



---

*Research article*

## **Greedy randomized and maximal weighted residual Kaczmarz methods with oblique projection**

**Fang Wang, Weiguo Li\*, Wendi Bao and Li Liu**

China University of Petroleum, Qingdao 266580, China

\* **Correspondence:** Email: [liwg@upc.edu.cn](mailto:liwg@upc.edu.cn).

**Abstract:** For solving large-scale consistent linear system, a greedy randomized Kaczmarz method with oblique projection and a maximal weighted residual Kaczmarz method with oblique projection are proposed. By using oblique projection, these two methods greatly reduce the number of iteration steps and running time to find the minimum norm solution, especially when the rows of matrix are highly linearly correlated. Theoretical proof and numerical results show that the greedy randomized Kaczmarz method with oblique projection and the maximal weighted residual Kaczmarz method with oblique projection are more effective than the greedy randomized Kaczmarz method and the maximal weighted residual Kaczmarz method respectively.

**Keywords:** oblique projection; convergence property; Kaczmarz method; correlation; large linear system

---

### **1. Introduction**

Consider to solve a large-scale consistent linear system

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

where the matrix  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ . One of the solutions of the system (1.1) is  $x^* = A^\dagger b$ , which is the least Euclidean norm solution. Especially, when the coefficient matrix  $A$  is full column rank,  $x^*$  is the unique solution of the system (1.1).

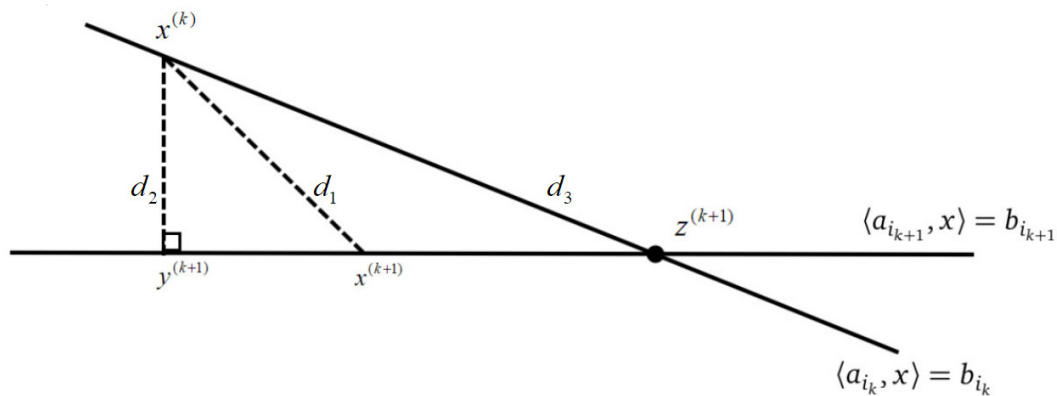
There are many researches on solving the system (1.1) through iterative methods, among which the Kaczmarz method is a representative and efficient row-action method. The Kaczmarz method [1] selects the rows of the matrix  $A$  by using the cyclic rule and in each iteration, the current iteration point is orthogonally projected onto the corresponding hyperplane. In 1970, Gordon et al. [2] first applied the Kaczmarz method, also known as algebraic reconstruction technique (ART), to the field of

computed tomography (CT). In the development of CT field, representative methods include filtered-back projection (FBP) method [3], ART and Maximum entropy method [4, 5], etc. However, if the collected data are incomplete, the FBP method performs very poorly and the ART method is widely used in this field [6, 7] due to its superior anti-interference performance, implicitness and low storage characteristics. Kaczmarz method is also widely applied in image reconstruction [8–11], distributed computing [12] and signal processing [13]; and so on [14–17]. In 1971, Tanabe [18] analyzed the theoretical convergence of the Kaczmarz method and obtained the conclusion that when initial vector  $x^{(0)} \in N(A)$ , the Kaczmarz method converges to the minimum norm solution  $x^*$  of the problem (1.1). In recent years, Kang et al. [19, 20] obtained the theoretical convergence rate proof of the Kaczmarz method.

Since the Kaczmarz method cycles through the rows of  $A$ , the performance may depend heavily on the ordering of these rows. A poor ordering may result in a very slow convergence rate. McCormick [21] proposed a maximal weighted residual Kaczmarz (MWRK) method and proved its convergence. In recent work, a new theoretical convergence estimate was proposed for the MWRK method in [22]. Strohmer and Vershynin [23] proposed a randomized Kaczmarz (RK) method which selects a given row with proportional to the Euclidean norm of the rows of the coefficient matrix  $A$  and proved its convergence. After the above work, research on the Kaczmarz-type methods was reignited recently, see for example, the randomized block Kaczmarz-type methods [24–27], the greedy version of Kaczmarz-type methods [28–32], the extended version of Kaczmarz-type methods [33–35] and many others [36–41]. The related works of Kaczmarz also accelerated the development of column action iterative methods represented by the coordinate descent (CD) method [42] ( see [43–50], etc.).

Recently, Bai and Wu [28] proposed a new randomized row index selection strategy, which is aimed at grasping larger entries of the residual vector at each iteration and constructed a greedy randomized Kaczmarz (GRK) method. They proved that the convergence of the GRK method is faster than that of the RK method. In [38, 51], Popa gave the definition of oblique projection, which broke the limitation of orthogonal projection. In [52], Lorenez and Rose et al. used oblique projection to construct random Kaczmarz methods with mismatched adjoint (RKMA). Recently, Li and Wang et al. [53] proposed a Kaczmarz method with oblique projection (KO). This method continuously projects the current iteration point to the intersection of two hyperplanes and can solve the problem that the rows of the coefficient matrix  $A$  are highly linearly correlated as well as the algorithms proposed in [36, 54]. They also proposed a uniform random version of KO method—random Kaczmarz method with oblique projection (RKO) and proved theoretically and numerically that KO method and RKO method are faster than Kaczmarz method [1] and RK method [23] respectively. In this paper, we first briefly introduce the oblique projection and give the relationship between the KO method and the CD method. Based on the row index selection rules of two representative randomized and non-randomized Kaczmarz-type methods—the GRK method and the MWRK method, we propose two new Kaczmarz-type methods with oblique projection (KO-type)—the greedy randomized Kaczmarz method with oblique projection (GRKO) and the maximal weighted residual Kaczmarz method with oblique projection (MWRKO) respectively and prove their convergence theoretically and numerically. We emphasize the efficiency of our proposed methods when the rows of the matrix  $A$  are highly linearly correlated and find that Kaczmarz-type methods based on orthogonal projection performed poorly when applied to this kind of matrices.

The organization of this paper is as follows. In Section 2, we introduce the KO-type method and



**Figure 1.** Oblique projection in different directions.

give its two lemmas. In Section 3, we propose the GRKO method and MWRKO method naturally and prove the convergence of the two methods. In Section 4, some numerical examples are provided to illustrate the efficiency of our new methods. Finally, some brief concluding remarks are described in Section 5.

In this paper,  $\langle \cdot, \cdot \rangle$  stands for the scalar product.  $\|x\|$  is the Euclidean norm of  $x \in \mathbb{R}^n$ . For a given matrix  $G = (g_{ij}) \in \mathbb{R}^{m \times n}$ ,  $g_i^T$ ,  $G^T$ ,  $G^\dagger$ ,  $R(G)$ ,  $N(G)$ ,  $\|G\|_F$  and  $\lambda_{\min}(G)$ , are used to denote the  $i$ th row, the transpose, the Moore-Penrose pseudoinverse [55], the range space, the null space, the Frobenius norm and the smallest nonzero eigenvalue of  $G$  respectively.  $P_C(x)$  is the orthogonal projection of  $x$  onto  $C$ ,  $\tilde{x}$  is any solution of the system (1.1). Let  $E_k$  denote the expected value conditional on the first  $k$  iterations, that is,

$$E_k[\cdot] = E[\cdot | j_0, j_1, \dots, j_{k-1}],$$

where  $j_s$  ( $s = 0, 1, \dots, k-1$ ) is the column chosen at the  $s$ th iteration.

## 2. Kaczmarz-type method with oblique projection and its lemmas

In this chapter, we first briefly introduce the definition of oblique projection and analyze the relationship between CD method [42] and KO-type method [53]. Finally, we give two lemmas of KO-type method, which provide a theoretical guarantee for the two oblique projection methods proposed in the next chapter.

### 2.1. Relationship between coordinate descent method and Kaczmarz-type method with oblique projection

The sets  $H_i = \{x \in \mathbb{R}^n \mid \langle a_i, x \rangle = b_i\}$  ( $i = 1, 2, \dots, m$ ) are the hyperplanes which associated to the  $i$ th equation of the system (1.1). To project the current iteration point  $x^{(k)}$  to one of the hyperplanes, the oblique projection [38, 51] can be expressed as follows:

$$x^{(k+1)} = P_{H_i}^d(x^{(k)}) = x^{(k)} - \frac{\langle a_i, x^{(k)} \rangle - b_i}{\langle d, a_i \rangle} d, \quad (2.1)$$

where  $d \in \mathbb{R}^n$  is a given direction. In Figure 1,  $x^{(k+1)}$  is obtained by oblique projection of the current iteration point  $x^{(k)}$  to the hyperplane  $H_{i_{k+1}}$  along the direction  $d_1$ , i.e.,  $x^{(k+1)} = P_{H_{i_{k+1}}}^{d_1}(x^{(k)})$ .  $y^{(k+1)}$  is the

iteration point obtained when the direction  $d_2 = a_{i_{k+1}}$ , i.e.,  $y^{(k+1)} = P_{H_{i_{k+1}}}^{a_{i_{k+1}}}(x^{(k)})$ . When the direction  $d = a_i$  ( $i = \text{mod}(k, m) + 1$ ), it is the classic Kaczmarz method. However, when the hyperplanes are close to linearly parallel, the Kaczmarz method based on orthogonal projection has a slow iteration speed. In this paper, we use a iteration direction  $d_3 = w_{i_k} = a_{i_{k+1}} - \frac{\langle a_{i_k}, a_{i_{k+1}} \rangle}{\|a_{i_k}\|^2} a_{i_k}$  mentioned in [53], to make the current iteration point approach to the intersection of two hyperplanes, i.e.,  $z^{(k+1)} = P_{H_{i_{k+1}}}^{w_{i_k}}(x^{(k)})$ .

The framework of KO-type method is given as follows:

---

**Algorithm 1** Kaczmarz-type method with oblique projection (KO-type)

---

**Require:**  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $x^{(0)} \in \mathbb{R}^n$ ,  $K$

- 1: For  $i = 1 : m$ ,  $M(i) = \|a_i\|^2$
  - 2: Choose  $i_1$  based on a certain selection rule
  - 3: Compute  $x^{(1)} = x^{(0)} + \frac{b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle}{M(i_1)} a_{i_1}$
  - 4: **for**  $k = 1, 2, \dots, K$  **do**
  - 5:     Choose  $i_{k+1}$  based on a certain selection rule
  - 6:     Compute  $D_{i_k} = \langle a_{i_k}, a_{i_{k+1}} \rangle$  and  $r_{i_{k+1}}^{(k)} = b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle$
  - 7:     Compute  $w_{i_k} = a_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} a_{i_k}$  and  $h_{i_k} (= \|w_{i_k}\|^2) = M(i_{k+1}) - \frac{D_{i_k}^2}{M(i_k)}$
  - 8:      $\alpha_{i_k}^{(k)} = \frac{r_{i_{k+1}}^{(k)}}{h_{i_k}}$  and  $x^{(k+1)} = x^{(k)} + \alpha_{i_k}^{(k)} w_{i_k}$
  - 9: **end for**
  - 10: Output  $x^{(K+1)}$
- 

For KO-type method, the residual satisfies

$$\begin{aligned} r^{(k+1)} &= b - Ax^{(k+1)} \\ &= b - A \left( x^{(k)} + \alpha_{i_k}^{(k)} \left( a_{i_{k+1}} - \frac{\langle a_{i_k}, a_{i_{k+1}} \rangle}{\|a_{i_k}\|^2} a_{i_k} \right) \right) \\ &= r^{(k)} - \alpha_{i_k}^{(k)} \left( Aa_{i_{k+1}} - \frac{\langle a_{i_k}, a_{i_{k+1}} \rangle}{\|a_{i_k}\|^2} Aa_{i_k} \right). \end{aligned} \quad (2.2)$$

In the next section, the certain selection rules in step 2 and step 5 of algorithm 1 will be given.

In order to get the relationship between the CD method and the KO-type method, we need to explain the construction idea of the CD method [42]. Consider a linear system

$$\tilde{A}x = b, \quad (2.3)$$

where the coefficient matrix  $\tilde{A} \in \mathbb{R}^{n \times n}$  is a positive semidefinite matrix and  $b \in \mathbb{R}^n$  is a real  $n$  dimensional vector. In this case, solving the system (2.3) is equivalent to the following strict convex quadratic minimization problem

$$f(x) = \frac{1}{2} x^T \tilde{A}x - b^T x.$$

From [42], the next iteration point  $x^{(k+1)}$  is the solution to  $\min_{t \in \mathbb{R}} f(x^{(k)} + td)$ , i.e.,

$$x^{(k+1)} = x^{(k)} + \frac{(b - \tilde{A}x^{(k)})^T d}{d^T \tilde{A}d} d, \quad (2.4)$$

where  $d$  is a nonzero direction and  $x^{(k)}$  is a current iteration point.

Since the requirement that matrix  $\tilde{A}$  is positive semidefinite is not general, problem (2.3) is usually transformed into the following two regularizing linear systems:

$$A^T Ax = A^T b, \quad (2.5)$$

and

$$\begin{cases} AA^T y = b, \\ x = A^T y, \end{cases} \quad (2.6)$$

where  $A \in \mathbb{R}^{m \times n}$  in (2.5) and (2.6) is an arbitrary matrix. Obviously, both  $A^T A$  and  $AA^T$  are positive semidefinite matrices, so we can apply systems (2.5) and (2.6) to iteration (2.4).

One natural choice of a set of easily computable search directions is to choose  $d$  by successively cycling through the set of canonical unit vectors  $\{e_1, \dots, e_n\}$ , where  $e_i \in \mathbb{R}^n$  ( $i = 1, \dots, n$ ). Applying system (2.5) to iteration (2.4), we can get:

$$x^{(k+1)} = x^{(k)} + \frac{\langle r^{(k)}, A_i \rangle}{\|A_i\|^2} e_i,$$

where  $i = \text{mod}(k, n) + 1$ ,  $A_i$  is the  $i$ -th column of matrix  $A$ . This is the iterative formula of CD method [42], also known as Gauss-Seidel method. When  $d = e_i$ , where  $e_i \in \mathbb{R}^m$  ( $i = 1, \dots, m$ ), applying system (2.6) to iteration (2.4), we can get:

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

where  $i = \text{mod}(k, m) + 1$ . This is the iterative formula of Kaczmarz method [1].

Next, we will prove that the KO-type method is an iterative form in the new direction  $d = e_{i_{k+1}} - \frac{\langle a_{i_{k+1}}, a_{i_k} \rangle}{\|a_{i_k}\|^2} e_{i_k} = e_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} e_{i_k}$ , where  $e_i \in \mathbb{R}^m$  ( $i = 1, \dots, m$ ). Applying system (2.6) to iteration (2.4), we get:

$$\begin{aligned} y^{(k+1)} &= y^{(k)} + \frac{(b - AA^T y^{(k)})^T (e_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} e_{i_k})}{\left\| A^T (e_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} e_{i_k}) \right\|^2} (e_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} e_{i_k}) \\ &= y^{(k)} + \frac{(b - Ax^{(k)})^T (e_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} e_{i_k})}{h_{i_k}} (e_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} e_{i_k}) \\ &= y^{(k)} + \frac{r_{i_{k+1}}^{(k)} - \frac{D_{i_k}}{M(i_k)} r_{i_k}^{(k)}}{h_{i_k}} (e_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} e_{i_k}). \end{aligned}$$

Multiply left by  $A^T$  on both sides of the above equation, we get

$$x^{(k+1)} = x^{(k)} + \frac{r_{i_{k+1}}^{(k)} - \frac{D_{i_k}}{M(i_k)} r_{i_k}^{(k)}}{h_{i_k}} w_{i_k}$$

$$= x^{(k)} + \alpha_{i_k}^{(k)} w_{i_k} - \frac{D_{i_k} r_{i_k}^{(k)}}{M(i_k) h_{i_k}} w_{i_k}, \quad k = 1, 2, \dots. \quad (2.7)$$

Now we prove that  $r_{i_k}^{(k)} = 0$ . In fact,

$$\begin{aligned} r_{i_k}^{(k)} &= b_{i_k} - \langle a_{i_k}, x^{(k)} \rangle \\ &= b_{i_k} - \langle a_{i_k}, x^{(k-1)} + \frac{r_{i_k}^{(k-1)} - \frac{D_{i_{k-1}} r_{i_{k-1}}^{(k-1)}}{M(i_{k-1})} w_{i_{k-1}}}{h_{i_{k-1}}} \rangle \\ &= r_{i_k}^{(k-1)} - r_{i_k}^{(k-1)} + \frac{D_{i_{k-1}} r_{i_{k-1}}^{(k-1)}}{M(i_{k-1})} r_{i_{k-1}}^{(k-1)} \\ &= \frac{D_{i_{k-1}} r_{i_{k-1}}^{(k-1)}}{M(i_{k-1})}, \quad k = 2, 3, \dots. \end{aligned}$$

Therefore, the following formula holds:

$$r_{i_k}^{(k)} = \prod_{s=1}^{k-1} \frac{D_{i_s}}{M(i_s)} r_{i_s}^{(s)}.$$

By the step 3 of Algorithm 1,  $r_{i_1}^{(1)} = 0$ , so

$$r_{i_k}^{(k)} = 0 \quad (\forall k > 0). \quad (2.8)$$

Thus the iteration (2.7) becomes

$$x^{(k+1)} = x^{(k)} + \alpha_{i_k}^{(k)} w_{i_k}, \quad k = 1, 2, \dots.$$

From the above deduction, we have confirmed our idea.

**Remark 1.** When  $d = e_{i_{k+1}} - \frac{\langle A_{i_{k+1}}, A_{i_k} \rangle}{\|A_{i_k}\|^2} e_{i_k}$ , where  $e_i \in \mathbb{R}^n$  ( $i = 1, \dots, n$ ), applying system (2.5) to iteration (2.4), we can get the Gauss-Seidel method with oblique direction, see [56] for details.

**Remark 2.** Formally, iteration (2.1) is the same as RKMA method [52]. Algorithm 1 can be regarded as a special case of  $v_{i_k} = a_{i_{k+1}} - \frac{\langle a_{i_k}, a_{i_{k+1}} \rangle}{\|a_{i_k}\|^2} a_{i_k}$  in RKMA method. However, the oblique projection  $v_{i_k}$  in RKMA method is more defined in a way of mismatched adjoint, which is different from the oblique projection concept proposed here, see [52] for details.

## 2.2. Theoretical guarantees of Kaczmarz-type method with oblique projection

In this section, we will give two very important lemmas of Algorithm 1 to serve the algorithms mentioned in Chapter 3. The selection rules of row indices  $i_1$  in step 2 and  $i_{k+1}$  ( $k > 0$ ) in step 5 of Algorithm 1 do not affect the lemmas.

**Lemma 2.1.** For the Kaczmarz-type method with oblique projection, the residual satisfies the following equations:

$$r_{i_{k-1}}^{(k)} = 0 \quad (\forall k > 1). \quad (2.9)$$

*Proof.* From the definition of the KO-type method, since  $k > 1$ ,

$$x^{(k)} = x^{(k-1)} + \alpha_{i_{k-1}}^{(k-1)} w_{i_{k-1}}.$$

We get

$$(b - Ax^{(k)})_{i_{k-1}} = (b - Ax^{(k-1)})_{i_{k-1}} - (A\alpha_{i_{k-1}}^{(k-1)} w_{i_{k-1}})_{i_{k-1}},$$

that is,

$$\begin{aligned} r_{i_{k-1}}^{(k)} &= r_{i_{k-1}}^{(k-1)} - \alpha_{i_{k-1}}^{(k-1)} \langle a_{i_{k-1}}, w_{i_{k-1}} \rangle \\ &\stackrel{(i)}{=} \alpha_{i_{k-1}}^{(k-1)} \langle a_{i_{k-1}}, a_{i_k} - \frac{\langle a_{i_{k-1}}, a_{i_k} \rangle}{\|a_{i_{k-1}}\|^2} a_{i_{k-1}} \rangle \\ &= 0. \end{aligned}$$

The equality (i) holds due to the Eq (2.8). Thus, the Eq (2.9) holds.

**Lemma 2.2.** *The iteration sequence  $\{x^{(k)}\}_{k=0}^{\infty}$  generated by the Kaczmarz-type method with oblique projection, satisfies the following equations:*

$$\|x^{(k+1)} - \tilde{x}\|^2 = \|x^{(k)} - \tilde{x}\|^2 - \|x^{(k+1)} - x^{(k)}\|^2 \quad (\forall k \geq 0), \quad (2.10)$$

where  $\tilde{x}$  is an arbitrary solution of the system (1.1). Especially, when  $P_{N(A)}(x^{(0)}) = P_{N(A)}(\tilde{x})$ ,  $x^{(k)} - \tilde{x} \in R(A^T)$ .

*Proof.* For  $k = 0$ , the iteration in step 3 of Algorithm 1 is the classical Kaczmarz iteration, so we have

$$\begin{aligned} \langle a_{i_1}, x^{(1)} - \tilde{x} \rangle &= \langle a_{i_1}, x^{(0)} - \tilde{x} + \frac{b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle}{M(i_1)} a_{i_1} \rangle \\ &= \langle a_{i_1}, x^{(0)} \rangle - b_{i_1} + \langle a_{i_1}, \frac{b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle}{M(i_1)} a_{i_1} \rangle \\ &= 0, \end{aligned}$$

which shows that  $x^{(1)} - \tilde{x}$  is orthogonal to  $a_{i_1}$ . Therefore, we know

$$(x^{(1)} - x^{(0)})^T (x^{(1)} - \tilde{x}) = 0.$$

It follows that

$$\|x^{(1)} - \tilde{x}\|^2 = \|x^{(0)} - \tilde{x}\|^2 - \|x^{(1)} - x^{(0)}\|^2.$$

For  $k > 0$ , we have

$$\begin{aligned} \langle w_{i_k}, x^{(k+1)} - \tilde{x} \rangle &= \langle w_{i_k}, x^{(k)} - \tilde{x} + \alpha_{i_k}^{(k)} w_{i_k} \rangle \\ &= \langle a_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} a_{i_k}, x^{(k)} - \tilde{x} \rangle + \langle w_{i_k}, \frac{r_{i_{k+1}}^{(k)}}{h_{i_k}} w_{i_k} \rangle \end{aligned}$$

$$\begin{aligned} & \stackrel{(ii)}{=} -r_{i_{k+1}}^{(k)} + \frac{D_{i_k}}{M(i_k)} r_{i_k}^{(k)} + r_{i_{k+1}}^{(k)} \\ & \stackrel{(iii)}{=} 0. \end{aligned}$$

The equality (ii) and equality (iii) hold due to  $h_{i_k} = \|w_{i_k}\|^2$  and the Eq (2.8) respectively. Thus we get that  $x^{(k+1)} - \tilde{x}$  is orthogonal to  $w_{i_k}$ . Therefore, we get that

$$(x^{(k+1)} - x^{(k)})^T (x^{(k+1)} - \tilde{x}) = 0.$$

It follows that

$$\|x^{(k+1)} - \tilde{x}\|^2 = \|x^{(k)} - \tilde{x}\|^2 - \|x^{(k+1)} - x^{(k)}\|^2 \quad (\forall k > 0). \quad (2.11)$$

Thus, from the above proof, the Eq (2.10) holds.

According to the iterative formula

$$\begin{cases} x^{(1)} = x^{(0)} + \frac{b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle}{M(i_1)} a_{i_1}, \\ x^{(k+1)} = x^{(k)} + \alpha_{i_k}^{(k)} w_{i_k} \quad (\forall k > 0), \end{cases}$$

we can get  $P_{N(A)}(x^{(k)}) = P_{N(A)}(x^{(k-1)}) = \dots = P_{N(A)}(x^{(0)})$ , and by the fact that  $P_{N(A)}(x^{(0)}) = P_{N(A)}(\tilde{x})$ , we can deduce that  $x^{(k)} - \tilde{x} \in R(A^T)$ .

### 3. Greedy randomized and maximal weighted residual Kaczmarz methods with oblique projection

In this section, we combine the oblique projection with the GRK method [28] and the MWRK method [21] to obtain the GRKO method and the MWRKO method and prove their convergence. Theoretical results show that the KO-type method can accelerate the convergence when there are suitable row index selection strategies.

#### 3.1. Greedy randomized Kaczmarz method with oblique projection

The core of the GRK method [28] is a new probability criterion, which can grasp the large items of the residual vector in each iteration and randomly select the item with probability in proportion to the retained residual norm. Theories and experiments prove that it can speed up convergence speed. This paper uses the row index selection rule in combination with the Algorithm 1 to obtain the GRKO method and the algorithm is as follows:



**Algorithm 2** Greedy randomized Kaczmarz method with oblique projection (GRKO)**Require:**  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $x^{(0)} \in \mathbb{R}^n$ ,  $K$ 

- 1: For  $i = 1 : m$ ,  $M(i) = \|a_i\|^2$
- 2: Uniformly randomly select  $i_1$  and compute  $x^{(1)} = x^{(0)} + \frac{b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle}{M(i_1)} a_{i_1}$
- 3: **for**  $k = 1, 2, \dots, K - 1$  **do**
- 4:   Compute  $\varepsilon_k = \frac{1}{2} \left( \frac{1}{\|b - Ax^{(k)}\|^2} \max_{1 \leq i_{k+1} \leq m} \left\{ \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\|a_{i_{k+1}}\|^2} \right\} + \frac{1}{\|A\|_F^2} \right)$
- 5:   Determine the index set of positive integers

$$\mathcal{U}_k = \left\{ i_{k+1} \mid |b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2 \geq \varepsilon_k \|b - Ax^{(k)}\|^2 \|a_{i_{k+1}}\|^2 \right\}$$

- 6:   Compute the  $i$ th entry  $\tilde{r}_i^{(k)}$  of the vector  $\tilde{r}^{(k)}$  according to

$$\tilde{r}_i^{(k)} = \begin{cases} b_i - \langle a_i, x^{(k)} \rangle, & \text{if } i \in \mathcal{U}_k \\ 0 & \text{otherwise} \end{cases}$$

- 7:   Select  $i_{k+1} \in \mathcal{U}_k$  with probability  $Pr(\text{row} = i_{k+1}) = \frac{|\tilde{r}_{i_{k+1}}^{(k)}|^2}{\|\tilde{r}^{(k)}\|^2}$
- 8:   Compute  $D_{i_k} = \langle a_{i_k}, a_{i_{k+1}} \rangle$
- 9:   Compute  $w_{i_k} = a_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} a_{i_k}$  and

$$h_{i_k} (= \|w_{i_k}\|^2) = M(i_{k+1}) - \frac{D_{i_k}}{M(i_k)} D_{i_k}$$

- 10:    $\alpha_{i_k}^{(k)} = \frac{\tilde{r}_{i_{k+1}}^{(k)}}{h_{i_k}} \left( = \frac{r_{i_{k+1}}^{(k)}}{h_{i_k}} \right)$  and  $x^{(k+1)} = x^{(k)} + \alpha_{i_k}^{(k)} w_{i_k}$

11: **end for**12: **Output**  $x^{(K)}$ 

The convergence of the GRKO method is provided as follows.

**Theorem 3.1.** Consider the consistent linear system (1.1), where the coefficient matrix  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ . Let  $x^{(0)} \in \mathbb{R}^n$  be an arbitrary initial approximation,  $\tilde{x}$  is a solution of system (1.1) such that  $P_{N(A)}(\tilde{x}) = P_{N(A)}(x^{(0)})$ . Then the iteration sequence  $\{x^{(k)}\}_{k=1}^{\infty}$  generated by the GRKO method obeys

$$E\|x^{(k)} - \tilde{x}\|^2 \leq \prod_{s=0}^{k-1} \zeta_s \|x^{(0)} - \tilde{x}\|^2. \quad (3.1)$$

where  $\zeta_0 = 1 - \frac{\lambda_{\min}(A^T A)}{m\|A\|_F^2}$ ,  $\zeta_1 = 1 - \frac{1}{2} \left( \frac{1}{\gamma_1} \|A\|_F^2 + 1 \right) \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \|A\|_F^2}$ ,  $\zeta_k = 1 - \frac{1}{2} \left( \frac{1}{\gamma_2} \|A\|_F^2 + 1 \right) \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \|A\|_F^2}$  ( $\forall k > 1$ ), which

$$\gamma_1 = \max_{\substack{1 \leq i \leq m \\ s=1 \\ s \neq i}} \sum_{s=1}^m \|a_s\|^2, \quad (3.2)$$

$$\gamma_2 = \max_{\substack{1 \leq i, j \leq m \\ i \neq j}} \sum_{\substack{s=1 \\ s \neq i, j}}^m \|a_s\|^2, \tag{3.3}$$

$$\Delta = \max_{j \neq k} \sin^2 \langle a_j, a_k \rangle (\in (0, 1]). \tag{3.4}$$

In addition, if  $x^{(0)} \in R(A^T)$ , the sequence  $\{x^{(k)}\}_{k=1}^\infty$  converges to the least-norm solution of the system (1.1), i.e.,  $\lim_{k \rightarrow \infty} x^{(k)} = x^* = A^\dagger b$ .

*Proof.* When  $k = 1$ , we can get

$$\begin{aligned} \varepsilon_1 \|A\|_F^2 &= \frac{\max_{1 \leq i_2 \leq m} \left\{ \frac{|b_{i_2} - \langle a_{i_2}, x^{(1)} \rangle|^2}{\|a_{i_2}\|^2} \right\}}{2 \sum_{i_2=1}^m \frac{\|a_{i_2}\|^2}{\|A\|_F^2} \cdot \frac{|b_{i_2} - \langle a_{i_2}, x^{(1)} \rangle|^2}{\|a_{i_2}\|^2}} + \frac{1}{2} \\ &\stackrel{(iv)}{=} \frac{\max_{1 \leq i_2 \leq m} \left\{ \frac{|b_{i_2} - \langle a_{i_2}, x^{(1)} \rangle|^2}{\|a_{i_2}\|^2} \right\}}{2 \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^m \frac{\|a_{i_2}\|^2}{\|A\|_F^2} \cdot \frac{|b_{i_2} - \langle a_{i_2}, x^{(1)} \rangle|^2}{\|a_{i_2}\|^2}} + \frac{1}{2} \\ &\geq \frac{1}{2} \left( \frac{\|A\|_F^2}{\sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^m \|a_{i_2}\|^2} + 1 \right) \\ &\geq \frac{1}{2} \left( \frac{1}{\gamma_1} \|A\|_F^2 + 1 \right). \end{aligned} \tag{3.5}$$

The equality (iv) holds due to the Eq (2.8).

When  $k > 1$ , we get

$$\begin{aligned} \varepsilon_k \|A\|_F^2 &= \frac{\max_{1 \leq i_{k+1} \leq m} \left( \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\|a_{i_{k+1}}\|^2} \right)}{2 \sum_{i_{k+1}=1}^m \frac{\|a_{i_{k+1}}\|^2}{\|A\|_F^2} \cdot \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\|a_{i_{k+1}}\|^2}} + \frac{1}{2} \\ &\stackrel{(v)}{=} \frac{\max_{1 \leq i_{k+1} \leq m} \left( \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\|a_{i_{k+1}}\|^2} \right)}{2 \sum_{\substack{i_{k+1}=1 \\ i_{k+1} \neq i_k, i_{k-1}}}^m \frac{\|a_{i_{k+1}}\|^2}{\|A\|_F^2} \cdot \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\|a_{i_{k+1}}\|^2}} + \frac{1}{2} \\ &\geq \frac{1}{2} \left( \frac{\|A\|_F^2}{\sum_{\substack{i_{k+1}=1 \\ i_{k+1} \neq i_k, i_{k-1}}}^m \|a_{i_{k+1}}\|^2} + 1 \right) \end{aligned} \tag{3.6}$$

$$\geq \frac{1}{2} \left( \frac{1}{\gamma_2} \|A\|_F^2 + 1 \right).$$

The equality (v) holds due to the Eqs (2.8) and (2.9).

Under the GRKO method, Lemma 2.2 still holds, so we can take the full expectation on both sides of the Eq (2.10) and get that for  $k = 0$ ,

$$\begin{aligned} E\|x^{(1)} - \tilde{x}\|^2 &= \|x^{(0)} - \tilde{x}\|^2 - E\|x^{(1)} - x^{(0)}\|^2 \\ &= \|x^{(0)} - \tilde{x}\|^2 - \frac{1}{m} \sum_{i_1=1}^m \left\| \frac{b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle}{M(i_1)} a_{i_1} \right\|^2 \\ &\stackrel{(vi)}{\leq} \|x^{(0)} - \tilde{x}\|^2 - \frac{1}{m} \frac{\|b - Ax^{(0)}\|^2}{\|A\|_F^2} \\ &\stackrel{(vii)}{\leq} \left( 1 - \frac{\lambda_{\min}(A^T A)}{m\|A\|_F^2} \right) \|x^{(0)} - \tilde{x}\|^2 \\ &= \zeta_0 \|x^{(0)} - \tilde{x}\|^2, \end{aligned} \quad (3.7)$$

and for  $k > 0$ ,

$$\begin{aligned} E_k\|x^{(k+1)} - \tilde{x}\|^2 &= \|x^{(k)} - \tilde{x}\|^2 - E_k\|x^{(k+1)} - x^{(k)}\|^2 \\ &= \|x^{(k)} - \tilde{x}\|^2 - \sum_{i_{k+1} \in \mathcal{U}_k} \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\sum_{i_{k+1} \in \mathcal{U}_k} |b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2} \cdot \frac{|r_{i_{k+1}}^{(k)}|^2}{\|w_{i_k}\|^2} \\ &\stackrel{(viii)}{\leq} \|x^{(k)} - \tilde{x}\|^2 - \sum_{i_{k+1} \in \mathcal{U}_k} \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\sum_{i_{k+1} \in \mathcal{U}_k} |b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2} \cdot \frac{|r_{i_{k+1}}^{(k)}|^2}{\Delta \cdot \|a_{i_{k+1}}\|^2} \\ &\stackrel{(ix)}{\leq} \|x^{(k)} - \tilde{x}\|^2 - \frac{\varepsilon_k}{\Delta} \|b - Ax^{(k)}\|^2 \\ &= \|x^{(k)} - \tilde{x}\|^2 - \frac{\varepsilon_k}{\Delta} \|A(\tilde{x} - x^{(k)})\|^2 \\ &\stackrel{(vii)}{\leq} \left( 1 - \frac{\varepsilon_k \lambda_{\min}(A^T A)}{\Delta} \right) \|x^{(k)} - \tilde{x}\|^2. \end{aligned} \quad (3.8)$$

The inequality (vi) of the Eq (3.7) is achieved with the use of the fact that  $\frac{|b_1|}{|a_1|} + \frac{|b_2|}{|a_2|} \geq \frac{|b_1|+|b_2|}{|a_1|+|a_2|}$  (if  $|a_1| > 0, |a_2| > 0$ ) and the inequality (viii) of the Eq (3.8) is achieved with the use of the fact that

$$\begin{aligned} \|w_{i_k}\|^2 &= \|a_{i_{k+1}}\|^2 - \frac{\langle a_{i_k}, a_{i_{k+1}} \rangle^2}{\|a_{i_k}\|^2} \\ &= \sin^2 \langle a_{i_k}, a_{i_{k+1}} \rangle \|a_{i_{k+1}}\|^2 \\ &\leq \Delta \cdot \|a_{i_{k+1}}\|^2, \end{aligned} \quad (3.9)$$

and the inequality (ix) of the Eq (3.8) is achieved with the use of the definition of  $\mathcal{U}_k$  which lead to

$$|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2 \geq \varepsilon_k \|b - Ax^{(k)}\|^2 \|a_{i_{k+1}}\|^2, \forall i_{k+1} \in \mathcal{U}_k.$$

Here in the last inequalities (vii) of the Eqs (3.7) and (3.8), we have used the estimate  $\|Au\|_2^2 \geq \lambda_{\min}(A^T A)\|u\|^2$ , which holds true for any  $u \in C^n$  belonging to the column space of  $A^T$ . According to the lemma 2.2, it holds.

By making use of the Eqs (3.5), (3.6) and (3.8), we get

$$\begin{aligned} E_1 \|x^{(2)} - \tilde{x}\|^2 &\leq \left[ 1 - \frac{1}{2} \left( \frac{1}{\gamma_1} \|A\|_F^2 + 1 \right) \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \|A\|_F^2} \right] \|x^{(1)} - \tilde{x}\|^2 \\ &= \zeta_1 \|x^{(1)} - \tilde{x}\|^2, \end{aligned}$$

$$\begin{aligned} E_k \|x^{(k+1)} - \tilde{x}\|^2 &\leq \left[ 1 - \frac{1}{2} \left( \frac{1}{\gamma_2} \|A\|_F^2 + 1 \right) \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \|A\|_F^2} \right] \|x^{(k)} - \tilde{x}\|^2 \\ &= \zeta_k \|x^{(k)} - \tilde{x}\|^2 \quad (\forall k > 1). \end{aligned}$$

Finally, by recursion and taking the full expectation, the inequality (3.1) holds.

**Remark 3.** In the GRKO method,  $h_{i_k}$  is not zero. Suppose  $h_{i_k} = 0$ , which means  $\exists \lambda > 0$ ,  $\lambda a_{i_k} = a_{i_{k+1}}$ . Due to the system is consistent, it holds  $\langle a_{i_{k+1}}, x^* \rangle = \lambda \langle a_{i_k}, x^* \rangle = \lambda b_{i_k} = b_{i_{k+1}}$ . According to the Eq (2.8), it holds  $r_{i_{k+1}}^{(k)} = \lambda r_{i_k}^{(k)} = 0$ . From step 5 of Algorithm 1, we can know that such index  $i_{k+1}$  will not be selected.

**Remark 4.** Set  $\tilde{\zeta}_k = 1 - \frac{1}{2} \left( \frac{1}{\gamma_1} \|A\|_F^2 + 1 \right) \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2}$  ( $\forall k > 0$ ) and the convergence of GRK method in [28] meets:

$$E_k \|x^{(k+1)} - x^*\|^2 \leq \tilde{\zeta}_k \|x^{(k)} - x^*\|^2.$$

Obviously,  $\forall \Delta \in (0, 1]$ ,  $\zeta_1 \leq \tilde{\zeta}_1$ ,  $\zeta_k < \tilde{\zeta}_k$  ( $\forall k > 1$ ) is satisfied, so the convergence speed of GRKO method is faster than GRK method.

**Remark 5.** In fact, in each iteration, the most computationally expensive part is computing the residual  $r^{(k)}$ . If  $B = AA^T$  is calculated before iteration, the GRK method [28] costs  $7m + 2n + 2$  flopping operations and the GRKO method costs  $9m + 3n + 6$  flopping operations, where the residual  $r^{(k)}$  is calculated according to Eq (2.2).

### 3.2. Maximal weighted residual Kaczmarz method with oblique projection

The selection strategy for the index  $i_k$  used in the maximal weighted residual Kaczmarz (MWRK) method [21] is: Set

$$i_k = \arg \max_{i \in \{1, 2, \dots, m\}} \frac{|a_i^T x^{(k)} - b_i|}{\|a_i\|}.$$

Mccormick proved the exponential convergence of the MWRK method. In [22], a new convergence conclusion of the MWRK method is given. We use its row index selection rule combined with KO-type method to obtain MWRKO method and the algorithm is as follows:

**Algorithm 3** Maximal Weighted Residual Kaczmarz Method with Oblique Projection (MWRKO)

**Require:**  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $x^{(0)} \in \mathbb{R}^n$ ,  $K$

- 1: For  $i = 1 : m$ ,  $M(i) = \|a_i\|^2$
- 2: Compute  $i_1 = \arg \max_{i \in \{1, 2, \dots, m\}} \frac{|a_i^T x^{(0)} - b_i|}{\|a_i\|}$  and  $x^{(1)} = x^{(0)} + \frac{b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle}{M(i_1)} a_{i_1}$
- 3: **for**  $k = 1, 2, \dots, K$  **do**
- 4:   Compute  $i_{k+1} = \arg \max_{i \in \{1, 2, \dots, m\}} \frac{|a_i^T x^{(k)} - b_i|}{\|a_i\|}$
- 5:   Compute  $D_{i_k} = \langle a_{i_k}, a_{i_{k+1}} \rangle$  and  $r_{i_{k+1}}^{(k)} = b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle$
- 6:   Compute  $w_{i_k} = a_{i_{k+1}} - \frac{D_{i_k}}{M(i_k)} a_{i_k}$  and  $h_{i_k} (= \|w_{i_k}\|^2) = M(i_{k+1}) - \frac{D_{i_k}^2}{M(i_k)}$
- 7:    $\alpha_{i_k}^{(k)} = \frac{r_{i_{k+1}}^{(k)}}{h_{i_k}}$  and  $x^{(k+1)} = x^{(k)} + \alpha_{i_k}^{(k)} w_{i_k}$
- 8: **end for**
- 9: Output  $x^{(K+1)}$

The convergence of the MWRKO method is provided as follows.

**Theorem 3.2.** Consider the consistent linear system (1.1), where the coefficient matrix  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ . Let  $x^{(0)} \in \mathbb{R}^n$  be an arbitrary initial approximation,  $\tilde{x}$  is a solution of system (1.1) such that  $P_{N(A)}(\tilde{x}) = P_{N(A)}(x^{(0)})$ . Then the iteration sequence  $\{x^{(k)}\}_{k=1}^{\infty}$  generated by the MWRKO method obeys

$$\|x^{(k)} - \tilde{x}\|^2 \leq \prod_{s=0}^{k-1} \rho_s \|x^{(0)} - \tilde{x}\|^2, \quad (3.10)$$

where  $\rho_0 = 1 - \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2}$ ,  $\rho_1 = 1 - \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \gamma_1}$ ,  $\rho_k = 1 - \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \gamma_2}$  ( $\forall k > 1$ ), which  $\gamma_1$ ,  $\gamma_2$  and  $\Delta$  are defined by Eqs (3.2), (3.3) and (3.4) respectively.

In addition, if  $x^{(0)} \in R(A^T)$ , the sequence  $\{x^{(k)}\}_{k=1}^{\infty}$  converges to the least-norm solution of the system (1.1), i.e.,  $\lim_{k \rightarrow \infty} x^{(k)} = x^* = A^\dagger b$ .

*Proof.* Under the MWRKO method, Lemma 2.2 still holds. For  $k = 1$ , we have

$$\begin{aligned} \|x^{(1)} - \tilde{x}\|^2 &= \|x^{(0)} - \tilde{x}\|^2 - \|x^{(1)} - x^{(0)}\|^2 \\ &= \|x^{(0)} - \tilde{x}\|^2 - \frac{|b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle|^2}{M(i_1)} \\ &= \|x^{(0)} - \tilde{x}\|^2 - \frac{|b_{i_1} - \langle a_{i_1}, x^{(0)} \rangle|^2}{M(i_1)} \cdot \frac{\|b - Ax^{(0)}\|^2}{\sum_{i=1}^m \frac{|b_i - \langle a_i, x^{(0)} \rangle|^2}{M(i)} \cdot M(i)} \\ &\leq \|x^{(0)} - \tilde{x}\|^2 - \frac{\|A(\tilde{x} - x^{(0)})\|^2}{\|A\|_F^2} \\ &\stackrel{(x)}{\leq} \|x^{(0)} - \tilde{x}\|^2 - \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2} \|x^{(0)} - \tilde{x}\|^2 \end{aligned} \quad (3.11)$$

$$\begin{aligned}
&= \left(1 - \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2}\right) \|x^{(0)} - x^*\|^2 \\
&= \rho_0 \|x^{(0)} - x^*\|^2.
\end{aligned}$$

For  $k = 1$ , we have

$$\begin{aligned}
\|x^{(2)} - \tilde{x}\|^2 &= \|x^{(1)} - \tilde{x}\|^2 - \|x^{(2)} - x^{(1)}\|^2 \\
&= \|x^{(1)} - \tilde{x}\|^2 - \frac{|b_{i_2} - \langle a_{i_2}, x^{(1)} \rangle|^2}{\|w_{i_1}\|^2} \\
&\stackrel{(xi)}{\leq} \|x^{(1)} - \tilde{x}\|^2 - \frac{|b_{i_2} - \langle a_{i_2}, x^{(1)} \rangle|^2}{\Delta \cdot M(i_2)} \cdot \frac{\|b - Ax^{(1)}\|^2}{\sum_{i=1, i \neq i_1}^m \frac{|b_i - \langle a_i, x^{(1)} \rangle|^2}{M(i)} \cdot M(i)} \quad (3.12) \\
&\stackrel{(xii)}{\leq} \|x^{(1)} - \tilde{x}\|^2 - \frac{\|A(\tilde{x} - x^{(1)})\|^2}{\Delta \cdot \gamma_1} \\
&\stackrel{(x)}{\leq} \|x^{(1)} - \tilde{x}\|^2 - \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \gamma_1} \|x^{(1)} - \tilde{x}\|^2 \\
&= \left(1 - \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \gamma_1}\right) \|x^{(1)} - \tilde{x}\|^2 \\
&= \rho_1 \|x^{(1)} - \tilde{x}\|^2,
\end{aligned}$$

where the inequality (xi) can be obtained by using Eqs (3.9) and (2.8). For inequality (xii), using the row index selection rule of the MWRKO method, we get:

$$\begin{aligned}
&\frac{|b_{i_2} - \langle a_{i_2}, x^{(1)} \rangle|^2}{\Delta \cdot M(i_2)} \cdot \frac{\|b - Ax^{(1)}\|^2}{\sum_{i=1, i \neq i_1}^m \frac{|b_i - \langle a_i, x^{(1)} \rangle|^2}{M(i)} \cdot M(i)} \\
&= \max_{i \in \{1, 2, \dots, m\}} \frac{|b_i - \langle a_i, x^{(1)} \rangle|^2}{\Delta \cdot M(i)} \cdot \frac{\|b - Ax^{(1)}\|^2}{\sum_{i=1, i \neq i_1}^m \frac{|b_i - \langle a_i, x^{(1)} \rangle|^2}{M(i)} \cdot M(i)} \quad (3.13) \\
&\geq \frac{\|b - Ax^{(1)}\|^2}{\Delta \cdot \sum_{i=1, i \neq i_1}^m M(i)} \\
&\geq \frac{\|b - Ax^{(1)}\|^2}{\Delta \cdot \gamma_1}.
\end{aligned}$$

For  $k > 1$ , we have

$$\begin{aligned}
\|x^{(k+1)} - \tilde{x}\|^2 &= \|x^{(k)} - \tilde{x}\|^2 - \|x^{(k+1)} - x^{(k)}\|^2 \\
&= \|x^{(k)} - \tilde{x}\|^2 - \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\|w_{i_k}\|^2} \\
&\stackrel{(xiii)}{\leq} \|x^{(k)} - \tilde{x}\|^2 - \frac{|b_{i_{k+1}} - \langle a_{i_{k+1}}, x^{(k)} \rangle|^2}{\Delta \cdot M(i_{k+1})} \cdot \frac{\|b - Ax^{(k)}\|^2}{\sum_{i=1, i \neq i_k, i_{k-1}}^m \frac{|b_i - \langle a_i, x^{(k)} \rangle|^2}{M(i)} \cdot M(i)} \\
&\stackrel{(xiv)}{\leq} \|x^{(k)} - \tilde{x}\|^2 - \frac{\|A(\tilde{x} - x^{(k)})\|^2}{\Delta \cdot \gamma_2} \\
&\stackrel{(x)}{\leq} \|x^{(k)} - \tilde{x}\|^2 - \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \gamma_2} \|x^{(k)} - \tilde{x}\|^2 \\
&= \left(1 - \frac{\lambda_{\min}(A^T A)}{\Delta \cdot \gamma_2}\right) \|x^{(k)} - \tilde{x}\|^2 \\
&= \rho_k \|x^{(k)} - \tilde{x}\|^2,
\end{aligned} \tag{3.14}$$

where the inequality (xiii) can be obtained by using Eqs (3.9), (2.8) and (2.9). For the inequality (xiv), it can be easily obtained by using a derivation similar to Eq (3.13). In the inequalities (x) of the Eqs (3.11), (3.12) and (3.14), we have used the estimate

$$\|Au\|_2^2 \geq \lambda_{\min}(A^T A) \|u\|^2,$$

which holds true for any  $u \in C^n$  belonging to the column space of  $A^T$ . According to the lemma 2.2, it holds.

From the Eqs (3.11), (3.12) and (3.14), the Eq (3.10) holds.

**Remark 6.** When multiple indicators  $i_{k+1}$  are met in Step 2 of Algorithm 2 in the iterative process, we randomly select any one of them.

**Remark 7.** In the MWRKO method, the reason of  $h_{i_k} \neq 0$  is similar to Remark 3.

**Remark 8.** Set  $\tilde{\rho}_0 = 1 - \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2}$ ,  $\tilde{\rho}_k = 1 - \frac{\lambda_{\min}(A^T A)}{\gamma_1}$  ( $\forall k > 0$ ), and the convergence of MWRK method in [22] meets:

$$\|x^{(k)} - x^*\|^2 \leq \prod_{s=0}^{k-1} \tilde{\rho}_s \|x^{(0)} - x^*\|^2.$$

Obviously,  $\forall \Delta \in (0, 1]$ ,  $\rho_k < \tilde{\rho}_k$  ( $\forall k > 1$ ),  $\rho_1 \leq \tilde{\rho}_1$  and  $\rho_0 = \tilde{\rho}_0$ , so the convergence speed of MWRKO method is faster than MWRK method. Note that  $\tilde{\rho}_k < \tilde{\zeta}_k$ ,  $\rho_k < \zeta_k$  ( $\forall k > 0, \forall \Delta \in (0, 1]$ ), that is  $V_{MWRK} < V_{MWRKO}$ ,  $V_{GRK} < V_{GRKO}$ ,  $V_{GRK} < V_{MWRK}$ ,  $V_{GRKO} < V_{MWRKO}$ , where  $V$  represents the convergence speed.

**Remark 9.** If  $B = AA^T$  is calculated before iteration, the MWRK method costs  $4m + 2n$  flopping operations and the MWRKO method costs  $6m + 3n + 4$  flopping operations, where the residual  $r^{(k)}$  is calculated according to Eq (2.2).

#### 4. Numerical experiments

In this section, some numerical examples are provided to illustrate the effectiveness of the greedy randomized Kaczmarz (GRK) method, the greedy randomized Kaczmarz method with oblique projection (GRKO), the maximal weighted residual Kaczmarz method (MWRK) and the maximal weighted residual Kaczmarz method (MWRKO). All experiments are carried out using MATLAB (version R2019b) on a personal laptop with 1.60 GHz central processing unit (Intel(R) Core(TM) i5-10210U CPU), 8.00 GB memory and Windows operating system (64 bit Windows 10).

In our implementations, the right vector  $b = Ax^*$  such that the exact solution  $x^* \in \mathbb{R}^n$  is a vector generated by the *rand* function. Define the relative residual error (RRE) at the  $k$ th iteration as follows:

$$\text{RRE} = \frac{\|b - Ax^{(k)}\|^2}{\|b\|^2}.$$

The initial point  $x^{(0)} \in \mathbb{R}^n$  is set to be a zero vector and the iterations are terminated once the relative solution error satisfies  $\text{RRE} < \omega$  or the number of iteration steps exceeds 100,000. If the number of iteration steps exceeds 100,000, it is denoted as "-".

We will compare the numerical performance of these methods in terms of the number of iteration steps (denoted as "IT") and the computing time in seconds (denoted as "CPU"). Here the CPU and IT mean the arithmetical averages of the elapsed running times and the required iteration steps with respect to 50 trials repeated runs of the corresponding method.

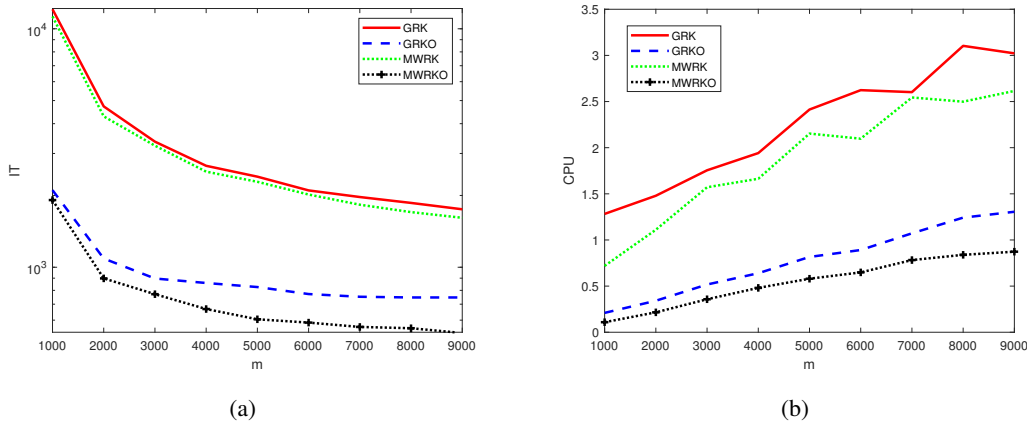
##### 4.1. Experiments for random matrix collection in $[0, 1]$

The random matrix collection in  $[0, 1]$  is randomly generated by using the MATLAB function *rand* and the numerical results are reported in Tables 1 and 2 and Figures 2 and 3. In this subsection, we let  $\omega = 0.5 \times 10^{-8}$ . According to the characteristics of the matrix generated by MATLAB function *rand*, Tables 1 and 2 are the experiments for the overdetermined consistent linear systems, underdetermined consistent linear systems respectively. Under the premise of convergence, all methods can find the unique least Euclidean norm solution.

**Table 1.** IT and CPU of GRK, GRKO, MWRK and MWRKO for  $m \times n$  matrices  $A$  with  $n = 500$  and different  $m$  when the consistent linear system is overdetermined.

$m$	IT				CPU			
	GRK	GRKO	MWRK	MWRKO	GRK	GRKO	MWRK	MWRKO
1000	12,072	2105	11,265	1913	1.2824	0.2099	0.7192	0.1089
2000	4726	1088	4292	898	1.4792	0.3413	1.1107	0.2157
3000	3362	897	3234	771	1.7550	0.5172	1.5711	0.3575
4000	2663	859	2517	668	1.9415	0.6396	1.6634	0.4807
5000	2398	826	2282	605	2.4134	0.8160	2.1528	0.5801
6000	2100	772	2018	586	2.6235	0.8912	2.0975	0.6486
7000	1970	752	1829	562	2.6019	1.0720	2.5441	0.7822
8000	1861	747	1703	555	3.1035	1.2421	2.4987	0.8390
9000	1750	747	1612	530	3.0223	1.3055	2.6148	0.8730

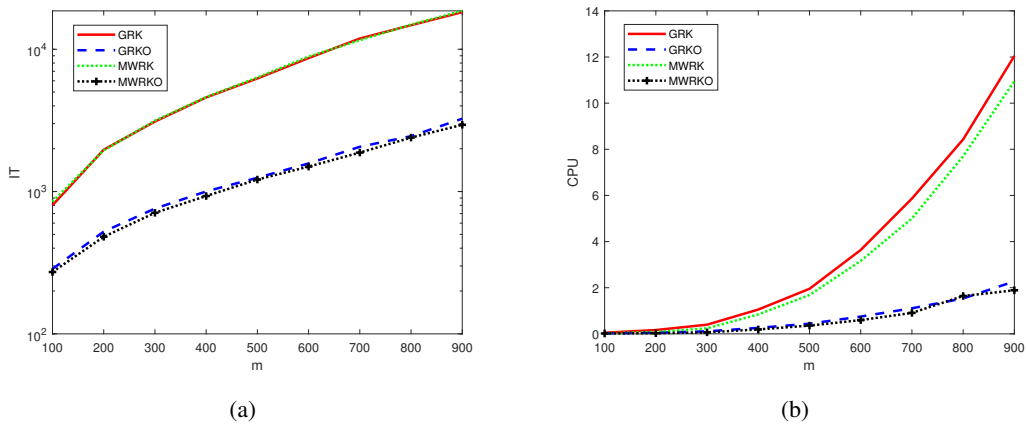




**Figure 2.** (a) IT and (b) CPU versus  $m$  for four methods with matrices  $A \in \mathbb{R}^{m \times 500}$  generated by the *rand* function in the interval  $[0, 1]$ .

**Table 2.** IT and CPU of GRK, GRKO, MWRK and MWRKO for  $m \times n$  matrices  $A$  with  $n = 2000$  and different  $m$  when the consistent linear system is underdetermined.

$m$	IT				CPU			
	GRK	GRKO	MWRK	MWRKO	GRK	GRKO	MWRK	MWRKO
100	802	286	848	272	0.0496	0.0223	0.0258	0.0165
200	1968	523	1948	481	0.1648	0.0496	0.0831	0.0276
300	3104	759	3148	709	0.3982	0.1090	0.2404	0.0664
400	4586	1002	4612	930	1.0539	0.2594	0.8433	0.1920
500	6233	1250	6336	1215	1.9528	0.4409	1.6836	0.3576
600	8671	1576	8882	1497	3.6363	0.7493	3.1625	0.5957
700	11,895	2063	11,575	1879	5.8642	1.1078	5.0029	0.9087
800	14,758	2451	14,888	2394	8.4280	1.5350	7.7007	1.6405
900	18,223	3250	18,608	2945	12.0469	2.2750	10.9511	1.8884



**Figure 3.** (a) IT and (b) CPU versus  $m$  for four methods with matrices  $A \in \mathbb{R}^{m \times 2000}$  generated by the *rand* function in the interval  $[0, 1]$ .

From Table 1 and Figure 2, we can see that when the linear system is overdetermined, with the increase of  $m$ , the IT of all methods decreases, but the CPU shows an increasing trend. Our new methods—the GRKO method and the MWRKO method, perform better than the GRK method and the MWRK method respectively in both iteration steps and running time. Among the four methods, the MWRKO method performs best. From Table 2 and Figure 3, we can see that in the case of underdetermined linear system, with the increase of  $m$ , the IT and CPU of all methods decrease.

In this group of experiments, whether it is an overdetermined or underdetermined linear system, whether in terms of the IT or CPU, the GRKO method and the MWRKO method perform very well compared with the GRK method and the MWRK method. These experimental phenomena are consistent with the theoretical convergence conclusions we got.

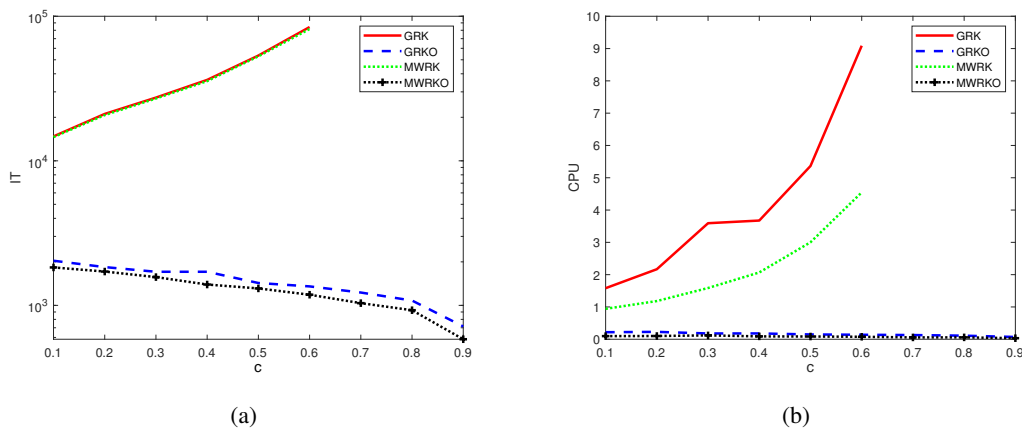
#### 4.2. Experiments for random matrix collection in $[c, 1]$

In this subsection, the entries of our coefficient matrix are randomly generated in the interval  $[c, 1]$ . This set of experiments was also done in [36] and [54] and pointed out that when the value of  $c$  is close to 1, the rows of matrix  $A$  are more linearly correlated. Theorems 3.1 and 3.2 have shown the effectiveness of the GRKO method and the MWRKO method in this case. In order to verify this phenomenon, we construct several  $1000 \times 500$  and  $500 \times 1000$  matrices  $A$ , which entries is independent identically distributed uniform random variables on some interval  $[c, 1]$ . Note that there is nothing special about this interval and other intervals yield the same results when the interval length remains the same. In the experiment of this subsection, we take  $\omega = 0.5 \times 10^{-8}$ .

**Table 3.** IT and CPU of GRK, GRKO, MWRK and MWRKO for matrices  $A \in \mathbb{R}^{1000 \times 500}$  generated by the *rand* function in the interval  $[c, 1]$ .

$c$	IT				CPU			
	GRK	GRKO	MWRK	MWRKO	GRK	GRKO	MWRK	MWRKO
0.1	14,757	2036	14,594	1830	1.5811	0.2180	0.9419	0.0969
0.2	21,103	1840	20,717	1714	2.1684	0.2287	1.1828	0.1003
0.3	27,375	1708	26,986	1569	3.5926	0.1789	1.5865	0.1195
0.4	36,293	1708	35,595	1394	3.6751	0.1802	2.0682	0.0885
0.5	53,485	1428	52,853	1310	5.3642	0.1486	3.0024	0.0847
0.6	84,204	1353	81,647	1185	9.0879	0.1388	4.5468	0.0767
0.7	-	1227	-	1036	-	0.1298	-	0.0564
0.8	-	1080	-	926	-	0.1107	-	0.0580
0.9	-	715	-	583	-	0.0707	-	0.0324

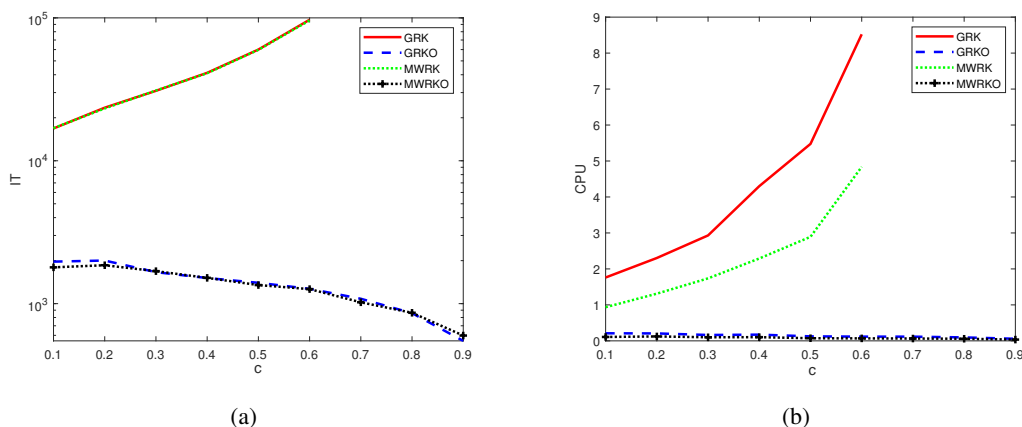
From Table 3 and Figure 4, it can be seen that when the linear system is overdetermined, with  $c$  getting closer to 1, the GRK method and the MWRK method have a significant increase in the number of iterations and running time. When  $c$  increases to 0.7, the GRK method and the MWRK method exceeds the maximum number of iterations. But the IT and CPU of the GRKO method and the MWRKO method have decreasing trends. From Figure 5, we can get that the numerical experiment of the coefficient matrix  $A$  in the underdetermined case has similar trends to the numerical experiment in the overdetermined case in Figure 4. In this group of experiments, it can be observed that when the rows of the matrix are more linearly correlated, the GRKO method and the MWRKO method can find the least Euclidean norm solution more quickly than the GRK method and the MWRK method.



**Figure 4.** (a) IT and (b) CPU versus  $c$  for four methods with matrices  $A \in \mathbb{R}^{1000 \times 500}$  generated by the *rand* function in the interval  $[c, 1]$ .

**Table 4.** IT and CPU of GRK, GRKO, MWRK and MWRKO for matrices  $A \in \mathbb{R}^{500 \times 1000}$  generated by the *rand* function in the interval  $[c, 1]$ .

$c$	IT				CPU			
	GRK	GRKO	MWRK	MWRKO	GRK	GRKO	MWRK	MWRKO
0.1	16,828	1968	16,913	1795	1.7612	0.2103	0.9353	0.1083
0.2	23,518	2003	23,234	1857	2.3037	0.2066	1.3119	0.1230
0.3	30,875	1661	31,017	1688	2.9310	0.1635	1.7373	0.0997
0.4	41,242	1511	40,986	1515	4.3004	0.1726	2.2899	0.1025
0.5	60,000	1399	59,750	1349	5.4754	0.1252	2.8920	0.0727
0.6	97,045	1270	95,969	1264	8.5229	0.1173	4.8380	0.0688
0.7	-	1082	-	1022	-	0.1168	-	0.0646
0.8	-	858	-	863	-	0.0960	-	0.0585
0.9	-	549	-	598	-	0.0582	-	0.0353



**Figure 5.** (a) IT and (b) CPU versus  $c$  for four methods with matrices  $A \in \mathbb{R}^{500 \times 1000}$  generated by the *rand* function in the interval  $[c, 1]$ .

### 4.3. Experiments for sparse matrix

In this subsection, we will give three examples to illustrate the effectiveness of our new methods applied to sparse matrix. The coefficient matrices  $A$  of these three examples are the practical problems from [57] and the two test problems from [58].

**Example 1.** We solve the problem (1.1) with the coefficient matrix  $A \in \mathbb{R}^{m \times n}$  chosen from the University of Florida sparse matrix collection [57]. the matrices are *divorce*, *photogrammetry*, *Ragusa18*, *Trec8*, *Stranke94* and *well1033*. In Table 5, we list some properties of these matrices, where density is defined as follows:

$$\text{density} = \frac{\text{number of nonzeros of m-by-n matrix}}{mn}.$$

In this group of experiments, we set  $\omega = 0.5 \times 10^{-5}$ . In order to solve Example 1, we list the IT, CPU and historical convergence of the GRK, GRKO, MWRK and MWRKO methods in Figure 6 and Table 6, respectively. It can be seen that IT and CPU of the MWRKO method are the least. Although the GRKO method is not faster than the MWRK method for most of the experiments in Table 6, it is always faster than the GRK method.

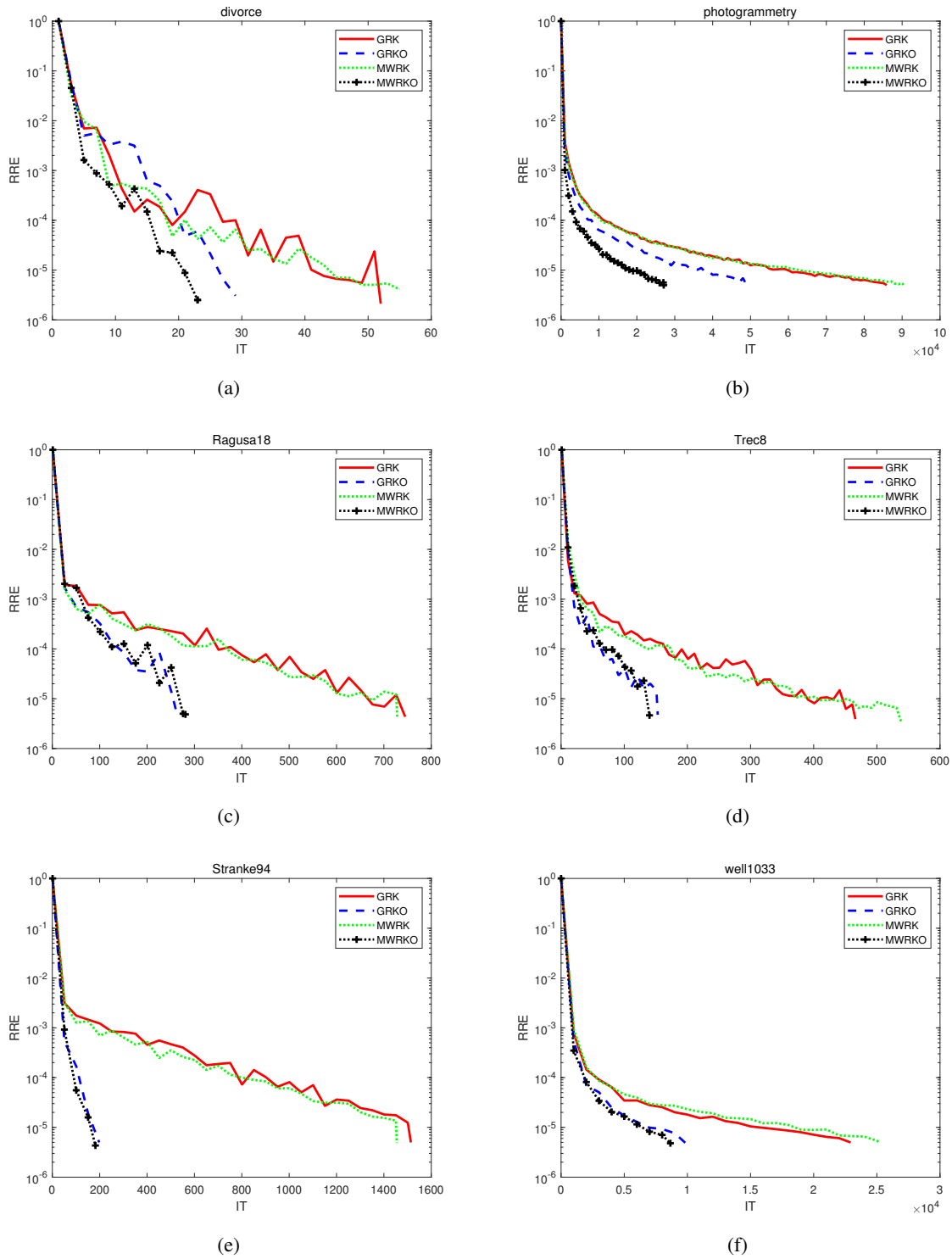
**Example 2.** We consider *fancurvedtomo*( $N, \theta, P$ ) test problem from the MATLAB package AIR Tools [58], which generates sparse matrix  $A$ , an exact solution  $x^*$  and  $b = Ax^*$ . We set  $N = 60$ ,  $\theta = 0 : 0.5 : 179.5^\circ$ ,  $P = 90$ , then resulting matrix is of size  $32400 \times 3600$ . We test RRE every 10 iterations and run these four methods until  $\text{RRE} < \omega$  is satisfied, where  $\omega = 0.5 \times 10^{-5}$ .

We first remove the rows of  $A$  where the entries are all 0 and perform row unitization processing on  $A$  and  $b$ . We emphasize that this will not cause a change in  $x^*$ . In Figure 7, we give  $60 \times 60$  images of the exact phantom and the approximate solutions obtained by the GRK, GRKO, MWRK, MWRKO methods. In Figure 7, we can see that the GRK, GRKO, MWRK and MWRKO method can all converge to the exact solution successfully. In the subgraph (f) of Figure 7, we can see that the MWRKO method needs the least iterative steps and the GRKO method has less iterative steps than the GRK method. It can be observed from Table 7 that the MWRKO method is the best in terms of IT and CPU.

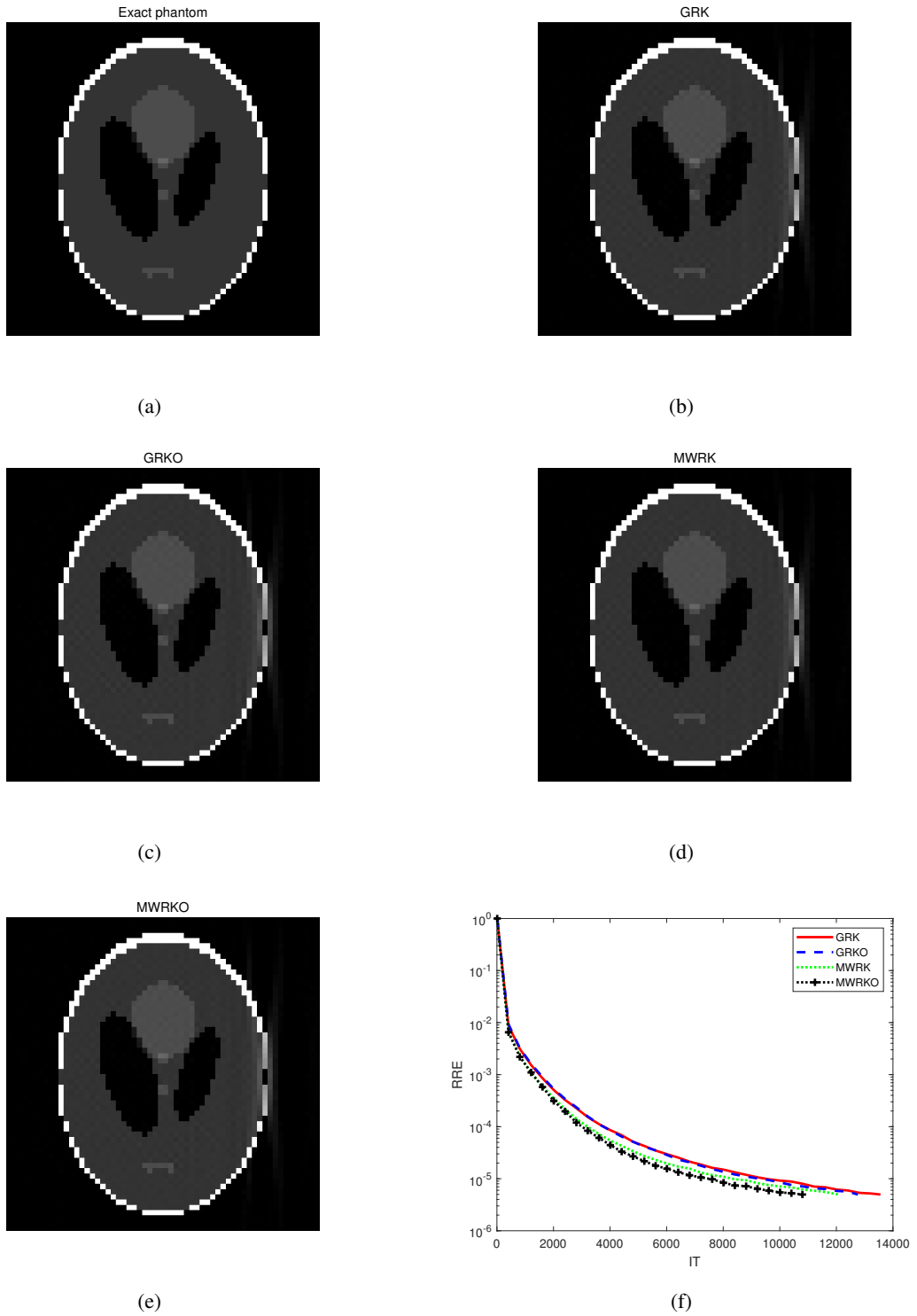
**Example 3.** We use an example from 2D seismic travel-time tomography reconstruction, implemented in the function *seismictomo*( $N, s, p$ ) in the MATLAB package AIR Tools [58], which generates sparse matrix  $A$ , an exact solution  $x^*$  and  $b = Ax^*$ . We set  $N = 50$ ,  $s = 150$ ,  $p = 120$ , then resulting matrix is of size  $18000 \times 2500$ . We test RRE every 50 iterations and run these four methods until  $\text{RRE} < \omega$  is satisfied, where  $\omega = 0.5 \times 10^{-6}$ .

**Table 5.** The properties of different sparse matrices.

A	divorce	photogrammetry	Ragusa18	Trec8	Stranke94	well1033
$m \times n$	$50 \times 9$	$1388 \times 390$	$23 \times 23$	$23 \times 84$	$10 \times 10$	$1033 \times 320$
rank	9	390	15	23	10	320
cond( $A$ )	19.3908	4.35e+8	3.48e+35	26.8949	51.7330	166.1333
density	50.00%	2.18%	12.10%	28.42%	90.00%	1.43%



**Figure 6.** Convergence history of the four methods for sparse matrices. (a) divorce, (b) photogrammetry, (c)Ragusa18, (d) Trec8, (e) Stranke94, (f) well1033.



**Figure 7.** Performance of the (a) exact phantom, (b) GRK, (c) GRKO, (d) MWRK, (e) MWRKO methods for *fancurvedtomo* test problem. (f) Convergence test history test of test four test methods.

**Table 6.** IT and CPU of GRK, GRKO, MWRK and MWRKO for different sparse matrices.

A	IT				CPU			
	GRK	GRKO	MWRK	MWRKO	GRK	GRKO	MWRK	MWRKO
divorce	51	28	54	22	0.0053	0.0037	0.0017	0.0013
photogrammetry	85,938	48,933	90,480	27,084	9.9917	8.0424	3.9809	2.5026
Ragusa18	744	262	727	280	0.0577	0.0270	0.0121	0.0098
Trec8	465	152	538	139	0.0382	0.0168	0.0111	0.0062
Stranke94	1513	197	1453	181	0.1291	0.0187	0.0208	0.0082
well1033	22,924	9825	25,250	8655	2.4278	1.5112	0.8491	0.5827

**Table 7.** IT and CPU of GRK, GRKO, MWRK and MWRKO for *f ancurved tomo* problem.

method	IT	CPU
GRK	13,550	581.17
GRKO	12,750	538.82
MWRK	12,050	504.83
MWRKO	10,790	452.86

**Table 8.** IT and CPU of GRK, GRKO, MWRK and MWRKO for *seismictomo* problem.

method	IT	CPU
GRK	21,500	378.6629
GRKO	15,850	324.9338
MWRK	19,250	334.3446
MWRKO	14,400	262.1489

We first remove the rows of  $A$  where the entries are all 0 and perform row unitization processing on  $A$  and  $b$ . In Figure 8, we give  $50 \times 50$  images of the exact phantom and the approximate solutions obtained by the GRK, GRKO, MWRK, MWRKO methods. From the subgraph (f) of Figure 8 and Table 8, we can see that the MWRKO method is the best in terms of IT and CPU.

#### 4.4. Comparison experiments with other methods

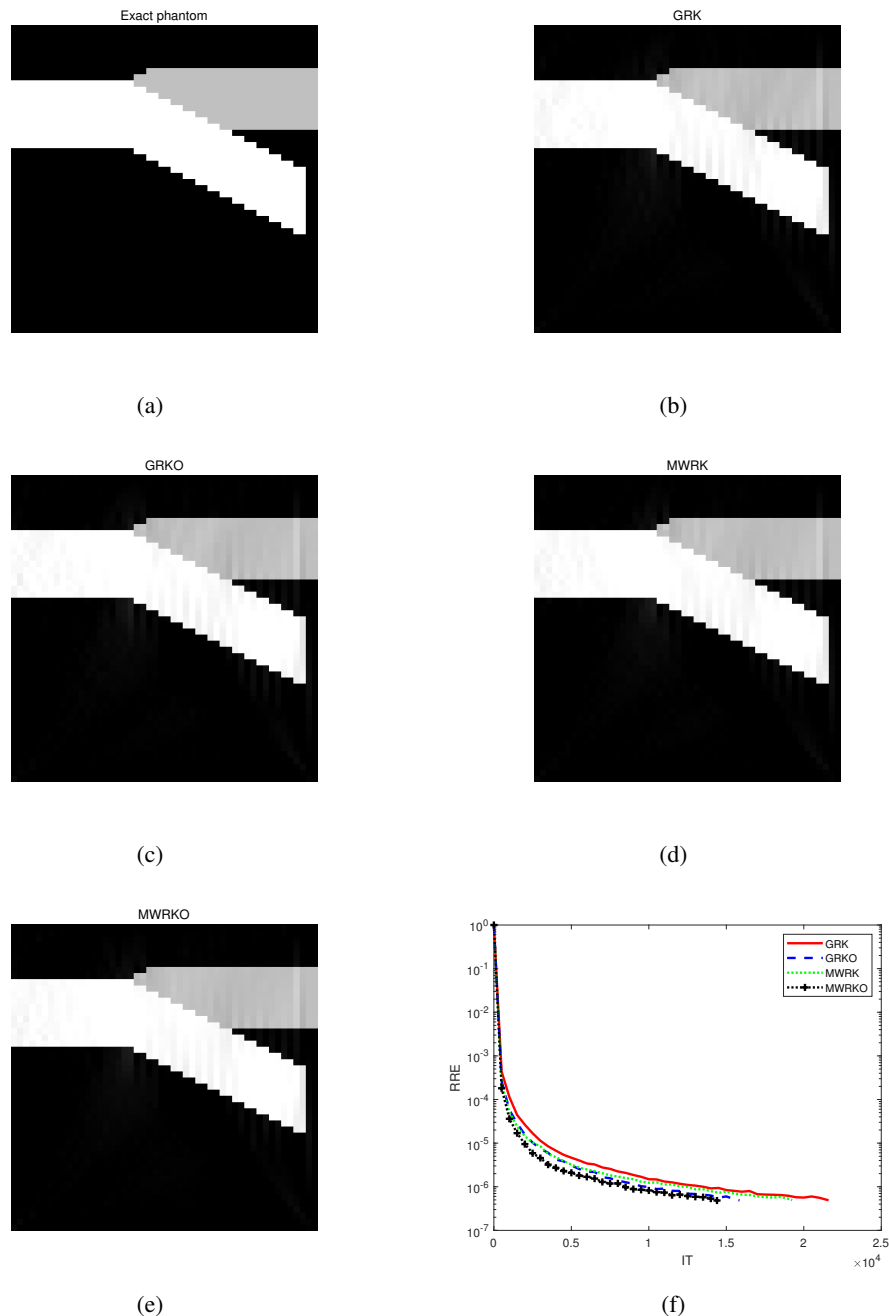
In this subsection, we compare the classical methods—Landweber method [59] and generalized minimum residual (GMRES) method [60] with our proposed methods—GRKO and MWRKO. The iterative expression of Landweber is:

$$x^{(k+1)} = x^{(k)} + \eta A^T (b - Ax^{(k)}),$$

where  $\eta$  is a real parameter satisfying  $0 < \eta < \frac{2}{\lambda_{\max}(A^T A)}$ . For convenience, we take  $\eta = \frac{2}{\|A\|_F^2}$  ( $< \frac{2}{\lambda_{\max}(A^T A)}$ ) in the following numerical experiments. There are also other new improvements to the Landweber method. Interested readers can refer [61, 62] and the references therein.

For the GMRES method, we use the matlab function  $gmres(A, b, n, RRE^*, n)$ , where  $RRE^* = \frac{\|b - Ax^{(k)}\|}{\|b\|} = \sqrt{RRE}$  and  $n$  is the dimension of matrix  $A$ . In this group of experiments, we take the parameter  $\omega = 0.5 \times 10^{-8}$  and sparse matrices  $A \in \mathbb{R}^{600 \times 600}$  are generated by the MATLAB function

$sprand(600, 600, density, 0.75)$  and then perform row unitization processing. Obviously, the biggest computational cost of the GRKO method and the MWRKO method is the update of the residual  $r$ . Therefore, we first calculate  $B = AA^T$  and then calculate the residual  $r$  according to Eq (2.2), which can improve CPU to a certain extent. Note that the calculation time of  $B$  is included in the CPU of the GRKO method and the MWRKO method.



**Figure 8.** Performance of the (a) exact phantom, (b) GRK, (c) GRKO, (d) MWRK, (e) MWRKO methods for *seismictomo* test problem. (f) Convergence history of four methods.



**Table 9.** IT and CPU of MWRKO, GRKO, Landweber and GMRES for sparse matrices  $A \in \mathbb{R}^{600 \times 600}$  under different densities.

density	IT				CPU			
	MWRKO	GRKO	Landweber	GMRES	MWRKO	GRKO	Landweber	GMRES
0.15	1423	1432	3177	600	0.1285	0.2557	0.4897	0.2522
0.3	1501	1518	3236	600	0.1808	0.3484	1.0245	0.2839
0.45	1546	1557	3221	600	0.2299	0.3708	3.3660	0.3239
0.6	1573	1593	3245	600	0.2633	0.4202	4.5986	0.3423
0.75	1581	1599	3236	600	0.2782	0.4282	5.7157	0.3671
0.9	1632	1652	3231	600	0.3132	0.4734	6.9041	0.4021

In Table 9, we compare IT and CPU of MWRKO, GRKO, Landweber and GMRES methods at different densities. Obviously as density increases, the CPU increases for all methods. Among them, the best CPU performance is still the MWRKO method, followed by the GMRES method.

## 5. Conclusions

Combined with the representative randomized and non-randomized row index selection strategies, two Kaczmarz-type methods with oblique projection for solving large-scale consistent linear systems are proposed, namely the GRKO method and the MWRKO method. The exponential convergences of the GRKO method and the MWRKO method are deduced. Theoretical and experimental results show that the convergence rates of the GRKO method and the MWRKO method are better than GRK method and MWRK method respectively. Numerical experiments show the effectiveness of these two methods, especially when the rows of the coefficient matrix  $A$  are highly linearly correlated.

## Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities [grant number 18CX02041A, 19CX05003A-2, 19CX02062A], the National Key Research and Development Program of China [grant number 2019YFC1408400] and the Science and Technology Support Plan for Youth Innovation of University in Shandong Province [No.YCX2021151].

## Conflict of interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

1. S. Kaczmarz, Approximate solution of systems of linear equations, *Int. J. Control*, **57** (1993), 1269–1271. <https://doi.org/10.1080/00207179308934446>
2. R. Gordon, R. Bender, G. Herman, Algebraic reconstruction techniques (ART) for three dimensional electron microscopy and X-ray photography, *J. Theor. Biol.*, **29** (1970), 471–481. [https://doi.org/10.1016/0022-5193\(70\)90109-8](https://doi.org/10.1016/0022-5193(70)90109-8)

3. A. J. Devaney, A filtered backpropagation algorithm for diffraction tomography, *Ultrason. Imaging*, **4** (1982), 336–350. [https://doi.org/10.1016/0161-7346\(82\)90017-7](https://doi.org/10.1016/0161-7346(82)90017-7)
4. S. J. Wernecke, L. R. D’Addario, Maximum entropy image reconstruction, *IEEE Trans. Comput.*, **26** (1997), 351–364. <https://doi.org/10.1109/TC.1977.1674845>
5. J. Skilling, R. K. Bryan, Maximum entropy image reconstruction-general algorithm, *Mon. Not. R. Astron. Soc.*, **211** (1984), 111–124. <https://doi.org/10.1109/TC.1977.1674845>
6. W. Guo, H. Chen, W. Geng, L. Li, A modified Kaczmarz algorithm for computerized tomographic image reconstruction, in *International Conference on Biomedical Engineering and Informatics IEEE*, (2019), 1–4. <https://doi.org/10.1109/BMEI.2009.5305654>
7. S. Lee, H. J. Kim, Noise properties of reconstructed images in a kilo-voltage on-board imaging system with iterative reconstruction techniques: A phantom study, *Phys. Med.*, **30** (2014), 365–373. <https://doi.org/10.1016/j.ejmp.2013.11.003>
8. D. Carmona-Ballester, J. M. Trujillo-Sevilla, Bonaque-González, Ó. Gómez-Cárdenes, J. M. Rodríguez-Ramos, Weighted nonnegative tensor factorization for atmospheric tomography reconstruction, *Astron. Astrophys.*, **614** (2018), A41. <https://doi.org/10.1051/0004-6361/201832597>
9. T. Li, D. Isaacson, J. C. Newell, G. J. Saulnier, Adaptive techniques in electrical impedance tomography reconstruction, *Physiol. Meas.*, **35** (2014), 1111–1124. <https://doi.org/10.1088/0967-3334/35/6/1111>
10. R. Ramlau, M. Rosensteiner, An efficient solution to the atmospheric turbulence tomography problem using Kaczmarz iteration, *Inverse Probl.*, **28** (2012), 095004. <https://doi.org/10.1088/0266-5611/28/9/095004>
11. G. Thoppe, V. S. Borkar, D. Manjunath, A stochastic Kaczmarz algorithm for network tomography, *Automatica*, **50** (2014), 910–914. <https://doi.org/10.1016/j.automatica.2013.12.016>
12. A. Hefny, D. Needell, A. Ramdas, Rows versus columns: randomized Kaczmarz or Gauss-Seidel for ridge regression, *SIAM J. Sci. Comput.*, **39** (2017), S528–S542. <https://doi.org/10.1137/16M1077891>
13. V. Borkar, N. Karamchandani, S. Mirani, Randomized Kaczmarz for rank aggregation from pairwise comparisons, in *2016 IEEE Information Theory Workshop (ITW)*, (2016), 389–393. <https://doi.org/10.1109/ITW.2016.7606862>
14. J. Loera, J. Haddock, D. Needell, A sampling Kaczmarz-Motzkin algorithm for linear feasibility, *SIAM J. Sci. Comput.*, **39** (2017), S66–S87. <https://doi.org/10.1137/16m1073807>
15. H. Q. Guan, R. Gordon, A projection access order for speedy convergence of ART (algebraic reconstruction technique): a multilevel scheme for computed tomography, *Phys. Med. Biol.*, **39** (1994), 2005–2022. <https://doi.org/10.1088/0031-9155/39/11/013>
16. X. Intes, V. Ntziachristos, J. P. Culver, A. Yodh, B. Chance, Projection access order in algebraic reconstruction technique for diffuse optical tomography, *Phys. Med. Biol.*, **47** (2002), N1–N10. <https://doi.org/10.1088/0031-9155/47/1/401>
17. X. L. Xu, J. S. Liow, S. C. Strother, Iterative algebraic reconstruction algorithms for emission computed tomography: A unified framework and its application to positron emission tomography, *Med. Phys.*, **20** (1993), 1675–1684. <https://doi.org/10.1118/1.596954>

18. K. Tanabe, Projection method for solving a singular system of linear equations and its applications, *Numer. Math.*, **17** (1971), 203–214. <https://doi.org/10.1007/BF01436376>
19. C. G. Kang, H. Zhou, The extension of convergence rates of kaczmarz type methods, *J. Comput. Appl. Math.*, **382** (2021), 113099. <http://doi.org/10.1016/j.cam.2020.113099>
20. C. G. Kang, Convergence rates of the Kaczmarz-Tanabe method for linear systems, *J. Comput. Appl. Math.*, **394** (2021), 113577. <https://doi.org/10.1016/j.cam.2021.113577>
21. S. F. McCormick, The methods of Kaczmarz and row orthogonalization for solving linear equations and least squares problems in Hilbert space, *Indiana Univ. Math. J.*, **26** (1977), 1137–1150. <http://www.jstor.org/stable/24891603>
22. K. Du, H. Gao, A new theoretical estimate for the convergence rate of the maximal residual Kaczmarz algorithm, *Numer. Math. Theor. Meth. Appl.*, **12** (2019), 627–639. <https://doi.org/10.4208/nmtma.OA-2018-0039>
23. T. Strohmer, R. Vershynin, A randomized Kaczmarz algorithm with exponential convergence, *J. Fourier Anal. Appl.*, **15** (2009), 262–278. <https://doi.org/10.1007/s00041-008-9030-4>
24. D. Needell, J. A. Tropp, Paved with good intentions: analysis of a randomized block Kaczmarz method, *Linear Algebra Appl.*, **441** (2014), 199–221. <https://doi.org/10.1016/j.laa.2012.12.022>
25. D. Needell, R. Zhao, A. Zouzias, Randomized block Kaczmarz method with projection for solving least squares, *Linear Algebra Appl.*, **484** (2015), 322–343. <https://doi.org/10.1016/j.laa.2015.06.027>
26. I. Necoara, Faster randomized block Kaczmarz algorithms, *SIAM J. Matrix Anal. Appl.*, **40** (2019), 1425–1452. <https://doi.org/10.1137/19M1251643>
27. K. Du, W. T. Si, X. H. Sun, Randomized extended average block Kaczmarz for solving least squares, *SIAM J. Sci. Comput.*, **42** (2020), A3541–A3559. <https://doi.org/10.1137/20M1312629>
28. 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. <https://doi.org/10.1137/17M1137747>
29. Z. Z. Bai, W. T. Wu, On relaxed greedy randomized Kaczmarz methods for solving large sparse linear systems, *Appl. Math. Lett.*, **83** (2018), 21–26. <https://doi.org/10.1016/j.aml.2018.03.008>
30. X. Yang, A geometric probability randomized Kaczmarz method for large scale linear systems, *Appl. Numer. Math.*, **164** (2021), 139–160. <https://doi.org/10.1016/j.apnum.2020.10.016>
31. J. J. Zhang, A new greedy Kaczmarz algorithm for the solution of very large linear systems, *Appl. Math. Lett.*, **91** (2019), 207–212. <https://doi.org/10.1016/j.aml.2018.12.022>
32. Y. Liu, C. Q. Gu, Variant of greedy randomized Kaczmarz for ridge regression, *Appl. Numer. Math.*, **143** (2019), 223–246. <https://doi.org/10.1016/j.apnum.2019.04.008>
33. Z. Z. Bai, W. T. Wu, On partially randomized extended Kaczmarz method for solving large sparse overdetermined inconsistent linear systems, *Linear Algebra Appl.*, **578** (2019), 225–250. <https://doi.org/10.1016/j.laa.2019.05.005>
34. 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. <https://doi.org/10.1137/15M1014425>

35. K. Du, Tight upper bounds for the convergence of the randomized extended Kaczmarz and Gauss-Seidel algorithms, *Numer. Linear Algebra Appl.*, **26** (2019), e2233. <https://doi.org/10.1002/nla.2233>
36. D. Needell, R. Ward, Two-subspace projection method for coherent overdetermined systems, *J. Fourier Anal. Appl.*, **19** (2013), 256–269. <https://doi.org/10.1007/s00041-012-9248-z>
37. J. Liu, S. J. Wright, An accelerated randomized Kaczmarz algorithm, *Math. Comput.*, **85** (2016), 153–178. <https://doi.org/10.1090/mcom/2971>
38. C. PoPa, T. Preclik, H. Köstler, U. Rűde, On Kaczmarz’s projection iteration as a direct solver for linear least squares problems, *Linear Algebra Appl.*, **436** (2012), 389–404. <https://doi.org/10.1016/j.laa.2011.02.017>
39. Y. J. Guan, W. G. Li, L. L. Xing, T. T. Qiao, A note on convergence rate of randomized Kaczmarz method, *Calcolo*, **57** (2020), 1–11. <https://doi.org/10.1007@s10092-020-00376-4>
40. Q. F. Wang, W. G. Li, W. D. Bao, X. Q. Gao, Nonlinear Kaczmarz algorithms and their convergence, *J. Comput. Appl. Math.*, **399** (2022), 113720. <https://doi.org/10.1016/j.cam.2021.113720>
41. F. Schöpfer, D. A. Lorenz, Linear convergence of the randomized sparse Kaczmarz method, *Math. Program.*, **1** (2016), 1–28. <https://doi.org/10.1007/s10107-017-1229-1>
42. D. Leventhal, A. Lewis, Randomized methods for linear constraints: convergence rates and conditioning, *Math. Oper. Res.*, **35** (2010), 641–654. <https://doi.org/10.1287/moor.1100.0456>
43. Z. Z. Bai, W. T. Wu, On greedy randomized coordinate descent methods for solving large linear least-squares problems, *Numer. Linear Algebra Appl.*, **26** (2019), 1–15. <https://doi.org/10.1002/nla.2237>
44. K. W. Chang, C. J. Hsieh, C. J. Lin, Coordinate descent method for large-scale l2-loss linear support vector machines, *J. Mach. Learn. Res.*, **9** (2008), 1369–1398. <https://doi.org/10.5555/1390681.1442778>
45. Z. Lu, L. Xiao, On the complexity analysis of randomized block-coordinate descent methods, *Math. Program.*, **152** (2015), 615–642. <https://doi.org/10.1007/s10107-014-0800-2>
46. I. Necoara, Y. Nesterov, F. Glineur, Random block coordinate descent methods for linearly constrained optimization over networks, *J. Optim. Theory Appl.*, **173** (2017), 227–254. <https://doi.org/10.1007/s10957-016-1058-z>
47. Y. Nesterov, S. Stich, Efficiency of the accelerated coordinate descent method on structured optimization problems, *SIAM J. Optim.*, **27** (2017), 110–123. <https://doi.org/10.1137/16M1060182>
48. P. Richtárik, M. Takáč, Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function, *Math. Program.*, **144** (2014), 1–38. <https://doi.org/10.1007/s10107-012-0614-z>
49. S. Wright, Coordinate descent algorithms, *Math. Program.*, **151** (2015), 3–34. <https://doi.org/10.1007/s10107-015-0892-3>
50. J. H. Zhang, J. H. Guo, On relaxed greedy randomized coordinate descent methods for solving large linear least-squares problems, *Appl. Numer. Math.*, **157** (2020), 372–384. <https://doi.org/10.1016/j.apnum.2020.06.014>

51. C. Popa, *Projection algorithms—classical results and developments*, Applications to image reconstruction, LAP Lambert Academic Publishing, Saarbrucken, 2012
52. D. A. Lorenz, S. Rose, Frank Schöpfer, The randomized Kaczmarz method with mismatched adjoint, *BIT*, **58** (2018), 1–20. <https://doi.org/10.1007/s10543-018-0717-x>
53. W. G. Li, Q. F. Wang, W. D. Bao, L. Liu, On Kaczmarz method with oblique projection for solving large overdetermined linear systems, preprint, arXiv:math/2106.13368
54. W. T. Wu, On two-subspace randomized extended Kaczmarz method for solving large linear least-squares problems, *Numer. Algor.*, **89** (2022), 1–31. <https://doi.org/10.1007/s11075-021-01104-x>
55. A. Ben-Israel, T. N. E. Thomas, *Generalized inverses: Theory and applications*, Springer Science & Business Media, 2003.
56. F. Wang, W. G. Li, W. D. Bao, Z. L. Lv, Gauss-Seidel method with oblique direction, *Results Appl. Math.*, **12** (2021), 100180. <https://doi.org/10.1016/j.rinam.2021.100180>
57. T. A. Davis, Y. Hu, The University of Florida sparse matrix collection, *ACM Trans. Math. Software*, **38** (2011), 1–25. <https://doi.org/10.1145/2049662.2049663>
58. P. C. Hansen, J. S. Jorgensen, AIR tools II: algebraic iterative reconstruction methods, improved implementation, *Numer. Algorithms*, **79** (2018), 107–137. <https://doi.org/10.1007/s11075-017-0430-x>
59. L. H. Landweber, An iteration formula for Fredholm integral equations of the first kind, *Am. J. Math.*, **73** (1951), 615–624. <https://doi.org/10.2307/2372313>
60. Y. Saad, M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.*, **7** (1986), 856–869. <https://doi.org/10.1137/0907058>
61. L. Liang, Y. Xu, Adaptive landweber method to deblur images, *IEEE Signal Process. Lett.*, **10** (2003), 129–132. <https://doi.org/10.1109/LSP.2003.810012>
62. X. L. Zhao, T. Z. Huang, X. M. Gu, L. J. Deng, Vector extrapolation based Landweber method for discrete ill-posed problems, *Math. Prob. Eng.*, **2017** (2017), 1375716. <https://doi.org/10.1155/2017/1375716>



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)