Asynchronous Iterations with Flexible Communication for Linear Systems

Andreas Frommer*
Daniel B. Szyld†

RÉSUMÉ. *Nous formulons plusieurs modèles algorithmiques et mathématiques pour des itérations asynchrones, y inclus le modèle des communications flexibles de Miellou, El Baz et Spitéri. Nous établissons la convergence de ces itérations dans le cadre d'une décomposition "deux étapes" pour des systèmes linéaires.*

ABSTRACT. *We formulate different computational and mathematical models for asynchronous parallel iterations, including the recent flexible communication model by Miellou, El Baz and Spitéri. Within this framework we establish the convergence of asynchronous two-stage schemes for solving linear systems.*

1. Introduction

Recently, Miellou, El Baz and Spitéri [7] introduced a new 'flexible' communication scheme for asynchronous iterative methods in a two-stage setting. They analyzed the convergence of this scheme under monotonicity conditions involving M-functions and M-super applications, assuming that the initial guess is a supersolution. They also assumed that the communication delays remain bounded. The purpose of the present note is to perform a similar analysis for flexible communications in the case where the operators involved are contractions with respect to a weighted maximum norm, without assuming bounded delays, and without any restrictions on the initial guess. In this sense we generalize the classical result of El Tarazi [2] to asynchronous iterations with flexible communication. The motivating special case behind this study is that of asynchronous two-stage iterative methods for solving linear systems. This special case will therefore be given particular attention. Since space restrictions prevent us from presenting other special cases or generalizations, let us just mention that our approach is also interesting for nonlinear asynchronous iterations (where the two-stage paradigm

* Fachbereich Mathematik, Universität Wuppertal, D-42097 Wuppertal, Germany, *frommer@math.uni-wuppertal.de*

† Department of Mathematics, Temple University, Philadelphia, PA 19122-6094, USA, *szyld@euclid.math.temple.edu*. Supported in part by the National Science Foundation grant DMS-9625865

comes up very naturally), and in asynchronous algebraic Schwarz type iterations including domain decomposition approaches with and without overlap; see [3, 9].

The paper is organized as follows: Section 2 explains the general framework we are considering and contains a systematic treatment of different communication schemes in asynchronous iterations. Section 3 gives our general convergence analysis whereas Section 4 contains more details for the specific case of linear systems.

2. Communication schemes

As our motivating example, we consider a linear system in $\mathbb{R}^n$ written as

$$Ax = b, \quad A \in \mathbb{R}^{n\times n} \text{ non-singular}.$$

A splitting $A = M - N$ (M non-singular) gives rise to the iteration

$$Mx^{k+1} = Nx^k + b, \;\; k = 0, 1, \ldots. \tag{1}$$

Let us assume that M is block diagonal, so that solving Eq. [1] for x^{k+1} decouples into L independent subproblems. To be specific, let M have the block diagonal structure $M = \text{diag}(M_1, \ldots, M_L)$ with non-singular matrices $M_l \in \mathbb{R}^{n_l\times n_l}$, $l = 1, \ldots, L, \; \sum_{l=1}^{L} n_l = n$. Moreover, assume that systems with the matrices M_l are still difficult to be solved directly, so that we approximate the block components x_l^{k+1} of x^{k+1} by performing several steps ($s(k)$, say) of an 'inner' iteration based on splittings $M_l = F_l - G_l, l = 1, \ldots, L$. Defining

$$F = \text{diag}(F_1, \ldots, F_L), \;\; G = \text{diag}(G_1, \ldots, G_L)$$

we then arrive at the following overall 'two-stage' iterative process:

for $k = 0, 1, \ldots$
    $y^{k,0} = x^k$
    for $p = 0, 1, \ldots, s(k) - 1$
        $Fy^{k,p+1} = Gy^{k,p} + Nx^k + b$
    $x^{k+1} = y^{k,s(k)}$

Such processes have been analyzed in [4, 6]. We now consider their realization on a parallel computer with L processors with a shared memory. Assume that each processor is assigned to update a given block x_l of x. Then a generic formulation of the resulting implementation for processor l reads

Algorithm 1

for $k = 0, 1, \ldots$
    read(x)
    $c_l = (Nx + b)_l$, set $y_l = x_l$
    for $p = 0, 1, \ldots, s(k) - 1$
        $z_l \leftarrow G_l y_l$

$y_l \leftarrow F_l^{-1}(z_l + c_l)$
write(y_l) to x_l
synchronize

The synchronize statement means that all processors wait for each other to complete their work for loop k. This kind of mechanism is necessary to ensure that the algorithm will indeed represent an implementation of the two-stage process just described.

On the other hand, synchronization may produce significant idle times on several processors if the computational work is not well balanced or if the processors work at different speeds (as is the case in heterogeneous or multi-user networks). The idea of an *asynchronous* modification of Algorithm 1 is precisely to remove the synchronization barrier. This results in a less structured iterative process, since at a given time different processors may work on different iteration steps k. The advantage is that we now have removed all idle times due to synchronization, and numerical experiments show that this can result in significant savings in run time as compared to the synchronous variant. In the two-stage setting we have different 'degrees' of asynchronism depending on how processors are allowed to access (read and write) the global iterate x. To systematically describe these variants we place ourselves in a more general context than before and assume that the two-stage method is defined via an iteration function $S : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n,\ (x, y) \mapsto S(x, y)$. As before, block components of S are denoted by S_l, and for the sake of practical feasibility we also assume that $S_l(x, y)$ only depends on x and the l-th block of y, i.e. $S_l(x, y) = S_l(x, y_l)$. In the linear situation described before we have $S(x, y) = F^{-1}(Gy + Nx + b)$.

for $k = 0, 1, \ldots$	for $k = 0, 1, \ldots$	for $k = 0, 1, \ldots$
read(x)	read(x)	for $p = 0, \ldots, s(k) - 1$
for $p = 0, \ldots, s(k) - 1$	for $p = 0, \ldots, s(k) - 1$	read(x)
$y_l \leftarrow S_l(x, y_l)$	$y_l \leftarrow S_l(x, y_l)$	$y_l \leftarrow S_l(x, y_l)$
write(y_l) as x_l	write(y_l) as x_l	write(y_l) as x_l
outer asynchronous	flexible	totally asynchronous

Figure 1: Computational models for different asynchronous schemes

Figure 1 describes the three different levels of asynchronism we are considering here: In the 'outer asynchronous' scheme (our terminology follows [5] and [7]), x is read and written only once per outer iteration. In the 'flexible' scheme, all inner iterates are made available to the other processors by writing them to x as soon as they have been computed. We remark that the 'flexible' scheme here is only one of several possible realizations of the method presented in [7]. Another variant would consist of reading x anew in each inner loop just before each step $y_l \leftarrow S_l(x, y_l)$. In this manner, one would systematically use the latest information available. Note that then there is no need for the inner loop any more. A tacit assumption behind the first and second scheme in Figure 1 is that it be possible to more or less 'split' the evaluation of S into a part which only depends on x (and which thus has to be done only once per outer iteration) and another part which depends on y. In this philosophy, the variant just described and the third, 'totally asynchronous', scheme from Figure 1 (like the alternative flexible scheme) appear less attractive since it requires to evaluate the 'x part' anew in each inner iteration; see further the discussion in [5]. We include this scheme mainly for the sake of completeness.

According to the terminology developed in [9], the pseudocodes given in Figure 1 may be considered as being *computational* models since they describe the way the methods would be implemented on a parallel computer. In order to perform a mathematical (convergence) analysis of the different schemes, we have to formulate *mathematical* models of these schemes. For this purpose we now count iteration steps in a completely different manner: We use a 'global' iteration counter m which is stepped by one any time a block component of x has been changed through a write operation by some processor. Let I_m denote that block component (or more generally: set of block components). These components of x have been computed by the respective processors using the value of x obtained through the preceding read statement. This vector x, in turn, consists of components which have been computed at some previous 'global' iteration which may differ from component to component. In this manner we arrive at the following mathematical model for the outer asynchronous iteration

$$x_l^{m+1} = \begin{cases} S_l^{s(m)}\left((x_1^{r_1(m)}, \ldots, x_L^{r_L(m)}), x_l^{r_l(m)}\right) & \text{for } l \in I_m \\ x_l^m & \text{otherwise} \end{cases} . \qquad [2]$$

Here, $S^p(x, y)$ denotes the p-fold composition with respect to the second argument, i.e. $S^1(x, y) = S(x, y)$ and $S^{p+1}(x, y) = S(x, S^p(x, y)),\ p = 1, 2, \ldots$. The indices $r_l(m)$ refer to previous iterations, i.e. $r_l(m) \leq m$ for all m and l.

In a similar manner we can get a mathematical model for the flexible communication scheme. Again, we step the global iteration counter any time x is changed through a write operation. Since these occur within the inner loop, there is no need to consider compositions of S any more, but we have now to account for the fact that the second argument y^l of S in the inner loop will usually correspond to an l-th component of a 'global' iteration vector x which is less delayed than that used in the first argument of S. This yields to

$$x_l^{m+1} = \begin{cases} S_l\left((x_1^{r_1(m)}, \ldots, x_L^{r_L(m)}), x_l^{t_l(m)}\right) & \text{for } l \in I_m \\ x_l^m & \text{otherwise} \end{cases} , \qquad [3]$$

with $r_l(m) \leq m$ and $t_l(m) \leq m$ for all m and l.

A mathematical model for the totally asynchronous scheme (and a convergence analysis for the linear case) can be found in [5].

3. Convergence analysis

In this section we use variants of El Tarazi's convergence theorem for asynchronous iterations (see [2]) to analyze the convergence of the outer and the flexible asynchronous iterative schemes. These results involve contraction properties with respect to a weighted maximum norm $||\cdot||_w$ in $\mathbb{R}^n$ which, for given positive numbers w_l and norms $||\cdot||_l$ on $\mathbb{R}^{n_l}$ is defined as

$$||x||_w = \max_{l=1}^{L} ||x_l||_l / w_l \quad \text{where } w = (w_1, \ldots, w_L).$$

Theorem 1 *For the outer asynchronous scheme Eq.* [2] *assume that we have*

(i) $r_l(m) \leq m$ *for* $l = 1, \ldots, L$ *and* $m = 0, 1, \ldots,$

(ii) $\lim_{m\to\infty} r_l(m) = \infty$ *for* $l = 1, \ldots, L$,

(iii) each $l \in \{1, \ldots, L\}$ *is an element of infinitely many sets* I_m.

Moreover, suppose that there exists a weighted maximum norm $||\cdot||_w$*, a constant* $\gamma \in [0, 1)$ *and a 'fixed point'* $x^* \in \mathbb{R}^n$ *with* $x^* = S(x^*, x^*)$ *such that for all* $p \in \mathbb{N}$ *we have*

$$||S^p(x, x) - x^*||_w \leq \gamma \cdot ||x - x^*||_w \quad \text{for all } x \in \mathbb{R}^n. \qquad [4]$$

Then $\lim_{m\to\infty} x^m = x^*$ *in the outer asynchronous scheme Eq.* [2].

Proof: This theorem was already formulated in [5], e.g., for a general family of mappings satisfying the contraction property Eq. [4]. It is a minor generalization of the basic theorem by El Tarazi [2] . □

We note that conditions (i) – (iii) in the above theorem represent the standard minimal assumptions in convergence results for asynchronous iterations. They are virtually always satisfied in computational practice, see, e.g., [1]. Condition (ii), like (v) below, is more general than that in [7] since the delays are not required to be bounded here.

Theorem 2 *Consider the asynchronous scheme with flexible communication. In addition to (i) – (iii) from Theorem 1, assume that we have*

(iv) $t_l(m) \leq m$ *for* $l = 1, \ldots, L$ *and* $m = 0, 1, \ldots,$

(v) $\lim_{m\to\infty} t_l(m) = \infty$ *for* $l = 1, \ldots, L$.

Moreover, suppose that there exist a weighted maximum norm $||\cdot||_w$*, a constant* $\gamma \in [0, 1)$ *and a 'fixed point'* $x^* \in \mathbb{R}^n$ *with* $x^* = S(x^*, x^*)$ *such that*

$$||S(x, y) - x^*||_w \leq \gamma \cdot \max\{||x - x^*||_w, ||y - x^*||_w\} \quad \text{for all } x, y \in \mathbb{R}^n. \qquad [5]$$

Then $\lim_{m\to\infty} x^m = x^*$ *in the flexible asynchronous scheme Eq.* [3].

Proof: We consider a new iteration in $\mathbb{R}^n \times \mathbb{R}^n$ by simply duplicating the iterates x^{m+1} from Eq. [3]. To describe this formally, we first observe that the given block decomposition in $\mathbb{R}^n$ induces a block decomposition on $\mathbb{R}^n \times \mathbb{R}^n$ with twice as many blocks which can be referred to by using double indices $(1, l), (2, l),\ l = 1, \ldots, L$. Defining $\tilde{x}^m = (x^m, x^m)$, $\tilde{I}_m = \{1, 2\} \times I_m$, and

$$\begin{aligned}
\tilde{S} \quad &: \quad \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n,\ (x, y) \mapsto (S(x, y), S(x, y)),\\
\tilde{r}(m) \quad &= \quad (\tilde{r}_{1,1}(m), \ldots, \tilde{r}_{1,L}(m), \tilde{r}_{2,1}(m), \ldots, \tilde{r}_{2,L}(m)),\\
&\quad\ \ \text{where } \tilde{r}_{1,l}(m) = r_l(m),\ \tilde{r}_{2,l}(m) = t_l(m),\ l = 1, \ldots, L,
\end{aligned}$$

we obtain

$$\tilde{x}_{\tilde{l}}^{m+1} = \begin{cases} \tilde{S}_{\tilde{l}}\left(\tilde{x}_{1,1}^{\tilde{r}_{1,1}(m)}, \ldots, \tilde{x}_{1,L}^{\tilde{r}_{1,L}(m)}, \tilde{x}_{2,1}^{\tilde{r}_{2,1}(m)}, \ldots, \tilde{x}_{2,L}^{\tilde{r}_{2,L}(m)}\right) & \text{for } \tilde{l} \in \tilde{I}_m \\ \tilde{x}_{\tilde{l}}^{m} & \text{otherwise} \end{cases} . \quad [6]$$

With $\tilde{x}^* = (x^*, x^*)$ and the weighted maximum norm $||\cdot||$ on $\mathbb{R}^n \times \mathbb{R}^n$ defined for $\tilde{x} = (x, y)$ as

$$||\tilde{x}|| = ||(x, y)|| = \max\{||x||_w, ||y||_w\}$$

we get, using Eq. [5],

$$||\tilde{S}(\tilde{x}) - \tilde{x}^*|| = ||S(x, y) - x^*||_w \leq \gamma \cdot ||\tilde{x} - \tilde{x}^*||.$$

Together with the assumptions (i) – (v) we thus see that the iteration Eq. [6] satisfies the hypothesis of the standard convergence theorem of El Tarazi [2]. Consequently, we have $\lim_{m\to\infty} \tilde{x}^m = \tilde{x}^*$ which implies $\lim_{m\to\infty} x^m = x^*$. □

4. The linear case

We now turn back to the linear case where we want to solve the linear system $Ax = b$ and where we are given splittings $A = M - N$ and $M = F - G$ as described in Section 2. Thus, $x^* = A^{-1}b$ and

$$S(x, y) = F^{-1}(Gy + Nx + b). \quad [7]$$

We use the following notation and terminology: A matrix or a vector is called positive (nonnegative) if all its entries are positive (nonnegative). The absolute value $|A|$ of a matrix A is the matrix obtained from A by replacing all entries by their absolute values.

The following theorem states conditions under which Theorems 1 and 2 can be applied to the particular iteration operator S from Eq. [7].

Theorem 3 *Assume that there exists a positive vector $u \in \mathbb{R}^n$ and $\gamma \in [0, 1)$ such that*

$$|F^{-1}N|u + |F^{-1}G|u \leq \gamma \cdot u. \quad [8]$$

Then

$$||S(x, y) - x^*||_u \leq \gamma \cdot \max\{||x - x^*||_u, ||y - x^*||_u\}. \quad [9]$$

Proof: Using $b = Fx^* - Gx^* - Nx^*$ we get

$$S(x, y) - x^* \;=\; F^{-1}(Gy + Nx - b) - x^* \;=\; F^{-1}G(y - x^*) + F^{-1}N(x - x^*).$$

We now make use of the fact that for any $B \in \mathbb{R}^{n\times n}$ and $z \in \mathbb{R}^n$ we have $|(Bz)_i| \leq (|B|u)_i \cdot ||z||_u$ to obtain, using Eq. [8] for component i,

$$\begin{aligned} \frac{1}{u_i}|S_i(x, y) - x_i^*| &\leq \frac{1}{u_i}\left((|F^{-1}G|u)_i \cdot ||y - x^*||_u + (|F^{-1}N|u)_i \cdot ||x - x^*||_u\right) \\ &\leq \frac{1}{u_i}\left((|F^{-1}G|u)_i + (|F^{-1}N|u)_i\right) \cdot \max\{||y - x^*||_u, ||x - x^*||_u\} \\ &\leq \gamma \cdot \max\{||y - x^*||_u, ||x - x^*||_u\}. \end{aligned}$$

Taking the maximum over i yields Eq. [9]. $\square$

Let us first notice that Eq. [9] implies Eq. [5] for *any* block partitioning of $\{1, \ldots, n\}$ into L disjoint blocks B_l, $l = 1, \ldots, L$. Indeed, if on the subspace corresponding to block B_l we set

$$\|x_l\|_l = \max_{i \in B_l} |x_i|/u_i$$

and if we take $w = (1, \ldots, 1) \in \mathbb{R}^L$, we get

$$\|x\|_w = \max_{l=1}^{L} \|x_l\|_l = \max_{i=1}^{n} |x_i|/u_i = \|x\|_u.$$

For this particular norm $\|\cdot\|_w$ Eq. [9] thus implies

$$\|S(x, y) - x^*\|_w \leq \gamma \cdot \max\{\|x - x^*\|_w, \|y - x^*\|_w\},$$

i.e. Eq. [5].

As a second remark we note that Eq. [5] actually implies Eq. [4], as can be seen by a trivial induction. Theorem 3 therefore gives conditions for convergence of outer asynchronous as well as flexible asynchronous linear two-stage iterations.

Our final theorem shows that the crucial assumption Eq. [8] of Theorem 3 is met for certain 'standard' splittings. Recall that a nonsingular matrix $B \in \mathbb{R}^{n \times n}$ is termed an M-*matrix* if all off-diagonal entries are nonpositive and $B^{-1} \geq 0$, and that it is termed an H-*matrix* if its comparison matrix $\langle B \rangle$ with

$$\langle B \rangle_{ij} = \begin{cases} |B_{ij}| & \text{if } i = j \\ -|B_{ij}| & \text{if } i \neq j \end{cases}$$

is an M-matrix.

Theorem 4 *Condition Eq.* [8] *is satisfied in either of the following two cases:*

(i) $A^{-1} \geq 0,\ F^{-1} \geq 0,\ F^{-1}N \geq 0$ *and* $F^{-1}G \geq 0$,

(ii) $\langle M \rangle - |N|$ *is an* M*-matrix and* $\langle M \rangle = \langle F \rangle - |G|$.

Proof: To show (i) note that $F^{-1}N + F^{-1}G = I - F^{-1}A$. Let $e = (1, \ldots, 1)$ and $u = A^{-1}(1, \ldots, 1)$ which is a positive vector. Since $|F^{-1}N| = F^{-1}N$ and $|F^{-1}G| = F^{-1}G$ we get

$$|F^{-1}N|u + |F^{-1}G|u = (I - F^{-1}A)u = u - F^{-1}e. \qquad [10]$$

But $F^{-1}e > 0$ so that we can find $\gamma \in [0, 1)$ with $u - F^{-1}e \leq \gamma u$. Thus, Eq. [10] implies Eq. [8].

To show (ii) we first note that $\langle M \rangle$ as well as $\langle F \rangle$ are H-matrices (see [4]) and that for any H-matrix B one has $|B^{-1}| \leq \langle B \rangle^{-1}$ (see [4, 8]). Therefore, $|F^{-1}N| \leq \langle F \rangle^{-1}|N|$ and $|F^{-1}G| \leq \langle F \rangle^{-1}|G|$, so that for any positive vector u we have

$$\begin{aligned} |F^{-1}N|u + |F^{-1}G|u &\leq \langle F \rangle^{-1}(|N| + |G|)u \\ &= \langle F \rangle^{-1}(|N| + \langle F \rangle - \langle M \rangle)u \\ &= (I - \langle F \rangle^{-1}(\langle M \rangle - |N|))u. \end{aligned}$$

Since $\langle M \rangle - |N|$ is an M-matrix, we can take $(\langle M \rangle - |N|)^{-1}e$ as our positive vector u. As for (i) we then obtain Eq. [8]. $\square$

We remark that (i) is fulfilled if $A^{-1} \geq 0$, the splitting $A = M - N$ is *regular* ($M^{-1} \geq 0, N \geq 0$) and the splitting $M = F - G$ is *weak regular* ($F^{-1} \geq 0, F^{-1}G \geq 0$). A discussion of (ii) can be found in [4].

References

[1] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation*. Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[2] Mouhamed Nabih El Tarazi. Some convergence results for asynchronous algorithms. *Numerische Mathematik*, 39:325–340, 1982.

[3] Andreas Frommer, Hartmut Schwandt, and Daniel B. Szyld. Asynchronous weighted additive Schwarz methods. *Electronic Transactions on Numerical Analysis*, 5:48–61, 1997.

[4] Andreas Frommer and Daniel B. Szyld. H-splittings and two-stage iterative methods. *Numerische Mathematik*, 63:345–356, 1992.

[5] Andreas Frommer and Daniel B. Szyld. Asynchronous two-stage iterative methods. *Numerische Mathematik*, 69:141–153, 1994.

[6] Paul J. Lanzkron, Donald J. Rose, and Daniel B. Szyld. Convergence of nested classical iterative methods for linear systems. *Numerische Mathematik*, 58:685–702, 1991.

[7] Jean Claude Miellou, Didier El Baz, and Pierre Spitéri. A new class of asynchronous iterative algorithms with order intervals. *Mathematics of Computation*. To appear.

[8] Arnold Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, 1990.

[9] Daniel B. Szyld. Different models of parallel asynchronous iterations with overlapping blocks. *Applied and Computational Mathematics*, 18, 1998. To appear.