Journal of Computational Mathematics
Vol.32, No.3, 2014, 284–296.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1401-CR1

PARALLEL QUASI-CHEBYSHEV ACCELERATION TO NONOVERLAPPING MULTISPLITTING ITERATIVE METHODS BASED ON OPTIMIZATION*

Ruiping Wen
Higher Education Key Laboratory of Engineering Science Computing in Shanxi Province,
Taiyuan Normal University, Taiyuan 030012, China
Email: wenrp@163.com
Guoyan Meng
Department of Computer Science, Xinzhou Normal University, Xinzhou 034000, China
Email: mgy1226@163.com
Chuanlong Wang
Higher Education Key Laboratory of Engineering Science Computing in Shanxi Province,
Taiyuan Normal University, Taiyuan 030012, China
Email: clwang1964@163.com

Abstract

In this paper, we present a parallel quasi-Chebyshev acceleration applied to the nonoverlapping multisplitting iterative method for the linear systems when the coefficient matrix is either an H-matrix or a symmetric positive definite matrix. First, m parallel iterations are implemented in m different processors. Second, based on l_1-norm or l_2-norm, the m optimization models are parallelly treated in m different processors. The convergence theories are established for the parallel quasi-Chebyshev accelerated method. Finally, the numerical examples show that the parallel quasi-Chebyshev technique can significantly accelerate the nonoverlapping multisplitting iterative method.

Mathematics subject classification: 65F10, 65F50, 15A06.
Key words: Parallel quasi-Chebyshev acceleration, Nonoverlapping multisplitting iterative method, Convergence, optimization.

1. Introduction and Preliminaries

A multisplitting of a nonsingular matrix $A \in R^{n\times n}$, as introduced in [12], is a collection of triples of $n \times n$ matrices $(M_i, N_i, E_i)_{i=1}^{m}(m \le n$, a positive integer) with

- $A = M_i - N_i, \quad i = 1, \cdots, m;$
- M_i nonsingular, $i = 1, \cdots, m;$
- for $i = 1, \cdots, m$, the weighting matrices $E_i = diag(e_1^{(i)}, \cdots, e_n^{(i)})$ being diagonal with nonnegative entries

$$e_j^{(i)} = \begin{cases} e_j^{(i)} > 0, & \text{for } j \in \mathcal{N}_i, \\ 0, & \text{for } j \notin \mathcal{N}_i, \end{cases} \qquad j = 1, \cdots, n,$$

* Received August 17, 2013 / Revised version received December 19, 2013 / Accepted January 15, 2014 / Published online May 22, 2014 /

such that $\sum_{i=1}^{m} E_i = I$ (the $n \times n$ identity matrix), where $\mathcal{N}_i, i = 1, \cdots, m$ are nonempty subsets of $\mathcal{N}, \mathcal{N} = \{1, \cdots, n\}$ satisfying $\mathcal{N} = \bigcup_{i=1}^{m} \mathcal{N}_i$, see also [4]. Then the (parallel linear) multisplitting iterative method for solving the linear system of equations

$$Ax = b \tag{1.1}$$

is

$$M_i x_i^{(k)} = N_i x^{(k-1)} + b, \quad i = 1, \cdots, m; \quad k = 1, \cdots, \tag{1.2}$$

$$x^{(k)} = \sum_{i=1}^{m} E_i x_i^{(k)}. \tag{1.3}$$

In particular, the above method is called a nonoverlapping multisplitting iterative method if $\mathcal{N}_{i_1} \cap \mathcal{N}_{i_2} = \emptyset$ $(i_1 \neq i_2)$.

Subsequently, many authors studied the above technique for the cases where A is an M-matrix, an H-matrix or a symmetric positive definite matrix respectively, we refer to [1,4,6,9, 11,13,17] and the references therein. The idea of minimizing the norm of either the error or the residual so that the numerically optimal value of the iteration parameter is determined, first introduced in [2], used to compute a numerically optimal relaxation parameter for the successive overrelaxation (SOR) iteration methods for solving the system of linear equations. Based on the standard quadratic programming technique, the authors of this paper and their collaborators [14,15] seem to be the first to introduce the auto-optimal weighting matrices $E_i^{(k)}, i = 1, \cdots, m;\ k = 1, \cdots$ for parallel multisplitting iterative methods. The self-adaptive weighting matrices enable more approximate to the exact solution for k-step iteration. These methods, however, just as introduced in [14] and [15], only one processor of the multiprocessor system to compute the global optimization model at every iteration step, the other $m-1$ processors must be in the waiting state until one of all these processors has finished its optimization task.

As is well known, one of the best accelerated methods is the Chebyshev semi-iteration, in which the optimum parameter ω is obtained by the Chebyshev polynomial. The one most recent result may be found the quasi-Chebyshev accelerated (QCA) method to convergent splitting iteration proposed in [16]. The method is, in spirit, analogous to the Chebyshev semi-iteration but the optimum parameter ω is generated by optimization model for solving the linear systems. These motivated us to accelerate the parallel multisplitting iterative method, resulting in a parallel quasi-Chebyshev accelerated (abbreviated as PQCA) method to the nonoverlapping multisplitting iterative method for the linear systems when the coefficient matrices are H-matrices or symmetric positive definite matrices. To make full use of the efficiency of a multiprocessor system and overcome the drawbacks of those methods in [14,15], we further divide the global optimization model into m sub-models and hence, the parallel computing is achieved in this paper.

The PQCA method determines the optimum parameters $\alpha^{(i)}, i = 1, \cdots, m$ through minimizing either the l_2-norm of the residual when the coefficient matrix of (1.1) is a symmetric positive definite matrix, or the l_1-norm of the residual when the coefficient matrix of (1.1) is an H-matrix, at each step of their iterates, at each processor of multiprocessor system, with a reasonably extra cost. In actual computations, that shows better numerical behaviors than the Method in [9] for both the symmetric positive definite matrix and the H-matrix. Numerical experiments show that the new PQCA method is feasible, efficient and robust for solving large sparse system of linear equations (1.1).

In this study, we first give some notations and preliminaries in Section 1, and then the parallel quasi-Chebyshev acceleration method is put forward in Section 2, the convergence of the PQCA method are established in Section 3. Finally, numerical results are given in Section 4 to show the effectiveness of this new acceleration to the nonoverlapping multisplitting iterative method.

Here are some essential notations and preliminaries. $R^{n\times n}$ is used to denote the $n \times n$ real matrix set, and R^n the n-dimensional real vector set. A^T denotes the transpose sample of the matrix A, and x^T the transpose sample of the vector x. $\langle A\rangle$ represents the comparison (Ostrowski) matrix of the matrix A, and $|A|$ stand for the absolute matrix of the matrix A. A matrix $A \in R^{n\times n}$ is called real symmetric positive definite, if for all $x \in R^n, x \neq 0$, it holds that $x^TAx > 0$. The l-step stationary multisplitting of a nonsingular matrix A is denoted by $(M_i, N_i, l, E_i)_{i=1}^m$, where l is the number of local iteration (see [9]).

$A = M - N$ is called a splitting of the matrix $A \in R^{n\times n}$ if $M \in R^{n\times n}$ is nonsingular, the splitting is convergent if $\rho(M^{-1}N) < 1$. If A is symmetric positive definite and $M^T + N$ is positive definite, then we call the splitting $A = M - N$ to be a P-regular splitting (see [5]).

In what follows, the matrix $A = (a_{ij}) \in R^{n\times n}$ is called a strictly diagonally dominant matrix if and only if

$$|a_{jj}| > \sum_{\substack{i=1\\ i\neq j}}^{n} |a_{ij}|, \quad j = 1, \cdots, n.$$

Definition 1.1 ([5]). *The matrix A is an H-matrix if there exists a positive diagonal matrix D such that the matrix DA is a strictly diagonally dominant matrix.*

Property 1.2 ([5]). *The matrix A is an H-matrix if and only if $\langle A\rangle$ is an M-matrix.*

Definition 1.3 ([7]). *Suppose that A is an H-matrix. Let $A = M - N$, which is called an H-compatible splitting if $\langle A\rangle = \langle M\rangle - |N|$.*

2. Parallel Quasi-Chebyshev Acceleration Method

In this section, we establish the parallel quasi-Chebyshev accelerated method applied to the nonoverlapping multisplitting iterative method in cases that the coefficient matrix $A \in R^{n\times n}$ of the system of linear equations (1.1) is an H-matrix or a symmetric positive definite matrix.

In fact, the foregoing nonoverlapping variant of a multisplitting iteration as follows:

$$A = M_i - N_i, \quad i = 1, \cdots, m; \tag{2.1}$$

$$E_i = diag(e_1^{(i)}, \cdots, e_n^{(i)}), \tag{2.2}$$

with

$$e_j^{(i)} = \begin{cases} 1, & \text{for } j \in \mathcal{N}_i, \\ 0, & \text{for } j \notin \mathcal{N}_i, \end{cases} \quad j = 1, \cdots, n.$$

Assume that the cardinal numbers of those sets $\mathcal{N}_i,\ i = 1, \cdots, m$ are n_i, we then have $n_1 + \cdots + n_m = n$. By introducing

$$T_i = M_i^{-1}N_i, \quad i = 1, \cdots, m,$$

the iteration matrix of kth step

$$T_k = \sum_{i=1}^{m} E_iM_i^{-1}N_i = \sum_{i=1}^{m} E_iT_i. \tag{2.3}$$

Method 2.1 (*PQCA method*)

Given a precision $\epsilon > 0$ and an initial point $x^{(0,0)}$. For $k = 0, 1, 2, \cdots$ until the iteration sequence $\{x^{(k,0)}\}_{k=0}^{\infty}$ convergence,

- Step 1. For $t = 1, \cdots, l$, compute in parallel the vectors $x_{n_i}^{(k,t)} = (x_{i1}^{(k,t)}, \cdots, x_{in_i}^{(k,t)})^T \in R^{n_i}$ by
$$M_i x_i^{(k,t)} = N_i x_i^{(k,t-1)} + b, \quad i = 1, \cdots, m$$
in ith processor. Here, $x_i^{(k,t)} = (x_{i1}^{(k,t)}, \cdots, x_{in}^{(k,t)})^T \in R^n$.

- Step 2. For $i = 1, \cdots, m$, let
$$x^{(i)} = ((x_{n_1}^{(k,l)})^T, \cdots, (\alpha^{(i)}(x_{n_i}^{(k,l)} - x_{n_i}^{(k,l-2)}) + x_{n_i}^{(k,l-2)})^T, \cdots, (x_{n_m}^{(k,l)})^T)^T \in R^n.$$
For $i = 1, \cdots, m$, compute in parallel the approximate solutions of $\alpha^{(i)}$ by the following optimization problems:

  (a) when A is a symmetric positive definite matrix
$$\min_{\alpha^{(i)}} \frac{1}{2}(x^{(i)})^T A x^{(i)} - (x^{(i)})^T b, \quad i = 1, \cdots, m; \tag{2.4}$$

  (b) when A is an H-matrix
$$\min_{\alpha^{(i)}} ||Ax^{(i)} - b||_1, \quad i = 1, \cdots, m. \tag{2.5}$$

  We denote the solutions of (2.4) (or (2.5)) by $\bar{\alpha}^{(i)}$ and then
$$\bar{x}_{n_i} = (\bar{\alpha}^{(i)}(x_{n_i}^{(k,l)} - x_{n_i}^{(k,l-2)}) + x_{n_i}^{(k,l-2)})^T \in R^{n_i}, \quad i = 1, \cdots, m.$$

- Step 3. Let $x^{(k+1,0)} = \arg\min_x \frac{1}{2}x^T Ax - x^T b$ or $x^{(k+1,0)} = \arg\min_x \|Ax - b\|_1$ with $x \in \{\bar{x}, x^{(k,l)}\}$, where
$$x^{(k,l)} = ((x_{n_1}^{(k,l)})^T, \cdots, (x_{n_m}^{(k,l)})^T)^T \in R^n,$$
$$\bar{x} = ((\bar{x}_{n_1})^T, (\bar{x}_{n_2})^T \cdots, (\bar{x}_{n_m})^T)^T \in R^n.$$

- Step 4. If $||Ax^{(k+1,0)} - b|| \le \epsilon$, stop; otherwise, $k \Leftarrow k + 1$, goto Step 1.

Remark 2.1. In fact, the exact solutions of the quadratic programming models (2.4) are as follows:
$$\bar{\alpha}^{(i)} = \Big((X_{k,l}^{(i)})^T A_{ii} X_{k,l}^{(i)}\Big)^{-1} (X_{k,l}^{(i)})^T \Big(b_{n_i} - \sum_{j \ne i} A_{ij} X_{k,l}^{(j)}\Big), \quad i = 1, \cdots, m, \tag{2.6}$$
where $X_{k,l}^{(i)} = (x_{n_i}^{(k,l)} - x_{n_i}^{(k,l-2)}) \in R^{n_i}$.

To avoid the tedious computation of $((X_{k,l}^{(i)})^T A_{ii} X_{k,l}^{(i)})^{-1}$, however, we can use the inexact line search to find the approximations of $\bar{\alpha}^{(i)}, \ i = 1, \cdots, m$.

A remark that is similar to Remark 2.1 can also be stated for the quadratic programming models (2.5).

In ith processor, the cost of each iterate of Method 2.1 is approximately $l[2nn_i+(n_i-1)^2]+2nn_i+2n_i-1$ and that of the Method in [9] is $l[2nn_i+(n_i-1)^2]$, then the cost of Method 2.1 is about

$$1+\frac{2nn_i+2n_i-1}{l[2nn_i+(n_i-1)^2]}$$

times of that of the Method in [9]. Particularly, the times is about

$$1+\frac{2n^2m+2nm-m^2}{l[(2m+1)n^2-2mn+m^2]}\approx 1+\frac{2m}{(2m+1)l},$$

when $n_i=n/m$ and n is reasonably large.

The average speed-up of a linear parallel iterative method, as defined in [3], is

$$S_m=\frac{W_s\times N_s}{W_p\times N_p}, \tag{2.7}$$

where $W_p=\max_{1\le i\le m}\{w_i\}$ represents the maximum work of m processors of a linear parallel iterative method and W_s the work of the corresponding linear serial iterative method at each iterate step, N_p and N_s denote their iteration step number respectively under the unified precision. Thus,

$$\frac{S_m}{S'_m}=\frac{N'_p\{l[2nn_{i_0}+(n_{i_0}-1)^2]+2nn_{i_0}+2n_{i_0}-1\}}{N_p\{l[2nn_{i_0}+(n_{i_0}-1)^2]\}},\quad n_{i_0}=\max_{1\le i\le m}\{n_i\} \qquad (2.8),$$

with S_m and S'_m are the average speed-up of the Method in [9] and Method 2.1, respectively. Hence, Method 2.1 is more efficient than the Method in [9] if and only if $S_m/S'_m<1$.

Specially, if $n_i=n/m, i=1,\cdots,m$ (say), then (2.8) can be written as

$$\frac{S_m}{S'_m}\approx\frac{[(2m+1)l+2m]N'_p}{[(2m+1)l]N_p}. \tag{2.9}$$

3. Convergence Analysis

This section is devoted to study the convergence theories for Method 2.1 under the reasonable assumptions.

Lemma 3.1([8, Theorem 2.4]). *Let A be a symmetric positive definite matrix. Then $A=M-N$ is a P-regular splitting if and only if there exists a positive real number r such that*

$$\|A^{\frac{1}{2}}TA^{-\frac{1}{2}}\|_2\le r<1.$$

Lemma 3.2. *Let $A=M_i-N_i, i=1,\cdots,m$ be m P-regular splittings of the symmetric positive definite matrix A and $E_i=diag(0,\cdots,0,I_i,0,\cdots,0)$ with $I_i, i=1,\cdots,m$ be $n_i\times n_i$ identity matrices. If $\|A^{\frac{1}{2}}M_i^{-1}N_iA^{-\frac{1}{2}}\|_2=r_i, i=1,\cdots,m$ and*

$$l>-\frac{1}{\ln r}\ln\Big(\sum_{i=1}^{m}||A^{\frac{1}{2}}E_iA^{-\frac{1}{2}}||_2\Big),$$

where $r=\max_{1\le i\le m}r_i$, then the splitting $A=P-Q$ induced by the stationary multisplitting $(M_i,N_i,l,E_i)_{i=1}^m$ is a P-regular splitting.

Proof. For Method 2.1, it follows, as in [9], that the iteration matrix is

$$T = P^{-1}Q = \sum_{i=1}^{m} E_i (M_i^{-1} N_i)^l. \tag{3.1}$$

Similar to the proof of Theorem 2.1 in [9], we have

$$\begin{aligned} \|A^{\frac{1}{2}} T A^{-\frac{1}{2}}\|_2 &\le \max_i \|A^{\frac{1}{2}} M_i^{-1} N_i A^{-\frac{1}{2}}\|_2^l \cdot \sum_{i=1}^{m} \|A^{\frac{1}{2}} E_i A^{-\frac{1}{2}}\|_2 \\ &\le \sum_{i=1}^{m} \|A^{\frac{1}{2}} E_i A^{-\frac{1}{2}}\|_2 \max_i r_i^l < 1. \end{aligned}$$

Thus, Lemma 3.1 implies that $A = P - Q$ is a P-regular splitting. □

Remark 3.1. In [9], the nonstationary multisplitting parallel iterative method with general weighting matrices is convergent when $s(i, k)$ is large enough. Here, we give the lower bound of stationary iteration number l for nonoverlaping multisplitting iterative method. In fact, this result can straightforwardly employ the nonstationary iteration number $s(i, k)$, introduced in [9], to obtain an analogs. Since our study focused on parallel quasi-Chebyshev acceleration, only the stationary multisplitting parallel iterative method is discussed.

Theorem 3.3. *Let $(M_i, N_i, l, E_i)_{i=1}^m$ be a stationary multisplitting of the symmetric positive definite matrix A and it satisfies the assumptions of Lemma* 3.2. *Then $\{x^{(k,0)}\}$ generated by Method* 2.1 *converges to the unique solution x_* of the linear system* (1.1).

Proof. From Lemma 3.2, the splitting $A = P - Q$ induced by the stationary multisplitting $(M_i, N_i, l, E_i)_{i=1}^m$ is a P-regular splitting. Let $\epsilon^{(k,0)} = x^{(k,0)} - x_*$. Then it holds that

$$\epsilon^{(k,l)} = T\epsilon^{(k,0)}, \tag{3.2}$$

where $T = \sum_{i=1}^m E_i (M_i^{-1} N_i)^l = P^{-1}Q$. On the other hand, (2.4) are equivalent to the following models in kth iteration,

$$\min_{\alpha^{(i)}} \frac{1}{2} (x^{(i)} - x_*)^T A (x^{(i)} - x_*), \quad i = 1, \cdots, m.$$

Therefore by noting that Method 2.1, we have

$$(\epsilon^{(k+1,0)})^T A \epsilon^{(k+1,0)} \le (\epsilon^{(k,l)})^T A \epsilon^{(k,l)}. \tag{3.3}$$

It follows from (3.2) and (3.3) that

$$\begin{aligned} \|A^{\frac{1}{2}} \epsilon^{(k+1,0)}\|_2 &\le \|A^{\frac{1}{2}} \epsilon^{(k,l)}\|_2 \le \|A^{\frac{1}{2}} T \epsilon^{(k,0)}\|_2 \\ &= \|A^{\frac{1}{2}} T A^{-\frac{1}{2}} A^{\frac{1}{2}} \epsilon^{(k,0)}\|_2 \le \|(A^{\frac{1}{2}} T A^{-\frac{1}{2}})\|_2 \|A^{\frac{1}{2}} \epsilon^{(k,0)}\|_2 \\ &\le \cdots \\ &\le \|A^{\frac{1}{2}} T A^{-\frac{1}{2}}\|_2^k \|A^{\frac{1}{2}} \epsilon^{(0,0)}\|_2. \end{aligned}$$

By noting Lemmas 3.1 and 3.2, there exists a positive real number θ such that

$$\|A^{\frac{1}{2}} T A^{-\frac{1}{2}}\|_2 \le \theta < 1.$$

Thus, we obtain

$$\lim_{k\to\infty}(\epsilon^{(k+1,0)})^T A\epsilon^{(k+1,0)} = 0,$$

which is alternatively equivalent to

$$\lim_{k\to\infty}\epsilon^{(k+1,0)} = 0.$$

This completes the proof of the theorem. □

Lemma 3.4. *Let $A = M - N$ be an H-compatible splitting of the strictly diagonally dominant matrix A. Then*

$$||NM^{-1}||_1 < 1. \tag{3.4}$$

Proof. We know that $\langle A\rangle = \langle M\rangle - |N|$ from the definition of H-compatible splitting. Hence, $M = (m_{ij}) \in R^{n\times n}$ is a strictly diagonally dominant matrix. Let $e = (1,\cdots,1)^T$. From Property 1.4, it holds that

$$||NM^{-1}||_1 \le |||N|\langle M\rangle^{-1}||_1 = \max_{1\le i\le n}|(e^T|N|\langle M\rangle^{-1})_i|.$$

Let $e^T|N|\langle M\rangle^{-1} = x^T$, where $x^T = (x_1,\cdots,x_n)$. We then have

$$e^T|N| = x^T\langle M\rangle. \tag{3.5}$$

Let $x_{i_0} = \max\limits_{1\le i\le n}\{x_i\}$ and $N = (n_{ij}) \in R^{n\times n}$. (3.5) implies that

$$\sum_{j=1}^n |n_{ji_0}| = m_{i_0i_0}x_{i_0} - \sum_{j\ne i_0} m_{ji_0}x_j \ge \Big(m_{i_0i_0} - \sum_{j\ne i_0} m_{ji_0}\Big)x_{i_0},$$

which yields

$$x_{i_0} \le \frac{\sum\limits_{j=1}^n |n_{ji_0}|}{m_{i_0i_0} - \sum\limits_{j\ne i_0} m_{ji_0}}. \tag{3.6}$$

By noting the splitting $\langle A\rangle = \langle M\rangle - |N|$ of the strictly diagonally dominant matrix $\langle A\rangle$ is an H-compatible splitting, it holds that the right hand side of (3.6) is bounded in $(0,1)$. We have obtained the lemma. □

Lemma 3.5. *Let $A = M - N$ be an H-compatible splitting of the H-matrix A. Then there exists a positive diagonal matrix D such that*

$$||DNM^{-1}D^{-1}||_1 < 1. \tag{3.7}$$

Proof. Because A is an H-matrix, there is a positive diagonal matrix D such that DA is a strictly diagonally dominant matrix. It follows that,

$$\langle DA\rangle = \langle DM\rangle - |DN|. \tag{3.8}$$

Let $\langle DA\rangle = \bar{A},\ \langle DM\rangle = \bar{M},\ |DN| = \bar{N}$. Then

$$\bar{A} = \bar{M} - \bar{N}$$

is a regular splitting of the strictly diagonally dominant matrix $\bar{A}$. From **Lemma 3.4**, we know that $||\bar{N}\bar{M}^{-1}||_1 < 1$. Hence,

$$||DNM^{-1}D^{-1}||_1 = ||\bar{N}\bar{M}^{-1}||_1 < 1, \tag{3.9}$$

which completed the proof. □

Theorem 3.6. *Let $A = M_i - N_i, i = 1, \cdots, m$ be m splittings of the H-matrix A and $E_i = diag(0, \cdots, 0, I_i, 0, \cdots, 0)$ with $I_i, i = 1, \cdots, m$ be $n_i \times n_i$ identity matrices. If the splitting $A = P - Q$ induced by the stationary multisplitting $(M_i, N_i, l, E_i)_{i=1}^m$ is an H-compatible splitting, then $\{x^{(k,0)}\}$ generated by Method 2.1 converges to the unique solution x_* of the linear system (1.1).*

Proof. Let $\epsilon^{(k,0)} = x^{(k,0)} - x_*$, we obtain

$$\epsilon^{(k,l)} = T\epsilon^{(k,0)},$$

where $T = \sum_{i=1}^m E_i(M_i^{-1}N_i)^l = P^{-1}Q$. Let D be a positive diagonal matrix so that DA is strictly diagonally dominant. From (2.5), analogously to Theorem 3.3 we can obtain that

$$\begin{aligned} ||DA\epsilon^{(k+1,0)}||_1 &\leq ||DA\epsilon^{(k,l)}||_1 = ||DAT\epsilon^{(k,0)}||_1 = ||DATA^{-1}D^{-1}D \cdot A\epsilon^{(k,0)}||_1 \\ &\leq ||DATA^{-1}D^{-1}||_1||DA\epsilon^{(k,0)}||_1 = ||D(ATA^{-1})D^{-1}||_1||DA\epsilon^{(k,0)}||_1 \\ &= ||D(QP^{-1})D^{-1}||_1||DA\epsilon^{(k,0)}||_1 \\ &\leq r||DA\epsilon^{(k,0)}||_1 \leq \cdots \leq r^{k+1}||DA\epsilon^{(0,0)}||_1, \end{aligned} \tag{3.10}$$

where $r = ||DQP^{-1}D^{-1}||_1$. By noting Lemma 3.5 it is easy to know that $r < 1$ and $\lim_{k\to\infty} ||A\epsilon^{(k+1,0)}||_1 = 0$, thus we obtain

$$\lim_{k\to\infty} \epsilon^{(k+1,0)} = 0. \tag{3.11}$$

We have completed the proof of this theorem. □

4. Numerical Experiments

In this section, numerical examples are given to illustrate the effectiveness of the PQCA to the nonoverlapping multisplitting iterative method (Method 2.1). We compare Method 2.1 and the Method in [9] in the senses of the iteration step (denoted as IT) and the total CPU time (denoted as CPU) in second. Moreover, parallel speed-up (denoted as SP) are listed in the following numerical tables. Here, the SP is defined to be the ratio of the parallel quasi-Chebyshev acceleration to the nonoverlapping multisplitting iterative method, i.e.,

$$\text{SP} = \frac{\text{CPU of the } \textbf{Method} \text{ in [9]}}{\text{CPU of the } \textbf{Method 2.1} \text{ (PQCA)}}.$$

In our implementations, all programs are performed in MATLAB with machine precision 10^{-32}. And all our tests are started from zero vector, the right-hand side $b = (1, \cdots, n)^T$ and the iteration is terminated once the current iterations $x^{(k,0)}$ obey

$$\frac{\|b - Ax^{(k,0)}\|}{\|b\|} < 10^{-6},$$

or the stopping criterion is not satisfied after 30000 iteration steps.

In our tests, the experimentally found optimal parameters $\alpha^{(i)},\ i=1,\cdots,m$ are obtained according to the Nelder-Mead simplex method (see [10]). When these optimal parameters are approximated to their true solutions under the given precision or the numbers of iteration steps are up to the maximum number of function evaluations allower (denoted as k_{opt}), we get the experimentally found optimal parameters $\alpha^{(i)},\ i=1,\cdots,m$.

Example 4.1. Consider the nine-point difference equations from discretizing the Possion equations, the resulted system of linear equations (1.1) are of the form

$$Ax=b, \tag{4.1}$$

where

$$A=\begin{bmatrix} D_s & B_s & & & \\ B_s & D_s & B_s & & \\ & \ddots & \ddots & \ddots & \\ & & B_s & D_s & B_s \\ & & & B_s & D_s \end{bmatrix}_{p\times p} \in R^{n\times n}, \quad n=ps,$$

with $D_s=tridiag(-4,20,-4)\in R^{s\times s}$, and $B_s=tridiag(-1,-4,-1)\in R^{s\times s}$.

Here and in the sequel, we use $\mu=[\frac{p}{3}]$ to denote the integer part of the number $\frac{p}{3}$ and $t=\mu s$.

We implement Method 2.1 with respect to different k_{opt} and the Method in [9] by the following three block splittings to solve the system of linear equations (4.1) in parallel processors, that is:

- (a) the first splitting is taken to be $N_1=M_1-A$ and $M_1=diag(M_{11}^{(1)},M_{22}^{(1)})\in R^{n\times n}$, where

$$M_{11}^{(1)}=\begin{bmatrix} D_s & & & \\ B_s & D_s & & \\ & \ddots & \ddots & \\ & & B_s & D_s \end{bmatrix}_{\mu\times\mu} \in R^{t\times t},$$
$$M_{22}^{(1)}=diag(D_s,D_s,\cdots,D_s)\in R^{(n-t)\times(n-t)};$$

- (b) the second splitting is taken to be $N_2=M_2-A$ and $M_2=diag(M_{11}^{(2)},M_{22}^{(2)},M_{33}^{(2)})\in R^{n\times n}$, where

$$M_{11}^{(2)}=diag(D_s,D_s,\cdots,D_s)\in R^{t\times t},$$
$$M_{22}^{(2)}=\begin{bmatrix} D_s & & & \\ B_s & D_s & & \\ & \ddots & \ddots & \\ & & B_s & D_s \end{bmatrix}_{\mu\times\mu} \in R^{t\times t},$$
$$M_{33}^{(2)}=diag(D_s,D_s,\cdots,D_s)\in R^{(n-2t)\times(n-2t)};$$

- (c) the third splitting is taken to be $N_3 = M_3 - A$ and $M_3 = diag(M_{11}^{(3)}, M_{22}^{(3)}) \in R^{n\times n}$, where

$$M_{11}^{(3)} = diag(D_s, D_s, \cdots, D_s) \in R^{2t\times 2t},$$

$$M_{22}^{(3)} = \begin{bmatrix} D_s & & & \\ B_s & D_s & & \\ & \ddots & \ddots & \\ & & B_s & D_s \end{bmatrix}_{\nu\times\nu} \in R^{(n-2t)\times(n-2t)},$$

with $\nu = p - 2\mu$.

In the following Table 4.1, we list the experimentally solved (4.1), IT,CPU and SP, with respect to different n.

Table 4.1 The comparisons of computational results for Example 4.1.

n		Method	Method 2.1			
		in [9]	k_{opt}=5	k_{opt}=10	k_{opt}=12	k_{opt}=15
3600	IT	2676	995	722	698	702
	CPU(s)	24.1675	16.5949	12.3783	12.1213	12.0575
	SP	1	1.5	2.0	2.0	2.0
6400	IT	4660	1741	1278	1223	1231
	CPU(s)	73.8825	49.8373	38.0831	37.0892	37.3659
	SP	1	1.5	1.9	2.0	2.0
10000	IT	7185	2697	1991	1937	1928
	CPU(s)	170.7108	119.4169	87.4981	88.9020	91.2372
	SP	1	1.4	2.0	1.9	1.9
14400	IT	10257	3878	2920	2856	2856
	CPU(s)	375.5195	242.1399	192.8085	194.1088	196.5161
	SP	1	1.6	1.9	1.9	1.9
19600	IT	13877	5272	3991	3949	3855
	CPU(s)	697.1244	462.2743	352.3912	357.6036	350.9744
	SP	1	1.5	2.0	1.9	2.0
25600	IT	18039	6885	5271	5182	5116
	CPU(s)	1587.5	1067.4	845.5334	847.6758	872.7531
	SP	1	1.5	1.9	1.9	1.8
32400	IT	22747	8700	6704	6623	6547
	CPU(s)	2759.9	1808.7	1420.8778	1451.4	1499.9
	SP	1	1.5	1.9	1.9	1.8
40000	IT	28004	10753	8317	8165	8084
	CPU(s)	4189.2	2283.1	2117.2	2094.8	2145.7
	SP	1	1.5	2.0	2.0	2.0

Example 4.2. Consider the generalized convection-diffusion equations in a two-dimensional case. The equation is

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + q \cdot \exp(x+y) \cdot x \cdot \frac{\partial u}{\partial x} + q \cdot \exp(x+y) \cdot y \cdot \frac{du}{dy} = f \qquad (4.2)$$

with the homogeneous Dirichlet boundary condition. We use the standard Ritz-Galerkin finite element method by P_1 conforming triangular element to approximate the following continuous solutions $u = x \cdot y \cdot (1-x) \cdot (1-y)$ in the domain $\Omega = [0,1] \times [0,1]$, the step-sizes along both x and y directions are the same, that is, $h = \frac{1}{2^m}$, $m = 5, 6, 7$. Let $q = 1$.

After discretization the matrix A of this equation is given by

$$A = \begin{bmatrix} A_{11} & B_{12} & & & \\ C_{21} & A_{22} & B_{23} & & \\ & \ddots & \ddots & \ddots & \\ & & C_{p-1,p-2} & A_{p-1,p-1} & B_{p-1,p} \\ & & & C_{p,p-1} & A_{p,p} \end{bmatrix} \in R^{n\times n},$$

where $A_{i,i},\ i = 1, \cdots, p$ are s-by-s nonsymmetric matrices and $B^T_{i,i+1} \neq C_{i+1,i}$, and $n = sp = 32^2, 64^2, 128^2$.

We implement Method 2.1 with respect to different k_{opt} and the Method in [9] by three splittings in the following, analogously to Example 4.1, to solve the system of linear equations (1.1) yielded in Example 4.2 in parallel processors, that is:

- (a) the first splitting is taken to be $N_1 = M_1 - A$ and $M_1 = diag(M^{(1)}_{11}, M^{(1)}_{22}) \in R^{n\times n}$, where

$$M^{(1)}_{11} = \begin{bmatrix} A_{11} & & & \\ C_{21} & A_{22} & & \\ & \ddots & \ddots & \\ & & C_{\mu,\mu-1} & A_{\mu,\mu} \end{bmatrix} \in R^{t\times t},$$

$$M^{(1)}_{22} = diag(A_{\mu+1,\mu+1}, A_{\mu+2,\mu+2}, \cdots, A_{p,p}) \in R^{(n-t)\times(n-t)};$$

- (b) the second splitting is taken to be $N_2 = M_2 - A$ and $M_2 = diag(M^{(2)}_{11}, M^{(2)}_{22}, M^{(2)}_{33}) \in R^{n\times n}$, where

$$M^{(2)}_{11} = diag(A_{11}, A_{22}, \cdots, A_{\mu,\mu}) \in R^{t\times t},$$

$$M^{(2)}_{22} = \begin{bmatrix} A_{\mu+1,\mu+1} & & & \\ C_{\mu+2,\mu+1} & A_{\mu+2,\mu+2} & & \\ & \ddots & \ddots & \\ & & C_{2\mu,\mu+1} & A_{2\mu,2\mu} \end{bmatrix} \in R^{t\times t},$$

$$M^{(2)}_{33} = diag(A_{2\mu+1,2\mu+1}, A_{2\mu+2,2\mu+2}, \cdots, A_{p,p}) \in R^{(n-2t)\times(n-2t)};$$

- (c) the third splitting is taken to be $N_3 = M_3 - A$ and $M_3 = diag(M^{(3)}_{11}, M^{(3)}_{22}) \in R^{n\times n}$, where

$$M^{(3)}_{11} = diag(A_{11}, A_{22}, \cdots, A_{2\mu,2\mu}) \in R^{2t\times 2t},$$

$$M^{(3)}_{22} = \begin{bmatrix} A_{2\mu+1,2\mu+1} & & & \\ C_{2\mu+2,2\mu+1} & A_{2\mu+2,2\mu+2} & & \\ & \ddots & \ddots & \\ & & C_{p,p-1} & A_{p,p} \end{bmatrix} \in R^{(n-2t)\times(n-2t)}.$$

In the following Table 4.2, we list the experimentally solved the linear system (1.1) yielded in Example 4.2, IT,CPU and SP, with respect to different n.

Table 4.2 The comparisons of computational results for Example 4.2.

n		Method in [9]	Method 2.1			
			k_{opt}=3	k_{opt}=5	k_{opt}=10	k_{opt}=15
32×32	IT	949	258	313	335	311
	CPU(s)	2.3971	1.1243	1.4091	1.7893	1.9108
	SP	1	2.1	1.7	1.3	1.3
64×64	IT	3510	961	1173	1258	1185
	CPU(s)	35.4218	17.8941	21.0246	23.6507	23.5787
	SP	1	2.0	1.7	1.5	1.5
128×128	IT	13597	4406	4533	4815	4556
	CPU(s)	673.8528	358.3477	371.7258	400.7144	392.9125
	SP	1	1.9	1.8	1.7	1.7

From Tables 4.1 and 4.2, the numerical results show that the Method 2.1 are feasible and efficient for solving the system of linear equations (1.1) on the multiprocessor systems. To our surprise, we know that the speed-up of the Method 2.1 is independent of the size of the linear systems. This shows that the Method 2.1 is much more stable.

Acknowledgments. The authors would like to thank the anonymous referees for their helpful comments and suggestions which greatly improved the original manuscript of this paper. This work is supported by NSF of China (11371275), NSF of Shanxi Province (2012011015-6) and STIP of Higher Education Institutions in Shanxi (2013144).

References

[1] Z.-Z. Bai, Parallel matrix multisplitting block relaxation iteration methods, *Math. Numer. Sinica*, **17** (1995), 238-252.

[2] Z.-Z. Bai and X.-B. Chi, Asymptotically optimal successive overrelaxation methods for systems of linear equations, *J. Comput. Math.*, **21**:5 (2003), 603-612.

[3] Z.-Z. Bai, Z.-F. Gao and T.-Z. Huang, Measure parameters of the effectiveness of the parallel iteration methods, *Math. Numer. Sinica*, **21** (1999), 325-330 (In Chinese).

[4] Z.-Z. Bai, J.-C. Sun and D.-R. Wang, A unified framework for the construction of various matrix multislpitting iterative methods for large sparse system of linear equations, *Comput. Math. Appl.*, **32** (1996), 51-76.

[5] A. Berman and R.J. Plemmons, Nonnegative Matrices in the Mathematical Science, Academic Press, New York, 1979.

[6] M.J. Castel, V. Migallón and J. Penadés, Convergence of non-stationary parallel multisplitting methods for hermitian positive definite matrices, *Math. Comput.*, **67** (1998), 209-220.

[7] A. Frommer and D.B. Szyld, H-splitting and two-stage iterative methods, *Numer. Math.*, **63** (1992), 345-356.

[8] A. Frommer and D.B. Syzld, Weighted max norms, splittings, and overlapping additive Schwarz iterations, *Numer. Math.*, **83** (1999), 259-278.

[9] V. Migallón, J. Penadés and D.B. Szyld, Nonstationary multisplittings with general weighting matrices, *SIAM J. Matrix Anal. Appl.*, **22** (2001), 1089-1094.

[10] J.A. Nelder and R. Mead, A simplex method for function minimization, *Comput. J.*, **7** (1965), 308-313.

[11] M. Neumann and R.J. Plemmons, Convergence of parallel multisplitting iterative methods for M-matrices, *Linear Algebra Appl.*, **88** (1987), 559-573.

[12] D.P. O'Leary and R.E. White, Multi-splittings of matrices and parallel solutions of linear systems, *SIAM J. Alg. Disc. Meth.*, **6** (1985), 630-640.

[13] C.-L. Wang and G.-Y. Meng, A practical two-term acceleration algorithm for linear systems, *Numer. Linear Algebra Appl.*, **19** (2012), 571-583.

[14] C.-L. Wang, G.-Y. Meng and X.-R. Yong, Modified parallel multisplitting iterative methods for non-Hermitian positive definite systems, *Adv. Comput. Math.*, **38** (2013), 859-872.

[15] R.-P. Wen, C.-L. Wang and X.-H. Yan, Generalizations of nonstationary multisplitting iterative method for symmetric positive definite linear systems, *Appl. Math. Comput.*, **216** (2010), 1707-1714.

[16] R.-P. Wen, G.-Y. Meng and C.-L. Wang, Quasi-Chebyshev accelerated iteration methods based on optimization for linear systems, *Comput. Math. Appl.*, **66** (2013), 934-942.

[17] R.E. White, Multisplitting with different weighting schemes, *SIAM J. Matrix Anal. Appl.*, **10** (1989), 481-493.