Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational and Applied Mathematics 180 (2005) 245–263

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Convergence of nonstationary multisplitting methods using ILU factorizations

Jae Heon Yun*

Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju, 48 Gaeshin-dong, Heungduk-Gu, Chungbuk 361-763, Republic of Korea

Received 13 January 2004; received in revised form 26 October 2004

Abstract

In this paper, we first study convergence of nonstationary multisplitting methods associated with a multisplitting which is obtained from the ILU factorizations for solving a linear system whose coefficient matrix is a large sparse *H*-matrix. We next study a parallel implementation of the *relaxed nonstationary two-stage multisplitting method* (called Algorithm 2 in this paper) using ILU factorizations as inner splittings and an application of Algorithm 2 to parallel preconditioner of Krylov subspace methods. Lastly, we provide parallel performance results of both Algorithm 2 using ILU factorizations as inner splittings and the BiCGSTAB with a parallel preconditioner which is derived from Algorithm 2 on the IBM p690 supercomputer.

MSC: 65F10; 65F15

Keywords: Nonstationary multisplitting method; ILU factorization; *H*-matrix; Multisplitting; Krylov subspace method

1. Introduction

In this paper, we consider parallel nonstationary multisplitting methods for solving a linear system of the form

$$Ax = b, \quad x, b \in \mathbb{R}^n, \tag{1}$$

* Tel.: 82 4326 12 250; fax: +82 4327 49 619.
E-mail address: gmjae@chungbuk.ac.kr (J.H. Yun).

0377-0427/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2004.10.015

where $A \in \mathbb{R}^{n\times n}$ is a large sparse H-matrix. Multisplitting method was introduced by O'Leary and White [14] and was further studied by many authors [6,13,19,23]. The multisplitting method can be thought of as an extension and parallel generalization of the classical block Jacobi method [3].

A representation $A = M - N$ is called a *splitting* of A when M is nonsingular. A splitting $A = M - N$ is called *regular* if $M^{-1} \geqslant 0$ and $N \geqslant 0$, and it is called *weak regular* if $M^{-1} \geqslant 0$ and $M^{-1}N \geqslant 0$ [1]. A collection of triples (M_k, N_k, E_k), $k = 1, 2, \dots, \ell$, is called a *multisplitting* of A if $A = M_k - N_k$ is a splitting of A for $k = 1, 2, \dots, \ell$, and E_k's, called weighting matrices, are nonnegative diagonal matrices such that $\sum_{k=1}^{\ell} E_k = I$. The *relaxed nonstationary multisplitting method* associated with this multisplitting and a positive relaxation parameter ω for solving a linear system $Ax = b$ is as follows.

Algorithm 1. Relaxed nonstationary multisplitting method

```
Given an initial vector $x_0$
For $i = 1, 2, \dots,$ until convergence
    For $k = 1$ to $\ell$
        $y_{k,0} = x_{i-1}$
        For $j = 1$ to $s(k, i)$
            $M_k y_{k,j} = N_k y_{k,j-1} + b$
    $x_i = \omega \sum_{k=1}^{\ell} E_k y_{k,s(k,i)} + (1 - \omega) x_{i-1}.$
```

Notice that Algorithm 1 with $\omega = 1$ is called the *nonstationary multisplitting method*. Mas et al. [10] showed the convergence of Algorithm 1 under certain conditions when A is an H-matrix. When (M_k, N_k, E_k), $k = 1, 2, \dots, \ell$, is a multisplitting of A and $M_k = B_k - C_k$ is a splitting of M_k for each k, the *relaxed nonstationary two-stage multisplitting method* with a positive relaxation parameter ω for solving a linear system $Ax = b$ is as follows.

Algorithm 2. Relaxed nonstationary two-stage multisplitting method

```
Given an initial vector $x_0$
For $i = 1, 2, \dots,$ until convergence
    For $k = 1$ to $\ell$
        $y_{k,0} = x_{i-1}$
        For $j = 1$ to $s(k, i)$
            $y_{k,j} = \omega B_k^{-1}(C_k y_{k,j-1} + N_k x_{i-1} + b) + (1 - \omega) y_{k,j-1}$
    $x_i = \sum_{k=1}^{\ell} E_k y_{k,s(k,i)}.$
```

In Algorithm 2, the splittings $A = M_k - N_k$ are called outer splittings and the splittings $M_k = B_k - C_k$ are called inner splittings. Bru et al. [3] showed the convergence of Algorithm 2 when A is a monotone matrix (i.e., $A^{-1} \geqslant 0$) or A is an H-matrix. If $\omega = 1$ in Algorithm 2, then Algorithm 2 reduces to the *nonstationary two-stage multisplitting method*. Notice that the loop k of Algorithms 1 and 2 can be executed completely in parallel by different processors. Also notice that the number of inner iterations $s(k, i)$ in Algorithms 1 and 2 depends on the iteration i and the splitting $A = M_k - N_k$. Throughout the paper, it is assumed that

$s(k,i) \geqslant 1$ for every k and i. If $s(k,i)=1$ for all k and i in Algorithm 1, then Algorithm 1 is called the *relaxed multisplitting method*.

For a large sparse matrix A, a convenient way of obtaining a multisplitting of A is to use the ILU factorizations of A which were first introduced by Varga [21] and studied by many authors [4,9,11,12]. One advantage of multisplitting methods associated with a multisplitting which is obtained from the ILU factorizations is that linear systems required for each iteration of multisplitting methods can be cheaply solved by using the forward and backward substitutions since many fill-in elements are dropped during the ILU factorization process. So, it is worth studying the convergence of multisplitting methods using the ILU factorizations. This paper is organized as follows. In Section 2, we present some notation, definitions and preliminary results which we refer to later. In Section 3, we present convergence results of the relaxed nonstationary multisplitting method (Algorithm 1) and the relaxed nonstationary two-stage multisplitting method (Algorithm 2) using ILU factorizations for solving the linear system (1). In Section 4, we study a parallel implementation of Algorithm 2 using ILU factorizations as inner splittings and an application of Algorithm 2 to parallel preconditioner of Krylov subspace methods such as the CGS [18], GMRES [17] and Bi-CGSTAB [20]. In Section 5, we provide parallel performance results of both Algorithm 2 using ILU factorizations as inner splittings and the BiCGSTAB with a parallel preconditioner which is derived from Algorithm 2 on the IBM p690 supercomputer. Lastly, some concluding remarks are drawn.

2. Preliminaries

For a vector $x \in \mathbb{R}^n$, $x \geqslant 0$ $(x > 0)$ denotes that all components of x are nonnegative (positive). For two vectors $x,\ y \in \mathbb{R}^n$, $x \geqslant y$ $(x > y)$ means that $x-y \geqslant 0$ $(x-y>0)$. For a vector $x \in \mathbb{R}^n$, $|x|$ denotes the vector whose components are the absolute values of the corresponding components of x. These definitions carry immediately over to matrices. It follows that $|A| \geqslant 0$ for any matrix A and $|AB| \leqslant |A||B|$ for any two matrices A and B of compatible size. For a square matrix B, $\operatorname{diag}(B)$ denotes a diagonal matrix whose diagonal part coincides with the diagonal part of B.

A matrix $A=(a_{ij}) \in \mathbb{R}^{n\times n}$ is called an *M-matrix* if $a_{ij} \leqslant 0$ for $i \neq j$ and $A^{-1} \geqslant 0$. The *comparison matrix* $\langle A\rangle = (\alpha_{ij})$ of a matrix $A=(a_{ij})$ is defined by

$$\alpha_{ij} = \begin{cases} |a_{ij}| & \text{if } i=j, \\ -|a_{ij}| & \text{if } i \neq j. \end{cases}$$

A matrix A is called an *H-matrix* if $\langle A\rangle$ is an M-matrix. Note that M-matrices and strictly or irreducibly diagonally dominant matrices are contained in the class of all H-matrices. A splitting $A=M-N$ is called an *H-compatible splitting* if $\langle A\rangle = \langle M\rangle - |N|$. It was shown in [7] that if A is an H-matrix and $A=M-N$ is an H-compatible splitting, then M is also an H-matrix. Let $\rho(A)$ denote the *spectral radius* of a square matrix A. Varga [22] showed that for any square matrices A and B, $|A| \leqslant B$ implies $\rho(A) \leqslant \rho(B)$.

Lemma 2.1 (*Frommer and Mayer [6]*). *Let $A=D-B$ be an H-matrix with $D=\operatorname{diag}(A)$. Then*

(a) *A and $|D|$ are nonsingular and* $\rho(|D|^{-1}|B|) < 1$.
(b) $|A^{-1}| \leqslant \langle A\rangle^{-1}$.

Lemma 2.2 (*Bru and Fuster* [2]). *Let T_i, $i=1,2,\ldots$, be a sequence of square matrices. If there exists a matrix norm $\|\cdot\|$ and a $\theta<1$ such that $\|T_i\|\leqslant\theta$ for all $i=1,2,\ldots$, then*

$$\lim_{i\to\infty} T_i T_{i-1}\cdots T_1 = 0.$$

For a vector $v>0$, the *weighted max norm* $\|x\|_v$ is defined by

$$\|x\|_v = \inf\{\beta>0: -\beta v\leqslant x\leqslant \beta v\}.$$

For a matrix B, $\|B\|_v$ denotes the matrix norm of B corresponding to the weighted max norm defined above. It is well-known that $\|B\|_v = \||B|v\|_v$ and $|x|\leqslant|y|$ implies $\|x\|_v\leqslant\|y\|_v$.

A general algorithm for building ILU factorization can be derived by performing Gaussian elimination and dropping some of the elements in predetermined off-diagonal positions. Let S_n denote the set of all pairs of indices of off-diagonal matrix entries, i.e.

$$S_n = \{(i,j)\mid i\neq j, 1\leqslant i\leqslant n, 1\leqslant j\leqslant n\}.$$

The following theorem shows the existence of the ILU factorization for an H-matrix A.

Theorem 2.3 (*Messaoudi* [12]). *Let A be an $n\times n$ H-matrix. Then, for every zero pattern set $Q\subset S_n$, there exist a unit lower triangular matrix $L=(l_{ij})$, an upper triangular matrix $U=(u_{ij})$, and a matrix $N=(n_{ij})$, with $l_{ij}=u_{ij}=0$ if $(i,j)\in Q$ and $n_{ij}=0$ if $(i,j)\notin Q$, such that $A=LU-N$. Moreover, the factors L and U are also H-matrices.*

In Theorem 2.3, $A=LU-N$ is called an *ILU factorization* of A corresponding to a zero pattern set $Q\subset S_n$. In particular, if Q is an empty set, then $N=0$ and thus a complete LU factorization of A such that A = LU is obtained. When A is an M-matrix, it was shown in [11] that the ILU factorization $A=LU-N$ in Theorem 2.3 is a regular splitting of A and L and U are also M-matrices. The following theorem shows the relations between the ILU factorizations of an H-matrix A and $\langle A\rangle$.

Theorem 2.4 (*Kim and Yun* [8], *Messaoudi* [12]). *Let A be an $n\times n$ H-matrix. Let $A=LU-N$ and $\langle A\rangle = \tilde{L}\tilde{U}-\tilde{N}$ be the ILU factorizations of A and $\langle A\rangle$ corresponding to a zero pattern set $Q\subset S_n$, respectively. Then each of the following holds*:

$$\text{(a) } |L^{-1}|\leqslant\tilde{L}^{-1},\quad \text{(b) } |U^{-1}|\leqslant\tilde{U}^{-1},\quad \text{(c) } |N|\leqslant\tilde{N},\quad \text{(d) } |(LU)^{-1}N|\leqslant(\tilde{L}\tilde{U})^{-1}\tilde{N}.$$

In Theorem 2.4, it is easy to show that LU is not an H-matrix and $\tilde{L}\tilde{U}$ is not an M-matrix even if L and U are H-matrices and $\tilde{L}$ and $\tilde{U}$ are M-matrices.

3. Convergence of nonstationary multisplitting methods

In this section, we present convergence results of nonstationary multisplitting methods associated with a multisplitting which is obtained from the ILU factorizations for solving linear system (1). First, we consider convergence of the relaxed nonstationary multisplitting method (Algorithm 1) using ILU factorizations. Algorithm 1 can be written as

$$x_i = H_{\omega,i}x_{i-1} + P_{\omega,i}b,\quad i=1,2,\ldots, \tag{2}$$

where

$$H_{\omega,i} = \omega \sum_{k=1}^{\ell} E_k (M_k^{-1} N_k)^{s(k,i)} + (1-\omega) I, \quad i = 1, 2, \ldots$$

and

$$P_{\omega,i} = \omega \sum_{k=1}^{\ell} E_k \left(\sum_{j=0}^{s(k,i)-1} (M_k^{-1} N_k)^j \right) M_k^{-1}, \quad i = 1, 2, \ldots .$$

The $H_{\omega,i}$'s are called iteration matrices for Algorithm 1. Then, it is easy to show that $P_{\omega,i} A = I - H_{\omega,i}$ for each i. Hence, the exact solution ξ of $Ax = b$ satisfies

$$\xi = H_{\omega,i} \xi + P_{\omega,i} b, \quad i = 1, 2, \ldots . \tag{3}$$

From (2) and (3), the error vector $e_i = x_i - \xi$ satisfies

$$e_i = H_{\omega,i} e_{i-1} = H_{\omega,i} H_{\omega,i-1} \cdots H_{\omega,1} e_0, \quad i = 1, 2, \ldots . \tag{4}$$

From (4), the sequence of vectors generated by the iteration (2) converges to the exact solution of $Ax = b$ for any initial vector x_0 if and only if

$$\lim_{i \to \infty} H_{\omega,i} H_{\omega,i-1} \cdots H_{\omega,1} = 0. \tag{5}$$

Theorem 3.1. *Let $A = D - B$ be an $n \times n$ H-matrix with $D = \mathrm{diag}(A)$. Let $J = |D|^{-1}|B|$ and let $Q_1, Q_2, \ldots, Q_\ell$ be zero pattern sets which are subsets of S_n. For each $1 \leqslant k \leqslant \ell$, let $A = L_k U_k - N_k$ be the ILU factorization of A corresponding to Q_k. Then, the relaxed nonstationary multisplitting method associated with the multisplitting $(L_k U_k, N_k, E_k)$, $k = 1, 2, \ldots, \ell$, converges to the exact solution of $Ax = b$ for any initial vector x_0 if $0 < \omega < \frac{2}{1+\rho}$, where $\rho = \rho(J)$.*

Proof. From Lemma 2.2, it suffices to show that there exists a matrix norm $\|\cdot\|$ and a $\theta < 1$ such that $\|H_{\omega,i}\| \leqslant \theta$ for all $i = 1, 2, \ldots$. Since $A = D - B$ and $D = \mathrm{diag}(A)$,

$$\langle A \rangle = |D| - |B| = |D|(I - J). \tag{6}$$

For each $1 \leqslant k \leqslant \ell$, let $\langle A \rangle = \tilde{L}_k \tilde{U}_k - \tilde{N}_k$ be the ILU factorization of $\langle A \rangle$ corresponding to Q_k. By some manipulation, it can be shown that $|D^{-1}| \leqslant (\tilde{L}_k \tilde{U}_k)^{-1}$ for all $k = 1, 2, \ldots, \ell$. It follows that for all $k = 1, 2, \ldots, \ell$

$$I \leqslant (\tilde{L}_k \tilde{U}_k)^{-1} |D|. \tag{7}$$

Using Theorem 2.4, one obtains

$$\begin{aligned} |H_{\omega,i}| &= \left| \omega \sum_{k=1}^{\ell} E_k ((L_k U_k)^{-1} N_k)^{s(k,i)} + (1-\omega) I \right| \\ &\leqslant \omega \sum_{k=1}^{\ell} E_k ((\tilde{L}_k \tilde{U}_k)^{-1} \tilde{N}_k)^{s(k,i)} + |1-\omega| I. \end{aligned} \tag{8}$$

We now use the argument presented in [10]. Let $e=(1,1,\ldots,1)^{\mathrm{T}}$. Since $J \geqslant 0$, $J+\varepsilon ee^{\mathrm{T}}>0$ for any $\varepsilon>0$ and thus there exists a Perron vector $x_\varepsilon>0$ such that

$$(J+\varepsilon ee^{\mathrm{T}})x_\varepsilon=\rho_\varepsilon x_\varepsilon, \tag{9}$$

where $\rho_\varepsilon=\rho(J+\varepsilon ee^{\mathrm{T}})$. Since $\rho<1$ from Lemma 2.1 and $0<\omega<\frac{2}{1+\rho}$ from the assumption, it can be easily shown that $|1-\omega|+\omega\rho<1$. By continuity of the spectral radius, there exists an ε_0 such that for all $0<\varepsilon\leqslant\varepsilon_0$

$$\rho_\varepsilon<1 \quad \text{and} \quad |1-\omega|+\omega\rho_\varepsilon<1.$$

Now, choose an ε such that $0<\varepsilon\leqslant\varepsilon_0$. Then, from (6), (7) and (9), one obtains

$$\begin{aligned}(\tilde{L}_k\tilde{U}_k)^{-1}\tilde{N}_k x_\varepsilon &= (I-(\tilde{L}_k\tilde{U}_k)^{-1}|D|(I-J))x_\varepsilon\\ &\leqslant (I-(\tilde{L}_k\tilde{U}_k)^{-1}|D|(I-(J+\varepsilon ee^{\mathrm{T}})))x_\varepsilon\\ &= x_\varepsilon-(1-\rho_\varepsilon)(\tilde{L}_k\tilde{U}_k)^{-1}|D|x_\varepsilon\\ &\leqslant x_\varepsilon-(1-\rho_\varepsilon)x_\varepsilon=\rho_\varepsilon x_\varepsilon.\end{aligned} \tag{10}$$

Hence, from (8) and (10), one obtains

$$\begin{aligned}|H_{\omega,i}|x_\varepsilon &\leqslant \omega\sum_{k=1}^{\ell}E_k((\tilde{L}_k\tilde{U}_k)^{-1}\tilde{N}_k)^{s(k,i)}x_\varepsilon+|1-\omega|x_\varepsilon\\ &\leqslant \omega\sum_{k=1}^{\ell}E_k\rho_\varepsilon^{s(k,i)}x_\varepsilon+|1-\omega|x_\varepsilon\\ &\leqslant \omega\sum_{k=1}^{\ell}E_k\rho_\varepsilon x_\varepsilon+|1-\omega|x_\varepsilon\\ &= (\omega\rho_\varepsilon+|1-\omega|)x_\varepsilon.\end{aligned} \tag{11}$$

Taking the weighted max norm $\|\cdot\|_{x_\varepsilon}$ to both sides of Eq. (11),

$$\|H_{\omega,i}\|_{x_\varepsilon}=\||H_{\omega,i}|x_\varepsilon\|_{x_\varepsilon}\leqslant\omega\rho_\varepsilon+|1-\omega|\equiv\alpha_\varepsilon.$$

Since i is arbitrary, $\|H_{\omega,i}\|_{x_\varepsilon}\leqslant\alpha_\varepsilon<1$ for all $i=1,2,\ldots$. Therefore, the proof is complete. □

Mas et al. [10] showed that the relaxed nonstationary multisplitting method converges to the exact solution of $Ax=b$ for any initial vector x_0 under the assumption that $A=M_k-N_k$ is an H-compatible splitting with $\mathrm{diag}(|M_k|)\leqslant|\mathrm{diag}(A)|$ for $k=1,2,\ldots,\ell$. It was shown in [24] that the ILU factorization $A=L_kU_k-N_k$ used in Theorem 3.1 is not an H-compatible splitting. This means that Theorem 3.1 provides a new convergence result for the relaxed nonstationary multisplitting method which is different from the convergence result in [10]. Since $\rho=\rho(J)<1$ in Theorem 3.1, Theorem 3.1 holds for $\omega=1$ and hence a convergence result for the nonstationary multisplitting method is obtained below.

Corollary 3.2. *Let A be an $n \times n$ H-matrix. Let $Q_1, Q_2, \ldots, Q_\ell$ be zero pattern sets which are subsets of S_n. For each $1 \leqslant k \leqslant \ell$, let $A = L_k U_k - N_k$ be the ILU factorization of A corresponding to Q_k. Then, the nonstationary multisplitting method associated with the multisplitting $(L_k U_k, N_k, E_k)$, $k = 1, 2, \ldots, \ell$, converges to the exact solution of $Ax = b$ for any initial vector x_0.*

The following theorem shows that the convergence result presented in Theorem 3.1 can be improved when J is irreducible.

Theorem 3.3. *Let $A = D - B$ be an $n \times n$ H-matrix with $D = \mathrm{diag}(A)$. Let $J = |D|^{-1}|B|$ and let $Q_1, Q_2, \ldots, Q_\ell$ be zero pattern sets which are subsets of S_n. For each $1 \leqslant k \leqslant \ell$, let $A = L_k U_k - N_k$ be the ILU factorization of A corresponding to Q_k. Suppose that J is irreducible and $v > 0$ is a Perron vector of J. Then, the relaxed nonstationary multisplitting method associated with the multisplitting $(L_k U_k, N_k, E_k)$, $k = 1, 2, \ldots, \ell$, converges to the exact solution of $Ax = b$ for any initial vector x_0 if $0 < \omega < \frac{2}{1+\theta}$, where $\theta = \sup\{\|H_i\|_v \mid i = 1, 2, \ldots\}$ and $H_i = \sum_{k=1}^{\ell} E_k((L_k U_k)^{-1} N_k)^{s(k,i)}$. Moreover, $\theta \leqslant \rho(J) < 1$.*

Proof. Since $v > 0$ is a Perron vector of J, $Jv = \rho(J)v$. For each $1 \leqslant k \leqslant \ell$, let$\langle A \rangle = \tilde{L}_k \tilde{U}_k - \tilde{N}_k$ be the ILU factorization of $\langle A \rangle$ corresponding to Q_k. Using Theorem 2.4, (6) and (7), one obtains that for every $1 \leqslant k \leqslant \ell$

$$\begin{aligned} |(L_k U_k)^{-1} N_k| v &\leqslant (\tilde{L}_k \tilde{U}_k)^{-1} \tilde{N}_k v = (I - (\tilde{L}_k \tilde{U}_k)^{-1} \langle A \rangle) v \\ &= (I - (\tilde{L}_k \tilde{U}_k)^{-1} |D| (I - J)) v \\ &= v - (1 - \rho(J)) (\tilde{L}_k \tilde{U}_k)^{-1} |D| v \\ &\leqslant v - (1 - \rho(J)) v = \rho(J) v. \end{aligned} \tag{12}$$

Using (12) and the fact that $\rho(J) < 1$, one obtains that for each i

$$\begin{aligned} |H_i| v &\leqslant \sum_{k=1}^{\ell} E_k |(L_k U_k)^{-1} N_k|^{s(k,i)} v \\ &\leqslant \sum_{k=1}^{\ell} E_k (\rho(J))^{s(k,i)} v \\ &\leqslant \sum_{k=1}^{\ell} E_k \rho(J) v = \rho(J) v. \end{aligned} \tag{13}$$

From (13), $\|H_i\|_v \leqslant \rho(J)$ for each i and hence $\theta \leqslant \rho(J)$. Notice that $H_{\omega,i} = \omega H_i + (1 - \omega) I$. It follows that $|H_{\omega,i}| v \leqslant \omega |H_i| v + |1 - \omega| v$. Using this relation, one obtains that for each i

$$\begin{aligned} \|H_{\omega,i}\|_v &= \| |H_{\omega,i}| v \|_v \leqslant \| \omega |H_i| v + |1 - \omega| v \|_v \\ &\leqslant \omega \|H_i\|_v + |1 - \omega| \leqslant \omega \theta + |1 - \omega| \equiv \alpha. \end{aligned} \tag{14}$$

If $0 < \omega < \frac{2}{1+\theta}$, then $\alpha < 1$. Hence, from (14) there exists an $\alpha < 1$ such that $\|H_{\omega,i}\|_v \leqslant \alpha$ for all $i = 1, 2, \ldots$. Therefore, the proof is complete from Lemma 2.2. □

It can be easily shown that if A is an irreducible H-matrix, then J in Theorem 3.3 is an irreducible matrix. If $s(k,i) = s(k)$ in Algorithm 1, i.e. the number of inner iterations $s(k,i)$ does not depend on the outer iteration i, then we can have the following convergence result for Algorithm 1 which also improves Theorem 3.1.

Theorem 3.4. *Let $A = D - B$ be an $n \times n$ H-matrix with $D = \mathrm{diag}(A)$. Let $J = |D|^{-1}|B|$ and let $Q_1, Q_2, \ldots, Q_\ell$ be zero pattern sets which are subsets of S_n. For each $1 \leqslant k \leqslant \ell$, let $A = L_k U_k - N_k$ be the ILU factorization of A corresponding to Q_k. Assume that $s(k,i) = s(k)$ for all $i = 1, 2, \ldots$. Then, the relaxed nonstationary multisplitting method associated with the multisplitting $(L_k U_k, N_k, E_k)$, $k = 1, 2, \ldots, \ell$, converges to the exact solution of $Ax = b$ for any initial vector x_0 if $0 < \omega < \frac{2}{1+\rho(H)}$, where $H = \sum_{k=1}^{\ell} E_k((L_k U_k)^{-1} N_k)^{s(k)}$. Moreover, $\rho(H) \leqslant \rho(J) < 1$.*

Proof. For each $1 \leqslant k \leqslant \ell$, let$\langle A \rangle = \tilde{L}_k \tilde{U}_k - \tilde{N}_k$ be the ILU factorization of $\langle A \rangle$ corresponding to Q_k. Let $\tilde{H} = \sum_{k=1}^{\ell} E_k((\tilde{L}_k \tilde{U}_k)^{-1} \tilde{N}_k)^{s(k)}$. Then, $\tilde{H}$ can be viewed as the iteration matrix of the nonstationary multisplitting method with $s(k,i) = s(k)$ for solving a linear system whose coefficient matrix is $\langle A \rangle$. Thus, Corollary 3.2 implies $\rho(\tilde{H}) < 1$. Let

$$\tilde{P} = \sum_{k=1}^{\ell} E_k \left(\sum_{j=0}^{s(k)-1} ((\tilde{L}_k \tilde{U}_k)^{-1} \tilde{N}_k)^j \right) (\tilde{L}_k \tilde{U}_k)^{-1}.$$

Then, $\tilde{P}\langle A \rangle = I - \tilde{H}$. Since $\rho(\tilde{H}) < 1$, $\tilde{P}$ is nonsingular and thus $\langle A \rangle = \tilde{P}^{-1} - \tilde{P}^{-1}\tilde{H}$. It is clear that

$$\langle A \rangle = \tilde{P}^{-1} - \tilde{P}^{-1}\tilde{H} = |D| - |B|$$

are weak regular splittings of $\langle A \rangle$. Since $|D^{-1}| \leqslant (\tilde{L}_k \tilde{U}_k)^{-1}$ for all $k = 1, 2, \ldots, \ell$,

$$(\tilde{P}^{-1})^{-1} = \tilde{P} \geqslant \sum_{k=1}^{\ell} E_k (\tilde{L}_k \tilde{U}_k)^{-1} \geqslant \sum_{k=1}^{\ell} E_k |D|^{-1} = |D|^{-1}. \tag{15}$$

Using Eq. (15) and the fact that $|B| \geqslant 0$, Elsner's comparison lemma [5] implies that

$$\rho(\tilde{H}) \leqslant \rho(J). \tag{16}$$

Notice that $\rho(J) < 1$ from Lemma 2.1. Since $|H| \leqslant \tilde{H}$ from Theorem 2.4, $\rho(H) \leqslant \rho(\tilde{H})$. Hence, Eq. (16) implies that

$$\rho(H) \leqslant \rho(\tilde{H}) \leqslant \rho(J) < 1.$$

Let $H_\omega = \omega H + (1-\omega)I$. Then, H_ω is the iteration matrix of the relaxed nonstationary multisplitting method with $s(k,i) = s(k)$. It can be easily shown that

$$\rho(H_\omega) \leqslant \omega\rho(H) + |1-\omega|. \tag{17}$$

Since $0 < \omega < \frac{2}{1+\rho(H)}$ from the assumption and $\rho(H) < 1$, it is easy to show that $\omega\rho(H) + |1-\omega| < 1$. From Eq. (17), $\rho(H_\omega) < 1$. Therefore, the proof is complete. □

Next, we give a convergence result for the relaxed nonstationary two-stage multisplitting method (Algorithm 2) using ILU factorizations as inner splittings. Let $R_{k,\omega} = \omega B_k^{-1} C_k + (1-\omega)I$. Then, Algorithm 2 can be written as

$$x_i = H^*_{\omega,i} x_{i-1} + P^*_{\omega,i} b, \quad i = 1, 2, \ldots, \tag{18}$$

where

$$H^*_{\omega,i} = \sum_{k=1}^{\ell} E_k R_{k,\omega}^{s(k,i)} + \omega \sum_{k=1}^{\ell} E_k \left(\sum_{j=0}^{s(k,i)-1} R^j_{k,\omega} \right) B_k^{-1} N_k, \quad i = 1, 2, \ldots$$

and

$$P^*_{\omega,i} = \omega \sum_{k=1}^{\ell} E_k \left(\sum_{j=0}^{s(k,i)-1} R^j_{k,\omega} \right) B_k^{-1}, \quad i = 1, 2, \ldots .$$

The $H^*_{\omega,i}$'s are called iteration matrices for Algorithm 2. It is easy to show that $P^*_{\omega,i} A = I - H^*_{\omega,i}$ for each i. Hence, the exact solution ξ of $Ax = b$ satisfies

$$\xi = H^*_{\omega,i} \xi + P^*_{\omega,i} b, \quad i = 1, 2, \ldots . \tag{19}$$

From (18) and (19), the error vector $e_i = x_i - \xi$ satisfies

$$e_i = H^*_{\omega,i} e_{i-1} = H^*_{\omega,i} H^*_{\omega,i-1} \cdots H^*_{\omega,1} e_0, \quad i = 1, 2, \ldots . \tag{20}$$

From (20), the sequence of vectors generated by the iteration (18) converges to the exact solution of $Ax = b$ for any initial vector x_0 if and only if

$$\lim_{i \to \infty} H^*_{\omega,i} H^*_{\omega,i-1} \cdots H^*_{\omega,1} = 0. \tag{21}$$

Theorem 3.5 (*Bru et al.* [3]). *Let $A^{-1} \geqslant 0$ be an $n \times n$ matrix. For each $1 \leqslant k \leqslant \ell$, let $A = M_k - N_k$ be a regular splitting of A and $M_k = B_k - C_k$ be a weak regular splitting of M_k. Then, the relaxed nonstationary two-stage multisplitting method with $A = M_k - N_k$ as outer splittings and $M_k = B_k - C_k$ as inner splittings converges to the exact solution of $Ax = b$ for any initial vector x_0 if $0 < \omega \leqslant 1$.*

Theorem 3.6. *Let A be an $n \times n$ H-matrix. Let $Q_1, Q_2, \ldots, Q_\ell$ be zero pattern sets which are subsets of S_n. For each $1 \leqslant k \leqslant \ell$, let $A = M_k - N_k$ be an H-compatible splitting and $M_k = L_k U_k - C_k$ be the ILU factorization of M_k corresponding to Q_k. Then, the relaxed nonstationary two-stage multisplitting method with $A = M_k - N_k$ as outer splittings and $M_k = L_k U_k - C_k$ as inner splittings converges to the exact solution of $Ax = b$ for any initial vector x_0 if $0 < \omega \leqslant 1$.*

Proof. Since $A = M_k - N_k$ is an H-compatible splitting of an H-matrix A, M_k is an H-matrix and thus $\langle M_k \rangle$ is an M-matrix. For each $1 \leqslant k \leqslant \ell$, let$\langle M_k \rangle = \tilde{L}_k \tilde{U}_k - \tilde{C}_k$ be the ILU factorization of $\langle M_k \rangle$ corresponding to Q_k and let $\tilde{R}_{k,\omega} = \omega(\tilde{L}_k \tilde{U}_k)^{-1} \tilde{C}_k + (1-\omega)I$. Since $0 < \omega \leqslant 1$ from the assumption, Theorem 2.4 implies that

$$|R_{k,\omega}| = |\omega(L_k U_k)^{-1} C_k + (1-\omega)I| \leqslant \tilde{R}_{k,\omega}. \tag{22}$$

From Eq. (22) and Theorem 2.4, one obtains

$$\begin{aligned}|H^*_{\omega,i}| &= \left|\sum_{k=1}^{\ell} E_k R_{k,\omega}^{s(k,i)} + \omega \sum_{k=1}^{\ell} E_k \left(\sum_{j=0}^{s(k,i)-1} R_{k,\omega}^{j}\right)(L_k U_k)^{-1} N_k\right| \\ &\leqslant \sum_{k=1}^{\ell} E_k \tilde{R}_{k,\omega}^{s(k,i)} + \omega \sum_{k=1}^{\ell} E_k \left(\sum_{j=0}^{s(k,i)-1} \tilde{R}_{k,\omega}^{j}\right)(\tilde{L}_k \tilde{U}_k)^{-1} |N_k|. \end{aligned} \tag{23}$$

Let $\tilde{H}^*_{\omega,i}$ denote the matrix in the right-hand side of Eq. (23). Then, the $\tilde{H}^*_{\omega,i}$'s are iteration matrices of the relaxed nonstationary two-stage multisplitting method with $\langle A\rangle = \langle M_k\rangle - |N_k|$ as outer splittings and $\langle M_k\rangle = \tilde{L}_k\tilde{U}_k - \tilde{C}_k$ as inner splittings for solving a linear system whose coefficient matrix is $\langle A\rangle$. Note that $\langle A\rangle = \langle M_k\rangle - |N_k|$ and $\langle M_k\rangle = \tilde{L}_k\tilde{U}_k - \tilde{C}_k$ are regular splittings of $\langle A\rangle$ and $\langle M_k\rangle$, respectively. Since $\langle A\rangle^{-1} \geqslant 0$,Theorem 3.5 implies that

$$\lim_{i\to\infty} \tilde{H}^*_{\omega,i}\tilde{H}^*_{\omega,i-1}\cdots\tilde{H}^*_{\omega,1} = 0. \tag{24}$$

Since $|H^*_{\omega,i}| \leqslant \tilde{H}^*_{\omega,i}$ from Eq. (23), one obtains

$$|H^*_{\omega,i}H^*_{\omega,i-1}\cdots H^*_{\omega,1}| \leqslant \tilde{H}^*_{\omega,i}\tilde{H}^*_{\omega,i-1}\cdots\tilde{H}^*_{\omega,1}. \tag{25}$$

From (24) and (25), $\lim_{i\to\infty} H^*_{\omega,i}H^*_{\omega,i-1}\cdots H^*_{\omega,1} = 0$. Therefore, the proof is complete. □

It was shown in [3] that the relaxed nonstationary two-stage multisplitting method converges to the exact solution of $Ax = b$ for any initial vector x_0 under the assumption that both outer splittings and inner splittings are H-compatible splittings. However, Theorem 3.6 uses the ILU factorizations instead of using H-compatible splittings as inner splittings.

4. Parallel implementation and application of Algorithm 2

In this section, we consider a parallel implementation of the relaxed nonstationary two-stage multisplitting method (Algorithm 2) using ILU factorizations as inner splittings and an application of Algorithm 2 with $s(k,i) = s(k)$ to parallel preconditioner of Krylov subspace iterative methods such as the CGS, GMRES and Bi-CGSTAB. First, we introduce a parallel implementation of Algorithm 2 using ILU factorizations as inner splittings for solving the linear system (1). Let ℓ denote the number of processors to be used. For simplicity of exposition, suppose that $\ell = 3$. Then, the H-matrix A is partitioned into a 3×3 block matrix of the form

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix},$$

where the diagonal blocks A_{ii} of A are square matrices. Let $A = M - N$, where

$$M = \begin{pmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{pmatrix}, \quad N = \begin{pmatrix} 0 & -A_{12} & -A_{13} \\ -A_{21} & 0 & -A_{23} \\ -A_{31} & -A_{32} & 0 \end{pmatrix} \equiv \begin{pmatrix} N^{(1)} \\ N^{(2)} \\ N^{(3)} \end{pmatrix}. \tag{26}$$

Then, we construct a multisplitting (M_k, N_k, E_k), $k = 1, 2, 3$, where

$$E_1 = \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad E_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{pmatrix}, \tag{27}$$

$M_k = M$ and $N_k = N$ for $k = 1, 2, 3$. Clearly, $A = M_k - N_k$ is an H-compatible splitting for each k. Let $M = LU - C$ be an ILU factorization of M corresponding to a zero pattern set $Q \subset S_n$. Observe that the L and U are of the form

$$L = \begin{pmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ 0 & 0 & L_3 \end{pmatrix}, \quad U = \begin{pmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{pmatrix}, \tag{28}$$

where L_i's are lower triangular matrices and U_i's are upper triangular matrices. Let $B_k = LU$ and $C_k = C$ for $k = 1, 2, 3$. Then, at the ith iteration of Algorithm 2 each processor k executes the following algorithm COM(k, ω):

Algorithm. COM(k, ω)

```
$y_{k,0} = x_{i-1}$
$\bar{b} = b + N x_{i-1}$
For $j = 1$ to $s(k, i)$
    $y_{k,j} = \omega (LU)^{-1} (C y_{k,j-1} + \bar{b}) + (1 - \omega) y_{k,j-1}$
Compute $E_k y_{k,s(k,i)}$.
```

In general, the ILU factorization of M does not compute C, but it computes only the L and U. So, the computational step $y_{k,j} = \omega (LU)^{-1} (C y_{k,j-1} + \bar{b}) + (1 - \omega) y_{k,j-1}$ in the COM(k,ω) is transformed into

$$y_{k,j} = y_{k,j-1} + \omega (LU)^{-1} (\bar{b} - M y_{k,j-1}). \tag{29}$$

Assume that b, $\bar{b}$, $y_{k,j}$ and x_i are partitioned into

$$b = \begin{pmatrix} b^{(1)} \\ b^{(2)} \\ b^{(3)} \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} \bar{b}^{(1)} \\ \bar{b}^{(2)} \\ \bar{b}^{(3)} \end{pmatrix}, \quad y_{k,j} = \begin{pmatrix} y_{k,j}^{(1)} \\ y_{k,j}^{(2)} \\ y_{k,j}^{(3)} \end{pmatrix}, \quad x_i = \begin{pmatrix} x_i^{(1)} \\ x_i^{(2)} \\ x_i^{(3)} \end{pmatrix}. \tag{30}$$

Making use of (26)–(30), at the *i*th iteration of Algorithm 2 each processor *k* needs to execute only the following algorithm MCOM(k, ω) which requires much less computation than COM(k, ω):

Algorithm. MCOM(k, ω)

```
$y_{k,0}^{(k)} = x_{i-1}^{(k)}$
$\bar{b}^{(k)} = b^{(k)} + N^{(k)} x_{i-1}$
For $j = 1$ to $s(k, i)$
    $y_{k,j}^{(k)} = y_{k,j-1}^{(k)} + \omega (L_k U_k)^{-1} (\bar{b}^{(k)} - A_{kk} y_{k,j-1}^{(k)})$
$x_i^{(k)} = y_{k,s(k,i)}^{(k)}$.
```

Notice that $(L_k U_k)^{-1} (\bar{b}^{(k)} - A_{kk} y_{k,j-1}^{(k)})$ in the MCOM(k, ω) is computed by solving the linear system $(L_k U_k) t_k = \bar{b}^{(k)} - A_{kk} y_{k,j-1}^{(k)}$ for t_k without computing $(L_k U_k)^{-1}$ explicitly.

Since Algorithm 2 described above requires too many iterations for convergence and thus it does not perform well (see Tables 1 and 2), we next consider an application of Algorithm 2 with $s(k, i) = s(k)$ to parallel preconditioner of Krylov subspace methods. Since $s(k, i) = s(k)$, from (18) $H_{\omega,i}^* = H_\omega^*$ and $P_{\omega,i}^* = P_\omega^*$ for all $i = 1, 2, \ldots$, where

$$H_\omega^* = \sum_{k=1}^{\ell} E_k R_{k,\omega}^{s(k)} + \omega \sum_{k=1}^{\ell} E_k \left(\sum_{j=0}^{s(k)-1} R_{k,\omega}^j \right) B_k^{-1} N_k,$$

$$P_\omega^* = \omega \sum_{k=1}^{\ell} E_k \left(\sum_{j=0}^{s(k)-1} R_{k,\omega}^j \right) B_k^{-1}.$$

If Algorithm 2 with $s(k, i) = s(k)$ converges to the exact solution of $Ax = b$ for any initial vector x_0, then $\rho(H_\omega^*) < 1$. It follows that the matrix P_ω^* such that $P_\omega^* A = I - H_\omega^*$ is nonsingular. Hence, $(P_\omega^*)^{-1}$ can be used as a preconditioner of Krylov subspace methods. Then, the preconditioner solver step which is one of the basic time-consuming computational kernels of Krylov subspace methods is equivalent to computing $P_\omega^* r$ for a vector $r \in \mathbb{R}^n$. Notice that $P_\omega^* r$ can be computed in parallel by computing each $\omega E_k (\sum_{j=0}^{s(k)-1} R_{k,\omega}^j) B_k^{-1} r$ on a different processor and then adding them in parallel. Also assume that M and E_k's are defined as in (26) and (27), and $M = LU - C$ is an ILU factorization of M corresponding to a zero pattern set $Q \subset S_n$. Let $M_k = M = LU - C$, $B_k = LU$, and $C_k = C$ for all $k = 1, 2, \ldots, \ell$. Since L and U are of the form (28), each processor *k* needs to execute only the following algorithm PREC(k, ω) for parallel computation of $z = P_\omega^* r$:

Algorithm. PREC(k, ω)

```
$t_0 = 0$
For $j = 1$ to $s(k)$
    $t_j = t_{j-1} + \omega (L_k U_k)^{-1} (r^{(k)} - A_{kk} t_{j-1})$
$z^{(k)} = t_{s(k)}$.
```

Table 1
Parallel performance of Algorithm 2 using ILU factorizations when $n = 256^2$, $\omega = 1$, $s(k) = \alpha$ for $1 \leqslant k \leqslant \frac{\ell}{2}$, and $s(k) = 3\alpha$ for $(\frac{\ell}{2} + 1) \leqslant k \leqslant \ell$

α	ℓ	Example 5.1		Example 5.2	
		Iter	*I*-time	Iter	*I*-time
1	2	7849	47.9	6846	42.0
	4	8008	24.2	6919	21.1
	8	8235	12.3	7248	11.0
	16	8672	7.07	7516	6.13
2	2	3961	45.2	3436	39.7
	4	4139	23.7	3536	20.3
	8	4394	12.2	3893	11.0
	16	4886	7.01	4208	6.18

Table 2
The number of iterations of Algorithm 2 using ILU factorizations when $n = 256^2$, $s(k) = \alpha$ for $1 \leqslant k \leqslant \frac{\ell}{2}$, and $s(k) = 3\alpha$ for $(\frac{\ell}{2} + 1) \leqslant k \leqslant \ell$

α	ω	Example 5.1				Example 5.2			
		$\ell = 2$	$\ell = 4$	$\ell = 8$	$\ell = 16$	$\ell = 2$	$\ell = 4$	$\ell = 8$	$\ell = 16$
1	0.9	8719	8895	9144	9621	7607	7687	8052	8345
	1.0	7849	8009	8235	8672	6846	6919	7248	7516
	1.1	7137	7283	7491	7895	6224	6291	6591	6838
	1.2	6544	6681	6872	7249	5706	5768	6043	6273
	1.3	6042	6167	6347	6701	5268	5325	5579	5795
	1.4	5611	NC	NC	NC	4891	NC	NC	NC
	1.5	NC	NC	NC	NC	NC	NC	NC	NC
2	0.9	4393	4573	4830	5325	3815	3913	4276	4591
	1.0	3961	4139	4394	4886	3436	3536	3893	4208
	1.1	3609	3787	4044	4539	3126	3232	3587	3908
	1.2	3317	3499	3760	4264	2869	2982	3342	3673
	1.3	3072	3261	3532	4052	2652	2777	3148	3495
	1.4	2865	3068	3353	3899	2467	2611	3002	3373
	1.5	2694	2917	3229	3816	2310	2488	2911	3319
	1.6	2562	2827	3183	3855	NC	NC	NC	NC
	1.7	NC	NC	NC	NC	NC	NC	NC	NC

The output vector z and the input vector r in the PREC(k, ω) are partitioned as in (30). Since other time-consuming computational kernels of Krylov subspace methods can be easily parallelized, Krylov subspace methods with the preconditioner $(P_\omega^*)^{-1}$ can be fully parallelized using the PREC(k, ω).

5. Numerical results

Numerical experiments for Algorithm 2 described in Section 4 are made with $s(k,i)=s(k)$, and Krylov subspace method used for numerical experiments is the BiCGSTAB with the right preconditioner $(P^*_\omega)^{-1}$ which is described in Section 4. All numerical experiments are carried out using 64-bit arithmetic on the IBM p690 supercomputer at KISTI (Korean Institute of Science and Technology Information), an SMP system with 32 processors. Parallel codes are written in OpenMP Fortran [15], and all nonzero elements of A are stored using the compressed row storage format [16]. For all timing runs, elapsed wall-clock time is measured in seconds using the IBM wall-clock timer **rtc**. For both Algorithm 2 and BiCGSTAB with the preconditioner $(P^*_\omega)^{-1}$, the initial vector x_0 is set to zero, the ILU factorization without fill-in elements is used, and the stopping criterion is $\|b-Ax_i\|_2/\|b\|_2<10^{-8}$, where $\|\cdot\|_2$ refers to L_2-norm.

For numerical experiments of both Algorithm 2 and BiCGSTAB with the preconditioner $(P^*_\omega)^{-1}$, the number of inner iterations $s(k)$ is set to α for $1\leqslant k\leqslant \frac{\ell}{2}$ and 3α for $(\frac{\ell}{2}+1)\leqslant k\leqslant \ell$, where α is a positive integer and ℓ is assumed to be a multiple of 2. For this choice of $s(k)$, the last half processors execute 3 times more inner iterations than the first half processors. Thus, in order to obtain a good load balance among the processors, the computational amount of the first half processors should be 3 times more than that of the last half processors. For test problems used in this paper, this can be achieved by partitioning $A\in\mathbb{R}^{n\times n}$ into an $\ell\times\ell$ block matrix such that the first $\frac{\ell}{2}$ diagonal blocks are square matrices of order $\frac{3n}{2\ell}$ and the remaining $\frac{\ell}{2}$ diagonal blocks are square matrices of order $\frac{n}{2\ell}$, where n is assumed to be a multiple of 2ℓ. There is no special reason for using the factor 3. If a factor p which is different from the factor 3 is used, then the factor 3 in the above arguments should be changed to p. Since it was seen that there is no change in numerical conclusion for different factors p, the factor 3 is used in this paper for numerical experiments.

The test matrix A used in this paper is obtained from five-point discretization of the following elliptic second-order PDE:

$$-(au_x)_x-(bu_y)_y+(cu)_x+(du)_y+fu=g \tag{31}$$

with $a(x,y)>0$, $b(x,y)>0$, $c(x,y)$, $d(x,y)$, and $f(x,y)$ defined on the unit square region $\Omega=(0,1)\times(0,1)$, and with the Dirichlet boundary condition $u(x,y)=0$ on the boundary of Ω. Only the discretized matrix A is of importance, so the right-hand side vector b is created from Ae, where $e=(1,1,\ldots,1)^{\mathrm{T}}\in\mathbb{R}^n$. Therefore, the right-hand side function $g(x,y)$ in (31) is not relevant.

Example 5.1. This example considers Eq. (31) with $a(x,y)=b(x,y)=1$, $c(x,y)=10e^{xy}$, $d(x,y)=10e^{-xy}$, and $f(x,y)=0$. We have used a uniform mesh of $\Delta x=\Delta y=1/(m+1)$, which leads to a matrix of order $n=m\times m$, where Δx and Δy refer to the mesh sizes in the x- and y-direction, respectively.

Example 5.2. This example considers Eq. (31) with $c(x,y)=10(x+y)$, $d(x,y)=10(x-y)$, $f(x,y)=0$, and $a(x,y)=b(x,y)$ defined as

$$a(x,y)=\begin{cases}10^3 & \text{if } \frac{1}{4}<x,y<\frac{3}{4},\\ 1 & \text{otherwise.}\end{cases}$$

We have used the same uniform mesh as Example 5.1.

Table 1 contains parallel performance results of Algorithm 2 for $n = 256^2$. In Table 1, Iter stands for the number of iterations of Algorithm 2 and *I*-time stands for the parallel execution time of Algorithm 2. Table 2 contains the number of iterations of Algorithm 2 for various values of ω. In Table 2, NC denotes that Algorithm 2 does not converge to the exact solution of $Ax = b$. Table 3 contains parallel performance results of BiCGSTAB with the preconditioner $(P_1^*)^{-1}$ for $n = 256^2$, and Table 4 contains parallel performance results of BiCGSTAB with the preconditioner $(P_\omega^*)^{-1}$ for $n = 384^2$ and various values of ω. In Tables 3 and 4, Iter stands for the number of iterations of BiCGSTAB, *I*-time stands for the parallel execution time of BiCGSTAB and $I\text{-tavg} = \frac{I\text{-time}}{\text{Iter}}$ which means an average parallel execution time of BiCGSTAB per iteration.

As can be seen in Table 1, Iter for Algorithm 2 increases significantly as ℓ (i.e., the number of processors) increases since the M used in the H-compatible splitting $A = M - N$ approximates A better for smaller ℓ. Thus, *I*-time for Algorithm 2 scales worse as ℓ increases. From Table 1, it can be also seen that Iter for $\alpha = 2$ is much smaller than that for $\alpha = 1$, but *I*-time for $\alpha = 2$ is about the same as that for $\alpha = 1$ except for $\ell = 2$. It means that parallel performance of Algorithm 2 does not depend a lot on the number of inner iterations $s(k)$. Theorem 3.6 showed that Algorithm 2 using ILU factorizations as inner splittings converges to the exact solution of $Ax = b$ for $0 < \omega \leqslant 1$ when outer splittings are H-compatible splittings. Numerical results in Table 2 also show this theoretical result. Actually, Algorithm 2 converges to the exact solution of $Ax = b$ upto about $\omega = 1.3$ for test problems used in this paper. As can be seen in Table 2, larger value of ω for which Algorithm 2 converges to the exact solution of $Ax = b$ provides better performance.

Algorithm 2 requires too many iterations for convergence which lead to poor performance (see Tables 1 and 2), while BiCGSTAB with the preconditioner $(P_\omega^*)^{-1}$ which is derived from Algorithm 2 performs very well (see Tables 3 and 4). From Table 3, it can be seen that Iter for $\alpha = 1$ is greater than that for $\alpha = 2$, but *I*-time for $\alpha = 1$ is smaller than that for $\alpha = 2$. It means that parallel performance of BiCGSTAB with the preconditioner $(P_1^*)^{-1}$ for $\alpha = 1$ is better than that for $\alpha = 2$. So, parallel performance results of BiCGSTAB with the preconditioner $(P_\omega^*)^{-1}$ only for $\alpha = 1$ are given in Table 4 for various values of ω. Since Iter for BiCGSTAB varies depending upon ℓ (i.e., the number of processors), *I*-tavg is

Table 3
Parallel performance of BiCGSTAB with the preconditioner $(P_1^*)^{-1}$ when $n = 256^2$, $s(k) = \alpha$ for $1 \leqslant k \leqslant \frac{\ell}{2}$, and $s(k) = 3\alpha$ for $(\frac{\ell}{2} + 1) \leqslant k \leqslant \ell$

α	ℓ	Example 5.1			Example 5.2		
		Iter	*I*-time	*I*-tavg	Iter	*I*-time	*I*-tavg
1	2	169	2.43	0.0144	123	1.80	0.0146
	4	172	1.23	0.0072	128	0.92	0.0072
	8	178	0.65	0.0037	131	0.48	0.0037
	16	196	0.40	0.0020	135	0.28	0.0021
2	2	113	2.82	0.0249	85	2.11	0.0248
	4	129	1.57	0.0122	94	1.14	0.0121
	8	144	0.88	0.0061	103	0.64	0.0062
	16	130	0.43	0.0033	112	0.39	0.0035

Table 4
Parallel performance of BiCGSTAB with the preconditioner $(P_\omega^*)^{-1}$ when $n = 384^2$, $s(k) = 1$ for $1 \leqslant k \leqslant \frac{\ell}{2}$, and $s(k) = 3$ for $(\frac{\ell}{2} + 1) \leqslant k \leqslant \ell$

ω	ℓ	Example 5.1			Example 5.2		
		Iter	*I*-time	*I*-tavg	Iter	*I*-time	*I*-tavg
0.9	2	260	8.34	0.0321	196	6.41	0.0327
	4	262	4.16	0.0159	211	3.41	0.0162
	8	266	2.16	0.0081	207	1.71	0.0083
	16	270	1.16	0.0043	223	1.01	0.0045
1.0	2	254	8.13	0.0320	185	6.05	0.0327
	4	246	3.89	0.0158	182	2.92	0.0160
	8	252	2.02	0.0080	188	1.55	0.0082
	16	268	1.13	0.0042	220	0.97	0.0044
1.1	2	250	8.00	0.0320	179	5.83	0.0326
	4	248	3.92	0.0158	180	2.89	0.0161
	8	269	2.15	0.0080	190	1.56	0.0082
	16	251	1.07	0.0043	183	0.83	0.0045
1.2	2	248	7.94	0.0320	168	5.47	0.0326
	4	241	3.81	0.0158	183	2.94	0.0161
	8	251	2.01	0.0080	187	1.53	0.0082
	16	258	1.11	0.0043	186	0.84	0.0045
1.3	2	251	8.04	0.0320	185	6.03	0.0326
	4	210	3.32	0.0158	190	3.05	0.0161
	8	236	1.88	0.0080	186	1.52	0.0082
	16	249	1.05	0.0042	179	0.80	0.0045

provided in Tables 3 and 4 to evaluate parallel efficiency of BiCGSTAB with the preconditioner $(P_\omega^*)^{-1}$ for only one iteration.

In Tables 2 and 4, note that *I*-time is proportional to Iter when α and ℓ are fixed. Iter for Algorithm 2 decreases as ω increases, while Iter for BiCGSTAB with the preconditioner $(P_\omega^*)^{-1}$ varies irregularly as ω increases. For example, ω for which Algorithm 2 performs best is 1.4 for $\ell = 2$ and 1.3 for $\ell \geqslant 4$ when $\alpha = 1$, while ω for which BiCGSTAB performs best varies between 1.1 and 1.3.

The scaling behaviors of BiCGSTAB with the preconditioner $(P_\omega^*)^{-1}$ for Example 5.2 when $n = 384^2$ and $\alpha = 1$ are depicted in Figs. 1 and 2 by log–log scale. The scaling behaviors for Example 5.1 are not depicted since they are similar to those for Example 5.2. *I*-tavg scales perfectly up to $\ell = 8$ (see Table 4 and Fig. 2). The reason why *I*-tavg for $\ell = 4$ is less than one half of *I*-tavg for $\ell = 2$ is that the computational amount of each processor for $\ell = 2$ is more than twice of that for $\ell = 4$ when the preconditioner solver step is computed in parallel (see Section 4). For $\ell > 8$, *I*-tavg does not scale perfectly because of memory

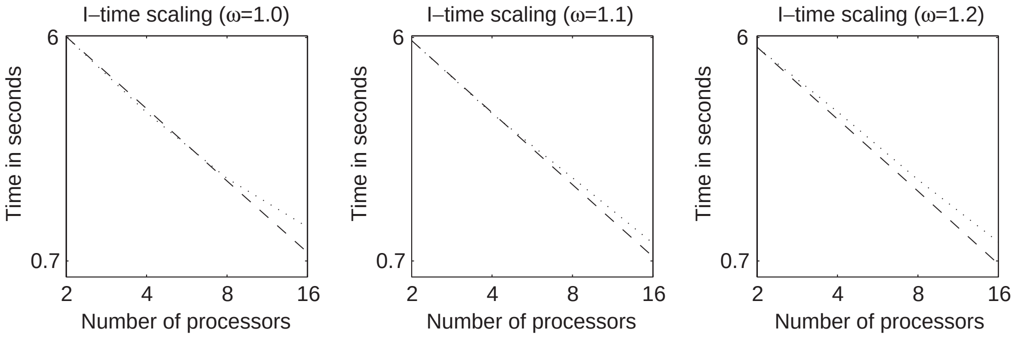

Fig. 1. Scaling behaviors of BiCGSTAB with the preconditioner $(P^*_\omega)^{-1}$ for Example 5.2 when $n = 384^2$ and $\alpha = 1$. I-time scalings: dotted, Perfect scalings: dashed.

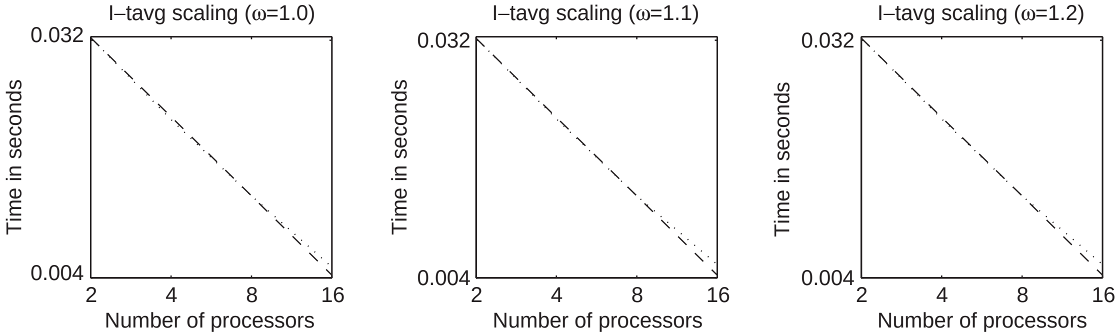

Fig. 2. Scaling behaviors of BiCGSTAB with the preconditioner $(P^*_\omega)^{-1}$ for Example 5.2 when $n = 384^2$ and $\alpha = 1$. I-tavg scalings: dotted, Perfect scalings: dashed.

access contention among the processors. Also notice that *I*-time scales worse than *I*-tavg since Iter is larger for larger ℓ in many cases (see Table 4 and Figs. 1 and 2).

6. Concluding remarks

In this paper, we considered the convergence of two relaxed types of nonstationary multisplitting methods using ILU factorizations, and we provided parallel performance results of the relaxed nonstationary two-stage multisplitting method (Algorithm 2) and BiCGSTAB with the parallel preconditioner $(P^*_\omega)^{-1}$ which is derived from Algorithm 2. Numerical experiments showed that Algorithm 2 itself does not perform well since it requires too many iterations for convergence. However, the methodology of combining Algorithm 2 with Krylov subspace methods such as BiCGSTAB works very well (i.e., BiCGSTAB with the preconditioner $(P^*_\omega)^{-1}$ performs very well as compared with Algorithm 2). It was also seen that the relaxation parameter ω for which Algorithm 2 performs best is not the same as the ω for which BiCGSTAB with the preconditioner $(P^*_\omega)^{-1}$ performs best. For test problems used in this paper, ω for which Algorithm 2 performs best is 1.3, while ω for which BiCGSTAB with the preconditioner $(P_\omega^*)^{-1}$ performs best varies between 1.1 and 1.3.

Theorem 3.6 showed that Algorithm 2 converges to the exact solution of $Ax=b$ for $0<\omega\leqslant 1$. For test problems used in this paper, Algorithm 2 converges to the exact solution of $Ax=b$ upto about $\omega=1.3$. The practical upper bound of ω guaranteeing the convergence of Algorithm 2 is usually greater than the theoretical upper bound 1 of ω, and it varies depending upon the problem to be considered. It means that when $(P_\omega^*)^{-1}$ is used as a parallel preconditioner of Krylov subspace methods, a range of ω providing good performance can be chosen from numerical experiments of Algorithm 2.

Acknowledgements

This work was supported by Grant no. R05-2003-000-10239-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

References

[1] A. Berman, R.J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, Academic Press, New York, 1979.
[2] R. Bru, R. Fuster, Parallel chaotic extrapolated Jacobi method, Appl. Math. Lett. 3 (1990) 65–69.
[3] R. Bru, V. Migallón, J. Penadés, D.B. Szyld, Parallel, synchronous and asynchronous two-stage multisplitting methods, Electron. Trans. Numer. Anal. 3 (1995) 24–38.
[4] J.J. Buoni, A stable method for the incomplete factorization of H-matrices, Linear Algebra Appl. 129 (1990) 143–154.
[5] L. Elsner, Comparisons of weak regular splittings and multisplitting methods, Numer. Math. 56 (1989) 283–289.
[6] A. Frommer, G. Mayer, Convergence of relaxed parallel multisplitting methods, Linear Algebra Appl. 119 (1989) 141–152.
[7] A. Frommer, D.B. Szyld, H-splittings and two-stage iterative methods, Numer. Math. 63 (1992) 345–356.
[8] S.W. Kim, J.H. Yun, Block ILU factorization preconditioners for a block-tridiagonal H-matrix, Linear Algebra Appl. 317 (2000) 103–125.
[9] T.A. Manteuffel, An incomplete factorization technique for positive definite linear systems, Math. Comput. 34 (1980) 473–497.
[10] J. Mas, V. Migallón, J. Penadés, D.B. Szyld, Nonstationary parallel relaxed multisplitting methods, Linear Algebra Appl. 241–243 (1996) 733–747.
[11] J.A. Meijerink, H.A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. Comput. 31 (1977) 148–162.
[12] A. Messaoudi, On the stability of the incomplete LU factorizations and characterizations of H-matrices, Numer. Math. 69 (1995) 321–331.
[13] M. Neumann, R.J. Plemmons, Convergence of parallel multisplitting methods for M-matrices, Linear Algebra Appl. 88–89 (1987) 559–574.
[14] D.P. O'Leary, R.E. White, Multisplittings of matrices and parallel solution of linear systems, SIAM J. Algebraic Discrete Methods 6 (1985) 630–640.
[15] OpenMP Architecture Review Board, OpenMP Fortran Application Program Interface Version 2.0, 2000.
[16] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, 1996.
[17] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[18] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 10 (1989) 36–52.
[19] D.B. Szyld, M.T. Jones, Two-stage and multisplitting methods for the parallel solution of linear systems, SIAM J. Matrix Anal. Appl. 13 (1992) 671–679.

[20] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 13 (1992) 631–644.

[21] R.S. Varga, Factorization and normalized iterative methods, in: R.E. Langer (Ed.), Boundary Problems in Differential Equations, University of Wisconsin Press, Madison, 1960.

[22] R.S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, 1962.

[23] R.E. White, Multisplitting of a symmetric positive definite matrix, SIAM J. Matrix Anal. Appl. 11 (1990) 69–82.

[24] J.H. Yun, Performance of ILU factorization preconditioners based on multisplittings, Numer. Math. 95 (2003) 761–779.