Parallel Strategy for Solving Block-Tridiagonal Linear Systems

Zhijian Duan [a], Xinrong Ma [b] and Mengmei Qin [c]

School of Mathematics and Information Science, Xianyang Normal University, Xianyang 712000, China

[a]zhijian_duan@126.com, [b]mxryx1021@163.com, [c]979639772@qq.com

Abstract

Efficient parallel iterative algorithm is investigated for solving block-tridiagonal linear systems on distributed-memory multi-computers. Based on Galerkin theory, the communication only need twice between the adjacent processors per iteration step. Furthermore, the condition for convergence was given when the coefficient matrix A is a symmetric positive definite matrix. Numerical experiments implemented on the cluster verify that our algorithm parallel acceleration rates and efficiency are higher than the multi-splitting one, and has the advantages over the multisplitting one of high efficiency and low memory space.

Keywords

Block-tridiagonal linear systems; Galerkin theory; Parallel strategy; Multisplitting algorithm

1. Introduction

In recent years, the high-performance parallel computing technology has been rapidly developed. The large banded linear systems are frequently encounted when finite difference or finite element methods are used to discretize partial differential equations in many practice scientific problems and engineering computing. These systems can be efficiently resolved on sequential computers but are difficult to solve on parallel computers, where the communications take a significant part of the total execution time. So we need more efforts study to investigate more efficient parallel algorithm to improve the experimental results.

The parallel algorithms on the problem have been widely investigated in Refs. [1-8]. Especially, the multisplitting algorithm in Ref. [1] is the most popular at present. In Ref. [3], the authors provide a method for solving block-tridiagonal linear systems in which local lower and upper triangular incomplete factors are combined into an effective approximation for global incomplete lower and upper triangular factors of coefficient matrix based on two-dimensional domain decomposition with small overlapping. The algorithm is applicable to any preconditioner of incomplete type. In Ref. [4], a parallel strategy based on the Galerkin principle for solving block-tridiagonal linear systems is presented. In Ref. [5], a parallel direct algorithm based on Divide-and-Conquer principle and the decomposition of the coefficient matrix is investigated for solving the block-tridiagonal linear systems on distributed-memory multi-computers. The communications of the algorithm is only twice between the adjacent processors. Ma et al developed an alternating direction parallel algorithm for banded linear systems in Ref. [6]. In Ref. [7], a direct method for solving circular-tridiagonal block linear systems is presented. Some parallel algorithms for solving the linear systems can be found in Refs. [9-17]. In Ref. [15], for solving two-dimensional Poisson equation, a new quadratic PEk method is proposed. Based on GPU, Liu developed iterative algorithm for complex linear equations of symmetric positive definite sparse matrices. The algorithm in this paper is discussed on the basis of the advantages of the one in Ref. [2].

The goal of this paper is to develop an efficient parallel iterative method on distributed-memory multi-computer, and to give some theoretical analysis. The organization of this paper is as follows. In section 2, the parallel iterative algorithm is described and the parallel iterative process is discussed. The analysis of convergence is done in Section 3. The numerical results are shown in Section 4. Finally, the results analysis is given in Section 5. Section 6 summarizes the conclusion of this paper.

2. Organization of the Text

2.1 Parallel algorithm

Parallel calculation scheme

Let a block-tridiagonal linear equations $\boldsymbol{Ax}=\boldsymbol{b}$ can be represented as

$$\begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_1 & & & \\ \boldsymbol{C}_2 & \boldsymbol{A}_2 & \boldsymbol{B}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{C}_{2n-1} & \boldsymbol{A}_{2n-1} & \boldsymbol{B}_{2n-1} \\ & & & \boldsymbol{C}_{2n} & \boldsymbol{A}_{2n} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{2n-1} \\ \boldsymbol{x}_{2n} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_{2n-1} \\ \boldsymbol{b}_{2n} \end{pmatrix}, \tag{1}$$

where $\boldsymbol{A}_i, \boldsymbol{B}_i, and\ \boldsymbol{C}_i$ are all $t\times t$ real square matrices, $\boldsymbol{x}_i\ and\ \boldsymbol{b}_i$ are t -dimensional real column vectors with $i=1,2,\cdots,2n$. In general, assuming that there are p processors available and $n=pm\ (m\geq 2, m\in Z)$, we denote the ith processor by $P_i\,(\text{for } i=1,2,\cdots,p)$.

Based on Galerkin theory, we choose the following basis functions

$$\boldsymbol{V}_m=\boldsymbol{W}_m=\begin{cases}\left(\boldsymbol{q}_1,\cdots,\boldsymbol{q}_{t-1},\boldsymbol{q}_{t+1},\cdots,\boldsymbol{q}_{(p-1)t-1},\boldsymbol{q}_{(p-1)t+1},\cdots,\boldsymbol{q}_{pt-1}\right)\\ \left(\boldsymbol{q}_t,\boldsymbol{q}_{2t},\cdots,\boldsymbol{q}_{pt}\right)\end{cases},$$

then the solution of the equation $\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\boldsymbol{V}_m\boldsymbol{y}^{(k)}$, here $\boldsymbol{y}^{(k)}$ satisfies

$$\left(\boldsymbol{V}_m^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_m\right)\boldsymbol{y}^{(k)}=\boldsymbol{V}_m^{\mathrm{T}}\boldsymbol{r}^{(k)}, \tag{2}$$

where $\boldsymbol{r}_i=\left(r^{(k)}_{i\atop \sum_{j=1} n_{j-1}+1},\cdots,r^{(k)}_{i \atop \sum_{j=1} n_j}\right)^{\mathrm{T}}\ (i=1,2,\cdots,pt)$.

When k is an odd, $\boldsymbol{V}_m^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_m$ is a block-diagonal matrix, then (2) can be written as

$$\begin{pmatrix}\boldsymbol{D}_1 & & & \\ & \boldsymbol{D}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{D}_p\end{pmatrix}\begin{pmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\\ \vdots\\ \boldsymbol{y}_p\end{pmatrix}=\begin{pmatrix}\overline{\boldsymbol{r}}_1^{\mathrm{T}}\overline{\boldsymbol{r}}_1\\ \overline{\boldsymbol{r}}_2^{\mathrm{T}}\overline{\boldsymbol{r}}_2\\ \vdots\\ \overline{\boldsymbol{r}}_p^{\mathrm{T}}\overline{\boldsymbol{r}}_p\end{pmatrix}$$

here

$$\boldsymbol{D}_i=\begin{pmatrix}\boldsymbol{r}^{\mathrm{T}}_{(i-1)t+1}\boldsymbol{A}_{(i-1)t+1}\boldsymbol{r}_{(i-1)t+1} & \boldsymbol{r}^{\mathrm{T}}_{(i-1)t+1}\boldsymbol{B}_{(i-1)t+1}\boldsymbol{r}_{(i-1)t+2} & & \\ \boldsymbol{r}^{\mathrm{T}}_{(i-1)t+2}\boldsymbol{C}_{(i-1)t+2}\boldsymbol{r}_{(i-1)t+1} & \boldsymbol{r}^{\mathrm{T}}_{(i-1)t+2}\boldsymbol{A}_{(i-1)t+2}\boldsymbol{r}_{(i-1)t+2} & \boldsymbol{r}^{\mathrm{T}}_{(i-1)t+2}\boldsymbol{B}_{(i-1)t+2}\boldsymbol{r}_{(i-1)t+3} & \\ \ddots & \ddots & \ddots & \\ & & \boldsymbol{r}^{\mathrm{T}}_{it-1}\boldsymbol{C}_{it-1}\boldsymbol{r}_{it-2} & \boldsymbol{r}^{\mathrm{T}}_{it-1}\boldsymbol{A}_{it-1}\boldsymbol{r}_{it-1}\end{pmatrix},\ \boldsymbol{y}_i=\begin{pmatrix}y_{(i-1)t+1}\\ y_{(i-1)t+2}\\ \vdots\\ y_{it-1}\end{pmatrix},\ \overline{\boldsymbol{r}}_i=\begin{pmatrix}\boldsymbol{r}_{(i-1)t+1}\\ \boldsymbol{r}_{(i-1)t+2}\\ \vdots\\ \boldsymbol{r}_{it-1}\end{pmatrix}(i=1,2,\cdots,p).$$

then we have

$$\begin{cases}\boldsymbol{x}^{(k+1)}_{(i-1)t+j}=\boldsymbol{x}^{(k)}_{(i-1)t+j}+\boldsymbol{y}_{(i-1)t+j}\boldsymbol{r}_{(i-1)t+j}\\ \boldsymbol{x}^{(k+1)}_{it}=\boldsymbol{x}^{(k)}_{it}\end{cases},\ (i=1,2,\cdots,p;\ j=1,2,\cdots,t-1),\ \boldsymbol{r}^{(k+1)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k+1)}.$$

When k is an odd, $\boldsymbol{V}_m^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_m$ is also a block-diagonal matrix, then we get

$$\begin{cases}\boldsymbol{x}^{(k+1)}_{it}=\boldsymbol{x}^{(k)}_{it}+\boldsymbol{y}_{it}\boldsymbol{r}_{it}\\ \boldsymbol{x}^{(k+1)}_{(i-1)t+j}=\boldsymbol{x}^{(k)}_{(i-1)t+j}\end{cases},\ (i=1,2,\cdots,p;\ j=1,2,\cdots,t-1),\ \boldsymbol{r}^{(k+1)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k+1)}.$$

The above is the detailed parallel calculation procedure.

Cycle process

(1) P_i computes

$$\begin{pmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_p \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix}=\begin{pmatrix} \bar{r}_1^{\mathrm{T}}\bar{r}_1 \\ \bar{r}_2^{\mathrm{T}}\bar{r}_2 \\ \vdots \\ \bar{r}_p^{\mathrm{T}}\bar{r}_p \end{pmatrix},\begin{cases} x_{(i-1)t+j}^{(k+1)}=x_{(i-1)t+j}^{(k)}+y_{(i-1)t+j}r_{(i-1)t+j} \\ x_{it}^{(k+1)}=x_{it}^{(k)} \end{cases},\ (i=1,2,\cdots,p;\ j=1,2,\cdots,t-1),$$

and $r^{(k+1)}=b-Ax^{(k+1)}$, then performs a parallel communication to send $x_{(i-1)t+1}^{(k+1)}$.

(2) P_i computes

$$\begin{pmatrix} r_t^{\mathrm{T}}A_t r_t & & & \\ & r_{2t}^{\mathrm{T}}A_{2t}r_{2t} & & \\ & & \ddots & \\ & & & r_{pt}^{\mathrm{T}}A_{pt}r_{pt} \end{pmatrix}\begin{pmatrix} y_t \\ y_{2t} \\ \vdots \\ y_{pt} \end{pmatrix}=\begin{pmatrix} r_t^{\mathrm{T}}r_t \\ r_{2t}^{\mathrm{T}}r_{2t} \\ \vdots \\ r_{pt}^{\mathrm{T}}r_{pt} \end{pmatrix},\begin{cases} x_{it}^{(k+1)}=x_{it}^{(k)}+y_{it}r_{it} \\ x_{(i-1)t+j}^{(k+1)}=x_{(i-1)t+j}^{(k)} \end{cases},\ (i=1,2,\cdots,p;\ j=1,2,\cdots,t-1),$$

and $r^{(k+1)}=b-Ax^{(k+1)}$, then performs a parallel communication to send $x_{it}^{(k+1)}$.

(3) On the P_i processor, judge whether the inequality $\left\|x_i^{(k+1)}-x_i^{(k)}\right\|<\varepsilon$ (ε is error bound, $i=1,2,\cdots,p$)holds. Stop if these inequalities hold on every processor, or return to (1) and continue cycling until all inequalities are satisfied.

2.2 Analysis of convergence

Theorem 1. *Let $A\in R^{n\times n}$ is a symmetric positive definite matrix. Then the following expression is established. That Is,* $\left\|e^{(k+1)}\right\|_A\le\left(1-\frac{1}{\left\|A^{-1}\right\|_2\left\|A\right\|_2}\right)^{(k+1)/2}\left\|e^{(0)}\right\|_A$.

Proof . Since $e^{(k+1)}=e^{(k)}-V_m\left(V_m^{\mathrm{T}}AV_m\right)^{+}V_m^{\mathrm{T}}Ae^{(k)}$, by Lemma 3.1[18], we have

$$V_m\left(V_m^{\mathrm{T}}AV_m\right)^{+}V_m^{\mathrm{T}}A=V_m\left(G^{\mathrm{T}}F^{\mathrm{T}}AFG\right)^{+}V_m^{\mathrm{T}}A=V_m\left(G\right)^{+}\left(F^{\mathrm{T}}AF\right)^{+}\left(G^{\mathrm{T}}\right)^{+}V_m^{\mathrm{T}}A=F\left(F^{\mathrm{T}}AF\right)^{-1}F^{\mathrm{T}}A$$

and

$$e^{(k+1)^{\mathrm{T}}}Ae^{(k+1)}=e^{(k)^{\mathrm{T}}}Ae^{(k)}-e^{(k)^{\mathrm{T}}}AF\left(F^{\mathrm{T}}AF\right)^{-1}F^{\mathrm{T}}Ae^{(k)}$$

And since $F^TF=I_r$, then there exist the matrix $S\in\mathbf{R}^{n\times(n-r)}$ such that $(F\vdots S)$ is a orthogonal matrix, that is, $(F\vdots S)=Q$, then we obtain $\left\|F^{\mathrm{T}}Ae\right\|_2\le\left\|Q^{\mathrm{T}}Ae\right\|_2=\left\|Ae\right\|_2$. By Lemma 3.2[18], we get

$$\left\|e^{(k+1)}\right\|_A^2\le\left\|e^{(k)}\right\|_A^2-\left\|e^{(k)^{\mathrm{T}}}AF\right\|_2^2\lambda_{\min}\left(\left(F^{\mathrm{T}}AF\right)^{-1}\right)\le\left\|e^{(k)}\right\|_A^2-\frac{\left\|e^{(k)^{\mathrm{T}}}AF\right\|_2^2}{\left\|A\right\|_2}\le\left\|e^{(k)}\right\|_A^2-\frac{\left\|r^{(k)}\right\|_2^2}{\left\|A\right\|_2}=\left\|e^{(k)}\right\|_A^2-\frac{\left(r^{(k)},r^{(k)}\right)}{\left(r^{(k)},A^{-1}r^{(k)}\right)}\frac{\left\|e^{(k)}\right\|_2^2}{\left\|A\right\|_2}.$$

By Lemma 3.3[18], we have

$$\left\|e^{(k+1)}\right\|_A^2\le\left\|e^{(k)}\right\|_A^2-\frac{\left\|e^{(k)}\right\|_A^2}{\left\|A^{-1}\right\|_2\left\|A\right\|_2}=\left(1-\frac{1}{\left\|A^{-1}\right\|_2\left\|A\right\|_2}\right)\left\|e^{(k)}\right\|_A^2\le\left(1-\frac{1}{\left\|A^{-1}\right\|_2\left\|A\right\|_2}\right)^{k+1}\left\|e^{(0)}\right\|_A^2,$$

then $\left\|e^{(k+1)}\right\|_A\le\left(1-\frac{1}{\left\|A^{-1}\right\|_2\left\|A\right\|_2}\right)^{(k+1)/2}\left\|e^{(0)}\right\|_A$.

2.3 Numerical examples

Example 1. Consider a block-tridiagonal linear system $Ax=b$, here

$$A=\begin{pmatrix} A_1 & B_1 & & & \\ C_2 & A_2 & B_2 & & \\ & \ddots & \ddots & \ddots & \\ & & C_{m-1} & A_{m-1} & B_{m-1} \\ & & & C_m & A_m \end{pmatrix}$$

$A_i = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}_{t\times t}$, $B_i = C_i = -I, C_1 = B_m = 0$, $b_i = (1,1,\cdots,1)^T_{t\times 1}$, $t = 50, m = 1000$, $x_i^{(0)} = (0,0,\cdots,0)^T_{t\times 1}$. The stopping criterion $\varepsilon = 1\times 10^{-10}$. The numerical results are shown as Tables 1, 2 and 3.

Example 2. Consider an elliptic partial differential equation

$$C_x \frac{\partial^2 u}{\partial x^2} + C_y \frac{\partial^2 u}{\partial y^2} + (C_1 \sin 2\pi x + C_2)\frac{\partial u}{\partial x} + (D_1 \sin 2\pi x + D_2)\frac{\partial u}{\partial x} + Eu = 0, 0 \le x, y \le 1$$

equipped with the boundary conditions $u|_{x=0} = u|_{x=1} = 10 + \cos \pi y$, $u|_{y=0} = u|_{y=1} = 10 + \cos \pi x$, here $C_x, C_y, C_1, C_2, D_1, D_2$ and E are all constants. We denote $C_x = C_y = E = 1, C_1 = C_2 = D_1 = D_2 = 0$, respectively. Using the finite difference method, we obtain two block-tridiagonal linear systems on condition that the step sizes $h = 1/100$. Then, we apply this algorithm with the optimal relaxation factor to the systems on the HP rx2600 cluster. The numerical results are shown in Table 4, Table 5 and Table 6, here P is the number of processor, T is the run times(seconds), the S is speedup(T of one processor/T of all processors), L is iteration times and E is the efficiency($E = S/P$).

Table 1 The results for model 1(the algorithm given in the paper)

P	1	2	4	8
T	30.4938	15.6170	7.9818	4.2221
S		1.9526	3.8204	7.2224
E		0.9763	0.9551	0.9028
L	162	206	178	209
Δ	1.3159e-11	1.3159e-11	2.5228e-11	1.3159e-11

Table 2 The results for model 1(the multisplitting method)

P	1	2	4	8
T	22.7881	15.4051	7.7742	6.3529
S		1.3149	2.6056	3.1886
E		0.6575	0.6514	0.5314
L	177	480	257	493

Table 3 The results for model 1(the row operation method)

P	1	2	4	8
T	34.4938	26.1079	25.7147	22.8668
S		1.3212	1.3414	1.5085
E		0.6606	0.3354	0.2515
L	3248	3248	3248	3292

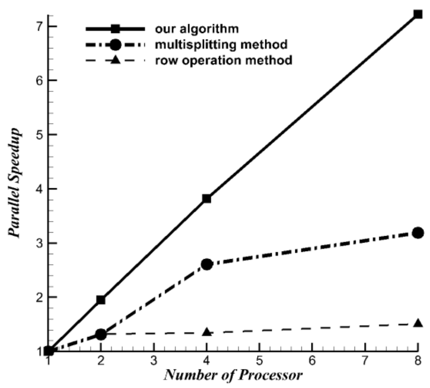

Fig. 1 The parallel speedup of Example 1

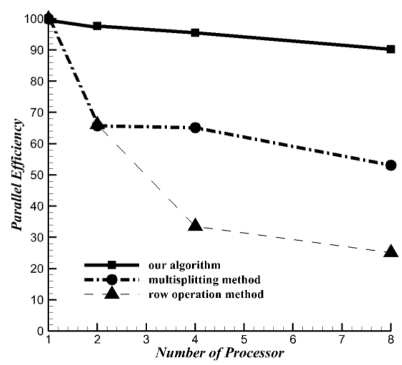

Fig. 2 The parallel efficiency of Example 1

Table 4 The results for model 2(the algorithm given in the paper)

P	1	2	4	8
T	963.9996	492.8244	251.7298	132.3971
S		1.9746	3.8658	7.3501
E		0.9873	0.9664	0.9188
L	4114	4124	4126	4126
Δ	9.8879e-11	9.9845e-11	9.8893e-11	9.8908e-11

Table 5 The results for model 2(the multisplitting method)

P	1	2	4	8
T	134.2484	69.4497	39.6882	25.3379
S		1.9330	3.3826	5.2983
E		0.9665	0.8456	0.6623
L	1053	1067	1067	1067
Δ	1.0002e-10	1.4842e-10	1.4842e-10	1.4842e-10

Table 6 The results for model 2(the row operation method)

P	1	2	4	6
T	39.2353	28.3779	23.9006	21.6830
S		1.3826	1.6416	1.8095
E		0.6913	0.4104	0.3016
L	3344	3344	3344	3389
Δ	1.1763e-09	1.1763e-09	1.2059e-09	1.1763e-09

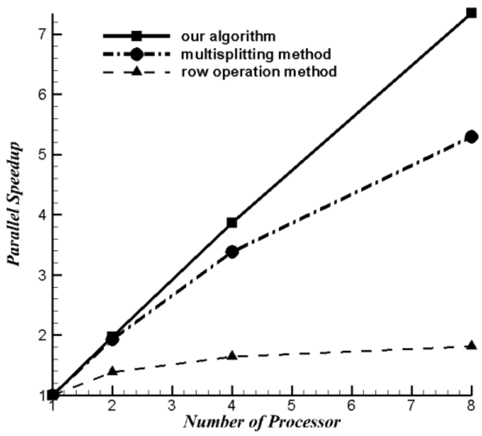

Fig. 3 The parallel speedup of Example 2

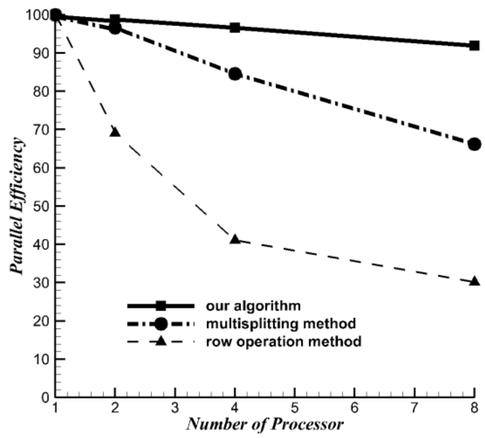

Fig. 4 The parallel efficiency of Example 2

From Table 1 to Table 6, we can get the following conclusion.

• By the numerical results, it can be known that the parallel one has good parallelism.

• As to the Examples 1 and 2, the results of the examples show that the efficiency of the algorithm is better than the multisplitting ones and the row action ones. Our algorithm has good parallel speedup same as BSOR methods to the Examples.

• The parallel algorithm is easily implemented on parallel computer and more flexible and simple than [1] in practice.

• The requirements on communication and memory space are low.

Obviously, our algorithm has better parallelism compared with the multisplitting method and the row operation method from Fig. 1 to Fig. 4.

3. Conclusion

An efficient parallel iterative method on a distributed-memory multi-computer has been presented for solving the large banded linear systems. Only twice requires the communications of the algorithm between the adjacent processors. Theoretical analysis and experiment show that the algorithm in this paper has good parallelism and high efficiency. When the coefficient matrix is a symmetric positive definite matrix, we know that the parallel algorithm is convergent. Our algorithm has an advantage over the multisplitting one of high efficiency. In summary, our method is more suitable for solving large-scale banded linear equations in MIMD distributed storage environment.

Acknowledgements

Thanks: This research was supported by National Natural Science Foundation of China under grant 61401383, Xianyang Normal University Foundation under grant 15XSYK032 and National College Students Innovation Training Program of Xianyang Normal University grant 201410722021, 1738 and 2015050.

References

[1] Bo Zhang, Tongxiang Gu, Zeyao Mo: *Principles and Methods of Numerical Parallel Computation*(National Defence Industry Press, China 1999), p. 166-168.

[2] Quanyi Lü, Tianqi Ye: An improve parallel algorithm for solving linear equations involving block tridiagonal coefficient matrix, J. Northwestern Polytech. Univ., Vol. 4 (1996) No.2, p. 314-317.

[3] Jianping Wu, Junqiang Song, Weimin Zhang et al: Parallel incomplete factorization preconditioning of block-tridiagonal linear systems with 2-D domain decomposition, Computational Physics, Vol. 26(2009)No. 2, p. 191-199.

[4] Zhijian Duan, Yong Yang, Quanyi Lv, and Xinrong Ma: Parallel strategy for solving block-tridiagonal linear systems, Computer Engineering and Applications, Vol. 47 (2011)No. 13, p. 46-49.

[5] Yanhong Fan: *The parallel algorithms for solving the large scale linear systems with typical structure*(MS. Northwestern Polytechnical University, China 2009), p. 9-19.

[6] Xinrong Ma, Sanyang Liu, Manyu Xiao et al: Parallel algorithm with parameters based on alternating direction for solving banded linear systems, Mathematical Problems in Engineering, Vol. 2014(2014), Article ID 752651, 8 pages.

[7] Salah M. EI-Sayed: A direct method for solving circulant tridiagonal block systems of linear equations, Applied Mathematics and Computation, Vol. 165(2005)No. 1, p. 23–30.

[8] Xining Cui, Quanyi Lü: A parallel algorithm for block-tridiagonal linear systems, Applied Mathematics and Computation, Vol. 173(2006)No. 2, p. 1107-1104.

[9] R. S. Varga: *Matrix Iterative Analysis*(Springer Science & Business Media, America 1962).

[10] Jiagan Hu: *Iterative method of linear algebraic equations*(Science Press, China 1999), p. 20–21.

[11] Andreas Frommer, Daniel B. Szyld: Weighted max norms, splitting, and overlapping additive Schwarz iterations, Numerische Mathematik, Vol. 83(1999)No. 2 , p. 259–278.

[12] Jianhu Feng, Gangming Che, Yufeng Nie: *Principle of numerical analysis*(Science Press, China 2002), p. 152–153.

[13] Petter Bj rstad, Mitchell Luskin: *Parallel Solution of Partial Differential Equations* (Springer-Verlag, Germany 2006), p. 31-55.

[14] W. H. Reed, T. R. Hill: *Triangle mesh methods for the Neutron transport equation*(Los Alamos Scientific Laboratory, America, Report LA-UR-73-479, 1973).

[15] Huali Liu: A new quadratic PE_k method for solving linear algebraic equations derived from two-dimensional Poisson equation, Chinese Journal of Guangxi Technology University, Vol. 27 (2016)No. 2, p. 100-103.

[16] Weiwei Li: Iterative algorithm for complex linear equations of symmetric positive sparse matrices based on GPU, Chinese Journal of Jilin University, Vol. 54 (2016)No. 2, p. 297-302.

[17] Zhigang Luo, Xiaomei Li: A distributed parallel algorithm of block tridiagonal linear equations, Chinese J.Comput., Vol. 23 (2000)No. 10, p. 1028-1034.

[18] Y. P. Cheng: *Matrix Theory*(Northwestern polytechnical University Press, China 2002), p. 389-396.