

# RANDOMLY SPARSIFIED RICHARDSON ITERATION: A DIMENSION-INDEPENDENT SPARSE LINEAR SOLVER \*

JONATHAN WEARE<sup>†</sup> AND ROBERT J. WEBBER<sup>‡</sup>

**Abstract.** Recently, a class of algorithms combining classical fixed point iterations with repeated random sparsification of approximate solution vectors has been successfully applied to eigenproblems with matrices as large as  $10^{108} \times 10^{108}$ . So far, a complete mathematical explanation for this success has proven elusive.

The family of methods has not yet been extended to the important case of linear system solves. In this paper we propose a new scheme based on repeated random sparsification that is capable of solving sparse linear systems in arbitrarily high dimensions. We provide a complete mathematical analysis of this new algorithm. Our analysis establishes a faster-than-Monte Carlo convergence rate and justifies use of the scheme even when the solution vector itself is too large to store.

**1. Introduction.** In this paper, we propose a randomized approach for solving a linear systems of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

involving a square matrix  $\mathbf{A} \in \mathbb{C}^{n \times n}$ , which is typically nonsymmetric, and a vector  $\mathbf{b} \in \mathbb{C}^n$ . Our new approach combines Richardson fixed-point iteration [55] with a strategy of random sparsification. The algorithm only requires examining a small, random subset of the columns of  $\mathbf{A}$ , which ensures the scalability to high dimensions  $n \geq 10^9$ . In the case of sparse columns, the algorithm can even be applied for  $n$  so large that the solution cannot be stored as a dense vector on a computer. The algorithm automatically discovers which entries of the solution vector are significant, leading to a high-accuracy sparse approximation. We will offer a full mathematical analysis and demonstrate the applicability to large-scale PageRank problems.

The classical Richardson iteration is presented as [Algorithm 1.1](#). The method can be applied to any linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , where  $\mathbf{A}$  and  $\mathbf{b}$  have been scaled so the eigenvalues  $\lambda_i(\mathbf{A})$  all satisfy  $|\lambda_i(\mathbf{A}) - 1| < 1$  (for more discussion of scaling, see [subsection 4.2](#)). Richardson iteration is based on rewriting  $\mathbf{A}\mathbf{x} = \mathbf{b}$  using the fixed-point formula

$$\mathbf{x} = \mathbf{G}\mathbf{x} + \mathbf{b}, \quad \text{where} \quad \mathbf{G} = \mathbf{I} - \mathbf{A}.$$

Motivated by this formula, Richardson iteration generates a sequence of approximations:

$$(1.1) \quad \begin{cases} \mathbf{x}_0 = \mathbf{0}, \\ \mathbf{x}_s = \mathbf{G}\mathbf{x}_{s-1} + \mathbf{b}. \end{cases}$$

The iterates  $\mathbf{x}_0, \mathbf{x}_1, \dots$  can be interpreted as an application of Horner's rule [37] for calculating the Neumann series  $\mathbf{x}_\star = \sum_{s=0}^{\infty} \mathbf{G}^s \mathbf{b}$ , and they converge to the solution vector  $\mathbf{x}_\star$  at an exponential rate specified by  $\mathbf{x}_s = \mathbf{x}_\star - \mathbf{G}^s \mathbf{x}_\star$ . Yet each step

---

\* **Funding:** JW acknowledges support from the Advanced Scientific Computing Research Program within the DOE Office of Science through award DE-SC0020427 and from the National Science Foundation through award DMS-2054306.

<sup>†</sup>Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 ([weare@nyu.edu](mailto:weare@nyu.edu))

<sup>‡</sup>Department of Mathematics, University of California San Diego, La Jolla, CA 92093 ([rwebber@ucsd.edu](mailto:rwebber@ucsd.edu))

---

**Algorithm 1.1** Classical Richardson iteration for solving  $\mathbf{Ax} = \mathbf{b}$  [55]

---

**Input:** Vector  $\mathbf{b} \in \mathbb{C}^n$ ; matrix  $\mathbf{A} \in \mathbb{C}^{n \times n}$ ; iteration count  $t$ 
**Output:** Approximate solution  $\mathbf{x}_t$  to  $\mathbf{Ax} = \mathbf{b}$ 

```

1  $\mathbf{x}_0 = \mathbf{0}$ .
2 for  $s = 1, 2, \dots, t$  do
3    $\mathbf{x}_s = (\mathbf{I} - \mathbf{A})\mathbf{x}_{s-1} + \mathbf{b}$ 
4 end for
5 Return  $\mathbf{x}_t$ 
```

---



---

**Algorithm 1.2** Randomly sparsified Richardson iteration for solving  $\mathbf{Ax} = \mathbf{b}$ 


---

**Input:** Vector  $\mathbf{b} \in \mathbb{C}^n$ ; program for evaluating columns of  $\mathbf{A} \in \mathbb{C}^{n \times n}$ ; parameters  $m, t_b$ ; iteration count  $t$ 
**Output:** Approximate solution  $\bar{\mathbf{x}}_t$  to  $\mathbf{Ax} = \mathbf{b}$ 

```

1  $\mathbf{x}_0 = \mathbf{0}$ 
2 for  $s = 1, 2, \dots, t-1$  do
3    $\phi_s(\mathbf{x}_{s-1}) = \text{sparsify}(\mathbf{x}_{s-1}, m)$  ▷ Sparsify using Algorithm 5.1
4    $\mathbf{x}_s = (\mathbf{I} - \mathbf{A})\phi_s(\mathbf{x}_{s-1}) + \mathbf{b}$ 
5 end for
6 Return  $\bar{\mathbf{x}}_t = \frac{1}{t-t_b} \sum_{s=t_b}^{t-1} \mathbf{x}_s$ 
```

---

of Richardson iteration can be computationally intensive, as it requires a full pass through the entries of  $\mathbf{G}$ .

Our new approach is called “randomly sparsified Richardson iteration” (RSRI). RSRI is similar to the classical Richardson iteration, except that we replace the deterministic iteration (1.1) with the randomized iteration

$$\begin{cases} \mathbf{x}_0 = \mathbf{0}, \\ \mathbf{x}_s = \mathbf{G}\phi_s(\mathbf{x}_{s-1}) + \mathbf{b}, \end{cases}$$

where  $\phi_s$  is a random operator that inputs  $\mathbf{x}_{s-1}$  and outputs a sparse random vector  $\phi_s(\mathbf{x}_{s-1})$  with no more than  $m$  nonzero entries. Due to sparsity, it is cheap to evaluate  $\mathbf{G}\phi_s(\mathbf{x}_{s-1})$ . Instead of a full pass through the matrix at each iteration, the algorithm only requires a multiplication involving a random subset of  $m$  columns, where  $m$  is a tunable parameter. The random sparsification introduces errors, which are reduced by averaging over successive iterates  $\mathbf{x}_{t_b}, \mathbf{x}_{t_b+1}, \dots, \mathbf{x}_{t-1}$ . See the pseudocode in Algorithm 1.2.

We will prove that the RSRI solution  $\bar{\mathbf{x}}_t$  converges as  $t \rightarrow \infty$  if  $\mathbf{G}$  is a strict 1-norm contraction:

$$\|\mathbf{G}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |\mathbf{G}(i, j)| < 1.$$

Under the contractivity assumption, Theorem 2.1 establishes a quantitative convergence rate. We will also prove that RSRI converges with weaker requirements on  $\mathbf{G}$ , which are defined in Theorem 6.1.

RSRI combines Richardson iteration with random sparsification, but the approach is *faster* than Richardson iteration for high-dimensional problems and *more accurate* than direct Monte Carlo sampling.

- *Speed:* Richardson iteration requires a complete pass through the matrix at

each iteration. In contrast, RSRI only requires reading  $m$  columns of  $\mathbf{A}$  at every iteration, where  $m$  is a tunable sparsity level that can be quite small (say,  $m \leq n/10^3$ ).

- *Accuracy:* Classical Monte Carlo strategies for solving linear systems [24, 65] give error bars of size  $\sim m^{-1/2}$  where  $m$  is the number of samples. Our tunable sparsity parameter  $m$  plays a similar role to a number of samples. As we increase  $m$ , we will highlight settings in which RSRI converges at a polynomial rate  $\sim m^{-p}$  for  $p > 1/2$  or at an exponential rate  $\sim e^{-cm}$  for  $c > 0$ .

The core component of RSRI is the random sparsification operator  $\phi_s$ , which inputs a vector  $\mathbf{x}_{s-1} \in \mathbb{C}^n$  and outputs a sparse random vector  $\phi_s(\mathbf{x}_{s-1}) \in \mathbb{C}^n$ . We will optimize  $\phi_s$  in section 5, leading to the high-performing sparsification operator described in Algorithm 5.2. With some probability  $p_i$ , the operator replaces the  $i$ th entry  $\mathbf{x}_{s-1}(i)$  with a higher-magnitude entry  $\mathbf{x}_{s-1}(i)/p_i$ ; with the remaining probability  $1 - p_i$ , the operator sets the  $i$ th entry to zero. The probabilities  $p_i$  increase proportionally to the magnitude  $|\mathbf{x}_{s-1}(i)|$ , reaching  $p_i = 1$  for the largest-magnitude entries. Therefore, these large-magnitude entries are *preserved exactly*. The combination of randomized rounding and exact preservation leads to an unbiased approximation of the input vector. Moreover, the input and output vectors are close if the input vector has rapidly decaying entries (Theorem 5.1).

The per-iteration runtime of RSRI is just  $\mathcal{O}(mn)$  operations when  $\mathbf{A}$  is a dense matrix. The per-iteration runtime of RSRI is even lower — just  $\mathcal{O}(mq)$  operations per iteration — when  $\mathbf{A}$  and  $\mathbf{b}$  are sparse with no more than  $q$  nonzero entries per column. In the sparse case, the runtime and memory costs are independent of dimension. Additionally, if the goal is to compute inner products with the exact solution, the memory cost can be reduced from  $\mathcal{O}(tmq)$  to  $\mathcal{O}(mq)$  by averaging each inner product over the iterates instead of storing the RSRI solution  $\bar{\mathbf{x}}_t$ .

Random sparsification is an approach of growing importance in numerical linear algebra [24, 63, 42, 43]. One example is stochastic gradient descent, which we contrast with RSRI in subsection 4.2. As another example, random sparsification has been applied to eigenvalue problems in quantum chemistry with matrices as large as  $10^{108} \times 10^{108}$ , as discussed in subsection 4.4. RSRI provides a new instantiation of the random sparsification approach for linear systems, and the present work gives mathematical and empirical demonstrations of RSRI’s effectiveness.

**1.1. Plan for paper.** The rest of this paper is organized as follows. Section 2 presents our main error bound for RSRI, section 3 applies RSRI to PageRank problems, section 4 discusses algorithms related to RSRI, section 5 analyzes random sparsification, and section 6 proves our main error bound for RSRI.

**1.2. Notation.** We use the shorthand  $\lfloor a \rfloor = \max\{z \in \mathbb{Z} : z \leq a\}$  and  $a \vee b = \max\{a, b\}$  for  $a, b \in \mathbb{R}$ . The complex conjugate of  $z \in \mathbb{C}$  is  $\bar{z}$ . We write vectors  $\mathbf{v} \in \mathbb{C}^n$  and matrices  $\mathbf{M} \in \mathbb{C}^{n \times n}$  in bold, and we write their elements as  $\mathbf{v}(i)$  or  $\mathbf{M}(i, j)$ . The conjugate transposes are  $\mathbf{v}^*$  and  $\mathbf{M}^*$ , while  $|\mathbf{v}|$  and  $|\mathbf{M}|$  denote the entry-wise absolute values. For any vector  $\mathbf{x} \in \mathbb{C}^n$ , the decreasing rearrangement  $\mathbf{x}^\downarrow \in \mathbb{C}^n$  is a vector with the same elements as  $\mathbf{x}$  but placed in weakly decreasing order:

$$|\mathbf{x}^\downarrow(1)| \geq |\mathbf{x}^\downarrow(2)| \geq \dots \geq |\mathbf{x}^\downarrow(n)|.$$

The decreasing rearrangement may not be unique, so we employ the notation only in contexts where it leads to an unambiguous statement. The vector 1-norm, Euclidean

norm, and  $\infty$ -norm are  $\|\mathbf{v}\|_1 = \sum_{i=1}^n |\mathbf{v}(i)|$ ,  $\|\mathbf{v}\| = (\sum_{i=1}^n |\mathbf{v}(i)|^2)^{1/2}$  and  $\|\mathbf{v}\|_\infty = \max_{1 \leq i \leq n} |\mathbf{v}(i)|$ . The number of nonzero entries is  $\|\mathbf{v}\|_0 = \#\{1 \leq i \leq n : \mathbf{v}(i) \neq 0\}$ . The matrix 1-norm is  $\|\mathbf{M}\|_1 = \max_{\|\mathbf{v}\|_1=1} \|\mathbf{M}\mathbf{v}\|_1$ .

**2. Main error bound for RSRI.** Our main result is the following detailed error bound for RSRI whose proof appears in [section 6](#).

**THEOREM 2.1** (Main error bound). *Suppose RSRI with sparsity level  $m$  is applied to an  $n \times n$  linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$  for which  $\mathbf{G} = \mathbf{I} - \mathbf{A}$  is a strict 1-norm contraction:*

$$\|\mathbf{G}\|_1 = \max_{1 \leq j \leq n} \sum_{1 \leq i \leq n} |\mathbf{G}(i, j)| < 1.$$

RSRI returns a solution  $\bar{\mathbf{x}}_t$  satisfying the bias-variance formula

$$(2.1) \quad \mathbb{E} \|\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{b}\|^2 = \underbrace{\|\mathbf{A} \mathbb{E}[\bar{\mathbf{x}}_t] - \mathbf{b}\|^2}_{\text{bias}^2} + \underbrace{\mathbb{E} \|\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{A} \mathbb{E}[\bar{\mathbf{x}}_t]\|^2}_{\text{variance}}.$$

Here the expectation averages over the random set of entries rounded to zero at each sparsification step. The square bias is bounded by

$$\text{bias}^2 \leq \left( \frac{2\|\mathbf{G}^{t_b} \mathbf{x}_\star\|_1}{t - t_b} \right)^2,$$

where  $\mathbf{x}_\star$  is the exact solution. The variance is bounded by

$$\text{variance} \leq \frac{8t}{(t - t_b)^2} \cdot \frac{1}{m} \left( \frac{\|\mathbf{b}\|_1}{1 - \|\mathbf{G}\|_1} \right)^2.$$

Additionally, if  $m \geq m_{\mathbf{G}} = 1/(1 - \|\mathbf{G}\|_1^2)$ , the variance is bounded by

$$(2.2) \quad \text{variance} \leq \frac{8t}{(t - t_b)^2} \cdot \min_{i \leq m - m_{\mathbf{G}}} \frac{1}{m - m_{\mathbf{G}} - i} \left( \sum_{j=i+1}^n \tilde{\mathbf{x}}^\downarrow(j) \right)^2.$$

Here  $\tilde{\mathbf{x}} \in \mathbb{R}^n$  is the solution to the regularized linear system  $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ , where  $\tilde{\mathbf{A}} = \mathbf{I} - |\mathbf{G}|$ ,  $\tilde{\mathbf{b}} = |\mathbf{b}|$ . As usual,  $\tilde{\mathbf{x}}^\downarrow$  is the decreasing rearrangement of  $\tilde{\mathbf{x}}$ .

In [Theorem 2.1](#), we bound the mean square error of RSRI as the sum of a square bias term and a variance term. The square bias in RSRI is the square bias that occurs in the deterministic Richardson iteration, after averaging over the iterates  $\mathbf{x}_{t_b}, \dots, \mathbf{x}_{t-1}$  (see [Proposition 6.2](#)). The square bias decays exponentially fast as we increase the burn-in time  $t_b$  and it is often small in practice. For example, in our experiments in [section 3](#), we obtain a negligibly small bias by setting  $t_b = t/2$ .

The variance term in [Theorem 2.1](#) decays at a rate  $\mathcal{O}(1/m)$  or faster, depending on the decay of the entries in the regularized solution vector  $\tilde{\mathbf{x}}$ . Note that the entries of the regularized solution  $\tilde{\mathbf{x}}$  lie above the entries of the true solution  $\mathbf{x}_\star$  due to the element-wise inequality

$$|\mathbf{x}_\star| = \left| \sum_{s=0}^{\infty} \mathbf{G}^s \mathbf{b} \right| \leq \sum_{s=0}^{\infty} |\mathbf{G}|^s |\mathbf{b}| = \tilde{\mathbf{x}}.$$

The vectors  $\tilde{\mathbf{x}}$  and  $\mathbf{x}_\star$  are identical if  $\mathbf{G}$  and  $\mathbf{b}$  are nonnegative-valued, which is the case for PageRank problems (see [section 3](#)).

**Theorem 2.1** leads to the main message of this work. There is a class of problems in which the entries of  $\tilde{\mathbf{x}}$  are decreasing quickly, and RSRI converges at a fast polynomial or exponential rate. We establish the following corollary of **Theorem 2.1**.

**COROLLARY 2.2** (Fast polynomial or exponential convergence). *Instate the notation of **Theorem 2.1**. If the sparsity level satisfies  $m \geq m_{\mathbf{G}} := 1/(1 - \|\mathbf{G}\|_1)$  and*

$$\sum_{j=i}^n \tilde{\mathbf{x}}^\downarrow(j) \leq i^{-p} \quad \text{for } 1 \leq i \leq n,$$

*then the RSRI variance (2.1) is bounded by*

$$\text{variance} \leq 16e^{\frac{(p + \frac{1}{2})t}{(t - t_b)^2}} \cdot (m - m_{\mathbf{G}})^{-2(p + \frac{1}{2})}.$$

*If the sparsity level satisfies  $m \geq m_{\mathbf{G}} + \frac{1}{2c}$  and*

$$\sum_{j=i}^n \tilde{\mathbf{x}}^\downarrow(j) \leq e^{-ci} \quad \text{for } 1 \leq i \leq n,$$

*then the RSRI variance (2.1) is bounded by*

$$\text{variance} \leq 16e^{\frac{ct}{(t - t_b)^2}} \cdot e^{-2c(m - m_{\mathbf{G}})}.$$

*Proof.* For the first bound, we take  $i = \lfloor \frac{2p}{2p+1}(m - m_{\mathbf{G}}) \rfloor$  and evaluate (2.2). For the second bound, we take  $i = \lfloor m - m_{\mathbf{G}} - \frac{1}{2c} \rfloor$  and evaluate (2.2) again.  $\square$

For specific problems in which the regularized solution vector  $\tilde{\mathbf{x}}$  is quickly decaying, **Corollary 2.2** establishes that RSRI is more efficient than a pure Monte Carlo method. If the tail  $\sum_{j=i}^n \tilde{\mathbf{x}}^\downarrow(j)$  decays polynomially with rate  $i^{-p}$ , then RSRI converges at the rate  $\mathcal{O}(m^{-p-1/2})$ . Additionally, if the tail  $\sum_{j=i}^n \tilde{\mathbf{x}}^\downarrow(j)$  decays exponentially with rate  $e^{-\Delta i}$ , then RSRI converges at the exponential rate  $\mathcal{O}(e^{-\Delta m})$ . In the next section, we will show examples of PageRank problems where the solution vector exhibits polynomial tail decay.

**3. The PageRank problem.** Consider a network of  $n$  websites. A restless web surfer chooses an initial website at random, according to a probability vector  $\mathbf{s} \in [0, 1]^n$ . At any time, there is a fixed probability  $\alpha \in (0, 1)$  that the web surfer follows a random hyperlink. In this case, the probability of transitioning from website  $j$  to website  $i$  is denoted as  $\mathbf{P}(i, j)$ . With the remaining probability  $1 - \alpha$ , the web surfer abandons the chain of websites and chooses a fresh website according to the probability vector  $\mathbf{s}$ . The PageRank problem asks: what is the long-run distribution of websites visited by the web surfer?

The PageRank problem can be solved using a linear system of equations. We let  $\mathbf{x} \in \mathbb{R}^n$  denote the long-run distribution of visited websites and let  $\mathbf{P} \in \mathbb{R}^{n \times n}$  denote the column-stochastic matrix of transition probabilities. Then  $\mathbf{x}$  satisfies

$$(3.1) \quad \mathbf{x} = \alpha \mathbf{P} \mathbf{x} + (1 - \alpha) \mathbf{s}.$$

By setting  $\mathbf{A} = \mathbf{I} - \alpha \mathbf{P}$  and  $\mathbf{b} = (1 - \alpha) \mathbf{s}$ , we can rewrite this linear system in the standard form

$$\mathbf{A} \mathbf{x} = \mathbf{b},$$

where the matrix  $\mathbf{A}$  is typically sparse and high-dimensional.

Since the late 1990s, Google has applied PageRank with  $\alpha = .85$  to help determine which websites show up first in their search results. Originally, Google solved the PageRank problem using the Richardson fixed-point iteration based on (3.1). For a network of  $N = 2.4 \times 10^7$  websites with  $3.2 \times 10^8$  hyperlinks, the developers of PageRank report that Richardson iteration converges to the target accuracy in 52 iterations, which made the algorithm practical for ranking websites in 1998 [52]. However, the 2025 internet has a greater number of websites and hyperlinks, leading to a more challenging PageRank problem.

Beyond the internet, PageRank problems arise in chemistry, biology, and data science, among other areas [28]. We focus especially on *personalized PageRank problems*, in which the probability vector  $\mathbf{s}$  has just one or a few nonzero entries. The personalized PageRank problem identifies which vertices are important in a local region associated with the nonzero entries of  $\mathbf{s}$ . For example, personalized PageRank problems arise when recommending items to particular users on Netflix or Amazon [29], when disambiguating the meanings of words in a sentence [1], or when constructing personalized reading lists [66].

The rest of this section is organized as follows. Subsection 3.1 gives a history of Monte Carlo algorithms for large-scale personalized PageRank problems, subsection 3.2 presents a mathematical analysis showing that RSRI is effective for the PageRank problems, and subsection 3.3 presents empirical tests. Alternative PageRank algorithms with different design principles are discussed in subsection 4.3.

**3.1. History of Monte Carlo PageRank algorithms.** Personalized PageRank problems depend most strongly on the vertices associated with the nonzero entries of  $\mathbf{s}$  and nearby regions, so a *local* search over the vertices can in principle produce a high-quality approximation. Based on this insight, several researchers have proposed algorithms to identify the most important vertices and sparsely approximate the PageRank solution vector. The Monte Carlo algorithms of [23, 6, 11] simulate a web surfer that moves from vertex to vertex, and the visited vertices determine the sparsity pattern. In contrast, the deterministic algorithms of [38, 9, 27, 58, 4] identify a set of significant vertices by progressively adding new vertices for which the residual is large.

Monte Carlo algorithms for the PageRank problem converge at the typical  $\sim m^{-1/2}$  error rate, where  $m$  is the number of samples. For example, the Monte Carlo scheme of Fogaras et al. [23] simulates  $m$  restless web surfers until the first time they become bored and abandon the search. The last websites visited by the surfers are recorded as  $Z_1, Z_2, \dots, Z_m \in \{1, \dots, n\}$ . Since each  $Z_i$  is an independent sample from the PageRank distribution, the PageRank vector  $\mathbf{x}_\star \in \mathbb{R}^n$  is approximated as  $\hat{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^m \mathbf{e}_{Z_i}$  where  $\mathbf{e}_j$  denotes the  $j$ th basis vector. The approximation satisfies the following sharp variance bound:

$$(3.2) \quad \begin{aligned} \mathbb{E} \|\hat{\mathbf{x}} - \mathbf{x}_\star\|^2 &= \sum_{i=1}^n \mathbb{E} |\hat{\mathbf{x}}(i) - \mathbf{x}_\star(i)|^2 = \frac{1}{m} \sum_{i=1}^n \mathbf{x}_\star(i)(1 - \mathbf{x}_\star(i)) \\ &\leq \frac{1}{m} \sum_{i=1}^n \mathbf{x}_\star(i) = \frac{1}{m}. \end{aligned}$$

Here, we use the fact that each  $\hat{\mathbf{x}}(i)$  is a rescaled binomial random variable with parameters  $m$  and  $\mathbf{x}_\star(i)$ , so the variance is  $\frac{1}{m} \mathbf{x}_\star(i)(1 - \mathbf{x}_\star(i))$ . The error bound (3.2) is completely independent of the dimension, but it signals a slow  $m^{-1/2}$  convergence in the root-mean-square error as we increase the number of samples  $m$ . Alternative

Monte Carlo algorithms such as [6, Alg. 3] and [11, Alg. 3] improve the runtime and variance by a constant factor, but they do not change the fundamental  $m^{-1/2}$  convergence rate.

**3.2. Faster PageRank by RSRI.** When we apply RSRI to solve the personalized PageRank problem and we use the minimal sparsity setting  $m = 1$ , it is equivalent to a standard Monte Carlo algorithm [6, Alg. 3]. However, as we raise  $m$ , RSRI exhibits more complicated behavior and it satisfies the following error bound, with the proof appearing in [section 6](#).

**PROPOSITION 3.1** (PageRank error bound). *If randomly sparsified Richardson iteration is applied to the PageRank problem  $\mathbf{x} = \alpha \mathbf{P}\mathbf{x} + (1 - \alpha)\mathbf{s}$  with parameters  $m \geq m_\alpha := 1/(1 - \alpha^2)$  and  $t \geq 2t_b$ , RSRI returns a solution vector  $\bar{\mathbf{x}}_t$  satisfying*

$$\mathbb{E}\|\bar{\mathbf{x}}_t - \mathbf{x}_\star\|^2 \leq \left[ \frac{4\alpha^{t_b}}{(1 - \alpha)t} \right]^2 + \frac{16}{(1 - \alpha)^{2t}} \cdot \min_{0 \leq i \leq m - m_\alpha} \frac{1}{m - m_\alpha - i} \left[ \sum_{j=i+1}^n \mathbf{x}_\star^\downarrow(j) \right]^2,$$

where  $\mathbf{x}_\star^\downarrow$  is the decreasing rearrangement of the solution vector  $\mathbf{x}_\star$ .

[Proposition 3.1](#) bounds the mean square PageRank error in the Euclidean norm. The variance term decays at a rate at least  $\mathcal{O}(m^{-1})$ , and it decays more rapidly than  $\mathcal{O}(m^{-1})$  when the entries of  $\mathbf{x}$  are decaying rapidly. Moreover, we observe that the entries of  $\mathbf{x}$  must decay rapidly if  $\mathbf{P}$  is sparse, which is typically the case in PageRank problems. We present the following sparsity-based estimate.

**LEMMA 3.2** (Decay of entries in the PageRank vector). *Consider the personalized PageRank problem  $\mathbf{x} = \alpha \mathbf{P}\mathbf{x} + (1 - \alpha)\mathbf{e}_i$ , and assume each column of  $\mathbf{P}$  has at most  $q$  nonzero entries, where  $q \geq 2$ . Then the entries of the solution  $\mathbf{x}_\star$  satisfy*

$$\sum_{j=i}^n \mathbf{x}_\star^\downarrow(j) \leq \alpha^{-1} i^{-\log_q(1/\alpha)}.$$

*Proof of Lemma 3.2.* Let  $\text{dist}(i, j)$  denote the length of the shortest path

$$i = k_0 \rightarrow k_1 \rightarrow \cdots \rightarrow k_{s-1} \rightarrow k_s = j,$$

which has positive probability of occurring, i.e.,  $\mathbf{P}(k_r, k_{r-1}) > 0$  for each  $r = 1, \dots, s$ . By the sparsity condition, there can be at most  $1 + q + q^2 + \cdots + q^{s-1} < q^s$  paths of length  $s - 1$  or shorter. Now let  $\sigma_1, \dots, \sigma_n$  be a permutation of the indices  $1, \dots, n$  so that  $\sigma_1 = i$  and  $\text{dist}(i, \sigma_j) \leq \text{dist}(i, \sigma_k)$  whenever  $j \leq k$ . For any index  $j \geq q^s$ , we must have  $\text{dist}(i, \sigma_j) \geq s$ .

Now we use the representation  $\mathbf{x}_\star = (1 - \alpha) \sum_{s=0}^{\infty} \alpha^s \mathbf{P}^s \mathbf{e}_i$  and the fact that  $\mathbf{P}^s$  is column-stochastic to calculate

$$\begin{aligned} \sum_{j \geq m} \mathbf{x}_\star(\sigma_j) &= (1 - \alpha) \sum_{j \geq m} \sum_{s=0}^{\infty} \alpha^s \mathbf{P}^s(\sigma_j, i) \\ &= (1 - \alpha) \sum_{j \geq m} \sum_{s=\lfloor \log_q(m) \rfloor}^{\infty} \alpha^s \mathbf{P}^s(\sigma_j, i) \\ &\leq (1 - \alpha) \sum_{s=\lfloor \log_q(m) \rfloor}^{\infty} \alpha^s = \alpha^{\lfloor \log_q(m) \rfloor} \leq \alpha^{-1} m^{-\log_q(1/\alpha)}, \end{aligned}$$

which establishes the result.  $\square$

By combining [Proposition 3.1](#) and [Lemma 3.2](#), we guarantee that RSRI converges at a  $\mathcal{O}(m^{-1/2 - \log_q(1/\alpha)})$  rate as we increase the sparsity parameter  $m$ . This rate is



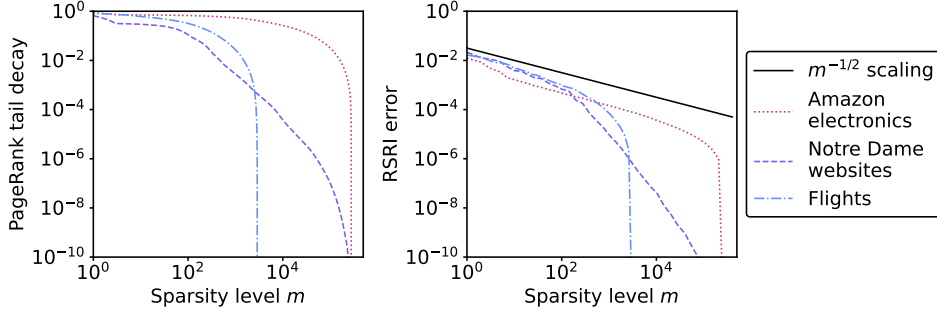


FIG. 1. (**RSRI error scaling.**) Left panel shows tail decay  $\sum_{i=m}^n \mathbf{x}^\downarrow(i)$  of the sorted PageRank solution  $\mathbf{x}^\downarrow$  for three personalized PageRank problems documented in subsection 3.3. Right panel shows RSRI error  $(\mathbb{E}\|\hat{\mathbf{x}} - \mathbf{x}_\star\|^2)^{1/2}$  with sparsity level  $m$ . The right panel is essentially the left panel multiplied by a factor of  $m^{-1/2}$ .

faster than the  $\sim m^{-1/2}$  Monte Carlo error scaling, and it theoretically separates RSRI from pure Monte Carlo methods. For many PageRank problems, we expect even faster convergence than Lemma 3.2 would suggest. For example, the PageRank solution decays especially rapidly when there are “hub” vertices which have many incoming edges [38].

**3.3. Empirical tests.** To test the empirical performance of RSRI, we apply the algorithm to three personalized PageRank problems:

- **Amazon electronics** [47, 3]. We rank  $n = 3.6 \times 10^5$  Amazon electronics products that received five-star reviews and were available in 2019. The transitions probabilities are determined by linking from an electronics product to a different random product receiving a 5-star review from the same reviewer.
- **Notre Dame websites** [2, 41]. We rank  $n = 3.2 \times 10^5$  websites within the 1999 University of Notre Dame web domain. The transition probabilities are determined by selecting a random outgoing hyperlink.
- **Airports** [50]. We rank  $n = 2.9 \times 10^3$  global airports. The transition probabilities are determined by selecting a random outgoing flight from the documented flight routes in 2010.

In all three problems, we compute the personalized PageRank vector for a randomly chosen vertex  $i \in \{1, \dots, n\}$ . If there is a dangling vertex  $j$  which lacks outgoing edges, we update the  $j$ th column of the transition matrix  $\mathbf{P}$  to be the basis vector  $\mathbf{e}_i$ . This is equivalent to solving the unnormalized problem  $\mathbf{x} = \alpha \mathbf{P} \mathbf{x} + (1 - \alpha) \mathbf{e}_i$  and rescaling the solution  $\mathbf{x}$  to sum to one [28, Thm. 2.5]. We apply RSRI with parameters  $\alpha = .85$ ,  $t = 1000$ ,  $t_b = t/2$  and present the root mean square error  $(\mathbb{E}\|\hat{\mathbf{x}}_t - \mathbf{x}_\star\|^2)^{1/2}$  in Figure 1 (right). The expectation is evaluated empirically over ten independent trials. See <https://github.com/rjwebber/rsri> for code for all experiments in this paper.

The results verify that RSRI provides a high-accuracy solution for all three PageRank problems, reaching error levels of  $10^{-6}$ – $10^{-3}$  even when  $m \leq n/10^2$ . Moreover, the error decays at a faster-than- $m^{-1/2}$  rate as we increase  $m$ . The precise rate of convergence depends on the tail decay  $\sum_{i=m}^n \mathbf{x}_\star^\downarrow(i)$ . The RSRI error (Figure 1, right) is approximately the PageRank tail decay (Figure 1, left) multiplied by a  $\mathcal{O}(m^{-1/2})$  prefactor term. In accordance with Proposition 3.1, RSRI converges fastest when the tail decays fastest, which occurs in the Notre Dame websites problem.



**4. Related algorithms.** Here we review the algorithms most closely related to RSRI.

**4.1. The Monte Carlo method for linear systems.** In the late 1940s, Ulam and von Neumann introduced a Monte Carlo strategy for solving linear systems  $\mathbf{A}\mathbf{x} = \mathbf{b}$  [24]. To motivate this strategy, let us write the solution vector  $\mathbf{x}_\star$  as

$$\mathbf{x}_\star = \sum_{s=0}^{\infty} \mathbf{G}^s \mathbf{b}, \quad \text{where } \mathbf{G} = \mathbf{I} - \mathbf{A},$$

and let us assume that  $\mathbf{b}$  and  $\mathbf{G}$  have nonnegative-valued entries with  $\sum_{i=1}^n \mathbf{b}(i) = 1$  and  $\sum_{i=1}^n \mathbf{G}(i, j) < 1$  for  $1 \leq j \leq n$ . Ulam and von Neumann interpreted each term  $\mathbf{G}^s \mathbf{b}$  using a finite-state Markov chain  $(X_s)_{s \geq 0}$  with transition probabilities

$$(4.1) \quad \mathbb{P}\{X_0 = i\} = \mathbf{b}(i), \quad \mathbb{P}\{X_s = i | X_{s-1} = j\} = \mathbf{G}(i, j).$$

The transition probabilities  $(\mathbf{G}(i, j))_{1 \leq i \leq n}$  sum to less than one, so there is a positive probability

$$(4.2) \quad p_j := 1 - \sum_{i=1}^n \mathbf{G}(i, j).$$

that the Markov chain occupying state  $j$  at time  $s$  is *killed*. In this case, define the killing time  $\tau$  to be  $\tau = s$ .

As a practical algorithm, Ulam and von Neumann suggested simulating the Markov chain with killing on a computer and estimating the vector  $\mathbf{x}_\star$  using

$$(4.3) \quad \hat{\mathbf{x}} = \frac{1}{p_{X_\tau}} \mathbf{e}_{X_\tau},$$

where  $p_j$  is the  $j$ th killing probability (4.2),  $\mathbf{e}_j$  is the  $j$ th basis vector, and  $X_\tau$  is the state of the Markov chain when the random killing occurs. This stochastic estimator is unbiased but has a high variance, so Ulam and von Neumann proposed averaging over many independent estimators  $\hat{\mathbf{x}}$  to bring down the variance. They also introduced a more complicated Monte Carlo procedure for solving systems in which  $\mathbf{b}$  and  $\mathbf{G}$  can have negative-valued entries [24], and further improvements were introduced by Wasow [65] soon after.

Since the 1950s, computational scientists have applied the Monte Carlo method to solve large-scale linear systems arising from numerical discretizations of PDEs [57, 64, 21, 8] and integral equations [40]. The method has been analyzed by numerical analysts [24, 65, 19, 15, 7, 18, 49, 39] and theoretical computer scientists [51, 61, 5]. The Monte Carlo method was even rediscovered in the PageRank community, leading to the Monte Carlo PageRank algorithms of Fogaras et al. (2005) [23] and Avrachenkov et al. (2007) [6] as discussed in subsection 3.1. Despite all this research, however, the method is fundamentally limited by a  $m^{-1/2}$  convergence rate. Modifications to the basic procedure do not fundamentally change the convergence rate, or else they change the character of the algorithm by reading  $\mathcal{O}(n)$  columns per iteration [35].

Randomly sparsified Richardson iteration is based on random sampling and is a Monte Carlo method in the case  $m = 1$ . However, for  $m > 1$ , RSRI improves on past Monte Carlo methods. We are not aware of another Monte Carlo-based algorithm that achieves faster-than- $m^{-1/2}$  convergence while requiring just  $\mathcal{O}(m)$  column evaluations per iteration.

**4.2. Stochastic gradient descent.** Stochastic gradient descent (SGD) is a class of methods that speed up traditional gradient descent by subsampling the terms in an expansion of the gradient [56, 46, 25]. Like traditional gradient descent, SGD is designed to minimize a scalar loss function  $L : \mathbb{R}^d \rightarrow \mathbb{R}$ . When solving a linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , there are a couple canonical loss functions that lead to a minimizer  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ .

- (a) When  $\mathbf{A}$  is strictly positive definite, one choice of loss function is  $L_1(\mathbf{x}) = \frac{1}{2}\mathbf{x}^*\mathbf{A}\mathbf{x} - \mathbf{x}^*\mathbf{b}$ .
- (b) When  $\mathbf{A}$  is any invertible matrix, a different choice of loss function is  $L_2(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ .

Gradient descent methods minimize either of these loss functions by taking steps in the negative gradient direction,  $-\nabla L_1(\mathbf{x})$  or  $-\nabla L_2(\mathbf{x})$ . In contrast, SGD methods subsample just one or a few terms from the gradient expansion

$$\begin{aligned} \nabla L_1(\mathbf{x}) &= \sum_{j=1}^n \mathbf{g}_{1,j}(\mathbf{x}), \quad \text{where } \mathbf{g}_{1,j}(\mathbf{x}) = [\mathbf{A}(j, \cdot)\mathbf{x} - \mathbf{b}(j)]\mathbf{e}_j, \\ \text{or } \nabla L_2(\mathbf{x}) &= \sum_{j=1}^n \mathbf{g}_{2,j}(\mathbf{x}), \quad \text{where } \mathbf{g}_{2,j}(\mathbf{x}) = [\mathbf{A}(j, \cdot)\mathbf{x} - \mathbf{b}(j)]\mathbf{A}(j, \cdot)^*. \end{aligned}$$

For example, uniform mini-batch SGD [30, 25] samples an index set  $S \subseteq \{1, \dots, n\}$  uniformly at random without replacement. Then, it makes an update

$$\mathbf{x}_s = \mathbf{x}_{s-1} - \frac{\alpha n}{|S|} \sum_{j \in S} \mathbf{g}_{i,j}(\mathbf{x}_{s-1}),$$

where  $\alpha > 0$  is a step size parameter and  $|S|$  is the cardinality of  $S$ . In contrast, importance sampling SGD [46, 45] samples  $S$  with replacement from a nonuniform probability distribution  $(p_i)_{1 \leq i \leq n}$ . Then, it makes an update

$$\mathbf{x}_s = \mathbf{x}_{s-1} - \frac{\alpha}{|S|} \sum_{j \in S} \frac{\mathbf{g}_{i,j}(\mathbf{x}_{s-1})}{p_j}.$$

Two types of importance sampling SGD are especially well-known. Randomized coordinate descent [42, 53] uses unequal sampling probabilities  $p_j = \mathbf{A}(j, j)/\text{tr}(\mathbf{A})$  to optimize the loss function  $L_1$ , while randomized Kaczmarz [63, 45] uses unequal sampling probabilities  $p_j = \|\mathbf{A}(j, \cdot)\|^2 / \|\mathbf{A}\|_F^2$  to optimize the loss function  $L_2$ .

On the surface, SGD is similar to RSRI. The expected value of the update in SGD is:

$$\begin{aligned} \mathbb{E}[\mathbf{x}_s - \mathbf{x}_{s-1} | \mathbf{x}_{s-1}] &= -\alpha \nabla L_1(\mathbf{x}_{s-1}) = \alpha(\mathbf{b} - \mathbf{A}\mathbf{x}_{s-1}) \\ \text{or } \mathbb{E}[\mathbf{x}_s - \mathbf{x}_{s-1} | \mathbf{x}_{s-1}] &= -\alpha \nabla L_2(\mathbf{x}_{s-1}) = \alpha \mathbf{A}^*(\mathbf{b} - \mathbf{A}\mathbf{x}_{s-1}). \end{aligned}$$

RSRI makes exactly the same update in expectation when applied to  $\alpha \mathbf{A}\mathbf{x} = \alpha \mathbf{b}$  or  $\alpha \mathbf{A}^* \mathbf{A}\mathbf{x} = \alpha \mathbf{A}^* \mathbf{b}$ .

Although the SGD and RSRI updates point in the same direction in expectation, the step size is chosen differently in these methods. When RSRI is applied to  $\alpha \mathbf{A}\mathbf{x} = \alpha \mathbf{b}$  or  $\alpha \mathbf{A}^* \mathbf{A}\mathbf{x} = \alpha \mathbf{A}^* \mathbf{b}$ , the step size can be any value  $\alpha \in (0, 2/\|\mathbf{A}\|)$  or  $\alpha \in (0, 2/\|\mathbf{A}\|^2)$  and still the algorithm converges under  $\ell_1$  contractivity assumptions (Theorem 6.1). In contrast, the step size needs to be much smaller for SGD methods with a constant batch size to converge. For example, randomized coordinate descent

[42, 53] produces iterates that satisfy

$$(4.4) \quad \mathbb{E}\|\mathbf{A}^{1/2}(\mathbf{x}_s - \mathbf{x}_\star)\|^2 = \mathbb{E}\|\mathbf{A}^{1/2}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2 \\ + \left[ \frac{\alpha^2 \text{tr}(\mathbf{A})}{|\mathbf{S}|} - 2\alpha \right] \mathbb{E}\|\mathbf{A}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2 + \alpha^2 \frac{|\mathbf{S}| - 1}{|\mathbf{S}|} \mathbb{E}\|\mathbf{A}^{3/2}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2.$$

Meanwhile, randomized Kaczmarz [63, 45] produces iterates that satisfy

$$(4.5) \quad \mathbb{E}\|\mathbf{x}_s - \mathbf{x}_\star\|^2 = \mathbb{E}\|\mathbf{x}_{s-1} - \mathbf{x}_\star\|^2 \\ + \left[ \alpha^2 \frac{\|\mathbf{A}\|_F^2}{|\mathbf{S}|} - 2\alpha \right] \mathbb{E}\|\mathbf{A}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2 + \alpha^2 \frac{|\mathbf{S}| - 1}{|\mathbf{S}|} \mathbb{E}\|\mathbf{A}^* \mathbf{A}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2.$$

See [Appendix A](#) for a derivation. Thus, the SGD iterates diverge when  $\alpha > 2|\mathbf{S}|/\text{tr}(\mathbf{A})$  for randomized coordinate descent or  $\alpha > 2|\mathbf{S}|/\|\mathbf{A}\|_F^2$  for randomized Kaczmarz. In uniform mini-batch SGD [30, 25], the step size parameter must be taken even smaller to prevent divergence. Indeed, randomized Kaczmarz and randomized coordinate descent can be interpreted as optimized SGD methods that take maximally large steps while ensuring stability [46].

In summary, RSRI uses a dimension-independent step size  $\alpha \in (0, 2/\|\mathbf{A}\|)$  or  $\alpha \in (0, 2/\|\mathbf{A}\|^2)$ , which can be larger than the SGD step size  $\alpha \in (0, 2|\mathbf{S}|/\text{tr}(\mathbf{A}))$  or  $\alpha \in (0, 2|\mathbf{S}|/\|\mathbf{A}\|_F^2)$  by a factor as high as  $n/|\mathbf{S}|$ . The small step size in SGD leads to a slow, dimension-dependent convergence rate. Indeed, optimizing the convergence rate in (4.4) or (4.5) leads to bounds that are well-known in the case  $|\mathbf{S}| = 1$  [63, 42] and hold if  $(|\mathbf{S}| - 1)(\lambda_{\max}(\mathbf{A}) - \lambda_{\min}(\mathbf{A})) \leq \text{tr}(\mathbf{A})$  or  $(|\mathbf{S}| - 1)(\sigma_{\max}(\mathbf{A})^2 - \sigma_{\min}(\mathbf{A})^2) \leq \|\mathbf{A}\|_F^2$ :

$$\mathbb{E}\|\mathbf{A}^{1/2}(\mathbf{x}_s - \mathbf{x}_\star)\|^2 \leq \left[ 1 - \frac{|\mathbf{S}| \lambda_{\min}(\mathbf{A})}{\text{tr}(\mathbf{A}) + (|\mathbf{S}| - 1) \lambda_{\min}(\mathbf{A})} \right] \mathbb{E}\|\mathbf{A}^{1/2}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2, \\ \text{or } \mathbb{E}\|\mathbf{x}_s - \mathbf{x}_\star\|^2 \leq \left[ 1 - \frac{|\mathbf{S}| \sigma_{\min}(\mathbf{A})^2}{\|\mathbf{A}\|_F^2 + (|\mathbf{S}| - 1) \sigma_{\min}(\mathbf{A})^2} \right] \mathbb{E}\|\mathbf{x}_{s-1} - \mathbf{x}_\star\|^2.$$

See [Appendix A](#) for a derivation. In both randomized coordinate descent and randomized Kaczmarz, the convergence rate is no faster than  $(1 - |\mathbf{S}|/n)^t$  and it is even slower when  $\mathbf{A}$  is ill-conditioned with singular values of varying sizes. Hence, SGD methods need to make many passes over the entries of  $\mathbf{A}$  to obtain a high-accuracy solution.

From an information theoretic perspective, the different convergence rates in SGD and RSRI are due to different access models to the matrix. SGD methods for linear systems are *row access* methods, in which each update depends on a few selected rows of the equation  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . RSRI is a *column access* method, in which each update depends on the output vector  $\mathbf{b}$  and a few selected columns of  $\mathbf{A}$ . Row access methods can outperform dense factorization strategies for solving linear systems [16, 54], and they are especially useful for solving overdetermined least-squares problems with a small number of unknowns [20, 26, 59]. However, in general, a row access method must access a number of rows proportional to the number of unknowns in the solution vector to produce a high-accuracy solution [20, 12].

**4.3. Deterministic sparse approximation.** The third approach to linear systems, which we call deterministic sparse iteration, takes advantage of information in the residual to determine which entries of the approximation to update. For example,

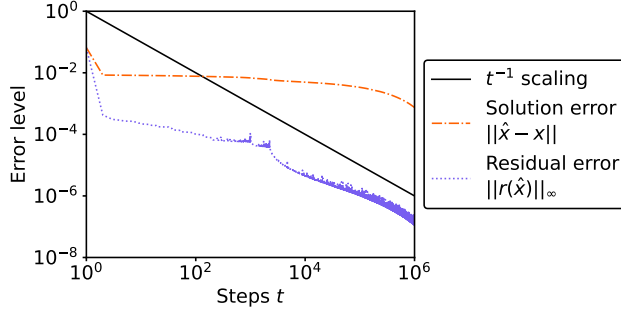


FIG. 2. (*Slow convergence of coordinate descent*). Error  $\|\hat{\mathbf{x}} - \mathbf{x}\|$  and residual error  $\|\mathbf{r}(\hat{\mathbf{x}})\|_\infty$  for coordinate descent with  $t$  update steps, compared to theoretical  $t^{-1}$  scaling. In the figure, we apply coordinate descent to the Amazon electronics PageRank problem, as documented in [subsection 3.3](#).

Cohen, Dahmen, and Devore [14] project the linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$  onto an *active set* of indices and solve the reduced system exactly. To determine the active set, their algorithm trades off between adding new indices corresponding to large-magnitude elements of the residual  $\mathbf{r}(\hat{\mathbf{x}}) = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}}$  and removing indices corresponding to low-magnitude entries of the approximate solution  $\hat{\mathbf{x}}$ .

As another example, the papers [9, 4] develop a deterministic sparse approximation for the PageRank problem  $\mathbf{x} = \alpha\mathbf{P}\mathbf{x} + (1 - \alpha)\mathbf{s}$ . At each step, their method identifies a single index  $i$  for which the residual vector

$$\mathbf{r}(\hat{\mathbf{x}}) = (1 - \alpha)\mathbf{s} - (\mathbf{I} - \alpha\mathbf{P})\hat{\mathbf{x}}$$

is largest, and the algorithm updates  $\hat{\mathbf{x}}(i) = \hat{\mathbf{x}}(i) + \mathbf{r}(\hat{\mathbf{x}})(i)$ . With this choice of coordinate-wise update, the residual vector remains nonnegative and the estimates  $\hat{\mathbf{x}}$  are entry-wise increasing. The algorithm ensures

$$(4.6) \quad \|\mathbf{r}(\hat{\mathbf{x}})\|_\infty < \varepsilon, \quad \text{after a maximum of } t = 1/\varepsilon \text{ steps,}$$

and each step requires accessing just one column of  $\mathbf{P}$  to update the residual.

Compared to the Monte Carlo strategies, the coordinate descent algorithm appears to converge at an accelerated  $\mathcal{O}(t^{-1})$  rate, where  $t$  denotes the number of update steps. However, the convergence is measured in a less meaningful norm. The error bound (4.6) only controls the quality of the residual; it does not control the quality of the solution directly. Additionally, the  $\ell_\infty$  error norm is smaller than the Euclidean norm. The conversion factor can be as high as  $\sqrt{n}$  given entries of the same magnitude, although it is smaller when the entries in the residual vector are rapidly decaying.

Empirically, coordinate descent leads to slow convergence for large-scale problems, as shown in [Figure 2](#). After running the algorithm for  $t$  update steps on the Amazon electronics PageRank problem described in [subsection 3.3](#), we confirm that the residual satisfies  $\|\mathbf{r}(\hat{\mathbf{x}})\|_\infty = \mathcal{O}(t^{-1})$  (purple bottom line). Nonetheless, the error stagnates when measured in the Euclidean norm  $\|\hat{\mathbf{x}} - \mathbf{x}\|$  (orange top line). The underlying problem is that the residual error is spread out over many entries, so each coordinate update produces just a tiny change in  $\hat{\mathbf{x}}$ .

More generally, it remains unclear whether any deterministic sparse iteration method exhibits dimension-independent convergence for a wide class of linear systems. The paper [14] proves specific error bounds for matrices with exponentially decaying

off-diagonal entries. Yet RSRI satisfies more general and powerful error bounds that have not been shown to hold for any deterministic sparse algorithm.

**4.4. Fast randomized iteration.** The work most related to RSRI is “fast randomized iteration” (FRI), which was proposed by Lim & Weare [43] and later developed in the papers [33, 34, 31, 32]. FRI is an approach for speeding up linear or nonlinear fixed-point iterations

$$\mathbf{x}_s = \mathbf{f}(\mathbf{x}_{s-1})$$

by repeatedly applying random sparsification

$$\mathbf{x}_s = \mathbf{f}(\phi_s(\mathbf{x}_{s-1})).$$

Here,  $\phi_s$  is a random sparsification operator, such as the pivotal sparsification operator (Algorithm 5.1).

FRI is an improvement and generalization of the “full configuration interaction quantum Monte Carlo” (FCIQMC) approach [10, 13] for solving eigenvalue problems in quantum chemistry. In FCIQMC, random walkers interact in “a game of life, death, and annihilation” [10] so that a weighted combination of basis vectors on the walkers’ locations approximates the leading eigenvector of a matrix. FCIQMC has been applied to matrices as large as  $10^{108} \times 10^{108}$  [60]. Such large matrices are possible because of the combinatorial explosion of basis elements, where each basis element represents an arrangement of electrons into spatial orbitals. Remarkably, FCIQMC can produce sparse eigenvector approximations for these large systems with eigenvalue errors of just 0.02% using  $\mathcal{O}(10^8)$  random walkers [60].

In contrast to FCIQMC, FRI replaces most of the random operations with deterministic operations in order to improve the efficiency. As a consequence, FRI produces a solution as accurate as FCIQMC but with a number of time steps that is reduced by a factor of  $10^1$ – $10^4$  [33, 34]. However, FRI has not yet been extended to linear systems.

Here, we extend FRI for the first time to solve linear systems by combining the approach with the deterministic Richardson iteration. Additionally, we establish the first error bound that explains FRI’s faster-than- $1/\sqrt{m}$  convergence rate (Theorem 2.1), which has been observed in past studies [33, Fig. 2] but lacked a theoretical explanation. Since the accelerated convergence rate is the main reason to use FRI in applications, our work considerably extends and improves the past FRI analyses [43, 44, 31].

**5. Random sparsification: design and analysis.** In this section, our goal is to design a sparsification operator  $\phi$  that yields small errors as measured in an appropriate norm. The results in this section are of interest beyond the specific setting of RSRI. We address the general question of how to accurately and efficiently approximate a dense vector by a random sparse vector.

Lim & Weare [43] argue that the most meaningful norm for analyzing sparsification error is the “triple norm”, defined for any random vector  $\mathbf{z} \in \mathbb{C}^n$  by

$$\|\mathbf{z}\| = \left( \max_{\|\mathbf{u}\|_\infty \leq 1} \mathbb{E} |\mathbf{u}^* \mathbf{z}|^2 \right)^{1/2}.$$

The triple norm is defined by looking at the worst-case square inner product  $\mathbb{E} |\mathbf{u}^* \mathbf{z}|^2$  with a vector  $\mathbf{u} \in \mathbb{C}^n$  satisfying  $\|\mathbf{u}\|_\infty \leq 1$ .

When we make a random approximation  $\hat{\mathbf{z}} \approx \mathbf{z}$ , we can use the triple norm error  $\|\hat{\mathbf{z}} - \mathbf{z}\|$  to bound the error of any inner product with any vector  $\mathbf{u} \in \mathbb{C}^n$ . When we approximate the dot product  $\mathbf{u}^* \mathbf{z}$  with  $\mathbf{u}^* \hat{\mathbf{z}}$ , the error is bounded by

$$\mathbb{E} |\mathbf{u}^* (\hat{\mathbf{z}} - \mathbf{z})|^2 \leq \|\mathbf{u}\|_\infty^2 \cdot \|\hat{\mathbf{z}} - \mathbf{z}\|^2.$$

The triple norm appears in the right-hand side of the error bound. Additionally, we observe that the triple norm is always larger than the  $L^2$  norm

$$\mathbb{E} \|\hat{\mathbf{z}} - \mathbf{z}\|^2 \leq \|\hat{\mathbf{z}} - \mathbf{z}\|^2$$

(see [Lemma 5.3](#)). In this sense, obtaining bounds in the triple norm is more powerful than obtaining bounds in the conventional  $L^2$  norm.

As we construct the sparsification operator  $\phi$ , we ideally want the operator to satisfy the following design criteria when applied to any vector  $\mathbf{v} \in \mathbb{C}^n$ .

1. The sparsification should yield at most  $m$  nonzero entries, i.e.,  $\|\phi(\mathbf{v})\|_0 \leq m$ .
2. The sparsification should be unbiased, i.e.,  $\mathbb{E}[\phi(\mathbf{v})] = \mathbf{v}$ .
3. The error  $\|\phi(\mathbf{v}) - \mathbf{v}\|$  should be as small as possible, subject to the sparsity and unbiasedness constraints.

In this section, we do not quite succeed in identifying an optimal sparsification operator that satisfies properties 1-3. However, we identify a “pivotal” sparsification operator that satisfies properties 1-2 and nearly optimizes the triple norm error, up to a factor of  $\sqrt{2}$ . See the following new result, which we will prove in [subsection 5.3](#):

**THEOREM 5.1** (Near-optimal error). *For any fixed vector  $\mathbf{v} \in \mathbb{C}^n$  and any random vector  $\mathbf{z} \in \mathbb{C}^n$  satisfying  $\|\mathbf{z}\|_0 \leq m$  and  $\mathbb{E}[\mathbf{z}] = \mathbf{v}$ , the pivotal sparsification operator satisfies*

$$\|\phi_{\text{piv}}(\mathbf{v}) - \mathbf{v}\| \leq \sqrt{2} \|\mathbf{Z} - \mathbf{v}\|.$$

*Additionally, the pivotal sparsification error is bounded by*

$$(5.1) \quad \|\phi_{\text{piv}}(\mathbf{v}) - \mathbf{v}\|^2 \leq \min_{0 \leq i \leq m} \frac{2}{m-i} \left( \sum_{j=i+1}^n |\mathbf{v}^\downarrow(j)| \right)^2.$$

As a result of [Theorem 5.1](#), pivotal sparsification hardly incurs any error if we can arrange the entries  $\mathbf{v}(i)$  so they are decreasing rapidly in magnitude.

The rest of the section is organized as follows. [Subsection 5.1](#) presents pseudocode for pivotal sparsification, [subsection 5.2](#) proves the optimality of pivotal sparsification in the  $L^2$  norm, and [subsection 5.3](#) proves near-optimality of pivotal sparsification in the triple norm.

**5.1. Pseudocode.** [Algorithm 5.1](#) describes the pivotal sparsification algorithm, which takes as input a vector  $\mathbf{v} \in \mathbb{C}^n$  and outputs a sparse random vector  $\phi_{\text{piv}}(\mathbf{v}) \in \mathbb{C}^n$ . The algorithm begins by identifying a set of indices  $D \subseteq \{1, \dots, N\}$  that indicate the largest-magnitude elements of  $\mathbf{v}$ . For simplicity, the pseudocode uses a simple recursive approach for constructing  $D$  by adding just one index  $i$  at a time. To improve the efficiency, we can modify [Algorithm 5.1](#) by adding multiple indices at a time or by preprocessing the input vector to identify the  $m$  largest-magnitude entries [\[22\]](#).

After identifying the largest-magnitude entries, we preserve these entries exactly, that is, we set  $\phi_{\text{piv}}(\mathbf{v})(i) = \mathbf{v}(i)$  for  $i \in D$ . In contrast, we randomly perturb the

**Algorithm 5.1** Pivotal sparsification [31]**Input:** Vector  $\mathbf{v} \in \mathbb{C}^n$ , sparsity parameter  $m$ **Output:** Vector  $\phi(\mathbf{v}) \in \mathbb{C}^n$  with no more than  $m$  nonzero entries that satisfies

$$\mathbb{E}[\phi(\mathbf{v})] = \mathbf{v}, \mathbb{E}|\phi(\mathbf{v})| = |\mathbf{v}|, \text{ and } \|\phi(\mathbf{v})\|_1 = \|\mathbf{v}\|_1.$$

1  $D = \emptyset$   
 2  $q = 0$   
 3 **if** there exists  $i \notin D$  such that  $|\mathbf{v}(i)| \geq \frac{1}{m-q} \sum_{j \notin D} |\mathbf{v}(j)|$  **then**  
 4      $D = D \cup \{i\}$   
 5      $q = q + 1$   
 6 **end if**  
 7 Define  $\mathbf{p} \in [0, 1]^n$  with

$$\mathbf{p}(i) = \begin{cases} 1, & i \in D, \\ (m-q) \cdot |\mathbf{v}(i)| / \sum_{j \notin D} |\mathbf{v}(j)|, & i \notin D. \end{cases}$$

8  $S = \text{sample}(\mathbf{p})$  ▷ Sample using Algorithm 5.2  
 9 Return  $\phi(\mathbf{v}) \in \mathbb{C}^n$  with

$$\phi(\mathbf{v})(i) = \begin{cases} \mathbf{v}(i), & i \in D, \\ \mathbf{v}(i)/\mathbf{p}(i), & i \in S, \\ 0, & i \notin D \cup S. \end{cases}$$

smallest-magnitude entries by setting them to zero or by increasing their magnitudes randomly. We choose which nonzero entries to retain using *pivotal sampling* [17], a stochastic rounding strategy that takes as input a vector of selection probabilities  $\mathbf{p} \in [0, 1]^n$  and rounds each entry  $\mathbf{p}(i)$  to 0 or 1 in an unbiased way, as described in Algorithm 5.2. Pivotal sampling can be implemented using a single pass through the vector of probabilities. All of the indices  $i \notin S$  that are selected by pivotal sampling are raised to an equal magnitude, which is deterministically chosen to preserve the  $\ell_1$ -norm of the input,  $\|\phi_{\text{piv}}(\mathbf{v})\|_1 = \|\mathbf{v}\|_1$ .

**5.2. Optimality in the  $L^2$  norm.** Here, we prove that pivotal sparsification is optimal at controlling error in the  $L^2$  norm. We previously reported a weaker result for pivotal sparsification in [31, Prop. 5.2]. By reworking the proof, we are able to obtain a more explicit bound (5.2), which will be used for bounding the RSRI variance in subsection 6.2.

**PROPOSITION 5.2** (Optimal  $L^2$  error). *For any vector  $\mathbf{v} \in \mathbb{C}^n$  and any random vector  $\mathbf{z} \in \mathbb{C}^n$  satisfying  $\|\mathbf{z}\|_0 \leq m$  and  $\mathbb{E}[\mathbf{z}] = \mathbf{v}$ , it holds that*

$$\mathbb{E}\|\phi_{\text{piv}}(\mathbf{v}) - \mathbf{v}\|^2 \leq \mathbb{E}\|\mathbf{z} - \mathbf{v}\|^2.$$

Additionally, pivotal sparsification satisfies the error bound

$$(5.2) \quad \mathbb{E}\|\phi_{\text{piv}}(\mathbf{v}) - \mathbf{v}\|^2 \leq \min_{0 \leq i \leq m} \frac{1}{m-i} \left( \sum_{j=i+1}^n |\mathbf{v}^\downarrow(j)| \right)^2.$$

*Proof of Proposition 5.2.* The proof is based on an explicit minimization of the square  $L^2$  error  $\mathbb{E}\|\mathbf{z} - \mathbf{v}\|^2$ , subject to the constraints. To that end, we introduce the random vector  $\mathbf{m} \in \{0, 1\}^n$  with entries  $\mathbf{m}(i) = \mathbb{1}\{\mathbf{z}(i) \neq 0\}$ . The entries  $\mathbf{m}(i)$



**Algorithm 5.2** Pivotal sampling [17]**Input:** Vector  $\mathbf{p} \in [0, 1]^n$  with entries that sum to an integer  $m = \sum_{i=1}^n \mathbf{p}(i)$ **Output:** Random set of indices  $S \subseteq \{1, \dots, n\}$  with  $\mathbb{P}\{i \in S\} = \mathbf{p}(i)$ 


---

```

1 Set  $S = \emptyset$ ,  $b = 0$ ,  $\ell = 0$ ,  $f = 1$ 
2 for  $i = 1, \dots, m$  do
3    $u = \max\{k : b + \sum_{j=f}^k \mathbf{p}(j) < 1\}$ 
4   Sample  $h \in \{\ell, f, f+1, \dots, u\}$  with probs. prop. to  $(b, \mathbf{p}(f), \mathbf{p}(f+1), \dots, \mathbf{p}(u))$ 
5    $b = b + \sum_{j=f}^{u+1} \mathbf{p}(j) - 1$ 
6   Set  $\ell = h$  with probability  $(\mathbf{p}(u+1) - b)/(1 - b)$ 
7   if  $\ell = h$  then
8      $S = S \cup \{u+1\}$ 
9   else
10     $S = S \cup \{h\}$ 
11  end if
12   $f = u + 2$ 
13 end for
14 Return  $S$ 

```

---

are thus dependent Bernoulli random variables with success probabilities  $\mathbb{E}[\mathbf{m}(i)]$ . Moreover, the constraint  $\|\mathbf{z}\|_0 \leq m$  implies that  $\sum_{i=1}^n \mathbf{m}(i) \leq m$ , and the constraint  $\mathbb{E}[\mathbf{Z}] = \mathbf{v}$  implies that

$$\mathbf{v}(i) = \mathbb{E}[\mathbf{z}(i)] = \mathbb{E}[\mathbf{z}(i) \mid \mathbf{m}(i) = 1] \cdot \mathbb{E}[\mathbf{m}(i)], \quad \text{for each } 1 \leq i \leq n.$$

Hence, we can decompose the square  $L^2$  error as

$$\begin{aligned} \mathbb{E}\|\mathbf{z} - \mathbf{v}\|^2 &= \sum_{i=1}^n \mathbb{E}|\mathbf{z}(i) - \mathbb{E}[\mathbf{z}(i) \mid \mathbf{m}(i)]|^2 + \sum_{i=1}^n \mathbb{E}|\mathbb{E}[\mathbf{z}(i) \mid \mathbf{m}(i)] - \mathbf{v}(i)|^2 \\ &= \sum_{i=1}^n \mathbb{E}\left|\mathbf{z}(i) - \frac{\mathbf{m}(i)}{\mathbb{E}[\mathbf{m}(i)]} \mathbf{v}(i)\right|^2 + \sum_{i=1}^n \mathbb{E}\left|\frac{\mathbf{m}(i)}{\mathbb{E}[\mathbf{m}(i)]} \mathbf{v}(i) - \mathbf{v}(i)\right|^2. \end{aligned}$$

We minimize the square  $L^2$  error by taking by taking  $\mathbf{z}(i) = \mathbf{v}(i) \cdot \mathbf{m}(i) / \mathbb{E}[\mathbf{m}(i)]$  for each  $1 \leq i \leq n$ , so the first term vanishes and the square  $L^2$  error becomes

$$\mathbb{E}\|\mathbf{z} - \mathbf{v}\|^2 = \sum_{i=1}^n \mathbb{E}\left|\frac{\mathbf{m}(i)}{\mathbb{E}[\mathbf{m}(i)]} \mathbf{v}(i) - \mathbf{v}(i)\right|^2 = \sum_{i=1}^n |\mathbf{v}(i)|^2 \left(\frac{1}{\mathbb{E}[\mathbf{m}(i)]} - 1\right).$$

Next, determine the optimal vector of success probabilities  $\mathbb{E}[\mathbf{m}] = \mathbf{p} \in [0, 1]^n$  by minimizing the objective function

$$(5.3) \quad f(\mathbf{p}) = \sum_{i=1}^n |\mathbf{v}(i)|^2 \left(\frac{1}{\mathbf{p}(i)} - 1\right),$$

subject to the constraints  $\sum_{i=1}^n \mathbf{p}(i) \leq m$  and  $\mathbf{p} \in [0, 1]^n$ .

We observe that  $\mathbf{p} \mapsto f(\mathbf{p})$  is a convex mapping on a closed, convex set. To find the global minimizer, we introduce the Lagrangian function

$$L(\mathbf{p}, \eta, \boldsymbol{\lambda}) = \sum_{i=1}^n |\mathbf{v}(i)|^2 \left(\frac{1}{\mathbf{p}(i)} - 1\right) + \eta \left(\sum_{i=1}^n \mathbf{p}(i) - m\right) + \sum_{i=1}^n \lambda(i) (\mathbf{p}(i) - 1).$$

By the Karush–Kuhn–Tucker conditions [48, pg. 321], the minimizer of (5.3) must satisfy  $\frac{\partial L}{\partial \mathbf{p}(i)} = 0$ , which leads to

$$\mathbf{p}(i) = \frac{|\mathbf{v}(i)|}{(\eta + \boldsymbol{\lambda}(i))^{1/2}}, \quad 1 \leq i \leq n,$$

If  $\eta = 0$ , the complementarity condition  $\boldsymbol{\lambda}(i)(\mathbf{p}(i) - 1) = 0$  implies that

$$(5.4) \quad \mathbf{p}(i) = 1, \quad \text{for each } 1 \leq i \leq n.$$

If  $\eta > 0$ , the complementarity condition implies that

$$(5.5) \quad \mathbf{p}(i) = \min \left\{ \frac{|\mathbf{v}(i)|}{\eta^{1/2}}, 1 \right\}, \quad \text{for each } 1 \leq i \leq n.$$

In this case, the additional complementarity condition  $\eta (\sum_{i=1}^n \mathbf{p}(i) - m) = 0$  implies that  $\sum_{i=1}^n \mathbf{p}(i) = m$ .

To better understand the minimizer, introduce a permutation  $\sigma_1, \dots, \sigma_n$  of the indices  $1, \dots, n$  such that  $|\mathbf{v}(\sigma_1)| \geq \dots \geq |\mathbf{v}(\sigma_n)|$ . In light of (5.4) and (5.5), there must be an exact preservation threshold  $q \in 0, \dots, n$  such that

$$\begin{aligned} \mathbf{p}(\sigma_i) &= 1, & i &\leq q, \\ \mathbf{p}(\sigma_i) &= \frac{(m-q)|\mathbf{v}(\sigma_i)|}{\sum_{j=q+1}^n |\mathbf{v}(\sigma_j)|} < 1, & q+1 \leq i \leq n. \end{aligned}$$

The objective function (5.3) can be rewritten as

$$(5.6) \quad f(q) = \frac{1}{m-q} \left( \sum_{i=q+1}^n |\mathbf{v}(\sigma_i)| \right)^2 - \sum_{i=q+1}^n |\mathbf{v}(\sigma_i)|^2,$$

and we can find the optimal  $q$  by minimizing (5.6) subject to the constraint

$$|\mathbf{v}(\sigma_{q+1})| < \frac{1}{m-q} \sum_{i=q+1}^n |\mathbf{v}(\sigma_i)|.$$

A direct computation reveals

$$f(q+1) - f(q) = \frac{m-q}{m-q-1} \left( |\mathbf{v}(\sigma_{q+1})| - \frac{1}{m-q} \sum_{i=q+1}^n |\mathbf{v}(\sigma_i)| \right)^2 \geq 0,$$

wherefore the objective function  $f(q)$  is nondecreasing in  $q$ . The optimal threshold is

$$(5.7) \quad q_* = \min \left\{ 0 \leq q \leq m : |\mathbf{v}(\sigma_{q+1})| < \frac{1}{m-q} \sum_{i=q+1}^n |\mathbf{v}(\sigma_i)| \right\}.$$

This matches the description of pivotal sparsification given in Algorithm 5.2.

Last, to bound the square  $L^2$  error of pivotal sparsification, we introduce

$$g(q) = \frac{1}{m-q} \left( \sum_{i=q+1}^n |\mathbf{v}(\sigma_i)| \right)^2$$

which is an upper bound on the objective function  $f(q)$  (5.6). A direct calculation shows that

$$g(q) - g(q+1) = \left( \frac{1}{m-q-1} \sum_{i=q+2}^n |\mathbf{v}(\sigma_i)| \right) \left[ |\mathbf{v}(\sigma_{q+1})| - \frac{1}{m-q} \sum_{i=q+1}^n |\mathbf{v}(\sigma_i)| \right].$$

Hence, incrementing  $q$  will reduce or leave unchanged the value of  $g(q)$  as long as

$$(5.8) \quad |\mathbf{v}(\sigma_{q+1})| \geq \frac{1}{m-q} \sum_{i=q+1}^n |\mathbf{v}(\sigma_i)|$$

is satisfied. From the characterization (5.7), there must be a minimizer  $\tilde{q}$  of  $g(q)$  satisfying  $q_* \leq \tilde{q}$ . Since the objective function  $f(q)$  is nondecreasing, we conclude

$$f(q_*) \leq f(\tilde{q}) \leq g(\tilde{q}) = \min_{0 \leq q \leq m} g(q).$$

We rewrite this inequality as

$$\mathbb{E} \|\phi_{\text{piv}}(\mathbf{v}) - \mathbf{v}\|^2 \leq \min_{0 \leq i \leq m} \frac{1}{m-i} \left( \sum_{j=i+1}^n |\mathbf{v}^\perp(j)| \right)^2,$$

to complete the proof.  $\square$

**5.3. Near-optimality in the triple norm.** In this section, we prove that pivotal sparsification is nearly optimal at controlling error in the triple norm. This result is entirely new.

To prove the near-optimality result, we need two lemmas. First, we prove a helpful comparison between the  $L^2$  norm and the triple norm. A similar result was presented in [43, Eq. 27], but the argument here is simpler.

**LEMMA 5.3** (Difference in norms). *Fix any random vector  $\mathbf{z} \in \mathbb{C}^n$ , and let  $\mathbf{u} \in \mathbb{C}^n$  be an independent random vector with entries that are independent and uniformly distributed on the complex unit circle, so  $|\mathbf{u}(1)| = \dots = |\mathbf{u}(n)| = 1$ . It holds that*

$$\mathbb{E} \|\mathbf{z}\|^2 = \mathbb{E} |\mathbf{u}^* \mathbf{z}|^2 \quad \text{and} \quad \|\mathbf{z}\|^2 = \max_{|\mathbf{v}(1)| = \dots = |\mathbf{v}(n)| = 1} \mathbb{E} |\mathbf{v}^* \mathbf{z}|^2.$$

Consequently,  $\mathbb{E} \|\mathbf{z}\|^2 \leq \|\mathbf{z}\|^2$ .

*Proof.* By direct calculation

$$\mathbb{E} \|\mathbf{z}\|^2 = \sum_{i=1}^n \mathbb{E} [\overline{\mathbf{z}(i)} \mathbf{z}(i)] = \sum_{i,j=1}^n \mathbb{E} [\overline{\mathbf{u}(i) \mathbf{z}(i)} \mathbf{u}(j) \mathbf{z}(j)] = \mathbb{E} |\mathbf{u}^* \mathbf{z}|^2.$$

This calculation uses the fact that  $\mathbf{u}(i)$  are uncorrelated, mean-zero, variance-one random variables.

Next, since  $\mathbf{v} \mapsto \mathbb{E} |\mathbf{v}^* \mathbf{z}|^2$  is a convex function, the maximum  $\max_{\|\mathbf{v}\|_\infty \leq 1} \mathbb{E} |\mathbf{v}^* \mathbf{z}|^2$  is achieved on an extreme point of the closed convex set  $\{\mathbf{v} \in \mathbb{C}^n : \|\mathbf{v}\|_\infty \leq 1\}$ . The extreme points are vectors  $\mathbf{v} \in \mathbb{C}^n$  with entries on the complex unit circle,  $|\mathbf{v}(1)| = \dots = |\mathbf{v}(n)| = 1$ .

The comparison between  $\mathbb{E} \|\mathbf{z}\|^2$  and  $\|\mathbf{z}\|^2$  then follows, since the inner product of  $\mathbf{z}$  with a random vector  $\mathbf{u}$  has a smaller mean square magnitude than the inner product with a worst-case vector  $\mathbf{v}$ .  $\square$

Next, we recall a lemma establishing the negative correlations of pivotal sampling, which was proved by Srinivasan [62].

**LEMMA 5.4** (Negative correlations [62]). *Given a vector of probabilities  $\mathbf{p} \in [0, 1]^n$ , pivotal sampling (Algorithm 5.2) returns a set  $S \subseteq \{1, \dots, n\}$  with negatively correlated selections, i.e.,*

$$(5.9) \quad \mathbb{P}\{i, j \in S\} \leq \mathbf{p}(i) \cdot \mathbf{p}(j), \quad 1 \leq i, j \leq n.$$

We are ready to prove [Theorem 5.1](#), which establishes the near-optimality of pivotal sparsification in the triple norm.

*Proof of Theorem 5.1.* Fix a vector  $\mathbf{v} \in \mathbb{C}^n$  and another vector  $\mathbf{u} \in \mathbb{C}^n$  satisfying  $\|\mathbf{u}\|_\infty \leq 1$ . Since pivotal sampling leads to negative selection probabilities ([Lemma 5.4](#)), the pivotal sparsification operator  $\phi = \phi_{\text{piv}}$  satisfies

$$\mathbb{E} \left[ \frac{\phi(\mathbf{v})(i)}{\mathbf{v}(i)} \cdot \frac{\phi(\mathbf{v})(j)}{\mathbf{v}(j)} \right] \leq 1, \quad \text{if } \mathbf{v}(i) \neq 0 \text{ and } \mathbf{v}(j) \neq 0.$$

Now define the index sets

$$\mathbf{P} = \{1 \leq i \leq n \mid \operatorname{Re}\{\overline{\mathbf{u}(i)}\mathbf{v}(i)\} > 0\} \quad \text{and} \quad \mathbf{N} = \{1 \leq i \leq n \mid \operatorname{Re}\{\overline{\mathbf{u}(i)}\mathbf{v}(i)\} < 0\}.$$

For each pair of indices  $i, j \in \mathbf{P}$  or  $i, j \in \mathbf{N}$ , it follows

$$\begin{aligned} & \mathbb{E}[\operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\} \cdot \operatorname{Re}\{\overline{\mathbf{u}(j)}\phi(\mathbf{v})(j)\}] \\ &= \operatorname{Re}\{\overline{\mathbf{u}(i)}\mathbf{v}(i)\} \cdot \operatorname{Re}\{\overline{\mathbf{u}(j)}\mathbf{v}(j)\} \cdot \mathbb{E} \left[ \frac{\phi(\mathbf{v})(i)}{\mathbf{v}(i)} \cdot \frac{\phi(\mathbf{v})(j)}{\mathbf{v}(j)} \right] \\ &\leq \operatorname{Re}\{\overline{\mathbf{u}(i)}\mathbf{v}(i)\} \cdot \operatorname{Re}\{\overline{\mathbf{u}(j)}\mathbf{v}(j)\}. \end{aligned}$$

Therefore the following covariance is negative:

$$(5.10) \quad \operatorname{Cov}[\operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\}, \operatorname{Re}\{\overline{\mathbf{u}(j)}\phi(\mathbf{v})(j)\}] \leq 0.$$

The negative covariance relation (5.10) allows us to calculate

$$\begin{aligned} \operatorname{Var}[\operatorname{Re}\{\mathbf{u}^* \phi(\mathbf{v})\}] &= \operatorname{Var} \left[ \sum_{i=1}^n \operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\} \right] \\ &\leq 2 \operatorname{Var} \left[ \sum_{i \in \mathbf{P}} \operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\} \right] + 2 \operatorname{Var} \left[ \sum_{i \in \mathbf{N}} \operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\} \right] \\ &\leq 2 \sum_{i \in \mathbf{P}} \operatorname{Var}[\operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\}] + 2 \sum_{i \in \mathbf{N}} \operatorname{Var}[\operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\}] \\ &= 2 \sum_{i=1}^n \operatorname{Var}[\operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\}]. \end{aligned}$$

Similarly, we calculate

$$\operatorname{Var}[\operatorname{Im}\{\mathbf{u}^* \phi(\mathbf{v})\}] \leq 2 \sum_{i=1}^n \operatorname{Var}[\operatorname{Im}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\}].$$

By combining the real and imaginary parts,

$$\begin{aligned} \operatorname{Var}[\mathbf{u}^* \phi(\mathbf{v})] &= \operatorname{Var}[\operatorname{Re}\{\mathbf{u}^* \phi(\mathbf{v})\}] + \operatorname{Var}[\operatorname{Im}\{\mathbf{u}^* \phi(\mathbf{v})\}] \\ &\leq 2 \sum_{i=1}^n \operatorname{Var}[\operatorname{Re}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\}] + 2 \sum_{i=1}^n \operatorname{Var}[\operatorname{Im}\{\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)\}] \\ &= 2 \sum_{i=1}^n \operatorname{Var}[\overline{\mathbf{u}(i)}\phi(\mathbf{v})(i)] \\ &\leq 2 \sum_{i=1}^n \operatorname{Var}[\phi(\mathbf{v})(i)] = 2 \mathbb{E} \|\phi(\mathbf{v}) - \mathbf{v}\|^2. \end{aligned}$$

The last line bounds the variance using the fact that  $\|\mathbf{u}\|_\infty \leq 1$ .

Last, for any random vector  $\mathbf{z} \in \mathbb{C}^n$  satisfying  $\|\mathbf{z}\|_0 \leq m$  and  $\mathbb{E}[\mathbf{z}] = \mathbf{v}$ , [Proposition 5.2](#) and [Lemma 5.3](#) allow us to calculate

$$2 \mathbb{E} \|\phi(\mathbf{v}) - \mathbf{v}\|^2 \leq 2 \mathbb{E} \|\mathbf{Z} - \mathbf{v}\|^2 \leq 2 \|\mathbf{Z} - \mathbf{v}\|^2.$$

We combine with the variance bound from [Proposition 5.2](#) to show

$$\|\phi(\mathbf{v}) - \mathbf{v}\|^2 \leq \min_{0 \leq i \leq m} \frac{2}{m-i} \left( \sum_{j=i+1}^n |\mathbf{v}^\downarrow(j)| \right)^2.$$

This verifies the error bound [\(5.1\)](#) and completes the proof.  $\square$

Last, by examining the proof of [Theorem 5.1](#), we record a simple corollary which allows us to sometimes remove a factor of two from the error bound.

**COROLLARY 5.5** (Nonnegative sparsification bound). *For any vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{C}^n$  satisfying  $\operatorname{Re}\{\overline{\mathbf{u}(i)}\mathbf{v}(i)\} \geq 0$  and  $\operatorname{Im}\{\overline{\mathbf{u}(i)}\mathbf{v}(i)\} \geq 0$  for all  $1 \leq i \leq n$ , the pivotal sparsification error is bounded by*

$$\mathbb{E}|\mathbf{f}^*(\phi_{\text{piv}}(\mathbf{v}) - \mathbf{v})|^2 \leq \min_{0 \leq i \leq m} \frac{1}{m-i} \left( \sum_{j=i+1}^n |\mathbf{v}^\downarrow(j)| \right)^2.$$

**6. RSRI error bounds.** Our main goal in this section is to prove [Theorem 6.1](#), which is a stronger version of [Theorem 2.1](#) from the introduction.

**THEOREM 6.1** (Extended error bound). *Suppose RSRI with sparsity level  $m$  is applied to an  $n \times n$  linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , and set  $\mathbf{G} = \mathbf{I} - \mathbf{A}$ . Then, RSRI returns a solution  $\bar{\mathbf{x}}_t$  satisfying the bias-variance formula*

$$(6.1) \quad \|\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{b}\|^2 \leq \underbrace{\|\mathbf{A}\mathbb{E}[\bar{\mathbf{x}}_t] - \mathbf{b}\|_1^2}_{\text{bias}^2} + \underbrace{\|\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{A}\mathbb{E}[\bar{\mathbf{x}}_t]\|^2}_{\text{variance}}.$$

The square bias is bounded by

$$\text{bias}^2 \leq \left( \frac{2 \sup_{s \geq 0} \|\mathbf{G}^s\|_1 \cdot \|\mathbf{G}^{t_b} \mathbf{x}_*\|_1}{t - t_b} \right)^2,$$

where  $\mathbf{x}_*$  is the exact solution. If  $\mathbf{G}$  is a strict 1-norm contraction,  $\|\mathbf{G}\|_1 < 1$ , the variance is bounded by

$$\text{variance} \leq \frac{8t}{(t - t_b)^2} \cdot \frac{1}{m} \left( \frac{\|\mathbf{b}\|_1}{1 - \|\mathbf{G}\|_1} \right)^2.$$

Alternately, if  $m \geq m_{\mathbf{G}} = \sum_{s=0}^{\infty} \|\mathbf{G}^s\|_1^2$ , the variance is bounded by

$$\text{variance} \leq \frac{8t \sup_{s \geq 0} \|\mathbf{G}^s\|_1^2}{(t - t_b)^2} \cdot \min_{0 \leq i \leq m - m_{\mathbf{G}}} \frac{1}{m - m_{\mathbf{G}} - i} \left( \sum_{j=i+1}^n \tilde{\mathbf{x}}^\downarrow(j) \right)^2.$$

Here,  $\tilde{\mathbf{x}} \in \mathbb{R}^n$  is the solution to the regularized linear system  $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ , where  $\tilde{\mathbf{A}} = \mathbf{I} - |\mathbf{G}|$ ,  $\tilde{\mathbf{b}} = |\mathbf{b}|$ , and  $\tilde{\mathbf{x}}^\downarrow \in \mathbb{R}^n$  is the decreasing rearrangement of  $\tilde{\mathbf{x}}$ .

*Proof.* For any  $\mathbf{u} \in \mathbb{C}^n$ , we observe

$$\mathbb{E}|\mathbf{u}^*(\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{b})|^2 = |\mathbf{u}^*(\mathbf{A}\mathbb{E}[\bar{\mathbf{x}}_t] - \mathbf{b})|^2 + \mathbb{E}|\mathbf{u}^*(\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{A}\mathbb{E}[\bar{\mathbf{x}}_t])|^2.$$

By taking the supremum over all  $\mathbf{u}$  satisfying  $\|\mathbf{u}\|_\infty \leq 1$ , we establish the bias-variance formula [\(6.1\)](#). We will prove the bias bounds in [subsection 6.1](#) and the variance bounds in [subsection 6.2](#).  $\square$

**Theorem 6.1** demonstrates that raising the sparsity threshold  $m$  has multiple benefits. First, raising  $m$  leads to faster-than- $1/\sqrt{m}$  convergence if the regularized solution  $\tilde{\mathbf{x}}$  has rapidly decaying entries. Second, raising  $m$  extends RSRI to any system in which the largest eigenvalue of  $|\mathbf{G}|$  is strictly less than one. Indeed, RSRI is guaranteed to converge as  $t \rightarrow \infty$ , if we set

$$m \geq m_{\mathbf{G}} = \sum_{s=0}^{\infty} \|\mathbf{G}^s\|_1^2.$$

In comparison, the deterministic Richardson iteration ([Algorithm 1.1](#)) converges if the spectral radius of  $\mathbf{G}$  is strictly less than one. The spectral radius of  $\mathbf{G}$  is always bounded from above by the largest eigenvalue of  $|\mathbf{G}|$  [[36](#), Thm. 8.3.2].

To measure error, [Theorem 6.1](#) uses the triple norm, which is more useful than the  $L^2$  norm. Thanks to our use of the triple norm, we are able to transfer our results to the PageRank problem and verify [Proposition 3.1](#).

*Proof of Proposition 3.1.* In the PageRank problem  $\mathbf{A}\mathbf{x} = \mathbf{b}$  where  $\mathbf{A} = \mathbf{I} - \alpha\mathbf{P}$  and  $\mathbf{b} = (1 - \alpha)\mathbf{s}$ , we observe that  $\mathbf{A}^{-1} = \sum_{s=0}^{\infty} \alpha^s \mathbf{P}^s$  has nonnegative columns that sum to  $\sum_{s=0}^{\infty} \alpha^s = 1/(1 - \alpha)$ . Therefore,  $\|\mathbf{A}^{-1}\|_1 = 1/(1 - \alpha)$ , and we calculate

$$\begin{aligned} \|\bar{\mathbf{x}}_t - \mathbf{x}_\star\|^2 &= \max_{\|\mathbf{u}\|_\infty \leq 1} \mathbb{E} \|\mathbf{u}^* (\bar{\mathbf{x}}_t - \mathbf{x}_\star)\|^2 \\ &\leq \|\mathbf{A}^{-1}\|_1^2 \max_{\|\mathbf{u}\|_\infty \leq 1} \mathbb{E} \|\mathbf{u}^* (\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{b})\|^2 = \|\mathbf{A}^{-1}\|_1^2 \|\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{b}\|^2 \\ &= \frac{1}{(1 - \alpha)^2} \|\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{b}\|^2. \end{aligned}$$

To verify [Proposition 3.1](#), we apply the bias and variance bounds from [Theorem 6.1](#) and use the fact that the regularized system  $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$  is the same as the original system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ .  $\square$

**6.1. Bias bound.** In this section, we bound RSRI's bias. RSRI has the same bias that would result from deterministic Richardson iteration ([Algorithm 1.1](#)) after averaging the iterates  $\mathbf{x}_{t_b}, \dots, \mathbf{x}_{t-1}$ , and we prove the following bound.

**PROPOSITION 6.2 (Bias bound).** *Suppose RSRI is applied to a linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , and set  $\mathbf{G} = \mathbf{I} - \mathbf{A}$ . RSRI returns an approximation  $\bar{\mathbf{x}}_t$  of the exact solution  $\mathbf{x}_\star$  with bias bounded by*

$$\|\mathbf{A} \mathbb{E}[\bar{\mathbf{x}}_t] - \mathbf{b}\|_1 \leq \frac{2 \sup_{s \geq 0} \|\mathbf{G}^s\|_1 \cdot \|\mathbf{G}^{t_b} \mathbf{x}_\star\|_1}{t - t_b}.$$

*Proof.* We can determine the bias of RSRI from the recursion

$$\mathbb{E}[\mathbf{x}_s] = \mathbf{G} \mathbb{E}[\mathbf{x}_{s-1}] + \mathbf{b}.$$

The recursion leads to

$$\mathbb{E}[\mathbf{x}_s] = \sum_{r=0}^{s-1} \mathbf{G}^r \mathbf{b} = \mathbf{x}_\star - \mathbf{G}^s \mathbf{x}_\star,$$

where we have substituted  $\mathbf{b} = \mathbf{A}\mathbf{x}_\star = (\mathbf{I} - \mathbf{G})\mathbf{x}_\star$ . Using  $\bar{\mathbf{x}}_t = \frac{1}{t - t_b} \sum_{s=t_b}^{t-1} \mathbf{x}_s$  and

$\mathbf{A} = \mathbf{I} - \mathbf{G}$ , we calculate

$$\begin{aligned} \mathbf{A} \mathbb{E}[\bar{\mathbf{x}}_t] - \mathbf{b} &= (\mathbf{I} - \mathbf{G}) \left( \frac{1}{t - t_b} \sum_{s=t_b}^{t-1} \mathbb{E}[\mathbf{x}_s] \right) - (\mathbf{I} - \mathbf{G}) \mathbf{x}_\star \\ &= (\mathbf{I} - \mathbf{G}) \left( -\frac{1}{t - t_b} \sum_{s=t_b}^{t-1} \mathbf{G}^s \mathbf{x}_\star \right) \\ &= \frac{1}{t - t_b} (\mathbf{G}^t - \mathbf{G}^{t_b}) \mathbf{x}_\star. \end{aligned}$$

We apply the series of upper bounds

$$(6.2) \quad \|\mathbf{A} \mathbb{E}[\bar{\mathbf{x}}_t] - \mathbf{b}\|_1 = \frac{1}{t - t_b} \|(\mathbf{G}^t - \mathbf{G}^{t_b}) \mathbf{x}_\star\|_1$$

$$(6.3) \quad \leq \frac{1}{t - t_b} \|\mathbf{G}^{t-t_b} - \mathbf{I}\|_1 \|\mathbf{G}^{t_b} \mathbf{x}_\star\|_1$$

$$(6.4) \quad \leq \frac{2}{t - t_b} \sup_{s \geq 0} \|\mathbf{G}^s\|_1 \|\mathbf{G}^{t_b} \mathbf{x}_\star\|_1,$$

where (6.3) is due to the submultiplicativity of the matrix 1-norm and (6.4) is due to the subadditivity of the matrix 1-norm.  $\square$

**6.2. Variance bounds.** In this section, we bound the variance of RSRI. To that end, we will need to bound the maximum square sparsification error

$$(6.5) \quad \sup_{t \geq 0} \|\phi_{t+1}(\mathbf{x}_t) - \mathbf{x}_t\|^2.$$

We first analyze (6.5) in the simple case that  $\mathbf{G}$  is a strict one-norm contraction.

**PROPOSITION 6.3** (Strict one-norm contraction bound). *Suppose RSRI with sparsity level  $m$  is applied to a linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , and set  $\mathbf{G} = \mathbf{I} - \mathbf{A}$ . If  $\|\mathbf{G}\|_1 < 1$ , the maximum square sparsification error is bounded by*

$$(6.6) \quad \sup_{t \geq 0} \|\phi_{t+1}(\mathbf{x}_t) - \mathbf{x}_t\|^2 \leq \frac{2}{m} \left( \frac{\|\mathbf{b}\|_1}{1 - \|\mathbf{G}\|_1} \right)^2.$$

*Proof.* The RSRI iterates are generated according to the recursion

$$\mathbf{x}_s = \mathbf{G} \phi_s(\mathbf{x}_{s-1}) + \mathbf{b},$$

which leads to the recursive inequality

$$\|\mathbf{x}_s\|_1 \leq \|\mathbf{G}\|_1 \|\phi_s(\mathbf{x}_{s-1})\|_1 + \|\mathbf{b}\|_1 = \|\mathbf{G}\|_1 \|\mathbf{x}_{s-1}\|_1 + \|\mathbf{b}\|_1,$$

where we use the fact that  $\|\phi_s(\mathbf{x}_{s-1})\|_1 = \|\mathbf{x}_{s-1}\|_1$  with probability one. The recursive inequality leads to the upper bound

$$\|\mathbf{x}_t\|_1 \leq \sum_{s=0}^t \|\mathbf{G}\|_1^s \|\mathbf{b}\|_1 \leq \frac{\|\mathbf{b}\|_1}{1 - \|\mathbf{G}\|_1}.$$

Last, we apply Theorem 5.1 to calculate

$$\|\phi_{t+1}(\mathbf{x}_t) - \mathbf{x}_t\|^2 \leq \frac{2}{m} \mathbb{E} \|\mathbf{x}_t\|_1^2 \leq \frac{2}{m} \left( \frac{\|\mathbf{b}\|_1}{1 - \|\mathbf{G}\|_1} \right)^2,$$

which confirms (6.6) and completes the proof.  $\square$



To obtain a more powerful bound on (6.5), we need a lemma that controls the absolute values of the entries of the RSRI iterates  $\mathbf{x}_0, \mathbf{x}_1, \dots$

LEMMA 6.4 (Stability lemma). *Suppose RSRI is applied to a linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , and set  $\mathbf{G} = \mathbf{I} - \mathbf{A}$ . Then, the RSRI iterates  $\mathbf{x}_0, \mathbf{x}_1, \dots$  satisfy*

$$(6.7) \quad \mathbb{E}|\mathbf{x}_t| \leq \sum_{s=0}^{t-1} |\mathbf{G}|^s |\mathbf{b}|, \quad \text{for each } t \geq 0.$$

*Proof.* At any iteration  $s \geq 0$ , the RSRI iterates satisfy

$$|\mathbf{x}_s| \leq |\mathbf{G}| |\phi_s(\mathbf{x}_{s-1})| + |\mathbf{b}|.$$

Taking expectations and using the fact that  $\mathbb{E}|\phi_s(\mathbf{x}_{s-1})| = \mathbb{E}|\mathbf{x}_{s-1}|$ , we obtain the recursion

$$\mathbb{E}|\mathbf{x}_s| \leq |\mathbf{G}| \mathbb{E}|\phi_s(\mathbf{x}_{s-1})| + |\mathbf{b}| = |\mathbf{G}| \mathbb{E}|\mathbf{x}_{s-1}| + |\mathbf{b}|$$

Since  $\mathbf{x}_0 = \mathbf{0}$ , this recursion validates the error bound (6.7).  $\square$

Last, we establish a more powerful error bound for the maximum square sparsification error (6.5).

PROPOSITION 6.5 (More powerful bound). *Suppose RSRI is applied to an  $n \times n$  linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , and set  $\mathbf{G} = \mathbf{I} - \mathbf{A}$ . If the sparsity level satisfies  $m \geq m_{\mathbf{G}} = \sum_{s=0}^{\infty} \|\mathbf{G}|^s\|_1^2$ , the maximum square sparsification error is bounded by*

$$\sup_{t \geq 0} \|\phi_{t+1}(\mathbf{x}_t) - \mathbf{x}_t\|^2 \leq \min_{0 \leq i \leq m - m_{\mathbf{G}}} \frac{2}{m - m_{\mathbf{G}} - i} \left( \sum_{j=i+1}^n \tilde{\mathbf{x}}^{\downarrow}(j) \right)^2.$$

Here,  $\tilde{\mathbf{x}} \in \mathbb{R}^n$  is the solution to the regularized linear system  $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ , where  $\tilde{\mathbf{A}} = \mathbf{I} - |\mathbf{G}|$ ,  $\tilde{\mathbf{b}} = |\mathbf{b}|$ , and  $\tilde{\mathbf{x}}^{\downarrow} \in \mathbb{R}^n$  is the decreasing rearrangement of  $\tilde{\mathbf{x}}$ .

*Proof.* Fix a set  $\mathbf{E} \subseteq \{1, \dots, N\}$  with  $|\mathbf{E}| \leq m - m_{\mathbf{G}}$ , and let  $\mathbf{u} \in \{0, 1\}^n$  be defined by  $\mathbf{u}(i) = 0$  for  $i \in \mathbf{E}$  and  $\mathbf{u}(i) = 1$  for  $i \notin \mathbf{E}$ . Then, Theorem 5.1 guarantees

$$(6.8) \quad \|\phi_{t+1}(\mathbf{x}_t) - \mathbf{x}_t\|^2 \leq \frac{2}{m - |\mathbf{E}|} \cdot \mathbb{E}|\mathbf{u}^*|\mathbf{x}_t|^2$$

We will proceed to derive an error bound on  $\mathbb{E}|\mathbf{u}^*|\mathbf{x}_t|^2$ .

For any  $0 \leq s \leq t - 1$ , we make the following calculation, which is based on expanding the square and using Lemma 6.4 to calculate the expectations:

$$\begin{aligned} & \mathbb{E}|\mathbf{u}^*|\mathbf{G}|^s|\mathbf{x}_{t-s}|^2 - \mathbb{E}|\mathbf{u}^*|\mathbf{G}|^{s+1}|\phi_{t-s}(\mathbf{x}_{t-s-1})|^2 \\ & \leq \mathbb{E}|\mathbf{u}^*|\mathbf{G}|^{s+1}|\phi_{t-s}(\mathbf{x}_{t-s-1})| + |\mathbf{u}^*|\mathbf{G}|^s|\mathbf{b}|^2 - \mathbb{E}|\mathbf{u}^*|\mathbf{G}|^{s+1}|\phi_{t-s}(\mathbf{x}_{t-s-1})|^2 \\ (6.9) \quad & \leq 2 \left( \sum_{r=s+1}^{t-1} \mathbf{u}^*|\mathbf{G}|^r|\mathbf{b}| \right) (\mathbf{u}^*|\mathbf{G}|^s|\mathbf{b}|) + (\mathbf{u}^*|\mathbf{G}|^s|\mathbf{b}|)^2 \\ & = \left( \sum_{r=s}^{t-1} \mathbf{u}^*|\mathbf{G}|^r|\mathbf{b}| \right)^2 - \left( \sum_{r=s+1}^{t-1} \mathbf{u}^*|\mathbf{G}|^r|\mathbf{b}| \right)^2. \end{aligned}$$

Next, we introduce a vector  $\mathbf{w} \in \mathbb{C}^n$  with entries

$$\mathbf{w}(i) = \begin{cases} \mathbf{u}^*|\mathbf{G}|^{s+1}\mathbf{e}_i \cdot \frac{\mathbf{x}_{t-s}(i)}{|\mathbf{x}_{t-s}(i)|}, & \mathbf{x}_{t-s}(i) \neq 0, \\ 0, & \mathbf{x}_{t-s}(i) = 0. \end{cases}$$

and observe that

$$\|\mathbf{w}\|_\infty = \max_{1 \leq i \leq n} \mathbf{u}^* |\mathbf{G}|^{s+1} \mathbf{e}_i \leq \|\mathbf{G}\|_1^{s+1}.$$

We make the following calculation, which is based on the conditional expectation  $\mathbb{E}[|\phi_{t-s}(\mathbf{x}_{t-s-1})| | \mathbf{x}_{t-s-1}] = |\mathbf{x}_{t-s-1}|$  and an application of [Corollary 5.5](#) to bound the variance:

$$\begin{aligned} (6.10) \quad & \mathbb{E}|\mathbf{u}^* |\mathbf{G}|^{s+1} |\phi_{t-s}(\mathbf{x}_{t-s-1})|^2 - \mathbb{E}|\mathbf{u}^* |\mathbf{G}|^{s+1} |\mathbf{x}_{t-s-1}|^2 \\ &= \mathbb{E}|\mathbf{u}^* |\mathbf{G}|^{s+1} [|\phi_{t-s}(\mathbf{x}_{t-s-1})| - |\mathbf{x}_{t-s-1}|]^2 \\ &= \mathbb{E}|\mathbf{w}^* [\phi_{t-s}(\mathbf{x}_{t-s-1}) - \mathbf{x}_{t-s-1}]|^2 \\ &\leq \frac{\|\mathbf{G}\|_1^{s+1}}{m - |\mathbf{E}|} \cdot \mathbb{E}|\mathbf{u}^* |\mathbf{x}_{t-s-1}|^2 \end{aligned}$$

Adding (6.9) to (6.10), we find that

$$\begin{aligned} (6.11) \quad & \mathbb{E}|\mathbf{u}^* |\mathbf{G}|^s |\mathbf{x}_{t-s}|^2 - \mathbb{E}|\mathbf{u}^* |\mathbf{G}|^{s+1} |\mathbf{x}_{t-s-1}|^2 \\ &\leq \left( \sum_{r=s}^{t-1} \mathbf{u}^* |\mathbf{G}|^r |\mathbf{b}| \right)^2 - \left( \sum_{r=s+1}^{t-1} \mathbf{u}^* |\mathbf{G}|^r |\mathbf{b}| \right)^2 + \frac{\|\mathbf{G}\|_1^{s+1}}{m - |\mathbf{E}|} \cdot \mathbb{E}|\mathbf{u}^* |\mathbf{x}_{t-s-1}|^2. \end{aligned}$$

We sum over (6.11) for  $s = 0, 1, \dots, t-1$  to obtain the recursion

$$\mathbb{E}|\mathbf{u}^* |\mathbf{x}_t|^2 \leq \left( \sum_{r=0}^{t-1} \mathbf{u}^* |\mathbf{G}|^r |\mathbf{b}| \right)^2 + \frac{1}{m - |\mathbf{E}|} \sum_{s=1}^t \|\mathbf{G}\|_1^s \mathbb{E}|\mathbf{u}^* |\mathbf{x}_{t-s}|^2.$$

The recursion implies that

$$\sup_{t \geq 0} \mathbb{E}|\mathbf{u}^* |\mathbf{x}_t|^2 \leq \frac{\left( \sum_{s=0}^{\infty} \mathbf{u}^* |\mathbf{G}|^s |\mathbf{b}| \right)^2}{1 - \frac{1}{m - |\mathbf{E}|} \sum_{s=0}^{\infty} \|\mathbf{G}\|_1^s} = \frac{\left( \sum_{i \notin \mathbf{E}} \tilde{\mathbf{x}}(i) \right)^2}{1 - \frac{m_{\mathbf{G}}}{m - |\mathbf{E}|}},$$

where we have identified  $\tilde{\mathbf{x}} = \sum_{s=0}^{\infty} |\mathbf{G}|^s |\mathbf{b}|$  and  $m_{\mathbf{G}} = \sum_{s=0}^{\infty} \|\mathbf{G}\|_1^s$ . Combining with (6.8), we verify

$$\sup_{t \geq 0} \|\phi_{t+1}(\mathbf{x}_t) - \mathbf{x}_t\|^2 \leq \frac{2}{m - m_{\mathbf{G}} - |\mathbf{E}|} \left( \sum_{i \notin \mathbf{E}} \tilde{\mathbf{x}}(i) \right)^2.$$

Last, we optimize over the set  $\mathbf{E}$  to complete the proof.  $\square$

Now, we are ready to bound the RSRI variance and complete the proof of [Theorem 6.1](#).

**PROPOSITION 6.6** (Variance bounds). *Suppose RSRI is applied to an  $n \times n$  linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$  and set  $\mathbf{G} = \mathbf{I} - \mathbf{A}$ . If  $\mathbf{G}$  is a strict 1-norm contraction,  $\|\mathbf{G}\|_1 < 1$ , then RSRI returns an estimate  $\bar{\mathbf{x}}_t$  with variance bounded by*

$$\|\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{A}\mathbb{E}[\bar{\mathbf{x}}_t]\|^2 \leq \frac{8t}{(t - t_b)^2} \cdot \frac{1}{m} \left( \frac{\|\mathbf{b}\|_1}{1 - \|\mathbf{G}\|_1} \right)^2.$$

Alternately, if  $m \geq m_{\mathbf{G}} = \sum_{s=0}^{\infty} \|\mathbf{G}\|_1^s$ , the variance is bounded by

$$\|\mathbf{A}\bar{\mathbf{x}}_t - \mathbf{A}\mathbb{E}[\bar{\mathbf{x}}_t]\|^2 \leq \frac{8t \sup_{s \geq 0} \|\mathbf{G}\|_1^s}{(t - t_b)^2} \cdot \min_{0 \leq i \leq m - m_{\mathbf{G}}} \frac{1}{m - m_{\mathbf{G}} - i} \left( \sum_{j=i+1}^n \tilde{\mathbf{x}}^\downarrow(j) \right)^2.$$

Here,  $\tilde{\mathbf{x}} \in \mathbb{R}^n$  is the solution to the regularized linear system  $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ , where  $\tilde{\mathbf{A}} = \mathbf{I} - |\mathbf{G}|$ ,  $\tilde{\mathbf{b}} = |\mathbf{b}|$ , and  $\tilde{\mathbf{x}}^\downarrow \in \mathbb{R}^n$  is the decreasing rearrangement of  $\tilde{\mathbf{x}}$ .

*Proof.* Fix  $\mathbf{u} \in \mathbb{C}^n$  and introduce the martingale

$$m_s = \mathbb{E}[\mathbf{u}^* \mathbf{A} \bar{\mathbf{x}}_t \mid \mathbf{x}_0, \dots, \mathbf{x}_s].$$

A brief calculation shows that the martingale differences are given by

$$\begin{aligned} m_s - m_{s-1} &= \frac{1}{t - t_b} \mathbf{u}^* (\mathbf{I} - \mathbf{G}) \sum_{r=s \vee t_b}^{t-1} (\mathbb{E}[\mathbf{x}_r \mid \mathbf{x}_s] - \mathbb{E}[\mathbf{x}_r \mid \mathbf{x}_{s-1}]) \\ &= \frac{1}{t - t_b} \mathbf{u}^* (\mathbf{I} - \mathbf{G}) \sum_{r=s \vee t_b}^{t-1} \mathbf{G}^{r-s+1} [\phi_s(\mathbf{x}_{s-1}) - \mathbf{x}_{s-1}] \\ &= \frac{1}{t - t_b} \mathbf{u}^* (\mathbf{G}^{(t_b-s+1) \vee 1} - \mathbf{G}^{t-s+1}) [\phi_s(\mathbf{x}_{s-1}) - \mathbf{x}_{s-1}] \end{aligned}$$

for  $1 \leq s \leq t$ . Moreover,  $\text{Var}[\mathbf{u}^* \mathbf{A} \bar{\mathbf{x}}_t]$  is given by the martingale variance formula

$$\text{Var}[\mathbf{f}^* \mathbf{A} \bar{\mathbf{x}}_t] = \text{Var}[m_t] = \sum_{s=1}^t \mathbb{E}|m_s - m_{s-1}|^2.$$

We bound the variance using the following upper bounds:

$$\begin{aligned} \|\mathbf{A} \bar{\mathbf{x}}_t - \mathbf{A} \mathbb{E}[\bar{\mathbf{x}}_t]\|^2 &= \max_{\|\mathbf{u}\|_\infty \leq 1} \text{Var}[\mathbf{u}^* \mathbf{A} \bar{\mathbf{x}}_t] \\ &\leq \frac{1}{(t - t_b)^2} \sum_{s=0}^{t-1} \max_{\|\mathbf{u}\|_\infty \leq 1} \mathbb{E}|\mathbf{u}^* (\mathbf{G}^{(t_b-s) \vee 1} - \mathbf{G}^{t-s}) [\phi_{s+1}(\mathbf{x}_s) - \mathbf{x}_s]|^2 \\ &\leq \frac{1}{(t - t_b)^2} \sum_{s=0}^{t-1} \|\mathbf{G}^{(t_b-s) \vee 1} - \mathbf{G}^{t-s}\|_1^2 \cdot \max_{\|\mathbf{u}\|_\infty \leq 1} \mathbb{E}|\mathbf{u}^* [\phi_{s+1}(\mathbf{x}_s) - \mathbf{x}_s]|^2 \\ &= \frac{1}{(t - t_b)^2} \sum_{s=0}^{t-1} \|\mathbf{G}^{(t_b-s) \vee 1} - \mathbf{G}^{t-s}\|_1^2 \|\phi_{s+1}(\mathbf{x}_s) - \mathbf{x}_s\|^2. \end{aligned}$$

Last, we use the fact that

$$\sum_{s=0}^{t-1} \|\mathbf{G}^{(t_b-s) \vee 1} - \mathbf{G}^{t-s}\|_1^2 \leq 4t \sup_{s \geq 0} \|\mathbf{G}^s\|_1^2$$

to establish the general variance formula

$$\|\mathbf{A} \bar{\mathbf{x}}_t - \mathbf{A} \mathbb{E}[\bar{\mathbf{x}}_t]\|^2 \leq \frac{4t \sup_{s \geq 0} \|\mathbf{G}^s\|_1^2}{(t - t_b)^2} \cdot \sup_{s \geq 0} \|\phi_{s+1}(\mathbf{x}_s) - \mathbf{x}_s\|^2.$$

We apply the bounds on  $\sup_{s \geq 0} \|\phi_{s+1}(\mathbf{x}_s) - \mathbf{x}_s\|^2$  from [Propositions 6.3](#) and [6.5](#) to complete the proof.  $\square$

**Conclusion.** We have introduced a new algorithm called “randomly sparsified Richardson iteration” or “RSRI” ([Algorithm 1.2](#)) for solving  $n \times n$  linear systems of equations  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . RSRI can be applied to high-dimensional systems with  $n \geq 10^9$ . At each iteration, the algorithm only needs to evaluate a random subset of  $m$  columns, where  $m$  is a parameter specified by the user. Therefore, RSRI only requires  $\mathcal{O}(mN)$  work per iteration if  $\mathbf{A}$  and  $\mathbf{b}$  are dense, or  $\mathcal{O}(mq)$  work per iteration if  $\mathbf{A}$  and  $\mathbf{b}$  are sparse with no more than  $q$  nonzero entries per column. Because of this scaling, RSRI can efficiently generate sparse approximations to the solution vector for problems so large that the exact solution cannot be stored as a dense vector on a computer.

RSRI is an extension of the FRI framework [43, 33, 34, 32, 31] for speeding up deterministic fixed-point iterations with random sparsification. In this paper, we have extended FRI for the first time to handle linear systems  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , and we have proved that RSRI achieves faster-than- $1/\sqrt{m}$  convergence. Proving such a result has been a significant obstacle in the mathematical understanding of FRI, and our analysis will serve as the foundation for future algorithmic and mathematical developments. In particular, extending the results in this paper to FRI methods for eigenproblems [31], remains an outstanding challenge.

**Acknowledgments.** We would like to acknowledge Bixing Qiao, who helped us investigate classical Monte Carlo schemes for linear systems during his Master's thesis at New York University in 2020. We would also like to thank Lek-Heng Lim who helped to instigate this work, as well as Tyler Chen, Christopher Musco, Kevin Miller, and Ethan N. Epperly who provided valuable comments on an earlier draft. Last, we would like to acknowledge Jackie Lok, who alerted us to an error in an earlier draft.

**Appendix A. Derivation of SGD error bounds.** This section derives the error bounds for randomized coordinate descent and randomized Kaczmarz in subsection 4.2.

At each iteration, randomized Kaczmarz chooses an index set  $S \subseteq \{1, \dots, n\}$  by sampling with replacement from the nonuniform probability distribution  $p_j = \|\mathbf{A}(j, \cdot)\|^2 / \|\mathbf{A}\|_F^2$ . Then it updates the iterate according to

$$\begin{aligned} \mathbf{x}_s &= \mathbf{x}_{s-1} - \frac{\alpha \|\mathbf{A}\|_F^2}{|S|} \sum_{j \in S} \frac{\mathbf{A}(j, \cdot) \mathbf{x}_{s-1} - \mathbf{b}(j)}{\|\mathbf{A}(j, \cdot)^*\|^2} \mathbf{A}(j, \cdot)^* \\ &= \mathbf{x}_{s-1} - \frac{\alpha \|\mathbf{A}\|_F^2}{|S|} \sum_{j \in S} \frac{\mathbf{A}(j, \cdot)^* \mathbf{A}(j, \cdot)}{\|\mathbf{A}(j, \cdot)^*\|^2} (\mathbf{x}_{s-1} - \mathbf{x}_*), \end{aligned}$$

where the equality holds because  $\mathbf{A}(j, \cdot) \mathbf{x}_* = \mathbf{b}(j)$  for  $\mathbf{x}_* = \mathbf{A}^{-1} \mathbf{b}$ .

The conditional expectation of each iterate is given by

$$\begin{aligned} \mathbb{E}[\mathbf{x}_s - \mathbf{x}_* | \mathbf{x}_{s-1}] &= \left( \mathbf{I} - \frac{\alpha \|\mathbf{A}\|_F^2}{|S|} \mathbb{E} \left[ \sum_{j \in S} \frac{\mathbf{A}(j, \cdot)^* \mathbf{A}(j, \cdot)}{\|\mathbf{A}(j, \cdot)^*\|^2} \right] \right) (\mathbf{x}_{s-1} - \mathbf{x}_*) \\ &= (\mathbf{I} - \alpha \mathbf{A}^* \mathbf{A}) (\mathbf{x}_{s-1} - \mathbf{x}_*), \end{aligned}$$

because the expectation of  $\sum_{j \in S} \mathbf{A}(j, \cdot)^* \mathbf{A}(j, \cdot) / \|\mathbf{A}(j, \cdot)^*\|^2$  is  $|S| \mathbf{A}^* \mathbf{A} / \|\mathbf{A}\|_F^2$ .

Since the indices  $j_1, \dots, j_{|S|} \in S$  are independent and identically distributed, the trace of the conditional covariance matrix satisfies

$$\begin{aligned} \text{tr Cov}[\mathbf{x}_s - \mathbf{x}_* | \mathbf{x}_{s-1}] &= \frac{\alpha^2 \|\mathbf{A}\|_F^4}{|S|} \text{tr Cov} \left[ \frac{\mathbf{A}(j_1, \cdot)^* \mathbf{A}(j_1, \cdot)}{\|\mathbf{A}(j_1, \cdot)^*\|^2} (\mathbf{x}_{s-1} - \mathbf{x}_*) \middle| \mathbf{x}_{s-1} \right] \\ &= \frac{\alpha^2 \|\mathbf{A}\|_F^4}{|S|} (\mathbf{x}_{s-1} - \mathbf{x}_*)^* \left( \mathbb{E} \left[ \frac{\mathbf{A}(j_1, \cdot)^* \mathbf{A}(j_1, \cdot)}{\|\mathbf{A}(j_1, \cdot)^*\|^2} \right] - \mathbb{E} \left[ \frac{\mathbf{A}(j_1, \cdot)^* \mathbf{A}(j_1, \cdot)}{\|\mathbf{A}(j_1, \cdot)^*\|^2} \right]^2 \right) (\mathbf{x}_{s-1} - \mathbf{x}_*) \\ &= \frac{\alpha^2 \|\mathbf{A}\|_F^4}{|S|} (\mathbf{x}_{s-1} - \mathbf{x}_*)^* \left[ \frac{\mathbf{A}^* \mathbf{A}}{\|\mathbf{A}\|_F^2} - \left( \frac{\mathbf{A}^* \mathbf{A}}{\|\mathbf{A}\|_F^2} \right)^2 \right] (\mathbf{x}_{s-1} - \mathbf{x}_*). \end{aligned}$$

As the batch size  $|S|$  increases, the trace of the conditional variance decreases at the Monte Carlo rate  $\sim 1/|S|$ .

Last, we use the conditional bias and covariance formulas to derive the conditional mean square error, according to a vector-valued version of the bias-variance decomposition:

$$\begin{aligned}\mathbb{E}[\|\mathbf{x}_s - \mathbf{x}_\star\|^2 | \mathbf{x}_{s-1}] &= \|\mathbb{E}[\mathbf{x}_s - \mathbf{x}_\star | \mathbf{x}_{s-1}]\|^2 + \text{tr Cov}[\mathbf{x}_s - \mathbf{x}_\star | \mathbf{x}_{s-1}] \\ &= \|\mathbf{x}_{s-1} - \mathbf{x}_\star\|^2 + \left[ \frac{\alpha^2 \|\mathbf{A}\|_F^2}{|\mathcal{S}|} - 2\alpha \right] \|\mathbf{A}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2 \\ &\quad + \alpha^2 \frac{|\mathcal{S}| - 1}{|\mathcal{S}|} \|\mathbf{A}^* \mathbf{A}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2.\end{aligned}$$

This establishes the mean square error (4.5).

Next, to optimize the variance of randomized Kaczmarz, introduce an eigenvalue decomposition  $\mathbf{A}^* \mathbf{A} = \mathbf{V}^* \mathbf{\Lambda} \mathbf{V}$  where  $\mathbf{V} = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_n]$  and  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$  for  $\lambda_1 \geq \dots \geq \lambda_n$ . Observe that

$$\begin{aligned}\mathbf{x}_{s-1} - \mathbf{x}_\star &= \sum_{i=1}^n \mathbf{v}_i \mathbf{v}_i^* (\mathbf{x}_{s-1} - \mathbf{x}_\star), \\ \|\mathbf{x}_{s-1} - \mathbf{x}_\star\|^2 &= \sum_{i=1}^n |\mathbf{v}_i^* (\mathbf{x}_{s-1} - \mathbf{x}_\star)|^2, \\ \|\mathbf{A}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2 &= \sum_{i=1}^n \lambda_i |\mathbf{v}_i^* (\mathbf{x}_{s-1} - \mathbf{x}_\star)|^2, \\ \|\mathbf{A}^* \mathbf{A}(\mathbf{x}_{s-1} - \mathbf{x}_\star)\|^2 &= \sum_{i=1}^n \lambda_i^2 |\mathbf{v}_i^* (\mathbf{x}_{s-1} - \mathbf{x}_\star)|^2.\end{aligned}$$

We define eigenvector overlap terms  $\xi_i = \mathbb{E}|\mathbf{v}_i^* (\mathbf{x}_{s-1} - \mathbf{x}_\star)|^2$  for  $i = 1, \dots, n$ , so the expression (4.5) for the mean square error yields

$$\begin{aligned}(\text{A.1}) \quad \mathbb{E}\|\mathbf{x}_s - \mathbf{x}_\star\|^2 &= \sum_{i=1}^n \xi_i f(\alpha, \lambda_i), \\ \text{for } f(\alpha, \lambda) &= 1 + \left[ \frac{\alpha^2 \|\mathbf{A}\|_F^2}{|\mathcal{S}|} - 2\alpha \right] \lambda + \alpha^2 \frac{|\mathcal{S}| - 1}{|\mathcal{S}|} \lambda^2.\end{aligned}$$

We will proceed to optimize this error expression by considering the function  $f(\alpha, \lambda)$ .

For any  $\alpha > 0$ , the function  $\lambda \mapsto f(\alpha, \lambda)$  is a convex quadratic, and it achieves its maximum value for  $\lambda \in [\lambda_n, \lambda_1]$  at one of the endpoints,  $\lambda \in \{\lambda_1, \lambda_n\}$ . Further observe that  $f(\alpha, \lambda_1) \leq f(\alpha, \lambda_n)$  holds if

$$f(\alpha, \lambda_1) - f(\alpha, \lambda_n) = \left[ \frac{\alpha^2 \|\mathbf{A}\|_F^2}{|\mathcal{S}|} - 2\alpha \right] (\lambda_1 - \lambda_n) + \alpha^2 \frac{|\mathcal{S}| - 1}{|\mathcal{S}|} (\lambda_1^2 - \lambda_n^2) \leq 0,$$

which is equivalent to

$$(\text{A.2}) \quad \alpha \leq \frac{2|\mathcal{S}|}{\|\mathbf{A}\|_F^2 + (|\mathcal{S}| - 1)(\lambda_1 + \lambda_n)}.$$

Hence, when the step size  $\alpha$  is sufficiently small, the worst-case error is achieved for the smallest eigenvalue  $\lambda = \lambda_n$ .

We can optimize the worst-case error with  $\lambda = \lambda_n$  by using the step size

$$\alpha = \underset{\beta > 0}{\text{argmin}} f(\lambda_n, \beta) = \frac{|\mathcal{S}|}{\|\mathbf{A}\|_F^2 + (|\mathcal{S}| - 1)\lambda_n}.$$

Observe that this step size satisfies the smallness criterion (A.2) if

$$(|\mathcal{S}| - 1)(\lambda_1 - \lambda_n) \leq \|\mathbf{A}\|_F^2.$$

Under the smallness criterion, we have shown

$$f(\alpha, \lambda) \leq f(\alpha, \lambda_n) = 1 - \frac{|S|\lambda_n}{\|\mathbf{A}\|_F^2 + (|S| - 1)\lambda_n} \quad \text{for each } \lambda \in [\lambda_n, \lambda_1].$$

The expression (A.1) for the randomized Kaczmarz error then yields

$$\mathbb{E}\|\mathbf{x}_s - \mathbf{x}_\star\|^2 \leq \left[1 - \frac{|S|\lambda_n}{\|\mathbf{A}\|_F^2 + (|S| - 1)\lambda_n}\right] \mathbb{E}\|\mathbf{x}_{s-1} - \mathbf{x}_\star\|^2$$

This completes the analysis of randomized Kaczmarz.

Last, we transfer the error bounds from randomized Kaczmarz to randomized coordinate descent. Using randomized coordinate descent to solve  $\mathbf{A}\mathbf{x} = \mathbf{b}$  produces the same distribution of iterates as using randomized Kaczmarz to solve  $\mathbf{A}^{1/2}\mathbf{y} = \mathbf{b}$  and then setting  $\mathbf{x} = \mathbf{A}^{-1/2}\mathbf{y}$  [16, Sec. 5.1]. We have already derived error bounds for iterates  $\mathbf{y}_s$  when randomized Kaczmarz is applied to  $\mathbf{A}^{1/2}\mathbf{y} = \mathbf{b}$ . We obtain corresponding error bounds for randomized coordinate descent by substituting  $\mathbf{y}_s = \mathbf{A}^{1/2}\mathbf{x}_s$  in the right places.

#### REFERENCES

- [1] E. AGIRRE AND A. SOROA, *Personalizing PageRank for word sense disambiguation*, in Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, 2009, p. 33–41, <https://dl.acm.org/doi/10.5555/1609067.1609070>.
- [2] R. ALBERT, H. JEONG, AND A.-L. BARABÁSI, *Diameter of the World-Wide Web*, *Nature*, 401 (1999), pp. 130–131, <https://doi.org/10.1038/43601>.
- [3] S. ANAND, *Kaggle datasets: Amazon product reviews*, 2019, <https://www.kaggle.com/dataset/s/saurav9786/amazon-product-reviews>. Version 1.
- [4] R. ANDERSEN, C. BORGS, J. CHAYES, J. HOPCRAFT, V. S. MIRROKNI, AND S.-H. TENG, *Local computation of PageRank contributions*, in 5th International Workshop on Algorithms and Models for the Web-Graph, 2007, [https://doi.org/10.1007/978-3-540-77004-6\\_12](https://doi.org/10.1007/978-3-540-77004-6_12).
- [5] A. ANDONI, R. KRAUTHGAMER, AND Y. POGROW, *On solving linear systems in sublinear time*, 2018, <https://arxiv.org/abs/1809.02995>.
- [6] K. AVRACHENKOV, N. LITVAK, D. NEMIROVSKY, AND N. OSIPOVA, *Monte Carlo methods in PageRank computation: When one iteration is sufficient*, *SIAM Journal on Numerical Analysis*, 45 (2007), pp. 890–904, <https://doi.org/10.1137/050643799>.
- [7] W. F. BAUER, *The Monte Carlo method*, *Journal of the Society for Industrial and Applied Mathematics*, 6 (1958), pp. 438–451, <https://doi.org/10.1137/0106028>.
- [8] M. BENZI, T. M. EVANS, S. P. HAMILTON, M. LUPO PASINI, AND S. R. SLATTERY, *Analysis of Monte Carlo accelerated iterative methods for sparse linear systems*, *Numerical Linear Algebra with Applications*, 24 (2017), <https://doi.org/10.1002/nla.2088>.
- [9] P. BERKHIN, *Bookmark-coloring algorithm for personalized PageRank computing*, *Internet Mathematics*, 3 (2006), pp. 41–62, <https://doi.org/10.1080/15427951.2006.10129116>.
- [10] G. H. BOOTH, A. J. W. THOM, AND A. ALAVI, *Fermion Monte Carlo without fixed nodes: A game of life, death, and annihilation in slater determinant space*, *The Journal of Chemical Physics*, 131 (2009), p. 054106, <https://doi.org/10.1063/1.3193710>.
- [11] C. BORGS, M. BRAUTBAR, J. CHAYES, AND S.-H. TENG, *Multiscale matrix sampling and sublinear-time PageRank computation*, *Internet Mathematics*, 10 (2014), pp. 20–48, <https://doi.org/10.1080/15427951.2013.802752>.
- [12] X. CHEN AND E. PRICE, *Active regression via linear-sample sparsification*, in Proceedings of the Thirty-Second Conference on Learning Theory, 2019, <https://proceedings.mlr.press/v99/chen19a.html>.
- [13] D. CLELAND, G. H. BOOTH, AND A. ALAVI, *Communications: Survival of the fittest: Accelerating convergence in full configuration-interaction quantum Monte Carlo*, *The Journal of Chemical Physics*, 132 (2010), p. 041103, <https://doi.org/10.1063/1.3302277>.
- [14] A. COHEN, W. DAHMEN, AND R. DEVORE, *Adaptive wavelet methods for elliptic operator equations: Convergence rates*, *Mathematics of Computation*, 70 (2001), pp. 27–75, <http://www.jstor.org/stable/2698924>.

- [15] J. H. CURTISS, “*Monte Carlo*” methods for the iteration of linear operators, *Journal of Mathematics and Physics*, 32 (1953), pp. 209–232, <https://doi.org/10.1002/sapm1953321209>.
- [16] M. DEREZIŃSKI, D. NEEDELL, E. REBROVA, AND J. YANG, *Randomized kacmarz methods with beyond-krylov convergence*, 2025, <https://arxiv.org/abs/2501.11673>.
- [17] J.-C. DEVILLE AND Y. TILLE, *Unequal probability sampling without replacement through a splitting method*, *Biometrika*, 85 (1998), pp. 89–101, <http://www.jstor.org/stable/2337311>.
- [18] I. DIMOV, *Minimization of the probable error for some Monte Carlo methods*, in *Proceedings of the International Conference on Mathematical Modeling and Scientific Computation*, 1991, pp. 159–170.
- [19] H. P. EDMUNDSON, *Monte Carlo matrix inversion and recurrent events*, *Mathematical Tables and Other Aids to Computation*, 7 (1953), pp. 18–21, <https://doi.org/10.2307/2002564>.
- [20] E. N. EPPERLY, G. GOLDSHLAGER, AND R. J. WEBBER, *Randomized kacmarz with tail averaging*, 2025, <https://arxiv.org/abs/2411.19877>.
- [21] T. M. EVANS, S. W. MOSHER, S. R. SLATTERY, AND S. P. HAMILTON, *A Monte Carlo synthetic-acceleration method for solving the thermal radiation diffusion equation*, *Journal of Computational Physics*, 258 (2014), pp. 338–358, <https://doi.org/10.1016/j.jcp.2013.10.043>.
- [22] R. W. FLOYD, *Algorithm 245: Treesort*, *Communications of the ACM*, 7 (1964), p. 701, <https://doi.org/10.1145/355588.365103>.
- [23] D. FOGARAS, B. RÁCZ, K. CSALOGÁNY, AND T. SARLÓS, *Towards scaling fully personalized PageRank: Algorithms, lower bounds, and experiments*, *Internet Mathematics*, 2 (2005), pp. 333–358, <https://doi.org/10.1080/15427951.2005.10129104>.
- [24] G. E. FORSYTHE AND R. A. LEIBLER, *Matrix inversion by a Monte Carlo method*, *Mathematical Tables and Other Aids to Computation*, 4 (1950), pp. 127–129, <http://www.jstor.org/stable/2002508>.
- [25] G. GARRIGOS AND R. M. GOWER, *Handbook of convergence theorems for (stochastic) gradient methods*, 2024, <https://arxiv.org/abs/2301.11235>.
- [26] A. GILYÉN, Z. SONG, AND E. TANG, *An improved quantum-inspired algorithm for linear regression*, *Quantum*, 6 (2022), p. 754, <https://doi.org/10.22331/q-2022-06-30-754>.
- [27] D. GLEICH AND M. POLITO, *Approximating personalized PageRank with minimal use of web graph data*, *Internet Mathematics*, 3 (2006), <https://doi.org/10.1080/15427951.2006.10129128>.
- [28] D. F. GLEICH, *PageRank beyond the web*, *SIAM Review*, 57 (2015), pp. 321–363, <https://doi.org/10.1137/140976649>.
- [29] M. GORI AND A. PUCCI, *ItemRank: A random-walk based scoring algorithm for recommender engines*, in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, p. 2766–2771, <https://dl.acm.org/doi/10.5555/1625275.1625720>.
- [30] R. M. GOWER, N. LOIZOU, X. QIAN, A. SAILANBAYEV, E. SHULGIN, AND P. RICHTÁRIK, *SGD: General analysis and improved rates*, in *Proceedings of the 36th International Conference on Machine Learning*, 2019, <https://proceedings.mlr.press/v97/qian19b.html>.
- [31] S. M. GREENE, R. J. WEBBER, T. C. BERKELBACH, AND J. WEARE, *Approximating matrix eigenvalues by subspace iteration with repeated random sparsification*, *SIAM Journal on Scientific Computing*, 44 (2022), pp. A3067–A3097, <https://doi.org/10.1137/21M1422513>.
- [32] S. M. GREENE, R. J. WEBBER, J. E. T. SMITH, J. WEARE, AND T. C. BERKELBACH, *Full configuration interaction excited-state energies in large active spaces from subspace iteration with repeated random sparsification*, *Journal of Chemical Theory and Computation*, 18 (2022), pp. 7218–7232, <https://doi.org/10.1021/acs.jctc.2c00435>.
- [33] S. M. GREENE, R. J. WEBBER, J. WEARE, AND T. C. BERKELBACH, *Beyond walkers in stochastic quantum chemistry: Reducing error using fast randomized iteration*, *Journal of Chemical Theory and Computation*, 15 (2019), pp. 4834–4850, <https://doi.org/10.1021/acs.jctc.9b00422>.
- [34] S. M. GREENE, R. J. WEBBER, J. WEARE, AND T. C. BERKELBACH, *Improved fast randomized iteration approach to full configuration interaction*, *Journal of Chemical Theory and Computation*, 16 (2020), pp. 5572–5585, <https://doi.org/10.1021/acs.jctc.0c00437>.
- [35] J. H. HALTON, *Sequential Monte Carlo techniques for the solution of linear systems*, *Journal of Scientific Computing*, 9 (1994), pp. 213–257, <https://doi.org/10.1007/bf01578388>.
- [36] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, second ed., 2012, <https://doi.org/10.1017/CBO9781139020411>.
- [37] W. G. HORNER AND D. GILBERT, *XXI. A new method of solving numerical equations of all orders, by continuous approximation*, *Philosophical Transactions of the Royal Society of London*, 109 (1819), pp. 308–335, <https://doi.org/10.1098/rstl.1819.0023>.
- [38] G. JEH AND J. WIDOM, *Scaling personalized web search*, in *Proceedings of the 12th International Conference on World Wide Web*, 2003, <https://doi.org/10.1145/775152.775191>.



- [39] H. JI, M. MASCAGNI, AND Y. LI, *Convergence analysis of Markov chain Monte Carlo linear solvers using Ulam–von Neumann algorithm*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 2107–2122, <https://doi.org/10.1137/130904867>.
- [40] Y. LAI, *Adaptive Monte Carlo methods for matrix equations with applications*, Journal of Computational and Applied Mathematics, 231 (2009), pp. 705–714, <https://doi.org/10.1016/j.cam.2009.04.008>.
- [41] J. LESKOVEC AND A. KREVL, *SNAP datasets: Notre Dame web graph*, 1999, <http://snap.stanford.edu/data/web-NotreDame.html>.
- [42] D. LEVENTHAL AND A. S. LEWIS, *Randomized methods for linear constraints: Convergence rates and conditioning*, Mathematics of Operations Research, 35 (2010), pp. 641–654, <https://doi.org/10.1287/moor.1100.0456>.
- [43] L.-H. LIM AND J. WEARE, *Fast randomized iteration: Diffusion Monte Carlo through the lens of numerical linear algebra*, SIAM Review, 59 (2017), pp. 547–587, <https://doi.org/10.1137/15M1040827>.
- [44] J. LU AND Z. WANG, *The full configuration interaction quantum Monte Carlo method through the lens of inexact power iteration*, SIAM Journal on Scientific Computing, 42 (2020), pp. B1–B29, <https://doi.org/10.1137/18M1166626>.
- [45] J. D. MOORMAN, T. K. TU, D. MOLITOR, AND D. NEEDELL, *Randomized Kaczmarz with averaging*, BIT Numerical Mathematics, 61 (2020), p. 337–359, <https://doi.org/10.1007/s10543-020-00824-1>.
- [46] D. NEEDELL, N. SREBRO, AND R. WARD, *Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm*, Mathematical Programming, 155 (2015), pp. 549–573, <https://doi.org/10.1007/s10107-015-0864-7>.
- [47] J. NI, J. LI, AND J. MCAULEY, *Justifying recommendations using distantly-labeled reviews and fine-grained aspects*, in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, <https://doi.org/10.18653/v1/D19-1018>.
- [48] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, second ed., 2006, <https://doi.org/10.1007/978-0-387-40065-5>.
- [49] G. ÖKTEN, *Solving linear equations by Monte Carlo simulation*, SIAM Journal on Scientific Computing, 27 (2005), pp. 511–531, <https://doi.org/10.1137/04060500X>.
- [50] T. OPSAHL, *Why Anchorage is not (that) important: Binary ties and sample selection*, 2011, <https://toreopsahl.com/2011/08/12/why-anchorage-is-not-that-important-binary-ties-and-sample-selection/>.
- [51] A. OZDAGLAR, D. SHAH, AND C. L. YU, *Asynchronous approximation of a single component of the solution to a linear system*, IEEE Transactions on Network Science and Engineering, 7 (2020), pp. 975–986, <https://doi.org/10.1109/TNSE.2019.2894990>.
- [52] L. PAGE, S. BRIN, R. MOTWANI, AND T. WINOGRAD, *The PageRank citation ranking: Bringing order to the web.*, Tech. Report 1999-66, Stanford InfoLab, 1999, <http://ilpubs.stanford.edu:8090/422/>.
- [53] Z. QU AND P. RICHTÁRIK, *Coordinate descent with arbitrary sampling i: algorithms and complexity*, Optimization Methods and Software, 31 (2016), pp. 829–857, <https://doi.org/10.1080/10556788.2016.1190360>.
- [54] P. RATHORE, Z. FRANGELLA, J. YANG, M. DEREZIŃSKI, AND M. UDELL, *Have askotch: A neat solution for large-scale kernel ridge regression*, 2025, <https://arxiv.org/abs/2407.10070>.
- [55] L. F. RICHARDSON AND R. T. GLAZEBROOK, *Ix. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam*, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 210 (1911), pp. 307–357, <https://doi.org/10.1098/rsta.1911.0009>.
- [56] F. ROSENBLATT, *The perceptron: A probabilistic model for information storage and organization in the brain.*, Psychological Review, 65 (1958), p. 386–408, <https://doi.org/10.1037/h0042519>.
- [57] E. SADEH AND M. FRANKLIN, *Monte Carlo solution of partial differential equations by special purpose digital computer*, IEEE Transactions on Computers, C-23 (1974), pp. 389–397, <https://doi.org/10.1109/T-C.1974.223954>.
- [58] T. SARLÓS, A. A. BENCZÜR, K. CSALOGÁNY, D. FOGARAS, AND B. RÁCZ, *To randomize or not to randomize: Space optimal summaries for hyperlink analysis*, in Proceedings of the 15th International Conference on World Wide Web, 2006, <https://doi.org/10.1145/1135777.1135823>.
- [59] C. SHAO AND A. MONTANARO, *Faster quantum-inspired algorithms for solving linear systems*, ACM Transactions on Quantum Computing, 3 (2022), <https://doi.org/10.1145/3520141>.

- [60] J. J. SHEPHERD, G. BOOTH, A. GRÜNEIS, AND A. ALAVI, *Full configuration interaction perspective on the homogeneous electron gas*, Physical Review B, 85 (2012), p. 081103, <https://doi.org/10.1103/PhysRevB.85.081103>.
- [61] N. SHYAMKUMAR, S. BANERJEE, AND P. LOFGREN, *Sublinear estimation of a single element in sparse linear systems*, in 54th Annual Allerton Conference on Communication, Control, and Computing, 2016, pp. 856–860, <https://doi.org/10.1109/ALLERTON.2016.7852323>.
- [62] A. SRINIVASAN, *Distributions on level-sets with applications to approximation algorithms*, in Proceedings 42nd IEEE Symposium on Foundations of Computer Science, 2001, pp. 588–597.
- [63] T. STROHMER AND R. VERSHYNIN, *A randomized Kaczmarz algorithm with exponential convergence*, Journal of Fourier Analysis and Applications, 15 (2008), pp. 262–278, <https://doi.org/10.1007/s00041-008-9030-4>.
- [64] Q. WANG, D. GLEICH, A. SABERI, N. ETEMADI, AND P. MOIN, *A Monte Carlo method for solving unsteady adjoint equations*, Journal of Computational Physics, 227 (2008), pp. 6184–6205, <https://doi.org/10.1016/j.jcp.2008.03.006>.
- [65] W. R. WASOW, *A note on the inversion of matrices by random walks*, Mathematical Tables and Other Aids to Computation, 6 (1952), pp. 78–81, <http://www.jstor.org/stable/2002546>.
- [66] A. WISSNER-GROSS, *Preparation of topical reading lists from the link structure of Wikipedia*, in Sixth IEEE International Conference on Advanced Learning Technologies, 2006, pp. 825–829, <https://doi.org/10.1109/ICALT.2006.1652568>.