IBM Research Report

A Randomized Asynchronous Linear Solver with Provable Convergence Rate

Haim Avron[1], Alex Druinsky[2], Anshul Gupta[1]

[1]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 208
Yorktown Heights, NY 10598
USA

[2]Tel Aviv University

A Randomized Asynchronous Linear Solver with Provable Convergence Rate

Haim Avron
IBM T.J. Watson Research Center
haimav@us.ibm.com

Alex Druinsky
Tel Aviv University
alexdrui@post.tau.ac.il

Anshul Gupta
IBM T.J. Watson Research Center
anshul@us.ibm.com

Abstract

Asynchronous methods for solving systems of linear equations have been researched since Chazan and Miranker published their pioneering paper on chaotic relaxation in 1969. The underlying idea of asynchronous methods is to avoid processor idle time by allowing the processors to continue to work and make progress even if not all progress made by other processors has been communicated to them.

Historically, work on asynchronous methods for solving linear equations focused on proving convergence in the limit. How the rate of convergence compares to the rate of convergence of the synchronous counterparts, and how it scales when the number of processors increase, was seldom studied and is still not well understood. Furthermore, the applicability of these methods was limited to restricted classes of matrices (e.g., diagonally dominant matrices).

We propose a shared-memory asynchronous method for general symmetric positive definite matrices. We rigorously analyze the convergence rate and prove that it is linear and close to that of our method's synchronous counterpart as long as not too many processors are used (relative to the size and sparsity of the matrix). A key component is randomization, which allows the processors to make guaranteed progress without introducing synchronization. Our analysis shows a convergence rate that is linear in the condition number of the matrix, and depends on the number of processors and the degree to which the matrix is sparse.

1 Introduction

Asynchronous methods for solving linear equations have been researched since Chazan and Miranker published their pioneering paper on chaotic relaxation in 1969 [4] (see review by Frommer and Szyld [6]). The underlying idea of asynchronous methods is to allow processors to continue to work even if not all progress made by other processors has been communicated to them, thereby eliminating synchronization points and their associated cost.

While asynchronous methods were successfully applied to many numerical problems [6], interest in them dwindled over the years. One reason is related to convergence rate. Historically, work on asynchronous methods for solving linear equations focused on proving convergence in the limit. How the rate of convergence compares to the rate of convergence of the synchronous counterparts, and how it scales when the number of processors increase, was seldom studied and is still not well understood. It was observed experimentally that asynchronous methods can sometimes be substantially slower than their synchronous counterparts [3]. Furthermore, the applicability of existing asynchronous methods is limited to rather restricted classes of matrices (e.g., diagonally dominant matrices).

We propose a shared-memory asynchronous method for symmetric positive definite matrices with a provable linear convergence rate under a mostly asynchronous computational model which assumes bounded delays. A key component of our algorithm is randomization, which allows the processors to make progress independently with only a small probability of interfering with each other. Our analysis shows a convergence rate that is linear in the condition number of the matrix, and depends on the number of processors and the degree to which the matrix is sparse. A slightly better bound is achieved if we occasionally synchronize the processors. In either case, as long the number of processors used is not too large (relative to the size and sparsity of the matrix), the convergence rate is close to that of the synchronous counterpart. Unlike previous asynchronous methods, the convergence rate does not depend on numerical classification of the matrix (e.g., diagonal dominance). In particular, our method will converge for essentially *any* large sparse symmetric positive definite matrix as long as not too many processors are used. We discuss previous work, and contrast them to ours, in Section 7, after we describe our results in detail.

The remainder of the paper is organized as follows. We describe the basic setup and give essential background on randomized Gauss-Seidel in Section 2. In Section 3 we propose two asynchronous models for executing randomized Gauss-Seidel: one assumes that consistent reads have been enforced, and one does not. Section 4 analyzes the convergence when consistent read is enforced. Section 5 shows that convergence can be improved if we control the step-size. In Section 6, we analyze convergence rate when we allow inconsistent reads. Finally, in Section 7 we discuss previous results and how our results relate to them.

2 Preliminaries

2.1 Setup and Notation

This paper concerns with solving the linear equation $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, and $\mathbf{b} \in \mathbb{R}^n$. For simplicity we will assume that $\mathbf{A}$ has a unit diagonal. This is easily accomplished using re-scaling. Our results can be easily generalized to allow an arbitrary diagonal, but making this assumption helps keep the presentation and notation more manageable. We denote the exact solution to this equation by $\mathbf{x}^\star$, i.e. $\mathbf{x}^\star = \mathbf{A}^{-1}\mathbf{b}$. We denote the largest eigenvalue of $\mathbf{A}$ by $\lambda_{\max}$, and the smallest eigenvalue by $\lambda_{\min}$. The condition number of $\mathbf{A}$, which is equal to $\lambda_{\max}/\lambda_{\min}$, is denoted by κ.

We are predominantly interested in the case that $\mathbf{A}$ is sparse and very large, and the number of non-zeros in each row is between C_1 and $C_2 \ll n$ with a small ratio between C_2 and C_1. This scenario frequently occurs in many scientific computing applications. Throughout the paper we refer to this scenario as the *reference scenario.* We state and prove more general results; we do not use the properties of the reference scenario in the proofs. The reference scenario is mainly useful for the interpretation of the practical implications of the results. Note that in the reference scenario we have $\lambda_{\max} \leq C_2 \ll n$, as $\mathbf{A}$ has a unit-diagonal (so off-diagonal entries must be smaller than or equal to one).

We use $(\cdot, \cdot)_{\mathbf{A}}$ to denote the $\mathbf{A}$ inner product. That is, $(\mathbf{x}, \mathbf{y})_{\mathbf{A}} \equiv \mathbf{y}^{\mathrm{T}} \mathbf{A} \mathbf{x}$ where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. The fact that $\mathbf{A}$ is a symmetric positive definite matrix guarantees that $(\cdot, \cdot)_{\mathbf{A}}$ is an inner product. The $\mathbf{A}$-norm is defined by $\|\mathbf{x}\|_{\mathbf{A}} \equiv \sqrt{(\mathbf{x}, \mathbf{x})_{\mathbf{A}}}$. We use $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \dots, \mathbf{e}^{(n)}$ to denote the n-dimensional identity vectors (i.e. $\mathbf{e}^{(i)}$ is one at at position i and zero elsewhere). $\mathbf{A}_i$ denotes row i of $\mathbf{A}$, and $\mathbf{A}_{ij}$ denotes the i, j entry of $\mathbf{A}$. We will generally use subscript indexes on vectors for iteration counters. The notation $(\mathbf{x})_i$ denotes the ith entry of $\mathbf{x}$.

Throughout the paper we describe algorithms that generate a series of approximations to $\mathbf{x}^{\star}$, denoted by $\mathbf{x}_0, \mathbf{x}_1, \dots$ (subscript index is the iteration counter), which are actually random vectors. We denote the expected squared $\mathbf{A}$-norm of the error of $\mathbf{x}_m$ by E_m. That is,

$$E_m = \mathbb{E}\left[\|\mathbf{x}_m - \mathbf{x}^{\star}\|_{\mathbf{A}}^2\right] .$$

2.2 Randomized Gauss-Seidel

Our asynchronous algorithm is based on the randomized variant of the Gauss-Seidel iteration, originally proposed by Leventhal and Lewis [8] (see also Griebel and Oswald [7]). The goal of this section is to describe and review the basic properties of the randomized Gauss-Seidel iteration.

Consider the following iteration applied to some arbitrary initial vector $\mathbf{x}_0 \in \mathbb{R}^n$, and a series of direction vectors $\mathbf{d}_0, \mathbf{d}_1, \dots$:

$$\mathbf{r}_j = \mathbf{b} - \mathbf{A}\mathbf{x}_j$$
$$\gamma_j = \mathbf{d}_j^{\mathrm{T}} \mathbf{r}_j$$
$$\mathbf{x}_{j+1} = \mathbf{x}_j + \gamma_j \mathbf{d}_j .$$

In terms of the analysis it is more convenient to write the iteration in the following equivalent form:

$$\begin{aligned} \gamma_j &= (\mathbf{x}^{\star} - \mathbf{x}_j, \mathbf{d}_j)_{\mathbf{A}} \\ \mathbf{x}_{j+1} &= \mathbf{x}_j + \gamma_j \mathbf{d}_j. \end{aligned} \tag{1}$$

The reason for listing both iterations is to show that even though $\mathbf{x}^{\star}$ (which is unknown) appears in (1), the iteration is computable.

In (1) the scalars $\gamma_0, \gamma_1, \dots$ are selected so as to minimize $\|\mathbf{x}^{\star} - \mathbf{x}_{j+1}\|_{\mathbf{A}}$ when $\mathbf{x}_{j+1}$ is obtained from $\mathbf{x}_j$ by taking a step in the direction $\mathbf{d}_j$. There are quite a few ways to set $\mathbf{d}_0, \mathbf{d}_1, \dots$. Each is associated with a different per-iteration cost, and different convergence properties. One well known method is setting $\mathbf{d}_i = \mathbf{e}^{((i \mod n)+1)}$. In that case, every n iterations corresponds to a single iteration of Gauss-Seidel (recall that we assume that the matrix has unit diagonal).

Leventhal and Lewis suggested using random directions instead of deterministic ones: $\mathbf{d}_0, \mathbf{d}_1, \dots$ are

Algorithm 1 Randomized Gauss-Seidel

1: **Input:** $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$, (pointer to) vector $\mathbf{x}$ (initial approximation and algorithm output) .
2:
3: **for** $j = 1, 2, \ldots$ **do**
4:     Pick a random r uniformly over $\{1, \ldots, n\}$
5:     Read the entries of $\mathbf{x}$ corresponding to non-zero entries in $\mathbf{A}_r$
6:     Using these entries, compute $\gamma \leftarrow (\mathbf{b})_r - \mathbf{A}_r \mathbf{x}$
7:     Update: $(\mathbf{x})_r \leftarrow (\mathbf{x})_r + \gamma$
8: **end for**

---

i.i.d. random vectors, taking $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(n)}$ with equal probability[1]. With this distribution of direction vectors, they proved the following bound on the expected error in the $\mathbf{A}$-norm [8]:

$$E_m \leq \left(1 - \frac{\lambda_{\min}}{n}\right)^m \|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{A}}^2 .$$

So, the randomized Gauss-Seidel iteration converges in expectation at a linear rate[2]. Note that the expected cost per iteration of randomized Gauss-Seidel is $\Theta(\mathbf{nnz}\,(\mathbf{A})\,/n)$, so n iterations are about as costly has a single Gauss-Seidel iteration. The proof of the last bound relies on the following lemma, which we use extensively in our analysis as well[3]:

Lemma 2.1. *Let $\mathbf{d}$ be a random vector taking $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(n)}$ with equal probability. Suppose that $\mathbf{x}$ and $\mathbf{d}$ are independent. Then,*

$$\frac{\lambda_{\min}}{n}\mathbb{E}\left[\|\mathbf{x} - \mathbf{x}^\star\|_{\mathbf{A}}^2\right] \leq \mathbb{E}\left[(\mathbf{x} - \mathbf{x}^\star, \mathbf{d})_{\mathbf{A}}^2\right] \leq \frac{\lambda_{\max}}{n}\mathbb{E}\left[\|\mathbf{x} - \mathbf{x}^\star\|_{\mathbf{A}}^2\right] .$$

3   Asynchronous Randomized Gauss-Seidel

Algorithm 1 contains a pseudo-code description of randomized Gauss-Seidel in which we made the read and update operations explicit. This obviously entails some details that are, in a sense, implementation specific. There are implementations of the randomized Gauss-Seidel iteration which do not match the description in Algorithm 1.

Consider a shared memory model with P processors. Each processor follows Algorithm 1 using the same $\mathbf{x}$, i.e. all processors read and update the same $\mathbf{x}$, which is stored in shared memory. The processors do not explicitly coordinate or synchronize their iterations. We do, however, impose assumptions, some of which may require enforcement in an actual implementation. The first assumption is rather simple: the update operation in each iteration is atomic.

---

[1]Leventhal and Lewis consider the more general setting where $\mathbf{A}$ does not have unit diagonal. For that case, they analyze non-uniform probabilities. When the matrix has unit diagonal, their algorithm and the convergence analysis reduces to the ones stated here.

[2]Some care should be employed with terminology. Some mathematicians or computer scientists might say this is an *exponential* or *geometric convergence rate.* However, numerical analysts refer to this rate as *linear,* as it is linear in $O(\log(\epsilon))$ where ϵ is the desired reduction factor of the error.

[3]The upper bound in the following lemma was not proven by Leventhal and Lewis [8], but it can be proved using the same technique they used to prove the lower bound. For completeness we include a proof in the appendix.

Assumption 3.1 (Atomic Write). *The update operation in line 7 is atomic.*

The update operation operates on a single coordinate in $\mathbf{x}$. For single precision or double precision floating point reals, updates of the form used in line 7 have hardware support on many modern processors (e.g. compare-and-exchange on recent Intel processors).

If atomic write is enforced, then for the sake of the analysis we can impose an order $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ on the values that $\mathbf{x}$ takes during the computation. Here $\mathbf{x}_j$ denotes the value of $\mathbf{x}$ after j updates have been applied (breaking ties in an arbitrary manner).

We now turn our attention to the read operation in line 5. Here we consider two possible models. In the first model, we assume the following consistent read assumption is enforced.

Assumption 3.2 (Consistent Read). *The values of the entries of $\mathbf{x}$ read in line 5 appeared together in $\mathbf{x}$ at some time before the update operation (line 7) is executed.*

Note that assumption 3.2 does not imply that none of the entries read during the execution of line 5 are modified while that line is being executed, but the opposite does hold (and this is one way of enforcing this assumption).

With consistent read we can denote by $k(j) \leq j$ the maximum iteration index such $\mathbf{x}_{k(j)}$ is equal to the values read on line 5, on the indexes read during the execution of line 5. The existence of such a $k(j)$ is guaranteed by assumption 3.2 (since all writes are atomic, all time intervals correspond to some iteration index). The iteration can then be written:

$$\begin{aligned}
\gamma_j &= (\mathbf{x}^\star - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}} \\
\mathbf{x}_{j+1} &= \mathbf{x}_j + \gamma_j \mathbf{d}_j .
\end{aligned}$$

(2)

We also consider a model where we allow inconsistent reads. Since every iteration changes a single coordinate, and we require all writes to be atomic, the value of $\mathbf{x}$ read in line 5 is the result of a subset of the updates that occurred before the write operation in line 7 is executed. Let us denote by $K(j) \subseteq \{0, 1, \ldots, j-1\}$ a maximal set of updates consistent with the computation of γ in iteration j (a formal definition of $K(j)$ is as follow: an update index i is in $K(j)$ if either it updates an entry of $\mathbf{x}$ not read for computing γ_j, or it updates an entry and the update was applied before that entry was read). The vector read is then

$$\mathbf{x}_{K(j)} = \mathbf{x}_0 + \sum_{i \in K(j)} \gamma_i \mathbf{d}_i .$$

The iteration can then be written as

$$\begin{aligned}
\gamma_j &= (\mathbf{x}^\star - \mathbf{x}_{K(j)}, \mathbf{d}_j)_{\mathbf{A}} \\
\mathbf{x}_{j+1} &= \mathbf{x}_j + \gamma_j \mathbf{d}_j .
\end{aligned}$$

Obviously, enforcing consistent read involves some overhead. However the bounds we obtain for the inconsistent read model are not as good as the ones achieved when we assume consistent reads. There is clearly a trade-off here, which we present but do not attempt to quantify. It is a complex trade-off that depends on many factors, including possible hardware features like transactional memory that may enable efficient enforcement of consistent reads. We do however note that in many cases even *without* any special provisions, the probability of an inconsistent read in a single iteration is extremely small, so much that we

4

do not expect it to happen much (or at all) in a normal execution of the algorithm. The reason is that in order to have an inconsistent read in a single iteration there has to be at least two updates to entries read during that iteration. Consider again the reference scenario. Each iteration reads at most $C_2 \ll n$ entries. Suppose there are u updates while reading those entries. Each such update affects a single random entry. Therefore, the probability that it will update one of the C_2 entries being read is at most C_2/n. The probability of getting two such updates is bounded by the probability of getting at least two in a binomial distribution with u experiments and probability C_2/n. Unless u is very large, this is an extremely small probability (since C_2/n is tiny).

We are mainly interested in algorithms with provable convergence rates. In a totally asynchronous model with arbitrary delays, there can also be an arbitrary delay in convergence. Therefore, we assume that asynchronism is bounded in the sense that delays are bounded.

Assumption 3.3 (Bounded Asynchronism). *There is a constant τ (measure of asynchronism) such that all updates that are older than τ iterations participate in the computation of iteration j, for all iterations $j = 1, 2, \ldots$.*

In the consistent read model, this assumption translates to requiring that

$$j - \tau \leq k(j) \leq j. \tag{3}$$

In the inconsistent read model, this assumption translates to requiring that

$$\{0, 1, \ldots, \max\{0, j - \tau - 1\}\} \subseteq K(j). \tag{4}$$

When the the variance in the number of non-zeros per row is not too large (with respect to the mean), as is the case in our reference scenario, then the time spent per iteration is roughly uniform. Therefore, in that case we expect τ to be of order of P.

We now discuss the relation between $k(0), k(1), \ldots$ or $K(0), K(1), \ldots$ and the random variables $\mathbf{d}_0, \mathbf{d}_1, \ldots$. If we inspect the pseudo-code of Algorithm 1 closely we will realize that $k(j)$ or $K(j)$ (depending on the model) depend on the random choices $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{j-1}$ made before the write operation, and more crucially on the random choice $\mathbf{d}_j$. The reason is that on line 5 we read only the relevant entries of $\mathbf{x}$, so only a small set of updates can be considered for inclusion, and this set of relevant entries is determined by the selection of $\mathbf{d}_j$. However, a completely adversarial model which allows dependence of $k(j)$ (or $K(j)$) on $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_j$ (for $j = 1, 2, \ldots$) and analyzes the worst-case behavior is not likely to be very faithful to the actual behavior of the algorithm. Therefore, we assume the delays are independent of the random choices, but allow them to be arbitrary (as long as the bounded asynchronism assumption holds).

Assumption 3.4 (Independent Delays). *We allow an arbitrary set of delays that satisfy (3) or (4) (depending on the context), but they do not depend on the random choices $\mathbf{d}_0, \mathbf{d}_1, \ldots$.*

4   Convergence Bound With Consistent Read

In this section we analyze iteration (2), which corresponds to the consistent read model (which assumes both assumptions 3.1 and 3.2), under assumptions 3.3 and 3.4. We first state and prove the results, and then discuss them.

Theorem 4.1. *Consider iteration* (2) *for an arbitrary starting vector* $\mathbf{x}_0$, *where* $\mathbf{d}_0, \mathbf{d}_1, \dots$ *are i.i.d. vectors that take* $\mathbf{e}^{(1)}, \dots, \mathbf{e}^{(n)}$ *with equal probability, and* $k(0), k(1), \dots$ *are such that* (3) *holds but are independent of the random choices of* $\mathbf{d}_0, \mathbf{d}_1, \dots$. *Let* $\rho = \max_l \left\{ \frac{1}{n} \sum_{r=1}^{n} |\mathbf{A}_{lr}| \right\}$. *Provided that* $2\rho\tau < 1$, *the following holds:*

(a) For every $m \geq \frac{\log(1/2)}{\log(1 - \lambda_{\max}/n)} \approx \frac{0.693n}{\lambda_{\max}}$ *we have*

$$E_m \leq \left(1 - \frac{(1 - 2\rho\tau)}{2\kappa} \right) E_0 .$$

(b) Let $T_0 = \left\lceil \frac{\log(1/2)}{\log(1 - \lambda_{\max}/n)} \right\rceil$ *and* $T = T_0 + \tau$. *For every* $m \geq rT$ $(r = 1, 2, \dots)$ *we have*

$$E_m \leq (1 - \frac{(1 - 2\rho\tau)}{2\kappa})(1 - \frac{(1 - 2\rho\tau)(1 - \lambda_{\max}/n)^\tau}{2\kappa} + \chi)^{r-1} E_0$$

where

$$\chi = \frac{\rho\tau^2 \lambda_{\max}(1 - \lambda_{\max}/n)^{-2\tau}}{n} .$$

Proof. In the proof we use the following abbreviations:

$$\nu_\tau = 1 - 2\rho\tau \qquad \delta_{\min} = \frac{\nu_\tau \lambda_{\min}}{n} \qquad \delta_{\max} = 1 - \frac{\lambda_{\max}}{n} .$$

Simple algebraic manipulations show that (see appendix for details):

$$\|\mathbf{x}_{j+1} - \mathbf{x}^\star\|_{\mathbf{A}}^2 = \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 - (\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 - 2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}} . \quad (5)$$

We see that the error decreases by a "progress term" $((\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2)$ which is always positive, and changes by an additional term $(2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_{k(j)} - \mathbf{x}_j, \mathbf{d}_j)_{\mathbf{A}})$ which might be positive or negative. When the iterations are synchronized $(k(j) = j)$ there is no additional term, and the analysis reduces to the analysis of synchronous randomized Gauss-Seidel. We first bound the additional term:

$$
\begin{aligned}
2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}} &= 2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}\left(\sum_{t=k(j)}^{j-1} \gamma_t \mathbf{d}_t, \mathbf{d}_j \right)_{\mathbf{A}} \\
&= \sum_{t=k(j)}^{j-1} 2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}^\star - \mathbf{x}_{k(t)}, \mathbf{d}_t)_{\mathbf{A}}(\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}} \quad (6) \\
&\geq -\sum_{t=k(j)}^{j-1} \left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 |(\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}}| + (\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_{\mathbf{A}}^2 |(\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}}| \right] .
\end{aligned}
$$

Since $k(j) \leq t < j$:

$$\mathbb{E}\left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)^2_{\mathbf{A}} | (\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}}|\right] = \mathbb{E}\left[\mathbb{E}\left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)^2_{\mathbf{A}} | (\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}}| | \mathbf{d}_0, \ldots, \mathbf{d}_{t-1}\right]\right]$$

$$= \mathbb{E}\left[\frac{1}{n^2} \sum_{l=1}^{n} \sum_{r=1}^{n} (\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{e}^{(l)})^2_{\mathbf{A}} \left|(\mathbf{e}^{(l)}, \mathbf{e}^{(r)})_{\mathbf{A}}\right|\right]$$

$$= \mathbb{E}\left[\frac{1}{n^2} \sum_{l=1}^{n} \sum_{r=1}^{n} (\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{e}^{(l)})^2_{\mathbf{A}} |\mathbf{A}_{lr}|\right]$$

$$\leq \rho \mathbb{E}\left[\frac{1}{n} \sum_{l=1}^{n} (\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{e}^{(l)})^2_{\mathbf{A}}\right] = \rho \mathbb{E}\left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)^2_{\mathbf{A}}\right].$$

Similarly, $\mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)^2_{\mathbf{A}} | (\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}}|\right] \leq \rho \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)^2_{\mathbf{A}}\right]$. Taking expectation of (6) and applying the last inequality we find that

$$\mathbb{E}\left[2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}}\right] \geq -\rho \sum_{t=k(j)}^{j-1} \left[\mathbb{E}\left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)^2_{\mathbf{A}}\right] + \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)^2_{\mathbf{A}}\right]\right]$$

$$= -\rho |j - k(j)| \mathbb{E}\left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)^2_{\mathbf{A}}\right] - \rho \sum_{t=k(j)}^{j-1} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)^2_{\mathbf{A}}\right]$$

$$\geq -\rho\tau \mathbb{E}\left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)^2_{\mathbf{A}}\right] - \rho \sum_{t=k(j)}^{j-1} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)^2_{\mathbf{A}}\right].$$

Taking expectation of (5), and plugging in the last inequality, we find that

$$E_{j+1} \leq E_j - (1 - \rho\tau)\mathbb{E}\left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)^2_{\mathbf{A}}\right] + \rho \sum_{t=k(j)}^{j-1} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)^2_{\mathbf{A}}\right]. \tag{7}$$

Unrolling the recursion, we find that for every m:

$$E_m \leq E_0 - (1 - \rho\tau) \sum_{i=0}^{m-1} \mathbb{E}\left[(\mathbf{x}_{k(i)} - \mathbf{x}^\star, \mathbf{d}_i)^2_{\mathbf{A}}\right] + \rho \sum_{i=0}^{m-1} \sum_{t=k(i)}^{i-1} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)^2_{\mathbf{A}}\right].$$

In the last sum of the previous inequality $(\rho \sum_{i=0}^{m-1} \sum_{t=k(i)}^{i-1} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)^2_{\mathbf{A}}\right])$, each term of the form $\mathbb{E}\left[(\mathbf{x}_{k(r)} - \mathbf{x}^\star, \mathbf{d}_r)^2_{\mathbf{A}}\right]$ appears at most τ times, each time with a coefficient ρ, so

$$E_m \leq E_0 - (1 - 2\rho\tau) \sum_{i=0}^{m-1} \mathbb{E}\left[(\mathbf{x}_{k(i)} - \mathbf{x}^\star, \mathbf{d}_i)^2_{\mathbf{A}}\right].$$

We now apply the bound $\mathbb{E}\left[(\mathbf{x}_{k(i)} - \mathbf{x}^\star, \mathbf{d}_i)^2_{\mathbf{A}}\right] \geq (\lambda_{\min}/n)E_{k(i)}$ (Lemma 2.1), to find that

$$E_m \leq E_0 - \delta_{\min} \sum_{i=0}^{m-1} E_{k(i)}. \tag{8}$$

Proof of (a). Lemma 2.1 implies that for any $b \geq a$ we have $E_b \geq \delta_{\max}^{b-a} E_a$ (see proof in Appendix), so in particular since $i \geq k(i)$,

$$E_{k(i)} \geq \delta_{\max}^{k(i)} E_0 \geq \delta_{\max}^i E_0 . \tag{9}$$

Plugging (9) into (8) we get the following inequality, which leads immediately to assertion (a):

$$E_m \leq \left(1 - \delta_{\min} \sum_{i=0}^{m-1} \delta_{\max}^i\right) E_0 = \left(1 - \frac{\delta_{\min}(1 - \delta_{\max}^m)}{1 - \delta_{\max}}\right) E_0 = (1 - \nu_\tau \kappa^{-1}(1 - \delta_{\max}^m)) E_0 .$$

Proof of (b). Let

$$C_i = \{rT + i - \tau \leq t \leq rT + i - 1 : t \geq rT\}$$

and

$$D_i = \{rT + i - \tau \leq t \leq rT + i - 1 : t < rT\} .$$

Unrolling the recursion in equation (7) starting at rT, we find that for $r \geq 1$ and $w \geq 0$

$$
\begin{aligned}
E_{(r+1)T+w} \;\leq\; & E_{rT} - (1 - \rho\tau) \sum_{i=0}^{T-1+w} \mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^\star, \mathbf{d}_{rT+i})_\mathbf{A}^2\right] + \rho \sum_{i=0}^{T-1+w} \sum_{t=k(rT+i)}^{rT+i-1} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right] \\
\leq\; & E_{rT} - (1 - \rho\tau) \sum_{i=0}^{T-1+w} \mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^\star, \mathbf{d}_{rT+i})_\mathbf{A}^2\right] + \rho \sum_{i=0}^{T-1+w} \sum_{t=rT+i-\tau}^{rT+i-1} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right] \\
\leq\; & E_{rT} - (1 - \rho\tau) \sum_{i=0}^{T-1+w} \mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^\star, \mathbf{d}_{rT+i})_\mathbf{A}^2\right] + \rho \sum_{i=0}^{T-1+w} \sum_{t \in C_i} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right] \\
& + \rho \sum_{i=0}^{T-1+w} \sum_{t \in D_i} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right] \\
\leq\; & E_{rT} - (1 - 2\rho\tau) \sum_{i=0}^{T-1+w} \mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^\star, \mathbf{d}_{rT+i})_\mathbf{A}^2\right] + \rho \sum_{i=0}^{\tau-1} \sum_{t \in D_i} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right] . \\
\leq\; & E_{rT} - (1 - 2\rho\tau) \sum_{i=\tau}^{T-1+w} \mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^\star, \mathbf{d}_{rT+i})_\mathbf{A}^2\right] + \rho \sum_{i=0}^{\tau-1} \sum_{t \in D_i} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right] . \tag{10}
\end{aligned}
$$

The second-to-last inequality follows from the fact that each term of the form $\mathbb{E}\left[(\mathbf{x}_{k(l)} - \mathbf{x}^\star, \mathbf{d}_l)_\mathbf{A}^2\right]$ appears at most τ times in $\rho \sum_{i=0}^{T-1+w} \sum_{t \in C_i} \mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right]$. We also use the fact that for $i \geq \tau$ we trivially have $D_i = \emptyset$.

We first bound $E_{rT} - (1 - 2\rho\tau) \sum_{i=\tau}^{T-1+w} \mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^\star, \mathbf{d}_{rT+i})_\mathbf{A}^2\right]$. Using Lemma 2.1,

$$E_{rT} - (1 - 2\rho\tau) \sum_{i=\tau}^{T-1+w} \mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^\star, \mathbf{d}_{rT+i})_\mathbf{A}^2\right] \leq E_{rT} - \delta_{\min} \sum_{i=\tau}^{T-1+w} E_{k(rT+i)} .$$

Since $i \geq \tau$ we have $k(kT + i) \geq kT$ so $E_{k(rT+i)} \geq \delta_{\max}^{k(rT+i)-rT} E_{rT} \geq \delta_{\max}^i E_{rT}$. Therefore

$$E_{rT} - (1 - 2\rho\tau) \sum_{i=\tau}^{T-1+w} \mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^\star, \mathbf{d}_{rT+i})_\mathbf{A}^2\right] \leq (1 - \delta_{\min}\delta_{\max}^\tau \sum_{i=0}^{T-1+w-\tau} \delta_{\max}^i) E_{rT} .$$

8

Noticing that $T - 1 + w - \tau = T_0 - 1 + w$ and bounding the geometric sum as in assertion (a), we find that $(1 - \delta_{\min}\delta_{\max}^{\tau}\sum_{i=0}^{T-1+w-\tau}\delta_{\max}^{i}) \leq (1 - \frac{\delta_{\max}^{\tau}\nu_{\tau}}{2\kappa})$, so

$$E_{rT} - (1 - 2\rho\tau)\sum_{i=\tau}^{T-1+w}\mathbb{E}\left[(\mathbf{x}_{k(rT+i)} - \mathbf{x}^{\star}, \mathbf{d}_{rT+i})_{\mathbf{A}}^2\right] \leq (1 - \frac{\delta_{\max}^{\tau}\nu_{\tau}}{2\kappa})E_{rT}. \qquad (11)$$

We now bound $\rho\sum_{i=0}^{\tau-1}\sum_{t\in D_i}\mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^{\star}, \mathbf{d}_t)_{\mathbf{A}}^2\right]$. Recall that for every $b \geq a$ we have $E_b \geq \delta_{\max}^{b-a}E_a$, so, for $i = 0, \ldots, \tau - 1$ and $t \in D_i$ we have

$$E_{k(t)} \leq \delta_{\max}^{k(t)-rT}E_{rT} \leq \delta_{\max}^{-2\tau}E_{rT}.$$

The last inequality follows from the fact that for $t \in D_i$, $k(t) - rT \geq -2\tau$ and $\delta_{\max} < 1$. We now bound

$$\rho\sum_{i=0}^{\tau-1}\sum_{t\in D_i}\mathbb{E}\left[(\mathbf{x}_{k(t)} - \mathbf{x}^{\star}, \mathbf{d}_t)_{\mathbf{A}}^2\right] \leq \rho\sum_{i=0}^{\tau-1}\sum_{t\in D_i}\frac{\lambda_{\max}\delta_{\max}^{-2\tau}}{n}E_{rT} \leq \frac{\rho\tau^2\lambda_{\max}\delta_{\max}^{-2\tau}}{n}E_{rT} = \chi E_{rT}.$$

Combine the last inequality with (11) and assertion (a) to complete the proof of assertion (b). □

Discussion:

- Assertion (a) shows that after we perform enough asynchronous iterations, we are guaranteed to reduce the expected error by a constant factor. In order to drive the expected error down to an arbitrary fraction of the input error, we can adopt the following scheme. We start with asynchronous iterations. After n iterations have been completed we synchronize the threads and restart the iterations. The matrix $\mathbf{A}$ has unit diagonal, so $\lambda_{\max} \geq 1$. Therefore, by performing $k \geq n$ iterations, we are guaranteeing a $1 - \nu_{\tau}/2\kappa$ factor reduction in the expected error. We then continue to iterate and synchronize until the expected error is guaranteed to be small enough. The number of outer iterations until convergence (reduction of error by a predetermined factor) is $O(\kappa/\nu_{\tau})$. This is also the number of synchronization points. When ν_{τ} is close to one, the number of synchronization points is asymptotically the same as in Jacobi, but the convergence rate is that of Gauss-Seidel. Furthermore, we do not need to really divide the iterations between processors (basically, every processor can do as many iterations as it can, until synchronization) and it is not important to synchronize exactly after n iterations. So, from a practical perspective, a time based scheme for synchronizing the processors should be sufficient, and will not suffer from large wait times due to load imbalance.

- Assertion (b) shows that even if we do not occasionally synchronize the threads, we still get long-term linear convergence, but at a slower rate. We say convergence is linear in the long term since we cannot guarantee a diminishing bound in every iteration, but we can prove a constant factor reduction over a large enough amount of iterations.

- The terms $\delta_{\max}^{\tau}$ and $\delta_{\max}^{-2\tau}$ that appear in assertion (b) might seem problematic as they are exponential in the number of processors (because $\tau = \Omega(P)$). However, in our reference scenario this is not an issue since in that scenario $\lambda_{\max} = O(1)$ and P and τ are much smaller than n, so $\delta_{\max}^{\tau}$ and $\delta_{\max}^{-2\tau}$ are actually very close to 1.

- The number of iterations to guarantee a $1 - \nu_\tau/2\kappa$ reduction of expected error (as in assertion (a)) in synchronous randomized Gauss-Seidel is approximately $\nu_\tau n/2\lambda_{\max}$.

- Consider our reference scenario in a weak-scaling regime (i.e., $P \approx cn$ for a very small c). In that case ν_τ is constant and close to one, so if we occasionally synchronize the threads we have a constant percent increase in the number of iterations due to asynchronism. That is, the asynchronous phases do not violate the weak-scaling (but the number of iterations can increase due to $\lambda_{\min}$ becoming smaller). As for the case where only asynchronous iterations are used, we have $\chi \approx c^2\lambda_{\max}^2$. So, χ itself exhibits weak scaling. However, its value should be interpreted with respect to κ^{-1}. If $\lambda_{\min}$ shrinks as n grows, as is the case in many applications, then the relative size of χ grows and we do not have weak scaling.

5 Improving Scalability By Controlling Step-Size

The bound in Theorem 4.1 requires $2\rho\tau < 1$. In this section we show that by introducing a step-size we can have a convergent method for any delay (as long as we set the step size small enough). By optimizing the step-size we can also improve the scaling (dependence on τ) in our bounds.

The idea in introducing a step-size is that instead of taking a full step, we take a partial step by multiplying it by a step-size β. That is, we now consider the iteration

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \beta\gamma_j\mathbf{d}_j$$

(γ_j is defined as before). Again, simple algebraic manipulations show that (see appendix for details):

$$\|\mathbf{x}_{j+1} - \mathbf{x}^\star\|_{\mathbf{A}}^2 = \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 - \beta(2-\beta)(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 - 2\beta(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}}. \quad (12)$$

As before, we continue with bounding the additional term.

$$
\begin{aligned}
2\beta(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}} &= 2\beta(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}\left(\sum_{t=k(j)}^{j-1}\beta\gamma_t\mathbf{d}_t, \mathbf{d}_j\right)_{\mathbf{A}} \quad (13) \\
&= \beta^2\sum_{t=k(j)}^{j-1} 2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}^\star - \mathbf{x}_{k(t)}, \mathbf{d}_t)_{\mathbf{A}}(\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}} \\
&\geq -\beta^2\sum_{t=k(j)}^{j-1}\left[(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2|(\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}}| + (\mathbf{x}_{k(t)} - \mathbf{x}^\star, \mathbf{d}_t)_{\mathbf{A}}^2|(\mathbf{d}_t, \mathbf{d}_j)_{\mathbf{A}}|\right].
\end{aligned}
$$

We see that the progress term is $O(\beta)$, but the additional term is $O(\beta^2)$. In synchronous randomized Gauss-Seidel the best bound on the expected error is achieved with $\beta = 1$, but for an asynchronous computation the best bound is achieved with some $\beta < 1$ (depending on τ). Continuing along the lines of the proof of Theorem 4.1 (we omit the details), we find the following modified bounds:

$$\text{(a)}:\ E_m \leq \left(1 - \frac{\nu_\tau(\beta)}{2\kappa}\right)E_0, \qquad \text{(b)}:\ E_m \leq \left(1 - \frac{\nu_\tau(\beta)}{2\kappa}\right)\left(1 - \frac{\nu_\tau(\beta)\delta_{\max}^\tau}{2\kappa} + \chi(\beta)\right)^{r-1}E_0$$

where

$$\nu_\tau(\beta) = 2\beta - \beta^2 - 2\rho\tau\beta^2 \qquad \chi(\beta) = \frac{\rho\tau^2\beta^2\lambda_{\max}\delta_{\max}^{-2\tau}}{n}.$$

Discussion:

- We see that for a sufficiently small β both bounds are useful, but the computation of the optimal β for assertion (b) (in terms of the bound) requires some approximation of the condition number.

- Alternatively, we can optimize only the value of $\nu_\tau(\beta)$. The optimum of that term is achieved at $\tilde{\beta} = 1/(1 + 2\rho\tau)$ and yields $\nu_\tau(\tilde{\beta}) = 1/(1 + 2\rho\tau)$. It is also the case that $\chi(\tilde{\beta}) < \chi(1)$, so both bounds are improved. From a practical perspective, the challenge of setting the step size to $\tilde{\beta}$ is that τ might not be known. However, under normal circumstances (and in the reference scenario) we have $\tau = O(P)$, which can provide a general guideline for setting the step-size.

6 Convergence Bound With Inconsistent Reads

We now analyze a model without consistent read (without assumption 3.2). To show convergence, we must use a step size, so we analyze the following iteration:

$$\gamma_j = (\mathbf{x}^\star - \mathbf{x}_{K(j)}, \mathbf{d}_j)_\mathbf{A}$$
$$\mathbf{x}_{j+1} = \mathbf{x}_j + \beta\gamma_j\mathbf{d}_j .$$

(14)

The rate of convergence is slower, and the scalability is worse as the dependence on τ is worse.

Theorem 6.1. *Consider iteration* (14) *for some* $0 \le \beta < 1$ *and an arbitrary starting vector* $\mathbf{x}_0$, *where* $\mathbf{d}_0, \mathbf{d}_1, \ldots$ *are i.i.d. vectors that take* $\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(n)}$ *with equal probability, and* $K(0), K(1), \ldots$ *are such that equation* (4) *holds but are independent of the random choices of* $\mathbf{d}_0, \mathbf{d}_1, \ldots$. *Let* $\rho_2 = \max_l \left\{ \frac{1}{n} \sum_{r=1}^n \mathbf{A}_{lr}^2 \right\}$. *Provided that* $2\beta(1 - \beta - \rho_2\tau^2\beta/2) < 1$, *the following holds:*

(a) For every $m \ge \frac{\log(1/2)}{\log(1-\lambda_{\max}/n)} \approx \frac{0.693n}{\lambda_{\max}}$ we have

$$E_m \le \left(1 - \frac{\beta(1 - \beta - \rho_2\tau^2\beta/2)}{\kappa} \right) E_0 .$$

(b) Let $T_0 = \left\lceil \frac{\log(1/2)}{\log(1-\lambda_{\max}/n)} \right\rceil$ and $T = T_0 + \tau$. For every $m \ge rT$ ($r = 1, 2, \ldots$) we have

$$E_m \le (1 - \frac{2\beta(1 - \beta - \rho_2\tau^2\beta/2)}{2\kappa})(1 - \frac{2\beta(1 - \beta - \rho_2\tau^2\beta/2)(1 - \lambda_{\max}/n)^\tau}{2\kappa} + \chi)^{r-1} E_0$$

where

$$\chi = \frac{\rho_2\tau^3\beta^2\lambda_{\max}(1 - \lambda_{\max}/n)^{-2\tau}}{n} .$$

Most of the proof is analogous to the proof of Theorem 4.1, so we give only a sketch of the proof that focuses on the unique parts. **Proof.** *(Sketch)* As before:

$$\|\mathbf{x}_{j+1} - \mathbf{x}^\star\|_\mathbf{A}^2 = \|\mathbf{x}_j - \mathbf{x}^\star\|_\mathbf{A}^2 - \beta(2 - \beta)(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}^2 - 2\beta(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}(\mathbf{x}_j - \mathbf{x}_{K(j)}, \mathbf{d}_j)_\mathbf{A} .$$

We now bound the additional term:

$$
\begin{aligned}
2\beta(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}(\mathbf{x}_j - \mathbf{x}_{K(j)}, \mathbf{d}_j)_\mathbf{A} &= 2\beta(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}(\sum_{t \in K^-(j)} \beta\gamma_t \mathbf{d}_t, \mathbf{d}_j)_\mathbf{A} \\
&= 2\beta^2(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}(\sum_{t \in K^-(j)} \gamma_t \mathbf{d}_t, \mathbf{d}_j)_\mathbf{A} \\
&\geq -\beta^2\left[(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}^2 + (\sum_{t \in K^-(j)} \gamma_t \mathbf{d}_t, \mathbf{d}_j)_A^2\right] \\
&\geq -\beta^2\left[(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}^2 + \right. \\
&\quad \left. |K^-(j)| \sum_{t \in K^-(j)} (\mathbf{x}_{K(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2(\mathbf{d}_t, \mathbf{d}_j)_\mathbf{A}^2\right] \\
&\geq -\beta^2\left[(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}^2 + \tau \sum_{t \in K^-(j)} (\mathbf{x}_{K(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2(\mathbf{d}_t, \mathbf{d}_j)_\mathbf{A}^2\right]
\end{aligned}
\tag{15}
$$

where $K^-(j) = \{0, \dots, j-1\} - K(j)$. Since $\mathbf{x}_{K(t)}$ does not depend on $\mathbf{d}_t$ or $\mathbf{d}_j$, we can bound as before,

$$
\mathbb{E}\left[(\mathbf{x}_{K(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2(\mathbf{d}_t, \mathbf{d}_j)_\mathbf{A}^2\right] \leq \rho_2 \mathbb{E}\left[(\mathbf{x}_{K(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right].
$$

Therefore,

$$
E_{j+1} \leq E_j - 2\beta(1 - \beta)\mathbb{E}\left[(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}^2\right] + \rho_2\tau\beta^2 \sum_{t \in K^-(j)} \mathbb{E}\left[(\mathbf{x}_{K(t)} - \mathbf{x}^\star, \mathbf{d}_t)_\mathbf{A}^2\right].
$$

After we unroll the recursion, we find that

$$
E_k \leq E_0 - 2\beta(1 - \beta - \rho_2\tau^2\beta/2)\sum_{i=0}^{k-1} \mathbb{E}\left[(\mathbf{x}_{K(i)} - \mathbf{x}^\star, \mathbf{d}_i)_\mathbf{A}^2\right].
$$

We can now continue to bound as in the proof of Theorem 4.1. The crucial observation is that $\mathbf{x}_{K(i)}$ is the result of $|K(i)|$ random single coordinate steps, so $\mathbb{E}\left[\|\mathbf{x}_{K(i)} - \mathbf{x}^\star\|_\mathbf{A}^2\right] \geq \delta_{\max}^{|K(i)|}E_0 \geq \delta_{\max}^i E_0$. $\square$

Remark 6.2. One might ask why we did not simply adapt equation (13) for the inconsistent read iteration, and instead developed equation (15). We could certainly adapt equation (13). The problem is that if we follow that path, we get expressions of the form $(\mathbf{x}_{K(j)} - \mathbf{x}^\star, \mathbf{d}_j)_\mathbf{A}^2 |(\mathbf{d}_t, \mathbf{d}_j)_\mathbf{A}|$ for $t \in K^-(j)$. This expression is hard to analyze since $\mathbf{x}_{K(j)}$ can depend of $\mathbf{d}_t$. An example is $K(j) = \{0, \dots, j-3, j-1\}$ and $t = j - 2$ (for some $j \geq 3$).

7   Related Work and Concluding Remarks

Asynchronous methods were first suggested by Chazan and Miranker [4] in their pioneering paper on chaotic relaxation. The theory and application of asynchronous iterations has since been studied and used by many

authors. Noteworthy is the seminal text by Bertsekas and Tsitsiklis [2]. A more recent review is by Frommer and Szyld [6].

Historically, work on asynchronous methods focused on proving that the methods converge in the limit, and not on convergence rate analysis. In particular, the relation to the convergence rate of synchronous counterparts, and the scaling of these methods, were seldom studied. We are aware of only two exceptions, but the results are quite unsatisfactory. Baudet [1] proves a theorem that can be used to analyze the rate of convergence of asynchronous iterations of contracting operators, so they are applicable to the solution of only certain types of linear systems. In addition, his theorem can be used to analyze the convergence of a specific instantiation, and not the worst case behavior of an asynchronous method. Bertsekas and Tsitsiklis [2, Section 7.2, Exercise 1.2] prove a linear convergence rate of certain asynchronous iterations for some classes of matrices (like weakly diagonally dominant matrices), but analyze how the rate of convergence depends on the measure of asynchronism only under very restrictive conditions and in a hard to interpret manner [2, Section 6.3.5].

Randomization is frequently used as an algorithmic tool in non-numerical asynchronous algorithms. However, until recently, it was not used as an algorithmic tool for asynchronous numerical computation (although, there is work on using asynchronous computation for inherently randomized methods like stochastic gradient descent [2]). Recently, two new articles suggested asynchronous algorithms whose provable convergence rates rely on randomization. Freris and Zouzias [5] suggested the use of an asynchronous variant of randomized Kaczmarz [10] to synchronize clocks in a wireless network. They analyze the convergence rate in a semi-asynchronous model that is suitable for wireless networks, but not for shared-memory numerical computations. Niu et al. [9] recently proposed and analyzed "Hogwild!", an asynchronous approach for parallelizing stochastic gradient descent.

Our work is very much inspired by "Hogwild!", but we study a different problem and our proof technique is different. In fact, requiring atomic writes and consistent reads and consequently writing the iteration as equation (2) is a direct descendant of Niu et al.'s [9] analysis. However, our analysis makes several significant improvement over simply applying the "Hogwild!" analysis to the solution of linear systems (linear systems can be solved using stochastic gradient descent):

1. We treat parallelism in a much more principled manner; we explicitly lay out the assumptions made by the algorithm, and study different read models. In particular, we analyze both a consistent and an inconsistent read model, while Niu et al. analyze only a consistent read model.

2. We prove a linear convergence rate, vs. a sub-linear convergence rate for "Hogwild!".

3. The analysis of "Hogwild!" assumes bounded stochastic gradients. This assumption is not realistic when stochastic gradient descent is used to solve symmetric positive definite systems.

4. Our dependence on τ is much better: "Hogwild!"'s analysis has a $O(\tau^4)$ dependence.

There are several questions that remain unanswered and are worthy of future work. Is that gap in the bound for consistent and inconsistent reads inherent, or an improved analysis will remove or narrow it? Is it possible to obtain comparable bounds when we allow $k(j)$ or $K(j)$ to depend on $\mathbf{d}_0, \ldots, \mathbf{d}_j$? In our reference scenario, we show weak-scaling only if we periodically synchronize the threads. Is the periodic synchronization essential, or is it an artifact of the analysis?

Acknowledgments

References

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

[2] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation*. Prentice Hall, 1989.

[3] 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. MIMS EPrint 2012.62, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, June 2012.

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

[5] Nikolaos M. Freris and Anastasios Zouzias. Fast distributed smoothing for network clock synchronization. In *IEEE Conference on Decision and Control*, 2012.

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

[7] Michael Griebel and Peter Oswald. Greedy and randomized versions of the multiplicative Schwarz method. *Linear Algebra and its Applications*, 437(7):1596 – 1610, 2012.

[8] D. Leventhal and A. S. Lewis. Randomized methods for linear constraints: Convergence rates and conditioning. *Math. Oper. Res.*, 35(3):641–654, 2010.

[9] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems (NIPS) 24*, pages 693–701, 2011.

[10] Thomas Strohmer and Roman Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15:262–278, 2009. 10.1007/s00041-008-9030-4.

8 Appendix

8.1 Proof of Lemma 2.1

Let $\mathbf{B}$ be the unique symmetric positive matrix such that $\mathbf{A} = \mathbf{B}^2$.

$$
\begin{aligned}
\mathbb{E}\left[(\mathbf{x} - \mathbf{x}^\star, \mathbf{d})_{\mathbf{A}}^2\right] &= \mathbb{E}\left[\mathbb{E}\left[(\mathbf{x} - \mathbf{x}^\star, \mathbf{d})_{\mathbf{A}}^2\right] \big| \mathbf{x}\right] \\
&= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x} - \mathbf{x}^\star, \mathbf{e}_i)_{\mathbf{A}}^2\right] \\
&= \frac{1}{n}\mathbb{E}\left[\|\mathbf{A}(\mathbf{x} - \mathbf{x}^\star)\|_2^2\right] \\
&= \frac{1}{n}\mathbb{E}\left[(\mathbf{x} - \mathbf{x}^\star)^{\mathsf{T}}\mathbf{A}^2(\mathbf{x} - \mathbf{x}^\star)\right] \\
&= \frac{1}{n}\mathbb{E}\left[\frac{(\mathbf{x} - \mathbf{x}^\star)^{\mathsf{T}}\mathbf{B}\mathbf{A}\mathbf{B}(\mathbf{x} - \mathbf{x}^\star)}{(\mathbf{x} - \mathbf{x}^\star)^{\mathsf{T}}\mathbf{B}\mathbf{B}(\mathbf{x} - \mathbf{x}^\star)} \cdot (\mathbf{x} - \mathbf{x}^\star)^{\mathsf{T}}\mathbf{B}\mathbf{B}(\mathbf{x} - \mathbf{x}^\star)\right] \\
&= \frac{1}{n}\mathbb{E}\left[\frac{(\mathbf{x} - \mathbf{x}^\star)^{\mathsf{T}}\mathbf{B}\mathbf{A}\mathbf{B}(\mathbf{x} - \mathbf{x}^\star)}{(\mathbf{x} - \mathbf{x}^\star)^{\mathsf{T}}\mathbf{B}\mathbf{B}(\mathbf{x} - \mathbf{x}^\star)} \cdot \|\mathbf{x} - \mathbf{x}^\star\|_{\mathbf{A}}^2\right] .
\end{aligned}
$$

According to the Courant-Fischer theorem, for every vector $\mathbf{y} \neq 0$ we have

$$
\lambda_{\min} \leq \frac{\mathbf{y}^{\mathsf{T}}\mathbf{A}\mathbf{y}}{\mathbf{y}^{\mathsf{T}}\mathbf{y}} \leq \lambda_{\max} .
$$

Applying the last inequality to the previous equality with $\mathbf{y} = \mathbf{B}(\mathbf{x} - \mathbf{x}^\star)$ completes the proof.

8.2 Proof of Equations (5) and (12)

We prove equation (12). Equation (5) follows by setting $\beta = 1$.

$$
\begin{aligned}
\|\mathbf{x}_{j+1} - \mathbf{x}^\star\|_{\mathbf{A}}^2 &= \|\mathbf{x}_j + \beta\gamma_j\mathbf{d}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + \|\beta\gamma_j\mathbf{d}_j\|_{\mathbf{A}}^2 + 2(\mathbf{x}_j - \mathbf{x}^\star, \beta\gamma_j\mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + \beta^2\gamma_j^2 + 2\beta\gamma_j(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + \beta^2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 - 2\beta(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + \beta^2(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 \\
&\qquad -2\beta(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}\left[(\mathbf{x}_j - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}} + (\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}\right] \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 - \beta(2 - \beta)(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 - 2\beta(\mathbf{x}_{k(j)} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}}
\end{aligned}
$$

In the above we use the fact that $\mathbf{A}$ has unit diagonal, so $(\mathbf{d}_i, \mathbf{d}_i)_{\mathbf{A}} = 1$ for all i.

8.3 Proof that $E_b \geq \delta_{\max}^{b-a} E_a$ for $b \geq a$

We prove that $E_{j+1} \geq \delta_{\max} E_j$ for the iteration involving the step size ($\mathbf{x}_{j+1} = \mathbf{x}_j + \beta\gamma_j\mathbf{d}_j$, Section 5), and that automatically implies the inequality for $b \geq a$ and for $\beta = 1$ (which is the case considered in Section 4).

First notice that

$$
\begin{aligned}
\mathbf{x}_{j+1} &= \mathbf{x}_j + \beta\gamma_j\mathbf{d}_j \\
&= \mathbf{x}_j + \beta(\mathbf{x}^\star - \mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}}\mathbf{d}_j \\
&= \mathbf{x}_j + (\beta\mathbf{x}^\star - \beta\mathbf{x}_{k(j)}, \mathbf{d}_j)_{\mathbf{A}}\mathbf{d}_j \\
&= \mathbf{x}_j + (\mathbf{x}^\star - \mathbf{y}, \mathbf{d}_j)_{\mathbf{A}}\mathbf{d}_j
\end{aligned}
$$

where $\mathbf{y} = (1-\beta)\mathbf{x}^\star + \mathbf{x}_{k(j)}$. Denote $\tilde{\gamma}_j = (\mathbf{x}^\star - \mathbf{y}, \mathbf{d}_j)_{\mathbf{A}}$. Now,

$$
\begin{aligned}
\|\mathbf{x}_{j+1} - \mathbf{x}^\star\|_{\mathbf{A}}^2 &= \|\mathbf{x}_j + \tilde{\gamma}_j\mathbf{d}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + \|\tilde{\gamma}_j\mathbf{d}_j\|_{\mathbf{A}}^2 + 2(\mathbf{x}_j - \mathbf{x}^\star, \tilde{\gamma}_j\mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + \tilde{\gamma}_j^2 + 2\tilde{\gamma}_j(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + (\mathbf{y} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 - 2(\mathbf{y} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + (\mathbf{y} - \mathbf{x}_j + \mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 - 2(\mathbf{y} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + (\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 + (\mathbf{y} - \mathbf{x}_j, \mathbf{d}_j)_{\mathbf{A}}^2 \\
&\quad + 2(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{y} - \mathbf{x}_j, \mathbf{d}_j)_{\mathbf{A}} \\
&\quad - 2(\mathbf{y} - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 + (\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 + (\mathbf{y} - \mathbf{x}_j, \mathbf{d}_j)_{\mathbf{A}}^2 \\
&\quad - 2(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}(\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}} \\
&= \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 - (\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 + (\mathbf{y} - \mathbf{x}_j, \mathbf{d}_j)_{\mathbf{A}}^2 \\
&\geq \|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2 - (\mathbf{x}_j - \mathbf{x}^\star, \mathbf{d}_j)_{\mathbf{A}}^2 .
\end{aligned}
$$

Taking expectation and applying Lemma 2.1 (notice that $\mathbf{y}$ is independent of $\mathbf{d}_j$), we find that $\mathbb{E}\left[\|\mathbf{x}_{j+1} - \mathbf{x}^\star\|_{\mathbf{A}}^2\right] \geq (1 - \lambda_{\max}/n)\mathbb{E}\left[\|\mathbf{x}_j - \mathbf{x}^\star\|_{\mathbf{A}}^2\right]$.

8.4   Proof of Equation (10)

Let

$$
C_i = \{rT + i - \tau \leq t \leq rT + i - 1 : t \geq rT\}
$$

and

$$
D_i = \{rT + i - \tau \leq t \leq rT + i - 1 : t < rT\} .
$$

Unrolling the recursion in equation (7) starting at rT, we find that for $r \geq 1$ and $w \geq 0$

$$
\begin{aligned}
E_{(r+1)T+w} \;\leq\; & E_{rT} - (1-\rho\tau)\sum_{i=0}^{T-1+w}\mathbb{E}\left[(\mathbf{x}_{k(rT+i)}-\mathbf{x}^\star,\mathbf{d}_{rT+i})_{\mathbf{A}}^2\right] + \rho\sum_{i=0}^{T-1+w}\sum_{t=k(rT+i)}^{rT+i-1}\mathbb{E}\left[(\mathbf{x}_{k(t)}-\mathbf{x}^\star,\mathbf{d}_t)_{\mathbf{A}}^2\right] \\
\leq\; & E_{rT} - (1-\rho\tau)\sum_{i=0}^{T-1+w}\mathbb{E}\left[(\mathbf{x}_{k(rT+i)}-\mathbf{x}^\star,\mathbf{d}_{rT+i})_{\mathbf{A}}^2\right] + \rho\sum_{i=0}^{T-1+w}\sum_{t=rT+i-\tau}^{rT+i-1}\mathbb{E}\left[(\mathbf{x}_{k(t)}-\mathbf{x}^\star,\mathbf{d}_t)_{\mathbf{A}}^2\right] \\
\leq\; & E_{rT} - (1-\rho\tau)\sum_{i=0}^{T-1+w}\mathbb{E}\left[(\mathbf{x}_{k(rT+i)}-\mathbf{x}^\star,\mathbf{d}_{rT+i})_{\mathbf{A}}^2\right] + \rho\sum_{i=0}^{T-1+w}\sum_{t\in C_i}\mathbb{E}\left[(\mathbf{x}_{k(t)}-\mathbf{x}^\star,\mathbf{d}_t)_{\mathbf{A}}^2\right] \\
& +\rho\sum_{i=0}^{T-1+w}\sum_{t\in D_i}\mathbb{E}\left[(\mathbf{x}_{k(t)}-\mathbf{x}^\star,\mathbf{d}_t)_{\mathbf{A}}^2\right] \\
\leq\; & E_{rT} - (1-2\rho\tau)\sum_{i=0}^{T-1+w}\mathbb{E}\left[(\mathbf{x}_{k(rT+i)}-\mathbf{x}^\star,\mathbf{d}_{rT+i})_{\mathbf{A}}^2\right] + \rho\sum_{i=0}^{\tau-1}\sum_{t\in D_i}\mathbb{E}\left[(\mathbf{x}_{k(t)}-\mathbf{x}^\star,\mathbf{d}_t)_{\mathbf{A}}^2\right] \;. \\
\leq\; & E_{rT} - (1-2\rho\tau)\sum_{i=\tau}^{T-1+w}\mathbb{E}\left[(\mathbf{x}_{k(rT+i)}-\mathbf{x}^\star,\mathbf{d}_{rT+i})_{\mathbf{A}}^2\right] + \rho\sum_{i=0}^{\tau-1}\sum_{t\in D_i}\mathbb{E}\left[(\mathbf{x}_{k(t)}-\mathbf{x}^\star,\mathbf{d}_t)_{\mathbf{A}}^2\right] \;.
\end{aligned}
$$

The second-to-last inequality follows from the fact that each term of the form $\mathbb{E}\left[(\mathbf{x}_{k(l)}-\mathbf{x}^\star,\mathbf{d}_l)_{\mathbf{A}}^2\right]$ appears at most τ times in $\rho\sum_{i=0}^{T-1+w}\sum_{t\in C_i}\mathbb{E}\left[(\mathbf{x}_{k(t)}-\mathbf{x}^\star,\mathbf{d}_t)_{\mathbf{A}}^2\right]$. We also use the fact that for $i\geq\tau$ we trivially have $D_i=\emptyset$.