On Greedy Randomized Kaczmarz Algorithm for the Solution of Tikhonov Regularization Problem

Yong Liu, Zhiyong Zhang

Abstract—**Tikhonov regularization technique is widely recognized as one of the most prevalent and well-established approaches for solving linear discrete ill-posed problems. The present study introduces two novel randomized iterative algorithms for the computation of numerical solutions to large-scale Tikhonov regularization problems. The first one applies the randomized Kaczmarz algorithm to an augmented regularized normal system of equations, the second one is an accelerated version of the first one by means of greedy probability criterion. In theory, we establish some convergence results for these two algorithms. Numerical experiments demonstrate the convergence properties and illustrate the performances of these two algorithms.**

Index Terms—**Tikhonov regularization, ill-posed problems, randomized Kaczmarz algorithm, greedy probability criterion, augmented regularized system, convergence property.**

I. Introduction

REGULARIZATION is the fundamental method for solving highly ill-conditioned linear least-squares problems of the following form

$$\min_{\mathrm{x}\in\mathbb{R}^{\mathrm{n}}} \|Ax-b\|_2, \tag{1}$$

where $A\in\mathbb{R}^{m\times n}$ is severely ill-conditioned matrix, $x\in\mathbb{R}^n$ is an unknown of dimension n, $\|\cdot\|_2$ represents the Euclidean norm of a vector or matrix. The vector $b\in\mathbb{R}^m$ is often contaminated by Gaussian white noise. This type of issue frequently emerges due to the discretization process applied to linear ill-posed problems, which can be found in many practical applications, such as statistical analysis [1, 2], machine learning [3], image reconstruction [4–12] and computerized tomograph [13–15]. To acquire a practical and stable approximate solution for (1), one can solve the Tikhonov regularization problem stated below

$$\min_{\mathrm{x}\in\mathbb{R}^{\mathrm{n}}} \left\{\|Ax-b\|_2^2+\alpha\|x\|_2^2\right\}. \tag{2}$$

Here $\alpha>0$ is a regularization parameter, which is used to balance the size of $\|Ax-b\|_2$ and $\|x\|_2$. The normal system associated with (2) is given by

$$(A^TA+\alpha I_n)x=A^Tb, \tag{3}$$

in which the matrix I_n is an $n\times n$ identity matrix. As the coefficient matrix of the linear system mentioned above is nonsingular, it can be inferred that equation (3) possesses a unique solution

$$x_\alpha=(A^TA+\alpha I_n)^{-1}A^Tb.$$

In general, finding an appropriate parameter α for a specific problem (3) is very challenging. At present, there are mainly two different kinds of methods for calculating α, which depend on an estimation of the size of the noise in b, e.g., discrepancy principle [7], and on the right-hand side b, e.g., L-curve criterion [16] and generalized cross validation [17]. This is not the focus of this paper, and we are interested in solving (2) by some randomized Kaczmarz-like iterative algorithms, such as VRK and VRGS algorithms [18]. For some prior research on the calculation of an approximate solution to equation (2), please refer to [19–22] and the references therein.

The Tikhonov regularization problem (2) is mathematically equivalent to solve the following augmented regularized linear system

$$\tilde{A}\tilde{x}=\tilde{b}, \tag{4}$$

in which

$$\tilde{A}=\begin{pmatrix}\sqrt{\alpha}I_m & A\\ A^T & -\sqrt{\alpha}I_n\end{pmatrix},\ \tilde{x}=\begin{pmatrix}u\\ x\end{pmatrix}\text{ and }\tilde{b}=\begin{pmatrix}b\\ 0\end{pmatrix}.$$

For this problem, Ivanov and Zhdanov [23] applied the Kaczmarz algorithm [24] to (4), and developed a more compact and cost-effective iteration through leveraging the special structure of the coefficient matrix $\tilde{A}$. The Kaczmarz algorithm produces the subsequent iteration $\tilde{x}_{k+1}$ from $\tilde{x}_0\in\mathbb{R}^{m+n}$ by utilizing

$$\tilde{x}_{k+1}=\tilde{x}_k+\frac{\tilde{b}^{(i_k)}-\tilde{A}^{(i_k)}\tilde{x}_k}{\|\tilde{A}^{(i_k)}\|_2^2}(\tilde{A}^{(i_k)})^T,$$

where $\tilde{A}^{(i_k)}$ is the i_kth row of the matrix $\tilde{A}$, $\tilde{b}^{(i_k)}$ is the i_kth entry of the vector $\tilde{b}$, the superscript ^{T} denotes the transpose of a vector or a matrix and the target row is numbered as $i_k=1,2,\cdots,m+n,1,2,\cdots$. In their numerical experiments, Ivanov and Zhdanov utilized the randomized Kaczmarz (RK) method [25], where the row i_k adheres to a probability distribution that is directly proportional to $\|A^{(i_k)}\|_2^2$, in view of this, we will refer to this algorithm as RKT (T stands for Tikhonov) below. The fact that $\tilde{A}$ is nonsingular ensures that the Kaczmarz algorithm can find the unique solution of the linear system (4), see also [24].

It is worth mentioning that the randomized extended Kaczmarz (REK) algorithm [26], proposed by Zouzias and Freris, deserves special mention as an exceptionally effective iterative approach for solving inconsistent linear least-square problems such as (1). Zouzias and Freris initially employed the RK algorithm to solve the linear system $A^Tz=0$, yielding a vector z_k that converges towards $b_{\mathcal{R}(A)^\perp}$, which

Manuscript received January 4, 2024; revised May 29, 2024.
This work was supported by the National Natural Science Foundation (61773003) and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (23KJB110001).
Yong Liu is a lecturer of the School of Mathematics and Statistics, Changshu Institute of Technology, Suzhou, Jiangsu 215500 China (Corresponding author, e-mail: liuyong@cslg.edu.cn).
Zhiyong Zhang is a professor of the School of Mathematics and Statistics, Changshu Institute of Technology, Suzhou, Jiangsu 215500 China (e-mail: gothel@gmail.com).

represents the projection of b onto the space that is orthogonal to $\mathcal{R}(A)$, where $\mathcal{R}(A)$ denotes the range space of coefficient matrix A. Subsequently, they applied the RK algorithm once again to solve the linear system $Ax = b - z_k$. The advantage of the REK algorithm lies in its ability to reduce the impact of the noise on the sequence $\{x_k\}_{k=1}^{\infty}$, thereby ensuring that $b - z_k$ belongs to $\mathcal{R}(A)$. Recently, Bai and Wu utilized the greedy randomized Kaczmarz algorithm [27, 28] to solve an augmented system derived from the REK procedure, denoted as

$$\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix},$$

and constructed a more stable and faster randomized iterative algorithm (GRAK) [29] than REK algorithm for inconsistent linear systems $Ax = b$.

In this paper, we are going to derive another more stable and efficient implementation for RKT and its accelerated version for the linear system (4) by using the greedy probability criterion introduced in [27] and establish the associated properties of convergence. Extensive numerical results are provided to validate the theoretical findings and demonstrate the performance of our algorithms.

This paper is structured as follows: Section II introduces the novel algorithms and provides their theoretical analysis. Section III presents the numerical findings acquired. Lastly, Section IV encapsulates the paper with succinct conclusions and remarks.

II. Kaczmarz-like Algorithms for Tikhonov Regularization Problem

Throughout this paper, we use $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ to respectively represent the maximum and minimum values of the non-zero eigenvalues of matrix A. For $A \in \mathbb{R}^{m\times n}$, we use $\|A\|_F$, $A^\dagger$, A^T, $A^{(i)}$, $A_{(j)}$ and $\mathcal{R}(A)$ to represent its Frobenius norm, Moore-Penrose pseudoinverse, transpose, ith row, jth column and range space, respectively. Similarly, we represent the Euclidean norm of any vector $u \in \mathbb{R}^n$ as $\|u\|_2$ and its ith entry as $b^{(i)}$. The identity matrix is denoted by I, with the dimension indicated by a subscript when necessary. Additionally, we use e_j to refer to the column vector representing the jth coordinate basis. Furthermore, we define $\mathbb{E}_k$ as the expected value given the first k iterations, denoted as $\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot|\ j_0, j_1, \cdots, j_{k-1}]$, where $j_t(t = 0, 1, \cdots, k-1)$ represents the tth column selected during the tth iteration. Consequently, by applying the law of iterated expectations, $\mathbb{E}[\cdot]$ can be expressed as $\mathbb{E}[\mathbb{E}_k[\cdot]]$.

In the RKT algorithm, it can be readily confirmed that when setting the initial vector $\tilde{x}_0 = (u_0^T, x_0^T)^T$ satisfying $A^T u_0 = \sqrt{\alpha} x_0$, there holds that $A^T u_k = \sqrt{\alpha} x_k$ at each iteration step k. Thus, iterating the RKT is equivalent to using the RK algorithm with $x_0 = \frac{1}{\sqrt{\alpha}} A^T u_0$ and $u_0 \in \mathbb{R}^m$ to solve

$$\bar{A}\tilde{x} = b$$

with

$$\bar{A} = \begin{pmatrix} \sqrt{\alpha} I_m & A \end{pmatrix} \text{ and } \tilde{x} = (u^T, x^T)^T. \tag{5}$$

We abbreviate this algorithm as RKT_r algorithm, which can be found in Algorithm 1. The theorem presented below serves as evidence for the convergence property of the RKT_r algorithm.

Algorithm 1 The RKT_r Algorithm

Require: α, A, ℓ, b, u_0 and $x_0 = \frac{1}{\sqrt{\alpha}} A^T u_0$.
Ensure: x_ℓ.

for $k = 0, 1, 2, \cdots, \ell - 1$ **do**

  Select $i_k \in \{1, 2, \cdots, m\}$ by

$$P(i_k) = \frac{\|A^{(i_k)}\|_2^2 + \alpha}{\|A\|_F^2 + m\alpha}$$

  Set

$$u_{k+1} = u_k + \frac{b^{(i_k)} - \sqrt{\alpha} u_k^{(i_k)} - A^{(i_k)} x_k}{\|A^{(i_k)}\|_2^2 + \alpha} \sqrt{\alpha} e_{i_k}$$

  and

$$x_{k+1} = x_k + \frac{b^{(i_k)} - \sqrt{\alpha} u_k^{(i_k)} - A^{(i_k)} x_k}{\|A^{(i_k)}\|_2^2 + \alpha} (A^{(i_k)})^T$$

end for

Theorem 1. *For a given positive regularization parameter α, initiated from any vector $\tilde{x}_0 = (u_0^T, x_0^T)^T$ with $u_0 \in \mathbb{R}^m$ and $x_0 = \frac{1}{\sqrt{\alpha}} A^T u_0$, the approximate sequence $\{\tilde{x}_k\}_{k=1}^{\infty}$ with $\tilde{x}_k$ being defined as $\tilde{x}_k = (u_k^T, x_k^T)^T$, generated by the RKT_$r$ algorithm, converges towards the solution $\tilde{x}_* = (u_*^T, x_*^T)^T$ of the augmented regularized linear system (4) in expectation. In addition, the expected error of the solution associated to iterations $\{x_k\}_{k=0}^{\infty}$ and $\{u_k\}_{k=0}^{\infty}$ follows*

$$\begin{aligned} &\mathbb{E}\left(\|x_k - x_*\|_2^2 + \|u_k - u_*\|_2^2\right) \leq \\ &\quad \begin{cases} v_1^k \left(\|x_0 - x_*\|_2^2 + \|u_0 - u_*\|_2^2\right), & \text{if } m > n, \\ v_2^k \left(\|x_0 - x_*\|_2^2 + \|u_0 - u_*\|_2^2\right), & \text{if } m < n, \end{cases} \end{aligned} \tag{6}$$

where

$$\begin{aligned} &v_1 = 1 - \frac{\alpha}{\|A\|_F^2 + m\alpha}, \ v_2 = 1 - \frac{\sigma_{\min}^2(A) + \alpha}{\|A\|_F^2 + m\alpha}, \\ &x_* = \sqrt{\alpha}(AA^T + \alpha I_m)^{-1} b, \ u_* = A^T (AA^T + \alpha I_m)^{-1} b. \end{aligned}$$

Proof: For any $u_0 \in \mathbb{R}^m$, it exists a vector $v_0 = \frac{1}{\sqrt{\alpha}} u_0 \in \mathbb{R}^m$ such that

$$\tilde{x}_0 = \begin{pmatrix} u_0 \\ x_0 \end{pmatrix} = \begin{pmatrix} u_0 \\ \frac{1}{\sqrt{\alpha}} A^T u_0 \end{pmatrix} = \bar{A}^T v_0,$$

i.e., $\tilde{x}_0 \in \mathcal{R}(\bar{A}^T)$, with $\bar{A}$ being defined in (5). Therefore, based on Theorem 2 in [25], using the notations $\tilde{x}_* = (u_*^T, x_*^T)^T$ and $\tilde{x}_k = (u_k^T, x_k^T)^T$, we can conclude

$$\mathbb{E}\|\tilde{x}_k - \tilde{x}_*\|_2^2 \leq \left(1 - \frac{\sigma_{\min}^2(\bar{A})}{\|A\|_F^2 + m\alpha}\right) \|\tilde{x}_0 - \tilde{x}_*\|_2^2. \tag{7}$$

For the quantity $\sigma_{\min}^2(\bar{A})$, we have

$$\begin{aligned}\sigma_{\min}^2(\bar{A}) &= \sigma_{\min}(\bar{A}^T\bar{A})\\ &= \sigma_{\min}(AA^T+\alpha I_m)\\ &= \sqrt{\lambda_{\min}\left((AA^T+\alpha I_m)(AA^T+\alpha I_m)^T\right)}\\ &= \sqrt{\lambda_{\min}^2(AA^T+\alpha I_m)}\\ &= \lambda_{\min}(AA^T+\alpha I_m)\\ &= \lambda_{\min}(AA^T)+\alpha\\ &= \begin{cases}\alpha, & \text{if } m>n;\\ \sigma_{\min}^2(A)+\alpha, & \text{if } m<n,\end{cases}\end{aligned} \tag{8}$$

where $\lambda_{\min}(M)$ represents the smallest eigenvalue of general real square matrices M. Substituting (8) into (7), we straightforwardly obtain the estimate (6). This completes the proof.

The authors Bai and Wu [27] proposed a more efficient probabilistic criterion in 2018 for solving linear systems of equations. Their strategy involves prioritizing the selection of more substantial components within the residual vector during each iteration of the RK algorithm, leading to the creation of a greedy randomized Kaczmarz (GRK) algorithm. Numerical experiments conducted in [27] demonstrated that the GRK algorithm superior to the RK algorithm in terms of both iteration counts and computing time. Therefore, we intend to introduce the innovative criterion directly into the RKT_r algorithm to construct its accelerated version, i.e., greedy RKT_r (GRKT_r for short) algorithm as follows.

The following theorem establishes the convergence property of the GRKT_r algorithm.

Theorem 2. *For given regularization parameter $\alpha > 0$, starting from any initial guess $\tilde{x}_0 = (u_0^T, x_0^T)^T$ with $u_0 \in \mathbb{R}^m$ and $x_0 = \frac{1}{\sqrt{\alpha}}A^T u_0$, the approximations $\{\tilde{x}_k\}_{k=1}^{\infty}$ with $\tilde{x}_k$ being defined as $\tilde{x}_k = (u_k^T, x_k^T)^T$, obtained through the GRKT_r algorithm, converges to the unique solution $\tilde{x}_* = (u_*^T, x_*^T)^T$ of the augmented regularized linear system (4) in expectation. Moreover, the solution error associated to sequences $\{x_k\}_{k=0}^{\infty}$ and $\{u_k\}_{k=0}^{\infty}$ follows*

$$\mathbb{E}\left(\|x_1-x_*\|_2^2+\|u_1-u_*\|_2^2\right) \le (1-\varrho)\left(\|x_0-x_*\|_2^2+\|u_0-u_*\|_2^2\right)$$

and

$$\mathbb{E}\left(\|x_k-x_*\|_2^2+\|u_k-u_*\|_2^2\right) \le \left(1-\frac{\varrho(\|A\|_F^2+m\alpha+\gamma)}{2\gamma}\right)^{k-1} \cdot (1-\varrho)\left(\|x_0-x_*\|_2^2+\|u_0-u_*\|_2^2\right),$$

where

$$\gamma = \max_{1\le i\le m}\sum_{j=1,j\ne i}^{m}(\|A^{(j)}\|_2^2+\alpha),\ \varrho = \frac{\sigma_{\min}(AA^T+\alpha I_m)}{\|A\|_F^2+m\alpha}$$

with

$$\sigma_{\min}(AA^T+\alpha I_m) = \begin{cases}\alpha, & \text{if } m>n;\\ \sigma_{\min}^2(A)+\alpha, & \text{if } m<n.\end{cases}$$

Proof: Using the notations given in Theorem 1 and following the lines of the proof of Theorem 3.2 shown in [27], we can easily prove Theorem 2.

Algorithm 2 The GRKT_r Algorithm

Require: α, A, ℓ, b, u_0 and $x_0 = \frac{1}{\sqrt{\alpha}}A^T u_0$.
Ensure: x_ℓ.

1: **for** $k = 0, 1, 2, \cdots, \ell-1$ **do**
2: Compute
$$r_k = b - Ax_k - \sqrt{\alpha}u_k$$
and
$$\epsilon_k = \frac{1}{2}\left(\frac{1}{\|r_k\|_2^2}\max_{1\le i_k\le m}\frac{|r_k^{(i_k)}|^2}{\|A^{(i_k)}\|_2^2+\alpha}+\frac{1}{\|A\|_F^2+m\alpha}\right)$$
3: Compute the positive integers index set
$$\mathcal{U}_k = \left\{i_k \middle| \ |r_k^{(i_k)}|^2 \ge \epsilon_k\|r_k\|_2^2(\|A^{(i_k)}\|_2^2+\alpha)\right\}$$
4: Set
$$r_k^{(i)} = \begin{cases} b^{(i)} - A^{(i)}x_k - \sqrt{\alpha}u_k^{(i)}, & \text{if } i \in \mathcal{U}_k,\\ 0, & \text{otherwise}\end{cases}$$
5: Select $i_k \in \mathcal{U}_k$ by $P(i_k) = \frac{|r_k^{(i_k)}|^2}{\|r_k\|_2^2}$
6: Set
$$u_{k+1} = u_k + \frac{b^{(i_k)} - \sqrt{\alpha}u_k^{(i_k)} - A^{(i_k)}x_k}{\|A^{(i_k)}\|_2^2+\alpha}\sqrt{\alpha}e_{i_k}$$
and
$$x_{k+1} = x_k + \frac{b^{(i_k)} - \sqrt{\alpha}u_k^{(i_k)} - A^{(i_k)}x_k}{\|A^{(i_k)}\|_2^2+\alpha}(A^{(i_k)})^T$$
7: **end for**

III. Numerical Experiments

In this section, we apply the VRK, VRGS, REK, GRAK, RKT_r and GRKT_r algorithms to address a linear discrete ill-posed problem (1). The matrix $A \in \mathbb{R}^{m\times n}$ is created by utilizing MATLAB function `randn(m,n)` with varying values of m and n and taken from some practical applications, such as one space-dimension image restoration problem from the Regularization Tools MATLAB package [6] and 2D diffusion problem from the Iterative Regularization Tools MATLAB package [30]. All experiments are conducted on a personal computer running MATLAB (R2016b) with an Intel(R) Core(TM) i5 CPU operating at 2.67 GHz, equipped with 4.00 GB of memory and the Windows 11 operating system.

The numerical behaviors of these algorithms are assessed based on the quantity of iterations performed (referred to as 'IT') and the duration of computation measured in seconds (referred to as 'CPU'). In this context, CPU and IT represent the average values obtained from 20 repeated runs of the aforementioned algorithms. The initial guess x_0 in all the above algorithms is set to be 0. All experiments terminated as soon as

$$\text{RSE} = \frac{\|x_k - x_\star\|_2}{\|x_\star\|_2} \le \varepsilon,$$

where ε is a user-supplied constant much less than 1 and $x_\star$ is the desired solution (explained in each of the following specific examples). Due to the presence of noise in the right-hand side b, the algorithms presented in this paper will appear semi-convergence phenomenon [8], setting a very small ε, e.g., 10^{-6}, may cause these algorithms to iterate all the time, and a larger ε may cause each algorithm to be unable to show its own advantages, so we set different ε according to the different scale of the problems. We should point out that for the VRK algorithm, the iterative solution $x_k = A^T\eta_k$ will be calculated at each iteration step k in order to verify whether the RSE satisfies the termination condition, not just at the maximum iterative step ℓ. In addition, the noise e in the right-hand side b is Gaussian white noise and satisfies $\|e\|_2 = \epsilon\|\bar{b}\|_2$, where ϵ is the pre-assigned noise level. The regularization parameter α in each example is determined by the discrepancy principle, which can be implemented by the function `discrep` in the Regularization Tools MATLAB package [6].

Example 1. *We use the MATLAB function* `randn(m,n)` *to produce some matrices A with varying values for m and n for (1). We randomly generate a solution vector x_*=*`randn(n,1)`*, the noise-free data vector in b is taken to be $\bar{b} = Ax_*$ and $b = \bar{b} + e$ with noise level $\epsilon = 0.01$. The unique solution of (1) is set to be $x_\star = A^\dagger\bar{b}$. In addition, we set the stopping tolerance $\varepsilon = 10^{-2}$, or IT exceeds 100,000 when $m > n$, and exceeds 10,000 when $m < n$.*

TABLE I: Numerical results for 5000-by-n random matrix different n.

Algorithm	Index	n 300	500	700	900
RKT_r	IT	3450.0	6991.0	8427.0	15997.0
	CPU	0.0468	0.1989	0.3873	0.9763
VRK	IT	3584.0	6447.0	9841.5	16367.5
	CPU	8.0878	21.0757	47.8659	128.6405
VRGS	IT	2495.5	4774.0	7526.0	10211.0
	CPU	4.1953	13.2624	26.9406	49.1330
REK	IT	3601.0	6622.0	10237.0	14484.0
	CPU	2.1674	4.4303	7.3553	11.0297
GRAK	IT	1335.0	2663.5	4300.5	6432.0
	CPU	4.5028	12.8754	29.6369	61.6561
GRKT_r	IT	453.0	783.0	1265.5	1924.5
	CPU	1.0934	2.3456	5.1162	10.7797

TABLE II: Numerical results for m-by-1000 random matrix with different m.

Algorithm	Index	m 100	200	300	400
RKT_r	IT	166.5	416.5	771.5	1536.5
	CPU	0.0041	0.0117	0.0282	0.0644
VRK	IT	434.5	1012.5	1792.0	3086.0
	CPU	0.1002	0.3201	0.7806	1.6397
VRGS	IT	10000.0	10000.0	10000.0	10000.0
	CPU	2.8608	3.6582	4.2591	5.2086
REK	IT	690.5	1682.0	3169.0	5726.5
	CPU	0.2369	0.5260	1.0844	1.8480
GRAK	IT	259.5	662.5	1390.5	2489.0
	CPU	0.1482	0.4000	1.1256	2.5139
GRKT_r	IT	94.0	228.0	431.0	680.5
	CPU	0.0269	0.0843	0.1948	0.4017

In Tables I-II, we report IT and CPU for all algorithms. As the results in Tables I-II show, we see that for all tested matrices A, RKT_r and VRK have almost the same iteration steps. However, RKT_r outperforms VRK significantly in terms of CPU. Hence, compared with VRK, RKT_r is the winner regarding IT and CPU. In addition, when $m > n$, GRKT_r outperforms RKT_r, REK and GRAK in terms of IT, but, like GRAK, it is not competitive with RKT_r and REK regarding CPU, especially for $n = 900$, the CPU time of GRKT_r and GRAK is at least ten and sixty times that of RKT_r, respectively. Generally speaking, RKT_r exhibits superior CPU performance compared to the REK and GRAK, and GRKT_r algorithm performs best in terms of iteration steps when $m > n$. When $m < n$, both RKT_r and GRKT_r algorithms significantly outperform REK and GRAK algorithms regarding IT and CPU.

Example 2. *This test problem is taken from a one space-dimension image deblurring test problem. It is generated by the function* `blur(n)` *within the MATLAB package Regularization Tools [6]. This specific function produces a sparse matrix $A \in \mathbb{R}^{n^2\times n^2}$, a solution $x_\star \in \mathbb{R}^{n^2}$ (which is depicted as visual representations in Figure 1 (top left)) and the vector $\bar{b} = Ax_\star$. The vector $b = \bar{b} + e$ with noise level $\epsilon = 0.01$. We set $n = 64$ and the stopping tolerance $\varepsilon = 10^{-1}$, or IT exceeds 100,000.*

We list the CPU, IT and RSE for these algorithms in Table III. Additionally, the corresponding restorations are displayed in Figure 1. From this table we see that RKT_r and GRKT_r are seen to require the minimum number of IT and least amount CPU in achieving better relative errors, with GRKT_r requiring much smaller IT and less CPU than, but giving a same restoration of high quality as, RKT_r. On the other hand, we also find from Table III and Figure 1 that VRK is still a competitive algorithm when compared with those algorithms that do not converge within 100,000 steps in terms of the relative restoration error, even if it requires more CPU times than the other algorithms. The main reason for this prediction is that the RSE corresponding to those algorithms that do not converge within 100,000 steps, are decrease more and more slowly as the iteration step increases, which makes it difficult to reach the stopping tolerance for these algorithms, as demonstrated in Figure 2 where we present the log10(RSE) curves with respect to both IT and CPU.

TABLE III: Numerical results for blur image deblurring problem.

Algorithm	Index IT	CPU($\times10^3$)	RSE
RKT_r	33315.5	0.0108	0.1000
VRK	47009.0	4.6363	0.1000
VRGS	100000.0	3.1404	0.1694
REK	100000.0	0.1049	0.3666
GRAK	100000.0	0.1147	0.1137
GRKT_r	4342.0	0.0033	0.1000

Example 3. *We turn to the following 2D diffusion problem*

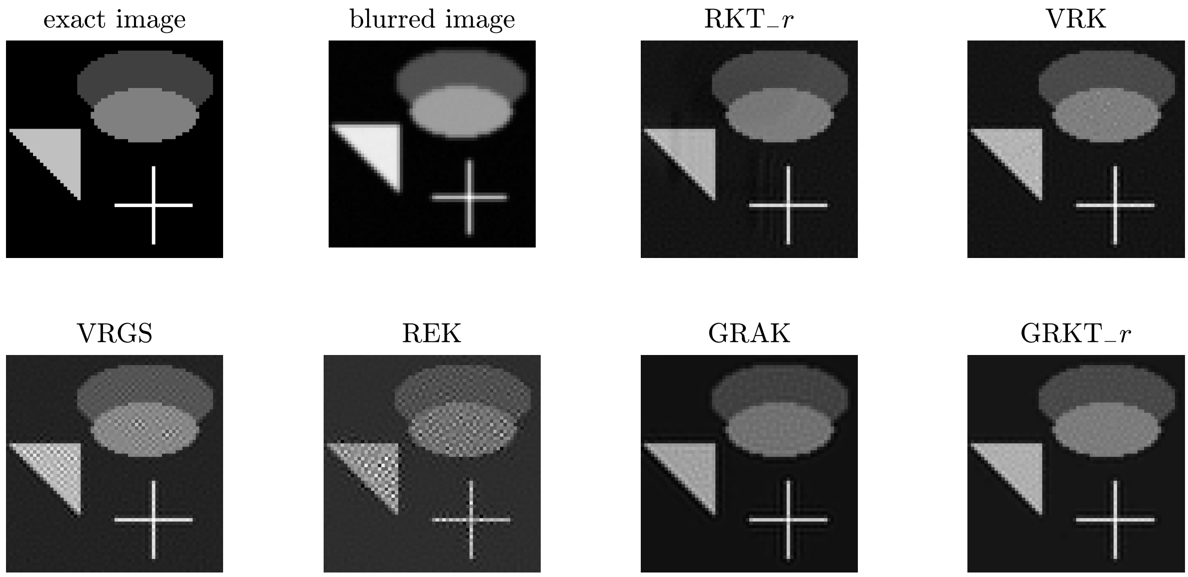

Fig. 1. Exact image and restorations corresponding to RKT_r, VRK, VRGS, REK, GRAK and GRKT_r algorithms for blur image deblurring problem.

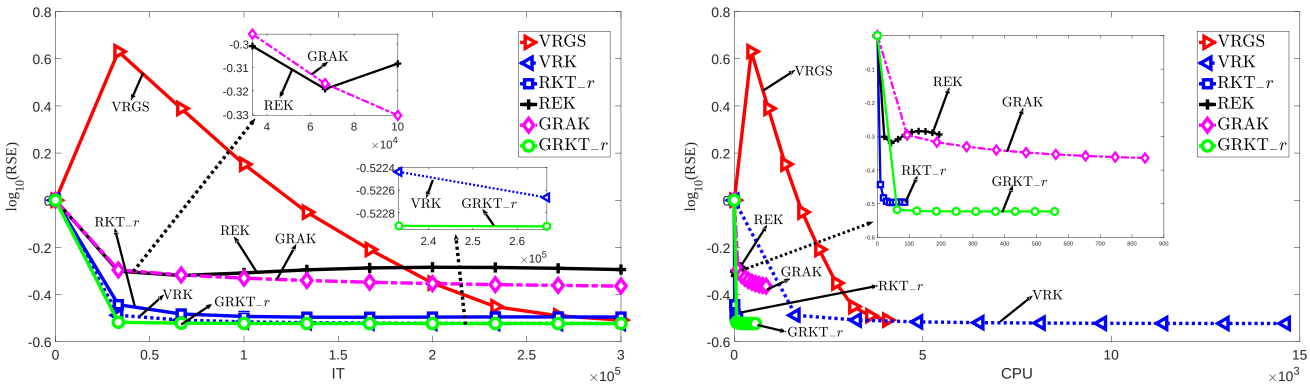

Fig. 2. $\log_{10}$(RSE) versus IT and CPU for RKT_r, VRK, VRGS, REK, GRAK and GRKT_r algorithms for blur image deblurring problem.

defined in the domain $[0, T] \times [0, 1] \times [0, 1]$

$$\frac{\partial u}{\partial t} = \nabla^2 u.$$

The forward problem involves mapping the initial condition u_0 *at time* $t = 0$ *to the solution* u_T *at time* $t = T$, *while the inverse problem entails reconstructing the initial condition* u_0 *from* u_T. *For such inverse problem, we can use MATLAB function* `PRdiffusion(n)` *from the Iterative Regularization Tools MATLAB package [30] to generate the forward computation matrix* $A \in \mathbb{R}^{n^2 \times n^2}$, *the noise-free data vector* $\bar{b} \in \mathbb{R}^{n^2}$ *and the true solution* $x_\star \in \mathbb{R}^{n^2}$. *In this example, we set a smaller* $n = 32$ *and convert* A *from a function handle into a sparse matrix of size* 1024×1024. *We add Gaussian white noise* e *with noise level* $\varrho = 0.01$ *to* $\bar{b}$ *and set the stopping tolerance* $\varepsilon = 0.1$, *or IT exceeds 300,000.*

We list IT, CPU, and RSE at 300,000th iteration for RKT_r, VRK, VRGS, REK, GRAK and GRKT_r in Table IV and plot the $\log_{10}$(RSE) curves with respect to IT and CPU in Figure 3. From Table IV we see that the RKT_r, VRK, VRGS and GRKT_r perform better than both REK and GRAK in terms of the RSE, and this can be seen intuitively from Figure 4, in which we display the restorations for the above six algorithms. However, the CPU times spend by these four algorithms varies greatly, roughly speaking, the CPU time of VRK, VRGS and GRKT_r is one hundred and seventy, forty-seven and six times that of RKT_r, respectively. In addition, we can see from the right of Figure 3 that RKT_r converges fastest among the above six algorithms, followed by GRKT_r. Hence, RKT_r and GRKT_r are the winners among all algorithms regarding restoration and CPU.

IV. Conclusion

In this work, we have proposed a more compact and economical version of the RKT algorithm, called the RKT_r algorithm, for solving Tikhonov regularization problems using the special structures of the coefficient matrix $\tilde{A}$. In addition, we have constructed an accelerated form named

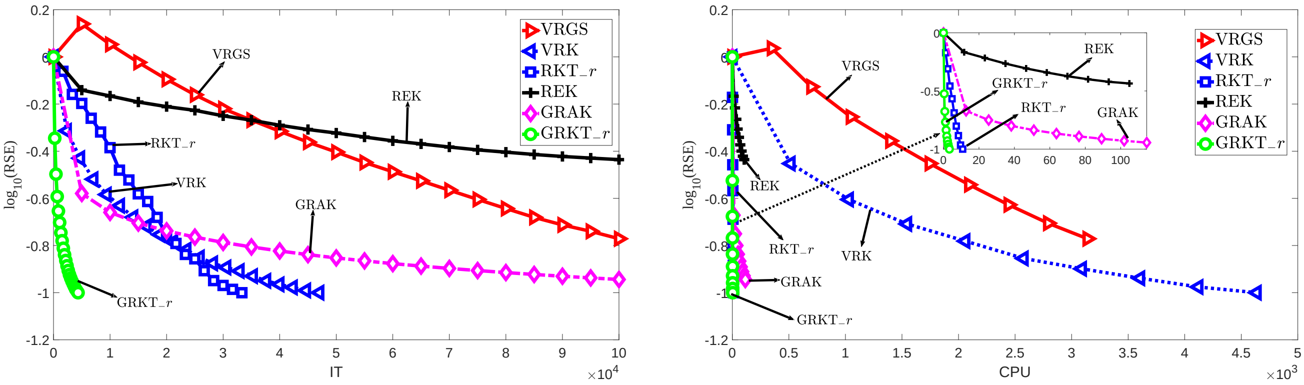

Fig. 3. $\log_{10}$(RSE) versus IT and CPU for RKT_r, VRK, VRGS, REK, GRAK and GRKT_r algorithms for 2D diffusion problem.

TABLE IV: Numerical results for 2D diffusion problem.

	Index		
Algorithm	IT	CPU($\times 10^3$)	RSE
RKT_r	300000.0	0.0860	0.3195
VRK	300000.0	14.6305	0.3051
VRGS	300000.0	4.0520	0.3102
REK	300000.0	0.1942	0.5080
GRAK	300000.0	0.8407	0.4324
GRKT_r	300000.0	0.5568	0.3012

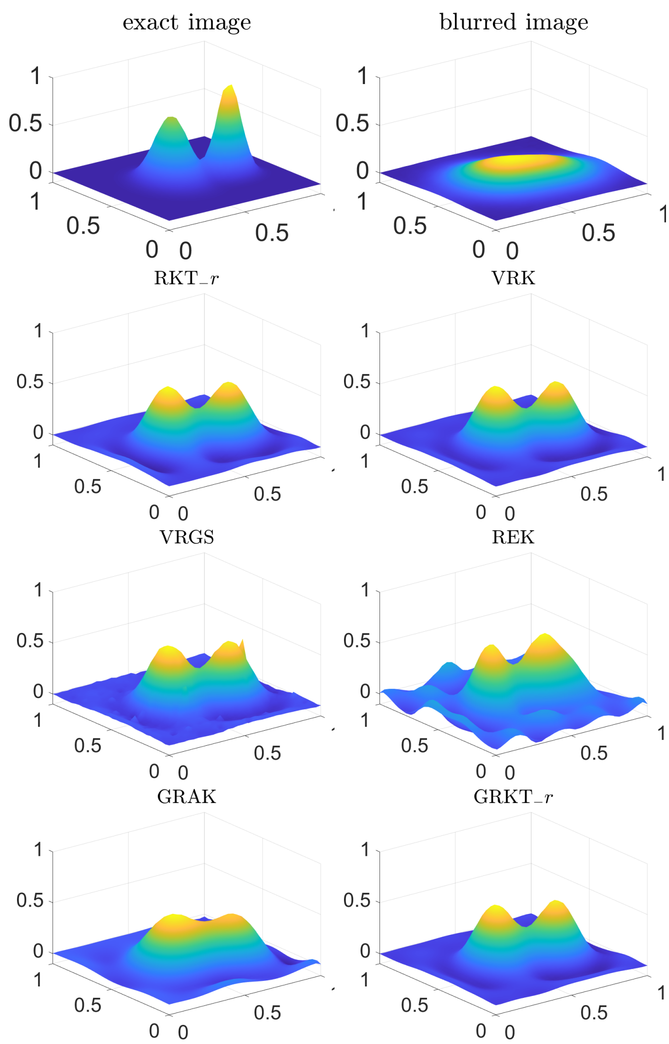

Fig. 4. Exact image and restorations corresponding to RKT_r, VRK, VRGS, REK, GRAK and GRKT_r algorithms for 2D diffusion problem.

GRKT_r for the RKT_r algorithm by making use of the greedy probability criterion proposed by Bai and Wu [27]. We have formulated the convergence theory for both RKT_r and GRKT_r with utmost precision and elegance, and performed a series of numerical experiments to show their numerical advantages over VRK, VRGS, REK, and GRAK algorithms. Generally speaking, RKT_r performs best of all these algorithms, because it costs the least CPU time when reaching the same stopping tolerance, especially for Example 2, there is a dramatic difference in CPU time between the other five algorithms and RKT_r algorithm. From the numerical experiments, it can be seen that all randomized iterative algorithms discussed in this paper cannot achieve satisfactory results because of the presence of noise in the original problem. Therefore, for Tikhonov regularization problems, it is a future topic to construct a randomized iterative algorithm similar to the REK algorithm to gradually reduce the impact of the noise during the iterative process, so as to achieve a good approximation to some actual application problems.

References

[1] Z. S. Ye, J. G. Li, and M. Zhang, "Application of ridge regression and factor analysis in design and production of alloy wheels," *J. Appl. Stat.,* vol. 41, no. 7-8, pp. 1436-1452, 2014.
[2] A. Hadgu, "An application of ridge regression analysis in the study of syphilis data," *Stat. Med.,* vol. 3, no. 3, pp. 293-299, 2010.
[3] T. Zhang, "On the dual formulation of regularized linear

systems with convex risks," *Mach. Learn.,* vol. 46, no. 1, pp. 91-129, 2002.
[4] 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, Int. J. Comput. Sci.*, vol. 47, no. 1, pp. 118-129, 2020.
[5] 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, Int. J. Comput. Sci.*, vol. 47, no. 1, pp. 68-74, 2020.
[6] P. C. Hansen, "Regularization tools version 4.0 for MATLAB 7.3," *Numer. Algorithms,* vol. 46, no. 2, pp. 189-194, 2007.
[7] P. C. Hansen, "Discrete Inverse Problems: Insight and Algorithms," Philadelphia, PA: SIAM, 2010.
[8] P. C. Hansen, J. S. Jørgensen, "AIR Tools II: algebraic iterative reconstruction methods, improved implementation," *Numer. Algorithms,* vol. 79, no. 1, pp. 107-137, 2018.
[9] Y. Liu and C. Q. Gu, "Two greedy subspace Kaczmarz algorithm for image reconstruction," *IAENG, Int. J. Appl. Math.,* vol. 50, no. 4, pp. 853-859, 2020.
[10] 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," *Eng. Lett.*, vol. 27, no. 1, pp. 151-165, 2019.
[11] M. X. Chen, B. X. Liu, D. S. Zeng, and W. Gao, "A framework for ontology-driven similarity measuring using vector learning tricks," *Eng. Lett.*, vol. 27, no. 3, pp. 549-558, 2019.
[12] G. T. Herman and L. B. Meyer, "Algebraic reconstruction techniques can be made computationally efficient," *IEEE Trans. Med. Imaging,* vol. 12, no. 3, pp. 600-609, 1993.
[13] G. T. Herman, "Fundamentals of Computerized Tomography: Image Reconstruction from Projections, 2nd. ed," New York: Springer, 2009.
[14] A. C. Kak and M. Slaney, "Principles of Computerized Tomographic Imaging," Philadelphia, PA: SIAM, 2001.
[15] F. Natterer, "The Mathematics of Computerized Tomography," Philadelphia, PA: SIAM, 2001.
[16] P. C. Hansen, "Analysis of discrete ill-posed problems by means of the L-curve," *SIAM Rev.,* vol. 34, no. 4, pp. 561-580, 1992.
[17] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics,* vol. 21, no. 2, pp. 215-223, 1979.
[18] A. Hefny, D. Needell, and A. Ramdas, "Rows versus columns: randomized Kaczmarz or Gauss-Seidel for ridge regression," *SIAM J. Sci. Comput.,* vol. 39, no. 5, pp. S528-S542, 2017.
[19] Z. Z. Bai, A. Buccini, K. Hayami, et al, "Modulus-based iterative methods for constrained Tikhonov regularization," *J. Comput. Appl. Math.,* vol. 319, pp. 1-13, 2017.
[20] S. Morigi, L. Reichel, F. Sgallari, et al, "An iterative method for linear discrete ill-posed problems with box constraints," *J. Comput. Appl. Math.,* vol. 198, no. 2, pp. 505-520, 2007.
[21] S. Morigi, L. Reichel, and F. Sgallari, "An interior-point method for large constrained discrete ill-posed problems,"*J. Comput. Appl. Math.,* vol. 233, no. 5, pp. 1288-1297, 2010.
[22] M. Rojas and T. Steihaug, "An interior-point trust-region-based method for large-scale non-negative regularization," *Inverse Problems,* vol. 18, no. 5, pp. 1291, 2002.
[23] A. A. Ivanov and A. I. Zhdanov, "Kaczmarz algorithm for Tikhonov regularization problem," *Appl. Math. E-Notes,* vol. 13, pp. 270-276, 2013.
[24] S. Kaczmarz, "Angenäherte Auflösung von Systemen linearer Gleichungen," *Bull. Int. Acad. Polonaise Sci. Lett.,* vol. 35, pp. 355-357, 1937.
[25] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *J. Fourier Anal. Appl.,* vol. 15, no. 2, pp. 262-278, 2009.
[26] A. Zouzias and N. M. Freris, "Randomized extended Kaczmarz for solving least-squares," *SIAM J. Matrix Anal. Appl.,* vol. 34, no. 2, pp. 773-793, 2013.
[27] Z. Z. Bai and W. T. Wu, "On greedy randomized Kaczmarz method for solving large sparse linear systems," *SIAM J. Sci. Comput.,* vol. 40, no. 1, pp. A592-A606, 2018.
[28] Y. Liu, S. M. Liu, and Y. Z. Zhang, "Accelerated greedy randomized Kaczmarz algorithm for solving linear systems," *Int. J. Comput. Sci.,* vol. 50, no. 3, pp. 1009-1015, 2023.
[29] Z. Z. Bai and W. T. Wu, "On greedy randomized augmented Kaczmarz method for solving large sparse inconsistent linear systems," *SIAM J. Sci. Comput.,* vol. 43, no. 6, pp. A3892-A3911, 2021.
[30] S. Gazzola, P. C. Hansen, and J. G. Nagy, "IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems," *Numer. Algorithms,* vol. 81, no. 3, pp. 773-811, 2019.