

# Two Greedy Subspace Kaczmarz Algorithm for Image Reconstruction

Yong Liu and Chuanqing Gu<sup>†</sup>

**Abstract**—We propose a two greedy subspace Kaczmarz algorithm for solving consistent systems of linear equations. At each step, our proposed algorithm projects the current iterate onto the solution space given by two greedily selected rows. It is proved that this algorithm converges to the unique solution of the consistent linear system. Numerical results on Gaussian models as well as on image reconstruction problems show that the proposed algorithm improves the convergence rate of the greedy randomized Kaczmarz algorithm.

**Index Terms**—greedy randomized Kaczmarz algorithm, two greedy subspace Kaczmarz algorithm, consistent linear systems, image reconstruction.

## I. INTRODUCTION

MANY applications in science [1], [2], [3] and engineering [4], [5] require the solution of very large consistent systems of linear equations of the form

$$Ax = b, \quad (1)$$

where  $A \in \mathbb{R}^{m \times n}$  is a real  $m$ -by- $n$  matrix,  $b \in \mathbb{R}^m$  is an  $m$ -dimensional real vector and  $x \in \mathbb{R}^n$  denotes the  $n$ -dimensional vector of unknowns. The Kaczmarz algorithm [6] is a classic while powerful iterative solver for computing an approximate solution for such equations (1), and it has been widely used in the field of image reconstruction [7], [8], [9], [10] as an algebraic reconstruction technique [11] due to its simplicity and light computation. More precisely, let  $A^{(i)}$  represent the  $i$ th row of the matrix  $A$ , and  $b^{(i)}$  the  $i$ th entry of the vector  $b$ , then given an initial point  $x_0$ , the Kaczmarz algorithm can be formulated as

$$x_{k+1} = x_k + \frac{b^{(i_k)} - A^{(i_k)}x_k}{\|A^{(i_k)}\|_2^2} (A^{(i_k)})^T, \quad k = 0, 1, \dots,$$

where  $(\cdot)^T$  denotes the transpose of the corresponding vector or matrix,  $\|\cdot\|_2$  denotes the Euclidean norm and  $i_k = (k \bmod m) + 1$ . Although the Kaczmarz algorithm is popular in practice, useful theoretical estimates of the convergence rate of this algorithm are difficult to obtain [12], [13] and the convergence of the algorithm is sometimes very slow (see e.g., [14] and [15]). To improve the convergence of the Kaczmarz algorithm, in 2009 Strohmer and Vershynin [16] proposed the randomized Kaczmarz (RK) algorithm by choosing the row index  $i_k$  from the set  $\{1, 2, \dots, m\}$

randomly with probability proportional to  $\|A^{(i_k)}\|_2^2$  rather than in their given order. Strohmer and Vershynin proved that the RK algorithm converges with expected exponential rate, known as “linear convergence”, for more details on convergence theory for the RK algorithm, we refer to [17], [18], [19] and the references therein. In 2018, Bai and Wu [20] proposed the greedy randomized Kaczmarz (GRK) algorithm by introducing a different but more effective probability criterion, which can grasp larger entries of the residual vector at each iteration; see also [21]. It is precisely because of the introduction of this greedy probability criterion that the GRK algorithm converges faster than the RK algorithm. The GRK algorithm is a valuable development of the RK algorithm, and was extended to ridge regression problem [22], generalized to so-called relaxed versions [23] and accelerated in block version [24]. The block version of the GRK algorithm can effectively reduce the executing time of the original algorithm, but it will require more computational cost per iteration than GRK, since it needs to solve a least squares problem at each iteration, which is the most expensive (arithmetic) step in block GRK algorithm.

In this paper, we present another accelerated iterative formula of the GRK algorithm, that is, two greedy subspace Kaczmarz (2GSK) algorithm, which, likes the GRK algorithm, requires the calculation of the residual vector corresponding to the linear system (1) at each iteration step. The difference between the two is that the 2GSK algorithm selects two distinct row indices  $s_k$  and  $t_k$  that corresponding to the first two distinct entries in the ordered residual vector (i.e., we order the absolute values of the entries of the residual vector  $r_k = b - Ax_k$  from largest to smallest) at each iteration  $k$ . In this way, we hope to make it simple to implement and computationally inexpensive when compared with the block version of the GRK algorithm. It should be noted that, in 2013 Needell and Ward [25] also proposed a two-subspace randomized Kaczmarz algorithm for coherent overdetermined systems, which randomly selects two different rows at each iteration step of the RK algorithm to iterate, so it is different from our algorithm.

Our paper is organized as follows. In Section II we propose the two greedy subspace Kaczmarz (2GSK) algorithm for linear system (1) and present its convergence analysis. In Section III, we report and discuss the numerical results. Finally, in Section IV we end the paper with brief conclusions.

## II. 2GSK ALGORITHM

In this paper,  $\lambda_{\min}(\cdot)$  is used to denote the smallest non-zero eigenvalue of the corresponding matrix. We denote the identity matrix by  $I$ , with the subscript denoting the dimension when needed. In what follows, we always assume

Manuscript received July 05, 2020; revised August 31, 2020. This work was supported by National Natural Science Foundation (11371243) and Key Program of Natural Science of Changzhou College of Information Technology (CXKZ201908Z).

Yong Liu is with the Department of Mathematics, Changzhou College of Information Technology, Changzhou, Jiangsu, 213164 China: (e-mail: liuyong@czcit.edu.cn).

Chuanqing Gu (Corresponding author) is with the Department of Mathematics, Shanghai University, Shanghai, 200444 China (e-mail: cqu@shu.edu.cn).

$A^{(i)} \neq 0$  for all  $i \in \{1, 2, \dots, m\}$ . Let  $\ell$  be a prescribed positive integer and  $x_0$  be a given initial point, then the 2GSK algorithm can be algorithmically described as follows.

**Algorithm 1** The 2GSK Algorithm

**Require:**  $A, b, \ell$  and  $x_0$ .

**Ensure:**  $x_\ell$ .

- 1: **for**  $k = 0, 1, 2, \dots, \ell - 1$  **do**
- 2:   Compute  $r_k = b - Ax_k$
- 3:   Select rows  $s_k$  and  $t_k$  that satisfy

$$s_k = \arg \min_{1 \leq i \leq m} |r_k^{(i)}| \text{ and } t_k = \arg \min_{i \in [m] \setminus s_k} |r_k^{(i)}|$$

- 4:   Set  $x_{k+1} = x_k + \frac{r_k^{(s_k)} \cdot (A^{(s_k)})^T}{\|A^{(s_k)}\|_2^2} + \frac{r_k^{(t_k)} \cdot (A^{(t_k)})^T}{\|A^{(t_k)}\|_2^2}$
- 5: **end for**

For the convergence property of the 2GSK algorithm, we can establish the following theorem.

**Theorem 1.** Consider the consistent linear system  $Ax = b$ , where  $A \in \mathbb{R}^{m \times n}$ ,  $A^{(1)}, A^{(2)}, \dots, A^{(m)}$  are mutually non-orthogonal and  $b \in \mathbb{R}^m$  is a given vector. Then the iteration sequence  $\{x_k\}_{k=0}^\infty$  generated by the 2GSK algorithm starting from any initial guess  $x_0 \in \mathbb{R}^n$  in the column space of  $A^T$ , converges to the unique solution  $x_* = A^\dagger b$  of  $Ax = b$  with  $A^\dagger$  the Moore-Penrose pseudoinverse of  $A$ . Moreover, the solution error for the iteration sequence  $\{x_k\}_{k=0}^\infty$  obeys

$$\|x_{k+1} - x_*\|_2^2 \leq \prod_{q=0}^k (1 - \lambda_{\min}(P_q P_q^T)) \|x_0 - x_*\|_2^2,$$

where  $P_k = P_{s_k} - P_{t_k}$  with

$$P_{s_k} = \frac{(A^{(s_k)})^T}{\|A^{(s_k)}\|_2} \cdot \frac{A^{(s_k)}}{\|A^{(s_k)}\|_2} \text{ and } P_{t_k} = \frac{(A^{(t_k)})^T}{\|A^{(t_k)}\|_2} \cdot \frac{A^{(t_k)}}{\|A^{(t_k)}\|_2}.$$

Here  $s_k$  and  $t_k$  are two distinct rows of the coefficient matrix  $A$  that selected at the  $k$ th iterate.

**Proof:** It is easy to verify that  $P_{s_k}$  and  $P_{t_k}$  defined in (2) are orthogonal projection matrices. Let  $e_k = x_k - x_*$ , due to the consistency of the linear system  $Ax = b$ , we have

$$\begin{aligned} \|e_{k+1}\|_2^2 &= \|(I - P_{s_k} - P_{t_k})e_k\|_2^2 \\ &= \|e_k\|_2^2 - \langle e_k, P_{s_k} e_k \rangle - \langle e_k, P_{t_k} e_k \rangle \\ &\quad + \langle P_{s_k} e_k, P_{t_k} e_k \rangle + \langle P_{t_k} e_k, P_{s_k} e_k \rangle \\ &= \|e_k\|_2^2 - \langle P_{s_k} e_k, e_k - P_{t_k} e_k \rangle - \langle P_{t_k} e_k, e_k - P_{s_k} e_k \rangle \\ &= \|e_k\|_2^2 - e_k^T P_{s_k} (I - P_{t_k}) e_k - e_k^T P_{t_k} (I - P_{s_k}) e_k \\ &= \|e_k\|_2^2 - e_k^T (P_{s_k} - P_{s_k} P_{t_k} + P_{t_k} - P_{t_k} P_{s_k}) e_k \\ &= \|e_k\|_2^2 - e_k^T (P_{s_k} - P_{t_k}) (P_{s_k} - P_{t_k}) e_k. \end{aligned} \tag{3}$$

Denote  $P_k = P_{s_k} - P_{t_k}$  and  $H_k = P_k P_k^T$ , then the matrix  $H_k$  is symmetric and positive semidefinite. Let  $H_k = Q_k \Lambda_k Q_k^T$  be the spectral decomposition of the matrix  $H_k$ , where  $Q_k \in \mathbb{R}^{n \times n}$  is an orthogonal matrix, and  $\Lambda_k = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \in \mathbb{R}^{n \times n}$  is a diagonal matrix, with its diagonal entries  $\{\lambda_i\}_{i=1}^n$ , or the eigenvalues of the matrix  $H_k$ , being ordered such that

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > \lambda_{r+1} = \dots = \lambda_n = 0,$$

where  $r$  is the rank of matrix  $H_k$ . Then (3) can be rewritten as

$$\|e_{k+1}\|_2^2 = \|e_k\|_2^2 - e_k^T H_k e_k = \|e_k\|_2^2 - (Q_k^T e_k)^T \Lambda_k Q_k^T e_k.$$

Assume that

$$Q_k^T e_k = (q_1, q_2, \dots, q_n)^T,$$

then it holds that

$$\begin{aligned} \|e_{k+1}\|_2^2 &= \|e_k\|_2^2 - (Q_k^T e_k)^T \Lambda_k Q_k^T e_k \\ &= \|e_k\|_2^2 - \sum_{i=1}^n \lambda_i q_i^2 \\ &\leq \|e_k\|_2^2 - \lambda_r \sum_{i=1}^n q_i^2 \\ &= \|e_k\|_2^2 - \lambda_r \|Q_k^T e_k\|_2^2 \\ &= (1 - \lambda_{\min}(H_k)) \|e_k\|_2^2. \end{aligned} \tag{4}$$

On the other hand, by noticing that if  $\lambda$  is an eigenvalue of  $P_k$ , then  $\lambda \in [-1, 1]$  (see, e.g., [26]), so  $\lambda_i \in [0, 1]$  for all  $i \in \{1, 2, \dots, n\}$  and  $\lambda_{\min}(H_k) = \lambda_r \in (0, 1]$ . We further claim that  $\lambda_{\min}(H_k) = 1$  if and only if the coefficient matrix  $A$  is orthogonal. More specifically, without loss of generality, we assume that  $\lambda = 1$ , let  $\alpha$  be an eigenvector corresponding to  $\lambda$ , then

$$(P_{s_k} - P_{t_k})\alpha = \alpha.$$

Using the fact  $P_{s_k} P_{s_k} = P_{s_k}$  and multiplying the above equation by  $P_{s_k}$  we can obtain  $P_{s_k} \alpha - P_{s_k} P_{t_k} \alpha = P_{s_k} \alpha$ , then  $P_{s_k} P_{t_k} \alpha = 0$ , i.e.,

$$\frac{(A^{(s_k)})^T}{\|A^{(s_k)}\|_2^2} \cdot \frac{A^{(s_k)}}{\|A^{(s_k)}\|_2} (A^{(t_k)})^T \cdot A^{(t_k)} \alpha = 0$$

which further implies that

$$A^{(s_k)} (A^{(t_k)})^T = 0 \text{ or } A^{(t_k)} \alpha = 0.$$

If  $A^{(t_k)} \alpha = 0$  and  $A^{(s_k)} (A^{(t_k)})^T \neq 0$ , then from the following relationship

$$P_{t_k} (P_{s_k} - P_{t_k}) \alpha = P_{t_k} \alpha,$$

we know that

$$P_{t_k} P_{s_k} \alpha = 2 P_{t_k} \alpha = 2 \frac{(A^{(t_k)})^T}{\|A^{(t_k)}\|_2^2} A^{(t_k)} \alpha = 0,$$

i.e.,

$$\frac{(A^{(t_k)})^T}{\|A^{(t_k)}\|_2^2} \cdot \frac{A^{(t_k)}}{\|A^{(s_k)}\|_2^2} \cdot A^{(s_k)} \alpha = 0,$$

since  $A^{(s_k)} (A^{(t_k)})^T \neq 0$ , there holds  $A^{(s_k)} \alpha = 0$  and

$$(P_{s_k} - P_{t_k}) \alpha = \frac{(A^{(s_k)})^T}{\|A^{(s_k)}\|_2^2} A^{(s_k)} \alpha - \frac{(A^{(t_k)})^T}{\|A^{(t_k)}\|_2^2} A^{(t_k)} \alpha = 0 \neq \alpha,$$

which leads to a contradiction.

Hence, it holds that  $A^{(s_k)} (A^{(t_k)})^T = 0$  for different  $s_k, t_k \in \{1, 2, \dots, m\}$ . If  $A$  is an orthogonal matrix, then we have  $P_{s_k} P_{t_k} = 0$ , i.e.,  $P_{s_k} - P_{t_k}$  is also an orthogonal projection matrix, which indicates that all eigenvalues of  $P_{s_k} - P_{t_k}$  are 0 or 1. Thus we have  $\lambda_{\min}(H_k) = 1$ .

Consequently

$$1 - \lambda_{\min}(H_k) < 1$$

is valid for all rows which are mutually non-orthogonal. Then, iterating the inequality (4) recursively yields

$$\|x_{k+1} - x_\star\|_2^2 \leq \prod_{q=0}^k (1 - \lambda_{\min}(P_q P_q^T)) \|x_0 - x_\star\|_2^2. \quad \square$$

From Theorem 1, we know that the upper bound for the convergence rate of the 2GSK algorithm can be represented as

$$\varrho_{2GSK} = \left( \prod_{q=0}^{k-1} (1 - \lambda_{\min}(P_q P_q^T)) \right)^{1/k}, \quad (5)$$

where  $k$  is the total number of iteration steps when a certain stopping rule is satisfied.

**Remark 1.** If all rows of the coefficient matrix  $A$  of the linear system (1) are mutually orthogonal, then, according to [23, Theorem 1], it is easy to show that

$$\begin{aligned} \|e_{k+1}\|_2^2 &= \|e_k\|_2^2 - \|P_{s_k} e_k\|_2^2 - \|P_{t_k} e_k\|_2^2 \\ &\leq \left( 1 - 2 \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2} \right) \|e_k\|_2^2. \end{aligned}$$

As a result, the upper bound of the convergence rate of the 2GSK is smaller than that of the GRK algorithm, which can be represented as [20]

$$\varrho_{GRK} = 1 - \frac{1}{2} \left( \frac{\|A\|_F^2}{\|A\|_F^2 - \min_{1 \leq i \leq m} \|A^{(i)}\|_2^2} + 1 \right) \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2}. \quad (6)$$

Hence, the former may converge much faster than the later.

### III. NUMERICAL EXPERIMENTS

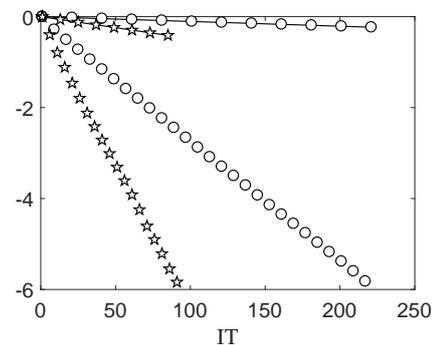
In this section, we are going to use the 2GSK and GRK algorithms to solve linear system (1) with the coefficient matrix  $A \in \mathbb{R}^{m \times n}$  being either generated randomly by MATLAB function `randn(m, n)` with different  $m$  and  $n$ , or taken from the University of Florida sparse matrix collection [27], and with  $A$  being taken from the field of 2D tomography image reconstruction problems.

In our implementations, the starting point is always the origin point, i.e.,  $x_0 = 0$ . We show the number of iteration steps (denoted as ‘IT’) and the computing time (denoted as ‘CPU’) in seconds of the above two iterative algorithms for solving different linear systems. For GRK algorithm, we compute the averaged results of IT and CPU over 50 independent simulations to reduce the statistical oscillations in the results. All experiments are performed by using MATLAB (R2016b) on a personal computer with 2.67 GHz central processing unit (Intel(R) Core(TM) i5 CPU), 4.00 GB memory, and Windows operating system (Windows 10).

**Example 1.** In this example, we compare the upper bound defined in (5) of the convergence rate of the 2GSK with that of the GRK defined in (6) for linear consistent system  $Ax = b$ . The matrix  $A$ , solution, and the right-hand side given by  $A = \text{randn}(1000, 100)$ ,  $x_\star = \text{randn}(100, 1)$  and  $b = Ax_\star$ , respectively. Iterations of GRK and 2GSK are terminated once the relative solution error (RSE) at the current iterate  $x_k$ , defined by

$$\text{RSE} = \frac{\|x_k - x_\star\|_2^2}{\|x_\star\|_2^2}$$

satisfies  $\text{RSE} < 10^{-6}$ . In Figure 1, we plot the curves of the RSE in base-10 logarithm, and the curves for the upper bounds  $\log_{10}(\varrho_{GRK}) \times \text{IT}$  and  $\log_{10}(\varrho_{2GSK}) \times \text{IT}$ , versus the iteration step.



**Fig. 1.** Picture of  $\log_{10}(\text{RSE})$  for GRK and 2GSK,  $\log_{10}(\varrho_{GRK}) \times \text{IT}$  and  $\log_{10}(\varrho_{2GSK}) \times \text{IT}$  versus IT when  $A = \text{randn}(1000, 100)$ .  $\log_{10}(\text{RSE})$  for GRK: “o o o”;  $\log_{10}(\text{RSE})$  for 2GSK: “☆☆☆”;  $\log_{10}(\varrho_{GRK}) \times \text{IT}$ : “o-o-”;  $\log_{10}(\varrho_{2GSK}) \times \text{IT}$ : “-☆☆”.

From Figure 1, we observe that the upper bound of the convergence rate of the 2GSK algorithm is smaller than that of the GRK algorithm, and the RSE of 2GSK is decaying more quickly than that of GRK when the iteration step is increasing. Hence, the 2GSK outperforms the GRK in terms of iteration counts.

**Example 2.** We use 2GSK and GRK algorithms to solve linear system (1) with the coefficient matrix  $A$  being dense and of different size. We use `randn(m, n)` to generate  $A$  with different  $m$  and  $n$  and  $x_\star = \text{randn}(n, 1)$ , and set  $b = Ax_\star$ . All computations are terminated once  $\text{RSE} < 10^{-6}$ .

We list the numbers of iteration steps and the computing times for both GRK and 2GSK algorithms in Tables I-II, and in these two tables we also report the speed-up of 2GSK against GRK, which is defined as

$$\text{speed-up} = \frac{\text{CPU of GRK}}{\text{CPU of 2GSK}}.$$

TABLE I: IT and CPU of GRK and 2GSK for  $m$ -by- $n$  matrices  $A$  with  $m = 5000$  and different  $n$ .

$m \times n$		5000 × 100	5000 × 300	5000 × 500
GRK	IT	164.9	529.9	998.3
	CPU	0.1497	1.0422	2.5137
2GSK	IT	63.0	219.0	434.0
	CPU	0.0715	0.6426	1.7937
speed-up		2.09	1.62	1.40

These numerical results in Tables I-II show that 2GSK significantly outperforms GRK in terms of both iteration counts and CPU times, whether the linear system (1) is overdetermined or underdetermined. More specifically, the speed-up is at least 1.40 and at most 2.09 for overdetermined linear system, and it is at least 1.22 and at most 1.79 for underdetermined linear system. The per iteration performance of GRK and 2GSK on this example are shown in Figures

TABLE II: IT and CPU of GRK and 2GSK for  $m$ -by- $n$  matrices  $A$  with  $n = 5000$  and different  $m$ .

$m \times n$		$100 \times 5000$	$300 \times 5000$	$500 \times 5000$
GRK	IT	220.8	823.1	1541.3
	CPU	0.1713	1.4184	4.5000
2GSK	IT	109.0	409.0	772.0
	CPU	0.0958	1.0843	3.7000
speed-up		1.79	1.31	1.22

2-7, in which we depict the curves of the RSE in base-10 logarithm versus the iteration step. From this figure, we see again that the RSE of 2GSK is decaying more quickly than that of GRK when the iteration step is increasing. In conclusion, our proposed algorithm is feasible and converges faster than GRK algorithm for Gaussian models.

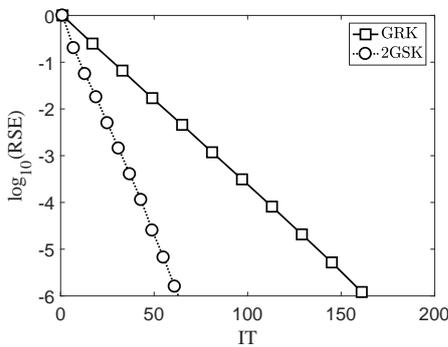


Fig. 2.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK when  $A = \text{randn}(5000, 100)$ .

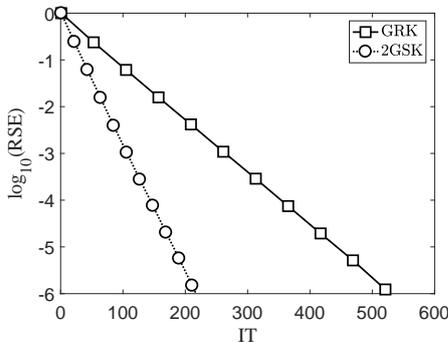


Fig. 3.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK when  $A = \text{randn}(5000, 300)$ .

**Example 3.** In this example, we use 2GSK and GRK algorithms to compute an approximate solution to linear system (1) with the coefficient matrix  $A \in \mathbb{R}^{m \times n}$  taken from the University of Florida sparse matrix collection [27], such as full-rank sparse matrices *ash958*, *Trefethen\_300*, *abtaha1*, *nemsafm*, *refine* and *bibd\_16\_8*. We set solution vector  $x_* = \text{randn}(n, 1)$ , and the right-hand side  $b = Ax_*$ . The termination criterion for this example is exactly the same as in Example 2. In Tables III-IV we report iteration counts and CPU times for 2GSK and GRK algorithms.

From Tables III-IV, we see that the iteration counts and CPU times of 2GSK are considerably smaller than those of

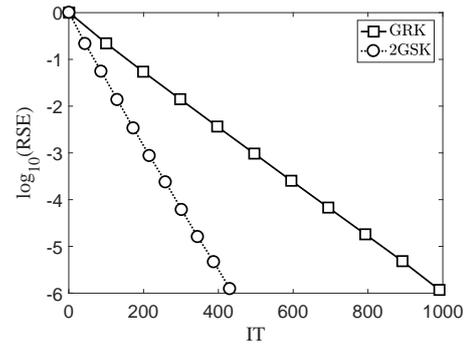


Fig. 4.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK when  $A = \text{randn}(5000, 500)$ .

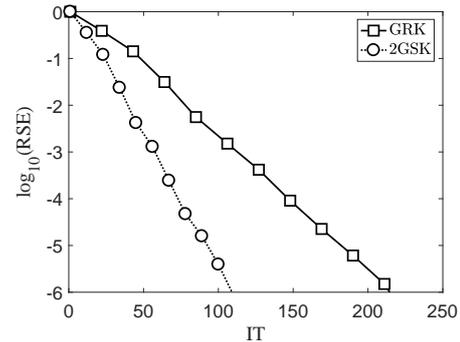


Fig. 5.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK when  $A = \text{randn}(100, 5000)$ .

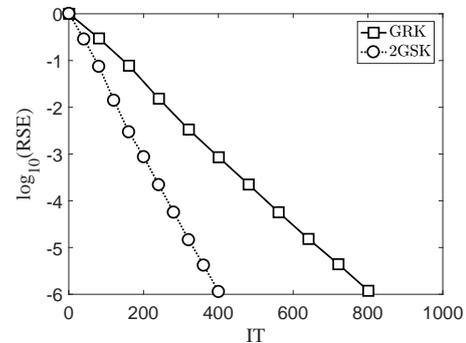


Fig. 6.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK when  $A = \text{randn}(300, 5000)$ .

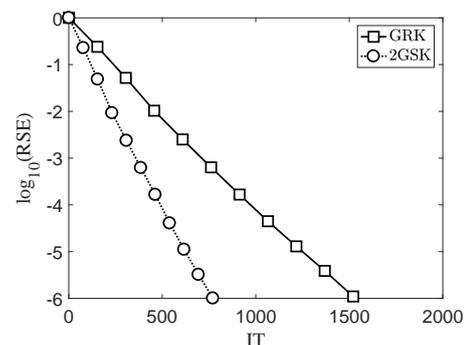


Fig. 7.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK when  $A = \text{randn}(500, 5000)$ .

TABLE III: IT and CPU of GRK and 2GSK for  $m$ -by- $n$  matrices  $A$  with different  $m$  and  $n$ .

name	ash958	Trefethen_300	abtaha1
$m \times n$	$958 \times 292$	$300 \times 300$	$14596 \times 209$
density	0.68%	5.20%	1.68%
cond(A)	3.20	1772.69	12.23
GRK	IT	790.0	3220.9
	CPU	0.3086	0.7227
2GSK	IT	373.0	1549.0
	CPU	0.1038	0.1806
speed-up	2.97	4.00	1.67

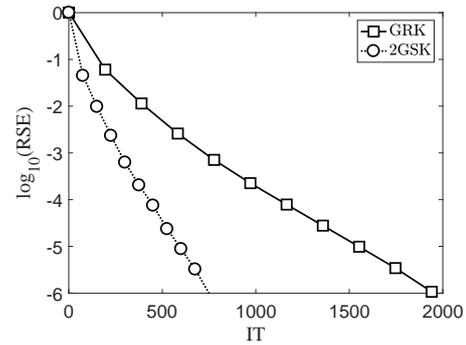


Fig. 10.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK with respect to the matrix abtaha1.

TABLE IV: IT and CPU of GRK and 2GSK for  $m$ -by- $n$  matrices  $A$  with different  $m$  and  $n$ .

name	nemsafm	refine	bibd_16_8
$m \times n$	$334 \times 2348$	$29 \times 62$	$120 \times 12870$
density	0.36%	8.51%	23.33%
cond(A)	4.77	66.67	9.54
GRK	IT	1199.3	669.7
	CPU	1.7000	0.0896
2GSK	IT	572.0	374.0
	CPU	0.9452	0.0116
speed-up	1.79	7.72	2.20

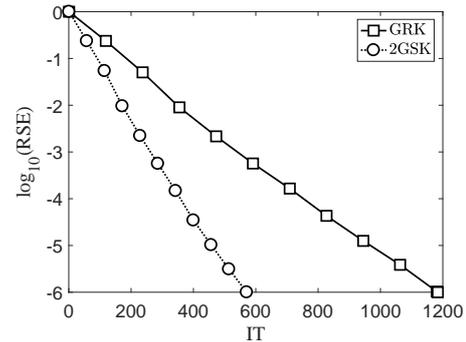


Fig. 11.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK with respect to the matrix nemsafm.

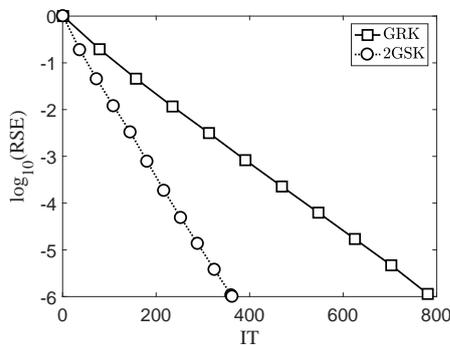


Fig. 8.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK with respect to the matrix ash958.

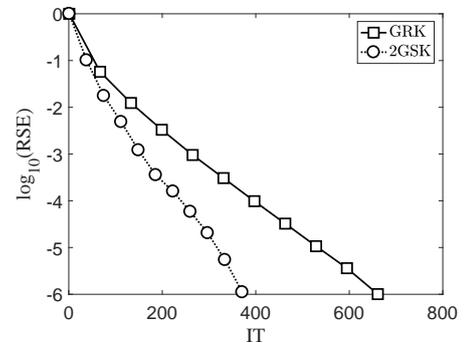


Fig. 12.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK with respect to the matrix refine.

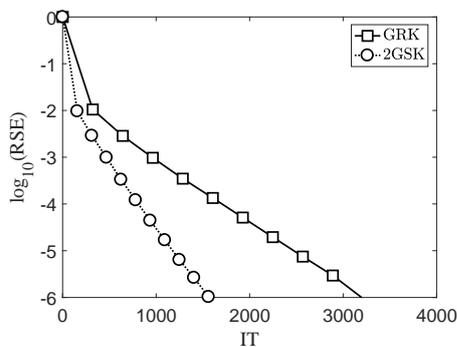


Fig. 9.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK with respect to the matrix Trefethen\_300.

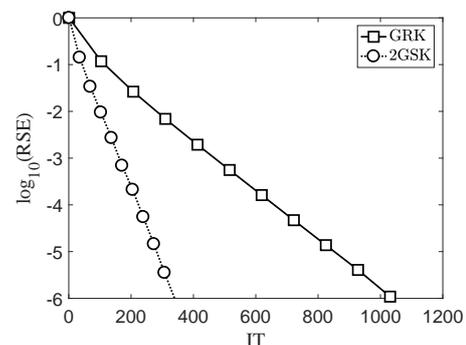
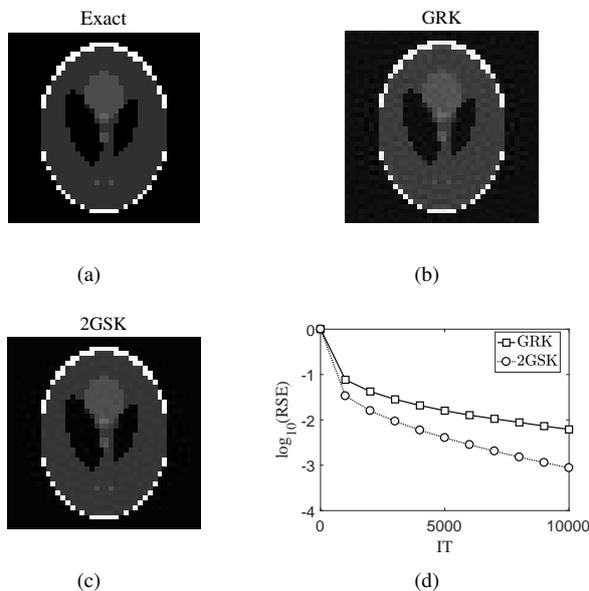


Fig. 13.  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK with respect to the matrix bibd\_16\_8.

GRK, and the speed-up is at least 1.67 (the matrix abtaha1) and the biggest even reaches 7.72 (the matrix refine). The above observations are intuitively shown in Figures 8-13, in which we plot the curves of the RSE in base-10 logarithm versus the iteration step for different matrices. This figure also demonstrates that the 2GSK algorithm offers the most improvement over the GRK algorithm in terms of iteration step. Hence, the 2GSK algorithm significantly outperforms the GRK algorithm for both iteration counts and CPU times.

**Example 4.** In this example, we use a test problem from tomographic image reconstruction, implemented in the function `paralleltomo(N,theta,p)` in the MATLAB package AIR TOOLS [10], which generates a sparse matrix  $A$ , an exact solution  $x_*$  (which is shown as image in Figure 14(a)) and right-hand side  $b = Ax_*$ . We set  $N = 40$ ,  $\theta = 0 : 2 : 178^\circ$  and  $p = 120$ , then the resulting matrix  $A$  is of size  $10800 \times 1600$ , and run 2GSK and GRK algorithms for  $10^4$  iterations on such linear consistent system  $Ax = b$ . Reconstruction results and per iteration performance for both 2GSK and GRK algorithms are shown in Figure 14.

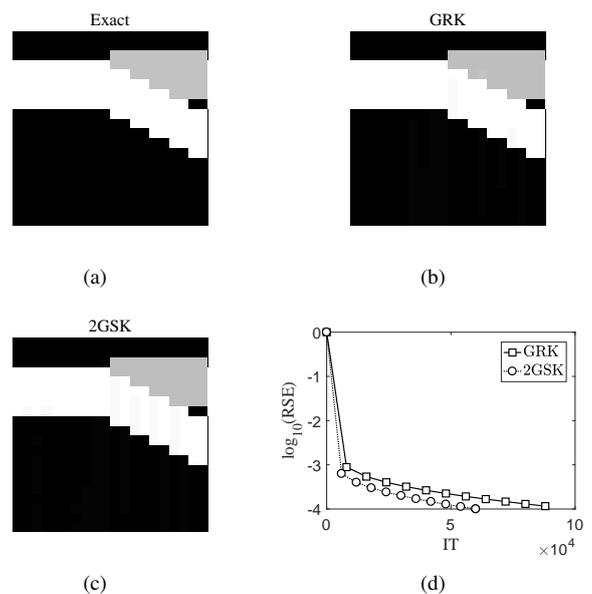


**Fig. 14.** Paralleltomo test problem: exact image  $x_*$  (a), restorations by GRK (b) and 2GSK (c). (d)  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK for this restoration.

Admittedly, we see from Figure 14(b)-14(c) that under the same number of iteration steps, 2GSK algorithm can achieve better reconstruction results than GRK algorithm. Moreover, from Figure 14(d), it is easy to see that 2GSK algorithm is obviously faster than the GRK algorithm.

**Example 5.** 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 [10], which also generates a sparse matrix  $A$ , an exact solution  $x_*$  (which is shown as image in Figure 15(a)) and right-hand side  $b = Ax_*$ . In this example, we set  $N = 20$ ,  $s = 60$  and  $p = 100$ , then the resulting matrix  $A$  is of size  $6000 \times 400$ . We run 2GSK and GRK on such system  $Ax = b$  until the RSE is below  $10^{-4}$ . Reconstruction results and per iteration performance for both 2GSK and

GRK algorithms are shown in Figure 15.



**Fig. 15.** Seismictomo test problem: exact image  $x_*$  (a), restorations by GRK (b) and 2GSK (c). (d)  $\log_{10}(\text{RSE})$  versus IT for 2GSK and GRK for this restoration.

We should point out that for this test problem, GRK algorithm and 2GSK algorithm require 80.5547 seconds and, respectively 46.9917 seconds when they reach the same stopping criterion  $\text{RSE} < 10^{-4}$ . In addition, from Figure 15(d), we know that 2GSK algorithm even requires much smaller iteration step than GRK algorithm in achieving the same reconstructions. Hence, the 2GSK algorithm significantly outperforms the GRK algorithm in terms of both iteration counts and CPU times, too.

#### IV. CONCLUSION

In this paper, we extended the GRK algorithm to the two greedy subspace Kaczmarz (2GSK) algorithm to solve a large consistent linear system of equations. We provide theoretical guarantees for the convergence of 2GSK algorithm. In the experiments section, we have shown some cases when 2GSK algorithm work better than GRK algorithm in terms of both iteration steps and computing times. Hence, when compared with the GRK algorithm, the 2GSK algorithm can be a useful tool for solving large-scale consistent linear systems. One of the interesting future directions of the current work would be to investigate three or multiple greedy subspace Kaczmarz algorithm rather than block versions of the GRK algorithm (since applying the pseudoinverse  $A_\tau^\dagger$  to a vector in block versions is the most expensive arithmetic) and provide theoretical estimates for it.

#### REFERENCES

- [1] A. Patrascu and I. Necoara, Nonasymptotic convergence of stochastic proximal point algorithms for constrained convex optimization, *Journal of Machine Learning Research*, vol. 18, pp. 1-42, 2018.
- [2] J. J. Cui, G. H. Peng, Q. Lu and Z. G. Huang, "A class of nonstationary upper and lower triangular splitting

- iteration methods for ill-posed inverse problems,” *IAENG International Journal of Computer Science*, vol. 47, no. 1, pp. 118-129, 2020.
- [3] Y. Wang, J. C. Wang, F. J. Li, Y. C. Zhou, M. Xiong and M. Li, “Face image super-resolution reconstruction via mapping matrix and multilayer model,” *IAENG International Journal of Computer Science*, vol. 47, no. 1, pp. 68-74, 2020.
- [4] K. S. Sim, F. F. Ting, J. W. Leong and C. P. Tso, “Signal-to-noise ratio estimation for SEM single image using cubic spline interpolation with linear least square regression,” *Engineering Letters*, vol. 27, no. 1, pp. 151-165, 2019.
- [5] M. X. Chen, B. X. Liu, D. S. Zeng and W. Gao, “A framework for ontology-driven similarity measuring using vector learning tricks,” *Engineering Letters*, vol. 27, no. 3, pp. 549-558, 2019.
- [6] S. Kaczmarz, “Angenäherte Auflösung von Systemen linearer Gleichungen,” in *Bulletin International de l’Académie Polonaise des Sciences et des Lettres. Classes des Sciences Mathématiques et Naturelles. Series A: Sciences Mathématiques 1937*, pp. 355-357.
- [7] Y. Censor, “Row-action methods for huge and sparse systems and their applications,” *SIAM Review*, vol. 23, no. 4, pp. 444-466, 1981.
- [8] P. C. Hansen and M. Saxild-Hansen, “AIR tools—a MATLAB package of algebraic iterative reconstruction methods,” *Journal of Computational and Applied Mathematics*, vol. 236, no. 8, pp. 2167-2178, 2012.
- [9] J. M. Blackledge and A. I. Al-Rawi, “Steganography using stochastic diffusion for the covert communication of digital images,” *IAENG International Journal of Applied Mathematics*, vol. 41, no. 4, pp. 270-298, 2011.
- [10] P. C. Hansen and J. S. Jørgensen, “AIR Tools II: algebraic iterative reconstruction methods improved implementation,” *Numerical Algorithms*, vol. 79, no. 1, pp. 107-137, 2018.
- [11] C. Onunka, G. Bright and R. Stopforth, “The forward inverse problem in robot control signal selection: EEG artefact selection using discrimination kernels with kaczmarz and algebraic reconstruction technique,” *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2017, 25-27 October, 2017, San Francisco, USA*, pp. 696-701.
- [12] F. Deutsch and H. Hundal, “The rate of convergence for the method of alternating projections,” *Journal of Mathematical Analysis and Applications*, vol. 205, no. 2, pp. 381-405, 1997.
- [13] A. Galántai, “On the rate of convergence of the alternating projection method in finite dimensional spaces,” *Journal of Mathematical Analysis and Applications*, vol. 310, no. 1, pp. 30-44, 2005.
- [14] Z. Z. Bai and X. G. Liu, “On the Meany inequality with applications to convergence analysis of several row-action iteration methods,” *Numerische Mathematik*, vol. 124, no. 2, pp. 215-236, 2013.
- [15] Z. Z. Bai and W. T. Wu, “On convergence rate of the randomized Kaczmarz method,” *Linear Algebra and its Applications*, vol. 553, pp. 252-269, 2018.
- [16] T. Strohmer and R. Vershynin, “A randomized Kaczmarz algorithm with exponential convergence,” *Journal of Fourier Analysis and Applications*, vol. 15, no. 2, pp. 262-278, 2009.
- [17] A. Ma, D. Needell and A. Ramdas, “Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods,” *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 4, pp. 1590-1604, 2015.
- [18] D. Needell, “Randomized Kaczmarz solver for noisy linear systems,” *BIT Numerical Mathematics*, vol. 50, no. 2, pp. 395-403, 2010.
- [19] A. Agaskar, C. Wang and Y. M. Lu, “Randomized Kaczmarz algorithms: exact MSE analysis and optimal sampling probabilities,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP), Atlanta, GA, 2014*, pp. 389-393.
- [20] Z. Z. Bai and W. T. Wu, “On greedy randomized Kaczmarz method for solving large sparse linear systems,” *SIAM Journal on Scientific Computing*, vol. 40, no. 1, pp. A592-A606, 2018.
- [21] Z. Z. Bai and W. T. Wu, “On relaxed greedy randomized Kaczmarz methods for solving large sparse linear systems,” *Applied Mathematics Letters*, vol. 83, pp. 21-26, 2018.
- [22] Y. Liu and C. Q. Gu, “Variant of greedy randomized Kaczmarz for ridge regression,” *Applied Numerical Mathematics*, vol. 143, pp. 223-246, 2019.
- [23] J. J. Zhang, “A new greedy Kaczmarz algorithm for the solution of very large linear systems,” *Applied Mathematics Letters*, vol. 91, pp. 207-212, 2019.
- [24] Y. Q. Niu and B. Zheng, “A greedy block Kaczmarz algorithm for solving large-scale linear systems,” *Applied Mathematics Letters*, 104 (2020) 106294.
- [25] D. Needell and R. Ward, “Two-subspace projection method for coherent overdetermined systems,” *Journal of Fourier Analysis and Applications*, vol. 19, no. 2, pp. 256-269, 2013.
- [26] A. Galántai, *Projectors and Projection Methods*, Kluwer Academic Publishers, Norwell, MA, 2004.
- [27] T. A. Davis and Y. Hu, “The University of Florida sparse matrix collection,” *Acm Transactions on Mathematical Software*, vol. 38, no. 1, pp. 1-25, 2011.