第五章 线性方程组的迭代法

高 云

方程组系数矩阵的分类

- 低阶稠密矩阵（例如，阶数不超过150）

  （一般用直接法来求解）

- 大型稀疏矩阵（即矩阵阶数高且零元素较多）

  （一般用迭代法来求解）

线性方程组的数值解法分类

- 直接法

  经过有限步算术运算，可求得方程组精确解的方法。

- 迭代法

  用某种极限过程去逐步逼近线性方程组精确解的方法。

迭代法

- 迭代法：从一个初始向量出发，按照一定的迭代格式，构造出一个趋向于真解的无穷序列

  ✓只需存储系数矩阵中的非零元素

  ✓运算量不超过 $O(kn^2)$，其中 k 为迭代步数

迭代解法是目前求解大规模线性方程组的主要方法。

研究内容:

(1) 迭代格式的建立

(2) 收敛性判断

(3) 误差估计和收敛速度

方程组的向量表示形式

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \quad\cdots \qquad\quad \cdots \qquad\qquad \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

$$\boldsymbol{AX = b}$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \qquad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

静态迭代法的基本

- 迭代格式的建立

$$Ax = b \Longleftrightarrow Mx = Nx + b$$

$$A = M - N$$

$$x = M^{-1}Nx + M^{-1}b$$

给定一个初始向量 $x^{(0)}$，可得迭代格式：

$$\boxed{x^{(k+1)} = Bx^{(k)} + g} \qquad k = 0, 1, 2, \ldots$$

其中 B 称为迭代矩阵。

若产生的迭代序列 $\{x^{(k)}\}$ 收敛到一个确定的向量 x^*，则 x^* 就是原方程组的解。

Jacobi 迭代

令 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$， 其中 $\boldsymbol{D}=\operatorname{diag}(a_{11},a_{22},\ldots,a_{nn})$,

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

则可得雅可比 (Jacobi) 迭代格式:

$$\boxed{\boldsymbol{x}^{(k+1)}=\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\boldsymbol{x}^{(k)}+\boldsymbol{D}^{-1}\boldsymbol{b}}\qquad k=0,1,2,\ldots$$

$\boldsymbol{B}_J=\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})$称为雅可比 (Jacobi) 迭代矩阵

Jacobi 迭代

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots \quad \ldots \quad \ldots \quad \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases} \xrightarrow{a_{ii} \neq 0}$$

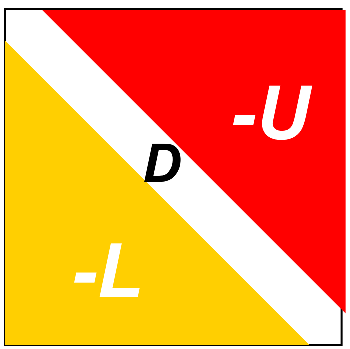
写成矩阵形式:

$$Ax = b \Leftrightarrow (D - (L+U))x = b$$

$$\Leftrightarrow Dx = (L+U)x + b$$

$$\Leftrightarrow x = \underbrace{D^{-1}(L+U)}_{B}x + \underbrace{D^{-1}b}_{\vec{f}}$$

A =



$$x_{k+1} = D^{-1}(L+U)x_k + D^{-1}b$$

Jacobi 迭代阵

$$D=\begin{pmatrix} a_{11} & & & & \\ & a_{22} & & & \\ & & a_{33} & & \\ & & & \ddots & \\ & & & & a_{nn} \end{pmatrix} \quad L=\begin{pmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ -a_{n1} & -a_{n2} & -a_{n3} & \cdots & 0 \end{pmatrix} U=\begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & 0 & \cdots & -a_{3n} \\ & & & \ddots & \vdots \\ & & & & 0 \end{pmatrix}$$

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

$$\begin{pmatrix} a_{11}^{-1} & & & & \\ & a_{22}^{-1} & & & \\ & & a_{33}^{-1} & & \\ & & & \ddots & \\ & & & & a_{nn}^{-1} \end{pmatrix}\left(\begin{pmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ -a_{n1} & -a_{n2} & -a_{n3} & \cdots & 0 \end{pmatrix}+\begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & 0 & \cdots & -a_{3n} \\ & & & \ddots & \vdots \\ & & & & 0 \end{pmatrix}\right)$$

$$B_J = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \cdots & -\frac{a_{2n}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0 & \cdots & -\frac{a_{3n}}{a_{33}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \cdots & 0 \end{pmatrix}$$

$Jacobi$迭代的矩阵形式

$$x_{k+1} = B_J x_k + g$$

其中 $B_J = D^{-1}(L+U)$

Jacobi 迭代的分量形式:

$$x_i^{(k+1)} = -\sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}} \qquad i=1,2,\cdots,n$$

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

Jacobi 迭代的分量形式:

$$\begin{cases} x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)}\right) \big/ a_{11} \\ x_2^{(k+1)} = \left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)}\right) \big/ a_{22} \\ \qquad\vdots \\ x_n^{(k+1)} = \left(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)}\right) \big/ a_{nn} \end{cases}$$

在计算 $x_i^{(k+1)}$ 时，如果用 $x_1^{(k+1)},\cdots,x_{i-1}^{(k+1)}$ 代替 $x_1^{(k)},\cdots,x_{i-1}^{(k)}$，则可能会得到更好的收敛效果。此时的迭代公式为

$$\begin{cases} x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)}\right) \big/ a_{11} \\ x_2^{(k+1)} = \left(b_2 - \boxed{a_{21}x_1^{(k+1)}} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)}\right) \big/ a_{22} \\ \qquad\vdots \\ x_n^{(k+1)} = \left(b_n - \boxed{a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \cdots - a_{n,n-1}x_{n-1}^{(k+1)}}\right) \big/ a_{nn} \end{cases}$$

G-S 迭代

写成矩阵形式:

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

解得

$$\boxed{\boldsymbol{x}^{(k+1)} = \left(\boldsymbol{D}-\boldsymbol{L}\right)^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)} + \left(\boldsymbol{D}-\boldsymbol{L}\right)^{-1}\boldsymbol{b}}$$

$k = 0, 1, 2, \ldots$

此迭代格式称为高斯-塞德尔 (Gauss-Seidel) 迭代

$B_G = \left(\boldsymbol{D}-\boldsymbol{L}\right)^{-1}\boldsymbol{U}$ 称为 GS 迭代矩阵

G-S 迭代公式的推导

作A的另一个分裂： $A=(D-L)-U$

$$Ax=b \Leftrightarrow ((D-L)-U)x=b$$

$$\Leftrightarrow (D-L)x=Ux+b$$

$$\Leftrightarrow x=(D-L)^{-1}Ux+(D-L)^{-1}b$$

$$\Leftrightarrow x_{k+1}=\underbrace{(D-L)^{-1}U}_{B_{G\text{-}S}}x_k+\underbrace{(D-L)^{-1}b}_{g}$$

其迭代格式的矩阵形式为 $x_{k+1}=B_{G-S}x_k+g_{G-S}$

Gauss-Seidel
迭代阵

从另外一个角度来说明

$$(D-L)x^{(k+1)} = Ux^{(k)} + b$$

$$\Leftrightarrow x^{(k+1)} = D^{-1}(Lx^{(k+1)} + Ux^{(k)}) + D^{-1}b$$

写成分量形式:

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

$$x_1^{(k+1)} = \frac{1}{a_{11}}(-a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} - \cdots - a_{1n}x_n^{(k)} + b_1)$$

$$x_2^{(k+1)} = \frac{1}{a_{22}}(-a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - a_{24}x_4^{(k)} - \cdots - a_{2n}x_n^{(k)} + b_2)$$

$$x_3^{(k+1)} = \frac{1}{a_{33}}(-a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - a_{34}x_4^{(k)} - \cdots - a_{3n}x_n^{(k)} + b_3)$$

$$\cdots \quad \cdots \quad \cdots \quad \cdots$$

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

只存一组向量即可。

SOR 迭代

在 GS 迭代中

$$x_i^{(k+1)}=\left(b_i-a_{i1}x_1^{(k+1)}-\cdots-a_{i,i-1}x_{i-1}^{(k+1)}-a_{i,i+1}x_{i+1}^{(k)}-\cdots-a_{i,n}x_n^{(k)}\right)\Big/a_{ii}$$

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

为了得到更好的收敛效果，可在修正项前乘以一个参数 ω，于是就得到所谓的**逐次超松弛迭代法**，简称 **SOR迭代**，其中 ω 称为松弛因子。此时

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

解得

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

$B_S=\left(D-\omega L\right)^{-1}\left[(1-\omega)D+U\right]$称为 SOR 迭代矩阵

Jacobi、GS 和 SOR算法

- **Jacobi 算法**

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

$$\boxed{x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}}$$

- **GS 算法**

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

$$\boxed{x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}}$$

- **SOR 算法**

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

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

举例

- 例：解线性方程组 $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$

$$x^* = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$$

取初始向量 $x^{(0)} = (\,0, 0, 0\,)$，迭代过程中小数点后保留4位。

解：Jacobi 迭代格式

$$\begin{cases} x_1^{(k+1)} = \left(1 + x_2^{(k)}\right)/2 \\ x_2^{(k+1)} = \left(8 + x_1^{(k)} + x_3^{(k)}\right)/3 \\ x_3^{(k+1)} = \left(-5 + x_2^{(k)}\right)/2 \end{cases}$$

令 $x = (x_1, x_2, x_3)^T$ 则迭代得：

$$x^{(1)} = (\,0.5000, 2.6667, -2.5000\,)^T$$

$$\vdots$$

$$x^{(21)} = (\,2.0000, 3.0000, -1.0000\,)^T$$

举例（续）

GS 迭代格式

$$\begin{cases} x_1^{(k+1)} = \left(1 + x_2^{(k)}\right)/2 \\ x_2^{(k+1)} = \left(8 + x_1^{(k+1)} + x_3^{(k)}\right)/3 \\ x_3^{(k+1)} = \left(-5 + x_2^{(k+1)}\right)/2 \end{cases}$$

得 $x^{(1)} = (\ 0.5000,\ 2.8333,\ -1.0833\ )^T$

$$\vdots$$

$x^{(9)} = (\ 2.0000,\ 3.0000,\ -1.0000\ )^T$

举例（续）

SOR 迭代格式

$$\begin{cases} x_1^{(k+1)} = x_1^{(k)} + \omega\left(1 - 2x_1^{(k)} + x_2^{(k)}\right)/2 \\ x_2^{(k+1)} = x_2^{(k)} + \omega\left(8 + x_1^{(k+1)} - 3x_2^{(k)} + x_3^{(k)}\right)/3 \\ x_3^{(k+1)} = x_3^{(k)} + \omega\left(-5 + x_2^{(k+1)} - 2x_3^{(k)}\right)/2 \end{cases}$$

取 ω

如何确定SOR迭代中的最优松弛因子是一件很困难的事。

$$x^{(1)} = (\ 0.5500,\ 3.1350,\ -1.0257\ )^T$$

$$\vdots$$

$$x^{(7)} = (\ 2.0000,\ 3.0000,\ -1.0000\ )^T$$

矩阵分裂法

$$\boxed{A = M - N} \longrightarrow x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b$$

- **Jacobi 迭代** $\boxed{x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b}$

$$\boxed{M = D, \quad N = M - A = L + U}$$

- **GS 迭代** $\boxed{x^{(k+1)} = (D-L)^{-1}Ux^{(k)} + (D-L)^{-1}b}$

$$\boxed{M = D - L, \quad N = U}$$

- **SOR 迭代** $\boxed{x^{(k+1)} = (D-\omega L)^{-1}\left[(1-\omega)D + \omega U\right]x^{(k)} + \omega(D-\omega L)^{-1}b}$

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