ELSEVIER

Journal of Computational and Applied Mathematics 123 (2000) 201–216

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

On asynchronous iterations

Andreas Frommer[a], Daniel B. Szyld[b, *, 1]

[a] *Fachbereich Mathematik, Bergische Universität GH Wuppertal, Gauss-Strasse 20, 42 097 Wuppertal, Germany*

[b] *Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, PA 19122-6094, USA*

Received 3 June 1999

Abstract

Asynchronous iterations arise naturally on parallel computers if one wants to minimize idle times. This paper reviews certain models of asynchronous iterations, using a common theoretical framework. The corresponding convergence theory and various domains of applications are presented. These include nonsingular linear systems, nonlinear systems, and initial value problems.

1. Introduction

With the advent of parallel computers, many new algorithms were devised or rediscovered for the new architectures. An important concept in the design of parallel algorithms is that of load balancing, which simply means that the work has to be approximately equally distributed among processors. Otherwise, some processors finish their task much earlier than others, and the waiting time (also called idle time) degrades the performance of the algorithm. This concept has been widely accepted as a requirement for efficient algorithms, and has dictated for example that when the geometric domain of a physical problem is divided into subdomains (to be processed by the different processors), each should be of approximately the same size.

In contrast to load balancing, the idea of asynchronous methods is to avoid processor idle time by eliminating as much as possible synchronization points, i.e., points at which a processor must wait for information from other processors. In this way, problems which naturally would decompose into processes of very different size, e.g., those with unstructured meshes, can do so without difficulty. The price one pays for this freedom is that some processors will perform extra computations, and it

* Corresponding author.

E-mail addresses: frommer@math.uni-wuppertal.de (A. Frommer), szyld@math.temple.edu (D.B. Szyld).

[1] Supported by National Science Foundation grants DMS-9625865 and DMS-9973219.

0377-0427/00/$ - see front matter
PII: S0377-0427(00)00409-X

is only when the load is not well balanced, or when communication between the processors is slow, that this approach is advantageous.

Since the publication of the pioneering paper in 1969 by Chazan and Miranker [21], the theory and application of asynchronous iterations has been studied and used by many authors. For early surveys of asynchronous iterative methods, see [3,13,14,33] (see also the recent papers [53,56]).

Asynchronous methods are not considered "mainstream" by many researchers, at least in numerical analysis. We believe that this is so in part because the load balancing requirement is so prevalent in the minds of many practitioners. Nevertheless, asynchronous methods are being increasingly used and studied, particularly so in connection with the use of heterogeneous workstation clusters where the available computational power of each processor becomes unpredictable. Experiments reported in the literature, e.g., in [18,35,42], show practical problems for which the asynchronous parallel times are about half to two-thirds of those reported for synchronous parallel times (which of course are much faster than sequential times); see further Section 4.3. In [18,35], asynchronous solutions of systems of several million variables are reported. In addition, asynchronous iterations are possibly the kind of methods which will allow the next generation of parallel machines to attain the expected potential. These machines are being designed today with thousands of processors.

Let us mention some recent papers where the application of asynchronous iterations to different areas is discussed: to the solution of partial differential equations [1,2,52]; to inverse problems in geophysics and oil exploration [48]; to continuous time Markov chains problems for queueing and stochastic automata networks [18]; to electrical power networks [9]; to network flow [60], to convex programming [58], and other optimization [25,26], and nonlinear problems [7,59,65]; and to singular systems of linear equations [5,50].

The purpose of this paper is to review some of the different models of asynchronous iterations which have been developed during the last three decades, using a common theoretical framework. We give some results on their convergence, and illustrate the use of these models and convergence results in various applications, including the iterative solution of linear systems, nonlinear systems, and initial value problems for systems of ordinary differential equations.

It is outside the scope of the paper to present a complete survey of the state-of-the-art in parallel asynchronous computations. Of the topics not covered we mention a few: analysis and implementation of stopping criteria [24,51] (and also [15]), enclosure methods for nonlinear systems of equations [32,34], the theory of multisplittings for the analysis of asynchronous iterations [17,20,55], and its application to domain decomposition methods using overlapping subdomains [7,35]. Our aim instead is to present a snapshot of some broad class of applications, together with a general theory which applies to them.

To that end, we present, in the next section, general computational and mathematical models representing asynchronous iterations. The computational models correspond to the way the methods are actually programmed in the parallel computers. The mathematical models are tools used to analyze the algorithms. In Section 3 we present very general convergence results which apply to these mathematical models, and in Section 4 we apply these results to specific problems.

2. Computational and mathematical models

To start, let us consider a structurally simple and quite general construct. Assume that we are given a product space $E = E_1 \times \cdots \times E_m$ and an application $H : E \to E$ whose components are

denoted H_i, i.e., we have

$$H : E \to E, \quad x = (x_1, \ldots, x_m) \to ((Hx)_1, \ldots, (Hx)_m), \tag{1}$$

where $x_i, (Hx)_i = H_i(x) \in E_i$, $i = 1, \ldots, m$. The problem at hand is to find a fixed point of H. A standard procedure is to approximate such fixed point by variants of the successive approximation procedure

$$x^{k+1} = H(x^k), \quad k = 0, 1, \ldots . \tag{2}$$

Assume for now that we are working with a (shared memory) parallel computer with p processors $P_1, \ldots, P_p$ ($p \leqslant m$) and associate a *block* of components $J_j \subseteq \{1, \ldots, m\}$ with each processor P_j. Then a parallel variant of the successive approximation procedure (2) can be implemented as follows (pseudocode for processor P_j):

Computational Model 2.1.

until convergence do
  read x from common memory
  compute $x_i^{\text{new}} = H_i(x)$ for $i \in J_j$
  overwrite x_i in common memory with x_i^{new}, $i \in J_j$.

If processors would wait for each other to complete each run through the loop we would indeed get a (parallel synchronous) implementation of the successive approximation scheme (2). Since here processors do not wait, we actually get a much less structured iterative process where, due to different run times for each loop, processors get out of phase. At a given time point, different processors will have achieved different numbers of iterations (the iteration number k in (2) looses its meaning in this context). No idle times occur, since processors never wait for each other.

In order to mathematically analyze the Computational Model 2.1, we now step the iteration counter k by 1 each time x is read from the common memory by some processor $P_{j(k)}$. Then this x is made up of components each of which has been written back to memory as the result of the computation belonging to some earlier iteration. We therefore have $x = (x_1^{s_1(k)}, \ldots, x_m^{s_m(k)})$ with iteration counts $s_\ell(k) \in \mathbb{N}_0$, $\ell = 1, \ldots, m$, prior to k, indicating the iteration when the ℓth component just read was computed. A set I^k is defined indicating which components are computed at the kth iteration, i.e., $I^k = J_{j(k)}$. Using these sets, and under the very weak assumptions (3) explained further below, the Computational Model 2.1 can be modeled mathematically according to the following definition; see, e.g., [33,57],

Definition 2.2. For $k \in \mathbb{N}$, let $I^k \subseteq \{1, \ldots, m\}$ and $(s_1(k), \ldots, s_p(k)) \in \mathbb{N}_0^m$ such that

$$\begin{aligned} &s_i(k) \leqslant k - 1 \quad \text{for } i \in \{1, \ldots, m\}, \quad k \in \mathbb{N}, \\ &\lim_{k \to \infty} s_i(k) = \infty \quad \text{for } i \in \{1, \ldots, m\}, \\ &|\{k \in \mathbb{N} : i \in I^k\}| = \infty \quad \text{for } i \in \{1, \ldots, m\}. \end{aligned} \tag{3}$$

Given an initial guess $x^0 \in E = E_1 \times \cdots \times E_m$, the iteration

$$x_i^k = \begin{cases} x_i^{k-1} & \text{for } i \notin I^k \\ H_i(x_1^{s_1(k)}, \ldots, x_m^{s_m(k)}) & \text{for } i \in I^k, \end{cases} \tag{4}$$

is termed an asynchronous iteration (with strategy I^k, $k \in \mathbb{N}$ and delays $d_i(k) = k - s_i(k)$, $i = 1,\ldots,n$, $k \in \mathbb{N}$).

The first hypothesis in (3) simply indicates that only components computed earlier (and not future ones) are used in the current approximation. The second one indicates that as the computation proceeds, eventually one reads newer information for each of the components. The third one indicates that no component fails to be updated as time goes on.

This mathematical model goes back at least to Baudet [10], although other authors had equivalent models; see the historical remarks in [57]. Note that Definition 2.2 includes as special cases the classical synchronous successive approximation method (2) ($s_i(k) = k - 1$, $I^k = \{1,\ldots,m\}$) as well as block Gauss–Seidel-type methods ($s_i(k) = k - 1$, $I^k = \{k \bmod m + 1\}$) or symmetric block Gauss–Seidel methods.

Let us mention at this point that asynchronous iterations on *local memory* machines (using message passing to communicate data) are also modeled by Definition 2.2.

The fundamental model (4) has a wide range of applications. Nevertheless, other various extensions to account for more general or more specific situations are possible. For example, some authors impose additional conditions on the sequence of delays $d_i(k) = k - s_i(k)$ such as being uniformly bounded; some others restrict them in such a way that overlap is not allowed; see some examples of these, e.g., in [57] and the bibliography therein. These additional restrictions appear to be necessary in the convergence theory for the solution of singular linear systems; see [5,41,50].

In several practical situations, the component H_i of H may be given only implicitly (or it may be expensive to compute) so that we will actually only compute an approximation (which may change at each step k) to $H_i(x_1^{s_1(k)},\ldots,x_m^{s_m(k)})$ in (4). We are then in a *non-stationary* setting, which includes in particular the case of *two-stage* iterations (with an "inner" and an "outer" iteration) which can be modeled by making H dependent of the iteration index k, i.e., we have the following process

$$x_i^k = \begin{cases} x_i^{k-1} & \text{for } i \notin I^k, \\ H_i^k(x_1^{s_1(k)},\ldots,x_m^{s_m(k)}) & \text{for } i \in I^k \end{cases} \tag{5}$$

with $H^k : E \to E$, for $k \in \mathbb{N}$, having the same fixed point as H.

One way to study the inner iterations is to consider a "splitting" of the application H of (1) into $K : E \times E \to E$ such that $K(x,x) = H(x)$, and the following model.

Computational Model 2.3.

until convergence do
  read (x) from common memory
  set $y = x$
    until convergence do
    compute $y_i^{\text{new}} = K_i(x, y)$ for $i \in J_j$
    overwrite x_i in common memory with y_i^{new}, $i \in J_j$
    set $y_i = y_i^{\text{new}}$ (in local memory).

This computational model describes in particular asynchronous methods with *flexible communication* (see [25,44]), in which new information is sent to the other processors as soon as it is computed,

even before the inner iterations have converged. A mathematical model for it can be obtained by introducing a second set of delays defined through iteration indices $r_\ell(k) \in \mathbb{N}_0$, $\ell = 1,\ldots,m$, $k \in \mathbb{N}$, satisfying the same first two hypotheses in (3), and by considering the following process, which is slightly more general than the one given in [38],

$$x_i^k = \begin{cases} x_i^{k-1} & \text{for } i \notin I^k, \\ K_i((x_1^{s_1(k)},\ldots,x_m^{s_m(k)}),(x_1^{r_1(k)},\ldots,x_m^{r_m(k)})) & \text{for } i \in I^k. \end{cases} \tag{6}$$

We note that further generalizations of these mathematical models are possible (and applicable to specific situations), where the domain of the application analogous to H (or K) consists of multiple copies of E, and each component of each copy of E may be subject to different delays; see [31,37,50].

It is crucial to realize that our Computational Models 2.1 and 2.3 do not preclude the blocks J_j to overlap, i.e., we may have $J_j \cap J_l \neq \emptyset$ for $j \neq l$. This situation cannot be modeled by the expression (1), but our mathematical models (4)–(6) are still applicable. In some instances, it turns out that a certain degree of overlapping together with a scheme for combining different contributions within the overlap will usually accelerate the overall iteration (see, e.g., [7,35]).

3. Convergence theory

A general convergence theorem for the asynchronous iteration (4) is the following result of Bertsekas [12] (see also [61]).

Theorem 3.1. *Assume that there are sets $E^k \subseteq E$ which satisfy*
(a) $E^k = E_1^k \times \cdots \times E_m^k$, $k \in \mathbb{N}_0$, (*box condition*)
(b) $H(E^k) \subseteq E^{k+1} \subseteq E^k$, $k \in \mathbb{N}_0$, (*nested sets condition*)
(c) *there exists x^* such that*

$$y^k \in E^k, \quad k \in \mathbb{N} \Rightarrow \lim_{k\to\infty} y^k = x^*$$

(*synchronous convergence condition*).

Then the sequence of asynchronous iterates x^k from (4) *converges to x^*, the unique fixed point of H, provided assumptions* (3) *hold.*

The idea of the proof is to show that starting in a box E^k, after some time all components x_i belong to some E_i^ℓ, $\ell > k$, and by collecting them we are now in the box E^{k+1}. A careful inspection of the proof of this result, e.g., in [12], reveals that we can easily obtain the following corollary for non-stationary iterations.

Corollary 3.2. *Replace* (b) *in Theorem* 3.1 *by*
(b′) $H^k(E^k) \subseteq E^{k+1} \subseteq E^k$, $k \in \mathbb{N}_0$.

Then the asynchronous nonstationary iterates x^k from (5) *converge to x^*, the unique common fixed point of all H^k.*

There are several special cases of Theorem 3.1 which merit further discussion. Let us first consider the case where each component space E_i is a normed linear space $(E_i, \|\cdot\|_i)$. Define $\|\cdot\|_w$ the weighted max-norm on E given as

$$\|x\|_w = \max_{i=1}^{m} \frac{\|x_i\|_i}{w_i}, \tag{7}$$

where $w = (w_1, \ldots, w_m)$ is a positive vector, i.e., $w_i > 0$ for $i = 1, \ldots, m$.

Theorem 3.3. *Assume that there exists $x^* \in E$ such that $H^k(x^*) = x^*$ for all k. Moreover, assume that there exists $\gamma \in [0, 1)$ and $w \in \mathbb{R}^m$ positive, such that for all k we have*

$$\|H^k(x) - x^*\|_w \leqslant \gamma \cdot \|x - x^*\|_w. \tag{8}$$

Then the asynchronous (non-stationary) iterates x^k from (5) converge to x^*, the unique common fixed point of all H^k.

For a proof, set $E^k = \{x \in E: \|x - x^*\|_w \leqslant \gamma^k \cdot \|x^0 - x^*\|_w\}$ and apply Corollary 3.2 (see [37]). Different proofs of similar theorems can be found in [28,29]. For the stationary case ($H^k = H$) the above theorem is known as El Tarazi's theorem [27].

An even more special case arises in the presence of *P-contractions*. The mapping H is called a P-contraction with respect to a fixed point x^*, if there exists a nonnegative matrix $P \in \mathbb{R}^{m\times m}$ with $\rho(P) < 1$ such that for all $x \in E$ we have

$$\begin{pmatrix} \|(Hx)_1 - x_1^*\|_1 \\ \vdots \\ \|(Hx)_m - x_m^*\|_m \end{pmatrix} \leqslant P \begin{pmatrix} \|x_1 - x_1^*\|_1 \\ \vdots \\ \|x_m - x_m^*\|_m \end{pmatrix},$$

where the inequality in $\mathbb{R}^m$ is componentwise [47]. It can be shown quite easily that a P-contraction with respect to x^* satisfies the assumption of Theorem 3.3 (w has to be taken as the Perron-vector of a positive matrix sufficiently close to P). We therefore have

Corollary 3.4 (Baudet [10])**.** *Assume that each H^k is a P-contraction with respect to x^* with P independent of k. Then the asynchronous (nonstationary) iterates x^k from (5) converge to x^*, the unique common fixed point of all H^k.*

The contraction conditions considered so far can be somewhat relaxed to account for situations where, instead of (8) one just has

$$x \neq x^* \Rightarrow \|Hx - x^*\|_w < \|x - x^*\|_w. \tag{9}$$

This is particularly interesting for certain M-functions and diagonally dominant functions in the sense of Moré [46] (see [30]). We mention here that if the implication in (9) is in both directions, such maps are called *paracontracting* (with respect to the weighted max morm) [28,29,50].

The following is a further generalization of El Tarazi's theorem which is applicable to process (6), and in particular to asynchronous methods with flexible communication [38].

Theorem 3.5. *Assume that there exists $x^* \in E$ such that $K(x^*, x^*) = x^*$. Moreover, assume, that there exists $\gamma \in [0,1)$ and a weighted max norm such that*

$$\|K(x,y) - x^*\|_w \leqslant \gamma \cdot \max\{\|x - x^*\|_w, \|y - x^*\|_w\} \quad \textit{for all } x, y \in E.$$

Then the asynchronous (with flexible communication) iterates x^k from (6) converge to x^*.

Another important special case arises for isotone mappings, i.e., mappings H where $x \leqslant y$ implies $Hx \leqslant Hy$. The following result goes back to Miellou [43]; see also [31] for the slightly more general version given here, as well as for a related result for isotonically decomposable mappings.

Theorem 3.6. *Assume that E is equipped with a partial ordering based on partial orderings for each component, and that the partial ordering is compatible with the topology on E so that we have*

$$x^0 \leqslant x^1 \leqslant x^2 \leqslant \cdots \leqslant x^k \leqslant \cdots \leqslant y^0 \Rightarrow \lim_{k\to\infty} x^k = x^* \textit{ exists and } x^* \leqslant y^0.$$

Assume further that H is continuous and isotone and that there exist $x^0 \leqslant y^0$ such that $x^0 \leqslant Hx^0 \leqslant Hy^0 \leqslant y^0$. Then the asynchronous iterates x^k from (4) converge to x^* with $x^* = Hx^*$.

For a proof, let $z^k = H(z^{k-1})$ with $z^0 = x^0$, let $x^* = \lim_{k\to\infty} z^k \leqslant y^0$ and take $E^k = \{x: z^k \leqslant x \leqslant x^*\}$ in Theorem 3.1.

4. Applications of the theory

In the remainder of the paper we show how the convergence theory for the general models (4)–(6) can be applied to a wide range of scientific problems.

4.1. Nonsingular linear systems

Let us start by considering a linear system of the form

$$Ax = b, \tag{10}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsingular. Let $A = M - N$ be a splitting of A, i.e., M is nonsingular. Let us define the *iteration operator*

$$H : \mathbb{R}^n \to \mathbb{R}^n, \quad x \to M^{-1}(Nx + b) \tag{11}$$

and analyze the convergence of its associated asynchronous iteration (4) in the case that $E = \mathbb{R}^n = E_1 \times \cdots \times E_n$ with $E_i = \mathbb{R}$, i.e. we allow each component to be treated individually. One example of such splitting is the *Jacobi* operator, when M is the diagonal part of A. Let $|H|$ denote the matrix of absolute values of entries of H.

Theorem 4.1. (a) *H is a P-contraction if and only if $\rho(|H|) < 1$.*

(b) *If $\rho(|H|) < 1$, then the asynchronous iteration (4) (with H) converges to x^*, the solution of $Ax = b$.*

(c) *If* $\rho(|H|)\geqslant 1$, *then there exists an asynchronous iteration i.e., a set of delays and strategies satisfying* (3), *and an initial guess* x^0 *such that the iterates* x^k *produced by* (4) *do not converge to* $x^* = A^{-1}b$.

Proof. (a) is a simple calculation, (b) follows from Corollary 3.4. Part (c) can be found in [21], where a version of a proof of (b) was given for the first time. Bertsekas and Tsitsiklis [13], Strikwerda [54], and Su et al. [56] suggested different constructions of the non-convergent sequences. □

We remark that the class of matrices with $\rho(|H|) < 1$ is just the class of H-matrices (see, e.g., [11]). H-matrices include M-matrices and strictly diagonally dominant or irreducibly diagonally dominant matrices [62].

If we think of grouping components together into (disjoint) blocks $B_i \subseteq \{1,\dots,n\}$, $i = 1,\dots,m$, we can write (10) in block notation as

$$\sum_{j=1}^{m} A_{ij}x_j = b_i, \quad i = 1,\dots,m, \tag{12}$$

where $x_j \in \mathbb{R}^{n_j}, n_j$ is the cardinality of $B_j, A_{ij} \in \mathbb{R}^{n_i \times n_j}$, $\sum_{i=1}^{m} n_i = n$. The corresponding *block Jacobi* operator H is given by (11), where now $M = \operatorname{diag}(A_{11},\dots,A_{mm})$ is the block diagonal of A which is assumed to be nonsingular, and $A = M - N$. In view of Theorem 3.3 we are now interested in cases where H is a contraction with respect to a weighted max-norm (7) where $\|\cdot\|_i$ is a norm on block i. Interestingly, this is again so for H-matrices.

Lemma 4.2. *Let* A *be an* H*-matrix and let* $x^* = A^{-1}b$. *Then there exist norms* $\|\cdot\|_i$ *on each block* i, $i \in \{1,\dots,m\}$ *such that with the* (*unweighted*) *max-norm* $\|x\| = \max_{i=1}^{n} \|x_i\|_i$, *the Block Jacobi operator* H *satisfies*

$$\|Hx - x^*\| \leqslant \gamma \cdot \|x - x^*\| \quad \text{with } \gamma \in [0,1).$$

Proof. One proceeds by showing $\rho(|H|) < 1$, which is true because the block Jacobi-splitting is an H-splitting [36]. This implies the existence of $v \in \mathbb{R}^n$, $v > 0$ with $|H|v \leqslant \gamma \cdot v$, $\gamma \in [0,1)$. One then defines $\|\cdot\|_i$ to be the weighted max-norm on block i with weights from the respective components of v. □

Alternatively, the following result can be helpful.

Lemma 4.3. *Let* $\|\cdot\|_i$ *be a norm on* $\mathbb{R}^{n_i}$, $i = 1,\dots,n$. *For each block* A_{ij} *let* $\|\cdot\|_{ij}$ *denote the corresponding matrix norm*

$$\|A_{ij}\|_{ij} = \max_{\|x_j\|_j = 1} \|A_{ij}x_j\|_i.$$

Let $P = (m_{ij}) \in \mathbb{R}^{n \times n}$ *with*

$$m_{ij} = \begin{cases} 0 & \text{if } i = j, \\ \|A_{ij}\|_{ij} \cdot \|A_{ii}^{-1}\|_{ii} & \text{if } i \neq j. \end{cases}$$

Then, if $\rho(P) < 1$ *we have that* H *is a* P*-contraction with respect to* x^*.

The proof is a straightforward computation.

In view of the above lemma one may thus generalize the concept of an H-matrix to block H-matrices, a block H-matrix being one for which

$$\begin{pmatrix} 1/\|A_{11}^{-1}\|_{11} & -\|A_{12}\|_{12} & \dots & -\|A_{1m}\|_{1m} \\ -\|A_{21}\|_{21} & 1/|A_{22}^{-1}\|_{22} & & -\|A_{2m}\|_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ -\|A_{m1}\|_{m1} & \dots & \dots & 1/\|A_{mm}^{-1}\|_{mm} \end{pmatrix}$$

is an M-matrix (see, e.g., [8,49]). This then implies that the block Jacobi operator is indeed a P-contraction, so that asynchronous iterations converge. Note also that in Lemma 4.3 we have freedom in the choice of the norms for each block.

Let us mention that the above results remain unchanged if we replace $\|A_{ii}^{-1}\|_{ii}$ in the definition of P by $1/m_{ii}$ if we assume each block A_{ii} to be *strongly accretive* with constant m_{ii} (see, e.g., [39]), i.e., we assume that for all $x_i \in (\mathbb{R}^{n_i}, \|\cdot\|_i)$ there exists a dual $l_i(x_i)$ of x_i such that

$$\langle A_{ii}x_i, l_i(x_i)\rangle \geqslant m_{ii}\|x_i\|_i^2.$$

Here, $\langle\cdot,\cdot\rangle$ denotes the bilinear form between $(\mathbb{R}^{n_i}, \|\cdot\|_i)$ as a Banach space and its dual, and $l_i(x_i)$ is an element of $(\mathbb{R}^{n_i}, \|\cdot\|_i)^*$ with

$$\|l_i(x_i)\|_i^* = \|x_i\|_i \quad \text{and} \quad \langle l_i(x_i), x_i\rangle = \|x_i\|_i^2.$$

In this asynchronous block Jacobi setting, each processor needs to solve a linear system with the coefficient matrix A_{ii} in (4) (see (11)). The solution of each of these systems by an iterative method based on a splitting $A_{ii} = F_i - G_i$ in $\mathbb{R}^{n_i} \times \mathbb{R}^{n_i}$ gives rise to a non-stationary process (5) with

$$H_i^k(x) = (F_i^{-1}G_i)^{\ell(i,k)}x_i + \sum_{j=0}^{\ell(i,k)-1} (F_i^{-1}G_i)^j(Nx + b),$$

where $F = \mathrm{diag}(F_1, \dots, F_m)$ and $G = \mathrm{diag}(G_1, \dots, G_m)$ are block diagonal, $M = F - G$, and $\ell(i,k)$ is the number of inner iterations. In the context of the Computational Model 2.3, we have a process of the form (6) with $K(x,y) = F^{-1}(Gy + Nx + b)$. Under suitable hypotheses on the splittings $A = M - N$, and $M = F - G$ (related to weak regular splittings and H-splittings), these methods can be shown to converge using Theorems 3.3 and 3.5, respectively (see [37,38] and also [66,67]).

In the case of overlapping variables, i.e., when the blocks B_i defining the partitions for (12) are not disjoint, one can still define a block Jacobi iteration with overlap by solving (or approximating) in different processors the linear systems

$$A_{ii}x_i = b_i - \sum_{j=1, j\neq i}^{m} A_{ij}x_j, \quad i = 1, \dots, m,$$

cf. the Computational Model 2.1. A consistent approximation to the solution of (10) can then be obtained by convex combinations of the elements in each component x_i belonging to nonempty intersections of the blocks B_i. The coefficients of these convex combinations, which can simply be ones and zeros, may change from one iteration to the next. A full mathematical description of this case will not be undertaken here, but we point out that for its analysis operators $H^k : E^m \to E^m$ are defined representing each asynchronous step. Convergence of this asynchronous additive algebraic Schwarz iteration is then obtained using Theorem 3.3 (see [7,35]).

We conclude this subsection with some comments on the case where A in (10) is singular. Any splitting $A = M - N$ (M nonsingular) of a singular matrix A produces an iteration matrix $H = M^{-1}N$ with 1 as an eigenvalue. Assume for simplicity that $H \geqslant 0$ so that $|H| = H$ and $\rho(H) = 1$ with 1 being an eigenvalue of H. The fixed points of H are the eigenvectors of H corresponding to the eigenvalue 1, and thus form a subspace of $\mathbb{R}^n$. Theorem 3.1 and its generalizations cannot be directly applied to this situation, since the nested set condition would normally be violated. In fact, part of the problem is that we no longer have a unique fixed point. In other words, the singular case lies outside the general theory of Section 3 and more restrictive hypotheses on the asynchronous iteration are needed in order to ensure convergence. For example, Lubachevsky and Mitra [41] consider the situation where (basically) H is irreducible, the starting vector is nonnegative and the asynchronous iteration is restricted in such a manner that for some fixed index i we always have $s_i(k) = k - 1$ whenever $i \in I^k$. In this case, one can then actually again construct a nested set of boxes E^l, $l = 0, 1, \ldots$ which converge to some singleton consisting of a fixed point of H, and for all l the asynchronous iterates x^k satisfy $x^k \in E^l$ for $l \geqslant l(k)$.

Another approach was taken in [5,50] using the concept of paracontractions. Again, additional restrictions have to be imposed on the asynchronous iteration in order to guarantee convergence. For example, Bahi [5] requires to do a "true" step of (synchronous) successive approximation every once in a while.

4.2. Nonlinear equations

Assume that we are given a nonlinear system of equations

$$F(x) = 0 \quad \text{where } F : D_F \subseteq \mathbb{R}^n \to \mathbb{R}^n. \tag{13}$$

Assume that this equation has exactly one solution x^* and let $H : D_H \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be an iteration function for this problem, i.e., x^* is the unique fixed point of H. Not too surprisingly, the following local version of Corollary 3.4 can be shown to hold [27].

Theorem 4.4. *Assume that x^* lies in the interior of D_H and that H is Fréchet differentiable at x^*. If $\rho(|H'(x^*)|) < 1$, then there exists a neighborhood $\mathcal{N}$ of x^* such that the asynchronous iterates* (4) *converge to x^*, provided $x^0 \in \mathcal{N}$.*

The standard iteration operator for (13) is the Newton operator $H_{\mathrm{N}}(x) = x - F'(x)^{-1}F(x)$. Here, $H'(x^*) = 0$ so that Theorem 4.4 can be applied. However, for practical reasons it is mandatory in asynchronous computations that the components of H can be evaluated individually at much lower cost than all components together. Due to the presence of the term $F'(x)^{-1}$ in H_{N} this is usually not the case. A favorable situation arises, for example, in the Durand–Kerner method [22,23], for the simultaneous computation of all zeros of a polynomial, which is Newton's method on the equations expressing the coefficients of the polynomial via the elementary symmetric functions on its roots, but where one has a simple, explicit formula for each component of H_{N}.

If $D(x)$ denotes the diagonal part of $F'(x) = D(x) - B(x)$, the Newton–Jacobi operator is given as $H_{\mathrm{NJ}}(x) = x - D^{-1}(x)F(x)$ [47]. We can regard H_{NJ} as a two-stage approximation to H_{N} with one inner step. Here it is trivial that components of H_{NJ} can be evaluated individually. We have $H'_{\mathrm{NJ}}(x^*) = D(x^*)^{-1}B(x^*)$. So, as in the remark following Theorem 4.1, we see that we get local

convergence of the asynchronous iterates if $H'(x^*)$ is an H-matrix. It is important to notice that functions F satisfying the above conditions arise in several applications, including discretizations of elliptic partial differential equations.

In a general way, we can define the components of the nonlinear Jacobi operator H_J for (13) through

$$y_i = (H_J)_i(x) \quad \Leftrightarrow \quad F_i(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n) = 0.$$

The generalization to a block nonlinear Jacobi operator should be obvious (cf. (16)). Another asynchronous approach to Newton's method includes the work by Bojańczyk [16], where the Newton operator can be viewed as $K(x, y) = x - F'(y)^{-1}F(x)$ (cf. Theorem 3.5). Yet another extension is to consider quasi-Newton methods [64].

Interestingly, there are several important situations where global convergence of asynchronous iterates for H_J can be proved. As a generalization of Lemma 4.3 it was shown in [39] that Theorem 3.3 holds for the (block) Jacobi operator H_J for certain mildly non-linear functions arising in discretizations of elliptic boundary value problems, the obstacle problem or the Hamilton–Jacobi–Bellman problem.

If the function F is an M-function (see [47]), one can also prove global convergence of the asynchronous iterates for H_J, now using Theorem 3.6 [68]. Generalizations for further classes of functions, including a nonlinear analog of H-matrices have been developed in [30].

4.3. Waveform relaxation

Waveform relaxation methods are parallel iterative methods for initial value problems based on a splitting principle. They were developed at the beginning of the 1980s for the simulation of electronic circuits (see [63]).

Consider the initial value problem

$$\begin{aligned} &\dot{x}(t) = F(t, x(t)), \ t \in [0, T], \\ &x(0) = x_0, \end{aligned} \tag{14}$$

where $F : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$, $x(t) \in \mathbb{R}^n$. Instead of solving (14) directly, waveform relaxation methods split the function F into a function $G : [0, T] \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ with $G(t, x, x) = F(t, x)$ for all t and x. Starting with a function $x^0 : [0, T] \to \mathbb{R}^n$ satisfying $x^0(0) = x_0$ (for example $x^0 \equiv x_0$) one then successively solves the systems

$$\begin{aligned} &\dot{x}^k(t) = G(t, x^k(t), x^{k-1}(t)), \ t \in [0, T], \\ &x^k(0) = x_0 \end{aligned} \tag{15}$$

for $k = 1, 2, \ldots$. Here, the function x^{k-1} is known and x^k is to be determined.

Note that the familiar *Picard iteration*

$$\begin{aligned} &\dot{x}^k(t) = F(t, x^{k-1}(t)), \ t \in [0, T], \\ &x^k(0) = x_0 \end{aligned}$$

is a special waveform relaxation method where $G(t, x, y) = F(t, y)$. Since the Picard iteration usually converges very slowly, one is interested in better splittings G which yield faster convergence. One

possibility is to take a block Jacobi type splitting which, given a block decomposition $x=(x_1,\dots,x_m)$ (and similarly $F(t,x)=((F(t,x))_1,\dots,(F(t,x))_m)$) defines block i of G as

$$(G(t,x,y))_i := (F(t,(x_1,\dots,x_{i-1},y_i,x_{i+1},\dots,x_m)))_i. \tag{16}$$

From now on, let us assume that G satisfies a one-sided Lipschitz condition with respect to the first argument, and a strong Lipschitz-condition with respect to the second argument, i.e., one has

$$\langle G(t,x,y)-G(t,\bar{x},y),x-\bar{x}\rangle \leqslant l\cdot\|x-\bar{x}\|^2, \qquad \|G(t,x,y)-G(t,x,\bar{y})\|\leqslant L\cdot\|y-\bar{y}\| \tag{17}$$

for all $t\in[0,T],]$; $x,\bar{x},y,\bar{y}\in\mathbb{R}^n$. The Lipschitz condition with respect to the first argument implies that (15) has a unique solution x^k. Moreover, since $F(t,x)=G(t,x,x)$, the function F also satisfies a (one-sided) Lipschitz condition which shows that (14) has a unique solution x^*.

Iteration (15) defines an operator H where $y=H(x)$ if

$$\begin{aligned}&\dot{y}(t)=G(t,y(t),x(t)),\ t\in[0,T],\\ &y(0)=x_0.\end{aligned} \tag{18}$$

Here, H acts on a space of functions. We take this space to be $\mathscr{C}([0,T],\mathbb{R}^n)$, the Banach space of all continuous functions from $[0,T]$ to $\mathbb{R}^n$ with the norm

$$|||x|||_\alpha = \max_{t\in[0,T]} e^{-\alpha t}\|x(t)\|_\infty, \quad \alpha>0. \tag{19}$$

It is crucial for us to notice that $|||\cdot|||_\alpha$ is in fact a maximum norm over the block components of x since

$$|||x|||_\alpha = \max_{i=1}^{m}]|x_i|[_\alpha, \tag{20}$$

where for continuous $f:[0,T]\to\mathbb{R}^{n_i}$ (n_i is the dimension of block i) the norm $]|\cdot|[_\alpha$ is given as

$$]|f|[_\alpha = \max_{t\in[0,T]} \mathrm{e}^{-\alpha t}\|f(t)\|_\infty. \tag{21}$$

The following theorem proved very recently by Martin [42] shows that for α sufficiently large the operator H from (18) is contracting with respect to the max-norm $|||x|||_\alpha$. Therefore, Theorem 3.3 shows that asynchronous iterations for H converge.

Theorem 4.5. *Assume that G satisfies* (17) *and let $x^*\in\mathscr{C}([0,T],\mathbb{R}^n)$ be the solution of* (14). *There exists α sufficiently large such that for all $x\in\mathscr{C}([0,T],\mathbb{R}^n)$ we have*

$$|||H(x)-x^*|||_\alpha \leqslant \tfrac{1}{2}\cdot|||x-x^*|||_\alpha.$$

Proof. Denote $y=H(x)$ and $u(t)=y(t)-x^*(t)$.

Then $\dot{u}(t)=G(t,y(t),x(t))-G(t,x^*(t),x^*(t))$, so that from (17) we get

$$\langle\dot{u}(t),u(t)\rangle\leqslant l\cdot\|u(t)\|^2+L\cdot\|u(t)\|\cdot\|x(t)-x^*(t)\|, \quad t\in[0,T].$$

Since $\langle\dot{u}(t),u(t)\rangle=(\mathrm{d}/\mathrm{d}t)\|u(t)\|^2$ whenever $u(t)\neq 0$, a standard argument from the theory of differential inequalities (see, e.g., [19]) yields

$$\|u(t)\|\leqslant L\cdot\mathrm{e}^{|l|t}\int_0^t\|x(s)-x^*(s)\|\mathrm{e}^{-|l|s}\,\mathrm{d}s.$$

Turning from $\|\cdot\|$ to $\|\cdot\|_\infty$ we get

$$\|u(t)\|_\infty \leqslant cL \cdot \mathrm{e}^{|l|t} \int_0^t \|x(s) - x^*(s)\|_\infty \mathrm{e}^{-|l|s}\,\mathrm{d}s$$

with some constant $c > 0$. From the latter inequality we conclude

$$\begin{aligned}\|u(t)\|_\infty &\leqslant cL \cdot \mathrm{e}^{|l|t} \int_0^t \|x(s) - x^*(s)\|_\infty \mathrm{e}^{-\alpha s}\mathrm{e}^{\alpha s}\mathrm{e}^{-|l|s}\,\mathrm{d}s \\ &\leqslant cL \cdot \mathrm{e}^{|l|t} \cdot |||x - x^*|||_\alpha \int_0^t \mathrm{e}^{(\alpha-|l|)s}\,\mathrm{d}s.\end{aligned}$$

For $\alpha > |l|$ the last integral is less than $\mathrm{e}^{(\alpha-|l|)t}/(\alpha - |l|)$, so that we get

$$\|u(t)\|_\infty e^{-\alpha t} \leqslant \frac{cL}{\alpha - |l|} |||x - x^*|||_\alpha, \quad t \in [0, T]$$

and thus

$$|||u|||_\alpha \leqslant \frac{cL}{\alpha - |l|} \cdot |||x - x^*|||_\alpha.$$

So taking $\alpha > 2cL + |l|$ proves the theorem. □

For an infinite time interval [45] gives a convergence result for asynchronous iterations under much more restrictive assumptions. For differential-algebraic systems and asynchronous iterations, see [4,6].

In [42] several numerical results on asynchronous waveform relaxation methods have been reported. These computations were done on a cluster of 8 SUN Ultra Sparc 10 workstations, connected via fast Ethernet. The example considered arises from a model describing the penetration of radioactively marked antibodies into cancerous tissue (MedicalAkzo from [40]). The total system size was 400, and the splitting G was obtained by a block Jacobi decomposition of F assigning a block of 50 to each processor. The asynchronous variant then needed only 66% (120 s) of the time required for the synchronous variant (180 s).

References

[1] D. Amitai, A. Averbuch, M. Israeli, S. Itzikowitz, On parallel asynchronous high-order solutions of parabolic PDEs, Numer. Algorithms 12 (1996) 159–192.

[2] D. Amitai, A. Averbuch, M. Israeli, S. Itzikowitz, Implicit-explicit parallel asynchronous solver for PDEs, SIAM J. Sci. Comput. 19 (1998) 1366–1404.

[3] D. Amitai, A. Averbuch, M. Israeli, S. Itzikowitz, E. Turkel, A survey of asynchronous finite-difference methods for parabolic PDEs on multiprocessors, Appl. Numer. Math. 12 (1993) 27–45.

[4] J. Bahi, Asynchronous Runge–Kutta methods for differential-algebraic systems, Adv. Parallel Comput. 11 (1996) 205–212.

[5] J. Bahi, Algorithmes parallèles asynchrones pour des systèmes singuliers, C.R. l'Acad. Sci. Sér. 326 (1998) 1421–1425.

[6] J. Bahi, E. Griepentrog, J.-C. Miellou, Parallel treatment of a class of differential-algebraic systems, SIAM J. Numer. Anal. 23 (1996) 1969–1980.

[7] J. Bahi, J.-C. Miellou, K. Rhofir, Asynchronous multisplitting methods for nonlinear fixed point problems, Numer. Algorithms 15 (1997) 315–345.
[8] Z.-Z. Bai, V. Migallón, J. Penadés, D.B. Szyld, Block and asynchronous two-stage methods for mildly nonlinear systems, Numer. Math. 82 (1999) 1–20.
[9] B. Barán, E. Kaszkurewicz, A. Bhaya, Parallel asynchronous team algorithms: Convergence and performance analysis, IEEE Trans. Parallel Distributed Systems 7 (1996) 677–688.
[10] G.M. Baudet, Asynchronous iterative methods for multiprocessors, J. Assoc. Comput. Mach. 25 (1978) 226–244.
[11] A. Berman, R.J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, Academic Press, New York, Third edition, 1979 (reprinted by SIAM, Philadelphia, 1994).
[12] D.P. Bertsekas, Distributed asynchronous computation of fixed points, Math. Programming 27 (1983) 107–120.
[13] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[14] D.P. Bertsekas, J.N. Tsitsiklis, Some aspects of parallel and distributed iterative algorithms – a survey, Automatica 27 (1991) 3–21.
[15] K. Blathras, D.B. Szyld, Y. Shi, Timing models and local stopping criteria for asynchronous iterative algorithms. J. Parallel Distributed Comput. 58 (1999) 446–465.
[16] A. Bojańczyk, Optimal asynchronous Newton method for the solution of nonlinear equations, J. Assoc. Comput. Mach. 31 (1984) 792–803.
[17] R. Bru, V. Migallón, J. Penadés, D.B. Szyld, Parallel, synchronous and asynchronous two-stage multisplitting methods, Electron. Trans. Numer. Anal. 3 (1995) 24–38.
[18] P. Buchholz, M. Fischer, P. Kemper, Distributed steady state analysis using Kronecker algebra, in: B. Plateau, W.J. Stewart (Eds.), Numerical Solution of Markov Chains (NSMC '99), Prensas Universitarias de Zaragoza, 1999, pp. 76–95.
[19] K. Burrage, Sequential and Parallel Methods for Ordinary Differential Equations, Oxford University Press, Oxford, 1995.
[20] M.J. Castel, V. Migallón, J. Penadés, Convergence of non-stationary multisplitting methods for Hermitian positive definite matrices, Math. Comput. 67 (1998) 209–220.
[21] D. Chazan, W. Miranker, Chaotic relaxation, Linear Algebra Appl. 2 (1969) 199–222.
[22] M. Cosnard, P. Fraigniaud, Asynchronous Durand-Kerner and Aberth polynomial root finding methods on a distributed memory multicomputer, in: D. Evans, G. Joubert, F. Peters (Eds.), Parallel Computing 89, North-Holland, Amsterdam, 1990.
[23] E. Durand, Solutions numériques des équations algébriques, Vol. I, Masson, Paris, 1960.
[24] D. El Baz, A method of terminating asynchronous iterative algorithms on message passing systems, Parallel Algorithms Appl. 9 (1996) 153–158.
[25] D. El Baz, D. Gazen, J.-C. Miellou, P. Spiteri, Mise en œuvre de méthodes itératives asynchrones avec communication flexible, application à la résolution d'une classe de problèmes d'optimisation, Calculateurs Parallèles 8 (1996) 393–410.
[26] D. El Baz, P. Spiteri, J.-C. Miellou, Distributed asynchronous iterative methods with order intervals for a class of nonlinear optimization problems, J. Parallel Distributed Comput. 38 (1996) 1–15.
[27] M.N. El Tarazi, Some convergence results for asynchronous algorithms, Numer. Math. 39 (1982) 325–340.
[28] L. Elsner, I. Koltracht, M. Neumann, On the convergence of asynchronous paracontractions with application to tomographic reconstruction from incomplete data, Linear Algebra Appl. 130 (1990) 65–82.
[29] L. Elsner, I. Koltracht, M. Neumann, Convergence of sequential and asynchronous nonlinear paracontractions, Numer. Math. 62 (1992) 305–319.
[30] A. Frommer, Generalized nonlinear diagonal dominance and applications to asynchronous iterative methods, J. Comput. Appl. Math. 38 (1991) 105–124.
[31] A. Frommer, On asynchronous iterations in partially ordered spaces, Numer. Functional Anal. Optim. 12 (1991) 315–325.
[32] A. Frommer, Asynchronous iterations for enclosing solutions of fixed point problems, in: L. Atanassova, J. Herzberger (Eds.), Computer Arithmetic and Enclosure Methods, Elsevier, Amsterdam, 1992, pp. 243–252.
[33] A. Frommer, Parallele asynchrone Iterationen. in: J. Herzberger (Ed.), Wissenschaftliches Rechnen, Akademie, Berlin, 1995, pp. 187–231 (Chapter 4).

[34] A. Frommer, H. Schwandt, Asynchronous parallel methods for enclosing solutions of nonlinear equations, J. Comput. Appl. Math. 60 (1995) 47–62.
[35] A. Frommer, H. Schwandt, D.B. Szyld, Asynchronous weighted additive Schwarz methods, Electron. Trans. Numer. Anal. 5 (1997) 48–61.
[36] A. Frommer, D.B. Szyld, H-splittings and two-stage iterative methods, Numer. Math. 63 (1992) 345–356.
[37] A. Frommer, D.B. Szyld, Asynchronous two-stage iterative methods, Numer. Math. 69 (1994) 141–153.
[38] A. Frommer, D.B. Szyld, Asynchronous iterations with flexible communication for linear systems, Calculateurs Parallèles 10 (1998) 421–429.
[39] L. Giraud, P. Spitéri, Résolution parallèle de problèmes aux limites non lineaires, Math. Modelling Numer. Anal. 25 (1991) 579–606.
[40] W. Lioen, J. de Swart, W. van der Veen, Test set for IVP solvers. CWI Amsterdam, Online at `http://www.cwi.nl/cwi/projects/IVPtestset`.
[41] B. Lubachevsky, D. Mitra, A chaotic asynchronous algorithm for computing the fixed point of a nonnegative matrix of unit spectral radius, J. Assoc. Comput. Mach. 33 (1986) 130–150.
[42] S. Martin, Parallele asynchrone Waveform-Relaxation für Anfangswertaufgaben, Master's thesis, Department of Mathematics, University of Wuppertal, 1999.
[43] J.C. Miellou, Itérations chaotiques à retards; études de la convergence dans le cas d'espaces partiellement ordonnés, C.R. l'Acad. Sci. Paris, Sér. A 280 (1975) 233–236.
[44] J.C. Miellou, D. El Baz, P. Spitéri, A new class of asynchronous iterative algorithms with order intervals, Math. Comput. 67 (1998) 237–255.
[45] D. Mitra, Asynchronous relaxations for the numerical solution of differential equations by parallel processors, SIAM J. Sci. Statist. Comput. 8 (1987) s43–s58.
[46] J.J. Moré, Nonlinear generalizations of matrix diagonal dominance with application to Gauss–Seidel iterations, SIAM J. Numer. Anal. 9 (1972) 357–378.
[47] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York and London, 1970.
[48] V. Pereyra, Asynchronous distributed solution of large scale nonlinear inversion problems, Appl. Numer. Math. 30 (1999) 31–40.
[49] B. Polman, Incomplete blockwise factorization of (block) H-matrices, Linear Algebra Appl. 90 (1987) 119–132.
[50] M. Pott, On the convergence of asynchronous iteration methods for nonlinear paracontractions and consistent linear systems, Linear Algebra Appl. 283 (1998) 1–33.
[51] S.A. Savari, D.P. Bertserkas, Finite termination of asynchronous iterative algorithms, Parallel Comput. 22 (1996) 39–56.
[52] W. Shangmeng, L. Xiaomei, A class of stable difference schemes for linear elliptic PDE's and their asynchronous parallel computation, Wuhan Univ. J. Nat. Sci. 1 (1996) 553–556.
[53] J. Šilc, B. Robič, T. Ungerer, Asynchrony in parallel computing: From dataflow to multithreading, Parallel Distributed Comput. Practices 1 (1998) 3–30.
[54] J.C. Strikwerda, A convergence theorem for chaotic asynchronous relaxation, Linear Algebra Appl. 253 (1997) 15–24.
[55] Y. Su, A. Bhaya, Convergence of pseudocontractions and applications to two-stage and asynchronous multisplittings, Technical Report SB97/NACAD/05, Laboratory for High Performance Computing, Graduate School of Engineering, Federal University of Rio de Janeiro, 1997.
[56] Y. Su, A. Bhaya, E. Kaszkurewicz, V.S. Kozyakin, Further results on convergence of asynchronous linear iterations, Linear Algebra Appl. 281 (1998) 11–24.
[57] D.B. Szyld, Different models of parallel asynchronous iterations with overlapping blocks, Comput. Appl. Math. 17 (1998) 101–115.
[58] X.-C. Tai, P. Tseng, An asynchronous space decomposition method, in: C.-H. Lai, P.E. Bjørstad, M. Cross, O.B. Widlund (Eds.), Eleventh International Conference on Domain Decomposition Methods (London), DDM.org Press, 1999, pp. 348–358.
[59] P. Tseng, On the rate of convergence of a partially asynchronous gradient projection algorithm, SIAM J. Optim. 1 (1991) 603–619.

[60] P. Tseng, D.P. Bertsekas, J.N. Tsitsiklis, Partially asynchronous, parallel algorithm for network flow and other problems, SIAM J. Control Optim. 28 (1990) 678–710.
[61] A. Üresin, M. Dubois, Sufficient conditions for the convergence of asynchronous iterations, Parallel Comput. 10 (1989) 83–92.
[62] R.S. Varga, On recurring theorems on diagonal dominance, Linear Algebra Appl. 13 (1976) 1–9.
[63] J. White, A. Sangiovanni-Vincentelli, Relaxation Techniques for the Simulation on VLSI Circuits, Kluwer Academic Publishers, Boston, 1987.
[64] J.-J. Xu, Convergence of partially asynchronous block quasi-Newton methods for nonlinear systems of equations, J. Comput. Appl. Math. 103 (1999) 307–321.
[65] J. Aral, V. Migallón, J. Penadés, Non-stationary parallel multisplitting algorithms for almost linear systems, Numer. Linear Algebra Appl. 6 (1999) 79–92.
[66] R. Bru, L. Elsner, M. Neumann, Models of parallel chaotic iteration methods, Linear Algebra Appl. 103 (1988) 175–192.
[67] R. Bru, L. Elsner, M. Neumann, Convergence of infinite products of matrices and inner–outer iteration schemes, Electron. Trans. Numer. Anal. 2 (1994) 183–193.
[68] D. El Baz, M-functions and parallel asynchronous algorithms, SIAM J. Numer. Anal. 27 (1990) 136–140.