第一讲 定常迭代法

目录

线性方程组的数值求解

- **直接法** - LU 分解, Cholesky 分解, ...
- **迭代法**
  - 定常 (经典, 基本) 迭代法: Jacobi, Gauss-Seidel, SOR, SSOR, ...
  - 现代 (Krylov 子空间) 迭代法: CG, MINRES, GMRES, ...
- **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, ...
  - 代数多重网格法 (Algebraic multigrid), ...

> 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.

1-1 定常迭代法

1.1 定常迭代法

迭代法基本思想

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

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

使得

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

1-1-1 矩阵分裂与迭代法

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

$$A = M - N$$

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

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

其中 $G = M^{-1}N$ 称为该迭代格式的 迭代矩阵.

1-1-2 经典迭代法 — Jacobi, G-S, SOR, SSOR, AOR

将矩阵 A 分裂为 $\boxed{A = D - L - U}$, 其中 D 为 A 的对角线部分, $-L$ 和 $-U$ 分别为 A 的严格下三角和严格上三角部分, 即

$$D = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}, \quad L = -\begin{bmatrix} 0 & & & \\ a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix}, \quad U = -\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{bmatrix}.$$

Jacobi 迭代法

Jacobi 迭代法: $\boldsymbol{M = D,\ N = L + U}$

- 迭代格式: $\boxed{x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b}$, $\quad k = 0, 1, \ldots$
- 迭代矩阵: $\boxed{G_{\mathrm{J}} = D^{-1}(L+U)}$
- 分量形式: $\boxed{x_i^{(k+1)} = \dfrac{1}{a_{ii}}\left(b_i - \displaystyle\sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)}$, $\quad i = 1, 2, \ldots, n.$

算法 Jacobi 迭代算法

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

- Jacobi 迭代中 $x_i^{(k+1)}$ 的更新顺序与 i 无关, 因此非常 **适合并行计算**.
- **停机准则**: $$\boxed{\frac{\|b - Ax^{(k)}\|}{\|b - Ax^{(0)}\|} < \mathsf{tol}}$$
- 也可以写为 $$\boxed{x^{(k+1)} = x^{(k)} + D^{-1}(b - Ax^{(k)}) = x^{(k)} + D^{-1} r_k}$$

Gauss-Seidel 迭代法

Gauss-Seidel 迭代法: $\boldsymbol{M = D - L,\ N = U}$

- 迭代格式: $\boxed{x^{(k+1)} = (D-L)^{-1}Ux^{(k)} + (D-L)^{-1}b}, \quad k = 0, 1, \ldots$

- 迭代矩阵: $\boxed{G_{\mathrm{GS}} = (D-L)^{-1}U}$

- 分量形式: $Dx^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + b$

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

算法 Gauss-Seidel 迭代算法

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

- Gauss-Seidel 迭代法充分利用了已经获得的最新数据, 有望获得更快的收敛速度.
- 由于 $x_i^{(k+1)}$ 的更新存在先后关系, 因此不适合并行.
  (在某些应用场景中可以通过重排序实现并行计算)

SOR 迭代法

SOR 迭代法: $M = \frac{1}{\omega}D - L,\ N = \frac{1}{\omega}(1-\omega)D + U$

- 基本思想: 将 G-S 迭代法的 $x^{(k+1)}$ 与 $x^{(k)}$ 加权平均, 以获得 更好 的近似解.
- 迭代格式: $x^{(k+1)} = (1-\omega)x^{(k)} + \omega D^{-1}\left(Lx^{(k+1)} + Ux^{(k)} + b\right)$

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

- 迭代矩阵: $\boxed{G_{\mathrm{SOR}} = (D-\omega L)^{-1}\big((1-\omega)D + \omega U\big)}$
- 分量形式: $\boxed{x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)}$

算法 求解线性方程组的 SOR 迭代方法

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

2: **while** not converge **do** % 停机准则

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**

- 当 $\omega = 1$ 时, SOR 即为 G-S 方法,

  当 $\omega < 1$ 时, 称为**低松弛**方法, 当 $\omega > 1$ 时, 称为**超松弛**方法.

- SOR 的优点是引入了松弛参数 ω, 通过选取适当的 ω 可以大大提高收敛速度. 但确定 **最优** 松弛因子通常是非常困难的!

- SOR 曾在很长一段时间内是求解大规模稀疏线性方程组的首选方法.

SSOR 迭代法

SSOR 迭代法: 两步交替 SOR

- 将 SOR 算法中的 L 和 U 相交换, 交替迭代, 就得到 **SSOR 迭代法**

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

- 迭代格式: $\boxed{x^{(k+1)} = G_{\text{SSOR}}x^{(k)} + g}$, 其中迭代矩阵

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

- 对应的矩阵分裂:

$$M = \frac{1}{\omega(2-\omega)}(D-\omega L)D^{-1}(D-\omega U).$$

AOR 迭代法

AOR 迭代法

- AOR (Accelerated over-relaxation) 由 Hadjidimos 于 1978 年提出.
- 迭代格式: $\boxed{x^{(k+1)} = G_{\mathrm{AOR}} x^{(k)} + g}$, 其中迭代矩阵

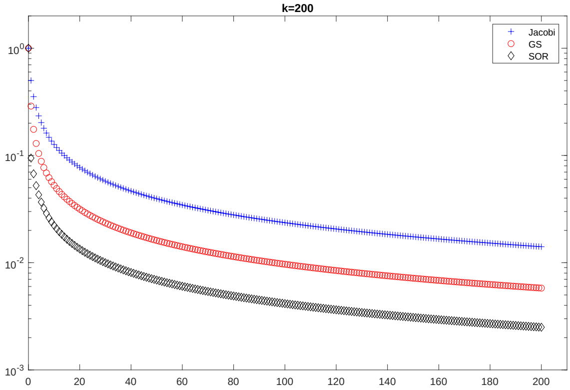
$$\boxed{G_{\mathrm{AOR}} = (D-\gamma L)^{-1}\big((1-\omega)D + (\omega-\gamma)L + \omega U\big)} \qquad (\gamma \text{ 和 } \omega \text{ 为松弛参数})$$

- 对应的矩阵分裂: $M = \dfrac{1}{\omega}(D-\gamma L), \quad N = \dfrac{1}{\omega}\big((1-\omega)D + (\omega-\gamma)L + \omega U\big).$
- 当 $\gamma = \omega$ 时, AOR 即为 SOR;
  当 $\gamma = \omega = 1$ 时, AOR 即为 Gauss-Seidel;
  当 $\gamma = 0,\ \omega = 1$ 时, AOR 即为 Jacobi.

例 编程实践: 用 Jacobi, G-S 和 SOR($\omega = 1.5$) 求解线性方程组 $Ax = b$, 其中

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

初始向量设为 $x^{(0)} = [0, 0, 0]^{\mathsf{T}}$.



1-1-3 Richardson 迭代法

Richardson 迭代法: $\boldsymbol{M = \frac{1}{\omega} I, \quad N = \frac{1}{\omega} I - A}$

- 迭代格式: $\boxed{x^{(k+1)} = x^{(k)} + \omega(b - Ax^{(k)})}, \quad k = 0, 1, 2, \ldots.$
- 迭代矩阵: $\boxed{G_{\mathrm{R}} = I - \omega A}$
- 如果在每次迭代时取不同的参数, 即

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

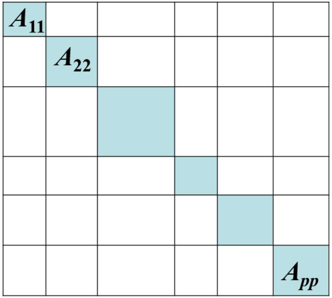
则称为 nonstationary Richardson 算法.

1-1-4 分块迭代法

在实际使用中, 为了**扩大定常迭代法的使用范围**, 同时**提升算法的计算效率**, 通常会采用分块形式进行迭代.

将 A 写成如下的分块形式:

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



分块 Jacobi 迭代

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

分块 Gauss-seidel 迭代

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

分块 SOR 迭代

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

$$i = 1, 2, \ldots, p.$$

第一讲 定常迭代法

目录

1-2 收敛性分析

1.2 收敛性分析

关于向量序列的收敛性及其判别方法见讲义附录.

1-2-1 定常迭代法的收敛性

考虑迭代法

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \triangleq Gx^{(k)} + g$$

定义 (收敛性) 设 $\{x^{(k)}\}_{k=1}^{\infty}$ 是由迭代法生成的向量序列, 若对任意初始向量 $x^{(0)}$, 都有

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

则称迭代法是收敛的, 否则就称其为发散的.

收敛性判断

误差分析

记 $\boxed{\varepsilon^{(k)} \triangleq x^{(k)} - x_*}$, 由 $x^{(k)} = Gx^{(k-1)} + g$ 和 $x_* = Gx_* + g$ 可知

$$\boxed{x^{(k)} - x_* = G\left(x^{(k-1)} - x_*\right) = \cdots = G^k\left(x^{(0)} - x_*\right)} \Longrightarrow \boxed{\|\varepsilon^{(k)}\| \leq \|G\|^k \|\varepsilon^{(0)}\|}$$

收敛性判断

- **充分条件**: 若存在范数 $\|\cdot\|$, 使得 $\|G\| < 1$, 则迭代法收敛.
- **充要条件**: 迭代法收敛的充要条件是 $\rho(G) < 1$.

收敛速度

收敛速度

设 G 是迭代矩阵，则迭代法的**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

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

平均收敛速度与迭代步数和范数有关，但渐进收敛速度只依赖于迭代矩阵谱半径

误差估计

定理 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) $\|x^{(k)} - x_*\| \leq \dfrac{q}{1-q} \|x^{(k)} - x^{(k-1)}\|$;

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

好的迭代法

一个好的迭代法 (基于矩阵分裂的单步定常迭代法) 应该满足:

(1) 以 M 为系数矩阵的线性方程组比较容易求解.

(2) $\rho(G)$ 尽可能小 ($M^{-1}A$ 的特征值尽可能接近于 1, 即 M 接近于 A).

收敛性分析：不可约对角占优情形

收敛性分析：不可约对角占优情形

- 若 A 严格对角占优或不可约对角占优，则 Jacobi 和 Gauss-Seidel 迭代法都收敛.
- 若 A 严格对角占优或不可约对角占优，且 $0 < \omega < 1$，则 SOR 迭代法都收敛.

> 关于对角占优矩阵和不可约矩阵的定义见讲义附录.

SOR 收敛的必要条件

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

$$0 < \omega < 2.$$

收敛性分析：对称正定矩阵情形

收敛性分析：对称正定情形

设 $A \in \mathbb{R}^{n\times n}$ 对称且对角线均为正, 则

- Jacobi 迭代法收敛的充要条件是 A 和 $2D - A$ 均正定.
- Gauss-Seidel 迭代法收敛的充要条件是 A 正定.
- SOR 迭代法收敛的充要条件是 A 正定且 $0 < \omega < 2$.

关于正定矩阵的定义见讲义附录.

Richardson 迭代法收敛性

定理 设 $A \in \mathbb{R}^{n\times n}$ 是对称正定矩阵, λ_1 和 λ_n 分别是 A 的最大和最小特征值, 则 Richardson 算法收敛当且仅当 $\boxed{0 < \omega < \lambda_1^{-1}}$. 最优参数为

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

即当 $\omega = \omega_*$ 时, 迭代矩阵的谱半径达到最小, 且有

$$\rho(G_{\mathrm{R}}) = \begin{cases} 1 - \omega\lambda_n & \text{if } \ \omega \le \omega_* \\ \dfrac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} = \dfrac{\kappa(A) - 1}{\kappa(A) + 1} & \text{if } \ \omega = \omega_* \\ \omega\lambda_1 - 1 & \text{if } \ \omega \ge \omega_*. \end{cases}$$

收敛性：迭代矩阵非负情形

收敛性分析：迭代矩阵非负情形 (Stein-Rosenberg 定理)

设 $A \in \mathbb{R}^{n\times n}$, 若 $G_{\rm J} \geq 0$, 则下面四个结论有且仅有一个成立:

(1) $\rho(G_{\rm GS}) = \rho(G_{\rm J}) = 0$,

(2) $0 < \rho(G_{\rm GS}) < \rho(G_{\rm J}) < 1$,

(3) $\rho(G_{\rm GS}) = \rho(G_{\rm J}) = 1$,

(4) $\rho(G_{\rm GS}) > \rho(G_{\rm J}) > 1$.

这表明, Jacobi 迭代法和 Gauss-Seidel 迭代法此时具有相同的收敛性.

而且当两者都收敛时, Gauss-Seidel 迭代法的收敛速度快于 Jacobi 迭代法.

该结论对一般矩阵并不成立：对某些矩阵, Jacobi 收敛, 但 Gauss-Seidel 却不收敛.

例 设 $A = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}$, 则 Jacobi 和 G-S 的迭代矩阵分别为

$$G_{\mathrm{J}} = \begin{bmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{bmatrix}, \quad G_{\mathrm{GS}} = \begin{bmatrix} 0 & -2 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 2 \end{bmatrix}.$$

直接计算可得

$$\det(\lambda I - G_{\mathrm{J}}) = \lambda^3, \quad \det(\lambda I - G_{\mathrm{GS}}) = \lambda(\lambda - 2)^2.$$

所以 $\rho(G_{\mathrm{J}}) = 0 < 1$, 但 $\rho(G_{\mathrm{GS}}) = 2 > 1$.

因此, Jacobi 迭代法收敛, 但 Gauss-Seidel 迭代法不收敛.

收敛性分析：相容次序矩阵

相容次序矩阵

设 $A \in \mathbb{R}^{n\times n}$ 的对角线元素全不为零，记 $\tilde{L} = D^{-1}L$, $\tilde{U} = D^{-1}U$. 若矩阵 $G(\alpha) = \alpha\tilde{L} + \frac{1}{\alpha}\tilde{U}$ 的特征值与 α 无关，则称 A 是 **相容次序矩阵**.

相容次序矩阵：G_{J} 与 G_{SOR} 的特征值

设 $A \in \mathbb{R}^{n\times n}$ 是相容次序矩阵且 $\omega \neq 0$, 则下列命题成立

(1) G_{J} 的特征值正负成对出现;

(2) 若 μ 是 G_{J} 的特征值且 λ 满足 $\boxed{(\lambda+\omega-1)^2 = \lambda\omega^2\mu^2}$, 则 λ 是 G_{SOR} 的特征值;

(3) 反之，若 $\lambda \neq 0$ 是 G_{SOR} 的特征值且 μ 满足上式，则 μ 是 G_{J} 的特征值.

收敛性：相容次序矩阵情形

设 A 是相容次序矩阵且对角线均非零.

- 我们有 $\rho(G_{\mathrm{GS}}) = \rho(G_{\mathrm{J}})^2$, 即当 Jacobi 收敛时, Gauss-Seidel 也收敛, 且收敛速度快一倍.
- 若 G_{J} 的特征值全部为实数, 则 SOR 收敛的充要条件是 $0 < \omega < 2$ 且 $\rho(G_{\mathrm{J}}) < 1$.

SOR 最优参数

设 A 是相容次序矩阵, G_{J} 的特征值全部为实数, 且 $\rho_{\mathrm{J}} = \rho(G_{\mathrm{J}}) < 1$, 则 SOR 算法的最优参数和对应的谱半径分别为:

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho_{\mathrm{J}}^2}}, \quad \rho(G_{\mathrm{SOR}}) = \omega_{opt} - 1 = \frac{\rho_{\mathrm{J}}^2}{\left(1+\sqrt{1-\rho_{\mathrm{J}}^2}\right)^2}.$$

进一步, 有

$$\rho(G_{\mathrm{SOR}}) = \begin{cases} \omega - 1, & \omega_{opt} \le \omega \le 2 \\ 1-\omega+\frac{1}{2}\omega^2\rho_{\mathrm{J}}^2 + \omega\rho_{\mathrm{J}}\sqrt{1-\omega+\frac{1}{4}\omega^2\rho_{\mathrm{J}}^2}\,, & 0 < \omega \le \omega_{opt} \end{cases}$$

1-2-6 应用举例：二维 Poisson 方程

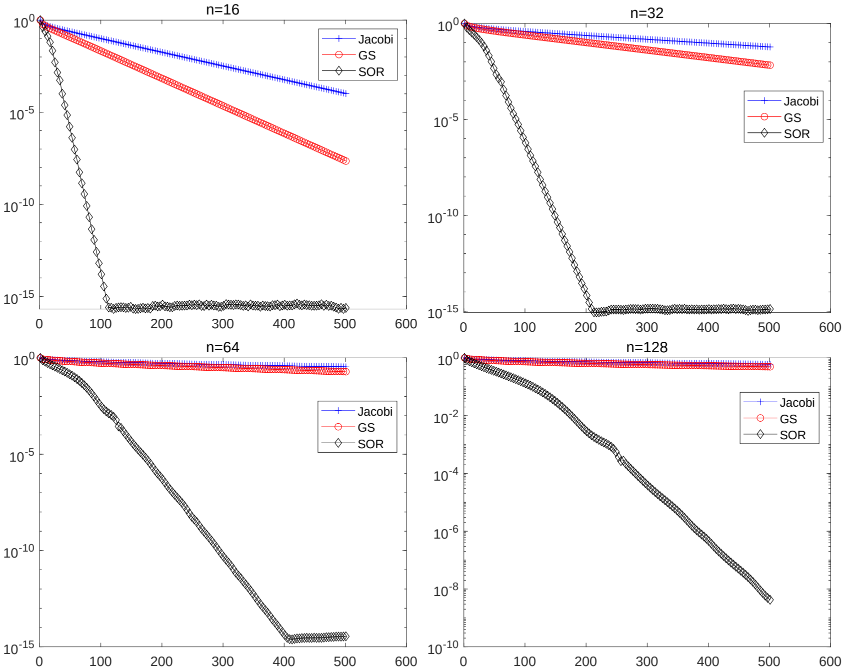
二维 Poisson 方程的五点差分离散

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

经五点差分离散后可得代数方程组 $T_N u = h^2 f$, 其中

$$T_N = I\otimes T_n + T_n\otimes I, \qquad T_n = \mathrm{tridiag}(-1,2,-1).$$

T_N 对称正定且不可约对角占优, 因此 Jacobi, Gauss-Seidel, SOR$(0<\omega<1)$ 都收敛.


n=16
n=32
n=64
n=128
Jacobi
GS
SOR
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
0
100
200
300
400
500
600

第一讲 定常迭代法

目录

1-3 交替方向迭代法

1.3 交替方向迭代法

https://math.ecnu.edu.cn/~jypan/Teaching/IMP

1-3-1 多步分裂迭代法

两步矩阵分裂迭代法

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

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

这就是**两步分裂迭代法**, 对应的分裂称为**两步矩阵分裂**.

两步分裂迭代法的收敛性

易知, 两步分裂迭代法的迭代矩阵为 $G = M_2^{-1} N_2 M_1^{-1} N_1$. 因此, 其收敛的充要条件是 $\rho(M_2^{-1} N_2 M_1^{-1} N_1) < 1$.

多步分裂迭代法

多步分裂迭代法

设 l 是一个正整数, 则 A 的 l 步矩阵分裂为

$$A = M_1 - N_1 = M_2 - N_2 = \cdots = M_l - N_l,$$

相应的 **多步分裂迭代法** 为

$$\begin{cases} M_1 x^{(k+\frac{1}{l})} = N_1 x^{(k)} + b, \\ M_2 x^{(k+\frac{2}{l})} = N_2 x^{(k+\frac{1}{l})} + b, \\ \quad \cdots\ \cdots \\ M_l x^{(k+1)} = N_l x^{(k+\frac{l-1}{l})} + b, \end{cases} \qquad k = 0, 1, 2, \ldots.$$

1-3-2 交替方向法

交替方向法

设 $A = A_1 + A_2$, 则交替方向法 (ADI, Alternating-direction implicit) 为

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

其中 $\alpha \in \mathbb{R}$ 是迭代参数, 迭代矩阵为

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

交替方向法的收敛性

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 $\rho(G_{\mathrm{ADI}}) < 1$, 即 ADI 迭代法收敛.

1-3-3 HSS 方法

HS 分裂与 HSS 迭代法

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

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

该分裂就称为**对称与斜对称分裂** (HSS, Hermitian and Skew-Hermitian Splitting).

类似于 ADI 方法, 我们可得下面的 HSS 方法

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

迭代矩阵: $G_{\mathrm{HSS}}(\alpha) = (\alpha I + S)^{-1}(\alpha I - H)(\alpha I + H)^{-1}(\alpha I - S)$.

收敛性与最优参数 α

通过分析可知

$$\rho(G_{\mathrm{HSS}}) \leq \max_{\lambda\in\sigma(H)} \left|\frac{\alpha-\lambda}{\alpha+\lambda}\right| \triangleq \sigma(\alpha).$$

收敛性与最优参数

设 $A \in \mathbb{R}^{n\times n}$ 正定, 则

(1) 对任意 $\alpha > 0$, HSS 迭代法都收敛.

(2) 极小极大问题 $\boxed{\min_{\alpha>0} \max_{\lambda_{\min}(H)\leq\lambda\leq\lambda_{\max}(H)} \sigma(\alpha)}$ 的解 $\boxed{\alpha_* = \sqrt{\lambda_{\max}(H)\lambda_{\min}(H)}}$

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 的推广

正规与斜对称分裂 (NSS)

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

其中 $A = N + S$, 这里 N 是正规矩阵, S 是斜对称矩阵.

正定与斜对称分裂 (PSS)

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

其中 $A = P + S$, 这里 P 是正定矩阵, S 是斜对称矩阵.

第一讲 定常迭代法

目录

1-4 迭代法的加速

1.4 迭代法的加速

https://math.ecnu.edu.cn/~jypan/Teaching/IMP

出发点

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

1-4-1 外推算法

设原迭代格式为

$$x^{(k+1)} = Gx^{(k)} + b$$

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

$$\boxed{x^{(k+1)} = (1-\omega)x^{(k)} + \omega(Gx^{(k)} + b)},$$

其中 ω 是参数. 这种加速方法就称为**外推算法**.

如何加速

选择 ω 使得迭代矩阵 $\boxed{G_\omega \triangleq (1-\omega)I + \omega G}$ 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, 其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 $\lambda_1 < 1$ 时, $d = 1-\lambda_1$, 当 $\lambda_n > 1$ 时, $d = \lambda_n - 1$.

注记

- 假定原迭代格式收敛, 即 $\rho(G) < 1$, 则当 $\omega_* \neq 1$ 时, 外推迭代比原迭代法收敛要更快.
- 假定原迭代格式不收敛, 则选取合适的参数后, 外推迭代算法仍有可能收敛.
- 最优参数依赖于 G 的特征值, 通常计算比较困难. 在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:

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

定理 (JOR 的收敛性) 设 A 对称正定. 若

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

则 JOR 算法收敛.

1-4-2 Chebyshev 加速方法

出发点

假定已经计算出 $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$, 考虑如何组合得到更精确的近似解.

线性组合

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 $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$ 的一个线性组合, 即

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

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

多项式加速

于是有

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

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

多项式加速

目标: 通过选取适当的参数 α_i, 使得 $\tilde{x}^{(k)} - x_*$ 尽可能地小, 从而使得 $\tilde{x}^{(k)}$ 收敛到 x_* 速度远远快于 $x^{(k)}$ 收敛到 x_* 速度. 这就是 **多项式加速**, 也称为 **半迭代方法**.

例 设 $p_n(t)$ 为 G 的特征多项式, 则 $p_n(G) = 0$, 所以选取 α_i 为 p_n 的系数, 则 $\tilde{x}^{(n)} - x_* = 0$. 但这种选取方法显然是不实用的.

参数 α_i 的实用选取方法

通过分析可知

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

因此参数 α_i 的选取转化为下面的极小化问题

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

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

极小化问题求解

一般来说, 求解非常困难的, 但在一些特殊情况下 (比如 G 对称), 可以给出 (近似) 最优解.

假设迭代矩阵 G 对称, 即 G 存在特征值分解:

$$G = Q\Lambda Q^{\mathsf{T}},$$

其中 Λ 是实对角矩阵, Q 是正交矩阵. 于是有

$$\begin{aligned}
\min_{p\in\mathbb{P}_k, p(1)=1} \|p(G)\|_2 &= \min_{p\in\mathbb{P}_k, p(1)=1} \|p(\Lambda)\|_2 \\
&= \min_{p\in\mathbb{P}_k, p(1)=1} \max_{1\le i\le n} \left\{|p(\lambda_i)|\right\} \\
&\le \min_{p\in\mathbb{P}_k, p(1)=1} \max_{\lambda\in[\lambda_n,\lambda_1]} \left\{|p(\lambda)|\right\},
\end{aligned} \tag{1.1}$$

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 多项式的性质

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

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

关于 Chebyshev 多项式的更多内容参见讲义附录.

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

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

的唯一解为

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

Chebyshev 加速方法

考虑迭代格式 $x^{(k+1)} = Gx^{(k)} + b$, 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 $\boxed{\min_{p\in\mathbb{P}_k, p(1)=1} \max_{\lambda\in[\lambda_n,\lambda_1]} \{|p(\lambda)|\}}$ 的解为

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

具体实施

实际计算时, 是否需要先计算 $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$, 然后通过线性组合

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

来计算 $\tilde{x}^{(k)}$?

事实上, 我们可以通过 Chebyshev 多项式的三项递推公式

$$T_k(t) = 2tT_{k-1}(t) - T_{k-2}(t), \qquad p_k(t) = \frac{T_k(t/\rho)}{T_k(1/\rho)}$$

由 $\tilde{x}^{(k-1)}$ 和 $\tilde{x}^{(k-2)}$ 直接计算出 $\tilde{x}^{(k)}$.

推导过程

令 $\mu_k = \dfrac{1}{T_k(1/\rho)}$, 即 $T_k(1/\rho) = \dfrac{1}{\mu_k}$. 由三项递推公式可得

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

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

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

其中

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

迭代格式 $x^{(k+1)} = Gx^{(k)} + b$ 的 Chebyshev 加速算法.

算法 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: $\quad k = k + 1$

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

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

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.


谢谢
THANK YOU