

NA Review

NA Review

Chapter 1 Mathematical Preliminaries

Chapter 2 Solutions of Equations in One Variable

Error Analysis for Iterative Methods 迭代法误差分析

Bisection Method 二分法

Fixed-Point Iteration 不动点法

Newton's Method 牛顿迭代法

优化牛顿迭代法

Accelerating Convergence 加速收敛

AITKEN Δ^2 方法

Steffensen 方法

Chapter 6 Direct Methods for Solving Linear Systems

Gaussian elimination with backward substitution 高斯消元法

Pivoting Strategies 选主元策略

Partial Pivoting

Scaling Partial Pivoting 比例因子选主元

Complete Pivoting 全主元选取

Matrix Factorization 矩阵分解

★ LU Factorization

Special Types of Matrices

Strictly Diagonally Dominant Matrices 严格对角占优

★ Positive Definite Matrices 正定矩阵

Tridiagonal Matrices 三对角矩阵

Chapter 7 Iterative Techniques in Matrix Algebra

Norms of Vectors and Matrices

Vector Norms

Matrix Norms

Eigenvalues and Eigenvectors 特征值和特征向量

Spectral Radius 谱半径

★ Iterative Techniques for Solving Linear Systems 迭代法求解线性方程组

Jacobi迭代法：直接分别表示出每一个元素的值即可

Gauss-Seidel迭代法 计算下一个值时 需要带入已经算出的值

收敛性 类似于第二章

Relaxation Methods 松弛法 SOR

Error Bounds and Iterative Refinement 条件数 condition number

Chapter 9 Approximating Eigenvalues

Power Method

Inverse power Method

Chapter 3 Interpolation and Polynomial Approximation

Interpolation and Lagrange Polynomials 拉格朗日插值

Neville 迭代插值

Divided Differences 差商

差商型 Newton 插值多项式 Newton's Interpolatory Divided-Difference formula

Hermite Interpolation | Hermite 插值

Piecewise polynomial approximation 分段多项式逼近

Cubic Spline Interpolation 三样条插值

Chapter 8 Approximation Theory

Discrete Least Squares Approximation 离散最小二乘逼近

Orthogonal Polynomials and Least Squares Approximation 正交多项式与最小二乘逼近

Chebyshev Polynomials and Economization of Power Series 切比雪夫多项式与幂级数的缩减

Chapter 4 Numerical Differentiation and Integration

Numerical Differentiation 数值微分

Elements of Numerical Integration 数值积分

Composite Numerical Integration 复合数值积分

Romberg Integration

Richardson's Extrapolation 外推法

Adaptive Quadrature Methods p188

Gauss Quadrature

Chapter 5 Initial-Value Problems for Ordinary Differential Equations

The Elementary Theory of Initial-Value Problems

Euler's Method

Implicit Euler Method 隐式欧拉法

Trapezoidal Method

Double-step Method

Higher Order Taylor Methods

Runge-Kutta Methods

二阶 runge-kutta (single step)

高阶 Runge-Kutta

Multistep Methods

Adams Bashforth 显示M步 具体推导 p258

Adams-Moulton 隐式m步

Adams 预测矫正

通过泰勒展开获得

Higher-Order Equations and Systems of Differential Equations 高阶常微分方程组

Stability 稳定性

Chapter 1 Mathematical Preliminaries

Truncation Error: 有限的求和来近似无穷级数的和

$$1 + \dots + \infty \rightarrow 1 + \dots + n$$

Roundoff Error: 有限位数浮点数代替无限位数产生的误差

- **Chopping:** $0.1119 \rightarrow 0.111$
- **Rounding:** $0.1119 \rightarrow 0.112$

Absolute error: $|p^* - p|$

Relative error: $\frac{|p^* - p|}{|p|}$

chopping:

$$\frac{|y - fl(y)|}{|y|} = \frac{0.d_{k+1}d_{k+2}\dots \times 10^{n-k}}{0.d_1d_2d_3\dots d_k \times 10^n} = \frac{0.d_{k+1}d_{k+2}\dots}{0.d_1d_2d_3\dots d_k} \times 10^{-k} < \frac{1}{0.1} \times 10^{-k} = 10^{-k+1}$$

Rounding:

$$\frac{|y - fl(y)|}{|y|} \leq \frac{0.5 \times 10^{n-k}}{0.d_1d_2d_3\dots d_k \times 10^n} = \frac{0.5}{0.1} \times 10^{-k} = 0.5 \times 10^{-k+1}$$

误差产生的情况:

- Subtraction of nearly equal numbers will cause a cancellation of significant digits. (两个近乎相等的数相减会导致有效数字的相消)
- Dividing by a number with small magnitude (除以一个小数 (或者乘以一个大数) 会放大绝对误差)

秦九韶算法降低误差 Horner's Method

Stability:

如果初始数据的**小变化**会导致最终结果的**小变化**，则称为**稳定**(stable)，否则称为**不稳定**(unstable)。

一个算法，如果只有在某些初始数据的选择下才是稳定的，则称为**条件稳定**(conditionally stable)。

Significant Digits: $\frac{|p^* - p|}{|p|} \leq 5 \times 10^{-t}$

假设 $E_0 > 0$ 是初始误差(initial error), E_n 是第 n 步的误差。

- 如果 $E_n \approx CnE_0$, 则称为 **线性增长**(linear growth)。

线性增长的误差通常是无法避免的, 当 C 和 E_0 都很小的时候, 结果通常是可以接受的。

- 如果 $E_n \approx C^n E_0$, 则称为 **指数增长**(exponential growth)。

指数增长的误差应该避免, 因为即使 E_0 很小, C^n 也会变得很大。这会导致不可接受的不准确性。

Chapter 2 Solutions of Equations in One Variable

diverge 发散 converge 收敛

Error Analysis for Iterative Methods 迭代法误差分析

假设 $\{p_n\}_{n=0}^{\infty}$ 收敛于 p , 其中对 $\forall p_n \neq p$ 。如果存在正数 λ 和 α , 使得

$$\lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} = \lambda$$

则称 $\{p_n\}_{n=0}^{\infty}$ 是 α 阶收敛的 (converges to p of order α), λ 称为 **渐进误差常数**(asymptotic error constant)。

Linearly convergent 线性收敛: $\alpha = 1$

Quadratically convergent 二次收敛: $\alpha = 2$

★ 高阶收敛性的序列收敛速度更快

收敛阶数计算 calculation of order of convergence: 通俗理解最低阶不为 0 的导数 $\lambda = \frac{g^{(a)}(p)}{a!}$

设 p 为 $g(x)$ 的不动点, 若

- 存在常数 $\alpha \geq 2$ 使得 $g \in C^{\alpha}[p - \delta, p + \delta]$
- $g'(p) = g''(p) = \dots = g^{(\alpha-1)}(p) = 0$
- $g^{(\alpha)}(p) \neq 0$

那么不动点迭代生成的 $\{p_n\}$ 以 α 阶收敛到 p 。

Bisection Method 二分法

中点计算: $a_1 + \frac{b_1 - a_1}{2}$

停止条件: $|T_n - t| \leq \frac{b-a}{2^n} < \epsilon$

优点: 简单一定收敛

缺点: 无法找到重根

Fixed-Point Iteration 不动点法

$$\begin{aligned} f(x) = 0 &\leftrightarrow x = g(x) \\ P_n &= g(p_{n-1}) \end{aligned}$$

不动点存在性、唯一性充分条件:

a. $g \in C[a, b]$ 且 $g(x) \in [a, b], \forall x \in [a, b]$, 则 g 在 $[a, b]$ 上有不动点。

b. $|g'(x)| \leq k < 1, \forall x \in (a, b)$, 则该不动点是唯一的。

则对于任意 $p_0 \in [a, b]$, 不动点迭代序列 $p_{n+1} = g(x_n)$ 收敛于不动点。

满足以上条件一定能说明存在, 不满足不能说明不存在。

误差按照k比例依次下降 k越小收敛越快

$$|p_n - p| \leq \frac{1}{1-k} |p_{n+1} - p_n| \quad |p_n - p| \leq \frac{k^n}{1-k} |p_1 - p_0|$$

$$\begin{aligned} |p_{n+1} - p_n| &= |g(p_n) - g(p_{n-1})| = |g'(\xi)| |p_n - p_{n-1}| \leq k |p_n - p_{n-1}| \\ &\leq k^2 |p_{n-1} - p_{n-2}| \leq \dots \leq k^n |p_1 - p_0| \end{aligned}$$

收敛性

$$\begin{aligned} p_{n+1} - p &= g(p_n) - g(p) = g'(\xi)(p_n - p) \Rightarrow \frac{|p_{n+1} - p|}{|p_n - p|} = |g'(\xi)| \\ &\Rightarrow \lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|} = \lim_{n \rightarrow \infty} |g'(\xi)| = |g'(p)| \end{aligned}$$

线性收敛 $\lambda = g'(p)$

多重根情况: $g'(p) = g''(p) = \dots = g^{m-1}(p) = 0$, 于是收敛阶数是 m , 渐进误差常数是 $\lambda = \frac{f^m(p)}{m!}$

Newton's Method 牛顿迭代法

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

初值存在定理：保证一阶导不为0

定理：设 $f \in C^2[a, b]$, 如果 $p \in [a, b]$ 满足 $f(p) = 0$ 且 $f'(p) \neq 0$, 则存在 $\delta > 0$, 使得对任何初始值 $p_0 \in (p - \delta, p + \delta)$, 牛顿迭代法产生收敛于 p 的序列 $\{p_n\}_{n=1}^\infty$.

收敛性质：

$$\begin{aligned} 0 = f(p) &= f(p_n) + f'(p_n)(p - p_n) + \frac{f''(\xi)}{2}(p - p_n)^2 \\ \Rightarrow p &= p_n - \frac{f(p_n)}{f'(p_n)} - \frac{f''(\xi)}{2f'(p_n)}(p - p_n)^2 \\ \Rightarrow \lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|^2} &= \lim_{n \rightarrow \infty} \frac{|f''(\xi)|}{2|f'(p_n)|} = \frac{|f''(p)|}{2|f'(p)|} \end{aligned}$$

二次收敛 $\lambda = \frac{|f''(p)|}{2|f'(p)|}$

多重根情况： 转变为线性收敛 $g'(p) = 1 - 1/m$

优化牛顿迭代法

优化牛顿法 (用于Lab2): 用 $\mu(x) = \frac{f(x)}{f'(x)}$ 替换原来的 $f(x)$, 即公式变成

$$p_n = p_{n-1} - \frac{\mu(x)}{\mu'(x)} = p_{n-1} - \frac{f(p_{n-1})f'(p_{n-1})}{[f'(p_{n-1})]^2 - f(p_{n-1})f''(p_{n-1})}$$

缺点：计算二阶导 分母 (denominator) 的两项可能相近导致舍入误差大

Accelerating Convergence 加速收敛

AITKEN Δ^2 方法

forward	difference	:
$\Delta p_n = p_{n+1} - p_n$	$\Delta^2 p_n = \Delta(p_{n+1} - p_n) = (p_{n+2} - p_{n+1}) - (p_{n+1} - p_n)$	

我们可以构造序列 $\{\hat{p}_n\}_{n=0}^\infty$, 其中

$$\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n} = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}$$

🌟 加速 p_n 线性收敛

使用流程：

$$\begin{aligned} p_0 & \\ p_1 &= g(p_0) \\ p_2 &= g(p_1) \\ \hat{p}_0 &= \Delta^2(p_0), p_3 = g(p_2) \\ &\dots \end{aligned}$$

Steffensen 方法

$$\begin{aligned} p_0^{(0)}, \quad p_1^{(0)} &= g(p_0^{(0)}), \quad p_2^{(0)} = g(p_1^{(0)}), \\ p_0^{(1)} &= \{\Delta^2\}(p_0^{(0)}), \quad p_1^{(1)} = g(p_0^{(1)}), \quad p_2^{(1)} = g(p_1^{(1)}), \\ p_0^{(2)} &= \{\Delta^2\}(p_0^{(1)}), \dots \dots \end{aligned}$$

Chapter 6 Direct Methods for Solving Linear Systems

Gaussian elimination with backward substitution 高斯消元法

计算次数

在第 k 步，计算 m_{ik} 需要 $(n - k)$ 次除法，更新 a_{ij} 和 b_i 需要考虑 i 从 $k + 1$ 到 n 共 $(n - k)$ 行，每行计算 $n - k$ 个 a_{ij} 和 1 个 b_i (每个都需要 1 次乘法)。因此，第 k 步需要的乘除法数量为

$$(n - k) + (n - k)(n - k + 1) = (n - k)(n - k + 2)$$

因此，化上三角的过程需要的乘除法数量为

$$\sum_{k=1}^{n-1} (n - k)(n - k + 2) = \frac{n^3}{3} + \frac{n^2}{2} + \frac{5}{6}n$$

而 backward-substitution 过程的乘除法数量就很好算了：

$$1 + \sum_{i=1}^{n-1} (n - i + 1) = \frac{n^2}{2} + \frac{n}{2}$$

总的乘除法次数为

$$\frac{n^3}{3} + n^2 + \frac{4}{3}n$$

加减法同理：

加法/减法：

$$\left(\frac{1}{3}n^3 - \frac{1}{3}n\right) + \left(\frac{1}{2}n^2 - \frac{1}{2}n\right) = \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n$$

Povoting Stratgies 选主元策略

除以一个小数 和 乘一个大数对导致误差变大

Partial Pivoting

从第 k 行开始，选取该列中绝对值最大的元素作为主元，然后再进行消元。

- Partial Pivoting: 需要 $O(n^2)$ 的额外比较

Scaling Partial Pivoting 比例因子选主元

从第 k 行开始，选取该列中 $\max\left(\frac{a_{kj}}{s_k}\right)$ 作为主元，然后再进行消元。其中 $s_k = \max|a_{ij}|$

比例因子只在初始过程中计算一次，然后在每次迭代过程中，比例因子也需要参与交换。

- 先找每一行最大的元素
- 在高斯消元过程中 选每一列中最大的结果的行作为主元
- 需要 $O(n^2)$ 的额外比较和 $O(n^2)$ 的额外除法

Complete Pivoting 全主元选取

比例因子每一次都要算一遍 需要 $O(n^3/3)$ 的额外比较

Matrix Factorization 矩阵分解

★ LU Factorization

$$A = LU$$

如果Gauss消去法在线性方程组 $A\vec{x} = \vec{b}$ 没有进行行交换 则可以LU分解为，并且分解唯一

$$\mathbf{L} = \mathbf{L}^{(1)} \mathbf{L}^{(2)} \cdots \mathbf{L}^{(n-1)} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & 0 \\ m_{31} & m_{32} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1 \end{bmatrix}$$

$$\mathbf{U} = \mathbf{A}^{(n)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn}^{(n)} \end{bmatrix}$$

- 分解方法 Doolittle：

直接对A进行高斯消元法转化为上三角方程组，并记录每次如何变化

对于在高斯消元法中需要进行 行交换的矩阵 往往能分解为

p是一个置换举证： $P^{-1} = P^t$

$$A = P^{-1}LU = P^t LU$$

- 先进行矩阵对角化，观察需要进行的行的交换，并完成矩阵P
- 先进行行交换得到 $PA\vec{x} = P\vec{b}$
- 对于 PA 进行LU分解得到 $PA = LU$
- 化简得到 $A = P^t LU$

Special Types of Matrices

Strictly Diagonally Dominant Matrices 严格对角占优

对角线上的元素的绝对值大于该行上其他元素的绝对值之和，则称A为严格对角占优矩阵。

定理：严格对角占优矩阵是非奇异(nonsingular)：存在逆矩阵的。而且，在此情况下，Gauss消去法可用在方程组中以得到唯一解，而且不需要进行或列交换，并且对于舍入误差的增长而言计算是稳定的。一定能LU分解

使用Jacobi和Gauss-Seidel 一定收敛

★ Positive Definite Matrices 正定矩阵

★ 条件:

1. 对称矩阵 symmetric
2. 对于任意非0向量 \vec{x} , $\vec{x}^t A \vec{x} > 0$

定理 充分条件

如果 A 是 $n \times n$ 的正定矩阵, 则

- a. A 是非奇异的。
- b. $a_{ii} > 0$, $i = 1, 2, \dots, n$ 。
- c. $\max_{1 \leq k, j \leq n} |a_{kj}| < \max_{1 \leq i \leq n} |a_{ii}|$, 其中 $k \neq j$ 。
- d. $(a_{ij})^2 < a_{ii}a_{jj}$, $i \neq j$ 。

充要条件

★★ 如果 A 是正定的, 则 A 的所有主子式都是正的。

Gauss消元法不需要进行行列交换, 且舍入误差增长稳定

$A = LDL^T$ 分解

把 $A = LU$ 中的 U 进一步分解为 $D\hat{U}$ 单位上三角矩阵 (对角线元素全1)

对称性质 所以 $A = LDL^T$, L 单位下三角矩阵 D 正对角矩阵

Cholesky分解

$$\begin{aligned} A &= \hat{L}\hat{L}^T \\ \hat{L} &= LD^{1/2} \\ D^{1/2} &= \begin{pmatrix} \sqrt{D_{ii}} & \cdots & \cdots & \cdots \\ 0 & \sqrt{D_{i+1i+1}} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \end{aligned}$$

无论什么分解都可以先使用LU分解, 再进行转化

Tridiagonal Matrices 三对角矩阵

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & 0 & \cdots & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix}$$

定理：假设**A**是三对角矩阵，对每个*i* = 2, 3, …, *n* - 1，有*a_{i,i-1}a_{i,i+1}* ≠ 0，如果|*a₁₁*| > |*a₁₂*|, |*a_{ii}*| > |*a_{i,i-1}*| + |*a_{i,i+1}*|, |*a_{nn}*| > |*a_{n,n-1}*|，则**A**是非奇异的，且在Crout分解中，*l_{ii}*的值都是非零的。

Crout 分解

如果*a_{ii}*=0 算法不能继续下去了 于是这个算法不能解决所有的问题

Crout分解是LU分解的一种特殊情况，我们可以求出具有形式

$$\mathbf{L} = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ 0 & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & l_{n,n-1} & l_{nn} \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 1 & u_{12} & 0 & \cdots & 0 \\ 0 & 1 & u_{23} & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

的三对角矩阵**A**的分解。

通过矩阵乘法，我们可以得到：

$$\begin{cases} a_{11} = l_{11} \\ a_{i,i-1} = l_{i,i-1} \\ a_{i,i} = l_{i,i-1}u_{i-1,i} + l_{i,i} \\ a_{i,i+1} = l_{i,i}u_{i,i+1} \end{cases}$$

在求解部分，我们可以先解**Lz = b**，然后再解**Ux = z**。有伪代码：

Chapter 7 Iterative Techniques in Matrix Algebra

类似于 Chapter 2的思路，只是对于矩阵求解而言

Norms of Vectors and Matrices

Vector Norms

$f(x)=0$ 迭代法求解 通过 $x_n - x$ 的大小结束迭代，再向量中表达为范数，即 **向量之间的距离**

定义条件

\mathbf{R}^n 上的向量范数是一个函数 $\|\cdot\| : \mathbf{R}^n \rightarrow \mathbf{R}$, 满足下列条件：

1. $\|\mathbf{x}\| \geq 0$, 且 $\|\mathbf{x}\| = 0$ 当且仅当 $\mathbf{x} = \mathbf{0}$; ($\mathbf{x} \in \mathbf{R}^n$)
2. $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$, 其中 $\alpha \in \mathbf{R}$, $\mathbf{x} \in \mathbf{R}^n$;
3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$. ($\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$)

1. p -范数: $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$, 其中 $p \geq 1$;

2. 无穷范数: $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$;

收敛性 (感觉不重要)：简单来说 无穷范数收敛 每一项都收敛

\mathbf{R}^n 上的向量序列 $\{\mathbf{x}^{(k)}\}_{k=1}^\infty$ 按照向量范数 $\|\cdot\|$ 收敛到向量 \mathbf{x} , 当且仅当对于任意的 $\epsilon > 0$, 存在整数 $N(\epsilon)$, 使得当 $k > N(\epsilon)$ 时, 有 $\|\mathbf{x}^{(k)} - \mathbf{x}\| < \epsilon$ 。

对于无穷范数, 如果向量序列 $\{\mathbf{x}^{(k)}\}_{k=1}^\infty$ 按照无穷范数 $\|\cdot\|_\infty$ 收敛到向量 \mathbf{x} , 当且仅当对于任意 $i = 1, 2, \dots, n$, 有 $\lim_{k \rightarrow \infty} x_i^{(k)} = x_i$ 。

等价性：类似于夹逼法则，所有范数都是等价的

等价性定义： \mathbf{R}^n 上的向量范数 $\|\cdot\|$ 和 $\|\cdot\|'$ 等价, 当且仅当存在正常数 c_1, c_2 , 使得对于任意的 $\mathbf{x} \in \mathbf{R}^n$, 有 $c_1\|\mathbf{x}\| \leq \|\mathbf{x}\|' \leq c_2\|\mathbf{x}\|$ 。

实际上, \mathbf{R}^n 上的所有范数都是等价的。也就是说, 如果 $\|\cdot\|$ 和 $\|\cdot\|'$ 是 \mathbf{R}^n 上的任意两个范数, 并且 $\{\mathbf{x}^{(k)}\}_{k=1}^\infty$ 按照 $\|\cdot\|$ 收敛到 \mathbf{x} , 那么 $\{\mathbf{x}^{(k)}\}_{k=1}^\infty$ 也按照 $\|\cdot\|'$ 收敛到 \mathbf{x} 。

Matrix Norms

$\mathbf{R}^{n \times n}$ 上的矩阵范数是一个函数 $\|\cdot\| : \mathbf{R}^{n \times n} \rightarrow \mathbf{R}$, 满足下列条件：

1. $\|\mathbf{A}\| \geq 0$, 且 $\|\mathbf{A}\| = 0$ 当且仅当 \mathbf{A} 是零矩阵; ($\mathbf{A} \in \mathbf{R}^{n \times n}$)
2. $\|\alpha\mathbf{A}\| = |\alpha|\|\mathbf{A}\|$, 其中 $\alpha \in \mathbf{R}$, $\mathbf{A} \in \mathbf{R}^{n \times n}$;
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$. ($\mathbf{A}, \mathbf{B} \in \mathbf{R}^{n \times n}$)
4. $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$. ($\mathbf{A}, \mathbf{B} \in \mathbf{R}^{n \times n}$)

Frobenius范数

$\mathbf{A} \in \mathbf{R}^{n \times n}$ 的 Frobenius 范数是 \mathbf{A} 的所有元素的平方和的平方根，即 $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2}$ 。

★ 定义： $\|A\| = \max_{\|x\|=1} \|Ax\| \leftrightarrow \|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$

推论： $\|A\| \|x\| \geq \|Ax\|$

1. p -范数： $\|\mathbf{A}\|_p = \max_{\mathbf{x} \neq 0} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p}$ ，其中 $p \geq 1$ ；

2. 无穷范数： $\|\mathbf{A}\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$ ；也就是 \mathbf{A} 的所有行和的最大值；

3. 1-范数： $\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$ ；也就是 \mathbf{A} 的所有列和的最大值；

4. 2-范数(spectral norm)： $\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^T \mathbf{A})}$ ，其中 $\lambda_{\max}(\mathbf{A}^T \mathbf{A})$ 是 $\mathbf{A}^T \mathbf{A}$ 的最大特征值。

二范数之后会提及

Eigenvalues and Eigenvectors 特征值和特征向量

特征多项式： $p(\lambda) = \det(A - \lambda I)$

如果 $p(\lambda) = 0 \rightarrow (A - \lambda I)\vec{x} = 0$ 有非零解

个人理解考试不考：

矩阵 A 视为是一个映射 如果是 $n \times n$ 的矩阵，那么 $\vec{y} = A\vec{x}$ 就是把 \vec{x} 映射到 \vec{y} 上，此时注意只是一个伸缩变化

特征值是 A 对于特征向量的一个伸缩变化，当我对于所有特征向量进行线性组合时，这样任意一个向量都能够视为伸缩变化。

$$\begin{aligned} \vec{t} &= (k_1 \vec{v}_1 + k_2 \vec{v}_2) \\ A\vec{t} &= (\lambda_1 k_1 \vec{v}_1 + \lambda_2 k_2 \vec{v}_2) \\ \text{伸缩比例} &= \frac{(\lambda_1, \lambda_2)}{(1, 1)} = \sqrt{\frac{\lambda_1^2 + \lambda_2^2}{2}} \leq \max \lambda = \rho(A) \end{aligned}$$

Spectral Radius 谱半径

$\mathbf{A} \in \mathbf{R}^{n \times n}$ 的谱半径定义为 $\rho(\mathbf{A}) = \max_{1 \leq i \leq n} |\lambda_i|$ ，其中 λ_i 是 \mathbf{A} 的特征值，这里的特征值可以是复数。

★ $\rho(A) \leq \|A\|$

矩阵收敛性

$$\lim_{k \rightarrow \infty} (A^k)_{ij} = 0$$

判定收敛：主要是通过谱半径来证明 求最大特征值

1. 矩阵 $\mathbf{A} \in \mathbf{R}^{n \times n}$ 是收敛的;
2. $\rho(\mathbf{A}) < 1$;
3. 对于某些自然范数 $\|\cdot\|$, 有 $\lim_{k \rightarrow \infty} \|\mathbf{A}^k\| = 0$ 。
4. 对于任意的自然范数 $\|\cdot\|$, 有 $\lim_{k \rightarrow \infty} \|\mathbf{A}^k\| = 0$ 。
5. 对于每一个 $\mathbf{x} \in \mathbf{R}^n$, 有 $\lim_{k \rightarrow \infty} \mathbf{A}^k \mathbf{x} = \mathbf{0}$ 。

★ Iterative Techniques for Solving Linear Systems 迭代法求解线性方程组

$$A\vec{x} = \vec{b} \leftrightarrow \vec{x} = T\vec{x} + c$$

$$A = D - L - U$$

Jacobi 迭代法：直接分别表示出每一个元素的值即可

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \quad \dots \quad \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \leftrightarrow \begin{cases} x_1 = \frac{1}{a_{11}}(-a_{12}x_2 - \dots - a_{1n}x_n + b_1) \\ x_2 = \frac{1}{a_{22}}(-a_{21}x_1 - \dots - a_{2n}x_n + b_2) \\ \dots \quad \dots \quad \dots \\ x_n = \frac{1}{a_{nn}}(-a_{n1}x_1 - \dots - a_{n(n-1)}x_{n-1} + b_n) \end{cases}$$

$$\begin{aligned} D\vec{x} &= (L + U)\vec{x} + b \\ \vec{x} &= D^{-1}(L + U)x + D^{-1}b \end{aligned}$$

Gauss-Seidel 迭代法 计算下一个值时 需要带入已经算出的值

$$\begin{aligned} x_2^{(k)} &= \frac{1}{a_{22}}(-a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)} - a_{24}x_4^{(k-1)} - \dots - a_{2n}x_n^{(k-1)} + b_2) \\ x_3^{(k)} &= \frac{1}{a_{33}}(-a_{31}x_1^{(k)} - a_{32}x_2^{(k)} - a_{34}x_4^{(k-1)} - \dots - a_{3n}x_n^{(k-1)} + b_3) \end{aligned}$$

$$x_n^{(k)} = \frac{1}{a_{nn}}(-a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - a_{n3}x_3^{(k)} - \dots - a_{n(n-1)}x_{n-1}^{(k)} + b_n)$$

计算方式：

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

$$(D - L)\vec{x} = U\vec{x} + b$$

$$\vec{x} = (D - L)^{-1}U\vec{x} + (D - L)^{-1}b$$

收敛性 类似于第二章

★ 收敛性判定: 一阶导数<1

对于任意一个 $\mathbf{x}^{(0)} \in \mathbf{R}^n$, 由

$$\mathbf{x}^{(k+1)} = \mathbf{T}\mathbf{x}^{(k)} + \mathbf{c}$$

定义的序列 $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ 收敛到 $\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{c}$ 的唯一解, 当且仅当 $\rho(\mathbf{T}) < 1$ 。

误差界:

如果对任意自然矩阵范数 $\|\mathbf{T}\| < 1$, \mathbf{c} 是给定的向量, 那么由 $\mathbf{x}^{(k+1)} = \mathbf{T}\mathbf{x}^{(k)} + \mathbf{c}$ 定义的序列 $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ 收敛到 $\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{c}$ 的唯一解, 且有误差界:

1. $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|\mathbf{T}\|^k \|\mathbf{x}^{(0)} - \mathbf{x}\|$;
2. $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|\mathbf{T}\|^k}{1-\|\mathbf{T}\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|$;

判断是否收敛 判断谱半径是否小于1 判断最大特征值是否小于1

严格对角占优举证能够使用 Jacobi 和 Gauss-Seidel 都是收敛的

Relaxation Methods 松弛法 SOR

在高斯希尔德基础之上 再添加一项

$$\begin{aligned} x_i^{(k)} &= \frac{-\sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} + b_i}{a_{ii}} \\ &= x_i^{(k-1)} + \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i}^n a_{ij}x_j^{(k-1)}) \\ &= x_i^{(k-1)} + \frac{r_i^{(k)}}{a_{ii}} \end{aligned}$$

添加一个参数来估计步长:

$$x_i^{(k)} = x_i^{(k-1)} + \omega \frac{r_i^{(k)}}{a_{ii}}$$

- $\omega < 1$: 增强收敛性

- $\omega = 1$ 高斯希尔德
- $\omega > 1$ 加快收敛速度

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

$$T_\omega = (D - \omega L)^{-1}[(1 - \omega)D + \omega U] \quad c_\omega = \omega(D - \omega L)^{-1}b$$

Kahan定理

如果 $a_{ii} \neq 0 (i = 1, 2, \dots, n)$, 那么 $\rho(T_\omega) \geq |\omega - 1|$ 。这表明, SOR方法当且仅当 $\omega \in (0, 2)$ 时收敛。

Ostrowski-Reich定理

如果 \mathbf{A} 是一个正定矩阵, 并且 $\omega \in (0, 2)$, 那么 SOR 方法对于任意的初始近似向量 $\mathbf{x}^{(0)} \in \mathbf{R}^n$ 都收敛。

ω 的最佳选择

如果 \mathbf{A} 是一个正定的三对角矩阵, 那么 $\rho(T_g) = [\rho(\mathbf{T}_j)]^2 < 1$, 并且 SOR 方法的最佳 ω 选择是:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - [\rho(\mathbf{T}_j)]^2}}$$

由此选择的 ω , 有 $\rho(T_\omega) = \omega - 1$ 。

Answer: First prove $\det(T_w) = (1 - w)^n = \prod_{i=1}^n \lambda_i$:

$$\begin{aligned}\det(T_w) &= \det(D - wL)^{-1} \det((1 - w)D + wU) = \det(D^{-1}) \det((1 - w)D + wU) \\ &= \det((1 - w)I + wD^{-1}U) = (1 - w)^n \\ \text{because } D^{-1}U \text{ has all 0 diagonal elements.}\end{aligned}$$

Second: Since $\rho(T_w)$ is the eigenvalue of largest magnitude, $\rho(T_w)^n \gg |(1 - w)^n| = |(1 - w)|^n$. Therefore, we get: $\rho(T_w) > |1 - w|$

Error Bounds and Iterative Refinement 条件数 condition number

condition number: $K(A) = ||A|| ||A^{-1}||$

求逆矩阵时 提高精度计算

$$A^{-1} = \frac{A^*}{|A|}$$

$K(A)$ 越大, 误差越大, $K(A)$ 接近于 1, A 误差小

1. $K(A) \geq 1$
2. A 对称 $K(A)_2 = \frac{|\lambda_{max}|}{|\lambda_{min}|}$
3. $K(Aa) = K(A)$
4. $K(A)_2 = 1$ 当且仅当 A 正交 (orthogonal) ($AA^t = I$)

5. $K(RA)_2 = K(AR)_2 = K(A)_2$ for all orthogonal matrix R.

迭代改进

Theorem: Suppose that \bar{x}^* is an approximation to the solution of $A\bar{x} = \bar{b}$, A is a nonsingular matrix, and $\bar{r} = \bar{b} - A\bar{x}$ is the residual vector of \bar{x}^* . Then for any natural norm, $\|\bar{x} - \bar{x}^*\| \leq \|\bar{r}\| \cdot \|A^{-1}\|$. And if $\bar{x} \neq \bar{0}$ and $\bar{b} \neq \bar{0}$,

$$\frac{\|\bar{x} - \bar{x}^*\|}{\|\bar{x}\|} \leq K(A) \frac{\|\bar{r}\|}{\|\bar{b}\|}.$$

迭代过程

1. $Ax = b \rightarrow x_1$
2. $r_1 = b - Ax_1$
3. $Ad_1 = r_1 \rightarrow d_1$
4. $x_2 = X_1 + d_1$
5. 回到2

Chapter 9 Approximating Eigenvalues

Power Method

Original Method

$$\begin{aligned}
 \text{Eigenvalue : } & |\lambda_1| \geq |\lambda_2| \dots \geq |\lambda_n| \\
 \text{Eigenvector : } & \{\vec{v}^{(1)}, \vec{v}^{(2)} \dots\} \\
 \vec{x}^{(0)} &= \sum_{j=1}^n \beta_j \vec{v}^{(j)} \\
 \vec{x}^{(1)} &= A\vec{x} = \sum_{j=1}^n \beta_j A\vec{v}^{(j)} = \sum_{j=1}^n \beta_j \lambda_j \vec{v}^{(j)} \\
 &\dots \\
 \vec{x}^{(n)} &= A^n \vec{x} = \sum_{j=1}^n \beta_j A\vec{v}^{(j)} = \sum_{j=1}^n \beta_j \lambda_j^n \vec{v}^{(j)} \\
 &= \lambda_1 \sum_{j=1}^n \beta_j \left(\frac{\lambda_j}{\lambda_1}\right) \vec{v}^{(j)} \\
 \lim_{j \rightarrow \infty} A^j \vec{x} &= \lim_{j \rightarrow \infty} \lambda_1^k \beta_1 \vec{v}^{(1)} \\
 \lambda_1 &\approx \frac{(\vec{x}^{(k)})_i}{(\vec{x}^{(k-1)})_i}
 \end{aligned}$$

迭代 $x^{(k)} = Ax^{(k-1)}$ 逼近 λ_1

但是，迭代过程中很可能出现发散的情况，于是需要适当方法调整x来保证不会发散。

Normalization 归一化

为了增强收敛性 只需要保证每一步对 $\vec{x}^{(k)}$ 归一化，使得 $\|\vec{x}^{(k)}\|_\infty = 1$

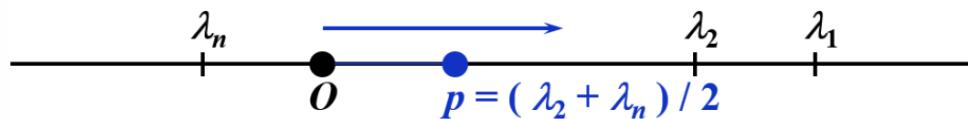
$$\begin{aligned}\|\vec{x}^{(k)}\|_\infty &= |\vec{x}_{p_k}^{(k)}| \\ \text{Normalization : } \vec{u}^{(k-1)} &= \frac{\vec{x}^{(k-1)}}{|\vec{x}_{p_{k-1}}^{(k-1)}|} \\ \vec{x}^{(k)} &= A\vec{u}^{(k-1)} \\ \text{result : } \vec{u}^{(k)} &\rightarrow \vec{v}_1 \\ \lambda_1 &= \frac{\vec{x}^{(k)}}{\vec{u}^{(k-1)}} = \vec{x}_{p_{k-1}}^{(k)}\end{aligned}$$

Note:

1. 主特征值的重数大于1 仍然有效
2. 如果存在特征值 $\lambda_1 = -\lambda_2$ 幂法失效
3. 因为我们无法确保对于任意的初始向量 $x^{(0)}$ 都有 $\beta_1 \neq 0$ ，所以迭代的结果可能不是 v_1 ，而是满足 $(x^{(0)}, v_m) \neq 0$ 的第一个向量 v_m ，相应地，得到的特征值为 λ_m 。
4. Aitken's Δ2 可以加速收敛，幂法线性收敛

收敛速度 Rate of Convergence

因为 $\mathbf{x}^{(k)} = \lambda_1^k \sum_{j=1}^n \beta_j (\frac{\lambda_j}{\lambda_1})^k \mathbf{v}_j$ ，假设 $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_n$ ，且 $|\lambda_2| \geq |\lambda_n|$ ，则我们的目标就是让 $\frac{\lambda_2}{\lambda_1}$ 尽可能小，这样收敛速度更快。



记 $\mathbf{B} = \mathbf{A} - p\mathbf{I}$ ，其中 $p = \frac{\lambda_2 + \lambda_n}{2}$ ，则 \mathbf{B} 的特征值为 $\lambda_1 - p, \lambda_2 - p, \dots, \lambda_n - p$ ，因为 $|\frac{\lambda_2 - p}{\lambda_1 - p}| < |\frac{\lambda_n}{\lambda_1}|$ ，所以此时 \mathbf{B} 的收敛速度更快。

但是我们并不知道 λ_2 和 λ_n ，所以这不一定是一个好的选择。

Inverse power Method

如果 \mathbf{A} 有特征值 $|\lambda_1| \geq |\lambda_2| \dots \geq |\lambda_n|$ 那么 \mathbf{A}^{-1} 也有特征值：
 $|\frac{1}{\lambda_n}| \geq |\frac{1}{\lambda_{n-1}}| \dots$

于是可以求特征值。

利用这个原理，如果已知某一个特征值 λ_k 趋近于 p ，那么构造矩阵 $(\mathbf{A} - p\mathbf{I})^{-1}$ 此时最大特征值就是 $\frac{1}{\lambda_k - p}$

Chapter 3 Interpolation and Polynomial Approximation

Interpolation and Lagrange Polynomials 拉格朗日插值

n次多项式 n+1个点插值

$$P(x) = a_0 + a_1x + \dots + a_nx^n, P(x_i) = y_i, \text{找到 } L_{n,i}(x) = \delta_{ij}$$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

$$L_{n,i}(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)}$$

$$P_n(x) = \sum_{i=0}^n L_{n,i}(x)y_i$$

拉格朗日插值多项式唯一

| 用n个点插值超过n次的多项式时，结果不唯一

余项分析 Remainder:

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

⚠ 注意：

1. 内插(Intrapolation) 会比 外插(Extrapolation) 更加准确
2. 高次的拉格朗日插值一般会比低次的插值更加准确，但是这不一定总成立。

Neville 迭代插值

$$P(x) = \frac{(x - x_i)P_{0,1,2,\dots,i-1,i+1,\dots,n}(x) - (x - x_j)P_{0,1,2,\dots,j-1,j+1,\dots,n}(x)}{(x_i - x_j)}$$

x_0	$P_0 = Q_{0,0}$				
x_1	$P_1 = Q_{1,0}$	$P_{0,1} = Q_{1,1}$			
x_2	$P_2 = Q_{2,0}$	$P_{1,2} = Q_{2,1}$	$P_{0,1,2} = Q_{2,2}$		
x_3	$P_3 = Q_{3,0}$	$P_{2,3} = Q_{3,1}$	$P_{1,2,3} = Q_{3,2}$	$P_{0,1,2,3} = Q_{3,3}$	
x_4	$P_4 = Q_{4,0}$	$P_{3,4} = Q_{4,1}$	$P_{2,3,4} = Q_{4,2}$	$P_{1,2,3,4} = Q_{4,3}$	$P_{0,1,2,3,4} = Q_{4,4}$

$$Q_{ij} = P_{i-j, i-j-1, \dots, i}$$

$$Q_{i,j}(x) = \frac{(x - x_{i-j})Q_{i,j-1}(x) - (x - x_i)Q_{i-1,j-1}(x)}{(x_i - x_j)}$$

Divided Differences 差商

只是一个表达符号

$$0 : f[x_i] = f(x_i)$$

$$1 : f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

$$\dots$$

$$n : f[x_0, x_1 \dots x_n] = \frac{f[x_1, x_2 \dots x_n] - f[x_0, x_1 \dots x_{n-1}]}{x_n - x_0}$$

Table 3.7

x	$f(x)$	First divided differences	Second divided differences	Third divided differences
x_0	$f[x_0]$			
		$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$	$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$	
x_1	$f[x_1]$			$f[x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}$
		$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1}$	$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$	
x_2	$f[x_2]$			$f[x_1, x_2, x_3, x_4] = \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1}$
		$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2}$	$f[x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2}$	
x_3	$f[x_3]$			$f[x_2, x_3, x_4, x_5] = \frac{f[x_3, x_4, x_5] - f[x_2, x_3, x_4]}{x_5 - x_2}$
		$f[x_3, x_4] = \frac{f[x_4] - f[x_3]}{x_4 - x_3}$		
x_4	$f[x_4]$		$f[x_3, x_4, x_5] = \frac{f[x_4, x_5] - f[x_3, x_4]}{x_5 - x_3}$	
		$f[x_4, x_5] = \frac{f[x_5] - f[x_4]}{x_5 - x_4}$		
x_5	$f[x_5]$			

差商型 Newton 插值多项式 Newton's Interpolatory Divided-Difference formula

联想展开

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, x_1 \dots x_n](x - x_0)(x - x_1) \dots (x - x_{n-1})$$

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

n阶差分的性质：

$$f[x_0, x_1, x_2 \dots x_k] = \frac{f^k(\xi)}{k!}$$

引入向前差分(forward difference)记号:

$$\begin{aligned}\Delta f(x_i) &= f(x_{i+1}) - f(x_i) \\ \Delta^2 f(x_i) &= \Delta f(x_{i+1}) - \Delta f(x_i) \\ \Delta^3 f(x_i) &= \Delta^2 f(x_{i+1}) - \Delta^2 f(x_i) \\ &\dots\end{aligned}$$

引入向后差分(backward difference)记号:

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

引入中心差分(central difference)记号:

$$\delta^k f_i = \delta^{k-1} f_{i+\frac{1}{2}} - \delta^{k-1} f_{i-\frac{1}{2}}$$

其中

$$f_{i \pm \frac{1}{2}} = f(x_i \pm \frac{h}{2})$$

等距前向牛顿插值 推导: P112

$$\begin{aligned}P_n(x) &= \sum_{k=0}^n C_s^k \Delta^k f(x_0) \\ x_n &= x_0 + sh\end{aligned}$$

等距后向牛顿插值

$$P_n(x_s) = \sum_{k=0}^n (-1)^k \binom{-s}{k} \nabla^k f(x_n)$$

Hermite Interpolation | Hermite 插值

密切多项式 osculation polynomial

1. $P(x_i) = f(x_i), f'(x_i) = P'(x_i) \dots p^{m_i}(x_i) = f^{m_i}(x_i)$

2. 有 $\sum_{i=0}^n (m_i) + n + 1$ 个条件 于是得到 $\sum_{i=0}^n m_i + n$ 阶多项式

$n=0$ 泰勒多项式

$m_i = 0$ n 次拉格朗日插值

Hermite 插值多项式 对于 $m=1$ 的情况

- 得到多项式次数不超过 $2n + 1$

三阶的特殊情况:

$$P(x) = \sum_{i=0}^2 f(x_i) h_{3,i}(x) + f'(x_1) \hat{h}_1(x)$$

$$h_0(x) = \frac{(x - x_1)^2(x - x_2)}{(x_0 - x_1)^2(x_0 - x_2)}$$

$h_2(x)$ same

$$h_1(x) = (Ax + B)(x - x_0)(x - x_2), h_1(x_1) = 1, h_1'(x_1) = 0$$

$$\hat{h}_1(x) = C(x - x_0)(x - x_1)(x - x_2), \hat{h}_1'(x_1) = 1$$

一般情况：

导数存在 加平方 加个待定系数

$$h_i(x_j) = 1, \text{ if } i = j$$

$$h_i(x_j) = 0, \text{ if } i \neq j$$

$$\hat{h}_i(x_j) = 1, \text{ if } i = j$$

$$\hat{h}_i(x_j) = 0, \text{ if } i \neq j$$

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

$$h_i(x) = (Ax + B)L_{n,j}^2(x)$$

$$h_i(x) = (1 - 2(x - x_i)L_{n,i}'(x))(L_{n,i}(x))^2$$

$$\hat{h}_i(x) = (x - x_i)(L_{n,i}(x))^2$$

余项：由于是一个 $2n+1$ 阶多项式

$$R_{2n+1}(x) = \frac{f^{2n+2}(\xi)}{(2n+2)!} \prod_{i=0}^{2n+2} (x - x_i)^2$$

Piecewise polynomial approximation 分段多项式逼近

简单的分段多项式逼近是分段线性逼近，即在每个子区间上用一个一次多项式逼近函数 f 。但是，分段线性逼近的函数不光滑，所以我们希望用更高次的多项式来逼近 f 。

Cubic Spline Interpolation 三样条插值

1. $S(x)$ 在每个子区间 $[x_i, x_{i+1}]$ 上是一个三次多项式, $i = 0, 1, \dots, n - 1$
2. $S(x_i) = f(x_i)$, $i = 0, 1, \dots, n$
3. $S_{i+1}(x_{i+1}) = S_i(x_{i+1})$, $i = 0, 1, \dots, n - 2$
4. $S'_{i+1}(x_{i+1}) = S'_i(x_{i+1})$, $i = 0, 1, \dots, n - 2$
5. $S''_{i+1}(x_{i+1}) = S''_i(x_{i+1})$, $i = 0, 1, \dots, n - 2$
6. 下列的边界条件之一成立:
 - a. $S''(x_0) = S''(x_n) = 0$, 称为**自由或自然边界(free or natural boundary)**
 - b. $S'(x_0) = f'(x_0)$, $S'(x_n) = f'(x_n)$, 称为**固支边界(clamped boundary)**
 - c. 其他边界条件 (上面两个条件其实已经足以满足目的了)

推导过于繁琐 跳过

$$\begin{aligned}
 h_j &= x_j - x_{j-1} \\
 \lambda_j &= \frac{h_{j+1}}{j_j + h_{j+1}} \\
 u_j &= \frac{h_j}{h_j + h_{j+1}} \\
 g_j &= \frac{6(f[x_k, x_{j+1}] - f[x_{j-1}, x_j])}{h_j + h_{j+1}} \\
 j &= 1 \dots n - 1 \\
 u_j M_{j-1} + 2M_j + \lambda_j M_{j+1} &= g_j
 \end{aligned}$$

$$\begin{bmatrix}
 \mu_1 & 2 & \lambda_1 & & & \\
 & \mu_2 & 2 & \lambda_2 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & \mu_{n-1} & 2 & \lambda_{n-1}
 \end{bmatrix}
 \begin{bmatrix}
 M_0 \\
 M_1 \\
 \vdots \\
 M_n
 \end{bmatrix}
 =
 \begin{bmatrix}
 g_1 \\
 g_2 \\
 \vdots \\
 g_{n-1}
 \end{bmatrix}$$

我们有 $n + 1$ 个未知数, $n - 1$ 个方程 \rightarrow 由边界条件增加两个方程

CLAMPED BOUNDARY | 固支边界

此时我们知道 $S'(x_0) = f'(x_0)$, $S'(x_n) = f'(x_n)$, 所以

$$\text{在 } [x_0, x_1] \text{ 上, } S'_1(x) = -M_0 \frac{(x_1-x)^2}{2h_1} + M_1 \frac{(x-x_0)^2}{2h_1} + f[x_0, x_1] - \frac{M_1-M_0}{6} h_1$$

$$\text{在 } [x_{n-1}, x_n] \text{ 上, } S'_n(x) = -M_{n-1} \frac{(x_n-x)^2}{2h_n} + M_n \frac{(x-x_{n-1})^2}{2h_n} + f[x_{n-1}, x_n] - \frac{M_n-M_{n-1}}{6} h_n$$

所以我们额外有两个方程:

$$\begin{cases} f'(x_0) = -M_0 \frac{h_1}{2} + f[x_0, x_1] - \frac{M_1-M_0}{6} h_1 \\ f'(x_n) = M_n \frac{h_n}{2} + f[x_{n-1}, x_n] - \frac{M_n-M_{n-1}}{6} h_n \end{cases} \Rightarrow \begin{cases} 2M_0 + M_1 = \frac{6}{h_1} (f[x_0, x_1] - f'(x_0)) \triangleq g_0 \\ M_{n-1} + 2M_n = \frac{6}{h_n} (f'(x_n) - f[x_{n-1}, x_n]) \triangleq g_n \end{cases}$$

所以我们可以得到

$$\begin{bmatrix} 2 & 1 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & 1 & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_n \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_n \end{bmatrix}$$

NATURAL BOUNDARY | 自由边界

根据之前的假设 $M_0 = S''(x_0) = y_0''$, $M_n = S''(x_n) = y_n''$, 则

$$\lambda_0 = 0, g_0 = 2y_0'', \mu_n = 0, g_n = 2y_n''$$

当 $S''(x_0) = S''(x_n) = 0$, 我们称之为**自由边界{free boundary}**, 此时 $g_0 = g_n = 0$ 。

$$\begin{bmatrix} 2 & 0 & 0 & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & 0 & 0 & 2 \end{bmatrix}_{(n+1) \times (n+1)} \begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix}_{(n+1) \times 1} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \\ g_n \end{bmatrix}_{(n+1) \times 1}$$

自由边界的情况下, 有 $S''(x_0) = S''(x_n) = 0$ 。

在解出M之后可以带入一下式子分别求出不同区间的值:

$$\begin{cases} S_j(x_{j-1}) = y_{j-1} \\ S_j(x_j) = y_j \end{cases} \Rightarrow \begin{cases} M_{j-1} \frac{h_j^2}{6} + A_j x_{j-1} + B_j = y_{j-1} \\ M_j \frac{h_j^2}{6} + A_j x_j + B_j = y_j \end{cases} \Rightarrow \begin{cases} A_j = \frac{y_j - y_{j-1}}{h_j} - \frac{M_j - M_{j-1}}{6} h_j \\ B_j = \frac{y_{j-1} x_j - y_j x_{j-1}}{h_j} - \frac{M_{j-1} x_j - M_j x_{j-1}}{6} h_j \end{cases}$$

所以

$$\begin{aligned} A_j x + B_j &= \frac{y_j - y_{j-1}}{h_j} x + \frac{y_{j-1} x_j - y_j x_{j-1}}{h_j} - \frac{M_j - M_{j-1}}{6} h_j x - \frac{M_{j-1} x_j - M_j x_{j-1}}{6} h_j \\ &= (y_{j-1} - \frac{M_{j-1}}{6} h_j^2) \frac{x_j - x}{h_j} + (y_j - \frac{M_j}{6} h_j^2) \frac{x - x_{j-1}}{h_j} \end{aligned}$$

$$S_j(x) = M_{j-1} \frac{(x_j - x)^3}{6h_j} + M_j \frac{(x - x_{j-1})^3}{6h_j} + A_j x + B_j$$

Properties of cubic splines | 三次样条的性质

- 只要系数矩阵严格对角占优（实际上是确保可逆），三次样条就可以由其边界条件唯一确定。
- 如果 $\frac{\max h_i}{\min h_i}$ 有界，那么 $S(x)$ 是收敛的。
- 增加点可以更贴近原函数。

Chapter 8 Approximation Theory

尽可能逼近 而不是全部穿过

Discrete Least Squares Approximation 离散最小二乘逼近

$$E(p) = \sum_{i=1}^n (y_i - P(x_i))^2$$

用多项式拟合 $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$

$$\begin{bmatrix} \sum_{i=1}^m x_i^0 & \sum_{i=1}^m x_i^1 & \cdots & \sum_{i=1}^m x_i^n \\ \sum_{i=1}^m x_i^1 & \sum_{i=1}^m x_i^2 & \cdots & \sum_{i=1}^m x_i^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^m x_i^n & \sum_{i=1}^m x_i^{n+1} & \cdots & \sum_{i=1}^m x_i^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m y_i x_i^0 \\ \sum_{i=1}^m y_i x_i^1 \\ \vdots \\ \sum_{i=1}^m y_i x_i^n \end{bmatrix}$$

Orthogonal Polynomials and Least Squares Approximation 正交多项式与最小二乘逼近

$$E = \int_a^b (f(x) - p(x))^2 dx$$

广义多项式(Generalized Polynomial): 用线性无关的函数 $\phi_0(x), \phi_1(x), \dots, \phi_n(x)$ 的线性组合 $P(x) = \sum_{i=0}^n a_i \phi_i(x)$ 来逼近 $f(x)$ ，这里的 $P(x)$ 称为广义多项式。

- Trigonometric polynomial: $\phi_i(x) = \cos(ix)$ or $\sin(ix)$
- Exponential polynomial: $\phi_i(x) = e^{k_i x}, k_i \neq k_j$
- 记 $\Pi_n(x)$ 为阶数最多为 n 的多项式的集合， $\Pi_n(x)$ 是一个线性空间， $\Pi_n(x)$ 的基为 $\{1, x, x^2, \dots, x^n\}$ ，可以拿来做广义多项式的基。

Weight Function | 权函数

不同点分配一个重要程度

$$E(p) = \sum_{i=1}^n w_i (y_i - P(x_i))^2$$

$$E = \int_a^b w(x) (f(x) - p(x))^2 dx$$

我们定义内积为

$$\langle f, g \rangle = \begin{cases} \sum_{i=1}^m w_i f(x_i)g(x_i) & \text{离散} \\ \int_a^b w(x) f(x)g(x) dx & \text{连续} \end{cases}$$

如果 $\langle f, g \rangle = 0$, 则称 f 和 g 正交。

我们定义范数为

$$\|f\| = \sqrt{\langle f, f \rangle}$$

所以我们可以把误差表达式写成

$$E = \langle f - p, f - p \rangle = \|f - p\|^2$$

★ 最小二乘最终形式 只需要记录这个就行

- 离散型求和 连续形求积分
- 注意权重函数只需要乘1遍

$$\begin{bmatrix} \langle \phi_0, \phi_0 \rangle & \langle \phi_0, \phi_1 \rangle & \cdots & \langle \phi_0, \phi_n \rangle \\ \langle \phi_1, \phi_0 \rangle & \langle \phi_1, \phi_1 \rangle & \cdots & \langle \phi_1, \phi_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \phi_n, \phi_0 \rangle & \langle \phi_n, \phi_1 \rangle & \cdots & \langle \phi_n, \phi_n \rangle \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \langle \phi_0, f \rangle \\ \langle \phi_1, f \rangle \\ \vdots \\ \langle \phi_n, f \rangle \end{bmatrix}$$

为了方便矩阵求解 以及降低希尔伯特矩阵带来的误差 于是进行正交化 构造正交多项式

只剩下一个对角矩阵了：

$$a_k = \frac{\langle \phi_k, f \rangle}{\langle \phi_k, \phi_k \rangle}$$

正交化方法(不要忘记权函数)：

$$\begin{aligned}\phi_0(x) &= 1 \\ \phi_1(x) &= x - B_1 \\ \phi_k = (x - B_k)\phi_{k-1} - C_k\phi_{k-2}(x) & k = 2, 3, 4.. \\ B_k &= \frac{\langle x\phi_{k-1}, \phi_{k-1} \rangle}{\langle \phi_{k-1}, \phi_{k-1} \rangle} \\ C_k &= \frac{\langle x\phi_{k-1}, \phi_{k-2} \rangle}{\langle \phi_{k-2}, \phi_{k-2} \rangle}\end{aligned}$$

Chebyshev Polynomials and Economization of Power Series 切比雪夫多项式与幂级数的缩减

之前解决的是二阶范数的最小值问题，现在尝试去解决找到最小 $\|p - y\|_\infty$ 于是就是找到一个最大值最小问题，

Deviation Point偏差点：满足 $P(x_0) - f(x_0) = \pm \|P_n - f\|_\infty$

性质

- 如果 f 不是一个 n 阶多项式，那么一定存在唯一 P_n
- 如果 $P_n(x)$ 存在，那么必须有正负偏差点
- **切比雪夫定理：** $P_n(x)$ 使得 $\|P_n - f\|_\infty$ 最小 \leftrightarrow 至少有 $n+2$ 个正负偏差点 存 在 $a \leq t_1 \leq t_2 \leq \dots t_{n+2} \leq b$ 使 得 $P_n(t_i) - f(x_i) = \pm(-1)^k \|p_n - f\|_\infty \quad \{tk\}$ 被称为 **切比雪夫交替序列** (*Chebyshev alternating sequence*)

于是我需要找到这样的唯一的一个多项式使得无穷范数最小，由于我已经掌握了性质，不妨根据性质去找偏差点，更具这些点来插值一个多项式，满足条件，于是任务就变成了找到 $n+1$ 个点去插值多项式。最小化一下的式子

$$|P_n(x) - f(x)| = |R_n(x)| = \left| \frac{f^{n+1}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i) \right|$$

以下是特殊化处理

现 在 需 要 找 到 $\{x_1, \dots, x_n\}$ 使 得 $\|\omega_n\|_\infty$ 在 $[-1, 1]$ 上 最 小 化 $\omega_n(x) = \prod_{i=0}^n (x - x_i) = x^n - P_{n-1}(x)$ ，于 是 就 能 找 到 $|P_{n-1}(x) - x^n|$ 的最 小 值。

由切比雪夫定理知道有 $n+1$ 个正负偏差点，如果能构造出这样的多项式 $P_{n-1}(x)$ 那么问题就解决了。

考 虑 使用 三 角 函 数 $\cos(n\theta)$ 在 $[0, \pi]$ 有 $n+1$ 个 最 大 最 小 值 交 替，令 $x = \cos(\theta)$, $x \in [-1, 1]$, 这样就能获得 **切比雪夫多项式**

$$T_n(x) = \cos(n\theta) = \cos(n \arccos(x)), x \in [-1, 1]$$

- 在 $x = \cos(\frac{k\pi}{n})$ 时取到交替极值点 $(-1)^k$
- 在 $x = \cos(\frac{2k-1}{2n}\pi)$ ($k = 1, \dots, n$) 取到根

递归方式求切比雪夫多项式：

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_n(x) &= 2xT_{n-1}(x) - T_{n-2}(x), n = 2, \dots \end{aligned}$$

- 最高次系数为 2^{n-1}
- 在 $[-1, 1]$ 上关于权函数 $\frac{1}{\sqrt{1-x^2}}$ 正交

$$\langle T_n, T_m \rangle = \int_{-1}^1 \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = \begin{cases} \pi & n = m = 0 \\ \frac{\pi}{2} & n = m \neq 0 \\ 0 & n \neq m \end{cases}$$

回到原来的问题这样就能得到 $\omega_n(x) = x^n - P_{n-1}(x) = \frac{T_n(x)}{2^{n-1}}$

所以 $\|\omega_n(x)\|_\infty = \frac{1}{2^{n-1}}$, 而此时 $\{x_1, \dots, x_n\}$ 是 $T_n(x)$ 的根

$$\max_{x \in [-1, 1]} |f(x) - P_{n-1}(x)| \leq \frac{1}{2^{n-1}(n)!} \max |f^{(n)}(x)|$$

找到这些点之后通过这些点插值得到 $P(x)$.

解决一般性问题

1. 如果 $x \in [a, b]$ 那么通过换元 $x = \frac{b-a}{2}t + \frac{a+b}{2}$, 把题目中原函数的 x 代换成 t
2. 用 n 阶多项式拟合的话: $R_n \leq \frac{1}{2^n(n+1)!} \max |f^{(n+1)}(t)|$
3. 计算出 n 的值 不妨 $n=k$ 此时用 k 阶多项式去拟合 $f(t)$, 于是需要 $k+1$ 个点去插值, 故取 $k+1$ 阶切比雪夫多项式 $T_{k+1}(t)$
4. 根据在 $x = \cos(\frac{2k-1}{2n}\pi)$ ($k = 1, \dots, n$) 取到根, 算出 $k+1$ 个根, 作为原函数插值点, 计算出 x
5. 用拉格朗日插值

Economization of Power Series 幂级数的缩减

使用更低的阶数拟合, 但是精度下降不多。

$$\begin{aligned}\max_{x \in [-1,1]} |f(x) - P_{n-1}(x)| &\leq \max_{x \in [-1,1]} |f(x) - P_n(x)| + \max_{x \in [-1,1]} |Q_n(x)| + \max_{x \in [-1,1]} |P_n(x) - P_{n-1}(x) - Q_n(x)| \\ &\leq \max_{x \in [-1,1]} |f(x) - P_n(x)| + \max_{x \in [-1,1]} |Q_n(x)|\end{aligned}$$

为了使得精确度的损失最小, $Q_n(x)$ 必须为

$$a_n \cdot \frac{T_n(x)}{2^{n-1}}$$

具体做法关注书 p460

Chapter 4 Numerical Differentiation and Integration

Numerical Differentiation 数值微分

两点法:

- Forward: $f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$
- Backward: $f'(x) = \frac{f(x) - f(x-h)}{h} + O(h)$

n+1点法: 构造拉格朗日插值多项式

$$\begin{aligned}f(x) &= \sum_{k=0}^n f(x_k) L_k(x) + \frac{(x - x_0) \cdots (x - x_n)}{(n+1)!} f^{(n+1)}(\xi_x) \\ f'(x_j) &= \sum_{k=0}^n f(x_k) L'_k(x_j) + \frac{f^{(n+1)}(\xi_j)}{(n+1)!} \prod_{k=0, k \neq j}^n (x_j - x_k)\end{aligned}$$

点的数量越多, 精确度越高, 到那时求值数目增加, 导致unsatble

数值微分不稳定

三点公式

三个点 x_1, x_2, x_3

$$\begin{aligned}f'(x) &= f(x_0) \left[\frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} \right] + f(x_1) \left[\frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \right] \\ &\quad + f(x_2) \left[\frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} \right] \\ &\quad + \frac{1}{6} f^{(3)}(\xi) (x - x_0)(x - x_1)(x - x_2)\end{aligned}$$

如果是等距的：

$$\begin{aligned} f'(x) &= f(x_0) \left[\frac{2x - 2x_0 - 3h}{2h^2} \right] + f(x_1) \left[\frac{2x - 2x_0 - 2h}{-h^2} \right] \\ &\quad + f(x_2) \left[\frac{2x - 2x_0 - h}{2h^2} \right] \\ &\quad + \frac{1}{6} f^{(3)}(\xi)(x - x_0)(x - x_1)(x - x_2) \end{aligned}$$

三个点带入 **中间点误差最小 三点公式**：

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

高阶导数 这里只介绍二阶导数

分别再 $x_0 + h$ 和 $x_0 - h$ 处泰勒展开，并将两个式子相加

$$f''(x_0) = \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} - \frac{h^2}{24} [f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})]$$

$$f''(x_0) = \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi)$$

Elements of Numerical Integration 数值积分

使用**数值求积 numerical quadrature** 求积分

对目标函数拉格朗日展开：

$$\begin{aligned} \int_a^b f(x) dx &= \int_a^b P_n(x) dx = \sum_{i=0}^n f(x_i) \int_a^b L_i(x) dx \\ error : & \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i) dx \end{aligned}$$

Degree of accuracy / Precision: 能使求积公式对 x^k 保持精准的最大整数 k

Newton-Cotes 求积分

$$x_i = a + ih, h = \frac{b-a}{n}, x = a + th$$

$$\begin{aligned} \int_a^b f(x) dx &= \int_a^b P_n(x) dx = \sum_{i=0}^n f(x_i) a_i \\ a_i &= \int_{x_0}^{x_n} L_i(x) dx = h \frac{(-1)^{n-i}}{i!(n-i)!} \int_0^n \prod_{j \neq i, j=0}^n (t - j) dt \end{aligned}$$

Trapezoidal Rule n=1

$$\int_a^b f(x)dx = \frac{h}{2}[f(a) + f(b)] - \frac{h^3}{12}f''(\xi)$$

- Precision = 1

Simpson Rule n=2

$$\int_a^b f(x)dx = \frac{h}{3}[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)] - \frac{h^5}{90}f^{(4)}(\xi)$$

- Precision=3

Simpson 3/8 n=3

$$\begin{aligned} \int_a^b f(x)dx &= \frac{3h}{8}[f(a) + 3f\left(\frac{2a+b}{3}\right) + 3f\left(\frac{a+2b}{3}\right) + f(b)] \\ &\quad - \frac{3h^5}{80}f^{(4)}(\xi) \end{aligned}$$

- Precision=3

n=4

$$\begin{aligned} \int_a^b f(x)dx &= \frac{2h}{45}[7f(a) + 32f\left(\frac{3a+b}{4}\right) + 12f\left(\frac{a+b}{2}\right) + 32f\left(\frac{a+3b}{4}\right) + 7f(b)] \\ &\quad - \frac{8h^7}{945}f^{(6)}(\xi) \end{aligned}$$

一般结论

- 当n是偶数时 精度为 n+1:

$$\int_a^b f(x)dx = \sum_{k=0}^n A_k f(x_k) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t^2(t-1)\dots(t-n)dt$$

- 当n为奇数时 精度为n

$$\int_a^b f(x)dx = \sum_{k=0}^n A_k f(x_k) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t(t-1)\dots(t-n)dt$$

Composite Numerical Integration 复合数值积分

简单说就是分段求积分

Composite Trapezoidal Rule

把 $[a, b]$ 分成 n 个子区间，每个区间的长度为 $h = \frac{b-a}{n}$

$$\int_{x_{k-1}}^{x_k} f(x) dx \approx \frac{x_k - x_{k-1}}{2} [f(x_{k-1}) + f(x_k)], \quad k = 1, \dots, n$$

$$\int_a^b f(x) dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) dx = \frac{h}{2} [f(a) + 2 \sum_{j=1}^{n-1} f(x_j) + f(b)] = T_n$$

error:

$$\frac{h^2}{12} (b-a) f''(\xi)$$

Composite Simpson Rule: n 为偶数

将区间 $[a, b]$ 分成 n 个子区间，每个子区间长度为 $h = \frac{b-a}{n}$ ，则

$$\int_{x_k}^{x_{k+1}} f(x) dx \approx \frac{h}{6} [f(x_k) + 4f(x_{k+\frac{1}{2}}) + f(x_{k+1})]$$

$$\int_a^b f(x) dx \approx \frac{h}{6} [f(a) + 4 \sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}}) + 2 \sum_{k=0}^{n-2} f(x_{k+1}) + f(b)] = S_n$$

其中， $x_i = a + ih$, $\xi \in [a, b]$ 。

误差项为

$$\int_a^b f(x) dx - S_n = -\frac{b-a}{180} \left(\frac{h}{2}\right)^4 f^{(4)}(\xi)$$

主要使用下面这个：

n 是子区间个数 n' 是点的数量

为简化表达，我们取 $n' = 2n$ ，则 $h' = \frac{b-a}{n'} = \frac{h}{2}$, $x_{2k} = x_k$, $x_{2k+1} = x_k + \frac{h}{2}$ ，则

$$\int_a^b f(x) dx \approx \frac{h'}{3} [f(a) + 4 \sum_{\text{odd } k} f(x_k) + 2 \sum_{\text{even } k} f(x_k) + f(b)] = S_{n'}$$

舍入误差是稳定的，无论分多少个区间都不会增加误差

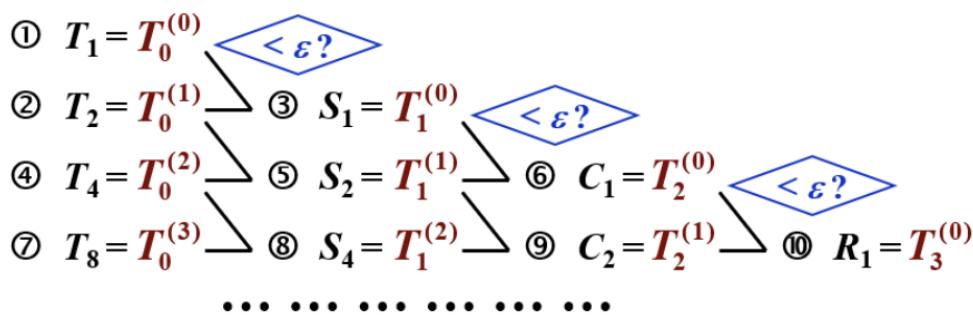
Romberg Integration

R_n 就是 Romberg

$$\frac{4T_{2n} - T_n}{4 - 1} = S_n$$

$$\frac{4^2 S_{2n} - S_n}{4^2 - 1} = C_n$$

$$\frac{4^3 C_{2n} - C_n}{4^3 - 1} = R_n$$



Richardson 's Extrapolation 外推法

用低阶公式近似结果提高到高阶级精确度：

对于低阶的误差有如下形式：

$$\begin{aligned} T_0(x) - f(x) &= a_1 x + a_2 x^2 + \dots \\ T_0(x/2) - f(x/2) &= a_1(x/2) + a_2(x/2)^2 + \dots \\ \frac{2T_0(x/2) - T_0(x)}{2 - 1} - f(x) &= -0.5a_2 x^2 + \dots \end{aligned}$$

于是就从 $O(x)$ 提升到 $O(x^2)$ 于是就有了：

$$\begin{aligned} T_1(h) &= \frac{2T_0(h/2) - T_0(h)}{2 - 1} \rightarrow O(h^2) \\ T_2(h) &= \frac{2^2 T_1(h/2) - T_1(h)}{2^2 - 1} \rightarrow O(h^3) \\ T_m(h) &= \frac{2^m T_{m-1}(h/2) - T_{m-1}(h)}{2^m - 1} \rightarrow O(h^{m+1}) \end{aligned}$$

Adaptive Quadrature Methods p188

简单来说，按照之前的复合求积分中等间距划分并不是适用于所有问题，于是这个方法就用来动态划分积分区间。

$$\int_a^b f(x)dx = S(a, b) - R_1$$

$$\int_a^b f(x)dx = s(a, \frac{a+b}{2}) + s(\frac{a+b}{2}, b) + R_2$$

$$R_2 = \frac{1}{15} |s(a, b) - s(a, \frac{a+b}{2}) - s(\frac{a+b}{2}, b)| \leq \epsilon$$

如果这样不满足条件就再对两个子区间再使用一次，保证误差小于 $\frac{\epsilon}{2}$ 即可

Gauss Quadrature

取 n 段 $n+1$ 个节点 使得积分精度达到 $2n+1$ ，此时需要控制选择的点。

定理：所有选择的点满足 $W(x) = \prod_{i=0}^n (x - x_i)$ 与所有不超过 n 的都是正交的。

于是需要构造出这样的正交多项式：

Legendre Polynomials： 定义在 $[-1, 1]$ 上并且 $W(x) = 1$

注意：是要考虑权函数的由于一般不强制规定权重，可以直接使用 1 作为权重：P199 页给出了 5 阶的式子

Legendre 多项式： $P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n]$ 。

其内积关系为： $(P_k, P_l) = \begin{cases} 0, & k \neq l \\ \frac{2}{2k+1}, & k = l \end{cases}$

根据 $P_0(x) = 1, P_1(x) = x$ ，我们有递推关系：

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

Gauss-Chebyshev 定义在 $[-1, 1]$ 上并且 $W(x) = \frac{1}{\sqrt{1-x^2}}$

$$x = \cos\left(\frac{2k-1}{2n}\pi\right) (k = 1, \dots, n)$$

步骤：

1. 如果不是在 $[-1, 1]$ 的范围内使用 $x = \frac{b-a}{2}t + \frac{a+b}{2}$ 代换
2. 明确需要几个点去计算，找到几个根
3. 带入到式子 $\int_{-1}^1 w(x)f(x)dx = \sum A_i f(x_i)$ 根据 n 个点 $2n-1$ 阶精确 把 $f(x)$ 分别带入 $1, x, x^2, x^3, \dots$ 得出系数 如果是 Legendre 在 p199 给出了结果。

4. 之后就是带入完成计算。

Chapter 5 Initial-Value Problems for Ordinary Differential Equations

找到初值 w_0, w_1, \dots 满足 $w_i = y(t_i)$

The Elementary Theory of Initial-Value Problems

Lipschitz condition

存在 $L > 0$ 使得 $|f(t, y_1) - f(t, y_2)| \leq L(y_1 - y_2)$ 此时 L 称为 Lipschitz constant. **偏导数存在上界**

$D = \{(t, y) | a \leq t \leq b, -\infty < y < +\infty\}$ 在 D 上有 Lipschitz constant 则 IVP 有解:

$$y'(t) = f(t, y), a \leq t \leq b, y(a) = \alpha$$

well-posed problem

1. 有唯一解
2. 添加一个扰动向，对结果影响小

1. A unique solution, $y(t)$, to the problem exists;
2. For any $\varepsilon > 0$, there exists a positive constant $k(\varepsilon)$, such that whenever $|\varepsilon_0| < \varepsilon$ and $\delta(t)$ is continuous with $|\delta(t)| < \varepsilon$ on $[a, b]$, a unique solution, $z(t)$, to

$$z'(t) = f(t, z) + \delta(t), \quad a \leq t \leq b, \quad z(a) = \alpha + \varepsilon_0 \quad (* \text{ perturbed problem}*) \\ \text{exists with } |z(t) - y(t)| < k(\varepsilon) \varepsilon, \text{ for all } a \leq t \leq b.$$

在 Lipschitz 条件下，初值问题是良态的

local truncation error:

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + h\phi(t_i, w_i) \\ \tau_{i+1}(h) &= \frac{y_{i+1} - (y_i + h\phi(t_i, y_i))}{h} = \frac{y_{i+1} - y_i}{h} - \phi(t_1, y_i) \end{aligned}$$

Euler's Method

Euler 法的思想是，用 $f(t, y)$ 在 (t_i, y_i) 处的线性近似值来代替 $f(t, y)$, Difference wquation:

$$w_{i+1} = w_i + h f(t_i, w_i)$$

Theorem: Suppose f is continuous and satisfies a Lipschitz condition with constant L on $D = \{(t, y) | a \leq t \leq b, -\infty < y < \infty\}$ and that a constant M exists with $|y''(t)| \leq M$ for all $a \leq t \leq b$. Let $y(t)$ denote the unique solution to the IVP $y'(t) = f(t, y)$, $a \leq t \leq b$, $y(a) = \alpha$, and w_0, w_1, \dots, w_n be the approximations generated by Euler's method for some positive integer n . Then for $i = 0, 1, \dots, n$

$$|y_i - w_i| \leq \frac{hM}{2L} [e^{L(t_i-a)} - 1]$$

➤ The roundoff error

$$w_0 = \alpha + \delta_0$$

$$w_{i+1} = w_i + h f(t_i, w_i) + \delta_{i+1} \quad (i = 0, \dots, n-1)$$

Theorem: Let $y(t)$ denote the unique solution to the IVP

$$y'(t) = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

and w_0, w_1, \dots, w_n be the approximations obtained using the above difference equations. If $|\delta_i| < \delta$ for $i = 0, 1, \dots, n$, then for each i

$$|y_i - w_i| \leq \frac{1}{L} \left(\frac{hM}{2} + \frac{\delta}{h} \right) [e^{L(t_i-a)} - 1] + |\delta_0| e^{L(t_i-a)}$$

- 要保证 $h \geq \sqrt{\frac{2\delta}{M}}$

local traction error

$$\begin{aligned} \tau_{i+1} &= \frac{y_{i+1} - w_{i+1}}{h} = \frac{[y_i + hy'(t_i) + \frac{h^2}{2} y''(\xi_i)] - [y_i + hf(t_i, y_i)]}{h} \\ &= \frac{h}{2} y''(\xi_i) = O(h) \end{aligned}$$

Method of order 1

Implicit Euler Method 隐式欧拉法

稳定性更好

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + h f(t_{i+1}, w_{i+1}) \\ \tau_{i+1} &= -\frac{h}{2} y''(\xi) = O(h) \end{aligned}$$

Trapezoidal Method

$$w_{i+1} = w_i + \frac{h}{2}(f(t_i, w_i) + f(t_{i+1}, w_{i+1}))$$

$$\tau_{i+1} = O(h^2)$$

缺点：需要迭代求解 solve iteratively

Double-step Method

$$w_0 = \alpha, w_1 = \beta$$

$$w_{i+1} = w_{i-1} + 2hf(t_i, w_i)$$

$$\tau_{i+1} = O(h^2)$$

Euler's explicit	Simple	Low order accuracy
Euler's implicit	<i>stable</i>	Low order accuracy and time consuming
Trapezoidal	More accurate	Time consuming
Double-step	More accurate, and explicit	Requires one extra initial point

Higher Order Taylor Methods

$$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2}f'(t_i, y_i) + \cdots + \frac{h^n}{n!}f^{(n-1)}(t_i, y_i) + \frac{h^{n+1}}{(n+1)!}f^{(n)}(\xi_i, y(\xi_i))$$

n 阶的 Taylor 法：

$$w_0 = \alpha$$

$$w_{i+1} = w_i + hT^{(n)}(t_i, w_i) \quad (i = 0, \dots, n-1)$$

where $T^{(n)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2}f'(t_i, w_i) + \dots + \frac{h^{n-1}}{n!}f^{(n-1)}(t_i, w_i)$

trunction error: $O(h)$

Runge-Kutta Methods

不需要计算泰勒方法中的高阶导数，同时能保证高阶数的截断误差

二阶 runge-kutta (single step)

$$\begin{cases} w_{i+1} = w_i + h[\lambda_1 K_1 + \lambda_2 K_2] \\ K_1 = f(t_i, w_i) \\ K_2 = f(t_i + ph, w_i + phK_1) \end{cases}$$

对 K_2 进行泰勒展开，再带入到第一个式子中，保证二阶截断误差，于是对 y_{i+1} 泰勒展开的结果应当和原式系数相同

$$\begin{aligned} w_{i+1} &= y_i + (\lambda_1 + \lambda_2)hy'(t_i) + \lambda_2 ph^2 y''(t_i) + O(h^3) \\ y_{i+1} &= y_i + hy'(t_i) + h^2/2y''(t_i) + O(h^3) \end{aligned}$$

$$\lambda_1 + \lambda_2 = 1, \lambda_2 p = \frac{1}{2}$$

- **Midpoint Method**

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + hf(t_i + h/2, w_i + h/2f(t_i, w_i)) \\ \lambda_1 &= 0, \lambda_2 = 1, p = \frac{1}{2} \\ w_{i+1} &= w_i + hK_2 \\ K_1 &= f(t_i, w_i) \\ K_2 &= f(t_i + \frac{h}{2}, w_i + \frac{h}{2}K_1) \end{aligned}$$

- **Modified Euler Method**

$$\begin{cases} w_0 = \alpha \\ w_{i+1} = w_i + h(\frac{1}{2}K_1 + \frac{1}{2}K_2) \\ K_1 = f(t_i, w_i) \\ K_2 = f(t_i + h, w_i + hK_1) \end{cases}$$

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + h(1/2f(t_i, w_i) + 1/2f(t_i + h, w_i + hf(t_i, w_i))) \\ \lambda_1 &= 0.5, \lambda_2 = 0.5, p = 1 \end{aligned}$$

注意：这个和之前的梯形法是有区别的，梯形法是隐式求解，这个是显示求解

- **Heun's Method**

$$\begin{aligned}
w_0 &= \alpha w_{i+1} = w_i + h\left(\frac{1}{4}K_1 + \frac{3}{4}K_2\right) \\
K_1 &= f(t_i, w_i) \\
K_2 &= f\left(t_i + \frac{2}{3}h, w_i + \frac{2}{3}hK_1\right) \\
\lambda_1 &= \frac{1}{4}, \lambda_2 = \frac{3}{4}, p = \frac{2}{3}
\end{aligned}$$

高阶Runge-Kutta

$$\begin{cases}
w_0 = \alpha \\
w_{i+1} = w_i + h(\lambda_1 K_1 + \lambda_2 K_2 + \dots + \lambda_m K_m) \\
K_1 = f(t_i, w_i) \\
K_2 = f(t_i + \alpha_2 h, w_i + \beta_{21} h K_1) \\
K_3 = f(t_i + \alpha_3 h, w_i + \beta_{31} h K_1 + \beta_{32} h K_2) \\
\vdots \\
K_m = f(t_i + \alpha_m h, w_i + \beta_{m1} h K_1 + \beta_{m2} h K_2 + \dots + \beta_{m,m-1} h K_{m-1})
\end{cases}$$

- 4 阶常用

$$\begin{cases}
w_0 = \alpha \\
w_{i+1} = w_i + h\left(\frac{1}{6}K_1 + \frac{1}{3}K_2 + \frac{1}{3}K_3 + \frac{1}{6}K_4\right) \\
K_1 = f(t_i, w_i) \\
K_2 = f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}K_1\right) \\
K_3 = f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}K_2\right) \\
K_4 = f(t_i + h, w_i + hK_3)
\end{cases}$$

我们给出每步的求值次数和局部阶段误差的阶之间的关系：

Evaluations per step	2	3	4	$5 \leq n \leq 7$	$8 \leq n \leq 9$	$10 \leq n$
Best possible local truncation error	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^{n-1})$	$O(h^{n-2})$	$O(h^{n-3})$

这说明了为什么人们更喜欢使用具有较小步长的小于 5 阶的 Runge-Kutta 方法。

因为 Runge-Kutta 方法是基于 Taylor 展开的，所以 y 必须足够平滑，才能获得更高阶方法的更高精度。通常情况下，人们更喜欢使用较小步长的低阶方法，而不是使用较大步长的高阶方法。

如果 y 不够光滑，那么高阶的 Runge-Kutta 方法也不会有很好的效果，所以一般会用低阶的 Runge-Kutta 方法，但是步长会更小。

Multistep Methods

求解初值问题

$$y' = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha$$

的 m 步多步法 (m -step multistep method) 的一般形式为

$$\begin{aligned} w_{i+1} = & a_{m-1}w_i + a_{m-2}w_{i-1} + \dots + a_0w_{i+1-m} \\ & + h[b_m f(t_{i+1}, w_{i+1}) + b_{m-1}f(t_i, w_i) + \dots + b_0 f(t_{i+1-m}, w_{i+1-m})] \end{aligned}$$

其中 $h = (b - a)/N$, 给定 m 个初始值 w_0, w_1, \dots, w_{m-1} , a_0, a_1, \dots, a_{m-1} 和 b_0, b_1, \dots, b_m 是常数。

$b_m = 0$ 的方法称为显式 (explicit) ; $b_m \neq 0$ 的方法称为隐式 (implicit)

Adams Bashforth 显示M步 具体推导p258

注意显示m步：只有m个点参与插值，m-1次多项式

就是对于初始 m 个点进行一个插值，书上推荐是使用牛顿的后向插值，也可以使用拉格朗日等等，结果应当是一致的。最终结果：

$$\begin{aligned} w_{i+1} &= w_i + h \int_0^1 P_{m-1}(t_i + sh) ds \\ &= \int_{t_i}^{t_{i+1}} f(t, y(t)) dt \\ w_{i+1} - w_i &= h[f(t_i, y(t_i)) + \frac{1}{2} \nabla f(t_i, y(t_i)) \dots] \end{aligned}$$

error

$$\begin{aligned} w_{i+1} &= w_i + h \int_0^1 P_{m-1}(t_i + sh) ds \\ &= w_i + h \sum_{k=0}^{m-1} \nabla^k f(t_i, y(t_i)) (-1)^k \int_0^1 \binom{-s}{k} ds \\ \int_{t_i}^{t_{i+1}} R_{m-1}(t) dt &= \frac{h^{m+1}}{m!} \int_0^1 f^{(m)}(\xi_i, y(\xi_i)) s(s+1)\dots(s+m-1) ds \\ &= \frac{h^{m+1} f^{(m)}(\mu_i, y(\mu_i))}{m!} \int_0^1 s(s+1)\dots(s+m-1) ds \\ &= h^{m+1} f^{(m)}(\mu_i, y(\mu_i)) (-1)^m \int_0^1 \binom{-s}{m} ds \end{aligned}$$

trunction error: $O(h^m)$

后向差分前的系数:

$$\text{• } (-1)^k \int_0^1 \binom{-s}{k} ds$$

k	$(-1)^k \int_0^1 \binom{-s}{k} ds$
0	1
1	$\frac{1}{2}$
2	$\frac{5}{12}$
3	$\frac{3}{8}$
4	$\frac{251}{720}$
5	$\frac{95}{288}$

★综上结果如下:

$$w_{i+1} = w_i + (b_m f_i + \dots + b_0 f_{i-m+1}) + A_k h^{m+1} f^{(m)}(u)$$

$$\tau_{i+1} = A_k h^m f^{(m)}$$

m	f_i	f_{i-1}	f_{i-2}	f_{i-3}	...	A_k
1	1					$\frac{1}{2}$
2	$\frac{3}{2}$	$-\frac{1}{2}$				$\frac{5}{12}$
3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$			$\frac{3}{8}$
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$		$\frac{251}{720}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

查表可得, Adams-Bashforth 显式 4 步方法的为

$$w_{i+1} = w_i + \frac{h}{24} (55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$$

Adams-Moulton 隐式m步

与显示不同 将选取m+1个点 $(t_{i+1}, f_{i+1}) - (t_{i+1-m}, f_{i+1-m})$

使用前向差分结果如下：

m	f_{i+1}	f_i	f_{i-1}	f_{i-2}	...	B_k
0	1					$-\frac{1}{2}$
1	$\frac{1}{2}$	$\frac{1}{2}$				$-\frac{1}{12}$
2	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$			$-\frac{1}{24}$
3	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$		$-\frac{19}{720}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

$$w_{i+1} = w_i + (b_{m+1}f_{i+1} + b_m f_i + \dots + b_0 f_{i-m+1}) + A_k h^{m+1} f^{(m)}(u)$$

$$\tau_{i+1} = B_k h^{m+1} f^{(m+2)}$$

这里三阶的一个例子：

$$w_{i+1} = w_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2})$$

Adams 预测矫正

1. Runge-Kutta 算m个初始值
2. 预测： Adams-bashforth 显示算出 $w_m \dots w_{n-1}$
3. 纠正： Admas-Moulton 隐式m步算法计算出 $w_m \dots w_{N-1}$

这三部局部截断误差必须相等！！

最常用 4 阶显示 初始值来自四阶级Runge-Kutta 三阶隐式

通过泰勒展开获得

- 展开的精度是由 多步决定的 如果是m步，则展开到m阶
- 此时y展开到m+1阶 f 展开到 m 阶级

$$w_{i+1} = a_2 w_i + a_1 w_{i-1} + a_0 w_{i-2} + h [b_3 f_i + b_2 f_{i-1} + b_1 f_{i-2} + b_0 f_{i-3}]$$

Solution: Expand $y_{i-1}, y_{i-2}, f_{i-1}, f_{i-2}, f_{i-3}$ and $y(t_{i+1})$ at t_i .

$$\tau_{i+1} = \frac{y_{i+1} - w_{i+1}}{h} = O(h^4) \text{ with the assumption } w_i = y_i$$

$$y_{i-1} = y_i - hy'_i + \frac{1}{2}h^2 y''_i - \frac{1}{6}h^3 y'''_i + \frac{1}{24}h^4 y^{(4)}_i + O(h^5)$$

$$y_{i-2} = y_i - 2hy'_i + 2h^2 y''_i - \frac{4}{3}h^3 y'''_i + \frac{2}{3}h^4 y^{(4)}_i + O(h^5)$$

$$f_{i-1} = y'_i - hy''_i + \frac{1}{2}h^2 y'''_i - \frac{1}{6}h^3 y^{(4)}_i + O(h^4)$$

$$f_{i-2} = y'_i - 2hy''_i + 2h^2 y'''_i - \frac{4}{3}h^3 y^{(4)}_i + O(h^4)$$

$$f_{i-3} = y'_i - 3hy''_i + \frac{9}{2}h^2 y'''_i - \frac{9}{2}h^3 y^{(4)}_i + O(h^4)$$

$$y(t_{i+1}) = \underline{y_i} + \underline{hy'_i} + \frac{1}{2}\underline{h^2 y''_i} + \frac{1}{6}\underline{h^3 y'''_i} + \frac{1}{24}\underline{h^4 y^{(4)}_i} + O(h^5)$$

5 equations and 7 unknowns

☞ Let $a_0 = a_1 = 0 \rightarrow$ Adams-Basforth explicit method

☞ Replace f_{i-1} by f_{i+1} and let $a_0 = a_1 = 0 \rightarrow$ Adams-Moulton implicit method

☞ Replace f_{i-3} by w_{i-3} , we can obtain another family of methods of order 4, including the explicit Milne's method:

$$w_{i+1} = w_{i-3} + \frac{4h}{3}(2f_i - f_{i-1} + 2f_{i-2})$$

which has local truncation error $\frac{14}{45}h^4 y^{(5)}(\xi_i)$, $\xi_i \in (t_{i-3}, t_{i+1})$

☞ Let $a_0 = 0$ and $a_1 = 1 \rightarrow$ Simpson implicit method

$$w_{i+1} = w_{i-1} + \frac{h}{3}(f_{i+1} + 4f_i + f_{i-1})$$

which has local truncation error $-\frac{h^4}{90} y^{(5)}(\xi_i)$, $\xi_i \in (t_{i-1}, t_{i+1})$

Higher-Order Equations and Systems of Differential Equations 高阶常微分方程组

对 m 阶微分方程组:

$$\begin{cases} u'_1(t) = f_1(t, u_1(t), u_2(t), \dots, u_m(t)) \\ \vdots \\ u'_m(t) = f_m(t, u_1(t), u_2(t), \dots, u_m(t)) \end{cases}$$

给定 m 个初始值 $u_1(a), u_2(a), \dots, u_m(a)$. 我们可以用 m 步的 Runge-Kutta 法来求解。

用矩阵的形式可以记作

$$\begin{cases} \mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t)) \\ \mathbf{y}(a) = \boldsymbol{\alpha} \end{cases}$$

和之前的Runge-Kutta一致，只是在计算k的时候需要计算f中的每一个k都成立P280

高阶的情况:

➤ Higher-order differential equation

$$\begin{cases} \mathbf{y}^{(m)}(t) = \mathbf{f}(t, \mathbf{y}, \mathbf{y}', \dots, \mathbf{y}^{(m-1)}), \quad a \leq t \leq b \\ \mathbf{y}(a) = \boldsymbol{\alpha}_1, \mathbf{y}'(a) = \boldsymbol{\alpha}_2, \dots, \mathbf{y}^{(m-1)}(a) = \boldsymbol{\alpha}_m \end{cases}$$



Reduce the higher-order differential equation into a system of 1st-order differential equations.

Let $u_1(t) = y(t), u_2(t) = y'(t), \dots, u_m(t) = y^{(m-1)}(t)$. This produces

$$\xrightarrow{\hspace{1cm}} \begin{cases} u'_1 = y' = u_2 \\ u'_2 = y'' = u_3 \\ \vdots \\ u'_{m-1} = y^{(m-1)} = u_m \\ u'_m = y^{(m)} = f(x, u_1, \dots, u_m) \end{cases} \quad \text{with the initial conditions:} \quad \begin{array}{l} u_1(a) = \alpha_1 \\ u_2(a) = \alpha_2 \\ \dots \\ u_m(a) = \alpha_m \end{array}$$

P284 例子书上例子更好

Stability 稳定性

相容 | Consistency

Definition: A one-step difference equation method with local truncation error $\tau_i(h)$ is said to be **consistent** with the differential equation it approximates if

$$\lim_{h \rightarrow 0} \max_{1 \leq i \leq n} |\tau_i(h)| = 0$$

For multistep methods it is also required that for $i = 1, 2, \dots, m - 1$

$$\lim_{h \rightarrow 0} |w_i - y_i| = 0.$$

注意到这个定义是在**局部**上的定义

收敛 | Convergence

Definition: A one-step difference equation method is said to be **convergent** with respect to the differential equation it approximates if

$$\lim_{h \rightarrow 0} \max_{1 \leq i \leq n} |w_i - y_i| = 0$$

Same for multistep methods.

这是对**整体**而言

特征方程与稳定性

已知方程

$$\begin{aligned} w_0 &= \alpha, w_1 = \alpha_1, \dots, w_{m-1} = \alpha_{m-1} \\ w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \dots + a_0w_{i+1-m} + hF(t_i, h, w_{i+1}, w_i, \dots, w_{i+1-m}) \end{aligned}$$

我们给出一个相关的多项式，称为**特征多项式** (characteristic polynomial)：

$$P(\lambda) = \lambda^m - a_{m-1}\lambda^{m-1} - a_{m-2}\lambda^{m-2} - \dots - a_0$$

- 如果 $P(\lambda)$ 的所有根的模都小于等于 1，且取等时为单根，则称该方法满足**根条件** (root condition)
- 如果有且仅有一个根的模等于 1，则该方法是**强稳定** (strongly stable) 的
- 如果有多个根的模等于 1，则该方法是**弱稳定** (weakly stable) 的
- 如果方法不满足**根条件**，则该方法是不稳定的

稳定性测试方程：

$$y' = \lambda y, y(0) = \alpha, \text{ where } \operatorname{Re}(\lambda) < 0$$

假设舍入误差只在初始点引入。如果这个初始误差会在某个步长 hh 下减小，那么这个方法就被称为**绝对稳定** (absolutely stable)，此时 $H = \lambda h$ 。所有这样的 H 的集合构成了**绝对稳定域** (region of absolute stability)。此时 H 可以考虑为是一个复数，求解范围一定是在复数域上的。 λ 实部是小于 0 的， h 是个实数

stability 和 traction error 没有关系

以下均为该方法的应用：

显示欧拉方法：对于 h 有要求，所以是条件稳定

$$\begin{aligned} w_{n+1} &= w_n + hf(t_i, w_n) \\ w_{i+1} &= w_n(1 + \lambda h) = w_n(1 + H) = \alpha(H + 1)^n \\ w_{n+1}^* &= (1 + H)^n(\alpha + \epsilon) \\ |1 + H| &< 1 \end{aligned}$$

隐式欧拉方法：对于所有的 λ 都是成立的

$$\begin{aligned} w_{i+1} &= w_i + hf(t_{i+1}, w_{i+1}) \\ w_{i+1} &= w_i + h\lambda w_{i+1}w_{i+1} = w_i \frac{1}{1 - H} \\ \left| \frac{1}{1 - H} \right| &< 1 \end{aligned}$$

二阶隐式runge-kutta

$$\begin{aligned} w_{i+1} &= w_i + hK_1 \\ K_1 &= f(t_i + \frac{h}{2}, w_i + \frac{h}{2}K_1) \\ K_1 &= \lambda(w_i + \frac{h}{2}K_1) \\ K_1 &= \frac{\lambda w_i}{1 - \frac{1}{2}h\lambda} \\ w_{i+1} &= w_i + hK_1 \\ &= (\frac{2 + H}{2 - H})^{i+1}\alpha \\ \left| \frac{2 + H}{2 - H} \right| &< 1 \end{aligned}$$

在四阶 显式 Runge-Kutta 法中，我们有

$$\begin{cases} w_{i+1} = w_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(t_i, w_i) \\ K_2 = f(t_i + \frac{1}{2}h, w_i + \frac{1}{2}hK_1) \\ K_3 = f(t_i + \frac{1}{2}h, w_i + \frac{1}{2}hK_2) \\ K_4 = f(t_i + h, w_i + hK_3) \end{cases}$$

在这个测试方程中，我们有

$$\begin{aligned} K_1 &= \lambda w_i \\ K_2 &= \lambda(w_i + \frac{1}{2}hK_1) = \lambda w_i(1 + \frac{1}{2}H) \\ K_3 &= \lambda(w_i + \frac{1}{2}hK_2) = \lambda w_i(1 + \frac{1}{2}H + \frac{1}{4}H^2) \\ K_4 &= \lambda(w_i + hK_3) = \lambda w_i(1 + H + \frac{1}{2}H^2 + \frac{1}{4}H^3) \\ w_{i+1} &= w_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ &= w_i + \frac{h}{6}\lambda w_i(1 + 2(1 + \frac{1}{2}H) + 2(1 + \frac{1}{2}H + \frac{1}{4}H^2) + (1 + H + \frac{1}{2}H^2 + \frac{1}{4}H^3)) \\ &= w_i + \frac{H}{6}w_i(6 + 3H + H^2 + \frac{1}{4}H^3) \\ &= w_i(1 + H + \frac{1}{2}H^2 + \frac{1}{6}H^3 + \frac{1}{24}H^4) \\ &= \alpha(1 + H + \frac{1}{2}H^2 + \frac{1}{6}H^3 + \frac{1}{24}H^4)^{i+1} \end{aligned}$$

对于微分方程组的稳定性：与之前不同在于这里 λ 取值是固定为系数矩阵特征值的

考虑一个微分方程组

$$\begin{cases} u'_1 = 9u_1 + 24u_2 + 5 \cos t - \frac{1}{3} \sin t, & u_1(0) = \frac{4}{3} \\ u'_2 = -24u_1 - 51u_2 - 9 \cos t + \frac{1}{3} \sin t, & u_2(0) = \frac{2}{3} \end{cases}$$

应用欧拉显式法，我们该如何选择步长 h 才能保证稳定性？

我们将条件改写为矩阵形式：

$$\begin{pmatrix} u'_1 \\ u'_2 \end{pmatrix} = \begin{pmatrix} 9 & 24 \\ -24 & -51 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 5 \cos t - \frac{1}{3} \sin t \\ -9 \cos t + \frac{1}{3} \sin t \end{pmatrix}$$

应用欧拉显式法，我们有

$$\begin{pmatrix} u_1^{i+1} \\ u_2^{i+1} \end{pmatrix} = \begin{pmatrix} u_1^i \\ u_2^i \end{pmatrix} + h \begin{pmatrix} 9 & 24 \\ -24 & -51 \end{pmatrix} \begin{pmatrix} u_1^i \\ u_2^i \end{pmatrix} + h \begin{pmatrix} 5 \cos t_i - \frac{1}{3} \sin t_i \\ -9 \cos t_i + \frac{1}{3} \sin t_i \end{pmatrix}$$

记 $\mathbf{A} = \begin{pmatrix} 9 & 24 \\ -24 & -51 \end{pmatrix}$, 则

$$\begin{pmatrix} u_1^{i+1} \\ u_2^{i+1} \end{pmatrix} = (\mathbf{I} + h\mathbf{A}) \begin{pmatrix} u_1^i \\ u_2^i \end{pmatrix} + h \begin{pmatrix} 5 \cos t_i - \frac{1}{3} \sin t_i \\ -9 \cos t_i + \frac{1}{3} \sin t_i \end{pmatrix}$$

我们给初值加上一个扰动项 ϵ_0 和 μ_0 , 即 $\mathbf{u}_0^* = \mathbf{u}_0 + \begin{pmatrix} \epsilon_0 \\ \mu_0 \end{pmatrix}$, 则

$$\begin{pmatrix} u_1^{i+1*} \\ u_2^{i+1*} \end{pmatrix} = (\mathbf{I} + h\mathbf{A}) \begin{pmatrix} u_1^{i*} \\ u_2^{i*} \end{pmatrix} + h \begin{pmatrix} 5 \cos t_i - \frac{1}{3} \sin t_i \\ -9 \cos t_i + \frac{1}{3} \sin t_i \end{pmatrix}$$

上面两个式子相减, 我们有

$$\begin{pmatrix} \epsilon_{i+1} \\ \mu_{i+1} \end{pmatrix} = (\mathbf{I} + h\mathbf{A}) \begin{pmatrix} \epsilon_i \\ \mu_i \end{pmatrix}$$

所以

$$\begin{pmatrix} \epsilon_{i+1} \\ \mu_{i+1} \end{pmatrix} = (\mathbf{I} + h\mathbf{A})^{i+1} \begin{pmatrix} \epsilon_0 \\ \mu_0 \end{pmatrix}$$

要确保稳定性, 我们需要 $|(\mathbf{I} + h\mathbf{A})^{i+1}| < 1$ 。

也就是说, 我们需要 $|1 + h\lambda| < 1 \Leftrightarrow -2 < h\lambda < 0$, 其中 λ 是 \mathbf{A} 的特征值。

我们求解 \mathbf{A} 的特征值, 得到 $\lambda_1 = -3$ 和 $\lambda_2 = -39$ 。

所以, 我们需要 $-2 < h\lambda_1 < 0 \Leftrightarrow \frac{2}{3} > h > 0$ 和 $-2 < h\lambda_2 < 0 \Leftrightarrow \frac{2}{39} > h > 0$ 。

所以, 我们需要 $0 < h < \frac{2}{39}$ 。

Finish!