

计算方法总结

龚翔天 PB18000286

2020 秋

目录

第一章 插值	7
1.1 绪论	7
1.2 Lagrange 插值多项式	7
1.3 Newton 插值多项式	8
1.4 Hermite 插值	8
1.5 三次样条函数	9
第二章 最小二乘拟合	11
2.1 绪论	11
2.1.1 范数	11
2.1.2 范数等价性	12
2.2 最小二乘法	12
2.2.1 线性拟合	12
2.2.2 多项式拟合	12
2.3 矛盾方程组	13
2.4 总结	13
第三章 非线性方程组求解	15
3.1 迭代法	15
3.1.1 实根的对分法	15
3.1.2 迭代法	15
3.1.3 绪论	15
3.2 Newton 迭代法	16
3.3 弦截法	16
3.4 非线性方程组	17
第四章 线性方程组直接解法	19
4.1 绪论	19
4.2 直接消元法	19
4.2.1 Gauss 消元法	19
4.2.2 Gauss-Jordan 消元法	19
4.3 矩阵分解	20
4.3.1 Doolittle 分解	20

4.3.2 Crout 分解	20
4.3.3 特殊矩阵的分解方法	20
4.4 向量范数	21
4.4.1 定义	21
4.4.2 一些常用范数	21
4.5 方阵范数	21
4.5.1 一些常用范数	22
4.6 条件数和病态矩阵	23
4.6.1 相对误差随 b 的变化	23
4.6.2 相对误差随 A 的变化	23
第五章 线性方程组迭代解法	25
5.1 绪论	25
5.2 Jacobi 迭代	26
5.3 Gauss-Seidel 迭代	27
5.4 总结	27
5.5 松弛迭代	28
第六章 数值微分与数值积分	29
6.1 数值积分	29
6.1.1 插值型数值积分	29
6.1.2 Newton-Cotes 积分	30
6.1.3 复化积分	31
6.1.4 自动控制误差计算	32
6.1.5 Romberg 积分算法 (外推算法)	33
6.2 重积分 *	34
6.3 Gauss 积分	34
6.3.1 绪论	34
6.3.2 Gauss 积分的性质	34
6.4 数值微分	35
第七章 常微分方程数值解	37
7.1 Euler 公式	37
7.1.1 基于数值微分的 Euler 公式	37
7.1.2 Euler 公式的收敛性	38
7.1.3 基于数值积分的 Euler 公式	38
7.2 线性多步法	39
7.3 Runge-Kutta 方法	39
7.3.1 基本思想	39
7.3.2 二阶 Runge-Kutta 方法	40
7.3.3 高阶 Runge-Kutta 方法	41
7.4 方程组与高阶微分方程	41

7.4.1	方程组	41
7.4.2	高阶微分方程	42
7.5	常微分方程解的稳定性	42
7.5.1	定义	42
7.5.2	差分方程的绝对稳定性	43

第一章 插值

用某种函数近似逼近未知函数.

1.1 绪论

设 $f(x)$ 是定义在区间 $[a, b]$ 上的函数, $a \leq x_0 < x_1 < \cdots < x_n \leq b$ 是插值节点; Φ 为给定的某个函数类, $\{\phi_0(x), \phi_1(x), \cdots, \phi_m(x)\}$ 为它的一组基. 在 Φ 上找一个插值函数 $g(x)$, 满足:

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

设 $g(x) = \sum_{i=1}^m a_i \phi_i(x)$, 那么: $g(x_j) = \sum_{i=0}^m a_i \phi_i(x_j) = f(x_j)$, $j = 0, 1, \cdots, n$, 因此 $\{a_i, i = 0, 1, \cdots, m\}$ 有唯一解当且仅当系数行列式不为 0. Vandermonde 行列式不为 0, 因此可以选为 $\Phi = \mathbb{P}_n(x) = \text{Span}\{1, x, x^2, \cdots\}$

1.2 Lagrange 插值多项式

插值基函数 $l_i(x)$ 满足:

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

求得:

$$l_i(x) = \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} \quad (1.1)$$

$$l_i(x) = \frac{\omega_n(x)}{\omega'_n(x)(x - x_i)}, \quad \omega(x) := \prod_{i=0}^n (x - x_i)$$

节点为 $\{(x_i, f(x_i)), i = 0, 1, \cdots, n\}$ 的插值函数为:

$$L_n(x) = \sum_{i=0}^n f(x_i) l_i(x) \quad (1.2)$$

插值误差:

$$R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^n (x - x_i) \quad (1.3)$$

1.3 Newton 插值多项式

Lagrange 插值多项式的缺点是没有承袭性质.

Newton 插值多项式:

$$a_n = f[x_0, x_1, \dots, x_n]$$

$$N(x) = f(x_0) + \sum_{k=1}^n \left(f[x_0, x_1, \dots, x_n] \prod_{i=0}^{k-1} (x - x_i) \right) \quad (1.4)$$

插值误差:

$$R(x) = f[x, x_0, x_1, \dots, x_n] \prod_{i=0}^n (x - x_i) \quad (1.5)$$

Lagrange 插值多项式和 Newton 插值多项式关系

对于同一组插值节点有:

$$L(x) = N(x) \quad (1.6)$$

二者完全相同, 误差也完全相同. 由此可以得出差商和导数的关系:

$$f[x, x_0, x_1, \dots, x_n] = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}$$

1.4 Hermite 插值

有些插值问题中还会涉及到光滑性条件, 即在某些节点上附加一阶导数的要求:

$$\left\{ (x_i, f(x_i), f'(x_i)) \right\}_{i=0}^n$$

得到 Hermite 插值多项式:

$$H_{2n+1}(x) = \sum_{i=0}^n h_i(x) f(x_i) + \sum_{i=0}^n g_i(x) f'(x_i) \quad (1.7)$$

其中:

$$\begin{cases} h_i(x_j) = \delta_{ij} \\ h'_i(x_j) = 0 \end{cases}, \quad \begin{cases} g_i(x_j) = 0 \\ g'_i(x_j) = \delta_{ij} \end{cases}$$

$$h_i(x) = \left(1 - 2(x - x_i) \sum_{j \neq i} \frac{1}{x_i - x_j} \right) l_i^2(x)$$

$$g_i(x) = (x - x_i) l_i^2(x)$$

误差：

$$R(x) = \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!} \prod_{i=0}^n (x - x_i)^2 \quad (1.8)$$

用 Newton 插值法完成 Hermite 插值：

$$\{z_{2i} = a_{2i+1} = x_i, \quad i = 0, 1, \dots, n\}$$

$$z_0 = z_1 = x_0, \quad z_2 = z_3 = x_1, \quad \dots, \quad z_{2n} = z_{2n+1} = x_n$$

$$f[z_{2i-1}, z_{2i}] = \frac{f(z_{2i}) - f(z_{2i-1})}{z_{2i} - z_{2i-1}}$$

$$f[z_{2i}, z_{2i+1}] = f'(x_i)$$

套用 Newton 插值公式即可：

$$H_{2n+1}(x) = f(z_0) + \sum_{k=1}^{2n+1} \left(f[z_0, z_1, \dots, z_k] \prod_{i=0}^{k-1} (x - z_i) \right) \quad (1.9)$$

套用 Newton 插值公式的误差即可得到 Hermite 插值多项式误差表达式^{1.8}.

1.5 三次样条函数

由于 Runge 现象（插值函数在端点附近震荡很大）在高次多项式插值过程中影响很大，故需要进行分段插值.

分段插值：三次样条插值（插值函数有连续的二阶导数）.

不总结，看书.

第二章 最小二乘拟合

2.1 绪论

2.1.1 范数

定义. 向量范数

映射 $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}^+ \cup 0$ 若满足:

1、非负性:

$$\|X\| \geq 0, \|X\| = 0 \Leftrightarrow X = 0$$

2、齐次性:

$$\forall a \in \mathbb{R}, \|aX\| = |a|\|X\|$$

3、三角不等式:

$$\|X + Y\| \leq \|X\| + \|Y\|$$

则称该映射为向量范数.

定义. p 范数:

对 $X = (x_1, x_2, \dots, x_n)^T$, 定义 x 的 p 范数为:

$$\|X\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

一些常用的范数

1-范数:

$$\|X\|_1 := \sum_{i=1}^n |x_i|$$

2-范数:

$$\|X\|_2 := \sqrt{\sum_{i=1}^n x_i^2}$$

∞ -范数:

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

2.1.2 范数等价性

\mathbb{R}^n 空间中任意两种范数 $\|\cdot\|_p, \|\cdot\|_q$ 称为等价的 ($\|\cdot\|_p \sim \|\cdot\|_q$), 若存在常数 $c, C > 0$ 使得:

$$\forall X \in \mathbb{R}^n, \quad c\|X\|_p \leq \|X\|_q \leq C\|X\|_p$$

例如:

$$\forall X \in \mathbb{R}^n:$$

$$\|X\|_2 \leq \|X\|_1 \leq \sqrt{n}\|X\|_2$$

$$\|X\|_{\infty} \leq \|X\|_1 \leq \sqrt{n}\|X\|_{\infty}$$

$$\|X\|_{\infty} \leq \|X\|_2 \leq \sqrt{n}\|X\|_{\infty}$$

2.2 最小二乘法

使误差的平方和最小的拟合方式称为最小二乘法. 它使误差在 2-范数意义下最小, 即:

对 $[a, b]$ 上的函数 $f(x)$, $\{x_1, x_2, \dots, x_n\}$ 为区间上 n 个互不相同的点, 最小二乘法拟合出的函数 $\phi(x)$ 满足距离最小:

$$\min \left\| \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix} - \begin{pmatrix} \phi(x_1) \\ \phi(x_2) \\ \vdots \\ \phi(x_n) \end{pmatrix} \right\|$$

2.2.1 线性拟合

$$\begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix} \quad (2.1)$$

2.2.2 多项式拟合

n 维多项式空间 \mathbb{R}^n , m 个数据点, 多项式函数 $\phi(x) = \sum_{i=1}^n a_i x^i \in \mathbb{R}^n$, 则其误差为:

$$R = \sum_{i=1}^m (\phi(x_i) - y_i)^2$$

是系数 $\{a_1, a_2, \dots, a_m\}$ 的多元函数. 对 a_i ($\forall i = 1, 2, \dots, m$) 求偏导令为 0, 得到:

$$\frac{\partial R}{\partial a_j} = 2 \sum_{i=1}^m (\phi(x_i) - y_i) x_i^j = 0$$

即:

$$\begin{pmatrix} n & \sum_{i=1}^m x_i \cdots & \sum_{i=1}^m x_i^n \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 \cdots & \sum_{i=1}^m x_i^{n+1} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^m x_i^n & \sum_{i=1}^m x_i^{n+1} \cdots & \sum_{i=1}^m x_i^{2n} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i y_i \\ \vdots \\ \sum_{i=1}^m x_i^n y_i \end{pmatrix} \quad (2.2)$$

2.3 矛盾方程组

线性方程组 $Ax = b$ ($A \in \mathbb{R}^{m \times n}$) 在 $\text{rank}(A, b) \neq \text{rank}(A)$ 时无解, 称为矛盾方程组. 最小二乘法的实质是, 求一个矛盾方程组的解 x , 使 $\|Ax - b\|_2$ 最小.

定理.

给定矩阵 $A \in \mathbb{R}^{m \times n}$, 向量 $b \in \mathbb{R}^m$, 那么:

- 1、线性方程组 $A^T Ax = A^T b$ 恒有唯一解;
- 2、 $\|Ax - b\|_2$ 达到最小, 当且仅当 $A^T Ax = A^T b$;
- 3、使 $\|Ax - b\|_2$ 达到最小的 x 是唯一的, 当且仅当 $\text{rank}(A) = n$.

证明见课本.

重新考察最小二乘法, 它对应矛盾方程:

$$Ax = b$$

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad x = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \in \mathbb{R}^n, \quad b = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m$$

方程 $A^T Ax = A^T b$ 就是最小二乘法方程2.2.

2.4 总结

- 1、若要求拟合的函数空间不是 \mathbb{P}_n , 而是其线性子空间, 以上过程仍然成立. (比如 $\text{Span}\{1, x^2\}$);
- 2、若给定的函数空间不是线性空间, 那么应该先进行坐标变换, 将其转化为多项式拟合问题.

第三章 非线性方程组求解

3.1 迭代法

非线性方程组通常使用迭代法求解.

3.1.1 实根的对分法

利用连续函数的介值定理, 每次将搜索范围减小一半.

3.1.2 迭代法

基本算法:

1、构造出原方程的等价形式 (不动点形式):

$$f(x) = 0 \Leftrightarrow x = \phi(x)$$

2、取合适的初值 x_0 , 构造迭代序列:

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

3、若极限:

$$x^* = \lim_{k \rightarrow +\infty} x_k$$

存在, 则 x^* 就是方程的解. 如果极限不存在, 就要换其他的初值或迭代格式.

3.1.3 绪论

定理. 压缩映射定理

$\phi(x)$ 满足:

- 1、 $a \leq \phi(x) \leq b$, $x \in [a, b]$
- 2、 $\exists 0 < L < 1$, $\forall x \in [a, b]$, $|\phi'(x)| \leq L$.

那么有:

- 1、存在唯一的 x^* 使 $x^* = \phi(x^*)$, 其中 x^* 称为不动点。
- 2、 $\forall x_0 \in [a, b]$, 迭代序列 x_k 收敛, 且有误差估计:

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

$\phi(x)$ 称为压缩映射.

证明见讲义.

3.2 Newton 迭代法

对任意非线性方程 $f(x) = 0$, 其 Newton 迭代格式为:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (3.1)$$

其思想是利用线性函数逼近原来的函数.

$$f(x) = 0 \Leftrightarrow x = \phi(x) = x - \frac{f(x)}{f'(x)} \Leftrightarrow x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

只要初始值与根相距足够近, 就会收敛.

定义. 收敛阶

如果对于迭代产生的数列 $\{x_k\}$, 其极限为 $x_k \rightarrow x^*$ ($k \rightarrow +\infty$), 定义误差 $\varepsilon_k = |x^* - x_k|$, 若存在正实数 $p > 1$ 和 $c > 0$ 使得:

$$\lim_{k \rightarrow +\infty} \frac{\varepsilon_{k+1}}{\varepsilon_k^p} = C$$

那么称迭代格式为 p 阶收敛.

对迭代格式 $\phi(x)$, 记其不动点为 x^* , 那么:

- 1、当 $\phi'(x) \neq 0$ 时, 是一阶收敛;
- 2、当 $\phi'(x) = 0$ 而 $\phi''(x) \neq 0$ 时, 是二阶收敛.

对 Newton 迭代格式:

$$\phi(x) = x - \frac{f(x)}{f'(x)}$$

$$\phi'(x) = \frac{f(x)f''(x)}{(f'(x))^2}$$

- 1、若 x^* 为 $\phi(x)$ 的单根, 则二阶收敛;
- 2、若 x^* 为 $\phi(x)$ 的 p 重根, 则一般是一阶收敛. 如果改用迭代格式: $x_{k+1} = \phi(x_k) = x_k - p \frac{f(x_k)}{f'(x_k)}$ 则仍是二阶收敛的.

3.3 弦截法

将 Newton 迭代格式中的导数用差商 $f[x_{k-1}, x_k] = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$ 替代, 得到迭代格式:

$$x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \quad (3.2)$$

称为弦截法, 它需要两个初始值才可以启动. 单根时, 收敛阶为 1.618, 比 Newton 迭代收敛得慢.

3.4 非线性方程组

二元方程组：

$$\begin{cases} f(x, y) = 0 \\ g(x, y) = 0 \end{cases}$$

将其在初始值 x_0, y_0 附近做展开：

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

当 Jacobi 行列式非零时：

$$J(x_0, y_0) := \begin{pmatrix} \frac{\partial f}{\partial x}(x_0, y_0) & \frac{\partial f}{\partial y}(x_0, y_0) \\ \frac{\partial g}{\partial x}(x_0, y_0) & \frac{\partial g}{\partial y}(x_0, y_0) \end{pmatrix} \neq 0$$

可以依据：

$$\begin{cases} -f(x_1, y_1) = f(x_0, y_0) + (x_1 - x_0) \frac{\partial f}{\partial x}(x_0, y_0) + (y_1 - y_0) \frac{\partial f}{\partial y}(x_0, y_0) \\ -g(x_1, y_1) = g(x_0, y_0) + (x_1 - x_0) \frac{\partial g}{\partial x}(x_0, y_0) + (y_1 - y_0) \frac{\partial g}{\partial y}(x_0, y_0) \end{cases}$$

求出迭代一次后的 x_1, y_1 . 重复此操作即可.

第四章 线性方程组直接解法

4.1 绪论

线性方程组的一般形式：

$$Ax = b$$

其中 A 为 n 阶方阵, $\text{rank}(A) = n$.

直接解法：Cramer 法则.

4.2 直接消元法

4.2.1 Gauss 消元法

对增广矩阵 (A, b) 进行消元, 化为上三角矩阵.

先列主元, 再消元: 第 k 次消元前, 将第 k 至第 n 行中 (从左数) 第一个非零项绝对值最大的一行和第 n 行对换. 这样做可以减小计算误差 (舍入误差).

4.2.2 Gauss-Jordan 消元法

将系数矩阵化为对角阵. 可以认为是在 Gauss 消元法完成后, 由下向上继续进行消元, 直至变成对角阵, 再求解.

总结

从矩阵角度看 Gauss 消元, 第 k 步消元等价于左乘一个单位下三角矩阵 P_k :

$$P_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & I_{k+1,k}^{(k)} & 1 & \\ & & \vdots & & \ddots \\ & & I_{nk}^{(k)} & & & 1 \end{pmatrix}$$

因此 Gauss 消元法等价于左乘一个单位下三角矩阵, 最后使系数矩阵 A 变为上三角矩阵.

于是, 要解方程组 $Ax = b$, 可以先将 A 进行 $A = LU$, 再转化为两个容易求解的方程组:

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

4.3 矩阵分解

4.3.1 Doolittle 分解

将矩阵 A 分解为单位下三角矩阵和上三角矩阵:

$$A = LU$$

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

计算顺序: U 的第一行, L 的第一列, U 的第二行, L 的第二列……

4.3.2 Crout 分解

将矩阵 A 分解为下三角矩阵和单位上三角矩阵:

$$A = LU$$

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

计算顺序: L 的第一列, U 的第一行, L 的第二列, U 的第二行……

4.3.3 特殊矩阵的分解方法

三对角阵的追赶法

对称正定矩阵的 LDL^T 分解

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix} \begin{pmatrix} 1 & l_{12} & \cdots & l_{1n} \\ & 1 & \cdots & l_{2n} \\ & & \ddots & \vdots \\ & & & 1 \end{pmatrix}$$

先计算 A 的 Doolittle 分解, 再取 D 为 U 的对角元组成的对角阵即可.

4.4 向量范数

4.4.1 定义

满足：

1、非负性：

$$\|X\| \geq 0, \|X\| = 0 \Leftrightarrow X = 0$$

2、齐次性：

$$\forall a \in \mathbb{R}, \|aX\| = |a|\|X\|$$

3、三角不等式：

$$\|X + Y\| \leq \|X\| + \|Y\|$$

定义 p 范数：对 $X = (x_1, x_2, \dots, x_n)^T$ ，定义 x 的 p 范数为：

$$\|X\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

4.4.2 一些常用范数

1-范数：

$$\|X\|_1 := \sum_{i=1}^n |x_i|$$

2-范数：

$$\|X\|_2 := \sqrt{\sum_{i=1}^n x_i^2}$$

∞ -范数：

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

4.5 方阵范数

满足：

1、非负性：

$$\|X\| \geq 0, \|X\| = 0 \Leftrightarrow X = 0$$

2、齐次性：

$$\forall a \in \mathbb{R}, \|aX\| = |a|\|X\|$$

3、三角不等式：

$$\|X + Y\| \leq \|X\| + \|Y\|$$

则称为矩阵范数.

若还满足：4、相容性：

$$\|AB\| \leq \|A\|\|B\|$$

则称为相容矩阵范数.

定义. (矩阵) 范数

设 $\|\cdot\|$ 是一种向量范数，那么可以由它定义矩阵范数：

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

这样定义的范数是相容矩阵范数.

4.5.1 一些常用范数

1-范数 (列模和的最大值)：

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

2-范数 (行模和的最大值)：

$$\|A\|_2 = \max_{1 \leq i \leq n} \sum_{j=1}^n |A_{ij}|$$

∞ -范数 (谱范数)：

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

其中谱半径 $\rho(A)$ 定义为：

$$\rho(A) := \max_{1 \leq i \leq n} |\lambda_i|$$

是 A 的模最大的特征值的模.

定理. 谱范数与矩阵范数的关系

λ 为矩阵 A 的特征值，对于任意诱导的矩阵范数 $\|\cdot\|$ ，有：

$$|\lambda| \leq \|A\|$$

$\overline{|\lambda|}$

$$\rho(A) \leq \|A\|$$

$$\rho = \max |\lambda|$$

4.6 条件数和病态矩阵

定义 p -范数下矩阵的条件数

$$\text{Cond}_p(A) := \|A\|_p \times \|A^{-1}\|_p$$

引入条件数，主要是为了讨论解随误差的变化、解的稳定性等.

4.6.1 相对误差随 b 的变化

当 b 有微小变化 δb 时，解的相对变化：

$$\frac{\|\delta x\|}{\|x\|} \leq \text{Cond}(A) \frac{\|\delta b\|}{\|b\|}$$

4.6.2 相对误差随 A 的变化

当 A 有微小变化 δA 时，解的相对变化：

$$\frac{\|\delta A\|}{\|A\|} \leq \frac{\text{Cond}(A) \frac{\|\delta A\|}{\|A\|}}{1 - \text{Cond}(A) \frac{\|\delta A\|}{\|A\|}} \approx \text{Cond}(A) \frac{\|\delta A\|}{\|A\|}$$

假设对角元 $a_{ii} \neq 0$ 都非零, 那么有:

$$\left\{ \begin{array}{l} x_1 = -\frac{1}{a_{11}}(a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n - b_1) \\ x_2 = -\frac{1}{a_{22}}(a_{21}x_1 + a_{23}x_3 + \cdots + a_{2n}x_n - b_2) \\ \quad \quad \quad \dots \\ x_n = -\frac{1}{a_{nn}}(a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n - b_n) \end{array} \right. \quad (5.2)$$

称为 Jacobi 迭代. 其迭代矩阵为:

$$G = I - D^{-1}A = -D^{-1}(L + U)$$

对于 Jacobi 迭代, 还有一些收敛的充分条件:

定理. Jacobi 迭代收敛的充分 (不必要) 条件
若 A 满足严格对角占优, 则 Jacobi 迭代收敛.

检验迭代矩阵 G 的 1 和无穷范数即可完成证明.

5.3 Gauss-Seidel 迭代

分量形式:

$$\begin{cases} x_1^{(k+1)} = -\frac{1}{a_{11}}(a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + a_{14}x_4^{(k)} + \cdots + a_{1n}x_n^{(k)} - b_1) \\ x_2^{(k+1)} = -\frac{1}{a_{22}}(a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)} + a_{24}x_4^{(k)} + \cdots + a_{2n}x_n^{(k)} - b_2) \\ x_3^{(k+1)} = -\frac{1}{a_{33}}(a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{34}x_4^{(k)} + \cdots + a_{3n}x_n^{(k)} - b_3) \\ \vdots \\ x_n^{(k+1)} = -\frac{1}{a_{nn}}(a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + \cdots + a_{nn-1}x_{n-1}^{(k+1)} - b_n) \end{cases} \quad (5.4)$$

$$x_j^{(k+1)} = -\frac{1}{a_{jj}} \left(\sum_{i=1}^{j-1} a_{ji}x_i^{(k+1)} + \sum_{i=j+1}^n a_{ji}x_i^{(k)} - b_j \right)$$

矩阵形式为:

$$\begin{aligned} x^{(k+1)} &= -D^{-1}(Lx^{(k+1)} + Ux^{(k)} - b) \\ (D + L)x^{(k+1)} &= -UX^{(k)} + b \end{aligned}$$

故迭代矩阵为:

$$G = -(D + L)^{-1}U = I - (D + L)^{-1}A$$

定理. Gauss-Seidel 迭代收敛的充分 (不必要) 条件

- 1、 A 为严格对角占优矩阵;
- 2、 A 为实对称正定阵.

证明参考课本/讲义.

5.4 总结

1、Jacobi 和 Gauss-Seidel 方法适用的方程不相互包含: Jacobi 迭代收敛的, Gauss-Seidel 迭代可能不收敛; Gauss-Seidel 迭代收敛的, Jacobi 迭代可能不收敛.

2、大多数情形下 Gauss-Seidel 方法比 Jacobi 方法收敛速度更快, 但也不绝对.

3、收敛快慢由 $\rho(G)$ 唯一确定, G 为迭代矩阵.

5.5 松弛迭代

在 Gauss-Seidel 迭代的基础上, 引入松弛因子 ω 对 $X^{(k)}$ 和由 Gauss-Serial 迭代得到的 $X^{(k+1)}$ 做加权平均, 有:

$$X^{(k+1)} = (1 - \omega)X^{(k)} + \omega X^{(k+1)}$$

$$\begin{cases} x_1^{(k+1)} = x_1^{(k)} - \frac{\omega}{a_{11}}(a_{11}x_1^{(k)} + a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + \cdots + a_{1n}x_n^{(k)} - b_1) \\ x_2^{(k+1)} = x_2^{(k)} - \frac{\omega}{a_{22}}(a_{21}x_1^{(k+1)} + a_{22}x_2^{(k)} + a_{23}x_3^{(k)} + \cdots + a_{2n}x_n^{(k)} - b_2) \\ x_3^{(k+1)} = x_3^{(k)} - \frac{\omega}{a_{33}}(a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{33}x_3^{(k)} + \cdots + a_{3n}x_n^{(k)} - b_3) \\ \vdots \\ x_n^{(k+1)} = x_n^{(k)} - \frac{\omega}{a_{nn}}(a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + \cdots + a_{nn}x_n^{(k)} - b_n) \end{cases} \quad (5.5)$$

$\omega = 1$ 时就是 Gauss-Seidel 迭代.

矩阵形式为:

$$(D + \omega L)x^{(k+1)} = [(1 - \omega)D - \omega U]x^{(k)} + \omega b$$

迭代矩阵:

$$G = I - \left(\frac{1}{\omega} D + L \right)^{-1} A$$

定理. 松弛迭代收敛

- 1、若松弛迭代收敛, 则 $0 < \omega < 2$.
- 2、若 A 为对称正定矩阵, 则 $0 < \omega < 2$ 时松弛迭代收敛.

证明略.

第六章 数值微分与数值积分

6.1 数值积分

积分真值:

$$I(f) = \int_a^b f(x)dx = \lim_{\Delta x_i \rightarrow 0} \left(\sum_{i=0}^n f(x_i) \Delta x_i \right)$$

数值积分是有限离散节点上函数值的某种线性组合:

$$I_n(f) = \sum_{i=0}^n a_i f(x_i)$$

其中 a_i 为积分系数, 与具体被积函数无关, 只与积分区间和积分节点有关.

定义. 代数精度

若数值积分 $I_n(f)$ 满足:

$$I(x^k) - I_n(x^k) = 0, \quad k = 0, 1, \dots, m$$

而

$$I(x^{m+1}) - I_n(x^{m+1}) \neq 0$$

则称 $I_n(f)$ 具有 m 阶代数精度.

6.1.1 插值型数值积分

对于给定的被积函数 $f(x)$, 用它在 $[a, b]$ 上的点列 $(x_i, f(x_i))$, $i = 0, 1, \dots, n$ 作 Lagrange 插值多项式 $L_n(x)$ 代替 $f(x)$ 进行积分.

$$I_n(f) = \int_a^b L_n(x)dx = \sum_{i=0}^n \left(\int_a^b l_i(x)dx \right) f(x_i)$$

Lagrange 插值的误差表达式为:

$$\begin{aligned} R_n(x) &= \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i) \\ &= f[x_0, x_1, \dots, x_n, x] \prod_{i=0}^n (x - x_i) \end{aligned}$$

误差表达式:

$$\begin{aligned} I(f) - I_n(f) &= \int_a^b R_n(x) dx \\ &= \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i) dx \\ &= \int_a^b f[x_0, x_1, \dots, x_n, x] \prod_{i=0}^n (x - x_i) dx \end{aligned}$$

可以看出, n 阶插值多项式形式的数值积分至少有 n 阶代数精度.

6.1.2 Newton-Cotes 积分

选取等距剖分的节点: 对区间 $[a, b]$ n 等分, 步长 $h = \frac{b-a}{n}$, 节点 $x_i = a + ih$, $i = 0, 1, \dots, n$.

记 Newton-Cotes 系数为:

$$c_i = \frac{1}{b-a} \int_a^b l_i(x) dx$$

它们满足:

$$\sum_{i=0}^n c_i = 1$$

可以算得, $n \leq 7$ 时, 所有的 c_i 都是正的, 因此积分是稳定的. 当 $n \geq 8$ 时, 积分的稳定性不能保证.

梯形积分

当 $n = 1$ 时, 即以 $(a, f(a))$ 和 $(b, f(b))$ 为插值节点构造线性函数 $L_1(x)$, 有:

$$I_1(f) = \int_a^b L_1(x) dx = \frac{b-a}{2} [f(a) + f(b)]$$

梯形积分代数精度:

1、直接计算 (第二个等式运用了积分中值定理):

$$\begin{aligned} E_1(f) &= \int_a^b \frac{f''(\xi(x))}{2!} (x-a)(x-b) dx \\ &= \frac{f''(\eta)}{2!} \int_a^b (x-a)(x-b) dx \\ &= -\frac{(b-a)^3}{12} f''(\eta) \end{aligned}$$

因此有 1 阶代数精度. 2、代入 $f(x) = x$, 发现 $I(x) - I_1(x) = 0$; 代入 $f(x) = x^2$, 发现 $I(x^2) - I_1(x^2) \neq 0$. 因此梯形公式有 1 阶代数精度.

Simpson 公式 (抛物线)

当 $n = 2$ 时,

$$I_2(f) = \int_a^b L_2(x) dx = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

Simpson 积分代数精度: 将 $f(x) = x^k$, $k = 0, 1, 2, \dots$ 逐一带入检验, 知 Simpson 积分有 3 阶代数精度.

Simpson 积分误差推导:

构造 3 次多项式 $P_3(x)$, 它满足:

$$P_3(a) = f(a), \quad P_3\left(\frac{a+b}{2}\right) = f\left(\frac{a+b}{2}\right), \quad P_3(b) = f(b), \quad P_3'\left(\frac{a+b}{2}\right) = f'\left(\frac{a+b}{2}\right)$$

那么:

$$\begin{aligned} E_2(f) &= I(f) - I_2(f) \\ &= I(f) - I(P_3) + I_2(P_3) - I_2(f) \\ &= I(f) - I(P_3) \\ &= \int_a^b \frac{f^{(4)}(\eta(x))}{4!} (x-a) \left(x - \frac{a+b}{2}\right)^2 (x-b) dx \\ &= \frac{f^{(4)}(\xi)}{4!} \int_a^b (x-a) \left(x - \frac{a+b}{2}\right)^2 (x-b) dx \\ &= -\frac{(b-a)^3}{2880} f^{(4)}(\xi) \end{aligned}$$

普适的代数精度分析

不加证明地给出以下结果:

当 n 为奇数时: $f \in C^{n+1}[a, b]$, 具有 n 阶代数精度, 误差满足:

$$E_n(f) = \frac{f^{(n+1)}(\eta)}{(n+1)!} \int_a^b \prod_{i=0}^n (x - x_i) dx$$

当 n 为偶数时: $f \in C^{n+2}[a, b]$, 具有 $n+1$ 阶代数精度, 误差满足:

$$E_n(f) = \frac{f^{(n+2)}(\eta)}{(n+2)!} \int_a^b x \prod_{i=0}^n (x - x_i) dx$$

6.1.3 复化积分

复化梯形积分

对 $[a, b]$ 作等距分割, $h = \frac{b-a}{n}$, $x_i = a + ih$, $i = 0, 1, \dots, n$, 在每一个小区间上应用梯形公式, 有:

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

截断误差:

$$E_n(f) = -\frac{(b-a)^3}{12n^2} f''(\xi)$$

给定误差时的最小区间个数:

$$n \geq \left\lceil \sqrt{\frac{(b-a)^3}{12\varepsilon} \max_{a \leq x \leq b} |f''(x)|} \right\rceil + 1$$

复化 Simpson 积分

对 m 个小区间分别作 Simpson 积分. 取等距节点共 $\boxed{n=2m}$ 个, $h = \frac{b-a}{n}$, $x_i = a + ih$, $i = 0, 1, \dots, n$.

$$S_n(f) = \frac{h}{3} \left[f(a) + 4 \sum_{i=0}^{m-1} + 2 \sum_{i=1}^{m-1} + f(b) \right]$$

节点系数如下所示:

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	4	1						
		1	4	1				
				1	4	1		
						1	4	1
1	4	2	4	2	1	2	4	1

首尾节点为 1, 奇数节点为 4, 偶数节点为 2.

截断误差:

$$E_n(f) = -\frac{(b-a)^5}{2880m^4} f^{(4)}(\xi) = -\frac{(b-a)^5}{180n^4} f^{(4)}(\xi)$$

给定误差时的最小区间个数:

$$m \geq \left\lceil \sqrt[4]{\frac{(b-a)^5}{2880\varepsilon} \max_{a \leq x \leq b} |f^{(4)}(x)|} \right\rceil + 1$$

随着 n 增大, 复化梯形或者复化 Simpson 积分公式都能收敛到 $I(f) = \int_a^b f(x)dx$. 若上界较大不易估计, 常采用自动控制误差算法.

6.1.4 自动控制误差计算

复化梯形

$$\begin{cases} I(f) - T_n(f) = -\frac{(b-a)^3}{12n^2} f''(\xi) \\ I(f) - T_{2n}(f) = -\frac{(b-a)^3}{12(2n)^2} f''(\eta) \\ f''(\xi) \approx f''(\eta) \end{cases}$$

故:

$$I(f) - T_{2n}(f) \approx \frac{1}{3}(T_{2n}(f) - T_n(f))$$

因此, 对给定的误差限 ε , 只需要 $|T_{2n}(f) - T_n(f)| < 3\varepsilon$ 就能保证 $|I(f) - T_{2n}(f)| < \varepsilon$, 即 $T_{2n}(f)$ 能够满足误差要求. 否则, 点的数量加倍, 再进行计算、判断.

复化 Simpson

$$\begin{cases} I(f) - S_n(f) = -\frac{(b-a)^5}{180m^4} f^{(4)}(\xi) \\ I(f) - S_{2n}(f) = -\frac{(b-a)^5}{180(2m)^4} f^{(4)}(\eta) \\ f^{(4)}(\xi) \approx f^{(4)}(\eta) \end{cases}$$

故：

$$I(f) - S_{2n}(f) \approx \frac{1}{15} (S_{2n}(f) - S_n(f))$$

因此，对给定的误差限 ε ，只需要 $|S_{2n}(f) - S_n(f)| < 15\varepsilon$ 就能保证 $|I(f) - S_{2n}(f)| < \varepsilon$ ，即 $S_{2n}(f)$ 能够满足误差要求。否则，点的数量加倍，再进行计算、判断。

6.1.5 Romberg 积分算法（外推算法）

在复化梯形积分的自动控制误差算法中有：

$$I(f) - T_{2n}(f) \approx \frac{1}{3} (T_{2n}(f) - T_n(f))$$

移项：

$$\begin{aligned} I(f) &\approx T_{2n}(f) + \frac{1}{3} (T_{2n}(f) - T_n(f)) \\ &= \frac{4}{3} T_{2n}(f) - \frac{1}{3} T_n(f) \\ &= S_n(f) \end{aligned}$$

于是得到了 Simpson 积分公式 $S_n(f)$ ，截断误差由 $O(n^{-2})$ 提高到了 $O(n^{-4})$ 。

同理，复化 Simpson 积分：

$$I(f) - S_{2n}(f) \approx \frac{1}{15} (S_{2n}(f) - S_n(f))$$

移项：

$$\begin{aligned} I(f) &\approx S_{2n}(f) + \frac{1}{15} (S_{2n}(f) - S_n(f)) \\ &= \frac{16}{15} S_{2n}(f) - \frac{1}{15} S_n(f) \\ &= C_n(f) \end{aligned}$$

于是得到了 Cote's 积分公式 $C_n(f)$ ，截断误差由 $O(n^{-4})$ 提高到了 $O(n^{-6})$ 。

再来一次：

$$R_n(f) = C_{2n}(f) + \frac{1}{63} (C_{2n}(f) - C_n(f)) = \frac{64}{63} C_{2n}(f) - \frac{1}{63} C_n(f)$$

得到了误差为 $O(n^{-8})$ 的 Romberg 积分公式。

以上方法就是外推法，每次获得的新公式具有更高的误差阶。

Romberg 计算公式：

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}$$

其中 2^k 是区间个数， j 标明积分公式：1 为梯形，2 为 Simpson，3 为 Cotes，4 为 Romberg...

6.2 重积分 *

化为累次积分, 然后应用单变量积分的计算方法.

6.3 Gauss 积分

本节中, 积分节点表示为 x_1, x_2, \dots, x_n 共 n 个, 与其他章节不同.

6.3.1 绪论

定义. 带权函数积分

权函数 $W(x) \in C[a, b]$, $W(x) \geq 0$ ($\forall x \in [a, b]$)

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

定理. 插值型积分的最高精度阶数

对 $I(f) = \int_a^b W(x)f(x)dx$, 具有 n 个节点 x_1, x_2, \dots, x_n 的数值积分公式:

$$\begin{cases} I_n(f) = \sum_{i=1}^n a_i f(x_i) \\ a_i = \int_a^b W(x)f(x)dx \end{cases}$$

其代数精度不超过 $2n - 1$ 阶.

用反证法证明: 对 $f(x) = \prod_{i=1}^n (x - x_i)^2 \in \mathbb{P}_{2n}$ (\mathbb{P}_n 指所有 n 阶多项式构成的集合), $I(f) = \int_a^b f(x)dx > 0$, 而 $I_n(f) = \sum_{i=1}^n a_i f(x_i) = 0$, 因此 $I_n(f)$ 的精度必不可能达到 $2n$ 阶.

定义. 带权内积

多项式函数构成的线性空间中, 可以定义关于权函数 $W(x)$ 的内积:

$$(f, g) := \int_a^b W(x)f(x)g(x)dx$$

定义正交: $f(x) \perp g(x) \Leftrightarrow (f, g) = 0$

利用 Gram-Schmidt 正交化, 可由一组基 $\{1, x, x^2, \dots\}$ 得到正交基 $\{p_0(x), p_1(x), \dots\}$. 当权函数 $W(x) = 1$ 时, $p_n(x) = P_n(x)$ 就是 Legendre 多项式.

6.3.2 Gauss 积分的性质

Gauss 积分的步骤.

- 1、利用 Gram-Schmidt 正交化求出区间 $[a, b]$ 上权函数为 $W(x)$ 的正交多项式 $p_n(x) \perp \mathbb{P}_{n-1}(x)$;
- 2、求出 $p_n(x)$ 的 n 个零点 $\{x_1, x_2, \dots, x_n\}$ 作为 Gauss 积分节点;
- 3、计算积分系数:

$$a_i = \int_a^b W(x)l_i(x)dx$$

其中 $l_i(x)$ 是对应于插值节点 x_i 的 Lagrange 插值函数. Gauss 积分公式为

$$G_n(f) = \sum_{i=1}^n a_i f(x_i)$$

Gauss 积分的误差:

$$R(x) = \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b W(x) \prod_{i=1}^n (x - x_i)^2 dx$$

定理. Gauss 积分代数精度

Gauss 积分具有 $2n - 1$ 阶代数精度.

证明: 以 $p_n(x)$ 的 n 个零点为节点构造 $n - 1$ 次 Lagrange 插值多项式, 记为 $L_{n-1}(x)$. 则 Gauss 积分的误差为:

$$\begin{aligned} E(f) &= I(f) - G_n(f) \\ &= I(f) - I(L_{n-1}(x)) + G_n(L_{n-1}(x)) - G_n(f) \\ &= I(f) - I(L_{n-1}(x)) \\ &= \int_a^b f[x_1, x_2, \dots, x_n, x] \prod_{i=1}^n (x - x_i) dx \end{aligned}$$

对 $\forall f \in \mathbb{P}_{2n-1}$, $f[x_1, x_2, \dots, x_n, x] \in \mathbb{P}_{n-1}$, 而 $\prod_{i=1}^n (x - x_i)$ 的零点分布与 $p_n(x)$ 完全相同, 故二者只差一常数倍, 则 $\prod_{i=1}^n (x - x_i) \perp \mathbb{P}_{n-1}$, 因此 $E(f) = 0$, 即 $G_n(f)$ 具有 $2n - 1$ 阶代数精度.

Gauss 积分的性质:

1、积分系数 $a_i > 0$, 且满足:

$$\sum_{i=1}^n a_i = \int_a^b W(x) dx$$

2、 $f \in C[a, b]$ 则:

$$\lim_{n \rightarrow +\infty} G_n(f) = I(f)$$

这是其他插值型数值积分不具备的性质.

6.4 数值微分

1、向前差商:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

误差:

$$R = f'(x_0) - \frac{f(x_0 + h) - f(x_0)}{h} = -\frac{h}{2} f''(\xi) = O(h)$$

2、向前差商:

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h}$$

误差：

$$R = f'(x_0) - \frac{f(x_0) - f(x_0 - h)}{h} = \frac{h}{2} f''(\xi) = O(h)$$

3、中心差商：

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

误差：

$$R = f'(x_0) - \frac{f(x_0 + h) - f(x_0 - h)}{2h} = -\frac{h^2}{6} f^{(3)}(\xi) = O(h^2)$$

步长 $h > 0$ ，太大则会由于割线与切线斜率相差大而误差大，太小则会因舍入误差大而误差大，故存在一个最合适的步长.

插值型数值微分：利用插值函数构造插值多项式，用它的导数近似原来函数的导数.

第七章 常微分方程数值解

本章主要讨论如下常微分方程的初值问题：

$$\begin{cases} y'(x) = f(x, y) \\ y(a) = y_0 \end{cases}, \quad a \leq x \leq b \quad (7.1)$$

求解基本步骤：

- 1、对区间作分割. 一般为等距分割. 记 $y(x_i)$ (真值) 的近似值 (数值解) 为 y_i , 称为格点函数.
- 2、建立差分格式 (递推关系) 这个方程应满足：解存在且唯一；稳定、收敛；相容.
- 3、解差分方程.

常微分方程基本问题：

- 1、收敛性：步长充分小时，所得到的数值解趋近于真实解.
- 2、误差：局部截断误差、误差主项 (误差中阶数最低的非零项)、精度阶数.
- 3、稳定性：误差是否会积累并扩大.

三种基本方法：

- 1、基于数值微分：Euler 公式.
- 2、基于数值积分：Euler 公式、梯形公式、线性多步法.
- 3、基于 Taylor 展开：Runge-Kutta 方法.

7.1 Euler 公式

7.1.1 基于数值微分的 Euler 公式

向前差商

$$y'(x_n) \approx \frac{y(x_{n+1}) - y(x_n)}{h}$$
$$\boxed{y_{n+1} = y_n + hf(x_n, y_n)} \quad (7.2)$$

向后差商

$$y'(x_{n+1}) \approx \frac{y(x_{n+1}) - y(x_n)}{h}$$
$$\boxed{y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})} \quad (7.3)$$

中心差商

$$y'(x_n) \approx \frac{y(x_{n+1}) - y(x_{n-1}))}{2h}$$

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n) \quad (7.4)$$

由于中心差商不是一个稳定的迭代格式，所以实际计算中几乎不使用。

7.1.2 Euler 公式的收敛性

定义. 局部截断误差、误差阶

认为由 $y_n, y_{n-1}, y_{n-2}, \dots$ 递推 y_{n+1} 的公式中除了 y_{n+1} 项都是准确的，也即 $y_n = y(x_n), y_{n-1} = y(x_{n-1}), \dots$ ，计算出的 $T_{n+1} = y(x_{n+1}) - y_{n+1}$ 称为局部截断误差。如果局部截断误差满足：

$$T_{n+1} = O(h^{p+1})$$

则称给定方法的局部截断误差是 p 阶的。

可以看出：向前 Euler 公式是 1 阶精度格式，向后 Euler 公式也是 1 阶精度格式，中心差商格式是 2 阶精度格式。

7.1.3 基于数值积分的 Euler 公式

对常微分方程

$$\frac{dy}{dx} = f(x, y)$$

等式两边在 $[x_n, x_{n+1}]$ 上积分，得到：

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))dx$$

用数值积分近似公式计算积分 $\int_{x_n}^{x_{n+1}} f(x, y(x))dx$ ：

1、使用左矩形公式得到向前 Euler 公式：

$$y_{n+1} = y_n + hf(x_n, y_n)$$

2、使用右矩形公式得到向后 Euler 公式：

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

3、使用梯形近似公式：

$$\int_{x_n}^{x_{n+1}} f(x, y(x))dx \approx \frac{1}{2}(x_{n+1} - x_n)(f(x_n, y_n) + f(x_{n+1}, y_{n+1}))$$

则得到梯形公式：

$$y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, y_{n+1})) \quad (7.5)$$

在梯形公式中，可以使用预估-校正方法：

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

也即改进型 Euler 公式：

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_n + f(x_n, y_n))] \quad (7.6)$$

7.2 线性多步法

对常微分方程

$$\frac{dy}{dx} = f(x, y(x))$$

在区间 $[x_{n-p}, x_{n+1}]$ 上积分得：

$$y(x_{n+1}) = y(x_{n-p}) + \int_{x_{n-p}}^{x_{n+1}} f(x, y(x))dx$$

用数值积分近似 $\int_{x_{n-p}}^{x_{n+1}} f(x, y)dx$ 来构造线性多步格式：

1、显式格式：用节点 $x_n, x_{n-1}, \dots, x_{n-q}$ 来构造插值函数计算积分：

$$\begin{cases} y_{n+1} = y_{n-p} + \sum_{i=0}^q a_{n-i} f(x_{n-i}, y_{n-i}) \\ a_{n-i} = \int_{x_{n-p}}^{x_{n+1}} \frac{(x - x_n) \cdots (x - x_{n-i+1})(x - x_{n-i-1}) \cdots (x - x_{n-q})}{(x_{n-i} - x_n) \cdots (x_{n-i} - x_{n-i+1})(x_{n-i} - x_{n-i-1}) \cdots (x_{n-i} - x_{n-q})} dx \end{cases}$$

2、隐式格式：用节点 $x_{n+1}, x_n, \dots, x_{n-q+1}$ 来构造插值函数计算积分：

$$\begin{cases} y_{n+1} = y_{n-p} + \sum_{i=0}^q a_{n+1-i} f(x_{n+1-i}, y_{n+1-i}) \\ a_{n+1-i} = \int_{x_{n-p}}^{x_{n+1}} \frac{(x - x_{n+1}) \cdots (x - x_{n-i+2})(x - x_{n-i}) \cdots (x - x_{n+1-q})}{(x_{n+1-i} - x_{n+1}) \cdots (x_{n+1-i} - x_{n-i+2})(x_{n+1-i} - x_{n-i}) \cdots (x_{n+1-i} - x_{n+1-q})} dx \end{cases}$$

利用节点构造 Lagrange 插值函数，用它代替 $f(x, y)$ 进行积分即可。

在线性多步格式中，有两个参数： p 描述积分区间， q 控制插值节点。

误差阶通过对得到的表达式在 $x(n-p)$ 点作 Taylor 展开而得到。

7.3 Runge-Kutta 方法

7.3.1 基本思想

Runge-Kutta 方法是基于 Taylor 展开的。

对 $y(x)$ 在 x_n 处作 Taylor 展开, 取 $x = x_{n+1}$:

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \cdots$$

注意到:

$$\begin{cases} y'(x) = f(x, y) \\ y''(x) = f_x(x, y) + f_y(x, y)f(x, y) \\ \dots \end{cases}$$

取前三项, 构造差分形式:

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

取的阶数越多, 方法的误差阶数越高, 但是计算量也越大.

为了避免计算 $f(x, y)$ 的高阶导数, Runge-Kutta 方法使用 $f(x, y)$ 在点 (x_n, y_n) 和它附近的另一个点 $(x_n + ah, y_n + bhf(x_n, y_n))$ 的函数值做线性组合后来达到相同的误差阶.

7.3.2 二阶 Runge-Kutta 方法

通过选取适当的参数 a, b, c_1, c_2 使得:

$$\begin{aligned} & c_1 f(x_n, y(x_n)) + c_2 f(x_n + ah, y(x_n) + bhf(x_n, y(x_n))) \\ & \approx f(x_n, y(x_n)) + \frac{h}{2} (f_x(x_n, y_n) + f_y(x_n, y_n)f(x_n, y_n)) \end{aligned}$$

进行 Taylor 展开:

$$\begin{aligned} & = c_1 f(x_n, y(x_n)) + c_2 f(x_n + ah, y(x_n) + bhf(x_n, y(x_n))) \\ & = c_1 f(x_n, y(x_n)) + c_2 \left[f(x_n, y(x_n)) + ahf_x(x_n, y(x_n)) + bhf_y(x_n, y(x_n))f(x_n, y(x_n)) + O(h^2) \right] \\ & = (c_1 + c_2)f(x_n, y(x_n)) + \frac{h}{2} [2c_2af_x(x_n, y(x_n)) + 2c_2bf_y(x_n, y(x_n))f(x_n, y(x_n))] + O(h^2) \\ & = f(x_n, y(x_n)) + \frac{h}{2} (f_x(x_n, y_n) + f_y(x_n, y_n)f(x_n, y_n)) + O(h^2) \end{aligned}$$

比较系数, 得到:

$$\begin{cases} c_1 + c_2 = 1 \\ 2c_2a = 1 \\ 2c_2b = 1 \end{cases}$$

得到二阶差分格式:

$$y_{n+1} = y_n + h \left[c_1 f(x_n, y_n) + c_2 f(x_n + ah, y_n + bhf(x_n, y_n)) \right] \quad (7.7)$$

通常也写为:

$$\begin{cases} y_{n+1} = y_n + h(k_1 + c_2k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + ah, y_n + bhk_1) \end{cases} \quad (7.8)$$

同时, 观察以上推导过程, 不难看出二阶 Runge-Kutta 方法的截断误差为 $O(h^3)$, 误差阶为 2 阶.

例如, 取 $c_1 = c_2 = 1/2$, $a = b = 1$ 就得到了改进 Euler 公式 7.6.

7.3.3 高阶 Runge-Kutta 方法

使用类似的方法，可以得到更高阶的 Runge-Kutta 公式.

常用的三阶 Runge-Kutta 公式：

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_2 + k_3) \\ k_1 = f(x_n, y_n) \\ k_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\ k_3 = f(x_n + h, y_n - hk_1 + 2hk_2) \end{cases}$$

常用的四阶 Runge-Kutta 公式：

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(x_n, y_n) \\ k_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\ k_3 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2\right) \\ k_4 = f(x_n + h, y_n + hk_3) \end{cases}$$

7.4 方程组与高阶微分方程

7.4.1 方程组

微分方程组初值问题：

$$\begin{cases} \frac{dy_1}{dx} = f_1(x, y_1, \dots, y_m) \\ \dots \\ \frac{dy_m}{dx} = f_m(x, y_1, \dots, y_m) \\ y_1(a) = \eta_1 \\ \dots \\ y_m(a) = \eta_m \end{cases}$$

写成向量的形式：

$$\begin{cases} \frac{dY}{dx} = F(x, Y) \\ Y(a) = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_m \end{pmatrix} \end{cases}$$

单个微分方程的各种方法都可以类似地运用过来.

7.4.2 高阶微分方程

$$\begin{cases} y^{(m)}(x) = f(x, y, y', \dots, y^{(m-1)}) \\ y(a) = \eta_1 \\ y'(a) = \eta_2 \\ \dots \\ y^{(m-1)}(a) = \eta_m \end{cases} \Rightarrow \begin{cases} \frac{dy_1}{dx} = y_2 \\ \dots \\ \frac{dy_{m-1}}{dx} = y_m \\ \frac{dy_m}{dx} = f(x, y, y', \dots, y^{(m-1)}) \\ y_1(a) = \eta_1 \\ \dots \\ y_m(a) = \eta_m \end{cases}$$

高阶方程可化为一阶方程组.

7.5 常微分方程解的稳定性

7.5.1 定义

初值问题:

$$\begin{cases} \frac{dy}{dx} = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

定义. (Lyapunov) 稳定

设 $y = y(x)$ 为方程取初值 y_0 时的解. 若对 $\forall \varepsilon > 0$, $\exists \delta(\varepsilon) > 0$ s.t. 只要 $|z_0 - y(x_0)| < \delta$, 方程取初值 z_0 的解 $z = z(x)$ 就存在唯一, 且满足:

$$|z(x) - y(x)| < \varepsilon \quad (\forall x \geq x_0)$$

那么称解 $y(x)$ 是 (Lyapunov) 稳定的.

只要初值变化足够小, $y(x)$ 的变化就足够小. 例如: $\frac{dy}{dx} = -\frac{x}{y}$ 它的解稳定, 但不渐进稳定.

定义. 渐进稳定

若解 $y(x)$ 是稳定的, 且满足:

$$\lim_{x \rightarrow +\infty} (z(x) - y(x)) = 0$$

则称 $y(x)$ 是渐进稳定的.

初值变化足够小, 当自变量趋向正无穷时, 解的变化仍足够小. 例如: $\frac{dy}{dx} = \lambda y$ ($\Re(\lambda) < 0$), 它的解都是渐进稳定的.

7.5.2 差分方程的绝对稳定性

差分格式的稳定性建立在微分方程初值问题解本身渐进稳定的基础之上.

设求微分方程数值解时采用差分格式为:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f(x_{n+j}, y_{n+j}) \quad (7.9)$$

定义. 绝对稳定

对于某差分格式, 若利用它求下述微分方程:

$$\frac{dy}{dx} = \lambda y \quad (\Re(\lambda) < 0) \quad (7.10)$$

的解时, 对任意初值, 总存在左半复平面上的一个区域, 当 $z := \lambda h$ 落在这个区域时, 差分方程的解趋于 0 (也即, $y_n \rightarrow 0$ ($n \rightarrow +\infty$)), 则称这种差分格式绝对稳定. 复平面上的这一区域称为差分格式的稳定区域.

用差分格式(7.9)求解上述微分方程(7.10), 得到:

$$\sum_{j=0}^k \alpha_j y_{n+j} = \lambda h \sum_{j=0}^k \beta_j y_{n+j} \quad (7.11)$$

由此得到差分格式(7.9)的绝对稳定条件:

定理. 差分格式(7.9)的绝对稳定条件 (充要条件)

存在左半复平面上的一个子区域 D , 当 $z := \lambda h$ 落在此区域时, 齐次差分方程(7.11)的特征方程:

$$\sum_{j=0}^k (\alpha_j - \lambda h \beta_j) \xi^j = 0$$

的所有根的模都小于 1.

误差

设给定的初始值 y_i ($0 \leq i \leq k-1$) 有误差 e_i 时, 由差分格式(7.9)得到的解 $\{\tilde{y}_{n+j}\}$ 与初始值没有误差时的 $\{y_{n+j}\}$ 之差为 e_{n+j} , 则 e_{n+j} 同样满足:

$$\sum_{j=0}^k \alpha_j e_{n+j} = \lambda h \sum_{j=0}^k \beta_j e_{n+j}$$

因此绝对稳定的差分格式会导致 $e_n \rightarrow 0$ ($n \rightarrow +\infty$), 即初值误差对数值求解结果没有太大影响.

Euler 公式的绝对稳定性

向前 Euler 公式 (显式) 绝对稳定, 稳定域为以 -1 为圆心的单位圆.

向后 Euler 公式 (隐式) 绝对稳定, 稳定域为左半复平面.

中心 Euler 公式不稳定.