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

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

$$e_{j}^{(i)} = \begin{cases} e_{j}^{(i)} > 0, & \text{for } j \in \mathcal{N}_i, \\ 0, & \text{for } j \not\in \mathcal{N}_i, \end{cases} \quad j = 1, \cdots, n.$$  

* 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

$$Ax = b$$  \hfill (1.1)

is

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

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

In particular, the above method is called a nonoverlapping multisplitting iterative method if $\mathcal{N}_i \cap \mathcal{N}_2 = \emptyset$ ($i_1 \neq i_2$).

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 $E_i^{(k)}, i = 1, \cdots, m; \quad k = 1, \cdots, m$ 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 $\omega$ 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 $\omega$ 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 \( x^T A x > 0 \). 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

\[ |a_{jj}| > \sum_{i=1, i \neq j}^n |a_{ij}|, \quad j = 1, \ldots, n. \]

**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:

\[ A = M_i - N_i, \quad i = 1, \ldots, m; \tag{2.1} \]

\[ E_i = diag(e^{(i)}_1, \ldots, e^{(i)}_n), \tag{2.2} \]

with

\[ e^{(i)}_j = \begin{cases} 1, & \text{for } j \in N_i, \\ 0, & \text{for } j \notin N_i, \end{cases} \quad j = 1, \ldots, n. \]

Assume that the cardinal numbers of those sets \( N_i, \quad i = 1, \ldots, m \) are \( n_i \), we then have \( n_1 + \cdots + n_m = n \). By introducing

\[ T_i = M_i^{-1}N_i, \quad i = 1, \ldots, m, \]

the iteration matrix of \( k \)th step

\[ T_k = \sum_{i=1}^m E_i M_i^{-1} N_i = \sum_{i=1}^m E_i T_i. \tag{2.3} \]
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 $(x^{(k,0)})^\infty_k$ convergence,

- Step 1. For $t = 1, \cdots, l$, compute in parallel the vectors $x^{(k,t)} = (x^{(k,t)}_1, \cdots, x^{(k,t)}_{\ell})^T \in R^n_x$ by
  \[ M_i x^{(k,t)}_i = N_i x^{(k,t-1)}_i + b, \quad i = 1, \cdots, \ell \]
  in $i$th processor. Here, $x^{(k,t)} = (x^{(k,t)}_1, \cdots, x^{(k,t)}_{\ell})^T \in R^n_x$.

- Step 2. For $i = 1, \cdots, \ell$, let
  \[ x^{(i)} = (x^{(i)}_n), (\alpha^{(i)}(x^{(i)}_n) - x^{(i-2)}_n), (\alpha^{(i)}(x^{(i)}_n) - x^{(i-2)}_n))^T \in R^n_x. \]

For $i = 1, \cdots, \ell$, compute in parallel the approximate solutions of $\alpha^{(i)}$ by the following optimization problems:

(a) when $A$ is a symmetric positive definite matrix

\[
\min_{\alpha^{(i)}} \frac{1}{2} (x^{(i)})^T A x^{(i)} - (x^{(i)})^T b, \quad i = 1, \cdots, \ell; \tag{2.4}
\]

(b) when $A$ is an $H$-matrix

\[
\min_{\alpha^{(i)}} ||Ax^{(i)} - b||_1, \quad i = 1, \cdots, \ell. \tag{2.5}
\]

We denote the solutions of (2.4) (or (2.5)) by $\bar{\alpha}^{(i)}$ and then

\[
\bar{x}_n = (\bar{\alpha}^{(i)}(x^{(i)}_n) - x^{(i-2)}_n), (\bar{\alpha}^{(i)}(x^{(i)}_n) - x^{(i-2)}_n))^T \in R^n_x, \quad i = 1, \cdots, \ell.
\]

- Step 3. Let $x^{(k+1,0)} = \arg \min_x \frac{1}{2} x^T A x - x^T b$ or $x^{(k+1,0)} = \arg \min_x ||Ax - b||_1$ with $x \in \{\bar{x}, x^{(k,0)}\}$, where
  \[ x^{(k,0)} = ((x^{(i)}_n), \cdots, (x^{(i)}_n))^T \in R^n_x, \]
  \[ \bar{x} = ((\bar{x}_n), (\bar{x}_n), \cdots, (\bar{x}_n))^T \in R^n_x. \]

- Step 4. If $||Ax^{(k+1,0)} - b|| \leq \epsilon$, 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:

\[
\bar{\alpha}^{(i)} = ((X^{(i)}_k)^T A_{i,i} X^{(i)}_k)^{-1} (X^{(i)}_k)^T (b_n - \sum_{j \neq i} A_{i,j} X^{(j)}_k), \quad i = 1, \cdots, \ell, \tag{2.6}
\]

where $X^{(i)}_k = (x^{(k,l)}_n - x^{(k,l-2)}_n) \in R^n_x$.

To avoid the tedious computation of $((X^{(i)}_k)^T A_{i,i} X^{(i)}_k)^{-1}$, however, we can use the inexact line search to find the approximations of $\bar{\alpha}^{(i)}$, $i = 1, \cdots, \ell$. 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 \( l[2n_{i} + (n_{i} - 1)^2] + 2n_{i} + 2n_{i} - 1 \) and that of the Method in [9] is \( l[2n_{i} + (n_{i} - 1)^2] \), then the cost of Method 2.1 is about
\[
1 + \frac{2n_{i} + 2n_{i} - 1}{l[2n_{i} + (n_{i} - 1)^2]}
\]
times of that of the Method in [9]. Particularly, the times is about
\[
1 + \frac{2n_{i}^2 + 2nm - m^2}{l[(2m + 1)n^2 - 2mn + m^2]} \approx 1 + \frac{2m}{(2m + 1)l'}
\]
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
\[
S_m = \frac{W_s \times N_s}{W_p \times N_p},
\]
where \( W_p = \max_{1 \leq i \leq m} \{ w_i \} \) 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,
\[
\frac{S_m}{S_m'} = \frac{N_p[l(2mn_{i} + (n_{i} - 1)^2) + 2mn_{i} + 2n_{i} - 1]}{N_p[l(2mn_{i} + (n_{i} - 1)^2)]}, \quad n_{i} = \max_{1 \leq i \leq m} \{ n_{i} \}
\]
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
\[
\frac{S_m}{S_m'} \approx \frac{[(2m + 1)l + 2m]N_p'}{[(2m + 1)l]N_p}.
\]

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
\[
\| A^{\frac{1}{2}}TA^{-\frac{1}{2}} \|_2 \leq r < 1.
\]

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 \( E_i = \text{diag}(0, \cdots, 0, I_i, 0, \cdots, 0) \) 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
\[
l > -\frac{1}{\ln r} \ln \left( \sum_{i=1}^{m} \| A^{\frac{1}{2}}E_iA^{-\frac{1}{2}} \|_2 \right),
\]
where \( r = \max_{1 \leq i \leq m} r_i \), 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
\[ T = P^{-1}Q = \sum_{i=1}^{m} E_i(M_i^{-1}N_i)^l. \] (3.1)

Similar to the proof of Theorem 2.1 in [9], we have
\[
\|A^{\frac{1}{2}}TA^{-\frac{1}{2}}\|_2 \leq \max_i \|A^{\frac{1}{2}}M_i^{-1}N_iA^{-\frac{1}{2}}\|_2 \sum_{i=1}^{m} \|A^{\frac{1}{2}}E_iA^{-\frac{1}{2}}\|_2 \\
\leq \sum_{i=1}^{m} \|A^{\frac{1}{2}}E_iA^{-\frac{1}{2}}\|_2 \max_i r_i^l < 1.
\]

Thus, Lemma 3.1 implies that \( A = P - Q \) is a \( P \)-regular splitting. \( \Box \)

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 nonoverlapping multisplitting iterative method. In fact, this result can straightforwardly employ the nonstationary iteration number \( s(i, k) \), introduced in [9], to obtain an analog. 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 \( e^{(k,0)} = x^{(k,0)} - x_* \). Then it holds that
\[
e^{(k,l)} = T e^{(k,0)},
\] (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 \( k \)th iteration,
\[
\min_{\alpha(i)} \frac{1}{2}(x^{(i)} - x_*)^TA(x^{(i)} - x_*), \quad i = 1, \ldots, m.
\]

Therefore by noting that Method 2.1, we have
\[
(e^{(k+1,0)})^TAe^{(k+1,0)} \leq (e^{(k,l)})^TAe^{(k,l)}. \quad (3.3)
\]
It follows from (3.2) and (3.3) that
\[
\|A^{\frac{1}{2}} e^{(k+1,0)}\|_2 \leq \|A^{\frac{1}{2}} e^{(k,l)}\|_2 \leq \|A^{\frac{1}{2}}Te^{(k,0)}\|_2 \\
= \|A^{\frac{1}{2}}TA^{-\frac{1}{2}}A^{\frac{1}{2}} e^{(k,0)}\|_2 \leq \|(A^{\frac{1}{2}}TA^{-\frac{1}{2}})\|_2 \|A^{\frac{1}{2}} e^{(k,0)}\|_2 \\
\leq \cdots \\
\leq \|A^{\frac{1}{2}}TA^{-\frac{1}{2}}\|_2 \|A^{\frac{1}{2}} e^{(0,0)}\|_2.
\]

By noting Lemmas 3.1 and 3.2, there exists a positive real number \( \theta \) such that
\[
\|A^{\frac{1}{2}}TA^{-\frac{1}{2}}\|_2 \leq \theta < 1.
\]
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 \( \epsilon = (1, \cdots, 1)^T \). From Property 1.4, it holds that

\[
\| NM^{-1} \|_1 \leq \| \langle N \rangle |\langle M \rangle^{-1} \|_1 = \max_{1 \leq i \leq n} |\epsilon^T |\langle N \rangle |(\langle M \rangle^{-1})_i|.
\]

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

\[
x^T |\langle N \rangle | = x^T(\langle M \rangle). \tag{3.5}
\]

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

\[
\sum_{j=1}^{n} |n_{j,i_0}| = m_{i_0,i_0} x_{i_0} - \sum_{j \neq i_0} m_{j,i_0} x_j \geq \left( m_{i_0,i_0} - \sum_{j \neq i_0} m_{j,i_0} \right) x_{i_0},
\]

which yields

\[
x_{i_0} \leq \frac{\sum_{j=1}^{n} |n_{j,i_0}|}{m_{i_0,i_0} - \sum_{j \neq i_0} m_{j,i_0}}. \tag{3.6}
\]

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 $\|N M^{-1}\|_1 < 1$. Hence,

$$\|D N M^{-1} D^{-1}\|_1 = \|N M^{-1}\|_1 < 1,$$  \hfill (3.9)

which completed the proof.

\begin{proof}
Let $A = M_i - N_i$, $i = 1, \cdots, m$ be $m$ splittings of the $H$-matrix $A$ and $E_i = \text{diag}(0, \cdots, 0, I_i, 0, \cdots, 0)$ 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 $T = \sum_{i=1}^{m} E_i (M_i^{-1} N_i) = P^{-1} Q$. 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

$$\|DA e^{(k+1,0)}\|_1 \leq \|DA e^{(k,0)}\|_1 = \|DAT e^{(k,0)}\|_1 = \|DAT A^{-1} D^{-1} D \cdot A e^{(k,0)}\|_1 \leq \|DAT A^{-1} D^{-1}\|_1 \|DA e^{(k,0)}\|_1 = \|D(QP^{-1}) D^{-1}\|_1 \|DA e^{(k,0)}\|_1 \leq r \|DA e^{(k,0)}\|_1 \leq \cdots \leq r^{k+1} \|DA e^{(0,0)}\|_1,$$  \hfill (3.10)

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

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

We have completed the proof of this theorem. \hfill \Box

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.,

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

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

$$\frac{\|b - A x^{(k,0)}\|}{\|b\|} < 10^{-6}.$$
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 Poisson equations, the resulted system of linear equations (1.1) are of the form

\[
Ax = b, \tag{4.1}
\]

where

\[
A = \begin{bmatrix}
D_s & B_s & & \\
B_s & D_s & B_s & \\
& \ddots & \ddots & \ddots \\
B_s & D_s & B_s & \\
& & B_s & D_s
\end{bmatrix} \in \mathbb{R}^{n \times n}, \quad n = ps,
\]

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

Here and in the sequel, we use \( \mu = \lfloor \frac{p}{3} \rfloor \) 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 \( M_1 = \text{diag}(M_{11}^{(1)}, M_{22}^{(1)}) \in \mathbb{R}^{n \times n}, \) where

  \[
  M_{11}^{(1)} = \begin{bmatrix}
  D_s & B_s & & \\
  B_s & D_s & B_s & \\
  & \ddots & \ddots & \ddots \\
  B_s & D_s & B_s & \\
  & & B_s & D_s
  \end{bmatrix} \mu \times \mu \in \mathbb{R}^{t \times t},
  \]

  \[
  M_{22}^{(1)} = \text{diag}(D_s, D_s, \cdots, D_s) \in \mathbb{R}^{(n-t) \times (n-t)};
  \]

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

  \[
  M_{11}^{(2)} = \text{diag}(D_s, D_s, \cdots, D_s) \in \mathbb{R}^{t \times t},
  \]

  \[
  M_{22}^{(2)} = \begin{bmatrix}
  D_s & B_s & \\
  B_s & D_s & \\
  & \ddots & \ddots \\
  & B_s & D_s
  \end{bmatrix} \mu \times \mu \in \mathbb{R}^{t \times t},
  \]

  \[
  M_{33}^{(2)} = \text{diag}(D_s, D_s, \cdots, D_s) \in \mathbb{R}^{(n-2t) \times (n-2t)};
  \]
• (c) the third splitting is taken to be $N_3 = M_3 = A$ and $M_3 = \text{diag}(M_{11}^{(3)}, M_{22}^{(3)}) \in R^{n \times n}$,
where

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

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

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>
<thead>
<tr>
<th>$n$</th>
<th>Method in [9]</th>
<th>Method 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_{opt}=5$</td>
<td>$k_{opt}=10$</td>
</tr>
<tr>
<td>3600</td>
<td>IT 2676</td>
<td>995</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 24.1675</td>
<td>16.5949</td>
</tr>
<tr>
<td></td>
<td>SP 1.5</td>
<td>2.0</td>
</tr>
<tr>
<td>6400</td>
<td>IT 4660</td>
<td>1741</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 73.8825</td>
<td>49.8373</td>
</tr>
<tr>
<td></td>
<td>SP 1.5</td>
<td>1.9</td>
</tr>
<tr>
<td>10000</td>
<td>IT 7185</td>
<td>2697</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 170.7108</td>
<td>119.4169</td>
</tr>
<tr>
<td></td>
<td>SP 1.4</td>
<td>2.0</td>
</tr>
<tr>
<td>14400</td>
<td>IT 10257</td>
<td>3878</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 375.5195</td>
<td>242.1399</td>
</tr>
<tr>
<td></td>
<td>SP 1.6</td>
<td>1.9</td>
</tr>
<tr>
<td>19600</td>
<td>IT 13877</td>
<td>5272</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 697.1244</td>
<td>462.2743</td>
</tr>
<tr>
<td></td>
<td>SP 1.5</td>
<td>2.0</td>
</tr>
<tr>
<td>25600</td>
<td>IT 18039</td>
<td>6885</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 1587.5</td>
<td>1067.4</td>
</tr>
<tr>
<td></td>
<td>SP 1.5</td>
<td>1.9</td>
</tr>
<tr>
<td>32400</td>
<td>IT 22747</td>
<td>8700</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 2759.9</td>
<td>1808.7</td>
</tr>
<tr>
<td></td>
<td>SP 1.5</td>
<td>1.9</td>
</tr>
<tr>
<td>40000</td>
<td>IT 28004</td>
<td>10753</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 4189.2</td>
<td>2283.1</td>
</tr>
</tbody>
</table>

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

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

(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, $h = \frac{1}{2m}$, $m = 5, 6, 7$. Let $q = 1$.

After discretization the matrix $A$ of this equation is given by

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

where $A_{i,i}$, $i = 1, \ldots, p$ are $s$-by-$s$ nonsymmetric matrices and $B_{i,i+1}^T \neq C_{i+1,i}$, 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 $M_1 = \text{diag}(M_{11}^{(1)}, M_{22}^{(1)}) \in R^{n \times n}$, where

  $$M_{11}^{(1)} = \begin{bmatrix} A_{11} \\ C_{21} & A_{22} \\ \vdots & \ddots & \ddots \\ C_{p-1+1,p-1} & A_{p-1,p-1} & B_{p-1,p} \end{bmatrix} \in R^{t \times t}, \quad M_{22}^{(1)} = \text{diag}(A_{p+1}, A_{p+2}, \ldots, A_{p,p}) \in R^{(n-t) \times (n-t)};$$

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

  $$M_{11}^{(2)} = \text{diag}(A_{11}, A_{22}, \ldots, A_{p,p}) \in R^{t \times t}, \quad M_{22}^{(2)} = \begin{bmatrix} A_{p+1,1} \\ C_{p+1,2} & A_{p+2,2} \\ \vdots & \ddots & \ddots \\ C_{p,p+1} & A_{p,p} \end{bmatrix} \in R^{t \times t}, \quad M_{33}^{(2)} = \text{diag}(A_{2p+1}, A_{2p+2},\ldots, A_{p,p}) \in R^{(n-2t) \times (n-2t)};$$

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

  $$M_{11}^{(3)} = \text{diag}(A_{11}, A_{22}, \ldots, A_{2p,2p}) \in R^{2t \times 2t}, \quad M_{22}^{(3)} = \begin{bmatrix} A_{2p+1,1} \\ C_{2p+1,2} & A_{2p+2,2} \\ \vdots & \ddots & \ddots \\ C_{p,p-1} & A_{p,p} \end{bmatrix} \in R^{(n-2t) \times (n-2t)}.$$

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.

<table>
<thead>
<tr>
<th>n</th>
<th>Method in [9]</th>
<th>Method 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(k_{opt}=3)</td>
<td>(k_{opt}=5)</td>
</tr>
<tr>
<td>32×32</td>
<td>IT 949</td>
<td>258</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 2.3971</td>
<td>1.1243</td>
</tr>
<tr>
<td>64×64</td>
<td>IT 3510</td>
<td>961</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 35.4218</td>
<td>17.8941</td>
</tr>
<tr>
<td>128×128</td>
<td>IT 13597</td>
<td>4406</td>
</tr>
<tr>
<td></td>
<td>CPU(s) 673.8528</td>
<td>358.3477</td>
</tr>
</tbody>
</table>

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


