SIAM J. MATRIX ANAL. APPL.
Vol. 42, No. 2, pp. 800–831

AN IMPLICIT REPRESENTATION AND ITERATIVE SOLUTION OF RANDOMLY SKETCHED LINEAR SYSTEMS*

VIVAK PATEL†, MOHAMMAD JAHANGOSHAHI‡, AND DANIEL A. MALDONADO§

Abstract. Randomized linear system solvers have become popular as they have the potential to reduce floating point complexity while still achieving desirable convergence rates. One particularly promising class of methods, random sketching solvers, has achieved the best known computational complexity bounds in theory, but is blunted by two practical considerations: there is no clear way of choosing the size of the sketching matrix a priori; and there is a nontrivial storage cost of the sketched system. In this work, we make progress towards addressing these issues by implicitly generating the sketched system and solving it simultaneously through an iterative procedure. As a result, we replace the question of the size of the sketching matrix with determining appropriate stopping criteria; we also avoid the costs of explicitly representing the sketched linear system; and our implicit representation also solves the system at the same time, which controls the per-iteration computational costs. Additionally, our approach allows us to generate a connection between random sketching methods and randomized iterative solvers (e.g., the randomized Kaczmarz method and randomized Gauss–Seidel). As a consequence, we exploit this connection to (1) produce a stronger, more precise convergence theory for such randomized iterative solvers under arbitrary sampling schemes (i.i.d., adaptive, permutation, dependent, etc.), and (2) improve the rates of convergence of randomized iterative solvers at the expense of a user-determined increase in per-iteration computational and storage costs. We demonstrate these concepts on numerical examples on 49 distinct linear systems.

Key words. random matrix sketching, orthogonalization, random iterative methods, linear systems

AMS subject classifications. 15A06, 15B52, 65F10, 65F25, 65N75, 65Y05, 68W20, 68W40

DOI. 10.1137/19M1259481

1. Introduction.

Over the past few decades, randomized linear system solvers have become popular as they have the potential to reduce floating point complexity or maintain limited memory footprints while still achieving desirable convergence rates (e.g., [35, 38]). In particular, the noniterative class of randomized linear system solvers, based on random matrix sketching (see [38]), has exceptionally low computational complexities, at least in theory. Unfortunately, the theoretical promise of these random matrix sketching solvers is blunted by their practical limitations: there is no clear way of choosing the size of the sketching matrix, and there is a nontrivial storage cost of the sketched system [27]. In fact, the practical challenges of random matrix sketching solvers have prevented them from being fully embraced by the numerical optimization community (e.g., [29]).

In this work, we begin to address these two primary practical issues of random matrix sketching, which we recall are the challenge of choosing the size of the sketching matrix, and the challenge of storing the sketched system. Our main insight is to recast the separate sketch-*then*-solve core of random sketching methods into an

*Received by the editors May 2, 2019; accepted for publication (in revised form) by P. Drineas March 9, 2021; published electronically June 8, 2021.
https://doi.org/10.1137/19M1259481
Funding: The work of the authors was supported by the UW-Madison WARF award AAD5914 and the DOE grant DE-AC02-06CH11347.
†Statistics, University of Wisconsin – Madison, Madison, WI 53706 USA (vivak.patel@wisc.edu).
‡Susquehanna International Group, Bala Cynwyd, PA 19004 USA (mjahangoshahi@uchicago.edu).
§Mathematics and Computer Science, Argonne National Laboratories, Lemont, IL 60439 USA (maldonadod@anl.gov).

equivalent, iterative sketch-*and*-solve, in which the sketching matrix is generated incrementally without being explicitly stored, and the system is incrementally solved from the implicitly derived sketched matrix.[1] As a result of our approach, (1) we can implicitly grow the size of the sketching matrix until a user-determined stopping criterion is reached without having to determine the size of the sketching matrix a priori; (2) we implicitly represent the sketched system without having to explicitly store the sketched system, which allows us to avoid the cost of storing the sketched system; and (3) we can naturally implement random sketching solvers within distributed and parallel computing paradigms. Thus, our approach of converting the usual sketch-*then*-solve procedure to a sketch-*and*-solve procedure begins to address the aforementioned practical challenges of random matrix sketching.

Moreover, our approach provides a bridge between random sketching methods and (what we will call) base randomized iterative methods[2] on a single spectrum of procedures, which has several immediate consequences. First, the number of rows of the sketching matrix that results in the solution (this number is a random quantity) connects to an alternative rate-of-convergence result for general base randomized iterative methods that guarantees a rate of convergence less than one for arbitrary sampling schemes—even for underdetermined systems (Theorem 4.2). Consequently, our results complement and improve on previous results in several ways. In particular, we allow for arbitrary sampling schemes—not just sampling schemes that are independent and identically distributed (i.i.d.) as in [16, Lemma 4.2], [32, Theorem 4.8], [41, Theorem 3.4], and [26, equation (3.10)]. Moreover, our results do away with the exactness assumption of [32, Assumption 2] and precisely characterize the inexactness that can occur for arbitrary sampling schemes (Theorems SM3.1 and 4.2). Additionally, our results define convergence on a maximal subset—effectively, a set occurring with probability one for sampling schemes of interest—which builds on the work of [5]. As example applications of our results, we supply rates of convergence with probability one for random permutation sampling methods (Proposition 4.5) and i.i.d. sampling schemes (asymptotically, see Proposition 4.6). As a more interesting application of our results, we specify generic conditions for the convergence of a broad class of adaptive schemes (see subsection 4.3), which can account for the maximum residual scheme, the maximum distance scheme, schemes that randomize over a greedy subset, and schemes that are greedy over randomized subsets [28, 17, 24, 4, 31, 3, 18]. We note that the rates that we provide as examples are rather loose in comparison to results that are specialized to each case, yet our results often supply information that is *not* available in these other results as discussed above.

Second, we can generate a series of "intermediate" procedures between sketching methods and base methods that trade off between computational resources (e.g., floating point operations, storage) and rates of convergence. Thus, we can take a sketching method and reduce its computational footprint in exchange for a slower rate of convergence, or increase the computational footprint of base methods to improve their rate of convergence (Algorithm SM1.1). Moreover, these "intermediate" procedures can be readily parallelized as we discuss in section 2.

Finally, by shifting our perspective from improving the sketch-*then*-solve procedure to improving the performance of base methods, we find that our approach is a

[1] Random sketching solvers have been used iteratively in a different sense (e.g., see [16]); the noniterative scheme is simply repeated to improve convergence.

[2] We will be more precise about what we refer to as base methods. For now, such methods are exemplified by randomized Kaczmarz [35] and randomized Gauss–Seidel [25].

randomized orthogonalization procedure in the row space of the coefficient matrix of the linear system. Thus, by presenting our approach from this latter perspective, we will simplify the introduction and the related theory of our approach. Now, before pursuing this further, we reiterate our main contributions.

1. First, we turn the typical sketch-*then*-solve noniterative random sketching solver into an iterative, sketch-*and*-solve method, which lays a foundation for addressing the previously enumerated practical challenges of random sketching solvers: there is no clear way of choosing the size of the sketching matrix a priori; and there is a nontrivial storage cost of the sketched system.

2. Second, through our approach, we place random sketching methods (e.g., [38, 36]) and base randomized iterative methods (e.g., randomized Kaczmarz, randomized Gauss–Seidel, Sketch-and-Project [16]) on a single spectrum of methods.

3. Third, owing to this connection, we are able to generate "intermediate" methods between random sketching and base methods, which can trade off between computational resources and rates of convergence.

4. Fourth, owing to this connection, we use the geometric implications of random sketching methods to develop an alternative rate-of-convergence result for general base methods for arbitrarily determined systems and *arbitrary sampling schemes*, which advances the with-probability-one results of [5], generalizes the deterministic cyclic results in [2, 11, 37], complements the mean-squared error results of [32], and accounts for a litany of adaptive methods considered in [28, 17, 24, 4, 31, 3, 18].

5. Finally, we provide a generic set of conditions for characterizing a broad class of adaptive methods and, from these conditions, prove convergence and rate-of-convergence results for a number of classical and emerging adaptive methods in the literature under a unified framework (see subsection 4.3).

The remainder of this paper is organized as follows. In section 2, we introduce our procedure; we state the connection between our procedure and random sketching methods, which allows us to convert the less practical sketch-*then*-solve approach to our sketch-*and*-solve approach; and, finally, we introduce our general algorithm and variants for low-memory environments, shared memory environments, distributed memory environments, and large, sparse, structured linear systems. In sections 3 and 4, we develop the convergence theory for the two methodological extremes—sketching and base methods—leaving the intermediate, more complex cases to future work, and discuss particular examples. In section 5, we test our algorithms on 49 distinct linear systems. In section 6, we conclude this work and preview future efforts.

2. Our procedure.

Our approach is best introduced from the perspective of base randomized iterative methods. Here, we overview these methods and introduce our procedure (subsection 2.1), leaving a heuristic derivation and detailed examples to section SM1. We then refine our procedure for the case of rank-one methods, which allows us to restate random sketching from a sketch-*then*-solve procedure to a sketch-*and*-solve procedure (subsection 2.2). We conclude this section with comments on algorithmic refinements for parallel platforms (subsection 2.3.1), reduced memory platforms (subsection 2.3.2), and communication-focused platforms (subsection 2.3.3).

2.1. Overview.

Let $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$ be the coefficient matrix and constant vector, respectively. Assuming consistency, our goal is to determine an $x^* \in \mathbb{R}^d$, not necessarily unique, such that $Ax^* = b$. In a base randomized iterative approach, a sequence of iterates $\{x_k : k+1 \in \mathbb{N}\}$ is generated that has the form $x_{k+1} = x_k + V_k(b - Ax_k)$, where $V_k \in \mathbb{R}^{d\times n}$ are possibly dependent random variables, which we

call residual projection matrices (RPMs).[3] The RPM defines the base technique which is being used, and a number of such examples as randomized Kaczmarz, randomized Gauss–Seidel, randomized block coordinate descent, and the sketch-and-project of [16, 32] are detailed in subsection SM1.1.

Our procedure modifies the base randomized method by augmenting V_k with a matrix M_k, specifically by $x_{k+1} = x_k + M_k V_k(b - Ax_k)$. The choice of $\{M_k\}$ is intuitively derived in subsection SM1.2. Here, we state our procedure and briefly interpret the less obvious quantities. Let $S_0 \in \mathbb{R}^{d\times d}$ be the identity matrix, I_d.[4] Given $\{V_k : k+1 \in \mathbb{N}\}$, let $x_0 \in \mathbb{R}^d$, and define

$$x_{k+1} = x_k + M_k V_k(b - Ax_k), \tag{2.1a}$$

$$M_k = S_k A' V_k' (V_k A S_k A' V_k')^\dagger, \tag{2.1b}$$

$$S_{k+1} = S_k - S_k A' V_k' (V_k A S_k A' V_k')^\dagger V_k A S_k. \tag{2.1c}$$

To interpret the terms in the above procedure, we begin by ignoring S_k (i.e., set it to the identity). In this case, M_k and its role in updating x_k to x_{k+1} is familiar: M_k serves to map the residual onto the row space of $V_k A$, thereby ensuring that x_{k+1} satisfies $V_k A x_{k+1} = V_k b$. If we now consider the role of S_k, we see that it is an orthogonal projector that "weights" the behavior of M_k to ensure that x_{k+1} satisfy $V_i A x_{k+1} = V_i b$ for $i \le k$. We will see these interpretations clearly and formally when we focus on the case of rank-one V_k next.

We pause here momentarily to discuss the relationship between our procedure, as specified by (2.1a)–(2.1c), and the sketch-and-project method in [16, 32]. At first glance, it may seem that our procedure is a special case of sketch-and-project with adaptive choices of the inner product at each iteration of the sketch-and-project update. Unfortunately, an effort to recast our approach as a special case of sketch-and-project breaks down at two fundamental points. First, the adaptive choices of the sketch-and-project inner product would have to be the inverse of S_k, which is orthogonal projection matrices. As a result, the inverse is ill-defined and the inner product is ill-defined. Of course, this can be rectified by allowing for a pseudo-metric, but this then results in the second major point of difficulty: the theory presented in [16, 32] relies on the determinism and invertibility of the matrix defining the metric space to prove convergence. Thus, sketch-and-project, without a substantial investment, cannot readily include our approach. On the other hand, we can state sketch-and-project as a base randomized iterative approach, as shown in subsection SM1.1, and then improve on it with our procedure via (2.1a)–(2.1c).

2.2. Rank-one refinements and random sketching. We will now focus on a particular refinement of (2.1a)–(2.1c) that occurs when $\{V_k\}$ are rank-one matrices, that is, when there exist pairs of vectors $\{(v_k, w_k)\}$ such that $V_k = v_k w_k'$ for each k. Our focus on this refinement is motivated by the simplifications of form (compare (2.1a)–(2.1c) to (2.2b) and (2.3)) and the resulting straightforward connection that such rank-one matrices will have with matrix sketching, which we discuss further below.

[3] Despite having the word "projection" in the name, V_k may not be actually be a projection.

[4] The choice of $S_0 = I_d$ may seem peculiar until we get to Lemma 3.1. As a brief preview, we will need S_0 to be a projection matrix. While other projection matrices can be used, they would need to be designed to ensure that the initial error is in the range of the projection.

In the case of $\{V_k = v_k w_k'\}$, (2.1b) and (2.1c) become

$$M_k = \frac{1}{w_k' A S_k A' w_k \left\|v_k\right\|_2^2} S_k A' w_k v_k' \mathbf{1}\left[S_k A' w_k \neq 0\right], \tag{2.2a}$$

$$S_{k+1} = S_k - \frac{1}{w_k' A S_k A' w_k} S_k A' w_k w_k' A S_k \mathbf{1}\left[S_k A' w_k \neq 0\right]. \tag{2.2b}$$

Moreover, if we substitute (2.2a) into (2.1a), we recover

$$x_{k+1} = x_k + \frac{1}{w_k' A S_k A' w_k} S_k A' w_k w_k' (b - A x_k) \mathbf{1}\left[S_k A' w_k \neq 0\right]. \tag{2.3}$$

It follows from (2.2b) and (2.3) that in the case of a rank-one RPM, *the left singular vector of the RPM is not important.*[5] As we now explain, this observation is critical for converting the impractical noniterative randomized sketch-*then*-solve methods into iterative randomized sketch-*and*-solve methods.

Recall that the fundamental sketch-then-solve procedure is to construct a specialized matrix $N^{\text{sketch}} \in \mathbb{R}^{k \times n}$ and then generate and solve the smaller, sketched problem $(N^{\text{sketch}} A)x = N^{\text{sketch}} b$ (see [38, Ch. 1]).[6] The special matrix N^{sketch}, called the sketching matrix, can be generated in a variety of ways such as making each entry an i.i.d. Gaussian random variable [20], or by setting the columns of N^{sketch} as uniformly sampled columns (with replacement) of the appropriately dimensioned identity matrix [7].

In order to convert the usual sketch-*then*-solve procedure into our sketch-*and*-solve procedure, we simply set $\{w_k : k+1 \in \mathbb{N}\} \subset \mathbb{R}^n$ to the transposed rows of N^{sketch}, which we will rigorously demonstrate in section 3. Of course, this requires that we have a streaming procedure for generating arbitrarily many rows of N^{sketch}. For the Gaussian strategy [20] and the sparse Count-Sketch strategy [7, 6], this is a straightforward task, and details are supplied in subsection SM1.4. Thus, if we let `RPMStrategy()` define a generic user-defined procedure for choosing $\{w_k : k+1 \in \mathbb{N}\}$, then this observation gives us Algorithm 2.1 for (1) converting the sketch-*then*-solve procedure into a sketch-*and*-solve procedure, and (2) adding orthogonalization to such base methods as randomized Kaczmarz and randomized Gauss–Seidel.

2.3. Algorithmic refinements considering the computing platform. Algorithm 2.1 implicitly assumes the traditional sequential programming paradigm. However, the performance of the algorithm can be improved by taking advantage of parallel computing architectures. Here, we will consider a handful of important computing architecture abstractions and how our procedure can adapt to different configurations. In subsection 2.3.1, we will consider the case of a parallel computing architecture for which the communication overhead, which is proportional to the dimension d, is not a limiting factor. In subsection 2.3.2, we consider a similar class of problems where the communication of $\mathcal{O}[d]$-sized vectors is acceptable and $n \gg d$, but that d is so large that storing and manipulating a matrix in $\mathbb{R}^{d \times d}$ is burdensome. Finally, in subsection 2.3.3 we will consider problems in which computational overhead becomes a bottleneck for scalability but for which we have structured systems that can be exploited to manage such overheads (e.g., [9]).

[5] Explicit examples for modifying randomized Kaczmarz and Gauss–Seidel are presented in subsection SM1.3.

[6] We note that the typical formulation considers linear regression rather than a linear system.

Algorithm 2.1 Rank-One RPM Method

INPUT: Initialization x_0; `RPMStrategy()` for $\{w_l\}$; `TerminationCriteria()`
$k \leftarrow 0$
$S \leftarrow I_d$
while `TerminationCriteria()` == false **do**
  # Compute search direction
  $w_k \leftarrow$ `RPMStrategy()`
  $q_k \leftarrow A'w_k$ *# Row of the sketched system, used later to update x_k and S_k*
  $u_k \leftarrow S_k q_k$ *# Search direction from $S_k A'w_k$ in (2.3)*
  # Check if $S_k A'w_k = 0$
  if $u_k == 0$ **then**
    $k \leftarrow k + 1$
    continue to next iteration
  end if
  # Update Iterate
  $r_k \leftarrow b'w_k - q_k'x_k$ *# Component of sketched residual; $w_k'(b - Ax_k)$ in (2.3)*
  $\gamma_k \leftarrow u_k'q_k$ *# Calculation of $w_k'AS_kA'w_k$ in (2.3)*
  $x_{k+1} \leftarrow x_k + u_k\,(r_k/\gamma_k)$
  # Update Projection Matrix
  $S_{k+1} \leftarrow (I - \frac{1}{\gamma_k}u_k q_k')S_k$
  # Update Iteration Counter
  $k \leftarrow k + 1$
end while
RETURN: x_{k+1}

2.3.1. Asynchronous parallelization on shared and distributed memory platforms. First, when we are using a matrix sketch for `RPMStrategy()`, one of the expensive components of the computation is determining $\begin{bmatrix} A & b \end{bmatrix}' w_k$. Fortunately, in our sketch-and-solve procedure, this expensive computation can be trivially asynchronously parallelized on a shared memory platform when

1. the data within the rows $\begin{bmatrix} A & b \end{bmatrix}$ are stored together, and
2. the `RPMStrategy()` generates $\{w_k : k + 1 \in \mathbb{N}\}$ that are either independent (e.g., the Gaussian strategy) or can be grouped into independent subsets (e.g., the Count-Sketch strategy).

When these two requirements are met, each processor can generate its own $\{w_k : k + 1 \in \mathbb{N}\}$ independently of the other processors and evaluate $\begin{bmatrix} A & b \end{bmatrix}' w_k$. It can then simply write the resulting row to an address reserved for performing the iterate and S_k matrix updates by the master processor. Importantly, this procedure does not require locking any of the rows of $\begin{bmatrix} A & b \end{bmatrix}$, and the reserved addresses can use fine grained locks to prevent any wasted calculations.

Similarly, in our sketch-and-solve procedure, computing $\begin{bmatrix} A & b \end{bmatrix}' w_k$ can be trivially asynchronously parallelized on a distributed memory platform using a Fork-join model, when

1. the rows of $\begin{bmatrix} A & b \end{bmatrix}$ are distributed across the different storages, and
2. the `RPMStrategy()` generates $\{w_k : k+1 \in \mathbb{N}\}$ such that w_k have independent groups of components (e.g., the Gaussian strategy and the Count-Sketch strategy).

When these two requirements are met, each processor can generate its own $\{w_k : k+1 \in \mathbb{N}\}$ and operate on the local rows of $\begin{bmatrix} A & b \end{bmatrix}$. It can then simply pass the resulting row to the master processor, which performs the iterate and S_k matrix updates. For each iteration, a scattering and gathering of the data is performed, but no other data exchange is required.

Table 1 summarizes the time and total computational costs of computing x_k and S_k from x_0 and S_0 in the following context: (1) the sequential platform refers to the case where there is a single processor with a sufficiently large memory to store the system and perform the necessary operations in Algorithm 2.1; (2) the shared memory platform assumes that there are $p+1$ processors that share a sufficiently large memory. One of the processors is dedicated to performing the iterate and matrix updates, while the remaining p processors compute $\begin{bmatrix} A & b \end{bmatrix}' w_k$; (3) the distributed memory architecture assumes that there are $p+1$ processors each with a sufficient memory capacity. The rows of $\begin{bmatrix} A & b \end{bmatrix}$ are split evenly or nearly evenly amongst p of the processors, and each process only manipulates its local information about A and b. Finally, the master processor is dedicated to performing the iterate and matrix updates.

TABLE 1

A summary of the time and total computational cost (effort) incurred by Algorithm 2.1 *and its parallelized variants. We do not report any advantages that should be exploited when A or w is sparse. In the shared and distributed memory platforms, we assume that there are p processors dedicated to computing $A'w$ and $b'w$, and one processor dedicated to computing the updates. The "Network" column refers to whether communication costs over a network are incurred.*

Total Time and Effort Costs to Iteration k					
Platform	Computing $\begin{bmatrix} A & b \end{bmatrix}' w$		Update costs		Network
	Time	Total effort	Iterate	Matrix	
Sequential	$\mathcal{O}[knd]$	$\mathcal{O}[knd]$	$\mathcal{O}[kd^2]$	$\mathcal{O}[kd^3]$	No
Shared memory	$\mathcal{O}[knd/p]$	$\mathcal{O}[knd]$	$\mathcal{O}[kd^2]$	$\mathcal{O}[kd^3]$	No
Distributed memory	$\mathcal{O}[knd/p^2]$	$\mathcal{O}[knd/p]$	$\mathcal{O}[kd^2]$	$\mathcal{O}[kd^3]$	Yes

2.3.2. Memory-reduced procedure. A notable aspect of Algorithm 2.1 (and its aforementioned parallel variants described above) is that it must store and manipulate the matrix S_k at each iteration, which is clearly expensive when d is large or is excessive when d^3 is comparable to n or greater than n. This difficulty motivates a partial orthogonalization approach, as described in Algorithm SM1.1. In this approach, a user-defined parameter $m < d$ specifies the number of d-dimensional vectors needed to implicitly store an approximate representation of S_k (based on Lemma 3.1). We denote this set of m d-dimensional vectors by $\mathcal{S}$. With this implicit representation, the cost of computing u_k reduces to $\mathcal{O}[md]$,[7] which, consequently, reduces the overall cost of updating x_k to x_{k+1} to $\mathcal{O}[md]$. Moreover, because S_k is implicitly represented by a set of m d-dimensional vectors, $\mathcal{S}$, there is no notable additional computational cost incurred for updating S_k to S_{k+1}. Thus, an entire iteration incurs a computational cost $\mathcal{O}[md]$ plus the cost of computing $\begin{bmatrix} A & b \end{bmatrix}' w_k$, which can be mollified under the strategies above in shared memory or distributed memory platforms.

[7] If q_k replaces u_k in the calculation of z_k, then the cost of computing u_k is $\mathcal{O}[dm^2]$ (see [13, Ch. 5.2]).

2.3.3. Optimizing communication overhead. Structured systems. In the above approaches, we take for granted that d is not so large that communicating $\mathcal{O}[d]$ vectors is acceptable during the procedure. However, for many problems coming from the solution of differential equations (e.g., see [9]), d and n are of the same order and are so large that communicating $\mathcal{O}[d]$ vectors at arbitrary points during the procedure is impossible. Fortunately, linear system problems in this class are highly sparse and structured [34, Ch. 2]. A simple example is the case where A is a square, banded system with nonzero bandwidth $\tilde{Q}+1$ for some $\tilde{Q} \ll n = d$; that is, $A_{ij} = 0$ if $|i-j| > \tilde{Q}$ and the remaining A_{ij} can take arbitrary values.

For such sparse and structured problems, our methodology can be efficiently implemented across a distributed memory platform with p processors under some additional qualifications. However, to understand these qualifications, let us first introduce some notation and concepts that define the communication pattern across the p nodes. Suppose somehow that we distribute the equations of our linear system of interest across p nodes. Figure 1 shows how the coefficient matrix of a 20×20 banded system with bandwidth 5 can be distributed across five nodes. Note that, in this example, the entries of the constant vector would be stored on the same processor as the corresponding rows of the coefficient matrix. Moreover, we need a way of tracking which components of x are manipulated by each node: let $\mathcal{X}_i$ be the set of indices of the components of x with nonzero coefficients at node i in the distributed system for $i = 1, \ldots, p$. In our example, $\mathcal{X}_1 = \{1, \ldots, 6\}$, $\mathcal{X}_2 = \{3, \ldots, 10\}$, $\mathcal{X}_3 = \{7, \ldots, 14\}$, $\mathcal{X}_4 = \{11, \ldots, 18\}$, and $\mathcal{X}_5 = \{15, \ldots, 20\}$. Finally, for any vector z and any set $\mathcal{X}$ over the indices of z, let $z[\mathcal{X}]$ be the vector whose elements are the elements of z indexed by $\mathcal{X}$.

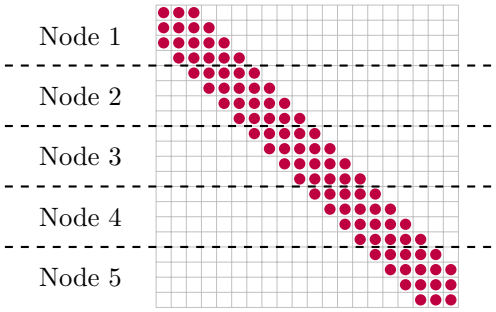

FIG. 1. *A representation of a* 20×20 *banded matrix with bandwidth* $\tilde{Q}+1 = 5$*, whose rows are split across five compute nodes (represented by the dashed line). Note that the empty grid points represent zeros, while the filled grid points represent nonzero values.*

From this example and from our discussion in subsection 2.3.1 of distributing the `RPMStrategy()`, we can use the local rows of A at Node 1 and a Gaussian sketch to generate a $q_1 \in \mathbb{R}^d$ such that $q_1[\{1, \ldots, 6\}]$ are arbitrarily valued and $q_1[\{7, \ldots, 20\}] = 0$. Thus, our vector q_k is highly sparse and can be generated locally on the node. However, following Algorithm 2.1, the next step of computing u_k requires computing the product between S_k and q_k, which, in a naive implementation, would require storing a dense $d \times d$ matrix S_k and computing a global matrix-vector product. Such a required computation raises several concerns, which we detail and address in the following enumeration.

1. Given that d is relatively large to the computing environment, is storing a $d \times d$ matrix even feasible? Generally, the answer will be that storing such a matrix

is infeasible. However, by exploiting the properties of S_k (see Lemma 3.1), we will approximately and implicitly store S_k as $\mathcal{S}$, which is a collection of orthonormal vectors.

2. Even if we use $\mathcal{S}$ in place of S_k, will the resulting implicit matrix-vector product and update of $\mathcal{S}$ incur prohibitive communication costs? To answer these questions completely, we will need to specify how the implicit matrix-vector product will be computed and how $\mathcal{S}$ will be stored. Here, we will compute the implicit matrix-vector product by using twice-iterated classical Gram–Schmidt (Algorithm SM1.3), which was shown to be numerically stable in the seminal work of [12]. Owing to this calculation pattern, we can store $\mathcal{S}$ in a distributed fashion across the p processors. As derived in detail in subsection SM1.6, synchronization of $\mathcal{S}$ will require $Q(F-1)+m(p^2-1)$ floating point numbers to be communicated per iteration, where Q represents the number of shared indices between two processors; F represents the maximum number of nodes that overlap; and m represents the memory storage parameter.

This procedure is detailed in Algorithm SM1.4.

3. Convergence theory for orthogonalization. Here, we will prove that the complete orthogonalization approach (i.e., Algorithm 2.1) converges to the solution under a variety of sampling RPM strategies. We anticipate that a similar argument would demonstrate the convergence of the general procedure described in (2.1).

3.1. Core results. We establish two key results. First, we establish that our procedure is an orthogonalization procedure: that is, the matrices $\{S_k\}$ project the current search direction onto a subspace that is orthogonal to previous search directions. Second, we characterize the limit point of our iterates, $\{x_k\}$, in terms of a true solution of the linear system and the subspace generated by the rank-one RPMs, $\{V_k\}$.

LEMMA 3.1. *Let $\{w_\ell : \ell+1 \in \mathbb{N}\} \subset \mathbb{R}^n$ be an arbitrary sequence in $\mathbb{R}^n$, and let $\mathcal{R}_0 = \{0\} \subset \mathbb{R}^d$ and $\mathcal{R}_\ell = \operatorname{span}[A'w_0, \dots, A'w_{\ell-1}]$ for $\ell \in \mathbb{N}$. Now, let $S_0 = I_d$ and $\{S_\ell : \ell \in \mathbb{N}\}$ be defined recursively as in* (2.2b)*. Then, for $\ell \geq 0$, S_ℓ is an orthogonal projection matrix onto $\mathcal{R}_\ell^\perp$.*

Proof. We will prove the result by induction. For the base case, $\ell = 0$, $S_0 = I_d$. It follows that S_0 is an orthogonal projection onto $\mathcal{R}_0^\perp = \mathbb{R}^d$ since $S_0^2 = I_d^2 = I_d = S_0$ and $\operatorname{range}(I_d) = \mathbb{R}^d$. Now suppose that the result holds for $\ell > 0$. If $S_\ell A'w_\ell = 0$, then there is nothing to show. Therefore, for the remainder of this proof, suppose $S_\ell A'w_\ell \neq 0$.

First, we show that $S_{\ell+1}$ is a projection matrix by verifying that $S_{\ell+1}^2 = S_{\ell+1}$ by direct calculation. Making use of the recursive definition of $S_{\ell+1}$ and the induction hypothesis that $S_\ell^2 = S_\ell$ (since S_ℓ is a projection),

$$
\begin{aligned}
S_{\ell+1}^2 &= \left(S_\ell - \frac{S_\ell A' w_\ell w_\ell' A S_\ell}{w_\ell' A S_\ell A' w_\ell}\right)\left(S_\ell - \frac{S_\ell A' w_\ell w_\ell' A S_\ell}{w_\ell' A S_\ell A' w_\ell}\right) \\
&= \left(S_\ell - \frac{S_\ell A' w_\ell w_\ell' A S_\ell}{w_\ell' A S_\ell A' w_\ell}\right)\left(I_d - \frac{A' w_\ell w_\ell' A S_\ell}{w_\ell' A S_\ell A' w_\ell}\right) \\
&= S_\ell - 2\frac{S_\ell A' w_\ell w_\ell' A S_\ell}{w_\ell' A S_\ell A' w_\ell} + \frac{S_\ell A' w_\ell w_\ell' A S_\ell}{w_\ell' A S_\ell A' w_\ell} = S_{\ell+1}.
\end{aligned} \tag{3.1}
$$

Second, we use the fact that a projection is orthogonal if and only if it is self-adjoint to show that $S_{\ell+1}$ is an orthogonal projection. By induction, because S_ℓ is an

orthogonal projection, $S'_\ell = S_\ell$, and so

$$S'_{\ell+1} = S'_\ell - \frac{S_\ell A' w_\ell w'_\ell A S_\ell}{w'_\ell A S_\ell A' w_\ell} = S_{\ell+1}. \tag{3.2}$$

Finally, let v be in the range of $S_{\ell+1}$, and we can decompose v into the components u and y such that $v = u + y$, $0 = u'y$, and $y \in \mathcal{R}_{\ell+1}$. We will show that $y = 0$, which characterizes the range of $S_{\ell+1}$ as being all vectors orthogonal to $\mathcal{R}_{\ell+1}$. To show this note that because $S_{\ell+1}$ is a projection matrix, we have that

$$u + y = v = S_{\ell+1} v = S_{\ell+1} u + S_{\ell+1} y. \tag{3.3}$$

By construction $\mathcal{R}_\ell \subset \mathcal{R}_{\ell+1}$ and so $u \in \mathcal{R}_\ell^\perp$. Using the induction hypothesis, we then have that $S_\ell u = u$. Moreover, because $u \in \mathcal{R}_{\ell+1}^\perp$ by construction, $u' A' w_\ell = 0$. Then, using the recursive definition of $S_{\ell+1}$, we have that

$$S_{\ell+1} u = S_\ell u - \frac{S_\ell A' w_\ell w'_\ell A S_\ell u}{w'_\ell A S_\ell A' w_\ell} = u - \frac{S_\ell A' w_\ell w'_\ell A u}{w'_\ell A S_\ell A' w_\ell} = u. \tag{3.4}$$

Therefore, $u = S_{\ell+1} u$ and, by (3.3), $y = S_{\ell+1} y$. We now decompose y into y_1 and y_2, where $y_1 \in \mathcal{R}_\ell$ and $y_2 \in \mathcal{R}_\ell^\perp \cap \mathcal{R}_{\ell+1}$. By the induction hypothesis, $\mathcal{R}_\ell^\perp \cap \mathcal{R}_{\ell+1} = \operatorname{span}[S_\ell A' w_\ell]$. Therefore, $S_\ell y = y_2$ and $\exists \alpha \in \mathbb{R}$ such that $y_2 = \alpha S_\ell A' w_\ell$. Finally, using the recursive formulation of $S_{\ell+1}$ and $S_\ell y = y = \alpha S_\ell A' w_\ell$,

$$y = S_{\ell+1} y = S_\ell y - \frac{S_\ell A' w_\ell w'_\ell A S_\ell y}{w'_\ell A S_\ell A' w_\ell} = \alpha S_\ell A' w_\ell - \alpha S_\ell A' w_\ell = 0. \tag{3.5}$$

Thus, we have shown that the range of $S_{\ell+1}$ is orthogonal to $\mathcal{R}_{\ell+1}$. ∎

From Lemma 3.1, we see that our procedure is an orthogonalization procedure just like quasi-Newton methods [30, Ch. 8] and conjugated direction methods [19]. In light of this relationship, we could follow the common strategy used in the proof of such procedures: demonstrate that the iterates are in a subspace generated by the initial iterate and the search directions, and then demonstrate that the iterates are the closest points to the true solutions within the given subspace. For our procedure, we effectively follow the same strategy. We see that the first part of the proof strategy is true since

$$x_{\ell+1} \in \operatorname{span}[x_0, S_0 A' w_0, \ldots, S_\ell A' w_\ell] = \operatorname{span}[x_0, A' w_0, \ldots, A' w_\ell], \tag{3.6}$$

where the latter equality holds by Lemma 3.1. We will prove the second part in the following lemma.

LEMMA 3.2. *Suppose $Ax = b$ admits a solution x^* (not necessarily unique). Let $w_0, w_1, \ldots \in \mathbb{R}^n$ be random variables. Let $x_0 \in \mathbb{R}^d$ be arbitrary and $S_0 = I_d$, and let $\{x_\ell : \ell \in \mathbb{N}\}$ and $\{S_\ell : \ell \in \mathbb{N}\}$ be defined as in* (2.2b) *and* (2.3). *Then, for all $\ell \geq 0$, $x_\ell - x^* = S_\ell (x_0 - x^*)$.*

Proof. We will prove this by induction. For $\ell = 0$, the statement holds since $S_0 = I_d$. Now suppose that this relationship holds for some $\ell > 0$. Using (2.3),

$$\begin{aligned} x_{\ell+1} - x^* &= x_\ell - x^* + \frac{S_\ell A' w_\ell w'_\ell}{w'_\ell A S_\ell A w_\ell} (A x^* - A x_\ell) \\ &= \left(I_d - \frac{S_\ell A' w_\ell w'_\ell A}{w'_\ell A S_\ell A w_\ell} \right) (x_\ell - x^*). \end{aligned} \tag{3.7}$$

Using the induction hypothesis, $x_\ell - x^* = S_\ell(x_0 - x^*)$, and (2.2b),

$$x_{\ell+1} - x^* = \left(I_d - \frac{S_\ell A' w_\ell w_\ell' A}{w_\ell' A S_\ell A w_\ell}\right) S_\ell(x_0 - x^*) = S_{\ell+1}(x_0 - x^*). \tag{3.8}$$

∎

While x_ℓ may be the closest point to the solution set in the generated subspaces, the subspaces may not account for the entire row space of A. In other words, we must account for the possibility that span $[A'w_0, \ldots, A'w_\ell]$ may not eventually converge to the row space of A and, thus, fail to capture the entire solution. Moreover, since $\{w_k\}$ are random variables, we need a way of accounting for the space that is generated by $\{A'w_k\}$. This is the content of the following definitions.

First, we begin by defining the maximal possible subspace that can be generated by a random quantity $A'w$. Let $w \in \mathbb{R}^n$ be a random variable defined on a space Ω, and let

$$\begin{aligned} \mathcal{N}(w) &= \operatorname{span}\left[z \in \mathbb{R}^d : \mathbb{P}\left[z'A'w = 0\right] = 1\right], \\ \mathcal{R}(w) &= \mathcal{N}(w)^\perp, \text{ and} \\ \mathcal{V}(w) &= \mathcal{R}(w)^{\perp \operatorname{row}(A)}, \end{aligned} \tag{3.9}$$

where $\perp$ row(A) indicates that $\mathcal{V}(w)$ is the orthogonal complement of $\mathcal{R}(w)$ with respect to row(A) (i.e., $\mathcal{V}(w) \oplus \mathcal{R}(w) = \operatorname{row}(A)$). The following result characterizes $\mathcal{R}(w)$.

LEMMA 3.3. *For $\mathcal{R}(w)$ as defined in* (3.9)*, $\mathcal{R}(w)$ is the smallest subspace of $\mathbb{R}^d$ such that $\mathbb{P}\left[A'w \in \mathcal{R}(w)\right] = 1$.*

Proof. First, we verify that $\mathbb{P}\left[A'w \in \mathcal{R}(w)\right] = 1$. Suppose that $\mathbb{P}\left[A'w \in \mathcal{R}(w)\right] < 1$. Then,

$$\mathbb{P}\left[\exists z \perp \mathcal{R}(w) : z'A'w \neq 0\right] > 0. \tag{3.10}$$

However, we know that for any z such that $z \perp \mathcal{R}(w)$, $z \in \mathcal{N}(w)$ and $z'A'w = 0$ with probability one, which is a contradiction. Hence, $\mathbb{P}\left[A'w \in \mathcal{R}(w)\right] = 1$.

Now suppose there is a proper subspace of $\mathcal{R}(w)$, U, such that $\mathbb{P}\left[A'w \in U\right] = 1$. Let $U^{\perp \mathcal{R}(w)}$ denote the subspace orthogonal to U relative to $\mathcal{R}(w)$. Then, for any $z \in U^{\perp \mathcal{R}(w)}$, $\mathbb{P}\left[z'A'w = 0\right] = 1$, which implies that $U^{\perp \mathcal{R}(w)} \subset \mathcal{N}(w)$. However, since $U^{\perp \mathcal{R}(w)} \subset \mathcal{R}(w) \perp \mathcal{N}(w)$, $U^{\perp \mathcal{R}(w)} = \{0\}$. Thus, $\mathcal{R}(w)$ is the smallest subspace such that $\mathbb{P}\left[A'w \in \mathcal{R}(w)\right] = 1$. ∎

Second, we must define when the maximal possible subspace of $A'w$ can be achieved by a sequence of random variables $\{A'w_0, \ldots, A'w_\ell\}$, which may or may not be related to $A'w$. Note that, by not requiring a relationship between $\{A'w_0, \ldots, A'w_\ell\}$ and $A'w$, our next result is particularly general and applies to a variety of situations, from the case in which $\{w_\ell\}$ are independent copies of w to the case where $\{w_\ell\}$ have complex dependencies. Now, let $\{w_\ell : \ell + 1 \in \mathbb{N}\} \subset \mathbb{R}^n$ be random variables defined on Ω, and let T be a stopping time defined by

$$T = \min\{\ell \geq 0 : \operatorname{span}\left[A'w_0, \ldots, A'w_\ell\right] \supset \mathcal{R}(w)\}.^8 \tag{3.11}$$

Finally, let P_W denote the orthogonal projection matrix onto a subspace $W \subset \mathbb{R}^d$. Using this notation, we have the following fundamental characterization result of the limit points of $\{x_\ell\}$.

[8] Below we will assume that $A'w \in \mathcal{R}(w)$ with probability one. If we relax this, this will change the results in a predictable manner but will require additional notation. To avoid such notation, we will leave this more general case to future work if there is a sampling case that merits it.

THEOREM 3.4. *Let w be a random variable, and let $\mathcal{R}(w)$ and $\mathcal{N}(w)$ be as defined above (see* (3.9)*). Moreover, let $w_0, w_1, \ldots \in \mathbb{R}^n$ be random variables such that $\mathbb{P}[A'w_\ell \in \mathcal{R}(w)] = 1$ for all $\ell + 1 \in \mathbb{N}$, and let T be as defined in* (3.11)*. Let $x_0 \in \mathbb{R}^d$ be arbitrary and $S_0 = I_d$, and let $\{x_\ell : \ell \in \mathbb{N}\}$ and $\{S_\ell : \ell \in \mathbb{N}\}$ be defined as in* (2.2b) *and* (2.3)*. On the event $\{T < \infty\}$, the following hold:*

1. *For any $s \geq T + 1$, $S_{T+1} = S_s$ and $x_{T+1} = x_s$.*
2. *If $Ax = b$ admits a solution x^* (not necessarily unique), then*

$$x_{T+1} = P_{\mathcal{N}(w)} x_0 + P_{\mathcal{R}(w)} x^*. \tag{3.12}$$

Proof. Recall that $\mathcal{R}_{\ell+1} = \operatorname{span}[A'w_0, \ldots, A'w_\ell]$. Therefore, by the definition of T, $\mathcal{R}_{T+1} = \mathcal{R}(w)$ on the event that $\{T < \infty\}$. Therefore, by Lemma 3.1, S_{T+1} is an orthogonal projection onto $\mathcal{N}(w)$, and its null space is $\mathcal{R}(w)$.

We now proceed by induction. Because $\ker(S_{T+1}) = \mathcal{R}(w)$ and $A'w_{T+1} \in \mathcal{R}(w)$ with probability one (by hypothesis), $S_{T+1}A'w_{T+1} = 0$. Therefore, by the recursion equations, (2.2b) and (2.3), $S_{T+2} = S_{T+1}$ and $x_{T+2} = x_{T+1}$. Suppose now that $S_{T+\ell} = S_{T+1}$ and $x_{T+\ell} = x_{T+1}$ for $\ell > 1$. Again, by hypothesis, $A'w_{T+\ell} \in \mathcal{R}(w) = \ker(S_{T+\ell})$. Therefore, $S_{T+\ell}A'w_{T+\ell} = 0$. By the recursion equations, (2.2b) and (2.3), $S_{T+\ell+1} = S_{T+\ell} = S_{T+1}$ and $x_{T+\ell+1} = x_{T+\ell} = x_{T+1}$.

For the second part of the result, note that S_{T+1} is a projection onto $\mathcal{N}(w)$ (i.e., $P_{\mathcal{N}(w)} = S_{T+1}$). Therefore, on the event $\{T < \infty\}$, by Lemma 3.2,

$$\begin{aligned} x_{T+1} &= x^* + S_{T+1}(x_0 - x^*) \\ &= \left(P_{\mathcal{N}(w)} + P_{\mathcal{R}(w)}\right) x^* + P_{\mathcal{N}(w)} x_0 - P_{\mathcal{N}(w)} x^* \\ &= P_{\mathcal{R}(w)} x^* + P_{\mathcal{N}(w)} x_0. \end{aligned} \tag{3.13}$$

∎

With Theorem 3.4 in hand, the natural subsequent question is when the limit point of the iterates is actually a solution to the original system. This question is addressed in the following corollary.

COROLLARY 3.5. *Under the setting of Theorem* 3.4*, on the event $\{T < \infty\}$, $Ax_{T+1} = b$ if and only if $P_{\mathcal{V}(w)} x_0 = P_{\mathcal{V}(w)} x^*$.*

Proof. Recall that $\operatorname{row}(A) \perp \ker(A)$. Because $\mathcal{R}(w) \subset \operatorname{row}(A)$, $\mathcal{N}(w) = \mathcal{V}(w) + \ker(A)$. Moreover, by the definition of $\mathcal{V}(w) \subset \operatorname{row}(A)$, $\mathcal{V}(w) \perp \ker(A)$. Therefore, $P_{\mathcal{N}(w)} = P_{\ker(A)} + P_{\mathcal{V}(w)}$. Now, using the characterization in Theorem 3.4,

$$Ax_{T+1} = AP_{\ker(A)} x_0 + AP_{\mathcal{V}(w)} x_0 + AP_{\mathcal{R}(w)} x^* = AP_{\mathcal{V}(w)} x_0 + AP_{\mathcal{R}(w)} x^*. \tag{3.14}$$

Similarly, because $I_d = P_{\ker(A)} + P_{\mathcal{V}(w)} + P_{\mathcal{R}(w)}$,

$$b = Ax^* = AP_{\ker(A)} x^* + AP_{\mathcal{V}(w)} x^* + AP_{\mathcal{R}(w)} x^* = AP_{\mathcal{V}(w)} x^* + AP_{\mathcal{R}(w)} x^*. \tag{3.15}$$

Setting these two quantities equal to each other, we conclude that $Ax_{T+1} = b$ if and only if $AP_{\mathcal{V}(w)} x^* = AP_{\mathcal{V}(w)} x_0$. Clearly, if $P_{\mathcal{V}(w)} x_0 = P_{\mathcal{V}(w)} x^*$, then $Ax_{T+1} = b$. So, what we have left to show is that $AP_{\mathcal{V}(w)} x^* = AP_{\mathcal{V}(w)} x_0$ implies $P_{\mathcal{V}(w)} x_0 = P_{\mathcal{V}(w)} x^*$.

Let $A^\dagger$ denote the Moore–Penrose pseudo-inverse of A, and recall that $A^\dagger A$ is a projection onto $\operatorname{row}(A)$. Moreover, $\operatorname{range}(P_{\mathcal{V}}) \subset \operatorname{row}(A)$. Therefore, since if $Ax_{T+1} = b$ then $AP_{\mathcal{V}(w)} x_0 = AP_{\mathcal{V}(w)} x^*$, if $Ax_{T+1} = b$ then

$$P_{\mathcal{V}(w)} x_0 = (A^\dagger A) P_{\mathcal{V}(w)} x_0 = A^\dagger (AP_{\mathcal{V}(w)} x_0) = A^\dagger AP_{\mathcal{V}(w)} x^* = P_{\mathcal{V}(w)} x^*. \tag{3.16}$$

∎

Corollary 3.5 provides criteria on the initial condition and on $\mathcal{V}(w)$ to determine when our procedure will solve the linear system. However, we would rarely have a way of choosing the initial condition a priori such that the requirement of Corollary 3.5 holds. Thus, the alternative is to design w so that $\mathcal{V}(w) = \{0\}$, which would guarantee that $Ax_{T+1} = b$ on the event $\{T < \infty\}$. It is worth reiterating that we have made very limited assumptions about the relationships between w and $\{w_\ell\}$ and amongst $\{w_\ell\}$. This is important because it allows us to apply the preceding results to a variety of common relationship patterns between w and $\{w_\ell\}$. In the next subsection, we explore some specific relationships and whether these relationships will result in $\mathcal{V}(w) = \{0\}$.

3.2. Common sampling patterns. Theorem 3.4 supplies a general result about the behavior of *any* sampling methodology on the solution of the system using (2.2b) and (2.3), yet it does not suggest a precise sampling methodology. Generally, the sampling methodology choice will depend on both the hardware environment and the nature of the problem. For example, a random permutation sampling methodology will limit the parallelism achievable in Algorithm SM1.4. On the other hand, a random permutation sampling methodology might be well advised in a sequential setting where very little is known about the coefficient matrix A. Thus, the precise sampling scheme should depend on the hardware environment and should exploit the structure of the problem.

Despite this, in practice, there are two general sampling schemes that form a basis for more problem and hardware specific sampling schemes: random permutation sampling and i.i.d. sampling. The former sampling pattern is exemplified by randomly permuting the equations of the linear system. More concretely, let $e_1, \dots, e_n \in \mathbb{R}^n$ be the standard basis; let w be a random variable with nonzero probability on each element of the basis; let $\{w_\ell\}$ be random variables sampled from $\{e_1, \dots, e_n\}$ without replacement (until the set is exhausted; then we repopulate the set with its original elements and repeat the sampling without replacement). The following statement provides a simple characterization of this sampling scheme.

PROPOSITION 3.6. *Let $\{W_1, \dots, W_N\} \subset \mathbb{R}^n$. Let w be a random variable such that*

$$\mathbb{P}[w = W_j] > 0, \quad j = 1, \dots, N, \quad \textit{and} \quad \sum_{j=1}^{N} \mathbb{P}[w = W_j] = 1. \tag{3.17}$$

Moreover, let $\{w_\ell : \ell + 1 \in \mathbb{N}\}$ be random variables sampled from $\{W_1, \dots, W_N\}$ without replacement (and once the set is exhausted, we repopulate the set with its original elements and repeat sampling without replacement). Then $T \leq N - 1$. Moreover, $Ax_{T+1} = b$ for every initialization if $\operatorname{span}[A'W_1, \dots, A'W_N] = \operatorname{row}(A)$, which holds if $\operatorname{span}[W_1, \dots, W_N] = \mathbb{R}^n$.

Proof. First, note that $\mathcal{N}(w) = \{z \in \mathbb{R}^d : z'A'W_j = 0 \ \forall j = 1, \dots, N\}$. Therefore,

$$\mathcal{R}(w) = \mathcal{N}(w)^\perp = \operatorname{span}[A'W_1, \dots, A'W_N]. \tag{3.18}$$

In turn, because $\{w_0, \dots, w_{N-1}\} = \{W_1, \dots, W_N\}$, T is at most $N - 1$.

By Corollary 3.5, $Ax_{T+1} = b$ if and only if $P_{\mathcal{V}(w)}x_0 = P_{\mathcal{V}(w)}x^*$ where x^* satisfies $Ax^* = b$. Given that $\mathcal{R}(w) + \mathcal{V}(w) = \operatorname{row}(A)$ and $\mathcal{R}(w) = \operatorname{span}[A'W_1, \dots, A'W_N]$, if $\operatorname{span}[A'W_1, \dots, A'W_N] = \operatorname{row}(A)$, then $\mathcal{V}(w) = \{0\}$. Therefore, $Ax_{T+1} = b$ for any initialization. The final claim is straightforward. ∎

The second sampling scheme, i.i.d. sampling, is exemplified by randomly sampling equations from the system with uniform discrete probability. However, we do not need to limit ourselves to sampling from a finite population of elements. As the next result shows, we can do much more.

PROPOSITION 3.7. *Suppose that $w, w_0, w_1, \ldots$ are i.i.d. random variables. There exists a $\pi \in (0,1)$ such that*

$$\inf_{\substack{z \in \mathcal{R}(w) \\ \|z\|_2 = 1}} \mathbb{P}\left[z' A' w \neq 0\right] \geq \pi. \tag{3.19}$$

Moreover, $T < \infty$ and $\mathbb{P}[T = \ell] \leq (\ell - r)^{r-1}(1-\pi)^{\ell - r}$, where $r = \dim(\mathcal{R}(w))$ and $\ell \geq r$.

Proof. First, we show that there exists $\pi > 0$ such that for any nontrivial proper subspace $V \subsetneq \mathcal{R}(w)$, $\mathbb{P}[A'w \notin V] \geq \pi$, which implies (3.19) when we take V to be the relative orthogonal compliment to the span of a unit vector $v \in \mathcal{R}(w)$. Suppose there is no such π. Then, for every $p \in (0,1)$, there is a nontrivial subspace $V \subsetneq \mathcal{R}(w)$ such that $\mathbb{P}\left[A'w \in V\right] \geq 1 - p$. Let r be the smallest integer between 0 and $\dim(\mathcal{R}(w))$ such that

$$\sup_{\substack{V \subsetneq \mathcal{R}(w) \\ \dim[V] = r}} \mathbb{P}[A'w \in V] = 1. \tag{3.20}$$

For $\epsilon > 0$, let $V_1 \subsetneq \mathcal{R}(w)$ be an r-dimensional subspace with $\mathbb{P}[A'w \in V_1] \geq 1 - \epsilon/2$. Note that, by Lemma 3.3, $\mathbb{P}[A'w \in V_1] < 1$. Therefore, let $V_2 \subsetneq \mathcal{R}(w)$ be an r-dimensional subspace with $\mathbb{P}[A'w \in V_2] > \mathbb{P}[A'w \in V_1] \geq 1 - \epsilon/2$. Given that V_1 and V_2 are distinct and the inclusion-exclusion principle,

$$\mathbb{P}[A'w \in V_1 \cap V_2] \geq \mathbb{P}[A'w \in V_1] + \mathbb{P}[A'w \in V_2] - 1 \geq 1 - \epsilon. \tag{3.21}$$

However, this contradicts the minimality of r since $\epsilon > 0$ is arbitrary and $\dim(V_1 \cap V_2) < r$. Thus, we conclude that such a π exists.

It follow from (3.19) that for any ℓ,

$$\mathbb{P}\left[\dim(\operatorname{span}\left[A'w_0, \ldots, A'w_\ell\right]) > \dim(\operatorname{span}\left[A'w_0, \ldots, A'w_{\ell-1}\right])\right] \geq \pi. \tag{3.22}$$

Therefore, we can bound $\mathbb{P}[T = \ell]$ by a negative binomial distribution. In particular,

$$\mathbb{P}[T = \ell] \leq \binom{\ell - 1}{r - 1} (1-\pi)^{\ell - r} \leq (\ell - r)^{r-1}(1-\pi)^{\ell - r}. \tag{3.23}$$

∎

In light of the two preceding results, we may be convinced that there is a gap between the convergence properties between random permutation sampling and the i.i.d. sampling. However, by modifying the structure of the rank-one RPM, we can find more intermediate cases. The next result demonstrates this behavior with a somewhat contrived example, and we will leave more complex cases to future work.

THEOREM 3.8. *Suppose $w, w_0, w_1, \ldots$ are i.i.d. random variables such that the entries of $A'w$ are i.i.d. sub-Gaussian random variables with mean zero and unit variance. Then, there exists a $\pi \in (0,1)$ depending only on the distribution of the entries of $A'w$, such that $\mathbb{P}\left[T = \ell\right] \geq 1 - \pi^\ell$ for $\ell \geq d$.*

Proof. Let H_ℓ denote an $\ell \times d$ ($\ell \geq d$) random matrix whose entries are i.i.d. sub-Gaussian random variables with zero mean and unit variance. As a consequence of [33, Theorem 1.1], there exists a π that depends on the distribution of the entries such that for all $\ell \geq d$, $\mathbb{P}[\sigma_{\min}(H_\ell) > 0] \geq 1 - \pi^\ell$. At iteration ℓ, let N_ℓ denote the matrix whose rows are given by $w_0, w_1, \ldots$. Then, by hypothesis, $N_\ell A$ has entries that are i.i.d. sub-Gaussian random with zero mean and unit variance. Therefore, there exists a $\pi \in (0,1)$ depending only on the distribution of the entries in $A'w$ such that $\mathbb{P}[T = \ell] = \mathbb{P}[\sigma_{\min}(N_\ell A) > 0] \geq 1 - \pi^\ell$ for $\ell \geq d$. ∎

4. Convergence theory for base methods. In the previous section, we proved convergence for the complete orthogonalization method (i.e., Algorithm 2.1) and explored some specific sampling patterns. Here, we will consider the extreme opposite of the complete orthogonalization method: the "base" randomized iterative approach (e.g., randomized Kaczmarz). That is, we consider when V_k is a rank-one matrix of one of two general classes.

In the first class, we consider Algorithm SM1.1 in the case $m = 0$. In this case, (2.3) supplies the simplified iteration scheme,

$$x_{k+1} = x_k + \frac{A'w_k w_k'(b - Ax_k)}{\|A'w_k\|_2^2}, \tag{4.1}$$

which encompasses randomized Kaczmarz if we choose w_k to be independent, random draws of the basis vectors in $\mathbb{R}^n$ with the probabilities proportional to the squared row norms as specified in subsection SM1.1.

Unfortunately, (4.1) would not include randomized Gauss–Seidel. This motivates the second class, which has the closely related iteration

$$x_{k+1} = x_k + \frac{w_k w_k' A'(b - Ax_k)}{\|Aw_k\|_2^2}. \tag{4.2}$$

In this class, we recover randomized Gauss–Seidel if we choose w_k to be a random draw of the basis vector in $\mathbb{R}^d$ with the probabilities proportional to the squared column norms as specified in subsection SM1.1.

While these two classes are distinct, we will see that their analyses are nearly identical. Specifically, when we move from the analysis of row-action methods to that of column-action methods, we will see that the errors, $x_k - x^*$, in the analysis of row-action methods will be replaced by the residuals, $r_k = Ax_k - b$, in the analysis of column-action methods; moreover, we will have to replace $A'w_k$ in the analysis of row-action methods with Aw_k for the analysis of column-action methods. Otherwise, the analyses will proceed almost identically. Owing to this, we will leave the analysis of column-action methods to section SM3.

Our analysis offers two highlights: (1) we can prove convergence with probability one for arbitrary sampling schemes—only the i.i.d. case is considered in [41, 16, 32]; and (2) we can provide rates of convergence with probability one, which complements the mean-squared error results of [41, 16, 32].

Our main approach is an extension of Meany's inequality combined with stopping time arguments. We will first state the extension and then describe how it will be used. Note that, owing to the extension's similarity to Meany's original proof, the proof is left to section SM2.

THEOREM 4.1. *Let $z_1, \ldots, z_k$ be unit vectors in $\mathbb{R}^n$ for some $k \in \mathbb{N}$. Let $S = \operatorname{span}[z_1, \ldots, z_k]$. Let $\mathcal{F}$ denote all matrices F where the columns of F are the vectors*

$\{f_1, \ldots, f_r\} \subset \{z_1, \ldots, z_k\}$ that are a maximal linearly independent subset. Then

$$\sup_{y \in S, \|y\|_2 = 1} \|Qy\|_2 \leq \sqrt{1 - \min_{F \in \mathcal{F}} \det(F'F)}, \tag{4.3}$$

where $Q = (I - z_k z_k')(I - z_{k-1} z_{k-1}') \cdots (I - z_1 z_1')$.

To preview how we will use this extension of Meany's inequality, we focus on the case (4.1). Suppose, further, that $x_0 - x^* \in \text{row}(A)$, where x^* is a solution to $Ax = b$. By (4.1), we see that $x_k - x^*$ is related to $x_0 - x^*$ by rank-one perturbations of the form in Theorem 4.1 with $z_j = A'w_j$ for $j \leq k$. Thus, if $x_0 - x^*$ is in the subspace spanned by $\{A'w_0, \ldots, A'w_k\}$, then Theorem 4.1 guarantees that $\|x_k - x^*\|_2$ is less than $\|x_0 - x^*\|_2$ by a factor less than one. Roughly, $x_0 - x^*$ may not fall into this subspace, but a relevant portion of it might, and the iterates at which this relevant portion lies in the desired subspace will be stopping times. Thus, at these stopping times, we will be guaranteed improvements in the error.

In subsection 4.1, we will begin by proving the convergence of methods in the family of row-action methods specified in (4.1). We then explore some common non-adaptive sampling patterns in subsection 4.2. In subsection 4.3, we develop a general framework for the analysis of a broad class of adaptive sampling schemes and provide concrete examples from the literature.

4.1. Main convergence result for row-action methods. Recall that $w \in \mathbb{R}^n$ is a random variable, and $\{w_\ell : \ell + 1 \in \mathbb{N}\}$ is a sequence of random variables taking value in $\mathbb{R}^n$ chosen such that $A'w_\ell \in \mathcal{R}(w)$.[9] We will now define a sequence of stopping times $\{\tau_\ell : \ell + 1 \in \mathbb{N}\}$ where $\tau_0 = 0$,

$$\tau_1 = \min\{k \geq 0 : \text{span}\,[A'w_0, \ldots, A'w_k] = \mathcal{R}(w)\}, \tag{4.4}$$

and, if $\tau_{\ell-1} < \infty$, we define

$$\tau_\ell = \min\{k > \tau_{\ell-1} : \text{span}\left[A'w_{\tau_{\ell-1}+1}, \ldots, A'w_k\right] = \mathcal{R}(w)\}; \tag{4.5}$$

else $\tau_\ell = \infty$. As an aside, it is worthwhile to note the commonalities between the definition of $\{\tau_\ell\}$ and the stopping time T from (3.11).

Moreover, whenever the stopping times are finite, we will define the collection, $\mathcal{F}_\ell$, for $\ell \in \mathbb{N}$, that contains all matrices F whose columns are well defined (i.e., ignore zero vectors), maximal linearly independent subsets of

$$\left\{ \frac{A'w_{\tau_{\ell-1}+1}}{\left\|A'w_{\tau_{\ell-1}+1}\right\|_2}, \ldots, \frac{A'w_{\tau_\ell}}{\|A'w_{\tau_\ell}\|_2} \right\}. \tag{4.6}$$

Moreover, define

$$\gamma_\ell = 1 - \min_{F \in \mathcal{F}_l} \det(F'F). \tag{4.7}$$

Note that it follows by Hadamard's inequality that $\gamma_\ell \in [0, 1)$.

[9] Again, we can avoid this requirement and consider set inclusions below. However, this generalization will require additional, cumbersome notation, and there is no practical reason for considering this case.

THEOREM 4.2. *Suppose $Ax = b$ admits a solution x^* (not necessarily unique). Let w be a random variable valued in $\mathbb{R}^n$, and let $\mathcal{R}(w)$, $\mathcal{N}(w)$, and $\mathcal{V}(w)$ be defined as above (see (3.9)). Moreover, let $\{w_\ell : \ell + 1 \in \mathbb{N}\}$ be random variables such that $\mathbb{P}[A'w_\ell \in \mathcal{R}(w)] = 1$ for all $\ell + 1 \in \mathbb{N}$. Let $x_0 \in \mathbb{R}^d$ be arbitrary, and let $\{x_k : k \in \mathbb{N}\}$ be defined as in (4.1). Then, for any ℓ, on the event $\{\tau_\ell < \infty\}$,*

$$\left\|x_{\tau_\ell+1} - x^* - P_{\mathcal{N}(w)}(x_0 - x^*)\right\|_2^2 \leq \left(\prod_{j=1}^{\ell} \gamma_j\right) \left\|P_{\mathcal{R}(w)}(x_0 - x^*)\right\|_2^2, \tag{4.8}$$

where γ_j are defined in (4.7) and $\gamma_j \in [0, 1)$. Therefore, for any k,

$$\left\|x_k - x^* - P_{\mathcal{N}(w)}(x_0 - x^*)\right\|_2^2 \leq \left(\prod_{j=1}^{L(k)} \gamma_j\right) \left\|P_{\mathcal{R}(w)}(x_0 - x^*)\right\|_2^2, \tag{4.9}$$

where $L(k) = \max\{\ell : k \geq \tau_\ell + 1\}$; and where we are on the event $\{\tau_{L(k)} < \infty\}$.

Proof. From the basic iteration stated in (4.1), we have

$$x_{k+1} - x^* = x_k - x^* - \frac{A'w_k w_k' A}{\|A'w_k\|_2^2}(x_k - x^*) = \left(I - \frac{A'w_k w_k' A}{\|A'w_k\|_2^2}\right)(x_k - x^*). \tag{4.10}$$

Iterating on this relationship, we conclude that

$$x_{k+1} - x^* = \left(I - \frac{A'w_k w_k' A}{\|A'w_k\|_2^2}\right) \cdots \left(I - \frac{A'w_0 w_0' A}{\|A'w_0\|_2^2}\right)(x_0 - x^*). \tag{4.11}$$

Moreover, by assumption, $A'w_\ell \in \mathcal{R}(w)$ with probability one, which implies that $A'w_\ell \perp \mathcal{N}(w)$. Therefore,

$$x_{k+1} - x^* = P_{\mathcal{N}(w)}(x_0 - x^*) + \left(I - \frac{A'w_k w_k' A}{\|A'w_k\|_2^2}\right) \cdots \left(I - \frac{A'w_0 w_0' A}{\|A'w_0\|_2^2}\right) P_{\mathcal{R}(w)}(x_0 - x^*), \tag{4.12}$$

and $P_{\mathcal{N}(w)}(x_k - x^*) = P_{\mathcal{N}(w)}(x_0 - x^*)$.

Note that, when τ_1 is finite, the span of $\{A'w_0, \ldots, A'w_{\tau_1}\}$ is $\mathcal{R}(w)$. Therefore, on the event $\tau_1 < \infty$, Theorem 4.1 implies that

$$\left\|x_{\tau_1+1} - x^* - P_{\mathcal{N}(w)}(x_0 - x^*)\right\|_2^2 \leq \gamma_1 \left\|P_{\mathcal{R}(w)}(x_0 - x^*)\right\|_2^2. \tag{4.13}$$

We now proceed by induction. Suppose (4.8) holds for some $\ell \in \mathbb{N}$. Using (4.12), for $k > \tau_\ell$,

$$\begin{aligned} &x_k - x^* - P_{\mathcal{N}(w)}(x_0 - x^*) \\ &\quad = \left(I - \frac{A'w_k w_k' A}{\|A'w_k\|_2^2}\right) \cdots \left(I - \frac{A'w_{\tau_\ell+1} w_{\tau_\ell+1}' A}{\|A'w_{\tau_\ell+1}\|_2^2}\right) P_{\mathcal{R}(w)}(x_{\tau_\ell+1} - x^*). \end{aligned} \tag{4.14}$$

Now, when $k = \tau_{\ell+1} + 1$, the conditions of Theorem 4.1 are satisfied. Therefore,

$$\begin{aligned} \left\|x_{\tau_{\ell+1}+1} - x^* - P_{\mathcal{N}(w)}(x_0 - x^*)\right\|_2^2 &\leq \gamma_{\ell+1} \left\|P_{\mathcal{R}(w)}(x_{\tau_\ell+1} - x^*)\right\|_2^2 \\ &= \gamma_{\ell+1} \left\|x_{\tau_\ell+1} - x^* - P_{\mathcal{N}(w)}(x_0 - x^*)\right\|_2^2. \end{aligned} \tag{4.15}$$

By applying the induction hypothesis, we conclude that (4.8) holds on the event $\{\tau_{\ell+1} < \infty\}$.

Now, for an orthogonal projection matrix, $I - vv'$, $\|I - vv'\|_2 = 1$. The bound on $x_k - x^* - P_{\mathcal{N}}(x_0 - x^*)$ follows by applying this fact and the definition of $L(k)$. ∎

As an analogue of Corollary 3.5, we have the following characterization of whether $\lim_{k\to\infty} x_k$ solves the system $Ax = b$.

COROLLARY 4.3. *Under the setting of Theorem* 4.2*, on the events* $\bigcap_{\ell=0}^{\infty}\{\tau_\ell < \infty\}$ *and* $\{\lim_{\ell\to\infty}\prod_{j=0}^{\ell}\gamma_j = 0\}$, $\lim_{k\to\infty} Ax_k = b$ *if and only if* $P_{\mathcal{V}(w)}x_0 = P_{\mathcal{V}(w)}x^*$.

Proof. By Theorem 4.2, and on the events $\bigcap_{\ell=0}^{\infty}\{\tau_\ell < \infty\}$ and $\{\lim_{\ell\to\infty}\prod_{j=1}^{\ell}\gamma_j = 0\}$,

$$\lim_{k\to\infty} x_k = x^* + P_{\mathcal{N}(w)}(x_0 - x^*) = x^* + P_{\ker(A)}(x_0 - x^*) + P_{\mathcal{V}(w)}(x_0 - x^*). \tag{4.16}$$

Therefore, $\lim_{k\to\infty} Ax_k = b + AP_{\mathcal{V}(w)}(x_0 - x^*)$, which implies $\lim_{k\to\infty} Ax_k = b$ if and only if $AP_{\mathcal{V}(w)}x_0 = AP_{\mathcal{V}(w)}x^*$. Clearly, if $P_{\mathcal{V}(w)}x_0 = P_{\mathcal{V}(w)}x^*$, then $AP_{\mathcal{V}(w)}x_0 = AP_{\mathcal{V}(w)}x^*$. Now, since $\mathcal{V}(w) \subset \mathrm{row}(A)$, if $AP_{\mathcal{V}(w)}x_0 = AP_{\mathcal{V}(w)}x^*$, then $P_{\mathcal{V}(w)}x_0 = P_{\mathcal{V}(w)}x^*$ follows from (3.16). ∎

Remark 4.4. We have not explicitly accounted for finite termination in the above discussion. Given the generality of $\{w_\ell\}$, it is possible that the procedure terminates at a nonsolution (i.e., $w_j'A(x_k - x^*) = 0$ for all $j \geq k$, but $P_{\mathcal{R}(w)}(x_\ell - x^*) \neq 0$), or it is possible that the procedure terminates at a solution. We can preclude the former case by the assumptions prescribed in subsection 4.3. In the latter case, if we denote the termination point as τ, we can account for finite termination by redefining $\tau_{L(\tau)+1} = \tau$ and $\gamma_{L(\tau)+1} = 0$.

4.2. Common nonadaptive sampling patterns. Just as for Theorem 3.4, Theorems SM3.1 and 4.2 are general results that characterize convergence for *any* sampling scheme. Following the discussion in subsection 3.2, the sampling scheme should depend on the hardware environment and the problem setting. Despite this, the two sampling patterns studied in subsection 3.2 form a foundation for most sampling schemes in practice and warrant a precise analysis. After this analysis, certain adaptive schemes have become popular and are also analyzed in a generic manner. We will focus on the case of row-action methods (corresponding to Theorem 4.2) as the column-action results (corresponding to Theorem SM3.1) are nearly identical.

The first result provides a proof of convergence when we sample without replacement from a finite population. We note that the result is quite general and does not depend on the nature of the sampling without replacement or the dependency of the samples whenever the finite population is exhausted. As a result, the bounds are loose, which may be unsatisfying. Should particular sampling patterns become sufficiently important to warrant a more detailed analysis, we will do so in future work.

PROPOSITION 4.5. *Let* w *and* $\{w_\ell : \ell + 1 \in \mathbb{N}\}$ *be defined as in Proposition* 3.6*. Then, under the setting of Theorem* 4.2*,*

1. $\tau_\ell - \tau_{\ell-1} \leq 2N$ *for all* $\ell \in \mathbb{N}$*, and*
2. $\lim_{\ell\to\infty}\prod_{j=1}^{\ell}\gamma_j = 0$.

Moreover, γ_j *are uniformly bounded by* $\gamma \in [0, 1)$*, where*

$$\gamma = 1 - \min_{F\in\mathcal{F}} \det(F'F), \tag{4.17}$$

and $\mathcal{F}$ is the set of all matrices whose columns are maximal linearly independent subsets of

$$\left\{ \frac{A'W_1}{\|A'W_1\|_2}, \dots, \frac{A'W_N}{\|A'W_N\|_2} \right\}. \tag{4.18}$$

Therefore, with probability one,

$$\left\| x_{2N\ell} - x^* - P_{\mathcal{N}(w)}(x_0 - x^*) \right\|_2^2 \leq \gamma^\ell \left\| P_{\mathcal{R}(w)}(x_0 - x^*) \right\|_2^2. \tag{4.19}$$

Proof. By the definition of w in Proposition 3.6, $\mathcal{R}(w) = \operatorname{span}[A'W_1, \dots, A'W_N]$. Moreover, by the definitions of $\{w_\ell\}$, we are sampling from $W_1, \dots, W_N$ without replacement. Then, we are guaranteed that $\{A'w_{\tau_{\ell-1}+1}, \dots, A'w_{\tau_\ell}\}$ spans $\mathcal{R}(w)$ if $\{W_1, \dots, W_N\} \subset \{w_{\tau_{\ell-1}+1}, \dots, w_{\tau_\ell}\}$. Now, suppose that at iteration $\tau_{\ell-1}$, $\mathcal{W} \subset \{W_1, \dots, W_N\}$ are exhausted. Then, to ensure that $\{W_1, \dots, W_N\}$ is contained in $\{w_{\tau_{\ell-1}+1}, \dots, w_{\tau_\ell}\}$, we need to exhaust $\mathcal{W}^c$ and then the entire set $\{W_1, \dots, W_N\}$. Since $|\mathcal{W}^c| \leq N$, we need at most $2N$ more iterations from $\tau_{\ell-1}$ to achieve τ_ℓ. Therefore, $\tau_\ell - \tau_{\ell-1} \leq 2N$.

Note that $\mathcal{F}_\ell \subset \mathcal{F}$. Therefore, $\gamma_j \leq \gamma$. Moreover, by Hadamard's inequality, $\gamma \in [0, 1)$. Hence, $\lim_{\ell\to\infty} \prod_{j=1}^{\ell} \gamma_j \leq \lim_{\ell\to\infty} \gamma^\ell = 0$. The result follows by Theorem 4.2. ∎

It is worth pausing here to compare our approach in Proposition 4.5 to previous results for cyclic row-action methods (e.g., cyclic Kaczmarz [21],[10] algebraic reconstruction technique [14], cyclic block Kaczmarz). Our use of Meany's inequality to analyze such methods is not novel: Meany's inequality has been used previously to analyze deterministic row-action methods [11, 2, 37] with even more sophisticated refinements of Meany's inequality than what we have here, and a detailed comparison of Meany's inequality and other approaches to analyzing these deterministic variants can be found in [8]. However, our use of Meany's inequality generalizes these deterministic approaches as it (1) allows for an arbitrary transformation (via $\{W_1, \dots, W_N\}$) of the original system, which has borne out to be a fruitful approach vis-à-vis matrix sketching [38]; and (2) allows for random cyclic sampling, which many have observed to be the most productive route in practice, and there is mounting theoretical evidence in adjacent fields that random cyclic sampling does indeed have practical benefits [23, 39]. While our generalizations are valuable, further improvements are to be found by marrying our randomization framework with the more nuanced refinements of Meany's inequality found in [11] and [2], which we leave to future efforts.

The next result revisits the case of i.i.d. sampling. The result makes intuitive sense as, for such a situation, we should expect the difference in the stopping times to be i.i.d., which results in the natural conclusion that γ_ℓ are also i.i.d. Moreover, we show that eventually, the rate of convergence is almost controlled by $\mathbb{E}[\gamma_1]$ with probability one. We again stress here that the generality of the results naturally makes them quite loose, and we discuss this further after the result.

PROPOSITION 4.6. *Let w and $\{w_\ell : \ell + 1 \in \mathbb{N}\}$ be defined as in Proposition 3.7. Then, under the setting of Theorem 4.2, $\tau_\ell < \infty$ almost surely for all $\ell \in \mathbb{N}$, and $\{\gamma_\ell : \ell \in \mathbb{N}\}$ are i.i.d. such that $\mathbb{E}[\gamma_1] = 1 - \mathbb{E}[\min_{F\in\mathcal{F}_1} \det(F'F)] < 1$. Hence, for all*

[10] This is a translated copy of Kaczmarz's original article, which was published in German [22].

$\ell \in \mathbb{N}$ *and* $\delta > 1$,

(4.20)
$$\mathbb{P}\left[\bigcup_{j=1}^{\infty}\bigcap_{\ell=j}^{\infty}\left\{\left\|x_{\tau_\ell+1} - x^* - P_{\mathcal{N}(w)}(x_0 - x^*)\right\|_2^2 \leq \mathbb{E}\left[\gamma_1\right]^{\frac{\ell}{\delta}}\left\|P_{\mathcal{R}(w)}(x_0 - x^*)\right\|_2^2\right\}\right] = 1,$$

where $\mathbb{E}\left[\gamma_\ell\right] \in [0,1)$. *Moreover,* $\lim_{\ell\to\infty} \tau_\ell/\ell = \mathbb{E}\left[\tau_1\right]$.

Remark 4.7. In the proof below, we also compute the probability for each j for which the conclusion of the preceding result holds. Thus, we can also make the usual "high-probability" statements without any additional effort.

Proof. Again, our main workhorse will be [10, Theorem 4.1.3]. By this result, conditioned on $\tau_{\ell-1}$, $\{A'w_{\tau_{\ell-1}+1}, A'w_{\tau_{\ell-1}+2}, \ldots\}$ are i.i.d. By this property, conditioned on $\tau_{\ell-1}$, $\tau_\ell - \tau_{\ell-1}$ is independent of $\tau_{\ell-1}$ and has the same distribution for all $\ell \in \mathbb{N}$. We conclude then that since γ_ℓ is a function of $\{A'w_{\tau_{\ell-1}+1}, \ldots, A'w_{\tau_\ell}\}$, then γ_ℓ are i.i.d. We now conclude that (4.8) holds with probability one by applying Theorem 4.2. For any $\delta > 1$, by Markov's inequality and independence,

$$\mathbb{P}\left[\prod_{j=1}^{\ell}\gamma_j > \mathbb{E}\left[\gamma_1\right]^{k/\delta}\right] \leq \left(\mathbb{E}\left[\gamma_1\right]^{1-\frac{1}{\delta}}\right)^k. \tag{4.21}$$

Since $\mathbb{E}\left[\gamma_1\right]^{1-\frac{1}{\delta}} < 1$, the Borel–Cantelli lemma implies that the probability that the product of γ_j is eventually less than $\mathbb{E}\left[\gamma_1\right]^{k/\delta}$ is one. ∎

Here, we again take a moment to compare this result to the results of [32]. Namely, we are interested in how the rate of convergence of Proposition 4.6 compares with the rate-of-convergence result in [32]. To make this comparison, we numerically estimate (via simulation) the theoretical rates of convergence proposed by our result and the result of [32] on five matrices from the `MatrixDepot` (as described in section 5). We show these comparisons in Table 2. As expected, the results of [32], which are specialized to the i.i.d. case and apply on average, are much tighter than our general results that apply to more than just i.i.d. case and hold with probability one.

TABLE 2

A comparison in the estimated theoretical bounds on the rates of convergence of Gaussian-sketched base randomized methods in ℓ^2 between this work and the results in [32]. *The estimates are made by simulation of the theoretical rates. The comparison is made on five different matrices available in the* `MatrixDepot`, *as described in section* 5. *The main message is that the results of* [32] *are tighter than our result, as they apply to the average case. This is expected as our result applies to more than just the i.i.d. sampling case and hold with probability one (asymptotically).*

Comparison of Estimated Theoretical Rates of Convergence		
Matrix name	Estimated rates by result	
	Theorem 4.8 of [32]	Proposition 4.6
deriv2	$1 - \mathcal{O}\left[10^{-4}\right]$	$1 - \mathcal{O}\left[10^{-35}\right]$
heat	$1 - \mathcal{O}\left[10^{-15}\right]$	$1 - \mathcal{O}\left[10^{-34}\right]$
randsvd	$1 - \mathcal{O}\left[10^{-15}\right]$	$1 - \mathcal{O}\left[10^{-71}\right]$
ursell	$1 - \mathcal{O}\left[10^{-16}\right]$	$1 - \mathcal{O}\left[10^{-161}\right]$
wing	$1 - \mathcal{O}\left[10^{-16}\right]$	$1 - \mathcal{O}\left[10^{-163}\right]$

4.3. Adaptive sampling schemes. To bookend this section, we discuss how our results can be applied to a broad set of adaptive methods that make use of the residual information at a given iterate whether deterministically (e.g., [28, 17, 24, 4]) or randomly (e.g., [31, 3, 18]). In [15], a mean-squared-error convergence analysis is developed for some specific examples of the broad class of methods defined below. We will begin with some formalism to establish a general class of adaptive methods; we then prove convergence and a rate of convergence for such methods; finally, we provide concrete examples at the end.

To be rigorous, let $x_0 \in \mathbb{R}^d$, and let $\varphi : (A, b, \{x_j : j \leq k\}) \mapsto w_k$ be an adaptive procedure for generating $\{w_k\}$ according to the following procedure: for $k+1 \in \mathbb{N}$,

$$\begin{aligned} w_k &= \varphi(A, b, \{x_j : j \leq k\}), \\ x_{k+1} &= x_k + \frac{A' w_k w_k' (b - A x_k)}{\|A' w_k\|_2^2}. \end{aligned} \tag{4.22}$$

Remark 4.8. While we will focus on the base methods of type (4.1), methods of the type (4.2) can be handled analogously.

While (4.22) is quite general, the vast majority of adaptive schemes make further restrictions that we abstract in the following definitions.

DEFINITION 4.9 (Markovian). *For a fixed integer η, an adaptive procedure, φ, is η-Markovian if the conditional distribution of $\varphi(A, b, \{x_j : j \leq k\})$ given $\{x_j : j \leq k\}$ is equal to the conditional distribution of $\varphi(A, b, \{x_j : j \leq k\})$ given $\{x_j : k - \eta < j \leq k\}$. If a procedure is 1-Markovian, we will frequently call it Markovian.*

A consequence of the η-Markovian property is that we can write $\varphi(A, b, \{x_j : j \leq k\})$ as $\varphi(A, b, \{x_j : k - \eta < j \leq k\})$. In the case of a 1-Markovian adaptive procedure, we will simply write $\varphi(A, b, x_k)$. The 1-Markovian property is readily satisfied for a number of common procedures analyzed in the literature (e.g., maximum residual, maximum distance, etc.), which may suggest that the η-Markovian notion is irrelevant for general η. We contend, though, that procedures that are memory-sensitive may be more apt to make use of the η-Markovian property for $\eta > 1$. For example, to demonstrate its potential value, consider a procedure that selects the equations with the top η residuals, pulls them into memory, and simply cycles through them deterministically or randomly. Then this simple procedure would be η-Markovian. However, owing to the lack of such procedures in the literature, we will focus on the 1-Markovian case for which we can write $\varphi(A, b, x)$, and note that the results and definitions are readily extendable.

The next definition establishes another key property of these adaptive schemes that rely on residuals.

DEFINITION 4.10 (magnitude invariance). *Let H represent the set of solutions to $Ax = b$, and let $P_H : \mathbb{R}^d \to H$ represent the projection of a vector onto H.*[11] *Then an adaptive procedure, φ, is magnitude invariant if, for any $x \notin H$ and any $\lambda > 0$, the distribution of $\varphi(A, b, x)$ is equal to the distribution of*

$$\varphi(A, b, P_H(x) + \lambda[x - P_H(x)]). \tag{4.23}$$

The magnitude invariance of a number of adaptive methods often follows from the following simple calculation that we state as a lemma for future reference.

[11] Since H is an affine subspace, P_H is not guaranteed to be a linear operator.

LEMMA 4.11. *Let $x \in \mathbb{R}^d$, and let $v_1, v_2 \in \mathbb{R}^n$. Then, for any $\lambda > 0$, if $|v_1'(Ax - b)| \geq |v_2'(Ax - b)|$, then*

$$|v_1'(A(P_H(x) + \lambda[x - P_H(x)]) - b)| \geq |v_2'(A(P_H(x) + \lambda[x - P_H(x)]) - b)|. \tag{4.24}$$

If the hypothesis holds with a strict inequality, then so does the conclusion.

Proof. Note that $AP_H(x) = b$. Therefore, $A(P_H(x)+\lambda[x-P_H(x)])-b = \lambda(Ax-b)$. From the hypothesis and $\lambda > 0$, $\lambda|v_1'(Ax - b)| \geq \lambda|v_2'(Ax - b)|$. Also owing to $\lambda > 0$, we can replace the inequalities with strict inequalities. ∎

Furthermore, the magnitude invariance property has hidden within it an additional feature: the projection of x onto the null space is irrelevant (as we might expect for a procedure depending on the residual). As a result, we can, without losing generality, focus our discussion on x that are in the row space of A, which has a unique intersection with H at a point that we denote x^*_{row}. Furthermore, the magnitude invariance property allows us to focus specifically on the Euclidean unit sphere around x^*_{row}, to which end we define $\mathbb{S}$ as the unit ball around the zero vector. This will be essential to the *practicality* of the next definition.

The final definition ensures that if (4.22) makes too much progress along one particular subspace, then it must have a nonzero probability of exploring an orthogonal subspace relative to, roughly, the row space of A. Before stating this definition, we need to be slightly careful here with using the row space of A: if the rows of A can be partitioned into two sets that are mutually orthogonal and x_0 is initialized in the span of one of these subsets, then we will never need to visit the other set and, consequently, we will never observe the entire row space of A. To account for this, we can focus on the restricted row space,

$$\text{rrow}(A) = \text{span}[A_{i,\cdot} : A'_{i,\cdot}x_0 \neq b_i]. \tag{4.25}$$

This definition may seem unnecessary as we can account for this (more generally) via $\mathcal{R}(w)$ by an appropriate choice of w. However, in our previous statements, we defined w before specifying x_0. Here, we would need to know x_0 in order to define w and, thus, $\mathcal{R}(w)$ appropriately. Fortunately, an examination of the preceding results shows that this ordering is not important and the results hold even if w is defined given x_0 or even future iterates. With this explanation in hand, we can now state the final definition.

DEFINITION 4.12 (exploratory). *Let $x_0 \in \mathbb{R}^d$, and define* rrow(A) *accordingly. An adaptive procedure, φ, is exploratory if for any proper subspace $V \subsetneq$ rrow(A), there exists $\pi \in (0, 1]$ such that*

$$\sup_{x \in x^*_{\text{row}} + \mathbb{S} \cap V} \mathbb{P}\left[A'\varphi(A, b, x) \perp V\right] \leq 1 - \pi. \tag{4.26}$$

Remark 4.13. If magnitude invariance does not hold, then we could specify the exploratory property to hold for any point in V that is distinct from x^*_{row}. For this modified definition of the exploratory property, the results below would still hold. Then, why should we keep the magnitude invariance property? It is out of practicality. The magnitude invariance property allows us to restrict the verification of the exploratory property to the unit ball, and then we can apply it to any iterate regardless of its distance to the solution.

For a Markovian, magnitude invariant, and exploratory adaptive scheme, φ, we will need one assumption before stating the result.

ASSUMPTION 4.14. *For $k \in \mathbb{N}$, let $\mathcal{F}_k$ denote the set of matrices whose columns are normalized, maximal linearly independent subsets of*

$$\{A'\varphi(A, b, \hat{x}_1), \ldots, A'\varphi(A, b, \hat{x}_k)\}, \tag{4.27}$$

where $\hat{x}_1, \ldots, \hat{x}_k \in \mathbb{R}^d$ are arbitrary vectors. There exists $\gamma \in [0, 1)$ such that

$$\inf_{k \in \mathbb{N}} \mathbb{P}\left[1 - \inf_{F \in \mathcal{F}_k} \det(F'F) \leq \gamma\right] = 1. \tag{4.28}$$

Remark 4.15. As we will see, Assumption 4.14 is sufficient for us to uniformly treat the many examples in the literature that are selecting equations or, more generally, are of the form in Proposition 3.6, rather than generating linear combinations of them. In the case of linear combinations, we could refine this assumption to account for the nature of the linear combinations as we do in Proposition 4.6.

THEOREM 4.16. *Suppose $Ax = b$ admits a solution x^* (not necessarily unique); let H denote the set of all solutions, and let P_H be the projection onto this affine subspace. Let $x_0 \in \mathbb{R}^d$, and let* $\operatorname{rrow}(A)$ *be defined as above (see* (4.25)*). Moreover, let φ be a* 1*-Markovian, magnitude invariant, and exploratory adaptive procedure satisfying Assumption* 4.14 *that generates $\{x_k\}$ and $\{w_k\}$ according to* (4.22) *and so that $\mathbb{P}[A'w_k \in \operatorname{rrow}(A)] = 1$ for all $k + 1 \in \mathbb{N}$. Then, there exists an increasing sequence of stopping times $\{\tau_\ell : \ell \in \mathbb{N}\}$ such that $\mathbb{P}[E_1 \cup E_2] = 1$, where the following hold:*

1. *E_1 is the event of iterates that terminate finitely to a solution of $Ax = b$; that is,*

$$E_1 = \bigcup_{\ell \in \mathbb{N}} \{x_{\tau_\ell + 1} \in H\}. \tag{4.29}$$

2. *E_2 is the event of iterates that infinitely converge to a solution of $Ax = b$; that is,*

$$E_2 = \bigcap_{\ell \in \mathbb{N}} \left\{\|x_{\tau_\ell+1} - P_H(x_0)\|_2^2 \leq \gamma^\ell \|x_0 - P_H(x_0)\|_2^2\right\}. \tag{4.30}$$

Moreover, on E_1, τ_ℓ has finite expectation for ℓ such that $x_{\tau_\ell+1} \in H$. Similarly, on E_2, τ_ℓ has finite expectation for all ℓ.

Proof. Without loss of generality, we will assume $x_0 \in \operatorname{row}(A)$. We will consider the nontrivial case where $x_0 \neq x^*_{\text{row}}$. Note that, by the construction of $\operatorname{rrow}(A)$, it must hold then that $x_0 - x^*_{\text{row}} \in \operatorname{rrow}(A)$. To prove the result, we will make three claims of the following rough nature and purpose, which we will make precise below.

1. Finite termination can only occur at a point x_{k+1} if and only if $A'\varphi(A, b, x_k)$ is parallel to $x_k - x^*_{\text{row}}$. We will use this claim to specify the set E_1.

2. For the first time the span of the iterate errors, $\operatorname{span}[\{x_k - x^*_{\text{row}}\}]$, fails to (nontrivially) increase in dimension; the corresponding $\{A'w_k\}$ up to this iterate span the subspace. As a result, with an appropriate definition of $\mathcal{R}(w)$, we will apply Theorem 4.2 to prove a multiplicative decrease in the iterate errors by a factor of γ.

3. Finally, we show that the first time that the span of the iterate errors fails to (nontrivially) increase in dimension must be finite with probability one and have bounded expectation. By combining the first claim with this claim, we have the property specified by the event E_1. By combining this claim with the second claim, we have the property specified by the event E_2. By this claim alone, we have that $\mathbb{P}[E_1 \cup E_2] = 1$.

To establish our claims, we need some additional notation. Let ξ be an arbitrary finite stopping time, and define

$$\mathcal{R}_k = \operatorname{span}\left[x_\xi - x^*_{\text{row}}, x_{\xi+1} - x^*_{\text{row}}, \ldots, x_{\xi+k} - x^*_{\text{row}}\right] \tag{4.31}$$

and $\mathcal{R}_k^0 = \operatorname{span}\left[x_{\xi+k} - x^*_{\text{row}}\right]$. Furthermore, define

$$\nu = \min\left\{k \geq 0 : x_{\xi+k+1} - x^*_{\text{row}} \in \mathcal{R}_k, x_{\xi+k+1} \neq x_{\xi+k}\right\}. \tag{4.32}$$

Note that ν corresponds to the first time that the span of the iterate errors, starting at ξ, fails to nontrivially increase in dimension. It will often be more succinct to specify the nontrivial cases by an indicator variable given by

$$\chi_{\xi+k} = \mathbf{1}\left[\varphi(A, b, x_{\xi+k})' A(x_{\xi+k} - x^*_{\text{row}}) \neq 0\right]. \tag{4.33}$$

By (4.22), we can readily replace $x_{\xi+k+1} \neq x_{\xi+k}$ in the definition of ν with $\chi_{\xi+k} = 1$. We now state and prove our claims precisely.

Claim 1. Suppose $x_\xi - x^*_{\text{row}} \neq 0$. We claim that $x_{\xi+1} = x^*_{\text{row}}$ if and only if $A'\varphi(A, b, x_\xi) \in \mathcal{R}_0 \setminus \{0\}$.

Note that this claim readily follows from

$$x_{\xi+1} - x^*_{\text{row}} = x_\xi - x^*_{\text{row}} - \frac{A'\varphi(A, b, x_\xi)\varphi(A, b, x_\xi)' A}{\|A'\varphi(A, b, x_\xi)\|_2^2}(x_\xi - x^*_{\text{row}}), \tag{4.34}$$

which, in turn, follows from (4.22).

Claim 2. Suppose ν is finite, and define $\mathcal{R}_\nu$. We claim that

$$\operatorname{span}\left[A'\varphi(A, b, x_\xi)\chi_\xi, \ldots, A'\varphi(A, b, x_{\xi+\nu})\chi_{\xi+\nu}\right] = \mathcal{R}_\nu. \tag{4.35}$$

We first note that $A'\varphi(A, b, x_{\xi+k})\chi_{\xi+k} \in \mathcal{R}_\nu$ for any $k \in [0, \nu]$ by (4.22). Therefore, we see that the span of $\Phi = \{A'\varphi(A, b, x_\xi)\chi_\xi, \ldots, A'\varphi(A, b, x_{\xi+\nu})\chi_{\xi+\nu}\}$ is contained in $\mathcal{R}_\nu$. To show that $\mathcal{R}_\nu$ is included in the span of Φ, note that, by the definition of $\mathcal{R}_\nu$ and by (4.22),

$$\mathcal{R}_\nu = \operatorname{span}\left[A'\varphi(A, b, x_\xi)\chi_\xi, \ldots, A'\varphi(A, b, x_{\xi+\nu-1})\chi_{\xi+\nu-1}, x_{\xi+\nu} - x^*_{\text{row}}\right]. \tag{4.36}$$

Moreover, the nonzero terms on the generating set on the right-hand side of (4.36) must be linearly independent, as anything else would contradict the minimality of ν. We are left to show that $x_{\xi+\nu} - x^*_{\text{row}}$ is in the span of Φ. To do this, we perform Gram–Schmidt on the generating set in (4.36) starting with $x_{\xi+\nu} - x^*_{\text{row}}$. Denote the remaining vectors in this set $\phi_1, \ldots, \phi_{r-1}$ where $r = \dim(\mathcal{R}_\nu)$. Then, by the definition of ν, $x_{\xi+\nu+1} - x^*_{\text{row}} \in \mathcal{R}_\nu$. Therefore, there exist constants $c_0, \ldots, c_{r-1}$ such that

$$\begin{aligned} &c_0(x_{\xi+\nu} - x^*_{\text{row}}) + \sum_{j=1}^{r-1} c_j \phi_j \\ &\quad = x_{\xi+\nu} - x^*_{\text{row}} - \frac{A'\varphi(A, b, x_{\xi+\nu})\varphi(A, b, x_{\xi+\nu})' A}{\|A'\varphi(A, b, x_{\xi+\nu})\|_2^2}(x_{\xi+\nu} - x^*_{\text{row}}). \end{aligned} \tag{4.37}$$

If $c_0 \neq 1$, we see that the claim follows. For a contradiction, suppose that $c_0 = 1$. Then $A'\varphi(A, b, x_{\xi+\nu})$ can be written as a linear combination of vectors that are orthogonal to $x_{\xi+\nu} - x^*_{\text{row}}$. This would imply then that $\chi_{\xi+\nu} = 0$, which contradicts the definition of ν. Hence, we see that the claim holds.

Claim 3. For any finite stopping time ξ, ν is finite with probability one and has bounded expectation.

To show this, we define a sequence of stopping times. Define

$$s_1 = \min\{k : \chi_{\xi+k} \neq 0\} \tag{4.38}$$

and

$$s_j = \min\left\{k : \chi_{\xi+s_1+\cdots+s_{j-1}+k} \neq 0\right\}. \tag{4.39}$$

By the definition of ν, ν can only take values in $\{\sum_{i=1}^{j} s_i : j \in \mathbb{N}\}$. Moreover, at each s_j, we must observe that either $\{\dim(\mathcal{R}_{\xi+s_1+\cdots+s_j+1}) = \dim(\mathcal{R}_{\xi+s_1+\cdots+s_j}) + 1\}$ or $\{\nu \leq \sum_{i=1}^{j} s_i\}$. Hence, at most, we see that ν can only take values in $\{\sum_{i=1}^{j} s_i : j = 1, \ldots, r\}$ where $r = \dim(\mathrm{rrow}(A))$. Thus, if we show that each s_j is finite and has bounded expectation, then ν must be finite and have bounded expectation. By the magnitude invariance, Markovian, and exploratory properties, we conclude that

$$\begin{aligned}&\mathbb{P}\left[s_j = k | \, \xi, s_1, \ldots, s_{j-1}, x_\xi, \ldots, x_{\xi+s_1+\cdots+s_{j-1}+1}\right] \\ &\quad \leq (1 - \pi(\mathcal{R}_{s_1+\cdots+s_{j-1}+1}))^{k-1} \pi(\mathcal{R}_{s_1+\cdots+s_{j-1}+1}).\end{aligned} \tag{4.40}$$

Therefore, we see that s_j is finite and has bounded expectation.

Conclusion. From these three claims we can now prove the result by induction.

Base case. Define $\mathfrak{E}_0^c = \{x_0 \neq x_{\mathrm{row}}^*\}$. On this event, we take $\xi = 0$ and define τ_1 to be the corresponding ν. On $\mathfrak{E}_0^c$, τ_1 is finite and has finite expectation by Claim 3. Then, we can define, as a subset of $\mathfrak{E}_0^c$,

$$\mathfrak{E}_1 = \{A'\varphi(A, b, x_{\tau_1}) \in \mathcal{R}_{\tau_1}^0 \setminus \{0\}\}, \tag{4.41}$$

and $\mathfrak{E}_1^c$ to be its relative complement on $\mathfrak{E}_0$.

Note the following:

1. By Claim 1, $\mathfrak{E}_1$ is equivalent to the event $x_{\tau_1+1} = x_{\mathrm{row}}^*$ up to a measure zero set.
2. By Claim 2, Theorem 4.2 with $\mathcal{R}(w) = \mathcal{R}_{\tau_1}$, and Assumption 4.14, $\mathfrak{E}_1^c$ is contained in the event on which

$$\|x_{\tau_1+1} - x_{\mathrm{row}}^*\|_2^2 \leq \gamma \|x_0 - x_{\mathrm{row}}^*\|_2^2 \tag{4.42}$$

up to a measure zero set.

Induction hypothesis. Let $\ell \in \mathbb{N}$. On the event $\mathfrak{E}_{\ell-1}^c$, we let $\xi = \tau_{\ell-1} + 1$, and, for the correspondingly defined ν, we can define $\tau_\ell = \tau_{\ell-1} + 1 + \nu$. Furthermore, on $\mathfrak{E}_{\ell-1}^c$, τ_ℓ is finite and has finite expectation. We can define, as a subset of $\mathfrak{E}_{\ell-1}^c$,

$$\mathfrak{E}_\ell = \{A'\varphi(A, b, x_{\tau_\ell}) \in \mathcal{R}_{\tau_\ell}^0 \setminus \{0\}\}, \tag{4.43}$$

and $\mathfrak{E}_\ell^c$ to be its relative complement on $\mathfrak{E}_{\ell-1}^c$.

Further,

1. $\mathfrak{E}_\ell$ is equivalent to the event $x_{\tau_\ell+1} = x_{\mathrm{row}}^*$ up to a measure zero set;
2. $\mathfrak{E}_\ell^c$ is contained in the event on which

$$\|x_{\tau_\ell+1} - x_{\mathrm{row}}^*\|_2^2 \leq \gamma \|x_\ell - x_{\mathrm{row}}^*\|_2^2 \tag{4.44}$$

up to a measure zero set.

Generalization. On the event $\mathfrak{E}_\ell^c$, we let $\xi = \tau_\ell + 1$, and, for the correspondingly defined ν, we can define $\tau_{\ell+1} = \tau_\ell + 1 + \nu$. On $\mathfrak{E}_\ell^c$, $\tau_{\ell+1}$ is finite and has finite expectation by Claim 3. Then, we can define, as a subset of $\mathfrak{E}_\ell^c$,

$$\mathfrak{E}_{\ell+1} = \{A'\varphi(A, b, x_{\tau_{\ell+1}}) \in \mathcal{R}^0_{\tau_{\ell+1}} \setminus \{0\}\} \tag{4.45}$$

and $\mathfrak{E}_{\ell+1}^c$ to be its relative complement on $\mathfrak{E}_\ell^c$.

1. By Claim 1, $\mathfrak{E}_{\ell+1}$ is equivalent to the event $x_{\tau_{\ell+1}+1} = x^*_{\text{row}}$ up to a measure zero set.

2. By Claim 2, Theorem 4.2 with $\mathcal{R}(w) = \mathcal{R}_{\tau_{\ell+1}}$, and Assumption 4.14, $\mathfrak{E}_{\ell+1}^c$ is contained in the event on which

$$\left\|x_{\tau_{\ell+1}+1} - x^*_{\text{row}}\right\|_2^2 \leq \gamma \left\|x_{\tau_\ell} - x^*_{\text{row}}\right\|_2^2 \tag{4.46}$$

up to a measure zero set.

Therefore, by the induction claims,

$$E_1 = \bigcup_{\ell \in \mathbb{N}} \mathfrak{E}_\ell \tag{4.47}$$

and

$$E_2 = \bigcap_{\ell \in \mathbb{N}} \mathfrak{E}_\ell^c, \tag{4.48}$$

and $\mathbb{P}[E_1 \cup E_2] = 1$. ∎

To demonstrate the utility of Theorem 4.16, we show that a number of classical and recent methods satisfy Definitions 4.9, 4.10, and 4.12 and Assumption 4.14. In fact, we will show that a stronger version of Definition 4.12 holds for these methods, which allows us to explicitly upper bound the elements of $\{\mathbb{E}[\tau_\ell] : \ell \in \mathbb{N}\}$ (when they are defined). The following proposition states these examples formally, and the proof is found in section SM4.

PROPOSITION 4.17. *Suppose $Ax = b$ admits a solution x^*. Let $x_0 \in \mathbb{R}^d$, and let* row(A) *be defined as above (see* (4.25)*). Suppose that we define $\{x_k\}$ and $\{w_k\}$ according to* (4.22) *for the following adaptive methods:*

1. *the maximum residual method (see* [1, Section 4]*);*
2. *the maximum distance method (see* [1, Section 3] *and* [28]*);*
3. *the Greedy Randomized Kaczmarz method (see* [3, Method 2]*);*
4. *the Sampling Kaczmarz–Motzkin method (see* [18, Page 4]*).*

Then, for each of the above methods, there exists a $\gamma \in [0, 1)$ such that the conclusions of Theorem 4.16 *hold. Moreover, there exists a constant κ such that for any finite τ_ℓ (as specified in Theorem* 4.16*), $\mathbb{E}[\tau_\ell] \leq \ell\kappa$.*

Remark 4.18. Greedy Randomized Kaczmarz is an example of methods that deterministically determine a threshold over residuals; select the equations whose residuals surpass this threshold; and then randomly select from this set. For this more general class, so long as the threshold satisfies the magnitude invariance property and the random selection does not give any equation less than zero probability, then the result applies to this more general class. Similarly, Sampling Kaczmarz–Motzkin is an example of methods that randomly determine a set of equations and then deterministically select from this subset of equations based on the residual values. So long as the random subset of equations does not give any equation less than zero probability (that is not already satisfied), then the result will apply to this more general class as well.

Remark 4.19. Our partial orthogonalization methods (see Algorithm SM1.1) do not satisfy the η-Markovian property, as the partial orthogonalizations have a dependence on every preceding iterate.

5. Numerical experiments. Here, we present a variety of numerical experiments to study the practicality of our approach in a sequential computing environment. Specifically, we test 49 systems with 500 equations and 500 unknowns. The coefficients are generated from 49 built-in matrices found in the `MatrixDepot` package for the `Julia` programming language [40]. The solution to the equation is then generated using a standard multivariate normal vector. The constant vector is generated by the product of the two. Then, using the generated coefficient matrix and the generated constant vector, we solve the systems by varying the sample generation method (i.e., the generation of w and $\{w_\ell\}$) and the solver. The sample generation method is either produced by the Count-Sketch approach, by the Gaussian approach, by uniformly sampling the equations of the matrix with replacement, or by uniformly sampling the equations of the matrix without replacement. The solver is either a base method, the complete method, an intermediate method with $m = 5$, or an intermediate method with $m = 10$. Finally, we initialize $x_0 = 0$.

We recorded the wall clock time and number of iterations to improve the initial residual norm by a factor of ten with an upper bound of three seconds. If the temporal upper bound is reached before a tenfold improvement in the initial residual norm is observed, the wall clock time is reported as "Inf." Inherently, this metric results in a disadvantage for complete orthogonalization methods as such methods pay more for marginal improvements but generate precise solutions with fewer iterations. However, with an eye towards solving much larger systems that require using a parallel or distributed environment, this metric of time-to-tenfold improvement is the appropriate choice as the complete method would not be appropriate in such environments owing to the high communication costs that would be incurred. For the Count-Sketch sampling method, the wall clock times and iterations are reported in Tables 3 and 4. For the remaining sampling approaches, the wall clock times and iterations are reported in the appendix.

While further analysis of each system would be necessary to understand the behavior of the solvers on each system, there are several important messages within Table 3. First, the base method often fails to achieve a tenfold improvement despite the substantial number of iterations that the base solver is allowed (again, on the order of 10^6). Unfortunately, the base method's poor behavior is observed even for the other sampling methods. Based on Theorem 4.2, this would imply that either the stopping times $\{\tau_\ell\}$ are large and/or the rates of convergence (determined by $\{\gamma_\ell\}$) are too small. Given that this behavior is observed even for the random cyclic sampling case (which, by Proposition 4.5, implies that the differences between the stopping times are bounded by a thousand), it is likely that the rate of convergence for such systems is close to unity.

However, we see a tremendous benefit even from a small amount of partial orthogonalization. That is, the intermediate solvers with $m = 5$ and $m = 10$ perform quite well. In particular, whenever complete orthogonalization achieves a tenfold improvement within the allotted time, the partial orthogonalization methods also achieve the tenfold improvement within the allotted time and often orders of magnitude faster. Thus, for cases when the base method performs poorly, a small amount of partial orthogonalization is able to remedy this poor behavior. One final observation is that the $m = 5$ method often outperforms the $m = 10$ method. This seems to be because

TABLE 3
Wall clock time for $10\times$ *improvement of all solvers under Count-Sketch sampling.*

System	Base	Partial, $m = 5$	Partial, $m = 10$	Complete
baart	Inf	6.0386×10^{-5}	1.1500×10^{-4}	6.7174×10^{-3}
cauchy	Inf	1.1437×10^{-4}	1.4446×10^{-4}	6.8039×10^{-3}
chebspec	Inf	2.0174×10^{-3}	3.4821×10^{-3}	1.8816×10^{-1}
chow	Inf	8.9687×10^{-3}	1.1738×10^{-2}	1.6859×10^{-1}
circul	Inf	2.2109×10^{-1}	1.7215×10^{-1}	7.0601×10^{-1}
clement	Inf	8.4406×10^{-2}	1.2590×10^{-1}	2.0954
companion	Inf	3.2478×10^{-5}	4.6921×10^{-5}	3.8471×10^{-3}
deriv2	3.7514×10^{-1}	8.6084×10^{-5}	2.3455×10^{-4}	2.4547×10^{-2}
dingdong	Inf	6.3064×10^{-2}	1.0183×10^{-1}	1.9053
erdrey	Inf	9.8817×10^{-2}	1.8960×10^{-1}	1.6046
fiedler	Inf	8.3640×10^{-5}	1.4840×10^{-4}	1.5384×10^{-2}
forsythe	Inf	7.0330×10^{-2}	1.1496×10^{-1}	1.8313
foxgood	Inf	3.1248×10^{-5}	4.4997×10^{-5}	3.7755×10^{-3}
frank	Inf	6.7833×10^{-2}	6.1152×10^{-2}	4.1107×10^{-1}
gilbert	Inf	1.4819×10^{-1}	2.1992×10^{-1}	1.7001
golub	Inf	9.7789×10^{-2}	1.2317×10^{-1}	1.2163
gravity	Inf	1.3287×10^{-4}	2.6159×10^{-4}	2.8156×10^{-2}
grcar	Inf	8.4665×10^{-2}	1.3493×10^{-1}	1.8459
hankel	Inf	2.3273×10^{-2}	1.4042×10^{-2}	1.9699×10^{-1}
heat	5.1398×10^{-2}	7.6801×10^{-4}	3.3231×10^{-4}	4.6707×10^{-2}
hilb	Inf	7.8676×10^{-5}	1.1786×10^{-4}	6.9384×10^{-3}
invol	Inf	Inf	Inf	Inf
kahan	Inf	9.6697×10^{-3}	4.0889×10^{-3}	1.4554×10^{-1}
kms	Inf	1.6637×10^{-1}	2.5975×10^{-1}	2.1075
lehmer	Inf	3.2529×10^{-5}	5.2778×10^{-5}	3.9027×10^{-3}
lotkin	Inf	1.4320×10^{-4}	1.1696×10^{-4}	2.9118×10^{-2}
magic	Inf	6.1573×10^{-5}	9.3895×10^{-5}	6.9376×10^{-3}
minij	Inf	1.2467×10^{-4}	1.8361×10^{-4}	6.8205×10^{-3}
moler	Inf	7.8435×10^{-5}	1.1515×10^{-4}	1.4089×10^{-2}
oscillate	Inf	1.1374×10^{-1}	1.9724×10^{-1}	1.7071
parter	Inf	6.2843×10^{-2}	9.6532×10^{-2}	1.8702
pei	Inf	1.5601×10^{-3}	1.7956×10^{-3}	1.5381×10^{-1}
phillips	Inf	3.8386×10^{-4}	2.2911×10^{-4}	2.3050×10^{-2}
prolate	Inf	1.7878×10^{-4}	3.0497×10^{-4}	2.8550×10^{-2}
randcorr	Inf	1.0372×10^{-1}	1.6375×10^{-1}	1.7069
rando	Inf	3.1795×10^{-1}	2.9357×10^{-1}	1.6219
randsvd	1.8895×10^{-2}	5.0995×10^{-2}	7.4617×10^{-2}	3.8380×10^{-1}
rohess	Inf	6.8904×10^{-2}	9.2650×10^{-2}	1.7814
sampling	Inf	1.8389×10^{-1}	3.0695×10^{-1}	1.6911
shaw	Inf	1.2868×10^{-4}	1.5171×10^{-4}	1.6849×10^{-2}
smallworld	Inf	1.1797×10^{-1}	1.6365×10^{-1}	1.6696
spikes	Inf	1.4840×10^{-4}	2.0070×10^{-4}	2.0368×10^{-2}
toeplitz	Inf	1.2258×10^{-4}	2.2206×10^{-4}	2.1445×10^{-2}
tridiag	Inf	9.5842×10^{-2}	1.8465×10^{-1}	1.5066
triw	Inf	4.0887×10^{-1}	1.8399×10^{-1}	1.0456
ursell	3.2883×10^{-5}	3.5417×10^{-5}	5.3220×10^{-5}	3.4169×10^{-3}
vand	Inf	Inf	Inf	Inf
wilkinson	Inf	1.0985×10^{-1}	2.0894×10^{-1}	1.7395
wing	2.0477×10^{-5}	2.9370×10^{-5}	5.3801×10^{-5}	3.4583×10^{-3}

of the memory-management and garbage collection time related to modifying the set $\mathcal{S}$, which we did not optimize in these experiments. Thus, a more complete study would require a detailed optimization of how $\mathcal{S}$ is handled.

TABLE 4

Number of iterations for $10\times$ *improvement of all solvers under Count-Sketch sampling. For those runs that failed to achieve the improvement threshold, the number of iterations within the time limit is listed.*

System	Base	Partial, $m = 5$	Partial, $m = 10$	Complete
baart	4×10^5	3	4	3
cauchy	4×10^5	5	4	3
chebspec	4×10^5	9×10^1	9×10^1	5×10^1
chow	4×10^5	4×10^2	3×10^2	4×10^1
circul	4×10^5	8×10^3	4×10^3	2×10^2
clement	4×10^5	3×10^3	3×10^3	5×10^2
companion	4×10^5	2	2	2
deriv2	5×10^4	4	7	7
dingdong	4×10^5	3×10^3	2×10^3	5×10^2
erdrey	4×10^5	4×10^3	4×10^3	5×10^2
fiedler	4×10^5	4	5	4
forsythe	4×10^5	3×10^3	3×10^3	5×10^2
foxgood	4×10^5	2	2	2
frank	4×10^5	3×10^3	2×10^3	1×10^2
gilbert	4×10^5	6×10^3	5×10^3	5×10^2
golub	4×10^5	4×10^3	3×10^3	3×10^2
gravity	4×10^5	6	8	7
grcar	4×10^5	3×10^3	3×10^3	5×10^2
hankel	4×10^5	8×10^2	3×10^2	5×10^1
heat	8×10^3	3×10^1	1×10^1	1×10^1
hilb	4×10^5	4	4	3
invol	1	1	1	1
kahan	4×10^5	4×10^2	1×10^2	4×10^1
kms	4×10^5	7×10^3	6×10^3	5×10^2
lehmer	4×10^5	2	2	2
lotkin	4×10^5	7	4	7
magic	4×10^5	3	3	3
minij	4×10^5	6	6	3
moler	4×10^5	4	4	4
oscillate	4×10^5	4×10^3	5×10^3	5×10^2
parter	4×10^5	2×10^3	2×10^3	5×10^2
pei	4×10^5	7×10^1	5×10^1	3×10^1
phillips	4×10^5	2×10^1	7	6
prolate	4×10^5	8	9	8
randcorr	4×10^5	4×10^3	4×10^3	5×10^2
rando	4×10^5	1×10^4	8×10^3	5×10^2
randsvd	3×10^3	2×10^3	2×10^3	1×10^2
rohess	4×10^5	3×10^3	2×10^3	5×10^2
sampling	4×10^5	7×10^3	7×10^3	5×10^2
shaw	4×10^5	5	5	5
smallworld	4×10^5	5×10^3	4×10^3	5×10^2
spikes	4×10^5	7	6	6
toeplitz	4×10^5	6	7	5
tridiag	4×10^5	4×10^3	4×10^3	4×10^2
triw	4×10^5	2×10^4	5×10^3	3×10^2
ursell	4	2	2	2
vand	1	1	1	1
wilkinson	4×10^5	5×10^3	5×10^3	5×10^2
wing	3	2	2	2

6. Conclusion. To reiterate, our motivation was to address the two practical challenges of the typical sketch-*then*-solve approach for solving linear systems. These practical challenges are as follows: there is no clear way of choosing the size of the

sketching matrix a priori; and there is a nontrivial storage costs of the sketched system. We made progress towards addressing these challenges by reformulating the sketch-*then*-solve approach as a sketch-*and*-solve approach in which the sketched system is implicitly constructed and solved simultaneously. The main idea of the reformulation is to construct the equations of the sketched system one at a time and then use an orthogonalization scheme to solve the system as each sketched equation is observed. As a result, we addressed the concern of determining the sketching matrix's dimensions because, under our reformulation, the sketching matrix could be grown to an arbitrary size during the calculation up to a user-defined stopping criterion, which may be based on closeness to a solution or based on a computational budget. Moreover, we addressed the cost of storing the sketched system because we do not need to explicitly form the entire sketched system under our reformulation. However, we traded this storage problem with another one—albeit less onerous—of storing the matrix S. Finally, we address the overlooked practical challenge of solving the sketched system by using our orthogonalization scheme to solve the implicitly sketched system under our reformulation.

When d becomes very large, storing and manipulating S becomes prohibitive. Because of the challenges introduced by S, we proposed intermediate methods that implicitly stored S using only a handful of vectors. The result was a collection of partial orthogonalization schemes, and, in the limit of not storing any vectors for S (i.e., S becomes the identity), we recovered what we called "base methods," which included the important special cases of randomized Kaczmarz and randomized Gauss–Seidel. As a result, we were able to make a conceptual connection between random sketching methods (i.e., complete orthogonalization methods under our formulation) and the usual randomized iterative methods (i.e., base methods under our formulation). Importantly, we were able to leverage this conceptual relationship between the two to connect the convergence theory of the complete orthogonalization method to the convergence theory of the base methods. The key ingredient here is that the stopping time that was defined for the complete orthogonalization method encoded information about exploration of a subspace that contained the solution of the sketched system. This stopping time was then used (in a repeated fashion) to guarantee that a certain amount of progress for the base methods is achieved. As a result, we were able to produce a convergence theory for these base methods that both was quite general and complemented and improved on previous results. In fact, we were able to use this theory to prove convergence for a broad class of adaptive sampling methods, and supply rates of convergence.

The predominant missing component of this work is the rigorous analysis of the so-called intermediate methods that reside between the base methods and the complete methods. Such an analysis is certainly warranted owing to the impressive numerical performance of these intermediate methods as demonstrated in our experiments. Owing, primarily, to the additional complexity of analyzing such intermediate methods and, secondarily, of space limitations, a rigorous analysis of these methods will be the focus of future work. Additionally, an efficient implementation at scale for challenging problems arising in partial differential equations with a detailed comparison to existing state-of-the-art methods will be included in future work.

Acknowledgment. We would like to thank the reviewers for their valuable comments, which have tremendously improved the content and quality of this paper.

REFERENCES

[1] S. Agmon, *The relaxation method for linear inequalities*, Canad. J. Math., 6 (1954), pp. 382–392.

[2] Z.-Z. Bai and X.-G. Liu, *On the Meany inequality with applications to convergence analysis of several row-action iteration methods*, Numer. Math., 124 (2013), pp. 215–236.

[3] Z.-Z. Bai and W.-T. Wu, *On greedy randomized Kaczmarz method for solving large sparse linear systems*, SIAM J. Sci. Comput., 40 (2018), pp. A592–A606, https://doi.org/10.1137/17M1137747.

[4] Y. Censor, *Row-action methods for huge and sparse systems and their applications*, SIAM Rev., 23 (1981), pp. 444–466, https://doi.org/10.1137/1023097.

[5] X. Chen and A. M. Powell, *Almost sure convergence of the Kaczmarz algorithm with random measurements*, J. Fourier Anal. Appl., 18 (2012), pp. 1195–1214.

[6] K. L. Clarkson and D. P. Woodruff, *Low-rank approximation and regression in input sparsity time*, J. ACM, 63 (2017), pp. 1–45.

[7] G. Cormode and S. Muthukrishnan, *An improved data stream summary: The count-min sketch and its applications*, J. Algorithms, 55 (2005), pp. 58–75.

[8] L. Dai and T. B. Schön, *On the exponential convergence of the Kaczmarz algorithm*, IEEE Signal Process. Lett., 22 (2015), pp. 1571–1574.

[9] J. J. Dongarra and D. Sørensen, *Linear algebra on high-performance computers*, Appl. Math. Comput., 20 (1986), pp. 57–88.

[10] R. Durrett, *Probability: Theory and Examples*, Cambridge University Press, Cambridge, UK, 2010.

[11] A. Galántai, *On the rate of convergence of the alternating projection method in finite dimensional spaces*, J. Math. Anal. Appl., 310 (2005), pp. 30–44.

[12] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof, *Rounding error analysis of the classical Gram-Schmidt orthogonalization process*, Numer. Math., 101 (2005), pp. 87–100.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, MD, 2013.

[14] R. Gordon, R. Bender, and G. T. Herman, *Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography*, J. Theoret. Biol., 29 (1970), pp. 471–481.

[15] R. Gower, D. Molitor, J. Moorman, and D. Needell, *Adaptive Sketch-and-Project Methods for Solving Linear Systems*, preprint, https://arxiv.org/abs/1909.03604, 2019.

[16] R. M. Gower and P. Richtárik, *Randomized iterative methods for linear systems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1660–1690, https://doi.org/10.1137/15M1025487.

[17] L. Gubin, B. T. Polyak, and E. Raik, *The method of projections for finding the common point of convex sets*, USSR Comput. Math. Math. Phys., 7 (1967), pp. 1–24.

[18] J. Haddock and A. Ma, *Greed Works: An Improved Analysis of Sampling Kaczmarz-Motkzin*, preprint, https://arxiv.org/abs/1912.03544, 2019.

[19] M. R. Hestenes, *Conjugate Direction Methods in Optimization*, Appl. Math. 12, Springer-Verlag, New York, Berlin, 1980.

[20] P. Indyk and R. Motwani, *Approximate nearest neighbors: Towards removing the curse of dimensionality*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 604–613.

[21] S. Kaczmarz, *Approximate solution of systems of linear equations*, Internat. J. Control, 57 (1993), pp. 1269–1271.

[22] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Internat. Acad. Polon. Sci. A, 35 (1937), pp. 355–357.

[23] C.-P. Lee and S. J. Wright, *Random permutations fix a worst case for cyclic coordinate descent*, IMA J. Numer. Anal., 39 (2019), pp. 1246–1275.

[24] A. Lent, *Maximum entropy and multiplicative art*, in Proceedings of the Conference on Image Analysis and Evaluation, SPSE, Toronto, 1976, pp. 249–257.

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

[26] A. Ma, D. Needell, and A. Ramdas, *Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1590–1604, https://doi.org/10.1137/15M1014425.

[27] M. W. Mahoney, *Lecture Notes on Randomized Linear Algebra*, preprint, https://arxiv.org/abs/1608.04481, 2016.

[28] T. S. Motzkin and I. J. Schoenberg, *The relaxation method for linear inequalities*, Canad. J. Math., 6 (1954), pp. 393–404.

[29] J. Nocedal, *Optimization methods for training neural networks*, in Proceedings of the 23rd International Symposium on Mathematical Programming, Bordeaux, France, 2018, https://ismp2018.sciencesconf.org/data/bookFullProgram.pdf.
[30] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.
[31] J. Nutini, B. Sepehry, I. Laradji, M. Schmidt, H. Koepke, and A. Virani, *Convergence Rates for Greedy Kaczmarz Algorithms, and Faster Randomized Kaczmarz Rules Using the Orthogonality Graph*, preprint, https://arxiv.org/abs/1612.07838, 2016.
[32] P. Richtárik and M. Takác, *Stochastic reformulations of linear systems: Algorithms and convergence theory*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 487–524.
[33] M. Rudelson and R. Vershynin, *Smallest singular value of a random rectangular matrix*, Comm. Pure Appl. Math., 62 (2009), pp. 1707–1739.
[34] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003, https://doi.org/10.1137/1.9780898718003.
[35] T. Strohmer and R. Vershynin, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl., 15 (2009), pp. 262–278.
[36] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher, *Practical sketching algorithms for low-rank matrix approximation*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1454–1485, https://doi.org/10.1137/17M1111590.
[37] T. Wallace and A. Sekmen, *Deterministic versus Randomized Kaczmarz Iterative Projection*, preprint, https://arxiv.org/abs/1407.5593, 2014.
[38] D. P. Woodruff, *Sketching as a tool for numerical linear algebra*, Found. Trends Theor. Comput. Sci., 10 (2014), pp. 1–157.
[39] S. Wright and C.-p. Lee, *Analyzing random permutations for cyclic coordinate descent*, Math. Comp., 89 (2020), pp. 2217–2248.
[40] W. Zhang and N. J. Higham, *Matrix Depot: An extensible test matrix collection for Julia*, PeerJ Comput. Sci., 2 (2016), e58.
[41] A. Zouzias and N. M. Freris, *Randomized extended Kaczmarz for solving least squares*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 773–793, https://doi.org/10.1137/120889897.