

第六讲 线性方程组基本迭代法

- 1 矩阵分裂与迭代法
- 2 收敛性分析
- 3 应用: Poisson 方程求解
- 4 加速方法
- 5 交替方向法与 HSS 算法

线性方程组的数值求解

➤ **直接法** PLU 分解, LDL^T 分解, Cholesky 分解等

➤ **迭代法**

- 基本迭代法 (经典迭代法): Jacobi, Gauss-Seidel, SOR, SSOR, ...
- 现代迭代法 (Krylov 子空间迭代法): **CG**, MINRES, **GMRES**, BiCGStab, ...

➤ **快速算法** (基于特殊结构和性质)

- 基于各类快速变换, 如 FFT, DCT, DST 等
- 代数多重网格法 (Algebraic multigrid)
- 快速多极子算法 (Fast multipole), Hierarchical Matrices

注记

- ▶ 有些方法可能只是对某些特定方程有效, 比如各类快速算法.
- ▶ 在实际应用中, 这些方法经常结合使用, 如混合算法, 预处理算法等.

本讲主要介绍经典 (定常, 不动点) 迭代法



- ▶ [Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods](#), SIAM, 1994.
- ▶ Saad and van der Vorst, [Iterative solution of linear systems in the 20th century](#), 2000.


1 | 基本迭代法

为什么迭代法

当直接求解方程组 $Ax = b$ 较困难时, 我们可以求解一个近似方程组

$$Mx = b$$

其中 M 是 A 的某个近似, 且该方程组较容易求解.

 设近似方程组的解为 $x^{(1)}$, 易知它与真解之间的误差满足

$$A(x_* - x^{(1)}) = b - Ax^{(1)}$$

如果 $x^{(1)}$ 已经满足精度要求, 则停止计算, 否则需要修正.

近似解的修正

设修正量为 Δx . 显然, 最优的 Δx 应该满足 $A(x^{(1)} + \Delta x) = b$, 即

$$A\Delta x = b - Ax^{(1)}.$$

但由于直接求解该方程比较困难, 因此我们还是求解近似

$$M\Delta x = b - Ax^{(1)}.$$

于是得到修正后的近似解

$$x^{(2)} = x^{(1)} + \Delta x = x^{(1)} + M^{-1}(b - Ax^{(1)})$$



若 $x^{(2)}$ 已经满足精度要求, 则停止计算, 否则继续按以上的方式进行修正.

不断重复以上步骤, 于是, 我们就得到一个序列

$$x^{(1)}, x^{(2)}, \dots, x^{(k)}, \dots$$


满足以下递推关系

$$x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)}), \quad k = 1, 2, \dots$$

这就构成了一个迭代格式.

迭代法需要考虑的两个基本要点

- (1) 以 M 为系数矩阵的线性方程组必须要比原线性方程组更容易求解;
- (2) M 应该是 A 的一个很好的近似, 或者迭代序列 $\{x^{(k)}\}$ 要收敛.

 本小节我们介绍几个常见的基于 **矩阵分裂** 的基本迭代法:

- Jacobi 迭代法
- Gauss-Seidel 迭代法
- SOR (Successive Over-Relaxation) 迭代法
- SSOR (Symmetric SOR) 迭代法
- AOR (Accelerated over-relaxation) 迭代法
- Richardson 迭代法
- 分块迭代法

2 | 矩阵分裂与迭代法

- 2.1 Jacobi 迭代法
- 2.2 Gauss-Seidel 迭代法
- 2.3 SOR 迭代法
- 2.4 SSOR 迭代法
- 2.5 AOR 迭代法 *
- 2.6 Richardson 迭代法
- 2.7 分块迭代法

迭代法

给定一个迭代初始值 $x^{(0)}$, 通过一定的迭代格式生成一个迭代序列

$$x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(k)}, \dots$$

使得

$$\lim_{k \rightarrow \infty} x^{(k)} = x_* \triangleq A^{-1}b$$

矩阵分裂

定义 (矩阵分裂 Matrix splitting) 设 $A \in \mathbb{R}^{n \times n}$ 非奇异, 称

$$A = M - N \quad (6.1)$$

为 A 的一个矩阵分裂, 其中 M 非奇异.

原方程组等价于 $Mx = Nx + b$. 于是我们就可以构造迭代格式

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \triangleq Gx^{(k)} + g, \quad k = 0, 1, \dots, \quad (6.2)$$

其中 $G = M^{-1}N$ 称为该迭代格式的**迭代矩阵**.

2.1 Jacobi 迭代法

将矩阵 A 分裂为

$$A = D - L - U,$$


其中 D 为 A 的对角线部分, $-L$ 和 $-U$ 分别为严格下三角和严格上三角部分

Jacobi 迭代法: $M = D, N = L + U$

$$x^{(k+1)} = D^{-1}(L + U)x^{(k)} + D^{-1}b, \quad k = 0, 1, 2, \dots \quad (6.3)$$

迭代矩阵

$$G_J = D^{-1}(L + U)$$

 写成分量形式:

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



由于 Jacobi 迭代法中 $x_i^{(k+1)}$ 的更新顺序与 i 无关, 即可以按顺序 $i = 1, 2, \dots, n$ 计算, 也可以按顺序 $i = n, n-1, \dots, 2, 1$ 计算, 或者乱序计算. 因此 Jacobi 迭代法非常适合并行计算.

算法 1 求解线性方程组的 Jacobi 迭代法

- 1: Choose an initial guess $x^{(0)}$
- 2: **while** not converge **do**
- 3: **for** $i = 1$ to n **do**
- 4: $x_i^{(k+1)} = \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right) / a_{ii}$
- 5: **end for**
- 6: **end while**

我们也可以将 Jacobi 迭代格式写为

$$x^{(k+1)} = x^{(k)} + D^{-1}(b - Ax^{(k)}) = x^{(k)} + D^{-1}r_k, \quad k = 0, 1, \dots,$$

其中 $r_k \triangleq b - Ax^{(k)}$ 是 k 次迭代后的残量。

2.2 Gauss-Seidel 迭代法

Gauss-Seidel (G-S) 迭代法: $M = D - L, N = U$

$$x^{(k+1)} = (D - L)^{-1}Ux^{(k)} + (D - L)^{-1}b, \quad k = 0, 1, 2, \dots \quad (6.4)$$

迭代矩阵

$$G_{GS} = (D - L)^{-1}U$$

将 G-S 迭代法改写为

$$Dx^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + b,$$

即可得分量形式

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right), \quad i = 0, 1, \dots, n.$$

算法 2 求解线性方程组的 G-S 迭代法

1: Choose an initial guess $x^{(0)}$

2: **while** not converge **do**

3: **for** $i = 1$ to n **do**

$$4: \quad x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right)$$

5: **end for**

6: **end while**



▣ G-S 迭代法的主要优点是充分利用了已经获得的最新数据

▣ 但 G-S 迭代法中未知量的更新是按自然顺序进行的, 不适合并行计算

2.3 SOR 迭代法

在 G-S 迭代法的基础上, 我们可以通过引入一个松弛参数 ω 来加快收敛速度.

这就是 SOR (Successive Overrelaxation) 迭代法:

$$x^{(k+1)} = (1 - \omega)x^{(k)} + \omega \left(D^{-1}(Lx^{(k+1)} + Ux^{(k)}) + D^{-1}b \right).$$

整理后即为

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

其中 ω 称为 **松弛参数**.

注记

- (1) 当 $\omega = 1$ 时, SOR 即为 G-S 迭代法,
- (2) 当 $\omega < 1$ 时, 称为 **低松弛 (under relaxation)** 迭代法,
- (3) 当 $\omega > 1$ 时, 称为 **超松弛 (over relaxation)** 迭代法.



- ▣ SOR 迭代法曾经在很长一段时间内是科学计算中求解大规模线性方程组的首选方法.
- ▣ 在大多数情况下, 当 $\omega > 1$ 时会取得比较好的收敛效果.

SOR 的迭代矩阵为

$$G_{\text{SOR}} = (D - \omega L)^{-1} ((1 - \omega)D + \omega U),$$

对应的矩阵分裂为

$$M = \frac{1}{\omega}D - L, \quad N = \frac{1 - \omega}{\omega}D + U.$$

SOR 迭代法的分量形式为

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

算法 3 求解线性方程组的 SOR 迭代法

1: Choose an initial guess $x^{(0)}$ and parameter ω

2: **while** not converge **do**

3: **for** $i = 1$ to n **do**

4:
$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right)$$

5: **end for**

6: **end while**



- SOR 最大的优点是引入了 ω , 通过选取适当的 ω 可以大幅提高算法的收敛速度.
- 但是 SOR 最大的难点就是如何选取最优的参数.

2.4 SSOR 迭代法

将 SOR 迭代法中的 L 和 U 相交换, 即可得迭代格式

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

将这个迭代格式与 SOR 相结合, 就可以得到下面的两步迭代法

$$\begin{cases} x^{(k+\frac{1}{2})} = (D - \omega L)^{-1} [(1 - \omega)D + \omega U] x^{(k)} + \omega(D - \omega L)^{-1}b \\ x^{(k+1)} = (D - \omega U)^{-1} [(1 - \omega)D + \omega L] x^{(k+\frac{1}{2})} + \omega(D - \omega U)^{-1}b \end{cases}$$

这就是 **SSOR 迭代法** (对称超松弛迭代法).



若 $\omega = 1$, 则得到 **SGS 迭代法**.

消去中间迭代向量 $x^{(k+\frac{1}{2})}$, 可得


$$x^{(k+1)} = G_{\text{SSOR}}x^{(k)} + g,$$

其中迭代矩阵

$$G_{\text{SSOR}} = (D - \omega U)^{-1}[(1 - \omega)D + \omega L](D - \omega L)^{-1}[(1 - \omega)D + \omega U].$$

对应的矩阵分裂为

$$\begin{aligned} M &= \frac{1}{\omega(2 - \omega)} [D - \omega(L + U) + \omega^2 LD^{-1}U] \\ &= \frac{1}{\omega(2 - \omega)} (D - \omega L)D^{-1}(D - \omega U), \\ N &= \frac{1}{\omega(2 - \omega)} [(1 - \omega)D + \omega L]D^{-1}[(1 - \omega)D + \omega U]. \end{aligned}$$

 对某些问题, SOR 不收敛, 但仍可能构造出收敛的 SSOR 迭代法.

2.5 AOR 迭代法 *

1978 年, Hadjidimos 提出了 AOR (Accelerated over-relaxation) 迭代法, 迭代矩阵为

$$G_{\text{AOR}} = (D - \gamma L)^{-1} [(1 - \omega)D + (\omega - \gamma)L + \omega U],$$

其中 γ 和 ω 为松弛参数. 对应的矩阵分裂为

$$M = \frac{1}{\omega}(D - \gamma L), \quad N = \frac{1}{\omega}[(1 - \omega)D + (\omega - \gamma)L + \omega U].$$

- (1) 当 $\gamma = \omega$ 时, AOR 即为 SOR 迭代法;
- (2) 当 $\gamma = \omega = 1$ 时, AOR 即为 G-S 迭代法;
- (3) 当 $\gamma = 0, \omega = 1$ 时, AOR 即为 Jacobi 迭代法.

与 SSOR 类似, 我们也可以定义 SAOR 迭代法.

2.6 Richardson 迭代法

Richardson 迭代法是一类形式非常简单的算法, 其迭代格式为

$$x^{(k+1)} = x^{(k)} + \omega(b - Ax^{(k)}), \quad k = 0, 1, 2, \dots$$

其迭代矩阵为

$$G_R = I - \omega A,$$

对应的矩阵分裂为

$$M = \frac{1}{\omega}I, \quad N = \frac{1}{\omega}I - A.$$

2.7 分块迭代法

将 A 写成如下的分块形式:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix}.$$

A_{11}					
	A_{22}				
					A_{pp}



分块迭代法采用更多的 3 级 BLAS 运算, 因此更有利于发挥现代计算机的性能.

分块 Jacobi 迭代法

$$A_{ii}\mathbf{x}_i^{(k+1)} = \mathbf{b}_i - \sum_{j=1, j \neq i}^p A_{ij}\mathbf{x}_j^{(k)}, \quad i = 1, 2, \dots, p.$$

分块 Gauss-seidel 迭代法

$$A_{ii}\mathbf{x}_i^{(k+1)} = \mathbf{b}_i - \sum_{j=1}^{i-1} A_{ij}\mathbf{x}_j^{(k+1)} - \sum_{j=i+1}^p A_{ij}\mathbf{x}_j^{(k)}, \quad i = 1, 2, \dots, p.$$

分块 SOR 迭代法

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + \omega A_{ii}^{-1} \left(\mathbf{b}_i - \sum_{j=1}^i A_{ij}\mathbf{x}_j^{(k+1)} - \sum_{j=i+1}^p A_{ij}\mathbf{x}_j^{(k)} \right),$$
$$i = 1, 2, \dots, p.$$

3 | 收敛性分析

3.1 基本概念

3.2 不可约对角占优矩阵

3.3 对称正定矩阵情形

3.4 相容次序矩阵

3.1 基本概念

向量序列的收敛

定义 (向量序列的收敛) 设 $\{x^{(k)}\}_{k=1}^{\infty}$ 是 \mathbb{R}^n (或 \mathbb{C}^n) 中的一个向量序列. 如果存在向量 $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ (或 \mathbb{C}^n) 使得

$$\lim_{k \rightarrow \infty} x_i^{(k)} = x_i, \quad i = 1, 2, \dots, n,$$

其中 $x_i^{(k)}$ 表示 $x^{(k)}$ 的第 i 个分量, 则称 $\{x^{(k)}\}$ (按分量) 收敛到 x , 即 x 是 $x^{(k)}$ 的极限, 记为

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

矩阵序列的收敛

定义 (矩阵序列的收敛) 设 $\{A^{(k)} = [a_{ij}^{(k)}]\}_{k=0}^{\infty}$ 是 $\mathbb{R}^{m \times n}$ (或 $\mathbb{C}^{m \times n}$) 中的一个矩阵序列. 如果存在矩阵 $A = [a_{ij}] \in \mathbb{R}^{m \times n}$ (或 $\mathbb{C}^{m \times n}$) 使得

$$\lim_{k \rightarrow \infty} a_{ij}^{(k)} = a_{ij}, \quad i = 1, 2, \dots, m, j = 1, 2, \dots, n,$$

则称 $A^{(k)}$ 收敛到 A , 即 A 是 $A^{(k)}$ 的极限, 记为

$$\lim_{k \rightarrow \infty} A^{(k)} = A.$$

收敛性基本判别方法

定理 设向量序列 $\{x^{(k)}\}_{k=0}^{\infty} \subset \mathbb{R}^n$ (或 \mathbb{C}^n), 矩阵序列 $\left\{A^{(k)} = \begin{bmatrix} a_{ij}^{(k)} \end{bmatrix}\right\}_{k=0}^{\infty} \subset \mathbb{R}^{m \times n}$ (或 $\mathbb{C}^{m \times n}$), 则

- (1) $\lim_{k \rightarrow \infty} x^{(k)} = x \iff \lim_{k \rightarrow \infty} \|x^{(k)} - x\| = 0$, 其中 $\|\cdot\|$ 为任一向量范数;
- (2) $\lim_{k \rightarrow \infty} A^{(k)} = A \iff \lim_{k \rightarrow \infty} \|A^{(k)} - A\| = 0$, 其中 $\|\cdot\|$ 为任一矩阵范数;
- (3) $\lim_{k \rightarrow \infty} A^{(k)} = 0 \iff \lim_{k \rightarrow \infty} A^{(k)}x = 0, \forall x \in \mathbb{R}^n$ (或 \mathbb{C}^n).

谱半径与范数的关系

- (1) 对任意矩阵范数, 有 $\rho(G) \leq \|G\|$;
- (2) 对任意 $\varepsilon > 0$, 都存在一个矩阵范数 $\|\cdot\|_\varepsilon$, 使得 $\|G\|_\varepsilon \leq \rho(G) + \varepsilon$.

定理 设矩阵 $A \in \mathbb{R}^{n \times n}$ (或 $\mathbb{C}^{n \times n}$), 则 $\lim_{k \rightarrow \infty} A^k = 0$ 当且仅当 $\rho(A) < 1$.

(板书)

引理 设 $G \in \mathbb{R}^{n \times n}$, 则对任意矩阵范数 $\|\cdot\|$, 有

$$\rho(G) = \lim_{k \rightarrow \infty} \|G^k\|^{\frac{1}{k}}.$$

(板书)

迭代法的收敛性

定义 (迭代法的收敛性) 考虑基本迭代法

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

如果对 **任意的初始向量** $x^{(0)}$, 都有

$$\lim_{k \rightarrow \infty} x^{(k)} \rightarrow x_*,$$

则称该迭代法是 **收敛** 的, 否则就称其为 **发散** 的.

收敛充分条件

记 $e^{(k)} \triangleq x^{(k)} - x_*$ 为第 k 步迭代解 $x^{(k)}$ 的误差向量.

引理 若存在算子范数 $\|\cdot\|$, 使得 $\|G\| < 1$, 则下面的迭代格式收敛:

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

(板书)



事实上, 引理中的算子范数也可以是一般的矩阵范数.

基本收敛定理

定理 (收敛性定理) 对任意迭代初始向量 $x^{(0)}$, 迭代格式

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

收敛的充要条件是 $\rho(G) < 1$.

(板书)

收敛速度

定义 设 G 是迭代矩阵, 则迭代法 $x^{(k+1)} = Gx^{(k)} + g$ 的**平均收敛速度**定义为

$$R_k(G) \triangleq -\ln \|G^k\|^{\frac{1}{k}},$$

渐进收敛速度定义为

$$R(G) \triangleq \lim_{k \rightarrow \infty} R_k(G) = -\ln \rho(G).$$



平均收敛速度与迭代步数和所用的范数有关, 但渐进收敛速度只依赖于迭代矩阵的谱半径.

误差估计

定理 考虑算法 $x^{(k+1)} = Gx^{(k)} + g$. 如果存在某个算子范数 $\|\cdot\|$ 使得 $\|G\| = q < 1$, 则

- (1) $\|x^{(k)} - x_*\| \leq q^k \|x^{(0)} - x_*\|$;
- (2) $\|x^{(k)} - x_*\| \leq \frac{q}{1-q} \|x^{(k)} - x^{(k-1)}\|$;
- (3) $\|x^{(k)} - x_*\| \leq \frac{q^k}{1-q} \|x^{(1)} - x^{(0)}\|$.

(板书)

一般来说, 好的迭代法应该满足:



- (1) $\rho(G)$ 很小;
- (2) 以 M 为系数矩阵的线性方程组比较容易求解.

3.2 不可约对角占优矩阵

Jacobi 和 G-S 的收敛性

定理 设 $A \in \mathbb{R}^{n \times n}$, 若 A 严格对角占优, 则 Jacobi 迭代法和 G-S 迭代法都收敛, 且

$$\|G_{GS}\|_{\infty} \leq \|G_J\|_{\infty} < 1.$$

(板书)

定理 设 $A \in \mathbb{R}^{n \times n}$, 若 A 不可约对角占优, 则 Jacobi 迭代法和 G-S 迭代法都收敛.

进一步, 若 A 是非负矩阵, 则

$$\rho(G_{GS}) < \rho(G_J) < 1.$$

(留作课外自习, 可参见参考资料)



上述结论对一般矩阵不成立: 对某些矩阵, Jacobi 迭代法收敛, 但 G-S 迭代法却不一定收敛.



思考: 若 $A \in \mathbb{R}^{n \times n}$ 严格对角占优且非负, 则是否有结论 $\rho(G_{GS}) \leq \rho(G_J)$?

SOR 的收敛性

关于 SOR 迭代法, 我们有下面的结论.

定理 设 $A \in \mathbb{R}^{n \times n}$, 若 A 严格对角占优且 $0 < \omega \leq 1$, 则 SOR 迭代法收敛.

(板书)

定理 设 $A \in \mathbb{R}^{n \times n}$, 若 A 不可约对角占优且 $0 < \omega \leq 1$, 则 SOR 迭代法收敛.

(留作课外自习)

3.3 对称正定矩阵情形

定理 (Richardson 迭代法的收敛性.) 设 $A \in \mathbb{R}^{n \times n}$ 对称正定, λ_1 和 λ_n 分别是 A 的最大和最小特征值, 则 Richardson 迭代法收敛当且仅当 $0 < \omega < 2/\lambda_1$.

另外, 最优参数为 $\omega_* = \arg \min_{\omega} \rho(G_R) = \frac{2}{\lambda_1 + \lambda_n}$, 且

$$\rho(G_R) = \begin{cases} 1 - \omega\lambda_n, & \text{if } \omega \leq \omega_*, \\ \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} = \frac{\kappa(A) - 1}{\kappa(A) + 1}, & \text{if } \omega = \omega_*, \\ \omega\lambda_1 - 1, & \text{if } \omega \geq \omega_*. \end{cases}$$

(留作练习)

Jacobi, Gauss-Seidel 和 SOR 的收敛性

首先给出两个引理.

引理 设 $A \in \mathbb{C}^{n \times n}$ 是 Hermite 矩阵, 且 $A = M - N$ 是 A 的一个矩阵分裂, 则 $M^* + N$ 也是 Hermite 矩阵, 且对任意 $x \in \mathbb{C}^n$ 有

$$x^*Ax - \tilde{x}^*A\tilde{x} = u^*(M^* + N)u,$$

其中 $\tilde{x} = M^{-1}Nx$, $u = x - \tilde{x}$.

(板书)

引理 设 $A \in \mathbb{R}^{n \times n}$ 对称, 矩阵分裂 $A = M - N$ 满足 $M^T + N$ 正定, 则 $\rho(M^{-1}N) < 1$ 当且仅当 A 是正定的. (板书)

SOR 收敛的必要条件

定理 对于 SOR 迭代法, 有 $\rho(G_{\text{SOR}}) \geq |1 - \omega|$, 故 SOR 收敛的必要条件是 $0 < \omega < 2$.

(板书)

对称正定情形迭代法的收敛性

定理 设 $A \in \mathbb{R}^{n \times n}$ 对称正定.

- (1) 若 $2D - A$ 正定, 则 Jacobi 迭代法收敛.
- (2) 若 $0 < \omega < 2$, 则 SOR 迭代法和 SSOR 迭代法收敛.
- (3) G-S 迭代法收敛.

(留作课外自习, 直接利用前面的引理即可)

定理 设 $A \in \mathbb{R}^{n \times n}$ 对称且 D 正定 (即 A 的对角线均为正).

- (1) 若 Jacobi 迭代法收敛, 则 A 和 $2D - A$ 都正定;
- (2) 若存在 $\omega \in (0, 2)$ 使得 SOR (或 SSOR) 迭代法收敛, 则 A 正定;
- (3) 若 G-S 迭代法收敛, 则 A 正定.

(板书, 只需证明 (1), 其余直接利用前面的引理)

对称正定情形: 收敛充要条件

推论 设 $A \in \mathbb{R}^{n \times n}$ 对称且 D 正定, 则

- (1) Jacobi 收敛的充要条件是 A 和 $2D - A$ 都正定;
- (2) G-S 收敛的充要条件是 A 正定;
- (3) SOR 收敛的充要条件是 A 正定且 $\omega \in (0, 2)$.

3.4 相容次序矩阵

- ▶ 针对一类特殊的矩阵, 这三种迭代法的谱半径之间存在一种特殊关系.
- ▶ 根据这个关系, 在某些场合下就可以确定 SOR 的最优参数的取值.

定义 设 $A \in \mathbb{R}^{n \times n}$ 的对角线元素全不为零, 记

$$\tilde{L} = D^{-1}L, \quad \tilde{U} = D^{-1}U,$$

即 $A = D(I - \tilde{L} - \tilde{U})$. 若矩阵

$$G(\alpha) \triangleq \alpha\tilde{L} + \frac{1}{\alpha}\tilde{U}$$

的特征值与 α 无关 ($\alpha \neq 0$), 则称 A 是相容次序矩阵.

一类特殊的相容次序矩阵

引理 设 $B \in \mathbb{R}^{n \times n}$ 具有下面的结构

$$B = \begin{bmatrix} 0 & B_{12} \\ B_{21} & 0 \end{bmatrix}, \quad \text{其中 } B_{12} \in \mathbb{R}^{k \times (n-k)}, B_{21} \in \mathbb{R}^{(n-k) \times k}, \quad 0 \leq k \leq n.$$

令 B_L 和 B_U 分别表示 B 的下三角和上三角部分, 则

- (1) 若 μ 是 B 的特征值, 则 $-\mu$ 也是 B 的特征值;
- (2) $B(\alpha)$ 的特征值与 α 无关, 其中

$$B(\alpha) = \alpha B_L + \frac{1}{\alpha} B_U, \quad \alpha \neq 0.$$

(板书)



$B(\alpha) + \beta I$ 的特征值也与 α 无关, 其中 β 为任意常数.

性质 A

定义 设 $A \in \mathbb{R}^{n \times n}$, 如果存在一个置换矩阵 P , 使得

$$PAP^T = \begin{bmatrix} D_1 & F \\ E & D_2 \end{bmatrix}, \quad (6.5)$$

其中 D_1, D_2 为对角矩阵, 则称 A 具有 **性质 A**.

定理 设 A 的对角线元素全不为零, 若 A 具有性质 A, 则存在置换矩阵 P , 使得 PAP^T 是相容次序矩阵.



性质 A 意味着指标集 $\mathbb{Z}_n = \{1, 2, \dots, n\}$ 可以分成两个互不相交的子集之和, 在同一个子集内的未知量相互独立, 因此在迭代更新时可以并行计算. (也就是说, $\mathbb{Z}_n = \mathcal{S} \cup \mathcal{T}$, $\mathcal{S} \cap \mathcal{T} = \emptyset$, 当 $i \neq j$ 且 $i, j \in \mathcal{S}$ 或 $i, j \in \mathcal{T}$ 时, 有 $a_{ij} = 0$.)



相容次序矩阵和性质 A 有不同的定义方式, 我们这里采用的是 Demmel (1997) 中的定义. 其他定义可参见 Varga (1962, 2000) 或 Young (1971).

事实上, 根据 Axelsson (1994) 中的说法, Yong (1950) 也是这样定义相容次序矩阵和性质 A 的.

Demmel (1997), *Applied Numerical Linear Algebra*.

Axelsson (1994), *Iterative Solution Methods*.

Varga (1962, 2000), *Matrix Iterative Analysis*.

Young (1950), *Iterative Methods for Solving Partial Difference Equations of Elliptic Type*, PhD thesis.

Young (1971), *Iterative Solution of Large Linear Systems*.

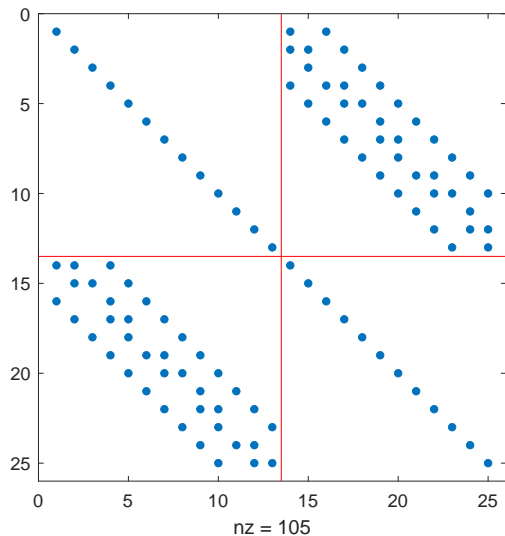
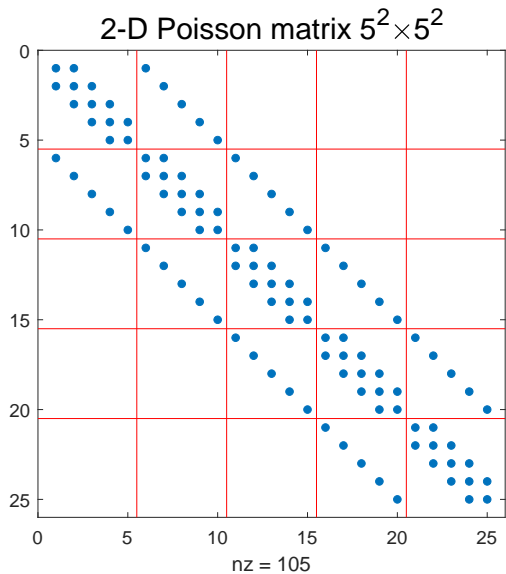
分块三对角矩阵情形

例 设 D_i 是非奇异的对角矩阵, 则任意块三对角矩阵

$$\begin{bmatrix} D_1 & A_1 & & \\ B_1 & \ddots & \ddots & \\ & \ddots & \ddots & A_{N-1} \\ & & B_{N-1} & D_N \end{bmatrix}$$

都是相容次序矩阵.

(留作练习)



有限差分离散的二维偏微分方程

Jacobi 和 SOR 迭代矩阵的特征值关系

定理 设 A 是相容次序矩阵且 $\omega \neq 0$, 则下列命题成立

- (1) Jacobi 迭代矩阵 G_J 的特征值正负成对出现;
- (2) 若 μ 是 G_J 的一个特征值且 λ 满足

$$(\lambda + \omega - 1)^2 = \lambda \omega^2 \mu^2, \quad (6.6)$$

则 λ 是 SOR 迭代矩阵 G_{SOR} 的特征值;

- (3) 反之, 若 $\lambda \neq 0$ 是 G_{SOR} 的一个特征值且 μ 满足 (6.6), 则 μ 是 G_J 的特征值.

(板书)

推论 设 A 是相容次序矩阵. 若 G_J 的特征值全部为实数, 则 SOR 迭代法收敛的充要条件是 $0 < \omega < 2$ 且 $\rho(G_J) < 1$. (留作练习)

推论 若 A 是相容次序矩阵, 则 $\rho(G_{GS}) = \rho(G_J)^2$, 即当 Jacobi 迭代法收敛时, Gauss-Seidel 迭代法比 Jacobi 迭代法快一倍.

SOR 迭代法的最优参数选取

定理 设 A 是相容次序矩阵. 若 G_J 的特征值全部为实数, 且 $\rho_J \triangleq \rho(G_J) < 1$, 则 SOR 迭代法的最优参数为

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho_J^2}},$$

此时

$$\rho(G_{SOR}) = \omega_{opt} - 1 = \frac{\rho_J^2}{\left(1 + \sqrt{1 - \rho_J^2}\right)^2}.$$

进一步, 有

$$\rho(G_{SOR}) = \begin{cases} \omega - 1, & \omega_{opt} \leq \omega \leq 2, \\ 1 - \omega + \frac{1}{2}\omega^2\rho_J^2 + \omega\rho_J\sqrt{1 - \omega + \omega^2\rho_J^2/4}, & 0 < \omega \leq \omega_{opt}. \end{cases}$$

(留作课外自习, 直接求解等式 (6.6), 分情况讨论即可.)

4 | 应用: Poisson 方程求解

4.1 一维 Poisson 方程

4.2 二维 Poisson 方程

4.3 收敛性分析

4.4 快速 Poisson 算法

在本讲中, 我们以一个典型的线性方程组为例, 逐个介绍各种迭代法, 并比较它们的性能.

这个方程组就是二维 Poisson 方程经过五点差分离散后得到的线性方程组.

4.1 一维 Poisson 方程

考虑如下带 Dirichlet 边界条件的一维 Poisson 方程

$$\begin{cases} -\frac{d^2u(x)}{dx^2} = f(x), & 0 < x < 1, \\ u(0) = a, & u(1) = b, \end{cases}$$

其中 $f(x)$ 是给定的函数, $u(x)$ 是需要计算的未知函数.

差分离散

取步长 $h = \frac{1}{n+1}$, 节点为 $x_i = ih$, $i = 0, 1, 2, \dots, n+1$.

▶ 在 x_i 点采用中心差分离散, 可得 ($i = 1, 2, \dots, n$)

$$-\frac{d^2u(x)}{dx^2} \Big|_{x_i} = \frac{2u(x_i) - u(x_{i-1}) - u(x_{i+1}))}{h^2} + O\left(h^2 \cdot \left\| \frac{d^4u}{dx^4} \right\|_{\infty}\right)$$

代入原微分方程, 舍去高阶项后可得 Poisson 方程在 x_i 点的近似离散方程

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i,$$

其中 $f_i = f(x_i)$, u_i 为 $u(x_i)$ 的近似.

令 $i = 1, 2, \dots, n$, 则可得 n 个线性方程, 写成矩阵形式

$$T_n u = f,$$

其中

$$T_n = \begin{bmatrix} 2 & -1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & -1 & \\ & & -1 & 2 & \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}, \quad f = \begin{bmatrix} f_1 + u_0 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n + u_{n+1} \end{bmatrix}.$$

系数矩阵 T_n 的性质

引理 T_n 的特征值和对应的特征向量分别为

$$\lambda_k = 2 - 2 \cos \frac{k\pi}{n+1},$$

$$z_k = \sqrt{\frac{2}{n+1}} \left[\sin \frac{k\pi}{n+1}, \sin \frac{2k\pi}{n+1}, \dots, \sin \frac{nk\pi}{n+1} \right]^T, \quad k = 1, 2, \dots, n,$$

即 $T_n = Z\Lambda Z^T$, 其中 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, $Z = [z_1, z_2, \dots, z_n]$.

引理 更一般地, 设 $T = \text{tridiag}(a, b, c) \in \mathbb{R}^{n \times n}$, 则 T 的特征值为

$$\lambda_k = b - 2\sqrt{ac} \cos \frac{k\pi}{n+1}, \quad k = 1, 2, \dots, n,$$

对应的特征向量为 z_k , 其第 j 个分量为

$$z_k(j) = \left(\frac{a}{c}\right)^{\frac{j}{2}} \sin \frac{jk\pi}{n+1}.$$

特别地, 若 $a = c = 1$, 则对应的单位特征向量为

$$z_k = \sqrt{\frac{2}{n+1}} \left[\sin \frac{k\pi}{n+1}, \sin \frac{2k\pi}{n+1}, \dots, \sin \frac{nk\pi}{n+1} \right]^T.$$

(留作练习)

由前面的结论可知, T_n 是对称正定的, 其最大特征值为

$$2 \left(1 - \cos \frac{n\pi}{n+1} \right) = 4 \sin^2 \frac{n\pi}{2(n+1)} \approx 4,$$

最小特征值为

$$2 \left(1 - \cos \frac{\pi}{n+1} \right) = 4 \sin^2 \frac{\pi}{2(n+1)} \approx \left(\frac{\pi}{n+1} \right)^2.$$

因此, 当 n 很大时, T_n 的谱条件数约为

$$\kappa_2(T_n) \approx \frac{4(n+1)^2}{\pi^2}$$

4.2 二维 Poisson 方程

现在考虑二维 Poisson 方程

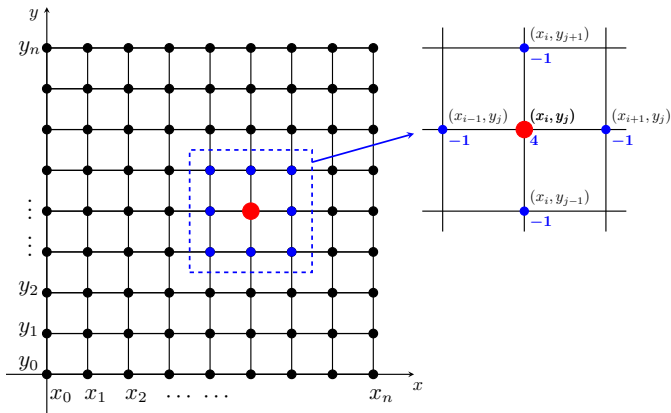
$$\begin{cases} -\Delta u(x, y) = -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y), & (x, y) \in \Omega, \\ u(x, y) = u_0(x, y), & (x, y) \in \partial\Omega \end{cases}$$

其中 $\Omega = [0, 1] \times [0, 1]$ 为求解区域, $\partial\Omega$ 表示 Ω 的边界.

五点差分离散

为了简单起见, 在 x -方向和 y -方向取相同的步长 $h = \frac{1}{n+1}$

节点: $x_i = ih, y_j = jh, \quad i, j = 0, 1, 2, \dots, n.$



在 x -方向和 y -方向同时采用中心差分离散可得

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{(x_i, y_j)} \approx \frac{2u(x_i, y_j) - u(x_{i-1}, y_j) - u(x_{i+1}, y_j)}{h^2}$$
$$\left. \frac{\partial^2 u}{\partial y^2} \right|_{(x_i, y_j)} \approx \frac{2u(x_i, y_j) - u(x_i, y_{j-1}) - u(x_i, y_{j+1})}{h^2}.$$

► 代入原方程, 即得二维 Poisson 方程在 (x_i, y_j) 点的近似离散方程

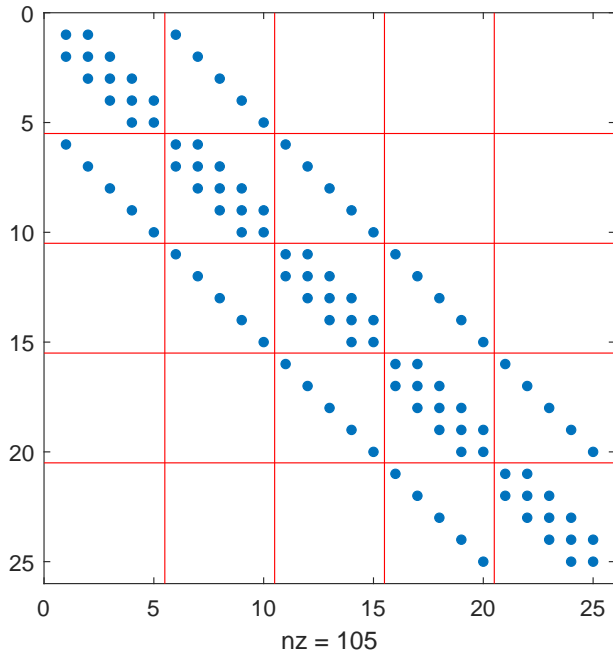
$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = h^2 f_{i,j},$$

其中 $f_{ij} = f(x_i, y_j)$, $u_{i,j}$ 为 $u(x_i, y_j)$ 的近似.

► 记 $T_N \triangleq I \otimes T_n + T_n \otimes I$, $N = n^2$, $u = [u_{1,1}, \dots, u_{n,1}, u_{1,2}, \dots, u_{n,2}, \dots \dots]$, 则

$$T_N u = h^2 f$$

2-D Poisson matrix $5^2 \times 5^2$



系数矩阵 T 的性质

因为 $T_N = I \otimes T_n + T_n \otimes I$, 由 Kronecker 乘积的性质即得

定理 设 $T_n = Z\Lambda Z^T$, 其中 $Z = [z_1, z_2, \dots, z_n]$ 为正交阵, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ 为对角阵, 则 T 的特征值分解为

$$T_N = (Z \otimes Z)(I \otimes \Lambda + \Lambda \otimes I)(Z \otimes Z)^T,$$

即 T 的特征值为 $\lambda_i + \lambda_j$, 对应的特征向量为 $z_i \otimes z_j$, $i, j = 1, 2, \dots, n$.

条件数

$$\kappa(T_N) = \frac{\lambda_{\max}(T_N)}{\lambda_{\min}(T_N)} = \frac{1 - \cos \frac{n\pi}{n+1}}{1 - \cos \frac{\pi}{n+1}} = \frac{\sin^2 \frac{n\pi}{2(n+1)}}{\sin^2 \frac{\pi}{2(n+1)}} \approx \frac{4(n+1)^2}{\pi^2}.$$

二维离散 Poisson 方程的 Jacobi 迭代法

算法 4 求解二维离散 Poisson 方程的 Jacobi 迭代法

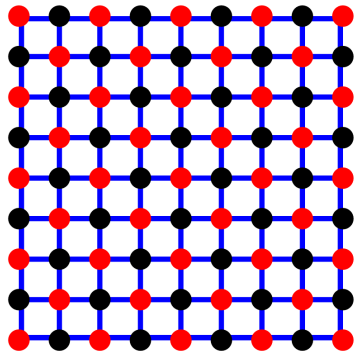
- 1: Choose an initial guess $v^{(0)}$
- 2: **while** not converge **do**
- 3: **for** $i = 1$ to N **do**
- 4: **for** $j = 1$ to N **do**
- 5: $u_{i,j}^{(k+1)} = \left(h^2 f_{i,j} + u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)} \right) / 4$
- 6: **end for**
- 7: **end for**
- 8: **end while**

G-S 算法的并行计算: 红黑排序

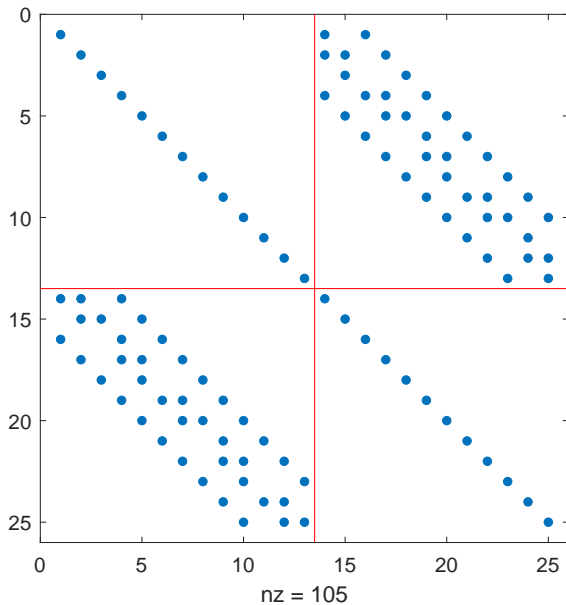
基于自然顺序的 G-S 方法不适合并行计算.

下面我们介绍一种新的适合并行计算的未知量排序方法:
红黑排序, 即将二维网格点依次做红黑记号, 如右图所示.

- ▶ 在计算过程中, 对未知量的值进行更新时, 我们可以先更新红色节点, 此时所使用的只是黑色节点的数据, 然后再更新黑色节点, 这时使用的是红色节点的数据.
- ▶ 由于在更新红点时, 各个点之间是相互独立的, 因此可以并行计算. 同样, 在更新黑点时, 各个点之间也是相互独立的, 因此也可以并行计算.



基于红黑排序的系数矩阵



算法 5 求解二维离散 Poisson 方程的红黑排序 G-S 迭代法

1: Choose an initial guess $v^{(0)}$

2: **while** not converge **do**

3: **for** (i, j) 为红色节点 **do**

4:
$$u_{i,j}^{(k+1)} = \frac{1}{4} \left(h^2 f_{i,j} + u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)} \right)$$

5: **end for**

6: **for** (i, j) 为黑色节点 **do**

7:
$$u_{i,j}^{(k+1)} = \frac{1}{4} \left(h^2 f_{i,j} + u_{i+1,j}^{(k+1)} + u_{i-1,j}^{(k+1)} + u_{i,j+1}^{(k+1)} + u_{i,j-1}^{(k+1)} \right)$$

8: **end for**

9: **end while**

算法 6 求解二维离散 Poisson 方程的红黑排序 SOR 迭代法

1: Choose an initial guess $v^{(0)}$ and parameter ω

2: **while** not converge **do**

3: **for** (i, j) 为红色节点 **do**

4:
$$u_{i,j}^{(k+1)} = (1 - \omega)v_{i,j}^{(k)} + \omega(h^2 f_{i,j} + u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)})/4$$

5: **end for**

6: **for** (i, j) 为黑色节点 **do**

7:
$$u_{i,j}^{(k+1)} = (1 - \omega)v_{i,j}^{(k)} + \omega(h^2 f_{i,j} + u_{i+1,j}^{(k+1)} + u_{i-1,j}^{(k+1)} + u_{i,j+1}^{(k+1)} + u_{i,j-1}^{(k+1)})/4$$

8: **end for**

9: **end while**

例 已知二维 Poisson 方程

$$\begin{cases} -\Delta u(x, y) = -1, & (x, y) \in \Omega \\ u(x, y) = \frac{x^2 + y^2}{4}, & (x, y) \in \partial\Omega \end{cases}$$

其中 $\Omega = (0, 1) \times (0, 1)$. 该方程的解析解是 $u(x, y) = \frac{x^2 + y^2}{4}$. 用五点差分格式离散后得到一个线性方程组.

- (1) 分别用 Jacobi, G-S 和 SOR 方法计算这个方程组的解.
- (2) 分别用 SOR 和 SSOR 方法求解方程, 观察参数 ω 对方法收敛的影响.

[Poisson_Jacobi_GS_SOR.m](#), [Poisson_SOR_omega.m](#), [Poisson_SSOR_omega.m](#)

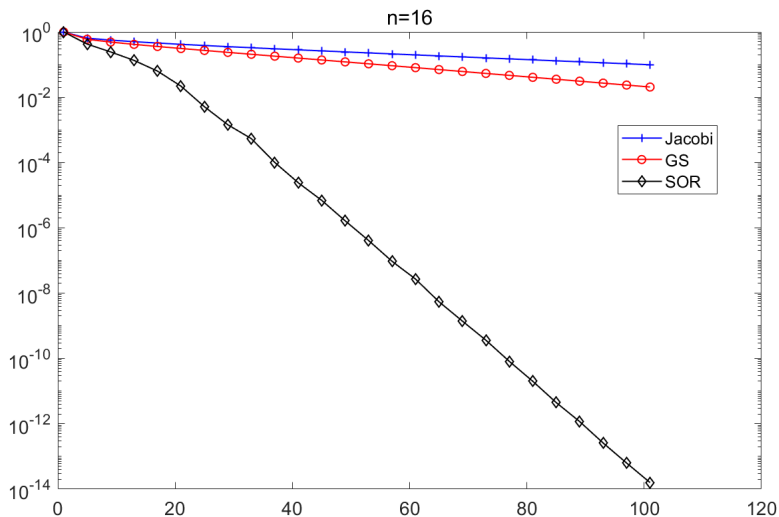


图 4.1 Jacobi, G-S 和 SOR 的近似解相对误差的下降曲线.

$n = 8$ (即 $N = 64$) 时, SOR 和 SSOR 收敛结果与参数 ω 取值之间的关系.

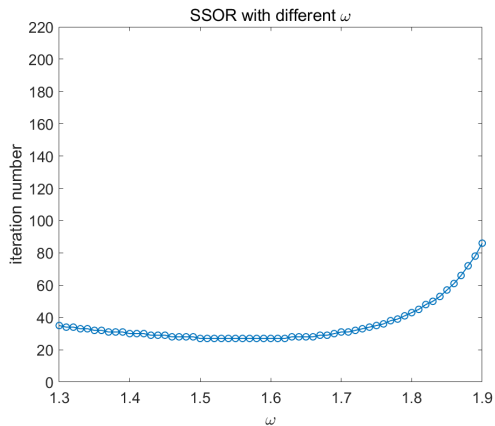
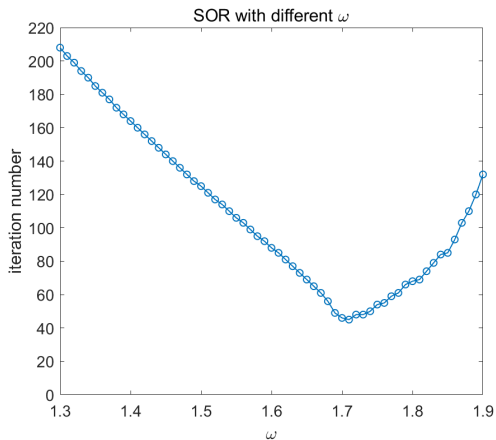


图 4.2 SOR 和 SSOR 的收敛结果与 ω 取值的关系.

4.3 收敛性分析

考虑二维离散 Poisson 方程, 系数矩阵为

$$A = T = I \otimes T_n + T_n \otimes I.$$

Jacobi 的收敛性

Jacobi 迭代矩阵

$$G_J = D^{-1}(L + U) = (4I)^{-1}(4I - T) = I - T/4$$

特征值 $1 - (\lambda_i + \lambda_j)/4 = \frac{1}{2} \left(\cos \frac{\pi i}{n+1} + \cos \frac{\pi j}{n+1} \right)$, 故

$$\rho(G_J) = \cos \frac{\pi}{n+1} < 1,$$



所以 Jacobi 迭代法是收敛的.

G-S 和 SOR 的收敛性

定理 设离散时采用红黑排序, 则有

$$\rho(G_{GS}) = \rho(G_J)^2 = \cos^2 \frac{\pi}{n+1} < 1$$
$$\rho(G_{SOR}) = \frac{\cos^2 \frac{\pi}{n+1}}{\left(1 + \sin \frac{\pi}{n+1}\right)^2} < 1, \quad \omega = \frac{2}{1 + \sin \frac{\pi}{n+1}}.$$

(这里的 ω 是最优参数)



事实上, 由于二维离散 Poisson 方程的系数矩阵 T_N 是不可约对角占优的, 因此 Jacobi 和 G-S 都收敛. 另外, T_N 是对称正定的, 故当 $0 < \omega < 2$ 时, SOR 迭代法收敛.

收敛速度比较

由 Taylor 公式可知, 当 n 很大时, 有

$$\rho(G_J) = \cos \frac{\pi}{n+1} \approx 1 - \frac{\pi^2}{2(n+1)^2} = 1 - O\left(\frac{1}{n^2}\right),$$
$$\rho(G_{\text{SOR}}) = \frac{\cos^2 \frac{\pi}{n+1}}{\left(1 + \sin \frac{\pi}{n+1}\right)^2} \approx 1 - \frac{2\pi}{n+1} = 1 - O\left(\frac{1}{n}\right).$$

由于当 n 很大时有

$$\left(1 - \frac{1}{n}\right)^k \approx 1 - \frac{k}{n} = 1 - \frac{kn}{n^2} \approx \left(1 - \frac{1}{n^2}\right)^{kn},$$

即 SOR 迭代 k 步后误差的减小量与 Jacobi 迭代 kn 步后误差减小量差不多.

G-S 大约是 Jacobi 的 2 倍, SOR 大约是 Jacobi 的 n 倍.

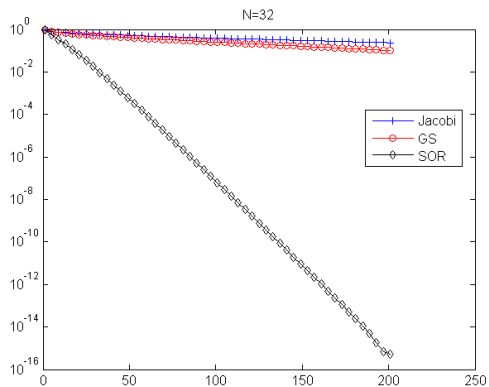
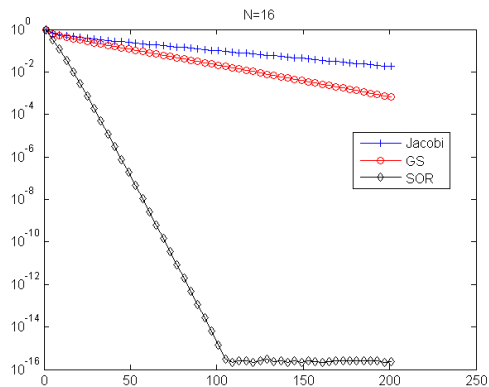
例 已知二维 Poisson 方程

$$\begin{cases} -\Delta u(x, y) = -1, & (x, y) \in \Omega \\ u(x, y) = \frac{x^2 + y^2}{4}, & (x, y) \in \partial\Omega \end{cases}$$

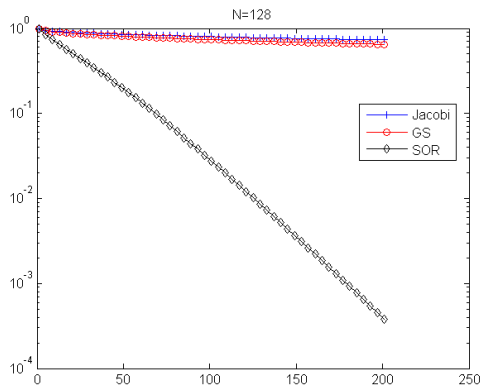
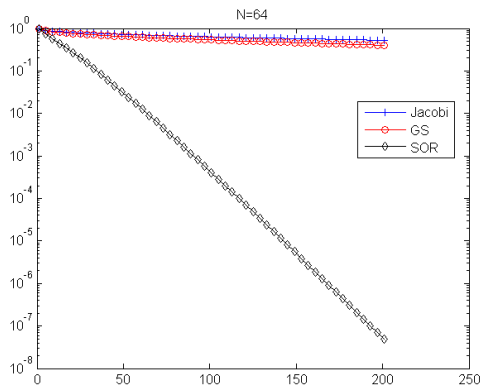
其中 $\Omega = (0, 1) \times (0, 1)$. 该方程的解析解是 $u(x, y) = \frac{x^2 + y^2}{4}$. 用五点差分格式离散后得到一个线性方程组, 分别用 Jacobi, G-S 和 SOR 计算这个方程组的解, 并比较收敛效果.

[Poisson_Jacobi_GS_SOR_convergence.m](#)

$N = 16, 32$ 时三种方法的相对误差下降情况.



$N = 64, 128$ 时三种方法的相对误差下降情况.



4.4 快速 Poisson 算法 *

📁 如果已经知道 A 的特征值分解 $A = X\Lambda X^{-1}$, 则 $Ax = b$ 的解可表示为

$$x = A^{-1}b = X\Lambda^{-1}X^{-1}b.$$

📁 如果 A 是正规矩阵, 则 X 可以取酉矩阵, 于是

$$x = A^{-1}b = X\Lambda^{-1}X^*b.$$

- ▶ 一般不会采用特征值分解来解线性方程组, 因为计算特征值分解通常比解线性方程组更困难.
- ▶ 但在某些特殊情况下, 我们可以由此得到快速算法.

考虑二维离散 Poisson 方程

$$Tu = h^2 f,$$

其中


$$T = I \otimes T_n + T_n \otimes I = (Z \otimes Z)(I \otimes \Lambda + \Lambda \otimes I)(Z \otimes Z)^T.$$

这里 $Z = [z_1, z_2, \dots, z_n]$ 是正交矩阵, 其中

$$z_k = \sqrt{\frac{2}{n+1}} \cdot \left[\sin \frac{k\pi}{n+1}, \sin \frac{2k\pi}{n+1}, \dots, \sin \frac{nk\pi}{n+1} \right]^T, \quad k = 1, 2, \dots, n.$$

所以, 方程组的解为

$$u = T^{-1}h^2 f = [(Z \otimes Z)(I \otimes \Lambda + \Lambda \otimes I)^{-1}(Z \otimes Z)^T] h^2 f.$$

 因此, 主要的运算是 $Z \otimes Z$ 与向量的乘积, 以及 $(Z \otimes Z)^T$ 与向量的乘积.

离散 Sine 变换

📁 离散 Sine 变换有多种定义, 这里介绍与求解 Poisson 方程有关的一种.

📁 设 $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$, 其 **离散 Sine 变换 (DST)** 定义为

$$y = \text{DST}(x) = [y_1, y_2, \dots, y_n]^T \in \mathbb{R}^n,$$

其中

$$y_k = \sum_{j=1}^n x_j \sin\left(\frac{kj\pi}{n+1}\right), \quad k = 1, 2, \dots, n.$$

📁 对应的离散 Sine 反变换记为 IDST, 即 $x = \text{IDST}(y)$, 其中

$$x_j = \frac{2}{n+1} \sum_{k=1}^n y_k \sin\left(\frac{jk\pi}{n+1}\right), \quad j = 1, 2, \dots, n.$$

DST 和 IDST 满足下面的性质:

$$\text{IDST}(\text{DST}(x)) = x, \quad \text{DST}(\text{IDST}(y)) = y.$$

在 MATLAB 中, 计算 DST 和 IDST 的函数分别为 `dst` 和 `idst`, 即:



```
y=dst(x), x=idst(y)
```

(测试代码见 `DST_test.m`)

Possion 方程与 DST

我们首先考虑矩阵 Z 与一个任意给定向量 b 的乘积.

设 $y = Zb$, 则

$$y_k = \sum_{j=1}^n Z(k, j)b_j = \sqrt{\frac{2}{n+1}} \sum_{j=1}^n b_j \sin\left(\frac{kj\pi}{n+1}\right) = \sqrt{\frac{2}{n+1}} \cdot [\text{DST}(b)]_k.$$

因此, 乘积 $y = Zb$ 可以通过 DST 来实现.

类似地, 乘积 $y = Z^T b = Z^{-1}b$ 可以通过离散 Sine 反变换 IDST 实现, 即

$$y = Z^T b = Z^{-1}b = \left(\sqrt{\frac{2}{n+1}}\right)^{-1} \text{IDST}(b).$$

一维离散 Poisson 方程

解为

$$u = T_n^{-1}(h^2 f) = (Z\Lambda^{-1}Z^T)(h^2 f) = h^2 Z\Lambda^{-1}Z^T f = h^2 \cdot \text{DST}(\Lambda^{-1} \text{IDST}(b)).$$

特征值计算

可以通过公式 $T_n = Z\Lambda Z^T$ 来计算一维离散 Poisson 矩阵的特征值:

► 记 $[\alpha_1, \alpha_2, \dots, \alpha_n]^T$ 和 $[\beta_1, \beta_2, \dots, \beta_n]^T$ 分别为 $Z^T T_n$ 和 Z^T 的第一列, 则 T_n 的特征值为

$$\lambda_i = \frac{\alpha_i}{\beta_i}.$$

对应的 MATLAB 代码为: `Lam=idst([2,-1,zeros(1,n-2)]') ./ idst(eye(n,1))`

二维离散 Poisson 方程

我们需要计算 $(Z \otimes Z)b$ 和 $(Z^T \otimes Z^T)b$.

📁 它们对应的是二维离散 Sine 变换和二维离散 Sine 反变换.

设 $b = [b_1^T, b_2^T, \dots, b_n^T]^T \in \mathbb{R}^{n^2}$, 其中 $b_k \in \mathbb{R}^{n \times n}$.

▶ 令 $B = [b_1, b_2, \dots, b_n] \in \mathbb{R}^{n \times n}$, 则由 Kronecker 乘积的性质可知

$$(Z \otimes Z)b = (Z \otimes Z)\text{vec}(B) = \text{vec}(ZBZ^T) = \text{vec}\left(\left(Z(ZB)^T\right)^T\right).$$

▶ 因此, 我们仍可以使用 DST 来计算 $(Z \otimes Z)b$.

▶ 类似地, 我们可以使用 IDST 来计算 $(Z^T \otimes Z^T)b$.

算法7 二维离散 Poisson 方程的快速算法

- 1: 计算 $b = h^2 f$
- 2: $B = \text{reshape}(b, n, n)$
- 3: $B_1 = (Z^\top B)^\top = (\text{IDST}(B))^\top$
- 4: $B_2 = (Z^\top B_1)^\top = (\text{IDST}(B_1))^\top$
- 5: $b_1 = (I \otimes \Lambda + \Lambda \otimes I)^{-1} \text{vec}(B_2)$
- 6: $B_3 = \text{reshape}(b_1, n, n)$
- 7: $B_4 = (Z B_3)^\top = (\text{DST}(B_3))^\top$
- 8: $B_5 = (Z B_4)^\top = (\text{DST}(B_4))^\top$
- 9: $u = \text{reshape}(B_5, n^2, 1)$

MATLAB 程序见 [Poisson_DST.m](#)

二维离散 Poisson 方程的常用算法

	方法	串行时间	存储空间
直接法	稠密 Cholesky 分解	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$
	显式求逆	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
	带状 Cholesky 分解	$\mathcal{O}(N^2)$	$\mathcal{O}(N^{3/2})$
	稀疏 Cholesky 分解	$\mathcal{O}(N^{3/2})$	$\mathcal{O}(N \log N)$
经典迭代	Jacobi	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
	Gauss-Seidel	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
	SOR	$\mathcal{O}(N^{3/2})$	$\mathcal{O}(N)$
	带 Chebyshev 加速的 SSOR	$\mathcal{O}(N^{5/4})$	$\mathcal{O}(N)$
Krylov 子空间迭代	CG (共轭梯度法)	$\mathcal{O}(N^{3/2})$	$\mathcal{O}(N)$
	CG (带修正 IC 预处理)	$\mathcal{O}(N^{5/4})$	$\mathcal{O}(N)$
快速算法	FFT (快速 Fourier 变换)	$\mathcal{O}(N \log N)$	$\mathcal{O}(N)$
	块循环约化	$\mathcal{O}(N \log N)$	$\mathcal{O}(N)$
	Multigrid	$\mathcal{O}(N)$	$\mathcal{O}(N)$

5 | 加速方法

5.1 外推技术

5.2 Chebyshev 加速

当迭代解 $x^{(0)}, x^{(1)}, x^{(2)}, \dots, x^{(k)}$ 已经计算出来后, 我们可以对其进行组合, 得到一个新的更好的近似解, 这样就可以对原算法进行加速.

5.1 外推技术

外推算法

设原迭代格式为

$$x^{(k+1)} = Gx^{(k)} + g. \quad (6.7)$$

由 $x^{(k)}$ 和 $x^{(k+1)}$ 加权组合后可得新的近似解

$$x^{(k+1)} = (1 - \omega)x^{(k)} + \omega(Gx^{(k)} + g), \quad (6.8)$$

其中 ω 是参数. 这种加速方法就称为 **外推算法**.

加速: 选择 ω 使得其迭代矩阵 $G_\omega \triangleq (1 - \omega)I + \omega G$ 的谱半径尽可能地小



假设 G 的特征值都是实数, 且最大特征值和最小特征值分别为 λ_1 和 λ_n , 于是

$$\rho(G_\omega) = \max_{\lambda \in \sigma(G)} |(1 - \omega) + \omega\lambda| = \max\{|1 - \omega + \omega\lambda_1|, |1 - \omega + \omega\lambda_n|\}.$$

定理 设 G 的特征值都是实数, 最大和最小特征值分别为 λ_1 和 λ_n , 且 $1 \notin [\lambda_n, \lambda_1]$, 则

$$\omega_* = \arg \min_{\omega} \rho(G_\omega) = \frac{2}{2 - (\lambda_1 + \lambda_n)},$$

此时 $\rho(G_{\omega_*}) = 1 - |\omega_*|d$, 其中 d 是 1 到 $[\lambda_n, \lambda_1]$ 的距离, 即当 $\lambda_n \leq \lambda_1 < 1$ 时, $d = 1 - \lambda_1$, 当 $\lambda_1 \geq \lambda_n > 1$ 时, $d = \lambda_n - 1$. (留作课外自习)



- ▶ 若 $\omega_* \neq 1$ 时, 则外推迭代比原迭代法收敛要更快一些.
- ▶ 实际应用时可以估计特征值所在的区间 $[a, b]$, 然后用 a, b 代替 λ_n 和 λ_1 .

JOR 算法

对 Jacobi 迭代进行外推加速, 则可得 JOR (Jacobi over-relaxation) 算法:

$$\begin{aligned}x^{(k+1)} &= (1 - \omega)x^{(k)} + \omega(D^{-1}(L + U)x^{(k)} + D^{-1}b) \\ &= x^{(k)} + \omega D^{-1}(b - Ax^{(k)}), \quad k = 0, 1, 2, \dots\end{aligned}$$

定理 设 A 对称正定. 若

$$0 < \omega < \frac{2}{\rho(D^{-1}A)},$$

则 JOR 算法收敛.

5.2 Chebyshev 加速

多项式加速

假定通过迭代格式 $x^{(k+1)} = Gx^{(k)} + g$ 已计算出 $x^{(0)}, x^{(1)}, \dots, x^{(k)}$, 下面考虑如何将这
些近似解进行组合, 以便得到更精确的近似解.

记 $\varepsilon_k = x^{(k)} - x_*$ 为第 k 步迭代解的误差, 则有

$$\varepsilon_k = G\varepsilon_{k-1} = G^2\varepsilon_{k-2} = \cdots = G^k\varepsilon_0.$$

设 $\tilde{x}^{(k)}$ 为 $x^{(0)}, x^{(1)}, \dots, x^{(k)}$ 的一个线性组合, 即

$$\tilde{x}^{(k)} = \alpha_0 x^{(0)} + \alpha_1 x^{(1)} + \cdots + \alpha_k x^{(k)},$$

其中 α_i 为待定系数, 且满足 $\sum_{i=0}^k \alpha_i = 1$.

于是

$$\tilde{x}^{(k)} - x_* = \alpha_0 \varepsilon_0 + \alpha_1 G \varepsilon_0 + \cdots + \alpha_k G^k \varepsilon_0 \triangleq p_k(G) \varepsilon_0, \quad (6.9)$$

其中 $p_k(t) = \sum_{i=0}^k \alpha_i t^i$ 为 k 次多项式, 且满足 $p_k(1) = 1$.

我们希望通过适当选取参数 α_i , 使得 $\tilde{x}^{(k)} - x_*$ 尽可能地小, 即使得 $\tilde{x}^{(k)}$ 收敛到 x_* 速度远远快于 $x^{(k)}$ 收敛到 x_* 速度. 这种加速方法就称为**多项式加速**或**半迭代法 (semi-iterative method)**.

例 设 $p_n(t)$ 为 G 的特征多项式, 令 $\tilde{p}_n(t) \triangleq p_n(t)/p_n(1)$, 则 $\tilde{p}_n(1) = 1$ 且 $\tilde{p}_n(G) = 0$, 所以选取 α_i 为 \tilde{p}_n 的系数, 则 $\tilde{x}^{(n)} - x_* = 0$. 但这种选取方法不实用, 原因是:

- (1) $\tilde{p}_n(t)$ 的系数并不知道;
- (2) 我们通常希望收敛所需的迭代步数 $\ll n$.

参数 α_i 的较实用的选取方法

由 (6.9) 可知

$$\|\tilde{x}^{(k)} - x_*\|_2 = \|p_k(G)\varepsilon_0\|_2 \leq \|p_k(G)\|_2 \|\varepsilon_0\|_2.$$

因此我们需要求解下面的极小化问题

$$\min_{p \in \mathbb{P}_k, p(1)=1} \|p(G)\|_2$$

其中 \mathbb{P}_k 表示所有次数不超过 k 的多项式组成的集合.

- ▶ 一般来说, 这个问题是非常困难的.
- ▶ 但在一些特殊情况下, 我们可以给出其最优解.

假设迭代矩阵 G 是对称的, 即 G 存在特征值分解:

$$G = Q\Lambda Q^T,$$

其中 Λ 是实对角矩阵, U 是正交矩阵. 于是有

$$\begin{aligned} \min_{p \in \mathbb{P}_k, p(1)=1} \|p(G)\|_2 &= \min_{p \in \mathbb{P}_k, p(1)=1} \|p(\Lambda)\|_2 \\ &= \min_{p \in \mathbb{P}_k, p(1)=1} \max_{1 \leq i \leq n} \{|p(\lambda_i)|\} \\ &\leq \min_{p \in \mathbb{P}_k, p(1)=1} \max_{\lambda \in [\lambda_n, \lambda_1]} \{|p(\lambda)|\}, \end{aligned} \quad (6.10)$$

其中 λ_1, λ_n 分别表示 G 的最大和最小特征值.

- 这是带归一化条件的多项式最佳一致逼近问题 (与零的偏差最小).
- 该问题的解与著名的 **Chebyshev 多项式** 有关.

Chebyshev 多项式

Chebyshev 多项式 $T_k(t)$ 可以通过下面的递推方式来定义:

$$T_0(t) = 1, \quad T_1(t) = t,$$

$$T_k(t) = 2tT_{k-1}(t) - T_{k-2}(t), \quad k = 2, 3, \dots,$$

► 也可以直接由下面的式子定义

$$T_k(t) = \frac{1}{2} \left[\left(t + \sqrt{t^2 - 1} \right)^k + \left(t + \sqrt{t^2 - 1} \right)^{-k} \right],$$

► 或者

$$T_k(t) = \begin{cases} \cos(k \arccos t), & |t| \leq 1 \\ \cosh(k \operatorname{arccosh} t), & |t| > 1 \end{cases},$$

Chebyshev 的一个重要性质是下面的最小最大性质.

定理 设 $\eta \in \mathbb{R}$ 满足 $|\eta| > 1$, 则下面的最小最大问题

$$\min_{p(t) \in \mathbb{P}_k, p(\eta)=1} \max_{-1 \leq t \leq 1} |p(t)|$$

的唯一解为

$$\tilde{T}_k(t) \triangleq \frac{T_k(t)}{T_k(\eta)}.$$

通过简单的仿射变换, 该定理的结论可以推广到一般区间.

定理 设 $\alpha, \beta, \eta \in \mathbb{R}$ 满足 $\alpha < \beta$ 且 $|\eta| \notin [\alpha, \beta]$. 则下面的最小最大问题

$$\min_{p(t) \in \mathbb{P}_k, p(\eta)=1} \max_{\alpha \leq x \leq \beta} |p(t)|$$

的唯一解为

$$\hat{T}_k(t) \triangleq \frac{T_k\left(\frac{2t - (\beta + \alpha)}{\beta - \alpha}\right)}{T_k\left(\frac{2\eta - (\beta + \alpha)}{\beta - \alpha}\right)}.$$

Chebyshev 加速方法

考虑迭代格式 $x^{(k+1)} = Gx^{(k)} + g$, 我们假定:

- (1) 迭代矩阵 G 的特征值都是实数;
- (2) 迭代矩阵谱半径 $\rho = \rho(G) < 1$, 故 $\lambda(G) \in [-\rho, \rho] \subset (-1, 1)$.

于是最小最大问题 (6.10) 就转化为

$$\min_{p \in \mathbb{P}_k, p(1)=1} \max_{\lambda \in [-\rho, \rho]} \{|p(\lambda)|\}.$$

由于 $1 \notin [-\rho, \rho]$, 上述问题的解为

$$p_k(t) = \frac{T_k(t/\rho)}{T_k(1/\rho)}.$$

$\tilde{x}^{(k)}$ 的迭代格式

实际计算时, 我们无需先计算 $x^{(0)}, x^{(1)}, \dots, x^{(k)}$, 然后通过线性组合来计算 $\tilde{x}^{(k)}$.

事实上, 可通过 Chebyshev 多项式的三项递推公式, 由 $\tilde{x}^{(k-1)}$ 和 $\tilde{x}^{(k-2)}$ 直接得到 $\tilde{x}^{(k)}$

令 $\mu_k = \frac{1}{T_k(1/\rho)}$, 即 $T_k(1/\rho) = \frac{1}{\mu_k}$. 由 Chebyshev 多项式的三项递推公式可得

$$\frac{1}{\mu_k} = \frac{2}{\rho} \cdot \frac{1}{\mu_{k-1}} - \frac{1}{\mu_{k-2}}.$$

所以

$$\begin{aligned}\tilde{x}^{(k)} - x_* &= p_k(G) \varepsilon_0 = \mu_k T_k(G/\rho) \varepsilon_0 \\ &= \mu_k \left[\frac{2G}{\rho} \cdot \frac{1}{\mu_{k-1}} (\tilde{x}^{(k-1)} - x_*) - \frac{1}{\mu_{k-2}} (\tilde{x}^{(k-2)} - x_*) \right].\end{aligned}$$

整理后可得

$$\tilde{x}^{(k)} = \frac{2\mu_k}{\mu_{k-1}} \cdot \frac{G}{\rho} \tilde{x}^{(k-1)} - \frac{\mu_k}{\mu_{k-2}} \tilde{x}^{(k-2)} + d_k,$$

其中

$$\begin{aligned} d_k &= x_* - \frac{2\mu_k}{\mu_{k-1}} \cdot \frac{G}{\rho} x_* + \frac{\mu_k}{\mu_{k-2}} x_* \\ &= x_* - \frac{2\mu_k}{\mu_{k-1}} \cdot \frac{x_* - g}{\rho} + \frac{\mu_k}{\mu_{k-2}} x_* \\ &= \mu_k \left(\frac{1}{\mu_k} - \frac{2}{\rho\mu_{k-1}} + \frac{1}{\mu_{k-2}} \right) x_* + \frac{2\mu_k g}{\mu_{k-1}\rho} \\ &= \frac{2\mu_k g}{\mu_{k-1}\rho}. \end{aligned}$$

由此, 我们可以得到迭代格式 (6.7) 的 Chebyshev 加速算法.

算法 8 Chebyshev 加速算法

1: Set $\mu_0 = 1, \mu_1 = \rho = \rho(G), \tilde{x}^{(0)} = x^{(0)}, k = 1$

2: compute $\tilde{x}^{(1)} = Gx^{(0)} + g$

3: **while** not converge **do**

4: $k = k + 1$

5:
$$\mu_k = \left(\frac{2}{\rho} \cdot \frac{1}{\mu_{k-1}} - \frac{1}{\mu_{k-2}} \right)^{-1}$$

6:
$$\tilde{x}^{(k)} = \frac{2\mu_k}{\mu_{k-1}} \cdot \frac{G}{\rho} \tilde{x}^{(k-1)} - \frac{\mu_k}{\mu_{k-2}} \tilde{x}^{(k-2)} + \frac{2\mu_k}{\mu_{k-1}\rho} \cdot g$$

7: **end while**

该算法的每步迭代的整体运算量与原迭代格式基本相当.

6 | 交替方向与 HSS 算法

6.1 多步迭代法

6.2 交替方向法

6.3 HSS 方法

6.1 多步迭代法


设 $A = M_1 - N_1 = M_2 - N_2$ 是 A 的两个矩阵分裂, 则可以构造迭代格式

$$\begin{cases} M_1 x^{(k+\frac{1}{2})} = N_1 x^{(k)} + b, \\ M_2 x^{(k+1)} = N_2 x^{(k+\frac{1}{2})} + b, \end{cases} \quad k = 0, 1, 2, \dots \quad (6.11)$$

这就是 **两步迭代法**, 对应的分裂称为 **二重分裂**

 易知, 两步迭代格式 (6.11) 的迭代矩阵为

$$G = M_2^{-1} N_2 M_1^{-1} N_1 \implies \text{算法收敛的充要条件是 } \rho(M_2^{-1} N_2 M_1^{-1} N_1) < 1$$

 类似地, 我们可以推广到多步迭代法.

6.2 交替方向法

交替方向法 (Alternating Direction Implicit, ADI) 本质上是一个两步迭代法.

设 $A = A_1 + A_2$, 构造矩阵分裂

$$\begin{aligned} A &= (\alpha I + A_1) - (\alpha I - A_2) \\ &= (\alpha I + A_2) - (\alpha I - A_1) \end{aligned}$$

 ADI 迭代格式为

$$\begin{cases} (\alpha I + A_1)x^{(k+\frac{1}{2})} = (\alpha I - A_2)x^{(k)} + b, \\ (\alpha I + A_2)x^{(k+1)} = (\alpha I - A_1)x^{(k+\frac{1}{2})} + b, \end{cases} \quad k = 0, 1, 2, \dots,$$

其中 $\alpha \in \mathbb{R}$ 是迭代参数.

易知 ADI 算法的迭代矩阵为

$$G_{\text{ADI}} = (\alpha I + A_2)^{-1}(\alpha I - A_1)(\alpha I + A_1)^{-1}(\alpha I - A_2).$$

定理 设 $A \in \mathbb{R}^{n \times n}$ 对称正定, $A = A_1 + A_2$, 其中 A_1 和 A_2 中有一个是对称正定, 另一个是对称半正定, 则对任意正数 $\alpha > 0$, 有 $\rho(G_{\text{ADI}}) < 1$, 即 ADI 迭代法收敛.




选取不同的 A_1 和 A_2 可以构造不同的 ADI 格式.

6.3 HSS 方法

设 $A = H + S$, 其中 H 和 S 分别是 A 的对称与斜对称部分, 即

$$H = \frac{1}{2}(A + A^T), \quad S = \frac{1}{2}(A - A^T).$$

该分裂就称为 HS 分裂 (Hermitian and Skew-Hermitian Splitting).

 类似于 ADI 方法, 我们可得下面的 HSS 方法

$$\begin{cases} (\alpha I + H)x^{(k+\frac{1}{2})} = (\alpha I - S)x^{(k)} + b, \\ (\alpha I + S)x^{(k+1)} = (\alpha I - H)x^{(k+\frac{1}{2})} + b, \end{cases} \quad k = 0, 1, 2, \dots$$

定理 设 $A \in \mathbb{R}^{n \times n}$ 正定, 则对任意 $\alpha > 0$, HSS 方法都收敛.

参数 α 的选取

定理 设 $A \in \mathbb{R}^{n \times n}$ 正定, 则极小极大问题

$$\min_{\alpha > 0} \max_{\lambda_{\min}(H) \leq \lambda \leq \lambda_{\max}(H)} \left| \frac{\alpha - \lambda}{\alpha + \lambda} \right|$$

的解为

$$\alpha_* = \sqrt{\lambda_{\max}(H)\lambda_{\min}(H)}.$$

此时

$$\sigma(\alpha_*) = \frac{\sqrt{\lambda_{\max}(H)} - \sqrt{\lambda_{\min}(H)}}{\sqrt{\lambda_{\max}(H)} + \sqrt{\lambda_{\min}(H)}} = \frac{\sqrt{\kappa(H)} - 1}{\sqrt{\kappa(H)} + 1}.$$

HSS 推广

PSS, NSS, AHSS 等, 感兴趣的读者可以参考相关文献.

谢谢
THANK YOU

