ASYNCHRONOUS RICHARDSON ITERATIONS: THEORY AND PRACTICE*

EDMOND CHOW†, ANDREAS FROMMER‡, AND DANIEL B. SZYLD§

Abstract. We consider asynchronous versions of the first and second order Richardson methods for solving linear systems of equations. These methods depend on parameters which are chosen *a priori*. We explore the parameter values that can be proven to give convergence of the asynchronous methods. This is the first such analysis for asynchronous second order methods. We find that for the first order method, the optimal parameter value for the synchronous case also gives an asynchronously convergent method. For second order method, the parameter ranges for which we can prove asynchronous convergence do not contain the optimal parameters for the synchronous iteration. In practice, however, the asynchronous second order iterations may still converge using the optimal parameter values, or close to the optimal parameter values, despite this result. We explore this behavior with a multithreaded parallel implementation of the asynchronous methods.

Key words. Asynchronous iterations. Parallel Computing. Second order Richardson method.

AMS subject classifications. 65F10, 65N22, 15A06

1. Introduction.

A parallel asynchronous iterative method for solving a system of equations is a fixed-point iteration in which processors do not synchronize at the end of each iteration. Instead, processors proceed iterating with the latest data that is available from other processors. Running an iterative method in such an asynchronous fashion may reduce solution time when there is an imbalance of the effective load between the processors because fast processors do not need to wait for slow processors. Solution time may also be reduced when interprocessor communication costs are high because computation continues while communication takes place. However, the convergence properties of a synchronous iterative method are changed when running the method asynchronously.

Consider the n-by-n system of equations $x = G(x)$ which can be written in scalar form as $x_i = g_i(x)$, $i = 1, \ldots, n$. An asynchronous iterative method for solving this system of equations can be defined mathematically as the sequence of updates [2, 3, 5],

$$x_i^k = \begin{cases} x_i^{k-1}, & \text{if } i \notin J_k \\ g_i(x_1^{s_1^i(k)}, x_2^{s_2^i(k)}, \ldots, x_n^{s_n^i(k)}), & \text{if } i \in J_k \end{cases}$$

where x_i^k denotes x_i at time instant k, J_k is the set of indices updated at instant k, and $s_j^i(k) \le k-1$ is the instant that x_j is read when computing g_i at instant k. We point out that (a) not all updates are performed at the same time instant, and (b) updates may use stale information, which models communication delays in reading or writing.

With some natural assumptions on the sequence of updates above, much work has been done on showing the conditions under which asynchronous iterative methods converge; see the survey [9]. For linear systems, asynchronous iterations converge for any initial vector if and only if $\rho(|T|) < 1$, where T is the iteration matrix for the standard, synchronous iterations, and $|\cdot|$ is taken elementwise. Since $\rho(T) \le \rho(|T|)$,

*This version dated July 3, 2020.

†Georgia Institute of Technology, Atlanta, GA, USA (echow@cc.gatech.edu).

‡Bergische Universität Wuppertal, Wuppertal, Germany (frommer@math.uni-wuppertal.de).

§Temple University, Philadelphia, PA, USA (szyld@temple.edu).

it appears that the condition for convergence of asynchronous iterations is more strict than that of synchronous iterations.

For linear systems, asynchronous iterative methods that are based on the Jacobi or block Jacobi splitting have been extensively studied (for some recent references, see [4, 14, 19, 20]), although these splittings generally give slow convergence. In this paper, we consider first and second order Richardson methods [16]. With estimates on the bounds of the spectrum of a problem, the second order Richardson method, in particular, converges rapidly. This paper explores the parameter values that can be proven to give convergence of asynchronous Richardson methods. This is the first such analysis for asynchronous second order methods.

Statements about the rate of convergence, however, cannot be made without a description of the sets J_k and $s_j^i(k)$. Such sets depend on properties of the parallel computation, including how the problem is partitioned among the processors, and computer characteristics such as computation speed and interprocessor communication latency and bandwidth. Indeed, one can imagine that in an asynchronous computation where communication is fast and the workload is balanced, the asynchronous computation may behave very much like the synchronous computation. In this paper, we also demonstrate the actual behavior of asynchronous first and second order Richardson methods using a parallel multithreaded implementation of the methods.

Our theoretical and experimental results are suggestive for an asynchronous version of the Chebyshev semi-iterative method. The Chebyshev method can be regarded as the non-stationary counterpart of the stationary method which is the second order Richardson method. If one uses the optimal parameter values in second order Richardson, i.e., the parameter values that minimize the spectral radius of the iteration operator, then, asymptotically, both second order Richardson and Chebychev iterations have the same convergence rate [13]. For a short historical description of the development of these methods, see [17]. Unlike Krylov subspace methods, the second order Richardson and Chebyshev methods do not require inner products, which allows the possibility of executing these methods asynchronously.

In recent related work, asynchronous versions of Schwarz and optimized Schwarz methods have been developed [10, 15, 21].

2. The setting.

We consider

$$\hat{A}x = \hat{b}, \quad \hat{A} \in \mathbb{C}^{n\times n}, \hat{b} \in \mathbb{C}^n.$$

From the beginning, we assume that this system is preconditioned with a nonsingular matrix M, that is, we have $\hat{A} = M - N$, $T = M^{-1}N$, $c = M^{-1}\hat{b}$, and the original linear system is equivalent to

$$Ax = c, \text{ where } A = M^{-1}\hat{A} = I - T, \quad c = M^{-1}\hat{b}. \tag{1}$$

We assume that A and M are such that $T \geq 0$ and is convergent, i.e., that

$$\rho = \rho(T) < 1,$$

and that the spectrum spec(A) is in $\mathbb{R}^+$. That is, we are assuming that $\hat{A} = M - N$ is a convergent weak splitting with the additional property that the spectrum of T is real. This includes of course the Jacobi and block Jacobi methods. In other words, with this splitting, a standard iterative method would be as follows. Given x^0, for $k = 0, 1, \ldots$, compute

$$x^{k+1} = Tx^k + c. \tag{2}$$

We note then that if we denote $\lambda_{\min}$ and $\lambda_{\max}$ the smallest and largest eigenvalue of A we have

$$\lambda_{\min} = 1 - \rho,\ \ \lambda_{\max} \leq 1 + \rho.$$

We also assume that T is irreducible, so that we have a positive Perron vector $w > 0$ with $Tw = \rho w$. (If T is reducible, we can consider small irreducible perturbations $T + \epsilon ee^*$ with $e = [1, \ldots, 1]^*$ of T and then go to the limit in the usual way, but we do not elaborate on this here.)

3. First order Richardson. The first order Richardson method consists of taking a linear combination of the previous iteration with that which would come from the standard iteration (2). This method can be seen as the simplest case of semi-iterative methods [6, 7, 18], and thus the sum of the coefficients of the linear combination must add up to one, since otherwise the method cannot produce iterates that converge towards $A^{-1}b$.[1]

We first consider the case where the parameter α defining the Richardson iteration is fixed for all iterations. This is a stationary iteration. We consider later the case where $\alpha = \alpha_k$, a nonstationary iteration.

This is the (synchronous) iteration

$$x^{k+1} = (1-\alpha)x^k + \alpha(Tx^k + c) = x^k + \alpha[c - (I-T)x^k] = x^k + \alpha r^k, \tag{3}$$

where $r^k = c - (I-T)x^k$ is the residual of the equivalent system (1)

The convergence analysis of this synchronous method consists of analyzing the spectral radius of the iteration matrix $T_\alpha = (1-\alpha)I + \alpha T = I - \alpha(I-T) = I - \alpha A$. Let $\mu \in \operatorname{spec}(T_\alpha)$, then, $\mu = 1 - \alpha + \alpha\lambda$, with $\lambda \in \operatorname{spec}(T)$, i.e., $\lambda \in [-\rho, \rho]$.

The convergence analysis of the synchronous method is straight-forward and well-known.

THEOREM 1. *We have that*

(i) iteration (3) converges if $\alpha \in (0, \frac{2}{\lambda_{\max}})$,

(ii) the optimal choice is $\alpha = 2/(\lambda_{\min} + \lambda_{\max})$ *in the sense that this choice minimizes* $\rho(T_\alpha)$,

(iii) the optimal choice w.r.t. the information $\operatorname{spec}(A) \subset [a, b]$, $a > 0$ *is* $\alpha = 2/(a+b)$.

Proof. We have $\operatorname{spec}(T_\alpha) = \{1 - \alpha\lambda : \lambda \in \operatorname{spec}(A)\}$ and thus

$$\rho(T_\alpha) = \max\{|1 - \alpha\lambda_{\min}|, |1 - \alpha\lambda_{\max}|\}.$$

From this we see that $\rho(T_\alpha) < 1$ iff $\alpha \in (0, \frac{2}{\lambda_{\max}})$, which is (i), and that $\rho(T_\alpha)$ is minimal if $1 - \alpha\lambda_{\min} = -(1 - \alpha\lambda_{\max})$ which gives (ii). Part (iii) follows from equating $1 - \alpha a$ with $-(1 - \alpha b)$. ∎

Note that in our situation we know $\operatorname{spec}(A) \subset [1-\rho, 1+\rho]$, and the optimal α w.r.t. this information is $\alpha = 1$.

For the asynchronous iteration we have to analyze when $\rho(|T_\alpha|) < 1$ [9], and we do so by showing that $|T_\alpha|w \leq \nu w$ for some $\nu \in [0, 1)$, $w > 0$ the Perron vector of T. That is, we show that the weighted-max norm $\|T_\alpha\|_w < 1$. The underlying vector norm $\|\cdot\|_w$ is defined for any positive vector w as $\|v\|_w = \max_i \dfrac{|v_i|}{w_i}$.

[1] Gene Golub in his thesis [12] calls this a method of averaging, following the nomenclature used by von Neumann.

THEOREM 2. *We have* $\rho(|T_\alpha|) < 1$ *if* $\alpha \in (0, \frac{2}{1+\rho})$, *where* $\frac{2}{1+\rho} > 1$.

Proof. We have

$$|T_\alpha|w \ \leq \ |1-\alpha|w + \alpha T w = (|1-\alpha| + \alpha\rho)w = \nu w \text{ with } \nu = |1-\alpha| + \alpha\rho.$$

For $0 < \alpha \leq 1$ we have $0 \leq \nu = (1-\alpha) + \rho\alpha = 1 - \alpha(1-\rho) < 1$, and for $1 < \alpha < \frac{2}{1+\rho}$ we have $0 < \nu = (\alpha - 1) + \rho\alpha = (1+\rho)\alpha - 1 < 1$. ∎

We note that $\alpha = 1$, the optimal parameter one obtains assuming that $\operatorname{spec}(A) \subseteq [1-\rho, 1+\rho]$ is covered by this theorem.

We discuss now the case in which $\alpha = \alpha_k$, i.e., the case, where the first order Richardson parameter changes from one iteration to the next. As long as $0 < \alpha_k < \frac{2}{1+\rho}$, the "non-stationary" asynchronous method converges as well, using [9, Corollary 3.2]. In fact, using the latter result, we have the following theorem.

THEOREM 3. *Let* $T_k : \mathbb{C}^n \to \mathbb{C}^n$, $k \in \mathbb{N}$ *be a pool of linear operators sharing the same fixed point* $x^* = A^{-1}b$ *and being all contractive w.r.t. this fixed point in the same weighted max-norm, i.e.,* $\|T_k - x^*\|_w \leq \gamma_k \|x - x^*\|$ *for all* $x \in \mathbb{C}^n$. *If* $0 \leq \gamma_k \leq \gamma < 1$ *for some* $\gamma \in [0, 1)$, *then the asynchronous iterations which at each step picks one of the operators form the pool as its iteration operator, produces iterates which converge to* x^*.

The result for non-stationary first order Richardson follows by taking as w the Perron vector of T and by observing with $T_k = (1-\alpha_k)I + \alpha_k T$ we have that $\|(1-\alpha_k)I + \alpha_k T\| \leq |1-\alpha_k| + \alpha_k\rho < |1-\alpha| + \alpha\rho < 1$.

4. Second order Richardson. The second order Richardson is the semi-iterative method one obtains with the linear combination of the standard iteration (2) with the two previous iterations. Again, all coefficients have to add up to one. Equivalently, one can take a linear combination of the first order Richardson iteration (3) with the previous step, as follows

$$\begin{aligned}
x^{k+1} &= (1+\beta)[(1-\alpha)x^k + \alpha(Tx^k + c)] - \beta x^{k-1} \\
&= -\beta x^{k-1} + (1+\beta)x^k + (1+\beta)\alpha[-x^k + Tx^k + c)] \\
&= x^k - \beta(x^{k-1} - x_k) + (1+\beta)\alpha[c - (I-T)x^k)] \\
&= x^k + \beta(x^k - x^{k-1}) + (1+\beta)\alpha(c - Ax^k) \qquad (4) \\
&= (1+\beta)(I-\alpha A)x^k - \beta x^{k-1} + (1+\beta)\alpha c, \ k = 1, 2, \ldots.
\end{aligned}$$

One needs to prescribe x^1 as well as x^0, and one can use one step of (2) or one step of first order Richardson [12].

The results to come are less nice than those for first order Richardson, since we can show the convergence of asynchronous second order Richardson only for parameter values which are quite far from the optimal ones.

We can write the three-term recurrence in (4) using a matrix of doubled size as follows, cf. [22],

$$\begin{bmatrix} x^{k+1} \\ x^k \end{bmatrix} = \underbrace{\begin{bmatrix} (1+\beta)(I-\alpha A) & -\beta I \\ I & 0 \end{bmatrix}}_{:=T_{\alpha,\beta}} \begin{bmatrix} x^k \\ x^{k-1} \end{bmatrix} + \begin{bmatrix} (1+\beta)\alpha c \\ 0 \end{bmatrix}.$$

We find in the literature for the synchronous implementation of (4) two approaches to analyze its convergence. Following [22], we note that the if λ is an eigenvalue of

$T_{\alpha,\beta}$ with eigenvector $(s^T, t^T)^T$, then, $s = \lambda t$, and $(1+\beta)[(I-\alpha A)]s - \beta t = \lambda s$, that is, $(1+\beta)(I-\alpha A)\lambda t - \beta t = \lambda^2 t$. Thus, assuming that $t \neq 0$, this implies that $\det[(1+\beta)(I-\alpha A)\lambda - \beta I - \lambda^2 I] = 0$, so that for $\mu \in \operatorname{spec}(A)$, the eigenvalues of $T_{\alpha,\beta}$ must satisfy the quadratic equation

$$\lambda^2 - (1+\beta)(1-\alpha\mu)\lambda + \beta = 0. \tag{5}$$

Figure 1 (first column) plots the spectral radius of $T_{\alpha,\beta}$ for three examples.

Frankel [8] shows that the parameters α and β minimizing the maximum of these polynomials is given by $\alpha = 2/(a+b)$, and $\beta = \left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right)^2 := q^2$, for A assumed to have $\operatorname{spec}(A) \subset [a,b]$ with $a > 0$. In other words, these parameters are optimal in the sense that they minimize $\rho(T_{\alpha,\beta})$, the spectral radius of the iteration operator.

On the other hand, if one uses these optimal parameters, Golub [12] (see also [13]) used the recurrence of the polynomials defining (4) to bound the 2-norm of the error as follows

$$\|x^k - x^*\|_2 \le \left[q^k \left(1 + k\frac{1-q^2}{1+q^2}\right)\right] \|x^0 - x^*\|_2, \tag{6}$$

where x^* is the solution of (1).

In summary, the following is known for the synchronous iteration.

THEOREM 4. *We have*

(i) The optimal parameters w.r.t. the information $\operatorname{spec}(A) \subset [a,b]$ *with* $a > 0$ *are* $\alpha = 2/(a+b)$ *and* $\beta = \left(\frac{b-a}{a+b+2\sqrt{ab}}\right)^2 = \left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right)^2$.

(ii) With these parameters, the asymptotic convergence factor $\rho(T_{\alpha,\beta})$ *is given in (6).*

For the asynchronous second order Richardson, the following theorem proves convergence for certain ranges for α and β.

THEOREM 5. *We have* $\rho(|T_{\alpha,\beta}|) < 1$, *provided*

$$\alpha > 0 \text{ and } |1+\beta|(|1-\alpha| + \alpha\rho) + |\beta| < 1. \tag{7}$$

Before we prove the theorem, consider the case $\alpha = 1$. Then the theorem states that asynchronous iterations converge for $-1 \le \beta < \frac{1-\rho}{1+\rho}$, as can be seen from considering the two cases $\beta \ge 0$ and $-1 < \beta < 0$ separately. If the information about the spectral interval is $\operatorname{spec}(A) \subset [1-\rho, 1+\rho]$, the optimal α from Theorem 4 is precisely $\alpha = 1$, and the corresponding optimal β will be close to 1 for ρ close to 1, whereas $1-\rho$, the bound for β from (7) for $\alpha = 1$, will be close to 0.

Proof of Theorem 5. Let $\gamma > 1$ and consider the vector $\left[\begin{smallmatrix} w \\ \gamma w \end{smallmatrix}\right]$. Then, if $\alpha > 0$, we have

$$\begin{aligned} |T_{\alpha,\beta}| \begin{bmatrix} w \\ \gamma w \end{bmatrix} &= \begin{bmatrix} |1+\beta| \cdot |I - \alpha A| & |\beta| I \\ I & 0 \end{bmatrix} \begin{bmatrix} w \\ \gamma w \end{bmatrix} \\ &= \begin{bmatrix} (|1+\beta| \cdot (|1-\alpha| + \alpha\rho) + |\beta|\gamma) w \\ w \end{bmatrix} < \sigma \begin{bmatrix} w \\ \gamma w \end{bmatrix}, \end{aligned}$$

with

$$\sigma = \max\{\tfrac{1}{\gamma}, |1+\beta| \cdot (|1-\alpha| + \alpha\rho) + |\beta|\gamma\}. \tag{8}$$

Now, if $|1+\beta|(|1-\alpha| + \alpha\rho) + |\beta| < 1$, choose $\gamma > 1$ close enough to 1 such that we have $|1+\beta|(|1-\alpha| + \alpha\rho) + \gamma|\beta| < 1$, which gives $\sigma < 1$ in (8). ∎

We note that for $\beta < -1$, the inequality $|1+\beta|(|1-\alpha|+\alpha\rho)+|\beta| < 1$ cannot be fulfilled. Denoting $\nu := |1-\alpha|+\alpha\rho$ we can distinguish the two cases $0 \leq \nu < 1$ and $\nu \geq 1$. In the first case, we obtain that $|1+\beta|\nu + |\beta| < 1$ if $-1 \leq \beta < \frac{1-\nu}{1+\nu}$. In the second case, there is no β which satisfies the inequality.

We want to study the eigenvalues of $|T_{\alpha,\beta}|$. We follow the same development as before for $T_{\alpha,\beta}$ and write:

$$|T_{\alpha,\beta}| \begin{bmatrix} s \\ t \end{bmatrix} = \lambda \begin{bmatrix} s \\ t \end{bmatrix} .$$

Looking at the second block row of $|T_{\alpha,\beta}|$, we conclude that for the eigenvalue λ it must hold that $s = \lambda t$.

Now, the first block row reads:

$$(|1+\beta||I-\alpha A|\lambda + |\beta| I - \lambda^2 I)t = 0.$$

This means that

$$\det(|1+\beta||I-\alpha A|\lambda + |\beta| I - \lambda^2 I) = 0.$$

For every eigenvalue $\mu = \mu_i$ of $|I-\alpha A|$ we then have that λ satisfies the quadratic equation

$$\lambda^2 - |1+\beta|\mu\lambda - |\beta| = 0. \tag{9}$$

Figure 1 (second column) plots the spectral radius of $|T_{\alpha,\beta}|$ for three examples.

5. An additional result. The following result shows how to find a starting vector for an asynchronous iteration that diverges. The setting here is $T \geq 0$ and $\rho(T) > 1$.

THEOREM 6. *Assume that $T \geq 0$ and that $\rho(T) > 1$. Then for any asynchronous iteration (i.e., choice of $s_j^i(k)$ and J_k defined in Section 1) there exists a starting error e^0 such that the iteration does not reduce the error to 0.*

Proof. Let $w > 0$ be a vector for which $Tw \geq \sigma w$ with $\sigma > 1$. Such w exists, take it as the Perron vector of $T + \epsilon E$, E the matrix of all ones, for $\epsilon > 0$ sufficiently small. Assume that the initial error satisfies $e^0 \geq w$, and that, inductively, the all errors e^ℓ to the $k-1$st satisfy $e^\ell \geq w$. Then, for those components $i \in I_k$ that we update in time instant k we have

$$e_i^k = T_i(e_1^{s_1^i(k)}, \ldots, e_n^{s_n^i(k)})^T \geq T_i w \geq w_i,$$

where the $s_j^i(k) \leq k-1$. Consequently, $e^k \geq w$. ∎

6. Discussion. For the second order Richardson method, Figure 1 plots the contours of the spectral radius of $T_{\alpha,\beta}$ (synchronous case) and of $|T_{\alpha,\beta}|$ (asynchronous case) as a function of α and β when $\lambda_{\min}(A) = 1-\rho$ and $\lambda_{\max}(A) = 1+\rho$, for ρ equal to 0.1, 0.5, and 0.9. The spectral radii were computed from the roots of the polynomials (5) and (9). In our setting, the optimal α is always 1.

In the synchronous case, as ρ increases, the optimal value of β increases from near 0 toward 1.

The plots for the asynchronous case are best explained in terms of the plots for the synchronous case. When $\beta \leq 0$, $\rho(|T_{\alpha,\beta}|)$ and $\rho(T_{\alpha,\beta})$ appear to be the same. When $\beta > 0$, it appears that $\rho(|T_{\alpha,\beta}|) > \rho(T_{\alpha,\beta})$. In particular, the region where the spectral radius is less than 1 is smaller in the asynchronous case than in the synchronous case.

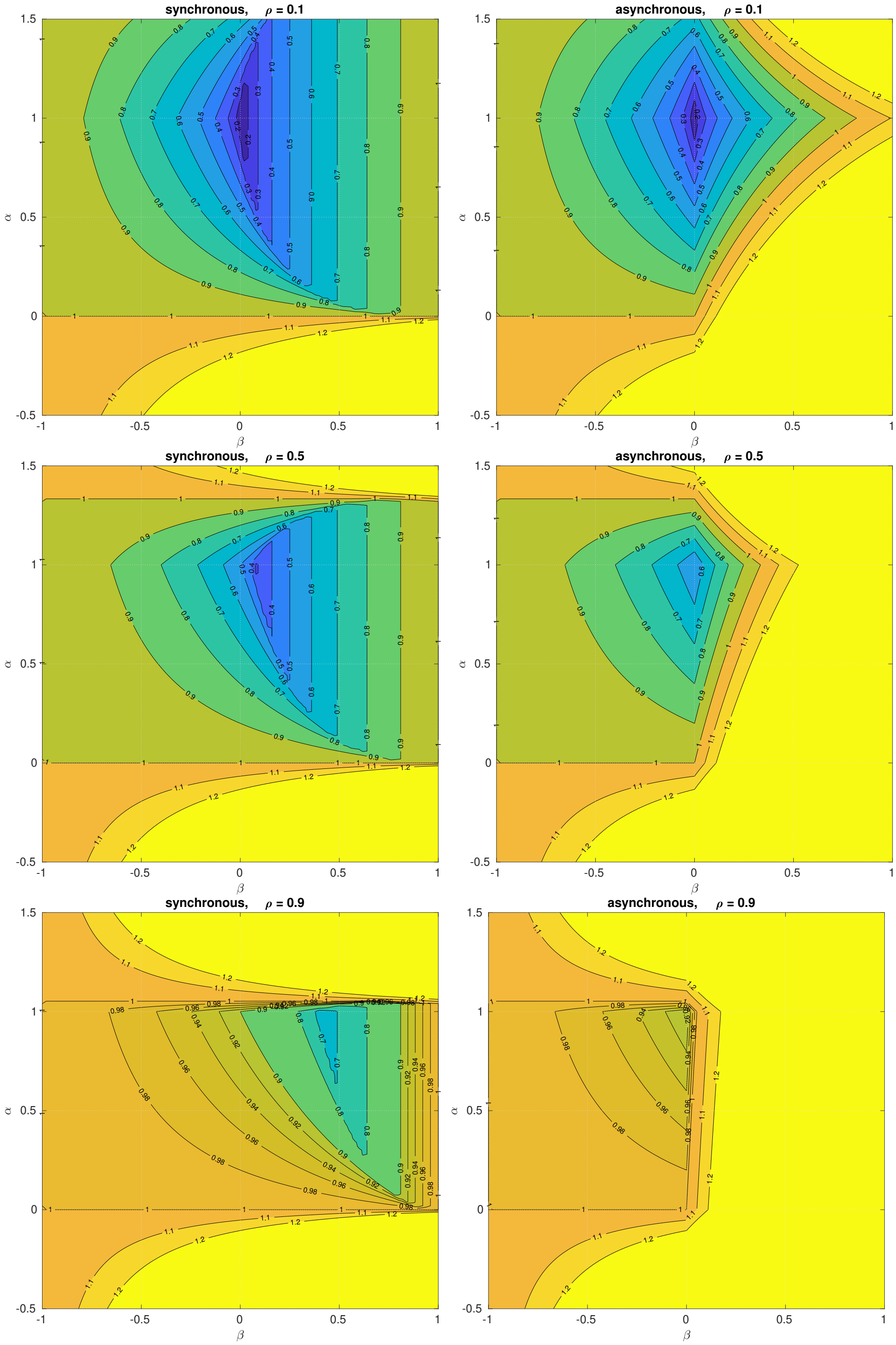

FIG. 1. *Spectral radius of $T_{\alpha,\beta}$ (synchronous case) and of $|T_{\alpha,\beta}|$ (asynchronous case) as a function of α and β when $\lambda_{\min}(A) = 1 - \rho$ and $\lambda_{\max}(A) = 1 + \rho$, for three values of ρ.*

The effect is that the asynchronous method has an optimal value for β of 0, which corresponds to the first order method. Here, optimal means minimizing $\rho(|T_{\alpha,\beta}|)$, although $\rho(|T_{\alpha,\beta}|)$ is only correctly used to ascertain asymptotic convergence and does not directly correspond to any convergence rate.

Consider $\rho = 0.5$. For the synchronous case, the optimal β is approximately 0.0718. Although the asynchronous method can converge for this value of β, the value of 0 gives a lower value of $\rho(|T_{\alpha,\beta}|)$. Now consider $\rho = 0.9$. For the synchronous case, the optimal β is approximately 0.3929. The asynchronous method has spectral radius greater than 1 for this value of β. To guarantee convergence, the asynchronous method must use a very small value of β.

These results are quite negative for the asynchronous second order method. However, in practice, the situation could be more favorable. The condition $\rho(|T_{\alpha,\beta}|) < 1$ for the asynchronous method guarantees that the method will converge for any initial vector and any sequence of asynchronous iterations, i.e., with any choice of specific delays, $k - s_j^i(k)$, and any choice of when components are updated (satisfying natural conditions). In practice, the asynchronous method may converge despite $\rho(|T_{\alpha,\beta}|) > 1$. One could imagine that the "degree of asynchrony" affects the convergence of the asynchronous method, and we explore this next with numerical experiments.

7. Numerical behavior.

The asynchronous first and second order Richardson methods were implemented in parallel using multithreading and shared memory. Tests were run on a dual processor Intel Xeon computer with a total of 20 cores. The threads were pinned to the cores using "scatter" thread affinity.

The test matrix is the standard finite difference Laplacian matrix on a 100×100 grid of unknowns, scaled so that its diagonal is all ones. This matrix satisfies the setting of this paper so that $\rho(T) < 1$, $T \geq 0$, and T is irreducible. A single right-hand side was chosen randomly and uniformly from $(-0.5, 0.5)$ and was the same for all tests. The initial vector was zero.

Different numbers of threads were used. Each thread was assigned approximately the same number unknowns to update. The iterations performed by each thread were terminated when the all the unknowns were updated an average of 500 times. Because the threads operate asynchronously, the number of updates performed on each unknown is generally different. We refer to the difference between the largest number of updates and the smallest number of updates as the *range*. When the iterations are terminated, we measure the residual norm relative to the initial residual norm. The residual norm is not calculated during the iterations, as such calculations involving dot products induce synchronization in the method.

7.1. First order Richardson.

For the asynchronous first order Richardson method, Table 1 shows the convergence results for different numbers of threads. For the given matrix, the optimal α is 1. For each number of threads, the method was run 100 times. Columns 2 and 3 of the table show the average range, and the average relative residual norm when the asynchronous iterations were terminated. For comparison, the relative residual norm attained after 500 iterations of the synchronous first order Richardson method is 1.691939e-02. Evidently, the convergence of the asynchronous method is *better* than the convergence of the synchronous method. This perhaps nonintuitive result is due to the fact that the asynchronous method has a multiplicative effect [19, 20], i.e., unknowns are not all updated at the same time, and when unknowns are updated, they are immediately available to other threads. Indeed, for a single thread, the asynchronous method corresponds to Gauss-Seidel, giving a relative residual norm of 7.421009e-03 which is lower than that of the synchronous

TABLE 1
Asynchronous first order Richardson for different numbers of threads. For comparison, the synchronous method attains an average relative residual norm of 1.691939e-02 for all numbers of threads. Timings for the asynchronous and synchronous methods are also given.

number of threads	average range	average rel. resid. norm	async time (s)	sync time (s)
1	0.0	7.421009e-03	0.060177	0.048345
2	17.1	7.491060e-03	0.034049	0.030291
3	76.1	7.686441e-03	0.022664	0.020642
4	98.3	7.624358e-03	0.018009	0.017360
5	129.6	7.940683e-03	0.015023	0.015171
6	138.1	7.902309e-03	0.012898	0.012751
7	144.6	8.021550e-03	0.011334	0.012374
8	172.2	8.149458e-03	0.010997	0.012067
9	240.4	8.500669e-03	0.010039	0.010737
10	191.4	8.248697e-03	0.009339	0.010642
11	222.4	8.363452e-03	0.009225	0.010741
12	215.5	8.311822e-03	0.008861	0.010590
13	248.9	8.450671e-03	0.009132	0.010339
14	227.7	8.416794e-03	0.007867	0.009669
15	253.7	8.403988e-03	0.009014	0.009998
16	292.2	8.610365e-03	0.008414	0.009871
17	284.6	8.530868e-03	0.008179	0.009668
18	305.9	8.573682e-03	0.007307	0.009660
19	288.4	8.445288e-03	0.007020	0.009496
20	297.3	8.448706e-03	0.007200	0.009249

method, which corresponds to the Jacobi method. As the number of threads is increased, convergence generally worsens slightly as the method departs from a pure Gauss-Seidel method. The convergence is always better than the convergence of the synchronous method for all numbers of threads tested.

The table also shows timings for the asynchronous method and the synchronous method different numbers of threads. For small numbers of threads, the synchronous method is faster in performing 500 iterations than the asynchronous method in performing an average of 500 iterations by each thread. This can be explained by two factors: (1) the asynchronous method has more work to do because each thread, after each iteration, needs to count how many iterations have been performed by other threads in order to decide whether to terminate, and (2) the asynchronous method has more write invalidations of cache lines compared to the synchronous method which writes new values of x to a separate array. However, for large numbers of threads, despite these two factors, the asynchronous method is faster, due to the elimination of thread synchronization. The overhead of threads waiting for other threads in the synchronous method is evidently larger when more threads are used.

7.2. Second order Richardson. For the asynchronous second order Richardson method, Table 2 shows the convergence results for different numbers of threads using the values $\alpha = 1$ and $\beta \approx 0.93968$ which are optimal for the synchronous method. For these values, the asynchronous method is not guaranteed to converge. For each number of threads, the method was run 100 times. The table shows the average range, the average relative residual norm, and the number of failures, which is the number of times the relative residual norm is greater than unity in the 100 runs.

When a single thread is used, the asynchronous method is mathematically identical to the synchronous method. When a small number of threads was used, the asynchronous method always converged in the 100 runs, with a degradation in the

Table 2

Asynchronous second order Richardson for different numbers of threads. The parameter values $\alpha = 1$ and $\beta \approx 0.93968$ that were used are optimal for synchronous *iterations. For comparison, the synchronous method attains an average relative residual norm of 1.258388e-07 for all numbers of threads. Timings for the asynchronous and synchronous methods are also given.*

number of threads	average range	average rel. resid. norm	number of failures	async time (s)	sync time (s)
1	0.0	1.258388e-07	0	0.053275	0.052961
2	40.8	4.235170e-07	0	0.031146	0.032542
3	104.3	6.175605e-06	0	0.019592	0.023368
4	115.7	1.444428e-05	0	0.016493	0.018801
5	166.0	1.495107e-04	0	0.013533	0.017519
6	163.0	4.524130e-04	0	0.011563	0.014606
7	200.1	1.868556e-03	0	0.010649	0.013078
8	151.5	9.259216e-03	0	0.009794	0.012843
9	246.0	4.035731e-02	1	0.008917	0.012560
10	203.2	1.088207e-01	1	0.009000	0.012371
11	209.4	4.582844e-01	21	0.008972	0.011905
12	185.5	1.678645e+00	25	0.008397	0.011527
13	227.6	1.046313e+01	32	0.008216	0.011698
14	205.9	3.971405e+01	43	0.007081	0.010863
15	239.3	5.207066e+02	35	0.007568	0.010828
16	166.8	2.317140e+02	24	0.007101	0.011470
17	226.3	3.303636e+01	22	0.006217	0.011161
18	191.8	6.415417e+01	30	0.005972	0.010969
19	237.6	2.377968e+01	23	0.006237	0.011147
20	173.8	3.136173e+01	46	0.006614	0.011012

"convergence rate" as the number of threads is increased. What we mean here with convergence rate is how small is the residual when the termination criterion is satisfied. When a larger number of threads was used, the number of failures of the asynchronous method generally increases. This is due to an increased degree of asynchrony, which is somewhat reflected by the increasing average range.

The table also shows timings for the asynchronous and synchronous second order Richardson methods. The asynchronous method is faster when more than 1 thread is used, and the difference is generally larger when more threads are used.

To attempt to make the asynchronous method more robust, we test using a smaller value of β. This is analogous to underestimating the bounds of the spectrum in the inexact Chebyshev method [11]. Table 3 shows the convergence results using $\alpha = 1$ and $\beta = 0.9$. With this value of β, the asynchronous method is still not guaranteed to converge, but it can be observed that convergence is always obtained in the 100 runs for each number of threads. However, the convergence rate is degraded for this choice of β, i.e., compared to Table 2 when a small number of threads is used.

Comparing the asynchronous first and second order Richardson methods, the second order method can converge faster than the first order method. Convergence can be reliable although it is not guaranteed. In this example, the asynchronous method for second order Richardson, as reported in Table 3, is about 30% faster than the synchronous first order method.

8. Conclusion.

Except to say whether or not an asynchronous iterative method will converge in the asymptotic limit, the convergence behavior of these methods is strongly problem-dependent and computer platform-dependent and not well covered by theory. For the first and second order Richardson methods, in the setting where $\rho(T) < 1$, $T \geq 0$, and T is irreducible, this paper provides a description of the pa-

TABLE 3
Asynchronous second order Richardson for different numbers of threads. Parameter values: $\alpha = 1$ *and* $\beta = 0.9$.

number of threads	average range	average rel. resid. norm	number of failures	time (sec.)
1	0.0	9.566179e-05	0	0.053059
2	47.7	1.032052e-04	0	0.030998
3	105.8	1.802432e-04	0	0.019752
4	122.3	1.499666e-04	0	0.016426
5	148.3	2.081259e-04	0	0.013676
6	154.7	2.091337e-04	0	0.011510
7	208.8	2.745261e-04	0	0.010352
8	182.9	2.802124e-04	0	0.010104
9	230.9	3.434991e-04	0	0.009003
10	190.7	2.701899e-04	0	0.008824
11	185.7	3.500390e-04	0	0.008086
12	154.8	3.445788e-04	0	0.008059
13	198.9	6.526787e-04	0	0.008342
14	219.4	2.479312e-03	0	0.007052
15	212.1	8.821667e-03	0	0.008112
16	158.8	2.594421e-03	0	0.006902
17	227.1	1.113219e-03	0	0.006715
18	191.0	6.389028e-03	0	0.006050
19	227.5	1.464582e-03	0	0.006365
20	173.2	4.955854e-03	0	0.006487

rameter values for which the asynchronous versions of these methods are guaranteed to converge. Numerically, however, we find that this theoretical description can give a pessimistic view of asynchronous iterative methods. For a standard test problem, a multithreaded parallel implementation of asynchronous iterations can converge reliably in cases where it is theoretically possible for such iterations to diverge. How likely divergence will occur depends on the degree of asynchrony in the computation, which is difficult to quantify. A possible theoretical approach is to analyze asynchronous iterative methods as randomized algorithms [1].

Although we did not demonstrate it here, asynchronous iterative methods can give much lower time-to-solution than their synchronous counterparts when the computation is effectively unbalanced among the processing units. In such cases where the synchronization costs are large, the asynchronous second order Richardson method could still be used effectively with an appropriate choice of parameter values if the degree of asynchrony is controlled.

Acknowledgments. Work on this paper commenced while the three authors were attending a workshop at the Centre International de Rencontres Mathématiques, Luminy, France in September 2019. The center's support for such an event is greatly appreciated. Work of the first and third authors was supported in part by the U.S. Department of Energy under grants DE-SC-0016564 and DE-SC-0016578.

REFERENCES

[1] H. Avron, A. Druinsky, and A. Gupta, *Revisiting asynchronous linear solvers: Provable convergence rate through randomization*, Journal of the ACM, 62 (2015), pp. 51:1–51:27.

[2] G. M. Baudet, *Asynchronous iterative methods for multiprocessors*, Journal of the ACM, 25 (1978), pp. 226–244.

[3] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, NJ, 1989.

[4] I. Bethune, J. M. Bull, N. J. Dingle, and N. J. Higham, *Performance analysis of asynchronous Jacobi's method implemented in MPI, SHMEM and OpenMP*, International Journal on High Performance Computing Applications, 28 (2014), pp. 97–111.

[5] D. Chazan and W. L. Miranker, *Chaotic relaxation*, Linear Algebra and its Applications, 2 (1969), pp. 199–222.

[6] M. Eiermann and W. Niethammer, *On the construction of semiiterative methods*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 1153–1160.

[7] M. Eiermann, W. Niethammer, and R. S. Varga, *A study of semiiterative methods for nonsymmetric systems of linear equations*, Numerische Mathematik, 47 (1985), pp. 505–533.

[8] S. P. Frankel, *Convergence rates of iterative treatments of partial differential equations*, Mathematical Tables and Aids to Computations, 4 (1950), pp. 65–75.

[9] A. Frommer and D. B. Szyld, *On asynchronous iterations*, Journal of Computational and Applied Mathematics, 123 (2000), pp. 201–216.

[10] C. Glusa, E. G. Boman, E. Chow, S. Rajamanickam, and D. B. Szyld, *Scalable asynchronous domain decomposition solvers*, Tech. Report 19-10-11, Department of Mathematics, Temple University, October 2019. Revised April 2020.

[11] G. Golub and M. Overton, *The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems*, Numerische Mathematik, 53 (1988), pp. 571–594.

[12] G. H. Golub, *The use of Chebichev matrix polynomials in the iterative solution of linear equations compared to the method of succesive relaxation*, PhD thesis, Department of Mathematics, University of Illinois, Urbana, 1959.

[13] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods, Part I*, Numerische Mathematik, 3 (1961), pp. 147–156.

[14] J. Hook and N. Dingle, *Performance analysis of asynchronous parallel Jacobi*, Advances in Engineering Software, 77 (2018), pp. 831–866.

[15] F. Magoulès, D. B. Szyld, and C. Venet, *Asynchronous optimized Schwarz methods with and without overlap*, Numerische Mathematik, 137 (2017), pp. 199–227.

[16] L. F. Richardson, *The approximate arithmetical solution by finite differences of physical problems involving differential equations with an application to the stresses to a masonry dam*, Philosophical Transactions of the Royal Society of London, Series A, Mathematical and Physical Sciences, 210 (1910), pp. 307–357.

[17] Y. Saad, *Iterative methods for linear systems of equations: A brief historical journey.* arXiv:1908.01083 [math.HO], To appear in *Mathematics of Computation 75 Years*, Susanne C. Brenner, Igor Shparlinski, Chi-Wang Shu, and Daniel B. Szyld, editors, American Mathematical Society, Providence, RI, 2020.

[18] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1962. Second Edition, revised and expanded, Springer, Berlin, 2000.

[19] J. Wolfson-Pou and E. Chow, *Convergence models and surprising results for the asynchronous Jacobi method*, in 2018 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2018, Vancouver, BC, Canada, May 21-25, 2018, 2018, pp. 940–949.

[20] J. Wolfson-Pou and E. Chow, *Modeling the asynchronous Jacobi method without communication delays*, Journal of Parallel and Distributed Computing, 128 (2019), pp. 84–98.

[21] I. Yamazaki, E. Chow, A. Bouteiller, and J. Dongarra, *Performance of asynchronous optimized Schwarz with one-sided communication*, Parallel Computing, 86 (2019), pp. 66–81.

[22] D. M. Young, *Second-degree iterative methods for the solution of large linear systems*, Journal of Approximation Theory, 5 (1972), pp. 137–148.