------------------|-----------------|------------------|------------------|-----------------|------------------|------------------|
| 1 | $\frac{1}{2}$ | $\frac{1}{2}$ | | | | | |
| 2 | $\frac{1}{6}$ | $\frac{4}{6}$ | $\frac{1}{6}$ | | | | |
| 3 | $\frac{1}{8}$ | $\frac{3}{8}$ | $\frac{3}{8}$ | $\frac{1}{8}$ | | | |
| 4 | $\frac{7}{90}$ | $\frac{16}{45}$ | $\frac{2}{15}$ | $\frac{16}{45}$ | $\frac{7}{90}$ | | |
| 5 | $\frac{19}{288}$ | $\frac{25}{96}$ | $\frac{25}{144}$ | $\frac{25}{144}$ | $\frac{25}{96}$ | $\frac{19}{288}$ | |
| 6 | $\frac{41}{840}$ | $\frac{9}{35}$ | $\frac{9}{280}$ | $\frac{34}{105}$ | $\frac{9}{280}$ | $\frac{9}{35}$ | $\frac{41}{840}$ |

例 4-1 用梯形求积公式、Simpson 求积公式和 Newton-Cotes 求积公式 (取 $n=4$) 计算定积分 $\int_{0.5}^1 \sqrt{x} dx$.

解 ① 用梯形求积公式:

$$\int_{0.5}^1 \sqrt{x} dx \approx \frac{0.5}{2} (\sqrt{0.5} + 1) \approx 0.4268$$

② 用 Simpson 求积公式:

$$\int_{0.5}^1 \sqrt{x} dx \approx \frac{0.5}{6} (\sqrt{0.5} + 4\sqrt{0.75} + 1) \approx 0.4309$$

③ 用 Newton - Cotes 求积公式:

$$\begin{aligned} \int_a^b f(x) dx &\approx (b-a) \sum_{i=0}^n c_i^{(n)} f(x_i) \\ \int_{0.5}^1 \sqrt{x} dx &\approx 0.5 \sum_{i=0}^4 c_i^{(4)} \sqrt{x_i} \\ &= 0.5 \left[\frac{7}{90} \sqrt{0.5} + \frac{16}{45} \sqrt{0.625} + \frac{2}{15} \sqrt{0.75} + \right. \\ &\quad \left. \frac{16}{45} \sqrt{0.875} + \frac{7}{90} \sqrt{1} \right] \approx 0.4310 \end{aligned}$$

4.2 求积公式的代数精确度

定义 4.1 对一般求积近似公式, 如果当 $f(x)$ 为任意一个次数不高于 n 次的代数多项式时, 积分近似公式

$$\int_a^b f(x) dx \approx \sum_{k=0}^m A_k f(x_k) \quad (4.10)$$

精确成立, 而当 $f(x)$ 为 $n+1$ 次代数多项式时不精确成立, 则称该积分近似公式具有 n 次代数精确度.

从代数精确度的定义可知, 求积近似公式的精确度越高, 就会有越多的 $f(x)$ 使求积近似公式精确成立.

由于任何一个求积近似公式都可以写成 (4.10) 式的形式, 从而可以使用代数精确度的定义来证明求积公式具有多少次代数精确度.

定理 4.1 梯形求积公式具有一次代数精确度.

证明 ① 证明当 $f(x)$ 为任意一个不超过一次代数多项式时, 梯形求积公式精确成立.

在推导梯形求积公式时, 以 a 和 b 作为插值节点构造出 $p_1(x)$, 用 $p_1(x)$ 近似代替 $f(x)$, 即有

$$f(x) \approx p_1(x)$$

根据插值的误差估计式有

$$f(x) - p_1(x) = \frac{f''(\xi)}{2!} (x-a)(x-b) \quad (4.11)$$

对 (4.11) 式从 a 到 b 求积分得

$$\int_a^b f(x) dx - \int_a^b p_1(x) dx = \int_a^b \frac{f''(\xi)}{2!} (x-a)(x-b) dx$$

即

$$\int_a^b f(x) dx - \frac{b-a}{2}[f(a) + f(b)] = \int_a^b \frac{f''(\xi)}{2!}(x-a)(x-b) dx \quad (4.12)$$

显然, 当 $f(x)$ 为不超过一次代数多项式时, $f''(\xi) = 0$. 所以

$$\int_a^b f(x) dx = \frac{b-a}{2}[f(a) + f(b)]$$

也就是说, 当 $f(x)$ 为不超过一次代数多项式时, 梯形求积公式精确成立.

②证明当 $f(x)$ 为二次代数多项式时, 梯形求积公式不精确成立.

因为当 $f(x) = x^2$ 时, 有

$$\begin{aligned} \int_a^b f(x) dx &= \int_a^b x^2 dx = \frac{b^3 - a^3}{3} \\ \frac{b-a}{2}[f(a) + f(b)] &= \frac{b-a}{2}(a^2 + b^2) \end{aligned}$$

所以当 $f(x) = x^2$ 时,

$$\int_a^b f(x) dx \neq \frac{b-a}{2}[f(a) + f(b)]$$

由此推出, 当 $f(x)$ 为二次代数多项式时

$$\int_a^b f(x) dx \neq \frac{b-a}{2}[f(a) + f(b)]$$

因此, 梯形求积公式具有一次代数精确度.

从定理 4.1 可知, 当被积函数为不超过一次代数多项式时, 用梯形求积公式求积分是没有误差的.

定理 4.2 Newton - Cotes 求积公式至少具有 n 次代数精确度. 当 n 为偶数时, 代数精确度至少为 $n+1$ 次.

证明 ①证明 Newton - Cotes 求积公式至少具有 n 次代数精确度.

在推导 Newton - Cotes 求积公式时, 以 x_0, x_1, \dots, x_n 为插值节点构造插值多项式 $p_n(x)$, 用 $p_n(x)$ 代替 $f(x)$, 即 $f(x) \approx p_n(x)$. 根据插值的误差估计式有

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x) \quad (4.13)$$

其中 $\omega(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$. 对 (4.13) 式两边从 a 到 b 求积分得

$$\int_a^b f(x) dx - \int_a^b p_n(x) dx = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x) dx$$

由 (4.9) 式得

$$\int_a^b f(x) dx - \sum_{i=0}^n (b-a) c_i^{(n)} f(x_i) = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x) dx \quad (4.14)$$

显然, 当 $f(x)$ 为任意不超过 n 次代数多项式时, $f^{(n+1)}(\xi) = 0$. 所以

$$\int_a^b f(x) dx = \sum_{i=0}^n (b-a) c_i^{(n)} f(x_i)$$

这说明, Newton - Cotes 求积公式至少具有 n 次代数精确度.

②证明当 n 为偶数时, Newton - Cotes 求积公式的代数精确度至少为 $n+1$ 次.

设 $f(x)$ 为 $n+1$ 次代数多项式, 则可令 $f(x) = \sum_{k=0}^{n+1} b_k x^k$, 其中 n 为偶数, 从而有

$$f^{(n+1)}(x) = (b_0 + b_1 x + \cdots + b_n x^n + b_{n+1} x^{n+1})^{(n+1)} = b_{n+1} (n+1)!$$

故

$$f^{(n+1)}(\xi) = b_{n+1} (n+1)!$$

再据(4.14)式有

$$\begin{aligned} \int_a^b f(x) dx - \sum_{i=0}^n (b-a) c_i^{(n)} f(x_i) &= \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x) dx \\ &= b_{n+1} \int_a^b \omega(x) dx \\ &= b_{n+1} \int_a^b (x-x_0)(x-x_1)\cdots(x-x_n) dx \\ &\stackrel{x=a+th}{=} b_{n+1} h^{n+2} \int_0^n t(t-1)(t-2)\cdots(t-n) dt \end{aligned}$$

因为 n 为偶数, 可令 $n=2k, k=1, 2, \cdots$. 此时,

$$\begin{aligned} &\int_0^n t(t-1)(t-2)\cdots(t-n) dt \\ &= \int_0^{2k} t(t-1)\cdots(t-k)[t-(k+1)]\cdots[t-(2k-1)](t-2k) dt \\ &\stackrel{u=t-k}{=} \int_{-k}^k (u+k)(u+k-1)\cdots u(u-1)\cdots(u-k+1)(u-k) du \end{aligned}$$

令 $H(u) = (u+k)(u+k-1)\cdots u(u-1)\cdots(u-k+1)(u-k)$, 则有 $H(-u) = -H(u)$, 即 $H(u)$ 为奇函数. 于是

$$\int_0^n t(t-1)(t-2)\cdots(t-n) dt = 0$$

所以, 当 $f(x)$ 为 $n+1$ 次代数多项式, 且 n 为偶数时, 有

$$\int_a^b f(x) dx = \sum_{i=0}^n (b-a) c_i^{(n)} f(x_i)$$

这说明, 当 n 为偶数时, Newton - Cotes 求积公式的代数精确度至少为 $n+1$ 次.

从定理 4.2 可知, 当被积函数为不超过 n 次代数多项式时, 用 Newton - Cotes 求积公式求积分是没有误差的.

定理 4.3 Simpson 求积公式的代数精确度为 3.

证明 由定理 4.2 知, Simpson 求积公式的代数精确度至少为 3 次.

现取 $f(x) = x^4$, 则由 (4.1) 式有

$$\int_a^b f(x) dx = \int_a^b x^4 dx = \frac{b^5 - a^5}{5}$$

$$\text{而} \quad \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] = \frac{b-a}{6} \left[a^4 + 4\left(\frac{a+b}{2}\right)^4 + b^4 \right]$$

$$\text{从而} \quad \int_a^b f(x) dx \neq \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

由此推知, 当 $f(x)$ 为 4 次代数多项式时

$$\int_a^b f(x) dx \neq \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

这说明, 当 $f(x)$ 为 4 次代数多项式时, Simpson 求积公式不能精确成立. 因此, Simpson 求积公式的代数精确度为 3.

从定理 4.3 可知, 当被积函数为不超过 3 次代数多项式时, 用 Simpson 求积公式求积分是没有误差的.

4.3 梯形求积公式和 Simpson 求积公式的误差估计

定理 4.4 若 $f(x) \in C^2[a, b]$, 则梯形求积公式有误差估计

$$\int_a^b f(x) dx - \frac{b-a}{2} [f(a) + f(b)] = -\frac{(b-a)^3}{12} f''(\eta) \quad (4.15)$$

其中 $a \leq \eta \leq b$.

证明 据 (4.12) 式

$$\int_a^b f(x) dx - \frac{b-a}{2} [f(a) + f(b)] = \int_a^b \frac{f''(\xi)}{2} (x-a)(x-b) dx$$

因为 ξ 依赖于 x , 所以 $f''(\xi)$ 是 x 的函数. 根据题意 $f''(\xi)$ 在 $[a, b]$ 上连续, 而 $(x-a)(x-b)$ 在 $[a, b]$ 上小于 0, 从而由积分中值定理可知在 $[a, b]$ 上存在一点 η , 使

$$\begin{aligned} \int_a^b \frac{f''(\xi)}{2} (x-a)(x-b) dx &= f''(\eta) \int_a^b (x-a)(x-b) dx \\ &= -\frac{(b-a)^3}{12} f''(\eta) \end{aligned}$$

于是, 得到梯形求积公式的误差估计为

$$\int_a^b f(x) dx - \frac{b-a}{2} [f(a) + f(b)] = -\frac{(b-a)^3}{12} f''(\eta)$$

其中 $a \leq \eta \leq b$.

定理 4.5 若 $f(x) \in C^4[a, b]$, 则 Simpson 求积公式有误差估计

$$\int_a^b f(x) dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] = -\frac{(b-a)^5}{2880} f^{(4)}(\eta) \quad (4.16)$$

其中 $a \leq \eta \leq b$.

证明 首先考虑构造一个三次插值多项式 $p_3(x)$ 满足条件: $p_3(a) = f(a)$, $p_3(b) = f(b)$, $p_3\left(\frac{a+b}{2}\right) = f\left(\frac{a+b}{2}\right)$ 和 $p'_3\left(\frac{a+b}{2}\right) = f'\left(\frac{a+b}{2}\right)$. 可以证明, 满足上述条件的 $p_3(x)$ 与 $f(x)$ 的误差为

$$f(x) - p_3(x) = \frac{f^{(4)}(\xi)}{4!} (x-a) \left[x - \frac{a+b}{2} \right]^2 (x-b) \quad (4.17)$$

其中 $a \leq \xi \leq b$.

对 (4.17) 式两边求积分得

$$\int_a^b f(x) dx - \int_a^b p_3(x) dx = \frac{1}{4!} \int_a^b f^{(4)}(\xi) (x-a) \left[x - \frac{a+b}{2} \right]^2 (x-b) dx$$

又因 $p_3(x)$ 为三次代数多项式, 而 Simpson 求积公式的代数精确度为 3, 所以有

$$\begin{aligned} \int_a^b p_3(x) dx &= \frac{b-a}{6} \left[p_3(a) + 4p_3\left(\frac{a+b}{2}\right) + p_3(b) \right] \\ &= \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \end{aligned}$$

从而有

$$\begin{aligned} \int_a^b f(x) dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \\ = \frac{1}{4!} \int_a^b f^{(4)}(\xi) (x-a) \left[x - \frac{a+b}{2} \right]^2 (x-b) dx \end{aligned}$$

据题设, $f^{(4)}(\xi)$ 在 $[a, b]$ 上连续, 而 $(x-a) \left[x - \frac{a+b}{2} \right]^2 (x-b) \leq 0$, 由积分中值定理可知, 在 $[a, b]$ 中存在一点 η , 使

$$\begin{aligned} \int_a^b f^{(4)}(\xi) (x-a) \left[x - \frac{a+b}{2} \right]^2 (x-b) dx \\ = f^{(4)}(\eta) \int_a^b (x-a) \left[x - \frac{a+b}{2} \right]^2 (x-b) dx \\ = -\frac{(b-a)^5}{120} f^{(4)}(\eta) \end{aligned}$$

因此

$$\int_a^b f(x) dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

$$\begin{aligned}
 &= \frac{1}{4!} \left[-\frac{(b-a)^5}{120} f^{(4)}(\eta) \right] \\
 &= -\frac{(b-a)^5}{2 \cdot 880} f^{(4)}(\eta)
 \end{aligned}$$

4.4 复化求积公式

在 4.1 节中介绍了梯形求积公式、Simpson 求积公式和 Newton - Cotes 求积公式. 从梯形求积公式和 Simpson 求积公式的几何意义来看, 使用它们求积分会产生较大的误差, 而使用 Newton - Cotes 公式似乎当 n 越大, 计算出来的结果与积分准确值的误差就会越小. 但事实并非如此, 可以证明, 当 $n \rightarrow \infty$ 时, 并非对所有的 $f(x)$, 都有

$$\int_a^b f(x) dx - \sum_{i=0}^n (b-a) c_i^{(n)} f(x_i) \rightarrow 0$$

这说明 Newton - Cotes 求积公式的收敛性对某些被积函数 $f(x)$ 得不到保证. 另一方面, 还可以证明当 $n \geq 8$ 时, Newton - Cotes 求积公式的稳定性也得不到保证.

基于上述两个原因, 在实际使用 Newton - Cotes 求积公式计算定积分时, 不能通过盲目加大 n 以达到提高精度的目的.

为了提高积分近似公式的精度, 可以使用复化公式. 本节介绍两个复化公式: 一个是复化梯形求积公式, 另一个是复化 Simpson 求积公式.

4.4.1 复化梯形求积公式及其误差估计

把积分区间 $[a, b]$ n 等分, 得到 $n+1$ 个分点 $x_0, x_1, \dots, x_k, \dots, x_n$, 其中 $x_k = a + kh; k=0, 1, \dots, n; h=(b-a)/n$, 使用这些分点把 $[a, b]$ 分为 n 个小区间 $[x_0, x_1], [x_1, x_2], \dots, [x_{n-1}, x_n]$. 据定积分的性质和梯形求积公式(4.4)得

$$\begin{aligned}
 \int_a^b f(x) dx &= \int_{x_0}^{x_1} f(x) dx + \int_{x_1}^{x_2} f(x) dx + \dots + \int_{x_{n-1}}^{x_n} f(x) dx \\
 &= \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x) dx \\
 &\approx \sum_{k=0}^{n-1} \frac{x_{k+1} - x_k}{2} [f(x_k) + f(x_{k+1})] \\
 &= \frac{h}{2} \sum_{k=0}^{n-1} [f(x_k) + f(x_{k+1})] \\
 &= \frac{h}{2} [f(a) + f(b) + 2 \sum_{k=0}^{n-1} f(a + kh)] \quad (4.18)
 \end{aligned}$$

(4.18) 式称为复化梯形求积公式, 记为 T_n , 其中 n 表示等分数.

使用复化梯形公式求积分算法

① 输入 a, b 和 n .

② $h = \frac{b-a}{n}$.

③ $\text{sum} = 0$.

对 $k = 1, 2, \dots, n-1$, 做

$\text{sum} = \text{sum} + f(a + kh)$

④ $T = \frac{h}{2}[f(a) + f(b) + 2 * \text{sum}]$

⑤ 输出 T .

定理 4.6 若 $f(x) \in C^2[a, b]$, 则对复化梯形求积公式 T_n , 有

$$\int_a^b f(x) dx - T_n = -\frac{b-a}{12} h^2 f''(\eta) \quad (4.19)$$

其中 $h = \frac{b-a}{n}$, $a \leq \eta \leq b$.

证明 把积分区间 $[a, b]$ n 等分, 得到分点 x_0, x_1, \dots, x_n , 使用这些分点把 $[a, b]$ 分为 n 个小区间 $[x_0, x_1], [x_1, x_2], \dots, [x_{n-1}, x_n]$, 在这些小区间上直接使用梯形求积公式的误差估计式(4.15), 得

$$\begin{aligned} \int_a^b f(x) dx - T_n &= \sum_{k=0}^{n-1} \left\{ \int_{x_k}^{x_{k+1}} f(x) dx - \frac{h}{2} [f(x_k) + f(x_{k+1})] \right\} \\ &= -\frac{h^3}{12} \sum_{k=0}^{n-1} f''(\eta_k) \end{aligned}$$

其中 $x_k \leq \eta_k \leq x_{k+1}$.

由题设可知, $f''(x)$ 在 $[a, b]$ 上连续, 根据连续函数的性质可知, 在 $[a, b]$ 中存在一点 η , 使

$$\frac{1}{n} \sum_{k=0}^{n-1} f''(\eta_k) = f''(\eta)$$

因此, 得到复化梯形求积公式的误差估计为

$$\int_a^b f(x) dx - T_n = -\frac{nh^3}{12} f''(\eta) = -\frac{b-a}{12} h^2 f''(\eta)$$

当使用复化梯形求积公式求积分时, 有时可以使用它们的误差估计式判断 n 应取多大才能满足给定的精度要求. 换句话说, 就是要把积分区间 $[a, b]$ 划分多少等分才能满足给定精度要求.

例 4-2 计算积分 $\int_0^1 e^x dx$ 的近似值, 要求保证有 5 位有效数字. 若用复化梯形求积公式计算, n 应取多少?

解 据复化梯形求积公式的误差估计式(4.19),有

$$\int_0^1 e^x dx - T_n = -\frac{1}{12n^2}e^\eta$$

其中 $0 \leq \eta \leq 1$. 对上式两边取绝对值得

$$\left| \int_0^1 e^x dx - T_n \right| = \frac{1}{12n^2}e^\eta \leq \frac{1}{12n^2}e$$

按题目要求,选择 n 使得用复化梯形求积公式计算该积分时具有 5 位有效数字. 据题意和有效数字的定义,即须选择 n 满足

$$\frac{1}{12n^2}e \leq \frac{1}{2} \times 10^{p-5}$$

其中 p 由 $\int_0^1 e^x dx$ 的准确值确定.

显然,

$$\int_0^1 e^x dx = e^1 - e^0 = 0.1718 \cdots = 0.1718 \cdots \times 10^1$$

从而得 $p=1$. 故须选择 n 满足

$$\begin{aligned} \frac{1}{12n^2}e &\leq \frac{1}{2} \times 10^{-4} \\ n &\geq \sqrt{\frac{e \times 10^4}{6}} \approx 67.3088 \end{aligned}$$

因此,只要取 $n=68$ 就有 $\frac{1}{12n^2}e^\eta \leq \frac{1}{2} \times 10^{-4}$, T_n 就具有 5 位有效数字.

定理 4.7 积分近似值 T_n 与 T_{2n} 有如下关系:

$$T_{2n} = \frac{1}{2} \left[T_n + h \sum_{k=1}^n f \left(a + (2k-1) \frac{b-a}{2n} \right) \right] \quad (4.20)$$

其中 $h = \frac{b-a}{n}$.

证明 把区间 $[a, b]$ $2n$ 等分,实际上是在将区间 $[a, b]$ n 等分的基础上,对每个小区间 $[x_k, x_{k+1}]$ 再平分一次,亦即在每个小区间 $[x_k, x_{k+1}]$ 上插入一个中点. 记这个中点为 $x_{k+\frac{1}{2}}$, 则 $x_{k+\frac{1}{2}} = x_k + \frac{h}{2} = a + \left[k + \frac{1}{2} \right] h, k=0, 1, \cdots, n-1$, 从而有

$$\begin{aligned} \int_a^b f(x) dx &= \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x) dx = \sum_{k=0}^{n-1} \left[\int_{x_k}^{x_{k+\frac{1}{2}}} f(x) dx + \int_{x_{k+\frac{1}{2}}}^{x_{k+1}} f(x) dx \right] \\ &\approx \sum_{k=0}^{n-1} \left\{ \frac{x_{k+\frac{1}{2}} - x_k}{2} [f(x_k) + f(x_{k+\frac{1}{2}})] + \frac{x_{k+1} - x_{k+\frac{1}{2}}}{2} [f(x_{k+\frac{1}{2}}) + f(x_{k+1})] \right\} \end{aligned}$$

$$\begin{aligned}
&= \frac{h}{4} \sum_{k=0}^{n-1} \left[f(x_k) + 2f\left(x_{k+\frac{1}{2}}\right) + f(x_{k+1}) \right] \\
&= \frac{1}{2} \left[T_n + h \sum_{k=0}^{n-1} f\left(x_{k+\frac{1}{2}}\right) \right] \\
&= \frac{1}{2} \left[T_n + h \sum_{k=1}^n f\left[a + \left(k-1 + \frac{1}{2}\right)h\right] \right] \\
&= \frac{1}{2} \left[T_n + h \sum_{k=1}^n f\left[a + (2k-1)\frac{b-a}{2n}\right] \right]
\end{aligned}$$

从而得到

$$T_{2n} = \frac{1}{2} \left[T_n + h \sum_{k=1}^n f\left[a + (2k-1)\frac{b-a}{2n}\right] \right]$$

4.4.2 复化 Simpson 求积公式及其误差估计

把积分区间 $[a, b]$ n 等分, n 为偶数, 不妨设 $n = 2m$ ($m = 1, 2, \dots$), 则分点为 x_0, x_1, \dots, x_{2m} . 其中 $x_k = a + kh$; $k = 0, 1, \dots, 2m$; $h = (b-a)/n$. 使用这些分点把 $[a, b]$ 分为 m 个小区间 $[x_0, x_2]$, $[x_2, x_4]$, \dots , $[x_{2(m-1)}, x_{2m}]$. 在每个小区间 $[x_{2(k-1)}, x_{2k}]$ ($k = 1, 2, \dots, m$) 上采用 Simpson 求积公式 (4.6) 得

$$\begin{aligned}
\int_{x_{2(k-1)}}^{x_{2k}} f(x) dx &= \int_{x_{2k-2}}^{x_{2k}} f(x) dx \\
&\approx \frac{x_{2k} - x_{2k-2}}{6} [f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k})] \\
&= \frac{h}{3} [f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k})]
\end{aligned}$$

从而有

$$\begin{aligned}
\int_a^b f(x) dx &= \int_{x_0}^{x_2} f(x) dx + \int_{x_2}^{x_4} f(x) dx + \dots + \int_{x_{2(m-1)}}^{x_{2m}} f(x) dx \\
&= \sum_{k=1}^m \int_{x_{2(k-1)}}^{x_{2k}} f(x) dx \approx \sum_{k=1}^m \frac{h}{3} [f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k})] \\
&= \frac{h}{3} [f(x_0) + \sum_{k=2}^m f(x_{2k-2}) + 4 \sum_{k=1}^m f(x_{2k-1}) + \sum_{k=1}^m f(x_{2k}) + f(x_{2m})] \\
&= \frac{h}{3} [f(a) + f(b) + 4 \sum_{k=1}^m f(x_{2k-1}) + 2 \sum_{k=1}^{m-1} f(x_{2k})] \quad (4.21)
\end{aligned}$$

(4.21) 式称为复化 Simpson 求积公式或复化抛物线求积公式, 记为 S_n , 其中 n 为等分数.

定理 4.8 若 $f(x) \in C^4[a, b]$, 则对复化 Simpson 求积公式 S_n 有

$$\int_a^b f(x) dx - S_n = -\frac{b-a}{2 \cdot 880} (2h)^4 f^{(4)}(\eta) \quad (4.22)$$

其中 $h = \frac{b-a}{n}$; $a \leq \eta \leq b$.

证明 把积分区间 $[a, b]$ n 等分, 其中 $n = 2m$ ($m = 1, 2, \dots$), 则分点为 x_0, x_1, \dots, x_{2m} . 这些分点把 $[a, b]$ 分为 m 个小区间 $[x_0, x_2], [x_2, x_4], \dots, [x_{2(m-1)}, x_{2m}]$, 在每个小区间 $[x_{2(k-1)}, x_{2k}]$ ($k = 1, 2, \dots, m$) 上采用 Simpson 求积公式的误差估计式 (4.16) 得

$$\begin{aligned} \int_a^b f(x) dx - S_n &= \sum_{k=1}^m \left\{ \int_{x_{2(k-1)}}^{x_{2k}} f(x) dx - \frac{x_{2k} - x_{2k-2}}{3} [f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k})] \right\} \\ &= -\frac{(2h)^5}{2 \cdot 880} \sum_{k=1}^m f^{(4)}(\eta_k) \end{aligned}$$

其中 $x_{2k-2} \leq \eta_k \leq x_{2k}$.

由题设可知 $f''(x)$ 在 $[a, b]$ 上连续, 根据连续函数的性质, 在 $[a, b]$ 中存在一点 η , 使

$$\frac{1}{m} \sum_{k=1}^m f^{(4)}(\eta_k) = f^{(4)}(\eta)$$

因此, 得到复化 Simpson 求积公式的误差估计式

$$\int_a^b f(x) dx - S_n = -\frac{b-a}{2 \cdot 880} (2h)^4 f^{(4)}(\eta)$$

使用复化 Simpson 公式求积分算法

① 输入 a, b 和 n .

② 计算 $h = \frac{b-a}{n}$.

③ $S1 = 0, S2 = 0$

对 $k = 1, 2, \dots, m$, 做

$$S1 = S1 + f(a + (2 * k - 1) * h)$$

对 $k = 1, 2, \dots, m-1$, 做

$$S2 = S2 + f(a + 2 * k * h)$$

④ 计算 $S = \frac{h}{3} (f(a) + f(b) + 4 * S1 + 2 * S2)$.

⑤ 输出 S .

4.5 自动选取步长梯形法

在 4.4 节介绍了复化梯形求积公式。使用复化梯形求积公式求积分的近似值

是一种比较有效的方法. 但是, 在使用复化梯形求积公式求积分之前必须选定 n .

对复化梯形求积公式的误差估计式(4.19)的两边求绝对值得

$$\left| \int_a^b f(x) dx - T_n \right| = \frac{b-a}{12} h^2 |f''(\eta)| \leq \frac{b-a}{12} h^2 \max_{a \leq x \leq b} |f''(x)| \quad (4.23)$$

从(4.23)式可以看到, 如果 n 取得太大, 虽然能使复化梯形求积公式与积分精确值的误差变得很小, 但会导致计算量的增加. 如果 n 取得太小, 复化梯形求积公式与积分精确值的误差就会很大, 从而使复化梯形求积公式的精度难以保证. 因此, 假如给定一个精度, 那如何选择 n 使积分近似值满足精度要求? 为了避免盲目选取 n , 可以使用自动选取步长梯形法.

自动选取步长梯形法需要反复使用公式(4.20):

$$T_{2n} = \frac{1}{2} \left[T_n + h \sum_{k=1}^n f \left(a + (2k-1) \frac{b-a}{2n} \right) \right]$$

根据复化梯形求积公式的误差估计式(4.19)得

$$\int_a^b f(x) dx - T_n = -\frac{b-a}{12} h^2 f''(\eta_n) \quad (a \leq \eta_n \leq b)$$

$$\text{和} \quad \int_a^b f(x) dx - T_{2n} = -\frac{b-a}{12} \left(\frac{h}{2} \right)^2 f''(\eta_{2n}) \quad (a \leq \eta_{2n} \leq b)$$

将上两式相减得

$$T_{2n} - T_n = -\frac{b-a}{12} \left[\frac{h}{2} \right]^2 [4f''(\eta_n) - f''(\eta_{2n})]$$

当 $f''(x)$ 在 $[a, b]$ 上连续, 并假设 n 充分大时 $f''(\eta_n) \approx f''(\eta_{2n})$, 则有

$$\frac{1}{3} (T_{2n} - T_n) \approx \int_a^b f(x) dx - T_{2n}$$

因此, 假如给定一个精度 ε , 要求 $\left| \int_a^b f(x) dx - T_{2n} \right| < \varepsilon$, 则可以用 $|T_{2n} - T_n| < 3\varepsilon$ 近似判断 T_{2n} 是否满足精度要求.

自动选取步长梯形法的计算步骤为:

①先用梯形求积公式(4.4)计算出积分的第一次近似值

$$T_1 = \frac{b-a}{2} [f(a) + f(b)]$$

②把积分区间 $[a, b]$ 2 等分, 并用(4.20)式求出积分的第二次近似值

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

然后判别 $|T_2 - T_1| < 3\varepsilon$ 是否成立, 若成立, 则计算停止, T_2 作为积分近似值; 否则, 进行下一步.

③把积分区间 $[a, b]$ 4等分,并用(4.20)式求出积分的第三次近似值

$$\begin{aligned} T_4 &= \frac{1}{2} \left[T_2 + \frac{b-a}{2} \sum_{k=1}^2 f \left[a + (2k-1) \frac{b-a}{4} \right] \right] \\ &= \frac{T_2}{2} + \frac{b-a}{4} \sum_{k=1}^2 f \left[a + (2k-1) \frac{b-a}{4} \right] \end{aligned}$$

然后判别 $|T_4 - T_2| < 3\varepsilon$ 是否成立,若成立,则计算停止, T_4 作为积分近似值;否则,进行下一步.

④把积分区间 $[a, b]$ 8等分,并用(4.20)式求出积分的第四次近似值

$$\begin{aligned} T_8 &= \frac{1}{2} \left[T_4 + \frac{b-a}{4} \sum_{k=1}^2 f \left[a + (2k-1) \frac{b-a}{8} \right] \right] \\ &= \frac{T_4}{2} + \frac{b-a}{8} \sum_{k=1}^4 f \left[a + (2k-1) \frac{b-a}{8} \right] \end{aligned}$$

然后判别 $|T_8 - T_4| < 3\varepsilon$ 是否成立,若成立,则计算停止, T_8 作为积分近似值;否则,要进行下一步.

.....

按这样一直做下去,直到 $|T_{2n} - T_n| < 3\varepsilon$ 成立为止. T_{2n} 即为满足精度 ε 的积分近似值.

自动选取步长梯形法算法

①输入 a, b 和 ε .

②计算 $h = \frac{b-a}{2}, T_1 = (f(a) + f(b)) \times h, n = 1$

③计算 $T_0 = T_1, S = 0$ (T_0 表示前次积分近似值, T_1 表示后次积分近似值).

④对 $k = 1, 2, \dots, n$,计算

$$S = S + f \left[a + (2k-1) \times \frac{h}{n} \right]$$

⑤ $T_1 = \frac{T_0}{2} + S \times \frac{h}{n}$

⑥若 $|T_1 - T_0| < 3\varepsilon$,则输出 T_1 的值,结束计算.否则, $n = 2n$,返回③.

4.6 数值方法中的加速收敛技巧——Richardson 外推算法

Richardson(李查逊)外推算法在数值分析中有很多应用,如用在数值积分上就产生一种新的数值积分方法——Romberg(龙贝格)求积法. Romberg 求积法在下一节介绍,本节介绍 Richardson 外推算法.

假设有一个量 F^* ,这个量可以是函数、微分和积分等,现用一个以步长 h 为变

量的函数 $F_0(h)$ 近似代替它, 并设 F^* 与 h 无关, 且给定 F^* 与 $F_0(h)$ 的误差估计为

$$F^* - F_0(h) = a_1 h^{p_1} + a_2 h^{p_2} + \cdots + a_k h^{p_k} + \cdots \quad (4.25)$$

其中 $0 < p_1 < p_2 < \cdots < p_k < \cdots; a_k \neq 0 (k=1, 2, \cdots)$ 是与 h 无关的常数.

从误差估计式 (4.25) 可以看到, 当 h 适当小时, 第一项 $a_1 h^{p_1}$ 的绝对值远比后面的项的绝对值大. 而第一项中的 h^{p_1} 是最主要的, 它对误差的影响最大, 通常称为 F^* 与函数 $F_0(h)$ 的误差的阶. 显然, 当 h 适当小时, 误差的阶 h^{p_1} 中的幂 p_1 越大, F^* 与函数 $F_0(h)$ 的误差的绝对值就会越小. 能否从函数 $F_0(h)$ 出发构造一个新的函数近似代替 F^* , 使 F^* 与这个函数的误差的阶的幂比 F^* 与函数 $F_0(h)$ 的误差的阶的幂更高一些? 答案是肯定的.

为了构造出一个新函数 $F_2(h)$ 近似代替 F^* , 使 F^* 与这个函数的误差的阶的幂比 F^* 与函数 $F_0(h)$ 的误差的阶的幂更高一些, 可以采用下面的做法.

首先把 (4.25) 式中的 h 用 qh 代替, 得

$$F^* - F_0(qh) = a_1 (qh)^{p_1} + a_2 (qh)^{p_2} + \cdots + a_k (qh)^{p_k} + \cdots \quad (4.26)$$

其中 q 为常数, 并满足 $1 - q^{p_1} \neq 0$. 再用 q^{p_1} 乘 (4.25) 式两边, 得

$$q^{p_1} F^* - q^{p_1} F_0(h) = a_1 (qh)^{p_1} + a_2 q^{p_1} h^{p_2} + \cdots + a_k q^{p_1} h^{p_k} + \cdots \quad (4.27)$$

(4.26) 式减 (4.27) 式并整理后得

$$\begin{aligned} & (1 - q^{p_1}) F^* - [F_0(qh) - q^{p_1} F_0(h)] \\ &= a_2 (q^{p_2} - q^{p_1}) h^{p_2} + a_3 (q^{p_3} - q^{p_1}) h^{p_3} + \cdots + a_k (q^{p_k} - q^{p_1}) h^{p_k} + \cdots \end{aligned} \quad (4.28)$$

用 $1 - q^{p_1}$ 除 (4.28) 式两边得

$$\begin{aligned} F^* - \frac{F_0(qh) - q^{p_1} F_0(h)}{1 - q^{p_1}} \\ = \frac{a_2 (q^{p_2} - q^{p_1}) h^{p_2}}{1 - q^{p_1}} + \frac{a_3 (q^{p_3} - q^{p_1}) h^{p_3}}{1 - q^{p_1}} + \cdots + \frac{a_k (q^{p_k} - q^{p_1}) h^{p_k}}{1 - q^{p_1}} + \cdots \end{aligned} \quad (4.29)$$

若记 $F_1(h) = \frac{F_0(qh) - q^{p_1} F_0(h)}{1 - q^{p_1}}$ 和 $a_k^{(1)} = \frac{a_k (q^{p_k} - q^{p_1})}{1 - q^{p_1}}, k=2, 3, \cdots$, 则由 (4.29) 式有

$$F^* - F_1(h) = a_2^{(1)} h^{p_2} + a_3^{(1)} h^{p_3} + \cdots + a_k^{(1)} h^{p_k} + \cdots \quad (4.30)$$

其中 $a_k^{(1)}$ 是与 h 无关的常数.

从 (4.30) 式可以看到, 若用函数 $F_1(h)$ 近似代替 F^* , 则 F^* 与 $F_1(h)$ 的误差的阶为 h^{p_2} . 而 $p_2 > p_1$, 因此, 只要 h 适当小, F^* 与 $F_1(h)$ 的误差的绝对值比 F^* 与 $F_0(h)$ 的误差的绝对值要小些.

有了 $F_1(h)$ 以后, 再从这个函数出发, 构造出一个新函数 $F_2(h)$ 近似代替 F^* , 使 F^* 与这个函数的误差的阶的幂比 F^* 与函数 $F_1(h)$ 的误差的阶的幂更高一些. 具体做法与前面的做法类似.

$$F^* - F_2(h) = a_3^{(2)} h^{p_3} + a_4^{(2)} h^{p_4} + \cdots + a_k^{(2)} h^{p_k} + \cdots \quad (4.31)$$

其中 $F_2(h) = \frac{F_1(qh) - q^{p_2} F_1(h)}{1 - q^{p_2}}, a_k^{(2)} = \frac{a_k^{(1)} (q^{p_k} - q^{p_2})}{1 - q^{p_2}} (k=3, 4, \cdots)$ 是与 h 无关的常数.

从(4.31)式可以看到,若用函数 $F_2(h)$ 近似代替 F^* ,则 F^* 与 $F_2(h)$ 的误差的阶为 h^{p_3} ,而 $p_3 > p_2$. 因此,只要 h 适当小, F^* 与 $F_2(h)$ 的误差的绝对值比 F^* 与 $F_1(h)$ 的误差的绝对值又要小些.

依此类推,就可以构造出一系列的函数 $F_0(h), F_1(h), F_2(h), \cdots$. 当 h 适当小时, F^* 与这些函数的误差的绝对值一个比一个小. 这些函数的一般形式为

$$F_m(h) = \frac{F_{m-1}(qh) - q^{p_m} F_{m-1}(h)}{1 - q^{p_m}} \quad (m = 1, 2, \cdots) \quad (4.32)$$

其中 q 为常数,并满足 $1 - q^{p_m} \neq 0$.

(4.32)式称为 Richardson 外推算法. 用归纳法可以证明:

$$F^* - F_m(h) = a_{m+1}^{(m)} h^{p_{m+1}} + a_{m+2}^{(m)} h^{p_{m+2}} + \cdots$$

其中, $a_k^{(m)} = \frac{a_k^{(m-1)} (q^{p_k} - q^{p_m})}{1 - q^{p_m}} (k = m+1, m+2, \cdots)$, 是与 h 无关的常数.

4.7 Romberg 求积法

Romberg 求积法是应用 Richardson 外推算法的一个典型例子. 在介绍 Romberg 求积法的具体计算过程之前,先介绍一下 Romberg 求积法的理论基础,推导一个求积分近似值的 Romberg 序列.

4.7.1 Romberg 序列的推导

4.4 节已经证明,(4.19)式是复化梯形求积公式误差估计式的一种形式.

还可以证明复化梯形求积公式的另一个误差估计式

$$\int_a^b f(x) dx - T_n = a_2 h^2 + a_4 h^4 + a_6 h^6 + \cdots \quad (4.33)$$

其中 $h = \frac{b-a}{n}$, a_2, a_4, a_6, \cdots 都是与步长 h 无关的常数.

从误差估计(4.33)式和(4.25)式可以看到, $p_1 = 2, p_2 = 4, p_3 = 6, \cdots$, 复化梯形求积公式与积分准确值的误差的阶为 h^2 .

由复化梯形求积公式(4.18)可知

$$T_n = \frac{h}{2} [f(a) + f(b) + 2 \sum_{k=1}^{n-1} f(a + kh)] = T_0(h)$$

把积分区间 $[a, b]$ $2n$ 等分, 并且采用复化梯形公式得

$$T_{2n} = \frac{h}{2} \left[f(a) + f(b) + 2 \sum_{k=1}^{2n-1} f\left(a + k \frac{h}{2}\right) \right] = T_0\left[\frac{h}{2}\right] \quad (4.34)$$

(4.34) 式实际上就是用 $\frac{h}{2}$ 替代 T_n 中的 h 得到的, 所以可以记为 $T_0\left[\frac{h}{2}\right]$.

再把积分区间 $[a, b]$ $4n$ 等分, 并且采用复化梯形公式得

$$T_{4n} = \frac{h}{2} \left[f(a) + f(b) + 2 \sum_{k=1}^{4n-1} f\left(a + k \frac{h}{2}\right) \right] = T_0\left[\frac{h}{2^2}\right] \quad (4.35)$$

(4.35) 式实际上就是用 $\frac{h}{2^2}$ 替代 T_n 中的 h 得到的, 所以可以记为 $T_0\left[\frac{h}{2^2}\right]$.

按此类推, 就能得到一个函数序列

$$T_0(h), T_0\left[\frac{h}{2}\right], T_0\left[\frac{h}{2^2}\right], \dots \quad (4.36)$$

可以证明, 函数序列 (4.36) 收敛于积分的准确值. 根据误差估计式 (4.33), 这个函数序列中的每一个函数与积分准确值的误差的阶为 h^2 .

按照 Richardson 外推算法, 可以从这个函数序列出发构造一个新的函数序列, 使新的函数序列收敛于积分准确值的速度比函数序列 (4.36) 收敛于积分准确值的速度更快些. 具体做法为:

取 $m=1$ 和 $q=\frac{1}{2}$ 代入 Richardson 外推公式 (4.32), 把 F 改为 T , 并注意到 $p_1=2$ 就得到一个新函数

$$T_1(h) = \frac{T_0\left[\frac{h}{2}\right] - \frac{1}{2^2}T_0(h)}{1 - \frac{1}{2^2}} \quad (4.37)$$

对 (4.37) 式用 $\frac{h}{2}$ 替代 h , 得

$$T_1\left[\frac{h}{2}\right] = \frac{T_0\left[\frac{h}{2^2}\right] - \frac{1}{2^2}T_0\left[\frac{h}{2}\right]}{1 - \frac{1}{2^2}}$$

再用 $\frac{h}{2^2}$ 替代 h , 得

$$T_1\left[\frac{h}{2^2}\right] = \frac{T_0\left[\frac{h}{2^3}\right] - \frac{1}{2^2}T_0\left[\frac{h}{2^2}\right]}{1 - \frac{1}{2^2}}$$

如此类推,就可以得到一个新的函数序列

$$T_1(h), T_1\left[\frac{h}{2}\right], T_1\left[\frac{h}{2^2}\right], \dots \quad (4.38)$$

可以证明,函数序列(4.38)收敛于积分的准确值.由 Richardson 外推算法可知,这些函数与积分准确值的误差的阶为 h^4 .所以,当 h 适当小时,该函数序列收敛于积分准确值的速度比函数序列(4.36)收敛于积分准确值的速度要快些.

取 $m=2$ 和 $q=\frac{1}{2}$ 代入 Richardson 外推公式(4.32),把 F 改为 T ,并注意到 $p_2=4$,仿照前面的方法,又可以得到一个新的函数序列

$$T_2(h), T_2\left[\frac{h}{2}\right], T_2\left[\frac{h}{2^2}\right], \dots \quad (4.39)$$

可以证明,函数序列(4.39)收敛于积分的准确值.由 Richardson 外推算法可知,这些函数与积分准确值的误差的阶为 h^6 .所以,当 h 适当小时,该函数序列收敛于积分准确值的速度比(4.36)和(4.38)两个函数序列收敛于积分准确值的速度要快些.

这样,不断地运用 Richardson 外推算法就得到一系列的函数序列:

$$\begin{aligned} &T_0(h), T_0\left[\frac{h}{2}\right], T_0\left[\frac{h}{2^2}\right], \dots \\ &T_1(h), T_1\left[\frac{h}{2}\right], T_1\left[\frac{h}{2^2}\right], \dots \\ &T_2(h), T_2\left[\frac{h}{2}\right], T_2\left[\frac{h}{2^2}\right], \dots \\ &\vdots \\ &T_m(h), T_m\left[\frac{h}{2}\right], T_m\left[\frac{h}{2^2}\right], \dots \\ &\vdots \end{aligned} \quad (4.40)$$

函数数列表(4.40)的函数计算可以归纳为统一公式

$$T_m\left[\frac{h}{2^k}\right] = \frac{T_{m-1}\left[\frac{h}{2^{k+1}}\right] - \frac{1}{2^{2m}}T_{m-1}\left[\frac{h}{2^k}\right]}{1 - \frac{1}{2^{2m}}} = \frac{4^m T_{m-1}\left[\frac{h}{2^{k+1}}\right] - T_{m-1}\left[\frac{h}{2^k}\right]}{4^m - 1} \quad (4.41)$$

其中, $T_0(h) = T_n$, $T_0\left[\frac{h}{2}\right] = T_{2n}$, $T_0\left[\frac{h}{2^2}\right] = T_{4n}, \dots; m=1, 2, \dots; k=0, 1, \dots$.

可以证明,函数数列表(4.40)中的每一行都收敛于积分的准确值.实际上,还可以证明函数序列

$$T_0(h), T_1(h), \dots, T_m(h), \dots \quad (4.42)$$

也收敛于积分准确值.函数序列(4.42)称为 Romberg 序列,使用该函数序列求积

分的方法称为 Romberg 求积法.

4.7.2 Romberg 求积法的计算过程

Romberg 求积法的计算过程如下:

$$\textcircled{1} T_n = T_0(h)$$

$$\textcircled{2} T_{2n} = T_0\left(\frac{h}{2}\right) \quad \textcircled{3} T_1(h)$$

$$\textcircled{4} T_{4n} = T_0\left(\frac{h}{2^2}\right) \quad \textcircled{5} T_1\left(\frac{h}{2}\right) \quad \textcircled{6} T_2(h)$$

$$\textcircled{7} T_{8n} = T_0\left(\frac{h}{2^3}\right) \quad \textcircled{8} T_1\left(\frac{h}{2^2}\right) \quad \textcircled{9} T_2\left(\frac{h}{2}\right) \quad \textcircled{10} T_3(h)$$

\vdots

按这样做下去,直到 $|T_m(h) - T_{m-1}(h)| < \varepsilon$ 为止.

需说明的是,在开始计算时,要先取定 n ,通常取 $n=1$.也要先取定 ε , ε 取得越小,积分近似值就越接近于积分准确值.

例 4-3 使用 Romberg 求积法计算积分 $\int_0^1 \frac{4}{1+x^2} dx$ 的值(给定 $\varepsilon=0.01$,且取 $n=1$).

解 $\textcircled{1} T_1 = T_0(h) = \frac{1-0}{2} \times (4+2) = 3$

$$\textcircled{2} T_2 = T_0\left(\frac{h}{2}\right) = \frac{1}{2} \left[T_1 + \frac{4}{1 + \left(\frac{1}{2}\right)^2} \right] = 3.1$$

$$\textcircled{3} T_1(h) = \frac{4T_0\left(\frac{h}{2}\right) - T_0(h)}{3} = 3.133\ 333\ 3$$

$$|T_1(h) - T_0(h)| = 0.133\ 333 > 0.01$$

进行下一步.

$$\textcircled{4} T_4 = T_0\left(\frac{h}{2^2}\right) = \frac{1}{2} \left[T_2 + \frac{4}{1 + \left(\frac{1}{4}\right)^2} + \frac{4}{1 + \left(\frac{3}{4}\right)^2} \right] = 3.131\ 18$$

$$\textcircled{5} T_1\left(\frac{h}{2}\right) = \frac{4T_0\left(\frac{h}{2^2}\right) - T_0\left(\frac{h}{2}\right)}{3} = 3.141\ 573$$

$$\textcircled{6} T_2(h) = \frac{4^2 T_1\left(\frac{h}{2}\right) - T_1(h)}{4^2 - 1} = 3.142\ 12$$

此时 $|T_2(h) - T_1(h)| = 0.008\ 79 < 0.01$

成立, 计算停止, $T_1(h)$ 作为满足精度要求的积分近似值.

前面介绍的是人工计算过程, 那么, 如何在计算机上实现 Romberg 求积法? 为了便于介绍在计算机上如何实现 Romberg 求积法, 把 $T_m \left[\frac{h}{2^k} \right]$ 记为 $T_m^{(k)}$, 则相应的

Romberg 求积法的计算过程如下所示:

$$\begin{array}{ccccccc}
 \textcircled{1} T_0^{(0)} & & & & & & \\
 \textcircled{2} T_0^{(1)} & \textcircled{3} T_1^{(0)} & & & & & \\
 \textcircled{4} T_0^{(2)} & \textcircled{5} T_1^{(1)} & \textcircled{6} T_2^{(0)} & & & & \\
 \textcircled{7} T_0^{(3)} & \textcircled{8} T_1^{(2)} & \textcircled{9} T_2^{(1)} & \textcircled{10} T_3^{(0)} & & & \\
 & & \vdots & & & &
 \end{array}$$

按这样做下去, 直到 $|T_m^{(0)} - T_{m-1}^{(0)}| < \varepsilon$ 成立为止.

由于在编写程序时, $T_m^{(k)}$ 的值要存放起来, 所以需使用一个二维数组 T , 即用数组元素 $T(m, k)$ 来存放 $T_m^{(k)}$ 的值.

Romberg 求积法算法

①输入 a, b 和 ε .

②计算 $T_0^{(0)}$:

$$T(0, 0) = \frac{b-a}{2} [f(a) + f(b)]$$

③ $k=1$ (其中 k 用来记录把积分区间 $[a, b]$ 2 等分的次数).

④按复化梯形公式计算 $T_0^{(k)}$:

$$T(0, k) = \frac{1}{2} \left[T(0, k-1) + \frac{b-a}{2^{k-1}} \sum_{i=1}^{2^{k-1}} f\left(a + (2i-1) \times \frac{b-a}{2^k}\right) \right]$$

⑤计算第 $k+1$ 行元素 $T_m^{(k-m)}$:

$$T(m, k-m) = \frac{4^m T(m-1, k-m+1) - T(m-1, k-m)}{4^m - 1} \quad (m=1, 2, \dots, k)$$

⑥精度控制.

对指定的精度 ε , 若 $|T(k, 0) - T(k-1, 0)| < \varepsilon$, 即 $|T_k^{(0)} - T_{k-1}^{(0)}| < \varepsilon$, 则终止计算, 并取 $T(k, 0)$ 即 $T_k^{(0)}$ 作为满足精度要求的积分近似值; 否则, $k=k+1$, 转回④继续计算.

4.8 Gauss 型求积公式

考虑定积分

$$\int_a^b \rho(x) f(x) dx$$

其中 $\rho(x)$ 为区间 $[a, b]$ 上的权函数.

以 n 个不等距的节点 $x_1, x_2, \dots, x_i, \dots, x_n, x_i \in [a, b]$ 作为插值节点, 根据代数插值的理论

$$f(x) = \sum_{k=1}^n f(x_k) \frac{\omega_n(x)}{(x-x_k)\omega'_n(x_k)} + f(x, x_1, x_2, \dots, x_n) \omega_n(x) \quad (4.43)$$

其中 $\omega_n(x) = (x-x_1)(x-x_2)\cdots(x-x_n)$.

将 (4.43) 式两边乘以权函数 $\rho(x)$ 并积分得

$$\begin{aligned} \int_a^b \rho(x) f(x) dx &= \int_a^b \rho(x) \left[\sum_{k=1}^n f(x_k) \frac{\omega_n(x)}{(x-x_k)\omega'_n(x_k)} \right] dx + \\ &\quad \int_a^b \rho(x) f(x, x_1, x_2, \dots, x_n) \omega_n(x) dx \\ &= \sum_{k=1}^n f(x_k) \int_a^b \frac{\rho(x) \omega_n(x)}{(x-x_k)\omega'_n(x_k)} dx + \\ &\quad \int_a^b \rho(x) f(x, x_1, x_2, \dots, x_n) \omega_n(x) dx \end{aligned} \quad (4.44)$$

令

$$\begin{aligned} A_k &= \int_a^b \frac{\rho(x) \omega_n(x)}{(x-x_k)\omega'_n(x_k)} dx \\ R[f] &= \int_a^b \rho(x) f(x, x_1, x_2, \dots, x_n) \omega_n(x) dx \end{aligned}$$

则有

$$\int_a^b \rho(x) f(x) dx = \sum_{k=1}^n A_k f(x_k) + R[f]$$

舍去余项 $R[f]$ 就得到积分近似计算公式

$$\int_a^b \rho(x) f(x) dx \approx \sum_{k=1}^n A_k f(x_k) \quad (4.45)$$

因为当 $f(x)$ 为不超过 $n-1$ 次代数多项式时, $f(x, x_1, x_2, \dots, x_n) = 0$, 从而推得

$$R[f] = \int_a^b \rho(x) f(x, x_1, x_2, \dots, x_n) \omega_n(x) dx = 0$$

因此, 公式 (4.45) 具有 $n-1$ 次代数精确度. 下面对给定的 $f(x)$, 寻找 x_k 和 A_k 使得求积公式 (4.45) 从 $n-1$ 次代数精确度提高到 $2n-1$ 次.

定理 4.9 若对任何一个不超过 $n-1$ 次的代数多项式 $q(x)$, 有

$$\int_a^b \rho(x) q(x) \omega_n(x) dx = 0 \quad (4.46)$$

则求积公式 (4.45) 具有 $2n-1$ 次代数精确度.

证明 设 $f(x)$ 是次数不高于 $2n-1$ 次的多项式, 用 $\omega_n(x)$ 除 $f(x)$, 若记 $q(x)$ 为商, $r(x)$ 为余式, 则有

$$f(x) = \omega_n(x) q(x) + r(x) \quad (4.47)$$

其中 $q(x)$ 和 $r(x)$ 都是不超过 $n-1$ 次的代数多项式. 用 $\rho(x)$ 乘 (4.47) 式两边, 并

求积分得

$$\int_a^b \rho(x)f(x) dx = \int_a^b \rho(x)q(x)\omega_n(x) dx + \int_a^b \rho(x)r(x) dx$$

又因为

$$\int_a^b \rho(x)q(x)\omega_n(x) dx = 0$$

再根据(4.44)式得

$$\int_a^b \rho(x)r(x) dx = \sum_{k=1}^n A_k r(x_k)$$

故

$$\int_a^b \rho(x)f(x) dx = \sum_{k=1}^n A_k r(x_k)$$

再因为

$$f(x_k) = \omega_n(x_k)q(x_k) + r(x_k) = r(x_k)$$

于是

$$\int_a^b \rho(x)f(x) dx = \sum_{k=1}^n A_k f(x_k)$$

这说明了当 $f(x)$ 为不超过 $2n-1$ 次代数多项式时,求积公式(4.45)准确成立.

还需证明求积公式(4.45)对 $f(x)$ 为 $2n$ 次代数多项式时是不准确成立的.为此,令 $f(x) = \omega_n^2(x)$.显然,它是 $2n$ 次多项式,且有

$$\int_a^b \rho(x)\omega_n^2(x) dx > 0$$

而另一方面,如果求积公式(4.45)对 $f(x) = \omega_n^2(x)$ 准确成立,就有

$$\int_a^b \rho(x)\omega_n^2(x) dx = \sum_{k=1}^n A_k \omega_n^2(x_k) = 0$$

这个矛盾说明,当 $f(x)$ 为 $2n$ 次多项式时,求积公式(4.45)是不准确成立的.因此,根据代数精确度的定义,求积公式(4.45)的代数精确度可达到 $2n-1$ 次.

这个事实说明,只要选取 x_k 满足(4.46)并取

$$A_k = \int_a^b \frac{\rho(x)\omega_n(x)}{(x-x_k)\omega'_n(x_k)} dx$$

则求积公式(4.45)的代数精确度可以从 $n-1$ 次提高到 $2n-1$ 次.

(4.46)式表明, n 次代数多项式 $\omega_n(x)$ 和不超过 $n-1$ 次代数多项式 $q(x)$ 在区间 $[a, b]$ 上对权函数 $\rho(x)$ 正交.根据正交条件可以解出 x_1, x_2, \dots, x_n .

下面考虑 $n=2$ 和 $\rho(x)=1$ 的情形.不失一般性,把积分区间 $[a, b]$ 取成 $[-1, 1]$.这是因为利用变换

$$x = \frac{a+b}{2} + \frac{b-a}{2}t$$

总可以把积分区间 $[a, b]$ 变成 $[-1, 1]$,即有

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 g(t) dt$$

其中 $g(t) = f\left[\frac{a+b}{2} + \frac{b-a}{2}t\right]$.

现在的问题是如何选取 x_1 和 x_2 , 使

$$\int_{-1}^1 f(x) dx \approx A_1 f(x_1) + A_2 f(x_2)$$

对 $f(x)$ 为三次代数多项式时准确成立.

按正交条件有

$$\int_{-1}^1 (x - x_1)(x - x_2) dx = 0$$

$$\int_{-1}^1 x(x - x_1)(x - x_2) dx = 0$$

从上面两式得 $x_1 + x_2 = 0$ 和 $x_1 x_2 = -\frac{1}{3}$, 从而得到 $x_2 = -x_1 = -\frac{1}{3}$, 而

$$A_1 = \int_{-1}^1 \frac{x - x_2}{x_1 - x_2} dx = 1$$

$$A_2 = \int_{-1}^1 \frac{x - x_1}{x_2 - x_1} dx = 1$$

于是

$$\int_{-1}^1 f(x) dx \approx f\left[-\frac{1}{3}\right] + f\left[\frac{1}{3}\right]$$

这个求积公式具有三次代数精确度, 称为 $n=2$ 时的高斯求积公式.

定义 4.2 一般地, 把按正交条件(4.46)求出的节点 x_k 和按公式(4.44)计算出的系数 A_k 组成的求积公式(4.45)称为 Gauss(高斯)型求积公式. 而节点 x_k 称为 Gauss 点.

Gauss 型求积公式的优点是代数精确度高, 但节点和系数的计算比较麻烦. 而利用区间 $[a, b]$ 上关于非负权函数 $\rho(x)$ 的正交多项式簇的性质可知, 正交多项式簇的 n 个零点就是 Gauss 求积公式的 n 个节点. 为此, 前人对某些特定的权函数与正交多项式事先计算出它们对应的 Gauss 点和系数表, 在计算时可以直接查表得到求积公式. 下面给出几种常用的 Gauss 型求积公式, 即给出它们的 Gauss 点和相应的系数.

4.8.1 Gauss - Legendre(勒让得)求积公式

首先, 引入 Legendre 多项式定义. Legendre 多项式为

$$L_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n$$

其中 $L_0(x) = 1$.

容易验证, $L_n(x)$ 是在区间 $[a, b]$ 上关于权函数 $\rho(x) = 1$ 的正交多项式簇. 这是因为

$$\int_{-1}^1 x^k \frac{d^n}{dx^n} (x^2 - 1)^n dx = x^k \frac{d^{n-1}}{dx^{n-1}} (x^2 - 1)^n \Big|_{-1}^1 - \int_{-1}^1 kx^{k-1} \frac{d^{n-1}}{dx^{n-1}} (x^2 - 1)^n dx$$

$$\begin{aligned}
 &= - \int_{-1}^1 kx^{k-1} \frac{d^{n-1}}{dx^{n-1}}(x^2 - 1)^n dx \\
 &= \cdots = (-1)^k k! \int_{-1}^1 \frac{d^{n-k}}{dx^{n-k}}(x^2 - 1)^n dx = 0 \quad (k = 0, 1, \cdots, n)
 \end{aligned}$$

因此,根据正交多项式簇的性质, $L_n(x)$ 的 n 个零点就是 Gauss-Legendre 求积公式的 n 个节点.

直接按公式(4.44)可以算出 A_k ,但这样计算比较复杂.考虑到

$$f(x) = \frac{\omega_n(x) \omega'_n(x)}{x - x_k}$$

是 $2n-2$ 次的代数多项式,而且

$$f(x_j) = \begin{cases} 0 & (j \neq k) \\ [\omega'_n(x_j)]^2 & (j = k) \end{cases}$$

于是

$$\begin{aligned}
 \int_{-1}^1 f(x) dx &= \int_{-1}^1 \frac{\omega'_n(x) \omega_n(x)}{x - x_k} dx \\
 &= \sum_{j=1}^n A_j f(x_j) = A_k [\omega'_n(x_k)]^2
 \end{aligned}$$

另一方面,由分部积分法得

$$\begin{aligned}
 \int_{-1}^1 \frac{\omega_n(x) \omega'_n(x)}{x - x_k} dx &= \int_{-1}^1 \frac{\omega_n(x)}{x - x_k} d(\omega_n(x)) \\
 &= \left. \frac{\omega_n^2(x)}{x - x_k} \right|_{-1}^1 - \int_{-1}^1 \omega_n(x) \left[\frac{\omega_n(x)}{x - x_k} \right]' dx \\
 &= \frac{2}{1 - x_k^2}
 \end{aligned}$$

从而有

$$A_k [\omega'_n(x_k)]^2 = \frac{2}{1 - x_k^2}$$

解得

$$A_k = \frac{2}{(1 - x_k^2)(\omega'_n(x_k))^2}$$

当 $n=1, 2, 3, 4, 5$ 时, Gauss - Legendre 求积公式的节点 x_k 和系数 A_k 见表4-2.

表4-2 Gauss - Legendre 求积公式节点 x_k 和系数 A_k

| n | x_k | A_k |
|-----|--|---|
| 1 | 0 | 2 |
| 2 | $\pm 0.577\ 350\ 3$ | 1 |
| 3 | $\pm 0.774\ 596\ 7=0$ | 0.555 555 6 0.888 888 9 |
| 4 | $\pm 0.861\ 136\ 3$ $\pm 0.339\ 981\ 0$ | 0.347 854 8 0.652 145 1 |
| 5 | $\pm 0.906\ 179\ 8$ $\pm 0.538\ 469\ 3=0$ | 0.236 926 8 0.478 628 7 0.568 888 9 |

例 4-4 使用两点的 Gauss - Legendre 求积公式计算积分 $\int_0^{\frac{\pi}{2}} \sin t dt$.

解
$$\int_0^{\frac{\pi}{2}} \sin t dt \stackrel{t = \frac{\pi(x+1)}{4}}{=} \frac{\pi}{4} \int_{-1}^1 \sin \frac{\pi(x+1)}{4} dx$$

查表得 $x_1 = 0.577\ 350\ 3, x_2 = -0.577\ 350\ 3, A_1 = A_2 = 1$, 而

$$\sin \frac{\pi(x_1+1)}{4} \approx 0.325\ 89 \quad \sin \frac{\pi(x_2+1)}{4} \approx 0.945\ 41$$

故
$$\begin{aligned} \int_0^{\frac{\pi}{2}} \sin t dt &= \frac{\pi}{4} \int_{-1}^1 \sin \frac{\pi(x+1)}{4} dx \\ &\approx \frac{\pi}{4} (0.325\ 89 + 0.945\ 41) \approx 0.998\ 475\ 8 \end{aligned}$$

4.8.2 Gauss - Leguerre(拉盖尔)求积公式

所谓 Leguerre 多项式为

$$L_n(x) = e^x \frac{d^n}{dx^n} (e^{-x} x^n)$$

它是在区间 $[0, \infty)$ 上关于权函数 $\rho(x) = e^{-x}$ 的正交多项式簇. 而求积公式

$$\int_0^{\infty} e^{-x} f(x) dx \approx \sum_{k=1}^n A_k f(x_k)$$

称为 Gauss - Leguerre 求积公式. 其中 x_k 是 Leguerre 多项式的零点; 系数 A_k 由下式计算:

$$A_k = \frac{(n!)^2}{x_k [L'_n(x_k)]^2}$$

当 $n = 1, 2, 3, 4, 5$ 时, Gauss - Leguerre 求积公式的节点 x_k 和系数 A_k 见表 4-3.

表 4-3 Gauss - Leguerre 求积公式节点 x_k 和系数 A_k

| n | x_k | A_k |
|-----|-------------|-------------|
| 1 | 1 | 1 |
| 2 | 0.585 786 4 | 0.853 553 4 |
| | 3.414 213 6 | 0.146 446 6 |
| 3 | 0.415 774 6 | 0.711 093 0 |
| | 2.294 280 4 | 0.278 517 7 |
| | 6.289 945 1 | 0.010 389 3 |

续上表

| n | x_k | A_k |
|-----|-------------|-------------|
| 4 | 0.322 547 7 | 0.603 154 1 |
| | 1.745 761 1 | 0.357 418 7 |
| | 4.536 620 3 | 0.038 887 9 |
| | 9.395 070 9 | 0.000 539 3 |
| 5 | 0.263 560 3 | 0.521 755 6 |
| | 1.413 403 1 | 0.398 666 8 |
| | 3.596 425 8 | 0.075 942 4 |
| | 7.085 810 0 | 0.003 611 8 |
| | 12.640 801 | 0.000 023 4 |

例 4-5 使用 Gauss - Leguerre 求积公式计算积分 $\int_0^\infty e^{-x} \sin x dx$.

解 取 $n=2$, 查表得 $x_1=0.585\ 786\ 4, x_2=3.414\ 213\ 6, A_1=0.853\ 553\ 4, A_2=0.146\ 446\ 6$. 故

$$\begin{aligned}\int_0^\infty e^{-x} \sin x dx &\approx 0.853\ 553\ 4 \sin(0.585\ 786\ 4) + 0.146\ 446\ 6 \sin(3.414\ 213\ 6) \\ &\approx 0.432\ 459\ 4\end{aligned}$$

4.8.3 Gauss - Hermite 求积公式

Hermite 多项式为

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} (e^{-x^2})$$

它是在区间 $(-\infty, +\infty)$ 上关于权函数 $\rho(x) = e^{-x^2}$ 的正交多项式簇. 而求积公式

$$\int_0^\infty e^{-x^2} f(x) dx \approx \sum_{k=1}^n A_k f(x_k)$$

称为 Gauss - Hermite 求积公式. 其中 x_k 是 Hermite 多项式的零点; 系数 A_k 由下式计算:

$$A_k = \frac{2^{n+1} n!}{[H'_n(x_k)]^2} \sqrt{\pi}$$

当 $n=1, 2, 3, 4, 5$ 时, Gauss - Hermite 求积公式的节点 x_k 和系数 A_k 见表 4-4.

表 4-4 Gauss-Hermite 求积公式节点 x_k 和系数 A_k

| n | x_k | A_k |
|-----|---------------------|-------------|
| 1 | 0 | 1.772 453 9 |
| 2 | $\pm 0.707\ 106\ 8$ | 0.886 226 9 |
| 3 | $\pm 1.224\ 744\ 9$ | 0.295 409 0 |
| | 0 | 1.181 635 9 |
| 4 | $\pm 1.650\ 680\ 1$ | 0.081 312 8 |
| | $\pm 0.524\ 647\ 6$ | 0.804 914 1 |
| 5 | $\pm 2.020\ 182\ 9$ | 0.019 953 2 |
| | $\pm 0.958\ 572\ 5$ | 0.393 619 3 |
| | 0 | 0.945 308 7 |

例 4-6 使用 Gauss-Hermite 求积公式计算积分 $\int_{-\infty}^{\infty} e^{-x^2} \sin^2 x dx$.

解 取 $n=2$, 查表得 $x_1=0.707\ 106\ 8, x_2=-0.707\ 106\ 8, A_1=A_2=0.886\ 226\ 9$. 故

$$\int_{-\infty}^{\infty} e^{-x^2} \sin^2 x dx \approx (\sin^2(0.707\ 106\ 8) + \sin^2(-0.707\ 106\ 8)) \times 0.886\ 226\ 9 \\ \approx 0.748\ 025\ 4$$

定理 4.10 设 $f(x) \in C^{2n}[a, b]$, 则 Gauss 求积公式的误差估计式为

$$\int_a^b \rho(x) f(x) dx - \sum_{k=1}^n A_k f(x_k) = \frac{f^{(2n)}(\eta)}{(2n)!} \int_a^b \rho(x) \omega_n^2(x) dx \quad (4.48)$$

其中 $a \leq \eta \leq b$.

证明 由 Hermite 插值得知, 在节点 x_1, x_2, \dots, x_n 上 $2n-1$ 次 Hermite 插值多项式 $H_{2n-1}(x)$ 与 $f(x)$ 误差估计为

$$f(x) - H_{2n-1}(x) = \frac{f^{(2n)}(\xi)}{(2n)!} \omega_n^2(x)$$

其中 $a \leq \xi \leq b$. 即

$$f(x) = H_{2n-1}(x) + \frac{f^{(2n)}(\xi)}{(2n)!} \omega_n^2(x)$$

对上式两边乘 $\rho(x)$, 并从 a 到 b 积分得

$$\int_a^b \rho(x) f(x) dx = \int_a^b \rho(x) H_{2n-1}(x) dx + \frac{1}{(2n)!} \int_a^b \rho(x) f^{(2n)}(\xi) \omega_n^2(x) dx$$

因为 Gauss 求积公式对 $2n-1$ 次多项式精确成立, 即有

$$\int_a^b \rho(x) H_{2n-1}(x) dx = \sum_{k=1}^n A_k H_{2n-1}(x_k) = \sum_{k=1}^n A_k f(x_k)$$

从而得到

$$\int_a^b \rho(x)f(x)dx - \sum_{k=1}^n A_k f(x_k) = \frac{1}{(2n)!} \int_a^b f^{(2n)}(\xi) \rho(x) \omega_n^2(x) dx$$

由于 $\rho(x) \omega_n^2(x) > 0$, $f^{(2n)}(x)$ 在 $[a, b]$ 上连续, 所以由积分中值定理得

$$\frac{1}{(2n)!} \int_a^b f^{(2n)}(\xi) \rho(x) \omega_n^2(x) dx = \frac{f^{(2n)}(\eta)}{(2n)!} \int_a^b \rho(x) \omega_n^2(x) dx$$

其中 $a \leq \eta \leq b$. 于是

$$\int_a^b \rho(x)f(x)dx - \sum_{k=1}^n A_k f(x_k) = \frac{f^{(2n)}(\eta)}{(2n)!} \int_a^b \rho(x) \omega_n^2(x) dx$$

练习与思考

1. 分别使用梯形求积公式和 Simpson 求积公式计算积分 $\int_1^2 e^{\frac{1}{x}} dx$ 的近似值, 并进行误差估计.

2. 使用 Newton - Cotes 求积公式计算积分 $\int_1^9 \sqrt{x} dx$ 的近似值 (取 $n=4$).

3. 若 $f''(x) < 0$, 证明使用梯形公式计算积分 $\int_a^b f(x) dx$ 所得到的值比准确值小, 并说明其几何意义.

4. 证明求积近似公式

$$\int_{-2}^2 f(x) dx = \frac{1}{3} [8f(-1) - 4f(0) + 8f(1)]$$

具有 3 次代数精确度.

5. 分别使用复化梯形求积公式和复化 Simpson 求积公式计算积分 $\int_0^\pi \sin x dx$ 的近似值, 取 $n=8$. 与精确值比较, 它们各有多少位有效数字?

6. 计算积分 $\int_0^1 e^x dx$ 的近似值, 要求保证有 5 位有效数字. 若用复化 Simpson 求积公式计算, n 应取多少?

7. 使用自动选取步长梯形法计算积分 $\int_0^1 \frac{2}{1+t^2} dt$ 的近似值. (给定 $\varepsilon = 0.01$)

8. 使用 Romberg 求积法计算积分 $\int_1^9 \sqrt{x} dx$ 的近似值. (给定 $\varepsilon = 0.01$, 且取 $n=1$)

9. 试确定 x_1, x_2, A_1, A_2 , 使求积公式

$$\int_0^1 f(x) dx \approx A_1 f(x_1) + A_2 f(x_2)$$

成为 Gauss 求积公式.

10. 使用两点的 Gauss - Legendre 求积公式计算积分 $\int_0^\pi e^x \cos x dx$ 的近似值.

5 解线性代数方程组的直接法

在自然科学和工程技术中,许多问题的解决常常归结为线性代数方程组的求解.例如,电学中的网络问题、船体放样中三次样条求解问题、求解非线性方程组问题、常微分方程及偏微分方程数值求解问题等,都可归结为求解线性代数方程组.有关线性代数方程组解的存在惟一性及解的结构等理论问题,线性代数已作了详细讨论.本书只介绍线性代数方程组的两类求解方法:直接法和迭代法.

直接法是指经过有限步计算后求得方程组精确解的方法.本章研究的对象是 n 阶线性代数方程组:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \quad (5.1)$$

若用矩阵和向量的记号来表示,方程组(5.1)可写成

$$AX = b \quad (5.2)$$

其中, $A = (a_{ij})$ 是 $n \times n$ 系数矩阵; $X = \{x_i\}$ 是 n 维向量,为所求的解; $b = \{b_i\}$, b 称为方程组(5.1)的右端项.

若 A 是非奇异矩阵,则方程组有惟一解.记 $D = \det A$,应用 Cramer 法则可得

$$x_i = \frac{D_i}{D} \quad (i = 1, 2, \cdots, n)$$

其中, D_i 是用 b 代替 A 中第 i 列而得到的相应行列式.然而在实际中,用 Cramer 法则求解(5.1)式并不可行. n 阶行列式共有 $n!$ 项,每项有 n 个因子,所以计算一个 n 阶行列式共需 $(n-1) \cdot n!$ 次乘法,应用 Cramer 法则求解方程组需要计算 $(n+1)$ 个行列式,另外要计算 x_i ,还需 n 次除法.因此,用 Cramer 法则求解方程组(5.1)的乘除次数为

$$N = (n^2 - 1) \cdot n! + n$$

当 $n = 10$ 时, $N = 359\,251\,210$; 当 $n = 20$ 时, $N = 9.707\,3 \times 10^{20}$, 可见, Cramer 法则虽然简单,但并不能直接用来求解线性代数方程组.

5.1 高斯消去法

5.1.1 顺序高斯消去法

高斯消去法是一个古老的方法,但实践证明它仍是目前计算机上一种常用的有效方法,其基本思想是通过初等变换将方程组(5.1)转化为一个等价的三角形方程组

$$\begin{cases} b_{11}x_1 + b_{12}x_2 + \cdots + b_{1n}x_n = g_1 \\ b_{22}x_2 + \cdots + b_{2n}x_n = g_2 \\ \vdots \\ b_{nn}x_n = g_n \end{cases} \quad (5.3)$$

这个过程称为消元,得到方程组(5.3)后,就可逐个求出 $x_n, x_{n-1}, \cdots, x_1$. 这个过程称为回代,具体过程如下:

从方程组(5.3)中的最后一个方程直接求出 x_n

$$x_n = \frac{g_n}{b_{nn}}$$

代入方程组(5.3)中倒数第二个方程,便有

$$x_{n-1} = \frac{g_{n-1} - b_{n-1,n}x_n}{b_{n-1,n-1}}$$

一般地,有

$$x_i = \frac{g_i - \sum_{j=i+1}^n b_{ij}x_j}{b_{ii}} \quad (i = n, n-1, \cdots, 1)$$

5.1.1.1 顺序高斯消去法的计算过程

为了统一符号,把方程组(5.1)改写成下面形式

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n = b_1^{(1)} \\ a_{21}^{(1)}x_1 + a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \vdots \\ a_{n1}^{(1)}x_1 + a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n = b_n^{(1)} \end{cases} \quad (5.4)$$

用矩阵表示即为

$$A^{(1)}X = b^{(1)} \quad (5.5)$$

其中

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

设 $a_{11}^{(1)} \neq 0$, 令 $m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}$, 将方程组 (5.4) 第 1 行乘 $-m_{i1}$ 后加到第 i 行 ($i=2, 3,$

\cdots, n), 则方程组 (5.4) 化为

$$\begin{cases} a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \cdots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ a_{22}^{(2)} x_2 + \cdots + a_{2n}^{(2)} x_n = b_2^{(2)} \\ \vdots \\ a_{n2}^{(2)} x_2 + \cdots + a_{nn}^{(2)} x_n = b_n^{(2)} \end{cases} \quad (5.6)$$

其中

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

用矩阵符号表示为

$$\mathbf{A}^{(2)} \mathbf{X} = \mathbf{b}^{(2)} \quad (5.7)$$

其中

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

类似地, 若设 $a_{22}^{(2)} \neq 0$, 令 $m_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}$, 将方程组 (5.6) 中第 2 个方程乘 $-m_{i2}$ 后加

到第 i 行 ($i=3, 4, \cdots, n$), 则方程组 (5.6) 化为

$$\begin{cases} a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + \cdots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ a_{22}^{(2)} x_2 + a_{23}^{(2)} x_3 + \cdots + a_{2n}^{(2)} x_n = b_2^{(2)} \\ a_{33}^{(3)} x_3 + \cdots + a_{3n}^{(3)} x_n = b_3^{(3)} \\ \vdots \\ a_{n3}^{(3)} x_3 + \cdots + a_{nn}^{(3)} x_n = b_n^{(3)} \end{cases} \quad (5.8)$$

其中

$$\begin{cases} a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2} a_{2j}^{(2)} & (i, j=3, 4, \cdots, n) \\ b_i^{(3)} = b_i^{(2)} - m_{i2} b_2^{(2)} & (i=3, 4, \cdots, n) \end{cases}$$

用矩阵符号表示为

$$\mathbf{A}^{(3)} \mathbf{X} = \mathbf{b}^{(3)} \quad (5.9)$$

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

若 $a_{33}^{(3)} \neq 0$, 上述过程可以继续下去, 重复 $n-1$ 次后, 可得到等价的三角形方程组

$$\begin{cases} a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \cdots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ a_{22}^{(2)} x_2 + \cdots + a_{2n}^{(2)} x_n = b_2^{(2)} \\ \vdots \\ a_{nn}^{(n)} x_n = b_n^{(n)} \end{cases} \quad (5.10)$$

用矩阵表示为

$$\mathbf{A}^{(n)} \mathbf{X} = \mathbf{b}^{(n)} \quad (5.11)$$

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

消元过程中元素的计算公式可以归纳为

$$\begin{cases} a_{ij}^{(k+1)} = a_{ij}^{(k)} & b_i^{(k+1)} = b_i^{(k)} & (1 \leq i \leq k, i \leq j \leq n) \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)} & & (i, j = k+1, \cdots, n) \\ b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)} & & (i = k+1, \cdots, n) \\ a_{ij}^{(k+1)} = 0 & & (1 \leq j \leq k < i \leq n) \end{cases}$$

从上述消元过程可以看到, 顺序高斯消去法能进行下去的充要条件是 \mathbf{A} 矩阵的所有顺序主子式不为零.

5.1.1.2 顺序高斯消去法求解算法

①输入系数矩阵 \mathbf{A} 、右端项 \mathbf{b} 及 ε .

②消元:

对 $k=1, 2, \cdots, n-1$ 循环

若 $|a_{kk}| \leq \varepsilon$, 则打印“求解失败”, 停机; 否则

for $i=k+1$ to n do

$$T = \frac{a_{ik}}{a_{kk}}$$

$$b_i = b_i - T \times b_k$$

for $j = k + 1$ to n do

$$a_{ij} = a_{ij} - T \times a_{kj}$$

③回代:

若 $|a_{nn}| \leq \varepsilon$, 则打印“求解失败”, 停机; 否则

$$x_n = \frac{b_n}{a_{nn}}$$

for $i = n - 1$ downto 1 do

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

④打印 $x_i, i = 1, 2, \dots, n$

上述算法中, 为了不增加存储量, 消元过程直接在系数矩阵 A 中进行. ε 的引用主要是为了控制溢出.

5.1.1.3 高斯消去法的计算量

(1) 消元过程的计算量

第 1 次消元:

计算 $m_{i1} (i = 2, 3, \dots, n)$ 需要 $n - 1$ 次除法;

计算 $a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)} (i, j = 2, 3, \dots, n)$ 需要 $(n - 1)^2$ 次乘法;

计算 $b_i^{(2)} = b_i^{(1)} - m_{i1}b_1^{(1)} (i = 2, 3, \dots, n)$ 需要 $n - 1$ 次乘法.

因此经过一次消元后乘除次数为 $n^2 - 1$, 加减法次数为 $(n - 1)^2 + (n - 1) = n(n - 1)$.

第 2 次消元:

第 2 次消元等价于一个 $n - 1$ 阶方程组的一次消元, 所需乘除法次数为 $(n - 1)^2 - 1$, 加减法次数为 $(n - 1)(n - 2)$. 因此消元过程的计算量为:

乘除法次数

$$\sum_{i=0}^{n-1} [(n - i)^2 - 1] = \sum_{i=1}^n i^2 - n = \frac{2n^3 + 3n^2 - 5n}{6}$$

加减法次数

$$\sum_{i=1}^n i(i - 1) = \sum_{i=1}^n i^2 - \sum_{i=1}^n i = \frac{n^3 - n}{3}$$

(2) 回代过程的计算量

乘除法次数

$$1 + \sum_{i=1}^{n-1} [(n - i) + 1] = \frac{n^2 + n}{2}$$

加减法次数

$$\sum_{i=1}^{n-1} [(n-i-1) + 1] = \frac{n^2 - n}{2}$$

从而顺序高斯消去法的计算量为:

乘除法次数

$$\frac{2n^3 + 3n^2 - 5n}{6} + \frac{n^2 + n}{2} = \frac{n^3}{3} + n^2 - \frac{n}{3}$$

加减法次数

$$\frac{n^3 - n}{3} + \frac{n^2 - n}{2} = \frac{n^3}{3} + \frac{n^2}{2} - \frac{5}{6}n$$

5.1.1.4 顺序高斯消去法的矩阵解释

在上述消元过程中用到了矩阵初等变换,即将某一行减去另一行的常数倍. 在后面的列主元和全主元高斯消去法中还会用到矩阵的行交换与列交换. 现对相应初等变换的矩阵表示说明如下:

(1) 交换矩阵的两行、两列的变换

利用单位坐标向量 $\mathbf{e}_i (i=1, 2, \dots, n)$, 单位矩阵 \mathbf{I} 可以表示成

$$\mathbf{I} = [\mathbf{e}_1, \dots, \mathbf{e}_n]$$

交换单位矩阵 \mathbf{I} 的 i, j 两行所得矩阵记为 \mathbf{P}_{ij}

$$\mathbf{P}_{ij} = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & 0 & & & 1 & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & 1 & & & & 0 & \\ & & & & & & & & 1 \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{bmatrix} \begin{matrix} i \\ j \end{matrix}$$

用 \mathbf{P}_{ij} 左乘矩阵 \mathbf{A} , 其作用就是交换 \mathbf{A} 的第 i 行和第 j 行; 用 \mathbf{P}_{ij} 右乘矩阵 \mathbf{A} , 其作用就是交换 \mathbf{A} 的第 i 列和第 j 列.

(2) 矩阵的第 $i (i=2, 3, \dots, n)$ 行减去 m_{i1} 乘第 1 行的变换

这种变换相当于用矩阵

$$\mathbf{M}_1 = \begin{bmatrix} 1 & & & \\ -m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ -m_{n1} & 0 & \cdots & 1 \end{bmatrix}$$

左乘矩阵 A .

高斯消去法的消元过程,从矩阵运算的角度来看,从(5.5)式化为(5.7)式相当于系数矩阵和左端项左乘一个初等变换矩阵 M_1 ,即

$$M_1 A^{(1)} X = M_1 b^{(1)}$$

$A^{(2)} = M_1 A^{(1)}, b^{(2)} = M_1 b^{(1)}$,类似地,记

$$M_i = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -m_{i+1,i} & 1 & & \\ & \vdots & & & \ddots & \\ & -m_{n,i} & & & & 1 \end{bmatrix}$$

则 $A^{(i+1)} = M_i A^{(i)}, b^{(i+1)} = M_i b^{(i)}$.

所以, $n-1$ 步以后得到的上三角矩阵 $A^{(n)}$ 、右端项 $b^{(n)}$ 和 $A^{(1)}, b^{(1)}$ 有下列关系式

$$\begin{cases} A^{(n)} = M_{n-1} \cdot M_{n-2} \cdot \cdots \cdot M_1 A^{(1)} \\ b^{(n)} = M_{n-1} \cdot M_{n-2} \cdot \cdots \cdot M_1 b^{(1)} \end{cases}$$

5.1.2 列主元高斯消去法

主元高斯消去法根据主元选取范围的不同,又分为列主元消去法和全主元消去法. 主元高斯消去法是为了控制舍入误差而提出来的一种算法. 在顺序高斯消去法的消元过程中,若出现 $a_{kk}^{(k)} = 0$,则消元无法进行;即使 $a_{kk}^{(k)} \neq 0$,但其值很小,用它作为除数也会导致其他元素量级的巨大增长和舍入误差的扩散,引起解的失真. 先看下面的例子.

例 5-1 解方程组

$$\begin{cases} 0.003\,000x_1 + 59.14x_2 = 59.17 \\ 5.291x_1 - 6.130x_2 = 46.78 \end{cases}$$

该方程组精确解为 $x_1^* = 10.00, x_2^* = 1.000$.

解 利用顺序高斯消去法取 4 位有效数字进行运算得

$$m_{21} = \frac{5.291}{0.003\,000} = 1\,763.666 \approx 1\,764$$

将第 1 个方程的 $-1\,764$ 倍加到第 2 个方程,得

$$\begin{cases} 0.003\,000x_1 + 59.14x_2 = 59.17 \\ -104\,300x_2 = -104\,400 \end{cases}$$

通过回代得 $x_2 = 1.001, x_1 = -10.00$. 这与准确解相差很大. 这是由于用 $0.003\,000$

作除数引起舍入误差造成的. 同样对上述例题, 可以交换一下方程的位置(由线性代数理论可知方程组的解并不发生改变), 再利用顺序高斯消去法求解.

例 5-2 解方程组

$$\begin{cases} 5.291x_1 - 6.130x_2 = 46.78 \\ 0.003\,000x_1 + 59.14x_2 = 59.17 \end{cases}$$

解 利用顺序高斯消去法取 4 位有效数字进行运算得

$$m_{21} = \frac{0.003\,000}{5.291} = 0.000\,567\,0$$

将第 1 个方程的 $-0.000\,567\,0$ 倍加到第 2 个方程, 得

$$\begin{cases} 5.291x_1 - 6.130x_2 = 46.78 \\ 59.14x_2 = 59.14 \end{cases}$$

通过回代, 得 $x_1 = 10.00, x_2 = 1.000$.

小主元是不稳定的根源. 因此, 为使这种不稳定现象发生的可能性减至最少, 必须在每次消元前选择一个相对较大的系数作为主元.

5.1.2.1 列主元高斯消去法计算过程

第一步: 在系数矩阵 $\mathbf{A}^{(1)}$ 的第 1 列元素中选出绝对值最大的元素 $a_{i_1,1}^{(1)}$, 称之为第 1 列的主元, 即 $|a_{i_1,1}^{(1)}| = \max_{1 \leq i \leq n} |a_{i1}^{(1)}|$. 如果 $i_1 \neq 1$, 则交换第 1 行和第 i_1 行元素及相应的右端项, 交换完后再进行消元得

$$\mathbf{A}^{(2)} \mathbf{X} = \mathbf{b}^{(2)} \quad (5.12)$$

第二步: 在系数矩阵 $\mathbf{A}^{(2)}$ 第 2 列 $a_{22}^{(2)}$ 以下 $n-1$ 个元素中选主元 $a_{i_2,2}^{(2)}$, 即 $|a_{i_2,2}^{(2)}| = \max_{2 \leq i \leq n} |a_{i2}^{(2)}|$. 如果 $i_2 \neq 2$, 则对调矩阵 (5.12) 中的第 2 行与第 i_2 行元素及相应的右端项, 交换后再进行消元得

$$\mathbf{A}^{(3)} \mathbf{X} = \mathbf{b}^{(3)}$$

一般地, 第 k 步求 i_k 使 $|a_{i_k,k}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$. 如果 $i_k \neq k$, 则交换第 k 行与第 i_k 行及相应的右端项, 经消元后得

$$\mathbf{A}^{(k+1)} \mathbf{X} = \mathbf{b}^{(k+1)}$$

列主元消去法除了每步需要按列选主元并可能进行矩阵行交换外, 其消元过程与顺序高斯消去法的过程一样, 归纳以上过程有以下算法.

5.1.2.2 列主元高斯消去算法

①输入 A 、 b 及 ε .

②选主元及消元:

for $k = 1$ to $n - 1$ do

选主元: $T = |a_{i_k, k}| = \max_{k \leq i \leq n} |a_{ik}|$

若 $T < \varepsilon$, 则打印"求解失败", 停机; 否则

若 $i_k \neq k$, 则交换 A 的第 i_k 行与第 k 行, 交换 b_{i_k} 与 b_k ; 否则

消元: for $i = k + 1$ to n do

$$T = \frac{a_{ik}}{a_{kk}}$$

$$b_i = b_i - T \times b_k$$

for $j = k + 1$ to n do

$$a_{ij} = a_{ij} - T \times a_{ik}$$

③回代:

若 $|a_{nn}| \leq \varepsilon$, 则打印"求解失败", 停机; 否则

$$x_n = \frac{b_n}{a_{nn}}$$

for $i = n - 1$ downto 1 do

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

④打印 $x_i (i = 1, 2, \dots, n)$

5.1.2.3 列主元高斯消去法的计算量

列主元高斯消去法的乘除及加减次数与顺序高斯消去法相同, 增加了选主元时 $n(n-1)$ 次比较以及交换方程次序所需时间.

5.1.2.4 列主元高斯消去法的矩阵解释

记 $|a_{i_1, 1}^{(1)}| = \max_{1 \leq i \leq n} |a_{i1}^{(1)}|$, 则说明第 1 个主元在第 i_1 行, 交换第 1 行与第 i_1 行(相当于用 P_{1, i_1} 左乘矩阵 $A^{(1)}$), 然后再进行消元, 用矩阵表示即为

$$A^{(2)} = M_1 P_{1, i_1} A^{(1)} \quad b^{(2)} = M_1 P_{1, i_1} b^{(1)}$$

同样道理, 有

$$A^{(n)} = M_{n-1} P_{n-1, i_{n-1}} A^{(n-1)} \quad b^{(n)} = M_{n-1} P_{n-1, i_{n-1}} b^{(n-1)}$$

由此得到等价的三角形方程组为 $A^{(n)} X = b^{(n)}$. 这时

$$\begin{cases} A^{(n)} = M_{n-1} P_{n-1, i_{n-1}} \cdots M_1 P_{1, i_1} A^{(1)} \\ b^{(n)} = M_{n-1} P_{n-1, i_{n-1}} \cdots M_1 P_{1, i_1} b^{(1)} \end{cases}$$

例 5-3 解方程组

$$\begin{cases} 12x_1 - 3x_2 + 3x_3 = 15 \\ -18x_1 + 3x_2 - x_3 = -15 \\ x_1 + x_2 + x_3 = 6 \end{cases}$$

(它的准确解为 $x_1 = 1, x_2 = 2, x_3 = 3$.)

解 第 1 步: 首先选取第 1 列绝对值最大的作主元. 对于本方程组, 绝对值最大的主元为 -18 , 交换第 1 行和第 2 行得

$$\begin{cases} -18x_1 + 3x_2 - x_3 = -15 \\ 12x_1 - 3x_2 + 3x_3 = 15 \\ x_1 + x_2 + x_3 = 6 \end{cases}$$

将第 1 个方程的 $\frac{2}{3}$ 倍加到第 2 个方程, 第 1 个方程的 $\frac{1}{18}$ 倍加到第 3 个方程得

$$\begin{cases} -18x_1 + 3x_2 - x_3 = -15 \\ -x_2 + \frac{7}{3}x_3 = 5 \\ \frac{7}{6}x_2 + \frac{17}{18}x_3 = \frac{31}{6} \end{cases}$$

第 2 步: 选取第 2 列 (第 2 个元素以下) 绝对值最大的主元, 对于本方程组, 绝对值最大的主元为 $\frac{7}{6}$, 交换第 2 行和第 3 行, 消元后得

$$\begin{cases} -18x_1 + 3x_2 - x_3 = -15 \\ \frac{7}{6}x_2 + \frac{17}{18}x_3 = \frac{31}{6} \\ \frac{22}{7}x_3 = \frac{66}{7} \end{cases}$$

把上面方程进行回代, 就可逐步解出 $x_1 = 1, x_2 = 2, x_3 = 3$.

5.1.3 全主元高斯消去法

全主元高斯消去法与列主元高斯消去法类似, 所不同的是选取主元的范围不同. 全主元高斯消去法在整个系数矩阵中找绝对值最大的元素作为主元, 以控制舍入误差的增长, 将舍入误差控制在一个最小的范围. 但找主元和交换行列次序要花费大量的机器时间. 因此, 有时不用全主元高斯消去法而用列主元高斯消去法.

5.1.3.1 全主元高斯消去法计算过程

第 1 步: 求 i_1, j_1 使

$$|a_{i_1, j_1}^{(1)}| = \max_{\substack{1 \leq i \leq n \\ 1 \leq j \leq n}} |a_{ij}^{(1)}|$$

如果 $i_1 \neq 1$, 则交换第 1 行和第 i_1 行及相应的右端项 (\mathbf{P}_{1,i_1} 左乘 $\mathbf{A}^{(1)}$); 若 $j_1 \neq 1$, 则交换第 1 列和第 j_1 列 (\mathbf{P}_{1,j_1} 右乘 $\mathbf{A}^{(1)}$), 并交换 x_1 与 x_{j_1} . 交换完后再进行消元得

$$\mathbf{M}_1 \mathbf{P}_{1,i_1} \mathbf{A}^{(1)} \mathbf{P}_{1,j_1} \mathbf{P}_{1,j_1}^{-1} \mathbf{X} = \mathbf{M}_1 \mathbf{P}_{1,i_1} \mathbf{b}^{(1)} \quad (5.13)$$

记 $\mathbf{A}^{(2)} = \mathbf{M}_1 \mathbf{P}_{1,i_1} \mathbf{A}^{(1)} \mathbf{P}_{1,j_1}, \mathbf{b}^{(2)} = \mathbf{M}_1 \mathbf{P}_{1,i_1} \mathbf{b}^{(1)}$. 上式变为

$$\mathbf{A}^{(2)} \mathbf{P}_{1,j_1}^{-1} \mathbf{X} = \mathbf{b}^{(2)}$$

一般地, 第 k 步 ($k = 1, 2, \dots, n-1$), 求 i_k, j_k 使

$$|a_{i_k, j_k}^{(k)}| = \max_{\substack{k \leq i \leq n \\ k \leq j \leq n}} |a_{ij}^{(k)}|$$

如果 $i_k \neq k$, 则交换 $\mathbf{A}^{(k)}$ 的第 k 行与第 i_k 行 (\mathbf{P}_{k,i_k} 左乘 $\mathbf{A}^{(k)}$) 及相应的右端项; 若 $j_k \neq k$, 则交换 $\mathbf{A}^{(k)}$ 的第 k 列和第 j_k 列 (\mathbf{P}_{k,j_k} 右乘 $\mathbf{A}^{(k)}$), 并交换 x_k 与 x_{j_k} . 交换完后进行消元得

$$\mathbf{M}_k \mathbf{P}_{k,i_k} \mathbf{A}^{(k)} \mathbf{P}_{k,j_k} \mathbf{P}_{k,j_k}^{-1} \cdots \mathbf{P}_{1,j_1}^{-1} \mathbf{X} = \mathbf{M}_k \mathbf{P}_{k,i_k} \mathbf{b}^{(k)}$$

记 $\mathbf{A}^{(k+1)} = \mathbf{M}_k \mathbf{P}_{k,i_k} \mathbf{A}^{(k)} \mathbf{P}_{k,j_k}, \mathbf{b}^{(k+1)} = \mathbf{M}_k \mathbf{P}_{k,i_k} \mathbf{b}^{(k)}$. 上式变为

$$\mathbf{A}^{(k+1)} \mathbf{P}_{k,j_k}^{-1} \cdots \mathbf{P}_{1,j_1}^{-1} \mathbf{X} = \mathbf{b}^{(k+1)}$$

归纳上述过程有以下算法:

5.1.3.2 全主元高斯消去法算法

①输入 A 、 b 及 ε .

②for $i=1$ to n do $d_i=i$ //记录未知量位置的变化//

③选主元消元:

for $k=1$ to $n-1$ do

求 i_k, j_k 使 $T = |a_{i_k, j_k}^{(k)}| = \max_{\substack{k \leq i \leq n \\ k \leq j \leq n}} |a_{ij}^{(k)}|$

若 $T < \varepsilon$, 则打印“求解失败”, 停机; 否则

若 $i_k \neq k$, 则交换 A 的第 i_k 行与第 k 行, 交换 b_{i_k} 与 b_k

若 $j_k \neq k$, 则交换 A 的第 j_k 列与第 k 列, 交换 d_{j_k} 与 d_k

for $i=k+1$ to n do

$$T = \frac{a_{ik}}{a_{kk}}$$

$$b_i = b_i - T \times b_k$$

for $j=k+1$ to n do

$$a_{ij} = a_{ij} - T \times a_{kj}$$

④回代:

$$Z_n = \frac{b_n}{a_{nn}}$$

for $i=n-1$ downto 1 do

$$Z_i = \frac{b_i - \sum_{j=i+1}^n a_{ij} Z_j}{a_{ii}}$$

⑤for $j=1$ to n do

$$x_{d_j} = Z_j$$

⑥打印 $x_i, i=1, 2, \dots, n$

5.1.3.3 全主元高斯消去法的计算量

全主元高斯消去法乘除及加减次数与顺序高斯消去法相同, 增加了选主元时

$$\sum_{i=1}^n (i^2 - 1) = \frac{n^3}{6} - \frac{n}{6}$$

次比较以及交换方程次序所需的时间。

5.1.3.4 全主元高斯消去法的矩阵解释

记 $|a_{i_1, j_1}^{(1)}| = \max_{\substack{1 \leq i \leq n \\ 1 \leq j \leq n}} |a_{ij}^{(1)}|$, 由上述过程知, 主元在第 i_1 行、第 j_1 列, 交换第 1 行与

第 i_1 行、第 1 列与第 j_1 列, 然后再消元即可得到 $A^{(2)}$, 用矩阵表示即为

$$\begin{cases} A^{(2)} = M_1 P_{1,i_1} A^{(1)} P_{1,j_1} \\ b^{(2)} = M_1 P_{1,i_1} b^{(1)} \end{cases}$$

同理,得

$$\begin{cases} A^{(n)} = M_{n-1} P_{n-1,i_{n-1}} A^{(n-1)} P_{n-1,j_{n-1}} \\ b^{(n)} = M_{n-1} P_{n-1,i_{n-1}} b^{(n-1)} \end{cases}$$

注意到 $A^{(1)} X = b^{(1)}$, 两边同乘 $M_{n-1} P_{n-1,i_{n-1}} \cdots M_1 P_{1,i_1}$, 得

$$M_{n-1} P_{n-1,i_{n-1}} \cdots M_1 P_{1,i_1} A^{(1)} X = b^{(n)} \quad (5.14)$$

由于 $A^{(n)} = M_{n-1} P_{n-1,i_{n-1}} \cdots M_1 P_{1,i_1} A^{(1)} P_{1,j_1} \cdots P_{n-1,j_{n-1}}$, 由 (5.14) 式知

$$A^{(n)} P_{n-1,j_{n-1}}^{-1} \cdots P_{1,j_1}^{-1} X = b^{(n)} \quad (5.15)$$

记

$$Z = P_{n-1,j_{n-1}}^{-1} \cdots P_{1,j_1}^{-1} X \quad (5.16)$$

则

$$A^{(n)} Z = b^{(n)} \quad (5.17)$$

通过 (5.17) 式可求得 Z . 由 (5.16) 式最终求得

$$X = P_{1,j_1} P_{2,j_2} \cdots P_{n-1,j_{n-1}} Z$$

5.2 LU 分解法

前面介绍的顺序高斯消去法可以把方程组 $A^{(1)} X = b^{(1)}$ 等价转化为一个上三角形方程组 $A^{(n)} X = b^{(n)}$. 而从消元过程可以看到, 从 $A^{(1)}$ 转化到 $A^{(n)}$ 的过程实际是进行若干次初等变换的过程, 由顺序高斯消去法的矩阵解释得到

$$A^{(n)} = M_{n-1} M_{n-2} \cdots M_1 A^{(1)}$$

从而

$$A^{(1)} = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} A^{(n)}$$

注意到 M_i^{-1} ($i=1, 2, \cdots, n-1$) 均为下三角矩阵且对角线元素为 1, 因此, $M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1}$ 为对角线元素为 1 的下三角矩阵. 记

$$L = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1}$$

$$U = A^{(n)}$$

由顺序高斯消去法看到, 只要方程

$$AX = b$$

的系数矩阵 A 的所有顺序主子式不为零, 则 A 一定可以分解成

$$A = LU$$

其中

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

L 为对角线元素为 1 的单位下三角矩阵; U 为上三角矩阵.

定理 5.1 如果 n 阶矩阵 A 的所有顺序主子式不为零, 则 A 有惟一的 LU 分解.

证明 设

$$A = LU = L_1 U_1$$

其中, L, L_1 为单位下三角矩阵; U, U_1 为上三角矩阵. 因为 A 非奇异, 故 $L_1^{-1} U_1^{-1}$ 存在. 于是有

$$L^{-1} L_1 = U U_1^{-1}$$

上式左端为单位下三角矩阵, 右端为上三角矩阵, 从而上式两边均为单位矩阵, 从而证得 $U = U_1, L = L_1$. A 的三角形矩阵分解的惟一性获证.

如果能实现这种分解, 则求解方程组 $AX = b$ 就相当简单. 注意到 $A = LU$, 因此 $LUX = b$. 记 $UX = Y$, 则

$$\begin{cases} LY = b \\ UX = Y \end{cases}$$

通过 $LY = b$, 得

$$\begin{cases} y_1 = b_1 \\ y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j \quad (i = 2, \cdots, n) \end{cases}$$

通过 $UX = Y$, 最终求得

$$\begin{cases} x_n = \frac{y_n}{u_{nn}} \\ x_i = \left[y_i - \sum_{j=i+1}^n u_{ij} x_j \right] / u_{ii} \quad (i = n-1, \cdots, 1) \end{cases}$$

5.2.1 直接 LU 分解法

下面详细讨论矩阵 A 的 LU 分解. 设 $A = LU$, 即

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

两个矩阵相等, 则其对应元素相等. 利用矩阵乘法, 由矩阵的第 1 行对应相等得

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

由矩阵的第 1 列对应相等得

$$l_{i1} = \frac{a_{i1}}{u_{11}} \quad (i = 2, 3, \dots, n)$$

这就求出了 U 矩阵的第 1 行和 L 矩阵的第 1 列元素.

一般地, 设 U 矩阵的前 $k-1$ 行和 L 矩阵的前 $k-1$ 列已经求出, 则由

$$a_{kj} = \sum_{m=1}^{k-1} l_{km} u_{mj} + u_{kj}$$

得

$$u_{kj} = a_{kj} - \sum_{m=1}^{k-1} l_{km} u_{mj} \quad (j = k, k+1, \dots, n)$$

又由

$$a_{ik} = \sum_{m=1}^{k-1} l_{im} u_{mk} + l_{ik} u_{kk}$$

得

$$l_{ik} = \frac{a_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk}}{u_{kk}} \quad (i = k+1, \dots, n)$$

综上所述, A 的 LU 分解公式如下

$$\begin{cases} u_{kj} = a_{kj} - \sum_{m=1}^{k-1} l_{km} u_{mj} & (j = k, k+1, \dots, n) \\ l_{ik} = \frac{a_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk}}{u_{kk}} & (i = k+1, \dots, n) \end{cases}$$

其中, $k = 1, 2, \dots, n$.

上述计算公式有如下特点: U 的元素按行逐行求, L 的元素按列逐列求; 先求 U 的第 k 行元素, 然后求 L 的第 k 列元素, U 和 L 的元素一行一列交叉进行, 如图 5-1 所示.

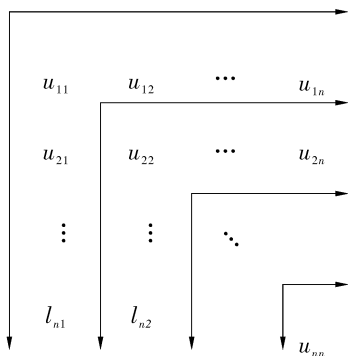


图 5-1

5.2.1.1 直接 LU 分解法算法

①输入 \mathbf{A}, \mathbf{b} 及 ε .②对 $k=1, 2, \dots, n$ 循环对 $j=k, k+1, \dots, n$ 循环

$$u_{kj} = a_{kj} - \sum_{m=1}^{k-1} l_{km} u_{mj}$$

如果 $|u_{kk}| < \varepsilon$, 则打印“求解失败”, 停机; 否则对 $i=k+1, \dots, n$, 计算

$$l_{ik} = \frac{a_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk}}{u_{kk}}$$

③求解 $\mathbf{LY} = \mathbf{b}$

$$y_1 = b_1$$

对 $i=2, 3, \dots, k$, 计算

$$y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j$$

④求解 $\mathbf{UX} = \mathbf{Y}$

$$x_n = y_n / u_{nn}$$

对 $i=n-1, n-2, \dots, 1$, 计算

$$x_i = \frac{y_i - \sum_{j=i+1}^n u_{ij} x_j}{u_{ii}}$$

⑤打印 $x_i (i=1, 2, \dots, n)$

每次计算结果 l_{ik} 和 u_{kj} 仍可存放在矩阵 \mathbf{A} 的相应元素 a_{ik} 和 a_{kj} 所占的单元内, 不必占用新的单元, 只要将上述算法中出现的 l_{ik} 和 u_{kj} 相应变换为 a_{ik} 和 a_{kj} 即可。

5.2.1.2 直接 LU 分解法的计算量(乘除次数)

LU 分解计算 U 的计算量:

$$\sum_{i=1}^n (n-i) \cdot i = \frac{n^3 - n}{6}$$

计算 L 的计算量:

$$\sum_{i=1}^{n-1} (n-i) \cdot i = \frac{n^3 - n}{6}$$

求解 $LY = b$ 的计算量:

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

求解 $UX = Y$ 的计算量:

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

总的计算量为:

$$\frac{n^3 + 3n^2 - n}{3}$$

例 5-4 用 LU 分解法求解方程组

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

解 先把系数矩阵进行 LU 分解得

$$L = \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ -1 & 0.5 & 1 & \end{bmatrix} \quad U = \begin{bmatrix} 1 & 2 & 1 & \\ & -2 & 1 & \\ & & 0.5 & \end{bmatrix}$$

求解 $LY = b$:

$$\begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ -1 & 0.5 & 1 & \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 2 \end{bmatrix}$$

得 $y_1 = 0, y_2 = 3, y_3 = 0.5$.

求解 $UX = Y$:

$$\begin{bmatrix} 1 & 2 & 1 & \\ & -2 & 1 & \\ & & 0.5 & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 0.5 \end{bmatrix}$$

解得 $x_1 = 1, x_2 = -1, x_3 = 1$. 故 $AX = b$ 的解为 $X = (1, -1, 1)^T$.

用直接 LU 分解法求解方程组所需要的计算量仍为 $\frac{n^3 + 3n^2 - n}{3}$, 和高斯消去法

乘除法次数相同. 但 LU 分解法把对系数矩阵的计算和对右端项的计算分开了, 这就使得求解系数矩阵相同而右端项不同的一系列方程组变得特别方便.

5.2.2 列主元 LU 分解法

直接进行 LU 分解时, u_{kk} 可能为零, 或者绝对值接近零, 这样会出现溢出, 或者因以一个小的数作除数, 引起舍入误差的积累, 最后引起解的失真. 为了避免这些问题, 如果 \mathbf{A} 非奇异, 可以采用列主元的方法, 叫做列主元 LU 分解.

假若第 $k-1$ 步的分解已完成, 在进行第 k 步分解时, 为避免出现小的 u_{kk} 作除数, 计算

$$S_i = a_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk}$$

选取行号 i_k , 使

$$|S_{i_k}| = \max_{k \leq i \leq n} |S_i|$$

若 $i_k \neq k$, 则对调 \mathbf{A} 的第 k 行与第 i_k 行, 再求 $u_{kj} (j = k, \dots, n)$ 和 $l_{ik} (i = k+1, \dots, n)$, 从而完成了 \mathbf{U} 的第 k 行和 \mathbf{L} 的第 k 列的计算. 算法如下.

列主元 LU 分解算法

①输入 \mathbf{A}, \mathbf{b} 及 ε

②对 $k=1, 2, \dots, n$ 循环

计算 $S_i = a_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk} \quad (i = k, k+1, \dots, n)$

选主元 $|S_{i_k}| = \max_{k \leq i \leq n} |S_i|$, 并记录 i_k

若 $|S_{i_k}| < \varepsilon$, 则打印“求解失败”, 停机; 否则

交换 \mathbf{A} 的第 k 行与第 i_k 行元素

计算 \mathbf{U} 的第 k 行元素和 \mathbf{L} 的第 k 列元素

$$u_{kj} = a_{ij} - \sum_{m=1}^{k-1} l_{km} u_{mj} \quad (j = k, k+1, \dots, n)$$

$$l_{ik} = \frac{a_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk}}{u_{kk}} \quad (i = k+1, \dots, n)$$

③将 \mathbf{b} 向量的第 k 个元素与第 i_k 个元素交换, $k=1, 2, \dots, n-1$.

④求解 $\mathbf{LY} = \mathbf{b}, \mathbf{UX} = \mathbf{Y}$ (过程与直接 LU 分解法相同)

5.3 对称正定矩阵的平方根法和 LDL^T 分解法

当 A 是对称正定矩阵时, 存在一个实的非奇异下三角矩阵 L_1 , 使

$$A = L_1 L_1^T$$

且当限定 L_1 的对角线为正时, 这种分解惟一. 这种分解称为矩阵的 Cholesky 分解.

定理 5.2 设 A 是对称正定矩阵, 则 A 有如下分解:

$$A = LDL^T$$

其中, L 是单位下三角阵; D 为对角阵, 且这种分解是惟一的.

证明 因为 A 对称正定, 从而 A 有惟一的 LU 分解

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

因为 $\Delta_k = u_{11} \cdots u_{kk} \neq 0 (k = 1, 2, \cdots, n)$, 故上式可化为

$$A = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & & & \\ & u_{22} & & \\ & & \ddots & \\ & & & u_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12}/u_{11} & \cdots & u_{1n}/u_{11} \\ & 1 & \cdots & u_{2n}/u_{22} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

将右端三角矩阵分别记为 L, D 和 R , 因 A 对称, 有

$$A = LDR = R^T DL^T$$

或者

$$A = LDR = R^T (DL^T)$$

因为 A 的 LU 分解惟一, 故 $L = R^T, L^T = R, A = LDL^T$, 此分解显然是惟一的.

定理 5.3 n 阶对称正定矩阵 A 一定有 Cholesky 分解 $A = L_1 L_1^T$, 当限定 L_1 的对角线为正时, 矩阵的 Cholesky 分解惟一.

证明 由定理 5.2 知, 存在惟一的单位下三角矩阵 L , 对角元全为正的矩阵

$$D = \text{diag}(d_1, d_2, \cdots, d_n)$$

使

$$A = LD^{\frac{1}{2}} (LD^{\frac{1}{2}})^T$$

又记 $L_1 = LD^{\frac{1}{2}}$, 它是一个非奇异下三角阵, 一对角线元素为正, 故 A 有 Cholesky 分解

$$A = L_1 L_1^T$$

分解的惟一性显而易见。

下边推导 Cholesky 分解的计算公式

设 $A = LL^T$, 即

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

其中 $a_{ij} = a_{ji} (i, j = 1, 2, \cdots, n)$, $l_{ii} > 0 (i = 1, 2, \cdots, n)$.

第 1 步: 由矩阵乘法有 $a_{11} = l_{11}^2$, 且 $a_{i1} = l_{i1} l_{11}$, 故求得

$$l_{11} = \sqrt{a_{11}} \quad l_{i1} = \frac{a_{i1}}{l_{11}} \quad (i = 2, 3, \cdots, n)$$

一般地, 设 L 矩阵的前 $k-1$ 列元素已求出, 则

第 k 步: 由矩阵乘法得

$$\sum_{m=1}^{k-1} l_{km}^2 + l_{kk}^2 = a_{kk} \quad \sum_{m=1}^{k-1} l_{im} l_{km} + l_{ik} l_{kk} = a_{ik}$$

于是

$$\begin{cases} l_{kk} = \sqrt{a_{kk} - \sum_{m=1}^{k-1} l_{km}^2} \\ l_{ik} = \left(a_{ik} - \sum_{m=1}^{k-1} l_{im} l_{km} \right) / l_{kk} \end{cases} \quad (i = k+1, \cdots, n; k = 2, 3, \cdots, n) \quad (5.18)$$

由于分解公式(5.18)中的每一步都有开方运算, 故又称 Cholesky 方法为平方根法.

注意到 $a_{kk} = \sum_{m=1}^{k-1} l_{km}^2$, 所以 $l_{km}^2 \leq a_{kk} \leq \max_{1 \leq k \leq n} a_{kk}$.

这说明在分解过程中元素 l_{km} 的平方不会超过 A 的最大对角元, 因而舍入误差的放大受到限制. 所以, 平方根法求解对称正定方程组时可以不考虑选主元的问题.

可以证明, 若用顺序高斯消去法求解对称正定方程组 $AX = b$, 则有

$$\max_{1 \leq i, j \leq n} |a_{ij}^{(k)}| \leq \max_{1 \leq i \leq n} a_{ii} \quad (k = 1, 2, \cdots, n)$$

其中 $a_{ij}^{(k)}$ 是第 k 步顺序高斯消去法过程所得到的元素. 这说明高斯消去法求解对称正定方程组也可以不用选主元.

从运算的角度看, 平方根法是有利的, 用平方根法求解 $AX = b$ 所需的乘除次数为 $\frac{1}{6}(n^3 + 9n^2 + 2n)$, 另外还有 n 次开平方运算, 乘除次数只是高斯消去法的一半左右.

例 5-5 用平方根法解方程组

$$\begin{bmatrix} 3 & 2 & 3 \\ 2 & 2 & 0 \\ 3 & 0 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 7 \end{bmatrix}$$

解 由平方根分解法可得

$$\begin{bmatrix} 3 & 2 & 3 \\ 2 & 2 & 0 \\ 3 & 0 & 12 \end{bmatrix} = \begin{bmatrix} \sqrt{3} & & \\ \frac{2}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} & \\ \sqrt{3} & -\sqrt{6} & \sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{3} & \frac{2}{\sqrt{3}} & \sqrt{3} \\ & \frac{\sqrt{2}}{\sqrt{3}} & -\sqrt{6} \\ & & \sqrt{3} \end{bmatrix}$$

求解

$$\begin{bmatrix} \sqrt{3} & & \\ \frac{2}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} & \\ \sqrt{3} & -\sqrt{6} & \sqrt{3} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 7 \end{bmatrix}$$

得 $Y = (5/\sqrt{3}, -1/\sqrt{6}, 1/\sqrt{3})^T$.

求解

$$\begin{bmatrix} \sqrt{3} & \frac{2}{\sqrt{3}} & \sqrt{3} \\ & \frac{\sqrt{2}}{\sqrt{3}} & -\sqrt{6} \\ & & \sqrt{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

得 $X = (1, 1/2, 1/3)^T$.

平方根求解算法

①输入 A, b 及 ε .

②对 $k=1, 2, \dots, n$, 循环

$$\text{计算 } S_k = a_{kk} - \sum_{m=1}^{k-1} l_{km}^2$$

若 $S_k < \varepsilon$, 则打印“求解失败”, 停机; 否则

计算

$$l_{kk} = \sqrt{a_{kk} - \sum_{m=1}^{k-1} l_{km}^2}$$

$$l_{ik} = \left(a_{ik} - \sum_{m=1}^{k-1} l_{im} l_{km} \right) / l_{kk} \quad (i = k+1, \dots, n)$$

③求解 $LY = b$

$$y_1 = b_1 / l_{11}$$

对 $i=2,3,\cdots,n$, 计算

$$y_i = \left[b_i - \sum_{j=1}^{i-1} l_{ij}y_j \right] / l_{ii}$$

④求解 $L^T X = Y$

$$x_n = y_n / l_{nn}$$

对 $i=n-1, n-2, \cdots, 1$, 计算

$$x_i = \left[y_i - \sum_{j=i+1}^n l_{ij}x_j \right] / l_{ii}$$

⑤打印 $x_i (i=1, 2, \cdots, n)$

为避免平方根法的开方运算, 利用定理 5.2, 也可以对 A 作 LDL^T 分解, 其分解方法可借助于 LU 分解法.

从定理 5.1 可以看到, 当 A 的所有顺序主子式不为零且对称时, A 的 LU 分解存在且惟一, 并且

$$U = DL^T$$

其中, $D = \text{diag}(u_{11}, u_{22}, \cdots, u_{nn})$, $l_{ij} = u_{ji}/u_{ii}$.

利用 LU 分解法先计算出 U 的第 k 行, 由 U 的第 k 行得到 D 的第 k 个对角元素及 L 的第 k 列.

$$d_i = u_{ii} \quad l_{ik} = u_{ki}/u_{kk}$$

因此 LDL^T 分解的乘除法计算次数为

$$\frac{n^3}{6} - \frac{n}{6} + \frac{n^2 - n}{2} = \frac{1}{6}n^3 + \frac{1}{2}n^2 - \frac{2}{3}n$$

大约只是高斯消去法的一半, 与平方根法大致相同, 但避免了开方运算.

例 5-6 用 LDL^T 分解法求方程组

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

解 由 LDL^T 分解法得

$$\begin{bmatrix} 5 & -4 & 1 \\ -4 & 6 & -4 \\ 1 & -4 & 6 \end{bmatrix} = \begin{bmatrix} 1 & & \\ -\frac{4}{5} & 1 & \\ \frac{1}{5} & -\frac{8}{7} & 1 \end{bmatrix} \begin{bmatrix} 5 & & \\ & \frac{14}{5} & \\ & & \frac{15}{7} \end{bmatrix} \begin{bmatrix} 1 & -\frac{4}{5} & \frac{1}{5} \\ & 1 & -\frac{7}{8} \\ & & 1 \end{bmatrix}$$

这样就得到

$$L = \begin{bmatrix} 1 & & & \\ -\frac{4}{5} & 1 & & \\ \frac{1}{5} & -\frac{8}{7} & 1 & \\ & & & 1 \end{bmatrix} \quad D = \begin{bmatrix} 5 & & & \\ & \frac{14}{5} & & \\ & & \frac{15}{7} & \\ & & & 1 \end{bmatrix}$$

解 $LY = b$ 得

$$Y = \left[2, \frac{3}{5}, -\frac{5}{7} \right]^T.$$

解 $DZ = Y$ 得 $Z = \left[\frac{2}{5}, \frac{3}{14}, -\frac{1}{3} \right]^T$.

最后解 $L^T X = Z$ 得 $X = \left[\frac{1}{3}, -\frac{1}{6}, -\frac{1}{3} \right]^T$.

LDL^T 分解算法

①输入 A, b 及 ε

②对 $k = 1, 2, \dots, n$, 循环

a) 对 $j = k, k+1, \dots, n$, 计算

$$u_{kj} = a_{kj} - \sum_{m=1}^{k-1} l_{km} u_{mj}$$

若 $|u_{kk}| < \varepsilon$, 则打印“求解失败”, 停机; 否则

b) $d_k = u_{kk}$

c) 对 $i = k+1, \dots, n$, 计算

$$l_{ik} = \frac{u_{ki}}{u_{kk}}$$

③求解 $LY = b$

④求解 $DZ = Y$

⑤求解 $L^T X = Z$

⑥打印 $x_i (i = 1, 2, \dots, n)$

为了更好地理解算法, 引入 l_{ik} 和 u_{kj} . 实际上, l_{ik} 和 u_{kj} 仍将存放在矩阵 A 的相应元素 a_{ik} 和 a_{kj} 所占的单元内, 不必占用新的单元. 要做到这一点, 只要将上述算法中出现的 l_{ik} 和 u_{kj} 相应变为 a_{ik} 和 a_{kj} 即可.

5.4 向量与矩阵范数

用直接法求解线性方程组 $AX = b$ 时, 由于有舍入误差, 只能得到近似解. 为了进行解的误差分析和迭代法的收敛性分析(后面介绍), 首先介绍向量范数和矩阵

范数的概念.

5.4.1 向量范数

向量范数是 n 维欧几里德空间中长度概念的推广.

定义 5.1 设向量 $X \in \mathbf{R}^n$, 若与 X 对应的非负实数 $\|X\|$ 满足下面三个条件:

①非负数: $\forall X \in \mathbf{R}^n$, 有 $\|X\| \geq 0$, 且 $\|X\| = 0$ 当且仅当 $X = 0$;

②齐次性: $\forall X \in \mathbf{R}^n, \alpha \in \mathbf{R}$ 有 $\|\alpha X\| \leq |\alpha| \cdot \|X\|$;

③三角不等式 $\forall X, Y \in \mathbf{R}^n$, 有 $\|X + Y\| \leq \|X\| + \|Y\|$.

则称 $\|X\|$ 为向量 X 的范数.

常用范数有:

(1) 向量的 1 范数

$$\|X\|_1 = \sum_{i=1}^n |x_i|$$

(2) 向量的 2 范数

$$\|X\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}$$

(3) 向量的 ∞ 范数

$$\|X\|_{\infty} = \max_{1 \leq i \leq n} |x_i|$$

(4) 更一般的 p 范数为

$$\|X\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

容易验证, 上面几种范数的定义满足范数的三个条件.

定义 5.2 设 $\|\cdot\|_{\alpha}$ 和 $\|\cdot\|_{\beta}$ 是 \mathbf{R}^n 上两种范数, 如果存在与 X 无关的两个正常数 C_1 和 C_2 , 使不等式

$$C_1 \|X\|_{\alpha} \leq \|X\|_{\beta} \leq C_2 \|X\|_{\alpha} \quad (\forall X \in \mathbf{R}^n)$$

成立, 则称范数 $\|\cdot\|_{\alpha}$ 和 $\|\cdot\|_{\beta}$ 等价.

定理 5.4 有限维空间上任何两种范数均等价.

定义 5.3 设 $\{X^{(k)}\}$ 是 \mathbf{R}^n 中的向量序列, 若有向量 $X^* \in \mathbf{R}^n$, 使

$$\lim_{k \rightarrow \infty} \|X^{(k)} - X^*\| = 0$$

则称 $\{X^{(k)}\}$ 收敛于 X^* , 记为 $\lim_{k \rightarrow \infty} X^{(k)} = X^*$.

该定义形式上依赖于所选择的向量范数, 但由于 \mathbf{R}^n 中范数等价, 因此 $\{X^{(k)}\}$ 的收敛性实际上与所选择的范数无关.

定理 5.5 设 $\{X^{(k)}\}$ 是 \mathbf{R}^n 中的向量序列, 则 $\lim_{k \rightarrow \infty} \|X^{(k)} - X^*\| = 0$ 当且仅当

$$\lim_{j \rightarrow \infty} x_j^{(k)} = x_j^*$$

其中, $x_j^{(k)}$ 和 x_j^* 分别表示 $X^{(k)}$ 和 X^* 的第 j 个分量.

5.4.2 矩阵范数

定义 设矩阵 $A \in \mathbf{R}^{n \times n}$, 若 A 的非负实数 $\|A\|$ 满足下列四个条件:

①非负数: $\forall X \in \mathbf{R}^{n \times n}, \|A\| \geq 0$, 且 $\|A\| = 0$ 当且仅当 $A = 0$;

②齐次性: $\forall A \in \mathbf{R}^{n \times n}, \forall \alpha \in \mathbf{R}, \|\alpha A\| = |\alpha| \cdot \|A\|$;

③三角不等式 $\forall A, B \in \mathbf{R}^{n \times n}, \|A + B\| \leq \|A\| + \|B\|$;

④乘法不等式: $\forall A, B \in \mathbf{R}^{n \times n}, \|AB\| \leq \|A\| \cdot \|B\|$.

则称 $\|A\|$ 为矩阵 A 的范数.

同向量范数一样, 矩阵范数也有下面定理.

定理 5.6 设 $\{A^{(k)}\}$ 是 $\mathbf{R}^{n \times n}$ 中矩阵序列, A 是 $\mathbf{R}^{n \times n}$ 中矩阵, 则 $\lim_{k \rightarrow \infty} \|A^{(k)} - A\| = 0$ 当且仅当 $\lim_{k \rightarrow \infty} a_{ij}^{(k)} = a_{ij}$, 其中 $a_{ij}^{(k)}$ 和 a_{ij} 分别表示 $A^{(k)}$ 和 A 的分量.

定义 5.4 对于给定的向量范数 $\|\cdot\|$ 与矩阵范数 $\|\cdot\|$, 如果 $\forall A \in \mathbf{R}^{n \times n}$ 和 $\forall X \in \mathbf{R}^n$, 并满足

$$\|AX\| \leq \|A\| \cdot \|X\|$$

则称此矩阵范数与向量范数是相容的.

定理 5.7 设 A 为 $n \times n$ 阶矩阵, $\|\cdot\|$ 是 \mathbf{R}^n 中的向量范数, 则

$$\|A\| = \max_{\|X\|=1} \|AX\|$$

是一种矩阵范数, 称其为由向量 $\|\cdot\|$ 诱导出的矩阵范数.

现在考察由向量范数 $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$ 产生的矩阵范数.

定理 5.8 设 $A = (a_{ij}) \in \mathbf{R}^{n \times n}$, 则

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

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

$$\|A\|_2 = [\lambda_{\max}(A^H A)]^{\frac{1}{2}}$$

其中 A^H 是 A 的共轭.

证明 下面只就 1 范数给出证明. 对 $\forall X \in \mathbf{R}^n, X \neq 0$, 有

$$\begin{aligned} \|AX\|_1 &= \sum_{i=1}^n \left| \sum_{j=1}^n a_{ij} x_j \right| \leq \sum_{i=1}^n \sum_{j=1}^n |a_{ij}| \cdot |x_j| \\ &\leq \left(\max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| \right) \sum_{j=1}^n |x_j| \leq \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| \cdot \|X\|_1 \end{aligned}$$

故有

$$\|A\|_1 = \max_{X \neq 0} \frac{\|AX\|_1}{\|X\|_1} \leq \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| \quad (5.19)$$

另一方面, 设

$$\sum_{i=1}^n |a_{ij_0}| = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$$

取 $X^* = e_{j_0} = (0, \dots, 0, 1, 0, \dots, 0)^T$, 则

$$AX^* = (a_{1j_0}, a_{2j_0}, \dots, a_{nj_0})^T$$

$$\|AX^*\|_1 = \sum_{i=1}^n |a_{ij_0}| = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$$

注意到 $\|X^*\|_1 = 1$,

$$\|A\|_1 \geq \frac{\|AX^*\|_1}{\|X^*\|_1} = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| \quad (5.20)$$

综合(5.19)式、(5.20)式得

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

在这3种范数中, 1范数与 ∞ 范数的计算比较简单, 2范数要计算矩阵的特征值, 比较复杂. 但2范数有一些好的性质, 常用于理论分析.

此外, 在 $\mathbf{R}^{n \times n}$ 上的一个常用且易于计算的矩阵范数为

$$\|A\|_F = \left[\sum_{i,j=1}^n |a_{ij}|^2 \right]^{\frac{1}{2}}$$

通常称作 Frobenius 范数, 它是 \mathbf{R}^{n^2} 上的 2 范数的自然推广.

5.4.3 谱半径

定义 5.5 $A \in \mathbf{R}^{n \times n}$, A 的特征值为 λ_i ($i = 1, 2, \dots, n$), 称 $\rho(A) = \max_{1 \leq i \leq n} |\lambda_i|$ 为 A 的谱半径.

定理 5.9 A 的谱半径不超过 A 的任何一个矩阵范数, 即

$$\rho(A) \leq \|A\|$$

证明 设 λ 是 A 的任一特征值, X 是相应的特征向量, 则有

$$AX = \lambda X$$

于是有

$$|\lambda| \cdot \|X\| = \|AX\| \leq \|A\| \cdot \|X\|$$

从而 $|\lambda| \leq \|A\|$. 于是

$$\rho(A) \leq \|A\|$$

定理 5.10 设 A 是任意 $n \times n$ 阶矩阵, 由 A 的各次幂所组成的矩阵序列

$$I, A, A^2, \dots, A^k, \dots$$

收敛于零, 即 $\lim_{k \rightarrow \infty} A^k = 0$ 的充要条件是 $\rho(A) < 1$.

证明 从略. 读者可参阅冯康等编的《数值计算方法》.

5.4.4 方程右端误差对解的影响

考虑线性代数方程组

$$AX = b$$

设右端项有误差 δb 时, 相应的方程组的解为 $X + \delta X$, 即

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

将 $AX = b$ 代入得

$$A\delta X = \delta b \quad \text{或} \quad \delta X = A^{-1}\delta b$$

这里假定 A 非奇异, 从而有不等式

$$\|\delta X\| = \|A^{-1}\delta b\| \leq \|A^{-1}\| \cdot \|\delta b\|$$

而

$$\|AX\| = \|b\| \leq \|A\| \cdot \|X\|$$

所以

$$\frac{\|\delta X\|}{\|X\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\delta b\|}{\|b\|}$$

在这种情形下, 数 $\|A\| \cdot \|A^{-1}\|$ 就是误差的放大率, 即解的相对误差不超过右端项相对误差的 $\|A\| \cdot \|A^{-1}\|$ 倍, 称 $\|A\| \cdot \|A^{-1}\|$ 为矩阵 A 的条件数, 记作

$$\text{Cond}(A) = \|A\| \cdot \|A^{-1}\|$$

5.4.5 系数矩阵误差对解的影响

设系数矩阵 A 有误差 δA , $X + \delta X$ 是系数矩阵为 $A + \delta A$ 时的准确解, 即

$$(A + \delta A)(X + \delta X) = b \quad (5.21)$$

首先一个问题是, 即使矩阵 A 非奇异, 其元素有扰动 δA 后, 矩阵 $A + \delta A$ 有可能变为奇异. 但是可以想像, 若 δA 很小, 矩阵 $A + \delta A$ 仍然可以保持它的非奇异性. 为了准确给出保持这种非奇异的条件, 先证明下述引理.

引理 设矩阵 F 的范数小于 1, 即 $\|F\| < 1$, 则矩阵 $(I + F)$ 非奇异, 且

$$\|(I + F)^{-1}\| \leq \frac{1}{1 - \|F\|}$$

证明 $\forall X \in \mathbf{R}^n$, 恒有

$$\|(I + F)X\| = \|X + FX\| \leq \|X\| + \|FX\| \geq (1 - \|F\|) \cdot \|X\|$$

因为 $1 - \|F\| > 0$, 所以, 对 $X \neq 0$, $\|(I + F)X\| > 0$.

从而可知, 方程组 $(I + F)X = 0$ 只有零解, 因此矩阵 $(I + F)$ 非奇异.

又

$$\begin{aligned} 1 &= \|I\| = \|(I + F)(I + F)^{-1}\| \\ &= \|(I + F)^{-1} + F(I + F)^{-1}\| \\ &\geq \|(I + F)^{-1}\| - \|(I + F)^{-1}\| \cdot \|F\| \\ &= \|(I + F)^{-1}\| (1 - \|F\|) > 0 \end{aligned}$$

从而必有

$$\|(I+F)^{-1}\| \leq \frac{1}{1-\|F\|}$$

再来考虑矩阵 $A + \delta A$ 的非奇异性条件. 由于

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

据前述引理即可知, 若

$$\|A^{-1}\delta A\| < 1 \quad (5.22)$$

则矩阵 $A + \delta A$ 非奇异, 且

$$\|(I + A^{-1}\delta A)^{-1}\| \leq \frac{1}{1 - \|A^{-1}\delta A\|}$$

在条件(5.21)式成立时, 据(5.21)式得

$$\begin{aligned} \delta X &= (A + \delta A)^{-1} - A^{-1} \mathbf{b} = (A + \delta A)^{-1} [A - (A + \delta A)] A^{-1} \mathbf{b} \\ &= (A + \delta A)^{-1} (-\delta A) A^{-1} \mathbf{b} = (A + \delta A)^{-1} (-\delta A) X \end{aligned}$$

所以

$$\begin{aligned} \frac{\|\delta X\|}{\|X\|} &\leq \|(A + \delta A)^{-1}\delta A\| = \|(I + A^{-1}\delta A)^{-1}A^{-1}\delta A\| \\ &\leq \|(I + A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\delta A\| \leq \frac{\|A^{-1}\delta A\|}{1 - \|A^{-1}\delta A\|} \end{aligned}$$

由于

$$\begin{aligned} \frac{\|A^{-1}\delta A\|}{1 - \|A^{-1}\delta A\|} &\leq \frac{\|A^{-1}\| \cdot \|\delta A\|}{1 - \|A^{-1}\| \cdot \|\delta A\|} \\ &= \frac{\|A\| \cdot \|A^{-1}\|}{1 - \|A\| \cdot \|A^{-1}\| \frac{\|\delta A\|}{\|A\|}} \cdot \frac{\|\delta A\|}{\|A\|} \end{aligned}$$

从而得到不等式

$$\frac{\|\delta X\|}{\|X\|} \leq \frac{\text{Cond}(A)}{1 - \text{Cond}(A) \frac{\|\delta A\|}{\|A\|}} \cdot \frac{\|\delta A\|}{\|A\|}$$

若 $\text{Cond}(A) \frac{\|\delta A\|}{\|A\|}$ 很小, 则 $\text{Cond}(A)$ 仍表示相对误差的近似放大率.

根据以上讨论可以看出, $\text{Cond}(A)$ 反映线性方程组 $AX = \mathbf{b}$ 的解对初始数据误差的灵敏度, 其值愈大, 这种灵敏度愈高. 即对很小的初始误差 $\delta \mathbf{b}$ 或 δA , 解的相对误差就有可能很大, 从而大大破坏了解的精确度.

练习与思考

1. 分别用顺序高斯消去法、列主元高斯消去法、全主元高斯消去法求解下列方程组

$$\begin{bmatrix} 2 & 3 & 5 \\ 3 & 4 & 7 \\ 1 & 3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \\ 5 \end{bmatrix}$$

2. 用 LU 分解法求解下列方程组

$$\begin{bmatrix} 2 & -3 & 2 \\ -4 & 2 & -6 \\ 2 & 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 14 \\ 8 \end{bmatrix}$$

3. 用平方根法求解下列方程组

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

4. 设 $A = (a_{ij}^{(1)})_{n \times n}$, 经顺序高斯消去法一步后 A 变为

$$A^{(2)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} & \alpha \\ 0 & A_2 \end{bmatrix}$$

其中 $\alpha = [a_{12}^{(1)} \cdots a_{1n}^{(1)}]$, A_2 为 $(n-1) \times (n-1)$ 方阵. 证明:

(1) 若 A 按行强对角占优, 则矩阵 A_2 按行强对角占优;

(2) 若 A 对称正定, 则矩阵 A_2 对称正定.

5. 比较下面两个方程组的解

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1.00 & 0.50 & 0.33 \\ 0.50 & 0.33 & 0.25 \\ 0.33 & 0.25 & 0.20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

6. 设有

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

其系数矩阵强对角占优. 试写出用 LU 分解法求其解的计算公式.

7. 对矩阵

$$A = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}$$

求 $\|A\|_\infty$, $\|A\|_2$, $\|A\|_1$, $\|A\|_F$ 和 $\text{Cond}(A)_2$.

8. 试证明 $\|A\|_2 \leq \|A\|_1 \cdot \|A\|_\infty$.

9. 画出 \mathbf{R}^2 中满足下列不等式的集合

(1) $\|X\|_1 \leq 1$;

(2) $\|X\|_2 \leq 1$;

(3) $\|X\|_\infty \leq 1$.

10. 设 $A, B \in \mathbf{R}^{n \times n}$ 均非奇异, $\|\cdot\|$ 是任何矩阵范数. 证明:

(1) $\|A^{-1}\| \geq \frac{1}{\|A\|}$;

(2) $\|A^{-1} - B^{-1}\| \leq \|A^{-1}\| \cdot \|B^{-1}\| \cdot \|A - B\|$.

11. 设有方程组 $AX = b$, 其中

$$A = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 2 & 1 \\ 0 & 2 & 2 \end{bmatrix} \quad b = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{3} \\ -\frac{1}{3} \end{bmatrix}$$

如果右端 b 有小扰动 $\|\delta b\|_\infty = 5 \times 10^{-7}$, 试估计由此引起的解的相对误差.

12. 设有方程组 $AX = b$.

$$\begin{bmatrix} 240 & -319 \\ -179 & 240 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

及扰动方程组 $(A + \delta A)(X + \delta X) = b$:

$$\begin{bmatrix} 240 & -319.5 \\ -179.5 & 240 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

(1) 试分别解出 X 和 $X + \delta X$;

(2) 利用扰动误差分析理论估计 $\frac{\|\delta X\|_\infty}{\|X\|_\infty}$, 并将理论结果与实际求解结果比较.

6 解线性代数方程组的迭代法

前一章已经介绍了解线性代数方程组

$$\mathbf{A}\mathbf{X}=\mathbf{b} \tag{6.1}$$

的一些直接法. 这里 \mathbf{A} 是 $n \times n$ 矩阵; \mathbf{X}, \mathbf{b} 都是 n 维向量. 而大多数计算过程均需对系数矩阵进行分解, 因而一般不能保持 \mathbf{A} 的稀疏性. 而在实际问题中, 特别是偏微分方程数值求解, 常会遇到大型稀疏矩阵. 另外, 从上一章分析可以看到, 高斯消去法、LU 分解法的乘除次数为 $\frac{1}{3}n^3 + O(n^2)$, LDL^T 分解法的乘除次数为 $\frac{1}{6}n^3 + O(n^2)$, 当 n 较大时, 计算量相当大. 迭代法的基本思想是构造一个向量序列 $\{\mathbf{X}^{(n)}\}$, 使其收敛至某个极限向量 \mathbf{X}^* , 而 \mathbf{X}^* 就是要求的方程组 $\mathbf{A}\mathbf{X}=\mathbf{b}$ 的准确解. 迭代法能充分利用系数矩阵的稀疏性, 当 n 较大时, 能有效控制计算量.

迭代法要解决的主要问题如下:

- (1) 如何构造迭代格式?
- (2) 构造的格式所产生的序列在什么情况下收敛?
- (3) 如果收敛, 收敛的速率如何?
- (4) 近似解的误差估计.

6.1 几种常用的迭代格式

6.1.1 简单迭代法 (Jacobi 迭代)

设有方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \tag{6.2}$$

用矩阵表示为

$$\mathbf{A}\mathbf{X}=\mathbf{b}$$

其中 \mathbf{A} 是系数矩阵, 非奇异; \mathbf{b}, \mathbf{X} 为 n 维向量.

假设 $a_{ii} \neq 0$, 将方程组 (6.2) 中第一个方程的 $a_{11}x_1$ 项保留在左端, 其余项移到右端, 然后两边再除以 a_{11} . 同理将第 k 个方程的 $a_{kk}x_k$ 项保留在左端, 其余项移到

右端,然后两边再除以 a_{kk} ($k=2,3,\cdots,n$). 这样,方程组(6.2)变成下列等价方程组

$$\begin{cases} x_1 = & b_{12}x_2 + b_{13}x_3 + \cdots & + b_{1n}x_n + g_1 \\ x_2 = & b_{21}x_1 & + b_{23}x_3 + \cdots & + b_{2n}x_n + g_2 \\ & \vdots & & \\ x_n = & b_{n1}x_1 + b_{n2}x_2 & + \cdots + b_{n,n-1}x_{n-1} & + g_n \end{cases} \quad (6.3)$$

其中

$$\begin{cases} b_{ij} = \frac{-a_{ij}}{a_{ii}} & (i \neq j) \\ b_{ii} = 0 & (i = 1, 2, \cdots, n) \\ g_i = \frac{b_i}{a_{ii}} & (i = 1, 2, \cdots, n) \end{cases}$$

若令

$$B = \begin{bmatrix} 0 & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & 0 & b_{23} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & b_{n3} & \cdots & 0 \end{bmatrix} \quad D = \begin{bmatrix} a_{11} & & & & \\ & a_{22} & & & \\ & & \ddots & & \\ & & & a_{nn} & \end{bmatrix} \quad g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$$

容易看出

$$B = D^{-1}(D - A) = I - D^{-1}A \\ g = D^{-1}b$$

方程(6.3)用矩阵可表示为

$$X = BX + g \quad (6.4)$$

由此便可构造一个迭代格式(Jacobi 迭代)

$$X^{(k+1)} = BX^{(k)} + g \quad (6.5)$$

当取定初始向量 $X^{(0)} \in \mathbf{R}^n$ 后,上式便产生一个向量序列

$$X^{(0)}, X^{(1)}, \cdots, X^{(k)}, \cdots$$

若它收敛于 X^* ,则 X^* 一定是方程(6.4)的解,当然也是原方程 $AX = b$ 的解. 此时,称 Jacobi 迭代关于初始向量 $X^{(0)}$ 收敛.

Jacobi 迭代算法

①输入 A, b , 初始向量 Y , 容许误差 ε , 容许最大迭代次数 M .

②置 $k = 1$.

③形成迭代矩阵 B (存放在 A 中).

对 $i = 1, 2, \cdots, n$, 循环

若 $|a_{ii}| < \varepsilon$, 则打印“求解失败”, 停机; 否则

$$T = a_{ii}$$

对 $j=1, 2, \dots, n$, 计算

$$a_{ij} = -\frac{a_{ij}}{T}$$

$$a_{ii} = 0, g_i = \frac{b_i}{T}$$

④迭代:

对 $i=1, 2, \dots, n$, 计算

$$x_i = g_i + \sum_{j \neq i} a_{ij} y_j$$

⑤若 $\|X - Y\| < \varepsilon$, 输出 X, k , 停机; 否则

⑥若 $k < M$, 则 $k = k + 1$, 将 X 赋值给 Y , 转④; 否则, 输出求解失败信息, 停机.

6.1.2 Seidel 迭代法

设有简单迭代法

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

将迭代矩阵 $B = (b_{ij})_{n \times n}$ 分解为 $B = L + U$, 其中

$$L = \begin{bmatrix} 0 & & & \\ b_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ b_{n1} & b_{n2} & \cdots & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & b_{12} & \cdots & b_{1n} \\ & 0 & \cdots & b_{2n} \\ & & \ddots & \vdots \\ & & & 0 \end{bmatrix}$$

则
$$X^{(k+1)} = LX^{(k)} + UX^{(k)} + g \quad (k=0, 1, 2, \dots)$$

将其修改为

$$X^{(k+1)} = LX^{(k+1)} + UX^{(k)} + g \quad (k=0, 1, 2, \dots) \quad (6.6)$$

上式称为 Seidel 迭代, 其分量形式为

$$x_i^{(k+1)} = \sum_{j=1}^{i-1} b_{ij} x_j^{(k+1)} + \sum_{j=i+1}^n b_{ij} x_j^{(k)} + g_i \quad (i=1, 2, \dots, n) \quad (6.7)$$

它的特点在于, 计算第 i 个分量 $x_i^{(k+1)}$ 时, 前边的 $i-1$ 个分量用的是最新算出的 $x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}$, 而不是旧值 $x_1^{(k)}, \dots, x_n^{(k)}$, 这样有可能提高收敛速度.

因 $(I - L)^{-1}$ 存在, 迭代格式 (6.6) 可以改写成

$$X^{(k+1)} = (I - L)^{-1} UX^{(k)} + (I - L)^{-1} g \quad (6.8)$$

称 $B_1 = (I - L)^{-1} U$ 为 Seidel 迭代法的迭代矩阵.

Seidel 迭代算法

①输入 A, b , 初始向量 Y , 容许误差 ε , 容许最大迭代次数 M

②置 $k=1$, 对 $i=1, 2, \dots, n, x_i = y_i$

③形成迭代矩阵 B (存放在 A 中)

对 $i=1, 2, \dots, n$, 循环

若 $|a_{ii}| < \varepsilon$, 则打印“求解失败”, 停机; 否则

$$T = a_{ii}$$

对 $j=1, 2, \dots, n$, 计算

$$a_{ij} = \frac{-a_{ij}}{T}$$

$$a_{ii} = 0 \quad g_i = \frac{b_i}{T}$$

④迭代:

对 $i=1, 2, \dots, n$, 计算

$$x_i = g_i + \sum_{j \neq i} a_{ij} x_j$$

⑤若 $\|X - Y\| < \varepsilon$, 输出 X, k , 停机; 否则

⑥若 $k < M$, 则 $k = k + 1$, 将 X 赋值给 Y , 转④; 否则, 输出求解失败信息, 停机

6.1.3 松弛法(SOR 迭代)

松弛法可以看作是 Seidel 迭代法的加速, Seidel 迭代法是松弛法的特例. Seidel 迭代格式为

$$X^{(k+1)} = LX^{(k+1)} + UX^{(k)} + g$$

$$\text{现令} \quad \Delta X = X^{(k+1)} - X^{(k)} = LX^{(k+1)} + UX^{(k)} + g - X^{(k)}$$

$$\text{于是 } X^{(k+1)} = X^{(k)} + \Delta X.$$

对于 Seidel 迭代法, $X^{(k+1)}$ 可以看成向量 $X^{(k)}$ 加上修正项 ΔX 得到. 现在, 若在修正项的前面加上一个参数 ω , 便得到松弛法的计算格式

$$X^{(k+1)} = X^{(k)} + \omega \Delta X = (1 - \omega) X^{(k)} + \omega (LX^{(k+1)} + UX^{(k)} + g) \quad (k=0, 1, \dots) \quad (6.9)$$

用分量形式即为

$$x_i^{(k+1)} = (1 - \omega) x_i^{(k)} + \omega \left[\sum_{j=1}^{i-1} b_{ij} x_j^{(k+1)} + \sum_{j=i+1}^n b_{ij} x_j^{(k)} + g_i \right] \\ (i=1, 2, \dots, n; k=0, 1, 2, \dots) \quad (6.10)$$

其中, ω 称为松弛因子, 当 $\omega > 1$ 时叫超松弛; $\omega < 1$ 时称低松弛; $\omega = 1$ 时就是 Seidel 迭代法.

迭代格式(6.9)还可改写成

$$\begin{aligned} X^{(k+1)} &= (I - \omega L)^{-1} [(1 - \omega)I + \omega U] X^{(k)} + \omega (I - \omega L)^{-1} g \\ B_\omega &= (I - \omega L)^{-1} [(1 - \omega)I + \omega U] \end{aligned}$$

称为松弛法的迭代矩阵.

松弛迭代法算法

①输入 A, b , 初始向量 Y , 松弛因子 ω , 容许误差 ε , 容许最大迭代次数 M .

②置 $k=1$, 对 $i=1, 2, \dots, n, x_i = y_i$.

③形成迭代矩阵 B (仍存放在 A 中):

对 $i=1, 2, \dots, n$, 循环

若 $|a_{ii}| < \varepsilon$, 则打印“求解失败”, 停机; 否则

$$T = a_{ii}$$

对 $j=1, 2, \dots, n$, 计算

$$a_{ij} = -\omega * \frac{a_{ij}}{T}$$

$$a_{ii} = 1 - \omega, \quad g_i = \omega * \frac{b_i}{T}$$

④迭代:

对 $i=1, 2, \dots, n$, 计算

$$x_i = g_i + \sum_{j=1}^n a_{ij} x_j$$

⑤若 $\|X - Y\| < \varepsilon$, 输出 X, k ; 停机; 否则

⑥若 $k < M$, 则 $k = k + 1$, 将 X 赋值给 Y , 转④; 否则, 输出求解失败信息, 停机

6.2 迭代法收敛性理论

任意选取初始向量, 利用迭代格式(6.5)构造向量序列 $\{X^{(k)}\}$, 向量序列是否一定收敛呢? 先看两个例子.

例 6-1 用 Jacobi 迭代法解方程组

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

解 上述方程组的准确解是 $x_1 = 1.1, x_2 = 1.2, x_3 = 1.3$.

先把方程改写成

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

取 $\mathbf{X}^{(0)} = (0, 0, 0)^T$, 利用 Jacobi 迭代, 计算结果如表 6-1 所示.

表 6-1 计算结果

| k | $x_1^{(k)}$ | $x_2^{(k)}$ | $x_3^{(k)}$ | k | $x_1^{(k)}$ | $x_2^{(k)}$ | $x_3^{(k)}$ |
|-----|-------------|-------------|-------------|-----|-------------|-------------|-------------|
| 0 | 0 | 0 | 0 | 5 | 1.095 1 | 1.195 1 | 1.294 1 |
| 1 | 0.72 | 0.83 | 0.84 | 6 | 1.098 3 | 1.198 3 | 1.298 0 |
| 2 | 0.971 | 1.070 | 1.150 | 7 | 1.099 4 | 1.199 8 | 1.299 3 |
| 3 | 1.057 | 1.157 1 | 1.248 2 | 8 | 1.099 8 | 1.199 8 | 1.299 7 |
| 4 | 1.085 3 | 1.185 3 | 1.282 8 | 9 | 1.099 9 | 1.199 9 | 1.299 9 |

从表 6-1 中可以看到, 近似解向量序列收敛, 且收敛到准确解.

例 6-2 用 Jacobi 迭代法解方程组

$$\begin{cases} x_1 - 10x_2 + 20x_3 = 11 \\ -10x_1 + x_2 - 5x_3 = -14 \\ 5x_1 - x_2 - x_3 = 3 \end{cases}$$

上述方程的准确解是 $x_1 = x_2 = x_3 = 1$.

同样取 $\mathbf{X}^{(0)} = (0, 0, 0)^T$, 利用 Jacobi 迭代, 计算结果如表 6-2 所示.

从表 6-2 中可以看到, 此迭代发散. 可以证明, 除 $\mathbf{X}^{(0)} = (1, 1, 1)^T$ 外, 无论选什么初值都不会收敛.

从上面两个例子可以看出, 迭代序列收敛是有条件的. 下面给出迭代法收敛性基本理论.

表 6-2 计算结果

| k | $x_1^{(k)}$ | $x_2^{(k)}$ | $x_3^{(k)}$ |
|-----|-------------|-------------|-------------|
| 0 | 0 | 0 | 0 |
| 1 | 11 | -14 | -3 |
| 2 | -69 | 81 | 66 |
| 3 | -499 | -374 | -429 |

定理 6.1 对任何初始向量 $\mathbf{X}^{(0)}$ 和常数项 \mathbf{f} , 由迭代格式

$$\mathbf{X}^{(k+1)} = \mathbf{M}\mathbf{X}^{(k)} + \mathbf{f} \quad (k=0, 1, 2, \cdots)$$

产生的向量序列 $\{\mathbf{X}^{(k)}\}$ 收敛的充要条件是迭代矩阵的谱半径

$$\rho(\mathbf{M}) < 1$$

证明 先证必要性. 假设 $\{\mathbf{X}^{(k)}\}$ 收敛到 \mathbf{X}^* , 即 $\lim \mathbf{X}^{(k)} = \mathbf{X}^*$, 则有

$$\mathbf{X}^* = \mathbf{M}\mathbf{X}^* + \mathbf{f}$$

令 $\varepsilon_k = \mathbf{X}^{(k)} - \mathbf{X}^*$, 则

$$\varepsilon_{k+1} = X^{(k+1)} - X^* = MX^{(k)} - MX^* = M(X^{(k)} - X^*)$$

所以有

$$\varepsilon_{k+1} = M\varepsilon_k = M^2\varepsilon_{k-1} = \cdots = M^{k+1}\varepsilon_0 \quad (6.11)$$

对任意初始向量 ε_0 , 要使向量序列 $\{M^k\varepsilon_0\}$ 收敛于零向量, 必须

$$\lim_{k \rightarrow \infty} M^k = 0 \quad (6.12)$$

由定理 5.10 可知

$$\rho(M) < 1$$

再证充分性. 假设 $\rho(M) < 1$, 则 $I - M$ 非奇异, 从而方程组 $(I - M)X = f$ 有惟一解, 并记为 X^* , 于是 (6.11) 式仍成立. 由 (6.12) 推出 $\lim_{k \rightarrow \infty} X^{(k)} = X^*$. 证毕.

在具体问题中, 谱半径 $\rho(M)$ 往往很难计算, 但由于 $\rho(M) \leq \|M\|$, 所以有时可以用 $\|M\|$ 作为 $\rho(M)$ 的一种估计. 当 $\|M\| < 1$ 时迭代一定收敛, 不过这只是充分条件.

定理 6.2 若迭代矩阵 M 的范数 $\|M\| = q < 1$, 则迭代格式 $X^{(k+1)} = MX^{(k)} + f$ 对任何初始向量 $X^{(0)}$ 一定收敛, 且

$$\|X^{(k)} - X^*\| \leq \frac{q}{1-q} \|X^{(k-1)} - X^{(k)}\| \leq \frac{q^k}{1-q} \|X^{(1)} - X^{(0)}\|$$

证 由迭代格式得

$$\begin{aligned} X^{(k+1)} - X^{(k)} &= M(X^{(k)} - X^{(k-1)}) = \cdots = M^k(X^{(1)} - X^{(0)}) \\ \|X^{(k+1)} - X^{(k)}\| &\leq q \|X^{(k)} - X^{(k-1)}\| \leq q^k \|X^{(1)} - X^{(0)}\| \end{aligned}$$

注意到

$$X^{(k)} - X^* = X^{(k)} - X^{(k+1)} + X^{(k+1)} - X^* = X^{(k)} - X^{(k+1)} + M(X^{(k)} - X^*)$$

从而

$$\|X^{(k)} - X^*\| \leq \|X^{(k)} - X^{(k+1)}\| + q \|X^{(k)} - X^*\|$$

故

$$\begin{aligned} \|X^{(k)} - X^*\| &\leq \frac{1}{1-q} \|X^{(k+1)} - X^{(k)}\| \leq \frac{q}{1-q} \|X^{(k)} - X^{(k-1)}\| \\ &\leq \frac{q^k}{1-q} \|X^{(1)} - X^{(0)}\| \end{aligned}$$

6.2.1 三种迭代法迭代矩阵的谱半径与系数矩阵 A 的关系

6.2.1.1 Jacobi 迭代

Jacobi 迭代的迭代矩阵 $B = D^{-1}(D - A)$. 由

$$|\lambda I - B| = 0$$

得

$$|\lambda I - D^{-1}(D - A)| = 0$$

即

$$|D^{-1}| \cdot |\lambda D - (D - A)| = 0$$

所以

$$|\lambda D - (D - A)| = 0$$

上式写成分量形式为

$$\begin{vmatrix} \lambda a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \lambda a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & \lambda a_{nn} \end{vmatrix} = 0 \quad (6.13)$$

由定理 6.1 可知:

定理 6.3 Jacobi 迭代收敛的充要条件是行列式 (6.13) 的所有根 $\lambda_i (i=1, 2, \dots, n)$ 的绝对值 (复数理解为模) 小于 1.

6.2.1.2 Seidel 迭代

对于 Seidel 迭代, $B = (I - L)^{-1}U$. 由

$$|\lambda I - B| = 0$$

得

$$|\lambda I - (I - L)^{-1}U| = 0$$

即

$$|\lambda(I - L)^{-1}(I - L) - (I - L)^{-1}U| = 0$$

所以

$$|\lambda(I - L) - U| = 0 \quad |\lambda D - \lambda DL - DU| = 0$$

写成分量的形式为

$$\begin{vmatrix} \lambda a_{11} & a_{12} & \cdots & a_{1n} \\ \lambda a_{21} & \lambda a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ \lambda a_{n1} & \lambda a_{n2} & \cdots & \lambda a_{nn} \end{vmatrix} = 0 \quad (6.14)$$

定理 6.4 Seidel 迭代收敛的充要条件是行列式 (6.14) 的所有根 $\lambda_i (i=1, 2, \dots, n)$ 的绝对值小于 1.

6.2.1.3 松弛迭代 (SOR 迭代)

对于松弛迭代,

$$B_\omega = (I - \omega L)^{-1}[(1 - \omega)I + \omega U]$$

由 $|\lambda I - B_\omega| = 0$ 得

$$|\lambda I - (I - \omega L)^{-1}[(1 - \omega)I + \omega U]| = 0$$

化简即得

$$|\lambda(I - \omega L) - [(1 - \omega)I + \omega U]| = 0$$

注意到 $|D| \neq 0$, 得

$$|\lambda D - \omega \lambda DL - [(1 - \omega)D + \omega DU]| = 0$$

写成分量形式为

$$\begin{vmatrix} (\lambda + \omega - 1)a_{11} & \omega a_{12} & \cdots & \omega a_{1n} \\ \omega \lambda a_{21} & (\lambda + \omega - 1)a_{22} & \cdots & \omega a_{2n} \\ \vdots & \vdots & & \vdots \\ \omega \lambda a_{n1} & \omega \lambda a_{n2} & \cdots & (\lambda + \omega - 1)a_{nn} \end{vmatrix} = 0 \quad (6.15)$$

定理 6.5 松弛迭代收敛的充要条件是行列式 (6.15) 的所有根 $\lambda_i (i = 1, 2, \dots, n)$ 的绝对值小于 1.

对于一些特定的系数矩阵 A , 有一些特定的判别方法. 为了说明方便起见, 在此先引进一些概念.

定义 6.1 如果矩阵 A 不能通过行交换和相应的列交换变成

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

其中 A_{11}, A_{22} 为方阵, 则称 A 为不可约.

定义 6.2 若矩阵 $A = (a_{ij})_{n \times n}$ 满足

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

且至少有一个 i 值使上式中不等式严格成立, 则称矩阵 A 具有对角优势 (又称为弱对角占优). 特别地, 若所有的 i 值对上式中不等式都严格成立, 则称矩阵 A 具有强对角占优.

定理 6.6 若 A 是强对角占优, 或者 A 弱对角占优且不可约, 则 $\det A \neq 0$.

矩阵的不可约与矩阵对应的邻接图有一个必然的联系.

定理 6.7 A 矩阵不可约的充要条件是 A 矩阵对应的邻接图是一个强连通图.

定理 6.8 若 A 是强对角占优, 或者 A 弱对角占优且不可约, 则

(1) Jacobi 迭代、Seidel 迭代一定收敛.

(2) 若松弛因子 ω 满足 $0 < \omega \leq 1$, 则松弛迭代一定收敛.

证明 现以 Jacobi 迭代为例予以证明, Seidel 迭代同样适用.

采用反证法. 若 Jacobi 迭代不收敛, 即存在一个 $\lambda_0, |\lambda_0| \geq 1$, 使

$$\begin{vmatrix} \lambda_0 a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \lambda_0 a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & \lambda_0 a_{nn} \end{vmatrix} = 0 \quad (6.16)$$

注意到 $|\lambda_0| \cdot |a_{ii}| \geq |a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \quad (i = 1, 2, \dots, n)$

因此 (6.16) 式构成的矩阵的对角占优性与 A 矩阵相同, 因此 (6.16) 式不可能成立. 说明 Jacobi 迭代收敛.

定理 6.9 松弛法收敛的必要条件是 $0 < \omega < 2$.

证明 因为松弛法收敛, 故有 $\rho(B_\omega) < 1$. 由矩阵 B_ω 的特征值的性质可知

$$|\det B_\omega| = |\lambda_1 \lambda_2 \cdots \lambda_n| < 1$$

其中 $\lambda_1, \lambda_2, \dots, \lambda_n$ 是矩阵 B_ω 的 n 个特征值. 而

$$B_\omega = (I - \omega L)^{-1} [(1 - \omega)I + \omega U]$$

所以

$$|\det \mathbf{B}_\omega| = |(1-\omega)^n| = |\lambda_1 \lambda_2 \cdots \lambda_n| < 1$$

即

$$|1-\omega| < 1, 0 < \omega < 2.$$

上述定理说明, 对于任何系数矩阵 \mathbf{A} , 若要松弛法收敛, 其松弛因子 $\omega \in (0, 2)$. 然而, 当松弛因子满足条件 $0 < \omega < 2$ 时, 并不是对所有系数矩阵 \mathbf{A} 松弛法均收敛.

定理 6.10 若矩阵 \mathbf{A} 对称且对角线元素均为正实数, 则当 $0 < \omega < 2$ 时, 松弛法收敛的充要条件是 \mathbf{A} 正定.

证明 松弛法的迭代矩阵

$$\mathbf{B}_\omega = (\mathbf{I} - \omega \mathbf{L})^{-1} [(1-\omega)\mathbf{I} + \omega \mathbf{U}]$$

设其特征值为 $\lambda_1, \lambda_2, \dots, \lambda_n$, 相应的特征向量为 $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_n$, 则

$$[(1-\omega)\mathbf{I} + \omega \mathbf{U}]\mathbf{Y}_k = \lambda_k (\mathbf{I} - \omega \mathbf{L})\mathbf{Y}_k$$

两边用 \mathbf{Y}_k 作内积得

$$\lambda_k = \frac{(1-\omega)(\mathbf{Y}_k, \mathbf{Y}_k) + \omega(\mathbf{U}\mathbf{Y}_k, \mathbf{Y}_k)}{(\mathbf{Y}_k, \mathbf{Y}_k) - \omega(\mathbf{L}\mathbf{Y}_k, \mathbf{Y}_k)} \quad (6.17)$$

设 $(\mathbf{L}\mathbf{Y}_k, \mathbf{Y}_k) = \alpha_k + \beta_k i$, $(\mathbf{Y}_k, \mathbf{Y}_k) = a$. 则

$$(\mathbf{U}\mathbf{Y}_k, \mathbf{Y}_k) = (\mathbf{L}^T \mathbf{Y}_k, \mathbf{Y}_k) = (\mathbf{Y}_k, \mathbf{L}\mathbf{Y}_k) = \alpha_k - \beta_k i$$

将上面结果代入(6.17)式得

$$\begin{aligned} \lambda_k &= \frac{(1-\omega)a + \omega\alpha_k - \omega\beta_k i}{(a - \omega\alpha_k) - i\omega\beta_k} \\ |\lambda_k|^2 &= \frac{[(1-\omega)a + \omega\alpha_k]^2 + \omega^2 \beta_k^2}{(a - \omega\alpha_k)^2 + \omega^2 \beta_k^2} \end{aligned} \quad (6.18)$$

由于 \mathbf{A} 正定, 则 $\mathbf{A}_1 = \mathbf{D}^{-1}\mathbf{A}$ 对称正定.

$$(\mathbf{A}_1 \mathbf{Y}_k, \mathbf{Y}_k) = ((\mathbf{I} - \mathbf{L} - \mathbf{U})\mathbf{Y}_k, \mathbf{Y}_k) = a - 2\alpha_k > 0$$

所以 $[(1-\omega)a + \omega\alpha_k]^2 - (a - \omega\alpha_k)^2 = (2-\omega)\omega a(2\alpha_k - a) < 0$

即(6.18)式中的分子 $[(1-\omega)a + \omega\alpha_k]^2 + \omega^2 \beta_k^2$ 小于分母 $(a - \omega\alpha_k)^2 + \omega^2 \beta_k^2$, 于是 $|\lambda_k|^2 < 1$, $\rho(\mathbf{B}_\omega) < 1$, 松弛迭代法收敛.

松弛迭代法的收敛速度与松弛因子 ω 有关. 下面举一个例子.

例 6-3 取初始向量 $\mathbf{X}^{(0)} = (1, 1, 1)^T$, 用 SOR 法求解方程组

$$\begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 24 \\ 30 \\ -24 \end{bmatrix}$$

使 $|\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}| < 10^{-5}$. 该方程组的精确解为 $\mathbf{X}^* = (3, 4, -5)$.

解 SOR 法的迭代公式为

$$\begin{cases} x_1^{(k+1)} = (1-\omega)x_1^{(k)} + \frac{\omega}{4}[24-3x_2^{(k)}] \\ x_2^{(k+1)} = (1-\omega)x_2^{(k)} + \frac{\omega}{4}[30-3x_1^{(k+1)}+x_3^{(k)}] \\ x_3^{(k+1)} = (1-\omega)x_3^{(k)} + \frac{\omega}{4}[-24+x_2^{(k+1)}] \end{cases}$$

分别取 $\omega = 1.8, \omega = 1.22$, 迭代结果如表 6-3 和表 6-4 所示.

表 6-3 $\omega = 1.8$ 时的迭代结果

| k | $x_1^{(k)}$ | $x_2^{(k)}$ | $x_3^{(k)}$ |
|----------|-------------|-------------|-------------|
| 0 | 1 | 1 | 1 |
| 1 | 8.65 | 1.472 4 | -10.937 37 |
| 2 | 1.892 126 | 4.845 811 | 0.130 513 5 |
| 3 | 2.744 454 | 5.977 064 | -8.214 29 |
| 4 | 0.535 393 2 | 4.298 935 | -2.293 696 |
| \vdots | \vdots | \vdots | \vdots |
| 64 | 3.000 001 | 3.999 999 | -4.999 996 |
| 65 | 3.000 001 | 4.000 001 | -5.000 002 |

表 6-4 $\omega = 1.22$ 时的迭代结果

| k | $x_1^{(k)}$ | $x_2^{(k)}$ | $x_3^{(k)}$ |
|----------|-------------|-------------|-------------|
| 0 | 1 | 1 | 1 |
| 1 | 6.185 | 3.575 732 | -6.449 404 |
| 2 | 2.687 512 | 3.937 199 | -4.700 285 |
| 3 | 3.126 210 | 3.989 747 | -5.069 064 |
| 4 | 2.981 615 | 3.998 013 | -4.985 412 |
| \vdots | \vdots | \vdots | \vdots |
| 10 | 3.000 000 | 3.999 998 | -4.999 999 |
| 11 | 3.000 002 | 4.000 000 | -5.000 000 |

从表 6-3、表 6-4 可以看到,在相同的初始条件下, $\omega = 1.8$ 时 SOR 法迭代了 65 步,而 $\omega = 1.22$ 时 SOR 法仅迭代了 11 次.

使松弛法收敛最快的松弛因子叫最优松弛因子,记为 ω_{opt} . 对于某些特殊类型

的矩阵,可以证明最优松弛因子为:

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{B})}}$$

其中 $\rho(\mathbf{B})$ 是迭代矩阵的谱半径. 对一般的矩阵(即使是正定对称矩阵),目前尚无法确定 ω_{opt} 的理论值. 实际计算时,大部分由经验或通过试算来确定 ω_{opt} 的一个近似值.

定理 6.11 设

① \mathbf{A} 为分块三对角阵,且 $a_{ii} \neq 0$;

② Jacobi 迭代的迭代矩阵 \mathbf{B} 的特征值为实值,且 $0 < \rho(\mathbf{B}) < 1$.

则

(1) 当 $0 < \omega < 2$ 时, SOR 法迭代收敛;

(2) SOR 法最优松弛因子

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{B})}}$$

练习与思考

1. 用简单迭代法, Seidel 迭代法和取 $\omega = 1.46$ 的超松弛迭代法解方程组

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

并写出相应的迭代矩阵.

2. 已知方程组

$$\begin{bmatrix} 10 & -2 & -2 \\ -2 & 10 & -1 \\ -1 & -2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1 \end{bmatrix}$$

(1) 构造简单迭代法和 Seidel 迭代法的迭代格式;

(2) 讨论这些迭代格式的收敛性.

3. 设有系数矩阵

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

证明:

(1) 对系数矩阵 A , 简单迭代法收敛, 而 Seidel 迭代法不收敛;

(2) 对系数矩阵 B , 简单迭代法收敛, 而 Seidel 迭代法收敛.

4. 设常数 $a \neq 0$, 试求 a 的取值范围, 使得方程组

$$\begin{bmatrix} a & 1 & 3 \\ 1 & a & 2 \\ -3 & 2 & a \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

的简单迭代法关于任何初始向量均收敛.

5. 设 $B \in \mathbf{R}^{n \times n}$, $\rho(B) > 1$, 但 B 有一个特征值满足 $|\lambda| < 1$. 试证明存在初始向量 $x^{(0)}$, 使得简单迭代

$$X^{(k+1)} = BX^{(k)} + f \quad (k=0, 1, 2, \dots)$$

关于此初始向量收敛.

6. 设 $A \in \mathbf{R}^{n \times n}$, 且对称正定, 其最小特征值和最大特征值分别是 λ_1, λ_n . 试证迭代法

$$X^{(k+1)} = X^{(k)} + \alpha(b - AX^{(k)})$$

收敛的充要条件是 $0 < \alpha < 2/\lambda_n$. 又问参数取何值时迭代矩阵的谱半径最小?

7. 对方程组

$$\begin{bmatrix} 1 & 2 \\ 0.3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

拟用迭代法

$$X^{(k+1)} = X^{(k)} + \alpha(AX^{(k)} - b)$$

求解. 试确定一个 α 的取值范围, 使得上述迭代公式收敛.

7 非线性方程和非线性方程组的数值解

求方程 $f(x) = 0$ 的根是最常见的数学问题之一. 当 $f(x)$ 是一次多项式时, 称 $f(x) = 0$ 为线性方程, 否则称之为非线性方程.

当 $f(x) = 0$ 是非线性方程时, 由于 $f(x)$ 的多样性, 求方程 $f(x) = 0$ 的根尚无一般的解析解法可用. 如果在满足一定的精度要求下, 能求出方程的近似根, 则可以认为求根的计算问题已经解决.

本章介绍求非线性方程实根近似值的对分法、迭代法、牛顿法和割线法. 使用这些方法要先确定求根区间 $[a, b]$, 或给出某根的近似值 x_0 . x_0 可以根据物理背景确定, 也可根据 $y = f(x)$ 的草图等方法确定, 还可用对分法大致确定根的分布情况. 本章最后简单介绍解非线性方程组的迭代法和牛顿法.

7.1 对分法

对分法适用于求有根区间内的单实根或奇重实根, 其理论根据是连续函数的性质.

设 $f(x)$ 在 $[a, b]$ 上连续, $f(a) \cdot f(b) < 0$, 即 $f(a) > 0, f(b) < 0$ 或 $f(a) < 0, f(b) > 0$, 则根据连续函数的介值定理, 在 (a, b) 内至少存在一点 ξ , 使 $f(\xi) = 0$.

对分法首先取区间 $[a, b]$ 的中点 $x_1 = \frac{a+b}{2}$, 把区间 $[a, b]$ 分为两个小区间 $[a, x_1]$ 、 $[x_1, b]$, 如果 $f(a) \cdot f(x_1) < 0$, 令 $a_1 = a, b_1 = x_1$; 否则令 $a_1 = x_1, b_1 = b$, 得到有根区间 $[a_1, b_1]$. 用同样的方法把区间 $[a_1, b_1]$ 对分, 得到有根区间 $[a_2, b_2]$ …… 这样一直下去, 可得到长度每次减半的有根区间序列 $[a_k, b_k]$. 当 k 足够大时, 有根区间 $[a_k, b_k]$ 长度趋于零, 区间任一点都可作为根的近似值, 如图 7-1 所示.

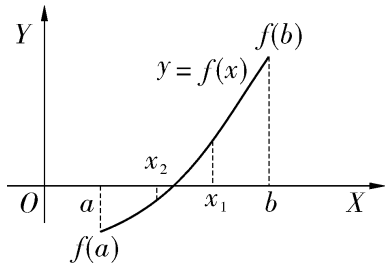


图 7-1

7.1.0.1 对分法算法

①给出精度 δ, ε , 令 $a_0 = a, b_0 = b, k = 0$.

②令 $x_k = (a_k + b_k)/2$, 计算 $f(x_k)$.

③若 $|f(x_k)| < \delta$, 则 x_k 是 $f(x) = 0$ 的根, 停止计算, 输出结果 $\xi = x_k$;

若 $f(a_k) \cdot f(x_k) < 0$, 则令 $a_{k+1} = a_k, b_{k+1} = x_k$;

否则令 $a_{k+1} = x_k, b_{k+1} = b_k$.

④若 $b_{k+1} - a_{k+1} \leq \varepsilon$ (ε 为预先给定的精度要求), 退出计算, 输出结果 $\xi \approx (a_{k+1} + b_{k+1})/2$; 反之, 令 $k = k + 1$, 返回②.

上面方法可得到区间长度每次缩小一半的区间序列 $\{[a_k, b_k]\}$, (a_k, b_k) 中总含有方程的根. 当区间长 $b_k - a_k$ 很小时, 取其中点 $x_k = \frac{a_k + b_k}{2}$ 为根的近似值.

7.1.0.2 收敛性分析

因为

$$|x_k - \xi| \leq \frac{1}{2}(b_k - a_k) = \frac{1}{2^2}(b_{k-1} - a_{k-1}) = \cdots = \frac{1}{2^{k+1}}(b - a) \quad (7.1)$$

所以, 当 $k \rightarrow \infty$ 时, x_k 收敛于 ξ .

公式(7.1)还可用于估计对分次数 k .

对分法的收敛速度与公比为 $1/2$ 的等比级数相同. 由于 $2^{10} = 1\,024$, 可知大约每对分 10 次, 近似根的精度就可提高三位小数位. 对分法的收敛速度较慢, 常用来试探实根的分布区间, 或求根的初始近似值.

例 7-1 求 $x^3 - 3x + 1 = 0$ 的实根分布情况, 并求 $[0, 1]$ 中的实根近似值. 要求实根近似值精确到三位小数.

解 从区间 $[-4, 4]$ 以步长为 1 计算 $f(x) = x^3 - 3x + 1 = 0$ 的函数值, 结果如表 7-1 所示.

表 7-1 函数值表

| x | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|--------|-----|-----|----|----|---|----|---|----|----|
| $f(x)$ | -51 | -17 | -1 | 3 | 1 | -1 | 3 | 19 | 53 |

可见, 在区间 $[-2, -1]$ 、 $[0, 1]$ 、 $[1, 2]$ 各有一实根, 如图 7-2 所示. 下面求 $[0, 1]$ 区间上的实根, 计算列表如表 7-2 所示.

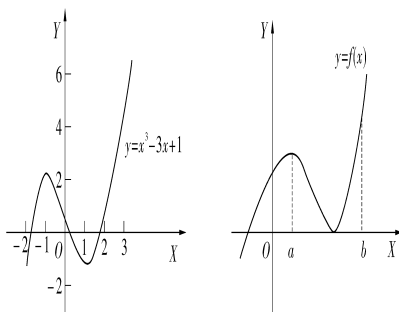


图 7-2

表 7-2 对分法试算表

| k | a_k | b_k | x_k | $f(x_k)$ |
|-----|---------------|---------------|---------------|--------------|
| 1 | 0 | 1 | 0.5 | -33.75 |
| 2 | 0 | 0.5 | 0.25 | 0.265 625 |
| 3 | 0.25 | 0.5 | 0.375 | -0.072 27 |
| 4 | 0.25 | 0.375 | 0.312 5 | 0.093 02 |
| 5 | 0.312 5 | 0.375 | 0.343 75 | 0.009 369 |
| 6 | 0.343 75 | 0.375 | 0.359 375 | -0.031 71 |
| 7 | 0.343 75 | 0.359 375 | 0.351 562 5 | -0.011 24 |
| 8 | 0.343 75 | 0.351 562 5 | 0.347 656 25 | -0.000 949 |
| 9 | 0.343 75 | 0.347 656 25 | 0.345 703 125 | 0.004 206 |
| 10 | 0.345 703 125 | 0.347 656 25 | 0.346 679 687 | 0.001 627 |
| 11 | 0.346 679 687 | 0.347 656 25 | 0.347 167 968 | 0.000 338 7 |
| 12 | 0.347 167 968 | 0.347 656 25 | 0.347 412 109 | -0.000 305 4 |
| 13 | 0.347 167 968 | 0.347 412 109 | 0.347 290 038 | 0.000 016 66 |

由于 $x \in [0.347\ 167\ 968, 0.347\ 412\ 109]$, 可取 $x \approx 0.347\ 290\ 033$ 作近似解.

7.1.0.3 对分法的特点

对分法的优点是对函数的要求低(只要求 $f(x)$ 连续),方法简单、可靠,程序设计容易,可事先估计计算次数.缺点是收敛速度较慢,不能求出偶重根.如图 7-3 所示,因为 $f(x)$ 在根 x 的两边附近函数值同号,不能用对分法求之.

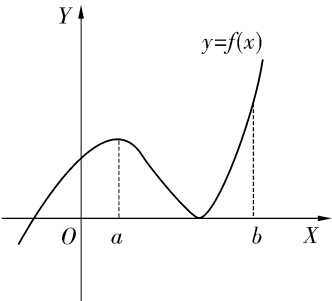


图 7-3

7.2 迭代法

7.2.1 迭代法

给定方程 $f(x) = 0$, 可用多种方法构造它的等价方程 $x = \varphi(x)$, 取定根的一个近似值 x_0 , 构造序列

$$x_{k+1} = \varphi(x_k) \quad (k=0, 1, 2, \cdots) \tag{7.2}$$

若 $\{x_k\}$ 收敛, 即 $\lim_{k \rightarrow \infty} x_k = x^*$, 并且 $\varphi(x)$ 连续, 则有

$$\lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} \varphi(x_k) = \varphi(\lim_{k \rightarrow \infty} x_k)$$

即 $x^* = \varphi(x^*)$

故, $\{x_k\}$ 的极限 x^* 是方程 $x = \varphi(x)$ 的根, 也就是方程 $f(x) = 0$ 的根.

当然, 若 $\{x_k\}$ 发散, 迭代法就失败.

7.2.1.1 迭代法算法

① 给定初值 x_0 , 精度 ε , $k=0$.

② 计算 $x_{k+1} = \varphi(x_k)$.

③ 若 $|x_{k+1} - x_k| < \varepsilon$, 输出 x_{k+1} ; 否则取 $k = k + 1$, 返回②

例 7-2 求 $x^3 - 3x + 1 = 0$ 在 0.5 附近的根. 要求精确到第四位小数.

解 构造等价方程 $x = \varphi_1(x) = \frac{x^3 + 1}{3}$, 格式 $x_{k+1} = \frac{x_k^3 + 1}{3}$, $k = 0, 1, 2, \dots$, 取 $x_0 =$

0.5. 计算结果如表 7-3 所示.

表 7-3 计算结果

| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|----------|----------|----------|----------|----------|----------|----------|
| x_k | 0.500 00 | 0.375 00 | 0.350 91 | 0.347 74 | 0.347 35 | 0.347 30 | 0.347 30 |

由表 7-3 可知, $x_5 = x_6 = 0.347\ 30$ 为根的近似值.

注意: 格式 $x_{k+1} = \frac{x_k^3 + 1}{3}$ 不能求出方程在 1.5 和 -2 附近的根, 请读者自己验证.

此外, 方程还可以有另外的格式, 如

$$x_{k+1} = \varphi_2(x_k) = \frac{1}{3 - x_k^2} \quad (k = 0, 1, 2, \dots) \quad (7.3)$$

$$x_{k+1} = \varphi_3(x_k) = \sqrt[3]{3x_k - 1} \quad (k = 0, 1, 2, \dots) \quad (7.4)$$

7.2.2 迭代法的几何意义

从上例可看到, 由 $f(x) = 0$ 构造出的等价方程 $x = \varphi(x)$ 可以有多种, 有的可能收敛, 有的可能不收敛, 这取决于 $\varphi(x)$ 的性态. 这里先叙述迭代法的几何意义, 关于迭代法的收敛性, 后面再讨论.

将 $x = \varphi(x)$ 写成

$$\begin{cases} y = x \\ y = \varphi(x) \end{cases}$$

则求解方程 $x = \varphi(x)$ 等价于求直线 $y = x$ 和曲线 $y = \varphi(x)$ 的交点的横坐标 x^* , 如图 7-4、图 7-5、图 7-6 和图 7-7 所示.

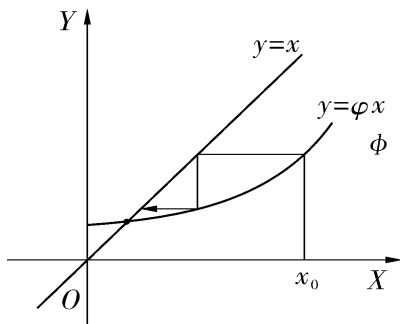


图 7-4

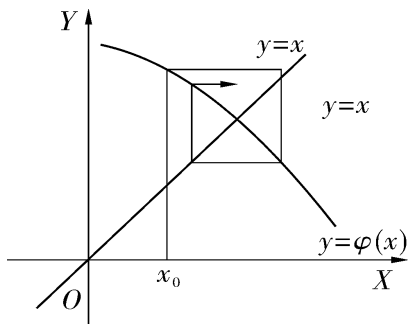


图 7-5

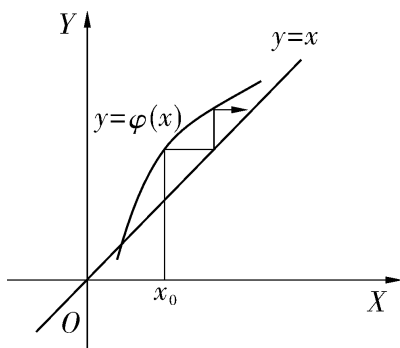


图 7-6

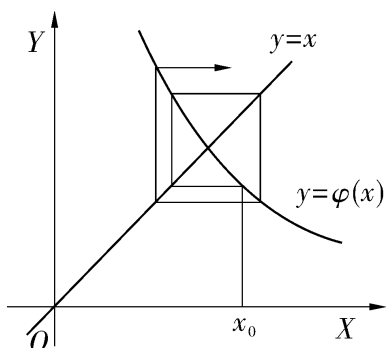


图 7-7

图 7-4、图 7-5 是收敛情况,而图 7-6、图 7-7 是发散情况.可以大致看出,如 $\varphi(x)$ 在 x^* 附近形状比较陡峭(不管是上升或是下降),则发散;若 $\varphi(x)$ 比较平坦,则收敛.

7.2.3 迭代法收敛条件

定义 7.1 如果在根 x^* 的某个邻域 $\mathbf{R}: |x - x^*| \leq \delta$ 中,对任意的 $x_0 \in \mathbf{R}$,格式 $x_{k+1} = \varphi(x_k)$, $k=0,1,2,\dots$ 收敛,则称格式在 x^* 附近局部收敛.

定理 7.1 设 $x^* = \varphi(x^*)$, $\varphi'(x)$ 在 x^* 的某个领域 \mathbf{R} 内连续,并且 $|\varphi'(x)| \leq q, q < 1$ 是常量,则

(1) 对任意 $x_0 \in \mathbf{R}$,由格式 $x_{k+1} = \varphi(x_k)$ 决定的序列 $\{x_k\}$ 收敛于 x^* .

$$(2) |x_k - x^*| \leq \frac{q}{1-q} |x_1 - x_0| \quad (7.5)$$

$$|x_k - x^*| \leq \frac{1}{1-q} |x_{k+1} - x_k| \quad (7.6)$$

证明 (1) 由拉格朗日中值定理,存在 $\xi \in \mathbf{R}$,使

$$x_k - x^* = \varphi(x_{k-1}) - \varphi(x^*) = \varphi'(\xi)(x_{k-1} - x^*)$$

有 $|x_k - x^*| \leq q |x_{k-1} - x^*| \leq \cdots \leq q^k |x_0 - x^*|$

所以 $\lim_{k \rightarrow \infty} x_k = x^*$.

(2) 由拉格朗日中值定理, 有

$$x_{k+1} - x_k = \varphi(x_k) - \varphi(x_{k-1}) = \varphi'(\xi)(x_k - x_{k-1})$$

$$|x_{k+1} - x_k| \leq q |x_k - x_{k-1}| \leq \cdots \leq q^k |x_1 - x_0|$$

因为

$$x_{k+p} - x_k = x_{k+p} - x_{k+p-1} + x_{k+p-1} - \cdots - x_{k+1} + x_{k+1} - x_k$$

所以

$$\begin{aligned} |x_{k+p} - x_k| &\leq |x_{k+p} - x_{k+p-1}| + |x_{k+p-1} - x_{k+p-2}| + \cdots + |x_{k+1} - x_k| \\ &\leq (q^p + q^{p-1} + \cdots + 1) |x_{k+1} - x_k| = \frac{q^{p+1} - 1}{q - 1} |x_{k+1} - x_k| \end{aligned}$$

令 $p \rightarrow \infty$, 有

$$|x^* - x_k| \leq \frac{1}{1-q} |x_{k+1} - x_k| \leq \frac{q}{1-q} |x_k - x_{k-1}| \leq \cdots \leq \frac{q^k}{1-q} |x_1 - x_0|$$

说明: (7.5) 式常用来估计次数, 但结果偏保守, 次数偏大; (7.6) 式常用来作为程序退出的条件, 即当 $|x_k - x_{k-1}| < \varepsilon$ 时退出.

定理 7.2 给定方程 $x = \varphi(x)$, 若 $\varphi(x)$ 满足:

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

② 对任意的 $x, y \in [a, b]$, 有 $|\varphi(x) - \varphi(y)| \leq q |x - y|$, $0 \leq q < 1$ 为常数.

则对任意的 $x_0 \in [a, b]$, 格式 $x_{k+1} = \varphi(x_k)$ 生成的序列 $\{x_k\}$ 收敛于 $x = \varphi(x)$ 的根 x^* .

证明与定理 7.1 类似, 不再重复.

注意: 满足条件①和②的函数 $\varphi(x)$ 称为区间的压缩映射.

定理 7.1 可用来粗略估计格式在根的附近的收敛情况.

例 7-3 讨论例 7-2 的三种格式在三个根附近的收敛情况, 这三个根近似地取 $x_1 = 0.347, x_2 = 1.53, x_3 = -1.88$.

解 格式一 $x_{k+1} = \frac{x_k^3 + 1}{3}, \varphi_1'(x) = x^2$

$$|\varphi_1'(x_1)| = 0.124\ 09 < 1 \quad \text{收敛}$$

$$|\varphi_1'(x_2)| = 2.340\ 9 > 1 \quad \text{不收敛}$$

$$|\varphi_1'(x_3)| = 3.534\ 4 > 1 \quad \text{不收敛}$$

格式二 $x_{k+1} = \frac{1}{3-x_k}, \varphi_2'(x) = \frac{2x}{(3-x^2)^2}$

$$|\varphi_2'(x_1)| = 0.08 < 1 \quad \text{收敛}$$

$$|\varphi_2'(x_2)| = 7.04 > 1 \quad \text{不收敛}$$

$$|\varphi_2'(x_3)| = 13.16 > 1 \quad \text{不收敛}$$

$$\text{格式三 } x_{k+1} = \sqrt[3]{3x_k - 1}, \varphi'_3(x) = \frac{1}{3}(3x - 1)^{-\frac{2}{3}}$$

$$|\varphi'_3(x_1)| = 8.41 > 1 \quad \text{不收敛}$$

$$|\varphi'_3(x_2)| = 0.426 < 1 \quad \text{收敛}$$

$$|\varphi'_3(x_3)| = 0.283 < 1 \quad \text{收敛}$$

7.2.4 迭代法的加速

由定理 7.1 可知,若由 $f(x) = 0$ 构造出的格式 $x = \varphi(x)$ 收敛时,收敛速度取决于 $|\varphi'(x)|$ 的大小,当 $|\varphi'(x)|$ 接近于 1 时,收敛可能很慢.能否从 $x = \varphi(x)$ 出发构造出新的格式,使收敛速度加快呢? 下面讨论两种常用的加速方法.

7.2.4.1 松弛法

已知 $\varphi(x_k)$ 与 x_k 同是 x^* 的近似值.那么 $x_{k+1} = (1 - \omega_k)x_k + \omega_k\varphi(x_k)$ 是两个近似值的加权平均,其中 ω_k 称为权重.现通过确定 ω_k 看能否得到加速.

因为
$$x = (1 - \omega)x + \omega\varphi(x) = \phi(x)$$

令 $\phi'(x) = 1 - \omega + \omega\varphi'(x) = 0$, 当 $\varphi'(x) \neq 1$ 时,解得 $\omega = \frac{1}{1 - \varphi'(x)}$. 即取

$\omega_k = \frac{1}{1 - \varphi'(x_k)}$, $1 - \omega_k = \frac{-\varphi'(x_k)}{1 - \varphi'(x_k)}$ 时,可望获得较好的加速效果.

松弛法格式如下:

$$\begin{cases} \omega_k = \frac{1}{1 - \varphi'(x_k)} \\ x_{k+1} = (1 - \omega_k)x_k + \omega_k\varphi(x_k) \end{cases} \quad (7.7)$$

松弛法算法

①给定初值 x_0 , 精度 $\varepsilon, k = 0$.

②计算:

$$\omega_k = \frac{1}{1 - \varphi'(x_k)}$$

$$x_{k+1} = (1 - \omega_k)x_k + \omega_k\varphi(x_k)$$

③若 $|x_{k+1} - x_k| < \varepsilon$, 输出 x_{k+1} ; 否则, 取 $k = k + 1$, 返回②.

松弛法的加速效果是明显的.

7.2.4.2 埃特金 (Aitken) 方法

松弛法要计算导数 $\varphi'(x_k)$, 使用中有时不方便, 为此引入埃特金方法.

设方程 $x = \varphi(x)$, x^* 是它的准确根, x_0 是其近似根. 取 $x_1 = \varphi(x_0)$, $x_2 = \varphi(x_1)$.

因为

$$x^* = x_2 + x^* - x_2 = x_2 + \varphi'(\xi)(x^* - x_1)$$

用差商

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

近似代替 $\varphi'(\xi)$, 有

$$x^* \approx x_2 + \frac{x_2 - x_1}{x_1 - x_0}(x^* - x_1)$$

解出 x^* 得

$$x^* \approx x_2 - \frac{(x_2 - x_1)^2}{x_2 - 2x_1 + x_0}$$

由此得到公式

$$\begin{cases} x_k^{(1)} = \varphi(x_k) \\ x_k^{(2)} = \varphi(x_k^{(1)}) \\ x_{k+1} = x_k^{(2)} - \frac{(x_k^{(2)} - x_k^{(1)})^2}{x_k^{(2)} - 2x_k^{(1)} + x_k} \end{cases} \quad (k=0, 1, 2, \dots) \quad (7.8)$$

这就是埃特金公式, 它的加速效果也十分明显(见例 7-4).

埃特金算法

① 给定初值 x_0 , 精度 $\varepsilon, k=0$.

② 计算:

$$x_k^{(1)} = \varphi(x_k)$$

$$x_k^{(2)} = \varphi(x_k^{(1)})$$

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

③ 若 $|x_{k+1} - x_k| < \varepsilon$, 输出 x_{k+1} ; 否则取 $k=k+1$, 返回②.

下面给出埃特金迭代算法的几何解释.

如图 7-8, 设 x_0 为方程 $x = \varphi(x)$ 的一个近似根, 由 $x_1^{(1)} = \varphi(x_0)$ 和 $x_1^{(2)} = \varphi(x_1^{(1)})$ 在曲线 $y = \varphi(x)$ 上可以定出两点 $p_0(x_0, x_1^{(1)})$, $p_1(x_1^{(1)}, x_1^{(2)})$, 作弦 $\overline{p_0 p_1}$ 与直线 $y = x$ 交于 p , 则 p 点的坐标 x_1 满足

$$x_1 = x_1^{(1)} - \frac{x_1^{(2)} - x_1^{(1)}}{x_1^{(1)} - x_0}(x_1 - x_1^{(1)})$$

解出 x_1 即得埃特金公式.

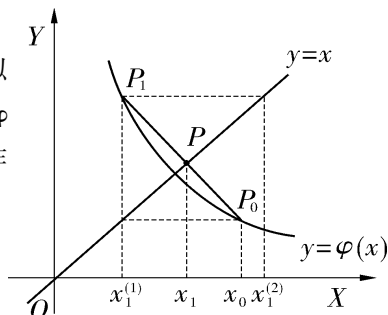


图 7-8

例 7-4 解方程 $x^3 - 3x + 1 = 0$, 格式 $x_{k+1} = \frac{x_k^3 + 1}{3}$, 用松弛法和埃特金法, 取 $x_0 = 0.5$, 精确到 6 位小数.

解 松弛法 $\varphi(x) = \frac{x^3 + 1}{3}, \varphi'(x) = x^2, \omega_k = \frac{1}{1 - x_k^2}$. 公式为

$$x_{k+1} = (1 - \omega_k)x_k + \omega_k \frac{x_k^3 + 1}{3}$$

计算结果如表 7-4 所示.

表 7-4 松弛法计算结果

| k | 0 | 1 | 2 | 3 | 4 |
|------------|-----------|-----------|-----------|-----------|-----------|
| x_k | 0.5 | 0.333 333 | 0.347 222 | 0.347 296 | 0.347 296 |
| ω_k | 1.333 333 | 1.125 000 | 1.137 091 | 1.137 158 | |

可见确有明显的加速效果.

埃特金法 $x_k^{(1)} = \frac{x_k^3 + 1}{3}$

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

计算结果如表 7-5 所示.

表 7-5 埃特金法计算结果

| k | x_k | $x_k^{(1)}$ | $x_k^{(2)}$ |
|-----|-----------|-------------|-------------|
| 0 | 0.5 | 0.375 | 0.350 911 |
| 1 | 0.345 161 | 0.347 040 | 0.347 265 |
| 2 | 0.347 296 | | |

例 7-5 对上题, 取 $x_0 = 1.5$, 用埃特金算法计算.

解 计算结果如表 7-6 所示.

表 7-6 埃特金法计算结果($x_0 = 1.5$)

| k | x_k | $x_k^{(1)}$ | $x_k^{(2)}$ |
|-----|-----------|-------------|-------------|
| 0 | 1.5 | 1.458 333 | 1.367 163 |
| 1 | 1.535 071 | 1.539 102 | 1.548 626 |
| 2 | 1.532 113 | 1.532 145 | 1.532 222 |
| 3 | 1.532 090 | 1.532 092 | 1.532 096 |
| 4 | 1.532 088 | 1.532 087 | 1.532 084 |
| 5 | 1.532 088 | | |

7.3 牛顿(Newton)法

7.3.1 牛顿公式

由于 $f(x) = 0$ 是非线性方程,一般解决非线性问题较困难,而解决线性问题较容易,故可以考虑将非线性问题线性化,采取线性化方法进行求解.把 $f(x)$ 在 x_0 作泰勒展开,有

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(\xi)}{2!}(x - x_0)^2 = p(x) + \frac{f''(\xi)}{2!}(x - x_0)^2$$

其中 $p(x) = f(x_0) + f'(x_0)(x - x_0) = 0$ 是线性方程.用 $p(x) = 0$ 作为 $f(x) = 0$ 的近似方程,即把求曲线 $f(x) = 0$ 与 X 轴交点的横坐标近似为求直线 $p(x) = 0$ 与 X 轴交点的横坐标.

$$p(x) = f(x_0) + f'(x_0)(x - x_0) = 0$$

的解为

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (\text{设 } f'(x_0) \neq 0)$$

记

$$\begin{aligned} x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} \\ &\vdots \\ x_{k+1} &= x_k - \frac{f(x_k)}{f'(x_k)} \quad (k = 0, 1, 2, \cdots) \end{aligned} \tag{7.9}$$

这就是牛顿法公式.

例 7-6 用牛顿法计算 $x^3 - 3x + 1 = 0$ 在 0.5 和 -2 附近的两个根.

解 因为 $f(x) = x^3 - 3x + 1, f'(x) = 3x^2 - 3$, 所以

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

计算结果如表 7-7 所示.

表 7-7 牛顿法计算结果

| k | 0 | 1 | 2 | 3 | 4 |
|-------------|-----|------------|------------|------------|------------|
| $x_k^{(1)}$ | 0.5 | 0.333 333 | 0.347 222 | 0.347 296 | 0.347 296 |
| $x_k^{(2)}$ | -2 | -1.888 889 | -1.879 452 | -1.879 385 | -1.879 385 |

牛顿法实质上是一般迭代法用松弛法加速,把 $f(x) = 0$ 化为等价方程 $x = x + f(x) = \varphi(x)$,对此使用松弛法:

$$\begin{aligned}\varphi'(x) &= 1 + f'(x) \\ \omega_k &= \frac{1}{1 - \varphi'(x_k)} = -\frac{1}{f'(x_k)}\end{aligned}$$

可得牛顿法公式:

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

由此可知牛顿法的收敛速度应当较快,例 7-6 也印证了这一点.

牛顿算法

①给定初值 x_0 ,精度 $\varepsilon, k=0$.

②计算 $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$.

③若 $|x_{k+1} - x_k| < \varepsilon$,输出 x_{k+1} ;否则取 $k = k + 1$,返回②.

下面给出牛顿法的几何意义.

$f(x) = 0$ 的解就是 $y = f(x)$ 与 X 轴的交点的横坐标.若 x_k 是近似解,过 $(x_k, f(x_k))$ 作 $y = f(x)$ 的切线,则切线方程为

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

它与 X 轴的交点横坐标是 $x_k - \frac{f(x_k)}{f'(x_k)}$,如图

7-9 所示.

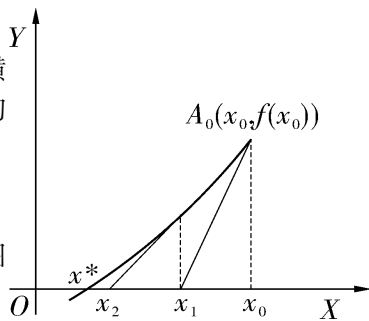


图 7-9

7.3.2 牛顿法的收敛速度

迭代法的基本要求是能收敛,而迭代法的优劣取决于收敛的快慢.为了衡量收敛序列的收敛速度,先给出收敛序列收敛阶定义.

定义 7.2 设序列 $\{x_k\}$ 收敛于 x^* , 令 $\varepsilon_k = x^* - x_k$, 设 $k \rightarrow \infty$ 时, 有

$$\frac{|\varepsilon_{k+1}|}{|\varepsilon_k|^p} \rightarrow c \quad (c > 0 \text{ 为常数})$$

则称序列 $\{x_k\}$ 是 p 阶收敛. 当 $p=1$ 时称线性收敛; 当 $p=2$ 时称二阶收敛(几何收敛); $1 < p < 2$ 时称超线性收敛.

下面是迭代法阶收敛定理.

定理 7.3 设 $x^* = \varphi(x^*)$, 在 x^* 的某个邻域 R 内 $\varphi^{(p)}(x)$ 连续, $p > 1$ 是常量, 并且满足

$$\varphi^{(l)}(x^*) = 0 \quad (l = 1, 2, \dots, p-1) \quad (7.10)$$

$$\varphi^{(p)}(x^*) \neq 0 \quad (7.11)$$

则由 $x_{k+1} = \varphi(x_k)$ 生成的序列 $\{x_k\}$ 收敛于 x^* , 并且序列 $\{x_k\}$ 是 p 阶收敛.

证明 由 $\varphi'(x^*) = 0$ 和定理 7.1 可知, $x_{k+1} = \varphi(x_k)$ 生成的序列 $\{x_k\}$ 收敛于 x^* . 再由泰勒展开

$$x_{k+1} = \varphi(x_k) = \varphi(x^*) + \varphi'(x^*)(x - x^*) + \dots + \frac{\varphi^{(p-1)}(x^*)}{(p-1)!}(x_k - x^*)^{p-1} + \frac{\varphi^{(p)}(\xi_k)}{p!}(x_k - x^*)^p$$

其中 ξ 在 x_k 和 x^* 之间. 代入 $x^* = \varphi(x^*)$ 和条件(7.10)得

$$\frac{e_{k+1}}{e_k^p} = \frac{\varphi^{(p)}(\xi_k)}{p!}$$

注意到当 $k \rightarrow \infty$, $x_k \rightarrow x^*$, 从而 $\xi_k \rightarrow x^*$, 又 $\varphi^{(p)}(x^*) \neq 0$, 定理得证.

下面讨论牛顿法的收敛速度.

因为牛顿法的格式为

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

$$\text{有} \quad \varphi'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}$$

当 $f'(x^*) \neq 0$ 时, 由于分子上的 $f(x^*) = 0$, 所以 $\varphi'(x^*) = 0$. 因此, 牛顿法至少是二阶收敛, 即牛顿法在单根附近至少是二阶收敛的, 而在重根附近, 牛顿法是线性收敛.

牛顿法的优点是收敛很快, 算法简单, 是常用的快速收敛法; 缺点是对重根收

敛较慢,对初值 x_0 要求较严,要求 x_0 相当接近真解 x^* . 实际应用中,它常与其他方法联用,先用其他方法确定初值 x_0 ,再用牛顿法提高精度.

7.4 割线法

已知 $f(x) = 0$ 的两个近似根 x_k, x_{k-1} , 过点 $(x_{k-1}, f(x_{k-1}))$ 、 $(x_k, f(x_k))$ 连一条直线,据两点式可写出直线方程:

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

把求曲线 $f(x) = 0$ 与 X 轴交点的横坐标近似为求直线(割线)与 X 轴交点的横坐标. 即把该直线与 X 轴的交点的横坐标作 x^* 的新的近似值 x_{k+1} :

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1}) \quad (k = 1, 2, \dots) \quad (7.12)$$

此种方法称为割线法.

很明显,也可由牛顿法用差商 $\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$ 近似代替微商 $f'(x_k)$ 得到. 它是牛顿法的一种近似方法.

割线法算法

- ① 给定初值 x_0, x_1 , 精度 $\varepsilon, k = 1$.
- ② 计算 $x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1})$.
- ③ 若 $|x_{k+1} - x_k| < \varepsilon$, 输出 x_{k+1} ; 否则取 $k = k + 1$, 返回②.

割线法需要两个初值 x_0, x_1 , 在单根附近是超线性收敛, 通过较复杂的推导可知收敛阶 p 在单根附近为 $1.618\dots$, 收敛也相当快, 而且不用计算导数值.

它的几何意义如图 7-10 所示. 它的收敛定理下面直接给出.

定理 7.4 如 $f(x)$ 在零点 x^* 附近有连续的 2 阶导数, $f'(x^*) \neq 0$, 且初值 x_0, x_1 充分接近 x^* , 则割线法收敛, 收敛速度为

$$|x_{k+1} - x^*| \approx \left| \frac{f''(x^*)}{2f'(x^*)} \right|^{0.618} |x_k - x^*|^{1.618}$$

割线法需要两个初值 x_0, x_1 , 故也称为双点割线法. 有时为了简单起见, 可只用一个初值 x_0 ,

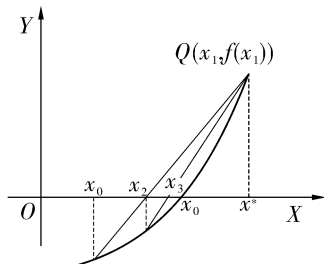


图 7-10

称为单点割线法. 单点割线法如下:

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_0)}(x_k - x_0) \quad (k = 1, 2, \cdots)$$

例 7-7 用双点割线法求 $x^3 - 3x + 1 = 0$ 在 0.5 附近的根, 精确到小数点后第六位.

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

取 $x_0 = 0.5, x_1 = 0.2$, 计算结果如表 7-8 所示.

表 7-8 割线法计算表

| k | 1 | 2 | 3 | 4 | 5 |
|-----------|-----|-----------|-----------|-----------|-----------|
| x_{k-1} | 0.5 | 0.2 | 0.356 322 | 0.347 731 | 0.347 295 |
| x_k | 0.2 | 0.356 322 | 0.347 731 | 0.347 295 | 0.347 296 |

7.5 解非线性方程组的迭代法和牛顿法

解一元非线性方程的迭代法和牛顿法可以推广到多元. 本节简单介绍解非线性方程组的迭代法和牛顿法.

n 个未知数 n 个方程的非线性方程组可表示为

$$\begin{cases} f_1(x_1, x_2, \cdots, x_n) = 0 \\ f_2(x_1, x_2, \cdots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \cdots, x_n) = 0 \end{cases} \tag{7.13}$$

其中 $\mathbf{X} = (x_1, x_2, \cdots, x_n)^T$ 为 n 维列向量; $f_i(\mathbf{X}) (i = 1, 2, \cdots, n)$ 中至少有一个是 \mathbf{X} 的非线性函数, 并假设自变量和函数值都是实数.

7.5.1 迭代法

记

$$\mathbf{F}(\mathbf{X}) = [f_1(\mathbf{X}), f_2(\mathbf{X}), \cdots, f_n(\mathbf{X})]^T$$

则方程组(7.13)可简写为

$$\mathbf{F}(\mathbf{X}) = 0 \tag{7.14}$$

显然, 函数 $\mathbf{F}(\mathbf{X})$ 的值也是 n 维列向量. 多元非线性方程组(7.14)与一元非线性方程 $f(x) = 0$ 具有相同的形式, 可以与一元函数一样讨论它的解法, 例如迭代法和牛顿法.

下面叙述迭代法. 首先把方程组(7.14)转换成等价方程组

$$\mathbf{X} = \Phi(\mathbf{X}) \quad (7.15)$$

其中

$$\Phi(\mathbf{X}) = [\varphi_1(\mathbf{X}), \varphi_2(\mathbf{X}), \dots, \varphi_n(\mathbf{X})]^T$$

并构造格式

$$\mathbf{X}^{(k+1)} = \Phi(\mathbf{X}^{(k)}) \quad (k=0, 1, 2, \dots) \quad (7.16)$$

对于给定的初始点 $\mathbf{X}^{(0)}$, 若由此生成的序列收敛, 如 $\lim_{k \rightarrow \infty} \mathbf{X}^{(k)} = \mathbf{X}^*$, 则 \mathbf{X}^* 满足 $\mathbf{X}^* = \Phi(\mathbf{X}^*)$, 即 \mathbf{X}^* 是方程组(7.14)的解.

非线性方程组的算法

① 给定初值 $\mathbf{X}^{(0)}$, 精度 $\varepsilon, k=1$.

② 计算 $\mathbf{X}^{(k+1)} = \Phi(\mathbf{X}^{(k)})$.

③ $\|\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}\| < \varepsilon$, 输出 $\mathbf{X}^{(k+1)}$; 否则取 $k = k+1$, 返回②.

定义 7.3 \mathbf{R}^n 中集合 M 称为闭集, 是指 M 中的任一个向量序列 $\{\mathbf{X}^{(k)}\}$, 若 $\mathbf{X}^{(k)} \rightarrow \mathbf{X}^*$ (即 $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \rightarrow 0$), 则 $\mathbf{x}^* \in \mathbf{R}^n$.

定义 7.4 设 $\Phi(\mathbf{X}) = [\varphi_1(\mathbf{X}), \varphi_2(\mathbf{X}), \dots, \varphi_n(\mathbf{X})]^T$ 是一个 n 维列向量函数, M 是 \mathbf{R}^n 中的子集合, 若满足:

① $\mathbf{X} \in M \Rightarrow \Phi(\mathbf{X}) \in M$;

② 存在常数 $q, 0 \leq q < 1$, 使 M 中任意 \mathbf{X}, \mathbf{Y} , 满足

$$\|\Phi(\mathbf{X}) - \Phi(\mathbf{Y})\| \leq q \|\mathbf{X} - \mathbf{Y}\|$$

则称 $\Phi(\mathbf{X})$ 为 M 上的压缩映射.

类似于一元函数, 有下面压缩映射定理.

定理 7.5 若 $\Phi(\mathbf{X})$ 为闭集 M 上的压缩映射, 则 $\mathbf{X} = \Phi(\mathbf{X})$ 在 M 上有惟一解. 且对任意 $\mathbf{X}^{(0)} \in M$, 由格式 $\mathbf{X}^{(k+1)} = \Phi(\mathbf{X}^{(k)})$ 产生的向量序列 $\{\mathbf{X}^{(k)}\}$ 收敛于这个解.

例 7-8 设有非线性方程组

$$\begin{cases} x_1^2 - 10x_1 + x_2^2 + 8 = 0 \\ x_1x_2^2 + x_1 - 10x_2 + 8 = 0 \end{cases} \quad (7.17)$$

把它写成等价形式

$$\begin{cases} x_1 = \varphi_1(x_1, x_2) = \frac{1}{10}(x_1^2 + x_2^2 + 8) \\ x_2 = \varphi_2(x_1, x_2) = \frac{1}{10}(x_1x_2^2 + x_1 + 8) \end{cases}$$

并由此构造格式

$$\begin{cases} x_1^{(k+1)} = \varphi_1(x_1^{(k)}, x_2^{(k)}) = \frac{1}{10}[(x_1^{(k)})^2 + (x_2^{(k)})^2 + 8] \\ x_2^{(k+1)} = \varphi_2(x_1^{(k)}, x_2^{(k)}) = \frac{1}{10}[x_1^{(k)}(x_2^{(k)})^2 + x_1^{(k)} + 8] \end{cases} \quad (k=0, 1, 2, \dots) \quad (7.18)$$

取初始点 $\mathbf{X}^{(0)} = (0,0)^T$, 计算结果如表 7-9 所示. 可见迭代收敛到方程组 (7.17) 的解 $\mathbf{X}^* = (1,1)^T$.

表 7-9 计算结果

| k | 0 | 1 | 2 | ... | 18 | 19 |
|-------------|-----|-----|---------|-----|---------------|---------------|
| $x_1^{(k)}$ | 0.0 | 0.8 | 0.928 0 | ... | 0.999 999 972 | 0.999 999 989 |
| $x_2^{(k)}$ | 0.0 | 0.8 | 0.931 2 | ... | 0.999 999 972 | 0.999 999 989 |

7.5.2 牛顿法

对于非线性方程组,也可以构造类似于一元方程的牛顿法,而且同样具有二次局部收敛性. 首先介绍多元收敛阶的定义.

定义 7.5 设序列 $\{\mathbf{X}^{(k)}\}_{k=0}^\infty \subset \mathbf{R}^n$ 收敛到 \mathbf{X}^* , 若有常数 $p \geq 1$ 和 $c > 0$, 使得

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{X}^{(k+1)} - \mathbf{X}^*\|}{\|\mathbf{X}^{(k)} - \mathbf{X}^*\|^p} = c$$

则称 p 为该序列的收敛阶. 当 $p = 1$ 时称为线性收敛; 当 $p > 1$ 时称为超线性收敛; 当 $p = 2$ 时称为二次收敛或平方收敛.

设 \mathbf{X}^* 是方程组 (7.13) 的解, $\mathbf{X}^{(k)}$ 是某个值. 用点 $\mathbf{X}^{(k)}$ 处的一阶泰勒展开近似每一个分量函数值 $f_i(\mathbf{X}^*) = 0$, 有

$$0 = f_i(\mathbf{X}^*) \approx f_i(\mathbf{X}^{(k)}) + \sum_{j=1}^n \frac{\partial f_i(\mathbf{X}^{(k)})}{\partial x_j} (x_j^* - x_j^{(k)}) \quad (i = 1, 2, \cdots, n)$$

或按方程组 (7.14) 用矩阵和向量表示为

$$0 = F(\mathbf{X}^*) \approx F(\mathbf{X}^{(k)}) + F'(\mathbf{X}^{(k)})(\mathbf{X}^* - \mathbf{X}^{(k)}) \tag{7.19}$$

其中函数 $F(\mathbf{X})$ 的导数

$$F'(\mathbf{X}) = \begin{bmatrix} \nabla f_1(\mathbf{X})^T \\ \nabla f_2(\mathbf{X})^T \\ \vdots \\ \nabla f_n(\mathbf{X})^T \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(\mathbf{X})}{\partial x_1} & \frac{\partial f_1(\mathbf{X})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{X})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{X})}{\partial x_1} & \frac{\partial f_2(\mathbf{X})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{X})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n(\mathbf{X})}{\partial x_1} & \frac{\partial f_n(\mathbf{X})}{\partial x_2} & \cdots & \frac{\partial f_n(\mathbf{X})}{\partial x_n} \end{bmatrix}$$

$F'(\mathbf{X})$ 称为函数 $F(\mathbf{X})$ 的 Jacobi 矩阵. 若矩阵 $F'(\mathbf{X}^{(k)})$ 非奇异, 则可从 (7.19) 中解出近似值, 并把它作为下一次值. 于是得到牛顿格式:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - F'(\mathbf{X}^{(k)})^{-1} F(\mathbf{X}^{(k)}) \quad (k = 0, 1, 2, \cdots) \tag{7.20}$$

其中, $\mathbf{X}^{(0)}$ 是给定的初始值.

非线性方程组的牛顿算法

- ① 给定初值 $\mathbf{X}^{(0)}$, 精度 $\varepsilon, k=1$.
- ② 计算 $\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - F'(\mathbf{X}^{(k)})^{-1} F(\mathbf{X}^{(k)})$.
- ③ $\|\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}\| < \varepsilon$, 输出 $\mathbf{X}^{(k+1)}$; 否则, 取 $k = k + 1$, 返回②.

牛顿法的收敛性, 可归为迭代法来讨论. 它的迭代函数是:

$$\Phi(\mathbf{X}) = \mathbf{X} - F'(\mathbf{X})^{-1} F(\mathbf{X})$$

定理 7.6 对于函数 $F: D \subset \mathbf{R}^n \rightarrow \mathbf{R}^n$. 设 $\mathbf{X}^* \in D$ 满足 $F(\mathbf{X}^*) = 0$. 若有 \mathbf{X}^* 的开邻域 $S_0 \subset D$, $F'(\mathbf{X})$ 在其上存在并连续, 而且 $F'(\mathbf{X}^*)$ 非奇异, 则存在 \mathbf{X}^* 的闭球 $S = S(\mathbf{X}^*, \delta) \subset S_0$ (其中 $\delta > 0$)

- (1) 牛顿函数 $\Phi(\mathbf{X})$ 对所有 $\mathbf{X} \in S$ 有定义, 并且 $\Phi(\mathbf{X}) \in S$.
- (2) 对于任何初值 $\mathbf{X}^{(0)} \in S$, 牛顿序列 $\{\mathbf{X}^{(k)}\}$ 超线性收敛于 \mathbf{X}^* .
- (3) 若有常数 $q > 0$, 使得

$$\|F'(\mathbf{X}) - F'(\mathbf{X}^*)\| \leq q \|\mathbf{X} - \mathbf{X}^*\|$$

$\forall \mathbf{X} \in S$, 则牛顿序列 $\{\mathbf{X}^{(k)}\}$ 至少二阶收敛于 \mathbf{X}^* .

例 7-9 用牛顿法求解例 7-8 中的方程组 (7.17).

解 此例的函数 $F(\mathbf{X})$ 和它的导数 $F'(\mathbf{X})$ 分别为

$$F(\mathbf{X}) = \begin{bmatrix} x_1^2 - 10x_1 + x_2^2 + 8 \\ x_1 x_2^2 + x_1 - 10x_2 + 8 \end{bmatrix} \quad F'(\mathbf{X}) = \begin{bmatrix} 2x_1 - 10 & 2x_2 \\ x_2^2 + 1 & 2x_1 x_2 - 10 \end{bmatrix}$$

取初值 $\mathbf{X}^{(0)} = (0, 0)^T$, 用牛顿法公式 (7.20), 计算结果如表 7-10.

表 7-10 计算结果

| k | 0 | 1 | 2 | 3 | 4 |
|-------------|-----|------|---------------|---------------|---------------|
| $x_1^{(k)}$ | 0.0 | 0.80 | 0.991 787 22 | 0.999 975 229 | 1.000 000 000 |
| $x_2^{(k)}$ | 0.0 | 0.88 | 0.991 711 737 | 0.999 968 524 | 1.000 000 000 |

可见, 牛顿法的收敛速度比例 7-8 中的迭代法 (7.18) 式要快.

练习与思考

1. 用对分法求下列方程的根, 要求绝对误差限 0.005.

- (1) $x^3 - x + 4 = 0$, 求根区间为 $(-2, -1)$;
- (2) $2x = e^{-x}$, 求根区间为 $(0, 1)$.

2. 举出对分法不能求偶数重根的例子.

3. 用对分法搜索方程 $6x^3 - 4x - 1 = 0$ 的实根分布情况, 初始搜索区间为

$(-2, 2)$, 并求出 $(0, 1)$ 中的根, 精确到 0.01.

4. 证明方程 $f(x) = 5x^2 - 120x + 30 = 0$ 在区间 $[0, 1]$ 中有惟一实根. 并验证下面两种格式的收敛性:

$$(1) x = \varphi_1(x) = 5x^2 - 119x + 3;$$

$$(2) x = \varphi_2(x) = \frac{5x^3 + 3}{120}.$$

5. 若 $\varphi(x) \in c[a, b]$, 且 $a \leq \varphi(x) \leq b$, 证明方程 $x = \varphi(x)$ 在 $[a, b]$ 上有惟一解.

6. 用迭代法求 $x^3 - 2x - 5 = 0$ 的正根. 下面有三种格式:

$$(1) x = \frac{x^3 - 5}{2};$$

$$(2) x = \frac{5}{x^2 - 2};$$

$$(3) x = \sqrt[3]{2x + 5}.$$

判别它们的收敛情况.

7. 证明方程 $x = \varphi(x) = \frac{1}{x+4}$ 在 $(-\infty, +\infty)$ 上有惟一解. 且对任意 $x_0 \in (-\infty, +\infty)$, 迭代法收敛.

8. 用迭代法求 $x = \cos x$ 的正根, 收敛较慢. 试用埃特金方法加速收敛, 求此根到 4 位小数, 并与迭代法的速度比较.

9. 写出牛顿法的格式, 并用牛顿法计算:

(1) 不开平方计算 $\sqrt{10}$, 精确到 5 位小数;

(2) 方程 $x^3 - x - 1 = 0$ 在 1.3 附近的根, 精确到 5 位小数.

10. 用割线法解方程 $6x^3 - 4x - 1 = 0$, 取初值 $x_0 = 1, x_1 = 1.3$, 精确到 5 位小数.

11. 给定方程 $f(x) = 0$, 取格式为 $x = x + cf(x)$, 其中 c 为非零常数. 若 $f(a) = 0, f'(a) \neq 0$, 求使 $f(x)$ 收敛于 a , c 的取值范围是多少?

12. 用迭代法解方程组

$$\begin{cases} x_1^2 - 10x_1 + x_2^2 + 23 = 0 \\ x_1x_2^2 + x_1 - 10x_2 + 2 = 0 \end{cases}$$

取 $x_0 = 3.8, y_0 = 0.9$, 迭代两次.

13. 用牛顿法解方程组

$$\begin{cases} x^2 + y^2 - 1 = 0 \\ x^2 - y = 0 \end{cases}$$

取 $x_0 = 0.8, y_0 = 0.6$, 迭代两次.

8 矩阵特征值和特征向量的数值解法

设 A 是 $n \times n$ 矩阵, 如果存在数 λ 和 n 维非零向量 x , 满足 $Ax = \lambda x$, 则称 λ 为矩阵 A 的一个特征值, x 称为与 λ 对应的特征向量.

本章讨论几种常用的计算矩阵特征值及特征向量的数值方法, 包括幂法、反幂法、Jacobi 法和 QR 法, 这里假设 A 是实矩阵.

8.1 幂法

8.1.1 幂法概述

幂法主要用于求矩阵按模最大的特征值和对应的特征向量.

设 A 具有 n 个线性无关的特征向量 x_1, x_2, \dots, x_n , 其相应的特征值 $\lambda_1, \lambda_2, \dots, \lambda_n$ 满足

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \quad (8.1)$$

现任取一个非零向量 μ_0 , 令

$$\mu_k = A\mu_{k-1} \quad (k = 1, 2, \dots) \quad (8.2)$$

得向量序列 $\{\mu_k\}$, $k = 0, 1, 2, \dots$. 因为 x_1, x_2, \dots, x_n 线性无关, 故 n 维向量 μ_0 可由它们线性表示为

$$\begin{aligned} \mu_0 &= a_1 x_1 + a_2 x_2 + \dots + a_n x_n \\ \text{所以 } \mu_k &= A\mu_{k-1} = A^2 \mu_{k-2} = \dots = A^k \mu_0 \\ &= a_1 A^k x_1 + a_2 A^k x_2 + \dots + a_n A^k x_n \\ &= a_1 \lambda_1^k x_1 + a_2 \lambda_2^k x_2 + \dots + a_n \lambda_n^k x_n \\ &= \lambda_1^k \left[a_1 x_1 + a_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k x_2 + \dots + a_n \left(\frac{\lambda_n}{\lambda_1} \right)^k x_n \right] \end{aligned} \quad (8.3)$$

设 $a_1 \neq 0$, 因为 $\left| \frac{\lambda_2}{\lambda_1} \right| < 1, \dots, \left| \frac{\lambda_n}{\lambda_1} \right| < 1$, 当 k 充分大时, 有 $\mu_k \approx \lambda_1^k a_1 x_1$, 且为非零向量, 所以 μ_k 可近似地作为 λ_1 对应的特征向量.

实际计算时, 为防止 μ_k 的模过大或过小产生的计算机运算的上下溢出, 通常每次迭代都对 μ_k 进行归一化, 使 $\|\mu_k\|_\infty = 1$. 因此, 幂法公式改进为

$$y_{k-1} = \frac{\mu_{k-1}}{\|\mu_{k-1}\|_\infty}$$

$$\mu_k = A y_{k-1} \quad (k=1, 2, \dots) \quad (8.4)$$

由迭代公式知

$$\begin{aligned} \mu_k &= \frac{A \mu_{k-1}}{\|\mu_{k-1}\|_\infty} = \frac{\frac{A^2 \mu_{k-2}}{\|\mu_{k-2}\|_\infty}}{\left\| \frac{A \mu_{k-2}}{\|\mu_{k-2}\|_\infty} \right\|_\infty} \\ &= \frac{A^2 \mu_{k-2}}{\|A \mu_{k-2}\|_\infty} = \dots = \frac{A^k \mu_0}{\|A^{k-1} \mu_0\|_\infty} \end{aligned} \quad (8.5)$$

对比(8.3)式和(8.5)式,可知(8.5)式中的 μ_k 仍收敛于 A 的对应于 λ_1 的特征向量.

当 k 相当大时, μ_k 和 μ_{k-1} 都可视为 λ_1 对应的特征向量. 由

$$\mu_k = A \frac{\mu_{k-1}}{\|\mu_{k-1}\|_\infty} \quad (8.6)$$

有

$$\begin{aligned} \mu_k &= \lambda_1 \frac{\mu_{k-1}}{\|\mu_{k-1}\|_\infty} \\ \|\mu_k\|_\infty &= |\lambda_1| \frac{\|\mu_{k-1}\|_\infty}{\|\mu_{k-1}\|_\infty} = |\lambda_1| \end{aligned} \quad (8.7)$$

故 $|\lambda_1|$ 就是 $\|\mu_k\|_\infty$, 剩下的问题是确定 λ_1 的符号. $\|\mu_k\|_\infty$ 是 μ_k 的各分量中绝对值最大者. 设 μ_k 和 μ_{k-1} 的绝对值最大的分量为 a_k 和 a_{k-1} , 由(8.7)式可得

$$a_k = \lambda_1 \frac{a_{k-1}}{\|\mu_{k-1}\|_\infty}$$

解得

$$\begin{aligned} \lambda_1 &= \frac{a_k}{a_{k-1}} \|\mu_{k-1}\|_\infty \\ \operatorname{sgn}(\lambda_1) &= \operatorname{sgn}\left(\frac{a_k}{a_{k-1}}\right) \end{aligned} \quad (8.8)$$

即当按(8.4)式迭代时,发现 $\|\mu_k\|_\infty$ 稳定后,取 $|\lambda_1| = \|\mu_k\|_\infty$, 而 λ_1 的符号由如下原则确定:当相邻两次的 μ_k 和 μ_{k-1} 对应分量符号相同, λ_1 的符号为正;否则为负.

幂法算法:

给定 $n \times n$ 矩阵 A , 精度 ε , 任取非零向量 μ_0 , 对 $k=1, 2, \dots$, 执行以下各步骤:

① 设 $|a_r| = \max_{1 \leq i \leq n} |a_i|$, 其中 a_i 是 $\mu_{k-1} = (a_1, a_2, \dots, a_n)^T$ 的分量.

② $y_{k-1} = \mu_{k-1} / |a_r|$.

③ $\mu_k = Ay_{k-1}$.

④ $t_k = \text{sgn}(a_r) \times (\mu_k)_r$.

⑤ 若 $|t_k - t_{k-1}| < \varepsilon$, 令 $\lambda_1 = t_k, x_1 = y_{k-1}$, 退出运算; 否则返回①.

例 8-1 求矩阵 A 按模最大的特征值 λ_1 和相应的特征向量.

$$A = \begin{bmatrix} 1 & -1 & 2 \\ -2 & 0 & 5 \\ 6 & -3 & 6 \end{bmatrix}$$

解 计算结果见表 8-1 所示.

表 8-1 计算结果

| k | μ_k^T | | | y_k^T | | | t_k |
|----------|-----------|----------|----------|----------|----------|----------|----------|
| 0 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | |
| 1 | 2.000 | 3.000 | 9.000 | 0.222 2 | 0.333 3 | 1.000 | 9 |
| 2 | 1.889 | 4.556 | 6.333 | 0.298 2 | 0.719 3 | 1.000 | 6.333 |
| 3 | 1.579 | 4.404 | 5.632 | 0.280 4 | 0.781 9 | 1.000 | 5.632 |
| 4 | 1.498 | 4.439 | 5.336 | 0.280 8 | 0.831 9 | 1.000 | 5.336 |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| 10 | 1.393 | 4.444 | 5.013 | 0.277 9 | 0.886 5 | 1.000 | 5.013 |
| 11 | 1.391 | 4.444 | 5.008 | | | | 5.008 |

所以 $\lambda_1 \approx 5.008, x_1 \approx (0.277 9, 0.886 5, 1)^T$. 故精确解为 $\lambda_1 = 5, x_1 = (0.277 8, 0.888 9, 1)^T$.

8.1.2 幂法的几点说明

以上推导过程中做了一些假设, 当这些假设不成立时, 幂法会出现一些复杂情况, 以下简单说明之.

(1) 假设 A 具有完全的特征向量系, 即 A 具有 n 个线性无关的特征向量. 当 A 不具有 n 个线性无关的特征向量时, 幂法不适用. 但事先往往无法判断这一点, 因

此在运用幂法时,如发现不收敛情况,要考虑此种可能.

(2)在(8.3)式中假设 $a_1 \neq 0$,这在选择 μ_0 时也无法判断,但这往往不影响幂法的成功使用. 因为若选 μ_0 ,使 $a_1 = 0$,由于舍入误差的影响,在迭代某一步会产生 μ_k ,它在 x_1 方向上的分量不为零,以后的迭代仍会收敛.

(3)假设了 $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$. 若不具备此条件,可能出现的情况有:

$$\textcircled{1} \lambda_1 = \lambda_2 = \cdots = \lambda_r, |\lambda_1| > |\lambda_{r+1}| \geq |\lambda_{r+2}| \geq \cdots \geq |\lambda_n|;$$

$$\textcircled{2} \lambda_1 = -\lambda_2;$$

$$\textcircled{3} \lambda_1 \text{ 和 } \lambda_2 \text{ 为共轭复数 } \overline{\lambda_1} = \lambda_2.$$

对情况①,归一化幂法(8.4)式仍适用,但选择不同的 μ_0 得到的特征向量 μ_k 是不同的;对情况②和③较复杂,(8.4)式得到的序列不收敛.

8.2 反幂法

8.2.1 反幂法概述

反幂法用于求 A 的按模最小的特征值. 由于 $Ax_i = \lambda_i x_i$, 推得

$$A^{-1}x_i = \frac{1}{\lambda_i}x_i$$

若有 $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{n-1}| > |\lambda_n|$, 则 $\frac{1}{\lambda_n}$ 是 A^{-1} 的按模最大的特征值, 因此只要用幂法求出 A^{-1} 的按模最大的特征值, 也就求出了 A 的按模最小的特征值. 这就是反幂法.

这样,反幂法的迭代公式变为 $\mu_k = A^{-1}y_{k-1}$. 但要先求逆阵 A^{-1} , 计算麻烦. 为了避免求逆阵, 由于 $A\mu_k = y_{k-1}$, 可知每迭代一次, 要解一次方程组, 即对任意初始向量 $\mu_0 \neq 0$, 计算

$$\begin{cases} y_{k-1} = \frac{\mu_{k-1}}{\|\mu_{k-1}\|_p} \\ A\mu_k = y_{k-1} \quad (k=1, 2, \cdots) \end{cases} \quad (8.9)$$

解出 μ_k . 为解方程组, 先把 A 作 LU 分解. 下面是反幂法算法.

反幂法算法

①对 A 进行 LU 分解, 求出 L, U .

②对任意非零向量 $\mu_0, k=1, 2, \dots$, 执行

a) $|a_r| = \max_{1 \leq i \leq n} |a_i|$, a_i 是 $\mu_{k-1} = (a_1, a_2, \dots, a_n)^T$ 的分量;

b) $y_{k-1} = \frac{\mu_{k-1}}{|a_r|}$;

c) 解 $\begin{cases} Lx_{k-1} = y_{k-1} \\ U\mu_k = x_{k-1} \end{cases}$ 得到 u_k ;

d) $t_k = \frac{\text{sgn}(a_r)}{(\mu_k)_r}$;

e) 当 $|t_k - t_{k-1}| < \varepsilon$, 令 $\lambda_n = t_k, x_n = y_{k-1}$, 退出运算; 否则返回 a).

例 8-2 求例 8-1 中矩阵 A 的按模最小特征值及相应的特征向量.

解 把 A 作 LU 分解得

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 6 & -1.5 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 1 & -1 & 2 \\ 0 & -2 & 9 \\ 0 & 0 & 7.5 \end{bmatrix}$$

计算结果如表 8-2 所示.

表 8-2 计算结果

| k | y_{k-1}^T | | | t_k |
|----------|-------------|----------|--------------|----------|
| 1 | 1.000 | 1.000 | 1.000 | -0.555 6 |
| 2 | -0.370 4 | -1.000 | -0.370 06 | -1.626 |
| 3 | 0.582 2 | 1 | -0.092 34 | -0.777 6 |
| \vdots | \vdots | \vdots | \vdots | \vdots |
| 8 | -0.499 2 | -1 | -0.001 283 | -1.003 |
| 9 | 0.500 3 | 1 | -0.000 492 4 | -0.999 0 |
| 10 | -0.500 0 | -1 | -0.000 185 4 | -1.000 |

取 $\lambda_3 \approx t_{10} = -1.000$, 得 $x_3 \approx y_9 = (-0.500\ 3, 1, -0.000\ 492\ 4)^T$. 其精确解为 $\lambda_3 = -1, x = (-0.5, -1, 0)^T$.

8.2.2 原点平移加速

由于幂法的收敛速度主要由比值 $\left| \frac{\lambda_2}{\lambda_1} \right|$ 确定, 当这个比值接近 1, 即 $|\lambda_1|$ 和 $|\lambda_2|$

很接近时,收敛将极慢.针对这一情况,一个加速的方法是原点平移.

设矩阵 $B = A - tI$, t 为选择的平移量, I 为单位阵.

设 A 的特征值为 $\lambda_1, \lambda_2, \dots, \lambda_n$, 则 B 的特征值为 $\lambda_1 - t, \lambda_2 - t, \dots, \lambda_n - t$, 且 A, B 的特征向量相同.

适当选择 t , 使 $\lambda_1 - t$ 仍是 B 的按模最大特征值, 且

$$\left| \frac{\lambda_2 - t}{\lambda_1 - t} \right| < \left| \frac{\lambda_2}{\lambda_1} \right|$$

这时对 B 用幂法求 $\lambda_1 - t$, 比直接对 A 求 λ_1 要快. 而求出 $\lambda_1 - t$ 自然也就求出了 λ_1 .

问题的关键在于如何选择合适的 t , 以便既能较明显提高收敛速度, 又不至于求出的是另一个特征值. 不妨设 A 的特征值都是实数, 且 $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_{n-1} > \lambda_n$. 显然, 不论如何选 t , B 的按模最大特征值要么是 $\lambda_1 - t$, 要么是 $\lambda_n - t$. 这时, 选择 t 应满足

$$\begin{aligned} |\lambda_1 - t| &\geq |\lambda_n - t|; \\ \max \left\{ \frac{|\lambda_2 - t|}{|\lambda_1 - t|}, \frac{|\lambda_n - t|}{|\lambda_1 - t|} \right\} &= \min. \end{aligned}$$

显然, 当 $\lambda_2 - t = -(\lambda_n - t)$, 即 $t = \frac{\lambda_2 + \lambda_n}{2}$ 时, 收敛速度最快. 但是, λ_2 与 λ_n 事先是未知的, 故以上结论只有理论上的意义. 实际上, 若 λ_1 是正值, 取 t 是一个小的正数, 若 λ_1 是负值, 取 t 为一绝对值较小的负数, 常能达到加速的目的, 但不排除求得的是另一特征值的可能. 若 A 的特征值全为实数时, 适当调整选择, 用幂法可以求出特征值的上下界.

在反幂法中也可结合原点平移, 除可能得到加速效果外, 耐心反复调整 t , 可能求得 A 的全部实特征值.

8.2.3 求已知特征值的特征向量

反幂法可求出实矩阵按模最小的特征值, 但反幂法的主要用途是已知某近似特征值后, 求对应的特征向量, 并改进此特征值的精度.

若已知 A 的某一特征值 λ_m 的近似值 λ'_m , 一般说总有

$$|\lambda_m - \lambda'_m| = \min |\lambda_i - \lambda'_m|$$

按原点平移法的思想, 取 $t = \lambda'_m$, 作 $B = A - \lambda'_m I$, 则 B 的按模最小的特征值是 $\lambda_m - \lambda'_m$, 对应的特征向量则与 A 的对应 λ_m 的特征向量相同.

对 B 作反幂法:

取初始向量 z_0 , 作迭代

$$\begin{cases} \mathbf{B}\mathbf{y}_k = \mathbf{z}_{k-1} \\ m_k = \max(\mathbf{y}_k) \\ \mathbf{z}_k = \mathbf{y}_k / m_k \quad (k=1, 2, \dots) \end{cases} \quad (8.10)$$

其中 $\max(\mathbf{y}_k)$ 表示 \mathbf{y}_k 的绝对值最大的分量.

当 $\mathbf{B}\mathbf{y}_k = \mathbf{z}_{k-1}$ 时,也可先对 \mathbf{B} 作 LU 分解,以免重复分解工作. 于是有

$$\lim_{k \rightarrow \infty} m_k = \frac{1}{\lambda_m - \lambda'_m}$$

$$\lim_{k \rightarrow \infty} \mathbf{z}_k = \frac{\mathbf{x}_m}{\max(\mathbf{x}_m)}$$

\mathbf{x}_m 是 λ_m 对应的特征向量. 收敛速度为 $\left| \frac{\lambda_m - \lambda'_m}{\lambda_p - \lambda'_m} \right|$, 下标 p 的选取原则是使

$|\lambda_p - \lambda'_m| = \min_{i \neq m} |\lambda_i - \lambda'_m|$, 这个比值一般很小, 所以迭代收敛很快. 当 k 较大时, 取

$$\mathbf{x}_m \approx \mathbf{z}_k, \lambda_m \approx \lambda'_m + \frac{1}{m_k}$$

反幂法收敛快, 精度高, 是已知特征值的近似值时, 求特征向量的最有效的方法, 常作其他求特征值方法的一个补充方法.

例 8-3 已知

$$\mathbf{A} = \begin{bmatrix} -1 & 2 & 1 \\ 2 & -4 & 1 \\ 1 & 1 & -6 \end{bmatrix}$$

其中一个近似的特征值为 -6.42 (精确值为 $-6.421\ 07$), 求其特征向量并改进此特征值解.

$$\text{解 } \mathbf{B} = \mathbf{A} - (-6.42)\mathbf{I} = \begin{bmatrix} 5.42 & 2 & 1 \\ 2 & 2.42 & 1 \\ 1 & 1 & 0.42 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0.369\ 003 & 1 & 0 \\ 0.184\ 502 & 0.375\ 148 & 1 \end{bmatrix} \begin{bmatrix} 5.42 & 2 & 1 \\ 0 & 1.681\ 993 & 0.630\ 993 \\ 0 & 0 & -0.001\ 218\ 85 \end{bmatrix}$$

$$\text{所以 } \mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0.369\ 003 & 1 & 0 \\ 0.184\ 502 & 0.374\ 148 & 1 \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} 5.42 & 2 & 1 \\ 0 & 1.681\ 993 & 0.630\ 993 \\ 0 & 0 & -0.001\ 218\ 85 \end{bmatrix}$$

取任一个初始向量 $\mathbf{y}_1 = (1, 1, 1)^\top$:

第一次迭代: $\mathbf{L}\mathbf{x}_1 = \mathbf{z}_1$, 得 $\mathbf{x}_1 = (1, 0.630\ 997, 0.578\ 781)^\top$.

$\mathbf{U}\mathbf{y}_2 = \mathbf{x}_1$, 得 $\mathbf{y}_2 = (20.741\ 1, 168.890, -449.197)^\top$.

$$m_2 = -449.197, z_2 = \frac{y_2}{m_2} = (-0.046\ 173\ 7, -0.375\ 982, 1)^T.$$

第二次迭代: $Lx_2 = z_2$, 得 $x_2 = (-0.046\ 173\ 7, -0.358\ 944, 1.141\ 514)^T$.

$Uy_3 = x_2$, 得 $y_3 = (43.218\ 8, 351.130, -936.552)^T$.

$$m_3 = 936.552, z_3 = (-0.046\ 146\ 7, -0.374\ 918, 1)^T.$$

于是特征值

$$\lambda \approx \lambda' + \frac{1}{m_3} = -6.42 + (-0.001\ 067) = -6.421\ 07$$

特征向量是 z_3 , 精度很高, 所有有效数字都是准确的. 可见, 反幂法是求已知特征值的特征向量和改进此特征值的有力工具.

8.3 雅可比(Jacobi)方法

雅可比方法用于求实对称矩阵的全部特征值和对应的特征向量. 要用到线性代数的如下结论:

(1) $A_{n \times n}$ 实对称, 则 A 的特征值全为实数, 它们对应的特征向量线性无关且两两正交.

(2) 相似矩阵具有相同的特征值.

(3) $A_{n \times n}$ 实对称, 则存在正交阵 U , 使 $U^T A U = D$, D 是一个对角阵. 这时, D 的对角元 $\lambda_1, \dots, \lambda_n$ 就是 A 的特征值, U 的第 i 列向量就是 λ_i 对应的特征向量.

雅可比方法就是基于以上原理, 用一系列的正交变换逐步消去 A 的非对角元, 使 A 对角化, 从而求得 A 的全部特征值.

8.3.1 平面旋转矩阵

在平面解析几何中二次曲线的标准化和线性代数的二次型标准化中, 已接触过旋转矩阵:

$$U = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

不难验证它是正交矩阵.

对二阶对称矩阵

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix}$$

$$U^T A U = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} \cos^2 \theta + a_{22} \sin^2 \theta + a_{12} \sin 2\theta & \frac{(a_{22} - a_{11}) \sin 2\theta}{2} + a_{12} \cos 2\theta \\ \frac{(a_{22} - a_{11}) \sin 2\theta}{2} + a_{12} \cos 2\theta & a_{11} \sin^2 \theta + a_{22} \cos^2 \theta - a_{12} \sin 2\theta \end{bmatrix}$$

只要选取 θ 满足 $\frac{(a_{22} - a_{11}) \sin 2\theta}{2} + a_{12} \cos 2\theta = 0$, 就能使上述矩阵变成对角阵,

从而求得 A 的特征值.

从上式解出 θ 应满足:

$$\tan 2\theta = \frac{-2a_{12}}{a_{22} - a_{11}} = \frac{2a_{12}}{a_{11} - a_{22}}$$

A 的特征值为

$$\lambda_1 = a_{11} \cos^2 \theta + a_{22} \sin^2 \theta + a_{12} \sin 2\theta$$

$$\lambda_2 = a_{11} \sin^2 \theta + a_{22} \cos^2 \theta - a_{12} \sin 2\theta$$

对应的特征向量为

$$U_1 = (\cos \theta, \sin \theta)^T$$

$$U_2 = (-\sin \theta, \cos \theta)^T$$

对二阶对称矩阵, 只有一对非对角元素, 所以只需一次旋转就能求得特征值.

当 $n \geq 3$ 时, 情况要复杂一些, 但仍可作如下类似的讨论.

设 A 是 n 阶对称矩阵, 其中非对角元 $a_{pq} \neq 0 (p \neq q)$. 现把它变为零元素, 做变换

$$\begin{cases} x_p = y_p \cos \theta - y_q \sin \theta \\ x_q = y_p \sin \theta + y_q \cos \theta \\ x_k = y_k \quad (k = 1, 2, \dots, n; k \neq p, q) \end{cases}$$

写成矩阵形式为

$$\begin{pmatrix} x_1 \\ \vdots \\ x_p \\ \vdots \\ x_q \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \cos \theta & \cdots & -\sin \theta & \\ & & \vdots & \ddots & \vdots & \\ & & \sin \theta & \cdots & \cos \theta & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_p \\ \vdots \\ y_q \\ \vdots \\ y_n \end{pmatrix} = U_{pq}(\theta) \begin{pmatrix} y_1 \\ \vdots \\ y_p \\ \vdots \\ y_q \\ \vdots \\ y_n \end{pmatrix}$$

记其中的变换矩阵为 $U_{pq}(\theta)$. 容易验证:

$$U_{pq}^T(\theta) U_{pq}(\theta) = I_n$$

记

$$B = U_{pq}^T(\theta) A U_{pq}(\theta) = (b_{ij})_{n \times n}$$

与二阶情况类似,只要适当选择 θ ,可将 \mathbf{A} 中的 a_{pq} 化为零元素,此时有

$$\begin{cases} b_{pj} = a_{pj} \cos \theta + a_{qj} \sin \theta & (j \neq p, q) \\ b_{qj} = -a_{pj} \sin \theta + a_{qj} \cos \theta & (j \neq p, q) \\ b_{ip} = a_{ip} \cos \theta + a_{iq} \sin \theta & (i \neq p, q) \\ b_{iq} = -a_{ip} \sin \theta + a_{iq} \cos \theta & (i \neq p, q) \\ b_{ij} = a_{ij} & (i, j \neq p, q) \end{cases} \quad (8.11)$$

$$\begin{cases} b_{pp} = a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta + 2a_{pq} \sin \theta \cos \theta \\ b_{qq} = a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta + 2a_{pq} \sin \theta \cos \theta \\ b_{pq} = \frac{(a_{qq} - a_{pp}) \sin 2\theta}{2} + a_{pq} \cos 2\theta \end{cases} \quad (8.12)$$

选择 θ , 使 $b_{pq} = 0$, 即

$$\frac{(a_{qq} - a_{pp}) \sin 2\theta}{2} + a_{pq} \cos 2\theta = 0$$

得

$$\tan 2\theta = \frac{2a_{pq}}{a_{pp} - a_{qq}} \quad (8.13)$$

为避免分母为零,同时避开求反三角函数及求 $\sin \theta$ 和 $\cos \theta$ 产生的误差,常采用如下公式直接确定 $\sin \theta$ 和 $\cos \theta$:

$$\begin{cases} \lambda = -a_{pq} \\ \mu = \frac{a_{pp} - a_{qq}}{2} \\ \omega = \frac{\operatorname{sgn}(\mu) \cdot \lambda}{\sqrt{\lambda^2 + \mu^2}} \quad (\mu \neq 0) \\ \omega = 1 \quad (\mu = 0) \end{cases} \quad (8.14)$$

$$\begin{cases} \sin \theta = \frac{\omega}{\sqrt{2(1 + \sqrt{1 - \omega^2})}} \\ \cos \theta = \sqrt{1 - \sin^2 \theta} \end{cases} \quad (8.15)$$

这样进行一次旋转变换,可将某一对非对角元 a_{ij} 化为零.但在进行某次旋转变换时,以往被化为零的非对角元又可能非零.因此,把三阶以上的矩阵 \mathbf{A} 化为对角阵一般不能通过有限次旋转变换实现,而是一个反反复复的过程.下面讨论这个过程的收敛性.

由于 $\mathbf{B} = \mathbf{U}^T \mathbf{A} \mathbf{U}$, 可以证明

$$\sum_{i,j=1}^n b_{ij}^2 = \sum_{i,j=1}^n a_{ij}^2$$

直接从(8.11)式、(8.12)式计算可得

$$\sum_{i=1}^n b_{ii}^2 = \sum_{i=1}^n a_{ii}^2 + 2a_{pq}^2$$

因此

$$\sum_{\substack{i,j=1 \\ i \neq j}}^n b_{ij}^2 = \sum_{\substack{i,j=1 \\ i \neq j}}^n a_{ij}^2 - 2a_{pq}^2 \quad (8.16)$$

(8.16)式说明,每次旋转变换使非对角元的平方和减少 $2a_{pq}^2$. 因此,为了提高收敛速度,每次旋转中 a_{pq} 的确定原则为:选 p, q , 使 $|a_{pq}| = \max_{i \neq j} |a_{ij}|$. 这种方法就是雅可比方法.

8.3.2 雅可比方法

根据上面分析可得到下面雅可比方法.

雅可比算法

①在 A 中找出绝对值最大的非对角元 a_{pq} , 若 $|a_{pq}| < \varepsilon$, ε 为已给出的误差限, 则 A 已近似于对角阵, 退出计算; 反之则进行②.

②用下面公式确定 $\sin\theta$ 和 $\cos\theta$.

$$\begin{cases} \lambda = -a_{pq} \\ \mu = \frac{a_{pp} - a_{qq}}{2} \\ \omega = \frac{\operatorname{sgn}(\mu) \cdot \lambda}{\sqrt{\lambda^2 + \mu^2}} \quad (\mu \neq 0) \\ \omega = 1 \quad (\mu = 0) \end{cases}$$

$$\begin{cases} \sin\theta = \frac{\omega}{\sqrt{2(1 + \sqrt{1 - \omega^2})}} \\ \cos\theta = \sqrt{1 - \sin^2\theta} \end{cases}$$

③用下面公式计算 $b_{i,j}$ ($i, j = 1, 2, \dots, n$), 确定矩阵 B .

$$\begin{cases} b_{pj} = a_{pj} \cos\theta + a_{qj} \sin\theta & (j \neq p, q) \\ b_{qj} = -a_{pj} \sin\theta + a_{qj} \cos\theta & (j \neq p, q) \\ b_{ip} = a_{ip} \cos\theta + a_{iq} \sin\theta & (i \neq p, q) \\ b_{iq} = -a_{ip} \sin\theta + a_{iq} \cos\theta & (i \neq p, q) \\ b_{ij} = a_{ij} & (i, j \neq p, q) \end{cases}$$

$$\begin{cases} b_{pp} = a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta + 2a_{pq} \sin \theta \cos \theta \\ b_{qq} = a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta - 2a_{pq} \sin \theta \cos \theta \\ b_{pq} = \frac{(a_{qq} - a_{pp}) \sin 2\theta}{2} + a_{pq} \cos 2\theta \end{cases}$$

④令 $A = B$, 返回①.

这样通过若干次旋转变换, 就能将 A 化为相似的对角阵, 求得足够精度的特征值 $\lambda_i (i = 1, 2, \dots, n)$.

定理 8.1 雅可比方法是收敛的.

证明 设 $B = U^T A U$, 由 (8.16) 式知

$$\sum_{\substack{i,j=1 \\ i \neq j}}^n b_{ij}^2 = \sum_{\substack{i,j=1 \\ i \neq j}}^n a_{ij}^2 - 2a_{pq}^2$$

因为 $|a_{pq}| = \max_{\substack{i,j=1 \\ i \neq j}} |a_{ij}|$, 所以有

$$\sum_{\substack{i,j=1 \\ i \neq j}}^n a_{ij}^2 \leq n(n-1)a_{pq}^2 \quad \text{即} \quad a_{pq}^2 \geq \frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^n a_{ij}^2$$

所以

$$\sum_{\substack{i,j=1 \\ i \neq j}}^n b_{ij}^2 \leq \left[1 - \frac{2}{n(n-1)} \right] \sum_{\substack{i,j=1 \\ i \neq j}}^n a_{ij}^2$$

即每做一次旋转变换, 非对角元平方和减少 $1 - \frac{2}{n(n-1)}$ 倍. 做 k 次变换, 则减少

$\left[1 - \frac{2}{n(n-1)} \right]^k$ 倍. 当 $k \rightarrow \infty$ 时, 必有非对角元平方和趋近于 0.

8.3.3 过关雅可比方法

雅可比方法每次选取矩阵中绝对值最大的非对角元素作为消去对象, 需在所有非对角元中进行比较选取. 这种比较选取的工作量大, 过关雅可比方法则是一种改进.

取一串正数, 比如常取:

$$V_k = \left[\sum_{\substack{i,j=1 \\ i \neq j}}^n a_{ij}^2 \right]^{\frac{1}{2}} / n^k \quad (k = 1, 2, \dots, r)$$

作为每次比较的阈值, 也称为“关”. 再按顺序检查矩阵的非对角元, 凡绝对值小于 V_k 的就让其过“关”, 不作处理; 凡绝对值大于等于 V_k 的就利用旋转变换使之为零. 当所有非对角元绝对值都小于 V_k 时, 将 V_k 除以 n 作为新的“关”, 重复以上计算直到需要的精度, 即所有非对角元绝对值小于 V_r . 这时, 也得到一个近似对角阵, 其主对角元素就是所求特征值的近似值.

雅可比方法收敛较慢,尤其是对高阶矩阵收敛更慢.它常用于求阶数不高的实对称矩阵的特征值和对应的特征向量.

例 8-4 用雅可比方法求矩阵 A 的全部特征值.

$$A = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 2 & 0.5 \\ 0.5 & 0.5 & 3 \end{bmatrix}$$

解 只列出计算中每步的结果,第五步得到所有对角元的绝对值都小于 0.000 1.

$$\begin{aligned} \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 2 & 0.5 \\ 0.5 & 0.5 & 3 \end{bmatrix} &\xrightarrow[\text{第一步}]{\text{化 } a_{12} \text{ 为 } 0} \begin{bmatrix} 0.792\ 9 & 0 & 0.270\ 6 \\ 0 & 2.207\ 1 & 0.653\ 3 \\ 0.270\ 6 & 0.653\ 3 & 3.000\ 0 \end{bmatrix} \\ &\xrightarrow[\text{第二步}]{\text{化 } a_{23} \text{ 为 } 0} \begin{bmatrix} 0.792\ 9 & -0.132\ 7 & 0.235\ 8 \\ -0.132\ 7 & 1.839\ 4 & 0 \\ 0.235\ 8 & 0 & 3.367\ 7 \end{bmatrix} \\ &\xrightarrow[\text{第三步}]{\text{化 } a_{13} \text{ 为 } 0} \begin{bmatrix} 0.771\ 5 & -0.132\ 2 & 0 \\ -0.132\ 2 & 1.839\ 4 & -0.012\ 0 \\ 0 & -0.012\ 0 & 3.389\ 1 \end{bmatrix} \\ &\xrightarrow[\text{第四步}]{\text{化 } a_{12} \text{ 为 } 0} \begin{bmatrix} 0.755\ 4 & 0 & -0.015\ 0 \\ 0 & 1.855\ 5 & -0.011\ 9 \\ -0.015\ 0 & -0.011\ 9 & 3.389\ 1 \end{bmatrix} \\ &\xrightarrow[\text{第五步}]{\text{化 } a_{13} \text{ 为 } 0} \begin{bmatrix} 0.755\ 4 & 0 & 0 \\ 0 & 1.855\ 4 & 0 \\ 0 & 0 & 3.389\ 2 \end{bmatrix} \end{aligned}$$

结果是:

$$\begin{cases} \lambda_1 = 0.755\ 4 \\ \lambda_2 = 1.855\ 4 \\ \lambda_3 = 3.389\ 2 \end{cases}$$

雅可比方法可同时求得矩阵的特征向量.由

$$A_k = U_k U_{k-1}^T \cdots U_1^T A U_1 U_2 \cdots U_k$$

其中 k 是进行旋转的总次数.记

$$P_k = U_1 U_2 \cdots U_k = (P_{ij})_{n \times n}$$

则

$$A_k = P_k^T A P_k \approx \Lambda$$

Λ 是近似对角阵,它的对角元即特征值近似值.

令 $P_0 = I, P_m = P_{m-1} U_m (m = 1, 2, \cdots, k)$, 每步的算式为:

$$\begin{cases} P_{ip}^{(m)} = P_{ip}^{(m-1)} \cos \theta + P_{iq}^{(m-1)} \sin \theta \\ P_{iq}^{(m)} = -P_{ip}^{(m-1)} \sin \theta + P_{iq}^{(m-1)} \cos \theta \quad (i = 1, 2, \cdots, n) \\ P_{ij}^{(m)} = P_{ij}^{(m-1)} \quad (j \neq p, q) \end{cases}$$

只要将以上算式加入算法的循环中,就可以在求出特征值的同时求出 P_m , P_m 的第 j 列就是 λ_j 的特征向量. 当然,也可以在求出特征值后用反幂法求特征向量.

8.4 QR 算法

QR 算法可用来求任意实的非奇异矩阵的全部特征值,是目前计算这类问题最有效的方法之一.

QR 算法及其理论比较复杂,要用到镜面反射矩阵 (Householder)、矩阵的正交三角分解 ($A = QR$)、矩阵的 Hessenberg 型以及任何实方阵可通过正交相似变换化为 Hessenberg 型等. 下面作简单介绍.

8.4.1 豪斯豪德尔 (Householder) 矩阵

定义 8.1 设 v 是 n 维向量,而且 $v^T v = I$,称 $H = I - 2vv^T$ 为 Householder 矩阵.

因为

$$H^T = (I - 2vv^T)^T = I - 2vv^T$$

$$HH^T = HH = (I - 2vv^T)(I - 2vv^T) = I - 4v^T + 4v(vv^T)v^T = I$$

所以, H 是对称正交阵.

定理 8.2 设 $x = w + u$, 其中 $u = cv$, $w^T v = 0$, $H = I - 2vv^T$, 则 $Hx = w - u$.

这里只给出定理 8.2 的几何意义: 将任一非零向量 x 分解为 u 和 w , w 和 u 垂直. 在 u 上取单位向量 v , 用 Q 表示与 v 垂直的所有向量的集合, Q 是 $n-1$ 维子空间, 用 v 构造 Householder 矩阵 H , Hx 恰好是 x 关于“镜面” Q 的像. 当 $n=3$ 时, 如图 8-1 所示.

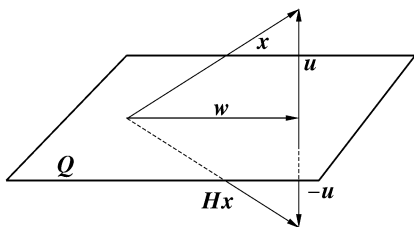


图 8-1

实际上, 线性变换 H 是一种镜像变换, 它不改变 x 的长度, 只要适当选择“镜面” Q , 就可以把 x 调整到任一方向.

如 x 是任一 n 维非零向量, y 为任一 n 维单位向量, 要使 Hx 平行于 y , 可按下列方式选择 Q 的法向量 v ($\|v\|_2 = 1$), 从而构造出 H 矩阵:

$$\text{令} \quad |a| = \|x\|_2 \quad (8.17)$$

$$\rho = \|x - ay\|_2 \quad (8.18)$$

$$v = \frac{(x - ay)}{\rho} \quad (8.19)$$

显然 \mathbf{v} 是 n 维单位向量. 取

$$\mathbf{H} = \mathbf{I} - 2\mathbf{v}\mathbf{v}^T \quad (8.20)$$

这时 $\mathbf{H}\mathbf{x} = (\mathbf{I} - 2\mathbf{v}\mathbf{v}^T)\mathbf{x} = \mathbf{x} - \frac{2}{\rho}\mathbf{v}(\mathbf{x} - a\mathbf{y})^T\mathbf{x} = \mathbf{x} - \frac{2}{\rho}\mathbf{v}(a^2 - a\mathbf{y}^T\mathbf{x})$

注意到

$$\rho^2 = (\mathbf{x}^T - a\mathbf{y}^T)(\mathbf{x} - a\mathbf{y}) = 2(a^2 - a\mathbf{y}^T\mathbf{x}) \quad (8.21)$$

所以 $\mathbf{H}\mathbf{x} = \mathbf{x} - \rho\mathbf{v} = a\mathbf{y}$. 达到了目的.

计算过程中有 a 的符号确定问题. (8.21) 式中, 为避免减法运算 (防止有效数位损失), 当 $\mathbf{y}^T\mathbf{x} \neq 0$ 时, 取 $a = -\text{sgn}(\mathbf{y}^T\mathbf{x}) \|\mathbf{x}\|_2$; 当 $\mathbf{y}^T\mathbf{x} = 0$ 时, 取 $a = \|\mathbf{x}\|_2$.

用 QR 算法求矩阵特征值时, Householder 矩阵有两个重要作用: 一是对 \mathbf{A} 作正交相似变换, 把 \mathbf{A} 化为拟上三角矩阵 (Hessenberg); 二是对矩阵作正交三角分解.

8.4.2 化一般矩阵为拟上三角矩阵

拟上三角矩阵也称为 Hessenberg 矩阵, 其形式为

$$\begin{bmatrix} * & * & \cdots & * \\ * & * & \cdots & * \\ & \ddots & \ddots & \vdots \\ & & * & * \end{bmatrix}$$

设 $\mathbf{A} = (a_{ij}^{(1)})$ 是 $n \times n$ 实矩阵, 取 $\mathbf{x} = (0, a_{21}^{(1)}, \dots, a_{n1}^{(1)})^T$, $\mathbf{y} = (0, 1, 0, \dots, 0)^T$, 记 $a_1 = a$, 按公式 (8.17) ~ (8.20) 形成 Householder 矩阵 \mathbf{H}_1

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & * & \cdots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \cdots & * \end{bmatrix}$$

$\mathbf{H}_1\mathbf{A}$ 的第一列为

$$\mathbf{H}_1 \begin{bmatrix} a_{11}^{(1)} \\ a_{21}^{(1)} \\ \vdots \\ a_{n1}^{(1)} \end{bmatrix} = \mathbf{H}_1\mathbf{x} + \mathbf{H}_1 \begin{bmatrix} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = a_1\mathbf{y} + \begin{bmatrix} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} \\ a_1 \\ \vdots \\ 0 \end{bmatrix}$$

又因为用 \mathbf{H}_1 右乘一个矩阵不改变后者的第一列, 于是

$$\mathbf{A}_2 = \mathbf{H}_1\mathbf{A}\mathbf{H}_1 = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ a_1 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}$$

又取 $\mathbf{x} = (0, 0, a_{32}^{(2)}, \dots, a_{n2}^{(2)})^T, \mathbf{y} = (0, 0, 1, 0, \dots, 0)^T$, 记 $a_2 = a$, 形成 \mathbf{H}_2 :

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix}$$

$\mathbf{H}_2 \mathbf{A}_2$ 的第一列同 \mathbf{A}_2 第一列, 而 $\mathbf{H}_2 \mathbf{A}_2$ 第二列变为

$$\mathbf{H}_2 \mathbf{x} + \mathbf{H}_2 \begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = a_2 \mathbf{y} + \begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ a_2 \\ \vdots \\ 0 \end{bmatrix}$$

而用 \mathbf{H}_2 右乘一个矩阵不改变它的第一、第二列. 所以

$$\mathbf{A}_3 = \mathbf{H}_2 \mathbf{A} \mathbf{H}_2 = \begin{bmatrix} * & * & * & \cdots & * \\ a_1 & * & * & \cdots & * \\ 0 & a_2 & * & \cdots & * \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix}$$

如此下去, 作 $n-2$ 次变换后, \mathbf{A} 变化为拟上三角矩阵 \mathbf{A}_{n-1} :

$$\begin{aligned} \mathbf{A}_{n-1} &= \mathbf{H}_{n-2} \mathbf{H}_{n-3} \cdots \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1 \mathbf{H}_2 \cdots \mathbf{H}_{n-3} \mathbf{H}_{n-2} \\ &= \begin{bmatrix} * & * & * & * & \cdots & * \\ a_1 & * & * & * & \cdots & * \\ & a_2 & * & * & \cdots & * \\ & & a_3 & * & \cdots & * \\ & & & \ddots & \ddots & \vdots \\ & & & & a_{n-1} & * \end{bmatrix} \end{aligned}$$

如果 \mathbf{A} 是对称阵, 则 \mathbf{A}_{n-1} 仍是对称矩阵. 因此, \mathbf{A}_{n-1} 将是对称三对角矩阵

$$\mathbf{A}_{n-1} = \begin{bmatrix} * & a_1 & & & & \\ a_1 & * & a_2 & & & \\ & a_2 & * & a_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & \\ & & & & & a_{n-1} & * \end{bmatrix}$$

A 和 A_{n-1} 相似, 具有相同特征值.

因此, 任何实矩阵都可以通过正交相似变换化为 Hessenberg 形. QR 算法前, 经常先把矩阵 A 化成 Hessenberg 矩阵, 然后再用 QR 算法可加速收敛.

8.4.3 矩阵的正交三角分解

设 $A^{(1)} = A = (a_{ij}^{(1)})$ 是 $n \times n$ 实矩阵, 取 $x = (a_{11}^{(1)}, a_{21}^{(1)}, \dots, a_{n1}^{(1)})^T, y = (1, 0, \dots, 0)^T$, 并记 $a_1 = a$, 按公式 (8.17) ~ (8.20) 形成 Householder 矩阵 H_1 . 则

$$A^{(2)} = H_1 A^{(1)} = \begin{pmatrix} a_1 & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}$$

又取 $x = (0, a_{22}^{(2)}, \dots, a_{n2}^{(2)})^T, y = (0, 1, \dots, 0)^T$, 记 $a_2 = a$, 构造 H_2 . 则

$$A^{(3)} = H_2 A^{(2)} = \begin{pmatrix} a_1 & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_2 & a_{23}^{(3)} & \cdots & a_{2n}^{(3)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{pmatrix}$$

作 $n-1$ 次变换后, A 被化为上三角阵 $A^{(n)}$

$$A^{(n)} = H_{n-1} H_{n-2} \cdots H_1 A = \begin{pmatrix} a_1 & * & * & \cdots & * \\ & a_2 & * & \cdots & * \\ & & \ddots & \ddots & \vdots \\ & & & & * \\ 0 & & & & a_n \end{pmatrix}$$

令 $Q = H_1 H_2 \cdots H_{n-1}$, 记 $R = A^{(n)}$, 有 $A = QR$. 因 Q 是正交阵的乘积, 它也是正交阵, R 是上三角矩阵, 这种分解称为 A 的正交三角分解, 简称 QR 分解. 对任何实的非奇异矩阵都可以分解为正交矩阵 Q 和上三角矩阵 R 的乘积, 而且当 R 的对角线元素符号取定时, 分解是惟一的.

8.4.4 QR 算法

令 $A = A_1$, 对 A_1 进行三角正交分解, 分解为正交矩阵 Q_1 和上三角矩阵 R_1 的乘积

$$A_1 = Q_1 R_1$$

然后将得到的因式矩阵 Q_1 与 R_1 逆序相乘,

$$\mathbf{A}_2 = \mathbf{R}_1 \mathbf{Q}_1$$

这就完成了 QR 算法的第一步,以 \mathbf{A}_2 代替 \mathbf{A}_1 ,重复上述步骤即可得出 \mathbf{A}_3, \dots 如此类推,则 QR 算法的计算公式为

QR 算法

$$\begin{cases} \mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k \\ \mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k = \mathbf{Q}_{k+1} \mathbf{R}_{k+1} \end{cases} \quad (k=1, 2, \dots) \quad (8.22)$$

QR 算法产生一个矩阵序列 $\{\mathbf{A}_k\}$, 这个矩阵序列有两个基本性质:

(1) 矩阵序列 $\{\mathbf{A}_k\}$ 中的每一个矩阵 \mathbf{A}_k 都与矩阵 \mathbf{A} 相似, 因为

$$\begin{aligned} \mathbf{A}_{k+1} &= \mathbf{R}_k \mathbf{Q}_k = \mathbf{Q}_k^{-1} \mathbf{A}_k \mathbf{Q}_k \\ &= \mathbf{Q}_k^{-1} \mathbf{Q}_{k-1}^{-1} \mathbf{A}_{k-1} \mathbf{Q}_{k-1} \mathbf{Q}_k = \dots = \mathbf{Q}_k^{-1} \mathbf{Q}_{k-1}^{-1} \dots \mathbf{Q}_1^{-1} \mathbf{A}_1 \mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_k \end{aligned}$$

若令 $\mathbf{E}_k = \mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_k$, 则上式可写成

$$\mathbf{A}_{k+1} = \mathbf{E}_k^{-1} \mathbf{A}_1 \mathbf{E}_k \quad (8.23)$$

因此, \mathbf{A}_k 与 \mathbf{A} 相似, \mathbf{A}_k 与 \mathbf{A} 有完全相同的特征值.

(2) 若令 $\mathbf{H}_k = \mathbf{R}_k \mathbf{R}_{k-1} \dots \mathbf{R}_1$, 则有

$$\begin{aligned} \mathbf{E}_k \mathbf{H}_k &= \mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_k \quad \mathbf{R}_k \mathbf{R}_{k-1} \dots \mathbf{R}_1 = \mathbf{E}_{k-1} \mathbf{A}_k \mathbf{H}_{k-1} \\ &= \mathbf{A}_1 \mathbf{E}_{k-1} \mathbf{H}_{k-1} = \mathbf{A}_1^2 \mathbf{E}_{k-2} \mathbf{H}_{k-2} \dots = \mathbf{A}_1^k \end{aligned} \quad (8.24)$$

这样就得到了 \mathbf{A}_1^k 的 QR 分解, 因为 $\mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_k$ 是正交矩阵, 所以 \mathbf{E}_k 也是正交矩阵, 同样 \mathbf{H}_k 也是上三角矩阵. 从分解的惟一性可知, \mathbf{E}_k 与 \mathbf{H}_k 必定就是 \mathbf{A}_1^k 的分解因子.

8.4.5 QR 算法的收敛性

为了求 \mathbf{A} 的特征值, 要求序列 $\{\mathbf{A}_k\}$ 能收敛于一种简单形式的矩阵, 例如三角型矩阵(或分块三角型矩阵), 其对角线元素(或块)有确定的极限即可. 因此约定, 只要 $\{\mathbf{A}_k\}$ 收敛于三角型矩阵(或分块三角型矩阵), 且对角线元素(或子块)有确定极限, 无论其对角线以外的元素是否有确定极限, 都称为方法是收敛的(或本质收敛). 下面对最简单的情况给出定理.

定理 8.3 假设

① $\mathbf{A}_1 = \mathbf{A} = \mathbf{X} \mathbf{D} \mathbf{X}^{-1}$, $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$;

② 矩阵 \mathbf{A} 的 n 个特征值满足条件 $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| > 0$;

③ 矩阵 $\mathbf{Y} = \mathbf{X}^{-1}$ 有三角分解式, 即 $\mathbf{Y} = \mathbf{L}_y \mathbf{U}_y$, 其中 \mathbf{L}_y 是单位下三角矩阵, \mathbf{U}_y 是上三角矩阵.

则阵列序列 $\{\mathbf{A}_k\}$ 本质收敛.

关于定理的证明和更一般的情况,请参考相关文献.

例 8-5 求矩阵 A 的特征值.

$$A = \begin{bmatrix} 5 & -2 & -5 & -1 \\ 1 & 0 & -3 & 2 \\ 0 & 2 & 2 & -3 \\ 0 & 0 & 1 & -2 \end{bmatrix}$$

解 A 的特征方程为

$$\lambda^4 - 5\lambda^3 + 7\lambda^2 - 7\lambda - 20 = 0$$

其特征值为 $-1, 4, 1 \pm 2i$.

现在用 QR 算法来求其特征值. 令 $A = A_1$ 并进行正交分解得

$$\begin{aligned} A &= A_1 = Q_1 R_1 \\ &= \begin{bmatrix} 0.9806 & -0.0377 & 0.1923 & -0.1038 \\ 0.1961 & 0.1887 & -0.8804 & -0.4192 \\ 0 & 0.9813 & 0.1761 & 0.0740 \\ 0 & 0 & 0.3962 & -0.8989 \end{bmatrix} \\ &\quad \begin{bmatrix} 5.0992 & -1.9612 & -5.4912 & -0.3922 \\ 0 & 2.0381 & 1.5852 & -2.5288 \\ 0 & 0 & 2.5242 & -3.2736 \\ 0 & 0 & 0 & 0.7822 \end{bmatrix} \end{aligned}$$

然后,将求得的 Q_1 和 R_1 逆序相乘,求出 A_2 :

$$A_2 = \begin{bmatrix} 4.6157 & 5.9508 & 1.5922 & 0.2390 \\ 0.3997 & 1.9401 & -2.5171 & 1.5361 \\ 0 & 2.4770 & -0.8525 & 3.1294 \\ 0 & 0 & 0.3099 & -0.7031 \end{bmatrix}$$

把 A_2 代替 A_1 重复上面过程,计算 11 次后得

$$A_{12} = \begin{bmatrix} 4.0000 & * & * & * \\ & 1.8789 & -3.5910 & * \\ & 1.3290 & 0.1211 & * \\ & & & -1.0000 \end{bmatrix}$$

至此,不难看出 A 的一个特征值是 4,另一个特征值是 -1 ,其他两个特征值是方程

$$\begin{vmatrix} 1.8789 - \lambda & -3.5910 \\ 1.3290 & 0.1211 - \lambda \end{vmatrix} = 0$$

的根,求得 λ 为 $1 \pm 2i$.

最后说明一下,QR 算法的收敛速度是线性的,这在实际计算时很不理想,而

且 QR 算法的工作量较大. 因此, 使用时经常先把矩阵 A 化成 Hessenberg 矩阵, 然后再用 QR 算法加速收敛. 关于这些问题的讨论这里不再详述, 请参考相关文献.

练习与思考

1. 用幂法求矩阵

$$A = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & -3 & 2 \\ 4 & 4 & -1 \\ 6 & 3 & 5 \end{bmatrix}$$

按模最大的特征值和对应的特征向量, 精确到小数点后三位.

2. 已知 A (取 $t=0.6$), B (取 $t=15$), 作原点平移的幂法, 求按模最大的特征值.

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} -11 & 11 & 1 \\ 11 & 9 & -2 \\ 1 & -2 & 13 \end{bmatrix}$$

3. 已知 A , 用反幂法求最接近 12 的特征值与相应的特征向量, 迭代两次.

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

4. 若 A 的特征值为 $\lambda_1, \lambda_2, \dots, \lambda_n$, t 是一实数. 证明: $\lambda_i - t$ 是 $A - tI$ 的特征值, 且特征向量不变.

5. 已知 $y = (3, 2, 2, 1)^T$, 求平面反射阵 H 使 $Hy = (*, *, 0, 0)^T$, 即把 y 的后两个元素化为零.

6. 已知 A , 试用雅可比旋转法作一次旋转, 消去最大的非对角元, 写出旋转矩阵, 求出 θ 角和结果.

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 3 & 1 \\ 2 & 1 & 6 \end{bmatrix}$$

7. 设 $T = \begin{bmatrix} T_1^{(3 \times 3)} & 0^{(3 \times 2)} \\ 0^{(2 \times 3)} & T_2^{(2 \times 2)} \end{bmatrix}$, 已知 λ 是 T_1 的特征值, 相应的特征向量为

$(a_1, a_2, a_3)^T$. 证明 λ 也是 T 的特征值, 相应的特征向量为 $(a_1, a_2, a_3, 0, 0)^T$.

8. 证明 (8.22) 式中的 A_k 和 A_{k+1} 相似.

9. 求矩阵 A, B 的 QR 分解.

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

9 常微分方程初值问题的数值解法

科学研究和工程技术中的问题往往归结为求某个常微分方程的定解问题.

常微分方程的理论指出,很多方程的定解虽然存在,但可能十分复杂,难于计算,也可能不能用简单的初等函数表示.因此,更常碰到的是要求满足一定精度的近似解.

常微分方程数值解法的特点是:对求解区间进行剖分,然后把微分方程离散成在节点上的近似公式或近似方程,最后结合定解条件求出近似解.因此,数值解法得到的近似解是一个离散的函数表.

一阶常微分方程的初值问题是:

$$\begin{cases} y' = f(x, y) & (a \leq x \leq b) \\ y(a) = y_0 \end{cases}$$

其中 $f(x, y)$ 是已知函数; $y(a) = y_0$ 是定解条件.

由常微分方程的理论,我们知当 $f(x, y)$ 定义在区域 $G = (a \leq x \leq b, |y| < \infty)$, 若存在正的常数 L , 使

$$|f(x, y_1) - f(x, y_2)| \leq L |y_1 - y_2|$$

对任意的 $x \in [a, b]$ 和任意的 y_1, y_2 成立, 则称 $f(x, y)$ 对 y 满足李普希兹 (Lipschitz) 条件. L 称李普希兹常数.

若 $f(x, y)$ 在区域 G 连续, 关于 y 满足李普希兹条件, 则上述一阶常微分方程的初值问题的解存在且惟一.

9.1 欧拉 (Euler) 法

解一阶常微分方程的初值问题

$$\begin{cases} y' = f(x, y) & (a \leq x \leq b) \\ y(a) = y_0 \end{cases} \quad \begin{matrix} (9.1) \\ (9.2) \end{matrix}$$

最简单而直观的方法是欧拉法. 欧拉法在精度要求不高时, 是一个实用的方法, 而且通过欧拉法的讨论, 容易弄清常微分方程初值问题数值解法的一些基本概念和构造方法的思路.

9.1.1 欧拉法的导出

把区间 $[a, b]$ 分为 n 个小区间, 取步长 $h = \frac{b-a}{n}$, 节点 $x_i = x_0 + ih (i = 0, 1, 2,$

$\cdots, n)$, 其中 $x_0 = a$. 又设 $y(x)$ 为上述问题的解.

把(9.1)式两端积分, 积分区间取 $[x_i, x_{i+1}]$, 得

$$\begin{aligned} \int_{x_i}^{x_{i+1}} y' dx &= \int_{x_i}^{x_{i+1}} f(x, y) dx \\ y(x_{i+1}) - y(x_i) &= \int_{x_i}^{x_{i+1}} f(x, y) dx \end{aligned} \quad (9.3)$$

对(9.3)式右端积分用左矩形积分公式得

$$\int_{x_i}^{x_{i+1}} f(x, y) dx = hf(x_i, y(x_i)) + \frac{h^2}{2} y''(\xi) \quad \xi \in (x_i, x_{i+1})$$

代入(9.3)式得

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

舍去 $\frac{h^2}{2} y''(\xi)$, 并令 $y_i = y(x_i)$, 得到

$$y_{i+1} = y_i + hf(x_i, y_i) \quad (i=0, 1, \cdots, n-1) \quad (9.4)$$

这就是欧拉公式.

欧拉公式还可由泰勒展开得到. 假定 $y(x)$ 二阶连续可导, 把 $y(x_{i+1})$ 在 x_i 点展开, 得

$$y(x_{i+1}) = y(x_i) + y'(x_i)h + \frac{y''(\xi)}{2!}h^2$$

舍去 h^2 项, 并令 $y_i = y(x_i)$, 注意到 $y'(x_i) = f(x_i, y_i)$, 得

$$y_{i+1} = y_i + hf(x_i, y_i) \quad (i=0, 1, \cdots, n-1)$$

9.1.2 欧拉隐式公式和欧拉中点公式

对(9.3)式的右端积分用右矩形积分公式代替, 即

$$y(x_{i+1}) - y(x_i) = hf(x_{i+1}, y(x_{i+1})) - \frac{h^2}{2} y''(\xi) \quad \xi \in (x_i, x_{i+1})$$

$i=0, 1, 2, \cdots, n-1$. 可得欧拉隐式公式:

$$y_{i+1} = y_i + hf(x_{i+1}, y_{i+1}) \quad (9.5)$$

不难看出, (9.5)式两端都含有 y_{i+1} , 一般情况下不能由 y_i 的值直接计算 y_{i+1} , 而需要解非线性方程, 故称为欧拉隐式公式.

把(9.3)式积分区间改为 $[x_{i-1}, x_{i+1}]$, 并对(9.3)式右端积分, 用矩形积分公式代替, 即

$$y(x_{i+1}) - y(x_{i-1}) = 2hf(x_i, y(x_i)) + \frac{h^3}{3} y'''(\xi) \quad \xi \in [x_{i-1}, x_{i+1}]$$

可得到欧拉中点公式:

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i) \quad (9.6)$$

容易看出,用中点公式计算 y_{i+1} 时,不仅需要 y_i 的值,还需要 y_{i-1} 的值.

凡是能从已知(或已算出)的 y_0, y_1, \dots, y_i 直接算出 y_{i+1} 的公式称显式公式,如(9.4)式、(9.6)式;否则称隐式公式,如(9.5)式.

在计算 y_{i+1} 时,只需要 y_i 的值,则称公式为单步法;若除 y_i 之外还需要以前的 y_{i-1} 等多个值,则称多步法. 公式(9.4)、(9.5)是单步法;公式(9.6)是多步法,确切地说是二步法.

9.1.3 局部截断误差和方法的阶

假定 y_i, y_{i-1} 的值没有误差,计算 y_{i+1} 时产生的误差称为局部截断误差,也称为单步截断误差. 局部截断误差表示为

$$R = |y(x_{i+1}) - y_{i+1}| \quad (9.7)$$

对欧拉公式:

$$R_1 = |y(x_{i+1}) - y(x_i) - hf(x_i, y(x_i))|$$

注意到用泰勒展开式推导欧拉公式的过程,可知

$$R_1 = \left| \frac{y''(\xi)}{2} h^2 \right| = O(h^2)$$

同样可以推出,欧拉隐式公式的局部截断误差为

$$R_2 = \left| \frac{y''(\xi)}{2} h^2 \right| = O(h^2)$$

中点公式的局部截断误差为:

$$R_3 = \left| \frac{y'''(\xi)}{3} h^3 \right| = O(h^3)$$

凡是局部截断误差为 $O(h^{p+1})$ 的方法称为 p 阶方法. 这样欧拉方法和欧拉隐式方法称为一阶方法,而欧拉中点方法称为二阶方法. 一般来说,高阶方法比低阶方法精确.

9.1.4 梯形公式及其预报校正法

对公式(9.3)的右端积分用梯形积分公式近似代替,得

$$y(x_{i+1}) - y(x_i) = \frac{h}{2} [f(x_i, y(x_i)) + f(x_{i+1}, y(x_{i+1}))] - \frac{h^3 y'''(\xi)}{12} \quad \xi \in [x_i, x_{i+1}]$$

$i=0, 1, \dots, n-1$. 把 $y(x_{i+1})$ 记为 y_{i+1} , 并略去余项 $-\frac{h^3 y'''(\xi)}{12}$, 得到梯形公式:

$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \quad (9.8)$$

显然它是单步隐式方法, 它的局部截断误差为 $O(h^3)$, 因此它也是一个二阶方法.

用隐式方法求 y_{i+1} 需要解方程, 可用迭代法解此方程. 初值 $y_{i+1}^{(0)}$ 由显式的欧拉公式给出. 计算公式为

$$\begin{aligned} y_{i+1}^{(0)} &= y_i + hf(x_i, y_i) \\ y_{i+1}^{(k+1)} &= y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k)})] \quad (k=0, 1, 2, \dots) \end{aligned} \quad (9.9)$$

迭代的收敛与否与步长 h 有关, 这里迭代函数是

$$\begin{aligned} \varphi(y) &= y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \\ \varphi'(y) &= \frac{h}{2} f_y(x, y) \end{aligned}$$

若 $M = \max_{a \leq x \leq b} |f_y(x, y)|$, 则 $|\varphi'(y)| \leq \frac{h}{2} M < 1, h < \frac{2}{M}$. 所以, 要得到收敛的迭代序列 $y_{i+1}^{(0)}, y_{i+1}^{(1)}, \dots, y_{i+1}^{(k)}, \dots$, h 要适当地小. 如迭代不收敛, 则应减小步长 h 重新计算.

实际上, 常采用预报校正技术, 即用显式欧拉公式预报出 y_{i+1} 的初值 $y_{i+1}^{(0)}$, 再用梯形公式进行校正得到 y_{i+1} . 梯形公式的预报校正法如下:

$$\begin{aligned} y_{i+1}^{(0)} &= y_i + hf(x_i, y_i) \\ y_{i+1} &= y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(0)})] \end{aligned}$$

上述预报校正公式等价于显式公式:

$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i))] \quad (9.10)$$

梯形公式的预报校正法通常也称为改进的欧拉法.

例 9-1 用欧拉方法、隐式欧拉方法和欧拉中点公式计算

$$\begin{cases} y' = -y & (0 \leq x \leq 1) \\ y(0) = 1 \end{cases}$$

的近似解, 取步长 $h=0.1$, 并与精确值比较.

解 欧拉方法的算式为

$$\begin{cases} y_{i+1} = y_i(1-h) \\ y_0 = 1 \end{cases}$$

$i=0, 1, \dots, 9$. 欧拉隐式方法在本题可解出方程, 不必迭代, 公式为

$$\begin{cases} y_{i+1} = \frac{y_i}{(1+h)} \\ y_0 = 1 \end{cases}$$

欧拉中点公式是两步法, 第一步 y_1 用欧拉公式的结果, 以后用公式

$$\begin{cases} y_{i+1} = y_{i-1} - 2hy_i \\ y_0 = 1 \end{cases}$$

本题精确解为 $y = e^{-x}$. 计算结果如表 9-1 所示.

表 9-1 计算结果

| x_i | $y(x_i) = e^{-x_i}$ | 欧拉方法 | 欧拉隐式法 | 欧拉中点公式 |
|-------|---------------------|-----------|-----------|-----------|
| 0.1 | 0.904 837 | 0.900 000 | 0.909 091 | 0.900 000 |
| 0.2 | 0.818 731 | 0.810 000 | 0.826 446 | 0.820 000 |
| 0.3 | 0.740 818 | 0.729 000 | 0.751 315 | 0.736 000 |
| 0.4 | 0.670 320 | 0.656 100 | 0.683 013 | 0.672 800 |
| 0.5 | 0.606 531 | 0.590 490 | 0.620 921 | 0.601 440 |
| 0.6 | 0.548 812 | 0.531 441 | 0.564 474 | 0.552 512 |
| 0.7 | 0.496 585 | 0.478 297 | 0.513 158 | 0.490 938 |
| 0.8 | 0.449 329 | 0.430 467 | 0.466 507 | 0.454 324 |
| 0.9 | 0.406 570 | 0.387 420 | 0.424 098 | 0.400 073 |
| 1.0 | 0.367 879 | 0.348 678 | 0.385 543 | 0.374 310 |

例 9-2 用欧拉公式和梯形公式的预报校正法计算

$$\begin{cases} y' = y - \frac{2x}{y} & (0 \leq x \leq 1) \\ y(0) = 1 \end{cases}$$

的数值解, 取 $h = 0.1$, 并与精确值比较. 方程的解析解为 $y = \sqrt{1+2x}$.

解 由于 $y_{i+1} = y_i + 0.1 \times \left[y_i - \frac{2x_i}{y_i} \right] = 1.1y_i - \frac{0.2x_i}{y_i}$

故欧拉公式为

$$\begin{cases} y_{i+1} = 1.1y_i - \frac{0.2x_i}{y_i} & (i=0, 1, 2, \dots, 9) \\ y_0 = 1 \end{cases}$$

梯形公式的预报校正公式为

$$\begin{cases} y_{i+1}^{(0)} = 1.1y_i - \frac{0.2x_i}{y_i} \\ y_{i+1} = y_i + 0.05 \times \left[y_i - \frac{2x_i}{y_i} + y_{i+1}^{(0)} - \frac{2x_{i+1}}{y_{i+1}^{(0)}} \right] & (i=0, 1, 2, \dots, 9) \\ y_0 = 1 & (x_0 = 0) \end{cases}$$

计算结果如表 9-2 所示.

表 9-2 计算结果

| x_i | 欧拉方法 | 梯形法 | 精确值 |
|-------|------------|------------|------------|
| 0. 1 | 1. 100 000 | 1. 095 909 | 1. 095 445 |
| 0. 2 | 1. 191 818 | 1. 184 097 | 1. 183 216 |
| 0. 3 | 1. 277 438 | 1. 266 201 | 1. 264 911 |
| 0. 4 | 1. 358 213 | 1. 343 360 | 1. 341 641 |
| 0. 5 | 1. 435 133 | 1. 416 402 | 1. 414 214 |
| 0. 6 | 1. 508 966 | 1. 485 956 | 1. 483 240 |
| 0. 7 | 1. 580 338 | 1. 552 515 | 1. 549 193 |
| 0. 8 | 1. 649 783 | 1. 616 475 | 1. 612 452 |
| 0. 9 | 1. 717 779 | 1. 678 167 | 1. 673 320 |
| 1. 0 | 1. 784 771 | 1. 737 868 | 1. 732 051 |

9.2 龙格 – 库塔(Runge – Kutta) 法

欧拉方法是显式的一步法,使用方便,但精度较低. 本节将构造出高精度的显式一步法:龙格 – 库塔法,简称 R – K 法.

9.2.1 二阶 R – K 法

欧拉法的公式为

$$y_{i+1} = y_i + hf(x_i, y_i) \quad (i = 0, 1, 2, \cdots, n - 1)$$

决定其精度的是函数 $f(x_i, y_i)$, 如能改进这个函数, 就可能提高公式的精度. 为此, 把公式改写成

$$y_{i+1} = y_i + h\varphi(x_i, y_i, h) \quad (i = 0, 1, 2, \cdots, n - 1) \tag{9.11}$$

选择函数 $\varphi(x_i, y_i, h)$, 其中一种方法是用若干个点的函数值的线性组合代替 $\varphi(x_i, y_i, h)$, 如

$$\left\{ \begin{array}{l} \varphi(x_i, y_i, h) = \sum_{j=1}^p c_j K_j \\ K_1 = f(x_i, y_i) \\ \vdots \\ K_j = f(x_i + a_j h, y_i + h \sum_{l=1}^{j-1} b_{jl} K_l) \quad (j = 2, 3, \cdots, p) \\ a_j = \sum_{l=1}^{j-1} b_{jl} \quad (j = 2, 3, \cdots, p) \end{array} \right.$$

其中 c_j, a_j, b_{jl} 是待定参数. a_j 和 b_{jl} 满足 $a_j = \sum_{l=1}^{j-1} b_{jl}$.

以上方法称为 p 级 R-K 法. 适当选择 c_j, a_j 和 b_{jl} 可使以上方法为 p 阶方法. 显然, 欧拉法就是一阶 R-K 法.

二级 R-K 法的形式是

$$\left\{ \begin{array}{l} y_{i+1} = y_i + h(c_1 K_1 + c_2 K_2) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + a_2 h, y_i + a_2 h K_1) \end{array} \right.$$

此时,

$$\begin{aligned} \varphi(x_i, y_i, h) &= c_1 f(x_i, y(x_i)) + c_2 f(x_i + a_2 h, y(x_i) + a_2 h f(x_i, y(x_i))) \\ &= c_1 y'(x_i) + c_2 f(x_i + a_2 h, y(x_i) + a_2 h y'(x_i)) \end{aligned} \quad (9.12)$$

由二元函数的泰勒展开:

$$f(x_i + a_2 h, y(x_i) + a_2 h y'(x_i)) = f(x_i, y(x_i)) + a_2 h f'_x + a_2 h y'(x_i) f'_y + O(h^2)$$

其中, 所有的偏导数都是它们在点 $(x_i, y(x_i))$ 的值, 下同.

又由于

$$\begin{aligned} y'(x) &= f(x, y(x)) \\ y''(x) &= f'_x + f'_y y'(x) \end{aligned}$$

所以 $f(x_i + a_2 h, y(x_i) + a_2 h y'(x_i)) = y'(x_i) + a_2 h y''(x_i) + O(h^2)$

代入 (9.12) 式, 得

$$\varphi(x_i, y_i, h) = (c_1 + c_2) y'(x_i) + a_2 c_2 h y''(x_i) + O(h^2)$$

代入 (9.11) 式, 得

$$y_{i+1} = y(x_i) + (c_1 + c_2) h y'(x_i) + a_2 c_2 h^2 y''(x_i) + O(h^3)$$

而由 Taylor 展开式

$$y(x_{i+1}) = y(x_i) + h y'(x_i) + \frac{h^2}{2} y''(x_i) + O(h^3)$$

二式相减, 得局部截断误差

$$R_{i+1} = y(x_{i+1}) - y_{i+1}$$

$$= (1 - c_1 - c_2)hy'(x_i) + \left[\frac{1}{2} - a_2c_2 \right] h^2 y''(x_i) + O(h^3)$$

$$\text{令} \quad \begin{cases} 1 - c_1 - c_2 = 0 \\ \frac{1}{2} - a_2c_2 = 0 \end{cases}$$

$$\text{可得} \quad \begin{cases} c_1 + c_2 = 1 \\ a_2c_2 = \frac{1}{2} \end{cases}$$

只要 c_1, c_2, a_2 满足以上方程, 就得到一个二阶的 R-K 法.

这是一个不定方程, 有无穷多解. 如

(1) 取 $c_1 = c_2 = \frac{1}{2}, a_2 = 1$, 得

$$\begin{cases} y_{i+1} = y_i + \frac{h}{2}(K_1 + K_2) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + h, y_i + hK_1) \end{cases} \quad (9.13)$$

这实际上是公式 (9.10), 即梯形公式的预报校正公式.

(2) 取 $c_1 = 0, c_2 = 1, a_2 = \frac{1}{2}$, 得

$$\begin{cases} y_{i+1} = y_i + hK_2 \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}K_1) \end{cases} \quad (9.14)$$

该公式又称为中点公式.

9.2.2 四阶 R-K 法

用类似的方法可以确定三级和四级 R-K 法的参数, 构造出三阶和四阶的 R-K 法. 常用的是四阶 R-K 法. 四阶 R-K 法也不只一个, 下面给出的是最常用的四阶经典的 R-K 公式:

$$\begin{cases} y_{i+1} = y_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}K_1) \\ K_3 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}K_2) \\ K_4 = f(x_i + h, y_i + hK_3) \end{cases} \quad (i=0, 1, 2, \dots, n-1) \quad (9.15)$$

例 9-3 用经典的四阶 R-K 法计算

$$\begin{cases} y' = y - \frac{2x}{y} & (0 \leq x \leq 1) \\ y_0 = 1 \end{cases}$$

取步长为 0.2, 且与准确值比较.

解 由 (9.15) 式得

$$\begin{cases} y_{i+1} = y_i + \frac{0.2}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = y_i - \frac{2x_i}{y_i} \\ K_2 = y_i + 0.1K_1 - 2\frac{x_i + 0.1}{y_i + 0.1K_1} \\ K_3 = y_i + 0.1K_2 - 2\frac{x_i + 0.1}{y_i + 0.1K_2} \\ K_4 = y_i + 0.2K_3 - 2\frac{x_i + 0.2}{y_i + 0.2K_3} \end{cases} \quad (i = 0, 1, 2, \dots, n-1)$$

计算结果如表 9-3 所示.

表 9-3 计算结果

| x_i | y_i | $y(x_i)$ |
|-------|-----------|-----------|
| 0 | 1 | 1 |
| 0.2 | 1.183 229 | 1.183 216 |
| 0.4 | 1.341 667 | 1.341 641 |
| 0.6 | 1.483 281 | 1.483 240 |
| 0.8 | 1.612 514 | 1.612 452 |
| 1.0 | 1.732 142 | 1.732 051 |

可见, 使用 $h = 0.2$ 计算, 也比一阶和二阶方法精度高得多.

9.3 线性多步法

单步法只利用前一步的结果, 只要给出初值, 就能开始计算. 但也因为它只利用前一步的值, 为了提高精度就要计算一些非节点处的函数值, 增加了计算量. R-K 法就是通过这一途径提高精度的. 下面介绍的线性多步法, 在求 y_{i+1} 时, 不仅用到 y_i 的值, 还用道前若干步的 y_{i-1}, \dots, y_{i-k} 的值. 这些值都是已知的, 因此可在计算量增加不多的情况下提高精度.

9.3.1 用待定系数法构造线性多步法

线性多步法的一般形式是:

$$\alpha_0 y_i + \alpha_1 y_{i+1} + \cdots + \alpha_k y_{i+k} = h(\beta_0 f_i + \beta_1 f_{i+1} + \cdots + \beta_k f_{i+k})$$

或写为

$$\sum_{j=0}^k \alpha_j y_{i+j} = h \sum_{j=0}^k \beta_j f_{i+j} \quad (i = 0, 1, 2, \cdots, n-k) \quad (9.16)$$

其中 $\alpha_j, \beta_j (j=0, 1, \cdots, k)$ 都是实常数, 且 $\alpha_k \neq 0, |\alpha_0| + |\beta_0| \neq 0, f_{i+j} = f(x_{i+j}, y_{i+j}), j = 0, 1, \cdots, k$. 由 (9.16) 式可看出, 要计算 y_{i+k} , 就要利用它前面的 k 个值 $y_i, y_{i+1}, \cdots, y_{i+k-1}$. 又因为 (9.16) 式关于 y_{i+j} 和 f_{i+j} 都是线性组合, 所以这一类方法都称为线性 k 步法. 欧拉法、隐式欧拉法和梯形法都是线性一步法, 欧拉中点公式是线性二步法.

$$\text{令} \quad L(y(x), h) = \sum_{j=0}^k [\alpha_j y(x+jh) - h\beta_j y'(x+jh)]$$

把 $y(x+jh)$ 和 $y'(x+jh)$ 作 Taylor 展开:

$$\begin{aligned} y(x+jh) &= y(x) + jhy'(x) + \frac{(jh)^2}{2!} y''(x) + \cdots + \frac{(jh)^p}{p!} y^{(p)}(x) + \cdots \\ y'(x+jh) &= y'(x) + jhy''(x) + \cdots + \frac{(jh)^{p-1}}{(p-1)!} y^{(p)}(x) + \cdots \end{aligned}$$

于是,

$$\begin{aligned} L(y(x), h) &= \sum_{j=0}^k \left[\alpha_j y(x) + (j\alpha_1 - \beta_j) hy'(x) + \cdots + \left[\frac{j^p \alpha_j}{p!} - \frac{j^{p-1} \beta_j}{(p-1)!} \right] h^p y^{(p)}(x) + \cdots \right] \\ &= C_0 y(x) + C_1 hy'(x) + \cdots + C_p h^p y^{(p)}(x) + \cdots \end{aligned}$$

其中

$$\begin{aligned} C_0 &= \alpha_0 + \alpha_1 + \cdots + \alpha_k \\ C_1 &= (\alpha_1 + 2\alpha_2 + \cdots + k\alpha_k) - (\beta_0 + \beta_1 + \cdots + \beta_k) \\ C_2 &= \left[\frac{1}{2}\alpha_1 + 2\alpha_2 + \cdots + \frac{k^2}{2}\alpha_k \right] - (\beta_1 + 2\beta_2 + \cdots + k\beta_k) \\ C_3 &= \left[\frac{1}{6}\alpha_1 + \frac{4}{3}\alpha_2 + \cdots + \frac{k^3}{6}\alpha_k \right] - \left[\frac{1}{2}\beta_1 + 2\beta_2 + \cdots + \frac{k^2}{2}\beta_k \right] \\ &\vdots \\ C_p &= \frac{\alpha_1 + 2^p \alpha_2 + \cdots + k^p \alpha_k}{p!} - \frac{\beta_1 + 2^{p-1} \beta_2 + \cdots + k^{p-1} \beta_k}{(p-1)!} \end{aligned} \quad (9.17)$$

若选择 α_j, β_j , 使 $C_0 = C_1 = \cdots = C_p = 0, C_{p+1} \neq 0$, 则

$$L(y(x), h) = \sum_{j=0}^k [\alpha_j y(x+jh) - h\beta_j y'(x+jh)] \\ = C_{p+1} h^{p+1} y^{(p+1)}(x) + O(h^{p+2})$$

将 $x = x_i$ 代入上式, 设 $\alpha_k = 1$, 并注意到 $y'(x_i + jh) = f(x_i + jh, y(x_i + jh))$, 可推出

$$y(x_{i+k}) = - \sum_{j=0}^{k-1} \alpha_j y(x_i + jh) + h \sum_{j=0}^k \beta_j f(x_i + jh, y(x_i + jh)) + \\ C_{p+1} h^{p+1} y^{(p+1)}(x_i) + O(h^{p+2})$$

即

$$y(x_{i+k}) = - \sum_{j=0}^{k-1} \alpha_j y(x_{i+j}) + h \sum_{j=0}^k \beta_j f(x_{i+j}, y(x_{i+j})) + \\ C_{p+1} h^{p+1} y^{(p+1)}(x_i) + O(h^{p+2}) \quad (9.18)$$

设 $y_i = y(x_i)$, $y_{i+1} = y(x_{i+1})$, \dots , $y_{i+k-1} = y(x_{i+k-1})$, 记左端为 y_{i+k} , 并舍去最后两项, (9.18) 式变为

$$y_{i+k} = - \sum_{j=0}^{k-1} \alpha_j y_{i+j} + h \sum_{j=0}^k \beta_j f(x_{i+j}, y_{i+j}) \quad (9.19)$$

这就是一种 p 阶的线性 k 步方法, $C_{p+1} h^{p+1} y^{(p+1)}(x_i)$ 称为局部截断误差的主项. 当 $\beta_k = 0$ 时, 是显式方法; 当 $\beta_k \neq 0$ 时, 是隐式方法.

下面通过例子构造几个实用的线性多步法公式.

例 9-4 形如

$$y_{i+4} = -\alpha_0 y_i + h(\beta_1 f_{i+1} + \beta_2 f_{i+2} + \beta_3 f_{i+3})$$

的线性 4 步法公式, 试确定 $\alpha_0, \beta_1, \beta_2, \beta_3$, 并求其局部截断误差主项.

解 由 (9.16) 式知, $\alpha_1 = \alpha_2 = \alpha_3 = 0, \alpha_4 = 1, \beta_0 = \beta_4 = 0$, 因为有 4 个待定系数, 由 (9.17) 式写出前 4 个方程:

$$\begin{cases} C_0 = \alpha_0 + 1 = 0 \\ C_1 = 4 - (\beta_1 + \beta_2 + \beta_3) = 0 \\ C_2 = \frac{4^2}{2!} - (\beta_1 + 2\beta_2 + 3\beta_3) = 0 \\ C_3 = \frac{4^3}{3!} - \frac{\beta_1 + 2^2\beta_2 + 3^2\beta_3}{2!} = 0 \end{cases}$$

解之, 得 $\alpha_0 = -1, \beta_1 = \beta_3 = \frac{8}{3}, \beta_2 = -\frac{4}{3}$. 故所求的公式为

$$y_{i+4} = y_i + \frac{4h}{3}(2f_{i+1} - f_{i+2} + f_{i+3}) \quad (9.20)$$

将 $\alpha_0, \beta_1, \beta_2, \beta_3$ 代入 C_4 :

$$C_4 = \frac{4^4}{4!} - \frac{\beta_1 + 2^3\beta_2 + 3^3\beta_3}{3!} = 0$$

再求 C_5 :

$$C_5 = \frac{4^5}{5!} - \frac{\beta_1 + 2^4\beta_2 + 3^4\beta_3}{4!} = \frac{14}{45}$$

公式(9.20)称为米尔尼(Milne)公式,它的局部截断误差为:

$$R = \frac{14}{45}h^5 y^{(5)}(x_i) + O(h^6)$$

它也可以写成

$$y_{i+1} = y_{i-3} + \frac{4h}{3}(2f_{i-2} - f_{i-1} + 2f_i) \quad (i=3, 4, \dots, n-1) \quad (9.21)$$

局部截断误差可写为

$$R = \frac{14}{45}h^5 y^{(5)}(\xi) \quad \xi \in (x_{i-3}, x_{i+1}) \quad (9.22)$$

例 9-5 试确定下列公式的系数和局部截断误差.

$$y_{i+1} = ay_i + by_{i-2} + h(cf_{i+1} + df_i + ef_{i-1})$$

解 按(9.16)式和(9.17)式可知, $k=3, \alpha_0 = -b, \alpha_1 = 0, \alpha_2 = -a, \alpha_3 = 1, \beta_0 = 0, \beta_1 = e, \beta_2 = d, \beta_3 = c$, 有 5 个未知参数, 写出前 5 个方程:

$$\begin{cases} C_0 = 1 - a - b = 0 \\ C_1 = -2a + 3 - (e + d + c) = 0 \\ C_2 = \frac{3^2 - 2^2 a}{2} - (e + 2d + 3c) = 0 \\ C_3 = \frac{3^3 - 2^3 a}{3!} - \frac{e + 2^2 d + 3^2 c}{2!} = 0 \\ C_4 = \frac{3^4 - 2^4 a}{4!} - \frac{e + 2^3 d + 3^3 c}{3!} = 0 \end{cases}$$

解之得 $a = \frac{9}{8}, b = -\frac{1}{8}, c = \frac{3}{8}, d = \frac{6}{8}, e = -\frac{3}{8}$, 代入 C_5 , 得

$$C_5 = \frac{3^5 - 2^5 a}{5!} - \frac{e + 2^4 d + 3^4 c}{4!} = -\frac{1}{40}$$

此公式称为哈明(Hamming)公式, 写为

$$y_{i+1} = \frac{1}{8}(9y_i - y_{i-2}) + \frac{3}{8}h(f_{i+1} + 2f_i - f_{i-1}), i=2, 3, \dots, n-1 \quad (9.23)$$

局部截断误差记为

$$R = -\frac{1}{40}h^5 y^{(5)}(\xi) \quad \xi \in (x_{i-2}, x_{i+1}) \quad (9.24)$$

它是一个四阶隐式三步法.

在(9.16)式中,若 $\alpha_0 = \alpha_1 = \cdots = \alpha_{k-2} = 0, \alpha_k = 1$, 则称此类方法为阿达姆斯(Adams)类方法. 当 $\beta_k = 0$ 时, 称显式, 否则为隐式. 这类方法也可以用以上待定系数法求出. 这类方法的通式为

$$y_{i+1} = y_i + h \sum_{j=0}^k \beta_j f_{i+1-k+j} \quad (9.25)$$

例 9-6 求(9.25)式中的隐式三步法.

解 $k=3, \alpha_0 = \alpha_1 = 0, \alpha_2 = -1, \alpha_3 = 1,$

$$\begin{cases} C_0 = \alpha_2 + \alpha_3 = 0 \\ C_1 = 2\alpha_2 + 3\alpha_3 - (\beta_0 + \beta_1 + \beta_2 + \beta_3) = 0 \\ C_2 = \frac{2^2\alpha_2 + 3^2\alpha_3}{2} - (\beta_1 + 2\beta_2 + 3\beta_3) = 0 \\ C_3 = \frac{2^3\alpha_2 + 3^3\alpha_3}{6} - \frac{\beta_1 + 2^2\beta_2 + 3^2\beta_3}{2} = 0 \\ C_4 = \frac{2^4\alpha_2 + 3^4\alpha_3}{24} - \frac{\beta_1 + 2^3\beta_2 + 3^3\beta_3}{6} = 0 \end{cases}$$

$\alpha_2 = -1, \alpha_3 = 1.$

$$\begin{cases} \beta_0 + \beta_1 + \beta_2 + \beta_3 = 1 \\ \beta_1 + 2\beta_2 + 3\beta_3 = \frac{5}{2} \\ \beta_1 + 4\beta_2 + 9\beta_3 = \frac{19}{3} \\ \beta_1 + 8\beta_2 + 27\beta_3 = \frac{65}{4} \end{cases}$$

解之, 得 $\beta_0 = \frac{1}{24}, \beta_1 = \frac{-5}{24}, \beta_2 = \frac{19}{24}, \beta_3 = \frac{9}{24}.$

公式为:

$$y_{i+1} = y_1 + \frac{h}{24}(f_{i-2} - 5f_{i-1} + 19f_i + 9f_{i+1}) \quad (9.26)$$

$$C_5 = \frac{1}{120}(2^5\alpha_2 + 3^5\alpha_3) - \frac{1}{24}(\beta_1 + 2^4\beta_2 + 3^4\beta_3) = -\frac{19}{720}$$

局部截断误差为

$$R = -\frac{19}{720}h^5 y^{(5)}(\xi) \quad \xi \in (x_{i-2}, x_{i+1}) \quad (9.27)$$

它是一个线性三步四阶隐式公式, 称为四阶阿达姆斯内插公式, 应用十分普遍.

此外, 常用的还有线性四步四阶显式阿达姆斯公式, 常称为阿达姆斯外推公式:

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}) \quad (i=3, 4, \dots, n-1) \quad (9.28)$$

$$R = \frac{251}{720}h^5 y^{(5)}(\xi) \quad \xi \in (x_{i-3}, x_{i+1}) \quad (9.29)$$

9.3.2 用数值积分法构造线性多步法公式

在(9.16)式中,若令 $\alpha_k = 1, \alpha_0, \alpha_1, \dots, \alpha_{k-1}$ 中只有一个不为零,这样的线性多步法公式也可用数值积分方法构造,如阿达姆斯外推公式.对方程

$$\begin{cases} y' = f(x, y(x)) \\ y(x_0) = y_0 \end{cases} \quad (a \leq x \leq b)$$

第一式两端在 $[x_i, x_{i+1}]$ 上积分,得

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} f(x, y(x)) dx \quad (9.30)$$

选择节点 $x_{i-3}, x_{i-2}, x_{i-1}, x_i$ 作插值节点,作函数 $f(x, y(x))$ 的三次插值多项式:

$$P_3(x) = \frac{(x-x_{i-1})(x-x_{i-2})(x-x_{i-3})}{h(2h)(3h)}f_i + \frac{(x-x_i)(x-x_{i-2})(x-x_{i-3})}{(-h)(h)(2h)}f_{i-1} + \\ \frac{(x-x_i)(x-x_{i-1})(x-x_{i-3})}{(-2h)(-h)(h)}f_{i-2} + \frac{(x-x_i)(x-x_{i-1})(x-x_{i-2})}{(-3h)(-2h)(-h)}f_{i-3}$$

且插值余项

$$R_3(x) = \frac{f^{(4)}(\xi)}{4!}(x-x_i)(x-x_{i-1})(x-x_{i-2})(x-x_{i-3})$$

把 $f(x, y(x)) = P_3(x) + R_3(x)$ 代入(9.30)式得

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} P_3(x) dx + \int_{x_i}^{x_{i+1}} R_3(x) dx$$

略去右边第二项:

$$y(x_{i+1}) \approx y(x_i) + \int_{x_i}^{x_{i+1}} P_3(x) dx \quad (i=3, 4, \dots, n-1) \quad (9.31)$$

计算积分得

$$\int_{x_i}^{x_{i+1}} P_3(x) dx = \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$$

把(9.31)式左边记为 y_{i+1} ,并把约等于号改为等号,得

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}) \quad (i=3, 4, \dots, n-1) \quad (9.32)$$

这就是阿达姆斯外推公式.

其局部截断误差为

$$R = \int_{x_i}^{x_{i+1}} R_3(x) dx = \int_{x_i}^{x_{i+1}} \frac{f^{(4)}(\xi)}{4!} (x - x_i)(x - x_{i-1})(x - x_{i-2})(x - x_{i-3}) dx$$

因为 $(x - x_i)(x - x_{i-1})(x - x_{i-2})(x - x_{i-3})$ 在 $[x_i, x_{i+1}]$ 上不变号, 设 $f^{(4)}(\xi)$ 在 $[x_i, x_{i+1}]$ 上连续, 则有

$$\begin{aligned} R &= \frac{f^{(4)}(\eta)}{4!} \int_{x_i}^{x_{i+1}} (x - x_i)(x - x_{i-1})(x - x_{i-2})(x - x_{i-3}) dx \\ &= \frac{251}{720} h^5 f^{(4)}(\eta) = \frac{251}{720} h^5 y^{(5)}(\eta) \quad (x_{i-3} < \eta < x_{i+1}) \end{aligned}$$

这与用待定系数法推导的结果完全一致.

用类似的方法还可以构造二步四阶的辛普生 (Simpson) 公式:

$$y_{i+1} = y_{i-1} + \frac{h}{3} (f_{i+1} + 4f_i + f_{i-1}) \quad (i = 2, 3, \dots, n-1) \quad (9.33)$$

它的局部截断误差为

$$R = -\frac{1}{90} h^5 y^{(5)}(\xi) \quad (x_{i-1} < \xi < x_{i+1}) \quad (9.34)$$

线性多步法要用到二个以上的初始值才能开始计算, 不能“自启动”. 一般应选择同阶或高阶的单步法算出前几个值, 再使用线性多步法. 由于线性多步法每一步实际上只计算一次函数值, 又能获得较高的精度, 所以应用比较广泛.

练习与思考

1. 对初值问题

$$\begin{cases} y' = \frac{1}{1+x^2} - 2y^2 & (0 \leq x \leq 1) \\ y(0) = 0 \end{cases}$$

试用欧拉法, 取步长为 0.1 和 0.2 分别计算 $y(1)$, 并与准确解 $y(x) = \frac{x}{1+x^2}$ 相比较. 近似值和准确解都取四位有效数字.

2. 用预报校正法 (改进的欧拉法) 和四阶经典 R-K 法, 取步长 $h = 0.1$, 计算下列初值问题的近似解, 并与精确值比较. 均保留四位小数:

$$\begin{cases} y' = x + y & (0 \leq x \leq 1) \\ y(0) = 1 \end{cases}$$

此问题的精确解是 $y(x) = -x - 1 + 2e^x$.

3. 证明二阶 R-K 法能精确地解微分方程 $y' = ax + b$.

4. 用待定系数法推导二步法公式

$$y_{n+1} = y_n + \frac{h}{12} (5f_{n+1} + 8f_n - f_{n-1})$$

并证明它是三阶公式, 求出它的局部截断误差.

参考文献

- 1 李岳生,黄友谦.数值逼近(第2版).北京:高等教育出版社,1987
- 2 冯康等编.数值计算方法.北京:国防工业出版社,1978
- 3 李庆杨,王能超,易大义.数值分析(第3版).武汉:华中理工大学出版社,1986
- 4 徐萃薇,孙绳武编著.计算方法引论(第2版).北京:高等教育出版社,2002
- 5 吴勃英等编.数值分析原理.北京:科学出版社,2003
- 6 孙志中等编.数值分析(第2版).南京:东南大学出版社,2002
- 7 [美]John H. Mathews, Kurtis D. Fink 著,陈愉等译.数值方法(第3版).北京:电子工业出版社,2002
- 8 Richard L. Burden, J. Douglas Faires 著. NUMERICAL ANALYSIS (Seventh Edition). 北京:高等教育出版社,2001