TAIWANESE JOURNAL OF MATHEMATICS
Vol. 26, No. 5, pp. 953–979, October 2022
DOI: 10.11650/tjm/220305

# On Relaxed Greedy Randomized Iterative Methods for the Solution of Factorized Linear Systems

Shi-Min Liu and Yong Liu*

Abstract. RK-RK and REK-RK methods are the two latest while very effective randomized iteration solvers for factorized linear system $UV\beta = y$ by interlacing randomized Kaczmarz (RK) and randomized extended Kaczmarz (REK) updates. This paper considers two latest randomized iterative methods for solving large-scale linear systems and linear least-squares problems—greedy randomized Kaczmarz (GRK) method and greedy randomized Gauss–Seidel (GRGS) method. By introducing a relaxation parameter $\omega$ into the iterates of GRK and GRGS, we construct relaxed GRK and GRGS methods, respectively. In addition, by interlacing their updates, we propose relaxed GRK-GRK and GRGS-GRK methods to solve consistent and inconsistent factorized linear systems, respectively. We prove the exponential convergence of these two interlaced methods and show that relaxed GRK-GRK and GRGS-GRK can be more efficient than RK-RK and REK-RK, respectively, if the relaxation parameters are chosen appropriately.

## 1. Introduction

We consider an iterative solution of systems of linear equations of the form

$$UV\beta = y \quad \text{with } U \in \mathbb{R}^{m\times k} \text{ and } V \in \mathbb{R}^{k\times n}, \tag{1.1}$$

where $y \in \mathbb{R}^m$ is an $m$-dimensional vector, $\beta \in \mathbb{R}^n$ is the $n$-dimensional unknown vector and we refer to this system as factorized linear system. Such a factorization of large and low-rank system arise naturally in many applications for some reasons, such as algorithmic choices [7, 17, 18], systems infrastructure constraints [19, 23, 32], and statistical motivation [11]. Recently, Ma, Needell and Ramdas [25] proposed two different stochastic iterative methods to solve (1.1) without needing to solve the full system

$$X\beta = y \quad \text{with } X = UV \in \mathbb{R}^{m\times n}. \tag{1.2}$$

Received June 26, 2021; Accepted March 27, 2022.
Communicated by Eric Chung.
2020 *Mathematics Subject Classification.* 65F10, 65F20, 65K05, 90C25, 15A06.
*Key words and phrases.* factorized linear systems, greedy randomized Kaczmarz, greedy randomized Gauss–Seidel, relaxation parameter.
The work is supported by National Natural Science Foundation (11371243).
*Corresponding author.

Both the stochastic iterative methods proposed in [25] for solving (1.1) utilize iterates of RK method [31] and (or) REK method [33]. More specifically, when the system (1.1) is consistent, in [25] Ma, Needell and Ramdas proposed the so-called RK-RK method, which uses an iterate of RK on the following subsystem

$$Ux = y, \tag{1.3}$$

intertwined again with an iterate of RK to solve subsystem

$$Vb = x, \tag{1.4}$$

and converges to the unique least-norm solution to (1.1). When the system (1.1) is inconsistent, they proposed using REK to solve (1.3) followed by RK to solve (1.4), and then obtained the so-called REK-RK method, which converges to the ordinary least-squares solution to (1.1). Admittedly, with numerical experiments it was shown in [25] that the RK-RK method and the REK-RK method can provide significant computational advantages for factorized linear systems over applying the RK or the REK on the full system (1.2) naively.

### 1.1. Motivation and contribution

Recently, Bai and Wu proposed the greedy randomized Kaczmarz (GRK) [3] and greedy randomized Gauss–Seidel (GRGS) [5] methods for solving large-scale consistent linear systems and linear least-squares problems. Both the GRK and GRGS converge faster than the RK and RGS (i.e., randomized coordinate descent [20, 30]) in both theory and experiments, respectively, because these two greedy methods introduce a practical and appropriate probability criterion used to select the working rows (or columns) from the coefficient matrix. Therefore, we can use GRK (or GRGS) instead of RK (or REK) to accelerate the convergence of the RK-RK and REK-RK methods.

The main contribution of our paper is to construct relaxed GRK (GRK($\omega$) for short) and GRGS (GRGS($\omega$) for short) methods (because the convergence of the Kaczmarz method can be accelerated by introducing relaxation [31] (see also [8, 12])) by introducing a relaxation parameter in their updates rather than in probability criterion discussed in [4] for GRK method, respectively. Based on this, we interlace iterates of the GRK($\omega$) to solve the subsystems (1.3) and (1.4) and find the least-norm solution of the system (1.1) when it is consistent, and use GRGS($\omega$) to solve (1.3) intertwined with an iterate of GRK($\omega$) to solve (1.4) and seek the least-squares solution of the system (1.1) when it is inconsistent. In theory, we prove the convergence of both GRK($\omega$) and GRGS($\omega$) with $\omega \in (0, 2)$ for consistent system and overdetermined system with full column rank, respectively. In addition, we also provide a proof that shows linear convergence in expectation to the (least-squares or least-norm) solution of (inconsistent or consistent) factorized linear systems for both intertwined methods (the central methods of this paper). And in computations we show that our methods significantly outperform the methods proposed in [25] for factorized systems in terms of both iteration counts and computing times.

### 1.2. Notations

Throughout this paper, for a matrix $X = (x_{ij}) \in \mathbb{R}^{m\times n}$, we use $X^{(i)}$, $X_{(j)}$, $X^T$, $\|X\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |x_{ij}|^2}$, $\sigma_{\min}(X)$ and $\mathcal{R}(X)$ to represent its $i$th row, $j$th column, transpose, Frobenius norm, smallest nonzero singular value and column space, respectively. For any vector $y \in \mathbb{R}^m$, $y^{(i)}$ and $y^T$ represent its $i$th entry and transpose, respectively. The inner product in $\mathbb{R}^n$ is represented by $\langle \cdot , \cdot \rangle$, $\|\cdot\|_2$ is used to denote the correspondingly induced Euclidean norm of either a vector or a matrix.

In addition, we use $\mathbb{E}_k$ to denote the expected value conditional on the first $k$ iterations, i.e.,

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

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

### 1.3. Paper outline

The organization of this paper is as follows. In Section 2 we describe the RK-RK and REK-RK methods for factorized linear system (1.1). In Section 3 we propose our intertwined methods for factorized linear system (1.1) by using the relaxed GRK and GRGS and establish their convergence rates. The numerical results are reported in Section 4. Finally, in Section 5, we end the paper with a few conclusions.

## 2. The RK-RK and REK-RK methods for factorized linear system (1.1)

In this section, we first introduce the randomized Kaczmarz method [31] and its extension, i.e., randomized extended Kaczmarz method [33], and then briefly describe the RK-RK and REK-RK methods [25] for solving factorized linear system (1.1).

### 2.1. RK and REK methods

In 2009, Strohmer and Vershynin [31] proposed the RK method for solving consistent systems of linear equations $X\beta = y$, where $X \in \mathbb{R}^{m\times n}$ $(m \geq n)$ is of full column rank and $y \in \mathbb{R}^m$, with expected exponential rate of convergence by using the rows of the coefficient matrix $X$ randomly, which can greatly improve the convergence of the original Kaczmarz method [15] sequentially cycling through the rows of $X$. Formally, the RK method for linear system $X\beta = y$ takes the following form

$$\beta_{t+1} = \beta_t + \frac{y^{(i_t)} - X^{(i_t)}\beta_t}{\|X^{(i_t)}\|_2^2}(X^{(i_t)})^T, \quad t = 0, 1, 2, \dots, \tag{2.1}$$

where the target row numbered as $i_t$ is selected according to the probability criterion $\Pr(\text{row} = i_t) = \frac{\|X^{(i_t)}\|_2^2}{\|X\|_F^2}$, and $\beta_0 \in \mathcal{R}(X^T)$.

The convergence of the RK method was first proved in [31] when $m \geq n$ and the coefficient matrix $X$ is of full column rank. Also when $m < n$, in [24], Ma, Needell and Ramdas gave the same convergence rate of the RK method. Recently, Bai and Wu proposed a more precise convergence rate for the RK method in [2]. In addition, from the update (2.1), it can be seen that the RK method is especially suitable for parallel computations and large-scale problems since each step only requires one row of the matrix $X$ and no matrix-vector products; see [1] for additional references.

For inconsistent systems, the RK method does not converge to the least-squares solution as one might desire [26], and to remedy this, Zouzias and Freris [33] proposed the REK method to solve linear systems in all settings. It can be thought of as a randomized variant of Popa's extended Kaczmarz method [28, 29]. More precisely, the REK update rule takes the following form

$$\beta_{t+1} = \beta_t + \frac{y^{(i_t)} - z_t^{(i_t)} - X^{(i_t)}\beta_t}{\|X^{(i_t)}\|_2^2}(X^{(i_t)})^T, \quad z_{t+1} = z_t - \frac{X_{(j_t)}^T z_t}{\|X_{(j_t)}\|_2^2} X_{(j_t)}$$

with initial guesses $\beta_0 = 0$ and $z_0 = y$. Here, row $i_t \in \{1, 2, \dots, m\}$ and column $j_t \in \{1, 2, \dots, n\}$ of $X$ are selected at random with probability

$$\Pr(\text{row} = i_t) = \frac{\|X^{(i_t)}\|_2^2}{\|X\|_F^2} \quad \text{and} \quad \Pr(\text{column} = j_t) = \frac{\|X_{(j_t)}\|_2^2}{\|X\|_F^2}.$$

In the overdetermined inconsistent setting, $z_k$ approximates the component of $y$ which is orthogonal to the range of matrix $X$, allowing for the iterates $\beta_k$ to converge linearly in expectation to the true least-squares solution to the system $X\beta = y$. Recently, Bai and Wu proposed a partially REK (PREK) method in [6], showing that the estimated upper bound for the expected solution error of the PREK method is much smaller than that of the REK method when the coefficient matrix is tall (i.e., $m \geq n$) and of full column rank and all columns of the coefficient matrix are mutually orthonormal. For more studies on the REK method, we refer to [9, 10, 27] and the references therein.

## 2.2. RK-RK and REK-RK methods

The main idea of the RK-RK method [25] for solving the factorized linear system (1.1) is first to apply the RK to the linear system (1.3), after obtaining an iterative solution $x_t$

at the $t$th iterate, then reuses an iterate of the RK on the linear system $Vb = x_t$. As a result, an approximate solution $b_t$ to the factorized linear system (1.1) can be obtained. We list the pseudo-code of the RK-RK method for factorized linear system (1.1) when it is consistent in Method 2.1.

**Method 2.1** The RK-RK Method for Factorized Linear System (1.1)

**Input:** $U$, $V$, $y$ and $x_0 = 0$.

**Output:** $b_\ell$.

1: **for** $t = 0, 1, 2, \ldots, \ell - 1$ **do**

2: Select $i_t \in \{1, 2, \ldots, m\}$ with probability $\Pr(\text{row} = i_t) = \frac{\|U^{(i_t)}\|_2^2}{\|U\|_F^2}$

3: Set $x_{t+1} = x_t + \frac{y^{(i_t)} - U^{(i_t)} x_t}{\|U^{(i_t)}\|_2^2} (U^{(i_t)})^T$

4: Select $j_t \in \{1, 2, \ldots, k\}$ with probability $\Pr(\text{row} = j_t) = \frac{\|V^{(j_t)}\|_2^2}{\|V\|_F^2}$

5: Set $b_{t+1} = b_t + \frac{x_{t+1}^{(j_t)} - V^{(j_t)} b_t}{\|V^{(j_t)}\|_2^2} (V^{(j_t)})^T$

6: **end for**

| Case | k | $X$, $U$, $V$ | Method [25] | Our method |
|---|---|---|---|---|
| Case I | $k < \min\{m, n\}$ | $X$ = Under<br>$U$ = Over, Consis<br>$V$ = Under | RK-RK | GRK($\omega$)-GRK($\alpha$) |
| Case II | $k < \min\{m, n\}$ | $X$ = Over, Consis<br>$U$ = Over, Consis<br>$V$ = Under | RK-RK | GRK($\omega$)-GRK($\alpha$) |
| Case III | $\min\{m, n\} < k < \max\{m, n\}$ | $X$ = Over, Consis<br>$U$ = Over, Consis<br>$V$ = Over, Consis | RK-RK | GRK($\omega$)-GRK($\alpha$) |
| Case IV | $k < \min\{m, n\}$ | $X$ = Over, Incon<br>$U$ = Over, Incon<br>$V$ = Under | REK-RK | GRGS($\omega$)-GRK($\alpha$) |

Table 2.1: Summary of optimal methods proposed in [25] for solving system (1.1) for four different types of matrices $U$ and $V$ for given $m$, $n$ and $k$ relations.

In fact, only when the system (1.3) is overdetermined consistent and the system (1.4) is underdetermined or overdetermined consistent can the iteration sequence $\{b_t\}_{t=0}^{\infty}$ generated by the RK-RK method converges to the least-norm solution of the system (1.1) [25, Section 3]. In [25, Section 3], the authors systematically discussed how the settings of (1.3) and (1.4) are determined by $X$ and when we can expect the RK-RK and REK-RK methods will be able to solve the system (1.1) utilizing the subsystems (1.3) and (1.4).

For the reader's convenience, we list all four cases in which the RK-RK and REK-RK methods have the potential to solve the system (1.1) in Table 2.1, and for simplicity, we will refer to the matrix $U$ of a linear system as consistent or inconsistent when the system itself is consistent or inconsistent.

In Table 2.1, 'Under', 'Over', 'Consis' and 'Incon' are abbreviations of underdetermined, overdetermined, consistent and inconsistent, respectively. From this table, we know that when $X$ is inconsistent, one can use REK to solve the subsystem (1.3) followed by RK to solve the subsystem (1.4) and then obtain the so-called REK-RK method [25]. We omit the pseudo-code of REK-RK method. In [25], the authors gave the convergence properties of these two methods, and this result is precisely restated below.

**Theorem 2.1.** *Let $X$ be low rank, $X = UV$ such that $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{k\times n}$ are full rank, and the systems $X\beta = y$, $Ux = y$ and $Vb = x$ have optimal solutions[1] $\beta_*$, $x_*$ and $b_*$ respectively. Setting $x_0 = b_0 = 0$ and assuming $k < m, n$. If $X\beta = y$ is consistent, then $b_* = \beta_*$ and RK-RK converges with expected error*

$$\mathbb{E}\|b_t - \beta_*\|_2^2 \le \left(1 - \frac{\sigma_{\min}^2(V)}{\|V\|_2^2}\right)^t \|b_*\|_2^2 + \frac{1}{\sigma_{\min}^2(V)}\left(1 - \frac{\sigma_{\min}^2(U)}{\|U\|_2^2}\right)^t \|x_*\|_2^2,$$

*and if $X\beta = y$ is inconsistent, then $b_* = \beta_*$ and REK-RK converges with expected error*

$$\mathbb{E}\|b_t - \beta_*\|_2^2 \le \left(1 - \frac{\sigma_{\min}^2(V)}{\|V\|_2^2}\right)^t \|b_*\|_2^2 + \frac{1}{\sigma_{\min}^2(V)}\left(1 - \frac{\sigma_{\min}^2(U)}{\|U\|_2^2}\right)^{\lfloor t/2\rfloor} (1 + 2\kappa_U^2)\|x_*\|_2^2,$$

*where $\kappa_U^2$ denotes the squared condition number of matrix $U$.*

## 3. The GRK($\omega$)-GRK($\alpha$) and GRGS($\omega$)-GRK($\alpha$) methods for factorized linear system (1.1)

### 3.1. Relaxed GRK and GRGS methods

In this subsection, we further generalize the greedy randomized Kaczmarz (GRK) method [3] and the greedy randomized Gauss–Seidel (GRGS) method [5] by introducing a relaxation parameter $\omega \in (0, 2)$ in their iterative updates, and obtain a class of relaxed GRK (GRK($\omega$) for short) methods and GRGS (GRGS($\omega$) for short) methods. Let $\omega \in (0, 2)$ be an arbitrary parameter and $\ell$ be a prescribed positive integer. Then the GRK($\omega$) method for linear system $X\beta = y$ can be algorithmically described in Method 3.1.

[1]The optimal solution to a system is either the least-norm, unique, or least-squares solution depending on whether the system is underdetermined, overdetermined consistent, or overdetermined inconsistent, respectively.

**Method 3.1** The GRK($\omega$) Method

**Input:** $X$, $y$, $\ell$, $\omega \in (0,2)$ and $\beta_0$.

**Output:** $\beta_\ell$.

1: **for** $t = 0, 1, 2, \ldots, \ell - 1$ **do**

2: Compute

$$\epsilon_t = \frac{1}{2}\left(\frac{1}{\|y - X\beta_t\|_2^2}\max_{1\le i_t\le m}\left\{\frac{\left|y^{(i_t)} - X^{(i_t)}\beta_t\right|^2}{\|X^{(i_t)}\|_2^2}\right\} + \frac{1}{\|X\|_F^2}\right)$$

3: Determine the index set of positive integers

$$\mathcal{U}_t = \left\{i_t \mid \left|y^{(i_t)} - X^{(i_t)}\beta_t\right|^2 \ge \epsilon_t\|y - X\beta_t\|_2^2\|X^{(i_t)}\|_2^2\right\}$$

4: Compute the $i$th entry $\widetilde{r}_t^{(i)}$ of the vector $\widetilde{r}_t$ according to

$$\widetilde{r}_t^{(i)} = \begin{cases} y^{(i)} - X^{(i)}\beta_t & \text{if } i \in \mathcal{U}_t, \\ 0 & \text{otherwise} \end{cases}$$

5: Select $i_t \in \mathcal{U}_t$ with probability $\Pr(\text{row} = i_t) = \frac{\left|\widetilde{r}_t^{(i_t)}\right|^2}{\|\widetilde{r}_t\|_2^2}$

6: Set $\beta_{t+1} = \beta_t + \omega\frac{y^{(i_t)} - X^{(i_t)}\beta_t}{\|X^{(i_t)}\|_2^2}(X^{(i_t)})^*$

7: **end for**

We remark that the GRK($\omega$) method is different from the relaxed greedy randomized Kaczmarz (RGRK) method discussed by Bai and Wu in [4], which introduces a relaxation parameter $\theta \in [0,1]$ in the probability criterion of GRK as follows:

$$\epsilon_t = \frac{\theta}{\|y - X\beta_t\|_2^2}\max_{1\le i_t\le m}\left\{\frac{\left|y^{(i_t)} - X^{(i_t)}\beta_t\right|^2}{\|X^{(i_t)}\|_2^2}\right\} + \frac{1-\theta}{\|X\|_F^2}.$$

In addition, it is easy to find that the main difference between GRK and GRK($\omega$) is the introduction of a relaxation parameter $\omega \in (0,2)$ in the iterative update in Step 6. When $\omega = 1$, the GRK($\omega$) method automatically reduces to the GRK method. Moreover, whatever the parameter $\omega$ is chosen, both GRK and GRK($\omega$) have exactly the same computational complexity at the $t$th iterate.

Inspired by the construction method of GRK, Bai and Wu [5] proposed the greedy randomized Gauss–Seidel (GRGS) method by introducing an effective probability criterion for selecting the working columns from the coefficient matrix of overdetermined linear system $X\beta = y$, where $X \in \mathbb{R}^{m\times n}$ is a rectangular matrix of full column rank. Therefore, let $e_j$ be the column vector with 1 in the $j$th position and 0 elsewhere, then we can give the following GRGS($\omega$) method directly.

**Method 3.2** The GRGS($\omega$) Method

**Input:** $X$, $y$, $\ell$, $\omega \in (0,2)$ and $\beta_0$.

**Output:** $\beta_\ell$.

**for** $t = 0, 1, 2, \ldots, \ell - 1$ **do**

Compute $r_t = y - X\beta_t$ and

$$\delta_t = \frac{1}{2}\left(\frac{1}{\|X^T r_t\|_2^2}\max_{1\le j_t\le n}\left\{\frac{\left|X_{(j_t)}^T r_t\right|^2}{\|X_{(j_t)}\|_2^2}\right\} + \frac{1}{\|X\|_F^2}\right)$$

Determine the index set of positive integers

$$\mathcal{V}_t = \left\{ j_t \;\middle|\; \left|X_{(j_t)}^T r_t\right|^2 \ge \delta_t \|X^T r_t\|_2^2 \|X_{(j_t)}\|_2^2 \right\}$$

Let $s_t = X^T r_t$ and compute the $j$th entry $\widetilde{s}_t^{(j)}$ of the vector $\widetilde{s}_t$ according to

$$\widetilde{s}_t^{(j)} = \begin{cases} s_t^{(j)} & \text{if } j \in \mathcal{V}_t, \\ 0 & \text{otherwise} \end{cases}$$

Select $j_t \in \mathcal{V}_t$ with probability $\Pr(\text{column} = j_t) = \frac{\left|\widetilde{s}_t^{(j_t)}\right|^2}{\|\widetilde{s}_t\|_2^2}$

Set $\beta_{t+1} = \beta_t + \omega \frac{s_t^{(j_t)}}{\|X_{(j_t)}\|_2^2} e_{j_t}$

**end for**

For the convergence properties of the GRK($\omega$) and GRGS($\omega$) methods with $\omega$ being a given positive constant in the interval $(0,2)$, we can establish the following theorem.

**Theorem 3.1.** *Let the linear system $X\beta = y$, with the coefficient matrix $X \in \mathbb{R}^{m\times n}$ and the right-hand side $y \in \mathbb{R}^m$, be consistent. Then the iteration sequence $\{\beta_t\}_{t=0}^{\infty}$, generated by the GRK($\omega$) method with $\omega \in (0,2)$ starting from any initial guess $\beta_0 \in \mathcal{R}(X^T)$, converges to the unique least-norm solution $\beta_*$ of the system $X\beta = y$ with expected error*

$$\mathbb{E}\|\beta_{t+1} - \beta_*\|_2^2 \le \xi(\omega, X)^{t+1}\|\beta_0 - \beta_*\|_2^2, \tag{3.1}$$

*where $\xi(\omega, X) = 1 - \omega(2-\omega)\frac{\sigma_{\min}^2(X)}{\|X\|_F^2}$. If the coefficient matrix $X$ is of full column rank and overdetermined, then the iteration sequence $\{\beta_t\}_{t=0}^{\infty}$, generated by the GRGS($\omega$) method with $\omega \in (0,2)$ starting from any initial guess $\beta_0 \in \mathbb{R}^n$, converges to the unique least-*

*squares solution* $\beta_*$ *of the system* $X\beta = y$ *with expected error*

$$\mathbb{E}\|\beta_{t+1} - \beta_*\|^2_{X^TX} \leq \xi(\omega, X)^{t+1}\|\beta_0 - \beta_*\|^2_{X^TX}, \tag{3.2}$$

*where* $\|u\|^2_{X^TX} = \|Xu\|^2_2$ *is the norm induced by Hermitian positive definite matrix* $X^TX$.

*Proof.* Let $P_{i_t} = \frac{(X^{(i_t)})^T X^{(i_t)}}{\|X^{(i_t)}\|_2^2}$, $i_t \in \{1, 2, \ldots, m\}$, and $Q_{j_t} = \frac{X_{(j_t)} X^T_{(j_t)}}{\|X_{(j_t)}\|_2^2}$, $j_t \in \{1, 2, \ldots, n\}$. Obviously, it holds that

$$P^T_{i_t} P_{i_t} = \frac{(X^{(i_t)})^T}{\|X^{(i_t)}\|_2^2} \cdot \frac{X^{(i_t)}(X^{(i_t)})^T}{\|X^{(i_t)}\|_2^2} \cdot X^{(i_t)} = P_{i_t} \tag{3.3}$$

and

$$Q^T_{j_t} Q_{j_t} = \frac{X_{(j_t)}}{\|X_{(j_t)}\|_2^2} \cdot \frac{X^T_{(j_t)} X_{(j_t)}}{\|X_{(j_t)}\|_2^2} \cdot X^T_{(j_t)} = Q_{j_t}. \tag{3.4}$$

We summarize the main procedures of this proof as follows:

$$\begin{aligned}
\text{GRK}(\omega): \quad & \mathbb{E}_t\big(\|\beta_{t+1} - \beta_*\|_2^2\big) \\
&= \mathbb{E}_t\big(\|\beta_t - \beta_* - \omega P_{i_t}(\beta_t - \beta_*)\|_2^2\big) \\
&\overset{\text{(i)}}{=} \mathbb{E}_t\big(\|\beta_t - \beta_*\|_2^2 - 2\omega(\beta_t - \beta_*)^T P_{i_t}(\beta_t - \beta_*) + \omega^2(\beta_t - \beta_*)^T P^T_{i_t} P_{i_t}(\beta_t - \beta_*)\big) \\
&\overset{\text{(ii)}}{=} \mathbb{E}_t\big(\|\beta_t - \beta_*\|_2^2 - \omega(2-\omega)\|P_{i_t}(\beta_t - \beta_*)\|_2^2\big) \\
&= \|\beta_t - \beta_*\|_2^2 - \omega(2-\omega)\mathbb{E}_t\big(\|P_{i_t}(\beta_t - \beta_*)\|_2^2\big) \\
&= \|\beta_t - \beta_*\|_2^2 - \omega(2-\omega)\mathbb{E}_t\left(\frac{|X^{(i_t)}(\beta_t - \beta_*)|^2}{\|X^{(i_t)}\|_2^2}\right) \\
&= \|\beta_t - \beta_*\|_2^2 - \omega(2-\omega)\sum_{i_t \in \mathcal{U}_t} \frac{\left|y^{(i_t)} - X^{(i_t)}\beta_t\right|^2}{\sum_{i \in \mathcal{U}_t}\left|y^{(i)} - X^{(i)}\beta_t\right|^2}\left(\frac{|X^{(i_t)}(\beta_t - \beta_*)|^2}{\|X^{(i_t)}\|_2^2}\right) \\
&\overset{\text{(iii)}}{\leq} \|\beta_t - \beta_*\|_2^2 - \omega(2-\omega)\epsilon_t\|X(\beta_t - \beta_*)\|_2^2 \\
&\overset{\text{(iv)}}{\leq} \left(1 - \omega(2-\omega)\frac{\sigma^2_{\min}(X)}{\|X\|_F^2}\right)\|\beta_t - \beta_*\|_2^2
\end{aligned}$$

and

$$\begin{aligned}
\text{GRGS}(\omega): \quad & \mathbb{E}_t\big(\|\beta_{t+1} - \beta_*\|^2_{X^TX}\big) \\
&= \mathbb{E}_t\big(\|X(\beta_t - \beta_*) - \omega Q_{j_t} X(\beta_t - \beta_*)\|_2^2\big) \\
&\overset{\text{(i}')}{=} \mathbb{E}_t\big(\|X(\beta_t - \beta_*)\|_2^2 - 2\omega(\beta_t - \beta_*)^T X^T Q_{j_t} X(\beta_t - \beta_*) \\
&\qquad + \omega^2(\beta_t - \beta_*)^T X^T Q^T_{j_t} Q_{j_t} X(\beta_t - \beta_*)\big) \\
&\overset{\text{(ii}')}{=} \mathbb{E}_t\big(\|X(\beta_t - \beta_*)\|_2^2 - \omega(2-\omega)\|Q_{j_t} X(\beta_t - \beta_*)\|_2^2\big)
\end{aligned}$$

$$
\begin{aligned}
&= \|X(\beta_t-\beta_*)\|_2^2 - \omega(2-\omega)\mathbb{E}_t\big(\|Q_{i_t}X(\beta_t-\beta_*)\|_2^2\big)\\
&= \|X(\beta_t-\beta_*)\|_2^2 - \omega(2-\omega)\mathbb{E}_t\left(\frac{|X_{(j_t)}^T X(\beta_t-\beta_*)|^2}{\|X_{(j_t)}\|_2^2}\right)\\
&= \|X(\beta_t-\beta_*)\|_2^2\\
&\quad - \omega(2-\omega)\sum_{j_t\in\mathcal{V}_t}\frac{\left|X_{(j_t)}^T(y-X\beta_t)\right|^2}{\sum_{j\in\mathcal{V}_t}\left|X_{(j)}^T(y-X\beta_t)\right|^2}\left(\frac{\left|X_{(j_t)}^T(y-X\beta_t)\right|^2}{\|X^{(j_t)}\|_2^2}\right)\\
&\overset{(\mathrm{iii}')}{\le} \|X(\beta_t-\beta_*)\|_2^2 - \omega(2-\omega)\delta_t\|X^TX(\beta_t-\beta_*)\|_2^2\\
&\overset{(\mathrm{iv}')}{\le} \left(1-\omega(2-\omega)\frac{\sigma_{\min}^2(X)}{\|X\|_F^2}\right)\|\beta_t-\beta_*\|_{X^TX}^2.
\end{aligned}
$$

Here, using the relation $\langle u,u\rangle = \|u\|_2^2$, for any $u\in\mathbb{R}^n$, then by straightforwardly computations we have (i) and (i′), in addition, by making use of (3.3) and (3.4), we can get (ii) and (ii′), respectively. The inequalities (iii) and (iii′) are achieved with the use of the definitions of $\mathcal{U}_t$ and $\mathcal{V}_t$, which lead to the following two inequalities

$$
\left|y^{(i_t)}-X^{(i_t)}\beta_t\right|^2 \ge \epsilon_t\|y-X\beta_t\|_2^2\|X^{(i_t)}\|_2^2 = \epsilon_t\|X(\beta_t-\beta_*)\|_2^2\|X^{(i_t)}\|_2^2, \quad \forall\, i_t\in\mathcal{U}_t
$$

and

$$
\begin{aligned}
\left|X_{(j_t)}^T(y-X\beta_t)\right|^2 &\ge \delta_t\|X^T(y-X\beta_t)\|_2^2\|X_{(j_t)}\|_2^2\\
&= \delta_t\|X^T(y-X\beta_*)+X^T(X\beta_*-X\beta_t)\|_2^2\|X_{(j_t)}\|_2^2\\
&= \delta_t\|X^TX(\beta_*-\beta_t)\|_2^2\|X_{(j_t)}\|_2^2, \quad \forall\, j_t\in\mathcal{V}_t.
\end{aligned}
$$

The last inequalities (iv) and (iv′) are achieved by using of the estimate

$$
\|Xu\|_2^2 \ge \sigma_{\min}^2(X)\|u\|_2^2, \tag{3.5}
$$

which holds true for any $u\in\mathbb{R}^n$ belonging to the column space of $X^T$ (see the work of [5, 25]) and the following facts that

$$
\epsilon_t = \frac{\max_{1\le i_t\le m}\left(\frac{|y^{(i_t)}-X^{(i_t)}\beta_t|^2}{\|X^{(i_t)}\|_2^2}\right)}{2\|X\|_F^2\sum_{i_t=1}^m\frac{\|X^{(i_t)}\|_2^2}{\|X\|_F^2}\frac{|y^{(i_t)}-X^{(i_t)}\beta_t|^2}{\|X^{(i_t)}\|_2^2}} + \frac{1}{2\|X\|_F^2} \ge \frac{1}{\|X\|_F^2}, \quad t=0,1,2,\dots \tag{3.6}
$$

and

$$
\delta_t = \frac{\max_{1\le j_t\le n}\left(\frac{|X_{(j_t)}^T(y-X\beta_t)|^2}{\|X_{(j_t)}\|_2^2}\right)}{2\|X\|_F^2\sum_{j_t=1}^n\frac{\|X_{(j_t)}\|_2^2}{\|X\|_F^2}\frac{|X_{(j_t)}^T(y-X\beta_t)|^2}{\|X_{(j_t)}\|_2^2}} + \frac{1}{2\|X\|_F^2} \ge \frac{1}{\|X\|_F^2}, \quad t=0,1,2,\dots.
$$

Furthermore, by taking full expectation on both sides of (iv) and (iv′), respectively, from $\mathbb{E}[\mathbb{E}_t[\,\cdot\,]] = \mathbb{E}[\,\cdot\,]$ we see that

$$\mathbb{E}\|\beta_{t+1} - \beta_*\|_2^2 \leq \left(1 - \omega(2-\omega)\frac{\sigma_{\min}^2(X)}{\|X\|_F^2}\right)\mathbb{E}\|\beta_t - \beta_*\|_2^2 \tag{3.7}$$

and

$$\mathbb{E}\|\beta_{t+1} - \beta_*\|_{X^TX}^2 \leq \left(1 - \omega(2-\omega)\frac{\sigma_{\min}^2(X)}{\|X\|_F^2}\right)\mathbb{E}\|\beta_t - \beta_*\|_{X^TX}^2. \tag{3.8}$$

Then, applying both bounds (3.7) and (3.8) recursively, we obtain the theorem. □

We finally remark that the GRK($\omega$) method can significantly outperform the GRK method in experiments if an appropriate relaxation parameter $\omega$ is available, and this can be verified experimentally in [22], in which Liu and Gu applied the GRK($\omega$) method to ridge regression, and achieved significant computational advantages for this problem.

### 3.2. GRK($\omega$)-GRK($\alpha$) and GRGS($\omega$)-GRK($\alpha$) methods

Similar to the construction methods of RK-RK and REK-RK, in this section, we construct our methods (the central algorithms of the paper) by intertwining two relaxed iterative methods discussed in Section 3 to solve the subsystem (1.3) followed by the subsystem (1.4). For the consistent setting, we propose Method 3.3, which uses an iterate of GRK($\omega$) with $\omega \in (0, 2)$ on (1.3) intertwined with an iterate of GRK($\alpha$) with $\alpha \in [1, 3/2)$ on (1.4). For the inconsistent setting, as shown in Method 3.4, we use GRGS($\omega$) with $\omega \in (0, 2)$ to solve (1.3) followed by GRK($\alpha$) with $\alpha \in [1, 3/2)$ to solve (1.4).

**Method 3.3** The GRK($\omega$)-GRK($\alpha$) Method for Factorized Linear System (1.1)

**Input:** $U$, $V$, $y$, $\omega \in (0, 2)$, $\alpha \in [1, 3/2)$ and $x_0 = b_0 = 0$.
**Output:** $b_\ell$.

1: **for** $k = 0, 1, 2, \ldots, \ell - 1$ **do**
2: Do Steps 2–5 of Method 3.1 by inputting variables $U$, $y$ and initial guess $x_0 = 0$
3: Set $x_{t+1} = x_t + \omega\frac{y^{(i_t)} - U^{(i_t)}x_t}{\|U^{(i_t)}\|_2^2}(U^{(i_t)})^T$
4: Do Steps 2–5 of Method 3.1 by inputting variables $V$, $x_{t+1}$ and initial guess $b_0 = 0$
5: Set $b_{t+1} = b_t + \alpha\frac{x_{t+1}^{(j_t)} - V^{(j_t)}b_t}{\|V^{(j_t)}\|_2^2}(V^{(j_t)})^T$
6: **end for**

The main reason for selecting parameter $\alpha$ in the interval $[1, 3/2)$ is to ensure that these two methods can converge to the desired solution. For a detailed explanation, please refer to the following proof of Theorem 3.2.

**Method 3.4** The GRGS($\omega$)-GRK($\alpha$) Method for Factorized Linear System (1.1)

**Input:** $U$, $V$, $y$, $\omega \in (0,2)$, $\alpha \in [1,3/2)$ and $x_0 = b_0 = 0$.

**Output:** $b_\ell$.

1: **for** $k = 0,1,2,\dots,\ell-1$ **do**
2: Do Steps 2–5 of Method 3.2 by inputting variables $U$, $y$ and initial guess $x_0 = 0$
3: Set $x_{t+1} = x_t + \omega \frac{s_t^{(j_t)}}{\|U_{(j_t)}\|_2^2} e_{j_t}$
4: Do Steps 2–5 of Method 3.1 by inputting variables $V$, $x_{t+1}$ and initial guess $b_0 = 0$
5: Set $b_{t+1} = b_t + \alpha \frac{x_{t+1}^{(j_t)} - V^{(j_t)} b_t}{\|V^{(j_t)}\|_2^2} (V^{(j_t)})^T$
6: **end for**

**Theorem 3.2.** *Let $X$ be low rank, $X = UV$ such that $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{k\times n}$ are full rank, and the systems $X\beta = y$, $Ux = y$ and $Vb = x$ have optimal solutions $\beta_*$, $x_*$ and $b_*$ respectively. Setting $x_0 = b_0 = 0$ and assuming $k < m,n$. If $X\beta = y$ is consistent, then $b_* = \beta_*$ and GRK($\omega$)-GRK($\alpha$) converges with expected error*

$$\mathbb{E}\|b_{t+1} - b_*\|_2^2 \le \zeta(\alpha, V)^{t+1}\|b_*\|_2^2 + \frac{2\alpha-1}{3-2\alpha}\frac{\|V\|_F^2}{\gamma\sigma_{\min}^2(V)}\xi(\omega,U)^{t+1}\|x_*\|_2^2,$$

*if $X\beta = y$ is inconsistent, then $b_* = \beta_*$ and GRGS($\omega$)-GRK($\alpha$) converges with expected error*

$$\mathbb{E}\|b_{t+1} - b_*\|_2^2 \le \zeta(\alpha, V)^{t+1}\|b_*\|_2^2 + \frac{2\alpha-1}{3-2\alpha}\frac{\|V\|_F^2}{\gamma\sigma_{\min}^2(V)\sigma_{\min}^2(U)}\xi(\omega,U)^{t+1}\|Ux_*\|_2^2,$$

*where*

$$\zeta(\alpha, V) = 1 - (3\alpha - 2\alpha^2)\frac{\sigma_{\min}^2(V)}{\|V\|_F^2}, \quad \xi(\omega, U) = 1 - \omega(2-\omega)\frac{\sigma_{\min}^2(U)}{\|U\|_F^2}, \quad \gamma = \min_{1\le i\le k}\left(\|V^{(i)}\|_2^2\right),$$

*with $\omega \in (0,2)$ and $\alpha \in [1,3/2)$ being the given positive constants.*

To prove this theorem, we need the following Lemmas 3.3 and 3.5. Let $\mathbb{E}^V$ denote the expected value taken over the choice of rows in $V$, $\mathbb{E}^U$ the expected value taken over the choice of rows in $U$ and when necessary the choice of columns in $U$ and the iterate $\widetilde{b}_t$, the GRK($\alpha$) with $\alpha \in [1,3/2)$ solving the linear system $Vb = x_*$ rather than $Vb = x_t$ at the $t$th iteration.

**Lemma 3.3.** *Let $\widetilde{b}_{t+1} = b_t + \alpha \frac{x_*^{(i_t)} - V^{(i_t)} b_t}{\|V^{(i_t)}\|_2^2}(V^{(i_t)})^T$. For any $\alpha \in [1,3/2)$ and $i_t \in \{1,2,\dots,k\}$, in Methods 3.3 and 3.4 we have that*

$$\langle \widetilde{b}_{t+1} - b_*, b_{t+1} - \widetilde{b}_{t+1}\rangle \le \frac{1}{2}(\alpha^2 - \alpha)\left(\frac{\left|V^{(i_t)}(b_* - b_t)\right|^2}{\|V^{(i_t)}\|_2^2} + \frac{\left|x_{t+1}^{(i_t)} - x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2}\right) \tag{3.9}$$

*and*

$$\|\widetilde{b}_{t+1} - b_*\|_2^2 = \|b_t - b_*\|_2^2 - \alpha(2-\alpha)\frac{\left|V^{(i_t)}(b_* - b_t)\right|^2}{\|V^{(i_t)}\|_2^2}. \tag{3.10}$$

*Proof.* We prove (3.9) by the following direct substitution and expansion:

$$\begin{aligned}
\langle \widetilde{b}_{t+1} - b_*, b_{t+1} - \widetilde{b}_{t+1} \rangle &= \left\langle b_t - b_* + \alpha \frac{x_*^{(i_t)} - V^{(i_t)} b_t}{\|V^{(i_t)}\|_2^2} \big(V^{(i_t)}\big)^T, \alpha \frac{x_{t+1}^{(i_t)} - x_*^{(i_t)}}{\|V^{(i_t)}\|_2^2} \big(V^{(i_t)}\big)^T \right\rangle \\
&= \alpha^2 \left\langle \frac{x_*^{(i_t)} - V^{(i_t)} b_t}{\|V^{(i_t)}\|_2^2} \big(V^{(i_t)}\big)^T, \frac{x_{t+1}^{(i_t)} - x_*^{(i_t)}}{\|V^{(i_t)}\|_2^2} \big(V^{(i_t)}\big)^T \right\rangle \\
&\quad + \alpha \left\langle b_t - b_*, \frac{x_{t+1}^{(i_t)} - x_*^{(i_t)}}{\|V^{(i_t)}\|_2^2} \big(V^{(i_t)}\big)^T \right\rangle \\
&= \alpha^2 \frac{V^{(i_t)}(b_* - b_t)(x_{t+1}^{(i_t)} - x_*^{(i_t)})}{\|V^{(i_t)}\|_2^2} - \alpha \frac{V^{(i_t)}(b_* - b_t)(x_{t+1}^{(i_t)} - x_*^{(i_t)})}{\|V^{(i_t)}\|_2^2} \\
&= (\alpha^2 - \alpha) \frac{V^{(i_t)}(b_* - b_t)}{\|V^{(i_t)}\|_2} \cdot \frac{x_{t+1}^{(i_t)} - x_*^{(i_t)}}{\|V^{(i_t)}\|_2} \\
&\le \frac{1}{2}(\alpha^2 - \alpha) \left( \frac{\big|V^{(i_t)}(b_* - b_t)\big|^2}{\|V^{(i_t)}\|_2^2} + \frac{\big|x_{t+1}^{(i_t)} - x_*^{(i_t)}\big|^2}{\|V^{(i_t)}\|_2^2} \right),
\end{aligned}$$

here the third equality is achieved by using the fact that

$$V^{(i_t)} b_* = x_* \quad \text{for any } i_t \in \{1, 2, \ldots, k\},$$

since $V$ is underdetermined, in the last inequality we have used the following inequality of arithmetic and geometric means

$$ab \le \frac{1}{2}(a^2 + b^2) \quad \text{for any } a \in \mathbb{R} \text{ and } b \in \mathbb{R},$$

and the condition $\alpha \in [1, 3/2)$, which directly gives $\alpha^2 - \alpha \ge 0$.

Also, for (3.10), it follows from straightforward computations that

$$\begin{aligned}
\|\widetilde{b}_{t+1} - b_*\|_2^2 &= \left\| b_t - b_* + \alpha \frac{x_*^{(i_t)} - V^{(i_t)} b_t}{\|V^{(i_t)}\|_2^2} (V^{(i_t)})^T \right\|_2^2 \\
&= \|b_t - b_*\|_2^2 + \alpha^2 \frac{|V^{(i_t)}(b_* - b_t)|^2}{\|V^{(i_t)}\|_2^2} - 2\alpha \frac{|V^{(i_t)}(b_* - b_t)|^2}{\|V^{(i_t)}\|_2^2} \\
&= \|b_t - b_*\|_2^2 - \alpha(2 - \alpha) \frac{|V^{(i_t)}(b_* - b_t)|^2}{\|V^{(i_t)}\|_2^2},
\end{aligned}$$

which completes the proof. $\square$

*Remark* 3.4. This lemma includes the case $\alpha = 1$, i.e., Lemma 4.4 in [25], as the special case.

**Lemma 3.5.** *In Methods* 3.3 *and* 3.4 *we can bound the expected norm squared error of* $b_{t+1}-b_*$ *as*

$$\begin{aligned}\mathbb{E}_t\|b_{t+1}-b_*\|_2^2 &\le \zeta(\alpha,V)\|b_t-b_*\|_2^2 \\ &\quad + (2\alpha^2-\alpha)\mathbb{E}_t^U\frac{\|x_{t+1}-x_*\|_2^2}{\gamma} \quad \textit{for any } \alpha\in[1,3/2),\end{aligned} \tag{3.11}$$

*where* $\zeta(\alpha,V)=1-(3\alpha-2\alpha^2)\frac{\sigma_{\min}^2(V)}{\|V\|_F^2}$ *and* $\gamma=\min_{1\le i\le k}\left(\|V^{(i)}\|_2^2\right)$.

*Proof.*

$$\begin{aligned}\|b_{t+1}-b_*\|_2^2 &= \|b_{t+1}-\widetilde{b}_{t+1}+\widetilde{b}_{t+1}-b_*\|_2^2 \\ &= \|b_{t+1}-\widetilde{b}_{t+1}\|_2^2+\|\widetilde{b}_{t+1}-b_*\|_2^2+2\langle\widetilde{b}_{t+1}-b_*,b_{t+1}-\widetilde{b}_{t+1}\rangle \\ &= \alpha^2\frac{\left|x_{t+1}^{(i_t)}-x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2}+\|\widetilde{b}_{t+1}-b_*\|_2^2+2\langle\widetilde{b}_{t+1}-b_*,b_{t+1}-\widetilde{b}_{t+1}\rangle \\ &\le \alpha^2\frac{\left|x_{t+1}^{(i_t)}-x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2}+\|b_t-b_*\|_2^2-\alpha(2-\alpha)\frac{|V^{(i_t)}(b_*-b_t)|^2}{\|V^{(i_t)}\|_2^2} \\ &\quad +(\alpha^2-\alpha)\left(\frac{|V^{(i_t)}(b_*-b_t)|^2}{\|V^{(i_t)}\|_2^2}+\frac{\left|x_{t+1}^{(i_t)}-x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2}\right) \\ &= \|b_t-b_*\|_2^2-(3\alpha-2\alpha^2)\frac{|V^{(i_t)}(b_*-b_t)|^2}{\|V^{(i_t)}\|_2^2}+(2\alpha^2-\alpha)\frac{\left|x_{t+1}^{(i_t)}-x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2},\end{aligned}$$

here the fourth inequality is application of (3.9) and (3.10).

Based on this inequality, we have

$$\begin{aligned}&\mathbb{E}_t^V\|b_{t+1}-b_*\|_2^2 \\ &\le \|b_t-b_*\|_2^2-(3\alpha-2\alpha^2)\mathbb{E}_t^V\frac{|V^{(i_t)}(b_*-b_t)|^2}{\|V^{(i_t)}\|_2^2}+(2\alpha^2-\alpha)\mathbb{E}_t^V\frac{\left|x_{t+1}^{(i_t)}-x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2} \\ &= \|b_t-b_*\|_2^2-(3\alpha-2\alpha^2)\sum_{i_t\in\mathcal{U}_t'}\frac{|x_*^{(i_t)}-V^{(i_t)}b_t|^2}{\sum_{i\in\mathcal{U}_t'}|x_*^{(i_t)}-V^{(i_t)}b_t|^2}\cdot\frac{|V^{(i_t)}(b_*-b_t)|^2}{\|V^{(i_t)}\|_2^2} \\ &\quad +(2\alpha^2-\alpha)\sum_{i_t\in\mathcal{U}_t'}\frac{\left|x_*^{(i_t)}-V^{(i_t)}b_t\right|^2}{\sum_{i\in\mathcal{U}_t'}\left|x_*^{(i_t)}-V^{(i_t)}b_t\right|^2}\cdot\frac{\left|x_{t+1}^{(i_t)}-x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2} \\ &\le \|b_t-b_*\|_2^2-(3\alpha-2\alpha^2)\epsilon_t'\|x_*-Vb_t\|_2^2 \\ &\quad +(2\alpha^2-\alpha)\sum_{i_t\in\mathcal{U}_t'}\frac{\left|x_*^{(i_t)}-V^{(i_t)}b_t\right|^2}{\sum_{i\in\mathcal{U}_t'}\left|x_*^{(i_t)}-V^{(i_t)}b_t\right|^2}\cdot\frac{\left|x_{t+1}^{(i_t)}-x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2} \\ &\le \|b_t-b_*\|_2^2-(3\alpha-2\alpha^2)\epsilon_t'\|x_*-Vb_t\|_2^2 \\ &\quad +(2\alpha^2-\alpha)\sum_{i_t\in\mathcal{U}_t'}\frac{\left|x_*^{(i_t)}-V^{(i_t)}b_t\right|^2}{\sum_{i\in\mathcal{U}_t'}\left|x_*^{(i_t)}-V^{(i_t)}b_t\right|^2}\cdot\frac{\|x_{t+1}-x_*\|_2^2}{\gamma}\end{aligned}$$

$$
\begin{aligned}
&= \|b_t - b_*\|_2^2 - (3\alpha - 2\alpha^2)\epsilon_t' \|V(b_* - b_t)\|_2^2 + (2\alpha^2 - \alpha)\frac{\|x_{t+1} - x_*\|_2^2}{\gamma} \\
&\le \left(1 - (3\alpha - 2\alpha^2)\frac{\sigma_{\min}^2(V)}{\|V\|_F^2}\right)\|b_t - b_*\|_2^2 + (2\alpha^2 - \alpha)\frac{\|x_{t+1} - x_*\|_2^2}{\gamma},
\end{aligned}
$$

where $\gamma = \min_{1\le i\le k}\left(\|V^{(i)}\|_2^2\right)$.

Here the second equality is obtained with the use of the following probability of row $i_t \in \mathcal{U}_t'$

$$
\Pr(\text{row} = i_t) = \frac{\left|x_*^{(i_t)} - V^{(i_t)}b_t\right|^2}{\sum_{i\in\mathcal{U}_t'}\left|x_*^{(i_t)} - V^{(i_t)}b_t\right|^2},
$$

where

$$
\mathcal{U}_t' = \left\{ i_t \mid \left|x_*^{(i_t)} - V^{(i_t)}b_t\right|^2 \ge \epsilon_t' \|x_* - Vb_t\|_2^2 \|V^{(i_t)}\|_2^2 \right\}
$$

with

$$
\epsilon_t' = \frac{1}{2}\left(\frac{1}{\|x_* - Vb_t\|_2^2}\max_{1\le i_t\le m}\left\{\frac{\left|x_*^{(i_t)} - V^{(i_t)}b_t\right|^2}{\|V^{(i_t)}\|_2^2}\right\} + \frac{1}{\|V\|_F^2}\right).
$$

$\mathcal{U}_t'$ and $\epsilon_t'$ are generated in the process of solving $Vb = x_*$ by GRK($\alpha$) with $\alpha \in [1, 3/2)$, which lead to the inequality

$$
\left|x_*^{(i_t)} - V^{(i_t)}b_t\right|^2 \ge \epsilon_t' \|x_* - Vb_t\|_2^2 \|V^{(i_t)}\|_2^2 \quad \text{for any } i_t \in \mathcal{U}_t'.
$$

From the above estimate and the condition $3\alpha - 2\alpha^2 > 0$ for any $\alpha \in [1, 3/2)$, we can get the third inequality. The fourth inequality is achieved by using the following estimate

$$
\frac{\left|x_{t+1}^{(i_t)} - x_*^{(i_t)}\right|^2}{\|V^{(i_t)}\|_2^2} \le \frac{\|x_{t+1} - x_*\|_2^2}{\|V^{(i_t)}\|_2^2} \le \frac{\|x_{t+1} - x_*\|_2^2}{\gamma}, \quad \forall\, i_t \in \mathcal{U}_t',
$$

where $\gamma = \min_{1\le i\le k}\left(\|V^{(i)}\|_2^2\right)$. The last inequality is application of (3.5) and (3.6) by simply replacing $X$ and $y$ with $V$ and $x_*$, respectively.

Therefore, it holds that

$$
\begin{aligned}
\mathbb{E}_t\|b_{t+1} - b_*\|_2^2 &= \mathbb{E}_t^U \mathbb{E}_t^V \|b_{t+1} - b_*\|_2^2 \\
&\le \left(1 - (3\alpha - 2\alpha^2)\frac{\sigma_{\min}^2(V)}{\|V\|_F^2}\right)\|b_t - b_*\|_2^2 + (2\alpha^2 - \alpha)\mathbb{E}_t^U\frac{\|x_{t+1} - x_*\|_2^2}{\gamma}.
\end{aligned}
$$

We then immediately achieve the desired bound that we were proving. □

Now, we can give the proof of Theorem 3.2.

*Proof of Theorem* 3.2. Note that $b_* = \beta_*$ is valid for $k < m, n$ when $X$ is underdetermined or overdetermined (also for $\min\{m, n\} < k < \max\{m, n\}$ when $X$ is overdetermined and consistent) [25, Section 3]. Therefore, we only need to bound the term $\mathbb{E}_t^U\|x_{t+1} - x_*\|_2^2$ in

(3.11) by using the bounds (3.1) or (3.2) depending on whether we are using Methods 3.3 or 3.4, respectively.

For Method 3.3, applying the bound (3.1) to the term $\mathbb{E}_t^U\|x_{t+1}-x_*\|_2^2$ in (3.11), we can immediately obtain

$$\mathbb{E}_t\|b_{t+1}-b_*\|_2^2 \le \zeta(\alpha,V)\|b_t-b_*\|_2^2 + (2\alpha^2-\alpha)\xi(\omega,U)^{t+1}\frac{\|x_*\|_2^2}{\gamma},$$

where $\zeta(\alpha,V)=1-(3\alpha-2\alpha^2)\frac{\sigma_{\min}^2(V)}{\|V\|_F^2}$.

Since $0<3\alpha-2\alpha^2\le 1$ for any $\alpha\in[1,3/2)$, it holds that

$$0<1-(3\alpha-2\alpha^2)\frac{\sigma_{\min}^2(V)}{\|V\|_F^2}<1.$$

Consequently, by taking expectations over the randomness from the first $t$ iterations and using the law of iterated expectation, we have

$$\begin{aligned}\mathbb{E}\|b_{t+1}-b_*\|_2^2 &\le \zeta(\alpha,V)^{t+1}\|b_*\|_2^2 + (2\alpha^2-\alpha)\xi(\omega,U)^{t+1}\frac{\|x_*\|_2^2}{\gamma}\sum_{h=0}^{t}\zeta(\alpha,V)^h\\ &\le \zeta(\alpha,V)^{t+1}\|b_*\|_2^2 + \frac{2\alpha^2-\alpha}{\gamma}\cdot\frac{1}{1-\zeta(\alpha,V)}\xi(\omega,U)^{t+1}\|x_*\|_2^2\\ &= \zeta(\alpha,V)^{t+1}\|b_*\|_2^2 + \frac{2\alpha-1}{3-2\alpha}\cdot\frac{\|V\|_F^2}{\gamma\sigma_{\min}^2(V)}\xi(\omega,U)^{t+1}\|x_*\|_2^2.\end{aligned}$$

For Method 3.4, with the substitution of estimate (3.5) into (3.11), we can further obtain

$$\mathbb{E}_t\|b_{t+1}-b_*\|_2^2 \le \zeta(\alpha,V)\|b_t-b_*\|_2^2 + \frac{2\alpha^2-\alpha}{\gamma\sigma_{\min}^2(U)}\mathbb{E}_t^U\|x_{t+1}-x_*\|_{U^TU}^2,$$

then, applying the bound (3.2) to the term $\mathbb{E}_t^U\|x_{t+1}-x_*\|_{U^TU}^2$, we have

$$\mathbb{E}_t\|b_{t+1}-b_*\|_2^2 \le \zeta(\alpha,V)\|b_t-b_*\|_2^2 + \frac{2\alpha^2-\alpha}{\gamma\sigma_{\min}^2(U)}\xi(\omega,U)^{t+1}\|x_*\|_{U^TU}^2,$$

taking expectations over the remaining randomness, we have

$$\begin{aligned}\mathbb{E}\|b_{t+1}-b_*\|_2^2 &\le \zeta(\alpha,V)^{t+1}\|b_*\|_2^2 + \frac{2\alpha^2-\alpha}{\gamma\sigma_{\min}^2(U)}\xi(\omega,U)^{t+1}\|x_*\|_{U^TU}^2\sum_{h=0}^{t}\zeta(\alpha,V)^h\\ &\le \zeta(\alpha,V)^{t+1}\|b_*\|_2^2 + \frac{2\alpha^2-\alpha}{\gamma\sigma_{\min}^2(U)}\cdot\frac{1}{1-\zeta(\alpha,V)}\xi(\omega,U)^{t+1}\|x_*\|_{U^TU}^2\\ &= \zeta(\alpha,V)^{t+1}\|b_*\|_2^2 + \frac{2\alpha-1}{3-2\alpha}\cdot\frac{\|V\|_F^2}{\gamma\sigma_{\min}^2(V)\sigma_{\min}^2(U)}\xi(\omega,U)^{t+1}\|Ux_*\|_2^2.\end{aligned}$$

This concludes the proof of the theorem. □

## 4. Numerical experiments

In this section, we implement the RK-RK, REK-RK, GRK($\omega$)-GRK($\alpha$) and GRGS($\omega$)-GRK($\alpha$) methods with different parameters $\omega$ and $\alpha$ in different settings, and show the numerical behaviors of these methods in terms of the number of iteration steps (denoted as "IT") and the computing time in seconds (denoted as "CPU"). Here, the CPU and IT mean the arithmetical averages of the elapsed CPU times and the required iteration steps with respect to 50 times repeated runs of the corresponding method. Besides, we report the speed-up of GRK($\omega$)-GRK($\alpha$) against RK-RK, which is defined as

$$\text{speed-up} = \frac{\text{CPU of GRK}(\omega)\text{-GRK}(\alpha)}{\text{CPU of RK-RK}}$$

when $\omega, \alpha \neq 1$. Naturally, for inconsistent systems, namely for Case IV in Table 2.1, speed-up represents the CPU of GRGS($\omega$)-GRK($\alpha$) against the CPU of REK-RK.

We are going to solve the factorized linear system (1.1) with $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{k\times n}$ from two sources. One is randomly generated matrices using the MATLAB function `randn` subject to the standard normal distribution `N(0,1)`. The other is derived from the MATLAB function `nnmf`, which factors the nonnegative matrix $X \in \mathbb{R}^{m\times n}$ into nonnegative matrices $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{k\times n}$ for fixed positive integer $k$. The nonnegative matrices $X \in \mathbb{R}^{m\times n}$ are taken from the real world data sets on wine quality and bike rental data, which can be download from the UCI Machine Learning Repository [21]. The wine quality set is a sample of $m = 1599$ red wines with $n = 11$ physio-chemical properties of each wine, and the Euclidean condition number of $X$ is $2.46 \times 10^3$. The bike rental data sets contains $m = 17379$ samples and $n = 9$ attributes per sample, the Euclidean condition number of $X$ is 94.27.

In our implementations, we divide the system (1.1) into consistent and inconsistent to discuss the numerical efficiency of the aforementioned various methods. When the system (1.1) is consistent, we set $m, n, k \in \{200, 150, 100\}$, which is in accordance with the experiments in [25], and $m, n, k \in \{1500, 1000, 750\}$ for the randomly generated matrices, respectively. The specific sizes of $U$ and $V$ correspond to the first three cases in Table 2.1, i.e., Case I, Case II and Case III. For example, for Case I, $k < m, n$ and $X = UV$ is underdetermined, then $k = 100$, $m = 150$ and $n = 200$ or $k = 750$, $m = 1000$ and $n = 1500$. When the system (1.1) is inconsistent, for randomized matrix, we set $m = 1200$, $n = 750$ and $k = 500$, which is also in accordance with the experiments in [25], for wine quality and bike rental data, we set $k = 5$ and $k = 8$, respectively, and the Euclidean condition numbers of the matrices $U$ and $V$ obtained by `nnmf(X,k)` are 23.9698, 4.1975 and 50.5527, 7.3502 (because this factorization is not unique, we cannot guarantee that $U$ and $V$ are exactly the same as in [25]), respectively. The solution vector $\beta_*$ is generated randomly by using the function `randn` such that its entries obey the independent standard normal distribution, and the right-hand side $y \in \mathbb{R}^m$ is taken to be $X\beta_*$ when $X$ is consistent, and $y = X\beta_* + r$ when $X$ is inconsistent, where $r \in \texttt{null}(X^T)$, and $\texttt{null}(X^T)$ is generated by making use of the MATLAB function `null`. All computations are started from the initial vectors $x_0 = 0$ and $b_0 = 0$, and terminated once the solution error, $\|b_t - \beta_*\|_2$, at the current iterate $b_t$, satisfies $\|b_t - \beta_*\|_2 < 10^{-6}$, or the number of iteration steps exceeds $200,000$. The latter is given a label '$-$' in the numerical tables. In addition, all experiments are carried out 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).

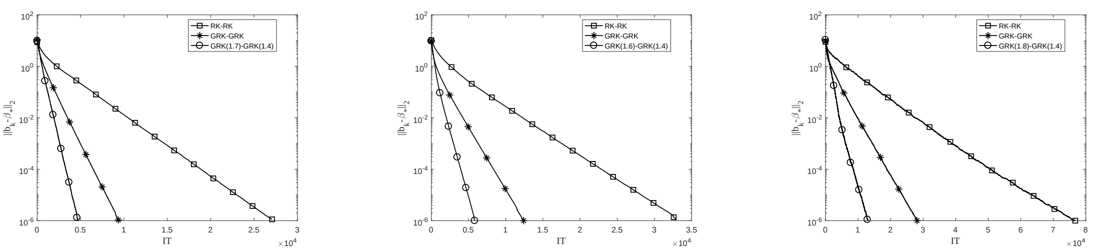
| Case | | I | II | III |
|---|---|---|---|---|
| $(m,n,k)$ | | $(150,200,100)$ | $(200,150,100)$ | $(200,100,150)$ |
| $(\omega,\alpha)$ | | $(1.7,1.4)$ | $(1.6,1.4)$ | $(1.8,1.4)$ |
| RK-RK | IT | 27286.4 | 33515.4 | 76730.4 |
| | CPU | 6.1675 | 7.5170 | 17.8449 |
| GRK-GRK | IT | 9432.2 | 12302.6 | 28140.8 |
| | CPU | 3.7520 | 4.9760 | 12.1450 |
| GRK($\omega$)-GRK($\alpha$) | IT | 4731.2 | 5867.2 | 13021.6 |
| | CPU | 1.8906 | 2.3623 | 5.6964 |
| speed-up | | 3.26 | 3.18 | 3.13 |

Table 4.1: IT and CPU of RK-RK, GRK-GRK and GRK($\omega$)-GRK($\alpha$) for the randomly generated matrices with $m,n,k \in \{150,200,100\}$.

In Tables 4.1 and 4.2, we list the numbers of iteration steps and the computing times for RK-RK, GRK-GRK (i.e., GRK($\omega$)-GRK($\alpha$) method with $\omega = \alpha = 1$) and GRK($\omega$)-GRK($\alpha$) methods when $X$ is consistent. The parameters $\omega$ and $\alpha$ are taken to be the experimentally computed optimal ones that minimize the total number of iteration steps of the GRK($\omega$)-GRK($\alpha$) method. The results in these two tables show that GRK($\omega$)-GRK($\alpha$) with appropriate choices of the relaxation parameters $\omega$ and $\alpha$ can always successfully compute an approximate solution to the factorized linear system (1.1), but RK-RK fails for large-scale factorized linear system (1.1) with $m,n,k \in \{1500,1000,750\}$ due to the number of the iteration steps exceeding $200,000$. For all convergent cases, both GRK-GRK and GRK($\omega$)-GRK($\alpha$) significantly outperform RK-RK in terms of iteration steps and CPU times, and when compared with GRK-GRK, GRK($\omega$)-GRK($\alpha$) requires much smaller iteration steps and costs much less CPU times than the GRK-GRK for appropriate parameters.

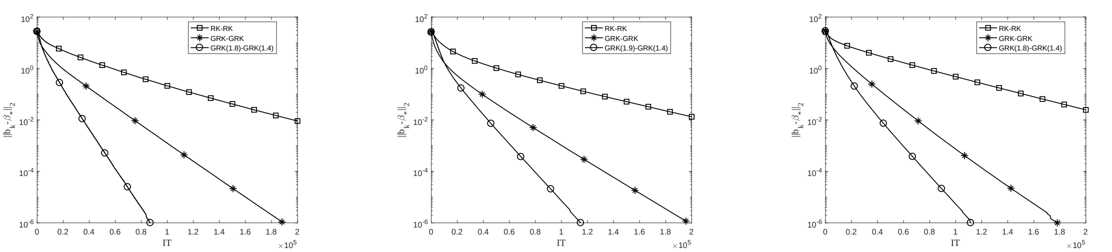
| Case | | I | II | III |
|---|---|---|---|---|
| $(m, n, k)$ | | $(1000, 1500, 750)$ | $(1500, 1000, 750)$ | $(1500, 750, 1000)$ |
| $(\omega, \alpha)$ | | $(1.8, 1.4)$ | $(1.9, 1.4)$ | $(1.8, 1.4)$ |
| RK-RK | IT | – | – | – |
| | CPU | – | – | – |
| GRK-GRK | IT | 189048.3 | 197693.6 | 178632.2 |
| | CPU | 1179.1540 | 1294.3189 | 1426.1928 |
| GRK($\omega$)-GRK($\alpha$) | IT | 87284.3 | 114011.0 | 112438.0 |
| | CPU | 542.2718 | 717.5274 | 969.0418 |
| speed-up | | – | – | – |

Table 4.2: IT and CPU of RK-RK, GRK-GRK and GRK($\omega$)-GRK($\alpha$) for the randomly generated matrices with $m, n, k \in \{1000, 1500, 750\}$.




(a) $m = 150$, $n = 200$, $k = 100$ (b) $m = 200$, $n = 150$, $k = 100$ (c) $m = 200$, $n = 100$, $k = 150$

Figure 4.1: $\|b_t - \beta_*\|_2$ versus IT for RK-RK, GRK-GRK and GRK($\omega$)-GRK($\alpha$) methods for the randomly generated matrices.



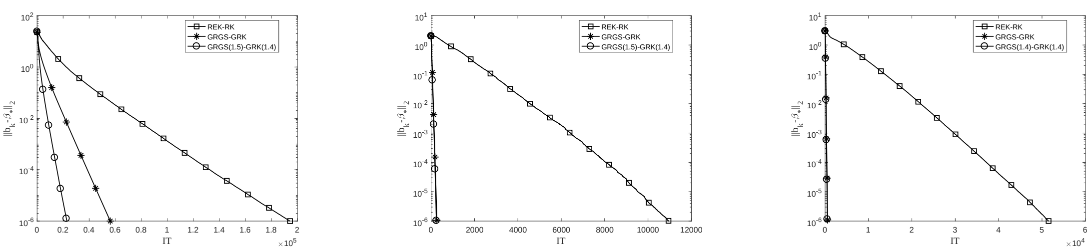

(a) $m = 1000$, $n = 1500$, $k = 750$ (b) $m = 1500$, $n = 1000$, $k = 750$ (c) $m = 1500$, $n = 750$, $k = 1000$

Figure 4.2: $\|b_t - \beta_*\|_2$ versus IT for RK-RK, GRK-GRK and GRK($\omega$)-GRK($\alpha$) methods for the randomly generated matrices.

Note that the speed-up of GRK($\omega$)-GRK($\alpha$) against RK-RK is at least 3.13, and the biggest reaches 3.26. The above observations are intuitively shown in Figures 4.1 and 4.2, in which we depict the curves of the solution error $\|b_t - \beta_*\|_2$ versus the iteration step. We observe that the solution error $\|b_t - \beta_*\|_2$ of GRK($\omega$)-GRK($\alpha$) with appropriate parameters is decaying more rapidly than that of RK-RK and GRK-GRK when the iteration step is increasing.

| Case | | IV | wine quality | bike rental data |
|---|---|---|---|---|
| $(m, n, k)$ | | $(1200, 750, 500)$ | $(1599, 11, 5)$ | $(17379, 9, 8)$ |
| $(\omega, \alpha)$ | | $(1.5, 1.4)$ | $(1.5, 1.4)$ | $(1.4, 1.4)$ |
| REK-RK | IT | 194359.9 | 11008.7 | 50928.8 |
| | CPU | 78.8764 | 4.2139 | 40.8345 |
| GRGS-GRK | IT | 56223.5 | 257.2 | 583.3 |
| | CPU | 127.2329 | 0.0691 | 0.3080 |
| GRGS($\omega$)-GRK($\alpha$) | IT | 22921.3 | 231 | 497.3 |
| | CPU | 47.6803 | 0.0618 | 0.2622 |
| speed-up | | 1.65 | 68.18 | 155.74 |

Table 4.3: IT and CPU of REK-RK, GRGS-GRK and GRGS($\omega$)-GRK($\alpha$) for the randomly generated matrix with $m = 1200$, $n = 750$, $k = 500$, wine quality and bike rental data.

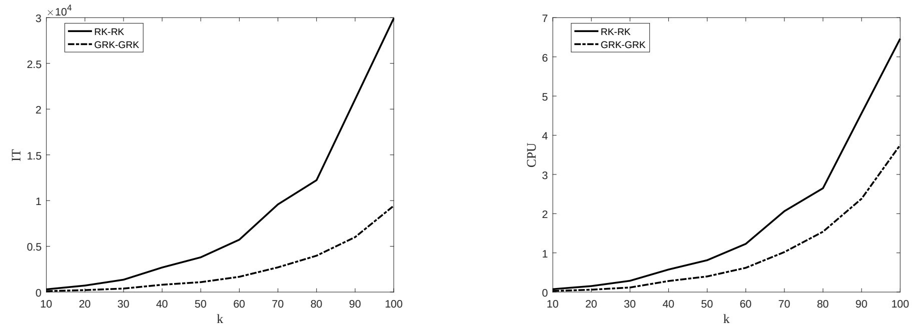



(a) $m = 1200$, $n = 750$, $k = 500$ (b) wine quality: $m = 1599$, $n = 11$, $k = 5$ (c) bike rental data: $m = 17379$, $n = 9$, $k = 8$

Figure 4.3: $\|b_t - \beta_*\|_2$ versus IT for REK-RK, GRGS-GRK and GRGS($\omega$)-GRK($\alpha$) methods for the randomly generated matrix.

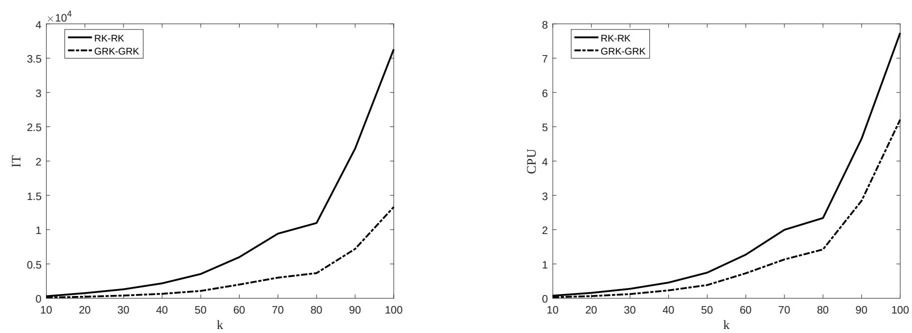
In Table 4.3, we report iteration counts and CPU times for REK-RK, GRGS-GRK (i.e., GRGS($\omega$)-GRK($\alpha$) method with $\omega = \alpha = 1$) and GRGS($\omega$)-GRK($\alpha$) methods when $X$ is inconsistent. The parameters $\omega$ and $\alpha$ are also taken to be the experimentally computed optimal ones that minimize the total number of iteration steps of GRGS($\omega$)-

GRK($\alpha$) method. The results in this table show that GRGS($\omega$)-GRK($\alpha$) with appropriate parameters performs much better in iteration counts and CPU times than REK-RK for any inconsistent matrix for this example, especially for bike rental data, the speed-up even attains 155.74. In addition, GRGS-GRK also takes many fewer iteration steps and much less CPU times than REK-RK except for the randomized matrix ($m = 1200$, $n = 750$ and $k = 500$) in terms of CPU time. Figure 4.3 shows the performance of these three methods on inconsistent systems. From this figure, we also find that for GRGS-GRK and GRGS($\omega$)-GRK($\alpha$) methods the solution error $\|b_t - \beta_*\|$ is decaying much more quickly than that of REK-RK with respect to the increase of the iteration step.




(a) IT (b) CPU

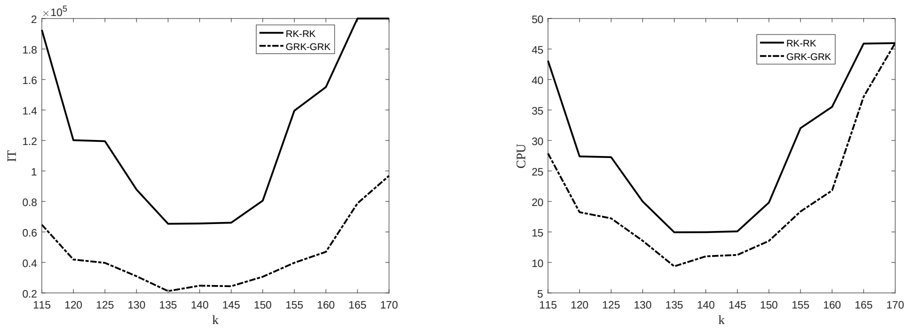
Figure 4.4: IT and CPU versus $k$ for RK-RK and GRK-GRK methods for consistent factorized linear system (1.1) with random matrices $U \in \mathbb{R}^{150\times k}$ and $V \in \mathbb{R}^{k\times 200}$ ($k < 150$).




(a) IT (b) CPU

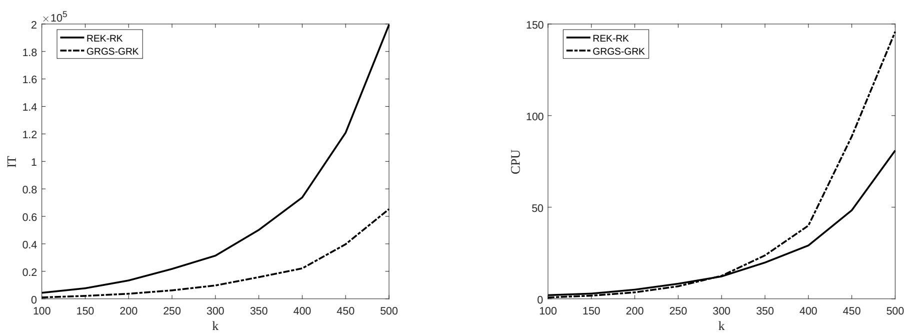
Figure 4.5: IT and CPU versus $k$ for RK-RK and GRK-GRK methods for consistent factorized linear system (1.1) with random matrices $U \in \mathbb{R}^{200\times k}$ and $V \in \mathbb{R}^{k\times 150}$ ($k < 150$).

In Figures 4.4–4.7, we plot the curves of the iteration step and CPU time versus the number of columns of the randomly generated matrix $U \in \mathbb{R}^{m\times k}$ for RK-RK, GRK-GRK, REK-RK and GRGS-GRK methods when the factorized linear system (1.1) is either consistent or inconsistent. For GRK($\omega$)-GRK($\alpha$) and GRGS($\omega$)-GRK($\alpha$) methods, there may be different optimal parameters $\omega$ and $\alpha$ with respect to different $k$. Therefore, in order to facilitate the comparison, we only report the results of GRK-GRK and GRGS-GRK methods rather than the relaxed version. As the results in Figures 4.4–4.6 show, for each tested value $k$ the GRK-GRK method takes many fewer iteration steps and less CPU times than the RK-RK method for consistent factorized linear system (1.1), especially when $k \geq 60$ with respect to $m, n \in \{150, 200\}$ and $k \in \{115, 120, \ldots, 170\}$ with respect to $m = 200$, $n = 100$. From Figure 4.7 we see that GRGS-GRK significantly outperforms REK-RK in terms of IT for all tested values of $k \in \{100, 150, \ldots, 500\}$, however, GRGS-GRK costs slightly less CPU time than REK-RK when $k < 300$ and REK-RK performs much better in CPU time than GRGS-GRK when $k > 300$.




(a) IT (b) CPU

Figure 4.6: IT and CPU versus $k$ for RK-RK and GRK-GRK methods for consistent factorized linear system (1.1) with random matrices $U \in \mathbb{R}^{200\times k}$ and $V \in \mathbb{R}^{k\times 100}$ ($100 < k < 200$).




(a) IT (b) CPU

Figure 4.7: IT and CPU versus $k$ for REK-RK and GRGS-GRK methods for inconsistent factorized linear system (1.1) with random matrices $U \in \mathbb{R}^{1200\times k}$ and $V \in \mathbb{R}^{k\times 750}$ ($k < 750$).

Last, we show the advantage of our algorithms on Phillips's famous test problem [13]. Consider the following Fredholm integral equation of first kind on the square $[-6,6] \times [-6,6]$

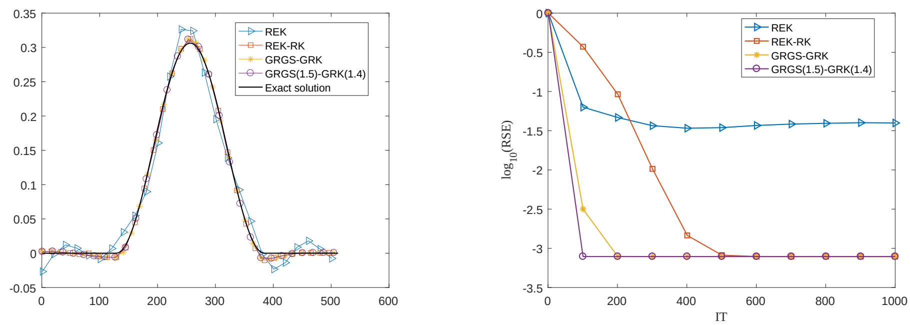
$$\int_{-6}^{6} K(s,t)\phi(t)\,dt = f(s),$$

where the kernel function $K(s,t) = \phi(s-t)$ with

$$\phi(t) = \begin{cases} 1+\cos(\pi t/3), & |t| < 3, \\ 0, & |t| \geq 3, \end{cases}$$

and the right-hand side

$$f(s) = (6-|s|)\left(1+\frac{1}{2}\cos(s\pi/3)\right) + \frac{9}{2\pi}\sin(|s|\pi/3).$$




(a) (b)

Figure 4.8: (a) The solutions obtained by using REK, REK-RK, GRGS-GRK and GRGS(1.5)-GRK(1.4) methods; and also the solution of the exact problem (black line). (b) $\log_{10}$(RSE) versus IT for REK, REK-RK, GRGS-GRK and GRGS(1.5)-GRK(1.4) methods.

We mainly consider the following perturbed linear system

$$X\beta = \widetilde{y},$$

where $\widetilde{y} = \frac{1}{K}\sum_{r=1}^{K} \widetilde{y}_r$, with $\widetilde{y}_r = y + \varepsilon_r$, $\varepsilon_r$ is Gaussian noise with $\frac{\|y-\widetilde{y}_r\|_2}{\|y\|_2} = 0.1$, $\|y-\widetilde{y}\|_2 \approx 0.014$ and $K = 50$ [16]. The coefficient matrix $X \in \mathbb{R}^{n\times n}$, the solution vector $\beta_* \in \mathbb{R}^n$ and the vector $y$ can be constructed by MATLAB function `phillips(n)`, which were published by Ivanov in [14]. For this test problem, we set $n = 512$, $k = 8$ and compute $U$ and $V$ using MATLAB's `nnmf()`. The Euclidean condition numbers of $X$, $U$ and $V$ are $1.8174 \times 10^9$, 3.5433 and 3.5371 respectively. We plot the curves of the relative solution error

$$\text{RSE} = \frac{\|b_t - \beta_*\|_2^2}{\|\beta_*\|_2^2}$$

in base-10 logarithm versus the iteration step averaged over 50 runs for REK-RK, GRGS-GRK, GRGS(1.5)-GRK(1.4) and REK alone in Figure 4.8. For each algorithm no more than 1000 iterations were made. From this figure, we also see that both GRGS-GRK and GRGS(1.5)-GRK(1.4) methods are significantly faster than REK-RK and REK. In addition, GRGS(1.5)-GRK(1.4) requires smaller iteration steps than GRGS-GRK.

## 5. Conclusion

In this paper, we further introduce a relaxation parameter in iteration schemes of the GRK and GRGS methods, respectively, and obtain a class of relaxed GRK and GRGS methods for solving large sparse systems of linear equations. We have established the linear convergence theories for the GRK($\omega$) and GRGS($\omega$) methods with $\omega \in (0, 2)$. In addition, based on these two new methods, we have proposed two randomized iterative methods, i.e., GRK($\omega$)-GRK($\alpha$) and GRGS($\omega$)-GRK($\alpha$) with $\omega \in (0, 2)$ and $\alpha \in [1, 3/2)$, to solve factorized linear system (1.1). As a solver, our methods also converge linearly in expectation to the (least-squares or least-norm) solution of (overdetermined or underdetermined) factorized linear systems. Numerical experiments have verified that both GRK($\omega$)-GRK($\alpha$) and GRGS($\omega$)-GRK($\alpha$) (with appropriate parameters $\omega$ and $\alpha$) significantly outperform RK-RK and REK-RK in iteration counts and computing times for consistent and inconsistent systems, respectively. While we do not have a further analysis for suggestions for the choice of the relaxation parameter. Finding more effective parameter selection criteria (e.g., adaptive parameter selection criteria) for relaxed GRK or GRGS methods should be important and valuable topics in the future study.

## References

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

[2] Z.-Z. Bai and W.-T. Wu, *On convergence rate of the randomized Kaczmarz method*, Linear Algebra Appl. **553** (2018), 252–269.

[3] ——, *On greedy randomized Kaczmarz method for solving large sparse linear systems*, SIAM J. Sci. Comput. **40** (2018), no. 1, A592–A606.

[4] ——, *On relaxed greedy randomized Kaczmarz methods for solving large sparse linear systems*, Appl. Math. Lett. **83** (2018), 21–26.

[5] ———, *On greedy randomized coordinate descent methods for solving large linear least-squares problems*, Numer. Linear Algebra Appl. **26** (2019), no. 4, e2237, 15 pp.

[6] ———, *On partially randomized extended Kaczmarz method for solving large sparse overdetermined inconsistent linear systems*, Linear Algebra Appl. **578** (2019), 225–250.

[7] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math. **9** (2009), no. 6, 717–772.

[8] Y. Censor, P. P. B. Eggermont and D. Gordon, *Strong underrelaxation in Kaczmarz's method for inconsistent systems*, Numer. Math. **41** (1983), no. 1, 83–92.

[9] K. Du, *Tight upper bounds for the convergence of the randomized extended Kaczmarz and Gauss–Seidel algorithms*, Numer. Linear Algebra Appl. **26** (2019), no. 3, e2233, 14 pp.

[10] B. Dumitrescu, *On the relation between the randomized extended Kaczmarz algorithm and coordinate descent*, BIT **55** (2015), no. 4, 1005–1015.

[11] J. Goes, T. Zhang, R. Arora and G. Lerman, *Robust stochastic principal component analysis*, in Proceedings of the 17th International Conference on Artificial Intelligence and Statistics, J. Mach. Learn. Res. **2014**, 266–274.

[12] M. Hanke and W. Niethammer, *On the acceleration of Kaczmarz's method for inconsistent linear systems*, Linear Algebra Appl. **130** (1990), 83–98.

[13] P. C. Hansen, *Regularization tools version 4.0 for Matlab 7.3*, Numer. Algorithms **46** (2007), no. 2, 189–194.

[14] A. A. Ivanov, *Regularization Kaczmarz Tools Version 1.4 for Matlab*, MATLAB Central File Exchange, https://www.mathworks.com/matlabcentral/fileexchange/43791.

[15] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Int. Acad. Pol. Sci. Lett. A **35** (1937), 355–357.

[16] A. K. Kaveev, G. I. Kropotov, E. V. Tsygankova, I. A. Tzibizov, S. D. Ganichev, S. N. Danilov, P. Olbrich, C. Zoth, E. G. Kaveeva, A. I. Zhdanov, A. A. Ivanov, R. Z. Deyanov and B. Redlich, *Terahertz polarization conversion with quartz waveplate sets*, Applied Optics **52** (2013), no. 4, B60–B69.

[17] R. H. Keshavan, A. Montanari and S. Oh, *Matrix completion from noisy entries*, J. Mach. Learn. Res. **11** (2010), 2057–2078.

[18] V. Koltchinskii, K. Lounici and A. B. Tsybakov, *Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion*, Ann. Statist. **39** (2011), no. 5, 2302–2329.

[19] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, in Peoceedings of Advances in Neural Information Processing Systems **13** (2001), 556–562.

[20] D. Leventhal and A. S. Lewis, *Randomized methods for linear constraints: Convergence rates and conditioning*, Math. Oper. Res. **35** (2010), no. 3, 641–654.

[21] M. Lichman, *UCI Machine Learning Repository*, 2013, available at http://archive.ics.uci.edu/ml/index.php.

[22] Y. Liu and C.-Q. Gu, *Variant of greedy randomized Kaczmarz for ridge regression*, Appl. Numer. Math. **143** (2019), 223–246.

[23] A. Ma, A. Flenner, D. Needell and A. G. Percus, *Improving image clustering using sparse text and the wisdom of the crowds*, in Procedings of the 48th Asilomar Conference on Signals, Systems and Computers, IEEE, **2014**, 1555–1557.

[24] A. Ma, D. Needell and A. Ramdas, *Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods*, SIAM J. Matrix Anal. Appl. **36** (2015), no. 4, 1590–1604.

[25] ———, *Iterative methods for solving factorized linear systems*, SIAM J. Matrix Anal. Appl. **39** (2018), no. 1, 104–122.

[26] D. Needell, *Randomized Kaczmarz solver for noisy linear systems*, BIT **50** (2010), no. 2, 395–403.

[27] I. Pomparău and C. Popa, *Supplementary projections for the acceleration of Kaczmarz algorithm*, Appl. Math. Comput. **232** (2014), 104–116.

[28] C. Popa, *Least-squares solution of overdetermined inconsistent linear systems using Kaczmarz's relaxation*, Int. J. Comput. Math. **55** (1995), no. 1-2, 79–89.

[29] ———, *Extensions of block-projections methods with relaxation parameters to inconsistent and rank-deficient least-squares problems*, BIT **38** (1998), no. 1, 151–176.

[30] P. Richtárik and M. Takáč, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Math. Program. **144** (2014), no. 1-2, Ser. A, 1–38.

[31] T. Strohmer and R. Vershynin, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl. **15** (2009), no. 2, 262–278.

[32] W. Xu, X. Liu and Y. Gong, *Document clustering based on non-negative matrix factorization*, in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, **2003**, 267–273.

[33] A. Zouzias and N. M. Freris, *Randomized extended Kaczmarz for solving least-squares*, SIAM J. Matrix Anal. Appl. **34** (2013), no. 2, 773–793.

Shi-Min Liu
Department of Mathematics, Hefei University, Hefei 230601, China
*E-mail address*: `liushimin12@126.com`

Yong Liu
Department of Mathematics, Shanghai University, Shanghai 200444, China
*E-mail address*: `liuyong@shu.edu.cn`