Randomized orthogonalization process with reorthogonalization

Yongseok Jang, Laura Grigori

Randomized orthogonalization process with reorthogonalization

A PREPRINT

Yongseok Jang[1,*] and Laura Grigori[2]

[1]LIP6, Sorbonne University, Paris, France. Email: yongseok.jang@lip6.fr.
[2]EPFL, Lausanne, Switzerland. Email: laura.grigori@epfl.ch.
[*]Corresponding author.

ABSTRACT

The dimension reduction technique of random sketching is advantageous in significantly reducing computational complexity. In orthogonalization processes like the *Gram-Schmidt* (GS) algorithm, incorporating random sketching results in a halving of computational costs compared to the *classical/modified Gram-Schmidt* (CGS/MGS) algorithms, while maintaining numerical stability comparable to the MGS algorithm. The *randomized Gram-Schmidt* (RGS) algorithm produces a set of sketched orthonormal vectors, and the loss of orthogonality in these vectors is linearly dependent on the condition number of the given matrix. We propose a new variant, RGS2, with reorthogonalization to obtain a set of l_2 orthonormal vectors. A round-off error analysis demonstrates that the loss of orthogonality is close to the unit round-off level. Numerical experiments exhibit the benefits of our proposed algorithm. Furthermore, we apply the RGS2 algorithm to the *Generalized Minimal Residual Method* (GMRES) and compare its numerical performance with other GMRES variants.

Keywords: Randomized Gram-Schmidt process, Reorthogonalization algorithm, Random sketching, Round-off error analysis, Randomized GMRES

MSC:15A06, 65F10, 65F15, 65N22, 68W20

1 Introduction

Random sketching is a technique used in numerical linear algebra to reduce the dimensionality of large data sets. This is achieved by multiplying the original matrix or vector with a random skteching matrix. Random sketching can be used to speed up computations and reduce the memory requirements of algorithms that operate on large matrices and vectors. Defining a suitable random sketching matrix is an essential aspect of this technique.

In practice, several variants of random sketching have been proposed and analyzed, such as Gaussian distributions, Rademacher distributions, and sub-sampled randomized Hadamard transform (SRHT). The quality of the approximation depends on the choice of sketching matrix. For more details on theoretical estimations and specific examples of suitable sketching matrices, we refer to literature for example, the works of Achlioptas [1], Halko et al.[2], and Woodruff [3]. Additionally, it is important to note that in order to ensure the effectiveness of the dimensionality reduction, the sketching matrix should have certain properties such as $\epsilon-$embedding and (ϵ, δ, d) oblivious l_2-subspace embedding. The size of the sketching matrix should also be chosen accordingly. For more details on these properties and specific examples of suitable sketching matrix sizes, we refer to literature such as Balabanov and Nouy[4], and Balabanov and Grigori[5].

Recently, Jang et al.[6] proposed new *Generalized Minimal Residual* (GMRES) methods that combine random sketching and deflated restarting. These methods exhibit improved stability in generating Krylov basis vectors and enhanced convergence rates by disregarding eigenvectors and singular vectors obtained via the randomized Rayleigh Ritz method. Since these methods rely on the *randomized Gram-Schmidt* (RGS) algorithm[5], the set of basis vectors is not l_2 orthonormal but sketched orthonormal. On the other hand, Nakatsukasa and Tropp[7] introduced a random sketching algorithm for GMRES and Rayleigh-Ritz methods, called *sGMRES* and *sRR*, respectively. Their works employed truncated Arnoldi process rather than full orthogonal-

ization, leading to faster computation but weaker stability. Furthermore, based on the truncated Arnoldi process, Burke and Güttel[8] developed another variant of the randomized GMRES with deflated restarting (GMRES-DR).

In this paper, we propose new randomized Gram-Schmidt algorithms inspired by reorthogonalization[9, 10]. Unlike the existing RGS algorithm, our method provides a set of l_2 orthonormal basis vectors with numerical stability. Additionally, we present a round-off error analysis to demonstrate that the loss of orthogonality is close to the unit round-off level. Finally, we apply the proposed algorithm to GMRES and GMRES-DR to compare numerical performance with the classical methods. Relevant MATLAB codes for the randomized Gram-Schmidt (RGS) algorithms and associated numerical experiments can be found on the author's GitHub repository (`https://github.com/Yongseok7717/RGS2`). These include implementations of various GS algorithms, along with scripts for GMRES and GMRES-DR methods.

We would like to highlight that our research provides fully l_2 orthogonalization with random sketching at first with relevant error analysis. We show that our method is computationally cheaper than other GS algorithms with reorthogonalization and the loss of orthogonality does not depend on the condition number of the input matrix. As seen in [6], to solve ill-conditioned systems, GMRES and GMRES-DR would not work properly if the quality of Kylov basis vectors is poor in finite arithmetic, i.e. if there exists the large loss of orthogonality. Therefore, since our algorithm ensures the quality of orthogonalization process, we can observe significantly improved numerical performance. For a summary of our work, Table 1 describes the comparison with other existing algorithms.

GS type	Numerical nonsingularity condition	Loss of orthogonality	Computational complexity	References
CGS	$\mathfrak{p}(m,n)\bar{u}\kappa(A)^2 < 1$	$\left\|I - V^TV\right\| \leq \mathfrak{p}(m,n)\bar{u}\kappa(A)^2$	$2nm^2$ *flops*	[11]
CGS2	$\mathfrak{p}(m,n)\bar{u}\kappa(A) < 1$	$\left\|I - V^TV\right\| \leq \mathfrak{p}(m,n)\bar{u}$	$4nm^2$ *flops*	[9]
MGS	$\mathfrak{p}(m,n)\bar{u}\kappa(A) < 1$	$\left\|I - V^TV\right\| \leq \mathfrak{p}(m,n)\bar{u}\kappa(A)$	$2nm^2$ *flops*	[12]
MGS2	$\mathfrak{p}(m,n)\bar{u}\kappa(A) < 1$	$\left\|I - V^TV\right\| \leq \mathfrak{p}(m,n)\bar{u}$	$4nm^2$ *flops*	[13]
RGS	$\mathfrak{p}(m)\bar{u}\kappa(A) < 1$	$\left\|I - S^TS\right\| \leq \mathfrak{p}(m,n)\bar{u}\kappa(A)$	nm^2 *flops*	[5]
RGS-L2	$\mathfrak{p}(m,n)\bar{u}\kappa(A) < 1$	$\left\|I - V^TV\right\| \leq \mathfrak{p}(m,n)\bar{u}$	$3nm^2$ *flops*	This paper

Table 1: Comparison of Gram-Schmidt algorithms applying to Arnoldi process with $n \times n$ matrix A for m dimensional Krylov subspace generating the orthogonal factor V (and the sketched orthogonal factor S for RGS), where $\mathfrak{p}$ is a some low degree polynomial, $\bar{u}$ is the unit round error and $\kappa(A)$ is the condition number of A.

Our paper is structured as follows. Section 2 provides an introduction to random sketching. In Section 3, we present two variants of the randomized Gram-Schmidt (RGS) algorithm with reorthogonalization. Section 4 includes rounding error analyses of our proposed algorithms along with numerical examples. In Section 5, we demonstrate the benefits of our method through numerical tests on solving linear systems. Finally, Section 6 concludes our findings with remarks.

2 Preliminary

We adopt standard notations for their broad applicability. Our focus is on real-valued systems, using bold lowercase for vectors and uppercase for matrices. For example, a vector of length n is denoted as $\boldsymbol{x} \in \mathbb{R}^n$ for $n \in \mathbb{N}$. The l_2 norm of a vector $\boldsymbol{x}$, computed by using the l_2 inner product $(\cdot,\cdot)$, is denoted as $\|\boldsymbol{x}\|$. The transpose of matrix X is represented as X^T, and the pseudo-inverse of X is denoted as $X^\dagger$. Additionally, $\kappa(X)$ denotes the condition number of X. We use $\sigma_{\min}(X)$ and $\sigma_{\max}(X)$ for the minimum and maximum singular values of X, respectively. The identity matrix of dimension n is denoted by I_n, and $O_{i\times j}$ represents the zero rectangular matrix of size i by j. For improved algorithm readability, we utilize MATLAB expressions; for example, $X(1:i, 1:j)$ denotes the submatrix comprising the first i rows and first j columns of X. For rounding error analysis, we denote the unit round-off by $\bar{u}$ (e.g. $\bar{u} = O(10^{-16})$ for IEEE double precision) and a low-degree polynomial by ξ_k for $k = 1, 2, \ldots$ such that depends on the problem dimensions but is independent of the condition number of the problem and $\bar{u}$.

Let $\Theta \in \mathbb{R}^{t\times n}$ be a sketching matrix with $t \ll n$. This matrix introduces a sketched product and its associated norm, referred to as the Θ-norm or sketched norm. These are employed to approximate the l_2 inner product and l_2 norm through the embedding of subspaces. Specifically, the random sketched product and sketched norm are defined as follows:

$$(\boldsymbol{v},\boldsymbol{w})_\Theta = (\Theta\boldsymbol{v},\Theta\boldsymbol{w}) \qquad \text{and} \qquad \|\boldsymbol{v}\|_\Theta = \|\Theta\boldsymbol{v}\|,$$

for any $\boldsymbol{v},\boldsymbol{w} \in \mathbb{R}^n$, respectively. This results in a l_2-subspace embedding, mapping subspaces of $\mathbb{R}^n$ into subspaces of $\mathbb{R}^t$. The effectiveness of this embedding depends on the choice of the sketching matrix Θ. For detailed theoretical insights, we refer to [5]. Hence, it is crucial to define a suitable sketching matrix, for example, using Gaussian distributions, Rademacher distributions, SRHT, etc., as discussed in [1, 2, 3].

Definition 1 ($\epsilon-$embedding)**.** *For $0 < \epsilon < 1$, the sketching matrix Θ is an $\epsilon-$embedding for subspace $V \subset \mathbb{R}^n$ if*

$$\forall \boldsymbol{x},\boldsymbol{y} \in V, \qquad |(\boldsymbol{x},\boldsymbol{y}) - (\boldsymbol{x},\boldsymbol{y})_\Theta| \leq \epsilon \|\boldsymbol{x}\| \|\boldsymbol{y}\|.$$

Definition 2 ((ϵ, δ, d) oblivious l_2-subspace embedding)**.** *The random sketching matrix Θ is a (ϵ, δ, d) oblivious l_2-subspace embedding for $V \subset \mathbb{R}^n$ of dimension d if*

$$\text{probability}(\Theta \textit{ is an } \epsilon-\textit{embedding for } V) \geq 1 - \delta.$$

According to [4, 5], we can define the random sketching matrix by rescaled Rademacher distribution and partial SRHT (P-SRHT) with the size of sketching matrix t given by

$$t \geq 7.87\epsilon^{-2}(6.9d + \log(1/\delta)) \qquad \text{and} \qquad t \geq 2(\epsilon^2 - \epsilon^3/3)^{-1}\left(\sqrt{d} + \sqrt{8\log(6n/\delta)}\right)^2 \log(3d/\delta),$$

respectively, to obtain (ϵ, δ, d) oblivious l_2-subspace embeddings.

Proposition 1 (l_2 norm and Θ norm)**.** *For a vector $\boldsymbol{v} \in \mathbb{R}^n$, we have the following bounds.*

- $\|\boldsymbol{v}\|_\Theta = \|\Theta\boldsymbol{v}\| \leq \|\Theta\| \, \|\boldsymbol{v}\|$.
- $\|\boldsymbol{v}\| = \left\|\Theta^\dagger\Theta\boldsymbol{v}\right\| \leq \left\|\Theta^\dagger\right\| \|\boldsymbol{v}\|_\Theta$ *if* $\Theta\boldsymbol{v} \neq \boldsymbol{0}$.
- $|\, \|\boldsymbol{v}\| - \|\boldsymbol{v}\|_\Theta \,| \leq \epsilon \, \|\boldsymbol{v}\|$ *if Θ is an ϵ-embedding for $V \subset \mathbb{R}^n$ and $\boldsymbol{v} \in V$.*
- $\|\Theta\|_F \leq \sqrt{(1+\epsilon)n}$ *with probability at least $1-\delta$ if Θ is an $(\epsilon, \delta/n, 1)$ oblivious l_2-subspace embedding, where $\|\cdot\|_F$ is the Frobenius norm.*
- *For $\Theta\boldsymbol{v} \neq \boldsymbol{0}$,*

$$\frac{1}{\|\Theta^\dagger\|} \|\boldsymbol{v}\| \leq \|\boldsymbol{v}\|_\Theta \leq \|\Theta\| \, \|\boldsymbol{v}\| \qquad \textit{and} \qquad \sigma_{\min}(\Theta) \, \|\boldsymbol{v}\| \leq \|\boldsymbol{v}\|_\Theta \leq \|\Theta\| \, \|\boldsymbol{v}\| .$$

We refer to [5, 6] for more details.

In order to obtain the benefit of random sketching in cost reduction, we introduce fast computations in sketching by *Fast Walsh Hadamard transformation* (FWHT) as follows.

Definition 3 (Fast Walsh Hadamard transformation)**.** *Let $\mathcal{H}_\mathfrak{m}$ be the $2^\mathfrak{m} \times 2^\mathfrak{m}$ Hadamard matrix defined by*

$$\mathcal{H}_\mathfrak{m} = \begin{bmatrix} \mathcal{H}_{\mathfrak{m}-1} & \mathcal{H}_{\mathfrak{m}-1} \\ \mathcal{H}_{\mathfrak{m}-1} & -\mathcal{H}_{\mathfrak{m}-1} \end{bmatrix},$$

for $\mathfrak{m} \in \mathbb{N}$ with $\mathcal{H}_0 = 1$. For a vector $\boldsymbol{a}$ of length $N = 2^\mathfrak{m}$, if we use the FWHT algorithm as depicted in Algorithm 1 to compute Walsh Hadamard transformation of $\boldsymbol{a}$, the complexity follows $N\log(N)$ flops rather than N^2 flops.

Algorithm 1 Fast Walsh Hadamard transformation

Input: vector $\boldsymbol{a} \in \mathbb{R}^N$
Output: $\boldsymbol{b} \in \mathbb{R}^N$

```
h = 1; b = a.
while h < N do
    for i = 1 : 2h : N do
        for j = i : (i + h − 1) do
            x = b(j); y = b(j + h).
            b(j) = x + y; b(j + h) = x − y.
        end for
    end for
    h = 2h
end while
return b.
```

Using the Hadamard matrix, we can define the sketch matrix of P-SRHT by

$$\Theta = \frac{1}{\sqrt{t}} P\mathcal{H}_\mathfrak{m} D, \tag{2.1}$$

where D is a random $N \times n$ rectangular diagonal matrix whose entries are independent random signs, $\mathfrak{m}$ satisfies $\log_2(n) \leq \log_2(N) = \mathfrak{m} < \log_2(n) + 1$ and P is a $t \times N$ random permutation matrix that restricts an $n-$dimensional vector to t coordinates. Therefore, the computational cost of sketching is $n\log(t)$ *flops*. For more details, please see [4] and the references therein.

3 Random sketching in Gram-Schmidt process

The Gram-Schmidt process is primarily employed for orthonormalizing a set of vectors or computing the QR factorization of a matrix. It operates iteratively, utilizing projection operators to achieve the desired orthogonalization. Let us consider the linearly independent $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m]$ and the projection operator $P_{\boldsymbol{u}}$ defined by

$$P_{\boldsymbol{u}}(\boldsymbol{w}) = \frac{(\boldsymbol{u}, \boldsymbol{w})}{(\boldsymbol{u}, \boldsymbol{u})}\boldsymbol{u} \qquad \text{for any } \boldsymbol{u} \text{ and } \boldsymbol{w}.$$

Then Gram-Schmidt process leads a set of orthogonal vectors $U_m = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$ where

$$\boldsymbol{u}_1 = \boldsymbol{w}_1, \qquad \boldsymbol{u}_j = \boldsymbol{w}_j - \sum_{i=1}^{j-1} P_{\boldsymbol{u}_i}(\boldsymbol{w}_j) \qquad \text{for } j = 2, \ldots, m, \qquad \text{and span}\{\boldsymbol{w}_1, \ldots \boldsymbol{w}_m\} = \text{span}\{\boldsymbol{u}_1, \ldots \boldsymbol{u}_m\}.$$

Therefore, it can also result in an orthonormal set by normalizing each vector. For example, obtaining the orthonormal matrix Q_m can be done as follows:

$$Q_m = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m] \text{ where } \boldsymbol{q}_j = \boldsymbol{u}_j / \|\boldsymbol{u}_j\| \qquad \forall j = 1, \ldots, m, \text{ and span}\{\boldsymbol{w}_1, \ldots \boldsymbol{w}_m\} = \text{span}\{\boldsymbol{q}_1, \ldots \boldsymbol{q}_m\}.$$

Various GS algorithms have been proposed and analyzed including the *classical Gram-Schmidt* algorithm (CGS) [14], the *modified Gram-Schmidt* algorithm (MGS) [14, 15, 12] and their re-orthogonalized algorithms[9, 10] (namely CGS2 and MGS2, respectively) . The latter methods improve computational orthogonality for practical implementation. Depending on the choice of Gram-Schmidt algorithms, the projection can be defined with respect to iterations by

$$\begin{aligned}
\text{CGS}: &\qquad P^{(j)} = I_n - Q_j Q_j^T, \\
\text{CGS2}: &\qquad P^{(j)} = (I_n - Q_j Q_j^T)(I_n - Q_j Q_j^T), \\
\text{MGS}: &\qquad P^{(j)} = (I_n - \boldsymbol{q}_j \boldsymbol{q}_j^T)(I_n - \boldsymbol{q}_{j-1} \boldsymbol{q}_{j-1}^T) \cdots (I_n - \boldsymbol{q}_1 \boldsymbol{q}_1^T), \\
\text{MGS2}: &\qquad P^{(j)} = (I_n - \boldsymbol{q}_j \boldsymbol{q}_j^T) \cdots (I_n - \boldsymbol{q}_1 \boldsymbol{q}_1^T)(I_n - \boldsymbol{q}_j \boldsymbol{q}_j^T) \cdots (I_n - \boldsymbol{q}_1 \boldsymbol{q}_1^T),
\end{aligned}$$

where $Q_j = \begin{bmatrix} \boldsymbol{q}_1, \ldots, \boldsymbol{q}_j \end{bmatrix}$ for each $j \in \{1, \ldots, m\}$. These projectors approximate the l_2 orthogonal projector $I_n - Q_j Q_j^\dagger$ onto the orthogonal complement of $\text{span}(Q_j)$ in the inexact arithmetic where the quality of computational orthogonality relies on $\kappa(Q_m)$.

3.1 RGS-L2 algorithms

The RGS algorithm [5] is based on random sketching. To be specific, it is motivated by approximating inner products of high dimension vectors with inner products of their low dimensional random projections, so called sketch. Thus, the approximate orthogonality depends on the random projection rather than the l_2 norm sense. For more theoretical proofs and details of RGS, we refer to [5] and the references therein. Unlike other variants of Gram-Schmidt process, e.g. CGS, MGS, CGS2 and MGS2, the randomized Gram-Schmidt process allows us to have Θ orthogonality rather than l_2 orthogonality. In other words, we will employ Θ orthogonal projector hence we define

$$\text{RGS}: \qquad P^{(j)} = I_n - Q_j (\Theta Q_j)^\dagger \Theta,$$

where $Q_j = \begin{bmatrix} \boldsymbol{q}_1, \ldots, \boldsymbol{q}_j \end{bmatrix}$ for each $j \in \{1, \ldots, m\}$. As in the CGS2 and MGS2 algorithms, the reorthogonalizing process with the l_2 orthogonal projector in the RGS algorithm leads us to define the following l_2 orthogonal projector:

$$\text{RGS-L2C}: \qquad P^{(j)} = (I_n - Q_j Q_j^T)(I_n - Q_j (\Theta Q_j)^\dagger \Theta), \tag{3.1}$$

$$\text{RGS-L2M}: \qquad P^{(j)} = (I_n - \boldsymbol{q}_j \boldsymbol{q}_j^T) \cdots (I_n - \boldsymbol{q}_1 \boldsymbol{q}_1^T)(I_n - Q_j (\Theta Q_j)^\dagger \Theta), \tag{3.2}$$

by combining the RGS with either CGS or MGS. Consequently, using (3.1) or (3.2), we can derive the full RGS-L2 algorithm where it is described in Algorithm 2.

As depicted in Algorithm 2, we introduce two types of reorthogonalization algorithm and we call the corresponding methods *RGS-L2C* and *RGS-L2M*, respectively, for convenience. The fundamental idea in the RGS-L2 algorithm is that we can reduce the computational cost by employing the sketched orthogonal projections and impose the l_2 orthogonality of Q rather than Θ orthogonality as in the RGS algorithm. Furthermore, in inexact arithmetic, the resulting matrix Q of RGS-L2 gets less loss of orthogonality than CGS, MGS and RGS. On the other hand, in exact arithmetic, we can easily show that Q is l_2 orthonormal.

Theorem 1. *Let Q and R be matrices given by performing Algorithm 2 on W where W has full rank. Suppose the sketching matrix Θ satisfying that $\Theta \boldsymbol{w}_j \neq 0$, for any $j = 1, \ldots, m$. Then it is true that $W = QR$ and Q is l_2 orthonormal in exact arithmetic.*

Algorithm 2 RGS algorithm with reorthogonalization

Input: matrix $W \in \mathbb{R}^{n\times m}$, sketching matrix $\Theta \in \mathbb{R}^{t\times n}$
Output: $Q \in \mathbb{R}^{n\times m}$, $R \in \mathbb{R}^{m\times m}$

1: **for** $j = 1 : m$ **do**
2: $\quad \boldsymbol{u}_j = \boldsymbol{w}_j$.
3: $\quad$ Sketch $\boldsymbol{w}_j$: $\boldsymbol{p}_j = \Theta \boldsymbol{w}_j$.
4: $\quad$ Solve $t \times (j-1)$ least squares problem: $R_1(1:j-1, j) = \arg\min_{\boldsymbol{y}\in\mathbb{R}_1^{j-1}} \left\| S_{j-1}\boldsymbol{y} - \boldsymbol{p}_j \right\|$
5: $\quad \boldsymbol{u}_j = \boldsymbol{w}_j - Q_{j-1}R_1(1:j-1, j)$
6: $\quad \boldsymbol{v}_j = \boldsymbol{u}_j$.

Reorthogonalizing (via CGS)	(via MGS)
	for $i = 1 : j-1$ **do**
7: $\quad R_2(1:j-1, j) = Q_{j-1}^T \boldsymbol{v}_j$	$\quad R_2(i, j) = \boldsymbol{q}_i^T \boldsymbol{v}_j$
$\quad \boldsymbol{v}_j = \boldsymbol{v}_j - Q_{j-1}R_2(1:j-1, j)$	$\quad \boldsymbol{v}_j = \boldsymbol{v}_j - \boldsymbol{q}_i R_2(i, j)$
	end for

8: $\quad R(1:j-1, j) = R_1(1:j-1, j) + R_2(1:j-1, j)$; $R(j, j) = \|\boldsymbol{v}_j\|$.
9: $\quad \boldsymbol{q}_j = \boldsymbol{v}_j / R(j, j)$.
10: $\quad$ Sketch $\boldsymbol{q}_j$: $\boldsymbol{s}_j = \Theta \boldsymbol{q}_j$.
11: $\quad S_j = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_j]$; $Q_j = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_j]$.
12: **end for**
return $Q = Q_m$ and R.

Proof. Let $j \in \{1, \ldots, m\}$. It is obvious that

$$\boldsymbol{w}_j = \boldsymbol{u}_j + Q_{j-1}R_1(1:j-1, j) \qquad \text{and} \qquad \boldsymbol{u}_j = \boldsymbol{v}_j + Q_{j-1}R_2(1:j-1, j) \tag{3.3}$$

where $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ are defined in Lines 5 and 7, respectively, for each j, in either RGS-L2C or RGS-L2M. By the definition of $\boldsymbol{q}_j$, we can write

$$\boldsymbol{w}_j = \boldsymbol{q}_j R(j, j) + Q_{j-1}(R_1(1:j-1, j) + R_2(1:j-1, j)) = Q_j R(1:j, j),$$

thus, to tidy up the result for all j, we have $W = QR$.

To prove the orthonormality of Q, we want to use mathematical induction. For $j = 1$, it is trivial that $\|\boldsymbol{q}_1\| = 1$. We suppose the induction hypothesis such that $Q_j^T Q_j = I_j$ for $1 \le j < m$. If we show $Q_{j+1}^T Q_{j+1} = I_{j+1}$, then the proof completes by induction.

Consider $Q_j^T \boldsymbol{q}_{j+1}$. By the definition of $\boldsymbol{q}_{j+1}$ and (3.3) for $j+1$, we can write it as

$$Q_j^T \boldsymbol{q}_{j+1} = \frac{Q_j^T \boldsymbol{v}_{j+1}}{R(j, j)} = \frac{Q_j^T(\boldsymbol{u}_{j+1} - Q_j R_2(1:j, j+1))}{R(j, j)} = \frac{Q_j^T \boldsymbol{u}_{j+1} - R_2(1:j, j+1)}{R(j, j)},$$

since $Q_j^T Q_j = I_j$. Corresponding to Line 7 of Algorithm 2, in the CGS variant, $R_2(1:j, j+1)$ is defined by $R_2(1:j, j+1) = Q_j^T \boldsymbol{u}_{j+1}$. On the other hand, the MGS variant leads us to rewrite $R_2(k, j+1)$ for each $k = 1, \ldots, j$ as

$$R_2(k, j+1) = \boldsymbol{q}_k^T \boldsymbol{u}_{j+1} - \sum_{i=1}^{k-1} \boldsymbol{q}_k^T \boldsymbol{q}_i R_2(i, j+1) = \boldsymbol{q}_k^T \boldsymbol{u}_{j+1},$$

by the induction assumption such as the orthogonality of Q_j. Hence, regardless of the choice of reorthogonalization, we have

$$Q_j^T \boldsymbol{q}_{j+1} = \frac{Q_j^T \boldsymbol{u}_{j+1} - R_2(1:j, j+1)}{R(j, j)} = \frac{Q_j^T \boldsymbol{u}_{j+1} - Q_j^T \boldsymbol{u}_{j+1}}{R(j, j)} = O_{j\times 1}.$$

Therefore, we can derive

$$Q_{j+1}^T Q_{j+1} = \begin{bmatrix} Q_j^T Q_j & Q_j^T \boldsymbol{q}_{j+1} \\ \boldsymbol{q}_{j+1}^T Q_j & \boldsymbol{q}_{j+1}^T \boldsymbol{q}_{j+1} \end{bmatrix} = \begin{bmatrix} I_j & O_{j\times 1} \\ O_{1\times j} & 1 \end{bmatrix} = I_{j+1},$$

so that we end the proof by induction. □

3.2 Cost analysis

Next, we analyze the computational complexity of RGS-L2 algorithm.

Theorem 2. *Supposed the random sketching in Algorithm 2 is performed by P-SRHT via FWHT. Also, we assume that one of common least squares solvers such as Givens QR solver, Householder solver, etc., is used for solving the least squares problems of size* $t \times (j-1)$*. Then the total computational complexity of the RGS-L2 algorithm is* $3nm^2$ *flops asymptotically, where* $\log(t) \ll m$*.*

Proof. The proof follows the cost analysis of the RGS algorithm in [6]. We first consider costs corresponding to lines for $j = 1, \ldots, m$.

- Random sketching: Lines 3 and 10. By introducing FWHT in P-SRHT, it costs $n\log(t)$ *flops*.
- Solving least square problems: Line 4. Using Givens QR or Householder solvers, it requires $O(t(j-1)^2)$ *flops*, e.g. $2t(j-1)^2$ *flops* or $3t(j-1)^2$ *flops*.
- Defining $\boldsymbol{u}_j$: Line 5. It consists of a matrix-vector multiplication and a vector addition. Hence, the cost is $2nj$ *flops*.
- Reorthogonalizing: Line 7. For both CGS and MGS variants, we need $4nj$ *flops*.
- Defining $R(1:j-1, j)$: Line 7. The addition of vectors of length $(j-1)$ requires $(j-1)$ *flops*.
- Normalization: Lines 8 and 9. To normalized a vector of length n, $(2n+2)$ *flops* is needed.

Tidying up all results, summing them up over $j = 1, \ldots, m$ gives

$$\begin{aligned}\text{Total complexity} &= \sum_{j=1}^{m} \big(2n\log(t) + O(t(j-1)^2) + 6nj + j - 1 + 2n + 2\big) \\ &\approx \sum_{j=1}^{m} 6nj \approx 3nm^2,\end{aligned}$$

since other computational costs are dominated by nj when j is large enough such as $\log(t) \ll j$. □

Although the complexity of RGS-L2 is larger than CGS and MGS about 50%, it is 25% cheaper than CGS2 and MGS2. Overall, the complexity of GS process follows

$$\text{RGS} : nm^2 \textit{ flops} \quad < \quad \text{CGS/MGS} : 2nm^2 \textit{ flops} \quad < \quad \text{RGS-L2} : 3nm^2 \textit{ flops} \quad < \quad \text{CGS2/MGS2} : 4nm^2 \textit{ flops},$$

respectively.

Remark (Other randomization methods for orthogonal process)**.** *While we employ the reorthogonalization process in* l_2 *norm, De Damas and Grigori[16] propose another reorthogonalization algorithm for sketched orthonormal matrix. Instead of combining* l_2 *orthogonal projector and* θ *orthogonal projector like ours, their randomized method performs the sketched projection twice. Hence, it produces the sketched orthonormal basis vectors with less computational cost rather than* l_2 *orthonormal.*

Although the Gram-Schmidt algorithm is the most common method for orthonormalization, Cholesky decomposition is also widely used for QR decomposition. In a randomized manner, Balabanov[17] develops several variants of randomized Cholesky QR factorization. The proposed methods generate a set of either sketched or l_2 *orthonormal vectors depending on the number of performing Cholesky decomposition. However, in order to ensure the stability, it requires positive definiteness.*

4 Loss of orthogonality

In our rounding error analysis, we follow the standard rounding model. Using the most common notation, we have

$$fl(x \text{ op } y) = (x \text{ op } y)(1+\delta), \qquad \text{op } \in \{+, -, *, /\}, \tag{4.1}$$

where $|\delta| \leq \bar{u}$, x and y are floating point numbers and $fl(x)$ denotes the nearest floating-point number to x. In the standard rounding model, the rounding error for $\boldsymbol{z} = Y\boldsymbol{x}$ is given as

$$\bar{\boldsymbol{z}} = Y\boldsymbol{x} + \delta\boldsymbol{z}, \qquad \text{with } |\delta\boldsymbol{z}| \leq \xi(n,m)\bar{u}, \tag{4.2}$$

where $\bar{\boldsymbol{z}}$ denotes the computed floating point number of $\boldsymbol{z}$, Y is a matrix of m by n, x is a vector of n length and $\xi(n,m) = O(nm)$. We refer to the book of Higham[15] for more details of the standard rounding model. On the other

hand, the probabilistic rounding model is widely used in the randomized linear algebra. In this model, δ is bounded by random variables and $\delta \boldsymbol{z}$ has zero means and its entries are linearly independent. For the probabilistic and stochastic models, we refer to [18, 19].

Let us consider $\boldsymbol{s} = \Theta \boldsymbol{x}$ in exact arithmetic and denote $\bar{\boldsymbol{s}}$ for finite precision arithmetic. We recall the rounding error analysis of random sketching from [5] such that

$$\Theta \boldsymbol{x} = \bar{\boldsymbol{s}} + \delta \boldsymbol{s}, \qquad \|\delta \boldsymbol{s}\| \leq \xi_s \|\boldsymbol{x}\| \bar{u}, \tag{4.3}$$

where ξ_s depends on the oblivious l_2 subspace embedding property. In our rounding error analysis, we assume that (4.3) holds with $\xi_s = O(1)$. Please see Corollary 2.6 in [5] for the condition of the oblivious l_2 subspace embedding to obtain (4.3) with $\xi_s = O(1)$. Therefore random sketching enables us to reduce the dimension of the problem without large loss of numerical precision. We refer to [5] and its supplementary material for the detail of the rounding error analysis of the RGS algorithm. However, for simplicity, we consider only fixed precision, not mixed precision. Furthermore, we follow the rounding error analysis of an elementary orthogonalization step [20, 14, 11] and we suppose that $nm\bar{u} \ll 1$.

4.1 Analysis of RGS-L2C

In our analysis, we consider the standard rounding model, while (4.3) is derived by the probabilistic rounding model. Indeed, the probabilistic rounding model is only required for random sketching. Hence, by assuming (4.3) for simplicity, we show the rounding error analysis with the standard rounding model.

Theorem 3. *Suppose $\bar{Q}$ and $\bar{R}$ are computed by Algorithm 2 in the CGS variant. Then it satisfies that*

$$W + \delta W = \bar{Q}\bar{R}, \qquad \|\delta W\| \leq \xi_C \|W\| \bar{u}, \tag{4.4}$$

where ξ_C is a low degree polynomial of n and m and is independent of $\bar{u}$, W, Q and R. The loss of orthogonality is bounded by

$$\left\| I_m - \bar{Q}^T \bar{Q} \right\| \leq \tilde{\xi}_C \bar{u}, \tag{4.5}$$

where $\tilde{\xi}_C = O(nm^{3/2})$, if $\tilde{\xi}_r \kappa(W) \bar{u} < 1$ holds for some $\tilde{\xi}_r = O(n^2 m^3)$.

Proof. Recall Algorithm 2 and use the standard rounding error results in the elementary orthogonalizing process. The vector $\bar{\boldsymbol{u}}_j$ computed in Line 5 satisfies

$$\bar{\boldsymbol{u}}_j = \boldsymbol{w}_j - \sum_{i=1}^{j-1} \bar{\boldsymbol{q}}_i \bar{r}_{i,j}^{(1)} + \delta \boldsymbol{u}_j \qquad \|\delta \boldsymbol{u}_j\| \leq \xi_2 \|\boldsymbol{w}_j\| \bar{u}, \tag{4.6}$$

where $\xi_2 = O(nm)$ and we can also obtain

$$\bar{\boldsymbol{v}}_j = \bar{\boldsymbol{u}}_j - \sum_{i=1}^{j-1} \bar{\boldsymbol{q}}_i \bar{r}_{i,j}^{(2)} + \delta \boldsymbol{v}_j \qquad \|\delta \boldsymbol{v}_j\| \leq \xi_2 \|\bar{\boldsymbol{u}}_j\| \bar{u}, \tag{4.7}$$

in Line 7. By introducing any least square solver satisfying the backward-stability property such that

$$\bar{R}_1(1:j-1,j) = \arg\min_y \left\| (\bar{S}_{j-1} + \delta S_{j-1}) \boldsymbol{y} - (\bar{\boldsymbol{p}}_j + \delta \boldsymbol{p}_j) \right\|,$$

we can derive

$$\left\| \bar{R}_1(1:j-1,j) \right\| \leq \xi_1 \|\boldsymbol{w}_j\|, \tag{4.8}$$

where $\xi_1 = O(1)$ by (4.3). For example, in [5] and its Supplementary material, it is given by $\xi_1 = 1.4$. Let α_j be the error of the least square problem such that

$$\alpha_j = \left\| (\bar{S}_{j-1} + \delta S_{j-1}) \bar{R}_1(1:j-1,j) - (\bar{\boldsymbol{p}}_j + \delta \boldsymbol{p}_j) \right\|.$$

This is equivalent to

$$\alpha_j = \left\| \Theta \bar{Q}_{j-1} \bar{R}_1(1:j-1,j) - \Theta \boldsymbol{w}_j \right\|.$$

By the subspace embedding property, we have

$$\alpha_j \geq (1-\epsilon) \left\| \bar{Q}_{j-1} \bar{R}_1(1:j-1,j) - \bar{\boldsymbol{w}}_j \right\|,$$

and so the backward error estimates of the least square solvers (e.g. see the work of Higham[15]) imply

$$\left\| \bar{Q}_{j-1} \bar{R}_1(1:j-1,j) - \bar{\boldsymbol{w}}_j \right\| \leq \alpha_j \,/(1-\epsilon) \leq \xi_\epsilon \bar{u} \|\boldsymbol{w}_j\|, \tag{4.9}$$

where $\xi_\epsilon = O(nm)$. We refer to the work of Balabanov and Grigori[5] for more details of the bounds (4.8) and (4.9).

On the other hand, the other orthogonalization coefficients $\bar{r}_{i,j}^{(2)}$ and the diagonal elements $\bar{r}_{j,j}$ fulfill

$$\bar{r}_{i,j}^{(2)} = \bar{\boldsymbol{q}}_i^T \bar{\boldsymbol{u}}_j + \delta r_{i,j}^{(2)}, \quad |\delta r_{i,j}^{(2)}| \leq n\bar{u} \left\|\bar{\boldsymbol{q}}_i\right\| \left\|\bar{\boldsymbol{u}}_j\right\|, \tag{4.10}$$

$$\bar{r}_{j,j} = \left\|\bar{\boldsymbol{v}}_j\right\| + \delta r_{j,j}, \qquad |\delta r_{j,j}| \leq n\bar{u} \left\|\bar{\boldsymbol{v}}_i\right\|, \tag{4.11}$$

respectively. By normalization, we have the computed $\bar{\boldsymbol{q}}_j$ such that

$$\bar{\boldsymbol{q}}_j = \bar{\boldsymbol{v}}_j / \left\|\bar{\boldsymbol{v}}_j\right\| + \delta \boldsymbol{q}_j, \qquad \left\|\delta \boldsymbol{q}_j\right\| \leq (n+4)\bar{u}, \qquad \left\|\bar{\boldsymbol{q}}_j\right\|^2 \leq 1 + (n+4)\bar{u}. \tag{4.12}$$

Combining (4.6) with (4.7), we can obtain

$$\boldsymbol{w}_j + \delta \boldsymbol{u}_j + \delta \boldsymbol{v}_j = \sum_{i=1}^{j-1} (\bar{r}_{i,j}^{(1)} + \bar{r}_{i,j}^{(2)}) \bar{\boldsymbol{q}}_i + \bar{\boldsymbol{v}}_j, \tag{4.13}$$

then collecting (4.13) for all $j = 1, \ldots, m$ gives

$$W + \delta W = \bar{Q}\bar{R}, \tag{4.14}$$

where $\delta W = \delta U + \delta V = [\delta \boldsymbol{u}_1, \ldots, \delta \boldsymbol{u}_m] + [\delta \boldsymbol{v}_1, \ldots, \delta \boldsymbol{v}_m]$. Then we shall show a bound for δW. We first note that, from (4.6),

$$\begin{aligned} \left\|\bar{\boldsymbol{u}}_j\right\| \leq & \left\|\boldsymbol{w}_j\right\| + \sum_{i=1}^{j-1} \left\|\bar{\boldsymbol{q}}_i\right\| |\bar{r}_{i,j}^{(1)}| + \left\|\delta \boldsymbol{u}_j\right\| \\ \leq & \left\|\boldsymbol{w}_j\right\| \left(1 + 1.4\sqrt{(j-1)}\sqrt{1 + (n+4)\bar{u}} + \xi_2 \bar{u}\right) \\ \leq & \xi_3 \left\|\boldsymbol{w}_j\right\|, \end{aligned} \tag{4.15}$$

by (4.8) and (4.12), where $\xi_3 = O(m^{1/2})$. With (4.15), (4.7) implies that

$$\left\|\delta \boldsymbol{v}_j\right\| \leq \xi_4 \left\|\boldsymbol{w}_j\right\| \bar{u}, \tag{4.16}$$

where $\xi_4 \leq \xi_2 \xi_3 = O(nm^{3/2})$. Therefore, taking into account (4.7) and (4.16), the perturbation matrix δW is bounded by

$$\left\|\delta W\right\| \leq \xi_C \left\|W\right\| \bar{u}, \tag{4.17}$$

where $\xi_C \leq \xi_2 + \xi_4$.

Next, we want to show the bound of the loss of orthogonality by induction such that

$$\left\|I_i - \bar{Q}_i^T \bar{Q}_i\right\| \leq \tilde{\xi}_i \bar{u}, \tag{4.18}$$

for some a low degree polynomial $\tilde{\xi}_i$ of n and m, for any $i = 1, \ldots, m$. For $i = 1$, it trivially holds (4.18). We suppose that (4.18) is true for $i = j - 1$ with $1 < j \leq m$. To prove the statement for j, we follow the argument described in [21, 11] such that

$$\text{if } \left\|\bar{Q}_{i-1}^T \bar{\boldsymbol{q}}_i\right\| \leq \xi_5 \bar{u} \qquad \text{for } i = 1, \ldots, j, \tag{4.19}$$

$$\text{then } \left\|I_j - \bar{Q}_j^T \bar{Q}_j\right\| \leq \max_{i=1,\ldots,j} \left\{\left\|\bar{\boldsymbol{q}}_i\right\| - 1 + \left\|\bar{Q}_{i-1}^T \bar{\boldsymbol{q}}_i\right\| \sqrt{2j}\right\} \leq \xi_6 \bar{u} \tag{4.20}$$

where $\xi_5 = O(nm)$ and $\xi_6 = O(nm^{3/2})$. Note that if (4.19) holds for $j - 1$, (4.20) implies that

$$\left\|\bar{Q}_{j-1}\right\| \leq \left(1 + \xi_6 \bar{u}\right)^{1/2}. \tag{4.21}$$

Clearly, (4.19) is true for $i = 1$. Hence, with the assumption of induction for $j - 1$, it is enough to show (4.19) for j to complete the proof.

Consider the bound of $\left\|\bar{Q}_{j-1}^T \bar{\boldsymbol{u}}_j\right\|$. By (4.6), (4.9) and (4.21), we have

$$\begin{aligned} \left\|\bar{Q}_{j-1}^T \bar{\boldsymbol{u}}_j\right\| \leq & \left\|\bar{Q}_{j-1}\right\| \left\|\boldsymbol{w}_j - \bar{Q}_{j-1} \bar{R}_1(1 : j-1, j)\right\| + \left\|\bar{Q}_{j-1}\right\| \left\|\delta \boldsymbol{u}_j\right\| \\ \leq & \xi_7 \left\|\boldsymbol{w}_j\right\| \bar{u}, \end{aligned} \tag{4.22}$$

where $\xi_7 = O(nm)$. After noting that

$$\bar{Q}_{j-1}^T \bar{\boldsymbol{v}}_j = \left(I_{j-1} - \bar{Q}_{j-1}^T \bar{Q}_{j-1}\right) \bar{Q}_{j-1}^T \bar{\boldsymbol{u}}_j + \bar{Q}_{j-1}^T \left(-\sum_{i=1}^{j-1} \bar{\boldsymbol{q}}_i \delta r_{i,j}^{(2)} + \delta \boldsymbol{v}_j\right),$$

the assumption of induction (4.18) with (4.10), (4.12), (4.20), and (4.22) yields that

$$\begin{aligned}\frac{\left\|\bar{Q}_{j-1}^T\bar{\boldsymbol{v}}_j\right\|}{\|\bar{\boldsymbol{v}}_j\|} &\leq \left\|I_{j-1}-\bar{Q}_{j-1}^T\bar{Q}_{j-1}\right\|\frac{\left\|\bar{Q}_{j-1}^T\bar{\boldsymbol{u}}_j\right\|}{\|\bar{\boldsymbol{v}}_j\|}+\left\|\bar{Q}_{j-1}\right\|\left(\sum_{i=1}^{j-1}\|\bar{\boldsymbol{q}}_i\|\,|\delta\bar{r}_{i,j}^{(2)}|+\|\delta\boldsymbol{v}_j\|\right)/\|\bar{\boldsymbol{v}}_j\| \\ &\leq \left(\xi_6\xi_7\bar{u}\frac{\|\boldsymbol{w}_j\|}{\|\bar{\boldsymbol{v}}_j\|}+\xi_8\frac{\|\bar{\boldsymbol{u}}_j\|}{\|\bar{\boldsymbol{v}}_j\|}\right)\bar{u}, \end{aligned} \tag{4.23}$$

where $\xi_8 = O(nm)$. Also, by (4.12), (4.21) and (4.23), we can write

$$\begin{aligned}\left\|\bar{Q}_{j-1}^T\bar{\boldsymbol{q}}_j\right\| \leq& \frac{\left\|\bar{Q}_{j-1}^T\bar{\boldsymbol{v}}_j\right\|}{\|\bar{\boldsymbol{v}}_j\|}+\left\|\bar{Q}_{j-1}^T\delta\boldsymbol{q}_j\right\| \\ &\leq \left(\xi_6\xi_7\bar{u}\frac{\|\boldsymbol{w}_j\|}{\|\bar{\boldsymbol{u}}_j\|}\frac{\|\bar{\boldsymbol{u}}_j\|}{\|\bar{\boldsymbol{v}}_j\|}+\xi_8\frac{\|\bar{\boldsymbol{u}}_j\|}{\|\bar{\boldsymbol{v}}_j\|}+\left(1+\xi_6\bar{u}\right)^{1/2}(n+4)\right)\bar{u}. \end{aligned} \tag{4.24}$$

Finally, to complete the proof, we consider the upper bounds of $\|\boldsymbol{w}_j\| / \|\bar{\boldsymbol{u}}_j\|$ and $\|\bar{\boldsymbol{u}}_j\| / \|\bar{\boldsymbol{v}}_j\|$.

- $\|\boldsymbol{w}_j\| / \|\bar{\boldsymbol{u}}_j\|$
  By recalling (4.13), we can write

$$W_j+\Delta_j = \bar{Q}_{j-1}\left[\bar{R}_{j-1},\bar{\boldsymbol{r}}_j\right], \tag{4.25}$$

  where $\Delta_j = [\delta U_{j-1}+\delta V_{j-1}, \delta\boldsymbol{u}_j-\bar{\boldsymbol{u}}_j]$, $\bar{R}_{j-1} = \bar{R}(1:j-1,1:j-1)$ and $\bar{\boldsymbol{r}}_j = \bar{R}_1(1:j-1,j)$. We introduce and follow the arguments in [13, 11] and the references therein to describe the distance to singularity. More precisely, while W_j is nonsingular, the right hand side of (4.25) is of rank $j-1$, thus the perturbation matrix Δ_j satisfies that

$$\sigma_{\min}(W) \leq \sigma_{\min}(W_j) \leq \|\Delta_j\|. \tag{4.26}$$

  Then dividing (4.26) by $\|W_j\|$ yields

$$\frac{1}{\kappa(W_j)} \leq \frac{\|\Delta_j\|}{\|W_j\|} \leq \frac{\|\Delta_j\|_F}{\|W_j\|}. \tag{4.27}$$

  Since (4.6), (4.16) and the definition of Δ_j give

$$\|\Delta_j\|_F \leq (j\xi_2+(j-1)\xi_4)\,\bar{u}\|W_j\|_F+\|\bar{\boldsymbol{u}}_j\|, \tag{4.28}$$

  we can derive

$$\kappa(W) \geq \kappa(W_j) \geq \frac{\|W_j\|}{\|\Delta_j\|_F}. \tag{4.29}$$

  Using the fact that $\|W_j\|_F \leq \sqrt{j}\,\|W_j\|$ and $\|\boldsymbol{w}_j\| \leq \|W_j\|$, we deduce (4.29) to

$$\kappa(W) \geq \frac{1}{(j\xi_2+(j-1)\xi_4)\,\bar{u}\frac{\|W\|_F}{\|W\|}+\frac{\|\bar{\boldsymbol{u}}_j\|}{\|W\|}} \geq \frac{1}{\sqrt{j}\,(j\xi_2+(j-1)\xi_4)\,\bar{u}+\frac{\|\bar{\boldsymbol{u}}_j\|}{\|\boldsymbol{w}_j\|}}, \tag{4.30}$$

  and we can rewrite it as

$$\frac{\|\boldsymbol{w}_j\|}{\|\bar{\boldsymbol{u}}_j\|} \leq \frac{\kappa(W)}{1-\sqrt{m}\,(m\xi_2+(m-1)\xi_4)\,\bar{u}\kappa(W)}. \tag{4.31}$$

  We assume that $\sqrt{m}\,(m\xi_2+(m-1)\xi_4)\,\bar{u}\kappa(W) \ll 1$ and we suppose there exists $\eta_0 = O(1)$ such that

$$\frac{\|\boldsymbol{w}_j\|}{\|\bar{\boldsymbol{u}}_j\|} \leq \eta_0\kappa(W). \tag{4.32}$$

- $\|\bar{\boldsymbol{u}}_j\| / \|\bar{\boldsymbol{v}}_j\|$
  To derive the upper bound of $\|\bar{\boldsymbol{u}}_j\| / \|\bar{\boldsymbol{v}}_j\|$, we consider the lower bound of its reciprocal. From the definition of $\bar{\boldsymbol{v}}_j$ and $\bar{r}_{i,j}^{(2)}$, taking l_2 norm and dividing it by $\|\bar{\boldsymbol{u}}_j\|$ gives

$$\frac{\|\bar{\boldsymbol{v}}_j\|}{\|\bar{\boldsymbol{u}}_j\|} \geq 1-\left\|\bar{Q}_{j-1}\right\|\frac{\left\|\bar{Q}_{j-1}^T\bar{\boldsymbol{u}}_j\right\|}{\|\bar{\boldsymbol{u}}_j\|}-\frac{\sum_{i=1}^{j-1}\left\|\bar{\boldsymbol{q}}_i\delta r_{i,j}^{(2)}\right\|}{\|\bar{\boldsymbol{u}}_j\|}-\frac{\|\delta\boldsymbol{v}_j\|}{\|\bar{\boldsymbol{u}}_j\|}. \tag{4.33}$$

Hence, employing (4.7), (4.10), (4.12), (4.15), (4.21) and (4.22) implies that

$$\frac{\|\bar{\boldsymbol{v}}_j\|}{\|\bar{\boldsymbol{u}}_j\|} \geq 1 - \left((1+\xi_6\bar{u})^{1/2}\,\xi_7\bar{u} + m(1+(n+4)\bar{u})n\xi_3\bar{u} + \xi_2\xi_3\bar{u} \right) \frac{\|\boldsymbol{w}_j\|}{\|\bar{\boldsymbol{u}}_j\|} = 1 - \xi_9\bar{u}\frac{\|\boldsymbol{w}_j\|}{\|\bar{\boldsymbol{u}}_j\|}, \tag{4.34}$$

for some $\xi_9 = O(nm^{3/2})$. Here, we assume that $\xi_9\eta_0\kappa(W)\bar{u} < 1$ then we have

$$\frac{\|\bar{\boldsymbol{u}}_j\|}{\|\bar{\boldsymbol{v}}_j\|} \leq \frac{1}{1-\xi_9\eta_0\kappa(W)\bar{u}}, \tag{4.35}$$

by (4.32).

Tidying up all results above, we can obtain

$$\left\|\bar{Q}_{j-1}^T\bar{\boldsymbol{q}}_j\right\| \leq \left(\xi_6\xi_7\bar{u}\frac{\eta_0\kappa(W)}{1-\xi_9\eta_0\kappa(W)\bar{u}} + \frac{\xi_8}{1-\xi_9\eta_0\kappa(W)\bar{u}} + (1+\xi_6\bar{u})^{1/2}\,(n+4) \right)\bar{u}. \tag{4.36}$$

To end the proof, we suppose that $\xi_6\xi_7\eta_0\kappa(W)\bar{u} \leq 1$ and $(1-\xi_9\eta_0\kappa(W)\bar{u})^{-1} = O(1)$. To sum up all assumptions we introduced, we need

$$\tilde{\xi}_r\bar{u}\kappa(W) < 1, \tag{4.37}$$

where $\tilde{\xi}_r = O(n^2m^3)$. As a result, it satisfies that

$$\left\|\bar{Q}_{j-1}^T\bar{\boldsymbol{q}}_j\right\| \leq \tilde{\xi}_5\bar{u}, \tag{4.38}$$

where $\tilde{\xi}_5 = O(nm)$ so that it completes the proof by induction. □

In the proof, the condition (4.37) is interpreted as the numerical nonsingularity of the matrix W.

4.2 Analysis of RGS-L2m

In a similar way with the proof of Theorem 3, we can show the rounding error analysis of the RGS-L2 with MGS variant.

Theorem 4. *Suppose $\bar{Q}$ and $\bar{R}$ are computed by Algorithm 2 in the MGS variant. Then it satisfies that*

$$W + \delta W = \bar{Q}\bar{R}, \qquad \|\delta W\| \leq \xi_M \|W\|\,\bar{u}, \tag{4.39}$$

where ξ_M is a low degree polynomial of n and m and is independent of $\bar{u}$, W, Q and R and under the assumption of numerical nonsingularity, the loss of orthogonality is bounded by

$$\left\|I_m - \bar{Q}^T\bar{Q}\right\| \leq \tilde{\xi}_M\bar{u}, \tag{4.40}$$

where $\tilde{\xi}_M = O(nm^{3/2})$.

Proof. We use similar arguments in the proof of Theorem 3 but we further introduce more sequences to define $\bar{\boldsymbol{v}}_j$ such that

$$\bar{\boldsymbol{v}}_j^{(i+1)} = \bar{\boldsymbol{v}}_j^{(i)} - \bar{\boldsymbol{q}}_i\bar{r}_{i,j}^{(2)} + \delta\boldsymbol{v}^{(i)}, \qquad \left\|\delta\boldsymbol{v}^{(i)}\right\| \leq \zeta_1\bar{u}\left\|\bar{\boldsymbol{v}}_j^{(i)}\right\| \qquad \text{for } i = 1,\ldots,j-1, \tag{4.41}$$

$$\bar{\boldsymbol{v}}_j^{(i+1)} = (I_n - \bar{\boldsymbol{q}}_i\bar{\boldsymbol{q}}_i^T)\bar{\boldsymbol{v}}_j^{(i)} + \delta\boldsymbol{\varphi}_j^{(i)}, \qquad \left\|\delta\boldsymbol{\varphi}_j^{(i)}\right\| \leq \zeta_2\bar{u}\left\|\bar{\boldsymbol{v}}_j^{(i)}\right\| \qquad \text{for } i = 1,\ldots,j-1, \tag{4.42}$$

with $\bar{\boldsymbol{v}}_j^{(1)} = \bar{\boldsymbol{u}}_j$ and $\bar{\boldsymbol{v}}_j = \bar{\boldsymbol{v}}_j^{(j)}$, where $\zeta_1 = O(1)$, $\zeta_2 = O(n)$ and the orthogonalization coefficients is written as

$$\bar{r}_{i,j}^{(2)} = \bar{\boldsymbol{q}}_i^T\bar{\boldsymbol{v}}_j^{(i)} + \delta r_{i,j}^{(2)}, \qquad |\delta r_{i,j}^{(2)}| \leq \zeta_2\bar{u}\left\|\bar{\boldsymbol{v}}_j^{(i)}\right\|. \tag{4.43}$$

Using the above arguments describing errors in elementary projections proved by Björck [20] and the extended results shown by Giraud and Langou [13], we have

$$\left\|\bar{\boldsymbol{v}}_j^{(i)}\right\| \leq \zeta_1\left\|\bar{\boldsymbol{u}}_j\right\| \qquad \text{for } i = 1,\ldots,j-1. \tag{4.44}$$

For instance, if $3.12m(n+2)\bar{u} < 0.01$, ζ_1 and ζ_2 can be defined by 1.45 and $2n+3$, respectively [20, 13]. Therefore, we can express $\bar{\boldsymbol{v}}_j$ as

$$\bar{\boldsymbol{v}}_j = \bar{\boldsymbol{u}}_j - \sum_{i=1}^{j-1}\bar{\boldsymbol{q}}_i\bar{r}_{i,j}^{(2)} + \delta\boldsymbol{v}_j \qquad \|\delta\boldsymbol{v}_j\| \leq \zeta_3\left\|\bar{\boldsymbol{u}}_j\right\|\bar{u}, \tag{4.45}$$

where $\delta \boldsymbol{v}_j = \sum_{i=1}^{j-1} \delta \boldsymbol{v}^{(i)}$ and $\zeta_3 = O(m)$.

In the same way in the CGS variant (4.13), we can derive

$$W + \delta W = \bar{Q}\bar{R},$$

where δW satisfies that

$$\|\delta W\| \leq \xi_M \|W\| \bar{u} \qquad \text{for some } \xi_M = O(nm^{3/2}).$$

Next, we shall show the bound of loss of orthogonality by mathematical induction. Recalling the induction assumptions for $i = j-1$ with an arbitrary $1 < j \leq m$ and the arguments such as (4.18)-(4.22), it is enough to show (4.19) holds for $i = j$.

The proof follows the same way in Theorem 3 except the bounds with respect to $\bar{\boldsymbol{v}}_j$. Before showing the loss of orthogonality, we first introduce the following argument described in [13]:

$$\|\bar{\boldsymbol{v}}_j\|^2 + \sum_{i=1}^{j-1} |\bar{r}_{i,j}^{(2)}|^2 \leq \zeta_4 \|\bar{\boldsymbol{u}}_j\|^2, \tag{4.46}$$

for some $\zeta_4 = O(1)$. This inequality can be derived by Pythagorean theorem and we can define $\zeta_4 = 1.01$ with the assumption $1.06(2.04 + 4.43)(j-1)\bar{u}$, e.g. see Giraud et al.[13] for more details.

Next, we consider orthogonality between $\bar{\boldsymbol{v}}_j$ and $\bar{\boldsymbol{q}}_k$ for any $k = 1, \ldots, j-1$. By summing (4.41) from $i = k+1$ to $i = j-1$, we get

$$\bar{\boldsymbol{v}}_j = \bar{\boldsymbol{v}}_j^{(k+1)} - \sum_{i=k+1}^{j-1} \bar{\boldsymbol{q}}_i \bar{r}_{i,j}^{(2)} + \sum_{i=k+1}^{j-1} \delta \boldsymbol{v}^{(i)},$$

multiplying this by $\bar{\boldsymbol{q}}_k^T$ and substituting (4.42) give

$$\bar{\boldsymbol{q}}_k^T \bar{\boldsymbol{v}}_j = - \sum_{i=k+1}^{j-1} \bar{\boldsymbol{q}}_k^T \bar{\boldsymbol{q}}_i \bar{r}_{i,j}^{(2)} + \bar{\boldsymbol{q}}_k^T \delta \boldsymbol{\varphi}_j^{(k)} + \sum_{i=k+1}^{j-1} \bar{\boldsymbol{q}}_k^T \delta \boldsymbol{v}^{(i)}.$$

Clearly, using (4.12), (4.41), (4.42), (4.44), (4.46) and the induction assumption (4.19) leads us to obtain

$$\begin{aligned} \left|\bar{\boldsymbol{q}}_k^T \bar{\boldsymbol{v}}_j\right| \leq & \left(\sum_{i=k+1}^{j-1} (\bar{\boldsymbol{q}}_k^T \bar{\boldsymbol{q}}_i)^2 \right)^{1/2} \left(\sum_{i=k+1}^{j-1} |\bar{r}_{i,j}^{(2)}|^2 \right)^{1/2} + \|\bar{\boldsymbol{q}}_k\| \left\| \delta \boldsymbol{\varphi}_j^{(k)} \right\| + \|\bar{\boldsymbol{q}}_k\| \sum_{i=k+1}^{j-1} \left\| \delta \boldsymbol{v}^{(i)} \right\| \\ \leq & \zeta_5 \bar{u} \|\bar{\boldsymbol{u}}_j\|, \end{aligned} \tag{4.47}$$

where $\zeta_5 = \zeta_4 \xi_5 + \zeta_1(\zeta_2 + \zeta_1(j-k-1))(1 + (n+4)\bar{u}) \leq O(nm)$. Since k is arbitrary and

$$\left\|\bar{Q}_{j-1}^T \bar{\boldsymbol{v}}_j\right\|^2 = \sum_{i=1}^{j-1} \left|\bar{\boldsymbol{q}}_i^T \bar{\boldsymbol{v}}_j\right|^2,$$

there exists $\zeta_6 = O(nm^{3/2})$ such that

$$\left\|\bar{Q}_{j-1}^T \bar{\boldsymbol{v}}_j\right\| \leq \zeta_6 \bar{u} \|\bar{\boldsymbol{u}}_j\|. \tag{4.48}$$

Hence, we have

$$\left\|\bar{Q}_{j-1}^T \bar{\boldsymbol{q}}_j\right\| \leq \frac{\left\|\bar{Q}_{j-1}^T \bar{\boldsymbol{v}}_j\right\|}{\|\bar{\boldsymbol{v}}_j\|} + \left\|\bar{Q}_{j-1}^T \delta \boldsymbol{q}_j\right\| \leq \zeta_6 \bar{u} \frac{\|\bar{\boldsymbol{u}}_j\|}{\|\bar{\boldsymbol{v}}_j\|} + (1 + \xi_6 \bar{u})^{1/2} (n+4)\bar{u}, \tag{4.49}$$

by (4.12), (4.21) and (4.48). As shown in Theorem 3, when (4.35) holds, it can complete the proof. Therefore, under the condition of numerical nonsingularity of W such as (4.37), combining (4.49) and (4.35) shows the induction step and completes the proof. □

In the rounding error analysis, our RGS-L2C and RGS-L2M exhibit similar stability results. As a result, the way of re-orthogonalization in the RGS-L2 algorithm is a matter of choice. However, in practice, there would be a difference in parallel efficiency. More precisely, since CGS is more suited for parallel computing, RGS-L2C is more beneficial in the implementation within HPC setting. Nevertheless, in this paper, we focus solely on numerical results regarding stability without concerning parallel computing efficiency.

4.3 Numerical example

Next, to validate our stability analysis, we compare the following numerical results obtained from various GS algorithms.

Let us consider a matrix W formed by

$$W_{ij} = f_\mu(x_i, y_j) = \frac{\sin(10(x_i + y_j))}{\cos(100(y_j - x_i)) + 1.1} \qquad \text{for } x_i = i\delta_x \text{ and } y_j = j\delta_y,$$

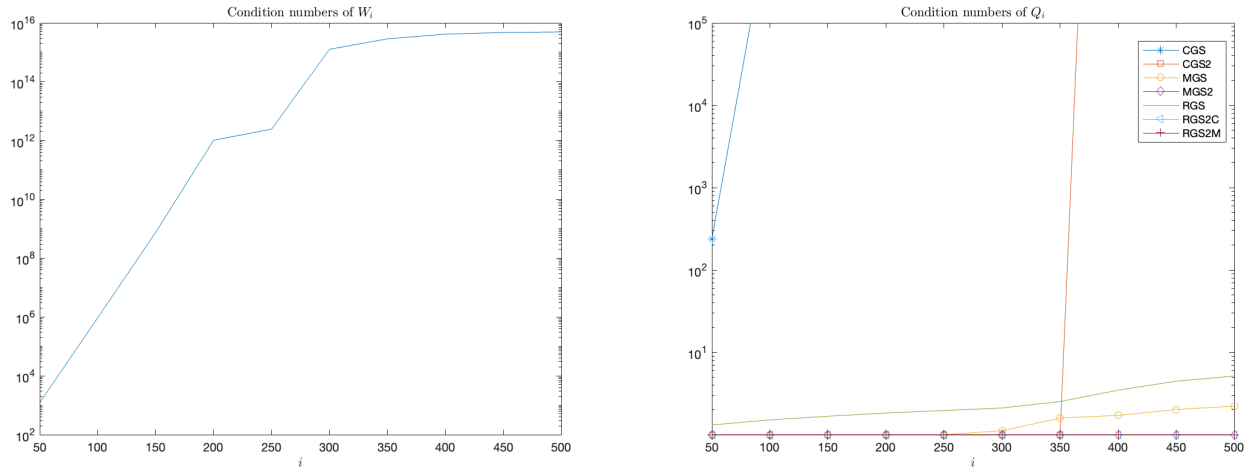
where $\delta_x = 1/n$ and $\delta_y = 1/m$ with $n = 10^6$ and $m = 500$. Then W is an ill-conditioned matrix of n by m such that $\kappa(W) = O(10^{15})$.

We decompose the submatrices of W by QR factorization using various the GS processes, such as CGS, MGS and RGS algorithms. To be specific, we perform QR factorization of W_i which is an $n \times i$ matrix extracted from the first i columns of W, resulting in $W_i = W(:, 1:i) = Q_i R_i$, with respect to the GS process. We then compute condition numbers of Q_i for $i = 1, \ldots, 500$ and approximation errors of the QR factorization with the quantity $\|W_i - Q_i R_i\| / \|W_i\|$. Note that the condition numbers close to 1 imply that the algorithms are numerically stable. Additionally, we evaluate the loss of orthogonality in l_2 norm with respect to the GS algorithms. In the RGS algorithms, the sketch size t is set to 2224 which satisfies the subspace embedding properties.

Table 2: Computing runtime in GS process.

GS type	Reorthogonalization	Time(s)
CGS	No	92.98
	Yes	186.54
MGS	No	109.39
	Yes	219.27
RGS	No	55.18
	Yes (CGS based)	149.71
	Yes (MGS based)	164.06

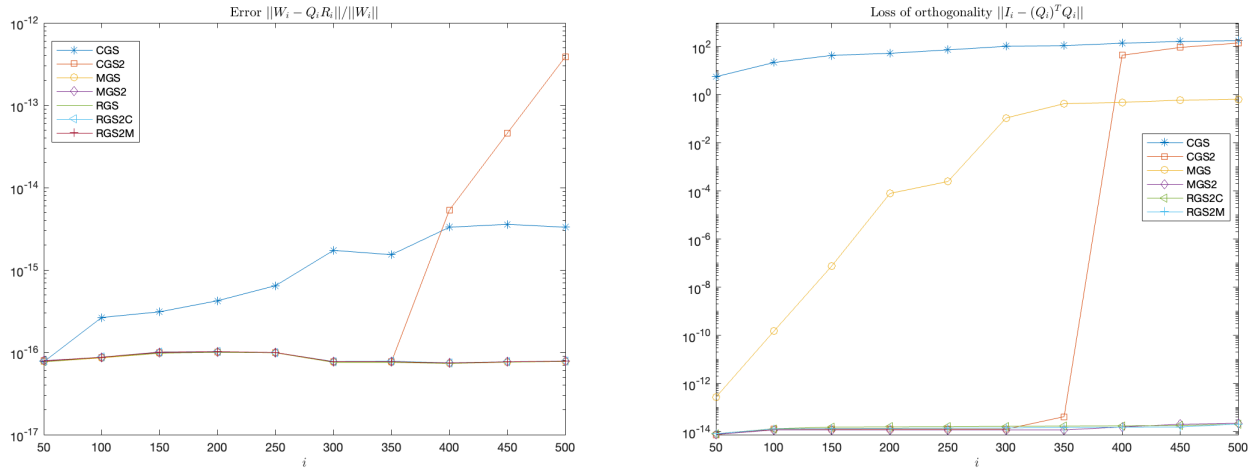
As provided in the previous section, we can observe the runtime reduction by random sketching in Table 2. More precisely, with/without reorthogonalization, the randomization leads to a 25% and 50% reduction in elapsed time, respectively, compared to CGS/MGS.



(a) $\kappa(W_i)$ (b) $\kappa(Q_i)$

Figure 1: Condition numbers of W_i and Q_i with respect to GS process.

Figure 1a illustrates that the condition number of W_i increases with i up to 5.0×10^{15}. With the given ill-conditioned W, the CGS algorithm causes the instability of Q_i as seen in Figure 1b. Initially, the reorthogonal process in the CGS variant maintains the stability for $i \leq 350$ but dramatic instability suddenly appears at $i > 350$. As we can see the numerical



(a) $\|W_i - Q_i R_i\| \,/\, \|W_i\|$ (b) $\left\|I_i - Q_i^T Q_i\right\|$

Figure 2: Approximation errors and loss of orthogonality with respect to GS process.

singularity of W_i in Figure 1a for $i > 250$, since the rounding error analysis of CGS2 relies on numerical singularity, CGS2 fails to orthogonalize vectors. Consequently, in Figure 2, there are large approximation errors and loss of orthogonality for CGS and CGS2, respectively. On the other hands, other GS algorithms lead us to obtain stable Q_i where $\kappa(Q_i)$ is close to 1. Despite increasing loss of orthogonality in MGS, the reorthogonal step can resolve it unlike CGS2. The reason why MGS2 outperforms CGS2 is that MGS orthogonalizes vectors sequentially, which means it adjusts the vector being orthogonalized step-by-step against each basis vector, while CGS orthogonalizes the vector in a single step against all previous vectors and then corrects it. Hence CGS process can allow significant numerical errors to accumulate before they are corrected, potentially reducing the overall accuracy. Notably, even without reorthogonalization, we can observe stability of RGS, e.g., $\kappa(Q_i) = 1 + \epsilon$. Furthermore, Figure 2b validates our round-off analyses of RGS-L2C and RGS-L2M, which impose l_2 orthonormality of Q_i.

5 Application to GMRES

In the classical GMRES method to solve $A\boldsymbol{x} = \boldsymbol{b}$, the orthonormal basis of Krylov subspace is computed by Arnoldi iterations associated with CGS or MGS. In a similar way, we can apply our RGS-L2 to Arnoldi iterations. In [5, 6], RGS is employed where the randomized (flexible) GMRES is introduced but the resulting basis vectors are not l_2 orthonormal. Therefore, the randomized GMRES is a quasi-optimal solver aimed at minimizing the residual in the sketched norm rather than l_2 norm. On the other hand, once we employ RGS-L2 to generate Krylov basis vectors, the corresponding GMRES is able to minimize the residual in the usual norm. For example, when we denote the residual vector of $m-$th iteration by $\boldsymbol{r}_m$,

$$\text{RGS-GMRES: minimize } \|\boldsymbol{r}_m\|_\Theta \,,$$

but

$$\text{RGS-L2-GMRES: minimize } \|\boldsymbol{r}_m\| \,.$$

Consequently, with the RGS-L2 variants of Arnoldi process, the randomized GMRES method fulfills the usual minimal residual principle:

$$\boldsymbol{b} - A\boldsymbol{x}_m \perp A\mathcal{K}_m(A, \boldsymbol{r}_0),$$

ensuring that the approximate solution $\boldsymbol{x}_m$ is an optimal minimizer of the residual norm.

Jang et al. [6] developed the randomized FGMRES-DR to accelerate the convergence of GMRES. To construct a set of Ritz pairs to be disregarded with RGS, it is necessary to introduce the randomized Rayleigh-Ritz method, since the provided bases of a subspace are not l_2 orthonormal but Θ orthonormal. However, with RGS-L2, we can immediately follow the harmonic Ritz pair formulation of Morgan[22]. Therefore, we want to find a pair $(\boldsymbol{y}, \lambda)$ such that satisfies

$$\boldsymbol{y} \in \mathcal{S} \qquad \text{and} \qquad B\boldsymbol{y} - \lambda\boldsymbol{y} \perp \mathcal{S}, \tag{5.1}$$

where $B = A^{-1}$ and $\mathcal{S} = A\mathcal{K}_m(A, \boldsymbol{b})$. In case of the randomized Rayleigh-Ritz formulation, $\perp$ in (5.1) will be replaced by $\perp_\Theta$ indicating sketched orthogonality. To obtain the harmonic Ritz pairs, we consider the following generalized eigenvalue

problem:

$$H_m^T H_m \boldsymbol{g} = \lambda \hat{H}_m^T \boldsymbol{g}, \tag{5.2}$$

where $\hat{H}_m = H_m(1:m, 1:m)$ and $\boldsymbol{y} = V_m \boldsymbol{g}$. If $\hat{H}^T$ is invertible, solving (5.2) is equivalent to solving

$$\left(\hat{H}_m + h_{m+1,m}^2 \hat{H}_m^{-T} \boldsymbol{e}_m \boldsymbol{e}_m^T\right) \boldsymbol{g} = \lambda \boldsymbol{g}, \tag{5.3}$$

where $h_{m+1,m} = H_m(m+1, m)$ and $\boldsymbol{e}_m$ is the m-th Cartesian basis vector of $\mathbb{R}^m$. We refer for the analysis of the deflated restarting to [22, 23].

By performing Arnoldi iterations associated with one of GS algorithms, we can generate a set of Krylov basis vectors. For example, we can define orthonormal $V_{m+1} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{m+1}]$ such that it satisfies

$$\operatorname{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{m+1}\} = \operatorname{span}\{\boldsymbol{r}_0, A\boldsymbol{r}_0 \ldots, \boldsymbol{A}^m \boldsymbol{r}_0\}.$$

In addition, we have the Arnoldi identity, $AV_m = V_{m+1} H^m$. In GMRES, using V_{m+1}, we can find an approximate solution for minimizing the residual norm as detailed in Algorithm 3 for GMRES(-DR).

Algorithm 3 GMRES with (deflated) restarting: GMRES-DR (m, k)

Input: matrix $A \in \mathbb{R}^{n\times n}$, vector $\boldsymbol{b} \in \mathbb{R}^n$, sketching matrix $\Theta \in \mathbb{R}^{t\times n}$, size of Krylov subspace m, number of deflated vectors k, tolerance $tol > 0$, and vector $\boldsymbol{x}_0 \in \mathbb{R}^n$.
Output: approximate solution $\boldsymbol{x}$ for $A\boldsymbol{x} = \boldsymbol{b}$.

$\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$; $\beta = \|\boldsymbol{r}_0\|$; $\boldsymbol{c} = [\beta, O_{1\times m}]^T$; $\boldsymbol{e}_m = [O_{1\times(m-1)}, 1]^T$.
Perform Arnoldi process to get V_{m+1} and H_m with the starting vector $\boldsymbol{r}_0/\beta$.
$\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}\in\mathbb{R}^m} \|\boldsymbol{c} - H_m\boldsymbol{y}\|$; $\boldsymbol{x}_m = \boldsymbol{x}_0 + V_m\boldsymbol{y}^*$; $\boldsymbol{x}_0 = \boldsymbol{x}_m$; $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$; $\beta = \|\boldsymbol{r}_0\|$; $\boldsymbol{\rho} = \boldsymbol{c} - H_m\boldsymbol{y}^*$.
while $\beta/\|\boldsymbol{b}\| > tol$ **do**
  if $k > 0$ **then**
    $h = H_m(m+1, m)$; $\hat{H}_m = H_m(1:m, 1:m)$.
    Compute k harmonic Ritz vectors by solving the eigenvalue problem:

$$\left(\hat{H}_m + h^2 \hat{H}_m^{-H} \boldsymbol{e}_m \boldsymbol{e}_m^T\right) \boldsymbol{g}_i = \lambda_i \boldsymbol{g}_i \qquad \text{for} \qquad i = 1, \ldots, k.$$

    Set $G_k = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k]$; $G_{k+1} = \begin{bmatrix} \begin{bmatrix} G_k \\ O_{1\times k} \end{bmatrix}, \boldsymbol{\rho} \end{bmatrix}$.
    Perform QR decomposition on G_{k+1}: $G_{k+1} = Q_{k+1}R_{k+1}$.
    Define V_{k+1} and H_k to satisfy $AV_k = V_{k+1}H_k$ by $V_{k+1} = V_{m+1}Q_{k+1}$; $H_k = Q_{k+1}^H H_m Q_{k+1}(1:m, 1:k)$.
    Update $\boldsymbol{c} = \begin{bmatrix} Q_{k+1}^T\boldsymbol{\rho} \\ O_{(m-k)\times 1} \end{bmatrix}$.
  else
    $\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$; $\boldsymbol{c} = \begin{bmatrix} \beta \\ O_{m\times 1} \end{bmatrix}$.
  end if
  Perform $(m-k)$ steps of Arnoldi process with V_{k+1} (if $k = 0$, $V_1 = \boldsymbol{v}_1$).
  Solve $\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}\in\mathbb{R}^m} \|\boldsymbol{c} - H_m\boldsymbol{y}\|$; $\boldsymbol{x}_m = \boldsymbol{x}_0 + V_m\boldsymbol{y}^*$.
  Update $\boldsymbol{x}_0 = \boldsymbol{x}_m$; $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$; $\beta = \|\boldsymbol{r}_0\|$; $\boldsymbol{\rho} = \boldsymbol{c} - H_m\boldsymbol{y}^*$.
end while
$\boldsymbol{x} = \boldsymbol{x}_m$.

The minimization problem for $\boldsymbol{r}_m$ is equivalent to solving least squares problem. For instance, using orthonormality of V_{m+1} and the Arnoldi identity, we have

$$\text{minimize } \|\boldsymbol{r}_m\| \Leftrightarrow \text{minimize } \|\boldsymbol{r}_0 - AV_m\boldsymbol{y}\| \Leftrightarrow \text{minimize } \|\boldsymbol{r}_0 - V_{m+1}H_m\boldsymbol{y}\| \Leftrightarrow \text{solve } \boldsymbol{y}^* = \arg\min_{\boldsymbol{y}\in\mathbb{R}^m} \left\|V_{m+1}^T\boldsymbol{r}_0 - H_m\boldsymbol{y}\right\|.$$

To simplify the least squares problem, we can reduce $V_{m+1}^T\boldsymbol{r}_0$ to $[\|\boldsymbol{r}_0\|, O_{1\times m}]$, since $\boldsymbol{r}_0 \perp \boldsymbol{v}_j$ for $j \leq 2$. When we employ the deflation formulation, $V_{m+1}^T\boldsymbol{r}_0$ can be rewritten as

$$V_{m+1}^T\boldsymbol{r}_0 = V_{m+1}^T V_{k+1} Q_{k+1}^T \boldsymbol{\rho} = \begin{bmatrix} Q_{k+1}^T\boldsymbol{\rho} \\ O_{(m-k)\times 1} \end{bmatrix},$$

by orthogonality of V_{m+1}. For more details, we refer to [23].

As seen above, the orthogonality of V_{m+1} plays an important role to derive least squares problems. In finite precision arithmetic, the problem of minimizing a residual norm will no longer be the same as the least squares problem if V_{m+1} does not maintain orthogonality. Furthermore, in GMRES-DR, the loss of orthogonality will be larger than that in GMRES due to additional QR decomposition for G_{k+1}. Therefore, GMRES and GMRES-DR require high-quality of orthogonalizing process.

5.1 Numerical experiments

We present some numerical examples of solving ill-conditioned linear systems with respect to different GS types. More precisely, we consider the matrix A from `SuiteSparse Matrix Collection` (`https://sparse.tamu.edu/Janna/ML_Geer`), namely `ML_Geer`. The right hand side $\boldsymbol{b}$ is defined by $\boldsymbol{b} = A\boldsymbol{b}_0/\|A\boldsymbol{b}_0\|$ where $\boldsymbol{b}_0 = [1\ldots,1]^T \in \mathbb{R}^n$. In this numerical experiment, we apply *incomplete LU* (ILU0) preconditioner to GMRES and GMRES-DR. We compare numerical stability and convergence of our proposed methods with the existing classical methods. We set `tol`=1e-08, $m = 400$, $k = 40$ if deflated restarting employed, and a zero initial vector $\boldsymbol{x}_0 = \boldsymbol{0}$. To satisfy the subspace embedding properties, the sketch size is set to $t = 1900$.

Table 3: Averaged condition numbers and loss of orthogonality for V_{m+1} during simulations.

GS type	Deflation	Condition number	Loss of orthogonality
CGS	No	1.94e+17	2.72e+02
	Yes	1.67e+17	1.92e+03
RGS-L2C	No	1.00	4.98e-14
	Yes	1.00	8.45e-14
RGS-L2M	No	1.00	5.00e-14
	Yes	1.00	7.81e-14

In Table 3, we compute the average values of the condition number of V_{m+1}, $\kappa(V_{m+1})$, and the loss of orthogonality, $\left\|I_{m+1} - V_{m+1}^T V_{m+1}\right\|$, for each GS algorithm. It is observed that the CGS based Arnoldi iterations exhibit significant instability in V_{m+1} as well as a loss of orthogonality. In contrast, employing both RGS-L2 algorithms provide stable V_{m+1} with sufficient numerical orthogonality, typically around $O(10^{-14})$, regardless of the deflation strategy.

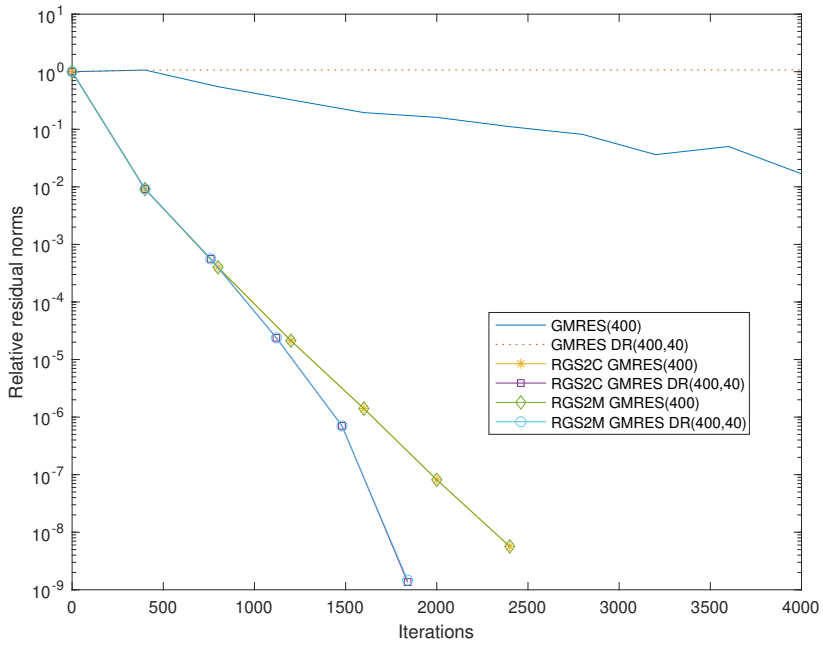


Figure 3: Convergence rates of GMRES and GMRES-DR with respect to GS algorithms.

Figure 3 illustrates the relative residual norm with respect to Arnoldi iteration for each GMRES methods. In case of using CGS based Arnoldi processes in GMRES, the poor quality of V_{m+1} leads to significantly worse convergence rates compared to RGS-L2 based GMRES. Moreover, while the deflated restarting improves convergence with the RGS-L2 algorithms, GMRES-DR with CGS exhibits residual norm stagnation during simulation. This highlights how crucial it is to maintain orthogonality in GMRES(-DR). Therefore, our proposed randomized methods, ensuring high-quality orthogonal processes, are expected to enhance numerical performance significantly when combined with GMRES-DR.

6 Conclusion

In this study, we explored the efficiency and stability of various Gram-Schmidt (GS) orthogonalization algorithms, in the context of solving ill-conditioned linear systems using GMRES and GMRES-DR methods. Our findings emphasize the critical role of orthogonalization quality in iterative solvers like GMRES, especially in scenarios involving ill-conditioned matrices.

Our new randomized Gram-Schmidt algorithms enable to generate a set of fully l_2 orthogonal vectors where the reorthogonalizing process led to improvement in numerical stability. The rounding error analysis indicates the decent quality of orthogonality in Q factor as other reorthogonalized methods, while the proposed method exhibits reduced computational complexity. Furthermore, as shown in the numerical examples, employing RGS-L2 improves numerical performance. Specifically, when solving linear systems by GMRES-DR, our method significantly enhances the convergence rates by ensuring the orthogonality of Krylov basis vectors.

References

[1] Achlioptas D. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences.* 2003;66(4):671–687.

[2] Halko N, Martinsson PG, Tropp JA. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review.* 2011;53(2):217–288.

[3] Woodruff DP. Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends in Theoretical Computer Science.* 2014;10(12):1-157. doi: 10.1561/0400000060

[4] Balabanov O, Nouy A. Randomized linear algebra for model reduction. Part I: Galerkin methods and error estimation. *Advances in Computational Mathematics.* 2019;45(5):2969–3019.

[5] Balabanov O, Grigori L. Randomized Gram–Schmidt process with application to GMRES. *SIAM Journal on Scientific Computing.* 2022;44(3):A1450–A1474.

[6] Jang Y, Grigori L, Martin E, Content C. Randomized Flexible GMRES with Deflated Restarting. *Numerical Algorithms.* 2023.

[7] Nakatsukasa Y, Tropp JA. Fast and accurate randomized algorithms for linear systems and eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications.* 2024;45(2):1183–1214.

[8] Burke L, Güttel S, Soodhalter KM. GMRES with randomized sketching and deflated restarting. *arXiv preprint arXiv:2311.14206.* 2023.

[9] Giraud L, Langou J, Rozloznik M. The loss of orthogonality in the Gram-Schmidt orthogonalization process. *Computers & Mathematics with Applications.* 2005;50(7):1069–1075.

[10] Giraud L, Langou J. When modified Gram–Schmidt generates a well-conditioned set of vectors. *IMA Journal of Numerical Analysis.* 2002;22(4):521–528.

[11] Giraud L, Langou J, Rozložník M, Eshof Jvd. Rounding error analysis of the classical Gram-Schmidt orthogonalization process. *Numerische Mathematik.* 2005;101(1):87–100.

[12] Björck Å, Paige CC. Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm. *SIAM journal on matrix analysis and applications.* 1992;13(1):176–190.

[13] Giraud L, Langou J. A Robust Criterion for the Modified Gram–Schmidt Algorithm with Selective Reorthogonalization. *SIAM Journal on Scientific Computing.* 2003;25(2):417–441.

[14] Björck Å. *Numerical methods for least squares problems*. SIAM, 1996.

[15] Higham NJ. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[16] Damas dJG, Grigori L. Randomized Implicitly Restarted Arnoldi method for the non-symmetric eigenvalue problem. *arXiv preprint arXiv:2407.03208.* 2024.

[17] Balabanov O. Randomized Cholesky QR factorizations. *arXiv preprint arXiv:2210.09953.* 2022.

[18] Connolly MP, Higham NJ, Mary T. Stochastic rounding and its probabilistic backward error analysis. *SIAM Journal on Scientific Computing.* 2021;43(1):A566–A585.

[19] Croci M, Fasi M, Higham NJ, Mary T, Mikaitis M. Stochastic rounding: implementation, error analysis and applications. *Royal Society Open Science.* 2022;9(3):211631.

[20] Björck Å. Solving linear least squares problems by Gram-Schmidt orthogonalization. *BIT Numerical Mathematics.* 1967;7(1):1–21.

[21] Hoffmann W. ITERATIVE ALGORITHMS FOR Gram-Schmidt ORTHOGONALIZATION. *Computing.* 1989;41:335–348.

[22] Morgan RB. GMRES with deflated restarting. *SIAM Journal on Scientific Computing.* 2002;24(1):20–37.

[23] Giraud L, Gratton S, Pinel X, Vasseur X. Flexible GMRES with deflated restarting. *SIAM Journal on Scientific Computing.* 2010;32(4):1858–1878.