Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 20/32

Edmond Chow, Andreas Frommer and Daniel B. Szyld

Asynchronous Richardson iterations

September 4, 2020

http://www.math.uni-wuppertal.de

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 whose values 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 the second order method, the parameter ranges for which we can prove asynchronous convergence do not contain the optimal parameter values for the synchronous iteration. In practice, however, the asynchronous second order iterations may still converge using the optimal parameter values, or parameter values close to the optimal ones, 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 system of equations $x = G(x)$ in fixed point form, where $G : \mathbb{R}^n \to \mathbb{R}^n$, which can be written componentwise 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, 4, 6],

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

where x_i^k denotes component i of the iterate at time instant k, J_k is the set of indices updated at instant k, and $s_j(k) \le k-1$ is the last instant component j was updated before being read when evaluating 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 [11]. For linear systems, where $G(x) = Tx + c$, $T \in \mathbb{R}^{n\times n}$, $c \in \mathbb{R}^n$, the pioneering result from [6] states that, under very mild conditions on the sets J_k and the sequences $s_j(k)$, any asynchronous iteration converges for any initial

*This version dated September 4, 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).

vector if and only if $\rho(|T|) < 1$. Here, $|T| \in \mathbb{R}^{n\times n}$ arises from T by taking absolute values for each entry and ρ denotes the spectral radius. The mild conditions on J_k and $s_j(k)$ are that

(1) $\lim_{k\to\infty} s_j(k) = \infty$ for $j = 1, \ldots, n$ and

(2) each $i \in \{1, \ldots, n\}$ appears infinitely many times in the sets J_k.

Since $\rho(T) \leq \rho(|T|)$, it appears that the condition for convergence of asynchronous iterations is more strict than that of synchronous iterations.

For linear systems $Ax = b$, asynchronous iterative methods that are based on the Jacobi or block Jacobi splitting, i.e., $T = I - D^{-1}A$ with D the diagonal or block diagonal of A, have been extensively studied, although these splittings generally give slow convergence; see [5, 16, 22, 23] for some recent references. In this paper, we consider first and second order Richardson methods [19]. If information on the bounds of the spectrum of A is available, this can be used to determine the parameter values to use for the Richardson methods, and the second order Richardson method, in particular, then converges rapidly. This paper explores the parameter values that can be proven to give convergence of asynchronous Richardson methods. In particular, it presents the first such analysis for second order methods.

Statements about the *rate of convergence*, however, cannot be made without a description of the sets J_k and the sequences $s_j(k)$. Both 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, while it may behave very differently if load is unbalanced or communication costs are high. In this paper, we will therefore not go into the details of an analysis of the convergence rate, but we will 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 [15]. For a short historical description of the development of these methods, see [20]. Unlike those Krylov subspace methods which rely on a variational principle, the second order Richardson and Chebyshev methods do not require inner products, which is what allows them to be easily executed asynchronously.

In recent related work, asynchronous versions of Schwarz and optimized Schwarz methods have been developed [12, 17, 25].

2. The setting.

From the beginning, we assume that the original system

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

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

$$Ax = c, \text{ where } A = M^{-1}\hat{A} = I - T, \;\; c = M^{-1}\hat{b}. \tag{3}$$

For the convergence results on asynchronous Richardson iterations to come, we will always assume that the following assumptions are met:

$$T \text{ is non-negative, i.e. } T \geq 0 \text{ where } \geq \text{ is to be understood entrywise,} \tag{4}$$

$$T \text{ is convergent, i.e. } \rho(T) < 1, \tag{5}$$

$$\operatorname{spec}(A) \subset \mathbb{R}^+. \tag{6}$$

In other words, we are assuming that $\hat{A} = M - N$ is a convergent weak splitting in the sense of [18][1] with the additional property that the spectrum of T is real. Note that if $\hat{A}$ is symmetric and positive definite (spd), and M is the diagonal or a block diagonal of $\hat{A}$, which then is also spd, (6) is fulfilled. If, in addition, $\hat{A}$ is a Stieltjes matrix, i.e. an M-matrix which is spd, and if again M is the diagonal or a block diagonal of $\hat{A}$, then (4) and (5) are also fulfilled; see [3, Chapter 5], [21, Section 3.5], [26, Chapter 11].

With the splitting $A = I - T$, the standard, synchronous iterative method is as follows. Given x^0, for $k = 0, 1, \ldots$, compute

$$x^{k+1} = Tx^k + c. \tag{7}$$

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

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

3. First order Richardson. The first order Richardson method consists of taking a linear combination of the previous iterate with that which would come from the standard iteration (7). This method can be seen as the simplest case of a semi-iterative method [8, 9, 21, 26]. The sum of the coefficients of the linear combination must add up to one, since otherwise the method will not produce iterates that converge towards $A^{-1}b$.[2]

We first consider the stationary case where the parameter α defining the Richardson iteration is fixed for all iterations. We consider later the non-stationary case where $\alpha = \alpha_k$ depends on the iteration number.

The synchronous stationary iteration is

$$x^{k+1} = (1-\alpha)x^k + \alpha(Tx^k + c) = x^k + \alpha[c - (I-T)x^k] = x^k + \alpha r^k, \tag{8}$$

where $r^k = c - (I - T)x^k$ is the residual of the equivalent system (3).

The convergence analysis of this synchronous method is straight-forward and well-known; see [26, Section 11.4]. The analysis consists of analyzing the spectral radius of the iteration matrix

$$T_\alpha = (1-\alpha)I + \alpha T = I - \alpha(I - T) = I - \alpha A.$$

If $\mu \in \operatorname{spec}(T_\alpha)$, then $\mu = 1 - \alpha + \alpha\lambda$, with $\lambda \in \operatorname{spec}(T)$, i.e., $\lambda \in [-\rho, \rho]$.

[1] See also [24, 7].

[2] Gene Golub in his thesis [14] calls this a method of averaging, following the nomenclature used by von Neumann.

THEOREM 1. *Let* $\mathrm{spec}(A) \subset \mathbb{R}^+$. *Then*
(i) *iteration* (8) *converges if* $\alpha \in (0,\ 2/\lambda_{\max})$,
(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* $\mathrm{spec}(A) \subset [a, b]$, $a > 0$ *is* $\alpha = 2/(a + b)$.

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

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

From this we see that $\rho(T_\alpha) < 1$ iff $\alpha \in (0,\ 2/\lambda_{\max})$, 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 $\mathrm{spec}(A) \subset [1 - \rho,\ 1 + \rho]$, and, by (iii) the optimal α w.r.t. this information is $\alpha = 1$.

For the asynchronous iteration, we analyze when $\rho(|T_\alpha|) < 1$. We adopt the notation $w > 0$ for $w \in \mathbb{R}^n$ if $w_i > 0$ for $i = 1, \ldots, n$. Our analysis relies on the following often-used fact from non-negative matrix theory which we restate with its proof for convenience.

LEMMA 2. *Let* $T \in \mathbb{R}^{n\times n}$, $T \geq 0$ *with spectral radius* ρ. *Then for every* $\varepsilon > 0$ *there exists a positive vector* $w_\varepsilon > 0$, $w_\varepsilon \in \mathbb{R}^n$, *such that*

$$T w_\varepsilon \leq (\rho + \varepsilon) w_\varepsilon.$$

Proof. For $\delta > 0$, let

$$T_\delta = T + \delta E, \quad \text{where } E = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}.$$

Then T_δ has only positive entries, and by the Perron-Frobenius Theorem, see [3, 21], e.g., there exists $w_\delta > 0$ such that $T_\delta w_\delta = \rho(T_\delta) w_\delta$ which, since $E w_\delta \geq w_\delta$, gives

$$T w_\delta \leq (\rho(T_\delta) - \delta) w_\delta. \tag{9}$$

By continuity of the spectral radius, we can choose $\delta = \delta(\varepsilon)$ such that $\rho(T_{\delta(\varepsilon)}) \leq \rho + \varepsilon$, so that (9) becomes the assertion of the lemma (with $w_\varepsilon = w_{\delta(\varepsilon)}$). ◻

In Theorem 3 below, as well as in Theorem 6, we will also use the fact that if, for $T \in \mathbb{R}^{n\times n}$, $T \geq 0$ and $w \in \mathbb{R}^n$, $w > 0$, we have $Tw \leq \nu w$, then $\rho(T) \leq \nu$. This follows immediately from observing that $Tw \leq \nu w$ is equivalent to $\|T\|_w \leq \nu$ where $\|\cdot\|_w$ is the matrix norm induced by the weighted maximum norm $\|x\|_w = \max_{i=1}^n |x_i/w_i|$ on $\mathbb{R}^n$.

THEOREM 3. *Assume that* (4), (5) *and* (6) *hold and let* $\rho = \rho(T)$. *Then* $\rho(|T_\alpha|) < 1$ *if* $\alpha \in (0, \frac{2}{1+\rho})$, *where* $\frac{2}{1+\rho} > 1$.

Proof. Let $\varepsilon > 0$ and, by Lemma 2, let $w_\varepsilon > 0$ be a vector for which $T w_\varepsilon \leq (\rho + \varepsilon) w_\varepsilon$. Then we have

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

For $0 < \alpha \leq 1$ we have $0 \leq \nu = (1-\alpha) + (\rho+\varepsilon)\alpha = 1 - \alpha(1-(\rho+\varepsilon))$ which is less than 1 if $\varepsilon > 0$ is taken small enough. For $1 < \alpha < \frac{2}{1+\rho}$ we have $0 < \nu = (\alpha - 1) + (\rho+\varepsilon)\alpha = (1+\rho+\varepsilon)\alpha - 1$ which, for α fixed, is again less than 1 for ε sufficiently small. ∎

We note that $\alpha = 1$, the optimal parameter value for the synchronous iteration w.r.t the information $\mathrm{spec}(A) \subseteq [1-\rho, 1+\rho]$, 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 $0 < \alpha_k \leq \overline{\alpha} < \frac{2}{1+\rho}$, the "non-stationary" asynchronous method converges as well, using [11, Corollary 3.2]. In fact, using the latter result, we have the following theorem.

THEOREM 4. *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 using the vectors w_ε from Lemma 2 for T and by observing that with $T_k = (1-\alpha_k)I + \alpha_k T$ we have

$$\|(1-\alpha_k)I + \alpha_k T\|_{w_\varepsilon} \leq |1-\alpha_k| + \alpha_k(\rho+\varepsilon) \leq |1-\overline{\alpha}| + \overline{\alpha}(\rho+\varepsilon).$$

Taking $\varepsilon > 0$ such that $\rho + \varepsilon < 1$ and $(1+\rho+\epsilon)\overline{\alpha} - 1 < 1$ gives $|1-\overline{\alpha}| + \overline{\alpha}(\rho+\varepsilon) < 1$.

4. Second order Richardson. The second order Richardson method is the semi-iterative method one obtains when correcting x^k with a linear combination of $(x^k - x^{k-1})$ and the residual at step k, rather than just the residual as used in the standard iteration (7). Equivalently, one can take x^{k+1} to be a linear combination of the first order Richardson iterate (8) with just x^{k-1}, as follows,

$$\begin{aligned}
x^{k+1} &= (1+\beta)[(1-\alpha)x^k + \alpha(Tx^k + c)] - \beta x^{k-1} \\
&= -\beta x^{k-1} + (1+\beta)x^k + (1+\beta)\alpha[-x^k + Tx^k + c)] \\
&= x^k - \beta(x^{k-1} - x_k) + (1+\beta)\alpha[c - (I-T)x^k)] \\
&= x^k + \beta(x^k - x^{k-1}) + (1+\beta)\alpha(c - Ax^k) \\
&= (1+\beta)(I-\alpha A)x^k - \beta x^{k-1} + (1+\beta)\alpha c, \; k = 1, 2, \ldots.
\end{aligned} \tag{10}$$

In addition to x^0, it is now necessary to also prescribe x^1, and for this it is possible to use one step of (7) or one step of first order Richardson [14].

The results to come are more restrictive 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 for the synchronous iteration.

We can write the three-term recurrence in (10) using a matrix of doubled size as follows, cf. [27],

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

For the synchronous iteration (10), two approaches have been used to analyze convergence. For the first approach [27], we note that if λ is an eigenvalue of $T_{\alpha,\beta}$ with eigenvector $\left[\begin{smallmatrix} s \\ t \end{smallmatrix}\right]$, 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 $\mu \in \mathrm{spec}(A)$, the eigenvalues of $T_{\alpha,\beta}$ must satisfy the quadratic equation

$$\lambda^2 - (1+\beta)(1-\alpha\mu)\lambda + \beta = 0. \tag{11}$$

Figure 1 (first column) plots the spectral radius of $T_{\alpha,\beta}$ as a function of α and β for three examples.

Frankel [10] shows that the values of the parameters α and β that minimize the maximum of the modulus of the solution of (11) are given by $\alpha = 2/(a+b)$ and $\beta = \left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right)^2 := q^2$, for A such that $\mathrm{spec}(A) \subset [a,b]$ with $a > 0$. The resulting minimal value for $\rho(T_{\alpha,\beta})$, the spectral radius of the iteration operator, is q.

For the second approach [14, 15], assuming one uses the above optimal parameters, the recurrence of the polynomials defining (10) is used to bound the 2-norm of the error as

$$\|x^k - x^*\|_2 \leq \left[q^k\left(1 + k\frac{1-q^2}{1+q^2}\right)\right]\|x^0 - x^*\|_2, \tag{12}$$

where x^* is the solution of (3). Here, it is assumed that the first iterate is $x^1 = x^0 + \alpha(b - Ax^0)$.

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

THEOREM 5. *Let A be spd. Then*

(i) the optimal parameters w.r.t. the information $\mathrm{spec}(A) \subset [a,b]$ *with $a > 0$ are $\alpha = 2/(a+b)$ and $\beta = \left(\frac{b-a}{a+b+2\sqrt{ab}}\right)^2 = \left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right)^2$, and the asymptotic convergence factor $\rho(T_{\alpha,\beta})$ is equal to $q = \frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}$,*

(ii) with these parameters and with $x^1 = x^0 + \alpha(b - Ax^0)$, a bound for the 2-norm of the errors is given in (12).

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

THEOREM 6. *Assume that* (4), (5) *and* (6) *are fulfilled and let $\rho = \rho(T)$. Then we have $\rho(|T_{\alpha,\beta}|) < 1$, provided*

$$\alpha > 0 \text{ and } |1+\beta|(|1-\alpha| + \alpha\rho) + |\beta| < 1. \tag{13}$$

Before we prove the theorem, consider the choice $\alpha = 1$. For this choice, the theorem states that asynchronous iterations converge for $-1 \leq \beta < \frac{1-\rho}{1+\rho}$, as can be seen from considering the two cases $\beta \geq 0$ and $-1 < \beta < 0$ separately. If the information about the spectral interval is $\mathrm{spec}(A) \subset [1-\rho, 1+\rho]$, Theorem 5 gives that the optimal α for the synchronous iteration is $\alpha = 1$, and the corresponding optimal β will be close to 1 for ρ close to 1. The range of β for which Theorem 6 guarantees convergence of the asynchronous iteration for $\alpha = 1$, however, has $1-\rho$ as an upper bound for β according to (13), and this will be close to 0 if ρ is close to 1.

Proof of Theorem 6. Let $\varepsilon > 0$ be small enough such that we still have

$$|1+\beta|(|1-\alpha| + \alpha(\rho+\varepsilon)) + |\beta| < 1,$$

and let $w_\varepsilon > 0$ be a vector with $Tw_\varepsilon \le (\rho+\varepsilon)w_\varepsilon$ which exists by Lemma 2. Let $\gamma > 1$ and consider the vector $\left[\begin{smallmatrix} w_\varepsilon \\ \gamma w_\varepsilon \end{smallmatrix}\right]$. Then, if $\alpha > 0$, we have

$$\begin{aligned} |T_{\alpha,\beta}| \begin{bmatrix} w_\varepsilon \\ \gamma w_\varepsilon \end{bmatrix} &= \begin{bmatrix} |1+\beta| \cdot |I-\alpha A| & |\beta| I \\ I & 0 \end{bmatrix} \begin{bmatrix} w_\varepsilon \\ \gamma w_\varepsilon \end{bmatrix} \\ &\le \begin{bmatrix} (|1+\beta| \cdot (|1-\alpha| + \alpha(\rho+\varepsilon)) + |\beta|\gamma) w_\varepsilon \\ w_\varepsilon \end{bmatrix} \le \sigma_\varepsilon \begin{bmatrix} w_\varepsilon \\ \gamma w_\varepsilon \end{bmatrix}, \end{aligned}$$

with

$$\sigma_\varepsilon = \max\{\tfrac{1}{\gamma},\ |1+\beta| \cdot (|1-\alpha| + \alpha(\rho+\varepsilon)) + |\beta|\gamma\}. \tag{14}$$

Now, since $|1+\beta|(|1-\alpha| + \alpha(\rho+\varepsilon)) + |\beta| < 1$, we can choose $\gamma > 1$ close enough to 1 such that we also have $|1+\beta|(|1-\alpha| + \alpha(\rho+\varepsilon)) + \gamma|\beta| < 1$, which gives $\sigma_\varepsilon < 1$ in (14). ∎

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 $0 \le \nu < 1$ and $\nu \ge 1$. In the first case, we obtain that $|1+\beta|\nu + |\beta| < 1$ if $-1 \le \beta < \frac{1-\nu}{1+\nu}$. In the second case, there is no β which satisfies the inequality.

To compare with (11), let us 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 see that $s = \lambda t$. Then, 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 thus have that λ satisfies the quadratic equation

$$\lambda^2 - |1+\beta|\mu\lambda - |\beta| = 0. \tag{15}$$

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

5. 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 (11) and (15). 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 $\beta \le 0$, $\rho(|T_{\alpha,\beta}|)$ 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. The effect is that $\rho(|T_{\alpha,\beta}|)$ is smallest for $\beta = 0$, which corresponds to the first order method.

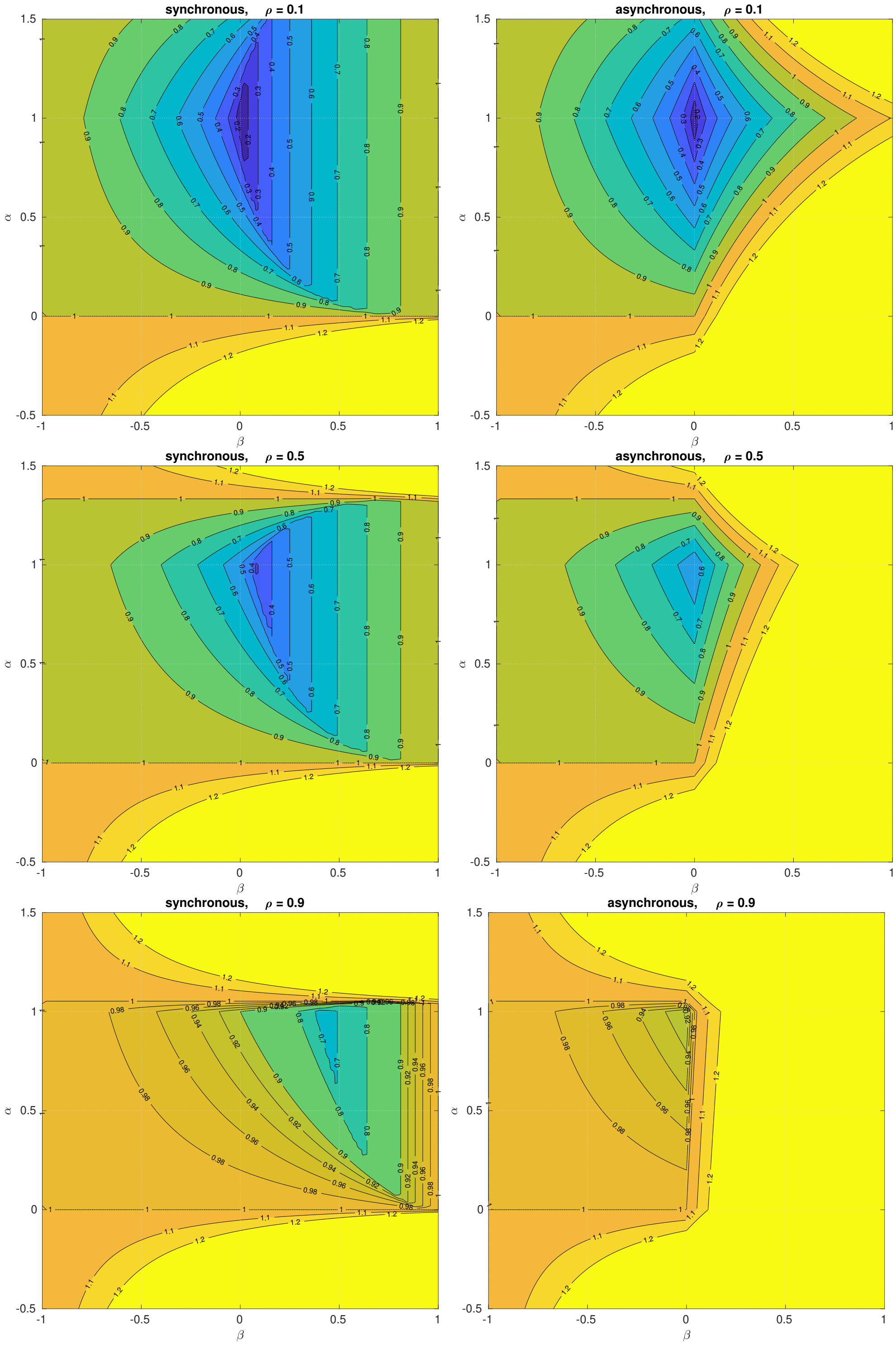

FIG. 1. *Spectral radius of $T_{\alpha,\beta}$ (synchronous case) and of $|T_{\alpha,\beta}|$ (asynchronous case) as a function of α and β when $\lambda_{\min}(A) = 1-\rho$ and $\lambda_{\max}(A) = 1+\rho$, for three values of ρ.*

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, minimizing $\rho(T_{\alpha,\beta})$, is approximately 0.3929. Regarding the asynchronous method we have that $|T_{\alpha,\beta}|$ 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 specific asynchronous iterations, i.e., any choice of the delays, $k - s_j(k)$, and any choice of the sets J_k of components to update (satisfying the mild conditions (1) and (2)). 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.

6. 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 A arises from the standard finite difference Laplacian matrix $\hat{A}$ on a 100×100 grid of unknowns. With Jacobi preconditioning, the preconditioned matrix A remains spd and thus satisfies (6), while $T = I - A$ is the iteration matrix which satisfies (4) and (5). A right-hand side vector was chosen randomly with components chosen independently from the uniform distribution on $(-0.5, 0.5)$. The same vector was used for all tests. The initial vector for all iterations was zero.

Different numbers of threads were used in different tests. Each thread was assigned approximately the same number of 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.

6.1. First order Richardson. For the asynchronous first order Richardson method, Table 1 shows the convergence results for tests with 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 1.691939×10^{-2}. 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 [22, 23], 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 7.421009×10^{-3} which is lower than that of the synchronous 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.

TABLE 1
Asynchronous first order Richardson for different numbers of threads. For comparison, the synchronous method attains an average relative residual norm of 1.691939×10^{-2} *for all numbers of threads. Timings for the asynchronous and synchronous methods for performing a fixed number of iterations are also given.*

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

The table also shows timings for the asynchronous method and the synchronous method for different numbers of threads (for performing a fixed number of iterations). 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 (in performing 500 iterations), 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.

6.2. 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

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 1.258388×10^{-7} 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	1.258388×10^{-7}	0	0.053275	0.052961
2	40.8	4.235170×10^{-7}	0	0.031146	0.032542
3	104.3	6.175605×10^{-6}	0	0.019592	0.023368
4	115.7	1.444428×10^{-5}	0	0.016493	0.018801
5	166.0	1.495107×10^{-4}	0	0.013533	0.017519
6	163.0	4.524130×10^{-4}	0	0.011563	0.014606
7	200.1	1.868556×10^{-3}	0	0.010649	0.013078
8	151.5	9.259216×10^{-3}	0	0.009794	0.012843
9	246.0	4.035731×10^{-2}	1	0.008917	0.012560
10	203.2	1.088207×10^{-1}	1	0.009000	0.012371
11	209.4	4.582844×10^{-1}	21	0.008972	0.011905
12	185.5	$1.678645 \times 10^{+0}$	25	0.008397	0.011527
13	227.6	$1.046313 \times 10^{+1}$	32	0.008216	0.011698
14	205.9	$3.971405 \times 10^{+1}$	43	0.007081	0.010863
15	239.3	$5.207066 \times 10^{+2}$	35	0.007568	0.010828
16	166.8	$2.317140 \times 10^{+2}$	24	0.007101	0.011470
17	226.3	$3.303636 \times 10^{+1}$	22	0.006217	0.011161
18	191.8	$6.415417 \times 10^{+1}$	30	0.005972	0.010969
19	237.6	$2.377968 \times 10^{+1}$	23	0.006237	0.011147
20	173.8	$3.136173 \times 10^{+1}$	46	0.006614	0.011012

asynchronous method always converged in the 100 runs, with a degradation in the "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 performing a fixed number of iterations) 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 [13]. 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.

6.3. Synchronous and asynchronous convergence timings. In the previous subsections, we compared the timings of asynchronous and synchronous iterations for a fixed number of iterations. In this subsection, we compare the residual norms that are achieved in parallel implementations of the synchronous and asynchronous methods as a function of time.

In these tests, we used ten threads on a single Intel Xeon processor with 10 cores, with each thread pinned to one of the two hyperthreads on each core. The test matrix is again from the standard finite difference Laplacian matrix, but now on a 300×300

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	9.566179×10^{-5}	0	0.053059
2	47.7	1.032052×10^{-4}	0	0.030998
3	105.8	1.802432×10^{-4}	0	0.019752
4	122.3	1.499666×10^{-4}	0	0.016426
5	148.3	2.081259×10^{-4}	0	0.013676
6	154.7	2.091337×10^{-4}	0	0.011510
7	208.8	2.745261×10^{-4}	0	0.010352
8	182.9	2.802124×10^{-4}	0	0.010104
9	230.9	3.434991×10^{-4}	0	0.009003
10	190.7	2.701899×10^{-4}	0	0.008824
11	185.7	3.500390×10^{-4}	0	0.008086
12	154.8	3.445788×10^{-4}	0	0.008059
13	198.9	6.526787×10^{-4}	0	0.008342
14	219.4	2.479312×10^{-3}	0	0.007052
15	212.1	8.821667×10^{-3}	0	0.008112
16	158.8	2.594421×10^{-3}	0	0.006902
17	227.1	1.113219×10^{-3}	0	0.006715
18	191.0	6.389028×10^{-3}	0	0.006050
19	227.5	1.464582×10^{-3}	0	0.006365
20	173.2	4.955854×10^{-3}	0	0.006487

grid of unknowns.

The $90,000$ unknowns were partitioned into 10 partitions and each thread was assigned to update the unknowns in one partition. Two types of partitionings were used: *balanced*, where each partition contains 9000 unknowns; and *unbalanced*, where 5 partitions contain 6000 unknowns and 5 partitions contain 12000 unknowns.

Figure 2 shows the results for the first order Richardson method (using the optimal $\alpha = 1$). This figure was generated by running the parallel method for a fixed number of iterations, t, in the synchronous case, or when all threads have executed an average of t iterations in the asynchronous case. The relative residual norm was then computed. For a given value of t, 20 tests were performed and the average relative residual norm was computed. These averages for different values of t are plotted in the figure, where the x-axis is the average execution time for tests with a given t. The variations in residual norms and timings for a given t are very small, and practically indiscernible from the averages if they were plotted.

Figure 2 shows that, for first order Richardson for the given test system, the asynchronous method is faster than the synchronous method. The unbalanced case is significantly slower than the balanced case for the synchronous iteration, whereas for the asynchronous iteration there is only a minor difference between the balanced and the unbalanced case.

Figure 3 shows the results for the second order Richardson method, using the optimal values of α and β. This figure was generated in the same way as the previous figure, but here, the result of each of the 20 tests is plotted individually in the asynchronous case, since the variations in the results are now much larger. For the synchronous iteration (solid lines), the unbalanced case is slower than the balanced case, as expected. The asynchronous iteration (circles) is sometimes faster and sometimes slower than the synchronous iteration (circles above and below the solid lines of the same color). We also observe that the asynchronous method is substantially

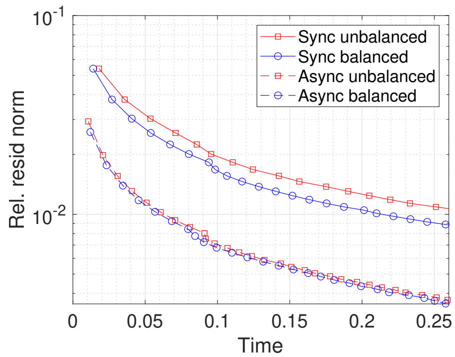

FIG. 2. *First order Richardson convergence with time.*

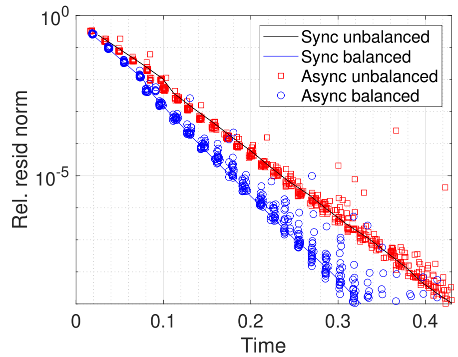

FIG. 3. *Second order Richardson convergence with time.*

slower when the partitions are unbalanced, compared to when they are balanced. This is in contrast to the observation for first order Richardson, which was not as sensitive to imbalance. As we had observed in Section 6.2, an increased degree of asynchrony is detrimental to the convergence of the second order Richardson method, and here it is the load imbalance that increases the degree of asynchrony.

Comparing the asynchronous first and second order Richardson methods for the given test problem with the use of optimal parameter values, the second order method can converge much faster than the first order method. Convergence can be reliable although it is not guaranteed.

7. 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 $\mathrm{spec}(A) \subset \mathbb{R}^+$, this paper provides a description of the parameter 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].

Asynchronous execution of the first order Richardson method can clearly give much lower time-to-solution than synchronous execution. Asynchronous execution of the second order Richardson method may be slightly faster than synchronous execution because each iteration performed by a thread is executed more rapidly. On the other hand, execution may be slower because asynchrony is detrimental to the convergence of the method. The second order method, with its use of not one but two previous iterates, appears to require much tighter coupling between the threads that are working in parallel.

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] Haim Avron, Alex Druinsky, and Anshul Gupta. Revisiting asynchronous linear solvers: Provable convergence rate through randomization. *Journal of the ACM*, 62(6):51:1–51:27, December 2015.

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

[3] Abraham Berman and Robert J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York, third edition, 1979. Reprinted by SIAM, Philadelphia, 1994.

[4] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, NJ, 1989.

[5] Iain Bethune, J Mark Bull, Nicholas J Dingle, and Nicholas J Higham. Performance analysis of asynchronous Jacobi's method implemented in MPI, SHMEM and OpenMP. *International Journal on High Performance Computing Applications*, 28(1):97–111, February 2014.

[6] Dan Chazan and Willard L. Miranker. Chaotic relaxation. *Linear Algebra and its Applications*, 2:199–222, 1969.

[7] Joan-Josep Climent and Carmen Perea. Some comparison theorems for weak nonnegative splittings of bounded operators. *Linear Algebra and its Applications*, 275–276:77–106, 1998.

[8] Michael Eiermann and Wilhelm Niethammer. On the construction of semiiterative methods. *SIAM Journal on Numerical Analysis*, 20:1153–1160, 1983.

[9] Michael Eiermann, Wilhelm Niethammer, and Richard S. Varga. A study of semiiterative methods for nonsymmetric systems of linear equations. *Numerische Mathematik*, 47:505–533, 1985.

[10] Stanley P. Frankel. Convergence rates of iterative treatments of partial differential equations. *Mathematical Tables and Aids to Computations*, 4:65–75, 1950.

[11] Andreas Frommer and Daniel B. Szyld. On asynchronous iterations. *Journal of Computational and Applied Mathematics*, 123:201–216, 2000.

[12] Christian Glusa, Erik G. Boman, Edmond Chow, Sivasankaran Rajamanickam, and Daniel B. Szyld. Scalable asynchronous domain decomposition solvers. Technical Report 19-10-11, Department of Mathematics, Temple University, October 2019. Revised April 2020and July 2020. To appear in *SIAM Journal on Scientific Computing*.

[13] Gene Golub and Michael Overton. The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems. *Numerische Mathematik*, 53(5):571–594, 1988.

[14] Gene 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.

[15] Gene H. Golub and Richard S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods, Part I. *Numerische Mathematik*, 3:147–156, 1961.

[16] James Hook and Nicholas Dingle. Performance analysis of asynchronous parallel Jacobi. *Advances in Engineering Software*, 77(3):831–866, 2018.

[17] Frédéric Magoulès, Daniel B. Szyld, and Cédric Venet. Asynchronous optimized Schwarz methods with and without overlap. *Numerische Mathematik*, 137:199–227, 2017.

[18] Ivo Marek and Daniel B. Szyld. Comparison theorems for weak splittings of bounded operators. *Numerische Mathematik*, 58:387–397, 1990.

[19] Lewis 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*, 210:307–357, 1910.

[20] Yousef 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.

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

[22] Jordi Wolfson-Pou and Edmond Chow. Convergence models and surprising results for the asyn-

chronous Jacobi method. In *2018 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2018, Vancouver, BC, Canada, May 21-25, 2018*, pages 940–949, 2018.

[23] Jordi Wolfson-Pou and Edmond Chow. Modeling the asynchronous Jacobi method without communication delays. *Journal of Parallel and Distributed Computing*, 128:84–98, 2019.

[24] Zbigniew I. Woźnicki. Nonnegative splitting theory. *Japan Journal of Industrial and Applied Mathematics*, 11:289–342, 1994.

[25] Ichitaro Yamazaki, Edmond Chow, Aurélien Bouteiller, and Jack Dongarra. Performance of asynchronous optimized Schwarz with one-sided communication. *Parallel Computing*, 86:66–81, 2019.

[26] David M. Young. *Iterative Solution of Large Linear Systems*. Academic Press, New York, 1971.

[27] David M. Young. Second-degree iterative methods for the solution of large linear systems. *Journal of Approximation Theory*, 5:137–148, 1972.