Scientific Research Publishing

Journal of Applied Mathematics and Physics, 2023, 11, 1036-1048
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Greedy Randomized Gauss-Seidel Method with Oblique Direction

Weifeng Li, Pingping Zhang

School of Science, Chongqing University of Posts and Telecommunications, Chongqing, China
Email: Liwf1417@163.com

How to cite this paper: Li, W.F. and Zhang, P.P. (2023) Greedy Randomized Gauss-Seidel Method with Oblique Direction. *Journal of Applied Mathematics and Physics*, **11**, 1036-1048.
https://doi.org/10.4236/jamp.2023.114068

Received: March 21, 2023
Accepted: April 24, 2023
Published: April 27, 2023

Abstract

For the linear least squares problem with coefficient matrix columns being highly correlated, we develop a greedy randomized Gauss-Seidel method with oblique direction. Then the corresponding convergence result is deduced. Numerical examples demonstrate that our proposed method is superior to the greedy randomized Gauss-Seidel method and the randomized Gauss-Seidel method with oblique direction.

Keywords

Oblique Direction, Linear Least Squares Problem, Gauss-Seidel Method

1. Introduction

We focus on the linear least squares problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^n} \|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2, \tag{1}$$

where $\boldsymbol{X}\in\mathbb{R}^{m\times n}$ $(m\geq n)$ is of full column rank, $\boldsymbol{y}\in\mathbb{R}^m$ and $\|\cdot\|_2$ denotes the Euclidean norm. There is a wide range of applications for the least squares problem in many fields, such as signal processing, image restoration, and so on. As we know, the coordinate descent method is an effective iteration method for (1). It applies the Gauss-Seidel method to the following equivalent normal Equation

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{\beta}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}, \tag{2}$$

leading to the following formula

$$\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_k+\frac{\boldsymbol{X}_{j_k}^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_k)}{\|\boldsymbol{X}_{j_k}\|_2^2}\boldsymbol{e}_{j_k},\quad j_k=(k \bmod n)+1,$$

where $\boldsymbol{X}_{j_k}$ denotes the j_kth column of $\boldsymbol{X}$, $\boldsymbol{e}_{j_k}$ is the j_kth unit coordinate

vector and the superscript T denotes the transpose. The convergence of the Gauss-Seidel method highly depends on the order of the column selected in each step.

Inspired by the fancy work of Strohmer and Vershynin [1], Leventhal and Lewis [2] proposed the randomized Gauss-Seidel (RGS) method and proved that it has an expected linear convergence rate. As Bai and Wu [3] pointed out an obvious flaw of the RGS method that the probability for selecting column will be a uniform column sampling if the coefficient matrix is scaled with a diagonal matrix. To tackle this problem, they proposed the greedy randomized coordinate descent (GRCD) or called the greedy randomized Gauss-Seidel (GRGS) method by adopting an effective probability for selecting the working column to capture larger entries of the residual vector with respect to (2). They showed that the GRGS method is significantly superior to the RGS method in terms of both theoretical analysis and numerical experiments. In addition, there are tons of attention about the Gauss-Seidel type methods [4] [5] [6] [7]. However, the convergence rate of the RGS method will be significantly reduced when the coefficient matrix columns are highly correlated. To improve the convergence rate, Wang *et al.* [8] proposed the randomized Gauss-Seidel method with oblique direction (RGSO) by combining two successive selected unit coordinate directions as the search direction

$$\boldsymbol{d}_k = \boldsymbol{e}_{j_{k+1}} - \frac{\boldsymbol{X}_{j_k}^{\mathrm{T}} \boldsymbol{X}_{j_{k+1}}}{\left\| \boldsymbol{X}_{j_k} \right\|_2^2} \boldsymbol{e}_{j_k}. \tag{3}$$

They showed that, in terms of both theory and experiments, the RGSO method outperforms the RGS method. For more discussions about the oblique projection, we refer the readers to other literatures [9] [10] and their references.

However, the RGSO method still exists in the same flaw of the RGS method that the probability for selecting column will be a uniform column sampling if the coefficient matrix is scaled with a diagonal matrix. In addition, it is worth noting that the convergence rate of the GRGS method would significantly decrease when the coefficient matrix columns are close to linear correlation. For the above mentioned limitations, we present a greedy randomized Gauss-Seidel method with oblique direction (GRGSO) for solving (1), by combining the oblique direction and the GRGS method. In theory, it is proved that the iterative solutions generated by the GRGSO method can converge to the least squares solution $\boldsymbol{\beta}_*$ when the coefficient matrix is of full column rank. Numerical results show that compared with the RGSO and GRGS methods, the GRGSO method has a significant advantage over the iteration steps and computing time, especially when the coefficient matrix columns are highly correlated.

The organization of this paper is as follows. In Section 2, some notation and lemmas are introduced. In Section 3, we propose the GRGSO method for solving (1) and give its convergence analysis. Some examples are used to demonstrate the competitiveness of our proposed method in Section 4. Finally, we draw some brief conclusions in Section 5.

2. Notion and Preliminaries

In the beginning of this section, we give some notation. For a Hermitian positive definite matrix $\boldsymbol{B}$ and a column vector $\boldsymbol{\beta}$ with appropriate dimension, we denote $\|\boldsymbol{\beta}\|_{\boldsymbol{B}}^2 = \langle \boldsymbol{B}\boldsymbol{\beta}, \boldsymbol{\beta} \rangle = \left\| \boldsymbol{B}^{\frac{1}{2}} \boldsymbol{\beta} \right\|_2^2$ and $\boldsymbol{\beta}^{(i)}$ the ith entry of $\boldsymbol{\beta}$. For a given matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n}$, $\sigma_{\min}(\boldsymbol{S})$ and $\|\boldsymbol{S}\|_F$ denote the smallest nonzero singular value and the Frobenius norm of matrix $\boldsymbol{S}$, respectively. Let $\boldsymbol{r}_k = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_k$ and $\boldsymbol{s}_k = \boldsymbol{X}^{\mathrm{T}} \boldsymbol{r}_k$, then $\boldsymbol{s}_k^{(j)} = \boldsymbol{X}_j^{\mathrm{T}} \boldsymbol{r}_k$ represents jth entry of $\boldsymbol{s}_k$. $\boldsymbol{\beta}_*$ is the optimal solution of the corresponding problem. We indicate by $\mathbb{E}_k$ the expected value conditional on the first k iterations, that is,

$$\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot \mid j_0, j_1, \cdots, j_{k-1}],$$

where $j_t, t = 0, 1, \cdots, k-1$ is the column selected at the t-th iteration.

In the following, we give a basic lemma.

Lemma 1 (See Bai and Wu [11]) If the vector $\boldsymbol{u}$ is in the column space of $\boldsymbol{A}^{\mathrm{T}}$, it holds

$$\|\boldsymbol{A}\boldsymbol{u}\|_2^2 \geq \sigma_{\min}^2(\boldsymbol{A}) \|\boldsymbol{u}\|_2^2.$$

3. GRGSO Method

In this section, we design the GRGSO method for solving (1), by combining the oblique direction with the GRGS method. The pseudo-code of GRGSO method is listed in Table 1. The difference between the RGSO method and GRGSO method is the selection strategy. The RGSO method utilizes the random selection strategy. Specially, the RGSO method selects j_{k+1} th column with probability $\frac{\|\boldsymbol{X}_{j_{k+1}}\|_2^2}{\|\boldsymbol{X}\|_F^2}$ in the numerical experiments, which can be equivalent to the uniform sampling if the Euclidean norms of all the columns of the matrix $\boldsymbol{X}$ are same; while the GRGSO method aims to grasp the larger entries of the residual vector at each iteration. Compared with the GRGS method, our proposed method considers the oblique projection, which is expected to have better convergence performance in some cases of coefficient matrix columns being highly correlated.

Remark Let $t_k = \arg\max_{1 \leq j \leq n} \left\{ \frac{|\boldsymbol{s}_k^{(j)}|_2^2}{\|\boldsymbol{X}_j\|_2^2} \right\}$, which implies $t_k \in V_k$. Therefore, for all iterative step k, the index set V_k generated by the GRGSO method is nonempty.

Remark In the GRGSO method, it holds that

$$\begin{aligned} \boldsymbol{s}_{k+1} &= \boldsymbol{X}^{\mathrm{T}} \boldsymbol{r}_{k+1} = \boldsymbol{X}^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_{k+1}) \\ &= \boldsymbol{X}^{\mathrm{T}} \left(\boldsymbol{y} - \boldsymbol{X} \left(\boldsymbol{\beta}_k + \eta_k^{(j_k)} \boldsymbol{w}_{j_k} \right) \right) \\ &= \boldsymbol{X}^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_k) - \eta_k^{(j_k)} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{X} \boldsymbol{w}_{j_k} \\ &= \boldsymbol{s}_k - \eta_k^{(j_k)} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{X} \boldsymbol{w}_{j_k}, \quad (k \geq 1) \end{aligned} \tag{4}$$

Table 1. GRGSO method.

Input: $\boldsymbol{X} \in \mathbb{R}^{m\times n}, \boldsymbol{y} \in \mathbb{R}^{m}, \boldsymbol{\beta}_0 \in \mathbb{R}^{n}$ and l.

Output: Approximate $\boldsymbol{\beta}_l$ solving (1).

1. Randomly select $j_1 \in \{1,2,\cdots,n\}$ with probability $\dfrac{\left\|\boldsymbol{X}_{j_1}\right\|_2^2}{\left\|\boldsymbol{X}\right\|_F^2}$ and compute $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 + \dfrac{\boldsymbol{s}_0^{(j_1)}}{\left\|\boldsymbol{X}_{j_1}\right\|_2^2}\boldsymbol{e}_{j_1}$.

2. **For** $k=1$ to $l-1$ **do**

3. Compute $\delta_k = \dfrac{1}{2}\left(\dfrac{1}{\left\|\boldsymbol{s}_k\right\|_2^2}\max\limits_{1\le j\le n}\left\{\dfrac{\left|\boldsymbol{s}_k^{(j)}\right|^2}{\left\|\boldsymbol{X}_j\right\|_2^2}\right\} + \dfrac{1}{\left\|\boldsymbol{X}\right\|_F^2}\right)$.

4. Construct set $V_k = \left\{j : \left|\boldsymbol{s}_k^{(j)}\right|^2 \ge \delta_k \left\|\boldsymbol{s}_k\right\|_2^2 \left\|\boldsymbol{X}_j\right\|_2^2\right\}$.

5. Compute $\tilde{\boldsymbol{s}}_k^{(j)} = \begin{cases} \boldsymbol{s}_k^{(j)}, & \text{if } j \in V_k, \\ 0, & \text{otherwise.} \end{cases}$

6. Select $j_{k+1} \in V_k$ with probability $\dfrac{\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2}{\left\|\tilde{\boldsymbol{s}}_k\right\|_2^2}$.

7. Compute $\boldsymbol{w}_{j_k} = \boldsymbol{e}_{j_{k+1}} - \dfrac{\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\boldsymbol{e}_{j_k}$ and $h_{j_k} = \boldsymbol{X}_{j_{k+1}}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}_{j_k}$.

8. Set $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \eta_k^{(j_k)}\boldsymbol{w}_{j_k}$, where $\eta_k^{(j_k)} = \dfrac{\tilde{\boldsymbol{s}}_k^{(j_{k+1})}}{h_{j_k}}$.

9. **End for**

and

$$\begin{aligned} \boldsymbol{s}_1 &= \boldsymbol{X}^{\mathrm{T}}\boldsymbol{r}_1 = \boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_1\right) \\ &= \boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\left(\boldsymbol{\beta}_0 + \frac{\boldsymbol{s}_0^{(j_1)}}{\left\|\boldsymbol{X}_{j_1}\right\|_2^2}\boldsymbol{e}_{j_1}\right)\right) \\ &= \boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0\right) - \frac{\boldsymbol{s}_0^{(j_1)}}{\left\|\boldsymbol{X}_{j_1}\right\|_2^2}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}_{j_1} \\ &= \boldsymbol{s}_0 - \frac{\boldsymbol{s}_0^{(j_1)}}{\left\|\boldsymbol{X}_{j_1}\right\|_2^2}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}_{j_1}. \end{aligned} \tag{5}$$

Therefore, the GRGSO method can be executed more effectively if $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ can be computed in an economical manner at the beginning.

Next, we give some lemmas which are useful to analyze the convergence of the GRGSO method.

Lemma 2 For the GRGSO method, we have

$$\boldsymbol{s}_k^{(j_k)} = 0, \quad (\forall k > 0), \tag{6}$$

$$\boldsymbol{s}_k^{(j_{k-1})} = 0, \quad (\forall k > 1). \tag{7}$$

Proof. For $k = 1$, one has

$$\boldsymbol{s}_1^{(j_1)} = \boldsymbol{X}_{j_1}^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_1\right) = \boldsymbol{X}_{j_1}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{X}_{j_1}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_0 + \frac{\boldsymbol{s}_0^{(j_1)}}{\left\|\boldsymbol{X}_{j_1}\right\|_2^2}\boldsymbol{e}_{j_1}\right) = 0. \tag{8}$$

For $k > 1$, we have

$$\begin{aligned}\boldsymbol{s}_k^{(j_k)} &= \boldsymbol{X}_{j_k}^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_k\right) = \boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_{k-1} + \eta_{k-1}^{(j_{k-1})}\boldsymbol{w}_{j_{k-1}}\right)\\ &= \boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{\beta}_{k-1} - \eta_{k-1}^{(j_{k-1})}\left(\boldsymbol{X}_{j_k}^{T}\boldsymbol{X}\boldsymbol{w}_{j_{k-1}}\right) = \tilde{\boldsymbol{s}}_{k-1}^{(j_k)} - \frac{\tilde{\boldsymbol{s}}_{k-1}^{(j_k)}}{h_{j_{k-1}}}h_{j_{k-1}}\\ &= 0,\end{aligned}$$

which together with (8) proves (6).

By (6), it follows for $k > 0$ that

$$\begin{aligned}\boldsymbol{s}_{k+1}^{(j_k)} &= \boldsymbol{X}_{j_k}^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_{k+1}\right) = \boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_k + \eta_k^{(j_k)}\boldsymbol{w}_{j_k}\right)\\ &= \boldsymbol{s}_k^{(j_k)} - \eta_k^{(j_k)}\left(\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}_{j_k}\right)\\ &= -\eta_k^{(j_k)}\left(\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{e}_{j_{k+1}} - \frac{\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\boldsymbol{e}_{j_k}\right)\right)\\ &= 0,\end{aligned}$$

which leads to (7).

Remark From (6) and (7), it is obvious that in kth iteration, the GRGSO method dose not select j_k, j_{k-1}, which means $j_{k+1} \neq j_k, j_{k-1}$. Thus, the direction $\boldsymbol{w}_{j_k}$ can be the combination of two unit coordinate directions. This is also the advantage of the GRGSO method compared with the RGSO method. Since the RGSO method randomly selects j_{k+1} in the kth iteration, which can not avoid selecting j_k and j_{k-1} while the GRGSO method can avoid.

Lemma 3 For h_{j_k} in the GRGSO method, it satisfies

$$h_{j_k} = \left\|\boldsymbol{X}\boldsymbol{w}_{j_k}\right\|_2^2 = \left\|\boldsymbol{X}_{j_{k+1}}\right\|_2^2 - \frac{\left|\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}\right|^2}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2} \leq \Delta\left\|\boldsymbol{X}_{j_{k+1}}\right\|_2^2,$$

where $\Delta = \max_{i \neq j} \sin^2\left\langle \boldsymbol{X}_i, \boldsymbol{X}_j \right\rangle$.

Proof. Since $j_k \neq j_{k+1}$, it holds that

$$\begin{aligned}h_{j_k} &= \boldsymbol{X}_{j_{k+1}}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{e}_{j_{k+1}} - \frac{\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\boldsymbol{e}_{j_k}\right) = \left\|\boldsymbol{X}_{j_{k+1}}\right\|_2^2 - \frac{\left|\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}\right|^2}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\\ &= \left\|\boldsymbol{X}_{j_{k+1}}\right\|_2^2 - \frac{\left\|\boldsymbol{X}_{j_k}\right\|_2^2\left\|\boldsymbol{X}_{j_{k+1}}\right\|_2^2\cos^2\left\langle \boldsymbol{X}_{j_k}, \boldsymbol{X}_{j_{k+1}} \right\rangle}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\\ &= \sin^2\left\langle \boldsymbol{X}_{j_k}, \boldsymbol{X}_{j_{k+1}} \right\rangle\left\|\boldsymbol{X}_{j_{k+1}}\right\|_2^2 \leq \Delta\left\|\boldsymbol{X}_{j_{k+1}}\right\|_2^2\end{aligned}$$

and

$$\left\|\boldsymbol{X}\boldsymbol{w}_{j_k}\right\|_2^2=\left(\boldsymbol{e}_{j_{k+1}}-\frac{\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\boldsymbol{e}_{j_k}\right)^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{e}_{j_{k+1}}-\frac{\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\boldsymbol{e}_{j_k}\right)$$
$$=\left\|\boldsymbol{X}_{j_{k+1}}\right\|_2^2-\frac{\left|\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}\right|^2}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}.$$

Hence, we complete this proof.

Lemma 4 The iteration sequence $\{\boldsymbol{\beta}_k\}_{k=0}^{\infty}$ generated by the GRGSO method satisfies

$$\left\|\boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_*\right)\right\|_2^2=\left\|\boldsymbol{X}\left(\boldsymbol{\beta}_k-\boldsymbol{\beta}_*\right)\right\|_2^2-\left\|\boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k\right)\right\|_2^2,\quad k=0,1,2,\cdots.$$

Proof. For $k=0$, we have

$$\begin{aligned}\boldsymbol{e}_{j_1}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_1-\boldsymbol{\beta}_*\right)&=\boldsymbol{X}_{j_1}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_0-\boldsymbol{\beta}_*+\frac{\boldsymbol{s}_0^{(j_1)}}{\left\|\boldsymbol{X}_{j_1}\right\|_2^2}\boldsymbol{e}_{j_1}\right)\\&=\boldsymbol{X}_{j_1}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{\beta}_0-\boldsymbol{X}_{j_1}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{\beta}_*+\boldsymbol{s}_0^{(j_1)}\\&=0.\end{aligned}$$

This means that the vector $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_1-\boldsymbol{\beta}_*\right)$ is perpendicular to the vector $\boldsymbol{e}_{j_1}$. Since $\boldsymbol{\beta}_1-\boldsymbol{\beta}_0$ is parallel to $\boldsymbol{e}_{j_1}$, the vector $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_1-\boldsymbol{\beta}_*\right)$ is perpendicular to $\boldsymbol{\beta}_1-\boldsymbol{\beta}_0$.

For $k>0$, It follows from Lemma 3 and Lemma 2 that

$$\begin{aligned}\boldsymbol{w}_{j_k}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_*\right)&=\boldsymbol{w}_{j_k}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_k-\boldsymbol{\beta}_*+\eta_k^{(j_k)}\boldsymbol{w}_{j_k}\right)\\&=\boldsymbol{w}_{j_k}^{\mathrm{T}}\left(-\boldsymbol{s}_k+\eta_k^{(j_k)}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}_{j_k}\right)\\&=-\boldsymbol{w}_{j_k}^{\mathrm{T}}\boldsymbol{s}_k+\eta_k^{(j_k)}\left\|\boldsymbol{X}\boldsymbol{w}_{j_k}\right\|_2^2\\&=-\left(\boldsymbol{e}_{j_{k+1}}-\frac{\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\boldsymbol{e}_{j_k}\right)^{\mathrm{T}}\boldsymbol{s}_k+\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\\&=-\boldsymbol{s}_k^{(j_{k+1})}+\frac{\boldsymbol{X}_{j_k}^{\mathrm{T}}\boldsymbol{X}_{j_{k+1}}}{\left\|\boldsymbol{X}_{j_k}\right\|_2^2}\boldsymbol{s}_k^{(j_k)}+\tilde{\boldsymbol{s}}_k^{(j_{k+1})}=0,\end{aligned}$$

which means that the vector $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_*\right)$ is perpendicular to the vector $\boldsymbol{w}_{j_k}$. Since $\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k$ is parallel to $\boldsymbol{w}_{j_k}$, the vector $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_*\right)$ is perpendicular to $\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k$. For all $k=0,1,\cdots$, it follows that

$$\left\langle\boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_*\right),\boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k\right)\right\rangle=\left\langle\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_*\right),\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k\right\rangle=0,$$

which together with Pythagoras theorem leads to the desired result.

Next, the convergence theory of the GRGSO method is deduced.

Theorem 5 For the least squares problem (1), the iteration sequence $\{\boldsymbol{\beta}_k\}_{k=0}^{\infty}$ generated by the GRGSO method from any initial guess $\boldsymbol{\beta}_0$ satisfies

$$\mathbb{E}\left\|\boldsymbol{X}\left(\boldsymbol{\beta}_k-\boldsymbol{\beta}_*\right)\right\|_2^2 \le \prod_{t=0}^{k-1}\zeta_t\left\|\boldsymbol{X}\left(\boldsymbol{\beta}_0-\boldsymbol{\beta}_*\right)\right\|_2^2, \quad \text{for } k=1,2,\cdots, \tag{9}$$

where $\zeta_0=1-\dfrac{\sigma_{\min}^2(\boldsymbol{X})}{\|\boldsymbol{X}\|_F^2}$, $\zeta_1=1-\dfrac{1}{2}\left(\dfrac{1}{\gamma_1}+\dfrac{1}{\|\boldsymbol{X}\|_F^2}\right)\dfrac{\sigma_{\min}^2(\boldsymbol{X})}{\Delta}$,

$\zeta_t=1-\dfrac{1}{2}\left(\dfrac{1}{\gamma_2}+\dfrac{1}{\|\boldsymbol{X}\|_F^2}\right)\dfrac{\sigma_{\min}^2(\boldsymbol{X})}{\Delta}$ $(t>1)$. Here $\gamma_1=\max\limits_{1\le i\le n}\sum\limits_{\substack{j=1\\ j\ne i}}^{n}\|\boldsymbol{X}_j\|_2^2$,

$\gamma_2=\max\limits_{\substack{1\le i,\,j\le n\\ i\ne j}}\sum\limits_{\substack{t=1\\ t\ne i,\,j}}^{n}\|\boldsymbol{X}_t\|_2^2$ and Δ is given as in Lemma 3.

Proof. By Lemma 2, it follows for $k>1$ that

$$\begin{aligned}\delta_k\|\boldsymbol{X}\|_F^2 &= \frac{1}{2}\left(\frac{\|\boldsymbol{X}\|_F^2}{\|\boldsymbol{s}_k\|_2^2}\max_{1\le j\le n}\left\{\frac{\left|\boldsymbol{s}_k^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}\right\}+1\right)=\frac{1}{2}\left(\frac{\max\limits_{1\le j\le n}\left\{\dfrac{\left|\boldsymbol{s}_k^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}\right\}}{\sum\limits_{j=1}^{n}\dfrac{\|\boldsymbol{X}_j\|_2^2}{\|\boldsymbol{X}\|_F^2}\dfrac{\left|\boldsymbol{s}_k^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}}+1\right)\\ &=\frac{1}{2}\left(\frac{\max\limits_{1\le j\le n}\left\{\dfrac{\left|\boldsymbol{s}_k^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}\right\}}{\sum\limits_{\substack{j=1\\ j\ne j_k,\,j_{k-1}}}^{n}\dfrac{\|\boldsymbol{X}_j\|_2^2}{\|\boldsymbol{X}\|_F^2}\dfrac{\left|\boldsymbol{s}_k^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}}+1\right)\ge\frac{1}{2}\left(\frac{\|\boldsymbol{X}\|_F^2}{\sum\limits_{\substack{j=1\\ j\ne j_k,\,j_{k-1}}}^{n}\|\boldsymbol{X}_j\|_2^2}+1\right)\\ &\ge\frac{1}{2}\left(\frac{\|\boldsymbol{X}\|_F^2}{\gamma_2}+1\right).\end{aligned} \tag{10}$$

For $k=1$, it follows from (6) that

$$\begin{aligned}\delta_1\|\boldsymbol{X}\|_F^2 &= \frac{1}{2}\left(\frac{\|\boldsymbol{X}\|_F^2}{\|\boldsymbol{s}_1\|_2^2}\max_{1\le j\le n}\left\{\frac{\left|\boldsymbol{s}_1^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}\right\}+1\right)=\frac{1}{2}\left(\frac{\max\limits_{1\le j\le n}\left\{\dfrac{\left|\boldsymbol{s}_1^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}\right\}}{\sum\limits_{j=1}^{n}\dfrac{\|\boldsymbol{X}_j\|_2^2}{\|\boldsymbol{X}\|_F^2}\dfrac{\left|\boldsymbol{s}_1^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}}+1\right)\\ &=\frac{1}{2}\left(\frac{\max\limits_{1\le j\le n}\left\{\dfrac{\left|\boldsymbol{s}_1^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}\right\}}{\sum\limits_{\substack{j=1\\ j\ne j_1}}^{n}\dfrac{\|\boldsymbol{X}_j\|_2^2}{\|\boldsymbol{X}\|_F^2}\dfrac{\left|\boldsymbol{s}_1^{(j)}\right|^2}{\|\boldsymbol{X}_j\|_2^2}}+1\right)\ge\frac{1}{2}\left(\frac{\|\boldsymbol{X}\|_F^2}{\sum\limits_{\substack{j=1\\ j\ne j_1}}^{n}\|\boldsymbol{X}_j\|_2^2}+1\right)\\ &\ge\frac{1}{2}\left(\frac{\|\boldsymbol{X}\|_F^2}{\gamma_1}+1\right).\end{aligned} \tag{11}$$

By Lemma 4, Lemma 3 and Lemma 1, for $k\ge 1$, we have

$$
\begin{aligned}
&\mathbb{E}_k \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_*\right)\right\|_2^2 \\
&= \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2 - \mathbb{E}_k \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k\right)\right\|_2^2 \\
&= \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2 - \sum_{j_{k+1}\in V_k} \frac{\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2}{\sum\limits_{j_{k+1}\in V_k}\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2} \left\| \eta_k^{(j_k)}\left(\boldsymbol{X}\boldsymbol{w}_{j_k}\right)\right\|^2 \\
&= \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2 - \sum_{j_{k+1}\in V_k} \frac{\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2}{\sum\limits_{j_{k+1}\in V_k}\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2} \frac{\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2 \left\| \boldsymbol{X}\boldsymbol{w}_{j_k}\right\|_2^2}{\left|h_{j_k}\right|^2} \\
&= \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2 - \sum_{j_{k+1}\in V_k} \frac{\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2}{\sum\limits_{j_{k+1}\in V_k}\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2} \frac{\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2}{h_{j_k}} \\
&\le \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2 - \sum_{j_{k+1}\in V_k} \frac{\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2}{\sum\limits_{j_{k+1}\in V_k}\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2} \frac{\left|\tilde{\boldsymbol{s}}_k^{(j_{k+1})}\right|^2}{\Delta\left\| \boldsymbol{X}_{j_{k+1}}\right\|_2^2} \\
&\le \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2 - \frac{\delta_k}{\Delta}\left\| \boldsymbol{s}_k\right\|_2^2 \\
&= \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2 - \frac{\delta_k}{\Delta}\left\| \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2 \\
&\le \left(1-\delta_k\frac{\sigma_{\min}^2\left(\boldsymbol{X}\right)}{\Delta}\right)\left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2,
\end{aligned}
$$

which together with (11) and (10) can lead to

$$\mathbb{E}_1 \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{2}-\boldsymbol{\beta}_*\right)\right\|_2^2 \le \zeta_1 \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{1}-\boldsymbol{\beta}_*\right)\right\|_2^2 \tag{12}$$

and

$$\mathbb{E}_k \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_*\right)\right\|_2^2 \le \zeta \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{k}-\boldsymbol{\beta}_*\right)\right\|_2^2, \quad k>1, \tag{13}$$

respectively. For $k=0$, we can similarly get

$$\mathbb{E}\left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{1}-\boldsymbol{\beta}_*\right)\right\|_2^2 \le \zeta_0 \left\| \boldsymbol{X}\left(\boldsymbol{\beta}_{0}-\boldsymbol{\beta}_*\right)\right\|_2^2.$$

Then taking the full expectation on both sides of (12) and (13) and by induction on the iteration index *k*, we can easily obtain (9). Thus, we complete the proof.

Remark Suppose that the upper bound for convergence rate of the GRGS method [3] is defined as

$$\rho_{GRCD} = 1-\frac{1}{2}\left(\frac{1}{\gamma_1}+\frac{1}{\left\|\boldsymbol{X}\right\|_F^2}\right)\sigma_{\min}^2\left(\boldsymbol{X}\right).$$

Since $\Delta = \max\limits_{i\neq j}\sin^2\left\langle \boldsymbol{X}_i, \boldsymbol{X}_j\right\rangle \in (0,1]$, $\gamma_2<\gamma_1<\left\|\boldsymbol{X}\right\|_F^2$ and $0<\sigma_{\min}^2\left(\boldsymbol{X}\right)\le\left\|\boldsymbol{X}\right\|_F^2$, we have

$$\zeta<\zeta_1\le\rho_{GRCD}<\zeta_0<1.$$

This implies that the upper bound on the convergence factor of the GRGSO method is smaller, uniformly with respect to the iterative step k, than that of the GRCD method.

4. Numerical Experiments

Some examples are designed in this section to verify the effectiveness of the GRGSO method. Specially, the GRGSO method is compared with the GRGS method [3], RGS method [2] and RGSO method [8]. We list the tested results of these methods in terms of the number of iteration steps (denoted by "IT") and the running time in seconds (denoted by "CPU").

The coefficient matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}\left(m \geq n\right)$ in numerical experiments is from two sources. One is the random matrix whose entries are randomly taken from the interval $\left[c,1\right]$ $\left(c \geq 0\right)$ by using the function *rand* in MATLAB. c being close to 1 implies that the matrix columns are highly correlated. Another is sparse matrices from the literature [12] listed in **Table 2**. We use $\mathrm{cond}\left(\boldsymbol{X}\right)$ to represent the condition number for $\boldsymbol{X}$, and define the density as

$$\text{density} = \frac{\text{number of nonzero entries of an } m\times n \text{ matrix}}{m\times n}.$$

We randomly generate the true vector $\boldsymbol{\beta}_*$ by utilizing the MATLAB function *randn* and construct the vector y by $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}_*$ when the linear system is consistent; while for the inconsistent linear systems the right-hand side is set by $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}_* + \boldsymbol{noise}$, where $\boldsymbol{noise} \in null\left(\boldsymbol{X}^{\mathrm{T}}\right)$. We take zero vector for the initial approximation of each iteration process. Since

$$\left\|\boldsymbol{X}\left(\boldsymbol{\beta}_k - \boldsymbol{\beta}_*\right)\right\|_2^2 = \left\|\boldsymbol{X}\boldsymbol{\beta}_* + \boldsymbol{noise} - \boldsymbol{r}_k - \boldsymbol{X}\boldsymbol{\beta}_*\right\|_2^2 = \left\|\boldsymbol{noise} - \boldsymbol{r}_k\right\|_2^2,$$

which was also used in the work of Wang *et al.* [8], we terminate the iteration process once

$$RSE = \frac{\left\|\boldsymbol{noise} - \boldsymbol{r}_k\right\|_2}{\left\|\boldsymbol{y}\right\|_2} < 10^{-6}.$$

We set "-" in the numerical tables if the number of iteration steps exceeds 300,000. All the results are averages from 20 repetitions. All experiments were implemented by using MATLAB (R2021b) on a computer with 2.30 GHz central processing unit (Intel(R) Core(TM) i7-10875H CPU), 16 GB memory.

Table 2. Sparse matrix properties of realistic problems [12].

Name	$m \times n$	density	cond($\boldsymbol{X}$)
abtaha1	14,596 × 209	1.68%	12.23
Cities	55 × 46	53.04%	207.17
WorldCities	315 × 100	23.87%	66.00
ash219	219 × 85	2.35%	3.02
divorce	50 × 9	50.00%	19.39

For the randomly generated matrices, with $c = 0$, the numerical results for consistent and inconsistent linear systems are listed in Table 3 and Table 4, respectively. It is easy to observe from Table 3 and Table 4 that the GRGSO method significantly outperforms the RGS, GRGS and RGSO methods in terms of both IT and CPU.

In the following, we compare these methods for solving (1) when the randomly generated matrix is with different c. We list the numerical results for the consistent and inconsistent systems in Table 5 and Table 6, respectively. From Table 5 and Table 6, it is easy to observe that the IT and CPU of the RGS and GRGS methods increase significantly with c increasing closer to 1. When c increases to 0.8, the IT of the RGS method exceeds the maximal number of iteration steps. And when c increases to 0.9, the IT of the GRGS method exceeds the maximal number of iteration steps. For the different c values, the methods with the oblique direction outperform the methods without the oblique direction. In addition, compared with the other methods, the GRGSO method performs best in terms of both IT and CPU.

For the full-rank sparse matrices from literature [12], the numerical results for the consistent and inconsistent linear systems are listed in Table 7 and Table 8, respectively. It can be seen that for the matrix abtaha1 the performance of the GRGSO method is similar to that of the GRGS method in terms of both IT and CPU, whereas for the other matrices the GRGSO method significantly performs better in terms of both IT and CPU than the other methods.

Table 3. The consistent system with $c = 0$: different m impacts on IT and CPU.

Method	$m \times n$	1000×100	2000×100	3000×100	4000×100	5000×100
RGS	IT	9650	8082	7113	7378	7209
	CPU	0.1919	0.1858	0.1859	0.2716	0.2953
GRGS	IT	3271	2753	2266	2581	2538
	CPU	0.0708	0.0664	0.0622	0.1003	0.1078
RGSO	IT	3914	3553	3243	3509	3499
	CPU	0.0921	0.0980	0.1092	0.1739	0.2113
GRGSO	IT	755	761	739	828	860
	CPU	0.0190	0.0210	0.0246	0.0412	0.0536

Table 4. The inconsistent system with $c = 0$: different m impacts on IT and CPU.

Method	$m \times n$	1000×100	2000×100	3000×100	4000×100	5000×100
RGS	IT	9499	7926	7310	7287	6910
	CPU	0.1958	0.1821	0.1927	0.3039	0.2766
GRGS	IT	2812	2550	2530	2547	2390
	CPU	0.0648	0.0610	0.0715	0.1113	0.1018
RGSO	IT	3715	3519	3464	3525	3285
	CPU	0.0890	0.0985	0.1187	0.1952	0.1966
GRGSO	IT	706	753	787	829	800
	CPU	0.0171	0.0201	0.0269	0.0472	0.0488

Table 5. The consistent system with $\boldsymbol{X} \in \mathbb{R}^{1000\times 100}$: different c impacts on IT and CPU.

Method	c	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
RGS	IT	13902	17612	26137	42060	66402	98900	237339	-	-
	CPU	0.2800	0.3624	0.5472	0.9380	1.4841	2.2364	5.2936	-	-
GRGS	IT	4096	5436	8152	11374	22992	28719	75396	154734	-
	CPU	0.0902	0.1223	0.1814	0.2748	0.5540	0.6953	1.8127	3.7762	-
RGSO	IT	4103	3864	3934	4266	4264	3680	4356	3860	3445
	CPU	0.0985	0.0952	0.0977	0.1127	0.1115	0.0972	0.1158	0.1031	0.0884
GRGSO	IT	765	740	774	800	816	704	834	761	681
	CPU	0.0203	0.0186	0.0197	0.0213	0.0220	0.0196	0.0226	0.0203	0.0184

Table 6. The inconsistent system with $\boldsymbol{X} \in \mathbb{R}^{1000\times 100}$: different c impacts on IT and CPU.

Method	c	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
RGS	IT	13212	17463	25396	36517	59507	113307	197337	-	-
	CPU	0.2773	0.3815	0.5726	0.8210	1.3082	2.5499	4.5799	-	-
GRGS	IT	4373	5373	7890	11197	19498	33295	60924	147676	-
	CPU	0.1006	0.1272	0.1917	0.2680	0.4583	0.8193	1.5151	3.5326	-
RGSO	IT	3949	3740	3997	3692	3936	4185	3729	3490	3266
	CPU	0.0995	0.0998	0.1075	0.0980	0.1027	0.1145	0.1035	0.0929	0.0890
GRGSO	IT	745	716	769	731	762	815	711	691	632
	CPU	0.0197	0.0193	0.0216	0.0198	0.0207	0.0221	0.0201	0.0183	0.0177

Table 7. The consistent system: IT and CPU time of test methods for different sparse matrices.

Method	Name	abtaha1	Cities	WorldCities	ash219	divorce
RGS	IT	167141	-	76327	3785	5312
	CPU	16.6240	-	1.4365	0.0665	0.0581
GRGS	IT	27246	96279	8069	654	1025
	CPU	2.8197	1.3852	0.1640	0.0129	0.0120
RGSO	IT	164535	245186	56391	3598	244
	CPU	22.0450	3.6255	1.2173	0.0720	0.0031
GRGSO	IT	26670	25115	3163	609	79
	CPU	3.5948	0.3780	0.0695	0.0122	0.0010

Table 8. The inconsistent system: IT and CPU time of test methods for different sparse matrices.

Method	Name	abtaha1	Cities	WorldCities	ash219	divorce
RGS	IT	166432	297734	70098	3502	5554
	CPU	16.9889	3.9135	1.3268	0.0645	0.0608
GRGS	IT	24268	92576	8204	676	959
	CPU	2.5255	1.3239	0.1729	0.0140	0.0113
RGSO	IT	168963	220109	52808	3464	260
	CPU	22.9293	3.2584	1.1308	0.0723	0.0033
GRGSO	IT	23454	23230	3216	632	81
	CPU	3.2018	0.3560	0.0732	0.0145	0.0010

5. Conclusion

In this manuscript, we construct the GRGSO method for the linear least squares problem. We have established the convergence analyses of the GRGSO method. Numerical experiments show that the GRGSO method is superior to the RGS, GRGS and RGSO methods in terms of both IT and CPU, especially when the coefficient matrix columns are highly correlated. It is natural to generalize the GRGSO method by introducing a relaxation parameter in its probability criterion. However, the choice of the optimal relaxation parameter is difficult in theory up to now. How to find the optimal relaxation parameter in theory is worthy of further study.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Strohmer, T. and Vershynin, R. (2009) A Randomized Kaczmarz Algorithm with Exponential Convergence. *Journal of Fourier Analysis and Applications*, **15**, 262-278. https://doi.org/10.1007/s00041-008-9030-4

[2] Leventhal, D. and Lewis, A.S. (2010) Randomized Methods for Linear Constraints: Convergence Rates and Conditioning. *Mathematics of Operations Research*, **35**, 641-654. https://doi.org/10.1287/moor.1100.0456

[3] Bai, Z.Z. and Wu, W.T. (2019) On Greedy Randomized Coordinate Descent Methods for Solving Large Linear Least-Squares Problems. *Numerical Linear Algebra with Applications*, **26**, e2237. https://doi.org/10.1002/nla.2237

[4] Du, K. (2019) Tight Upper Bounds for the Convergence of the Randomized Extended Kaczmarz and Gauss-Seidel Algorithms. *Numerical Linear Algebra with Applications*, **26**, e2233. https://doi.org/10.1002/nla.2233

[5] Liu, Y., Jiang, X.L. and Gu, C.Q. (2021) On Maximum Residual Block and Two-Step Gauss-Seidel Algorithms for Linear Least-Squares Problems. *Calcolo*, **58**, Article No. 13. https://doi.org/10.1007/s10092-021-00404-x

[6] Zhang, J.H. and Guo, J.H. (2020) On Relaxed Greedy Randomized Coordinate Descent Methods for Solving Large Linear Least-Squares Problems. *Applied Numerical Mathematics*, **157**, 372-384. https://doi.org/10.1016/j.apnum.2020.06.014

[7] Niu, Y.Q. and Zheng, B. (2021) A New Randomzied Gauss-Seidel Method for Solving Linear Least-Squares Problems. *Applied Mathematics Letters*, **116**, Article ID: 107057. https://doi.org/10.1016/j.aml.2021.107057

[8] Wang, F., Li, W.G., Bao, W.D. and Lv, Z.L. (2021) Gauss-Seidel Method with Oblique Direction. *Results in Applied Mathematics*, **12**, Article ID: 100180. https://doi.org/10.1016/j.rinam.2021.100180

[9] Li, W.G., Wang, Q., Bao, W.D. and Xing, L.L. (2022) Kaczmarz Method with Oblique Projection. *Results in Applied Mathematics*, **16**, Article ID: 100342. https://doi.org/10.1016/j.rinam.2022.100342

[10] Wang, F., Li, W.G., Bao, W.D. and Liu, L. (2022) Greedy Randomized and Maximal Weighted Residual Kaczmarz Methods with Oblique Projection. *Electronic Research Archive*, **30**, 1158-1186. https://doi.org/10.3934/era.2022062

[11] Bai, Z.Z. and Wu, W.T. (2018) On Greedy Randomized Kaczmarz Method for Solving Large Sparse Linear Systems. *SIAM Journal on Scientific Computing*, **40**, A592-A606. https://doi.org/10.1137/17M1137747

[12] Davis, T.A. and Hu, Y. (2011) The University of Florida Sparse Matrix Collection. *ACM Transactions on Mathematical Software*, **38**, 1-25. https://doi.org/10.1145/2049662.2049663