Electronic Transactions on Numerical Analysis.
Volume 3, pp. 24-38, March 1995.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

PARALLEL, SYNCHRONOUS AND ASYNCHRONOUS TWO-STAGE MULTISPLITTING METHODS *

RAFAEL BRU†, VIOLETA MIGALLÓN‡, JOSÉ PENADÉS†, AND DANIEL B. SZYLD§

Abstract. Different types of synchronous and asynchronous two-stage multisplitting algorithms for the solution of linear systems are analyzed. The different algorithms which appeared in the literature are reviewed, and new ones are presented. Convergence properties of these algorithms are studied when the matrix in question is either monotone or an H-matrix. Relaxed versions of these algorithms are also studied. Computational experiments on a shared memory multiprocessor vector computer are presented.

Key words. asynchronous methods, two-stage iterative methods, linear systems, multisplittings, parallel algorithms.

AMS subject classifications. 65F10, 65F15.

1. Introduction.

In this paper we present various types of synchronous and asynchronous two-stage multisplitting algorithms for the solution of linear systems of the form

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

where A is an $n \times n$ nonsingular matrix. We first give a general convergence result, and then, for the most part, concentrate our study on two important cases: when A is monotone, i.e., when $A^{-1} \geq O$ [1], and when A is an H-matrix [30], [41]. The multisplitting algorithm was introduced by O'Leary and White [28] and was further studied by many authors; see, e.g., Frommer and Mayer [10], [11], Neumann and Plemmons [26], or White [42], [44].

One can think of the multisplitting algorithm as an extension and parallel generalization of the classical Block Jacobi algorithm, see, e.g., Varga [40], which we review in what follows. Let the matrix A is partitioned into $L \times L$ blocks

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1L} \\ A_{21} & A_{22} & \cdots & A_{2L} \\ \vdots & \vdots & & \vdots \\ A_{L1} & A_{L2} & \cdots & A_{LL} \end{bmatrix}, \tag{1.2}$$

with the diagonal blocks $A_{\ell\ell}$ being square nonsingular of order n_ℓ, $\ell = 1, \ldots, L$, $\sum_{\ell=1}^{L} n_\ell = n$, and the vectors x and b are partitioned conformally.

* Received October 3, 1994. Accepted February 27, 1995. Communicated by Richard Varga.

† Departament de Matemàtica Aplicada, Universitat Politècnica de València, E-46071 València, Spain (`rbru@mat.upv.es`). This research was partially supported by both Spanish CICYT grant number TIC91-1157-C03-01 and the ESPRIT III basic research programme of the EC under contract No. 9072 (project GEPPCOM).

‡ Departament de Tecnologia Informàtica i Computació, Universitat d'Alacant, E-03080 Alacant, Spain (`violeta@dtic.ua.es`). (`jpenades@dtic.ua.es`). This research was supported by Spanish CICYT grant number TIC91-1157-C03-01.

§ Department of Mathematics, Temple University, Philadelphia, Pennsylvania 19122-2585, USA (`szyld@euclid.math.temple.edu`). This research was supported by the National Science Foundation grant DMS-9201728.

ALGORITHM 1. (BLOCK JACOBI). Given the initial vector $x_0^T = [(x_0^{(1)})^T, \ldots, (x_0^{(L)})^T]$.

For $i = 1, 2, \ldots,$ until convergence.

For $\ell = 1$ to L

$$A_{\ell\ell} x_i^{(\ell)} = b^{(\ell)} - \sum_{k=1, k \neq \ell}^{L} A_{\ell k} x_{i-1}^{(k)} \ . \tag{1.3}$$

Each system (1.3) can be solved in parallel by a different processor of a parallel computer, and the vector iterate at each step is

$$x_i^T = [(x_i^{(1)})^T, (x_i^{(2)})^T, \ldots, (x_i^{(L)})^T].$$

The multisplitting algorithm [28] consists of having a collection of splittings

$$A = M_\ell - N_\ell, \quad \ell = 1, \ldots, L, \tag{1.4}$$

and diagonal nonnegative weighting matrices E_ℓ which add to the identity, and performing the following algorithm.

ALGORITHM 2. (MULTISPLITTING). Given the initial vector x_0.

For $i = 1, 2, \ldots,$ until convergence.

For $\ell = 1$ to L

$$M_\ell y_\ell = N_\ell x_{i-1} + b \tag{1.5}$$

$$x_i = \sum_{\ell=1}^{L} E_\ell y_\ell \ . \tag{1.6}$$

As it can be appreciated, Algorithm 1 can be seen as a special case of Algorithm 2 when all splittings are the same, namely, $M_\ell =$ Diag $(A_{11}, \ldots, A_{LL})$, the block-diagonal matrix consisting of the diagonal blocks of A, and the diagonal weighting matrices E_ℓ, have ones in the entries corresponding to the diagonal block $A_{\ell\ell}$ and zero otherwise. In this case we say that the matrices E_ℓ form a *partition of the identity*. A rendition of Algorithm 1 can also be obtained from Algorithm 2 by setting

$$M_\ell = \text{Diag } (I, \ldots, I, A_{\ell\ell}, I, \ldots, I), \quad \text{and} \tag{1.7}$$

$$E_\ell = \text{Diag } (O, \ldots, O, I, O, \ldots, O), \tag{1.8}$$

for $\ell = 1, \ldots, L$, i.e., the same partition of the identity just discussed.

Convergence of the multisplitting algorithm was first established for $A^{-1} \geq O$ by O'Leary and White [28] when the splittings (1.4) are weak regular, i.e., when $M_\ell^{-1} \geq O$ and $M_\ell^{-1} N_\ell \geq O$, where the inequalities are understood component-wise [1], [29], [40]. Comparison of convergence of different splittings (1.4) when they are M-splittings, i.e., when M_ℓ is an M-matrix (defined in the next section) and $N_\ell \geq O$ [23], [35], was studied by Neumann and Plemmons [26]. If the splittings (1.4) correspond to diagonal blocks of A of larger size than the corresponding number of ones in the partition of the identity, i.e, if $A_{\ell\ell}$ in (1.7) has order $\tilde{n}_\ell > n_\ell$, the order of I in (1.8), then the multisplitting Algorithm 2 corresponds to an overlapping algorithm, which was shown to have better asymptotic convergence rate than the Block Jacobi algorithm (without overlap); see Frommer and Pohl [12], [13], and also Jones and Szyld [21]. Weighting matrices which are not partitions of the identity, and therefore induce another type of overlap, have been also studied; see, e.g., O'Leary and White [28], White [43], or Frommer and Mayer [11].

As is the case for Block Jacobi, the systems (1.5) can be solved by different processors in parallel. Moreover, the components of y_ℓ corresponding to zeros in the diagonal of E_ℓ need not be computed; see (1.6).

When the linear systems (1.5), or (1.3), are not solved exactly, but rather their solutions approximated by iterative methods, we are in the presence of a two-stage method. Two-stage methods, sometimes called inner-outer iterations, were studied, e.g., by Nichols [27], Golub and Overton [17], [18], Lanzkron, Rose and Szyld [22], Frommer and Szyld [14], [15], Bru, Elsner and Neumann [3], and Bru, Migallón and Penadés [6]. When the systems (1.5) are solved iteratively in each processor, using the splittings

$$M_\ell = B_\ell - C_\ell, \quad \ell = 1, \ldots, L, \tag{1.9}$$

and performing a fixed number s of iterations, one obtains the following algorithm.

ALGORITHM 3. (TWO-STAGE MULTISPLITTING). Given the initial vector x_0, and the fixed number s of inner iterations.

For $i = 1, 2, \ldots,$ until convergence.
 For $\ell = 1$ to L
  $y_{\ell,0} = x_{i-1}$
  For $j = 1$ to s

$$B_\ell y_{\ell,j} = C_\ell y_{\ell,j-1} + N_\ell x_{i-1} + b \tag{1.10}$$

$$x_i = \sum_{\ell=1}^{L} E_\ell y_{\ell,s} \ . \tag{1.11}$$

Convergence of this algorithm for any number of inner iterations s was established for $A^{-1} \geq O$ by Szyld and Jones [38] when the outer splittings (1.4) are regular splittings, i.e., when $M_\ell^{-1} \geq O$ and $N_\ell \geq O$ [1], [40], and the inner splittings (1.9) are weak regular splittings.

Algorithm 3 reduces to Algorithm 2 when the inner splitting (1.9) is the trivial $M_\ell = M_\ell - O$, and $s = 1$. We also note that, under certain circumstances, if the number of inner iterations is large enough, i.e., as $s \to \infty$, the iterates produced by Algorithm 3 resemble those produced by Algorithm 2, i.e., $y_{\ell,s} \to y_\ell$, as $s \to \infty$, where y_ℓ is the solution of (1.5); cf. [14, section 2] and the proof of Theorem 3.1.

When the number of inner iterations varies for each splitting and for each outer iteration, i.e., when $s = s(\ell, i)$ in Algorithm 3, we say that we have a NON-STATIONARY TWO-STAGE MULTISPLITTING ALGORITHM (ALGORITHM 4). Model A in Bru, Elsner and Neumann [2] is a special case of this algorithm, when the outer splittings (1.4) are all $A = A - O$.

A relaxation parameter $\omega > 0$ can be introduced and replace the computation of $y_{\ell,j}$ in (1.10) with the equation

$$B_\ell y_{\ell,j} = \omega(C_\ell y_{\ell,j-1} + N_\ell x_{i-1} + b) + (1-\omega) B_\ell y_{\ell,j-1}. \tag{1.12}$$

This is equivalent to replacing the splitting (1.9) by $M_\ell = \frac{1}{\omega} B_\ell - \left(\frac{1-\omega}{\omega} B_\ell + C_\ell \right)$. Clearly, with $\omega = 1$, equation (1.10) is recovered. In the case of $\omega \neq 1$ we have a RELAXED NON-STATIONARY TWO-STAGE MULTISPLITTING ALGORITHM (ALGORITHM

5); see also Bru and Fuster [4], Bru, Migallón and Penadés [6], Deren [7], Frommer and Mayer [10], Fuster, Migallón and Penadés [16], and Sun [36] where relaxed (or extrapolated) algorithms are considered.

In this paper we study the convergence of the Non-stationary Two-stage Multisplitting Algorithm 4, together with its relaxed version, Algorithm 5, see section 3. We also study their extension to two different asynchronous algorithms, where the (approximate) solutions of the systems (1.5), by repeated solutions of (1.10) or (1.12), proceed in each processor without waiting for the completion of the computation of the iterates in the other processors; see section 4.

For each equation (1.12) one can work with a different relaxation parameter ω_ℓ, $\ell = 1, \ldots, L$, as is done, e.g., in the MSOR method; see, e.g., [19]. We point out that the convergence results in this paper can be readily extended to this multi-parameter case without any difficulty.

In the next section we present some notation, definitions and preliminary results which we refer to later, while in section 5 we present some numerical experiments, which illustrate the performance of the algorithms studied.

Some of the results in this paper were presented in preliminary form in [5] and [37].

2. Preliminaries. We say that a vector x is nonnegative (positive), denoted $x \geq 0$ ($x > 0$), if all its entries are nonnegative (positive). Similarly, a matrix B is said to be nonnegative, denoted $B \geq O$, if all its entries are nonnegative or, equivalently, if it leaves invariant the set of all nonnegative vectors. We compare two matrices $A \geq B$, when $A - B \geq O$, and two vectors $x \geq y$ ($x > y$) when $x - y \geq 0$ ($x - y > 0$). Given a matrix $A = (a_{ij})$, we define the matrix $|A| = (|a_{ij}|)$. It follows that $|A| \geq O$ and that $|AB| \leq |A|\ |B|$ for any two matrices A and B of compatible size.

Let $Z^{n\times n}$ denote the set of all real $n \times n$ matrices which have all non–positive off–diagonal entries. A nonsingular matrix $A \in Z^{n\times n}$ is called M-matrix if $A^{-1} \geq O$, i.e., if A is a monotone matrix; see, e.g., Berman and Plemmons [1] or Varga [40]. By $\rho(A)$ we denote the spectral radius of the square matrix A.

For any matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, we define its comparison matrix $\langle A\rangle = (\alpha_{ij})$ by $\alpha_{ii} = |a_{ii}|, \quad \alpha_{ij} = -|a_{ij}|,\ i \neq j$. Following Ostrowski [30], [31], a nonsingular matrix A is said to be an H-matrix if $\langle A\rangle$ is an M-matrix. Of course, M-matrices are special cases of H-matrices. H-matrices, which are not necessarily monotone, arise in many applications and were studied by a number of authors in connection with iterative solutions of linear systems; see, e.g., the classical paper by Varga [41], or Frommer and Szyld [14] for an extensive bibliography and some examples.

Lemma 2.1. *Let $A, B \in \mathbb{R}^{n\times n}$.*
(a) If A is an M-matrix, $B \in Z^{n\times n}$, and $A \leq B$, then B is an M-matrix.
(b) If A is an H-matrix, then $|A^{-1}| \leq \langle A\rangle^{-1}$.
(c) If $|A| \leq B$ then $\rho(A) \leq \rho(B)$.

Proof. (a) and (c) can be found, e.g., in [29], 2.4.10 and 2.4.9, respectively. Part (b) goes back to Ostrowski [30]; see also, e.g., Neumaier [25]. □

Definition 2.2. *Let $A \in \mathbb{R}^{n\times n}$. The representation $A = M - N$ is called a splitting if M is nonsingular. It is called a convergent splitting if $\rho(M^{-1}N) < 1$. A splitting $A = M - N$ is called*
(a) regular if $M^{-1} \geq O$ and $N \geq O$ [39], [40],
(b) weak regular if $M^{-1} \geq O$ and $M^{-1}N \geq O$ [1], [29],
(c) H-splitting if $\langle M\rangle - |N|$ is an M-matrix [14], *and*

(d) H-compatible splitting if $\langle A\rangle = \langle M\rangle - |N|$ [14].

LEMMA 2.3. *Let $A = M - N$ be a splitting.*
(a) If the splitting is weak regular, then $\rho(M^{-1}N) < 1$ if and only if $A^{-1} \geq O$.
(b) If the splitting is an H-splitting, then A and M are H-matrices and $\rho(M^{-1}N) \leq \rho(\langle M\rangle^{-1}|N|) < 1$.
(c) If the splitting is an H-compatible splitting and A is an H-matrix, then it is an H-splitting and thus convergent.

Proof. (a) can be found, e.g., in [1], [29], [40]. The first part of (b) was shown in [24], [25]. The second part as well as (c) is found in [14]. □

LEMMA 2.4. [34] *Let $H_1, H_2, \ldots, H_i, \ldots$ be a sequence of nonnegative matrices in $\mathbb{R}^{n\times n}$. If there exist a real number $0 \leq \theta < 1$, and a vector $v > 0$ in $\mathbb{R}^n$, such that*

$$H_j v \leq \theta v, \text{ for all } j = 1, 2, \ldots,$$

then $\rho(V_i) \leq \theta^i < 1$, where $V_i = H_i \cdots H_2 \cdot H_1$, and $\lim_{i\to\infty} V_i = O$.

3. Synchronous Iterations. In this section we give convergence results for the Non-stationary Two-stage Multisplitting Algorithm 4 and its relaxed variant Algorithm 5. We first show that if the outer and inner splittings (1.4) and (1.9) are convergent, and if enough inner iterations are performed, then, the algorithms are convergent. Later we show that if certain conditions are imposed on the splittings, the methods converge for *any* number of inner iterations.

Let $x_\star$ be the solution of (1.1) and let $e_i = x_\star - x_i$ be the error at the ith outer iteration of Algorithm 4. Let $R_\ell = B_\ell^{-1}C_\ell$. We rewrite (1.11) as

$$x_i = \sum_{\ell=1}^{L} E_\ell[R_\ell^{s(\ell,i)} x_{i-1} + \sum_{j=0}^{s(\ell,i)-1} R_\ell^j B_\ell^{-1}(N_\ell x_{i-1} + b)];$$

cf. [15], [22]. Thus,

$$e_i = H(i)e_{i-1} = H(i)H(i-1)\cdots H(1)e_0, \tag{3.1}$$

where

$$\begin{aligned} H(i) &= \sum_{\ell=1}^{L} E_\ell[R_\ell^{s(\ell,i)} + \sum_{j=0}^{s(\ell,i)-1} R_\ell^j B_\ell^{-1} N_\ell] \\ &= \sum_{\ell=1}^{L} E_\ell[R_\ell^{s(\ell,i)} + (I - R_\ell^{s(\ell,i)})M_\ell^{-1}N_\ell] = I - Q(i)A, \end{aligned} \tag{3.2}$$

with

$$Q(i) = \sum_{\ell=1}^{L} E_\ell(I - R_\ell^{s(\ell,i)})M_\ell^{-1} = \sum_{\ell=1}^{L} E_\ell \sum_{j=0}^{s(\ell,i)-1} R_\ell^j B_\ell^{-1} \geq O. \tag{3.3}$$

Similarly, if $e_i = x_\star - x_i$ is the error at the ith outer iteration of Algorithm 5 and

$$S_\ell = S_\ell(\omega) = (1-\omega)I + \omega R_\ell, \tag{3.4}$$

it follows from (1.12) and (1.11) that $e_i = T(i)e_{i-1} = T(i)T(i-1)\cdots T(1)e_0$, where

$$T(i) = \sum_{\ell=1}^{L} E_\ell[S_\ell^{s(\ell,i)} + \sum_{j=0}^{s(\ell,i)-1} S_\ell^j B_\ell^{-1} N_\ell], \tag{3.5}$$

i.e., the iteration matrix for Algorithm 5, is similar to (3.2) of Algorithm 4, where R_ℓ is replaced by S_ℓ.

THEOREM 3.1. *Let A be non-singular. Let the splittings (1.4) be such that*

$$\|M_\ell^{-1}N_\ell\|_\infty < 1, \qquad \ell = 1, \ldots, L, \tag{3.6}$$

and let the splittings (1.9) be convergent. Assume further that $\lim_{i\to\infty} s(\ell, i) = \infty$, $\ell = 1, \ldots, L$. Then, the Non-stationary Two-stage Multisplitting Algorithm 4 converges to $x_\star$ with $Ax_\star = b$. If in addition, we assume that $0 < \omega < \dfrac{2}{1+\rho}$ where $\rho = \max\{\rho(B_\ell^{-1}C_\ell),\ 1 \le \ell \le L\}$, then the theorem holds for the Relaxed Non-stationary Two-stage Multisplitting Algorithm 5.

Proof. Consider first $\omega = 1$, i.e., Algorithm 4. Let $R_\ell = B_\ell^{-1}C_\ell$. Since $\rho(R_\ell) < 1$, $\lim_{i\to\infty} R_\ell^i = O$, for $\ell = 1, \ldots, L$. Then, given an $\epsilon > 0$, there exists an integer s_0 such that $\|R_\ell^s\|_\infty \le \epsilon$, for all $s \ge s_0$, $\ell = 1, \ldots, L$. Since $\lim_{i\to\infty} s(\ell, i) = \infty$, there exists an i_0 such that $\|R_\ell^{s(\ell,i)}\|_\infty \le \epsilon$, for all $i \ge i_0$, $\ell = 1, \ldots, L$. Let β be a real constant such that $\|M_\ell^{-1}N_\ell\|_\infty \le \beta < 1$, for $\ell = 1, \ldots, L$. The existence of such β follows from (3.6). Then, for $i \ge i_0$

$$\begin{aligned}
\|H(i)\|_\infty &\le \max_{1\le\ell\le L}\left[\|R_\ell^{s(\ell,i)} + (I - R_\ell^{s(\ell,i)})M_\ell^{-1}N_\ell\|_\infty\right] \\
&\le \max_{1\le\ell\le L}\left[\|R_\ell^{s(\ell,i)}\|_\infty + \left(1 + \|R_\ell^{s(\ell,i)}\|_\infty\right)\|M_\ell^{-1}N_\ell\|_\infty\right] \\
&\le \max_{1\le\ell\le L}\left[\epsilon + (1+\epsilon)\|M_\ell^{-1}N_\ell\|_\infty\right] \\
&\le \epsilon + (1+\epsilon)\beta \equiv \alpha_\epsilon.
\end{aligned}$$

Setting $\epsilon < \dfrac{1-\beta}{1+\beta}$ we have $\alpha_\epsilon < 1$ and the errors (3.1) convergence to zero.
For the case of Algorithm 5, i.e., for $\omega \ne 1$, observe that it follows from (3.4) that

$$\rho(S_\ell(\omega)) < 1, \text{ for } \ell = 1, \ldots, L \tag{3.7}$$

and the proof follows in the same way. □

The proof of Theorem 3.1 resembles that of [14, Theorem 2.4]. Here we need the additional hypothesis (3.6) since we have L different splittings. We also note that Theorem 3.1 can be proved in the same way if the norm in assumption (3.6) is replaced by any norm such that if for arbitrary matrices U, U_ℓ, and weighting matrices E_ℓ, $\ell = 1, \ldots, L$, such that $U = \sum_{\ell=1}^{L} E_\ell U_\ell$, then $\|U\| \le \max_{1\le\ell\le L}\|U_\ell\|$; see Bru and Fuster [4]. In particular, one can use any weighted max–norm associated with a positive vector; see, e.g., Householder [20], Rheinboldt and Vandergraft [33], or Frommer and Szyld [15], for descriptions and applications of these norms.

THEOREM 3.2. *Let $A^{-1} \ge O$. Let the splittings (1.4) be regular and the splittings (1.9) be weak regular. Then, the Non-stationary Two-stage Multisplitting Algorithm 4 converges to $x_\star$ with $Ax_\star = b$ for any initial vector x_0 and any sequence of numbers of inner iterations $s(\ell, i) \ge 1$, $\ell = 1, \ldots, L$, $i = 1, \ldots$. If in addition, we assume $0 < \omega < 1$, the theorem holds for the Relaxed Non-stationary Two-stage Multisplitting Algorithm 5.*

Proof. Let $0 < \omega \leq 1$. Algorithm 4 corresponds to $\omega = 1$, while Algorithm 5 corresponds to $\omega < 1$. Let $e_i = x_\star - x_i$ be the error at the ith outer iteration of either Algorithm 4 or Algorithm 5. We rewrite (3.5) as

$$T(i) = \sum_{\ell=1}^{L} E_\ell T_\ell(i), \tag{3.8}$$

where

$$T_\ell(i) = S_\ell^{s(\ell,i)} + \sum_{j=0}^{s(\ell,i)-1} S_\ell^j B_\ell^{-1} N_\ell \geq O, \tag{3.9}$$

where the inequality follows from the assumptions on the outer and inner splittings. Similarly to (3.2) and (3.3), it follows from (3.9) that

$$T_\ell(i) = I - P_\ell(i)A, \tag{3.10}$$

where $P_\ell(i) = (I - S_\ell^{s(\ell,i)})M_\ell^{-1} = \displaystyle\sum_{j=0}^{s(\ell,i)-1} S_\ell^j (I - S_\ell) M_\ell^{-1}$. From (3.4) it follows that $I - S_\ell = \omega B_\ell^{-1} M_\ell$, and thus,

$$P_\ell(i) = \omega \sum_{j=0}^{s(\ell,i)-1} S_\ell^j B_\ell^{-1}. \tag{3.11}$$

Consider any fixed vector $e > 0$ (e.g., with all components equal to 1), and $v = A^{-1}e$. Since $A^{-1} \geq O$ and no row of A^{-1} can have all null entries, we get $v > 0$. By the same arguments $B_\ell^{-1} e > 0$, $\ell = 1, \ldots, L$. We have from (3.10) and (3.11) that

$$T_\ell(i)v = (I - P_\ell(i)A)v = v - P_\ell(i)e = v - \omega B_\ell^{-1} e - \omega \sum_{j=1}^{s(\ell,i)-1} S_\ell^j B_\ell^{-1} e.$$

We have that $\displaystyle\sum_{j=1}^{s(\ell,i)-1} S_\ell^j B_\ell^{-1} e \geq 0$. Moreover, since $T_\ell(i)v \geq 0$ and $v - \omega B_\ell^{-1} e < v$, there exist constants $0 \leq \theta_\ell < 1$ such that $v - \omega B_\ell^{-1} e \leq \theta_\ell v$. Thus, $T_\ell(i)v \leq \theta_\ell v$ for all $\ell = 1, \ldots, L$, $i = 1, 2, \ldots$. Let $\theta = \max\{\theta_\ell,\ 1 \leq \ell \leq L\}$. From (3.8) we have then that

$$T(i)v \leq \theta v, \text{ for all } i = 1, 2, \ldots. \tag{3.12}$$

By Lemma 2.4 this implies that the product $V(i) = T(i) \cdot T(i-1) \cdots T(1)$ tends to zero as $i \to \infty$, and thus $\lim_{i\to\infty} e_i = 0$. □

The proof of Theorem 3.2 uses techniques similar to ones used in Theorem 2.1 in [2], in theorems 4.3 and 4.4 in [14] and in Theorem 2.2 in [15]. We point out that the bounds (3.12) are independent of the sequence $s(\ell, i) \geq 1$, $\ell = 1, \ldots, L$, $i = 1, 2, \ldots$.

THEOREM 3.3. *Let A be an H-matrix. Let the splittings (1.4) and (1.9) be H-compatible splittings. Then, the Non-stationary Two-stage Multisplitting Algorithm 4 converges to $x_\star$ with $Ax_\star = b$ for any initial vector x_0 and any sequence of numbers* *of inner iterations $s(\ell, i) \geq 1$, $\ell = 1, \ldots, L$, $i = 1, \ldots$. If in addition, we assume $0 < \omega < 1$, the theorem holds for the Relaxed Non-stationary Two-stage Multisplitting Algorithm 5.*

Proof. Let $0 < \omega \leq 1$. Algorithm 4 corresponds to $\omega = 1$, while Algorithm 5 corresponds to $\omega < 1$. Let $e_i = x_\star - x_i$ be the error at the ith outer iteration of either Algorithm 4 or Algorithm 5. From (3.4), using Lemma 2.1 (b) we obtain the following bound

$$|S_\ell| \leq (1-\omega)I + \omega|B_\ell^{-1}||C_\ell| \leq (1-\omega)I + \omega\langle B_\ell\rangle^{-1}|C_\ell| \equiv \hat{S}_\ell.$$

Thus, from (3.5), and again using Lemma 2.1 (b), we obtain

$$|T(i)| \leq \sum_{\ell=1}^{L} E_\ell[\hat{S}_\ell^{s(\ell,i)} + \sum_{j=0}^{s(\ell,i)-1} \hat{S}_\ell^j \langle B_\ell\rangle^{-1}|N_\ell|] \equiv \hat{T}(i). \tag{3.13}$$

The matrix $\hat{T}(i)$ is the iteration matrix corresponding to the ith iteration of a Relaxed Non-stationary Two-stage Multisplitting Algorithm for the monotone matrix $\langle A\rangle = \langle M_\ell\rangle - |N_\ell|$ with the regular splittings $\langle M_\ell\rangle - |N_\ell|$ and $\langle M_\ell\rangle = \langle B_\ell\rangle - |C_\ell|$. These matrices and splittings satisfy the hypothesis of Theorem 3.2 and we have, as in (3.12),

$$|T(i)|v \leq \hat{T}(i)v \leq \theta v, \text{ for all } i = 1, \ldots \tag{3.14}$$

for some $v \in \mathbb{R}^n$, $v > 0$ and $\theta \in [0, 1)$. Let $V(i) = T(i) \cdot T(i-1) \cdots T(1)$. We can bound $|V(i)| \leq |T(i)| \cdot |T(i-1)| \cdots |T(1)|$. Therefore by (3.14) and Lemma 2.4, $V(i)$ tends to zero as $i \to \infty$, implying $\lim_{i\to\infty} e_i = 0$. □

The proof of Theorem 3.3 follows from that of Theorem 3.2 in a way similar to the way that [15, Theorem 2.3] follows from [15, Theorem 2.2]. Here we need the additional hypothesis that the outer splittings be H-compatible, so that $\langle A\rangle = \langle M_\ell\rangle - |N_\ell|$ for all $\ell = 1, \ldots, L$.

4. Asynchronous Iterations. All methods described in section 1 and studied in section 3 are synchronous in the sense that step (1.11) is performed only after all approximations to the solutions of (1.5) are completed ($\ell = 1, \ldots, L$). Alternatively, if the weighting matrices form a partition of the identity, each part of x_i, say $x_i^{(\ell)} = E_\ell x_i$, can be updated as soon as the approximation to the solution of the corresponding system (1.5) is completed, without waiting for the other parts of x_i to be updated. Thus, the previous iterate x_{i-1} is no longer available for the computation of (1.5) or (1.10). Instead, parts of the current iterate are updated using a vector composed of parts of different previous, not necessarily the latest, iterates; cf. [2, Model B], [15, section 3], [16, section 4].

As is customary in the description and analysis of asynchronous algorithms, the iteration subscript is increased every time any part of the iteration vector is computed; see, e.g., the references in [2], [6], [8], [9], [15], [16]. In a formal way, the sets $J_i \subseteq \{1, 2, \ldots, L\}$, $i = 1, 2, \ldots$, are defined by $\ell \in J_i$ if the ℓth part of the iteration vector is computed at the ith step. The subscripts $r(k, i)$ are used to denote the iteration number of the kth part being used in the computation of any part in the ith iteration, i.e., the iteration number of the kth part available at the beginning of the computation of $x_i^{(\ell)}$, if $\ell \in J_i$.

Any arbitrary $n \times n$ matrix U can be decomposed into L $n \times n$ matrices $U^{(\ell)}$, $\ell = 1, \ldots, L$, so that for any vector u, $Uu = \sum_{\ell=1}^{L} U^{(\ell)}u$, where the nonzeros in $U^{(\ell)}u$

ETNA
Kent State University
etna@mcs.kent.edu

correspond to the nonzero elements in E_ℓ which form a partition of the identity, i.e., $U^{(\ell)} = E_\ell U$. With this notation, we write the following algorithm.

ALGORITHM 6. (OUTER ASYNCHRONOUS TWO-STAGE MULTISPLITTING).
Given the initial vector $x_0 = x_0^{(1)} + \cdots + x_0^{(L)}$
For $i = 1, 2, \ldots$

$$x_i^{(\ell)} = \begin{cases} x_{i-1}^{(\ell)} & \text{if } \ell \notin J_i \\ H^{(\ell)}(i)\left(x_{r(1,i)}^{(1)} + \cdots + x_{r(L,i)}^{(L)}\right) + Q^{(\ell)}(i)b & \text{if } \ell \in J_i. \end{cases} \tag{4.1}$$

with $H(i)$ and $Q(i)$ as defined in (3.2) and (3.3), respectively.

For easy comparison with Algorithm 8, we rewrite (4.1) explicitly as

$$x_i^{(\ell)} = \begin{cases} x_{i-1}^{(\ell)} & \text{if } \ell \notin J_i \\ E_\ell[R_\ell^{s(\ell,i)} x_{r(\ell,i)}^{(\ell)} + \displaystyle\sum_{j=0}^{s(\ell,i)-1} R_\ell^j B_\ell^{-1}(N_\ell \sum_{k=1}^{L} x_{r(k,i)}^{(k)} + b)] & \text{if } \ell \in J_i. \end{cases} \tag{4.2}$$

We always assume that the asynchronous iterations satisfy the following conditions. They are very natural in asynchronous computations; see, e.g., [9], [15].

$$r(\ell, i) < i \ \text{ for all } \ell = 1, \ldots, L, \ \ i = 1, 2, \ldots. \tag{4.3}$$

$$\lim_{i\to\infty} r(\ell, i) = \infty \ \text{ for all } \ell = 1, \ldots, L. \tag{4.4}$$

$$\text{The set } \{i \mid \ell \in J_i\} \ \text{ is unbounded for all } \ell = 1, \ldots, L. \tag{4.5}$$

Condition (4.3) simply states that only components previously computed are used, and not future ones. Condition (4.5) is equivalent to what Bru, Elsner and Neumann [2] and others, e.g., Bru, Migallón and Penadés [6], call an admissible sequence, i.e., implying that every component is updated infinitely often. These authors also use the concept of a regulated sequence, i.e., implying that the number of iterations between two updates of the same component is uniformly bounded. Our condition (4.4) is slightly more general since no such bound is assumed.

If in (4.1), i.e., in Algorithm 6, we replace $H(i)$ by $T(i)$ as defined in (3.5), using $S_\ell(\omega)$ as defined in (3.4), $\omega > 0$, and if we replace $Q(i)$ by $P(i) = \sum_{\ell=1}^{L} E_\ell P_\ell(i)$, where $P_\ell(i)$ are as in (3.11), we obtain a RELAXED OUTER ASYNCHRONOUS TWO-STAGE MULTISPLITTING ALGORITHM (ALGORITHM 7). This algorithm, which is the asynchronous version of Algorithm 5, can also be obtained by replacing R_ℓ by $S_\ell(\omega)$ in (4.2).

THEOREM 4.1. *Let $A^{-1} \geq O$. Let the splittings (1.4) be regular and the splittings (1.9) be weak regular. Assume that the sequence $r(\ell, i)$ and the sets J_i, $\ell = 1, \ldots, L$, $i = 1, 2, \ldots$, satisfy conditions (4.3)–(4.5). Then, the Outer Asynchronous Two-stage Multisplitting Algorithm 6 converges to $x_\star$ with $Ax_\star = b$ for any initial vector x_0 and for any sequence of numbers of inner iterations $s(\ell, i) \geq 1$, $\ell = 1, \ldots, L$, $i = 1, 2, \ldots$. If in addition, we assume $0 < \omega < 1$, the theorem holds for the Relaxed Outer Asynchronous Two-stage Multisplitting Algorithm 7.*

Proof. The proof follows in the same way as the proof of [15, Theorem 3.3] which in turn is based on [8, Theorem 3.4]. □

THEOREM 4.2. *Let A be an H-matrix. Let the splittings (1.4) and (1.9) be H-compatible splittings. Assume that the sequence $r(\ell, i)$ and the sets J_i, $\ell = 1, \ldots, L$,*

$i = 1, 2, \ldots,$ satisfy conditions (4.3)–(4.5). Then, the Outer Asynchronous Two-stage Multisplitting Algorithm 6 converges to $x_\star$ with $Ax_\star = b$ for any initial vector x_0 and for any sequence of numbers of inner iterations $s(\ell, i) \geq 1$, $\ell = 1, \ldots, L$, $i = 1, 2, \ldots$. If in addition, we assume $0 < \omega < 1$, the theorem holds for the Relaxed Outer Asynchronous Two-stage Multisplitting Algorithm 7.

Proof. The proof follows in the same way as the proof of [15, Theorem 3.4]. □

We consider now asynchronous two-stage multisplittings algorithm where, at each inner iteration, the most recent information from the other parts of the iterate is used. In other words, the parts $x^{(k)}_{r(k,i)}$ in the sum in (4.2) may differ for different values of j, $j = 0, \ldots, s(\ell, i) - 1$ ($\ell \in J_i$). To reflect this, we therefore use indices of the form $r(k, j, i)$. Thus, for example, if the subvector $x^{(1)}_i = E_1 x_i$ is being computed by one processor, one component at a time, the computed components can be read by the other processors and used in the computation of the other subvectors $x^{(\ell)}$, before all components of $x^{(1)}_i$ are computed, and $x^{(1)}_{r(1,j,i)}$ may change for different j, $j = 0, \ldots, s(\ell, i)-1$ ($\ell \in J_i$) while $x^{(1)}_i$ is being computed. These algorithms are called totally asynchronous to distinguish them from the outer asynchronous ones; see [15, section 4] for a discussion of possible advantages of these methods.

Algorithm 8. (Totally Asynchronous Two-stage Multisplitting).
Given the initial vector $x_0 = x^{(1)}_0 + \cdots + x^{(L)}_0$
For $i = 1, 2, \ldots$

$$x^{(\ell)}_i = \begin{cases} x^{(\ell)}_{i-1} & \text{if } \ell \notin J_i \\ E_\ell[R^{s(\ell,i)}_\ell x^{(\ell)}_{r(\ell,0,i)} + \displaystyle\sum_{j=0}^{s(\ell,i)-1} R^j_\ell B^{-1}_\ell (N_\ell \sum_{k=1}^{L} x^{(k)}_{r(k,j,i)} + b)] & \text{if } \ell \in J_i. \end{cases} \tag{4.6}$$

Analogous to (4.3)–(4.4) we now assume

$$\begin{cases} r(k, j, i) < i, \text{ for all } k = 1, \ldots, L, \; j = 0, \ldots, s(k, i) - 1, \; i = 1, 2, \ldots, \\ \lim\limits_{i\to\infty} \min\limits_{j=0,\ldots,s(k,i)-1} r(k, j, i) = \infty, \text{ for all } k = 1, \ldots, L. \end{cases} \tag{4.7}$$

Again, if R_ℓ is replaced by $S_\ell(\omega)$ in (4.6) we obtain a Relaxed Totally Asynchronous Two-stage Multisplitting Algorithm (Algorithm 9).

Theorem 4.3. *Let $A^{-1} \geq O$. Let the splittings (1.4) be regular and the splittings (1.9) be weak regular. Assume that the numbers $r(k, j, i)$ and the sets J_i, $k = 1, \ldots, L$, $i = 1, 2, \ldots,$ satisfy conditions (4.5) and (4.7). Then, the Totally Asynchronous Two-stage Multisplitting Algorithm 8 converges to $x_\star$ with $Ax_\star = b$ for any initial vector x_0 and for any sequence of numbers of inner iterations $s(\ell, i) \geq 1$, $\ell = 1, \ldots, L$, $i = 1, 2, \ldots$. If in addition, we assume $0 < \omega < 1$, the theorem holds for the Relaxed Totally Asynchronous Two-stage Multisplitting Algorithm 9.*

Proof. The proof follows in the same way as the proof of [15, Theorem 4.3] which in turn is based on [9, Theorem 2.1]. □

Theorem 4.4. *Let A be an H-matrix. Let the splittings (1.4) and (1.9) be H-compatible splittings. Assume that the numbers $r(k, j, i)$ and the sets J_i, $k = 1, \ldots, L$, $i = 1, 2, \ldots,$ satisfy conditions (4.5) and (4.7). Then, the Totally Asynchronous Two-stage Multisplitting Algorithm 8 converges to $x_\star$ with $Ax_\star = b$ for any initial vector x_0 and for any sequence of numbers of inner iterations $s(\ell, i) \geq 1$, $\ell = 1, \ldots, L$, $i = 1, 2, \ldots$. If in addition, we assume $0 < \omega < 1$, the theorem holds for the Relaxed Totally Asynchronous Two-stage Multisplitting Algorithm 9.*

Proof. The proof follows in the same way as the proof of [15, Theorem 4.4]. □

ETNA
Kent State University
etna@mcs.kent.edu

5. Numerical Experiments. We implemented the synchronous and outer asynchronous algorithms described in this paper on a shared memory computer Alliant FX/80 with 8 vector processors. We considered the problem of Laplace's equation satisfying Dirichlet boundary conditions in the rectangle $\Omega = [0, a] \times [0, b]$. We discretized, using five point finite differences, the domain Ω with $J \times K$ points equally spaced by h. This discretization yields the linear system (1.1), where A is block tridiagonal, $A = \text{tridiag } [-I, B, -I]$, where I and B are $K \times K$ matrices, I is the identity, and the matrix $B = \text{tridiag } [-1, 4, -1]$. Thus, A has $J \times J$ blocks of size $K \times K$. It follows directly that A is an M-matrix, and thus also an H-matrix.

In order to fully utilize the 8 processors, we let $L = 8$. The matrix A can be written as

$$A = \begin{bmatrix} D_1 & -I & & & \\ -I & D_2 & -I & & \\ & -I & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & D_8 \end{bmatrix},$$

where $D_\ell = \text{tridiag } [-I, B, -I]$ is of order n_ℓ, $\ell = 1, \ldots, 8$; cf. (1.2). Consider an (outer) Block Jacobi Algorithm, and thus we have the matrices $M_\ell = \text{ Diag } [D_1, D_2, \ldots, D_8]$, $\ell = 1, \ldots, 8$, and N_ℓ such that $A = M_\ell - N_\ell$. The matrices E_ℓ, $\ell = 1, \ldots, 8$, are partitioned according to the structure of the matrices M_ℓ, with the ℓth block equal to the identity block and 0's elsewhere, i.e., they form a partition of the identity; see the discussion after Algorithm 2. Thus, when we perform $s(\ell, i)$ steps of an iterative method to solve the linear system $M_\ell y = N_\ell x_{i-1} + b$, we only need to update the ℓth block of y; see (1.5) and (1.6). In the experiments reported here, $s(\ell, i)$ steps of the *SOR* method are performed to approximate the linear systems (1.5) .

Experiments were performed with matrices of different orders. The results are similar for all matrices; see [32]; and therefore we only present here results for $n = 5632$. The matrix A has $J = 11$ diagonal blocks B of size $K = 512$, and we chose $n_\ell = 1024$, for $\ell = 1, 2, 3$, and $n_\ell = 512$, for $\ell = 4, \ldots, 8$, to study the effect of different number of nonzeros in E_ℓ. With this choice, we can obtain a good load balance by having processors 4–8 perform twice as many (inner) iterations as processors 1–3. Recall that in this example the number of nonzeros is proportional to the order of the blocks. This example is similar to the situation in which the diagonal blocks correspond to natural physical structures of different size, or to a discretization of varying resolution.

We report the *CPU* time in seconds (for the concurrent compilation) as function of different relaxation parameters $\omega > 0$ for selected values of the non-stationary parameters $s(\ell, i)$. In Figure 5.1 the results for the synchronous algorithms are presented. We use the notation $2^3 4^5$ to represent the non-stationary parameters $s(\ell, i) = 2$ for $\ell = 1, 2, 3$ and $s(\ell, i) = 4$ for $\ell = 4, \ldots, 8$, for all $i = 1, \ldots,$ i.e., to represent that each of the first three processors updates its vector twice and each of the last five processors updates its vector four times, in any outer iteration. Similar notation is used for other values of the non-stationary parameters. In Figure 5.1 it can be observed that the *CPU* time for the algorithm with one inner iteration ($s(\ell, i) = 1$, or 1^8) is larger than that for the other cases, both in the stationary case with more inner iterations

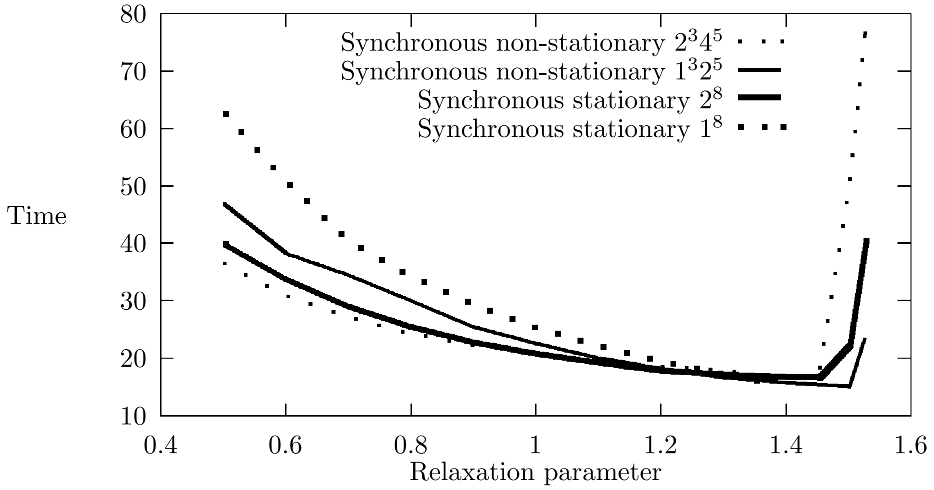

FIG. 5.1. *Synchronous algorithms,* $n = 5632$.

or in the non-stationary cases.

We point out that in the proofs of theorems 3.2–4.4, the restriction $\omega \leq 1$ is needed so that (3.7) holds. Nevertheless, as can be seen in the experiments reported here, the conclusions of the theorems hold in these examples for certain $\omega > 1$.

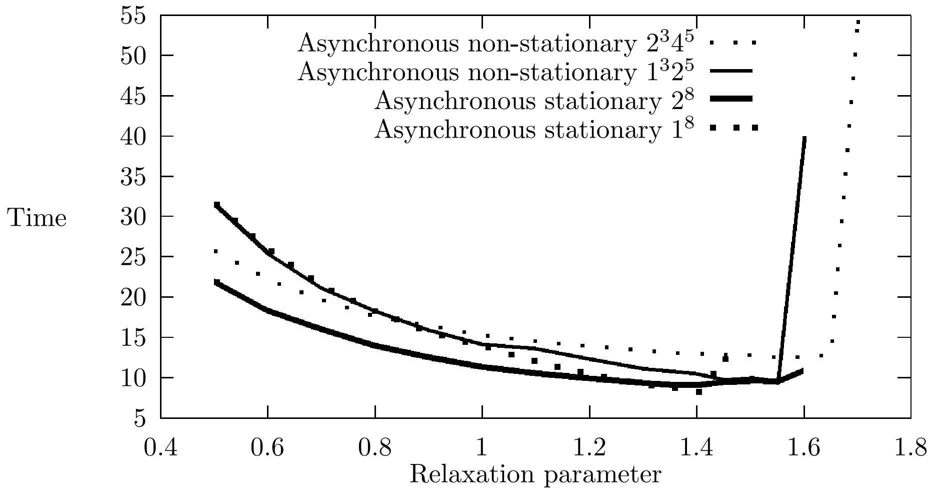

FIG. 5.2. *Asynchronous algorithms,* $n = 5632$.

In Figure 5.2 the CPU time of the corresponding outer asynchronous algorithms are presented. In this case, non-stationary algorithms behave better than the stationary ones. Observe that when ω is optimal, all versions give similar times, but when ω is less than the optimal value, in particular when $\omega = 1$, the non-stationary schemes produce faster convergence. Finally, we note that, with the choice of the orders n_ℓ used to achieve good load balancing, the asynchronous algorithms are better than the synchronous algorithms. Figure 5.3 illustrates this consideration for the non-stationary parameters $2^3 4^5$.

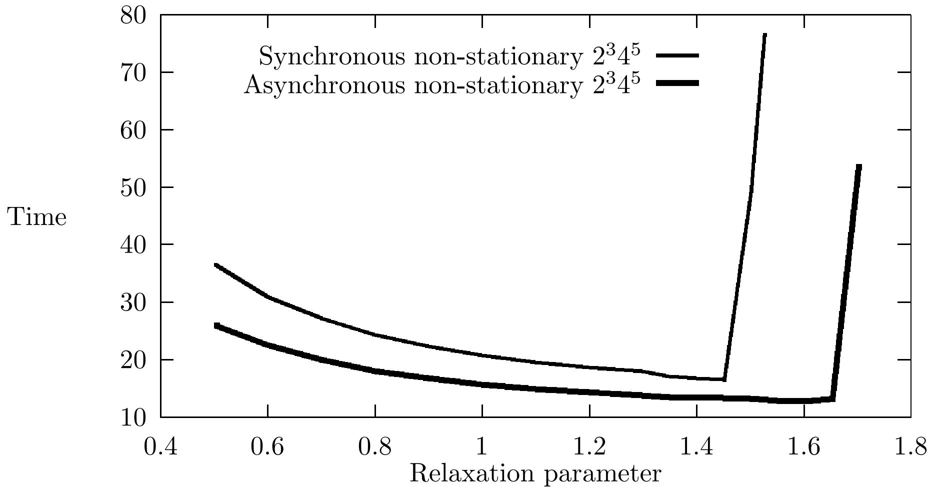

FIG. 5.3. *Synchronous and asynchronous algorithms,* $n = 5632$.

6. Conclusion. We have shown that, under certain conditions, the synchronous and asynchronous two-stage multisplitting algorithms presented in this paper converge for any number of inner iterations $s(\ell, i)$. This theory permits the use of different stopping criteria for the inner iterations, e.g., by specifying a certain tolerance for the inner residual; cf. Golub and Overton [17], [18].

The choice of optimal sequences $s(\ell, i)$ – or simply good ones – is problem dependent, and not very well understood. In our experience, often few inner iterations produce good overall convergence results. Moreover, in multiprocessors, a good choice for this sequence is one which counterweights the work in each processor, producing a good overall load balance. The experiments in section 5 illustrate such choices.

Acknowledgments. We acknowledge and appreciate the referees' questions and recommendations which led to improvements in the paper.

REFERENCES

[1] ABRAHAM BERMAN AND ROBERT J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York, third edition, 1979. Reprinted by SIAM, Philadelphia, 1994.
[2] RAFAEL BRU, LUDWIG ELSNER, AND MICHAEL NEUMANN, *Models of parallel chaotic iteration methods*, Linear Algebra Appl., 103 (1988), pp. 175–192.
[3] ———, *Convergence of infinite products of matrices and inner-outer iteration schemes*, Electron. Trans. Numer. Anal., 2 (1994), pp. 183–193.
[4] RAFAEL BRU AND ROBERT FUSTER, *Parallel chaotic extrapolated Jacobi method*, Appl. Math. Lett., 3 (1990), pp. 65–69.

[5] Rafael Bru, Violeta Migallón, and José Penadés, *Chaotic inner–outer iterative schemes*, in Proc. of the Fifth SIAM Conference on Applied Linear Algebra, J. G. Lewis, ed., SIAM Press, Philadelphia, 1994, pp. 434–438.
[6] ———, *Chaotic methods for the parallel solution of linear systems*, Comput. Systems Engrg., to appear.
[7] Wang Deren, *On the convergence of the parallel multisplitting AOR method*, Linear Algebra Appl., 154 (1991), pp. 473–486.
[8] Mouhamed Nabih El Tarazi, *Some convergence results for asynchronous algorithms*, Numer. Math., 39 (1982), pp. 325–340.
[9] Andreas Frommer, *On asynchronous iterations in partially ordered spaces*, Numer. Funct. Anal. Optim., 12 (1991), pp. 315–325.
[10] Andreas Frommer and Günter Mayer, *Convergence of relaxed parallel multisplitting methods*, Linear Algebra Appl., 119 (1989), pp. 141–152.
[11] ———, *On the theory and practice of multisplitting methods in parallel computation*, Computing, 49 (1992), pp. 63–74.
[12] Andreas Frommer and Bert Pohl, *A comparison result for multisplittings based on overlapping blocks and its application to waveform relaxation methods*, Technical Report 93-05, Eidgenössiche Technische Hochschule, Seminar für Angewandte Mathematik, Zurich, May 1993. Numer. Linear Algebra Appl., 2(4), (1995), to appear.
[13] ———, *Comparison results for splittings based on overlapping blocks*, in Proc. of the Fifth SIAM Conference on Applied Linear Algebra, J. G. Lewis, ed., SIAM Press, Philadelphia, 1994, pp. 29–33.
[14] Andreas Frommer and Daniel B. Szyld, *H -splittings and two-stage iterative methods*, Numer. Math., 63 (1992), pp. 345–356.
[15] ———, *Asynchronous two-stage iterative methods*, Numer. Math., 69 (1994), pp. 141–153.
[16] Robert Fuster, Violeta Migallón, and José Penadés, *Parallel chaotic extrapolated like-Jacobi methods*, Linear Algebra Appl., to appear.
[17] Gene H. Golub and Michael L. Overton, *Convergence of a two-stage Richardson iterative procedure for solving systems of linear equations* in Numer. Anal., G.A. Watson, ed., Proc. of the Ninth Biennial Conference, Dundee, Scotland, 1981, Lecture Notes in Mathematics 912, Springer Verlag, New York, 1982, pp. 128–139.
[18] ———, *The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems*, Numer. Math., 53 (1988), pp. 571–593.
[19] Apostolos Hadjidimos and Dimitrios Noustos, *On a matrix identity connecting iteration operators associated with a p-cyclic matrix*, Linear Algebra Appl., 182 (1993), pp. 157–177.
[20] Alston S. Householder, *The Theory of Matrices in Numerical Analysis*, Blaisdell, Waltham, Mass. 1964. Reprinted by Dover, New York, 1975.
[21] Mark T. Jones and Daniel B. Szyld, *Two-stage multisplitting methods with overlapping blocks*, Technical Report 94-31, Department of Mathematics, Temple University, Philadelphia, Pa., March 1994. Also Available as Technical Report CS-94-224, Computer Science Department, University of Tenessee, Knoxville, March 1994. Numer. Linear Algebra Appl., to appear.
[22] Paul J. Lanzkron, Donald J. Rose, and Daniel B. Szyld, *Convergence of nested classical iterative methods for linear systems*, Numer. Math., 58 (1991), pp. 685–702.
[23] Ivo Marek and Daniel B. Szyld, *Splittings of M-operators: Irreducibility and the index of the iteration operator*, Numer. Funct. Anal. Optim., 11 (1990), pp. 529–553.
[24] Günter Mayer, *Comparison theorems for iterative methods based on strong splittings*, SIAM J. Numer. Anal., 24 (1987), pp. 215–227.
[25] Arnold Neumaier, *New techniques for the analysis of linear interval equations*, Linear Algebra Appl., 58 (1984), pp. 273–325.
[26] Michael Neumann and Robert J. Plemmons, *Convergence of parallel multisplitting iterative methods for M-matrices*, Linear Algebra Appl., 88 (1987), pp. 559–573.
[27] Nancy K. Nichols, *On the convergence of two-stage iterative processes for solving linear equations*, SIAM J. Numer. Anal., 10 (1973), pp. 460–469.
[28] Dianne P. O'Leary and Robert E. White,. *Multi-splittings of matrices and parallel solution of linear systems*, SIAM J. on Algebraic and Discrete Math., 6 (1985), pp. 630–640.
[29] James M. Ortega and Werner C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York and London, 1970.
[30] Alexander M. Ostrowski, *Über die Determinanten mit überwiegender Hauptdiagonale*, Comentarii Mathematici Helvetici, 10 (1937), pp. 69–96.
[31] ———, *Determinanten mit überwiegender Hauptdiagonale und die absolute Konvergenz von linearen Iterationsprozessen*, Comentarii Mathematici Helvetici, 30 (1956), pp. 175–210.

[32] José Penadés, *Métodos Iterativos Paralelos para la Resolución de Sistemas Lineales basados en Multiparticiones*, PhD thesis, Departamento de Tecnología Informática y Computación, Universidad de Alicante, November 1993, in Spanish.

[33] Werner C. Rheinboldt and James S. Vandergraft, *A simple approach to the Perron-Frobenius theory for positive operators on general partially-ordered finite-dimentional linear spaces*, Math. Comp., 27 (1973), pp. 139–145.

[34] F. Robert, M. Charnay, and F. Musy, *Itérations chaotiques série-parallèle pour des équations non-linéaires de point fixe*, Aplikace Matematiky, 20 (1975), pp. 1–38.

[35] Hans Schneider, *Theorems on M-splittings of a singular M-matrix which depend on graph structure*, Linear Algebra Appl., 58 (1984), pp. 407–424.

[36] Xiaodi Sun, *Convergence of parallel multisplitting chaotic iterative methods*, Numer. Math. J. Chinese Univ. (Gao Deng Xue Xiao Ji Suan Shu Xue Xue Bao), 14 (1992), pp. 183–187, in Chinese.

[37] Daniel B. Szyld, *Synchronous and asynchronous two-stage multisplitting methods*, in Proc. of the Fifth SIAM Conference on Applied Linear Algebra, J. G. Lewis, ed., SIAM Press, Philadelphia, 1994, pp. 39–44.

[38] Daniel B. Szyld and Mark T. Jones, *Two-stage and multisplitting methods for the parallel solution of linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 671–679.

[39] Richard S. Varga, *Factorization and normalized iterative methods*, in Boundary Problems in Differential Equations, Rudolph E. Langer, ed., The University of Wisconsin Press, Madison, 1960, pp. 121–142..

[40] ———, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

[41] ———, *On recurring theorems on diagonal dominance*, Linear Algebra Appl., 13 (1976), pp. 1–9.

[42] Robert E. White, *Block Multisplittings and parallel iterative methods*, Comput. Methods Appl. Mech. Engrg., 64 (1987), pp. 567–577.

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

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