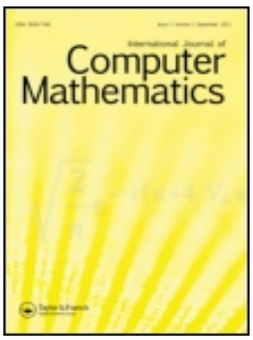

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: http://www.tandfonline.com/loi/gcom20

# Multiple search direction conjugate gradient method II: theory and numerical experiments

Tongxiang Gu , Xingping Liu , Zeyao Mo & Xuebin Chi

**To cite this article:** Tongxiang Gu , Xingping Liu , Zeyao Mo & Xuebin Chi (2004) Multiple search direction conjugate gradient method II: theory and numerical experiments, International Journal of Computer Mathematics, 81:10, 1289-1307, DOI: 10.1080/00207160412331289065

**To link to this article:** http://dx.doi.org/10.1080/00207160412331289065

*International Journal of Computer Mathematics*
Vol. 81, No. 10, October 2004, pp. 1289–1307

Taylor & Francis
Taylor & Francis Group

# MULTIPLE SEARCH DIRECTION CONJUGATE GRADIENT METHOD II: THEORY AND NUMERICAL EXPERIMENTS

TONGXIANG GU[a,b,*], XINGPING LIU[a], ZEYAO MO[a] and XUEBIN CHI[b]

[a]*Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, P.O. Box 8009, Beijing 100088, P.R. China;*
[b]*Supercomputing Center of Computer Network Information Center, Chinese Academy of Science, P.O. Box 349, Beijing 100080, P.R. China*

*(Revised 24 September 2003; In final form 10 March 2004)*

In this article, we give the convergence and consistency of the MSD-CG method [Gu, T.-X., Liu, X.-P., Mo, Z.-Y. and Chi, X.-B. (in press). Multiple search direction conjugate gradient method I: Methods and their propositions. *Int. J. Comput. Math.*] and estimate the convergence rate of this method. Numerical experiments on two distributed parallel computers, Dawning 3000 and P-II Cluster, show the efficiency of our method and show that it compares favorably with some domain decomposition methods.

*Keywords*: Linear systems; Conjugate gradient-type method; Massively parallel computing; Inner product; Global communication

*AMS Subject Classifications*: 65F10; 65Y99

*C.R. Categories*: G1.3; G1.8

## 1 SOME ESTIMATES

We proposed the MSD-CG and the GIPF-CG method and their preconditioned versions in Ref. [10]. In this section, we give some estimates for the proof of convergence of these methods. The notations and denotations we take here are the same as those in Ref. [10].

Let $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ be the minimum and maximum eigenvalue of $A$, respectively, $\underline{v}$ and $\bar{v}$ be their associate eigenvectors.

LEMMA 1.1 [22] *If A is SPD, then*

$$\lambda_{\max}(A) = \max_{v\neq 0} \frac{(v, Av)}{(v, v)} = \frac{(\bar{v}, A\bar{v})}{(\bar{v}, \bar{v})}$$
$$\lambda_{\min}(A) = \min_{v\neq 0} \frac{(v, Av)}{(v, v)} = \frac{(\underline{v}, A\underline{v})}{(\underline{v}, \underline{v})} \tag{1}$$

* Corresponding author. E-mail: txgu@iapcm.ac.cn

ISSN 0020-7160 print; ISSN 1029-0265 online 
DOI:10.1080/00207160412331289065

In this article, we assume the following basic assumption holds.

*Basic assumption* $A \in \mathbb{R}^{n\times n}$ is SPD, and $p_l^k \neq 0$ for each $k$ and all $l$ or matrix $P_k \in \mathbb{R}^{n\times L}$ is column full rank for each $k$.

We have from Eq. (15) of Ref. [10] that

$$e^{k+1} = (I - S_k)e^k = (I - P_k(P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A)e^k$$

LEMMA 1.2 *Under the basic assumption, there exists a invertible matrix* $\bar{P}_k \in \mathbb{R}^{L\times L}$ *such that* $\tilde{P}_k = P_k \bar{P}_k$ *satisfies* $\tilde{P}_k^{\mathrm{T}} \tilde{P}_k = I_L (L \times L$ *identity matrix).*

*Proof* Since $P_k = [p_1^k, p_2^k, \ldots, p_L^k] \in \mathbb{R}^{n\times L}$, we have from the structure of $p_l^k$ and the assumption that

$$P_k^{\mathrm{T}} P_k = \begin{pmatrix} \|p_1^k\| & 0 & \cdots & 0 \\ 0 & \|p_2^k\|_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \|p_L^k\|_2^2 \end{pmatrix} \in \mathbb{R}^{L\times L}$$

Denote $\bar{P}_k = \operatorname{diag}\left(\|p_1^k\|_2^{-1}, \|p_2^k\|_2^{-1}, \ldots, \|p_L^k\|_2^{-1}\right) \in \mathbb{R}^{L\times L}$, then

$$\tilde{P}_k^{\mathrm{T}} \tilde{P}_k = \bar{P}_k^{\mathrm{T}} P_k^{\mathrm{T}} P_k \bar{P}_k = I_L \qquad \blacksquare$$

From this lemma, we have

$$\begin{aligned} e^{k+1} &= (I - P_k(P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A)e^k \\ &= (I - \tilde{P}_k(\tilde{P}_k^{\mathrm{T}} A \tilde{P}_k)^{-1} \tilde{P}_k^{\mathrm{T}} A)e^k \end{aligned} \tag{2}$$

Therefore, so as not to lose generality, we let $P_k^{\mathrm{T}} P_k = I_L$.

THEOREM 1.1 *Under the basic assumption, then*

$$\|C_k^{-1}\| = \|(P_k^{\mathrm{T}} A P_k)^{-1}\| \leq \frac{1}{\lambda_{\min}(A)} \leq \|A^{-1}\| \tag{3}$$

*Proof* Let $R_k = P_k P_k^{\mathrm{T}}$ be the projection on to $\mathbb{P}_k = \mathrm{Range}(P_k)$, $B_k = R_k A R_k$. From Cauchy–Schwartz inequality, we have

$$(w, B_k w) \leq \|w\| \cdot \|B_k w\|, \quad \forall w \in \mathbb{R}^n$$

and

$$\|(P_k^{\mathrm{T}} A P_k)^{-1}\|^2 = \max_{v\in\mathbb{R}^L\setminus\{0\}} \frac{\left\|(P_k^{\mathrm{T}} A P_k)^{-1} v\right\|^2}{\|v\|^2} = \max_{v\in\mathbb{R}^L\setminus\{0\}} \frac{\|v\|^2}{\left\|(P_k^{\mathrm{T}} A P_k) v\right\|^2}$$

Now let $v = P_k^{\mathrm{T}} w$, $w \in \mathbb{P}_k$, and note that $\|w\| = \|v\|$; we obtain

$$\begin{aligned}\|(P_k^{\mathrm{T}} A P_k)^{-1}\|^2 &= \max_{v \in \mathbb{R}^L \setminus \{0\}} \frac{\|w\|^2}{\|R_k A R_k w\|^2} = \max_{v \in R^L \setminus \{0\}} \frac{\|w\|^2}{\|B_k w\|^2} \\ &\le \max_{v \in \mathbb{R}^L \setminus \{0\}} \|w\|^2 \frac{\|w\|^2}{\|(w, B_k w)\|^2} = \max_{v \in \mathbb{R}^L \setminus \{0\}} \frac{\|w\|^4}{|(\omega, B_k w)|^2} \\ &= \frac{1}{\lambda_{\min}^2(A)} \le \|A^{-1}\|^2\end{aligned}$$

This completes the proof. ■

Let $\mathbb{P}_k = Range(P_k)$; then from Propositions 3.1 and 3.2 of Ref. [10], we have

THEOREM 1.2 *Suppose the basic assumption holds and there exists a constant c independent of initial approximate value $x^0$ such that*

$$0 < \|A\| \cdot \|x\|^2 \le c(x, Ax), \quad \forall x \in \mathbb{P}_k \setminus 0 \tag{4}$$

*Then breakdown is impossible at any step in the MSD-CG method, and we have*

$$\|e^{k+1}\| \le (1 + c)\|e^k\| \tag{5}$$

$$\|x^{k+1}\| \le \|x^k\| + c\|e^k\| \tag{6}$$

*Proof* Since each column of $P_k$ is nonzero, it forms a base of $\mathbb{P}_k$ and $C_k = P_k^{\mathrm{T}} A P_k$ is SPD. Hence, $C_k$ is nonsingular, and breakdown is impossible at any step in the MSD-CG method.

So as not to lose generality, we assume $P_k^{\mathrm{T}} P_k = I_L$, then

$$\begin{aligned}\|x^{k+1}\| &= \left\|x^k + P_k (P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A e^k\right\| \\ &\le \|x^k\| + \left\|(P_k^{\mathrm{T}} A P_k)^{-1}\right\| \|P_k^{\mathrm{T}} A e^k\| \\ &\le \|x^k\| + \frac{1}{\min\limits_{x \in \mathbb{P}_k \setminus \{0\}, \|x\| = 1} |x^{\mathrm{T}} A x|} \|A\| \|e^k\| \\ &\le \|x^k\| + \sup_{x \in \mathbb{P}_k \setminus \{0\}, \|x\| = 1} \frac{\|x\|^2}{|x^{\mathrm{T}} A x|} \|A\| \|e^k\| \\ &\le \|x^k\| + c\|e^k\|\end{aligned}$$

For the same reason, we can obtain Eq. (4) since

$$e^{k+1} = e^k - P_k (P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A e^k$$

This completes the proof. ■

We can obtain the following corollary from the above proof.

COROLLARY 1.1 *Under the basic assumption, then*

$$\|e^{k+1}\| \leq \left(1 + \frac{\|A\|}{\lambda_{\min}(A)}\right)\|e^k\| \tag{7}$$

$$\|x^{k+1}\| \leq \|x^k\| + \frac{\|A\|}{\lambda_{\min}(A)}\|e^k\| \tag{8}$$

*Remark* Estimates (5) and (7) say that projection operator $I - S_k$ is uniformly bounded. But these cannot ensure the convergence of the method.

THEOREM 1.3 *Under the basic assumption, then*

$$\lambda_{\min}(C_k^{-1}) = \lambda_{\min}((P_k^{\mathrm{T}} A P_k)^{-1}) \geq \lambda_{\min}(A^{-1}) \tag{9}$$

*Proof* From the condition, we get that $C_k$ and hence, $C_k^{-1}$ is SPD and

$$\lambda_{\min}(C_k^{-1}) = \lambda_{\min}((P_k^{\mathrm{T}} A P_k)^{-1}) = \min_{v\neq 0} \frac{(v, (P_k^{\mathrm{T}} A P_k)^{-1} v)}{(v, v)} = \min_{v\neq 0} \frac{(v, (P_k^{\mathrm{T}} A P_k) v)}{\|(P_k^{\mathrm{T}} A P_k) v\|^2}$$

$$\geq \min_{v\neq 0} \frac{(v, Av)}{\|P_k^{\mathrm{T}} A v\|^2} \geq \min_{v\neq 0} \frac{(v, Av)}{\|Av\|^2} = \min_{v\neq 0} \frac{(v, A^{-1} v)}{\|v\|^2} \geq \lambda_{\min}(A^{-1}) \qquad \blacksquare$$

LEMMA 1.3 *If A is SPD, then*

$$\lambda_{\min}(A^{-1}) \geq \frac{1}{\|A\|} \tag{10}$$

*Proof* Since $A$ is SPD, we have

$$\lambda_{\min}(A^{-1}) = \min \frac{(v, A^{-1} v)}{(v, v)} = \min \frac{(v, Av)}{\|Av\|^2} \geq \frac{1}{\|A\|} \qquad \blacksquare$$

## 2 CONVERGENCE OF THE MSD-CG METHOD

In this section, we give the convergence theorems for the MSD-CG and the GIPF-CG methods. One can see in the following that two methods are convergent under more general conditions and the rate of convergence is at least as fast as that of the steepest descent method.

THEOREM 2.1 *Let $\{x^k\}$ be the iteration sequence generated by the MSD-CG method and let $\{e^k\}$ be the associated error sequence. Suppose the basic assumption holds; then the MSD-CG method is convergent and*

$$\|e^{k+1}\|_A \leq \left(1 - \frac{1}{\|A^{-1/2}\|\|A\|}\right)^{(k+1)/2} \|e^0\|_A \tag{11}$$

*Proof* Since

$$e^{k+1} = e^k - P_k (P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A e^k$$

Hence,

$$(e^{k+1})^{\mathrm{T}} A e^{k+1} = (e^k)^{\mathrm{T}} A e^k - (e^k)^{\mathrm{T}} A e^{k+1} P_k (P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A e^k$$

$A$ is SPD, so $A^{1/2}$ and $A^{-1/2}$ have definition. From Eqs. (9) and (10)

$$\begin{aligned}\|e^{k+1}\|_A^2 &\le \|e^k\|_A^2 - \|e^{k^{\mathrm{T}}} A P_k\|^2 \lambda_{\min}((P_k^{\mathrm{T}} A P_k)^{-1}) \\ &\le \|e^k\|_A^2 - \frac{\|e^{k^{\mathrm{T}}} A P_k\|^2}{\|A\|} \le \|e^k\|_A^2 - \frac{\|r^k\|^2}{\|A\|} = \|e^k\|_A^2 - \frac{(r^k, r^k)}{(r^k, A^{-1} r^k)} \frac{\|e^k\|_A^2}{\|A\|} \\ &= \|e^k\|_A^2 - \frac{(r^k, r^k)}{(A^{-1/2} r^k, A^{-1/2} r^k)} \frac{\|e^k\|_A^2}{\|A\|} \le \|e^k\|_A^2 - \frac{\|e^k\|_A^2}{\|A^{-1/2}\| \|A\|} \\ &= \left(1 - \frac{1}{\|A^{-1/2}\| \|A\|}\right) \|e^k\|_A^2\end{aligned}$$

So far, we have proved the convergence of the MSD-CG method and Eq. (11). ■

THEOREM 2.2 *Under the condition of Theorem* 2.1, *we have*

$$\|e^{k+1}\|_A \le \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^{k+1} \|e^0\|_A \tag{12}$$

*Proof* In order to prove the convergence of the MSD-CG method, we need only to prove that there is a positive number $\delta$ independent of the number of iteration $k$, such that

$$\|e^{k+1}\|_A \le (1 - \delta) \|e^k\|_A$$

holds for arbitrary $k$.

The choice of $\alpha_k$ in the MSD-CG method means that the energy norm of error is minimized. This is equivalent to minimizing the energy norm of error on the affine space $e^k + \mathbb{P}_k$. Let

$$\tilde{e}^{k+1} = e^k - \lambda A e^k$$

We choose $\tilde{\lambda}$ such that $\|\tilde{e}^{k+1}\|$ is minimized. This is equivalent to requiring that $A\tilde{e}^{k+1} \perp Ae^k$. Note that $Ae^k = r^k \in \mathbb{P}_k$, so this minimal problem takes up a smaller space and we have

$$\|e^{k+1}\|_A \le \|\tilde{e}^{k+1}\|_A \tag{13}$$

Since

$$0 = (A\tilde{e}^{k+1}, r^k) = (e^k - \tilde{\lambda} A e^k, Ar^k) = (e^k, AAe^k) - \tilde{\lambda}(Ae^k, AAe^k)$$

we get

$$\tilde{\lambda} = \frac{(e^k, Ae^k)_A}{(Ae^k, Ae^k)_A}$$

Hence,

$$\begin{aligned}\|\tilde{e}^{k+1}\|_A^2 &= \|e^k\|_A^2 - 2\tilde{\lambda}(e^k, Ae^k)_A + \tilde{\lambda}^2 (Ae^k, Ae^k)_A \\ &= \|e^k\|_A^2 - \frac{(e^k, Ae^k)_A^2}{(Ae^k, Ae^k)_A}\end{aligned}$$

By applying the Kantorovich inequality, we have

$$\begin{aligned}\frac{\|\tilde{e}^{k+1}\|_A^2}{\|e^k\|_A^2} &= 1 - \frac{(e^k, Ae^k)_A^2}{(e^k, e^k)_A(Ae^k, Ae^k)_A} = 1 - \frac{(r^k, r^k)}{(r^k, e^k)(r^k, Ar^k)} \\ &= 1 - \frac{4\lambda_{\max}(A)\lambda_{\min}(A)}{(\lambda_{\max}(A) + \lambda_{\min}(A))} = \left(\frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)}\right)^2 \\ &= \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^2\end{aligned}$$

Since Eq. (13) holds, we have

$$\begin{aligned}\|e^{k+1}\|_A \le \|\tilde{e}^{k+1}\|_A &\le \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)\|e^k\|_A \\ &\le \cdots \le \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^{k+1}\|e^0\|_A\end{aligned}$$

This proves the convergence of the MSD-CG method and Eq. (12). ■

*Remark* This theorem says that the convergent rate of the MSD-CG method is at least as fast as that of the steepest descent method.

THEOREM 2.3 *Let $\{x^k\}$ be the iteration sequence generated by the GIPF-CG method and $\{e^k\}$ be the associate error sequence. Suppose the basic assumption holds; then the GIPF-CG method is convergent.*

*Proof* The GIPF-CG method uses a convergent iterative method to solve small systems (5) and (6) in the MSD-CG method of Ref. [10]. Let $\alpha_*^k$ be the solution of Eq. (5) in Ref. [10] and $\alpha_j^k$ and $e_j^\alpha = \alpha_j^k - \alpha_*^k$ be its $k$th approximate solution and error, respectively. Since the iterative method for the small systems is convergent, there exists $j_0$ for arbitrary $\varepsilon > 0$ such that the relative error satisfies the following relation for $j \ge j_0$

$$\frac{\|e_j^\alpha\|_{C_k}}{\|\alpha_*^k\|_{C_k}} \le \sqrt{\varepsilon}$$

From the proof of Proposition 2.3 in Ref. [10], we have

$$\|e^{k+1}\|_A^2 = \|e^k\|_A^2 - (\alpha^k)^{\mathrm{T}} C_k \alpha^k = \|e^k\|_A^2 - \|\alpha^k\|_{C_k}^2 \tag{14}$$

Hence,

$$\|e^{k+1}\|_A^2 = \|e^k\|_A^2 - \|\alpha^k\|_{C_k}^2 \le \left(1 - \frac{\|e_j^\alpha\|_{C_k}^2}{\varepsilon\|e^k\|_A^2}\right)\|e^k\|_A^2$$

Denoting $\varepsilon_k = 1 - \|e_j^\alpha\|_{c_k}^2/\varepsilon\|e^k\|_A^2$, we can choose $\varepsilon$ such that $0 < \varepsilon_k < \delta < 1$. In order to do so, it only requires

$$\frac{\|e_j^\alpha\|_{C_k}^2}{\|e^k\|_A^2} = \frac{(C_k(\alpha_j^k - \alpha_*^k), (\alpha_j^k - \alpha_*^k))}{(A(x^k - x^*), (x^k - x^*))} < \varepsilon$$

Therefore,

$$\|e^{k+1}\|_A^2 \le \sqrt{\varepsilon_k}\|e^k\|_A \le \cdots \le \sqrt{\prod_{i=1}^{k}\varepsilon_i}\|e^0\|_A < \delta^{k/2}\|e^0\|_A$$

Since $0 < \delta < 1$, we have

$$\lim_{k\to\infty}\|e^{k+1}\|_A = 0$$

This completes the proof. ■

And we have the following theorem about the convergence of the MSD-CG method. Its proof is very simpler.

THEOREM 2.4 *Under the condition of Theorem* 2.1, *the MSD-CG method is convergent.*

*Proof* From Eq. (14), we have

$$\begin{aligned}\|e^{k+1}\|_A^2 &= \|e^k\|_A^2 - (\alpha^k)^{\mathrm{T}}C_k\alpha^k = \|e^k\|_A^2 - (\alpha^k)^{\mathrm{T}}P_k^{\mathrm{T}}AP_k\alpha^k \\ &= \|e^k\|_A^2 - \|P_k\alpha^k\|_A^2 = \|e^k\|_A^2 - \|e^{k+1} - e^k\|_A^2 \ge 0\end{aligned}$$

If $\|e^k\|_A^2 - \|e^{k+1} - e^k\|_A^2 = 0$, $x^{k+1}$ is the exact solution of Eq. (1) in Ref. [10] and the convergence is obtained. From the hypothesis and Proposition 3.2 in Ref. [10], we know

$$\|e^{k+1}\|_A^2 \le \|e^k\|_A^2$$

Thereby

$$0 < \frac{\|e^{k+1} - e^k\|_A^2}{\|e^k\|_A^2} < 1$$

Denote $\varepsilon_k = 1 - \|e^{k+1} - e^k\|_A^2/\|e^k\|_A^2$; then

$$\|e^{k+1}\|_A^2 = \sqrt{\varepsilon_k}\|e^k\|_A = \cdots = \sqrt{\prod_{i=0}^{k}\varepsilon_i}\|e^0\|_A$$

Due to $0 < \varepsilon_i < 1$ for $i = 1, \ldots, k, \ldots$, therefore,

$$\lim_{k\to\infty}\|e^{k+1}\|_A = 0$$

This completes the proof. ■

## 3 CONSISTENCY OF THE MSD-CG METHOD

From Eq. (14) of Ref. [10], we have

$$\begin{aligned}x^{k+1} &= x^k + P_k\alpha^k = (I - S_k)x^k + S_kx^* \\ &= \cdots = \prod_{j=k}^{0}(I - S_j)x^0 + \left[S_k + \sum_{i=1}^{k}\prod_{j=k}^{i}(I - S_j)S_i\right]x^* \\ &= E_kx^0 + C_k \qquad (15)\end{aligned}$$

where $E_k = (I - S_k)(I - S_{k-1}) \cdots (I - S_0)$, $c_k = [S_k + \sum_{i=1}^{k} \Pi_{j=k}^{i}(I - S_j)S_j]x^*$. From Eq. (15), one can see that the MSD-CG method is a nonstationary iterative method. We know from the basic concept of nonstationary iterative method that condition $\rho(I - S_k) < 1$ is neither a sufficient condition nor the necessary condition for the method to converge. In fact, we know (see for example [22]) that the MSD-CG method, *i.e.*, Eq. (15), is convergent if and only if $\{c_k\}$ is convergent and

$$\lim_{k\to\infty} E_k = 0 \tag{16}$$

Furthermore, we have

$$\begin{aligned} e^{k+1} &= e^k + P_k\alpha^k = (I - S_k)e^k \\ &= \cdots = \prod_{j=k}^{0}(I - S_j)e^0 = E_k e^0 \end{aligned} \tag{17}$$

and

$$\begin{aligned} r^{k+1} &= r^k - Q_k\alpha^k = r^k - AP_k(P_k^{\mathrm{T}}AP_k)^{-1}P_k^{\mathrm{T}}r^k \\ &= \left[I - AP_k(P_k^{\mathrm{T}}AP_k)^{-1}P_k^{\mathrm{T}}\right]r^k = (I - S_k^{\mathrm{T}})r^k \\ &= \cdots = \prod_{j=k}^{0}(I - S_j^{\mathrm{T}})r^0 \end{aligned} \tag{18}$$

We first give the following theorem for the consistence of the MSD-CG method.

THEOREM 3.1 *Under the condition of Theorem* 2.1, *we have*

$$\lim_{k\to\infty} \|E_k\|_A = 0 \tag{19}$$

*Proof* From Theorem 2.1 and Eq. (17), we have

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

Thereby,

$$\|E_k\|_A \le \left(1 - \frac{1}{\|A^{-1/2}\|\|A\|}\right)^{(k+1)/2}$$

and Eq. (19) holds. ■

THEOREM 3.2 *Under the condition of Theorem* 2.1, *the MSD-CG* (*nonstationary iterative*) *method is convergent.*

*Proof* What we need to prove is that $\{c_k\}$ is convergent because $\lim_{k\to\infty} \|E_k\|_A = 0$ has been proved in Theorem 3.1. In fact, if we denote $E_{k,i} = \Pi_{j=k}^{i}(I - S_j)$, then $E_k = E_{k,0}$, and

$$\begin{aligned} \prod_{j=k}^{i}(I - S_j)S_i &= \prod_{j=k}^{i}(I - S_j) - \prod_{j=k}^{i-1}(I - S_j) \\ &= E_{k,i} - E_{k,i-1} \end{aligned}$$

Consequently, from Eq. (15) we have

$$S_k + \sum_{i=1}^{k} \prod_{j=k}^{i} (I - S_j) S_i = S_k + E_{k,k} - E_{k,0} = S_k + (I - S_k) - E_k = I - E_k$$

*i.e.*,

$$c_k = (I - E_k)x^* \tag{20}$$

Then from Eq. (19), we have

$$\lim_{k \to \infty} c_k = 0$$

By now, we prove the convergence of the MSD-CG method by using the theory of the non-stationary iterative method, too. ■

THEOREM 3.3 *The MSD-CG method is entirely consistent with original system* (1) *of Ref.* [10].

*Proof* We prove the consistency, first. To this end, for each $j = 1, 2, \ldots$, we consider the linear stationary iterative method

$$x^{k+1} = (I - S_j)x^k + S_j x^*, \quad k \geq 0 \tag{21}$$

If for some $k$, $x^m$ is the solution of (1) in Ref. [10], *i.e.*, $x^m = x^*$, then

$$x^{m+1} = (I - S_j)x^* + S_j x^* = x^*$$

and for the same reason, $x^{m+2} = \cdots = x^*$. Hence, Eq. (21) is consist for each $j$, and non-stationary iterative Eq. (15) is consistent.

From Theorems 3.1 and 3.2, we have

$$\lim_{k \to \infty} x^{k+1} = \lim_{k \to \infty} E_k x^0 + \lim_{k \to \infty} c_k = x^*$$

This is the skew-consistency. Hence, the MSD-CG method, *i.e.*, the nonstationary iterative method (11) is completely consistent with original linear system (1) in Ref. [10]. ■

*Remark* Due to convergence and consistence Theorems 3.2 and 3.3, iteration sequence $\{x^k\}$ generated by the MSD-CG method is convergent and converges to the solution of system (1) in Ref. [10], *i.e.*,

$$x^* = A^{-1}b$$

The remainder of this section is dedicated to another proof of the convergence of the MSD-CG method, which is in favor of estimating the convergent rate. First, we outline a proposition about $E_k$.

PROPOSITION 3.1 *If denote* $E_{-1} = I$, *then*

$$I - E_k = \sum_{i=0}^{k} S_i E_{i-1} \tag{22}$$

*and for arbitary* $v \in \mathbb{R}^n$,

$$\|v\|_A^2 - \|E_k v\|_A^2 = \sum_{i=0}^{k} ((2I - S_i)E_{i-1}v, S_i E_{i-1}v)_A \tag{23}$$

*Proof* From the definition of $E_k$, we know $E_i = (I - S_i)E_{i-1}$ for $i = 1, 2, \ldots, k$. Therefore,

$$E_{i-1} - E_i = S_i E_{i-1}$$

One can obtain Eq. (22) by summing both sides from 1 to $k$ for $i$.

For any $v \in \mathbb{R}^n$

$$\begin{aligned}
\|E_{i-1}v\|_A^2 - \|E_i v\|_A^2 &= (E_{i-1}v, E_{i-1}v)_A - (E_i v, E_i v)_A \\
&= (E_{i-1}v, E_{i-1}v)_A - ((I - S_i)E_{i-1}v, E_{i-1}v)_A \\
&\quad + ((I - S_i)E_{i-1}v, S_i E_{i-1}v)_A \\
&= (S_i E_{i-1}v, E_{i-1}v)_A + ((I - S_i)E_{i-1}v, S_i E_{i-1}v)_A \\
&= ((2I - S_i)E_{i-1}v, S_i E_{i-1}v)_A
\end{aligned}$$

We can obtain Eq. (23) by summing both sides from 1 to $k$ for $i$. ■

Since

$$\begin{aligned}
S_k^{\mathrm{T}} A S_k &= A P_k (P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A P_k (P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A \\
&= A P_k (P_k^{\mathrm{T}} A P_k)^{-1} P_k^{\mathrm{T}} A = S_k^{\mathrm{T}} A
\end{aligned}$$

for arbitrary $v \in \mathbb{R}^n$, and any $k = 0, 1, \ldots$, we have

$$\begin{aligned}
(S_k v, S_k v)_A &= (v, S_k^{\mathrm{T}} A S_k v) = (v, S_k^{\mathrm{T}} A v) = (S_k v, v)_A \\
((2I - S_i)E_{i-1}v, S_i E_{i-1}v)_A &= (S_i E_{i-1}v, E_{i-1}v)_A
\end{aligned}$$

Submitting this into Eq. (23)

$$\|v\|_A^2 - \|E_k v\|_A^2 = \sum_{i=0}^{k} (S_i E_{i-1}v, E_{i-1}v)_A \tag{24}$$

Due to

$$\begin{aligned}
(S_i v, v)_A &= (S_i v, E_{i-1}v)_A + (S_i v, (I - E_{i-1})v)_A \\
&= (S_i v, E_{i-1}v)_A + \sum_{j=0}^{i-1} (S_i v, S_j E_{j-1}v)_A
\end{aligned} \tag{25}$$

By applying Cauchy–Schwarz inequality, we obtain

$$\sum_{i=0}^{k} (S_i v, E_{i-1}v)_A \le \left( \sum_{i=0}^{k} (S_i v, v)_A \right)^{1/2} \left( \sum_{i=0}^{k} (S_i E_{i-1}v, E_{i-1}v)_A \right)^{1/2} \tag{26}$$

In order to estimate $\sum_{i=0}^{k}\sum_{j=0}^{i-1}(S_i v, S_j E_{j-1} v)_A$ for $i > j$, we define $\varepsilon_{ij} \in (0, 1]$

$$\varepsilon_{ij}^2 = \rho(P_j^{\mathrm{T}} S_i P_j), \quad \varepsilon_{ji} = \varepsilon_{ij}, \quad \varepsilon_{ii} = 1 \tag{27}$$

Then for each $i \geq j$, $\varepsilon_{ij}$ is the minimum number which satisfies the following expression

$$(S_i v_j, v_j)_A \leq \varepsilon_{ij}^2 (v_j, v_j)_A, \quad \forall v_j \in \mathbb{P}_j$$

By defining $K := \max_{1\leq i\leq k} \sum_{i=0}^{k} \varepsilon_{ij}$, we can show for $i \geq j$ by applying Cauchy–Schwarz inequality that

$$\begin{aligned}(S_i v, S_j E_{j-1} v)_A &\leq (S_i v, v)_A^{1/2} (S_i S_j E_{j-1} v, S_j E_{j-1} v)_A^{1/2} \\ &\leq (S_i v, v)_A^{1/2} \varepsilon_{ij} (S_j E_{j-1} v, S_j E_{j-1} v)_A^{1/2} \\ &= \varepsilon_{ij} (S_i v, v)_A^{1/2} (S_j E_{j-1} v, E_{j-1} v)_A^{1/2}\end{aligned} \tag{28}$$

Summing both sides for $i > j$ and applying Cauchy–Schwarz inequality, we obtain

$$\sum_{i=0}^{k}\sum_{j=0}^{i-1}(S_i v, S_j E_{j-1} v)_A \leq K \left(\sum_{i=0}^{k}(S_i v, v)_A\right)^{1/2} \left(\sum_{j=0}^{k}(S_j E_{j-1} v, E_{j-1} v)_A\right)^{1/2} \tag{29}$$

Submitting Eqs. (26) and (29) into Eq. (25), dividing both sides by $1 + K\left(\sum_{i=0}^{k}(S_i v, v)_A\right)^{1/2}$ and then squaring both sides we have

$$\sum_{i=0}^{k}(S_i E_{i-1} v, E_{i-1} v)_A \geq \frac{1}{(1+K)^2}\sum_{i=0}^{k}(S_i v, v)_A \tag{30}$$

Submitting this expression into Eq. (24)

$$\|v\|_A^2 - \|E_k v\|_A^2 \geq \frac{1}{(1+K)^2}\sum_{i=0}^{k}(S_i v, v)_A \tag{31}$$

When $k$ is large enough, we get

$$\sum_{i=0}^{k}(S_i v, v)_A \geq (v, v)_A$$

and hence,

$$\|E_k v\|_A \leq \sqrt{1 - \frac{1}{(1+K)^2}}\|v\|_A, \quad \forall v \in \mathbb{R}^n \tag{32}$$

By now, we have given another proof of Theorem 3.1, *i.e.*, $\lim_{k\to\infty} E_k = 0$.

*Remark* Since $K$ increases with increasing $k$, we can see from Eq. (32) that $\|E_k\|$ decreases very rapidly in the beginning of the MSD-CG (and the GIPF-CG) method because $K$ is small. In the latter numerical experiments (see for example Fig. 2 in Sec. 4), we can see that the

decreasing rate of residual norm of the GIPF-CG method is even faster than that of the CG method. As $k$ increases, the decreasing rate of $\|E_k\|$ tends to be slow, and this results in the convergent rate of the GIPF-CG method becoming slightly slower than that of the CG method. But from the result of Theorem 3.2, the MSD-CG method converges at least as fast as the steepest descent method. What we find is that the convergent rate of steepest descent method is a lower bound for that of the MSD-CG method. Hence, as a whole, the convergent rate of the MSD-CG method is at least as fast as that of the steepest descent method.

## 4 NUMERICAL EXPERIMENTS

Consider a two-dimensional nonlinear nonsteady diffusion equation

$$q\frac{\partial u}{\partial t} = c_x\frac{\partial^2 u}{\partial x^2} + c_y\frac{\partial^2 u}{\partial y^2} + b_x\frac{\partial u}{\partial x} + b_y\frac{\partial u}{\partial x} + eu + g$$

on the unit square. The initial condition is $u(x, y, 0) = f(x, y)$ and zero Dirichlet boundary conditions are used, where

$$q = c_v + \frac{a_0 u^3}{\rho}; \quad c_x = \frac{a_1 u^{m_x}}{\rho}; \quad c_y = \frac{a_2 u^{m_y}}{\rho};$$
$$b_x = c_1 \sin(2\pi x) + c_2; \quad b_y = d_1 \sin(2\pi y) + d_2;$$

and $c_v, a_0, a_1, a_2, p, m_x, m_y, c_1, c_2, d_1, d_2, e$ and $g$ are constants.

We take two models as follows

*Model 1* Take $c_v = a_0 = m_x = m_y = c_1 = c_2 = d_1 = d_2 = e = g = 0.0$; $a_1 = a_2 = 1512.733$; and $\rho = 0.79$;

*Model 2* Take $c_v = 93.0$; $a_0 = 0.007568$; $m_x = m_y = 3$; $c_2 = d_2 = e = g = 0.0$; $c_1 = d_1 = 9.0$; $a_1 = a_2 = 1512.733$; and $\rho = 0.79$.

The way of domain decomposition depends on the number of processors. For example, if there are 16 processors available, we can take 1*16, 2*8, 4*4 and so on (see Fig. 1), Subdomains are ordered from left to right and from lower to upper. Nodes in each subdomain are numbered in nature ordering. We use the MPI parallel programming platform which maps each subdomain on the processor, correspondingly.

We carry out a discretization of the above equation by the silence coefficient method and five, seven (two type) or nine point difference scheme; the linear system obtained is a block five diagonal, block seven diagonal or block nine diagonal. For example, take 16*16 grid

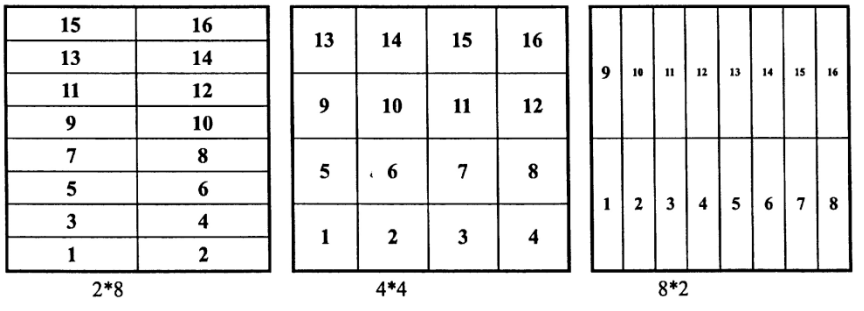


FIGURE 1 Examples for domain decomposition.

partitioning, 4*4 domain decomposition and a nine point difference scheme, the coefficient matrix of the obtained linear system is as follows:

$$A = \begin{pmatrix}
A_1 & B_1 & & & C_1 & E_1 & & & & & & & & & & \\
F_1 & A_2 & B_2 & & D_2 & C_2 & E_2 & & & & & & & & & \\
 & F_2 & A_3 & B_3 & & D_3 & C_3 & E_3 & & & & & & & & \\
 & & F_3 & A_4 & & & D_4 & C_4 & & & & & & & & \\
G_1 & H_2 & & & A_5 & B_5 & & & C_5 & E_5 & & & & & & \\
I_1 & G_2 & H_3 & & F_5 & A_6 & B_6 & & D_6 & C_6 & E_6 & & & & & \\
 & I_2 & G_3 & H_4 & & F_6 & A_7 & B_7 & & D_7 & C_7 & E_7 & & & & \\
 & & I_3 & G_4 & & & F_7 & A_8 & & & D_8 & C_8 & & & & \\
 & & & & G_5 & H_6 & & & A_9 & B_9 & & & C_9 & E_9 & & \\
 & & & & I_5 & G_6 & H_7 & & F_9 & A_{10} & B_{10} & & D_{10} & C_{10} & E_{10} & \\
 & & & & & I_6 & G_7 & H_8 & & F_{10} & A_{11} & B_{11} & & D_{11} & C_{11} & E_{11} \\
 & & & & & & I_7 & G_8 & & & F_{11}^{\mathrm{T}} & A_{12} & & & D_{12} & C_{12} \\
 & & & & & & & & G_9 & H_{10} & & & A_{13} & B_{13} & & \\
 & & & & & & & & I_9 & G_{10} & H_{11} & & F_{13} & A_{14} & B_{14} & \\
 & & & & & & & & & I_{10} & G_{11} & H_{12} & & F_{14} & A_{15} & B_{15} \\
 & & & & & & & & & & I_{11} & G_{12} & & & F_{15} & A_{16}
\end{pmatrix}$$

where $A_i$ is nine diagonal, $B_i$ and $C_i$ are tridiagonal, $D_i$ and $E_i$ are signal point matrices. Their nonzeros patterns are shown in Figure 2. The nonzero patterns of matrices, $F_i$, $G_i$, $H_i$ and $I_i$ are symmetric with those of matrices $B_i$, $C_i$, $D_i$ and $E_i$, respectively.

We distribute the $i$th row block of matrix $A$ on the $i$th processor. When a vector is multiplied by matrix $A$, it requires communication between the $i$th and $j$th processor for $i \neq j$. We see that in this structure of $A$, each processor needs only communication with its neighbor processors. For different distributed memory parallel computer and a different number of processors, we can choose a suitable mesh partition, domain decomposition, and discrete scheme in order to bring the latency performance of parallel computers into play.

The system obtained is symmetric positive definite for Model 1 and positive definite but nonsymmetric for Model 2. It is suitable to solve this by the GIPF-CG method. We do a lot of experiments which compare the convergence and speedup of the CG method, the GIPF-CG method, the IBJ-CG($m$) and two preconditioned methods on two parallel systems: Dawning 3000 and P-II Cluster.

Dawning 3000 is a scalable superserver system produced by the Institute of Computing Technology of China Academy of Science in 2001. The system is based on distributed memory and message passing. It contains 70 nodes, and a console with 64 nodes is used for computing disposed 4-way Power-II CPU, 375 MHz main frequency and 6 nodes for severing disposed IBM 4-way PowerPC RS64-III CPU, 400 MHz main frequency. This provides 403.2 Gflop/s peak value velocity, 168 GB main memory, 630 GB special HD and 3TB share HD.

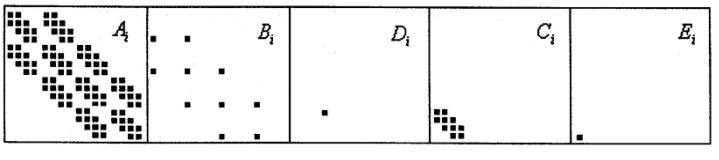


FIGURE 2 Nonzeros pattern of nine point matrix.

P-II Cluster is a distributed memory cluster system design by ourselves. It contains nine high-performance PCs (Pentium-II micro-processor, 400 MHz main frequency, 32 and 512 KB of two levels cache, 400 Mflop/s peak value velocity and 256 MB main memory) connected by a 100 Mbps switchboard.

In the following figures, Figures 3–7 are for Model 1. 400 * 400 and so on denote the scale of the problems (*i.e.*, cases for grid partitioning). 4 * 4 CPU and so on denote the ways of domain decomposition (*i.e.*, cases for mapping on processor). Number 2 in GIPF-CG_2 indicates that the small systems are solved by two Jacobi iterations.

In Figure 3, (a)–(c) show the residual reductions of CG and GIPF-CG_2 on 16 CPU of Dawning 3000. Results for other decompositions and the scales of the problem are similar. They state that in order to obtain a high accuracy ($<10^{-6}$) the convergence rate of the GIPF-CG is 0–30% slower than CG and a special case given in (a) shows that GIPF-CG is faster than CG. For partitions of CPU and scales of the problem, we see that GIPF-CG is faster than CG at the beginning (before $<10^{-2}$) of their implement. The increase in number of Jacobi iterations for the small systems can improve the convergence of the GIPF-CG method but needs more CPU and wall time.

Comparisons of wall time between the CG and the GIPF-CG methods are shown in (a)–(c) of Figure 4. The number of GIPF-CG iterations for each scale takes the same number for the residual 2 normal of CG reduced to $10^{-2}$. In Figure 4, the ordinate denotes wall time in seconds, the abscissa equal to 0 denotes the CG method and 1–9 denotes GIPF-CG in which the small systems are solved by 1–9 Jacobi iterations. We can see that the GIPF-CG method takes a wall time less than or equal to that of the CG method when the Jacobi iteration is 1–4. This is because the global communication has been replaced by local communication.

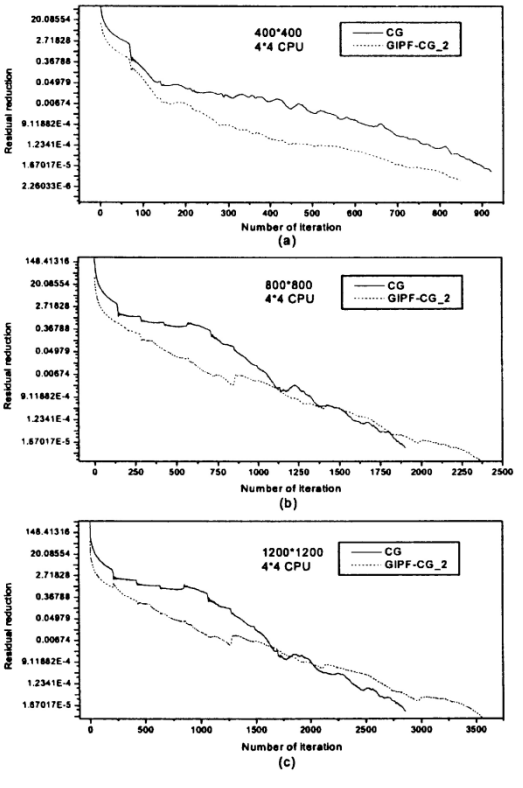


FIGURE 3 Residual reductions of CG and GIPF-CG_2.

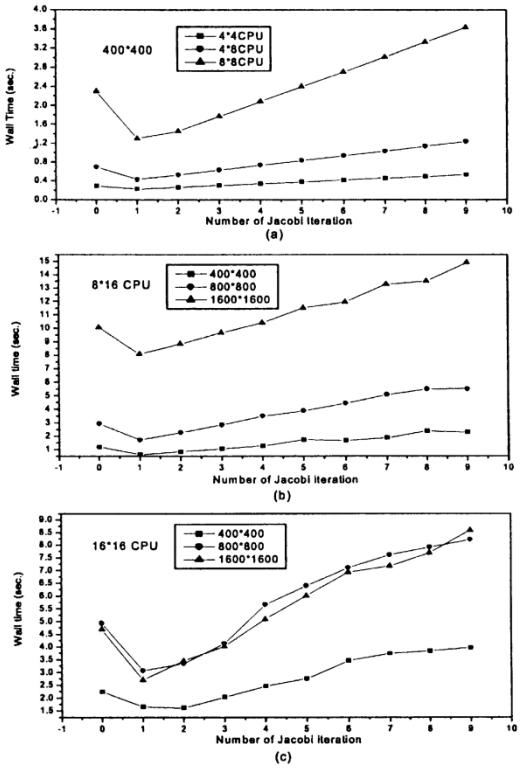


FIGURE 4 Wall time of CG and GIPF-CG.

GIPF-CG can be more accurate when the Jacobi iteration equals 5–9, but this takes more wall time. Therefore, for the same accuracy, the GIPF-CG method can achieve a reasonable balance convergence, computing time and communication time when the small system is solved by 2–4 Jacobi iterations.

In Figure 5, (a)–(c) show the speedup of CG and GIPF-CG_2 on Dawning 3000 when 2000 iterations are performed. These show that the rate of speedup for the CG method is lower than that of the GIPF-CG method when the number of processors increases. We cannot obtain an ideal acceleration for the CG method when the number of processors is large enough. This is just the aim for us to propose the MSD-CG and the GIPF-CG methods. Also, the increase in speedup for GIPF-CG is almost linear (or even superlinear).

Figure 6(a) and (b) show the residual reduction for the CG and the GIPF-CG methods when they take no preconditioning (No), diagonal scaled preconditioning (Diagscal) and main diagonal block approximate inverse (AINV) preconditioning on P-II Cluster. They show that the diagonal block AINV preconditioned method is fastest and is almost three times faster than no preconditioned method. But the constructed overhead for this kind of preconditioner is almost unacceptable, and this shows that an AINV preconditioner is not very suitable for large-scale problems. However, the construction and performance for a diagonal scaled preconditioner is very easy and has nature parallelism. The construction and choice of preconditioner should be considerations for future research.

Comparisons of residual reduction of the CG, the GIPF-CG and the IBJ-CG(10) methods are given in (a)–(c) of Figure 7. Although the GIPF-CG method is slower than CG for a high degree of accuracy ($<10^{-6}$), it is much better than the IBJ-CG($m$) method because of the IBJ-CG($m$) lack of transportation of global information although its parallelism is better. The

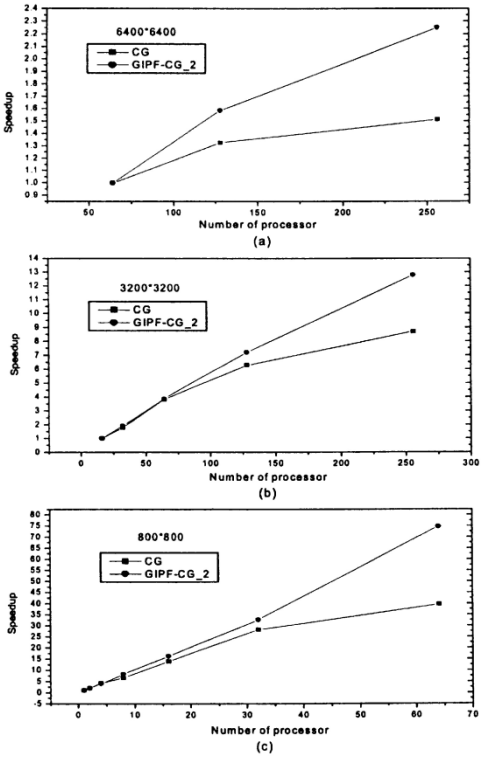


FIGURE 5 Speedup of CG and GIPF-CG_2.

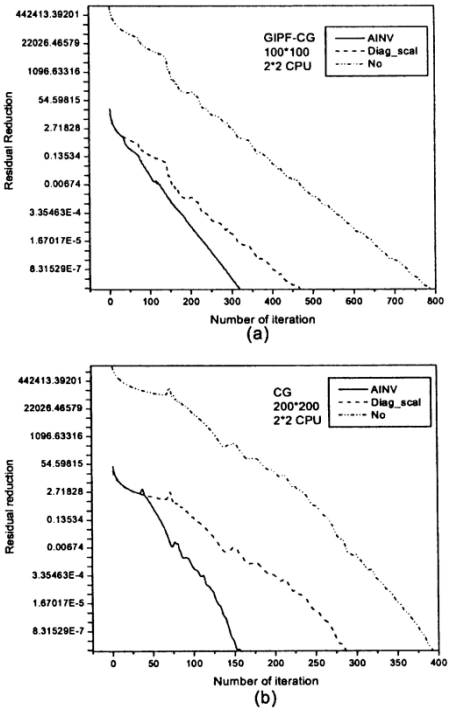


FIGURE 6 CG and GIPF-CG with or without preconditioner.

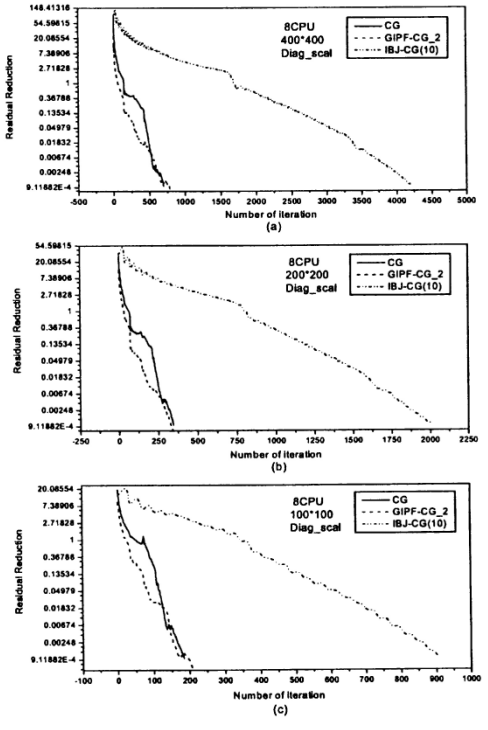


FIGURE 7 CG, the GIPF-CG and IBJ-CG(10) methods.

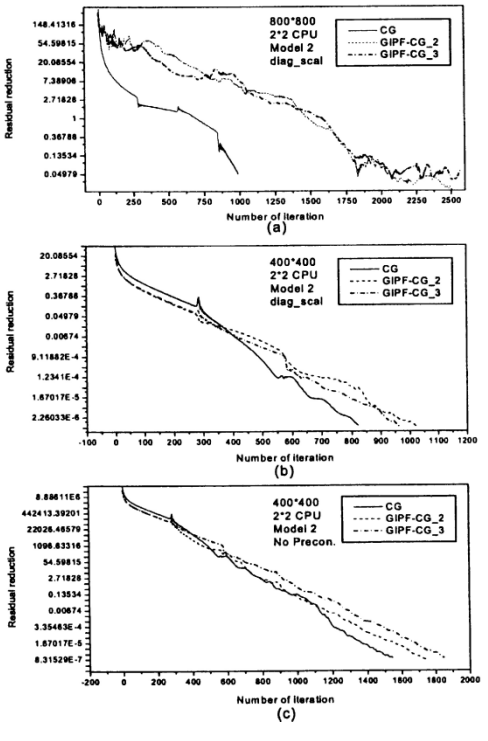


FIGURE 8 CG and the GIPF-CG method for Model 2.

GIPF-CG method eliminates global communication, but the global information contains the solution of the small systems. Therefore, its convergence rate does not decrease too much (associated with the CG method and high accuracy).

Figure 8(a)–(c) compare the residual reduction of the CG and the GIPF-CG methods with or without diagonal scaled preconditioning for Model 2 on P-II Cluster. They say that the convergence curve is similar to that of Model 1 for $400 * 400$. But they are not similar for $800 * 800$, *i.e.*, the residual reduction curve of the GIPF-CG method shows a higher oscillation. The theory of the GIPF-CG method for nonsymmetric problems should be studied further.

## 5 CONCLUSION

In Ref. [10], we proposed the MSD-CG method by introducing multiple search directions. The method replaces computing of inner products in the CG method by solving small linear systems. Its approximate version, the GIPF-CG method, eliminates global communication entirely for problems which have a special structure. Since it has a better parallelism, it is a good alternative to the CG method on massively parallel computers.

From a large number of numerical experiments, we can see that although the convergence rate of the GIPF-CG method is slightly lower than that of the CG method, it can be remedied by the gains from eliminating global communication. Preconditioning can improve the performance of the GIPF-CG method enormously. A simple and natural parallel diagonal scaled preconditioner is particularly useful and efficient. Serial computing, e.g., using an AINV preconditioner is not suitable for large-scale parallel computation since it needs a large amount of time for construction.

GIPF-CG can achieve a reasonable balance between convergence, computing time and communication time when it takes two to four inner Jacobi iterations for the small systems. Our method is four to five times faster than the IBJ-CG($m$) method. Future studies should address how to choose a better parallel preconditioner and the theory for nonsymmetric problems.

It is necessary to point out that the MSD-CG and the GIPF-CG methods are based on the idea of the domain decomposition method. Because small linear systems need to be solved at each step, the global information is consistent in small systems, *i.e.*, the GIPF-CG method communicates the global information at each step. This overcomes the limitations of the additive Schwarz method, in which the convergent rate decreases considerably with increasing number of domains because of a lack of global information.

The MSD-CG method can be viewed as a more general nonstationary iterative of the nonstationary Richardson iterative method with Chebyshev semi-iterative acceleration. From the proof of convergence, the MSD-CG method is at least as fast as the steepest descent method.

### *Acknowledgements*

This project is supported partly by the State Hi-Tech Research and Development Program (863), the Natural Science Fund of China (60373015) and the Foundation of National Key Laboratory of Computational Physics.

### *References*

[1] Axelsson, O. and Vassilevski, P. S. (1991). A black box generalized conjugate gradient solver with inner iterations and variable-step preconditioning. *SIAM J. Matrix Anal. Appl.*, **12**, 625–644.

[2] Basermann, A. (1997). Conjugate gradient and Lanczos methods for sparse matrices on distributed memory multiprocessors. *J. Parallel Distr. Comput.*, **45**, 46–52.

[3] Basermann, A., Reichel, B. and Schelthoff, C. (1997). Preconditioned CG methods for sparse matrices on massively parallel machines. *Parallel Comput.*, **23**, 381–398.
[4] Chronopoulos, A. T. and Gear, C. W. (1989). s-Step iterative methods for symmetric linear systems. *J. Comp. Appl. Math.*, **25**, 153–168.
[5] Crone, L. and van der Vorst, H. A. (1993). Communication aspects of the conjugate gradient method on distributed-memory machines. *Supercomputer*, **X**(6), 4–9.
[6] D'Azevedo, E. F. and Romine, C. (1993). LAPACK working note 56: reducing communication costs in the conjugate gradient algorithm on distributed memory multiprocessors. *Technical Reports*. Computer Science Department, University of Knoxville, Knoxville, TN.
[7] da Sturler, E. (1996). A performance model for Krylov subspace methods on mesh-based parallel computers. *Parallel Comput.*, **22**, 57–74.
[8] Demmel, J. W., Health, M. T. and van der Vorst, H. A. (1993). *Parallel Numerical Linear Algebra*. In Acta Numerica 1993. Cambridge University Press, Cambridge.
[9] Field, M. R. (1998). Optimizing a parallel conjugate gradient solver. *SIAM J. Sci. Comput.*, **19**, 27–37.
[10] Gu, T.-X., Liu, X.-P., Mo, Z.-Y. and Chi, X.-B. (in press). Multiple search direction conjugate gradient method I: Methods and their propositions. *Int. J. Comput. Math.*
[11] Hesrenses, M. R. and Stiefel, E. (1952). Method of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, **49**, 409–436.
[12] Meurant, G. (1984). The block preconditioned conjugate gradient method on vector computers. *BIT*, **24**, 623–633.
[13] Meurant, G. (1987). Multitasking the conjugate gradient method on the CRAY X-MP/48. *Parallel Comput.*, **5**, 267–280.
[14] O'Leary, D. P. (1980). The block conjugate gradient algorithm and related methods. *Lin. Alg. Appl.*, **29**, 293–322.
[15] Radicati di Brozolo, G. and Robert, Y. (1989). Parallel conjugate gradient-like algorithms for solving sparse non-symmetric systems on a vector multiprocessor. *Parallel Comput.*, **11**, 223–329.
[16] Reid, J. K. (1971). On the method of conjugate gradients for the solution of large sparse systems of linear equation. In: J. K. Reid (Ed.) *Large Sparse Sets of Linear Equations*. Academic Press, pp. 231–254.
[17] Saad, Y. (1981). Krylov subspace methods for solving large unsymmetric linear systems. *Math. Comput.*, **155**, 105–126.
[18] Saad, Y. (1989). Krylov subspace methods on supercomputers. *SIAM J. Sci. Comput.*, **10**, 1200–1232.
[19] Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston.
[20] Tang, W. P. (1992) Generalized Schwarz splittings. *SIAM J. Sci. Stat. Comput.*, **13**, 573–595.
[21] Jinchao Xu (1992). Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, **34**(4), 581–613.
[22] Young, D. M. (1971). *Iterative Solution of Large Linear Systems*. Academic Press, NY.