椭圆型差分方程的并行迭代算法* 1)

刘 兴 平

(北京应用物理与计算数学研究所)

PARALLEL LTERATIVE ALGORITHMS FOR ELLIPTIC DIFFERENCE EQUATIONS

Liu Xing-ping

(Institute of Applied Physics and Computational Mathematics, Beijing)

Abstract

In this paper, a parallel strongly implicit procedure iterative method for solving a system of algebraic equations Ax=b is proposed. The rate of convergence is analyzed under the condition that the coefficient matrix A is strongly diagonally dominant. Numerical examples are given.

§1. 引 言

由于当代超级计算机技术日新月异地发展，研究出适应超级计算机运行的高效的计算方法是当务之急．现在已有许多著名学者对三对角方程组的解法(如文献[1,2,3])进行了讨论或用多色排序形成的线性代数方程组 $Ax=b$ 使某些算法可并行计算(如文献[9])，但在许多实际问题中往往是多对角的或差分方程的求解区域是不规则区域形成的矩阵．矩阵性质也是多种多样的，所以前一种方法就不能用了，后一种方法就要求工程技术人员在使用并行算法前必须对多色排序有足够的认识和对矩阵非零元素的分布进行必要的研究．然而对于使用者来说又增加了不少的工作量，不易被接受．SIP 算法是所有串行算法中比较好的一种算法，但它仅有 50% 左右的计算量可向量或并行化处理，另 50%的计算量只能作串行计算，即 SIP 算法向量或并行化处理的主要困难是 $(LU)^{-1}r^{(s)}$ 的计算．本文第二节提出一实用并行迭代算法，可以克服 $(LU)^{-1}r^{(s)}$ 向量或并行化处理的困难．这种算法格式简单明了，收敛速度快．第三节讨论了收敛性，第四节给出了计算实例．

* 1991 年 1 月 21 日收到.

1) 国家自然科学基金资助项目.

§2. 新 算 法

我们设线性代数方程组

$$Ax=b, \tag{2.1}$$

其中

$$A=\begin{bmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & C_{m-1} \\ & & & A_m & B_m \end{bmatrix}, \tag{2.2}$$

$$\left\{\begin{aligned} &B_i=\begin{bmatrix} b_{i1} & c_{i1} & & & \\ a_{i2} & b_{i2} & c_{i2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{i,n-1} \\ & & & a_{i,n} & b_{i,n} \end{bmatrix}, \quad A_i=\begin{bmatrix} \beta_{i1} & \gamma_{i1} & & & \\ \alpha_{i2} & \beta_{i2} & \gamma_{i2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_{i,n-1} \\ & & & \alpha_{i,n} & \beta_{i,n} \end{bmatrix}, \\ &C_i=\begin{bmatrix} q_{i1} & r_{i1} & & & \\ p_{i2} & q_{i2} & r_{i2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & r_{i,n-1} \\ & & & p_{i,n} & q_{i,n} \end{bmatrix}. \end{aligned}\right. \tag{2.3}$$

又设 A 的不完全 LU 分解矩阵(即 $A\approx LU$) 如下：

$$L=\begin{bmatrix} L_1 & & & \\ K_2 & L_2 & & \\ & \ddots & \ddots & \\ & & K_m & L_m \end{bmatrix}, \quad U=\begin{bmatrix} U_1 & V_1 & & \\ & U_2 & V_2 & \\ & & \ddots & \ddots \\ & & & \ddots & V_{m-1} \\ & & & & U_m \end{bmatrix}, \tag{2.4}$$

其中

$$\left\{\begin{aligned} &K_i=\begin{bmatrix} e_{i1} & f_{i1} & & & \\ d_{i2} & e_{i2} & f_{i2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & f_{i,n-1} \\ & & & d_{i,n} & e_{i,n} \end{bmatrix}, \quad L_i=\begin{bmatrix} l_{i1} & & & \\ m_{i2} & l_{i2} & & \\ & \ddots & \ddots & \\ & & m_{i,n} & l_{i,n} \end{bmatrix}, \end{aligned}\right. \tag{2.5}$$

$$\left\{\begin{array}{l} U_i = \begin{bmatrix} 1 & u_{i1} & & & \\ & 1 & u_{i2} & & \\ & & \ddots & \ddots & \\ & & & \ddots & u_{i,n-1} \\ & & & & 1 \end{bmatrix}, \quad V_i = \begin{bmatrix} w_{i1} & t_{i1} & & & \\ s_{i2} & w_{i2} & t_{i2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & t_{i,n-1} \\ & & & s_{i,n} & w_{i,n} \end{bmatrix}. \end{array}\right.$$

由[5,11,14,15,16]且有如下递推关系式：

$$\begin{cases} d_{ij} = \alpha_{ij} - \theta d_{ij} s_{i-1,j-1}, \\ e_{ij} = \beta_{ij} - d_{ij} u_{i-1,j-1} - \theta d_{ij} s_{i-1,j-1}, \\ f_{ij} = \gamma_{ij} - e_{ij} u_{i-1,j-1} - \theta f_{ij} u_{i-1,j+1}, \end{cases}$$

$$\begin{cases} m_{ij} = a_{ij} - d_{ij} w_{i-1,j-1} - e_{ij} s_{i-1,j} - \theta m_{ij} u_{i,j-1}, \\ l_{ij} = b_{ij} - d_{ij} t_{i-1,j-1} - e_{ij} w_{i-1,j+1} - f_{ij} s_{i-1,j+1} - m_{ij} u_{i,j-1} \\ \qquad + \theta(d_{ij} s_{i-1,j-1} + f_{ij} u_{i-1,j+1} + f_{ij} t_{i-1,j+1} + m_{ij} s_{i,j-1}), \\ u_{ij} = (c_{ij} - e_{ij} t_{i-1,j} - f_{ij} w_{i-1,j+1} - \theta f_{ij} u_{i-1,j+1})/l_{ij}, \end{cases} \tag{2.6}$$

$$\begin{cases} s_{ij} = (p_{ij} - m_{ij} w_{i,j-1} - \theta m_{ij} s_{i,j-1})/l_{ij}, \\ w_{ij} = (q_{ij} - m_{ij} t_{i,j-1} - \theta f_{ij} t_{i-1,j+1})/l_{ij}, \\ t_{ij} = (r_{ij} - \theta f_{ij} t_{i-1,j+1})/l_{ij}, \end{cases}$$

其中 θ 为一参数．取(2.3)和(2.5)中的 $\gamma_{ij} = r_{ij} = f_{ij} = s_{ij} = 0$ 则得到七对角阵的递推关系式

$$\begin{cases} d_{ij} = \alpha_{ij}, \\ e_{ij} = \beta_{ij} - d_{ij} u_{i-1,j-1} - \theta e_{ij} u_{i-1,j}, \\ m_{ij} = a_{ij} - d_{ij} w_{i-1,j-1} - \theta m_{ij} w_{i,j-1}, \\ l_{ij} = b_{ij} - d_{ij} t_{i-1,j-1} - e_{ij} w_{i-1,j} - m_{ij} u_{i,j-1} + \theta(e_{ij} u_{i-1,j} + m_{ij} w_{i,j-1}), \\ u_{ij} = (c_{ij} - e_{ij} t_{i-1,j} - \theta e_{ij} u_{i-1,j})/l_{ij}, \\ w_{ij} = (q_{ij} - m_{ij} t_{i,j-1} - \theta m_{ij} w_{i,j-1})/l_{ij}, \\ t_{ij} = r_{ij}/l_{ij}. \end{cases} \tag{2.7}$$

取(2.3)和(2.5)中的 $\alpha_{ij} = r_{ij} = d_{ij} = t_{ij} = 0$ 则得到另一七对角阵的递推关系式

$$\begin{cases} e_{ij} = \beta_{ij}, \\ f_{ij} = \gamma_{ij} - e_{ij} u_{i-1,j} - \theta f_{ij} u_{i-1,j+1}, \\ m_{ij} = a_{ij} - e_{ij} s_{i-1,j} - \theta m_{ij} u_{i,j-1}, \\ l_{ij} = b_{ij} - e_{ij} w_{i-1,j} - f_{ij} s_{i-1,j+1} - m_{ij} u_{i,j-1} + \theta(f_{ij} u_{i-1,j+1} + m_{ij} s_{i,j-1}), \\ u_{ij} = (c_{ij} - f_{ij} w_{i-1,j+1} - \theta f_{ij} u_{i-1,j+1})/l_{ij}, \\ s_{ij} = (p_{ij} - m_{ij} w_{i,j-1} - \theta m_{ij} s_{i,j-1})/l_{ij}, \\ w_{ij} = q_{ij}/l_{ij}. \end{cases} \tag{2.8}$$

分别称它们为 7_1 和 7_2 对角阵的递推关系式． 若取 (2.3) 和 (2.5) 中的 $\alpha_{ij} = \gamma_{ij} = p_{ij} = r_{ij} = d_{ij} = f_{ij} = s_{ij} = t_{ij}$ 则得到五对角阵的递推关系式

$$
\begin{cases}
e_{ij} = \beta_{ij} - \theta e_{ij} u_{i-1,j}, \\
m_{ij} = a_{ij} - \theta m_{ij} w_{i,j-1}, \\
l_{ij} = b_{ij} - e_{ij} w_{i-1,j} - m_{ij} w_{i,j-1} + \theta(e_{ij} u_{i-1,j} + m_{ij} w_{i,j-1}), \\
u_{ij} = (c_{ij} - \theta e_{ij} u_{i-1,j})/l_{ij}, \\
w_{ij} = (q_{ij} - \theta m_{ij} w_{i,j-1})/l_{ij}.
\end{cases} \tag{2.9}
$$

为了解线性代数方程组(2.1)我们取算法

$$
\begin{cases}
r^{(s)} = b - Ax^{(s)}, & s = 0,1,\cdots, \\
p^{(s)} = Mr^{(s)}, & s = 0,1,\cdots, \\
x^{(s+1)} = x^{(s)} + p^{(s)}, & s = 0,1,\cdots.
\end{cases} \tag{2.10}
$$

它的迭代矩阵是

$$
S_{\text{psip}} = I - MA, \tag{2.11}
$$

其中

$$
\begin{cases}
M = (I + \tilde{U} + \cdots + \tilde{U}^l)(I + \tilde{L} + \cdots + \tilde{L}^l)D_1^{-1}, \\
\tilde{L} = I - D_1^{-1}L, \tilde{U} = I - U, \ D_1 = \operatorname{diag} L,
\end{cases} \tag{2.12}
$$

L 和 U 分别是(2.4)中的 L 和 U.

我们把 (2.10) 称之为 PSIP (Parallel Strongly Implicit Procedure) 算法. 在 k 个 CPU 且每个 CPU 是一个向量处理机(如 CRAY 系列机或 YH 系列机)上,计算步骤如下:

```
step1  [r^(s)]_k = [b - Ax^(s)]_k
step2  [y^(s)]_k = [r^(s)/D_1]_k
step3  do a j = 1,l
       [y_1^(s)]_k = [L~ y^(s)]_k
       IF(j > 1) go to b
       [y_2^(s)]_k = [y^(s)]_k + [y_1^(s)]_k
       [y^(s)]_k = [y_1^(s)]_k
       go to a
   b   [y_2^(s)]_k = [y_2^(s)]_k + [y_1^(s)]_k
       [y^(s)]_k = [y_1^(s)]_k
   a   continue
step4  do cc j = 1, l
       [y_3^(s)]_k = [U~ y_2^(s)]_k
       if(j > 1) go to c
       [p^(s)]_k = [y_2^(s)]_k + [y_3^(s)]_k
       [y_2^(s)]_k = [y_3^(s)]_k
       go to cc
   c   [p^(s)]_k = [p^(s)]_k + [y_3^(s)]_k
       [y_2^(s)]_k = [y_3^(s)]_k.
  cc   continue
step5  [x^(s+1)]_k = [x^(s)]_k + [p^(s)]_k
```

$[\cdot]_k$ 表示用 k 个 CPU 同时计算，每个 CPU 计算的工作量是该表达式的 k 分之一. 例如, $[\tilde{U}y_2^{(s)}]_k$ 是 $\tilde{U}$ 的全体行数的前 k 分之一乘向量 $y_2^{(s)}$ 送到第一个处理机,第二个 k 分之一乘向量 $y_2^{(s)}$ 送到第二个处理机, 依此类推, 直到最后一个 k 分之一乘向量 $y_2^{(s)}$ 送到最后一个处理机.

我们还可以用同样的思想得到 PPSD, PGCR, PPCG 等等各种适用于并行的算法. 另外,从上面的并行执行步骤中可以看出, 在这个算法中没有对前处理 LU 分解进行并行化处理,关于 LU 分解的并行问题有待进一步的研究,目前是不能并行的;但它对整个算法的并行性影响不大,因为在整个计算过程中,它只计算一次, 只占全体计算量的极少部分.

§3. 收敛性

在这一节我们设 A 是强对角优势矩阵,即矩阵元素满足

$$|b_{ij}| > |\alpha_{ij}| + |\beta_{ij}| + |\gamma_{ij}| + |a_{ij}| + |b_{ij}| + |c_{ij}| + |p_{ij}| + |q_{ij}| + |r_{ij}|. \tag{3.1}$$

不失一般性,我们设

$$\begin{cases} b_{ij} = 1,\ \alpha_{ij} = O(k),\ \beta_{ij} = O(z),\ \gamma_{ij} = O(k),\ a_{ij} = O(h), \\ c_{ij} = O(h),\ p_{ij} = O(k),\ q_{ij} = O(z),\ r_{ij} = O(k), \end{cases} \tag{3.2}$$

则我们有:

定理 1. 若 A 是强对角优势矩阵,则 PSIP 算法和 SIP 算法迭代矩阵的谱半径满足如下关系式:

$$\begin{cases} \rho_{SIP9} \leqslant O(hk),\ \ \rho_{PSIP9} \leqslant O((h+k+z)^{l+1} + hk), \\ \rho_{SIP7_2} \leqslant O(hk),\ \ \rho_{PSIP7_2} \leqslant O((h+k+z)^{l+1} + hk), \\ \rho_{SIP7_1} \leqslant O(hz),\ \ \rho_{PSIP7_1} \leqslant O((h+k+z)^{l+1} + hz), \\ \rho_{SIP5} \leqslant O(hz),\ \ \rho_{PSIP5} \leqslant O((h+z)^{l+1} + hz), \end{cases} \tag{3.3}$$

其中 ρ 下标中的数字分别表示九点,两种类型的七点和五点 SIP 和 PSIP 格式。

证明. ρ_{SIP9}, ρ_{SIP7_1}, ρ_{SIP7_2}, ρ_{SIP5} 的证明见文献[5,11,14,15].

现在我们证 ρ_{PSIP9}, ρ_{PSIP7_1}, ρ_{PSIP7_2}, ρ_{PSIP5} 的估计式. 记 $N = LU - A$,则(2.12)变为

$$\begin{aligned} S_{PSIP9} &= I - \left(\sum_{i=0}^{l} \widetilde{U}^i\right)\left(\sum_{i=0}^{l} \widetilde{L}^i\right) D_1^{-1}(LU - N) \\ &= I - (I - \widetilde{U})^{-1}(I - \widetilde{U}^{l+1})(I - \widetilde{L}^{l+1})(I - \widetilde{L})^{-1}D_1^{-1}(LU - N) \\ &= I - U^{-1}(I - \widetilde{U}^{l+1})(I - \widetilde{L}^{l+1})(U - L^{-1}N), \end{aligned}$$

相似于

$$\begin{aligned} &I - (I - \widetilde{U}^{l+1})(I - \widetilde{L}^{l+1})(I - L^{-1}NU^{-1}) \\ &\quad = \widetilde{U}^{l+1} + \widetilde{L}^{l+1} - \widetilde{U}^{l+1}\widetilde{L}^{l+1} + (I - \widetilde{U}^{l+1})(I - \widetilde{L}^{l+1})L^{-1}NU^{-1}, \end{aligned} \tag{3.4}$$

所以

$$\begin{aligned} \rho_{PSIP9} &\leqslant \|\widetilde{U}^{l+1} + \widetilde{L}^{l+1} - \widetilde{U}^{l+1}\widetilde{L}^{l+1} + (I - \widetilde{U}^{l+1})(I - \widetilde{L}^{l+1})L^{-1}NU^{-1}\|_\infty \\ &\leqslant \|\widetilde{U}^{l+1} + \widetilde{L}^{l+1}\|_\infty + \|\widetilde{U}^{l+1}\widetilde{L}^{l+1}\|_\infty + \|(I - \widetilde{U}^{l+1})(I - \widetilde{L}^{l+1})L^{-1}NU^{-1}\|_\infty \\ &\leqslant \|\widetilde{U}^{l+1} + \widetilde{L}^{l+1}\|_\infty + \|\widetilde{U}^{l+1}\widetilde{L}^{l+1}\|_\infty \\ &\quad + \|(I - \widetilde{U}^{l+1})\|_\infty \|(I - \widetilde{L}^{l+1})\|_\infty \|L^{-1}NU^{-1}\|_\infty. \end{aligned} \tag{3.5}$$

由文献[5,11,14,15]我们有

$$\begin{cases} \|\widetilde{U}^{l+1} + \widetilde{L}^{l+1}\| \leqslant O((h+z+k)^{l+1}), \\ \|\widetilde{U}^{l+1}\widetilde{L}^{l+1}\| \leqslant O((h+z+k)^{2(l+1)}), \\ \|I - \widetilde{U}^{l+1}\| \leqslant 1 + O((h+z+k)^{l+1}), \\ \|I - \widetilde{L}^{l+1}\| \leqslant 1 + O((h+z+k)^{l+1}), \end{cases} \tag{3.6}$$

$$\|L^{-1}NU^{-1}\| \leqslant O(hk), \tag{3.7}$$

所以

$$\rho_{PSIP_9} \leqslant O((h + z + k)^{l+1} + hk). \tag{3.8}$$

我们可以以同样的方式证明

$$\begin{cases} \rho_{PSIP7_1} \leqslant O((h + k + z)^{l+1} + hk) \quad \rho_{PSIP7_2} \leqslant O((h + k + z)^{l+1} + hz), \\ \rho_{PSIP5} \leqslant O((k + z)^{l+1} + hz). \end{cases} \tag{3.9}$$

定理证毕。

从定理 1 我们可以看到，PSIP 算法的谱半径 ρ 与 l, h, k, z 的关系。显然谱半径 ρ 是 l 的递减函数,即当 l 逐渐增大时，ρ 就逐渐递减,直至与 SIP 的谱半径一样。

另一方面,每迭代一次的计算量也随着 l 的增大而增加。l, ρ 和每迭代一次的计算量见表 1。事实上,我们从表 1 可以看到九对角阵,当 l 大到使 $(h + z + k)^{l+1} \ll hk$ 时，l 对迭代次数的影响越来越小，而计算量的增加速度没有减慢，最后将失去并行的优越性。所以 l 应取在使 $(h + z + k)^{l+1}$ 约小于 hk 时比较好,对七，五对角阵也有同样的性质。从微分方程(2.2)和差分方程(3.1),我们可以看到,当 $C_x \approx C_y$ 时，$l = 5$ 比较理想。

表 1　计算量与谱半径的关系

l	KN	$\rho \leqslant$	乘法数	除法数
1	1	$O((h + z + k)^2 + hk)$	$17n^2$	n^2
	2	$O((h + z + k)^2 + hz)$	$13n^2$	n^2
	3	$O((h + z + k)^2 + hk)$	$13n^2$	n^2
	4	$O((h + z)^2 + hz)$	$9n^2$	n^2
2	1	$O((h + z + k)^3 + hk)$	$25n^2$	n^2
	2	$O((h + z + k)^3 + hz)$	$19n^2$	n^2
	3	$O((h + z + k)^3 + hk)$	$19n^2$	n^2
	4	$O((h + z)^3 + hz)$	$13n^2$	n^2
3	1	$O((h + z + k)^4 + hk)$	$33n^2$	n^2
	2	$O((h + z + k)^4 + hz)$	$25n^2$	n^2
	3	$O((h + z + k)^4 + hk)$	$25n^2$	n^2
	4	$O((h + z)^4 + hz)$	$17n^2$	n^2
$l>4$	1	$O((h + z + k)^{l+1} + hk)$	$(9 + 8l)n^2$	n^2
	2	$O((h + z + k)^{l+1} + hz)$	$(7 + 6l)n^2$	n^2
	3	$O((h + z + k)^{l+1} + hk)$	$(6 + l)n^2$	n^2
	4	$O((h + z)^{l+1} + hz)$	$(5 + 4l)n^2$	n^2

表中 KN = 1,2,3,4 分别表示矩阵 A 是九对角,七对角的两种类型和五对角阵。

§4. 数 值 结 果

由于受条件的限制,我们的数值实验不能在多处理机上作,对算法的优越性不能用数值实验充分的反映出来,这是一个极大的遗憾。有幸的是，我们可以在 YH-1 机上作数值实验,虽然这个数值实验不能充分的反映出该算法的优越性,但也能说明一些问题。

我们取微分方程

$$C_x \frac{\partial^2 u}{\partial x^2} + C_y \frac{\partial^2 u}{\partial y^2} + (C_1 \sin 2\pi x + C_2) \frac{\partial u}{\partial x} + (D_1 \sin 2\pi y + D_2) \frac{\partial u}{\partial y} + E u = 0,$$

$$0 \leqslant x,\ y \leqslant 1.$$

边界条件是

$$\left.\frac{\partial u}{\partial y}\right|_{y=0} = 0,\quad \left.\frac{\partial u}{\partial y}\right|_{y=1} = 0,\quad u|_{x=0} = 0,\quad u|_{x=1} = 10 + \cos \pi y,$$

这里 $C_x, C_y, C_1, C_2, D_1, D_2$ 和 E 是常数。计算结果如下:

表 2 并行各算法的比较

	J2P*		J		BJ		PSIP	
KN	s	CPU	s	CPU	s	CPU	s	CPU
$C_1=0,\ C_2=0,\ D_1=0,\ D_2=0,\ E=0,\ C_x=1,\ C_y=1\ h^{-1}=31$								
1	436	5.15	2902	10.82	2185	8.99	157	2.943
2	626	5.97	4169	13.12	2193	7.76	211	3.651
3	626	5.96	4167	13.12	2193	7.75	101	2.4667
4	626	4.52	4168	10.73	2192	6.49	234	2.687
$C_1=0,\ C_2=0,\ D_1=0,\ D_2=0,\ E=0,\ C_x=1,\ C_y=1\ h^{-1}=61$								
1	1507	48.05	+	+	7592	86.54	875	43.43
2	2177	56.54	+	+	7598	75.27	822	33.47
3	2177	56.47	+	+	7598	75.23	779	31.64
4	2176	43.51	+	+	7598	63.94	822	25.98
$C_1=1,\ C_2=1,\ D_1=0,\ D_2=0,\ E=0,\ C_x=1,\ C_y=1\ h^{-1}=31$								
1	488	5.76	32[illegible]	12.12	2447	10.06	176	3.281
2	703	6.71	4665	14.68	2454	8.68	260	3.904
3	703	6.69	4663	14.70	2454	8.68	103	2.512
4	703	5.08	4663	12.00	2454	7.26	260	2.982
$C_1=1,\ C_2=1,\ D_1=0,\ D_2=0,\ E=0,\ C_x=1,\ C_y=1\ h^{-1}=61$								
1	1689	53.85	+	+	8479	96.64	626	31.15
2	2438	63.31	+	+	8485	84.05	920	37.41
3	2438	63.20	+	+	8485	84.00	872	35.39
4	2438	48.75	+	+	8485	71.38	920	29.05

注 * $x^{(s+1)} = x^{(s)} - \omega M^{-1}(Ax^{(s)} - b)$,[6].

这里 $M^{-1} = (I + B + B^2 + \cdots + B^{2k-1})D^{-1}$ 和 $B = I - D^{-1}A$ 表中的 $k = 2$

注 1. 表中"+"表示迭代次数超过 9500 次, h 为网格大小, SOR 和 BSOR 方法的参数取 1.6; SIP 和 J2P 方法的参数是最优参数, PSIP 方法的 $l = 5$, s 和 CPU 分别是满足迭代控制误差 $\left\|\frac{u^{(s+1)} - u^{(s)}}{u^{(s+1)}}\right\|_\infty < 10^{-7}$ 的迭代次数和时间,单位是秒(YH 机)。

通过大量的计算证明,五对角矩阵,当 $(h+z)^{l+1} > hz$ 时, l 每增加 1, 收敛速度可以大大的提高, CPU 也大大的降低。当 $(h+z)^{l+1} < hz$ 时, l 每增加 1, 收敛速度也

表 3　串行各算法与并行算法 PSIP 的比较

	SOR		BSOR		PE		SBGS		SIP		PSIP	
KN	s	CPU	s	CPU	s	CPU	s	CPU	s	CPU	s	CPU
$C_1=0,\ C_2=0,\ D_1=0,\ D_2=0,\ E=0,\ C_x=1,\ C_y=1\ h^{-1}=31$												
1	433	32.40	317	21.72	307	49.70	633	57.43	91	5.751	157	2.943
2	618	42.91	318	21.04	308	46.62	635	52.41	156	7.937	211	3.651
3	618	43.09	318	20.97	308	46.63	635	52.39	65	3.401	101	2.4667
4	630	41.61	318	16.19	308	40.14	635	44.36	156	6.995	234	2.687
$C_1=0,\ C_2=0,\ D_1=0,\ D_2=0,\ E=0,\ C_x=1,\ C_y=1\ h^{-1}=61$												
1	1514	441.2	1140	293.1	1061	644.3	2188	751.1	298	72.60	875	43.43
2	2173	588.3	1141	281.9	1961	599.4	2190	681.2	533	105.1	822	33.47
3	2173	589.7	1141	281.6	1061	599.9	2190	680.5	229	46.09	779	31.64
4	2194	564.1	1141	220.2	1061	530.6	2190	585.9	533	92.18	822	25.98
$C_1=1,\ C_2=1,\ D_1=0,\ D_2=0,\ E=0,\ C_x=1,\ C_y=1\ h^{-1}=31$												
1	500	37.43	357	24.60	345	55.85	710	64.37	101	6.386	176	3.281
2	692	48.04	358	23.70	346	52.45	712	58.81	151	7.703	260	3.904
3	692	48.24	358	23.61	346	52.38	712	58.73	55	2.890	103	2.512
4	706	46.61	358	18.26	346	45.09	712	49.84	150	6.694	260	2.982
$C_1=1,\ C_2=1,\ D_1=0,\ D_2=0,\ E=0,\ C_x=1,\ C_y=1\ h^{-1}=61$												
1	1720	501.6	1278	328.7	1189	722.4	2450	841.0	333	81.29	626	31.15
2	2430	658.2	1279	316.3	1190	672.0	2452	763.3	536	106.1	920	37.41
3	2430	659.9	1279	315.9	1190	672.4	2452	762.1	195	39.46	872	35.39
4	2454	631.1	1279	246.9	1190	594.8	2452	655.7	536	92.56	920	29.05

有所提高，但 CPU 几乎没有降低，当 $(h+z)^{l+1} \ll hz$ 时，l 每增加 1，CPU 时间反而增加了，七、九对角矩阵也有同样的性质，这与我们上节的理论分析是一致的。

从表 2，3 的数值结果可知，PSIP 算法确实比别的并行算法好，更比串行算法好。所以 PSIP 算法的确是一个比较好的解大稀疏线性代数方程组 $Ax=b$ 的并行算法。

胡家赣教授阅读了本文的初稿，并提出了许多很好的修改意见。在此谨向他表示衷心的感谢！审稿同志也提出了许多很有价值的修改意见，在此也向他们表示衷心的感谢！

参 考 文 献

[1] D. J. Evans, Group Explicit Iterative Methods for Solving Large Linear System, Intern. J. Computer Math. Vol. 17, (1985), pp. 87—108.

[2] D. J. Evans, New Paralled Algorithms for Partial Differential Equation, Int. Conf. "Paralle. Computing 83".

[3] Zhang Bao-lin, Alternating Segment Explicit Implicit Method for the Diffusion Equation, IAPCM. "On lecture 1989".

[4] Hu Jia-gan, The Estimates of $\|M^{-1}N\|_\infty$ and The Optimally Scaled Matrix, JCM, Vol. 2, No. 2(1984), pp. 122—129.

[5] Hu jia-gan, seven-diagonal and nime-diagonal SIP Algorithms, JCM, Vol.5, No. 2.

[6] N.M.Missirlis, Convergence of a Parallel Jacobi-Type Method, Intern. J.Computer Math., Vol. 14(1983) p. 371—384.

[7] D.M. Young, Iterative Solution of Large Linear System, Academic Press New York, (1971).
[8] R.S. Varga, Matrix Iterative Analysis. prentice-Hall 1962.
[9] James M. Ortega, Parallel and Vector Solution of Linear Systems, University of Virginia report (1986).
[10] 胡家赣,尺度变换和矩阵分解的收敛性,计算数学 5: 2(1983).
[11] 胡家赣,五,七,九对角线的 SIP 格式及其收敛性,科技学报, 1987 年第 1 期.
[12] 胡家赣, $\|B^{-1}A\|$ 的估计及其应用,计算数学, 4: 3(1982) p. 272—282.
[13] 胡家赣,尺度矩阵和矩阵分解的收敛性. 计算数学, 5: 1(1983)p. 71—78.
[14] 刘兴平, SIP 格式与参数的选择,计算物理, Vol.6, No. 1 (1989)P94—103.
[15] 刘兴平,七、九对角阵 SIP 格式与参数的选择,1986 年全国第四届数值代数会议文献.
[16] Liu Xing-ping, Accelerated SIP Algorithms and Convergence, Chinese J. Num. Math. & Appl. Vol. 11 No. 1(1989) p70—81.