Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas e Faculdade de Engenharia
Programa de Pós-Graduação em Modelagem Computacional

Convergence Analysis and Implementation of Randomized Block Kaczmarz Methods for Linear Systems

Computational Linear Algebra - Final Project

Student: Berilo de Oliveira Santos
Professors: José Jerônimo Camata, Carlos Cristiano

January
2026

Abstract

This work investigates the practical performance of Kaczmarz-type methods for solving consistent linear systems, with particular emphasis on randomized and block variants. Motivated by recent theoretical advances, especially the Randomized Block Kaczmarz (RBK) method proposed by Necoara (2019), we analyze how different algorithmic strategies behave under distinct geometric properties of the system matrix. A set of controlled numerical experiments is designed to isolate the effects of row coherence and spectral conditioning, ensuring that these properties are enforced at the level of matrix geometry rather than through superficial row scaling. The study compares classical, randomized, block, and adaptive block variants across small- and medium-scale test problems. The results demonstrate that block strategies significantly outperform single-row methods in highly correlated systems by exploiting local subspace information and effectively mitigating zig-zag behavior. Conversely, in globally ill-conditioned systems, block methods provide improvement relative to single-row methods but do not fully overcome the limitations imposed by poor spectral conditioning. Furthermore, for consistent and noise-free problems, we observe that exact block projections with fixed step size ($\alpha = 1$) are more efficient than adaptive step-size strategies, which tend to introduce unnecessary under-relaxation. Overall, the findings highlight the importance of matching algorithmic enhancements to the underlying geometric source of difficulty and confirm the practical advantages of randomized block projections in coherent linear systems.

1 Introduction

Solving large-scale systems of linear equations of the form $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, is a fundamental problem in scientific computing, underpinning applications ranging from computerized tomography and signal processing to machine learning and computational fluid dynamics (Saad, 2003; Gordon et al., 1970). In the era of big data, the system matrix A is often too large to fit entirely in memory, rendering direct methods such as Gaussian elimination or QR decomposition computationally prohibitive. As a consequence, iterative row-action methods, which process the matrix one row at a time, have experienced renewed interest.

The Kaczmarz method is an iterative projection-based algorithm for solving linear systems, originally introduced by Kaczmarz (1937). At each iteration, the current approximation is projected onto the solution hyperplane associated with a single equation. Although the classical method with cyclic row selection is straightforward to implement, its convergence rate is highly sensitive to the ordering of the rows. In particular, when consecutive equations define nearly parallel hyperplanes, the method may exhibit extremely slow convergence due to pronounced zig-zagging behavior (Deutsch, 1985). Despite this limitation, the Kaczmarz method remains attractive due to its simplicity, low memory footprint, and clear geometric interpretation, which have led to its widespread use in large-scale and sparse problems, especially in imaging and inverse problems.

For several decades, the method received relatively limited attention, largely overshadowed by the success of direct solvers and Krylov subspace methods. This situation changed significantly with the introduction of the Randomized Kaczmarz (RK) algorithm by Strohmer and Vershynin (2009). By selecting rows randomly with probability proportional to their squared Euclidean norms, the authors established linear convergence in expectation for consistent systems, with a rate governed by the scaled condition number of the matrix. This result not only revived interest in Kaczmarz-type methods but also revealed strong connections between randomized projection algorithms and stochastic gradient descent (SGD) methods used in optimization (Needell et al., 2014).

Despite its favorable theoretical guarantees, the single-row nature of RK limits its ability to exploit modern parallel architectures and does not fully address slow convergence in systems with high row coherence. This observation motivated the development of block variants, in which multiple equations are processed simultaneously at each iteration. Subsequent works explored refinements of the randomized framework, including alternative sampling strategies (Needell and Tropp, 2014), relaxation schemes (Elfving, 1980), and extensions to inconsistent systems (Zouzias and Freris, 2013). These studies emphasized that the convergence behavior of Kaczmarz-type methods is strongly influenced by geometric properties of the system matrix, such as row coherence, scaling, and spectral conditioning. From a computational perspective, block methods also can offer practical advantages by favoring matrix-matrix operations over memory-bound vector operations.

In this context, the Randomized Block Kaczmarz (RBK) method analyzed by Necoara (2019) provides a comprehensive theoretical framework describing how block size, sampling rules, and step-size parameters affect convergence rates. The analysis shows that block projections can significantly accelerate convergence in highly coherent systems by mitigating the classical zig-zag behavior observed in single-row methods. At the same time, it indicates that such improvements are inherently local and may not fully overcome difficulties arising from global spectral ill-conditioning.

Despite the growing theoretical literature, a gap remains between convergence guarantees and observed practical performance, particularly when multiple sources of numerical difficulty coexist. It is often unclear whether performance gains stem primarily from improved handling of row correlation, better conditioning of local subproblems, or increased per-iteration computational effort. Motivated by these considerations, this work investigates the practical behavior of Kaczmarz-type algorithms, with a focus on the Randomized Block Kaczmarz method (Necoara, 2019), in comparison with classical and randomized single-row variants. Inspired by the benchmark constructions of Lokrantz (2019), we conduct controlled numerical experiments on Gaussian, highly correlated, and ill-conditioned systems, aiming to quantify the trade-offs between convergence speed and computational cost and to identify the regimes in which block strategies provide a decisive advantage.

2 Preliminaries and Problem Formulation

2.1 Linear Systems and Hyperplanes

We consider the system of linear equations

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ is the coefficient matrix with rows $\mathbf{a}_1^\top, \dots, \mathbf{a}_m^\top$, and $\mathbf{b} \in \mathbb{R}^m$ is the observation vector. Throughout this work, we assume that the system is consistent, meaning that there exists at least one solution $\mathbf{x}^* \in \mathbb{R}^n$ such that $A\mathbf{x}^* = \mathbf{b}$.

From a geometric perspective, each equation $\langle \mathbf{a}_i, \mathbf{x} \rangle = b_i$ defines an affine hyperplane

$$H_i := \{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i\}, \tag{2}$$

whose normal direction is given by the row vector $\mathbf{a}_i$. The solution set of the linear system can therefore be expressed as the intersection of these hyperplanes:

$$\mathcal{X}^* = \bigcap_{i=1}^{m} H_i. \tag{3}$$

Kaczmarz-type methods exploit this geometric structure by iteratively projecting a current estimate $\mathbf{x}_k$ onto one or more hyperplanes. In its simplest form, each iteration enforces a single equation exactly, generating a sequence of iterates that gradually approaches the intersection $\mathcal{X}^*$. Randomized and block variants differ primarily in how these hyperplanes are selected and combined at each step, but they all share this common projection-based interpretation.

2.2 Geometric Sources of Difficulty

The convergence behavior of projection methods is dictated not merely by the size of the system, but by the geometric arrangement of the hyperplanes $\{H_i\}_{i=1}^m$. In this work, we distinguish between two fundamentally different sources of numerical difficulty: global spectral ill-conditioning and local row coherence.

Conditioning (Global Geometry) The global geometry of the problem is governed by the singular value spectrum of the matrix A. The condition number

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} \tag{4}$$

measures the anisotropy of the linear transformation induced by A. When $\kappa(A)$ is large, the level sets of the least-squares objective function $\|A\mathbf{x} - \mathbf{b}\|^2$ become highly elongated ellipsoids.

In such settings, progress along directions associated with small singular values is intrinsically slow. Projection-based methods, including Kaczmarz-type algorithms, tend to exhibit oscillatory behavior across narrow valleys of the solution landscape, requiring many iterations to reduce error components aligned with poorly conditioned directions. Importantly, this difficulty is global in nature and persists regardless of the order in which rows are selected or whether projections are performed sequentially or in blocks.

Row Coherence (Local Geometry) In contrast to spectral conditioning, row coherence describes a local geometric property of the system. It quantifies the degree of alignment between pairs (or groups) of rows of A. A common measure of coherence between two rows $\mathbf{a}_i$ and $\mathbf{a}_j$ is given by the normalized inner product

$$\mu_{ij} = \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{\|\mathbf{a}_i\| \, \|\mathbf{a}_j\|}. \tag{5}$$

High values of μ_{ij} indicate that the corresponding hyperplanes H_i and H_j intersect at a small angle.

In highly coherent systems, sequential projections onto nearly parallel hyperplanes produce only marginal progress toward the intersection. After projecting onto H_i, the subsequent projection onto H_j largely cancels the previous step, resulting in the well-known zig-zag phenomenon. Crucially, this effect is not eliminated by row normalization, as coherence depends on angular relationships rather than scaling.

Implications for Block Methods Block Kaczmarz methods are specifically designed to address the difficulties induced by row coherence. By projecting onto the intersection of multiple hyperplanes simultaneously, these methods effectively orthogonalize correlated directions within each block. From a geometric standpoint, a block projection replaces a sequence of short, poorly aligned steps by a single projection onto a higher-dimensional affine subspace, thereby suppressing local zig-zag behavior.

However, while block strategies can significantly accelerate convergence in coherent systems, they do not fundamentally alter the global spectral properties of A in the sense of preconditioning. As a result, when slow convergence is primarily driven by severe ill-conditioning, block projections alone are insufficient to overcome the inherent anisotropy of the problem. This distinction highlights the importance of identifying the dominant geometric source of difficulty when designing or selecting algorithmic enhancements.

These observations motivate the experimental framework adopted in this work, where coherence and conditioning are controlled independently. This separation allows for a clearer assessment of the regimes in which randomized and block Kaczmarz variants provide genuine advantages over classical approaches.

3 Kaczmarz-Type Methods

In this section, we briefly review the Kaczmarz method and its randomized and block variants, emphasizing their geometric interpretation and their connection to optimization-based viewpoints. Throughout, we assume that the linear system $A\mathbf{x} = \mathbf{b}$ is consistent.

3.1 Classical and Randomized Kaczmarz

The classical Kaczmarz method is an iterative projection algorithm that enforces one equation of the system at a time. Given a current iterate $\mathbf{x}_k$ and a selected row $\mathbf{a}_i^\top$, the next iterate is obtained by solving the projection problem

$$\min_x \|\mathbf{x} - \mathbf{x}_k\|_2^2 \quad \text{s.t.} \quad \mathbf{a}_i^\top \mathbf{x} = b_i,$$

which yields the closed-form update

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{b_i - \mathbf{a}_i^\top \mathbf{x}_k}{\|\mathbf{a}_i\|_2^2} \mathbf{a}_i. \tag{6}$$

In the classical algorithm, rows are processed cyclically. While simple, this deterministic strategy may lead to extremely slow convergence when consecutive hyperplanes are nearly parallel, as discussed in Section 2.

The Randomized Kaczmarz (RK) method replaces cyclic selection by random sampling. Following Strohmer and Vershynin (2009), rows are typically sampled with probability

$$\mathbb{P}(i) = \frac{\|\mathbf{a}_i\|_2^2}{\|A\|_F^2},$$

which leads to linear convergence in expectation. From an optimization perspective, RK can be interpreted either as stochastic gradient descent applied to the least-squares objective

$$f(x) = \frac{1}{2m}\|A\mathbf{x} - \mathbf{b}\|_2^2,$$

or equivalently as randomized coordinate descent applied to the dual problem. This interpretation clarifies the role of the normalization by $\|\mathbf{a}_i\|_2^2$ and explains why a constant step size $\alpha = 1$ corresponds to an exact orthogonal projection.

3.2 Randomized Block Kaczmarz

While RK improves robustness with respect to row ordering, it still processes a single equation per iteration and therefore remains sensitive to strong row coherence. The Randomized Block Kaczmarz (RBK) method generalizes RK by projecting onto multiple hyperplanes simultaneously.

At iteration k, a random subset of row indices $J_k \subseteq \{1, \dots, m\}$ is selected according to a prescribed sampling distribution. Let A_{J_k} and $\mathbf{b}_{J_k}$ denote the corresponding submatrix and subvector. The block projection step is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + A_{J_k}^{\dagger}(\mathbf{b}_{J_k} - A_{J_k}\mathbf{x}_k), \tag{7}$$

where $A_{J_k}^{\dagger}$ denotes the Moore-Penrose pseudoinverse.

Geometrically, this update corresponds to the orthogonal projection of $\mathbf{x}_k$ onto the intersection of the hyperplanes associated with the block J_k. By resolving correlations within each block, RBK suppresses the local zig-zag behavior that plagues single-row methods in highly coherent systems.

From an optimization viewpoint, RBK can be interpreted as a batch stochastic gradient or block coordinate descent method applied to the primal or dual least-squares formulations, depending on the choice of weights within the block. This connection was formalized and extensively analyzed by Necoara (2019), who showed that RBK admits linear convergence rates that depend explicitly on the geometric properties of both the full matrix A and its submatrices A_{J_k}.

3.3 Relaxation and Step-Size Choices

A general form of both RK and RBK updates can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{8}$$

where $\mathbf{d}_k$ is the projection direction and $\alpha_k > 0$ is a relaxation parameter. For $\alpha_k = 1$, the update corresponds to an exact orthogonal projection. Values $\alpha_k \in (0, 2)$ are commonly considered in the literature to ensure convergence, while $\alpha_k > 2$ corresponds to extrapolated steps.

In the context of consistent, noise-free linear systems, the choice $\alpha_k = 1$ is particularly natural: the exact projection minimizes the distance to the solution set along the chosen subspace and does not introduce unnecessary damping. For this reason, $\alpha = 1$ is used as the baseline step size in our numerical experiments.

3.4 Adaptive Step-Size RBK

Necoara (2019) further proposes an adaptive step-size strategy for RBK, designed to approximate online an optimal extrapolated step size without explicitly computing spectral quantities of the block matrices. At each iteration, the step size α_k is chosen based on local curvature information derived from the current residual and the selected block.

From a theoretical standpoint, this adaptive strategy is appealing, as it preserves linear convergence guarantees and can outperform fixed step-size schemes in more general settings, including inconsistent or noisy systems. However, in the specific regime considered in this work-consistent linear systems with exact arithmetic-the adaptive rule often leads to conservative updates. Since the exact block projection already provides the optimal correction along the selected subspace, reducing the step size below $\alpha = 1$ results in unnecessarily small progress per iteration.

Consistent with this observation, our numerical experiments show that adaptive RBK typically underperforms standard RBK with $\alpha = 1$ in noise-free settings, while incurring additional computational overhead. For this reason, adaptive strategies are included only for comparative purposes and are not pursued further in the most computationally demanding scenarios.

4 Experimental Setup and Methodology

This section details the computational implementation, performance metrics, and the suite of benchmark problems designed to evaluate the Kaczmarz-type algorithms. The experimental design aims to isolate specific geometric properties, such as row coherence and spectral conditioning, to quantify the advantages of block and randomized strategies in different regimes.

4.1 Implementation Details

The algorithms were implemented in `Python` leveraging the `NumPy` and `SciPy` libraries for high-performance numerical linear algebra. The framework is modular, supporting flexible configuration of row selection strategies (cyclic vs. randomized) and update rules (single-row vs. block). The complete source code is available at `https://github.com/beriloosantos/computational_linear_algebra`.

The central component of the Randomized Block Kaczmarz (RBK) method is the solution of the underdetermined local least-squares problem on the selected block of rows. For a block index set J_k, the update direction d_k is the minimum-norm solution to:

$$A_{J_k}\mathbf{d}_k = \mathbf{b}_{J_k} - A_{J_k}\mathbf{x}_k. \tag{9}$$

Numerical stability is prioritized over raw speed; thus, we utilize `NumPy`'s `lstsq` driver, which employs a robust decomposition (SVD or QR), avoiding the explicit and potentially unstable computation of the Moore-Penrose pseudoinverse.

```
# Extract block submatrices
A_block = A[indices]
b_block = b[indices]

# Compute local residual
r = b_block - A_block @ x

# Solve for minimum-norm direction using robust least-squares
direction, _, _, _ = np.linalg.lstsq(A_block, r, rcond=None)

# Update iterate (alpha=1 for exact projection)
x = x + alpha * direction
```

Listing 1: Python implementation of the RBK update step

Experiments were conducted using double-precision arithmetic. Consistent with the theoretical focus on consistent systems, a fixed relaxation parameter $\alpha = 1$ (exact projection) was employed unless otherwise noted.

4.2 Performance Metrics

To evaluate algorithmic efficiency, we employ two complementary metrics:

1. **Relative Residual Norm:** Defined as $r_k = \|A\mathbf{x}_k - \mathbf{b}\|_2 / \|A\mathbf{x}_0 - \mathbf{b}\|_2$. This scale-invariant metric tracks the reduction in approximation error and serves as the primary stopping criterion.

2. **Computational Cost:** We report both *iteration counts* and *wall-clock time*. While iteration counts provide a hardware-independent measure of algorithmic complexity, wall-clock time is crucial for assessing the trade-off between the faster convergence of block methods and their higher per-iteration computational cost.

4.3 Benchmark Problems

The numerical experiments are inspired by the benchmark design proposed in Lokrantz (2019). We consider three controlled scenarios, each targeting a distinct geometric source of difficulty. In all cases, the system is constructed to be consistent.

Preliminary Validation: Low-Dimensional Example As an initial sanity check, all methods were tested on a small 4×2 linear system commonly used for geometric visualization (Scipython, 2018). Although such a problem is too small to draw conclusions about asymptotic convergence, it provides qualitative insight into the projection behavior of each method.

In this setting, classical and randomized Kaczmarz methods exhibit characteristic zig-zag trajectories when rows are nearly aligned. Introducing relaxation ($\alpha < 1$) further slows convergence by deliberately understepping the projection distance. In contrast, the block method, with block size equal to the number of rows, projects directly onto the solution in a single iteration, illustrating the geometric advantage of resolving multiple constraints simultaneously.

Scenario A: Gaussian Systems (Baseline) This scenario represents a "well-behaved" random regime.

- **Construction:** Elements of $A \in \mathbb{R}^{m \times n}$ are drawn i.i.d. from a standard normal distribution $\mathcal{N}(0, 1)$.

- **Characteristics:** These matrices exhibit low row coherence and benign condition numbers.

- **Hypothesis:** Standard randomized methods should perform efficiently. Block methods are expected to reduce iterations, though potentially with diminishing returns regarding total time due to overhead.

Scenario B: Highly Correlated Systems (Coherence) This scenario targets the specific weakness of Kaczmarz methods: correlated rows.

- **Construction:** A Gaussian matrix is modified by adding a scaled random direction $v \in \mathbb{R}^n$ to every row: $\mathbf{a}_i \leftarrow \mathbf{a}_i + \eta \mathbf{v}$, where η controls correlation strength.
- **Characteristics:** The hyperplanes become nearly parallel (high coherence), creating narrow "valleys" in the optimization landscape.
- **Hypothesis:** Single-row methods will exhibit severe stagnation (zig-zagging). Block methods should theoretically "whiten" this local correlation, recovering fast convergence.

Scenario C: Ill-Conditioned Systems (Spectral) This scenario introduces global spectral difficulty independent of local row alignment.

- **Construction:** We generate A with a predefined spectrum σ_i decaying exponentially, resulting in a high condition number $\kappa(A) \gg 1$.
- **Characteristics:** The solution manifold is an extremely elongated ellipsoid.
- **Hypothesis:** Unlike row correlation, global ill-conditioning is intrinsic. We expect block methods to offer only marginal improvements over standard methods, as the difficulty is global rather than local.

5 Results and Discussion

5.1 Visual Trajectory Analysis

To provide geometric intuition into the behavior of the different algorithms, we first analyze the solution trajectories for a low-dimensional (4×2) consistent linear system, using a stopping tolerance of $\varepsilon < 10^{-6}$. This setting allows for direct visualization of the iterates in the solution space and highlights qualitative differences between projection strategies.

As shown in Figure 1, the classical Kaczmarz method exhibits the characteristic zig-zag behavior. This phenomenon arises because the method projects sequentially onto hyperplanes whose normals may be poorly aligned with the current error direction, particularly when the angle between consecutive hyperplanes is small (Strohmer and Vershynin, 2009). As a result, successive projections correct the error in nearly orthogonal directions, leading to oscillatory trajectories and slow progress toward the intersection.

The randomized Kaczmarz method alters the order of projections through random row selection, which changes the trajectory path but does not fundamentally eliminate the geometric limitation imposed by row correlation. This effect is visible in Figure 1 (middle), where the path becomes less structured but still oscillatory.

In contrast, the Randomized Block Kaczmarz method with block size $\tau = 2$ selects both equations simultaneously and projects directly onto the intersection of the corresponding hyperplanes. In this particular example, this results in convergence in a single iteration. Although this represents an idealized scenario, it clearly illustrates the geometric advantage of block projections: by resolving multiple constraints at once, RBK eliminates local zig-zag artifacts and performs a direct step toward the solution subspace.

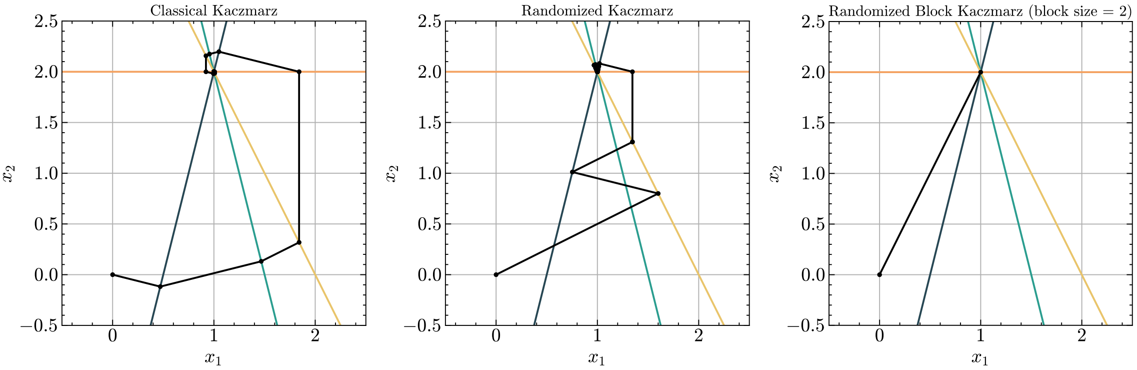

Figure 1: Trajectory comparison in the 2D solution space. The classical method (left) and the randomized method (middle) exhibit oscillatory behavior, while the block method (right) projects directly onto the intersection of the hyperplanes.

Figure 2 illustrates the effect of relaxation in the classical Kaczmarz method. When a relaxation parameter $\alpha = 0.8$ is used, each update moves in the same general direction as the exact projection but deliberately understeps the hyperplane intersection. As a consequence, the iterates no longer land exactly on the selected hyperplane at each iteration, leading to slower convergence.

While over-relaxation ($\alpha > 1$) may accelerate convergence in certain problems, it can also be counterproductive and induce instability, depending on the system geometry. Similar effects are observed when relaxation is applied to the randomized and block variants. This sensitivity to problem characteristics motivates the development of adaptive step-size strategies, such as the adaptive RBK method, which aim to balance stability and progress dynamically.

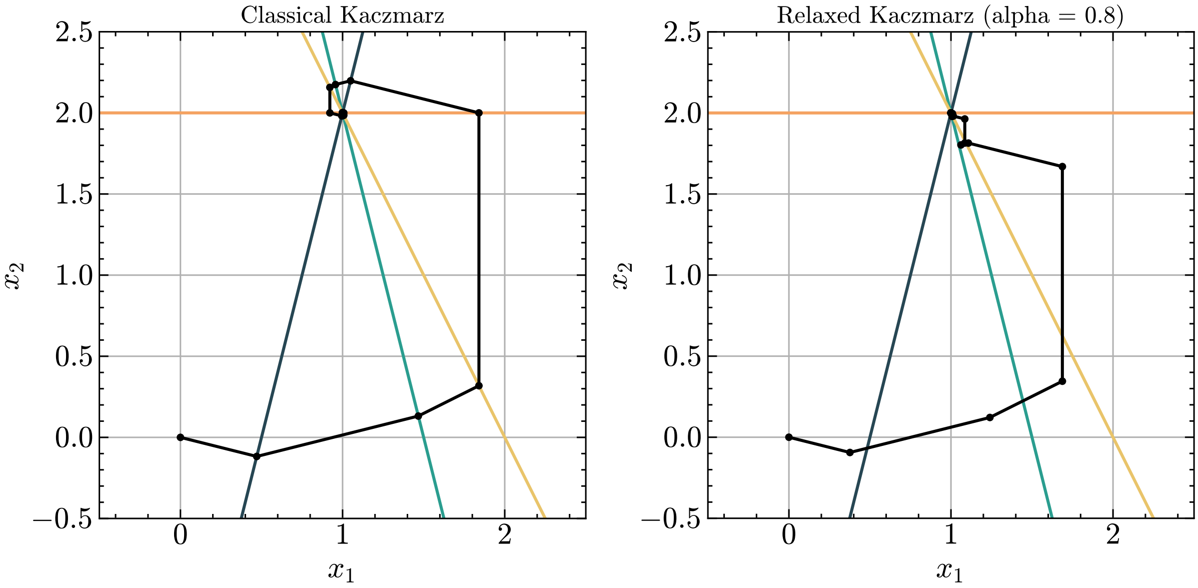

Figure 2: Trajectory comparison between the classical Kaczmarz method (left) and the relaxed Kaczmarz method with $\alpha = 0.8$ (right).

Finally, Figure 3 presents the convergence histories of all method variants tested in this preliminary example. As expected, RBK converges in a single iteration, achieving the fastest convergence in terms of iteration count. The detrimental effect of relaxation is also evident, as relaxed variants of all methods require more iterations to reach the prescribed tolerance.

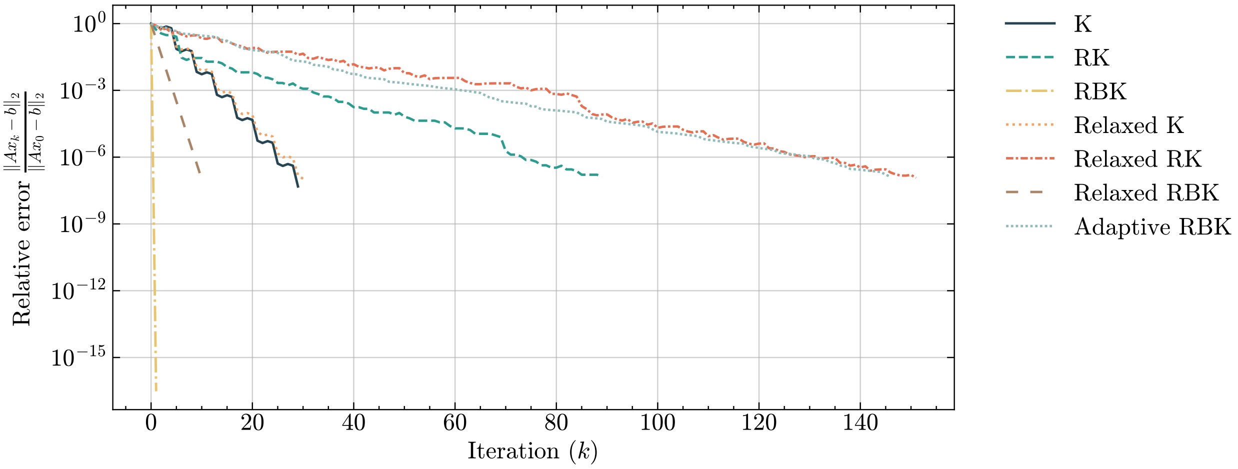

Figure 3: Convergence comparison for low-dimensional linear system.

The adaptive RBK method performs poorly in this particular setting, which is small-scale and well-conditioned, offering little opportunity for adaptive step-size corrections to be beneficial. Interestingly, the randomized Kaczmarz method converges more slowly than the classical variant in this example, despite exhibiting less structured trajectories. While classical Kaczmarz converges relatively quickly, it does so with the most oscillatory path among the tested methods.

5.2 Quantitative Convergence Analysis

We now assess the performance of the proposed methods on higher-dimensional systems, focusing on two regimes: a small-scale validation problem (10×10) and a medium-scale test designed to evaluate scalability and geometric robustness (500×100). To avoid excessive visual clutter, convergence plots are shown for a representative scenario only, while numerical results for all cases are summarized in tables.

5.2.1 Small-Scale Validation

Table 1 reports the numerical results for the 10×10 systems using a full block size ($\tau = 10$). Even at this modest scale, the superiority of block-based strategies is evident across all system types.

Table 1: Performance Comparison for Small Scale 10×10 Systems ($\tau = 10$).

System Type	Method	Iterations	Time [s]	Final Error
Gaussian	K	140,430	1.1519	9.99×10^{-7}
	RK	299,897	6.3078	9.83×10^{-7}
	RBK	12,649	0.4133	9.68×10^{-7}
	RBK-Adaptive	196,392	9.0073	9.99×10^{-7}
Correlated	K	6,109	0.0608	9.92×10^{-7}
	RK	10,712	0.2384	9.96×10^{-7}
	RBK	195	0.0074	9.85×10^{-7}
	RBK-Adaptive	683	0.0323	9.61×10^{-7}
Ill-conditioned	K	496,180	4.7233	1.00×10^{-6}
	RK	685,827	16.603	9.99×10^{-7}
	RBK	1,706	0.0622	7.94×10^{-7}
	RBK-Adaptive	8,435	0.4132	9.89×10^{-7}

In the correlated scenario, the RBK method reduces the iteration count by approximately a factor of 30 compared to the classical Kaczmarz method. This improvement directly reflects the ability of block projections to resolve local row coherence, eliminating the zig-zag behavior that severely hampers single-row updates.

For Gaussian systems, all methods converge reliably; however, the computational overhead associated with block least-squares solves leads to smaller relative gains. In this regime, the adaptive RBK method performs noticeably worse than both the standard RBK and the classical method, indicating that adaptive under-relaxation is unnecessary for well-conditioned, noise-free systems.

In the ill-conditioned case, single-row methods exhibit pronounced instability and require several orders of magnitude more iterations to converge. In contrast, block-based methods remain stable and converge significantly faster, suggesting that block projections partially mitigate, though do not fully eliminate, the effects of unfavorable global geometry.

Figure 4 illustrates the convergence history for the correlated 10×10 system. The RBK method clearly outperforms all alternatives, achieving rapid and stable convergence. The adaptive RBK method converges more slowly due to systematic under-relaxation, while the classical and randomized Kaczmarz methods display oscillatory behavior and significantly longer convergence times.

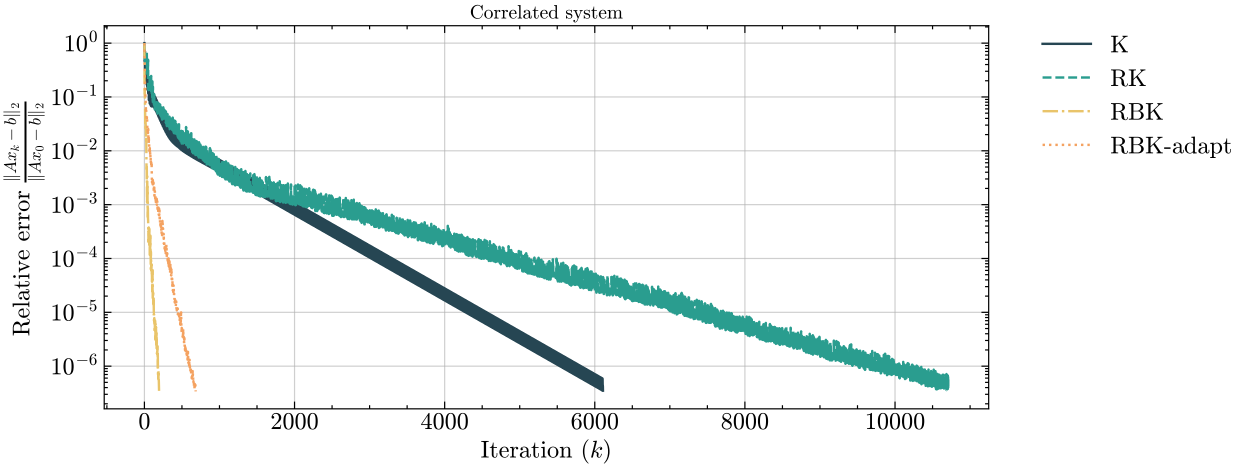

Figure 4: Convergence comparison for the correlated 10×10 system.

Although only the correlated case is shown, similar qualitative trends were observed in the Gaussian and ill-conditioned scenarios. In Gaussian systems, both RBK variants exhibit mild instability due to over-solving well-conditioned subproblems, whereas in ill-conditioned systems, single-row methods become highly erratic while block methods maintain stable descent.

5.2.2 Scalability and Geometric Robustness

The results for the larger 500×100 systems, summarized in Table 2, further confirm the theoretical predictions of Necoara (2019). As the problem dimension increases, the performance gap between block and single-row methods becomes even more pronounced.

In the correlated scenario, both classical and randomized Kaczmarz methods stagnate due to severe row coherence. In contrast, RBK with block size $\tau = 50$ converges in only 59 iterations, compared to over 4,300 iterations for the classical method. This behavior is consistent with the improvement of the stochastic conditioning parameter $\lambda_{\max}^{\text{block}}$ when correlated rows are grouped, allowing for substantially larger effective step sizes.

For Gaussian systems, all methods scale reasonably well, though RBK maintains a clear advantage in iteration count. In ill-conditioned systems, single-row methods become prohibitively slow, whereas block methods remain computationally feasible, despite the intrinsic difficulty imposed by the global spectrum.

Table 2: Performance Comparison for 500×100 Systems ($\tau = 50$).

System Type	**Method**	**Iterations**	**Time [s]**	**Final Error**
Gaussian	K	4,227	0.0721	9.95×10^{-7}
	RK	5,871	0.2601	9.99×10^{-7}
	RBK	66	0.0105	7.96×10^{-7}
	RBK-Adaptive	17,678	4.1099	9.99×10^{-7}
Correlated	K	4,336	0.0862	1.00×10^{-6}
	RK	5,302	0.2096	9.74×10^{-7}
	RBK	59	0.0071	8.46×10^{-7}
	RBK-Adaptive	421	0.0859	9.77×10^{-7}
Ill-Conditioned	K	1,604,860	28.255	9.98×10^{-7}
	RK	1,629,045	81.251	9.96×10^{-7}
	RBK	1,371	0.1384	9.57×10^{-7}
	RBK-Adaptive	90,396	22.927	1.00×10^{-6}

Figure 5 shows the convergence curves for the correlated 500×100 system. Compared to the small-scale case, the trajectories exhibit fewer oscillations overall, yet the relative performance ordering of the methods remains unchanged.

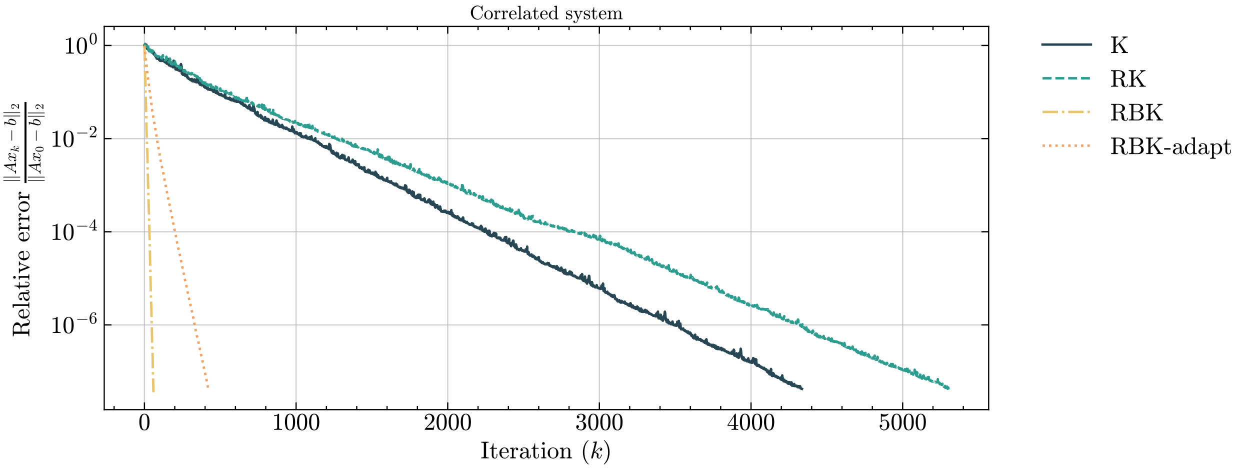

Figure 5: Convergence comparison for the correlated 500×100 system.

5.3 Analysis of Adaptive Step-Size Performance

An important observation from Tables 1 and 2 is the consistently inferior performance of the adaptive RBK method when compared to the standard RBK with fixed step size $\alpha = 1$. Although Necoara (2019) proves that the adaptive step-size rule guarantees convergence and is essential for inconsistent or noisy systems, all experiments in this work focus on consistent linear systems, where an exact solution exists.

In this setting, the optimal update corresponds to an exact orthogonal projection onto the block intersection, which is achieved with $\alpha = 1$. The adaptive strategy computes a step size based on local curvature / spectral information to ensure robustness, often resulting in $\alpha_k < 1$ (see Eq. (4.6) in Necoara (2019)). In noise-free problems, this conservatism is unnecessary and leads to systematic under-relaxation, increasing the iteration count without improving stability.

As a result, the additional computational overhead required to compute the adaptive step size, combined with smaller update magnitudes, renders the adaptive RBK method less efficient than the standard block projection in the scenarios considered.

5.3.1 Block Size Study

The block size τ plays a central role in the performance of the RBK method. Increasing τ generally accelerates convergence by enforcing more constraints simultaneously, thereby reducing the impact of row correlation and improving the quality of the projection direction.

However, larger blocks also increase the per-iteration computational cost due to the solution of larger local least-squares problems. Figure 6 illustrates this trade-off: moderate block sizes achieve the best balance between iteration count, total runtime, and convergence stability.

These results suggest that block size selection should account for both the degree of row correlation and available computational resources. Extremely large blocks may offer diminishing returns, while overly small blocks may fail to capture the underlying geometric structure of the problem.

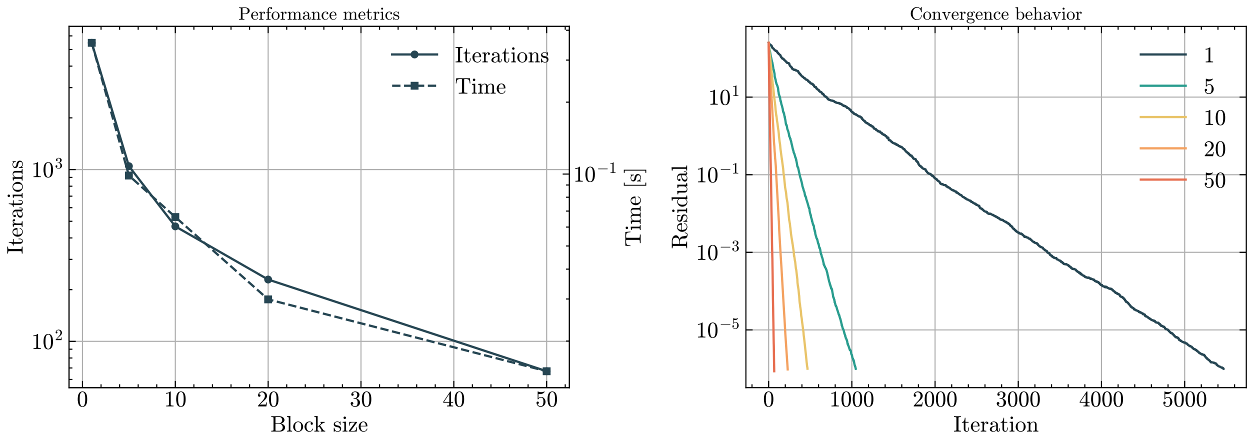

Figure 6: Iteration count and runtime as functions of block size (left). Residual convergence behavior (right).

6 Conclusion

This work presented a detailed numerical study of Kaczmarz-type methods for solving consistent linear systems, with particular emphasis on the Randomized Block Kaczmarz (RBK) algorithm analyzed by Necoara (2019). By designing controlled experiments that isolate distinct geometric sources of numerical difficulty, we were able to bridge theoretical convergence results with observed practical performance.

The experimental results support the following main conclusions:

1. **Effectiveness in Correlated Systems:** The Randomized Block Kaczmarz method significantly outperforms single-row variants in systems with high row coherence. By enforcing multiple constraints simultaneously, RBK exploits local subspace information and effectively suppresses the classical zig-zag behavior. In both small- and medium-scale experiments, this resulted in reductions of up to two orders of magnitude in iteration count when compared to classical and randomized single-row methods.

2. **Intrinsic Limits of Local Projections:** While block projections substantially improve convergence in coherent systems, they do not fully overcome global spectral ill-conditioning. In ill-conditioned scenarios, RBK consistently outperformed single-row methods but still required significantly more iterations than in well-conditioned Gaussian systems. This confirms that projection-based methods remain fundamentally constrained by the global condition number $\kappa(A)$, and that block strategies primarily address local geometric difficulties.

3. **Role of Relaxation and Adaptivity:** For the consistent, noise-free systems considered in this work, exact block projections ($\alpha = 1$) were found to be optimal. Adaptive step-size strategies, although theoretically essential for robustness in inconsistent or noisy problems, led to systematic under-relaxation and increased computational cost in ideal settings. Consequently, the adaptive RBK variant consistently underperformed the standard RBK method in all scenarios examined.

From an applied perspective, these findings are particularly relevant for problems in computational modeling, such as flow in porous media, tomographic reconstruction, and large-scale inverse problems, where discretization often produces highly correlated linear systems. In such settings, block Kaczmarz methods provide a robust and computationally attractive alternative to classical row-action schemes. Moreover, the block structure naturally exposes opportunities for parallelism, as local least-squares solves are independent of the global system size.

Future work should focus on extending this analysis to inconsistent and noisy systems, where the adaptive step-size rules proposed by Necoara (2019) are expected to play a decisive role. Additional promising directions include the use of RBK as a smoother within multigrid frameworks, hybridization with preconditioning techniques to address global ill-conditioning, and efficient implementations on modern parallel architectures such as GPUs.

References

Deutsch, F. (1985). Rate of convergence of the method of alternating projections. *International Series of Numerical Mathematics*, 72:96–107.

Elfving, T. (1980). Block-iterative methods for consistent and inconsistent linear equations. *Numerische Mathematik*, 35(1):1–12.

Gordon, R., Bender, R., and Herman, G. T. (1970). Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography. *Journal of theoretical biology*, 29(3):471–481.

Kaczmarz, S. (1937). Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques*, 35:355–357.

Lokrantz, C. (2019). *A Review of the Kaczmarz Method*. Bachelor's thesis, KTH Royal Institute of Technology.

Necoara, I. (2019). Faster randomized block kaczmarz algorithms. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1425–1452.

Needell, D., Srebro, N., and Ward, R. (2014). Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Mathematical Programming*, 155:549–573.

Needell, D. and Tropp, J. A. (2014). Paved with good intentions: analysis of a randomized block kaczmarz method. *Linear Algebra and its Applications*, 441:199–221.

Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM.

Scipython (2018). Visualizing kaczmarz's algorithm. `https://scipython.com/blog/visualizing-kaczmarzs-algorithm/`. Accessed: 2026-01-08.

Strohmer, T. and Vershynin, R. (2009). A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278.

Zouzias, A. and Freris, N. M. (2013). Randomized extended kaczmarz for solving least squares. *SIAM Journal on Matrix Analysis and Applications*, 34(2):773–793.