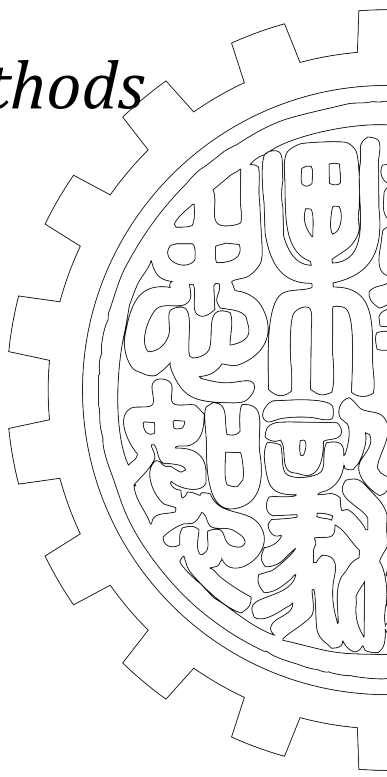
计算方法撷英

Notes on Computing Methods

作者：王天浩，尤佳睿

2019 年 11 月 24 日



钱学森书院学业辅导中心

Qian Yuan Xue Fu

XI'AN JIAOTONG UNIVERSITY

作品信息

➤ **标题：**计算方法撷英 - *Notes on Computing Methods*
➤ **作者：**王天浩，尤佳睿
➤ **校对排版：**xjtu-blacksmith
➤ **出品时间：**2019 年 11 月 24 日
➤ **总页数：**61

许可证说明

本作品已发布于 GitHub 之上，发布地址为：
`https://github.com/qyxf/notes-on-computing-methods`
本作品的版本号为 1.1。

前言

计算方法是本校（西安交通大学）钱学森班、少年班等拔尖班的同学必修的一门数学基础课程，其内容包括解决一些计算问题（如解线性方程组、插值与逼近、数值微积分等）的常用算法，及针对这些算法之误差与稳定性的相关数学理论。相较于其他的数学课程，计算方法课程具有内容丰富、实用性强等特点，但也兼有一般数学课程内容深、考点多的性质。可以说，计算方法课程是工科学生在走向专业课程之前所要翻越的最后一座数学理论「大山」。

2014 级少年班的王天浩同学，于此方面做了相当好的工作，在学习本课程期间尽心尽力整理了十分详尽的课程笔记，基本上达到了以上所提到的要求。尽管该份笔记是纸质版扫描而成，但一经发布便广泛流传开来，成为当届及本届学生复习时重要的参考资料。为方便之后的同学复习计算方法课程，经与王天浩学长协商，我自 2019 年 6 月 6 日开始将原有的纸质扫描笔记以 LaTeX 整理为电子版，补足了语言上的一些省略、缺失，增写了许多注记、说明、插图，并扩充了一些新的条目。新的电子版本定名为《计算方法撷英》，以示此份文档与「计算方法」课程之关系。

正文格式说明

电子版笔记基本上遵循了原有纸质笔记的框架，但相较与原来的样式更为清晰、简明。这份笔记的正文中包含了这样几类内容：

条目 用 ➢ 符号引导的内容，附有编号，为一般性质的知识点、说明。

例题 用 ✐ 符号引导的内容，附有编号，通常是对其正上方那一个或几个知识点的具体呈现与演示。一些给出详细过程，一些仅给出答案（供读者自行练习时参考）。

定理/定义 用 ☞ 符号引导的内容，附有编号，表示一些比较重要的定理和概念定义。

关键词 用**粗体**标注的内容，表示比较关键的概念。可在本笔记后的「索引」栏目查找本笔记中所有的关键词。

强调 用下划线标注的内容，表示强调，以与其他内容区分开来。

注记 用脚注的方式给出，通常是对正文内容的进一步阐释，或对正文中略去之内容的说明。

撰写说明

此份笔记自 2019.6.6 开始撰写，期间编者尚在美国交流，只能利用课余时间零碎增补；至 9 月回国时，正文内容几近完成。在编者缓考完成后，对内容的润色工作暂时搁置，直至 11 月时才重新启动，并最终完成。在编者整理稿件的过程中，除以王天浩同学的纸质笔记为底稿之外，也在获得许可的情况下参考了 2017 级钱班吴思源同学的计算方法笔记，在此向二位深表感谢！

在此之外，编者还要向授课的马军老师，及编撰本校计算方法相关教材的李乃成、邓建中、梅立泉等诸位老师表示敬意。作为一门与科技前沿紧密相关的数学基础课，授课者们担负的责任异常重大，而他们的表现则异常令人钦佩！

帮助我们改进这份笔记

一本好的教科书，来自于相关教师历经数代、数十年的逐次再版改进；一份好的笔记，同样也需要长期的维护、改进才能够最终创造出来。本份笔记还未经这样久的磨砺，在内容、布局、细节等方面都相当欠缺；因此，恳请诸位读者在使用本份笔记时，留意以下几点：

- 检查正文中存在的笔误、错别字、公式错误等；
- 考察此份笔记是否缺少课程相关的知识点、章节内容；
- 评价本笔记中是否有详略不得体、例题过少、描述不明晰的内容。

以上三点，「境界」逐次提高，却都能够有效提高这份笔记的可用程度。若读者在阅读过程中发现以上三点问题，请将问题整理好，寄送到编者的电子邮箱：`yjr134@163.com`；如您常年使用 GitHub，也可在本份笔记的开源仓库页面 `https://github.com/qyxf/notes-on-computing-methods` 上发布 issue 或 pull request，提出改进建议。

非常感谢各位读者的贡献。祝愿大家在考试中取得令人满意的成绩，并能够在实际问题中更为熟练的应用各类数值计算方法。

能动少 C71 尤佳睿 [1]

2019 年 11 月 24 日

[1] 个人博客：`https://www.cnblogs.com/xjtu-blacksmith/`。

目录

第一章　误差

§1.1　真值与误差

➢**1.1**　有测量就会有误差。通常，将某数学量、物理量的**真值**记为 x（不加任何修饰符），而将测量或计算所得的 x 的**近似值**记作 $\tilde{x}$。

➢**1.2**　两种误差：$\begin{cases}\Delta x = x - \tilde{x} \text{ (绝对误差)} \\ \delta x = \frac{x-\tilde{x}}{x} \text{ (相对误差)}\end{cases}$

➢**1.3**　两种误差限：$\begin{cases}|\Delta x| \le \varepsilon \text{ (绝对误差限)} \\ |\delta x| \le \varepsilon_r \text{ (相对误差限)}\end{cases}$

➢**1.4**　相对误差较小时，有近似计算式[1]：$|\delta x| \approx \frac{|x-\tilde{x}|}{\tilde{x}} = \frac{\Delta x}{\tilde{x}} \le \frac{|\varepsilon|}{\tilde{x}}$

➢**1.5**　若 $|\Delta x| = |x - \tilde{x}| \le 0.5 \times 10^{-n}$，则称 x 的近似值 $\tilde{x}$ 准确到第 n 位小数。

✐**1.6**　设 $x = 0.31682$，则 $\tilde{x}_1 = 0.3$ 精确到 1 位有效数字，$\tilde{x}_2 = 0.32$ 精确到 2 位，$\tilde{x}_3 = 0.317$ 精确到 3 位，$\tilde{x}_4 = 0.3168$ 精确到 4 位。若取 $\tilde{x}_5 = 0.3169$ 为 x 的近似值，则其仅精确到 3 位小数。

§1.2　浮点运算与浮点数集

➢**1.7**　在计算机中，实数将被储存为**浮点数**，故计算机中的实数运算常被称作**浮点运算**。为此，有下面的一些概念与理论。

➢**1.8**　**浮点运算量**：记一次加法和一次乘法（如 $a + b \times c$）所需的时间为一个**时间单位**，记为 flop。

✐**1.9**　设 $\mathbf{A}_1$ 为一 10×20 的矩阵，$\mathbf{A}_2$ 为一 20×50 的矩阵，欲计算 $\mathbf{A}_1 \cdot \mathbf{A}_2$，则运算量为 $10 \times 20 \times 50 = 10000$ flop [2]。

---

[1] 在估计误差时，真值 x 往往难以确定，但绝对误差 $|\Delta x|$ 或绝对误差限 ε 往往能够确定下来。

[2] $\mathbf{A}_1$ 的行乘以 $\mathbf{A}_2$ 的列，每次的浮点运算量为 20 flop，总计 10×50 种组合。

➢**1.10** **浮点数集**：在 10 进制中，浮点数 $\tilde{x}$（或一实数 x 的近似 t 位有效数字的浮点数 $\tilde{x}$）可表示如下：

$$fl(x) = \tilde{x} = \pm\left\{\frac{x_1}{10} + \frac{x_2}{10^2} + \frac{x_3}{10^3} + \cdots + \frac{x_t}{10^t}\right\} \times 10^l \ (\tilde{x} = 0.x_1x_2x_3\cdots x_t \times 10^l)$$

其中 $1 \le x_1 < 10$，$0 \le x_j < 10$，$j = 2, 3, \cdots, t$。类似的，在 β 进制中，一个数的表示方式：

$$fl(x) = \tilde{x} = \pm\left\{\frac{x_1}{\beta} + \frac{x_2}{\beta^2} + \cdots + \frac{x_t}{\beta^t}\right\} \times \beta^l$$

其中 $1 \le x_1 < \beta$，$0 \le x_j < \beta$，$j = 2, 3, \cdots, t$。β^l 称为指数部分，指数 l 满足 $L \le l \le U$，L 与 U 分别为下界与上界；$0.x_1x_2\cdots x_t$ 称为尾数。$fl(x)$ 称为一个**规格化浮点数**。

➢**1.11** 称计算机中所能表示的全体数的集合称为**浮点数集**，记为 $F(\beta, t, L, U)$。

$$F(\beta, t, L, U) = \{0\} \cup \left\{\pm\left(\frac{x_1}{\beta} + \frac{x_2}{\beta^2} + \cdots + \frac{x_t}{\beta^t}\right) \times \beta^l : L \le l \le U\right\} \tag{1.1}$$

✎**1.12** C++ 里的 `float`：4 字节、32 位，如图 1.1 所示。可以将这一浮点数集记为 $F(2, 23, -128, 127)$。

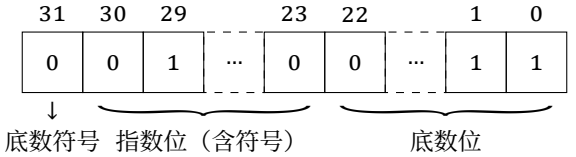


图 1.1: C++ 中 `float` 类型变量的储存原理

➢**1.13** 浮点数集中的数的个数：$N = 2 \cdot (\beta - 1) \cdot \beta^{t-1} \cdot (U - L + 1) + 1$

➢**1.14** 浮点数 $fl(x)$ 与对应真值 x 的误差：

- 绝对误差：$|x - fl(x)| \le \frac{1}{2}\beta^{-t} \times \beta^l = \frac{1}{2}\beta^{l-t}$
- 相对误差：由 $|x| \ge 0.1 \times \beta^l$，有 $\frac{|x - fl(x)|}{|x|} \le \frac{\beta^{l-t}/2}{\beta^{l-1}} = \frac{1}{2}\beta^{1-t}$

此类误差称为**舍入误差**。

➢**1.15** 计算结果的错误/误差：

1. $l \notin [L, U]$：**上溢**（$l \ge L$）会出错，**下溢**（$l \le U$）变为 0。
2. 尾数多于 t 位：自动进行舍入处理，造成误差
3. 有效数字丢失：「大数吃小数」

✍**1.16** 设计算时保留 4 位有效数字，则 $1234+0.3678=1234.3678\approx 1234$，在此发生了「大数吃小数」的现象。

➢**1.17** 设计数值计算的算法时，应结合浮点数具有的特性，避免上面所提到的各类计算错误。为此，提出以下几条浮点运算原则：

1. 避免产生大结果的运算，避免小数作为除数。
2. 避免「大」、「小」数相加减，防止大数吃小数。
3. 避免相近数直接相减，防止有效数字损失。
4. 简化运算步骤，减少运算次数[(3)]。

若原有的计算公式不符合以上的这些原则，则可以通过对原式的等价变换或近似处理，使之符合上面的原则。

✍**1.18** 设 $|x|\ll 1$，则可改写数值计算公式 $\ln\dfrac{1-\sqrt{1-x^2}}{|x|}$ 为以下形式，以避免小数作分母：

$$\ln\frac{1-\sqrt{1-x^2}}{|x|}=\ln\frac{x^2}{|x|\cdot(1+\sqrt{1-x^2})}=\ln\frac{|x|}{1+\sqrt{1-x^2}}$$

§1.3 计算方法的研究内容

➢**1.19** 计算方法课程，并不仅仅包含各类数值计算方法。归结而言，计算方法课程的研究内容可以归纳为：

1. 某一问题的数值计算算法（即通常意义上的「计算方法」）；
2. 这些算法的误差、复杂性或收敛速度之估计。

后者至关重要。对于算法的误差或复杂性分析，使这门课程区别于一般的工具性课程。

➢**1.20** 针对一些模型，还存在着一类特定的问题，即**病态问题**。为度量这类问题的性质，需要用到**条件数**。

- 根据输入数据的微小变化能引起问题之解变化的大小程度，可以将数值计算问题区别为两类：若由此能引起解的很大变化，则称问题是**病态**的；否则，称一个问题是**良态**的。病态问题不易精确求解。

[(3)] 由此避免各类误差的逐次累计。——编者注

- **条件数**：输入数据 $x,\tilde{x}$，输出 $f(x),f(\tilde{x})$。设 $x\neq 0$，$f(x)\neq 0$，若存在 $m>0$ 使：

$$\frac{|f(x)-f(\tilde{x})|}{|f(x)|}\leq m\cdot\frac{|x-\tilde{x}|}{|x|}\ \text{(输出误差}\leq m\cdot\text{输入误差)}$$

  则将 m 称为该问题的**条件数**，记为 $\mathrm{Cond}(f)$。

✎**1.21** $y=\varphi(x_1,x_2,\cdots,x_n)$。输入为 $\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_n$，近似解 $\tilde{y}=\varphi(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_n)$，则有

$$\Delta y=\varphi(x_1,x_2,\cdots,x_n)-\varphi(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_n)\approx\sum_{i=1}^{n}\frac{\partial\varphi(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_n)}{\partial x_i}\Delta x_i \tag{1.2}$$

$$\delta y=\frac{\Delta y}{y}\approx\sum_{i=1}^{n}\frac{\partial\varphi(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_n)}{\partial x_i}\cdot\frac{\Delta x_i}{y}=\sum_{i=1}^{n}\frac{\partial\varphi(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_n)}{\partial x_i}\cdot\frac{x_i}{y}\cdot\delta x_i \tag{1.3}$$

故可见 $\left|\frac{\partial\varphi(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_n)}{\partial x_i}\cdot\frac{x_i}{y}\right|$ 即条件数。

➢**1.22** **稳定性**（数值稳定性）：运算中**舍入误差积累**是否影响结果的可靠性。

✎**1.23** 欲用数值计算方法求解由 $I_k=\mathrm{e}^{-1}\int_0^1 x^k\mathrm{e}^x\mathrm{d}x,\ k=0,1,\cdots,7$ 所定义的一系列定积分的值。

- 算法 1：构建递推公式

$$\begin{cases}I_0=\mathrm{e}^{-1}\displaystyle\int_0^1\mathrm{d}x=1-\frac{1}{\mathrm{e}}\\ I_k=\mathrm{e}^{-1}\displaystyle\int_0^1 x^k\mathrm{e}^x\mathrm{d}x=1-kI_{k-1}\end{cases} \tag{1.4}$$

  利用递推关系依次计算 $I_0\to I_1\to I_1\to\cdots\to I_7$。
- 算法 2：近似计算 I_7，利用递推关系 [4] 依次计算 $I_7\to I_6\to\cdots\to I_0$。

实际上就整体而言，算法 2 精度更高。对算法 1 递推公式 $I_k=1-kI_{k-1}$。若 I_{k-1} 有舍入误差 ΔI_{k-1}（或记作 Δ），则 $\tilde{I}_k=1-k(I_{k-1}+\Delta)=I_k-k\cdot\Delta$，误差被放大 [5]。

[4] 即将 (1.4) 式移项，反得 $I_{k-1}=\frac{1-I_k}{k}$。——编者注

[5] 对算法 2 的递推公式做类似分析，可见 $\tilde{I}_{k-1}=I_{k-1}-\Delta/k$，即误差被减小到原来的 $1/k$ 倍，这是大大缩小了。故算法 2 较算法 1 更为稳定。——编者注

第二章　线性方程组直接解法

§2.1　Gauss 消元法的引入

➢$_{2.1}$ 整体思路：

1. 先推得 $\mathbf{Ax}=\mathbf{b}\ \Rightarrow\ \mathbf{x}=\mathbf{A}^{-1}\mathbf{b}$，再求 $\mathbf{A}^{-1}$（初等行变换法、伴随矩阵法、Gauss-Jordan 消去法）
2. Crammer 法则：$\mathbf{Ax}=\mathbf{b}\ \Rightarrow\ x_i=\frac{|\mathbf{A}_i|}{|\mathbf{A}|}$，浮点运算量 $N=(n^2-1)\cdot n!+n$ flop（很大）

➢$_{2.2}$ **Gauss 消去法**：降维（$n\to(n-1)\to\cdots\to 1$）

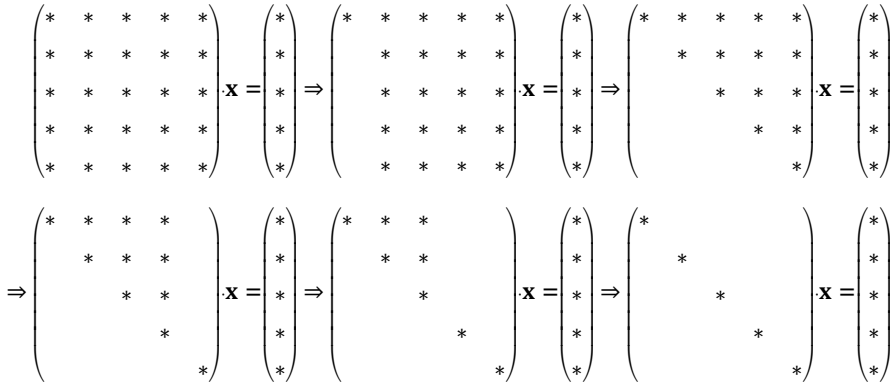


图 2.1: Gauss 消去法步骤示意图（上排降维消元，下排回代求解）

- 消去运算量：$N_1=\sum_{k=1}^{n-1}(n-k)(n-k+2)=\frac{n^3}{3}+n^2-\frac{5n}{6}$
- 回代运算量：$N_2=1+2+\cdots+n=\frac{n(n+1)}{2}$
- 总计运算量：$N=N_1+\frac{3}{2}N_2=\frac{n^3}{3}+n^2-\frac{n}{3}=O(n^3)$

➢$_{2.3}$ 可能出现的问题：（主要是消去过程中）

1. $a_{kk}^{(k-1)}=0$，无法进行
2. $|a_{kk}^{(k-1)}| \ll |a_{ik}^{(k-1)}|\ (i=k+1,k+2,\cdots,n)$，误差极大（大/小 = 大，误差被放大）

对问题 1，只要满足：(1)**A** 是方阵；(2)$|\mathbf{A}|\neq 0$，即可通过换行达到解决问题。对问题 2，则不易解决[1]。

☞2.4 $a_{kk}^{(k-1)}$ 不为 0 的充要条件是：**A** 的 1 阶与 k 阶主子式均不为 0，即

$$a_{kk}^{(k-1)}\neq 0 \Leftrightarrow D_1=a_{11}\neq 0,\ D_k=\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{vmatrix}$$

☞2.5 设矩阵 **A** 满足 $\sum_{j=1,j\neq i}^{n}|a_{ij}|<|a_{ii}|$，则称 **A** 是**严格对角占优矩阵**。

➢2.6 Gauss 消去法顺利进行条件（满足其一即可）：

1. **A** 各阶顺序主子式不等于 0。
2. **A** 是对称正定阵。
3. **A** 是严格对角占优矩阵。

§2.2 Gauss 消元法的改进

➢2.7 **列主元 Gauss 消元法**. 消去进行到第 k 步时如下所示：

$$\begin{pmatrix} a_{11}^{(k-1)} & a_{12}^{(k-1)} & \cdots & a_{1k}^{(k-1)} & \cdots & a_{1n}^{(k-1)} \\ 0 & a_{22}^{(k-1)} & \cdots & a_{2k}^{(k-1)} & \cdots & a_{2n}^{(k-1)} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{nk}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{pmatrix}$$

选取 $\max(|a_{ik}^{(k-1)}|)\ i=k,k+1,\cdots,n$ 的一行与第 k 行互换，继续消去（算法较稳定）

[1] 参见下一节中的「列主元 Gauss 消元法」。

➢**2.8** Gauss 消去法矩阵形式：将消去前后过程用矩阵表示，可以有

$$\mathbf{A}=\mathbf{A}^{(0)}=\begin{pmatrix}*&*&*&*\\*&*&*&*\\*&*&*&*\\*&*&*&*\end{pmatrix}\Rightarrow\mathbf{A}^{(1)}=\begin{pmatrix}*&*&*&*\\&*&*&*\\&*&*&*\\&*&*&*\end{pmatrix}$$

则存在唯一的单位下三角阵[2] $\mathbf{L}_1$ 使 $\mathbf{A}^{(1)}=\mathbf{L}_1\mathbf{A}^{(0)}$，其中 $\mathbf{L}_1=\begin{pmatrix}1&0&\cdots&0\\-l_{21}&1&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\-l_{n1}&0&\cdots&1\end{pmatrix}$。按此方式依次变换得一系列 $\mathbf{L}_i$：

$$\mathbf{A}^{(n)}=\mathbf{L}_n\mathbf{A}^{(n-1)}=\cdots=\mathbf{L}_n\mathbf{L}_{n-1}\cdots\mathbf{L}_2\mathbf{L}_1\mathbf{A}^{(0)}$$

反推得到

$$\mathbf{A}=\mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\cdots\mathbf{L}_{n-1}^{-1}\mathbf{A}^{(n-1)}$$

记 $\mathbf{U}=\mathbf{A}^{(n-1)}=\begin{pmatrix}\mu_{11}&\mu_{12}&\cdots&\mu_{1n}\\0&\mu_{22}&\cdots&\mu_{2n}\\\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&\mu_{nn}\end{pmatrix}$ 为一上三角阵，则 $\mathbf{A}=\mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\cdots\mathbf{L}_{n-1}^{-1}\mathbf{U}=\mathbf{LU}$。

☞**2.9** 设 $\mathbf{A}$ 为 n 阶矩阵，$D_k\neq0$，则 $\mathbf{A}$ 可唯一分解为一单位下三角阵 $\mathbf{L}$ 与一上三角阵 $\mathbf{U}$ 之积

$$\mathbf{A}=\mathbf{LU}\tag{2.1}$$

称为 **LU 分解**（Doolittle 分解）。

➢**2.10** LU 分解的算法实现：设 $\mathbf{L}=\begin{pmatrix}1&0&\cdots&0\\-l_{21}&1&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\-l_{n1}&0&\cdots&1\end{pmatrix}$，$\mathbf{U}=\mathbf{A}^{(n-1)}=\begin{pmatrix}\mu_{11}&\mu_{12}&\cdots&\mu_{1n}\\0&\mu_{22}&\cdots&\mu_{2n}\\\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&\mu_{nn}\end{pmatrix}$，根据式 (2.1) 可知：$\mathbf{A}$ 中的元素 a_{ij} 满足

$$\begin{aligned}a_{ij}&=(l_{i1}\ l_{i2}\ \cdots\ l_{i,i-1}\ 1\ 0\ \cdots\ 0)\cdot(\mu_{1j}\ \mu_{2j}\ \cdots\ \mu_{jj}\ 0\ \cdots\ 0)^{\mathrm{T}}\\&=\begin{cases}\sum_{k=1}^{i-1}l_{ik}\mu_{kj}+\mu_{ij} & ,j\geq i,\ i=1,2,\cdots,n\\\sum_{k=1}^{j}l_{ik}\mu_{kj} & ,j<i,\ i=1,2,\cdots,n\end{cases}\end{aligned}$$

[2]即线性代数中用于表征初等行变换的初等矩阵。

由此可以推得迭代算式为：

$$\mu_{1j} = a_{1j} \qquad j = 1, 2, \cdots, n \tag{2.2}$$

$$l_{i1} = a_{i1}/\mu_{11} \qquad i = 2, 3, \cdots, n \tag{2.3}$$

$$\mu_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}\mu_{kj} \qquad j = i, i+1, \cdots, n;\ i = 2, 3, \cdots, n \tag{2.4}$$

$$l_{ki} = \frac{1}{\mu_{ii}}\left(a_{ki} - \sum_{t=1}^{i-1} l_{kt}\mu_{ti}\right) \qquad k = i+1, \cdots, n;\ i = 2, 3, \cdots, n \tag{2.5}$$

➢**2.11 LU 分解算法**：

1. 照抄[3]系数矩阵 **A** 第 1 行；
2. 用式 (2.3) 写出第 1 列（**A** 对应位置元素除以 μ_{ii} 后再照抄）；
3. 用式 (2.4) 写出第 2 行（**A** 对应位置元素减去一系列 LU 乘积）；
4. 用式 (2.5) 写出第 2 列（**A** 对应位置元素减去一系列 LU 乘积，最终除以 μ_{ii}）；
5. 以此类推，重复应用式 (2.4)、式 (2.5)，生成行与列。
6. 沿对角线分成 **L**、**U** 两矩阵。

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & & & \end{pmatrix} \Rightarrow \begin{pmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ l_{21} & & & \\ \vdots & & & \\ l_{n1} & & & \end{pmatrix} \Rightarrow \begin{pmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ l_{21} & \mu_{22} & \cdots & \mu_{2n} \\ \vdots & & & \\ l_{n1} & & & \end{pmatrix}$$

$$\overset{\cdots}{\Rightarrow} \begin{pmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ l_{21} & \mu_{22} & \cdots & \mu_{2n} \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & \mu_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} + \begin{pmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ & \mu_{22} & \cdots & \mu_{2n} \\ & & \ddots & \vdots \\ & & & \mu_{nn} \end{pmatrix}$$

图 2.2: LU 分解算法示意图

➢**2.12** 需注意：重复第 5 步时，有如下的小妙招。

1. 在第 m 步生成行时，对某个元素 μ_{ij}，首先在其上方紧贴顶部的位置[4]找到 $m-1$ 个元素的列向量 $(\mu_{1j}, \mu_{2j}, \cdots, \mu_{m-1,j})$；再向左靠近左边缘位置，

[3] 即利用式 (2.2)。

[4] 而非紧贴该元素的位置。

找到 $m-1$ 个元素的行向量 $(l_{i1}, l_{i2}, \dots, l_{i,m-1})$，则 μ_{ij} 等于 $\mathbf{A}$ 对应位置元素减去以上两向量的内积。

2. 类似的，在第 m 步生成列时，先在其左侧靠边缘部找 $m-1$ 个元素的行向量，再向上靠顶部位置找到 $m-1$ 个元素的列向量，生成值为矩阵 $\mathbf{A}$ 对应位置元素值减两向量内积之后除以μ_{kk}。
3. 易错点：生成列时，不要忘记除以μ_{kk}！

$$\begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1j} & \cdots & \mu_{1n} \\ l_{21} & \mu_{22} & \cdots & \mu_{2j} & \cdots & \mu_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ l_{i1} & l_{i2} & \cdots & \mu_{ij} & \cdots & \mu_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nj} & \cdots & \mu_{nn} \end{bmatrix}$$

图 2.3: LU 分解算法图示（以一个行元素 μ_{ij} 的生成为例）

✍**2.13** 分解矩阵 $\mathbf{A}$ 为 $\mathbf{LU}$，其中：$\mathbf{A} = \begin{pmatrix} 4 & -2 & 0 & 4 \\ -2 & 2 & -3 & 1 \\ 0 & -3 & 13 & -7 \\ 4 & 1 & -7 & 23 \end{pmatrix}$

（答案：$\mathbf{L} = \begin{pmatrix} 1 & & & \\ -\frac{1}{2} & 1 & & \\ 0 & -3 & 1 & \\ 1 & 3 & \frac{1}{2} & 1 \end{pmatrix}$，$\mathbf{U} = \begin{pmatrix} 4 & -2 & 0 & 4 \\ & 1 & -3 & 3 \\ & & 4 & 2 \\ & & & 9 \end{pmatrix}$）

➢**2.14** 利用 LU 分解求解线性方程组：

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{LUx} = \mathbf{b} \Rightarrow \begin{cases} \mathbf{Ux} = \mathbf{y} \\ \mathbf{Ly} = \mathbf{b} \end{cases}$$

转化为两个对角阵方程，易于求解。（计算量仍为 $O(n^3)$，来自于 LU 分解本身。）

➢**2.15 LDU 分解**：令 $\mathbf{D} = \mathrm{diag}(\mu_{11}, \mu_{22}, \cdots, \mu_{nn})$，则有：

$$\mathbf{A} = \mathbf{LU} = \mathbf{L} \cdot \mathbf{I} \cdot \mathbf{U} = \mathbf{LDD}^{-1}\mathbf{U} = \mathbf{LDM}^{\mathrm{T}}$$

其中 $\mathbf{M}^{\mathrm{T}} = \mathbf{D}^{-1}\mathbf{U}$，$\mathbf{M}^{\mathrm{T}}$ 是一单位上三角阵。则：

$$\mathbf{M} = \begin{pmatrix} 1 & & & \\ m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ m_{n1} & m_{n2} & \cdots & 1 \end{pmatrix},\ m_{ji} = \frac{\mu_{ij}}{\mu_{ii}}\text{（即每行元素除以排头元素）} \tag{2.6}$$

称 $\mathbf{A} = \mathbf{LDM}^{\mathrm{T}}$ 为矩阵的 LDU 分解。

➢**2.16** LDU 计算式不必死记，只需在 LU 分解后变换为相应形式即可。

➢**2.17** 对于对称阵，可分解为：$\mathbf{A} = \mathbf{LDL}^{\mathrm{T}}$（前提：各阶顺序主子式非 0）

➢2.18 对于对称正定阵，$\mathbf{D}$ 的元素均非负（且对角线非 0），则进一步有 $\mathbf{A}=\mathbf{G}\mathbf{G}^{\mathrm{T}}$（Cholesky 分解）

➢2.19 **平方根法**：$\mathbf{A}=\mathbf{G}\mathbf{G}^{\mathrm{T}} \Rightarrow \mathbf{A}\mathbf{x}=\mathbf{b} \Rightarrow \mathbf{G}\mathbf{G}^{\mathrm{T}}\mathbf{x}=\mathbf{b} \Rightarrow \begin{cases}\mathbf{G}\mathbf{y}=\mathbf{b}\\ \mathbf{G}^{\mathrm{T}}\mathbf{x}=\mathbf{y}\end{cases}$

➢2.20 **改进平方根法**[5]：$\mathbf{A}=\mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}} \Rightarrow \mathbf{A}\mathbf{x}=\mathbf{b} \Rightarrow \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}\mathbf{x}=\mathbf{b} \Rightarrow \begin{cases}\mathbf{L}\mathbf{y}=\mathbf{b}\\ \mathbf{D}\mathbf{z}=\mathbf{y}\\ \mathbf{L}^{\mathrm{T}}\mathbf{x}=\mathbf{b}\end{cases}$

➢2.21 **稀疏矩阵**：大量元素为 0，非零元很少。以带状矩阵为例：

$$\begin{array}{c} \overbrace{}^{q+1} \\ \left(\begin{array}{cccccc} * & * & * & & & \\ * & * & * & * & & \\ * & * & * & * & * & \\ & * & * & * & * & * \\ & & * & * & * & * \\ & & & * & * & * \end{array}\right) \end{array} \Big\}p+1$$

图 2.4: 上带宽为 q，下带宽为 p 的带状矩阵

$p=q=1$ 时的带状矩阵称为三**对角矩阵**，通常为严格对角占优矩阵。

➢2.22 解三对角系数矩阵线性方程组的**追赶法**：设系数矩阵 $\mathbf{T}$ 为三对角阵，则可用 LU 分解求解方程组：

$$\mathbf{T}=\mathbf{L}\mathbf{U},\mathbf{T}\mathbf{x}=\mathbf{d} \Rightarrow \mathbf{L}\mathbf{U}\mathbf{x}=\mathbf{d} \Rightarrow \begin{cases}\mathbf{L}\mathbf{y}=\mathbf{d} & \text{（追）}\\ \mathbf{U}\mathbf{x}=\mathbf{y} & \text{（赶）}\end{cases}$$

其中 $\mathbf{L}$ 是带宽为 p 的下三角阵，$\mathbf{U}$ 是带宽为 q 的上三角阵[6]。求解前一方程时，从上至下依次带入（「追」）；求解后一方程时，从下至上依次回代（「赶」）。

§2.3 病态问题理论

➢2.23 **误差向量** $\mathbf{e}=\mathbf{x}^*-\tilde{\mathbf{x}}$ 与 **残向量** $\mathbf{r}=\mathbf{b}-\mathbf{A}\tilde{\mathbf{x}}$：前者为里，后者为表。

[5]相较于平方根法少去若干开方运算，故称「改进」。

[6]对于三对角阵，其 LU 分解的结果可用简单的表达式直接写出，参见李乃成、梅立泉《数值分析》第 36 页；但若是针对考试做准备，则可仅按一般的 LU 分解处理三对角矩阵。

➢2.24 如何衡量误差（向量）的大小？可采用向量的**范数**衡量。

☞2.25 **向量范数**：称 $\|\mathbf{x}\|$ 为一个向量的范数，若 $\|\mathbf{x}\| \in \mathbb{R}$ 满足：

1. 非负性：$\forall \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| \geq 0$且$\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$
2. 齐次性：$\forall \alpha \in \mathbb{R}, x \in \mathbb{R}^n, \|\alpha\mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$
3. 三角不等式：$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

➢2.26 常用向量范数：

- 1-范数：$\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$.
- 2-范数：$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$.
- ∞-范数：$\|\mathbf{x}\|_\infty = \max\limits_{1 \leq i \leq n} |x_i|$.

☞2.27 **矩阵范数**：称 $\|\mathbf{A}\| \in \mathbb{R}$ 为一个矩阵范数，若其满足：

1. 非负性：$\forall \mathbf{A}, \|\mathbf{A}\| \geq 0$ 且 $\|\mathbf{A}\| = 0 \Leftrightarrow \mathbf{A} = \mathbf{O}$
2. 齐次性：$\forall \alpha \in \mathbb{R}, \|\alpha\mathbf{A}\| = |\alpha| \cdot \|\mathbf{A}\|$
3. 三角不等式：$\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
4. $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$

☞2.28 向量范数与矩阵范数的**相容性**：若 $\|\mathbf{Ax}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\|$，则称矩阵范数 $\|\mathbf{A}\|$ 与向量范数 $\|\mathbf{x}\|$ 为相容或协调的。

➢2.29 **算子范数**：称 $\|\mathbf{A}\|_p = \max \dfrac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p} = \max\limits_{\|\mathbf{x}\|_p = 1} \|\mathbf{Ax}\|_p$ 为（由向量范数 $\|\mathbf{x}\|_p$ 诱导的）矩阵范数，容易证明 $\|\mathbf{A}\|_p$ 与 $\|\mathbf{x}\|_p$ 满足相容条件。

1. 1-范数：$\|\mathbf{A}\|_1 = \max\limits_{1 \leq j \leq n} \left\{ \sum\limits_{i=1}^{n} |a_{ij}| \right\}$，即列和最大值。
2. 2-范数：$\|\mathbf{A}\|_2 = \sqrt{\mathbf{A}^{\mathrm{T}}\mathbf{A}\text{的最大特征值}}$.
3. ∞-范数：$\|\mathbf{A}\|_\infty = \max\limits_{1 \leq i \leq n} \left\{ \sum\limits_{j=1}^{n} |a_{ij}| \right\}$，即行和最大值。

➢2.30 矩阵的**谱半径**：$\rho(\mathbf{A}) = \max\limits_{1 \leq i \leq n} |\lambda_i|$，性质：$\rho(\mathbf{A}) \leq \|\mathbf{A}\|$

☞2.31 设 $\|\mathbf{B}\| \leq 1$，则 $\mathbf{I} - \mathbf{B}$ 可逆，且有

$$\|(\mathbf{I} - \mathbf{B})^{-1}\| \leq \frac{1}{1 - \|\mathbf{B}\|} \tag{2.7}$$

➢2.32 舍入误差对解的影响：

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} \Rightarrow \mathbf{A}\tilde{\mathbf{x}} = \mathbf{b} - \mathbf{r} \neq \mathbf{b}$$

为分析舍入误差的相对水平，先分析误差向量的大小：

$$\mathbf{e}=\mathbf{A}^{-1}\mathbf{r}\ \Rightarrow\ \|\mathbf{e}\|=\|\mathbf{A}^{-1}\mathbf{r}\|\le\|\mathbf{A}^{-1}\|\cdot\|\mathbf{r}\|$$

由于 $\mathbf{A}\mathbf{x}^{*}=\mathbf{b}$，故

$$\|\mathbf{b}\|=\|\mathbf{A}\mathbf{x}^{*}\|\le\|\mathbf{A}\|\cdot\|\mathbf{x}^{*}\|\ \Rightarrow\ \frac{1}{\|\mathbf{x}^{*}\|}\le\frac{\|\mathbf{A}\|}{\|\mathbf{b}\|}$$

故相对误差水平 $\dfrac{\|\mathbf{x}^{*}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}^{*}\|}\le\|\mathbf{A}\|\cdot\|\mathbf{A}^{-1}\|\cdot\dfrac{\|\mathbf{r}\|}{\|\mathbf{b}\|}$，其中的系数即可定义为矩阵的**条件数** $\mathrm{Cond}(\mathbf{A})=\|\mathbf{A}\|\|\mathbf{A}^{-1}\|$.

➢**2.33** 易知 $\mathrm{Cond}(\mathbf{A})\ge 1$（$\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\ge\|\mathbf{A}\cdot\mathbf{A}^{-1}\|=1$）

➢**2.34** 残向量 $\mathbf{r}$ 不能完全反映偏差水平，因 $\mathbf{r}$ 小，$\mathbf{e}$ 也不一定小。

➢**2.35** 系数扰动对解的影响：设系数矩阵 $\mathbf{A}$ 有扰动 $\Delta\mathbf{A}$，则

$$(\mathbf{A}+\Delta\mathbf{A})\mathbf{x}=\mathbf{b}\ \Rightarrow\ \mathbf{r}=\mathbf{b}-\mathbf{A}\tilde{\mathbf{x}}=\Delta\mathbf{A}\tilde{\mathbf{x}}\ \Rightarrow\ \|\mathbf{r}\|\le\|\Delta\mathbf{A}\|\|\tilde{\mathbf{x}}\|$$

可见若 $\Delta\mathbf{A}$ 小，则 $\mathbf{r}$ 也小。故相对误差水平

$$\frac{\|\mathbf{x}^{*}-\tilde{\mathbf{x}}\|}{\mathbf{x}^{*}}\le\mathrm{Cond}(\mathbf{A})\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}\le\mathrm{Cond}(\mathbf{A})\cdot\frac{\|\tilde{\mathbf{x}}\|}{\|\mathbf{x}^{*}\|}\cdot\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}$$

➢**2.36** 舍入误差与系数扰动的共同影响：假设总的影响可归结为 $(\mathbf{A}-\Delta\mathbf{A})(\mathbf{x}-\Delta\mathbf{x})=\mathbf{b}-\Delta\mathbf{b}$，当 $\|\Delta\mathbf{A}\|\cdot\|\mathbf{A}^{-1}\|<1$ 时，记 $\varepsilon_{\mathbf{A}}=\|\Delta\mathbf{A}\|/\|\mathbf{A}\|$ 与 $\varepsilon_{\mathbf{b}}=\|\Delta\mathbf{b}\|/\|\mathbf{b}\|$ 为各系数在范数意义下的相对偏差，则有

$$\frac{\|\mathbf{x}^{*}-\tilde{\mathbf{x}}\|}{\mathbf{x}^{*}}\le\frac{\|\mathbf{A}^{-1}\|\cdot\|\mathbf{A}\|}{1-\|\mathbf{A}^{-1}\|\cdot\|\mathbf{A}\|\cdot\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}}\left(\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}+\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}\right)=\frac{\varepsilon_{\mathbf{b}}+\varepsilon_{\mathbf{A}}}{[\mathrm{Cond}(\mathbf{A})]^{-1}-\varepsilon_{\mathbf{A}}}\tag{2.8}$$

易见当 $\mathrm{Cond}(\mathbf{A})$ 较大时，方程组为病态方程组；反之，当 $\mathrm{Cond}(\mathbf{A})$ 较小时，方程组仍为良态方程组。

➢**2.37** $\mathrm{Cond}(\mathbf{A})$ 的估计：利用算子范数的相容性，有

$$\mathbf{A}\mathbf{x}=\mathbf{b}\ \Rightarrow\ \mathbf{x}=\mathbf{A}^{-1}\mathbf{b}\ \Rightarrow\ \|\mathbf{x}\|\le\|\mathbf{A}^{-1}\|\cdot\|\mathbf{b}\|\ \Rightarrow\ \frac{\|\mathbf{x}\|}{\|\mathbf{b}\|}\le\|\mathbf{A}^{-1}\|$$

随机选取 p 个向量 $\mathbf{b}_1,\mathbf{b}_2,\cdots,\mathbf{b}_p$，解方程 $\mathbf{A}\mathbf{x}^{(k)}=\mathbf{b}_k$（$1\le k\le p$），得到 $\mathbf{x}^{(k)}$，由上面结论可知

$$\max_{1\le k\le p}\frac{\|\mathbf{x}^{(k)}\|}{\|\mathbf{b}_k\|}\le\|\mathbf{A}^{-1}\|$$

故可近似认为：$\mathrm{Cond}(\mathbf{A})\approx\|\mathbf{A}\|\cdot\max\limits_{1\le k\le p}\dfrac{\|\mathbf{x}^{(k)}\|}{\|\mathbf{b}_k\|}$.

第三章　线性方程组迭代解法

§3.1　迭代方法概要

➢**3.1**　思想：$f(x^*)=0 \Rightarrow \cdots \Rightarrow x^*=\phi(x^*)$，给出初值 x_0 和递推公式 $x_{k+1}=\phi(x_k)$，假设 $\{x_k\}$ 收敛，求极限——设 $\lim\limits_{k\to\infty} x_k=x$，则有 $x=\phi(x)$，从而必有 $f(x)=0$。

☞**3.2**　**向量序列收敛**：$\mathbf{x}^{(k)}=(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^{\mathrm{T}}$，若 $\lim\limits_{k\to\infty} x_i^{(k)}=x_i^*$，则 $\mathbf{x}^{(k)}\to\mathbf{x}^*\ (k\to\infty)$，记作 $\lim\limits_{k\to\infty}\mathbf{x}^{(k)}=\mathbf{x}^*$。

☞**3.3**　**矩阵序列收敛**：$\mathbf{A}^{(k)}=(a_{ij}^{(k)})_{m\times n}$，若 $\lim\limits_{k\to\infty} a_{ij}^{(k)}=a_{ij}$，则称 $\mathbf{A}^{(k)}\to\mathbf{A}=(a_{ij})_{m\times n}$ 记作 $\lim\limits_{k\to\infty}\mathbf{A}^{(k)}=\mathbf{A}$。[(1)]

☞**3.4**　**序列收敛定理**：

- 对向量，$\mathbf{x}^{(k)}\to\mathbf{x}^* \Leftrightarrow \lim\limits_{k\to\infty}\|\mathbf{x}^*-\mathbf{x}^{(k)}\|=0$
- 对矩阵，$\mathbf{A}^{(k)}\to\mathbf{A} \Leftrightarrow \lim\limits_{k\to\infty}\|\mathbf{A}-\mathbf{A}^{(k)}\|=0$

☞**3.5**　定理：设 $\mathbf{B}\in\mathbb{R}^{m\times n}$，则 $\lim\limits_{k\to\infty}\mathbf{B}^k=0 \Leftrightarrow \rho(\mathbf{B})<1$。

➢**3.6**　**迭代格式**的构造：设 $\mathbf{Ax}=\mathbf{b}$，同解变形得 $\mathbf{x}=\mathbf{Bx}+\mathbf{g}$，可构造对应的迭代格式：

$$\mathbf{x}^{(k+1)}=\mathbf{Bx}^{(k)}+\mathbf{g} \tag{3.1}$$

由于是同解变形，故 $\mathbf{x}^*=\lim\limits_{k\to\infty}\mathbf{x}^{(k)}$ 满足 $\mathbf{x}^*=\mathbf{Bx}^*+\mathbf{g} \Rightarrow \mathbf{Ax}^*=\mathbf{b}$，从而可通过 (3.1) 式不断迭代以逼近方程的解。

[(1)]此处不用 $\mathbf{A}^*$，以免与伴随矩阵的符号 $\mathbf{A}^*$ 弄混。

§3.2 三种基本迭代法

➢3.7 **Jacobi 迭代法**：以第 i 行为例，有：

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ii}x_i + \cdots + a_{in}x_n = b_i$$

可解出

$$\begin{aligned} x_i &= \frac{1}{a_{ii}}[b_i - (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{i,i-1}x_{i-1} + a_{i,i+1}x_{i+1} + \cdots + a_{in}x_n)] \\ &= \frac{1}{a_{ii}}\left[b_i - \sum_{j=1,j\neq i}^{n} a_{ij}x_j\right] \end{aligned} \tag{3.2}$$

依上式即可构造 Jacobi 迭代格式：

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1,j\neq i}^{n} a_{ij}x_j^{(k)}\right] \quad (i = 1, 2, \cdots, n) \tag{3.3}$$

按照 $\mathbf{x} = \mathbf{Bx} + \mathbf{g}$ 的标准格式，整理成矩阵格式：

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \cdots & -\frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix} \tag{3.4}$$

➢3.8 **Gauss-Seidel 迭代法**：

$$\begin{cases} x_1^{(k+1)} = \dfrac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)}) \\ x_2^{(k+1)} = \dfrac{1}{a_{22}}(b_2 - a_{21}\boxed{x_1^{(k+1)}} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)}) \\ x_3^{(k+1)} = \dfrac{1}{a_{33}}(b_2 - a_{31}\boxed{x_1^{(k+1)}} - a_{32}\boxed{x_2^{(k+1)}} - \cdots - a_{3n}x_n^{(k)}) \\ \quad \cdots\cdots \\ x_n^{(k+1)} = \dfrac{1}{a_{nn}}(b_2 - a_{n1}\boxed{x_1^{(k+1)}} - a_{n2}\boxed{x_2^{(k+1)}} - \cdots - a_{n,n-1}\boxed{x_{n-1}^{(k+1)}}) \end{cases} \tag{3.5}$$

形式与 Jacobi 法相同，但区别在于进行下一步变量的迭代时采用了「新解」(即上式中框出的部分)。

➢**3.9** **超松弛迭代法**（SOR[(2)] 法）：对 Gauss-Seidel 法的通用格式进行改写；若记每次迭代时的误差为 $\mathbf{r}^{(k+1)}$，即

$$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right] = x_i^{(k)} + \frac{r_i^{(k+1)}}{a_{ii}}$$

当 $k \to \infty$ 时总有 $r_i^{(k+1)}/a_{ii} \to 0$，故可以乘一个系数 ω 以加快收敛：

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right]$$

整理得

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right] \tag{3.6}$$

整体迭代格式：

$$\begin{cases} x_1^{(k+1)} = (1-\omega)x_1^{(k)} + \dfrac{\omega}{a_{11}}\left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_{11}^{(k)}\right) \\ x_2^{(k+1)} = (1-\omega)x_2^{(k)} + \dfrac{\omega}{a_{11}}\left(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_{11}^{(k)}\right) \\ \quad \cdots\cdots \\ x_n^{(k+1)} = (1-\omega)x_n^{(k)} + \dfrac{\omega}{a_{n1}}\left(b_n - a_{n1}x_1^{(k+1)} - \cdots - a_{n,n-1}x_{11}^{(k+1)}\right) \end{cases} \tag{3.7}$$

➢**3.10** 迭代的矩阵表示法：将 $\mathbf{A}$ 分解为 $\mathbf{D},\mathbf{E},\mathbf{F}$ 三部分：$\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F}$，其中

$$\mathbf{D} = \begin{pmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{pmatrix}, \mathbf{E} = \begin{pmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ -a_{n1} & -a_{n2} & -a_{n3} & \cdots & 0 \end{pmatrix}, \mathbf{F} = \begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & 0 & \cdots & -a_{3n} \\ & & & \ddots & \vdots \\ & & & & 0 \end{pmatrix} \tag{3.8}$$

- Jacobi 迭代法：迭代格式中除对角阵 $\mathbf{D}$ 以外的系数均挪到了方程右侧，相当于

$$(\mathbf{D}-\mathbf{E}-\mathbf{F})\mathbf{x} = \mathbf{b} \;\Rightarrow\; \mathbf{D}\mathbf{x} = (\mathbf{E}+\mathbf{F})\mathbf{x} + \mathbf{b} \;\Rightarrow\; \mathbf{x} = \mathbf{D}^{-1}(\mathbf{E}+\mathbf{F})\mathbf{x} + \mathbf{D}^{-1}\mathbf{b}$$

可见有

$$\begin{cases} \mathbf{B} = \mathbf{D}^{-1}(\mathbf{E}+\mathbf{F}) = \mathbf{D}^{-1}(\mathbf{D}-\mathbf{A}) = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A} \\ \mathbf{g} = \mathbf{D}^{-1}\mathbf{b} \end{cases} \tag{3.9}$$

[(2)] Successive over-relaxation.

- Gauss-Seidel 迭代法：与 Jacobi 法不同的是，迭代中表示预先求得之「新解」的系数阵 $\mathbf{E}$ 留在方程左侧，相当于

$$(\mathbf{D}-\mathbf{E}-\mathbf{F})\mathbf{x}=\mathbf{b}\ \Rightarrow\ (\mathbf{D}-\mathbf{E})\mathbf{x}=\mathbf{F}\mathbf{x}+\mathbf{b}$$
$$\Rightarrow\ \mathbf{x}=(\mathbf{D}-\mathbf{E})^{-1}\mathbf{F}\mathbf{x}+(\mathbf{D}-\mathbf{E})^{-1}\mathbf{b}$$

可见有

$$\begin{cases}\mathbf{B}=(\mathbf{D}-\mathbf{E})^{-1}\mathbf{F}\\ \mathbf{g}=(\mathbf{D}-\mathbf{E})^{-1}\mathbf{b}\end{cases}\tag{3.10}$$

- 超松弛迭代法：在 Gauss-Seidel 法的基础上，将 $(1-\omega)$ 大小的对角系数 $\mathbf{D}$ 移至方程右侧（作为误差），相当于

$$\omega(\mathbf{D}-\mathbf{E}-\mathbf{F})\mathbf{x}=\omega\mathbf{b}\ \Rightarrow\ (\mathbf{D}-\omega\mathbf{E})\mathbf{x}=[(1-\omega)\mathbf{D}+\omega\mathbf{F}]\mathbf{x}+\omega\mathbf{b}$$
$$\Rightarrow\ \mathbf{x}=(\mathbf{D}-\omega\mathbf{E})^{-1}[(1-\omega)\mathbf{D}+\omega\mathbf{F}]\mathbf{x}+(\mathbf{D}-\omega\mathbf{E})^{-1}\omega\mathbf{b}$$

可见有

$$\begin{cases}\mathbf{B}=(\mathbf{D}-\omega\mathbf{R})^{-1}[(1-\omega)\mathbf{D}+\omega\mathbf{F}]\\ \mathbf{g}=\omega(\mathbf{D}-\omega\mathbf{E})^{-1}\mathbf{b}\end{cases}\tag{3.11}$$

§3.3 迭代收敛理论

➢**3.11** 下面给出迭代格式收敛的条件（非常重要！个人猜测必考[(3)]）。

☞**3.12** 定理 1：若 $\|\mathbf{B}\|\le 1$，则 $\forall\mathbf{x}^{(0)}$，迭代格式 $\mathbf{x}^{(k+1)}=\mathbf{B}\mathbf{x}^{(k)}+\mathbf{g}$ 收敛于解 $\mathbf{x}^*$，且有误差估计式：

$$\|\mathbf{x}^*-\mathbf{x}^{(k)}\|\le\frac{\|\mathbf{B}\|}{1-\|\mathbf{B}\|}\|\mathbf{x}^{(k)}-\mathbf{x}^{(k-1)}\|\quad\text{（事后估计或后验估计）}\tag{3.12}$$

$$\|\mathbf{x}^*-\mathbf{x}^{(k)}\|\le\frac{\|\mathbf{B}\|^k}{1-\|\mathbf{B}\|}\|\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\|\quad\text{（事前估计或先验估计）}\tag{3.13}$$

➢**3.13** 估计式的推导：利用迭代格式及 $\mathbf{x}^*=\mathbf{B}\mathbf{x}^*+\mathbf{g}$ 条件将 $\mathbf{x}^*-\mathbf{x}^{(k)}$ 展开并变形，最终在等式两侧取范数，应用范数若干性质和条目 2.31 的结果获得不等号。

☞**3.14** 定理 2：$\forall\mathbf{x}^{(0)}$，迭代格式 $\mathbf{x}^{(k+1)}=\mathbf{B}\mathbf{x}^{(k)}+\mathbf{g}$ 收敛于解 $\mathbf{x}^*$ 的充要条件是下列两条件之一成立：

[(3)] 王天浩的评论。

1. $\mathbf{B}^k \to \mathbf{O}$;
2. $\mathbf{B}$ 的谱半径 $\rho(\mathbf{B}) < 1$。

☞3.15 推论：SOR 法收敛的必要条件是 $0 < \omega < 2$。

$$\|\mathbf{B}\| = \|(\mathbf{D} - \omega\mathbf{E})^{-1}\| \cdot \|(1-\omega)\mathbf{D} + \omega\mathbf{F}\| = \frac{\|(1-\omega)\mathbf{D}\|}{\|\mathbf{D}\|} = (1-\omega)^n$$

$$\rho(\mathbf{B}) < 1 \;\Rightarrow\; |\lambda_1 \cdots \lambda_n| < 1 \;\Rightarrow\; |1-\omega| < 1 \;\Rightarrow\; 0 < \omega < 2.$$

☞3.16 对于三种常用迭代法，有更为实用的结论：

- 引理：若 $\mathbf{A}$ 是严格对角占优矩阵，$0 \le \omega \le 1$ 且 $\lambda \ge 1$ 时，矩阵 $(\lambda + \omega - 1)\mathbf{D} - \lambda\omega\mathbf{E} - \omega\mathbf{F}$ 也是严格对角占优矩阵。
- 推论 2：若 $\mathbf{A}$ 是严格对角占优矩阵，则 $\forall \mathbf{x}^{(0)}$，Jacobi 法、G-S 法、SOR 法（$0 < \omega \le 1$）均收敛 [4]。
- 推论 3：若 $\mathbf{A}$ 为对称正定阵，则 $\forall \mathbf{x}^{(0)}$，Jacobi 法收敛的充要条件是：$2\mathbf{D} - \mathbf{A}$ 也是对称正定阵。
- 推论 4：若 $\mathbf{A}$ 是对称正定阵，则 $\forall \mathbf{x}^{(0)}$，SOR 法收敛充要条件为 $0 < \omega < 2$。

➢3.17 对不同情况下的审敛法做总结：

1. 对角占优矩阵 $\mathbf{A}$：Jacobi 法收敛，G-S 法收敛，$0 < \omega \le 1$ 时 SOR 法收敛。
2. 对称正定阵 $\mathbf{A}$：
   - Jacobi 法收敛 $\Leftrightarrow$ $2\mathbf{D} - \mathbf{A}$ 收敛；
   - SOR 法收敛 $\Leftrightarrow$ $0 < \omega < 2$。
3. 一般矩阵 $\mathbf{A}$：
   - $\|\mathbf{B}\| < 1$ 时，$\mathbf{B}$ 的对应方法收敛；
   - $\mathbf{B}^k \to \mathbf{O}$ 和 $\rho(B) < 1$ 中之一成立，则 $\mathbf{B}$ 的对应方法收敛。
   - SOR 法收敛的必要条件是 $0 < \omega < 2$。

➢3.18 针对考试而言，一般的系数矩阵仍需通过求 $\rho(\mathbf{B})$ 或 $\|\mathbf{B}\|$ 来判断收敛性。

✍3.19 判断系数矩阵为 $\mathbf{A} = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 6 & 10 \\ 4 & 10 & 20 \end{pmatrix}$ 时各迭代法的收敛性。（答案：Jacobi 法发散，Gauss-Seidel 迭代法及 $0 < \omega < 2$ 的 SOR 法收敛。）

[4] 推导：反设 $\mathbf{B}$ 有一特征值 $|\lambda| \ge 1$，代入上面的推论中，推出 $|\lambda\mathbf{I} - \mathbf{B}| \ne 0$，故所有的 $|\lambda| < 1$，进而 $\rho(\mathbf{B}) < 1$。参见李乃成、梅立泉《数值分析》第 88 页推论 3.3.2 的证明。

第四章　插值法

§4.1　插值法思想概要

➢**4.1**　插值的动机：

1. 离散型：给定 $m+1$ 个数据点 (x_0, y_0)、(x_1, y_1)、……(x_m, y_m)，找到一个函数 $P(x)$ 通过所有的点。
2. 连续型：给定一未知函数 $f(x)$ 及其 $m+1$ 个数据点，要求另一已知函数 $P(x)$ 在这些数据点上与 $f(x)$ 一致，并使其在定义与上与 $f(x)$ 的偏差尽量小。（称 $f(x)$ 为**被插函数**。）

☞**4.2**　设以上的 $P(x)$ 满足

$$P(x_i) = y_i \quad (i = 0, 1, \cdots, m), \tag{4.1}$$

则称 $P(x)$ 为这 $m+1$ 个数据点上的**插值函数**，称数据点为**插值点**，并称式 (4.1) 为**插值条件**。

➢**4.3**　多项式插值：要求插值函数 $P(x)$ 为一个多项式

$$P(x) = \sum_{k=0}^{n} a_k x^k = a_0 + a_1 x + \cdots + a_n x^n, \tag{4.2}$$

则称此时的 $P(x)$ 为一个**插值多项式**，对应的插值条件

$$P(x_i) = y_i \;\Rightarrow\; a_0 + a_1 x_i + \cdots + a_i x_i^n = y_i \quad (i = 0, 1, \cdots, m) \tag{4.3}$$

可视为一个线性方程组：

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}, \tag{4.4}$$

➢**4.4**　关于方程组 (4.4) 的解分析：

- $m > n$，方程一般无解；
- $m < n$，方程有无穷多解；
- $m = n$ 时，系数矩阵对应的行列式是 Van der monde 行列式：

$$|V| = \prod_{j=1}^{n}\left[\prod_{i=0}^{j-1}(x_j - x_i)\right] \tag{4.5}$$

当 $x_i \neq x_j\ (i \neq j)$ 时，$|V| \neq 0$，方程组有唯一解。

☞**4.5** 对于给定的 $n+1$ 个插值点，对应的 n 次插值多项式 $P_n(x)$ 唯一存在。

➢**4.6** **误差多项式**：若被插函数 $f(x)$ 满足 $f^{(n)}(x)$ 在 $[a,b]$ 上连续，$f^{(n+1)}$ 在 (a,b) 内存在，则误差（多项式）可估计为

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n). \tag{4.6}$$

记

$$\pi_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n), \tag{4.7}$$

则有

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x). \tag{4.8}$$

➢**4.7** 实用误差估计式：设 $P_n(x)$ 为在 $(x_0, x_1, \cdots, x_n)$ 上的插值多项式，$P_n^*(x)$ 为在 $(x_0, x_1, \cdots, x_n, x_{n+1})$ 上的插值多项式，则

$$f(x) - P_n(x) \approx \frac{P_n^*(x) - P_n(x)}{x_{n+1} - x_0}(x - x_0) \tag{4.9}$$

$$f(x) - P_n^*(x) \approx \frac{P_n^*(x) - P_n(x)}{x_{n+1} - x_0}(x - x_{n+1}) \tag{4.10}$$

➢**4.8** 采用式 (4.4) 求解插值多项式的问题：方程条件数随 n 的增加而急剧上升，解不稳定、不精确；计算量太大。

§4.2 Lagrange 插值法与 Newton 插值法

➢**4.9** **Lagrange 插值法**：对 $n+1$ 个数据点，构造对应的 $n+1$ 个 **Lagrange 插值基函数** $l_0(x), l_1(x), \cdots, l_n(x)$ 使

$$P_n(x) = y_0 \cdot l_0(x) + y_1 \cdot l_1(x) + \cdots + y_n \cdot l_n(x). \tag{4.11}$$

➢**4.10** 插值基函数公式：由式 (4.11) 易知，插值基函数应在数据点上满足

$$l_i(x_j)=\delta_{ij}=\begin{cases}1 & ,i=j\\ 0 & ,i\neq j\end{cases} \tag{4.12}$$

又要求插值基函数为 n 次多项式（与最终插值多项式的次数一致），则可解出

$$l_i(x)=\frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}. \tag{4.13}$$

此即 Lagrange 插值基函数公式。其分母 $\pi_{n+1}(x)/(x-x_i)$ 可直接写出，故只需计算插值基函数的系数

$$c_i=[(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)]^{-1}. \tag{4.14}$$

✍**4.11** 给定数据点 $(-1,7)$、$(1,7)$、$(2,4)$、$(5,35)$，用 Lagrange 插值法求解各插值基函数的系数。（答案：$c_0=-\frac{1}{36}$，$c_1=\frac{1}{8}$，$c_2=-\frac{1}{9}$，$c_3=\frac{1}{72}$。）

➢**4.12** 常将 Lagrange 基函数记为

$$l_i(x)=\frac{\pi_{n+1}(x)}{(x-x_i)\pi'_{n+1}(x_i)}. \tag{4.15}$$

➢**4.13** Lagrange 插值法的特点：

- 构造方便、格式统一；
- 系数的表示方法简单，但乘除运算量大；
- 插值基函数具有全局性质，「牵一发而动全身」，数据点变动后须全部重新计算。

➢**4.14** **Newton 插值法**：对 $(n+1)$ 个数据点，采用「累进」的插值基函数

$$n_i(x)=\prod_{k=0}^{i-1}(x-x_k)=(x-x_0)(x-x_1)\cdots(x-x_{i-1}), \tag{4.16}$$

构造得到的 **Newton 插值多项式**为

$$N_n(x)=c_0+c_1(x-x_0)+c_2(x-x_0)(x-x_1)+\cdots+c_n(x-x_0)\cdots(x-x_n) \tag{4.17}$$

再求解系数 c_i。

➢**4.15** **差商**：导数的一种离散形式，递归定义：

- 零阶差商：$f[x_i]=f(x_i)=y_i \quad (i=0,1,\cdots,n)$.
- 一阶差商：$f[x_i,x_{i+1}]=\frac{f[x_{i+1}]-f[x_i]}{x_{i+1}-x_i} \quad (i=0,1,\cdots,n-1)$.
- 二阶差商：$f[x_i,x_{i+1},x_{i+2}]=\frac{f[x_i,x_{i+2}]-f[x_i,x_{i+1}]}{x_{i+2}-x_{i+1}} \quad (i=0,1,\cdots,n-2)$.
- ……
- n 阶差商：$f[x_0,x_1,\cdots,x_n]=\frac{f[x_0,x_1,\cdots,x_{n-2},x_n]-f[x_0,x_1,\cdots,x_{n-2},x_{n-1}]}{x_n-x_{n-1}}$.

➢**4.16** 易证式 (4.17) 中系数满足 $c_0=f[x_0]$，$c_1=f[x_0,x_1]$，$c_2=f[x_0,x_1,x_2]$ ……从而 Newton 插值多项式为

$$N_n(x)=\sum_{k=0}^{n} f[x_0,\cdots,x_k]\cdot n_k(x). \tag{4.18}$$

➢**4.17** 由插值多项式的唯一性，对同样的 $n+1$ 个数据点构造的 Lagrange 插值多项式 $P_n(x)$ 与 Newton 插值多项式 $N_n(x)$，必有 $P_n(x)=N_n(x)$，即两种方法得到的结果相同。

☞**4.18** 推论：比较 Newton 插值多项式与 Lagrange 插值多项式的最高次（n 次）项系数，有

$$f[x_0,x_1,\dots,x_n]=\sum_{i=0}^{n}\frac{f(x_i)}{\pi'_{n+1}(x_i)}. \tag{4.19}$$

☞**4.19** 差商的性质：

1. 差商仅与选取的具体点 $(x_i,f(x_i))$ 有关，与它们的排列次序无关。
2. $f[x_i,x_{i+1},\dots,x_{i+k}]=f^{(k)}(\xi)/(k!)$，其中 $\xi\in(\min\{x_i\},\max\{x_i\})$。
3. 在以上结论中取 $x_i=x_{i+1}=\dots=x_{i+k}$，得

$$f[x_i,x_i,\dots,x_i]=\frac{f^{(k)}(x_i)}{k!}. \tag{4.20}$$

4. $\frac{\mathrm{d}f[x_0,x_1,\dots,x_{k-1},x]}{\mathrm{d}x}=\frac{f^{(k+1)}(\xi)}{(k+1)!}$。

✍**4.20** 设 $f(x)=x^3+2px+5qx+c$，其中 p,q,c 均为实数。若 $f[1,2,m]=0$，试求 $f[0,1,m]$。（答案：-2）

➢**4.21** **差商表**：依次计算差商的工具，如图 4.1 所示。

$$P_3(x) = y_0 + y_0' \cdot (x - x_0) + y_0'' \cdot (x - x_0)(x - x_1) + y_0''' \cdot (x - x_0)(x - x_1)(x - x_2)$$

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
x_0	$\boxed{y_0}$			
x_1	y_1	$\boxed{y_0' = \frac{y_1 - y_0}{x_1 - x_0}}$		
x_2	y_2	$y_1' = \frac{y_2 - y_1}{x_2 - x_1}$	$\boxed{y_0'' = \frac{y_1' - y_0'}{x_2 - x_0}}$	
x_3	y_3	$y_2' = \frac{y_3 - y_2}{x_3 - x_2}$	$y_1'' = \frac{y_2' - y_1'}{x_3 - x_1}$	$\boxed{y_0''' = \frac{y_1'' - y_0''}{x_3 - x_0}}$

图 4.1: 差商表及其构造步骤

➢**4.22** **Newton 插值法余项公式**及估计：

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

即：Newton 插值公式 $N_n(x)$ 的余项，可估计为高一阶的插值多项式 $N_{n+1}(x)$ 之最后一项。

➢**4.23** **Hermite 插值多项式**：满足导数条件 $P'(x_j) = y_j'$ 的插值多项式。

➢**4.24** Newton 插值法构造 Hermite 插值多项式（**重节点法**）：在含导数条件的数据点处增加「重节点」，仍按差商表迭代，但利用条件 (4.20) 计算含重节点的差商。

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
x_0	y_0			
x_1	y_1	$y_0' = \frac{y_1 - y_0}{x_1 - x_0}$		
x_1	y_1	$\boxed{y_1'}$	$y_0'' = \frac{y_1' - y_0'}{x_1 - x_0}$	
x_2	y_2	$y_2' = \frac{y_2 - y_1}{x_2 - x_1}$	$y_1'' = \frac{y_2' - y_1'}{x_2 - x_1}$	$y_0''' = \frac{y_1'' - y_0''}{x_2 - x_0}$

图 4.2: 重节点法示意

➢**4.25** Hermite 插值多项式的误差估计：利用 Newton 插值的余项估计即可。

✎**4.26** 对以下的数据点求解其 Hermite 插值多项式，并估计误差。

x_i	-1	0	1
y_i	0	-4	5
y_i'		0	5
y_i''	6		

（答案：$H_5(x)=x^5-2x^3+3x^2-4$，$R_5(x)=\frac{f^{(6)}(\xi)}{6!}(x+1)(x-1)^2x^3$。）

➢**4.27** Lagrange 插值法构造 Hermite 插值多项式：对 $n+1$ 个含导数条件的数据点，在导数条件下构造插值基函数 $h_i(x)$ 与 $\overline{h}_i(x)$：

$$H_{2n+1}(x)=\sum_{i=0}^{n}h_i(x)f(x_i)+\sum_{i=0}^{n}\overline{h}_i(x)f'(x_i) \tag{4.21}$$

$$h_i(x_j)=\delta_{ij},\quad h_i'(x_j)=0\quad \overline{h}_i(x_j)=0\quad \overline{h'}_i(x_j)=\delta_{ij}. \tag{4.22}$$

分析零点重数可推得

$$h_i(x)=(ax+b)l_i^2(x),\quad \overline{h}_i(x)=(x-x_i)l_i^2(x) \tag{4.23}$$

再求解系数 a 与 b 即可。

➢**4.28** 不建议使用 Lagrange 插值法求解 Hermite 插值多项式：优势尽失。

§4.3 分段插值与三次样条插值

➢**4.29** **Runge 现象**：采用高次的插值多项式，全局误差可能比低次多项式更大。（示例：对函数 $f(x)=\frac{1}{1+25x^2}$ $(-1\le x\le 1)$ 以越来越多的节点等距插值。）这表明，与其在全局应用高次插值多项式，不如采用分段低次插值多项式。

➢**4.30** **分段线性插值**的误差估计：

$$R_{1,j}(x)=\left|\frac{f''(\xi)}{2}(x-x_{i-1})(x-x_i)\right|\le\frac{M_2 i}{2}\cdot\frac{1}{4}(x_i-x_{i-1})^2\le\frac{1}{8}M_2\Delta^2 \tag{4.24}$$

其中 M_2 为 $f''(x)$ 在插值区间上的最大值，Δ 为各相邻插值节点的最大距离。

➢**4.31** 分段线性插值的缺陷：在各个节点处导数不连续。

➢**4.32** **分段二次插值**：在 $[x_{i-1},x_{i+1}]$ 上作 Newton 二次插值多项式 $N_{2i}(x)$，则

$$f(x)\approx N_{2i}(x)=y_{i-1}+f[x_{i-1},x_i](x-x_{i-1})+f[x_{i-1},x_i,x_{i+1}](x-x_{i-1})(x-x_i). \tag{4.25}$$

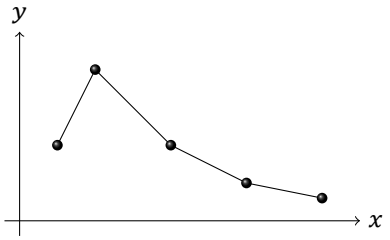


图 4.3: 分段线性插值图示

➢**4.33** 分段二次插值的误差估计：

$$R_{2i}(x)=\frac{f'''(\xi)}{3!}(x-x_{i-1})(x-x_i)(x-x_{i+1})\le\frac{M_3}{6}\cdot\frac{1}{4}\Delta^2\cdot 2\Delta=\frac{1}{12}M_3\Delta^3 \quad (4.26)$$

其中 M_3 为 $f'''(x)$ 在插值区间上的最大值，Δ 为各相邻插值节点的最大距离。

➢**4.34** 分段二次插值的缺陷：在一半的节点上导数仍不连续；要求有奇数个插值节点。

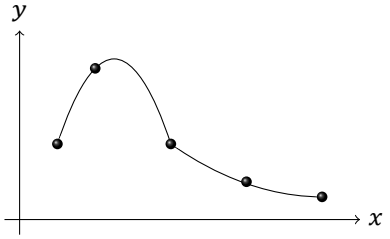


图 4.4: 分段二次插值图示

➢**4.35** **分段三次 Hermite 插值**：在相邻插值节点处利用两个函数值、两个导数值构造插值多项式。

➢**4.36** 分段三次 Hermite 插值的误差估计：

$$R_{3,i}(x)=\frac{f^{(4)}(\xi)}{4!}(x-x_{i-1})^2(x-x_i)^2\le\frac{M_4}{24}\left[\frac{1}{4}(x-x_{i-1})^2\right]^2\le\frac{M_4}{384}\Delta^4. \quad (4.27)$$

➢**4.37** 分段三次 Hermite 插值的缺陷：二阶导数仍不连续；导数条件太苛刻，插值时一般缺少相应导数。

➢**4.38** **三次样条插值**：给定 $n+1$ 个数据点，要求分段插值曲线的二阶导数连续。由此可知，插值函数及其导数也应连续。此时：

- 需求：在 n 个插值子区间上构造分段三次插值函数，共计 $4n$ 个未知数；

- 约束条件：$n+1$ 个数据点，$3(n-1)$ 个各阶导数连续条件，共计 $4n-2$ 个方程；
- 补充 2 个边界条件：一阶导数边界条件 $f'(a)$、$f'(b)$，或二阶导数边界条件 $f'(a)$、$f'(b)$，或周期性边界条件 $f'(a)=f'(b)$、$f''(a)=f''(b)$。

由此即可求解出所有的分段三次插值函数，称这些函数为三**次样条函数**。

➢**4.39** **三弯矩方程**：设 $S(x)$ 在 x_i 处二阶导数值为 M_i，根据插值函数为三次可知 $S''(x)$ 为连续的分段线性函数，对 $S''(x)$ 积两次分并用 n 个数据点条件消去未知参数得：

$$\begin{aligned} S(x)=&\frac{(x_i-x)^3}{6h_i}M_{i-1}+\frac{(x-x_{i-1})^3}{6h_i}M_i+\left(y_{i-1}-\frac{h_i^2}{6}M_{i-1}\right)\frac{x_i-x}{h_i}\\ &+\left(y_i-\frac{h_i^2}{6}M_i\right)\frac{x-x_{i-1}}{h_i},\ x_{i-1}\le x\le x_i \end{aligned} \tag{4.28}$$

为求出 M_i，可对 $S(x)$ 在不同区间上的表达式求导，并应用 $(n-1)$ 个一阶导数连续条件获得 $(n-1)$ 个方程：

$$\mu_i M_{i-1}+2M_i+\lambda M_{i+1}=d_i,\quad i=1,2,\cdots,n-1 \tag{4.29}$$

其中

$$\mu_i=\frac{h_i}{h_i+h_{i+1}},\quad \lambda_i=\frac{h_{i+1}}{h_i+h_{i+1}}=1-\mu_i,\quad d_i=6f[x_{i-1},x_i,x_{i+1}]$$

若将 $S(x)$ 视为一根梁的挠度，则以上的 M_i 可视为作用在 x_i 处的弯矩，故称方程 (4.29) 为三弯矩方程。

➢**4.40** 三种边界条件下的三弯矩方程：三弯矩方程不足以解出 $n+1$ 个未知量，补充两个边界条件后方程即可封闭。

- 一阶导数边条：对式 (4.28) 求一次导，并代入边界导数 $f'(a)$ 与 $f'(b)$，可最终获得两个补充方程：

  $$2M_0+M_1=d_0,\quad M_{n-1}+2M_n=d_n \tag{4.30}$$

  其中 d_0 与 d_n 与其他 d_i 的定义一致。方程组变为 $n+1$ 个式子的

  $$\begin{cases} 2M_0+M_1=d_0\\ \mu_i M_{i-1}+2M_i+\lambda_i M_{i+1}=d_i,\quad i=1,2,\dots,n-1\\ M_{n-1}+2M_n=d_n \end{cases} \tag{4.31}$$

  此时的系数矩阵是严格三对角占优矩阵（$\mu_i+\lambda_i=1<2$），可用追赶法快速求解。

- 二阶导数边条：$M_0 = f''(a)$ 和 $M_n = f''(b)$ 给定，方程组变为 $n-1$ 个式子的

$$\begin{cases} 2M_1 + \lambda_1 M_2 = d_1 - \mu_1 M_0 \\ \mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i, \quad i = 2, 3, \dots, n-2 \\ \mu_{n-1} M_{n-2} + 2M_{n-1} = d_{n-1} - \lambda_{n-1} M_n \end{cases} \tag{4.32}$$

  此时系数矩阵也是严格三对角占优矩阵。

- 周期性边条：根据周期性 [1]，从边界点处「回环」得满足三弯矩方程的序列 (M_{n-1}, M_n, M_1) 与 (M_n, M_1, M_2)，由此补充两个方程，方程组变为 n 个式子的：

$$\begin{cases} 2M_1 + \lambda_1 M_2 + \mu_1 M_n = d_1 \\ \mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i, \quad i = 2, 3, \dots, n-1 \\ \lambda_n M_1 + \mu_n M_{n-1} + 2M_n = d_n \end{cases} \tag{4.33}$$

  此时的系数矩阵为严格对角占优矩阵（不是三对角），可用一般的 LU 分解求解 [2]。

➢**4.41** 三次样条插值的特点：

√ 求解的三弯矩方程为严格对角占优矩阵，方程形式简洁且容易求解，解存在唯一、稳定性好；

√ 为提高精度，只需增加插值节点，不需要提高次数；

√ 随节点增多，$S(x)$ 及其一二阶导数一致收敛于 $f(x)$ 及其对应导数；若节点间等距，则 $S'''(x)$ 随节点加密而一致收敛于 $f'''(x)$。

× 三弯矩方程计算量仍很大；

× 需要额外的边界条件，有时难以获得；

× 对图形的控制不够灵活，绘图上使用并不方便 [3]。

---

[1] 此时 $M_0 = M_n$，同时少去一个未知数和一个方程，仍需补充两个方程。

[2] 在 LU 分解的过程中，会发现此形式的系数矩阵有与追赶法类似的简化算法，参见李乃成、梅立泉《数值分析》习题 2.4。

[3] 绘图上常用 Bézier 曲线、B 样条曲线等专用于拟合几何边界的插值/曲线构造方式。

第五章　函数最优逼近

➢**5.1** 逼近与插值的区别：插值要求通过数据点，逼近则不要求；逼近追求在给定的函数形式下整个区间或所有值点上的总体误差最小。

➢**5.2** 用多项式逼近：便于计算；便于用其做微积分。

§5.1　内积、范数与正交多项式

☞**5.3** **函数在点集上的内积**：设 $f(x)$，$g(x)$ 在 $X=\{x_1,x_2,\dots,x_m\}$ 上有定义，并有相应的 m 个权系数 ω_i，则定义

$$(f,g)=\sum_{i=1}^{m}\omega_i f(x_i)g(x_i)$$

为 f 与 g 在 X 上关于权系数 ω_i 的内积。

☞**5.4** **函数在区间上的内积**：设 $f,g\in C[a,b]$，$\omega(x)\in C[a,b]$ 为权系数，定义

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

为 f 与 g 在 $[a,b]$ 上关于 $\omega(x)$ 的内积。

➢**5.5** 对权系数的要求：$\omega_i>0$ 或 $\omega(x)\geq 0$。常取 $\omega_i=1$ 及 $\omega(x)\equiv 1$。

☞**5.6** 内积的性质[(1)]：

1. 对称性：$(f,g)=(g,f)$。
2. 齐性：对任意的常数 α，$(\alpha f,g)=(f,\alpha g)=\alpha\cdot(f,g)$。
3. 可加性：$(f+h,g)=(f,g)+(h,g)$。
4. 正定性：$(f,f)\geq 0$，且仅当 $f\equiv 0$ 时 $(f,f)=0$。

[(1)] 该四条性质对一切「内积空间」都是成立的。

☞5.7 **函数在点集或区间上的范数**：对点集 X 上所有函数 f，或对区间上所有连续函数 $f \in C[a,b]$ 定义运算 $\|f\|$，满足：

1.（正定性）$\|f\| \geq 0$，且仅当 $f \equiv 0$ 时 $\|f\| = 0$。
2.（线性）对任意的常数 α，$\|\alpha f\| = |\alpha| \cdot \|f\|$。
3.（三角不等式）$\|f + g\| \leq \|f\| + \|g\|$。

➢5.8 常用范数：

- 由函数内积导出的 2-范数：$\|f(x)\|_2 = \sqrt{(f,f)}$。
- 1-范数：$\|f\|_1 = \sum_{i=1}^{m} |f(x_i)|$ 或 $\|f\|_1 = \int_a^b |f(x)|\mathrm{d}x$。
- ∞-范数：$\|f\|_\infty = \max_{1\leq i\leq m} |f(x_i)|$ 或 $\|f\|_\infty = \max_{a\leq x\leq b} |f(x)|$。

➢5.9 函数的**正交**：

- 在权函数 ω_i（或 $\omega(x)$）下，若 $(f,g) = 0$，则称两函数关于权函数 ω_i（或 $\omega(x)$）正交，称 f 与 g 为**正交函数**。
- 设有函数族 $\{g_k\}$，若其中各函数 $g_0(x), g_1(x), \cdots, g_k(x), \cdots$ 满足 $(g_i, g_j) = 0\ (i \neq j)$，则称 $\{g_k\}$ 为关于 ω **正交**的函数族。
- 若正交函数族 $\{g_k\}$ 进一步满足 $\|g_i(x)\| = 1\ (\forall i)$，则称其为一个 **标准正交函数族**。
- 若正交函数族 $\{g_k\}$ 中的 $g_k(x)$ 为 k 次多项式，则称 $g_0(x), \cdots, g_k(x), \cdots$ 为 **正交多项式**。

☞5.10 正交多项式基本性质：各正交多项式之间线性无关 [2]。（由此，可用一族正交多项式线性表示各阶多项式。）

☞5.11 推论 1：对 $k < n$，k 次多项式 $P_k(x)$ 与 n 次正交多项式 $g_n(x)$ 正交 [3]。

☞5.12 推论 2：在区间 $[a,b]$（连续）或 $[\min x_i, \max x_i]$（离散）上，n 次正交多项式 $g_n(x)$ 恰有 n 个不同实零点 [4]。

☞5.13 推论 3：设 $g_0(x), g_1(x), \cdots, g_k(x), \cdots$ 均为首一[5]正交多项式，则

$$g_0(x) = 1,\ g_1(x) = x - b_0,\ g_{k+1}(x) = (x - b_x)g_k(x) - c_k g_{k-1}(x) \qquad (5.1)$$

$$b_k = \frac{\beta_k}{\gamma_k}, c_k = \frac{\gamma_k}{\gamma_{k-1}}, \beta_k = (xg_k, g_k), \gamma_k = (g_k, g_k).$$

[2]证明：利用正交性与正定性。

[3]证明：$P_k(x)$ 可用 $g_0(x), \cdots, g_k(x)$ 线性表出，进而由 $g_k(x)$ 的正交性推得其与 $g_n(x)$ 正交。

[4]证明思路：有实根 ⇒ 无偶重根 ⇒ 均为实根 ⇒ 无多于 1 重的奇实根。

[5]即最高次项系数为 1。

称以上公式为正交多项式的三**项递推关系**[6]。

➢**5.14** **Legendre 多项式**：在区间 $[-1,1]$ 上，定义为

$$P_k(x)=\frac{1}{2^k\cdot k!}\frac{\mathrm{d}^k}{\mathrm{d}x^k}\left[(x^2-1)^k\right].\quad(k=0,1,2,\ldots)\tag{5.2}$$

其关于权函数 $\omega(x)\equiv 1$ 正交。最高次项系数为 $\alpha_k=\frac{(2k)!}{2^k\cdot(k!)}$。

➢**5.15** 其他常见正交多项式：

- **Laguerre 多项式**：$L_k(x)=\mathrm{e}^x\frac{\mathrm{d}(x^k\mathrm{e}^{-x})}{\mathrm{d}x^k}$，正交区间为 $[0,+\infty)$，权函数为 $\omega(x)=\mathrm{e}^{-x}$。
- **Hermite 多项式**：$H_k(x)=(-1)^k\mathrm{e}^{x^2}\frac{\mathrm{d}^k\mathrm{e}^{-x^2}}{\mathrm{d}x^k}$，正交区间为 $(-\infty,+\infty)$，权函数为 $\omega(x)=\mathrm{e}^{-x^2}$。
- **Cheybyshev 多项式**：$T_k(x)=\cos(k\arccos x)$，正交区间为 $[-1,1]$，权函数为 $\omega(x)=\frac{1}{\sqrt{1-x^2}}$。

§5.2 最优平方逼近

➢**5.16** 最优平方逼近：对函数 $f(x)$，构造逼近多项式 $p(x)$，使按 2-范数度量得到的误差之平方 $S=\|p-f\|_2^2$ 最小[7]。

➢**5.17** 为系统地实现这一构造过程，常用一多项式函数族 $\{\varphi_k(x)\}$ 表出 $p(x)$：

$$p(x)=c_0\varphi_0(x)+\cdots+c_n\varphi_n(x),\tag{5.3}$$

从而问题需落实为：

1. 多项式族 $\{\varphi_k(x)\}$ 的选取或导出；
2. 系数 c_k 的求解。

☞**5.18** 若 f 为列表函数，则称逼近其的 $p(x)$ 为**最小二乘拟合多项式**，误差表达式为

$$\|p-f\|_2=\sqrt{\sum_{i=1}^{m}\omega_i\Big[p(x_i)-y_i\Big]^2}.$$

[6]证明略去，参见李乃成、梅立泉《数值分析》第 146 页性质 5.1.3 的证明。会用即可。

[7]取平方是因为 2-范数中的根号不易处理。更常见的说法是「均方误差（MSE）最小」，此概念在统计学、机器学习等领域更为常见。

☞5.19 若 f 为连续函数，则称逼近其的 $p(x)$ 为**最优平方逼近多项式**，误差为

$$\|p-f\|_2=\sqrt{\int_a^b (p-f)^2\mathrm{d}x}.$$

➢5.20 **正规方程组**：按定义将 $S=\|p-f\|_2^2$ 拆开：

$$S=(p-f,p-f)=(p,p)-2(p,f)+(f,f)$$

$$S=\sum_{i=0}^{n}c_i\sum_{j=0}^{n}c_j(\varphi_i,\varphi_j)-2\sum_{i=0}^{n}c_i(\varphi_i,f)+(f,f). \tag{5.4}$$

视 S 为系数 $c_0,\cdots,c_n$ 的函数，欲使 S 达到极小值，令

$$\frac{\partial S}{\partial c_k}=2\sum_{i=0}^{n}c_j(\varphi_k,\varphi_j)-2(\varphi_k,f)=0$$

从而得用于求解系数 c_i 的**正规方程组**

$$\sum_{i=0}^{n}c_j(\varphi_k,\varphi_j)=(\varphi_k,f) \tag{5.5}$$

或

$$\begin{pmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n)\\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n)\\ \vdots & \vdots & \ddots & \vdots\\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n)\end{pmatrix}\begin{pmatrix}c_0\\ c_1\\ \vdots\\ c_n\end{pmatrix}=\begin{pmatrix}(\varphi_0,f)\\ (\varphi_1,f)\\ \vdots\\ (\varphi_n,f)\end{pmatrix}. \tag{5.6}$$

➢5.21 正规方程组中，系数矩阵为对称正定阵，有唯一解。故最优平方逼近的结果存在且唯一。

➢5.22 正规方程组可改写为

$$(\varphi_k,p-f)=0\quad(k=0,1,\dots,n). \tag{5.7}$$

意义：误差向量 $(p-f)$ 在由有限个向量 φ_k 张成的空间下无投影，控制「低维误差」。

➢5.23 正规方程组解法 1：选取一组线性无关、内积易求的简单函数作 $\{\varphi_k\}$（常用 $1,x,x^2,\cdots$），计算内积，代入正规方程组直接求解（在问题简单时最为便捷）。

➢**5.24** 正规方程组解法 2：选取 $\{\varphi_k\}$ 为一组正交多项式，则正规方程组简化为

$$(\varphi_k,\varphi_k)c_k = (\varphi_k,f),$$

从而易得 $c_k = \frac{(\varphi_k,f)}{(\varphi_k,\varphi_k)}$，进而得逼近多项式为

$$P(x) = \sum_{k=0}^{n} \frac{(\varphi_k,f)}{(\varphi_k,\varphi_k)}\varphi_k(x). \tag{5.8}$$

➢**5.25** 函数族 $\varphi_k(x)$ 的选取原则：

1. 直观性原则：观察 (x_i,y_i) 的分布，选取函数族。
2. 比较性原则：对不同的函数族，可分别拟合，再比较它们的误差向量孰大孰小。
3. 根据实际问题背景选择函数族（如对周期性变化的函数，可用三角函数族）。

➢**5.26** 离散的正规方程组（**最小二乘法**）：记

$$G = \begin{pmatrix} \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \varphi_0(x_2) & \varphi_1(x_2) & \cdots & \varphi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_m) & \varphi_1(x_m) & \cdots & \varphi_n(x_m) \end{pmatrix},\ W = \mathrm{diag}(\omega_1,\omega_2,\cdots,\omega_m),$$

$$y = (y_1\ y_2\ \cdots\ y_m)^T,\ c = (c_0\ c_2\ \cdots\ c_n)^T,$$

则原正规方程组可化为

$$(G^TWG)c = (G^TW)y. \tag{5.9}$$

特别地，若取 $\omega_i = 1$，则方程进一步简化为

$$G^TGc = G^Ty, \tag{5.10}$$

此即最小二乘方程，具体形式为

$$\begin{pmatrix} \sum_{i=1}^{m} 1 & \sum_{i=1}^{m} x_i & \cdots & \sum_{i=1}^{m} x_i^n \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 & \cdots & \sum_{i=1}^{m} x_i^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{m} x_i^n & \sum_{i=1}^{m} x_i^{n+1} & \cdots & \sum_{i=1}^{m} x_i^{2n} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_iy_i \\ \vdots \\ \sum_{i=1}^{m} x_i^ny_i \end{pmatrix} \tag{5.11}$$

➢**5.27** 当 $m=n+1$ 时，最小二乘拟合多项式即插值多项式。

✍**5.28** 对于若干数据点 (x_i,y_i)，欲采用 $y=be^{ax}$ 进行拟合，其中 a,b 为待定常数；可对该表达式取对数得

$$\ln y=\ln b+ax$$

再考虑误差函数

$$S(a,b)=\sum_{i=1}^{m}(\ln b+ax_i-\ln y_i)^2$$

令其对 a、b 的偏导数为 0，即可解出参数值。

➢**5.29** 最小二乘拟合/最优平方逼近的一般方法：

1. 可自选一组线性无关的基函数，求解正规方程组 (5.6) 或 (5.9)。一般选取 $\varphi_k(x)=x^k$。（难点在列、解方程）
2. 可利用三项递推关系 (5.1) 求解一族首一正交多项式 $g_0(x),\cdots,g_n(x)$，利用式 (5.8) 计算逼近的多项式。（难点在递推、计算内积。）
3. 可先通过变量代换 $x=\frac{a+b}{2}+\frac{b-a}{2}t$ 将区间 $[a,b]$ 替换为 $[-1,1]$，对变换后的函数 $\overline{f}(t)$ 用正交的 Legendre 多项式求出 $\overline{P}(t)$，最后用 $t=\frac{2x-a-b}{b-a}$ 还原到 $P(x)$。（难点在记 Legendre 多项式、展开和计算内积。）

✍**5.30** 求 $y=\sqrt{x}$ 在 $[0,1]$ 上最优平方逼近一次多项式。（答案：$p(x)=\frac{4}{5}x+\frac{4}{15}$。）

1. 用 $\varphi_0(x)=1$，$\varphi_1(x)=x$，可列出正规方程组

$$\begin{pmatrix}1 & 1/2\\ 1/2 & 1/3\end{pmatrix}\begin{pmatrix}c_0\\ c_1\end{pmatrix}=\begin{pmatrix}2/3\\ 2/5\end{pmatrix}$$

   容易求得 $c_0=\frac{4}{15}$，$c_1=\frac{4}{5}$，从而 $p(x)=\frac{4}{5}x+\frac{4}{15}$。

2. 用三项递推式，有 $g_0(x)=1$，

$$g_1(x)=x-\frac{(x,1)}{(1,1)}=x-\frac{1}{2},\tag{5.12}$$

   故可进一步计算

$$(g_1,g_2)=(x-\frac{1}{2},x-\frac{1}{2})=\int_0^1(x-\frac{1}{2})^2\mathrm{d}x=\frac{1}{12}$$

$$(f,g_0)=(\sqrt{x},1)=\frac{2}{3}$$

$$(f,g_1)=(\sqrt{x},x)-(\sqrt{x},\frac{1}{2})=\frac{2}{5}-\frac{1}{2}\cdot\frac{2}{3}=\frac{1}{15},$$

从而有

$$p(x)=\frac{2}{3}+\frac{1/15}{1/12}\cdot\left(x-\frac{1}{2}\right)=\frac{4}{5}x+\frac{4}{15}.$$

3. 用 Legendre 多项式求解，为此先作变量代换 $x=\frac{1}{2}+\frac{1}{2}t$，被插函数变为 $\overline{f}(t)=\sqrt{\frac{1+t}{2}}$，进而可用前两个 Legendre 多项式

$$P_0(t)=1,\ P_1(t)=t$$

逼近 $\overline{f}(t)$。可以计算各个内积为

$$(P_0,P_0)=\int_{-1}^{1}\mathrm{d}t=2$$

$$(P_1,P_1)=\int_{-1}^{1}t^2\mathrm{d}t=\frac{2}{3}$$

$$(f,P_0)=\int_{-1}^{1}\sqrt{\frac{1+t}{2}}=\frac{\sqrt{2}}{2}\int_{-1}^{1}\sqrt{1+t}\mathrm{d}(1+t)=\frac{\sqrt{2}}{2}\cdot\frac{2}{3}\cdot(2\sqrt{2})=\frac{4}{3}$$

$$\begin{aligned}(f,P_1)&=\int_{-1}^{1}\sqrt{\frac{1+t}{2}}=\frac{\sqrt{2}}{2}\int_{=1}^{1}(\sqrt{1+t})^3\mathrm{d}(1+t)-(f,P_0)\\&=\frac{\sqrt{2}}{2}\cdot\frac{2}{5}\cdot(4\sqrt{2})-\frac{4}{3}=\frac{4}{15}\end{aligned}$$

从而

$$\overline{p}(t)=\frac{4/3}{2}+\frac{4/15}{2/3}t=\frac{2}{3}+\frac{2}{5}t$$

回代 $t=2x-1$ 即得 $p(x)=\frac{4}{5}x+\frac{4}{15}$。

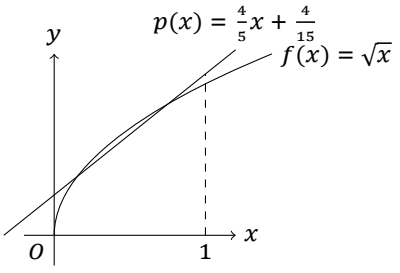


图 5.1: $y=\sqrt{x}$ 的最优逼近一次多项式示意图

第六章　数值积分与数值微分

➢**6.1** 常规（解析）积分的局限性：往往难以求得原函数，无法应用 Newton-Leibniz 公式；有时仅知函数在个别离散点的取值（列表函数）。

➢**6.2** 数值积分/数值微分的共同思路：用个别点处函数值$f(x_i)$的线性组合，来估计整个区间上的积分值或个别点附近的微分值。

$$\int_a^b f(x)\mathrm{d}x = \sum_{i=0}^{n} A_i f(x_i) \tag{6.1}$$

$$f'(x) = \sum_{i=0}^{n} B_i f(x_i) \tag{6.2}$$

➢**6.3** 主要问题：节点 x_i 的选取；求积系数 A_k 的计算；误差估计与评价。

➢**6.4** 记号：对于 $f(x)$，记其准确积分值为 $I[f] = \int_a^b f(x)\mathrm{d}x$，而记其数值估计公式为

$$Q[f] = \sum_{i=0}^{n} A_i f(x_i) \approx I[f].$$

并将数值积分公式的误差记为 $R[f] = I[f] - Q[f]$.

§6.1　插值型积分及其延伸

➢**6.5** 两种近似积分思路：

1. 按 Riemann 积分定义，作区间的划分，按若干小矩形估算：

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=o}^{n} f(x_i)(x_{i+1} - x_i)$$

2. 取若干数据点插值，用插值多项式 $p(x)$ 的积分估计函数的积分：

$$\int_a^b f(x)\mathrm{d}x \approx \int_a^b p(x)\mathrm{d}x = \int_a^b \left[\sum_{i=0}^{n} l_i(x)f(x_i)\right]\mathrm{d}x$$
$$= \sum_{i=0}^{n}\left(\int_a^b l_i(x)\mathrm{d}x\right)\cdot f(x_i).$$

共同特点是：用若干点上函数值 $f(x_i)$ 之线性组合估计整个区间上的积分值。问题：无法估计误差。应当从此种思路出发，直接构造估计公式，进而就可估计误差。

➢**6.6 Newton-Cotes 型求积公式**：在区间上等距取定若干插值点，构造插值多项式并以其积分代替 $f(x)$ 的积分。

➢**6.7** 记号：设节点数为 $n+1$ 个：$x_0, x_1, \cdots, x_n$（一般常取 $n = 1, 2, 4$），记此时各点的距离为 $h = \frac{b-a}{n}$。

➢**6.8 梯形公式**：当 $n = 1$ 时，取定区间的左右两端点为 x_0 与 x_1，可得到

$$Q_1[f] = \frac{b-a}{2}[f(a) + f(b)]. \tag{6.3}$$

此即梯形求积公式，常将 $Q_1[f]$ 记为 T_1。由第一积分中值定理 **(1)** 可估计误差为

$$R_1[f] = \frac{(b-a)^3}{12}f''(\eta),\ \eta \in (a,b) \tag{6.4}$$

➢**6.9 Simpson 公式**：当 $n = 2$ 时，$h = \frac{b-a}{2}$，将取得以下等距节点：

$$x_0 = a,\ x_1 = \frac{a+b}{2},\ x_2 = b.$$

作插值多项式并积分，可得求积公式为

$$Q_2[f] = \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b).\right] \tag{6.5}$$

此即 Simpson 公式，常将 $Q_2[f]$ 简记为 S_1。利用之后的方法 **(2)** 可估计误差为

$$R_2[f] = -\frac{(b-a)^5}{2880}f^{(4)}(\eta). \tag{6.6}$$

(1) 设 $f, g \in C[a,b]$ 且 g 在 $[a,b]$ 不变号，则存在 $\eta \in [a,b]$ 使 $\int_a^b f(x)g(x)\mathrm{d}x = f(\eta)\int_a^b g(x)\mathrm{d}x$。

(2) 即广义 Peano 定理与「24K 金法」。

➢**6.10 Cotes 求积公式**：当 $n=4$ 时，$h=\frac{b-a}{4}$，插值并积分可得

$$Q_4[f]=\frac{b-a}{90}[7f(a)+32f(a+h)+12f(a+2h)+32f(a+3h)+7f(b).] \quad (6.7)$$

此即 Cotes 求积公式，常将 $Q_4[f]$ 简记为 C_1。利用之后的方法可估计误差为

$$R_4[f]=-\frac{(b-a)^7}{1935360}f^{(6)}(\eta). \quad (6.8)$$

➢**6.11 代数精度**：设有数值积分公式，且对于不超过 m 次的多项式 f 均有 $R[f]=0$ 成立，而对某 $m+1$ 次多项式 f 即有 $R[f]\neq 0$，则称该数值积分公式的代数精度为 m。

➢**6.12** 对梯形公式，其代数精度为 1；对 Simpson 公式，其代数精度为 2；对 Cotes 公式，其代数精度为 4。

➢**6.13** 不宜采用次数过高的多项式插值：Runge 现象 [3]。

➢**6.14 复化求积公式**：对每个小区间 $[x_{k-1},x_k]$ 分别运用积分公式，再将结果求和。

➢**6.15** 公式表述：对由 $n+1$ 个数据点所划分的 n 个小区间，对以上三种插值型积分公式的结果如下。

- 取 $n=1$，得**复化梯形公式**：

$$T_n=\frac{h}{2}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right], \quad (6.9)$$

  其误差估计为 $R_{T_n}=-\frac{b-a}{12}h^2f''(\eta)$。

- 取 $n=2$，得**复化 Simpson 公式**：

$$S_n=\frac{h}{2}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right], \quad (6.10)$$

  其误差估计为 $R_{S_n}=-\frac{b-a}{2880}h^4f^{(4)}(\eta)$。

[3] 且根据后面的内容可知，次数过高的 Newton-Cotes 公式稳定性差，结果易发散

- 取 $n=4$，得**复化 Cotes 公式**：

$$C_n=\frac{h}{90}\left\{7f(a)+32\sum_{i=0}^{n-1}\left[f\left(x_i+\frac{h}{4}\right)+f\left(x_i+\frac{3h}{4}\right)\right]+12\sum_{i=0}^{n-1}f\left(x_i+\frac{h}{2}\right)+14\sum_{i=1}^{n-1}f(x_i)+7f(b)\right\}. \tag{6.11}$$

其误差估计为 $R_{c_n}=-\frac{b-a}{1935360}h^6f^{(6)}(\eta)$。

➢**6.16** **变步长积分法**：为进一步减小误差，考虑在用复化求积公式时不断缩小步长。为此，先凑出一个近似的误差估计式（减小步长的依据）。设用复化梯形公式 T_n 求积分，当取得 $n+1$ 个点时的误差估计式为

$$R_n[f]=I[f]-T_n=-\frac{b-a}{12}h^2f''(\eta_1) \tag{6.12}$$

将子区间数量翻倍，则取得 $2n+1$ 个点时的误差估计式为

$$R_{2n}[f]=I[f]-T_{2n}=-\frac{b-a}{12}\left(\frac{h}{2}\right)^2f''(\eta_2) \tag{6.13}$$

若估计 $f''(\eta_1)\approx f''(\eta_2)$，则联立两式即可得到

$$R_{2n}[f]=I[f]-T_{2n}\approx\frac{1}{3}(T_{2n}-T_n) \tag{6.14}$$

即积分误差可以用变步长结果间的差距 $T_{2n}-T_n$ 所估计。因此，若希望实现 $R[f]<\varepsilon$，可按如下步骤实施变步长复化积分：

1. $n=1$，$h=b-a$，计算 T_n；
2. 缩小步长 h 至原来的一半，计算 T_{2n}；
3. 验证误差估计条件 $|T_{2n}-T_n|<\varepsilon$ 是否满足，若满足则停止，若不满足则重复上一步。

➢**6.17** T_{2n} 的迭代算法：

$$T_{2n}=\frac{1}{2}T_n+\frac{h}{2}\sum_{k=1}^{n}f\left(\frac{x_{k-1}+x_k}{2}\right) \tag{6.15}$$

➢**6.18** 变步长积分公式中，同样可采用复化 Simpson 公式或复化 Cotes 公式：

$$I-S_{2n}\approx\frac{1}{4^2-1}(S_{2n}-S_n), \tag{6.16}$$

$$I-C_{2n}\approx\frac{1}{4^3-1}(C_{2n}-C_n). \tag{6.17}$$

➢**6.19** 启发：由误差估计式 (6.14) 可以猜想，以下的估计式同样成立：

$$I \approx T_{2n} + \frac{1}{4-1}(T_{2n} - T_n)$$

而将 T_n 与 T_{2n} 的表达式代入之后「竟然」发现

$$T_{2n} + \frac{1}{4-1}(T_{2n} - T_n) = S_n \tag{6.18}$$

由此说明，可用低精度公式组合出高精度公式[(4)]。对复化 Simpson 公式实行类似操作可推得

$$S_{2n} + \frac{1}{4^2-1}(S_{2n} - S_n) = \frac{4^2 S_{2n} - S_n}{4^2-1} = C_n \tag{6.19}$$

➢**6.20** **Romberg 积分**：进一步考虑对 Cotes 公式的结果，将得到一个新的积分公式：

$$C_{2n} + \frac{1}{4^3-1}(C_{2n} - C_n) = \frac{4^3 C_{2n} - C_n}{4^3-1} = R_n \tag{6.20}$$

称由上式表述的 R_n 为 **Romberg 积分**，其代数精度 $m = 7$，误差估计式为

$$R[f] = K \cdot h^8 f^{(8)}(\eta). \tag{6.21}$$

➢**6.21** 一般不在 Romberg 积分的基础上进一步递推，因被积函数的高阶导数性态不易估计，仍可能出现 Runge 现象；且 $R_{2n} - R_n$ 一项很小，舍入误差较大；计算量也过大。

➢**6.22** Romberg 积分可由最基本的复化梯形公式逐阶递推而来，常用 **Romberg 计算图**辅助计算。

T		S		C		R
T_1						
	↘					
T_2	→	S_1				
	↘		↘			
T_3	→	S_2	→	C_1		
	↘		↘		↘	
T_4	→	S_3	→	C_2	→	R_1
	↘		↘		↘	
T_5	→	S_4	→	C_3	→	R_2
⋮		⋮		⋮		⋮

图 6.1: Romberg 计算图

[(4)] 本质上是由插值点的增加所致。

§6.2 待定系数法与 Gauss 型积分

☞**6.23** 积分公式 $Q[f]$ 之代数精度为 m 的充要条件是

$$R[x^k]=0,\ (k=0,1,2,\cdots,m),\ R[x^{m+1}]\neq 0.$$

可利用此结果，在积分公式中分别代入 $f(x)=x^k$，验算积分公式的代数精度。

➢**6.24** **待定系数法**：根据数值积分公式的一般形式 $Q[f]=\sum_{i=0}^{k}A_if(x_i)$，在给定条件下（如达到指定的代数精度、给定已知节点），代入若干不同的 $f(x)$，求解出公式中未知的节点 x_i 与系数 A_i。

✍**6.25** 待定系数法导出梯形公式：设 $Q[f]=A_1f(a)+A_2f(b)$，希望其代数精度为 $m=1$，则有

- 令 $f(x)=1$，由 $I[f]=Q[f]$ 有 $A_1-A_2=b-a$；
- 令 $f(x)=x$，由 $I[f]=Q[f]$ 有 $A_1\cdot a+A_2\cdot b=\frac{1}{2}(b^2-a^2)$。

解得 $A_1=A_2=\frac{b-a}{2}$。故积分公式为

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

此即梯形公式。

➢**6.26** **广义 Peano 定理**：设 $Q[f]$ 的截断误差 $R[f]$ 是区间上 $m+1$ 阶导数连续的函数之线性泛函[5]，且其代数精度为 m，则有 $R[f(x)]=R[e(x)]$，其中

$$e(x)=\frac{f^{(m+1)}(\xi)}{(m+1)!}(x-\tilde{x}_0)(x-\tilde{x}_1)\cdots(x-\tilde{x}_m) \tag{6.22}$$

$\tilde{x}_0,\tilde{x}_1,\cdots,\tilde{x}_m$ 为 $[a,b]$ 上任意一点，$\xi\in[a,b]$ 与这 $m+1$ 个点的选取有关。

➢**6.27** 通过合适的取点，可由广义 Peano 定理求得各积分公式的误差估计。一般要求：

- $e(x)$ 表达式中的 $(x-\tilde{x}_0)(x-\tilde{x}_1)\cdots(x-\tilde{x}_m)$ 项在 $[a,b]$ 不变号；
- 选取 $\tilde{x}_0,\tilde{x}_1,\cdots,\tilde{x}_m$ 最好使 $Q[e]$ 为 0，从而 $R[f]=I[e]$ 可直接算出[6]。

[5]线性泛函满足：$R[c_1f_1+c_2f_2]=c_1R[f_1]+c_2R[f_2]$。

[6]若 $Q[f]$ 的形式对称，则可以通过取若干对称的点实现这一点，参见下面的例子（6.28）。

✍**6.28** 用广义 Peano 定理估计 Simpson 公式

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx Q[f] = \frac{1}{3}[f(-1) + 4f(0) + f(1)]$$

的误差，可考虑取 $-1, 0, 0, 1$ 四点（其中 0 为重节点），代入 (6.22) 式即得

$$e(x) = \frac{f^{(4)}(\xi)}{4!}(x+1)x^2(x-1)$$

可以算得 $Q[e] = 0$，故直接有

$$\begin{aligned} R[f] = I[e] &= \int_{-1}^{1} \frac{f^{(4)}(\xi)}{4!}(x+1)(x-1)x^2\mathrm{d}x \\ &= \frac{1}{90}f^{(4)}(\eta)\ (\eta \in [a,b]) \end{aligned}$$

其中，在 $(x+1)(x-1)x^2$ 不变号的前提下应用了第一积分中值定理。

➢**6.29** **简化解法**：设数值积分公式 $Q[f]$ 的代数精度为 m，据广义 Peano 定理知其误差项一定为 $Kf^{(m+1)}(\eta)$ 的形式；代入 $f(x) = x^{m+1}$ 即有

$$K \cdot m! = R[f] = I[f] - Q[f]$$

将 $I[f]$ 与 $Q[f]$ 算出，即可求得系数 K。（当 $m = 4$ 时，方程左侧为 $24K$，故戏称此法为「24K 金法」。）

✍**6.30** 用「24K 金法」估计 Simpson 公式的误差。（答案与例 6.28 相同。）

✍**6.31** 用「24K 金法」估计积分公式

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx Q[f] = \frac{4}{3}f\left(-\frac{1}{2}\right) - \frac{2}{3}f(0) + \frac{4}{3}f\left(\frac{1}{2}\right)$$

的误差。（答案：$R[f] = \frac{7}{720}f^{(4)}(\eta)$。）

➢**6.32** 在一般的积分公式 $Q[f] = \sum_{i=0}^{n} A_i f(x_i)$ 中，全部的待定参数共 $2n+2$ 个，可将它们全部通过待定系数法解出（而不必预先给定）；因此，对由 $n+1$ 个节点确定的积分公式，可以期望其最高具有 $m = 2n+1$ 的代数精度，进而列出 $m+1 = 2n+2$ 个方程将待定参数解出。

➢**6.33** 对代数精度为 $2n+1$ 的积分公式 $Q[f] = \sum_{i=0}^{n} A_i f(x_i)$，可代入以下的 $2n+2$ 次多项式

$$p(x) = (x-x_0)^2(x-x_1)^2\cdots(x-x_n)^2$$

则有 $Q[p(x)]=\sum_{i=0}^{n}A_ip(x_i)=0$，而 $I[p(x)]$ 必然为正数，故可见 $Q[f]$ 的代数精度不可能达到 $2n+2$，任何情况下的最高代数精度只能是 $2n+1$。

✎**6.34** 求解使积分公式

$$\int_{-1}^{1}f(x)\mathrm{d}x=Q[f]=A_0f(x_0)+A_1f(x_1)$$

代数精度尽可能高的 A_0,A_1,x_0,x_1，并估计误差。（答案：$Q[f]=f(-1/\sqrt{3})+f(1/\sqrt{3})$，$m=3$，$R[f]=\frac{1}{135}f^{(4)}(\eta)$.）

➢**6.35 Gauss 求积公式**：具有 $n+1$ 个节点，代数精度达到 $2n+1$ 的数值积分公式。相应的节点称为 **Gauss 点**。

☞**6.36** 求积公式 $Q[f]=\sum_{i=0}^{n}A_if(x_i)$ 是 Gauss 求积公式的**充要条件**为：

- x_i 为 $[a,b]$ 上关于权系数 $\omega(x)$ 正交的多项式 $g_{n+1}(x)$ 之零点；
- 求积系数 A_i 按下式确定：

$$A_i=\int_a^b\omega(x)l_i(x)\mathrm{d}x \tag{6.23}$$

☞**6.37** 引理：设 $\{g_k(x)\}$ 为 $[a,b]$ 上关于 $\omega(x)$ 正交的首一正交多项式，则有

$$\frac{g_{k+1}(x)}{x-x_i}=\frac{\gamma_n}{g_n(x_i)}\sum_{k=0}^{n}\frac{g_k(x)g_n(x_i)}{\gamma_k} \tag{6.24}$$

其中 $x_i\ (i=0,1,\dots,n)$ 为 g_{n+1} 的零点，$\gamma_k=(g_k,g_k)$ 与正交多项式三项递推关系（条目 5.13）中的含义一致 **(7)**。

☞**6.38 Gauss 求积公式系数公式**：给定一组首一正交多项式 $\{g_k(x)\}$，则 Gauss 求积公式 $Q[f]=\sum_{i=0}^{n}A_if(x_i)$ 中的系数 A_i 可按

$$A_i=\frac{\gamma_n}{g'_{n+1}(x_i)g_n(x_i)}\quad(i=0,1,\dots,n) \tag{6.25}$$

计算 **(8)**。

---

(7) 推导也须采用 5.13 中的三项递推关系：对其中 $g_{k+1}(x)$ 的表达式进行若干变换，再对 k 求和就可得到此式。

(8) 证明思路：根据 x_i 为 $g_{n+1}(x)$ 的条件，将式 (6.23) 中的 Langrange 插值基函数变换为 $l_i(x)=\frac{g_{n+1}(x)}{(x-x_i)g'_{n+1}(x_i)}$ 即可。

☞**6.39 Gauss 求积公式截断误差**：若 $f \in C^{2n+2}[a,b]$，且 $g_{n+1}(x)$ 为一个首一多项式，则有

$$R[f] = \frac{\gamma_{n+1}}{(2n+2)!} f^{(2n+2)}(\eta). \tag{6.26}$$

☞**6.40** Gauss 型求积公式的系数全大于 0。**(9)**

➢**6.41** Gauss 型求积公式的求解过程：

1. 按三项递推关系 (5.1) 求出在 $[a,b]$ 上关于 $\omega(x)$ 正交的多项式 $g_{n+1}(x)$；
2. 求出 $g_{n+1}(x)$ 的 $n+1$ 个根 $x_0, x_1, \cdots, x_n$。
3. 按式 (6.25) 求解积分公式中的系数。
4. 按 (6.26) 式的结果求解误差。

➢**6.42** Gauss 求积公式的稳定性：设计算函数值 $f(x_i)$ 时的舍入误差可表示为

$$|f(x_i) - \tilde{f}(x_i)| \le \varepsilon$$

其中 ε 为各子区间上最大的舍入误差限，则总的舍入误差估计为

$$E = \left| \sum_{i=0}^{n} A_i f(x_i) - \sum_{i=0}^{n} A_i \tilde{f}(x_i) \right| \le \sum_{i=0}^{n} |A_i| \varepsilon. \tag{6.27}$$

对 Gauss 求积公式，所有的 A_i 均为正，故有

$$\sum_{i=0}^{n} |A_i| = \sum_{i=0}^{n} A_i = \int_a^b \omega(x)\mathrm{d}x = \gamma_0$$

从而说明 $E \le \gamma_0 \varepsilon$，舍入误差可控。这说明 Gauss 求积公式是稳定的。

§6.3 数值微分公式

➢**6.43** 最基本的算法：割线代切线

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = A_i f(x_i) + A_{i+1} f(x_{i+1})$$

可见数值微分公式与数值积分公式本质相同，均为个别点函数值的组合。其缺陷与近似积分法（条目 6.5）相同：误差无法估计。为此，应直接从一般的公式入手，在给定条件下直接求解节点与系数，并估计误差。

(9) 证明：对 $f(x) = l_k^2(x)$ 应用 Gauss 求积公式。与之相反，Newton-Cotes 公式的系数无法满足此种条件。

➢**6.44** **插值型数值微分公式**：用插值多项式 $L_n(x)$ 代替原有函数 $f(x)$，求其导数：

$$f(x)=L_n(x)+R_n(x)\ \Rightarrow\ f^{(k)}(x)=L_n^{(k)}(x)+R_n^{(k)}(x).$$

➢**6.45** 插值型公式的误差估计：

$$R_n^{(k)}(x)=\frac{\mathrm{d}^k}{\mathrm{d}x^k}\left[\frac{f^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x)\right]=\frac{\mathrm{d}^k}{\mathrm{d}x^k}\big(f[x_0,x_1,\cdots,x_n,x]\pi_{n+1}(x)\big)$$

☞**6.46** 差商的导数有如下性质：

$$\frac{\mathrm{d}^k}{\mathrm{d}x^k}\big(f[x_0,x_1,\cdots,x_m,x]\big)=(k!)\cdot f[x_0,x_1,\cdots,x_m,x,x,\cdots,x]. \tag{6.28}$$

➢**6.47** 两点数值微分公式（$n=1,k=1$）：对 x_0,x_1,x 三点作 Newton 插值[(10)]，利用差商导数的性质对插值多项式 $f(x)$ 求导，可以求得

$$f'(x)=f[x_0,x_1]+f[x_0,x_1,x,x](x-x_0)(x-x_1)+f[x_0,x_1,x](2x-x_0-x_1)$$

分别代入 x_0 与 x_1 可得两个数值微分公式：

$$f'(x_0)=\frac{f(x_1)-f(x_0)}{x_1-x_0}+\frac{1}{2}f''(\xi_0)(x_0-x_1) \tag{6.29}$$

$$f'(x_1)=\frac{f(x_1)-f(x_0)}{x_1-x_0}+\frac{1}{2}f''(\xi_1)(x_0-x_1) \tag{6.30}$$

可用记号 $h=x_1-x_0$ 简化写法。

➢**6.48** 三点数值微分公式（$n=2,k=1$）：设三个等距（间距为 h）节点分别为 x_0,x_1,x_2，用 Newton 插值多项式可得到

$$f'(x_0)=\frac{1}{2h}[-3f(x_0)+4f(x_1)-f(x_1)]+\frac{h^2}{3}f'''(\xi_0) \tag{6.31}$$

$$f'(x_1)=\frac{1}{2h}[-f(x_0)+f(x_2)]-\frac{h^2}{6}f'''(\xi_1) \tag{6.32}$$

$$f'(x_2)=\frac{1}{2h}[f(x_0)-4f(x_1)+3f(x_2)]+\frac{h^2}{3}f'''(\xi_2). \tag{6.33}$$

➢**6.49** 三点二阶数值微分公式（$n=2,k=2$）：在以上公式推导过程中再求一

[(10)] 也可用 Lagrange 插值法，并直接使用 Lagrange 插值法的误差估计式。

次导，分别代入三个等距节点即得

$$f''(x_0)=\frac{1}{h^2}[f(x_0)-2f(x_1)+f(x_2)]-hf'''(\xi_1)+\frac{h^2}{6}f^{(4)}(\xi_2) \tag{6.34}$$

$$f''(x_1)=\frac{1}{h^2}[f(x_2)-2f(x_1)+f(x_2)]-\frac{h^2}{12}f^{(4)}(\xi) \tag{6.35}$$

$$f''(x_2)=\frac{1}{h^2}[f(x_0)-2f(x_1)+f(x_2)]+hf'''(\xi_1)+\frac{h^2}{6}f^{(4)}(\xi_2). \tag{6.36}$$

➢**6.50** 一般，常用以下两个在区间中点 x_1 取得的数值微分公式：

$$f'(x)=\frac{f(x+h)-f(x-h)}{2h}-\frac{h^2}{6}f'''(\xi), \tag{6.37}$$

$$f''(x)=\frac{f(x+h)-2f(x)+f(x+h)}{h^2}-\frac{h^2}{12}f^{(4)}(\xi). \tag{6.38}$$

➢**6.51** 数值微分的**待定系数法**：类似于数值积分，用待定系数法追求更高的代数精度 **(11)** 。

✎**6.52** 给定 x_0,x_1，为求得下列形式的数值微分公式

$$f''(x_0)\approx c_0f(x_0)+c_1f'(x_0)+c_2f(x_1)$$

中系数 c_0,c_1,c_2 的值，首先期望该公式具有 $m=2$ 的代数精度 **(12)** ，进而可依次代入 $f(x)=1$、$f(x)=x$、$f(x)=x^2$：

$$\begin{cases}-c_0-c_2=0\\-c_1-c_2h=0\\2-c_2h^2=0\end{cases}\Rightarrow\begin{cases}c_0=-\frac{2}{h^2}\\c_1=-\frac{2}{h}\\c_2=\frac{2}{h^2}\end{cases}$$

可将 $f(x)=x^3$ 代入最终的公式中，解得 $R[f]=-2h\neq 0$，故该公式的代数精度止于 2。此处可用广义 Peano 定理，取 x_0,x_0,x_1 三个节点构筑 $e(x)$，从而

$$\begin{aligned}R[f]=R[e(x)]&=e''(x_0)-\left[-\frac{2}{h^2}e(x_0)+\frac{2}{h}e'(x_0)-\frac{2}{h^2}e(x_1)\right]\\&=e''(x_0)=-\frac{h}{3}f'''(\xi).\end{aligned}$$

（当然，也可以用「24K 金法」，将 $f(x)=x^3$ 直接代入 $I[f]=R[f]$ 之中，求得系数 $K=-\frac{h}{3}$。）

(11) 此处，代数精度的定义与插值函数、数值积分中的相同。

(12) 此时刚好可列出三个方程解出三个系数。

➢6.53 **Richardson 外推法**：类似于 Romberg 积分法，用低精度公式递推出高精度公式。

➢6.54 **变步长微分法**：类于数值积分中的类似方法。

✎6.55 欲求得数值微分公式

$$f''(x) \approx Af(x-h) + Bf(x) + Cf(x+h)$$

中的系数 A, B, C，可利用 Taylor 公式展开等式右侧的 $f(x-h)$ 与 $f(x-h)$：

$$f''(x) = (A+B+C)f(x) + (-A+C)hf'(x) + (A+C)\frac{h^2}{2}f''(x) + R(x)$$

其中 $R(x)$ 为余项。左右对应项相等，故有

$$\begin{cases} A+B+C=0 \\ -A+C=0 \\ A+C=\frac{2}{h^2} \end{cases} \Rightarrow \begin{cases} A=-\frac{1}{h^2} \\ B=-\frac{2}{h^2} \\ C=\frac{1}{h^2} \end{cases}$$

故可推出数值微分公式为

$$D_h[f] = \frac{1}{h^2}[f(x-h) - 2f(x) + f(x+h)]$$

为提高该公式的精度，可再取 $h/2$ 为步长[13]，能递推出近似关系

$$f''(x) - D_{\frac{1}{2}h}[f] \approx \frac{1}{4}(f''(x) - D_h[f]) \Rightarrow f''(x) \approx \frac{1}{3}\left[4D_{\frac{h}{2}}[f] - D_h[f]\right].$$

记 $\overline{D}[f] = \frac{4}{3}D_{\frac{h}{2}}[f] - \frac{1}{3}D_h[f]$，该公式相较于上式更为精确。

[13] 从这里出发，就可以如 Romberg 积分法的推导过程一样，导出形式类似的 Richardson 外推法。详见李乃成、梅立泉《数值分析》第 205 – 206 页 6.4.3 节「外推求导法」。

第七章　非线性方程迭代解法

➢**7.1** **非线性方程**：在化简方程为 $f(x)=0$ 的形式后，$f(x)$ 中含有 x 的非线性项，如 x^2、$\sin x$、e^x 等。线性方程以外的代数、函数方程均是非线性方程。

➢**7.2** 非线性方程的迭代解法：泛指在已知结果的基础上，从给定初值 $x^{(0)}$ 出发，利用统一的迭代格式 $x^{(k+1)}=f(x^{(k)})$，使递推数列 $\{x^{(k)}\}$ 逼近方程解 x^* 的解法。

➢**7.3** 迭代解的要素：

1. 迭代格式 $x^{(k+1)}=f(x^{(k)})$ 的构造；
2. 初值 $x^{(0)}$ 的选取；
3. 迭代数列 $\{x^{(k)}\}$ 的收敛性与正确性；
4. 迭代终止条件与误差估计。

§7.1　迭代法简述

➢**7.4** **二分法**：根据零点存在定理，不断二分方程之解所在的区间，用「区间套」逼近方程之解。二分法过程稳定，但运算量大（需反复计算函数值），且收敛较慢[(1)]，故常用于初步确定初值 $x^{(0)}$ 的范围（而不用于确定最终解）。

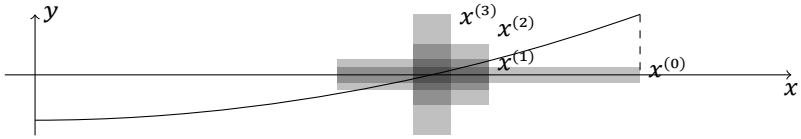


图 7.1: 二分法示意（重叠的灰色格子即不断缩窄的区间套）

➢**7.5** **简单迭代法**：构造非线性方程 $f(x)=0$ 的同解变形 $x=\varphi(x)$，例如

$$x=x-f(x),\quad x=\sqrt{x^2-kf(x)}$$

[(1)] 每迭代 10 次，解所在的区间范围缩小到原来的 $1/1024\approx 1/1000$，相当于提高了 3 位有效数字。也即：需要迭代超过 3 次才能增加解的一位有效数字。

等，再构造对应迭代格式为

$$x_{k+1} = \varphi(x_k). \tag{7.1}$$

若 $\{x_k\}$ 收敛到 x^*，且 $\varphi(x)$ 也在 x^* 处连续，则对以上方程左右取极限即得 $x^* = \varphi(x^*)$，说明该迭代法可以逼近方程之解。

➢7.6 **Newton 法**：设非线性方程 $f(x) = 0$ 之解为 x^*，即 $x^* = 0$，将其在 x_k 处展开可得

$$f(x_k) + f'(x_k)(x^* - x_k) + \cdots = 0 \tag{7.2}$$

舍去二阶项，有 $f(x_k) + f'(x_k)(x^* - x_k) \approx 0$，进而可变换得到

$$x^* \approx x_k - \frac{f(x_k)}{f'(x_k)}$$

改写其为一个迭代格式即得

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{7.3}$$

可验证该方程确为 $f(x) = 0$ 的同解变形，故该迭代格式收敛时必能得到方程之解。（根据 Newton 法的几何意义，其又被称为**切线法**。）

➢7.7 **改进 Newton 法**：在上面的展开式 (7.2) 中保留到二次项，可解得：

$$x^* \approx x_k + \frac{-f'(x_k) \pm \sqrt{f'(x_k)^2 - 2f(x_k)f''(x_k)}}{f''(x_k)}$$

分母中的正负号不定，故可写出两种迭代格式：

$$\tilde{x}_{k+1} = x_k - \frac{f'(x_k) + \operatorname{sgn}(f'(x_k))\sqrt{f'(x_k)^2 - 2f(x_k)f''(x_k)}}{f''(x_k)}, \tag{7.4}$$

$$\overline{x}_{k+1} = x_k - \frac{2f(x_k)}{f'(x_k) + \operatorname{sgn}(f'(x_k))\sqrt{f'(x_k)^2 - 2f(x_k)f''(x_k)}}. \tag{7.5}$$

选取 $\tilde{x}_{k+1}$ 与 $\overline{x}_{k+1}$ 中距离 x_k 更近者，作为下一步迭代时的 x_{k+1} 即可。

➢7.8 **简化 Newton 法**：若 $f'(x)$ 难以计算，可将迭代格式 (7.3) 改写为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)} \tag{7.6}$$

即始终采用初值点处导数近似表示 $f'(x_k)$。

➢7.9 **弦割法**[(2)]：将迭代格式 (7.3) 中的导数 $f'(x_k)$ 用两点数值微分公式代替，可得到：

[(2)]弦割法的提出已有相当历史，西安交大 120 周年校庆时提出的「风云两甲子，弦割三世纪」即是为纪念其悠久的历史而作。

- 若用 x_k 与 x_{k-1} 作为迭代格式中的两点，则得

$$x_{k+1} = x_k - \frac{f(x_k)}{\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}} = x_k - \frac{f(x_k)(x_k-x_{k-1})}{f(x_k)-f(x_{k-1})} \tag{7.7}$$

此即**两点弦割法**。

- 若用 x_k 与 x_0 作为迭代格式中的两点，则得

$$x_{k+1} = x_k - \frac{f(x_k)}{\frac{f(x_k)-f(x_0)}{x_k-x_0}} = x_k - \frac{f(x_k)(x_k-x_0)}{f(x_k)-f(x_0)} \tag{7.8}$$

此即**单点弦割法**。

➢**7.10** 弦割法需要给定两步初值 x_0 与 x_1 才能进行，它们通常由其他方法获得。

➢**7.11** 切线法与弦割法的几何意义：如图 7.2 所示。

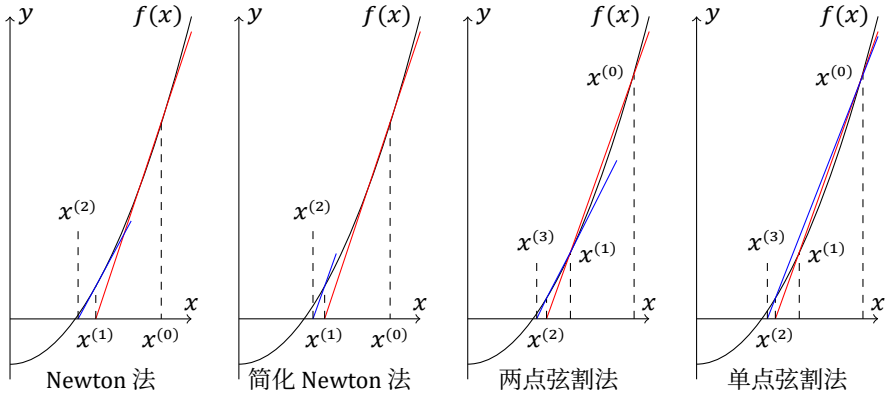


图 7.2: 迭代解法的几何意义

§7.2 迭代法的收敛理论

➢**7.12** 迭代法的两种**收敛性**：设非线性方程 $f(x)=0$ 在 $[a,b]$ 上有解 x^*，取定迭代格式 $x_{k+1}=\varphi(x_k)$ 与初值 x_0。若

- 存在 x^* 的一个小邻域，使得只要 x_0 在此邻域中就能保证 $\{x_k\}$ 收敛[3]，则称迭代格式**局部收敛**；
- 对任意的 $x_0 \in [a,b]$，都有 $\{x_k\}$ 收敛，则称迭代格式**全局收敛**。

[3]根据迭代格式的意义，当 $\{x_k\}$ 收敛时其必收敛于 x^*。

☞7.13 **全局收敛定理**（压缩映射原理）：设 $\varphi(x) \in C[a,b]$，且满足：

1. $x \in [a,b]$ 时，$\varphi(x) \in [a,b]$；
2. 存在 $L \in [0,1)$，使得对任意的 $x \in [a,b]$ 都有

$$|\varphi'(x)| \le L < 1. \tag{7.9}$$

则由 $\varphi(x)$ 所构造的 $x_{k+1} = \varphi(x_k)$ 将是一个在 $[a,b]$ 上全局收敛的迭代数列，且其满足以下误差估计式：

$$|x^* - x_k| \le \frac{L^k}{1-L}|x_1 - x_0|, \quad \text{（先验估计）} \tag{7.10}$$

$$|x^* - x_k| \le \frac{1}{1-L}|x_{k+1} - x_k|. \quad \text{（事后估计）} \tag{7.11}$$

➢7.14 以上定理的结果，仅是全局收敛的充分条件，而非必要条件。

☞7.15 **局部收敛定理**：设方程 $f(x) = 0$ 有解 x^*，利用其同解变形构造了一个迭代格式 $x_{k+1} = \varphi(x_k)$。若存在 x^* 的某邻域 $[x^* - \delta, x^* + \delta]$ 使

$$|\varphi'(x)| \le L < 1$$

则迭代格式 $x_{k+1} = \varphi(x_k)$ 在该邻域上局部收敛。

☞7.16 Newton 迭代法的局部收敛定理：设非线性方程 $f(x) = 0$ 有解 x^*，在 x^* 附近 $f(x)$ 二阶连续可微，$f'(x^*) \neq 0$，则 Newton 迭代格式 (7.3) 在 x^* 的充分小邻域内局部收敛。（或：当初值 x_0 充分靠近 x^* 时，Newton 迭代法收敛。）

☞7.17 Newton 迭代法的全局收敛定理：设非线性方程 $f(x) = 0$ 在 $[a,b]$ 上有解 x^*，且 $f(x)$ 在 $[a,b]$ 上二阶连续可微。则 Newton 迭代格式 (7.3) 在 $[a,b]$ 上全局收敛的充分条件是：

1. $f(a) \cdot f(b) < 0$；
2. 对任意的 $x \in [a,b]$，$f'(x) \neq 0$；
3. $f''(x)$ 在 $[a,b]$ 上不变号；
4. 初值 x_0 满足 $f(x_0)f''(x_0) > 0$。

☞7.18 弦割法局部收敛的条件与 Newton 法类似，只要 x_0, x_1 两步初值充分靠近 x^*。

☞7.19 弦割法全局收敛定理：在 Newton 法的所有条件之基础上，还应附加满足 $f(x_1)f''(x_1) > 0$。

☞7.20 **收敛阶**：设 $\{x_k\}$ 按某种迭代格式收敛于方程的解 x^*，若存在常数 $p \geq 1$ 与 $c > 0$ 使条件

$$\lim_{k\to\infty} \frac{|x^* - x_{k+1}|}{|x^* - x_k|^p} = c \neq 0 \tag{7.12}$$

成立，则称 $\{x_k\}$ 以 p 阶收敛到方程之解。称 c 为渐进误差常数。

➢7.21 $p = 1$ 时，称为**线性收敛**，否则称为**超线性收敛**。

☞7.22 （整数）**收敛阶定理**：设 $\varphi(x)$ 在不动点上 x^* 的邻域内有连续的 p 阶导数，则由 $x_{n+1} = \varphi(x_k)$ 生成的 $\{x_k\}$ 以 p 阶收敛的充要条件是：

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

➢7.23 常见迭代格式的收敛阶：

- 简单迭代法：当 $0 < |\varphi'(x^*)| < 1$ 时线性收敛。
- Newton 法：当 $f'(x) \neq 0$ 时二阶收敛。
- 两点弦割法：满足收敛条件时具有 $p = \frac{1+\sqrt{5}}{2} \approx 1.618$ 的收敛阶，超线性收敛。
- 单点弦割法：满足收敛条件时线性收敛。

➢7.24 **加速收敛方法**：促使原来收敛较慢的算法更快收敛，或使原来发散的迭代格式变为收敛。

➢7.25 **松弛因子加速法**：对迭代格式 $x = \varphi(x)$，在方程左右减去一带松弛因子 ω 的项 ωx，得

$$x - \omega x = \varphi(x) - \omega x$$

进而可得新的收敛格式

$$x = \frac{\varphi(x) - \omega x}{1 - \omega} = \psi(x) \tag{7.13}$$

为检查该迭代格式的收敛性，可以对其求导得

$$\psi'(x) = \frac{1}{1-\omega}(\varphi'(x) - \omega)$$

若代入 $\omega = \varphi'(x^*)$，则在 x^* 附近时 $|\psi(x^*)|$ 相当小，由此即可保证 $|\psi'(x)| \leq L < 1$ 的条件成立（将发散的迭代格式改进为收敛的），并提高收敛速率。

➢7.26 **Aitken 加速法**：略[4]。

[4]参见李乃成、梅立泉《数值分析》第 228 页「艾特肯加速法」，该加速方法是在假设迭代格式线性收敛的情况下作出的。

第八章　常微分方程数值解

➢**8.1** 本章只涉及一阶常微分方程初值问题[(1)]：

$$\begin{cases} y'(x) = f(x, y(x)), \quad x \in [a, b] \\ y(a) = y_0. \end{cases} \tag{8.1}$$

☞**8.2** **解的存在唯一性**：在由式 (8.1) 所述的初值问题中，若 $f(x, y)$ 在区域

$$D = \{(x, y) | a \le x \le b, -\infty < y < +\infty\}$$

中连续，且存在一个常数 L 使得对任意选取的 y 与 $\bar{y}$ 都有[(2)]

$$|f(x, y) - f(x, \bar{y})| \le \left| \frac{\partial f}{\partial y}(y - \bar{y}) \right| \le L|y - \bar{y}|, \quad x \in [a, b]$$

则该初值问题在 $[a, b]$ 上存在唯一连续可微解 $y = y(x)$。

➢**8.3** 数值解：用数值计算方法，求出初值问题 (8.1) 在给定的若干节点 $x_i \in [a, b]$ 上的解 $y_i \approx y(x_i)$。

➢**8.4** 一般记 x_i 处的精确解为 $y(x_i)$，数值近似解则记为 y_i。

§8.1　常用解法的导出

➢**8.5** **数值微分法**：在初值问题 (8.1) 中，用某一数值微分公式

$$y'(x_i) = y_i' + R[y]$$

（其中 y_i' 为近似解，$R[y]$ 为误差项）替换原方程中的 $y'(x_i)$，进而求出一个求解 y_i 的数值公式，误差估计由 $R[y]$ 给出。

---

(1)其他可能的问题包括：更高阶的常微分方程初值问题，以及常微分方程边值问题等。

(2)不等式中的第一个不等号，来自于多元函数的拟微分中值定理。

➢**8.6** **Euler 法**：在 (8.1) 中代入 $x = x_i$，并利用 x_i 与 x_{i+1} 间的两点数值微分公式 (6.29) 替换 $y'(x_i)$，可得到关系式

$$\frac{y(x_{i+1}) - y(x_i)}{h} - \frac{1}{2}y''(\xi_i)h = f(x_i, y(x_i))$$

作局部化假设 $y(x_i) = y_i$ （即认为 x_i 处的解已准确算出），并略去关于 h 的高阶项，则可最终获得关于 y_{i+1} 的递推关系

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{8.2}$$

此法称为 Euler 法。

➢**8.7** **局部截断误差**：根据两点数值微分公式的误差，可知 Euler 法的截断误差为

$$R[x_i] = \frac{h^2}{2}y''(\xi_i). \tag{8.3}$$

➢**8.8** **全局截断误差**：在局部化假设处处成立 [3] 之前提下，全局误差可估计为

$$R(x_N) = \sum_{i=0}^{N-1} R[x_i] = \frac{b-a}{2} \cdot hf''(\xi). \tag{8.4}$$

➢**8.9** **后退 Euler 法**：在初值问题中代入 $x = x_{i+1}$，并利用 x_i 与 x_{i+1} 间的两点数值微分公式 (6.30) 替换 $y'(x_i)$，可得到关系式

$$\frac{y(x_{i+1}) - y(x_i)}{h} - \frac{1}{2}y''(\xi_i)h = f(x_{i+1}, y(x_{i+1}))$$

作局部化假设 $y_{i+1} = y(x_{i+1})$ 并略去关于 h 的高阶项，可获得关系式

$$y_{i+1} = y_i + hf(x_{i+1}, y_{i+1}) \tag{8.5}$$

此法称为后退 Euler 法。

➢**8.10** 后退 Euler 法中，y_{i+1} 不能显式解出，在 f 为非线性函数时必须采用迭代解法，称该法为**隐式解法**；Euler 法中可直接由 x_i 解出 y_{i+1}，故称为**显式解法**。

➢**8.11** 后退 Euler 法的局部截断误差为

$$R[x_i] = -\frac{h^2}{2}y''(\xi_i), \tag{8.6}$$

与 Euler 法的符号相反。

[3] 即每一步所用的前提条件都准确无误。

➢**8.12** **中点法**：在初值问题中代入 $x = x_{i+1}$，并代入中点处的三点微分中值公式 (6.35)，可得到关系式

$$\frac{y(x_{i+1}) - y(x_{i-1})}{2h} - \frac{h^2}{6}y'''(\xi_i) = f(x_i, y(x_i))$$

作局部化假设 $y_{i+1} = y(x_{i+1})$ 并略去关于 h 的高阶项，可获得关系式

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i) \tag{8.7}$$

此法称为中点法。

➢**8.13** 中点法中，应用了 x_{i-1} 与 x_i 两个前提条件，为此需在初值条件 y_0 的基础上额外补充表头元素 y_1，为此称中点法为一个**多步法**。与此相对，两种 Euler 法都是**单步法**。

➢**8.14** 多步法的表头元素，常用单步法提供。

➢**8.15** 一般不采用点数更多的数值微分公式：精度提高不明显，且需要补充更多表头元素；精度受表头元素的制约。

☞**8.16** 公式的**精度**：若某一微分方程的数值解法的局部截断误差记为 $R[x_i]$，其满足

$$R[x_i] = y(x_{i+1}) - y_{i+1} = O(h^{p+1})$$

则称该解法（公式、方法）的精度为 p 阶。

➢**8.17** 两种 Euler 法的精度均为一阶，中点法为二阶。

➢**8.18** 数值积分法：对微分方程初值问题 (8.1) 作积分，得到同解的积分方程问题

$$y(x) = y(a) + \int_a^x f(x, y(x))\mathrm{d}x \quad (x \in [a, b]) \tag{8.8}$$

在问题中替换区间 $[a, b]$ 为 $[x_i, x_{i+1}]$，再利用数值积分公式替代方程中的积分表达式，求得关于 y_{i+1} 的关系式。

➢**8.19** 对子区间上积分 $\int_{x_i}^{x_{i+1}} f(x, y(x))\mathrm{d}x$，采用

- 矩形数值积分公式：可分别获得数值解法

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{8.9}$$

$$y_{i+1} = y_i + hf(x_{i+1}, y_{i+1}) \tag{8.10}$$

此即 Euler 法与倒退 Euler 法，均为一阶公式。

- 梯形公式：可获得数值解法

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \tag{8.11}$$

此公式也称**梯形公式**，是一个二阶隐式公式，误差估计为

$$R[x_i] = -\frac{h^3}{12} f'''(\xi_i). \tag{8.12}$$

- Simpson 公式：可获得数值解法

$$y_{i+1} = y_{i-1} + \frac{h}{3}[f(x_{i-1}, y_{i-1}) + 4f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \tag{8.13}$$

此公式也称为 **Simpson 公式**，是一个四阶两步隐式公式，误差估计为

$$R[x_i] = -\frac{h^5}{90} y^{(5)}(\xi_i). \tag{8.14}$$

➢**8.20 Adams 显式公式**：为求 y_{i+1} 的值，给定之前的 $(x_{i-k}, f_{i-k}), (x_{i-k+1}, f_{i-k+1}), \cdots, (x_i, f_i)$ 共 $(k+1)$ 个点（其中 $f_n = f(x_n, y_n)$），作插值多项式

$$f(x, y) = L_k(x) + R_k(x)$$

以获得 $y'(x)$ 的近似表示，进而在同解积分问题中近似代入

$$\int_{x_i}^{x_{i+1}} y'(x)\mathrm{d}x \approx \int_{x_i}^{x_{i+1}} L_k(x)\mathrm{d}x$$

从而可以获得一显式的数值解公式

$$y_{i+1} = y_i + \frac{h}{A} \cdot (b_0 f_i + b_1 f_{i-1} + \cdots + b_k f_{i-k}) \quad (i \geq k) \tag{8.15}$$

$$R[x_i] = B_k h^{k+2} y^{(k+2)}(\xi_i) \tag{8.16}$$

称为 Adams 显式公式。式中的系数 A、f_i、B_k 可据实推导，亦可直接查现成的系数表 [4]。

➢**8.21** Adams 隐式公式：与 Adams 显式公式推导类似，但在给定的 $k+1$ 数据点中，将最前的 (x_{i-k}, y_{i-k}) 替换为待求的 (x_{i+1}, y_{i+1})，仍作插值多项式，进而可求解得到一隐式的数值解公式

$$y_{i+1} = y_i + \frac{h}{A^*} \cdot (b_0^* f_{i+1} + b_1^* f_i + \cdots + b_k^* f_{i-k+1}) \quad (i \geq k) \tag{8.17}$$

$$R[x_i] = B_k^* h^{k+2} y^{(k+2)}(\xi_i) \tag{8.18}$$

称为 Adams 隐式公式。式中系数可据实推导，也可直接查表 [5]。

[4] 系数表参见李乃成、梅立泉《数值分析》第 269 页表 9.1。

[5] 系数表参见李乃成、梅立泉《数值分析》第 270 页表 9.2。

§8.2 预测-校正方法与一般性理论

➢**8.22** 一般而言，显式法易于计算，但隐式法的稳定性较显式法高。

➢**8.23** 可以考虑在求解过程中，先用（低阶）显式公式初步求得 y_{i+1} 的近似值，再直接代入（高阶）隐式公式中以直接求得更稳定的解。此类方法统称**预估-校正法**。

➢**8.24** **改进 Euler 法**：用 Euler 法预估，再用倒退 Euler 法校正：

$$\begin{cases} p_{i+1} = y_i + h(x_i, y_i) \\ y_{i+1} = y_i + h(x_{i+1}, p_{i+1}) \end{cases} . \tag{8.19}$$

此法仍是一阶解法，但稳定性比 Euler 法更高，也不用像倒退 Euler 法一样隐式求解。

➢**8.25** **Heun 方法**：用 Euler 法预估，用梯形公式校正：

$$\begin{cases} p_{i+1} = y_i + h(x_i, y_i) \\ y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, p_{i+1})] \end{cases} \tag{8.20}$$

此法是二阶方法。

➢**8.26** 为考虑更一般的情形，可将 Heun 方法改写为

$$\begin{cases} y_{i+1} = y_i + \frac{1}{2}K_1 + \frac{1}{2}K_2 \\ K_1 = hf(x_i, y_i) \\ K_2 = hf(x_i + h, y_i + K_1) \end{cases} \tag{8.21}$$

式中，K_1 与 K_2 是利用已有的 (x_i, y_i) 信息依次用 f 推演所得的补充信息。

➢**8.27** **Runge-Kutta 法**（RK 法）：将以上所言的「补充信息」一般化，可得如下的解法：

$$\begin{cases} y_{i+1} &= y_i + \lambda_1 K_1 + \lambda_2 K_2 + \cdots + \lambda_m K_m \\ K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + \alpha_2 h, y_i + \beta_{21} K_1) \\ K_3 &= hf(x_i + \alpha_3 h, y_i + \beta_{31} K_1 + \beta_{32} K_2) \\ \vdots & \\ K_m &= hf(x_i + \alpha_m h, y_i + \beta_{m1} K_1 + \cdots + \beta_{m,m-1} K_{m-1}) \end{cases} \tag{8.22}$$

式中的 $\lambda_k, \alpha_k, \beta_{ij}$ 等系数待定。为追求 m 阶的精度，即达到

$$R[x_i] = o(h^{m+1}) \tag{8.23}$$

的截断误差，可将 $R[x_i]$ 的表达式在 x_i 处展开，使其低于 $m+1$ 次各项系数为 0，进而列方程确定式 (8.22) 中各项系数。此类方法统称为 Runge-Kutta 法。

➢**8.28** 实际问题中，方程数往往小于系数个数，此时可自由选取个别系数，以确定其他系数。（由此得到的是不同的解法。）

✎**8.29** 二阶 Runge-Kutta 法：改进 Euler 法，变形 Euler 法[6]。

✎**8.30** 四阶 Runge-Kutta 法：最常用者为标准四级四阶 R-K 法：

$$\begin{cases} y_{i+1} &= y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 &= hf(x_i, y_i) \\ K_2 &= hf\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1\right) \\ K_3 &= hf\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_2\right) \\ K_4 &= hf(x_i + h, y_i + K_3) \end{cases} \tag{8.24}$$

➢**8.31** Runge-Kutta 是单步显式方法，可直接求解，精度很高。缺点是计算量大。

➢**8.32** 可将之前所提的所有解法统一成以下的形式：

$$y_{i+1} = \sum_{j=0}^{K} \alpha_j y_{i-j} + h \sum_{j=-1}^{K} \beta_j f_{i-j}. \tag{8.25}$$

当 $K = 0$ 时，其表示了一个单步法，否则表示了一个多步法。当 $\beta_{-1} = 0$ 时，其表示了一个显式公式，否则表示的是隐式公式。含 y_{i-j} 的项决定了方程的步数，而含 f_{i-j} 的项决定了方程的阶数。

➢**8.33** **待定系数法**：对由 (8.25) 概括的数值解法，可计算其截断误差

$$R[x_i] = y(x_{i+1}) - y_{i+1} = y(x_{i+1}) - \sum_{j=0}^{K} \alpha_j y(x_{i-j}) - h \sum_{j=-1}^{K} \beta_j y'(x_{i-j}). \tag{8.26}$$

利用 Taylor 公式将 $R[x_i]$ 中各项在 x_i 处展开，为尽可能地使方法有高的精度，设其各低次项的系数为 0，由此即可列解系数方程。进而可用广义 Peano 定理或「24K 金法」求解截断误差的系数。

[6] 参见李乃成、梅立泉《数值分析》第 275 页「变形欧拉法」。

✍**8.34** 为确定所有可能的三阶两步方法，根据一般形式 (8.25) 可设出如下形式的解法：

$$y_{i+1} = \alpha_0 y_i + \alpha_1 y_{i-1} + h(\beta_{-1} f_{i+1} + \beta_0 f_i + \beta_1 f_{i-1})$$

计算其截断误差为

$$R[x_i] = y(x_{i+1}) - \alpha_0 y(x_i) - \alpha_1 y(x_{i-1}) - h[\beta_{-1} y'(x_i + h) + \beta_0 y'(x_i) + \beta_1 y'(x_i - h)]$$

取 $x_i = 0$，并令 $R[x^k] = 0$ $(k = 0, 1, 2, 3)$（3 阶精度），可得

$$\begin{cases} 1 - \alpha_0 - \alpha_1 = 0 \\ h[1 + \alpha_1 - (\beta_{-1} + \beta_0 + \beta_1)] = 0 \\ h^2[1 - \alpha_1 - 2(\beta_{-1} - \beta_1)] = 0 \\ h^3[1 + \alpha_1 - 3(\beta_{-1} + \beta_1)] = 0 \end{cases}$$

此时有 4 个方程 5 个未知数，其中一个未知数可任意决定。设 α_1 任意，则可将其他 4 个未知数表示为：

$$\begin{cases} \alpha_0 = 1 - \alpha_1 \\ \beta_{-1} = \frac{5 - \alpha_1}{12} \\ \beta_0 = \frac{2 + 2\alpha_1}{3} \\ \beta_1 = \frac{5\alpha_1 - 1}{12} \end{cases}$$

考虑 α_1 的不同取值：

1. 若取 $\alpha_1 = 0$，则有 $\alpha_0 = 1$，$\beta_{-1} = \frac{5}{12}$，$\beta_0 = \frac{2}{3}$，$\beta_1 = -\frac{1}{12}$；此即 $k = 2$ 的 Adams 隐式公式。
2. 若取 $\alpha_1 = 1$，则有 $\alpha_1 = 0$，$\beta_{-1} = \beta_1 = \frac{1}{3}$，$\beta_0 = \frac{4}{3}$；此即 Simpson 公式，其精度事实上为 4 阶。
3. 也可以取 α_1 为其他值，得到不同的公式。

➢**8.35** 恭喜你把笔记看完了，请喝口水休息一下。

附录：考试内容评析

- 在本校，与「数值计算方法」相关的课程大致可以分为工科生的计算方法与数学系学生的数值分析两大类。后者较前者难度更高，对计算与证明过程有更详细的考察；而目前流传的「往年试卷」中，往往以数值分析的考卷居多，这会给修计算方法课程的同学造成误导。因此，请不要过于相信这些「往年考卷」。
- 关于复习：复习过程中并不需要做太多的「练习」，只需牢记相关知识点和例题即可。仅就这份笔记而言，考试中所有可能出现的知识点均已涉及到了。
- 关于考试题型：从近几年的情况来看，一般分为填空题和计算题两大类，各占一半左右的分数。其中：
  - 填空题主要考察知识点，基本上不需要计算（口算就能解决）。大多数题目都有「窍门」，很多看似复杂的题目之结果其实非常简单。课程中所有的基础知识点都有可能涉及到。分值很高，注意不要丢分。
  - 计算题更像是「证明题」或「简答题」，主要目的在于考察学生对各类计算方法原理的理解应用能力（主要）或推导证明能力（次要）。具体的数值计算量并不大。
- 计算题常见考点：以下考点基本上是固定的，在作业题中也时常操练，不需要特别担心。
  - 对矩阵做 LU 分解；
  - 判断三种线性方程迭代法的收敛性（近年来较少考，但有可能出此类题目）；
  - 对给定数据点做 Newton 插值；
  - 利用待定系数法推导简单的数值积分公式，使之达到指定代数精度；
  - 判断非线性方程迭代格式的收敛性。

索引