J. Nonlinear Funct. Anal. 2022 (2022) 2 https://doi.org/10.23952/jnfa.2022.2

MATHRES

Journal of Nonlinear Functional Analysis

Available online at http://jnfa.mathres.org

SOME IMPROVEMENTS OF RANDOMIZED KACZMARZ ALGORITHMS

SONGNIAN HE[1], ZITING WANG[1], XIAOLONG QIN[2,*]

[1]College of Science, Civil Aviation University of China, Tianjin, China

[2]National Yunlin University of Science and Technology, Douliou, Taiwan

Abstract. In this paper, we first introduce a modified randomized Kaczmarz algorithm, and obtain a better convergence rate estimate than that of the randomized Kaczmarz algorithm. Second, we study a greedy Kaczmarz algorithm, which can be regarded as a simplification of the greedy randomized Kaczmarz algorithm. A deterministic (not in the sense of expectation) convergence rate estimate is obtained for the greedy Kaczmarz algorithm. We find that the greedy Kaczmarz algorithm not only needs less computational workload in each iteration, but also has faster convergence speed than the greedy randomized Kaczmarz algorithm. Numerical results are provided to support the theoretical analysis in this paper.

Keywords. Convergence rate; Greedy randomized Kaczmarz algorithm; Projection; Randomized Kaczmarz algorithm; System of linear equations.

1. Introduction

Consider a large-scale linear system

$$Ax = b, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$ is the coefficient matrix with the row vectors A_i, $i = 1,2,\cdots,m$, $b = (b_1, b_2, \cdots, b_m)^\top \in \mathbb{R}^m$ is a known column vector, and $x = (x_1, x_2, \cdots, x_n)^\top \in \mathbb{R}^n$ is the unknown column vector, respectively. It is known that each equation $A_i x = b_i$, $i = 1,2,\cdots,m$, of (1.1) represents a hyperplane:

$$H_i = \{x \in \mathbb{R}^n \,|\, A_i x = b_i\}, \quad i = 1,2,\cdots,m.$$

Throughout this paper, we always assume that problem (1.1) is consistent, that is, its solution set, denoted by $\mathscr{S}$, is nonempty. For convenience, we will use the following notations:

- $\mathscr{I} = \{1,2,\cdots,m\}$ denotes an index set.
- $\lambda_{\min}$ denotes the minimum nonzero eigenvalue of $A^\top A$.
- $\|A\|_F$ denotes the Frobenius norm of A.

*Corresponding author.

E-mail address: qinxl@yuntech.edu.tw (X. Qin).

Received June 2, 2021; Accepted November 25, 2021.

- $\alpha_F(A) = \max_{i\in\mathscr{I}}\{\|A\|_F^2 - \|A_i\|^2\}$ denotes a positive constant.
- $x^\dagger$ denotes the minimum norm solution of problem (1.1).
- $P_i : \mathbb{R}^n \to H_i$ denotes the orthogonal projection operator from $\mathbb{R}^n$ onto H_i.

The Kaczmarz algorithm (see, e.g., [3, Chapter 8] and [6]), which was often used to solve (1.1), starts with an initial guess x^0 selected in $\mathbb{R}^n$ arbitrarily and generates a sequence $\{x^k\}_{k=0}^\infty$ by the iteration step:

$$x^{k+1} = P_{i(k)}x^k,$$

where $i(k) = (k \mod m) + 1$. It was proved [6] that $\{x^k\}_{k=0}^\infty$ converges to the solution, which is closest to x^0. However, it is difficult to estimate the convergence rate for the Kaczmarz algorithm.

In 2009, Strohmer and Vershynin [8] proposed the following randomized Kaczmarz (RK) algorithm.

Algorithm 1.1 Randomized Kaczmarz Algorithm

Input: A, b and ℓ(the maximum number of iteration steps)
Output: x^ℓ
Step 0: Take $x^0 \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ arbitrarily.
for $k = 0, 1, \cdots, \ell - 1$ **do**
Step 1: Select an index $i_k \in \mathscr{I}$ with probability $p_{i_k} = \frac{\|A_{i_k}\|^2}{\|A\|_F^2}$ and calculate $x^{k+1} = P_{i_k}x^k$.
endfor

Recently, Strohmer and Vershynin [8], Ma, Needell, and Ramdas, [7], and Gower and Richtárik [4] studied the convergence of Algorithm 1.1, respectively. From these results, we have the following fact.

Let the initial guess x^0 be selected in $\operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ arbitrarily. Then the sequence $\{x^k\}_{k=0}^\infty$ generated by Algorithm 1.1 converges linearly in expectation to $x^\dagger$ for linear system (1.1), i.e.,

$$\mathbb{E}\left[\|x^k - x^\dagger\|^2\right] \le \left(1 - \frac{\lambda_{\min}}{\|A\|_F^2}\right)^k \|x^0 - x^\dagger\|^2, \ \ \forall k \ge 1. \tag{1.2}$$

In 2018, Bai and Wu [1] proposed the greedy randomized Kaczmarz (GRK) algorithm for solving (1.1) as follows.

Indeed, they proved the following interesting convergence result.

Theorem 1.1. ([1]) *Choose $x^0 \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ arbitrarily and let $\{x^k\}_{k=0}^\infty$ be a sequence generated by Algorithm 1.2. Then $\{x^k\}_{k=0}^\infty$ converges linearly to $x^\dagger \in \mathscr{S}$. Precisely, for all $k \ge 0$, there holds the convergence rate estimate:*

$$\mathbb{E}\left[\|x^{k+1} - x^\dagger\|^2\right] \le \left[1 - \frac{1}{2}\left(\frac{\|A\|_F^2}{\alpha_F(A)} + 1\right)\frac{\lambda_{\min}}{\|A\|_F^2}\right]^k \left(1 - \frac{\lambda_{\min}}{\|A\|_F^2}\right)\|x^0 - x^\dagger\|^2. \tag{1.6}$$

We note that the same index $i \in \mathscr{I}$ may be selected in two adjacent iterations in the implementation of Algorithm 1.1, which is a defect because it affects the convergence speed of the algorithm. In this paper, we first give a modified version of Algorithm 1.1 (MRK) and give a convergence rate estimate, which is better than (1.2). Second, we study the greedy Kaczmarz

Algorithm 1.2 Greedy Randomized Kaczmarz Algorithm (GRK)

Input: A, b and ℓ
Output: x^ℓ
Step 0: Take $x^0 \in \text{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ arbitrarily.
for $k = 0, 1, \cdots, \ell - 1$ **do**
Step 1: Compute

$$\varepsilon_k = \frac{1}{2}\left(\frac{1}{\|b - Ax^k\|^2} \max_{1\le i\le m}\left\{\frac{|b_i - A_i x^k|^2}{\|A_i\|^2}\right\} + \frac{1}{\|A\|_F^2}\right) \tag{1.3}$$

Step 2: Determine the index set of positive integers

$$\mathscr{U}_k = \left\{ i \in \mathscr{I} \mid |b_i - A_i x^k|^2 \ge \varepsilon_k \|b - Ax^k\|^2 \|A_i\|^2 \right\} \tag{1.4}$$

Step 3: Compute the ith entry of the column vector $\tilde{r}^k$ according to

$$\tilde{r}_i^k = \begin{cases} b_i - A_i x^k, & \text{if} \quad i \in \mathscr{U}_k, \\ 0, & \text{otherwise} \end{cases} \tag{1.5}$$

Step 4: Select $i_k \in \mathscr{U}_k$ with probability $p_{i_k} = \frac{|\tilde{r}_{i_k}^k|^2}{\|\tilde{r}^k\|^2}$
Step 5: Set $x^{k+1} = P_{i_k} x^k$
endfor

(GK) algorithm, which can be regarded as a simplified form of Algorithm 1.2 (the greedy randomized Kaczmarz (GRK) algorithm), in which the random steps in Algorithm 1.2 are deleted. A deterministic (not in the sense of expectation) convergence rate estimate is obtained for the GK algorithm. We find that the GK algorithm not only needs less computational workload in each iteration, but also has a better convergence rate than the GRK algorithm. Numerical results support the theoretical analysis in this paper.

2. PRELIMINARIES

In this section, we list some basic lemmas that will be used in the convergence analysis.

Let v be a fixed nonzero vector in $\mathbb{R}^n$, and let d be a number in $\mathbb{R}$. Then the subset

$$H := \{x \in \mathbb{R}^n \mid \langle x, v\rangle = d\}, \tag{2.1}$$

is called a hyperplane.

Lemma 2.1. ([2, 5]) *Let H be a hyperplane of form (2.1). Then*

$$P_H u = u - \frac{\langle u, v\rangle - d}{\|v\|^2} v, \ \forall u \in \mathbb{R}^n. \tag{2.2}$$

Lemma 2.2. ([9]) *Let C be a closed convex subset of $\mathbb{R}^n$. Given $x \in \mathbb{R}^n$ and $z \in C$, $z = P_C x$ if and only if*

$$\langle x - z, c - z\rangle \le 0, \quad \forall c \in C. \tag{2.3}$$

Moreover, if C is a hyperplane, then $z = P_C x$ if and only if

$$\langle x - z, c - z\rangle = 0, \quad \forall c \in C. \tag{2.4}$$

Lemma 2.3. [4] *Let $A \in \mathbb{R}^{m\times n}$ be a matrix with row vectors $A_1, A_2, \cdots, A_m$. Then the inequality*

$$\|Ax\|^2 \geq \lambda_{\min}\|x\|^2 \tag{2.5}$$

holds for all $x \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$.

3. A Modified Randomized Kaczmarz Algorithm

It is easy to see that the same index $i \in \mathscr{I}$ may be selected in two adjacent iterations in the implementation of the RK algorithm (Algorithm 1.1), which is a defect of the RK algorithm because this makes the latter iteration meaningless, and hence the convergence speed of the RK algorithm would be affected. To overcome this defect, we give the modified randomized Kaczmarz (MRK) algorithm as follows.

Algorithm 3.1 Modified Randomized Kaczmarz Algorithm

Input: A, b and ℓ
Output: x^ℓ
Step 0: Take $x^0 \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ arbitrarily.
Step 1: Select an index $i_0 \in \mathscr{I}$ with probability $p_{i_0} = \frac{\|A_{i_0}\|^2}{\|A\|_F^2}$ and calculate $x^1 = P_{i_0}x^0$.
for $k = 1, 2, \cdots, \ell-1$ **do**
Step 2: For the current x^k, $k \geq 1$, select an index $i_k \in \mathscr{I} - \{i_{k-1}\}$ with probability
$p_{i_k} = \frac{\|A_{i_k}\|^2}{\|A\|_F^2 - \|A_{i_{k-1}}\|^2}$ and calculate $x^{k+1} = P_{i_k}x^k$.
endfor

We are in a position to prove the following convergence result for Algorithm 3.1.

Theorem 3.1. *Let the initial guess x^0 be selected in $\operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ arbitrarily. Then the sequence $\{x^k\}_{k=0}^\infty$ generated by Algorithm 3.1 converges linearly in expectation to $x^\dagger$ for linear system (1.1), i.e.,*

$$\mathbb{E}\left[\|x^{k+1} - x^\dagger\|^2\right] \leq \left(1 - \frac{\lambda_{\min}}{\alpha_F(A)}\right)^k \left(1 - \frac{\lambda_{\min}}{\|A\|_F^2}\right) \|x^0 - x^\dagger\|^2, \ \forall k \geq 0. \tag{3.1}$$

Proof. For any $k \geq 0$, note that $A_{i_k}x^\dagger = b_{i_k}$. From Algorithm 3.1 and (2.2), we conclude that

$$\begin{aligned}
\|x^{k+1} - x^\dagger\|^2 =& \|P_{i_k}x^k - x^\dagger\|^2 \\
=& \left\| x^k - x^\dagger - \frac{A_{i_k}x^k - b_{i_k}}{\|A_{i_k}\|^2} A_{i_k}^\top \right\|^2 \\
=& \|x^k - x^\dagger\|^2 - 2\left\langle x^k - x^\dagger, \frac{A_{i_k}x^k - b_{i_k}}{\|A_{i_k}\|^2} A_{i_k}^\top \right\rangle + \frac{|A_{i_k}x^k - b_{i_k}|^2}{\|A_{i_k}\|^2} \\
=& \|x^k - x^\dagger\|^2 - \frac{|A_{i_k}x^k - b_{i_k}|^2}{\|A_{i_k}\|^2}.
\end{aligned} \tag{3.2}$$

For $k = 0$, we find that

$$\begin{aligned}\mathbb{E}\left[\|x^1 - x^\dagger\|^2\right] &= \|x^0 - x^\dagger\|^2 - \sum_{i=1}^{m} \frac{\|A_i\|^2}{\|A\|_F^2} \frac{|A_i x^0 - b_i|^2}{\|A_i\|^2} \\ &= \|x^0 - x^\dagger\|^2 - \frac{1}{\|A\|_F^2}\|Ax^0 - b\|^2 \\ &= \|x^0 - x^\dagger\|^2 - \frac{1}{\|A\|_F^2}\|A(x^0 - x^\dagger)\|^2.\end{aligned} \tag{3.3}$$

Using Lemma 2.3 and (3.3), we obtain

$$\mathbb{E}\left[\|x^1 - x^\dagger\|^2\right] \leq \left(1 - \frac{\lambda_{\min}}{\|A\|_F^2}\right)\|x^0 - x^\dagger\|^2. \tag{3.4}$$

For $k \geq 1$, under the condition that x^k has been obtained, we have the conditional expectation of $\|x^{k+1} - x^\dagger\|^2$ that

$$\begin{aligned}\mathbb{E}\left[\|x^{k+1} - x^\dagger\|^2 \,\middle|\, x^k\right] &= \|x^k - x^\dagger\|^2 - \sum_{i \in \mathscr{I} - \{i_{k-1}\}} \frac{\|A_i\|^2}{\|A\|_F^2 - \|A_{i_{k-1}}\|^2} \frac{|A_i x^k - b_i|^2}{\|A_i\|^2} \\ &= \|x^k - x^\dagger\|^2 - \frac{1}{\|A\|_F^2 - \|A_{i_{k-1}}\|^2} \sum_{i \in \mathscr{I} - \{i_{k-1}\}} |A_i x^k - b_i|^2.\end{aligned} \tag{3.5}$$

Note that $x^k = P_{i_{k-1}} x^{k-1}$ implies $A_{i_{k-1}} x^k = b_{i_{k-1}}$. Then (3.5) can be rewritten as

$$\begin{aligned}\mathbb{E}\left[\|x^{k+1} - x^\dagger\|^2 \,\middle|\, x^k\right] &= \|x^k - x^\dagger\|^2 - \frac{1}{\|A\|_F^2 - \|A_{i_{k-1}}\|^2} \sum_{i=1}^{m} |A_i x^k - b_i|^2 \\ &\leq \|x^k - x^\dagger\|^2 - \frac{1}{\alpha_F(A)}\|Ax^k - b\|^2 \\ &= \|x^k - x^\dagger\|^2 - \frac{1}{\alpha_F(A)}\|A(x^k - x^\dagger)\|^2.\end{aligned} \tag{3.6}$$

Applying Lemma 2.3 to (3.6), we have

$$\mathbb{E}\left[\|x^{k+1} - x^\dagger\|^2 \,\middle|\, x^k\right] \leq \left(1 - \frac{\lambda_{min}}{\alpha_F(A)}\right)\|x^k - x^\dagger\|^2. \tag{3.7}$$

Taking full expectation for both sides of (3.7), we obtain

$$\mathbb{E}\left[\|x^{k+1} - x^\dagger\|^2\right] \leq \left(1 - \frac{\lambda_{min}}{\alpha_F(A)}\right)\mathbb{E}\left[\|x^k - x^\dagger\|^2\right], \ \forall k \geq 1. \tag{3.8}$$

By (3.8), it can be obtained by mathematical induction that

$$\mathbb{E}\left[\|x^{k+1} - x^\dagger\|^2\right] \leq \left(1 - \frac{\lambda_{min}}{\alpha_F(A)}\right)^k \mathbb{E}\left[\|x^1 - x^\dagger\|^2\right], \ \forall k \geq 1. \tag{3.9}$$

Hence, (3.1) follows from (3.4) and (3.9). This completes the proof. □

Remark 3.2. Obviously, the estimation of convergence rate in (3.1) is better than that in (1.2).

4. Convergence Rate of the Greedy Kaczmarz Algorithm

In this section, we mainly analyze the convergence rate of the greedy Kaczmarz algorithm.

First, let us sketch the basic idea of the GK algorithm. Suppose that the iteration step of the successive projection algorithm for solving (1.1) is

$$x^{k+1} = P_{i_k} x^k, \tag{4.1}$$

where $i_k \in \mathscr{I}$ is selected by some strategy. From (2.2), we have

$$x^{k+1} = x^k - \frac{A_{i_k} x^k - b_{i_k}}{\|A_{i_k}\|^2} A_{i_k}^\top. \tag{4.2}$$

Similar to (3.2), for any $x^* \in \mathscr{S}$, we have

$$\|x^{k+1} - x^*\|^2 = \|x^k - x^*\|^2 - \frac{|A_{i_k} x^k - b_{i_k}|^2}{\|A_{i_k}\|^2}. \tag{4.3}$$

Consequently, for every positive integer N, we obtain

$$\sum_{k=0}^{N} \frac{|A_{i_k} x^k - b_{i_k}|^2}{\|A_{i_k}\|^2} = \|x^0 - x^*\|^2 - \|x^{N+1} - x^*\|^2. \tag{4.4}$$

If $\{x^k\}_{k=0}^{\infty}$ converges to some $\tilde{x} \in S$, replacing x^* with $\tilde{x}$ and letting $N \to \infty$ in (4.4), we arrive at

$$\sum_{k=0}^{\infty} \frac{|A_{i_k} x^k - b_{i_k}|^2}{\|A_{i_k}\|^2} = \|x^0 - \tilde{x}\|^2. \tag{4.5}$$

(4.5) tells us that, for the selected initial guess x^0, the sum of the series $\sum_{k=0}^{\infty} \frac{|A_{i_k} x^k - b_{i_k}|^2}{\|A_{i_k}\|^2}$ is a fixed constant $\|x^0 - \tilde{x}\|^2$. Therefore, it is easy to see that in order to make the algorithm obtain the fastest convergence speed, the optimal selection strategy of $i_k \in \mathscr{I}$ is to make i_k satisfy the condition:

$$\frac{|A_{i_k} x^k - b_{i_k}|^2}{\|A_{i_k}\|^2} = \max_{i \in \mathscr{I}} \frac{|A_i x^k - b_i|^2}{\|A_i\|^2}.$$

Based on the above analysis, the GK algorithm follows below.

Algorithm 4.1 Greedy Kaczmarz Algorithm

Input: A, b and ℓ
Output: x^ℓ
step 0: Take $x^0 \in \text{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ arbitrarily.
for $k = 0, 1, \cdots, \ell - 1$ **do**
step 1: For the current x^k, select an index $i_k \in \mathscr{I}$ such that

$$\frac{|A_{i_k} x^k - b_{i_k}|^2}{\|A_{i_k}\|^2} = \max_{i \in \mathscr{I}} \frac{|A_i x^k - b_i|^2}{\|A_i\|^2}, \tag{4.6}$$

and compute

$$x^{k+1} = P_{i_k} x^k. \tag{4.7}$$

endfor

Remark 4.1. Obviously, Algorithm 4.1 can be regarded as a simplified version of Algorithm 1.2, i.e., the GRK algorithm. Because it does not need to construct the index set $\mathscr{U}_k$ and randomly select an index in $\mathscr{U}_k$ in each iteration step, Algorithm 4.1 has less computational workload than the GRK algorithm.

We now analyze the convergence rate of Algorithm 4.1 as follows.

Theorem 4.2. *Choose $x^0 \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ arbitrarily and let $\{x^k\}_{k=0}^\infty$ be a sequence generated by Algorithm 4.1. Then $\{x^k\}_{k=0}^\infty$ converges linearly to $x^\dagger$. Precisely, for all $k \geq 0$, there holds the convergence rate estimate:*

$$\|x^{k+1} - x^\dagger\|^2 \leq \left(1 - \frac{\lambda_{\min}}{\alpha_F(A)}\right)^k \left(1 - \frac{\lambda_{\min}}{\|A\|_F^2}\right) \|x^0 - x^\dagger\|^2. \tag{4.8}$$

Proof. From (2.2) and (4.7), it is easy to see that $x^k \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ holds for all $k \geq 0$ thanks to $x^0 \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$. Thus $x^k - x^\dagger \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ also holds for all $k \geq 0$ due to $x^\dagger \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$. Similar to (3.2), for all $k \geq 0$, we have

$$\|x^{k+1} - x^\dagger\|^2 = \|x^k - x^\dagger\|^2 - \frac{|A_{i_k} x^k - b_{i_k}|^2}{\|A_{i_k}\|^2}. \tag{4.9}$$

For $k = 0$, we have from (4.6) that

$$\begin{aligned} \frac{|A_{i_0} x^0 - b_{i_0}|^2}{\|A_{i_0}\|^2} &\geq \sum_{i \in \mathscr{I}} \frac{\|A_i\|^2}{\|A\|_F^2} \frac{|A_i x^0 - b_i|^2}{\|A_i\|^2} \\ &= \frac{1}{\|A\|_F^2} \sum_{i \in \mathscr{I}} |A_i(x^0 - x^\dagger)|^2 \\ &= \frac{1}{\|A\|_F^2} \|A(x^0 - x^\dagger)\|^2 \geq \frac{\lambda_{\min}}{\|A\|_F^2} \|x^0 - x^\dagger\|^2, \end{aligned} \tag{4.10}$$

where the last inequality holds thanks to $x^0 - x^\dagger \in \operatorname{span}\{A_1^\top, A_2^\top, \cdots, A_m^\top\}$ and Lemma 2.3. Combining (4.9) and (4.10), we obtain

$$\begin{aligned} \|x^1 - x^\dagger\|^2 &= \|x^0 - x^\dagger\|^2 - \frac{|A_{i_0} x^0 - b_{i_0}|^2}{\|A_{i_0}\|^2} \\ &\leq \left(1 - \frac{\lambda_{\min}}{\|A\|_F^2}\right) \|x^0 - x^\dagger\|^2. \end{aligned} \tag{4.11}$$

For each $k \geq 1$, note that $A_{i_{k-1}}x^k - b_{i_{k-1}} = A_{i_{k-1}}(x^k - x^\dagger) = 0$ due to $x^k = P_{i_{k-1}}x^{k-1}$. By using (4.6) and Lemma 2.3, we obtain that

$$\begin{aligned}
\frac{|A_{i_k}x^k - b_{i_k}|^2}{\|A_{i_k}\|^2} &\geq \sum_{i \in \mathscr{I} - \{i_{k-1}\}} \frac{\|A_i\|^2}{\|A\|_F^2 - \|A_{i_{k-1}}\|^2} \frac{|A_i x^k - b_i|^2}{\|A_i\|^2} \\
&= \frac{1}{\|A\|_F^2 - \|A_{i_{k-1}}\|^2} \sum_{i \in \mathscr{I}} |A_i(x^k - x^\dagger)|^2 \\
&\geq \frac{1}{\alpha_F(A)} \|A(x^k - x^\dagger)\|^2 \\
&\geq \frac{\lambda_{\min}}{\alpha_F(A)} \|x^k - x^\dagger\|^2.
\end{aligned} \tag{4.12}$$

Consequently, it follows from (4.9) and (4.12) that

$$\|x^{k+1} - x^\dagger\|^2 \leq \left(1 - \frac{\lambda_{\min}}{\alpha_F(A)}\right) \|x^k - x^\dagger\|^2. \tag{4.13}$$

Hence, in view of (4.11) and (4.13), we obtain (4.8). This completes the proof. □

Remark 4.3. Note that

$$1 - \frac{\lambda_{\min}}{\alpha_F(A)} \leq 1 - \frac{1}{2}\left(\frac{\|A\|_F^2}{\alpha_F(A)} + 1\right) \frac{\lambda_{\min}}{\|A\|_F^2}.$$

holds, and (4.8) is deterministic, however (1.6) is only in the sense of expectation. Hence the convergence rate estimate of Algorithm 4.1 is better than that of Algorithm 1.2.

5. Numerical Results

In this section, we perform several experiments to compare the convergence speed of the RK algorithm (Algorithm 1.1) with the MRK algorithm (Algorithm 3.1), and the GRK algorithm (Algorithm 1.2) with the GK algorithm (Algorithm 4.1), respectively.

We use "IT" and "CPU" to represent the average number of iterations required and the average CPU time taken to run these algorithms 50 times, respectively. We report the speed-up1 of MRK against RK, which is defined as

$$\text{speed-up1} = \frac{\text{CPU of RK}}{\text{CPU of MRK}}.$$

Similarly, we report the speed-up2 of GK against GRK, which is defined as

$$\text{speed-up2} = \frac{\text{CPU of GRK}}{\text{CPU of GK}}.$$

To test these algorithms, we use the function $unifrnd$ in MATLAB to randomly construct various types of $m \times n$ matrices A and, set the entries of A to be independent identically distributed uniform random variables on some interval $[c, 1]$. Changing the value of c will appropriately change the coherence of A.

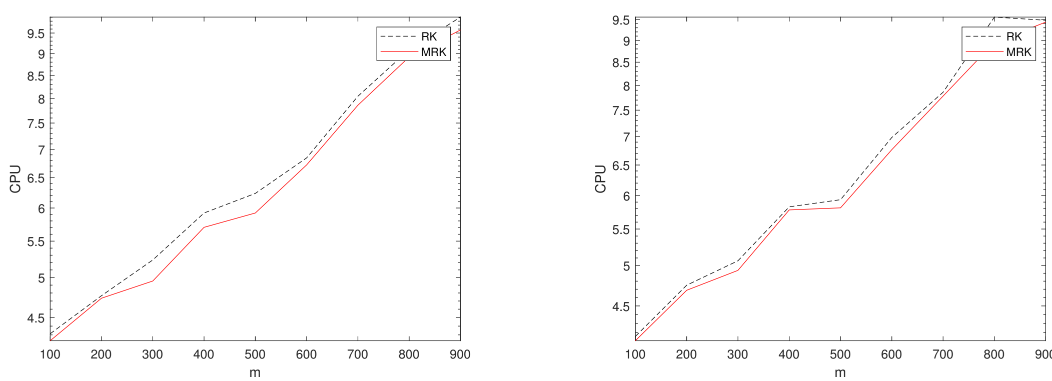
In the algorithm implementation, we randomly generate a vector x and calculate $b = Ax$ to get b, so that we can ensure that problem (1.1) is consistent. For each implementation, RK and MRK start with the same initial $x^0 = 0$, and stop when it reaches the number of iteration steps

TABLE 1. CPU of RK and MRK for m-by-n matrices with $n = 1000$ and different m when $c = 0$.

m	100	300	500	700	900
RK-CPU	4.3125	5.2344	6.2344	8.0469	9.9063
MRK-CPU	4.2344	4.9531	5.9219	7.8594	9.5781
speed-up1	1.0184	1.0568	1.0528	1.0239	1.0343

TABLE 2. CPU of RK and MRK for m-by-n matrices with $n = 1000$ and different m when $c = 0.9$.

m	100	300	500	700	900
RK-CPU	4.1563	5.0625	5.9375	7.8594	9.4844
MRK-CPU	4.1094	4.9375	5.8125	7.7813	9.4375
speed-up1	1.0114	1.0253	1.0215	1.0100	1.0050



(a) (b)

FIGURE 1. CPU versus m with $n = 1000$ matrices for RK and MRK, when $c = 0$(a) and $c = 0.9$(b).

$\ell = 500000$. GRK and GK start with the same initial $x^0 = 0$, and stop once the relative error of the current x^k, defined by

$$\text{RSE} = \frac{\|x^k - x^\dagger\|^2}{\|x^\dagger\|^2},$$

satisfies $\text{RSE} \le 10^{-6}$.

In addition, all experiments are carried out using MATLAB (version R2018a) on a personal computer with 2.80 GHz central processing unit (Intel(R) Core(TM) i7-1165G7 CPU), 16.00GB memory, and Windows operating system (Windows 10).

The experimental results are presented in the tables and figures.

From Table 1-4 and Figure 1-2, it is clear that the MRK algorithm always outperforms the RK algorithm in terms of CPU time.

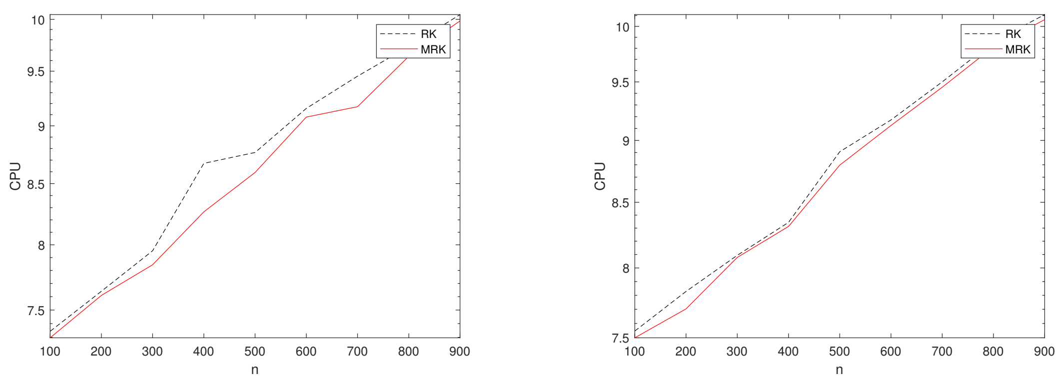
From Table 5-6 and Figure 3-4, we find that, for any type of the underdetermined linear equations, the GK algorithm always has a great advantage over the GRK algorithm, no matter in terms of CPU time or the number of iteration steps.

TABLE 3. CPU of RK and MRK for m-by-n matrices with $m = 1000$ and different n when $c = 0$.

n	100	300	500	700	900
RK-CPU	7.3438	7.9531	8.7656	9.4531	10.0469
MRK-CPU	7.2969	7.8438	8.5938	9.1719	9.9844
speed-up1	1.0064	1.0139	1.0200	1.0307	1.0063

TABLE 4. CPU of RK and MRK for m-by-n matrices with $m = 1000$ and different n when $c = 0.9$.

n	100	300	500	700	900
RK-CPU	7.5469	8.0938	8.9063	9.5000	10.1094
MRK-CPU	7.5000	8.0781	8.7969	9.4531	10.0625
speed-up1	1.0063	1.0019	1.0124	1.0050	1.0047



(a) (b)

FIGURE 2. CPU versus n with $m = 1000$ matrices for RK and MRK, when $c = 0$(a) and $c = 0.9$(b).

TABLE 5. IT and CPU of GRK and GK for $m = 100$ matrices with different n when $c = 0$.

n		1000	2000	3000	4000	5000
GRK	CPU	5.9063	8.9844	14.3438	17.7031	24.3594
	IT	7.7495×10^3	6.9039×10^3	7.0558×10^3	7.0150×10^3	6.6983×10^3
GK	CPU	0.3125	0.3438	0.5781	0.6406	3.1094
	IT	1.3674×10^3	1.3652×10^3	1.3082×10^3	1.2002×10^3	1.3380×10^3
speed-up2		18.9000	26.1364	24.8108	27.6341	7.8342

6. CONCLUDING REMARK

In this paper, we first proposed a modified randomized Kaczmarz (MRK) algorithm and obtain a better convergence rate estimate than that of the randomized Kaczmarz (RK) algorithm. Second, we proved that the greedy Kaczmarz (GK) algorithm not only needs less computational

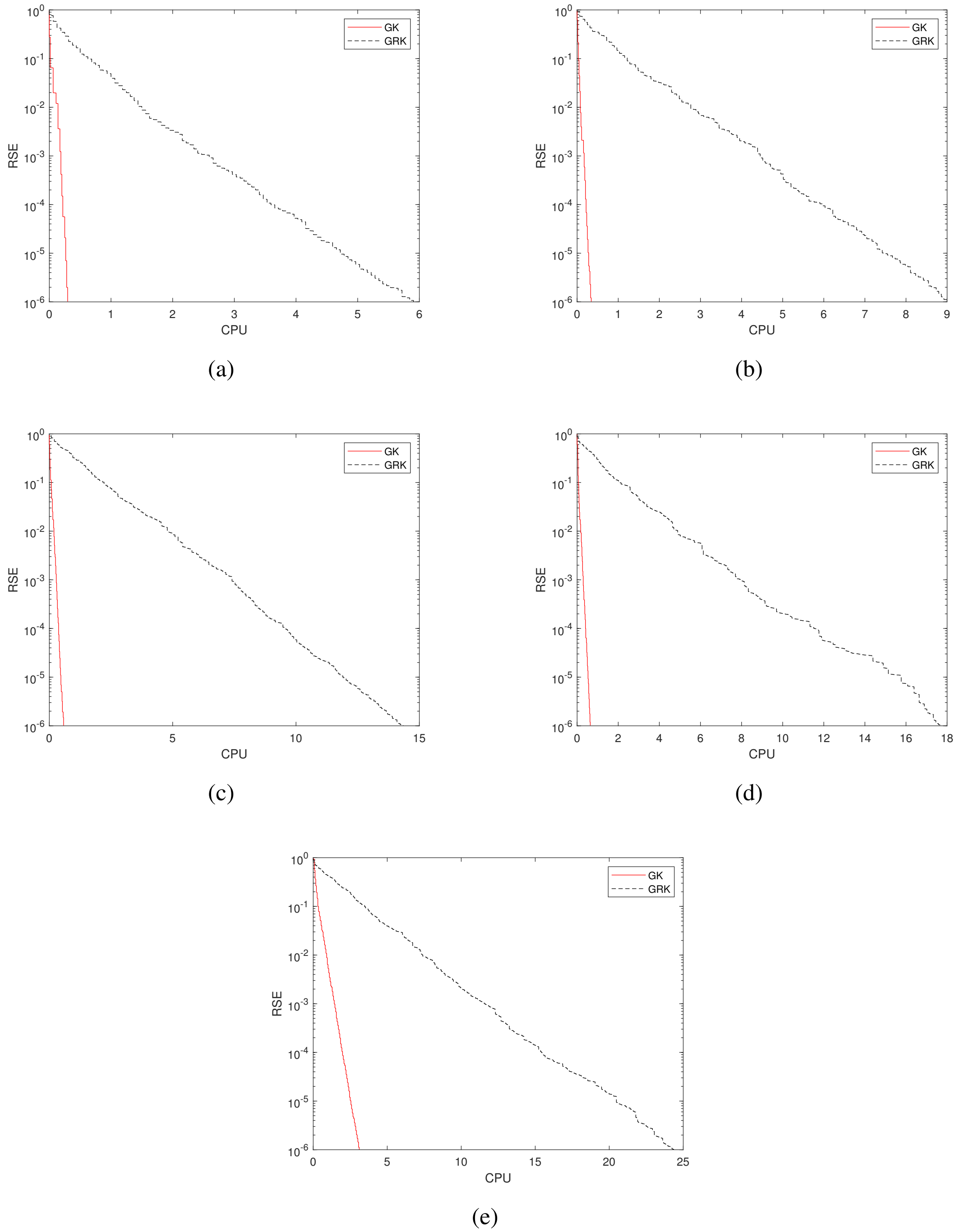


FIGURE 3. RSE versus CPU with $m = 100$ and $c = 0$ matrices for GRK and GK, when $n = 1000$(a), $n = 2000$(b), $n = 3000$(c), $n = 4000$(d) and $n = 5000$(e).

workload in each iteration, but also has faster convergence speed than the greedy randomized Kaczmarz (GRK) algorithm. Numerical results show the advantages of the MRK algorithm and the GK algorithm.

TABLE 6. IT and CPU of GRK and GK for $m = 100$ matrices with different n when $c = 0.9$.

n		1000	2000	3000	4000	5000
GRK	CPU	2.3169×10^3	3.0237×10^3	4.1600×10^3	5.5289×10^3	7.2854×10^3
	IT	2.1794×10^6	1.7453×10^6	1.4632×10^6	1.5745×10^6	1.4107×10^6
GK	CPU	55.7031	83.3438	1.4405×10^2	1.8498×10^2	9.8561×10^2
	IT	3.8308×10^5	3.5505×10^5	3.5157×10^5	3.5583×10^5	3.4260×10^5
speed-up2		41.5933	36.2795	28.8794	29.8887	7.3918

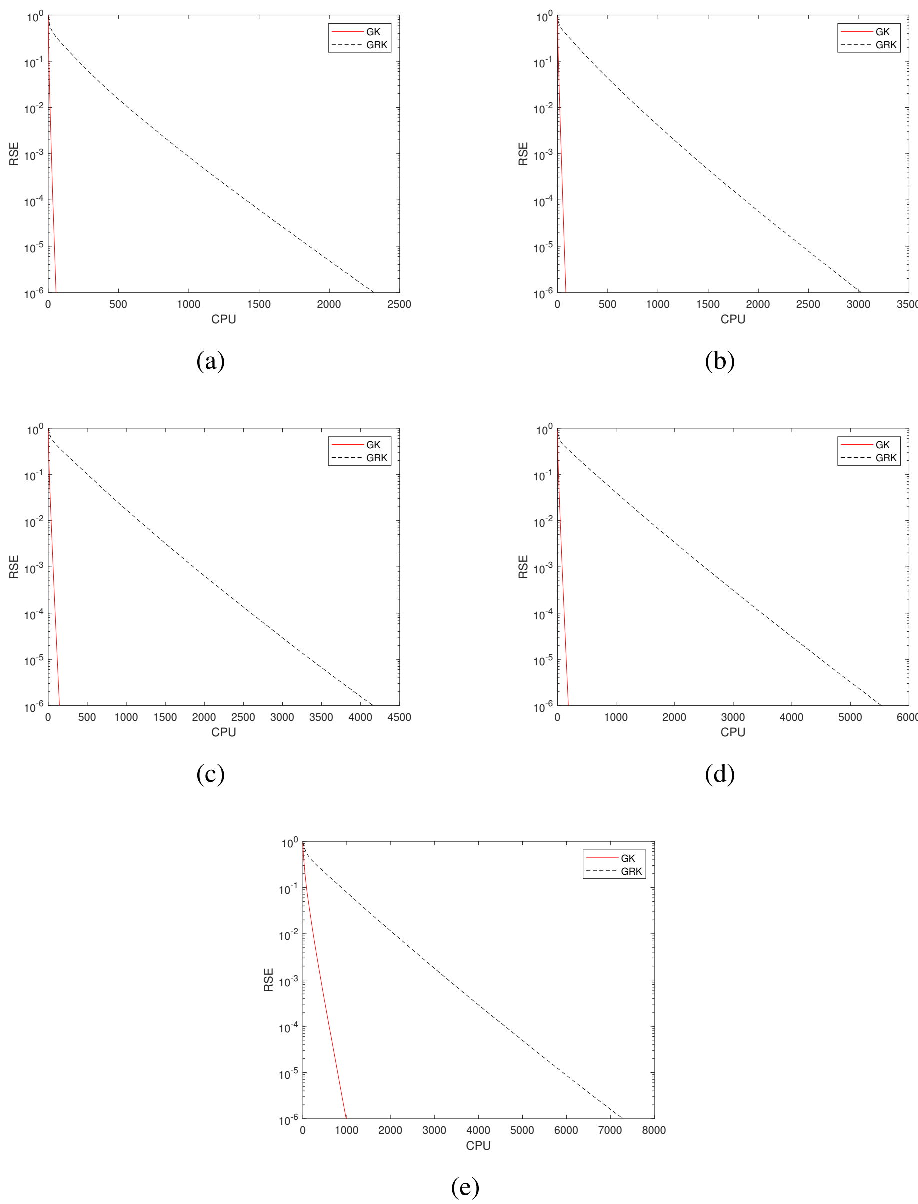


FIGURE 4. RSE versus CPU with $m = 100$ and $c = 0.9$ matrices for GRK and GK, when $n = 1000$(a), $n = 2000$(b), $n = 3000$(c), $n = 4000$(d) and $n = 5000$(e).

Acknowledgements
The authors are grateful to the anonymous referees for useful comments which improved the presentation of this paper.

REFERENCES

[1] Z.Z. Bai, W.T. Wu, On greedy randomized Kaczmarz method for solving large sparse linear systems, SIAM J. Sci. Comput. 40 (2018) A592-A606.
[2] H.H. Bauschke, P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, Springer, New York, NY, 2010.
[3] C.L. Byrne, Iterative Optimization in Inverse Problems, CRC Press, Boca Raton, 2014.
[4] R.M. Gower, P. Richtárik, Stochastic dual ascent for solving linear systems, arXiv: 1512.06890, 2015.
[5] S. He, C. Yang, C. Duan, Realization of the hybrid method for Mann iterations, Appl. Math. Comput. 217 (2010) 4239-4247.
[6] S. Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichungen, Bull. Int. Acad. Polon. Sci. Lett. A, 35 (1937) 355-357.
[7] A. Ma, D. Needell, A. Ramdas, Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods, SIAM J. Matrix Anal. Appl. 36 (2015) 1590-1604.
[8] T. Strohmer, R. Vershynin, A randomized Kaczmarz algorithm with exponential convergence, J. Fourier Anal. Appl. 15 (2009) 262-278.
[9] H. Zhou, X. Qin, Fixed Points of Nonlinear Operators, De Gruyter, Berlin, 2020.