



数值分析

期末复习要点总结



第一章 误差

一. 误差的来源:

1. 模型误差
2. 观测误差
3. 截断误差
4. 舍入误差

二. 绝对误差、相对误差和有效数字

定义 设 x 为准确值 x^* 的一个近似值，称

$$e(x) = x - x^*$$

为近似值 x 的**绝对误差**，简称**误差**。

若

$$|e(x)| = |x - x^*| \leq \varepsilon$$

通常称 ε 为近似值 x 的**绝对误差限**，简称**误差限**。

定义 设 x 为准确值 x^* 的近似值， 称绝对误差与准确值之比为近似值 x 的**相对误差**，记为 $e_r(x)$

即

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

由于在计算过程中准确值 x^* 总是未知的，
故一般取相对误差为

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

如果存在正数 ε_r 使得

$$|e_r(x)| = \left| \frac{e(x)}{x} \right| \leq \varepsilon_r$$

则称 ε_r 为 x 的相对误差限。

有效数字

一般地，如果近似值的规格化形式为

$$x = \pm 0.a_1 a_2 \cdots a_n \cdots \times 10^m$$

其中 m 为整数， $a_1 \neq 0, a_i (i = 1, 2, \dots)$ 为0到9之间的整数.

如果

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

则称近似值有 n 位有效数字.

误差传播规律

$$\begin{cases} e(x_1 \pm x_2) = e(x_1) \pm e(x_2) \\ e_r(x_1 \pm x_2) = \frac{x_1}{x_1 \pm x_2} e_r(x_1) \pm \frac{x_2}{x_1 \pm x_2} e_r(x_2) \end{cases}$$

$$\begin{cases} e(x_1 x_2) \approx x_2 e(x_1) + x_1 e(x_2) \\ e_r(x_1 x_2) \approx e_r(x_1) + e_r(x_2) \end{cases}$$

$$\begin{cases} e\left(\frac{x_1}{x_2}\right) \approx \frac{1}{x_2} e(x_1) - \frac{x_1}{x_2^2} e(x_2) \\ e_r\left(\frac{x_1}{x_2}\right) \approx e_r(x_1) - e_r(x_2) \end{cases}$$

例 设近似数 $a = 1.557$ 是某真值 x 经四舍五入所得, 试求其绝对误差限和相对误差限.

解 由于 a 经四舍五入得到, 故

$$\begin{aligned} |e(a)| &\leq \frac{1}{2} \times 10^{-3} \\ |e_r(a)| &= \left| \frac{e(a)}{a} \right| \leq \frac{\frac{1}{2} \times 10^{-3}}{1.577} \\ &= 3.1705 \times 10^{-4} \end{aligned}$$

数值计算中的一些原则

1. 避免两个相近的数相减
2. 避免大数“吃”小数的现象
3. 避免除数的绝对值远小于被除数的绝对值
4. 要简化计算，减少运算次数，提高效率
5. 要有数值稳定性,即能控制舍入误差的传播

例如 为提高数值计算精度,当正数 x 充分大时,应将

$$\sqrt{2x+1} - \sqrt{2x-1}$$

改写为

$$\frac{(\sqrt{2x+1} - \sqrt{2x-1})(\sqrt{2x+1} + \sqrt{2x-1})}{\sqrt{2x+1} + \sqrt{2x-1}} = \frac{2}{\sqrt{2x+1} + \sqrt{2x-1}}$$

例 如何计算下列函数值才比较精确

$$(1) \frac{1}{1+2x} - \frac{1}{1+x} \quad \text{对 } |x| \ll 1 \quad (2) \sqrt{x + \frac{1}{x}} - \sqrt{x - \frac{1}{x}} \quad \text{对 } |x| \gg 1$$

解 (1) 要使计算准确, 应避免两个相近的数相减
故变换所给公式

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

(2) 要使计算准确, 应避免两个相近的数相减

故变换所给公式

$$\sqrt{x + \frac{1}{x}} - \sqrt{x - \frac{1}{x}} = \frac{(\sqrt{x + \frac{1}{x}} - \sqrt{x - \frac{1}{x}})(\sqrt{x + \frac{1}{x}} + \sqrt{x - \frac{1}{x}})}{\sqrt{x + \frac{1}{x}} + \sqrt{x - \frac{1}{x}}} = \frac{\frac{2}{x}}{\sqrt{x + \frac{1}{x}} + \sqrt{x - \frac{1}{x}}}$$

第二章 非线性方程求解

• 区间对分法（二分法）

1. 计算 $f(x)$ 在有解区间 $[a, b]$ 端点处的值, $f(a)$, $f(b)$ 。
 2. 计算 $f(x)$ 在区间中点处的值 $f(x_1)$ 。
 3. 判断若 $f(x_1) = 0$, 则 x_1 即是根, 否则检验:
 - (1) 若 $f(x_1)$ 与 $f(a)$ 异号, 则知解位于区间 $[a, x_1]$,
 $b_1 = x_1, a_1 = a$;
 - (2) 若 $f(x_1)$ 与 $f(a)$ 同号, 则知解位于区间 $[x_1, b]$,
 $a_1 = x_1, b_1 = b$ 。
- 反复执行步骤2、3, 便可得到一系列有根区间:

$$(a, b), (a_1, b_1), \dots, (a_k, b_k), \dots$$

4、当 $b_{n+1} - a_{n+1} < \varepsilon$ 时

5、则 $x_{n+1} = \frac{1}{2}(a_k + b_k)$ 即为根的近似

先验误差估计:

$$|x_{k+1} - x^*| \leq \frac{1}{2^{k+1}}(b-a) \quad (k=1,2,\cdots)$$

理论基础:

定理1: 设函数 $f(x)$ 在区间 $[a, b]$ 上连续, 如果 $f(a) \cdot f(b) < 0$, 则方程 $f(x) = 0$ 在 $[a, b]$ 内至少有一实根 x^* 。

例 试用对分区间法求方程

$$x^5 + 3x - 1 = 0$$

在 $[0, 0.5]$ 内的根,要求误差不超过 $\frac{1}{2} \times 10^{-2}$.

[解] 令 $f(x) = x^5 + 3x - 1$ 则

$$f(0) = -1 < 0, f(0.5) = 0.531 > 0 \quad f'(x) = 5x^4 + 3 > 0,$$

故在 $[0, 0.5]$ 内 $f(x) = x^5 + 3x - 1 = 0$ 有唯一实根.

为达到要求

$$n \geq \frac{\ln \frac{(b-a)}{\varepsilon}}{\ln 2} - 1$$

即

$$n \geq \frac{\ln \frac{(0.5-0)}{\frac{1}{2} \times 10^{-2}}}{\ln 2} - 1 = \frac{\ln 10^2}{\ln 2} - 1 = 5.64$$

取 $n = 6$

计算结果见下表。

$$f(x) = x^5 + 3x - 1$$

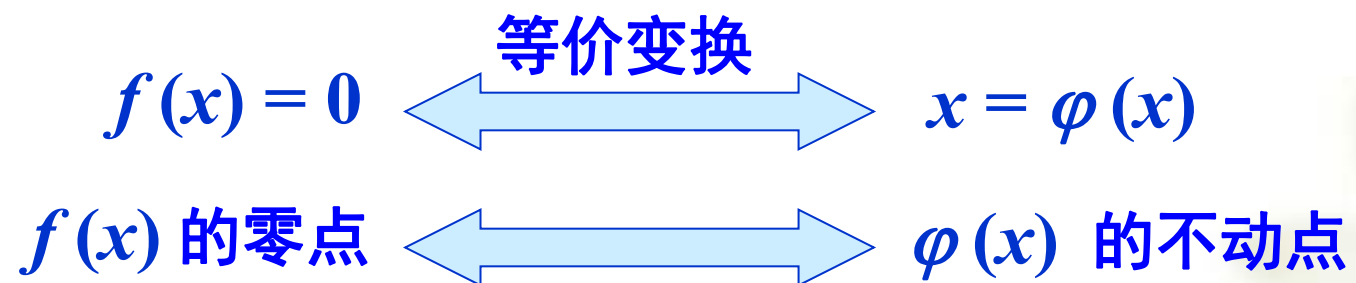
n	a_n	b_n	x_n	$f(x_n)$
0	0	0.5	0.25	-0.249023
1	0.25	0.5	0.375	0.1324158
2	0.25	0.375	0.3125	-0.0595197
3	0.3125	0.375	0.34375	0.0360497
4	0.3125	0.34375	0.328125	-0.0118214
5	0.328125	0.34375	0.3359375	0.01209101
6	0.328125	0.3359375	0.33203125	0.00012923

故 $x_6 = 0.33203125$ 即为符合精度要求的解.

不动点迭代

□ 基本思想

- 构造 $f(x) = 0$ 的一个等价方程: $x = \varphi(x)$



不动点迭代

□ 具体过程

- 任取一个迭代初始值 x_0 ，计算

$$x_{k+1} = \varphi(x_k) \quad k = 0, 1, 2, \dots$$

得到一个迭代序列： $x_0, x_1, x_2, \dots, x_n, \dots$

几何含义：求曲线 $y = \varphi(x)$ 与直线 $y = x$ 的交点

□ 收敛性分析

设 $\varphi(x)$ 连续, 若 $\{x_k\}_{k=0}^{\infty}$ 收敛, 即 $\lim_{k \rightarrow \infty} x_k = x^*$, 则

$$\lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} \varphi(x_k) = \varphi\left(\lim_{k \rightarrow \infty} x_k\right)$$

→ $x^* = \varphi(x^*)$ 即 $f(x^*) = 0$

性质: 若 $\lim_{k \rightarrow \infty} x_k = x^*$, 则不动点迭代收敛, 且 x^* 是 $f(x)=0$ 的解; 否则迭代法发散。

不动点迭代的收敛性

定理：设 $\varphi(x) \in C[a,b]$ 且满足

(1) 对任意的 $x \in [a,b]$ 有 $\varphi(x) \in [a,b]$

(2) 存在常数 $0 < L < 1$ ，使得任意的 $x, y \in [a,b]$ 有

$$|\varphi(x) - \varphi(y)| \leq L|x - y|$$

则对任意初始值 $x_0 \in [a,b]$ ，不动点迭代 $x_{k+1} = \varphi(x_k)$ 收敛，且

$$|x_k - x^*| \leq \frac{L}{1-L} |x_k - x_{k-1}| \leq \frac{L^k}{1-L} |x_1 - x_0|$$

定理：若 $\varphi(x) \in C^1[a,b]$ 且对任意 $x \in [a,b]$ 有

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

则上述定理中的结论成立。

收敛性结论表明：收敛性与初始值的选取无关

全局收敛

例：求 $f(x) = x^3 - x - 1 = 0$ 在区间 $[1, 2]$ 中的根

(1) $\varphi(x) = x^3 - 1 \longrightarrow 0 \leq \varphi(x) \leq 7 \quad (x \in [1, 2])$

$\varphi'(x) = 3x^2 \longrightarrow |\varphi'(x)| > 1$



(2) $\varphi(x) = \sqrt[3]{x+1} \longrightarrow 1 \leq \varphi(x) \leq 2 \quad (x \in [1, 2])$

$\varphi'(x) = \frac{1}{3}(x+1)^{-\frac{2}{3}} \longrightarrow |\varphi'(x)| \leq \frac{1}{3}\sqrt[3]{0.25} < 1$



局部收敛

定义：设 x^* 是 $\varphi(x)$ 的不动点，若存在 x^* 的某个 δ -邻域 $U(x^*) = [x^* - \delta, x^* + \delta]$ ，对任意 $x_0 \in U(x^*)$ ，不动点迭代

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

产生的点列都收敛到 x^* ，则称该迭代局部收敛。

定理：设 x^* 是 $\varphi(x)$ 的不动点，若 $\varphi'(x)$ 在 x^* 的某个邻域内连续，且

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

则不动点迭代 $x_{k+1} = \varphi(x_k)$ 局部收敛

定义：设迭代 $x_{k+1} = \varphi(x_k)$ 收敛到 $\varphi(x)$ 的不动点 x^* ，记 $e_k = x_k - x^*$ ，若

$$\lim_{k \rightarrow \infty} \frac{|e_{k+1}|}{|e_k|^r} = C$$

其中常数 $C > 0$ ，则称该迭代为 r 阶收敛。

- (1) 当 $r=1$ 时称为线性收敛，此时 $C < 1$
- (2) 当 $r=2$ 时称为二次收敛，或平方收敛
- (3) 当 $r > 1$ 时称为超线性收敛

- 二分法是线性收敛的
- 若 $\varphi'(x^*) \neq 0$ ，则不动点迭代 $x_{k+1} = \varphi(x_k)$ 线性收敛

定理：设 x^* 是 $\varphi(x)$ 的不动点，若 $\varphi^{(p)}(x)$ 在 x^* 的某邻域内连续，且

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

$$\varphi^{(p)}(x^*) \neq 0$$

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

例：求 $f(x) = x^2 - 3 = 0$ 的正根 $x^* = \sqrt{3}$

(1) $\varphi(x) = x^2 - 3 + x \Rightarrow \varphi'(x^*) = 2\sqrt{3} + 1 > 1$

(2) $\varphi(x) = x - \frac{x^2 - 3}{4} \Rightarrow \varphi'(x^*) = 1 - \frac{\sqrt{3}}{2} \approx 0.314 < 1$

(3) $\varphi(x) = \frac{1}{2} \left(x + \frac{3}{x} \right) \Rightarrow \begin{aligned} \varphi'(x^*) &= 0 \\ \varphi''(x^*) &= \frac{2}{\sqrt{3}} \neq 0 \end{aligned}$

局部收敛

二次收敛

例 用简单迭代法求方程 $e^x - 4x = 0$ 在 $[0,1]$ 内的根
精确到3位有效数字.

[解] $f(x) = e^x - 4x$, $f(0) = 1 > 0$, $f(1) = e - 4 < 0$

$f(x)$ 在 $[0,1]$ 内连续, $f'(x) = e^x - 4 < 0, x \in [0,1]$ 故 $f(x) = 0$
在 $[0,1]$ 内有唯一实根.

$e^x - 4x = 0$ 的等价方程为 $x = \frac{1}{4}e^x$

$$\varphi(x) = \frac{1}{4}e^x, |\varphi'(x)| = \frac{1}{4}e^x \leq \frac{1}{4}e < 1, x \in [0,1]$$

又当 $x \in [0,1], \varphi(x) \in [0,1]$

故 $x_{n+1} = \frac{1}{4}e^{x_n}$ 对 $\forall x_0 \in [0,1]$ 均收敛. 取

$x_0 = 0.5$ 计算得

$$x_{n+1} = \frac{1}{4} e^{x_n} \quad x_0 = 0.5$$

$$\begin{aligned} x_1 &= 0.4122, & x_2 &= 0.3775, & x_3 &= 0.3675, \\ x_4 &= 0.3600, & x_5 &= 0.3583, & x_6 &= 0.3571, \\ x_7 &= 0.3575 \end{aligned}$$

故, $x^* = 0.358$

Newton 法

□ 基本思想

将非线性方程线性化

- 设 x_k 是 $f(x)=0$ 的近似根, 将 $f(x)$ 在 x_k 处 Taylor 展开

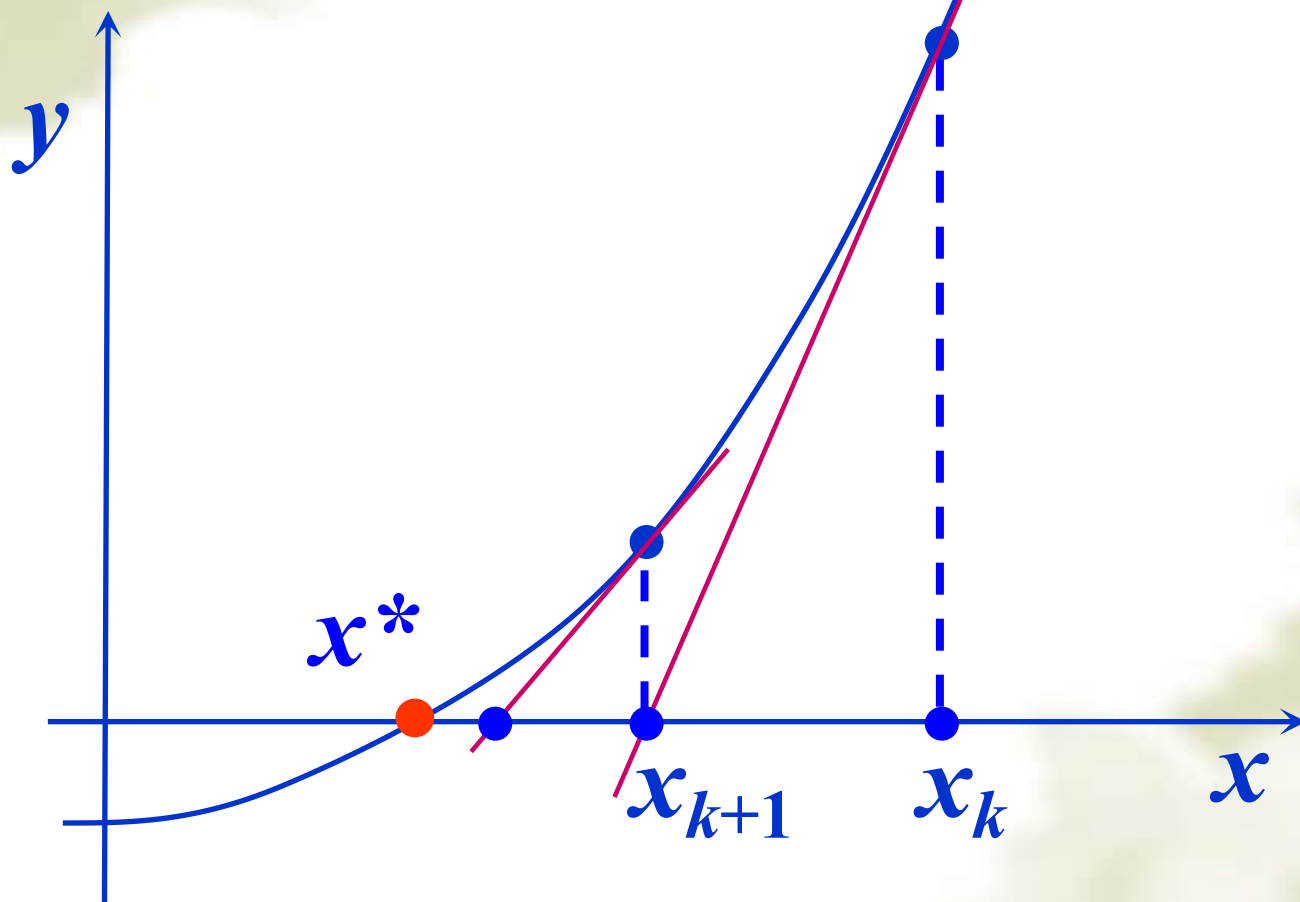
$$\begin{aligned} f(x) &= f(x_k) + f'(x_k)(x - x_k) + \frac{f''(\xi)}{2!}(x - x_k)^2 \\ &\approx f(x_k) + f'(x_k)(x - x_k) \triangleq P(x) \end{aligned}$$

令: $P(x) = 0$

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

条件: $f'(x) \neq 0$

Newton 法



例7 用Newton法求方程 $f(x) = x^3 + 2x^2 + 10x - 20 = 0$

在(1,2)的根，取初值 $x_0 = 1$

[解] 因为 $f(1) = -7, f(2) = 16$

$$f'(x) = 3x^2 + 4x + 10 > 0, \quad x \in [1, 2]$$

故在[1,2]内方程有唯一实根

根据 Newton迭代

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

得牛顿迭代公式为

$$x_{n+1} = x_n - \frac{x_n^3 + 2x_n^2 + 10x_n - 20}{3x_n^2 + 4x_n + 10}$$

取初值 $x_0 = 1$ **得**

$$x_{n+1} = x_n - \frac{x_n^3 + 2x_n^2 + 10x_n - 20}{3x_n^2 + 4x_n + 10} \quad x_0 = 1$$

代入初值得

$$x_1 = 1.411764706, \quad x_2 = 1.369336471$$

$$x_3 = 1.368808189, \quad x_4 = 1.368808108,$$

$$x_5 = 1.368808108$$

$$|x_5 - x_4| < \frac{1}{2} \times 10^{-9}$$

故取

$$x \approx x_5 = 1.368808108$$

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

● 迭代函数

$k = 0, 1, 2, \dots$

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

$$\Rightarrow \varphi'(x^*) = 0, \quad \varphi''(x^*) = \frac{f''(x^*)}{2f'(x^*)}$$

牛顿法至少二阶局部收敛

$$\lim_{k \rightarrow \infty} \frac{x_{k+1} - x^*}{(x_k - x^*)^2} = \frac{\varphi''(x^*)}{2!} = \frac{f''(x^*)}{2f'(x^*)}$$

那牛顿法的全局部收敛性呢？

例 证明牛顿迭代法对于重根为线性收敛。

例3. 设 x^* 是方程 $f(x)=0$ 的 $m(\geq 2)$ 重根,证明迭代法

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

为线性收敛

证明: 因为 x^* 是方程 $f(x)=0$ 的 m 重根,故

$$f(x) = (x - x^*)^m g(x) \quad \text{且 } g(x^*) \neq 0, m \geq 2$$

$$\text{所以 } f'(x) = m(x - x^*)^{m-1} g(x) + (x - x^*)^m g'(x)$$

$$\begin{aligned} x_{k+1} &= x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{(x_k - x^*)^m g(x_k)}{m(x_k - x^*)^{m-1} g(x_k) + (x_k - x^*)^m g'(x_k)} \\ &= x_k - \frac{(x_k - x^*) g(x_k)}{m g(x_k) + (x_k - x^*) g'(x_k)} \end{aligned}$$

$$x_{k+1} - x^* = (x_k - x^*) \left(1 - \frac{g(x_k)}{mg(x_k) + (x_k - x^*)g'(x_k)} \right)$$

$$\lim_{k \rightarrow \infty} \frac{x_{k+1} - x^*}{x_k - x^*} = \lim_{k \rightarrow \infty} \left(1 - \frac{g(x_k)}{mg(x_k) + (x_k - x^*)g'(x_k)} \right) = 1 - \frac{1}{m}$$

$$m \geq 2 \text{ 时, } 1 - \frac{1}{m} > 0 \quad \text{由定义1}$$

该迭代法对 $m(\geq 2)$ 重根是线性收敛的

第三章 线性方程组的直接方法

Gauss 消去法

例：直接法解线性方程组

$$\begin{cases} x_1 - 2x_2 + 2x_3 = -2 \\ 2x_1 - 3x_2 - 3x_3 = 4 \\ 4x_1 + x_2 + 6x_3 = 3 \end{cases}$$

解：

$$(A, b) = \left[\begin{array}{ccc|c} 1 & -2 & 2 & -2 \\ 2 & -3 & -3 & 4 \\ 4 & 1 & 6 & 3 \end{array} \right] \rightarrow \left[\begin{array}{ccc|c} 1 & -2 & 2 & -2 \\ 0 & 1 & -7 & 8 \\ 0 & 9 & -2 & 11 \end{array} \right]$$

$$\left[\begin{array}{ccc|c} 1 & -2 & 2 & -2 \\ 0 & 1 & -7 & 8 \\ 0 & 0 & 61 & -61 \end{array} \right] \rightarrow \begin{cases} x_3 = -1 \\ x_2 = 8 + 7x_3 = 1 \\ x_1 = -2 + 2x_2 - 2x_3 = 2 \end{cases}$$

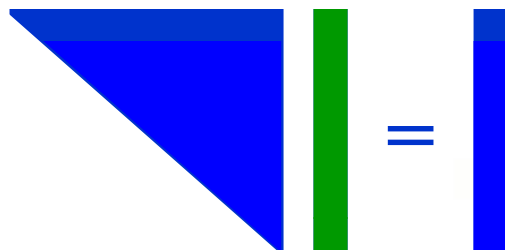
Gauss 消去法

考虑 n 阶线性方程组：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \xrightarrow{\text{矩阵形式}} Ax = b$$

高斯消去法的主要思路：

将系数矩阵 A 化为上三角矩阵，然后回代求解。



计算 LU 分解

利用矩阵乘法直接计算 LU 分解

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

$L \times U = A$

① 比较等式两边的第一行得：

$$u_{1j} = a_{1j} \quad (j=1, \dots, n) \quad \leftarrow U \text{ 的第一行}$$

比较等式两边的第一列得：

$$l_{i1} = a_{i1} / u_{11} \quad (i=2, \dots, n) \quad \leftarrow L \text{ 的第一列}$$

② 比较等式两边的第二行得：

$$u_{2j} = a_{2j} - l_{21}u_{1j} \quad (j=2, \dots, n) \quad \leftarrow U \text{ 的第二行}$$

比较等式两边的第二列得：

$$l_{i2} = (a_{i2} - l_{i1}u_{12}) / u_{22} \quad (i=3, \dots, n) \quad \leftarrow L \text{ 的第二列}$$

LU 分解算法

例 求下列矩阵的 LU 分解

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix}$$

解： 设

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

$$\Rightarrow u_{11} = 3, u_{12} = -1, u_{13} = 2$$

向量范数

■ 常见的向量范数

① **1-范数** $\|x\|_1 = \sum_{i=1}^n |x_i|$

② **2-范数** $\|x\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}$

③ **无穷范数（最大范数）**

$$\|x\|_{\infty} = \max_{1 \leq i \leq n} |x_i|$$

范数性质

■ 范数的性质

(1) 等价性

设 $\|\cdot\|_s$ 和 $\|\cdot\|_t$ 是 \mathbf{R}^n 上的任意两个范数，则存在常数 c_1 和 c_2 ，使得对任意的 $x \in \mathbf{R}^n$ 有

$$c_1 \|x\|_s \leq \|x\|_t \leq c_2 \|x\|_s$$

范数性质

(2) Cauchy-Schwarz 不等式

$$|(x, y)| \leq \|x\|_2 \times \|y\|_2$$

(3) 向量序列的收敛性

$$\lim_{k \rightarrow \infty} x^{(k)} = x^* \iff \lim_{k \rightarrow \infty} \|x^{(k)} - x^*\| = 0$$

● 矩阵的谱: $\sigma(A) = \{ A \text{ 的所有特征值} \}$

● 矩阵的谱半径: $\rho(A) = \max_{\lambda \in \sigma(A)} \{ |\lambda| \}$

矩阵范数

■ 常见的矩阵范数

(1) F-范数 (Frobenious 范数)

$$\|A\|_F = \left(\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 \right)^{\frac{1}{2}}$$

(2) 算子范数 (从属范数、诱导范数)

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

其中 $\|\cdot\|$ 是 R^n 上的任意一个范数

算子范数

■ 常见的算子范数

① 1-范数（列范数）

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$$

② 2-范数（谱范数）

$$\|A\|_2 = \sqrt{\rho(A^T A)}$$

③ 无穷范数（行范数）

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

例： 设 $A = \begin{pmatrix} 1 & -2 \\ -3 & 4 \end{pmatrix}$ 计算 $\|A\|_1, \|A\|_2, \|A\|_\infty, \|A\|_F$

矩阵范数性质

■ 矩阵范数的性质

(1) 等价性： 设 $\|\cdot\|_s$ 和 $\|\cdot\|_t$ 是 $\mathbf{R}^{n \times n}$ 上的任意两个矩阵范数，则存在常数 c_1 和 c_2 ，使得对任意的 $A \in \mathbf{R}^{n \times n}$ 有

$$c_1 \|A\|_s \leq \|A\|_t \leq c_2 \|A\|_s$$

(2) 若 A 是对称矩阵，则 $\rho(A) = \|A\|_2$

算子范数性质

■ 算子范数的性质

定理：设 $\|\cdot\|$ 是 R^n 上的任一向量范数，其对应的算子范数也记为 $\|\cdot\|$ ，则有

$$\|Ax\| \leq \|A\| \times \|x\|$$

定理：设 $\|\cdot\|$ 是任一算子范数，则 $\rho(A) \leq \|A\|$

稳定性理论分析

□ 理论分析:

(1) 由于右端项的扰动而引起的解的变化

设 $A(x + \delta x) = b + \delta b \rightarrow \delta x = A^{-1} \cdot \delta b$

$\rightarrow \left. \begin{array}{l} \|\delta x\| \leq \|A^{-1}\| \cdot \|\delta b\| \\ \text{又 } \|b\| \leq \|A\| \cdot \|x\| \end{array} \right\} \rightarrow \frac{\|\delta x\|}{\|x\|} \leq \boxed{\|A\| \cdot \|A^{-1}\|} \frac{\|\delta b\|}{\|b\|}$

$Ax = b$ 的条件数
矩阵 A 的条件数

矩阵条件数

$$\mathbf{Cond}(A) = \|A\| \cdot \|A^{-1}\|$$

- 条件数与范数有关，常用的有无穷范数和2-范数

$$\mathbf{Cond}(A)_{\infty} = \|A^{-1}\|_{\infty} \|A\|_{\infty}$$

$$\mathbf{Cond}(A)_2 = \|A\|_2 \|A^{-1}\|_2$$

- $\mathbf{Cond}(A)_2$ 称为谱条件数，当 A 对称时有

$$\mathbf{Cond}(A)_2 = \|A\|_2 \|A^{-1}\|_2 = \frac{\max_{1 \leq i \leq n} |\lambda_i|}{\min_{1 \leq i \leq n} |\lambda_i|}$$

举例

例： $A = \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix}$ 计算 $\text{Cond}(A)_\infty$ 和 $\text{Cond}(A)_2$

解：

$$A^{-1} = \begin{bmatrix} 10001 & -10000 \\ -10000 & 10000 \end{bmatrix}$$

第四章 线性方程组的迭代方法

矩阵分裂迭代法

矩阵分裂迭代法基本思想

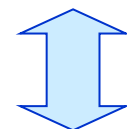
A 的一个
矩阵分裂

$$Ax = b$$

$$A = M - N$$

M 非奇异

$$Mx = Nx + b$$



$$x = M^{-1}Nx + M^{-1}b$$

给定一个初始向量 $x^{(0)}$, 可得 迭代格式

$$x^{(k+1)} = Bx^{(k)} + f \quad k = 0, 1, 2, \dots$$

其中 $B = M^{-1}N$ 称为迭代矩阵



收敛性分析

$$x^{(k+1)} = Bx^{(k)} + f$$

基本收敛定理

定理：对任意初始向量 $x^{(0)}$ ，上述迭代格式收敛的充要条件是

$$\rho(B) < 1$$

定理：若存在算子范数 $\|\cdot\|$ ，使得 $\|B\| < 1$ ，对任意的初始向量 $x^{(0)}$ ，上述迭代格式收敛。

例：考虑迭代法 $x^{(k+1)} = Bx^{(k)} + f$ 的收敛性，其中

$$B = \begin{bmatrix} 0.9 & 0 \\ 0.3 & 0.8 \end{bmatrix}$$

充分条件

Jacobi 迭代

考虑线性方程组

$$Ax = b$$

其中 $A=(a_{ij})_{n \times n}$ 非奇异，且对角线元素全不为 0。

● 将 A 分裂成 $A = D - L - U$ ，其中

$$D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$$

$$L = \begin{bmatrix} 0 & & & \\ -a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ & 0 & \ddots & \vdots \\ & & \ddots & -a_{n-1,n} \\ & & & 0 \end{bmatrix}$$

Jacobi 迭代

Jacobi 迭代

令 $M = D$, $N = L + U$, 可得 雅可比 (Jacobi) 迭代方法

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b} \quad k = 0, 1, 2, \dots$$

● 迭代矩阵记为: $\mathbf{J} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$

● 分量形式:

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

$$i = 1, 2, \dots, n, \quad k = 0, 1, 2, \dots$$



Gauss-Seidel 迭代

$$\begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)})/a_{11} \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)})/a_{22} \\ \vdots \\ x_n^{(k+1)} = (b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots - a_{nn}x_n^{(k)})/a_{nn} \end{cases}$$

- 在计算 $x_i^{(k+1)}$ 时，如果用 $x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}$ 代替 $x_1^{(k)}, \dots, x_{i-1}^{(k)}$ ，则可能会得到更好的收敛效果。

$$\begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)})/a_{11} \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)})/a_{22} \\ \vdots \\ x_n^{(k+1)} = (b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \cdots - a_{nn}x_n^{(k+1)})/a_{nn} \end{cases}$$

Gauss-Seidel 迭代

写成矩阵形式:

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1} \left(\mathbf{b} + \mathbf{L}\mathbf{x}^{(k+1)} + \mathbf{U}\mathbf{x}^{(k)} \right)$$

可得

$$\mathbf{x}^{(k+1)} = (\mathbf{D} - \mathbf{L})^{-1} \mathbf{U}\mathbf{x}^{(k)} + (\mathbf{D} - \mathbf{L})^{-1} \mathbf{b} \quad k = 0, 1, 2, \dots$$

此迭代方法称为 高斯-塞德尔 (Gauss-Seidel) 迭代法

● 迭代矩阵记为: $\mathbf{G} = (\mathbf{D} - \mathbf{L})^{-1} \mathbf{U}$

$$\begin{aligned} \mathbf{x}^{(k+1)} &= (\mathbf{D} - \mathbf{L})^{-1} (\mathbf{D} - \mathbf{L} - \mathbf{A})\mathbf{x}^{(k)} + (\mathbf{D} - \mathbf{L})^{-1} \mathbf{b} \\ &= \left(\mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1} \mathbf{A} \right) \mathbf{x}^{(k)} + (\mathbf{D} - \mathbf{L})^{-1} \mathbf{b} = \mathbf{x}^{(k)} + (\mathbf{D} - \mathbf{L})^{-1} (\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}) \end{aligned}$$



收敛性

收敛性定理

- Jacobi 迭代收敛的充要条件 $\rho(J) < 1$
- G-S 迭代收敛的充要条件 $\rho(G) < 1$

- Jacobi 迭代收敛的充分条件 $\|J\| < 1$
- G-S 迭代收敛的充分条件 $\|G\| < 1$

定理：若 A 对称，且对角线元素均大于 0，则

- (1) Jacobi 迭代收敛的充要条件是 A 与 $2D-A$ 均正定；
- (2) G-S 迭代收敛的充要条件是 A 正定。

举例

例：设 $A = \begin{bmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{bmatrix}$ ，给出 Jacobi 和 G-S 收敛的充要条件

解：A 对称，且对角线元素均大于 0，故

(1) Jacobi 收敛的充要条件是 A 和 $2D-A$ 均正定

(2) G-S 收敛的充要条件是 A 正定

$$\begin{aligned} A \text{ 正定} &\iff D_1 = 1 > 0, D_2 = 1 - a^2 > 0, D_3 = (1 - a)^2(1 + 2a) > 0 \\ &\iff -0.5 < a < 1 \end{aligned}$$

$$\begin{aligned} 2D-A \text{ 正定} &\iff D_1 = 1 > 0, D_2 = 1 - a^2 > 0, D_3 = (1 + a)^2(1 - 2a) > 0 \\ &\iff -0.5 < a < 0.5 \end{aligned}$$



Jacobi 收敛的充要条件是： $-0.5 < a < 0.5$

G-S 收敛的充要条件是： $-0.5 < a < 1$

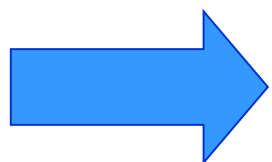
举例

解法二： Jacobi 的迭代矩阵为

$$J = \begin{bmatrix} 0 & -a & -a \\ -a & 0 & -a \\ -a & -a & 0 \end{bmatrix}$$

设 λ 是 J 的特征值，则由 $\det(\lambda I - J) = 0$ 可得

$$(\lambda - a)^2 (\lambda + 2a) = 0$$



Jacobi 收敛的充要条件是 $\rho(J) < 1 \Leftrightarrow |\lambda| < 1$,
即 $-0.5 < a < 0.5$

极小化方法

设 A 对称正定，求解的线性方程组为

$$Ax = b \quad (1)$$

其中 $A = (a_{ij}) \in R^{n \times n}$, $x = (x_1, x_2, \dots, x_n)^T$, $b = (b_1, b_2, \dots, b_n)^T$

对应的二次函数 $\varphi: R^{n \times n} \rightarrow R$ ，称为模函数，定义为

$$\varphi(x) = \frac{1}{2}(Ax, x) - (b, x) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j - \sum_{i=1}^n b_i x_i \quad (2)$$

例： 设 $A = \begin{bmatrix} 1 & 2 \\ 2 & 6 \end{bmatrix}$, $b = \begin{bmatrix} 4 \\ 10 \end{bmatrix}$

$$\varphi(x) = \frac{1}{2}(x_1^2 + 6x_2^2 + 4x_1x_2) - (4x_1 + 10x_2)$$

(2) 对一切 $x, y \in R^n, \alpha \in R$

$$\begin{aligned}\varphi(x + \alpha y) &= \frac{1}{2}(A(x + \alpha y), x + \alpha y) - (b, x + \alpha y) \\&= \frac{1}{2}(Ax, x) - (b, x) + \alpha(Ax, y) - \alpha(b, y) + \frac{\alpha^2}{2}(Ay, y) \\&= \varphi(x) + \alpha(Ax - b, y) + \frac{\alpha^2}{2}(Ay, y)\end{aligned}$$

$$\varphi(x) = \frac{1}{2}(Ax, x) - (b, x)$$

定理1 设 $A = (a_{ij})_{n \times n}$ 为**实对称正定矩阵**, $b, x \in R^n$,
则 x 使二次函数

$$f(x) = \frac{1}{2}(Ax, x) - (b, x)$$

取极小值 $\Leftrightarrow x$ 是线性方程组 $Ax = b$ 的解。

求二次函数 $\varphi(x)$ 极小值点的一般方法是：

构造一个向量序列 $\{x^{(k)}\}$ ，使 $\varphi(x^{(k)}) \rightarrow \min \varphi(x)$

可以采取以下方法：

(1) 任取一个初始向量 $x^{(0)}$ ，

(2) 构造迭代格式

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad (k = 0, 1, \dots)$$

其中 $p^{(k)}$ 是搜索方向， α_k 是搜索步长，

(3) 选择 $p^{(k)}$ 和 α_k 使得

$$\varphi(x^{(k+1)}) = \varphi(x^{(k)} + \alpha_k p^{(k)}) < \varphi(x^{(k)})$$

则当 $k \rightarrow \infty$ 时，有 $\varphi(x^{(k)}) \rightarrow \varphi(x^*) = \min_{x \in R^n} \varphi(x)$

(4) 算出误差限 ε ，直到

$$\|x^{(k)} - x^{(k-1)}\| < \varepsilon \text{ 或 } \|r^{(k)}\| = \|b - Ax^{(k)}\| < \varepsilon$$

迭代为止。



最速下降法

最速下降算法：

(1) 选取 $x^{(0)} \in R^n$

(2) 对 $k = 0, 1, 2, \dots$

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

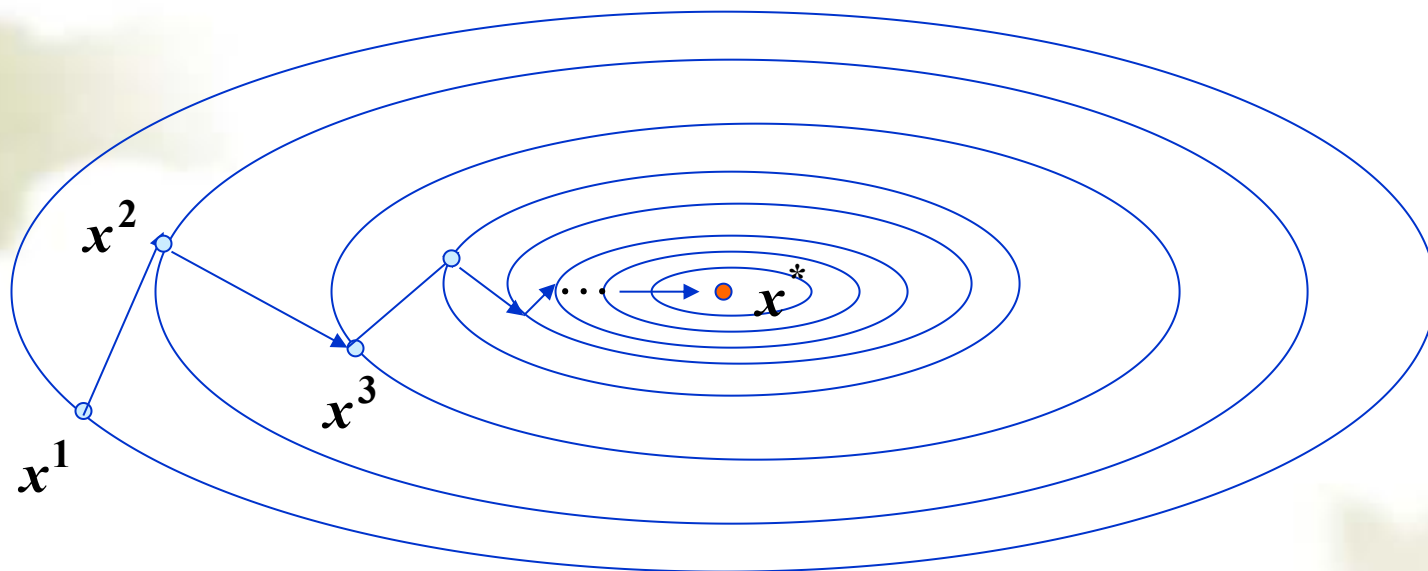
$$\alpha_k = \frac{(r^{(k)}, r^{(k)})}{(Ar^{(k)}, r^{(k)})}$$

$$x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}$$

(3) 当 $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$ 时，终止迭代。

$$\alpha = -\frac{(r^{(k)}, p^{(k)})}{(Ap^{(k)}, p^{(k)})}$$

目标函数为二次函数，其等值面为椭球面。



注 最速下降方向反映了目标函数的一种局部性质。它只是局部目标函数值下降最快的方向。
最速下降法是线性收敛的算法。

不难验证，相邻两次的搜索方向是正交的，即

$$(r^{(k+1)}, r^{(k)}) = 0$$

容易看到, $\{\varphi(x^{(k)})\}$ 是单调下降有界序列, 它存在极限,

可以证明 $\lim_{k \rightarrow \infty} x^{(k)} = x^* = A^{-1}b$

而且 $\|x^{(k)} - x^*\|_A \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \right)^k \|x^{(0)} - x^*\|_A$

其中 λ_1, λ_n 分别是对称正定阵 A 的最大、最小特征值,

$$\|u\|_A = (Au, u)^{1/2}$$

当 $\lambda_1 \gg \lambda_n$ 时, 收敛是很慢的,

当 $\|r^{(k)}\|$ 很小时, 因舍入误差的影响, 计算将出现不稳定现象。

共轭梯度法

定义 A 对称正定, 若 R^n 中向量组 $\{p^{(0)}, \dots, p^{(l)}\}$ 满足

$$(Ap^{(i)}, p^{(j)}) = 0, \quad i \neq j$$

则称它为 R^n 中的一个 A -共轭向量组, 或称 A -正交向量组。

注:

- 1、当 $l < n$ 时, 不含零向量的 A -共轭向量组线性无关;
- 2、当 $A = I$ 时, A -共轭性质就是一般的正交性;
- 3、给了一组线性无关的向量, 可以按 $Schmidt$ 正交化的方法得到对应的 A -共轭向量组。

原始的CG算法：

$$(1) x^{(0)} \in R^{(n)}$$

$$(2) r^{(0)} = b - Ax^{(0)}, p^{(0)} = r^{(0)}$$

$$(3) k = 0, 1, \dots,$$

$$\alpha_k = \frac{(r^{(k)}, r^{(k)})}{(Ap^{(k)}, p^{(k)})},$$

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)},$$

$$r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)},$$

$$\beta_k = -\frac{(r^{(k+1)}, Ap^{(k)})}{(p^{(k)}, Ap^{(k)})},$$

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)},$$

改进的CG算法：

$$(1) x^{(0)} \in R^{(n)}$$

$$(2) r^{(0)} = b - Ax^{(0)}, p^{(0)} = r^{(0)}$$

$$(3) k = 0, 1, \dots,$$

$$\alpha_k = \frac{(r^{(k)}, r^{(k)})}{(Ap^{(k)}, p^{(k)})},$$

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)},$$

$$r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)},$$

$$\beta_k = \frac{(r^{(k+1)}, r^{(k+1)})}{(r^{(k)}, r^{(k)})},$$

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)},$$

注：

- (1) 剩余向量相互正交，而 R^n 中至多有 n 个相互正交的非零向量，所以 $r^{(0)}, r^{(1)}, \dots, r^{(n)}$ 中至少有一个向量为零。若 $r^{(k)} = 0$ ，则 $x^{(k)} = x^*$ 。
- (2) 实际计算中，由于舍入误差的影响， n 步内得不到准确解，故还需继续迭代。一般因 $p^{(0)}, p^{(1)}, \dots, p^{(n-1)}$ 是一组 A -共轭向量组，继续迭代时，要取 $x^{(0)} = x^{(n)}$ 。
- (3) 由误差估计式

$$\|x^{(k)} - x^*\|_A \leq 2 \left[\frac{\sqrt{\text{cond}(A)_2} - 1}{\sqrt{\text{cond}(A)_2} + 1} \right]^k \|x^{(0)} - x^*\|_A$$

当 A 的条件数很小时，共轭斜量法收敛很快，但当 A 是病态严重的矩阵时，共轭斜量法收敛速度很慢。可采用预处理技术，降低 A 的条件数。

非线性方程组的牛顿法

$$\min F(x, y) \Rightarrow \begin{cases} \frac{\partial f(x, y)}{\partial x} = 0 \\ \frac{\partial f(x, y)}{\partial y} = 0 \end{cases} \Rightarrow \begin{cases} f_1(x, y) = 0 \\ f_2(x, y) = 0 \end{cases}$$

在点 (x_0, y_0) 作二元Taylor展开,

$$\begin{cases} f_1(x, y) = f_1(x_0, y_0) + \frac{\partial f_1(x_0, y_0)}{\partial x}(x - x_0) + \frac{\partial f_1(x_0, y_0)}{\partial y}(y - y_0) \approx 0 \\ f_2(x, y) = f_2(x_0, y_0) + \frac{\partial f_2(x_0, y_0)}{\partial x}(x - x_0) + \frac{\partial f_2(x_0, y_0)}{\partial y}(y - y_0) \approx 0 \end{cases}$$

即

$$\begin{cases} \frac{\partial f_1(x_0, y_0)}{\partial x}(x - x_0) + \frac{\partial f_1(x_0, y_0)}{\partial y}(y - y_0) \approx -f_1(x_0, y_0) \\ \frac{\partial f_2(x_0, y_0)}{\partial x}(x - x_0) + \frac{\partial f_2(x_0, y_0)}{\partial y}(y - y_0) \approx -f_2(x_0, y_0) \end{cases}$$

令

$$J(X_0) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} \bigg|_{\substack{x=x_0 \\ y=y_0}}$$

若

$$|J(X_0)| = \begin{vmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{vmatrix} \bigg|_{\substack{x=x_0 \\ y=y_0}} \neq 0$$

设

$$\Delta x = x - x_0, \Delta y = y - y_0,$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, X_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \Delta X = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad B = \begin{bmatrix} -f_1(x_0, y_0) \\ -f_2(x_0, y_0) \end{bmatrix}$$

则求解非线性方程组的牛顿迭代格式

$$J(X) \cdot \Delta X = B \quad \Delta X = J^{-1}(X) \cdot B$$

例 解非线性方程组

$$\begin{cases} f_1(x, y) = 4 - x^2 - y^2 = 0 \\ f_2(x, y) = 1 - e^x - y = 0 \end{cases} \quad \text{initial values} \quad \begin{cases} x_0 = 1 \\ y_0 = -1.7 \end{cases}$$

解 Jacobi矩阵:

$$J(x, y) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} -2x & -2y \\ -e^x & -1 \end{pmatrix}$$

$$J(x_0, y_0) = \begin{pmatrix} -2x & -2y \\ -e^x & -1 \end{pmatrix}_{(x_0, y_0)} = \begin{pmatrix} -2 & -3.4 \\ -2.71828 & -1 \end{pmatrix}$$

$$\begin{pmatrix} f_1(x_0, y_0) \\ f_2(x_0, y_0) \end{pmatrix} = \begin{pmatrix} 0.11 \\ -0.01828 \end{pmatrix}$$

$$\begin{cases} -2\Delta x + 3.4\Delta y = -0.11 \\ -2.71828\Delta x - \Delta y = -0.01828 \end{cases} \Rightarrow \begin{cases} \Delta x = 0.004256 \\ \Delta y = -0.029846 \end{cases}$$

$$\therefore \begin{cases} x_1 = x_0 + \Delta x = 1 + 0.004256 = 1.004256 \\ y_1 = y_0 + \Delta y = -1.7 - 0.029846 = -1.729846 \end{cases}$$

继续做下去，直到 $\max(|\Delta x|, |\Delta y|) < 10^{-6}$ 时停止。

第五章 特征值求解

§ 1 幂法和反幂法

1.1 幂法

用于求矩阵的按模最大的特征值与相应的特征向量的近似值。

设 A 为 n 阶实矩阵，

λ_i, u_i ($i = 1, 2, \dots, n$)为 A 的特征值和相应的特征向量，

且满足： $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$

u_1, u_2, \dots, u_n ，线性无关。

对任意向量 $x^{(0)}$, 有 $x^{(0)} = \sum_{i=1}^n \alpha_i u_i$, α_i 不全为零 .

$$x^{(k+1)} = Ax^{(k)} = A^{k+1}x^{(0)}$$

$$= \sum_{i=1}^n A^{k+1} \alpha_i u_i = \sum_{i=1}^n \alpha_i \lambda_i^{k+1} u_i$$

$$= \lambda_1^{k+1} \left[\alpha_1 u_1 + \left(\frac{\lambda_2}{\lambda_1} \right)^{k+1} \alpha_2 u_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1} \right)^{k+1} \alpha_n u_n \right]$$

$$\approx \lambda_1^{k+1} \alpha_1 u_1$$

定理：设 $A \in R^{n \times n}$, 特征值 $\lambda_i (i = 1, 2, \dots, n)$ 满足

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|,$$

且与 λ_i 对应的特征向量 u_1, u_2, \dots, u_n 线性无关,

则对任意非零初始向量 $x^{(0)} (\alpha_1 \neq 0)$, 向量序列

$$x^{(k)} = A^k x^{(0)} \rightarrow \lambda_1^k \alpha_1 u_1, \quad \frac{x_i^{(k+1)}}{x_i^{(k)}} \rightarrow \lambda_1 (k \rightarrow \infty).$$

相应的特征向量为 $x^{(k+1)}$.

注： $x^{(k+1)} \approx \lambda_1^{k+1} \alpha_1 u_1$, 实际计算时将 $x^{(k+1)}$ 标准化。

幂法的收敛速度取决于 比值 $\left| \frac{\lambda_2}{\lambda_1} \right|$, 比值越小, 收敛越快.

1.2 幂法的加速

(一) 原点移位法

λ_i 是 A 的特征值, 则 $\lambda_i - \lambda_0$ 是 $A - \lambda_0 I$ 的特征值

$$\begin{aligned} x^{(k+1)} &= (A - \lambda_0 I) x^{(k)} \\ &= (\lambda_1 - \lambda_0)^{k+1} [\alpha_1 u_1 + (\frac{\lambda_2 - \lambda_0}{\lambda_1 - \lambda_0})^{k+1} \alpha_2 u_2 + \cdots (\frac{\lambda_n - \lambda_0}{\lambda_1 - \lambda_0})^{k+1} \alpha_n u_n] \end{aligned}$$

(二)幂法的埃特肯 (Aitken) 加速

若 $\{a_k\}$ 收敛与 a , 且 $\lim_{k \rightarrow \infty} \frac{a_{k+1} - a}{a_k - a} = c \neq 0$

即 $\{a_k\}$ 线性收敛, 当 k 充分大时, 有

$$\frac{a_{k+1} - a}{a_k - a} \approx \frac{a_{k+2} - a}{a_{k+1} - a}$$

$$y_{n+2} = x_{n+2} - \frac{(x_{n+2} - x_{n+1})^2}{x_{n+2} - 2x_{n+1} + x_n}$$

$$\Rightarrow a \approx a_k - \frac{(a_{k+1} - a_k)^2}{a_{k+2} - 2a_{k+1} + a_k} := \hat{a}_k$$

用 \hat{a}_k 逼近 a , 这种方法称为 *Aitken* 加速法.

3. 对称矩阵的Rayleigh商加速法

❖ 定义 设 A 对称, $x \neq 0$, 则称

$$R(x) = \frac{x^T A x}{x^T x}$$

❖ 为 x 关于 A 的Rayleigh商

$$\left\{ \begin{array}{l} y^{(k)} = \frac{x^{(k)}}{\max(x^{(k)})} = \frac{A^k x^{(0)}}{\max(A^k x^{(0)})} \\ x^{(k+1)} = A y^{(k)} = \frac{A^{k+1} x^{(0)}}{\max(A^k x^{(0)})} \\ R(y^{(k)}) = \frac{(y^{(k)})^T A y^{(k)}}{(y^{(k)})^T (y^{(k)})} = \frac{(A^k x^{(0)})^T A^{k+1} x^{(0)}}{(A^k x^{(0)})^T A^k x^{(0)}} \end{array} \right.$$

$$R(y^{(k)}) = \frac{(A^k x^{(0)})^T A^{k+1} x^{(0)}}{(A^k x^{(0)})^T A^k x^{(0)}} = \frac{\sum_{j=1}^n \alpha_j^2 \lambda_j^{2k+1}}{\sum_{j=1}^n \alpha_j^2 \lambda_j^{2k}} \approx \lambda_1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)$$

反幂法

基本思想: $Ax = \lambda x \Rightarrow x = A^{-1}(\lambda x)$, 则 $A^{-1}x = \frac{1}{\lambda}x$

(1) A 与 A^{-1} 的特征值互为倒数, 特征向量不变,

求 A 的按模最小的特征值 λ_n

\Leftrightarrow 求 A^{-1} 的按模最大的特征值 $\frac{1}{\lambda_n}$.

(2) 计算 $x^{(k+1)} = A^{-1}y^{(k)} \Leftrightarrow$ 解方程组 $Ax^{(k+1)} = y^{(k)}$

例：用反幂法求矩阵 $A = \begin{bmatrix} 2 & -1 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$

求按模最小的特征值及 特征向量，取 $x^{(0)} = (0, 0, 1)^T$ ；

解：求 A 按模最小的特征值及其 特征向量

用反幂法

$$x^{(k+1)} = A^{-1}y^{(k)} \Leftrightarrow Ax^{(k+1)} = y^{(k)}$$

将 A 进行 LU 分解，

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 \\ 0 & 2 & -1 \\ 0 & 0 & \frac{3}{2} \end{bmatrix} = LU$$



$$y^{(0)} = x^{(0)} = (0,0,1)^T$$

$$x^{(1)} = A^{-1}y^{(0)} \Leftrightarrow L U x^{(1)} = y^{(0)} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & -1 & 0 \\ 0 & 2 & -1 \\ 0 & 0 & \frac{3}{2} \end{bmatrix}$$

$$Lz = y^{(0)} \quad z = (0,0,1)^T$$

$$Ux^{(1)} = z \quad x^{(1)} = \left(\frac{1}{6}, \frac{1}{3}, \frac{2}{3}\right)^T \quad \mu = \frac{2}{3}$$

$$\lambda^{(1)} = \frac{1}{\mu} = 1.5 \quad y^{(1)} = \left(\frac{1}{4}, \frac{1}{2}, 1\right)^T$$

$$Lz = y^{(1)} \quad z = \left(\frac{1}{4}, \frac{1}{2}, \frac{5}{4}\right)^T$$

$$Ux^{(2)} = z \quad x^{(2)} = \left(\frac{11}{24}, \frac{2}{3}, \frac{5}{6}\right)^T \quad \mu = \frac{5}{6}$$

$$\lambda^{(2)} = \frac{1}{\mu} = 1.2 \quad y^{(2)} = \left(\frac{11}{20}, \frac{4}{5}, 1\right)^T$$

第六章插值

插值区间

已知函数 $y = f(x)$ 在 $[a, b]$ 上有定义，且已经测得在点

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

处的函数值为 $y_0 = f(x_0), \dots, y_n = f(x_n)$

插值节点

如果存在一个简单易算的函数 $P(x)$ ，使得

$$P(x_i) = f(x_i), \quad i = 1, 2, \dots, n$$

则称 $P(x)$ 为 $f(x)$ 的插值函数

插值节点无需递增排列，但必须确保互不相同！

插值条件

求插值函数 $P(x)$ 的方法就称为插值法

基函数法

$n+1$ 维线性空间

记 $Z_n(x) = \{\text{次数不超过 } n \text{ 的多项式的全体}\}$

设 $z_0(x), z_1(x), \dots, z_n(x)$ 构成 $Z_n(x)$ 的一组基，则插值多项式

$$P(x) = a_0 z_0(x) + a_1 z_1(x) + \cdots + a_n z_n(x)$$

通过基函数来构造插值多项式的方法就称为基函数插值法

基函数法基本步骤

- ① 寻找合适的基函数
- ② 确定插值多项式在这组基下的表示系数

Lagrange插值

Lagrange插值基函数

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

$$l_k(x_j) = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}$$

则称 $l_k(x)$ 为节点 x_0, x_1, \dots, x_n 上的拉格朗日插值基函数

线性与抛物线插值

两种特殊情形

$n=1$

$$L_1(x) = y_0 l_0(x) + y_1 l_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}$$

线性插值多项式（一次插值多项式）

$n=2$

$$L_2(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

抛物线插值多项式（二次插值多项式）

例：已知函数 $y = \ln x$ 的函数值如下

x	0.4	0.5	0.6	0.7	0.8
$\ln x$	-0.9163	-0.6931	-0.5108	-0.3567	-0.2231

试分别用线性插值和抛物线插值计算 $\ln 0.54$ 的近似值

解：为了减小截断误差，通常选取插值点 x 邻接的插值节点

线性插值：取 $x_0=0.5, x_1=0.6$ 得

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

将 $x=0.54$ 代入可得： $\ln 0.54 \approx L_1(0.54) = -0.6202$

抛物线插值：取 $x_0=0.4$, $x_1=0.5$, $x_2=0.6$, 可得

$$\ln 0.54 \approx L_2(0.54) = -0.6153$$

 $\ln 0.54$ 的精确值为： $-0.616186\cdots$

可见，抛物线插值的精度比线性插值要高

Lagrange插值多项式简单方便，只要取定节点就可写出基函数，进而得到插值多项式，易于计算机实现。

Lagrange插值

$l_k(x)$ 的表达式

由构造法可得

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

性质

$l_0(x), l_1(x), \dots, l_n(x)$ 构成 $Z_n(x)$ 的一组基

注意

$l_0(x), l_1(x), \dots, l_n(x)$ 与插值节点有关,
但与函数 $f(x)$ 无关

误差估计

如何估计误差

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

插值余项

定理

设 $f(x) \in C^n[a, b]$ (n 阶连续可微), 且 $f^{(n+1)}(x)$ 在 (a, b) 内存在, 则对 $\forall x \in [a, b]$, 有

$$R_n(x) = f(x) - L_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \omega_{n+1}(x)$$

其中 $\xi_x \in (a, b)$ 且与 x 有关, $\omega_{n+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$

插值余项

几点说明

- 余项公式只有当 $f(x)$ 的高阶导数存在时才能使用
- ξ_x 与 x 有关, 通常无法确定, 实际使用中通常是估计其上界

如果 $|f^{(n+1)}(x)| \leq M_{n+1}$, 则 $R_n(x) \leq \frac{M_{n+1}}{(n+1)!} \prod_{i=0}^n |x - x_i|$

- 计算插值点 x 上的近似值时, 应选取与 x 相近插值节点

插值误差举例

例：已知函数 $y = \ln x$ 的函数值如下

x	0.4	0.5	0.6	0.7	0.8
$\ln x$	-0.9163	-0.6931	-0.5108	-0.3567	-0.2231

试估计线性插值和抛物线插值计算 $\ln 0.54$ 的误差

解

线性插值

$$R_1(x) = \frac{f^{(2)}(\xi)}{2}(x - x_0)(x - x_1)$$

$$x_0=0.5, x_1=0.6, \xi \in (0.5, 0.6) \quad \longrightarrow \quad |f^{(2)}(\xi)| \leq |-\xi^{-2}| \leq 4$$

$$\longrightarrow \quad |R_1(0.54)| \leq |2(0.54 - 0.5)(0.54 - 0.6)| = 0.0048$$

Newton 插值

为什么 Newton 插值

Lagrange 插值简单易用，但若增加一个节点时，全部基函数 $l_k(x)$ 都需重新计算，不太方便。

解决办法

设计一个可以逐次生成插值多项式的算法，即 n 次插值多项式可以通过 $n-1$ 次插值多项式生成 —— Newton 插值法

什么是差商

设函数 $f(x)$, 节点 x_0, \dots, x_n

$$f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i} \rightarrow f(x) \text{ 关于点 } x_i, x_j \text{ 的一阶差商}$$

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

$\rightarrow f(x)$ 关于点 x_i, x_j, x_k 的二阶差商

差商的一般定义

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0} \rightarrow k \text{ 阶差商}$$

差商的性质

- k 阶差商与 k 阶导数之间的关系：若 $f(x)$ 在 $[a,b]$ 上具有 k 阶导数，则至少存在一点 $\xi \in (a, b)$ ，使得

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

差商举例

例：已知 $y = f(x)$ 的函数值表，试计算其各阶差商

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

解：差商表如下

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

Newton / Lagrange

Newton 插值多项式与 **Lagrange** 插值多项式

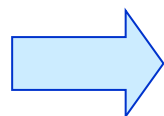
$f(x)$ 在 x_0, x_1, \dots, x_n 上的 n 次插值多项式是唯一的！


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

余项也相同



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



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

将 x 看作节点

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

插值举例

插值节点无需递增排列，但必须确保互不相同！

例：已知函数 $y = \ln x$ 的函数值如下

x	0.4	0.5	0.6	0.7	0.8
$\ln x$	-0.9163	-0.6931	-0.5108	-0.3567	-0.2231

试分别用牛顿线性插值和抛物线插值计算 $\ln 0.54$ 的近似值

解：取节点 **0.5, 0.6, 0.4** 作差商表

$$N_1(x) = -0.6931 + 1.8230(x-0.5)$$

$$N_1(0.54) = -0.6202$$

$$N_2(x) = -0.6931 + 1.8230(x-0.5) - 2.0450(x-0.5)(x-0.6)$$

$$N_2(0.54) = -0.6153$$

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

分段线性插值

插值节点满足: $x_0 < x_1 < \cdots < x_n$ 已知

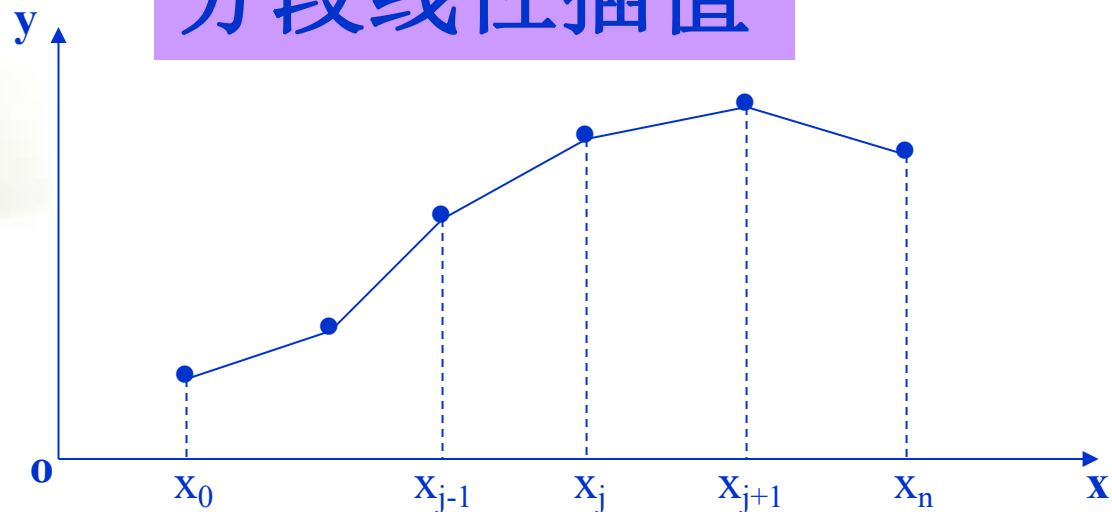
$$y_j = f(x_j) \quad (j = 0, 1, 2, \cdots, n)$$

$x \in [x_j, x_{j+1}]$ 时, 线性插值函数

$$L_h(x) = \frac{x_{j+1} - x}{x_{j+1} - x_j} y_j + \frac{x - x_j}{x_{j+1} - x_j} y_{j+1}$$

$$(j = 0, 1, \cdots, n-1)$$

分段线性插值



$$L_n(x) = \sum_{j=0}^n y_j l_j(x)$$

$$l_j(x) = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}}, & x_{j-1} \leq x \leq x_j \\ \frac{x - x_{j+1}}{x_j - x_{j+1}}, & x_j \leq x \leq x_{j+1} \\ 0, & \text{其它} \end{cases}$$

计算量与 n 无关;

n 越大, 误差越小.



$$\lim_{n \rightarrow \infty} L_n(x) = g(x), x_0 \leq x \leq x_n$$



定义 5.4 给定区间 $[a, b]$ 上的一个分划:

$$a = x_0 < x_1 < \dots < x_n = b$$

已知 $f(x_j) = y_j$ ($j = 0, 1, \dots, n$), 如果

$$S(x) = \begin{cases} S_1(x), x \in [x_0, x_1] \\ S_2(x), x \in [x_1, x_2] \\ \dots\dots\dots \\ S_n(x), x \in [x_{n-1}, x_n] \end{cases}$$

满足: (1) $S(x)$ 在 $[x_j, x_{j+1}]$ 上为三次多项式;

(2) $S''(x)$ 在区间 $[a, b]$ 上连续;

(3) $S(x_j) = y_j$ ($j = 0, 1, \dots, n$).

则称 $S(x)$ 为三次样条插值函数.

(1)自然边界条件: $S''(x_0)=0, S''(x_n)=0$

(2)周期边界条件: $S'(x_0)=S'(x_n), S''(x_0)=S''(x_n)$

(3)固定边界条件: $S'(x_0)=f'(x_0), S'(x_n)=f'(x_n)$

例2 5.7 已知 $f(-1) = 1, f(0) = 0, f(1) = 1$.求 $[-1, 1]$ 上的三次自然样条(满足自然边界条件).

解 设

$$S(x) = \begin{cases} a_1x^3 + b_1x^2 + c_1x + d_1, & x \in [-1, 0] \\ a_2x^3 + b_2x^2 + c_2x + d_2, & x \in [0, 1] \end{cases}$$

则有:

$$-a_1 + b_1 - c_1 + d_1 = 1,$$

$$d_1 = 0,$$

$$a_2 + b_2 + c_2 + d_2 = 1$$

$$d_1 = d_2,$$

$$c_1 = c_2,$$

$$b_1 = b_2$$



由自然边界条件:

$$-6a_1+2b_1=0, \quad 6a_2+2b_2=0$$

解方程组,得

$$a_1=-a_2=1/2, \quad b_1=b_2=3/2,$$

$$c_1=c_2=d_1=d_2=0.$$

问题的解

$$S(x) = \begin{cases} \frac{1}{2}x^3 + \frac{3}{2}x^2, & x \in [-1, 0] \\ -\frac{1}{2}x^3 + \frac{3}{2}x^2, & x \in [0, 1] \end{cases}$$

三次Hermite插值

$$H(x) = y_0\alpha_0(x) + y_1\alpha_1(x) + m_0\beta_0(x) + m_1\beta_1(x)$$

$$\alpha_0(x) = \left(1 + 2\frac{x - x_0}{x_1 - x_0}\right)\left(\frac{x_1 - x}{x_1 - x_0}\right)^2 \quad \beta_0(x) = (x - x_0)\left(\frac{x_1 - x}{x_1 - x_0}\right)^2$$

$$\alpha_1(x) = \left(1 + 2\frac{x_1 - x}{x_1 - x_0}\right)\left(\frac{x - x_0}{x_1 - x_0}\right)^2 \quad \beta_1(x) = (x - x_1)\left(\frac{x - x_0}{x_1 - x_0}\right)^2$$

x	x_0	x_1
$\alpha_0(x)$	1	0
$\alpha'_0(x)$	0	0
$\alpha_1(x)$	0	1
$\alpha'_1(x)$	0	0

x	x_0	x_1
$\beta_0(x)$	0	0
$\beta'_0(x)$	1	0
$\beta_1(x)$	0	0
$\beta'_1(x)$	0	1

第七章 拟合、函数逼近

三个问题

问题一

已知一个函数的数值表

x	x_1	x_2	x_n
y	y_1	y_2	y_n

能否找到一个简单易算的 $p(x)$ ，使得 $p(x_i) = y_i$ 。

问题二

函数 $f(x)$ 的表达式非常复杂，能否找到一个简单易算的 $p(x)$ ，使得 $p(x)$ 是 $f(x)$ 的一个合理的逼近。

问题三

问题一的表中的数值带有误差，能否找到一个简单易算的 $p(x)$ ，可以近似地表示这些数据。

插值

拟合、逼近

曲线拟合

已知 $f(x)$ 在某些点的函数值:

x	x_0	x_1	\dots	x_m
$f(x)$	y_0	y_1	\dots	y_m

能否找到一个简单易算的 $p(x)$, 使得 $f(x) \approx p(x)$

但是 (1) m 通常很大

(2) y_i 本身是测量值, 不准确, 即 $y_i \neq f(x_i)$

这时不要求 $p(x_i) = y_i$, 而只要

$p(x_i) - y_i$ 总体上尽可能小

最小二乘

曲线拟合的最小二乘问题

已知函数值表 (x_i, y_i) ，在函数空间 Φ 中求 $S^*(x)$ ，使得

$$\sum_{i=0}^m \omega_i [S^*(x_i) - y_i]^2 = \min_{s(x) \in \Phi} \sum_{i=0}^m \omega_i [S(x_i) - y_i]^2$$

其中 ω_i 是点 x_i 处的权。

- 这个问题实质上是最佳平方逼近问题的离散形式。可以将求连续函数的最佳平方逼近函数的方法直接用于求解该问题。

最小二乘求解

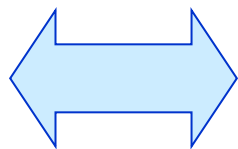
对任意 $S(x) \in \Phi = \text{span}\{\varphi_0, \varphi_1, \dots, \varphi_n\}$, 可设

$$S(x) = a_0\varphi_0 + a_1\varphi_1 + \dots + a_n\varphi_n(x)$$

则求 $S^*(x)$ 等价于求下面的多元函数的最小值点

$$\begin{aligned} I(a_0, a_1, \dots, a_n) &= \sum_{i=0}^m \omega_i [S(x_i) - y_i]^2 \\ &= \sum_{i=0}^m \omega_i \left[\sum_{j=0}^n a_j \varphi_j(x_i) - y_i \right]^2 \end{aligned}$$

最小值点



$$\frac{\partial I}{\partial a_k} = 0$$

$$k = 0, 1, \dots, n$$

最小二乘求解

$$\longrightarrow \sum_{j=0}^n (\varphi_j, \varphi_k) a_k = (f, \varphi_k) \quad (k = 0, 1, \dots, n)$$

这里的内积是离散带权内积，即

$$(\varphi_j, \varphi_k) = \sum_{i=0}^m \omega_i \varphi_j(x_i) \varphi_k(x_i), \quad (f, \varphi_k) = \sum_{i=0}^m \omega_i f(x_i) \varphi_k(x_i)$$

\longrightarrow 法方程

$$\begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \cdots & (\varphi_n, \varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix}$$

法方程

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

最小二乘求解

设法方程的解为: $a_0^*, a_1^*, \dots, a_n^*$, 则

$$S^*(x) = a_0^* \varphi_0 + a_1^* \varphi_1 + \cdots + a_n^* \varphi_n(x)$$

结论

$S^*(x)$ 是 $f(x)$ 在 Φ 中的 最小二乘解

举例

例： 求下面数据表的二次最小二乘拟合多项式

x_i	0	0.25	0.50	0.75	1.00
$f(x_i)$	1.0000	1.2840	1.6487	2.1170	2.7183

解： 设二次拟合多项式为 $p_2(x) = a_0 + a_1x + a_2x^2$

得法方程

$$\begin{bmatrix} 5 & 2.5 & 1.875 \\ 2.5 & 1.875 & 1.5625 \\ 1.875 & 1.5625 & 1.3828 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 8.7680 \\ 5.4514 \\ 4.4015 \end{bmatrix}$$

解得 $a_0 = 1.0052, a_1 = 0.8641, a_2 = 0.8437$

所以此组数据的二次最小二乘拟合多项式为

$$p_2(x) = 1.0052 + 0.8641x + 0.8437x^2$$

函数逼近

最佳逼近

记 H_n 为所有次数不超过 n 的多项式组成的集合，给定函数 $f(x) \in C[a, b]$ ，若 $P^*(x) \in H_n$ 使得

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

则称 $P^*(x)$ 为 $f(x)$ 在 $C[a, b]$ 上的最佳逼近多项式

函数逼近

最佳平方逼近

$$\|f(x) - P^*(x)\|_2 = \min_{P \in H_n} \|f(x) - P(x)\|_2$$

最小二乘拟合

给定 $f(x) \in C[a, b]$ 的数据表

x	x_0	x_1	\dots	x_n
y	y_0	y_1	\dots	y_n

寻找 $P^*(x)$ ，使得下面的离散 2-范数最小

$$\sum_{i=1}^n |f(x_i) - P(x_i)|^2$$

正交多项式

定义

设 $\varphi_n(x)$ 是首项系数不为 0 的 n 次多项式, 若

$$(\varphi_j, \varphi_k) = \int_a^b \rho(x) \varphi_j(x) \varphi_k(x) dx = \begin{cases} 0, & j \neq k \\ A_k \neq 0, & j = k \end{cases}$$

则称 $\{\varphi_n\}_{n=0}^{\infty}$ 为 $[a, b]$ 上带权 $\rho(x)$ 正交

称 $\varphi_n(x)$ 为 n 次正交多项式

Chebyshev 多项式

Chebyshev 多项式

$$\rho(x) = \frac{1}{\sqrt{1-x^2}}$$

在 $[-1, 1]$ 上带权 $\rho(x)$ 的正交多项式称为切比雪夫多项式

- 切比雪夫多项式的表达式

$$T_n(x) = \cos(n \arccos x)$$

$$x \in [-1, 1], \quad n = 0, 1, 2, \dots$$

- 令 $x = \cos \theta$, 则 $T_n(x) = \cos(n\theta)$, 展开后即得

$$\begin{aligned} T_n(x) &= \cos^n \theta - C_n^2 \cos^{n-2} \theta \sin^2 \theta + C_n^4 \cos^{n-4} \theta \sin^4 \theta + \dots \\ &= x^n - C_n^2 x^{n-2} (1-x^2) + C_n^4 x^{n-4} (1-x^2)^2 + \dots \end{aligned}$$



Chebyshev 零点插值多项式

Chebyshev 插值

以 Chebyshev 多项式的零点作为插值节点进行插值

好处：误差最小

$$x_k = \cos \left(\frac{2k+1}{2(n+1)} \pi \right)$$

定理

设 $f(x) \in C^{n+1}[-1, 1]$, 插值节点 x_0, x_1, \dots, x_n 为 $T_{n+1}(x)$ 的 $n+1$ 个零点, 则

$$\|f(x) - L_n(x)\|_{\infty} = \min_{p \in H_n(x)} \|f(x) - p(x)\|_{\infty}$$

Legendre 多项式

Legendre 多项式

在 $[-1, 1]$ 上带权 $\rho(x)=1$ 的正交多项式称为 勒让德多项式

记号: P_0, P_1, P_2, \dots

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

Legendre多项式

$$P_0(x) = 1 \quad (n+1)P_{n+1}(x) = (2n+1)x P_n(x) - nP_{n-1}(x)$$

$$P_1(x) = x \quad \text{其中 } P_0(x) = 1, P_1(x) = x,$$

$$P_2(x) = (3x^2 - 1)/2$$

$$P_3(x) = (5x^3 - 3x)/2$$

$$P_4(x) = (35x^4 - 30x^2 + 3)/8$$

$$P_5(x) = (63x^5 - 70x^3 + 15x)/8$$

⋮

最佳平方逼近

设 $f(x) \in C[a, b]$, $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x) \in C[a, b]$ 线性无关, 令

$$\Phi = \text{span} \{ \varphi_0, \varphi_1, \dots, \varphi_n \}$$

求 $S^*(x) \in \Phi$, 使得

$$\|f(x) - S^*(x)\|_2^2 = \min_{S \in \Phi} \|f(x) - S(x)\|_2^2$$

$S^*(x)$ 称为 $f(x)$ 在 Φ 中的最佳平方逼近函数

其中

$$\begin{aligned} \|f(x) - S(x)\|_2^2 &= (f - S, f - S) \\ &= \int_a^b \rho(x) (f(x) - S(x))^2 dx \end{aligned}$$

最佳平方逼近

即
$$\frac{\partial I}{\partial a_k} = 2 \int_a^b \rho(x) \left(f(x) - \sum_{j=0}^n a_j \varphi_j(x) \right) \varphi_k(x) dx = 0$$

$$\iff \sum_{j=0}^n a_j \int_a^b \rho(x) \varphi_j(x) \varphi_k(x) dx = \int_a^b \rho(x) f(x) \varphi_k(x) dx$$

$$\iff \sum_{j=0}^n a_j (\varphi_k, \varphi_j) = (\varphi_k, f) \quad k = 0, 1, \dots, n$$

$$\begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \cdots & (\varphi_n, \varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix}$$

法方程

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

举例

例：求 $f(x)=\sqrt{1+x^2}$ 在 $[0, 1]$ 上的一次最佳平方逼近多项式

解： $d_0 = (\varphi_0, f) = (1, f) = \int_0^1 \sqrt{1+x^2} \, dx \approx 1.147$

$$d_1 = (\varphi_1, f) = (x, f) = \int_0^1 x\sqrt{1+x^2} \, dx \approx 0.609$$

$$\begin{bmatrix} 1 & 1/2 \\ 1/2 & 1/3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \end{bmatrix} \Rightarrow a_0 = 0.934, \quad a_1 = 0.426$$

$$\Rightarrow S^*(x) = 0.934 + 0.426 x$$

$$\|\delta(x)\|_2^2 = \|f(x)\|_2^2 - \sum_{j=0}^n a_j^* (\varphi_j, f) = 0.0026$$

$$\|\delta(x)\|_\infty \approx 0.066$$

第8章 数值积分



$$I(f) = \int_a^b f(x) \, dx$$

● 微积分基本公式： $\int_a^b f(x) \, dx = F(b) - F(a)$

● 但是在许多实际计算问题中

(1) $F(x)$ 表达式较复杂时，计算较困难。如 $f(x) = \frac{1}{1+x^6}$

(2) $F(x)$ 难求！甚至有时不能用初等函数表示。

如 $f(x) = \frac{1}{x} \sin x, \quad e^{-x^2}$

(3) $f(x)$ 表达式未知，只有通过测量或实验得来的数据表

数值积分公式的一般形式

一般地，用 $f(x)$ 在 $[a, b]$ 上的一些离散点

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

上的函数值的加权平均作为 $f(\xi)$ 的近似值，可得

$$\int_a^b f(x) dx \approx \sum_{i=0}^n A_i f(x_i)$$

Diagram illustrating the components of the numerical integration formula:

- $\int_a^b f(x) dx$ is the integral to be approximated.
- $\sum_{i=0}^n$ is the summation over $n+1$ points.
- A_i is the weight coefficient, labeled "求积系数" (Integration Coefficient).
- $f(x_i)$ is the function value at the node x_i , labeled "求积节点" (Integration Node).
- The entire formula is labeled "机械求积方法" (Mechanical Quadrature Method).

- 将定积分计算转化成被积函数的函数值的计算
- 无需求原函数
- 易于计算机实现

定义：如果对于所有次数不超过 m 的多项式 $f(x)$ ，公式

$$\int_a^b f(x) dx \approx \sum_{i=0}^n A_i f(x_i)$$

精确成立，但对某个次数为 $m+1$ 的多项式不精确成立，则称该求积公式具有 m 次代数精度

代数精度的验证方法

- 将 $f(x) = 1, x, x^2, \dots, x^m$ 依次代入，公式精确成立；
- 但对 $f(x) = x^{m+1}$ 不精确成立。即：

$$\sum_{i=0}^n A_i x_i^k = \int_a^b x^k dx = \frac{b^{k+1} - a^{k+1}}{k+1}$$

$$\sum_{i=0}^n A_i x_i^{m+1} \neq \int_a^b x^{m+1} dx = \frac{b^{m+2} - a^{m+2}}{m+2}$$

($k = 0, 1, \dots, m$)

例：试确定系数 A_i ，使得下面的求积公式具有尽可能高的代数精度，并求出此求积公式的代数精度。

$$\int_{-1}^1 f(x)dx \approx A_0 f(-1) + A_1 f(0) + A_2 f(1)$$

解：将 $f(x)=1, x, x^2$ 代入求积公式，使其精确成立，可得

$$\begin{cases} A_0 + A_1 + A_2 = (b^1 - a^1) / 1 = 2 \\ -A_0 + A_2 = (b^2 - a^2) / 2 = 0 \\ A_0 + A_2 = (b^3 - a^3) / 3 = 2 / 3 \end{cases}$$

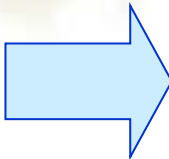
解得 $A_0=1/3, A_1=4/3, A_2=1/3$ 。所以求积公式为

$$\int_{-1}^1 f(x)dx \approx [f(-1) + 4f(0) + f(1)]/3$$

易验证该公式对 $f(x)=x^3$ 也精确成立，但对 $f(x)=x^4$ 不精确成立，所以此求积公式具有 3 次代数精度。

设求积节点为: $a \leq x_0 < x_1 < \cdots < x_n \leq b$

若 $f(x_i)$ 已知, 则可做 n 次多项式插值: $L_n(x) = \sum_{i=0}^n l_i(x) f(x_i)$


$$\int_a^b f(x) dx \approx \int_a^b L_n(x) dx = \sum_{i=0}^n f(x_i) \int_a^b l_i(x) dx \equiv \sum_{i=0}^n A_i f(x_i)$$

其中 $A_i = \int_a^b l_i(x) dx$ $\sum_{k=0}^n A_k = b - a$

插值型求积公式

误差: $R[f] = \int_a^b (f(x) - L_n(x)) dx = \int_a^b R_n(x) dx$

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

Newton-Cotes 求积公式

基于等分点的插值型求积公式

- 积分区间: $[a, b]$
- 求积节点: $x_i = a + i \times h$
- 求积公式:

$$h = \frac{b - a}{n}$$

$$\int_a^b f(x) \, dx \approx (b - a) \sum_{i=0}^n C_i^{(n)} f(x_i)$$

$n = 1:$ $C_0^{(1)} = \frac{1}{2}, \quad C_1^{(1)} = \frac{1}{2}$

$$\int_a^b f(x)dx \approx \frac{b-a}{2}[f(a) + f(b)] \triangleq T$$

梯形公式

代数精度 = 1

$n = 2:$ $C_0^{(2)} = \frac{1}{6}, \quad C_1^{(2)} = \frac{2}{3}, \quad C_2^{(2)} = \frac{1}{6}$

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

抛物线公式
Simpson公式

代数精度 = 3

- 梯形公式 (n=1) 的余项

$$R[f] = -\frac{1}{12}(b-a)^3 f''(\eta) \quad \boxed{\eta \in (a, b)}$$

- Simpson公式 (n=2) 的余项

$$R[f] = -\frac{1}{180} \left(\frac{b-a}{2} \right)^4 f^{(4)}(\eta) \quad \boxed{\eta \in (a, b)}$$

□ 提高积分计算精度的常用两种方法

- 用 复合公式
- 用 非等距节点

复合求积公式

- 将积分区间分割成多个小区间
- 在每个小区间上使用低次牛顿—科特斯求积公式

- 将 $[a, b]$ 分成 n 等分 $[x_i, x_{i+1}]$, 其中

$$x_i = a + i \cdot h, \quad h = \frac{b-a}{n} \quad (i = 0, 1, \dots, n)$$

复合梯形公式

$$\begin{aligned} \int_a^b f(x) \, dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) \, dx \approx \sum_{i=0}^{n-1} \frac{h}{2} [f(x_i) + f(x_{i+1})] \\ &= \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b) \right] \triangleq T_n \end{aligned}$$

- 余项

$$R[f] = -\frac{h^3}{12} \sum_{i=0}^{n-1} f''(\eta_i) = -\frac{b-a}{12} h^2 f''(\eta) \quad \eta \in (a, b)$$

复合 Simpson 公式

$$\begin{aligned}\int_a^b f(x) \, dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) \, dx \approx \int_a^b \sum_{i=0}^{n-1} \frac{h}{6} [f(x_i) + 4f(x_{i+\frac{1}{2}}) + f(x_{i+1})] \\ &= \frac{h}{6} \left[f(a) + 4 \sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b) \right] \triangleq S_n\end{aligned}$$

● 余项

$$R[f] = -\frac{h^5}{2880} \sum_{i=0}^{n-1} f^{(4)}(\eta_i) = -\frac{b-a}{2880} h^4 f^{(4)}(\eta) \quad \boxed{\eta \in (a, b)}$$

性质：复合梯形公式和复合Simpson 公式都是收敛的，也都是稳定的。

例2 分别用复化梯形公式与复化Simpson公式
计算积分

$$I = \int_0^1 e^x dx$$

的近似值，为使截断误差的绝对值不超过 $\frac{1}{2} \times 10^{-4}$
至少应将[0,1]的多少等份.

[解] (1)用复化梯形公式, 截断误差为

$$R_T(f) = -\frac{b-a}{12} h^2 f''(\xi), \quad \xi \in (a, b) \quad \left| f^{(k)}(x) \right| = e^x \leq e, \quad x \in [0, 1]$$

$$|R_T| = \frac{1}{12} h^2 |f''(\xi)| \leq \frac{h^2}{12} e = \frac{1}{12} \frac{1}{n^2} e \leq \frac{1}{2} \times 10^{-4},$$

$$n^2 \geq \frac{2}{12} \times 10^4 e, \quad n \geq 67.3$$

故取 $n=68$

(2)若用复化Simpson公式，截断误差为

$$R_S(f) = -\frac{b-a}{180}h^4 f^{(4)}(\xi) \quad \xi \in (a,b)$$

$$|R_S(f)| = \left| -\frac{b-a}{180}h^4 f^{(4)}(\xi) \right| = \frac{1-0}{180}h^4 e^\xi \leq \frac{1}{180} \frac{1}{n^4} e$$

$$\leq \frac{1}{2} \times 10^{-4}, \xi \in (0,1)$$

$$n^4 \geq \frac{2e}{180} \times 10^4, \quad n \geq 4.1688$$

故取 $n=6$

定义 如果一组节点 $x_1, x_2, \dots, x_n \in [a, b]$ 能使求积公式

$$\int_a^b \rho(x) f(x) dx \approx \sum_{k=1}^n A_k f(x_k)$$

具有 $2n+1$ 次代数精确度, 则称这组节点 $\{x_k\}$ 为Gauss点, 称为带权函数 $\rho(x)$ 的Gauss型求积公式.

例5 试确定求积公式

$$\int_{-4}^4 f(x) dx \approx \frac{8}{3} [2f(-2) - f(0) + 2f(2)]$$

代数精确度, 它是否是 Gauss型求积公式.

$$\int_{-4}^4 f(x) dx \approx \frac{8}{3} [2f(-2) - f(0) + 2f(2)]$$

解 由于当 $f(x)$ 分别取 $1, x, x^2, x^3$ 有

$$\int_{-4}^4 dx = 8 = \frac{8}{3} [2 - 1 + 2] = 8$$

$$\int_{-4}^4 x dx = 0 = \frac{8}{3} [2 \times (-2) - 0 + 2 \times 2] = 0$$

$$\int_{-4}^4 x^2 dx = \frac{1}{3} x^3 \Big|_{-4}^4 = \frac{128}{3} = \frac{8}{3} [2 \times (-2)^2 - 0 + 2 \times 2^2] = \frac{128}{3}$$

$$\int_{-4}^4 x^3 dx = \frac{1}{4} x^4 \Big|_{-4}^4 = 0 = \frac{8}{3} [2 \times (-2)^3 - 0 + 2 \times (2)^3] = 0$$

即求积公式精确成立.

而当 $f(x) = x^4$

$$\int_{-4}^4 x^4 dx = \frac{1}{5} x^5 \Big|_{-4}^4 = \frac{2048}{5}, \quad \frac{8}{3} [2 \times (-2)^4 - 0 + 2 \times (2)^4] = \frac{512}{3} \neq \frac{2048}{5}$$

因此,求积公式的代数精度为3.

它不是Gauss型求积公式.

第九章 常微分方程数值解法

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n) \\ y_0 = y(a) \end{cases} \quad (n=0, 1, \dots)$$

上式称为显式欧拉公式.

Euler法的局部截断误差也可表示为

$$R_{n+1} = \frac{1}{2} h^2 y''(x_n) + O(h^3) = O(h^2)$$

Euler方法是一阶方法.

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

这就是梯形公式.

梯形公式的局部截断误差为

$$\begin{aligned} R_{n+1} &= y(x_{n+1}) - y(x_n) - \frac{h}{2} [f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))] \\ &= -\frac{h^3}{12} y'''(\xi) \end{aligned} \quad x_n < \xi < x_{n+1}$$

故梯形公式为二阶方法.

一般地,RK方法设近似公式为

$$\begin{cases} y_{n+1} = y_n + h \sum_{i=1}^p c_i K_i \\ K_1 = f(x_n, y_n) \\ K_i = f(x_n + a_i h, y_n + h \sum_{j=1}^{i-1} b_{ij} K_j) \quad (i=2, 3, \dots, p) \end{cases}$$

其中 a_i, b_{ij}, c_i , 都是参数, 确定它们的原则是使近似公式在 (x_n, y_n) 处的Taylor展开式与 $y(x)$ 在 x_n 处的Taylor展开式的前面的项尽可能多地重合, 这样就使近似公式有尽可能高的精度。

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + h, y_n + hK_1) \end{cases}$$

这就是改进Euler公式,它是一种二阶龙格-库塔公式。

$$\begin{cases} \bar{y}_{n+1} = y_n + hf(x_n, y_n) & \text{预测} \\ y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})] & \text{校正} \end{cases}$$

上式称为由Euler公式和梯形公式得到的预测—校正公式，也叫改进Euler法，它是显式单步法。

改进Euler法的局部截断误差为 $O(h^3)$,

故它是二阶方法。

例1用欧拉法求初值问题

$$\begin{cases} y' = -\frac{0.9}{1+2x}y \\ y(x_0) = 1 \quad x_0 = 0 \end{cases}$$

当 $h = 0.02$ 时在区间 $[0, 0.10]$ 上的数值解。

解 把 $f(x, y) = -\frac{0.9}{1+2x}y$ 代入欧拉法计算公式。就得

$$y_{n+1} = y_n - h \frac{0.9}{1+2x_n} y_n = \left(1 - \frac{0.018}{1+2x_n}\right) y_n \quad n = 0, 1, \dots, 5$$

具体计算结果如下表：

n	x_n	y_n	$y(x_n)$	$\varepsilon_n = y(x_n) - y_n$
0	0	1.0000	1.0000	0
1	0.02	0.9820	0.9825	0.0005
2	0.04	0.9650	0.9660	0.0005
3	0.06	0.9489	0.9503	0.0014
4	0.08	0.9336	0.9354	0.0018
5	0.10	0.9192	0.923	0.0021



例 用改进的Euler法求初值问题的数值解, 取 $h = 0.1$.

$$\begin{cases} y' = x^2 + x - y & 0 \leq x \leq 0.3 \\ y(0) = 0 \end{cases}$$

解 改进的Euler公式为

$$\begin{cases} \bar{y}_{n+1} = y_n + hf(x_n, y_n) \\ y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})] \end{cases}$$

$$f(x, y) = x^2 + x - y \quad \text{所以}$$

$$\bar{y}_{n+1} = y_n + 0.1(x_n^2 + x_n - y_n) = 0.9y_n + 0.1(x_n^2 + x_n)$$

$$\begin{aligned} y_{n+1} &= y_n + \frac{0.1}{2}[x_n^2 + x_n - y_n + x_{n+1}^2 + x_{n+1} - \bar{y}_{n+1}] \\ &= y_n + \frac{0.1}{2}[x_n^2 + x_n - y_n + (x_n + 0.1)^2 + (x_n + 0.1) - 0.9y_n - 0.1(x_n^2 + x_n)] \\ &= y_n + 0.05[1.9x_n^2 + 2.1x_n - 1.9y_n + 0.11] \end{aligned}$$

$$\begin{aligned}y_{n+1} &= y_n + \frac{0.1}{2}[x_n^2 + x_n - y_n + x_{n+1}^2 + x_{n+1} - \bar{y}_{n+1}] \\&= y_n + 0.05[1.9x_n^2 + 2.1x_n - 1.9y_n + 0.11]\end{aligned}$$

将 $x_0 = 0, y_0 = 0$ 代入上式得

$$\begin{aligned}y_1 &= y_0 + 0.05[1.9x_0^2 + 2.1x_0 - 1.9y_0 + 0.11] \\&= 0.05 \times 0.11 = 0.00550\end{aligned}$$

将 $x_1 = 0.1, y_1 = 0.00550$ 代入上式得

$$\begin{aligned}y_2 &= y_1 + 0.05[1.9x_1^2 + 2.1x_1 - 1.9y_1 + 0.11] \\y_2 &= 0.00550 + 0.05[1.9 \times 0.1^2 + 2.1 \times 0.1 - 1.9 \times 0.0055 + 0.11] \\&= 0.021927\end{aligned}$$

将 $x_2 = 0.2, y_2 = 0.021927$ 代入上式得

$$y_3 = y_2 + 0.05[1.9x_2^2 + 2.1x_2 - 1.9y_2 + 0.11] = 0.050144$$

例12 用二阶方程初值问题 $y'' = 3y' - 2y, y(0) = y'(0) = 1$
化为一阶方程组初值问题,并写出欧拉方法求解的
计算公式.

解 令 $z = y'$ **可得**

$$\begin{aligned} y' &= z, & y(0) &= 1 \\ z' &= 3z - 2y, & z(0) &= 1 \end{aligned}$$

计算公式为

$$\begin{cases} y_{n+1} = y_n + h z_n \\ z_{n+1} = z_n + h(3z_n - 2y_n) \end{cases}$$

$$y_0 = 1, z_0 = 1$$

例4.3 对初值问题
$$\begin{cases} y' + y = 0 \\ y(0) = 1 \end{cases}$$

证明用梯形公式求得的近似解为 $y_n = \left(\frac{2-h}{2+h} \right)^n, x = nh$

并证明当步长 $h \rightarrow 0$ 时, y_n 收敛于精确解 e^{-x}

证明：解初值问题的梯形公式为：

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

$$\because f(x, y) = -y \quad \therefore y_{n+1} = y_n + \frac{h}{2} [-y_n - y_{n+1}]$$

整理成显式 $y_{n+1} = \left(\frac{2-h}{2+h} \right) y_n$ 反复迭代, 得到:

$$y_{n+1} = \left(\frac{2-h}{2+h} \right) y_n$$

$$\begin{aligned} y_{n+1} &= \left(\frac{2-h}{2+h} \right) y_n = \left(\frac{2-h}{2+h} \right)^2 y_{n-1} \\ &= \left(\frac{2-h}{2+h} \right)^3 y_{n-2} = \dots = \left(\frac{2-h}{2+h} \right)^{n+1} y_0 \end{aligned}$$

$$\because y_0 = 1 \qquad y_n = \left(\frac{2-h}{2+h} \right)^n$$

⋮



$$y_n = \left(\frac{2-h}{2+h} \right)^n$$

由于 $x = nh$, 有:

$$\lim_{h \rightarrow 0} y_n = \lim_{h \rightarrow 0} \left(\frac{2-h}{2+h} \right)^{\frac{x}{h}}$$

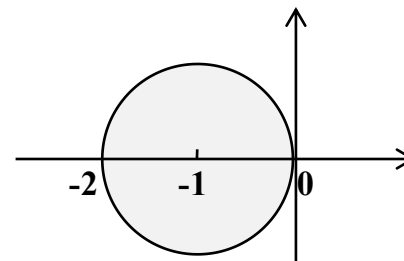
$$= \lim_{h \rightarrow 0} \frac{\left(1 - \frac{h}{2} \right)^{\left(-\frac{2}{h} \right) \left(-\frac{x}{2} \right)}}{\left(1 + \frac{h}{2} \right)^{\left(\frac{2}{h} \right) \left(\frac{x}{2} \right)}} = \frac{e^{-\frac{x}{2}}}{e^{\frac{x}{2}}} = e^{-x}$$

$$\therefore \lim_{h \rightarrow 0} y_n = e^{-x}$$

例：考察显式欧拉法 $y_{i+1} = y_i + h\lambda y_i = (1 + \bar{h})^{i+1} y_0$

$$\varepsilon_0 = y_0 - \bar{y}_0 \rightarrow \bar{y}_{i+1} = (1 + \bar{h})^{i+1} \bar{y}_0$$

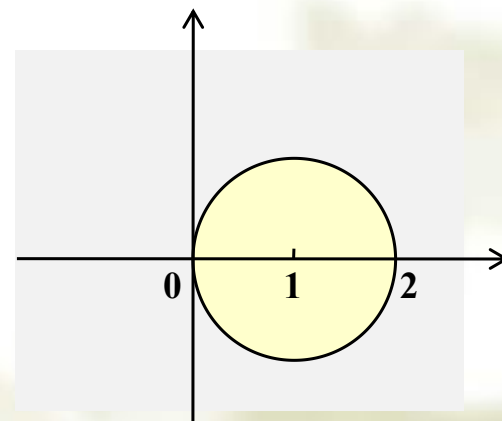
$$\rightarrow \varepsilon_{i+1} = y_{i+1} - \bar{y}_{i+1} = (1 + \bar{h})^{i+1} \varepsilon_0$$



由此可见，要保证初始误差 ε_0 以后逐步衰减， $\bar{h} = \lambda h$ 必须满足： $|1 + \bar{h}| < 1$

例：考察隐式欧拉法 $y_{i+1} = y_i + h\lambda y_{i+1}$

$$y_{i+1} = \left(\frac{1}{1 - \bar{h}} \right) y_i \rightarrow \varepsilon_{i+1} = \left(\frac{1}{1 - \bar{h}} \right)^{i+1} \varepsilon_0$$



可见绝对稳定区域为： $|1 - \bar{h}| > 1$

注：一般来说，隐式欧拉法的绝对稳定性比同阶的显式法的好。