PAVED WITH GOOD INTENTIONS: ANALYSIS OF A RANDOMIZED BLOCK KACZMARZ METHOD

DEANNA NEEDELL AND JOEL A. TROPP

ABSTRACT. The block Kaczmarz method is an iterative scheme for solving overdetermined least-squares problems. At each step, the algorithm projects the current iterate onto the solution space of a subset of the constraints. This paper describes a block Kaczmarz algorithm that uses a randomized control scheme to choose the subset at each step. This algorithm is the first block Kaczmarz method with an (expected) linear rate of convergence that can be expressed in terms of the geometric properties of the matrix and its submatrices. The analysis reveals that the algorithm is most effective when it is given a good *row paving* of the matrix, a partition of the rows into well-conditioned blocks. The operator theory literature provides detailed information about the existence and construction of good row pavings. Together, these results yield an efficient block Kaczmarz scheme that applies to many overdetermined least-squares problem.

1. INTRODUCTION

The Kaczmarz method [Kac37] is an iterative algorithm for solving overdetermined least-squares problems. Because of its simplicity and performance, this scheme has found application in fields ranging from image reconstruction to digital signal processing [SS87, CFM+92, FS95, Nat01]. At each iteration, the basic Kaczmarz method makes progress by enforcing a single constraint, while the block Kaczmarz method [Elf80] enforces many constraints at once. This paper introduces a randomized version of the block Kaczmarz method that converges with an expected linear rate, and we characterize the performance of this algorithm using geometric properties of the blocks of equations. This analysis leads us to consider the concept of a *row paving* of a matrix, a partition of the rows into well-conditioned blocks. We summarize the literature on row pavings, and we explain how this theory interacts with the block Kaczmarz method. Together, these results yield an efficient block Kaczmarz scheme that applies to many overdetermined least-squares problems.

1.1. Standing Assumptions and Notation.

Let $\boldsymbol{A}$ be a real or complex $n \times d$ matrix with full column rank, and suppose that $\boldsymbol{b}$ is a vector with dimension n. Consider the overdetermined least-squares problem

$$\text{minimize} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2. \tag{1.1}$$

The symbol $\|\cdot\|_p$ refers to the ℓ_p vector norm for $p \in [1, \infty]$. We write $\boldsymbol{x}_\star$ for the unique minimizer of (1.1), and we introduce the residual vector $\boldsymbol{e} := \boldsymbol{A}\boldsymbol{x}_\star - \boldsymbol{b}$.

To streamline our discussion, we will often assume that each *row* $\boldsymbol{a}_i$ of the matrix $\boldsymbol{A}$ shares the same ℓ_2 norm:

$$\|\boldsymbol{a}_i\|_2 = 1 \quad \text{for each } i = 1, \dots, n. \tag{1.2}$$

We say that $\boldsymbol{A}$ is *standardized* when (1.2) is in force.

The spectral norm is denoted by $\|\cdot\|$, while $\|\cdot\|_{\mathrm{F}}$ represents the Frobenius norm. When applied to an Hermitian matrix, the maps $\lambda_{\min}$ and $\lambda_{\max}$ return the algebraic minimum and maximum eigenvalues. For a $p \times q$ matrix $\boldsymbol{W}$, we arrange the singular values as follows.

$$\sigma_{\max}(\boldsymbol{W}) := \sigma_1(\boldsymbol{W}) \geq \sigma_2(\boldsymbol{W}) \geq \cdots \geq \sigma_{\min\{p,q\}}(\boldsymbol{W}) =: \sigma_{\min}(\boldsymbol{W}).$$

Date: 17 August 2012.

The minimum singular value is positive if and only if $\boldsymbol{W}\boldsymbol{W}^*$ or $\boldsymbol{W}^*\boldsymbol{W}$ is nonsingular. We define the *condition number* $\kappa(\boldsymbol{W}) := \sigma_{\max}(\boldsymbol{W})/\sigma_{\min}(\boldsymbol{W})$. The dagger $\dagger$ denotes the Moore–Penrose pseudoinverse. When $\boldsymbol{W}$ has full row rank, its pseudoinverse is determined by the formula $\boldsymbol{W}^\dagger := \boldsymbol{W}^*(\boldsymbol{W}\boldsymbol{W}^*)^{-1}$.

1.2. **The Simple Kaczmarz Method.** The Kaczmarz method is an iterative algorithm that produces an approximation to the minimizer $\boldsymbol{x}_\star$ of the least-squares problem (1.1). The method commences with an arbitrary guess $\boldsymbol{x}_0$ for the solution. At the jth iteration, we select a row index $t = t(j)$ of the matrix $\boldsymbol{A}$, and we project the current iterate $\boldsymbol{x}_{j-1}$ onto the solution space of the equation $\langle \boldsymbol{a}_t, \boldsymbol{x}\rangle = b_t$. That is,

$$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} + \frac{b_t - \langle \boldsymbol{a}_t, \boldsymbol{x}_{j-1}\rangle}{\|\boldsymbol{a}_t\|_2^2}\,\boldsymbol{a}_t. \tag{1.3}$$

This process continues until it triggers an appropriate convergence criterion.

To develop a complete algorithm, we also need a control mechanism that specifies how to select rows. For example, the most classical approach cycles through the rows in order. Instead, we focus on a modern formulation that uses a *randomized* control mechanism. Randomization has several benefits: the resulting algorithm is easy to analyze, it is simple to implement, and it is often effective in practice.

Our primary reference is the randomized Kaczmarz algorithm recently proposed by Strohmer and Vershynin [SV09b]. When $\boldsymbol{A}$ is standardized, their method operates as follows. At iteration j, independently of all previous random choices, the algorithm draws the row index $t(j)$ uniformly at random from the set $\{1, \dots, n\}$ of all row indices. Then the current iterate is updated using the rule (1.3). The paper [SV09b] provides a short, elegant proof that this iteration converges at an expected linear rate to the solution $\boldsymbol{x}_\star$ of a consistent least-squares problem (i.e., where the residual $\boldsymbol{e}$ is zero).

Needell [Nee10] has extended the argument of [SV09b] to the case of an *inconsistent* least-squares problem. For a standardized matrix $\boldsymbol{A}$, Needell's error estimate reads

$$\mathbb{E}\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 \le \left[1 - \frac{\sigma_{\min}^2(\boldsymbol{A})}{n}\right]^j \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 + \frac{n\|\boldsymbol{e}\|_\infty^2}{\sigma_{\min}^2(\boldsymbol{A})}. \tag{1.4}$$

In words, the randomized Kaczmarz method converges in expectation at a linear rate[1] until it reaches a fixed ball about the true solution $\boldsymbol{x}_\star$, at which point the error may cease to decay. The radius of this ball roughly equals the second term in (1.4), while the convergence rate is controlled by the bracket. When the residual $\boldsymbol{e}$ is zero, the bound (1.4) reduces to the error estimate from [SV09b].

When $\boldsymbol{A}$ is an $n \times d$ standardized matrix whose columns are well conditioned, the minimum singular value $\sigma_{\min}^2(\boldsymbol{A}) \ge \text{const} \cdot n/d$. In this case, the error bound (1.4) simplifies to

$$\mathbb{E}\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 \lesssim \left[1 - \frac{\text{const}}{d}\right]^j \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 + \frac{d}{\text{const}}\|\boldsymbol{e}\|_\infty^2.$$

It follows that $\mathrm{O}(d)$ iterations of the Kaczmarz method suffice to reduce the error by a constant fraction, provided that the squared error is substantially larger than $d\|\boldsymbol{e}\|_\infty^2$.

1.3. **The Block Kaczmarz Method.** In some situations [EHL81], practitioners prefer to use a block version of the Kaczmarz method to solve the least-squares problem (1.1). We consider a formulation due to Elfving [Elf80]. This procedure begins with an initial guess $\boldsymbol{x}_0$ for the solution. At each iteration j, we select a subset $\tau = \tau(j)$ of the row indices of $\boldsymbol{A}$, and we project the current iterate $\boldsymbol{x}_{j-1}$ onto the solution space of $\boldsymbol{A}_\tau \boldsymbol{x} = \boldsymbol{b}_\tau$, the set of equations listed in τ. That is,

$$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} + (\boldsymbol{A}_\tau)^\dagger(\boldsymbol{b}_\tau - \boldsymbol{A}_\tau \boldsymbol{x}_{j-1}). \tag{1.5}$$

[1] Mathematicians often use the term *exponential convergence* for the concept numerical analysts call *linear convergence*.

Algorithm 1.1 Block Kaczmarz Method with Uniform Random Control

Input:

- Matrix $\boldsymbol{A}$ with dimension $n \times d$
- Right-hand side $\boldsymbol{b}$ with dimension n
- Partition $T = \{\tau_1, \dots, \tau_m\}$ of the row indices $\{1, \dots, n\}$
- Initial iterate $\boldsymbol{x}_0$ with dimension d
- Convergence tolerance $\varepsilon > 0$

Output: An estimate $\hat{\boldsymbol{x}}$ for the solution to $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$

$j \leftarrow 0$
repeat
  $j \leftarrow j + 1$
  Choose a block τ uniformly at random from T
  $\boldsymbol{x}_j \leftarrow \boldsymbol{x}_{j-1} + (\boldsymbol{A}_\tau)^\dagger(\boldsymbol{b}_\tau - \boldsymbol{A}_\tau \boldsymbol{x}_{j-1})$ { Solve least-squares problem }
until $\|\boldsymbol{A}\boldsymbol{x}_j - \boldsymbol{b}\|_2^2 \le \varepsilon^2$
$\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}_j$

This process continues until it has converged. We have written $\boldsymbol{A}_\tau$ for the row submatrix of $\boldsymbol{A}$ indexed by τ, while $\boldsymbol{b}_\tau$ is the subvector of $\boldsymbol{b}$ with components listed in τ. We assume that the row submatrix $\boldsymbol{A}_\tau$ is fat[2], so the pseudoinverse (1.5) returns the solution to an underdetermined least-squares problem.

To specify a block Kaczmarz algorithm, one must decide what blocks of indices are permissible, as well as the mechanism for selecting a block at each iteration. In this paper, we study a version that is based on two design decisions. First, this algorithm requires a partition $T = \{\tau_1, \dots, \tau_m\}$ of the row indices of $\boldsymbol{A}$. The method only considers blocks of indices that appear in the partition T. Second, we use a simple randomized control scheme to choose which block to enforce. At each iteration, independently of all previous choices, we draw a block τ uniformly at random from the partition T. These decisions lead to Algorithm 1.1. We postpone a detailed discussion on implementation to Section 4.

We make no claim that randomized selection provides the optimal sequence for choosing blocks. As with the simple Kaczmarz method, the scaling of the rows of the matrix can play a significant role in the behavior of the algorithm. See [CHJ09, SV09a] for a discussion of this issue.

1.4. **Desiderata for the Partition.** The implementation and behavior of the block Kaczmarz method depend heavily on the properties of the submatrices $\boldsymbol{A}_\tau$ indexed by the blocks τ in the partition T. Let us explain how the structure of the submatrices plays a role in the implementation; the claims about performance will emerge from the theoretical results in Section 1.5.

The most expensive (arithmetic) step in Algorithm 1.1 occurs when we apply the pseudoinverse $\boldsymbol{A}_\tau^\dagger$ to a vector. We can perform this calculation efficiently *provided that* each submatrix $\boldsymbol{A}_\tau$ has well-conditioned rows. Indeed, in this case, we can invoke an iterative least-squares solver [Bjö96], such as `CGLS`, to apply the pseudoinverse $\boldsymbol{A}_\tau^\dagger$ approximately using a small number of matrix–vector multiplies with $\boldsymbol{A}_\tau$ and $\boldsymbol{A}_\tau^*$. In particular, we never need to form the pseudoinverse.

This observation highlights how important it is to control the geometric properties of the submatrices $\boldsymbol{A}_\tau$ induced by T. Let us make a definition that encapsulates the information that we will need.

Definition 1.1 (Row Paving)**.** An (m, α, β) *row paving* of a matrix $\boldsymbol{A}$ is a partition $T = \{\tau_1, \dots, \tau_m\}$ of the row indices that verifies

$$\alpha \le \lambda_{\min}(\boldsymbol{A}_\tau \boldsymbol{A}_\tau^*) \quad \text{and} \quad \lambda_{\max}(\boldsymbol{A}_\tau \boldsymbol{A}_\tau^*) \le \beta \quad \text{for each } \tau \in T.$$

[2] A $p \times q$ matrix is *fat* when $p \le q$.

The number m of blocks is called the *size* of the paving. The numbers α and β are called *lower* and *upper paving bounds*. The ratio β/α gives a uniform bound on the squared condition number $\kappa^2(\boldsymbol{A}_\tau)$ for each τ. Note that $\alpha = 0$ unless each submatrix $\boldsymbol{A}_\tau$ is fat.

Every partition T of the rows of a matrix $\boldsymbol{A}$ has associated paving parameters (m, α, β). In a moment, we will see how these quantities play a role in the performance of the algorithm. Roughly speaking, it is best that the size m, the upper bound β, and the conditioning β/α of the paving are small. Later, in Sections 1.7 and 3, we will discuss what kind of bounds we can expect on the paving parameters, as well as computational methods for producing good pavings. Note that, for a row paving to be useful in our context, the cost of producing the paving must not exceed the cost of solving the least-squares problem by other means!

1.5. **Convergence of Randomized Block Kaczmarz.** The main result of this paper provides information about the convergence properties of the randomized block Kaczmarz method, Algorithm 1.1, in terms of the parameters of the row paving T.

Theorem 1.2 (Convergence)**.** *Suppose $\boldsymbol{A}$ is a matrix with full column rank that admits an (m, α, β) row paving T. Consider the least-squares problem*

$$\text{minimize} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

Let $\boldsymbol{x}_\star$ be the unique minimizer, and define the residual $\boldsymbol{e} := \boldsymbol{A}\boldsymbol{x}_\star - \boldsymbol{b}$. For any initial estimate $\boldsymbol{x}_0$, the randomized block Kaczmarz method, Algorithm 1.1, produces a sequence $\{\boldsymbol{x}_j : j \geq 0\}$ of iterates that satisfies

$$\mathbb{E}\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 \leq \left[1 - \frac{\sigma_{\min}^2(\boldsymbol{A})}{\beta m}\right]^j \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 + \frac{\beta}{\alpha} \cdot \frac{\|\boldsymbol{e}\|_2^2}{\sigma_{\min}^2(\boldsymbol{A})}. \tag{1.6}$$

Turn to Section 2 for the proof of Theorem 1.2.

The expression (1.6) states that the block Kaczmarz method exhibits an expected linear rate of convergence until it reaches a ball about the true solution. The radius of this ball, which we call the *convergence horizon*, is comparable with the second term on the right-hand side of (1.6). The bracket controls the convergence rate. The minimum singular value of $\boldsymbol{A}$ affects both the rate of convergence and the convergence horizon. In each case, we prefer $\sigma_{\min}(\boldsymbol{A})$ to be as large as possible.

The properties of the row paving play an interesting role in Theorem 1.2. Curiously, the rate of convergence depends only on the upper paving bound β and the number m of blocks in the paving. On the other hand, the convergence horizon reflects the conditioning β/α of the paving. Thus, the conditioning of the paving only affects the error bound when the least-squares problem is inconsistent (i.e., $\boldsymbol{e}$ is nonzero). Nevertheless, as Section 1.4 suggests, we usually want the paving to be well conditioned to ensure that we can apply the block update rule (1.5) efficiently.

1.6. **Simple Kaczmarz versus Block Kaczmarz.** First, notice that Theorem 1.2 improves on the earlier result (1.4) for the simple Kaczmarz method. Indeed, the simple Kaczmarz method is equivalent to using a row paving with n blocks, where each block contains exactly one of the n rows. When $\boldsymbol{A}$ is standardized, the paving constants satisfy $\alpha = \beta = 1$, and we reach the error bound

$$\mathbb{E}\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 \leq \left[1 - \frac{\sigma_{\min}^2(\boldsymbol{A})}{n}\right]^j \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 + \frac{\|\boldsymbol{e}\|_2^2}{\sigma_{\min}^2(\boldsymbol{A})}.$$

The convergence horizon $\|\boldsymbol{e}\|^2 \leq n\|\boldsymbol{e}\|_\infty^2$, so the displayed bound beats (1.4) when $\boldsymbol{A}$ is standardized.

More generally, suppose $\boldsymbol{A}$ is a standardized matrix with an (m, α, β) row paving T. Let us compare the simple Kaczmarz algorithm with uniformly random control (Section 1.2) to the block Kaczmarz method, Algorithm 1.1. Both methods satisfy an error bound of the form

$$\mathbb{E}\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 \leq \mathrm{e}^{-j\rho} \cdot \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 + h$$

where the convergence rate ρ and the convergence horizon h depend on the choice of algorithm.

First, we compare the convergence rates of the two methods. The bounds (1.4) and (1.6) imply

$$\rho_{\text{simp}} \geq \frac{\sigma_{\min}^2(\boldsymbol{A})}{n} \quad \text{and} \quad \rho_{\text{block}} \geq \frac{\sigma_{\min}^2(\boldsymbol{A})}{\beta m} \tag{1.7}$$

because $\log(1-t) \leq -t$ when $t < 1$. In other words, the simple method requires a factor $n/(\beta m)$ more iterations than the block method to achieve the same reduction in error.

Next, we examine the convergence horizons. The bounds (1.4) and (1.6) yield

$$h_{\text{simp}} = \frac{n\|\boldsymbol{e}\|_\infty^2}{\sigma_{\min}^2(\boldsymbol{A})} \quad \text{and} \quad h_{\text{block}} = \frac{\beta}{\alpha} \cdot \frac{\|\boldsymbol{e}\|_2^2}{\sigma_{\min}^2(\boldsymbol{A})}.$$

Their ratio satisfies

$$\frac{h_{\text{block}}}{h_{\text{simp}}} = \frac{\beta}{\alpha} \cdot \frac{\|\boldsymbol{e}\|_2^2}{n\|\boldsymbol{e}\|_\infty^2} \leq \frac{\beta}{\alpha}.$$

We see that the convergence horizon h_{block} for the block method never exceeds h_{simp} by a factor larger than the conditioning β/α of the row paving. But h_{block} may be substantially smaller than h_{simp} if the components of the residual vector are highly nonuniform.

To make more detailed claims about the relative merits of the two algorithms, we describe two situations where we have additional information about the structure of the matrix $\boldsymbol{A}$, or a lack thereof.

1.6.1. *Example 1: Fast Updates.* First, we consider the case where the computational cost of the block update rule (1.5) is roughly comparable with the cost of the simple update rule (1.3). This situation can occur when

- Each submatrix $\boldsymbol{A}_\tau$ admits a fast multiply; and
- Each submatrix $\boldsymbol{A}_\tau$ is well conditioned.

See Section 4.3.1 for a numerical example where these properties hold.

In this setting, one iteration of the block method has roughly the same cost as one iteration of the standard method. As a consequence, the comparison (1.7) of the convergence rates provides a reasonable assessment of how much each algorithm reduces the error per unit of arithmetic. We see that the block algorithm really is about $n/(\beta m)$ times faster than the simple Kaczmarz method. When βm is small in comparison with n, this represents a massive acceleration.

1.6.2. *Example 2: Unstructured Submatrices.* On the other hand, suppose that each submatrix $\boldsymbol{A}_\tau$ is unstructured and dense. Then the block method may involve much more arithmetic per iteration than the simple method. As a consequence, the comparison (1.7) is unfair to the simple method.

In this case, it is more appropriate to examine the convergence rate per *epoch*, the minimum number of iterations it takes the algorithm to touch each row of $\boldsymbol{A}$ once. For the simple Kaczmarz method, an epoch consists of n iterations; for the block method, an epoch consists of m iterations. In this setting, each algorithm requires about the same amount of arithmetic in one epoch, so we consider the per-epoch convergence rates:

$$m \cdot \rho_{\text{block}} \geq \frac{\sigma_{\min}^2(\boldsymbol{A})}{\beta} \quad \text{and} \quad n \cdot \rho_{\text{simp}} \geq \sigma_{\min}^2(\boldsymbol{A}).$$

We see that, in theory, the per-epoch convergence of the block method is worse, and the disadvantage increases with the upper bound β on the paving. The best case for the block method occurs if $\beta = 1$. This may happen, for example, when each block in the paving contains a single row of the matrix ($m = n$).

In practice, the block method displays better convergence behavior than this estimate suggests. The block method accrues further advantages because of subtle computational issues involving data transfer

and basic linear algebra subroutines (BLASx). We discuss these points in Section 4.1, and we provide some numerical support in Section 4.3.2.

1.7. **Existence of Good Pavings.** So far, we have assumed that the matrix $\boldsymbol{A}$ comes packaged with a natural row paving T. For some of the applications we have in mind, this hypothesis is reasonable. Nevertheless, the block Kaczmarz method would be more versatile if we could construct row pavings for a broad class of matrices. To that end, Popa [Pop99] has developed an approach for producing a paving of a sparse matrix; see also [Pop01, Pop04]. But we can travel much farther down this road. It is an astonishing fact that *every* standardized matrix admits a good row paving.

Proposition 1.3 (Existence of Good Row Pavings)**.** *Fix a number $\delta \in (0,1)$. Let $\boldsymbol{A}$ be a standardized matrix with n rows. Then $\boldsymbol{A}$ admits a row paving whose parameters satisfy*

$$m \le \mathrm{C}_{\mathrm{pave}} \cdot \delta^{-2} \|\boldsymbol{A}\|^2 \log(1+n) \quad \textit{and} \quad 1-\delta \le \alpha \le \beta \le 1+\delta.$$

The number $\mathrm{C}_{\mathrm{pave}}$ is a positive, universal[3] *constant.*

Proposition 1.3 follows most directly from the recent results [Ver06, Cor. 1.5] and [Tro09, Thm. 1.2], whose provenance can be traced to the celebrated papers [BT87, BT91]. Although Proposition 1.3 is only an existential result, the literature describes several efficient algorithms for constructing row pavings. In particular, under some additional conditions, it is possible to pave a matrix by partitioning its rows *at random*. See Section 3 for more results and background on paving.

1.7.1. *Understanding the Paving Theorem.* Before we continue, let us take a moment to explain some of the key aspects of Proposition 1.3. The main point is that the size of the paving depends only on the spectral norm of the matrix—not on the smallest singular value. As a consequence, it is possible to pave matrices with substantial null spaces!

A second point is that we can make the squared conditioning β/α as close to one as we desire. In particular, the choice $\delta = 0.5$ yields $\beta/\alpha \le 3$. This property licenses us to apply an iterative algorithm to solve least-squares problems involving the submatrices induced by the paving, as described in Section 1.4. We remark that the dependence of the paving size m on the parameter δ is optimal[4] as $\delta \to 0$.

Next, we develop some intuition about the role of the spectral norm in Proposition 1.3. Suppose that $\boldsymbol{A}$ is a standardized matrix with n rows. If the lower paving bound $\alpha > 0$, then a row paving T of $\boldsymbol{A}$ must contain at least $n/\mathrm{rank}(\boldsymbol{A})$ blocks. Otherwise, $\boldsymbol{A}_\tau$ is rank deficient for some $\tau \in T$ simply because this submatrix has more than $\mathrm{rank}(\boldsymbol{A})$ rows. Now, using the fact $\|\boldsymbol{A}\|_{\mathrm{F}}^2 \le \mathrm{rank}(\boldsymbol{A})\|\boldsymbol{A}\|^2$, we easily verify that

$$\|\boldsymbol{A}\|^2 \ge \frac{n}{\mathrm{rank}(\boldsymbol{A})}.$$

This bound is sharp. (Consider the two extreme examples: a matrix with orthonormal rows and a matrix with identical rows.) Therefore, we can view the squared spectral norm as a proxy for the minimal number of blocks in a row paving whose lower bound $\alpha > 0$.

We conclude that Proposition 1.3 delivers a row paving whose size falls within a logarithmic factor of optimal. One may wonder whether it is possible to remove the logarithm. For general matrices, this question remains open. It is known [And79, BHKW88, BT91] that an affirmative answer would imply the long-standing conjecture of Kadison and Singer [KS59].

[3] A *universal* constant has no dependence on any parameter.

[4] To verify this point, consider a large matrix $\boldsymbol{A}$ whose entries have equal magnitude and independent random signs. Use the Bai–Yin Law [BY93, Thm. 2] to estimate singular values and norms.

1.8. **Paved with Good Intentions.** We conclude the Introduction by merging our theorem on the convergence of the block Kaczmarz method with the result on the existence of pavings.

Corollary 1.4 (Block Kaczmarz with a Good Row Paving)**.** *Suppose that $\boldsymbol{A}$ is a standardized matrix with full column rank. Let T be a good row paving of $\boldsymbol{A}$, as guaranteed by Proposition 1.3 with $\delta = 1/2$. Under the notation of Theorem 1.2, the block Kaczmarz method, Algorithm 1.1 admits the convergence estimate*

$$\mathbb{E}\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|^2 \le \left[1 - \frac{1}{6\mathrm{C}_{\mathrm{pave}}\kappa^2(\boldsymbol{A})\log(1+n)}\right]^j \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 \;+\; \frac{3\|\boldsymbol{e}\|_2^2}{\sigma_{\min}^2(\boldsymbol{A})}$$

where $\mathrm{C}_{\mathrm{pave}}$ *is the constant from Proposition 1.3.*

To summarize, Proposition 1.3 yields a small paving of the matrix $\boldsymbol{A}$ with exceptional conditioning. With this choice of paving, we can perform the block Kaczmarz update (1.5) quickly using an iterative least-squares algorithm. (Indeed, to apply $\boldsymbol{A}_\tau^\dagger$ with a fixed level of precision, it suffices to perform a constant number of matrix–vector multiplies with $\boldsymbol{A}_\tau$ and $\boldsymbol{A}_\tau^*$.) Furthermore, we see that the block Kaczmarz method converges linearly with a rate that is controlled by the condition number of the matrix $\boldsymbol{A}$, and the convergence horizon is on the same order as the size of the residual. This is essentially the best outcome one might hope for.

1.9. **Organization.** The rest of the paper has the following structure. Section 2 contains a proof of the main result, Theorem 1.2. In Section 3, we give an overview of the literature on pavings. Section 4 discusses numerical aspects of the block Kaczmarz method. We discuss related work on Kaczmarz methods and future directions in Section 5. Finally, Appendix A offers a proof of a supplemental result.

2. Analysis of the Randomized Block Kaczmarz Algorithm

This section contains the proof of the main result, Theorem 1.2, on the convergence of Algorithm 1.1. We commence with two simple lemmas. The first step provides a deterministic bound on how much one iteration of the algorithm reduces the error.

Lemma 2.1. *Instate the hypotheses and notation of Theorem 1.2. Then the error at iteration j satisfies the deterministic bound*

$$\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 \le \|(\mathbf{I} - \boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau)(\boldsymbol{x}_{j-1} - \boldsymbol{x}_\star)\|_2^2 \;+\; \frac{1}{\alpha}\|\boldsymbol{e}_\tau\|_2^2,$$

where $\tau = \tau(j)$ is the block selected at iteration j.

Proof. According to the update rule (1.5), block Kaczmarz computes

$$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} + \boldsymbol{A}_\tau^\dagger(\boldsymbol{b}_\tau - \boldsymbol{A}_\tau \boldsymbol{x}_{j-1}) = \boldsymbol{x}_{j-1} + \boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau(\boldsymbol{x}_\star - \boldsymbol{x}_{j-1}) - \boldsymbol{A}_\tau^\dagger \boldsymbol{e}_\tau,$$

where we have introduced the decomposition $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_\star - \boldsymbol{e}$, restricted to the coordinates listed in τ. Subtract $\boldsymbol{x}_\star$ from both sides to obtain

$$\boldsymbol{x}_j - \boldsymbol{x}_\star = (\mathbf{I} - \boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau)(\boldsymbol{x}_{j-1} - \boldsymbol{x}_\star) - \boldsymbol{A}_\tau^\dagger \boldsymbol{e}_\tau.$$

The range of $\boldsymbol{A}_\tau^\dagger$ and the range of $\mathbf{I} - \boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau$ are orthogonal, so we may invoke the Pythagorean Theorem to reach

$$\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 = \|(\mathbf{I} - \boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau)(\boldsymbol{x}_{j-1} - \boldsymbol{x}_\star)\|_2^2 + \|\boldsymbol{A}_\tau^\dagger \boldsymbol{e}_\tau\|_2^2.$$

The second term on the right-hand side satisfies

$$\|\boldsymbol{A}_\tau^\dagger \boldsymbol{e}_\tau\|_2^2 \le \sigma_{\max}^2(\boldsymbol{A}_\tau^\dagger)\|\boldsymbol{e}_\tau\|_2^2 \le \frac{1}{\sigma_{\min}^2(\boldsymbol{A}_\tau)}\|\boldsymbol{e}_\tau\|_2^2 \le \frac{1}{\alpha}\|\boldsymbol{e}_\tau\|_2^2,$$

where α is the lower bound on the row paving T. Combine the last two displays to wrap up. □

The second lemma gives us a means to average the two quantities appearing in Lemma 2.1 over a random choice of the block $\tau = \tau(j)$.

Lemma 2.2. *Instate the hypotheses and notation of Theorem 1.2. Suppose that τ is chosen uniformly at random from the row paving T. For fixed vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, it holds that*

$$\mathbb{E}\|(\mathbf{I} - \boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau)\boldsymbol{u}\|_2^2 \le \left[1 - \frac{\sigma_{\min}^2(\boldsymbol{A})}{\beta m}\right]\|\boldsymbol{u}\|_2^2 \quad \textit{and} \quad \mathbb{E}\|\boldsymbol{v}_\tau\|_2^2 = \frac{1}{m}\|\boldsymbol{v}\|_2^2.$$

Proof. The second identity emerges from a very short calculation:

$$\mathbb{E}\|\boldsymbol{v}_\tau\|_2^2 = \frac{1}{m}\sum\nolimits_{\omega\in T}\|\boldsymbol{v}_\omega\|_2^2 = \frac{1}{m}\|\boldsymbol{v}\|_2^2,$$

which depends on the fact that the blocks of T partition the components of $\boldsymbol{v}$.

Since $\boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau$ is an orthogonal projector, we may apply the Pythagorean Theorem to obtain the relation

$$\mathbb{E}\|(\mathbf{I} - \boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau)\boldsymbol{u}\|_2^2 = \|\boldsymbol{u}\|_2^2 - \mathbb{E}\|\boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau \boldsymbol{u}\|_2^2.$$

We control the remaining expectation as follows.

$$\begin{aligned}\mathbb{E}\|\boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau \boldsymbol{u}\|_2^2 &\ge \mathbb{E}\left[\sigma_{\min}^2(\boldsymbol{A}_\tau^\dagger)\|\boldsymbol{A}_\tau \boldsymbol{u}\|_2^2\right] \ge \frac{1}{\beta}\cdot\mathbb{E}\|\boldsymbol{A}_\tau \boldsymbol{u}\|_2^2 \\ &= \frac{1}{\beta m}\sum_{\omega\in T}\|\boldsymbol{A}_\omega \boldsymbol{u}\|_2^2 = \frac{1}{\beta m}\|\boldsymbol{A}\boldsymbol{u}\|_2^2 \ge \frac{\sigma_{\min}^2(\boldsymbol{A})}{\beta m}\|\boldsymbol{u}\|_2^2.\end{aligned}$$

The second inequality depends on the bound $\sigma_{\min}^2(\boldsymbol{A}_\tau^\dagger) = \sigma_{\max}^{-2}(\boldsymbol{A}_\tau) \ge \beta^{-1}$. The fourth relation holds because the blocks in a paving partition the row indices of $\boldsymbol{A}$. To complete the proof, we simply combine the last two displays. □

The main result follows quickly once we merge the two lemmas.

Proof of Theorem 1.2. First, we bound the expected error at iteration j in terms of the error at iteration $j-1$. Average the bound from Lemma 2.1 over the randomness in $\tau = \tau(j)$ to reach

$$\begin{aligned}\mathbb{E}_\tau\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 &\le \mathbb{E}_\tau\|(\mathbf{I} - \boldsymbol{A}_\tau^\dagger \boldsymbol{A}_\tau)(\boldsymbol{x}_{j-1} - \boldsymbol{x}_\star)\|_2^2 + \frac{1}{\alpha}\cdot\mathbb{E}_\tau\|\boldsymbol{e}_\tau\|_2^2 \\ &\le \left[1 - \frac{\sigma_{\min}^2(\boldsymbol{A})}{\beta m}\right]\|\boldsymbol{x}_{j-1} - \boldsymbol{x}_\star\|_2^2 + \frac{1}{\alpha m}\|\boldsymbol{e}\|_2^2.\end{aligned}$$

The second inequality follows from Lemma 2.2, with $\boldsymbol{u} = \boldsymbol{x}_{j-1} - \boldsymbol{x}_\star$ and $\boldsymbol{v} = \boldsymbol{e}$.

By applying this result repeatedly, we can control the expected error after j iterations in terms of the initial error. Abbreviating $\gamma := 1 - \sigma_{\min}^2(\boldsymbol{A})/(\beta m)$, we obtain the estimate

$$\begin{aligned}\mathbb{E}\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 &= \mathbb{E}_{\tau(1)}\mathbb{E}_{\tau(2)}\cdots\mathbb{E}_{\tau(j)}\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2^2 \\ &\le \gamma^j\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 + \frac{1}{\alpha m}\|\boldsymbol{e}\|_2^2\left(\textstyle\sum_{i=0}^{j-1}\gamma^i\right) \\ &\le \gamma^j\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 + \frac{1}{\alpha m(1-\gamma)}\|\boldsymbol{e}\|_2^2.\end{aligned}$$

Reintroduce the value of γ in this expression, and simplify to complete the proof. □

3. A Conversation about Pavings

As we have seen, the properties of the row paving T of the matrix $\boldsymbol{A}$ have a significant effect on the behavior of the block Kaczmarz method, Algorithm 1.1. It is natural to ask if every standardized matrix $\boldsymbol{A}$ admits a good paving, and—if so—how we can exhibit such a paving.

This section summarizes the main results from the literature on row pavings, with particular attention to algorithmic techniques for constructing pavings. We focus on two methods in particular. The first approach, which is more general, extracts a well-conditioned row submatrix from $\boldsymbol{A}$ and repeats this process until the paving is complete. The second approach, which is more automatic, simply forms a random partition of the rows with an appropriate number of blocks.

For clarity, we only discuss row paving theorems for standardized matrices. For general matrices, it is often more natural to consider an alternative definition of a paving where the rows of the matrix are reweighted. For the reader's convenience, we include some citations that address the general case.

Remark (Paving a Square Matrix). The operator theory literature uses the term *paving* to refer to a partition $T = \{\tau_1, \dots, \tau_m\}$ of the coordinates of a *square* matrix $\boldsymbol{H}$ with a zero diagonal in which each diagonal block satisfies the bound

$$\|\boldsymbol{H}_{\tau\tau}\| \le (1-\delta)\|\boldsymbol{H}\| \quad \text{for each } \tau \in T,$$

where the parameter $\delta \in (0,1)$. We can obtain row paving results for a standardized matrix $\boldsymbol{A}$ by applying paving results for a square matrix to the hollow Gram matrix $\boldsymbol{H} = \boldsymbol{A}\boldsymbol{A}^* - \mathbf{I}$. This approach generally leads to an estimate for the size m of the paving that has an excessive dependence on the spectral norm of $\boldsymbol{A}$.

3.1. Subset Selection Theorems.

The first approach to paving relies on a type of result called a *subset selection theorem*. This class of result asserts that, under appropriate conditions, a matrix contains a (large) set of rows with distinguished geometric properties. Proposition 1.3 depends on the following subset selection theorem.

Proposition 3.1 (Subset Selection)**.** *Fix a number $\delta \in (0,1)$. Let $\boldsymbol{A}$ be a standardized matrix with n rows. Then there exists a subset τ of row indices with the properties*

$$|\tau| \ge \frac{\mathrm{c_{ss}} \cdot \delta^2 n}{\|\boldsymbol{A}\|^2} \quad \textit{and} \quad 1-\delta \le \lambda_{\min}(\boldsymbol{A}_\tau \boldsymbol{A}_\tau^*) \le \lambda_{\max}(\boldsymbol{A}_\tau \boldsymbol{A}_\tau^*) \le 1+\delta.$$

The number $\mathrm{c_{ss}}$ is a positive, universal constant.

Proposition 3.1 follows from [Tro09, Thm. 1.2], once we track the parameter δ through the proof. This result has been attributed to Bourgain and Tzafriri [BT91], but the earliest reference seems to be Vershynin's paper [Ver06, Cor. 1.5]. See Section 3.1.1 for further background.

Proposition 3.1 ensures that each standardized matrix contains a large set of well-conditioned rows. To construct a paving of a standardized matrix $\boldsymbol{A}$ with row indices $\{1, \dots, n\}$, we apply this result to identify a large subset τ_1 of the row indices. We apply the same result to the set of remaining rows $\{1, \dots, n\} \setminus \tau_1$ to bite off another subset τ_2, and so forth. After

$$m \le \mathrm{C_{pave}} \cdot \delta^{-2} \|\boldsymbol{A}\|^2 \log(1+n)$$

steps, we have exhausted the entire matrix. This argument yields Proposition 1.3.

The paper [Tro09] contains an efficient computational method for identifying the subset τ promised by Proposition 3.1. This algorithm chooses a random set ω of rows from the matrix $\boldsymbol{A}$ with twice the cardinality of the desired subset τ. Then it computes a matrix factorization of the submatrix $\boldsymbol{A}_\omega$ that exposes a well-conditioned subset τ of rows inside ω.

3.1.1. *Related Results.* The literature contains a variety of subset selection theorems that can be used to construct pavings. The first major result in this direction is the Restricted Invertibility Principle of Bourgain and Tzafriri [BT87, Thm. 1.2], which guarantees that every standardized matrix contains a set of $\text{const}/\|\boldsymbol{A}\|^2$ rows whose minimal singular value is bounded away from zero. In a similar vein, Kashin and Tzafriri [KT94] establish that every standardized matrix contains a set of $\text{const}/\|\boldsymbol{A}\|^2$ rows whose norm is constant. Both results are based on random selection combined with matrix factorization. Neither result yields a submatrix whose singular values lie arbitrarily close to one.

In another notable paper, Bourgain and Tzafriri [BT91, Cor. 1.2] prove that every standardized matrix contains a set of $\text{const}\cdot\delta^2/\|\boldsymbol{A}\|^4$ rows whose squared singular values fall in the range $[1-\delta, 1+\delta]$. This theorem offers quantitative control on the conditioning of the submatrix, but it gives the wrong dependence on the spectral norm of the matrix. Proposition 3.1 is based on the same type of argument as this result of Bourgain and Tzafriri. The iterative argument we use to draw the paving result, Proposition 1.3, as a corollary of Proposition 3.1 also appears in the paper [BT91].

The last few years have witnessed some striking advances in this area. Indeed, Spielman and Srivastava [SS12] have recently invented an elementary proof of the Restricted Invertibility Principle. Their method only involves linear algebra, and it leads to sharp constants. Youssef [You12b, Thm. 4.2] has adapted these ideas to obtain an elementary proof of the Kashin–Tzafriri theorem [KT94]. These results are appealing because they construct the required subsets using an algorithmic procedure that admits a polynomial-time implementation. See [Sri10, Nao11] for further exposition.

Vershynin [Ver01] has obtained a theory of subset selection for matrices that are not necessarily standardized. In his results, the squared Frobenius norm $\|\boldsymbol{A}\|_{\mathrm{F}}^2$ plays the role of the number n of rows. Srivastava's dissertation [Sri10, Chap. 3] contains an algorithm for (weighted) subset selection that applies to general matrices. See Youssef's works [You12b, You12a] for the latest developments in this direction.

Finally, let us mention that subset selection theorems have applications throughout mathematics and engineering. See the paper [CT06] for a discussion and references.

3.2. **Randomized Methods for Paving.** The second approach to paving has the benefit of utmost simplicity, but it is more limited in scope. The idea is to divide the rows of the matrix into random blocks of approximately equal size. Under additional assumptions, each submatrix induced by this partition is likely to be well conditioned. To describe this idea in more detail, we first introduce the concept of a random partition.

Definition 3.2 (Random Partition)**.** Suppose that π is a permutation on $\{1,2,\dots,n\}$, chosen uniformly at random. For each $i = 1,2,\dots,m$, define the set

$$\tau_i = \{\pi(k) : k = \lfloor (i-1)n/m \rfloor + 1, \lfloor (i-1)n/m \rfloor + 2, \dots, \lfloor in/m \rfloor\}.$$

It is clear that $T = \{\tau_1, \tau_2, \dots, \tau_m\}$ is a partition of $\{1,2,\dots,n\}$ into m blocks of approximately equal size. We say that T is a *random partition* of $\{1,2,\dots,n\}$ into m blocks.

For every standardized matrix, we can use a random partition to construct a paving whose upper bound β is relatively small.

Proposition 3.3 (Random Paving: Upper Bound)**.** *Let $\boldsymbol{A}$ be a tall*[5]*, standardized matrix with n rows. Consider a randomized partition T of the row indices with $m \geq \|\boldsymbol{A}\|^2$ blocks. Then T is a row paving with upper bound $\beta \leq 6\log(1+n)$, with probability at least $1 - n^{-1}$.*

Proposition 3.3 results from an argument based on the matrix Chernoff inequality [Tro12, Thm. 1.1] and a union bound. A model for this type of proof appears in the paper [Tro11]. We omit the details.

[5] A $p \times q$ matrix is *tall* when $p \geq q$.

In contrast, if we wish to construct a paving with a nontrivial lower bound α, we must place additional assumptions on the matrix. An example [BT91, Ex. 2.2] of Bourgain and Tzafriri implies that we must assume the rows of the matrix $\boldsymbol{A}$ are weakly correlated to obtain a random paving with $\alpha > 0$. It is natural to carve out a class of matrices that meet this requirement.

Definition 3.4 (Incoherence)**.** Suppose $\boldsymbol{A}$ is a matrix with rows $\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_n$. We say that $\boldsymbol{A}$ is *incoherent* when

$$\max_{i \neq \ell} |\langle \boldsymbol{a}_i, \boldsymbol{a}_\ell \rangle| \leq \frac{\mathrm{c_{inc}}}{\log(1+n)}.$$

The number $\mathrm{c_{inc}}$ is a positive, universal constant.

Incoherent matrices arise, for example, in signal processing problems [DH01]. Every incoherent, standardized matrix admits a random paving with controlled lower and upper bounds.

Proposition 3.5 (Random Paving)**.** *Suppose that $\boldsymbol{A}$ is an incoherent, standardized matrix with n rows. Let T be a random partition of the row indices into m blocks where $m \geq \mathrm{C_{rand}} \cdot \delta^{-2} \|\boldsymbol{A}\|^2 \log(1+n)$. Then T is a row paving of $\boldsymbol{A}$ whose paving bounds satisfy $1-\delta \leq \alpha \leq \beta \leq 1+\delta$, with probability at least $1 - n^{-1}$.*

Proposition 3.5 follows from [Tro08a, Cor. 5.2], along with some standard arguments [BT91, Tro08b]. The paper [CD12] contains superior estimates for the constants in this analysis. See Section 3.2.3 for some further background.

Random paving is a striking idea because it is almost completely automatic. Given a guarantee that the matrix $\boldsymbol{A}$ is incoherent and an estimate for the spectral norm, we can obtain a good paving of the matrix without any further computation.

3.2.1. *The Fast Incoherence Transform.* Prima facie, random paving has limited applicability because the incoherence hypothesis in Proposition 3.5 is rather stringent. Fortunately, there is a fast linear transformation that can be used to convert any standardized matrix into an incoherent matrix that is nearly standardized.

Definition 3.6 (Fast Incoherence Transform)**.** The *fast incoherence transform* is the $n \times n$ random matrix $\boldsymbol{S} = \mathbf{F}\boldsymbol{E}$ where $\mathbf{F}$ is the $n \times n$ unitary discrete Fourier transform (DFT) and $\boldsymbol{E}$ is an $n \times n$ diagonal matrix whose entries are independent Rademacher[6] random variables.

The matrix $\boldsymbol{S}$ is unitary, so it preserves Euclidean structure in a problem. Furthermore, $\boldsymbol{S}$ can be applied to a vector in $\mathrm{O}(n \log n)$ arithmetic operations by means of the FFT algorithm. Thus, for an $n \times d$ matrix $\boldsymbol{A}$, we can form the product $\boldsymbol{SA}$ with $\mathrm{O}(nd \log n)$ arithmetic operations. When the matrix $\boldsymbol{A}$ is sparse or admits a fast matrix–vector multiply, it may be more appropriate to work directly with the factorized representation $\boldsymbol{SA}$, because it inherits a fast multiply from its two factors. To emphasize, in many contexts, it is unnecessary to form $\boldsymbol{SA}$ explicitly.

The critical fact is that the fast incoherence transform converts a large class of standardized matrices into incoherent matrices that are nearly standardized.

Proposition 3.7 (Fast Incoherence Transform)**.** *Suppose that $\boldsymbol{A}$ is a standardized matrix with n rows whose norm satisfies*

$$\|\boldsymbol{A}\|^2 \leq \frac{\mathrm{c_{fit}} \cdot n}{\log^3(1+n)}. \tag{3.1}$$

Let $\boldsymbol{S}$ be an $n \times n$ fast incoherence transform. Then the matrix $\boldsymbol{W} = \boldsymbol{SA}$ satisfies the probability bound

$$\mathbb{P}\left\{ \max_{i,\ell} |\langle \boldsymbol{w}_i, \boldsymbol{w}_\ell \rangle - \delta_{i\ell}| \geq \frac{\mathrm{c_{inc}}}{\log(1+n)} \right\} \leq \frac{1}{n},$$

where $\boldsymbol{w}_i$ is the ith row of $\boldsymbol{W}$ and $\delta_{i\ell}$ is the Kronecker delta.

[6] A *Rademacher* random variable takes the values ± 1 with equal probability.

We do not have a reference for Proposition 3.7, but the result is probably not new. Appendix A offers a short proof based on the Hanson–Wright inequality [HW71] for Rademacher chaos.

Let us pause to examine the hypothesis (3.1). Recall that, for a standardized matrix $\boldsymbol{A}$ with n rows, the squared spectral norm $\|\boldsymbol{A}\|^2$ attains its maximal value n when the rows are identical. Therefore, the bound (3.1) stipulates that the rows of $\boldsymbol{A}$ must exhibit a small amount of diversity. In this case, the fast incoherence transform spreads out the diversity evenly so that no pair of columns is correlated too strongly.

3.2.2. *Incoherence, Paving, Kaczmarz.* This discussion suggests the following approach for solving the overdetermined least-squares problem (1.1) with a standardized matrix $\boldsymbol{A}$:

(1) Apply the fast incoherence transform $\boldsymbol{S}$ to the objective function:

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \|\boldsymbol{S}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{S}\boldsymbol{b}\|_2^2 =: \|\tilde{\boldsymbol{A}}\boldsymbol{x} - \tilde{\boldsymbol{b}}\|_2^2.$$

(2) Draw a random partition T of the rows of $\tilde{\boldsymbol{A}}$ with the number of blocks determined according to Proposition 3.5.
(3) Apply the randomized block Kaczmarz method, Algorithm 1.1, with the matrix $\tilde{\boldsymbol{A}}$, the right-hand side $\tilde{\boldsymbol{b}}$, and the paving T to solve the transformed problem.

Provided that the hypothesis (3.1) is in force, Proposition 3.7 shows that the fast incoherence transform is likely to convert the matrix $\boldsymbol{A}$ into an incoherent matrix $\tilde{\boldsymbol{A}}$. Proposition 3.5 shows that we can obtain a good paving of $\tilde{\boldsymbol{A}}$ from a random partition of the rows. If these randomized procedures are successful, when we apply the block Kaczmarz algorithm to solve the transformed least-squares problem, we obtain the same convergence guarantees outlined in Corollary 1.4.

For a well-conditioned $n \times d$ matrix $\boldsymbol{A}$ with $d \gg \log n$, we can use this approach to solve the least-squares problem to a fixed level of precision after $\mathrm{O}(nd \log n)$ arithmetic operations. In theory, this bound is almost comparable with the conjugate gradient method, which requires $\mathrm{O}(nd)$ operations to achieve fixed precision in this setting.

3.2.3. *Related Results.* The idea that randomness might help us to construct a paving is already inherent in the Bourgain–Tzafriri paper [BT87] on the Restricted Invertibility Principle, where they use randomized row selection and matrix factorization to perform subset selection. A result [BT91, Thm. 2.3] in their subsequent paper demonstrates that, in the presence of incoherence assumptions, a random partition induces a row paving with $\mathrm{const} \cdot \|\boldsymbol{A}\|^4 \log(1+n)$ blocks; the extra factorization step is not necessary. Furthermore, under an incoherence assumption slightly stricter than Definition 3.4, they prove that a random partition induces a paving with $\mathrm{const} \cdot \|\boldsymbol{A}\|^4$ blocks; no logarithmic factor is necessary. See the paper [Tro08b] for a modern proof of the latter result.

The aforementioned theorems all yield the wrong dependence on the spectral norm in the size of a random row paving. The précis [Tro08a] shows how to obtain the correct quadratic dependence that is quoted in Proposition 3.5. If we were to strengthen the incoherence requirement in Proposition 3.5, it is likely that we could remove the logarithmic factor from the size of the paving by adapting an argument [BT91, Prop. 2.7] of Bourgain and Tzafriri. On the other hand, the logarithmic factor is necessary at the incoherence level we have imposed [BT91, Ex. 2.2].

The fast incoherence transform is based on ideas of Ailon and Chazelle [AC09], who use the random matrix $\boldsymbol{S}$ to perform dimension reduction. We believe that the first application of $\boldsymbol{S}$ for randomized linear algebra appears in the paper Woolfe et al. [WLRT08], where they use this transform to aid in computing matrix decompositions. See the works [AMT10, HMT11, BG12] for further results in this direction. Liberty's dissertation [Lib09] describes other randomized maps that can play a similar role.

4. Numerical Aspects of Block Kaczmarz

The main goal of this paper is to study the theoretical properties of the block Kaczmarz method, but we believe that a short discussion about numerics can also provide some useful insights. In Section 4.1, we outline some of the situations where we expect the block Kaczmarz method to outperform the simple Kaczmarz algorithm. Afterward, in Section 4.2, we consider some of the questions that arise when implementing the block Kaczmarz method. Finally, in Section 4.3, we offer some simple computational examples to illustrate our main points.

4.1. **Why Use Block Kaczmarz?** It is natural to ask when it might be better to use the block Kaczmarz scheme instead of the simple Kaczmarz scheme. The answer to this question depends heavily on the specific application at hand, as well as the architecture of the computers on which the algorithm is implemented.

First, least-squares problems sometimes involve a matrix that admits a natural row paving. In this case, it may be advantageous to exploit the paving algorithmically. For example, certain signal processing applications involve multi-sampling schemes, where we collect several batches of uniform time samples of a signal. Each sample set produces a set of equations that is easy to solve. The block Kaczmarz method provides an effective way to use this structure [FS95]. See Section 4.3.1 for a related numerical example.

Second, the block Kaczmarz algorithm can be implemented more efficiently than the simple Kaczmarz algorithm in many computer architectures. This claim rests on two facts. First, data transfer now plays a major role in the cost of numerical algorithms. The block Kaczmarz algorithm is efficient in this regard, because it moves a large block of equations into working memory and operates with it for some time. In contrast, the simple Kaczmarz method repeatedly transfers new equations into working memory. Second, the block Kaczmarz algorithm can exploit high-level basic linear algebra subroutines (`BLASx`). Indeed, inner products dominate the arithmetic in the simple Kaczmarz method, while it is possible to implement the block Kaczmarz algorithm using matrix–vector products. As a result, block Kaczmarz relies on `BLAS2`, rather than `BLAS1`. A more detailed discussion of these issues falls outside the scope of this paper.

4.2. **Implementing Block Kaczmarz Methods.** The randomized block Kaczmarz method, Algorithm 1.1, is easy to describe, but there remain several implementation issues that require attention.

4.2.1. *The Block Update Rule.* Most of the arithmetic in the algorithm occurs when we apply the pseudoinverse $\boldsymbol{A}_\tau^\dagger$ to a vector in the update (1.5). Equivalently, we must solve an (underdetermined) least-squares problem at each iteration. The appropriate numerical method depends on a wide variety of issues [Bjö96], so we cannot give a universal prescription. Here are some factors worth considering.

- When the blocks in the paving are very small, a direct method based on QR decomposition or the SVD is likely to be fastest. A direct method may also be appropriate when the conditioning β/α of the paving is high and the matrix is dense.
- In case the conditioning β/α of the paving is small, we recommend using an iterative method, such as `CGLS`, `LSQR`, or the Chebyshev semi-iterative method. These techniques may also be appropriate for very sparse problems. It is not necessary to run these iterations until they have converged fully, and the algorithms will benefit from warm starts provided by the convergence of the outer iteration.

4.2.2. *Setting the Convergence Tolerance.* Theorem 1.2 implies that Algorithm 1.1 can reduce the error $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_\star\|_2^2$ to a level comparable with the convergence horizon. Let us examine how this fact affects our choice of the convergence tolerance.

Combine the convergence criterion $\|\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{b}\|_2^2 \le \varepsilon^2$ with the decomposition $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_\star - \boldsymbol{e}$ to obtain

$$\|\boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_\star) + \boldsymbol{e}\|_2^2 \le \varepsilon^2.$$

The residual $\boldsymbol{e}$ is orthogonal to the range of $\boldsymbol{A}$, so we can apply the Pythagorean theorem and bound the ℓ_2 norm below to reach

$$\sigma_{\min}^2(\boldsymbol{A})\|\hat{\boldsymbol{x}} - \boldsymbol{x}_\star\|_2^2 + \|\boldsymbol{e}\|_2^2 \le \varepsilon^2.$$

It follows that we must set the convergence tolerance ε so that

$$\varepsilon^2 > \left[1 + \frac{\beta}{\alpha}\right]\|\boldsymbol{e}\|_2^2. \tag{4.1}$$

Otherwise, we have no guarantee that the algorithm will terminate.

4.2.3. *Checking for Convergence.* The convergence criterion $\|\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{b}\|^2 \le \varepsilon^2$ may be somewhat expensive to verify because it involves a multiply with the full matrix $\boldsymbol{A}$. As a consequence, it is better to check for convergence only on occasion. For instance, if the paving consists of m blocks, we might only evaluate the convergence criterion every m iterations.

4.2.4. *Randomized Cyclic Control.* Finally, in practice, the block Kaczmarz algorithm is more effective if we use a randomized control scheme different from the one we have analyzed. Recall that Algorithm 1.1 samples a uniformly random block at each iteration, independent of all previous choices. Instead, we recommend using an alternative scheme that samples blocks *without* replacement. We can express this method formally as follows.

(1) For each $q = 0, 1, 2, \dots$, draw an independent, uniformly random permutation π_q on the indices $\{1, \dots, m\}$ of the blocks.
(2) For each iteration $j \ge 1$, decompose $j = qm + r$, where q and r are nonnegative integers with $0 \le r < m$. Select the block $\tau(j) = \tau_{\pi_q(r+1)}$.

In other words, at each epoch, we cycle through all the blocks in random order.

In Section 4.3.1, we offer some numerical evidence that this alternative control scheme is more effective than the approach used in Algorithm 1.1. At present, compelling explanations for this phenomenon are lacking. See [RR12] for some discussion and conjectures.

4.3. **Numerical Experiments.** In this section, we present some numerical experiments to complement our discussions about the implementation and theoretical performance of the randomized block Kaczmarz method, Algorithm 1.1. In Section 4.3.1, we consider an example where the block method has a clear advantage over the simple method. In Section 4.3.2, we examine some other cases where the benefits of the block method are more subtle.

4.3.1. *Submatrices with Fast Multiplies.* First, we study the situation where the matrix $\boldsymbol{A}$ comes with a natural partition of the rows into well-conditioned blocks, each admitting a fast matrix–vector multiply. As we have discussed in Section 1.6.1, the block method has a significant advantage over the simple method in this setting. Our experiments bear out this point.

We build a 300×100 matrix $\boldsymbol{A}_{\text{circ}}$ by stacking 15 partial random circulant matrices:

$$\boldsymbol{A}_{\text{circ}} = \begin{bmatrix} \boldsymbol{C}_1 \\ \vdots \\ \boldsymbol{C}_{15} \end{bmatrix} \quad \text{where} \quad \boldsymbol{C}_i = \boldsymbol{R}\mathbf{F}^*\boldsymbol{E}_i\mathbf{F} \quad \text{for } i = 1, \dots, 15.$$

In this expression,

- $\boldsymbol{R}$ is the restriction to the first 20 coordinates,
- $\mathbf{F}$ is the 100×100 unitary DFT, and

- $\boldsymbol{E}_i$ is a random diagonal matrix whose entries are independent Rademacher random variables. Each $\boldsymbol{E}_i$ is drawn independently from the others.

This construction ensures that each $\boldsymbol{C}_i$ is a section of a circulant matrix with orthonormal rows. As a consequence, the pseudoinverse equals the adjoint: $\boldsymbol{C}_i^\dagger = \boldsymbol{C}_i^*$. Furthermore, we can apply either $\boldsymbol{C}_i$ or $\boldsymbol{C}_i^*$ to a vector quickly[7] using one FFT and one inverse FFT, each of length d.

Using this matrix, we perform a small MATLAB experiment to compare the behavior of randomized block Kaczmarz and randomized simple Kaczmarz.

(1) Draw a random matrix $\boldsymbol{A}_{\text{circ}}$ according to the recipe above, and fix it. Let T be the row partition with 15 blocks induced by the circulant structure.
(2) Let $\boldsymbol{x}_\star = [1, \dots, 1]^*$, and form $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_\star$. Set the initial iterate $\boldsymbol{x}_0 = \boldsymbol{0}$.
(3) For each of 100 trials,
   (a) Apply the simple Kaczmarz method from Section 1.2 to produce iterates $\{\boldsymbol{x}_j^{\text{simp}} : j \geq 0\}$.
   (b) Apply Algorithm 1.1 to produce iterates $\{\boldsymbol{x}_j^{\text{block}} : j \geq 0\}$.
(4) For each algorithm, at each iteration j, compute the minimum, median, and maximum value of the approximation error $\|\boldsymbol{x}_j^{\text{alg}} - \boldsymbol{x}_\star\|_2$ over the 100 trials.

In this experiment, we form the matrix $\boldsymbol{A}_{\text{circ}}$ in the first step and store it. To perform the update (1.3), the simple Kaczmarz method loads the required row from memory without doing any extra computation, so the cost of the simple update rule is just $4d$ complex flops, where $d = 100$. The block Kaczmarz method uses the structure of the circulant blocks, as described in the previous paragraph, to perform the update (1.5) with about $4d \log_2(d) + 4d$ complex flops.

In Figure 1 [left panel], we plot the approximation error as a function of the number of flops expended by each algorithm. The heavy lines indicate the median error over 100 trials, while the minimum and maximum errors describe the boundaries of the shaded region. The block algorithm reduces the error to 10^{-11} in about 1.6×10^6 complex flops, while the simple algorithm requires about 3.2×10^7 complex flops to achieve the same result. This amounts to a 20-fold reduction in the amount of arithmetic!

In Section 4.2.4, we claim that the block Kaczmarz algorithm is more effective when we use an alternative control scheme that samples blocks without replacement. To illustrate this point, let us perform an experiment with the block circulant matrix using the same methodology as above. Figure 1 [right panel] compares the performance of the block Kaczmarz method when we sample blocks with and without replacement. This chart shows that sampling without replacement reduces the amount of computation by about 15%. In other experiments, we have seen even more substantial improvements.

4.3.2. *Unstructured Submatrices.* Next, we consider some examples where we cannot exploit the structure of the submatrices to accelerate the block Kaczmarz method. Although our theory does not yield higher rates of convergence, the numerical evidence still suggests that the block Kaczmarz method offers a decisive improvement over the simple Kaczmarz method.

First, we consider a 300×100 real matrix $\boldsymbol{A}_{\text{rdm}}$ whose rows are drawn independently and uniformly at random from the ℓ_2 unit sphere. We partition the rows of this matrix into 10 equal blocks of 30 consecutive rows each. A heuristic application of the Bai–Yin Law [BY93, Thm. 2] shows that the singular values of each 30×100 submatrix fall in the range $1 \pm \sqrt{30/100}$. Thus, the paving parameters $\alpha \approx 0.2$ and $\beta \approx 2.4$, and the condition number of each block is bounded by $\beta/\alpha \approx 4.8$.

We perform an experiment with this matrix that follows the same methodology as in Section 4.3.2. To implement the block Kaczmarz update rule (1.5), we use the `CGLS` algorithm to solve the underdetermined least-squares problem. We use warm starts, and we halt the least-squares iteration before it

[7] Recall that an FFT or inverse FFT of length d requires roughly $d \log_2(d)$ complex floating-point operations (flops), although the precise count depends on arithmetic properties of the number d.

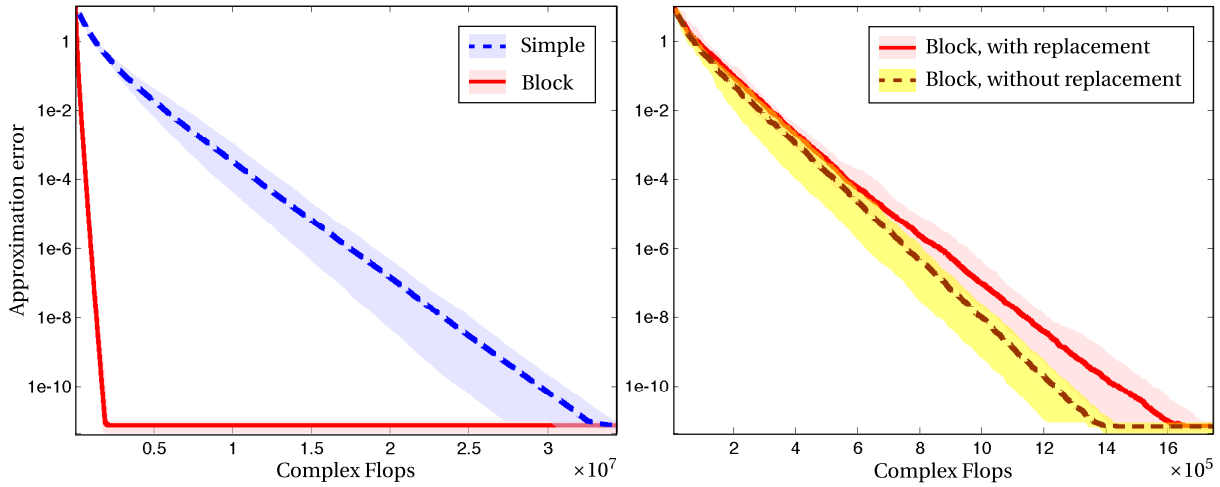

Figure 1 (Convergence with a Block Circulant Matrix)**.** The matrix $\boldsymbol{A}_{\text{circ}}$ is a fixed 300×100 matrix consisting of 15 partial circulant blocks. For each algorithm, we chart the approximation error $\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2$ as a function of the number of complex flops performed. The heavy line denotes the median error over 100 independent trials; the minimum and maximum errors envelop the shaded region. See Section 4.3.1 for details. **[Left]** Convergence of the randomized block Kaczmarz method, Algorithm 1.1, versus the randomized simple Kaczmarz method (Section 1.2). **[Right]** Convergence of two variants (Section 4.2.4) of the randomized block Kaczmarz method. One samples blocks independently with replacement; the other samples blocks without replacement in each epoch.

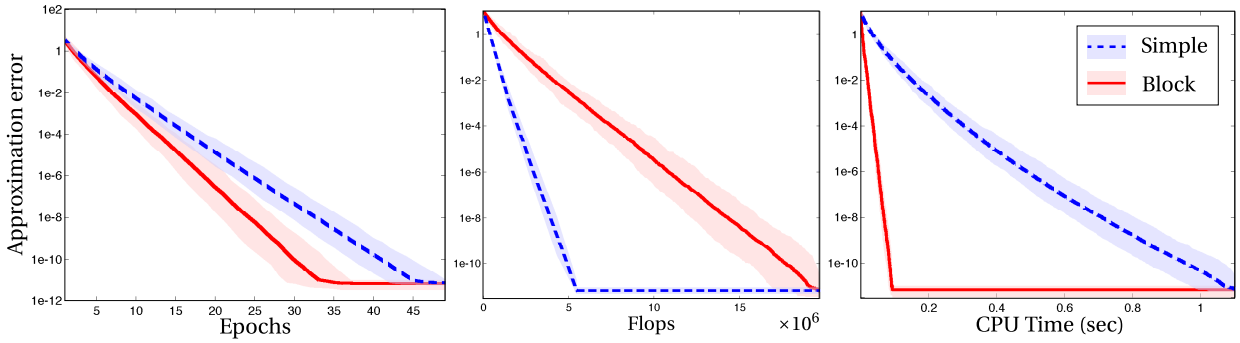

Figure 2 (Convergence with a Random Matrix)**.** The matrix $\boldsymbol{A}_{\text{rdm}}$ is a fixed 300×100 matrix whose rows are drawn uniformly at random from the Euclidean unit sphere and partitioned into 10 blocks of equal size. For the iterates generated by each algorithm, we plot the decay of the approximation error $\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2$ as a function of three computational resources. The heavy line denotes the median error over 100 independent trials; the minimum and maximum errors envelop the shaded region. **[Left]** Approximation error as a function of the number of epochs elapsed. **[Center]** Approximation error as a function of flops. **[Right]** Approximation error as a function of CPU time. See Section 4.3.2 for details.

has fully converged to control the computational cost. Our code counts the actual number of (real) flops during each iteration, and it measures the actual CPU time that is expended.

Figure 2 shows three different views of the data we collected during this experiment. The left panel shows the approximation error for the simple and block Kaczmarz method as a function of the number of epochs elapsed. As we expect from the discussion in Section 1.6.2, the two algorithms show a similar rate of convergence in this view. In fact, the block method does somewhat better than the theory predicts. In the center panel, we chart the approximation error as a function of the number of flops performed.

The simple algorithm requires about 5×10^6 flops to achieve an error of 10^{-11}, while the block algorithm takes 2×10^7 flops to reach the same point. Thus, the block method needs four times as much arithmetic. But the story is not over. The right panel displays the approximation error as a function of CPU time. We discover that our implementation of the block method is 10 times faster than the simple method! It is dangerous to draw conclusions about the efficiency of an algorithm from the behavior of a MATLAB script. Nevertheless, we regard this experiment as limited evidence of the computational advantages of the block method that we outlined in Section 4.1.

Finally, we consider one other type of unstructured matrix, where our analysis offers little guidance. Let $\tilde{\boldsymbol{A}}_{\text{coh}}$ be a 300×100 matrix whose entries are drawn independently at random from the uniform distribution on $[0.5, 1]$. We form a standardized matrix $\boldsymbol{A}_{\text{coh}}$ by normalizing the rows of $\tilde{\boldsymbol{A}}_{\text{coh}}$. Since this matrix is dense and positive, the rows of this matrix tend to be strongly correlated with each other; the maximum inner product between rows is typically 0.98 or higher. Furthermore, the norm of this matrix is usually close to the maximum possible value $\sqrt{300}$. Our theory provides no map for this *terra incognita*. Yet we brazenly partition the rows into 10 equal blocks, each with 30 contiguous rows, as in the previous example.

We perform the same experiment for $\boldsymbol{A}_{\text{coh}}$ as we did with $\boldsymbol{A}_{\text{rdm}}$ to compare the behavior of the simple Kaczmarz method and the randomized Kaczmarz method. Figure 3 shows the results of this trial. We see that the simple Kaczmarz method scarcely reduces the error at all, while the block method achieves a healthy rate of convergence. The paper [NW12] provides an analysis of this example in the case where each block contains two rows, but we do not yet have a complete explanation for the performance of Algorithm 1.1 when the blocks are larger.

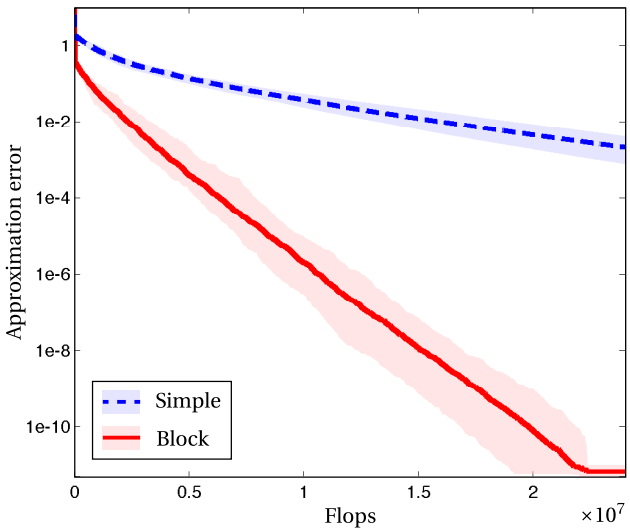

Figure 3 (Convergence with a Coherent Matrix)**.** The matrix $\boldsymbol{A}_{\text{coh}}$ is a fixed 300×100 matrix with strongly correlated rows that are partitioned into 10 blocks of equal size. We compare the approximation error $\|\boldsymbol{x}_j - \boldsymbol{x}_\star\|_2$ for the iterates generated by each algorithm as a function of the number of flops performed. The heavy line denotes the median error over 100 trials; the minimum and maximum error describe the boundaries of the shaded region. See Section 4.3.2 for details.

5. Related Work and Future Directions

We conclude with a short discussion about recent research on randomized Kaczmarz methods and block variants of the simple Kaczmarz method. Finally, we offer some musings about opportunities for further research.

5.1. **Kaczmarz Methods with Randomized Control.** The Kaczmarz method was originally introduced in the paper [Kac37]. It was reinvented by researchers in tomography [GBH70] under the appellation "algebraic reconstruction technique" (ART). See Byrne's book [Byr08] for a contemporary summary of this literature.

The classical variants of the Kaczmarz method rely on deterministic mechanisms for selecting a row at each iteration. Indeed, the simplest version just cycles through the rows in order. It has long been known that the cyclic control scheme performs badly when the rows are arranged in an unhappy order [HS78].

The literature contains empirical evidence [HM93] that *randomized* control mechanisms may be more effective, but until recently there was no compelling theoretical analysis to support this observation.

The paper [SV09b] of Strohmer and Vershynin is significant because it provides the first explicit convergence proof for a randomized variant of the Kaczmarz algorithm. This work establishes that a randomized control scheme leads to an expected linear convergence rate, which can be written in terms of geometric properties of the matrix. In contrast, deterministic convergence analyses that appear in the literature often lead to expressions, e.g., [XZ02, Eqn. (1.2)], whose geometric meaning is not evident.

In the wake of Strohmer and Vershynin's work [SV09b], several other researchers have written about randomized versions of the Kaczmarz scheme and related topics. In particular, Needell demonstrates that the randomized Kaczmarz method converges, even when the linear system is inconsistent [Nee10]. Zouzias and Freris [ZF12] exhibit a randomized procedure, based on ideas from [Pop98], that can reduce the size of the residual $\boldsymbol{e}$. Leventhal and Lewis [LL10] provide an analysis of a randomized iteration for solving least-squares problems with polyhedral constraints, while Richtárik and Takáč have extended these ideas to more general optimization problems [RT11]. Some other references include [EN11, RT11, CP12, NW12].

5.2. **Block Kaczmarz Methods.** The block Kaczmarz update rule (1.5) we are studying is originally due to Elfving [Elf80, Eqn. (2.2)]. This update is a special case of a general framework due to Eggermont et al. [EHL81]. Byrne describes a number of other block Kaczmarz methods in his book [Byr08, Chap. 9].

By now, there is an extensive literature on the convergence behavior of block projection methods, a class of algorithms that includes block Kaczmarz schemes. In particular, we call out the work of Xu and Zikatanov [XZ02], which contains a refined convergence analysis that applies to a wide range of algorithms. Nevertheless, to our knowledge, Algorithm 1.1 is the only block Kaczmarz method that offers an (expected) linear rate of convergence that depends explicitly on geometric properties of the system matrix $\boldsymbol{A}$ and its submatrices $\boldsymbol{A}_\tau$.

Most of the literature on block Kaczmarz methods assumes that a partition T of the rows of the matrix is provided as part of the problem data. We are aware of some research on the prospects for partitioning a matrix in a manner that is favorable for block Kaczmarz methods. In particular, Popa [Pop99] has introduced an algorithm for partitioning a sparse matrix so that each block contains mutually orthogonal rows. Popa has pursued this idea in a sequence of papers, including [Pop01, Pop04]. We believe that our work is the first to recognize the natural connection between the paving literature and the block Kaczmarz method.

5.3. **Some Future Directions.** There are a number of interesting open questions connected with Algorithm 1.1. First, empirical experiments make it clear that a control strategy based on sampling *without* replacement is far more effective than our strategy based on sampling *with* replacement. At present, there is no compelling explanation for this phenomenon (but see [RR12]). Second, we think that other types of block Kaczmarz update rules [Byr08, Chap. 9] would also submit to a convergence analysis similar to the proof of Theorem 1.2. Third, it might be worthwhile to extend the argument of Zouzias and Freris [ZF12] to develop a method with a smaller convergence horizon.

Block algorithms for numerical linear algebra are almost as old as the field itself. Gauss himself suggested a block version of his algorithm for solving linear systems [Gau95, Ben09]. Many other algorithms for numerical linear algebra [GVL96] and least-squares problems [Bjö96] admit natural block variants. The field of optimization also contains a wide variety of block methods, such as [AC89, Tse93].

We believe that row pavings can also play a role in the development and analysis of other types of block algorithms. For instance, it is straightforward to develop a block version of the randomized Gauss–Seidel algorithm introduced in [LL10]. Using the techniques in this paper, we can easily bound the rate

of convergence for this algorithm in terms of the properties of a *column paving* of the system matrix. It would be interesting to pursue this example and others.

Appendix A. The Fast Incoherence Transform

The goal of this Appendix is to prove Proposition 3.7, which states that the fast incoherence transform makes a standardized matrix incoherent with high probability. To that end, suppose that $\boldsymbol{A}$ is a matrix with n unit-norm rows, denoted $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$. Recall that the fast incoherence transform is the matrix $\boldsymbol{S} = \mathbf{F}\boldsymbol{E}$, where $\mathbf{F}$ is the $n \times n$ unitary DFT and $\boldsymbol{E}$ is a diagonal matrix whose entries $\xi_1, \dots, \xi_n$ are independent Rademacher random variables.

Consider the matrix $\boldsymbol{W} := \boldsymbol{S}\boldsymbol{A}$, and introduce the Gram matrix of its rows $\boldsymbol{G} := \boldsymbol{W}\boldsymbol{W}^* = \boldsymbol{S}\boldsymbol{A}\boldsymbol{A}^*\boldsymbol{S}^*$. Expanding this product, we find that the (i, ℓ) entry of $\boldsymbol{G}$ takes the form

$$g_{i\ell} = \sum\nolimits_{jk} \xi_j \xi_k \cdot \mathrm{f}_{ij} \bar{\mathrm{f}}_{\ell k} \cdot \langle \boldsymbol{a}_j, \boldsymbol{a}_k \rangle,$$

where we write f_{ij} for the (i, j) entry of the DFT matrix. (We use the convention that the inner product is antilinear in the second coordinate.) This expression allows us to calculate the expectation of the Gram matrix with ease. Since the rows $\mathbf{f}_1, \dots, \mathbf{f}_n$ of the DFT form an orthonormal family,

$$\mathbb{E} g_{i\ell} = \sum\nolimits_j \mathrm{f}_{ij} \bar{\mathrm{f}}_{\ell k} \|\boldsymbol{a}_j\|_2^2 = \langle \mathbf{f}_i, \mathbf{f}_\ell \rangle = \delta_{i\ell},$$

where $\delta_{i\ell}$ is the Kronecker delta. Note that we have invoked the standardization assumption here.

The real content of the argument is to obtain a bound on *how much* the entries of the Gram matrix deviate from their expected values. Fix a pair (i, ℓ) of indices, not necessarily distinct, and define the random variable

$$Y := \left| g_{i\ell} - \delta_{i\ell} \right|.$$

We can rewrite Y is a more symmetric manner:

$$Y = \left| \sum\nolimits_{j \neq k} \xi_j \xi_k \cdot y_{jk} \right| \quad \text{where} \quad y_{jk} := \frac{1}{2} \left(\mathrm{f}_{ij} \bar{\mathrm{f}}_{\ell k} + \bar{\mathrm{f}}_{ik} \mathrm{f}_{\ell j} \right) \langle \boldsymbol{a}_j, \boldsymbol{a}_k \rangle. \tag{A.1}$$

We see that the random variable Y is a symmetric, homogeneous, second-order Rademacher chaos. We can bound the probability that Y is large by invoking a result of Hanson and Wright [HW71]; see [FR12, Chap. 8] for a modern proof.

Proposition A.1 (Hanson–Wright)**.** *Consider the chaos variable Y defined in* (A.1)*. Let $\boldsymbol{Y}$ be the Hermitian matrix whose (j, k) entry is y_{jk} and whose diagonal entries are zero. Then*

$$\mathbb{P}\{Y \geq t\} \leq 2 \exp\left\{ -\mathrm{c}_{\mathrm{hw}} \cdot \min\left\{ \frac{t}{\|\boldsymbol{Y}\|}, \frac{t^2}{\|\boldsymbol{Y}\|_{\mathrm{F}}^2} \right\} \right\} \quad \textit{for } t \geq 0.$$

The number c_{hw} is a positive, universal constant.

To apply this result, we need to obtain bounds for the Frobenius norm and spectral norm of the matrix $\boldsymbol{Y}$. Note that

$$\boldsymbol{Y} = \frac{1}{2}(\boldsymbol{Z} + \boldsymbol{Z}^*) \quad \text{where} \quad \boldsymbol{Z} := \mathrm{diag}(\mathbf{f}_i) \cdot (\boldsymbol{A}\boldsymbol{A}^* - \mathbf{I}) \cdot \mathrm{diag}(\mathbf{f}_\ell)^*,$$

and $\mathrm{diag}(\cdot)$ converts a vector into a diagonal matrix. We may now calculate the required norms of $\boldsymbol{Y}$. First,

$$\|\boldsymbol{Y}\| \leq \|\boldsymbol{Z}\| \leq \|\mathrm{diag}(\mathbf{f}_i)\| \cdot \|\boldsymbol{A}\boldsymbol{A}^* - \mathbf{I}\| \cdot \|\mathrm{diag}(\mathbf{f}_\ell)\| = \frac{1}{n} \|\boldsymbol{A}\boldsymbol{A}^* - \mathbf{I}\| \leq \frac{1}{n} \|\boldsymbol{A}\|^2. \tag{A.2}$$

The first inequality depends on the convexity of the spectral norm and its invariance under the conjugate transpose. The third relation holds because the entries of the vectors $\mathbf{f}_i$ and $\mathbf{f}_\ell$ all have magnitude $n^{-1/2}$. The last inequality follows because $\|\boldsymbol{A}\boldsymbol{A}^* - \mathbf{I}\| \leq \max\{\|\boldsymbol{A}\|^2 - 1, 1\} \leq \|\boldsymbol{A}\|^2$. For similar reasons,

$$\|\boldsymbol{Y}\|_{\mathrm{F}}^2 \leq \|\boldsymbol{Z}\|_{\mathrm{F}}^2 = \frac{1}{n^2} \|\boldsymbol{A}\boldsymbol{A}^* - \mathbf{I}\|_{\mathrm{F}}^2 < \frac{1}{n^2} \|\boldsymbol{A}\boldsymbol{A}^*\|_{\mathrm{F}}^2 \leq \frac{1}{n} \|\boldsymbol{A}\|^2. \tag{A.3}$$

The strict inequality holds because $\boldsymbol{A}\boldsymbol{A}^*$ has a unit diagonal, which the identity matrix cancels off. The final bound follows from the interpolation $\|\boldsymbol{P}\|_{\mathrm{F}}^2 \le \mathrm{trace}(\boldsymbol{P})\|\boldsymbol{P}\|$, valid when $\boldsymbol{P}$ is positive semidefinite.

Next, let us instate the hypothesis (3.1) from Proposition 3.7:

$$\|\boldsymbol{A}\|^2 \le \frac{\mathrm{c}_{\mathrm{fit}} \cdot n}{\log^3(1+n)}.$$

Introduce this assumption into the bounds (A.2) and (A.3), and apply the Hanson–Wright inequality, Proposition A.1, to reach

$$\mathbb{P}\{Y \ge t\} \le 2\exp\left\{-\frac{\mathrm{c}_{\mathrm{hw}}}{\mathrm{c}_{\mathrm{fit}}} \cdot \log^3(1+n) \cdot \min\{t, t^2\}\right\}.$$

We may set the constant $\mathrm{c}_{\mathrm{fit}} := \mathrm{c}_{\mathrm{hw}}\mathrm{c}_{\mathrm{inc}}{}^2/3$. For the choice $t = \mathrm{c}_{\mathrm{inc}}/\log(1+n)$, we discover that

$$\mathbb{P}\left\{Y \ge \frac{\mathrm{c}_{\mathrm{inc}}}{\log(1+n)}\right\} \le 2(1+n)^{-3}.$$

In other words, it is unlikely that any single pair of rows from $\boldsymbol{W} = \boldsymbol{S}\boldsymbol{A}$ has an inner product much different from its expectation.

To complete the argument, we unfix the pair (i, ℓ) of indices. Forming a union bound over all $n(n+1)/2$ choices where $i \le \ell$, we conclude that

$$\mathbb{P}\left\{\max_{i,\ell} \left|g_{i\ell} - \delta_{i\ell}\right| \ge \frac{\mathrm{c}_{\mathrm{inc}}}{\log(1+n)}\right\} \le (1+n)^{-1}.$$

Therefore, with high probability $\boldsymbol{W}$ is an incoherent matrix that is nearly standardized.

ACKNOWLEDGMENTS

We would like to thank Michael Mahoney, Ben Recht, Thomas Strohmer, Steve Wright for helpful discussions about randomized linear algebra and numerical experiments. Roman Vershynin provided insight on the random paving literature. Michael McCoy explained advanced plotting techniques in MATLAB, and Margot Stokol shared her expertise on color theory. JAT was supported in part by ONR awards N00014-08-1-0883 and N00014-11-1002, AFOSR award FA9550-09-1-0643, DARPA award N66001-08-1-2065, and a Sloan Research Fellowship.

REFERENCES

[AC89] R. Aharoni and Y. Censor. Block-iterative projection methods for parallel computation of solutions to convex feasibility problems. In *Proceedings of the Fourth Haifa Matrix Theory Conference (Haifa, 1988)*, volume 120, pages 165–175, 1989.

[AC09] N. Ailon and B. Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009.

[AMT10] H. Avron, P. Maymounkov, and S. Toledo. Blendenpik: supercharging Lapack's least-squares solver. *SIAM J. Sci. Comput.*, 32(3):1217–1236, 2010.

[And79] J. Anderson. Extensions, restrictions, and representations of states on C^*-algebras. *Trans. Amer. Math. Soc.*, 249(2):303–329, 1979.

[Ben09] M. Benzi. Key moments in the history of numerical analysis. Presented at 2009 SIAM Applied Linear Algebra Conference, Oct. 2009.

[BG12] C. Boutsidis and A. Gittens. Improved matrix algorithms via the Subsampled Randomized Hadamard Transform. Available at `arXiv:1204.0062`, Apr. 2012.

[BHKW88] K. Berman, H. Halpern, V. Kaftal, and G. Weiss. Matrix norm inequalities and the relative Dixmier property. *Integral Equations Operator Theory*, 11(1):28–48, 1988.

[Bjö96] Å. Björck. *Numerical methods for least squares problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[BT87] J. Bourgain and L. Tzafriri. Invertibility of "large" submatrices with applications to the geometry of Banach spaces and harmonic analysis. *Israel J. Math.*, 57(2):137–224, 1987.

[BT91] J. Bourgain and L. Tzafriri. On a problem of Kadison and Singer. *J. Reine Angew. Math.*, 420:1–43, 1991.

[BY93] Z. D. Bai and Y. Q. Yin. Limit of the smallest eigenvalue of a large-dimensional sample covariance matrix. *Ann. Probab.*, 21(3):1275–1294, 1993.

[Byr08] C. L. Byrne. *Applied iterative methods*. A K Peters Ltd., Wellesley, MA, 2008.

[CD12] S. Chrétien and S. Darses. Invertibility of random submatrices via tail decoupling and a matrix Chernoff inequality. *Statist. Probab. Lett.*, 82(7):1479–1487, 2012.

[CFM+92] C. Cenker, H. G. Feichtinger, M. Mayer, H. Steier, and T. Strohmer. New variants of the POCS method using affine subspaces of finite codimension, with applications to irregular sampling. In *Proc. SPIE: Visual Communications and Image Processing*, pages 299–310, 1992.

[CHJ09] Y. Censor, G. T. Herman, and M. Jiang. A note on the behavior of the randomized Kaczmarz algorithm of Strohmer and Vershynin [mr2500924]. *J. Fourier Anal. Appl.*, 15(4):431–436, 2009.

[CP12] X. Chen and A. Powell. Almost sure convergence of the Kaczmarz algorithm with random measurements. *J. Fourier Anal. Appl.*, pages 1–20, 2012. 10.1007/s00041-012-9237-2.

[CT06] P. G. Casazza and J. C. Tremain. The Kadison-Singer problem in mathematics and engineering. *Proc. Natl. Acad. Sci. USA*, 103(7):2032–2039 (electronic), 2006.

[DH01] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inform. Theory*, 47(7):2845–2862, 2001.

[EHL81] P. P. B. Eggermont, G. T. Herman, and A. Lent. Iterative algorithms for large partitioned linear systems, with applications to image reconstruction. *Linear Algebra Appl.*, 40:37–67, 1981.

[Elf80] T. Elfving. Block-iterative methods for consistent and inconsistent linear equations. *Numer. Math.*, 35(1):1–12, 1980.

[EN11] Y. C. Eldar and D. Needell. Acceleration of randomized Kaczmarz method via the Johnson-Lindenstrauss lemma. *Numer. Algorithms*, 58(2):163–177, 2011.

[FR12] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Springer, 2012. To appear.

[FS95] H. G. Feichtinger and T. Strohmer. A Kaczmarz-based approach to nonperiodic sampling on unions of rectangular lattices. In *SampTA '95: 1995 Workshop on Sampling Theory and Applications*, pages 32–37, Jurmala, Latvia, Sep. 1995.

[Gau95] C. F. Gauss. *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae, Supplementum*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), 1995. Transl. G.W. Stewart.

[GBH70] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theoret. Biol.*, 29:471–481, 1970.

[GVL96] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[HM93] G. Herman and L. Meyer. Algebraic reconstruction techniques can be made computationally efficient. *IEEE Trans. Medical Imaging*, 12(3):600–609, 1993.

[HMT11] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

[HS78] C. Hamaker and D. C. Solmon. The angles between the null spaces of X-rays. *J. Math. Anal. Appl.*, 62(1):1–23, 1978.

[HW71] D. L. Hanson and F. T. Wright. A bound on tail probabilities for quadratic forms in independent random variables. *Ann. Math. Statist.*, 42:1079–1083, 1971.

[Kac37] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bull. Int. Acad. Polon. Sci. Lett. Ser. A*, pages 335–357, 1937.

[KS59] R. V. Kadison and I. M. Singer. Extensions of pure states. *Amer. J. Math.*, 81:383–400, 1959.

[KT94] B. Kashin and L. Tzafriri. Some remarks on coordinate restriction of operators to coordinate subspaces. Insitute of Mathematics Preprint 12, Hebrew University, Jerusalem, 1993–1994.

[Lib09] E. Liberty. *Accelerated Dense Random Projections*. Phd dissertation, Yale University, New Haven, CT, 2009.

[LL10] D. Leventhal and A. S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Math. Oper. Res.*, 35(3):641–654, 2010.

[Nao11] A. Naor. Sparse quadratic forms and their geometric applications. Technical Report No. 1033, Séminaire Bourbaki, Jan. 2011.

[Nat01] F. Natterer. *The mathematics of computerized tomography*, volume 32 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. Reprint of the 1986 original.

[Nee10] D. Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT*, 50(2):395–403, 2010.

[NW12] D. Needell and R. Ward. Two-subspace projection method for coherent overdetermined linear systems. Available at `arXiv:1204.0277`, Apr. 2012.

[Pop98] C. Popa. Extensions of block-projections methods with relaxation parameters to inconsistent and rank-deficient least-squares problems. *BIT*, 38(1):151–176, 1998.

[Pop99] C. Popa. Block-projections algorithms with blocks containing mutually orthogonal rows and columns. *BIT*, 39(2):323–338, 1999.

[Pop01] C. Popa. A fast Kaczmarz-Kovarik algorithm for consistent least-squares problems. *Korean J. Comput. Appl. Math.*, 8(1):9–26, 2001.

[Pop04] C. Popa. A Kaczmarz-Kovarik algorithm for symmetric ill-conditioned matrices. *An. Ştiinţ. Univ. Ovidius Constanţa Ser. Mat.*, 12(2):135–146, 2004.

[RR12] B. Recht and C. Ré. Beneath the valley of the noncommutative arithmetic–geometric mean inequality: Conjectures, case studies, and consequences. In *Proc. 25th Ann. Conf. Learning Theory*, Edinburgh, June 2012.

[RT11] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Available at `arXiv:1107.2848`, Apr. 2011.

[Sri10] N. Srivastava. *Spectral sparsification and restricted invertibility*. Phd dissertation, Yale University, New Haven, CT, 2010.

[SS87] M. I. Sezan and H. Stark. Incorporation of a priori moment information into signal recovery and synthesis problems. *J. Math. Anal. Appl.*, 122(1):172–186, 1987.

[SS12] D. A. Spielman and N. Srivastava. An elementary proof of the restricted invertibility theorem. *Israel J. Math.*, 190:83–91, 2012.

[SV09a] T. Strohmer and R. Vershynin. Comments on the randomized Kaczmarz method. *J. Fourier Anal. Appl.*, 15(4):437–440, 2009.

[SV09b] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, 15(2):262–278, 2009.

[Tro08a] J. A. Tropp. Norms of random submatrices and sparse approximation. *C. R. Math. Acad. Sci. Paris*, 346(23-24):1271–1274, 2008.

[Tro08b] J. A. Tropp. The random paving property for uniformly bounded matrices. *Studia Math.*, 185(1):67–82, 2008.

[Tro09] J. A. Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 978–986, Philadelphia, PA, 2009. SIAM.

[Tro11] J. A. Tropp. Improved analysis of the subsampled randomized Hadamard transform. *Adv. Adapt. Data Anal.*, 3(1-2):115–126, 2011.

[Tro12] J. A. Tropp. User-Friendly Tail Bounds for Sums of Random Matrices. *Found. Comput. Math.*, 12(4):389–434, 2012.

[Tse93] P. Tseng. Dual coordinate ascent methods for non-strictly convex minimization. *Math. Programming*, 59(2, Ser. A):231–247, 1993.

[Ver01] R. Vershynin. John's decompositions: selecting a large part. *Israel J. Math.*, 122:253–277, 2001.

[Ver06] R. Vershynin. Random sets of isomorphism of linear operators on Hilbert space. In *High dimensional probability*, volume 51 of *IMS Lecture Notes Monogr. Ser.*, pages 148–154. Inst. Math. Statist., Beachwood, OH, 2006.

[WLRT08] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Appl. Comput. Harmon. Anal.*, 25(3):335–366, 2008.

[XZ02] J. Xu and L. Zikatanov. The method of alternating projections and the method of subspace corrections in Hilbert space. *J. Amer. Math. Soc.*, 15(3):573–597, 2002.

[You12a] P. Youssef. A note on column subset selection. Available at `arXiv:1212.0976`, Dec. 2012.

[You12b] P. Youssef. Restricted invertibility and the Banach–Mazur distance to the cube. Available at `arXiv:1206.0654`, June 2012.

[ZF12] A. Zouzias and N. M. Freris. Randomized extended Kaczmarz for solving least-squares. Available at `arXiv:1205.5770`, May 2012.