Electronic Transactions on Numerical Analysis.
Volume 12, pp. 88-112, 2001.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

ON PARALLEL TWO-STAGE METHODS FOR HERMITIAN POSITIVE DEFINITE MATRICES WITH APPLICATIONS TO PRECONDITIONING*

M. JESÚS CASTEL†, VIOLETA MIGALLÓN†, AND JOSÉ PENADÉS†

Abstract. Parallel two-stage iterative methods for the solution of linear systems are analyzed. Convergence properties of both block and multisplitting two-stage methods are investigated either when the number of inner iterations becomes sufficiently large or when the matrix of the linear system is Hermitian positive definite. Comparison theorems for the parallel two-stage methods, based on the number of inner iterations performed, are given. Computational results of these methods on two parallel computing systems are included to illustrate the theoretical results. Also, the use of these methods as preconditioners is studied from the experimental point of view.

Key words. linear systems, two-stage methods, block methods, multisplitting methods, Hermitian matrix, positive definite matrix, preconditioners, parallel algorithms, monotonicity, distributed memory.

AMS subject classifications. 65F10, 65F15.

1. Introduction. We are interested in the iterative solution, on a parallel computer, of nonsingular linear systems of equations

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

where $A \in C^{n\times n}$, and x and b are n–vectors. Suppose that A is partitioned into $r \times r$ blocks, with square diagonal blocks of order n_j, $\sum_{j=1}^{r} n_j = n$, such that system (1.1) can be written as

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1r} \\ A_{21} & A_{22} & \cdots & A_{2r} \\ \vdots & \vdots & & \vdots \\ A_{r1} & A_{r2} & \cdots & A_{rr} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_r \end{bmatrix}, \tag{1.2}$$

where x and b are partitioned according to the size of the blocks of A. This partition may arise naturally due to the structure of the problem or it may be obtained using some block partitioning algorithm; see e.g., [35].

Classical block iterative methods can be used for the solution of (1.2). The advantages over point methods lie in the adaptability of these methods for parallel processing and, generally, in faster convergence. Descriptions of these methods can be found, e.g., in Berman and Plemmons [2], Ortega [37], [38] or Varga [47]. Particularly, in the Block-Jacobi type methods we use a splitting $A = M - N$ (i.e., M nonsingular), where M is a block diagonal matrix, denoted by

$$M = \mathrm{diag}(M_1, \ldots, M_j, \ldots, M_r), \tag{1.3}$$

and the blocks M_j are of order n_j, $1 \le j \le r$. With this notation we have the following algorithm.

*Received July 29, 1999. Accepted for publication March 22, 2001. Recommended by D. B. Szyld.

†Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, E-03071 Alicante, Spain. This research was supported by Spanish DGESIC grant number PB98-0977. E-mail: chus@dccia.ua.es, violeta@dccia.ua.es, jpenades@dccia.ua.es

ALGORITHM 1. (BLOCK-JACOBI TYPE).

Given an initial vector $x^{(0)} = \left((x_1^{(0)})^T, \ldots, (x_r^{(0)})^T\right)^T$.

For $l = 0, 1, 2, \ldots$, until convergence.

For $j = 1$ to r

$$M_j x_j^{(l+1)} = (Nx^{(l)} + b)_j. \tag{1.4}$$

Note that in the standard Block-Jacobi method, the block diagonal matrix M, defined in (1.3), consists of the diagonal blocks of A in (1.2).

At each iteration $l,\ l = 0, 1, 2, \ldots$, of a Block-Jacobi type method, r independent linear systems of the form (1.4) need to be solved; therefore each linear system (1.4) can be solved by a different processor. However, when the order of the diagonal blocks $M_j,\ 1 \le j \le r$, is large, it is natural to approximate their solutions by using an iterative method, and thus we are in the presence of a two-stage iterative method; see e.g., [16], [26], [29], [33]. In a formal way, for a block two-stage method, let us consider the splittings

$$M_j = F_j - G_j, \quad 1 \le j \le r, \tag{1.5}$$

and at each *outer* lth iteration perform, for each $j,\ 1 \le j \le r$, $q(l,j)$ *inner* iterations of the iterative procedure defined by the splittings (1.5) to approximate the solution of (1.4); i.e., the following algorithm is performed.

ALGORITHM 2. (BLOCK TWO-STAGE).

Given an initial vector $x^{(0)} = \left((x_1^{(0)})^T, \ldots, (x_r^{(0)})^T\right)^T$, and a sequence of numbers of inner iterations $q(l,j),\ 1 \le j \le r,\ l = 0, 1, 2, \ldots$.

For $l = 0, 1, 2, \ldots$, until convergence.

For $j = 1$ to r

$y_j^{(0)} = x_j^{(l)}$

For $k = 1$ to $q(l,j)$

$$F_j y_j^{(k)} = G_j y_j^{(k-1)} + (Nx^{(l)} + b)_j \tag{1.6}$$

$$x^{(l+1)} = \left((y_1^{(q(l,1))})^T, (y_2^{(q(l,2))})^T, \ldots, (y_r^{(q(l,r))})^T\right)^T.$$

When the number of inner iterations $q(l,j)$ used to approximate each of the linear systems (1.4) is the same for each $j,\ 1 \le j \le r$, and for each outer step $l = 0, 1, 2, \ldots$, it is said that the method is *stationary*, while a *non-stationary* block two-stage method is such that a different number of inner iterations may be performed in each block and/or each outer step; see e.g., [6] and [17]. Convergence properties of these algorithms were studied when the number of inner iterations becomes sufficiently large. Furthermore, convergence for monotone matrices and H–matrices was shown for any number of inner iterations; see e.g., Berman and Plemmons [2] and Ostrowski [39] for definitions. Parallel generalizations of those block two-stage methods, called two-stage multisplitting methods, have been analyzed by Bru, Migallón, Penadés and Szyld [8]; see also, [6], [40] and [45]. Convergence was shown under conditions similar to those for the block methods.

The multisplitting technique was introduced by O'Leary and White [34] and was further studied by other authors, e.g., Frommer and Mayer [12], [13], Neumann and Plemmons [32], White [48], [49], [50], Szyld and Jones [46], Mas, Migallón, Penadés and Szyld [28] and Fuster, Migallón and Penadés [18]. This method consists of having a collection of splittings

$$A = P_j - Q_j, \quad 1 \le j \le r, \tag{1.7}$$

and diagonal nonnegative weighting matrices E_j which add to the identity, and the following iteration is performed

$$x^{(l+1)} = \sum_{j=1}^{r} E_j P_j^{-1} Q_j x^{(l)} + \sum_{j=1}^{r} E_j P_j^{-1} b, \quad l = 0, 1, 2, \ldots, \tag{1.8}$$

where $x^{(0)}$ is an arbitrary initial vector. As it can be appreciated, Algorithm 1 can be seen as a special case of the iterative scheme (1.8) when all the splittings (1.7) are the same, with $P_j = M = \mathrm{diag}(M_1, \ldots, M_r)$ and the diagonal matrices E_j have ones in the entries corresponding to the diagonal block M_j and zeros otherwise. As in the case of the Block-Jacobi type methods, at each iteration l of (1.8), r independent linear systems need to be solved. When these linear systems are not solved exactly, but rather their solutions approximated by using iterative methods based on splittings of the form

$$P_j = B_j - C_j, \quad 1 \leq j \leq r, \tag{1.9}$$

we obtain a two-stage multisplitting method, which corresponds to the following algorithm; see [8].

ALGORITHM 3. (TWO-STAGE MULTISPLITTING).

Given an initial vector $x^{(0)}$, and a sequence of numbers of inner iterations $q(l, j)$, $1 \leq j \leq r$, $l = 0, 1, 2, \ldots$.

For $l = 0, 1, 2, \ldots$, until convergence.
For $j = 1$ to r
$y_j^{(0)} = x^{(l)}$
For $k = 1$ to $q(l, j)$

$$B_j y_j^{(k)} = C_j y_j^{(k-1)} + (Q_j x^{(l)} + b) \tag{1.10}$$

$$x^{(l+1)} = \sum_{j=1}^{r} E_j y_j^{(q(l,j))} .$$

Note that Algorithm 2 can be seen as a particular case of Algorithm 3, setting $P_j = M$, $1 \leq j \leq r$, where M is the block diagonal matrix defined in (1.3), $B_j = \mathrm{diag}(F_1, \ldots, F_r)$, with F_j, $1 \leq j \leq r$, defined in (1.5), and E_j, $1 \leq j \leq r$, are block matrices partitioned according to the size of the blocks of A, with the jth diagonal block equal to the identity and zeros elsewhere.

On the other hand, Algorithm 3 reduces to the multisplitting method (1.8) when the inner splittings are $P_j = P_j - O$ and $q(l, j) = 1$, $1 \leq j \leq r$, $l = 0, 1, 2, \ldots$. Moreover, Model A in Bru, Elsner and Neumann [4] is a special case of Algorithm 3 when the outer splittings (1.7) are all $A = A - O$. The convergence of that Model A has been established for monotone matrices [4], H–matrices [28], and Hermitian positive definite matrices [9].

In this paper we concentrate our study on Algorithms 2 and 3 when the coefficient matrix of the linear system (1.1) is Hermitian positive definite. Furthermore, we investigate the case when the number of inner iterations becomes sufficiently large. Convergence of Algorithm 2 together with its generalization to the two-stage multisplitting Algorithm 3 is analyzed in §3. We show that the convergence properties of Algorithm 2 cannot always be extended to Algorithm 3. In §4 we study monotonicity results for the two-stage methods. Finally, in §5 we give some numerical results on distributed memory multicomputers, using these block two-stage methods not only as iterative methods but also as preconditioners for the conjugate gradient method. In §2, we present some definitions and preliminaries used later in the paper.

2. Notation and preliminaries. The transpose and the conjugate transpose of a matrix $A \in C^{n\times n}$ are denoted by A^T and A^H, respectively. Similarly, given a vector $x \in C^n$, x^T and x^H denote the transpose and the conjugate transpose of x, respectively. A matrix $A \in C^{n\times n}$ is said to be symmetric if $A = A^T$, and Hermitian if $A = A^H$. Clearly, a real symmetric matrix is a particular case of a Hermitian matrix. A complex, not necessarily Hermitian matrix A, is called positive definite (positive semidefinite) if the real part of $x^H Ax$ is positive (nonnegative), for all complex $x \neq 0$. When A is Hermitian, this is equivalent to requiring that $x^H Ax > 0$ ($x^H Ax \geq 0$), for all complex $x \neq 0$. We use the notation $A \succ O$ ($A \succeq O$) for a matrix to be Hermitian positive definite (Hermitian positive semidefinite). In addition, a general matrix A is positive definite (positive semidefinite) if and only if the Hermitian matrix $A + A^H$ is positive definite (positive semidefinite). Given a matrix $A \in C^{n\times n}$, the splitting $A = M - N$ is called P–regular if the matrix $M^H + N$ is positive definite. For any Hermitian positive definite matrix $A \in C^{n\times n}$, $\langle x, y\rangle = x^H Ay$ defines an inner product on C^n. Thus, $\|x\|_A = (x^H Ax)^{1/2}$ is a vector norm on C^n. The matrix norm induced by that vector norm will also be denoted by $\|\cdot\|_A$. Furthermore, by $\|\cdot\|_\infty$ we denote the infinite matrix norm; see e.g., [2], [37], [38], for an extensive bibliography on Hermitian matrices and positive definite matrices.

THEOREM 2.1. *Let $A = M - N$ be a P–regular splitting of a Hermitian matrix A. Then $\rho(M^{-1}N) < 1$ if and only if A is positive definite.*

Proof. The proof of this theorem can be found, e.g., in Berman and Plemmons [2, Corollary 7.5.44] and Keller [25, Theorem 3]. ◻

LEMMA 2.2. *Given a nonsingular matrix A, and a matrix T such that $(I - T)^{-1}$ exists, there is a unique pair of matrices P, Q such that P is nonsingular, $T = P^{-1}Q$ and $A = P - Q$. The matrices are $P = A(I - T)^{-1}$ and $Q = P - A$.*

Proof. See Lemma 8 of Lanzkron, Rose and Szyld [26]. ◻

LEMMA 2.3. *Let $A \succ O$. Assume the splitting $A = B - C$ is P–regular. Given $s \geq 1$, there exists a unique splitting $A = P - Q$ such that $(B^{-1}C)^s = P^{-1}Q$. Moreover, the splitting is P–regular.*

Proof. This result is Lemma 3.1 of [9]. ◻

LEMMA 2.4. *Let $T^{(l)}$, $l = 0, 1, 2, \ldots$, be a sequence of square complex matrices. If there exists a matrix norm $\|\cdot\|$ such that $\|T^{(l)}\| \leq \theta < 1$, $l = 0, 1, 2, \ldots$, then $\lim_{l\to\infty} T^{(l)}T^{(l-1)}\cdots T^{(0)} = O$.*

Proof. See Lemma 2 of Bru and Fuster [5]. ◻

THEOREM 2.5. *Let $A, B \in C^{n\times n}$ be Hermitian matrices. If either is positive definite, then AB has real eigenvalues and is diagonalizable. If both A and B are positive definite, then the eigenvalues of AB are positive. Conversely, if AB has positive eigenvalues and either A or B is positive definite, then both are positive definite.*

Proof. See Theorem 6.2.3 of [37]. ◻

3. Convergence. In order to analyze the convergence of the block two-stage method and its generalization to the two-stage multisplitting method, we write Algorithm 3 as the following iteration.

$$x^{(l+1)} = \sum_{j=1}^{r} E_j \left[(B_j^{-1}C_j)^{q(l,j)}x^{(l)} + \sum_{i=0}^{q(l,j)-1} (B_j^{-1}C_j)^i B_j^{-1} \left(Q_j x^{(l)} + b\right)\right], \tag{3.1}$$

cf. [8]. Let ξ be the exact solution of (1.1) and let $\epsilon^{(l+1)} = x^{(l+1)} - \xi$ be the error at the $l+1$ iteration. It is easy to prove that ξ is a fixed point of (3.1). Thus

$$\epsilon^{(l+1)} = T^{(l)}\epsilon^{(l)} = \ldots = T^{(l)}T^{(l-1)}\cdots T^{(0)}\epsilon^{(0)}, \quad l = 0, 1, 2, \ldots,$$

ETNA
Kent State University
etna@mcs.kent.edu

where $T^{(l)}$ are the iteration matrices

$$T^{(l)} = \sum_{j=1}^{r} E_j \left[(B_j^{-1}C_j)^{q(l,j)} + \sum_{i=0}^{q(l,j)-1} (B_j^{-1}C_j)^i B_j^{-1} Q_j \right],$$

or equivalently

$$T^{(l)} = \sum_{j=1}^{r} E_j \left[(B_j^{-1}C_j)^{q(l,j)} + \left(I - (B_j^{-1}C_j)^{q(l,j)} \right) P_j^{-1} Q_j \right], \tag{3.2}$$

cf. [8]. Thus, the sequence of error vectors $\{\epsilon^{(l)}\}_{l=0}^{\infty}$ generated by iteration (3.1) converges to the null vector if and only if $\lim_{l\to\infty} T^{(l)}T^{(l-1)}\cdots T^{(0)} = O$. Since the product of convergent matrices is not necessarily convergent, see e.g., Johnson and Bru [23], or Robert, Charnay and Musy [42], the convergence of Algorithm 3 needs tools other than the spectral radius for its analysis.

We first analyze the convergence of Algorithm 3 for any convergent outer and inner splittings, and requiring that enough inner iterations are performed.

THEOREM 3.1. *Consider the nonsingular linear system (1.1). Suppose that the outer splittings (1.7) are all $A = P - Q$, such that $\rho(P^{-1}Q) < 1$. Let $P = B_j - C_j$, $1 \le j \le r$, be convergent splittings. If $\lim_{l\to\infty} q(l,j) = \infty$, $1 \le j \le r$, then the two-stage multisplitting Algorithm 3 converges to the solution of the linear system (1.1), for any initial vector $x^{(0)}$.*

Proof. Since $A = P - Q$ is a convergent splitting, there exists an induced matrix norm $\|\cdot\|$ such that $\rho(P^{-1}Q) \le \|P^{-1}Q\| < 1$. Moreover, since $\rho(B_j^{-1}C_j) < 1$, and $\lim_{l\to\infty} q(l,j) = \infty$, then $\lim_{l\to\infty} (B_j^{-1}C_j)^{q(l,j)} = O$, $1 \le j \le r$. Therefore, this implies $\lim_{l\to\infty} \sum_{j=1}^{r} E_j (B_j^{-1}C_j)^{q(l,j)} = O$. Hence, for all $\epsilon > 0$ there exists an index l_0 such that $\| \sum_{j=1}^{r} E_j (B_j^{-1}C_j)^{q(l,j)} \| \le \epsilon$, for all $l \ge l_0$. Then, for $l \ge l_0$, from (3.2) we obtain

$$\begin{aligned} \|T^{(l)}\| &= \| \sum_{j=1}^{r} E_j (B_j^{-1}C_j)^{q(l,j)} + (I - \sum_{j=1}^{r} E_j (B_j^{-1}C_j)^{q(l,j)}) P^{-1}Q \| \\ &\le \epsilon + (1+\epsilon)\|P^{-1}Q\| = \alpha_\epsilon. \end{aligned}$$

Setting $\epsilon < \frac{1-\|P^{-1}Q\|}{1+\|P^{-1}Q\|}$, we have $\|T^{(l)}\| \le \alpha_\epsilon < 1$. Hence, from Lemma 2.4, the convergence is proved. $\square$

Theorem 3.1 may be regarded as an extension of Theorem 2.4 of [16]. Moreover, it generalizes Theorem 3.1 of [8], when all the outer splittings in Algorithm 3 are the same, and also Theorem 1 of [7]. Specifically, the assumption on the outer splitting $A = P - Q$, in [7] and [8], was $\|P^{-1}Q\|_\infty < 1$. Here we weaken that assumption by the more general $\rho(P^{-1}Q) < 1$. We point out that the formulation of Algorithm 3, with all the outer splittings being the same, allows us to include not only the block two-stage Algorithm 2 but methods with overlapping in the above convergence result; see e.g., [14], [15], [24]. However, when there are different outer splittings, Theorem 3.1 may not be true as we illustrate with the following example.

EXAMPLE 3.2. Consider the matrix

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix},$$

and the splittings $A = P_1 - Q_1 = P_2 - Q_2$, where

$$P_1 = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad P_2 = \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}.$$

Since $P_1^{-1}Q_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $P_2^{-1}Q_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, then both splittings are convergent. Furthermore, consider the trivial inner splittings $P_1 = P_1 - O$ and $P_2 = P_2 - O$. Setting $E_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $E_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, for any number of inner iterations the iteration matrices of Algorithm 3 are

$$T^{(l)} = E_1P_1^{-1}Q_1 + E_2P_2^{-1}Q_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad l = 0, 1, 2, \ldots,$$

which are not convergent.

We recall, as was shown in [8], that Theorem 3.1 holds for different outer splittings $A = P_j - Q_j,\ 1 \le j \le r$, with the additional hypothesis $\|P_j^{-1}Q_j\| < 1$, where $\|\cdot\|$ denotes a weighted max-norm associated with a positive vector such as the infinite norm; see e.g., [22], [41].

REMARK 3.3. As an immediate consequence of Theorem 3.1 we obtain the convergence of Algorithm 3, and therefore of Algorithm 2, applied to Hermitian positive definite matrices, when all outer splittings are the same, and both inner and outer splittings are P-regular.

On the other hand, numerical experiments reported by some authors, see e.g., [6], [7], [8], show that often few inner iterations produce good overall convergence results. In the rest of this section we analyze the convergence of algorithms 2 and 3 for any number of inner iterations. When A is Hermitian and positive definite, algorithms 2 and 3 may not converge even if the splittings are P–regular. We give here an example that illustrates this situation.

EXAMPLE 3.4. Consider the symmetric positive definite matrix

$$A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix},$$

and let the P–regular splitting $A = P - Q$ be given by

$$P = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Note that the matrix Q is not positive semidefinite. Consider further the P–regular splittings of $P = B_1 - C_1 = B_2 - C_2$, where

$$B_1 = B_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad C_1 = C_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Setting $E_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $E_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, a simple calculation shows that the iteration matrices of the two-stage multisplitting method with $q(l,j) = 1,\ j = 1, 2,\ l = 0, 1, 2, \ldots,$ are all

$$T^{(l)} = \begin{bmatrix} -0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}, \ l = 0, 1, 2, \ldots,$$

that have spectral radius equal to 1, and thus are not convergent.

THEOREM 3.5. *Consider the linear system (1.2), with $A \succ O$. Let $A = M - N$, where $M = \mathrm{diag}(M_1, \ldots, M_r)$ is the block diagonal matrix defined in (1.3). Suppose that M is Hermitian and N is positive semidefinite. Let $M_j = F_j - G_j$, $1 \le j \le r$, be P–regular splittings. Assume that the sequences of inner iterations $q(l, j)$, $l = 0, 1, 2, \ldots,$ $1 \le j \le r$, are bounded. Then, the block two-stage Algorithm 2 converges to the solution of the linear system (1.2), for any initial vector $x^{(0)}$.*

Proof. In order to analyze the convergence of the block two-stage Algorithm 2, let us denote

$$H^{(l)} = \mathrm{diag}\left((F_1^{-1}G_1)^{q(l,1)}, \ldots, (F_r^{-1}G_r)^{q(l,r)} \right),\ l = 0, 1, 2, \ldots. \tag{3.3}$$

Then, from (3.2), the iteration matrices of Algorithm 2 can be written as follows

$$T^{(l)} = H^{(l)} + (I - H^{(l)})M^{-1}N. \tag{3.4}$$

By hypotheses, $M_j = F_j - G_j$ is a P–regular splitting of the Hermitian positive definite matrix M_j, $1 \le j \le r$. Then, from Theorem 2.1, $\rho(F_j^{-1}G_j) < 1$, $1 \le j \le r$, and therefore $I - (F_j^{-1}G_j)^{q(l,j)}$ is a nonsingular matrix, for all j, l, $1 \le j \le r$, $l = 0, 1, 2, \ldots$. Thus, for each j, l, from Lemma 2.2, there exists a unique pair of matrices $P_j^{(l)} = M_j(I - (F_j^{-1}G_j)^{q(l,j)})^{-1}$ and $Q_j^{(l)} = P_j^{(l)} - M_j$ such that $(F_j^{-1}G_j)^{q(l,j)} = (P_j^{(l)})^{-1}Q_j^{(l)}$. Moreover, from Lemma 2.3, the splitting $M_j = P_j^{(l)} - Q_j^{(l)}$ is also P–regular.
For each l, $l = 0, 1, 2, \ldots,$ let us consider the matrices

$$P^{(l)} = \mathrm{diag}(P_1^{(l)}, \ldots, P_r^{(l)}),\ \ Q^{(l)} = \mathrm{diag}(Q_1^{(l)}, \ldots, Q_r^{(l)}).$$

Clearly, $M = P^{(l)} - Q^{(l)}$ is a P–regular splitting and $P^{(l)} = M(I - H^{(l)})^{-1}$.
From (3.4), it follows that

$$\begin{aligned} T^{(l)} &= I - (I - H^{(l)})(I - M^{-1}N) \\ &= I - (I - H^{(l)})M^{-1}A = I - (P^{(l)})^{-1}A. \end{aligned} \tag{3.5}$$

Thus, it is easy to see that

$$\begin{aligned} A - (T^{(l)})^H A T^{(l)} &= \left((P^{(l)})^{-1}A \right)^H \left[(P^{(l)})^H + P^{(l)} - A \right] \left((P^{(l)})^{-1}A \right) \\ &= \left((P^{(l)})^{-1}A \right)^H \left[(P^{(l)})^H + Q^{(l)} + N \right] \left((P^{(l)})^{-1}A \right). \end{aligned} \tag{3.6}$$

Since $(P^{(l)})^H + Q^{(l)}$ is positive definite and N is positive semidefinite, from (3.6) it follows that the Hermitian matrix $A - (T^{(l)})^H A T^{(l)}$ is positive definite. Then, using the vector norm $\| \cdot \|_A$ we obtain

$$\|T^{(l)}x\|_A^2 = x^H (T^{(l)})^H A T^{(l)} x < x^H A x = \|x\|_A^2, \ \text{ for all } x \ne 0,\ \ l = 0, 1, 2, \ldots.$$

Thus, $\|T^{(l)}\|_A < 1$, $l = 0, 1, 2, \ldots$. Since the sequences $\{q(l, j)\}_{l=0}^{\infty}$, $1 \le j \le r$, are bounded, there is a finite number of different iteration matrices and therefore there exists a real constant $0 \le \theta < 1$ such that $\|T^{(l)}\|_A \le \theta < 1$, $l = 0, 1, 2, \ldots$. Then, from Lemma 2.4, the proof is completed. □

We point out that in Theorem 3.5, the hypotheses on the outer splitting $A = M - N$ imply that this splitting is P–regular. However, Example 3.4 shows that the P–regularity of the outer

ETNA
Kent State University
etna@mcs.kent.edu

splitting $A = M - N$ alone does not guarantee convergence, when N is not positive semidefinite. On the other hand, the hypothesis on the sequences of inner iterations in Theorem 3.5 is very realistic in practice, since there is always a maximum number of inner iterations in each block. Nevertheless, following [5] this condition can be weakened by the assumption that there exists a subsequence $\{l_k\}_{k=0}^{\infty}$ such that the sequences $\{q(l_k, j)\}_{k=0}^{\infty}$, $1 \le j \le r$, are bounded.

On the other hand, the proof of Theorem 3.5 shows that each iteration matrix of the block two-stage Algorithm 2 is induced by a unique P–regular splitting. We establish this result in the following corollary.

COROLLARY 3.6. *Consider the linear system (1.2), with $A \succ O$. Let $A = M - N$, where $M = \operatorname{diag}(M_1, \dots, M_r)$ is the block diagonal matrix defined in (1.3). Suppose that M is Hermitian and N is positive semidefinite. Let $M_j = F_j - G_j$, $1 \le j \le r$, be P–regular splittings. Then, the unique splitting induced by each iteration matrix $T^{(l)}$, $l = 0, 1, 2, \dots$, of the block two-stage Algorithm 2 is P–regular.*

Proof. From the proof of Theorem 3.5, $\rho(T^{(l)}) < 1$, $l = 0, 1, 2, \dots$. Then, for each l, from Lemma 2.2, there exists a unique pair of matrices $B_l = A(I - T^{(l)})^{-1}$ and $C_l = B_l - A$ such that $T^{(l)} = B_l^{-1} C_l$. From (3.5), $B_l = A(I - T^{(l)})^{-1} = M(I - H^{(l)})^{-1}$, where $H^{(l)}$ is given in (3.3). Now, following the proof of Theorem 3.5, the matrix $B_l^H + C_l = B_l^H + B_l - A$ is positive definite. □

Convergence results of Theorem 3.5 are based on P–regular splittings of a Hermitian matrix. It is well-known when the Jacobi, Gauss-Seidel and SOR splittings (and their blocks versions) of a Hermitian positive definite matrix are P–regular; see e.g., [2], [36]. Another class of P–regular splittings is given by the unique splitting induced by the iteration matrix of an alternating method, such as the SSOR method, based on two P–regular splittings; see [1]. However, in Theorem 3.5 we have also assumed that in the outer splitting $A = M - N$, N is positive semidefinite. A simple way to ensure the hypotheses on the outer splitting in that theorem is done as follows. Let $A = \bar{M} - \bar{N}$ be the Block-Jacobi splitting of A, i.e., $\bar{M} = \operatorname{diag}(A_{11}, \dots, A_{rr})$. Let us consider square diagonal nonnegative matrices D_j, of size n_j, $1 \le j \le r$, such that $\bar{N} + \operatorname{diag}(D_1, \dots, D_r)$ is positive semidefinite. Then the splitting $A = M - N$, where

$$M = \operatorname{diag}(M_1, \dots, M_r), \quad M_j = A_{jj} + D_j, \quad N = \bar{N} + \operatorname{diag}(D_1, \dots, D_r), \tag{3.7}$$

satisfies the assumptions of Theorem 3.5.

Another way to get splittings satisfying the hypotheses of that theorem consists in using a relaxation parameter ω in the outer splitting so that the new splitting will be $A = \frac{1}{\omega} M - (\frac{1-\omega}{\omega} M + N)$. If $A = M - N$ is a P-regular splitting, for certain values of ω ($0 < \omega \le 0.5$) we obtain the assumptions needed in Theorem 3.5.

THEOREM 3.7. *Consider the linear system (1.1), with $A \succ O$. Let $A = P_j - Q_j$, $1 \le j \le r$, where P_j is Hermitian and Q_j is positive semidefinite. Let $P_j = B_j - C_j$, $1 \le j \le r$, be P–regular splittings and $E_j = \alpha_j I$, $1 \le j \le r$, with $\alpha_j > 0$, $\sum_{j=1}^{r} \alpha_j = 1$. Assume further that the sequences of numbers of inner iterations $q(l, j)$, $l = 0, 1, 2, \dots$, $1 \le j \le r$, are bounded. Then, the two-stage multisplitting Algorithm 3 converges to the solution of the linear system (1.1), for any initial vector $x^{(0)}$.*

Proof. Define $T_j^{(l)} = (B_j^{-1} C_j)^{q(l,j)} + \left(I - (B_j^{-1} C_j)^{q(l,j)}\right) P_j^{-1} Q_j$, $l = 0, 1, 2, \dots$, $1 \le j \le r$. For each l, j, $T_j^{(l)}$ is the iteration matrix of a two-stage iterative method to solve the linear system (1.1), where the outer splitting is $A = P_j - Q_j$, the inner splitting is $P_j = B_j - C_j$, and $q(l, j)$ is the number of inner iterations, cf. [33]. Then, from Corollary 2.1

of [29], the unique splitting $A = R_j^{(l)} - S_j^{(l)}$ induced by the iteration matrix $T_j^{(l)}$ is P–regular. Thus, we can write the iteration matrices $T^{(l)}$ defined in (3.2) as follows.

$$T^{(l)} = \sum_{j=1}^{r} E_j T_j^{(l)}, \quad \text{with} \quad T_j^{(l)} = (R_j^{(l)})^{-1} S_j^{(l)}.$$

For each $l,\ l = 0, 1, 2, \ldots$, the matrix $T^{(l)}$ can be viewed as the iteration matrix of a multisplitting method using the P–regular splittings $A = R_j^{(l)} - S_j^{(l)},\ 1 \le j \le r$, and weighting matrices $E_j,\ 1 \le j \le r$. Hence, from Theorem 3.2 of [31], if $E_j = \alpha_j I$, the unique splitting $A = U^{(l)} - V^{(l)}$ induced by each iteration matrix $T^{(l)}$ is also P–regular, and then $\rho(T^{(l)}) < 1,\ l = 0, 1, 2, \ldots$. Taking into account that $T^{(l)} = (U^{(l)})^{-1} V^{(l)}$, it is easy to see that, for $l = 0, 1, 2, \ldots$,

$$A - (T^{(l)})^H A T^{(l)} = \left((U^{(l)})^{-1} A\right)^H \left[(U^{(l)})^H + U^{(l)} - A\right] \left((U^{(l)})^{-1} A\right),$$

and then, since the splittings $A = U^{(l)} - V^{(l)}$ are P–regular, the Hermitian matrices $A - (T^{(l)})^H A T^{(l)}$ are positive definite, i.e.,

$$x^H \left[A - (T^{(l)})^H A T^{(l)}\right] x = x^H A x - (T^{(l)} x)^H A T^{(l)} x > 0, \text{ for all } x \ne 0.$$

Then $\|T^{(l)} x\|_A < \|x\|_A$, for all $x \ne 0$, and therefore $\|T^{(l)}\|_A < 1,\ l = 0, 1, 2, \ldots$. Since the sequences of number of inner iterations $q(l, j),\ l = 0, 1, 2, \ldots,\ 1 \le j \le r$, are bounded, there is a finite number of different iteration matrices $T^{(l)}$. Thus, there exists a real constant θ, $0 \le \theta < 1$, such that $\|T^{(l)}\|_A \le \theta < 1,\ l = 0, 1, 2, \ldots$, and, from Lemma 2.4, the proof is completed. ∎

We point out that under the hypotheses of the above theorem, each iteration matrix of the two-stage multisplitting Algorithm 3 is induced by a unique P–regular splitting. On the other hand, the assumption on the weighting matrices is very restrictive in practice because it forces each processor to update all components of the actual iterate. This result can be seen more as a theoretical result than as a computational recipe. The following example shows that the hypothesis on the weighting matrices in Theorem 3.7, cannot be weakened as was done in Theorem 3.5 for Algorithm 2.

EXAMPLE 3.8. Consider the symmetric positive definite matrix

$$A = \begin{bmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{bmatrix},$$

and the outer splittings $A = P_1 - Q_1 = P_2 - Q_2$, with $P = P_1 = P_2,\ Q = Q_1 = Q_2$, where

$$P = \begin{bmatrix} 0.75 & 0 \\ 0 & 0.75 \end{bmatrix}, \ Q = \begin{bmatrix} 0.25 & -0.25 \\ -0.25 & 0.25 \end{bmatrix}.$$

Note that P is Hermitian and Q is positive semidefinite. Consider also the P–regular inner splittings $P_1 = B_1 - C_1$, and $P_2 = B_2 - C_2$, where

$$B_1 = \begin{bmatrix} 4 & 1 \\ -1 & 0.5 \end{bmatrix}, \text{ and } B_2 = \begin{bmatrix} 0.5 & -1 \\ 1 & 4 \end{bmatrix}.$$

ETNA
Kent State University
etna@mcs.kent.edu

Setting $E_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $E_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, and for all $l = 0, 1, 2, \ldots,\ j = 1, 2,\ q(l, j) = 1$, the iteration matrices of Algorithm 3 are

$$T^{(l)} = \sum_{j=1}^{2} E_j \left[B_j^{-1}C_j + \left(I - B_j^{-1}C_j\right) P^{-1}Q\right] = \begin{bmatrix} 1 & 0.125 \\ 0.125 & 1 \end{bmatrix},\ l = 0, 1, 2, \ldots,$$

which have spectral radius equal to 1.125, and therefore Algorithm 3 is not convergent.

We finish this section with a few remarks about the use of relaxation parameters in Algorithms 2 and 3. A relaxation parameter $\omega \neq 0$ can be introduced in Algorithm 2 and replace the computation of $y_j^{(k)}$ in (1.6) with the equations

$$F_j y_j^{(k)} = \omega \left(G_j y_j^{(k-1)} + (Nx^{(l)} + b)_j\right) + (1 - \omega) F_j y_j^{(k-1)}.$$

This is equivalent to replacing the splittings (1.5) by

$$M_j = \frac{1}{\omega} F_j - \left(\frac{1 - \omega}{\omega} F_j + G_j\right),\quad 1 \leq j \leq r.$$

In the case of $\omega \neq 1$, we have a relaxed block two-stage method. It is easy to show that, if we assume $0 < \omega \leq 1$, Theorem 3.5 and Corollary 3.6 hold for that relaxed method. In the same way, we can introduce a relaxation parameter $\omega \neq 0$ in Algorithm 3 by replaying the computation of $y_j^{(k)}$ in (1.10) with the equations

$$B_j y_j^{(k)} = \omega \left(C_j y_j^{(k-1)} + Q_j x^{(l)} + b\right) + (1 - \omega) B_j y_j^{(k-1)}.$$

With $0 < \omega \leq 1$, Theorem 3.7 holds for the relaxed two-stage multisplitting algorithm. If, in addition, we assume $0 < \omega < 2/(1 + \rho)$, with $\rho = \max_{1 \leq j \leq r} \rho(B_j^{-1} C_j)$, Theorem 3.1 also holds for that relaxed algorithm.

4. Monotonicity. In this section we give comparison theorems for Algorithms 2 and 3 based on the number of inner iterations performed. For that purpose we use the following result given by Nabben [31]. Here we use the partial order introduced by Löwner [27]; see also e.g., [21]. Let $B, C \in C^{n\times n}$ be Hermitian matrices, then $B \succ C$ ($B \succeq C$) if and only if $B - C$ is positive definite (positive semidefinite).

THEOREM 4.1. *Let $A = M_1 - N_1 = M_2 - N_2$ be two splittings of A, with $A \succ O$. If $O \preceq N_1 \preceq N_2$, then*

$$\rho(M_1^{-1}N_1) \leq \rho(M_2^{-1}N_2) < 1. \tag{4.1}$$

If $O \preceq N_1 \prec N_2$, then

$$\rho(M_1^{-1}N_1) < \rho(M_2^{-1}N_2) < 1. \tag{4.2}$$

Note that with the assumptions of Theorem 4.1 we have $N_1 \prec N_2 \Leftrightarrow M_1 \prec M_2 \Leftrightarrow M_1^{-1} \succ M_2^{-1}$; see e.g., the proof of Corollary 7.7.4 of [21] and the remarks of Theorem 2.4 of [31].

THEOREM 4.2. *Let $A \succ O$. Consider the splitting $A = M - N$, where $M = \mathrm{diag}(M_1, \ldots, M_r)$ is the block diagonal matrix defined in (1.3). Suppose that M is Hermitian, and N is positive semidefinite. Let $M_j = F_j - G_j$, $1 \leq j \leq r$, such that F_j is*

Hermitian and G_j is positive definite. Let $q_1(j),\ q_2(j),\ 1 \leq j \leq r$, be positive integers. Consider further two block two-stage methods differing only in the number of inner iterations of each block, staying fixed at each outer iteration, $q_1(j),\ 1 \leq j \leq r$, in one case and $q_2(j),\ 1 \leq j \leq r$, in the other. Let T_1, T_2 be the iteration matrices of Algorithm 2 with $q_1(j)$ and $q_2(j),\ 1 \leq j \leq r$, inner iterations, respectively. If $q_1(j) > q_2(j),\ 1 \leq j \leq r$, then $\rho(T_1) < \rho(T_2) < 1$.

Proof. Let us consider, for $k = 1, 2$

$$T_k = H_k + (I - H_k)M^{-1}N, \tag{4.3}$$

with $H_k = \operatorname{diag}\left((F_1^{-1}G_1)^{q_k(1)}, \ldots, (F_r^{-1}G_r)^{q_k(r)}\right)$. Note that T_k is the iteration matrix of a two-stage iterative method defined by Algorithm 2 when the number of inner iterations are $q_k(j),\ 1 \leq j \leq r$. From Corollary 3.6 the unique splitting $A = M_{T_k} - N_{T_k}$, induced by the iteration matrix T_k is P–regular, and $M_{T_k} = M(I - H_k)^{-1}$. Then, the matrix M_{T_k} and therefore the matrix $M_{T_k}^{-1}$ are positive definite. On the other hand, it is easy to see that

$$M_{T_k}^{-1} = \operatorname{diag}\left(\sum_{i=0}^{q_k(1)-1} (F_1^{-1}G_1)^i F_1^{-1}, \ldots, \sum_{i=0}^{q_k(r)-1} (F_r^{-1}G_r)^i F_r^{-1}\right).$$

Since M_j and F_j are Hermitian, $G_j = F_j - M_j,\ 1 \leq j \leq r$, is also Hermitian. Then, it is easy to see that $\displaystyle\sum_{i=0}^{q_k(j)-1} (F_j^{-1}G_j)^i F_j^{-1}$ is Hermitian. Thus the block diagonal matrix $M_{T_k}^{-1}$, and therefore M_{T_k}, and $N_{T_k} = M_{T_k} - A$ are Hermitian matrices.

Next, let us consider the Hermitian matrix $N_{T_k} = M_{T_k}T_k$. Clearly, this matrix can be written as

$$N_{T_k} = M_{T_k}H_k + N. \tag{4.4}$$

Since N and N_{T_k} are Hermitian, the matrix $M_{T_k}H_k$ is also Hermitian. Moreover, from Theorem 2.5, $(F_j^{-1}G_j)^{q_k(j)},\ 1 \leq j \leq r$, and therefore H_k have positive eigenvalues. Since $M_{T_k}^{-1}$ is a Hermitian positive definite matrix, again from Theorem 2.5 it follows that $M_{T_k}H_k$ is positive definite. Then, from (4.4) it follows that N_{T_k} is positive definite. Now suppose that $q_1(j) > q_2(j)$, and consider the Hermitian matrix $M_{T_1}^{-1} - M_{T_2}^{-1}$. Clearly

$$M_{T_1}^{-1} - M_{T_2}^{-1} = \operatorname{diag}\left(\sum_{i=q_2(1)}^{q_1(1)-1} (F_1^{-1}G_1)^i F_1^{-1}, \ldots, \sum_{i=q_2(r)}^{q_1(r)-1} (F_r^{-1}G_r)^i F_r^{-1}\right).$$

It is easy to see that $\displaystyle\sum_{i=q_2(j)}^{q_1(j)-1} (F_j^{-1}G_j)^i,\ 1 \leq j \leq r$, has positive eigenvalues. Then, since $F_j,\ 1 \leq j \leq r$, is Hermitian positive definite, from Theorem 2.5 it follows that $\displaystyle\sum_{i=q_2(j)}^{q_j(1)-1} (F_j^{-1}G_j)^i F_j^{-1},\ 1 \leq j \leq r$, is positive definite and therefore the block diagonal matrix $M_{T_1}^{-1} - M_{T_2}^{-1}$ is also positive definite. Then, from Theorem 4.1, the proof is complete. □

The result of the above theorem seems intuitive but, as we illustrated in Example 1 of [30], if the conditions shown are not satisfied, the result may not hold. On the other hand, we remark that if $q_k(j),\ 1 \leq j \leq r,\ k = 1, 2$ are even we can weaken the assumption on the matrix G_j, $1 \le j \le r$, by the P-regularity of the splitting $M_j = F_j - G_j$. With these hypotheses one obtains $\rho(T_1) \le \rho(T_2)$. This is due to the fact that, now, the matrices N_{T_k}, $k = 1, 2$ and $M_{T_1}^{-1} - M_{T_2}^{-1}$ are positive semidefinite.

THEOREM 4.3. *Let $A \succ O$. Consider the splittings $A = P_j - Q_j$, $1 \le j \le r$, such that P_j is Hermitian and Q_j is positive semidefinite. Let $P_j = B_j - C_j$, $1 \le j \le r$, such that B_j is Hermitian and C_j is positive definite. Consider $E_j = \alpha_j I$, $1 \le j \le r$, with $\alpha_j > 0$, $\sum_{j=1}^{r} \alpha_j = 1$. Consider further two two-stage multisplitting methods differing only in the number of inner iterations of each outer splitting, staying fixed at each outer iteration, $q_1(j)$, $1 \le j \le r$, in one case and $q_2(j)$, $1 \le j \le r$, in the other. Let T_1, T_2 be the iteration matrices of Algorithm 3 with $q_1(j)$ and $q_2(j)$, $1 \le j \le r$, inner iterations, respectively. If $q_1(j) > q_2(j)$, $1 \le j \le r$, then $\rho(T_1) < \rho(T_2) < 1$.*

Proof. For $k = 1, 2$ define $T_{k,j} = (B_j^{-1}C_j)^{q_k(j)} + \left(I - (B_j^{-1}C_j)^{q_k(j)}\right) P_j^{-1}Q_j$, $1 \le j \le r$. For each j, $T_{k,j}$ is the iteration matrix of a two-stage method where the outer splitting is $A = P_j - Q_j$, the inner splitting is $P_j = B_j - C_j$, and $q_k(j)$ is the number of inner iterations. Then, from Corollary 2.1 of [29] and Lemma 2.2, the unique splitting $A = R_{k,j} - S_{k,j}$ induced by the iteration matrix $T_{k,j}$ is P–regular. Thus, each iteration matrix T_k, $k = 1, 2$, can be written as

$$T_k = \sum_{j=1}^{r} E_j T_{k,j}, \quad \text{with} \quad T_{k,j} = R_{k,j}^{-1} S_{k,j} \quad \text{and} \quad R_{k,j} = P_j \left(I - (B_j^{-1}C_j)^{q_k(j)}\right)^{-1}.$$

Each matrix T_k can be viewed as the iteration matrix of a multisplitting method using the P–regular splittings $A = R_{k,j} - S_{k,j}$, $1 \le j \le r$, and weighting matrices E_j, $1 \le j \le r$. Moreover, if we suppose $q_1(j) > q_2(j)$, taking into account that $R_{k,j} = \sum_{i=0}^{q_k(j)-1} (B_j^{-1}C_j)^i B_j^{-1}$, and reasoning in a similar way as in Theorem 4.2, it is easy to see that

$$O \preceq S_{1,j} \prec S_{2,j}. \tag{4.5}$$

Hence, from Theorem 3.3 of [31], if $E_j = \alpha_j I$, for $k = 1, 2$, the unique splitting $A = U_k - V_k$ induced by each iteration matrix T_k satisfies $U_k^{-1} = \sum_{j=1}^{r} E_j R_{k,j}^{-1}$, and $V_k = U_k - A$ are Hermitian positive definite matrices. Moreover, from (4.5), it is easy to see that $O \preceq U_1 \prec U_2$ and therefore $O \preceq V_1 \prec V_2$. Then, from Theorem 4.1, the proof is complete. ☐

5. Numerical experiments on distributed memory multiprocessors. In this section we implemented block iterative methods based on Algorithm 2 described in §1, and we use them as preconditioners for the conjugate gradient method. We present results for three model problems. In the first numerical experiments we use two test problems arising from finite element approximations to problems in structural engineering. These problems are included in the Harwell-Boeing sparse matrix collection [11] under the name of LANPRO. Here we use the problems LANPRO (NOS5) and LANPRO (NOS7). The stiffness matrices are symmetric and positive definite of order 468 and 729, respectively.

In the second numerical experiments the problem to be solved comes from the discretization of the Laplace's equation, $\nabla^2 u = u_{ss} + u_{tt} = 0$, satisfying Dirichlet boundary conditions on the unit square $\Omega = [0,1] \times [0,1]$, $u(s,0) = u(0,t) = u(s,1) = 0$, $u(1,t) = 100$, $0 \le s \le 1$, $0 \le t \le 1$. The discretization of the domain Ω, using five point finite differences, with $J \times J$ points equally spaced by h, yields a linear system $Ax = b$, where A is block tridiagonal, $A = \mathrm{tridiag}[-I, C, -I]$, where I and C are $J \times J$ matrices, I is the identity, and $C = \mathrm{tridiag}[-1, 4, -1]$. Note that A has $J \times J$ blocks of size $J \times J$. Clearly, A is a symmetric positive definite matrix. Here we discuss the results for matrices of sizes 4096, 16384, 40000 and 262144, which correspond to square grid sizes of 64, 128, 200 and 512 nodes in each direction, respectively.

The last model problem used can be associated with the deflection $u(s,t)$ of a square plate, clamped at all four sides. If $p(s,t)$ denotes the load on the plate at the point (s,t), this deflection can be modeled using an elliptic partial differential equation which relates the biharmonic operator

$$\nabla^4 = \frac{\partial^4}{\partial s^4} + 2\frac{\partial^4}{\partial s^2 \partial t^2} + \frac{\partial^4}{\partial t^4},$$

with the load on the plate p. More specifically, the deflection $u(s,t)$ satisfies

$$\begin{aligned} &\nabla^4 u = p, && \text{in the interior,} \\ &u = 0,\ \frac{\partial u}{\partial \vec{v}} = 0, && \text{on the boundary.} \end{aligned} \tag{5.1}$$

The discretization for the problem of the clamped plate, using $J \times J$ points equally spaced by h, yields (see e.g., [43]) a linear system $Ax = b$, where A is a block pentadiagonal matrix $A = \mathrm{pentadiag}(I, G, B, G, I)$, where I is the identity, $B = \mathrm{pentadiag}(1, -8, 20, -8, 1)$ and $G = \mathrm{tridiag}(2, -8, 2)$. For this problem we present results with matrices of sizes 4096, 10000 and 40000.

In order to ensure the convergence of Algorithm 2 we define the block outer splitting $A = M - N$ as in (3.7), where $D = \mathrm{diag}\left(\sum_{j=1, j\neq 1}^{n} |\bar{n}_{1j}|, \ldots, \sum_{j=1, j\neq n}^{n} |\bar{n}_{nj}|\right)$. In the experiments reported here, we use as inner iterative procedures the Gauss-Seidel, SOR or SSOR methods. The outer stopping criterion used was $\sum_{i=1}^{n} |x_i^{(l)} - x_i^{(l-1)}| < 10^{-\delta}$. For the LANPRO problems we set $\delta = 4$, while for the Laplace and biharmonic problems we use a δ such that $\tau^T \tau < 10^{-7}$, where τ is the residual at the corresponding iteration. This is done in that way in order to compare the two-stage methods with preconditioned conjugate gradient methods we will present in this section. The right hand side for the LANPRO and biharmonic problems was $b = (1, 1, \ldots, 1)^T$ and for the Laplace problem $b = (b_1^T, b_2^T, \ldots, b_J^T)^T$, $b_i \in R^J$, $b_i = (0, \ldots, 0, 100)^T$. On the other hand, the initial vector for the LANPRO problems was $x^{(0)} = (0.5, 0.5, \ldots, 0.5)^T$ and for the Laplace and biharmonic problems we used the null vector.

The parallel experiments were run on two different parallel computer systems. The first platform is an IBM RS/6000 SP with 8 nodes (“SP2”). These nodes are 120 MHz Power2 Super Chip and they are connected through a high performance switch with latency time of 40 microseconds and a bandwidth of 30 to 35 Mbytes per second. The second platform (“cluster”) is an Ethernet network of five 120 MHz Pentiums. The peak performance of this network is 100 Mbytes per second with a bandwidth around 6.5 Mbytes per second. In order to manage the parallel environment we have used the PVM library of parallel routines [19]. Also, we used the BLAS [3] routines for vector computations and the SPARSKIT [44] routines for handling sparse matrices. All times are reported in seconds. The conclusions are similar on both multiprocessors. However, obviously the computing platform has an influence in the performance of a parallel implementation.

In the first results presented in Table 5.1 we show the behaviour, for the LANPRO problems, of some block two-stage methods on both multiprocessors using as inner procedure

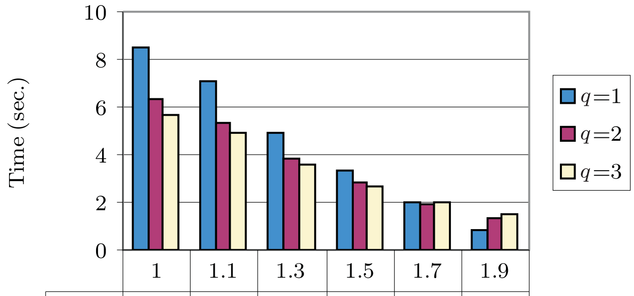
		TWO-STAGE				BLOCK-JACOBI		
# Proc.	Size of blocks	q	It.	Time Cluster	Time SP2	It.	Time Cluster	Time SP2
2		1	3318	11.19	1.40	909	10.22	1.07
	234	2	2151	8.42	1.10			
	234	3	1789	8.53	1.06			
		4	1615	8.26	1.10			
		5	1518	8.73	1.17			
		6	1458	9.44	1.25			
3		1	3784	13.44	1.65	1711	14.82	1.53
	156	2	2702	10.65	1.31			
	156	3	2382	10.09	1.29			
	156	4	2240	10.39	1.34			
		5	2165	10.86	1.42			
		6	2119	11.50	1.53			
2		1	65	0.24	0.04	15	1.03	0.53
	364	2	42	0.19	0.02			
	365	3	33	0.17	0.02			
		4	29	0.17	0.02			
		5	27	0.18	0.02			
		6	25	0.18	0.02			
3		1	66	0.28	0.02	17	0.68	0.19
	240	2	46	0.23	0.02			
	245	3	38	0.21	0.02			
	244	4	35	0.20	0.02			
		5	33	0.22	0.02			
		6	31	0.21	0.02			

TABLE 5.1
Block two-stage methods with Gauss-Seidel inner iterations for LANPRO problems.

the Gauss-Seidel method and performing at each block a fixed number of inner iterations ($q(l,j) = q, 1 \leq j \leq$ # Proc., $l = 0, 1, 2, \ldots$). We have considered block partitions of these matrices such that processor computations were balanced. We compare these methods with the well-known Block-Jacobi method. In this case, the subdomain problems are solved by using the Choleski complete factorization (see e.g., [38]). One can observe that the use of two-stage methods gives better results than the use of the Block-Jacobi method.

Table 5.2 shows the behaviour of these block two-stage methods for the Laplace's problem using, as above, as inner procedure the Gauss-Seidel method. In Table 5.3 we report the results for the Block-Jacobi method. The speed-up is calculated as $\frac{\text{CPU time of sequential algorithm}}{\text{REAL time of parallel algorithm}}$. One can observe that the best sequential two-stage algorithm does not obtain the best time when it solves the same problem in parallel. This is due to the fact that when the parameters $q(l,j)$ increase, the number of global iterations of the block two-stage method decreases. Therefore the communications among processors are reduced. Moreover, if the decrease in the number of global iterations balances the realization of more inner updates then, less execution time is observed. On the other hand, we observed that the efficiency ($\frac{\text{Speed-up}}{\text{processors's number}}$) increases with the number of inner iterations. However, the efficiency decreases notoriously when the number of processors increases. This fact is due to the inadequate use of the processors when the number of processors increases for a fixed

ETNA
Kent State University
etna@mcs.kent.edu



	1	1.1	1.3	1.5	1.7	1.9
q=1	8.5	7.1	4.9	3.3	2	0.8
q=2	6.3	5.3	3.8	2.8	1.9	1.3
q=3	5.7	4.9	3.6	2.7	2	1.5

Relaxation parameter

FIG. 5.1. *Block two-stage SOR for Laplace's problem. Size of matrix A:4096. IBM RS/6000 SP using two processors.*

matrix, because the cost of the operations performed in parallel can be smaller than the cost of communication. For example, in the last block partitioning of Table 5.2 using four processors for the cluster of Pentiums we obtained REAL times between 23.99 and 51.14 seconds. However, the CPU times were between 14.72 and 20.59 seconds. Here the network is very slow compared with the network in the other computing platform.

One of the most critical problems in a two-stage method is the choice of the number of inner iterations. Table 5.4 presents results obtained for the matrix of Table 5.2 partitioned into three blocks using as inner procedure the Gauss-Seidel method and varying the number of inner iterations at some outer step. In this table *Niter* indicates the outer iteration count and $[\cdot]$ denotes the integer part of a real number. If we compare this table with the numerical results of Table 5.2 we observe no significant differences. Our experience indicates that an optimal sequence of inner iterations is a little greater than one and constant, producing a priori a load balance based on the block size assigned to each processor. We also experimented using different stopping criteria for the number of inner iterations by specifying a certain tolerance for the inner residual and the conclusions were similar.

The use of these inner iterations can be also seen in Figures 5.1 and 5.2, where we illustrate the influence of the relaxation parameter when we use as inner procedures the SOR and SSOR methods, respectively. We consider different block two-stage methods depending on the number of inner iterations performed at each block, and for each method we recorded the time in seconds on the IBM RS/6000 SP in relation to different relaxation parameters. In these figures we have considered the Laplace matrix of size 4096 partitioned into two blocks of equal size. Note that the size of the two blocks are 2048. Note that the choice of one method or another depends on the chosen relaxation parameter. That is, for $\omega = 1$, it is more efficient to use the method with symmetric Gauss-Seidel inner iterations; on the other hand, as the relaxation parameter increases the conclusion inverts. For example, for the optimal ω the best result is obtained when the SOR method is used as inner procedure. This fact is illustrated in Figure 5.3 for the Laplace matrix of size 16384 using four processors and with a block partition of sizes 4096.

In Table 5.5 the behaviour of the block two-stage methods for the biharmonic problem is

ETNA
Kent State University
etna@mcs.kent.edu

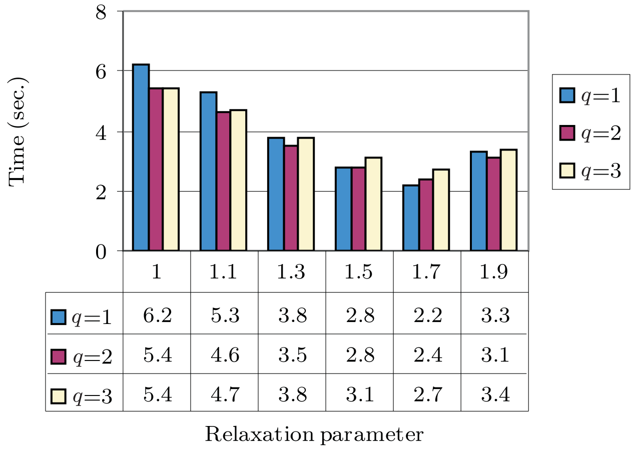
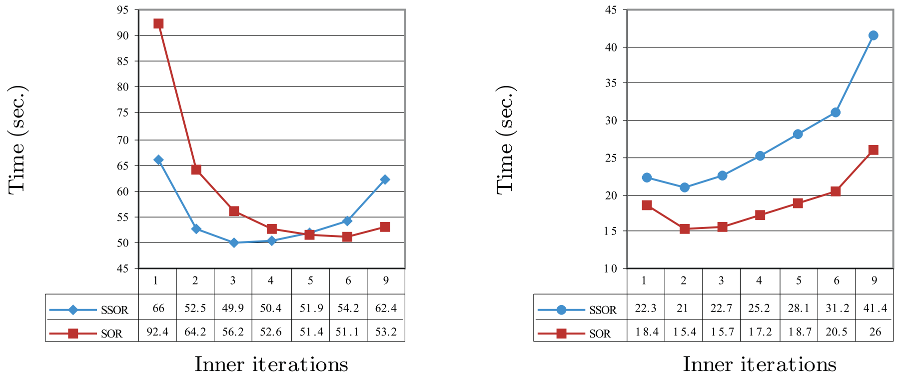


FIG. 5.2. *Block two-stage SSOR for Laplace's problem. Size of matrix A: 4096. IBM RS/6000 SP using two processors.*



(a) $\omega = 1$ (b) $\omega = 1,8$

FIG. 5.3. *Block two-stage SOR and SSOR for Laplace's problem. Size of matrix A:16384. IBM RS/6000 SP using four processors.*

illustrated. The results correspond to matrices of sizes 4096 and 10000 on the IBM RS/6000 SP. As it can be appreciated, for this problem the method needs too many iterations for convergence, and thus the execution times are very high. Later, we will present more results about this problem using the block two-stage method as preconditioners.

We want to point out that, in the above numerical experiments, we have chosen as outer splitting a Block-Jacobi type splitting satisfying (3.7). This choice has been made in this way in order to ensure the convergence of Algorithm 2 according to the theoretical results of §3. However, this choice does not include the classical Block-Jacobi splitting. Since the matrices arising from the discretization of Laplace's equation are not only positive definite but also M–matrices, following [8], we could have selected as outer splitting the splitting of the Block-Jacobi method. This is not the case for the LANPRO and biharmonic matrices. In selected cases, we ran experiments using this outer splitting for Laplace's problem. We have observed that comparing the times with those corresponding to the other outer splitting, the Block-Jacobi two-stage method is only about $5 - 10\%$ faster.

In order to improve these numerical results, it seems natural to construct a parallel preconditioned conjugate gradient (PCG) algorithm based on block two-stage methods. Note that the theoretical study of the convergence of the block two-stage methods made in this work allows us to use these methods as preconditioners; see also [10]. The idea of the PCG method consists of applying the conjugate gradient method (see [20]) to a better conditioned linear system $\hat{A}\hat{x} = \hat{b}$, where $\hat{A} = SAS^T$, $\hat{x} = S^{-T}x$, and $\hat{b} = Sb$. The matrix $\mathcal{M} = (S^TS)^{-1}$ is called the preconditioner or preconditioning matrix. The PCG method may be applied without computing $\hat{A}$, but solving the auxiliary system

$$\mathcal{M}s = \tau,$$

at each conjugate gradient iteration, where $\tau = b - Ax$ is the residual at the corresponding iteration; see e.g., [38].

We construct a preconditioned conjugate gradient method where the preconditioning matrix $\mathcal{M}$ is obtained using truncated series preconditioning as follows. To solve the auxiliary system $\mathcal{M}s = \tau$ of the PCG algorithm we use m steps of a Block two-stage method toward the solution of $As = \tau$, choosing $s^{(0)} = 0$. Particularly, in the numerical experiments we use as outer splitting the same as the above results and as inner procedure the SSOR method (say m-step Block two-stage SSOR PCG). The convergence test used was $\tau^T\tau < 10^{-7}$, and the initial vector was the null vector.

It is well-known that a bound on the convergence rate of PCG is given by the condition number of $\hat{A}$. In Table 5.6 we show the condition number of $\hat{A}$ as a function of the number of inner iterations and the number of steps of the preconditioning, for two and four processors, using symmetric Gauss-Seidel inner iterations. These condition numbers have been calculated for two matrices of size 1024 corresponding to the Laplace and biharmonic problems and using MATLAB. The condition number for the Laplace matrix is 440.68 and for the biharmonic matrix is 65549.09. It is observed that the condition number of $\hat{A}$ is always less than the condition number of A. Moreover, the condition number decreases when the number of inner iterations or the number of steps increases. Furthermore, one can observe, as is to be expected, that the larger the number of processor (i.e., the number of diagonal blocks in the outer block Jacobi type splitting), the larger the condition number of $\hat{A}$.

Table 5.7 shows the behaviour of this Block two-stage PCG method for the Laplace matrix of size 4096 using two processors and with a block partition of size 2048. If we compare the parallel times of this table with Figures 5.1 and 5.2, it can be appreciated that the best times are obtained with the m-step Block two-stage SSOR PCG methods. Similar conclusions can be observed for the biharmonic problem, as we show in Figure 5.4 for the matrix of size 10000, using two processors (see also Table 5.5). However, as it can be appreciated in Table 5.7, when the matrix is small, this parallel method is slower than the sequential m-step SSOR preconditioned conjugated gradient (SSOR PCG) method (see e.g., [38]). Different conclusions were obtained when the size of the matrix increases. In this way, Table 5.8 shows the results for the Block two-stage PCG methods with symmetric Gauss-Seidel (SGS) inner iterations and for the sequential m-step symmetric Gauss-Seidel PCG, for a Laplace matrix of size 40000. We use two processors and the matrix is partitioned into two blocks of sizes

ETNA
Kent State University
etna@mcs.kent.edu

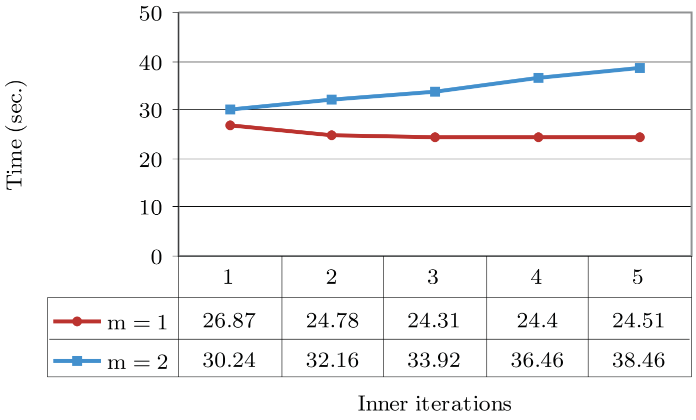


FIG. 5.4. *Block two-stage SGS PCG method for biharmonic problem. Size of matrix A: 10000. IBM RS/6000 SP using two processors.*

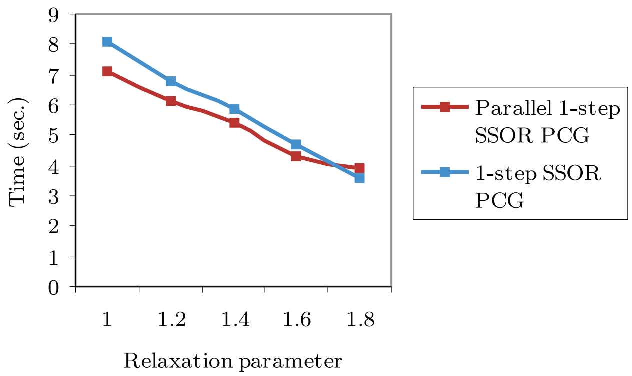


FIG. 5.5. *1-step Block two-stage SSOR PCG (q=2) versus 1-step SSOR PCG for Laplace's problem. Size of matrix A: 40000. IBM RS/6000 SP using two processors.*

20000. It can be observed in this table that Block two-stage SGS PCG methods accelerate the classical sequential SGS PCG method.

Figures 5.5 and 5.6 compare the behaviour of these methods for the Laplace problem using the matrix of size 40000 and a matrix of size 262144, respectively, in relation to the relaxation parameter used in the SSOR procedure. Here we also use two processors and the matrix of size 262144 is partitioned into two blocks of sizes 131072. Figure 5.7 shows the behaviour of the Block two-stage SSOR PCG methods for a biharmonic matrix of size 40000 and using two processors. We want to point out that, in general in our experiments, the optimal number of steps m was one or two, and the best times were obtained with ω around 1.8 in the Laplace problem and 1.9 in the biharmonic problem. On the other hand, it is observed that the parallel times and the sequential time are similar when the relaxation parameter is close to the optimal. However, this optimal parameter is not easy to obtain

ETNA
Kent State University
etna@mcs.kent.edu

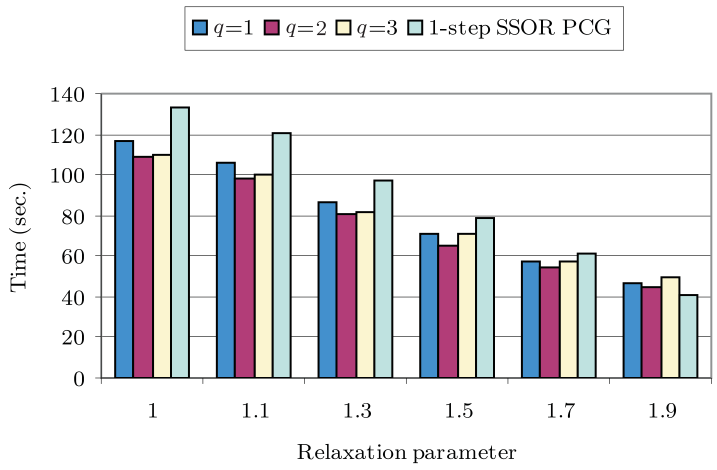
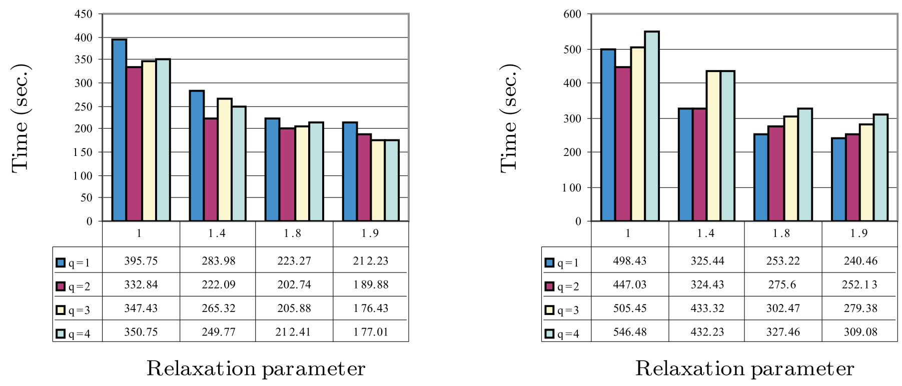


FIG. 5.6. *Block two-stage SSOR PCG (1-step) versus 1-step SSOR PCG for Laplace's problem. Size of matrix A: 262144. IBM RS/6000 SP using two processors.*



(a) Steps of the preconditioning: 1. (b) Steps of the preconditioning: 2.

FIG. 5.7. *Block two-stage SSOR PCG for biharmonic problem. Size of matrix A: 40000. IBM RS/6000 SP using two processors.*

a priori, and then it is customary to choose a relaxation parameter neither too big nor too small to ensure the convergence. In this way, Figure 5.8 compares the behaviour of some Block two-stage SSOR PCG methods for the Laplace matrix of size 262144 in relation to the number of SSOR inner iterations performed ($q = 1, 2, 3, 4$ and 5) for $\omega = 1.5$, and using three processors with block partitions of sizes 87040, 87040 and 88064. As it can be observed in this figure, the parallel algorithms always accelerate the sequential PCG algorithm.

Acknowledgments. The authors wish to thank the anonymous referees for their helpful comments. Their careful reading and insight are appreciated. The authors also thank

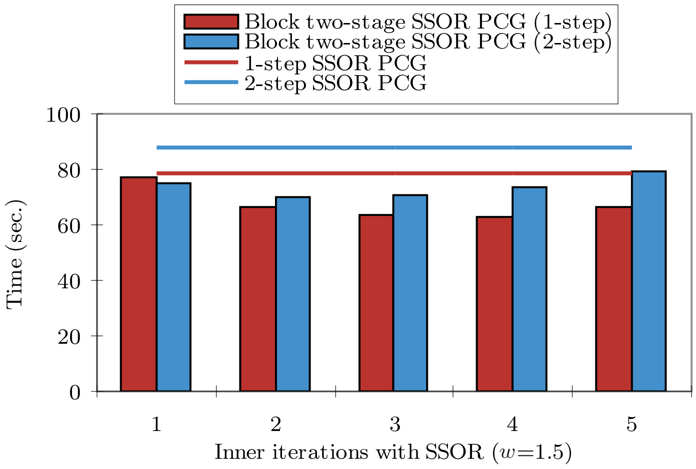


FIG. 5.8. *Block two-stage SSOR PCG for Laplace's problem with* $w = 1.5$. *Size of matrix A: 262144. IBM RS/6000 SP using three processors.*

Daniel B. Szyld for helpful comments on the manuscript. These comments helped improve our presentation.

REFERENCES

[1] M. BENZI AND D. B. SZYLD, *Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods*, Numer. Math., 76 (1997), pp. 309–322.
[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Third edition, Academic Press, New York, 1979. Reprinted by SIAM, Philadelphia, 1994.
[3] BLAS, *Basic Linear Algebra Subroutine*. Available online from http://www.netlib.org/index.html.
[4] R. BRU, L. ELSNER, AND M. NEUMANN, *Models of parallel chaotic iteration methods*, Linear Algebra Appl., 103 (1988), pp. 175–192.
[5] R. BRU AND R. FUSTER, *Parallel chaotic extrapolated Jacobi method*, Appl. Math. Lett., 3 (1990), pp. 65–69.
[6] R. BRU, V. MIGALLÓN, AND J. PENADÉS, *Chaotic inner-outer iterative schemes*, in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, J. Lewis, ed., SIAM Press, Philadelphia, 1994, pp. 434–438.
[7] ———, *Chaotic methods for the parallel solution of linear systems*, Computing Systems in Engineering, 6 (1995), pp. 385–390.
[8] R. BRU, V. MIGALLÓN, J. PENADÉS, AND D. B. SZYLD, *Parallel, synchronous and asynchronous two–stage multisplitting methods*, Electron. Trans. Numer. Anal., 3 (1995), pp. 24–38.
[9] M. J. CASTEL, V. MIGALLÓN, AND J. PENADÉS, *Convergence of non-stationary parallel multisplitting methods for Hermitian positive definite matrices*, Math. Comp., 67 (1998), pp. 209–220.
[10] ———, *Block two-stage preconditioners*, Appl. Math. Lett., to appear.
[11] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
[12] A. FROMMER AND G. MAYER, *Convergence of relaxed parallel multisplitting methods*, Linear Algebra Appl., 119 (1989), pp. 141–152.
[13] ———, *Parallel interval multisplittings*, Numer. Math., 56 (1989), pp. 255–267.
[14] A. FROMMER AND B. POHL, *Comparison results for splittings based on overlapping blocks*, in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, J. Lewis, ed., SIAM Press, Philadelphia, 1994, pp. 29–33.
[15] ———, *A comparison result for multisplittings and waveform relaxation methods*, Numer. Linear Algebra Appl., 2 (1995), pp. 335–346.
[16] A. FROMMER AND D. B. SZYLD, *H-splittings and two-stage iterative methods*, Numer. Math., 63 (1992),

pp. 345–356.

[17] A. FROMMER AND D. B. SZYLD, *Asynchronous two-stage iterative methods*, Numer. Math., 69 (1994), pp. 141–153.

[18] R. FUSTER, V. MIGALLÓN, AND J. PENADÉS, *Non-stationary parallel multisplitting AOR methods*, Electron. Trans. Numer. Anal., 4 (1996), pp. 1–13.

[19] A. GEIST, A. BEGUELIN, J. DONGARRA, W. JIANG, R. MANCHEK, AND V. SUNDERAM, PVM 3 User's Guide and Reference Manual. Technical Report ORNL/TM-12187, Oak Ridge National Laboratory, Tennessee, 1994.

[20] M. R. HESTENES AND S. R. STEIFEL, *Methods of conjugate gradient for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.

[21] R. A. HORN AND C. A. JOHNSON, *Matrix Analysis*, Cambridge University Press, 1985.

[22] A. S. HOUSEHOLDER, *The Theory of Matrices in Numerical Analysis*, Blaisdell, Waltham, Massachusetts, 1964. Reprinted by Dover, New York, 1975.

[23] C. R. JOHNSON AND R. BRU, *The spectral radius of a product of nonnegative matrices*, Linear Algebra Appl., 141 (1990), pp. 227–240.

[24] M. T. JONES AND D. B. SZYLD, *Two-stage multisplitting methods with overlapping blocks*, Numer. Linear Algebra Appl., 3 (1996), pp. 113–124.

[25] H. B. KELLER, *On the solution of singular and semidefinite linear systems by iteration*, SIAM J. Numer. Anal., 2 (1965), pp. 281–290.

[26] P. J. LANZKRON, D. J. ROSE, AND D. B. SZYLD, *Convergence of nested iterative methods for linear systems*, Numer. Math., 58 (1991), pp. 685–702.

[27] K. LÖWNER, *Über monotone matrixfunktionen*, Math. Z., 38 (1934), pp. 177–216.

[28] J. MAS, V. MIGALLÓN, J. PENADÉS, AND D. B. SZYLD, *Non-stationary parallel relaxed multisplitting methods*, Linear Algebra Appl., 241/243 (1996), pp. 733–748.

[29] V. MIGALLÓN AND J. PENADÉS, *Convergence of two-stage iterative methods for Hermitian positive definite matrices*, Appl. Math. Lett., 10 (1997), pp. 79–83.

[30] ———, *The monotonicity of two-stage iterative methods*, Appl. Math. Lett., 12 (1999), pp. 73–76.

[31] R. NABBEN, *A note on comparison theorems for splittings and multisplittings of Hermitian positive definite matrices*, Linear Algebra Appl., 233 (1995), pp. 67–80.

[32] M. NEUMANN AND R. J. PLEMMONS, *Convergence of parallel multisplitting iterative methods for M–matrices*, Linear Algebra Appl., 88–89 (1987), pp. 559–573.

[33] N. K. NICHOLS, *On the convergence of two-stage iterative processes for solving linear equations*, SIAM J. Numer. Anal., 10 (1973), pp. 460–469.

[34] D. P. O'LEARY AND R. E. WHITE, *Multi-splittings of matrices and parallel solution of linear systems*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 630–640.

[35] J. O'NEIL AND D. B. SZYLD, *A block ordering method for sparse matrices*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 811–823.

[36] J. M. ORTEGA, *Numerical Analysis: A Second Course*, Academic Press, New York, 1972. Reprinted by SIAM, Philadelphia, 1990.

[37] J. M. ORTEGA, *Matrix Theory*, Plenum Press, New York, 1987.

[38] J. M. ORTEGA, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, 1988.

[39] A. M. OSTROWSKI, *Über die determinanten mit überwiegender hauptdiagonale*, Comment. Math. Helv., 10 (1937), pp. 69–96.

[40] J. PENADÉS, *Métodos Iterativos Paralelos para la Resolución de Sistemas Lineales basados en Multiparticiones*, PhD thesis, Departamento de Tecnología Informática y Computación, Universidad de Alicante, December 1993. In Spanish.

[41] W. C. RHEINBOLDT AND J. S. VANDERGRAFT, *A simple approach to the Perron-Frobenius theory for positive operators on general partially-ordered finite-dimensional linear spaces*, Math. Comp., 27 (1973), pp. 139–145.

[42] F. ROBERT, M. CHARNAY, AND F. MUSY, *Itérations chaotiques série-parallèle pour des équations nonlinéaires de point fixe*, Apl. Mat., 20 (1975), pp. 1–38.

[43] H. RUTISHAUSER, *Lectures on Numerical Mathematics*, Birkhäuser Boston, Boston, Massachusetts, 1990.

[44] Y. SAAD, SPARSKIT: *A basic tool kit for sparse matrix computations*, Technical Report 90-20, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffet Field, CA, 1990. Second version of SPARSKIT available online from `http://www.cs.emn.edu/Research/arpa/SPARSKIT`.

[45] D. B. SZYLD, *Synchronous and asynchronous two-stage multisplitting methods*, in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, J. Lewis, ed., SIAM Press, Philadelphia, 1994, pp. 39–44.

[46] D. B. SZYLD AND M. T. JONES, *Two–stage and multisplitting methods for the parallel solution of linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 671–679.

[47] R. S. VARGA, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, New Jersey, 1962. Second Edition,

ETNA
Kent State University
etna@mcs.kent.edu

revised and expanded, Springer, Berlin, 2000.

[48] R. E. WHITE, *Multisplittings and parallel iterative methods*, Comput. Methods Appl. Mech. Engrg., 64 (1987), pp. 567–577.

[49] ———, *Multisplitting with different weighting schemes*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 481–493.

[50] ———, *Multisplitting of a symmetric positive definite matrix*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 69–82.

ETNA
Kent State University
etna@mcs.kent.edu

				IBM RS/6000 SP			CLUSTER		
# Proc.	Size of blocks	q	Iter.	Par. t.	Seq. t.	Speed-up	Par. t.	Seq. t.	Speed-up
2		1	4337	8.5	8.88	1.04	41.92	42.14	0.99
		2	2412	6.35	8.24	1.29	30.49	37.85	1.24
		3	1741	5.74	8.34	1.45	26.95	37.80	1.40
	2048	4	1400	5.60	8.66	1.54	25.96	38.80	1.49
	2048	5	1194	5.57	8.97	1.61	26.06	40.27	1.54
3		1	4428	9.19	9.18	0.99	44.19	43.74	0.98
		2	2506	6.30	8.57	1.36	29.52	39.25	1.32
	1344	3	1804	5.43	8.61	1.58	25.25	39.51	1.56
	1344	4	1503	5.09	9.04	1.77	23.87	40.96	1.71
	1408	5	1300	4.98	9.99	2.00	23.20	43.05	1.85
4		1	4513	10.76	9.47	0.88	51.14	44.38	0.86
	1024	2	2593	7.03	8.72	1.24	33.37	39.54	1.18
	1024	3	1932	5.86	8.97	1.53	27.57	39.96	1.44
	1024	4	1599	5.35	9.47	1.77	25.10	41.76	1.66
	1024	5	1399	5.15	10.09	1.95	23.99	44.18	1.84

TABLE 5.2
Block two-stage methods with Gauss-Seidel inner iterations for Laplace's problem. Size of matrix A: 4096.

		IBM RS/6000 SP			CLUSTER		
# Proc.	Iter.	Par. t.	Seq. t.	Speed-up	Par. t.	Seq. t.	Speed-up
2	235	13.55	26.25	1.93	90.77	177.82	1.95
3	296	7.46	19.14	2.56	55.50	157.77	2.84
4	362	5.24	16.06	3.06	41.30	149.49	3.61

TABLE 5.3
Block-Jacobi for Laplace's problem. Size of matrix A: 4096.

		# Inner Iter. $=$ $\max(1, q - [\frac{Niter}{k}])$		# Inner Iter. $=$ $q + [\frac{Niter}{k}]$	
q	k	Iter.	Par. t.	Iter.	Par. t.
1	250	4428	9.19	1681	5.60
5	250	2494	6.25	1115	5.12
6	250	1694	5.05	1043	5.20
7	250	1250	4.78	980	5.22
8	250	1122	4.88	932	5.35
1	500	4428	9.19	2162	6.08
5	500	1482	5.03	1210	5.04
6	500	1254	4.91	1112	5.07
7	500	1115	4.94	1036	5.13

TABLE 5.4
Block two-stage methods with Gauss-Seidel inner iterations for Laplace's problem. Size of matrix A: 4096. IBM RS/6000 SP using three processors.

		$\omega = 1.2$		$\omega = 1.4$		$\omega = 1.6$	
	q	Iter.	Time	Iter.	Time	Iter.	Time
4096	2	451488	2222.18	368562	1812.82	313507	1543.73
	3	373215	2497.43	318090	2125.96	279363	1869.03
	4	334473	2829.03	293481	2482.21	263936	2231.12
	6	296216	3556.91	269384	3234.95	249737	2997.69
10000	2	3916601	26942.73	1878772	21525.19	1499680	17189.03
	3	2417216	26792.23	1550807	24834.74	1288571	20630.13

TABLE 5.5

Block two-stage SSOR for the biharmonic problem. IBM RS/6000 SP using two processors.

		Laplace		Biharmonic	
		2 proc.	4 proc.	2 proc.	4 proc.
m	q	cond($\hat{A}$)	cond($\hat{A}$)	cond($\hat{A}$)	cond($\hat{A}$)
1	1	66.67	76.89	7227.35	8669.72
	2	39.84	50.03	4686.32	6116.39
	3	31.53	41.80	3904.07	5359.72
	4	27.66	37.96	3532.66	5008.51
	5	25.47	35.79	3317.80	4808.51
2	1	35.59	38.69	3613.92	4335.11
	2	20.17	25.26	2343.41	3058.44
	3	16.02	21.15	1952.28	2680.11
	4	14.08	19.23	1766.58	2504.50
	5	12.99	18.14	1659.15	2404.50
3	1	22.56	25.96	2409.45	2890.24
	2	13.62	17.01	1562.44	2039.13
	3	10.85	14.27	1301.69	1786.90
	4	9.56	12.99	1177.88	1669.83
	5	8.83	12.26	1106.26	1603.17
4	1	17.04	19.60	1807.21	2167.80
	2	10.34	12.88	1171.95	1592.47
	3	8.26	10.83	976.39	1340.30
	4	7.30	9.87	883.54	1252.50
	5	6.75	9.33	829.82	1202.50
5	1	13.74	15.78	1445.87	1734.34
	2	8.37	10.41	937.66	1223.67
	3	6.72	8.77	781.21	1072.34
	4	5.94	8	706.93	1002.10
	5	5.51	7.56	663.96	962.10
6	1	11.53	13.23	1204.97	1445.37
	2	7.07	8.76	781.47	1019.81
	3	5.68	7.39	651.09	893.70
	4	5.04	6.75	589.19	835.16
	5	4.68	6.39	553.38	801.83

TABLE 5.6

Condition number of matrix $\hat{A}$, using symmetric Gauss-Seidel inner iterations. Laplace matrix 1024, cond(A)=440.68. Biharmonic matrix 1024, cond(A)=65549.09.

ETNA
Kent State University
etna@mcs.kent.edu

		m-step Block two-stage SSOR PCG			Sequential m-step SSOR PCG	
m	ω	q	Iter.	Time	Iter.	Time
1	1	1	65	0.52	62	0.30
1	1.7	1	42	0.32	33	0.16
1	1.9	1	59	0.48	27	0.13
1	1	2	48	0.44		
1	1.7	2	34	0.32		
1	1.9	2	44	0.40		
1	1	3	39	0.40		
1	1.7	3	33	0.35		
1	1.9	3	40	0.42		
2	1	1	46	0.56	43	0.33
2	1.7	1	29	0.36	22	0.17
2	1.9	1	41	0.50	18	0.14
2	1	2	48	0.55		
2	1.7	2	34	0.39		
2	1.9	2	44	0.53		
2	1	3	39	0.56		
2	1.7	3	33	0.47		
2	1.9	3	40	0.56		

TABLE 5.7

m-step Block two-stage SSOR PCG and sequential m-step SSOR PCG methods for Laplace's problem. Size of matrix A: 4096. IBM RS/6000 SP using two processors.

	m-step Block two-stage SGS PCG					Sequential m-step SGS PCG	
m	q	Iter.	Par. t.	Seq. t.	Speed-up	Iter.	Time
1	1	171	7.79	8.22	1.05	167	8.12
1	2	122	7.17	9.05	1.26		
1	3	104	7.51	10.38	1.38		
2	1	120	7.99	9.28	1.66	117	9.17
2	2	86	8.06	11.11	1.37		
2	3	74	8.92	13.42	1.50		

TABLE 5.8

Block two-stage SGS PCG and sequential m-step SGS PCG methods for Laplace's problem. Size of matrix A: 40000. IBM RS/6000 SP using two processors.