计算方法

王新茂

中国科学技术大学数学科学学院

http://mcm.ustc.edu.cn/∼xwang/jsff.pdf

2023 年 3 月 10 日

目录

第一章　简介 (2 学时)

§1.1　课程安排

- 《计算方法》课程主要介绍一些常见数学问题的数值求解算法，介绍算法的数学原理，分析算法的准确性、复杂性和稳定性.
- 针对具体的实际问题，需要选取恰当的求解方法. 学生有必要掌握相关算法的数学原理.
- 为了更好地理解算法，感受算法的实际效果，学生也需要在计算机上编程实现算法.
- 受实现方式的影响，算法的实际效果通常与理论效果有较大差距.
- 课程计划授课 15 周，每周 2 次，每次 2 节课.
- 总评成绩 $= f($平时作业成绩 $* 20\% +$ 期末考试成绩 $* 80\%) +$ 调整，f 单调递增.

- 平时作业为教材中的习题和附录 1．可根据个人情况自选．禁止抄袭．
- 期末考试为闭卷考试．可以使用计算器．
- 遇到数学问题或有学习困难，请主动联系授课教师和助教寻求帮助．

教材与参考书

- 数值计算方法与算法（第四版），张韵华、王新茂、陈效群、张瑞著，科学出版社，2022 年 7 月．
- 数值计算方法扩充教程，童伟华编，
  http://staff.ustc.edu.cn/~tongwh/NA_2023/slides/book.pdf.
- 国外数学名著系列（影印版）5：Numerical Mathematics，Alfio Quarteroni, Riccardo Sacco, Fausto Saleri 著，科学出版社，2006 年 1 月．
- 科学计算导论（第 2 版），Michael T. Heath 著，张威、贺华、冷爱萍译，清华大学出版社，2005 年 10 月．
- 数值分析（原书第 3 版），David Kincaid, Ward Cheney 著，王国荣、俞耀明、徐兆亮译，机械工业出版社，2005 年 9 月．

§1.2 基本概念

§1.2.1 准确性

定义 1.1. 通常用计算结果的**误差**来定量衡量算法的准确性.

$$\textbf{绝对误差} = \text{近似值} - \text{真实值}, \quad \textbf{相对误差} = \frac{\text{绝对误差}}{\text{真实值}}.$$

可能产生误差的原因有很多，大致可分为以下两类.

- **系统误差**. 例如：对实际问题的建模带来的误差、输入数据带来的误差、算法本身带来的误差.
- **随机误差**. 例如：计算机浮点运算带来的舍入误差.

例 1.1. IEEE-754 是 1980 年代以来最广泛使用的浮点数运算标准，为许多计算机系统所采用.

每个单精度 float 浮点数 $x = y \cdot 2^{e-23}$ 占用 32bit 内存，其中 $1 - 2^{23} \leqslant y \leqslant 2^{23}$ 占用 24bit，$-127 \leqslant e \leqslant 128$ 占用 8bit. $2^{-23} \approx 1.19 \cdot 10^{-7}$，单精度计算 $1 + 5 \cdot 10^{-8}$ 的结果是 1.

每个双精度 double 浮点数 $x = y \cdot 2^{e-52}$ 占用 64bit 内存，其中 $1 - 2^{52} \leqslant y \leqslant 2^{52}$ 占用 53bit，$-1023 \leqslant e \leqslant 1024$ 占用 11bit. $2^{-52} \approx 2.2 \cdot 10^{-16}$，双精度计算 $1 + 10^{-16}$ 的结果是 1. □

定义 1.2. 设 x 是有限位的十进制小数. 从 x 的首个非零位到末位的数字称为 x 的**有效数字**，有效数字的个数称为**有效位数**. 在保留 x 的 k 位有效数字时，需要对第 $k+1$ 位有效数字作四舍五入.

例如，0.0100 有 3 位有效数字，float 浮点数的有效位数 ≈ 8，double 浮点数的有效位数 ≈ 16，π 保留 5 位有效数字，得 3.1416．在计算过程中使用更高的有效位数有助于减少误差、增强算法的稳定性．

§1.2.2 复杂性

定义 1.3. 通常用编程实现算法并运行程序时，所需要消耗资源的数量来衡量算法的复杂性，主要包括**空间复杂度** (存储空间) 和**时间复杂度** (运行时间)．算法的复杂性是评价算法优劣的一项重要标准．在衡量误差、复杂度等量的时候，我们常使用记号 $\sim, O, o$ 表示函数的阶．设 $g(x)$ 是正值函数．考虑 $x \to a$ 的情况，a 可以是常数或 $\pm\infty$．

表达式	$f(x) \sim g(x)$	$f(x) = O(g(x))$	$f(x) = o(g(x))$
含义	$\lim\limits_{x\to a} \dfrac{f(x)}{g(x)} = 1$	$\overline{\lim\limits_{x\to a}} \left\lvert\dfrac{f(x)}{g(x)}\right\rvert < \infty$	$\lim\limits_{x\to a} \dfrac{f(x)}{g(x)} = 0$

例 1.2. 给定正整数 n 和实数 $a_0, a_1, \cdots, a_n, x$．下面是计算 $S = a_0 + a_1 x + \cdots + a_n x^n$ 的两个算法．

算法1: `s = a[0]; t = x; for(i=1; i<=n; i++) { s += a[i] * t; t *= x; }`

算法2: `s = a[n]; for(i=n-1; i>=0; i--) s = s * x + a[i];`

不考虑程序代码所占用的存储空间．假设每个整数/实数均占用 32bit = 4byte 的内存．算法 1 占用了 $4(n+6)$ byte 内存，算法 2 占用了 $4(n+5)$ byte 内存．算法 1 和算法 2 的空间复杂度都为 $O(n)$．

算法的时间复杂度可以用算法中各种基本运算的次数来衡量．假设不考虑赋值运算、逻辑运算和 `for(...)` 中的运算．算法 1 包含了 $2n$ 次加法、$2n$ 次乘法．算法 2 包含了 n 次加法、n 次减法、n 次乘法．时间复杂度都为 $O(n)$．算法 2 比算法 1 快． □

§1.2.3 稳定性

定义 1.4. 算法的**稳定性**指的是计算过程中的各种扰动（如输入误差、舍入误差等）对于计算结果的影响程度．算法的稳定性也是评价算法优劣的一项重要标准．

算法的稳定性与其数学问题的敏感性有紧密联系．数学问题的**敏感性**指的是问题中各种参数的扰动对于解的影响程度．分析算法的稳定性有助于找出参数的适用范围．

例 1.3. 输入实数 $a \geqslant -0.25$，输出方程 $ax^2 + x - 1 = 0$ 的最小正实根 r．

算法1: `if(a==0) r = 1; else r = (sqrt(0.25+a)-0.5) / a;`

算法2: `r = 1 / (sqrt(0.25+a)+0.5);`

当 $a \approx 0$ 时，算法 1 的结果有较大误差，算法 1 是不稳定的．算法 2 始终是稳定的． □

例 1.4. 输入实数 b，输出方程 $x^2+bx-1=0$ 的正实根 r.

算法1: `r = (sqrt(4+b*b)-b) / 2;`

算法2: `r = 2 / (sqrt(4+b*b)+b);`

当 $b\to+\infty$ 时，算法 1 的结果有较大误差，算法 1 是不稳定的，算法 2 是稳定的.

当 $b\to-\infty$ 时，算法 2 的结果有较大误差，算法 2 是不稳定的，算法 1 是稳定的. □

例 1.5. 考虑递推数列 $x_0=1$，$x_1=\dfrac{1}{3}$，$x_n=\dfrac{1}{3}(13x_{n-1}-4x_{n-2})$，$n\geqslant 2$. 理论上，数列的通项公式 $x_n=\dfrac{1}{3^n}$. 在实际计算中，舍入误差被不断放大，导致 x_n 的误差越来越大. 故算法是不稳定的. □

§1.2.4 范数

在矩阵计算问题中，经常利用向量和矩阵的范数对误差进行估计.

定义 1.5. 设向量 $\alpha=(a_i)\in\mathbb{C}^n$，矩阵 $A=(a_{ij})\in\mathbb{C}^{m\times n}$.

1. $\|\alpha\|_p=\left(\sum\limits_{i=1}^{n}|a_i|^p\right)^{\frac{1}{p}}$ 称为 α 的 p **范数**，其中 $p\geqslant 1$.
2. $\|\alpha\|_\infty=\lim\limits_{p\to+\infty}\|\alpha\|_p=\max\limits_{1\leqslant i\leqslant n}|a_i|$ 称为 α 的 ∞ **范数**.

3. $\|A\|_F = \sqrt{\mathrm{tr}(A^H A)} = \sqrt{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}|a_{ij}|^2}$ 称为 A 的 **Frobenius 范数**. (视 A 为向量)

4. $\|A\|_p = \max\limits_{\alpha\neq\mathbf{0}} \dfrac{\|A\alpha\|_p}{\|\alpha\|_p}$ 称为 A 的 p **范数**，其中 $p \geqslant 1$ 或 $p = +\infty$. (视 A 为变换)

5. $\mathrm{cond}(A) = \|A\|\ \|A^{-1}\|$ 称为可逆方阵 A 的**条件数**，其中 $\|\cdot\|$ 是某种矩阵范数.

对于一个连续函数 $f : [a, b] \to \mathbb{R}$，也可以同上类似地定义其范数

$$\|f\|_p = \left(\int_a^b |f(x)|^p dx\right)^{\frac{1}{p}},\ \ p \geqslant 1, \qquad \|f\|_\infty = \max_{a\leqslant x\leqslant b} |f(x)|.$$

定义 1.6. 一般地，设 V 是实数域 $\mathbb{R}$ 上的线性空间. 具有下列性质的映射 $\|\cdot\| : V \to \mathbb{R}$ 称为 V 上的**范数**，$(V, \|\cdot\|)$ 称为**赋范线性空间**.

- 非负性：$\|\alpha\| > 0$，$\forall \alpha \in V$，$\alpha \neq \mathbf{0}$.
- 齐次性：$\|\lambda\alpha\| = |\lambda|\ \|\alpha\|$，$\forall \lambda \in \mathbb{R}$，$\alpha \in V$.
- 三角不等式：$\|\alpha + \beta\| \leqslant \|\alpha\| + \|\beta\|$，$\forall \alpha, \beta \in V$.

定理 1.1. 向量和矩阵的范数具有下列性质，其中 $x=(x_i)\in\mathbb{C}^n$，$y=(y_i)\in\mathbb{C}^n$，$A=(a_{ij})\in\mathbb{C}^{m\times n}$，$p,q\in[1,+\infty]$.

1. $\|x+y\|_p\leqslant\|x\|_p+\|y\|_p$.
2. 当 $p<q$ 时，$\|x\|_p\geqslant\|x\|_q$，$\dfrac{\|x\|_p}{\sqrt[p]{n}}\leqslant\dfrac{\|x\|_q}{\sqrt[q]{n}}$.
3. 设 $\dfrac{1}{p}+\dfrac{1}{q}=1$，则 $|x\cdot y|\leqslant\|x\|_p\ \|y\|_q$.
4. $\|A+B\|_F\leqslant\|A\|_F+\|B\|_F$，　$\|A+B\|_p\leqslant\|A\|_p+\|B\|_p$.　$\|AB\|_p\leqslant\|A\|_p\ \|B\|_p$.
5. $\|A\|_2=\sigma_1(A)=\sqrt{\lambda_1(A^HA)}$. 特别，若 A 是酉方阵，则 $\|A\|_2=1$.
6. $\|A\|_1=\max\limits_{1\leqslant j\leqslant n}\left(\sum\limits_{i=1}^{m}|a_{ij}|\right)$，　$\|A\|_\infty=\max\limits_{1\leqslant i\leqslant m}\left(\sum\limits_{j=1}^{n}|a_{ij}|\right)$.
7. $\dfrac{\|A\|_1}{\sqrt{m}}\leqslant\|A\|_2\leqslant\sqrt{n}\|A\|_1$，　$\dfrac{\|A\|_\infty}{\sqrt{n}}\leqslant\|A\|_2\leqslant\sqrt{m}\|A\|_\infty$.
8. $\|A\|_2\leqslant\|A\|_F\leqslant\sqrt{r}\|A\|_2$，其中 $r=\mathrm{rank}(A)$.
9. 设 $\dfrac{1}{p}+\dfrac{1}{q}=1$，则 $\|A^H\|_p=\|A\|_q$ 并且 $\|A\|_2\leqslant\sqrt{\|A\|_p\ \|A\|_q}$.

证明.

1. $\sum\limits_{i=1}^{n}|x_i+y_i|^p \leqslant \sum\limits_{i=1}^{n}|x_i|\cdot|x_i+y_i|^{p-1}+\sum\limits_{i=1}^{n}|y_i|\cdot|x_i+y_i|^{p-1}$. 根据结论 3，

$$\sum_{i=1}^{n}|x_i+y_i|^p \leqslant \Big(\sum_{i=1}^{n}|x_i|^p\Big)^{\frac{1}{p}}\Big(\sum_{i=1}^{n}|x_i+y_i|^{(p-1)q}\Big)^{\frac{1}{q}}+\Big(\sum_{i=1}^{n}|y_i|^p\Big)^{\frac{1}{p}}\Big(\sum_{i=1}^{n}|x_i+y_i|^{(p-1)q}\Big)^{\frac{1}{q}}$$

即 $\Big(\|x+y\|_p\Big)^p \leqslant \Big(\|x\|_p+\|y\|_p\Big)\Big(\|x+y\|_p\Big)^{\frac{p}{q}}$. 由此可得，$\|x+y\|_p \leqslant \|x\|_p+\|y\|_p$.

2. 设 $f(t)=\Big(|x_1|^t+\cdots+|x_n|^t\Big)^{\frac{1}{t}}$，其中 $x_i\neq 0$，$\forall i$. 由对数函数的凹凸性，得

$$\begin{aligned}\frac{f'(t)}{f(t)} &= \frac{1}{t}\frac{|x_1|^t\ln|x_1|+\cdots+|x_n|^t\ln|x_n|}{|x_1|^t+\cdots+|x_n|^t}-\frac{1}{t^2}\ln\Big(|x_1|^t+\cdots+|x_n|^t\Big)\\ &\leqslant \frac{1}{t^2}\ln\frac{|x_1|^{2t}+\cdots+|x_n|^{2t}}{|x_1|^t+\cdots+|x_n|^t}-\frac{1}{t^2}\ln\Big(|x_1|^t+\cdots+|x_n|^t\Big)=\frac{1}{t^2}\ln\frac{|x_1|^{2t}+\cdots+|x_n|^{2t}}{\Big(|x_1|^t+\cdots+|x_n|^t\Big)^2}\leqslant 0.\end{aligned}$$

因此，$f(t)$ 在 $(0,+\infty)$ 上单调递减.

设 $g(t)=\Big(\frac{|x_1|^t+\cdots+|x_n|^t}{n}\Big)^{\frac{1}{t}}$，其中 $x_i\neq 0$，$\forall i$.

$$\frac{g'(t)}{g(t)}=\frac{1}{t}\frac{|x_1|^t\ln|x_1|+\cdots+|x_n|^t\ln|x_n|}{|x_1|^t+\cdots+|x_n|^t}-\frac{1}{t^2}\ln\frac{|x_1|^t+\cdots+|x_n|^t}{n}=\frac{G(t)}{t^2}.$$

根据 Cauchy 不等式，

$$G'(t)=t\frac{|x_1|^t\ln^2|x_1|+\cdots+|x_n|^t\ln^2|x_n|}{|x_1|^t+\cdots+|x_n|^t}-t\left(\frac{|x_1|^t\ln|x_1|+\cdots+|x_n|^t\ln|x_n|}{|x_1|^t+\cdots+|x_n|^t}\right)^2$$
$$=\frac{t}{(|x_1|^t+\cdots+|x_n|^t)^2}\left(\left(\sum_{i=1}^n|x_i|^t\right)\left(\sum_{i=1}^n|x_i|^t\ln^2|x_i|\right)-\left(\sum_{i=1}^n|x_i|^t\ln|x_i|\right)^2\right)\geqslant 0.$$

故 $G(t)\geqslant G(0)=0$．因此，$g(t)$ 在 $(0,+\infty)$ 上单调递增．

3. 由对数函数的凹凸性，得 $\ln(ab)=\dfrac{\ln a^p}{p}+\dfrac{\ln b^q}{q}\leqslant\ln\Big(\dfrac{a^p}{p}+\dfrac{b^q}{q}\Big)$，$\forall a,b>0$．设 $a=\dfrac{|x_i|}{\|x\|_p}$，$b=\dfrac{|y_i|}{\|y\|_q}$，得 $\dfrac{|x_iy_i|}{\|x\|_p\cdot\|y\|_q}\leqslant\dfrac{1}{p}\Big(\dfrac{|x_i|}{\|x\|_p}\Big)^p+\dfrac{1}{q}\Big(\dfrac{|y_i|}{\|y\|_q}\Big)^q$．从而，$|x\cdot y|\leqslant\sum\limits_{i=1}^n|x_iy_i|\leqslant\|x\|_p\cdot\|y\|_q$．

4. 矩阵范数的定义结合结论 1 的推论．

5. 设 $A=P\Sigma Q$ 是奇异值分解，其中 P,Q 是酉方阵，$\Sigma=\operatorname{diag}(\sigma_1,\cdots,\sigma_r,O)$，$\sigma_1\geqslant\cdots\geqslant\sigma_r>0$．

$$\|A\|_2=\max_{x^Hx=1}\sqrt{x^HA^HAx}=\max_{y^Hy=1}\sqrt{y^H\Sigma^H\Sigma y}=\sigma_1,\qquad y=Qx.$$

由 $A^HA=Q^H\operatorname{diag}(\sigma_1^2,\cdots,\sigma_r^2)Q$，得 $\sigma_1=\sqrt{\lambda_1(A^HA)}$．

6. 设 $C_1=\max\limits_{1\leqslant j\leqslant n}\Big(\sum\limits_{i=1}^m|a_{ij}|\Big)=\sum\limits_{i=1}^m|a_{ik}|$，则 $\|A\|_1=\max\limits_{\|x\|_1=1}\|Ax\|_1=\max\limits_{\|x\|_1=1}\sum\limits_{i=1}^m\Big|\sum\limits_{j=1}^n a_{ij}x_j\Big|\leqslant$

$\max\limits_{\|x\|_1=1} \sum\limits_{j=1}^{n}\sum\limits_{i=1}^{m} |a_{ij}|\,|x_j| \leqslant C_1$. 当 $x=\mathbf{e}_k$ 时，$\|Ax\|_1 = C_1$.

设 $C_2 = \max\limits_{1\leqslant i\leqslant m}\Big(\sum\limits_{j=1}^{n}|a_{ij}|\Big) = \sum\limits_{j=1}^{n}|a_{kj}|$，则 $\|A\|_\infty = \max\limits_{\|x\|_\infty=1}\|Ax\|_\infty = \max\limits_{\|x\|_\infty=1}\max\limits_{1\leqslant i\leqslant m}\left|\sum\limits_{j=1}^{n}a_{ij}x_j\right|$

$\leqslant \max\limits_{\|x\|_\infty=1}\max\limits_{1\leqslant i\leqslant m}\sum\limits_{j=1}^{n}|a_{ij}| \leqslant C_2$. 当 $x=\left(\frac{\overline{a_{k1}}}{|a_{k1}|},\cdots,\frac{\overline{a_{kn}}}{|a_{kn}|}\right)^T$ 时，$\|Ax\|_\infty = C_2$.

7. 一方面，存在 k 使得 $\|A\|_1 = \|A\mathbf{e}_k\|_1 \leqslant \sqrt{m}\|A\mathbf{e}_k\|_2 \leqslant \sqrt{m}\|A\|_2$. 另一方面，存在 $\|x\|_2=1$ 使得 $\|A\|_2 = \|Ax\|_2 \leqslant \|Ax\|_1 \leqslant \|A\|_1\|x\|_1 \leqslant \sqrt{n}\|A\|_1$. 综上，$\dfrac{\|A\|_1}{\sqrt{m}} \leqslant \|A\|_2 \leqslant \sqrt{n}\|A\|_1$.

   同理，$\dfrac{\|A^H\|_1}{\sqrt{n}} \leqslant \|A^H\|_2 \leqslant \sqrt{m}\|A^H\|_1$，即 $\dfrac{\|A\|_\infty}{\sqrt{n}} \leqslant \|A\|_2 \leqslant \sqrt{m}\|A\|_\infty$.

8. 设 $A=P\Sigma Q$ 是奇异值分解，则 $\|A\|_F = \sqrt{\mathrm{tr}(A^HA)} = \sqrt{\sigma_1^2+\cdots+\sigma_r^2}$. 故 $\sigma_1 \leqslant \|A\|_F \leqslant \sqrt{r}\sigma_1$.

9. 设 $\|x\|_p = \|y\|_q = 1$. 由结论 3，$|x^HAy| \leqslant \|A^Hx\|_p \leqslant \|A^H\|_p$，等号可取到. 同理，$|x^HAy| \leqslant \|Ay\|_q \leqslant \|A\|_q$，等号亦可取到. 故 $\|A^H\|_p = \|A\|_q = \max\limits_{\|x\|_p=\|y\|_q=1}|x^HAy|$.

   设非零向量 z 使得 $A^HAz = \sigma_1^2 z$，则 $\sigma_1^2\|z\|_p = \|A^HAz\|_p \leqslant \|A^H\|_p\,\|A\|_p\,\|z\|_p$. 故 $\|A\|_2 = \sigma_1 \leqslant \sqrt{\|A\|_p\,\|A\|_q}$. □

例 1.6. 考虑线性方程组 $Ax = b$ 的扰动问题. 设 $\|\cdot\|$ 是任意 p 范数. 结合 $\|b\| \leqslant \|A\| \cdot \|x\|$，得

$$Ax = b \quad \Rightarrow \quad (\delta A)x + A(\delta x) \approx \delta b \quad \Rightarrow \quad \delta x \approx A^{-1}(\delta b) - A^{-1}(\delta A)x$$

$$\Rightarrow \quad \|\delta x\| \leqslant \|A^{-1}\| \cdot \|\delta b\| + \|A^{-1}\| \cdot \|\delta A\| \cdot \|x\| \quad \Rightarrow \quad \frac{\|\delta x\|}{\|x\|} \leqslant \mathrm{cond}(A) \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right).$$

□

方阵的特征值与范数之间还存如下联系.

定义 1.7. $A \in \mathbb{C}^{n\times n}$ 的**谱半径**

$$\rho(A) = \max(|\lambda_1|, \cdots, |\lambda_n|)$$

其中 $\lambda_1, \cdots, \lambda_n$ 是 A 的所有特征值.

定理 1.2. 关于 $A \in \mathbb{C}^{n\times n}$ 的谱半径，有下列常用结论.

(1) 对于任意矩阵范数 $\|\cdot\|$，$\rho(A) \leqslant \|A\|$. *(2)* $\lim\limits_{k\to+\infty} A^k = O$ 当且仅当 $\rho(A) < 1$.

证明. (1) 设 λ, α 是 A 的任意特征值及相应的特征向量，由 $A\alpha = \lambda\alpha$，得 $|\lambda|\|\alpha\| = \|A\alpha\| \leqslant \|A\|\,\|\alpha\|$，故 $\rho(A) \leqslant \|A\|$. (2) 存在可逆方阵 P 使得 $P^{-1}AP = \mathrm{diag}\Big(J_{n_1}(\lambda_1), \cdots, J_{n_k}(\lambda_k)\Big)$ 是 Jordan 标准形. 故只需考虑 $A = J_n(\lambda) = \lambda I_n + N$ 情形，$N = \begin{pmatrix} & I_{n-1} \\ 0 & \end{pmatrix}$. 当 $k \to +\infty$ 时，$A^k = \sum\limits_{i=0}^{n} C_k^i \lambda^{k-i} N^i \to O$ 当且仅当 $|\lambda| < 1$. □

课堂练习

1. 设 $A=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$，则 $\|A\|_F=$ _____，$\|A\|_1=$ _____，$\|A\|_2=$ _____，$\|A\|_\infty=$ _____，$\rho(A)=$ _____.

2. 设用公式 $\left(1+\frac{1}{n}\right)^n$ 计算 e 的近似值，并且希望误差的绝对值不超过 0.001．如何选取 n？每步计算保留多少位有效数字？请给出你的算法.

第二章　数值逼近 (12 学时)

§2.1　插值

定义 2.1. 给定函数族 Φ 和平面点列 $(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n)$. 若函数 $\phi \in \Phi$ 满足

$$\phi(x_i) = y_i, \quad \forall i$$

则 ϕ 称为 Φ 中关于**插值点** $\{(x_i, y_i) \mid 0 \leqslant i \leqslant n\}$ 的**插值函数**，$x_0, x_1, \cdots, x_n$ 称为**插值节点**. 例如，

- 多项式插值：$\Phi = \mathbb{R}[x]$
- Hermite 插值：求 $\phi \in \mathbb{R}[x]$ 满足 $\phi(x_i) = y_i$，$\phi'(x_i) = z_i, \cdots$
- k 次样条插值：$\Phi = \{$分段光滑且每段是次数不超过 k 的多项式$\}$
- 三角多项式插值：$\Phi = \left\{ \sum\limits_{k=0}^{n} a_k \cos(kx) + \sum\limits_{k=1}^{n} b_k \sin(kx) \mid a_k, b_k \in \mathbb{R},\ n \in \mathbb{N} \right\}$.

利用未知连续函数 $f(x)$ 在若干节点处的值，构造 $f(x)$ 关于这些节点的插值函数 $\phi(x)$，作为 $f(x)$ 的近似，从而得到 $f(x)$ 在其他点处的近似值．这就是**插值**的字面含义．

§2.1.1 多项式插值

定理 2.1. 设实数 $x_0, x_1, \cdots, x_n$ 两两不同，则对于任意实数 $y_0, y_1, \cdots, y_n$，存在唯一的 $\phi \in \mathbb{R}[x]$ 满足 $\deg(\phi) \leqslant n$ 并且 $\phi(x_i) = y_i$，$\forall i = 0, 1, \cdots, n$.

证明. 设 $\phi(x) = c_0 + c_1 x + \cdots + c_n x^n$，线性方程组

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{2.1}$$

有唯一解 $(c_0, c_1, \cdots, c_n)$. □

通常可作线性预处理 $\hat{x}_i = ax_i + b$，$\hat{y}_i = cy_i + d$ 以改善线性方程组 (2.1) 的求解结果．

定理 2.2. 设 x_i, y_i, ϕ 同定理 *2.1*，则有 **Lagrange 插值公式**

$$\phi(x) = \sum_{i=0}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

Lagrange 插值公式还可以表示为

$$\phi(x) = \sum_{i=0}^{n} w_i \prod_{j \neq i} (x - x_j) = p(x) \sum_{i=0}^{n} \frac{w_i}{x - x_i}$$

其中

$$p(x) = \prod_{i=0}^{n} (x - x_i), \quad w_i = \frac{y_i}{\prod\limits_{j \neq i} (x_i - x_j)} = \frac{y_i}{p'(x_i)}.$$

当需要计算 $\phi(x)$ 在多个点处的值时，可先计算 $w_0, w_1, \cdots, w_n$，再利用上式求值.

例 2.1. 已知 $z_1, \cdots, z_n \in \mathbb{C}$ 是方程 $x^n = x + 1$ 的所有根，求 $n - 1$ 次多项式 $\phi(x) \in \mathbb{C}[x]$ 使得

$$\phi(z_i) = \overline{z_i}, \quad \forall i.$$

解答. 在 Lagrange 插值公式中设 $p(x) = x^n - x - 1$，得

$$\phi(x) = (x^n - x - 1) \sum_{i=1}^{n} \frac{w_i}{x - z_i}, \quad w_i = \frac{\overline{z_i}}{n z_i^{n-1} - 1} = \frac{|z_i|^2}{(n-1) z_i + n}.$$ □

Lagrange 插值公式可以理解为：把 ϕ 表示为 $\mathbb{R}_{n+1}[x]$ 的基 $\{L_0, L_1, \cdots, L_n\}$ 的线性组合，其中基函数

$$L_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

是 n 次多项式. 如果取 $\mathbb{R}_{n+1}[x]$ 的基为 $\{T_0, T_1, \cdots, T_n\}$，其中基函数

$$T_i(x) = \prod_{j<i} (x - x_i)$$

是 i 次多项式，则得到 Newton 插值公式.

定义 2.2. 设实数 $x_0, x_1, \cdots, x_n$ 两两不同，函数 $f(x)$ 关于 $x_0, x_1, \cdots, x_n$ 的 n 阶**差商**定义为

$$f[x_0] = f(x_0), \quad f[x_0, x_1, \cdots, x_n] = \frac{f[x_0, \cdots, x_{n-2}, x_n] - f[x_0, \cdots, x_{n-2}, x_{n-1}]}{x_n - x_{n-1}}, \quad n \geqslant 1.$$

差商 $f[x_0, x_1, \cdots, x_n]$ 的定义仅依赖于 f 在 $x_0, x_1, \cdots, x_n$ 处的值 $y_0, y_1, \cdots, y_n$.

定理 2.3. 设 x_i, y_i, ϕ 同定理 *2.1*，则有 **Newton 插值公式**

$$\phi(x) = \sum_{i=0}^{n} f[x_0, \cdots, x_i] T_i(x).$$

证明. 设 $\phi(x) = \sum_{i=0}^{n} c_i T_i(x)$，得线性方程组

$$\begin{pmatrix} T_0(x_0) & & & \\ T_0(x_1) & T_1(x_1) & & \\ \vdots & \vdots & \ddots & \\ T_0(x_n) & T_1(x_n) & \cdots & T_n(x_n) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}. \tag{2.2}$$

作初等行变换，第 $i+1$ 行减去第 1 行，然后除以 $x_i - x_0$，得

$$\begin{pmatrix} T_1^{(1)}(x_1) & & & \\ T_1^{(1)}(x_2) & T_2^{(1)}(x_2) & & \\ \vdots & \vdots & \ddots & \\ T_1^{(1)}(x_n) & T_2^{(1)}(x_n) & \cdots & T_n^{(1)}(x_n) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f[x_0, x_1] \\ f[x_0, x_2] \\ \vdots \\ f[x_0, x_n] \end{pmatrix}$$

其中 $T_i^{(k)}(x) = \prod_{k \leqslant j < i} (x - x_j)$. 不断进行下去，得 $c_k = f[x_0, x_1, \cdots, x_k]$，$\forall k$. □

Lagrange 插值公式与 Newton 插值公式是同一个多项式 $\phi(x)$ 的不同表达形式.

由定理 2.3 可知，$f[x_0, x_1, \cdots, x_n]$ 是插值多项式 $\phi(x)$ 的最高项系数，与 $x_0, x_1, \cdots, x_n$ 的顺序无关. 即对于 $0, 1, \cdots, n$ 的任意排列 σ，有

$$f[x_{\sigma_0}, x_{\sigma_1}, \cdots, x_{\sigma_n}] = f[x_0, x_1, \cdots, x_n].$$

除了利用线性方程组 (2.2) 之外，也可以利用**差商表** $(y_{ij})_{0\leqslant i\leqslant j\leqslant n}$ 求 Newton 插值公式，其中

$$y_{ij}=f[x_{j-i},\cdots,x_j]=\begin{cases}f(x_j), & i=0;\\ \dfrac{y_{i-1,j}-y_{i-1,j-1}}{x_j-x_{j-i}}, & i\geqslant 1.\end{cases}$$

定理 2.4. *设 $f(x)$ 在 $[a,b]$ 上有 n 阶连续导函数，$a\leqslant x_0<\cdots<x_n\leqslant b$，则存在 $\xi\in[a,b]$ 使得*

$$f[x_0,\cdots,x_n]=\frac{f^{(n)}(\xi)}{n!}.$$

证明. 记 $g(x)=f(x)-\sum\limits_{i=0}^{n}f[x_0,\cdots,x_i]T_i(x)$，则有 $g(x_0)=\cdots=g(x_n)=0$. 根据微分中值定理，存在 $\xi\in[a,b]$ 使得 $g^{(n)}(\xi)=0$，即 $f^{(n)}(\xi)-n!f[x_0,\cdots,x_n]=0$. □

由定理 2.4 可得多项式插值的如下误差估计.

定理 2.5. *设 $f(x)$ 在 $[a,b]$ 上有 $n+1$ 阶连续导函数，$a\leqslant x_0<\cdots<x_n\leqslant b$，$\phi(x)$ 是 $f(x)$ 关于节点 $x_0,\cdots,x_n$ 的插值多项式. 对于任意 $x\in[a,b]$，存在 $\xi\in[a,b]$ 使得*

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

证明. 根据定理 2.4，f 关于节点 $x_0,\cdots,x_n,x$ 的插值多项式

$$g(t)=\phi(t)+f[x_0,\cdots,x_n,x](t-x_0)\cdots(t-x_n)=\phi(t)+\frac{f^{(n+1)}(\xi)}{(n+1)!}(t-x_0)\cdots(t-x_n),$$

其中 ξ 与 t 无关．令 $t=x$，得 $f(x)=g(x)=\phi(x)+\dfrac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)\cdots(x-x_n)$． □

思考：Lagrange 插值与 Newton 插值的复杂度有多大？哪个方法的误差小？哪个方法的稳定性好？

§2.1.2 离散 Fourier 变换

在前面的多项式插值问题中，插值节点是区间 $[a,b]$ 中的若干实数．插值节点也可以是复数，Lagrange 插值公式和 Newton 插值公式对于复数值的数据仍然成立．例如，设插值节点 $1,\omega,\cdots,\omega^{n-1}$，其中 $\omega=\mathrm{e}^{\frac{2\pi\sqrt{-1}}{n}}$ 是 n 次单位根，则插值函数 $\phi(x)=\sum\limits_{k=0}^{n-1}c_kx^k$ 对应线性方程组 $\Omega\boldsymbol{c}=\boldsymbol{y}$，其中

$$\Omega=\begin{pmatrix}1 & 1 & \cdots & 1\\ 1 & \omega & \cdots & \omega^{n-1}\\ \vdots & \vdots & \omega^{ij} & \vdots\\ 1 & \omega^{n-1} & \cdots & \omega^{(n-1)^2}\end{pmatrix}.$$

由 $\Omega\overline{\Omega}=nI$，得 $\boldsymbol{c}=\frac{1}{n}\overline{\Omega}\boldsymbol{y}$.

定义 2.3. 设 Ω 如上. $\mathbb{C}^n$ 上的线性变换 $\mathcal{F}(\alpha)=\Omega\alpha$ 称为**离散 Fourier 变换**. 当 $n=2^m$ $(m\in\mathbb{N})$ 时，下述计算 $\mathcal{F}(\alpha)$ 的算法称为**快速 Fourier 变换**，简称 **FFT**，其时间复杂度为 $O(n\log n)$.

设 $\alpha=(a_1,\cdots,a_n)^T$，$\mathcal{F}(\alpha)=(b_1,\cdots,b_n)^T$，则 $b_k=f(\omega^{k-1})$，其中 $f(x)=\sum\limits_{k=1}^{n}a_kx^{k-1}$. 注意到

$$f(x)=g(x^2)+xh(x^2),\quad g(x)=\sum_{1\leqslant k\leqslant n/2}a_{2k}x^{k-1},\quad h(x)=\sum_{1\leqslant k\leqslant n/2}a_{2k-1}x^{k-1}.$$

可先计算 $g_k=g(\omega^{2k-2})$ 和 $h_k=h(\omega^{2k-2})$，再计算 $b_k=g_k+\omega^{k-1}h_k$ 和 $b_{k+n/2}=g_k-\omega^{k-1}h_k$，$1\leqslant k\leqslant n/2$. 设上述递归算法包含的复数加减乘运算[1]的次数为 $T(n)$，则有

$$T(n)=2T(n/2)+1.5n+O(1)\qquad\Rightarrow\qquad T(n)=1.5n\log_2 n+O(n).$$

对于一般的 n，存在与 FFT 类似的快速算法，时间复杂度为 $O(n\log^2 n)$，不再详述.

FFT 有许多实际应用. 例如，可利用 FFT 快速计算两个复系数多项式 f,g 的乘积.

例 2.2. 设 $d_1=\deg(f)$，$d_2=\deg(g)$，$n=2^{\lceil\log_2(1+d_1+d_2)\rceil}$，$\omega=\mathrm{e}^{\frac{2\pi\sqrt{-1}}{n}}$. 记 $f(x)=\sum\limits_{i=1}^{n}f_ix^{i-1}$，$g(x)=\sum\limits_{i=1}^{n}g_ix^{i-1}$. 先用 FFT 计算 $(a_1,\cdots,a_n)=\mathcal{F}(f_1,\cdots,f_n)$，$(b_1,\cdots,b_n)=\mathcal{F}(g_1,\cdots,g_n)$，再用 FFT 计

[1] 1 次复数加减 = 2 次实数加减. 1 次复数乘 = 4 次实数乘 +2 次实数加减，或者 1 次复数乘 = 3 次实数乘 +5 次实数加减.

算 $(c_1,\cdots,c_n)=\mathcal{F}(\overline{a_1b_1},\cdots,\overline{a_nb_n})$，则 $(h_1,\cdots,h_n)=\frac{1}{n}(\overline{c_1},\cdots,\overline{c_n})$ 是 fg 的系数，$fg=\sum\limits_{i=1}^{n}h_ix^{i-1}$.
上述算法包含 $4.5n\log_2 n+O(n)$ 次复数加减乘运算．经典算法则包含 $2d_1d_2+O(d_1+d_2)$ 次运算． □

§2.1.3 Hermite 插值

在构造插值多项式 $\phi(x)$ 的时候，通常指定节点处的函数值 $\phi(x_i)=y_i$．如果还指定节点处的各阶导数值 $\phi^{(j)}(x_i)=y_i^{(j)}$，$0\leqslant i\leqslant n$，$0\leqslant j\leqslant k_i$，则这样的插值问题称为 **Hermite 插值**.

Hermite 插值问题可以转化为 Newton 插值类型问题．把每个节点 x_i 扩展为 k_i+1 个重复节点 $\underbrace{x_i,\cdots,x_i}_{k_i+1\text{个}}$，并规定 $f[\underbrace{x_i,\cdots,x_i}_{j+1\text{个}}]=\dfrac{y_i^{(j)}}{j!}$，进而利用差商表构造 $\sum\limits_{i=0}^{n}k_i$ 个节点的 Newton 插值多项式.

Hermite 插值问题也可以转化为 Lagrange 插值类型问题．此时需要构造基函数 $L_{p,q}(x)$ 使得

$$\frac{d^jL_{p,q}}{dx^j}(x_i)=\begin{cases}1, & (i,j)=(p,q)\\ 0, & (i,j)\neq(p,q)\end{cases}\tag{2.3}$$

其中 $0\leqslant i,p\leqslant n$，$0\leqslant j,q\leqslant k_i$．此方法较繁琐，仅适用于 $k_0,\cdots,k_n\leqslant 2$ 或 n 较小的情形.

例 2.3. 给定 $x_0,\cdots,x_n,y_0,\cdots,y_n,z_0,\cdots,z_n$，构造不超过 $2n+1$ 次的多项式 $\phi(x)$，使得

$$\phi(x_i)=y_i,\quad \phi'(x_i)=z_i,\quad \forall i=0,\cdots,n.$$

方法 1. 记 $t_{2i}=t_{2i+1}=x_i$，$0\leqslant i\leqslant n$. 构造差商表 $(q_{ij})_{0\leqslant i\leqslant j\leqslant 2n+1}$，其中

$$\begin{cases} q_{0,2j}=q_{0,2j+1}=y_j, \\ \qquad 0\leqslant j\leqslant n; \end{cases} \quad \begin{cases} q_{1,2j}=\dfrac{y_j-y_{j-1}}{x_j-x_{j-1}}, & 1\leqslant j\leqslant n; \\ q_{1,2j+1}=z_j, & 0\leqslant j\leqslant n; \end{cases} \quad \begin{cases} q_{i,j}=\dfrac{q_{i-1,j}-q_{i-1,j-1}}{t_j-t_{j-i}}, \\ \qquad 2\leqslant i\leqslant j\leqslant 2n+1. \end{cases}$$

得 $\phi(x)=q_{00}+\sum\limits_{i=1}^{2n+1} q_{ii}\prod\limits_{j=0}^{i-1}(x-t_j)$. □

方法 2. 构造基函数 $\{L_{ij}\mid 0\leqslant i\leqslant n,\ 0\leqslant j\leqslant 1\}$ 满足 (2.3)，得 $\varphi(x)=\sum\limits_{i=0}^{n}(y_iL_{i0}(x)+z_iL_{i1}(x))$.

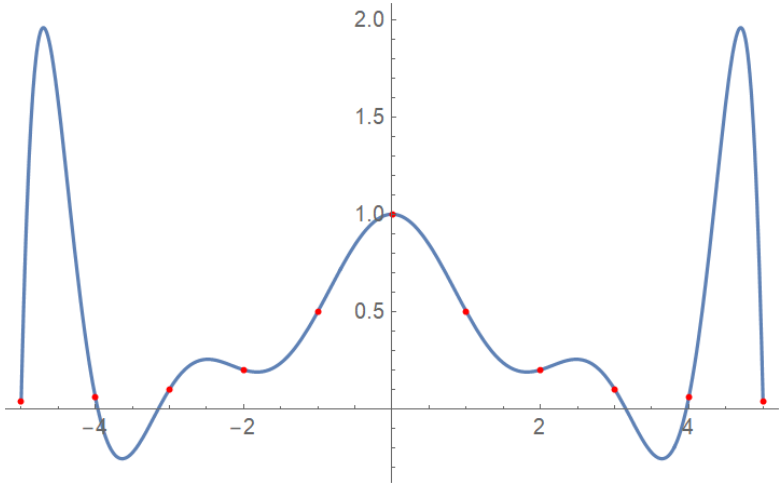
$$L_{i0}(x)=(a_ix+b_i)\prod_{k\neq i}\left(\frac{x-x_k}{x_i-x_k}\right)^2,\quad L_{i1}(x)=(x-x_i)\prod_{k\neq i}\left(\frac{x-x_k}{x_i-x_k}\right)^2$$

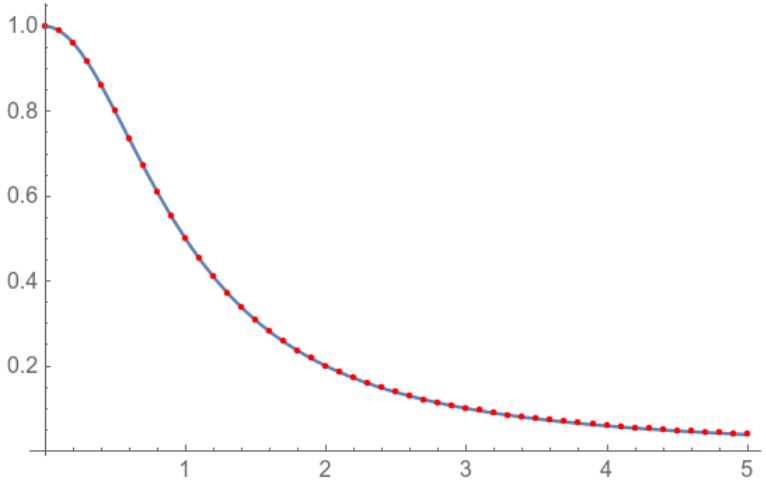
其中 $a_i=\sum\limits_{k\neq i}\dfrac{-2}{x_i-x_k}$，$b_i=1-a_ix_i$. 故 $L_{i0}(x)=\left(1+2(x_i-x)\sum\limits_{k\neq i}\dfrac{1}{x_i-x_k}\right)\prod\limits_{k\neq i}\left(\dfrac{x-x_k}{x_i-x_k}\right)^2$. □

思考：比较例 2.3 的两种方法的优劣.

§2.1.4 样条插值

例 2.4. 设 $f(x)=\dfrac{1}{1+x^2}$. 下面左图是 $f(x)$ 关于节点 $0,\pm1,\cdots,\pm5$ 的 10 次插值多项式的函数图像，右图是 $f(x)$ 关于节点 $0,0.1,\cdots,5.0$ 的 50 次插值多项式的函数图像.





左图这种插值效果比较差的现象称为 **Runge 现象**. Runge 现象的成因比较复杂，不详细解释. □

在绘制高次多项式的函数图像时，舍入误差可能会使得事实上应该平缓的函数图像变得剧烈震荡. 这种现象应当与 Runge 现象区分开来.

为了避免高次多项式插值可能带来的较大误差，可以使用样条函数进行插值.

定义 2.4. 设 k,n 都是正整数，$x_0<x_1<\cdots<x_n$. 若定义在 $[x_0,x_n]$ 上的函数 $\phi(x)$ 满足

① $\phi(x)$ 在每个 $[x_{i-1}, x_i]$ 上都是次数不超过 k 的多项式，$1 \leqslant i \leqslant n$；

② $\phi(x)$ 在每个 x_i 处有 $k-1$ 阶连续导函数，$1 \leqslant i \leqslant n-1$；

则 $\phi(x)$ 称为以 $x_0, x_1, \cdots, x_n$ 为节点的 k 次**样条函数**. 利用样条函数来插值的方法称为**样条插值**.

定理 2.6. *以 $x_0 < x_1 < \cdots < x_n$ 为节点的 k 次样条函数的集合 Φ 在函数的加法和数乘运算下构成实线性空间，$\dim \Phi = n + k$，$\{1, x, \cdots, x^k, S_1, \cdots, S_{n-1}\}$ 是 Φ 的基，$S_i(x) = (\max(0, x - x_i))^k$.*

证明. 任意 $\phi \in \Phi$ 的 k 阶导函数 g 是分段常值函数，有至多 $n-1$ 个间断点 $x_1, \cdots, x_{n-1}$.
对 g 作 k 次积分，得 ϕ 形如 $\sum\limits_{i=0}^{k} c_i x^k + \sum\limits_{i=1}^{n-1} c_{k+i} S_i(x)$. □

根据定理 2.6，可设样条插值函数 $\phi(x) = \sum\limits_{i=0}^{k} c_i (x-a)^i + \sum\limits_{i=1}^{n-1} c_{k+i} S_i(x)$，然后求解线性方程组

$$\begin{pmatrix} 1 & (x_0-a) & \cdots & (x_0-a)^k & & & \\ 1 & (x_1-a) & \cdots & (x_1-a)^k & & & \\ 1 & (x_2-a) & \cdots & (x_2-a)^k & (x_2-x_1)^k & & \\ \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \\ 1 & (x_n-a) & \cdots & (x_n-a)^k & (x_n-x_1)^k & \cdots & (x_n-x_{n-1})^k \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_k \\ \vdots \\ c_{n+k-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}. \tag{2.4}$$

(2.4) 式有 $n+1$ 个方程、$n+k$ 个未知数，可适当补充 $k-1$ 个约束．例如，

$$\phi'(x_0)=0,\quad \phi''(x_0)=0,\quad \phi'(x_n)=0,\quad \phi''(x_n)=0,\quad \cdots$$

Φ 有其他形式的基函数，如 B 样条，此处不详细叙述．对于 3 次样条插值，还有更有效的方法．

思考：比较 1、2、3、4 次样条函数，各有何优缺点？为何 3 次样条函数较常用？

M 关系式方法

设 $M_i=\phi''(x_i)$，$0\leqslant i\leqslant n$．当 $x\in[x_{i-1},x_i]$ 时，记 $h_i=x_i-x_{i-1}$，由

$$\phi''(x)=\frac{(x-x_{i-1})M_i+(x_i-x)M_{i-1}}{h_i},\quad \phi(x_{i-1})=y_{i-1},\quad \phi(x_i)=y_i$$

待定系数法解得

$$\begin{cases}\phi(x)=\dfrac{M_i}{6h_i}(x-x_{i-1})^3+\dfrac{M_{i-1}}{6h_i}(x_i-x)^3+\left(\dfrac{y_i}{h_i}-\dfrac{M_ih_i}{6}\right)(x-x_{i-1})+\left(\dfrac{y_{i-1}}{h_i}-\dfrac{M_{i-1}h_i}{6}\right)(x_i-x)\\ \phi'(x)=\dfrac{M_i}{2h_i}(x-x_{i-1})^2-\dfrac{M_{i-1}}{2h_i}(x_i-x)^2+\dfrac{y_i}{h_i}-\dfrac{y_{i-1}}{h_i}+\dfrac{(M_{i-1}-M_i)h_i}{6}.\end{cases}$$

再由 $\phi'(x)$ 在 x_i 处的连续性，得

$$\frac{y_i-y_{i-1}}{h_i}+\frac{(2M_i+M_{i-1})h_i}{6}=\frac{y_{i+1}-y_i}{h_{i+1}}-\frac{(M_{i+1}+2M_i)h_{i+1}}{6}.$$

从而有

$$\begin{pmatrix} a_1 & 2 & 1-a_1 & & & \\ & a_2 & 2 & 1-a_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{n-1} & 2 & 1-a_{n-1} \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{pmatrix}, \tag{2.5}$$

其中 $a_i = \dfrac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}$，$b_i = 6f[x_{i-1}, x_i, x_{i+1}]$.

(2.5) 式有 $n-1$ 个方程、$n+1$ 个未知数，可适当补充两个约束条件．例如，指定 M_0, M_n 的值，指定 $\phi'(x_0), \phi'(x_n)$ 的值，要求 $M_0 = M_n$，$\phi'(x_0) = \phi'(x_n)$ 等.

m 关系式方法

设 $m_i = \phi'(x_i)$，$0 \leqslant i \leqslant n$．当 $x \in [x_{i-1}, x_i]$ 时，记 $h_i = x_i - x_{i-1}$，$z_i = \dfrac{y_i - y_{i-1}}{x_i - x_{i-1}}$，由 Newton 插值形式的 Hermite 插值公式可得

$$\begin{cases} \phi(x) = y_{i-1} + m_{i-1}(x - x_{i-1}) + \dfrac{z_i - m_{i-1}}{h_i}(x - x_{i-1})^2 + \dfrac{m_{i-1} - 2z_i + m_i}{h_i^2}(x - x_{i-1})^2(x - x_i), \\ \dfrac{\phi''(x)}{2} = \dfrac{z_i - m_{i-1}}{h_i} + \dfrac{m_{i-1} - 2z_i + m_i}{h_i^2}(3x - 2x_{i-1} - x_i). \end{cases}$$

再由 $\phi''(x)$ 在 x_i 处的连续性，得

$$\frac{m_{i-1}+2m_i-3z_i}{h_i}=\frac{3z_{i+1}-2m_i-m_{i+1}}{h_{i+1}}.$$

从而有

$$\begin{pmatrix} 1-a_1 & 2 & a_1 & & & \\ & 1-a_2 & 2 & a_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1-a_{n-1} & 2 & a_{n-1} \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ \vdots \\ m_n \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix}, \tag{2.6}$$

其中 $a_i=\dfrac{x_i-x_{i-1}}{x_{i+1}-x_{i-1}}$，$c_i=3\dfrac{(x_{i+1}-x_i)z_i+(x_i-x_{i-1})z_{i+1}}{x_{i+1}-x_{i-1}}$.

(2.6) 式有 $n-1$ 个方程、$n+1$ 个未知数，可适当补充两个约束条件．例如，指定 m_0,m_n 的值，指定 $\phi''(x_0),\phi''(x_n)$ 的值，要求 $m_0=m_n$，$\phi''(x_0)=\phi''(x_n)$ 等．

比较三次样条插值的基函数方法、M 关系式方法、m 关系式方法．如果补充的约束条件相同，则得到同一个样条函数 $\phi(x)$ 的三种不同表达形式．求解线性方程组 (2.4) 的时间复杂度为 $O(n^2)$，求解 (2.5) 和 (2.6) 的时间复杂度为 $O(n)$．故在确定 $\phi(x)$ 的表达形式时，M,m 关系式方法优于基函数方法．但是，在计算 $\phi(x)$ 在多个点处的值时，M,m 关系式方法的时间复杂度要大于基函数方法．

样条插值的误差估计比较复杂，此处不详细叙述.

定理 2.7. 设 $x_0 < x_1 < \cdots < x_n$，$f(x)$ 在 $[x_0, x_n]$ 上有 *2* 阶连续导函数，$\phi(x)$ 是 $f(x)$ 关于节点 $x_0, x_1, \cdots, x_n$ 的 *1* 次样条插值函数，则对于任意 $x \in [x_{i-1}, x_i]$，存在 $\xi \in [x_{i-1}, x_i]$ 使得

$$\phi(x) = f(x) - \frac{f''(\xi)}{2}(x - x_{i-1})(x - x_i).$$

证明. 定理 2.5 的特例. □

§2.2 拟合

鉴于多项式插值可能出现诸如 Runge 现象一样的缺陷，我们要求函数族 Φ 具有良好的性质，不要求近似函数 ϕ 满足 $\phi(x_i)=y_i,\ \forall i$，只要 $\phi(x_i)\approx y_i$ 即可．这种构造近似函数的方法称为**拟合**.

定义 2.5. 给定函数族 Φ 和平面点列 $(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)$．记

$$\boldsymbol{x}=(x_1,x_2,\cdots,x_m),\quad \boldsymbol{y}=(y_1,y_2,\cdots,y_m),\quad \phi(\boldsymbol{x})=(\phi(x_1),\phi(x_2),\cdots,\phi(x_m))$$

其中 $\phi\in\Phi$．设 $\|\cdot\|$ 是 $\mathbb{R}^m$ 上的一个范数，则 $\|\phi(\boldsymbol{x})-\boldsymbol{y}\|$ 可以作为衡量拟合效果的标准．特别，使得 $\|\phi(\boldsymbol{x})-\boldsymbol{y}\|_2$ 最小的 ϕ 称为**最小二乘拟合函数**.

定理 2.8. 设 Φ 是 n 维线性空间，$\{\phi_1,\phi_2,\cdots,\phi_n\}$ 是 Φ 的基，则 $\phi=\sum\limits_{i=1}^{n}c_i\phi_i$ 是最小二乘拟合函数当且仅当 $\boldsymbol{c}=(c_1,c_2,\cdots,c_n)^T$ 满足 $A^TA\boldsymbol{c}=A^T\boldsymbol{y}$，其中 $A=\Big(\phi_j(x_i)\Big)\in\mathbb{R}^{m\times n}$.

证明. 二次函数 $f(\boldsymbol{c})=\|\phi(\boldsymbol{x})-\boldsymbol{y}\|_2^2=\|A\boldsymbol{c}-\boldsymbol{y}\|_2^2=\boldsymbol{c}^TA^TA\boldsymbol{c}-2\boldsymbol{y}^TA\boldsymbol{c}+\boldsymbol{y}^T\boldsymbol{y}$ 取得最小值当且仅当 $\dfrac{\partial f}{\partial\boldsymbol{c}}=2c^TA^TA-2\boldsymbol{y}^TA=\boldsymbol{0}$，即 $A^TA\boldsymbol{c}=A^T\boldsymbol{y}$. □

线性方程组 $A^TA\boldsymbol{c}=A^T\boldsymbol{y}$ 的解 $\boldsymbol{c}$ 也称为线性方程组 $A\boldsymbol{c}=\boldsymbol{y}$ 的**最小二乘解**．通常不直接求解 $A^TA\boldsymbol{c}=A^T\boldsymbol{y}$，而是利用奇异值分解 $A=P\begin{pmatrix}\Sigma & \\ & O\end{pmatrix}Q$，求得 $\boldsymbol{c}=Q^T\begin{pmatrix}\Sigma^{-1} & \\ & O\end{pmatrix}P^T\boldsymbol{y}$.

如果 Φ 不是线性空间，最小二乘拟合函数不容易求得，可先预处理数据，再求解最小二乘拟合问题.

例 2.5. 对数据 $\{(x_i, y_i) \mid 1 \leqslant i \leqslant n\}$ 作预处理，求形如 $y = \dfrac{a+bx}{1+cx}$ 的拟合.

解答.

$$y \approx \frac{a+bx}{1+cx} \quad \Leftrightarrow \quad a+bx \approx y+cxy \quad \Leftrightarrow \quad \begin{pmatrix} 1 & x_1 & -x_1y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & -x_ny_n \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} \approx \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

□

分析预处理方法的数学原理. 设原问题是求参数 θ 使得 $S = \sum\limits_{i=1}^{n} \phi^2(\theta, x_i, y_i)$ 最小，对数据作预处理之后，问题转化为求 θ 使得 $T = \sum\limits_{i=1}^{n} \varphi^2(\theta, x_i, y_i)$ 最小. 这相当于是对原目标函数 $\phi(\theta, x_i, y_i)$ 赋了权值 $w_i \geqslant 0$，使得 $T = \sum\limits_{i=1}^{n} w_i\phi^2(\theta, x_i, y_i)$，改变了数据 (x_i, y_i) 对于 θ 的影响程度.

例如，例 2.5 的原目标函数 $S = \sum\limits_{i=1}^{n} \left(\dfrac{a+bx_i}{1+cx_i} - y_i\right)^2$，预处理后变为 $T = \sum\limits_{i=1}^{n} \big(a+bx_i-(1+cx_i)y_i\big)^2$，权值 $w_i = (1+cx_i)^2$. 为了兼顾拟合效果和便于计算，还可以选取与 a, b, c 无关的权值 $\lambda_i \geqslant 0$，求 a, b, c 使得 $\sum\limits_{i=1}^{n} \lambda_i\big(a+bx_i-(1+cx_i)y_i\big)^2$ 最小.

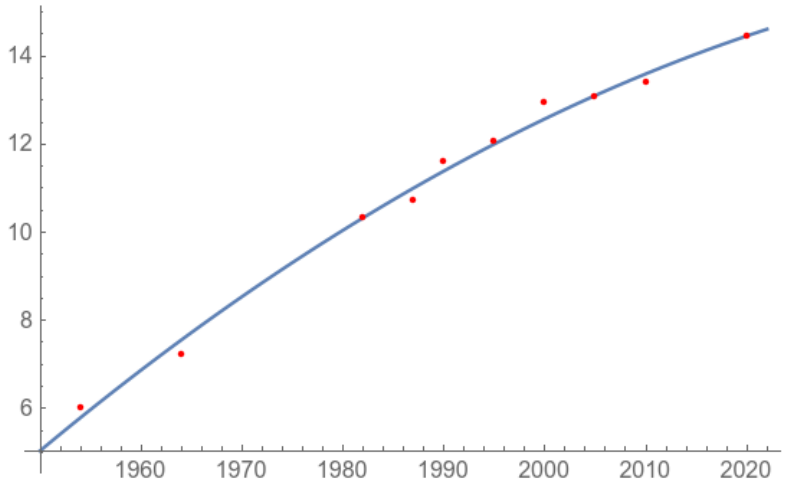
例 2.6. 用二次函数最小二乘拟合我国人口增长数据.

x_i	1954	1964	1982	1987	1990	1995	2000	2005	2010	2020
y_i	6.0194	7.2307	10.3188	10.7233	11.6002	12.0778	12.9533	13.0756	13.3972	14.4350

解答. 设 $a = 1990$，$\phi(x) = \sum_{j=1}^{3} c_j (x-a)^{j-1}$，得近似线性方程组 $A\boldsymbol{c} \approx \boldsymbol{y}$，其中 $A = \left((x_i - a)^{j-1}\right)$.

$$A^T A = \begin{pmatrix} s_0 & s_1 & s_2 \\ s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \end{pmatrix} = \begin{pmatrix} 10 & 7 & 3695 \\ 7 & 3695 & -25271 \\ 3695 & -25271 & 3172019 \end{pmatrix}, \quad A^T \boldsymbol{y} = \begin{pmatrix} \sum y_i \\ \sum (x_i - a) y_i \\ \sum (x_i - a)^2 y_i \end{pmatrix} = \begin{pmatrix} 111.831 \\ 567.633 \\ 36335.7 \end{pmatrix},$$

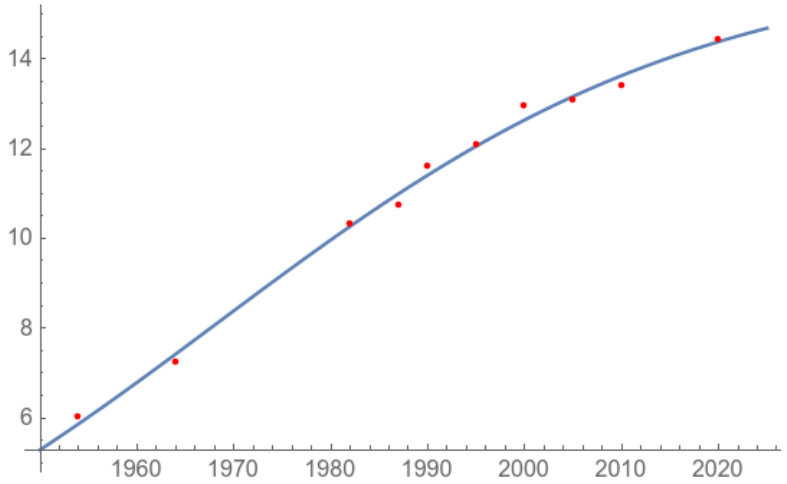
其中 $s_k = \sum_{i=1}^{10} (x_i - a)^k$. 解得 $\boldsymbol{c} = (11.3936,\ 0.126503,\ -0.000809161)$，$\|A\boldsymbol{c} - \boldsymbol{y}\|_2^2 = 0.471379$. □



例 2.7. 对例 2.6 的数据建立 Logistic 模型，求形如 $y=\dfrac{a}{1+b\,\mathrm{e}^{-cx}}$ 的拟合.

方法 1. 设 $f(x)=\dfrac{a}{1+b\,\mathrm{e}^{-0.01c(x-1990)}}$. 求 a,b,c 使得 $S=\sum\limits_{i=1}^{10}(f(x_i)-y_i)^2$ 最小.

解得 $a=16.2247$，$b=0.422203$，$c=3.97188$，$S=0.335118$. 拟合效果如图所示. □



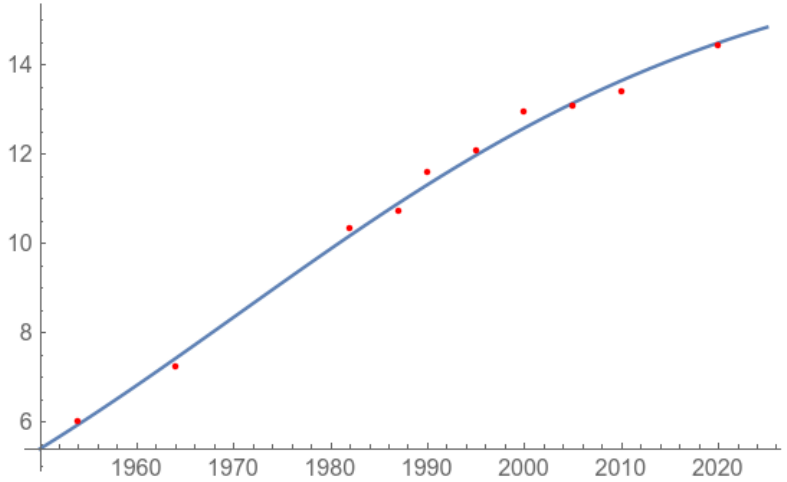
方法 2. 设 $y=f(x)=\dfrac{a}{1+b\,\mathrm{e}^{-c(x-1990)}}$，则有 $f'(x)=cy\left(1-\dfrac{y}{a}\right)$.

首先，求 a,c 使得 $\sum\limits_{i=1}^{10}\left(cy_i(1-\dfrac{y_i}{a}-z_i)\right)^2$ 最小，其中 z_i 是 $f'(x_i)$ 的数值微分.

$$z_1=\frac{y_2-y_1}{x_2-x_1},\quad z_{10}=\frac{y_{10}-y_9}{x_{10}-x_9},\quad z_i=\frac{y_i-y_{i-1}}{x_i-x_{i-1}}+\frac{y_{i+1}-y_i}{x_{i+1}-x_i}-\frac{y_{i+1}-y_{i-1}}{x_{i+1}-x_{i-1}},\quad 2\leqslant i\leqslant 9.$$

解得 $c=0.0366826$．然后，固定 c，求 a,b 使得 $\sum\limits_{i=1}^{10}\left(1+b\,\mathrm{e}^{-c(x_i-1990)}-\dfrac{a}{y_i}\right)^2$ 最小．

解得 $a=16.856$，$b=0.488565$，$\sum\limits_{i=1}^{10}(f(x_i)-y_i)^2=0.387238$．拟合效果如图所示． □



比较例 2.6 的二次函数模型和例 2.7 的 Logistic 模型，各有所长，很难说哪种模型更优．

例 2.7 中的两种拟合方法均对数据作了预处理．如果不作预处理，尝试一下结果如何．

比较例 2.7 中的两种拟合方法，从误差的角度看，很难说哪种方法更优；

从计算复杂度的角度看，方法 1 需要求非线性函数的最值点，方法 2 使用线性最小二乘拟合．

§2.3 函数逼近

插值和拟合都是求有限点列的近似函数的方法. 如何求无穷点列的近似函数呢？换句话说，假设不知道函数 $f(x)$ 的具体表达式，但是知道 $f(x)$ 在任意点处的取值，如何求 $f(x)$ 的近似函数？

定义 2.6. 设 V 是实数域 $\mathbb{R}$ 上的线性空间.

- 具有下列性质的映射 $\rho : V \times V \to \mathbb{R}$ 称为 V 上的**内积**.

  ① $\rho(\lambda\alpha + \mu\beta, \gamma) = \lambda\rho(\alpha, \beta) + \mu\rho(\alpha, \gamma)$，$\rho(\alpha, \lambda\beta + \mu\gamma) = \lambda\rho(\alpha, \beta) + \mu\rho(\alpha, \gamma)$；

  ② $\rho(\alpha, \beta) = \rho(\beta, \alpha)$；③ 当 $\alpha \neq \mathbf{0}$ 时，$\rho(\alpha, \alpha) > 0$；其中 $\alpha, \beta, \gamma \in V$，$\lambda, \mu \in \mathbb{R}$.

- 设 ρ 是 V 上的内积，则 (V, ρ) 称为**内积空间**，$\|\alpha\| = \sqrt{\rho(\alpha, \alpha)}$ 称为 ρ 的**诱导范数**.

- 若内积空间 (V, ρ) 中的向量组 $S = \{\alpha_i\}_{i \in I}$ 满足 $\rho(\alpha_i, \alpha_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$，$\forall i, j \in I$，则 S 称为**标准正交向量组**. 特别，若 S 还是 V 的基，则 S 称为**标准正交基**.

例 2.8. 设实线性空间 $\mathcal{C}[a, b]$ 由 $[a, b]$ 上的所有连续函数构成. 容易验证

$$\rho(f, g) = \int_a^b f(x)g(x)w(x)dx$$

是 V 上的**内积**，其中 $w \in \mathcal{C}[a, b]$ 满足 $w(x) > 0$，$\forall x \in [a, b]$，称为**权函数**. □

定理 2.9. 任意可数维或有限维实内积空间 (V,ρ) 都有标准正交基.

证明. 任取 V 的基 $\{\alpha_1,\alpha_2,\cdots\}$，施行如下 **Gram-Schmidt 标准正交化**过程，

$$\beta_k=\alpha_k-\sum_{i=1}^{k-1}\rho(\alpha_k,\gamma_i)\gamma_i,\quad \gamma_k=\frac{\beta_k}{\|\beta_k\|},\quad k=1,2,\cdots$$

由 $\rho(\beta_k,\gamma_i)=0$，$\forall i=1,\cdots,k-1$，得 $S=\{\gamma_1,\gamma_2,\cdots\}$ 是标准正交向量组. 再由 $\mathrm{Span}(\gamma_1,\cdots,\gamma_k)=\mathrm{Span}(\alpha_1,\cdots,\alpha_k)$，得 S 是 V 的标准正交基. □

定理 2.10. 设 U 是实内积空间 (V,ρ) 的有限维子空间，$\{e_1,\cdots,e_n\}$ 是 U 的标准正交基. 对于任意 $\alpha\in V$，存在唯一的 $\beta\in U$ 使得 $\|\alpha-\beta\|$ 最小. $\beta=\sum_{i=1}^{n}\rho(\alpha,e_i)e_i$ 称为 α 在 U 中的**投影**.

证明. 记 $u=\sum_{i=1}^{n}\rho(\alpha,e_i)e_i$，$v=\alpha-u$. 由 $\rho(e_i,v)=0$, $\forall i$，得 $\rho(u,v)=0$. 对于任意 $\beta\in U$，$\|\alpha-\beta\|^2=\|(u-\beta)+v\|^2=\|u-\beta\|^2+\|v\|^2$. 当且仅当 $\beta=u$ 时，$\|\alpha-\beta\|$ 取得最小值 $\|v\|$. □

定义 2.7. 设内积空间 $V=\mathcal{C}[a,b]$ 的内积同例 2.8，V 的子空间 U 由所有次数不超过 n 的一元实系数多项式构成. 任意 $f\in V$ 在 U 中的投影 p 称为 f 的 n 次**最佳平方逼近**多项式.

类似地，若 $p\in U$ 使得 $\max\limits_{a\leqslant x\leqslant b}|f(x)-p(x)|$ 最小，则 $p(x)$ 称为 $f(x)$ 的 n 次**最佳一致逼近**多项式.

换句话说，最佳平方逼近使得 $\|f-p\|$ 最小，最佳一致逼近使得 $\|f-p\|_\infty$ 最小.

例 2.9. 设 $f(x)=\mathrm{e}^x$，$x\in[-1,1]$. 求线性函数 $p(x)=ax+b$ 使得 $\max\limits_{-1\leqslant x\leqslant 1}|f(x)-p(x)|$ 最小.

解答. 由函数图像可设 $a>0$，$f(x)-p(x)$ 在 $x=\pm1$ 处取得最大值 c，在 $x=\ln a$ 处取得最小值 $-c$.

$$\begin{cases}\dfrac{1}{\mathrm{e}}+a-b=c\\ \mathrm{e}-a-b=c\\ a-a\ln a-b=-c\end{cases}\quad\Rightarrow\quad\begin{cases}a=\dfrac{\mathrm{e}-\mathrm{e}^{-1}}{2}\approx 1.1752\\ b=\dfrac{\mathrm{e}-a\ln a}{2}\approx 1.26428\\ c=\dfrac{\mathrm{e}+a\ln a}{2}-a\approx 0.278802\end{cases}$$ □

通常很难求得一个连续函数的 n 次最佳一致逼近多项式.

定理 2.11. 设 S 是次数不超过 n 的一元实系数多项式构成的集合，f 是 $[a,b]$ 上的连续函数，$f\notin S$.

1. 若 $a\leqslant x_1<\cdots<x_{n+2}\leqslant b$ 使得 $f(x_1),\cdots,f(x_{n+2})$ 是正负交错的非零实数，
   则 $\|f-p\|_\infty\geqslant\min\limits_{1\leqslant k\leqslant n+2}|f(x_k)|$，$\forall p\in S$.
2. 若 $a\leqslant x_1<\cdots<x_{n+2}\leqslant b$ 使得 $f(x_1),\cdots,f(x_{n+2})$ 是正负交错的并且 $|f(x_k)|=\|f\|_\infty$，$\forall k$，
   则不存在 $p\in S$ 使得 $\|f-p\|_\infty<\|f\|_\infty$.
3. 若不存在 $a\leqslant x_1<\cdots<x_{n+2}\leqslant b$ 使得 $f(x_1),\cdots,f(x_{n+2})$ 是正负交错的并且 $|f(x_k)|=\|f\|_\infty$，
   $\forall k$，则存在 $p\in S$ 使得 $\|f-p\|_\infty<\|f\|_\infty$.
4. f 的 n 次最佳一致逼近多项式 p 是存在且唯一的.

证明.

1. 注意到 $\{f(x_k)p(x_k) \mid 1 \leqslant k \leqslant n+2\}$ 不可能都是正数；否则，$p(x)$ 有至少 $n+1$ 个零点，矛盾. 故存在 $f(x_i)p(x_i) \leqslant 0$，得 $\|f-p\|_\infty \geqslant |f(x_i)-p(x_i)| \geqslant \min\limits_{1\leqslant k\leqslant n+2} |f(x_k)|$.
2. 对于任意 $p \in S$，根据结论 1，$\|f-p\|_\infty \geqslant \min\limits_{1\leqslant k\leqslant n+2} |f(x_k)| = \|f\|_\infty$.
3. 设集合 $X = \{x \in [a,b] : |f(x)| = \|f\|_\infty\} = \{x_1, \cdots, x_m\}$. 根据 $f(x_1), \cdots, f(x_m)$ 的正负号，可以把 X 分成 k 个两两不相交的子集 $X_1 \cup \cdots \cup X_k$，$k \leqslant n+1$，使得

$$f(u)f(v) > 0,\ \forall u, v \in X_i; \quad u < v,\ f(u)f(v) < 0,\ \forall u \in X_i,\ v \in X_{i+1}.$$

   取 $y_1, \cdots, y_n$，使得 $u < y_i < v$，$\forall u \in X_i,\ v \in X_{i+1}$. 令 $p(x) = \lambda(x-y_1)\cdots(x-y_n)$，其中 λ 是充分小的非零数，使得 $p(x_i)f(x_i) > 0$，$\forall 1 \leqslant i \leqslant m$. 从而，$\|f-p\|_\infty < \|f\|_\infty$.
4. 由 S 是闭集可得 p 的存在性. 假设 p_1, p_2 都是 f 的 n 次最佳一致逼近多项式，则 $q = \frac{1}{2}(p_1+p_2)$ 满足 $\|f-q\|_\infty \leqslant \frac{1}{2}(\|f-p_1\|_\infty + \|f-p_2\|_\infty)$. 故 q 也是 f 的 n 次最佳一致逼近多项式. 根据结论 3，存在 $a \leqslant x_1 < \cdots < x_{n+2} \leqslant b$ 使得 $f(x_1)-q(x_1), \cdots, f(x_{n+2})-q(x_{n+2})$ 是正负交错的并且 $|f(x_k)-q(x_k)| = \|f-q\|_\infty$，$\forall k$. 故 $f(x_k)-p_1(x_k) = f(x_k)-p_2(x_k)$，$\forall k$. 即 $p_1 = p_2$.

□

例 2.10. 设 $V=\mathbb{R}[x]$ 上的内积 $\rho(f,g)=\int_a^b f(x)g(x)dx$，则 $\{c_nL_n \mid n\in\mathbb{N}\}$ 是 (V,ρ) 的标准正交基，其中

$$L_n(x)=\frac{d^n}{dx^n}\Big((x-a)^n(x-b)^n\Big),\quad c_n=\frac{\sqrt{2n+1}}{n!(b-a)^{n+\frac{1}{2}}}.$$

证明. 设 $m,n\in\mathbb{N}$，$p(x)=(x-a)^n(x-b)^n$．易知 $\deg(L_n)=n$．当 $m\leqslant n$ 时，分部积分得

$$\int_a^b L_m(x)L_n(x)dx=-\int_a^b L_m'(x)p^{(n-1)}(x)dx=\cdots=(-1)^m(2m)!\int_a^b p^{(n-m)}(x)dx.$$

当 $m<n$ 时，$\rho(L_m,L_n)=0$．当 $m=n$ 时，$\rho(L_n,L_n)=(2n)!\int_a^b(x-a)^n(b-x)^ndx$

$$=(2n)!(b-a)^{2n+1}\int_0^1 x^n(1-x)^ndx=\frac{n!^2(b-a)^{2n+1}}{2n+1}.$$

□

设 $a=1$，$b=-1$

n	0	1	2	3	4	5
c_nL_n	$\frac{1}{\sqrt{2}}$	$\frac{\sqrt{3}x}{\sqrt{2}}$	$\frac{\sqrt{5}(3x^2-1)}{2\sqrt{2}}$	$\frac{\sqrt{7}(5x^3-3x)}{2\sqrt{2}}$	$\frac{105x^4-90x^2+9}{8\sqrt{2}}$	$\frac{\sqrt{11}(63x^5-70x^3+15x)}{8\sqrt{2}}$

下面给出 $\{L_n\}$ 的递推关系，从而可以用 $n+1$ 阶下三角整数矩阵计算并存储 $L_0,\cdots,L_n$ 的系数.

记 $p=(x-a)(x-b)$. 由

$$L_k=(p^k)^{(k)}=(kp'p^{k-1})^{(k-1)}=kp'L_{k-1}+2k(k-1)(p^{k-1})^{(k-2)}$$

得 $(p^{k-1})^{(k-2)}=\dfrac{L_k-kp'L_{k-1}}{2k(k-1)}$. 同理，$(p^k)^{(k-1)}=\dfrac{L_{k+1}-(k+1)p'L_k}{2k(k+1)}$. 由

$$\begin{cases}(p^k)^{(k-1)}=(kp'p^{k-1})^{(k-2)}=kp'(p^{k-1})^{(k-2)}+2k(k-2)(p^{k-1})^{(k-3)}\\ (p^k)^{(k-1)}=(pp^{k-1})^{(k-1)}=pL_{k-1}+(k-1)p'(p^{k-1})^{(k-2)}+(k-1)(k-2)(p^{k-1})^{(k-3)}\end{cases}$$

消去 $(p^{k-1})^{(k-3)}$ 项，得

$$(k+1)(p^k)^{(k-1)}=2kpL_{k-1}+k(k-1)p'(p^{k-1})^{(k-2)}.$$

代入 $(p^{k-1})^{(k-2)}$ 和 $(p^k)^{(k-1)}$，得

$$\frac{1}{2k}\Big(L_{k+1}-(k+1)p'L_k\Big)=2kpL_{k-1}+\frac{kp'}{2k}\Big(L_k-kp'L_{k-1}\Big).$$

在上式中令 $k=n-1$，$p'=2x-(a+b)$，展开并化简，得

$$L_n=(2n-1)(2x-a-b)L_{n-1}-(n-1)^2(b-a)^2L_{n-2}.$$

设 $L_i(x)=\sum\limits_{j=0}^{i}l_{ij}x^j$. 规定 $l_{ij}=0$，$\forall i<0$ 或 $j<0$，则 $L_0,\cdots,L_n$ 的系数矩阵 $(l_{ij})_{0\leqslant j\leqslant i\leqslant n}$ 满足

$$l_{00}=1,\quad l_{ij}=(4i-2)l_{i-1,j-1}-(2i-1)(a+b)l_{i-1,j}-(i-1)^2(b-a)^2l_{i-2,j},\quad i\geqslant 1.$$

例 2.11. 设 $V=\mathbb{R}[x]$ 上的内积 $\rho(f,g)=\displaystyle\int_{-1}^{1}\frac{f(x)g(x)}{\sqrt{1-x^2}}dx$，则 $\{c_nT_n \mid n\in\mathbb{N}\}$ 是 (V,ρ) 的标准正交基，其中 $T_n(x)=\cos(n\arccos x)$ 称为 n 次 **Chebyshev 多项式**，$c_0=\sqrt{\frac{1}{\pi}}$，$c_n=\sqrt{\frac{2}{\pi}}$，$n\geqslant 1$.

证明. 设 $m,n\in\mathbb{N}$. 换元 $x=\cos\theta$，得

$$\rho(T_m,T_n)=\int_0^{\pi}\cos(m\theta)\cos(n\theta)d\theta=\int_0^{\pi}\frac{\cos\big((m-n)\theta\big)+\cos\big((m+n)\theta\big)}{2}d\theta.$$

当 $m\neq n$ 时，$\rho(T_m,T_n)=0$. 当 $n=0$ 时，$\rho(T_n,T_n)=\pi$. 当 $n\geqslant 1$ 时，$\rho(T_n,T_n)=\dfrac{\pi}{2}$. □

n	0	1	2	3	4	5	6
T_n	1	x	$2x^2-1$	$4x^3-3x$	$8x^4-8x^2+1$	$16x^5-20x^3+5x$	$32x^6-48x^4+18x^2-1$

设 $T_i(x)=\displaystyle\sum_{j=0}^{i}t_{ij}x^j$. 规定 $t_{ij}=0$，$\forall j<0$. 由和差化积公式，可得 $T_n=2xT_{n-1}-T_{n-2}$，$\forall n\geqslant 2$. 故 $T_0,\cdots,T_n$ 的系数矩阵 $(t_{ij})_{0\leqslant j\leqslant i\leqslant n}$ 满足

$$t_{00}=1,\quad t_{10}=0,\quad t_{11}=1,\quad t_{ij}=2t_{i-1,j-1}-t_{i-2,j},\quad i\geqslant 2.$$

例 2.12. 设 V 是 $(-\infty,\infty)$ 上周期 2π 的函数全体构成实线性空间，V 上的内积

$$\rho(f,g)=\frac{1}{\pi}\int_0^{2\pi} f(x)g(x)dx$$

则 $S=\left\{\frac{1}{\sqrt{2}},\cos x,\sin x,\cdots,\cos nx,\sin nx\right\}$ 是 (V,ρ) 中的标准正交向量组．任意 $f\in V$ 在 $\mathrm{Span}(S)$ 中的投影 $p=\frac{a_0}{2}+\sum\limits_{k=1}^{n}(a_k\cos kx+b_k\sin kx)$，其中 $a_k=\rho(f,\cos kx)$，$b_k=\rho(f,\sin kx)$.

当 f 有 2 阶连续导函数时，$a_k=O(\frac{1}{k^2})$，$b_k=O(\frac{1}{k^2})$，故 $\|f-p\|_\infty=\max\limits_{x\in\mathbb{R}}|f(x)-p(x)|=O(\frac{1}{n})$． □

例 2.12 表明，对于周期 2π 的偶函数 $f(\theta)$，可用其三角函数展开式构造 $f(\theta)$ 的一致逼近 $\sum\limits_{k=0}^{n}a_k\cos k\theta$.

对于 $f\in\mathcal{C}[-1,1]$，设 $\sum\limits_{k=0}^{n}a_k\cos k\theta$ 是 $f(\cos\theta)$ 的一致逼近，则 $\sum\limits_{k=0}^{n}a_kT_k(x)$ 是 $f(x)$ 的一致逼近.

$$a_0=\frac{1}{2\pi}\int_0^{2\pi}f(\cos\theta)d\theta=\frac{1}{\pi}\int_{-1}^{1}\frac{f(x)T_0(x)}{\sqrt{1-x^2}}dx,$$

$$a_k=\frac{1}{\pi}\int_0^{2\pi}f(\cos\theta)\cos(k\theta)d\theta=\frac{2}{\pi}\int_{-1}^{1}\frac{f(x)T_k(x)}{\sqrt{1-x^2}}dx,\quad k\geqslant 1.$$

结合例 2.11，$\sum\limits_{k=0}^{n}a_kT_k$ 就是 f 在 $\mathrm{Span}(T_0,T_1,\cdots,T_n)$ 中的投影.

假设已知离散点列 $x_k = \dfrac{2k\pi}{m}$，$y_k = f(x_k)$，$1 \leqslant k \leqslant m = 2n+1$．注意到，当 $p, q = 0, 1, \cdots, n$ 时，

$$\sum_{k=1}^{m} \cos(px_k)\sin(qx_k) = 0, \quad \sum_{k=1}^{m} \cos(px_k)\cos(qx_k) = \sum_{k=1}^{m} \sin(px_k)\sin(qx_k) = \begin{cases} \dfrac{m}{2}, & p = q \geqslant 1 \\ 0, & p \neq q \end{cases}. \quad (*)$$

以 $\{(x_k, y_k) \mid 1 \leqslant k \leqslant m\}$ 为插值点构造三角多项式插值函数

$$\phi(x) = \sum_{k=0}^{n} (\lambda_k \cos kx + \mu_k \sin kx).$$

由 (*) 式可解得

$$\lambda_0 = \frac{1}{m}\sum_{i=1}^{m} y_i, \quad \lambda_k = \frac{2}{m}\sum_{i=1}^{m} y_i \cos(kx_i), \quad \mu_k = \frac{2}{m}\sum_{i=1}^{m} y_i \sin(kx_i), \quad k \geqslant 1.$$

每个 λ_k, μ_k 可以看作是 $\rho(f, \cos kx), \rho(f, \sin kx)$ 的数值积分的结果．当 f 有 2 阶连续导函数时，

$$\lambda_k = a_k + O(\frac{1}{n^2}), \quad \mu_k = b_k + O(\frac{1}{n^2}) \qquad \Rightarrow \qquad \phi(x) = p(x) + O(\frac{1}{n^2}),\ \forall x.$$

故 $\|f - \phi\|_\infty = O(\frac{1}{n})$.

第三章　方程求根 (4 学时)

§3.1　一元方程

本节考虑一元方程的求根问题：设 $f(x)$ 是 $\mathbb{R}$ 上的连续函数，$f(x)$ 的表达形式可能未知，但是知道 $f(x)$ 在每点处的取值．已知 $f(x)=0$ 有根 $\alpha\in[a,b]$，求 α 的近似值 x 使得 $|x-\alpha|\leqslant\varepsilon$.

定理 3.1. *若连续函数 f 满足 $f(a)f(b)<0$，则存在 $\alpha\in(a,b)$ 使得 $f(\alpha)=0$.*

由定理 3.1 可得求方程根的**二分法**，其中循环次数 n 使得 $\dfrac{b-a}{2^n}<\varepsilon$.

```
x1=a; x2=b; y1=f(x1); y2=f(x2); if(y1*y2>0) return(null);
for(i=1; i<=n; i++)
{
  x3=0.5(x1+x2); y3=f(x3);
```

```
  if(y1*y3>0) (x1,y1)=(x3,y3); else (x2,y2)=(x3,y3);
}
return(x3);
```

二分法的缺点是：f 需要满足 $f(a)f(b)<0$，并且只能计算 f 的实数根.

定理 3.2. 设 $\phi:[a,b]\to[a,b]$ 有连续的导函数，并且存在正实数 $C<1$ 使得

$$|\phi'(x)|\leqslant C,\quad \forall x\in[a,b]$$

则对于任意 $x_0\in[a,b]$，递推数列 $x_{n+1}=\phi(x_n)$ 收敛到 $\phi(x)$ 的唯一不动点 α.

证明. $\forall x,y\in[a,b]$，$|\phi(x)-\phi(y)|=|\phi'(\xi)(x-y)|\leqslant C|x-y|$. 记 $\phi_n=\underbrace{\phi\circ\cdots\circ\phi}_{n个}$，则

$$|\phi_n(x)-\phi_n(y)|\leqslant C|\phi_{n-1}(x)-\phi_{n-1}(y)|\leqslant\cdots\leqslant C^n|x-y|\to 0.$$

故 $\lim\limits_{n\to\infty}\phi_n(x)\equiv\alpha$ 是常值函数. □

根据定理 3.2，可得方程求根的**不动点法**或**迭代法**. 设 α 是方程的根. 构造 $\phi(x)$ 使得 $\phi(\alpha)=\alpha$ 并且 $|\phi'(\alpha)|<1$. 当 $|x_0-\alpha|$ 足够小时，递推数列 $x_{n+1}=\phi(x_n)$ 必收敛到 α. 如何构造具有良好性质的 $\phi(x)$，是此类方法的核心.

下面是两个常用的迭代公式.

- **Newton 迭代法** (或**切线法**)：$x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$.
- **弦截法** (或**割线法**)：$x_{n+1} = x_n - \dfrac{f(x_n)}{\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}}$.

Newton 迭代法的设计思想为

$$0 = f(\alpha) \approx f(x_n) + f'(x_n)(\alpha - x_n) \qquad \Rightarrow \qquad \alpha \approx x_n - \frac{f(x_n)}{f'(x_n)}.$$

当 $f(x)$ 的表达式未知时，可用第六章中数值微分的方法计算 $f'(x)$，也可用 $\dfrac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$ 近似 $f'(x_n)$，由此得到弦截法．Newton 迭代法和弦截法还有另一种解释．

$$g(x) = f(x_n) + f'(x_n)(x - x_n) \quad \text{和} \quad h(x) = f(x_n) + \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}(x - x_n)$$

分别是 $f(x)$ 的 Hermite 插值和 Newton 插值函数．用 $g(x) = 0$ 和 $h(x) = 0$ 的根近似 $f(x) = 0$ 的根．

定义 3.1. 设 $\lim\limits_{n\to+\infty} x_n = \alpha$，$p \geqslant 1$．若存在常数 C 使得

$$|x_{n+1} - \alpha| < C|x_n - \alpha|^p, \quad \forall n$$

则称数列 $\{x_n\}$ **有 p 阶收敛阶数**.

定理 3.3. 设 α 是 $f(x)=0$ 的 m 重根，即 $f(x)=(x-\alpha)^m g(x)$，其中 $g(x)$ 有连续导函数且 $g(\alpha)\neq 0$.

1. 当 $m=1$ 时，*Newton* 迭代的收敛阶数 $p=2$，弦截法迭代的收敛阶数 $p=\frac{1+\sqrt{5}}{2}$.

2. 当 $m\geqslant 2$ 时，*Newton* 迭代和弦截法迭代的收敛阶数都是 *1*.

证明. 对于 Newton 迭代法，

$$x_{n+1}-\alpha=\left(1-\frac{g(x_n)}{mg(x_n)+(x_n-\alpha)g'(x_n)}\right)(x_n-\alpha).$$

- 当 $m=1$ 时，$x_{n+1}-\alpha\approx\dfrac{g'(\alpha)}{g(\alpha)}(x_n-\alpha)^2$，收敛阶数 $p=2$.
- 当 $m\geqslant 2$ 时，$x_{n+1}-\alpha\approx\dfrac{m-1}{m}(x_n-\alpha)$，收敛阶数 $p=1$.

对于弦截法，

$$x_{n+1}-\alpha=\frac{f(x_n)(x_{n-1}-\alpha)-f(x_{n-1})(x_n-\alpha)}{f(x_n)-f(x_{n-1})}=\frac{\frac{f(x_n)}{x_n-\alpha}-\frac{f(x_{n-1})}{x_{n-1}-\alpha}}{f(x_n)-f(x_{n-1})}(x_n-\alpha)(x_{n-1}-\alpha).$$

当 $m=1$ 时，$x_{n+1}-\alpha\approx\dfrac{g'(\alpha)}{g(\alpha)}(x_n-\alpha)(x_{n-1}-\alpha)$，收敛阶数 p 满足 $p^2=p+1$，得 $p=\dfrac{1+\sqrt{5}}{2}$.

当 $m\geqslant 2$ 时，$x_{n+1}-\alpha\approx\dfrac{(x_n-\alpha)^{m-1}-(x_{n-1}-\alpha)^{m-1}}{(x_n-\alpha)^m-(x_{n-1}-\alpha)^m}(x_n-\alpha)(x_{n-1}-\alpha)$，收敛阶数 $p=1$. □

当 m 已知时，Newton 迭代公式可修正为

$$x_{n+1}=x_n-m\frac{f(x_n)}{f'(x_n)} \qquad \Rightarrow \qquad x_{n+1}-\alpha=\frac{g'(x_n)(x_n-\alpha)^2}{mg(x_n)+(x_n-\alpha)g'(x_n)}\approx\frac{g'(x_n)}{mg(x_n)}(x_n-\alpha)^2.$$

修正的 Newton 迭代的收敛阶数是 2．弦截法没有类似的修正公式.

当 m 未知时，如下的 **Aitken 外推法**可加速迭代 $x_{n+1}=\phi(x_n)$ 的收敛．记

$$y_{n+1}=\frac{x_{n+1}x_{n-1}-x_n^2}{x_{n+1}-2x_n+x_{n-1}} \quad \text{或} \quad y_{n+1}=\varphi(y_n),\ \varphi(x)=\frac{x\phi(\phi(x))-(\phi(x))^2}{\phi(\phi(x))-2\phi(x)+x}.$$

情形 1：$p=1$．设 $x_{n+1}-\alpha=C_1(x_n-\alpha)+C_2(x_n-\alpha)^q+o(|x_n-\alpha|^q)$，其中 $|C_1|<1$，$C_2\neq 0$，$q>1$.

$$x_n-\alpha\approx C_1(x_{n-1}-\alpha),\ \ x_{n+1}-\alpha\approx C_1(x_n-\alpha) \qquad \Rightarrow \qquad C_1\approx\frac{x_n-x_{n+1}}{x_{n-1}-x_n}$$

$$\Rightarrow \qquad y_{n+1}-\alpha=\frac{(x_{n-1}-x_n)(x_{n+1}-\alpha)-(x_n-x_{n+1})(x_n-\alpha)}{(x_{n-1}-x_n)-(x_n-x_{n+1})}$$

$$=\frac{x_{n+1}-\alpha-\frac{x_{n+1}-x_n}{x_n-x_{n-1}}(x_n-\alpha)}{1-\frac{x_{n+1}-x_n}{x_n-x_{n-1}}}\approx\frac{C_2}{1-C_1}(x_n-\alpha)^q.$$

对于 Newton 迭代，$C_1=\frac{m-1}{m}$．故可先估计出 C_1 和 m，再使用修正的 Newton 迭代公式.

情形 2：$p > 1$．设 $x_{n+1} - \alpha = C(x_n - \alpha)^p + o(|x_n - \alpha|^p)$，其中 $C \neq 0$．

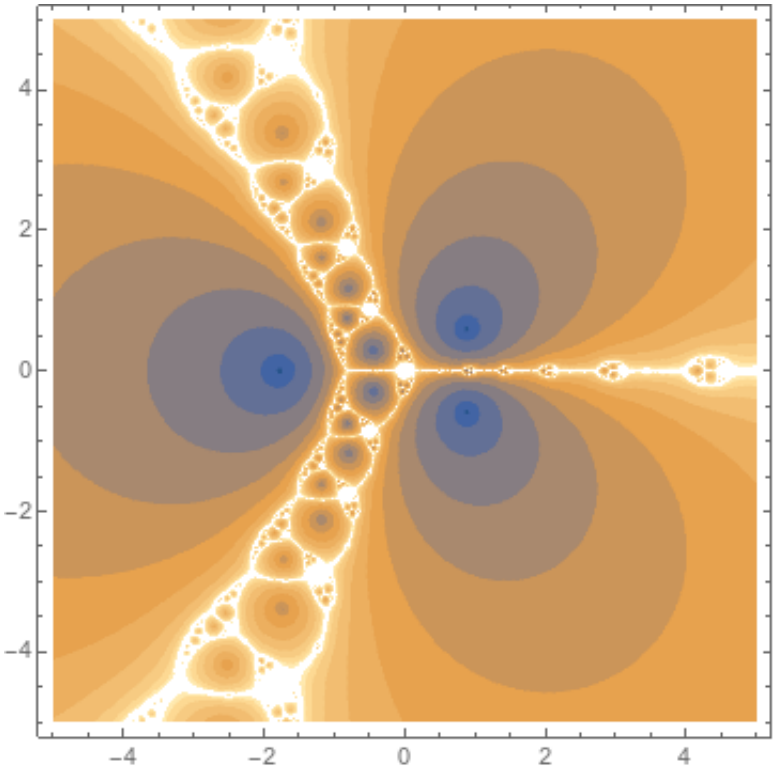
$$x_n - \alpha \approx C(x_{n-1} - \alpha)^p, \quad x_{n+1} - \alpha \approx C(x_n - \alpha)^p \approx C^{p+1}(x_{n-1} - \alpha)^{p^2}$$

$$\Rightarrow \qquad y_{n+1} - \alpha = \frac{(x_{n+1} - \alpha)(x_{n-1} - \alpha) - (x_n - \alpha)^2}{(x_{n+1} - \alpha) + (x_{n-1} - \alpha) - 2(x_n - \alpha)} \approx \frac{-(x_n - \alpha)^2}{x_{n-1} - \alpha} \approx -C^2(x_{n-1} - \alpha)^{2p-1}$$

当 $p > 1$ 时，$p^2 > 2p - 1$．故 Aitken 外推法仅对线性收敛的加速效果明显．

Newton 迭代法和弦截法都要求初值必须接近方程的根，否则迭代有可能发散或陷入循环．

例 3.1. $f(x) = x^3 - 2x + 2$ 有三个根 $\alpha_1 \approx -1.76929$，$\alpha_{2,3} \approx 0.884646 \pm 0.589743\,\mathrm{i}$．当 $x_0 \approx 0$ 或 1 时，Newton 迭代 $x_{n+1} = \dfrac{2x_n^3 - 2}{3x_n^2 - 2}$ 在 0 和 1 之间循环．右图反映了迭代关于初值的收敛情况，颜色越深表示所需迭代步数越少，白色表示不收敛． □



§3.2 多元方程组

Newton 迭代法的思想可以很容易地应用于求解 n 个方程和 n 个变元的方程组

$$\{f_i(x_1,\cdots,x_n)=0 \mid i=1,2,\cdots,n\}.$$

设 $\alpha=(\alpha_1,\cdots,\alpha_n)$ 是方程组的一个根．记 $\boldsymbol{x}=(x_1,\cdots,x_n)$，$F(\boldsymbol{x})=\big(f_1(\boldsymbol{x}),\cdots,f_n(\boldsymbol{x})\big)$，则

$$\begin{pmatrix}0\\ \vdots\\ 0\end{pmatrix}=\begin{pmatrix}f_1(\alpha)\\ \vdots\\ f_n(\alpha)\end{pmatrix}\approx\begin{pmatrix}f_1(\boldsymbol{x})\\ \vdots\\ f_n(\boldsymbol{x})\end{pmatrix}+\begin{pmatrix}\frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n}\\ \vdots & \cdots & \vdots\\ \frac{\partial f_n(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_n(\boldsymbol{x})}{\partial x_n}\end{pmatrix}\begin{pmatrix}\alpha_1-x_1\\ \vdots\\ \alpha_n-x_n\end{pmatrix}$$

$$\Rightarrow\qquad\begin{pmatrix}\alpha_1\\ \vdots\\ \alpha_n\end{pmatrix}\approx\begin{pmatrix}x_1\\ \vdots\\ x_n\end{pmatrix}-\begin{pmatrix}\frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n}\\ \vdots & \cdots & \vdots\\ \frac{\partial f_n(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_n(\boldsymbol{x})}{\partial x_n}\end{pmatrix}^{-1}\begin{pmatrix}f_1(\boldsymbol{x})\\ \vdots\\ f_n(\boldsymbol{x})\end{pmatrix}.$$

由此，得多元方程组的 Newton 迭代公式

$$\Phi(\boldsymbol{x})=\boldsymbol{x}-\left(\frac{\partial F(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^{-1}F(\boldsymbol{x}). \tag{3.1}$$

n 阶方阵 $\left(\frac{\partial F(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)$ 称为 $F(\boldsymbol{x})$ 的 **Jacobi 矩阵.**

例 3.2. 求 n 次复系数多项式 $f(x)=a_0+a_1x+\cdots+a_{n-1}x^{n-1}+x^n$ 的所有复数根.

方法 1. 设 $\alpha_1,\cdots,\alpha_n$ 是 $f(x)$ 的所有根. 给定 $z_1,\cdots,z_n$，则

$$f(x)=\prod_{i=1}^{n}(x-\alpha_i)\approx\prod_{i=1}^{n}(x-z_i)+\sum_{i=1}^{n}(z_i-\alpha_i)\prod_{j\neq i}(x-z_j)$$

$$\Rightarrow\qquad f(z_i)\approx(z_i-\alpha_i)\prod_{j\neq i}(z_i-z_j)\qquad\Rightarrow\qquad \alpha_i\approx z_i-\frac{f(z_i)}{\prod\limits_{j\neq i}(z_i-z_j)}.$$

由此，得 **Weierstrauss 方法** 或 **Durand-Kerner 方法** 的迭代公式

$$z_i^{new}=z_i-\frac{f(z_i)}{\prod\limits_{j\neq i}(z_i-z_j)},\quad 1\leqslant i\leqslant n. \tag{3.2}$$

方法 2. 设 $\boldsymbol{z}=(z_1,\cdots,z_n)$，$F(\boldsymbol{z})=\big(\cdots,(-1)^k\sigma_k(\boldsymbol{z})-a_{n-k},\cdots\big)$ 是 $\prod\limits_{i=1}^{n}(x-z_i)-f(x)$ 的系数向量，其中 $\sigma_k(\boldsymbol{z})=\sum\limits_{1\leqslant i_1<\cdots<i_k\leqslant n}z_{i_1}\cdots z_{i_k}$ 是 n 元 k 次基本对称多项式，$1\leqslant k\leqslant n$. 对 $F(\boldsymbol{z})$ 应用 Newton 迭代公式 (3.1) 也可以得到 (3.2) 式. 不详细证明. 例如，$n=2$，$f(x)=a_0+a_1x+x^2$，$(x-z_1)(x-z_2)-f(x)=(-z_1-z_2-a_1)x+(z_1z_2-a_0)$，$F(\boldsymbol{z})=(-z_1-z_2-a_1,z_1z_2-a_0)$，则

$$\Phi(\boldsymbol{z})=\begin{pmatrix}z_1\\ z_2\end{pmatrix}-\begin{pmatrix}-1 & -1\\ z_2 & z_1\end{pmatrix}^{-1}\begin{pmatrix}-z_1-z_2-a_1\\ z_1z_2-a_0\end{pmatrix}=\begin{pmatrix}z_1\\ z_2\end{pmatrix}-\begin{pmatrix}\frac{f(z_1)}{z_1-z_2}\\ \frac{f(z_2)}{z_2-z_1}\end{pmatrix}.$$ □

第四章　数值优化 (6 学时)

如下形式的问题常称为**优化问题**，其中 f 称为**目标函数**，S 称为**约束条件**，α 称为**最优解**.

给定 $S \subset \mathbb{R}^n$ 和连续函数 $f : S \to \mathbb{R}$，求 $\alpha \in S$ 使得 $f(\alpha)$ 取得最小值（或最大值）.

通常不易求得优化问题的最优解，只能求得局部最优解，称为**极值点**.

例如，在对数据作最小二乘拟合的时候，求参数向量 α 使得 $f(\alpha) = \sum_{i=1}^{m} \big(\phi(\alpha, x_i) - y_i\big)^2$ 最小. 此时，$\phi(\alpha, x)$ 可能没有明确的解析表达式，计算 $f(\alpha)$ 的复杂度可能非常大.

§4.1　无约束优化

定理 4.1. *设 $f : \mathbb{R}^n \to \mathbb{R}$ 有 2 阶连续偏导数，$\nabla f(\boldsymbol{x}) = \left(\dfrac{\partial f}{\partial x_i}\right)$ 和 $H(\boldsymbol{x}) = \left(\dfrac{\partial^2 f}{\partial x_i \partial x_j}\right)$ 分别是 f 在 $\boldsymbol{x}$ 处的梯度向量和 Hesse 矩阵.*

1. 若 α 是 $f(\boldsymbol{x})$ 的极值点，则 $\nabla f(\alpha)=\mathbf{0}$.

2. 若 $\nabla f(\alpha)=\mathbf{0}$ 并且 $H(\alpha)$ 是正定的，则 α 是极小值点.

证明. 根据 Taylor 展开式，当 $\boldsymbol{x}\to\mathbf{0}$ 时，

$$f(\alpha+\boldsymbol{x})=f(\alpha)+\boldsymbol{x}^T\nabla f(\alpha)+\frac{1}{2}\boldsymbol{x}^T H(\alpha)\boldsymbol{x}+o(\|\boldsymbol{x}\|^2).$$

1. 若 $\nabla f(\alpha)\neq\mathbf{0}$，则存在 $\boldsymbol{x}\to\mathbf{0}$ 使得 $f(\alpha-\boldsymbol{x})<f(\alpha)<f(\alpha+\boldsymbol{x})$，得 α 不是极值点.
2. 设 $\nabla f(\alpha)=\mathbf{0}$ 并且 $H(\alpha)$ 是正定的. 当 $\boldsymbol{x}\to\mathbf{0}$ 时，$\boldsymbol{x}^T H(\alpha)\boldsymbol{x}>0$，$f(\alpha+\boldsymbol{x})>f(\alpha)$.
   故 α 是极小值点.

□

当 $H(\boldsymbol{x})$ 具有良好性质，初值 $\boldsymbol{x}_0\approx\alpha$ 时，可用 Newton 迭代法求 $\nabla f(\boldsymbol{x})=\mathbf{0}$ 的根 α.

$$\Phi(\boldsymbol{x})=\boldsymbol{x}-H^{-1}(\boldsymbol{x})\cdot\nabla f(\boldsymbol{x})$$

当 $\|\boldsymbol{x}_0-\alpha\|$ 较大时，Newton 迭代可能发散或收敛到其他驻点. 当 $f(\boldsymbol{x})$ 的表达式未知时，数值计算 ∇f 和 H 可能有较大的误差，使得 Newton 迭代的实际表现达不到预期的效果.

下面是几种搜索极小值点的方法．首先考虑 1 维空间中的搜索问题．假设存在 $\alpha \in [a,b]$ 使得连续函数 $f(x)$ 在 $[a,\alpha]$ 上严格单调减，在 $[\alpha,b]$ 上严格单调增．即 $f(x)$ 是 $[a,b]$ 上的**单谷 (峰)** 函数．

例 4.1 (**黄金分割法**). 输入 a,b 满足 $a<b$，输出 $f(x)$ 在 $[a,b]$ 中的极小值点 α.

取 $x_1=\lambda a+(1-\lambda)b$，$x_2=(1-\lambda)a+\lambda b$，其中 $\lambda=\frac{\sqrt{5}-1}{2}$.

- 若 $f(a)\leqslant f(x_1)$，则 $\alpha\in[a,x_1]$，把 (a,b) 换成 (a,x_1)，开始下一轮搜索.
- 若 $f(b)\leqslant f(x_2)$，则 $\alpha\in[x_2,b]$，把 (a,b) 换成 (x_2,b)，开始下一轮搜索.
- 若 $f(a)>f(x_1)<f(x_2)$，则 $\alpha\in[a,x_2]$，把 (a,b) 换成 (a,x_2)，开始下一轮搜索.
- 若 $f(x_1)>f(x_2)<f(b)$，则 $\alpha\in[x_1,b]$，把 (a,b) 换成 (x_1,b)，开始下一轮搜索.
- 若 $f(x_1)=f(x_2)$，则 $\alpha\in[x_1,x_2]$，把 (a,b) 换成 (x_1,x_2)，开始下一轮搜索.

经过每次搜索，区间的长度被缩短为原长度的至多 λ 倍．收敛速度是线性的，与二分法类似． □

取 $\lambda=\frac{\sqrt{5}-1}{2}$ 而不是 $\lambda=\frac{1}{3}$，可以重复利用已搜索过的节点，减少函数值的计算.

在搜索过程中，仅仅比较函数值的大小，缺乏对于函数值的有效利用.

例 4.2 (抛物线法). 输入 a, b, c 满足 $a < b < c$，输出 $f(x)$ 在 $[a, c]$ 中的极小值点 α.

- 若 $f(a) \leqslant f(b) < f(c)$，则 $\alpha \in [a, b]$，把 (a, b, c) 换成 $(a, \frac{a+b}{2}, b)$，开始下一轮搜索.
- 若 $f(a) > f(b) \geqslant f(c)$，则 $\alpha \in [b, c]$，把 (a, b, c) 换成 $(b, \frac{b+c}{2}, c)$，开始下一轮搜索.

下设 $f(a) > f(b) < f(c)$. 以 a, b, c 为节点构造插值函数

$$\phi(x) = f(a) + f[a, c](x - a) + f[a, b, c](x - a)(x - c).$$

$\phi(x)$ 的最小值点 $d = \frac{a+b}{2} - \frac{f[a,b]}{2f[a,b,c]} > \frac{a+b}{2}$. 同理，$d = \frac{b+c}{2} - \frac{f[b,c]}{2f[a,b,c]} < \frac{b+c}{2}$.

- 若 $d \approx b$，则随机扰动 b，把 (a, b, c) 换成 (a, b', c)，开始下一轮搜索.
- 若 $f(d) < f(b)$ 且 $d < b$，则 $\alpha \in [a, b]$，把 (a, b, c) 换成 (a, d, b) 开始下一轮搜索.
- 若 $f(d) < f(b)$ 且 $d > b$，则 $\alpha \in [b, c]$，把 (a, b, c) 换成 (b, d, c)，开始下一轮搜索.
- 若 $f(d) > f(b)$ 且 $d < b$，则 $\alpha \in [d, c]$，把 (a, b, c) 换成 (d, b, c) 开始下一轮搜索.
- 若 $f(d) > f(b)$ 且 $d > b$，则 $\alpha \in [a, d]$，把 (a, b, c) 换成 (a, b, d)，开始下一轮搜索.

当 $a = b$ 或 $b = c$ 时，停止搜索，输出 b. □

尽管搜索法的收敛速度慢，但是保证收敛，通常是为快速收敛的方法提供良好的初值. 多种优化方法需结合起来使用. 可参考《华罗庚文集：应用数学卷 II》，杨德庄主编，科学出版社，2010 年.

前面介绍了 1 维空间中的搜索方法，下面介绍 n 维空间中的搜索方法.

例 4.3. 设 $f:\mathbb{R}^n\to\mathbb{R}$ 有 1 或 2 阶连续偏导数. 输入点 $\boldsymbol{p}$，输出 f 在 $\boldsymbol{p}$ 附近的最小值点.

① 选择一个搜索方向 $\boldsymbol{v}$，用 1 维搜索方法求 $g(t)=f(\boldsymbol{p}+t\boldsymbol{v})$ 在 $t=0$ 附近的极小值点 s.

② 若 $\|s\boldsymbol{v}\|>\varepsilon$，则把 $\boldsymbol{p}$ 换成 $\boldsymbol{p}+s\boldsymbol{v}$；否则，换个搜索方向 $\boldsymbol{v}$. `goto` ①，开始新一轮搜索.

③ 当搜索不到 $f(\boldsymbol{x})<f(\boldsymbol{p})$ 时，结束搜索，输出 $\boldsymbol{p}$.

其中搜索方向 $\boldsymbol{v}$ 可有多种选择方式. 例如，

- $\boldsymbol{v}=\boldsymbol{e}_i$ 是沿坐标轴方向的单位向量. 每次调整一个变量 x_i.
- $\boldsymbol{v}=-\nabla f(\boldsymbol{p})$. 由于 $f(\boldsymbol{x})$ 沿梯度反方向下降最快，这种方法称为**最速下降法**.
- $\boldsymbol{v}=-H^{-1}(\boldsymbol{p})\cdot\nabla f(\boldsymbol{p})$ 是 Newton 迭代的增量. 这种方法称为**阻尼 Newton 法**.
- $\boldsymbol{v}=-\widetilde{H}^{-1}\cdot\nabla f(\boldsymbol{p})$，其中 $\widetilde{H}\approx H(\boldsymbol{p})$. 这种方法称为**拟 Newton 法**.
- $\boldsymbol{v}_k=-\nabla f(\boldsymbol{p})-\sum\limits_{i=1}^{r}\lambda_i\boldsymbol{v}_{k-i}$ 与 $\{\boldsymbol{v}_{k-1},\cdots,\boldsymbol{v}_{k-r}\}$ 都正交. 这种方法称为**共轭梯度法**.

阻尼 Newton 法是为了增加 Newton 迭代的稳健性. 拟 Newton 法是为了解决 Hesse 矩阵的计算问题. 共轭梯度法是为了解决搜索方向反复震荡问题. 共轭梯度法要求 s 必须是 $g(t)$ 的极小值点，其他方法只需 $g(s)<g(0)$ 即可. □

§4.2 约束优化

考虑带约束条件 S 的优化问题．S 通常由一些等式和不等式定义．如果极值点落在 S 的内部，则问题等价于无约束优化．如果极值点落在 S 的边界上，则约束条件可化为等式型．故考虑如下形式的约束优化问题.

$$\text{设 } f \text{ 是 } S \to \mathbb{R} \text{ 的连续函数，求 } \alpha \in S \text{ 使得 } f(\alpha) = \min_{\boldsymbol{x} \in S} f(\boldsymbol{x}).$$

其中 $S = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_1(\boldsymbol{x}) = \cdots = g_m(\boldsymbol{x}) = 0\}$，$g_1, \cdots, g_m$ 都是 $\mathbb{R}^n \to \mathbb{R}$ 的连续函数，$m < n$.

定理 4.2. *设 $f, g_1, \cdots, g_m$ 都是 $\mathbb{R}^n \to \mathbb{R}$ 的连续函数，f 有 2 阶连续偏导数，$\nabla f(\boldsymbol{x})$ 和 $H(\boldsymbol{x})$ 分别是 f 在 $\boldsymbol{x}$ 处的梯度向量和 Hesse 矩阵，每个 g_i 有连续偏导数，$\nabla g_i(\boldsymbol{x})$ 是 g_i 在 $\boldsymbol{x}$ 处的梯度向量.*

1. 若 α 是 $f(\boldsymbol{x})$ 在约束 $\boldsymbol{x} \in S$ 下的极值点，则 $\nabla f(\alpha) \in V = \mathrm{Span}(\nabla g_1(\alpha), \cdots, \nabla g_m(\alpha))$.

2. 若 $\alpha \in S$，$\nabla f(\alpha) \in V$ 并且对于任意非零向量 $v \perp V$ 都有 $v^T H(\alpha) v > 0$，则 α 是极小值点.

证明. 根据隐函数定理，当 $\boldsymbol{x}$ 在 S 中变化时，$d\boldsymbol{x} \perp \nabla g_i(\boldsymbol{x})$，$\forall i$．根据 Taylor 展开式，当 $\|\boldsymbol{x}\| \to 0$ 时，

$$f(\alpha + \boldsymbol{x}) = f(\alpha) + \boldsymbol{x}^T \nabla f(\alpha) + \frac{1}{2} \boldsymbol{x}^T H(\alpha) \boldsymbol{x} + o(\|\boldsymbol{x}\|^2).$$

1. 若 α 是极值点，则对于任意 $v \perp V$ 都有 $v \perp \nabla f(\alpha)$. 故 $\nabla f(\alpha) \in V$.

2. 对于任意充分小的非零向量 $\boldsymbol{x} \perp V$ 都有 $f(\alpha + \boldsymbol{x}) > f(\alpha)$. 故 α 是条件极小值点.

□

根据定理 4.2，可设 $\nabla f(\alpha) = \lambda_1 \nabla g_1(\alpha) + \cdots + \lambda_m \nabla g_m(\alpha)$. 结合 $g_1(\alpha) = \cdots = g_m(\alpha) = 0$，共有 $m + n$ 个方程和 $m + n$ 个变元. 这种方法可以看作是求 **Lagrange 函数**

$$F(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \lambda_1 g_1(\boldsymbol{x}) - \cdots - \lambda_m g_m(\boldsymbol{x})$$

的驻点，常称为 **Lagrange 乘子法**，$\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_m)$ 称为 **Lagrange 乘子**.

用 Newton 迭代法求 $\nabla F(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{0}$ 的根，得

$$\begin{pmatrix} \boldsymbol{x}^{new} \\ \boldsymbol{\lambda}^{new} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{pmatrix} - \begin{pmatrix} H & -J^T \\ -J & O \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{h} \\ -\boldsymbol{g} \end{pmatrix},$$

$$H = \left(\frac{\partial^2 F(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial x_i \partial x_j} \right) \in \mathbb{R}^{n \times n},\ J = \left(\frac{\partial g_i(\boldsymbol{x})}{\partial x_j} \right) \in \mathbb{R}^{m \times n},\ \boldsymbol{h} = \left(\frac{\partial F(\boldsymbol{x})}{\partial x_i} \right) \in \mathbb{R}^n,\ \boldsymbol{g} = \Big(g_i(\boldsymbol{x}) \Big) \in \mathbb{R}^m.$$

为了提高算法的稳健性，可以把 Lagrange 函数 $F(\boldsymbol{x},\boldsymbol{\lambda})$ 换成某种**惩罚函数**

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + \mu\left(g_1^2(\boldsymbol{x}) + \cdots + g_m^2(\boldsymbol{x})\right)$$

其中 $\mu > 0$ 称为**惩罚参数**．$F(\boldsymbol{x})$ 在“极小化 $f(\boldsymbol{x})$”和“要求 $g_i(\boldsymbol{x}) = 0$”之间达成平衡．可以通过 $F(\boldsymbol{x})$ 的无约束极小值点寻找 $f(\boldsymbol{x})$ 的条件极小值点．

定理 4.3. *设 $f, g_1, \cdots, g_m$ 都是 $\mathbb{R}^n \to \mathbb{R}$ 的连续函数，并且 f 有下界．当 $\mu \to +\infty$ 时，惩罚函数 $F(\boldsymbol{x})$ 的无约束极小值点 β_μ 收敛到 $f(\boldsymbol{x})$ 在约束 $\boldsymbol{x} \in S$ 下的条件极小值点．*

证明. 对于任意 $\alpha \in S$，由 $F(\beta_\mu) \leqslant F(\alpha)$ 可得 $\sum\limits_{i=1}^{m} g_i^2(\beta_\mu) \leqslant \frac{1}{\mu}(f(\alpha) - f(\beta_\mu))$．故当 $\mu \to +\infty$ 时，$\mathrm{distance}(\beta_\mu, S) \to 0$．再由 $f(\beta_\mu) \leqslant f(\alpha)$，$\forall \alpha \in S$，得 β_μ 的极限点是 $f(\boldsymbol{x})$ 的条件极小值点． □

随着 μ 的增大，惩罚函数 $F(\boldsymbol{x})$ 的 Hesse 矩阵越来越病态，不易精确求得 β_μ．

- 一种处理方法是对于一系列递增的 $\{\mu_n\}$，求解无约束优化问题 P_n，并把 P_n 的结果 β_{μ_n} 作为 P_{n+1} 的初值．这种方法通常称为**延拓方法**或**同伦方法**．

- 另一种处理方法是把 Lagrange 乘子法和惩罚函数法结合起来，定义**增广的 Lagrange 函数**

$$F(\boldsymbol{x},\boldsymbol{\lambda}) = f(\boldsymbol{x}) - \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) + \mu \sum_{i=1}^{m} g_i^2(\boldsymbol{x}), \quad \lambda > 0.$$

用 Newton 迭代法求 $\nabla F(\boldsymbol{x},\boldsymbol{\lambda}) = \boldsymbol{0}$ 的根，得

$$\begin{pmatrix} \boldsymbol{x}^{new} \\ \boldsymbol{\lambda}^{new} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{pmatrix} - \begin{pmatrix} H & -J^T \\ -J & O \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{h} \\ -\boldsymbol{g} \end{pmatrix},$$

$H = \left(\dfrac{\partial^2 F(\boldsymbol{x},\boldsymbol{\lambda})}{\partial x_i \partial x_j}\right) \in \mathbb{R}^{n\times n}$, $J = \left(\dfrac{\partial g_i(\boldsymbol{x})}{\partial x_j}\right) \in \mathbb{R}^{m\times n}$, $\boldsymbol{h} = \left(\dfrac{\partial F(\boldsymbol{x})}{\partial x_i}\right) \in \mathbb{R}^{n\times 1}$, $\boldsymbol{g} = \Big(g_i(\boldsymbol{x})\Big) \in \mathbb{R}^{m\times 1}$.

设 H_1 是 $F_1(\boldsymbol{x}) = f(\boldsymbol{x}) - \sum\limits_{i=1}^{m} \lambda_i g_i(\boldsymbol{x})$ 的 Hesse 矩阵，H_2 是 $F_2(\boldsymbol{x}) = \sum\limits_{i=1}^{m} g_i^2(\boldsymbol{x})$ 的 Hesse 矩阵，则 $H = H_1 + \mu H_2$．当 H_2 是正定方阵时，适当选取 μ 有助于改善 H 和 $\begin{pmatrix} H & -J^T \\ -J & O \end{pmatrix}$ 的病态性．

- 其他形式的辅助函数 $F(\boldsymbol{x},\boldsymbol{\lambda})$ 也常被用于求解带约束条件的优化问题．此处不详细叙述．
- 还可以对约束条件 S 和目标函数 f 作局部近似线性化，把约束优化问题化为下一节的线性规划问题．

§4.3 线性规划

线性规划是一类特殊的优化问题，其目标函数是线性函数，约束条件是线性的等式或不等式.

定义 4.1. 线性规划问题的**一般形式**为：给定实数 a_{ij}, b_i, c_i，求实数 $x_1, \cdots, x_n$ 满足下列约束条件

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n & \geqslant = \leqslant \ b_1 \\ \vdots & \vdots \quad \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n & \geqslant = \leqslant \ b_m \end{cases}$$

并且使得 $f(x_1, \cdots, x_n) = c_1x_1 + \cdots + c_nx_n$ 取得最小值 (或最大值). 通常把约束条件定义的区域 $S \subset \mathbb{R}^n$ 称为**可行域**，S 中的点称为线性规划问题的**可行解**，使 f 取得最值的可行解称为**最优解**.

例 4.4 (投入产出问题). 设某工厂用 m 种原材料 $Q_1, \cdots, Q_m$ 生产 n 种产品 $P_1, \cdots, P_n$. 每生产 1 个单位的 P_j 可获利 p_j，同时需要消耗 a_{ij} 个单位的 Q_i. 若 Q_i 的库存量为 b_i 个单位，如何用这些原材料获取最多利润?

解答. 设生产 x_j 个单位的 P_j，则原材料 Q_i 的消耗量 $y_i = \sum\limits_{j=1}^{n} a_{ij}x_j$，共获利 $\sum\limits_{j=1}^{n} p_jx_j$. 得线性规划问题：求 $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ 满足 $\boldsymbol{x} \geqslant 0$ 且 $A\boldsymbol{x} \leqslant \boldsymbol{b}$，使得 $\boldsymbol{p}^T\boldsymbol{x}$ 最大. □

例 4.5 (物流问题). 设某种商品有 m 个仓储地 $A_1,\cdots,A_m$ 和 n 个配送地 $B_1,\cdots,B_n$，A_i 的库存量为 a_i，B_j 的需求量为 b_j，单位商品从 A_i 运到 B_j 的费用为 c_{ij}．求满足需求且费用最少的物流方案.

解答. 设从 A_i 运到 B_j 的商品数量为 x_{ij}，则 x_{ij} 应满足 ① $\sum\limits_{j=1}^{n} x_{ij} \leqslant a_i, \forall i$，② $\sum\limits_{i=1}^{m} x_{ij} \geqslant b_j, \forall j$，物流费用为 $\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij}$．得线性规划问题：求 $X \in \mathbb{R}^{m\times n}$ 满足 $X \geqslant O$ 且 $X\mathbf{1} \leqslant \boldsymbol{a}$ 且 $X^T\mathbf{1} = \boldsymbol{b}$，使得 $\mathrm{tr}(C^TX)$ 最小，其中 $\mathbf{1} = (1,\cdots,1)^T$，$\boldsymbol{a} = (a_i) \in \mathbb{R}^{m\times 1}$，$\boldsymbol{b} = (b_i) \in \mathbb{R}^{n\times 1}$，$C = (c_{ij}) \in \mathbb{R}^{m\times n}$． □

定理 4.4. 通过引入新变量，每个线性规划问题可以转化为如下的**标准形式**.

$$\text{求 } \boldsymbol{x} \in \mathbb{R}^{n\times 1} \text{ 满足 } \boldsymbol{x} \geqslant \mathbf{0} \text{ 且 } A\boldsymbol{x} = \boldsymbol{b}, \text{ 使得 } \boldsymbol{c}^T\boldsymbol{x} \text{ 最小}. \tag{4.1}$$

其中 $A \in \mathbb{R}^{m\times n}$ 是行满秩的，$\boldsymbol{b} \in \mathbb{R}^{m\times 1}$，$\boldsymbol{c} \in \mathbb{R}^{n\times 1}$，$\alpha \geqslant \beta$ 表示 $\alpha - \beta$ 的元素都是非负实数.

证明. 对于形如 $\sum\limits_{i=1}^{n} a_{ij}x_j \geqslant b_i$ 或 $\leqslant b_i$ 的约束条件，引入新变量 $w_i = \pm\left(\sum\limits_{j=1}^{n} a_{ij}x_j - b_i\right)$，约束条件化为 $w_i \geqslant 0$ 和上述等式．还可设每个变量 $x_i = u_i - v_i$，其中 $u_i \geqslant 0$，$v_i \geqslant 0$．以这些 u_i, v_i, w_i 为变量，线性规划问题化为 (4.1) 形式．当 $\mathrm{rank}(A) < m$ 且线性方程组 $A\boldsymbol{x} = \boldsymbol{b}$ 有解时，可删去冗余方程，使得 A 是行满秩的． □

定理 4.5. 设线性规划问题 (4.1) 有最优解，$A=(\alpha_1,\cdots,\alpha_n)$ 是行满秩的，则存在极大线性无关组 $\alpha_{j_1},\cdots,\alpha_{j_m}$，使得由如下线性方程组确定的 $\boldsymbol{x}$ 是问题 (4.1) 的一个最优解.

$$x_{j_1}\alpha_{j_1}+\cdots+x_{j_m}\alpha_{j_m}=\boldsymbol{b} \text{ 并且 } x_j=0,\ \forall j\notin\{j_1,\cdots,j_m\}.$$

证明. 设 $\boldsymbol{x}$ 是问题 (4.1) 的最优解，使得 $\boldsymbol{x}$ 的非零元素最少．设 $\{j\mid x_j>0\}=\{j_1,\cdots,j_k\}$．下面证明 $\alpha_{j_1},\cdots,\alpha_{j_k}$ 线性无关，从而可以扩充为极大线性无关组 $\alpha_{j_1},\cdots,\alpha_{j_m}$，满足题设．否则，

- 存在不全为 0 的 $\lambda_1,\cdots,\lambda_k$ 使得 $\lambda_1\alpha_{j_1}+\cdots+\lambda_k\alpha_{j_k}=\boldsymbol{0}$．设 $\boldsymbol{y}=\lambda_1\boldsymbol{e}_{j_1}+\cdots+\lambda_k\boldsymbol{e}_{j_k}$，则 $A\boldsymbol{y}=\boldsymbol{0}$.
- 不妨设 $\left|\dfrac{x_{j_1}}{\lambda_1}\right|\leqslant\cdots\leqslant\left|\dfrac{x_{j_k}}{\lambda_k}\right|$．若 $\boldsymbol{c}^T\boldsymbol{y}=\delta\neq0$，则 $\boldsymbol{z}=\boldsymbol{x}-\dfrac{\delta}{|\delta|}\left|\dfrac{x_{j_1}}{\lambda_1}\right|\boldsymbol{y}$ 满足 $\boldsymbol{z}\geqslant0$ 且 $A\boldsymbol{z}=\boldsymbol{b}$ 且 $\boldsymbol{c}^T\boldsymbol{z}=\boldsymbol{c}^T\boldsymbol{x}-\left|\dfrac{x_{j_1}\delta}{\lambda_1}\right|<\boldsymbol{c}^T\boldsymbol{x}$，与 $\boldsymbol{x}$ 是最优解矛盾．故 $\boldsymbol{c}^T\boldsymbol{y}=0$.
- $\boldsymbol{w}=\boldsymbol{x}-\dfrac{x_{j_1}}{\lambda_1}\boldsymbol{y}$ 满足 $w_{j_1}=0$ 且 $\boldsymbol{w}\geqslant0$ 且 $A\boldsymbol{w}=\boldsymbol{b}$ 且 $\boldsymbol{c}^T\boldsymbol{w}=\boldsymbol{c}^T\boldsymbol{x}$，与 $\boldsymbol{x}$ 的零元素最多矛盾.

□

根据定理 4.5及其证明，可得求解线性规划问题 (4.1) 的单纯形方法.

例 4.6 (**单纯形方法**). 输入行满秩矩阵 $A=(\alpha_1,\cdots,\alpha_n)\in\mathbb{R}^{m\times n}$，$\boldsymbol{b}\in\mathbb{R}^{m\times 1}$，$\boldsymbol{c}\in\mathbb{R}^{n\times 1}$，
输出线性无关组 $\alpha_{j_1},\cdots,\alpha_{j_m}$ 和问题 (4.1) 的一个最优解 $\boldsymbol{x}$ 满足 $x_j=0$，$\forall j\notin\{j_1,\cdots,j_m\}$.

① 若已知线性无关组 $\alpha_{j_1},\cdots,\alpha_{j_m}$ 和 $\boldsymbol{x}\geqslant\boldsymbol{0}$ 满足 $A\boldsymbol{x}=\boldsymbol{b}$ 且 $x_j=0$，$\forall j\notin\{j_1,\cdots,j_m\}$，则 goto ②.
否则，不妨设 $\boldsymbol{b}\geqslant\boldsymbol{0}$，先用单纯形方法求解如下线性规划问题.

$$\text{求 }\boldsymbol{y}\in\mathbb{R}^{(m+n)\times 1}\text{ 满足 }\boldsymbol{y}\geqslant\boldsymbol{0}\text{ 且 }\begin{pmatrix}I_m & A\end{pmatrix}\boldsymbol{y}=\boldsymbol{b}\text{，使得 }y_1+\cdots+y_m\text{ 最小.}\qquad(*)$$

问题 (*) 有显然的可行解 $\begin{pmatrix}\boldsymbol{b}\\ \boldsymbol{0}\end{pmatrix}$，对应 $(j_1,\cdots,j_m)=(1,\cdots,m)$. 若问题 (*) 有最优解 $\boldsymbol{y}$ 满足 $y_1=\cdots=y_m=0$，则存在线性无关组 $\alpha_{j_1},\cdots,\alpha_{j_m}$，使得 $\boldsymbol{x}=(y_{m+1},\cdots,y_{m+n})^T$ 满足 $A\boldsymbol{x}=\boldsymbol{b}$ 且 $x_j=0$，$\forall j\notin\{j_1,\cdots,j_m\}$. 否则，问题 (4.1) 的可行域为空集. 这种做法称为**两阶段方法**.

② 设 $\alpha_j=\lambda_1\alpha_{j_1}+\cdots+\lambda_m\alpha_{j_m}$，$\boldsymbol{y}_j=\boldsymbol{e}_j-(\lambda_1\boldsymbol{e}_{j_1}+\cdots+\lambda_m\boldsymbol{e}_{j_m})$，$\delta_j=\boldsymbol{c}^T\boldsymbol{y}_j$，其中 $j\notin\{j_1,\cdots,j_m\}$.
注意到 $\{\boldsymbol{y}_j\mid j\neq j_1,\cdots,j_m\}$ 是线性方程组 $A\boldsymbol{x}=\boldsymbol{0}$ 的基础解系. 若所有 $\delta_j\geqslant 0$，则 $\boldsymbol{x}$ 是问题 (4.1) 的最优解，输出 $j_1,\cdots,j_m$ 和 $\boldsymbol{x}$.

③ 依次取 $j\notin\{j_1,\cdots,j_m\}$ 满足 $\delta_j<0$. 设 $T=\{t\mid\lambda_t>0\}$. 若 $T=\varnothing$，则 $\boldsymbol{y}_j\geqslant\boldsymbol{0}$，当 $\mu\to+\infty$ 时，$\boldsymbol{z}=\boldsymbol{x}+\mu\boldsymbol{y}_j$ 满足 $\boldsymbol{z}\geqslant\boldsymbol{0}$ 且 $A\boldsymbol{z}=\boldsymbol{b}$ 且 $\boldsymbol{c}^T\boldsymbol{z}=\boldsymbol{c}^T\boldsymbol{x}+\mu\delta_j\to-\infty$. 问题 (4.1) 无最小值，输出 $\varnothing$.

设 $T \neq \varnothing$，$\mu = \min\limits_{t \in T} \dfrac{x_{j_t}}{\lambda_t}$．若 $\mu = 0$，则 `goto ③`，取下一个 j．若对所有 j 都有 $\mu = 0$，则 $\boldsymbol{x}$ 是问题 (4.1) 的最优解，输出 $j_1, \cdots, j_m$ 和 $\boldsymbol{x}$．

若 $\mu \neq 0$，不妨设 $\mu = \dfrac{x_{j_t}}{\lambda_t}$，则 $\boldsymbol{z} = \boldsymbol{x} + \mu \boldsymbol{y}_j$ 满足 $z_j = \mu$ 且 $z_{j_t} = 0$ 且 $\boldsymbol{z} \geqslant \boldsymbol{0}$ 且 $A\boldsymbol{z} = \boldsymbol{b}$ 且 $\boldsymbol{c}^T \boldsymbol{z} < \boldsymbol{c}^T \boldsymbol{x}$．此时，$\{\alpha_{j_1}, \cdots, \alpha_{j_m}, \alpha_j\} \setminus \{\alpha_{j_t}\}$ 线性无关．把 j_t 换成 j，$\boldsymbol{x}$ 换成 $\boldsymbol{z}$，`goto ②`，重新开始．

□

关于单纯形方法的一些说明：

- 设 $\alpha = (a_1, \cdots, a_n) \in \mathbb{R}^n$ 是非零向量．方程 $a_1 x_1 + \cdots + a_n x_n = b$ 的解集称为**平面**，α 称为此平面的**法向量**．不等式 $a_1 x_1 + \cdots + a_n x_n \leqslant b$ 的解集称为**半空间**，α 称为此半空间的**外法向量**．若干个半空间的有界交集称为**单纯形**．若线性规划问题的可行域 S 是有界的，则 S 构成单纯形．定理 4.5 说明若线性规划问题有最优解，则 S 的某个顶点是最优解．
- 容易验证，若 $\boldsymbol{x}, \boldsymbol{y} \in S$，$0 \leqslant t \leqslant 1$，则线段 $\boldsymbol{xy}$ 上的任意点 $\boldsymbol{z} = (1-t)\boldsymbol{x} + t\boldsymbol{y} \in S$．故 S 是凸集．若 $\boldsymbol{x}, \boldsymbol{y}$ 都是线性规划问题的最优解，则 $\boldsymbol{z}$ 也是最优解．故最优解的全体构成一个子单纯形，可能是 S 的顶点、棱、面 $\cdots$．
- 通过对 A 施行一系列初等行变换，可把 $\begin{pmatrix} \alpha_{j_1} & \cdots & \alpha_{j_m} \end{pmatrix}$ 变成 I_m，同时把所有 α_j 表示成 $\lambda_1 \alpha_{j_1} +$

$\cdots + \lambda_m \alpha_{j_m}$ 的形式，其中 $j \notin \{j_1, \cdots, j_m\}$．每次更换基向量 α_{j_t} 时，施行初等行变换以更新 A，并重新计算 δ_*．

- 由于 $\boldsymbol{x}$ 由 $\{j_1, \cdots, j_m\}$ 唯一确定，$\{j_1, \cdots, j_m\}$ 至多有 $\dfrac{n!}{m!(n-m)!}$ 种可能，所以单纯形方法的计算过程在有限多步后终止．
- 第二阶段 ② 和 ③ 的所有计算过程都可以在一个 $(m+1) \times (n+1)$ 的矩阵上进行．此矩阵称为**单纯形表**．单纯形方法的空间复杂度为 $O(mn)$．

定义 4.2. 设 $A \in \mathbb{R}^{m \times n}$ 是行满秩的，$\boldsymbol{b} \in \mathbb{R}^{m \times 1}$，$\boldsymbol{c} \in \mathbb{R}^{n \times 1}$．下述问题称为问题 (4.1) 的**对偶问题**．

$$\text{求 } \boldsymbol{y} \in \mathbb{R}^{m \times 1} \text{ 满足 } A^T \boldsymbol{y} \leqslant \boldsymbol{c}\text{，使得 } \boldsymbol{b}^T \boldsymbol{y} \text{ 最大．} \tag{4.2}$$

对于一般形式的线性规划问题，也可以定义其对偶问题，此处不详细叙述．

定理 4.6. 设线性规划问题 (4.2) 有最优解，$A = (\alpha_1, \cdots, \alpha_n)$ 是行满秩的，则存在极大线性无关组 $\alpha_{j_1}, \cdots, \alpha_{j_m}$，使得由如下线性方程组确定的 $\boldsymbol{y}$ 是问题 (4.2) 的一个最优解.

$$\alpha_j^T \boldsymbol{y} = c_j,\ \forall j \in \{j_1, \cdots, jm\}.$$

证明. 设 $\boldsymbol{y}$ 是问题 (4.2) 的最优解，使得 $\boldsymbol{z} = \boldsymbol{c} - A^T\boldsymbol{y}$ 的零元素最多. 设 $\{j \mid z_j = 0\} = \{j_1, \cdots, j_k\}$. 下面证明 $\operatorname{rank}(\alpha_{j_1}, \cdots, \alpha_{j_k}) = m$，从而存在极大线性无关组 $\alpha_{j_1}, \cdots, \alpha_{j_m}$，满足题设. 否则，

- 若 $\boldsymbol{b}$ 不是 $\alpha_{j_1}, \cdots, \alpha_{j_k}$ 的线性组合，则存在 $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ 使得 $\boldsymbol{b}^T\boldsymbol{x} = 1$ 且 $\alpha_{j_1}^T\boldsymbol{x} = \cdots = \alpha_{j_k}^T\boldsymbol{x} = 0$. 设 $\mu = \min\limits_{j\notin\{j_1,\cdots,j_k\}} \dfrac{z_j}{|\alpha_j^T\boldsymbol{x}|}$，则 $\boldsymbol{w} = \boldsymbol{y} + \mu \pm x$ 满足 $A^T\boldsymbol{w} \leqslant \boldsymbol{c}$ 且 $\boldsymbol{b}^T\boldsymbol{w} = \boldsymbol{b}^T\boldsymbol{y} + \mu$，与 $\boldsymbol{y}$ 是最优解矛盾.
- 设 $\boldsymbol{b}$ 是 $\alpha_{j_1}, \cdots, \alpha_{j_k}$ 的线性组合，α_i 不是 $\alpha_{j_1}, \cdots, \alpha_{j_k}$ 的线性组合，则存在 $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ 使得 $\alpha_i^T\boldsymbol{x} = 1$ 且 $\alpha_{j_1}^T\boldsymbol{x} = \cdots = \alpha_{j_k}^T\boldsymbol{x} = 0$. 设 $\mu = \min\limits_{\alpha_j^T\boldsymbol{x}>0} \dfrac{z_j}{\alpha_j^T\boldsymbol{x}}$，则 $\boldsymbol{w} = \boldsymbol{y} + \mu\boldsymbol{x}$ 满足 $A^T\boldsymbol{w} \leqslant \boldsymbol{c}$ 且 $\boldsymbol{b}^T\boldsymbol{w} = \boldsymbol{b}^T\boldsymbol{y}$ 且 $A^T\boldsymbol{w} - \boldsymbol{c}$ 有至少 $k+1$ 个零元素，与 $A^T\boldsymbol{y} - \boldsymbol{c}$ 的零元素最多矛盾.

□

根据定理 4.6 及其证明，可得求解对偶线性规划问题 (4.2) 的对偶单纯形方法. 此处不详细叙述.

定理 4.7. 问题 (4.1) 有最优解当且仅当问题 (4.2) 有最优解. 其最优解 $\boldsymbol{x},\boldsymbol{y}$ 满足 $\boldsymbol{c}^T\boldsymbol{x}=\boldsymbol{b}^T\boldsymbol{y}$.

证明. 若问题 (4.1), (4.2) 分别有可行解 $\boldsymbol{x},\boldsymbol{y}$，则 $\boldsymbol{c}^T\boldsymbol{x}\geqslant\boldsymbol{y}^TA\boldsymbol{x}=\boldsymbol{y}^T\boldsymbol{b}=\boldsymbol{b}^T\boldsymbol{y}$.

- 设问题 (4.1) 有最优解 $\boldsymbol{x}$. 根据定理 4.5，不妨设 $A=\begin{pmatrix}A_1 & A_2\end{pmatrix}$，$\boldsymbol{x}=\begin{pmatrix}A_1^{-1}\boldsymbol{b}\\ \boldsymbol{0}\end{pmatrix}$. 由 $\boldsymbol{c}^T\boldsymbol{x}$ 最小性，得 $\boldsymbol{c}^T\begin{pmatrix}-A_1^{-1}A_2\\ I_{n-m}\end{pmatrix}\geqslant\boldsymbol{0}$. 设 $\boldsymbol{c}=\begin{pmatrix}\boldsymbol{c}_1\\ \boldsymbol{c}_2\end{pmatrix}$，$\boldsymbol{y}=A_1^{-T}\boldsymbol{c}_1$，得 $A^T\boldsymbol{y}=\begin{pmatrix}\boldsymbol{c}_1\\ A_2^TA_1^{-T}\boldsymbol{c}_1\end{pmatrix}\leqslant\boldsymbol{c}$ 且 $\boldsymbol{b}^T\boldsymbol{y}=\boldsymbol{x}^TA^T\boldsymbol{y}=\boldsymbol{x}^T\boldsymbol{c}$. 故 $\boldsymbol{y}$ 是问题 (4.2) 的最优解.
- 设问题 (4.2) 有最优解 $\boldsymbol{y}$. 根据定理 4.6，不妨设 $A^T=\begin{pmatrix}B_1\\ B_2\end{pmatrix}$，$\boldsymbol{c}=\begin{pmatrix}\boldsymbol{c}_1\\ \boldsymbol{c}_2\end{pmatrix}$，$B_1\boldsymbol{y}=\boldsymbol{c}_1$，$B_2\boldsymbol{y}<\boldsymbol{c}_2$.

  (i) 若 $\boldsymbol{b}$ 不在 B_1 的行向量张成的凸集中，则存在一个平面把两者隔开，即存在 $\boldsymbol{w}$ 使得 $B_1\boldsymbol{w}\geqslant\boldsymbol{0}$ 且 $\boldsymbol{b}^T\boldsymbol{w}<0$. 当正数 $\mu\to0$ 时，$\boldsymbol{z}=\boldsymbol{y}-\mu\boldsymbol{w}$ 满足 $B_1\boldsymbol{z}=\boldsymbol{c}_1-\mu B_1\boldsymbol{w}\leqslant\boldsymbol{c}_1$ 且 $B_2\boldsymbol{z}=B_2\boldsymbol{y}-\mu B_2\boldsymbol{w}<\boldsymbol{c}_2$ 且 $\boldsymbol{b}^T\boldsymbol{z}=\boldsymbol{b}^T\boldsymbol{y}-\mu\boldsymbol{b}^T\boldsymbol{w}>\boldsymbol{b}^T\boldsymbol{y}$，与 $\boldsymbol{y}$ 是最优解矛盾.

  (ii) 若 $\boldsymbol{b}$ 在 B_1 的行向量张成的凸集中，则存在 $\boldsymbol{w}\geqslant\boldsymbol{0}$ 使得 $\boldsymbol{b}=B_1^T\boldsymbol{w}$. 由 $\boldsymbol{x}=\begin{pmatrix}\boldsymbol{w}\\ \boldsymbol{0}\end{pmatrix}$ 满足 $A\boldsymbol{x}=B_1^T\boldsymbol{w}=\boldsymbol{b}$ 且 $\boldsymbol{c}^T\boldsymbol{x}=\boldsymbol{c}_1^T\boldsymbol{w}=\boldsymbol{y}^TB_1^T\boldsymbol{w}=\boldsymbol{y}^T\boldsymbol{b}$，得 $\boldsymbol{x}$ 是问题 (4.1) 的最优解. □

根据定理 4.7 的证明，只要得到问题 (4.1), (4.2) 之一的最优解，就可以求得另一问题的最优解.

第五章　线性方程组 (10 学时)

线性方程组的求解问题是线性代数这一数学分支的起源和重要研究对象．对于小规模的线性方程组，通用的消元解法非常有效．对于大规模的线性方程组，需要有针对性的数值解法．对于某些特殊类型的线性方程组，其系数矩阵的存储方式甚至可能不是通常的数表形式，需要特殊处理．

一般地，设线性方程组为

$$A\boldsymbol{x}=\boldsymbol{b} \quad \text{或} \quad \begin{cases} a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n &= b_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n &= b_2 \\ \vdots \qquad\quad \vdots \qquad\qquad \vdots & \quad \vdots \\ a_{m1}x_1+a_{m2}x_2+\cdots+a_{mn}x_n &= b_m \end{cases} \tag{5.1}$$

其中 $A=(a_{ij})\in\mathbb{R}^{m\times n}$，$\boldsymbol{b}=(b_i)\in\mathbb{R}^{m\times 1}$．$A$ 和 $\begin{pmatrix} A & b \end{pmatrix}$ 称为线性方程组的**系数矩阵**和**增广矩阵**.

§5.1 直接解法

§5.1.1 消元法

消元法常被用于求解多元多项式方程组. 其基本思想是：选定某个变元 x，对方程组作同解变形，使得仅有 1 个方程含 x，其他方程都不含 x，从而达到化简方程组的目的. 对于线性方程组，可通过线性函数的加法和数乘运算来消元，作一系列称为初等变换的同解变形.

定义 5.1. 下列对于线性方程组（或矩阵的行向量）的三类操作称为**初等变换** (或**初等行变换**).

① 交换两个方程的位置. ② 某个方程遍乘非零常数. ③ 某个方程乘上常数加至另一方程.

定理 5.1. *线性方程组 (5.1) 可经一系列初等变换变为如下阶梯形.*

$$\begin{cases} \widetilde{a}_{1,p_1}x_{p_1} + \cdots\cdots\cdots\cdots + \widetilde{a}_{1n}x_n &= \widetilde{b}_1 \\ \qquad \ddots \qquad\quad \cdots\cdots\cdots & \quad\vdots \\ \qquad\qquad \widetilde{a}_{r,p_r}x_{p_r} + \cdots + \widetilde{a}_{rn}x_n &= \widetilde{b}_r \\ \qquad\qquad\qquad\qquad\qquad \mathbf{0} &= \widetilde{\boldsymbol{b}}_{r+1} \end{cases}$$

其中 $r = \mathrm{rank}(A)$，$1 \leqslant p_1 < p_2 < \cdots < p_r \leqslant n$ *并且* $\widetilde{a}_{1,p_1}\cdots\widetilde{a}_{r,p_r} \neq 0$.

根据定理 5.1，可得求解线性方程组的 Gauss 消元法.

例 5.1 (Gauss 消元法). 输入可逆方阵 $A \in \mathbb{R}^{n\times n}$ 和 $\boldsymbol{b} \in \mathbb{R}^{n\times 1}$，输出 $\boldsymbol{x} = A^{-1}\boldsymbol{b}$.

```
for(i=1; i<n; i++)
{
  if(a[i,i]==0)
  {
    for(k=i+1; k<=n&&a[k,i]==0; k++); /* 当A可逆时, 存在a[k,i]!=0 */
    t=b[i]; b[i]=b[k]; b[k]=t; for(j=i; j<=n; j++) { t=a[i,j]; a[i,j]=a[k,j]; a[k,j]=t; }
  }
  for(k=i+1; k<=n; k++)
  {
    t=a[k,i]/a[i,i]; b[k]-=t*b[i]; for(j=i+1; j<=n; j++) a[k,j]-=t*a[i,j];
  }
} /* 此时方程组为上三角形, 下面回代求解 */
for(i=n; i>0; i--) { for(j=i+1; j<=n; j++) b[i]-=a[i,j]*b[j]; b[i]/=a[i,i]; }
return b;
```

Gauss 消元法共计包含了 $\frac{2n^3}{3}+\frac{3n^2}{2}-\frac{7n}{6}$ 次对矩阵和向量元素的四则运算．上述对于 A 和 $\boldsymbol{b}$ 的操作可以合并为对于增广矩阵 $(A\,\boldsymbol{b})$ 的操作．为了便于理解，程序作了换行操作．在实际编程中，无需换行操作，可用一个排列 $(p_1,\cdots,p_n)$ 来记录现在的第 i 行对应初始的第 p_i 行． □

与 Gauss 消元法把线性方程组变为上三角形不同，Gauss-Jordan 消元法把线性方程组变为对角形．Gauss-Jordan 消元法常用于求可逆方阵的逆矩阵．

例 5.2 (**Gauss-Jordan 消元法**). 输入可逆方阵 $A\in\mathbb{R}^{n\times n}$ 和 $\boldsymbol{b}\in\mathbb{R}^{n\times 1}$，输出 $\boldsymbol{x}=A^{-1}\boldsymbol{b}$.

```
for(i=1; i<n; i++)
{
  if(a[i,i]==0)
  {
    for(k=i+1; k<=n&&a[k,i]==0; k++); /* 当A可逆时，存在a[k,i]!=0 */
    t=b[i]; b[i]=b[k]; b[k]=t; for(j=i; j<=n; j++) { t=a[i,j]; a[i,j]=a[k,j]; a[k,j]=t; }
  }
  for(k=i+1; k<=n; k++)
  {
```

```
    t=a[k,i]/a[i,i]; b[k]-=t*b[i]; for(j=i+1; j<=n; j++) a[k,j]-=t*a[i,j];
  }
} /* 下面化方程组为对角形 */
for(i=n; i>0; i--) { b[i]/=a[i,i]; for(j=1; j<i; j++) b[j]-=a[j,i]*b[i]; }
return b;
```

Jordan-Gauss 消元法的运算量与 Gauss 消元法相同，只是运算顺序不同，故计算误差略有不同．假设矩阵元素按照行优先的顺序存储，则回代法读写矩阵元素的效率比 Jordan-Gauss 消元法略高． □

在例 5.1 和例 5.2 中，即使 $|a_{ii}|$ 非常小，只要 $a_{ii} \neq 0$，消元过程都可继续下去．这有可能带来数值计算的巨大误差和不稳定性．为了解决此问题，可以每次选取 $a_{ii}, \cdots, a_{ni}$ 中绝对值最大的元素 a_{ki}，交换 $A, \boldsymbol{b}$ 的第 i, k 两行．这种做法称为**选列主元**，算法如下．

例 5.3 (**列主元法**). 输入可逆方阵 $A \in \mathbb{R}^{n\times n}$ 和 $\boldsymbol{b} \in \mathbb{R}^{n\times 1}$，输出 $\boldsymbol{x} = A^{-1}\boldsymbol{b}$.

```
for(i=1; i<n; i++)
{
  for(k=i,j=i+1; j<=n; j++) if(abs(a[j,i])>abs(a[k,i])) k=j;
  if(k!=i)
```

```
  {
    t=b[i]; b[i]=b[k]; b[k]=t; for(j=i; j<=n; j++) { t=a[i,j]; a[i,j]=a[k,j]; a[k,j]=t; }
  }
  for(k=i+1; k<=n; k++)
  {
    t=a[k,i]/a[i,i]; b[k]-=t*b[i]; for(j=i+1; j<=n; j++) a[k,j]-=t*a[i,j];
  }
}
for(i=n; i>=1; i--) { for(j=i+1; j<=n; j++) b[i]-=a[i,j]*b[j]; b[i]/=a[i,i]; }
return b;
```

列主元法仅比 Gauss 消元法多了 $\frac{n^2}{2}+O(n)$ 次查找，运算量仍是 $\frac{2n^3}{3}+O(n^2)$. □

与选列主元类似，也有**选全主元**的做法. 即选取 $p,q\geqslant i$ 使得 $|a_{pq}|=\max\limits_{s,t\geqslant i}|a_{st}|$，交换 $A,\boldsymbol{b}$ 的第 i,p 两行，并交换 A 的第 i,q 两列以及 x_i,x_q 的位置. 全主元法以 $\frac{n^3}{3}+O(n^2)$ 次查找和复杂的编程实现为代价，换取算法的准确性和稳定性的提高. 通常使用列主元法，不用全主元法.

§5.1.2 矩阵分解法

假设需要求解两个线性方程组 $A\boldsymbol{x}=\boldsymbol{b}_1$ 和 $A\boldsymbol{x}=\boldsymbol{b}_2$. 如果用 Gauss 消元法分别求解，则有许多对于 A 中元素的重复运算. 为了节省运算量，一种做法是对增广矩阵 $(A\ \boldsymbol{b}_1\ \boldsymbol{b}_2)$ 作初等变换，同时求解两个方程组. 如果一个方程组必须在另一方程组之前求解，则需要把初等变换过程记录下来，以备后用.

例 5.4. 在例 5.1 中，假设没有遇到换行操作，则消元过程可以表示为

$$L_{n-1}\cdots L_1 A=U \quad 或 \quad A=LU \tag{5.2}$$

其中 $L=(l_{ij})_{i\geqslant j}=L_1^{-1}\cdots L_{n-1}^{-1}$ 是单位下三角方阵，$U=(u_{ij})_{i\leqslant j}$ 是上三角方阵，

$$L_i=\begin{pmatrix} I_{i-1} & & & \\ & 1 & & \\ & -l_{i+1,i} & 1 & \\ & \vdots & & \ddots \\ & -l_{n,i} & & 1 \end{pmatrix}, \quad \forall i. \tag{5.3}$$

当算法结束时，$a_{ij}=u_{ij}\ (i\leqslant j)$ 或 $a_{ij}=l_{ij}\ (i>j)$. 矩阵分解 (5.2) 称为 **LU 分解** 或 **Dolittle 分解**. 对 A^T 作 LU 分解可得 **Crout 分解** $A=L'U'$，其中 L' 是下三角方阵，U' 是单位上三角方阵. □

定理 5.2. 可逆方阵 A 有 LU 分解当且仅当 A 的顺序主子式都 $\neq 0$. 分解方式是唯一的.

证明.（$\Leftarrow$）设 A 的顺序主子式都 $\neq 0$. 应用数学归纳法. 若 $A_{n-1} = LU$，则

$$A = \begin{pmatrix} A_{n-1} & * \\ * & * \end{pmatrix} = \begin{pmatrix} I_{n-1} & \\ * & 1 \end{pmatrix} \begin{pmatrix} A_{n-1} & * \\ & d \end{pmatrix} = \begin{pmatrix} L & \\ * & 1 \end{pmatrix} \begin{pmatrix} U & * \\ & d \end{pmatrix}, \quad d = \frac{\det(A)}{\det(A_{n-1})} \neq 0.$$

（$\Rightarrow$）设 $A = LU$，则 U 可逆，A 的 k 阶顺序主子矩阵 $A_k = L_k U_k$，$\det(A_k) = u_{11} \cdots u_{kk} \neq 0$.
由数学归纳法易知分解方式是唯一的. □

例 5.5. 在例 5.2 中，消元过程可以表示为

$$L_{n-1} P_{n-1} \cdots L_1 P_1 A = U \tag{5.4}$$

其中 L_i 形如 (5.3) 式，$|l_{ji}| \leqslant 1$ $(\forall j)$，P_i 是交换 I 的 i, k $(k \geqslant i)$ 行所得置换方阵，U 是上三角方阵.
对于任意 $i > j$，交换 L_j 中 l_{kj} 和 l_{ij} 的位置可得单位下三角方阵 $\hat{L}_j$ 满足 $P_i L_j = \hat{L}_j P_i$. 故有

$$\widetilde{L}_{n-1} \cdots \widetilde{L}_1 P_{n-1} \cdots P_1 A = U \quad \text{或} \quad A = P^T L U \tag{5.5}$$

其中 $\widetilde{L}_i$ 形如 (5.3) 式，$L = \widetilde{L}_1^{-1} \cdots \widetilde{L}_{n-1}^{-1}$，$P = P_{n-1} \cdots P_1$. 矩阵分解 (5.5) 称为 **PLU 分解**.
可用如下算法求 PLU 分解. $P = (\boldsymbol{e}_{p_1}, \cdots, \boldsymbol{e}_{p_n})^T$ 由 $\mathtt{p} = (p_1, \cdots, p_n)$ 表示，L 的严格下三角部分是 $P \cdot \mathtt{lu}$ 的严格下三角部分，U 是 $P \cdot \mathtt{lu}$ 的上三角部分.

```
for(i=1; i<n; i++) { p[i]=i; for(j=1; j<=n; j++) lu[i,j]=a[i,j]; }
for(i=1; i<n; i++)
{
  for(k=i,j=i+1; j<=n; j++) if(abs(lu[p[j],i])>abs(lu[p[k],i])) k=j;
  if(k!=i) { t=p[i]; p[i]=p[k]; p[k]=t; }
  for(k=i+1; k<=n; k++)
  {
    t=lu[p[k],i]/lu[p[i],i]; for(j=i+1; j<=n; j++) lu[p[k],j]-=t*lu[p[i],j];
  }
}
return (p,lu);
```

有了 PLU 分解之后，可用如下算法求 $\boldsymbol{x} = A^{-1}\boldsymbol{b}$.

```
for(i=1; i<=n; i++) x[i]=b[p[i]];
for(i=1; i<n; i++) for(j=i+1; j<=n; j++) x[j]-=lu[p[j],i]*x[i];
for(i=n; i>=1; i--) { for(j=i+1; j<=n; j++) x[i]-=lu[p[i],j]*x[j]; x[i]/=lu[p[i],i]; }
return x;
```

在上述算法中，PLU 分解包含了 $\dfrac{4n^3-3n^2-n}{6}$ 次对矩阵元素的四则运算，方程求解包含了 $2n^2-n$ 次对矩阵和向量元素的四则运算，合计运算量与 Gauss 消元法、列主元法相同. □

定理 5.3. *对于任意可逆方阵 A，存在置换方阵 P、单位下三角方阵 $L=(l_{ij})$、上三角方阵 U，使得 $A=P^TLU$ 并且 $|l_{ij}|\leqslant 1$，$\forall i,j$.*

注意：*PLU 分解的方式不唯一.*

下面以列主元法为例，分析线性方程组 $A\boldsymbol{x}=\boldsymbol{b}$ 的求解误差. 求解过程可视为计算

$$\boldsymbol{x}=U^{-1}L_{n-1}P_{n-1}\cdots L_1P_1\boldsymbol{b}.$$

根据矩阵乘积的微分公式，得

$$\|d\boldsymbol{x}\|\leqslant\|U^{-1}\|\cdot\|L_{n-1}\|\cdots\|L_1\|\cdot\|\boldsymbol{b}\|\cdot\Big(\|dU^{-1}\|+\|dL_{n-1}\|+\cdots+\|dL_1\|\Big).$$

取 $\|\cdot\|$ 为 ∞ 范数. 由 $\|L_i\|\leqslant 2$，$\|dL_i\|\leqslant\varepsilon$，$\|dU^{-1}\|\leqslant n\varepsilon$ 知 $d\boldsymbol{x}$ 受 U 的影响最大，受 $\prod\limits_{i=1}^{n-1}\|L_i\|$ 和 n 的影响也不可忽视. 为了更好地减少误差，可以对 A 施行正交变换，化 A 为三角形或对角形.

定理 5.4. *对于任意可逆方阵 A，存在正交方阵 Q 使得 $A=QR$，其中 R 是上三角方阵并且对角元素均大于 0. 上述分解方式是唯一的，称为 A 的* **QR 分解**.

证明.（存在性）有三种计算 QR 分解的常用算法.

1. **Gram-Schmidt 标准正交化** $A=(\alpha_1,\cdots,\alpha_n)$ 为正交方阵 $Q=(\gamma_1,\cdots,\gamma_n)$.

$$\beta_i=\alpha_i-\sum_{j=1}^{i-1}(\gamma_j^T\alpha_i)\gamma_j,\quad \gamma_i=\frac{\beta_i}{\|\beta_i\|},\quad i=1,2,\cdots,n.$$

由 $\alpha_i=\sum\limits_{j=1}^{i-1}(\gamma_j^T\alpha_i)\gamma_j+\|\beta_i\|\gamma_i$，得 $A=QR$.

2. **Givens 变换**消元，化为上三角.

$$\frac{1}{r}\begin{pmatrix}a_{ii} & a_{ji}\\ -a_{ji} & a_{ii}\end{pmatrix}\begin{pmatrix}a_{ii}\\ a_{ji}\end{pmatrix}=\begin{pmatrix}r\\ 0\end{pmatrix},\quad r=\sqrt{a_{ii}^2+a_{ji}^2}.$$

存在 Givens 方阵 $G_1,\cdots,G_m$，$m=\frac{n(n-1)}{2}$ 使得 $\underbrace{G_m\cdots G_1}_{Q^T}A$ 是上三角方阵且对角元素均大于 0.

3. **Householder 变换**消元，化为上三角.

$$\forall\alpha\in\mathbb{R}^{n\times1},\ \text{设}\ r=\sqrt{\alpha^T\alpha},\ \beta=\alpha-r\boldsymbol{e}_1,\ H=2I_n-\frac{2}{\beta^T\beta}\beta\beta^T\qquad\Rightarrow\qquad H\alpha=r\boldsymbol{e}_1.$$

存在 Householder 方阵 $H_1,\cdots,H_{n-1}$ 使得 $\underbrace{H_{n-1}\cdots H_1}_{Q^T}A$ 是上三角方阵且对角元素均大于 0.

（唯一性）若 $A = Q_1R_1 = Q_2R_2$，则 $Q_2^TQ_1 = R_2R_1^{-1} = P$ 既是正交方阵，又是上三角方阵，并且对角元素都大于 0．故 P 是单位方阵． □

使用 QR 分解方法，线性方程组 $A\boldsymbol{x} = \boldsymbol{b}$ 的解 $\boldsymbol{x} = R^{-1}Q^T\boldsymbol{b}$ 的误差满足

$$\|d\boldsymbol{x}\|_2 \leqslant \|R^{-1}\|_2 \cdot \|\boldsymbol{b}\|_2 \cdot \|dR^{-1}\|_2.$$

由 $\|dR^{-1}\|_2 \leqslant \sqrt{n}\varepsilon$ 知 $d\boldsymbol{x}$ 受 R 的影响最大，受 n 的影响可忽略不计．

§5.1.3 特殊方程组

前面考虑的是一般形式的线性方程组，其系数以矩阵 $A = (a_{ij})$ 的形式存储．本小节考虑一些特殊形式的线性方程组，其系数可能以其他形式存储．

例 5.6. 三对角方阵 $A = \begin{pmatrix} u_1 & v_1 & & \\ w_1 & u_2 & \ddots & \\ & \ddots & \ddots & v_{n-1} \\ & & w_{n-1} & u_n \end{pmatrix}$ 可由三个向量 $(u_i), (v_i), (w_i)$ 表示．

当 A 的顺序主子式都 $\neq 0$ 时，Gauss 消元法修改为

```
for(i=1; i<n; i++) { t=w[i]/u[i]; b[i+1]-=t*b[i]; u[i+1]-=t*v[i]; }
```

```
b[n]/=u[n]; for(i=n-1; i>0; i--) b[i]=(b[i]-v[i]*b[i+1])/u[i];
return b;
```

如果在消元过程中有交换两行的操作，则 A 的三对角性质被破坏．由 A 的 PLU 分解所得上三角方阵 $U=(u_{ij})$ 满足 $u_{ij}=0$，$\forall j\geqslant i+3$．此时，A 可由四个向量表示．列主元法修改为

```
b[n+1]=b[n+2]=v[n]=0; for(i=1; i<=n; i++) z[i]=0;
for(i=1; i<n; i++)
{
  if(abs(u[i])<abs(w[i]))
  { t=b[i]; b[i]=b[i+1]; b[i+1]=t; t=u[i]; u[i]=w[i]; w[i]=t;
    t=v[i]; v[i]=u[i+1]; u[i+1]=t; z[i]=v[i+1]; v[i+1]=0; }
  t=w[i]/u[i]; b[i+1]-=t*b[i]; u[i+1]-=t*v[i]; v[i+1]-=t*z[i];
}
for(i=n; i>0; i--) b[i]=(b[i]-v[i]*b[i+1]-z[i]*b[i+2])/u[i];
return b;
```

对于满足 $a_{ij}=0,\ \forall j\notin\{i-q,\cdots,i+p\}$ 的**带状方阵**，Gauss 消元法和列主元法可作类似修改． □

例 5.7. 对称方阵 $A=(a_{ij})\in\mathbb{R}^{n\times n}$ 可由一个 $m=\frac{n(n+1)}{2}$ 维向量 $(a_1,\cdots,a_m)$ 表示．当引用 a_{ij} 时，实际指向某个 a_k，k 由 i,j 确定．

如果在消元过程中有交换两行的操作，则 A 的对称性质被破坏，无法用 m 维向量表示．当 A 的顺序主子式都 $\neq 0$ 时，Gauss 消元法无需交换两行的操作，可顺利进行下去．

特别，当 A 是正定方阵时，A 的任意主子式都 $\neq 0$．根据定理 5.5，A 必有 LDL^T 分解．设下三角方阵 $\hat{L}=L\sqrt{D}$，得 **Cholesky 分解** $A=\hat{L}\hat{L}^T$．

定理 5.5. *设对称实方阵 A 的顺序主子式都 $\neq 0$，则存在单位下三角方阵 L 和对角阵 D，使得 $A=LDL^T$．上述分解方式是唯一的，称为 A 的* **$\mathbf{LDL}^T$ 分解**.

证明. 定理 5.2 的推论．设 $A=LU$，则 $D=L^{-1}AL^{-T}$ 既是对称的又是上三角的，故是对角的． □

例 5.8. 设 Vandermonde 方阵 $A=\begin{pmatrix}1 & x_1 & \cdots & x_1^{n-1}\\ \vdots & \vdots & \cdots & \vdots\\ 1 & x_n & \cdots & x_n^{n-1}\end{pmatrix}$，其中 $x_1,\cdots,x_n$ 两两不同．

线性方程组 $A\boldsymbol{x}=\boldsymbol{b}$ 等价于求 $f(x)=\sum_{i=0}^{n-1}c_ix^i$ 使得 $f(x_i)=b_i$，$\forall i$．多项式的系数 $(c_0,\cdots,c_{n-1})^T$ 即是线性方程组的解．在某些情形下，利用多项式插值可有效地求解线性方程组．例如，$x_1,\cdots,x_n$ 是某个已知的 n 次多项式 $p(x)$ 的全体根，求 f 是为了计算 f 在其他点处的值． □

例 5.9. 循环方阵 $A=\begin{pmatrix} a_1 & a_n & \cdots & a_2 \\ a_2 & a_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_n \\ a_n & \cdots & a_2 & a_1 \end{pmatrix}$ 由第 1 列确定，可以表示成矩阵多项式的形式.

$$A=f(Z),\quad f(x)=\sum_{i=1}^{n}a_ix^{i-1},\quad Z=\begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & & 1 \\ & & 1 & \end{pmatrix}.$$

注意到 Z 的特征多项式 $\varphi_Z(x)=x^n-1$ 也是 Z 的最小多项式，并且

$$A\boldsymbol{c}=\boldsymbol{b}\qquad\Leftrightarrow\qquad f(Z)g(Z)=h(Z)，\text{其中 } g(x)=\sum_{i=1}^{n}c_ix^{i-1},\ h(x)=\sum_{i=1}^{n}b_ix^{i-1}.$$

线性方程组 $A\boldsymbol{x}=\boldsymbol{b}$ 等价于求 $g(x)$ 使得 $f(x)g(x)\equiv h(x) \bmod x^n-1$. 设 $\omega=\mathrm{e}^{\frac{2\pi\sqrt{-1}}{n}}$ 是 n 次单位根. A 是可逆方阵 $\Leftrightarrow f(x)$ 与 x^n-1 互素 $\Leftrightarrow f(\omega^k)\neq 0$，$\forall k$. 得 $g(\omega^k)=\dfrac{h(\omega^k)}{f(\omega^k)}$，$\forall k$. 故 $\boldsymbol{x}=A^{-1}\boldsymbol{b}$ 满足线性方程组 $\left(\omega^{ij}\right)\boldsymbol{x}=\left(g(\omega^i)\right)$，其中 $0\leqslant i,j\leqslant n-1$. 易知 $\left(\omega^{ij}\right)^{-1}=\left(\frac{1}{n}\omega^{-ij}\right)$. 因此，

$$x_k=\frac{1}{n}\sum_{j=0}^{n-1}\frac{h(\omega^j)}{f(\omega^j)}\omega^{(1-k)j},\quad k=1,\cdots,n.$$

用经典方法计算上式需 $O(n^2)$ 次复数的四则运算，用多项式快速算法只需 $O(n\log^2 n)$ 次运算. □

§5.2 迭代解法

当一般方阵 A 的阶数 n 很大时，直接求解线性方程组 $A\boldsymbol{x}=\boldsymbol{b}$ 难以在可接受的时间内完成. 在这种情况下，可用迭代方法求线性方程组的一个近似解. 当特殊方阵 A 乘向量的操作可以快速完成时（例如 A 是稀疏矩阵），用迭代方法求解线性方程组也是一种良好的选择.

§5.2.1 线性迭代法

线性迭代法是一类求解线性方程组的常用迭代方法，其基本思想如下.

定理 5.6. *设 n 阶方阵 A 可逆. 当且仅当 C 满足 $\rho(I-CA)<1$ 时，对于任意 $\boldsymbol{x}_0\in\mathbb{R}^{n\times 1}$，*

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-C(A\boldsymbol{x}_k-\boldsymbol{b}),\quad k\in\mathbb{N} \tag{5.6}$$

都收敛到 $A^{-1}\boldsymbol{b}$.

证明. $\boldsymbol{x}_{k+1}-A^{-1}\boldsymbol{b}=(I-CA)(\boldsymbol{x}_k-A^{-1}\boldsymbol{b})=\cdots=(I-CA)^k(\boldsymbol{x}_0-A^{-1}\boldsymbol{b})$. 根据定理 1.2，$\boldsymbol{x}_k\to A^{-1}\boldsymbol{b}$ 当且仅当 $\rho(I-CA)<1$. □

收敛性与初值无关是线性迭代法的一个显著优点. 由 AC 和 CA 有相同的特征多项式，可得

$$\rho(I-AC)=\rho(I-CA).$$

例 5.10 (**线性迭代法**). 输入可逆方阵 A，向量 $\boldsymbol{b}$，方阵 C 满足 $\rho(I-CA)<1$，输出 $A^{-1}\boldsymbol{b}$.

```
Initalize(x);
for(k=1; k<=MaxCount; k++)
{
  y=Multiplication(A,x)-b; if(Norm[y]<epsilon) break;
  y=Multiplication(C,y); x-=y;
}
return x;
```

线性迭代法以 $\|A\boldsymbol{x}-\boldsymbol{b}\|$ 作为判断迭代是否收敛的标准．矩阵 A, C 经常不是以 $(a_{ij}), (c_{ij})$ 的形式存储，矩阵乘法 $A\boldsymbol{x}, C\boldsymbol{y}$ 需要有独立不同的编程实现．例如，$C=B^{-1}$，计算 $C\boldsymbol{y}$ 等价于求解 $B\boldsymbol{x}=\boldsymbol{y}$.

定义 5.2. 在本小节中，设 A 是可逆方阵，L, D, U 分别是 A 的严格下三角部分、对角部分、严格上三角部分，即 $A=L+D+U$．下面是一些常见的线性迭代公式.

- 取 $C=\omega^{-1}A^T$，$\omega>\|A^TA\|_p$，$p\geqslant 1$.
- 取 $C=D^{-1}$，则 (5.6) 式称为 **Jacobi 迭代**.
- 取 $C=(D+L)^{-1}$，则 (5.6) 式称为 **Gauss-Seidel 迭代**.

- 取 $C = \omega D^{-1}$，$\omega > 0$，则 (5.6) 式称为**松弛 Jacobi 迭代**，ω 称为**松弛因子**.
- 取 $C = (\omega^{-1}D + L)^{-1}$，$\omega > 0$，则 (5.6) 式称为**松弛 Gauss-Seidel 迭代**，ω 称为**松弛因子**.

关于松弛因子 ω 的选取，见定理 5.8.

例 5.11 (**松弛 Jacobi 迭代**). 输入可逆方阵 $A = (a_{ij})$，向量 $\boldsymbol{b}$，松弛因子 w，输出 $A^{-1}\boldsymbol{b}$.

```
Initalize(x);
for(k=1; k<=MaxCount; k++)
{
  s=0; for(i=1; i<=n; i++)
  {
    t=b[i]; for(j=1; j<=n; j++) t-=a[i,j]*x[j]; s+=t*t; y[i]=t*w/a[i,i];
  }
  if(sqrt(s)<epsilon) break;
  for(i=1; i<=n; i++) x[i]+=y[i];
}
return x;
```

例 5.12 (**松弛 Gauss-Seidel 迭代**). 输入可逆方阵 $A = (a_{ij})$，向量 $\boldsymbol{b}$，松弛因子 w，输出 $A^{-1}\boldsymbol{b}$.

```
Initalize(x);
for(k=1; k<=MaxCount; k++)
{
  s=0; for(i=1; i<=n; i++)
  {
    t=b[i]; for(j=1; j<=n; j++) t-=a[i,j]*x[j]; s+=t*t; x[i]+=t*w/a[i,i];
  }
  if(sqrt(s)<epsilon) break;
}
return x;
```

比较 Jacobi 迭代和 Gauss-Seidel 迭代，它们的算法程序几乎相同．差别是在每一轮迭代中，Jacobi 迭代延时更新 x_i，Gauss-Seidel 迭代实时更新 x_i，参与到 $x_{i+1}, \cdots, x_n$ 的计算．算法流程的不同引起矩阵 C 的不同.

定理 5.7. 设 $A=(a_{ij})\in\mathbb{R}^{n\times n}$. 若 $|a_{ii}|>\sum\limits_{j\neq i}|a_{ij}|,\ \forall i$，则 A 称为**严格行对角优**的.

若 $|a_{jj}|>\sum\limits_{i\neq j}|a_{ij}|,\ \forall j$，则 A 称为**严格列对角优**的. 严格行、列对角优统称**严格对角优**.

1. 严格对角优方阵一定是可逆的.

2. 若 A 是严格对角优的，则 *Jacobi* 迭代和 *Gauss-Seidel* 迭代都收敛.

3. 若 A 是正定的对称方阵，则 *Gauss-Seidel* 迭代收敛.

证明.

1. 设 A 是严格行对角优的. 若 A 不可逆，则存在 $\boldsymbol{x}\neq\mathbf{0}$ 使得 $A\boldsymbol{x}=\mathbf{0}$. 设 $|x_i|=\max\limits_{1\leqslant j\leqslant n}|x_j|>0$，则 $|a_{ii}x_i|=|\sum_{j\neq i}a_{ij}x_j|\leqslant\sum_{j\neq i}|a_{ij}|\,|x_j|<|a_{ii}|\,|x_i|$，矛盾. 故 A 是可逆的.

   设 A 是严格列对角优的，则 A^T 是严格行对角优的. 由 A^T 是可逆的，得 A 也是可逆的.

2. 设 λ 是 $I-D^{-1}A$ 的任意特征值，$\boldsymbol{u}$ 是相应的特征向量. 由 $(I-D^{-1}A)\boldsymbol{u}=\lambda\boldsymbol{u}$ 得 $(A-D+\lambda D)\boldsymbol{u}=\mathbf{0}$. 若 $|\lambda|\geqslant 1$，则 $A-D+\lambda D$ 是严格对角优的，与 $A-D+\lambda D$ 的不可逆性矛盾. 故 $|\lambda|<1$.

   设 μ 是 $I-(D+L)^{-1}A$ 的任意特征值，$\boldsymbol{v}$ 是相应的特征向量. 由 $(I-(D+L)^{-1}A)\boldsymbol{v}=\mu\boldsymbol{v}$ 得 $(\mu(D+L)-U)\boldsymbol{v}=\mathbf{0}$. 若 $|\mu|\geqslant 1$，则 $\mu(D+L)-U$ 是严格对角优的. 与 $\mu(D+L)-U$ 的不可逆性矛盾. 故 $|\mu|<1$.

3. 设 A 是正定的对称方阵，则 $U = L^H$ 且 D 也是正定的．设 λ 是 $I - (D+L)^{-1}A$ 的任意特征值，α 是相应的特征向量．由 $(I - (D+L)^{-1}A)\alpha = \lambda\alpha$ 得

$$\lambda = \frac{-\alpha^H U\alpha}{\alpha^H D\alpha + \alpha^H L\alpha} = \frac{-\alpha^H U\alpha}{\alpha^H A\alpha - \alpha^H U\alpha}.$$

设 $\alpha^H L\alpha = x + y\,\mathrm{i}$，$\alpha^H U\alpha = x - y\,\mathrm{i}$．注意到 $\alpha^H A\alpha > 0$ 且 $\alpha^H D\alpha > 0$．若 $x \leqslant 0$，则 $|\alpha^H A\alpha - \alpha^H U\alpha| > |\alpha^H U\alpha|$．若 $x \geqslant 0$，则 $|\alpha^H D\alpha + \alpha^H L\alpha| > |\alpha^H U\alpha|$．总有 $|\lambda| < 1$．

□

定理 5.8. 关于松弛迭代，有如下结论．

1. 设 A 满足 *Jacobi* 迭代收敛．当 $0 < \omega \leqslant 1$ 时，松弛 *Jacobi* 迭代收敛．

2. 设 A 是严格对角优的．当 $0 < \omega \leqslant 1$ 时，松弛 *Gauss-Seidel* 迭代收敛．

3. 设 A 是正定的对称方阵．当 $0 < \omega < \dfrac{2}{\rho(D^{-1}A)}$ 时，松弛 *Jacobi* 迭代收敛．

4. 设 A 是正定的对称方阵．当且仅当 $0 < \omega < 2$ 时，松弛 *Gauss-Seidel* 迭代收敛．

证明. 结论不作要求，仅供参考，证明略． □

§5.2.2 其他迭代法

收敛速度慢是线性迭代法的一个显著缺点. 下面介绍一些提高收敛速度的方法.

例 5.13 (用 **Newton 迭代法求逆矩阵**). 设方阵 $\{M_k\}$ 满足

$$(I - M_{k+1}A) = (I - M_kA)^2.$$

化简得

$$M_{k+1} = 2M_k - M_kAM_k, \quad k \in \mathbb{N}. \tag{5.7}$$

当且仅当 $\rho(I - M_0A) < 1$ 时,

$$I - M_kA = (I - M_0A)^{2^k} \to O, \quad M_k \to A^{-1}.$$

(5.7) 式也称为 **Newton 迭代**. Newton 迭代法可以有效地提高收敛速度, 其代价是矩阵的存储空间以及矩阵乘积的运算量. 当下列条件得到满足时, 可使用 Newton 迭代法计算 A^{-1}.

① 有较好的初值 M_0;

② 可以有效地存储矩阵 A 和 M_k;

③ 可以快速地计算矩阵乘积 M_kAM_k. □

例 5.14 (**化线性方程组问题为优化问题**). 设 A 是正定的对称方阵，则二次函数

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$$

有唯一的最小值点 $\boldsymbol{x} = A^{-1}\boldsymbol{b}$. 可以使用例 4.3 中的各种方法搜索 f 的最小值点. 设 $\boldsymbol{v}$ 是搜索方向，则局部目标函数

$$g(t) = f(\boldsymbol{x} + t\boldsymbol{v}) = \frac{t^2}{2}\boldsymbol{v}^T A\boldsymbol{v} + t\boldsymbol{v}^T(A\boldsymbol{x} - \boldsymbol{b}) + f(\boldsymbol{x}).$$

的最小值点 $t = \dfrac{\boldsymbol{v}^T(\boldsymbol{b} - A\boldsymbol{x})}{\boldsymbol{v}^T A\boldsymbol{v}}$. 例如，

- 设 $\boldsymbol{v} = -\nabla f(\boldsymbol{x}) = \boldsymbol{b} - A\boldsymbol{x}$，得求解线性方程组 $A\boldsymbol{x} = \boldsymbol{b}$ 的**最速下降法**的迭代公式

$$\boldsymbol{v}_k = \boldsymbol{b} - A\boldsymbol{x}_k, \quad t_k = \frac{\boldsymbol{v}_k^T\boldsymbol{v}_k}{\boldsymbol{v}_k^T A\boldsymbol{v}_k}, \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t_k\boldsymbol{v}_k, \quad k \in \mathbb{N}.$$

  上式中的 $\boldsymbol{v}_k$ 满足

$$\boldsymbol{v}_0 = \boldsymbol{b} - A\boldsymbol{x}_0, \quad \boldsymbol{v}_{k+1} = \boldsymbol{v}_k - t_k A\boldsymbol{v}_k, \quad k \in \mathbb{N}.$$

  为了减少重复计算，最速下降法的迭代公式可以重新改写成

$$\boldsymbol{v}_0 = \boldsymbol{b} - A\boldsymbol{x}_0, \quad t_k = \frac{\boldsymbol{v}_k^T\boldsymbol{v}_k}{\boldsymbol{v}_k^T A\boldsymbol{v}_k}, \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t_k\boldsymbol{v}_k, \quad \boldsymbol{v}_{k+1} = \boldsymbol{v}_k - t_k A\boldsymbol{v}_k, \quad k \in \mathbb{N}.$$

  在每一轮迭代中，至多需要计算一次 $A\boldsymbol{v}_k$，至多需要 $n^2 + 6n + O(1)$ 次四则运算.

- 设 $\boldsymbol{v}_0 = -\nabla f(\boldsymbol{x}_0)$，$\boldsymbol{v}_k = -\nabla f(\boldsymbol{x}_k) - \lambda \boldsymbol{v}_{k-1}$，$k \geqslant 1$，其中 λ 使得 $\boldsymbol{v}_k \perp A\boldsymbol{v}_{k-1}$．由此可得求解线性方程组 $A\boldsymbol{x} = \boldsymbol{b}$ 的**共轭梯度法**的迭代公式

$$\boldsymbol{u}_k = \boldsymbol{b} - A\boldsymbol{x}_k, \quad s_k = \frac{\boldsymbol{v}_{k-1}^T A\boldsymbol{u}_k}{\boldsymbol{v}_{k-1}^T A\boldsymbol{v}_{k-1}}, \quad \boldsymbol{v}_k = \boldsymbol{u}_k - s_k\boldsymbol{v}_{k-1}, \quad t_k = \frac{\boldsymbol{v}_k^T\boldsymbol{u}_k}{\boldsymbol{v}_k^T A\boldsymbol{v}_k}, \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t_k\boldsymbol{v}_k, \quad k \in \mathbb{N},$$

$\boldsymbol{u}_k$ 满足 $\boldsymbol{u}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$，$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - t_k A\boldsymbol{v}_k \perp \boldsymbol{v}_k$，$k \in \mathbb{N}$．利用数学归纳法可以证明 [1]

$$\boldsymbol{u}_j^T\boldsymbol{u}_k = \boldsymbol{v}_j^T\boldsymbol{u}_k = \boldsymbol{v}_j^T A\boldsymbol{v}_k = 0, \quad \forall k > j \geqslant 0.$$

为了减少重复计算，共轭梯度法的迭代公式可以重新改写成

$$\begin{aligned} &\boldsymbol{u}_0 = \boldsymbol{v}_0 = \boldsymbol{b} - A\boldsymbol{x}_0, \quad && t = \frac{\boldsymbol{u}_k^T\boldsymbol{u}_k}{\boldsymbol{v}_k^T A\boldsymbol{v}_k}, && \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t\boldsymbol{v}_k, && \boldsymbol{u}_{k+1} = \boldsymbol{u}_k - tA\boldsymbol{v}_k, \\ &&& s = \frac{\boldsymbol{u}_{k+1}^T\boldsymbol{u}_{k+1}}{\boldsymbol{v}_k^T A\boldsymbol{v}_k}, && \boldsymbol{v}_{k+1} = \boldsymbol{u}_{k+1} - s\boldsymbol{v}_k, && k \in \mathbb{N}. \end{aligned}$$

在每一轮迭代中，至多需要计算一次 $A\boldsymbol{v}_k$，至多需要 $n^2 + 9n + O(1)$ 次四则运算．

---

[1] 容易验证 $\boldsymbol{u}_0^T\boldsymbol{u}_1 = \boldsymbol{v}_0^T\boldsymbol{u}_1 = \boldsymbol{v}_0^T A\boldsymbol{v}_1 = 0$．下设 $k \geqslant 1$．由 t_k, s_k 的选取可得 $\boldsymbol{v}_k^T\boldsymbol{u}_{k+1} = \boldsymbol{v}_k^T A\boldsymbol{v}_{k+1} = 0$．由归纳假设，$\boldsymbol{u}_j^T\boldsymbol{u}_{k+1} = (\boldsymbol{v}_j + s_j\boldsymbol{v}_{j-1})^T\boldsymbol{u}_{k+1} = 0$，$\forall j \leqslant k$；$\boldsymbol{v}_j^T\boldsymbol{u}_{k+1} = \boldsymbol{v}_j^T(\boldsymbol{u}_k - t_k A\boldsymbol{v}_k) = 0$，$\forall j < k$；$\boldsymbol{v}_j^T A\boldsymbol{v}_{k+1} = \boldsymbol{v}_j^T A(\boldsymbol{u}_{k+1} - s_k\boldsymbol{v}_k) = \boldsymbol{v}_j^T A\boldsymbol{u}_{k+1} = \frac{1}{t_j}(\boldsymbol{u}_j - \boldsymbol{u}_{j+1})^T\boldsymbol{u}_{k+1} = 0$，$\forall j < k$．

第六章　数值微积分 (8 学时)

§6.1　数值微分

基本思想：给定 $x_0, x_1, \cdots, x_n$ 和正整数 k，求常数 $\lambda_1, \cdots, \lambda_n$，使得对于未知可微函数 $f(x)$ 都有

$$f^{(k)}(x_0) \approx \lambda_1 f(x_1) + \cdots + \lambda_n f(x_n).$$

例 6.1. 下列 g_1, g_2 都是 $f'(x_0)$ 的近似公式.

$$g_1 = \frac{f(x_0+h) - f(x_0)}{h}, \qquad g_2 = \frac{f(x_0+h) - f(x_0-h)}{2h}$$

当 $h > 0$ 时，g_1 称为**向前差商公式**. 当 $h < 0$ 时，g_1 称为**向后差商公式**. g_2 称为**中心差商公式**.

假设 $f(x)$ 有 3 阶连续导函数. 根据 Taylor 展开式，

$$g_1 = f'(x_0) + \frac{f''(x_0)}{2}h + \frac{f'''(\xi)}{6}h^2, \qquad g_2 = f'(x_0) + \frac{f'''(\eta)}{6}h^2$$

其中 $\xi \in [x_0, x_0+h]$，$\eta \in [x_0-h, x_0+h]$．当 $h \to 0$ 时，g_2 比 g_1 的近似效果好． □

例 6.2. 设 $f(x)$ 在 $[x_0-h, x_0+h]$ 上有 $n>k$ 阶连续导函数，$x_0-h<x_1<\cdots<x_n \leqslant x_0+h$，则

$$f(x_i)=\sum_{j=0}^{n-1}\frac{(x_i-x_0)^j}{j!}f^{(j)}(x_0)+\frac{(x_i-x_0)^n}{n!}f^{(n)}(\xi_i),\quad \xi_i\in[x_0,x_i].$$

欲使 $\sum\limits_{i=1}^{n}\lambda_i f(x_i)=f^{(k)}(x_0)+O(h^n)$，只需

$$\begin{pmatrix}\lambda_1 & \cdots & \lambda_n\end{pmatrix}\begin{pmatrix}1 & x_1-x_0 & \cdots & (x_1-x_0)^{n-1}\\ \vdots & \vdots & \cdots & \vdots\\ 1 & x_n-x_0 & \cdots & (x_n-x_0)^{n-1}\end{pmatrix}=\Big(\underbrace{0\ \cdots\ 0}_{k\text{个}}\ k!\ 0\ \cdots\ 0\Big).$$

线性方程组有唯一解．另外，设 $\phi(x)=\sum\limits_{j=0}^{n-1}c_j(x-x_0)^j$ 是 f 关于节点 $x_1,\cdots,x_n$ 的插值多项式．

$$\begin{pmatrix}1 & x_1-x_0 & \cdots & (x_1-x_0)^{n-1}\\ \vdots & \vdots & \cdots & \vdots\\ 1 & x_n-x_0 & \cdots & (x_n-x_0)^{n-1}\end{pmatrix}\begin{pmatrix}c_0\\ c_1\\ \vdots\\ c_{n-1}\end{pmatrix}=\begin{pmatrix}f(x_1)\\ \vdots\\ f(x_n)\end{pmatrix}\quad\Rightarrow\quad \phi^{(k)}(x_0)=k!c_k=\sum_{i=1}^{n}\lambda_i f(x_i).$$

即，可以把 $\phi^{(k)}(x)$ 作为 $f^{(k)}(x)$ 的数值微分公式．特别，$\phi^{(n-1)}(x)\equiv(n-1)!f[x_1,\cdots,x_n]$． □

当 $x_0 - h = x_1 < \cdots < x_n = x_0 + h$ 是等距分点时，下面列出了 $f^{(k)}(x_0)$ 的几个近似公式及其误差.

- $n = 3$，$f'(x_0) = \dfrac{1}{2h}\Big(-f(x_1) + f(x_3)\Big) + O(h^2)$，
  $f''(x_0) = \dfrac{1}{2h^2}\Big(f(x_1) - 2f(x_2) + f(x_3)\Big) + O(h^2)$.
- $n = 4$，$f'(x_0) = \frac{1}{16h}\Big(f(x_1) - 27f(x_2) + 27f(x_3) - f(x_4)\Big) + O(h^4)$，
  $f''(x_0) = \frac{9}{16h^2}\Big(f(x_1) - f(x_2) - f(x_3) + f(x_4)\Big) + O(h^2)$，
  $f^{(3)}(x_0) = \frac{9}{16h^3}\Big(-f(x_1) + 3f(x_2) - 3f(x_3) + f(x_4)\Big) + O(h^2)$.
- $n = 5$，$f'(x_0) = \frac{1}{6h}\Big(f(x_1) - 8f(x_2) + 8f(x_4) - f(x_5)\Big) + O(h^4)$，
  $f''(x_0) = \frac{1}{6h^2}\Big(-f(x_1) + 16f(x_2) - 30f(x_3) + 16f(x_4) - f(x_5)\Big) + O(h^4)$，
  $f^{(3)}(x_0) = \frac{2}{3h^3}\Big(-f(x_1) + 2f(x_2) - 2f(x_4) + f(x_5)\Big) + O(h^2)$，
  $f^{(4)}(x_0) = \frac{2}{3h^4}\Big(f(x_1) - 4f(x_2) + 6f(x_4) - 4f(x_4) + f(x_5)\Big) + O(h^2)$.

适当选取插值节点 $x_1, \cdots, x_n$ 使得 $\phi(x_0) = f(x_0) + O(h^p)$ 并且 p 尽可能大. 从而，$\phi^{(k)}(x_0) = f^{(k)}(x_0) + O(h^{p-k})$ 可以有较高的准确度.

定理 6.1. 设 $f(x)$ 在 $[x_0-h, x_0+h]$ 上有 $n>k$ 阶连续导函数，$x_0-h \leqslant x_1 < \cdots < x_n \leqslant x_0+h$，$\phi(x)$ 是 $f(x)$ 关于节点 $x_1, \cdots, x_n$ 的插值多项式，则有

$$|\phi^{(k)}(x_0) - f^{(k)}(x_0)| \leqslant \frac{h^{n-k}}{(n-k)!} \max_{|x-x_0|\leqslant h} \left|f^{(n)}(x)\right|.$$

证明. 由 $\phi(x)-f(x)$ 有零点 $x_1, \cdots, x_n$，得 $\phi^{(k)}(x)-f^{(k)}(x)$ 在 $[a,b]$ 中有零点 $z_1 < \cdots < z_{n-k}$，即 $\phi^{(k)}(x)$ 是 $f^{(k)}(x)$ 关于节点 $z_1, \cdots, z_{n-k}$ 的插值多项式. 根据定理 2.5，存在 $\xi \in [x_0-h, x_0+h]$ 使得

$$\phi^{(k)}(x_0) = f^{(k)}(x_0) - \frac{f^{(n)}(\xi)}{(n-k)!}(x_0-z_1)\cdots(x_0-z_{n-k}).$$

从而结论成立. □

从理论上说，h 越小，则 $|\phi^{(k)}(x_0) - f^{(k)}(x_0)|$ 越小. 事实上，由于数值计算过程中的舍入误差，h 不宜过小，否则计算 $\phi^{(k)}(x_0)$ 的误差会变大.

下面以中心差商公式为例，说明节点的选取和数值微分的计算. 这种方法称为**外推法**.

取 $h_k = \lambda^k$，$0 < \lambda \leqslant 0.5$，依次计算 $p_k = \dfrac{f(x_0+h_k) - f(x_0-h_k)}{2h_k}$ 并观察 $p_k - p_{k-1}$. 根据例 6.1，

$$p_k \approx f'(x_0) + C\lambda^{2k} \qquad \Rightarrow \qquad p_k - p_{k-1} \approx C(1-\lambda^{-2})\lambda^{2k}.$$

当 $|p_k - p_{k-1}| < (\lambda^{-2}-1)\varepsilon$ 或 $p_{k+1} - p_k \approx \lambda^2(p_k - p_{k-1})$ 不再成立时，结束计算并输出 $\dfrac{p_k - \lambda^2 p_{k-1}}{1-\lambda^2}$ 作为 $f'(x_0)$ 的近似值.

§6.2 数值积分

基本思想：给定 $x_1, \cdots, x_n$，求常数 $\lambda_1, \cdots, \lambda_n$，使得对于未知可积函数 $f(x)$ 都有

$$\int_a^b f(x)dx \approx \lambda_1 f(x_1) + \cdots + \lambda_n f(x_n).$$

定义 6.1. $I(f) = \sum_{i=1}^{n} \lambda_i f(x_i)$ 称为**数值积分公式**，其中 $x_1, \cdots, x_n$ 两两不同.

- 若 $\lambda_1, \cdots, \lambda_n$ 都是非负实数，则 $I(f)$ 称为**稳定**的，否则称为**不稳定**的.
- 若对于 $p \in \mathbb{R}[x]$ 且 $\deg(p) \leqslant m$ 都有 $I(p) = \int_a^b p(x)dx$，则称 $I(f)$ 有 m 阶**代数精度**.
- $I(f) = \int_a^b \phi(x)dx$ 称为**插值型**公式，其中 $\phi(x)$ 是 $f(x)$ 关于节点 $x_1, \cdots, x_n$ 的插值多项式.

§6.2.1 插值型积分

定理 6.2. 数值积分公式 $I(f)$ 有 $n-1$ 阶代数精度当且仅当 $I(f)$ 是**插值型**公式.

证明.（$\Leftarrow$）当 $\det(f) \leqslant n-1$ 时，$\phi = f$，$I(f) = \int_a^b f(x)dx$. 故 $I(f)$ 有 $n-1$ 阶代数精度.

（$\Rightarrow$）由 $\deg(\phi) \leqslant n-1$，得 $I(f) = I(\phi) = \int_a^b \phi(x)dx$. □

下面给出两种方法来计算 $I(f)=\sum\limits_{i=1}^{n}\lambda_i f(x_i)$ 的系数 $\lambda_1,\cdots,\lambda_n$.

方法 1. 由 $\phi(x)=\sum\limits_{i=1}^{n}f(x_i)\prod\limits_{j\neq i}\dfrac{x-x_j}{x_i-x_j}$ 得 $\int_a^b\phi(x)dx=\sum\limits_{i=1}^{n}\lambda_i f(x_i)$，其中 $\lambda_i=\int_a^b\prod\limits_{j\neq i}\dfrac{x-x_j}{x_i-x_j}dx$. □

方法 2. 设 $c=\dfrac{a+b}{2}$. 对于 $p(x)=(x-c)^j$，由

$$\sum_{i=1}^{n}\lambda_i p(x_i)=\int_a^b p(x)dx,\quad \forall j=0,1,\cdots,n-1$$

得

$$\begin{pmatrix}\lambda_1 & \cdots & \lambda_n\end{pmatrix}\begin{pmatrix}1 & (x_1-c) & \cdots & (x_1-c)^{n-1}\\ \vdots & \vdots & \cdots & \vdots\\ 1 & (x_n-c) & \cdots & (x_n-c)^{n-1}\end{pmatrix}=\begin{pmatrix}s_1 & s_2 & \cdots & s_n\end{pmatrix},$$

其中 $s_j=\int_a^b(x-c)^{j-1}dx=\dfrac{1-(-1)^j}{j}\left(\dfrac{b-a}{2}\right)^j$. 上述线性方程组有唯一解 $(\lambda_1,\cdots,\lambda_n)$. □

当 n 是奇数并且 $x_1,\cdots,x_n$ 关于 c 对称时，由 $x_i+x_{n+1-i}=2c$ 可得 $\lambda_i=\lambda_{n+1-i}$ 且 $\sum\limits_{i=1}^{n}\lambda_i(x_i-c)^n=0=s_{n+1}$. 即 $I(p)=\int_a^b p(x)dx$ 对于 $p(x)=x^n$ 也成立. 此时，$I(f)$ 有 n 阶代数精度.

例 6.3. 当 $a = x_1 < \cdots < x_n = b$ 是 $n-1$ 等分点时，$I(f)$ 称为**封闭型 Newton-Cotes 公式**.
当 $a < x_1 < \cdots < x_n < b$ 是 $n+1$ 等分点时，$I(f)$ 称为**开放型 Newton-Cotes 公式**. 特别，

- $n = 1$，$I(f) = (b-a) f(\frac{a+b}{2})$ 有 1 阶代数精度，称为**中点公式**或**矩形公式**.
- $n = 2$，$I(f) = \dfrac{b-a}{2}\Big(f(a) + f(b)\Big)$ 有 1 阶代数精度，称为**梯形公式**.
- $n = 3$，$I(f) = \dfrac{b-a}{6}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right)$ 有 3 阶代数精度，称为 **Cavalieri-Simpson 公式**.
- $n \geqslant 5$ 的开放型 Newton-Cotes 公式和 $n \geqslant 9$ 的封闭型 Newton-Cotes 公式都是不稳定的.

□

下面给出上述几个公式的误差估计.

定理 6.3. *设 $f(x)$ 在 $[a,b]$ 上有 2 或 4 阶连续导函数，$c = \dfrac{a+b}{2}$，则存在 $\xi_i \in [a,b]$ 使得*

$$\int_a^b f(x)dx = (b-a)f(c) + \frac{f''(\xi_1)}{24}(b-a)^3$$

$$\int_a^b f(x)dx = \frac{b-a}{2}\Big(f(a) + f(b)\Big) - \frac{f''(\xi_2)}{12}(b-a)^3$$

$$\int_a^b f(x)dx = \frac{b-a}{6}\left(f(a) + 4f(c) + f(b)\right) - \frac{f^{(4)}(\xi_3)}{2880}(b-a)^5.$$

证明. 根据 Taylor 展开式、积分中值定理、Newton 插值公式，存在 $\xi_i \in [a,b]$ 使得

$$\begin{aligned}
\int_a^b f(x)dx - (b-a)f(c) &= \int_a^b \Big(f(x) - f(c) - f'(c)(x-c)\Big)dx \\
&= \frac{f''(\xi_1)}{2}\int_a^b (x-c)^2 dx = \frac{f''(\xi_1)}{24}(b-a)^3,
\end{aligned}$$

$$\begin{aligned}
\int_a^b f(x)dx - \frac{b-a}{2}\Big(f(a)+f(b)\Big) &= \int_a^b f[a,b,x](x-a)(x-b)dx \\
&= \frac{f''(\xi_2)}{2}\int_a^b (x-a)(x-b)dx = -\frac{f''(\xi_2)}{12}(b-a)^3,
\end{aligned}$$

$$\begin{aligned}
\int_a^b f(x)dx - \frac{b-a}{6}\left(f(a)+4f(c)+f(b)\right) &= \int_a^b f[a,b,c,x](x-a)(x-b)(x-c)dx \\
&= -\int_a^b f[a,b,c,x,x]\left(\int_a^x (t-a)(t-b)(t-c)dt\right)dx \\
&= -\frac{f^{(4)}(\xi_3)}{24}\int_a^b\int_a^x (t-a)(t-b)(t-c)dt\,dx = -\frac{f^{(4)}(\xi_3)}{2880}(b-a)^5.
\end{aligned}$$

□

对于一般的插值型数值积分公式，根据定理 2.5，有如下误差估计.

定理 6.4. 设 $f(x)$ 在 $[a,b]$ 上有 n 阶连续导函数，$x_1 < \cdots < x_n$，$I(f) = \sum_{i=1}^{n} \lambda_i f(x_i)$ 是插值型数值积分公式，则 $\left|I(f) - \int_a^b f(x)dx\right| \leqslant \frac{1}{n!} \int_a^b |(x-x_1)\cdots(x-x_n)|dx \max_{a\leqslant x\leqslant b} \left|f^{(n)}(x)\right|$.

定理 6.5. 数值积分公式 $I(f) = \sum_{i=1}^{n} \lambda_i f(x_i)$ 至多有 $2n-1$ 阶代数精度．$I(f)$ 有 $2n-1$ 阶代数精度当且仅当 $x_1, \cdots, x_n$ 是例 *2.10* 中的多项式 $L_n(x)$ 的根．此时，$I(f)$ 称为 **Gauss 积分**公式．

证明. $p(x) = (x-x_1)^2 \cdots (x-x_n)^2$ 满足 $\int_a^b p(x)dx > 0 = I(p)$．故 $I(f)$ 无 $2n$ 阶代数精度．

$$
\begin{aligned}
I(f) \text{ 有 } 2n-1 \text{ 阶代数精度} \quad &\Leftrightarrow \quad I(p) = \int_a^b p(x)dx,\ \forall p \in \mathbb{R}[x] \text{ 满足 } \deg(p) \leqslant 2n-1 \\
&\Leftrightarrow \quad \int_a^b p[x_1, \cdots, x_n, x](x-x_1)\cdots(x-x_n)dx = 0 \\
&\Leftrightarrow \quad \int_a^b x^j (x-x_1)\cdots(x-x_n)dx = 0, \quad \forall j = 0, 1, \cdots, n-1 \\
&\Leftrightarrow \quad (x-x_1)\cdots(x-x_n) = \frac{n!}{(2n)!} L_n.
\end{aligned}
$$

□

Gauss 积分公式是代数精度最高的插值型公式．设 $I_1(f), I_2(f)$ 都是计算 $\int_a^b f(x)dx$ 的数值积分公式．$I_1(f)$ 的代数精度比 $I_2(f)$ 高，并不表示 $I_1(f)$ 的误差一定比 $I_2(f)$ 小．

§6.2.2 复合型积分

随着节点数 n 的增加，插值型数值积分公式 $I_n(f)=\sum\limits_{i=1}^{n}\lambda_i f(x_i)$ 有可能变得不稳定，$\lambda_1,\cdots,\lambda_n$ 的计算也会带来许多误差，并且无法保证 $I_n(f)$ 收敛到 $\int_a^b f(x)dx$. 为此，可以把 $[a,b]$ 分割成若干小段，在每个小段上使用插值型积分，以保证数值积分收敛到 $\int_a^b f(x)dx$. 由而可得复合型数值积分公式.

例 6.4. 记 $S=\int_a^b f(x)dx$，$h=\dfrac{b-a}{n}$，$x_t=a+th$. 根据定理 6.3，有如下结论.

- **复化中点公式** $I_n(f)=\sum\limits_{i=1}^{n}hf(x_{i-1/2})$，$|I_n(f)-S|\leqslant\dfrac{(b-a)^3}{24n^2}\max\limits_{a\leqslant x\leqslant b}\left|f''(x)\right|$.
- **复化梯形公式** $I_n(f)=\sum\limits_{i=1}^{n}h\dfrac{f(x_{i-1})+f(x_i)}{2}=h\left(\dfrac{f(a)}{2}+\dfrac{f(b)}{2}+\sum\limits_{i=1}^{n-1}f(x_i)\right)$,
$$|I_n(f)-S|\leqslant\frac{(b-a)^3}{12n^2}\max_{a\leqslant x\leqslant b}\left|f''(x)\right|.$$
- **复化 Simpson 公式**
$$I_n(f)=\sum_{i=1}^{n}h\frac{f(x_{i-1})+4f(x_{i-1/2})+f(x_i)}{6}=h\left(\frac{f(a)}{2}+\frac{f(b)}{2}+\frac{1}{3}\sum_{i=1}^{n-1}f(x_i)+\frac{2}{3}\sum_{i=1}^{n}f(x_{i-1/2})\right),$$
$$|I_n(f)-S|\leqslant\frac{(b-a)^5}{2880n^4}\max_{a\leqslant x\leqslant b}\left|f^{(4)}(x)\right|.$$

□

定义 6.2. 对于给定的 f，设 $I_h(f)$ 是一个计算 $S=\int_a^b f(x)dx$ 的复合型数值积分公式，其中 $[a,b]$ 被等分成长度 h 的小段．若存在与 h 无关的常数 C 和正整数 r 使得

$$|I_h(f)-S|\leqslant Ch^r$$

则称积分公式 $I_h(f)$ **有 r 阶截断误差.**

假设存在与 h 无关的非零常数 C_1,C_2 和 $p>0$，使得

$$I_h(f)=S+C_1h^r+C_2h^{r+p}+o(h^{r+p}).$$

取正整数 $m\geqslant 2$，令 $\lambda=\dfrac{1}{m}$，得

$$\frac{I_{\lambda h}(f)-\lambda^r I_h(f)}{1-\lambda^r}=S+\frac{\lambda^r(1-\lambda^p)}{1-\lambda^r}C_2h^{r+p}+o(h^{r+p})$$

因此，数值积分公式

$$\widetilde{I}_{\lambda h}(f)=\frac{I_{\lambda h}(f)-\lambda^r I_h(f)}{1-\lambda^r}$$

有 $r+p$ 阶截断误差．$\widetilde{I}_{\lambda h}(f)$ 也是一种复合型的数值积分公式．这种构造具有更高阶截断误差的新公式的方法称为 **Richardson 外推法.**

例 6.5. 外推法的思想也常被用于后验估计复合型数值积分的误差. 取 $h=\dfrac{b-a}{2^n}$,

$$\begin{cases} S \approx I_{2h}(f) - C(2h)^r \\ S \approx I_h(f) - Ch^r \end{cases} \quad \Rightarrow \quad \begin{cases} I_h(f) - I_{2h}(f) \approx 2^{-r}\big(I_{2h}(f) - I_{4h}(f)\big) \\ S - I_h(f) \approx \dfrac{I_h(f) - I_{2h}(f)}{2^r - 1} \end{cases}.$$

当 $|I_h(f) - I_{2h}(f)| < (2^r - 1)\varepsilon$ 时，可认为计算结果已满足精度要求，输出 $I_h(f) + \dfrac{I_h(f) - I_{2h}(f)}{2^r - 1}$.
当 $I_{h/2}(f) - I_h(f) \approx 2^{-r}\big(I_h(f) - I_{2h}(f)\big)$ 不再成立时，可认为是舍入误差影响了计算结果，应结束计算并输出 $I_h + \dfrac{I_h(f) - I_{2h}(f)}{2^r - 1}$.
在上述计算过程中，$I_h(f)$ 与 $I_{2h}(f)$ 有公共积分节点，无需重复计算，只需计算增量即可. 例如，

- 复化梯形公式 $T_h(f) = \dfrac{1}{2}T_{2h}(f) + \dfrac{1}{2}\widetilde{T}_{2h}(f)$，增量 $\widetilde{T}_{2h}(f) = 2h\sum\limits_{i=1}^{2^{n-1}} f(a + (2i-1)h)$.
- 复化 Simpson 公式 $S_h(f) = \dfrac{1}{3}T_h(f) + \dfrac{2}{3}\widetilde{T}_h(f)$，增量 $\widetilde{T}_h(f) = h\sum\limits_{i=1}^{2^n} f(a + (i - \frac{1}{2})h)$.

□

例 6.6. 外推法的思想还被用于设计 Romberg 积分公式.

- 设 $I_n^{(1)}$ 是 n 等分的复化梯形公式，具有 2 阶截断误差.

- $I_n^{(1)}$ 外推为复化 Simpson 公式 $I_{2n}^{(2)} = \dfrac{1}{3}\left(4I_{2n}^{(1)} - I_n^{(1)}\right)$，具有 4 阶截断误差.
- $I_n^{(2)}$ 外推为复化 Newton-Cotes 公式 $I_{2n}^{(3)} = \dfrac{1}{15}\left(16I_{2n}^{(2)} - I_n^{(2)}\right)$，具有 6 阶截断误差.
- 不断进行下去，$I_n^{(i)}$ 外推为 $I_{2n}^{(i+1)} = \dfrac{1}{4^i - 1}\left(4^i I_{2n}^{(i)} - I_n^{(i)}\right)$，具有 $2i+2$ 阶截断误差.

从而可用一个上三角方阵 $R = (r_{ij})_{1\leqslant i\leqslant j\leqslant n}$ 计算 $S = \int_a^b f(x)dx$，其中 $r_{ij} = I_{2^j}^{(i)}(f)$，满足

$$r_{i,j} = \frac{4^{i-1}r_{i-1,j} - r_{i-1,j-1}}{4^{i-1} - 1}, \quad 2 \leqslant i \leqslant j \leqslant n.$$

当 $i, j \to \infty$ 时，$r_{ij} \to S$. 上式称为 **Romberg 积分公式.** □

无需计算 R 的所有元素，按列计算至收敛即可. R 的第一行也只需增量计算.

§6.2.3 重积分

区间 $[a,b]$ 上的数值积分公式可自然地推广到矩形区域上二重积分

$$\iint\limits_{[a,b]\times[c,d]} f(x,y)dxdy \approx \sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_{ij} f(x_i, y_j),$$

以及更高维的重积分. 例如，

- 复化梯形公式

$$I_{m,n}(f)=\frac{h_1h_2}{4}\sum_{i=1}^{m}\sum_{j=1}^{n}\Big(f(x_{i-1},y_{j-1})+f(x_{i-1},y_j)+f(x_i,y_{i-1})+f(x_i,y_j)\Big).$$

- 复化 Simpson 公式

$$I_{m,n}(f)=\frac{h_1h_2}{36}\sum_{i=1}^{m}\sum_{j=1}^{n}\begin{pmatrix}f(x_{i-1},y_{j-1})+4f(x_{i-1},y_{j-1/2})+f(x_{i-1},y_j)+\\ 4f(x_{i-1/2},y_{j-1})+16f(x_{i-1/2},y_{j-1/2})+4f(x_{i-1/2},y_j)\\ +f(x_i,y_{j-1})+4f(x_i,y_{j-1/2})+f(x_i,y_j)\end{pmatrix}.$$

其中 $h_1=\dfrac{b-a}{m}$，$h_2=\dfrac{d-c}{n}$，$x_i=a+ih_1$，$y_j=c+jh_2$.

例 6.7. 求曲面 $x^2+y^2\leqslant 1,\ x^2+z^2\leqslant 2,\ y^2+z^2\leqslant 3$ 围成的空间几何体的体积.

解答. 在 $[-1,1]\times[-1,1]\times[-\sqrt{2},\sqrt{2}]$ 上数值积分如下函数

$$f(x,y,z)=\mathrm{Boole}(x^2+y^2\leqslant 1,\ x^2+z^2\leqslant 2,\ y^2+z^2\leqslant 3).$$

□

第七章　常微分方程数值解 (8 学时)

微分方程是和微积分一起发展起来的重要数学分支，在科学研究和生产实践等方面都有着广泛应用. 只有极少数的微分方程可以求得解析解. 微分方程的解的存在性、唯一性等问题都需要仔细研究.

定义 7.1. 关于函数及其导函数的方程称为**微分方程**. 若方程含有偏导数，则称为**偏微分方程**，否则称为**常微分方程**. 本章主要研究如下形式的一阶常微分方程**初值问题**（也称 **Cauchy 问题**）：

$$\text{求可微函数 } y(x) \text{ 满足} \begin{cases} y'(x) = f(x, y(x)), \quad x \in [a, b] \\ y(a) = c \end{cases} \tag{7.1}$$

其中 $f(x, y)$ 是给定的二元可微函数，a, b, c 是给定的实数. 为了简化问题，假设问题 (7.1) 有唯一解，并且存在 Lipschitz 常数 L 使得 $|f(x, y) - f(x, z)| \leqslant L|y - z|$，$\forall y, z$.

取定 $a = x_0 < \cdots < x_n = b$. 设 y_k 是 $y(x_k)$ 的近似值，则 $(y_0, \cdots, y_n)$ 称为问题 (6.1) 的一个**数值解**. 若每个 y_{k+1} 都由 y_k 唯一确定，则构造此数值解的方法称为**单步法**，否则称为**多步法**.

§7.1 Euler 法

例 7.1. 根据数值微分和数值积分公式，可得问题 (7.1) 的数值解的下列公式．记 $h_k = x_{k+1} - x_k$.

① 由向前差商公式 $y'(x) \approx \dfrac{y(x+h)-y(x)}{h}$，得 $y_{k+1} = y_k + h_k f(x_k, y_k)$.

② 由向后差商公式 $y'(x) \approx \dfrac{y(x)-y(x-h)}{h}$，得 $y_{k+1} = y_k + h_k f(x_{k+1}, y_{k+1})$.

③ 由中心差商公式 $y'(x) \approx \dfrac{y(x+h)-y(x-h)}{2h}$，得 $y_{k+1} = y_{k-1} + (h_k + h_{k-1}) f(x_k, y_k)$.

④ 由梯形公式 $\int_x^{x+h} f(x)dx \approx \dfrac{f(x)+f(x+h)}{2}h$，得 $y_{k+1} = y_k + h_k \dfrac{f(x_k, y_k) + f(x_{k+1}, y_{k+1})}{2}$.

⑤ 结合①,④，得 **Heun 公式** $y_{k+1} = y_k + h_k \dfrac{f(x_k, y_k) + f(x_{k+1}, \widetilde{y}_{k+1})}{2}$，$\widetilde{y}_{k+1} = y_k + h_k f(x_k, y_k)$.

公式 ①,②,④,⑤ 都是单步法，③ 是多步法，①,③,⑤ 称为**显式公式**，②,④ 称为**隐式公式**.

通常用不动点法或迭代法求解隐式公式．例如，设 **Picard 迭代**

$$z_0 = y_k, \quad z_{i+1} = y_k + h_k \frac{f(x_k, y_k) + f(x_{k+1}, z_i)}{2}, \quad i \in \mathbb{N}.$$

若迭代收敛，则 z_1 是 ① 中的 y_{k+1}，z_2 是 ⑤ 中的 y_{k+1}，z_∞ 是 ④ 中的 y_{k+1}，任意 z_i 都可以看作是一种计算 y_{k+1} 的显式公式，称为**预估-校正公式**（把预估 ① 校正成 ④）. □

定义 7.2. 设 $(y_0,\cdots,y_n)$ 是问题 (7.1) 的一个数值解，$0\leqslant k\leqslant n-1$. 若 $y_i=y(x_i),\ \forall i\leqslant k$ 并且 y_{k+1} 由 $y_0,\cdots,y_k$ 唯一确定，则 $y_{k+1}-y(x_{k+1})$ 称为**局部截断误差**. 若还存在常数 C 和正整数 r 使得

$$|y_{k+1}-y(x_{k+1})|\leqslant C(x_{k+1}-x_k)^{r+1}$$

则称 y_{k+1} **有 r 阶局部截断误差**.

例 7.2. 考虑例 7.1 中数值解的局部截断误差. 设 $y_i=y(x_i)$，$\forall i\leqslant k$. 记 $h=x_{k+1}-x_k$. 注意到

$$y(x_{k+1})-y(x_k)=hf(\xi,y(\xi)),\quad \xi\in[x_k,x_{k+1}].$$

① $y_{k+1}-y(x_{k+1})=h\Big(f(x_k,y_k)-f(\xi,y(\xi))\Big)=h^2g'(\eta)=O(h^2)$，其中 $g(x)=f(x,y(x))$.

② $y_{k+1}-y(x_{k+1})=h\Big(f(x_{k+1},y_{k+1})-f(\xi,y(\xi))\Big)=h^2g'(\eta)=O(h^2)$，其中 $g(x)=f(x,y(x))$.

③ 假设 $x_k-x_{k-1}=h=x_{k+1}-x_k$. 根据定理 6.3 的中点公式，

$$y(x_{k+1})-y(x_{k-1})=\int_{x_{k-1}}^{x_{k+1}}f(x,y(x))dx=2hf(x_k,y(x_k))+O(h^3).$$

由此可得 $y_{k+1}-y(x_{k+1})=O(h^3)$.

④ 根据定理 6.3 的梯形公式，

$$y(x_{k+1})-y(x_k)=\int_{x_k}^{x_{k+1}} f(x,y(x))dx=h\frac{f(x_k,y(x_k))+f(x_{k+1},y(x_{k+1}))}{2}+O(h^3)$$

$$\Rightarrow \qquad y_{k+1}-y(x_{k+1})=\frac{f(x_{k+1},y_{k+1})-f(x_{k+1},y(x_{k+1}))}{2}h+O(h^3)$$

$$=\frac{h}{2}\frac{\partial f}{\partial y}(x_{k+1},\zeta)\Big(y_{k+1}-y(x_{k+1})\Big)+O(h^3),$$

其中 ζ 介于 y_{k+1} 和 $y(x_{k+1})$ 之间．由此可得 $y_{k+1}-y(x_{k+1})=O(h^3)$．

⑤ 由 ①，$\widetilde{y}_{k+1}-y(x_{k+1})=O(h^2)$．同 ④，得

$$y_{k+1}-y(x_{k+1})=h\frac{f(x_{k+1},\widetilde{y}_{k+1})-f(x_{k+1},y(x_{k+1}))}{2}+O(h^3)$$

$$=\frac{h}{2}\frac{\partial f}{\partial y}(x_{k+1},\zeta)\Big(\widetilde{y}_{k+1}-y(x_{k+1})\Big)+O(h^3)=O(h^3).$$

综上，公式 ①,② 有 1 阶局部截断误差，③,④,⑤ 有 2 阶局部截断误差． □

定理 7.1. 设 $a=x_0<\cdots<x_n=b$ 是等距分点，$h=\dfrac{b-a}{n}$，$(y_0,\cdots,y_n)$ 是问题 (7.1) 的任意数值解．若 $y_0=y(x_0)$，每个 y_{k+1} 由 y_k 唯一确定，并且 y_{k+1} 都有 r 阶局部截断误差，则存在常数 C 使得

$$|y_n-y(x_n)|\leqslant Ch^r.$$

由此可得问题 (7.1) 的数值解的收敛性．

证明. 设 $z(x)$ 满足 $\begin{cases} z'(x) = f(x, z(x)) \\ z(x_k) = y_k \end{cases}$. 由局部截断误差的定义，可设

$$|y_{k+1} - z(x_{k+1})| \leqslant M(x_{k+1} - x_k)^{r+1}, \tag{*}$$

其中 M 是常数. 再由

$$|y'(x) - z'(x)| = |f(x, y) - f(x, z)| \leqslant L|y(x) - z(x)|,$$

解得

$$|y(x_{k+1}) - z(x_{k+1})| \leqslant \mathrm{e}^{L(x_{k+1} - x_k)} |y(x_k) - z(x_k)|. \tag{**}$$

结合 (*) 和 (**)，得

$$\begin{aligned}
& |y_{k+1} - y(x_{k+1})| \leqslant \mathrm{e}^{Lh} |y_k - y(x_k)| + Mh^{r+1} \\
\Rightarrow \quad & \mathrm{e}^{-(k+1)Lh} |y_{k+1} - y(x_{k+1})| \leqslant \mathrm{e}^{-kLh} |y_k - y(x_k)| + \mathrm{e}^{-(k+1)Lh} Mh^{r+1} \\
\Rightarrow \quad & \mathrm{e}^{-nLh} |y_n - y(x_n)| \leqslant \sum_{k=0}^{n-1} \mathrm{e}^{-(k+1)Lh} Mh^{r+1} \leqslant \frac{Mh^{r+1}}{\mathrm{e}^{Lh} - 1} \leqslant \frac{Mh^r}{L} \\
\Rightarrow \quad & |y_n - y(x_n)| \leqslant \frac{\mathrm{e}^{L(b-a)} M}{L} h^r = Ch^r.
\end{aligned}$$

□

定义 7.3. 下面考虑问题 (7.1) 及其数值解法的稳定性．设 ε 是充分小的正数．

- 若存在常数 C，使得对于任意实数 δ 和可微函数 $g(x)$ 满足 $|\delta| \leqslant \varepsilon$ 且 $\max\limits_{a\leqslant x\leqslant b} |g(x)| \leqslant \varepsilon$，方程
$$\begin{cases} z'(x) = f(x, z(x)) + g(x), \quad x \in [a, b] \\ z(a) = c + \delta \end{cases}$$
的解都存在，并且满足 $\max\limits_{a\leqslant x\leqslant b} |z(x) - y(x)| \leqslant C\varepsilon$，则问题 (7.1) 称为 **Liapunov 稳定**的．

- 若存在常数 C 和正数 h，使得对于任意 $a = x_0 < \cdots < x_n = b$ 满足 $\max\limits_{1\leqslant k\leqslant n} (x_k - x_{k-1}) \leqslant h$，对数值解法中的所有 y_k 作不超过 ε 的任意扰动，所得数值解 $(z_0, \cdots, z_n)$ 都满足 $\max\limits_{1\leqslant k\leqslant n} |z_k - y_k| \leqslant C\varepsilon$，则数值解法称为 **0 稳定**的．0 稳定性等价于 $h \to 0$ 时数值解的收敛性．

  条件 $|f(x, y) - f(x, z)| \leqslant L|y - z|$ 保证了问题 (7.1) 的 Liapunov 稳定性及其单步解法的 0 稳定性．

- 给定正数 h，设 $y(x)$ 的定义域为 $[a, \infty)$，$x_k = a + kh$．若数值解 $\{y_k \mid k \in \mathbb{N}\}$ 是有界的，则数值解法称为**绝对稳定**的．数值解法的绝对稳定性显然依赖于 $f(x, y)$ 和 h．

  考虑典型方程 $\begin{cases} y'(x) = \lambda y(x), \quad x \geqslant 0 \\ y(0) = 1 \end{cases}$．当 $\mathrm{Re}(z) < 0$ 时，解析解 $y(x) = \mathrm{e}^{\lambda x}$ 满足 $\lim\limits_{x\to+\infty} y(x) = 0$．若 $\lim\limits_{n\to\infty} y_n = 0$，则数值解法称为**绝对稳定**的．使得数值解法是绝对稳定的复数 λh 的全体称为此数值解法的**绝对稳定区域**．算法的绝对稳定性与算法的准确性无关．

例 7.3. 考虑例 7.1 中数值解法公式的绝对稳定性．设 $\mathrm{Re}(\lambda) < 0$．记绝对稳定区域为 A.

① 由 $y_{k+1} = (1+\lambda h)y_k$，得 $y_n = (1+\lambda h)^n$.

$A = \{z \in \mathbb{C} : |1+z| < 1\}$ 是复平面上以 -1 为圆心半径 1 的圆的内部.

② 由 $y_{k+1} = y_k + \lambda h y_{k+1}$，得 $y_{k+1} = \dfrac{y_k}{1-\lambda h}$，故 $y_n = \dfrac{1}{(1-\lambda h)^n}$.

$A = \{z \in \mathbb{C} : |1-z| > 1\}$ 是复平面上以 1 为圆心半径 1 的圆的外部.

对于任意 $h > 0$，$\lim\limits_{n\to\infty} y_n = 0$.

③ 由 $y_{k+1} = y_{k-1} + 2\lambda h y_k$，得

$$\begin{pmatrix} y_{k+1} \\ y_k \end{pmatrix} = \begin{pmatrix} 2\lambda h & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} y_k \\ y_{k-1} \end{pmatrix} \qquad \Rightarrow \qquad \begin{pmatrix} y_n \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 2\lambda h & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} y_1 \\ 1 \end{pmatrix}.$$

$M = \begin{pmatrix} 2\lambda h & 1 \\ 1 & 0 \end{pmatrix}$ 的行列式 $\det(M) = -1$，特征值 $\lambda h \pm \sqrt{1+(\lambda h)^2}$，谱半径 $\rho(M) > 1$.

不存在 λ, h 使得 $\lim\limits_{n\to\infty} y_n = 0$ 恒成立．$A = \varnothing$.

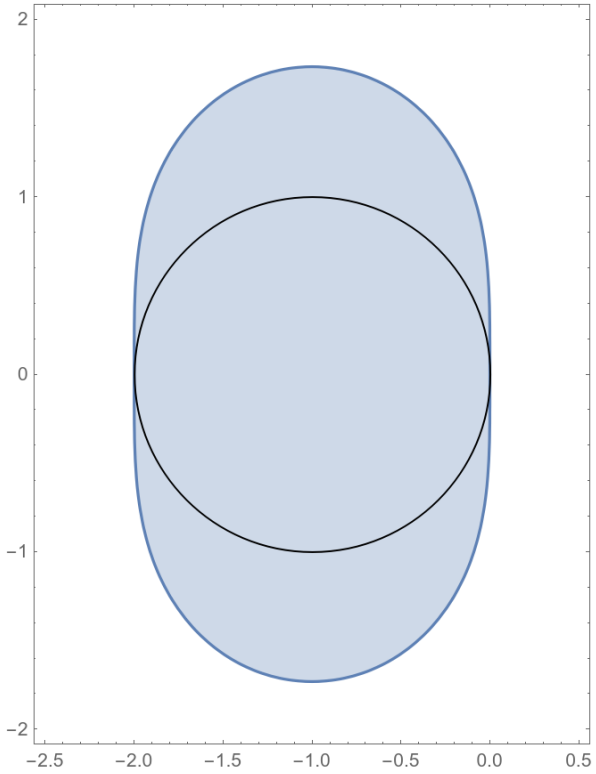
④ 由 $y_{k+1} = y_k + \dfrac{\lambda h}{2}(y_k + y_{k+1})$，得 $y_{k+1} = \dfrac{2+\lambda h}{2-\lambda h} y_k$，故 $y_n = \left(\dfrac{2+\lambda h}{2-\lambda h}\right)^n$.

$A = \{z \in \mathbb{C} : |2+z| < |2-z|\} = \{z \in \mathbb{C} : \mathrm{Re}(z) < 0\}$ 是复平面的 y 轴左侧区域.

对于任意 $h > 0$，$\lim\limits_{n\to\infty} y_n = 0$.

⑤ 由 $y_{k+1} = y_k + \dfrac{\lambda h}{2}\Big(y_k + (1+\lambda h)y_k\Big) = \left(1 + \lambda h + \dfrac{1}{2}(\lambda h)^2\right) y_k$，得 $y_n = \left(1 + \lambda h + \dfrac{1}{2}(\lambda h)^2\right)^n$.

$A = \{z \in \mathbb{C} : |1 + z + \frac{1}{2}z^2| < 1\} = \{z \in \mathbb{C} : |(z+1)^2 + 1| < 2\}$，如下图所示. □



§7.2 Runge-Kutta 法

定义 7.4. 若问题 (7.1) 的数值解法公式具有如下形式，

$$y_{k+1} = y_k + h\sum_{i=1}^{s} c_i K_i, \quad K_i = f\Big(x_k + a_i h,\ y_k + h\sum_{j=1}^{s} b_{ij} K_j\Big) \tag{7.2}$$

其中 $h = x_{k+1} - x_k$，a_i, b_{ij}, c_i 是常数，则 (7.2) 式称为 s 级 **Runge-Kutta 公式.**

记 $B = (b_{ij})$. 若 B 是严格下三角的，即 $b_{ij} = 0, \forall j \geqslant i$，则 (7.2) 式称为**显式**的，否则称为**隐式**的.

下面推导 Runge-Kutta 公式. (7.2) 式有 r 阶局部截断误差当且仅当对于

$$y(x) = x^m, \quad f(x,y) = \lambda(y - x^m) + mx^{m-1}, \quad \forall \lambda \in \mathbb{R},\ 1 \leqslant m \leqslant r$$

都有 $y_{k+1} = y(x_{k+1}) + O(h^{r+1})$ 恒成立. 由

$$K_i = \lambda x_k^m + \lambda h\sum_{j=1}^{s} b_{ij} K_j - \lambda(x_k + a_i h)^m + m(x_k + a_i h)^{m-1}$$

得

$$\begin{pmatrix} K_1 \\ \vdots \\ K_s \end{pmatrix} = (I - \lambda h B)^{-1} \begin{pmatrix} \lambda x_k^m - \lambda(x_k + a_1 h)^m + m(x_k + a_1 h)^{m-1} \\ \vdots \\ \lambda x_k^m - \lambda(x_k + a_s h)^m + m(x_k + a_s h)^{m-1} \end{pmatrix}.$$

把 K_i 带入 $x_k^m + h\sum_{i=1}^{s} c_i K_i = (x_k+h)^m + O(h^{r+1})$，得

$$\begin{pmatrix} c_1 & \cdots & c_s \end{pmatrix} (I-\lambda hB)^{-1} \begin{pmatrix} \lambda x_k^m - \lambda(x_k+a_1h)^m + m(x_k+a_1h)^{m-1} \\ \vdots \\ \lambda x_k^m - \lambda(x_k+a_sh)^m + m(x_k+a_sh)^{m-1} \end{pmatrix} = \frac{(x_k+h)^m - x_k^m}{h} + O(h^r).$$

注意到，当 h 充分小时，$(I-\lambda hB)^{-1} = \sum_{i=0}^{\infty} (\lambda hB)^i$．比较上面式子中 h^i 的各项系数，得

$$\begin{pmatrix} c_1 & \cdots & c_s \end{pmatrix} \begin{pmatrix} 1 & a_1 & \cdots & a_1^{r-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & a_s & \cdots & a_s^{r-1} \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{r} \end{pmatrix}, \tag{7.3}$$

$$\begin{aligned} \begin{pmatrix} c_1 & \cdots & c_s \end{pmatrix} B \begin{pmatrix} 1 & a_1 & \cdots & a_1^{r-2} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & a_s & \cdots & a_s^{r-2} \end{pmatrix} &= \begin{pmatrix} c_1 & \cdots & c_s \end{pmatrix} \begin{pmatrix} a_1 & a_1^2 & \cdots & a_1^{r-1} \\ \vdots & \vdots & \cdots & \vdots \\ a_s & a_s^2 & \cdots & a_s^{r-1} \end{pmatrix} \begin{pmatrix} 1 & & & \\ & \frac{1}{2} & & \\ & & \ddots & \\ & & & \frac{1}{r-1} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{1\cdot 2} & \frac{1}{2\cdot 3} & \cdots & \frac{1}{(r-1)r} \end{pmatrix}. \end{aligned} \tag{7.4}$$

例 7.4. 一些显式 Runge-Kutta 公式. 设 $K_1 = f(x_k, y_k)$.

- $r = s = 1$, $y_{k+1} = y_k + hf(x_k, y_k)$.
- $r = s = 2$, $y_{k+1} = y_k + h\Big(c_1 f(x_k, y_k) + c_2 f(x_k + a, y_k + hbf(x_k, y_k))\Big)$, 其中 a, b, c_i 满足

$$\begin{pmatrix} c_1 & c_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & a \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{2} \end{pmatrix}, \quad \begin{pmatrix} c_1 & c_2 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ b & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{2} \quad \text{即} \quad \begin{cases} c_1 = 1 - c_2 \\ a = b = \dfrac{1}{2c_2} \end{cases}.$$

- $r = s = 3$, $y_{k+1} = y_k + h(c_1 K_1 + c_2 K_2 + c_3 K_3)$, $K_1 = f(x_k, y_k)$, $K_2 = f(x_k + a_2 h, y_k + hb_{21} K_1)$, $K_3 = f(x_k + a_3 h, y_k + hb_{31} K_1 + hb_{32} K_2)$, 其中 a_i, b_{ij}, c_i 满足

$$\begin{pmatrix} c_1 & c_2 & c_3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & a_2 & a_2^2 \\ 1 & a_3 & a_3^2 \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \end{pmatrix}, \quad \begin{pmatrix} c_1 & c_2 & c_3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ b_{21} & 0 & 0 \\ b_{31} & b_{32} & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & a_2 & a_2^2 \\ 1 & a_3 & a_3^2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & a_2 \\ 1 & a_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} \end{pmatrix}.$$

  共计 5 个方程和 8 个未知数.

□

关于 Runge-Kutta 公式的几个注记：

- 由于 Runge-Kutta 方法是单步法，定理 7.1 的结论同样适用于 Runge-Kutta 方法.
- 给定 r, s，满足条件 (7.3) 和 (7.4) 的常数 a_i, b_{ij}, c_i 可能不存在，也可能不唯一.
- 由 (7.3) 式可得 $r \leqslant 2s$. 并且，存在 s 级隐式 Runge-Kutta 公式具有 $2s$ 阶局部截断误差.
- 当 Runge-Kutta 公式是显式时，由 $B^s = O$，结合 (7.3)和 (7.3) 式，得 $r \leqslant s$.
- 下表是一些 s 级显式 Runge-Kutta 公式的局部截断误差所能达到的最高阶数 r.

s	1	2	3	4	5	6	7	8	9	10	11
r	1	2	3	4	4	5	6	6	7	7	8

§7.3 线性多步法

定义 7.5. 若问题 (7.1) 的数值解法公式具有形式

$$y_k = \sum_{i=1}^{p} a_i y_{k-i} + \sum_{i=0}^{p} b_i f(x_{k-i}, y_{k-i}), \quad k \geqslant p \tag{7.5}$$

其中 p 是给定的正整数，a_i, b_i 是常数，$y_1, \cdots, y_{p-1}$ 由其他方法确定，则 (7.5) 式称为 p **步的线性多步法**公式．若 $b_0 = 0$，则 (7.5) 式称为**显式**的，否则称为**隐式**的．特别，形如

$$y_k = y_{k-1} + \sum_{i=0}^{q} b_i f(x_{k-i}, y_{k-i}), \quad k \geqslant q$$

的线性多步法称为 **Adams 方法**，显式的 Adams 方法称为 **Adams-Bashforth 方法**，隐式的 Adams 方法称为 **Adams-Moulton 方法**.

Adams 方法源于数值积分

$$y(x_k) = y(x_{k-1}) + \int_{x_{k-1}}^{x_k} f(x, y(x))dx \approx y(x_{k-1}) + \int_{x_{k-1}}^{x_k} \phi(x)dx$$

其中 $\phi(x)$ 是 $f(x, y(x))$ 的以 $\{x_{k-q}, \cdots, x_{k-1}\}$ 或 $\{x_{k-q}, \cdots, x_k\}$ 为插值节点的插值多项式．根据定理 6.4，数值积分的误差为 $O((x_k - x_{k-1})^{q+1})$ 或 $O((x_k - x_{k-1})^{q+2})$．因此，数值积分方法得到的显式 Adams 公式有 q 阶局部截断误差，隐式 Adams 公式有 $q+1$ 阶局部截断误差.

线性多步法 (7.5) 的收敛性和稳定性分析比较复杂，此处不详细叙述.

例 7.5. 设 $a = x_0 < \cdots < x_n = b$，$h = \dfrac{b-a}{n}$，$p, q$ 是正整数. 构造线性多步法公式

$$x_k = x_{k-p} + h\sum_{i=0}^{q} b_i f(x_{k-i}, y_{k-i}).$$

解答. $b_0, \cdots, b_q$ 应满足

$$\int_0^{ph} \phi(x)dx = h\sum_{i=0}^{q} b_i \phi(ih), \quad \forall \phi(x) = 1, x, \cdots, x^q.$$

由此得线性方程组

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & q \\ \vdots & \vdots & (j-1)^{i-1} & \vdots \\ 0 & 1 & \cdots & q^q \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_q \end{pmatrix} = \begin{pmatrix} p \\ \frac{p^2}{2} \\ \vdots \\ \frac{p^{q+1}}{q+1} \end{pmatrix}.$$

特别，对于显式公式，$b_0 = 0$，

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & q \\ \vdots & \vdots & j^{i-1} & \vdots \\ 1 & 2^{q-1} & \cdots & q^{q-1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_q \end{pmatrix} = \begin{pmatrix} p \\ \frac{p^2}{2} \\ \vdots \\ \frac{p^q}{q} \end{pmatrix}.$$

□

§7.4 常微分方程组

Cauchy 问题 (7.1) 可以自然地推广到一元常微分方程组初值问题，写成向量形式，

$$\begin{cases} \boldsymbol{y}'(x) = F(x, \boldsymbol{y}(x)), \quad x \in [a, b] \\ \boldsymbol{y}(a) = \boldsymbol{c} \end{cases}$$

其中 $\boldsymbol{y} = (y_1, \cdots, y_m)$，$F = (f_1, \cdots, f_m)$，每个 $f_i(x, y_1, \cdots, y_m)$ 是给定的多元可微函数，a, b 是给定的实数，$\boldsymbol{c} = (c_1, \cdots, c_m)$ 是给定的实数向量．前面几节所述的各种数值解法及其误差估计都可以自然的推广过来.

例 7.6. 记 $h = x_{k+1} - x_k$.

- 向前 Euler 公式 $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + hF(x_k, \boldsymbol{y}_k)$．向后 Euler 公式 $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + hF(x_k, \boldsymbol{y}_{k+1})$.
- Heun 公式 $\widetilde{\boldsymbol{y}}_{k+1} = \boldsymbol{y}_k + hF(x_k, \boldsymbol{y}_k)$，$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h\dfrac{F(x_k, \boldsymbol{y}_k) + F(x_{k+1}, \widetilde{\boldsymbol{y}}_{k+1})}{2}$.
- s 级显式 Runge-Kutta 公式 $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h\sum\limits_{i=1}^{s} c_i K_i$，其中 $K_i = f(x_k, \boldsymbol{y}_k + h\sum\limits_{j=1}^{i-1} b_{ij} K_j)$，$\forall i$.
- q 阶显式 Adams 公式 $\boldsymbol{y}_k = \boldsymbol{y}_{k-1} + h\sum\limits_{i=1}^{q} b_i f(x_{k-i}, \boldsymbol{y}_{k-i})$.

例 7.7. 考虑 m 阶常微分方程初值问题

$$\begin{cases} y^{(m)}(x) = f(x, y(x), y'(x), \cdots, y^{(m-1)}(x)), \quad x \in [a, b] \\ y^{(i)}(a) = c_i, \quad i = 0, 1, \cdots, m-1. \end{cases}$$

记 $\boldsymbol{y} = (y_1, \cdots, y_m)$，$\boldsymbol{c} = (c_0, \cdots, c_{m-1})$，$F(x, \boldsymbol{y}) = \Big(y_2, \cdots, y_m, f(x, y_1, \cdots, y_m)\Big)$. 上述问题可以化为常微分方程组初值问题

$$\begin{cases} \boldsymbol{y}'(x) = F(x, \boldsymbol{y}(x)), \quad x \in [a, b] \\ \boldsymbol{y}(a) = \boldsymbol{c} \end{cases}$$

方程组的解 $y_1(x)$ 就是原方程的解 $y(x)$.

第八章　矩阵特征值 (6 学时)

定义 8.1. 设 $A \in \mathbb{C}^{n\times n}$．若 $\lambda \in \mathbb{C}$ 和非零向量 $\alpha \in \mathbb{C}^{n\times 1}$ 满足

$$A\alpha = \lambda\alpha$$

则 λ 称为 A 的一个**特征值**，α 称为 A 的一个与 λ 对应的**特征向量**．一元多项式

$$\varphi_A(x) = \det(xI - A)$$

称为 A 的**特征多项式**．$\varphi_A(x)$ 的所有复数根（含重根）即为 A 的所有特征值（含重数）.

一方面，当 $\varphi_A(x)$ 容易精确求得时，可以通过求 $\varphi_A(x)$ 的根得到 A 的特征值.

另一方面，也可以通过求 $A = \begin{pmatrix} & & & c_0 \\ -1 & & & c_1 \\ & \ddots & & \vdots \\ & & -1 & c_{n-1} \end{pmatrix}$ 的特征值，得到 $\varphi_A(x) = x^n + \sum\limits_{i=0}^{n-1} c_i x^i$ 的根.

定理 8.1. 关于复数方阵 $A \in \mathbb{C}^{n\times n}$ 的所有特征值 $\lambda_1, \cdots, \lambda_n$，有如下常用结论.

1. $\lambda_1 - \mu, \cdots, \lambda_n - \mu$ 是 $A - \mu I$ 的所有特征值. 当 A 可逆时，$\frac{1}{\lambda_1}, \cdots, \frac{1}{\lambda_n}$ 是 A^{-1} 的所有特征值.

2. 存在可逆方阵 P，使得 $P^{-1}AP$ 是 *Jordan* 标准形 $\mathrm{diag}\Big(J_{n_1}(\lambda_1), \cdots, J_{n_k}(\lambda_k)\Big)$，其中

$$J_{n_i}(\lambda_i) = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \lambda_i & 1 \\ & & & \lambda_i \end{pmatrix} \in \mathbb{C}^{n_i \times n_i}.$$

3. 存在酉方阵 P，使得 $P^{-1}AP$ 是上三角方阵并且对角元素依次为 $\lambda_1, \cdots, \lambda_n$.

4. 若 A 是 *Hermite* 的，则 $\lambda_1, \cdots, \lambda_n$ 都是实数，并且存在酉方阵 P 使得 $P^{-1}AP = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$.

定理 8.2. 关于实数方阵 $A \in \mathbb{R}^{n\times n}$ 的所有特征值 $\lambda_1, \cdots, \lambda_n$，有如下常用结论.

1. 若 λ 是 A 的特征值，则 $\overline{\lambda}$ 也是 A 的特征值.

2. 若 n 是奇数，则 A 必有实数特征值.

3. 设 $|\lambda_1| \geqslant \cdots \geqslant |\lambda_n|$. 若 $|\lambda_1| > |\lambda_2|$，则 λ_1 是实数. 若 $|\lambda_{n-1}| > |\lambda_n|$，则 λ_n 是实数.

4. 存在正交方阵 P，使得 $P^{-1}AP$ 是准上三角方阵，并且每个准对角块是 *1* 或 *2* 阶方阵.

5. 若 A 是对称的，则 $\lambda_1, \cdots, \lambda_n$ 都是实数，并且存在正交方阵 P 使得 $P^{-1}AP = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$.

§8.1 幂法反幂法

定理 8.3. 设 $A \in \mathbb{C}^{n\times n}$ 的特征值 $\lambda_1, \cdots, \lambda_n$ 满足 $|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|$. 记

$$\beta_k = A\alpha_k, \quad x_k = \alpha_k^T \beta_k, \quad \alpha_{k+1} = \frac{\beta_k}{\|\beta_k\|_2}, \quad k \in \mathbb{N}.$$

当 $\prod_{i=2}^{n}(A - \lambda_i I)\alpha_0 \neq \mathbf{0}$ 时，$\lim_{k\to\infty} x_k = \lambda_1$ 并且 $\lim_{k\to\infty} \alpha_k$ 是 λ_1 对应的特征向量.

证明. 根据定理 8.1 (2)，存在可逆方阵 P，使得 $P^{-1}AP = \mathrm{diag}(\lambda_1, B)$，其中 $\varphi_B(x) = \prod_{i=2}^{n}(x - \lambda_i)$. 由 $\prod_{i=2}^{n}(A - \lambda_i I)\alpha_0 = P \begin{pmatrix} \varphi_B(\lambda_1) & \\ & O \end{pmatrix} P^{-1}\alpha_0 \neq \mathbf{0}$，得 $P^{-1}\alpha_0 = (c_1, \cdots, c_n)^T$，$c_1 \neq 0$. 当 $k \to \infty$ 时，

$$\lambda_1^{-k} A^k \alpha_0 = P \begin{pmatrix} 1 & \\ & \lambda_1^{-k} B^k \end{pmatrix} P^{-1}\alpha_0 \to c_1 P\boldsymbol{e}_1.$$

因此，$\alpha_k = \dfrac{A^k\alpha_0}{\|A^k\alpha_0\|_2} \to \dfrac{P\boldsymbol{e}_1}{\|P\boldsymbol{e}_1\|_2}$ 是 λ_1 对应的特征向量，$\beta_k \to \lambda_1\alpha_k$，$x_k \to \lambda_1$. □

由定理 8.3 可得计算 A 的最大模特征值及其对应的特征向量的幂法.

例 8.1 (**幂法**). 输入方阵 $A \in \mathbb{C}^{n\times n}$ 满足 $|\lambda_1| > |\lambda_2|$，输出 λ_1 及对应的特征向量 α_1 的近似值.

```
u=RandomUnitVector(); for(k=1; k<=MaxCount; k++)
{
  v=Multiplication(a,u); t=0; for(i=1; i<=n; i++) t+=Conjugate(u[i])*v[i];
  if(Norm(t*u-v)<epsilon) return (t,u); r=Norm(v); for(i=1; i<n; i++) u[i]=v[i]/r;
}
return (t,u);
```

对 A^{-1} 使用幂法，可得计算 A 的最小模特征值及其对应的特征向量的反幂法.

例 8.2 (**反幂法**). 输入方阵 $A \in \mathbb{C}^{n\times n}$，假设 $|\lambda_n| < |\lambda_{n-1}|$，输出 λ_n 及对应的特征向量 α_n 的近似值.

```
u=RandomUnitVector(); for(k=1; k<=MaxCount; k++)
{
  v=LinearSolve(a,u); t=0; for(i=1; i<=n; i++) t+=Conjugate(u[i])*v[i];
  if(Norm(u-v/t)<epsilon) return (1/t,u); r=Norm(v); for(i=1; i<n; i++) u[i]=v[i]/r;
}
return (1/t,u);
```

在 A^{-1} 未知或者不容易求得的情形下，一般通过线性方程组 $A\boldsymbol{x}=\boldsymbol{u}$ 计算 $A^{-1}\boldsymbol{u}$. 故反幂法的计算量通常大于幂法的计算量.

在使用幂法、反幂法之前，通常对 A 作预处理. 由定理 8.3 还可知，幂法的收敛速度依赖于 $\dfrac{|\lambda_2(A)|}{|\lambda_1(A)|}$，反幂法的收敛速度依赖于 $\dfrac{|\lambda_n(A)|}{|\lambda_{n-1}(A)|}$，比值越小收敛越快. 根据定理 8.1 (1)，可通过 $B=A-\mu I$ 的特征值求 A 的特征值，这种方法称为**位移法**. 在不知道所有特征值的近似值的情况下，求 μ 使得 $\dfrac{|\lambda_2(B)|}{|\lambda_1(B)|}$ 比较小是非常困难的. 在知道 λ_n 的近似值的情况下，求 μ 使得 $\dfrac{|\lambda_n(B)|}{|\lambda_{n-1}(B)|}$ 比较小是容易办到的. 由此可把反幂法与位移法相结合.

例 8.3 (**位移反幂法**). 输入方阵 $A\in\mathbb{C}^{n\times n}$ 和 $b\in\mathbb{C}$，假设 A 有唯一的特征值 λ 使得 $|\lambda-b|$ 最小，输出 λ 及对应的特征向量 α 的近似值.

```
s=b; u=InitialUnitVector(); for(k=1; k<=MaxCount; k++)
{
  v=LinearSolve(a-s*I,u); t=0; for(i=1; i<=n; i++) t+=Conjugate(u[i])*v[i]; s+=1/t;
  if(Norm(u-v/t)<epsilon) return (s,u); r=Norm(v); for(i=1; i<n; i++) u[i]=v[i]/r;
}
return (s,u);
```

尽管在位移反幂法中，$A - s_k I$ 变得越来越奇异，但是位移反幂法的实际效果非常良好.

位移法是一种预处理. Hessenberg 约化方法是另一种预处理.

满足 $a_{ij} = 0$，$\forall i \geqslant j+2$ 的方阵 (a_{ij}) 称为 **Hessenberg 方阵**.

例 8.4 (Hessenberg 约化). 设酉方阵 P 使得 $\widetilde{A} = (\widetilde{a}_{ij}) = PAP^H$ 是 Hessenberg 方阵，则 $\widetilde{A}\boldsymbol{v}$ 和 $\widetilde{A}^{-1}\boldsymbol{v}$ 的计算量都可以大幅降低. 酉方阵 P 的构造如下.

- 设 $r = \sqrt{\sum_{i=2}^{n} |a_{i1}|^2}$，$\boldsymbol{u} = (a_{21}, \cdots, a_{n1})^T - r\boldsymbol{e}_1$，则 $Q_1 = I_{n-1} - \dfrac{2}{\boldsymbol{u}^H\boldsymbol{u}}\boldsymbol{u}\boldsymbol{u}^H$ 是酉方阵，$Q_1\boldsymbol{u} = r\boldsymbol{e}_1$.

  酉方阵 $P_1 = \begin{pmatrix} 1 & \\ & Q_1 \end{pmatrix}$ 使得 $P_1 A P_1^H = \begin{pmatrix} a_{11} & * \\ r\boldsymbol{e}_1 & A_1 \end{pmatrix}$.

- 对 A_1 重复上述操作，得酉方阵 $P_2, \cdots, P_{n-2}$，则 $P = P_{n-2}\cdots P_1$ 为所求.

设 λ 和 α 是 $\widetilde{A}$ 的特征值和对应的特征向量，则 λ 和 $P^H\alpha$ 是 A 的特征值和对应的特征向量. □

利用幂法的思想，也可以求 A 的前 m 个最大模特征值，$\forall m = 1, \cdots, n$.

定理 8.4. 设 $A \in \mathbb{C}^{n\times n}$ 的特征值 $\lambda_1, \cdots, \lambda_n$ 满足 $|\lambda_1| > \cdots > |\lambda_m| > |\lambda_{m+1}| \geqslant \cdots \geqslant |\lambda_n|$. 根据 *Gram-Schmidt* 标准正交化算法，对于任意 $k \in \mathbb{N}$，AP_k 可唯一地表示成

$$AP_k = P_{k+1}R_k \tag{8.1}$$

其中 $P_k \in \mathbb{C}^{n\times m}$ 满足 $P_k^H P_k = I_m$，$R_k \in \mathbb{C}^{m\times m}$ 是上三角方阵且对角元素都是正数.

当 $\prod_{i=m+1}^{n}(A-\lambda_i I)P_0$ 是列满秩时，$\lim_{k\to\infty} P_k^H A P_k$ 是上三角方阵且对角元素是 $\lambda_1,\cdots,\lambda_m$.

证明. $P_k = A^k P_0 (R_{k-1}\cdots R_0)^{-1}$，与定理 8.3 的证明类似，详细过程略. □

例 8.5 (QR 迭代法). 设可逆方阵 $A \in \mathbb{C}^{n\times n}$ 的特征值 $\lambda_1,\cdots,\lambda_n$ 满足 $|\lambda_1| > \cdots > |\lambda_n|$. 记

$$A = Q_0 R_0, \quad R_k Q_k = Q_{k+1} R_{k+1}, \quad k \in \mathbb{N} \tag{8.2}$$

其中 $Q_k R_k$ 是 QR 分解（即 Q_k 是酉方阵，R_k 是上三角方阵且对角元素都是正数），则 $\lim_{k\to\infty} R_k Q_k$ 收敛到上三角. (4.9) 式称为 **QR 迭代**.

- 注意到 $A^k = (Q_0R_0)^k = Q_0(R_0Q_0)^{k-1}R_0 = Q_0(Q_1R_1)^{k-1}R_0$

  $$= Q_0Q_1(R_1Q_1)^{k-2}R_1R_0 = Q_0Q_1(Q_2R_2)^{k-2}R_1R_0 = \cdots = (Q_0\cdots Q_{k-1})(R_{k-1}\cdots R_0).$$

  在 (8.1) 式中，设 $m=n$，$P_0 = I$，则 $P_k = Q_0\cdots Q_{k-1}$ 并且 (8.1) 式中的 $R_0,\cdots,R_{k-1}$ 与 (8.2) 式中的相同. 因此，QR 迭代法可以看作是幂法在特殊情形下的变形.

- R_kQ_k 与 Q_kR_k 是酉相似的，故它们都与 A 是酉相似的. QR 迭代实际上是计算 A 的酉相似于上三角的标准形. QR 迭代法也可以与位移法相结合，以提高收敛速度. □

§8.2 Jacobi 方法

例 8.6. 对于任意 Hermite 方阵 $A \in \mathbb{C}^{n\times n}$，可以选取一系列酉方阵 $\{P_k\}$，构造序列

$$A_0 = A, \quad A_{k+1} = P_k^H A_k P_k, \quad k \in \mathbb{N}$$

使得 A_k 收敛到对角方阵，其对角元素即为 A 的所有特征值. 这种方法称为 **Jacobi 方法**. 原理如下.

① 设 $A = (a_{ij})$，寻找 $s < t$ 使得 $|a_{st}| = \max\limits_{i\neq j} |a_{ij}|$.

② 求 2 阶酉方阵 $Q = \begin{pmatrix} q_{ss} & q_{st} \\ q_{ts} & q_{tt} \end{pmatrix}$ 使得 $Q^H \begin{pmatrix} a_{ss} & a_{st} \\ a_{ts} & a_{tt} \end{pmatrix} Q = \begin{pmatrix} b_{ss} & 0 \\ 0 & b_{tt} \end{pmatrix}$.

③ 扩充 Q 为 n 阶酉方阵 $P = (p_{ij})$，$p_{ij} = \begin{cases} q_{ij}, & i,j \in \{s,t\} \\ \delta_{ij}, & \text{其他} \end{cases}$. 设 $B = P^H AP = (b_{ij})$.

由 P, Q 是酉方阵，得 $\sum\limits_{i,j} |b_{ij}|^2 = \sum\limits_{i,j} |a_{ij}|^2$，$|b_{ss}|^2 + |b_{tt}|^2 = |a_{ss}|^2 + |a_{tt}|^2 + 2|a_{st}^2|$.

再由 $b_{ii} = a_{ii}$，$\forall i \notin \{s,t\}$，得 $\sum\limits_{i\neq j} |b_{ij}|^2 = \sum\limits_{i\neq j} |a_{ij}|^2 - 2|a_{st}|^2 \leqslant \dfrac{n^2-n-2}{n^2-n} \sum\limits_{i\neq j} |a_{ij}|^2$.

④ 随着迭代次数的增加，$\sum\limits_{i\neq j} |b_{ij}|^2 \to 0$. 把 A 换成 B，`goto` ①，开始下一轮迭代. □

尽管 Jacobi 方法简单易行，但是收敛速度比较慢．通常先用 Hessenberg 约化方法把 Hermite 方阵 A 酉相似成三对角方阵，再使用其他方法求 A 的所有特征值．

作为矩阵特征值问题的应用，可用如下方法计算矩阵的奇异值分解．

例 8.7. 对于任意 $A \in \mathbb{C}^{m\times n}$，$m \leqslant n$，设 $AA^H = P\,\mathrm{diag}(\lambda_1, \cdots, \lambda_m)P^H$，其中 $P \in \mathbb{C}^{m\times m}$ 是酉方阵，$\lambda_1 \geqslant \cdots \geqslant \lambda_m \geqslant 0$．$P^HA$ 的行向量相互正交．标准化 P^HA 的非零行向量，并扩充为酉方阵 Q．由此可得

$$A = P \begin{pmatrix} \sqrt{\lambda_1} & & \\ & \ddots & \\ & & \sqrt{\lambda_n} \end{pmatrix} Q.$$

上式称为 A 的**奇异值分解**．当 $m \geqslant n$ 时，可由 A^T 的奇异值分解求得 A 的奇异值分解．

对于任意 $A \in \mathbb{R}^{m\times n}$，可把上述结论中的酉方阵 P, Q 换成正交方阵 P, Q，结论仍然成立． □