Accelerated Greedy Randomized Kaczmarz Algorithm for Solving Linear Systems

Yong Liu, Shimin Liu and Zhiyong Zhang

***Abstract*—The greedy randomized Kaczmarz (GRK) algorithm is more powerful than the randomized Kaczmarz (RK) algorithm for large sparse consistent linear systems. Motivated by GRK algorithm and Nesterov's acceleration scheme, we propose an accelerated greedy randomized Kaczmarz (AGRK) algorithm by using an equal probability criterion. We present convergence analysis for the AGRK algorithm and compare its performance and effectiveness with the GRK algorithm on random matrices as well as real-world datasets. Numerical results show that the AGRK algorithm performs better than the GRK algorithm in both iteration counts and computing times.**

***Index Terms*—consistent linear systems, greedy randomized Kaczmarz, equal probability criterion, Nesterov acceleration, convergence property.**

I. Introduction

We consider an iterative solution of a consistent linear system of the form

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $m \geq n$, is of full column rank, $b \in \mathbb{R}^m$ is an m-dimensional vector and $x \in \mathbb{R}^n$ is an n-dimensional unknown vector. Under these assumptions, we know that the linear system (1) admits a unique solution $x_* = (A^TA)^{-1}A^Tb = A^\dagger b$, where $A^\dagger$ denotes the Moore-Penrose pseudoinverse of the matrix A and the superscript ^{T} is the transpose of a vector or a matrix; see, e.g., [1]-[3]. The randomized Kaczmarz (RK) algorithm [4] proposed in 2009 is a powerful iterative projection algorithm for solving such system (1), and it is a worthy development and quality improvement of the original Kaczmarz algorithm [5] by introducing a probabilistic frame for the selection of target rows in each iteration. More precisely, let $A^{(i)}$ and $b^{(i)}$ be the i-th row of the matrix A and the i-th entry of the vector b, respectively, then given initial estimate x_0, the RK algorithm can be described as

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

where $\|\cdot\|_2$ is the Euclidean norm and the target row i_k is randomly selected from all rows of the coefficient matrix A with probability $P(\text{row} = i_k) = \frac{\|A^{(i_k)}\|_2^2}{\|A\|_F^2}$, in which $\|A\|_F$ represents the Frobenius norm of matrix A.

To further improve the convergence behavior of the RK algorithm, Bai and Wu [1] proposed a new probability criterion involving the residual vector of the linear system at each iteration step of the RK algorithm, and constructed a greedy randomized Kaczmarz (GRK) algorithm, which performs better than the RK algorithm in both theory and experiments; see also [6]-[8]. The GRK algorithm has been widely discussed and generalized by many researchers. For example, Liu and Gu [9] applied the GRK algorithm to ridge regression problem [10],[11], Zhang [12] proposed a different greedy Kaczmarz algorithm for (1) by using the greedy idea of the GRK, Jiang et al. [13], Gu [14], Niu and Zheng [15] proposed some block variants of the GRK algorithm. Both the RK and the GRK algorithms are especially preferred for the case of $m \gg n$. However, when m is close to n, both algorithms converge very slowly, see [16],[17]. For this problem, Liu and Wright [18] proposed an accelerated RK (ARK) algorithm by using Nesterov's acceleration scheme [19], and their numerical experiments show that the ARK converges much faster than the RK when m is close to n. Inspired by the work of [18], Morshed et al. [20] applied also Nesterov's acceleration scheme to the generalized sampling Kaczmarz Motzkin (SKM) algorithm for solving large-scale linear feasibility problems and showed that the resulting method, i.e., accelerated SKM (ASKM) algorithm, has higher computational efficiency than the original method. For more details about the application of Nesterov's acceleration scheme, we refer to [21]-[25].

Motivated by the success of the ARK and the ASKM, we intend to use Nesterov's acceleration scheme to construct an accelerated version of the GRK algorithm. Compared with the ARK algorithm, the difficulty of our generalization is that it seems impossible to choose the working row with equal probability (even if the coefficient matrix A of (1) is row-normalized), which is the key to demonstrate the convergence of the ARK algorithm. To solve this problem, we will additionally make use of the uniform probability to construct the accelerated GRK (AGRK) algorithm. In theory, we prove that the AGRK algorithm for (1) converges with the expected exponential rate. And in computations, we show that the AGRK algorithm has higher computing efficiency than the GRK algorithm in both iteration steps and computation times.

The paper is structured as follows. In Section II we first describe the GRK algorithm and then present the algorithmic description of the AGRK algorithm. In Section III we prove the convergence of the AGRK algorithm. Numerical results are given in Section IV. Finally, in Section V we end the paper with succinct concluding remarks.

Manuscript received December 27, 2022; revised July 19, 2023.

This work was supported by the National Natural Science Foundation (61773003) and the Youth Foundation of Changshu Institute of Technology (22ZK3762).

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).

Shimin Liu is a lecturer of the School of Mathematics and Statistics, Changshu Institute of Technology, Suzhou, Jiangsu, 215500, China (E-mail: liushimin@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).

II. The AGRK algorithm

Throughout this paper, we use A_τ to represent the row submatrix of A indexed by a subset τ of the row indices of A and b_τ to represent the subvector of b with components listed in τ. For a real matrix $A \in \mathbb{R}^{m\times n}$, we use A^T, $A^\dagger$, $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ to represent its transpose, Moore-Penrose pseduinverse, smallest nonzero eigenvalue and largest eigenvalue, respectively. The inner product in $\mathbb{R}^n$ is represented by $\langle\cdot,\cdot\rangle$. Given a positive definite matrix M, $\|x\|_M = \sqrt{x^TMx} = \|Mx\|_2$ denotes the energy norm (induced by M) of any $x \in \mathbb{R}^n$.

For a positive integer ℓ, the authors of [1] constructed the following GRK algorithm.

Algorithm 1 GRK Algorithm [1]

Require: A, x_0, b and ℓ.
Ensure: x_ℓ.
1: **for** $k = 0, 1, 2, \cdots, \ell-1$ **do**
2: Compute $\eta_k = \frac{\|b-Ax_k\|_2^2}{\|A\|_F^2}$ and

$$\epsilon_k = \frac{1}{2}\left(\max_{1\le i_k\le m}\left\{\frac{\left|b^{(i_k)} - A^{(i_k)}x_k\right|^2}{\|A^{(i_k)}\|_2^2}\right\} + \eta_k\right)$$

3: Compute the index set $\mathcal{U}_k$ by

$$\mathcal{U}_k = \left\{i_k \middle| \ |b^{(i_k)} - A^{(i_k)}x_k|^2 \ge \epsilon_k\|A^{(i_k)}\|_2^2\right\}$$

4: Determine the ith entry $\tilde{r}_k^{(i)}$ of the vector $\tilde{r}_k$ by using

$$\tilde{r}_k^{(i)} = \begin{cases} b^{(i)} - A^{(i)}x_k, & \text{if } i \in \mathcal{U}_k, \\ 0, & \text{otherwise} \end{cases}$$

5: Pick $i_k \in \mathcal{U}_k$ with probability Pr(row$= i_k$)$= \frac{|\tilde{r}_k^{(i_k)}|^2}{\|\tilde{r}_k\|_2^2}$
6: Set $x_{k+1} = x_k + \frac{b^{(i_k)} - A^{(i_k)}x_k}{\|A^{(i_k)}\|_2^2}(A^{(i_k)})^T$
7: **end for**

Analogous to the ARK algorithm, the main idea of the AGRK algorithm is to introduce Nesterov's accelerated procedure into the GRK algorithm, so it can be easily described as Algorithm 2.

From Algorithm 2, we know that the AGRK algorithm can be viewed as the GRK algorithm with equal probability criterion in conjunction with Nesterov's acceleration scheme.

III. Convergence analysis

In this section, we will give the convergence theory for AGRK. Here and in the sequel we set $\|A^{(i)}\|_2 = 1$ for any $i \in \{1, 2, \cdots, m\}$. We let $\mathbb{E}_k$ to be the expected value conditional on the first k iterations, i.e.,

$$\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot|\ i_0, i_1, \cdots, i_{k-1}],$$

where $i_t (t = 0, 1, \cdots, k-1)$ is the t-th row chosen at the t-th iterate. Then, from the law of iterated expectations, we have $\mathbb{E}[\mathbb{E}_k[\cdot]] = \mathbb{E}[\cdot]$.

For the AGRK algorithm, we can establish the following convergence theorem.

Algorithm 2 AGRK Algorithm

Require: A, b, ℓ, x_0, $v_0 = x_0$, $\gamma_{-1} = 0$.
Ensure: x_ℓ.
1: **for** $k = 0, 1, 2, \cdots, \ell-1$ **do**
2: Compute $\eta_k = \frac{\|b-Ax_k\|_2^2}{\|A\|_F^2}$ and

$$\epsilon_k = \frac{1}{2}\left(\max_{1\le i_k\le m}\left\{\frac{\left|b^{(i_k)} - A^{(i_k)}x_k\right|^2}{\|A^{(i_k)}\|_2^2}\right\} + \eta_k\right) \quad (2)$$

3: Compute the index set $\mathcal{U}_k$ by

$$\mathcal{U}_k = \left\{i_k \middle| \ \frac{|b^{(i_k)} - A^{(i_k)}x_k|^2}{\|A^{(i_k)}\|_2^2} \ge \epsilon_k\right\} \quad (3)$$

4: Choose $\lambda_k \in \left[0, \lambda_{\min}(A_{\mathcal{U}_k}^T A_{\mathcal{U}_k})\right)$
5: Compute the large root γ_k of

$$\gamma_k^2 - \frac{1}{|\mathcal{U}_k|}\gamma_k = \left(1 - \frac{\lambda_k}{|\mathcal{U}_k|}\gamma_k\right)\gamma_{k-1}^2 \quad (4)$$

6: Set α_k and β_k as follows:

$$\alpha_k = \frac{|\mathcal{U}_k| - \lambda_k\gamma_k}{\gamma_k\left(|\mathcal{U}_k|^2 - \lambda_k\right)} \quad (5)$$

$$\beta_k = 1 - \frac{\lambda_k}{|\mathcal{U}_k|}\gamma_k \quad (6)$$

7: Set

$$y_k = \alpha_k v_k + (1-\alpha_k)x_k \quad (7)$$

8: Determine the ith entry $\tilde{\tilde{r}}_k^{(i)}$ of the vector $\tilde{\tilde{r}}_k$ by using

$$\tilde{\tilde{r}}_k^{(i)} = \begin{cases} 1, & \text{if } i \in \mathcal{U}_k, \\ 0, & \text{otherwise} \end{cases}$$

9: Choose $i_k \in \mathcal{U}_k$ with probability Pr(row$= i_k$)$= \frac{1}{|\mathcal{U}_k|}$
10: Set $x_{k+1} = y_k - \frac{A^{(i_k)}y_k - b^{(i_k)}}{\|A^{(i_k)}\|_2^2}(A^{(i_k)})^T$
11: Set

$$v_{k+1} = \beta_k v_k + (1-\beta_k)y_k - \gamma_k\frac{A^{(i_k)}y_k - b^{(i_k)}}{\|A^{(i_k)}\|_2^2}(A^{(i_k)})^T \quad (8)$$

12: **end for**

Theorem 1. *For any initial vector x_0 in the column space of A^T, the sequence $\{x_k\}$ generated by the AGRK algorithm converges linearly to the unique solution $x_* = A^\dagger b$ of (1) in expectation with error estimate as follows*

$$\mathbb{E}\left(\|x_{k+1} - x_*\|^2_{(A_{\mathcal{U}_k}^T A_{\mathcal{U}_k})^\dagger}\right) < \frac{4\lambda_{\min}^k\|x_0 - x_*\|_2^2}{(\check{\delta}_k - \hat{\delta}_k)^2},$$

in which

$$\hat{\delta}_k = \prod_{i=0}^{k}\left(1 - \frac{\sqrt{\lambda_{\min}^k}}{2|\mathcal{U}_i|}\right) \text{ and } \check{\delta}_k = \prod_{i=0}^{k}\left(1 + \frac{\sqrt{\lambda_{\min}^k}}{2|\mathcal{U}_i|}\right),$$

where

$$\lambda_{\min}^k = \min\{\lambda_0, \lambda_1 \cdots, \lambda_k\} \text{ with } \lambda_k \in \left[0, \lambda_{\min}(A_{\mathcal{U}_k}^T A_{\mathcal{U}_k})\right).$$

The proof of Theorem 1 follows the framework derived by Liu and Wright [18] for the ARK algorithm and the

convergence proofs for the ASKM algorithm proposed by Morshed et al. [20]. We commence with two simple lemmas.

Lemma 1. *For any $y \in \mathbb{R}^n$, we have*

$$\mathbb{E}_k\left(\left\|\left(A^{(i_k)}y - b^{(i_k)}\right)(A^{(i_k)})^T\right\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^\dagger}\right) \le \frac{1}{|\mathcal{U}_k|}\|A_{\mathcal{U}_k}y - b_{\mathcal{U}_k}\|_2^2,$$

where the random variable i_k follows the uniform distribution over the set $\mathcal{U}_k$ defined in (3).

Proof: Let $A_{\mathcal{U}_k} = U_{\mathcal{U}_k}\Sigma_{\mathcal{U}_k}V^T_{\mathcal{U}_k}$ be the singular value decomposition of $A_{\mathcal{U}_k}$, then $(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1} = V_{\mathcal{U}_k}\Sigma^{-2}_{\mathcal{U}_k}V^T_{\mathcal{U}_k}$. As a result, we see that

$$\begin{aligned}
&\mathbb{E}_k\left(\left\|\left(A^{(i_k)}y - b^{(i_k)}\right)(A^{(i_k)})^T\right\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\right)\\
&= \frac{1}{|\mathcal{U}_k|}trace\left[U^T_{\mathcal{U}_k}diag\left(A_{\mathcal{U}_k}y - b_{\mathcal{U}_k}\right)^2U_{\mathcal{U}_k}\right]\\
&= \frac{1}{|\mathcal{U}_k|}\left\|diag\left(A_{\mathcal{U}_k}y - b_{\mathcal{U}_k}\right)^2U_{\mathcal{U}_k}\right\|_F^2\\
&= \frac{1}{|\mathcal{U}_k|}\sum_{i_k\in\mathcal{U}_k}\left(A^{(i_k)}y - b^{(i_k)}\right)^2\left\|U^{(i_k)}_{\mathcal{U}_k}\right\|_2^2\\
&\le \frac{1}{|\mathcal{U}_k|}\sum_{i_k\in\mathcal{U}_k}\left(A^{(i_k)}y - b^{(i_k)}\right)^2\\
&= \frac{1}{|\mathcal{U}_k|}\|A_{\mathcal{U}_k}y - b_{\mathcal{U}_k}\|_2^2.
\end{aligned}$$

Lemma 2. *For the solution x_* of (1), we have*

$$\mathbb{E}_k\left(\|x_{k+1} - x_*\|_2^2\right) = \|y_k - x_*\|_2^2 - \frac{1}{|\mathcal{U}_k|}\|A_{\mathcal{U}_k}y_k - b_{\mathcal{U}_k}\|_2^2.$$

Proof: According to the step 10 of the AGRK algorithm, we can obtain

$$\begin{aligned}
&\mathbb{E}_k\left(\|x_{k+1} - x_*\|_2^2\right)\\
&= \mathbb{E}_k\left(\|y_k - (A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T - x_*\|_2^2\right)\\
&= \|y_k - x_*\|_2^2 + \mathbb{E}_k(|A^{(i_k)}y_k - b^{(i_k)}|^2)\\
&\quad - 2\mathbb{E}_k\sum_{i_k\in\mathcal{U}_k}\left\langle y_k - x_*, (A^{(i_k)})^T(A^{(i_k)}y_k - b^{(i_k)})\right\rangle\\
&= \|y_k - x_*\|_2^2 + \frac{1}{|\mathcal{U}_k|}\sum_{i_k\in\mathcal{U}_k}|A^{(i_k)}y_k - b^{(i_k)}|^2\\
&\quad - \frac{2}{|\mathcal{U}_k|}\sum_{i_k\in\mathcal{U}_k}(A^{(i_k)}y_k - b^{(i_k)})\cdot A^{(i_k)}(y_k - x_*)\\
&= \|y_k - x_*\|_2^2 + \frac{1}{|\mathcal{U}_k|}\sum_{i_k\in\mathcal{U}_k}|A^{(i_k)}y_k - b^{(i_k)}|^2\\
&\quad - \frac{2}{|\mathcal{U}_k|}\sum_{i_k\in\mathcal{U}_k}|A^{(i_k)}y_k - b^{(i_k)}|^2\\
&= \|y_k - x_*\|_2^2 - \frac{1}{|\mathcal{U}_k|}\|A_{\mathcal{U}_k}y_k - b_{\mathcal{U}_k}\|_2^2.
\end{aligned}$$

Here in the fourth equality we have used the consistency of the system (1), i.e., $A^{(i_k)}x_* = b^{(i_k)}$.

Proposition 1. *Assume that $\gamma_k \le 1/\sqrt{\lambda_{k+1}}$, then it follows that both α_k and β_k, defined in (5) and (6) respectively, are lie in (0,1] for all k.*

Proof: Define quadratic function $f_k : \mathbb{R} \to \mathbb{R}$ as follows:

$$f_k(\gamma) = \gamma^2 - \frac{\gamma}{|\mathcal{U}_k|}(1 - \lambda_k\gamma_{k-1}^2) - \gamma_{k-1}^2, \quad k \in \{0, 1, 2, \cdots\}.$$

Then it follows that $f_k(0) < 0$ and $f_k(1/|\mathcal{U}_k|) < 0$, which imply that $\gamma_k > 1/|\mathcal{U}_k|$ for all $k \ge 0$.

Using the assumption that $\gamma_k \le 1/\sqrt{\lambda_{k+1}}$, we have

$$f_k(\gamma_{k-1}) = -(\gamma_{k-1}/|\mathcal{U}_k|)(1 - \lambda_k\gamma_{k-1}^2) \le 0$$

and

$$\begin{aligned}
f_k(1/\sqrt{\lambda_k}) &= \frac{1}{\lambda_k} - \frac{1}{|\mathcal{U}_k|\sqrt{\lambda_k}}(1 - \lambda_k\gamma_{k-1}^2) - \gamma_{k-1}^2\\
&= \frac{1}{\lambda_k} - \frac{1}{|\mathcal{U}_k|\sqrt{\lambda_k}} + \gamma_{k-1}^2\left(\frac{\sqrt{\lambda_k}}{|\mathcal{U}_k|} - 1\right)\\
&\ge \frac{1}{\lambda_k} - \frac{1}{|\mathcal{U}_k|\sqrt{\lambda_k}} + \frac{1}{\lambda_k}\left(\frac{\sqrt{\lambda_k}}{|\mathcal{U}_k|} - 1\right) = 0,
\end{aligned}$$

which indicate that

$$\gamma_k \in \left[\gamma_{k-1}, 1/\sqrt{\lambda_k}\right].$$

The above relation straightforwardly leads to

$$0 < |\mathcal{U}_k| - \sqrt{\lambda_k} \le |\mathcal{U}_k| - \lambda_k\gamma_k < \gamma_k|\mathcal{U}_k|^2 - \lambda_k\gamma_k$$

and

$$0 < 1 - \frac{\sqrt{\lambda_k}}{|\mathcal{U}_k|} \le 1 - \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|} \le 1.$$

Recalling that the definitions in (5) and (6), we immediately achieve the conclusion that we were proving.

Based on Lemma 1, Lemma 2, and Proposition 1, we can give the following proof of Theorem 1:

Let $r_k = \|v_k - x_*\|_{(A^T_{\mathcal{U}_{k-1}}A_{\mathcal{U}_{k-1}})^{-1}}$ with $\mathcal{U}_{-1} = \mathcal{U}_0$, then it holds that

$$\begin{aligned}
r_{k+1}^2 &= \|v_{k+1} - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\\
&= \|\beta_kv_k + (1-\beta_k)y_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\\
&\quad + \gamma_k^2\|(A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\\
&\quad - 2\gamma_k\Big\langle\beta_kv_k + (1-\beta_k)y_k - x_*,\\
&\qquad (A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}(A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\Big\rangle\\
&= \|\beta_kv_k + (1-\beta_k)y_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\\
&\quad + \gamma_k^2\|(A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\\
&\quad - 2\gamma_k\Big\langle\beta_k\left(\frac{1}{\alpha_k}y_k - \frac{1-\alpha_k}{\alpha_k}x_k\right) + (1-\beta_k)y_k\\
&\quad - x_*, (A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}(A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\Big\rangle\\
&= \|\beta_kv_k + (1-\beta_k)y_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\\
&\quad + \gamma_k^2\|(A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\\
&\quad + 2\gamma_k\Big\langle x_* - y_k + \frac{1-\alpha_k}{\alpha_k}\beta_k(x_k - y_k),\\
&\quad (A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}(A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\Big\rangle, \qquad (9)
\end{aligned}$$

where the last equality follows from (7).

Now we are going to estimate the three terms in (9), respectively. For the first term in (9): since $\|\cdot\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}$ is a convex function and $\beta_k = 1 - \frac{\lambda_k}{|\mathcal{U}_k|}\gamma_k \in (0,1]$, we have

$$\begin{aligned}
&\|\beta_k v_k + (1-\beta_k)y_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} \\
&\quad\le \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + (1-\beta_k)\|y_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} \\
&\quad= \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}\|y_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} \\
&\quad= \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \\
&\qquad\qquad \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}\langle (A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}(y_k - x_*), y_k - x_*\rangle \\
&\quad= \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \\
&\qquad\qquad \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}\langle V_{\mathcal{U}_k}\Sigma^{-2}_{\mathcal{U}_k}V^T_{\mathcal{U}_k}(y_k - x_*), y_k - x_*\rangle \\
&\quad= \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \\
&\qquad\qquad \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}(y_k - x_*)^T V_{\mathcal{U}_k}\Sigma^{-2}_{\mathcal{U}_k}V^T_{\mathcal{U}_k}(y_k - x_*) \\
&\quad= \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}\|\Sigma^{-1}_{\mathcal{U}_k}V^T_{\mathcal{U}_k}(y_k - x_*)\|_2^2 \\
&\quad\le \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}\|\Sigma^{-1}_{\mathcal{U}_k}V^T_{\mathcal{U}_k}\|_2^2\|y_k - x_*\|_2^2 \\
&\quad\le \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}\|\Sigma^{-1}_{\mathcal{U}_k}\|_2^2\|y_k - x_*\|_2^2 \\
&\quad= \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}\frac{\|y_k - x_*\|_2^2}{\lambda_{\min}(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})} \\
&\quad< \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \frac{\gamma_k}{|\mathcal{U}_k|}\|y_k - x_*\|_2^2, \qquad (10)
\end{aligned}$$

where the last inequality follows from $\lambda_k < \lambda_{\min}(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})$.

For the second term in (9): using Lemmas 1 and 2, we easily obtain

$$\begin{aligned}
&\mathbb{E}_k\left(\gamma_k^2\|(A^{(i_k)}y - b^{(i_k)})(A^{(i_k)})^T\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\right) \\
&\quad= \gamma_k^2\mathbb{E}_k\left(\|(A^{(i_k)}y - b^{(i_k)})(A^{(i_k)})^T\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}}\right) \\
&\quad\le \gamma_k^2\|y_k - x_*\|_2^2 - \gamma_k^2\mathbb{E}_k(\|x_{k+1} - x_*\|_2^2). \qquad (11)
\end{aligned}$$

For the third term in (9): we have

$$\begin{aligned}
&\mathbb{E}_k\Big[2\gamma_k\langle x_* - y_k + \frac{1-\alpha_k}{\alpha_k}\beta_k(x_k - y_k), \\
&\qquad (A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}(A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\rangle\Big] \\
&\quad= 2\gamma_k\Big\langle x_* - y_k + \frac{1-\alpha_k}{\alpha_k}\beta_k(x_k - y_k), \\
&\qquad (A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}\mathbb{E}_k\left((A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\right)\Big\rangle \\
&\quad= \frac{2\gamma_k}{|\mathcal{U}_k|}\Big\langle x_* - y_k + \frac{1-\alpha_k}{\alpha_k}\beta_k(x_k - y_k), \\
&\qquad (A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}\sum_{i_k\in\mathcal{U}_k}(A^{(i_k)}y_k - b^{(i_k)})(A^{(i_k)})^T\Big\rangle \\
&\quad= \frac{2\gamma_k}{|\mathcal{U}_k|}\Big\langle x_* - y_k + \frac{1-\alpha_k}{\alpha_k}\beta_k(x_k - y_k), \\
&\qquad (A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}A^T_{\mathcal{U}_k}A_{\mathcal{U}_k}(y_k - x_*)\Big\rangle \\
&\quad= \frac{2\gamma_k}{|\mathcal{U}_k|}\left\langle x_* - y_k + \frac{1-\alpha_k}{\alpha_k}\beta_k(x_k - y_k), y_k - x_*\right\rangle \\
&\quad= \frac{2\gamma_k}{|\mathcal{U}_k|}\left(-\|y_k - x_*\|_2^2 + \frac{1-\alpha_k}{\alpha_k}\beta_k\langle x_k - y_k, y_k - x_*\rangle\right) \\
&\quad= \frac{2\gamma_k}{|\mathcal{U}_k|}\Big(-\|y_k - x_*\|_2^2 + \frac{1-\alpha_k}{2\alpha_k}\beta_k\Big(\|x_k - x_*\|_2^2 \\
&\qquad\qquad - \|y_k - x_*\|_2^2 - \|x_k - y_k\|_2^2\Big)\Big) \\
&\quad\le -\left(\frac{2\gamma_k}{|\mathcal{U}_k|} + \gamma_{k-1}^2\beta_k\right)\|y_k - x_*\|_2^2 + \gamma_{k-1}^2\beta_k\|x_k - x_*\|_2^2. \\
&\qquad\qquad (12)
\end{aligned}$$

Here, the last inequality follows from the definition in (5) of the quantity α_k and the quadratic equation (4), which lead to the following equality

$$\frac{1-\alpha_k}{\alpha_k} = |\mathcal{U}_k|\cdot\frac{|\mathcal{U}_k|\gamma_k - 1}{|\mathcal{U}_k| - \lambda\gamma_k} = \frac{|\mathcal{U}_k|\gamma_{k-1}^2}{\gamma_k}.$$

Hence, with the substitution of (10), (11) and (12) into (9) we can obtain

$$\begin{aligned}
\mathbb{E}_k(r_{k+1}^2) &< \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + \frac{\gamma_k}{|\mathcal{U}_k|}\|y_k - x_*\|_2^2 \\
&\quad + \gamma_k^2\|y_k - x_*\|_2^2 - \gamma_k^2\mathbb{E}_k(\|x_{k+1} - x_*\|_2^2) \\
&\quad - \left(\frac{2\gamma_k}{|\mathcal{U}_k|} + \gamma_{k-1}^2\beta_k\right)\|y_k - x_*\|_2^2 \\
&\quad + \gamma_{k-1}^2\beta_k\|x_k - x_*\|_2^2 \\
&= \beta_k\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} - \gamma_k^2\mathbb{E}_k(\|x_{k+1} - x_*\|_2^2) \\
&\quad + \gamma_{k-1}^2\beta_k\|x_k - x_*\|_2^2, \qquad (13)
\end{aligned}$$

where the last equality is a consequence of the quadratic equation (4).

Defining two scalar sequences $\{A_k\}$ and $\{B_k\}$ as follows:

$$B_{k+1}^2 = \frac{B_k^2}{\beta_k}, A_{k+1}^2 = \gamma_k^2 B_{k+1}^2$$

with

$$A_k \ge 0,\ B_k \ge 0 \text{ and } B_0 \ne 0.$$

It is easy to see that $B_{k+1} \ge B_k$ for any $k \ge 0$ due to $\beta_k \in (0,1]$. On the other hand, by using equation (4), we can obtain

$$A_{k+1}^2 = \frac{B_k^2\gamma_k^2}{\beta_k} = \frac{A_k^2\gamma_k^2}{\beta_k\gamma_{k-1}^2} = \frac{A_k^2\gamma_k^2}{\gamma_k^2 - \gamma_k/|\mathcal{U}_k|},$$

which implies that $\{A_k\}$ is an increasing sequence. Also, it follows from directly computations that

$$B_{k+1}^2\gamma_k^2 = A_{k+1}^2, B_{k+1}^2\beta_k = B_k^2, B_{k+1}^2\beta_k\gamma_{k-1}^2 = A_k^2.$$

Hence, multiplying both sides of (13) by B_{k+1}^2, we have

$$\begin{aligned}
&B_{k+1}^2\mathbb{E}_k(r_{k+1}^2) + A_{k+1}^2\mathbb{E}_k(\|x_{k+1} - x_*\|_2^2) \\
&\quad< B_k^2\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + A_k^2\|x_k - x_*\|_2^2,
\end{aligned}$$

i.e.,

$$\begin{aligned}
&\mathbb{E}_k(B_{k+1}^2 r_{k+1}^2 + A_{k+1}^2(\|x_{k+1} - x_*\|_2^2)) \\
&\quad< B_k^2\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + A_k^2\|x_k - x_*\|_2^2. \qquad (14)
\end{aligned}$$

By taking the full expectation for both sides of (14), we can obtain

$$\begin{aligned}
&\mathbb{E}(B_{k+1}^2 r_{k+1}^2 + A_{k+1}^2(\|x_{k+1} - x_*\|_2^2)) \\
&\quad< \mathbb{E}(B_k^2\|v_k - x_*\|^2_{(A^T_{\mathcal{U}_k}A_{\mathcal{U}_k})^{-1}} + A_k^2\|x_k - x_*\|_2^2).
\end{aligned}$$

By applying induction to the above inequality, we can get

$$\begin{aligned}&\mathbb{E}(B_{k+1}^2 r_{k+1}^2 + A_{k+1}^2(\|x_{k+1}-x_*\|_2^2))\\ &< \mathbb{E}(B_k^2\|v_k - x_*\|^2_{(A_{\mathcal{U}_k}^T A_{\mathcal{U}_k})^{-1}} + A_k^2\|x_k - x_*\|_2^2)\\ &< \cdots\cdots\\ &< B_0^2 r_0^2 + A_0^2\|x_0 - x_*\|_2^2\\ &= B_0^2 r_0^2.\end{aligned}$$

Therefore, we have

$$\mathbb{E}(\|x_{k+1}-x_*\|_2^2) < \frac{B_0^2}{A_{k+1}^2} r_0^2. \tag{15}$$

Besides, it is straightforward to check that

$$B_k^2 = B_{k+1}^2\left(1 - \frac{\lambda_k}{|\mathcal{U}_k|}\gamma_k\right) = B_{k+1}^2\left(1 - \frac{\lambda_k A_{k+1}}{|\mathcal{U}_k| B_{k+1}}\right),$$

hence,

$$\frac{\lambda_k}{|\mathcal{U}_k|} A_{k+1}B_{k+1} = (B_{k+1}+B_k)(B_{k+1}-B_k).$$

Using the fact that $B_k \le B_{k+1}$, we can obtain

$$\begin{aligned}\frac{\lambda_k}{|\mathcal{U}_k|} A_{k+1}B_{k+1} &= (B_{k+1}+B_k)(B_{k+1}-B_k)\\ &\le 2B_{k+1}(B_{k+1}-B_k),\end{aligned}$$

i.e.,

$$B_{k+1} \ge B_k + \frac{\lambda_k}{2|\mathcal{U}_k|} A_{k+1}.$$

Furthermore, by denoting

$$\lambda_{\min}^k := \min\{\lambda_0, \lambda_1, \cdots, \lambda_k\},$$

we immediately get

$$B_{k+1} \ge B_k + \frac{\lambda_{\min}^k}{2|\mathcal{U}_k|} A_{k+1}. \tag{16}$$

On the other hand, we have

$$\begin{aligned}\frac{A_{k+1}^2}{B_{k+1}^2} - \frac{A_{k+1}}{|\mathcal{U}_k| B_{k+1}} &= \gamma_k^2 - \frac{\gamma_k}{|\mathcal{U}_k|} = \left(1 - \frac{\lambda_k\gamma_k}{|\mathcal{U}_k|}\right)\gamma_{k-1}^2\\ &= \beta_k\gamma_{k-1}^2 = \frac{\beta_k A_k^2}{B_k^2} = \frac{A_k^2}{B_{k+1}^2},\end{aligned}$$

and furthermore

$$\begin{aligned}\frac{1}{|\mathcal{U}_k|} A_{k+1}B_{k+1} &= A_{k+1}^2 - A_k^2\\ &= (A_{k+1}+A_k)(A_{k+1}-A_k)\\ &\le 2A_{k+1}(A_{k+1}-A_k),\end{aligned}$$

therefore,

$$A_{k+1} \ge A_k + \frac{B_{k+1}}{2|\mathcal{U}_k|} \ge A_k + \frac{B_k}{2|\mathcal{U}_k|}. \tag{17}$$

By combining the inequalities (16) and (17), we can obtain

$$\begin{aligned}\begin{pmatrix}A_{k+1}\\ B_{k+1}\end{pmatrix} &\ge \begin{pmatrix}1 & \frac{1}{2|\mathcal{U}_k|}\\ \frac{\lambda_{\min}^k}{2|\mathcal{U}_k|} & 1\end{pmatrix}\begin{pmatrix}A_k\\ B_k\end{pmatrix}\\ &\ge \begin{pmatrix}1 & \frac{1}{2|\mathcal{U}_k|}\\ \frac{\lambda_{\min}^k}{2|\mathcal{U}_k|} & 1\end{pmatrix}\begin{pmatrix}1 & \frac{1}{2|\mathcal{U}_{k-1}|}\\ \frac{\lambda_{\min}^k}{2|\mathcal{U}_{k-1}|} & 1\end{pmatrix}\begin{pmatrix}A_{k-1}\\ B_{k-1}\end{pmatrix}\\ &\ge \cdots\\ &\ge \begin{pmatrix}1 & \frac{1}{2|\mathcal{U}_k|}\\ \frac{\lambda_{\min}^k}{2|\mathcal{U}_k|} & 1\end{pmatrix}\begin{pmatrix}1 & \frac{1}{2|\mathcal{U}_{k-1}|}\\ \frac{\lambda_{\min}^k}{2|\mathcal{U}_{k-1}|} & 1\end{pmatrix}\cdots\begin{pmatrix}0\\ B_0\end{pmatrix}.\end{aligned}$$

Using the Jordan decomposition of the following matrix

$$\begin{aligned}\begin{pmatrix}1 & \frac{1}{2|\mathcal{U}_k|}\\ \frac{\lambda_{\min}^k}{2|\mathcal{U}_k|} & 1\end{pmatrix} &= \begin{pmatrix}-\sqrt{\frac{1}{\lambda_{\min}^k}} & \sqrt{\frac{1}{\lambda_{\min}^k}}\\ 1 & 1\end{pmatrix}\cdot\\ &\begin{pmatrix}1-\frac{\sqrt{\lambda_{\min}^k}}{2|\mathcal{U}_k|} & 0\\ 0 & 1+\frac{\sqrt{\lambda_{\min}^k}}{2|\mathcal{U}_k|}\end{pmatrix}\cdot\begin{pmatrix}-\frac{1}{2}\sqrt{\lambda_{\min}^k} & \frac{1}{2}\\ \frac{1}{2}\sqrt{\lambda_{\min}^k} & \frac{1}{2}\end{pmatrix},\end{aligned}$$

we can obtain

$$\begin{aligned}\begin{pmatrix}A_{k+1}\\ B_{k+1}\end{pmatrix} \ge &\begin{pmatrix}-\sqrt{\frac{1}{\lambda_{\min}^k}} & \sqrt{\frac{1}{\lambda_{\min}^k}}\\ 1 & 1\end{pmatrix}\begin{pmatrix}1-\frac{\sqrt{\lambda_{\min}^k}}{2|\mathcal{U}_k|} & 0\\ 0 & 1+\frac{\sqrt{\lambda_{\min}^k}}{2|\mathcal{U}_k|}\end{pmatrix}\\ &\cdots\begin{pmatrix}-\frac{1}{2}\sqrt{\lambda_{\min}^k} & \frac{1}{2}\\ \frac{1}{2}\sqrt{\lambda_{\min}^k} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}0\\ B_0\end{pmatrix}.\end{aligned}$$

Letting

$$\hat{\delta}_k = \prod_{i=0}^{k}\left(1 - \frac{\sqrt{\lambda_{\min}^k}}{2|\mathcal{U}_i|}\right) \quad\text{and}\quad \check{\delta}_k = \prod_{i=0}^{k}\left(1 + \frac{\sqrt{\lambda_{\min}^k}}{2|\mathcal{U}_i|}\right),$$

then the above inequality can be rewritten as

$$\begin{aligned}\begin{pmatrix}A_{k+1}\\ B_{k+1}\end{pmatrix} &\ge \begin{pmatrix}-\sqrt{\frac{1}{\lambda_{\min}^k}} & \sqrt{\frac{1}{\lambda_{\min}^k}}\\ 1 & 1\end{pmatrix}\begin{pmatrix}\hat{\delta}_k & 0\\ 0 & \check{\delta}_k\end{pmatrix}\\ &\begin{pmatrix}-\frac{1}{2}\sqrt{\lambda_{\min}^k} & \frac{1}{2}\\ \frac{1}{2}\sqrt{\lambda_{\min}^k} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}0\\ B_0\end{pmatrix} = \begin{pmatrix}\frac{B_0}{2\sqrt{\lambda_{\min}^k}}(\check{\delta}_k - \hat{\delta}_k)\\ \frac{B_0}{2}(\check{\delta}_k + \hat{\delta}_k)\end{pmatrix},\end{aligned}$$

which implies

$$A_{k+1} \ge \frac{B_0}{2\sqrt{\lambda_{\min}^k}}(\check{\delta}_k - \hat{\delta}_k) \quad\text{and}\quad B_{k+1} \ge \frac{B_0}{2}(\check{\delta}_k + \hat{\delta}_k).$$

Hence, with the substitution of these two bounds into (15) we have

$$\mathbb{E}\left(\|x_{k+1}-x_*\|^2_{(A_{\mathcal{U}_k}^T A_{\mathcal{U}_k})^{-1}}\right) < \frac{4\lambda_{\min}^k\|x_0 - x_*\|_2^2}{(\check{\delta}_k - \hat{\delta}_k)^2}.$$

IV. Numerical Experiments

In this section, we implement the GRK algorithm and our proposed AGRK algorithm on a consistent linear system (1) in MATLAB (R2016b). The coefficient matrix $A \in \mathbb{R}^{m\times n}$ of (1) is selected from the University of Florida sparse matrix collection [26] and the matrix collection randomly generated by MATLAB function `randn(m,n)` with different m and n. We set the right-hand side b is set to be Ax_*, where x_* is the solution vector generated by `randn(n,1)`. In addition, we set $\|A^{(i)}\|_2 = 1$ for any $i \in \{1, 2, \cdots, m\}$. All experiments are performed on a workstation with 2.67 GHz central processing unit (Intel(R) Core(TM) i5 CPU), 4.00 GB memory, and Windows operating system (Windows 10).

The numerical performance of GRK and AGRK are evaluated with respect to the iteration counts (shorten as 'IT') and the computation time in seconds (shorten as 'CPU'). All experimental results are the average of 50 times of repeated runs of the GRK and AGRK algorithms. We set the initial guess x_0 to be zero vector for all experiments and terminate the iterations when the relative solution error (RSE) at the current iterates x_k satisfy

$$\text{RSE} = \frac{\|x_k - x_*\|_2^2}{\|x_*\|_2^2} \le 10^{-6}.$$

A. Comparison of GRK and AGRK on random data

In order to select an appropriate parameter λ_k at the kth iterate of the AGRK algorithm, we must calculate the minimum nonzero eigenvalue of $A_{\mathcal{U}_k}^T A_{\mathcal{U}_k}$, which is the extremely arithmetically expensive step in AGRK algorithm. Considering the substantial amount of computation needed per iteration, in our experiments, we use $(1 - \sqrt{|\mathcal{U}_k|/n})^2$ as the approximate value of $\lambda_{\min}(A_{\mathcal{U}_k}^T A_{\mathcal{U}_k})$ according to the random matrix theory [27]. In addition, since the coefficient matrix $A \in \mathbb{R}^{m\times n}$ has been normalized to have unit row norm, $\lambda_{\min}(A_{\mathcal{U}_k}^T A_{\mathcal{U}_k}) \le 1$, and note that $\lambda_{\min}(A_{\mathcal{U}_k}^T A_{\mathcal{U}_k}) < 1$

unless $|\mathcal{U}_k| = 1$. Therefore, the parameter λ_k at the kth iterate of the AGRK algorithm can be chosen as

$$\lambda_k \approx (1 - \sqrt{|\mathcal{U}_k|/n})^{2p} \quad \text{for any} \quad p \in \{1, 2, \cdots\}. \tag{18}$$

For the random matrices, we study the computational behavior of the AGRK algorithm under different p values, and the corresponding algorithm is abbreviated as AGRK(p). We show the $\log_{10}$(RSE) versus the IT and the CPU time in Figures 1-2. From these two figures, we observe that, for each $p \in \{1, 2, 3, 4, 5, 6\}$, AGRK($p$) algorithm succeeds in solving the Gaussian systems and converges faster than the GRK algorithm in terms of both iteration counts and computing times. Loosely speaking, for a smaller value of n such as $n = 50$, the iteration counts and the computation times of the AGRK(4) are at least two and two times of those of the GRK, while for a larger value of n such as $n = 80$, the iteration counts and the computation times of the AGRK(4) are at least two and four times of those of the GRK, respectively. By comparing Figures 1 and 2, we see that when m is very close to n, the AGRK algorithm converges faster and faster with respect to the increase of the p value.

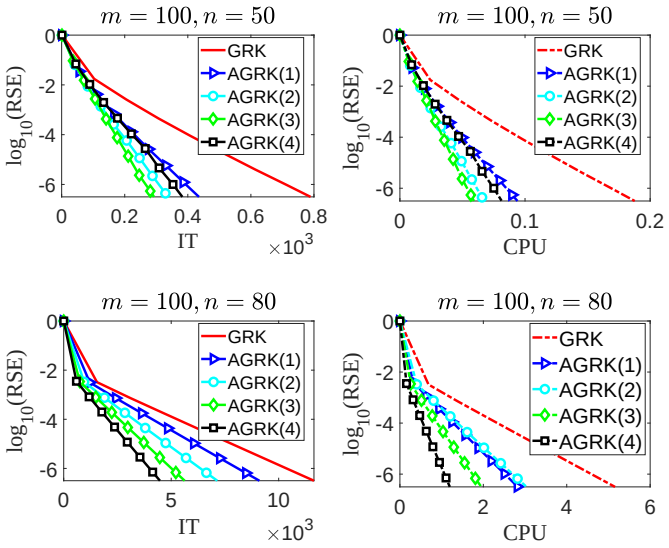

Fig. 1. $\log_{10}$(RSE) versus IT and CPU for GRK and AGRK(p) on random Gaussian system with $m = 100$ and $n = 50, 80$.

B. Comparison of GRK and AGRK for real-world non-random data

For a random matrix $A \in \mathbb{R}^{m\times n}$ whose entries are i.i.d. Gaussian with mean 0 and variance $\sigma^2 = 1$, we can use (18) to estimate the minimum nonzero eigenvalue $\lambda_{\min}(A^TA)$ of symmetric matrix A^TA. However, for a non-random matrix $A = (a_{ij}) \in \mathbb{R}^{m\times n}$, that is, the mean of a_{ij}, $i = 1, 2, \cdots, m$; $j = 1, 2, \cdots, n$ is not 0, and the variance σ^2 is not 1, the quantity (18) can no longer be used to estimate $\lambda_{\min}(A^TA)$. In this case, we are giving here the following quantity presented in [28]

$$(1 - \sqrt{m/n})^2\sigma^2$$

to estimate $\lambda_{\min}(A^TA)$, because the nonzero mean only affects $\lambda_{\max}(A^TA)$, and the variance that is not equal to 1 only affects the multiple of $\lambda_{\min}(A^TA)$ (see, e.g., [29], [30]). Therefore, the quantity $(1-\sqrt{m/n})^2\sigma^2$ can be used as an upper limit on the value of $\lambda_{\min}(A_{\mathcal{U}_k}^TA_{\mathcal{U}_k})$ at each iteration step of the AGRK. Of course, this estimation is not optimal, but it is much cheaper than directly calculating $\lambda_{\min}(A_{\mathcal{U}_k}^TA_{\mathcal{U}_k})$, so we set $\lambda_k = (1 - \sqrt{m/n})^2\sigma^2$ with σ^2 being the variance of the corresponding sparse matrices *Stranke94*, *Can_24*, *Cities* and *well1033*. Some features of the above four sparse matrices are listed in the following table.

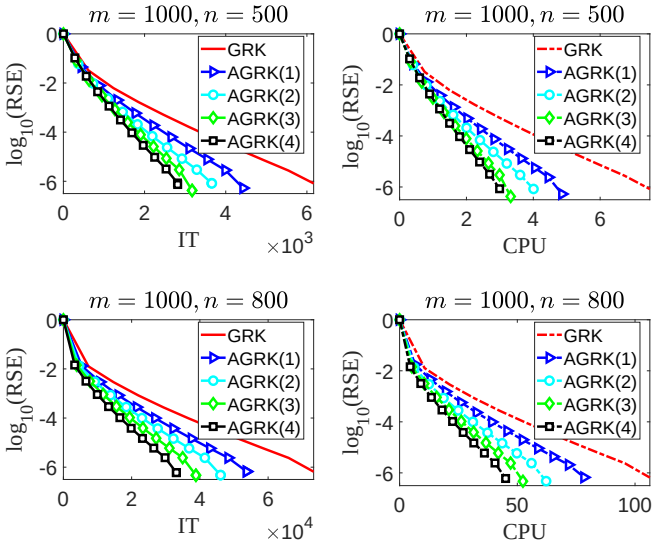

Fig. 2. $\log_{10}$(RSE) versus IT and CPU for GRK and AGRK(p) on random Gaussian system with $m = 1000$ and $n = 500, 800$.

TABLE I: Some properties of Florida sparse matrices.

Name	Stranke94	Can_24	Cities	well1033
$m \times n$	10×10	24×24	55×46	1033×320
density	90.00%	27.77%	53.04%	1.43%
cond(A)	51.73	77.75	207.15	166.13

In Figure 3, we show the $\log_{10}$(RSE) versus IT and CPU for the above four sparse matrices. The curves in Figure 3 show that the RSE of the AGRK is decaying more quickly than that of the GRK when the iteration counts or CPU time is increasing.

After analyzing Figures 1-3, we can conclude that when the coefficient matrix of (1) is very close to square matrix, the AGRK algorithm with appropriate choice of λ_k converges much faster than the GRK algorithm.

V. Conclusion

Nesterov's acceleration scheme was first successfully employed in the coordinate descent algorithm, later in the RK and the SKM algorithms, and has been further applied to the GRK algorithm. We have established the convergence theory for the resulting algorithm and derived an estimate about its convergence rate depending on the geometric properties of the submatrices of the coefficient matrix and on the size of a subset of the constraints at each step. Numerical experiments show that our proposed algorithm performs better than the GRK algorithm in both IT and CPU for consistent linear system with $m \geq n$, especially for the case where m is very close to n. In addition, the proposed algorithm can be more efficient than the GRK algorithm if the parameter λ_k in Algorithm 2 can be estimated more accurately. In fact, for our proposed algorithm, it is difficult to determine the best parameter λ_k due to the difficulty of estimating the smallest nonzero eigenvalue of the submatrix at each step. Hence, in the future, we plan to use the quantity ϵ_k defined in (2) instead of $\mathcal{U}_k$ defined in (3) and select λ_k associated with ϵ_k to construct a new accelerated GRK algorithm to avoid estimating the smallest nonzero eigenvalue of the submatrix at each step.

References

[1] 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.

[2] 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.

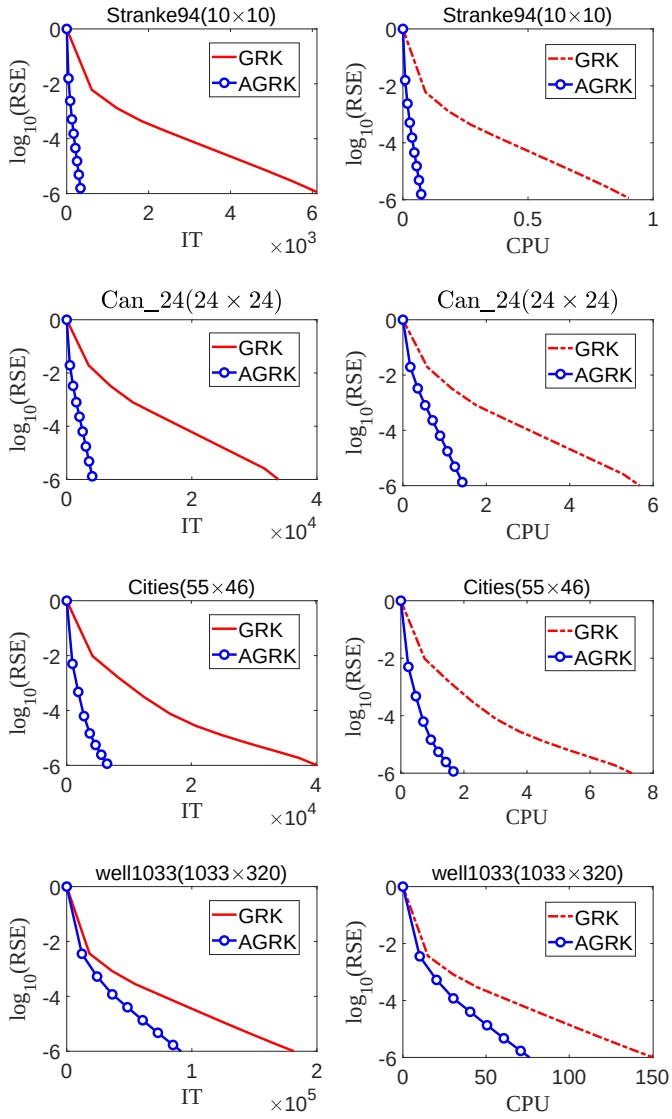

Fig. 3. $\log_{10}$(RSE) versus IT and CPU for GRK and AGRK on sparse matrices *Stranke94*, *Can_24*, *Cities* and *well1033*.

[3] A. Ma, D. Needell, and A. Ramdas, "Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods," *SIAM J. Matrix Anal. Appl.,* vol. 36, no. 4, pp. 1590-1604, 2015.

[4] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *J. Fourier Anal. Appl.,* vol. 15, no. 2, pp. 262-278, 2009.

[5] S. Kaczmarz, "Angenäherte Auflösung von Systemen linearer Gleichungen," *Bull. Int. Acad. Polonaise Sci. Lett.,* pp. 355-357, 1937.

[6] Z. Z. Bai and W. T. Wu, "On relaxed greedy randomized Kaczmarz methods for solving large sparse linear systems," *Appl. Math. Lett.,* vol. 83, pp. 21-26, 2018.

[7] Z. Z. Bai and W. T. Wu, "On greedy randomized coordinate descent methods for solving large linear least-squares problems," *Numer. Linear Algebra Appl.,* vol. 26, no. 4, pp. 1-15, 2019.

[8] Y. Liu and C. Q. Gu, "Two greedy subspace Kaczmarz algorithm for image reconstruction," *IAENG International Journal of Applied Mathematics,* vol. 50, no. 4, pp. 853-859, 2020.

[9] Y. Liu and C. Q. Gu, "Variant of greedy randomized Kaczmarz for ridge regression," *Appl. Numer. Math.,* vol. 143, pp. 223-246, 2019.

[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," *Engineering Letters,* vol. 27, no. 1, pp. 151-165, 2019.

[11] 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.

[12] J. J. Zhang, "A new greedy Kaczmarz algorithm for the solution of very large linear systems," *Appl. Math. Lett.,* vol. 91, pp. 207-212, 2019.

[13] X. L. Jiang, K. Zhang, and J. F. Yin, "Randomized block Kaczmarz methods with k-means clustering for solving large linear systems," *J. Comput. Appl. Math.,* vol. 403, pp. 113828, 2022.

[14] Y. Liu and C. Q. Gu, "On greedy randomized block Kaczmarz method for consistent linear systems," *Linear Algebra Appl.,* vol. 616, pp. 178-200, 2021.

[15] Y. Q. Niu and B. Zheng, "A greedy block Kaczmarz algorithm for solving large-scale linear systems," *Appl. Math. Lett.,* vol. 104, pp. 106294, 2020.

[16] Z. Z. Bai and X. G. Liu, "On the Meany inequality with applications to convergence analysis of several row-action iteration methods," *Numer. Math.,* vol. 124, no. 2, pp. 215-236, 2013.

[17] Z. Z. Bai and W. T. Wu, "On convergence rate of the randomized Kaczmarz method," *Linear Algebra Appl.,* vol. 553, pp. 252-269, 2018.

[18] J. Liu and S. J. Wright, "An accelerated randomized Kaczmarz algorithm," *Math. Comput.,* vol. 85, no. 297, pp. 153-178, 2016.

[19] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.,* vol. 140, no. 1, pp. 125-161, 2013.

[20] M. S. Morshed, M. S. Islam, and M. Noor-E-Alam, "Accelerated sampling Kaczmarz Motzkin algorithm for linear feasibility problem," *J. Global Optim.,* vol. 77, no. 2, pp. 1-22, 2019.

[21] J. L. Yan and L. H. Zheng, "A class of momentum-preserving fourier pseudo-spectral schemes for the korteweg-de vries equation," *IAENG International Journal of Applied Mathematics,* vol. 49, no. 4, pp 548-560, 2019.

[22] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in Proceedings of the International Conference on Machine Learning, vol. 28, pp. 1139-1147, 2013.

[23] F. S. Chen, L. X. Shen, B. W. Suter, and Y. S. Xu, "Nesterov's algorithm solving dual formulation for compressed sensing," *J. Comput. Appl. Math.,* vol. 260, pp. 1-17, 2014.

[24] L. Tang, Y. F. Fan, F. P. Ye, and W. Z. Lu, "Estimation of IMU orientation using Nesterov's accelerated gradient improved by fuzzy control rule," *Sensor. Actuat. A-Phys.,* vol. 332, no. 1, pp. 113062, 2021.

[25] X. Xiang, X. Liu, W. Tan, and X. Dai, "An accelerated randomized extended Kaczmarz algorithm," *J. Phys.: Conf. Ser.,* vol. 814, pp. 012017, 2017.

[26] T. A. Davis and Y. Hu, "The university of florida sparse matrix collection," *ACM Trans. Math. Softw.,* vol. 38, no. 1, pp. 1-25, 2011.

[27] Z. D. Bai and Y. Q. Yin, "Limit of the smallest eigenvalue of a large dimensional sample covariance matrix," *Ann. Probab.,* vol. 21, no. 3, pp. 1275-1294, 1993.

[28] D. Jonsson, "Some limit theorems for the eigenvalues of a sample covariance matrix," *J. Multivariate Anal.,* vol. 12, no. 1, pp. 1-38, 1982.

[29] K. W. Wachter, "The strong limits of random matrix spectra for sample matrices of independent elements," *Ann. Probab.,* vol. 6, no. 1, pp. 1-18, 1978.

[30] Y. Q. Yin, "Limiting spectral distribution for a class of random matrices," *J. Multivariate Anal.,* vol. 20, no. 1, pp. 50-68, 1986.