$Ax=b$

直接法　　　　迭代法

Gauss-消去法

LU分解法

Gauss 消去法

上述消元过程除第一个方程不变以外，第2—第 n 个方程全消去了变量 χ_1，而系数和常数项全得到新值：

$$\begin{cases} a_{11}^{(1)}x_1+a_{12}^{(1)}x_2+a_{13}^{(1)}x_3+\cdots+a_{1n}^{(1)}x_n=b_1^{(1)} \\ \qquad\qquad a_{22}^{(2)}x_2+a_{23}^{(2)}x_3+\cdots+a_{2n}^{(2)}x_n=b_2^{(2)} \\ \qquad\qquad a_{32}^{(2)}x_2+a_{33}^{(2)}x_3+\cdots+a_{3n}^{(2)}x_n=b_3^{(2)} \\ \qquad\qquad\qquad\cdots\cdots \\ \qquad\qquad a_{n2}^{(2)}x_2+a_{n3}^{(2)}x_3+\cdots+a_{nn}^{(2)}x_n=b_n^{(2)} \end{cases}$$

Gauss 消去法

第 $n-1$ 步消去过程后，得到等价三角方程组。

$$A^{(n)}x = b^{(n)}$$

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \cdots + a_{1n}^{(1)}x_n = b_1^{(1)} \\ \qquad a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} \\ \qquad\qquad a_{33}^{(3)}x_3 + \cdots + a_{3n}^{(3)}x_n = b_3^{(3)} \\ \qquad\qquad\qquad \cdots\cdots \\ \qquad\qquad\qquad\qquad a_{nn}^{(n)}x_n = b_n^{(n)} \end{cases}$$

回代过程

$$\begin{cases} a_{11}^{(1)}x_1+\cdots+a_{1i}^{(1)}x_i+\cdots+a_{1n}^{(1)}x_n=b_1^{(1)} \\ \qquad\qquad \cdots\cdots \\ \qquad\qquad a_{ii}^{(i)}x_i+\cdots+\cdots+a_{in}^{(i)}x_n=b_i^{(i)} \\ \qquad\qquad\qquad \cdots\cdots \\ \qquad\qquad\qquad a_{n-1n-1}^{(n-1)}x_{n-1}+a_{n-1n}^{(n-1)}x_n=b_{n-1}^{(n-1)} \\ \qquad\qquad\qquad\qquad\qquad a_{nn}^{(n)}x_n=b_n^{(n)} \end{cases}$$

$$x_n=b_n^{(n)}\Big/a_{nn}^{(n)}$$

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

LU 分解

$$A = LU$$

$$L = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ & \cdots & \cdots & & \\ l_{n1} & l_{n2} & & l_{n(n-1)} & 1 \end{pmatrix} \qquad \mathrm{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & & u_{1\mathrm{n}} \\ & u_{22} & \cdots & & u_{2\mathrm{n}} \\ & & & & \\ & & \cdots & \cdots & \cdots \\ & & & & u_{\mathrm{nn}} \end{pmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{44} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

先一行后一列. 行知道u
列知道l.

$$
\begin{bmatrix}
u_{11} & u_{12} & u_{13} & u_{14} \\
l_{21}u_{11} & l_{21}u_{12}+u_{22} & l_{21}u_{13}+u_{23} & l_{21}u_{14}+u_{24} \\
l_{31}u_{11} & l_{31}u_{12}+l_{32}u_{22} & l_{31}u_{13}+l_{32}u_{23}+u_{33} & l_{31}u_{14}+l_{32}u_{24}+u_{34} \\
l_{41}u_{11} & l_{41}u_{12}+l_{42}u_{22} & l_{41}u_{13}+l_{42}u_{23}+l_{43}u_{33} & l_{41}u_{14}+l_{42}u_{24}+l_{43}u_{34}+u_{44}
\end{bmatrix}
$$

$u_{11}=a_{11}$, $u_{12}=a_{12}$, $u_{13}=a_{13}$, $u_{14}=a_{14}$

$l_{21}=a_{21}/u_{11}$, $l_{31}=a_{31}/u_{11}$, $l_{41}=a_{41}/u_{11}$

$u_{22}=a_{22}-u_{12}l_{21}$, $u_{23}=a_{23}-l_{21}u_{13}$, $u_{24}=a_{24}-l_{21}u_{14}$

$l_{32}=(a_{32}-l_{31}u_{12})/u_{22}$, $l_{42}=(a_{42}-l_{41}u_{12})/u_{22}$

$a_{32}-l_{31}u_{12}=a_{32}-$

矩阵 L 的对角元素为 1，矩阵 U 的第一行和 A 相同。

步骤:

1. 矩阵L的对角元素为1，矩阵U的第一行和A相同。

2. 迭代，$j=1,2,\cdots n-1$

算L的第j列，$L_{i,j}=\frac{A_{i,j}-\sum_{r=1}^{j-1}L_{i,r}U_{r,j}}{U_{j,j}}, i=j+1,j+2,\cdots,n$

算U的第$j+1$行，$U_{j+1,k}=\frac{A_{j+1,k}-\sum_{r=1}^{j}L_{j+1,r}U_{r,k}}{L_{j+1,j+1}}, k=j+1,j+2$

3. 回代，

$$y_i=b_i-\sum_{j=1}^{i-1}L_{i,j}y_j, i=1,2,\cdots,n$$

$$x_i=\frac{y_i-\sum_{j=i+1}^{n}x_j\cdot U_{i,j}}{U_{i,i}},\ i=n,n-1,\cdots,1$$

1. 对称正定矩阵的Cholesky分解

- 对称正定矩阵：（1） $A = A^T$

（2） $\boldsymbol{x}^T A\boldsymbol{x} > 0 \quad (\boldsymbol{x} \neq 0)$

（2′） $$A_k = \begin{vmatrix} a_{11} & \cdots & a_{1k} \\ \cdots & \cdots & \cdots \\ a_{k1} & \cdots & a_{kk} \end{vmatrix} > 0 \quad (k = 1,2,\ldots n)$$

A的各阶顺序主子式均大于零

$$A=\begin{bmatrix}4 & -1 & 1\\ -1 & 2 & -2\\ 1 & -2 & 3\end{bmatrix} \qquad x=\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}\neq 0 \qquad x^T=\left(x_1,x_2,x_3\right)$$

$$x^T Ax=\left(x_1,x_2,x_3\right)\begin{bmatrix}4x_1-x_2+x_3\\ -x_1+2x_2-2x_3\\ x_1-2x_2+3x_3\end{bmatrix}$$

$$= \quad (4x_1^2-x_1x_2+x_1x_3)+(-x_1x_2+2x_2^2-2x_2x_3)+(x_3x_1-2x_3x_2+3x_3^2)$$

$$=4x_1^2+2x_2^2+3x_3^2-2x_1x_2+2x_1x_3-4x_2x_3$$

$$=2x_1^2+(x_1-x_2+x_3)^2+(x_2-x_3)^2+x_3^2$$

$$=\begin{cases}>0, & x_3\neq 0\\ x_1^2+(x_1^2+x_2^2)+(x_1-x_2)^2>0 & x_3=0\end{cases}$$

对角占优：① 至少有一行对角线的 |值| > ∑| |

② 每行的对角线的 |值| ≥ ∑| 其它 |

$$
\begin{bmatrix}
2 & -1 & & & & & & \\
-1 & 2 & -1 & & & & & \\
 & -1 & 2 & -1 & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & -1 & 2 & -1 & & \\
 & & & & \ddots & \ddots & \ddots & \\
 & & & & & -1 & 2 & -1 \\
 & & & & & & -1 & 2
\end{bmatrix}
$$

由法国炮兵军官André-Louis Cholesky(1875-1918)所提出，当初是为了解决测地计算问题。

第一次世界大战结束前的几个月，Cholsky在一次战斗中牺牲。

该方法在其死后由后任军官发表

Cholesky分解

$$A = LU_1$$

$$A^T = (LU_1)^T = U_1^T L^T$$

$$A = A^T \longrightarrow A = U_1^T L^T$$

下三角矩阵

$$U_1^T = \begin{bmatrix} u_{11} & & & \\ u_{12} & u_{22} & & \\ u_{13} & u_{23} & u_{33} & \\ u_{14} & u_{24} & u_{34} & u_{44} \end{bmatrix}$$

$$U_1^T = \begin{bmatrix} u_{11} & & & \\ u_{12} & u_{22} & & \\ u_{13} & u_{23} & u_{33} & \\ u_{14} & u_{24} & u_{34} & u_{44} \end{bmatrix} \longrightarrow$$

$$\underbrace{\begin{bmatrix} 1 & & & \\ u_{12}/u_{11} & 1 & & \\ u_{13}/u_{11} & u_{23}/u_{22} & 1 & \\ u_{14}/u_{11} & u_{24}/u_{22} & u_{34}/u_{33} & 1 \end{bmatrix}}_{U^T} \underbrace{\begin{bmatrix} u_{11} & & & \\ & u_{22} & & \\ & & u_{33} & \\ & & & u_{44} \end{bmatrix}}_{D}$$

$$U_1^T = U^T D \Rightarrow U_1 = DU$$

$$U_1 = DU$$

$$A = LU_1 = LDU$$

$$A^T = (LDU)^T = U^T D L^T$$

$$\Rightarrow L = U^T$$

$$\Rightarrow A = LDL^T = L\sqrt{D}\cdot\sqrt{D}L^T = (L\sqrt{D})\cdot(L\sqrt{D})^T$$

$$D > 0 \longrightarrow D = \left(\sqrt{D}\right)^2$$

$$\sqrt{D} = \begin{pmatrix} \sqrt{d_1} & & & \\ & \sqrt{d_2} & & \\ & & \sqrt{d_3} & \\ & & & \sqrt{d_4} \end{pmatrix}$$

$$A = \tilde{L}\tilde{L}^T \qquad \tilde{L} = L\sqrt{D}$$

Cholesky分解的求法

设A对称正定 ,则$A = LL^T$

令$L = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \cdots & \cdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$

如何求 l_{ij}?以$n = 3$为例。

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

Cholesky分解的求法

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

先行后对角线

$$= \begin{bmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & l_{21}l_{31} + l_{22}l_{32} \\ l_{11}l_{31} & l_{21}l_{31} + l_{22}l_{32} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{bmatrix}$$

$k=1$时：由 $a_{11}=l_{11}^2$，得 $l_{11}=\sqrt{a_{11}}$；

由 $a_{21}=l_{21}l_{11}$，得 $l_{21}=\dfrac{a_{21}}{l_{11}}$；同理得 $l_{31}=\dfrac{a_{31}}{l_{11}}$。

$k=2$时：由 $a_{22}=l_{21}^2+l_{22}^2$，得 $l_{22}=\sqrt{a_{22}-l_{21}^2}$；

由 $a_{32}=l_{31}l_{21}+l_{32}l_{22}$，得 $l_{32}=\dfrac{a_{32}-l_{31}l_{21}}{l_{22}}$。

$k=3$时：由 $a_{33}=l_{31}^2+l_{32}^2+l_{33}^2$，得 $l_{33}=\sqrt{a_{33}-\sum_{i=1}^{2}l_{3i}^2}$

推广到 n 阶矩阵,有

$$\begin{cases} l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2\right)^{\frac{1}{2}} \\ l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}\right) / l_{jj} \end{cases} \qquad j = 1,\ 2,\ \ldots,\ n;\quad i = j+1,\ \ldots,\ n$$

用$Cholesky$分解法解线性方程组

$$AX=b \Leftrightarrow \begin{cases} Ly=b \\ L^T x=y \end{cases} \quad 其中A=LL^T$$

- Cholesky分解法缺点及优点
- 优点：可以减少存储单元。
- 缺点：存在开方运算

改进的cholesky分解$A=LDL^T$

$$L=\begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \dots & \dots & \dots & \ddots & \\ l_{n1} & l_{n2} & \dots & l_{nn-1} & 1 \end{bmatrix} \qquad D=\begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & d_{nn} \end{bmatrix}$$

由 $A=L(DL^T)$

$$=\begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \dots & \dots & \dots & \ddots & \\ l_{n1} & l_{n2} & \dots & l_{nn-1} & 1 \end{bmatrix}\begin{bmatrix} d_1 & d_1 l_{21} & d_1 l_{31} & \dots & d_1 l_{n1} \\ & d_2 & d_2 l_{32} & \dots & d_2 l_{n2} \\ & & d_3 & \dots & d_3 l_{n3} \\ \dots & \dots & \dots & \ddots & \\ & & & & d_n \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & & \\ l_{21} & 1 & \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & d_1 l_{21} & d_1 l_{31} \\ & d_2 & d_2 l_{32} \\ & & d_3 \end{bmatrix}$$

$$= \begin{bmatrix} d_1 & d_1 l_{21} & d_1 l_{31} \\ d_1 l_{21} & d_1 l_{21}^2 + d_2 & d_1 l_{21} l_{31} + d_2 l_{32} \\ d_1 l_{31} & d_1 l_{21} l_{31} + d_2 l_{32} & d_1 l_{31}^2 + d_2 l_{32}^2 + d_3 \end{bmatrix}$$

逐行相乘，并注意到$i > j$有

$$a_{ij} = \sum_{k=1}^{j-1} l_{ik} d_k l_{jk} + l_{ij} d_j \qquad (j = 1, 2, .., i-1)$$

$$a_{ii} = \sum_{k=1}^{j-1} l_{ik}^2 d_k + d_i \qquad (i = 1, 2, ..., n)$$

由此可得

$$\begin{cases} d_i = a_{ii} - \sum_{k=1}^{j-1} l_{ik}^2 d_k \quad i = 1, 2, ..., n \\ l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}) / d_j \quad (j = 1, 2, .., i-1) \end{cases}$$

为减少计算量,可令$c_{ij}=l_{ij}d_j$，则$l_{ij}=\dfrac{c_{ij}}{d_j}$

所以可将上述公式改写成

$$
\begin{cases}
c_{ij}=a_{ij}-\sum\limits_{k=1}^{j-1}c_{ik}l_{jk} \\
l_{ij}=\dfrac{c_{ij}}{d_j} \qquad\qquad (i=2,3,\ldots,n,\ \ j=1,2,\ldots,i-1) \\
d_i=a_{ii}-\sum\limits_{k=1}^{i-1}c_{ik}l_{ik}
\end{cases}
$$

用 *Cholesky* 分解法解线性方程组 $Ax=b$

即等价于求 $\begin{cases} Ly=b \\ L^T x = D^{-1} y \end{cases}$

$$D^{-1}=\begin{bmatrix} \frac{1}{d_1} & & & \\ & \frac{1}{d_2} & & \\ & & \ddots & \\ & & & \frac{1}{d_n} \end{bmatrix}$$

故 $D^{-1}y=(\frac{y_1}{d_1},\frac{y_2}{d_2},\ldots,\frac{y_n}{d_n})^T$

例题

例　　　试用改进的$Choleskey$分解算法解方程组

$$\begin{bmatrix} 4 & -1 & 1 \\ -1 & 2 & -2 \\ 1 & -2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \\ 6 \end{bmatrix}$$

$$A=\begin{bmatrix}4 & -1 & 1\\ -1 & 2 & -2\\ 1 & -2 & 3\end{bmatrix}=\begin{bmatrix}1 & & \\ -0.25 & 1 & \\ 0.25 & -1 & 1\end{bmatrix}\begin{bmatrix}4 & & \\ & 1.75 & \\ & & 1\end{bmatrix}\begin{bmatrix}1 & -0.25 & 0.25\\ & 1 & -1\\ & & 1\end{bmatrix}=LDL^T$$

由Ly=b得

$$\begin{bmatrix} 1 & & \\ -0.25 & 1 & \\ 0.25 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \\ 6 \end{bmatrix}$$

得 $y_1 = 5, y_2 = -1.75, y_3 = 3$

即 $y = (5, \ -\dfrac{7}{4}, \ 3)^T$

而 $D^{-1}y=(\frac{5}{4},-1,3)^T$，由 $L^Tx=D^{-1}y$

得 $$\begin{bmatrix}1 & -0.25 & 0.25\\ & 1 & -1\\ & & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1.25\\-1\\3\end{bmatrix}$$

得 $x_3=3, x_2=2, x_1=1$

所以方程的解： $x=(1,2,3)^T$

2. 三对角矩阵的追赶法

设三对角矩阵$A=(a_{ij})_{n\times n}$，对一切$|i-j|>1$有$a_{ij}=0$,

即

$$A=\begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{bmatrix}$$

LU 分解：

$$\overset{A}{\begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{bmatrix}} = \overset{L}{\begin{bmatrix} 1 & & & & \\ p_2 & 1 & & & \\ & p_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & p_n & 1 \end{bmatrix}} \overset{U}{\begin{bmatrix} q_1 & c_1 & & & \\ & q_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & q_{n-1} & c_{n-1} \\ & & & & q_n \end{bmatrix}}$$

$$= \begin{bmatrix} q_1 & c_1 & & & & \\ p_2 q_1 & p_2 c_1 + q_2 & c_2 & & & \\ & p_3 q_2 & p_3 c_2 + q_3 & c_3 & & \\ & & & & c_{n-1} & \\ & & & p_n q_{n-1} & p_n c_{n-1} + q_n & \end{bmatrix}$$

$$\begin{cases} q_1 = a_1 \\ p_i = \dfrac{b_i}{q_{i-1}} \\ q_i = a_i - p_i c_{i-1} \end{cases}$$

$$= a_i - \frac{b_i}{q_{i-1}} c_{i-1}$$

$LUX=f \quad UX=y$

则求Ax=f等价于求$\begin{cases} Ly = f \\ Ux = y \end{cases}$

其中$f=(b_1, f_2, \dots, f_n)^T$，故有

$$\begin{bmatrix} 1 & & & & \\ p_2 & 1 & & & \\ & p_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & p_n & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{bmatrix}$$

解得 $\begin{cases} y_1 = f_1 \\ y_i = f_i - p_i y_{i-1} \end{cases} (i=2,\dots,n)$

再由
$$
\begin{bmatrix} q_1 & c_1 & & & \\ & q_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & q_{n-1} & c_{n-1} \\ & & & & q_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}
$$

解得
$$
\begin{cases} x_n = \dfrac{y_n}{q_n} \\ x_i = \dfrac{y_i - c_i x_{i+1}}{q_i} \quad (i = n-1, \ldots, 1) \end{cases}
$$

以上称为解三对角方程组的追赶法。

$$Ax=b$$

直接法　　　　迭代法

简单的重复

迭代法

直接法：适用于阶数不高的线性方程组。

实际应用中，常会遇到一类阶数很高，非零元素很少的所谓高阶稀疏方程组。对这类方程组用迭代法求解，可以充分利用稀疏矩阵的特性减少计算工作量，节省存贮量。

迭代法所要解决的几个主要问题是:

(1) 构造一种迭代格式，把所给方程组

$$\boldsymbol{Ax = b \to x = Bx + d}$$

迭代公式

$$x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)})^T \in R^n$$

$$x^{(k+1)} = Bx^{(k)} + d$$

(2) 迭代矩阵B满足什么条件时，迭代序列收敛于Ax=b的精确解。

(3) 讨论如何估计误差的大小以决定迭代次数

$$e^{(k)} = x^{*} - x^{(k)}$$

Jacobi 迭代法

$f(x)=0$

$\Downarrow$

$ax+b=0$

Jacobi 迭代法是最简单的一种迭代法

$x_1 = 1/a_{11}[b_1-(a_{12}x_2+\cdots+a_{12}x_n)]$

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

最容易求解：

对角矩阵.

$a_{nn}x_n = b_n-(a_{n1}x_1+\cdots+a_{nn-1}x_{n-1})$

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

Jacobi method

From Wikipedia, the free encyclopedia

In numerical linear algebra, the **Jacobi method** (or **Jacobi iterative method**[1]) is an algorithm for determining the solutions of a diagonally dominant system of linear equations. Each diagonal element is solved for, and an approximate value is plugged in. The process is then iterated until it converges. This algorithm is a stripped-down version of the Jacobi transformation method of matrix diagonalization. The method is named after Carl Gustav Jacob Jacobi.

Carl Gustav Jacob Jacobi

From Wikipedia, the free encyclopedia

Carl Gustav Jacob Jacobi (/dʒəˈkoʊbi/;[1] German: [jaˈkoːbi]; 10 December 1804 – 18 February 1851) was a German mathematician, who made fundamental contributions to elliptic functions, dynamics, differential equations, and number theory. His name is occasionally written as **Carolus Gustavus Iacobus Iacobi** in his Latin books, and his first name is sometimes given as Karl.

Jacobi was the first Jewish mathematician to be appointed professor at a German university.[2]

Jacobi（1804～1851），出生于德国 Potsdam，卒于柏林。他对数学主要的贡献是在椭圆函数及椭圆积分上，并把这些理论应用在数论上而得到很好的结果。雅可比很早就展现了他的数学天份。他从欧拉及 Lagrange 的著作中学习代数及微积分，并被吸引到数论的领域。他处理代数问题的手腕只有欧拉与印度的 Ramanujan 可以相提并论。

Jacobi 少 Abel 两岁。他不知道 Abel 从1820年起就在作五次式的问题，他也去作，但是没有完满的结果。年轻的时候，Jacobi 有许多发现都跟高斯的结果重叠，但高斯并没有发表这些结果。高斯很看重雅可比，1839年 Jacobi 还去拜访了高斯。1849年45岁时，除了高斯之外，Jacobi 已经是欧洲最有名的数学家了。

复数函数（单变量）是十九世纪的一个大领域。高斯已经证明了：要解一个代数方程，我们必需要复数，而这也是充分的。是否还有其它的「数」呢？ 椭圆函数理论是与复变函数论互为补充的理论。椭圆函数的一个主宰性质是他的双周期性，1825年被 Abel 发现的。

Jacobi 应用椭圆函数论到整数论的问题上，他证明了 Fermat 宣称的：每个整数 1, 2, 3, ... 都可以写成整数（包含 0）的平方和，而且他还能算出共有几种方法。当 n 为奇时，有 n 的所有因子（包括 1 及 n）之和的 8 倍个方法；当 n 为偶时，有 n 的所有奇因子之和的 24 倍个方法。

他在数学物理上也有番建树，在量子力学中他的 Hamilton-Jacobi 方程扮演了一个革命性的角色。

Carl Gustav Jacob Jacobi

Jacobi 迭代公式

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right) \qquad (i = 1, 2, \cdots, n; \qquad k = 0, 1, 2, \cdots)$$

终止条件： $\| x^{(k+1)} - x^{(k)} \| < \varepsilon$

Jacobi迭代方法的矩阵形式

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}
= \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 \end{bmatrix}}_{\boldsymbol{L}}
+ \underbrace{\begin{bmatrix} a_{11} & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \end{bmatrix}}_{\boldsymbol{D}}
+ \underbrace{\begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} \\ 0 & 0 & a_{23} & a_{24} \\ 0 & 0 & 0 & a_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{U}}
$$

$$A = L + D + U$$

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

$$\Rightarrow Dx = -(L + U)x + b$$

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

$$= Bx + f$$

$$x^{(k+1)} = Bx^{(k)} + f$$

Jacobi迭代矩阵

- 例： 用Jacobi迭代法求解方程组

$$\begin{cases} 10x_1 - 2x_2 - x_3 = 3 \\ -2x_1 + 10x_2 - x_3 = 15 \\ -x_1 - 2x_2 + 5x_3 = 10 \end{cases}$$

解： 从原方程组中分别解出 $x_i (i = 1, 2, 3)$

$$\begin{cases} x_1 = \dfrac{1}{10}(3 + 2x_2 + x_3) \\ x_2 = \dfrac{1}{10}(15 + 2x_1 + x_3) \\ x_3 = \dfrac{1}{\underset{5}{\cancel{10}}}(10 + x_1 + 2x_2) \end{cases}$$

- 因此得迭代格式

$$\begin{cases} x_1^{(k+1)} = 0.3 + 0.2x_2^{(k)} + 0.1x_3^{(k)} \\ x_2^{(k+1)} = 1.5 + 0.2x_1^{(k)} + 0.1x_3^{(k)} \\ x_3^{(k+1)} = 2 + 0.2x_1^{(k)} + 0.4x_2^{(k)} \end{cases}$$

K=1,2,3, …。若取初始向量 $x^{(0)} = (0,0,0)^T$

计算所得向量列于表中

$$x^* = (1,2,3)^T$$

K	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\|x^{(k)} - x^*\|_\infty$
0	0	0	0	
1	0.3000	1.5000	2.0000	1.0000
2	0.8000	1.7600	2.6600	0.3600
3	0.9180	1.9260	2.8640	0.1360
4	0.9716	1.9700	2.9540	0.0460
5	0.9894	1.9897	2.9823	0.0177
6	0.9962	1.9961	2.9938	0.0062
7	0.9986	1.9986	2.9977	0.0023
8	0.9995	1.9995	2.9992	0.0008
9	0.9998	1.9998	2.9997	0.0003

最大误差

改进

~~一般~~地，$x^{(k+1)}$ 比 $x^{(k)}$ 更接近准确解

$$
\begin{cases}
x_1^{(k+1)} = \dfrac{1}{a_{11}}\left[b_1 - (a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + a_{14}x_4^{(k)} + \ldots + a_{1n}x_n^{(k)})\right] \\
x_2^{(k+1)} = \dfrac{1}{a_{22}}\left[b_2 - (a_{21}x_1^{(k)} + a_{23}x_3^{(k)} + a_{24}x_4^{(k)} + \ldots + a_{2n}x_n^{(k)})\right] \\
x_3^{(k+1)} = \dfrac{1}{a_{33}}\left[b_3 - (a_{31}x_1^{(k)} + a_{32}x_2^{(k)} + a_{34}x_4^{(k)} + \ldots + a_{3n}x_n^{(k)})\right] \\
\qquad \cdots \\
x_n^{(k+1)} = \dfrac{1}{a_{nn}}\left[b_n - (a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + a_{n3}x_3^{(k)} + \ldots + a_{n(n-1)}x_{n-1}^{(k)})\right]
\end{cases}
$$

左大屏
大屏主机

一般地，$x^{(k+1)}$ 比 $x^{(k)}$ 更接近准确解

$$
\begin{cases}
x_1^{(k+1)} = \dfrac{1}{a_{11}}\left[b_1 - (a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + a_{14}x_4^{(k)} + \ldots + a_{1n}x_n^{(k)})\right] \\
x_2^{(k+1)} = \dfrac{1}{a_{22}}\left[b_2 - (a_{21}x_1^{(k)} + a_{23}x_3^{(k)} + a_{24}x_4^{(k)} + \ldots + a_{2n}x_n^{(k)})\right] \\
x_3^{(k+1)} = \dfrac{1}{a_{33}}\left[b_3 - (a_{31}x_1^{(k)} + a_{32}x_2^{(k)} + a_{34}x_4^{(k)} + \ldots + a_{3n}x_n^{(k)})\right] \\
\cdots \\
x_n^{(k+1)} = \dfrac{1}{a_{nn}}\left[b_n - (a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + a_{n3}x_3^{(k)} + \ldots + a_{n(n-1)}x_{n-1}^{(k)})\right]
\end{cases}
$$

$x_1^{(k+1)}$

$x_1^{(k+1)}$ $x_2^{(k+1)}$

$x_1^{(k+1)}$ $x_2^{(k+1)}$ $x_3^{(k+1)}$ $x_4^{(k+1)}$

$$
\begin{cases}
x_1^{(k+1)} = \dfrac{1}{a_{11}}\left[b_1 - (a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + a_{14}x_4^{(k)} + \ldots + a_{1n}x_n^{(k)})\right] \\
x_2^{(k+1)} = \dfrac{1}{a_{22}}\left[b_2 - (a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)} + a_{24}x_4^{(k)} + \ldots + a_{2n}x_n^{(k)})\right] \\
x_3^{(k+1)} = \dfrac{1}{a_{33}}\left[b_3 - (a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{34}x_4^{(k)} + \ldots + a_{3n}x_n^{(k)})\right] \\
\qquad\qquad \cdots \\
x_n^{(k+1)} = \dfrac{1}{a_{nn}}\left[b_n - (a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + a_{n3}x_3^{(k+1)} + \ldots + a_{n(n-1)}x_{n-1}^{(k+1)})\right]
\end{cases}
$$

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

Gauss-Seidel迭代

GS迭代方法的矩阵形式

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

$$\begin{cases}a_{11}x_1^{(k+1)}=\left[b_1-(a_{12}x_2^{(k)}+a_{13}x_3^{(k)}+a_{14}x_4^{(k)}+\ldots+a_{1n}x_n^{(k)})\right]\\ a_{21}x_1^{(k+1)}+a_{22}x_2^{(k+1)}=\left[b_2-(a_{23}x_3^{(k)}+a_{24}x_4^{(k)}+\ldots+a_{2n}x_n^{(k)})\right]\\ a_{31}x_1^{(k+1)}+a_{32}x_2^{(k+1)}+a_{33}x_3^{(k+1)}=\left[b_3-(a_{34}x_4^{(k)}+\ldots+a_{3n}x_n^{(k)})\right]\\ \cdots\\ a_{n1}x_1^{(k+1)}+a_{n2}x_2^{(k+1)}+a_{n3}x_3^{(k+1)}+\ldots+a_{n(n-1)}x_{n-1}^{(k+1)}+a_{nn}x_n^{(k+1)}=b_n\end{cases}$$

$$\begin{cases}a_{11}x_1^{(k+1)}=\left[b_1-(a_{12}x_2^{(k)}+a_{13}x_3^{(k)}+a_{14}x_4^{(k)}+\ldots+a_{1n}x_n^{(k)})\right]\\ a_{21}x_1^{(k+1)}+a_{22}x_2^{(k+1)}=\left[b_2-(a_{23}x_3^{(k)}+a_{24}x_4^{(k)}+\ldots+a_{2n}x_n^{(k)})\right]\\ a_{31}x_1^{(k+1)}+a_{32}x_2^{(k+1)}+a_{33}x_3^{(k+1)}=\left[b_3-(a_{34}x_4^{(k)}+\ldots+a_{3n}x_n^{(k)})\right]\\ \cdots\\ a_{n1}x_1^{(k+1)}+a_{n2}x_2^{(k+1)}+a_{n3}x_3^{(k+1)}+\ldots+a_{n(n-1)}x_{n-1}^{(k+1)}+a_{nn}x_n^{(k+1)}=b_n\end{cases}$$

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

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

小组屏01

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

Jacobi迭代矩阵

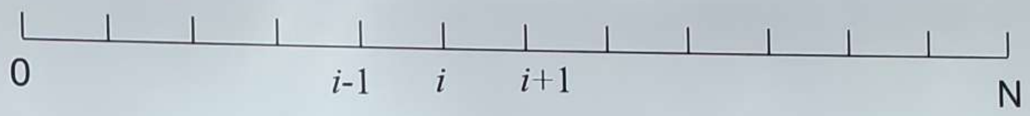
$$\boldsymbol{x}^{(k+1)} = -\boxed{(L+D)^{-1}}U\boldsymbol{x}^{(k)} + \boxed{(L+D)^{-1}}b$$

GS迭代矩阵

由于利用了最新的信息(或者说，迭代矩阵更接近原矩阵的逆)

GS迭代比Jacobi迭代收敛更快（一般）

小组屏01



$$T_{i-1} - 2T_i + T_{i+1} = 0$$

Jacobi迭代：

$$T_i^{(k+1)} = \frac{1}{2}\left(T_{i-1}^{(k)} + T_{i+1}^{(k)}\right) \qquad i=1, 2, \ldots, N\text{-}1$$

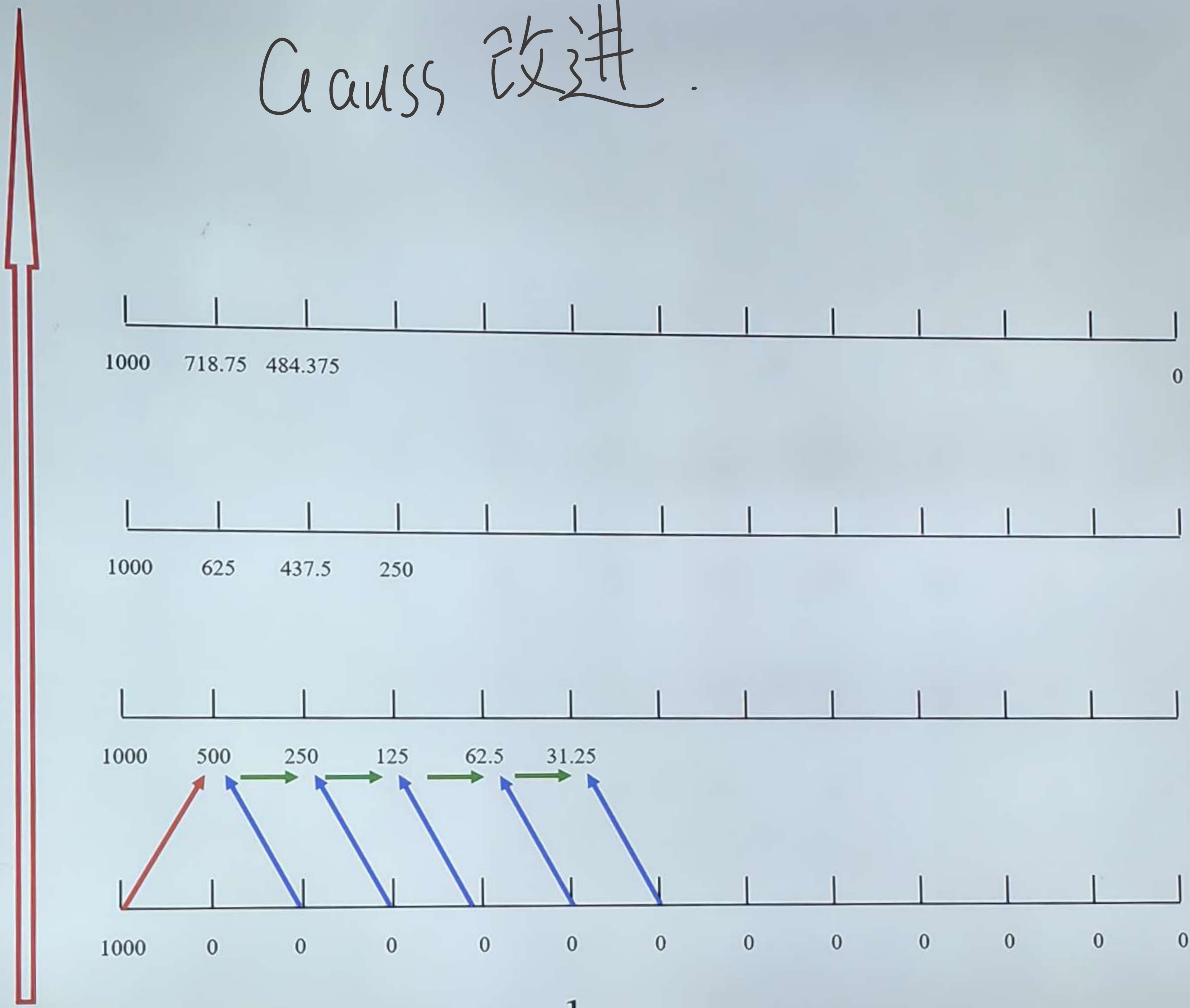
GS迭代：

$$T_i^{(k+1)} = \frac{1}{2}\left(T_{i-1}^{(k+1)} + T_{i+1}^{(k)}\right) \qquad i=1, 2, \ldots, N\text{-}1$$

边界条件

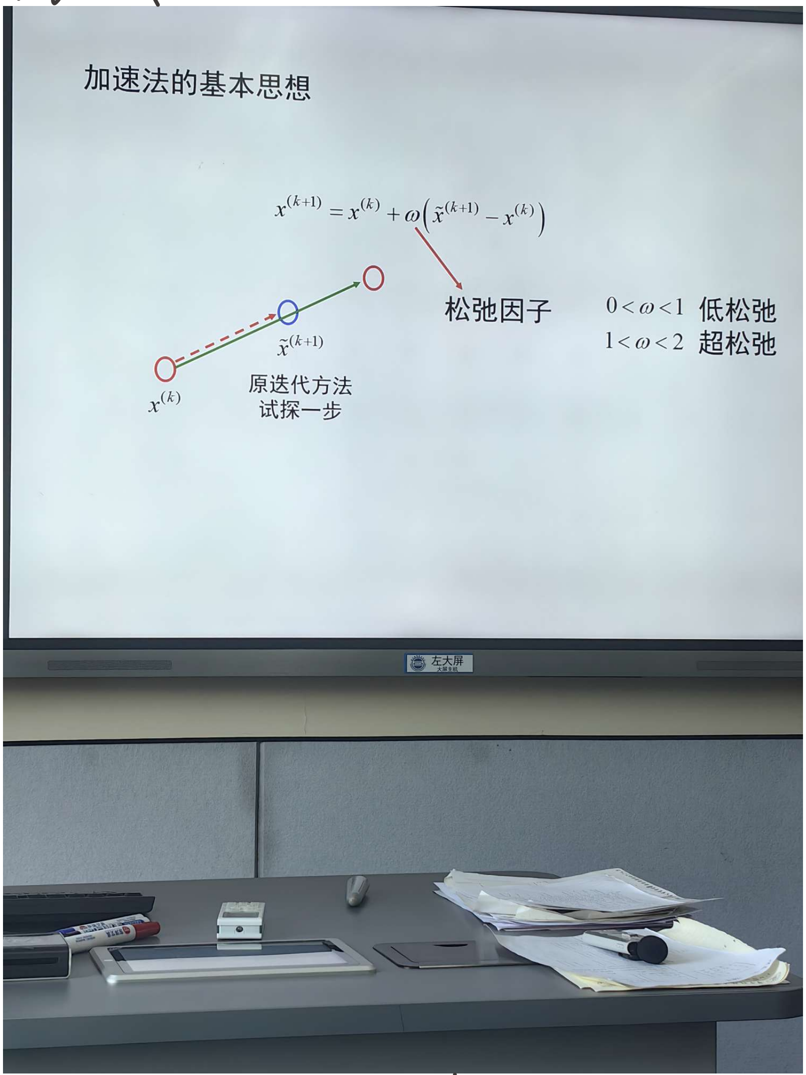
$$T_0 = 1000 \qquad T_N = 0$$

Gauss 改进.



$$T_i^{(k+1)} = \frac{1}{2}\left(T_{i-1}^{(k+1)} + T_{i+1}^{(k)}\right)$$

迭代方法的改造:



Jacbi

$$AX=b \Leftrightarrow DX^{(k+1)}=b-(L+U)X^{(k)}.$$

Gauss

$$AX=b \Leftrightarrow (L+D)X^{(k+1)}=b-UX^{(k)}$$

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

$$(D-wL)x^{(k+1)}=((1-w)D+wU)x^{(k)}+wb$$

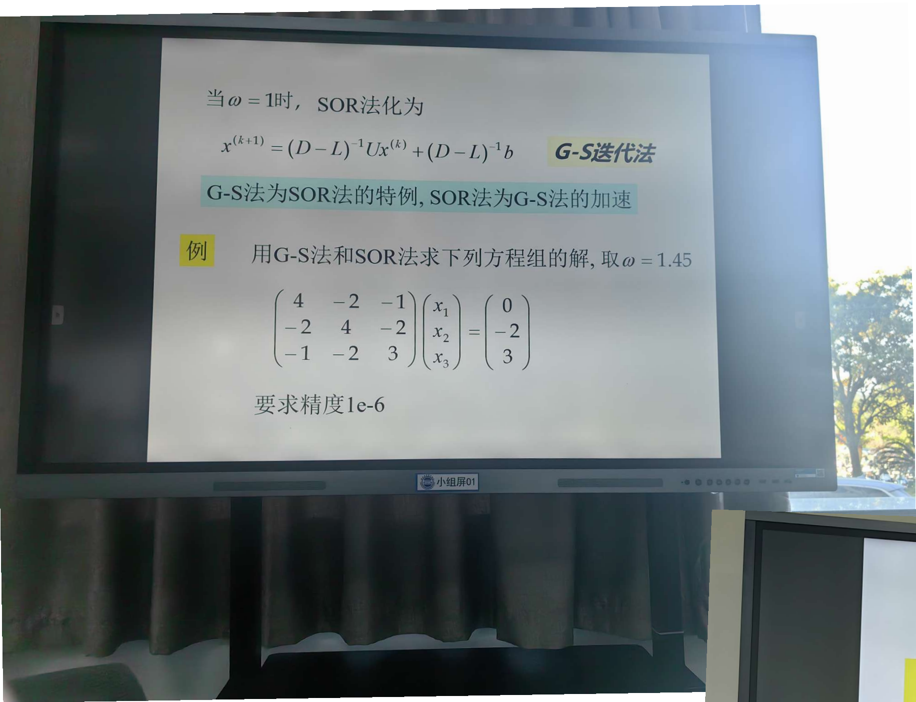
令 $B_w=(D-wL)^{-1}((1-w)D+wU)$

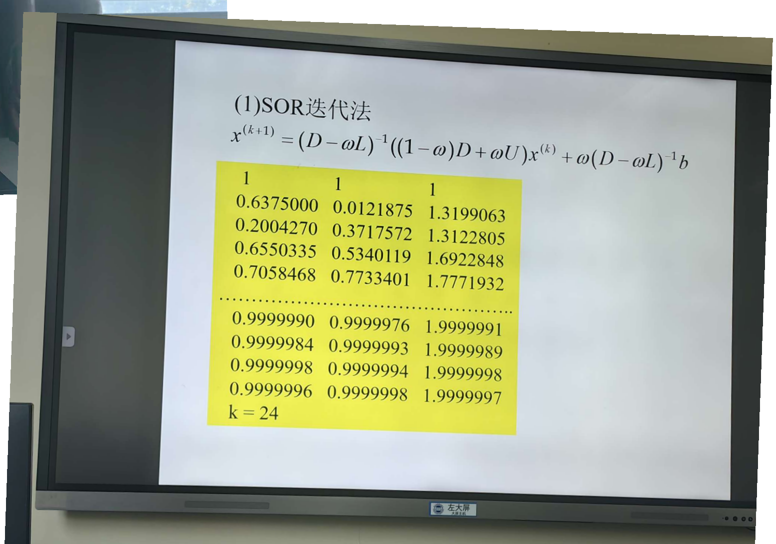
$f_w=w(D-wL)^{-1}b$

$x^{(k+1)}=B_w x^k+f_w$

上式为逐次超松弛法(SOR迭代法)的矩阵形式.

B_w为SOR法的迭代矩阵





迭代法的收敛性

非线性方程迭代求解收敛性

$$x_{k+1}=\varphi(x_k) \qquad |\varphi'(x)|\le L<1$$

线性方程组迭代求解收敛性

$$x^{(k+1)}=Bx^{(k)}+f \qquad \|B\|<1$$

向量和矩阵的范数

范数：一种描述向量大小的度量，

向量范数 $\|x\|_p=(|x_1|^p+\cdots+|x_n|^p)^{\frac{1}{p}} \quad 1\le p\le\infty$

∞ 时为取最大值

- 在数值计算中，常用的向量范数有三种。设 $x=(x_1,x_2,\cdots x_n)^T\in R^n$，规定

$$
\left.\begin{array}{lll}
(1) & \text{向量的"1"范数} & \|x\|_1=\sum\limits_{i=1}^{n}|x_i| \\
(2) & \text{向量的"2"范数} & \|x\|_2=(\sum\limits_{i=1}^{n}x_i^2)^{1/2} \\
(3) & \text{向量的"}\infty\text{"范数} & \|x\|_\infty=\max\limits_{1\le i\le n}|x_i|
\end{array}\right\}
$$

- 常用的向量范数

$$\|x\|_p=\left(|x_1|^p+\cdots+|x_n|^p\right)^{\frac{1}{p}},\quad 1\le p\le\infty$$

- 常用的矩阵范数

$$\|A\|_p=\sup\frac{\|Ax\|_p}{\|x\|_p},\quad 1\le p\le\infty$$

- 矩阵的谱半径

$$\rho(A)=\max\left(|\lambda_1|,\cdots,|\lambda_n|\right)$$

- 例：计算矩阵 $A=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$ 的范数和谱半径。
- 例：范数在误差估计中的应用

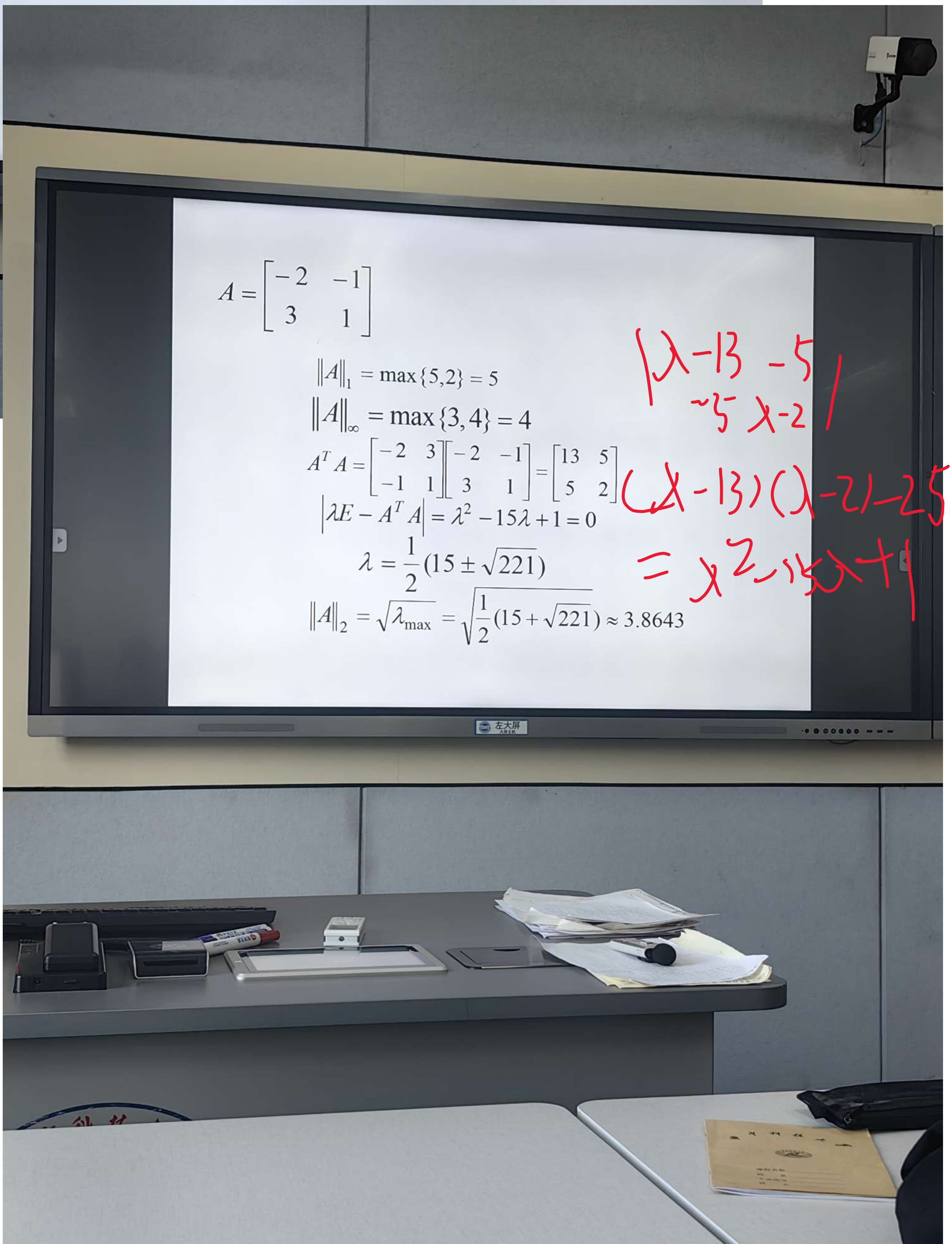
小组屏01

(1)矩阵的列范数：$\|A\|_1 = \max\limits_{1\le j\le n}\sum\limits_{i=1}^{n}|a_{ij}|$

(2)矩阵的行范数：$\|A\|_\infty = \max\limits_{1\le i\le n}\sum\limits_{j=1}^{n}|a_{ij}|$

(3)矩阵的欧氏范数：$\|A\|_E = (\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}a_{ij}^2)^{1/2}$

(4)矩阵的谱范数：$\|A\|_2 = \sqrt{\lambda_{max}}, \lambda_{max}$



小　结

- 本章主要介绍了解线性方程组的直接法和迭代法
- 直接法的基础是Gauss消去法及其矩阵形式的LU分解。
- 选取主元素是保证消去法计算稳定性及提高精度的有效方法，列主元比较常用。
- 利用对称正定矩阵的矩阵形式的特殊性，可以简化LU分解，得到追赶法及平方根法，这两个方法是解决两类特殊形式方程组的有效方法。

- 对于大型稀疏方程组可采用迭代法求解，比较简便有效的迭代法是Jacobi迭代和Gauss-Seidel方法；
- 当选取合适的松弛因子，SOR方法可获得较快的收敛速度，被广泛应用.

判断收敛：

- Jacobi迭代：$\tilde{A} = D$

定理：A行对角优、或A列对角优，Jacobi迭代收敛。

- Gauss-Seidel迭代：$\tilde{A} = D + L$

定理：A行对角优、或A列对角优、或A正定，Gauss-Seidel迭代收敛。

- 松弛迭代：$\tilde{A} = w^{-1}D + L$

定理：松弛迭代收敛 $0<w<2$

定理：A正定且$0<w<2$ 松弛迭代收敛