# 数值分析讲义*

潘建瑜

jypan@math.ecnu.edu.cn

2018 年 5 月 31 日

*本讲义仅供《数值分析》课堂教学使用

纸上得来终觉浅，绝知此事要躬行。

# 目 录

# 第一讲 数值分析引论

## 1.1 数值分析介绍

### 1.1.1 科学计算

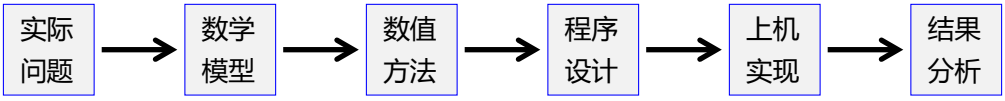
计算机是二十世纪最伟大的科学技术发明之一. 计算机对人类的生产活动和社会活动产生了极其重要的影响. 特别是随着网络的出现, 计算机已经彻底改变了人们的生活, 学习和工作. 它是人类进入信息时代的重要标志之一.

随着计算机技术的飞速发展, 数学方法及计算已成为当今科学研究中不可缺少的手段, 从宇宙飞船到家用电器, 从质量控制到市场营销, 通过建立数学模型, 应用数学理论和方法, 并结合计算机解决实际问题已成为十分普遍的研究模式.

> 一门科学, 只有当它成功地运用数学时, 才能达到真正完善的地步.
>
> — 马克思 (1818.05.05 – 1883.03.14)

> 一个国家只有数学蓬勃发展, 才能表现她的国力强大.
>
> — 陈省身 (1911.10.26 – 2004.12.03)

科学计算就是指借助计算机高速计算的能力, 解决现代科学, 工程, 经济或人文中的复杂问题. 科学计算已经与实验, 理论并列成为当今科学研究的三大方法. 一般来说, 运用科学计算解决实际问题主要分以下几个步骤:




### 1.1.2 数值分析的研究内容

计算机由硬件和软件组成. 硬件包括 CPU, 内存, 主板, 硬盘等. 软件则是计算机的灵魂, 而软件的核心就是算法, 包括数值算法和非数值算法. 数值分析主要研究的就是数值算法的构造与分析.

狭义的科学计算是针对某些特定的数学问题, 设计有效的计算方法来求解, 即为数值计算, 数值分析, 计算方法. 目前, 科学计算的主要研究内容有矩阵计算, 函数逼近, 微分方程数值解, 最优化等.

**数值分析的主要任务**

- 设计求解各种实际问题的高效可靠的数值方法
  - 可用：可以并易于在计算机上实现
  - 可靠：收敛性稳定性等有数学理论保证
  - 高效：尽可能地节省计算时间和存储空间
  - 数值实验：要通过数值试验来证明是行之有效的
- 对求得的数值解进行评估: 误差估计, 稳定性分析
- 研究数值算法在计算机上的实现

**数值计算的主要特点**

- 方法是近似的, 所以求出的解是有误差的;
- 与计算机紧密结合: 方法必须能够在计算机上实现.

♣ 对于同一问题，不同的算法在计算性能上可能相差百万倍或者更多!

**例 1.1** 线性方程组求解: 克莱姆法则与高斯消去法.

- 克莱姆法则需要计算 $n+1$ 个 $n$ 阶行列式, 在不计加减运算情况下, 至少需要 $n!(n^2-1)$ 次乘除运算;
- 高斯消去法只需约 $2n^3/3$ 次乘除运算.

以 $n=20$ 为例, 克莱姆法则大概需要 $20!(20^2-1)\approx 9.7\times 10^{20}$ 次运算, 如果用每秒运算 30 亿次 (主频 3.0G) 的计算机求解时, 大约需要 10000 年的时间! 如果使用高斯消去法, 则 1 秒钟都不要.

### 1.1.3 学习建议

在学习数值分析时, 大家要注意以下几点:

- 注意掌握数值方法的基本思想和原理
- 注意数值方法的常用技巧
- 要重视误差分析、收敛性和稳定性的基本理论
- 适量的数值训练 (笔算, 上机)

### 1.1.4 推荐参考资料

[1] Richard L. Burden and J. Douglas Faires, Numerical Analysis, 9th Edition, Brooks/Cole, Cengage Learning, 2011. (数值分析, 第七版有中文翻译)

[2] K. Artkinson and Weimin Han, Elementary Numerical Analysis, 3rd Edition, John Wiley & Sons, 2003. (数值分析导论, 有中文翻译)

[3] M.T. Heath, Scientific Computing: An Introductory Survey, 2nd Edition, 2001 (科学计算导论, 有中文翻译)

[4] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, 3rd Edition, Springer, 2002. (数值分析导论, 第二版有中文翻译)

其中 [1-3] 比较基础, 并介绍了相关软件, 文献 [4] 比较偏理论.

**本讲义中常用的数学记号**

| | |
|---|---|
| $\mathbb{R}$ | 实数域 |
| $\mathbb{C}$ | 复数域 |
| $\mathbb{R}^n$ | $n$ 维实向量空间 |
| $\mathbb{C}^n$ | $n$ 维复向量空间 |
| $C[a,b]$ | $[a,b]$ 上的连续函数空间 |
| $C^n[a,b]$ | $[a,b]$ 上的 $n$ 次连续可导函数空间 |
| $H_n$ | 所有次数不超过 $n$ 的实系数多项式组成的集合 |
| $\tilde{H}_n$ | 所有首项系数为 1 的 $n$ 次实系数多项式组成的集合 |

## 1.2 线性代数基础

### 1.2.1 线性空间基本概念

线性空间是线性代数最基本的概念之一, 它是定义在某个数域上并满足一定条件的一个集合. 我们首先给出数域的概念.

**定义 1.1 (数域)** 设 $\mathbb{F}$ 是包含 0 和 1 的一个数集, 如果 $\mathbb{F}$ 中的任意两个数的和, 差, 积, 商 (除数不为 0) 仍然在 $\mathbb{F}$ 中, 则称 $\mathbb{F}$ 为一个数域.

**例 1.2** 常见的数域有: 有理数域 $\mathbb{Q}$, 实数域 $\mathbb{R}$ 和复数域 $\mathbb{C}$.

**定义 1.2 (线性空间)** 设 $\mathcal{S}$ 是一个非空集合, $\mathbb{F}$ 是一个数域. 在 $\mathcal{S}$ 上定义一种代数运算, 称为加法, 记为 "$+$" (即对任意 $x, y \in \mathcal{S}$, 都存在唯一的 $z \in \mathcal{S}$, 使得 $z = x + y$), 并定义一个从 $\mathbb{F} \times \mathcal{S}$ 到 $\mathcal{S}$ 的代数运算, 称为数乘, 记为 "$\cdot$" (即对任意 $\alpha \in \mathbb{F}$ 和任意 $x \in \mathcal{S}$, 都存在唯一的 $y \in \mathcal{S}$, 使得 $y = \alpha \cdot x$). 如果这两个运算满足下面的规则, 则称 $(\mathcal{S}, +, \cdot)$ 是数域 $\mathbb{F}$ 上的一个线性空间 (通常简称 $\mathcal{S}$ 是数域 $\mathbb{F}$ 上的一个线性空间):

- 加法四条规则
  - (1) 交换律: $x + y = y + x, \quad \forall\, x, y \in \mathcal{S}$;
  - (2) 结合律: $(x + y) + z = x + (y + z), \quad \forall\, x, y, z \in \mathcal{S}$;
  - (3) 零元素: 存在一个元素 0, 使得 $x + 0 = x, \quad \forall\, x \in \mathcal{S}$;
  - (4) 逆运算: 对任意 $x \in \mathcal{S}$, 都存在负元素 $y \in \mathcal{S}$, 使得 $x + y = 0$, 记 $y = -x$;
- 数乘四条规则
  - (1) 单位元: $1 \cdot x = x, \quad 1 \in \mathbb{F}, \forall\, x \in \mathcal{S}$;
  - (2) 结合律: $\alpha \cdot (\beta \cdot x) = (\alpha\beta) \cdot x, \quad \forall\, \alpha, \beta \in \mathbb{F},\ x \in \mathcal{S}$;
  - (3) 分配律: $(\alpha + \beta) \cdot x = \alpha \cdot x + \beta \cdot x, \quad \forall\, \alpha, \beta \in \mathbb{F},\ x \in \mathcal{S}$;
  - (4) 分配律: $\alpha \cdot (x + \beta) = \alpha \cdot x + \alpha \cdot y, \quad \forall\, \alpha \in \mathbb{F},\ x, y \in \mathcal{S}$.

为了表示方便, 通常省略数乘符号, 即将 $\alpha \cdot x$ 写成 $\alpha x$.

**例 1.3** 常见的线性空间有:

- $\mathbb{R}^n \to$ 所有 $n$ 维实向量组成的集合, 是 $\mathbb{R}$ 上的线性空间.
- $\mathbb{C}^n \to$ 所有 $n$ 维复向量组成的集合, 是 $\mathbb{C}$ 上的线性空间.
- $\mathbb{R}^{m\times n} \to$ 所有 $m \times n$ 阶实矩阵组成的集合, 是 $\mathbb{R}$ 上的线性空间.
- $\mathbb{C}^{m\times n} \to$ 所有 $m \times n$ 阶复矩阵组成的集合, 是 $\mathbb{C}$ 上的线性空间.
- $\mathbb{H}_n \to$ 所有次数不超过 $n$ 的多项式组成的集合.
- $C[a,b] \to$ 区间 $[a,b]$ 上所有连续函数组成的集合.
- $C^p[a,b] \to$ 区间 $[a,b]$ 上所有 $p$ 次连续可微函数组成的集合.

### 线性相关性和维数

设 $\mathcal{S}$ 是数域 $\mathbb{F}$ 上的一个线性空间, $x_1, x_2, \dots, x_k$ 是 $\mathcal{S}$ 中的一组向量. 如果存在 $k$ 个不全为零的数 $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{F}$, 使得

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k = 0,$$

则称 $x_1, x_2, \dots, x_k$ 线性相关, 否则就是线性无关.

设 $x_1, x_2, \dots, x_k$ 是 $\mathcal{S}$ 中的一组向量. 如果 $x \in \mathcal{S}$ 可以表示为

$$x = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k,$$

其中 $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{F}$, 则称 $x$ 可以由 $x_1, x_2, \dots, x_k$ 线性表示, 或者称 $x$ 是 $x_1, x_2, \dots, x_k$ 的线性组合, $\alpha_1, \alpha_2, \dots, \alpha_k$ 称为 线性表出系数

设向量组 $\{x_1, x_2, \dots, x_m\}$, 如果存在其中的 $r$ ($r \le m$) 个线性无关向量 $x_{i_1}, x_{i_2}, \dots, x_{i_r}$, 使得所有向量都可以由它们线性表示, 则称 $x_{i_1}, x_{i_2}, \dots, x_{i_r}$ 为向量组 $\{x_1, x_2, \dots, x_m\}$ 的一个极大线性无关组, 并称这组向量的秩为 $r$, 记为 $\operatorname{rank}(\{x_1, x_2, \dots, x_m\}) = r$.

设 $x_1, x_2, \dots, x_n$ 是 $\mathcal{S}$ 中的一组线性无关向量. 如果 $\mathcal{S}$ 中的任意一个向量都可以由 $x_1, x_2, \dots, x_n$ 线性表示, 则称 $x_1, x_2, \dots, x_n$ 是 $\mathcal{S}$ 的一组基, 并称 $\mathcal{S}$ 是 $n$ 维的, 即 $\mathcal{S}$ 的维数为 $n$, 记为 $\dim(\mathcal{S}) = n$. 如果 $\mathcal{S}$ 中可以找到任意多个线性无关向量, 则称 $\mathcal{S}$ 是无限维的.

### 子空间

设 $\mathcal{S}$ 是一个线性空间, $\mathcal{W}$ 是 $\mathcal{S}$ 的一个非空子集合. 如果 $\mathcal{W}$ 关于 $\mathcal{S}$ 上的加法和数乘也构成一个线性空间, 则称 $\mathcal{W}$ 为 $\mathcal{S}$ 的一个线性子空间, 有时简称子空间.

**例 1.4** 设 $\mathcal{S}$ 是一个线性空间, 则由零向量组成的子集 $\{0\}$ 是 $\mathcal{S}$ 的一个子空间, 称为零子空间. 另外, $\mathcal{S}$ 本身也是 $\mathcal{S}$ 的子空间. 这两个特殊的子空间称为 $\mathcal{S}$ 的平凡子空间, 其他子空间都是非平凡子空间.

下面给出子空间的判别定理.

**定理 1.1** 设 $\mathcal{S}$ 是数域 $\mathbb{F}$ 上的一个线性空间, $\mathcal{W}$ 是 $\mathcal{S}$ 的一个非空子集合. 则 $\mathcal{W}$ 是 $\mathcal{S}$ 的一个子空间的充要条件是 $\mathcal{W}$ 关于加法和数乘封闭, 即

(1) 对任意 $x, y \in \mathcal{W}$, 有 $x + y \in \mathcal{W}$;

(2) 对任意 $\alpha \in \mathbb{F}$ 和任意 $x \in \mathcal{W}$, 有 $\alpha x \in \mathcal{W}$.

### 1.2.2 范数与赋范线性空间

**定义 1.3 (范数与赋范线性空间)** 设 $\mathcal{S}$ 为数域 $\mathbb{F}$ ($\mathbb{F}$ 可以是 $\mathbb{R}$ 或 $\mathbb{C}$) 上的线性空间, 若对任意 $x \in \mathcal{S}$, 存在唯一实数与之对应, 记为 $\|x\|$, 它满足条件:

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

(2) $\|\alpha x\| = |\alpha| \cdot \|x\|, \forall\, \alpha \in \mathbb{F}$; (正齐次性)

(3) $\|x + y\| \leq \|x\| + \|y\|, \forall\, x, y \in S$; (三角不等式)

则称 $\|\cdot\|$ 为线性空间 $\mathcal{S}$ 上的范数, 定义了范数的线性空间称为赋范线性空间.

♣ 范数是从 $\mathcal{S}$ 到 $\mathbb{R}_+ \cup \{0\}$ 的一元函数, 其中 $\mathbb{R}_+$ 表示所有正实数组成的集合.

**例 1.5** $\mathbb{R}^n$ 上的常用范数: 设 $x = [x_1, x_2, \ldots, x_n]^\intercal \in \mathbb{R}^n$

- 1-范数: $\|x\|_1 = \sum\limits_{i=1}^{n} |x_i|$;
- 2-范数: $\|x\|_2 = \left( \sum\limits_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}}$;
- $p$-范数: $\|x\|_p = \left( \sum\limits_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$;
- $\infty$-范数: $\|x\|_\infty = \max\limits_{1 \leq i \leq n} |x_i|$.

♣ 1-范数, 2-范数, $\infty$-范数可以看作是 $p$-范数在 $p = 1, 2, \infty$ 时的特殊情形.

**例 1.6** $C[a, b]$ 上的常用范数: 设 $f(x) \in C[a, b]$

- 1-范数: $\|f\|_1 = \int_a^b |f(x)|\, \mathrm{d}x$;
- 2-范数: $\|f\|_2 = \left( \int_a^b |f(x)|^2\, \mathrm{d}x \right)^{\frac{1}{2}}$;
- $p$-范数: $\|f\|_p = \left( \int_a^b |f(x)|^p\, \mathrm{d}x \right)^{\frac{1}{p}}$.
- $\infty$-范数: $\|f\|_\infty = \max\limits_{a \leq x \leq b} |f(x)|$;

### 1.2.3 内积与内积空间

**定义 1.4 (内积与内积空间)** 设 $\mathcal{S}$ 是数域 $\mathbb{F}$ ($\mathbb{R}$ 或 $\mathbb{C}$) 上的一个线性空间, 定义一个从 $\mathcal{S} \times \mathcal{S}$ 到 $\mathbb{F}$ 的代数运算, 记为 "$(\cdot, \cdot)$", 即对任意 $x, y \in \mathcal{S}$, 都存在唯一的 $f \in \mathbb{F}$, 使得 $f = (x, y)$. 如果该运算满足

(1) $(x, y) = \overline{(y, x)}, \quad \forall\, x, y \in \mathcal{S}$;

(2) $(\alpha x, y) = k(x, y), \quad \forall\, \alpha \in \mathbb{F},\ x, y \in \mathcal{S}$;

(3) $(x+y,z)=(x,z)+(y,z),\quad \forall\, x,y,z\in\mathcal{S}$;

(4) $(x,x)\geq 0$, 等号当且仅当 $x=0$ 时成立;

则称 $(\cdot,\cdot)$ 为 $\mathcal{S}$ 上的一个 内积, 有时也称为 标量积. 定义了内积的线性空间称为 内积空间.

♣ 定义在实数域 $\mathbb{R}$ 上的内积空间称为欧氏空间, 定义在复数域 $\mathbb{C}$ 上的内积空间称为酉空间.

♣ $\overline{(u,v)}$ 表示 $(u,v)$ 的共轭. 当 $\mathbb{F}=\mathbb{R}$ 时, 该条件即为 $(v,u)=(u,v)$.

♣ 内积是从线性空间 $\mathcal{S}$ 到数域 $\mathbb{F}$ 的二元函数.

♣ 由内积定义可以立即得出:

- $(u,\beta v)=\bar{\beta}(u,v)$
- $(u,v+w)=(u,v)+(u,w)$
- $(u_1+u_2+\cdots+u_n,v)=(u_1,v)+(u_2,v)+\cdots+(u_n,v)$

**例 1.7** 考虑 $n$ 维线性空间 $\mathbb{R}^n$, 对任意 $x=[x_1,x_2,\ldots,x_n]^{\mathsf{T}}\in\mathbb{R}^n$, $y=[y_1,y_2,\ldots,y_n]^{\mathsf{T}}\in\mathbb{R}^n$, 定义

$$(x,y)\triangleq y^{\mathsf{T}}x=\sum_{i=1}^{n}x_iy_i,$$

则 $(x,y)$ 是 $\mathbb{R}^n$ 上的内积, 因此 $\mathbb{R}^n$ 是一个欧氏空间. 这种方式定义的内积就称为 欧拉内积.

**例 1.8** 考虑 $n$ 维线性空间 $\mathbb{C}^n$, 对任意 $x=[x_1,x_2,\ldots,x_n]^{\mathsf{T}}\in\mathbb{C}^n$, $y=[y_1,y_2,\ldots,y_n]^{\mathsf{T}}\in\mathbb{C}^n$, 定义

$$(x,y)\triangleq y^{*}x=\sum_{i=1}^{n}x_i\bar{y}_i,$$

这里 $y^*$ 表示 $y$ 的共轭转置. 则 $(x,y)$ 是 $\mathbb{C}^n$ 上的内积, 因此 $\mathbb{C}^n$ 是一个酉空间.

♣ 以上定义的内积是 $\mathbb{R}^n$ 和 $\mathbb{C}^n$ 上的标准内积, 若不特别声明, $\mathbb{R}^n$ 和 $\mathbb{C}^n$ 上的内积均是指标准内积.

**例 1.9** 在实数域 $\mathbb{R}$ 上线性空间 $C[a,b]$ 中, 对任意 $f(x),g(x)\in C[a,b]$, 定义

$$(f,g)=\int_a^b f(x)g(x)\,\mathrm{d}x,$$

则 $(f,g)$ 是 $C[a,b]$ 上的内积, 因此 $C[a,b]$ 是一个欧氏空间.

**定义 1.5** 若 $(u,v)=0$, 则称 $u$ 与 $v$ 是正交的.

**定理 1.2 (Cauchy-Schwarz 不等式)** 设 $\mathcal{S}$ 是内积空间, 则对任意 $u,v\in\mathcal{S}$, 有

$$|(u,v)|^2\leq(u,u)\cdot(v,v). \tag{1.1}$$

证明. 若 $v=0$, 则结论显然成立.

假定 $v \neq 0$, 则 $(v,v) > 0$. 对任意实数 $\lambda$ 有

$$0 \leq (u+\lambda v, u+\lambda v) = (u,u) + \lambda(v,u) + \bar{\lambda}(u,v) + |\lambda|^2(v,v).$$

取 $\lambda = -\dfrac{(u,v)}{(v,v)}$, 代入可得

$$0 \leq (u,u) - \frac{(u,v)(v,u)}{(v,v)} - \frac{\overline{(u,v)}(u,v)}{(v,v)} + \frac{|(u,v)|^2}{(v,v)} = (u,u) - \frac{|(u,v)|^2}{(v,v)}.$$

即

$$|(u,v)|^2 \leq (u,u)\cdot(v,v).$$

□

思考 Cauchy-Schwarz 不等式 (1.1) 中, 等号成立的充要条件是什么?

**定理 1.3** 设 $\mathcal{S}$ 是内积空间, $u_1, u_2, \ldots, u_n \in \mathcal{S}$, 则 $u_1, u_2, \ldots, u_n$ 线性无关的充要条件是矩阵 $G$ 非奇异, 其中

$$G = \begin{bmatrix} (u_1,u_1) & (u_2,u_1) & \cdots & (u_n,u_1) \\ (u_1,u_2) & (u_2,u_2) & \cdots & (u_n,u_2) \\ \vdots & \vdots & & \vdots \\ (u_1,u_n) & (u_2,u_n) & \cdots & (u_n,u_n) \end{bmatrix}$$

这个矩阵就称为 $u_1, u_2, \ldots, u_n$ 的 Gram 矩阵.

证明. 矩阵 $G$ 非奇异当且仅当 $Gx = 0$ 只有零解. 由 $G$ 的定义可知 $Gx = 0$ 即为

$$\sum_{i=1}^{n}(u_i,u_k)x_i = 0, \quad k = 1,2,\ldots,n,$$

即

$$0 = \sum_{i=1}^{n}(x_iu_i,u_k) = \left(\sum_{i=1}^{n}x_iu_i,u_k\right), \quad k = 1,2,\ldots,n.$$

所以

$$\left(\sum_{i=1}^{n}x_iu_i,\sum_{i=1}^{n}x_iu_i\right) = \sum_{k=1}^{n}\bar{x}_k\left(\sum_{i=1}^{n}x_iu_i,u_k\right) = 0.$$

因此 $\sum\limits_{i=1}^{n}x_iu_i = 0$. 若 $u_1, u_2, \ldots, u_n$ 线性无关, 则 $x_i = 0$, 即 $Gx = 0$ 只有零解.

反之, 设 $Gx = 0$ 只有零解. 若 $u_1, u_2, \ldots, u_n$ 线性相关, 则存在一组不全为零的数 $\alpha_i$, 使得 $\sum\limits_{i=1}^{n}\alpha_iu_i = 0$. 于是

$$0 = \left(\sum_{i=1}^{n}\alpha_iu_i,u_k\right) = \sum_{i=1}^{n}\alpha_i(u_i,u_k), \quad k = 1,2,\ldots,n.$$

因此 $x = [\alpha_1, \alpha_2, \ldots, \alpha_n]^{\mathsf{T}}$ 是 $Gx = 0$ 的非零解, 与条件矛盾. 故 $u_1, u_2, \ldots, u_n$ 线性无关. □

### 1.2.4 带权内积与范数

设 $\mathcal{S}$ 是内积空间, 对任意 $u \in \mathcal{S}$, 定义

$$\|u\| \triangleq (u,u)^{\frac{1}{2}},$$

则可以验证, $\|u\|$ 是 $\mathcal{S}$ 上的范数. 这就是由内积导出的范数.

♣ 任意一个内积都可以导出一个相应的范数.

**例 1.10** $\mathbb{R}^n$ 上由标准内积导出的范数为

$$\|x\| = (x,x)^{\frac{1}{2}} = \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}},$$

这就是 2-范数.

设 $\omega_1, \omega_2, \ldots, \omega_n$ 为给定的正实数, 对任意 $x, y \in \mathbb{R}^n$, 定义

$$(x,y)_\omega \triangleq \sum_{i=1}^{n} \omega_i x_i y_i.$$

可以验证, $(x,y)_\omega$ 是 $\mathbb{R}^n$ 上的内积, 我们称其为带权内积, 其中 $\omega_i$ 称为权系数.

同样, 在 $\mathbb{C}^n$ 中也可以定义带权内积

$$(x,y)_\omega \triangleq \sum_{i=1}^{n} \omega_i x_i \bar{y}_i, \quad \forall\, x, y \in \mathbb{C}^n,$$

其中 $\omega_i$ 都是正实数.

为了在 $C[a,b]$ 上定义带权内积, 我们首先定义权函数.

**定义 1.6** 设 $\rho(x)$ 是 $[a,b]$ 上的非负函数, 如果 $\rho(x)$ 满足

(1) 对任意非负整数 $k$, $\int_a^b x^k \rho(x)\,\mathrm{d}x$ 存在且为有限值;

(2) 对 $[a,b]$ 上的任意非负连续函数 $g(x)$, 如果 $\int_a^b g(x)\rho(x)\,\mathrm{d}x = 0$, 则 $g(x) \equiv 0$;

则称 $\rho(x)$ 是 $[a,b]$ 上的一个权函数.

♣ $[a,b]$ 是有限或无限空间, 即 $a$ 可以是 $-\infty$, $b$ 可以是 $\infty$; 权函数与定义区间相关.

**例 1.11** 常见的权函数有

- $\rho(x) = 1$ 是 $[-1,1]$ 上的权函数;
- $\rho(x) = \dfrac{1}{\sqrt{1-x^2}}$ 是 $[-1,1]$ 上的权函数;
- $\rho(x) = e^{-x}$ 是 $[0,\infty)$ 上的权函数;
- $\rho(x) = e^{-x^2}$ 是 $(-\infty,\infty)$ 上的权函数.

**例 1.12** 设 $\rho(x)$ 是 $[a,b]$ 上的权函数, 对任意 $f(x), g(x) \in C[a,b]$, 定义

$$(f,g)_\rho = \int_a^b \rho(x) f(x) g(x)\,\mathrm{d}x,$$

则 $(f,g)_\rho$ 是 $C[a,b]$ 上的带权内积. 对应的带权内积导出范数可定义为

$$\|f\|_\rho = (f,f)_\rho^{\frac{1}{2}} = \left(\int_a^b \rho(x) f^2(x)\,\mathrm{d}x\right)^{\frac{1}{2}}.$$

特别地, 当 $\rho(x) \equiv 1$ 时, 内积导出范数为

$$\|f\| = (f,f)^{\frac{1}{2}} = \left(\int_a^b f^2(x)\,\mathrm{d}x\right)^{\frac{1}{2}}.$$

这就是 $f(x)$ 的 2-范数.

根据定理 1.3, 我们可以得到下面的推论.

**推论 1.4** 设 $\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x) \in C[a,b]$, 则 $\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x)$ 线性无关的充要条件是矩阵 $G$ 非奇异, 其中 $G$ 是 Gram 矩阵:

$$G = \begin{bmatrix} (\varphi_1,\varphi_1) & (\varphi_2,\varphi_1) & \cdots & (\varphi_n,\varphi_1) \\ (\varphi_1,\varphi_2) & (\varphi_2,\varphi_2) & \cdots & (\varphi_n,\varphi_2) \\ \vdots & \vdots & & \vdots \\ (\varphi_1,\varphi_n) & (\varphi_2,\varphi_n) & \cdots & (\varphi_n,\varphi_n) \end{bmatrix}.$$

### 1.2.5 向量范数与矩阵范数

#### 向量范数

定义在 $\mathbb{R}^n$ (或 $\mathbb{C}^n$) 上的范数就是向量范数. $\mathbb{R}^n$ 上常见的范数有 1-范数, 2-范数, $p$-范数和 $\infty$-范数, 见例 1.5. 下面给出向量范数的一些重要性质.

**定理 1.5 (向量范数的连续性)** 设 $\|\cdot\|$ 是 $\mathbb{R}^n$ 上的一个向量范数, 则 $f(x) \triangleq \|x\|$ 关于 $x$ 的每个分量是连续的. (板书)

**定义 1.7 (范数的等价性)** 设 $\|\cdot\|_\alpha$ 与 $\|\cdot\|_\beta$ 是 $\mathbb{C}^n$ 空间上的两个向量范数, 若存在正常数 $c_1, c_2$, 使得

$$c_1\|x\|_\alpha \le \|x\|_\beta \le c_2\|x\|_\alpha$$

对任意 $x \in \mathbb{C}^n$ 都成立, 则称 $\|\cdot\|_\alpha$ 与 $\|\cdot\|_\beta$ 是等价的.

**定理 1.6** $\mathbb{R}^n$ 上的所有向量范数都是等价的, 特别地, 有

$$\|x\|_2 \le \|x\|_1 \le \sqrt{n}\,\|x\|_2,$$

$$\|x\|_\infty \le \|x\|_1 \le n\,\|x\|_\infty,$$

$$\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\,\|x\|_\infty.$$

(板书)

**定理 1.7 (Cauchy-Schwartz 不等式)** 设 $(\cdot,\cdot)$ 是 $\mathbb{R}^n$ 上的内积, 则对任意 $x, y \in \mathbb{R}^n$, 有

$$|(x,y)| \le \|x\|_2 \cdot \|y\|_2$$

更一般地, 我们有下面的 Holder 不等式.

**定理 1.8 (Holder 不等式)** 设 $(\cdot,\cdot)$ 是 $\mathbb{R}^n$ 上的内积, 则对任意 $x, y \in \mathbb{R}^n$, 有

$$|(x,y)| \le \|x\|_p \cdot \|y\|_q,$$

其中 $p, q > 0$, 且 $\dfrac{1}{p} + \dfrac{1}{q} = 1$.

### 矩阵与矩阵范数

首先简要给出一些有关矩阵的基础知识, 更多内容可参考有关矩阵理论的书籍, 如 [1, 2].

- 向量与矩阵: 定义, 基本运算, 行列式 ($\det(A)$).
- 特征值与特征向量: 特征多项式, 左特征向量, (右) 特征向量, 矩阵相似, 矩阵合同

  思考 $A^\intercal$ 和 $A$, 它们的特征值和特征向量有什么关系?
  $A^{-1}$ 和 $A$, 它们的特征值和特征向量有什么关系?
  设 $p(t)$ 是一个多项式, 则 $p(A)$ 的特征值与特征向量是什么?

- 矩阵的谱: $\sigma(A) \triangleq \{A\text{的所有特征值}\}$
- 矩阵谱半径: $\rho(A) \triangleq \max_{\lambda \in sigma(A)} |\lambda|$
- 矩阵的迹: $\mathrm{tr}(A) \triangleq a_{11} + a_{22} + \cdots + a_{nn}$
  性质: (1) $\det(A) = \lambda_1\lambda_2\cdots\lambda_n$; (2) $\mathrm{tr}(A) = \lambda_1 + \lambda_2 + \cdots + \lambda_n$
- 一些特殊矩阵: 对角矩阵、三角矩阵、三对角矩阵、对称矩阵、Hermite 对称矩阵、对称正定矩阵、正交矩阵、酉矩阵、初等置换阵、置换阵 ( 排列阵 )
- 上 Hessenberg 矩阵 (upper Hessenberg matrix): $a_{ij} = 0$ for $i - j > 1$ ;

$$\begin{bmatrix} * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ & * & * & \cdots & * \\ & & \ddots & \ddots & \vdots \\ & & & * & * \end{bmatrix}$$

下面给出几个基本结论.

**定理 1.9 (解的存在唯一性)** 教材 141 页

**定理 1.10 (对称正定矩阵的性质)** 教材 141 页

**定理 1.11 (对称矩阵正定的充分条件)** 教材 141 页

**定理 1.12 (Jordan 标准型)** 教材 142 页

**定义 1.8 (矩阵范数)** 若函数 $f:\mathbb{R}^{n\times n}\to\mathbb{R}$ 满足

(1) $f(A)\geq 0,\forall\, A\in\mathbb{R}^{n\times n}$ 且等号当且仅当 $A=0$ 时成立;

(2) $f(\alpha A)=|\alpha|\cdot f(A),\forall\, A\in\mathbb{R}^{n\times n},\alpha\in\mathbb{R}$;

(3) $f(A+B)\leq f(A)+f(B),\forall\, A,B\in\mathbb{R}^{n\times n}$;

(4) $f(AB)\leq f(A)f(B),\forall\, A,B\in\mathbb{R}^{n\times n}$

则称 $f(x)$ 为 $\mathbb{R}^{n\times n}$ 上的矩阵范数, 通常记作 $\|\cdot\|$.

一类常用的矩阵范数就是由向量范数导出的算子范数.

**引理 1.1 (算子范数, 诱导范数, 导出范数)** 设 $\|\cdot\|$ 是 $\mathbb{R}^n$ 上的向量范数, 则

$$\|A\|\triangleq\sup_{x\in\mathbb{R}^n,\,x\neq 0}\frac{\|Ax\|}{\|x\|}=\max_{\|x\|=1}\|Ax\|$$

是 $\mathbb{R}^{n\times n}$ 上的范数, 称为 算子范数, 也称为 诱导范数 或 导出范数. (板书)

♣ 对于算子范数, 我们有下面的性质: 设 $A\in\mathbb{R}^{n\times n},x\in\mathbb{R}^n$, 则

$$\|Ax\|\leq\|A\|\cdot\|x\|.$$

**例 1.13** $\mathbb{R}^{n\times n}$ 上常见的矩阵范数:

- $F$-范数

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

- $p$-范数 (算子范数)

$$\|A\|_p=\sup_{x\neq 0}\frac{\|Ax\|_p}{\|x\|_p},\quad p=1,2,\infty.$$

**引理 1.2** 可以证明:

(1) 1-范数 (也称为 列范数): $\|A\|_1=\max\limits_{1\leq j\leq n}\left(\sum\limits_{i=1}^{n}|a_{ij}|\right)$;

(2) $\infty$-范数 (也称为 行范数): $\|A\|_\infty = \max\limits_{1\le i\le n}\left(\sum\limits_{j=1}^{n}|a_{ij}|\right)$;

(3) 2-范数 (也称为 谱范数): $\|A\|_2 = \sqrt{\rho(A^\intercal A)}$.

证明. 结论 (1) 的证明留作练习. 这里只证明 (2), (3). (板书) □

**例 1.14** 设 $A = \begin{bmatrix} 1 & -2 \\ -3 & 4 \end{bmatrix}$, 计算 $\|A\|_1, \|A\|_2, \|A\|_\infty, \|A\|_F$. (板书)

♣ 计算 2-范数时需要求谱半径, 因此通常比计算 1-范数和 $\infty$-范数更困难. 但在某些情况下可以用下面的范数等价性来估计一个矩阵的 2-范数.

**定理 1.13 (矩阵范数的连续性)** 设 $\|\cdot\|$ 是 $\mathbb{R}^{n\times n}$ 上的一个矩阵范数, 则 $f(A) \triangleq \|A\|$ 关于 $A$ 的每个分量是连续的. (板书)

**定理 1.14 (矩阵范数的等价性)** $\mathbb{R}^{n\times n}$ 空间上的所有范数都是等价的, 特别地, 有

$$\frac{1}{\sqrt{n}}\|A\|_1 \le \|A\|_2 \le \sqrt{n}\,\|A\|_1,$$
$$\frac{1}{\sqrt{n}}\|A\|_\infty \le \|A\|_2 \le \sqrt{n}\,\|A\|_\infty,$$
$$\frac{1}{n}\|A\|_\infty \le \|A\|_1 \le n\,\|A\|_\infty,$$
$$\frac{1}{\sqrt{n}}\|A\|_1 \le \|A\|_F \le \sqrt{n}\,\|A\|_2.$$

(板书)

**定理 1.15** 若 $A$ 是对称矩阵, 则 $\|A\|_2 = \rho(A)$. (证明留作练习)

### 算子范数的性质

**定理 1.16** 设 $\|\cdot\|$ 是 $\mathbb{R}^{n\times n}$ 上的任一算子范数, 则

$$\rho(A) \le \|A\|, \quad \forall\, A \in \mathbb{R}^{n\times n}.$$

(板书)

♣ 上述性质对 F-范数也成立, 即 $\rho(A) \le \|A\|_F$.

**推论 1.17** 设 $A \in \mathbb{R}^{n\times n}$, 则 $\|A\|_2^2 \le \|A\|_1 \cdot \|A\|_\infty$, 且

$$\max_{1\le i,j\le n}\{|a_{ij}|\} \le \|A\|_2 \le n \max_{1\le i,j\le n}\{|a_{ij}|\}.$$

**定理 1.18** 设 $A \in \mathbb{R}^{n\times n}$, 则对任意 $\varepsilon > 0$, 总存在一个算子范数 $\|\cdot\|_\varepsilon$, 使得

$$\|A\|_\varepsilon \leq \rho(A) + \varepsilon.$$

证明. 证明过程可参见相关资料. □

**定理 1.19** 设 $\|\cdot\|$ 是 $\mathbb{R}^{n\times n}$ 上的任一算子范数, 若 $\|B\| < 1$, 则 $I \pm B$ 非奇异, 且

$$\left\|(I \pm B)^{-1}\right\| \leq \frac{1}{1 - \|B\|}.$$

证明. 这里仅证明减号情形, 加号情形的证明类似.

先证明 $I - B$ 非奇异. 反证法, 假定 $I - B$ 奇异, 则 $(I - B)x = 0$ 有非零解, 设为 $\tilde{x}$, 即 $B\tilde{x} = \tilde{x}$. 所以 $\|B\| \geq \frac{\|B\tilde{x}\|}{\|\tilde{x}\|} = 1$, 与条件 $\|B\| < 1$ 矛盾. 因此 $I - B$ 非奇异.

由 $(I - B)^{-1}(I - B) = I$ 可知

$$(I - B)^{-1} = I + (I - B)^{-1}B.$$

从而

$$\|(I - B)^{-1}\| = \|I + (I - B)^{-1}B\| \leq \|I\| + \|(I - B)^{-1}\| \cdot \|B\|.$$

解得

$$\|(I - B)^{-1}\| \leq \frac{\|I\|}{1 - \|B\|} = \frac{1}{1 - \|B\|}.$$

□

除此之外, 算子范数还有下面的性质:

(1) 对任意算子范数 $\|\cdot\|$, 有 $\|A^k\| \leq \|A\|^k$ ;

(2) 对任意算子范数 $\|\cdot\|$, 有 $\|Ax\| \leq \|A\| \cdot \|x\|$, $\|AB\| \leq \|A\| \cdot \|B\|$;

(3) $\|Ax\|_2 \leq \|A\|_F \cdot \|x\|_2$, $\|AB\|_F \leq \|A\|_F \cdot \|B\|_F$;

(4) $F$-范数不是算子范数;

(5) $\|\cdot\|_2$ 和 $\|\cdot\|_F$ 是 酉不变范数, 即对任意酉矩阵 (或正交矩阵) $U, V$, 有

$$\|UA\|_2 = \|AV\|_2 = \|UAV\|_2 = \|A\|_2 ,$$
$$\|UA\|_F = \|AV\|_F = \|UAV\|_F = \|A\|_F$$

(6) $\|A^\intercal\|_2 = \|A\|_2$, $\|A^\intercal\|_1 = \|A\|_\infty$ .

## 1.3 数值计算中的误差

数值方法的特点之一就是所求得的解是近似解, 总是存在一定的误差. 因此, 误差分析是数值分析中一个很重要的课题.

误差是人们用来描述数值计算中近似解的精确程度, 是科学计算中的一个十分重要的概念.

科学计算中误差的来源主要有以下几个方面:

- 模型误差: 从实际问题中抽象出数学模型, 往往是抓住主要因素, 忽略次要因素, 因此, 数学模型与实际问题之间总会存在一定的误差.
- 观测误差: 模型中往往包含各种数据或参量, 这些数据一般都是通过测量和实验得到的, 也会存在一定的误差.
- 截断误差: 也称方法误差, 是指对数学模型进行数值求解时产生的误差.
- 舍入误差: 由于计算机的机器字长有限, 做算术运算时存在一定的精度限制, 也会产生误差.

在数值分析中, 我们总假定数学模型是准确的, 因而不考虑模型误差和观测误差, 主要研究截断误差和舍入误差对计算结果的影响.

**例 1.15** 近似计算 $\int_0^1 e^{-x^2}\,\mathrm{d}x$ 的值.

**解.** 这里我们采用 Taylor 展开, 即

$$\begin{aligned}\int_0^1 e^{-x^2}\,\mathrm{d}x &= \int_0^1 \left(x - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!} - \cdots\right) \\ &= 1 - \frac{1}{3} + \frac{1}{2!}\times\frac{1}{5} - \frac{1}{3!}\times\frac{1}{7} + \frac{1}{4!}\times\frac{1}{9} - \cdots \\ &\triangleq S_4 + R_4\end{aligned}$$

其中 $S_4$ 为前四项的部分和, $R_4$ 为剩余部分. 如果我们以 $S_4$ 作为定积分的近似值, 则 $R_4$ 就是由此产生的误差, 这种误差就称为截断误差.

在计算 $S_4$ 的值, 假定我们保留小数点后 4 位有效数字, 则

$$S_4 = 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42} \approx 1 - 0.3333 + 0.1000 - 0.0238 = 0.7429$$

这就是我们最后得到的近似值. 这里, 在计算 $S_4$ 时所产生的误差就是舍入误差. □

### 1.3.1 绝对误差

**定义 1.9** 设 $\tilde{x}$ 是 $x$ 的近似值, 则称

$$\epsilon \triangleq \tilde{x} - x$$

为近似值 $\tilde{x}$ 的绝对误差, 简称误差.

由定义可知:

- 绝对误差可能是正的, 也可能是负的,
- 由于精确值通常是不知道的, 因此绝对误差也是不可知的.

**定义 1.10** 设 $\tilde{x}$ 是 $x$ 的近似值, 若存在 $\varepsilon > 0$ 使得

$$|\epsilon| = |\tilde{x} - x| \leq \varepsilon,$$

则称 $\varepsilon$ 为绝对误差限, 简称误差限.

♣ 在工程中, 通常用 $x = \tilde{x} \pm \varepsilon$ 表示 $\tilde{x}$ 的误差限为 $\varepsilon$.

- 误差估计所求的是误差限;

- 误差限越小越好;
- 但绝对误差限却不能很好地表示近似值的精确程度.

### 1.3.2 相对误差

**定义 1.11** 设 $\tilde{x}$ 是 $x$ 的近似值, 称

$$\epsilon_r \triangleq \frac{\tilde{x}-x}{x}$$

为近似值 $\tilde{x}$ 的 相对误差.

♣ 由于精确值 $x$ 通常难以求出, 因此我们有时也采用下面的定义

$$\epsilon_r \triangleq \frac{\tilde{x}-x}{\tilde{x}}.$$

**定义 1.12** 设 $\tilde{x}$ 是 $x$ 的近似值, 若存在 $\varepsilon_r > 0$ 使得

$$|\epsilon_r| \leq \varepsilon_r,$$

则称 $\varepsilon_r$ 为相对误差限.

- 近似值的精确程度取决于相对误差的大小;
- 实际计算中我们所能得到的是绝对误差限或相对误差限.

### 1.3.3 有效数字

**定义 1.13** 若近似值 $\tilde{x}$ 的误差限是某一位的半个单位, 且该位到 $\tilde{x}$ 的第一位非零数字共有 $n$ 位, 则称 $\tilde{x}$ 有 $n$ 位有效数字.

关于有效数字的判断, 我们可以使用下面的方法.

**性质 1.1** 设 $\tilde{x}$ 是 $x$ 的近似值, 若 $\tilde{x}$ 可表示为

$$\tilde{x} = \pm a_1.a_2 \cdots a_n \cdots \times 10^m,$$

其中 $a_i$ 是 0 到 9 中的数字, 且 $a_1 \neq 0$. 若

$$0.5 \times 10^{m-n} < |\tilde{x}-x| \leq 0.5 \times 10^{m-n+1},$$

则 $\tilde{x}$ 有 $n$ 位有效数字.

♣ 换而言之, 若 $0.5 \times 10^{k-1} < |\tilde{x}-x| \leq 0.5 \times 10^k$, 则 $\tilde{x}$ 至少有 $m-k+1$ 位有效数字.

**例 1.16** 已知 $\pi = 3.14159265\cdots$, 近似值

$$x_1 = 3.1415, \quad x_2 = 3.1416,$$

试问: $x_1, x_2$ 分别有几位有效数字? 解答: 4, 5

**例 1.17** 根据四舍五入原则写出下列各数的具有 5 位有效数字的近似值:

$$187.9325, \quad 0.03785551, \quad 8.000033.$$

解答: 187.93, 0.037856, 8.0000

♣ 按四舍五入原则得到的数字是有效数字.

♣ 一个数末尾的 0 不可以随意添加或省略.

**定理 1.20 (有效数字与相对误差限)** *设 $\tilde{x}$ 是 $x$ 的近似值, 若 $\tilde{x}$ 可表示为*

$$\tilde{x} = \pm a_1.a_2 \dots a_n \dots \times 10^m,$$

*其中 $a_i$ 是 0 到 9 中的数字, 且 $a_1 \neq 0$. 若 $\tilde{x}$ 具有 $n$ 位有效数字, 则其相对误差满足*

$$|\epsilon_r| \leq \frac{1}{2a_1} \times 10^{-n+1}.$$

*反之, 若 $\tilde{x}$ 的相对误差满足*

$$|\epsilon_r| \leq \frac{1}{2(a_1+1)} \times 10^{-n+1}.$$

*则 $\tilde{x}$ 至少有 $n$ 位有效数字.*

**证明.** 由 $\tilde{x}$ 的表达式可知

$$a_1 \times 10^m \leq |\tilde{x}| \leq (a_1+1) \times 10^m.$$

若 $\tilde{x}$ 具有 $n$ 位有效数字, 则

$$|\epsilon_r| = \frac{|\tilde{x}-x|}{|\tilde{x}|} \leq \frac{0.5 \times 10^{m-n+1}}{a_1 \times 10^m} = \frac{1}{2a_1} \times 10^{-n+1}.$$

反之, 若 $|\epsilon_r| \leq \frac{1}{2(a_1+1)} \times 10^{-n+1}$, 则

$$|\tilde{x}-x| = |\tilde{x}| \cdot |\epsilon_r| \leq 0.5 \times 10^{m-n+1}.$$

故 $\tilde{x}$ 至少有 $n$ 位有效数字. □

♣ 从这个定理可以看出, 有效数字越多, 相对误差越小. 同样, 如果相对误差越小, 则有效数字越多.

### 1.3.4 误差估计基本公式

**四则运算**

设 $\tilde{x}_1$ 和 $\tilde{x}_2$ 的误差限分别为 $\varepsilon(\tilde{x}_1)$ 和 $\varepsilon(\tilde{x}_2)$, 则

$$\begin{aligned}
&\varepsilon(\tilde{x}_1 \pm \tilde{x}_2) \leq \varepsilon(\tilde{x}_1) + \varepsilon(\tilde{x}_2),\\
&\varepsilon(\tilde{x}_1\tilde{x}_2) \leq |\tilde{x}_2|\varepsilon(\tilde{x}_1) + |\tilde{x}_1|\varepsilon(\tilde{x}_2) + \varepsilon(\tilde{x}_1)\varepsilon(\tilde{x}_2) \lessapprox |\tilde{x}_2|\varepsilon(\tilde{x}_1) + |\tilde{x}_1|\varepsilon(\tilde{x}_2),\\
&\varepsilon\left(\frac{\tilde{x}_1}{\tilde{x}_2}\right) \lessapprox \frac{|\tilde{x}_2|\varepsilon(\tilde{x}_1) + |\tilde{x}_1|\varepsilon(\tilde{x}_2)}{|\tilde{x}_2|^2}.
\end{aligned}$$

**单变量可微函数**

一般地, 设 $\tilde{x}$ 是 $x$ 的近似值, 若 $f(x)$ 可导, 则有

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

由于 $\tilde{x} - x$ 相对较小, 所以当 $|f''(x)|$ 与 $|f'(x)|$ 的比值不是很大时, 我们可以忽略二阶项, 即

$$|f(\tilde{x}) - f(x)| \approx |f'(x)| \times |\tilde{x} - x|.$$

因此, 可得函数值的误差限

$$\varepsilon(f(\tilde{x})) \approx |f'(x)|\,\varepsilon(\tilde{x}) \approx |f'(\tilde{x})|\,\varepsilon(\tilde{x}).$$

**多变量可微函数**

关于多元函数 $f(x_1, x_2, \ldots, x_n)$, 我们可以得到类似的结论:

$$\varepsilon(f(\tilde{x})) \approx \sum_{k=1}^{n} \left|\frac{\partial f(\tilde{x})}{\partial x_k}\right| \varepsilon(\tilde{x}_k),$$

其中 $\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n]^\mathsf{T}$ 是 $x = [x_1, x_2, \ldots, x_n]^\mathsf{T}$ 的近似值.

**例 1.18** 测得某场地的长 $L$ 和宽 $D$ 分别为: $\tilde{L} = 110\text{m}$, $\tilde{W} = 80\text{m}$, 其测量误差限分别为 0.2m 和 0.1m. 试求面积 $S = L \times W$ 的绝对误差限和相对误差限.

**解.** 由于 $\varepsilon(\tilde{L}) = 0.2\text{m}$, $\varepsilon(\tilde{W}) = 0.1\text{m}$, 故

$$\begin{aligned}
\varepsilon(\tilde{S}) &\approx \left|\frac{\partial S(\tilde{L}, \tilde{W})}{\partial L}\right| \varepsilon(\tilde{L}) + \left|\frac{\partial S(\tilde{L}, \tilde{W})}{\partial W}\right| \varepsilon(\tilde{W})\\
&= |\tilde{W}| \cdot \varepsilon(\tilde{L}) + |\tilde{L}| \cdot \varepsilon(\tilde{W})\\
&= 80 \times 0.2 + 110 \times 0.1 = 27(\text{m}^2).
\end{aligned}$$

相对误差限

$$\varepsilon_r(\tilde{S}) = \frac{\varepsilon(\tilde{S})}{|\tilde{S}|} \approx \frac{27}{110 \times 80} \approx 0.0031.$$

□

## 1.4 误差分析与数值稳定性

- 数值计算中的误差分析很重要, 但也很复杂;

- 在计算过程中, 误差会传播、积累、对消;
- 实际计算中的运算次数通常都在千万次以上, 因此对每一步运算都做误差分析比较不切实际.

### 1.4.1 误差分析方法

误差分析可分为定量分析和定性分析.

**定量分析**

- 主要方法有: 向后误差分析法, 向前误差分析法, 区间误差分析法, 概率分析法等.
- 向后误差分析法比较有效, 其它方法仍不完善.
- 定量分析通常工作量大, 而且得到的误差界也往往不太实用.

**定性分析**

- 目前在数值计算中更关注的是误差的定性分析;
- 定性分析包括研究数学问题的适定性, 数学问题与原问题的相容性, 数值算法的稳定性, 避免扩大误差的准则等;
- 定性分析的核心是原始数据的误差和计算过程中产生的误差对最终计算结果的影响.

♣ 算法有 “优劣” 之分, 问题有 “好坏” 之别, 即使不能定量地估计出最终误差, 但是若能确保计算过程中误差不会被任意放大, 那就能放心地实施计算, 这就是研究定性分析的初衷.

### 1.4.2 数值稳定性

**数学问题的适定性**

如果数学问题满足

(1) 对任意满足一定条件的输入数据, 存在一个解,

(2) 对任意满足一定条件的输入数据, 解是唯一的,

(3) 问题的解关于输入数据是连续的,

则称该数学问题是适定的 (well-posed), 否则就称为不适定的 (ill-posed).

♣ 如果输入数据的微小扰动会引起输出数据 (即计算结果) 的很大变化 (误差), 则称该数值问题是病态的, 否则就是 良态的.

**例 1.19** 解线性方程组 $\begin{cases} x+\alpha y=1 \\ \alpha x+y=0 \end{cases}$

**解.** 易知当 $\alpha=1$ 时, 方程组无解. 当 $\alpha\neq 1$ 时, 解为

$$x=\frac{1}{1-\alpha^2},\quad y=\frac{-\alpha}{1-\alpha^2}.$$

如果 $\alpha\approx 1$, 则 $\alpha$ 的微小误差可能会引起解的很大变化. 比如当 $\alpha=0.999$ 时, $x\approx 500.25$. 假定输入数据 $\alpha$ 带有 0.0001 的误差, 即输入数据为 $\alpha^*=0.9991$, 则此时有 $\tilde{x}\approx 555.81$, 解的误差约为 55.56, 是输

入数据误差的五十多万倍, 因此该问题的病态的. □

### 病态问题与条件数

设 $f(x)$ 可导, $\tilde{x}$ 是精确值 $x_*$ 的近似值, 则由 Taylor 公式可知

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

当 $\tilde{x}$ 很接近 $x_*$ 时, $(\tilde{x} - x_*)^2$ 非常小, 因此有

$$\left|\frac{f(\tilde{x}) - f(x_*)}{f(x_*)}\right| \approx \left|\frac{x_* f'(x_*)}{f(x_*)}\right| \times \left|\frac{\tilde{x} - x_*}{x_*}\right|,$$

即函数值的相对误差大约是输入数据相对误差的 $\left|\frac{x_* f'(x_*)}{f(x_*)}\right|$ 倍. 这个值就定义为函数 $f(x)$ 的 **条件数**, 几位 $C_p$, 即

$$C_p \triangleq \left|\frac{x f'(x)}{f(x)}\right|.$$

- 一般情况下, 条件数大于 10 时, 就认为问题是病态的;
- 条件数越大问题病态就越严重;
- 病态是问题本身固有的性质, 与数值算法无关;
- 对于病态问题, 选择数值算法时需要谨慎.

### 算法的稳定性

在数值计算过程中, 如果误差不增长或能得到有效控制, 则称该算法是 **稳定** 的, 否则为 **不稳定** 的.

♣ 在数值计算中, 不要采用不稳定的算法!

**例 1.20** 近似计算

$$S_n = \int_0^1 \frac{x^n}{x+5}\,\mathrm{d}x, \quad n = 1, 2, \ldots, 8.$$

**解.** 通过观察可知

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

因此,

$$S_n = \frac{1}{n} - 5S_{n-1}. \tag{1.2}$$

易知 $S_0 = \ln 6 - \ln 5 \approx 0.182$ (保留三位有效数字), 利用上面的递推公式可得 (保留三位有效数字)

$$S_1 = 0.0900, \quad S_2 = 0.0500, \quad S_3 = 0.0833, \quad S_4 = -0.166,$$
$$S_5 = 1.03, \quad S_6 = -4.98, \quad S_7 = 25.0, \quad S_8 = -125.$$

另一方面, 我们有

$$\frac{1}{6(n+1)} = \int_0^1 \frac{x^n}{6}\,\mathrm{d}x \le \int_0^1 \frac{x^n}{x+5}\,\mathrm{d}x \le \int_0^1 \frac{x^n}{5}\,\mathrm{d}x = \frac{1}{5(n+1)}. \tag{1.3}$$

因此, 上面计算的 $S_4, \ldots, S_8$ 显然是不对的. 原因是什么呢? 误差!

设 $S_n^*$ 是 $S_n$ 的近似值, 则

$$e(S_n^*) = S_n^* - S_n = \left(\frac{1}{n} - 5S_{n-1}^*\right) - \left(\frac{1}{n} - 5S_{n-1}\right) \approx -5(S_{n-1}^* - S_{n-1}) = -5e(S_{n-1}^*).$$

即误差是以 5 倍速度增长, 这说明计算过程是不稳定的, 因此我们不能使用该算法.

事实上, 递推公式 (1.2) 可以改写为

$$S_{n-1} = \frac{1}{5n} - \frac{1}{5}S_n.$$

因此, 我们可以先估计 $S_8$ 的值, 然后通过反向递推, 得到其它值.

我们可以根据 (1.3) 对 $S_8$ 做简单的估计, 即

$$S_8 \approx \frac{1}{2}\left(\int_0^1 \frac{x^n}{6}\,\mathrm{d}x + \int_0^1 \frac{x^n}{5}\,\mathrm{d}x\right) \approx 0.0204.$$

于是

$$S_7 = 0.0209, \quad S_6 = 0.0244, \quad S_5 = 0.0285, \quad S_4 = 0.0343,$$
$$S_3 = 0.0431, \quad S_2 = 0.0580, \quad S_1 = 0.0884, \quad S_0 = 0.182.$$

通过误差分析可知, 误差是以 $\frac{1}{5}$ 的速度减小, 因此计算过程是稳定的. □

♣ 在数值计算中, 误差不可避免, 算法的稳定性是一个非常重要的性质.

♣ 用计算机进行整数之间的加减运算和乘法运算时, 没有误差. (不考虑溢出情况)

### 1.4.3 避免误差危害

为了尽量避免误差给计算结果带来的危害, 在计算过程中, 我们应注意以下几点.

#### (1) 避免相近的数相减

如果两个相近的数相减, 则会损失有效数字, 如 $0.12346 - 0.12345 = 0.00001$, 操作数有 5 位有效数字, 但结果却只有 1 为有效数字.

**例 1.21** 计算 $\sqrt{9.01} - 3$, 计算过程中保留 3 位有效数字.

**解.** 如果直接计算的话, 可得

$$\sqrt{9.01} = 3.0016662039607\cdots \approx 3.00.$$

所以 $\sqrt{9.01} - 3 \approx= 0.00$, 一个有效数字都没有!

但如果换一种计算方法, 如

$$\sqrt{9.01} - 3 = \frac{9.01 - 3^2}{\sqrt{9.01} + 3} \approx \frac{0.01}{3.00 + 3} \approx 0.00167.$$

通过精确计算可知 $\sqrt{9.01} - 3 = 0.0016662039607\cdots$. 因此第二种计算能得到三位有效数字! □

通过各种等价公式来计算两个相近的数相减, 是避免有效数字损失的有效手段之一. 下面给出几个

常用的等价公式:

$$\sqrt{x+\varepsilon}-\sqrt{\varepsilon}=\frac{\varepsilon}{\sqrt{x+\varepsilon}+\sqrt{x}}$$
$$\ln(x+\varepsilon)-\ln(x)=\ln\left(1+\frac{\varepsilon}{x}\right)$$
$$1-\cos(x)=2\sin^2\frac{x}{2},\quad |x|\ll 1$$
$$\mathrm{e}^x-1=x\left(1+\frac{1}{2}x+\frac{1}{6}x^2+\cdots\right),\quad |x|\ll 1$$

**例 1.22** 在 MATLAB 中用双精度数计算

$$E_1=\frac{1-\cos(x)}{\sin^2(x)}\quad 和\quad E_2=\frac{1}{1+\cos(x)}.$$

**解.** MATLAB 程序见 `ex_significance.m`

| $x$ | $E_1$ | $E_2$ |
|---|---|---|
| 1.0000000000 | 0.649223205204762 | 0.649223205204762 |
| 0.1000000000 | 0.501252086288577 | 0.501252086288571 |
| 0.0100000000 | 0.500012500208481 | 0.500012500208336 |
| 0.0010000000 | 0.500000124992189 | 0.500000125000021 |
| 0.0001000000 | 0.499999998627931 | 0.500000001250000 |
| 0.0000100000 | 0.500000041386852 | 0.500000000012500 |
| 0.0000010000 | 0.500044450291337 | 0.500000000000125 |
| 0.0000001000 | 0.499600361081322 | 0.500000000000001 |
| 0.0000000100 | 0.000000000000000 | 0.500000000000000 |
| 0.0000000010 | 0.000000000000000 | 0.500000000000000 |
| 0.0000000001 | 0.000000000000000 | 0.500000000000000 |

□

**(2) 避免数量级相差很大的数相除**

可能会产生溢出, 即超出计算机所能表示的数的范围.

**(3) 避免大数吃小数**

如 $(10^9+10^{-9}-10^9)/10^{-9}$, 直接计算的话, 结果为 0.

另外, 在对一组数求和时, 应按绝对值从小到大求和.

**(4) 简化计算**

尽量减少运算次数, 避免误差积累.

**例 1.23** 多项式计算. 设多项式

$$p(x) = 5x^5 + 4x^4 + 3x^3 + 2x^2 + 2x + 1.$$

试计算 $p(3)$ 的值.

**解.** 如果直接计算的话, 可得

$$p(3) = 5 \times 3^5 + 4 \times 3^4 + 3 \times 3^3 + 2 \times 3^2 + 2 \times 3 + 1.$$

需要做 15 次乘法和 5 次加法.

在实际计算中, 当计算 $x^k$ 时, 由于前面已经计算出 $x^{k-1}$, 因此只需做一次乘法就可以了. 这样整个计算过程可以减少到 9 次乘法和 5 次加法. 但这并不是最佳方案.

事实上, 我们可以将多项式改写为

$$p(x) = ((((5x + 4)x + 3)x + 2)x + 2)x + 1.$$

这样就只需做 5 次乘法和 5 次加法. 显然这是更佳的计算方案. □

♣ 在计算多项式的值时, 我们都是将多项式改写成

$$\begin{aligned} p(x) &= a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0 \\ &= ((\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots)x + a_1)x + a_0. \end{aligned}$$

这种方法就是有名的 秦九韶方法 或 Horner 方法. 如果直接计算的话, 需要 $\frac{n(n+1)}{2}$ 次乘法和 $n$ 次加法. 但如果采用秦九韶方法或 Horner 方法的话, 只需做 $n$ 次乘法和 $n$ 次加法.

**(5) 选用稳定的算法**

不要使用不稳定的算法.

## 1.5 课后练习

**练习 1.1** 设 $x > 0$, $x$ 的相对误差是 $\delta$, 求 $\ln x$ 的误差.
(提示: 本题计算的是误差限)

**练习 1.2** 设 $x$ 的相对误差为 2%, 求 $x^n$ 的相对误差.
(提示: 本题计算的是误差限)

**练习 1.3** 下列各数都是经过四舍五入得到的近似数, 即误差限不超过最后一位的半个单位, 试指出它们是几位有效数字.

$$\tilde{x}_1 = 1.1021,\ \tilde{x}_2 = 0.031,\ \tilde{x}_3 = 385.6,\ \tilde{x}_4 = 56.430,\ \tilde{x}_5 = 7 \times 1.0$$

(提示: $\tilde{x}_5$ 中的 7 表示整数, 不考虑舍入误差.)

**练习 1.4** 利用公式 (2.3) 求下列各近似值的误差限:
(1) $\tilde{x}_1 + \tilde{x}_2 + \tilde{x}_4$;

(2) $\tilde{x}_1\tilde{x}_2\tilde{x}_3$;

(3) $\tilde{x}_2/\tilde{x}_4$;

其中 $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4$ 均为第 3 题所给的数.

**练习 1.5** 计算球体积, 要使相对误差限为 1%, 问半径 $R$ 所允许的相对误差限是多少?

**练习 1.6** 设 $Y_0 = 28$, 按递推公式

$$Y_n = Y_{n-1} - \frac{1}{100}\sqrt{783}, \quad n = 1, 2, \ldots$$

计算到 $Y_{100}$. 若取 $\sqrt{783} \approx 27.982$（5 位有效数字）, 试问 $Y_{100}$ 将有多大误差?

**练习 1.7** 求方程 $x^2 - 56x + 1 = 0$ 的两个根, 使它至少具有 4 位有效数字. ($\sqrt{783} \approx 27.982$)

**练习 1.8** 当 $x \approx y$ 时计算 $\ln(x) - \ln(y)$, 有效数字会损失. 改用 $\ln(x) - \ln(y) = \ln\dfrac{x}{y}$ 是否能减少舍入误差?

(提示: 考虑对数函数何时出现病态)

**练习 1.9** 正方形的边长大约为 100 cm, 应怎样测量才能使其面积误差不超过 1 $\text{cm}^2$?

**练习 1.10** 设 $S = \dfrac{1}{2}gt^2$, 假定 $g$ 是准确的, 而对 $t$ 的测量有 $\pm 0.1$ 秒的误差. 证明: 当 $t$ 增加时 $S$ 的绝对误差增加, 而相对误差却减少.

**练习 1.11** 序列 $\{y_n\}$ 满足递推关系

$$y_n = 10y_{n-1} - 1, \quad n = 1, 2, \ldots,$$

若 $y_0 = \sqrt{2} \approx 1.41$（保留三位有效数字）, 计算到 $y_{10}$ 时, 误差有多大? 这个计算过程稳定吗?

(提示: 整数运算 (加, 减, 乘, 幂) 在不溢出的情况下不用考虑舍入误差.)

# 第二讲 函数插值

## 2.1 引言

### 2.1.1 为什么要插值

- 许多实际问题都可用函数来表示某种内在规律的数量关系;
- 但函数表达式无法给出，只有通过实验或观测得到的数据表;
- 如何根据这些数据推测或估计其它点的函数值?

**例 2.1** 已测得在某处海洋不同深度处的水温如下:

| 深度 (M) | 466 | 741 | 950 | 1422 | 1634 |
|---|---|---|---|---|---|
| 水温 (°C) | 7.04 | 4.28 | 3.40 | 2.54 | 2.13 |

根据这些数据, 请合理地估计出其它深度 (如 500, 600, 800 米 ...) 处的水温.

### 2.1.2 什么是插值

**定义 2.1** 已知函数 $y = f(x)$ 在区间 $[a, b]$ 上有定义, 且已经测得其在点

$$a \le x_0 < x_1 < \cdots < x_n \le b \tag{2.1}$$

处的值为 $y_0 = f(x_0), \cdots, y_n = f(x_n)$. 如果存在一个简单易算的函数 $p(x)$, 使得

$$p(x_i) = y_i, \quad i = 0, 1, \ldots, n, \tag{2.2}$$

则称 $p(x)$ 为 $f(x)$ 的插值函数. 区间 $[a, b]$ 称为插值区间, $x_i$ $(i = 0, 1, \ldots, n)$ 称为插值节点, 条件 (2.2) 称为插值条件.

♣ 插值节点无需递增排列, 但必须确保互不相同!

- 求插值函数 $p(x)$ 的方法就称为插值法.
- 常见的插值方法有:
  - 多项式插值: $p(x)$ 为多项式, 多项式是常用的插值函数;
  - 分段多项式插值: $p(x)$ 为分段多项式, 用分段多项式插值是常用的插值法;
  - 有理插值: $p(x)$ 为有理函数;
  - 三角插值: $p(x)$ 为三角函数;
  - ... ...

♣ 需要掌握多项式插值和分段多项式插值.

### 2.1.3 多项式插值

**定义 2.2 (多项式插值)** 已知函数 $y = f(x)$ 在区间 $[a, b]$ 上 $n+1$ 个点

$$a \leq x_0 < x_1 < \cdots < x_n \leq b$$

处的函数值为 $y_0 = f(x_0), \cdots, y_n = f(x_n)$. 多项式插值就是寻找一个次数不超过 $n$ 的 多项式的函数

$$p(x) = c_0 + c_1 + \cdots + c_n x^n, \tag{2.3}$$

使得

$$p(x_i) = y_i, \quad i = 0, 1, \ldots, n.$$

♣ 需要指出的是, $p(x)$ 的次数有可能小于 $n$.

**定理 2.1 (多项式插值存在唯一性)** 满足插值条件的多项式 $p(x)$ 存在唯一.

证明. 待定系数法. 设 $p(x)$ 的表达式为 (2.3). 将插值条件 $p(x_i) = y_i$ 代入, 得到一个关于 $c_0, c_1, \ldots, c_n$ 的线性方程组, 其系数矩阵正好是一个关于 $x_0, x_1, \ldots, x_n$ 的 Vandermonde 矩阵. 因此当插值节点互不相同时, 其行列式不为 0. 所以系数矩阵可逆, 即解存在唯一. □

♣ 该定理的证明过程事实上也给出了一种求 $p(x)$ 的方法, 但这个方法比较复杂, 当插值点较多时, 需要解一个很大的线性方程组, 不实用, 后面将给出几个较简单的计算方法.

**例 2.2** 线性插值: 求一个一次多项式 $p(x)$, 满足:

$$p(x_0) = y_0, p(x_1) = y_1.$$

解. 由于 $p(x)$ 是一次多项式, 即代表一条直线. 因此由点斜式可知

$$\begin{aligned} p(x) &= y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \\ &= y_0 - \frac{y_0(x - x_0)}{x_1 - x_0} + y_1 \frac{x - x_0}{x_1 - x_0} \\ &= y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}. \end{aligned}$$

记 $l_0(x) = \dfrac{x - x_1}{x_0 - x_1}, l_1(x) = \dfrac{x - x_0}{x_1 - x_0}$, 则 $p(x)$ 就可以表示成 $l_0(x)$ 和 $l_1(x)$ 的线性组合, 即

$$p(x) = y_0 l_0(x) + y_1 l_1(x).$$

我们进一步观察可知

$$\begin{aligned} l_0(x_0) = 1, \quad l_0(x_1) = 0; \\ l_1(x_0) = 0, \quad l_1(x_1) = 1. \end{aligned}$$

□

**例 2.3** 抛物线插值: 求一个二次多项式 $p(x)$, 满足:

$$p(x_0)=y_0, p(x_1)=y_1, p(x_2)=y_2.$$

**解.** 借鉴线性插值思想, 如果能构造出三个二次多项式 $l_0(x), l_1(x), l_2(x)$, 满足

$$\begin{aligned} &l_0(x_0)=1, \quad l_0(x_1)=0, \quad l_0(x_2)=0;\\ &l_1(x_0)=0, \quad l_1(x_1)=1, \quad l_1(x_2)=0;\\ &l_2(x_0)=0, \quad l_2(x_1)=0, \quad l_2(x_2)=1. \end{aligned}$$

则由插值条件可知, $p(x)$ 可以表示成

$$p(x)=y_0l_0(x)+y_1l_1(x)+y_2l_2(x).$$

现在的问题是如何构造 $l_0(x), l_1(x), l_2(x)$. 我们可以使用待定系数法.

由于 $l_0(x)$ 是二次多项式, 且满足 $l_0(x_1)=l_0(x_2)=0$, 因此 $l_0(x)$ 可以写成

$$l_0(x)=\alpha(x-x_1)(x-x_2),$$

其中 $\alpha$ 是待定系数 (常数). 将 $l_0(x_0)=y_0$ 代入可得 $\alpha=\dfrac{1}{(x_0-x_1)(x_0-x_2)}$. 所以

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

同理可得

$$\begin{aligned} l_1(x)&=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)},\\ l_2(x)&=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}. \end{aligned}$$

□

我们注意到, $p(x)$ 之所以可以写成 $l_0(x)$, $l_1(x)$ 和 $l_2(x)$ 的线性组合, 是因为 $l_0(x)$, $l_1(x)$, $l_2(x)$ 组成了线性空间

$$Z_2(x)\triangleq\{\text{次数不超过 2 的多项式的全体}\}$$

的一组基, 而 $p(x)\in Z_2(x)$, 因此 $p(x)$ 可以由 $l_0(x)$, $l_1(x)$, $l_2(x)$ 线性表出.

♣ 这种利用基函数来计算插值函数的方法就是基函数插值法.

### 2.1.4 基函数插值法

记

$$H_n(x)\triangleq\{\text{次数不超过 } n \text{ 的多项式的全体}\},$$

则 $H_n$ 构成一个 $n+1$ 维的线性空间. 易知, $n$ 次多项式插值就是在 $H_n$ 中寻找一个多项式 $p(x)$, 使得插值条件 (2.2) 成立.

设 $\{\phi_0(x), \phi_1(x), \ldots, \phi_n(x)\}$ 是 $H_n$ 的一组基, 则 $p(x)$ 可以表示成

$$p(x)=a_0\phi_0(x)+a_1\phi_1(x)+\cdots+a_n\phi_n(x),$$

其中 $a_0, a_1, \ldots, a_n$ 是线性表出系数. 这样, 多项式插值就转化为下面两个问题

(1) 寻找合适的基函数;

(2) 确定插值多项式在这组基下的线性表出系数.

## 2.2 Lagrange 插值

将线性插值和抛物线插值的思想推广到一般情形, 就得到 Lagrange 插值法, 它是基函数插值法的典型代表.

### 2.2.1 Lagrange 基函数

**定义 2.3** 设 $l_k(x)$ 是 $n$ 次多项式, 且在插值节点 $x_0, x_1, \dots, x_n$ 上满足

$$l_k(x_i) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \quad i, k = 0, 1, 2, \dots, n. \tag{2.4}$$

则称 $l_k(x)$ 为节点 $x_0, x_1, \dots, x_n$ 上的 $n$ 次 Lagrange 基函数.

下面利用构造法计算 $l_k(x)$ 的表达式. 由条件 (2.4) 可知 $x_0, \dots, x_{k-1}, x_{k+1}, \dots, x_n$ 是 $l_k(x)$ 的零点, 又 $l_k(x)$ 是 $n$ 次多项式, 故可设

$$l_k(x) = \alpha(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n),$$

其中 $\alpha$ 是待定系数. 将条件 $l_k(x_k) = 1$ 代入可得

$$\alpha = \frac{1}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}.$$

所以

$$\begin{aligned} l_k(x) &= \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)} \\ &= \prod_{i=0,\ i \neq k}^{n} \frac{x - x_i}{x_k - x_i}. \end{aligned}$$

- 容易证明: $l_0(x), l_1(x), \dots, l_n(x)$ 线性无关, 因此它们构成 $H_n(x)$ 的一组基.
- $l_0(x), l_1(x), \dots, l_n(x)$ 与插值节点有关，但与 $f(x)$ 无关.

### 2.2.2 如何用 Lagrange 基函数求插值多项式

由于 $l_0(x), l_1(x), \dots, l_n(x)$ 构成 $H_n(x)$ 的一组基, 所以插值多项式 $p(x)$ 可以写成

$$p(x) = a_0 l_0(x) + a_1 l_1(x) + \cdots + a_n l_n(x).$$

将插值条件 (2.2) 代入可得

$$a_i = y_i, \quad i = 0, 1, 2, \dots, n.$$

所以

$$p(x) = y_0 l_0(x) + y_1 l_1(x) + \cdots + y_n l_n(x).$$

我们将这个多项式记为 $L_n(x)$, 它就是 Lagrange 插值多项式, 即

$$L_n(x)=\sum_{k=0}^{n}y_k l_k(x)=\sum_{k=0}^{n}y_k\prod_{i=0,\ i\neq k}^{n}\frac{x-x_i}{x_k-x_i}. \tag{2.5}$$

当 $n=1$ 和 2 时, 就可以得到线性插值多项式和抛物线插值多项式.

♣ $L_n(x)$ 通常是 $n$ 次的, 但也可能会低于 $n$ 次. 如: 抛物线插值中, 如果三点共线, 则 $L_2(x)$ 是一次多项式.

**例 2.4** 已知函数 $f(x)=\ln(x)$ 的函数值如下:

| $x$ | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 |
|---|---|---|---|---|---|
| $f(x)$ | -0.9163 | -0.6931 | -0.5108 | -0.3567 | -0.2231 |

试分别用线性插值和抛物线插值计算 $\ln(0.54)$ 的近似值.

**解.** 为了减小截断误差, 通常选取离插值点 $x$ 比较近的点作为插值节点.

线性插值: 取插值节点 $x_0=0.5$, $x_1=0.6$. 根据 Lagrange 插值公式, 可得 $f(x)$ 在区间 $[0.5,0.6]$ 上的线性插值为

$$L_1(x)=y_0\frac{x-x_1}{x_0-x_1}+y_1\frac{x-x_0}{x_1-x_0}\approx 0.1823x-1.6046.$$

将 $x=0.54$ 代入可得 $\ln(0.54)\approx -0.6202$.

抛物线插值: 取插值节点 $x_0=0.4$, $x_1=0.5$, $x_2=0.6$. 根据 Lagrange 插值公式, 可得 $f(x)$ 在 $x=0.54$ 上的近似值为

$$\begin{aligned}\ln(0.54)&\approx L_2(0.54)\\&=y_0\frac{(0.54-x_1)(0.54-x_2)}{(x_0-x_1)(x_0-x_2)}+y_1\frac{(0.54-x_0)(0.54-x_2)}{(x_1-x_0)(x_1-x_2)}+y_2\frac{(0.54-x_0)(0.54-x_1)}{(x_2-x_0)(x_2-x_1)}\\&\approx -0.6153.\end{aligned}$$

□

♣ 我们将需要插值的点称为 插值点, 比如上面例题中的 0.54. 在实际计算中, 一般不需要给出插值多项式的具体表达式, 可以直接将插值点代入进行计算, 得到近似值.

♣ Lagrange 插值简单方便, 只要给定插值节点就可写出基函数, 易于计算机实现.

### 2.2.3 插值余项

Lagrange 插值多项式的余项记为

$$R_n(x)\triangleq f(x)-L_n(x).$$

**定理 2.2** 设 $f(x)\in C^n[a,b]$ (即存在 $n$ 阶连续导数), 且 $f^{(n+1)}(x)$ 在 $(a,b)$ 内存在. 则对 $\forall\, x\in[a,b]$, 都存在 $\xi_x\in(a,b)$ 使得

$$R_n(x)=\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\,\omega_{n+1}(x), \tag{2.6}$$

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

证明. 只需证明等式 (2.6) 对所有 $x\in[a,b]$ 都成立即可.

首先证明等式 (2.6) 在所有插值节点 $x_i$ 上都成立. 易知等式左边 $R_n(x_i)=f(x_i)-L_n(x_i)=0$. 又由 $\omega_{n+1}(x)$ 的表达式可知 $\omega(x_i)=0$. 因此等式 (2.6) 成立.

下面证明等式 (2.6) 对其他点也成立. 由 $R_n(x_i)=f(x_i)-L_n(x_i)=0$ 可知 $x_0,x_1,x_2,\ldots,x_n$ 是 $R_n(x)$ 的零点, 因此 $R_n(x)$ 可以写成

$$R_n(x)=K(x)(x-x_0)(x-x_1)\cdots(x-x_n)=K(x)\omega_{n+1}(x), \tag{2.7}$$

其中 $K(x)$ 待定.

设 $\tilde{x}$ 是 $[a,b]$ 中的任意一点, 且 $\tilde{x}\neq x_i, i=0,1,2,\ldots,n$. 下面证明等式 (2.6) 在点 $\tilde{x}$ 上也成立, 即证明: **存在 $\xi_{\tilde{x}}$ 使得 $K(\tilde{x})=\dfrac{f^{(n+1)}(\xi_{\tilde{x}})}{(n+1)!}$.**

构造函数

$$\varphi(x)\triangleq f(x)-L_n(x)-K(\tilde{x})\omega_{n+1}(x). \tag{2.8}$$

由 (2.7) 可知,

$$\varphi(\tilde{x})=f(\tilde{x})-L_n(\tilde{x})-K(\tilde{x})\omega_{n+1}(\tilde{x})=R_n(\tilde{x})-K(\tilde{x})\omega_{n+1}(\tilde{x})=0.$$

又

$$\varphi(x_i)=f(x_i)-L_n(x_i)-K(\tilde{x})\omega_{n+1}(x_i)=0-0=0,\quad i=0,1,2,\ldots,n,$$

所以 $\varphi(x)$ 在 $[a,b]$ 内至少有 $n+2$ 个互不相同的零点. 根据条件, $\varphi(x)$ 是 $n$ 阶连续可导且存在 $n+1$ 阶导数, 因此由罗尔定理可知, $\varphi'(x)$ 在 $(a,b)$ 内至少有 $n+1$ 个不同的零点. 再根据罗尔定理, $\varphi''(x)$ 在 $(a,b)$ 内至少有 $n$ 个不同的零点. 以此类推, 可知 $\varphi^{(n+1)}(x)$ 在 $(a,b)$ 内至少有 1 个零点, 不妨设为 $\xi_{\tilde{x}}$, 即 $\varphi^{(n+1)}(\xi_{\tilde{x}})=0$. 代入 (2.8) 可得

$$f^{(n+1)}(\xi_{\tilde{x}})-L_n^{(n+1)}(\xi_{\tilde{x}})-K(\tilde{x})\omega_{n+1}^{(n+1)}(\xi_{\tilde{x}})=0.$$

又 $L_n^{(n+1)}(x)=0$, $\omega_{n+1}^{(n+1)}(x)=(n+1)!$, 因此

$$K(\tilde{x})=\frac{f^{(n+1)}(\xi_{\tilde{x}})}{(n+1)!}.$$

由此可知, 等式 (2.6) 对 $[a,b]$ 内的所有点都成立. □

♣ 余项中的 $\xi_x$ 与 $x$ 是相关的.

特别地, 当 $n=1$ 时, 线性插值的余项是 (假定 $x_0<x_1$)

$$R_1(x)=\frac{1}{2}f''(\xi_x)(x-x_0)(x-x_1),\quad \xi_x\in(x_0,x_1).$$

当 $n=2$ 时, 抛物线插值的余项是 (假定 $x_0<x_1<x_2$)

$$R_2(x)=\frac{1}{6}f'''(\xi_x)(x-x_0)(x-x_1)(x-x_2),\quad \xi_x\in(x_0,x_2).$$

- 余项公式只有当 $f(x)$ 的高阶导数存在时才能使用;
- $\xi_x$ 与 $x$ 有关, 通常无法确定, 因此在实际应用中, 通常是估计其上界, 即

$$\text{如果有 } \max_{a\le x\le b}|f^{(n+1)}(x)|=M_{n+1},\quad \text{则 } |R_n(x)|\le\frac{M_{n+1}}{(n+1)!}|\omega_{n+1}(x)|.$$

- 在利用插值方法计算插值点 $x$ 上的近似值时, 应尽量选取与 $x$ 相近的插值节点.

### 2.2.4 Lagrange 基函数的两个重要性质

如果 $f(x)$ 是一个次数不超过 $n$ 的多项式, 则 $f^{(n+1)}(x)\equiv 0$, 因此

$$R_n(x)=\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x)\equiv 0.$$

所以我们有下面的性质.

**性质 2.1** 当 $f(x)$ 为一个次数不超过 $n$ 的多项式时, 其 $n$ 次插值多项式是精确的.

设 $f(x)=x^m$, 其中 $0\le m\le n$, 则由性质 2.1 可知 $f(x)=L_n(x)$, 即

$$\sum_{k=0}^{n}x_k^m l_k(x)=x^m,\quad 0\le m\le n. \tag{2.9}$$

特别地, 当 $m=0$ 时, 有

$$\sum_{k=0}^{n}l_k(x)=1.$$

这是一个很有趣的性质.

**例 2.5** 设 $l_i(x)$ 是关于点 $x_0,x_1,\dots,x_5$ 的 Lagrange 插值基函数. 证明:

$$\sum_{i=0}^{5}(x_i-x)^2 l_i(x)=0.$$

**证明.** 直接展开可得

$$\begin{aligned}\sum_{i=0}^{5}(x_i-x)^2 l_i(x)&=\sum_{i=0}^{5}(x_i^2-2x_ix+x^2)l_i(x)\\&=\sum_{i=0}^{5}x_i^2 l_i(x)-2x\sum_{i=0}^{5}x_i l_i(x)+x^2\sum_{i=0}^{5}l_i(x)\end{aligned} \tag{2.10}$$

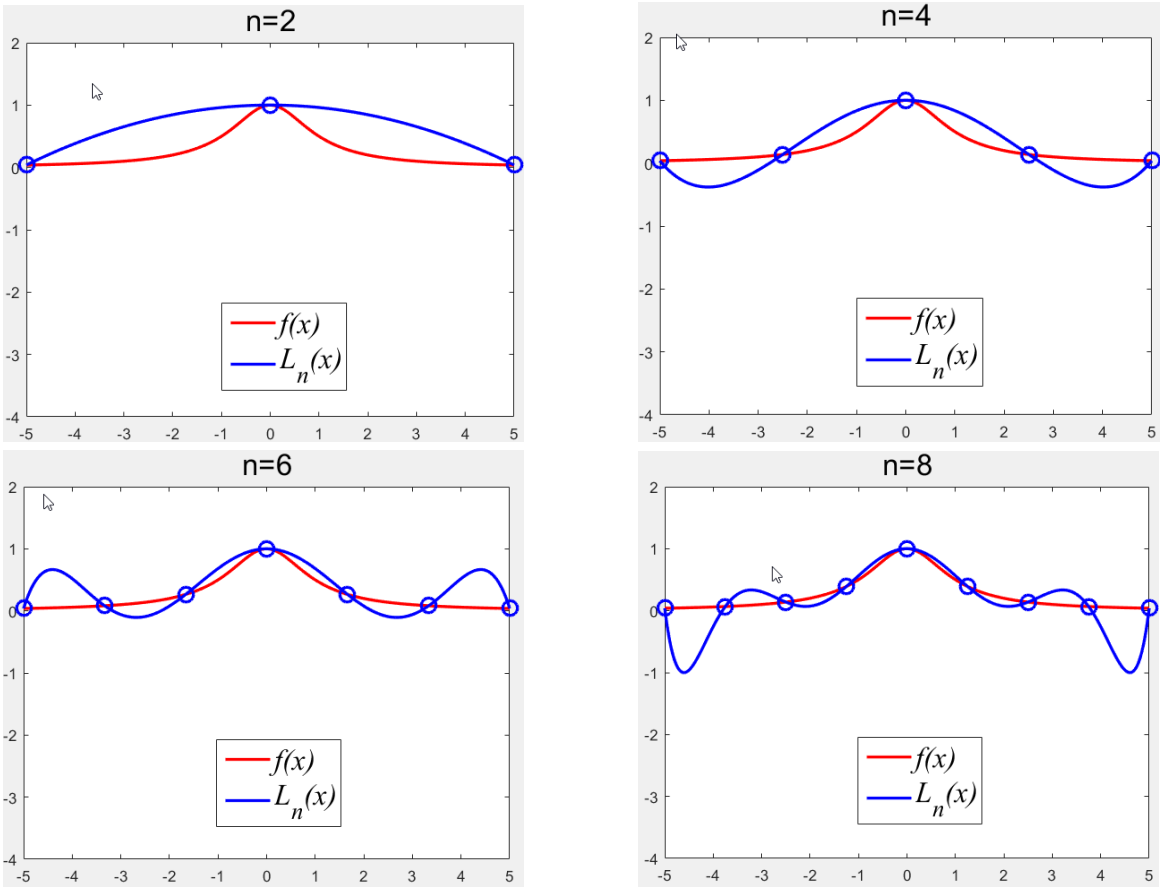
根据等式 (2.9) 可知

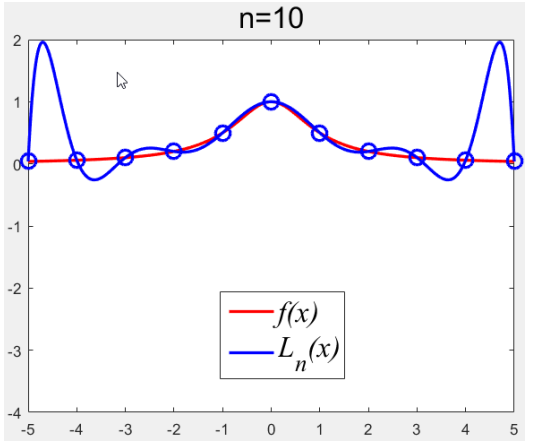
$$\sum_{i=0}^{5} x_i^2 l_i(x) = x^2, \quad \sum_{i=0}^{5} x_i l_i(x) = x, \quad \sum_{i=0}^{5} l_i(x) = 1,$$

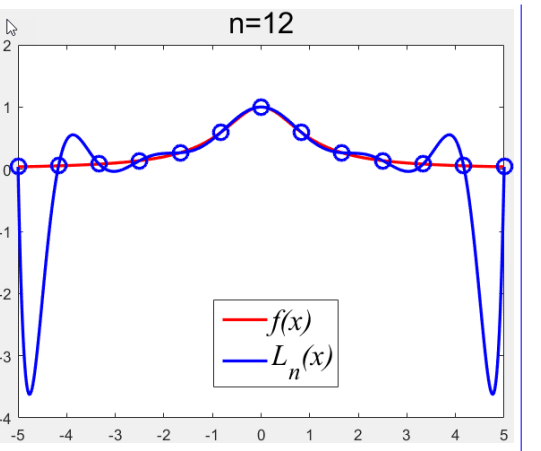
代入 (2.10), 即可知结论成立. □

**例 2.6 (编程)** Runge 现象: 已知函数 $f(x) = \dfrac{1}{1+x^2}$, 插值区间为 $[-5,5]$, 取等距插值节点, 即 $x_i = -5 + \dfrac{10i}{n}$, $i = 0,1,2,\ldots,n$, 画出当 $n = 2,4,6,8,10,12$ 时, 插值多项式 $L_n$ 的图像. 并由此说明, 插值多项式并不一定是次数越高就越好!

**解.** 当 $n = 2,4,6,8,10$ 时, 原函数 $f(x)$ 和插值多项式 $L_n$ 的图像如下:






MATLAB 源代码 2.1. Runge 现象

```
clear; clc;
f=@(x) 1./(1+x.^2);
a=-5; b=5; % 插值区间
xh=-5:0.1:5; % 绘图点
yt=f(xh);   % 函数精确值
for n=2:2:14
    X=a:(b-a)/n:b; % 插值节点
    Y=f(X);      % 函数在插值节点上的值
    % 下面利用插值多项式计算函数在绘图点上的近似值
    yh=zeros(1,length(xh));
    for i=1:length(xh)
        for k=0:n
            x1=xh(i)-X([1:k,k+2:end]); % 分子
            x2=X(k+1)-X([1:k,k+2:end]); % 分母
            yh(i)=yh(i)+Y(k+1)*prod(x1./x2);
        end
    end
   % 绘图
    plot(xh,yt,'r+-', xh,yh,'bo-','LineWidth',1);
    axis([-5,5,-4,2]);    % 控制坐标显示范围
    tit = ['n=',int2str(n)]; % 设置标题
    title(tit,'FontSize',20);
    legend('f(x)','L_n(x)'); % 添加图例
    pause
end
```

**例 2.7** 设 $f(x)\in C^2[a,b]$ (二阶连续可导), 证明:

$$\max_{a\le x\le b}\left|f(x)-\left[f(a)+\frac{f(b)-f(a)}{b-a}(x-a)\right]\right|\le\frac{1}{8}M_2(b-a)^2,$$

其中 $M_2=\max\limits_{a\le x\le b}|f''(x)|$.

**证明.** 记

$$L_1(x)=f(a)+\frac{f(b)-f(a)}{b-a}(x-a),$$

则

$$L_1(a)=f(a),\quad L_1(b)=f(b).$$

因此, $L_1(x)$ 是 $f(x)$ 关于插值节点 $x_0=a, x_1=b$ 的线性插值多项式. 由多项式插值的余项公式 (2.6) 可知

$$R_1(x)=f(x)-L_1(x)=\frac{f''(\xi_x)}{2!}(x-a)(x-b),\quad \xi_x\in(a,b).$$

因此当 $x\in[a,b]$ 时, 有

$$\begin{aligned}|f(x)-L_1(x)|&=\frac{|f''(\xi_x)|}{2}|(x-a)(x-b)|\\&\le\frac{M_2}{2}(x-a)(b-x)\\&\le\frac{M_2}{2}\left(\frac{(x-a)+(b-x)}{2}\right)^2=\frac{1}{8}M_2(b-a)^2.\end{aligned}$$

所以结论成立. □

## 2.3 Newton 插值

### 2.3.1 为什么 Newton 插值

Lagrange 插值简单易用, 但若增加插值节点时, 全部基函数 $l_k(x)$ 都需重新计算, 很不方便!

解决办法就是寻找新的基函数组, 使得当节点增加时, 只需在原有基函数的基础上再增加一些新的基函数即可. 这样, 原有的基函数仍然可以使用.

基于这种基函数的选取方法, 我们还可能设计一个可以逐次生成插值多项式的算法, 即

$$p_{n+1}(x)=p_n(x)+u_{n+1}(x),$$

其中 $p_{n+1}(x)$ 和 $p_n(x)$ 分别为 $f(x)$ 的 $n+1$ 次和 $n$ 次插值多项式, 而且 $u_{n+1}(x)$ 可以很容易地给出.

### 2.3.2 Newton 插值基函数

设插值节点为 $x_0,x_1,\dots,x_n$, 考虑函数组

$$\begin{aligned}\phi_0(x)&=1\\\phi_1(x)&=x-x_0\\\phi_2(x)&=(x-x_0)(x-x_1)\\&\cdots\cdots\end{aligned}$$

$$\phi_n(x) = (x - x_0)(x - x_1)\cdots(x - x_{n-1}).$$

显然, $\phi_k(x)$ 是 $k$ 次多项式, 且 $\phi_0(x), \phi_1(x), \ldots, \phi_n(x)$ 线性无关, 因此它们组成多项式线性空间 $H_n(x)$ 的一组基.

这组基函数的优点是当增加一个新的插值节点 $x_{n+1}$ 时, 只需在原有的基的基础上增加下面的函数即可

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

这意味着原来的基函数仍然可以使用. Newton 插值法就是基于这组基函数组的插值方法.

设 $p_n(x)$ 是 $f(x)$ 在节点 $x_0, x_1, \ldots, x_n$ 上的 $n$ 次插值多项式. 由于 $\phi_0(x)$, $\phi_1(x), \ldots, \phi_n(x)$ 构成 $H_n(x)$ 上的一组基, 因此 $p_n(x)$ 可以写成

$$p_n(x) = a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_n\phi_n(x).$$

下面我们需要解决两个问题

(1) 怎样确定参数 $a_0, a_1, \ldots, a_n$?

(2) 如果得到从 $p_n(x)$ 到 $p_{n+1}(x)$ 的递推方法?

要解决以上问题, 我们需要用到差商.

### 2.3.3 差商及其计算

**定义 2.4** 设节点 $x_0, x_1, \ldots, x_n$, 我们称

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

为 $f(x)$ 关于点 $x_i, x_j$ 的一阶差商; 称

$$f[x_i, x_j, x_k] = \frac{f[x_j, x_k] - f[x_i, x_j]}{x_k - x_i}$$

为 $f(x)$ 关于点 $x_i, x_j, x_k$ 的二阶差商; 一般地, 称

$$f[x_0, x_1, \ldots, x_k] = \frac{f[x_1, x_2, \ldots, x_k] - f[x_0, x_1, \ldots, x_{k-1}]}{x_k - x_0}$$

为 $f(x)$ 关于点 $x_0, x_1, \ldots, x_k$ 的 $k$ 阶差商.

差商有时也称为均差. 下面是关于差商的几个基本性质.

- 差商可以表示为函数值的线性组合, 即 (可以用归纳法证明)

$$f[x_0, x_1, \ldots, x_k] = \sum_{j=0}^{k} \frac{f(x_j)}{\prod\limits_{i=0,\, i\neq j}^{k} (x_j - x_i)} = \sum_{j=0}^{k} \frac{f(x_j)}{\omega'_{k+1}(x_j)}, \tag{2.11}$$

其中 $\omega_{k+1}(x) = (x - x_0)(x - x_1)\cdots(x - x_k)$. 由此可见, 差商与节点的排序无关, 即差商具有下面的对称性:

$$f[x_0, x_1, \ldots, x_k] = f[x_{i_0}, x_{i_1}, \ldots, x_{i_k}]$$

其中 $i_0, i_1, \ldots, i_k$ 是 $0, 1, \ldots, k$ 的一个任意排列.

- 根据差商的对称性, 我们可以给出差商的等价定义, 如:
$$f[x_0, x_1, \ldots, x_k] = \frac{f[x_0, \ldots, x_{k-2}, x_k] - f[x_0, \ldots, x_{k-2}, x_{k-1}]}{x_k - x_{k-1}}.$$
- 若 $h(x) = \alpha f(x)$, 其中 $\alpha$ 是常数, 则
$$h[x_0, x_1, \ldots, x_k] = \alpha f[x_0, x_1, \ldots, x_k].$$
- 若 $h(x) = f(x) + g(x)$, 则
$$h[x_0, x_1, \ldots, x_k] = f[x_0, x_1, \ldots, x_k] + g[x_0, x_1, \ldots, x_k].$$
- $k$ 阶差商与 $k$ 阶导数之间的关系: 若 $f(x)$ 在 $[a, b]$ 上有 $k$ 阶导数, 则至少存在一点 $\xi \in (a, b)$, 使得
$$f[x_0, x_1, \ldots, x_k] = \frac{f^{(k)}(\xi)}{k!}. \tag{2.12}$$

**差商的计算**

利用差商的递推定义, 我们可以构造下面的差商表来计算差商.

| $x_i$ | $f(x_i)$ | 一阶差商 | 二阶差商 | 三阶差商 | $\cdots$ | $n$ 阶差商 |
|---|---|---|---|---|---|---|
| $x_0$ | $f(x_0)$ | | | | | |
| $x_1$ | $f(x_1)$ | $f[x_0, x_1]$ | | | | |
| $x_2$ | $f(x_2)$ | $f[x_1, x_2]$ | $f[x_0, x_1, x_2]$ | | | |
| $x_3$ | $f(x_3)$ | $f[x_2, x_3]$ | $f[x_1, x_2, x_3]$ | $f[x_0, x_1, x_2, x_3]$ | | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | |
| $x_n$ | $f(x_n)$ | $f[x_{n-1}, x_n]$ | $f[x_{n-2}, x_{n-1}, x_n]$ | $f[x_{n-3}, x_{n-2}, x_{n-1}, x_n]$ | $\cdots$ | $f[x_0, x_1, \ldots, x_n]$ |

MATLAB 源代码 2.2. 计算差商

```
function [p,q] = mycs(x,y)
%
% 输入参数:
%   x 是向量, 包含所有插值节点
%   y 是向量, 在插值节点的函数值, 与 x 的长度必须一致
% 输出参数:
%   p 是矩阵, 包含差商表中的所有值
%   q 是向量, 为差商表中对角线的值, 供 Newton 插值用

m=length(x);
x=x(:);
p=zeros(m,m+1);
p(:,1)= x;
p(:,2)=y(:);
for k=3:m+1
   p(k-1:m,k)=diff(p(k-2:m,k-1)) ./ (x(k-1:m)-x(1:m+2-k));
end
q=diag(p(1:m,2:m+1));
```

**例 2.8** 已知 $y = f(x)$ 的函数值表如下, 试计算其各阶差商

| $i$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $x_i$ | -2 | -1 | 1 | 2 |
| $f(x_i)$ | 5 | 3 | 17 | 21 |

**解.** (MATLAB 程序见 `ex23.m, ex24.m`) 差商表如下

| $x_i$ | $f(x_i)$ | 一阶差商 | 二阶差商 | 三阶差商 |
|---|---|---|---|---|
| -2 | 5 | | | |
| -1 | 3 | -2 | | |
| 1 | 17 | 7 | 3 | |
| 2 | 21 | 4 | -1 | -1 |

□

### 2.3.4 Newton 插值公式

下面我们开始推导 Newton 插值公式. 由差商的定义可知

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

所以

$$f(x) = f(x_0) + f[x, x_0](x - x_0). \tag{2.13}$$

同理, 由

$$f[x, x_0, x_1] = \frac{f[x, x_0] - f[x_0, x_1]}{x - x_1}$$

可得

$$f[x, x_0] = f[x_0, x_1] + f[x, x_0, x_1](x - x_1). \tag{2.14}$$

以此类推, 我们有

$$f[x, x_0, x_1] = f[x_0, x_1, x_2] + f[x, x_0, x_1, x_2](x - x_2) \tag{2.15}$$

$$\vdots$$

$$f[x, x_0, \ldots, x_{n-1}] = f[x_0, x_1, \ldots, x_n] + f[x, x_0, x_1, \ldots, x_n](x - x_n). \tag{2.16}$$

将等式 (2.13)-(2.16) 联立可得 (依次将后面一式代入前面一式)

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

所以

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

其中

$$N_n(x) = a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0)\cdots(x - x_{n-1}), \tag{2.17}$$

$$a_0 = f(x_0), \quad a_i = f[x_0, x_1, \ldots, x_i], \quad i = 1, 2, \ldots, n.$$

$$R_n(x) = f[x, x_0, \ldots, x_n](x - x_0)\cdots(x - x_{n-1})(x - x_n).$$

我们注意到, $N_n(x)$ 是一个 $n$ 次多项式. 而且通过直接验证可知

$$R_n(x_i) = 0, \quad i = 0, 1, 2, \ldots, n.$$

所以

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

即 $N_n(x)$ 是满足插值条件 (2.2) 的 $n$ 次插值多项式. 我们称之为 Newton 插值多项式.

由 $N_n(x)$ 的表达式, 我们可以立即得到下面的递推公式:

$$N_{n+1}(x) = N_n(x) + f[x_0, x_1, \ldots, x_{n+1}]\prod_{i=0}^{n}(x - x_i).$$

由插值多项式的存在唯一性可知, Newton 插值多项式与 Lagrange 插值多项式是一样的, 即

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

所以, 它们的插值余项也一样, 即

$$f[x, x_0, \ldots, x_n]\prod_{i=0}^{n}(x - x_i) \equiv \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\prod_{i=0}^{n}(x - x_i).$$

由此, 我们立即可以得到下面的结论.

**性质 2.2** 设 $f(x) \in C^n[a, b]$ ($n$ 阶连续可导), 且 $f^{(n+1)}(x)$ 在 $(a, b)$ 内存在. 则对 $\forall x \in [a, b]$, 存在 $\xi_x \in (a, b)$ 使得

$$f[x, x_0, \ldots, x_n] = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}.$$

这就是我们前面提到的差商与导数之间的关系 (2.12), 即

$$f[x_0, x_1, \ldots, x_k] = \frac{f^{(k)}(\xi)}{k!}.$$

♣ Newton 插值余项更具实用性, 因为它仅涉及插值点与插值节点的差商, 而不需要计算导数, 因此在导数不存在的情况下仍然可以使用. 但在计算差商 $f[x, x_0, \ldots, x_n]$ 时, 由于 $f(x)$ 未知, 只能使用插值得到的近似值, 因此得到的差商可能具有一定的偏差.

**例 2.9** 已知函数 $f(x) = \ln(x)$ 的函数值如下:

| $x$ | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 |
|---|---|---|---|---|---|
| $f(x)$ | -0.9163 | -0.6931 | -0.5108 | -0.3567 | -0.2231 |

试分别用 Newton 线性插值和抛物线插值计算 $\ln(0.54)$ 的近似值.

**解.** (MATLAB 程序见 `ex25.m`) 取插值节点 $x_0 = 0.5, x_1 = 0.6, x_2 = 0.4$, 做差商表

| $x_i$ | $f(x_i)$ | 一阶差商 | 二阶差商 |
|---|---|---|---|
| 0.5 | -0.6931 | | |
| 0.6 | -0.5108 | 1.8230 | |
| 0.4 | -0.9163 | 2.0275 | -2.0450 |

于是可得, Newton 线性插值在 $x = 0.54$ 上的近似值为

$$N_1(x) = f(x_0) + f[x_0, x_1](x - x_0) \approx -0.6931 + 1.8230(x - 0.5) \approx -0.6202.$$

Newton 抛物线插值在 $x = 0.54$ 上的近似值为

$$N_2(x) = N_1(x) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \approx -0.6153.$$

□

思考: 这里插值节点顺序为什么这么取?

♣ 在 Newton 插值法中, 我们只需要使用差商表中对角线部分的值.

可以看出, 当增加一个节点时, 牛顿插值公式只需在原来的基础上增加一项, 前面的计算结果仍然可以使用. 与拉格朗日插值相比, 牛顿插值具有灵活增加插值节点的优点!

♣ 增加插值节点时, 新增的插值点必须加在已有插值节点的后面.

### 2.3.5 差分

在实际应用中, 为了计算方便, 我们通常采用等距节点, 即:

$$x_i = x_0 + i \times h, \quad i = 0, 1, 2, \ldots, n, \tag{2.18}$$

这里 $h > 0$ 为**步长**. 此时, 我们可以用差分简化 Newton 插值公式.

**定义 2.5 (向前差分)** 设 $\{x_i\}$ 是由 (2.18) 定义的等距节点, 则函数 $f(x)$ 在 $x_i$ 处以 $h$ 为步长的**一阶 (向前) 差分**定义为

$$\Delta f_i = f(x_{i+1}) - f(x_i) = f(x_i + h) - f(x_i).$$

类似地, 称

$$\Delta^2 f_i = \Delta(\Delta f_i) = \Delta f_{i+1} - \Delta f_i$$

为 $x_i$ 处的二阶差分. 一般地, 称

$$\Delta^n f_i = \Delta(\Delta^{n-1} f_i) = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i$$

为 $x_i$ 处的 $n$ 阶差分.

为了表示方便, 我们引入两个算子:

$$\mathbf{I} f_i = f_i, \quad \mathbf{E} f_i = f_{i+1},$$

分别称为不变算子和位移算子. 于是

$$\Delta f_i = f_{i+1} - f_i = \mathbf{E} f_i - \mathbf{I} f_i = (\mathbf{E} - \mathbf{I}) f_i.$$

所以我可以得到下面的差分与函数值之间的关系:

$$\begin{aligned} \Delta^n f_i = (\mathbf{E} - \mathbf{I})^n f_i &= \left[ \sum_{k=0}^{n} (-1)^k \binom{n}{k} \mathbf{E}^{n-k} \right] f_i \\ &= \sum_{k=0}^{n} (-1)^k \frac{n(n-1)\cdots(n-k+1)}{k!} f_{n-k+i}. \end{aligned}$$

反之, 有

$$f_{n+i} = \mathbf{E}^n f_i = (\mathbf{I} + \Delta)^n f_i = \sum_{k=0}^{n} \binom{n}{k} \Delta^k f_i.$$

**差分与差商之间的关系**

由差商定义可知

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{1}{h} \Delta f_0.$$

事实上, 对于任意两个相邻的节点, 都有上面的等式, 即

$$f[x_k, x_{k+1}] = \frac{f_{k+1} - f_k}{x_{k+1} - x_k} = \frac{1}{h} \Delta f_k.$$

对于任意三个相邻的节点, 有

$$\begin{aligned} f[x_k, x_{k+1}, x_{k+2}] &= \frac{f[x_{k+2}, x_{k+1}] - f[x_{k+1}, x_k]}{x_{k+2} - x_k} \\ &= \frac{1}{2h} \left( \frac{1}{h} \Delta f_{k+1} - \frac{1}{h} \Delta f_k \right) \\ &= \frac{1}{2} \frac{1}{h^2} \Delta^2 f_k. \end{aligned}$$

一般地, 对于任意 $m+1$ 个相邻的节点, 有

$$f[x_k, x_{k+1}, \ldots, x_{k+m}] = \frac{1}{m!} \frac{1}{h^m} \Delta^m f_k.$$

所以由差商与导数之间的关系 (2.12) 可知

$$\Delta^m f_k = h^m m! \times f[x_k, x_{k+1}, \ldots, x_{k+m}] = h^m f^{(m)}(\xi), \quad \xi \in (x_k, x_{k+m}).$$

**差分的计算**

与差商表类似, 我们也可以通过下面的差分表来计算差分.

| $x_i$ | $f(x_i)$ | 一阶差分 | 二阶差分 | 三阶差分 | $\cdots$ | $n$阶差分 |
|---|---|---|---|---|---|---|
| $x_0$ | $f(x_0)$ | $\Delta f_0$ | $\Delta^2 f_0$ | $\Delta^3 f_0$ | $\cdots$ | $\Delta^n f_0$ |
| $x_1$ | $f(x_1)$ | $\Delta f_1$ | $\Delta^2 f_1$ | $\Delta^3 f_1$ | | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | |
| $x_{n-2}$ | $f(x_{n-2})$ | $\Delta f_{n-2}$ | $\Delta^2 f_{n-2}$ | | | |
| $x_{n-1}$ | $f(x_{n-1})$ | $\Delta f_{n-1}$ | | | | |
| $x_n$ | $f(x_n)$ | | | | | |

♣ MATLAB 中计算差分的函数为: `diff(x)`, 可以计算高阶差分, 如: `diff(x,2)`.

### 2.3.6 Newton 向前插值公式

如果采用等距插值节点 $x_i = x_0 + i\,h$, 其中 $h > 0$ 为步长, 则我们可以用差分来简化 Newton 插值公式. 设 $x = x_0 + th$, 则

$$\begin{aligned} N_n(x) &= N_n(x_0 + th) \\ &= f(x_0) + t\Delta f_0 + \frac{t(t-1)}{2!}\Delta^2 f_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 f_0 + \cdots \\ &\quad + \frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n f_0. \end{aligned} \tag{2.19}$$

这就是Newton 向前插值公式. 插值余项为

$$R_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}t(t-1)\cdots(t-n)h^{n+1}.$$

**例 2.10** 给定 $f(x) = \cos(x)$ 在等距节点 $0 : 0.1 : 0.5$ 处的函数值, 试用 4 次 Newton 向前插值公式计算 $f(0.048)$ 的近似值, 并估计误差.

**解.** 取等距节点 $x = 0 : 0.1 : 0.4$, 做差分表

| $x_i$ | $f(x_i)$ | 一阶差分 | 二阶差分 | 三阶差分 | 四阶差分 |
|---|---|---|---|---|---|
| 0.0 | 1.00000 | −0.00500 | −0.00993 | −0.00013 | −0.00012 |
| 0.1 | 0.99500 | −0.01493 | −0.00980 | −0.00025 | |
| 0.2 | 0.98007 | −0.02473 | −0.00955 | | |
| 0.3 | 0.95534 | −0.03428 | | | |
| 0.4 | 0.92106 | | | | |

插值点 $x = 0.048$, 则 $t = (x - x_0)/h = 0.48$. 所以由 Newton 向前插值公式 (2.19) 可知, $f(0.048)$ 的近似值为

$$\begin{aligned} N_4(0.048) &= 1.00000 + 0.48 \times (-0.00500) \\ &\quad + \frac{1}{2!} \times 0.48(0.48-1)(-0.00993) \end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{3!}\times 0.48(0.48-1)(0.48-2)(-0.00013)\\
&+\frac{1}{4!}\times 0.48(0.48-1)(0.48-2)(0.48-3)(-0.00012)\\
&\approx 0.99884.
\end{aligned}
$$

余项为

$$
\begin{aligned}
|R_4(0.048)| &= \left|\frac{f^{(5)}(\xi)}{5!}t(t-1)(t-2)(t-3)(t-4)h^5\right|\\
&\le \frac{h^5}{5!}\cdot|t(t-1)(t-2)(t-3)(t-4)|\cdot\max_{0\le x\le 0.4}|f^{(5)}(x)|\\
&\approx 1.09212\times 10^{-7}.
\end{aligned}
$$

% 四次 Newton 向前插值公式只需用到前面 5 个等距插值节点. □

♣ Newton 向前插值公式只需用到差分表的第一行.

### 2.3.7 向后差分与中心差分

与向前差分类似, 我们还可以定义向后差分:

$$
\begin{aligned}
&\nabla f_i = f(x_i)-f(x_{i-1}),\\
&\nabla^k f_i = \nabla(\nabla^{k-1}f_i) = \nabla^{k-1}f_i - \nabla^{k-1}f_{i-1},\quad k=2,3,\dots
\end{aligned}
$$

和中心差分:

$$
\begin{aligned}
&\delta f_i = f(x_{i+\frac{1}{2}})-f(x_{i-\frac{1}{2}}),\\
&\delta^k f_i = \delta(\delta^{k-1}f_i) = \delta^{k-1}f_{i+\frac{1}{2}} - \delta^{k-1}f_{i-\frac{1}{2}},\quad k=2,3,\dots.
\end{aligned}
$$

## 2.4 Hermite 插值

### 2.4.1 为什么 Hermite 插值

在许多实际应用中, 不仅要求函数值相等, 而且还要求若干阶导数也相等, 如机翼设计等.

设插值节点为 $x_0, x_1, \dots, x_n$, 如果要求插值多项式满足

$$
p(x_i)=f(x_i),\ p'(x_i)=f'(x_i),\ p''(x_i)=f''(x_i),\ \cdots,p^{(m)}(x_i)=f^{(m)}(x_i).
$$

则计算这类插值多项式的方法就称为Hermite 插值.

♣ 在 Hermite 插值中, 并不一定需要在所有插值节点上的导数都相等, 在有些情况下, 可能只需要在部分插值点上的导数值相等即可.

### 2.4.2 重节点差商

首先介绍差商的一个重要性质.

**定理 2.3** 设 $x_0, x_1, \dots, x_n$ 为 $[a,b]$ 上的互异节点, $f(x) \in C^n[a,b]$, 则 $f[x_0, x_1, \dots, x_n]$ 是其变量的连续函数.

例如, $f[x,y] = \dfrac{f(y)-f(x)}{y-x}$ 关于 $x$ 和 $y$ 都连续, 且当 $y \to x$ 时有

$$f[x,x] \triangleq \lim_{y\to x} f[x,y] = f'(x).$$

这就是 $f$ 在 $x$ 点的一阶重节点差商.

一般地, $f$ 在点 $x$ 的 $n$ 阶重节点差商定义为

$$f[\underbrace{x,x,\dots,x}_{n+1\text{ 个}}] \triangleq \lim_{x_i\to x} f[x,x_1,x_2,\dots,x_n] = \frac{1}{n!}f^{(n)}(x).$$

### 2.4.3 Taylor 插值

在 Newton 插值公式 (2.17) 中, 令 $x_i \to x_0$, $i = 1,2,\dots,n$, 则

$$\begin{aligned} N_n(x) &= f(x_0) + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) + \cdots \\ &\quad + f[x_0,x_1,\dots,x_n](x-x_0)\cdots(x-x_{n-1}) \\ &= f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2!}f''(x_0)(x-x_0)^2 + \cdots + \frac{1}{n!}f^{(n)}(x_0)(x-x_0)^n. \end{aligned}$$

这就是 Taylor 插值, 也即是 $f(x)$ 在 $x_0$ 点的 Taylor 展开式. 余项为

$$R_n(x) = \frac{1}{(n+1)!}f^{(n+1)}(\xi)(x-x_0)^{n+1}.$$

♣ Taylor 插值就是在一个插值点 $x_0$ 的 $n$ 次 Hermite 插值.

### 2.4.4 两个典型的 Hermite 插值

一般来说, 给定 $m+1$ 个插值条件, 就可以构造出一个 $m$ 次 Hermite 插值多项式. 这里介绍两个典型的 Hermite 插值: 三点三次 Hermite 插值 和 两点三次 Hermite 插值.

#### (1) 三点三次 Hermite 插值

设插值节点为 $x_0, x_1, x_2$, 则满足插值条件

$$p(x_0) = f(x_0), \quad p(x_1) = f(x_1), \quad p(x_2) = f(x_2), \quad p'(x_1) = f'(x_1),$$

的多项式 $p(x)$ 就称为三点三次 Hermite 插值多项式.

由于 $p(x_i) = f(x_i)$, 仿照 Newton 插值多项式, 我们可以设

$$\begin{aligned} p(x) &= f(x_0) + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) \\ &\quad + a(x-x_0)(x-x_1)(x-x_2). \end{aligned} \tag{2.20}$$

其中 $a$ 是待定系数. 将 $p'(x_1)=f'(x_1)$ 代入, 可得

$$a=\frac{f'(x_1)-f[x_0,x_1]-f[x_0,x_1,x_2](x_1-x_0)}{(x_1-x_0)(x_1-x_2)}.$$

根据插值条件, 我们可以将插值余项写成

$$R_3(x)=f(x)-p(x)=K(x)(x-x_0)(x-x_1)^2(x-x_2),$$

其中 $K(x)$ 待定. 与 Lagrange 插值余项公式的推导过程类似, 可得

$$R_3(x)=\frac{f^{(4)}(\xi_x)}{4!}(x-x_0)(x-x_1)^2(x-x_2),$$

其中 $\xi_x$ 位于由 $x_0,x_1,x_2$ 和 $x$ 所界定的区间内.

**例 2.11** 已知函数 $f(x)=x^{\frac{3}{2}}$, 插值条件为

$$f\left(\frac{1}{4}\right)=\frac{1}{8},\quad f(1)=1,\quad f\left(\frac{9}{4}\right)=\frac{27}{8},\quad f'(1)=\frac{3}{2},$$

是给出三次 Hermite 插值多项式, 并写出余项. (计算过程中不要做近似计算)

**解.** 做差商表

| $x_i$ | $f(x_i)$ | 一阶差商 | 二阶差商 |
|---|---|---|---|
| 1/4 | 1/8 | | |
| 1 | 1 | 7/6 | |
| 9/4 | 27/8 | 19/10 | 11/30 |

所以三次插值多项式可设为

$$p(x)=\frac{1}{8}+\frac{7}{6}\left(x-\frac{1}{4}\right)+\frac{11}{30}\left(x-\frac{1}{4}\right)(x-1)+\alpha\left(x-\frac{1}{4}\right)(x-1)\left(x-\frac{9}{4}\right).$$

将 $p'(1)=f'(1)=\frac{3}{2}$ 代入, 可得 $\alpha=-\frac{14}{225}$, 所以三次 Hermite 插值多项式为

$$\begin{aligned}p(x)&=\frac{1}{8}+\frac{7}{6}\left(x-\frac{1}{4}\right)+\frac{11}{30}\left(x-\frac{1}{4}\right)(x-1)-\frac{14}{225}\left(x-\frac{1}{4}\right)(x-1)\left(x-\frac{9}{4}\right)\\&=-\frac{14}{225}x^3+\frac{263}{450}x^2+\frac{233}{450}x-\frac{1}{25}.\end{aligned}$$

余项为

$$\begin{aligned}R(x)=f(x)-p(x)&=\frac{f^{(4)}(\xi_x)}{4!}\left(x-\frac{1}{4}\right)(x-1)^2\left(x-\frac{9}{4}\right)\\&=\frac{9\xi_x^{-5/2}}{16\times 4!}\left(x-\frac{1}{4}\right)(x-1)^2\left(x-\frac{9}{4}\right).\end{aligned}$$

□

**(2) 两点三次 Hermite 插值**

设插值节点为 $x_0,x_1$, 则满足插值条件

$$p(x_0)=f(x_0),\quad p(x_1)=f(x_1),\quad p'(x_0)=f'(x_0),\quad p'(x_1)=f'(x_1)$$

的多项式 $p(x)$ 就称为两点三次 Hermite 插值多项式, 记为 $H_3(x)$.

仿照 Lagrange 多项式的思想, 可设

$$H_3(x) = a_0\alpha_0(x) + a_1\alpha_1(x) + b_0\beta_0(x) + b_1\beta_1(x),$$

其中 $\alpha_0(x), \alpha_1(x), \beta_0(x), \beta_1(x)$ 均为三次多项式, 且满足

$$\begin{aligned}
&\alpha_0(x_0) = 1,\ \alpha_0(x_1) = 0,\ \alpha_0'(x_0) = 0,\ \alpha_0'(x_1) = 0;\\
&\alpha_1(x_0) = 0,\ \alpha_1(x_1) = 1,\ \alpha_1'(x_0) = 0,\ \alpha_1'(x_1) = 0;\\
&\beta_0(x_0) = 0,\ \beta_0(x_1) = 0,\ \beta_0'(x_0) = 1,\ \beta_0'(x_1) = 0;\\
&\beta_1(x_0) = 0,\ \beta_1(x_1) = 0,\ \beta_1'(x_0) = 0,\ \beta_1'(x_1) = 1.
\end{aligned}$$

根据插值条件可得

$$H_3(x) = f(x_0)\alpha_0(x) + f(x_1)\alpha_1(x) + f'(x_0)\beta_0(x) + f'(x_1)\beta_1(x).$$

剩下的问题就是如何确定 $\alpha_0(x), \alpha_1(x), \beta_0(x), \beta_1(x)$ 的表达式.

我们首先考虑 $\alpha_0(x)$. 由于 $\alpha_0(x)$ 是三次多项式, 且 $\alpha_0(x_1) = 0, \alpha_0'(x_1) = 0$, 所以可设

$$\alpha_0(x) = (ax + b)\left(\frac{x - x_1}{x_0 - x_1}\right)^2.$$

将 $\alpha_0(x_0) = 1, \alpha_0'(x_0) = 0$ 代入可得

$$a = \frac{2}{x_1 - x_0}, \quad b = \frac{x_1 - 3x_0}{x_1 - x_0} = 1 - \frac{2x_0}{x_1 - x_0}.$$

所以

$$\alpha_0(x) = \left(1 + 2\frac{x - x_0}{x_1 - x_0}\right)\left(\frac{x - x_1}{x_0 - x_1}\right)^2.$$

同理可得

$$\alpha_1(x) = \left(1 + 2\frac{x - x_1}{x_0 - x_1}\right)\left(\frac{x - x_0}{x_1 - x_0}\right)^2.$$

下面考虑 $\beta_0(x)$. 根据插值条件 $\beta_0(x_0) = 0, \beta_0(x_1) = 0, \beta_0'(x_1) = 0$, 可设

$$\beta_0(x) = a(x - x_0)\left(\frac{x - x_1}{x_0 - x_1}\right)^2.$$

将 $\beta_0'(x_0) = 0$ 代入可得 $a = 1$, 所以

$$\beta_0(x) = (x - x_0)\left(\frac{x - x_1}{x_0 - x_1}\right)^2.$$

同理可得

$$\beta_1(x) = (x - x_1)\left(\frac{x - x_0}{x_1 - x_0}\right)^2.$$

所以

$$\begin{aligned}
H_3(x) = {} & f(x_0)\left(1 + 2\frac{x - x_0}{x_1 - x_0}\right)\left(\frac{x - x_1}{x_0 - x_1}\right)^2 + f(x_1)\left(1 + 2\frac{x - x_1}{x_0 - x_1}\right)\left(\frac{x - x_0}{x_1 - x_0}\right)^2\\
& + f'(x_0)(x - x_0)\left(\frac{x - x_1}{x_0 - x_1}\right)^2 + f'(x_1)(x - x_1)\left(\frac{x - x_0}{x_1 - x_0}\right)^2.
\end{aligned} \tag{2.21}$$

插值余项为

$$R_3(x) = \frac{f^{(4)}(\xi_x)}{4!}(x-x_0)^2(x-x_1)^2.$$

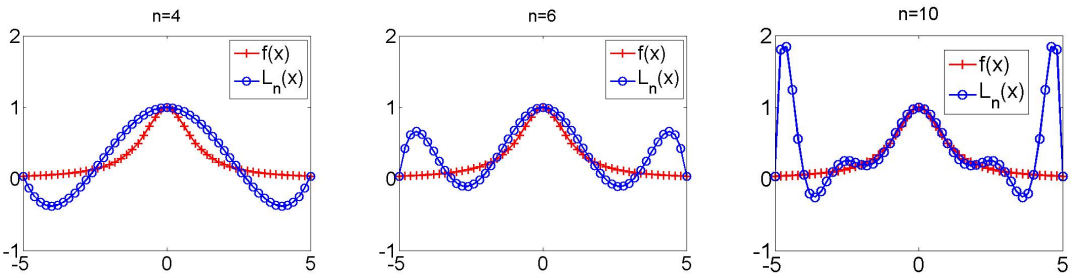
## 2.5 分段低次插值

### 2.5.1 为什么分段插值

当 $n \to \infty$ 时, 插值多项式 $p_n(x)$ 并不一定收敛于 $f(x)$. 我们可以通过下面的例子来演示这一点.

**例 2.12 (Runge 函数的等距节点插值多项式)** 设 $f(x) = \dfrac{1}{1+x^2}$, $x \in [-5,5]$, 取等距插值节点 $x_i = -5 + \frac{10i}{n}$, 其中 $n$ 是区间等分数. 试画出 $f(x)$ 的 $n$ 次 Lagrange 插值多项式 $L_n(x)$ 的图形, 并比较与原函数的误差.

**解.** MATLAB 演示见 `ex22.m`. 下图给出了 $f(x)$ 的图形与 $n = 4, 6, 10$ 时 $L_n(x)$ 的图形.




我们发现, 当 $n$ 增大时, 两端点附近的插值误差会越来越大. 事实上, 可以证明, 存在常数 $c \approx 3.63$, 当 $|x| \le c$ 时, $\lim\limits_{n\to\infty} L_n(x) = f(x)$, 而当 $|x| > c$ 时, $\{L_n(x)\}$ 发散. □

♣ Runge 现象说明插值多项式的次数并非越高越好.

为了提高插值精度, 我们需要尽可能地控制插值余项的大小. 通常, 插值余项有两部分组成: $f(x)$ 的导数和 $\omega_{n+1}(x)$. 由于 $f(x)$ 是给定的, 因此其导数值也是确定的. 所以我们只能想办法尽量降低 $\max\limits_{a\le x\le b}|\omega_n(x)|$ 的大小.

一个切实可行的方法就是分段插值方法, 即将插值区间分割成若干小区间, 然后在每个小区间上进行插值.

下面我们介绍两个常用的分段插值方法: 分段线性插值和分段三次 Hermite 插值.

### 2.5.2 分段线性插值

**定义 2.6** 设 $a = x_0 < x_1 < \cdots < x_n = b$ 为 $[a,b]$ 上的互异节点, 已知 $f(x)$ 在这些节点上的函数值为 $f_0, f_1, \ldots, f_n$. 求分段函数 $I_h(x)$ 满足

(1) $I_h(x) \in C[a,b]$;

(2) $I_h(x_k) = f_k, \quad k = 0, 1, 2, \ldots, n$;

(3) $I_h(x)$ 在每个小区间 $[x_k, x_{k+1}]$ 上是线性多项式.

这就是分段线性插值, $I_h(x)$ 就称为 $f(x)$ 在 $[a, b]$ 上的分段线性插值函数.

由定义直接可知 $I_h(x)$ 在小区间 $[x_k, x_{k+1}]$ 上的表达式为

$$I_h(x) = \frac{x - x_{k+1}}{x_k - x_{k+1}} f_k + \frac{x - x_k}{x_{k+1} - x_k} f_{k+1}, \quad x \in [x_k, x_{k+1}], \tag{2.22}$$

且在 $[x_k, x_{k+1}]$ 上余项满足

$$\max_{x_k \le x \le x_{k+1}} |f(x) - I_h(x)| \le \frac{1}{2!} \max_{x_k \le x \le x_{k+1}} |f''(x)| \cdot \max_{x_k \le x \le x_{k+1}} |(x - x_k)(x - x_{k+1})| \tag{2.23}$$

$$\le \frac{h_k^2}{8} \max_{x_k \le x \le x_{k+1}} |f''(x)|, \tag{2.24}$$

其中 $h_k = x_{k+1} - x_k$. 令 $h = \max\limits_{0 \le k \le n-1} \{h_k\}$, 我们有下面的结论.

**定理 2.4** 若 $f(x) \in C^2[a, b]$, 则分段线性插值函数 $I_h(x)$ 满足

$$\max_{a \le x \le b} |f(x) - I_h(x)| \le \frac{M_2}{8} h^2,$$

其中 $M_2 = \max\limits_{a \le x \le b} |f''(x)|$. 所以

$$\lim_{h \to 0} I_h(x) = f(x)$$

在 $[a, b]$ 上一致成立, 即 $I_h(x)$ 在 $[a, b]$ 上一致收敛到 $f(x)$.

证明. 由 (2.24) 可知

$$\max_{x_k \le x \le x_{k+1}} |f(x) - I_h(x)| \le \frac{h_k^2 M_2}{8} \le \frac{h^2 M_2}{8}.$$

所以

$$\max_{a \le x \le b} |f(x) - I_h(x)| \le \max_{0 \le k \le n-1} \max_{x_k \le x \le x_{k+1}} |f(x) - I_h(x)| \le \max_{0 \le k \le n-1} \frac{h^2 M_2}{8} = \frac{h^2 M_2}{8}.$$

□

♣ 分段线性插值简单易用, 但插值函数在插值节点不可导.

思考 如果是分段抛物线插值, 则结论是怎样的? (见习题中的思考题)

### 2.5.3 分段三次 Hermite 插值

**定义 2.7** 设 $a = x_0 < x_1 < \cdots < x_n = b$ 为 $[a, b]$ 上的互异节点, 已知 $f(x)$ 在这些节点上的函数值和导数分别为 $f_0, f_1, \ldots, f_n$ 和 $f_0', f_1', \ldots, f_n'$. 求分段函数 $I_h(x)$ 满足

(1) $I_h(x) \in C^1[a, b]$;

(2) $I_h(x_k) = f_k, I_h'(x_k) = f_k', \quad k = 0, 1, 2, \ldots, n$;

(3) $I_h(x)$ 在每个小区间 $[x_k, x_{k+1}]$ 上是三次多项式.

这就是分段三次 Hermite 插值, $I_h(x)$ 就称为 $f(x)$ 在 $[a,b]$ 上的分段三次 Hermite 插值函数.

由两点三次 Hermite 插值公式 (2.21) 可知, $I_h(x)$ 在小区间 $[x_k, x_{k+1}]$ 上的表达式为

$$\begin{aligned} I_h(x) = & \left(1+2\frac{x-x_k}{x_{k+1}-x_k}\right)\left(\frac{x-x_{k+1}}{x_k-x_{k+1}}\right)^2 f_k \\ & + \left(1+2\frac{x-x_{k+1}}{x_k-x_{k+1}}\right)\left(\frac{x-x_k}{x_{k+1}-x_k}\right)^2 f_{k+1} \\ & + (x-x_k)\left(\frac{x-x_{k+1}}{x_k-x_{k+1}}\right)^2 f'_k + (x-x_{k+1})\left(\frac{x-x_k}{x_{k+1}-x_k}\right)^2 f'_{k+1}, \\ & x \in [x_k, x_{k+1}]. \end{aligned} \tag{2.25}$$

由两点三次 Hermite 插值法的余项公式, 可知

$$\max_{x_k\le x\le x_{k+1}} |f(x)-I_h(x)| \le \frac{h_k^4}{384} \max_{x_k\le x\le x_{k+1}} |f^{(4)}(x)|,$$

其中 $h_k = x_{k+1}-x_k$. 令 $h = \max\limits_{0\le k\le n-1}\{h_k\}$, 我们有下面的结论.

**定理 2.5** 若 $f(x)\in C^4[a,b]$, 则分段三次 Hermite 插值函数 $I_h(x)$ 满足

$$\max_{a\le x\le b} |f(x)-I_h(x)| \le \frac{M_4}{384}h^4,$$

其中 $M_4 = \max\limits_{a\le x\le b}|f^{(4)}(x)|$. 所以, $I_h(x)$ 在区间 $[a,b]$ 上一致收敛到 $f(x)$.

♣ 由余项公式可知, 当 $h$ 比较小时, 分段三次 Hermite 插值比分段线性插值具有更高的精度, 且 $h$ 越小, 误差下降也越快.

♣ 分段三次 Hermite 插值的缺点: 需要知道 $f(x)$ 在插值节点的导数值, 而且插值函数只有一阶导数, 光滑度不高.

**例 2.13 (编程)** 设 $f(x)=\dfrac{1}{1+x^2}$, $x\in[-5,5]$, 取等距插值节点 $x_k=-5+k$ (即 10 等分插值区间). 试分别用分段线性插值和分段三次 Hermite 插值画出 $f(x)$ 的近似图像.

**解.** MATLAB 程序见 `ex27.m`. □

MATLAB 源代码 2.3. 分段线性插值和分段三次 Hermite 插值

```
clear; clc;
f = @(x) 1./(1+x.^2); % 函数表达式
df= @(x) -(2*x)./(x.^2+1).^2; % 一阶导数, 用于 Hermite 插值
a=-5; b=5; % 插值区间
n=10;      % 插值区间等分数
h=(b-a)/n; % 步长
xi=a:h:b;  % 插值节点
fi=f(xi);  % 插值节点上的函数值
```

```
dfi=df(xi); % 一阶导数值

x=a:(b-a)/50:b; % 需要插值的点，绘图用

% 定义线性插值函数
L1=@(x,x0,x1,f0,f1) f0*(x-x1)/(x0-x1)+f1*(x-x0)/(x1-x0);

% 定义两点三次 Hermite 插值函数
H3=@(x,x0,x1,f0,f1,df0,df1) ...
   (f0*(1+2*(x-x0)/(x1-x0))+df0*(x-x0))*((x-x1)/(x0-x1))^2 + ...
   (f1*(1+2*(x-x1)/(x0-x1))+df1*(x-x1))*((x-x0)/(x1-x0))^2;

% 分段线性插值
N=length(x);
y1=zeros(1,N); % 分段线性插值
y2=zeros(1,N); % 分段三次 Hermite 插值
for j=1:N
   for k=1:n+1 % 寻找 x(j) 所在的小区间 [x_k, x_{k+1}]
      if xi(k) >= x(j)
         break; % 找到第一个不小于 x(j) 的插值节点
      end
   end
   if k>1
      k=k-1;
   end
   y1(j)=L1(x(j),xi(k),xi(k+1),fi(k),fi(k+1));
   y2(j)=H3(x(j),xi(k),xi(k+1),fi(k),fi(k+1),dfi(k),dfi(k+1));
end

% 绘图
hold on;
plot(x,f(x),'r-'); % f(x) 图像
plot(x,y1,'b-'); % 分段线性插值图像
plot(x,y2,'k-'); % 分段三次 Hermite 插值图像
legend('f(x)','L1(x)','H3(x)')
plot(xi,fi,'^g','markersize',15); % 绘制插值节点
hold off
```

## 2.6 三次样条插值

为了增加分段插值函数的光滑性, 我们可以使用样条函数进行插值. 目前常用的为三次样条函数, 它具有二阶连续导数.

**定义 2.8** 设 $a = x_0 < x_1 < \cdots < x_n = b$ 为 $[a,b]$ 上的互异节点, 已知 $f(x)$ 在这些节点上的函数值为 $f(x_k) = f_k, k = 0,1,\ldots,n$. 求插值函数 $S(x)$ 满足

(1) $S(x) \in C^2[a,b]$, 即二阶连续可导;

(2) $S(x_k) = y_k, \quad k = 0,1,2,\ldots,n$;

(3) $S(x)$ 是分段三次函数, 即在每个小区间 $[x_k, x_{k+1}]$ 上是三次多项式.

这就是三次样条插值, $S(x)$ 就称为 $f(x)$ 在 $[a,b]$ 上的三次样条插值函数.

### 2.6.1 三次样条函数

**定义 2.9 (三次样条函数)** 设 $a = x_0 < x_1 < \cdots < x_n = b$ 为 $[a,b]$ 上的互异节点, 若函数 $S(x) \in C^2[a,b]$, 且在每个小区间 $[x_k, x_{k+1}]$ 上是三次多项式, 则称其为三次样条函数.

我们可以将 $S(x)$ 在小区间 $[x_k, x_{k+1}]$ 上的表达式记为 $s_k(x)$, 即

$$S(x) = s_k(x), \quad x \in [x_k, x_{k+1}], \quad k = 0,1,2,\ldots,n-1,$$

其中 $s_k(x)$ 是三次多项式, 且满足

$$s_k(x_k) = f_k, \quad s_k(x_{k+1}) = f_{k+1}. \tag{2.26}$$

于是

$$S(x) = \begin{cases} s_0(x), & x \in [x_0, x_1] \\ s_1(x), & x \in [x_1, x_2] \\ \quad\vdots & \\ s_{n-1}(x), & x \in [x_{n-1}, x_n]. \end{cases} \tag{2.27}$$

由于 $S(x) \in C^2[a,b]$, 所以 $S'(x_k^-) = S'(x_k^+)$, $S''(x_k^-) = S''(x_k^+)$, 即

$$s'_{k-1}(x_k^-) = s'_k(x_k^+), \quad s''_{k-1}(x_k^-) = s''_k(x_k^+), \quad k = 1,2,\ldots,n-1. \tag{2.28}$$

每个 $s_k(x)$ 均为三次多项式, 有 4 个待定系数, 所以共有 $4n$ 个待定系数, 故需 $4n$ 个方程. 由 (2.26) 和 (2.28) 可以得到 $2n + 2(n-1) = 4n - 2$ 个方程, 还缺 2 个方程!

♣ 实际问题中, 通常会对样条函数 $S(x)$ 在两个端点 $x = a$ 和 $x = b$ 处的状态有一定的要求, 这就是边界条件.

### 2.6.2 边界条件

我们这里介绍三类常用的边界条件.

(1) 第一类边界条件: 指定函数在两端点处的一阶导数, 即

$$S'(x_0) = f'_0, \quad S'(x_n) = f'_n$$

(2) 第二类边界条件: 指定函数在端点处的二阶导数, 即

$$S''(x_0) = f_0'', \quad S''(x_n) = f_n''.$$

如果 $f_0'' = f_n'' = 0$, 则称为自然边界条件, 此时 $S(x)$ 称为自然样条函数.

(3) 第三类边界条件: 假定 $f(x)$ 是周期函数, 并设 $x_n - x_0$ 是一个周期, 于是要求 $S(x)$ 也是周期函数, 即

$$S(x_0) = S(x_n), \quad S'(x_0^+) = S'(x_n^-), \quad S''(x_0^+) = S''(x_n^-).$$

此时 $S(x)$ 称为周期样条函数.

♣ 由于 $S(x_0) = f_0$ 和 $S(x_n) = f_n$ 是已知的, 所以第三类边界中只有后面两个才是新增加的约束.

### 2.6.3 三次样条函数的计算

由于 $S(x)$ 二阶可导, 所以可设

$$S''(x_k) = M_k, \quad k = 0, 1, 2, \ldots, n,$$

下面我们用 $M_k$ 来表示 $S(x)$. 考虑 $S(x)$ 在区间 $[x_k, x_{k+1}]$ 上的表达式 $s_k(x)$, 满足

$$s_k''(x_k) = M_k, \quad s_k''(x_{k+1}) = M_{k+1}.$$

由于 $s_k(x)$ 是三次多项式, 故 $s_k''(x)$ 为线性函数. 所以由线性插值公式可知

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

其中 $h_k = x_{k+1} - x_k$. 两边在 $[x_k, x_{k+1}]$ 上积分两次后可得

$$s_k(x) = \frac{(x_{k+1} - x)^3}{6h_k} M_k + \frac{(x - x_k)^3}{6h_k} M_{k+1} + c_1 x + c_2, \tag{2.29}$$

其中 $c_1, c_2$ 为积分常数. 将 $s_k(x_k) = f_k$, $s_k(x_{k+1}) = f_{k+1}$ 代入后可得

$$c_1 = \frac{1}{h_k}(f_{k+1} - f_k) - \frac{h_k}{6}(M_{k+1} - M_k) = \frac{1}{h_k}\left[\left(f_{k+1} - \frac{M_{k+1}h_k^2}{6}\right) - \left(f_k - \frac{M_k h_k^2}{6}\right)\right],$$

$$c_2 = f_k - \frac{M_k h_k^2}{6} - c_1 x_k = \frac{x_{k+1}}{h_k}\left(f_k - \frac{M_k h_k^2}{6}\right) - \frac{x_k}{h_k}\left(f_{k+1} - \frac{M_{k+1}h_k^2}{6}\right).$$

代入 (2.29), 整理后可得

$$\begin{aligned} s_k(x) = {} & \frac{(x_{k+1} - x)^3}{6h_k} M_k + \frac{(x - x_k)^3}{6h_k} M_{k+1} \\ & + \frac{x_{k+1} - x}{h_k}\left(f_k - \frac{M_k h_k^2}{6}\right) + \frac{x - x_k}{h_k}\left(f_{k+1} - \frac{M_{k+1}h_k^2}{6}\right). \end{aligned} \tag{2.30}$$

即 $s_k(x)$ 可表示成 $x_{k+1} - x$ 和 $x - x_k$ 的奇次项的线性组合.

将 $x_{k+1} = x_k + h_k$ 代入 (2.30), 整理后可得

$$\begin{aligned} s_k(x) = {} & \frac{M_{k+1} - M_k}{6h_k}(x - x_k)^3 + \frac{M_k}{2}(x - x_k)^2 \\ & + \left(\frac{f_{k+1} - f_k}{h_k} - \frac{h_k(M_{k+1} + 2M_k)}{6}\right)(x - x_k) + f_k. \end{aligned} \tag{2.31}$$

♣ 现在, 问题转化为如何确定 $M_0, M_1, \ldots, M_n$ 的值?

由于 $S(x) \in C^2[a,b]$, 所以在节点处的一阶导数也存在, 故

$$S'(x_k^-) = S'(x_k^+), \quad k = 1, 2, \ldots, n-1,$$

也即

$$s'_{k-1}(x_k^-) = s'_k(x_k^+).$$

所以可得方程

$$\frac{h_{k-1}}{6}M_{k-1} + \frac{h_{k-1}+h_k}{3}M_k + \frac{h_k}{6}M_{k+1} = \frac{f_{k+1}-f_k}{h_k} - \frac{f_k - f_{k-1}}{h_{k-1}}.$$

为了书写方便, 我们记

$$\begin{aligned} \mu_k &= \frac{h_{k-1}}{h_{k-1}+h_k}, \quad \lambda_k = \frac{h_k}{h_{k-1}+h_k}, \\ d_k &= \frac{6(f[x_k, x_{k+1}] - f[x_{k-1}, x_k])}{h_{k-1}+h_k} = 6f[x_{k-1}, x_k, x_{k+1}], \end{aligned}$$

则上面的方程可写为

$$\mu_k M_{k-1} + 2M_k + \lambda_k M_{k+1} = d_k, \quad k = 1, 2, \ldots, n-1. \tag{2.32}$$

♣ 上述方程中的系数有个重要性质: $\mu_k + \lambda_k = 1$

♣ 方程 (2.32) 也可以写成

$$h_{k-1}M_{k-1} + 2(h_{k-1}+h_k)M_k + h_k M_{k+1} = 6(f[x_k, x_{k+1}] - f[x_{k-1}, x_k])$$

这里有 $n+1$ 个变量, 但只有 $n-1$ 个方程. 此时需要通过边界条件增加两个方程. 下面对三种边界条件分别讨论.

**(1) 第一类边界条件**

给出函数在两端点处的一阶导数: $S'(x_0) = f'_0$ 和 $S'(x_n) = f'_n$, 即

$$s'_0(x_0^+) = f'_0, \quad s'_n(x_n^-) = f'_n.$$

因此可得

$$\begin{aligned} 2M_0 + M_1 &= \frac{6}{h_0}(f[x_0, x_1] - f'_0) \\ M_{n-1} + 2M_n &= \frac{6}{h_{n-1}}(f'_n - f[x_{n-1}, x_n]). \end{aligned}$$

令 $d_0 = \frac{6}{h_0}(f[x_0, x_1] - f'_0)$, $d_n = \frac{6}{h_{n-1}}(f'_n - f[x_{n-1}, x_n])$, 则上式与 (2.32) 联立可得方程组

$$\begin{bmatrix} 2 & 1 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & 1 & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}. \tag{2.33}$$

这是一个 $(n+1) \times (n+1)$ 的线性方程组, 且系数矩阵严格对角占优, 因此存在唯一解. 我们可以使用 Gauss 消去法或追赶法来求解 (具体过程在后面的章节中描述).

**(2) 第二类边界条件**

给出函数在端点处的二阶导数: $S''(x_0) = f''_0$ 和 $S''(x_n) = f''_n$, 即

$$M_0 = f''_0, \quad M_n = f''_n,$$

可得方程组

$$\begin{bmatrix} 2 & \lambda_1 & & & \\ \mu_2 & 2 & \lambda_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-2} & 2 & \lambda_{n-2} \\ & & & \mu_{n-1} & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \begin{bmatrix} d_1 - \mu_1 f''_0 \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} - \lambda_{n-1} f''_n \end{bmatrix}. \tag{2.34}$$

这是一个 $(n-1) \times (n-1)$ 的线性方程组, 系数矩阵也严格对角占优, 因此存在唯一解.

**(3) 第三类边界条件**

要求 $S(x)$ 是周期函数, 满足

$$S'(x_0^+) = S'(x_n^-), \quad S''(x_0^+) = S''(x_n^-),$$

即

$$s'_0(x_0^+) = s'_n(x_n^-), \quad s''(x_0^+) = s''(x_n^-).$$

可得

$$\lambda_n M_1 + \mu_n M_{n-1} + 2M_n = d_n, \quad M_0 = M_n,$$

其中

$$\lambda_n = \frac{h_0}{h_0 + h_{n-1}}, \quad \mu_n = \frac{h_{n-1}}{h_0 + h_{n-1}}, \quad d_n = \frac{6(f[x_0, x_1] - f[x_{n-1}, x_n]}{h_0 + h_{n-1}}.$$

与 (2.32) 联立可得方程组

$$\begin{bmatrix} 2 & \lambda_1 & & & \mu_1 \\ \mu_2 & 2 & \lambda_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ \lambda_n & & & \mu_n & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}. \tag{2.35}$$

这是一个 $n \times n$ 的线性方程组, 系数矩阵也严格对角占优, 因此存在唯一解.

♣ 由于 $M_k$ 在力学中解释为细梁在 $x_k$ 截面处的弯矩, 因此方程组 (2.33), (2.34) 和 (2.35) 在工程中称为三弯矩方程.

### 2.6.4 具体计算过程

由上面的分析可知, 三次样条插值的具体计算过程如下:

(1) 根据给定的插值条件和边界条件写出关于 $M_0, M_1, \ldots, M_n$ 的线性方程组;

(2) 解线性方程组, 求得 $M_k$;

(3) 将 $M_k$ 代入 $s_k(x)$ 的表达式 (2.31), 得到 $S(x)$ 在插值区间 $[a, b]$ 上的分段表达式.

♣ MATLAB 提供了计算三次样条插值的函数: `spline`, 其输出结果为 $[a_3, a_2, a_1, a_0]$, 表示

$$s_k(x) = a_3(x - x_k)^3 + a_2(x - x_k)^2 + a_1(x - x_k) + a_0,$$

因此, 我们在计算时也可以将 $s_k(x)$ 写成上述形式, 即 (2.31) 式.

♣ 三次样条插值函数具有二阶可导, 但仅利用插值节点上的函数值, 再加上边界条件.

**例 2.14** 函数 $f(x)$ 定义在 $[27.7, 30]$ 上, 插值节点及相应函数值下表, 试求三次样条插值多项式 $S(x)$, 满足边界条件 $S'(27.7) = 3.0$, $S'(30) = -4.0$.

| $x$ | 27.7 | 28 | 29 | 30 |
|---|---|---|---|---|
| $f(x)$ | 4.1 | 4.3 | 4.1 | 3.0 |

**解.** (MATLAB 程序见 `ex28.m`) 做差商表

| $x_i$ | $f(x_i)$ | 一阶差商 | 二阶差商 |
|---|---|---|---|
| 27.7 | 4.1 | | |
| 28 | 4.3 | $2/3$ | |
| 29 | 4.1 | $-0.2$ | $-2/3$ |
| 30 | 3.0 | $-1.1$ | $-9/20$ |

由题意可知 $h_0 = 0.3, h_1 = 1.0, h_2 = 1.0$, 所以

$$\mu_1 = \frac{h_0}{h_0 + h_1} = \frac{3}{13}, \quad \lambda_1 = 1 - \mu_1 = \frac{10}{13},$$
$$\mu_2 = \frac{h_1}{h_1 + h_2} = \frac{1}{2}, \quad \lambda_2 = 1 - \mu_2 = \frac{1}{2},$$
$$d_0 = \frac{6}{h_0}(f[x_0, x_1] - f_0') = 20\left(\frac{2}{3} - 3.0\right) = -\frac{140}{3},$$
$$d_1 = 6f[x_0, x_1, x_2] = -4,$$
$$d_2 = 6f[x_1, x_2, x_3] = -2.7,$$
$$d_3 = \frac{6}{h_2}(f_3' - f[x_2, x_3]) = 6(-4 + 1.1) = -17.4.$$

因此可得线性方程组

$$\begin{bmatrix} 2 & 1 & & \\ \frac{3}{13} & 2 & \frac{10}{13} & \\ & \frac{1}{2} & 2 & \frac{1}{2} \\ & & 1 & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} -\frac{140}{3} \\ -4 \\ -2.7 \\ -17.4 \end{bmatrix},$$

解得 (追赶法)

$$M_3 = -\frac{4603}{505} \approx -9.115, \quad M_2 = \frac{419}{505} \approx 0.830,$$
$$M_1 = -\frac{40}{101} \approx 0.396, \quad M_0 = -\frac{7130}{303} \approx -23.531.$$

代入 $s_k(x)$ 的表达式 (2.30) 可得

$$S(x) = \begin{cases} 13.293(x - 27.7)^3 - 11.766(x - 27.7)^2 + 3.000(x - 27.7) + 4.1, & x \in [27.7, 28] \\ 0.072(x - 28)^3 + 0.198(x - 28)^2 - 0.470(x - 28) + 4.3, & x \in [28, 29] \\ -1.657(x - 29)^3 + 0.415(x - 29)^2 + 0.143(x - 29) + 4.1, & x \in [29, 30]. \end{cases}$$

□

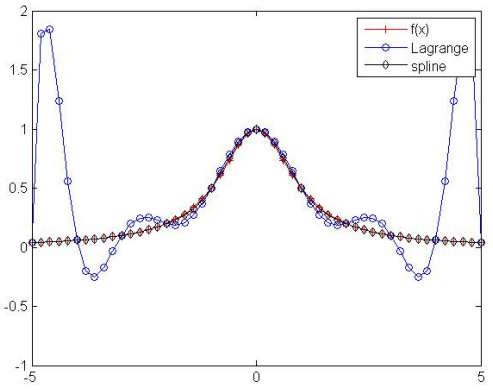
MATLAB 源代码 2.4. 样条插值举例

```
clear; clc;
X=[27.7, 28, 29, 30];
Y=[4.1, 4.3, 4.1, 3.0];
df0=3.0; dfn=-4.0;

n=length(X)-1;
H=diff(X); % 每个插值小区间的长度
mu=H(1:n-1) ./ (H(1:n-1)+H(2:n));
lambda=1-mu;
% 计算二阶差商
Y1=diff(Y) ./ diff(X); % 一阶差商
Y2=diff(Y1) ./ (X(3:end)-X(1:end-2)); % 二阶差商
% 计算右端项
```

```
d=6*Y2;
d0=6/H(1)*(Y1(1)-df0);
dn=6/H(end)*(dfn-Y1(end));
% 给出三弯矩方程组的系数矩阵和右端项
A=2*eye(n+1)+diag([mu(:);1],-1)+diag([1;lambda(:)],1);
b=[d0; d(:); dn];
M=A\b; % 解出 M_k

% 计算 s_k(x) 的系数, 两种形式
p1=zeros(n,4); p2=zeros(n,4);
for k=1:n
    p1(k,1)=(M(k+1)-M(k))/(6*H(k));
    p1(k,2)=M(k)/2;
    p1(k,3)=(Y(k+1)-Y(k))/H(k)-H(k)/6*(2*M(k)+M(k+1));
    p1(k,4)=Y(k);
    p2(k,1)=M(k)/(6*H(k));
    p2(k,2)=M(k+1)/(6*H(k));
    p2(k,3)=(Y(k)-M(k)*H(k)*H(k)/6)/H(k);
    p2(k,4)=(Y(k+1)-M(k+1)*H(k)*H(k)/6)/H(k);
end

pp=spline(X, [df0;Y(:);dfn]); % 调用 Matlab 的三次样条插值函数

%输出结果
fprintf('按形式一输出: \n'); disp(p2);
fprintf('按形式二输出: \n'); disp(p1);
fprintf('spline 的结果: \n'); disp(pp.coefs);
```

**例 2.15** 函数 $f(x)\dfrac{1}{1+x^2}$, 插值区间 $[-5,5]$, 取 11 个等距节点 (10 等分), 试画出 10 次插值多项式 $L_{10}(x)$ 与三次样条插值多项式 $S(x)$ 的函数图形.

**解**. MATLAB 程序见 ex29.m, 图形为

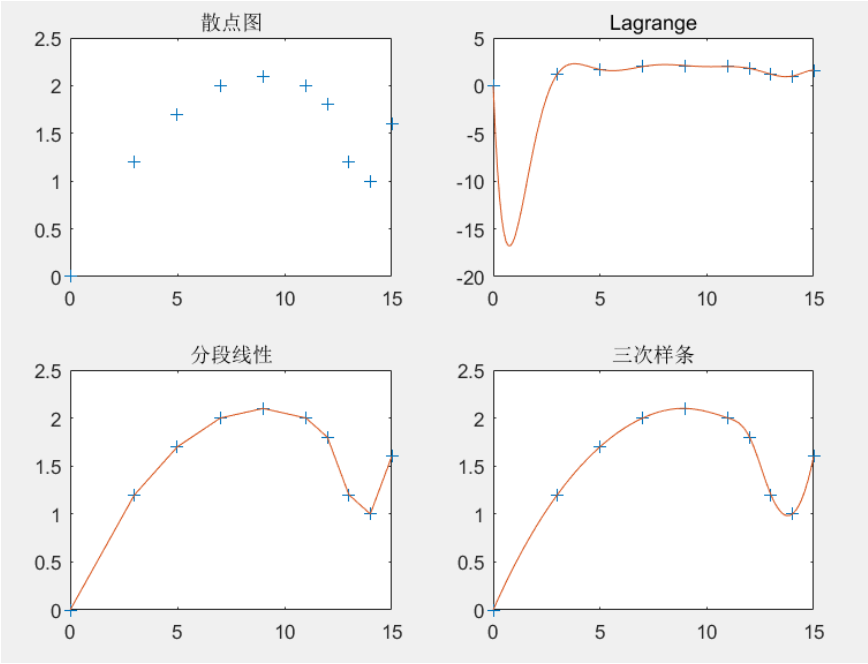



□

**例 2.16** 机床加工. 待加工零件的外形根据工艺要求由一组数据 $(x, y)$ 给出, 用程控铣床加工时每一刀只能沿 $x$ 方向和 $y$ 方向走非常小的一步, 这就需要从已知数据得到加工所要求的步长很小的 $(x, y)$ 坐标. 下表中给出的 $x, y$ 数据位于机翼断面的下轮廓线上, 假设需要得到 $x$ 坐标每改变 0.1 时的 $y$ 坐标. 试完成加工所需数据, 画出曲线. 要求用 Lagrange, 分段线性和三次样条三种插值方法计算.

| $x$ | 0 | 3 | 5 | 7 | 9 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|
| $y$ | 0 | 1.2 | 1.7 | 2.0 | 2.1 | 2.0 | 1.8 | 1.2 | 1.0 | 1.6 |

**解.** MATLAB 程序见 `ex2a.m`, 图形为



可以看出, Lagrange 插值的结果根本不能用. 分段线性插值的光滑性较差 (特别是在 $x = 14$ 附近弯曲处), 建议选用三次样条插值的结果. □

### 2.6.5 误差估计

**定理 2.6** 设 $f(x) \in C^4[a,b]$, $S(x)$ 为满足第一类或第二类边界条件的三次样条函数, 则

$$\max_{a\le x\le b}|f(x)-S(x)| \le \frac{5}{384}\max_{a\le x\le b}|f^{(4)}(x)|h^4,$$

$$\max_{a\le x\le b}|f'(x)-S'(x)| \le \frac{1}{24}\max_{a\le x\le b}|f^{(4)}(x)|h^3,$$

$$\max_{a\le x\le b}|f''(x)-S''(x)| \le \frac{3}{8}\max_{a\le x\le b}|f^{(4)}(x)|h^2,$$

其中 $h = \max\limits_{0\le k\le n-1}\{h_k\}$.

证明. 不作要求, 只需了解定理内容. □

♣ 该定理说明, 当 $h \to 0$ 时, $S(x)$ 及其一阶导数 $S'(x)$ 和二阶导数 $S''(x)$ 均收敛到 $f(x)$ 及其一阶导数 $f'(x)$ 和二阶导数 $f''(x)$.

## 2.7 课后练习

**练习 2.1** 当 $x=1,-1,2$ 时, $f(x)=0,-3,4$, 求 $f(x)$ 的二次插值多项式.

(1) 用单项式基函数;

(2) 用 Lagrange 基函数;

(3) 用 Newton 基函数.

证明三种方法得到的多项式是相同的.

**练习 2.2** 给出 $f(x)=\ln(x)$ 的数值表, 用线性插值和二次插值计算 $\ln(0.54)$ 的近似值.

**练习 2.3** 已知 $\cos(x)$ 在等距插值节点的函数值, 步长为 $h=1'=(1/60)^\circ$. 若函数值具有 5 位有效数字, 研究用线性插值求 $\cos(x)$ 近似值时的总误差界.

**练习 2.4** 设 $x_0,x_1,\dots,x_n$ 为互异节点, 求证:

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

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

**练习 2.5** 设 $f(x)\in C^2[a,b]$ 且 $f(a)=f(b)=0$, 求证:

$$\max_{a\le x\le b}|f(x)|\le\frac{1}{8}(b-a)^2\max_{a\le x\le b}|f''(x)|.$$

**练习 2.6** 已知 $f(x)=\mathrm{e}^x$ 在 $-4\le x\le 4$ 上的等距节点函数值表, 若用二次插值求 $\mathrm{e}^x$ 的近似值, 要使截断误差不超过 $10^{-6}$, 问步长 $h$ 应取多少?

**练习 2.7** 证明 $n$ 阶均差有下列性质:

(1) 若 $F(x)=cf(x)$, 则

$$F[x_0,x_1,\dots,x_n]=cf[x_0,x_1,\dots,x_n];$$

(2) 若 $F(x)=f(x)+g(x)$, 则

$$F[x_0,x_1,\dots,x_n]=f[x_0,x_1,\dots,x_n]+g[x_0,x_1,\dots,x_n].$$

**练习 2.8** 已知 $f(x)=x^7+x^4+3x+1$, 求 $f[2^0,2^1,\dots,2^7]$ 和 $f[2^0,2^1,\dots,2^8]$.

**练习 2.9** 证明: $\Delta(f_kg_k)=f_k\Delta g_k+g_{k+1}\Delta f_k$.

**练习 2.10** 证明: $\sum\limits_{k=0}^{n-1} f_k\Delta g_k=f_ng_n-f_0g_0-\sum\limits_{k=0}^{n-1} g_{k+1}\Delta f_k$.

**练习 2.11** 证明: $\sum\limits_{k=0}^{n-1}\Delta^2 y_k=\Delta y_n-\Delta y_0$.

**练习 2.12** 若 $f(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n$ 有 $n$ 个不同的零点 $x_1,x_2,\dots,x_n$, 证明

$$\sum_{j=1}^{n}\frac{x_j^k}{f'(x_j)}=\begin{cases}0, & 0\le k\le n-2;\\ a_n^{-1}, & k=n-1.\end{cases}$$

**练习 2.13** 求次数不超过 3 的多项式 $p(x)$, 满足

$$p(x_0)=f(x_0),\quad p(x_1)=f(x_1),\quad p'(x_0)=f'(x_0),\quad p''(x_0)=f''(x_0).$$

**练习 2.14** 求次数不超过 3 的多项式 $p(x)$, 满足

$$p(0)=0,\ p'(0)=1,\quad p(1)=1,\ p'(1)=2.$$

**练习 2.15** 证明两点三次 Hermite 插值余项为

$$R_3(x)=\frac{f^{(4)}(\xi_x)}{4!}(x-x_0)^2(x-x_1)^2,$$

其中 $\xi_x\in(x_0,x_1)$ 且与 $x$ 相关. 并由此给出分段三次 Hermite 插值的误差限.

**练习 2.16** 求一个次数不超过 4 的多项式 $p(x)$, 满足

$$p(0)=p'(0)=0,\quad p(1)=p'(1)=1,\quad p(2)=1.$$

(提示: 这里需要使用非标准的插值计算方法, 可以采用一些技巧, 不要死算!)

**练习 2.17** 设 $f(x)=\dfrac{1}{1+x^2}$, 在 $[-5,5]$ 上取 $n$ 等分点做分段线性插值, 计算 $n=10$ 时插值函数 $I_h(x)$ 在各区间中点处的值, 并估计误差.

**练习 2.18** 求 $f(x)=x^2$ 在 $[a,b]$ 上分段线性插值函数 $I_h(x)$, 并估计误差.

**练习 2.19** 求 $f(x)=x^4$ 在 $[a,b]$ 上分段三次 Hermite 插值函数 $I_h(x)$, 并估计误差.

**练习 2.20** 给定数据表如下:

| $x_k$ | 0.25 | 0.30 | 0.39 | 0.45 | 0.53 |
|---|---|---|---|---|---|
| $y_k$ | 0.5000 | 0.5477 | 0.6245 | 0.6708 | 0.7280 |

试求三次样条插值函数 $S(x)$, 满足

(1) $S'(0.25)=1.0000$, $S'(0.53)=0.6868$;

(2) $S''(0.25)=S''(0.53)=0$.

**练习 2.21** 设 $f(x)\in C^2[a,b]$, $S(x)$ 是三次样条函数

(1) 证明:

$$\begin{aligned}&\int_a^b[f''(x)]^2\,\mathrm{d}x-\int_a^b[S''(x)]^2\,\mathrm{d}x\\&=\int_a^b[f''(x)-S''(x)]^2\,\mathrm{d}x+2\int_a^bS''(x)[f''(x)-S''(x)]\,\mathrm{d}x;\end{aligned}$$

(2) 设插值节点 $a=x_0<x_1<\cdots<x_{n-1}<x_n=b$, 若 $S(x_k)=f(x_k)$, 证明:

$$\int_a^bS''(x)[f''(x)-S''(x)]\,\mathrm{d}x=S''(b)[f'(b)-S'(b)]-S''(a)[f'(a)-S'(a)].$$

## 思考题

**练习 2.22** 设 $a=x_0<x_1<\cdots<x_{2n}=b$ 为 $[a,b]$ 上的 $2n$ 个等距插值节点, 定义分段等距抛物线插值函数 $I_h(x)$ 如下:

(1) $I_h(x)\in C[a,b]$;

(2) $I_h(x_k)=f(x_k),\quad k=0,1,2,\ldots,2n$;

(3) $I_h(x)$ 在每个小区间 $[x_{2k},x_{2(k+1)}]$ 上是二次多项式, $k=0,1,2,\ldots,n$.

类似于定理 2.4 中的结论, 证明: 若 $f(x) \in C^3[a,b]$, 则

$$\max_{a\le x\le b} |f(x) - I_h(x)| \le \frac{\sqrt{3}}{27} M_3 h^3,$$

其中 $h = \frac{b-a}{2n}$, $M_3 = \max\limits_{a\le x\le b} |f^{(3)}(x)|$.

# 第三讲 函数逼近

函数逼近的基本思想就是使用简单易算的函数去近似表达式复杂的函数. 最简单的函数莫过于多项式函数, 因此, 人们首先考虑用多项式函数去做函数逼近. 对于闭区间上任意连续函数 $f(x)$, 由 Weierstrass 定理可知, 存在一个多项式序列一致收敛到 $f(x)$. 这就是用多项式函数逼近一般连续函数的理论基础.

## 3.1 基本概念与预备知识

### 3.1.1 什么是函数逼近

对于一个给定的复杂函数 $f(x)$, 在某个表达式较简单的函数类 $\Phi$ 中寻找一个函数 $p^*(x)$, 使其在某种度量下距离 $f(x)$ 最近, 即最佳逼近. 这就是函数逼近.

- 函数 $f(x)$ 通常较复杂, 但一般是连续的. 我们这里主要考虑 $[a,b]$ 上的连续函数, 即 $f(x)\in C[a,b]$;
- 函数类 $\Phi$ 通常为多项式函数, 或分段多项式函数, 或有理函数, 或三角多项式函数, 等等;
- 在不同的度量下, $f(x)$ 的最佳逼近可能不一样;
- 函数逼近通常采用基函数法.

若只给出函数在部分节点上的值, 且这些数值带有一定的误差. 在函数类 $\Phi$ 中寻找一个函数 $p(x)$, 使其在某种度量下是这些数据的最佳逼近, 这就是曲线拟合, 可以看作是离散情况下的函数逼近.

### 3.1.2 多项式逼近的理论基础

**定理 3.1 (Weierstrass 逼近定理)** 设 $f\in C[a,b]$, 则对任意的 $\varepsilon>0$ 存在一个多项式 $p(x)$, 使得

$$\max_{a\le x\le b}|f(x)-p(x)|<\varepsilon$$

在 $[a,b]$ 上一致成立.

证明. 该定理有多种证明方法, 其中 Bernstein 方法是一种构造性证明, 不仅证明了多项式的存在性, 而且也给出了构造方法. 详细的证明方法可参见相关数值逼近的文献, 这里不再给出. □

♣ 该定理也称为 Weierstrass 第一定理. 该定理表明, 任意一个闭区间上的连续函数都可以用多项式来一致逼近, 即实系数多项式构成的集合在 $C[a,b]$ 内是处处稠密的.

### 3.1.3 最佳逼近多项式

**定义 3.1** 设 $\Phi$ 为某个函数空间, 给定函数 $f(x) \in C[a,b]$, 若存在 $g^*(x) \in \Phi$, 使得

$$\|f(x) - g^*(x)\| = \min_{g(x)\in\Phi} \|f(x) - g(x)\|,$$

则称 $g^*(x)$ 为 $f(x)$ 在 $\Phi$ 中的 $[a,b]$ 上的最佳逼近函数.

♣ $g^*(x)$ 与函数空间 $\Phi$, 范数 $\|\cdot\|$ 和区间 $[a,b]$ 有关.

**定义 3.2** 设 $H_n$ 为所有次数不超过 $n$ 的多项式组成的函数空间, 给定函数 $f(x) \in C[a,b]$, 若存在 $p^*(x) \in H_n$, 使得

$$\|f(x) - p^*(x)\| = \min_{p(x)\in H_n} \|f(x) - p(x)\|,$$

则称 $p^*(x)$ 为 $f(x)$ 在 $[a,b]$ 上的 $n$ 次最佳逼近多项式. 若使用的范数为 $\|\cdot\|_\infty$, 则称 $p^*(x)$ 为 $n$ 次最佳一致逼近多项式; 若使用的范数为 $\|\cdot\|_2$, 则称 $p^*(x)$ 为 $n$ 次最佳平方逼近多项式.

如果只知道 $f(x)$ 在某些节点上的函数值 $f(x_i) = y_i$ $(i = 0,1,2,\ldots,m)$, 在某个函数空间 $\Phi$ 中寻找 $g^*(x)$ 使得

$$\sum_{i=1}^{m} |y_i - g^*(x_i)|^2 = \min_{g(x)\in\Phi} \sum_{i=1}^{m} \|y_i - g(x_i)\|^2,$$

则称 $g^*(x)$ 为 $f(x)$ 的最小二乘拟合. 若 $\Phi$ 取为 $H_n$, 则称 $g^*(x)$ 为 $f(x)$ 的 $n$ 次最小二乘拟合多项式.

♣ 这里一般有 $n < m$.

## 3.2 正交多项式

### 3.2.1 正交函数族与正交多项式

**定义 3.3 (正交函数)** 设 $f(x), g(x) \in C[a,b]$, $\rho(x)$ 是 $[a,b]$ 上的权函数, 若

$$(f,g) = \int_a^b \rho(x) f(x) g(x)\,\mathrm{d}x = 0,$$

则称 $f(x)$ 与 $g(x)$ 在 $[a,b]$ 上带权 $\rho(x)$ 正交.

♣ 正交与所使用的内积和权函数有关.

**定义 3.4 (正交函数族)** 设函数 $\varphi_0(x), \varphi_1(x), \ldots, \varphi_n(x), \ldots \in C[a,b]$, $\rho(x)$ 是 $[a,b]$ 上的权函数, 若

$$(\varphi_i, \varphi_j) = \int_a^b \rho(x)\varphi_i(x)\varphi_j(x)\,\mathrm{d}x = \begin{cases} 0, & i \neq j \\ A_i > 0, & i = j \end{cases} \quad i,j = 0,1,2,\ldots,$$

则称 $\{\varphi_n(x)\}_{n=0}^{\infty}$ 是 $[a,b]$ 上带权 $\rho(x)$ 的正交函数族.

♣ 如果所有的 $A_i$ 都等于 1, 则称为标准正交函数族.

**例 3.1** 三角函数系

$$1, \quad \cos x, \quad \sin x, \quad \cos 2x, \quad \sin 2x, \dots$$

在 $[-\pi, \pi]$ 上是带权 $\rho(x) = 1$ 的正交函数族.

**证明.** 由三角函数的 "积化和差" 公式可知 (也可以利用被积函数的奇偶性)

$$(1,1) = \int_{-\pi}^{\pi} \mathrm{d}x = 2\pi;$$

$$(\sin nx, \sin mx) = \int_{-\pi}^{\pi} \sin nx \sin mx \,\mathrm{d}x = \begin{cases} \pi, & m = n \\ 0, & m \neq n \end{cases} \quad m, n = 1, 2, \dots,$$

$$(\cos nx, \cos mx) = \int_{-\pi}^{\pi} \cos nx \cos mx \,\mathrm{d}x = \begin{cases} \pi, & m = n \\ 0, & m \neq n \end{cases} \quad m, n = 1, 2, \dots,$$

$$(\cos nx, \sin mx) = \int_{-\pi}^{\pi} \cos nx \sin mx \,\mathrm{d}x = 0, \quad m, n = 0, 1, 2, \dots.$$

□

**定义 3.5 (正交多项式)** 设 $p_n(x)$ 是首项系数不为零的 $n$ 次多项式, $n = 0, 1, 2, \dots$, $\rho(x)$ 是 $[a, b]$ 上的权函数, 若

$$(p_i, p_j) = \int_a^b \rho(x) p_i(x) p_j(x) \,\mathrm{d}x = \begin{cases} 0, & i \neq j \\ A_i > 0, & i = j \end{cases} \quad i, j = 0, 1, 2, \dots,$$

则称 $\{p_n(x)\}_{n=0}^{\infty}$ 为 $[a, b]$ 上带权 $\rho(x)$ 正交, 并称 $p_n(x)$ 为 $n$ 次正交多项式.

设 $\{p_n(x)\}_{n=0}^{\infty}$ 为 $[a, b]$ 上带权 $\rho(x)$ 正交多项式, 则显然 $p_0(x), p_1(x), p_2(x), \dots, p_n(x)$ 线性无关, 所以它们构成 $H_n$ 的一组正交基.

由于 $p_n(x)$ 与 $p_0(x), p_1(x), \dots, p_{n-1}(x)$ 都正交, 故 $p_n(x)$ 与 $H_{n-1}$ 中的任意多项式都正交, 即

$$(p_n(x), p(x)) = \int_a^b \rho(x) p_n(x) p(x) \,\mathrm{d}x = 0, \quad \forall\, p(x) \in H_{n-1}. \tag{3.1}$$

下面我们给出一个正交多项式的三项递推公式.

**定理 3.2** 设 $\{p_k(x)\}_{k=0}^{\infty}$ 为 $[a, b]$ 上带权 $\rho(x)$ 正交多项式, 且首项系数均为 1, 则有

$$p_{n+1}(x) = (x - \alpha_n) p_n(x) - \beta_n p_{n-1}(x), \quad n = 0, 1, 2, \dots,$$

其中 $p_{-1}(x) = 0, p_0(x) = 1$,

$$\alpha_n = \frac{(x p_n, p_n)}{(p_n, p_n)}, \quad \beta_n = \frac{(p_n, p_n)}{(p_{n-1}, p_{n-1})}, \quad n = 0, 1, 2, \dots.$$

**证明.** 由于 $\{p_0(x), p_1(x), \dots, p_n(x)\}$ 构成 $H_{n+1}$ 的一组基, 而 $x p_n(x) \in H_{n+1}$ 且首项系数为 1, 故 $x p_n(x)$ 可以表示为

$$x p_n(x) = \alpha_0 p_0(x) + \alpha_1 p_1(x) + \cdots + \alpha_n p_n(x) + p_{n+1}(x).$$

两边用 $p_k(x)$ 做内积, 利用 $\{p_k(x)\}_{k=0}^{\infty}$ 的正交性可得

$$(xp_n, p_k) = (p_{n+1}, p_k) + \sum_{i=0}^{n}(p_i, p_k) = \alpha_k(p_k, p_k).$$

又

$$(xp_n, p_k) = \int_a^b \rho(x)xp_n(x)p_k(x)\,\mathrm{d}x = (p_n, xp_k).$$

因此, 由正交多项式的性质 (3.1) 可知,

$$\alpha_k(p_k, p_k) = (xp_n, p_k) = (p_n, xp_k) = 0, \quad k = 0, 1, 2, \ldots, n-2.$$

即 $\alpha_0 = \alpha_1 = \cdots = \alpha_{n-2} = 0$. 于是 $xp_n(x) = p_{n+1}(x) + \alpha_n p_n(x) + \alpha_{n-1}p_{n-1}(x)$. 为了书写方便, 我们用 $\beta_n$ 来表示 $\alpha_{n-1}$, 即

$$xp_n(x) = p_{n+1}(x) + \alpha_n p_n(x) + \beta_n p_{n-1}(x).$$

两边用 $p_n(x)$ 做内积, 可得

$$\alpha_n = \frac{(xp_n, p_n)}{(p_n, p_n)}.$$

同理, 两边用 $p_{n-1}(x)$ 做内积, 可得

$$\beta_n = \frac{(xp_n, p_{n-1})}{(p_{n-1}, p_{n-1})}.$$

又

$$(xp_n, p_{n-1}) = (p_n, xp_{n-1}) = (p_n, p_n) + (p_n, xp_{n-1} - p_n) = (p_n, p_n),$$

其中 $(p_n, xp_{n-1} - p_n) = 0$ 是因为 $xp_{n-1} - p_n \in H_{n-1}$. 因此

$$\beta_n = \frac{(p_n, p_n)}{(p_{n-1}, p_{n-1})}.$$

□

♣ 所有首项系数为 1 的正交多项式族都满足这个公式, 该公式也给出了正交多项式的一个递推计算方法.

**定理 3.3** 设 $\{p_n(x)\}_{n=0}^{\infty}$ 是 $[a, b]$ 上带权 $\rho(x)$ 的正交多项式, 则当 $n \geq 1$ 时, $p_n(x)$ 在 $(a, b)$ 内有 $n$ 个不同零点.

证明. 假设 $p_n(x)$ 在 $(a, b)$ 内没有零点, 则 $p_n(x)$ 在 $(a, b)$ 内不变号, 故

$$\left|\int_a^b \rho(x)p_n(x)\,\mathrm{d}x\right| > 0.$$

另一方面, 由 (3.1) 可知

$$0 = (p_n, 1) = \int_a^b \rho(x)p_n(x)\,\mathrm{d}x.$$

矛盾. 因此 $p_n(x)$ 在 $(a, b)$ 内至少有一个零点.

假设 $p_n(x)$ 在 $(a, b)$ 内没有奇数重零点, 则 $p_n(x)$ 在 $(a, b)$ 内不变号, 同样可以导出矛盾. 因此 $p_n(x)$ 在 $(a, b)$ 内至少有一个奇数重零点.

设 $p_n(x)$ 在 $(a,b)$ 的所有奇数重零点为 $x_1, x_2, \dots, x_l$ $(1 \le l \le n)$. 构造多项式 $p_l(x) = (x-x_1)(x-x_2)\cdots(x-x_l)$. 则 $p_l(x)p_n(x)$ 在 $(a,b)$ 内只有偶数重零点, 因此在 $(a,b)$ 内不变号. 于是

$$|(p_n, p_l)| = \left| \int_a^b \rho(x) p_n(x) p_l(x)\, \mathrm{d}x \right| > 0.$$

如果 $l < n$, 则由 (3.1) 可知, $(p_n, p_l) = 0$, 矛盾. 因此 $l = n$. □

### 3.2.2 Gram-Schmidt 正交化

事实上, 任意一组线性无关的向量组 (或函数族), 均可通过 Gram-Schmidt 正交化过程产生一组正交的线性无关组.

易知 $\{1, x, x^2, \dots, x^n, \dots\}$ 是线性无关的. 相应的 Gram-Schmidt 正交化过程可描述为:

$$\begin{aligned} &p_0(x) = 1, \\ &p_k(x) = x^k - \sum_{j=0}^{k-1} c_{kj} p_j(x), \quad c_{kj} = \frac{(x^k, p_j)}{(p_j, p_j)}, \quad k = 1, 2, 3, \dots. \end{aligned}$$

### 3.2.3 Legendre 多项式

设 $[a,b] = [-1,1]$, 权函数 $\rho(x) = 1$, 将线性无关函数组 $\{1, x, x^2, \dots, x^n, \dots\}$ 正交化后得到的正交多项式就是 Legendre 多项式 (勒让德多项式), 记为

$$P_0(x), P_1(x), P_2(x), \dots.$$

Legendre 多项式的一般形式为

$$P_0(x) = 1, \quad P_n(x) = \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^2-1)^n, \quad x \in [-1,1], \quad n = 1, 2, \dots. \tag{3.2}$$

- $P_n(x)$ 的首项系数为 $\dfrac{(2n)!}{2^n (n!)^2}$;
- 若令 $\tilde{P}_n(x) = \dfrac{2^n (n!)^2}{(2n)!} P_n(x)$, 则称 $\tilde{P}_n(x)$ 为首项系数为 1 的 Legendre 多项式.

**定理 3.4 (正交性)**

$$(P_n, P_m) = \int_a^b P_n(x) P_m(x)\, \mathrm{d}x = \begin{cases} 0, & m \ne n \\ \dfrac{2}{2n+1}, & m = n \end{cases} \tag{3.3}$$

(板书)

**定理 3.5 (奇偶性)** $P_{2k}(x)$ 只含偶次幂, $P_{2k+1}(x)$ 只含奇次幂, 故

$$P_n(-x) = (-1)^n P_n(x).$$

证明. 由于 $(x^2-1)^n$ 至包含偶数次项, 因此由表达式 (3.2) 可知, 当 $n$ 是奇数时, $P_n(x)$ 只包含奇数次幂; 当 $n$ 是偶数时, $P_n(x)$ 只包含偶数次幂. □

**定理 3.6 (递推公式)**

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \quad n = 1, 2, \ldots,$$

其中 $P_0(x) = 1, P_1(x) = x$.

证明. 推导过程与定理 3.2 类似. 自己练习. □

**定理 3.7 (零点)** $P_n(x)$ 在 $(-1, 1)$ 内有 $n$ 个不同的零点.

证明. 直接由定理 3.3 可得. □

**例 3.2** 给出 5 次 Legendre 多项式 $P_5(x)$ 的表达式.

解. (MATLAB 程序见 `ex31.m`) 由递推公式可知

$$\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x \\
P_2(x) &= \frac{1}{2}(3xP_1(x) - P_0(x)) = \frac{1}{2}(3x^2 - 1) \\
P_3(x) &= \frac{1}{3}(5xP_2(x) - 2P_1(x)) = \frac{1}{2}(5x^3 - 3x) \\
P_4(x) &= \frac{1}{4}(7xP_3(x) - 3P_2(x)) = \frac{1}{8}(35x^4 - 30x^2 + 3) \\
P_5(x) &= \frac{1}{5}(9xP_4(x) - 4P_3(x)) = \frac{1}{8}(63x^5 - 70x^3 + 15x)
\end{aligned}$$

□

### 3.2.4 Chebyshev 多项式

设 $[a, b] = [-1, 1]$, 权函数 $\rho(x) = \dfrac{1}{\sqrt{1-x^2}}$, 将线性无关函数组 $\{1, x, x^2, \ldots, x^n, \ldots\}$ 正交化后得到的正交多项式就是 Chebyshev 多项式, 记为 $T_0(x), T_1(x), T_2(x), \ldots$.

Chebyshev 多项式的一般形式为

$$T_n(x) = \cos(n \arccos x), \quad x \in [-1, 1], \quad n = 1, 2, \ldots.$$

- 显然, $|T_n(x)| \leq 1$.
- 令 $x = \cos\theta, \theta \in [0, \pi]$, 则 $\theta = \arccos x$, 所以

$$\begin{aligned}
T_n(x) = \cos(n\theta) &= \cos^n\theta - C_n^2 \cos^{n-2}\theta \sin^2\theta + C_n^4 \cos^{n-4}\theta \sin^4\theta + \cdots \\
&= x^n - C_n^2 x^{n-2}(1-x^2) + C_n^4 x^{n-4}(1-x^2)^2 + \cdots.
\end{aligned}$$

故 $T_n(x)$ 是 $n$ 次多项式.

**定理 3.8 (正交性)**

$$(T_n, T_m) = \int_{-1}^{1} \rho(x) T_n(x) T_m(x)\,\mathrm{d}x = \begin{cases} 0, & m \neq n \\ \dfrac{\pi}{2}, & m = n \neq 0, \\ \pi, & m = n = 0. \end{cases}$$

证明. 留作练习. □

**定理 3.9 (奇偶性)** $T_{2n}(x)$ 只含偶次幂, $T_{2n+1}(x)$ 只含奇次幂, 故

$$T_n(-x) = (-1)^n T_n(x).$$

证明. 该性质可以通过下面的递推公式得到. □

**定理 3.10 (递推公式)**

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n = 1, 2, \ldots$$

其中 $T_0(x) = 1, T_1(x) = x$.

证明. 令 $x = \cos\theta$, 则由三角恒等式

$$\cos(n+1)\theta + \cos(n-1)\theta = 2\cos\theta\cos n\theta$$

可得

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

□

**定理 3.11 (零点)** $T_n(x)$ 在 $(-1, 1)$ 内有 $n$ 个不同的零点:

$$x_k = \cos\frac{2k-1}{2n}\pi, \quad k = 1, 2, \ldots, n.$$

证明. 由 $T_n(x) = 0$ 可得 $n\arccos x = \left(k - \dfrac{1}{2}\right)\pi$, 即

$$x_k = \cos\frac{2k-1}{2n}\pi, \quad k = 1, 2, \ldots, n.$$

□

**定理 3.12 (极值点)** $T_n(x)$ 在 $[-1, 1]$ 内有 $n+1$ 个极值点 (含两个端点):

$$\tilde{x}_k = \cos\frac{k\pi}{n}, \quad k = 0, 1, 2, \ldots, n.$$

证明. 直接求导可得

$$T_n'(x) = \frac{n\sin(n\arccos x)}{\sqrt{1-x^2}}.$$

令 $T_n'(x)=0$ 可得极值点

$$\tilde{x}_k=\cos\frac{k\pi}{n},\quad k=1,2,\dots,n-1.$$

再加上两个端点 $\tilde{x}_0=1$ 和 $\tilde{x}_n=-1$. □

**定理 3.13** $T_n(x)$ 的首项系数为 $2^{n-1}$.

证明. 直接由递推公式可知, $T_n(x)$ 的首项系数为 $2^{n-1}$. □

♣ 令 $\tilde{T}_n(x)=\frac{1}{2^{n-1}}T_n(x)$, 则称 $\tilde{T}_n(x)$ 为首项系数为 1 的 Chebyshev 多项式.

**例 3.3** 给出 5 次 Chebyshev 多项式 $T_5(x)$ 的表达式.

解. (MATLAB 程序见 `ex32.m`) 由递推公式可知

$$\begin{aligned}
T_0(x)&=1\\
T_1(x)&=x\\
T_2(x)&=2xT_1(x)-T_0(x)=2x^2-1\\
T_3(x)&=2xT_2(x)-T_1(x)=4x^3-3x\\
T_4(x)&=2xT_3(x)-T_2(x)=8x^4-8x^2+1\\
T_5(x)&=2xT_4(x)-T_3(x)=16x^5-20x^3+5x
\end{aligned}$$

□

### 3.2.5 Chebyshev 多项式零点插值

我们首先介绍 Chebyshev 多项式的一个重要性质.

**定理 3.14** 设 $\tilde{T}_n(x)$ 是首项系数为 1 的 Chebyshev 多项式, 即 $\tilde{T}_n(x)=\frac{1}{2^{n-1}}T_n(x)$, 则

$$\max_{-1\le x\le 1}|\tilde{T}_n(x)|\le\max_{-1\le x\le 1}|p(x)|,\quad\forall\,p(x)\in\tilde{H}_n,$$

其中 $\tilde{H}_n$ 表示次数不超过 $n$ 的所有首项系数为 1 的多项式组成的集合. 且

$$\max_{-1\le x\le 1}|\tilde{T}_n(x)|=\frac{1}{2^{n-1}}.$$

证明. 可参见相关文献. □

这个性质表明, 在次数不超过 $n$ 的所有首项系数为 1 的多项式中, $\tilde{T}_n(x)$ 在 $[-1,1]$ 上与零的偏差是最小的 (在无穷范数意义下).

♣ 该性质等价形式为

$$\|\tilde{T}_n(x)\|_\infty=\min_{p(x)\in\tilde{H}_n}\|p(x)\|_\infty,$$

即 $\tilde{T}_n(x)$ 是 $\tilde{H}_n$ 中无穷范数最小的. (注: 这里的 $\|\cdot\|_\infty$ 是指 $C[-1,1]$ 上的无穷范数)

♣ 该性质可用于计算首项系数非零的 $n$ 次多项式在 $[-1,1]$ 上的 $n-1$ 次最佳一致逼近多项式, 见第 3.4.4 小节.

利用定理 3.14, 我们可以采用 Chebyshev 多项式的零点作为节点进行多项式插值, 以使得插值的总体误差达到最小化.

设 $L_n(x)$ 是 $f(x)$ 在 $[-1,1]$ 上的 $n$ 次插值多项式, 插值节点为 $x_0, x_1, \dots, x_n$, 则插值余项为

$$f(x) - L_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x).$$

所以总体插值误差为

$$\max_{-1\le x\le 1} |f(x) - L_n(x)| \le \frac{M_{n+1}}{(n+1)!} \max_{-1\le x\le 1} |\omega_{n+1}(x)|,$$

其中 $M_{n+1} = \max\limits_{-1\le x\le 1} |f^{(n+1)}(x)|$. 因此, 要使得插值误差最小化, 我们就需要

$$\max_{-1\le x\le 1} |\omega_{n+1}(x)| = \|\omega_{n+1}(x)\|_\infty$$

尽可能地小. 由定理 3.14 可知, 当 $\omega_{n+1}(x) = \tilde{T}_{n+1}(x)$ 时, $\|\omega_{n+1}(x)\|_\infty$ 最小, 且最小值为 $\dfrac{1}{2^n}$. 相应的插值节点即为 $\tilde{T}_{n+1}(x)$ 的零点, 也就是 $T_{n+1}(x)$ 的零点.

**定理 3.15** 设 $f(x) \in C^{n+1}[-1,1]$, 插值节点为 $T_{n+1}(x)$ 的零点, 即

$$x_k = \cos\frac{2k+1}{2(n+1)}\pi, \quad k = 0, 1, 2, \dots, n.$$

令 $L_n(x)$ 是 $f(x)$ 在 $[-1,1]$ 上的 $n$ 次插值多项式, 则插值误差满足

$$\|f(x) - L_n(x)\|_\infty \le \frac{1}{2^n(n+1)!}\|f^{(n+1)}(x)\|_\infty. \tag{3.4}$$

♣ 上面的定理表明: 若 $f(x) \in C^{n+1}[-1,1]$, 则当 $n \to \infty$ 时, $L_n(x)$ 一致收敛到 $f(x)$.

如果插值区间是 $[a,b]$, 则需要做变量替换

$$x(t) = \frac{b-a}{2}t + \frac{b+a}{2}, \quad t \in [-1,1].$$

令 $t_k$ 为 Chebyshev 多项式 $T_{n+1}$ 的零点, 则插值节点为

$$x_k = \frac{b-a}{2}t_k + \frac{b+a}{2} = \frac{b-a}{2}\cos\frac{2k+1}{2(n+1)}\pi + \frac{b+a}{2}, \quad k = 0, 1, 2, \dots, n, \tag{3.5}$$

总体插值误差

$$\begin{aligned}\max_{a\le x\le b} |f(x) - L_n(x)| &\le \frac{1}{2^n(n+1)!} \max_{-1\le t\le 1} \left|\frac{\mathrm{d}^{n+1} f}{\mathrm{d}t^{n+1}}\right| \\ &= \frac{1}{2^n(n+1)!}\frac{(b-a)^{n+1}}{2^{n+1}} \max_{-1\le t\le 1} |f^{(n+1)}(x(t))| \\ &= \frac{(b-a)^{n+1}}{2^{2n+1}(n+1)!} \max_{a\le x\le b} |f^{(n+1)}(x)|.\end{aligned}$$

♣ 上面的总体误差也可以直接将插值点 (3.5) 代入 $\omega_{n+1}(x)$ 后获得. 事实上, 由于

$$x - x_k = \frac{b-a}{2}(t - t_k), \quad k = 0, 1, 2, \ldots, n,$$

故

$$\omega_{n+1}(x) = \prod_{k=0}^{n}(x - x_k) = \prod_{k=0}^{n}\frac{b-a}{2}(t - t_k) = \frac{(b-a)^{n+1}}{2^{n+1}}\tilde{T}_{n+1}(t).$$

因此

$$\max_{a\le x\le b}|\omega_{n+1}(x)| = \frac{(b-a)^{n+1}}{2^{n+1}}\max_{-1\le t\le 1}\tilde{T}_{n+1}(t) = \frac{(b-a)^{n+1}}{2^{n+1}}\cdot\frac{1}{2^n}$$

♣ 为了尽可能地减小插值误差, 在可以自由选取插值节点时, 我们尽量使用 Chebyshev 多项式零点.

**例 3.4** 求 $f(x) = \mathrm{e}^x$ 在 $[0, 1]$ 上的四次插值多项式 $L_4(x)$, 插值节点为 $T_5(x)$ 的零点, 并估计总体误差. (板书)

**例 3.5** 设 $f(x) = \dfrac{1}{1+x^2}$, 在 $[-5, 5]$ 上分别用等距节点和 Chebyshev 多项式零点做 10 次多项式插值, 绘图比较两种插值的数值效果.

**解.** MATLAB 程序见 `ex33.m` □

### 3.2.6 第二类 Chebyshev 多项式

在区间 $[-1, 1]$ 上, 带权 $\rho(x) = \sqrt{1-x^2}$ 正交的多项式就称为 **第二类 Chebyshev 多项式**, 其一般表达式为

$$U_n(x) = \frac{\sin((n+1)\arccos x)}{\sqrt{1-x^2}}, \quad x \in [-1, 1], \quad n = 0, 1, 2, \ldots. \tag{3.6}$$

- **正交性**:

$$(U_n, U_m) = \int_{-1}^{1}\rho(x)U_n(x)U_m(x)\,\mathrm{d}x = \begin{cases} 0, & m \ne n \\ \dfrac{\pi}{2}, & m = n \end{cases}$$

- **递推公式**:

$$U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x), \quad n = 1, 2, \ldots,$$

其中 $U_0(x) = 1, U_1(x) = 2x$.

### 3.2.7 Laguerre 多项式

在区间 $[0, \infty]$ 上, 带权 $\rho(x) = \mathrm{e}^{-x}$ 正交的多项式就称为 **Laguerre 多项式**, 其一般表达式为

$$L_n(x) = \mathrm{e}^x\frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^n\mathrm{e}^{-x}), \quad x \in [0, \infty], \quad n = 0, 1, 2, \ldots.$$

- **正交性**:

$$(L_n, L_m) = \int_0^{\infty}\rho(x)L_n(x)L_m(x)\,\mathrm{d}x = \begin{cases} 0, & m \ne n \\ (n!)^2, & m = n \end{cases}$$

- 递推公式:

$$L_{n+1}(x) = (2n+1-x)L_n(x) - n^2 L_{n-1}(x), \quad n = 1, 2, \ldots,$$

其中 $L_0(x) = 1, L_1(x) = 1 - x$.

#### 3.2.8 Hermite 多项式

在区间 $[-\infty, \infty]$ 上, 带权 $\rho(x) = \mathrm{e}^{-x^2}$ 正交的多项式就称为 Hermite 多项式, 其一般表达式为

$$H_n(x) = (-1)^n \mathrm{e}^{x^2} \frac{\mathrm{d}^n}{\mathrm{d}x^n}(\mathrm{e}^{-x^2}), \quad x \in [-\infty, \infty], \quad n = 0, 1, 2, \ldots.$$

- 正交性:

$$(H_n, H_m) = \int_{-\infty}^{\infty} \rho(x) H_n(x) H_m(x)\,\mathrm{d}x = \begin{cases} 0, & m \neq n \\ 2^n n! \sqrt{n}, & m = n \end{cases}$$

- 递推公式:

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \quad n = 1, 2, \ldots,$$

其中 $H_0(x) = 1, H_1(x) = 2x$.

### 3.3 最佳平方逼近

#### 3.3.1 什么是最佳平方逼近

设 $f(x) \in C[a, b]$, 在某个给定的简单易算的函数集 $\Phi \subset C[a, b]$ 中寻找 $S^*(x)$, 使得

$$\|f(x) - S^*(x)\|_2^2 = \min_{S(x) \in \Phi} \|f(x) - S(x)\|_2^2.$$

我们称 $S^*(x)$ 为 $f(x)$ 在 $\Phi$ 中的最佳平方逼近函数. 这里的范数 $\|\cdot\|_2$ 是 $C[a, b]$ 上的带权内积导出的范数, 即

$$\|f(x) - S(x)\|_2^2 = \int_a^b \rho(x)(f(x) - S(x))^2\,\mathrm{d}x.$$

#### 3.3.2 怎样求最佳平方逼近

设 $\varphi_0(x), \varphi_1(x), \ldots, \varphi_n(x)$ 是 $\Phi$ 的一组基, 则对任意 $S(x) \in \Phi$, $S(x)$ 可表示为

$$S(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x).$$

因此

$$\|f(x) - S(x)\|_2^2 = \int_a^b \rho(x) \left( f(x) - \sum_{i=0}^{n} a_i \varphi_i(x) \right)^2 \mathrm{d}x \triangleq I(a_0, a_1, \ldots, a_n).$$

这是一个关于 $a_0, a_1, \ldots, a_n$ 的多元函数. 于是, 求最佳逼近函数 $S^*(x)$ 就转化为求多元函数 $I(a_0, a_1, \ldots, a_n)$ 的最小值问题. 易知 $I(a_0, a_1, \ldots, a_n)$ 是一个正定二次型, 所以 $I(a_0, a_1, \ldots, a_n)$ 取最小值的充要条件是

$$\frac{\partial I(a_0, a_1, \ldots, a_n)}{\partial a_k} = 0, \quad k = 0, 1, 2, \ldots, n.$$

通过求导, 上式可化为

$$2\int_a^b \rho(x)\left(f(x)-\sum_{i=0}^{n} a_i\varphi_i(x)\right)\varphi_k(x)\,\mathrm{d}x=0, \tag{3.7}$$

也即

$$\sum_{i=0}^{n} a_i\int_a^b \rho(x)\varphi_i(x)\varphi_k(x)\,\mathrm{d}x=\int_a^b \rho(x)f(x)\varphi_k(x)\,\mathrm{d}x.$$

写成内积形式即为

$$\sum_{i=0}^{n}(\varphi_k,\varphi_i)a_i=(\varphi_k,f),\quad k=0,1,2,\ldots,n.$$

写成矩阵形式可得

$$\begin{bmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n)\\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n)\\ \vdots & \vdots & & \vdots\\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n)\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_n\end{bmatrix}=\begin{bmatrix}(\varphi_0,f)\\ (\varphi_1,f)\\ \vdots\\ (\varphi_n,f)\end{bmatrix}. \tag{3.8}$$

我们称这个方程为 **法方程**, 系数矩阵记为 $G$, 右端项记为 $d$. 由于 $\varphi_0(x),\varphi_1(x),\ldots,\varphi_n(x)$ 线性无关, 由推论 1.4 可知, 法方程 (3.8) 的系数矩阵 $G$ 非奇异, 因此法方程存在唯一解.

♣
- 求 $S^*(x)\Longleftrightarrow$ 解法方程 $Ga=d$.
- 存在唯一解 $\Longleftrightarrow G$ 非奇异 $\Longleftrightarrow \varphi_0,\varphi_1,\ldots,\varphi_n$ 线性无关.
- $S^*(x)$ 是 $f(x)$ 在 $\Phi$ 中的最佳逼近 $\Longleftrightarrow (f-S^*,\varphi_k)=0, k=0,1,\ldots,n\Longleftrightarrow (f-S^*,S)=0,$ $\forall\, S\in\Phi$.

**定理 3.16** 设 $a_0^*,a_1^*,\ldots,a_n^*$ 是法方程 (3.8) 的解, 则 $S^*(x)$ 是 $f(x)$ 在 $\Phi$ 中的最佳平方逼近函数, 其中

$$S^*(x)=a_0^*\varphi_0(x)+a_1^*\varphi_1(x)+\cdots+a_n^*\varphi_n(x).$$

**证明.** 对任意 $S(x)\in\Phi$, 有 $S(x)-S^*(x)\in\Phi$. 由于 $a_0^*,a_1^*,\ldots,a_n^*$ 是法方程 (3.8) 的解, 故

$$(f-S^*,\varphi_k)=0,\quad k=0,1,2,\ldots,n.$$

所以有

$$(f-S^*,S(x)-S^*(x))=0.$$

于是

$$\begin{aligned}\|f-S\|_2^2-\|f-S^*\|_2^2&=\int_a^b[(f-S)^2-(f-S^*)^2]\,\mathrm{d}x\\ &=\int_a^b[(S-S^*)^2-2(f-S^*)(S-S^*)]\,\mathrm{d}x\\ &=\int_a^b(S-S^*)^2\,\mathrm{d}x\geq 0,\end{aligned}$$

其中等号当且仅当 $S-S^*=0$, 即 $S=S^*$ 时成立. □

♣ 该定理给出了计算最佳平方逼近函数的一个方法.

记 $\delta(x)=f(x)-S^*(x)$ 为平方逼近误差. 由 (3.7) 可知 $(f-S^*,\varphi_k)=0$, 因此

$$\|\delta(x)\|_2^2=(f-S^*,f-S^*)=(f-S^*,f)=\|f\|_2^2-\sum_{i=0}^{n}a_i^*(\varphi_i,f).$$

### 3.3.3 用正交函数计算最佳平方逼近

通过法方程计算最佳平方逼近时, 需要解一个方程组, 当 $n$ 较大时, 会带来一定的困难. 因此我们考虑用正交函数族来计算最佳平方逼近.

设 $\varphi_0(x),\varphi_1(x),\dots,\varphi_n(x)$ 是 $\Phi$ 的一组正交基, 则法方程的系数矩阵就为一个对角矩阵, 即

$$G=\begin{bmatrix}(\varphi_0,\varphi_0)&&&\\&(\varphi_1,\varphi_1)&&\\&&\ddots&\\&&&(\varphi_n,\varphi_n)\end{bmatrix},$$

故法方程的解为

$$a_k^*=\frac{(\varphi_k,f)}{(\varphi_k,\varphi_k)},\quad k=0,1,2,\dots,n.$$

于是 $f(x)$ 在 $\Phi$ 中的最佳平方逼近函数为

$$S^*(x)=\sum_{k=0}^{n}\frac{(\varphi_k,f)}{(\varphi_k,\varphi_k)}\varphi_k(x), \tag{3.9}$$

误差

$$\|\delta(x)\|_2^2=\|f(x)-S^*(x)\|_2^2=(f,f)-(S^*,f)=\|f\|_2^2-\sum_{k=0}^{n}\frac{(\varphi_k,f)^2}{(\varphi_k,\varphi_k)}.$$

由于 $\|\delta(x)\|_2^2\geq 0$, 所以有

$$\sum_{k=0}^{n}\frac{(\varphi_k,f)^2}{(\varphi_k,\varphi_k)}\leq\|f\|_2^2,\quad\forall\, f\in C[a,b].$$

上述不等式就称为 Bessel 不等式.

### 3.3.4 广义 Fourier 级数

设 $\{\varphi_n(x)\}_{n=0}^{\infty}$ 是正交函数族, 对 $f(x)\in C[a,b]$, 构造级数

$$a_0^*\varphi_0(x)+a_1^*\varphi_1(x)+\cdots+a_n^*\varphi_n(x)+\cdots \tag{3.10}$$

其中 $a_k^*=\dfrac{(\varphi_k,f)}{(\varphi_k,\varphi_k)}$. 这就是关于 $f(x)$ 的广义 Fourier 级数, 它是 Fourier 级数的推广, 其中 $a_k^*$ 称为广义 Fourier 系数. 若正交函数族取为

$$1,\cos x,\sin x,\cos 2x,\sin 2x,\dots,$$

则级数 (3.10) 就是 Fourier 级数.

### 3.3.5 最佳平方逼近多项式

设 $\Phi = H_n$ (次数不超过 $n$ 的所有多项式组成的集合), 则 $f(x)$ 在 $\Phi$ 中的最佳平方逼近就称为 $f(x)$ 的 $n$ 次最佳平方逼近多项式, 记为 $S_n^*(x)$.

**例 3.6** 设 $[a,b]=[0,1]$, 权函数 $\rho(x)\equiv 1$, 取 $H_n$ 的一组基 $\{1,x,x^2,\dots,x^n\}$. 求 $f(x)\in C[0,1]$ 的 $n$ 次最佳平方逼近多项式.

**解.** 由于 $\varphi_i(x)=x^i$, 直接计算可得

$$(\varphi_i,\varphi_j)=\int_0^1 x^{i+j}\,\mathrm{d}x=\frac{1}{i+j+1}.$$

所以法方程的系数矩阵为

$$G=\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{n+1} & \frac{1}{n+2} & \frac{1}{n+3} & \cdots & \frac{1}{2n+1} \end{bmatrix} \triangleq H,$$

这就是著名的 Hilbert 矩阵. 右端项 $d=[d_0,d_1,\dots,d_n]^\intercal$, 其中

$$d_k=(\varphi_k,f)=\int_0^1 x^k f(x)\,\mathrm{d}x,\quad k=0,1,2,\dots,n.$$

解法方程可得 $a_k^*$, 于是 $f(x)$ 的 $n$ 次最佳平方逼近多项式为

$$S_n^*(x)=\sum_{k=0}^{n} a_k^* x^k.$$

□

**例 3.7** 设 $f(x)=\sqrt{1+x^2}$, 求 $f(x)$ 在 $[0,1]$ 上的一次最佳平方逼近多项式. (板书)

♣ Hilbert 矩阵对称正定, 但高度病态, 因此当维数较大时, 会给数值求解带来很大的困难. 因此, 我们很少用这种方法来求最佳平方逼近多项式.

### 3.3.6 用正交多项式计算最佳平方逼近多项式

**定理 3.17** 设 $f(x)\in C[a,b]$, $\{\varphi_n(x)\}_{n=0}^{\infty}$ 是正交多项式族, $S_n^*(x)$ 是由 (3.9) 给出的 $n$ 次最佳平方逼近多项式, 则

$$\lim_{n\to\infty}\|f(x)-S_n^*(x)\|_2=0,$$

即 $S_n^*(x)$ 一致收敛到 $f(x)$.

**证明.** 参见相关文献. □

下面考虑用 Legendre 多项式来计算最佳平方逼近多项式.

**定理 3.18** 设 $f(x)\in C[-1,1]$, 权函数 $\rho(x)\equiv 1$, 则 $f(x)$ 在 $[-1,1]$ 上的 $n$ 次最佳平方逼近多项式为

$$S_n^*(x)=a_0^*P_0(x)+a_1^*P_1(x)+\cdots+a_n^*P_n(x),$$

其中 $P_k(x)$ 为 $k$ 次 Legendre 多项式,

$$a_k^*=\frac{(P_k,f)}{(P_k,P_k)}=\frac{2k+1}{2}\int_{-1}^{1}P_k(x)f(x)\,\mathrm{d}x.$$

误差

$$\|\delta_n(x)\|_2^2=\|f(x)-S_n^*(x)\|_2^2=\|f(x)\|_2^2-\sum_{k=0}^{n}a_k^*(P_k,f)=\int_{-1}^{2}f^2(x)\,\mathrm{d}x-\sum_{k=0}^{n}\frac{2\left(a_k^*\right)^2}{2k+1}.$$

♣ 该定理给出了求解最佳平方逼近多项式的计算公式和误差表达式, 通常都使用这种方法计算最佳平方逼近多项式.

**例 3.8** 设 $f(x)=\mathrm{e}^x$, 求 $f(x)$ 在 $[-1,1]$ 上的三次最佳平方逼近多项式. (板书)

**定理 3.19** 设 $f(x)\in C^2[-1,1]$, 则对 $\forall\, x\in[-1,1]$ 和 $\forall\,\varepsilon>0$, 当 $n$ 充分大时, 有

$$|f(x)-S_n^*(x)|\leq\frac{\varepsilon}{\sqrt{n}}.$$

证明. 参见相关文献. □

**定理 3.20** 在所有首项系数为 1 的 $n$ 次多项式中, $\tilde{P}_n(x)$ 在 $[-1,1]$ 上与零的平方逼近误差最小, 即

$$\|\tilde{P}_n(x)\|_2=\min_{p(x)\in\hat{H}_n}\|p(x)\|_2=\min_{p(x)\in\hat{H}_n}\left(\int_{-1}^{1}p^2(x)\,\mathrm{d}x\right)^{\frac{1}{2}},$$

其中 $\tilde{P}_n(x)$ 是首项系数为 1 的 Legendre 多项式, $\hat{H}_n$ 表示所有首项系数为 1 的 $n$ 次多项式组成的集合. (板书)

♣ 这是 Legendre 多项式的一个重要性质, 与 $\tilde{T}_n(x)$ 的 "无穷范数最小" 性质 (见定理 3.14) 相类似.

### 3.3.7 如何计算一般区间上的最佳平方逼近多项式

计算过程如下:

(1) 做变换替换 $x(t)=\dfrac{b-a}{2}t+\dfrac{b+a}{2}$, 将 $f(x)$ 转化为 $g(t)=f(x(t)),\ t\in[-1,1]$;

(2) 通过 Legendre 多项式计算出 $g(t)$ 在 $[-1,1]$ 上的最佳平方逼近多项式 $S^*(t)$;

(3) 将 $t=\dfrac{2x-b-a}{b-a}$ 代入 $S^*(t)$, 给出 $f(x)$ 在 $[a,b]$ 上的最佳平方逼近多项式

$$S^*\left(\frac{2x-b-a}{b-a}\right).$$

## 3.4 最佳一致逼近

### 3.4.1 什么是最佳一致逼近

设 $f(x)\in C[a,b]$, 在某个给定的函数集 $\Phi\subset C[a,b]$ 中寻找 $S^*(x)$, 使得

$$\|f(x)-S^*(x)\|_\infty=\min_{S(x)\in\Phi}\|f(x)-S(x)\|_\infty.$$

我们称 $S^*(x)$ 为 $[a,b]$ 上 $f(x)$ 在 $\Phi$ 中的最佳一致逼近函数. 若 $\Phi=H_n$, 则称 $S^*(x)$ 为 $f(x)$ 在 $[a,b]$ 上的 $n$ 次最佳一致逼近多项式. 这里的范数 $\|\cdot\|_\infty$ 是 $C[a,b]$ 上的无穷范数, 即

$$\|f(x)-S(x)\|_\infty=\max_{a\le x\le b}|f(x)-S(x)|.$$

### 3.4.2 最佳一致逼近多项式的存在唯一性

**定理 3.21 (Chebyshev 定理)** 设 $f(x)\in C[a,b]$, 则 $f(x)$ 在 $[a,b]$ 上存在唯一的 $n$ 次最佳一致逼近多项式, 且 $p_n^*(x)$ 是 $f(x)$ 的 $n$ 次最佳一致逼近多项式的充要条件是 $f(x)-p_n^*(x)$ 在 $[a,b]$ 内至少存在 $n+2$ 个偏差点 $x_0,x_1,x_2,\dots,x_{n+1}$, 即

$$f(x_i)-p_n^*(x_i)=(-1)^i\max_{a\le x\le b}|p_n^*(x)-f(x)|,\quad i=0,1,2,\dots,n+1.$$

(证明可参见相关资料)

### 3.4.3 零次与一次最佳一致逼近多项式

作为例子, 我们考虑 $n=0$ 和 $n=1$ 时的情形.

**例 3.9** 设 $f(x)\in C[a,b]$, 则 $f(x)$ 的零次最佳一致逼近多项式为

$$p_0^*(x)=\frac{1}{2}\left(\min_{a\le x\le b}f(x)+\max_{a\le x\le b}f(x)\right).$$

**例 3.10** 设 $f(x)\in C^2[a,b]$ 且 $f''(x)>0, x\in[a,b]$, 求 $f(x)$ 的一次最佳一致逼近多项式.

**解.** 设 $f(x)$ 的一次最佳一致逼近多项式为

$$p_1^*(x)=\alpha x+\beta,\quad \alpha,\beta\in\mathbb{R}.$$

由 Chebyshev 定理 3.21 可知, $f(x)-p_1^*(x)$ 在 $[a,b]$ 内至少存在 3 个偏差点, 不妨设为 $x_0<x_1<x_2$. 则 $x_1$ 必定在 $(a,b)$ 内, 且 $x_1$ 为 $p_1^*(x)-f(x)$ 的驻点, 即

$$f'(x_1)-(p_1^*)'(x_1)=0.$$

所以 $f'(x_1)=\alpha$. 又 $f''(x)>0$, 即 $f'(x)$ 严格单调递增. 因此

$$\big(f(x)-p_1^*(x)\big)'=f'(x)-\alpha$$

在 $[a,b]$ 内不能再有其它零点. 因此 $f(x)-p_1^*(x)$ 至多只有 3 个偏差点, 且其它两个偏差点为端点. 所以 $f(x)-p_1^*(x)$ 只有 3 个偏差点 $x_0,x_1,x_2$, 且 $x_0=a, x_2=b$. 从而

$$f(a)-p_1^*(a)=-\big(f(x_1)-p_1^*(x_1)\big)=f(b)-p_1^*(b).$$

代入后解得

$$\alpha = \frac{f(b)-f(a)}{b-a}, \quad \beta = \frac{f(a)+f(x_1)}{2} - \frac{(a+x_1)(f(b)-f(a))}{2(b-a)},$$

其中 $x_1$ 由下面的等式确定:

$$f'(x_1) = \frac{f(b)-f(a)}{b-a}.$$

□

♣ 当 $n \geq 2$ 时, 求最佳一致逼近多项式是非常困难的.

### 3.4.4 $n$ 次多项式的 $n-1$ 次最佳一致逼近多项式

如果 $f(x)$ 是一个 $n$ 次多项式, 则我们可以利用首项系数为 1 的 Chebyshev 多项式与零偏差最小的性质 (见定理 3.14), 构造其 $n-1$ 次的最佳一致逼近多项式.

**定理 3.22** 设 $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, 其中 $a_n \neq 0$, 则

$$p_{n-1}^*(x) = f(x) - a_n \tilde{T}_n(x)$$

是 $f(x)$ 在 $[-1,1]$ 上的 $n-1$ 次最佳一致逼近多项式.

证明. 留作练习. □

**例 3.11** 设 $f(x) = 2x^3 + x^2 + 2x - 1$, 求 $f(x)$ 在 $[-1,1]$ 上的 2 次最佳一致逼近多项式.

证明. 由例 3.22 可知, $f(x)$ 在 $[-1,1]$ 上的 2 次最佳一致逼近多项式为

$$p_2^*(x) = f(x) - 2\tilde{T}_3(x) = x^2 + \frac{7}{2}x - 1.$$

□

思考 思考: 如何计算首项系数非零的 $n$ 次多项式 $f(x)$ 在 $[a,b]$ 上的 $n-1$ 次最佳一致逼近多项式?

### 3.4.5 Chebyshev 级数与近似最佳一致逼近

对于任意一个 $f(x) \in C[a,b]$, 计算其最佳一致逼近多项式是非常困难的, 目前还没有一个可以经过有限步计算出 $f(x)$ 的 $n$ 次最佳一致逼近多项式的方法.

♣ 关于 $n$ 次最佳一致逼近多项式的构造, 可以采用 Remes 算法, 但这是一种迭代近似算法, 而且比较复杂, 运算量也较大, 详细介绍可以参考相关文献. 在实际应用中, 人们往往更倾向于寻找近似的最佳一致逼近多项式.

Chebyshev 展开就是一种很有效的计算近似最佳一致逼近多项式的方法. 设 $f \in C[-1,1]$, 权函数 $\rho(x) = \dfrac{1}{\sqrt{1-x^2}}$, 在 $f(x)$ 的广义 Fourier 级数 (3.10) 中取 $\varphi_k = T_k(x)$, 则可得

$$\frac{a_0^*}{2} + \sum_{k=1}^{\infty} a_k^* T_k(x),$$

这就是 $f(x)$ 在 $[-1,1]$ 上的 Chebyshev 级数, 其中

$$a_k^* = \frac{2}{\pi}\int_{-1}^{1}\frac{T_k(x)f(x)}{\sqrt{1-x^2}}\,\mathrm{d}x = \frac{2}{\pi}\int_0^{\pi} f(\cos\theta)\cos k\theta\,\mathrm{d}\theta, \quad k=0,1,2,\ldots.$$

注意: 由于 $(T_0,T_0)=\pi$, 因此这里的 $a_0^* = 2\dfrac{(T_0,f)}{(T_0,T_0)}$.

**定理 3.23** 若 $f''(x)$ 在 $[-1,1]$ 上分段连续, 则 Chebyshev 级数一致收敛, 即

$$f(x) = \frac{a_0^*}{2} + \sum_{k=1}^{\infty} a_k^* T_k(x).$$

记部分和

$$C_n^*(x) = \frac{a_0^*}{2} + \sum_{k=1}^{n} a_k^* T_k(x),$$

则余项为 (级数一致收敛, 余项收敛到 0)

$$f(x) - C_n^*(x) = \sum_{k=n+1}^{\infty} a_k^* T_k(x) \approx a_{n+1}^* T_{n+1}(x).$$

由于 $\|\tilde{T}_{n+1}(x)\|_\infty$ 在 $\hat{H}_{n+1}$ 中最小, 故 $C_n^*(x)$ 可看作是 $f(x)$ 在 $[-1,1]$ 上的近似最佳一致逼近多项式.

**例 3.12** 求 $f(x)=\mathrm{e}^x$ 在 $[-1,1]$ 上的 Chebyshev 级数部分和 $C_3^*(x)$. (板书)

## 3.5 曲线拟合与最小二乘

### 3.5.1 曲线拟合介绍

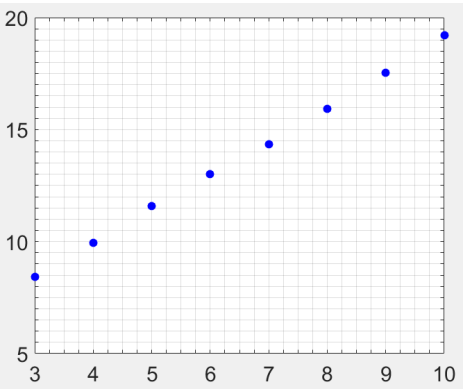
曲线拟合 (curve fitting) 是指选择适当的曲线来拟合通过观测或实验所获得的数据. 科学和工程中遇到的很多问题, 往往只能通过诸如采样、实验等方法获得若干离散的数据. 根据这些数据, 如果能够找到一个连续的函数 ( 即曲线 ) 或者更加密集的离散方程, 使得实验数据与方程的曲线能够在最大程度上近似吻合, 就可以根据曲线方程对数据进行理论分析和数值预测, 对某些不具备测量条件的位置的结果进行估算.

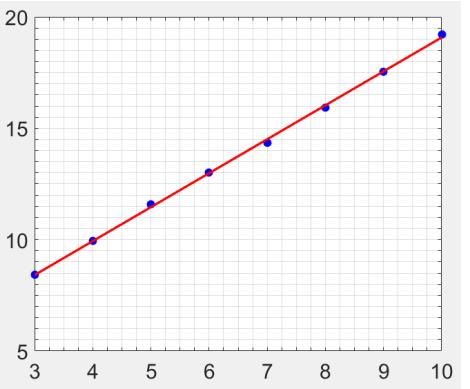
我们首先看一个简单的线性数据拟合问题.

**例 3.13** 回想一下中学物理课的 “速度与加速度” 实验: 假设某物体正在做加速运动, 加速度未知, 实验人员从时间 $t_0=3$ 秒时刻开始, 每隔 1 秒时间对这个物体进行测速, 得到一组速度和时间的离散数据 (见下表). 请根据实验数据推算该物体的加速度.

| 时间 $t$ (秒) | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|
| 速度 $v$ (米/秒) | 8.41 | 9.94 | 11.58 | 13.02 | 14.33 | 15.92 | 17.54 | 19.22 |

**解.** 实验法: 在坐标纸中画出这些点, 如下图 (左图) 所示.




可以看出, 测量结果呈现典型的线性特征. 沿着该线性特征画一条直线, 使尽量多的测量点能够位于直线上, 或与直线的偏差尽量小, 见上图 (右图). 这条直线就是我们根据测量结果拟合的速度与时间的函数关系. 最后测量出直线的斜率 $k$, 它就是被测物体的加速度. 经过测量, 我们实验测到的物体加速度值约为 1.52 米/秒 $^2$.

数学方法: 设加速度为 $a$ (米/秒 $^2$), 则速度 $v$ 与时间 $t$ 之间的关系式为

$$v = v_0 + a(t - t_0).$$

其中 $v_0 = 8.41, t_0 = 3$. 将实验数据 $(t_i, v_i)$ 依次代入可得

$$\begin{cases} 9.94 = 8.41 + a \\ 11.58 = 8.41 + 2a \\ 13.02 = 8.41 + 3a \\ 14.33 = 8.41 + 4a \\ 15.92 = 8.41 + 5a \\ 17.54 = 8.41 + 6a \\ 19.22 = 8.41 + 7a \end{cases}$$

显然, 这个方程组是无解的.

事实上, 由于实验存在误差, 上面的每个方程并不需要严格成立, 因此我们只需要求偏差尽可能地小即可. 也就是说, 使得偏差平方和尽可能地小, 即转化为下面的最小化问题

$$\min_{a \in \mathbb{R}} \sum_{i=1}^{7} |v_i - v_0 - a(t_i - t_0)|^2.$$

因此, 我们需要做的是: 怎样求解上面的最小化问题. 这就是数据拟合的最小二乘法. □

### 3.5.2 最小二乘与法方程

给定数据

| $x_i$ | $x_0$ | $x_1$ | $x_2$ | $\cdots$ | $x_m$ |
|---|---|---|---|---|---|
| $y_i$ | $y_0$ | $y_1$ | $y_2$ | $\cdots$ | $y_m$ |

在函数族 $\Phi \triangleq \text{span}\{\varphi_0(x), \varphi_1(x), \ldots, \varphi_n(x)\}$ 中寻找函数 $S^*(x)$, 使得它与这组数据的偏差平方和最小, 即

$$\sum_{i=0}^{m} |S^*(x_i) - y_i|^2 = \min_{S(x)\in\Phi} \sum_{i=0}^{m} |S(x_i) - y_i|^2.$$

这就是曲线拟合的最小二乘法. 这里的 $n$ 通常远远小于 $m$, 即 $n \ll m$.

♣ 在进行数据拟合时, 也可以使用其他标准 (拟合方法), 如极小化偏差的最大值, 即

$$\max_{0\le i\le m} |S^*(x_i) - y_i| = \min_{S(x)\in\Phi} \max_{0\le i\le m} |S(x_i) - y_i|.$$

但上述极小化问题求解很复杂.

另一种拟合方法是极小化偏差之和, 即

$$\sum_{i=0}^{m} |S^*(x_i) - y_i| = \min_{S(x)\in\Phi} \sum_{i=0}^{m} |S(x_i) - y_i|.$$

但由于目标函数不可导, 求解也很困难.

**带权最小二乘**

在某些应用中, 在各个点上的权重可能不一样, 因此我们需要带权的数据拟合问题, 即

$$\min_{S(x)\in\Phi} \sum_{i=0}^{m} \omega_i |S(x_i) - y_i|^2, \tag{3.11}$$

其中 $\omega_i$ 都是正实数, 代表在各个点处的权重.

离散数据拟合的最小二乘问题实质上可以看作是最佳平方逼近问题的离散形式, 因此, 可以将求连续函数的最佳平方逼近函数的方法直接用于求解该问题.

下面考虑问题 (3.11) 的求解. 对任意 $S(x) \in \Phi = \text{span}\{\varphi_0(x), \varphi_1(x), \ldots, \varphi_n(x)\}$, 可设

$$S(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x),$$

则原问题就转化为求下面的多元函数的最小值点:

$$I(a_0, a_1, \ldots, a_n) \triangleq \sum_{i=0}^{m} \omega_i |S(x_i) - y_i|^2 = \sum_{i=0}^{m} \omega_i \left[\sum_{j=0}^{n} a_j\varphi_j(x_i) - y_i\right]^2.$$

由于 $I(a_0, a_1, \ldots, a_n)$ 是正定的, 因此其最小值点就是其驻点. 令偏导数为零, 可得

$$0 = \frac{\partial I(a_0, a_1, \ldots, a_n)}{\partial a_k} = 2\sum_{i=0}^{m} \omega_i\varphi_k(x_i) \left[\sum_{j=0}^{n} a_j\varphi_j(x_i) - y_i\right], \quad k = 0, 1, 2, \ldots, n.$$

整理后可写为

$$\sum_{j=0}^{n} \left[\sum_{i=0}^{m} \omega_i\varphi_k(x_i)\varphi_j(x_i) - y_i\right] a_j = \sum_{i=0}^{m} \omega_i y_i\varphi_k(x_i), \quad k = 0, 1, 2, \ldots, n.$$

引入记号

$$(\varphi_j, \varphi_k) \triangleq \sum_{i=0}^{m} \omega_i\varphi_j(x_i)\varphi_k(x_i), \quad (y, \varphi_k) \triangleq \sum_{i=0}^{m} \omega_i y_i\varphi_k(x_i), \tag{3.12}$$

则上面的方程可简写为

$$\sum_{j=0}^{n}(\varphi_j,\varphi_k)a_j=(y,\varphi_k),\quad k=0,1,2,\dots,n.$$

写成矩阵形式即可得**法方程**:

$$Ga=d, \tag{3.13}$$

其中

$$G=\begin{bmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n)\\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n)\\ \vdots & \vdots & & \vdots\\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n)\end{bmatrix},\quad a=\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_n\end{bmatrix},\quad d=\begin{bmatrix}(y,\varphi_0)\\ (y,\varphi_n)\\ \vdots\\ (y,\varphi_n)\end{bmatrix}.$$

将法方程的解记为 $a_0^*,a_1^*,a_2^*,\dots,a_n^*$, 则最佳平方逼近函数为

$$S^*(x)=a_0^*\varphi_0(x)+a_1^*\varphi_1(x)+\cdots+a_n^*\varphi_n(x).$$

为了确保法方程的解存在唯一, 我们要求系数矩阵 $G$ 非奇异.

♣ 需要指出的是, 我们在 (3.12) 中引入的记号 $(\varphi_j,\varphi_k)$ 并不构成 $C[a,b]$ 或 $\Phi$ 中的内积, 因此仅仅凭借 $\varphi_0(x),\varphi_1(x),\dots,\varphi_n(x)$ 的线性无关并不能推出 $G$ 是非奇异的.

**定理 3.24** 设 $\varphi_0(x),\varphi_1(x),\dots,\varphi_n(x)\in C[a,b]$ 线性无关. 如果 $\varphi_0(x),\varphi_1(x),\dots,\varphi_n(x)$ 的任意 (非零) 线性组合在点集 $\{x_0,x_1,\dots,x_m\}$ 上至多只有 $n$ 个不同的零点, 则 $G$ 非奇异, 此时法方程 (3.13) 存在唯一解.

♣ 上述定理中的条件称为 Haar 条件.

### 3.5.3 多项式拟合

在数据拟合时, 如果取 $\Phi=H_n\triangleq\operatorname{span}\{1,x,x^2,\dots,x^n\}$, 则相应的法方程 (3.13) 为

$$\begin{bmatrix}\sum\limits_{i=0}^{m}\omega_i & \sum\limits_{i=0}^{m}\omega_i x_i & \cdots & \sum\limits_{i=0}^{m}\omega_i x_i^n\\ \sum\limits_{i=0}^{m}\omega_i x_i & \sum\limits_{i=0}^{m}\omega_i x_i^2 & \cdots & \sum\limits_{i=0}^{m}\omega_i x_i^{n+1}\\ \vdots & \vdots & & \vdots\\ \sum\limits_{i=0}^{m}\omega_i x_i^n & \sum\limits_{i=0}^{m}\omega_i x_i^{n+1} & \cdots & \sum\limits_{i=0}^{m}\omega_i x_i^{2n}\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_n\end{bmatrix}=\begin{bmatrix}\sum\limits_{i=0}^{m}\omega_i y_i\\ \sum\limits_{i=0}^{m}\omega_i x_i y_i\\ \cdots\\ \sum\limits_{i=0}^{m}\omega_i x_i^n y_i\end{bmatrix}.$$

此时,

$$S^*(x)=a_0^*+a_1^*x+\cdots+a_n^*x^n$$

即为 $f(x)$ 的 $n$ 次最小二乘拟合多项式.

### 3.5.4 非线性最小二乘

*To be continued ...*

## 3.6 有理逼近

略

## 3.7 三角多项式逼近与快速 Fourier 变换

略

## 3.8 课后练习

练习 3.1 略

练习 3.2 略

练习 3.3 证明函数 $1, x, x^2, \dots, x^n$ 线性无关.

练习 3.4 计算 $f(x)$ 关于 $C[0,1]$ 的 $\|f\|_\infty$, $\|f\|_1$ 和 $\|f\|_2$.

(1) $f(x) = (x-1)^3$;

(2) $f(x) = \left|x - \dfrac{1}{2}\right|$;

(3) $f(x) = x^m(1-x)^n$, $m, n$ 均为正整数.

练习 3.5 证明 $\|f-g\| \geq \|f\| - \|g\|$.

练习 3.6 设 $f(x), g(x) \in C^1[a,b]$, 定义

(1) $(f,g) = \displaystyle\int_a^b f'(x)g'(x)\,\mathrm{d}x$;

(2) $(f,g) = \displaystyle\int_a^b f'(x)g'(x)\,\mathrm{d}x + f(a)g(a)$;

问它们是否构成内积.

练习 3.7 令 $T_n^*(x) = T_n(2x-1)$, $x \in [0,1]$, 试证 $\{T_n^*\}$ 是在 $[0,1]$ 上的带权 $\rho(x) = \dfrac{1}{\sqrt{x-x^2}}$ 的正交多项式, 并求 $T_0^*(x), T_1^*(x), T_2^*(x), T_3^*(x)$.

练习 3.8 设权函数 $\rho(x) = 1+x^2$, 试求区间 $[-1,1]$ 上的首项系数为 1 的正交多项式 $\varphi_n(x)$, $n = 0,1,2,3$.

练习 3.9 试证明由 (3.6) 给出的第二类 Chebyshev 多项式是 $[-1,1]$ 上带权

$$\rho(x) = \sqrt{1-x^2}$$

的正交多项式.

练习 3.10 设 $n \geq 1$, 证明: 对每一个 Chebyshev 多项式 $T_n(x)$, 有

$$\int_{-1}^{1} \frac{(T_n(x))^2}{\sqrt{1-x^2}}\,\mathrm{d}x = \frac{\pi}{2}.$$

练习 3.11 用 $T_3(x)$ 的零点做插值节点, 求 $f(x) = \mathrm{e}^x$ 在区间 $[-1,1]$ 上的二次插值多项式, 并估计最大误差界.

练习 3.12 设 $f(x) = x^2 + 3x + 2$, $x \in [0,1]$, 试求 $f(x)$ 在 $[0,1]$ 上关于 $\rho(x) = 1$, $\Phi = \mathrm{span}\{1, x\}$ 的最佳平方逼近多项式. 若取 $\Phi = \mathrm{span}\{1, x, x^2\}$, 则最佳平方逼近多项式是什么?

练习 3.13 求 $f(x) = x^3$ 在 $[-1,1]$ 上关于 $\rho(x) = 1$ 的最佳平方逼近二次多项式.

练习 3.14 求 $f(x)$ 在指定区间上对于 $\Phi = \mathrm{span}\{1, x\}$ 的最佳平方逼近多项式:

(1) $f(x) = \dfrac{1}{x}$, $x \in [1,3]$;

(2) $f(x) = \mathrm{e}^x$, $x \in [0,1]$;

(3) $f(x) = \cos \pi x$, $x \in [0,1]$;

(4) $f(x) = \ln x$, $x \in [1,2]$.

**练习 3.15** 设 $f(x)=\sin\dfrac{\pi x}{2}$, 利用 Legendre 多项式求 $f(x)$ 在 $[-1,1]$ 上的三次最佳平方逼近多项式.

### 附加题

**练习 3.16** 给出 Cauchy-Schwarz 不等式 (1.1) 中等号成立的充要条件, 并证明.

**练习 3.17** 设 $X$ 是数域 $\mathbb{C}$ 上的内积空间, 证明:

$$\|x\|=(x,x)^{\frac{1}{2}}$$

是 $X$ 上的范数.

**练习 3.18** 设 $\omega_1,\omega_2,\dots,\omega_n$ 为给定的正实数, 对任意 $x,y\in\mathbb{R}^n$, 定义

$$(x,y)_\omega=\sum_{i=1}^{n}\omega_i x_i y_i,$$

证明: $(x,y)_\omega$ 是 $\mathbb{R}^n$ 上的内积.

**练习 3.19** 证明例 3.22 中的结论, 即:

$p_{n-1}^*(x)=f(x)-a_n\tilde{T}_n(x)$ 是 $n$ 次多项式 $f(x)=a_nx^n+\cdots+a_1x+a_0$ 在 $[-1,1]$ 上的 $n-1$ 次最佳一致逼近多项式, 其中 $a_n\neq 0$.

# 第四讲 数值积分与数值微分

在微积分中, 计算定积分的主要方法之一是 Newton-Leibnitz 公式. 但在很多情况下, 被积函数的原函数是很难求出的, 或者原函数很复杂, 或者无法用初等函数表示. 甚至在某些实际应用中, 被积函数的表达式是不知道的, 只是通过实验或测量等手段给出了某些离散点上的值. 在这些情况下, 我们就需要考虑数值积分, 即通过近似方法来计算定积分的近似值.

## 4.1 数值积分基本概念

### 4.1.1 为什么要数值积分

考虑定积分

$$I = \int_a^b f(x)\,\mathrm{d}x. \tag{4.1}$$

在微积分中, 我们可以使用 Newton-Leibnitz 公式

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a),$$

其中 $F(x)$ 是被积函数 $f(x)$ 的一个原函数. 但是

- 在很多情况下, 被积函数的原函数很难求出, 或者原函数很复杂, 如 $f(x) = \dfrac{1}{1+x^6}$ 的一个原函数为

$$F(x) = \frac{1}{3}\arctan x + \frac{1}{6}\arctan\left(x - \frac{1}{x}\right) + \frac{1}{4\sqrt{3}}\ln\frac{x^2 + x\sqrt{3} + 1}{x^2 - x\sqrt{x} + 1} + C.$$

- 原函数无法用初等函数表示, 如

$$f(x) = \frac{\sin x}{x}, \quad f(x) = \mathrm{e}^{-x^2}, \quad f(x) = \sqrt{1 + k^2\sin^2 x}.$$

- 在某些实际应用中, 被积函数 $f(x)$ 的表达式是未知的, 只是通过实验或测量等手段给出了某些离散点上的值.

在这些情况下, 我们就需要考虑通过近似方法来计算定积分的近似值, 即数值积分.

### 4.1.2 数值积分主要研究的问题

数值积分的基本思想是用函数值的线性组合来近似定积分, 有时也会用到导数值. 数值积分主要考虑以下问题:

(1) 求积公式的构造;

(2) 精确程度的衡量;

(3) 余项的估计.

### 4.1.3 机械求积公式

设 $f(x)\in C[a,b]$, 取节点 $a=x_0<x_1<x_2<\cdots<x_n<x_{n+1}=b$, 根据定积分的定义, 有

$$\int_a^b f(x)\,\mathrm{d}x=\lim_{h\to 0}\sum_{i=0}^{n}h_i f(\xi_i),\quad \xi_i\in[x_i,x_{i+1}],$$

其中 $h_i=x_{i+1}-x_i$, $h=\max\limits_i\{h_i\}$. 当 $h$ 充分小, $n$ 充分大时, 我们就有下面的近似公式

$$\int_a^b f(x)\,\mathrm{d}x\approx\sum_{i=0}^{n}h_i f(\xi_i). \tag{4.2}$$

这就是目前常用的求积公式. 为了方便起见, 我们将上述公式改写为 (将记号 $h_i$ 和 $\xi_i$ 更换为 $A_i$ 和 $x_i$)

$$\int_a^b f(x)\,\mathrm{d}x\approx\sum_{i=0}^{n}A_i f(x_i)\triangleq I_n(f) \tag{4.3}$$

这里 $x_i$ 称为求积节点, 满足 $a\le x_0<x_1<\cdots<x_n\le b$, 系数 $A_i$ 称为求积系数, 与函数 $f(x)$ 无关. 该求积公式就称为机械求积公式.

♣ 机械求积公式只包含函数值, 但求积公式并且局限于机械求积公式, 有些求积公式可能会包含其它信息, 如导数值等.

### 4.1.4 代数精度

根据 Weierstrass 逼近定理 3.1 可知, 任意一个连续函数都可以通过多项式来一致逼近, 因此, 如果一个求积公式能对比较高阶的多项式精确成立, 那么我们就认为该求积公式具有较高的精度. 基于这样的想法, 我们给出下面的代数精度概念.

**定义 4.1** 如果一个求积公式对所有次数不超过 $m$ 的多项式精度成立, 但对 $m+1$ 次多项式不精确成立, 则称该求积公式具有 $m$ 次代数精度.

由定义可知, 一个求积公式具有 $m$ 次代数精度当且仅当求积公式

(1) 对 $f(x)=1,x,x^2,\ldots,x^m$ 精确成立;

(2) 对 $f(x)=x^{m+1}$ 不精确成立.

这给出了计算一个求积公式的代数精度的方法.

**例 4.1** 试确定系数 $A_i$, 使得下面的求积公式具有尽可能高的代数精度, 并求出此求积公式的代数精度.

$$\int_{-1}^{1}f(x)\,\mathrm{d}x\approx A_0f(-1)+A_1f(0)+A_2f(1).$$

**解.** 分别取 $f(x)=1,x,x^2$, 令求积公式精确成立, 可得

$$\begin{cases}A_0+A_1+A_2=b-a\\ -A_0+A_2=0\\ A_0+A_2=\dfrac{2}{3}\end{cases}$$

求解该方程组, 可得 $A_0 = \frac{1}{3}$, $A_1 = \frac{4}{3}$, $A_2 = \frac{1}{3}$. 因此求积公式为

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \frac{1}{3}f(-1) + \frac{4}{3}f(0) + \frac{1}{3}f(1).$$

将 $f(x) = x^3$ 代入可得, 公式左边为

□

♣ 一般来说, 总是能确定 $A_i$ 的值, 使得求积公式 (4.3) 至少具有 $n$ 次代数精度.

**例 4.2 (非机械求积公式)** 试确定下面求积公式中的系数，使其具有尽可能高的代数精度:

$$\int_{0}^{1} f(x)\,\mathrm{d}x \approx A_0 f(0) + A_1 f(1) + B_0 f'(0).$$

**解.** (板书) □

**引理 4.1** *设机械求积公式 (4.3) 具有 $m(\geq 0)$ 次代数精度, 则有*

$$A_0 + A_1 + \cdots A_n = b - a. \tag{4.4}$$

**证明.** 将 $f(x) = 1$ 代入求积公式 (4.3), 令等式精确成立即可. (事实上, 该性质也可以从 (4.2) 中得出) □

### 4.1.5 收敛性与稳定性

**定义 4.2** *设求积公式的余项为 $R[f]$, 若*

$$\lim_{h\to 0} R[f] = 0,$$

*则称求积公式是收敛的, 其中 $h = \max\limits_{0\leq i\leq n-1}\{x_{i+1} - x_i\}$.*

在利用机械求积公式计算定积分时, 需要计算函数值. 由于存在一定的舍入误差, 因此最后的结果也会带有一定的误差. 求积公式的稳定性就是指这些舍入误差对计算结果的影响.

**定义 4.3** *考虑机械求积公式 (4.3). 设 $\tilde{f}_k$ 是计算 $f(x_k)$ 时得到的近似值. 如果对任给的 $\varepsilon > 0$, 都存在 $\delta > 0$, 使得当 $|f(x_k) - \tilde{f}_k| < \delta$ 时, 有*

$$\left|\sum_{k=0}^{n} A_k f(x_k) - \sum_{k=0}^{n} A_k \tilde{f}_k\right| < \varepsilon,$$

*则称求积公式 (4.3) 是稳定的.*

下面给出一个判别机械求积公式稳定性的充分条件.

**定理 4.1** *若机械求积公式 (4.3) 中的求积系数 $A_i$ 都是正数, 则求积公式是稳定的.*

**证明.** (板书) □

### 4.1.6 插值型求积公式

一个构造求积公式的常用方法就是使用插值多项式. 设 $L_n(x)$ 是 $f(x)$ 关于节点 $a \le x_0 < x_1 < \cdots < x_n \le b$ 的 $n$ 次插值多项式, 则

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x \approx \int_a^b L_n(x)\,\mathrm{d}x &= \int_a^b \sum_{k=0}^n l_k(x) f(x_k)\,\mathrm{d}x \\ &= \sum_{k=0}^n \left(\int_a^b l_k(x)\,\mathrm{d}x\right) f(x_k) \triangleq \sum_{k=0}^n A_k f(x_k). \end{aligned} \tag{4.5}$$

这就是插值型求积公式, 其中 $l_k(x)$ 是 $n$ 次 Lagrange 基函数, $A_k = \int_a^b l_k(x)\,\mathrm{d}x$. 由插值余项公式可知, 插值型求积公式 (4.5) 的余项为

$$R[f] = \int_a^b f(x) - L_n(x)\,\mathrm{d}x = \int_a^b R_n(x)\,\mathrm{d}x = \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x)\,\mathrm{d}x, \tag{4.6}$$

其中

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

♣ 由于 $\xi_x$ 是 $x$ 未知函数, 上面的余项是无法计算的, 但有时可以用来估计误差, 即

$$\big|\,R[f]\,\big| = \left|\int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x)\,\mathrm{d}x\right| \le \frac{M_{n+1}}{(n+1)!}\int_a^b |\omega_{n+1}(x)|\,\mathrm{d}x,$$

其中 $M_{n+1} = \max\limits_{a\le x\le b} |f^{(n+1)(x)}|$.

**引理 4.2** 插值型求积公式 (4.5) 至少具有 $n$ 次代数精度.

证明. 当 $f(x) = 1, x, x^2, \ldots, x^n$ 时, $R_n(x) = 0$. 故求积公式对 $f(x) = 1, x, x^2, \ldots, x^n$ 精确成立. □

事实上, 我们有下面的性质.

**定理 4.2** 机械求积公式 (4.3) 至少具有 $n$ 次代数精度的充要条件是该公式是插值型的.

证明. (板书) □

♣ 当机械求积公式具有尽可能高的代数精度时, 它总是插值型的.

## 4.2 Newton-Cotes 公式

**定义 4.4** 如果插值型求积公式 (4.5) 中的节点为等距节点, 即

$$x_k = a + kh, \quad h = \frac{b-a}{n},$$

则该求积公式就称为 Newton-Cotes 公式, 记为

$$I_n(f) = (b-a)\sum_{k=0}^n C_k^{(n)} f(x_k), \tag{4.7}$$

其中 $C_k^{(n)}$ 称为 Cotes 系数, 其值为

$$C_k^{(n)} = \frac{1}{b-a}\int_a^b l_k(x)\,\mathrm{d}x = \frac{h}{b-a}\int_0^n \prod_{i=0,i\neq k}^{n}\frac{t-i}{k-i}\,\mathrm{d}t = \frac{(-1)^{n-k}}{n\,k!(n-k)!}\int_0^n \prod_{i=0,i\neq k}(t-i)\,\mathrm{d}t.$$

Cotes 系数具有下面的性质.

(1) $\sum_{k=0}^{n} C_k^{(n)} = 1$;

(2) $C_k^{(n)} = C_{n-k}^{(n)}, \quad k = 0, 1, 2, \ldots, n.$

### 4.2.1 常用的低次 Newton-Cotes 公式

下面给出几个常用的低次 Newton-Cotes 公式:

- 当 $n = 1$ 时, 可得 $C_0^{(1)} = C_1^{(1)} = \frac{1}{2}$, 此时的 Newton-Cotes 公式为

$$I_1(f) = \frac{b-a}{2}\left(f(a) + f(b)\right). \tag{4.8}$$

这就是梯形公式.

- 当 $n = 2$ 时, 可得 $C_0^{(2)} = \frac{1}{6}, C_1^{(2)} = \frac{4}{6}, C_2^{(2)} = \frac{1}{6}$, 此时的 Newton-Cotes 公式为

$$I_2(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right). \tag{4.9}$$

这就是 Simpson 公式 或抛物线公式.

- 当 $n = 3$ 时, 可得 $C_0^{(3)} = \frac{1}{8}, C_1^{(3)} = \frac{3}{8}, C_2^{(3)} = \frac{3}{8}, C_3^{(3)} = \frac{1}{8}$, 此时的 Newton-Cotes 公式为

$$I_3(f) = \frac{b-a}{8}\big(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\big). \tag{4.10}$$

该公式称为 Simpson 3/8 公式 或 Boole 公式 或 Milne 公式.

- 当 $n = 4$ 时, 可得 $C_0^{(4)} = \frac{7}{90}, C_1^{(4)} = \frac{32}{90}, C_2^{(4)} = \frac{12}{90}, C_3^{(4)} = \frac{32}{90}, C_4^{(4)} = \frac{7}{90}$, 此时的 Newton-Cotes 公式为

$$I_4(f) = \frac{b-a}{90}\big(7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)\big). \tag{4.11}$$

这公式就是 Cotes 公式.

♣ 当 $n > 7$ 时, Cotes 系数中会出现负数, 会导致算法的不稳定, 因此我们不考虑 $n > 7$ 时的 Newton-Cotes 公式.

**定理 4.3** 当 $n$ 是奇数时, Newton-Cotes 公式至少具有 $n$ 次代数精度. 当 $n$ 是偶数时, Newton-Cotes 公式至少具有 $n+1$ 次代数精度.

证明. 由于 Newton-Cotes 公式是插值型的, 因此它至少具有 $n$ 次代数精度.

当 $n$ 是偶数时, 将 $f(x) = x^{n+1}$ 代入余项公式 (4.6) 可得

$$R[f] = \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x)\,\mathrm{d}x = \int_a^b \omega_{n+1}(x)\,\mathrm{d}x$$

做变量代换 $x = a + th$. 由于 $x_k = a + kh$, 所以

$$R[f] = \int_a^b \omega_{n+1}(x)\,\mathrm{d}x = h^{n+2}\int_0^n \prod_{k=0}^{n}(t-k)\,\mathrm{d}t$$

由于 $n$ 是偶数, 再做变量代换 $t = n - s$, 可得

$$R[f] = h^{n+2}\int_n^0 (-1)^{n+2}\prod_{k=0}^{n}(s-(n-k))\,\mathrm{d}s = (-1)^{n+3}\int_0^n \prod_{i=0}^{n}(s-i))\,\mathrm{d}s = -R[f].$$

所以 $R[f] = 0$, 故 Newton-Cotes 公式对 $f(x) = x^{n+1}$ 精确成立, 即具有 $n+1$ 次代数精度. □

### 4.2.2 求积公式余项的推导

首先给出几个引理.

**引理 4.3** 设基于节点 $x_0, x_1, x_2, \ldots, x_n$ 的插值型求积公式 (4.5) 具有 $m$ $(m \geq n)$ 次代数精度, 则对任意 $f(x) \in C[a,b]$, 有

$$\sum_{k=0}^{n} A_k f(x_k) = \int_a^b p_m(x)\,\mathrm{d}x,$$

其中 $p_m(x)$ 是 $f(x)$ 关于节点 $x_0, x_1, x_2, \ldots, x_n$ 的 $m$ 次插值多项式, 即 $p_m(x)$ 满足

$$p_m(x_k) = f(x_k), \quad k = 0, 1, 2, \ldots, n.$$

(注: 当 $m > n$ 时, 其它 $m - n$ 个插值条件可以任取, 如要求导数值相等).

证明. 由于求积公式具有 $m$ 次代数精度, 即对 $m$ 次多项式精确成立, 所以

$$\sum_{k=0}^{n} A_k f(x_k) = \sum_{k=0}^{n} A_k p_m(x_k) = \int_a^b p_m(x)\,\mathrm{d}x.$$

□

上述结论也可以推广到包含导数的求积公式.

**引理 4.4** 设基于节点 $x_0, x_1, x_2, \ldots, x_n$ 的求积公式为

$$I_n(f) = \sum_{k=0}^{n} A_k f(x_k) + \sum_{j\in\mathbb{Z}_1} B_j f'(x_j). \tag{4.12}$$

其中 $\mathbb{Z}_1 \subset \{0, 1, 2, \ldots, n\}$, 即可以只包含部分节点上的导数值. 若该求积公式具有 $m$ $(m \geq n)$ 次代数精度, 则对任意 $f(x) \in C[a,b]$, 有

$$I_n(f) = \int_a^b p_m(x)\,\mathrm{d}x,$$

其中 $p_m(x)$ 是 $f(x)$ 关于节点 $x_0, x_1, x_2, \ldots, x_n$ 的 $m$ 次 Hermite 插值多项式, 即 $p_m(x)$ 满足

$$\begin{cases} p_m(x_k) = f(x_k), & k = 0, 1, 2, \ldots, n \\ p_m'(x_j) = f'(x_j), & j \in \mathbb{Z}_1. \end{cases}$$

(注: 如果插值条件个数小于 $m+1$, 则其它插值条件可以任取)

♣ 假设 $\mathbb{Z}_1$ 中的元素个数为 $r$ ($r \leq n+1$), 则一般总是有 $m \geq n+r$.

♣ 如果求积公式中包含二阶以上的导数 (如后面会提到的修正 Simpson 公式 (4.19), 我们仍可以得到相类似的结论.

下面是关于差商的连续性.

**引理 4.5** 设 $a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$, 函数 $f(x) \in C^1[a,b]$, 则差商

$$g(x) \triangleq f[x, x_0, x_1, \ldots, x_n]$$

关于 $x$ 在 $[a,b]$ 上连续. 这里 $g(x)$ 在节点 $x_k$ 上的值是通过重节点差商来定义的. 进一步, 若 $f(x) \in C^2[a,b]$, 则有

$$g'(x) = f[x, x, x_0, x_1, \ldots, x_n],$$

且 $g'(x)$ 也关于 $x$ 在 $[a,b]$ 上连续.

证明. 参见: E. Isaacson and H. Keller, Analysis of Numerical Methods, John Wiley and Sons, London–New York, 1966. 第 255 页. □

上述结论可以推广到重节点情形, 即允许节点 $x_0, x_1, x_2, \ldots, x_n$ 有重合的.

**引理 4.6** 设 $a = x_0 \leq x_1 \leq x_2 \leq \cdots \leq x_{n-1} \leq x_n = b$, 函数 $f(x) \in C^{n+1}[a,b]$, 则差商

$$g(x) \triangleq f[x, x_0, x_1, \ldots, x_n]$$

关于 $x$ 在 $[a,b]$ 上连续. 进一步, 若 $f(x) \in C^{n+2}[a,b]$, 则有

$$g'(x) = f[x, x, x_0, x_1, \ldots, x_n],$$

且 $g'(x)$ 也关于 $x$ 在 $[a,b]$ 上连续.

### 梯形公式的余项

**定理 4.4** 设 $f(x) \in C^2[a,b]$, 则梯形求积公式的余项为

$$R[f] = -\frac{(b-a)^3}{12} f''(\eta), \quad \eta \in (a,b),$$

所以, 带余项的梯形公式可写为

$$\int_a^b f(x)\,\mathrm{d}x = \frac{b-a}{2}\Big(f(a)+f(b)\Big) - \frac{(b-a)^3}{12} f''(\eta), \quad \eta \in (a,b). \tag{4.13}$$

证明. 设 $p_1(x)$ 是 $f(x)$ 关于节点 $x_0 = a, x_1 = b$ 的一次插值多项式. 由 Newton 插值的余项公式可知

$$f(x) - p_1(x) = f[x, x_0, x_1]\omega_2(x),$$

其中 $\omega_2(x) = (x-x_0)(x-x_1)$. 由于梯形公式的代数精度为 1, 故

$$\int_a^b p_1(x)\,\mathrm{d}x = \frac{b-a}{2}\Big(p_1(a)+p_1(b)\Big).$$

所以

$$\begin{aligned} R[f] &= \int_a^b f(x)\,\mathrm{d}x - \frac{b-a}{2}\Big(f(a)+f(b)\Big) \\ &= \int_a^b f(x)\,\mathrm{d}x - \frac{b-a}{2}\Big(p_1(a)+p_1(b)\Big) \qquad (\text{插值条件}: p_1(a)=f(a), p_1(b)=f(b)) \\ &= \int_a^b \Big(f(x)-p_1(x)\Big)\,\mathrm{d}x \\ &= \int_a^b f[x,x_0,x_1]\omega_2(x)\,\mathrm{d}x. \end{aligned}$$

由于 $f[x,x_0,x_1]$ 关于 $x$ 在 $[a,b]$ 上连续, $\omega_2(x)$ 在 $[a,b]$ 内不变号, 所以由积分中值定理可知, 存在 $z\in(a,b)$, 使得

$$R[f] = f[z,x_0,x_1]\int_a^b \omega_2(x)\,\mathrm{d}x = -\frac{(b-a)^3}{6}f[z,x_0,x_1].$$

由差商与导数之间的关系可知, 存在 $\eta\in(a,b)$, 使得 $f[z,x_0,x_1]=\dfrac{f''(\eta)}{2!}$. 这里 $\eta$ 与 $z$ 有关. 所以

$$R[f] = -\frac{(b-a)^3}{12}f''(\eta), \quad \eta\in(a,b).$$

□

♣ 如果使用 Lagrange 插值余项公式, 则可得

$$R[f] = \int_a^b \frac{1}{2!}f''(\xi_x)(x-a)(x-b)\,\mathrm{d}x, \quad \xi_x\in(a,b).$$

易知 $(x-a)(x-b)$ 在 $[a,b]$ 内不变号, 如果 $f''(\xi_x)$ 关于 $x$ 连续, 则由积分中值定理可知, 存在 $\eta\in(a,b)$ 使得

$$R[f] = \frac{1}{2!}f''(\eta)\int_a^b (x-a)(x-b)\,\mathrm{d}x = -\frac{(b-a)^3}{12}f''(\eta).$$

需要指出的是, 这里要证明 $f''(\xi_x)$ 关于 $x$ 是连续的. 事实上, 由 Lagrange 插值余项公式可知

$$f(x)-p_1(x) = \frac{1}{2!}f''(\xi_x)(x-a)(x-b),$$

即

$$f''(\xi_x) = \frac{2!(f(x)-p_1(x))}{(x-a)(x-b)} \triangleq g(x).$$

如果 $f\in C^2[a,b]$, 则上式右端 (即 $g(x)$) 显然在除插值节点以外的所有点都连续. 应用 L'Hôpital 法则, 我们可以求得 $g(x)$ 在插值节点处的极限 (在两端点处为右极限或左极限). 而根据余项公式, 在插值节点处, $f''(\xi_x)$ 可以取任意值. 因此, 我们可以将 $g(x)$ 的极限设为 $f''(\xi_x)$ 在节点处的值, 这样 $f''(\xi_x)$ 就在整个区间上连续. 以上结论可推广到一般情形, 即:

设 $f\in C^{n+1}[a,b]$, 可得 $n$ 次多项式插值余项

$$f(x)-p_n(x) = \frac{1}{(n+1)!}f^{(n+1)}(\xi_x)(x-x_0)(x-x_1)\cdots(x-x_n), \quad \xi_x\in[a,b],$$

通过定义 $f^{(n+1)}(\xi_x)$ 在插值节点处的值, 可使得 $f^{(n+1)}(\xi_x)$ 关于 $x$ 是连续的.

**Simpson 公式的余项**

**定理 4.5** 设 $f(x)\in C^4[a,b]$, 则 Simpson 求积公式的余项为

$$R[f]=-\frac{(b-a)^5}{2880}f^{(4)}(\eta),\quad \eta\in(a,b),$$

所以, 带余项的 Simpson 公式可写为

$$\int_a^b f(x)\,\mathrm{d}x=\frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right)-\frac{(b-a)^5}{2880}f^{(4)}(\eta),\quad \eta\in(a,b). \tag{4.14}$$

**证明.** 基本思想与梯形求积公式余项类似, 但具体做法有所不同.

由于 Simpson 公式具有 3 次代数精度, 构造 $f(x)$ 关于点 $x_0=a$, $x_1=\frac{1}{2}(a+b)$, $x_2=b$ 的三点三次 Hermite 插值多项式 $H_3(x)$, 满足

$$H_3(x_k)=f(x_k),\ k=0,1,2,\quad H_3'(x_1)=f'(x_1).$$

于是有

$$\int_a^b H_3(x)\,\mathrm{d}x=\frac{b-a}{6}\Big(H_3(x_0)+4H_3(x_1)+H_3(x_2)\Big), \tag{4.15}$$

且插值余项为

$$R(x)=\frac{f^{(4)}(\xi_x)}{4!}(x-x_0)(x-x_1)^2(x-x_2).$$

所以

$$\begin{aligned}
R[f]&=\int_a^b f(x)\,\mathrm{d}x-\frac{b-a}{6}\left(f(x_0)+4f(x_1)+f(x_2)\right)\\
&=\int_a^b f(x)\,\mathrm{d}x-\frac{b-a}{6}\left(H_3(x_0)+4H_3(x_1)+H_3(x_2)\right)\qquad(\text{插值条件})\\
&=\int_a^b f(x)\,\mathrm{d}x-\int_a^b H_3(x)\,\mathrm{d}x\qquad(\text{由}(4.15))\\
&=\int_a^b\Big(f(x)-H_3(x)\Big)\,\mathrm{d}x\\
&=\int_a^b\frac{f^{(4)}(\xi_x)}{4!}(x-x_0)(x-x_1)^2(x-x_2)\,\mathrm{d}x.
\end{aligned}$$

由于 $f^{(4)}(\xi_x)$ 是 $x$ 的连续函数, 且 $(x-x_0)(x-x_1)^2(x-x_2)$ 在 $[a,b]$ 内不变号, 所以由积分中值定理可知, 存在 $\eta\in(a,b)$, 使得

$$R[f]=\int_a^b\frac{f^{(4)}(\xi_x)}{4!}(x-x_0)(x-x_1)^2(x-x_2)\,\mathrm{d}x=\frac{f^{(4)}(\eta)}{4!}\int_a^b(x-x_0)(x-x_1)^2(x-x_2)\,\mathrm{d}x.$$

将 $x_0=a$, $x_1=\frac{1}{2}(a+b)$, $x_2=b$ 代入, 可求得

$$\int_a^b(x-x_0)(x-x_1)^2(x-x_2)\,\mathrm{d}x=-\frac{(b-a)^5}{120}.$$

因此, Simpson 公式的余项为

$$R[f]=-\frac{f^{(4)}(\eta)}{4!}\cdot\frac{(b-a)^5}{120}=-\frac{(b-a)^5}{2880}f^{(4)}(\eta),\quad \eta\in(a,b).$$

□

♣ 事实上, 我们可以将 $H_3(x)$ 写成 Newton 插值形式, 即

$$H_3(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_1](x - x_0)(x - x_1) + f[x_0, x_1, x_1, x_2](x - x_0)(x - x_1)^2.$$

这可以看作是重节点 Newton 插值. 因此, 插值余项可表示为

$$R_3(x) = f[x, x_0, x_1, x_1, x_2](x - x_0)(x - x_1)^2(x - x_2).$$

请读者自行验证.

♣ 计算求积公式余项小结 (三步曲):

(1) 计算求积公式的代数精度, 设为 $m$;

(2) 构造 $m$ 次插值多项式, 写出插值条件和插值余项 (除导数部分外, 要确保不变号);

(3) 利用积分中值定理, 计算出求积公式的余项.

### 4.2.3 Newton-Cotes 公式余项的一般形式

**定理 4.6** 当 $n$ 是奇数时, 若 $f(x) \in C^{n+1}[a, b]$, 则 Newton-Cotes 公式的余项为

$$R[f] = \frac{h^{n+2} f^{(n+1)}(\eta)}{(n+1)!} \int_0^n t(t-1)(t-2)\cdots(t-n)\,\mathrm{d}t, \quad \eta \in (a, b).$$

当 $n$ 是偶数时, 若 $f(x) \in C^{n+2}[a, b]$, 则 Newton-Cotes 公式的余项为

$$R[f] = \frac{h^{n+3} f^{(n+2)}(\eta)}{(n+2)!} \int_0^n t^2(t-1)(t-2)\cdots(t-n)\,\mathrm{d}t, \quad \eta \in (a, b).$$

证明. 参见: J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, 3rd Edition, Springer, 2002. □

### 4.2.4 一般求积公式余项

考虑更一般的求积公式 (带导数信息)

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{k=0}^{n} A_k f(x_k) + \sum_{i_1 \in \mathbb{Z}_1} A_{i_1} f'(x_{i_1}) + \sum_{i_2 \in \mathbb{Z}_2} A_{i_2} f''(x_{i_1}) + \cdots + \sum_{i_r \in \mathbb{Z}_r} A_{i_r} f^{(r)}(x_{i_r}). \tag{4.16}$$

其中 $\mathbb{Z}_j \subset \{0, 1, 2, \ldots, n\}$, 即求积公式中可以包含全部或部分节点上的导数信息.

设求积公式 (4.16) 的代数精度为 $m$, 若 $f(x) \in C^{m+1}[a, b]$, 则其余项可表示为

$$R[f] = \int_a^b f^{(m+1)}(t) K(t)\,\mathrm{d}t,$$

其中 $K(t)$ 称为 Peano 核, 具体表达式可参见 “Introduction to Numerical Analysis” (J. Stoer and R. Bulirsch, 3rd Edition, Springer, 2002).

在大多数求积公式中, $K(t)$ 在 $[a, b]$ 内不变号, 此时, 根据积分中值定理可知, 存在 $\eta \in (a, b)$ 使得

$$R[f] = f^{(m+1)}(\eta) \int_a^b K(t)\,\mathrm{d}t. \tag{4.17}$$

需要注意的是, 余项公式 (4.17) 并不是对所有求积公式 (4.16) 都成立.

**例 4.3** 给出下面求积公式的余项

$$\int_0^1 f(x)\,\mathrm{d}x \approx \frac{2}{3}f(0)+\frac{1}{3}f(1)+\frac{1}{6}f'(0).$$

**解.** 首先求出代数精度, 然后构造插值多项式 $p(x)$, 满足

$$p(0)=f(0),\quad p(1)=f(1),\quad p'(0)=f'(0),$$

并写出插值余项 $R(x)$. 代入求积公式后, 利用积分中值定理给出求积公式的余项表达式. 具体过程请读者自行完成. □

## 4.3 复合求积公式

与分段插值的想法类似, 为了提高计算精度, 我们也可以将积分区间分割成若干小区间, 然后再在每个小区间使用低次 Newton-Cotes 求积公式.

为了简单起见, 我们通常等分积分区间. 本节主要介绍两类常用的复合求积公式: 复合梯形公式和复合 Simpson 公式.

### 4.3.1 复合梯形公式

将 $[a,b]$ 划分为 $n$ 等份, 即取节点

$$x_k=a+kh,\quad h=\frac{b-a}{n},\quad k=0,1,2,\dots,n.$$

在每个小区间 $[x_k,x_{k+1}]$ 上采用梯形公式, 可得

$$\int_{x_k}^{x_{k+1}} f(x)\,\mathrm{d}x \approx \frac{h}{2}(f(x_k)+f(x_{k+1})).$$

所以

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &= \sum_{k=0}^{n-1}\int_{x_k}^{x_{k+1}} f(x)\,\mathrm{d}x \approx \sum_{k=0}^{n-1}\frac{h}{2}(f(x_k)+f(x_{k+1}))\\ &= h\left(\frac{f(a)+f(b)}{2}+\sum_{k=1}^{n-1}f(x_k)\right).\end{aligned}$$

这就是**复合梯形公式** (Composite Trapezoidal rule), 通常记为 $T_n$, 即

$$T_n = h\left(\frac{f(a)+f(b)}{2}+\sum_{k=1}^{n-1}f(x_k)\right).$$

设 $f(x)\in C^2[a,b]$, 则在每个小区间 $[x_k,x_{k+1}]$ 上的余项为 $-\dfrac{h^3}{12}f''(\eta_k)$, 所以整体余项为

$$R_n[f]=\int_a^b f(x)\,\mathrm{d}x - T_n = -\frac{h^3}{12}\sum_{k=0}^{n-1}f''(\eta_k).$$

由于 $f''(x)$ 在 $[a,b]$ 上连续, 且

$$\min_{a\le x\le b} f''(x) \le \frac{1}{n}\sum_{k=0}^{n-1}f''(\eta_k) \le \max_{a\le x\le b} f''(x),$$

所以由介值定理可知, 存在 $\eta \in (a,b)$, 使得 $f''(\eta) = \frac{1}{n}\sum_{k=0}^{n-1} f''(\eta_k)$. 故

$$R_n[f] = -\frac{nh^3}{12} f''(\eta) = -\frac{b-a}{12} h^2 f''(\eta), \quad \eta \in (a,b).$$

由此可知, 当 $n \to \infty$ 时, $R_n[f] \to 0$, 所以复合梯形公式是收敛性的. 易知公式中的求积系数都是正的, 因此复合梯形公式是稳定的.

♣ 复合梯形公式看似精度不高, 但如果被积函数是以 $b-a$ 为周期的周期函数, 则其具有 $n$ 阶三角多项式精度 [? , page 65], 即

$$\int_a^b f(x)\,\mathrm{d}x - h\sum_{k=0}^{n-1} f(a+kh) = \begin{cases} -(b-a), & \text{若 } m \neq 0 \text{ 被 } n \text{ 整除}, \\ 0, & \text{其他}, \end{cases}$$

其中 $f(x) = \exp\left(\frac{2\pi \mathbf{i} mx}{b-a}\right)$, $\mathbf{i}$ 表示虚部单位, $h = \frac{b-a}{n}$.

### 4.3.2 复合 Simpson 公式

相类似地, 我们可以得到复合 Simpson 公式 (Composite Simpson's Rule), 通常记为 $S_n$:

$$\begin{aligned} \int_a^b f(x)\,\mathrm{d}x &\approx \frac{h}{6}\sum_{k=0}^{n-1}\left(f(x_k) + 4f(x_{k+\frac{1}{2}}) + f(x_{k+1})\right) \\ &= \frac{h}{6}\left(f(a) + f(b) + 2\sum_{k=1}^{n-1} f(x_k) + 4\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}})\right). \end{aligned}$$

设 $f(x) \in C^4[a,b]$, 则复合 Simpson 公式的余项为

$$R_n[f] = \int_a^b f(x)\,\mathrm{d}x - S_n = -\frac{b-a}{2880} h^4 f^{(4)}(\eta).$$

易知, 复合 Simpson 公式是收敛的, 也是稳定的.

**例 4.4** 已知 $f(x) = \dfrac{\sin x}{x}$ 的取值如下表, 试分别用复合梯形公式和复合 Simpson 公式计算 $\displaystyle\int_0^1 f(x)\,\mathrm{d}x$ 的近似值, 并估计误差.

| $x$ | 0 | 1/8 | 2/8 | 3/8 | 4/8 | 5/8 | 6/8 | 7/8 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| $f(x)$ | 1.0000 | 0.9974 | 0.9896 | 0.9767 | 0.9589 | 0.9362 | 0.9089 | 0.8772 | 0.8415 |

**解.** (板书) □

## 4.4 带导数的求积公式

### 4.4.1 带导数的梯形公式

带余项的复合梯形求积公式为

$$\int_a^b f(x)\,\mathrm{d}x = T_n - \frac{h^3}{12}\sum_{k=0}^{n-1} f''(\eta_k),$$

其中 $\eta_k \in [x_k, x_{k+1}]$. 根据定积分的定义, 当 $h$ 充分小时, 有

$$h\sum_{k=0}^{n-1} f''(\eta_k) \approx \int_a^b f''(x)\,\mathrm{d}x = f'(b) - f'(a).$$

于是, 我们可以得到下面的求积公式

$$\int_a^b f(x)\,\mathrm{d}x \approx T_n - \frac{h^2}{12}\Big(f'(b) - f'(a)\Big).$$

这就是带端点导数值的复合梯形公式.

当 $n = 1$ 时,

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{2}\Big(f(a) + f(b)\Big) - \frac{(b-a)^2}{12}\Big(f'(b) - f'(a)\Big). \tag{4.18}$$

这就是修正的梯形公式 (Corrected Trapezoidal Rule). 可以验证, 求积公式 (4.18) 的余项为

$$R[f] = \frac{(b-a)^5}{720} f^{(4)}(\eta), \quad \eta \in (a, b).$$

### 4.4.2 带导数的 Simpson 公式

相类似地, 我们可以给出带端点导数的复合 Simpson 公式

$$\int_a^b f(x)\,\mathrm{d}x \approx S_n - \frac{h^4}{2880}\Big(f'''(b) - f'''(a)\Big).$$

当 $n = 1$ 时,

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) - \frac{h^4}{2880}\Big(f'''(b) - f'''(a)\Big). \tag{4.19}$$

这就是修正的 Simpson 公式 (Corrected Simpson's Rule).

## 4.5 Romberg 求积公式

### 4.5.1 外推技巧

在使用复合求积公式时, 由于一开始并不清楚 $n$ 该取多大. 如果 $n$ 太小的话就达不到计算精度要求, 反之, 如果 $n$ 太大, 则会增加额外的工作量. 因此我们可以采用动态的方法, 即将区间不断对分, 直到所得到的计算结果满足指定的精度为止. 下面以梯形公式为例来说明这个递推过程.

将积分区间 $n$ 等分, 可得复合求积公式

$$T_n = \frac{h}{2}\sum_{i=0}^{n-1}(f(x_i) + f(x_{i+1})), \quad h = \frac{b-a}{n}.$$

如果再将每个小区间 $[x_i, x_{i+1}]$ 二等分, 则新的复合梯形公式为

$$T_{2n} = \frac{h}{4}\sum_{i=0}^{n-1}(f(x_i) + 2f(x_{i+\frac{1}{2}}) + f(x_{i+1})) = \frac{1}{2}T_n + \frac{h}{2}\sum_{i=0}^{n-1} f\left(x_i + \frac{1}{2}h\right). \tag{4.20}$$

这就是梯形法的递推公式.

下面我们给出具体的计算过程.

(1) 记 $h^{(0)} = b - a$, 计算: (梯形公式)

$$T^{(0)} = \frac{h^{(0)}}{2}(f(x_0) + f(x_1)), \quad x_i = a + ih^{(0)}, i = 0, 1.$$

(2) 将积分区间二等分, 记 $h^{(1)}=\frac{b-a}{2}=\frac{1}{2}h^{(0)}$, 计算: (复合梯形公式)

$$T^{(1)}=\frac{h^{(1)}}{2}\sum_{i=0}^{2-1}(f(x_i)+f(x_{i+1}))=\frac{1}{2}T^{(0)}+h^{(1)}\sum_{i=0}^{2^0-1}f(x_{2i+1}),$$

其中 $x_i=a+ih^{(1)}, i=0,1,2.$

(3) 再将每个小区间二等分, 记 $h^{(2)}=\frac{b-a}{2^2}=\frac{1}{2}h^{(1)}$, 计算: (复合梯形公式)

$$T^{(2)}=\frac{h^{(2)}}{2}\sum_{i=0}^{2^2-1}(f(x_i)+f(x_{i+1}))=\frac{1}{2}T^{(1)}+h^{(2)}\sum_{i=0}^{2^1-1}f(x_{2i+1}),$$

其中 $x_i=a+ih^{(2)}, i=0,1,\ldots,2^2.$

(4) 再将每个小区间二等分, 记 $h^{(3)}=\frac{b-a}{2^3}=\frac{1}{2}h^{(2)}$, 计算: (复合梯形公式)

$$T^{(3)}=\frac{h^{(3)}}{2}\sum_{i=0}^{2^3-1}(f(x_i)+f(x_{i+1}))=\frac{1}{2}T^{(2)}+h^{(3)}\sum_{i=0}^{2^2-1}f(x_{2i+1}),$$

其中 $x_i=a+ih^{(3)}, i=0,1,\ldots,2^3.$

(5) 依此类推, 对于 $k=4,5,6,\ldots$, 记 $h^{(k)}=\frac{b-a}{2^k}=\frac{1}{2}h^{(k-1)}$, 计算: (复合梯形公式)

$$T^{(k)}=\frac{h^{(k)}}{2}\sum_{i=0}^{2^k-1}(f(x_i)+f(x_{i+1}))=\frac{1}{2}T^{(k-1)}+h^{(k)}\sum_{i=0}^{2^{k-1}-1}f(x_{2i+1}),$$

其中 $x_i=a+ih^{(k)}, i=0,1,\ldots,2^k.$

**例 4.5** 用梯形法的递推公式计算定积分 $\int_0^1\frac{\sin x}{x}\,\mathrm{d}x$, 要求计算精度满足 $|T_{2n}-T_n|<\varepsilon=10^{-7}$.

**解.** (板书) □

### 4.5.2 Romberg 算法

梯形递推公式算法简单, 易于计算机实现, 但收敛速度较慢. 下面我们介绍一个加速技巧, 即 Romberg 算法.

为了讨论方便, 我们记 $T(h)\triangleq T_n$, 则 $T_{2n}=T\left(\frac{h}{2}\right)$. 由复合梯形公式余项可知

$$I(f)\triangleq\int_a^b f(x)\,\mathrm{d}x=T_n-\frac{b-a}{12}h^2f''(\eta).$$

**定理 4.7** *设 $f(x)\in C^\infty[a,b]$, 则有*

$$T(h)=I(f)+\alpha_1h^2+\alpha_2h^4+\alpha_3h^6+\cdots$$

*其中 $\alpha_k$ 与 $f(x)$ 有关, 但与 $h$ 无关.*

**证明.** 参见 [**?** , page 66]. 也可利用 $f(x)$ 的 Taylor 展开来证明. □

由定理 4.7 可知 $T(h)-I(f)=O(h^2)$, 即复合梯形公式的误差阶为 $O(h^2)$. 由于系数 $\alpha_k$ 与 $h$ 无关, 将积分区间再次二等分后可得

$$T\left(\frac{h}{2}\right)=I(f)+\alpha_1\frac{h^2}{4}+\alpha_2\frac{h^4}{1}6+\alpha_3\frac{h^6}{32}+\cdots.$$

因此

$$S(h) \triangleq \frac{4T\left(\frac{h}{2}\right) - T(h)}{3} = I(f) + \beta_1 h^4 + \beta_2 h^6 + \cdots . \tag{4.21}$$

如果用 $S(h)$ 来近似 $I(f)$, 则误差阶提高到了 $O(h^4)$. 事实上, $S(h)$ 就是复合 Simpson 公式. 上面的这种通过线性组合提高误差阶的方法就是外推算法, 或 Richardson 外推, 这是一个提高计算精度的非常重要的技巧.

类似地, 我们有

$$C(h) \triangleq \frac{16S\left(\frac{h}{2}\right) - S(h)}{15} = I(f) + \gamma_1 h^6 + \gamma_2 h^8 + \cdots . \tag{4.22}$$

这时, 误差阶提高到了 $O(h^6)$. 事实上, $C(h)$ 就是复合 Cotes 公式.

这样不断利用外推技巧, 我们就可以不断提高计算精度. 这个外推过程就是 Romberg 算法.

### 4.5.3 Romberg 算法计算过程

Romberg 算法的计算过程如下.

**算法 4.1.** Romberg Algorithm

1: 令 $k = 0, h = b - a$
2: 计算 $T_0^{(0)} = \frac{h}{2}(f(a) + f(b))$
3: 令 $k = 1$
4: 利用梯形法的递推公式计算 $T_0^{(k)} = T\left(\frac{h}{2^k}\right)$
5: **for** $m = 1, 2, \ldots, k$ **do**
6: 计算 $T_m^{(k-m)} = \dfrac{4^m T_{m-1}^{(k-m+1)} - T_{m-1}^{(k-m)}}{4^m - 1}$
7: **end for**
8: 若 $|T_k^{(0)} - T_{k-1}^{(0)}| < \varepsilon$, 则终止计算, 取 $T_k^{(0)}$ 为定积分近似值.
9: 令 $k = k + 1$, 转到第 4 步

Romberg 算法的计算过程也可以用下面的表格来描述.

| k | $h^{(k)}$ | $T_0^{(k)}$ | $T_1^{(k)}$ | $T_2^{(k)}$ | $T_3^{(k)}$ | $T_4^{(k)}$ | $\cdots$ |
|---|---|---|---|---|---|---|---|
| 0 | $b-a$ | $T_0^{(0)}$ | | | | | |
| 1 | $\frac{b-a}{2}$ | $T_0^{(1)}$ | $T_1^{(0)}$ | | | | |
| 2 | $\frac{b-a}{2^2}$ | $T_0^{(2)}$ | $T_1^{(1)}$ | $T_2^{(0)}$ | | | |
| 3 | $\frac{b-a}{2^3}$ | $T_0^{(3)}$ | $T_1^{(2)}$ | $T_2^{(1)}$ | $T_3^{(0)}$ | | |
| 4 | $\frac{b-a}{2^4}$ | $T_0^{(4)}$ | $T_1^{(3)}$ | $T_2^{(2)}$ | $T_3^{(1)}$ | $T_4^{(0)}$ | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ |

## 4.6 自适应求积方法

*To be continued ...*

## 4.7 Gauss 求积公式

### 4.7.1 为什么 Gauss 求积

在 Newton-Cotes 公式中, 我们选取的是等距节点, 这样做的好处就是计算方便. 但等距节点不一定是最好的选择, 事实上, 我们可以更好地选取节点, 使得求积公式具有更高的代数精度.

**例 4.6** 试确定 $A_i$ 和 $x_i$, 使得下面的求积公式具有尽可能高的代数精度, 并求出该求积公式的代数精度.

$$\int_{-1}^{1} f(d)\,\mathrm{d}x \approx A_0 f(x_0) + A_1 f(x_1). \tag{4.23}$$

**解.** (板书) □

由此可见, 采用不等距节点的两点求积公式 (4.23) 比采用等距节点的求积公式 (即梯形公式) 具有更高的代数精度.

### 4.7.2 Gauss 求积公式

**定义 4.5** *设 $\rho(x)$ 是 $[a,b]$ 上的权函数, 若求积公式*

$$\int_a^b \rho(x) f(x)\,\mathrm{d}x \approx \sum_{i=0}^{n} A_i f(x_i), \tag{4.24}$$

*具有 $2n+1$ 次代数精度, 则称该公式为 Gauss 求积公式, 节点 $x_i$ 称为 Gauss 点, $A_i$ 称为 Gauss 系数.*

♣ 求积公式 (4.9) 的右端只包含 $f(x)$ 的函数值, 与权函数无关.

由于求积公式 (4.9) 中含有 $2n+2$ 的待定参数, 即 $A_i$ 和 $x_i$, $i = 0, 1, 2, \ldots, n$. 因此我们可以将 $f(x) = 1, x, x^2, \ldots, x^{2n+1}$ 代入, 并令求积公式 (4.9) 精确成立, 然后解出 $A_i$ 和 $x_i$. 这样就可以使得求积公式至少具有 $2n+1$ 次代数精度, 所以, Gauss 求积公式总是存在的.

事实上, 求积公式 (4.9) 的代数精度不可能超过 $2n+1$. 取 $2n+2$ 次多项式 $f(x) = (x-x_0)(x-x_1)^2\cdots(x-x_n)^2$, 则 $\sum\limits_{i=0}^{n} A_i f(x_i) = 0$, 但显然

$$\int_a^b \rho(x) f(x)\,\mathrm{d}x > 0,$$

即求积公式 (4.9) 对 $2n+2$ 次多项式 $f(x)$ 不精确成立, 所以它的代数精度小于 $2n+2$.

**定理 4.8** *Gauss 求积公式是具有最高代数精度的插值型求积公式.*

我们所关心的是如何构造 Gauss 求积公式. 从例 4.6 可以看出, 我们可以将 $f(x) = 1, x, x^2, \ldots, x^{2n+1}$ 代入, 并令求积公式 (4.9) 精确成立, 这样就能解出 $A_i$ 和 $x_i$. 但这时需要解一个非线性方程组, 而一般情况下, 求解非线性方程组是非常困难的. 因此, 当 $n > 2$ 时, 这种方法是不可行的.

一个比较可行的方法是将 $x_i$ 和 $A_i$ 分开计算, 即先通过特殊的方法求出 Gauss 点 $x_i$, 然后再用待定系数法解出 $A_i$. 这也是目前构造 Gauss 公式的通用方法. 下面我们就介绍如何计算 Gauss 点.

### 4.7.3 Gauss 点的计算

**定理 4.9** 设节点 $a \leq x_0 < x_1 < \cdots < x_n \leq b$, 则插值型求积公式是 Gauss 公式的充要条件是多项式

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

与所有次数不超过 $n$ 的多项式正交, 即

$$\int_a^b \rho(x)\omega_{n+1}(x)p(x) = 0, \quad \forall\, p(x) \in H_n.$$

证明. (板书) □

这个定理给出了计算 Gauss 点的一般方法, 即

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

(2) 利用 $\omega_{n+1}(x)$ 与 $p(x) = 1, x, x^2, \ldots, x^n$ 正交 (带权) 的性质, 得到 $n+1$ 个线性方程, 解出 $a_k$, $k = 0, 1, 2, \ldots, n$, 这样就能确定多项式 $\omega_{n+1}(x)$;

(3) 求出多项式 $\omega_{n+1}(x)$ 的 $n+1$ 个零点, 这就是 Gauss 点.

**例 4.7** 试确定 $A_i$ 和 $x_i$, 使得下面的求积公式具有尽可能高的代数精度

$$\int_0^1 \sqrt{x} f(x)\,\mathrm{d}x \approx A_0 f(x_0) + A_1 f(x_1).$$

解. (板书) □

### 4.7.4 Gauss 求积公式的余项

设 $p_{2n+1}(x)$ 是 $f(x)$ 关于点 $x_0, x_1, x_2, \ldots, x_n$ 的 $2n+1$ 次 Hermite 插值多项式, 满足

$$p_{2n+1}(x_i) = f(x_i), \quad p'_{2n+1}(x_i) = f'(x_i), \quad i = 0, 1, 2, \ldots, n.$$

若 $f(x) \in C^{2n+2}[a, b]$, 则可以验证, 插值余项为

$$R_n(x) \triangleq f(x) - p_{2n+1}(x) = \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!}\omega_{n+1}^2.$$

由于求积公式 (4.9) 具有 $2n+1$ 次代数精度, 故

$$\int_a^b \rho(x)p_{2n+1}(x) = \sum_{i=0}^{n} A_i p_{2n+1}(x_i).$$

所以, Gauss 求积公式的余项为

$$\begin{aligned} R_n[f] &\triangleq \int_a^b \rho(x)f(x) - \sum_{i=0}^{n} A_i f(x_i) \\ &= \int_a^b \rho(x)f(x) - \sum_{i=0}^{n} A_i p_{2n+1}(x_i) \\ &= \int_a^b \rho(x)\Big(f(x) - p_{2n+1}(x)\Big)\,\mathrm{d}x \\ &= \int_a^b \rho(x)\frac{f^{(2n+2)}(\xi_x)}{(2n+2)!}\omega_{n+1}^2\,\mathrm{d}x. \end{aligned}$$

假定 $f^{(2n+2)}(\xi_x)$ 在 $[a,b]$ 上关于 $x$ 是连续的, 则由积分中值定理可知, 存在 $\eta \in (a,b)$, 使得

$$R_n[f] = \frac{f^{(2n+2)}(\eta)}{(2n+2)!} \int_a^b \rho(x)\omega_{n+1}^2 \,\mathrm{d}x. \tag{4.25}$$

### 4.7.5 Gauss 公式的收敛性与稳定性

**定理 4.10** 设 $f(x) \in C[a,b]$, 则 Gauss 求积公式是收敛的, 即

$$\lim_{n\to\infty} \sum_{i=0}^{n} A_i f(x_i) = \int_a^b \rho(x) f(x) \,\mathrm{d}x.$$

证明. 参见相关文献. □

**定理 4.11** Gauss 求积公式中的系数 $A_i$ 全是正数, 因此 Gauss 求积公式是稳定的.

证明. 令 $f(x) = l_k^2(x) = \left( \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i} \right)^2 \in H_{2n}$. 由于 Gauss 公式具有 $2n+1$ 次代数精度, 所以

$$\int_a^b \rho(x) l_k(x) \,\mathrm{d}x = \sum_{i=0}^{n} A_i l_k(x_i) = A_k.$$

上式左边显然大于 0, 故 $A_k > 0$, 所以结论成立. □

### 4.7.6 Gauss-Legendre 公式

根据定理 4.9, 如果 $\{p_n(x)\}_{n=0}^{\infty}$ 是一组在 $[a,b]$ 上带权 $\rho(x)$ 正交的多项式族, 则 $p_{n+1}(x)$ 的零点就是 Gauss 点. 因此, 我们可以使用已知的正交多项式, 如 Legendre 多项式和 Chebyshev 多项式等.

设 $[a,b] = [-1,1]$, 权函数 $\rho(x) = 1$, 则 Gauss 点即为 Legendre 多项式 $P_{n+1}(x)$ 的零点, 此时的 Gauss 公式就称为 Gauss-Legendre 公式, 简称 G-L 公式.

下面介绍几个低次 G-L 公式:

- 当 $n = 0$ 时, $P_{n+1}(x) = x$, 因此 Gauss 点为 $x_0 = 0$. 将 $f(x) = 1$ 代入公式, 令等式精确成立, 即可解出 $A_0 = 2$. 所以 G-L 公式为

$$\int_{-1}^{1} f(x) \,\mathrm{d}x \approx 2f(0).$$

- 当 $n = 1$ 时, $P_{n+1}(x) = \frac{1}{2}(3x^2 - 1)$, 因此 Gauss 点为 $x_0 = -\frac{\sqrt{3}}{3}$, $x_1 = \frac{\sqrt{3}}{3}$. 将 $f(x) = 1, x$ 代入公式, 令等式精确成立, 即可解出 $A_0 = 1, A_1 = 1$. 所以 G-L 公式为

$$\int_{-1}^{1} f(x) \,\mathrm{d}x \approx f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right) \approx f(-0.5774) + f(0.5774).$$

- 当 $n=2$ 时, 类似地, 可以得到 G-L 公式为

$$\begin{aligned}\int_{-1}^{1} f(x)\,\mathrm{d}x &\approx \frac{5}{9}f\left(-\frac{\sqrt{15}}{5}\right)+\frac{8}{9}f(0)+\frac{5}{9}f\left(\frac{\sqrt{15}}{5}\right)\\ &\approx 0.5556f(-0.7746)+0.8889f(0)+0.5556f(0.7746).\end{aligned}$$

- 当 $n=3$ 时, 可得 G-L 求积公式为

$$\begin{aligned}\int_{-1}^{1} f(x)\,\mathrm{d}x \approx{}& \frac{90-5\sqrt{30}}{180}f\left(-\sqrt{\frac{15+2\sqrt{30}}{35}}\right)+\frac{90+5\sqrt{30}}{180}f\left(-\sqrt{\frac{15-2\sqrt{30}}{35}}\right)\\ &+\frac{90+5\sqrt{30}}{180}f\left(\sqrt{\frac{15-2\sqrt{30}}{35}}\right)+\frac{90-5\sqrt{30}}{180}f\left(\sqrt{\frac{15+2\sqrt{30}}{35}}\right)\\ \approx{}& 0.3479f(-0.8611)+0.6521f(-0.3400)\\ &+0.6521f(0.3400)+0.3479f(0.8611).\end{aligned}$$

当 $n\geq 4$ 时, 我们可以使用数值方法计算 $P_{n+1}(x)$ 的零点.

**例 4.8** 用三点 G-L 公式 ($n=2$) 计算定积分 $\displaystyle\int_0^{\frac{\pi}{2}} x^2\cos x\,\mathrm{d}x$.

**解.** (板书) □

### G-L 公式的余项

由于此时 $\omega_{n+1}(x)=\tilde{P}_{n+1}(x)$. 由余项公式 (4.25) 和 Legendre 多项式性质 (3.3) 可知, G-L 公式的余项为

$$R_n[f]=\frac{f^{(2n+2)}(\eta)}{(2n+2)!}\int_{-1}^{1}\tilde{P}_{n+1}^2\,\mathrm{d}x=\frac{2^{2n+3}[(n+1)!]^4}{(2n+3)[(2n+2)!]^3}f^{(2n+2)}(\eta),\quad \eta\in(-1,1).$$

这表明 G-L 公式具有很高的精度, 如

$$R_1[f]=\frac{1}{135}f^{(4)}(\eta),\quad R_2[f]=\frac{1}{15750}f^{(6)}(\eta).$$

### 4.7.7 一般区间上的 Gauss-Legendre 公式

当积分区间是 $[a,b]$ 时, 我们可以做一个变量代换

$$x(t)=\frac{b-a}{2}t+\frac{b+a}{2},$$

可得

$$\int_a^b f(x)\,\mathrm{d}x=\frac{b-a}{2}\int_{-1}^{1}f(x(t))\,\mathrm{d}t=\frac{b-a}{2}\int_{-1}^{1}f\left(\frac{b-a}{2}t+\frac{b+a}{2}\right)\mathrm{d}t,$$

然后对上式右端的定积分使用 G-L 公式即可.

♣ 对于分段光滑函数, 可以采用复合 G-L 求积公式.

### 4.7.8 Gauss-Chebyshev 公式

设 $[a,b]=[-1,1]$, 权函数 $\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$, 则 Gauss 点即为 Chebyshev 多项式 $T_{n+1}(x)$ 的零点, 此时的 Gauss 公式就称为 Gauss-Chebyshev 公式, 简称 G-C 公式.

易知 $T_{n+1}(x)$ 的零点为

$$x_i=\cos\frac{2(n+1)}{2i+1}\pi,\quad i=0,1,2,\dots,n.$$

利用待定系数法可以求得

$$A_i=\frac{\pi}{n+1}.$$

所以 G-C 公式为

$$\int_a^b\frac{1}{\sqrt{1-x^2}}f(x)\,\mathrm{d}x\approx\frac{\pi}{n+1}\sum_{i=0}^{n}f\left(\cos\frac{2(n+1)}{2i+1}\pi\right).\tag{4.26}$$

由公式 (4.25) 可知, G-C 公式的余项为

$$R[f]=\frac{f^{(2n+2)}(\eta)}{(2n+2)!}\int_a^b\frac{1}{\sqrt{1-x^2}}\tilde{T}_{n+1}^2\,\mathrm{d}x=\frac{2\pi}{2^{2n+2}(2n+2)!}f^{(2n+2)}(\eta),\quad\eta\in(-1,1).$$

**例 4.9** 用五点 G-C 公式 ($n=4$) 计算奇异积分 $\displaystyle\int_{-1}^{1}\frac{\mathrm{e}^x}{\sqrt{1-x^2}}\,\mathrm{d}x$.

**解.** (板书) □

### 4.7.9 无穷区间上的 Gauss 公式

略.

### 4.7.10 复合 Gauss 公式

Gauss 公式最突出的优点就是具有最高的代数精度. 但缺点是 Gauss 点比较难计算, 而且当节点增加时, 需要重新计算 Gauss 点. 因此我们在实际应用中, 通常是将积分区间分割成若干小区间, 然后在每个小区间上使用低次的 Gauss 公式, 这就是复合 Gauss 公式.

## 4.8 多重积分

计算多重积分的基本思想是化为累次积分, 然后逐个进行数值积分.

对于二重积分, 如果积分区域 $\Omega$ 为矩形区域, 则

$$\iint_\Omega f(x,y)\,\mathrm{d}x\,\mathrm{d}y=\int_a^b\left(\int_c^d f(x,y)\,\mathrm{d}y\right)\mathrm{d}x.$$

如果积分区域 $\Omega$ 是 $x$ 型区域, 则

$$\iint_\Omega f(x,y)\,\mathrm{d}x\,\mathrm{d}y=\int_a^b\left(\int_{y_1(x)}^{y_2(x)} f(x,y)\,\mathrm{d}y\right)\mathrm{d}x.$$

如果积分区域 $\Omega$ 是 $y$ 型区域, 则

$$\iint_\Omega f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int_c^d \left( \int_{x_1(y)}^{x_2(y)} f(x,y)\,\mathrm{d}x \right) \mathrm{d}y.$$

**例 4.10** 利用两点 Gauss 公式计算二重积分 $\int_\Omega (x^2+2y^2)\,\mathrm{d}x$, 其中 $\Omega=[-1,1]\times[-1,1]$.

**解.** (板书) □

♣ 为了提高计算精度, 通常也使用复合求积公式.

## 4.9 数值微分

基本想法与数值积分类似, 即用函数值的线性组合来近似某点的导数值.

### 4.9.1 插值型求导公式

设 $p_n(x)$ 是 $f(x)$ 基于节点 $x_0,x_1,x_2,\ldots,x_n$ 的插值多项式, 则可以用 $p_n(x)$ 的导数来近似 $f(x)$ 的导数, 即

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

这就是插值型求导公式. 由插值余项公式可知

$$\begin{aligned} f'(x)-p_n'(x) &= \frac{d}{dx}\left(\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x)\right) \\ &= \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}'(x) + \omega_{n+1}(x)\frac{d}{dx}\left(\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\right). \end{aligned}$$

这里假定 $f^{(n+1)}(\xi_x)$ 关于 $x$ 可导. 由于 $\xi_x$ 是关于 $x$ 的未知函数, 因此右端第二项是不可求的. 但当 $x$ 是节点时, 即 $x=x_i$, 有

$$f'(x_i)-p_n'(x_i) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}'(x_i) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\prod_{k=0,k\neq i}^{n}(x_i-x_k). \tag{4.27}$$

♣ 一般情况下, 我们只考虑函数在节点处的导数值.

### 4.9.2 一阶导数的差分近似

#### 两点公式

设节点 $x_0,x_1$, 记 $h=x_1-x_0$, 则可得一次插值多项式

$$p_1(x) = \frac{x-x_1}{x_0-x_1}f(x_0) + \frac{x-x_0}{x_1-x_0}f(x_1).$$

因此

$$\begin{aligned} f'(x_0) &= p_1'(x_0) + \frac{f''(\xi_0)}{2}(x_0-x_1) = \frac{1}{h}(f(x_1)-f(x_0)) - \frac{h}{2}f''(\xi_0) \\ f'(x_1) &= p_1'(x_1) + \frac{f''(\xi_1)}{2}(x_1-x_0) = \frac{1}{h}(f(x_1)-f(x_0)) + \frac{h}{2}f''(\xi_1). \end{aligned}$$

**三点公式**

考虑等距节点 $x_0, x_1 = x_0 + h, x_2 = x_0 + 2h$, 对应的二次 Lagrange 插值多项式为

$$p_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2).$$

做变量代换: $x = x_0 + th$, 则可得

$$p_2(t) = \frac{1}{2}(t-1)(t-2)f(x_0) + t(t-2)f(x_1) + \frac{1}{2}t(t-1)f(x_2).$$

于是

$$\frac{dp_2}{dt} = \frac{1}{2}\big((2t-3)f(x_0) - 4(t-1)f(x_1) + (2t-1)f(x_2)\big).$$

所以

$$\frac{dp_2}{dx} = \frac{dp_2}{dt}\Big/\frac{dx}{dt} = \frac{1}{2h}\big((2t-3)f(x_0) - 4(t-1)f(x_1) + (2t-1)f(x_2)\big).$$

分别令 $x = x_0, x_1, x_2$, 即 $t = 0, 1, 2$, 可得

$$\begin{aligned}
p_2'(x_0) &= \frac{1}{2h}\big(-3f(x_0) + 4f(x_1) - f(x_2)\big),\\
p_2'(x_1) &= \frac{1}{2h}\big(f(x_2) - f(x_0)\big),\\
p_2'(x_2) &= \frac{1}{2h}\big(f(x_0) - 4f(x_1) + 3f(x_2)\big).
\end{aligned}$$

由公式 (4.27) 可知

$$\begin{aligned}
f'(x_0) &= \frac{1}{2h}\big(-3f(x_0) + 4f(x_1) - f(x_2)\big) + \frac{h^2}{3}f^{(3)}(\xi_0),\\
f'(x_1) &= \frac{1}{2h}\big(f(x_2) - f(x_0)\big) - \frac{h^2}{6}f^{(3)}(\xi_1),\\
f'(x_2) &= \frac{1}{2h}\big(f(x_0) - 4f(x_1) + 3f(x_2)\big) + \frac{h^2}{3}f^{(3)}(\xi_2).
\end{aligned} \tag{4.28}$$

这就是三点求导公式, 特别公式 (4.28), 是常用的计算一阶导数的 中心差分公式.

### 4.9.3 二阶导数的差分近似

计算 $p_2(x)$ 关于 $x$ 的二阶导数可得

$$\frac{d^2p_2}{dx^2} = \frac{1}{h^2}\big(f(x_0) - 2f(x_1) + f(x_2)\big).$$

结合公式 (4.27) 可知

$$f''(x_1) = \frac{1}{h^2}\big(f(x_0) - 2f(x_1) + f(x_2)\big) - \frac{h^2}{12}f^{(4)}(\xi).$$

这就是计算二阶导数常用的中心差分公式.

♣ 事实上, 以上计算一阶导数和二阶导数的中心差分公式都可以从 Taylor 展开式得到.

### 4.9.4 三次样条求导

略

### 4.9.5 数值微分的外推算法

略

## 4.10 课后练习

**练习 4.1** 确定下列求积公式中的待定参数, 使其具有尽可能高的代数精度, 并指出所构造的求积公式的代数精度.

(1) $\displaystyle\int_{-1}^{1} f(x)\,\mathrm{d}x \approx A_{-1}f(-1)+A_0f(0)+A_1f(1)$

(2) $\displaystyle\int_{-2}^{2} f(x)\,\mathrm{d}x \approx A_{-1}f(-1)+A_0f(0)+A_1f(1)$

(3) $\displaystyle\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \frac{1}{3}\Big(f(-1)+2f(x_1)+3f(x_2)\Big)$

(4) $\displaystyle\int_{0}^{h} f(x)\,\mathrm{d}x \approx \frac{h}{2}\Big(f(0)+f(h)\Big)+ah^2\Big(f'(0)-f'(h)\Big)$

**练习 4.2** 分别用复合梯形法和复合 Simpson 方法计算下列定积分:

(1) $\displaystyle\int_0^2 \frac{x}{4+x^2}\,\mathrm{d}x, \quad n=8$

(2) $\displaystyle\int_1^7 \sqrt{x}\,\mathrm{d}x, \quad n=4$

**练习 4.3** (教材习题 4) 用 Simpson 公式计算定积分 $\displaystyle\int_1^2 e^{-x}\,\mathrm{d}x$, 并估计误差.

(提示: 用余项公式估计误差)

**练习 4.4** (教材习题 5) 推导下列三种求积公式:

(1) $\displaystyle\int_a^b f(x)\,\mathrm{d}x=(b-a)f(a)+\frac{f'(\eta)}{2}(b-a)^2$

(2) $\displaystyle\int_a^b f(x)\,\mathrm{d}x=(b-a)f(b)-\frac{f'(\eta)}{2}(b-a)^2$

(3) $\displaystyle\int_a^b f(x)\,\mathrm{d}x=(b-a)f\left(\frac{a+b}{2}\right)+\frac{f''(\eta)}{24}(b-a)^3$

(提示: 实际上是证明相应求积公式的余项公式, 用标准的三步法证明)

**练习 4.5** (教材习题 7) 设 $f''(x)>0, x\in[a,b]$, 证明用梯形公式计算定积分 $I=\displaystyle\int_a^b f(x)\,\mathrm{d}x$ 所得结果比准确值大, 并说明其几何意义.

**练习 4.6** (教材习题 10, 有修改) 已知 $\rho(x)=\dfrac{1}{\sqrt{x}}$ 是 $[0,2]$ 上的权函数, 试构造 Gauss 求积公式

$$\int_0^2 \frac{1}{\sqrt{x}}f(x)\approx A_0f(x_0)+A_1f(x_1).$$

计算过程中保留小数点后两位数字. (提示: 采用标准方法, 即先求 Gauss 点, 然后求 Gauss 系数.)

**练习 4.7** (教材习题 11, 有修改) 用 $n=2$ 的 Gauss-Legendre 求积公式计算定积分:

$$\int_0^2 e^x\sin(x)\,\mathrm{d}x,$$

计算过程中保留小数点后两位数字.

**练习 4.8** (教材习题 17, 有修改) 试确定下面数值微分公式的截断误差表达式

$$f'(x_0)\approx\frac{1}{2h}\Big(3f(x_0+h)-f(x_0)-2f(x_0+2h)\Big).$$

(提示: 利用插值多项式)

练习 4.9 (教材习题 18, 有修改) 用三点公式计算 $f(x)$ 在 $x = 1.2$ 处的一阶导数和二阶导数. 函数值如下: $f(1.1) = 0.227$, $f(1.2) = 0.207$, $f(1.3) = 0.189$.

## 思考题

练习 4.10 给出复合两点 Gauss-Legendre 求积公式的余项公式.

# 第五讲 线性方程组直接解法

本章主要研究的是如何求解下面的线性方程组

$$Ax = b, \quad A \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^n.$$

在自然科学和工程技术的应用中, 很多问题的解决最终都归结为求解一个或多个线性方程组. 目前求解线性方程组的方法可以分为两大类: 直接法和迭代法. 本章主要介绍直接法. 直接法具有良好的稳定性和健壮性, 是当前求解中小规模线性方程组的首选方法, 同时也是求解某些具有特殊结构的大规模线性方程组的主要方法.

在本章中，我们总是假定系数矩阵 $A$ 是非奇异的.

## 5.1 Gauss 消去法

首先看一个例子.

**例 5.1** 求解下面的线性方程组

$$\begin{cases} x_1 - 2x_2 + 2x_3 = -2 \\ 2x_1 - 3x_2 - 3x_3 = 4 \\ 4x_1 + x_2 + 6x_3 = 3. \end{cases}$$

**解.** 利用 Gauss 消去法求解: 先写出增广矩阵, 然后通过初等变换将其转换为阶梯形, 最后通过回代求解. 具体过程可写为

$$\begin{bmatrix} 1 & -2 & 2 & -2 \\ 2 & -3 & -3 & 4 \\ 4 & 1 & 6 & 3 \end{bmatrix} \xrightarrow[③-①\times 4]{②-①\times 2} \begin{bmatrix} 1 & -2 & 2 & -2 \\ 0 & 1 & -7 & 8 \\ 0 & 9 & -2 & 11 \end{bmatrix} \xrightarrow{③-②\times 9} \begin{bmatrix} 1 & -2 & 2 & -2 \\ 0 & 1 & -7 & 8 \\ 0 & 0 & 61 & -61 \end{bmatrix}$$

通过回代求解可得

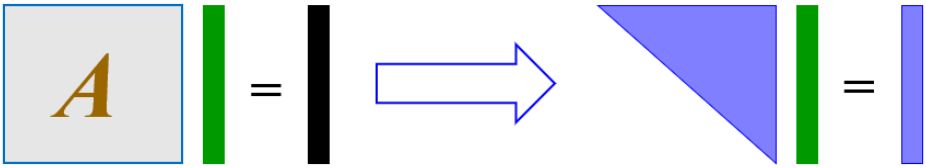
$$\begin{cases} x_3 = -1, \\ x_2 = 8 + 7x_3 = 1, \\ x_1 = -2 + 2x_2 - 2x_3 = 2. \end{cases}$$

□

将以上的做法推广到一般线性方程 $Ax = b$, 即

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

高斯消去法的主要思路: 将系数矩阵 $A$ 化为上三角矩阵, 然后回代求解:




### 5.1.1 Gauss 消去过程

本节给出 Gauss 消去过程的详细执行过程, 写出相应算法, 并编程实现.

记 $A^{(1)}=[a_{ij}^{(1)}]_{n\times n}=A$, $b^{(1)}=[b_1^{(1)},b_2^{(1)},\dots,b_n^{(1)}]^{\intercal}=b$, 即

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

第 1 步： 消第 1 列.

设 $a_{11}^{(1)}\neq 0$, 计算 $m_{i1}=\dfrac{a_{i1}^{(1)}}{a_{11}^{(1)}}$, $i=2,3,\dots,n$. 对增广矩阵进行 $n-1$ 次初等变换, 即依次将增广矩阵的第 $i$ 行减去第 1 行的 $m_{i1}$ 倍, 将新得到的矩阵记为 $A^{(2)}$, 即

$$A^{(2)}=\left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & \vdots & \ddots & \vdots & \vdots \\ & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right],$$

其中

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

第 2 步： 消第 2 列.

设 $a_{22}^{(2)}\neq 0$, 计算 $m_{i2}=\dfrac{a_{i2}^{(2)}}{a_{22}^{(2)}}$, $i=3,4,\dots,n$. 依次将 $A^{(2)}$ 的第 $i$ 行减去第 2 行的 $m_{i2}$ 倍, 将新得到的矩阵记为 $A^{(3)}$, 即

$$A^{(3)}=\left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & b_3^{(3)} \\ & & \vdots & \ddots & \vdots & \vdots \\ & & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} & b_n^{(3)} \end{array}\right],$$

其中

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

依此类推, 经过 $k-1$ 步后, 可得新矩阵 $A^{(k)}$:

$$A^{(k)}=\left[\begin{array}{ccccc|c} a_{11}^{(1)} & \cdots & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & \ddots & \vdots & & \vdots & \vdots \\ & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & b_k^{(k)} \\ & & \vdots & \ddots & \vdots & \vdots \\ & & a_{nk}^{(k)} & \cdots & a_{nn}^{(3k)} & b_n^{(k)} \end{array}\right],$$

**第 $k$ 步**： 消第 $k$ 列.

设 $a_{kk}^{(k)} \neq 0$, 计算 $m_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, i = k+1, k+2, \ldots, n$. 依次将 $A^{(k)}$ 的第 $i$ 行减去第 $k$ 行的 $m_{ik}$ 倍, 将新得到的矩阵记为 $A^{(k+1)}$, 矩阵元素的更新公式为

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)}, \quad i, j = k+1, k+2, \ldots, n. \tag{5.1}$$

**这样, 经过 $n-1$ 步后**, 即可得到一个上三角矩阵 $A^{(n)}$:

$$A^{(n)} = \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & a_{nn}^{(n)} & b_n^{(n)} \end{array}\right].$$

**最后, 回代求解**

$$x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}}, \quad x_i = \frac{1}{a_{ii}^{(i)}}\left(b_i^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)x_j}\right), \quad i = n-1, n-2, \ldots, 1.$$

♣ 由上面的求解过程可知, Gauss 消去法能顺利进行下去的充要条件是 $a_{ii}^{(i)} \neq 0, i = 1, 2, \ldots, n$, 这些元素被称为 **主元**.

**定理 5.1** 所有主元都不为零的充要条件是 $A$ 的所有顺序主子式都不为零.

**推论 5.2** Gauss 消去法能顺利完成的充要条件是 $A$ 的所有顺序主子式都不为零.

### Gauss 消去法的运算量

下面统计整个 Gauss 消去法的乘除运算的次数.

在第 $k$ 步中, 我们需要计算

$$\begin{aligned} m_{ik} &= a_{ik}^{(k)}/a_{kk}^{(k)}, i = k+1, \ldots, n \quad \to \quad n-k \text{ 次} \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, i = k+1, \ldots, n \quad \to \quad (n-k)^2 \text{ 次} \\ b_i^{(k+1)} &= b_i^{(k)} - m_{ik}b_k^{(k)}, i = k+1, \ldots, n \quad \to \quad n-k \text{ 次} \end{aligned}$$

所以整个消去过程的乘除运算为

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

易知, 回代求解过程的乘除运算为

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

所以整个 Gauss 消去法的乘除运算为

$$\frac{n^3}{3} + n^2 - \frac{n}{3}.$$

同理, 也可统计出, Gauss 消去法中的加减运算次数为

$$\frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}.$$

♣ 评价算法的一个主要指标是执行时间, 但这依赖于计算机硬件和编程技巧等, 因此直接给出算法执行时间是不太现实的. 所以我们通常是统计算法中算术运算 (加减乘除) 的次数. 在数值算法中, 大多仅仅涉及加减乘除和开方运算. 一般地, 加减运算次数与乘法运算次数具有相同的量级, 而除法运算和开方运算次数具有更低的量级.

♣ 为了尽可能地减少运算量, 在实际计算中, 数, 向量和矩阵做乘法运算时的先后执行次序为: 先计算数与向量的乘法, 然后计算矩阵与向量的乘法, 最后才计算矩阵与矩阵的乘法.

### 5.1.2 Gauss 消去法与 LU 分解

换个角度看 Gauss 消去过程: 每次都是做矩阵初等变换, 因此也可理解为不断地左乘初等矩阵. 将所有这些初等矩阵的乘积记为 $\tilde{L}$, 则可得 $\tilde{L}A = U$, 其中 $U$ 是一个上三角矩阵. 记 $L \triangleq \tilde{L}^{-1}$, 则

$$A = LU,$$

这就是著名的矩阵 LU 分解.

♣ 矩阵分解，即将一个较复杂的矩阵分解成若干具有简单结构的矩阵的乘积，是矩阵计算中的一个很重要的技术.

假定 Gauss 消去过程能顺利进行, 那么 $U$ 一定是一个非奇异上三角矩阵. 下面我们主要研究 $L$ 具有什么样的特殊结构或者特殊性质.

我们只需考察第 $k$ 步的情形, 即 $A^{(k+1)}$ 与 $A^{(k)}$ 之间的关系式. 通过观察, 由 Gauss 消去过程 (即更新公式 (5.1)) 可得

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

其中

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -m_{n,k} & & 1 \end{bmatrix}, \quad m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}},\ i = k+1, k+2, \ldots, n. \tag{5.2}$$

将所有 Gauss 消去过程结合在一起可得

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

即

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

**引理 5.1** 下面两个等式成立:

$$L_k^{-1}=\begin{bmatrix}1&&&&\\&\ddots&&&\\&&1&&\\&&m_{k+1,k}&1&\\&&\vdots&&\ddots\\&&m_{n,k}&&1\end{bmatrix},\quad L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}=\begin{bmatrix}1&&&&&\\m_{21}&1&&&&\\m_{31}&m_{32}&1&&&\\m_{41}&m_{42}&m_{4,3}&1&&\\\vdots&\vdots&\vdots&&\ddots&\\m_{n1}&m_{n2}&m_{n3}&\cdots&m_{n,n-1}&1\end{bmatrix}.$$

(证明留作练习)

记 $L\triangleq L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}$, $U\triangleq A^{(n)}$, 则

$$A=LU, \tag{5.3}$$

其中 $L$ 是单位下三角矩阵, $U$ 是非奇异上三角矩阵.

由推论 5.2 可知, 如果 $A$ 的所有顺序主子式都不为零, 则 Gauss 消去过程能顺利进行, 因此 LU 分解 (5.3) 也存在. 事实上, 我们有下面的结论.

**定理 5.3 (LU 分解的存在性和唯一性)** 设 $A\in\mathbb{R}^{n\times n}$. 则存在唯一的单位下三角矩阵 $L$ 和非奇异上三角矩阵 $U$, 使得 $A=LU$ 的充要条件是 $A$ 的所有顺序主子矩阵 $A_k=A(1:k,1:k)$ 都非奇异, $k=1,2,\ldots,n$.

**证明. 必要性**: 设 $A_{11}$ 是 $A$ 的 $k$ 阶顺序主子矩阵, 将 $A=LU$ 写成分块形式

$$\begin{bmatrix}A_{11}&A_{12}\\A_{21}&A_{22}\end{bmatrix}=\begin{bmatrix}L_{11}&0\\L_{21}&L_{22}\end{bmatrix}\begin{bmatrix}U_{11}&U_{12}\\0&U_{22}\end{bmatrix}=\begin{bmatrix}L_{11}U_{11}&L_{11}U_{12}\\L_{21}U_{11}&L_{21}U_{12}+L_{22}U_{22}\end{bmatrix}.$$

可得 $A_{11}=L_{11}U_{11}$. 由于 $L_{11}$ 和 $U_{11}$ 均非奇异, 所以 $A_{11}$ 也非奇异.

**充分性**: 用归纳法.

当 $n=1$ 时, 结论显然成立.

假设结论对 $n-1$ 阶矩阵都成立, 即对任意 $n-1$ 阶矩阵, 如果其所有的顺序主子矩阵都非奇异, 则存在 LU 分解.

考虑 $n$ 阶的矩阵 $A$, 写成分块形式

$$A=\begin{bmatrix}A_{11}&A_{12}\\A_{21}&A_{22}\end{bmatrix},$$

其中 $A_{11}\in\mathbb{R}^{(n-1)\times(n-1)}$ 是 $A$ 的 $n-1$ 阶顺序主子矩阵. 由归纳假设可知, $A_{11}$ 存在 LU 分解, 即存在单位下三角矩阵 $L_{11}$ 和非奇异上三角矩阵 $U_{11}$ 使得

$$A_{11}=L_{11}U_{11}.$$

令

$$L_{21}=A_{21}U_{11}^{-1},\quad U_{12}=L_{11}^{-1}A_{12},U_{22}=A_{22}-L_{21}U_{12},$$

则

$$A=\begin{bmatrix}A_{11}&A_{12}\\A_{21}&A_{22}\end{bmatrix}=\begin{bmatrix}L_{11}U_{11}&L_{11}U_{12}\\L_{21}U_{11}&U_{22}+L_{21}U_{12}\end{bmatrix}=\begin{bmatrix}L_{11}&0\\L_{21}&1\end{bmatrix}\begin{bmatrix}U_{11}&U_{12}\\0&U_{22}\end{bmatrix}\triangleq LU.$$

易知 $U$ 非奇异, 所以 $A$ 存在 LU 分解.

下面证明唯一性. 设 $A$ 存在两个不同的 LU 分解:

$$A = LU = \tilde{L}\tilde{U},$$

其中 $L$ 和 $\tilde{L}$ 为单位下三角矩阵, $U$ 和 $\tilde{U}$ 为非奇异上三角矩阵. 则有

$$L^{-1}\tilde{L} = U\tilde{U}^{-1},$$

该等式左边为下三角矩阵, 右边为上三角矩阵, 所以只能是对角矩阵. 由于单位下三角矩阵的逆仍然是单位下三角矩阵, 所以 $L^{-1}\tilde{L}$ 的对角线元素全是 1, 故

$$L^{-1}\tilde{L} = I,$$

即 $\tilde{L} = L, \tilde{U} = U$.

由归纳法可知, 结论成立. □

♣ 如果 $A$ 存在 LU 分解, 则 $Ax = b$ 可写为 $LUx = b$, 因此就等价于求解下面两个方程组

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

由于 $L$ 是单位下三角, $U$ 是非奇异上三角, 因此上面的两个方程组都非常容易求解.

### LU 分解算法

将上面的 LU 分解过程写成算法, 描述如下:

**算法 5.1.** LU 分解

```
Set L = I, U = 0    % 将 L 设为单位矩阵, U 设为零矩阵
for k = 1 to n − 1 do
    for i = k + 1 to n do
        m_ik = a_ik/a_kk    % 计算 L 的第 k 列
    end for
    for i = k to n do
        u_ki = a_ki    % 计算 U 的第 k 行
    end for
    for i = k + 1 to n do
        for j = k + 1 to n do
            a_ij = a_ij − m_ik u_kj    % 更新 A(k + 1 : n, k + 1 : n)
        end for
    end for
end for
```

**矩阵 $L$ 和 $U$ 的存储**

当 $A$ 的第 $i$ 列被用于计算 $L$ 的第 $i$ 列后, 在后面的计算中不再被使用. 同样地, $A$ 的第 $i$ 行被用于计算 $U$ 的第 $i$ 行后, 在后面的计算中也不再被使用. 因此, 为了节省存储空间, 我们可以在计算过程中将 $L$ 的第 $i$ 列存放在 $A$ 的第 $i$ 列, 将 $U$ 的第 $i$ 行存放在 $A$ 的第 $i$ 行, 这样就不需要另外分配空间存储 $L$ 和 $U$. 计算结束后, $A$ 的上三角部分为 $U$, 其绝对下三角部分为 $L$ 的绝对下三角部分 ($L$ 的对角线全部为 1, 不需要存储). 此时算法可以描述为:

**算法 5.2.** LU 分解

```
for k = 1 to n − 1 do
    for i = k + 1 to n do
        a_ik = a_ik / a_kk
        for j = k + 1 to n do
            a_ij = a_ij − a_ik a_kj
        end for
    end for
end for
```

### 5.1.3 列主元 Gauss 消去法与 PLU 分解

我们知道, 只要系数矩阵 $A$ 非奇异, 则线性方程组就存在唯一解. 但 Gauss 消去法却不一定有效.

**例 5.2** 求解线性方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

由于主元 $a_{11} = 0$, 因此 Gauss 消去法无法顺利进行下去.

在实际计算中, 即使主元都不为零, 但如果主元的值很小, 由于舍入误差的原因, 也可能会给计算结果带来很大的误差.

**例 5.3** 求解线性方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 0.02 & 61.3 \\ 3.43 & -8.5 \end{bmatrix}$, $b = \begin{bmatrix} 61.5 \\ 25.8 \end{bmatrix}$, 要求在运算过程中保留 3 位有效数字.

**解.** 根据 LU 分解算法 5.1, 我们可得

$$
\begin{aligned}
&l_{11} = 1.00,\ l_{21} = a_{21}/a_{11} = 1.72 \times 10^2,\ l_{22} = 1.00, \\
&u_{11} = a_{11} = 2.00 \times 10^{-2},\ u_{12} = a_{12} = 6.13 \times 10, \\
&u_{22} = a_{22} - l_{21}u_{12} \approx -8.5 - 1.05 \times 10^4 \approx -1.05 \times 10^4,
\end{aligned}
$$

即

$$
A \approx \begin{bmatrix} 1.00 & 0 \\ 1.72 \times 10^2 & 1.00 \end{bmatrix} \begin{bmatrix} 2.00 \times 10^{-2} & 6.12 \times 10 \\ 0 & -1.05 \times 10^4 \end{bmatrix}.
$$

解方程组 $Ly = b$ 可得

$$y_1 = 6.15 \times 10, \quad y_2 = b_2 - l_{21}y_1 \approx -1.06 \times 10^4.$$

解方程组 $Ux = y$ 可得

$$x_2 = y_2/u_{22} \approx 1.01, \quad x_1 = (y_1 - u_{12} * x_2)/u_{11} \approx -0.413/u_{11} \approx -20.7$$

易知, 方程的精确解为 $x_1 = 10.0$ 和 $x_2 = 1.00$. 我们发现 $x_1$ 的误差非常大. 导致这个问题的原因就是 $|a_{11}|$ 太小, 用它做主元时会放大舍入误差. □

解决上面问题的一个有效方法就是选主元. 具体做法就是, 在执行 Gauss 消去过程的第 $k$ 步之前, 插入下面的选主元过程.

① 选取 列主元 : $|a_{i_k,k}^{(k)}| = \max\limits_{k\le i\le n}\left\{|a_{i,k}^{(k)}|\right\}$

② 交换: 如果 $i_k \neq k$, 则交换第 $k$ 行与第 $i_k$ 行

上面选出的 $a_{i_k,k}^{(k)}$ 就称为 列主元. 加入这个选主元过程后, 就不会出现主元为零的情况 (除非 $A$ 是奇异的). 由此, Gauss 消去法就不会失效. 这种带选主元的 Gauss 消去法就称为 列主元 Gauss 消去法 或 部分选主元 Gauss 消去法 (Gaussian Elimination with Partial Pivoting, GEPP).

下面给出列主元 Gauss 消去法的完整算法, 其中 $L$ 存放在 $A$ 的下三角部分, $U$ 存放在 $A$ 的上三角部分.

**算法 5.3.** 列主元 Gauss 消去法

**for** $k = 1$ to $n-1$ **do**
  $a_{i_k,k} = \max_{k\le i\le n}|a_{i,k}|$ % 选列主元
  **if** $i_k \neq k$ **then**
    **for** $j = 1$ to $n$ **do**
      $a_{tmp} = a_{i_k,j}, a_{i_k,j} = a_{k,j}, a_{k,j} = a_{tmp}$ % 交换 $A$ 的第 $i_k$ 行与第 $k$ 行
    **end for**
    $b_{tmp} = b_{i_k}, b_{i_k} = b_k, b_k = b_{tmp}$ % 交换 $b$ 的第 $i_k$ 与第 $k$ 个分量
  **end if**
  **for** $i = k+1$ to $n$ **do**
    $a_{ik} = a_{ik}/a_{kk}$ % 计算 $L$ 的第 $i$ 列
    **for** $j = k+1$ to $n$ **do**
      $a_{ij} = a_{ij} - a_{ik} * a_{ij}$ % 更新 $A(k+1:n, k+1:n)$
    **end for**
    $b_i = b_i - a_{ik}b_k$
  **end for**
**end for**
$x_n = b_n/a_{nn}$ % 向后回代求解 $Ux = y$
**for** $i = n-1$ to $1$ **do**
  **for** $j = i+1$ to $n$ **do**

20: $b_i = b_i - a_{ij}x_j$
21: **end for**
22: $x_i = b_i/a_{ii}$
23: **end for**

列主元 Gauss 消去法对应的矩阵分解称为 PLU 分解.

**定理 5.4** *若矩阵 $A$ 非奇异, 则存在置换矩阵 $P$, 使得*

$$PA = LU,$$

*其中 $L$ 为单位下三角矩阵, $U$ 为上三角矩阵.*

证明. 用归纳法.

当 $n = 1$ 时, 取 $P = 1, U = A$ 即可.

假设结论对 $n - 1$ 成立.

设 $A \in \mathbb{R}^{n\times n}$ 是 $n$ 阶非奇异矩阵, 则 $A$ 的第一列至少存在一个非零元, 取置换矩阵 $\hat{P}_1$ 使得

$$\hat{P}_1 A = \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

其中 $a_{11} \neq 0$, $A_{22} \in \mathbb{R}^{(n-1)\times(n-1)}$. 记

$$u_{11} = a_{11}, \quad U_{12} = A_{12}, \quad L_{21} = A_{21}/a_{11}, \quad U_{22} = A_{22} - L_{21}U_{12}.$$

则有

$$\begin{bmatrix} 1 & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \hat{P}_1 A.$$

两边取行列式可得

$$0 \neq \det(P_1 A) = \det\left(\begin{bmatrix} 1 & 0 \\ L_{21} & I \end{bmatrix}\right) \cdot \det\left(\begin{bmatrix} u_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}\right) = a_{11} \cdot \det(U_{22}).$$

所以 $\det(U_{22}) \neq 0$, 即 $U_{22} \in \mathbb{R}^{(n-1)\times(n-1)}$ 非奇异. 由归纳假设可知, 存在置换矩阵 $\tilde{P}_1$ 使得

$$\tilde{P}_1 U_{22} = \tilde{L}_{22}\tilde{U}_{22} \quad 或 \quad U_{22} = \tilde{P}_1^{\intercal}\tilde{L}_{22}\tilde{U}_{22},$$

其中 $\tilde{L}_{22}$ 为单位下三角矩阵, $\tilde{U}_{22}$ 为非奇异上三角矩阵. 取

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1 \end{bmatrix} \hat{P}_1,$$

则有

$$\begin{aligned} P_1 A &= \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ \tilde{P}_1 L_{21} & \tilde{P}_1 \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & \tilde{P}_1^{\intercal}\tilde{L}_{22}\tilde{U}_{22} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ \tilde{P}_1 L_{21} & \tilde{P}_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1^{\intercal}\tilde{L}_{22} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & \tilde{U}_{22} \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
&= \begin{bmatrix} 1 & 0 \\ \tilde{P}_1 L_{21} & \tilde{L}_{22} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & \tilde{U}_{22} \end{bmatrix} \\
&\triangleq LU,
\end{aligned}
$$

其中 $L$ 为单位下三角矩阵, $U$ 为非奇异上三角矩阵. 由归纳法可知, 结论成立. □

♣ 列主元 Gauss 消去法比普通 Gauss 消去法要多做一些比较运算, 但 (1) 要求低, 只需系数矩阵非奇异即可; (2) 比普通 Gauss 消去法更稳定.

♣ 列主元 Gauss 消去法是当前求解线性方程组的直接法中的首选算法.

**全主元 Gauss 消去法**

在列主元 Gauss 消去法中, 我们只对所需处理的列进行选主元. 事实上, 我们也可以在剩余的子矩阵中选取主元, 以获得更好的稳定性.

① 选取 全主元 : $|a_{i_k,j_k}^{(k)}| = \max\limits_{k\le i,j\le n} \left\{|a_{i,j}^{(k)}|\right\}$

② 行交换: 如果 $i_k \neq k$, 则交换第 $k$ 行与第 $i_k$ 行

③ 列交换: 如果 $j_k \neq k$, 则交换第 $k$ 列与第 $j_k$ 列

需要指出的是, 如果有列交换, 则会改变 $x_i$ 的顺序. 因此需要记录每次的列交换次序, 以便解完后再换回来.

♣ 全主元高斯消去法具有更好的稳定性, 但很费时, 在实际计算中一般很少采用.

## 5.2 矩阵分解法

矩阵分解是矩阵计算中的一个非常重要的技术. 通过矩阵分解, 将原矩阵分解成若干个结构简单的矩阵的乘积, 从而将原本复杂的问题转化为若干个相对简单的问题. 这也是我们在实际生活中解决问题的一个基本思想.

### 5.2.1 LU 分解与 PLU 分解

如果存在一个单位下三角矩阵 $L$ 和一个非奇异上三角矩阵 $U$, 使得

$$A = LU,$$

则称 $A$ 存在 LU 分解. 相对应的, 有 Crout 分解, 即存在非奇异下三角矩阵 $\tilde{L}$ 和单位上三角矩阵 $\tilde{U}$, 使得

$$A = \tilde{L}\tilde{U}.$$

另外, 还有 LDR 分解, 即存在单位下三角矩阵 $L$, 单位上三角矩阵 $R$ 和对角矩阵 $D$, 使得

$$A = LDR.$$

显然, 这三种分解在本质上没有任何区别, 在实际计算中可以根据需要选择其中的一种. 这里我们只讨论 LU 分解.

除了利用 Gauss 消去过程来计算 LU 分解外, 还可以通过待定系数法来计算. 设 $A$ 的 LU 分解为

$$A = LU \quad \text{即} \quad \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}.$$

首先观察等式两边的第一行, 可知

$$u_{1j} = a_{1j}, \quad j = 1, 2, \ldots, n.$$

即 $U$ 的第一行就计算出来了. 再观察等式两边的第一列, 可得

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

于是 $L$ 的第一列就计算出来了. 然后再观察等式两边的第二行, 可得

$$u_{2j} = a_{2j} - l_{21}u_{1j}, \quad j = 2, 3, \ldots, n.$$

这就是 $U$ 的第二行. 同理, 由等式两边的第二列可得

$$l_{i2} = (a_{i2} - l_{i1}u_{12})/u_{22}, \quad i = 3, 4, \ldots, n.$$

这就是 $L$ 的第二列. 依此类推, 我们就可以把 $L$ 和 $U$ 的所有元素都确定下来. 这就是待定系数法.

为了写出具体的算法, 我们先写出一般过程, 即第 $k$ 步的计算公式. 此时 $U$ 的前 $k-1$ 行和 $L$ 的前 $k-1$ 列已经知道. 比较等式两边的第 $k$ 行, 可得

$$u_{kj} = a_{kj} - (l_{k1}u_{1j} + l_{k2}u_{2j} + \cdots + l_{k,k-1}u_{k-1,j}), \quad j = k, k+1, \ldots.$$

比较等式两边的第 $k$ 列, 可得

$$l_{ik} = \frac{1}{u_{kk}}(a_{ik} - l_{i1}u_{1k} - \cdots - l_{i,k-1}u_{k-1,k}), \quad i = k+1, k+2, \ldots, n.$$

由此, 我们就可以写出具体的算法. 同样地, 我们将 $L$ 存放在 $A$ 的下三角部分, $U$ 存放在 $A$ 的上三角部分.

**算法 5.4.** LU 分解 (待定系数法)

```
for k = 1 to n do
    for j = k to n do
        for i = 1 to k − 1 do
            a_kj = a_kj − a_ki a_ij
        end for
    end for
    for i = k + 1 to n do
        for j = 1 to k − 1 do
            a_ik = a_ik − a_ij a_jk
        end for
        a_ik = a_ik / a_kk
```

12: **end for**
13: **end for**

这种通过待定系数法计算 LU 分解的算法也称为 Doolittle 方法.

最后, 需要求解两个三角线性方程组, 即 $Ly = b$ 和 $Ux = y$. 完整求解过程可描述如下.

**算法 5.5.** LU 分解求解线性方程组

1: 计算 $A$ 的 LU 分解 (此处省略, 可参见算法 5.4)
2: $y_1 = b_1$ % 向前回代求解 $Ly = b$
3: **for** $i = 2$ to $n$ **do**
4: **for** $k = 1$ to $i - 1$ **do**
5: $b_i = b_i - a_{ik} y_k$
6: **end for**
7: $y_i = b_i$
8: **end for**
9: $x_n = y_n / a_{nn}$ % 向后回代求解 $Ux = y$
10: **for** $i = n - 1$ to $1$ **do**
11: **for** $k = i + 1$ to $n$ **do**
12: $y_i = y_i - a_{ik} x_k$
13: **end for**
14: $x_i = y_i / a_{ii}$
15: **end for**

下面考虑列主元 LU 分解 (PLU ), 即

$$PA = LU,$$

其中 $P$ 是一个置换矩阵. 因此, $PA$ 就相当于对 $A$ 的行进行了重新排列. 为了节省存储量, 我们并不存储这个置换矩阵, 而只是用一个向量来表示这个重新排列. 我们将这个向量记为 $p$, 其元素是 $\{1, 2, \ldots, n\}$ 的一个重排列. 比如 $p = [1, 3, 2]$ 表示交换矩阵的第 2 行和第 3 行, 而 $p = [2, 3, 1]$ 则表示将矩阵的第 2 行移到最前面, 将第 3 行移到第 2 行, 将第 1 行移到最后.

易知, 在计算过程中, $p$ 的初始值为 $p = [1, 2, \ldots, n]$. 在选列主元的过程中, 如果出现行交换, 即交换第 $i_k$ 行和第 $k$ 行, 则需要相应地交换 $p$ 的第 $i_k$ 位置和第 $k$ 位置上的值. 具体算法如下.

**算法 5.6.** PLU: 列主元 LU 分解或部分选主元 LU 分解

1: $p = [1, 2, \ldots, n]$ % 用于记录置换矩阵
2: **for** $k = 1$ to $n - 1$ **do**
3: $a_{i_k, k} = \max_{k \le i \le n} |a_{i,k}|$ % 选列主元
4: **if** $i_k \neq k$ **then**
5: **for** $j = 1$ to $n$ **do**

$a_{tmp} = a_{i_k,j}, a_{i_k,j} = a_{k,j}, a_{k,j} = a_{tmp}$ % 交换 $A$ 的第 $i_k$ 行与第 $k$ 行
**end for**
$p_{tmp} = p_{i_k}, p_{i_k} = p_k, p_k = p_{tmp}$ % 更新置换矩阵
**end if**
**for** $i = k+1$ to $n$ **do**
$a_{ik} = a_{ik}/a_{kk}$ % 计算 $L$ 的第 $i$ 列
**for** $j = k+1$ to $n$ **do**
$a_{ij} = a_{ij} - a_{ik} * a_{ij}$ % 更新 $A(k+1:n, k+1:n)$
**end for**
**end for**
**end for**

将求解两个三角线性方程组结合起来, 就得到完整的求解过程.

**算法 5.7.** PLU 分解求解线性方程组

计算 $A$ 的 PLU 分解 (此处省略, 可参见算法 5.6)
$y_1 = b_{p_1}$ % 向前回代求解 $Ly = Pb$
**for** $i = 2$ to $n$ **do**
**for** $j = 1$ to $i-1$ **do**
$b_{p_i} = b_{p_i} - a_{ij}y_j$
**end for**
$y_i = b_{p_i}$
**end for**
$x_n = b_{p_n}/a_{nn}$ % 向后回代求解 $Ux = y$
**for** $i = n-1$ to $1$ **do**
**for** $j = i+1$ to $n$ **do**
$y_i = y_i - a_{ij}x_j$
**end for**
$x_i = y_i/a_{ii}$
**end for**

♣ 算法 5.7 可用于计算矩阵的逆, 也可用于计算矩阵的行列式.

### 5.2.2 Cholesky 分解与平方根法

前面讨论的是一般线性方程组, 并没有考虑系数矩阵的特殊结构. 如果 $A$ 是对称正定的, 则可以得到更加简洁高效的方法.

**定理 5.5 (Cholesky 分解)** 设 $A$ 对称正定, 则存在唯一的对角线元素全为正的下三角矩阵 $L$, 使得

$$A = LL^{\intercal}.$$

该分解称为 Cholesky 分解.

证明. 首先证明存在性, 我们用数学归纳法来构造矩阵 $L$.

当 $n=1$ 时, 由 $A$ 的对称正定性可知 $a_{11}>0$. 取 $l_{11}=\sqrt{a_{11}}$ 即可.

假定结论对所有不超过 $n-1$ 阶的对称正定矩阵都成立. 设 $A\in\mathbb{R}^{n\times n}$ 是 $n$ 阶对称正定, 则 $A$ 可分解为

$$A=\begin{bmatrix} a_{11} & A_{12} \\ A_{12}^{\intercal} & A_{22} \end{bmatrix}=\begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^{\intercal} & I \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & \tilde{A}_{22} \end{bmatrix}\begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^{\intercal} & I \end{bmatrix}^{\intercal},$$

其中 $\tilde{A}_{22}=A_{22}-A_{12}^{\intercal}A_{12}/a_{11}$. 易知, $\begin{bmatrix} 1 & 0 \\ 0 & \tilde{A}_{22} \end{bmatrix}$ 对称正定, 故 $\tilde{A}_{22}$ 是 $n-1$ 阶对称正定矩阵. 根据归纳假设, 存在唯一的对角线元素为正的下三角矩阵 $\tilde{L}$, 使得 $\tilde{A}_{22}=\tilde{L}\tilde{L}^{\intercal}$. 令

$$L=\begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^{\intercal} & I \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & \tilde{L} \end{bmatrix}=\begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^{\intercal} & \tilde{L} \end{bmatrix}.$$

易知, $L$ 是对角线元素均为正的下三角矩阵, 且

$$LL^{\intercal}=\begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^{\intercal} & I \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & \tilde{L} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & \tilde{L}^{\intercal} \end{bmatrix}\begin{bmatrix} \sqrt{a_{11}} & 0 \\ \frac{1}{\sqrt{a_{11}}}A_{12}^{\intercal} & I \end{bmatrix}^{\intercal}=A.$$

由归纳法可知, 对任意对称正定实矩阵 $A$, 都存在一个对角线元素为正的下三角矩阵 $L$, 使得

$$A=LL^{\intercal}.$$

唯一性可以采用反证法, 留做作业. □

♣ Cholesky 分解仅针对对称正定矩阵成立.

**定理 5.6** *若 $A$ 对称, 且所有顺序主子式都不为 0, 则 $A$ 可唯一分解为*

$$A=LDL^{\intercal},$$

*其中 $L$ 为单位下三角矩阵, $D$ 为对角矩阵.* (证明留作练习)

**如何计算 Cholesky 分解**

我们仍然使用待定系数法. 设

$$A=LL^{\intercal}\quad 即\quad \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}=\begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n,2} & \cdots & l_{nn} \end{bmatrix}\begin{bmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ & l_{22} & \cdots & l_{n2} \\ & & \ddots & \vdots \\ & & & l_{n,n} \end{bmatrix}.$$

类似于 LU 分解, 直接比较等式两边的元素, 可得一般公式

$$a_{ij}=\sum_{k=1}^{n}l_{ik}l_{jk}=\sum_{k=1}^{j-1}l_{ik}l_{jk}+l_{jj}l_{ij},\quad j=1,2,\ldots,n,\ i=j,j+1,\ldots,n.$$

根据这个计算公式即可得下面的算法 (这里我们将 $L$ 存放在 $A$ 的下三角部分).

**算法 5.8.** Cholesky 分解

**for** $j = 1$ to $n$ **do**
  **for** $k = 1$ to $j - 1$ **do**
    $a_{jj} = a_{jj} - a_{jk}^2$
  **end for**
  $a_{jj} = \sqrt{a_{jj}}$
  **for** $i = j + 1$ to $n$ **do**
    **for** $k = 1$ to $j - 1$ **do**
      $a_{ij} = a_{ij} - a_{ik}a_{jk}$
    **end for**
    $a_{ij} = a_{ij}/a_{jj}$
  **end for**
**end for**

♣ Cholesky 分解算法的乘除运算量为 $\frac{1}{3}n^3 + \mathcal{O}(n^2)$, 大约为 LU 分解的一半. 另外, Cholesky 分解算法是稳定的 (稳定性与全主元 Gauss 消去法相当), 因此不需要选主元.

利用 Cholesky 分解求解线性方程组的方法就称为 **平方根法**, 具体描述如下:

**算法 5.9.** Cholesky 分解求解线性方程组

计算 Cholesky 分解 (此处省略, 参见算法 5.8)
$y_1 = b_1/a_{11}$ % 向前回代求解 $Ly = b$
**for** $i = 2$ to $n$ **do**
  **for** $j = 1$ to $i - 1$ **do**
    $b_i = b_i - a_{ij}y_j$
  **end for**
  $y_i = b_i/a_{ii}$
**end for**
$x_n = b_n/a_{nn}$ % 向后回代求解 $L^\intercal x = y$
**for** $i = n - 1$ to $1$ **do**
  **for** $j = i + 1$ to $n$ **do**
    $y_i = y_i - a_{ji}x_j$
  **end for**
  $x_i = y_i/a_{ii}$
**end for**

**改进的 Cholesky 分解**

在 Cholesky 分解中, 需要计算平方根. 为了避免计算平方根, 我们可以采用改进的 Cholesky 分解, 即

$$A = LDL^{\intercal},$$

其中 $L$ 为单位下三角矩阵, $D$ 为对角矩阵. 这个分解也称为 $\mathrm{LDL}^{\intercal}$ 分解. 由待定系数法, 设

$$A = LDL^{\intercal} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} 1 & l_{21} & \cdots & l_{n1} \\ & 1 & \ddots & \vdots \\ & & \ddots & l_{n,n-1} \\ & & & 1 \end{bmatrix}.$$

比较等式两边的元素, 可得

$$a_{ij} = d_j l_{ij} + \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}, \quad j = 1, 2, \ldots, n, \quad i = j+1, j+2, \ldots, n.$$

基于以上分解的求解对称正定线性方程组的算法就称为**改进的平方根法**, 描述如下 (将 $L$ 存放在 $A$ 的下三角部分, $D$ 存放在 $A$ 的对角部分).

**算法 5.10.** *改进的平方根法*

**for** $j = 1$ to $n$ **do**
  **for** $k = 1$ to $j-1$ **do**
    $a_{jj} = a_{jj} - l_{jk}^2 a_{kk}$
  **end for**
  **for** $i = j+1$ to $n$ **do**
    **for** $k = 1$ to $j-1$ **do**
      $a_{ij} = a_{ij} - a_{ik} a_{kk} a_{jk}$
    **end for**
    $a_{ij} = a_{ij} / a_{jj}$
  **end for**
**end for**
$y_1 = b_1$ % 向前回代求解 $Ly = b$
**for** $i = 2$ to $n$ **do**
  **for** $j = 1$ to $i-1$ **do**
    $b_i = b_i - a_{ij} y_j$
  **end for**
  $y_i = b_i$
**end for**
$x_n = b_n / a_{nn}$ % 向后回代求解 $DL^{\intercal} x = y$
**for** $i = n-1$ to $1$ **do**
  $x_i = y_i / a_{ii}$
  **for** $j = i+1$ to $n$ **do**

23: $x_i = x_i - a_{ji}x_j$
24: **end for**
25: **end for**

### 5.2.3 三对角矩阵的追赶法

这里介绍一种求解三对角线性方程组的特殊方法.

考虑三对角线性方程组 $Ax = f$, 其中 $A$ 是三对角矩阵:

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_1 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_{n-1} & b_n \end{bmatrix}.$$

我们假定

$$|b_1| > |c_1| > 0, \quad |b_n| > |a_{n-1}| > 0, \tag{5.4}$$

且

$$|b_i| \geq |a_{i-1}| + |c_i|, \quad a_i c_i \neq 0, \quad i = 1, \ldots, n-1. \tag{5.5}$$

即 $A$ 是不可约弱对角占优的. 此时, 我们可以得到下面的三角分解 (Crout 分解)

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_1 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_{n-1} & b_n \end{bmatrix} = \begin{bmatrix} \alpha_1 & & & \\ a_1 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & a_{n-1} & \alpha_n \end{bmatrix} \begin{bmatrix} 1 & \beta_1 & & \\ & 1 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & 1 \end{bmatrix} \triangleq LU. \tag{5.6}$$

由待定系数法, 我们可以得到递推公式:

$$\begin{aligned} &\alpha_1 = b_1, \quad \beta_1 = c_1/\alpha_1 = c_1/b_1, \\ &\begin{cases} \alpha_i = b_i - a_{i-1}\beta_{i-1}, \\ \beta_i = c_i/\alpha_i = c_i/(b_i - a_{i-1}\beta_{i-1}), \end{cases} \quad i = 2, 3, \ldots, n-1 \\ &\alpha_n = b_n - a_{n-1}\beta_{n-1}. \end{aligned}$$

为了使得算法能够顺利进行下去, 我们需要证明 $\alpha_i \neq 0$.

**定理 5.7** 设三对角矩阵 $A$ 满足条件 (5.4) 和 (5.5). 则 $A$ 非奇异, 且

(1) $|\alpha_1| = |b_1| > 0$;

(2) $0 < |\beta_i| < 1, i = 1, 2, \ldots, n-1$;

(3) $0 < |c_i| \leq |b_i| - |a_{i-1}| < |\alpha_i| < |b_i| + |a_{i-1}|, \quad i = 2, 3, \ldots, n$;

证明. 由于 $A$ 是不可约且弱对角占优, 所以 $A$ 非奇异.

结论 (1) 是显然的.

下面我们证明结论 (2) 和 (3).

由于 $0 < |c_1| < |b_1|$, 且 $\beta_1 = c_1/b_1$, 所以 $0 < |\beta_1| < 1$. 又 $\alpha_2 = b_2 - a_1\beta_1$, 所以

$$|\alpha_2| \geq |b_2| - |a_1| \cdot |\beta_1| > |b_2| - |a_1| \geq |c_2| > 0, \tag{5.7}$$

$$|\alpha_2| \leq |b_2| + |a_1| \cdot |\beta_1| < |b_2| + |a_1|. \tag{5.8}$$

再由结论 (5.7) 和 $\beta_2$ 的计算公式可知 $0 < |\beta_2| < 1$. 类似于 (5.7) 和 (5.7), 我们可以得到

$$|\alpha_3| \geq |b_3| - |a_2| \cdot |\beta_2| > |b_3| - |a_2| \geq |c_3| > 0,$$

$$|\alpha_3| \leq |b_3| + |a_2| \cdot |\beta_2| < |b_3| + |a_2|.$$

依此类推, 我们就可以证明结论 (2) 和 (3). □

由定理 5.7 可知, 分解 (5.6) 是存在的. 因此, 原方程就转化为求解 $Ly = f$ 和 $Ux = y$. 由此便可得求解三对角线性方程组的**追赶法** 也称为 Thomas **算法** (1949), 其乘除运算量大约为 $5n$, 加减运算大约为 $3n$.

**算法 5.11.** 追赶法

$\alpha_1 = b_1$
$\beta_1 = c_1/b_1$
$y_1 = f_1/b_1$
**for** $i = 2$ to $n-1$ **do**
    $\alpha_i = b_i - a_{i-1}\beta_{i-1}$
    $\beta_i = c_i/\alpha_i$
    $y_i = (f_i - a_{i-1}y_{i-1})/\alpha_i$
**end for**
$\alpha_n = b_n - a_{n-1}\beta_{n-1}$
$y_n = (f_n - a_{n-1}y_{n-1})/\alpha_n$
$x_n = y_n$
**for** $i = n-1$ to $1$ **do**
    $x_i = y_i - \beta_i x_{i+1}$
**end for**

♣ 具体计算时, 由于求解 $Ly = f$ 与矩阵 LU 分解是同时进行的, 因此, $\alpha_i$ 可以不用存储. 但 $\beta_i$ 需要存储.

♣ 由于 $|\beta_i| < 1$, 因此在回代求解 $x_i$ 时, 误差可以得到有效控制.

需要指出的是, 我们也可以考虑下面的分解

$$A = \begin{bmatrix} b_1 & c_1 & & \\ a_1 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_{n-1} & b_n \end{bmatrix} = \begin{bmatrix} 1 & & & \\ \gamma_1 & 1 & & \\ & \ddots & \ddots & \\ & & \gamma_{n-1} & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 & c_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & c_{n-1} \\ & & & \alpha_n \end{bmatrix}. \tag{5.9}$$

但此时 $|\gamma_i|$ 可能大于 1. 比如 $\gamma_1 = a_1/b_1$, 因此当 $|b_1| < |a_1|$ 时, $|\gamma_1| > 1$. 所以在回代求解时, 误差可能得不到有效控制. 另外一方面, 计算 $\gamma_i$ 时也可能会产生较大的舍入误差 (大数除以小数). 但如果 $A$ 是列对角占优, 则可以保证 $|\gamma_i| < 1$.

♣ 如果 $A$ 是 (行) 对角占优, 则采用分解 (5.6); 如果 $A$ 是列对角占优, 则采用分解 (5.9).

## 5.3 误差分析

### 5.3.1 矩阵条件数

**定义 5.1** 考虑线性方程组 $Ax = b$, 如果 $A$ 或 $b$ 的微小变化会导致解的巨大变化, 则称此线性方程组是 病态 的, 并称矩阵 $A$ 是病态的, 反之则是 良态 的.

**例 5.4** 考虑线性方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix}$, $b = [2, 2]^\intercal$, 可求得解为 $x = [2, 0]^\intercal$. 如果 $b$ 的第二个元素出现细微误差, 变为 $b = [2, 2.0001]^\intercal$, 则解变为 $x = [1, 1]^\intercal$. 由此可见, 当右端项出现细微变化时, 解会出现很大的变化, 因此该线性方程组是病态的, 系数矩阵 $A$ 也是病态的.

怎样来判断一个矩阵是否病态? 目前比较常用的一个指标就是 矩阵条件数.

**定义 5.2** 设 $A$ 非奇异, $\|\cdot\|$ 是任一算子范数, 则称

$$\mathrm{Cond}(A) \triangleq \|A^{-1}\| \cdot \|A\|$$

为 $A$ 的 条件数.

♣ 常用的矩阵条件数有

$$\mathrm{Cond}(A)_2 \triangleq \|A^{-1}\|_2 \cdot \|A\|_2, \quad \mathrm{Cond}(A)_1 \triangleq \|A^{-1}\|_1 \cdot \|A\|_1, \quad \mathrm{Cond}(A)_\infty \triangleq \|A^{-1}\|_\infty \cdot \|A\|_\infty.$$

♣ $\mathrm{Cond}(A)_2$ 也称为 谱条件数, 当 $A$ 对称时, 有

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

**引理 5.2** 条件数具有以下性质:

- $\mathrm{Cond}(A) \ge 1, \quad \forall\, A \in \mathbb{R}^{n\times n}$
- 对任意非零常数 $\alpha \in \mathbb{R}$, 都有

$$\mathrm{Cond}(\alpha A) = \mathrm{Cond}(A).$$

- 对任意正交矩阵 $Q \in \mathbb{R}^{n\times n}$, 有 $\mathrm{Cond}(Q)_2 = 1$.
- 设 $Q$ 是正交矩阵, 则对任意矩阵 $A$ 有

$$\mathrm{Cond}(QA)_2 = \mathrm{Cond}(A)_2 = \mathrm{Cond}(AQ)_2.$$

证明. 直接验证即可. □

### 5.3.2 条件数与病态之间的关系

考虑线性方程组 $Ax = b$. 假定系数矩阵 $A$ 是精确的, 而右端项 $b$ 有个微小扰动 $\delta b$. 因此我们实际求解的是线性方程组

$$A\hat{x} = b + \delta b.$$

我们称这个方程组为 **扰动方程组**, 其解记为 $\hat{x}$.

记 $\delta x \triangleq \hat{x} - x_*$, 其中 $x_* = A^{-1}b$ 为精确解. 则

$$\delta x = A^{-1}(b + \delta b) - x_* = A^{-1}\delta b.$$

所以

$$\|\delta x\| \leq \|A^{-1}\| \cdot \|\delta b\|. \tag{5.10}$$

又由 $Ax_* = b$ 可知

$$\|b\| \leq \|A\| \cdot \|x_*\|, \quad \text{即} \quad \frac{1}{\|x_*\|} \leq \frac{\|A\|}{\|b\|}. \tag{5.11}$$

由 (5.10) 和 (5.11) 两边相乘可得

$$\frac{\|\delta x\|}{\|x_*\|} \leq \|A^{-1}\| \cdot \|A\| \cdot \frac{\|\delta b\|}{\|b\|}$$

**定理 5.8** 设 $\|\cdot\|$ 是任一向量范数（当该范数作用在矩阵上时就是相应的算子范数）, 若 $A$ 是精确的, $b$ 有个小扰动 $\delta b$, 则有

$$\frac{\|\delta x\|}{\|x_*\|} \leq \|A^{-1}\| \cdot \|A\| \cdot \frac{\|\delta b\|}{\|b\|}.$$

♣ 上述结论说明, 由于右端项的扰动而产生的解的相对误差, 大约被放大了 $\|A^{-1}\| \cdot \|A\|$ 倍. 这个倍数正好是系数矩阵的条件数. 另外, 需要指出的是, 这是最坏的情况.

如果 $A$ 也有微小的扰动, 设为 $\delta A$, 则扰动方程组为

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

假定 $\|\delta A\|$ 很小, 满足 $\|A^{-1}\| \cdot \|\delta A\| < 1$, 则 $\|A^{-1}\delta A\| \leq \|A^{-1}\| \cdot \|\delta A\| < 1$. 由定理 1.19 可知 $I + A^{-1}\delta A$ 非奇异, 所以 $A + \delta A = A(I + A^{-1}\delta A)$ 也非奇异. 于是

$$\begin{aligned}\delta x = \hat{x} - x_* &= (A + \delta A)^{-1}(b + \delta b - Ax_* - \delta A x_*) \\ &= (I + A^{-1}\delta A)^{-1}A^{-1}(-\delta A x_* + \delta b).\end{aligned}$$

由定理 1.19 可知

$$\begin{aligned}\frac{\|\delta x\|}{\|x_*\|} &\leq \|(I + A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\| \cdot \left(\|\delta A\| + \frac{\|\delta b\|}{\|x_*\|}\right) \\ &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \cdot \|\delta A\|} \cdot \left(\|\delta A\| + \frac{\|\delta b\|}{\|x_*\|}\right) \\ &= \frac{\|A^{-1}\| \cdot \|A\|}{1 - \|A^{-1}\| \cdot \|A\| \cdot \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|x_*\|}\right)\end{aligned}$$

$$\leq \frac{\mathrm{Cond}(A)}{1-\mathrm{Cond}(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right)$$

当 $\|\delta A\|\to 0$ 时, 我们可得

$$\frac{\|\delta x\|}{\|x_*\|}\leq \frac{\mathrm{Cond}(A)}{1-\mathrm{Cond}(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right)\to \mathrm{Cond}(A)\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right)$$

**定理 5.9** 设 $\|\cdot\|$ 是任一向量范数（当该范数作用在矩阵上时就是相应的算子范数）, 假定 $\|A^{-1}\|\cdot\|\delta A\|<1$, 则有

$$\frac{\|\delta x\|}{\|x_*\|}\leq \frac{\mathrm{Cond}(A)}{1-\mathrm{Cond}(A)\cdot\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right).$$

当 $\delta b=0$ 时, 有

$$\frac{\|\delta x\|}{\|x_*\|}\leq \frac{\mathrm{Cond}(A)}{1-\mathrm{Cond}(A)\cdot\frac{\|\delta A\|}{\|A\|}}\cdot\frac{\|\delta A\|}{\|A\|}.$$

事实上, 由于 $x_*$ 通常是未知的, 因此一个更加实际的情况是考虑 $\delta x$ 与 $\hat{x}$ 之间的关系.

**定理 5.10** 设 $\|\cdot\|$ 是任一向量范数（当该范数作用在矩阵上时就是相应的算子范数）, 则 $\delta x$ 与 $\hat{x}$ 满足下面的关系式

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

当 $\delta b=0$ 时, 有

$$\frac{\|\delta x\|}{\|\hat{x}\|}\leq \kappa(A)\frac{\|\delta A\|}{\|A\|} \tag{5.12}$$

**证明.** 由等式 $(A+\delta A)\hat{x}=b+\delta b=Ax_*+\delta b$ 可知 $A(\hat{x}-x_*)=-\delta A\hat{x}+\delta b$, 即

$$\delta x=A^{-1}(-\delta A\hat{x}+\delta b).$$

所以

$$\|\delta x\|\leq \|A^{-1}\|\cdot(\|\delta A\|\cdot\|\hat{x}\|+\|\delta b\|), \tag{5.13}$$

即

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

若 $\delta b=0$, 则可得

$$\frac{\|\delta x\|}{\|\hat{x}\|}\leq \kappa(A)\frac{\|\delta A\|}{\|A\|}$$

□

♣ 上面的结论具有理论指导作用, 但如果无法获得 $\delta A$ 和 $\delta b$ 的大小, 则很难用来估计 $\delta x$ 的大小. 此时, 我们可以通过残量来估计.

记残量 (残差) 为 $r = b - A\hat{x}$, 则有

$$\delta x = \hat{x} - x_* = \hat{x} - A^{-1}b = A^{-1}(A\hat{x} - b) = -A^{-1}r,$$

所以可得

$$\|\delta x\| \leq \|A^{-1}\| \cdot \|r\|$$

这个估计式的优点是不需要知道 $\delta A$ 和 $\delta b$ 的大小. 而且在实际计算中, $r$ 通常是可计算的, 因此该估计式比较实用.

## 5.4 解的改进

当矩阵 $A$ 是病态时, 即使残量 $r = b - A\hat{x}$ 很小, 所求得的数值解 $\hat{x}$ 仍可能带有较大的误差. 此时需要通过一些方法来提高解的精度.

### 5.4.1 高精度运算

在计算中, 尽可能采用高精度的运算. 比如, 原始数据是单精度的, 但在计算时都采用双精度运算, 或者更高精度的运算. 但更高精度的运算会带来更大的开销.

### 5.4.2 矩阵元素缩放 (Scaling)

如果 $A$ 的元素在数量级上相差很大, 则在计算过程中很可能会出现大数与小数的加减运算, 这样就可能会引入更多的舍入误差. 为了避免由于这种情况而导致的舍入误差, 我们可以在求解之前先对矩阵元素进行缩放 (Scaling), 即在矩阵两边同时乘以两个适当的对角矩阵.

**例 5.5** 考虑线性方程组

$$\begin{bmatrix} -4000 & 2000 & 2000 \\ 2000 & 0.78125 & 0 \\ 2000 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 400 \\ 1.3816 \\ 1.9273 \end{bmatrix}.$$

用部分选主元 Gauss 消去法求解, 计算过程中保留 8 位有效数字, 最后求得的数值解为

$$\tilde{x} = [0.00096365, -0.698496, 0.90042329]^\intercal.$$

而精确解为 $x = [1.9273..., -0.698496..., 0.9004233...]^\intercal$. 数值解的第一个分量存在很大的误差.

我们考虑对矩阵元素进行缩放, 即在方程两边同时乘以一个对角矩阵 $D = \mathrm{diag}(0.00005, 1, 1)$, 然后求解一个新的方程组

$$DADy = Db.$$

最后令 $\tilde{x} = Dy$, 即可求得比较精确的数值解.

♣ 为了平衡矩阵元素的大小, 一种好的方案是左乘一个对角矩阵 $D^{-1}$ (即对 $A$ 的行进行缩放), 其中 $D_{ii} = \sum_{j=1}^{n} |a_{ij}|$. 然后再执行部分选主元 LU 分解.

### 5.4.3 迭代改进法

设近似解 $\hat{x}$, 残量 $r = b - A\hat{x}$. 当 $\hat{x}$ 没达到精度要求时, 可以考虑方程组 $Az = r$. 设 $z_*$ 是该方程组的精确解, 则

$$A(\hat{x} + z_*) = A\hat{x} + Az_* = (b - r) + r = b,$$

因此 $\hat{x} + z_*$ 就是原方程组的精确解.

在实际计算中, 我们可能得到的是近似解 $\hat{z}$, 但通常 $\|r - A\hat{z}\|$ 应该比较小, 特别地, 比 $\|r\|$ 更小. 因此 $\hat{x} + \hat{z}$ 应该比 $\hat{x}$ 更接近精确解.

如果新的近似解 $\hat{x} + \hat{z}$ 还不满足精度要求, 则可重复以上过程. 这就是通过迭代来提高解的精度.

算法 5.12. 通过迭代改进解的精度

1: 设 $PA = LU$, $\hat{x}$ 是 $Ax = b$ 的近似解
2: **while** 近似解 $\hat{x}$ 不满足精度要求, **do**
3: 计算 $r = b - A\hat{x}$
4: 求解 $Ly = Pr$, 即 $y = L^{-1}Pr$
5: 求解 $Uz = y$, 即 $z = U^{-1}y$
6: 令 $\hat{x} = \hat{x} + z$
7: **end while**

由于每次迭代只需计算一次残量和求解两个三角线性方程组, 因此运算量为 $\mathcal{O}(n^2)$. 所以相对来讲还是比较经济的.

♣ 为了提高计算精度, 在计算残量 $r$ 时最好使用原始数据 $A$, 而不是 $P^{\intercal}LU$, 因此对 $A$ 做 LU 分解时需要保留矩阵 $A$, 不能被 $L$ 和 $U$ 覆盖.

♣ 实际计算经验表明, 当 $A$ 病态不是很严重时, 即 $\varepsilon_u \kappa_\infty(A) < 1$, 迭代法可以有效改进解的精度, 最后达到机器精度. 但 $\varepsilon_u \kappa_\infty(A) \geq 1$ 时, 一般没什么效果. 这里 $\varepsilon_u$ 表示机器精度.

## 5.5 本章小结

*To be continued ...*

## 5.6 课后练习

练习 5.1 设 $A$ 是对称矩阵且 $a_{11} \neq 0$, 经过一步 Gauss 消去法后, $A$ 约化为

$$\begin{bmatrix} a_{11} & a_1^{\mathsf{T}} \\ 0 & A_{22} \end{bmatrix}.$$

试证明 $A_{22}$ 是对称矩阵.

练习 5.2 设 $A$ 对称正定, 经过一步 Gauss 消去法后, $A$ 约化为

$$\begin{bmatrix} a_{11} & a_1^{\mathsf{T}} \\ 0 & A_{22} \end{bmatrix}.$$

试证明 $A_{22}$ 是对称正定矩阵.

练习 5.3 (教材习题 5) 设 $Ux = b$, 其中 $U$ 是对角线均非零的上三角矩阵, 试推导求解公式, 并写出算法, 统计乘除法的次数.

练习 5.4 (教材习题 7, 有修改) 用列主元消去法求解线性方程组:

$$\begin{cases} 2x_1 + 5x_2 + 3x_3 = 2, \\ 3x_1 + 3x_2 - x_3 = -6, \\ x_1 + 2x_2 + x_3 = -1, \end{cases}$$

并求出系数矩阵 $A$ 的行列式.

练习 5.5 (教材习题 8, 有修改)

练习 5.6 (教材习题 9, 有修改) 用追赶法解三对角方程组 $Ax = b$，其中

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

练习 5.7 (教材习题 10 修改版) 用改进的平方根法 (即 $\mathrm{LDL}^{\mathrm{T}}$ 分解) 求解线性方程组

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

练习 5.8 (教材习题 11 修改版) 下列矩阵是否存在 LU 分解 ($L$ 单位下三角, $U$ 非奇异上三角)

$$A = \begin{bmatrix} 2 & 1 & 4 \\ 3 & 3 & 1 \\ 3 & 2 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 3 & 0 \\ 0 & 1 & 4 \end{bmatrix}.$$

练习 5.9 (教材习题 12 修改版) 设矩阵

$$A = \begin{bmatrix} 5 & -4 & 2 \\ -4 & 5 & -2 \\ 2 & -2 & 1 \end{bmatrix}.$$

计算 $A$ 的行范数, 列范数, 2-范数和 F-范数.

**练习 5.10** (教材习题 17 修改版) 设矩阵

$$A = \begin{bmatrix} \lambda & 2\lambda \\ 1 & 1 \end{bmatrix}, \quad \lambda \neq 0.$$

当 $\lambda$ 取何值时, $\mathrm{cond}_\infty(A)$ 达到最小.

**练习 5.11** (教材习题 18 修改版) 设矩阵

$$A = \begin{bmatrix} 10 & 9 \\ 9 & 8 \end{bmatrix}.$$

计算 $\mathrm{cond}_2(A)$ 和 $\mathrm{cond}_\infty(A)$.

# 第六讲 线性方程组迭代方法

直接法的运算量为 $O(n^3)$, 所以随着矩阵规模的增大, 直接法的运算量也随之快速增长. 对于大规模的线性方程组, 由于运算量太大, 直接法一般不再被采用, 取而代之的是迭代方法.

考虑线性方程组

$$Ax = b,$$

其中 $A \in \mathbb{R}^{n\times n}$ 非奇异. 迭代方法的基本思想: 给定一个迭代初始值 $x^{(0)}$, 通过一定的迭代格式生成一个迭代序列 $\{x^{(k)}\}_{k=0}^{\infty}$, 使得

$$\lim_{k\to\infty} x^{(k)} = x_* \triangleq A^{-1}b.$$

目前常用的迭代方法主要有两类, 一类是定常迭代方法, 如 Jacobi, Gauss-Seidel, SOR 等. 另一类是子空间迭代法, 如 CG, GMRES 等.

## 6.1 迭代法基本概念

### 6.1.1 向量序列与矩阵序列的收敛性

首先给出向量序列收敛的定义.

**定义 6.1 (向量序列的收敛)** 设 $\left\{x^{(k)}\right\}_{k=0}^{\infty}$ 是 $\mathbb{R}^n$ 中的一个向量序列. 如果存在向量 $x = [x_1, x_2, \ldots, x_n]^{\mathsf{T}} \in \mathbb{R}^n$ 使得

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

其中 $x_i^{(k)}$ 表示 $x^{(k)}$ 的第 $i$ 个分量. 则称 $\left\{x^{(k)}\right\}$ (按分量) 收敛到 $x$, 记为

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

♣ 我们称 $x$ 为序列 $\left\{x^{(k)}\right\}$ 的 极限.

相类似地, 我们可以给出矩阵序列收敛的定义.

**定义 6.2 (矩阵序列的收敛)** 设 $\left\{A^{(k)} = \left[a_{ij}^{(k)}\right]\right\}_{k=0}^{\infty}$ 是 $\mathbb{R}^{n\times n}$ 中的一个矩阵序列. 如果存在矩阵 $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ 使得

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

则称 $A^{(k)}$ 收敛到 $A$, 记为

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

我们称 $A$ 为 $A^{(k)}$ 的极限.

**例 6.1** 设 $0<|a|<1$, 考虑矩阵序列 $\{A^{(k)}\}$, 其中

$$A^{(k)}=\begin{bmatrix} a & 1 \\ 0 & a \end{bmatrix}^k, \quad k=1,2,\dots.$$

易知当 $k\to\infty$ 时, 有

$$A^{(k)}=\begin{bmatrix} a & 1 \\ 0 & a \end{bmatrix}^k=\begin{bmatrix} a^k & ka^{k-1} \\ 0 & a^k \end{bmatrix}\to\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

关于向量序列和矩阵序列的收敛性, 我们有下面的结论.

**定理 6.1** 设向量序列 $\{x^{(k)}\}_{k=0}^{\infty}\subset\mathbb{R}^n$, 矩阵序列 $\left\{A^{(k)}=\left[a_{ij}^{(k)}\right]\right\}_{k=0}^{\infty}\subset\mathbb{R}^{n\times n}$, 则

(1) $\lim\limits_{k\to\infty}x^{(k)}=x\iff\lim\limits_{k\to\infty}\|x^{(k)}-x\|=0$, 其中 $\|\cdot\|$ 为任一向量范数;

(2) $\lim\limits_{k\to\infty}A^{(k)}=A\iff\lim\limits_{k\to\infty}\|A^{(k)}-A\|=0$, 其中 $\|\cdot\|$ 为任一矩阵范数;

**证明.** 只需证明无穷范数情形即可, 其它范数可通过范数等价性来证明. □

由定理 6.1 可以立即得到下面的两个结论:

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

和

$$\lim_{k\to\infty}A^{k}=0\iff\lim_{k\to\infty}\|A^{k}\|=0$$

**定理 6.2** 设矩阵序列 $\left\{A^{(k)}=\left[a_{ij}^{(k)}\right]\right\}_{k=0}^{\infty}\subset\mathbb{R}^{n\times n}$, 则

$$\lim_{k\to\infty}A^{(k)}=0\iff\lim_{k\to\infty}A^{(k)}x=0,\quad\forall\,x\in\mathbb{R}^n.$$

**证明.** 先证明必要性 ("$\Longrightarrow$"). 由条件 $\lim\limits_{k\to\infty}A^{(k)}=0$ 可知, 对任意算子范数都有 $\lim\limits_{k\to\infty}\|A^{(k)}\|=0$. 因此, 对任意 $x\in\mathbb{R}^n$ 有

$$\|A^{(k)}x\|\le\|A^{(k)}\|\cdot\|x\|\to 0\ (k\to\infty)\quad\text{即}\quad\lim_{k\to\infty}A^{(k)}x=0.$$

下面证明充分性("$\Longrightarrow$"). 取 $x=e_i$, 即单位矩阵的第 $i$ 列, 则由 $\lim\limits_{k\to\infty}A^{(k)}e_i=0$ 可知, $A^{(k)}$ 的第 $i$ 列的极限为 0. 令 $i=1,2,\dots,n$, 则可得 $\lim\limits_{k\to\infty}A^{(k)}=0$. □

**定理 6.3** 设 $B\in\mathbb{R}^{n\times n}$, 若存在矩阵范数使得 $\|B\|<1$, 则 $\lim\limits_{k\to\infty}B^k=0$.

**证明.** 由条件 $\|B\| < 1$ 可知,

$$\|B^k\| \leq \|B\|^k \to 0 \ (k \to \infty).$$

所以 $\lim\limits_{k\to\infty} B^k = 0.$ □

**定理 6.4** 设 $B \in \mathbb{R}^{n\times n}$, 则 $\lim\limits_{k\to\infty} B^k = 0$ 当且仅当 $\rho(B) < 1$.

**证明.** 先证明必要性 ("$\Longrightarrow$"). 反证法. 假设 $\rho(B) \geq 1$, 则 $B$ 存在特征值 $\lambda$, 满足 $|\lambda| \geq 1$. 设其对应的特征向量为 $x \neq 0$, 即 $Bx = \lambda x$, 则 $B^k x = \lambda^k x$. 由于 $|\lambda| \geq 1$, 当 $k \to \infty$ 时 $\lambda^k x$ 不可能收敛到 0, 这与条件矛盾. 所以结论成立, 即 $\rho(B) < 1$.

下面证明充分性("$\Longrightarrow$"). 令 $\varepsilon = \frac{1}{2}\big(1 - \rho(B)\big)$, 则 $\varepsilon > 0$. 因此, 由定理 1.18 可知, 存在某个矩阵范数 $\|\cdot\|_\varepsilon$, 使得

$$\|B\|_\varepsilon \leq \rho(B) + \varepsilon = \frac{1}{2}(1 + \rho(B)) < 1.$$

所以由定理 6.3 可知 $\lim\limits_{k\to\infty} B^k = 0.$ □

**定理 6.5** 设 $B \in \mathbb{R}^{n\times n}$, 则 $\lim\limits_{k\to\infty} B^k = 0$ 当且仅当 $\rho(B) < 1$.

**证明.** 先证明必要性 ("$\Longrightarrow$"). 反证法. 假设 $\rho(B) \geq 1$, 则 $B$ 存在特征值 $\lambda$, 满足 $|\lambda| \geq 1$. 设其对应的特征向量为 $x \neq 0$, 即 $Bx = \lambda x$, 则 $B^k x = \lambda^k x$. 由于 $|\lambda| \geq 1$, 当 $k \to \infty$ 时 $\lambda^k x$ 不可能收敛到 0, 这与条件矛盾. 所以结论成立, 即 $\rho(B) < 1$.

下面证明充分性("$\Longrightarrow$"). 令 $\varepsilon = \frac{1}{2}\big(1 - \rho(B)\big)$, 则 $\varepsilon > 0$. 因此, 由定理 1.18 可知, 存在某个矩阵范数 $\|\cdot\|_\varepsilon$, 使得

$$\|B\|_\varepsilon \leq \rho(B) + \varepsilon = \frac{1}{2}(1 + \rho(B)) < 1.$$

所以由定理 6.3 可知 $\lim\limits_{k\to\infty} B^k = 0.$ □

**推论 6.6** 设 $B \in \mathbb{R}^{n\times n}$, 则 $\lim\limits_{k\to\infty} B^k = 0$ 的充要条件是存在某个矩阵范数 $\|\cdot\|$, 使得 $\|B\| < 1$.

下面的结论是谱半径与算子范数之间的一个非常重要的性质.

**引理 6.1** 设 $B \in \mathbb{R}^{n\times n}$, 则对任意矩阵范数 $\|\cdot\|$, 有

$$\rho(B) = \lim_{k\to\infty} \|B^k\|^{\frac{1}{k}}.$$

**证明.** 首先, 我们有

$$\big(\rho(B)\big)^k = \rho(B^k) \leq \|B^k\|.$$

另一方面, 对任意 $\varepsilon > 0$, 记

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

则 $\rho(B_\varepsilon)<1$, 故 $\lim\limits_{k\to\infty} B_\varepsilon^k=0$. 因此存在正整数 $N$, 使得当 $k>N$ 时, 有 $\|B_\varepsilon^k\|<1$, 即

$$\left\|\frac{B^k}{(\rho(B)+\varepsilon)^k}\right\|=\frac{\|B^k\|}{(\rho(B)+\varepsilon)^k}<1.$$

所以

$$\rho(B)\le\|B^k\|^{\frac{1}{k}}\le\rho(B)+\varepsilon.$$

由 $\varepsilon$ 的任意性可知, 当 $k\to\infty$ 时, 有 $\|B^k\|^{\frac{1}{k}}\to\rho(B)$. □

### 6.1.2 基于矩阵分裂的迭代法

当直接求解 $Ax=b$ 比较困难时, 我们可以求解一个近似线性方程组 $Mx=b$, 其中 $M$ 可以看作是 $A$ 在某种意义下的近似.

设 $Mx=b$ 的解为 $x^{(1)}$. 易知它与原方程的解 $x_*=A^{-1}b$ 之间的误差满足

$$A(x_*-x^{(1)})=b-Ax^{(1)}.$$

如果 $x^{(1)}$ 已经满足精度要求, 即非常接近真解 $x_*$, 则可以停止计算, 否则需要修正.

设 $\Delta x\triangleq x_*-x^{(1)}$, 则 $\Delta x$ 满足方程

$$A\Delta x=b-Ax^{(1)}.$$

但由于直接求解该方程比较困难, 因此我们还是通过求解近似方程

$$M\Delta x=b-Ax^{(1)},$$

得到一个近似的修正量 $\tilde{\Delta}x$. 于是修正后的近似解为

$$x^{(2)}=x^{(1)}+\tilde{\Delta}x=x^{(1)}+M^{-1}(b-Ax^{(1)}).$$

如果 $x^{(2)}$ 已经满足精度要求, 则停止计算, 否则继续按以上的方式进行修正.

不断重复以上步骤, 于是, 我们就得到一个向量序列

$$x^{(1)},x^{(2)},\ldots,,x^{(k)},\ldots.$$

它们都是真解 $x_*$ 的近似值, 且满足下面的递推关系

$$x^{(k+1)}=x^{(k)}+M^{-1}(b-Ax^{(k)}),\quad k=1,2,\ldots.$$

这就构成了一个迭代方法. 由于每次迭代的格式是一样的, 因此称为 定常迭代.

通常, 构造一个好的定常迭代, 需要考虑以下两点:

(1) 以 $M$ 为系数矩阵的线性方程组必须比原线性方程组更容易求解;

(2) $M$ 应该是 $A$ 的一个很好的近似, 且迭代序列 $\{x_k\}$ 收敛.

目前一类比较常用的定常迭代方法是基于矩阵分裂的迭代方法, 如 Jacobi 方法, Gauss-Seidel (G-S) 方法, 超松弛 (SOR, Successive Over-Relaxation) 方法等等. 下面给出矩阵分裂的定义.

**定义 6.3 (矩阵分裂 Matrix Splitting)** 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, 我们称

$$A = M - N \tag{6.1}$$

为 $A$ 的一个矩阵分裂, 其中 $M$ 非奇异.

给定一个矩阵分裂 (6.1), 则原方程组 $Ax = b$ 就等价于 $Mx = Nx + b$. 于是我们就可以构造出以下的迭代格式

$$Mx^{(k+1)} = Nx^{(k)} + b, \quad k = 0, 1, \dots, \tag{6.2}$$

或

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \triangleq Bx^{(k)} + f, \quad k = 0, 1, \dots, \tag{6.3}$$

其中 $B \triangleq M^{-1}N$ 称为迭代矩阵. 这就是基于矩阵分裂 (6.1) 的迭代方法. 易知, 选取不同的 $M$, 就可以构造出不同的迭代方法.

### 6.1.3 迭代法的收敛性

**定义 6.4** 设 $\{x^{(k)}\}$ 是由迭代方法 6.3 生成的向量序列, 如果 $\lim\limits_{k\to\infty} x^{(k)}$ 存在, 则称迭代方法 6.3 收敛, 否则就称为 发散

**引理 6.2** 设迭代序列 $\{x^{(k)}\}$ 收敛, 且 $\lim\limits_{k\to\infty} x^{(k)} = x_*$, 则 $x_*$ 一定是原方程组的真解.

证明. 对迭代格式 6.3 两边取极限可得

$$x_* = \lim_{k\to\infty} x^{(k+1)} = \lim_{k\to\infty} \left(M^{-1}Nx^{(k)} + M^{-1}b\right) = M^{-1}Nx_* + M^{-1}b.$$

整理后可得 $(M - N)x_* = b$, 即 $Ax_* = b$, 结论成立. □

下面给出迭代方法 6.3 的基本收敛性定理.

**定理 6.7** 对任意初始向量 $x^{(0)}$, 迭代方法 6.3 收敛的充要条件是 $\rho(B) < 1$.

证明. 先证明必要性 ("$\Longrightarrow$"). 对任意向量 $\tilde{x} \in \mathbb{R}^n$, 令 $x^{(0)} = \tilde{x} - x_*$, 则

$$x^{(k)} - x_* = (Bx^{(k-1)} + f) - (Bx_* + f) = B\left(x^{(k)} - x_*\right) = \cdots = B^k(x^{(0)} - x_*) = B^k\tilde{x}.$$

由迭代方法 6.3 的收敛性可知 $B^k\tilde{x} = x^{(k)} - x_* \to 0$ $(k \to \infty)$. 根据定理 6.2, $\lim\limits_{k\to\infty} B^k = 0$. 再由定理 6.4 可知 $\rho(B) < 1$.

下面证明充分性("$\Longrightarrow$"). 根据条件 $\rho(B) < 1$, 可得 $\lim\limits_{k\to\infty} B^k = 0$. 所以对任意 $x^{(0)} \in \mathbb{R}^n$, 当 $k \to \infty$ 时, 有

$$x^{(k)} - x_* = B^k(x^{(0)} - x_*) \to 0.$$

故 $\lim\limits_{k\to\infty} x^{(k)} = x_*$, 即迭代方法 6.3 收敛. □

由于对任意算子范数都有 $\rho(B) < \|B\|$, 因此我们可以立即得到下面的结论.

**定理 6.8** *若存在算子范数使得 $\|B\| < 1$, 则迭代方法 6.3 收敛*

♣ 由于计算 $\rho(B)$ 通常比较复杂, 而 $\|B\|_1$, $\|B\|_\infty$ 相对比较容易计算, 因此在判别迭代方法收敛性时, 可以先验算一下迭代矩阵的 1-范数或 $\infty$-范数是否小于 1.

♣ 定理 6.8 是充分条件, 但不是必要条件, 因此判断一个迭代方法不收敛仍然需要使用定理 6.7.

**例 6.2** 讨论迭代方法 $x^{(k+1)} = Bx^{(k)} + f$ 的收敛性, 其中 $B = \begin{bmatrix} 0.9 & 0 \\ 0.3 & 0.8 \end{bmatrix}$.

**解.** 由于 $B$ 是下三角矩阵, 因此其特征值分别为 $\lambda_1 = 0.9$, $\lambda_2 = 0.8$. 所以 $\rho(B) = 0.9 < 1$. 故迭代方法收敛. □

**定理 6.9** *若存在算子范数使得 $q \triangleq \|B\| < 1$, 则*

(1) $\|x^{(k)} - x_*\| \le q^k \|x^{(0)} - x_*\|$;

(2) $\|x^{(k)} - x_*\| \le \dfrac{q}{1-q} \|x^{(k)} - x^{(k-1)}\|$;

(3) $\|x^{(k)} - x_*\| \le \dfrac{q^k}{1-q} \|x^{(1)} - x^{(0)}\|$.

**证明.**

(1) 由 $x^{(k+1)} = Bx^{(k)} + g$ 和 $x_* = Bx_* + g$ 可得

$$x^{(k+1)} - x^{(k)} = B\left(x^{(k)} - x^{(k-1)}\right) \quad 和 \quad x^{(k+1)} - x_* = B\left(x^{(k)} - x_*\right).$$

因此有

$$\|x^{(k+1)} - x^{(k)}\| \le q \|x^{(k)} - x^{(k-1)}\|, \tag{6.4}$$

和

$$\|x^{(k+1)} - x_*\| \le q \|x^{(k)} - x_*\|. \tag{6.5}$$

反复利用结论 (6.5) 即可得 $\|x^{(k)} - x_*\| \le q^k \|x^{(0)} - x_*\|$.

(2) 由 (6.5) 可得

$$\begin{aligned} \|x^{(k+1)} - x^{(k)}\| &= \|(x^{(k+1)} - x_*) - (x^{(k)} - x_*)\| \\ &\ge \|x^{(k)} - x_*\| - \|x^{(k+1)} - x_*\| \\ &\ge (1-q)\|x^{(k)} - x_*\|. \end{aligned}$$

结合 (6.4) 可得

$$\|x^{(k)} - x_*\| \le \frac{1}{1-q} \|x^{(k+1)} - x^{(k)}\| \le \frac{q}{1-q} \|x^{(k)} - x^{(k-1)}\|.$$

(3) 反复利用 (6.4) 即可得

$$\|x^{(k+1)} - x^{(k)}\| \le q^k \|x^{(1)} - x^{(0)}\|.$$

所以

$$\|x^{(k)} - x_*\| \le \frac{1}{1-q} \|x^{(k+1)} - x^{(k)}\| \le \frac{q^k}{1-q} \|x^{(1)} - x^{(0)}\|.$$

□

### 6.1.4 迭代方法的收敛速度

考虑迭代方法 6.3, 第 $k$ 步的误差为

$$\varepsilon^{(k)} \triangleq x^{(k)} - x_* = B^k(x^{(0)} - x_*) = B^k\varepsilon^{(0)}.$$

所以

$$\frac{\left|\varepsilon^{(k)}\right|}{\left|\varepsilon^{(0)}\right|} \leq \|B^k\|.$$

因此平均每次迭代后误差的压缩率为 $\|B^k\|^{1/k}$.

**定义 6.5** 迭代方法 6.3 的**平均收敛速度**定义为

$$R_k(B) \triangleq -\ln\|B^k\|^{\frac{1}{k}},$$

**渐进收敛速度**定义为

$$R(B) \triangleq \lim_{k\to\infty} R_k(B) = -\ln\rho(B).$$

♣ 如果 $0 < \rho(B) < 1$, 则迭代方法 (6.2) 线性收敛.

♣ 一般来说, $\rho(B)$ 越小, 迭代方法 6.3 的收敛速度越快.

如果事先给定一个精度要求, 比如要求相对误差满足

$$\frac{\|x^{(k)} - x_*\|}{\|x^{(0)} - x_*\|} < \varepsilon,$$

则可根据下面的公式估计所需迭代步数 $k$:

$$\|B^k\| < \varepsilon \Longrightarrow \ln\|B^k\|^{1/k} \leq \frac{1}{k}\ln(\varepsilon) \Longrightarrow k \geq \frac{-\ln(\varepsilon)}{-\ln\|B^k\|^{1/k}} \approx \frac{-\ln(\varepsilon)}{R(B)}.$$

## 6.2 Jacobi, Gauss-Seidel 和 SOR

### 6.2.1 Jacobi 迭代方法

将矩阵 $A$ 分裂为

$$A = D - L - U,$$

其中 $D$ 为 $A$ 的对角线部分, $-L$ 和 $-U$ 分别为 $A$ 的严格下三角和严格上三角部分.

在矩阵分裂 $A = M - N$ 中取 $M = D$, $N = L + U$, 则可得 Jacobi 迭代 方法:

$$x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b, \quad k = 0, 1, 2, \ldots. \tag{6.6}$$

对应的迭代矩阵为

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

写成分量形式即为

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

♣ 由于 Jacobi 迭代中 $x_i^{(k+1)}$ 的更新顺序与 $i$ 无关, 即可以按顺序 $i = 1, 2, \ldots, n$ 计算, 也可以按顺序 $i = n, n-1, \ldots, 2, 1$ 计算, 或者乱序计算. 因此 Jacobi 迭代非常适合并行计算.

**算法 6.1.** Jacobi 迭代

1: Given an initial guess $x^{(0)}$
2: **while** not converge **do** % 停机准则
3: **for** $i = 1$ to $n$ **do**
4: $x_i^{(k+1)} = \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k)}\right) \Big/ a_{ii}$
5: **end for**
6: **end while**

♣ 在编程实现该算法时, "停机准则" 一般是要求相对残量满足一定的精度, 即

$$\frac{\|b - Ax^{(k)}\|}{\|b - Ax^{(0)}\|} < \text{tol},$$

其中 tol 是一个事前给定的精度要求, 如 $10^{-6}$ 或 $10^{-8}$, 等等.

我们有时也将 Jacobi 迭代格式写为

$$x^{(k+1)} = x^{(k)} + D^{-1}(b - Ax^{(k)}) = x^{(k)} + D^{-1}r_k, \quad k = 0, 1, 2, \ldots,$$

其中 $r_k \triangleq b - Ax^{(k)}$ 是 $k$ 次迭代后的残量. 这表明, $x^{(k+1)}$ 是通过对 $x^{(k)}$ 做一个修正得到的.

### 6.2.2 Gauss-Seidel 迭代方法

在分裂 $A = M - N$ 中取 $M = D - L$, $N = U$, 即可得 Gauss-Seidel (G-S) 迭代 方法:

$$x^{(k+1)} = (D - L)^{-1}Ux^{(k)} + (D - L)^{-1}b, \quad k = 0, 1, 2, \ldots. \tag{6.7}$$

对应的迭代矩阵为

$$G = (D - L)^{-1}U.$$

将 G-S 迭代改写为

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

即可得分量形式

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

**算法 6.2.** Gauss-Seidel 迭代

1: Given an initial guess $x^{(0)}$
2: **while** not converge **do**
3: **for** $i = 1$ to $n$ **do**
4: $x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right)$
5: **end for**
6: **end while**

♣ G-S 方法的主要优点是充分利用了已经获得的最新数据.

### 6.2.3 SOR 迭代方法

在 G-S 方法的基础上, 我们可以通过引入一个松弛参数 $\omega$ 来加快收敛速度. 这就是 SOR (Successive Overrelaxation) 方法. 该方法的基本思想是将 G-S 方法中的第 $k+1$ 步近似解与第 $k$ 步近似解做一个加权平均, 从而给出一个新的近似解, 即

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega D^{-1} \left( Lx^{(k+1)} + Ux^{(k)} + b \right). \tag{6.8}$$

整理后即为

$$x^{(k+1)} = (D - \omega L)^{-1} \left( (1-\omega)D + \omega U \right) x^{(k)} + \omega (D - \omega L)^{-1} b, \tag{6.9}$$

其中 $\omega$ 称为松弛参数.

- 当 $\omega = 1$ 时, SOR 即为 G-S 方法,
- 当 $\omega < 1$ 时, 称为低松弛方法,
- 当 $\omega > 1$ 时, 称为超松弛方法.

在大多数情况下, 当 $\omega > 1$ 时会取得比较好的收敛效果.

♣ SOR 方法曾经在很长一段时间内是科学计算中求解线性方程组的首选方法.

SOR 的迭代矩阵为

$$\mathcal{L}_\omega = (D - \omega L)^{-1} \left( (1-\omega)D + \omega U \right),$$

对应的矩阵分裂为

$$M = \frac{1}{\omega} D - L, \quad N = \frac{1-\omega}{\omega} D + U.$$

由 (6.8) 可知 SOR 迭代的分量形式为

$$\begin{aligned} x_i^{(k+1)} &= (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \\ &= x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right) \end{aligned}$$

**算法 6.3.** *求解线性方程组的 SOR 迭代方法*

1: Given an initial guess $x^{(0)}$ and parameter $\omega$
2: **while** not converge **do**
3: **for** $i = 1$ to $n$ **do**
4: $x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$
5: **end for**
6: **end while**

♣ SOR 方法最大的优点是引入了松弛参数 $\omega$: 通过选取适当的 $\omega$ 就可以大大提高方法的收敛速度.

♣ 如何确定 SOR 的最优松弛因子是一件非常困难的事!

**例 6.3** 分别用 Jacobi, G-S 和 SOR($\omega = 1.1$) 迭代方法求解线性方程组 $Ax = b$, 其中

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}.$$

初始向量设为 $x^{(0)} = [0, 0, 0]^\intercal$, 迭代过程中保留小数点后四位.

**解.** Jacobi 迭代方法: 迭代格式为

$$\begin{cases} x_1^{(k+1)} = \frac{1}{2}(1 + x_2^{(k)}) \\ x_2^{(k+1)} = \frac{1}{3}(8 + x_1^{(k)} + x_3^{(k)}) \\ x_3^{(k+1)} = \frac{1}{2}(-5 + x_2^{(k)}) \end{cases}$$

直接计算可得

$$x^{(1)} = [0.5000, 2.6667, -2.5000]^\intercal, \ \ldots, \ x^{(\mathbf{21})} = [2.0000, 3.0000, -1.0000]^\intercal.$$

G-S 迭代方法: 迭代格式为

$$\begin{cases} x_1^{(k+1)} = \frac{1}{2}(1 + x_2^{(k)}) \\ x_2^{(k+1)} = \frac{1}{3}(8 + x_1^{(k+1)} + x_3^{(k)}) \\ x_3^{(k+1)} = \frac{1}{2}(-5 + x_2^{(k+1)}) \end{cases}$$

直接计算可得

$$x^{(1)} = [0.5000, 2.8333, -1.0833]^\intercal, \ \dots, \ x^{(\mathbf{9})} = [2.0000, 3.0000, -1.0000]^\intercal.$$

SOR 迭代方法: 迭代格式为

$$\begin{cases} x_1^{(k+1)} = x_1^{(k)} + \omega\dfrac{1}{2}(1 - 2x_1^{(k)} + x_2^{(k)}) \\ x_2^{(k+1)} = x_2^{(k)} + \omega\dfrac{1}{3}(8 + x_1^{(k+1)} - 3x_2^{(k)} + x_3^{(k)}) \\ x_3^{(k+1)} = x_3^{(k)} + \omega\dfrac{1}{2}(-5 + x_2^{(k+1)} - 2x_3^{(k)}) \end{cases}$$

直接计算可得

$$x^{(1)} = [0.5500, 3.1350, -1.0257]^\intercal, \ \dots, \ x^{(\mathbf{7})} = [2.0000, 3.0000, -1.0000]^\intercal.$$

□

**例 6.4** 编程实践: 分别用 Jacobi, G-S 和 SOR($\omega = 1.5$) 迭代方法求解线性方程组 $Ax = b$, 其中

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

初始向量设为 $x^{(0)} = [0, 0, 0]^\intercal$.

**解.** MATLAB 程序见 `ex62.m` □

## 6.3 收敛性分析

根据迭代方法基本收敛性定理 6.7 和定理 6.8, 我们可以直接得到下面的结论:

- Jacobi 迭代收敛的 充要 条件: $\rho(J) < 1$
- G-S 迭代收敛的 充要 条件: $\rho(G) < 1$
- SOR 迭代收敛的 充要 条件: $\rho(\mathcal{L}_\omega) < 1$
- Jacobi 迭代收敛的 充分 条件: $\|J\| < 1$
- G-S 迭代收敛的充分条件 充分 条件: $\|G\| < 1$
- SOR 迭代收敛的充分条件 充分 条件: $\|\mathcal{L}_\omega\| < 1$

### 6.3.1 不可约与对角占优

**定义 6.6 (不可约)** 设 $A \in \mathbb{R}^{n\times n}$, 如果存在置换矩阵 $P$, 使得 $PAP^\intercal$ 为块上三角矩阵, 即

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

其中 $A_{11} \in \mathbb{R}^{k\times k}$ $(1 \le k < n)$, 则称 $A$ 为可约矩阵, 否则称为 不可约矩阵.

关于不可约矩阵, 我们由以下结论

**引理 6.3** 设 $A \in \mathbb{R}^{n\times n}$, 如果 $A$ 的所有元素都非零, 则 $A$ 不可约.

**引理 6.4** 设 $A \in \mathbb{R}^{n\times n}$, 且 $n \geq 2$. 如果 $A$ 可约, 则 $A$ 的零元素个数 $\geq n-1$.

以上两个结论可以直接验证. 下面给出一个比较有用的结论, 证明可参加相关资料, 这里不再给出.

**引理 6.5** 设 $A \in \mathbb{R}^{n\times n}$ 是三对角矩阵, 且三条对角线上的元素都非零, 则 $A$ 不可约.

思考 设 $A \in \mathbb{R}^{n\times n}$ 是三对角矩阵, 且上, 下次对角线元素均非零, 则 $A$ 是不是不可约?

**定义 6.7 (对角占优)** 设 $A \in \mathbb{R}^{n\times n}$, 若

$$|a_{ii}| \geq \sum_{j\neq i} |a_{ij}|$$

对所有 $i = 1, 2, \ldots, n$ 都成立, 且至少有一个不等式严格成立, 则称 $A$ 为 弱行对角占优, 简称 弱对角占优. 若对所有 $i = 1, 2, \ldots, n$ 不等式都严格成立, 则称 $A$ 是严格行对角占优, 简称 严格对角占优.

♣ 类似地, 可以定义弱列对角占优 和 严格列对角占优.

**定理 6.10** 若 $A \in \mathbb{R}^{n\times n}$ 是严格对角占优矩阵, 则 $A$ 非奇异.

证明. 我们使用反证法. 假设 $A$ 奇异, 即 $Ax = 0$ 存在非零解, 不妨设为 $x = [x_1, x_2, \ldots, x_n]^\intercal$. 不失一般性, 设下标 $k$ 满足 $\|x\|_\infty = |x_k|$, 则 $|x_k| > 0$. 考察 $Ax = 0$ 的第 $k$ 个方程:

$$a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n = 0.$$

可得

$$|a_{kk}| = \frac{1}{|x_k|}\left|\sum_{j=1,j\neq i}^{n} a_{kj}x_j\right| \leq \sum_{j=1,j\neq i}^{n} |a_{kj}| \cdot \frac{|x_j|}{|x_k|} \leq \sum_{j=1,j\neq i}^{n} |a_{kj}|,$$

这与 $A$ 严格对角占优矛盾. 所以 $A$ 非奇异. □

**定理 6.11** 设 $A \in \mathbb{R}^{n\times n}$ 是不可约的弱对角占优矩阵, 则 $A$ 非奇异.

证明. 略. 证明过程可参见相关资料. □

### 6.3.2 Jacobi 和 G-S 的收敛性

**定理 6.12** 设 $A \in \mathbb{R}^{n\times n}$, 若 $A$ 严格对角占优, 则 Jacobi 和 G-S 迭代方法都收敛.

证明. 由于 $A$ 严格行对角占优, 故 $\sum\limits_{j\neq i} |a_{ij}|/|a_{ii}| < 1$. 所以

$$\|J\|_\infty = \|D^{-1}(L+U)\|_\infty = \max_{1\leq i\leq n} \sum_{j\neq i} \frac{|a_{ij}|}{|a_{ii}|} < 1.$$

故 Jacobi 迭代方法收敛.

设 $\lambda$ 是 G-S 迭代矩阵 $G$ 的任意一个特征值, 即 $\det(\lambda I - G) = 0$. 于是

$$\det(\lambda(D-L)-U) = \det((D-L)^{-1}) \cdot \det(\lambda I - G) = 0.$$

若 $|\lambda| \geq 1$, 则有

$$|\lambda a_{ii}| > \sum_{j=1, j\neq i}^{n} |\lambda| \cdot |a_{ij}| \geq \sum_{j=1}^{i-1} |\lambda a_{ij}| + \sum_{j=i+1}^{n} |a_{ij}|, \quad i = 1, 2, \ldots, n.$$

因此 $\lambda(D-L)-U$ 是严格对角占优矩阵, 所以非奇异, 与 $\det(\lambda(D-L)-U) = 0$ 矛盾. 因此 $|\lambda| < 1$. 故 $\rho(G) < 1$, 即 G-S 迭代方法收敛. □

**定理 6.13** 设 $A \in \mathbb{R}^{n\times n}$, 若 $A$ 是不可约弱对角占优, 则 Jacobi 方法和 G-S 方法都收敛.

证明. 略. 请参见相关资料. □

**定理 6.14** 设 $A \in \mathbb{R}^{n\times n}$ 对称且 $D$ 正定, 则 Jacobi 收敛的充要条件是 $2D - A$ 正定.

证明. 略. 请参见相关资料. □

**定理 6.15** 设 $A \in \mathbb{R}^{n\times n}$ 对称且 $D$ 正定, 则 G-S 收敛的充要条件是 $A$ 正定.

证明. 略. 请参见相关资料. □

### 6.3.3 SOR 的收敛性

我们首先给出 SOR 迭代收敛的一个必要条件.

**定理 6.16** 若 SOR 迭代方法收敛, 则 $0 < \omega < 2$.

证明. SOR 的迭代矩阵为

$$\mathcal{L}_\omega = (D-\omega L)^{-1}\big((1-\omega)D + \omega U\big) = \left(I - \omega\tilde{L}\right)^{-1}\left((1-\omega)I + \omega\tilde{U}\right).$$

所以 $\mathcal{L}_\omega$ 的行列式为

$$\begin{aligned}\det\left(\mathcal{L}_\omega\right) &= \det\left((I-\omega\tilde{L})^{-1}\right) \cdot \det\left((1-\omega)I + \omega\tilde{U}\right) \\ &= \left(\det(I-\omega\tilde{L})\right)^{-1} \cdot (1-\omega)^n \\ &= (1-\omega)^n.\end{aligned}$$

设 $\mathcal{L}_\omega$ 的特征为 $\lambda_1, \lambda_2, \ldots, \lambda_n$, 则

$$\lambda_1\lambda_2\cdots\lambda_n = \det\left(\mathcal{L}_\omega\right) = (1-\omega)^n,$$

故至少有一个特征值的绝对值不小于 $|1-\omega|$, 因此 $\rho(\mathcal{L}_\omega) \geq |1-\omega|$.

若 SOR 收敛, 则 $\rho(\mathcal{L}_\omega) < 1$, 所以 $|1-\omega| < 1$, 即 $0 < \omega < 2$. □

当 $A$ 对角占优时, 我们有下面的结论.

**定理 6.17** 设 $A \in \mathbb{R}^{n\times n}$, 若 $A$ 严格对角占优 (或不可约弱对角占优) 且 $0 < \omega \leq 1$, 则 SOR 收敛.

证明. 略. 请参见相关资料. □

如果 $A$ 是对称正定矩阵, 则有下面的收敛性定理.

**定理 6.18** 设 $A \in \mathbb{R}^{n\times n}$ 对称正定, 则 SOR 迭代方法收敛的充要条件是 $0 < \omega < 2$.

证明. *To be continued ...* □

**例 6.5** 考虑线性方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{bmatrix}$. 试给出 Jacobi, G-S 和 SOR 收敛的充要条件.

解. *To be continued ...* □

## 6.4 共轭梯度法

### 6.4.1 最速下降法

### 6.4.2 共轭梯度法

## 6.5 本章小结

## 6.6 课后练习

**练习 6.1** (教材习题 5, 有修改) 已知线性方程组 $\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$, 迭代方法

$$x^{(k+1)} = x^{(k)} + \alpha(Ax^{(k)} - b), \quad k = 0, 1, 2 \dots.$$

试问: $\alpha$ 取何值时迭代收敛? 何时收敛最快?

**练习 6.2** (教材习题 9, 有修改) 设线性方程组 $Ax = b$, 其中 $A$ 对称正定且 $0 < \alpha \leq \lambda(A) \leq \beta$, 迭代方法

$$x^{(k+1)} = x^{(k)} + \omega(b - Ax^{(k)}), \quad k = 0, 1, 2 \dots.$$

试证明: 当 $0 < \omega < \frac{2}{\beta}$ 时, 迭代方法收敛.

**练习 6.3** (教材习题 1 (1), 有修改) 已知线性方程组

$$\begin{cases} 5x_1 + 2x_2 + x_3 = 3, \\ -2x_1 + 4x_2 + 2x_3 = 7, \\ 3x_1 - 5x_2 + 8x_3 = 2. \end{cases}$$

考察 Jacobi 方法和 Gauss-Seidel 方法的收敛性.

**练习 6.4** (教材习题 2, 有修改) 已知线性方程组

$$(1) \begin{cases} x_1 + 0.4x_2 + 0.5x_3 = 2, \\ 0.4x_1 + x_2 + 0.8x_3 = 1, \\ 0.5x_1 + 0.8x_2 + x_3 = 3. \end{cases} \qquad (2) \begin{cases} 2x_1 + 4x_2 - 3x_3 = 3, \\ x_1 + 2x_2 + x_3 = 1, \\ 2x_1 + 2x_2 + 2x_3 = 2. \end{cases}$$

考察 Jacobi 方法和 Gauss-Seidel 方法的收敛性.

**练习 6.5** (教材习题 3, 有修改) 设线性方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1, \\ a_{21}x_1 + a_{22}x_2 = b_2, \end{cases}$$

其中 $a_{11}a_{22} \neq 0$. 证明: 求解该线性方程组的 Jacobi 迭代和 G-S 迭代具有相同的收敛性, 并求两种方法的收敛速度之比.

**练习 6.6** (教材习题 4, 有修改) 已知线性方程组 $Ax = f$, 其中

$$A = \begin{bmatrix} 8 & a & 0 \\ b & 8 & b \\ 0 & a & 4 \end{bmatrix}$$

非奇异, 试给出 Jacobi 方法和 Gauss-Seidel 方法收敛的充要条件.

**练习 6.7** (补充题) 已知线性方程组:

$$\begin{bmatrix} 5 & 2 & 2 \\ 2 & 5 & 4 \\ 2 & 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ 7 \end{bmatrix}.$$

写出 Jacobi, Gauss-Seidel 和 SOR($\omega = 1.5$) 方法的分量格式, 并判断这三个方法的收敛性.

# 参考文献

[1] R.A. Horn and C.R. Johnson, *Matrix Analysis*, 2nd edition, Cambridge University Press, New York, 2013. Cited on page 10.

[2] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991. Cited on page 10.