SIAM J. MATRIX ANAL. APPL.
Vol. 40, No. 4, pp. 1425–1452

FASTER RANDOMIZED BLOCK KACZMARZ ALGORITHMS*

ION NECOARA†

Abstract. The Kaczmarz algorithm is a simple iterative scheme for solving consistent linear systems. At each step, the method projects the current iterate onto the solution space of a single constraint. Hence, it requires low cost per iteration and storage, and it has a linear rate of convergence. Distributed implementations of Kaczmarz have recently become the de facto architectural choice for large-scale linear systems. Therefore, in this paper we develop a family of randomized block Kaczmarz algorithms that uses at each step a subset of the constraints and extrapolated stepsizes, and can be deployed on distributed computing units. Our approach is based on several new ideas and tools, including stochastic selection rules for the blocks of rows, stochastic conditioning of linear systems, and novel strategies for designing extrapolated stepsizes. We prove that randomized block Kaczmarz algorithms converge linearly in expectation, with a rate depending on the geometric properties of the matrix and its submatrices and on the size of the blocks. Our convergence analysis reveals that the algorithm is most effective when it is given a good sampling of the rows into well-conditioned blocks. Besides providing a general framework for the design and analysis of randomized block Kaczmarz methods, our results resolve an open problem in the literature related to the theoretical understanding of observed practical efficiency of extrapolated block Kaczmarz methods. We also propose an accelerated block Kaczmarz scheme, that is, acceleration in the sense of Chebyshev semi-iterative methods, where the stepsize is chosen based on the roots of Chebyshev polynomials, and we derive convergence rates depending on the square root of the geometric properties of the matrix. Finally, numerical examples illustrate the benefits of the new algorithms.

Key words. consistent linear systems, Kaczmarz algorithm, random blocks of rows, expected linear convergence

AMS subject classifications. 15A06, 90C20, 90C06

DOI. 10.1137/19M1251643

1. Introduction.

Given a real matrix $A \in \mathbb{R}^{m\times n}$ and a real vector $b \in \mathbb{R}^m$, in this paper we search for a solution of the linear system $Ax = b$:

$$\text{Find } x \quad \text{s.t.} \quad Ax = b. \tag{1.1}$$

We assume throughout the paper that the system is consistent; that is, there exists a vector $x^* \in \mathbb{R}^n$ for which $Ax^* = b$, and $a_i \neq 0$ for all $i \in [m](:= \{1, \dots, m\})$. Inconsistent systems are treated, e.g., in [25, 37], while general convex feasibility problems are considered, e.g., in [2, 22, 23]. Let us denote the set of solutions by $\mathcal{X} = \{x \in \mathbb{R}^n : \ Ax = b\}$. Linear systems represent a modeling paradigm for solving many engineering and physics problems, such as partial differential equations [28], sensor networks [36], filtering [13], signal processing [10], computerized tomography [11], and machine learning and optimal control [29]. In these applications it is usually sufficient to find a point which is not too far from the solution set $\mathcal{X}$. In particular, one chooses the error tolerance $\varepsilon > 0$ and aims to find a point x satisfying $\|x - \Pi_{\mathcal{X}}(x)\|^2 \leq \varepsilon$, where $\Pi_{\mathcal{X}}(\cdot) = \arg\min_{y\in\mathcal{X}} \|\cdot - y\|$ is the projection function onto set $\mathcal{X}$, and $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^n$. When a randomized algorithm is used to find x, which

*Received by the editors March 22, 2019; accepted for publication (in revised form) by M. Stoll August 13, 2019; published electronically November 26, 2019.
https://doi.org/10.1137/19M1251643
Funding: The work of the author was supported by the Executive Agency for Higher Education, Research and Innovation Funding (UEFISCDI), Romania, PNIII-P4-PCE-2016-0731, project ScaleFreeNet grant 39/2017.
†Department of Automatic Control and Systems Engineering, University Politehnica Bucharest, Bucharest, 060042, Romania (ion.necoara@acse.pub.ro).

renders x a random vector, one replaces this condition with $\mathbf{E}\left[\|x - \Pi_{\mathcal{X}}(x)\|^2\right] \leq \varepsilon$, where $\mathbf{E}[\cdot]$ denotes the expectation with respect to the randomness of the algorithm.

1.1. Iterative methods. In practice, m and n are usually large so that iterative methods, e.g., the so-called row-action methods, are preferred (in a row-action method only one block of rows of A is used in a certain iteration [3]). One of these methods is the iterative method of Kaczmarz [12, 32, 16]. In some situations, it is even more efficient than the conjugate gradient method, which is the most popular iterative algorithm for solving large linear systems [28]. In fact, the Kaczmarz algorithm was implemented by Hounsfield in the very first medical scanner [11]. At each step, the Kaczmarz algorithm projects the current iterate onto the solution space of a single row $a_{i_k}^T$ and then chooses the next iterate along the line connecting the current iterate and the projection, leading to the following iterative process:

$$x^{k+1} = x^k - \alpha_k \frac{a_{i_k}^T x^k - b_{i_k}}{\|a_{i_k}\|^2} a_{i_k}. \tag{1.2}$$

Usually, the stepsize α_k is chosen in the interval $(0,\ 2)$. For $\alpha_k = 1$ we recover the basic Kaczmarz algorithm [12]. Note that this update rule requires low cost per iteration and storage of order $\mathcal{O}(n)$. In contrast, in *block* Kaczmarz methods a subset of rows A_{J_k} is used at each iteration, with $J_k \subseteq [m]$ and $|J_k| > 1$. We usually distinguish two approaches. The first variant is simply a block generalization of the basic Kaczmarz algorithm; that is, we project the current iterate onto the solution space of the *entire* block A_{J_k} and then choose the next iterate along the line connecting the current iterate and the projection:

$$x^{k+1} = x^k - \alpha_k A_{J_k}^{\dagger}(A_{J_k} x^k - b_{J_k}), \tag{1.3}$$

where $A_{J_k}^{\dagger}$ denotes the pseudoinverse of A_{J_k}. Usually, the stepsize α_k is chosen as 1. This is the approach followed, e.g., in [6, 9, 24, 31, 2], and we refer to this iterative process as the *block projection* Kaczmarz algorithm. The main drawback of (1.3) is that each iteration is expensive, since we need to apply the pseudoinverse to a vector or, equivalently, we must solve a least-squares problem at each iteration, having cost per iteration of order $\mathcal{O}(\tau^2 n)$, where $\tau = |J_k|$. Moreover, (1.3) is not adequate for distributed implementations. The second variant of block Kaczmarz avoids these issues by projecting the current estimate onto *each individual* row that forms the block matrix A_{J_k}, and the resulting projections are averaged to form the next iterate. This leads to the following iteration:

$$x^{k+1} = x^k - \alpha_k \left(\sum_{i \in J_k} \omega_i \frac{a_i^T x^k - b_i}{\|a_i\|^2} a_i \right), \tag{1.4}$$

where the weights $\omega_i \in [0,\ 1]$ such that $\sum_{i \in J_k} \omega_i = 1$, and $\alpha_k \in (0,\ 2)$. Note that update (1.4) is very easy to implement on distributed computing units, and it is comparable in terms of cost per iteration to the basic Kaczmarz update (1.2), i.e., of order $\mathcal{O}(\tau n)$. This is the scheme considered, e.g., in [1, 3, 20, 22, 30, 15, 17], and we also analyze it in this paper and refer to it as the *average block Kaczmarz algorithm*. When $\alpha_k \in (0,\ 2)$ is assumed, the iterative process (1.2) is known to converge linearly [16, 32] (see also section 3.3). Moreover, linear convergence results for the iteration (1.3), with particular stepsize $\alpha_k = 1$, were recently derived in [9, 24, 31]. *However, we are not aware of any convergence rates depending on the size of the blocks $|J_k|$ and the geometric properties of the matrix A and its submatrices A_{J_k} for the iterative process* (1.4).

1.2. Extrapolation. It is well known that the practical performance of block Kaczmarz method (1.4) can be enhanced, and often dramatically so, using *extrapolation*. This refers to the practice of moving *further* along the line connecting the last iterate and the average of the projections by using a stepsize $\alpha_k \geq 2$; see, e.g., [1]. For example, since the iterative process (1.4) can be slow, in [20, 30] an extrapolated variant of (1.4) was introduced with the following *adaptive* choice for the stepsize α_k:

$$\alpha_k \simeq 2\frac{\sum_{i\in J_k}\bar{\omega}_i(a_i^T x^k - b_i)^2}{\|\sum_{i\in J_k}\bar{\omega}_i(a_i^T x^k - b_i)a_i\|^2}, \tag{1.5}$$

where we use the notation $\bar{\omega}_i = \omega_i/\|a_i\|^2$, and for convenience we define $0/0 = 1$. From Jensen's inequality it follows that $\alpha_k \geq 2$. However, in numerical experiments, it has been observed that the extrapolation parameter α_k from (1.5) can be much larger than 2. Moreover, the sequence x^k generated by the iterative process (1.4) using the extrapolated adaptive stepsize α_k from (1.5) usually converges much faster than the same sequence x^k from (1.4) but generated with stepsize $\alpha_k \in (0,\ 2)$ [1, 3, 4, 20, 30]. However, despite more than 80 years of research on block Kaczmarz methods, the empirical success of extrapolation schemes is not supported by theory. *That is, to the best of our knowledge, there is no theory explaining why these methods with* $\alpha_k > 1$ *require fewer iterations than their nonextrapolated variants* $\alpha_k = 1$.

1.3. Row importance. While selecting the index set $J \subseteq [m]$ uniformly random appears as the most natural choice, it is likely the case that some blocks of rows of A are more important than others. As an illustration, consider the simple scenario in which there exists $T \subset [m]$ such that $\mathcal{X} = \{x \in \mathbb{R}^n : A_T x = b_T\}$, where A_T denotes the block matrix of A whose rows are indexed in the set T. Clearly, the rows a_i for $i \in T$ are more important than the rows a_i for $i \notin T$. Moreover, when multiple Ts exist, then the rows of the matrix A_T having the best conditioning are more important than the others. These are extreme scenarios: if such a T is known, one should simply remove the unimportant rows from the representation. However, even if none of the rows can be removed, it is often the case that some (blocks of) rows are more important than others in the sense that one should project on these more often. In fact, the operator theory shows that some sampling strategies of the blocks of rows are more effective than others in terms of conditioning; see, e.g., [24, 34]. *We are not aware of any paper on block Kaczmarz method* (1.4) *that takes importance of blocks of rows into consideration.* An exception to this are some recent works [24, 9, 31], but on the block projection, Kaczmarz algorithm (1.3) (i.e., [24, 9, 31] analyze row importance for the method that projects the current estimate on the entire solution space of $A_J x = b_J$, as opposed to our algorithm (1.4), where we only project on the individual rows of the submatrix A_J and then average).

1.4. Outline. In section 2 we summarize selected key contributions of this paper. In section 3 we present some preliminary results for the Kaczmarz algorithm. In section 4 we define general random average block Kaczmarz algorithms and derive new convergence rates. In section 5 we present an acceleration of the block Kaczmarz algorithm using Chebyshev-based stepsizes and derive the corresponding convergence rates. Finally, in section 6 we corroborate our theoretical results through numerical experiments.

1.5. Notation. For $x \in \mathbb{R}^n$, the standard Euclidean norm is denoted by $\|x\| = \sqrt{x^T x}$. For a positive integer m, let $[m] = \{1, 2, \ldots, m\}$. By e_i we denote the ith column of the identity matrix $I_n \in \mathbb{R}^{n\times n}$. Let $A \in \mathbb{R}^{m\times n}$ be a real matrix. By

$\|A\|_F$, $\|A\|$, rank(A), range(A), and a_i^T we denote its Frobenius norm, spectral norm, rank, range, and ith row, respectively. For an index set $J \subset [m]$, by $A_J \in \mathbb{R}^{|J|\times n}$ we denote the matrix with the rows a_i^T for $i \in J$. If A is a symmetric matrix, then $\lambda_{\min}^{\text{nz}}(A) = \min_{x\in\text{range}(A)\setminus 0} x^TAx/x^Tx$ and $\lambda_{\max}(A) = \max_{x\neq 0} x^TAx/x^Tx$ denote the smallest nonzero eigenvalue and the largest eigenvalue, respectively. The projection of a point x onto a closed convex set X is denoted by $\Pi_X(x) = \arg\min_z\{\|x-z\| : z \in X\}$. A matrix is called *normalized* if all its rows have the Euclidean norm equal to 1.

2. Contributions. In this section we briefly review our key contributions and results, leaving the theoretical details to the rest of the paper.

2.1. General framework. We *develop a unified framework for studying extrapolation and row importance questions for consistent linear systems, together with randomized block Kaczmarz methods for solving such systems of linear equalities.* We define a probability space $([m], \mathcal{F}, \mathbf{P})$, where $\mathcal{F} \subseteq 2^{[m]}$ (power set of $[m]$) is a σ-algebra. By sampling $J \sim \mathbf{P}$, with $J \subseteq [m]$, we are *choosing* a block of rows A_J from the matrix A. In this way we achieve the following two goals at the same time:

(i) First, this sampling defines a general *stochastic selection rule,* which we use to design a *randomized block Kaczmarz method*, described in section 2.2 below.

(ii) Second, the choice of probability measure is a natural way to assign *importance* to the blocks of A.

Note that the probability $\mathbf{P}$ is a *parameter* playing the dual role of controlling the representation of the solution set $\mathcal{X}$ as an intersection of blocks of rows of matrix A, and defining the importance sampling procedure, which in turn defines the algorithm. For matrices with normalized rows (i.e., each row has norm 1), we have identified the following *stochastic conditioning* parameter:

$$\lambda_{\max}^{\text{block}} = \max_{J\sim\mathbf{P}} \lambda_{\max}(A_J^TA_J), \tag{2.1}$$

where the maximum is taken over all possible draws of J according to the probability $\mathbf{P}$, as the key quantity characterizing importance sampling. In particular, our analysis reveals that the most effective importance rule is the one that makes $\lambda_{\max}^{\text{block}}$ small; i.e., there is a sampling of the blocks of the rows into well-conditioned blocks. Moreover, the operator theory literature provides detailed information about the existence and construction of such good sampling (see section 4.3).

2.2. Algorithms. We propose an average block Kaczmarz algorithmic framework that uses a randomized scheme to choose a subset of the constraints at each iteration (see sections 4 and 5):

$$\text{(RaBK)}: \qquad \begin{array}{l}\text{Draw at each step a sample } J_k \sim \mathbf{P} \text{ and update:}\\ x^{k+1} = x^k - \alpha_k \left(\sum_{i\in J_k} \omega_i^k \dfrac{a_i^Tx^k - b_i}{\|a_i\|^2} a_i\right),\end{array}$$

where the weights satisfy $\omega_i^k \in [0,\ 1]$ such that $\sum_{i\in J_k} \omega_i^k = 1$. One important property of our algorithmic framework is the use of several *extrapolated* stepsizes α_k that, in general, are much larger than the stepsize $\alpha_k \in (0,\ 2)$ usually used in the literature [3, 4, 9]. More precisely, we analyze three choices for the stepsize α_k:

(i) one depending on the geometric properties of the submatrices of A of the form $\mathcal{O}(1/\lambda_{\max}^{\text{block}})$,

(ii) an adaptive stepsize similar to (1.5), and

(iii) a stepsize based on the roots of Chebyshev polynomials.

TABLE 1

The key convergence results obtained in this paper for algorithm RaBK for the three choices of the extrapolated stepsize. Here, matrix A is normalized, and $\lambda_{\max}$ and $\lambda_{\min}(\lambda_{\min}^{nz})$ denote the largest and smallest (nonzero) eigenvalues of AA^T, respectively.

RaBK algorithm	Convergence rates	Remarks
constant stepsize normalized A	$\mathbf{E}\left[\|x^k - x_k^*\|^2\right] \le \left(1 - \frac{\tau}{m}\frac{\lambda_{\min}^{\text{nz}}}{\lambda_{\max}^{\text{block}}}\right)^k \|x^0 - x_0^*\|^2$	Theorem 4.1
adaptive stepsize normalized A	$\mathbf{E}\left[\|x^k - x_k^*\|^2\right] \le \left(1 - \frac{\tau}{m}\frac{\lambda_{\min}^{\text{nz}}}{\lambda_{\max}^{\text{block}}}\right)^k \|x^0 - x_0^*\|^2$	Theorem 4.2
Chebyshev stepsize normalized A & $\lambda_{\min} > 0$	$\|\mathbf{E}\left[x^k - x_k^*\right]\|^2 \le \left(1 - \sqrt{\frac{\lambda_{\min}}{\lambda_{\max}}}\right)^{2k} \|x^0 - x_0^*\|^2$	Theorem 5.1

All three extrapolation procedures yield $\alpha_k \ge 2$, and hence they accelerate drastically the convergence of the RaBK algorithm (see also the numerical results of section 6). Another feature of our algorithm is that it allows one to project in *parallel* onto several rows, thus providing flexibility in matching the implementation of the algorithm on the distributed architecture at hand. Moreover, the RaBK algorithm can be interpreted, for some particular choices of the weights and stepsize, as a *minibatch* stochastic gradient descent or *block* coordinate descent method applied to a specific optimization problem.

2.3. Convergence rates. To the best of our knowledge, convergence rates of Kaczmarz type methods were only previously derived for stepsizes belonging to the interval $(0,\ 2)$ [3, 4, 9, 24, 31, 32]. Moreover, the existing convergence estimates for block Kaczmarz algorithm (1.4) do not show any dependence on the size of the blocks $|J|$ or on the geometric properties of the block submatrices A_J [1, 3, 4, 20, 30]. On the other hand, our convergence analysis for the randomized average block Kaczmarz (RaBK) algorithm is one of the first proving an (expected) linear rate of convergence that is expressed explicitly in terms of the geometric properties of the matrix and its submatrices and of the size of the blocks. Moreover, our analysis allows one to derive convergence estimates for all three choices of the extrapolated stepsize. *To the best of our knowledge, this is the first time the randomized block Kaczmarz algorithm with extrapolation ($|J| > 1$ and $\alpha_k > 2$) is shown to have a better convergence rate than its basic variant* (1.2) *($|J| = 1$ and $\alpha_k = 1$); see Table* 1. We have identified $\lambda_{\max}^{\text{block}}$ as the key quantity determining whether extrapolation helps or not, and by how much (the smaller $\lambda_{\max}^{\text{block}}$, the more it helps). For example, for normalized matrices, RaBK with the extrapolation rules (i)–(ii) has an expected linear rate for the square distance of the iterates to the optimal solution set of the form (see Table 1)

$$\mathcal{O}\left(\frac{m\lambda_{\max}^{\text{block}}}{\tau\lambda_{\min}^{\text{nz}}}\log\frac{1}{\varepsilon}\right),$$

where $\lambda_{\min}^{\text{nz}}$ denotes the smallest nonzero eigenvalue of AA^T. Thus, we obtain a convergence rate depending on the geometric properties of the matrix A and its submatrices A_J and on the size of the blocks $\tau = |J|$. When comparing RaBK with the basic Kaczmarz in terms of total computational cost to achieve an ε solution, we get

$$\mathcal{O}\left(\tau n \cdot \frac{m\lambda_{\max}^{\text{block}}}{\tau\lambda_{\min}^{\text{nz}}}\log\frac{1}{\varepsilon}\right) \quad \text{versus} \quad \mathcal{O}\left(n \cdot \frac{m}{\lambda_{\min}^{\text{nz}}}\log\frac{1}{\varepsilon}\right).$$

Therefore, our convergence rate also explains *why* and *when* the randomized block Kaczmarz algorithm with the constant extrapolated stepsize (4.2) or adaptive extrapolated stepsize (1.5) works better compared to its basic counterpart. In particular,

the analysis reveals that a distributed implementation of the extrapolated RaBK algorithm is most effective when the sampling of the blocks of rows yields a partition into well-conditioned blocks; that is, the stochastic conditioning parameter $\lambda_{\max}^{\text{block}}$ is small.

For the third choice of the extrapolated stepsize, depending on the roots of Chebyshev polynomials, and for normalized matrices having $\lambda_{\min} > 0$, we get a linear rate for the expected iterates of the form (see Table 1):

$$\mathcal{O}\left(\sqrt{\tfrac{\lambda_{\max}}{\lambda_{\min}}}\log\tfrac{1}{\varepsilon}\right),$$

where $\lambda_{\min}$ ($\lambda_{\max}$) denotes the smallest (largest) eigenvalue of AA^T. Note that this convergence estimate is the same as that for the conjugate gradient method and is optimal for this class of iterative schemes, as the condition number of the matrix is square rooted. If $\lambda_{\min}=0$, we get a sublinear rate of order $\mathcal{O}(1/\sqrt{\varepsilon})$.

3. Preliminaries. Note that the problem of finding a solution of the linear system $Ax = b$ can be posed as a quadratic optimization problem, the so-called linear least-squares problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2m}\|Ax-b\|^2 \qquad \left(:= \frac{1}{2m}\sum_{i=1}^{m}(a_i^T x - b_i)^2\right). \tag{3.1}$$

A more particular formulation is to find the least-norm solution of the linear system:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x\|^2 \quad \text{s.t.} \quad Ax = b. \tag{3.2}$$

The dual of optimization problem (3.2) also takes the form of a quadratic program:

$$\min_{y\in\mathbb{R}^m} \frac{1}{2}\|A^T y\|^2 - b^T y, \tag{3.3}$$

where the primal variable x and the dual variable y are related through the relation $x = A^T y$. Let us define the primal and dual objective functions $f(x) = (1/2m)\|Ax - b\|^2$ and $g(y) = 1/2\|A^T y\|^2 - b^T y$, respectively. Recall that the set of solutions is denoted $\mathcal{X} = \{x \in \mathbb{R}^n : \ Ax = b\}$, and for any given x we define its projection onto $\mathcal{X}$ by $x^* = \Pi_{\mathcal{X}}(x)$.

3.1. Basic Kaczmarz algorithm. The Kaczmarz algorithm is an iterative scheme for solving the linear system $Ax = b$ that requires only $\mathcal{O}(n)$ cost per iteration and storage and has a linear rate of convergence. At each iteration k, the algorithm selects (cyclically or randomly) a row $i_k \in [m]$ of the linear system and does an orthogonal projection of the current estimate vector x^k onto the corresponding hyperplane $a_{i_k}^T x = b_{i_k}$:

$$\min_x \|x - x^k\|^2 \quad \text{s.t.} \quad a_{i_k}^T x = b_{i_k}.$$

Then, we choose the next iterate along the line connecting the current iterate and the projection. This leads to the iteration shown in Algorithm 3.1 for the randomized/cyclic Kaczmarz algorithm [12, 32].

Usually, α_k is chosen constant in interval $(0,\ 2)$. For $\alpha_k = 1$ we recover the basic Kaczmarz algorithm [12].

Algorithm 3.1 (algorithm Kaczmarz).

choose $x^0 \in \mathbb{R}^n$
for $k \geq 0$ **do**
  choose an index $i_k \in [m]$ (cyclically or randomly) and update:
  $x^{k+1} = x^k - \alpha_k \frac{a_{i_k}^T x^k - b_{i_k}}{\|a_{i_k}\|^2} a_{i_k}$.
end for

3.2. Interpretations. We can view the randomized Kaczmarz algorithm, i.e., when i_k is chosen randomly, as an optimization method for solving a specific primal or dual optimization problem. More precisely, the Kaczmarz algorithm is a particular case of the following.

SGD (stochastic gradient descent). The randomized Kaczmarz (Algorithm 3.1) is equivalent to one step of the SGD method [26] applied to the finite sum problem (3.1). Specifically, a component function i_k, $f_{i_k}(x) = 1/2(a_{i_k}^T x - b_{i_k})^2$, is chosen randomly, and a negative gradient step (having $\nabla f_{i_k}(x) = (a_{i_k}^T x - b_{i_k})a_{i_k}$) of this partial function in x^k with stepsize $\alpha_k/\|a_{i_k}\|^2$ is considered:

$$x^{k+1} = x^k - \frac{\alpha_k}{\|a_{i_k}\|^2} \nabla f_{i_k}(x^k).$$

RCD (random coordinate descent). The randomized Kaczmarz (Algorithm 3.1) is equivalent to one step of the RCD method [27] applied to the dual problem (3.3). Specifically, a negative gradient step in the random i_kth component of y (having the expression $\nabla_{i_k} g(y) = a_{i_k}^T A^T y - b_{i_k}$) with stepsize $\alpha_k/\|a_{i_k}\|^2$ is taken, yielding

$$y^{k+1} = y^k - \frac{\alpha_k}{\|a_{i_k}\|^2} \nabla_{i_k} g(y^k) \cdot e_{i_k},$$

where e_i denotes the ith column of the identity matrix in $\mathbb{R}^{n \times n}$. We easily recover the iteration of Algorithm 3.1 by simply multiplying this update with A^T and using the relation between the primal and dual variables given by $x = A^T y$. Note that in both interpretations, we need to choose a specific stepsize in order to prove convergence; see [26, 27].

3.3. Convergence properties. It is known that Algorithm 3.1 with cyclic selection of rows converges to the minimum norm solution of $Ax = b$ when it is initialized with $x^0 = 0$, but the speed of convergence is not simple to quantify, and it especially depends on the ordering of the rows [5]. The situation changes if one considers a randomization such that in each step one chooses a row of the system matrix at random, according to a probability $\mathbf{P}$. In the seminal paper [32] it was shown that sampling the rows of A with probability $\mathbf{P}(i = i_k) = \frac{\|a_{i_k}\|^2}{\|A\|_F^2}$ for all $i \in [m]$ and using constant stepsize $\alpha = 1$, we get a linear convergence rate in expectation of the form

$$\mathbf{E}\left[\|x^k - x_k^*\|^2\right] \leq \left(1 - \frac{\lambda_{\min}^{\mathrm{nz}}(A^T A)}{\|A\|_F^2}\right)^k \|x^0 - x_0^*\|^2, \tag{3.4}$$

where $\lambda_{\min}^{\mathrm{nz}}(\cdot)$ denotes the minimum nonzero eigenvalue of a given matrix and $x_k^* = \Pi_{\mathcal{X}}(x^k)$. For completeness, let us derive this convergence rate. Considering the stepsize α_k constant in the interval $(0,\ 2)$ and using that $\langle x - x^*, (a_i^T x - b_i)a_i \rangle = (a_i^T x - b_i)^2$

for any x^* a solution of $Ax = b$, we get

$$\|x^{k+1} - x^*\|^2 = \|x^k - x^*\|^2 - 2\alpha \frac{(a_i^T x^k - b_i)^2}{\|a_i\|^2} + \alpha^2 \frac{(a_i^T x^k - b_i)^2}{\|a_i\|^2}$$
$$= \|x^k - x^*\|^2 - \alpha(2-\alpha)\frac{(a_i^T x^k - b_i)^2}{\|a_i\|^2}.$$

Now, taking the conditional expectation under the probability $\mathbf{P}(i = i_k) = \frac{\|a_{i_k}\|^2}{\|A\|_F^2}$, we get

$$\mathbf{E}_i \left[\|x^{k+1} - x^*\|^2 | x^k \right] \leq \|x^k - x^*\|^2 - \frac{\alpha(2-\alpha)}{\|A\|_F^2} \|Ax^k - b\|^2.$$

Further, it is well known from the Courant–Fischer theorem that for any matrix A we have $\|Ax\|^2 \geq \lambda_{\min}^{\text{nz}}(AA^T)\|x\|^2$ for all $x \in \text{range}(A^T)$. Moreover, we have that $x - \Pi_{\mathcal{X}}(x) \in \text{range}(A^T)$ for any x. In conclusion, if we denote $x_k^* = \Pi_{\mathcal{X}}(x^k)$, we get

$$\|Ax^k - b\|^2 = \|A(x^k - x_k^*)\|^2 \geq \lambda_{\min}^{\text{nz}}(AA^T)\|x^k - x_k^*\|^2.$$

Using this inequality in the recurrence above and taking expectation over the entire history, we get the following linear convergence rate in expectation:

$$\mathbf{E}\left[\|x^{k+1} - x_k^*\|^2 \right] \leq \left(1 - \frac{\alpha(2-\alpha)\lambda_{\min}^{\text{nz}}(AA^T)}{\|A\|_F^2} \right) \mathbf{E}\left[\|x^k - x_k^*\|^2 \right]. \tag{3.5}$$

For the optimal choice $\alpha^* = 1$ (i.e., $\alpha^* = \arg\max_\alpha \alpha(2-\alpha)$) we get the simpler convergence estimate (3.4) derived in [32]. Note that for ill-conditioned problems, i.e., $\lambda_{\min}^{\text{nz}}(AA^T)$ small and $\|A\|_F$ large, this linear convergence is very slow, using a constant stepsize $\alpha \in (0,\ 2)$. In the next sections we prove that block variants of randomized Kaczmarz (Algorithm 3.1) with properly chosen extrapolated stepsize α_k larger than 2 can substantially accelerate the convergence rate (3.5). Note that extensions of these results to general convex feasibility problems can be found in [22, 23].

3.4. Preliminary probability results. Let J be a random set-valued map with values in $2^{[m]}$. Any realization $J \subseteq [m]$ of this random variable, referred to as sampling and having the same notation as the random variable, is characterized by the probability distribution $\mathbf{P}(J)$. We also define the probability with which an index $i \in [m]$ can be found in J as

$$p_i = \mathbf{P}(i \in J).$$

Then, for any scalars θ_i, with $i \in [m]$, the following relation holds in expectation:

$$\mathbf{E}_J \left[\sum_{i \in J} \theta_i \right] = \sum_{J \subseteq [m]} \left(\sum_{i \in J} \theta_i \right) \mathbf{P}(J) = \sum_{i \in [m]} \theta_i \left(\sum_{J : i \in J} \mathbf{P}(J) \right) = \sum_{i \in [m]} p_i \theta_i. \tag{3.6}$$

The following examples for sampling blocks of rows of $A \in \mathbb{R}^{m \times n}$ will be used in our subsequent analysis.

Uniform sampling. One natural choice is the *uniform* sampling of τ unique indexes of rows that make up J, i.e., $|J| = \tau$ for all samplings, with $1 \leq \tau \leq m$ fixed. For this choice of the random variable J, we observe that we have a total number

Algorithm 4.1 (algorithm RaBK).

choose $x^0 \in \mathbb{R}^n$, stepsize sequence $(\alpha_k)_{k\geq 0}$, and weights sequence $(\omega_k)_{k\geq 0}$
for $k \geq 0$ **do**
  draw sample $J_k \sim \mathbf{P}$ and update:
  $x^{k+1} = x^k - \alpha_k \left(\sum_{i \in J_k} \omega_k^i \frac{a_i^T x^k - b_i}{\|a_i\|^2} a_i \right)$.
end for

of $\binom{m}{\tau}$ possible values that J can take. Thus, for the uniform sampling we have $\mathbf{P}(J) = 1/\binom{m}{\tau}$. We can also express p_i for the uniform sampling as

$$p_i = \mathbf{P}(i \in J) = \sum_{J: i \in J} \mathbf{P}(J) = \frac{\binom{m-1}{\tau-1}}{\binom{m}{\tau}} = \frac{\tau}{m}. \tag{3.7}$$

Partition sampling. Another choice is the *partition* sampling; i.e., consider a partition of $[m]$ given by $\{J_1, \ldots, J_\ell\}$, and then take $\mathbf{P}(J) = 1/\ell$ or $\mathbf{P}(J) = \|A_J\|_F^2/\|A\|_F^2$ for all $J \in \{J_1, \ldots, J_\ell\}$. For example, for the first probability choice of the partition sampling, p_i is given by

$$p_i = \frac{1}{\ell}. \tag{3.8}$$

In particular, if all the subsets in the partition have the same cardinality, i.e., $|J_l| = \tau$ for all $l \in [\ell]$, and A is normalized, then the two probabilities are the same, and $\ell = m/\tau$. Hence, $p_i = \frac{\tau}{m}$. These preliminary results will help us in the convergence analysis of the randomized block Kaczmarz algorithms we propose next.

4. Randomized average block Kaczmarz algorithms. In this section we design new variants of randomized Kaczmarz (Algorithm 3.1), considering at each step a block of rows of the linear system $Ax = b$, average projections to each row of the block, and different choices for the stepsize. For all these methods, which we refer to as *randomized average block Kaczmarz algorithms*, we prove expected linear convergence rates. Note that average block Kaczmarz methods have also been considered in other works; see, e.g., [1, 3, 20, 30] and the references therein. *Nevertheless, to the best of our knowledge, this paper is the first to provide an expected linear rate of convergence that depends explicitly on geometric properties of the system matrix A and its submatrices A_J.* Moreover, the convergence estimates hold for several extrapolated stepsizes. In our RaBK algorithm, at each iteration, instead of projecting onto only one hyperplane, we consider projections onto several hyperplanes and then take as a new direction a convex combination of these projections with some stepsize (see Algorithm 4.1). Here $J_k = \{i_k^1, \ldots, i_k^{\tau_k}\} \subseteq [m]$ is the set of indexes corresponding to the rows selected at iteration k of size $\tau_k \in [1, m]$, and $\mathbf{P}$ denotes the probability distribution over the collection of subsets of indexes of $[m]$. In the rest of the paper we assume that the probability $\mathbf{P}$ is chosen such that the probability with which any index $i \in [m]$ is in J satisfies $p_i > 0$. Moreover, the weights $\omega_k = (\omega_k^i)_{i \in J_k}$ are chosen positive and summing to 1. Thus, in our analysis we assume bounded weights satisfying $0 < \omega_{\min} \leq \omega_k^i \leq \omega_{\max} < 1$ for all $i \in J_k$ and $k \geq 0$. Two simple choices for the weights are, e.g., $\omega_k^i = \|a_i\|^2 / \sum_{i \in J_k} \|a_i\|^2$ and $\omega_k^i = 1/\tau_k$ for all $k \geq 0$. In these two

particular cases we get the following compact updates:

$$x^{k+1} = x^k - \alpha_k \frac{A_{J_k}^T (A_{J_k} x^k - b_{J_k})}{\|A_{J_k}\|_F^2} \quad \text{and} \quad x^{k+1} = x^k - \alpha_k \frac{A_{J_k}^T D_{J_k} (A_{J_k} x^k - b_{J_k})}{\tau_k},$$

respectively, where the diagonal matrix $D_J = \text{diag}(1/\|a_i\|^2,\ i \in J) \in \mathbb{R}^{\tau \times \tau}$. Several choices for the stepsize will be given in the next sections, based on overrelaxations (extrapolations), i.e., $\alpha_k > 2$. Similarly, as for the Kaczmarz algorithm, RaBK (Algorithm 4.1) can be interpreted as the following.

BSGD (batch stochastic gradient descent). One iteration of the RaBK algorithm can be viewed as one step of the BSGD [26] applied to the finite sum problem (3.1) when the weights ω_k are chosen in a particular fashion. Specifically, if we choose the particular weights $\omega_k^i = \|a_i\|^2 / \sum_{i \in J_k} \|a_i\|^2$ and uniform probability, then we recover the BSGD with a certain choice of the stepsize:

$$x^{k+1} = x^k - \frac{\tau_k \alpha_k}{\sum_{i \in J_k} \|a_i\|^2} \left(\frac{1}{\tau_k} \sum_{i \in J_k} (a_i^T x^k - b_i) a_i \right).$$

RBCD (randomized block coordinate descent). One iteration of the RaBK algorithm can be viewed as one step of the RBCD [21, 27] applied to the dual problem (3.3) when the weights ω_k are chosen in a particular fashion. Specifically, if we choose the particular weights $\omega_k^i = \|a_i\|^2 / \sum_{i \in J_k} \|a_i\|^2$, then we recover the block coordinate descent method with a certain choice of the stepsize:

$$x^{k+1} = x^k - \frac{\alpha_k}{\sum_{i \in J_k} \|a_i\|^2} \left(\sum_{i \in J_k} (a_i^T x^k - b_i) a_i \right).$$

However, for general weights ω_k and stepsize α_k, the RaBK algorithm cannot be interpreted in these ways, and thus our scheme is more general. In the following, we denote $x_k^* = \Pi_{\mathcal{X}}(x^k)$, that is, the projection of x^k onto the solution set $\mathcal{X}$ of the linear system $Ax = b$.

4.1. Randomized average block Kaczmarz algorithm with constant stepsize. In this section we investigate the convergence rate of the RaBK algorithm for constant extrapolated stepsize $\alpha_k = \alpha > 2$ and weights $\omega_k^i = \omega_i$ for all k. Thus, the iteration of RaBK (Algorithm 4.1) becomes, in this case,

$$x^{k+1} = x^k - \alpha \left(\sum_{i \in J_k} \omega_i \frac{a_i^T x^k - b_i}{\|a_i\|^2} a_i \right). \tag{4.1}$$

The weights are chosen to satisfy $0 < \omega_{\min} \leq \omega_i \leq \omega_{\max} < 1$ for all i and sum to 1. Let us also define the following stochastic conditioning parameter depending on the geometric properties of the submatrices A_J:

$$\lambda_{\max}^{\text{block}} = \max_{J \sim \mathbf{P}} \lambda_{\max} \left(A_J^T \text{diag} \left(\frac{1}{\|a_i\|^2}, i \in J \right) A_J \right).$$

Then, we consider an extrapolated constant stepsize of the form

$$0 < \alpha < \frac{2\omega_{\min}}{\omega_{\max}^2 \lambda_{\max}^{\text{block}}}. \tag{4.2}$$

From basic linear algebra we have that the sum of the eigenvalues of a symmetric positive semidefinite matrix is equal to its trace; consequently, if the rank of the matrix is at least 2, then its maximum eigenvalue is strictly less than the trace. Therefore, when we choose a random variable such that all the samplings satisfy $|J| = \tau$, with $\tau \in [1, m]$, then it follows that $\lambda_{\max}^{\text{block}} < \tau$ provided that $\text{rank}(A_J) \geq 2$ for all $J \sim \mathbf{P}$. Hence, in this case we use an overrelaxed (extrapolated) stepsize, since usually $2\omega_{\min}/\omega_{\max}^2 \lambda_{\max}^{\text{block}} > 2$. For example, for $\omega_i = 1/\tau$, we get $2\tau/\lambda_{\max}^{\text{block}} > 2$. Using (3.6) we also define the positive semidefinite matrix W as

$$W = \mathbf{E}_J \left[\sum_{i \in J} \frac{a_i a_i^T}{\|a_i\|^2} \right] = \sum_{i \in [m]} p_i \frac{a_i a_i^T}{\|a_i\|^2} = A^T \text{diag}\left(\frac{p_i}{\|a_i\|^2}, i \in [m] \right) A.$$

To the best of our best knowledge, the choice (4.2) for the stepsize in the block Kaczmarz algorithm seems to be new. The next theorem proves the convergence rate of this algorithm, which depends explicitly on the geometric properties of the system matrix A and its submatrices A_J.

THEOREM 4.1. *Let $\{x^k\}_{k \geq 0}$ be generated by RaBK (Algorithm 4.1) with the particular update (4.1); i.e., the weights satisfy $0 < \omega_{\min} \leq \omega_i \leq \omega_{\max} < 1$ for all $i \in [m]$ and the stepsize $\alpha = \frac{(2-\delta)\omega_{\min}}{\omega_{\max}^2 \lambda_{\max}^{block}}$ for some $\delta \in (0, 1]$. Then, we have the following linear convergence rate in expectation:*

$$\mathbf{E}\left[\|x^k - x_k^*\|^2 \right] \leq \left(1 - \frac{(2-\delta)\omega_{\min}^2 \lambda_{\min}^{nz}(W)}{\omega_{\max}^2 \lambda_{\max}^{block}} \right)^k \|x^0 - x_0^*\|^2. \tag{4.3}$$

Proof. Since we assume a consistent linear system, that is, there is x^* such that $Ax^* = b$, we have

$$\begin{aligned}
\|x^{k+1} - x^*\|^2 &= \left\| x^k - x^* - \alpha \left(\sum_{i \in J_k} \omega_i \frac{a_i^T x^k - b_i}{\|a_i\|^2} a_i \right) \right\|^2 \\
&= \left\| x^k - x^* - \alpha \left(\sum_{i \in J_k} \omega_i \frac{a_i a_i^T}{\|a_i\|^2} (x^k - x^*) \right) \right\|^2 \\
&= \left\| \left(I_n - \alpha \left(\sum_{i \in J_k} \omega_i \frac{a_i a_i^T}{\|a_i\|^2} \right) \right) (x^k - x^*) \right\|^2 \\
&= (x^k - x^*)^T \left(I_n - 2\alpha \sum_{i \in J_k} \omega_i \frac{a_i a_i^T}{\|a_i\|^2} + \alpha^2 \left(\sum_{i \in J_k} \omega_i \frac{a_i a_i^T}{\|a_i\|^2} \right)^2 \right) (x^k - x^*).
\end{aligned}$$

We need to take conditional expectation over J_k. However, for a general random sampling J we have from (3.6) that the expectation over the first sum from above yields the lower bound

$$\begin{aligned}
\mathbf{E}_J \left[\sum_{i \in J} \omega_i \frac{a_i a_i^T}{\|a_i\|^2} \right] &\succeq (\min_{i \in J} \omega_i) \mathbf{E}_J \left[\sum_{i \in J} \frac{a_i a_i^T}{\|a_i\|^2} \right] = \omega_{\min} \sum_{i \in [m]} p_i \frac{a_i a_i^T}{\|a_i\|^2} \\
&= \omega_{\min} A^T \text{diag}\left(\frac{p_i}{\|a_i\|^2}, i \in [m] \right) A = \omega_{\min} W.
\end{aligned}$$

Thus, we obtained

$$\mathbf{E}_J\left[\sum_{i\in J}\omega_i\frac{a_ia_i^T}{\|a_i\|^2}\right]\succeq\omega_{\min}W. \tag{4.4}$$

Moreover, using that for any $Q\succeq 0$ we have $Q^2\preceq\lambda_{\max}(Q)Q$, the expectation over the second sum also yields the following upper bound:

$$\begin{aligned}\mathbf{E}_J\left[\left(\sum_{i\in J}\omega_i\frac{a_ia_i^T}{\|a_i\|^2}\right)^2\right]&\preceq\mathbf{E}_J\left[\lambda_{\max}\left(\sum_{i\in J}\omega_i\frac{a_ia_i^T}{\|a_i\|^2}\right)\left(\sum_{i\in J}\omega_i\frac{a_ia_i^T}{\|a_i\|^2}\right)\right]\\&\preceq(\max_{i\in J}\omega_i)\mathbf{E}_J\left[\lambda_{\max}\left(\sum_{i\in J}\frac{a_ia_i^T}{\|a_i\|^2}\right)\left(\sum_{i\in J}\omega_i\frac{a_ia_i^T}{\|a_i\|^2}\right)\right]\\&\preceq(\max_{i\in J}\omega_i)\mathbf{E}_J\left[\lambda_{\max}\left(A_J^T\mathrm{diag}\left(\frac{1}{\|a_i\|^2},i\in J\right)A_J\right)\left(\sum_{i\in J}\omega_i\frac{a_ia_i^T}{\|a_i\|^2}\right)\right]\\&\preceq(\max_{i\in J}\omega_i)^2\lambda_{\max}^{\text{block}}\mathbf{E}_J\left[\sum_{i\in J}\frac{a_ia_i^T}{\|a_i\|^2}\right]=\omega_{\max}^2\lambda_{\max}^{\text{block}}W,\end{aligned}$$

where we recall that $\lambda_{\max}^{\text{block}}=\max_{J\sim\mathbf{P}}\lambda_{\max}(A_J^T\mathrm{diag}(\frac{1}{\|a_i\|^2},i\in J)A_J)$. Therefore, taking conditional expectation w.r.t. the block J_k over entire history $\mathcal{F}_k=\{J_0,\dots,J_{k-1}\}$ in the recurrence above, we get

$$\mathbf{E}_J\left[\|x^{k+1}-x^*\|^2|\mathcal{F}_k\right]\leq(x^k-x^*)^T\left(I_n-2\alpha\omega_{\min}W+\alpha^2\omega_{\max}^2\lambda_{\max}^{\text{block}}W\right)(x^k-x^*).$$

In order to ensure decrease we need $2\alpha\omega_{\min}-\alpha^2\omega_{\max}^2\lambda_{\max}^{\text{block}}\geq 0$, that is, we get an extrapolated stepsize $\alpha\leq\frac{2\omega_{\min}}{\omega_{\max}^2\lambda_{\max}^{\text{block}}}$, and the optimal stepsize is obtained by maximizing $2\alpha\omega_{\min}-\alpha^2\omega_{\max}^2\lambda_{\max}^{\text{block}}$ in α, which leads to $\alpha^*=\frac{\omega_{\min}}{\omega_{\max}^2\lambda_{\max}^{\text{block}}}$. Hence, taking stepsize $\alpha=(2-\delta)\omega_{\min}/\omega_{\max}^2\lambda_{\max}^{\text{block}}$ for some $\delta\in(0,1]$, we get

$$\mathbf{E}_J\left[\|x^{k+1}-x^*\|^2|\mathcal{F}_k\right]\leq(x^k-x^*)^T\left(I_n-(2-\delta)\frac{\omega_{\min}^2}{\omega_{\max}^2\lambda_{\max}^{\text{block}}}W\right)(x^k-x^*).$$

On the other hand, it is known from the Courant–Fischer theorem that for any matrix A we have $\|Ax\|^2\geq\lambda_{\min}^{\text{nz}}(AA^T)\|x\|^2$ for all $x\in\text{range}(A^T)$. Moreover, we have that $x-\Pi_{\mathcal{X}}(x)\in\text{range}(A^T)$ for any x. Thus, for $W=A^TDA$, with the diagonal matrix $D=\mathrm{diag}(\frac{p_i}{\|a_i\|^2},i\in[m])$ nonsingular (recall that we assume probability distribution $\mathbf{P}$ such that $p_i>0$ for all $i\in[m]$ and the rows $a_i\neq 0$ for all $i\in[m]$), we get

$$\begin{aligned}(x^k-x_k^*)^TW(x^k-x_k^*)&=\|D^{1/2}A(x^k-x_k^*)\|^2\geq\lambda_{\min}^{\text{nz}}(A^TDA)\|x^k-x_k^*\|^2\\&=\lambda_{\min}^{\text{nz}}(W)\|x^k-x_k^*\|^2.\end{aligned}$$

Using this inequality in the recurrence above and taking expectation over the entire history we get

$$\mathbf{E}\left[\|x^{k+1}-x_{k+1}^*\|^2\right]\leq\left(1-(2-\delta)\frac{\omega_{\min}^2\lambda_{\min}^{\text{nz}}(W)}{\omega_{\max}^2\lambda_{\max}^{\text{block}}}\right)\mathbf{E}\left[\|x^k-x_k^*\|^2\right],$$

which shows an expected linear convergence rate for RaBK depending on the parameters $\lambda_{\min}^{\text{nz}}(W)$ and $\lambda_{\max}^{\text{block}}$ associated to the system matrix A and its submatrices A_J, respectively. ∎

Now, let us consider the uniform and partition sampling examples of section 3.4, where all the block samplings have the same size $|J| = \tau$. In this case we have $p_i = \frac{\tau}{m}$. Let us also consider the particular choices $\delta = 1$, weights $\omega_i = 1/\tau$, and matrices A with normalized rows, i.e., $\|a_i\| = 1$ for all $i \in [m]$. Hence, $\|A\|_F^2 = m$. Then, our convergence rate (4.3) becomes

$$\mathbf{E}\left[\|x^k - x_k^*\|^2\right] \leq \left(1 - \frac{\tau}{\lambda_{\max}^{\text{block}}} \frac{\lambda_{\min}^{\text{nz}}(A^T A)}{m}\right)^k \|x^0 - x_0^*\|^2. \tag{4.5}$$

Comparing (4.5) with the convergence rate (3.4) of the basic Kaczmarz method (recall that for normalized matrices, $\|A\|_F^2 = m$), we get an improvement of $\frac{\tau}{\lambda_{\max}^{\text{block}}} > 1$, which shows that for the RaBK algorithm with the new extrapolated stepsize (4.2), we can get a speed-up even of order approximately τ compared to the rate of the basic Kaczmarz algorithm on matrices with well-conditioned blocks (i.e., on matrices having $\lambda_{\max}^{\text{block}} \ll \tau$). Section 4.3 provides choices for the sampling that lead to a small stochastic conditioning parameter $\lambda_{\max}^{\text{block}}$.

4.2. Randomized average block Kaczmarz algorithm with adaptive stepsize. Since the previous algorithm involved a stepsize depending on $\lambda_{\max}^{\text{block}}$, which may be difficult to compute in large-scale settings (i.e., when the random variable J is complicated and the number of rows m is large), in this section we design a randomized block Kaczmarz algorithm with an adaptive stepsize, which does not require the computation of $\lambda_{\max}^{\text{block}}$. More precisely, we consider a variant of RaBK (Algorithm 4.1) with variable weights and an adaptive stepsize approximating online $\lambda_{\max}^{\text{block}}$. For simplicity of the notation, we define $\bar{\omega}_i^k = \frac{\omega_i^k}{\|a_i\|^2}$. Then we consider the iteration of RaBK (Algorithm 4.1) with an adaptive extrapolated stepsize of the form

$$0 < \alpha_k < 2L_k, \quad \text{where } L_k = \begin{cases} \frac{\sum_{i \in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i)^2}{\|\sum_{i \in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i) a_i\|^2} & \text{if } \exists i \in J_k \text{ s.t. } a_i^T x^k - b_i \neq 0, \\ \frac{1}{\lambda_{\max}\left(A_{J_k}^T \operatorname{diag}\left(\bar{\omega}_i^k, i \in J_k\right) A_{J_k}\right)} & \text{otherwise.} \end{cases} \tag{4.6}$$

Note that we do not need to compute L_k for the second case when implementing the algorithm, since then $a_i^T x^k - b_i = 0$ for all $i \in J_k$. Recall that we consider weights satisfying $0 < \omega_{\min} \leq \omega_i^k \leq \omega_{\max} < 1$ for all k, i and summing to 1. Hence, from Jensen's inequality we always have $L_k \geq 1$ and consequently $2L_k \geq 2$, i.e., we use extrapolation. Further, in our convergence analysis we take a stepsize of the form $\alpha_k = (2 - \delta)L_k$ for some $\delta \in (0,\ 1]$. Moreover, we denote $x_k^* = \Pi_{\mathcal{X}}(x^k)$, that is, the projection of x^k onto the solution set of the linear system. It has been observed in practice that block Kaczmarz iteration with this adaptive choice for the stepsize has better performance than the same algorithm but with stepsize $\alpha_k \in (0, 2)$; see, e.g., [1, 3, 4, 19, 20, 22, 23, 30]. However, to the best of our knowledge, there is no theory explaining *why* and *when* this adaptive method works. The next theorem proves the convergence rate of the adaptive algorithm depending explicitly on the geometric properties of the system matrix A and its submatrices A_J and answers the question related to the theoretical understanding of observed practical efficiency of extrapolated block Kaczmarz methods. Extensions of this result to general convex feasibility problems can be found in [22] and to convex optimization problems with many constraints in [23].

THEOREM 4.2. *Let $\{x^k\}_{k\geq 0}$ be generated by RaBK (Algorithm 4.1) with the adaptive stepsize $\alpha_k = (2-\delta)L_k$ for some $\delta \in (0,1]$ and the weights satisfying $0 < \omega_{\min} \leq \omega_i^k \leq \omega_{\max} < 1$ for all k, i. Then we have the following linear convergence in expectation:*

$$\mathbf{E}\left[\|x^k - x_k^*\|^2\right] \leq \left(1 - \frac{\delta(2-\delta)\omega_{\min}\lambda_{\min}^{nz}(W)}{\omega_{\max}\lambda_{\max}^{block}}\right)^k \|x^0 - x_0^*\|^2. \tag{4.7}$$

Proof. Using that $\langle x - x^*, (a_i^T x - b_i)a_i\rangle = (a_i^T x - b_i)^2$ in the update of RaBK, we get

$$\begin{aligned}
\|x^{k+1} - x_{k+1}^*\|^2 &= \left\| x^k - \alpha_k \left(\sum_{i\in J_k} \omega_k^i \frac{a_i^T x^k - b_i}{\|a_i\|^2} a_i \right) - x_{k+1}^* \right\|^2 \\
&= \|x^k - x_{k+1}^*\|^2 - 2\alpha_k \left(\sum_{i\in J_k} \omega_i^k \frac{(a_i^T x^k - b_i)^2}{\|a_i\|^2} \right) + \alpha_k^2 \left\| \sum_{i\in J_k} \omega_i^k \frac{(a_i^T x^k - b_i)^2}{\|a_i\|^2} a_i \right\|^2 \\
&= \|x^k - x_{k+1}^*\|^2 - 2(2-\delta) \frac{\left(\sum_{i\in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i)^2 \right)^2}{\| \sum_{i\in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i) a_i \|^2} + (2-\delta)^2 \frac{\left(\sum_{i\in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i)^2 \right)^2}{\| \sum_{i\in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i) a_i \|^2} \\
&= \|x^k - x_{k+1}^*\|^2 - \delta(2-\delta) \frac{\sum_{i\in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i)^2}{\| \sum_{i\in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i) a_i \|^2} \sum_{i\in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i)^2 \\
&= \|x^k - x_{k+1}^*\|^2 - \delta(2-\delta) L_k \sum_{i\in J_k} \bar{\omega}_i^k (a_i^T x^k - b_i)^2.
\end{aligned}$$

Note that we get the same recurrence also for the trivial choice

$$L_k = 1/\lambda_{\max}\left(A_{J_k}^T \mathrm{diag}\left(\bar{\omega}_i^k, i \in J_k\right) A_{J_k}\right).$$

Now let us bound L_k. For the nontrivial case, using the definition of the maximum eigenvalue and that $\lambda_{\max}(MN) = \lambda_{\max}(NM)$ for any two matrices M and N of appropriate dimensions, we have

$$\begin{aligned}
L_k &= \frac{(A_{J_k} x^k - b_{J_k})^T \mathrm{diag}\left(\bar{\omega}_i^k, i \in J_k\right) (A_{J_k} x^k - b_{J_k})}{\|A_{J_k}^T \mathrm{diag}\left(\bar{\omega}_i^k, i \in J_k\right) (A_{J_k} x^k - b_{J_k})\|^2} \\
&\geq \frac{1}{\lambda_{\max}\left(\mathrm{diag}\left(\sqrt{\bar{\omega}_i^k}, i \in J_k\right) A_{J_k} A_{J_k}^T \mathrm{diag}\left(\sqrt{\bar{\omega}_i^k}, i \in J_k\right)\right)} \\
&= \frac{1}{\lambda_{\max}\left(A_{J_k}^T \mathrm{diag}\left(\bar{\omega}_i^k, i \in J_k\right) A_{J_k}\right)}.
\end{aligned}$$

This inequality holds trivially for the second choice (case) of L_k. Therefore, we can

further bound L_k for all the cases as follows:

$$\begin{aligned} L_k &\geq \frac{1}{\lambda_{\max}\left(A_{J_k}^T \operatorname{diag}\left(\bar{\omega}_i^k, i \in J_k\right) A_{J_k}\right)} \\ &\geq \frac{1}{(\max_{i\in J_k} \omega_i^k)\lambda_{\max}\left(A_{J_k}^T \operatorname{diag}\left(1/\|a_i\|^2, i \in J_k\right) A_{J_k}\right)} \\ &\geq \frac{1}{\omega_{\max} \max_{J\sim \mathbf{P}} \lambda_{\max}\left(A_J^T \operatorname{diag}\left(1/\|a_i\|^2, i \in J\right) A_J\right)} \\ &= \frac{1}{\omega_{\max}\lambda_{\max}^{\text{block}}}. \end{aligned} \tag{4.8}$$

Using this bound in the recurrence above, we get

$$\begin{aligned} \|x^{k+1} - x_{k+1}^*\|^2 &\leq \|x^k - x_k^*\|^2 - \delta(2-\delta)\frac{1}{\omega_{\max}\lambda_{\max}^{\text{block}}} \sum_{i\in J_k} \omega_i^k \frac{(a_i^T x^k - b_i)^2}{\|a_i\|^2} \\ &\leq \|x^k - x_k^*\|^2 - \delta(2-\delta)\frac{\omega_{\min}}{\omega_{\max}\lambda_{\max}^{\text{block}}} \sum_{i\in J_k} \frac{(a_i^T x^k - b_i)^2}{\|a_i\|^2} \\ &= \|x^k - x_k^*\|^2 - \delta(2-\delta)\frac{\omega_{\min}}{\omega_{\max}\lambda_{\max}^{\text{block}}} (x^k - x_k^*)^T \sum_{i\in J_k} \frac{a_i a_i^T}{\|a_i\|^2} (x^k - x_k^*). \end{aligned}$$

Now, taking the conditional expectation and using again (3.6), we get

$$\begin{aligned} &\mathbf{E}_J\left[\|x^{k+1} - x_{k+1}^*\|^2 | \mathcal{F}_k\right] \\ &\leq \|x^k - x_k^*\|^2 - \delta(2-\delta)\frac{\omega_{\min}}{\omega_{\max}\lambda_{\max}^{\text{block}}} (x^k - x_k^*)^T A^T \operatorname{diag}\left(\frac{p_i}{\|a_i\|^2}, i \in [m]\right) A(x^k - x_k^*) \\ &= \|x^k - x_k^*\|^2 - \delta(2-\delta)\frac{\omega_{\min}}{\omega_{\max}\lambda_{\max}^{\text{block}}} (x^k - x_k^*)^T W (x^k - x_k^*). \end{aligned}$$

It is also known from the Courant–Fischer theorem that for any matrix A we have $\|Ax\|^2 \geq \lambda_{\min}^{\text{nz}}(AA^T)\|x\|^2$ for all $x \in \text{range}(A^T)$. Moreover, we have that $x - \Pi_{\mathcal{X}}(x) \in \text{range}(A^T)$ for any x. In conclusion, using that $W = A^T D A$, with the diagonal matrix $D = \operatorname{diag}\left(\frac{p_i}{\|a_i\|^2}, i \in [m]\right)$ invertible, we get that

$$\begin{aligned} (x^k - x_k^*)^T W (x^k - x_k^*) &= \|D^{1/2} A (x^k - x_k^*)\|^2 \geq \lambda_{\min}^{\text{nz}}(A^T D A)\|x^k - x_k^*\|^2 \\ &= \lambda_{\min}^{\text{nz}}(W)\|x^k - x_k^*\|^2. \end{aligned}$$

Using this inequality in the recurrence above and taking expectation over the entire history we get

$$\mathbf{E}\left[\|x^{k+1} - x_{k+1}^*\|^2\right] \leq \left(1 - \delta(2-\delta)\frac{\omega_{\min}}{\omega_{\max}\lambda_{\max}^{\text{block}}} \lambda_{\min}^{\text{nz}}(W)\right) \mathbf{E}\left[\|x^k - x_k^*\|^2\right]$$

and hence prove the statement of the theorem. ∎

There is a tight connection between the constant stepsize (4.2) and the adaptive stepsize (4.6). Indeed, for simplicity let us consider uniform weights $\omega_i^k = 1/\tau$ and normalized matrices ($\|a_i\| = 1$ for all i, k). Then, from the definition of the maximum eigenvalue and (4.8), we obtain

$$L_k = \tau \frac{\|A_{J_k} x^k - b_{J_k}\|^2}{\|A_{J_k}^T (A_{J_k} x^k - b_{J_k})\|^2} \geq \tau \frac{1}{\lambda_{\max}\left(A_{J_k}^T A_{J_k}\right)} \geq \tau \frac{1}{\lambda_{\max}^{\text{block}}}.$$

Hence, L_k represents an online approximation of $\tau/\lambda_{\max}^{\text{block}}$, and therefore

$$\alpha_k = 2L_k \geq \alpha = 2\frac{\omega_{\min}}{\omega_{\max}^2 \lambda_{\max}^{\text{block}}} = 2\tau \frac{1}{\lambda_{\max}^{\text{block}}}. \tag{4.9}$$

In conclusion, the adaptive stepsize (4.6) can be viewed as a practical online approximation of the constant extrapolated stepsize (4.2). Finally, let us simplify the convergence rate (4.7) for the uniform and partition sampling examples of section 3.4 having all the blocks sampling the same size $|J| = \tau$. In this case we have $p_i = \frac{\tau}{m}$. Let us also consider the particular choices $\delta = 1$, weights $\omega_i = 1/\tau$, and normalized matrices A. Then, our convergence rate (4.7) becomes

$$\mathbf{E}\left[\|x^k - x_k^*\|^2\right] \leq \left(1 - \frac{\tau}{\lambda_{\max}^{\text{block}}} \frac{\lambda_{\min}^{\text{nz}}(A^T A)}{m}\right)^k \|x^0 - x_0^*\|^2. \tag{4.10}$$

We observe that this convergence rate coincides with (4.5). However, the adaptive block Kaczmarz scheme has more chances to accelerate, since from (4.9) the variable stepsize is, in general, larger than the constant stepsize counterpart.

4.3. When block Kaczmarz works. Comparing the convergence rates of the RaBK algorithm with the constant stepsize (4.2) and the adaptive stepsize (4.6) given in (4.5) and (4.10), respectively, with the convergence rate of the basic Kaczmarz method given in (3.4) for normalized matrices, we obtain an improvement of $\frac{\tau}{\lambda_{\max}^{\text{block}}} > 1$ for the block variants. Recall that the stochastic conditioning parameter $\lambda_{\max}^{\text{block}}$ depends also on τ (i.e., $\lambda_{\max}^{\text{block}} = \lambda_{\max}^{\text{block}}(\tau)$), since it is defined as

$$\lambda_{\max}^{\text{block}} = \max_{J \sim \mathbf{P}} \lambda_{\max}\left(A_J^T \text{diag}\left(\frac{1}{\|a_i\|^2}, i \in J\right) A_J\right).$$

Therefore, we can get a speed-up even of order approximately τ for well conditioned matrices, i.e., for matrices having $\lambda_{\max}^{\text{block}} \ll \tau$. This shows that the probability $\mathbf{P}$ plays a key role in defining the importance sampling procedure and consequently in the convergence behavior of RaBK. Ideally, we should maximize the function $\tau \mapsto \tau/\lambda_{\max}^{\text{block}}(\tau)$ over $\tau \in [1:m]$ in order to get the best convergence rate for RaBK. However, this is a very difficult maximization problem. Fortunately, the operator theory literature provides detailed information about the existence of good probabilities defining the importance sampling that makes $\lambda_{\max}^{\text{block}}$ small and consequently $\tau/\lambda_{\max}^{\text{block}}$ large. This is usually referred to in the literature as *good paving* [24]. This section summarizes the main results from the literature on row paving and provides a technique for constructing a good paving. The idea is to find a random partition of the rows of the matrix A such that all subsets are of approximately equal size. Results on existence of good row pavings were derived, e.g., in [35].

LEMMA 4.3 ([35]). *Let A be a normalized matrix with m rows, and let $\theta \in (0,1)$. Then, there is a randomized partition $\{J_1, \ldots, J_\ell\}$ of the row indices with*

$$\ell \geq \mathcal{O}(\|A\|^2 \log(1+m)/\theta^2)$$

such that $\lambda_{\max}^{block} \leq 1 + \theta$.

Although this is only an existential result, the literature describes several efficient algorithms for constructing good row pavings. For example, assume that κ is

a permutation of the set $[m] = \{1, 2, \ldots, m\}$, chosen uniformly at random. For each $i = 1 : \ell$, define the subsets

$$J_i = \left\{ \kappa(l) : l = \left\lfloor (i-1)\frac{m}{\ell} \right\rfloor + 1, \ldots, \left\lfloor \frac{m}{\ell} \right\rfloor \right\}.$$

It is clear that $\{J_1, \ldots, J_\ell\}$ is a random partition of $[m]$ into ℓ blocks of approximately equal size. For every normalized matrix, such a random partition leads to a row paving whose $\lambda_{\max}^{\text{block}}$ is relatively small.

LEMMA 4.4 ([34]). *Let A be a normalized matrix with m rows. Consider a randomized partition $\{J_1, \ldots, J_\ell\}$ of the row indices, as described above, with $\ell \geq \|A\|^2$ subsets. Then, $\{J_1, \ldots, J_\ell\}$ is a row paving with the upper bound $\lambda_{\max}^{block} \leq 6\log(1+m)$ with probability at least $1 - m^{-1}$.*

A proof of this type of result appears in [34]; see also [24]. By merging our theorems on the convergence of the RaBK algorithm with the previous result on the good paving, we obtain the following.

THEOREM 4.5. *Let A be a normalized matrix, and let $\{J_1, \ldots, J_\ell\}$ be a random partition of the rows of A, as given by Lemma* 4.4*, such that $\tau = m/\ell$ is a positive integer. Under the assumptions of Theorems* 4.1 *and* 4.2*, the RaBK method (Algorithm* 4.1*), with weights $\omega_i^k = 1/\tau = \ell/m$ for all i, k, and constant stepsize* (4.2) *or adaptive stepsize* (4.6) *with $\delta = 1$, admits with probability at least $1 - m^{-1}$ the convergence estimate*

$$\mathbf{E}\left[\|x^k - x_k^*\|^2\right] \leq \left(1 - \frac{\lambda_{\min}^{nz}(A^T A)}{6\log(1+m)\|A\|^2}\right)^k \|x^0 - x_0^*\|^2. \tag{4.11}$$

In conclusion, our new convergence analysis shows when a block variant of the Kaczmarz algorithm really works; i.e., we can choose a subset of rows $\tau > 1$ at each step when $\lambda_{\max}^{\text{block}} \ll \tau$. Hence, a distributed implementation of the RaBK algorithm is most effective when the probability distribution $\mathbf{P}$ yields a partition of the rows into well-conditioned blocks. Otherwise, we can just apply the basic Kaczmarz algorithm with $\tau = 1$. Moreover, our analysis shows that the *optimal block size* is of order $\tau^* \sim m/\|A\|^2$. Assuming, for simplicity, that $\tau = m/\ell$ is a positive integer, from Lemma 4.4,

$$\lambda_{\max}^{\text{block}} \leq 6\log(1+m) \ll \tau = \frac{m}{\ell} \simeq \frac{m}{\|A\|^2}$$

holds with high probability, provided that the matrix A satisfies the following inequality:

$$\|A\|^2 \ll \frac{m}{6\log(1+m)}. \tag{4.12}$$

Recall that, for a normalized matrix A with m rows, the squared spectral norm $\|A\|^2$ attains its maximal value m when $\operatorname{rank}(A) = 1$, i.e., its rows are identical. Therefore, the inequality (4.12) stipulates that the rows of A must exhibit a large amount of diversity in order for the RaBK algorithm with extrapolated stepsize (4.2) or (4.6) to perform better than the basic Kaczmarz scheme.

The convergence rate we have obtained in (4.11) for RaBK is the best we can hope for from an algorithm: the rate depends on the condition number of the matrix A, $\lambda_{\min}^{\text{nz}}(A^T A)/\|A\|^2$, and the dimension of the problem, m (number of rows of A), enters logarithmically into the rate. Moreover, our theory also yields the optimal block size,

$\tau^* = m/\|A\|^2$, that we should choose in an implementation. Note that convergence rates similar to (4.11) have been derived in [24] for the block projection Kaczmarz algorithm (1.3) with the particular stepsize $\alpha_k = 1$. However, RaBK requires the computation of τ scalar products in $\mathbb{R}^n$ at each iteration, so that its computational cost per iteration is at most $\mathcal{O}(\tau n)$ flops, much less when A is sparse, and thus cheaper than the one corresponding to block projection Kaczmarz (1.3) that requires solving a least-squares problem at each iteration in about $\mathcal{O}(\tau^2 n)$ flops.

5. Randomized block Kaczmarz algorithm with Chebyshev-based stepsize. Finally, we show that we can also choose extrapolated stepsizes in RaBK (Algorithm 4.1) based on the roots of Chebyshev polynomials. For simplicity, we consider either the uniform or the partition sampling of section 3.4 having $|J| = \tau$ (see Remark 5.3 for more general settings). We also assume normalized matrices A and constant weights $\omega_k^i = 1/\tau$ for all k, i. Under these settings, for the RaBK algorithm with Chebyshev-based stepsize we derive linear or sublinear convergence estimates depending on whether $\lambda_{\min}(AA^T) > 0$ or $\lambda_{\min}(AA^T) = 0$, respectively. Below we investigate these two cases.

5.1. Case 1: $\boldsymbol{\lambda_{\min}(AA^T) > 0}$. We get the following linear convergence for this variant of RaBK.

THEOREM 5.1. *Assume normalized matrix A such that $\lambda_{\min}(AA^T) > 0$. Let $\{x^k\}_{k\geq 0}$ be generated by RaBK (Algorithm 4.1) with the uniform or partition sampling and the weights $\omega_k^i = 1/\tau$ for all k, i. Further, for a fixed number of iterations k the stepsizes $\{\alpha_j\}_{j=0}^{k-1}$ depend on the roots of the Chebyshev polynomial of degree k (see Appendix A) as follows:*

$$\alpha_j = \frac{2m}{(\lambda_{\max}(AA^T) + \lambda_{\min}(AA^T)) + (\lambda_{\max}(AA^T) - \lambda_{\min}(AA^T))\cos\left(\frac{2\kappa(j)+1}{2k}\pi\right)},$$

where κ is a given permutation of $[0 : k-1]$. Then we have the following linear convergence for the expected iterates:

$$\|\mathbf{E}\left[x^k - x_k^*\right]\|^2 \leq \frac{4\lambda_{\max}(AA^T)}{\lambda_{\min}(AA^T)}\left(1 - \sqrt{\frac{\lambda_{\min}(AA^T)}{\lambda_{\max}(AA^T)}}\right)^{2k}\|x^0 - x_0^*\|^2. \tag{5.1}$$

Proof. For the iteration of RaBK (Algorithm 4.1) we have for any solution $x^* \in \mathcal{X}$,

$$\begin{aligned} x^{k+1} - x^* &= x^k - x^* - \alpha_k\left(\sum_{i\in J_k}\omega_k^i \frac{a_i^T x^k - b_i}{\|a_i\|^2}a_i\right) \\ &\overset{\omega_k^i = 1/\tau, \|a_i\|=1}{=} x^k - x^* - \frac{\alpha_k}{\tau}\left(\sum_{i\in J_k}(a_i^T x^k - b_i)a_i\right) \\ &= x^k - x^* - \frac{\alpha_k}{\tau}\left(\sum_{i\in J_k} a_i a_i^T (x^k - x^*)\right) = \left(I_n - \frac{\alpha_k}{\tau}\left(\sum_{i\in J_k} a_i a_i^T\right)\right)(x^k - x^*). \end{aligned}$$

Taking conditional expectation and using (3.6) with $p_i = \tau/m$ for uniform or partition

sampling, we get

$$\mathbf{E}_J\left[x^{k+1}-x^*|\mathcal{F}_k\right] = \left(I_n - \frac{\alpha_k}{\tau}\left(\sum_{i\in[m]} p_i a_i a_i^T\right)\right)(x^k - x^*) \tag{5.2}$$
$$\overset{p_i=\tau/m}{=} \left(I_n - \frac{\alpha_k}{m}A^TA\right)(x^k-x^*).$$

Multiplying this recurrence from the left with A we get

$$\mathbf{E}_J\left[Ax^{k+1}-Ax^*|\mathcal{F}_k\right] = \left(A - \frac{\alpha_k}{m}AA^TA\right)(x^k-x^*) = \left(I_m - \frac{\alpha_k}{m}AA^T\right)(Ax^k - Ax^*),$$

or, equivalently, using that $Ax^* = b$ and taking expectations over the entire history, we obtain

$$\mathbf{E}\left[Ax^{k+1}-b\right] = \left(I_m - \frac{\alpha_k}{m}AA^T\right)\mathbf{E}\left[Ax^k - b\right].$$

Iterating this recurrence and defining the matrix $G = \frac{1}{m}AA^T \in \mathbb{R}^{m\times m}$, we obtain

$$\mathbf{E}\left[Ax^k-b\right] = \prod_{j=0}^{k-1}\left(I_m - \alpha_j\frac{1}{m}AA^T\right)(Ax^0-b) = \prod_{j=0}^{k-1}\left(I_m-\alpha_jG\right)(Ax^0-b).$$

If we define the polynomial in the matrix G as $P_k(G) = \prod_{j=0}^{k-1}(I_m-\alpha_jG)$, then we can bound the norm of the expected residual by

$$\|\mathbf{E}\left[Ax^k-b\right]\| = \|P_k(G)(Ax^0-b)\| \le \|P_k(G)\|\cdot\|Ax^0-b\|.$$

Recall that we consider a consistent linear system with $\lambda_{\min}(AA^T) > 0$. Then, the spectrum of $G = \frac{1}{m}AA^T$ satisfies $\Lambda(G) \subset \mathbb{R}_{++}$. More precisely,

$$0 < \underbrace{\frac{1}{m}\lambda_{\min}(AA^T)}_{=\ell} \le \lambda_i(G) \le \underbrace{\frac{1}{m}\lambda_{\max}(AA^T)}_{=u} < \infty \quad \forall i = 1:m.$$

Therefore, if we denote by λ_i the ith eigenvalue of G, we have the following bound [8]:

$$\|\mathbf{E}\left[Ax^k-b\right]\| \le \|P_k(G)\|\cdot\|Ax^0-b\| \le \max_{i=1:m}|P_k(\lambda_i)|\cdot\|Ax^0-b\| \le \max_{\lambda\in[\ell,u]}|P_k(\lambda)|\cdot\|Ax^0-b\|.$$

In conclusion, we can choose the stepsizes α_j for $j = 0:k-1$ such that $P_k(\lambda) = \prod_{j=0}^{k-1}(1-\alpha_j\lambda)$ is the polynomial least deviating from zero on the interval $[\ell, u] = [\lambda_{\min}(AA^T)/m,\ \lambda_{\max}(AA^T)/m]$ and satisfying $P_k(0) = 1$. It is well known that this is the polynomial given in terms of a Chebyshev polynomial (see Appendix A for a brief review of the main properties of Chebyshev polynomials):

$$P_k(\lambda) = T_k\left(\frac{2\lambda}{u-\ell} - \frac{u+\ell}{u-\ell}\right)\Big/ T_k\left(-\frac{u+\ell}{u-\ell}\right).$$

Then we can guarantee the following linear convergence in expectation (see Lemma A.1 in Appendix A):

(5.3)
$$\|\mathbf{E}\left[Ax^k - b\right]\| \leq 2\left(\frac{\sqrt{u}-\sqrt{\ell}}{\sqrt{u}+\sqrt{\ell}}\right)^k \|Ax^0 - b\| \leq 2\left(1 - \sqrt{\frac{\lambda_{\min}(AA^T)}{\lambda_{\max}(AA^T)}}\right)^k \|Ax^0 - b\|.$$

The stepsizes α_j, for $j = 0 : k-1$, are chosen as the inverse roots of polynomial $P_k(\lambda)$ (see Appendix A):

$$\begin{aligned}\alpha_j &= 2/\left((u+\ell) + (u-\ell)\cos\left(\frac{2\kappa(j)+1}{2k}\pi\right)\right)\\ &= 2m/\Big((\lambda_{\max}(AA^T) + \lambda_{\min}(AA^T)) + (\lambda_{\max}(AA^T)\\ &\qquad -\lambda_{\min}(AA^T))\cos\left(\frac{2\kappa(j)+1}{2k}\pi\right)\Big),\end{aligned}$$

where κ is some fixed permutation of $[0 : k-1]$. We can also derive convergence rates in $\mathbf{E}\left[x^k - x_k^*\right]$ using that $\mathbf{E}\left[x^k - x_k^*\right] \in \text{range}(A^T)$, and consequently from the Courant–Fischer lemma and (5.3) we have

$$\begin{aligned}&\lambda_{\min}(AA^T)\|\mathbf{E}\left[x^k - x_k^*\right]\|^2 \leq \|A\mathbf{E}\left[x^k - x_k^*\right]\|^2 = \|\mathbf{E}\left[Ax^k - b\right]\|^2\\ &\leq 4\left(1 - \sqrt{\frac{\lambda_{\min}(AA^T)}{\lambda_{\max}(AA^T)}}\right)^{2k} \|Ax^0 - b\|^2\\ &\leq 4\lambda_{\max}(AA^T)\left(1 - \sqrt{\frac{\lambda_{\min}(AA^T)}{\lambda_{\max}(AA^T)}}\right)^{2k} \|x^0 - x_0^*\|^2,\end{aligned}$$

and thus prove the linear convergence estimate of the theorem. ∎

From Jensen's inequality we have $\|\mathbf{E}[\cdot]\| \leq \mathbf{E}[\|\cdot\|]$. In conclusion, $\|\mathbf{E}[\cdot]\|$ is a weaker criterion than $\mathbf{E}[\|\cdot\|]$. Note that convergence rates in the weaker criterion $\|\mathbf{E}\left[x^k - x_k^*\right]\|$ have also been given for another variant of the Kaczmarz algorithm in [31] and for the random coordinate descent method in [33]. Note that the convergence estimate (5.1) depends on the square root of the condition number of the matrix. We usually refer to an algorithm having a convergence rate of this form as an *accelerated* RaBK method; see also [27, 14, 31]. Note also that the convergence rate from Theorem 5.1 is the same as that for the conjugate gradient method, and it is optimal for this class of iterative schemes. However, since this rate does not depend on the size of the blocks $|J|$, we usually implement this accelerated variant of Kaczmarz by sampling single rows; that is, we choose $|J| = 1$. For this case, based on the previous derivations and the recurrence relation for the Chebyshev polynomials, we can also easily devise an iterative process that depends on x^k and x^{k-1},

$$x^{k+1} = \beta_k\left(x^k - \alpha\frac{a_{i_k}^T x^k - b_{i_k}}{\|a_{i_k}\|^2}a_{i_k}\right) + \gamma_k x^{k-1},$$

for some appropriate parameters α, β_k, and γ_k. Due to space limitations we omit these details here.

5.2. Case 2: $\lambda_{\min}(AA^T) = 0$. In this case we get sublinear convergence for this variant of RaBK.

THEOREM 5.2. *Assume a normalized matrix A such that $\lambda_{\min}(AA^T) = 0$. Let $\{x^k\}_{k\geq 0}$ be generated by RaBK (Algorithm 4.1) with the uniform or partition sampling and the weights $\omega_k^i = 1/\tau$ for all k, i. Further, for a fixed number of iterations k the stepsizes $\{\alpha_j\}_{j=0}^{k-1}$ depend on the roots of the Chebyshev polynomial of degree k as follows:*

$$\alpha_j = \frac{m\left(1-\cos\left(\frac{2k+1}{2(k+1)}\pi\right)\right)}{\lambda_{\max}(AA^T)\left(\cos\left(\frac{2\kappa(j)+1}{2(k+1)}\pi\right)-\cos\left(\frac{2k+1}{2(k+1)}\pi\right)\right)},$$

where κ is some permutation of $[0:k-1]$. Then, we have the following sublinear convergence for the residual of the normal system in expectation:

$$\|\mathbf{E}\left[A^TAx^k - A^Tb\right]\| = \|\mathbf{E}\left[Ax^k - b\right]\|_{(AA^T)} \leq \frac{\pi\lambda_{\max}(AA^T)}{2(k+1)^2}\|x^0 - x^*\|. \tag{5.4}$$

Proof. From (5.2) we also get the relation

$$\mathbf{E}\left[x^k - x^*\right] = \prod_{j=0}^{k-1}\left(I_n - \alpha_j\frac{1}{m}A^TA\right)(x^0 - x^*).$$

Now, if we consider the normal system $A^TAx = A^Tb$, which coincides with $\nabla f(x) = 0$, we have

$$\|\mathbf{E}\left[A^TAx^k - A^Tb\right]\| = \|\mathbf{E}\left[A^TA(x^k - x^*)\right]\| = \left\|A^TA\prod_{j=0}^{k-1}\left(I_n - \alpha_j\frac{1}{m}A^TA\right)(x^0 - x^*)\right\|,$$

where x^* denotes any solution of $Ax = b$ (recall that we consider consistent linear systems). If we define the matrix $G = \frac{1}{m}A^TA$ and the polynomial $Q_k(G) = G\prod_{j=0}^{k-1}(I_n - \alpha_jG)$, then we obtain the following bound for the residual of the normal system in expectation:

$$\|\mathbf{E}\left[A^TAx^k - A^Tb\right]\| = m\|Q_k(G)(x^0 - x^*)\| \leq m\|Q_k(G)\|\|x^0 - x^*\|.$$

Since we assume $\lambda_{\min}(AA^T) = \lambda_{\min}(A^TA) = 0$, the spectrum of $G = \frac{1}{m}A^TA$ satisfies

$$0 \leq \lambda_i(G) \leq \underbrace{\frac{1}{m}\lambda_{\max}(A^TA)}_{=u} < \infty \quad \forall i = 1:m.$$

Therefore, if we denote by λ_i the ith eigenvalue of G, we have the following bound:

$$\begin{aligned}\|\mathbf{E}\left[A^TAx^k - A^Tb\right]\| &\leq m\|Q_k(G)\| \cdot \|x^0 - x^*\| \leq m\max_{i=1:m}|Q_k(\lambda_i)| \cdot \|x^0 - x^*\| \\ &\leq m\max_{\lambda\in[0,u]}|Q_k(\lambda)| \cdot \|x^0 - x^*\|.\end{aligned}$$

In conclusion, we can choose the stepsizes α_j for $j = 0:k-1$ such that $Q_k(\lambda) = \lambda\prod_{j=0}^{k-1}(1-\alpha_j\lambda)$ of degree $k+1$ is the polynomial least deviating from zero on the interval $[0,u]$ and satisfying $Q_k(0) = 0$ and $Q_k'(0) = 1$. We show below that this

polynomial is also given in terms of a Chebyshev polynomial. Indeed, let us consider the closest root to -1 of the Chebyshev polynomial of degree $k+1$ (i.e., T_{k+1}):

$$r_{k+1} = \cos\left(\frac{2k+1}{2(k+1)}\pi\right) = \cos\left(\pi - \frac{1}{2(k+1)}\pi\right).$$

Then we define the polynomial

$$Q_k(\lambda) = \frac{u}{1-r_{k+1}} \frac{T_{k+1}\left(r_{k+1} + \frac{1-r_{k+1}}{u}\lambda\right)}{T'_{k+1}(r_{k+1})}.$$

Note that this polynomial satisfies the required properties $\deg(Q_k) = k+1$, $Q_k(0) = \frac{uT_{k+1}(r_{k+1})}{(1-r_{k+1})T'_{k+1}(r_{k+1})} = 0$ (recall that r_{k+1} is the $k+1$ root of T_{k+1}), and $Q'_k(0) = \frac{T'_{k+1}(r_{k+1})}{T'_{k+1}(r_{k+1})} = 1$. In conclusion, we get the following bound for this choice of $Q_k(\lambda)$:

$$\begin{aligned} m \max_{\lambda\in[0,u]} |Q_k(\lambda)| &= m \max_{\lambda\in[0,u]} \left| \frac{u}{1-r_{k+1}} \frac{T_{k+1}\left(r_{k+1} + \frac{1-r_{k+1}}{u}\lambda\right)}{T'_{k+1}(r_{k+1})} \right| \\ &\leq m \frac{u}{|T'_{k+1}(r_{k+1})|} = \frac{\lambda_{\max}(A^TA)}{|T'_{k+1}(r_{k+1})|}, \end{aligned}$$

where in the inequality we used that $|T_{k+1}(x)| \leq 1$ for any $x \in [-1,\ 1]$ and that the root $r_{k+1} \leq 0$ (see Appendix A). Further, since $T_{k+1}(\cos(\theta)) = \cos((k+1)\theta)$, if we differentiate we get $\sin(\theta)T'_{k+1}(\cos(\theta)) = (k+1)\sin((k+1)\theta)$. Now, for $r_{k+1} = \cos(\pi - \pi/(2k+2))$ we obtain

$$|T'_{k+1}(r_{k+1})| = \frac{(k+1)|\sin((k+1)\pi - \pi/2)|}{|\sin(\pi - \pi/(2k+2))|} = \frac{k+1}{|\sin(\pi - \pi/(2k+2))|} = \frac{2(k+1)^2}{\pi}$$

for k sufficiently large (we used that $\sin(\pi - \theta) \sim \theta$ for θ small). In conclusion, we get the following sublinear convergence (using the notation $\|u\|_{(AA^T)} = \|A^Tu\|$):

$$\|\mathbf{E}\left[A^TAx^k - A^Tb\right]\| = \|\mathbf{E}\left[Ax^k - b\right]\|_{(AA^T)} \leq \frac{\pi\lambda_{\max}(A^TA)}{2(k+1)^2}\|x^0 - x^*\|$$

for k sufficiently large (i.e., for k such that $\sin(\pi - \pi/(2k+2)) \sim \pi/(2k+2)$). Finally, using that $\lambda_{\max}(A^TA) = \lambda_{\max}(AA^T)$ we get (5.4). The stepsizes α_j, for $j = 0 : k-1$, are chosen as the inverse roots of polynomial $Q_k(\lambda)$ (see Appendix A):

$$\begin{aligned} \alpha_j &= (1-r_{k+1})u^{-1} / \left(\cos\left(\frac{2\kappa(j)+1}{2(k+1)}\pi\right) - r_{k+1}\right) \\ &= \frac{m\left(1 - \cos\left(\frac{2k+1}{2(k+1)}\pi\right)\right)}{\lambda_{\max}(AA^T)\left(\cos\left(\frac{2\kappa(j)+1}{2(k+1)}\pi\right) - \cos\left(\frac{2k+1}{2(k+1)}\pi\right)\right)}, \end{aligned}$$

where κ is some fixed permutation of $[0 : k-1]$. ∎

Remark 5.3. Note that the convergence results from Theorems 5.1 and 5.2 hold for general matrices A (not necessarily normalized) and general probability distributions $\mathbf{P}$ (not necessarily uniform). Under these more general settings the matrix G defined above takes the form $G = A^TDA$, where $D = \mathrm{diag}(p_i/(\tau\|a_i\|^2), i \in [m])$. Moreover, in this case $\ell = \lambda_{\min}(A^TDA)$ and $u = \lambda_{\max}(A^TDA)$, which, however, cannot be written explicitly in terms of the eigenvalues of A^TA as before.

Note that the RaBK algorithm with Chebyshev-based stepsize belongs to the class of Chebyshev semi-iterative methods [8]. However, to the best of our knowledge, this work is the first to use the properties of the Chebyshev polynomials in order to accelerate the convergence rate of the RaBK algorithm. Other types of acceleration of the Kaczmarz algorithm have been proposed, e.g., in [31, 14, 7, 10, 18]. For example, in [31] two dependent steps of the basic randomized Kaczmarz algorithm are taken, one from x^k and one from x^{k-1}, and then an affine combination of the results produces the next iterate x^{k+1}. For this scheme, [31] derives a convergence rate similar to that in Theorem 5.1. In [14], Nesterov's accelerated random coordinate descent method from [27] is applied to the dual problem (3.3), leading in the primal space to an accelerated randomized Kaczmarz scheme with momentum. For this accelerated Kaczmarz scheme, [14] derives the convergence rate $\mathbf{E}\left[\|x^k - x_k^*\|^2\right] \leq (1 - \sqrt{\lambda_{\min}(AA^T)}/m)^k \|x^0 - x_0^*\|^2$. Although this rate is worse than (5.1) in terms of constants, it is given in the stronger criterion $\mathbf{E}\left[\|x^k - x_k^*\|^2\right]$. It remains an open problem whether Theorems 5.1 and 5.2 can also be given in the stronger criterion $\mathbf{E}\left[\|x^k - x_k^*\|^2\right]$. From some preliminary numerical simulations, we have observed a large variance of the last iterate generated by RaBK with Chebyshev stepsize. Hence, more work is needed in this direction.

6. Simulations. In this section we study the computational behavior of RaBK on a variety of test problems. The simulations were performed in MATLAB on an Intel Core i7 computer with 16GB RAM. We start by comparing several variants of RaBK algorithms for dense A, and then for sparse A. We also compare RaBK with random block projection Kaczmarz (RB-proj-K) from [24] (see (1.3)) and conjugate gradient (CG). We consider the following RaBK variants:

1. RaBK with uniform sampling and constant stepsize $\alpha = 1.95$ (referred to as RaBK constant (RaBK-c).
2. RaBK with uniform sampling and constant extrapolated stepsize (4.2): $\alpha = \frac{1.95\tau}{\lambda_{\max}^{\text{block}}}$ (referred to as RaBK with extrapolation (RaBK-e)).
3. RaBK with uniform sampling and adaptive extrapolated stepsize (4.6): $\alpha_k = 1.95 L_k$ (referred to as adaptive RaBK (RaBK-a)).
4. RaBK with partition sampling as in Lemma 4.4 and constant extrapolated stepsize (4.2): $\alpha = \frac{1.95\tau}{\lambda_{\max}^{\text{block}}}$ (referred to as RaBK with extrapolation using good paving (RaBK-e-paved)).
5. RaBK with partition sampling as in Lemma 4.4 and adaptive extrapolated stepsize (4.6): $\alpha_k = 1.95 L_k$ (referred to as adaptive RaBK with good paving (RaBK-a-paved))

We measure performance by plotting the residual error $\|Ax - b\|$ against the number of full iterations ($\frac{k\tau}{m}$), so that the number of operations for one pass through the rows of A are the same for all the algorithms. A random initial point x^0 and constant weights $\omega_k^i = 1/\tau$ are used in all algorithms.

6.1. RaBK on dense data. Synthetic data for this test is generated as follows: all elements of the data matrix $A \in \mathbb{R}^{m \times n}$ and the optimal solution $x^* \in \mathbb{R}^n$ are chosen independent and identically distributed $\mathcal{N}(0, 1)$. The length of all rows in A is normalized to 1. The right-hand side is set to $b = Ax^*$. For the uniform sampling variants we choose $\tau = 10$, while for partition sampling schemes based on good pavings we choose optimal $\tau = \lfloor \frac{m}{\|A\|^2} \rfloor$.

From Figures 1–3, we observe the following behavior for the previous variants of RaBK on dense data: as predicted by theory, RaBK based on good paving partition

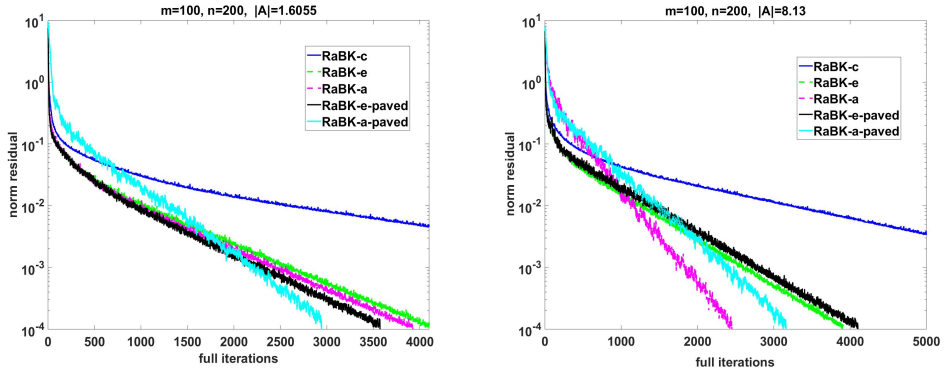

FIG. 1. *Comparison among RaBK algorithms on dense data. Left: small* $\|A\|$. *Right: large* $\|A\|$. *As predicted by the theory, RaBK variants based on good paving partition work better on matrices with* $\|A\|^2 \ll m$.

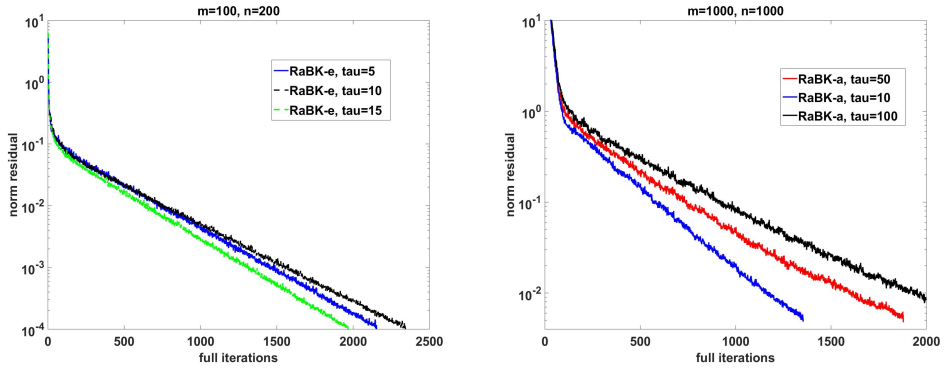

FIG. 2. *Behavior of RaBK-e (left) and RaBK-a (right) on dense data for varying* τ. *Increasing minibatch size* τ *over some threshold does not necessarily lead to a better convergence of RaBK.*

sampling works better when the norm of the matrix is small, and the best convergence is achieved for optimal batch size $\tau_{\text{opt}} = \lfloor m/\|A\|^2 \rfloor$ (see Lemma 4.4); for all variants of RaBK, increasing batch size does not necessarily lead to a better complexity.

6.2. RaBK on sparse data. We also compare variants of RaBK, basic Kaczmarz (K), RB-proj-K from [24] (see (1.3)), and CG on sparse matrices using either synthetic data generated by $\mathcal{N}(0,1)$ or coming from a finite difference discretization on a uniform grid with N points on each axis of a 2-dimensional heat equation on a square. The optimal solution and right-hand side are generated as in the dense case. For the uniform sampling variants $\tau = 50$, while for paved partition sampling schemes $\tau = \lfloor \frac{m}{\|A\|^2} \cdot \rfloor$.

From Figures 4–5, we observe the following behavior for the variants of RaBK, K, RB-proj-K given in (1.3) (see also [24]) and CG on sparse data: RaBK based on good paving partition sampling (RaBK-a-paved) usually performs better than RB-proj-K and the basic Kaczmarz for large dimensions; RaBK based on good paving partition sampling is also faster than CG when the dimension is large.

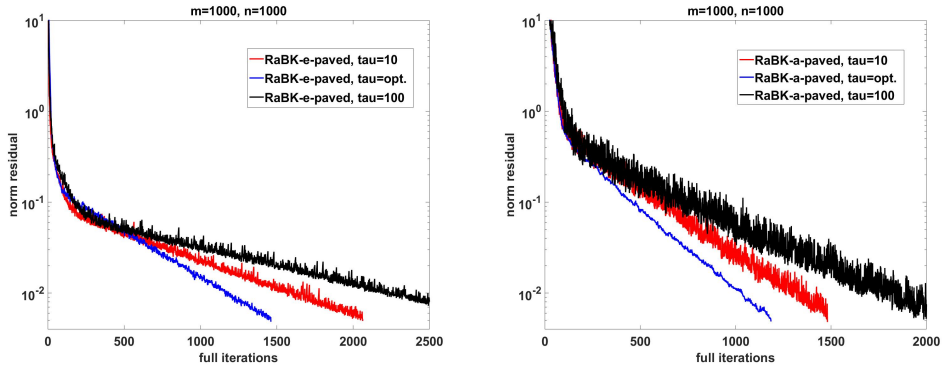

FIG. 3. *Behavior of RaBK-e-paved (left) and RaBK-a-paved (right) on dense data for varying* τ. *Optimal minibatch size* $\tau_{opt} = \lfloor m/\|A\|^2 \rfloor$ *ensures the best convergence compared to other choices of* τ*'s.*

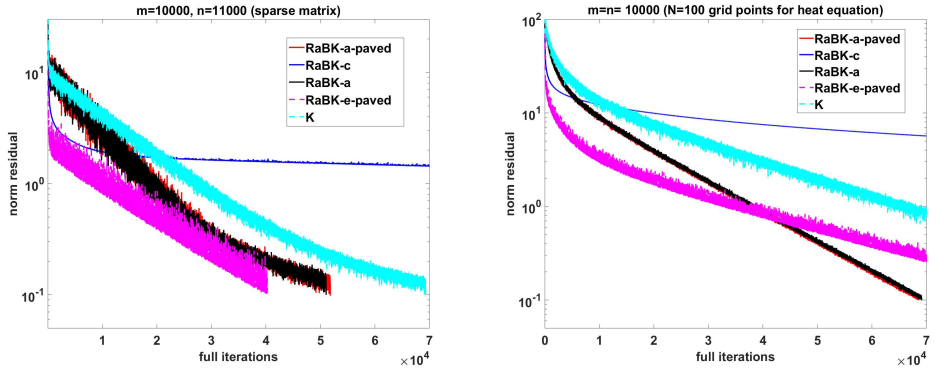

FIG. 4. *Comparison of RaBK variants and K. Left: sparse matrix. Right: discretization of heat equation with* $N = 100$ *grid points.*

In conclusion, besides providing a general framework for the design and analysis of randomized block Kaczmarz methods, our convergence results provide a theoretical understanding of observed practical efficiency of extrapolated block Kaczmarz algorithms. The numerical examples also illustrate the benefits of the new algorithms.

Appendix A. Chebyshev polynomials. In this appendix some properties of the Chebyshev polynomials are briefly reviewed. We refer the reader to, e.g., [28] for more details on Chebyshev polynomials. The Chebyshev polynomials $T_k(x)$, where $\deg(T_k) = k$ and $k \geq 0$, are defined by the recursive relation

$$T_0(x) = 1, \quad T_1(x) = 1, \quad T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x).$$

From the above recurrence we observe that the leading coefficient of $T_k(x)$ is 2^{k-1}, i.e., $T_k(x) = 2^{k-1}x^k +$ lower powers of x. In particular, for $x \in [-1, 1]$, the Chebyshev polynomials can be written equivalently as

$$T_k(x) = \cos(k \arccos(x)).$$

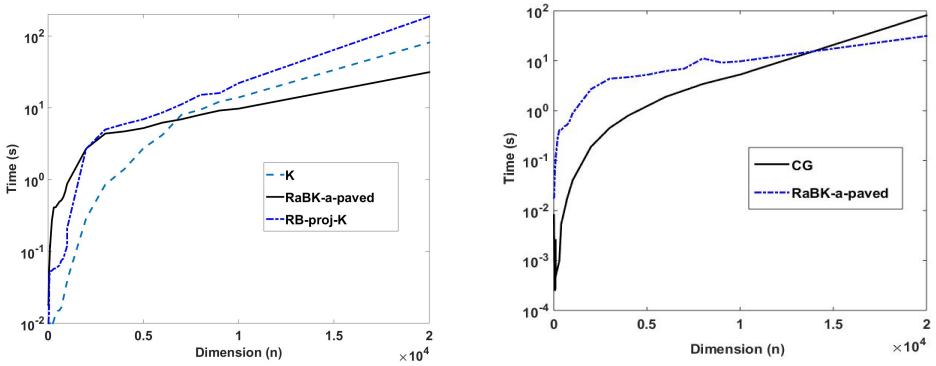

FIG. 5. *Time comparison for variable size n of the square matrix A (stopping criterion* $\|Ax - b\| \leq 10^{-2}$*). Left: K, RaBK-a-paved, and RB-proj-K given in* (1.3) *(see* [24]*) on sparse matrices. Right: RaBK-a-paved and CG on positive definite sparse matrices.*

The equivalence can be verified as follows using that $x = \cos(\theta)$:

$$\begin{aligned} T_k(x) &= 2x\cos((k-1)\arccos(x)) - \cos((k-2)\arccos(x)) \\ &= 2\cos(\theta)\cos((k-1)\theta) - \cos((k-2)\theta) \\ &= \cos(k\theta) + \cos((k-2)\theta) - \cos((k-2)\theta) = \cos(k\theta) = \cos(k\arccos(x)). \end{aligned}$$

It follows that $T_k(1) = 1$. From this representation of $T_k(x)$ it also follows that

$$\max_{x\in[-1,\,1]} |T_k(x)| = 1.$$

Moreover, all the k roots of $T_k(x)$ are given by

$$x_i = \cos\left(\frac{2i-1}{2k}\pi\right) \quad \text{for } \; i = 1:k.$$

In conclusion, we get also the following representation for $T_k(x)$:

$$T_k(x) = 2^{k-1} \cdot \prod_{i=1}^{k} \left(x - \cos\left(\frac{2i-1}{2k}\pi\right)\right).$$

It is also easy to see the interval transformation $[\ell,\ u] \to [-1,\ 1]$ through the relation

$$-1 \leq \frac{2x}{u-\ell} - \frac{u+\ell}{u-\ell} \leq 1 \quad \text{for } \; \ell \leq x \leq u.$$

One important property of the Chebyshev polynomials is that $\frac{1}{2^{k-1}}T_k(x)$ has minimal deviation from 0 among all polynomials of degree k with leading coefficient 1 on $[-1,\ 1]$:

(A.1)

$$\max_{x\in[-1,\,1]} \frac{1}{2^{k-1}}|T_k(x)| \leq \max_{x\in[-1,\,1]} |P_k(x)| \; \forall\, P_k(x) \text{ with leading coefficient 1 and } \deg(P_k) = k.$$

The following lemma is an immediate consequence of the above property valid for Chebyshev polynomials.

LEMMA A.1. *Let* $0 < \ell < u$ *and* $T_k^{(\ell,u)}(x) = T_k\left(\frac{2x}{u-\ell} - \frac{u+\ell}{u-\ell}\right)$. *Then, the optimal value and the optimal polynomial* P_k^* *of the following optimization problem are*

$$\min_{P_k(x):\ deg(P_k)=k, P_k(0)=1} \max_{x\in[\ell,\ u]} |P_k(x)| = \frac{1}{T_k^{(\ell,u)}(0)}$$
$$\leq 2\left(\frac{\sqrt{u}-\sqrt{\ell}}{\sqrt{u}+\sqrt{\ell}}\right)^k \quad \textit{and} \quad P_k^*(x) = \frac{T_k^{(\ell,u)}(x)}{T_k^{(\ell,u)}(0)}.$$

Acknowledgments. The author thanks Yu. Nesterov and F. Glineur from Universite Catholique de Louvain for useful discussions on the Chebyshev-based Kaczmarz scheme.

REFERENCES

[1] H. Bauschke, P. Combettes, and S. Kruk, *Extrapolation algorithm for affine-convex feasibility problems*, Numer. Algorithms, 41 (2006), pp. 239–274.
[2] J. Briskman and D. Needell, *Block Kaczmarz method with inequalities*, J. Math. Imaging Vision, 52 (2015), pp. 385–396.
[3] Y. Censor, *Row-action methods for huge and sparse systems and their applications*, SIAM Rev., 23 (1981), pp. 444–466, https://doi.org/10.1137/1023097.
[4] Y. Censor, W. Chen, P. Combettes, R. Davidi, and G. Herman, *On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints*, Comput. Optim. Appl., 51 (2012), pp. 1065–1088.
[5] F. Deutsch and H. Hundal, *The rate of convergence for the method of alternating projections*, J. Math. Anal. Appl., 205 (1997), pp. 381–405.
[6] T. Elfving, *Block-iterative methods for consistent and inconsistent linear equations*, Numer. Math., 35 (1980), pp. 1–12.
[7] Y. Eldar and D. Needell, *Acceleration of randomized Kaczmarz method via the Johnson-Lindenstrauss lemma*, Numer. Algorithms, 58 (2011), pp. 163–177.
[8] G. Golub and R. Varga, *Chebyshev semi-iterative methods, successive over-relaxation methods and second-order Richardson iterative methods* I, II, Numer. Math., 3 (1961), pp. 147–168.
[9] R. M. Gower and P. Richtárik, *Randomized iterative methods for linear systems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1660–1690, https://doi.org/10.1137/15M1025487.
[10] M. Hanke and W. Niethammer, *On the acceleration of Kaczmarz's method for inconsistent linear systems*, Linear Algebra Appl., 130 (1990), pp. 83–98.
[11] G. Hounsfield, *Computerized transverse axial scanning (tomography): Part* I: *Description of the system*, British J. Radiology, 46 (1973), pp. 1016–1022.
[12] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Int. Acad. Polon. Sci. Lett. A, 35 (1937), pp. 355–357.
[13] U. Khan and J. Moura, *Distributed Kalman filters in sensor networks: Bipartite fusion graphs*, in Proceedings of the 2007 IEEE/SP 14th Workshop on Statistical Signal Processing, 2007, pp. 700–704.
[14] J. Liu and S. Wright, *An accelerated randomized Kaczmarz algorithm*, Math. Comp., 85 (2016), pp. 153–178.
[15] J. Liu, S. Wright, and S. Sridhar, *An Asynchronous Parallel Randomized Kaczmarz Algorithm*, preprint, https://arxiv.org/abs/1401.4780, 2014.
[16] D. Leventhal and A. Lewis, *Randomized methods for linear constraints: Convergence rates and conditioning*, Math. Oper. Res., 35 (2010), pp. 641–654.
[17] X. Lian, Y. Huang, Y. Li, and J. Liu, *Asynchronous parallel stochastic gradient for nonconvex optimization*, in Proceeding of the 28th International Conference on Neural Information Processing Systems (NIPS'15) Volume 2, MIT Press, 2015, pp. 2737–2745.
[18] Y. Li, K. Mo, and H. Ye, *Accelerating random Kaczmarz algorithm based on clustering information*, in Proceeding of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16), AAAI Press, 2016, pp. 1823–1829.

[19] D. A. Lorenz, F. Schöpfer, and S. Wenger, *The linearized Bregman method via split feasibility problems: Analysis and generalizations*, SIAM J. Imaging Sci., 7 (2014), pp. 1237–1262, https://doi.org/10.1137/130936269.
[20] Y. Merzlyakov, *On a relaxation method of solving systems of linear inequalities*, USSR Comput. Math. Math. Phys., 2 (1963), pp. 504–510.
[21] I. Necoara and D. Clipici, *Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds*, SIAM J. Optim., 26 (2016), pp. 197–226, https://doi.org/10.1137/130950288.
[22] I. Necoara, P. Richtárik, and A. Patrascu, *Randomized projection methods for convex feasibility: Conditioning and convergence rates*, SIAM J. Optim., 29 (2019), pp. 2814–2852, https://doi.org/10.1137/18M1167061.
[23] A. Nedić and I. Necoara, *Random minibatch subgradient algorithms for convex problems with functional constraints*, Appl. Math. Optim., 80 (2019), pp. 801–833.
[24] D. Needell and J. Tropp, *Paved with good intentions: Analysis of a randomized block Kaczmarz method*, Linear Algebra Appl., 441 (2014), pp. 199–221.
[25] D. Needell, R. Zhao, and A. Zouzias, *Randomized block Kaczmarz method with projection for solving least squares*, Linear Algebra Appl., 484 (2015), pp. 322–343.
[26] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM J. Optim., 19 (2009), pp. 1574–1609, https://doi.org/10.1137/070704277.
[27] Yu. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362, https://doi.org/10.1137/100802001.
[28] M. A. Olshanskii and E. E. Tyrtyshnikov, eds., *Iterative Methods for Linear Systems: Theory and Applications*, SIAM, 2014, https://doi.org/10.1137/1.9781611973464.
[29] A. Patrascu and I. Necoara, *Nonasymptotic convergence of stochastic proximal point algorithms for constrained convex optimization*, J. Mach. Learn. Res., 18 (2018), pp. 1–42.
[30] G. Pierra, *Decomposition through formalization in a product space*, Math. Programming, 28 (1984), pp. 96–115.
[31] P. Richtárik and M. Takáč, *Stochastic Reformulations of Linear Systems: Algorithms and Convergence Theory*, preprint, https://arxiv.org/abs/1706.01108, 2017.
[32] T. Strohmer and R. Vershynin, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl., 15 (2009), pp. 262–278.
[33] R. Sun and Y. Ye, *Worst-Case Complexity of Cyclic Coordinate Descent: $O(n^2)$ Gap with Randomized Version*, preprint, https://arxiv.org/abs/1604.07130, 2016.
[34] J. Tropp, *Improved analysis of the subsampled randomized Hadamard transform*, Adv. Adapt. Data Anal., 3 (2011), pp. 115–126.
[35] J. A. Tropp, *Column subset selection, matrix factorization, and eigenvalue optimization*, in Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '09), SIAM, 2009, pp. 978–986, https://doi.org/10.1137/1.9781611973068.106.
[36] L. Xiao, S. Boyd, and S. Lall, *A scheme for robust distributed sensor fusion based on average consensus*, in Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005), IEEE Press, 2005, pp. 63–70.
[37] A. Zouzias and N. M. Freris, *Randomized extended Kaczmarz for solving least squares*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 773–793, https://doi.org/10.1137/120889897.