计算传热学的近代进展

第五章 代数方程的求解

主讲 陶文铨

西安交通大学能源与动力工程学院
热流中心 CFD-NHT-EHT CENTER
2016年6月13日, 西安

第五章目录

5.1 PDMA 方法

5.1.1 二维高阶格式代数方程的系数矩阵

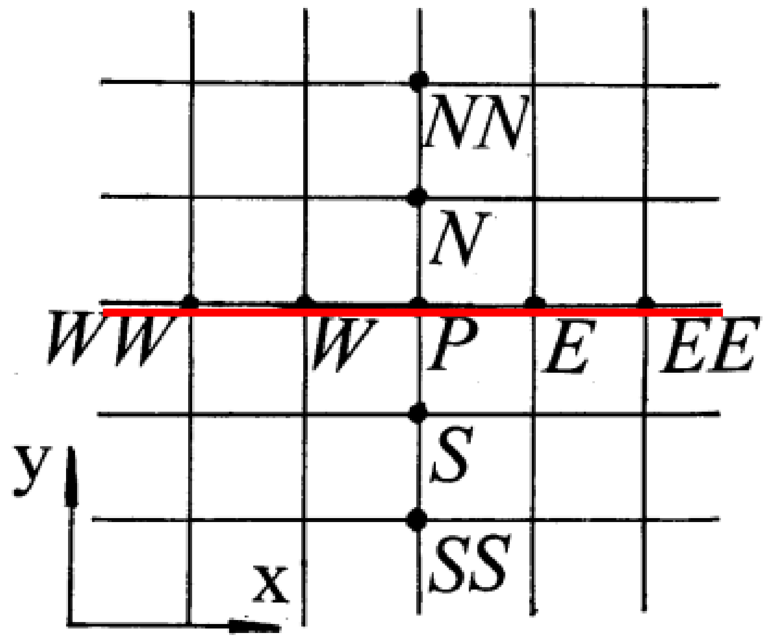
5.1.2 PDMA算法的基本思想

5.1.3 PDMA算法的实施细节

5.1 PDMA算法

5.1.1 二维高阶格式代数方程的系数矩阵

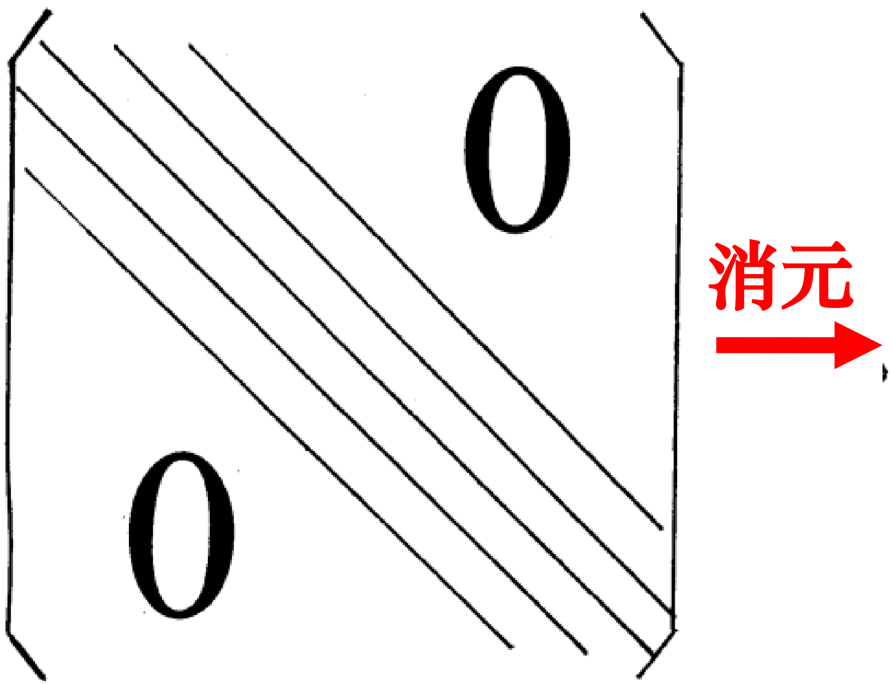
对流项采用三阶格式时代数方程为九对角阵，每一个坐标方向上有五个未知数；如果同一个坐标方向未知数直接求解，另一个方向上未知数归入源项，则有：

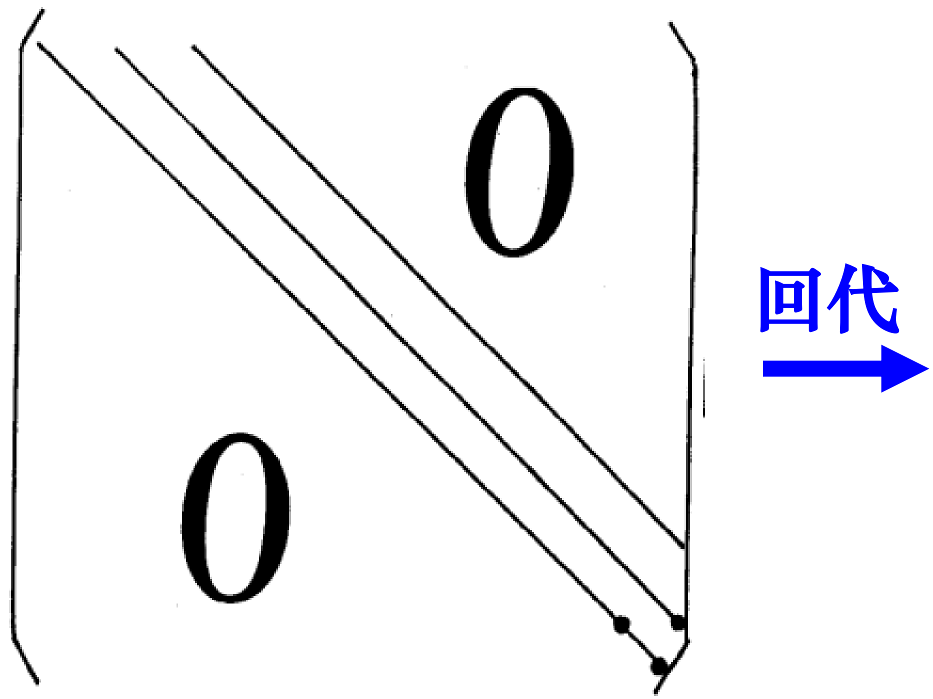


$$a_P\phi_P = a_{EE}\phi_{EE} + a_E\phi_E + a_W\phi_W + a_{WW}\phi_{WW}$$

$$+ \underline{(b + a_N\phi_N^* + a_{NN}\phi_{NN}^* + a_S\phi_S^* + a_{SS}\phi_{SS}^*)}$$

5.1.2 PDMA算法的基本思想





5.1.3 PDMA算法的实施细节

1.消元过程

设消元后的上三角阵方程可以表示成为：

$$\phi_i = \mathrm{A}_{\mathrm{i}}\phi_{i+2} + \mathrm{B}_{\mathrm{i}}\phi_{i+1} + \mathrm{C}_{\mathrm{i}},\ i = 1,2,3\ldots\ldots M \qquad \text{(a)}$$

消元过程就是要获得系数A_i,B_i,C_i 与原方程中系数间的关系：

$$a_P\phi_i = a_{EE}\phi_{i+2} + a_E\phi_{i+1} + a_W\underline{\phi_{i-1}} + a_{WW}\underline{\phi_{i-2}} + b^P \qquad \text{(b)}$$

设法从（b)式中逐一消去 ϕ_{i-1},ϕ_{i-2} ，将所得结果与（a) 相比，即可得出系数A_i,B_i,C_i 与原方程系数间的关系式。

首先消去 ϕ_{i-2}，将式(a)对i-2 写出，

$$\phi_{i-2} = A_{i-2}\phi_i + B_{i-2}\phi_{i-1} + C_{i-2} \quad \text{(a')}$$

与式(b)相比

$$a_P\phi_i = a_{EE}\phi_{i+2} + a_E\phi_{i+1} + a_W\phi_{i-1} + \underline{a_{WW}\phi_{i-2}} + b^P \quad \text{(b)}$$

需对 (a')式两端各乘以 a_{ww}

$$\cancel{a_{WW}\phi_{i-2}} = a_{WW}A_{i-2}\phi_i + a_{WW}B_{i-2}\phi_{i-1} + a_{WW}C_{i-2} \quad \text{(c)}$$

将 (C) 与 (b) 相加

$$a_P\phi_i = a_{EE}\phi_{i+2} + a_E\phi_{i+1} + a_W\phi_{i-1} + \cancel{a_{WW}\phi_{i-2}} + b^P \quad \text{(b)}$$

$$(c)+(b)$$

可得出消去 ϕ_{i-2} 项的结果，记为式(d):

$$(a_P - a_{WW}A_{i-2})\phi_i = a_{EE}\phi_{i+2} + a_E\phi_{i+1} +$$

$$(a_W + a_{WW}B_{i-2})\phi_{i-1} + b^P + a_{WW}C_{i-2} \quad \textbf{(d)}$$

类似地，将(a) 对i-1点写出:。

$$\phi_{i-1} = A_{i-1}\phi_{i+1} + B_{i-1}\phi_i + C_{i-1} \qquad \textbf{(a”)}$$

将(a”) 乘以式(d)中相应的系数 $(a_W + a_{WW}B_{i-2})$

$$(a_W + a_{WW}B_{i-2})\phi_{i-1} = A_{i-1}(a_W + a_{WW}B_{i-2})\phi_{i+1} +$$

$$B_{i-1}(a_W + a_{WW}B_{i-2})\phi_i + C_{i-1}(a_W + a_{WW}B_{i-2}) \quad \textbf{(f)}$$

将(d) 与式(f) 相加，消去 ϕ_{i-1} 项:

$$(a_P - a_{WW}A_{i-2})\phi_i = a_{EE}\phi_{i+2} + a_E\phi_{i+1} + (a_W + a_{WW}B_{i-2})\phi_{i-1} + b^P + a_{WW}C_{i-2} \quad \text{(d)}$$

$$(a_W + a_{WW}B_{i-2})\phi_{i-1} = A_{i-1}\underline{(a_W + a_{WW}B_{i-2})}\phi_{i+1} + B_{i-1}\underline{(a_W + a_{WW}B_{i-2})}\phi_i + C_{i-1}\underline{(a_W + a_{WW}B_{i-2})} \quad \text{(f)}$$

令 $D = a_W + a_{WW}B_{i-2}$

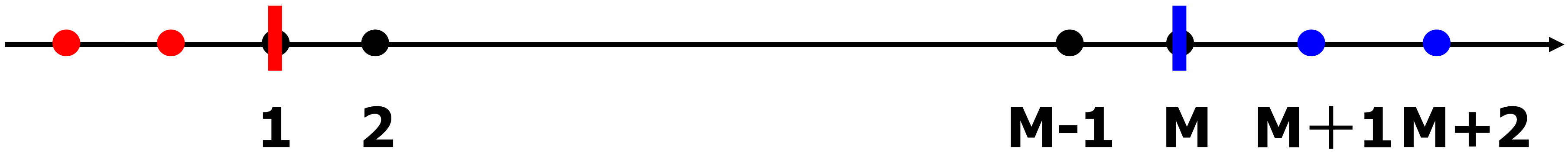
$$\phi_i\underline{[(a_P - a_{WW}A_{i-2}) - B_{i-1}D]} = \underline{a_{EE}}\phi_{i+2} + \underline{(a_E + A_{i-1}D)}\phi_{i+1} + \underline{(b^P + a_{WW}C_{i-2} + C_{i-1}D)}$$

将此式表示成与(a)相对应的形式:

$$\phi_i = \underbrace{\left(\frac{a_{EE}}{a_P - a_{WW}A_{i-2} - B_{i-1}D}\right)}_{\boxed{A_i}}\phi_{i+2} + \underbrace{\left(\frac{a_E + DA_{i-1}}{a_P - a_{WW}A_{i-2} - B_{i-1}D}\right)}_{\boxed{B_i}}\phi_{i+1} + \underbrace{\left(\frac{DC_{i-1} + a_{WW}C_{i-1} + b^P}{a_P - a_{WW}A_{i-2} - B_{i-1}D}\right)}_{\boxed{C_i}} \qquad \text{(g)}$$

注意到：系数的计算式是递归的，需要知道在节点1，2 上的A,B,C 的计算式。

利用端部条件来导出节点1，2的系数计算式。



获得递归系数的初值

获得回代过程初值

据公式(b)：

$$a_P\phi_i = a_{EE}\phi_{i+2} + a_E\phi_{i+1} + a_W\underline{\phi_{i-1}} + a_{WW}\underline{\phi_{i-2}} + b^P$$

显然对计算区域的左端有：

节点1 $\boxed{(a_W)_1 = 0,\ (a_{WW})_1 = 0}$

节点2 $\boxed{(a_{WW})_2 = 0}$

于是对 i =1点写出(b)，有：

$$a_P\phi_1 = \underline{a_{EE}\phi_3} + \underline{a_E\phi_2} + \cancelto{0}{a_W}\phi_W + \cancelto{0}{a_{WW}}\phi_{WW} + b^P$$

$$(a_P)_1\phi_1 = (a_{EE})_1\phi_3 + (a_E)_1\phi_2 + 0 + 0 + (b^P)_1$$

$$\phi_1 = \underbrace{\left(\frac{a_{EE}}{a_P}\right)_1}_{\boxed{A_1}}\phi_3 + \underbrace{\left(\frac{a_E}{a_P}\right)_1}_{\boxed{B_1}}\phi_2 + \underbrace{\left(\frac{b^P}{a_P}\right)_1}_{\boxed{C_1}} \qquad \text{(i)}$$

再对 i =2点写出(b)，有：

$$(a_P)_2\phi_2=(a_{EE})_2\phi_4+(a_E)_2\phi_3+(a_W)_2\phi_1+0+(b^P)_2$$

已知 $$\phi_1=\left(\frac{a_{EE}}{a_P}\right)_1\phi_3+\left(\frac{a_E}{a_P}\right)_1\phi_2+\left(\frac{b^P}{a_P}\right)_1 \qquad \text{(j)}$$

将(j)式代入上式，消去 ϕ_1 从而有：

$$\phi_2=(\frac{a_{EE}}{a_P})_2\phi_4+(\frac{a_{EE}}{a_P})_2\phi_3+(\frac{a_W}{a_P})_2[(\underbrace{\frac{a_{EE}}{a_P}}_{A_1})_1\phi_3+(\underbrace{\frac{a_E}{a_P}}_{B_1})_1\phi_2+(\underbrace{\frac{b^P}{a_P}}_{C_1})_1]+(b^P)_2$$

归并同类项，最后得出：

$$\phi_2=(\underbrace{\frac{(a_{EE})_2}{(a_P)_2-B_1(a_W)_2}}_{\boxed{A_2}})\phi_4+(\underbrace{\frac{(a_E)_2+A_1(a_W)_2}{(a_P)_2-B_1(a_W)_2}}_{\boxed{B_2}})\phi_3+\underbrace{\frac{(b^P)_2+C_1(a_W)_2}{(a_P)_2-B_1(a_W)_2}}_{\boxed{C_2}}$$

2.回代过程

将式(a)对 *i* =M点写出： $(a_E)_M=0,\ (a_{EE})_M=0$

$$\phi_i = \mathrm{A_i}\phi_{i+2} + \mathrm{B_i}\phi_{i+1} + \mathrm{C_i}, \; i = 1,2,3\ldots\ldots M \qquad \textbf{(a)}$$

$$\phi_M = \cancelto{0}{\mathrm{A_M}\phi_{M+2}} + \cancelto{0}{\mathrm{B_M}\phi_{M+1}} + \mathrm{C_M} \qquad \boxed{\phi_M = \mathrm{C}_M}$$

将式(a)对 i =M-1点写出： $(a_{EE})_{M\text{-}1} = 0$

$$\phi_{M-1} = \cancelto{0}{\mathrm{A_{M\text{-}1}}\phi_{M+1}} + \mathrm{B_{M\text{-}1}}\phi_M + \mathrm{C_{M\text{-}1}}$$

$$\boxed{\phi_{M-1} = \mathrm{B_{M-1}}\phi_{\mathrm{M}} + \mathrm{C_{M-1}}}$$

由 ϕ_M, ϕ_{M-1} 对i=M-2,M-3,……3,2,1点逐一进行会代可得出所有各点之解。

第五章目录

5.2 Krylov子空间方法简介

5.2 Krylov子空间方法简介

5.2.1 加速代数方程迭代求解收敛速度的方法

（1）加速边界条件影响的传入－ADI；

（2）加速局部守恒条件的满足－块修正技术;

（3）加速不同波长误差分量均匀的衰减－多重网格;

（4）适当增加直接求解的份量－ Krylov子空间方法。

5.2.2 古典迭代法与近代迭代法

1. 古典迭代法：Jacobi, Gauss-Seidel , SOR；

收敛条件：代数方程不可约而且对角占优；收敛速度较低;

2. 近代迭代法：从总体上说是一种迭代求解方法，但其中求解步骤则增加了直接求解成分，在数值代数界称之为Krylov子空间方法。在数值代数界已经得到广泛应用，但在CFD,NHT中应用还不广。

5.2.3 Krylov子空间方法分类

按所求解代数方程系数矩阵的不同，可分为：

（1）求解对称正定系数矩阵代数方程的方法：CG，PCG；均分网格的常物性导热问题属于此类；

（2）求解对称非正定系数矩阵代数方程的方法：如SYMMLQ

（3）求解非对称非正定系数矩阵代数方程的方法：流场求解问题属之：GMRES，CGS，Bi-CGSTAB，等。

5.2.4 共轭梯度法CG

1. 实施步骤

给定代数方程组 $\vec{A}\vec{\phi}=\vec{b}$

其中系数矩阵与列矢量均为定值，系数矩阵为大型稀疏矩阵，对称正定。求解步骤如下：

（1）设定初场 $\overrightarrow{\phi^{0}}$ （本质上是迭代方法）

(2) 计算余量 $\overrightarrow{r^0} = \vec{b} - \vec{A}\overrightarrow{\phi^0}$ （如果小于规定值就收敛）

(3) 令 $\overrightarrow{p^0} = \overrightarrow{r^0}$ （由 $\overrightarrow{r^0}, \overrightarrow{p^0}$ 启动一次迭代计算！）

(4) 对于k=0,1….直到 $r^k < allowed\ value$

(a) 计算系数 $\alpha^k = (\overrightarrow{r^k}, \overrightarrow{r^k})/(\vec{A}\overrightarrow{p^k}, \overrightarrow{p^k})$

(b) 更新被求值 $\overrightarrow{\phi^{k+1}} = \overrightarrow{\phi^k} + \alpha^k \overrightarrow{p^k}$

(c) 更新余量 $\overrightarrow{r^{k+1}} = \overrightarrow{r^k} - \alpha^k \vec{A}\overrightarrow{p^k}$

(d) 确定系数Beta, $\beta^k = (\overrightarrow{r^{k+1}}, \overrightarrow{r^{k+1}})/(\overrightarrow{r^k}, \overrightarrow{r^k})$

(e) 更新 *p* 矢量 $\overrightarrow{p^{k+1}} = \overrightarrow{r^{k+1}} + \beta^k \overrightarrow{p^k}$

根据 $\overrightarrow{r^{k+1}}, \overrightarrow{p^{k+1}}$ 重复(a)-(e)的迭代计算!

2. 数值实例

给定代数方程组 $\vec{A}\vec{\phi} = \vec{b}$

其中 $\vec{A} = \begin{vmatrix} 2 & 3 & -1 \\ 3 & 4 & -2 \\ -1 & -2 & 1 \end{vmatrix}$, $\vec{b} = \begin{vmatrix} 5 \\ 5 \\ -2 \end{vmatrix}$

$\boxed{\text{对称正定系数矩阵}}$

该代数方程组的精确解为 $\overrightarrow{\phi^T} = \begin{vmatrix} 1 & 2 & 3 \end{vmatrix}$

用CG方法迭代式地求解如下：

(1) 设定初场 $\overrightarrow{\phi^0}$ $\overrightarrow{(\phi^0)^T} = |0 \quad 0 \quad 0|$ 迭代初场

(2) 计算余量 $\overrightarrow{r^0} = \vec{b} - \vec{A}\overrightarrow{\phi^0}$

$$\vec{A} = \begin{vmatrix} 2 & 3 & -1 \\ 3 & 4 & -2 \\ -1 & -2 & 1 \end{vmatrix}$$

$$r_1^0 = 5 - [2\times(0) + 3\times(0) - 1\times(0)] = 5$$

$$r_2^0 = 5 - [3\times(0) + 4\times(0) - 2\times(0)] = 5$$

$$r_3^0 = -2 - [-1\times(0) - 2\times(0) + 1\times(0)] = -2$$

$$\vec{b} = \begin{vmatrix} 5 \\ 5 \\ -2 \end{vmatrix}$$

(3) 令 $\overrightarrow{p^0} = \overrightarrow{r^0}$ $\overrightarrow{(p^0)^T} = |5 \quad 5 \quad -2|$

(4) 对于k=0,1….直到 $\overrightarrow{r^k} < allowed\ value$

对k=0:

(a)计算系数Alfa $\alpha^0 = (\overrightarrow{r^0}, \overrightarrow{r^0}) / (\vec{A}\overrightarrow{p^0}, \overrightarrow{p^0})$

$$(\overrightarrow{r^0}, \overrightarrow{r^0}) = 5\times5+5\times5+2\times2 = 54$$

$$\vec{A}\overrightarrow{p^0} = \begin{vmatrix} 2 & 3 & -1 \\ 3 & 4 & -2 \\ -1 & -2 & 1 \end{vmatrix} \begin{vmatrix} 5 \\ 5 \\ -2 \end{vmatrix} = \begin{vmatrix} 2\times5+3\times5+1\times2 \\ 3\times5+4\times5+2\times2 \\ -1\times5+(-2\times5)+1\times(-2) \end{vmatrix} = \begin{vmatrix} 27 \\ 39 \\ -17 \end{vmatrix}$$

$$(\vec{A}\overrightarrow{p^0}, \overrightarrow{p^0}) = \begin{vmatrix} 27 \\ 39 \\ -17 \end{vmatrix} \begin{vmatrix} 5 \\ 5 \\ -2 \end{vmatrix} = 27\times5+39\times5+17\times2 = 364$$

$$\alpha^0 = (\overrightarrow{r^0}, \overrightarrow{r^0}) / (\vec{A}\overrightarrow{p^0}, \overrightarrow{p^0}) \qquad \alpha^0 = \frac{54}{364} = \frac{27}{182}$$

(b) 更新被求值 $\overrightarrow{\phi^{k+1}} = \overrightarrow{\phi^{k}} + \alpha^{k}\overrightarrow{p^{k}}$

$$\overrightarrow{r^{0}} = \begin{vmatrix} 5 \\ 5 \\ -2 \end{vmatrix}$$

$$\overrightarrow{\phi^{1}} = \overrightarrow{\phi^{0}} + \frac{27}{182}\begin{vmatrix} 5 \\ 5 \\ 2 \end{vmatrix} = \frac{1}{182}\begin{vmatrix} 135 \\ 135 \\ -54 \end{vmatrix}$$

(c) 更新余量 $\overrightarrow{r^{k+1}} = \overrightarrow{r^{k}} - \alpha^{k}\overrightarrow{A}\overrightarrow{p^{k}}$

$$\alpha^{0} = \frac{27}{182}$$

$$\overrightarrow{r^{1}} = \overrightarrow{r^{0}} - \alpha^{0}\overrightarrow{A}\overrightarrow{p^{0}} = \begin{vmatrix} 5 - \dfrac{27\times 27}{182} \\ 5 - \dfrac{27\times 39}{182} \\ -2 + \dfrac{27\times 17}{182} \end{vmatrix} = \frac{1}{182}\begin{vmatrix} 181 \\ -143 \\ 95 \end{vmatrix}$$

$$\overrightarrow{A}\overrightarrow{p^{0}} = \begin{vmatrix} 27 \\ 39 \\ -17 \end{vmatrix}$$

尚未收敛

(d) 确定系数Beta $\beta^k = (\overrightarrow{r^{k+1}}, \overrightarrow{r^{k+1}})/(\overrightarrow{r^k}, \overrightarrow{r^k})$

$$\beta^0 = (\overrightarrow{r^1}, \overrightarrow{r^1})/(\overrightarrow{r^0}, \overrightarrow{r^0}) = \frac{\frac{1}{182^2}(181^2+143^2+95^2)}{54} \qquad \overrightarrow{r^1} = \frac{1}{182}\begin{vmatrix}181\\-143\\95\end{vmatrix}$$

$$= \frac{1.87885}{54} \qquad (\overrightarrow{r^0}, \overrightarrow{r^0}) = 54$$

(e)更新 *p* 矢量 $\overrightarrow{p^{k+1}} = \overrightarrow{r^{k+1}} + \beta^k \overrightarrow{p^k}$

$$\overrightarrow{p^1} = \overrightarrow{r^1} + \beta^0 \overrightarrow{p^0} = \frac{1}{182}\begin{vmatrix}181\\-143\\95\end{vmatrix} + \frac{1.87885}{54}\begin{vmatrix}5\\5\\-2\end{vmatrix} = \frac{1}{54}\begin{vmatrix}63.097\\-33.034\\24.429\end{vmatrix}$$

一个迭代过程的计算结束。对本例，当k=2时得精确解。

5.2.5 预条件矩阵共轭梯度法(PCG)

1. CG方法收敛速度取决于其条件数（condition number)

矩阵的条件数是其特征值的最大值与最小值之比。条件数越大收敛越慢。

流动与传热问题离散所得矩阵的条件数大致正比于某个坐标方向的节点个数的平方：100个节点大致为10000；迭代收敛很慢，使CG的应用大受限制；

1978年Kershaw在JCP上提出了预条件矩阵方法，形成了预条件共轭梯度法，大大提高了线性代数方程迭代求解的收敛速度。

2. PCG方法的基本思想

预先将系数矩阵 $\overrightarrow{A}$ 进行处理，找出一个与之接近而条件数很小的矩阵 $\overrightarrow{M}$ ，称为预条件矩阵。再通过算法的设计将求解 $\overrightarrow{A}\vec{\phi}=\vec{b}$ 转化成为求解 $\overrightarrow{M}\overrightarrow{Z}=\vec{r}$ ，达到加速收敛的目的。

3. 获得预条件矩阵的不完全Cholesky分解法

（1）预条件矩阵与原矩阵间的关系

预条件矩阵与原矩阵间有以下关系：

$$\overrightarrow{A}=\overrightarrow{M}+\overrightarrow{E}$$

矩阵 $\overrightarrow{E}$ 是一个“小阵”，即：$\overrightarrow{E}\vec{\phi}\cong 0$

对预条件矩阵的要求:

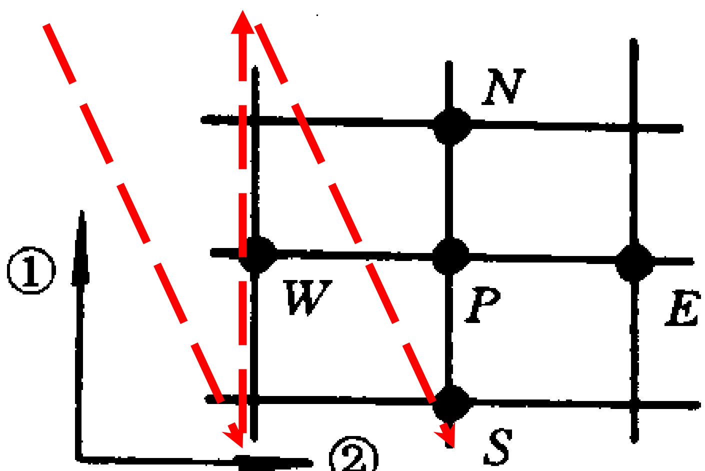
(a) 条件数要小;

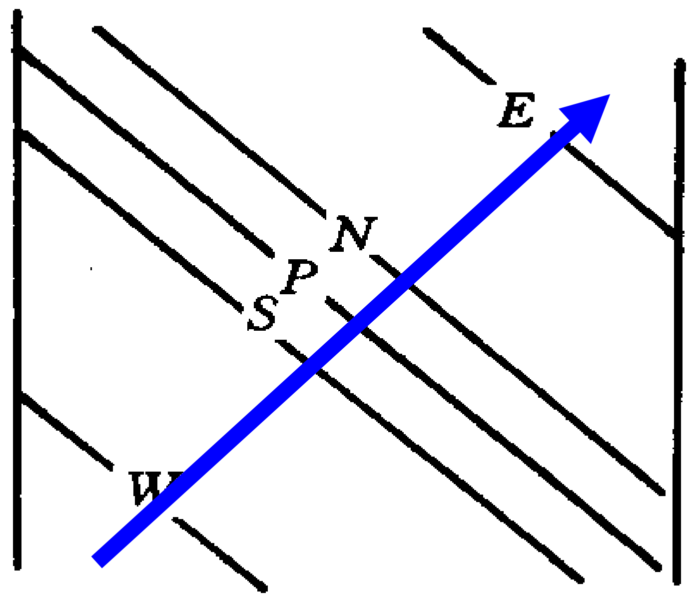
(b) 便于对 $\overrightarrow{M}\vec{\phi}=\vec{r}$ 的求解。

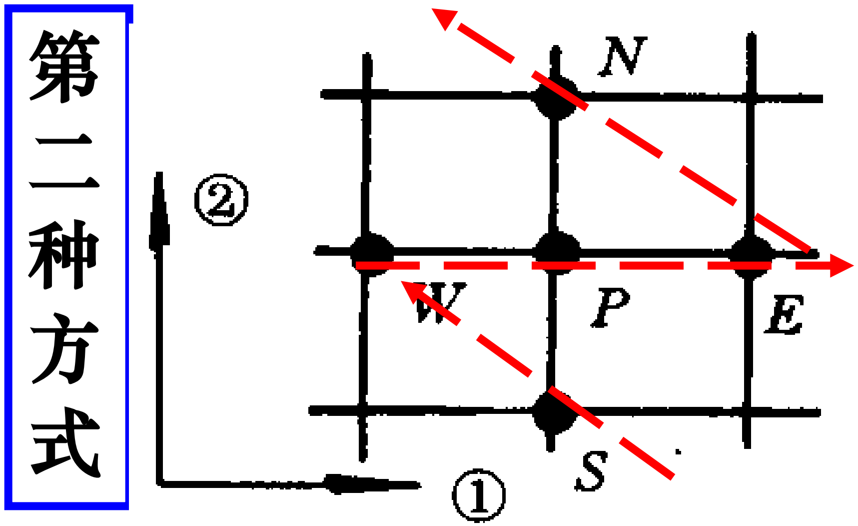
依据构造预条件矩阵的不同形成了不同的PCG。

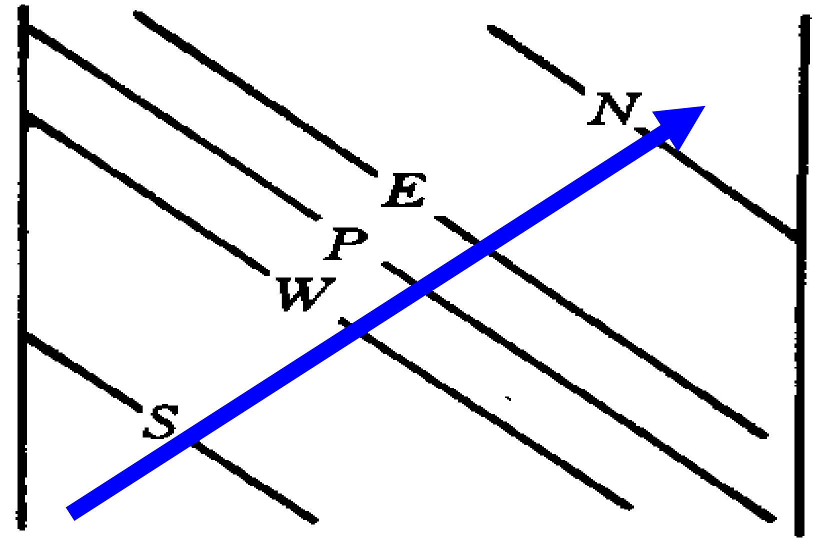
(2) 变量的一维存储方式

第一种方式







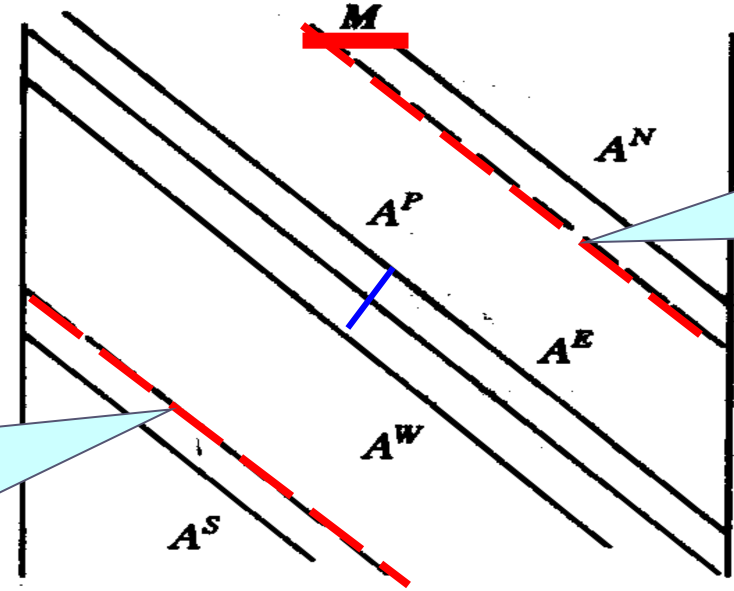


这里采用第 二种方式进行一维存储。

（3）不完全Cholesky分解

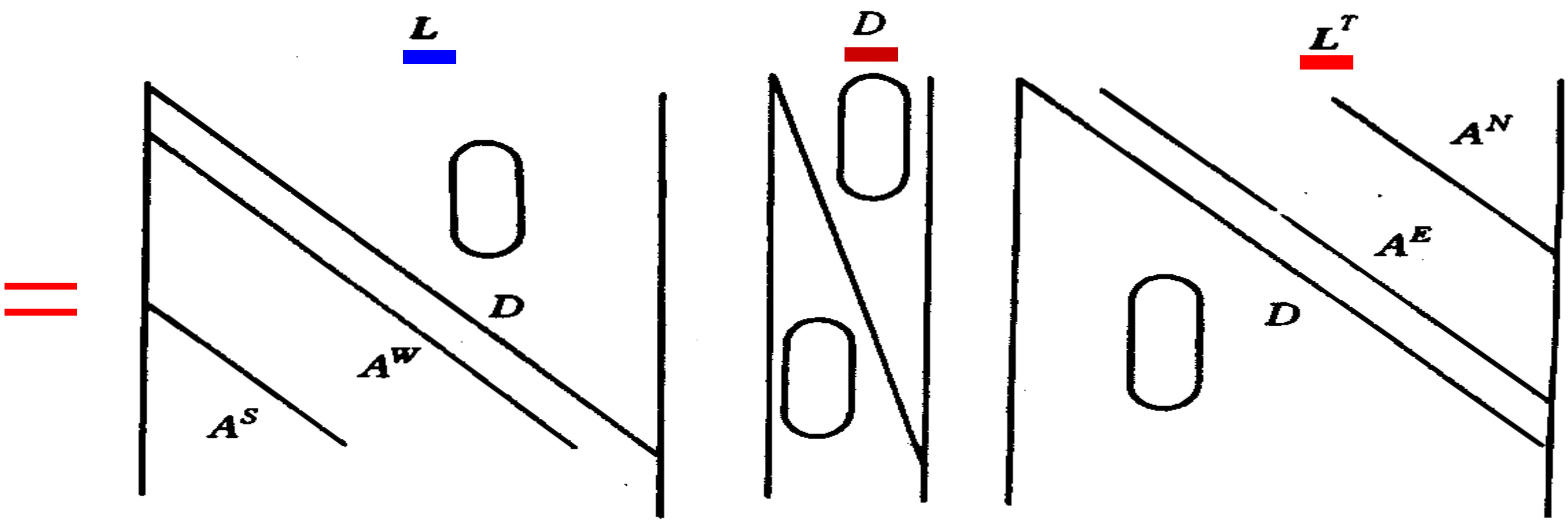
$$\overrightarrow{M} \longleftarrow \overrightarrow{L}\,\overrightarrow{D}\,\overrightarrow{U}$$

$\overrightarrow{L}, \overrightarrow{D}, \overrightarrow{U}, \overrightarrow{M}$ 与原矩阵 $\overrightarrow{A}$ 的非零系数项的关系如下：



M阵不同于A阵的元素，由矩阵相乘得出

M阵不同于A阵的元素，由矩阵相乘得出



A^S, A^W, A^P, A^E, A^N 为原矩阵中相对应对角线上的元素；

D矩阵中的主对角元为：

$$D_{i,j}^P = \frac{1}{A_{i,j}^P - D_{i-1,j}^P A_{i,j}^W (A_{i-1,j}^E + \alpha A_{i-1,j}^N) - D_{i,j-1}^P A_{i,j}^S (A_{i,j-1}^N + \alpha A_{i,j-1}^E)}$$

L与U 矩阵中的主对角元则为 $1/D_{i,j}^P$

$A_{i,j}^P$ **中的下标 i,j 表示了变量** $\phi_{i,j}$ **的位置，在该位置上有NEWS以及P 的5个系数，** $\alpha = 0$ **称为不完全分解。**

4. ICCG算法

采用不完全Cholesky分解方法生成预条件矩阵的PCG称为ICCG（CGILU），计算步骤如下：

（1）设定初场 $\vec{\phi}^0$ **(本质上是迭代方法）**

(2) 计算余量 $\overrightarrow{r^0} = \vec{b} - \overrightarrow{A}\overrightarrow{\phi^0}$ (如果小于规定值就收敛)

(3) 令 $\overrightarrow{p^0} = \overrightarrow{r^0}$; $S^0 = 10^{30}$ (由 $\overrightarrow{r^0}, \overrightarrow{p^0}, \overrightarrow{S^0}$ 启动一次迭代)

(4) 对于k=1,2….直到 $\overrightarrow{r^k} < allowed\ value$

(a) 求解 $\overrightarrow{M}\overrightarrow{Z}^k = \vec{r}^{k-1}$ $(\vec{L}\vec{D}\vec{U}\vec{Z}^k = \vec{r}^{k-1})$ ($\vec{r}^0$ 已设定)

(b) 计算 $S^k = (\overrightarrow{r^{k-1}}, \overrightarrow{Z^k})$ ($\overrightarrow{Z^k}$ 已解出)

(c) 确定Beta $\beta^k = \overrightarrow{S^k} / \overrightarrow{S^{k-1}}$ ($\vec{S}^0$ 已设定)

(d) 确定 p 矢量 $\overrightarrow{p^{k+1}} = \overrightarrow{Z^k} + \beta^k \overrightarrow{p^{k-1}}$ ($\overrightarrow{p^{k-1}}$ 已设定)

(e) 确定Alfa $\alpha^k = S^k / (\vec{p}^k, \vec{A}\vec{p}^k)$

(f) 更新被求值 $\overrightarrow{\phi^{k+1}} = \overrightarrow{\phi^k} + \alpha^k \overrightarrow{p^k}$

(g) 更新余量 $\overrightarrow{r^{k+1}} = \overrightarrow{r^k} - \alpha^k \vec{A}\overrightarrow{p^k}$

根据 $\overrightarrow{r^{k+1}}, \overrightarrow{p^{k+1}}, S^k$ 重复(a)-(g)的迭代计算!

5. 讨论

（1）ICCG与CG相比主要区别在于增加了求解由预条件矩阵所形成的方程，$\vec{M}\vec{Z}^{k}=\vec{r}^{k-1}$ 并通过中间变量 $\vec{Z}$ 去改进 $\vec{\phi}$ ；而检验迭代法收敛的条件是原方程的余量小于规定值，因此收敛于原方程的解。

(2) 采用三种方法求解某个算例的迭代次数:

求解方法	迭代次数
Gauss－Seidel	208000
Block SOR	765
ICCG	25

5.2.6 关于Krylov子空间方法的讨论

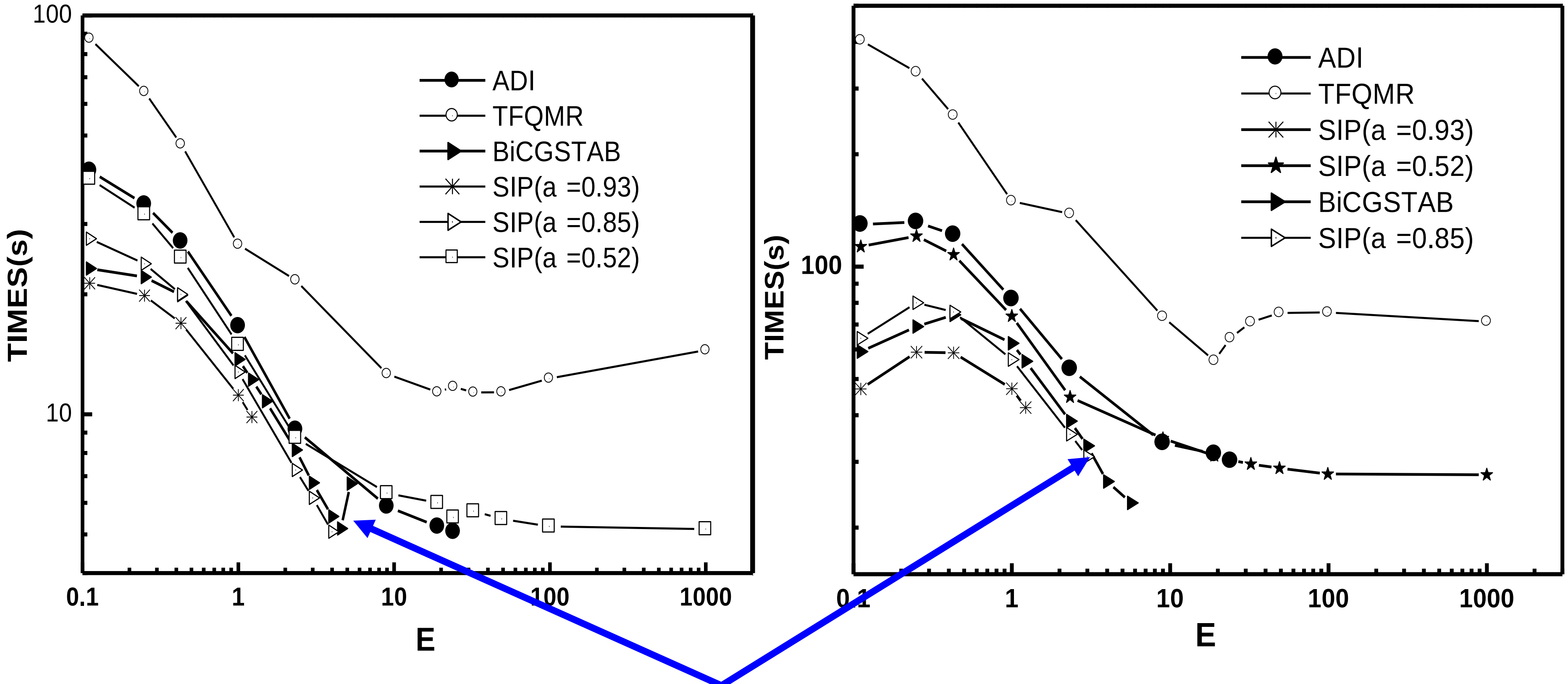
(1) 用于流场求解等非线性问题应用Krylov子空间法存在的问题:

对于常系数矩阵的代数方程此法优点明显;

但对于非线性问题由于当前的系数矩阵是需要不断更新的，不断构造预条件件矩阵使计算工作量大大增加，部分抵消了对每组系数求解收敛性增加带来的收益；我们的实践表明采用各类Krylov子空间法提高的收益大致在20－50%左右。

(2) 算法的收敛性与代数方程求解方法有关。

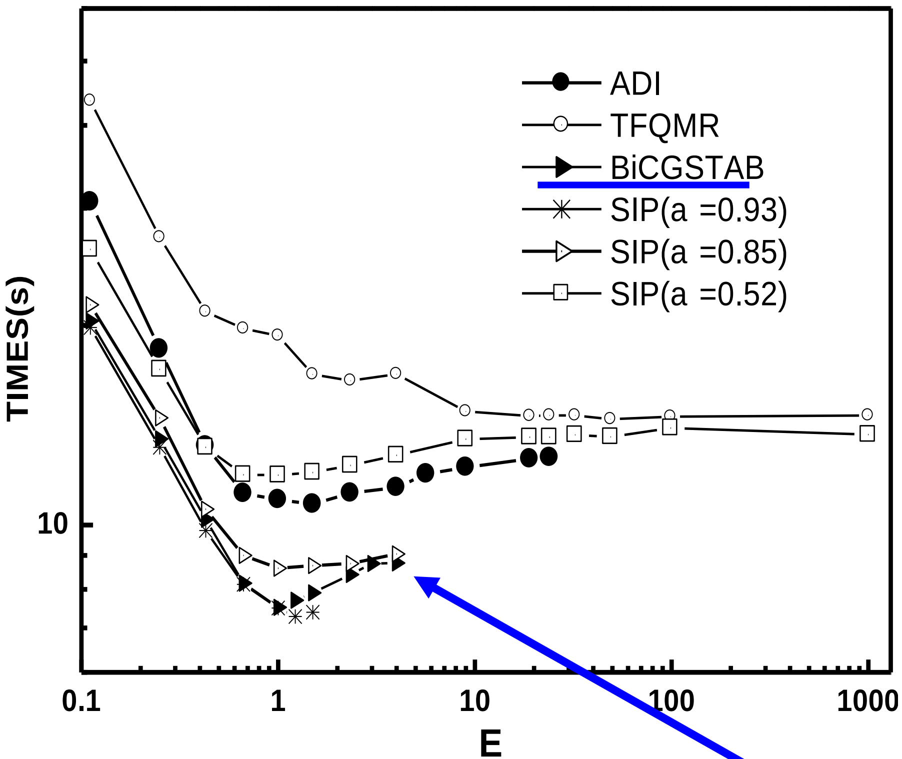
SIMPLER算法



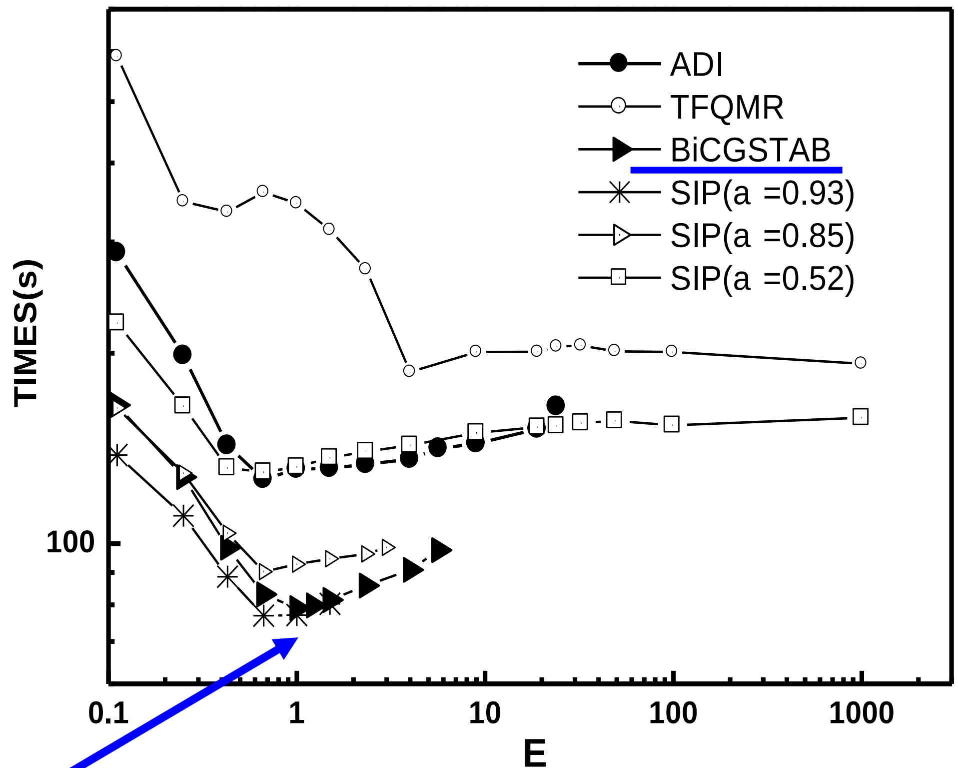
(a) 网格52×52　　**BiCGSTAB**　　(b) 网格82×82

顶盖驱动流健壮性比较结果（Re＝100）

SIMPLER算法



(a) 网格42×42



(b) 网格82×82

BiCGSTAB

方腔自然对流健壮性比较结果（Ra=10,000 ）

5.2.7 BiCGSTAB的计算步骤

(1) 设定初场 $\overrightarrow{\phi^0}$ (本质上是迭代方法)

(2) 计算余量 $\overrightarrow{r^0} = \vec{b} - \overrightarrow{A}\overrightarrow{\phi^0}$ (如果小于规定值就收敛)

(3) $\rho^0 = \alpha^0 = \omega^0 = 1;\ \mathbf{v}^0 = \mathbf{p}^0 = 0$

(4) 对于k=1,2….直到 $\overrightarrow{r^k} < allowed\ value$

$(a)\ \rho^k = (\mathbf{r}^0, \mathbf{r}^{k-1})$

$(b)\ \beta^k = (\rho^k / \rho^{k-1})(\alpha^{k-1} / \omega^{k-1}) = \rho^k \alpha^{k-1} / \rho^{k-1}\omega^{k-1}$

$(c)\ \mathbf{p}^k = \mathbf{r}^{k-1} + \beta(\mathbf{p}^{k-1} - \omega^{k-1}\mathbf{v}^{k-1})$ (构建第一次用预矩阵求解方程的右端项)

$(d)\ \mathbf{My} = \mathbf{p}^k$ (第一次用预矩阵求解方程)

$(e)\ \mathbf{v}^k = \mathbf{Ay}$ ($\mathbf{v}^0$已经给定；y已经解出)

$(f)\alpha^k = \rho^k / (\mathbf{r}^0, \mathbf{v}^k)$ (更新Alfa ）

$(g)\ \mathbf{s} = \mathbf{r}^{k-1} - \alpha\, \mathbf{v}^k$ (构建第二次用预矩阵求解方程的右端项）

$(h)\ \mathbf{Mz} = \mathbf{s}$ (第二次用预矩阵求解方程）

$(i)\ \mathbf{t} = \mathbf{Az}$ (用于更新Omega，z 已经解出 ）

$(j)\ \omega^k = (\mathbf{t},\mathbf{s}) / (\mathbf{t},\mathbf{t})$ (更新Omega ）

$(k)\ \phi^k = \phi^{k-1} + \alpha^k \mathbf{y} + \omega^k \mathbf{z}$ (用两次预矩阵求解结果改进被求量）

$(l)\ \mathbf{r}^k = \mathbf{s} - \omega^k \mathbf{t}$ (更新余量 ）

根据 $\omega^k, \mathbf{p}^k$ 等重复上述计算直到满足收敛条件。

Van der Vorst H A. BiCGSTAB: a fast and smoothy convergent variant of Bi-CG for the solution of non-symmetric linear system. SIAM J Sci Stat Comput. 1992, 13:631-644

5.3 SIP (强隐过程）方法简介

5.3.1 SIP迭代法的基本思想与求解步骤

5.3.2 L 与 U 矩阵的构造方法

5.3.3 SIP方法讨论

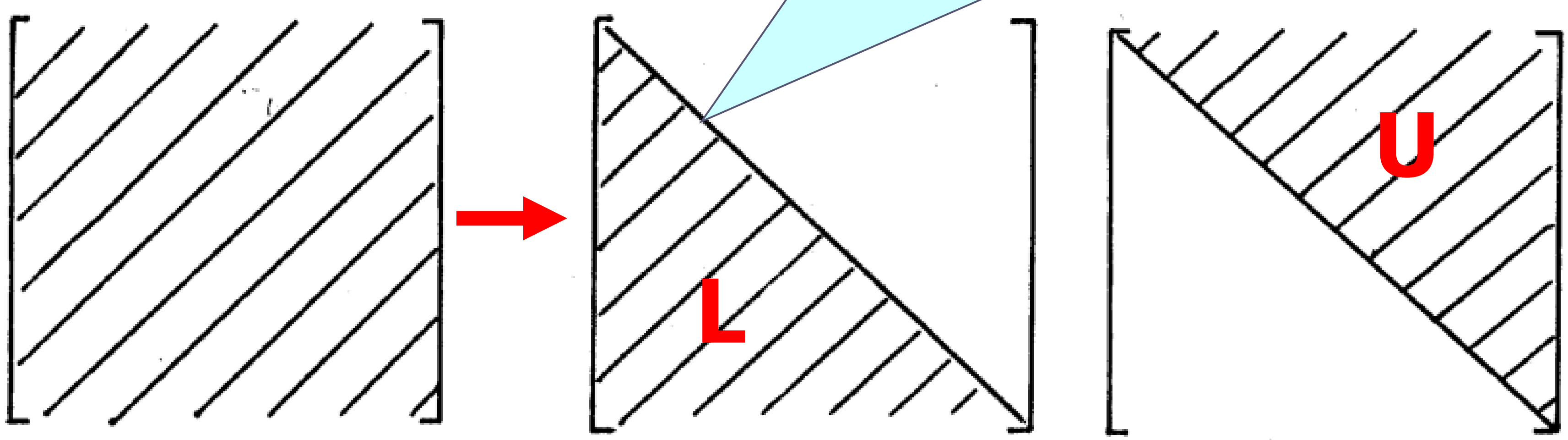
5.3 SIP（强隐过程）方法简介

5.3.1 SIP迭代法的基本思想

1. 上、下三角矩阵分解的求解方法

对一般的线性代数方程组LU分解是一种有效的直接求解方法：

单位下三角阵，对角元为1.0



$$\vec{A}\vec{\phi}=\vec{b} \longrightarrow \vec{L}\vec{U}\vec{\phi}=\vec{b}$$

$$\longrightarrow \vec{L}(\underline{\vec{U}\vec{\phi}})=\vec{b}$$

$$\vec{L}\underline{\vec{Y}}=\vec{b}$$

因为L矩阵中的对角元素为1，由此解出Y，相当于TDMA中的回代过程，

$$\vec{U}\vec{\phi}=\vec{Y}$$

由Y求出变量 $\vec{\phi}$

但对由流动与传热问题形成的代数方程组采用上下三角分解方法必须利用矩阵的稀疏特点： 例如对10000个未知数情形，对常规的代数方程系数矩阵，有10^8个不为零的系数，对这样的大型矩阵作LU分解计算工作量巨大无比；实际上对二维二阶格式，每一个未知数只有5个系数不为零。

$$a_P\phi_P = a_{EE}\phi_{EE} + a_E\phi_E + a_W\phi_W + a_{WW}\phi_{WW} + b$$

Stone在1968年提出的不完全上下分解方法就是用于稀疏矩阵的LU分解的。

所谓矩阵 $\vec{A}$ 的不完全分解，是指要构造上下三角矩阵L，U，它们满足下列关系：

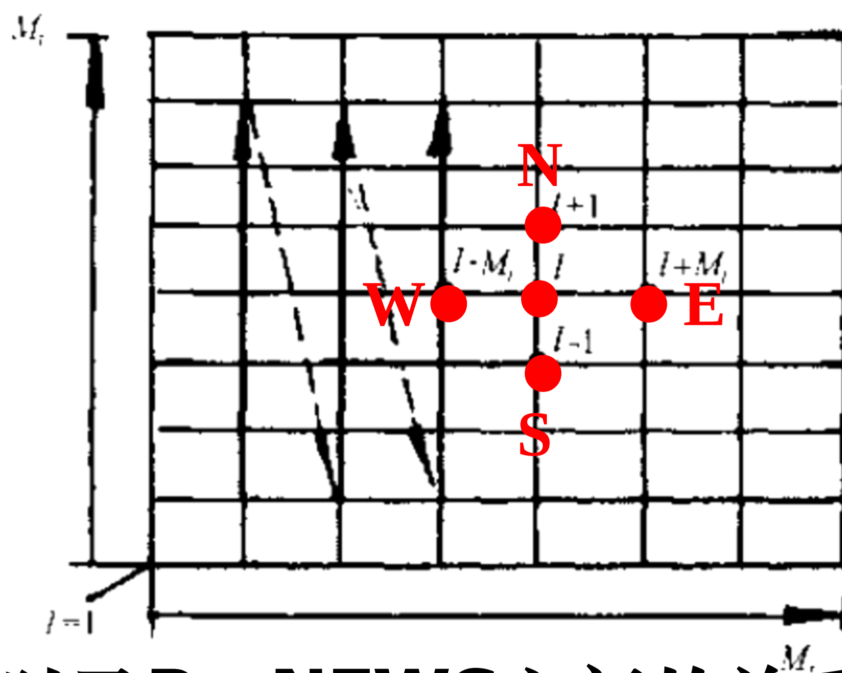
$$\vec{L}\vec{U} = \vec{A} + \vec{N} = \vec{M}$$ 其中 $\vec{N}$ 为“小阵”：

$$\vec{N}\vec{\phi} \cong 0$$

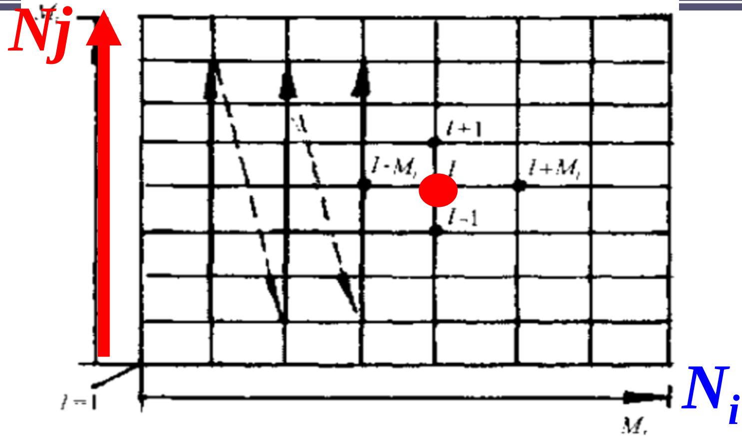
5.3.2 Stone提出的 L , U构造方法

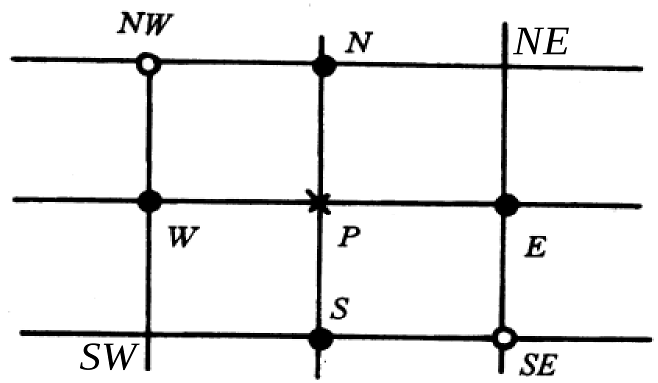
1. 1D压缩存储

W → S → P → N → E



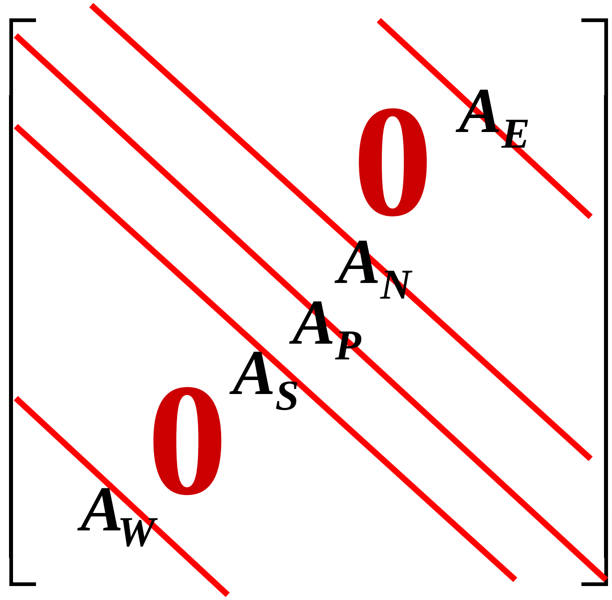
1D存储编号与二维节点（i,j）以及P－NEWS之间的关系:





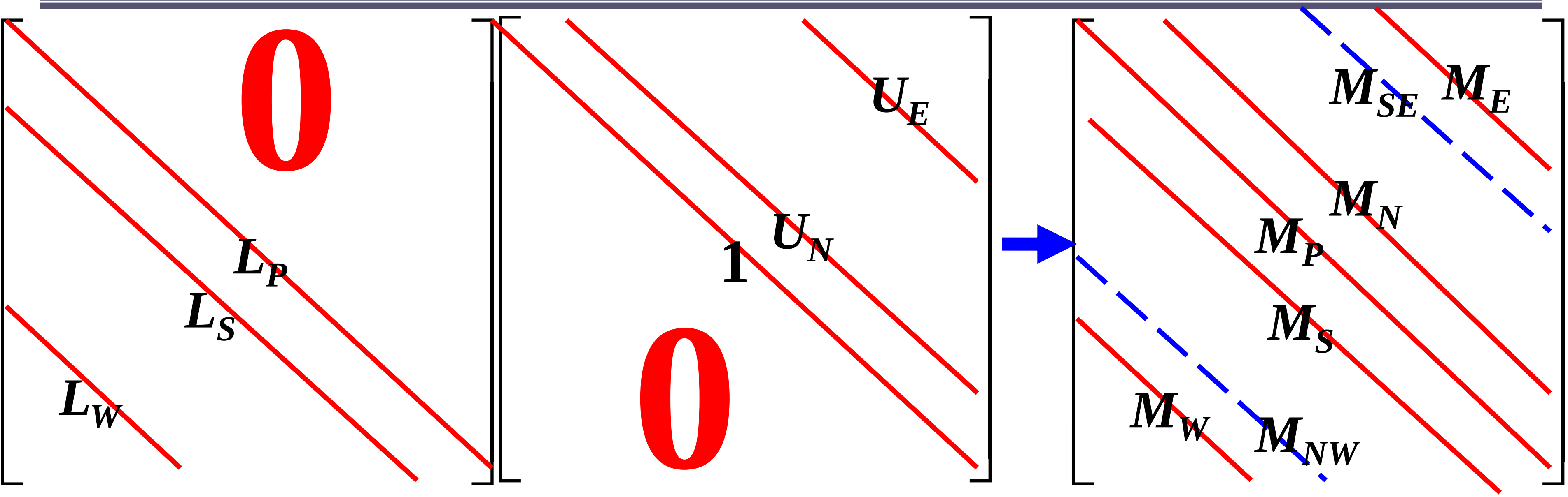
节点位置（*i,j*）	P-NEWS	**1D存储表示 (在1维中的位置)**
i, j	P	$l=(i-1)Nj+j$
i－1, *j*	W	$l-Nj$
i＋1, *j*	E	$l+Nj$
i, j －1	S	$l-1$
i, j+1	N	$l+1$
i-1,*j*＋1	NW	$l+1-Nj$
i+1,*j*+1	NE	$l+1+Nj$

2. Stone提出的构造L，U的步骤



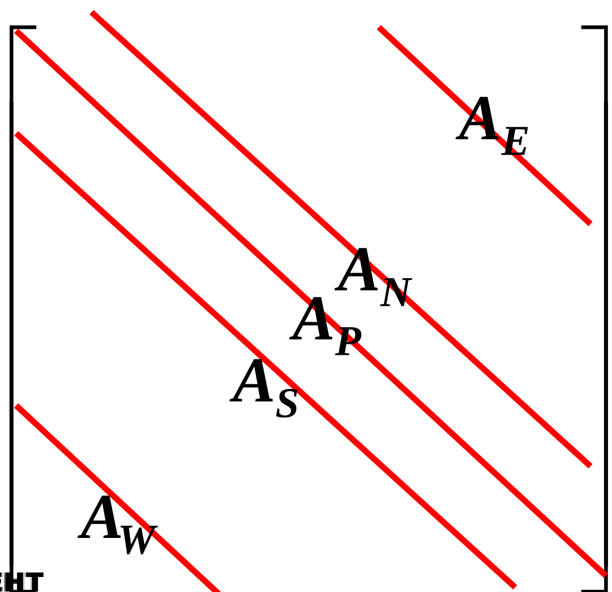
原矩阵的结构A

（1）L,U各为稀疏的三角阵，而且其非零元素的位置与A阵一致；



下三角阵 L　　上三角阵 U　　M＝A＋N

N 为小阵



如何确定LU中的非零元素?

或者如何确定小阵N?

（2）根据矩阵相乘原则可以确定M矩阵与L,U矩阵非零元素间的关系：

第 *l* 个变量的西侧系数

$$M_W^l = L_W^l \bullet 1; \qquad M_{NW}^l = L_W^l U_N^{l-N_j}$$

$$M_S^l = L_S^l \bullet 1; \qquad M_P^l = L_W^l U_E^{l-N_j} + L_S^l U_N^{l-1} + L_P^l$$

$$M_N^l = L_P^l U_N^l; \qquad M_{SE}^l = L_S^l U_E^{l-1}; \qquad M_E^l = L_P^l U_E^l \qquad \text{(a)}$$

因此，如果M阵元素已知，则可推出LU的元素。于是问题归结为求解M矩阵。

（3）因为M＝A+N，只要小阵非零元素确定即可；

不宜将M_{NW}及M_{SE}定义为小阵，收敛速度很慢；

认为小阵N也是7对角阵，且非零元素位置与M阵相同：$N_W, N_{SW}, N_S, N_P, N_N, N_{SE}, N_E$, 根据小阵的要求有：

$$N_P\phi_P + N_N\phi_N + N_S\phi_S + N_E\phi_E + N_W\phi_W + N_{NW}\phi_{NW} + N_{SE}\phi_{SE} \cong 0$$

如何确定这样一个小阵的7个非零元素?

设N_{NW}, N_{SE}分别等于M_{NW}, M_{SE}; 而且其它5条非零元素的作用就是抵消M_{NW}，M_{SE}的作用而使N成为小阵：

$$N_P\phi_P + N_N\phi_N + N_S\phi_S + N_E\phi_E + N_W\phi_W + \mathrm{M}_{NW}\phi_{NW} + \mathrm{M}_{SE}\phi_{SE} \cong 0$$

考虑到节点NW附近有N,W及P，SE附近有S,E及P;

可以设想上式中的最后两项用以下方式来抵消前5项：

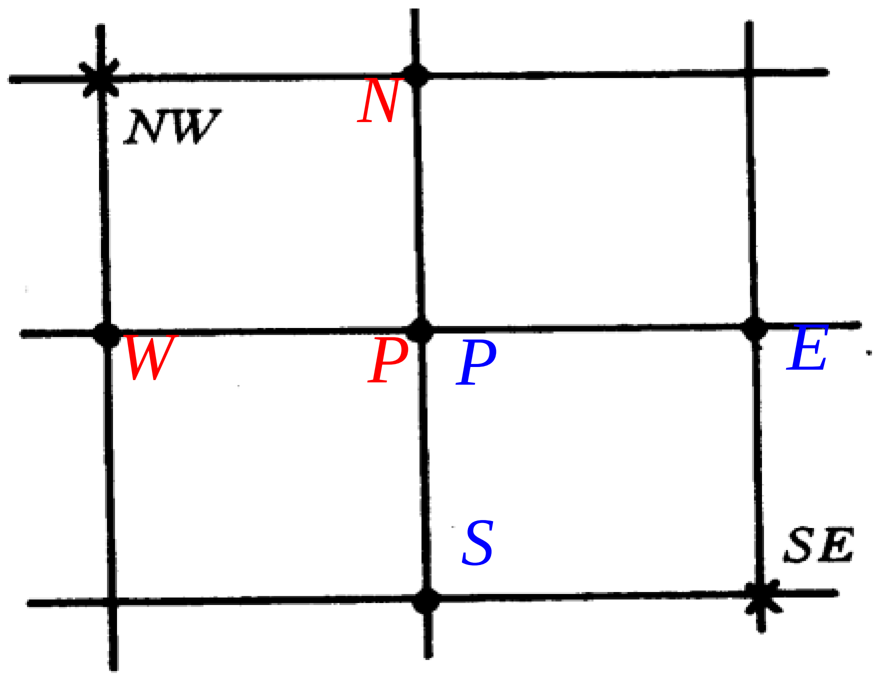
$$\mathrm{M}_{NW}(\phi_{NW} - \phi_{NW}^{*}) + \mathrm{M}_{SE}(\phi_{SE} - \phi_{SE}^{*}) \cong 0 \qquad \textbf{(b)}$$

ϕ_{NW}^{*} 反映了 (N,W,P) 的作用 ϕ_{SE}^{*} 反映了 (S,E,P) 的作用 $M_{NW}\phi_{NW}^{*}$ **以及** $M_{SE}\phi_{SE}^{*}$ **可以看成是 由其它非零元素用来抵消** $M_{NW}\phi_{NW}$ **以及** $M_{SE}\phi_{SE}$ **的。**

假设变量分布光滑，ϕ_{NW}^{*}, ϕ_{SE}^{*} 可由邻点的代数和来表示：

$$\phi_{NW}^{*} = \alpha(\phi_W + \phi_N - \phi_P) \quad \phi_{SE}^{*} = \alpha(\phi_S + \phi_E - \phi_P)$$

（c）



从NW引入N,W,P三点之值，从SE引入P,S,E三点之值，从而使7点之值均被引入到(b)式，为构造小阵的系数创造了条件。

Alfa 称为抵消参数，其值在0－1之间变化。

将(c) 代入 (b) 可确定小阵元素，再用式(a)，以及M=A+N的关系，最后可得L,U的系数：

$$L_W^l = A_W^l /(1+\alpha U_N^{l-Nj})$$

$$L_S^l = A_S^l /(1+\alpha U_E^{l-1})$$

$$L_P^l = A_P^l + \alpha(L_W^l U_N^{l-Nj} + L_S^l U_E^{l-1}) - L_W^l U_E^{l-Nj} - L_S^l U_N^{l-1}$$

$$U_N^l = A_N^l - \alpha L_W^l U_N^{l-Nj} / L_P^l$$

$$U_E^l = A_E^l - \alpha L_S^l U_E^{l-1} / L_P^l$$

5.3.3 Stone 强隐过程算法的求解过程

有代数方程组： $\vec{A}\vec{\phi}=\vec{b}$

设采用Stone方法获得了上下三角分解矩阵，则可采用以下方式**由第 k 次迭代推进到第 k+1 次迭代：**

迭代收敛时有： $\vec{A}\vec{\phi}=\vec{b}$，即有: $\vec{b}-\vec{A}\vec{\phi}=0$

$$(\vec{A}+\vec{N})\vec{\phi}=(\vec{A}+\vec{N})\vec{\phi}+(\vec{b}-\vec{A}\vec{\phi})$$

迭代快收敛时: $\underline{(\vec{A}+\vec{N})}\vec{\phi}^{k+1}=\underline{(\vec{A}+\vec{N})}\vec{\phi}^{k}+(\vec{b}-\vec{A}\vec{\phi}^{k})$

即 $$(\vec{A}+\vec{N})(\underbrace{\vec{\phi}^{k+1}-\vec{\phi}^{k}}_{\vec{\delta}^{k+1}})=(\underbrace{\vec{b}-\vec{A}\vec{\phi}^{k}}_{\vec{r}^{k}})$$

即 $$(\vec{A}+\vec{N})\vec{\delta}^{k+1}=\vec{r}^{k}$$

因此有: $$(\vec{L}\vec{U})\vec{\delta}^{k+1}=\vec{r}^{k}$$

可通过如下两步计算解出 $\vec{\delta}$:

$$\vec{L}(\vec{U}\vec{\delta}^{k+1})=\vec{r}^{k};\quad \vec{L}\vec{R}^{k}=\vec{r}^{k};\quad \vec{R}^{k}=\vec{L}^{-1}\vec{r}^{k}$$

$$(\vec{U}\vec{\delta}^{k+1})=\vec{R}^{k};\quad \vec{\delta}^{k+1}=\vec{U}^{-1}\vec{R}^{k}$$

于是改进值: $\vec{\phi}^{k+1} = \vec{\phi}^{k} + \vec{\delta}^{k+1}$

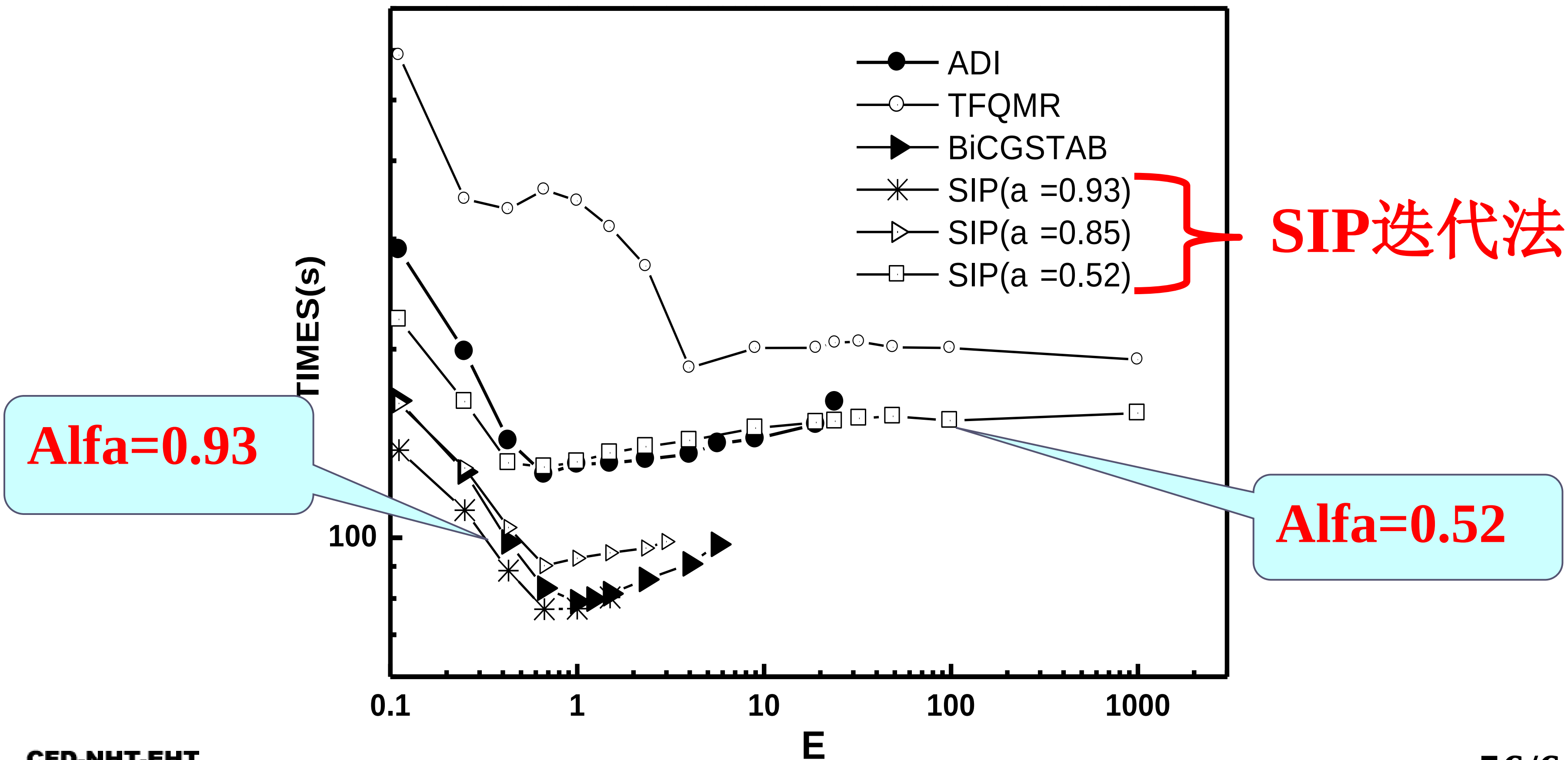
两点说明:

1）这一求解方法总体上是迭代的，即从假设的初场出发不断改进；但每一步又是直接解法－上下三角矩阵分解。所谓“强隐”(Strong implicit)就是指用直接解法来获得每一步的修正值。

2）解的改进采用迭加修正值的方式进行，对第一类边界条件，边界值已知，修正值为零；**因此在LU矩阵中凡是与边界相关的系数一律取为零。**

5.3.4 SIP迭代法讨论

小阵非零元素的构造方式对收敛解没有影响，但关系到收敛的快慢：



5.4 多重网格方法简介

5.4.1 两种选择

5.4.2 在疏网格上求解的是密网格上的近似值

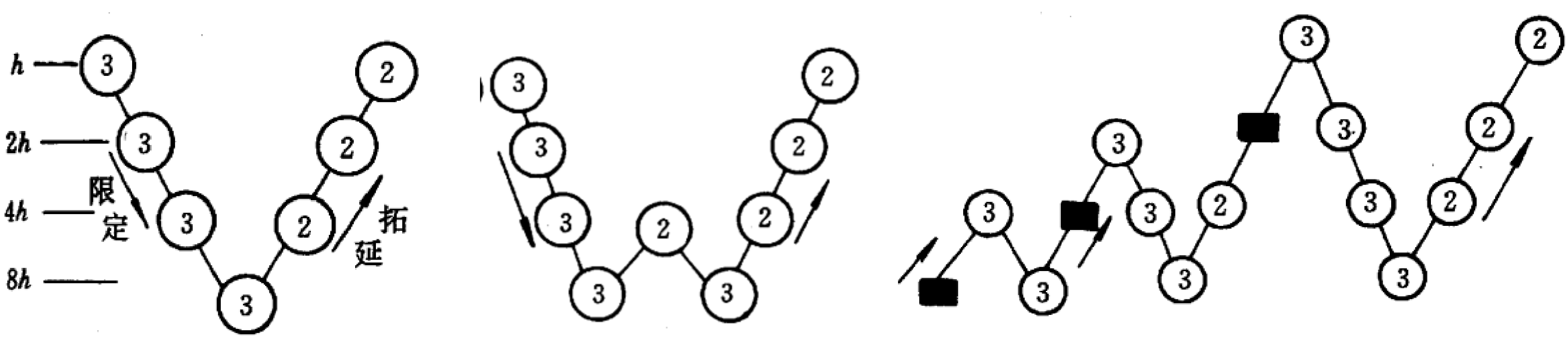
5.4.3 流场求解中采用多重网格的一些特殊问题

5.4 多重网格方法简介

5.4.1 两种选择

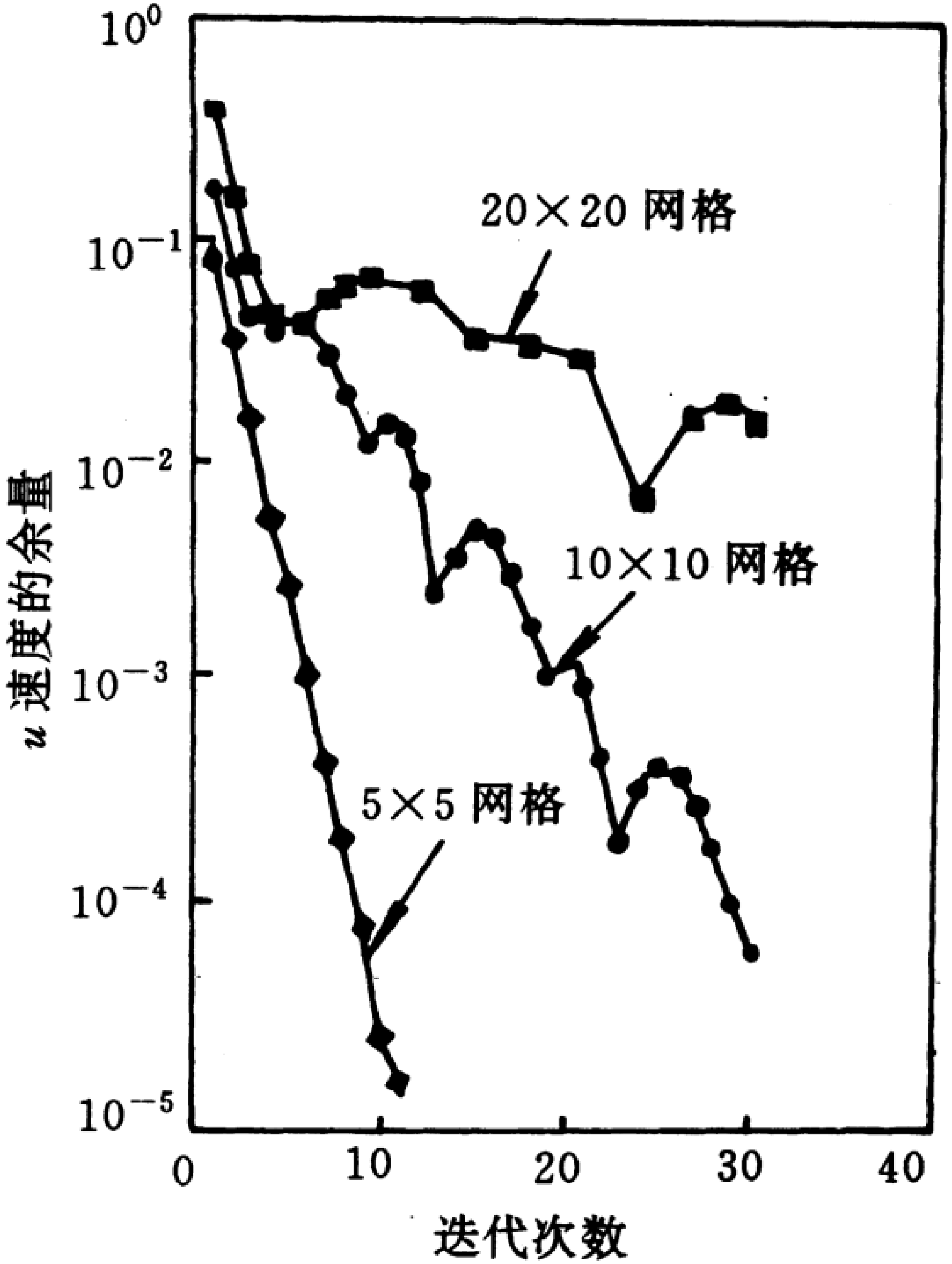
1. 求解变量的选择：求解修正值(CS)；求解变量本身(FAS)；

2. 循环方式选择：V型，W型，FMG型



(a) V(3.2)循环 (b) W(3.2)循环 (c) FMG V(3,2)循环

网格疏密对收敛速度影响的例子



5.4.2 在疏网格上求解的是密网格上的近似值

$$\vec{A}^{(k-1)}\phi^{(k-1)}=\vec{b}^{(k-1)}+I_k^{k-1}(\underline{\vec{b}^{(k)}-\vec{A}^{(k)}\phi^{(k)}})$$

密网格上的余量

将k网格上的值传递到(k-1)上的算子

在k-1网格上按k网格的解计算的系数矩阵

在k-1网格上按k网格的解计算所得的源项

在k-1网格上的k网格的解

图 从密网格向疏网格解的传递规律

$$\phi_{rev}^{(k)} = \phi_{old}^{k} + I_{k-1}^{k}\left(\phi^{(k-1)} - I_{k}^{k-1}\phi_{old}^{(k)}\right)$$

密网格上原来的解传递到k-1网格上的值

改进后密网格上的解

k-1网格上得到的密网格的新解

密网格上原来的解

在k-1网格上密网格解的修正量传递到密网格上

图 从疏网格向密网格解的传递规律

5.4.3 流场求解中采用多重网格的一些特殊问题

1.对流项格式选择：

亏损修正(defect correction)：疏网格低阶，密网格高阶（规定外迭代的次数）;建议采用SGSD;

2.变量求解顺序：

与单重网格上一样：*u*, *v*, *p*’, *T* ；

3.网格切换依据：

简单地规定外迭代的次数，一般为1–5次;

规定外迭代的次数与余量范数变化相结合的方法。

4.网格切换实际上实现了非线性问题的一次外迭代

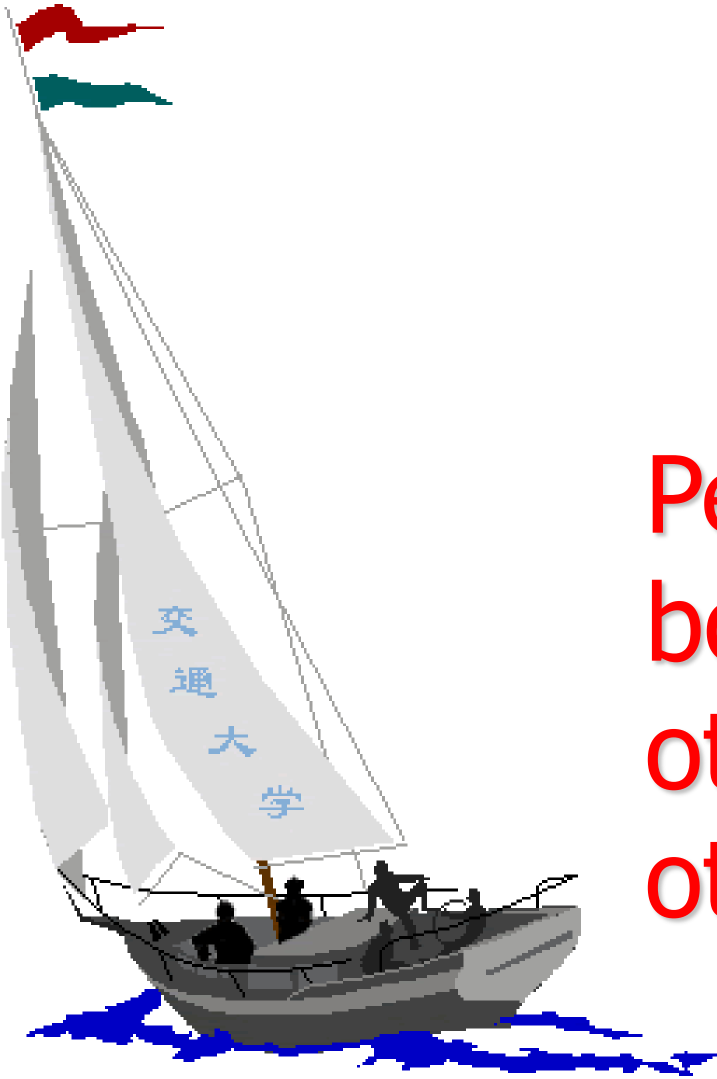
切换网格后要在新网格上重新计算离散方程系数，相当于一次外迭代。

[2] 张玮，王元. 西安交通大学学报，2001，35（7）：670－674.

[3]马亮栋，等. 西安交通大学学报，2005，39（7）：975－977.

[4]金巍巍，等. 西安交通大学学报，2005，39（9）：966－968.

[5]金巍巍，等. 工程热物理学报，2007，28（3）：478－480.



同舟共济
渡彼岸！

People in the same boat help each other to cross to the other bank, where….