



浙江大學  
ZHEJIANG UNIVERSITY

# 数值分析方法笔记

## 数值分析

姓名 \_\_\_\_\_ Soleil

学号 \_\_\_\_\_ lalalala~

学院 \_\_\_\_\_ 信电学院 (开课学院)

2025 年 1 月 8 日

# 数值分析

## 摘要

如题，期末复习向，重点是把学过的公式进行总结和推导，以及一些英文词汇的辨识

**总体是顺着 PPT 的思路进行的，因为期末考试只能带 PPT**

数值分析是用来解决什么问题的——

Introduce the applied numerical methods, including Numerical approaches for

- linear(线性) equations 线性方程
- nonlinear equations 非线性方程
- differentiation and integration 微分与积分
- partial differential equations 偏微分方程

主要分为以上四个大方向，那我们下面也从这四个方向进行展开

## 1 计算机算术、算法与收敛性 | Computer Arithmetic 、 Algorithms and Convergence

Basics: 1 Byte(字节) = 8 Bits(比特)

### 1.1 浮点数与截断误差

- Long real format
  - Default data type in MATLAB, 'double' in C
  - Base: 2

1 bit	11 bits	52 bits
s	c	f

$$(-1)^s 2^{c-1023} (1+f)$$

- Ex: 0 10000000011 101110...0
- Maximum /  $2^{1023} \cdot (2 - 2^{-52}) \approx 0.17977 \times 10^{309}$
- Minimum /  $2^{-1022} \cdot (1 + 0) \approx 0.22251 \times 10^{-307}$
- Overflow / underflow
- [http://en.wikipedia.org/wiki/IEEE\\_floating\\_point](http://en.wikipedia.org/wiki/IEEE_floating_point)

图 1: Long-real-format

1. 会发现，能表示的最大位数并不是想象中的  $2^{1024}$  是因为占用了最高位和最低位 Inf and NaN
2. 在有效位数截断时，有两种方式，一种直接丢弃 (way of chopping)，另一种则是在对应位加上 5 再舍去 (way of Roundoff)

- Base: 10
- $k$ -digit decimal machine number:

$$\pm 0.d_1d_2\dots d_k \times 10^n, \quad 1 \leq d_1 \leq 9, \quad 0 \leq d_i \leq 9$$

- Any positive number within the numerical range can be written as

$$y = 0.d_1d_2\dots d_kd_{k+1}d_{k+2}\dots \times 10^n$$

- Two ways to represent  $y$  with  $k$  digits:

- ① Chopping:

$$f(y) = 0.d_1d_2\dots d_k \times 10^n$$

- ② Roundoff: Add  $5 \times 10^{n-(k+1)}$  and chop:

$$f(y) = 0.\delta_1\delta_2\dots\delta_k \times 10^n$$

- Roundoff error

图 2: ERROR-ANALYSIS

## 1.2 误差与有效位数 | Errors and Significant digits

**定义 1.1.** If  $p^*$  is an approximation (近似) to  $p$ , the **absolute error** (绝对误差) is  $|p - p^*|$ , and the **relative error** is  $\frac{|p - p^*|}{|p|}$ , provided that (前提是/在... 条件下)  $p \neq 0$ .

**定义 1.2. 有效数字:** 下图公式中使得等式成立的最大整数  $t$  值称为  $p^*$  的有效位数

Significant digits:

The number  $p^*$  is said to approximate  $p$  to  $t$  significant digits if  $t$  is the largest nonnegative integer for which

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

图 3: 有效数字定义

- 机器的浮点数运算通常会涉及到四种基本运算：加法、减法、乘法和除法，每种浮点运算都是浮点数相运算，最后结果也存储为浮点数

- 需要注意：两个数值非常接近时，它们的差值可能会丢失许多有效数字，这样会使得误差大大增长

考虑一个 equation  $x^2 + 62.10x + 1 = 0$ , 如果我们采用四舍五入算法保留一定的有效位数（四位），最后的误差达到了 24%，非常高；但是但是，如果将无理数放在分母上，计算误差就会降低到  $6.2 \times 10^{-4}$

## 1.3 Algorithms(收敛)

### 1.3.1 误差增长速率

主要有两种典型

1. 误差线性增长
2. 误差指数增长

### 1.3.2 误差收敛速率

#### Definition

Suppose  $\{\beta_n\}_{n=1}^{\infty}$  is a sequence converging to zero, and  $\{\alpha_n\}_{n=1}^{\infty}$  converges to a number  $\alpha$ . If a positive constant  $K$  exists with

$$|\alpha_n - \alpha| \leq K|\beta_n|, \quad \text{for large } n,$$

then we say that  $\{\alpha_n\}_{n=1}^{\infty}$  converges to  $\alpha$  with rate of convergence  $O(\beta_n)$ , indicated by  $\alpha_n = \alpha + O(\beta_n)$ .

#### Polynomial rate of convergence

Normally we will use

$$|\beta_n| = \frac{1}{n^p}$$

and look for the largest value  $p > 0$  such that  $\alpha_n = \alpha + O(\frac{1}{n^p})$

图 4: 数列收敛速度定义

**Definition**

Suppose that  $\lim_{h \rightarrow 0} G(h) = 0$  and  $\lim_{h \rightarrow 0} F(h) = L$ . If a positive constant  $K$  exists with

$$|F(h) - L| \leq K|G(h)|, \quad \text{for sufficiently small } h,$$

then we write  $F(h) = L + O(G(h))$ .

**Polynomial rate of convergence**

Normally we will use

$$G(h) = h^p,$$

and look for the largest value  $p > 0$  such that  $F(h) = L + O(h^p)$ .

→ Take a ‘log’ operator..., very helpful for finite element method, finite difference method...

图 5: 函数收敛速度定义

## 2 单变量求根 | Solutions of Equations in One Variable

参考文章: 知乎

**定理 2.1.** *Rolle's Theorem*(罗尔中值定理)

**定理 2.2.** *Mean Value Theorem* (又叫拉格朗日中值定理)

**定理 2.3.** *Intermediate Value Theorem* (介值定理)

### 2.1 寻根问题 | The Way of Root Finding Problem

Forward problem and Inverse problem (正问题与反问题, 本节问题显然是反问题)

1. 二分法 | The Bisection Method(Because of IVT)
2. 不动点法 | The Fixed-Point Problem
3. 牛顿法 | Newton's Method

## 2.2 二分法 | The Bisection Method

### 2.2.1 Method

在区间  $(a, b)$  中有零点，求出区间中点  $\frac{a+b}{2}$  的函数值，记为  $f(x_1)$ ，那么当  $f(x_1) = 0$ ，证明找到零点了，迭代结束，不是的话，假设  $f(a)f(x_1) < 0$ ，则在  $(a, x_1)$  中迭代求解<sup>\*</sup>，当然也可能  $f(b)f(x_1) < 0$ ，那么就在  $(x_1, b)$  中迭代求解。

### 2.2.2 收敛性 | Convergence

如果根位于  $r$ ，经过  $n$  次迭代后区间为  $[a_n, b_n]$ ，则每次迭代后区间长度减半，即

$$|b_n - a_n| = \frac{|b_0 - a_0|}{2^n}.$$

设  $\varepsilon_n$  表示第  $n$  次迭代的误差，即中点与实际根  $r$  之间的距离。经过  $n$  次迭代后，误差满足

$$\varepsilon_n = |c_n - r| \leq \frac{|b_0 - a_0|}{2^n}.$$

因此，经过  $n$  次迭代后，最大误差指数性减小

$$x_n = x + O\left(\frac{1}{2^n}\right), \quad \text{收敛速度为 } O\left(\frac{1}{2^n}\right).$$

## 2.3 不动点法 | The Fixed-Point Problem

For function  $f(p) = 0$ , convert it to  $f(p) = g(p) - p$ , then the problem convert to The Fixed-Point Problem

**核心是构造**  $f(p) = g(p) - p$

### 2.3.1 存在性 | Existance

$f \in [a, b]$  且在  $[a, b]$  上为自身映射 ( $f$  在  $[a, b]$  上的值域包含于  $[a, b]$ ，同济高数第一章介绍过)，则其在  $[a, b]$  中存在不动点。特别地，当  $f$  可微且  $\exists 0 < k < 1, \forall x \in [a, b], |f'(x)| \leq k$  时，不动点的存在性是唯一的。**强调，只有是稳定不动点（对于  $k$  的要求在  $(0,1)$  范围内）的时候，不动点迭代才会收敛（压缩映射原理）**

### 2.3.2 收敛性 | Convergence

In the condition of  $|g'(x)| \leq k < 1$ , for  $\forall x \in [a, b]$ , we have

$$\begin{aligned} |p_n - p| &= |g(p_{n-1}) - p| = |g'(\xi)||p_{n-1} - p| \\ &\leq k|p_{n-1} - p| \end{aligned}$$

$$\begin{aligned} &\leq k^2 |p_{n-2} - p| \\ &\leq k^n |p_1 - p| \end{aligned}$$

and it's obviously that the  $p_n$  is astringent (收敛的), and this Theorem is called Contraction Mapping (压缩映射原理)

### 2.3.3 误差界 | Error Bound

$p_n$  近似  $p$  的误差界限为

$$|p_n - p| \leq k^n \max\{p_0 - a, b - p_0\}$$

以及

$$|p_n - p| \leq \frac{k^n}{1-k} |p_1 - p_0|, \quad \text{对于所有 } n \geq 1.$$

为什么逆问题这么困难——> 因为逆问题的时间复杂度要比正问题高得多 (举例: 求矩阵的逆)

## 2.4 牛顿法 | Newton's Method

### 2.4.1 Conception

考虑 Taylor 公式, 在根附近展开, 我们有

$$0 = f(p_0) + f'(p_0)(p - p_0)$$

由此得到

$$p = p_0 - \frac{f(p_0)}{f'(p_0)}$$

迭代公式转化为

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

记

$$g(p_{n-1}) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

则不难发现 Netwon's Method 其实也是一种特殊的不动点法

### 2.4.2 收敛性 | Convergence

对于一般的不动点法来说, 要求  $|g'(x)| \leq k < 1$ , 那我们来看 Netwon's Method, 有

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

与此同时，由于  $f(x_0) = 0$ , 那么我们就有  $g'(x_0) = 0$ , 那么

$$\exists \delta > 0, \forall x \in [x_0 - \delta, x_0 + \delta], \text{ 我们总有 } g'(x) = k < 1$$

这时就会发现在零点附近的邻域，Netwon's Method 一定是收敛的不动点而不是发散的，而且收敛速度也会很快

### 2.4.3 Adjoint Method (伴随方法 or BT (即神经网络中的反向传播梯度算法))

Netwon's Method 方法收敛速度很快，但是他也给计算机计算带来了问题——导数的计算也是存在一定的困难，那么这时，引入 BT 算法就会在一定程度上解决这个问题（BT 算法也是很不容易）另：Hessian(求解二阶梯度) 方法

## 2.5 收敛阶数 | Order of Convergence

### Order of Convergence

Suppose  $\{p_n\}_{n=0}^{\infty}$  is a sequence that converges to  $p$ , with  $p_n \neq p$  for all  $n$ . If positive constants  $\lambda$  and  $\alpha$  exist with

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

then  $\{p_n\}_{n=0}^{\infty}$  converges to  $p$  of order  $\alpha$ , with asymptotic error constant  $\lambda$ .

- If  $\alpha = 1$ , the sequence is linearly convergent.
- If  $\alpha = 2$ , the sequence is quadratically convergent.

图 6: Order of Convergence

### 2.5.1 Error Analysis for Fix-point Methods

#### Proof

- Since  $g'$  exists on  $(a, b)$ , applying the MVT, we have

$$p_{n+1} - p = g(p_n) - g(p) = g'(\xi_n)(p_n - p)$$

- Since  $\{p_n\}_{n=0}^{\infty}$  converges to  $p$ ,  $\{\xi_n\}_{n=0}^{\infty}$  also converges to  $p$ .
- Thus,

$$\lim_{n \rightarrow \infty} \frac{p_{n+1} - p}{p_n - p} = \lim_{n \rightarrow \infty} g'(\xi_n) = g'(p)$$

Hence, if  $g'(p) \neq 0$ , fixed-point iteration exhibits linear convergence with asymptotic error constant  $|g'(p)|$ .

图 7: Error Analysis for Fix-point Methods

### 2.5.2 Error Analysis for Netwon's Methods

#### Proof (2/2)

- When  $x = p_n$ , we have

$$p_{n+1} = g(p_n) = p + \frac{g''(\xi)}{2}(p_n - p)^2$$

- Thus,

$$\lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|^2} = \frac{|g''(p)|}{2} < \frac{M}{2}$$

图 8: Netwon's Methods

一般不动点是一阶收敛，而牛顿法是二阶收敛，收敛速度更快

## 3 非线性系统解法 | The Solutions of Nonlinear Systems

### 3.1 范数 | Norm

Tips: 此处所指向量不加说明默认为列向量

#### 3.1.1 向量范数

范数的一些基本性质（向量空间中范数的定义以及在线性空间中体现出来的性质）

#### Definition: Vector Norm

A **vector norm** on  $\mathbb{R}^n$  is a function,  $\|\cdot\|$ , from  $\mathbb{R}^n$  into  $\mathbb{R}$  with the following properties:

- ①  $\|\mathbf{x}\| \geq 0$  for all  $\mathbf{x} \in \mathbb{R}^n$
- ②  $\|\mathbf{x}\| = 0$  iff  $\mathbf{x} = 0$
- ③  $\|\alpha \mathbf{x}\| = |\alpha| \|\mathbf{x}\|$  for all  $\alpha \in \mathbb{R}$  and  $\mathbf{x} \in \mathbb{R}^n$
- ④  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$  for all  $\|\mathbf{x}\|, \|\mathbf{y}\| \in \mathbb{R}^n$ .

图 9: DefVectorNorm

范数不同的定义

### Definition: $L_1$ , $L_2$ , and $L_\infty$ Norms

The norms for the vector  $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$  are defined by:

- $L_1$  Norm

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$$

- $L_2$  Norm

$$\|\mathbf{x}\|_2 = \left\{ \sum_{i=1}^n x_i^2 \right\}^{1/2}$$

- $L_\infty$  Norm

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

Note that each of these norms reduces to the absolute value in the case  $n = 1$ .

图 10: Vector Norm

### 3.1.2 相关定理 | A Limit of a Sequence of Vectors in $\mathbb{R}^n$

**定理 3.1.** For each  $\mathbf{x} \in \mathbb{R}^n$ , we have  $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n} \|\mathbf{x}\|_\infty$

**定理 3.2.** 任意形式范数都满足三角不等式  $\|\mathbf{x} + \mathbf{y}\| = \|\mathbf{x}\| + \|\mathbf{y}\|$

### 3.1.3 矩阵范数 | Matrix Norm

详细见下一章

## 3.2 多变量函数的不动点法 | Fixed Points for Functions of Several Variables

A system of nonlinear equations has the form:

$$f_1(x_1, x_2, x_3, \dots, x_n) = 0,$$

$$f_2(x_1, x_2, x_3, \dots, x_n) = 0,$$

$$f_3(x_1, x_2, x_3, \dots, x_n) = 0,$$

...

$$f_n(x_1, x_2, x_3, \dots, x_n) = 0,$$

which can be represented by

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

The functions  $f_1, f_2, \dots, f_n$  are called the coordinate functions(坐标函数) of  $\mathbf{F}$ .

但是要注意,  $n$  个方程要求是相互独立的 (岂不是废话 bushi)

### 3.2.1 存在性 | Existence of Fix-Point Method

A function  $\mathbf{G}$  from  $\mathbf{D} \in \mathbb{R}^n$  into  $\mathbb{R}^n$  has a fixed point at  $\mathbf{p} \in \mathbf{D}$  if  $\mathbf{G}(\mathbf{p}) = \mathbf{p}$ .

#### Fixed Points in $\mathbb{R}^n$

A function  $\mathbf{G}$  from  $\mathbf{D} \in \mathbb{R}^n$  into  $\mathbb{R}^n$  has a **fixed point** at  $\mathbf{p} \in D$  if  $\mathbf{G}(\mathbf{p}) = \mathbf{p}$ .

#### Theorem (Existence of Fixed Points)

Let  $\mathbf{D} = \{(x_1, x_2, \dots, x_n)^t | a_i \leq x_i \leq b_i, \text{for each } i = 1, 2, \dots, n\}$  for some collection of constants  $a_1, a_2, \dots, a_n$  and  $b_1, b_2, \dots, b_n$ .

Suppose  $\mathbf{G}$  is a continuous function from  $\mathbf{D} \in \mathbb{R}^n$  into  $\mathbb{R}^n$  with the property that  $\mathbf{G}(\mathbf{x}) \in \mathbf{D}$  whenever  $\mathbf{x} \in \mathbf{D}$ . Then  $\mathbf{G}$  has a fixed point in  $\mathbf{D}$ .

#### Theorem (Fixed Point Theorem)

Let  $\mathbf{D} = \{(x_1, x_2, \dots, x_n)^t | a_i \leq x_i \leq b_i, \text{for each } i = 1, 2, \dots, n\}$  for some collection of constants  $a_1, a_2, \dots, a_n$  and  $b_1, b_2, \dots, b_n$ . Suppose  $\mathbf{G}$  is a continuous function from  $\mathbf{D} \in \mathbb{R}^n$  into  $\mathbb{R}^n$  with the property that  $\mathbf{G}(\mathbf{x}) \in \mathbf{D}$  whenever  $\mathbf{x} \in \mathbf{D}$ . Then  $\mathbf{G}$  has a fixed point in  $\mathbf{D}$ .

Moreover, suppose that all the component functions of  $\mathbf{G}$  have continuous partial derivatives and a constant  $K < 1$  exists with

$$\left| \frac{\partial g_i(\mathbf{x})}{\partial x_j} \right| \leq \frac{K}{n}, \quad \text{whenever } \mathbf{x} \in \mathbf{D},$$

for each  $j = 1, 2, \dots, n$  and each component function  $g_i$ . Then the sequence  $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$  defined by an arbitrarily selected  $\mathbf{x}^{(0)}$  in  $\mathbf{D}$  and generated by

$$\mathbf{x}^{(k)} = \mathbf{G}(\mathbf{x}^{(k-1)}), \quad \text{for each } k \geq 1$$

converges to the unique fixed point  $\mathbf{p} \in \mathbf{D}$  and

$$\|\mathbf{x}^{(k)} - \mathbf{p}\|_{\infty} \leq \frac{K^k}{1-K} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|_{\infty}$$

图 11: fix-point-define

#### 原理推导:

$\mathbf{D}$  是一个  $n$  元列向量, 那对于从  $\mathbf{D}$  到  $\mathbf{D}$  映射的函数  $\mathbf{G}$  而言, 如果存在不动

“点”  $\mathbf{p}$ , 使得

$$\mathbf{p} = \mathbf{G}(\mathbf{p}) \text{ 亦即 } p_i = g_i(\mathbf{p}), \forall i = 1, 2, 3, \dots, n$$

那么我们接下来讨论其收敛特性：类比一元函数的不动点收敛的充分条件，仍采用类似压缩映射 +MVT（拉格朗日中值定理）的思路，使用多元的 MVT 公式：

$$f(\mathbf{x}) - f(\mathbf{y}) = \nabla f(\boldsymbol{\zeta}) \cdot (\mathbf{x} - \mathbf{y})$$

将上式变形，并使用  $g_i$  来代替  $f$ ，并使用柯西不等式可以得到：

$$|g_i(\mathbf{x}) - g_i(\mathbf{y})| = \|\nabla f(\boldsymbol{\zeta}) \cdot (\mathbf{x} - \mathbf{y})\|_2 \leq \|\nabla f(\boldsymbol{\zeta})\|_2 \cdot \|\mathbf{x} - \mathbf{y}\|_2$$

此时我们将  $\mathbf{y} =$  不动点  $\mathbf{p}$  带入得到：

$$|g_i(\mathbf{x}) - g_i(\mathbf{p})| \leq \|\nabla f(\boldsymbol{\zeta})\|_2 \cdot \|\mathbf{x} - \mathbf{p}\|_2$$

由于  $g_i(\mathbf{p}) = \mathbf{p}$  进而有

$$|g_i(\mathbf{x}) - \mathbf{p}|_2 \leq \|\nabla f(\boldsymbol{\zeta})\|_2 \cdot \|\mathbf{x} - \mathbf{p}\|_2$$

即

$$\frac{\|\mathbf{G}(\mathbf{x}) - \mathbf{p}\|_2}{\|\mathbf{x} - \mathbf{p}\|_2} \leq \|\nabla f(\boldsymbol{\zeta})\|_2 \triangleq K$$

我们让  $K$  小于 1，那么就不难推出这样的迭代是收敛的（公式写的有点问题，意会一下，大致思路是这样）

### 3.3 牛顿法 | Newton's Method

$$x^{(k)} = x^{(k-1)} - \frac{1}{f'(x^{(k-1)})} f(x^{(k-1)})$$

Newton's Method for Nonlinear Systems

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \mathbf{J}(\mathbf{x}^{(k-1)})^{-1} \mathbf{F}(\mathbf{x}^{(k-1)}),$$

where  $\mathbf{J}(\mathbf{x})$  is the Jacobian matrix

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix}.$$

## 更快捷的解法（计算技巧）

- In practice, explicit computation of  $\mathbf{J}(\mathbf{x})^{-1}$  is avoided by performing the operation in a two-step manner.
  - ① A vector  $\mathbf{y}$  is found that satisfies  $\mathbf{J}(\mathbf{x}^{(k-1)})\mathbf{y} = -\mathbf{F}(\mathbf{x}^{(k-1)})$
  - ② The new approximation,  $\mathbf{x}^{(k)}$ , is obtained by adding  $\mathbf{y}$  to  $\mathbf{x}^{(k-1)}$ .

牛顿法的优点是收敛速度很快，缺点是初始值需要尽量精确

## 3.4 梯度下降算法 | Gradient Descent Techniques

其实这种方法首先多用于机器学习中进行梯度下降

用处：找到函数的最低点，有助于使用牛顿法等其他方法求根

### Gradient Descent Method

A system of nonlinear equations has the form

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0, \\ f_2(x_1, x_2, \dots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0. \end{aligned}$$

Then the following function has the minimal value of 0:

$$g(x_1, x_2, \dots, x_n) = \sum_{i=1}^n f_i^2(x_1, x_2, \dots, x_n)$$

For  $g : \mathbb{R}^n \rightarrow \mathbb{R}$ , the gradient of  $g$  at  $\mathbf{x} = (x_1, x_2, \dots, x_n)^\top$  is denoted  $\nabla g(\mathbf{x})$  and defined by

$$\nabla g(\mathbf{x}) = \left( \frac{\partial g}{\partial x_1}(\mathbf{x}), \frac{\partial g}{\partial x_2}(\mathbf{x}), \dots, \frac{\partial g}{\partial x_n}(\mathbf{x}) \right)^\top.$$

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \alpha \nabla g(\mathbf{x}^{(k-1)}),$$

where  $\alpha$  is the step size.

简化梯度求解：转化到雅各比矩阵的求解，梯度等于  $2 * \text{雅各比矩阵的转置} * \mathbf{F}(\mathbf{X})$

$$\begin{aligned}\nabla g(x_1, x_2, x_3) \equiv \nabla g(\mathbf{x}) &= \left( 2f_1(\mathbf{x}) \frac{\partial f_1}{\partial x_1}(\mathbf{x}) + 2f_2(\mathbf{x}) \frac{\partial f_2}{\partial x_1}(\mathbf{x}) + 2f_3(\mathbf{x}) \frac{\partial f_3}{\partial x_1}(\mathbf{x}), \right. \\ &\quad 2f_1(\mathbf{x}) \frac{\partial f_1}{\partial x_2}(\mathbf{x}) + 2f_2(\mathbf{x}) \frac{\partial f_2}{\partial x_2}(\mathbf{x}) + 2f_3(\mathbf{x}) \frac{\partial f_3}{\partial x_2}(\mathbf{x}), \\ &\quad \left. 2f_1(\mathbf{x}) \frac{\partial f_1}{\partial x_3}(\mathbf{x}) + 2f_2(\mathbf{x}) \frac{\partial f_2}{\partial x_3}(\mathbf{x}) + 2f_3(\mathbf{x}) \frac{\partial f_3}{\partial x_3}(\mathbf{x}) \right) \\ &= 2\mathbf{J}(\mathbf{x})^T \mathbf{F}(\mathbf{x}).\end{aligned}$$

## 4 线性系统求解 | The solution of Linear Systems

(线性方程组的寻根问题)

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = 0, \quad (1)$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = 0, \quad (2)$$

$$\vdots \quad (3)$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = 0. \quad (4)$$

which can be represented as

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

### 4.1 高斯消元法 | Gaussian elimination with backward substitution

#### 4.1.1 算法内容

解方程组  $\mathbf{A}\vec{x} = \vec{b}$ , 记  $\mathbf{A}^{(1)} = \mathbf{A} = a_{ij}^{(1)}$ ,  $\vec{b}^{(1)} = \vec{b} = \begin{pmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{pmatrix}$ ,  $\vec{x}^{(1)} = \vec{x} = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_n^{(1)} \end{pmatrix}$ , 则

增广矩阵  $\tilde{\mathbf{A}}$  为:

$$\tilde{\mathbf{A}} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{bmatrix}$$

通过高斯消元我们可以得到新的增广矩阵  $\tilde{\mathbf{A}}^{(k)}$ :

$$\tilde{\mathbf{A}}^{(k)} = \begin{bmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1n}^{(k)} & b_1^{(k)} \\ 0 & a_{22}^{(k)} & \cdots & a_{2n}^{(k)} & b_2^{(k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn}^{(k)} & b_n^{(k)} \end{bmatrix}$$

注意，对于得到的新的增广矩阵中，我们要求  $a_{ii}^k \neq 0$  (即要保证方程有唯一解)。但是在高斯消元的过程，会发现，有可能某一项是 0，怎样来避免呢？采用**主元策略**

#### 4.1.2 主元 | Pivot

- **主元**: 在一个系数矩阵 ( $\mathbf{A}$ ) 中，用于消去其他行中的相应元素的元素 ( $a_k$ ) 被称为第 ( $k$ ) 个主元。具体来说，当我们进行高斯消去法 (Gaussian elimination) 时，主元是那个用来将 ( $k+1$ )、( $k+2$ ) 到 ( $n$ ) 行的元素消去 (即让它们变成零) 的系数。

#### 4.1.3 主元策略 (Pivoting Strategy):

主元策略是指在进行高斯消去法时如何选择主元，以提高算法的稳定性和减少数值误差。常见的主元策略包括以下两种：

1. **避免主元为零**: 如果在消去过程中选到的主元 ( $a_k$ ) 为零 (或接近于零)，消去过程将无法进行 (或者可能会产生非常大的误差)。因此，在进行消去时，可以通过交换行 (或者列) 来选取一个非零的元素作为主元。这称为**部分主元法** (Partial Pivoting) 或**完全主元法** (Complete Pivoting)。
2. **减少误差**: 即使主元不为零，为了减少计算误差，我们通常希望选择一个数值较大的元素作为主元。这也是一种常见的主元策略，目的是减少由于计算中的舍入误差而引入的不准确性。

另：归一化主元策略

#### Scaled Pivoting Strategies

- The effect of scaling is to ensure that **the largest element in each row has a relative magnitude of 1** before the comparison for row interchange is performed.

具体步骤：

1. 找到每一行最大的数  $s_i$
2. 在使用主元法时比较  $a_{ik}/s_i$  和  $a_{jk}/s$  而不是  $a_{ik}$  和  $a_{jk}$
3. 找到最大的一项进行行交换

#### 4.1.4 总结

总的来说，这段话描述了在高斯消去法中如何选择主元，以及选择主元的策略。通过合理选择主元，可以避免消去失败或数值不稳定的情况，提高计算的准确性。

### Gaussian Elimination Algorithm

**INPUT** number of unknowns and equations  $n$ ; augmented matrix  $A = [a_{ij}]$ , where  $1 \leq i \leq n$  and  $1 \leq j \leq n + 1$ .

**OUTPUT** solution  $x_1, x_2, \dots, x_n$  or message that the linear system has no unique solution.

**Step 1** For  $i = 1, \dots, n - 1$  do Steps 2–4. (*Elimination process.*)

**Step 2** Let  $p$  be the smallest integer with  $i \leq p \leq n$  and  $a_{pi} \neq 0$ .  
If no integer  $p$  can be found  
then OUTPUT ('no unique solution exists');  
STOP.

**Step 3** If  $p \neq i$  then perform  $(E_p) \leftrightarrow (E_i)$ .

**Step 4** For  $j = i + 1, \dots, n$  do Steps 5 and 6.

**Step 5** Set  $m_{ji} = a_{ji}/a_{ii}$ .

**Step 6** Perform  $(E_j - m_{ji}E_i) \rightarrow (E_j)$ ;

**Step 7** If  $a_{nn} = 0$  then OUTPUT ('no unique solution exists');  
STOP.

**Step 8** Set  $x_n = a_{n,n+1}/a_{nn}$ . (*Start backward substitution.*)

**Step 9** For  $i = n - 1, \dots, 1$  set  $x_i = \left[ a_{i,n+1} - \sum_{j=i+1}^n a_{ij}x_j \right] / a_{ii}$ .

**Step 10** OUTPUT  $(x_1, \dots, x_n)$ ; (*Procedure completed successfully.*)

$O(n)$

$O(n)$

$O(n)$

$O(n^3)$

$O(n^2)$

32/1

图 12: 算法流程及时间复杂度

时间复杂度主要考虑两方面——乘法计算的次数和 loop 循环的次数，亦即说明 Gauss 消元法的时间复杂度是  $O(n^3)$

## 4.2 LU 分解 | LU Factorization

高斯消元法实际上是一种特殊的 LU 分解

### Permutation Matrices

An  $n \times n$  permutation matrix  $P = [p_{ij}]$  is a matrix obtained by rearranging the rows of  $I_n$ , the identity matrix.

### Example

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$PA = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

图 13: 置换矩阵 (注意和交换矩阵不是一个概念)

矩阵的 LU 分解实质上是高斯消元过程的结果，其中原矩阵被分解为一个下三角矩阵  $\mathbf{L}$  和一个上三角矩阵  $\mathbf{U}$  的乘积，有时还需要引入置换矩阵  $\mathbf{P}$  来反映行交换，具体求解方法为：通过高斯消元得到  $\mathbf{U}$ ，即有  $\mathbf{L} = \mathbf{A}\mathbf{U}^{-1}$

#### 4.2.1 主对角占优与严格主对角占优 | Diagonally Dominant Matrices and Strictly Diagonally Dominant

##### Definition (Diagonally Dominant Matrices)

The  $n \times n$  matrix  $A$  is said to be **diagonally dominant** when

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|$$

holds for each  $i = 1, 2, \dots, n$ .

##### Definition (Strictly Diagonally Dominant)

A diagonally dominant matrix is said to be **strictly diagonally dominant** when

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$$

holds for each  $i = 1, 2, \dots, n$ .

图 14: 主对角占优和严格主对角占优

**Theorem**

A strictly diagonally dominant matrix  $A$  is nonsingular. Moreover, in this case, Gaussian elimination can be performed on any linear system of the form  $Ax = b$  to obtain its unique solution *without row or column interchanges*, and the computations will be *stable with respect to the growth of round-off errors*.

图 15: 主对角占优矩阵在高斯消元中的性质

设矩阵  $A$  是一个  $n \times n$  的严格对角占优矩阵，即对于矩阵  $A$  的每一行  $i$ ，满足以下条件：

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|$$

我们将证明：1. 严格对角占优的矩阵是非奇异的。2. 在这种情况下进行高斯消去法时，解是唯一的且数值稳定，不需要行或列交换。

### Step1:

我们可以使用反证法来证明这一点。

假设矩阵  $A$  是奇异的，那么它存在一个非零向量  $x$ ，使得  $Ax = 0$ 。

由于  $x \neq 0$ ，我们可以设  $x$  的某一分量  $x_k$  的绝对值最大，即  $|x_k| = \max_{1 \leq i \leq n} |x_i|$ 。

根据方程  $Ax = 0$ ，我们有：

$$\sum_{j=1}^n a_{ij}x_j = 0, \quad \text{对于每一个 } i = 1, 2, \dots, n$$

特别是当  $i = k$  时，方程为：

$$a_{kk}x_k + \sum_{j \neq k} a_{kj}x_j = 0$$

取绝对值，可以得到：

$$|a_{kk}||x_k| = \left| \sum_{j \neq k} a_{kj}x_j \right| \leq \sum_{j \neq k} |a_{kj}||x_j|$$

因为  $|x_j| \leq |x_k|$  对于所有  $j \neq k$ ，可以得到：

$$|a_{kk}||x_k| \leq \sum_{j \neq k} |a_{kj}||x_k|$$

将  $|x_k|$  从不等式两边消去，得到：

$$|a_{kk}| \leq \sum_{j \neq k} |a_{kj}|$$

这与严格对角占优的定义  $|a_{kk}| > \sum_{j \neq k} |a_{kj}|$  矛盾。因此，假设  $A$  是奇异矩阵的假设不成立，说明  $A$  必须是非奇异（可逆）的。

### Step2:

在高斯消去过程中，通常需要确保主元（pivot）足够大，以避免数值不稳定性。然而，由于矩阵  $A$  是严格对角占优的，主对角线上元素  $a_{ii}$  始终满足  $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ ，这确保了每一列的主元都足够大，使得我们在消去过程中不需要进行行交换操作。同时，由于主对角线上的元素始终保持较大的值，相比其他非对角线元素，舍入误差在计算过程中不会显著放大。这确保了数值计算的稳定性，因此整个高斯消去过程可以在稳定的情况下完成。

iff 是当且仅当的意思

## 4.2.2 正定矩阵 | Positive Definite Matrices

正定矩阵首先是对称矩阵（在实数域上）

### Definition (Positive Definite)

A matrix  $A$  is **positive definite** if it is symmetric and if  $\mathbf{x}^\top A \mathbf{x} > 0$  for every  $n$ -dimensional vector  $\mathbf{x} \neq 0$ .

### Theorem

If  $A$  is an  $n \times n$  positive definite matrix, then

- ①  $A$  has an inverse;
- ②  $a_{ii} > 0$ , for each  $i = 1, 2, \dots, n$ ;
- ③  $\max_{1 \leq k, j \leq n} |a_{kj}| \leq \max_{1 \leq i \leq n} |a_{ii}|$
- ④  $(a_{ij})^2 < a_{ii}a_{jj}$ , for each  $i \neq j$ .

图 16: 正定矩阵

### Theorem

The symmetric matrix  $A$  is **positive definite** iff Gaussian elimination without row interchanges can be performed on the linear system  $A\mathbf{x} = \mathbf{b}$  with all pivot elements **positive**. Moreover, in this case, the computations are **stable** with respect to the growth of round-off errors.

### Corollary

The matrix  $A$  is **positive definite** iff  $A$  can be factored in the form  $LDL^\top$ , where  $L$  is lower triangular with 1s on its diagonal, and  $D$  is a diagonal matrix with positive diagonal entries.

### Corollary

The matrix  $A$  is **positive definite** iff  $A$  can be factored in the form  $LL^\top$ , where  $L$  is lower triangular with nonzero diagonal entries.

52 / 54

图 17: 高斯消元法中正定矩阵的性质

### Theorem

如果对称矩阵  $A$  是正定的, 那么在不进行行交换的情况下可以对线性系统  $A\mathbf{x} = \mathbf{b}$  进行高斯消去, 且所有的主元都是正的。此外, 在这种情况下, 计算对舍入误差的增长是稳定的。

**推论 | Corollary**

如果矩阵  $A$  是正定的，那么  $A$  可以分解成形式  $LDL^\top$ ，其中  $L$  是对角线为 1 的下三角矩阵， $D$  是具有正对角元的对角矩阵。

**Corollary**

如果矩阵  $A$  是正定的，那么  $A$  可以分解成形式  $LL^\top$ ，其中  $L$  是具有非零对角元的下三角矩阵。

**正定矩阵的一般性质：**

1. 正定矩阵必可逆（证明利用线性方程组有非零解进行反证）
2. 对角线上的元素都大于零
3. 每一个元素的绝对值都小于对角线上绝对值最大的元素的绝对值，即
$$\max_{1 \leq k, j \leq n} |a_{kj}| \leq \max_{1 \leq i \leq n} |a_{ii}|$$
4. 非对角线上的元素的平方小于其对应的两个对角线元素的乘积，即  $(a_{ij})^2 \leq a_{ii}a_{jj}$
5. 当且仅当对称矩阵的每一个前导主子矩阵的行列式都大于 0，其为正定矩阵
6. 当且仅当对称矩阵能在不进行行列交换下进行高斯消去，才为正定矩阵；其相对于舍入误差的增长也是稳定的
7. 正定矩阵能被表示成一个对角线为一的下三角矩阵  $L$  乘一个对角矩阵  $D$  再乘一个  $L$  的转置
8. 正定矩阵能表示成一个对角线非零的下三角矩阵  $L$  乘以其转置

关于上述第三点的证明：

For  $k \neq j$ , define  $\mathbf{x} = (x_i)$  by

$$x_i = \begin{cases} 0, & \text{if } i \neq j \text{ and } i \neq k, \\ 1, & \text{if } i = j, \\ -1, & \text{if } i = k. \end{cases}$$

Since  $\mathbf{x} \neq \mathbf{0}$ ,

$$0 < \mathbf{x}' A \mathbf{x} = a_{jj} + a_{kk} - a_{jk} - a_{kj}.$$

But  $A^T = A$ , so  $a_{jk} = a_{kj}$ , which implies that

$$2a_{kj} < a_{jj} + a_{kk}.$$

Now define  $\mathbf{z} = (z_i)$  by

$$z_i = \begin{cases} 0, & \text{if } i \neq j \text{ and } i \neq k, \\ 1, & \text{if } i = j \text{ or } i = k. \end{cases}$$

Then  $\mathbf{z}' A \mathbf{z} > 0$ , so

$$-2a_{kj} < a_{kk} + a_{jj}.$$

Equations (6.11) and (6.12) imply that for each  $k \neq j$ ,

$$|a_{kj}| < \frac{a_{kk} + a_{jj}}{2} \leq \max_{1 \leq i \leq n} |a_{ii}|, \quad \text{so} \quad \max_{1 \leq k, j \leq n} |a_{kj}| \leq \max_{1 \leq i \leq n} |a_{ii}|.$$

关于第四点的证明：

For  $i \neq j$ , define  $\mathbf{x} = (x_k)$  by

$$x_k = \begin{cases} 0, & \text{if } k \neq j \text{ and } k \neq i, \\ \alpha, & \text{if } k = i, \\ 1, & \text{if } k = j, \end{cases}$$

where  $\alpha$  represents an arbitrary real number. Because  $\mathbf{x} \neq \mathbf{0}$ ,

$$0 < \mathbf{x}' A \mathbf{x} = a_{ii}\alpha^2 + 2a_{ij}\alpha + a_{jj}.$$

As a quadratic polynomial in  $\alpha$  with no real roots, the discriminant of  $P(\alpha) = a_{ii}\alpha^2 + 2a_{ij}\alpha + a_{jj}$  must be negative. Thus

$$4a_{ij}^2 - 4a_{ii}a_{jj} < 0 \quad \text{and} \quad a_{ij}^2 < a_{ii}a_{jj}. \quad \blacksquare \blacksquare \blacksquare$$

## 5 线性系统的迭代解 | Iterative solution of Linear Systems

**定义 5.1.** 矩阵范数：对于  $n \times n$  复矩阵空间  $\mathbb{C}^{n \times n}$ ，我们也希望定义一个长度衡量矩阵的大小，定义距离比较两个矩阵之间的接近程度，由此我们引进了矩阵范数  
设  $\mathbb{C}^{n \times n} \rightarrow \mathbb{R}$  的函数  $\|\bullet\|$  满足

1. 正定性： $\forall A \in \mathbb{C}^{n \times n}, \|A\| \geq 0$  其中  $\|A\| = 0$  当且仅当  $A = 0$
2. 齐次性： $\|\alpha A\| = |\alpha| \cdot \|A\|, \forall A \in \mathbb{R}^{n \times n}, \alpha \in \mathbb{R}$
3. 相容性： $\|AB\| \leq \|A\| \|B\|, \forall A, B \in \mathbb{C}^{n \times n}, \|Ax\| \leq \|A\| \|x\|, \forall A \in \mathbb{C}^{n \times n}, x \in \mathbb{C}^n$
4. 三角不等式： $\forall A, B \in \mathbb{C}$ , 有  $\|A + B\| \leq \|A\| + \|B\|$

则称函数  $\|\bullet\|$  为  $\mathbb{C}^{n \times n}$  上的**矩阵范数**

(当然了，由于矩阵并没有长度的概念，因此将矩阵范数看作是对矩阵长度的度量并不可行)

### 范数有着不同的定义

#### Frobenius Norm

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2}$$

#### The $p$ -norms

$$\|A\|_p = \sup_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p}$$

图 18: 两种范数的定义

下面着重介绍诱导范数

## 5.1 诱导范数 | Natural or Induced Matrix norm

### 5.1.1 定义

**定义 5.2.** 矩阵的诱导范数定义如下：

$$\|A_{(m,n)}\| = \sup_{\substack{x \in \mathbb{C}^n \\ x \neq 0}} \frac{\|Ax\|_{(m)}}{\|x\|} = \sup_{\substack{x \in \mathbb{C}^n \\ \|x\|_n=1}} \|Ax\|_{(m)}.$$

通常来说我们取  $\|x\| = 1$

### 5.1.2 相关性质

**命题 5.3.** If  $\mathbf{A} = (a_{ij})$  is a  $n \times n$  matrix, then

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

设  $x = (x_1, x_2, \dots, x_n)^T \neq 0$ , 我们有

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

记

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

于是我们有

$$\|\mathbf{A}x\|_\infty = \max_{1 \leq i \leq n} \left| \sum_{j=1}^n a_{ij}x_j \right| \leq \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| |x_j| \leq \|x\|_\infty \cdot \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| = \|x\|_\infty \mu$$

则

$$\|\mathbf{A}\|_\infty = \max_{x \neq 0} \frac{\|\mathbf{A}x\|_\infty}{\|x\|_\infty} \leq \mu$$

另一个方向:

设  $\mu = \sum_{j=1}^n |a_{i_0j}| = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$ , 即第  $i_0$  列是最大的, 令

$$x_0 = (sgn(a_{i_01}), sgn(a_{i_02}), \dots, sgn(a_{i_0n}))^T$$

则  $\mathbf{A}x_0$  的第  $i_0$  个分量为  $\sum_{j=1}^n a_{i_0j} sgn(a_{i_0j}) = \sum_{j=1}^n |a_{i_0j}| = \mu$ , 且由  $x_0$  的表达式可以得出:

$$\|x_0\|_\infty = 1$$

从而有

$$\forall x \neq 0 \text{ 有 } \max_{x \neq 0} \|\mathbf{A}x\|_\infty \geq \sum_{j=1}^n a_{i_0j} sgn(a_{i_0j}) = \sum_{j=1}^n |a_{i_0j}| = \mu$$

$$\frac{\|\mathbf{A}x_0\|_\infty}{\|x_0\|_\infty} \geq \mu$$

于是夹逼得到:

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

## 5.2 特征值和特征向量 | Eigenvalues and Eigenvectors

**定义 5.4.** 谱半径: 定义为矩阵  $A$  的特征值的绝对值的最大值

$$\rho(A) = \max\{|\lambda|\}$$

**推论 5.5.** 相应的, 谱范数有如下两个性质

$$1. (\|A\|_2)^2 = \rho(A^T A)$$

$$2. \rho(A) \leq \|A\|$$

对于第一个式子,

Proof (i)

Step 1: Let  $\mu = [\rho(A^T A)]^{1/2}$ ,

$$\|Ax\|_2^2 = \mathbf{x}^T A^T A \mathbf{x} \leq \mu^2 \mathbf{x}^T \mathbf{x}$$

Thus,

$$\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} \|Ax\|_2 \leq \mu$$

Step 2: If  $\mathbf{u}$  is an eigenvector of  $A^T A$  corresponding to  $\mu^2$ , then

$$\mathbf{u}^T A^T A \mathbf{u} = \mu^2 \mathbf{u}^T \mathbf{u},$$

which shows that equality holds.

对于第二个式子: 证明很简单, 略

## 5.3 收敛矩阵 | Convergent Matrix

**定义 5.6.** We call an  $n \times n$  matrix  $A$  **convergent** if

$$\lim_{k \rightarrow \infty} (A^{(k)})_{ij} = A_0$$

**定义2:** 设  $\mathbb{C}^{n \times n}$  上的矩阵序列  $\{A_k\}$  以及矩阵  $A_0$  满足: 对给定的矩阵范数  $\|\bullet\|_v$ ,  
 $\lim_{k \rightarrow \infty} \|A_k - A_0\|_v = 0$ , 则称序列  $\{A_k\}$  依范数  $\|\bullet\|_v$  收敛于  $A_0$ , 并记为  
 $\lim_{k \rightarrow \infty} A_k = A_0$

对于依范数收敛, 我们有如下定理

**定理3:** 对于  $\mathbb{C}^{n \times n}$  上的矩阵序列  $\{A_k\}$  以及矩阵  $A_0$ , 其中  $A_k = (a_{ij})^{n \times n}$ ,  $A_0 = (a_{ij})^{n \times n}$ ,  
则下面两个命题等价

- (1)  $\{A_k\}$  依矩阵范数  $\|\bullet\|_v$  收敛于  $A_0$
- (2) 对  $\forall 1 \leq i, j \leq n$ , 有  $\lim_{k \rightarrow \infty} a_{ij}^{(k)} = a_{ij}$  (即按坐标收敛或者按元素收敛)

以后我们将矩阵序列依范数收敛简称为矩阵序列收敛

图 19: 依范数收敛

### 命题 5.7. 如下几种收敛矩阵的等价命题

The following statements are equivalent.

- (i)  $A$  is a convergent matrix.
- (ii)  $\lim_{n \rightarrow \infty} \|A^n\| = 0$ , for some natural norm.
- (iii)  $\lim_{n \rightarrow \infty} \|A^n\| = 0$ , for all natural norms.
- (iv)  $\rho(A) < 1$ .
- (v)  $\lim_{n \rightarrow \infty} A^n \mathbf{x} = \mathbf{0}$ , for every  $\mathbf{x}$ .

#### 5.3.1 The Jacobi Iterative Method

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & a_{nn} \end{bmatrix} = D - L - U$$

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

The equation

$$A\mathbf{x} = (\mathbf{D} - \mathbf{L} - \mathbf{U})\mathbf{x} = \mathbf{b}$$

is then transformed into

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

and, if  $D^{-1}$  exists, that is, if  $a_{ii} \neq 0$  for each  $i$ , then

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

This results in the matrix form of the Jacobi iterative technique:

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

For each  $k \geq 1$ , generate the components  $x_i^{(k)}$  of  $\mathbf{x}^{(k)}$  from the components of  $\mathbf{x}^{(k-1)}$  by

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

for  $i = 1, 2, \dots, n$ .

### 5.3.2 The Gauss-Seidel Method

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

区别：

对于前面已经计算过的  $x_i$ , 更加精确, 所以不再用  $x_i^{(k-1)}$  来进行后面的迭代 转化成矩阵实质是:

with the definitions of  $D$ ,  $L$ , and  $U$  given previously, we have the Gauss-Seidel method represented by

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

and

$$\mathbf{x}^{(k)} = (D - L)^{-1}U\mathbf{x}^{(k-1)} + (D - L)^{-1}\mathbf{b}, \quad \text{for each } k = 1, 2, \dots \quad (7.9)$$

## 5.4 一般迭代法与收敛性分析 | General Iteration Method

### 5.4.1 一般迭代法 | General Iteration Method

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{for each } k = 1, 2, \dots,$$

### 5.4.2 收敛性分析

引理 5.8. 关于矩阵的级数求和:

If the spectral radius satisfies  $\rho(T) < 1$ , then  $(I - T)^{-1}$  exists, and

$$(I - T)^{-1} = I + T + T^2 + \dots = \sum_{j=0}^{\infty} T^j.$$

证明:

**Proof** Because  $T\mathbf{x} = \lambda\mathbf{x}$  is true precisely when  $(I - T)\mathbf{x} = (1 - \lambda)\mathbf{x}$ , we have  $\lambda$  as an eigenvalue of  $T$  precisely when  $1 - \lambda$  is an eigenvalue of  $I - T$ . But  $|\lambda| \leq \rho(T) < 1$ , so  $\lambda = 1$  is not an eigenvalue of  $T$ , and 0 cannot be an eigenvalue of  $I - T$ . Hence,  $(I - T)^{-1}$  exists.

Let  $S_m = I + T + T^2 + \cdots + T^m$ . Then

$$(I - T)S_m = (1 + T + T^2 + \cdots + T^m) - (T + T^2 + \cdots + T^{m+1}) = I - T^{m+1},$$

and, since  $T$  is convergent, Theorem 7.17 implies that

$$\lim_{m \rightarrow \infty} (I - T)S_m = \lim_{m \rightarrow \infty} (I - T^{m+1}) = I.$$

$$\text{Thus, } (I - T)^{-1} = \lim_{m \rightarrow \infty} S_m = I + T + T^2 + \cdots = \sum_{j=0}^{\infty} T^j. \quad \blacksquare \blacksquare \blacksquare$$

那么我们回过头来看一般迭代方法的收敛性判断依据：

For any  $\mathbf{x}^{(0)} \in \mathbb{R}^n$ , the sequence  $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$  defined by

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{for each } k \geq 1,$$

converges to the unique solution of  $\mathbf{x} = T\mathbf{x} + \mathbf{c}$  if and only if  $\rho(T) < 1$ .

即

收敛的充要条件

命题 5.9. 收敛的充要条件是谱半径小于一

证明：

**Proof** First assume that  $\rho(T) < 1$ . Then,

$$\begin{aligned} \mathbf{x}^{(k)} &= T\mathbf{x}^{(k-1)} + \mathbf{c} \\ &= T(T\mathbf{x}^{(k-2)} + \mathbf{c}) + \mathbf{c} \\ &= T^2\mathbf{x}^{(k-2)} + (T + I)\mathbf{c} \\ &\vdots \\ &= T^k\mathbf{x}^{(0)} + (T^{k-1} + \cdots + T + I)\mathbf{c}. \end{aligned}$$

Because  $\rho(T) < 1$ , Theorem 7.17 implies that  $T$  is convergent, and

$$\lim_{k \rightarrow \infty} T^k \mathbf{x}^{(0)} = \mathbf{0}.$$

Lemma 7.18 implies that

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \lim_{k \rightarrow \infty} T^k \mathbf{x}^{(0)} + \left( \sum_{j=0}^{\infty} T^j \right) \mathbf{c} = \mathbf{0} + (I - T)^{-1} \mathbf{c} = (I - T)^{-1} \mathbf{c}.$$

Hence, the sequence  $\{\mathbf{x}^{(k)}\}$  converges to the vector  $\mathbf{x} \equiv (I - T)^{-1} \mathbf{c}$  and  $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ .

误差界：

**Corollary**

If  $\|T\| < 1$  for any natural matrix norm and  $\mathbf{c}$  is a given vector, then the sequence  $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$  defined by  $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$  converges, for any  $\mathbf{x}^{(0)} \in \mathbb{R}^n$ , to a vector  $\mathbf{x} \in \mathbb{R}^n$ , with  $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ , and the following error bounds hold:

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

Refer to Page 60

即对于范数小于 1 的矩阵，进行上述迭代所引入的误差界与前面第 2 章所求不动点迭代法的误差收敛性与误差界是基本一致的

**Theorem (Stein-Rosenberg)**

If  $a_{ij} \leq 0$ , for each  $i \neq j$  and  $a_{ii} > 0$ , for each  $i = 1, 2, \dots, n$ , then one and only one of the following statements holds:

- ①  $0 \leq \rho(T_g) < \rho(T_j) < 1$ ;
- ②  $1 < \rho(T_j) < \rho(T_g)$ ;
- ③  $\rho(T_j) = \rho(T_g) = 0$ ;
- ④  $\rho(T_j) = \rho(T_g) = 1$ ;

图 20: Stein-Rosenberg 定理

计算谱半径比较复杂，直接给出收敛性判定的一般结论：

1. 如果  $A$  严格对角占优：

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|, \quad i = 1, 2, \dots, n$$

则 Jacobi 迭代和 Gauss-Seidel 迭代收敛；

2. 如果  $A$  对称正定，Gauss-Seidel 迭代收敛，以及：

3. 如果  $A$  对称正定且  $(2D - A)$  对称正定，Jacobi 迭代也收敛。

直观解释：严格对角占优，对角阵就记录了  $A$  大部分信息； $A$  对称正定，Gauss-Seidel 利用下三角和对角来近似  $A$ ，实际上已经包含了  $A$  的所有信息；如果  $A$  对称正定且  $(2D - A)$  对称正定也是为了保证对角线元素比重更大。

图 21: 收敛性分析

### 5.4.3 误差界与条件数 | Condition Number

#### Residual Vector

Suppose  $\tilde{\mathbf{x}} \in \mathbb{R}^n$  is an approximation to the solution of the linear system defined by  $A\mathbf{x} = \mathbf{b}$ . The **residual vector** for  $\tilde{\mathbf{x}}$  with respect to this system is  $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$ .

#### Condition Number

The **condition number** of a nonsingular matrix  $A$  relative to a norm  $\|\cdot\|$  is

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

图 22: Condition Number

当  $K(A)$  接近 1 时为 well-conditioned, 远大于 1 时为 ill-conditioned, 同时注意到  $K(A)$  的值是恒大于 1 的

#### Conditioning

A matrix  $A$  is **well-conditioned** if  $K(A)$  is close to 1, and is **ill-conditioned** when  $K(A)$  is significantly greater than 1.

#### Remarks

- For any nonsingular matrix  $A$  and natural norm  $\|\cdot\|$ ,

$$1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = K(A)$$

- Conditioning refers to the relative security that a small residual vector implies a correspondingly accurate approximate solution.

抓住最主要的特征——> 找最大特征值

**命题 5.10.** 解和近似解的差的范数小于剩余向量的范数乘  $A$  的逆的范数

#### Theorem

Suppose that  $\tilde{\mathbf{x}}$  is an approximation to the solution of  $A\mathbf{x} = \mathbf{b}$ ,  $A$  is a nonsingular matrix, and  $\mathbf{r}$  is the residual vector for  $\tilde{\mathbf{x}}$ . Then for any natural norm,

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\mathbf{r}\| \cdot \|A^{-1}\|$$

and if  $\mathbf{x} \neq \mathbf{0}$  and  $\mathbf{b} \neq \mathbf{0}$ ,

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq K(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

证明过程很简单, 略

**Theorem 7.29** Suppose  $A$  is nonsingular and

$$\|\delta A\| < \frac{1}{\|A^{-1}\|}.$$

The solution  $\tilde{\mathbf{x}}$  to  $(A + \delta A)\tilde{\mathbf{x}} = \mathbf{b} + \delta \mathbf{b}$  approximates the solution  $\mathbf{x}$  of  $A\mathbf{x} = \mathbf{b}$  with the error estimate

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{K(A)\|A\|}{\|A\| - K(A)\|\delta A\|} \left( \frac{\|\delta \mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\delta A\|}{\|A\|} \right). \quad (7.25)$$

■

The estimate in inequality (7.25) states that if the matrix  $A$  is well-conditioned (that is,  $K(A)$  is not too large), then small changes in  $A$  and  $\mathbf{b}$  produce correspondingly small changes in the solution  $\mathbf{x}$ . If, on the other hand,  $A$  is ill-conditioned, then small changes in  $A$  and  $\mathbf{b}$  may produce large changes in  $\mathbf{x}$ .

The theorem is independent of the particular numerical procedure used to solve  $A\mathbf{x} = \mathbf{b}$ . It can be shown, by means of a backward error analysis (see [Wil1] or [Wil2]), that if Gaussian elimination with pivoting is used to solve  $A\mathbf{x} = \mathbf{b}$  in  $t$ -digit arithmetic, the numerical solution  $\tilde{\mathbf{x}}$  is the actual solution of a linear system:

$$(A + \delta A)\tilde{\mathbf{x}} = \mathbf{b}, \quad \text{where } \|\delta A\|_\infty \leq f(n)10^{1-t} \max_{i,j,k} |a_{ij}^{(k)}|.$$

for some function  $f(n)$ . Wilkinson found that in practice  $f(n) \approx n$  and, at worst,  $f(n) \leq 1.01(n^3 + 3n^2)$ .

这段话讨论了矩阵  $A$  的条件数与线性方程组  $Ax = b$  解的稳定性之间的关系。

1. **条件数的概念**: 当矩阵  $A$  是良好条件的 (即条件数  $K(A)$  较小), 小的  $A$  和  $b$  的变化会导致相应的小的解  $x$  的变化。换句话说, 解对输入的微小扰动是稳健的。
2. **不良条件的情况**: 如果矩阵  $A$  是不良条件的 (即条件数  $K(A)$  较大), 那么小的变化可能导致解  $x$  出现大的变化。这意味着解对输入的扰动非常敏感, 可能不可靠。
3. **独立性**: 这个定理与具体的数值解法无关, 适用于所有求解  $Ax = b$  的方法。
4. **回退误差分析**: 通过回退误差分析可以显示, 当使用带有主元的高斯消元法在  $t$  位数算术中求解  $Ax = b$  时, 得到的数值解  $\tilde{x}$  实际上是一个线性系统的解:

$$(A + \delta A)\tilde{x} = b$$

这里的  $\delta A$  是某种扰动。

5. **扰动的限制**: 对  $\delta A$  的无限范数 (最大元素绝对值) 有一个上界, 表示为:

$$\|\delta A\|_\infty \leq f(n)10^{1-t} \max_{i,j,k} |a_{ij}^{(k)}|$$

函数  $f(n)$  代表了问题规模  $n$  的一个界限。

6. Wilkinson 的发现: Wilkinson 发现, 在实践中,  $f(n)$  大约等于  $n$ , 并且在最坏情况下  $f(n) \leq 1.01(n^3 + 3n^2)$ , 这表明在大规模问题中, 矩阵的条件会显著影响数值解的稳定性。

综上所述, 条件数影响了解的稳定性, 尤其是在数值计算中, 良好的条件数可以确保解的准确性, 而不良条件数则可能导致解的不可靠性。

## 6 特征值估计 | Approximating Eigenvalues

在进行数值计算的过程中不难看出, 对于问题的求解, 总是会在最后通过各种近似或者降阶, 归结为矩阵代数的问题。在矩阵迭代的过程中, 对于矩阵而言, 最重要的因素之一便是矩阵的特征值, 本章就来研究矩阵的特征值或近似特征值相关性质。

### 6.1 线性代数和特征值 | Linear Algebra and Eigenvalues

#### 6.1.1 线性无关的特征向量 | Linearly Independent Eigenvectors

If  $A$  is a matrix and  $\lambda_1, \dots, \lambda_k$  are distinct eigenvalues of  $A$  with associated eigenvectors  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ , then  $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}\}$  is a linearly independent set. ■

#### 6.1.2 正交向量组和正交矩阵 | Orthogonal Vectors and Orthogonal Matrices

简单复习一下线性代数的知识 (阴沉)

**正交向量组:** 是指两两正交的向量构成的向量组合

**正交矩阵:**  $\alpha_1, \alpha_2, \dots, \alpha_n$  为  $\mathbb{R}^n$  中的一个标准正交基  $\Leftrightarrow$  这  $n$  个标准正交基构成的矩阵  $\mathbf{U}$  是  $n$  阶正交矩阵

**正交矩阵的每一行/列都是正交向量**

**正交向量组之间必定线性无关**

**正交矩阵的性质:**

Suppose that  $Q$  is an orthogonal  $n \times n$  matrix. Then

- (i)  $Q$  is invertible with  $Q^{-1} = Q^t$ ;
- (ii) For any  $\mathbf{x}$  and  $\mathbf{y}$  in  $\mathbb{R}^n$ ,  $(Q\mathbf{x})^t Q\mathbf{y} = \mathbf{x}^t \mathbf{y}$ ;
- (iii) For any  $\mathbf{x}$  in  $\mathbb{R}^n$ ,  $\|Q\mathbf{x}\|_2 = \|\mathbf{x}\|_2$ .

这三条性质说明正交变换不改变范数大小、逆矩阵为自身转置、同时可以推出正交矩阵的特征值只能取正负一

### 6.1.3 相似矩阵 | Similar Matrices

#### Definition

Two matrices  $A$  and  $B$  are said to be similar if a nonsingular matrix  $S$  exists with  

$$A = S^{-1}BS.$$

**命题 6.1.** 假设矩阵  $A$  和  $B$  是相似矩阵，即存在一个可逆矩阵  $S$  使得

$$A = S^{-1}BS$$

并且  $\lambda$  是  $A$  的特征值，对应的特征向量是  $x$ ，那么  $\lambda$  也是  $B$  的特征值，对应的特征向量是  $Sx$ 。

简而言之，相似矩阵有相同的特征值，并且  $A$  的特征向量  $x$  通过变换  $S$  可以得到  $B$  的特征向量  $Sx$ 。

### 6.1.4 对角相似矩阵 | Diagonally Similar Matrices

一个  $n \times n$  矩阵  $A$  当且仅当存在  $n$  个线性无关的特征向量（同时也是一个矩阵的特征向量都不相同，则它和对角矩阵相似）时，才与一个对角矩阵  $D$  相似。在这种情况下，我们有

$$D = S^{-1}AS,$$

其中  $S$  的列由  $A$  的特征向量组成，而  $D$  的第  $i$  个对角元素是对应于  $S$  的第  $i$  列的  $A$  的特征值。

### 6.1.5 对称矩阵 | Symmetric Matrices

**Definitions:** 一个  $n \times n$  矩阵  $A$  当且仅当存在一个对角矩阵  $D$  和一个正交矩阵  $Q$  使得

$$A = QDQ^T$$

时是对称的。

**Properties:**

- 假设  $A$  是一个对称的  $n \times n$  矩阵。 $A$  有  $n$  个特征向量组成的正交归一集合，并且  $A$  的特征值都是实数。
- 一个对称矩阵  $A$  是正定的，当且仅当  $A$  的所有特征值都为正。

## 6.2 特征值的估计——戈氏圆盘第一定理 | Gershgorin Circle

令  $\mathbf{A}$  是一个  $n \times n$  的矩阵,  $R_i$  表示复平面中的一个圆, 其圆心为  $a_{ii}$ , 半径为  $\sum_{j=1, j \neq i}^n |a_{ij}|$ ; 即

$$R_i = \left\{ z \in \mathbb{C} \mid |z - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}| \right\},$$

其中  $\mathbb{C}$  表示复平面。矩阵  $\mathbf{A}$  的特征值包含在这些圆的并集  $R = \bigcup_{i=1}^n R_i$  内。此外, 任意不与其余  $n - k$  个圆相交的  $k$  个圆的并集, 恰好包含矩阵  $\mathbf{A}$  的  $k$  个特征值 (重数计数在内)。

假设我们有一个  $5 \times 5$  矩阵, 对应 5 个 Gershgorin 圆盘。假设其中 3 个圆盘互相不相交, 且也不与其他 2 个圆盘相交。根据 Gershgorin 圆盘定理的推论, 这 3 个独立的圆盘区域中包含矩阵的 3 个特征值, 而剩下的 2 个特征值会落在其余 2 个圆盘的范围内。

## 6.3 幂方法 | The Power Method

### 概述 | Overview

CFL 与最大特征值的关系 (思考?)

考虑到一个  $n \times n$  的矩阵  $A$  的特征值最大值没有重根, 特征值按序号、模值大小排序, 亦即

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \geq 0$$

### 基本定义

设  $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ , 其特征值为  $\lambda_i$ , 对应特征向量为  $x_i$  ( $i = 1, \dots, n$ ), 即  $Ax_i = \lambda_i x_i$ , 且  $\{x_1, \dots, x_n\}$  线性无关。任取一个非零向量  $v_0 \in \mathbb{R}^n$ , 构造关于矩阵  $A$  的乘幂向量序列:

$$v_k = Av_{k-1} = A^2 v_{k-2} = \cdots = A^k v_0$$

称  $v_k$  为迭代向量。

设特征值  $\lambda_i$  的前  $r$  个为绝对值最大的特征值，即：

$$|\lambda_1| = |\lambda_2| = \cdots = |\lambda_r| > |\lambda_{r+1}| \geq \cdots \geq |\lambda_n|$$

由于  $\{x_1, \dots, x_n\}$  线性无关，构成  $\mathbb{R}^n$  的一组基，因此  $v_0$  可表示为：

$$v_0 = \sum_{i=1}^n \alpha_i x_i$$

其中，设  $\alpha_1, \dots, \alpha_r$  非全零。

由  $Ax_i = \lambda_i x_i$  可得：

$$v_k = Av_{k-1} = \cdots = A^k v_0 = \sum_{i=1}^n A^k \alpha_i x_i = \sum_{i=1}^n \lambda_i^k \alpha_i x_i = \lambda_1^k \left( \sum_{i=1}^r \alpha_i x_i + \varepsilon_k \right)$$

其中：

$$\varepsilon_k = \sum_{i=r+1}^n \left( \frac{\lambda_i}{\lambda_1} \right)^k \alpha_i x_i$$

由于  $|\lambda_1|$  最大，因此  $\left| \frac{\lambda_i}{\lambda_1} \right| < 1$  ( $i = r+1, \dots, n$ )，从而：

$$\lim_{k \rightarrow \infty} \left( \frac{\lambda_i}{\lambda_1} \right)^k = 0 \quad (i = r+1, \dots, n)$$

因此：

$$\lim_{k \rightarrow \infty} \varepsilon_k = 0$$

$$\lim_{k \rightarrow \infty} v_k = \lim_{k \rightarrow \infty} \lambda_1^k \left( \sum_{i=1}^r \alpha_i x_i + \varepsilon_k \right) = \lim_{k \rightarrow \infty} \lambda_1^k \left( \sum_{i=1}^r \alpha_i x_i \right)$$

由此可见，迭代到后期， $v_{k+1}$  和  $v_k$  的各个元素有固定比值  $\lambda_1$ ，即：

$$\lim_{k \rightarrow \infty} \frac{(v_{k+1})_i}{(v_k)_i} = \lambda_1$$

这样，收敛到主特征值后，还可计算其对应的一个特征向量：

$$\lim_{k \rightarrow \infty} \frac{v_k}{\lambda_1^k} = \sum_{i=1}^r \alpha_i x_i$$

其中，收敛速度由比值  $\left| \frac{\lambda_{r+1}}{\lambda_1} \right|$  决定，越小收敛越快。

## 改进 | Improve

在上述定义中，随着不断迭代，如果  $|\lambda_1| > 1$ ，则  $v_k$  会趋于无穷大；如果  $|\lambda_1| < 1$ ，则  $v_k$  会趋于零。在计算机中可能导致溢出或精度丢失。因此，需要在每次迭代中对向量进行规范化处理，以防止数值溢出或精度丢失。规范化的过程是将向量除以其长度，使其长度保持为 1。常用的规范化方法是使用无穷范数 ( $\|\cdot\|_\infty$ )，即向量各分量绝对值的最大值。设  $X_0 = [1, 1, \dots, 1]^\top$ ,  $Y_k = AX_k$ , 并且

$$X_{k+1} = \frac{1}{c_{k+1}} Y_k$$

其中  $c_{k+1} = \|Y_k\|_\infty$ 。则有

$$\lim_{k \rightarrow \infty} X_k = \nu^{(1)}, \quad \lim_{k \rightarrow \infty} c_k = \lambda_1.$$

### 证明

我们可以很容易地证明

$$X_k = \frac{\lambda_1^k}{c_1 c_2 \cdots c_k} \sum_{j=1}^n \beta_j \left( \frac{\lambda_j}{\lambda_1} \right)^k \nu^{(j)},$$

其中  $\beta_j$  是取决于初始向量的常数， $\nu^{(j)}$  是特征向量。

这意味着

$$\lim_{k \rightarrow \infty} X_k = \lim_{k \rightarrow \infty} \frac{\beta_1 \lambda_1^k}{c_1 c_2 \cdots c_k} \nu^{(1)}.$$

由于  $\|X_k\|_\infty = \|\nu^{(1)}\|_\infty = 1$ ，我们得到

$$\lim_{k \rightarrow \infty} \frac{\beta_1 \lambda_1^k}{c_1 c_2 \cdots c_k} = 1, \quad \text{且 } \lim_{k \rightarrow \infty} X_k = \nu^{(1)}.$$

进一步地，由

$$\frac{\beta_1 \lambda_1^k}{c_1 c_2 \cdots c_k} \approx 1,$$

和

$$\frac{\beta_1 \lambda_1^{k-1}}{c_1 c_2 \cdots c_{k-1}} \approx 1,$$

可以得到  $\lambda_1 \approx c_k$ 。

## 6.4 反幂法

### 反幂法算法

1. 任取初始向量  $v_0 = u_0 \neq 0$ ;
2. 计算  $v_k = A^{-1}u_{k-1}$  ( $k = 1, 2, \dots$ );
3. 令  $\max\{v_k\} \Rightarrow m_k$ , 并计算  $u_k = v_k/m_k$ ;
4. 如果  $k$  从某时刻以后有

$$\frac{(v_k)_j}{(v_{k-1})_j} \approx c \quad (\text{常数}) \quad (j = 1, 2, \dots, n),$$

则取  $\lambda_n \approx \frac{1}{c}$ , 而  $u_k$  就是与  $\lambda_n$  对应的特征向量。

Suppose the matrix  $A$  has eigenvalues  $\lambda_1, \dots, \lambda_n$  with linearly independent eigenvectors  $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}$ . The eigenvalues of  $(A - qI)^{-1}$ , where  $q \neq \lambda_i$ , for  $i = 1, 2, \dots, n$ , are

$$\frac{1}{\lambda_1 - q}, \quad \frac{1}{\lambda_2 - q}, \quad \dots, \quad \frac{1}{\lambda_n - q},$$

with these same eigenvectors  $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$ . (See Exercise 15 of Section 7.2.)

## 7 插值与多项式近似

对于有限元分析, 三角形网格是属于一阶精度 (三角形边界以切线贴合曲线边界), 对于矩形网格来说, 是属于 0 阶精度 (矩形边界无法很好贴合曲线边界), 并且对于三角形边界获得的雅可比矩阵是常数, 而对于四边形来说雅可比矩阵是随位置变化, 计算更加困难

### 7.1 逼近定理 | Weierstrass Approximation Theorem

Suppose that  $f$  is defined and continuous on  $[a, b]$ . For each  $\epsilon > 0$ , there exists a polynomial  $P(x)$ , with the property that

$$|f(x) - P(x)| < \epsilon, \quad \text{for all } x \text{ in } [a, b].$$

■

## 7.2 泰勒多项式 | Taylor Polynomials

### Taylor's Theorem

Suppose  $f \in C^n[a, b]$ , that  $f^{(n+1)}$  exists on  $[a, b]$ , and  $x_0 \in [a, b]$ . For every  $x \in [a, b]$ , there exists a number  $\xi(x)$  between  $x_0$  and  $x$  with

$$f(x) = P_n(x) + R_n(x),$$

where

$$\begin{aligned} P_n(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \\ R_n(x) &= \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{n+1} \end{aligned}$$

Here,  $P_n(x)$  is called the *n*th Taylor polynomial for  $f$  about  $x_0$ , and  $R_n(x)$  is called the remainder term (or truncation error) associated with  $P_n(x)$ .

## 7.3 拉格朗日插值 | Lagrange Polynomials

拉格朗日插值就是构造一个次数至多为  $n$  次的多项式使它通过  $n + 1$  个给定的点，这个多项式就是拉格朗日多项式。

构造  $P(x) = a_0 + a_1x$ , 使得  $P(x_0) = y_0$ ,  $P(x_1) = y_1$ 。

则  $P(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) = \frac{x - x_1}{x_0 - x_1}y_0 + \frac{x - x_0}{x_1 - x_0}y_1$ 。

其中  $\frac{x - x_1}{x_0 - x_1}$  和  $\frac{x - x_0}{x_1 - x_0}$  分别记作  $L_{1,0}(x)$  和  $L_{1,1}(x)$  (第一个下标即为  $n$  的值, 第二个下标为样本点的序号), 这称为拉格朗日基函数 (Lagrange Basis)。

那我们不难发现, 对于  $L_{n,i}(x_j)$  是满足  $\delta_{ij}$  函数的性质的:

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

推广到  $n$  次插值, 构造  $P(x) = a_0 + a_1x + \cdots + a_nx^n$ , 使得  $P(x_i) = y_i$ ,  $i = 0, 1, \dots, n$ 。就是要找到  $L_{n,i}(x)$  使得  $L_{n,i}(x_j) = \delta_{ij}$

分析可知, 这里的  $L_{n,i}(x)$  有  $n$  个根, 分别为  $x_0, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ 。所以可以构造出

$$L_{n,i}(x) = C(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)$$

又因为  $L_{n,i}(x_i) = 1$ , 所以

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

即

$$L_{n,i}(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

于是我们根据拉格朗日基函数构造出了  $n$  次拉格朗日插值多项式

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

### 拉格朗日插值具有唯一性

对  $n$  个不同的点,  $n$  次拉格朗日插值多项式是唯一的

证明. 如果不唯一, 假设存在另一个多项式  $Q_n(x)$ , 使得  $Q_n(x_i) = y_i$ ,  $i = 0, 1, \dots, n$ , 且  $Q_n(x) \neq P_n(x)$ 。

则  $R_n(x) = P_n(x) - Q_n(x)$  是一个次数不超过  $n$  的多项式, 且  $R_n(x_i) = 0$ ,  $i = 0, 1, \dots, n$ 。

由于  $R_n(x)$  的次数不超过  $n$ ,  $n$  次多项式不可能有  $n+1$  个解, 所以  $R_n(x) = 0$ , 即  $P_n(x) = Q_n(x)$ , 与假设矛盾。  $\square$

### 拉格朗日插值的余项

假定  $a \leq x_0 < x_1 < \dots < x_n \leq b$ ,  $f \in C[a, b]$ ,  $P_n(x)$  是  $f(x)$  在  $x_0, x_1, \dots, x_n$  上的拉格朗日插值多项式, 则对任意  $x \in [a, b]$ , 存在  $\xi(x) \in (a, b)$ , 使得

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

证明. 我们构造函数

$$h(t) = f(t) - P_n(t) + (f(x) - P_n(x)) \left( \prod_{i=0}^n \frac{t - x_i}{x - x_i} \right)$$

不难发现,  $h(t)$  有  $n+1$  个零点, 反复使用罗尔中值定义, 我们可以得到

$$h^{(n+1)}(\xi(x)) = 0$$

亦即

$$f^{(n+1)}(\xi(x)) - P^{(n+1)}(\xi(x)) - (f(x) - P(x)) \left( \prod_{i=0}^n \frac{t - x_i}{x - x_i} \right)^{(n)}|_{t=\xi(x)} = 0$$

由于  $\left( \prod_{i=0}^n \frac{t - x_i}{x - x_i} \right)^{(n)}|_{t=\xi(x)} = \frac{(n+1)!}{\prod_{i=0}^n (x - x_i)}$ , 同时有  $P^{(n+1)}(t) = 0$

接着有

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

显然，将式子整理，不难发现这个式子就是所求

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

□

## 7.4 Neville 迭代插值法 | Neville's Method

**命题 7.1.** 设  $f$  在  $x_0, x_1, \dots, x_n$  上有定义,  $m_1, m_2, \dots, m_k$  是  $k$  个不同的整数,  $0 \leq m_i \leq n$ ,  $i = 1, 2, \dots, k$ 。记在这  $k$  个点上与  $f(x)$  对应的拉格朗日多项式为  $P_{m_1, m_2, \dots, m_k}(x)$ 。

**定理 7.2.** 设  $f$  在  $x_0, x_1, \dots, x_n$  上有定义, 让  $x_i$  和  $x_j$  是这个集合中的两个不同的数。则

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

描述了对  $f$  在  $x_0, x_1, \dots, x_k$  这  $k+1$  个点上的  $k$  次插值多项式。

证明. 对于任意  $0 \leq r \leq k$ ,  $r \neq i$  和  $r \neq j$ , 分子上的两个插值多项式在  $x_r$  处都等于  $f(x_r)$ , 所以  $P(x_r) = f(x_r)$ 。

分子上的第一个多项式在  $x_i$  处等于  $f(x_i)$ , 而第二个多项式在  $x_i$  处为 0, 所以  $P(x_i) = f(x_i)$ 。同理  $P(x_j) = f(x_j)$ 。

所以  $P(x)$  在  $x_0, x_1, \dots, x_k$  上与  $f(x)$  相同, 因为  $P(x)$  是  $k$  次多项式, 所以  $P(x) = P_{0,1,\dots,k}(x)$

□

---

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

---

图 23: Method

### Comments

- This result implies that the interpolating polynomials can be generated recursively.
- For example we have

$$\begin{aligned} P_{0,1} &= \frac{1}{x_1 - x_0} [(x - x_0)P_1 - (x - x_1)P_0] \\ P_{1,2} &= \frac{1}{x_2 - x_1} [(x - x_1)P_2 - (x - x_2)P_1] \\ P_{0,1,2} &= \frac{1}{x_2 - x_0} [(x - x_0)P_{1,2} - (x - x_2)P_{0,1}] \end{aligned}$$

and so on.

图 24: example

## 7.5 牛顿差分插值多项式 | Newton's Divided Difference Interpolating Polynomial

上节讲到 Lagrange 插值时候，其实它有一个缺点，就是如果我们增删一个节点的话，我们需要更改所有的插值基函数，这样就太不方便了，所以我们提出牛顿插值。

$f(x)$  在  $x_0, x_1, \dots, x_n$  处取值为  $f(x_0), f(x_1), \dots, f(x_n)$ ,

称  $f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i}$  为  $f(x)$  在  $x_i, x_j$  处的一阶差商,

称  $f[x_i, x_j, x_k] = \frac{f[x_j, x_k] - f[x_i, x_j]}{x_k - x_i}$  为  $f(x)$  在  $x_i, x_j, x_k$  处的二阶差商,

称  $f[x_0, x_1, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0}$  为  $f(x)$  在  $x_0, x_1, \dots, x_k$  处的  $k$  阶差商.

- As might be expected from the evaluation of  $a_0$  and  $a_1$ , the required constants are

$$a_k = f[x_0, x_1, x_2, \dots, x_k],$$

for each  $k = 0, 1, \dots, n$ .

- So  $P_n(x)$  can be rewritten in a form called **Newton's Divided-Difference**:

$$P_n(x) = f[x_0] + \sum_{k=1}^n f[x_0, x_1, \dots, x_k] (x - x_0) \cdots (x - x_{k-1}).$$

## 7.6 样条插值 | Spline Interpolant

样条多项式插值是一种将插值问题分解为多个低次多项式区间段来进行逼近的方法。最常见的是三次样条插值 (Cubic Spline Interpolant)。

在每个区间  $[x_i, x_{i+1}]$  上，使用一个三次多项式  $S_i(x)$ :

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

这些多项式段需要满足以下条件:

1. 插值点处的函数值连续性条件:  $S_i(x_i) = f(x_i)$
2. 一阶导数的连续性条件:  $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$
3. 二阶导数的连续性条件:  $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1})$

这时候我们发现方程是欠定方程，即缺少条件，这时候我们补充边界条件:

4. 边界条件
  - (a) 自由边界条件  $S''_0(x_0) = 0$  和  $S''_n(x_n) = 0$
  - (b) 紧束缚边界条件  $S'_0(x_0) = f'(x_0)$  和  $S'_n(x_n) = f'(x_n)$

我们发现，总共有  $2n + 2n - 2 + 2 = 4n$  个方程，而我们刚刚好有  $n \times 4 = 4n$  个未知数

### 参数计算

记  $h_j = x_j - x_{j-1}$ , 由条件 3, 可得

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3, \quad j = 0, 1, \dots, n-1$$

又由条件 4, 因为  $S'(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$ , 所以

$$b_{j+1} = S'_{j+1}(x_{j+1}) = S'_j(x_{j+1}) = b_j + 2c_j h_j + 3d_j h_j^2, \quad j = 0, 1, \dots, n-1$$

又由条件 5, 因为  $S''(x) = 2c_j + 6d_j(x - x_j)$ , 所以

$$c_{j+1} = \frac{S''_{j+1}(x_{j+1})}{2} = c_j + 3d_j h_j, \quad j = 0, 1, \dots, n-1$$

所以

$$\begin{cases} a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\ b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 \\ c_{j+1} = c_j + 3d_j h_j \end{cases}, \quad j = 0, 1, \dots, n-1$$

把最后一个式子代入前两个式子，消去  $d_j$ ，得到

$$\begin{cases} a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \\ b_{j+1} = b_j + h_j(c_j + c_{j+1}) \end{cases}, \quad j = 0, 1, \dots, n-1$$

为了减少未知数，我们有

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \Rightarrow \begin{cases} b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) \\ b_{j-1} = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j) \end{cases}$$

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \Rightarrow b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j)$$

所以

$$\begin{cases} b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) \\ b_{j-1} = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j) \\ b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j) \end{cases}$$

$$\Rightarrow \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j) + h_{j-1}(c_{j-1} + c_j)$$

$$\Rightarrow h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_jc_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}) \quad (j = 1, 2, \dots, n-1)$$

因为  $a_j, h_j$  已知，所以上式未知量仅为  $c_j$ ，而且求出  $c_j$  后， $b_j$  也就求出了。

$$(b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}))$$

所以我们有  $n-1$  个方程， $n+1$  个未知数，所以我们需要额外的两个方程。

### 7.6.1 自由边界条件

书上给的是  $S''(a) = S''(b) = 0$ ，实际上，我们在做题中扩展到了  $S''(a) = s_0$ ， $S''(b) = s_n$ ，此时

$$c_0 = \frac{S''(a)}{2} = \frac{s_0}{2}, \quad c_n = \frac{S''(b)}{2} = \frac{s_n}{2}$$

所以，我们可以将上面的方程组写成  $\mathbf{Ax} = \mathbf{b}$  的形式，其中  $\mathbf{A}$  为  $(n+1) \times (n+1)$  的矩阵

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \cdots & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1 + h_2) & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

**b** 和 **x** 为  $(n+1) \times 1$  的向量

$$\mathbf{b} = \begin{bmatrix} \frac{s_0}{2} \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ \frac{s_n}{2} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix}$$

因为矩阵 **A** 是严格对角占优的，所以该方程组有唯一解。

### 7.6.2 固定边界

固定边界要求  $S'(a) = f'(a)$ ,  $S'(b) = f'(b)$ 。

因为  $f'(a) = S'(a) = S'(x_0) = b_0$ ,  $f'(b) = S'(b) = S'(x_n) = b_n$ , 所以

$$\begin{cases} f'(a) = b_0 = \frac{1}{h_0}(a_1 - a_0) - \frac{h_0}{3}(2c_0 + c_1) \\ f'(b) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n) = \frac{1}{h_{n-1}}(a_n - a_{n-1}) + \frac{h_{n-1}}{3}(c_{n-1} + 2c_n) \end{cases}$$

$$\Rightarrow \begin{cases} 2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{cases}$$

所以，我们可以将上面的方程组写成  $\mathbf{Ax} = \mathbf{b}$  的形式，其中 **A** 为  $(n+1) \times (n+1)$  的矩阵

$$\mathbf{A} = \begin{bmatrix} 2h_0 & h_0 & 0 & \cdots & 0 & 0 & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \cdots & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1 + h_2) & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix}$$

**b** 和 **x** 为  $(n+1) \times 1$  的向量

$$\mathbf{b} = \begin{bmatrix} \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix}$$

因为矩阵 **A** 是严格对角占优的，所以该方程组有唯一解。

### 7.6.3 应用举例

## 8 逼近论 (Approximation Theory)

逼近和插值的区别在于，插值是要求通过所有的数据点，而逼近没有这个限制，而是要求逼近的函数和原函数的误差尽可能小——尽可能接近每个点。

### 8.1 离散最小二乘逼近 | Discrete Least Squares Approximation

#### 8.1.1 误差表达

设  $p(x)$  是逼近函数， $y_i$  是给定的  $n$  个数据点，那么逼近误差的三种表达方式如下：

##### Minimax problem

$$E_\infty(p) = \max\{|y_i - f(x)|\}$$

这用初等技术是解决不了的。

##### Absolute deviation

$$E_1(p) = \sum_{i=1}^n |y_i - f(x)|$$

困难在于绝对值函数在零点不可微，可能无法求解多元函数的最小值。

##### Least squares

$$E_2(p) = \sum_{i=1}^n (y_i - f(x))^2$$

此即为**最小二乘**的误差表达，也是最常用的逼近方法。

我们的目标是找到一个  $p(x)$ ，使得  $E_2(p)$  最小。

##### 离散最小二乘逼近

定义： $P_n(x)$  是  $m$  个数据点的**离散最小二乘逼近**，如果  $P_n(x)$  是  $n$  次多项式，且满足

$$p = \arg \min_{p \in \mathbb{P}_n} \sum_{i=1}^m (y_i - p(x_i))^2$$

其中  $\mathbb{P}_n$  是  $n$  次多项式的集合,  $n$  应远远小于  $m$ , 如果  $n = m - 1$ , 其即为 Lagrange 插值。

### 离散最小二乘逼近的解

设  $P_n(x) = a_0 + a_1x + \cdots + a_nx^n = \sum_{i=0}^n a_i x^i$

$$E_2 = \sum_{i=1}^m (y_i - P_n(x_i))^2$$

为了使  $E_2$  最小, 则其必要条件是

$$\frac{\partial E_2}{\partial a_k} = 0, \quad k = 0, 1, \dots, n$$

即

$$\begin{aligned} \frac{\partial E_2}{\partial a_k} &= 2 \sum_{i=1}^m (P_n(x_i) - y_i) \frac{\partial P_n(x_i)}{\partial a_k} \\ &= 2 \sum_{i=1}^m \left( \sum_{j=0}^n a_j x_i^j - y_i \right) x_i^k \\ &= 2 \left( \sum_{j=0}^n \left( a_j \sum_{i=1}^m x_i^{j+k} \right) - \sum_{i=1}^m y_i x_i^k \right) = 0 \end{aligned}$$

即

$$\sum_{j=0}^n \left( a_j \sum_{i=1}^m x_i^{j+k} \right) = \sum_{i=1}^m y_i x_i^k, \quad k = 0, 1, \dots, n$$

也就是

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

## 8.2 对于直线拟合 7 更加紧凑的形式

### Linear Least Squares Problem

$$\{a_0^*, a_1^*\} = \underset{a_0, a_1}{\operatorname{argmin}} \sum_{i=1}^m [y_i - (a_1 x_i + a_0)]^2$$

$$\mathbf{a}^* = \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|_2^2$$

$$0 = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{a})$$

Normal equation:

$$\mathbf{X}^T \mathbf{X} \mathbf{a} = \mathbf{X}^T \mathbf{y}$$

$$\Rightarrow \mathbf{a} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

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

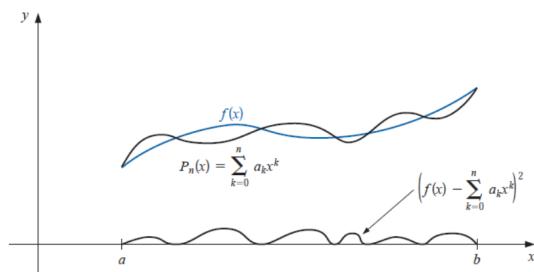
(实在是讲的太快了跟不上 www, 后面贴了不少图)

刚刚是离散化的最小二乘逼近, 现在是连续的最小二乘逼近。

### Approximation of Functions

Suppose  $f \in C[a, b]$  and that a polynomial  $P_n(x)$  of degree at most  $n$  is required that will minimize the error

$$E = \int_a^b [f(x) - P_n(x)]^2 dx = \int_a^b \left( f(x) - \sum_{k=0}^n a_k x^k \right)^2 dx$$



## Approximation of Functions

Since

$$E = \int_a^b [f(x)]^2 dx - 2 \sum_{k=0}^n a_k \int_a^b x^k f(x) dx + \int_a^b \left( \sum_{k=0}^n a_k x^k \right)^2 dx,$$

we have

$$\frac{\partial E}{\partial a_j} = -2 \int_a^b x^j f(x) dx + 2 \sum_{k=0}^n a_k \int_a^b x^{j+k} dx.$$

Hence, to find  $P_n(x)$ , the  $(n+1)$  linear **normal equations**

$$\sum_{k=0}^n a_k \int_a^b x^{j+k} dx = \int_a^b x^j f(x) dx, \text{ for each } j = 0, 1, \dots, n.$$

must be solved for the  $(n+1)$  unknowns  $a_j$ . The normal equations always have a **unique** solution provided that  $f \in C[a, b]$ .

## Disadvantages

- ① The coefficients  $a_0, a_1, \dots, a_n$  in the linear system are of the form

$$\int_a^b x^{j+k} dx = \frac{b^{j+k+1} - a^{j+k+1}}{j+k+1},$$

a linear system that does not have an easily computed numerical solution.

- ② The calculations that were performed in obtaining the best  $n$ -th-degree polynomial,  $P_n(x)$ , do not lessen the amount of work required to obtain  $P_{n+1}(x)$ .

### 8.3.1 线性无关函数/正交集 | Linearly Independent Functions

#### Definition: Linearly Independent

The set of functions  $\{\phi_0, \dots, \phi_n\}$  is said to be **linearly independent** on  $[a, b]$  if, whenever

$$c_0 \phi_0(x) + c_1 \phi_1(x) + \dots + c_n \phi_n(x) = 0, \text{ for all } x \in [a, b],$$

we have  $c_0 = c_1 = \dots = c_n = 0$ . Otherwise the set of functions is said to be **linearly dependent**.

#### Theorem

Suppose that  $\{\phi_0(x), \phi_1(x), \dots, \phi_n(x)\}$  is a collection of linearly independent polynomials in  $\prod_n$ . Then any polynomial in  $\prod_n$  can be written uniquely as a linear combination of  $\phi_0(x), \phi_1(x), \dots, \phi_n(x)$ .

**Definition: Weight function**

An integrable function  $w$  is called a **weight function** on the interval  $I$  if  $w(x) \geq 0$ , for all  $x$  in  $I$ , but  $w(x) \neq 0$  on any subinterval of  $I$ .

**Example**

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

**Definition: Orthogonal Set of Functions**

$\{\phi_0(x), \phi_1(x), \dots, \phi_n(x)\}$  is said to be an **orthogonal set of functions** for the interval  $[a, b]$  with respect to the weight function  $w$  if

$$\int_a^b w(x) \phi_k(x) \phi_j(x) dx = \begin{cases} 0, & \text{when } j \neq k, \\ \alpha_j > 0, & \text{when } j = k. \end{cases}$$

If, in addition,  $\alpha_j = 1$  for each  $j = 0, 1, \dots, n$ , the set is said to be **orthonormal**.

## 8.4 有理函数近似 | Rational Function Approximation

### 8.4.1 Pade 逼近 | Pade Approximation

Consider the difference

$$f(x) - r(x) = f(x) - \frac{p(x)}{q(x)} = \frac{f(x)q(x) - p(x)}{q(x)} = \frac{f(x) \sum_{i=0}^m q_i x^i - \sum_{i=0}^n p_i x^i}{q(x)},$$

$$(a_0 + a_1 x + \dots)(1 + q_1 x + \dots + q_m x^m) - (p_0 + p_1 x + \dots + p_n x^n), \quad (8.15)$$

通过  $f(x)$  的麦克劳林展开, 求得每一项的  $p, q$ , 使  $f(x) - r(x)$  的前  $n$  项为 0

## 9 数值微分 | Numerical Differentiation

### 9.1 一阶导数

#### 9.1.1 两点法

最简单的方法: 用两个点, 取  $h > 0$

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

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

构造由  $x_0$  和  $x_0 + h$  确定的一次 Lagrange 插值多项式:

$$\begin{aligned} f(x) &= \frac{f(x_0)(x - x_0 - h)}{x_0 - x_0 - h} + \frac{f(x_0 + h)(x - x_0)}{x_0 + h - x_0} + \frac{(x - x_0)(x - x_0 - h)}{2!} f''(\xi_x) \\ f'(x) &= \frac{f(x_0 + h) - f(x_0)}{h} + \frac{2(x - x_0) - h}{2} f''(\xi_x) + \frac{(x - x_0)(x - x_0 - h)}{2!} \frac{d}{dx} f''(\xi_x) \\ f'(x_0) &= \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''(\xi_x) \end{aligned}$$

最后我们将  $f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} - \frac{h}{2} f''(\xi_x)$  用于误差估计,  $\delta = |\frac{h}{2} f''(\xi_x)|$

### 9.1.2 一般方法

对于  $n+1$  个点, 使用拉格朗日插值

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

总体而言, 更多的评估点会产生更高的准确性。另一方面, 功能评估的数量增加, 舍入误差也会增加。因此, 数值微分是不稳定的!

### 9.1.3 对于三个点的 Lagrange polynomial

$$\begin{aligned} f'(x_j) &= f(x_0) \frac{2x_j - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{2x_j - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \\ &\quad + f(x_2) \frac{2x_j - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} + \frac{f^{(3)}(\xi_j)}{3!} \prod_{k=0, k \neq j}^2 (x_j - x_k) \end{aligned}$$

如果三个点等间距, 即  $x_0, x_1, x_2$  等距, 即  $x_1 = x_0 + h, x_2 = x_0 + 2h$ , 则

$$\begin{aligned} f'(x_j) &= f(x_0) \frac{2x_j - 2x_0 - 3h}{2h^2} + f(x_1) \frac{2x_j - 2x_0 - 2h}{-h^2} \\ &\quad + f(x_2) \frac{2x_j - 2x_0 - h}{2h^2} + \frac{f^{(3)}(\xi_j)}{3!} \prod_{k=0, k \neq j}^2 (x_j - x_k) \end{aligned}$$

由此

$$\begin{aligned} f'(x_0) &= \frac{1}{h} \left( -\frac{3}{2} f(x_0) + 2f(x_0 + h) - \frac{1}{2} f(x_0 + 2h) \right) + \frac{h^2}{3} f^{(3)}(\xi_0) \\ f'(x_1) &= \frac{1}{h} \left( -\frac{1}{2} f(x_0) + \frac{1}{2} f(x_0 + 2h) \right) - \frac{h^2}{6} f^{(3)}(\xi_1) \\ f'(x_2) &= \frac{1}{h} \left( \frac{1}{2} f(x_0) - 2f(x_0 + h) + \frac{3}{2} f(x_0 + 2h) \right) + \frac{h^2}{3} f^{(3)}(\xi_2) \end{aligned}$$

可以看出, 位于中间位置的导数, 即  $f'(x_1) = f'(x_0 + h)$  的精度是相对来说比较好的点 (而且还是只需要计算两次函数值就可以了),

$$f'(x_0) = \frac{1}{2h} (f(x_0 + h) - f(x_0 - h)) - \frac{h^2}{6} f^{(3)}(\xi)$$

### 9.1.4 其他点数的数值微分计算

#### Five-Point Formulas

##### Five-Point Endpoint Formula

$$\begin{aligned} f'(x_0) = & \frac{1}{12h} [-25f(x_0) + 48f(x_0 + h) - 36f(x_0 + 2h) \\ & + 16f(x_0 + 3h) - 3f(x_0 + 4h)] + \frac{h^4}{5} f^{(5)}(\xi) \end{aligned}$$

where  $\xi$  lies between  $x_0$  and  $x_0 + 4h$ .

##### Five-Point Midpoint Formula

$$f'(x_0) = \frac{1}{12h} [f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)] + \frac{h^4}{30} f^{(5)}(\xi)$$

where  $\xi$  lies between  $x_0 - 2h$  and  $x_0 + 2h$ .

## 9.2 高阶导数

利用 Taylor 展开，我们可以得到：

$$\begin{aligned} f(x_0 + h) &= f(x_0) + f'(x_0)h + \frac{1}{2}f''(x_0)h^2 + \frac{1}{6}f'''(x_0)h^3 + \frac{1}{24}f^{(4)}(\xi_1)h^4 \\ f(x_0 - h) &= f(x_0) - f'(x_0)h + \frac{1}{2}f''(x_0)h^2 - \frac{1}{6}f'''(x_0)h^3 + \frac{1}{24}f^{(4)}(\xi_{-1})h^4 \end{aligned}$$

两式相加，我们有

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

不妨设  $\xi_{-1} \geq \xi_1$

$$\exists \xi \in [\xi_1, \xi_{-1}], f^{(4)}(\xi) = f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})$$

于是

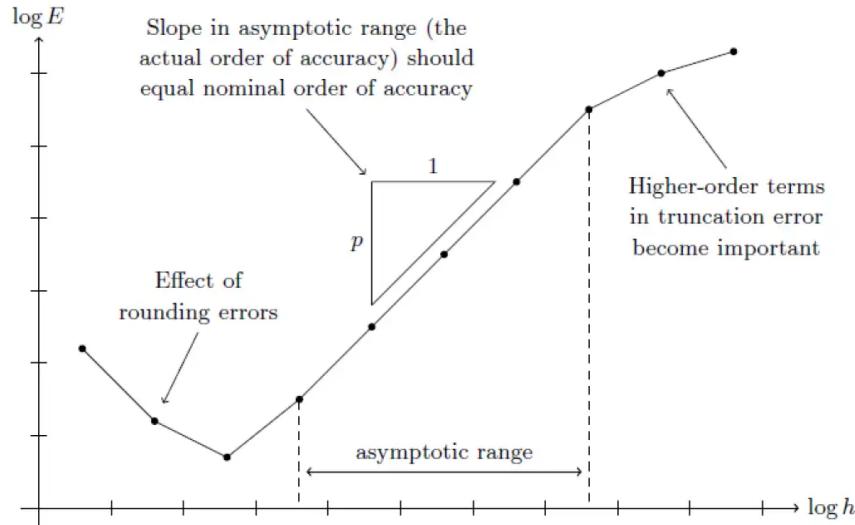
$$f''(x_0) = \frac{1}{h^2} [f(x_0 + h) + f(x_0 - h)] - \frac{h^4}{24} f^{(4)}(\xi)$$

从这里看，是不是  $h$  越小越好？注意，如果  $h$  过小，即网格选取过小，发生的问题：先不管求解效率的影响，一个数值求解的结果是否正确，要从三个角度来分析：Consistency, Stability 和 Convergence。

其中 Consistency 和 Convergence 分别指的就是当网格大小趋于 0 时，方程以及解分别趋于原方程以及解析解。Stability 是稳定性，一般 Consistency + Stability = Convergence。

在 Convergence 得到保证的情况下，误差主要有三个：truncation error (截断误差), discretization error (离散误差) 以及 rounding error (浮点计算误差)。截断误差指的是对

微分算子进行泰勒展开的误差，有限元里主要被单元阶次影响。离散误差就是数值离散解和真实解之间的误差。分析误差和网格大小的方法叫做 Grid-Refinement Study。典型 Grid-Refinement Study 的结果如下图：



其中最重要的，是 asymptotic range，在这个区域里面，误差和单元尺寸的关系是： $\text{error} = O(h^n)$ ，我们说这时候，算法是 n 次收敛的。

从图中也可以看到，如果单元尺寸太小，进入浮点误差主导的区域，那么网格尺寸越小，误差一般只会越大。

除此之外当然还要考虑 Stability，比如在动力学里面，空间离散的尺寸必须和时间离散的尺寸满足一定的关系才可以保证稳定性，这时候如果盲目减小网格尺寸，只会使得解发散。

### 9.3 理查德森外推法 | Richardson's Extrapolation

作为数值微分部分的结束，我们介绍一种通过低阶估计之间的相互运算达到更高阶截断误差的方法：Richardson 外推。

其思想非常简单：虽然我不知道截断误差的高阶项究竟是什么，但是我可以通过取不同步长进行估计来相对地消除低阶截断误差项。

考虑用  $h$  的函数估计真实值  $M$ ，现在已知  $N_1(h)$  是一个截断误差  $O(h)$  的估计，即  $M = N_1(h) + K_1 h + K_2 h^2 + \dots$

我们令  $h = \frac{h}{2}$ ，得到

$$M = N_1\left(\frac{h}{2}\right) + \frac{K_1}{2}h + \frac{K_2}{4}h^2 + \dots$$

那么我们现在就能够消去  $h$  项了：考虑

$$2N_1\left(\frac{h}{2}\right) - N_1(h) = N_2(h)$$

为我们新的对于 M 的估计，则有：

$$M - N_2(h) = -\frac{K_2}{2}h^2 + \dots$$

我们发现  $N_2(h)$  的截断误差变为了  $O(h^2)$ , 比原先更优了。详情查看

<https://zhuanlan.zhihu.com/p/49917447>

## 10 Numerical Integration (数值积分)

H 加密和 P 加密要平衡

quadrature = Numerical Integration

**数值积分的基本思想与数值微分相同，仍然是利用 Lagrange 插值公式转化为对多项式的积分。**

$$P_n(x) = \sum_{i=0}^n f(x_i)L_i(x)$$

由上插值公式，

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

误差

$$\int_a^b f(x)dx - \sum_{i=0}^n f(x_i)a_i = \int_a^b (f(x) - P_n(x))dx = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i)dx$$

### 10.0.1 精确度

求积公式的精确度 (precision/degree of accuracy) 是使得求积公式对  $x^k$  精确成立的最大正整数  $k$ 。

## Measuring Precision

### Rationale

- The standard derivation of quadrature error formulas is based on determining the class of polynomials for which these formulas produce exact results.
- The following definition is used facilitate the discussion of this derivation.

### Definition

The **degree of accuracy** or **precision**, of a quadrature formula is the largest positive integer  $n$  such that the formula is exact for  $x^k$ , for each  $k = 0, 1, \dots, n$

This implies that the Trapezoidal and Simpson's rules have degrees of precision one and three, respectively.

## 10.1 通用法则 - Newton-Cotes 求积公式

在等距节点上 ( $h = \frac{b-a}{n}$ ), 考察系数  $a_i$  的值, 我们可以得到一些通用的求积法则:

$$a_i = \int_{x_0}^{x_n} L_i(x) dx = \int_{x_0}^{x_n} \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} dx$$

令  $x = a + th$ , 则

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

### 10.1.1 梯形法则

当  $n = 1$  时:

$$\begin{aligned} a_i &= h \cdot \frac{(-1)^{1-i}}{i!(1-i)!} \cdot \int_0^1 \prod_{j=0, j \neq i}^1 (t - j) dt \\ a_0 &= h \cdot \frac{(-1)^{1-0}}{0!(1-0)!} \cdot \int_0^1 (t - 1) dt = \frac{1}{2}h \\ a_1 &= h \cdot \frac{(-1)^{1-1}}{1!(1-1)!} \cdot \int_0^1 (t - 0) dt = \frac{1}{2}h \end{aligned}$$

此时,  $n = 1$  的求积公式为

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

此即为 梯形法则 (Trapezoidal Rule)。

### 10.1.2 Simpson 法则 | Simpson's Rule

当  $n = 2$  时：

$$\begin{aligned} a_i &= h \cdot \frac{(-1)^{2-i}}{i!(2-i)!} \cdot \int_0^2 \prod_{j=0, j \neq i}^2 (t-j) dt \\ a_0 &= h \cdot \frac{(-1)^{2-0}}{0!(2-0)!} \cdot \int_0^2 (t-1)(t-2) dt = \frac{1}{3}h \\ a_1 &= h \cdot \frac{(-1)^{2-1}}{1!(2-1)!} \cdot \int_0^2 (t-0)(t-2) dt = \frac{4}{3}h \\ a_2 &= h \cdot \frac{(-1)^{2-2}}{2!(2-2)!} \cdot \int_0^2 (t-0)(t-1) dt = \frac{1}{3}h \end{aligned}$$

此时， $n = 2$  的求积公式为

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

此即为 Simpson 法则 (Simpson's Rule)。

其精确度为  $k = 3$ 。

或者使用 Taylor 公式和数值微分公式进行结合可以得到 Simpson 法则

### 10.1.3 Others

#### Closed Newton-Cotes Formulas

- $n = 1$ : Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12}f''(\xi)$$

- $n = 2$ : Simpson's rule

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90}f^{(4)}(\xi)$$

- $n = 3$ : Simpson's Three-Eighths rule

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] - \frac{3h^5}{80}f^{(4)}(\xi)$$

- $n = 4$ :

$$\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] - \frac{8h^7}{945}f^{(6)}(\xi)$$

## 10.2 复合数值积分 | Composite Numerical Integration

Newton-Cotes 以等距节点的插值多项式为基础。由于高次多项式的振荡性，这个过程在大的区间上是不精确的。为了解决这个问题，我们采用低阶 Newton-Cotes 的分段（piecewise）方法。

### 10.2.1 复合梯形法则 | Composite Trapezoidal Rule

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

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

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

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

误差项为

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

### 10.2.2 复合 Simpson 法则 | Composite Simpson's Rule

$n$  必须是偶数

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

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

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

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

Let  $f \in C^4[a, b]$ ,  $n$  be even,  $h = (b - a)/n$ , and  $x_j = a + jh$ , for each  $j = 0, 1, \dots, n$ . There exists a  $\mu \in (a, b)$  for which the **Composite Simpson's rule** for  $n$  subintervals can be written with its error term as

$$\int_a^b f(x) dx = \frac{h}{3} \left[ f(a) + 2 \sum_{j=1}^{(n/2)-1} f(x_{2j}) + 4 \sum_{j=1}^{n/2} f(x_{2j-1}) + f(b) \right] - \frac{b-a}{180} h^4 f^{(4)}(\mu).$$

■

### 10.3 Romberg Integration

To approximate the integral  $\int_a^b f(x) dx$  we use the results of the Composite Trapezoidal rule with  $n = 1, 2, 4, 8, 16, \dots$ , and denote the resulting approximations, respectively, by  $R_{1,1}, R_{2,1}, R_{3,1}$ , etc. We then apply extrapolation in the manner given in Section 4.2, that is, we obtain  $O(h^4)$  approximations  $R_{2,2}, R_{3,2}, R_{4,2}$ , etc., by

$$R_{k,2} = R_{k,1} + \frac{1}{3}(R_{k,1} - R_{k-1,1}), \quad \text{for } k = 2, 3, \dots$$

Then  $O(h^6)$  approximations  $R_{3,3}, R_{4,3}, R_{5,3}$ , etc., by

$$R_{k,3} = R_{k,2} + \frac{1}{15}(R_{k,2} - R_{k-1,2}), \quad \text{for } k = 3, 4, \dots$$

In general, after the appropriate  $R_{k,j-1}$  approximations have been obtained, we determine the  $O(h^{2j})$  approximations from

$$R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1} - 1}(R_{k,j-1} - R_{k-1,j-1}), \quad \text{for } k = j, j+1, \dots$$

$k$	$O(h_k^2)$	$O(h_k^4)$	$O(h_k^6)$	$O(h_k^8)$	$O(h_k^{2n})$
1	$R_{1,1}$				
2	$R_{2,1}$	$R_{2,2}$			
3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$		
4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
$n$	$R_{n,1}$	$R_{n,2}$	$R_{n,3}$	$R_{n,4}$	$\cdots$
					$R_{n,n}$

# 11 常微分方程数值解 | Solution of Differential Equations

## 11.1 Initial-Value Problems for ODEs | 初始值问题

微分方程的初值问题即，对于这个方程给定初始值  $y(a) = \alpha$  的求解：

$$\frac{dy}{dt} = f(t, y)$$

### 11.1.1 李普希斯条件 | Lipschitz Condition

#### Definition: Lipschitz Condition

A function  $f(t, y)$  is said to satisfy a **Lipschitz condition** in the variable  $y$  on a set  $D \subset \mathbb{R}^2$  if a constant  $L > 0$  exists with

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|$$

whenever  $(t, y_1)$  and  $(t, y_2)$  are in  $D$ . The constant  $L$  is called a **Lipschitz constant** for  $f$ .

实际上  $L$  相当于是

$$\max\left\{\frac{\partial f(t, y)}{\partial y}\right\}$$

即偏导数存在上界（不严谨，详见 11.1.2）

在 Lipschitz 条件下，初值问题的解是唯一的

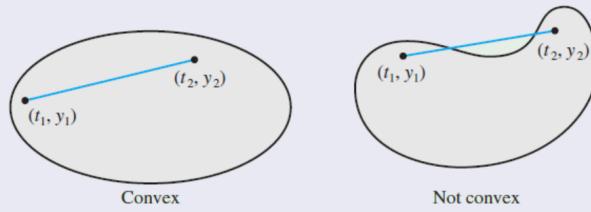
### 11.1.2 凸集 | Convex Set

#### Convex Set

A set  $D \subset \mathbb{R}^2$  is said to be convex if whenever  $(t_1, y_1)$  and  $(t_2, y_2)$  belong to  $D$ , then

$$((1 - \lambda)t_1 + \lambda t_2, (1 - \lambda)y_1 + \lambda y_2)$$

also belongs to  $D$  for every  $\lambda$  in  $[0, 1]$ .



### Theorem: Sufficient Conditions

Suppose  $f(t, y)$  is defined on a convex set  $D \subset \mathbb{R}^2$ . If a constant  $L > 0$  exists with

$$\left| \frac{\partial f}{\partial y}(t, y) \right| \leq L, \text{ for all } (t, y) \in D$$

then  $f$  satisfies a Lipschitz condition on  $D$  in the variable  $y$  with Lipschitz constant  $L$ .

### 11.1.3 解的存在性

#### Theorem: Existence & Uniqueness

Suppose that  $D = \{(t, y) | a \leq t \leq b \text{ and } -\infty < y < \infty\}$  and that  $f(t, y)$  is continuous on  $D$ . If  $f$  satisfies a Lipschitz condition on  $D$  in the variable  $y$ , then the initial-value problem

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

has a unique solution  $y(t)$  for  $a \leq t \leq b$ .

### 11.1.4 适定性 | well-posed

性质：存在唯一且连续

#### Definition: Well-Posed Problem (cont'd)

- ① A unique solution,  $y(t)$ , to the problem exists, and
- ② There exist constants  $\epsilon_0 > 0$  and  $k > 0$  such that for any  $\epsilon$ , with  $\epsilon_0 > \epsilon > 0$ , whenever  $\delta(t)$  is continuous with  $|\delta(t)| < \epsilon$  for all  $t$  in  $[a, b]$ , and when  $|\delta_0| < \epsilon$ , the initial-value problem

$$\frac{dz}{dt} = f(t, z) + \delta(t), \quad a \leq t \leq b, \quad z(a) = \alpha + \delta_0,$$

has a unique solution  $z(t)$  that satisfies

$$|z(t) - y(t)| < k\epsilon \text{ for all } t \text{ in } [a, b].$$

## 11.2 欧拉法 | Euler's Method

欧拉法的核心思想是用  $f(t, y)$  在  $t_i, y_i$  处的线性近似值来代替  $f(t, y)$ , 即:

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

我们称其为差分方程 (difference equation)

### 11.2.1 误差分析 | Error Bounds for Euler's Method

#### Error Bounds for Euler's Method–Theorem

Suppose  $f$  is continuous and satisfies a Lipschitz condition on

$$D = \{(t, y) | a \leq t \leq b \text{ and } -\infty < y < \infty\}$$

and that a constant  $M$  exists with

$$|y''(t)| \leq M, \quad \text{for all } t \in [a, b]$$

where  $y(t)$  denotes the unique solution to the initial-value problem

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

Let  $w_0, w_1, \dots, w_N$  be the approximations generated by Euler's method for some positive integer  $N$ . Then, for each  $i = 0, 1, 2, \dots, N$

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

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + hf(t_i, w_i), \quad \text{for each } i = 0, 1, \dots, N-1 \end{aligned}$$

#### Round-off Error

$$\begin{aligned} u_0 &= \alpha + \delta_0 \\ u_{i+1} &= u_i + hf(t_i, u_i) + \delta_{i+1}, \quad \text{for each } i = 0, 1, \dots, N-1 \end{aligned}$$

#### The Error Bound with Round-off Error

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

The error bound is no longer linear and it approaches infinity when  $h \approx 0$ . In fact, the optimal  $h$  can be achieved when

$$h = \sqrt{\frac{2\delta}{M}}.$$

这说明,  $h$  的最优值为  $\sqrt{\frac{2\delta}{M}}$ , 不能再小了

### 11.2.2 截断误差 | Local Truncation Error

#### Local Truncation Error

IVP

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

Definition of LTE

The difference method

$$w_0 = \alpha$$

$$w_{i+1} = w_i + h\phi(t_i, w_i), \quad \text{for each } i = 0, 1, \dots, N-1,$$

has local truncation error

$$\tau_{i+1}(h) = \frac{y_{i+1} - (y_i + h\phi(t_i, y_i))}{h} = \frac{y_{i+1} - y_i}{h} - \phi(t_i, y_i),$$

for each  $i = 0, 1, \dots, N-1$ , where  $y_i$  and  $y_{i+1}$  denote the solution at  $t_i$  and  $t_{i+1}$ , respectively.

38/68

局部截断误差只考虑一步的误差，即假设前面没有误差：

$$w_0 = \alpha$$

$$w_{i+1} = w_i + h\phi(t_i, w_i), \quad i = 0, 1, \dots, N-1$$

有局部截断误差

$$\tau_{i+1}(h) = \frac{y_{i+1} - (y_i + h\phi(t_i, y_i))}{h} = \frac{y_{i+1} - y_i}{h} - \phi(t_i, y_i)$$

The local truncation error of Euler's method

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

**Method of order 1**

### 11.3 高阶 Taylor 法 | Higher-Order Taylor Methods

Euler 法实际上就是高阶 Taylor 法的一阶近似。高阶 Taylor 法

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

$n$  阶的 Taylor 法:

$$w_0 = \alpha$$

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

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

其局部截断误差为  $O(h^n)$  (如果  $y \in C^{n+1}[a, b]$ )。

## 11.4 Runge-Kutta 法 | Runge-Kutta Methods

泰勒方法需要计算  $f(t, y)$  的导数并求值, 这是一个复杂耗时的过程。Runge-Kutta 方法具有 Taylor 方法的高阶局部截断误差, 但是不需要计算  $f(t, y)$  的导数。

Runge-Kutta 法: 对  $f(t, y)$  进行 Taylor 展开——另一种思路

### Basic Structure of RK2 Methods

Our starting point is to assume that the numerical method has the following structure:

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + h \cdot \color{red}a_1\color{black} f(t_i + \color{blue}\alpha_1\color{black}, w_i + \color{purple}\beta_1\color{black}) \end{aligned}$$

for  $i = 0, 1, \dots, N - 1$ , where  $\color{red}a_1$ ,  $\color{blue}\alpha_1$ , and  $\color{purple}\beta_1$  are parameters to be determined to ensure a local truncation error of  $O(h^2)$ .

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

## 12 微分方程组解法 | Solution of Differential Equations

## 13 示例

这是一个带有浅灰色底纹的文字块，适合用于展示普通文本的填充背景效果。

这是一个带有浅绿色底纹的文字块，用于展示文本的填充背景效果。

这是一个带有浅红色底纹的文字块，用于展示另一种填充背景效果。

这是一个带有浅蓝色底纹的文字块，适合用于区分文本段落。

**引理 13.1.**

**命题 13.2.** *adf*

**定义 13.3.** *adsf*

**例 13.4.** *asdfasfd*

在一个线性电路中，电流的积分与电压的积分之和等于零。

asdfas[1]

Code Listing 1: Python example

```
1 def hello_world():
2     print("Hello, world!")
```

## 参考文献

- [1] John Smith and Jane Doe. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2019:1–13, 2019.