SIAM J. SCI. COMPUT.
Vol. 44, No. 3, pp. A1450–A1474

RANDOMIZED GRAM–SCHMIDT PROCESS WITH APPLICATION TO GMRES[*]

OLEG BALABANOV[†] AND LAURA GRIGORI[†]

Abstract. A randomized Gram–Schmidt algorithm is developed for orthonormalization of high-dimensional vectors or QR factorization. The proposed process can be less computationally expensive than the classical Gram–Schmidt process while being at least as numerically stable as the modified Gram–Schmidt process. Our approach is based on random sketching, which is a dimension reduction technique consisting in estimation of inner products of high-dimensional vectors by inner products of their small efficiently computable random images, so-called sketches. In this way, an approximate orthogonality of the full vectors can be obtained by orthogonalization of their sketches. The proposed Gram–Schmidt algorithm can provide computational cost reduction in any architecture. The benefit of random sketching can be amplified by performing the nondominant operations in higher precision. In this case the numerical stability can be guaranteed with a working unit roundoff independent of the dimension of the problem. The proposed Gram–Schmidt process can be applied to Arnoldi iteration and results in new Krylov subspace methods for solving high-dimensional systems of equations or eigenvalue problems. Among them we chose the randomized GMRES method as a practical application of the methodology.

Key words. Gram–Schmidt orthogonalization, QR factorization, randomization, random sketching, numerical stability, rounding errors, loss of orthogonality, multiprecision arithmetic, Krylov subspace methods, Arnoldi iteration, generalized minimal residual method

AMS subject classifications. 15B52, 65G50, 65Fxx

DOI. 10.1137/20M138870X

1. Introduction.

The orthonormalization of a set of high-dimensional vectors serves as basis for many algorithms in numerical linear algebra and other fields of science and engineering. The Gram–Schmidt (GS) process is one of the easiest and most powerful methods to perform this task.

The numerical stability of the standard implementations of GS, which are the classical GS algorithm (CGS) and the modified GS algorithm (MGS), were analyzed in [1,7]. The analysis of CGS was improved in [12,13]. In [1,13,22,30] the authors discussed more sophisticated variants of the GS process and in particular the CGS algorithm with re-orthogonalization (CGS2). Versions of the GS process well-suited for modern extreme-scale computational architectures were developed in [23,31].

In this article we propose a probabilistic way to reduce the computational cost of the GS process by using the random sketching technique [16,28,34]. This approach recently became a popular tool for solving high-dimensional problems arising in such fields as theoretical computer science, signal processing, data analysis, model order reduction, and machine learning [33,34]. The key idea of random sketching technique relies in the estimation of inner products of high-dimensional vectors by inner products of their low-dimensional images through a random matrix. The random sketching matrix is chosen depending on the computational architecture so that it

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section December 28, 2020; accepted for publication (in revised form) January 5, 2022; published electronically June 6, 2022.
https://doi.org/10.1137/20M138870X
Funding: This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program through grant agreement 810367).

[†] Alpines, Inria, Sorbonne Université, Université de Paris, CNRS, Laboratoire Jacques-Louis Lions, F-75012 Paris, France (oleg.balabanov@inria.fr, Laura.Grigori@inria.fr).

can be efficiently applied to a vector. In this way, one is able to efficiently embed a set (or a subspace) of high-dimensional vectors, defining the problem of interest, into a low-dimensional space and then tackle the problem in this low-dimensional space. In the context of the GS process, this implies orthogonalizing the sketches rather than high-dimensional vectors. Along with the randomized variant of the GS process, here referred to as the randomized GS process (RGS), we also provide precise conditions on the sketch to guarantee the approximate orthogonality of the output vectors in finite precision arithmetic. They rely on the ε-embedding property of the random sketching matrix for the subspace spanned by the output vectors. This property is shown to hold for standard random matrices with high probability if the set of vectors to be orthogonalized is provided a priori. Furthermore, an efficient procedure for the a posteriori certification of the ε-embedding property is presented. Besides the certification of the output, this procedure can be used for the adaptive selection of the size of the random sketching matrix or for improving the robustness of algorithms as depicted in Remark 3.10.

Furthermore, we show how the efficiency gains of the RGS algorithm can be amplified by using a multiprecision arithmetic. In particular, it is proposed to perform expensive high-dimensional operations in low precision, which represents the working precision, while computing the efficient random projections and low-dimensional operations in high precision. By exploiting statistical properties of rounding errors [9, 19], we are able to prove the stability of RGS for the working precision unit roundoff independent of the high dimension of the problem. Clearly, the presented analysis directly implies stability guarantees also for the unique precision model.

The randomization entails a possible failure of an algorithm. The probability of this happening, however, is a user-specified parameter that can be chosen very small (e.g., 10^{-10}) without considerable impact on the overall computational costs.

One of the uses of the GS process is the computation of an orthonormal basis of a Krylov subspace. This procedure may be used for the solution of high-dimensional eigenvalue problems or systems of equations. In the context of minimal residual methods, such an approach is, respectively, referred to as the Arnoldi iteration for eigenvalue problems or the generalized minimal residual (GMRES) method for linear systems of equations. For the presentation of these methods, see section 4. The numerical properties of GMRES were analyzed in [10, 15, 24, 26]. The usage of variable (or multi) precision arithmetic for Krylov methods, and in particular GMRES, was discussed in [8, 11, 14, 29, 32, 35]. In the present article we chose the GMRES method as a practical application of the RGS algorithm.

The organization of the article is as follows. In subsection 1.1 we describe the basic notations. Subsection 1.2 introduces a general GS process and particularizes it to few classical variants. Section 2 at first discusses the general idea of the random sketching technique. Then in subsection 2.2, we analyze the rounding errors of a sketched matrix-vector product. A version of the GS process, based on random sketching, is proposed in subsection 2.3. Its performance in different computational architectures is then studied in subsection 2.4. Section 3 is devoted to the a priori as well as a posteriori stability analysis of the RGS process. Section 4 discusses the incorporation of the methodology into the Arnoldi iteration and GMRES algorithms. Section 5 provides the experimental validation of proposed algorithms. Finally, section 6 concludes the article.

For better presentation most of the proofs of theorems and propositions are provided as supplementary material (M138870SupMat.pdf [local/web 390KB]).

1.1. Preliminaries. Throughout the manuscript we work with real numbers noting that the presented methodology can be naturally extended to complex numbers.

Algebraic vectors are here denoted by bold lowercase letters, e.g., letter $\mathbf{x}$. For given vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$, we denote matrix $[\mathbf{x}_1, \ldots, \mathbf{x}_k]$ by $\mathbf{X}_k$ (with a bold capital letter) and the (i,j)th entry of $\mathbf{X}_k$ by $x_{i,j}$ (with a lowercase letter). The notation $\mathbf{X}_k$ can be further simplified to $\mathbf{X}$ if k is constant. Furthermore, we let $[\mathbf{X}_k]_{(N_1:N_2,M_1:M_2)}$ denote the block of entries $x_{i,j}$ of $\mathbf{X}_k$ with $(i,j) \in \{N_1, N_1+1, \ldots, N_2\} \times \{M_1, M_1+1, \ldots, M_2\}$. For a special case of $M_1 = M_2$, we denote the vector $[\mathbf{X}_k]_{(N_1:N_2,M_1:M_2)}$ by simply $[\mathbf{X}_k]_{(N_1:N_2,M_1)}$. Moreover, if $\mathbf{X}_k$ is a vector, $[\mathbf{X}_k]_{(N_1:N_2,1)}$ is denoted by $[\mathbf{X}_k]_{(N_1:N_2)}$. The minimal and the maximal singular values of $\mathbf{X}$ are denoted by $\sigma_{min}(\mathbf{X})$ and $\sigma_{max}(\mathbf{X})$ and the condition number by $\mathrm{cond}(\mathbf{X})$. We let $\langle \cdot, \cdot \rangle$ and $\|\cdot\| = \sigma_{max}(\cdot)$ be the ℓ_2-inner product and ℓ_2-norm, respectively. $\|\cdot\|_\mathrm{F}$ denotes the Frobenius norm. For two matrices (or vectors) $\mathbf{X}$ and $\mathbf{Y}$ we say that $\mathbf{X} \leq \mathbf{Y}$ if the entries of $\mathbf{X}$ satisfy $x_{i,j} \leq y_{i,j}$. Furthermore, for a matrix (or a vector) $\mathbf{X}$, we denote by $|\mathbf{X}|$ the matrix $\mathbf{Y}$ with entries $y_{i,j} = |x_{i,j}|$. We also let $\mathbf{X}^\mathrm{T}$ and $\mathbf{X}^\dagger$, respectively, denote the transpose and the Moore–Penrose inverse of $\mathbf{X}$. Finally, we let $\mathbf{I}_{k \times k}$ be the $k \times k$ identity matrix.

For a quantity or an arithmetic expression, X, we use notation $\mathrm{fl}(X)$ or $\hat{X}$ to denote the computed value of X with finite precision arithmetic.

1.2. GS process. The GS process is a method to orthonormalize a set of vectors or compute QR factorization of a matrix. We are concerned with a column-oriented variant of the process. It proceeds recursively, at each iteration selecting a new vector from the set and orthogonalizing it with respect to the previously selected vectors, as is depicted in Algorithm 1.1.

The projector $\mathbf{\Pi}^{(j)}$ in Algorithm 1.1 is usually taken as approximation to the (ℓ_2-)orthogonal projector $\mathbf{I}_{n \times n} - \mathbf{Q}_j(\mathbf{Q}_j)^\dagger$ onto $\mathrm{span}(\mathbf{Q}_j)^\perp$, $1 \leq j \leq m-1$. If Algorithm 1.1 is used with infinite precision arithmetic, then considering

$$\mathbf{\Pi}^{(j)} = \mathbf{I}_{n \times n} - \mathbf{Q}_j(\mathbf{Q}_j)^\mathrm{T} \tag{1.1}$$

will produce an exact QR factorization of $\mathbf{W}$. This fact can be shown by induction. In short, we can show that $\mathbf{\Pi}^{(i-1)}$ being an orthogonal projector implies $\mathbf{Q}_i$ being an orthonormal matrix, which in its turn implies that $\mathbf{\Pi}^{(i)}$ is an orthonormal projector. Algorithm 1.1 with the choice (1.1) is referred to as the CGS process. With finite precision arithmetic, however, matrix $\mathbf{Q}_j$ can be guaranteed to be orthonormal only approximately. This can make the CGS algorithm suffer from numerical instabilities. In particular, in this case the orthogonality of Q factor, measured by $\|\mathbf{I}_{m \times m} - \mathbf{Q}^\mathrm{T}\mathbf{Q}\|$, can grow as $\mathrm{cond}(\mathbf{W})^2$ or more, depending on the method used for normalization [7, 30].

Algorithm 1.1. GS process

```
Given: $n \times m$ matrix $\mathbf{W}$, $m \leq n$
Output: $n \times m$ factor $\mathbf{Q}$ and $m \times m$ upper triangular factor $\mathbf{R}$.
for $i = 1 : m$ do
    1. Compute a projection $\mathbf{q}_i = \mathbf{\Pi}^{(i-1)}\mathbf{w}_i$ (also yielding $[\mathbf{R}]_{(1:i-1,i)}$).
    2. Normalize $\mathbf{q}_i$ (also yielding $r_{i,i}$).
end for
```

Besides the projector (1.1), there are a couple of other standard choices for $\mathbf{\Pi}^{(j)}$. They can yield a better numerical stability but require more computational cost in terms of flops, storage consumption, scalability, or amount of communication between processors. The MGS algorithm uses the projector

$$\mathbf{\Pi}^{(j)} = \left(\mathbf{I}_{n\times n} - \mathbf{q}_j(\mathbf{q}_j)^{\mathrm{T}}\right)\left(\mathbf{I}_{n\times n} - \mathbf{q}_{j-1}(\mathbf{q}_{j-1})^{\mathrm{T}}\right)\dots\left(\mathbf{I}_{n\times n} - \mathbf{q}_1(\mathbf{q}_1)^{\mathrm{T}}\right),\ 1 \le j \le m-1.$$

In this case the orthogonality measure of $\mathbf{Q}$ depends only linearly on cond($\mathbf{W}$) [7]. Numerical stability of the MGS algorithm is sufficient for most applications and is often considered as benchmark for characterizing the stability of algorithms for orthogonalizing a set of vectors or computing a QR factorization. Another choice for $\mathbf{\Pi}^{(j)}$ is

$$\mathbf{\Pi}^{(j)} = \left(\mathbf{I}_{n\times n} - \mathbf{Q}_j(\mathbf{Q}_j)^{\mathrm{T}}\right)\left(\mathbf{I}_{n\times n} - \mathbf{Q}_j(\mathbf{Q}_j)^{\mathrm{T}}\right),\ 1 \le j \le m-1,$$

which results in a so-called CGS process with re-orthogonalization (CGS2). This projector can be shown to yield a similar (or better) stability as the MGS process.

In this work we are concerned with a scenario when $\mathbf{W}$ is a large matrix with a moderate number of columns, i.e., when $m \ll n$. For this situation, we propose a new *randomized* projector $\mathbf{\Pi}^{(j)}$ that can yield more efficiency than the CGS process while providing no less numerical stability than the MGS process. Unlike standard approaches, our RGS algorithm provides a Q factor that is not ℓ_2-orthogonal even under exact arithmetic but that is very well-conditioned with very high probability. This property is sufficient for a number of applications. For instance, as is shown in subsection 4.2, a small condition number of the Q factor guarantees an almost optimal convergence of the GMRES solution. For other cases, the Q factor produced by the RGS algorithm should be post processed with a Cholesky QR.

2. RGS algorithm.

2.1. Introduction to random sketching.

Let $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$, with $k \ll n$, be a sketching matrix. This matrix shall be seen as an embedding of subspaces of $\mathbb{R}^n$ into subspaces of $\mathbb{R}^k$ and is therefore referred to as a ℓ_2-subspace embedding. The ℓ_2-inner products between vectors in subspaces of $\mathbb{R}^n$ are estimated by

$$\langle\cdot,\cdot\rangle \approx \langle\mathbf{\Theta}\cdot,\mathbf{\Theta}\cdot\rangle.$$

For a given (low-dimensional) subspace of interest $V \subset \mathbb{R}^n$, the quality of such an estimation can be characterized by the following property of $\mathbf{\Theta}$.

Definition 2.1. *For $\varepsilon < 1$, the sketching matrix $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$ is said to be an ε-subspace embedding for $V \subset \mathbb{R}^n$ if we have*

$$\forall \mathbf{x},\mathbf{y} \in V, \quad |\langle\mathbf{x},\mathbf{y}\rangle - \langle\mathbf{\Theta x},\mathbf{\Theta y}\rangle| \le \varepsilon\|\mathbf{x}\|\|\mathbf{y}\|. \tag{2.1}$$

Let $\mathbf{V}$ be a matrix whose columns form a basis for V. To ease presentation in the next sections, an ε-subspace embedding for V shall be often referred to simply as an ε-embedding for $\mathbf{V}$.

Corollary 2.2. *If $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$ is an ε-embedding for $\mathbf{V}$, then the singular values of $\mathbf{V}$ are bounded by*

$$(1+\varepsilon)^{-1/2}\sigma_{min}(\mathbf{\Theta V}) \ \le \sigma_{min}(\mathbf{V}) \le \sigma_{max}(\mathbf{V}) \le (1-\varepsilon)^{-1/2}\sigma_{max}(\mathbf{\Theta V}).$$

Proof. Let $\mathbf{a} \in \mathbb{R}^{\dim(V)}$ be an arbitrary vector and $\mathbf{x} = \mathbf{Va}$. By definition of $\mathbf{\Theta}$,

$$(1+\varepsilon)^{-1}\|\mathbf{\Theta x}\|^2 \leq \|\mathbf{x}\|^2 \leq (1-\varepsilon)^{-1}\|\mathbf{\Theta x}\|^2, \text{ which implies that}$$

$$(1+\varepsilon)^{-1/2}\|\mathbf{\Theta Va}\| \leq \|\mathbf{Va}\| \leq (1-\varepsilon)^{-1/2}\|\mathbf{\Theta Va}\|.$$

The statement of proposition then follows by using definitions of the minimal and the maximal singular values of a matrix. □

Corollary 2.2 implies that to make the condition number of matrix $\mathbf{V}$ close to 1, it can be sufficient to orthonormalize small sketched matrix $\mathbf{\Theta V}$. This observation serves as basis for the RGS process in subsection 2.3. Note that the orthogonalization of $\mathbf{\Theta V}$ with respect to the ℓ_2-inner product is equivalent to orthonormalization of $\mathbf{V}$ with respect to the product $\langle \mathbf{\Theta}\cdot, \mathbf{\Theta}\cdot\rangle$. Note also that in our applications there will be no practical benefit of considering very small values for ε. The usage of $\varepsilon \leq 1/2$ or $\varepsilon \leq 1/4$ will be sufficient.

We here proceed with sketching matrices that do not require any a priori knowledge of V to guarantee (2.1). Instead, $\mathbf{\Theta}$ is generated from a carefully chosen distribution such that it satisfies (2.1) for any low-dimensional subspace with high probability.

Definition 2.3. *The sketching matrix $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$ is called a (ε, δ, d) oblivious ℓ_2-subspace embedding if it is an ε-embedding for any fixed d-dimensional subspace $V \subset \mathbb{R}^n$ with probability at least $1-\delta$.*

In general, such *oblivious subspace embeddings* with high probability have a bounded norm, as is shown in Corollary 2.4.

Corollary 2.4. *If $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$ is a $(\varepsilon, \delta/n, 1)$ oblivious ℓ_2-subspace embedding, then with probability at least $1-\delta$, we have*

$$\|\mathbf{\Theta}\|_{\mathrm{F}} \leq \sqrt{(1+\varepsilon)n}.$$

Proof. It directly follows from Definition 2.3 and the union bound argument that $\mathbf{\Theta}$ is an ε-embedding for each canonical (Euclidean) basis vector. This implies that the ℓ_2-norms of the columns of $\mathbf{\Theta}$ are bounded from above by $\sqrt{1+\varepsilon}$. The statement of the corollary then follows immediately. □

There are several distributions that are known to satisfy the (ε, δ, d) oblivious ℓ_2-subspace embedding property when k is sufficiently large. The standard examples include Gaussian, Rademacher distributions, subsampled randomized Hadamard transform (SRHT) and Fourier transform, CountSketch matrix, and more [2, 16, 34]. In this work we shall rely on Rademacher matrices and partial SRHT (P-SRHT). A (rescaled) Rademacher matrix has independent and identically distributed entries equal to $\pm 1/\sqrt{k}$ with probabilities $1/2$. The efficiency of multiplication by Rademacher matrices can be attained due to proper exploitation of computational architectures. For instance, the products of Rademacher matrices with vectors can be implemented with standard SQL primitives and are embarrassingly parallelizable. For n being a power of 2, SRHT is defined as a product of a diagonal matrix of random signs with a Walsh–Hadamard matrix, followed by a uniform subsampling matrix and scaling factor $1/\sqrt{k}$. Random sketching with SRHT can improve efficiency in terms of number of flops. Products of SRHT matrices with vectors require only $n\log_2(n)$ flops using the fast Walsh–Hadamard transform or $2n\log_2(k+1)$ flops using the procedure in [3]. P-SRHT is used instead of SRHT when n is not a power of 2 and is defined as the first n columns of an SRHT matrix of size s, where s is the power of 2 such that $n \leq s < 2n$. Furthermore, for both (P-)SRHT and Rademacher matrices a seeded random number generator can be utilized to allow efficient storage and application of $\mathbf{\Theta}$. This is particularly important for limited-memory and distributed computational architectures. It follows that the rescaled Rademacher distribution with

$$k \geq 7.87\varepsilon^{-2}(6.9d + \log(1/\delta)) \tag{2.2a}$$

and the P-SRHT distribution with

$$k \geq 2(\varepsilon^2 - \varepsilon^3/3)^{-1}\left(\sqrt{d} + \sqrt{8\log(6n/\delta)}\right)^2 \log(3d/\delta), \tag{2.2b}$$

respectively, are (ε, δ, d) oblivious ℓ_2-subspace embeddings [4]. We see that the bounds (2.2) are independent or only logarithmically dependent on the dimension n and probability of failure and are proportional to the low dimension d. This implies that one can use $\mathbf{\Theta}$ of a small size even for very large problems and very small probabilities of failure.

2.2. Rounding errors in a sketched matrix-vector product.

Let us fix a realization of an oblivious ℓ_2-subspace embedding $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$ of sufficiently large size and consider a matrix-vector product

$$\mathbf{x} = \mathbf{Y}\mathbf{z} \text{ with } \mathbf{Y} \in \mathbb{R}^{n\times m},\ \mathbf{z} \in \mathbb{R}^m,$$

computed in finite precision arithmetic with unit roundoff $u < 0.01/m$. Note that elementary linear algebra operations on vectors such as addition or multiplication by a constant can be also viewed as matrix-vector products. Define rounding error vector $\mathbf{\Delta x} = \widehat{\mathbf{x}} - \mathbf{x}$. The standard worst-case scenario rounding analysis provides an upper bound for $\mathbf{\Delta x}$ of the following form [18]:

$$|\mathbf{\Delta x}| \leq \mathbf{u}. \tag{2.3}$$

In a general case, the vector $\mathbf{u}$ can be taken as[1]

$$\mathbf{u} = 1.02mu|\mathbf{Y}||\mathbf{z}|. \tag{2.4}$$

In some situations, e.g., if the matrix $\mathbf{Y}$ is sparse, this bound can be improved. Here, we are particularly interested in the case when $\mathbf{Y} = a\mathbf{I}_{n\times n}$, i.e., when $\mathbf{Y}\mathbf{z}$ represents a multiplication of $\mathbf{z}$ by a constant, and when $\mathbf{Y}\mathbf{z} = \mathbf{Y}'\mathbf{z}' + \mathbf{h}$, i.e., when it represents a sum of a matrix-vector product with a vector. Then in the first case, one can take

$$\mathbf{u} = u|a\mathbf{z}|,$$

and in the second case,[2]

$$\mathbf{u} = 1.02u(|\mathbf{h}| + m|\mathbf{Y}'||\mathbf{z}'|).$$

Let us now address bounding the rounding error of the sketch $\mathbf{\Theta}\widehat{\mathbf{x}}$. This will become particularly handy in section 3 to simplify stability analysis of the RGS algorithm proposed in subsection 2.3. We here seek a bound of the form

$$\|\mathbf{\Theta}\mathbf{\Delta x}\| \leq D\|\mathbf{u}\|, \tag{2.5}$$

[1] We here used the fact that $\frac{mu}{1-mu} \leq 1.02mu$.

[2] We have, by the standard worst-case scenario analysis,

$$|\mathbf{\Delta x}| \leq \frac{(m-1)u}{1-(m-1)u}|\mathbf{Y}'||\mathbf{z}'| + u\left(\left(1 + \frac{(m-1)u}{1-(m-1)u}\right)|\mathbf{Y}'||\mathbf{z}'| + |\mathbf{h}|\right) \leq 1.02u(|\mathbf{h}| + m|\mathbf{Y}'||\mathbf{z}'|).$$

where D is a coefficient possibly depending on n. Clearly, we have

$$\|\mathbf{\Theta}\mathbf{\Delta x}\| \leq \|\mathbf{\Theta}\|\|\mathbf{u}\|, \tag{2.6}$$

which combined with Corollary 2.4 implies that with high probability the relation (2.5) holds with $D = \mathcal{O}(\sqrt{n})$.[3]

Next we notice that taking $D = \mathcal{O}(\sqrt{n})$ accounts for a very improbable worst-case scenario and is pessimistic in practice. If $\mathbf{x}$ is independent of $\mathbf{\Theta}$, then with high probability the relation (2.5) holds for $D = \mathcal{O}(1)$.[4] Furthermore (as is argued below in detail), one can expect the relation (2.5), with $D = \mathcal{O}(1)$, to hold for $\mathbf{\Theta}$ of moderate size even when $\mathbf{x}$ depends on $\mathbf{\Theta}$ (i.e., when $\mathbf{Y}$ and $\mathbf{z}$ are chosen depending on $\mathbf{\Theta}$), since the rounding error vector $\mathbf{\Delta x}$ should in practice have only a minor correlation with $\mathbf{\Theta}$. This property can be viewed as a sketched version of the standard "rule of thumb" stating that in practice one can reduce the worst-case scenario error constants (e.g., constant $\gamma_n = \frac{nu}{1-nu}$ in [18]) by a factor of $\sqrt{n}$. It has an important meaning in the context of (oblivious) randomized algorithms: the sketching step does not in practice multiply the rounding errors by a factor depending on n. In other words, with random sketching one is able to efficiently reduce the dimension of the problem without a loss of numerical precision.

To provide a precise guarantee that (2.5) holds for $D = \mathcal{O}(1)$ we shall need to explore the properties of $\mathbf{\Delta x}$ as a vector of rounding errors. For this we shall consider a probabilistic rounding model, where

- the rounding errors ξ due to each elementary arithmetic operation x op y, i.e.,
$$\xi = \frac{\mathrm{fl}(x \text{ op } y) - (x \text{ op } y)}{(x \text{ op } y)} \text{ with op} = +, -, *, /,$$
are bounded random variables possibly depending on each other but are independently centered (i.e., have zero mean);
- the computation of each entry of $\widehat{\mathbf{x}}$ is done independently of other entries, in other words, the entries of $\mathbf{\Delta x}$ are drawn independently of each other.

This model corresponds to [9, Model 4.7]. Its particular case is the so-called stochastic rounding model (see [9]), which recently gained attention in the machine learning community to improve the accuracy and the efficiency of training neural networks. The analysis of standard numerical linear algebra algorithms and, in particular, the rigorous foundation of the "rule of thumb", with the probabilistic rounding model is provided in [9, 17, 19, 20]. Note that the rounding model used here does not assume the rounding errors to be independent random variables as in [17, 19, 20] but only mean-independent with zero mean, which is a weaker and more realistic assumption, as is argued in [9].

Let us deduce that under the described probabilistic model, the vector $\mathbf{\Delta x}$ has entries that are independent centered random variables. It then follows from Theorem 2.5 that $\mathbf{\Theta}$ shall satisfy (2.5) with $D = \mathcal{O}(1)$ with probability at least $1 - 2\delta$ if $\mathbf{\Theta}$ is a $(\varepsilon, \binom{n}{d}^{-1}\delta, d)$, with $d = \mathcal{O}(\log(1/\delta))$, oblivious ℓ_2-subspace embedding. According to (2.2), this property is satisfied if $\mathbf{\Theta}$ is a Rademacher matrix with $\mathcal{O}(\log(n)\log(1/\delta))$ rows or P-SRHT matrix with $\mathcal{O}(\log^2(n)\log^2(1/\delta))$ rows.

[3] If $\mathbf{\Theta}$ is a $(\varepsilon, \delta/n, 1)$ oblivious subspace embedding, then we have $D = \sqrt{1+\varepsilon}\sqrt{n}$ with probability at least $1 - \delta$.

[4] If $\mathbf{\Theta}$ is a $(\varepsilon, \delta, 1)$ oblivious subspace embedding, then we have $D = \sqrt{1+\varepsilon}$ with probability at least $1 - \delta$.

Theorem 2.5. *Draw a realization* $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$ *of* $(\varepsilon/4, \binom{n}{d}^{-1}\delta, d)$ *oblivious* ℓ_2*-subspace embedding, with* $d = 4.2c^{-1}\log(4/\delta)$*, where* $c \leq 1$ *is some universal constant. Let* $\varphi \in \mathbb{R}^n$ *be a vector with entries that are independent random variables from distributions that can depend on* $\mathbf{\Theta}$*. If* φ *has zero mean, i.e.,* $E(\varphi|\mathbf{\Theta}) = \mathbf{0}$*, and* $|\varphi| \leq \gamma$ *for some vector* $\gamma \in \mathbb{R}^n$*, then*

$$||\varphi\|^2 - \|\mathbf{\Theta}\varphi\|^2| \leq \varepsilon\|\gamma\|^2 \tag{2.7}$$

holds with probability at least $1 - 2\delta$*.*

Proof. The proof of Theorem 2.5 will rely on the following property of $\mathbf{\Theta}$:

$$(1-\varepsilon)\|\mathbf{a}\|^2 \leq \|\mathbf{\Theta}\mathbf{a}\|^2 \leq (1+\varepsilon)\|\mathbf{a}\|^2 \text{ for all } d\text{-sparse vectors } \mathbf{a} \in \mathbb{R}^n, \tag{2.8}$$

called the restricted isometry property of level ε and order d, or simply (ε, d)-RIP.[5] This is a well-known fact that oblivious ℓ_2-subspace embeddings satisfy the RIP with high probability, as is shown in Proposition 2.8. The statement of Theorem 2.5 then follows by combining Proposition 2.8 with Theorem 2.9, and the union bound argument. □

Corollary 2.6. *Consider the probabilistic rounding model. If* $\mathbf{\Theta}$ *is a* $(\varepsilon/4, \binom{n}{d}^{-1}\delta, d)$ *oblivious* ℓ_2*-subspace embedding, with* $d = 4.2c^{-1}\log(4/\delta)$*, where* $c \leq 1$ *is some universal constant, then the bound* (2.5) *holds with* $D = \sqrt{1+\varepsilon}$ *with probability at least* $1 - 2\delta$*.*

Remark 2.7. The universal constant c in Theorem 2.5 and Corollary 2.6 is same as that in the Hanson–Wright inequality (see [33, Theorem 6.2.1]). It can be shown that this constant is greater than 1/64 [25].

Proposition 2.8. *An* $(\varepsilon, \binom{n}{d}^{-1}\delta, d)$ *oblivious* ℓ_2*-subspace embedding* $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$ *satisfies* (ε, d)*-RIP with probability at least* $1 - \delta$*.*

Proof. Let $\mathcal{B}$ denote the canonical (Euclidean) basis for $\mathbb{R}^n$. It follows directly from the definition of $\mathbf{\Theta}$ and the union bound argument that $\mathbf{\Theta}$ is an ε-embedding for all subspaces spanned by d vectors from $\mathcal{B}$, simultaneously, with probability at least $1-\delta$. Since every d-sparse vector $\mathbf{a} \in \mathbb{R}^n$ belongs to a subspace spanned by d vectors from $\mathcal{B}$, we conclude that $\mathbf{\Theta}$ satisfies (2.8) with probability at least $1 - \delta$. □

Theorem 2.9. *Let* $\mathbf{\Theta} \in \mathbb{R}^{k\times n}$ *be a matrix satisfying* $(\varepsilon/4, 2d)$*-RIP with* $d = 2.1c^{-1}\log(4/\delta)$*, where* $c \leq 1$ *is some universal constant. Let* $\varphi \in \mathbb{R}^n$ *be a vector with entries that are independent random variables. If* φ *has zero mean and* $|\varphi| \leq \gamma$ *for some vector* $\gamma \in \mathbb{R}^n$*, then*

$$||\varphi\|^2 - \|\mathbf{\Theta}\varphi\|^2| \leq \varepsilon\|\gamma\|^2$$

holds with probability at least $1 - \delta$*.*

Proof. See supplementary material (M138870SupMat.pdf [local/web 390KB]). □

2.3. RGS process. Consider the variants of Algorithm 1.1 where the projector $\mathbf{\Pi}^{(i-1)}$ has the form

$$\mathbf{\Pi}^{(i-1)}\mathbf{w}_i = \mathbf{w}_i - \mathbf{Q}_{i-1}\mathbf{x} \tag{2.9}$$

[5] A vector $\mathbf{a} \in \mathbb{R}^n$ is called d-sparse if it has at most d nonzero entries.

with $\mathbf{x} = \mathbf{R}_{(1:i-1,i)}$ computed from $\mathbf{Q}_{i-1}$ and $\mathbf{w}_i$. The classical methods proceed with taking $\mathbf{x}$ as an approximation of $\mathbf{Q}_{i-1}^{\dagger}\mathbf{w}_i$ or, equivalently, as an approximate solution to the following least-squares problem:

$$\min_{\mathbf{y}} \|\mathbf{Q}_{i-1}\mathbf{y} - \mathbf{w}_i\|. \tag{2.10}$$

The stability of Algorithm 1.1 in this case can be directly linked to the accuracy of $\mathbf{x}$. The CGS and MGS algorithms belong to the aforementioned category of GS processes with $\mathbf{x}$ taken as, respectively, $\mathbf{x} = \mathbf{Q}_{i-1}^{\mathrm{T}}\mathbf{w}_i$ and $\mathbf{x} = \mathbf{T}_{i-1}(\mathbf{Q}_{i-1}^{\mathrm{T}}\mathbf{w}_i)$, for some triangular matrix $\mathbf{T}_{i-1}$ [31]. The connection of CGS and MGS with solving (2.10) was explored in [27]. A similar formulation was also used in [6].

In this work we develop new variants of the GS process that satisfy (2.9), but this time that produce the output Q factor orthonormal with respect to the sketched product $\langle \mathbf{\Theta}\cdot, \mathbf{\Theta}\cdot\rangle$ rather than the ℓ_2-inner product as in standard methods. Thus, the Q factor is no longer ℓ_2-orthonormal even in exact arithmetic, though, according to Corollary 2.2, it has a small (ℓ_2-)condition number and yields a reduced computational cost. Such factorization corresponds to taking $\mathbf{x}$ in (2.9) as an approximation of $(\mathbf{\Theta}\mathbf{Q}_{i-1})^{\dagger}(\mathbf{\Theta}\mathbf{w}_i)$ or, equivalently, a minimizer of the sketched residual:

$$\min_{\mathbf{y}} \|(\mathbf{\Theta}\mathbf{Q}_{i-1})\mathbf{y} - \mathbf{\Theta}\mathbf{w}_i\|.$$

Furthermore, the normalization of $\mathbf{q}_i$ at step 2 of Algorithm 1.1 has to be performed accordingly: $\mathbf{q}_i = \mathbf{q}_i/\|\mathbf{\Theta}\mathbf{q}_i\|$. We see that unlike in standard methods, here the computation of $\mathbf{x}$ requires only (efficient) evaluation of random projections and operations on small vectors and matrices with no standard operations on high-dimensional vectors. The GS process with such a projector is depicted in Algorithm 2.1.

In general, stability of Algorithm 2.1 directly depends on the accuracy and stability of the least-squares solver used in step 2. One should prioritize least-squares solvers that are as accurate and stable as possible. They can be based on Givens rotations or Householder transformation as is considered in our stability analysis (see subsection 3.1). Such standard solvers should yield a negligible computational cost when matrix $\mathbf{S}_m = \mathbf{\Theta}\mathbf{Q}_m$ is sufficiently small, which happens in most applications. However, when $\mathbf{S}_m$ is of moderate size, the least-squares solution with standard methods can entail a considerable computational cost and has to be avoided. In such cases, by using the fact that $\mathbf{S}_{i-1}$ is approximately orthonormal, one can compute $\mathbf{x} = [\mathbf{R}]_{(1:i-1,i)}$ from the normal equation:

$$(\mathbf{S}_{i-1})^{\mathrm{T}}\mathbf{S}_{i-1}\mathbf{x} = (\mathbf{S}_{i-1})^{\mathrm{T}}\mathbf{p}_i$$

with several Richardson iterations $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{S}_{i-1}^{\mathrm{T}}(\mathbf{p}_i - \mathbf{S}_{i-1}\mathbf{x})$ requiring a minor computational cost. The resulting algorithm can be viewed as a sketched version of the CGS process with re-orthogonalizations. The case $\mathbf{x} = (\mathbf{S}_{i-1})^{\mathrm{T}}\mathbf{p}_i$ with only one Richardson iteration exactly corresponds to the orthogonalization of $\mathbf{W}$ with respect to $\langle \mathbf{\Theta}\cdot, \mathbf{\Theta}\cdot\rangle$ with the CGS process defined for a general inner product. Moreover, instead of using the Richardson iterations we could also compute $\mathbf{x}$ by orthogonalizing $\mathbf{p}_i$ to $\mathbf{S}_{i-1}$ with an MGS step. In this case Algorithm 2.1 would correspond to orthogonalization of $\mathbf{W}$ with respect to $\langle \mathbf{\Theta}\cdot, \mathbf{\Theta}\cdot\rangle$ with the MGS process. The ways for efficient and stable solution of the sketched least-squares problem are addressed in details in our subsequent work on the block variants of the RGS process.

At $i = 1$ of Algorithm 2.1 we used a conventional notation that $[\mathbf{R}]_{(1:i-1,i)}$ is a 0-by-1 matrix and $\mathbf{Q}_{i-1}$ is an n-by-0 matrix, implying that $\mathbf{q}_i' = \mathbf{w}_i$ and $\mathbf{s}_i' = \mathbf{p}_i$.

Algorithm 2.1. RGS algorithm

```
Given: $n \times m$ matrix $\mathbf{W}$ and $k \times n$ matrix $\mathbf{\Theta}$, $m \leq k \ll n$.
Output: $n \times m$ factor $\mathbf{Q}$ and $m \times m$ upper triangular factor $\mathbf{R}$.
for $i = 1 : m$ do
    1. Sketch $\mathbf{w}_i$: $\mathbf{p}_i = \mathbf{\Theta}\mathbf{w}_i$.                                # macheps: $u_{fine}$
    2. Solve $k \times (i-1)$ least-squares problem:
          $[\mathbf{R}]_{(1:i-1,i)} = \arg\min_{\mathbf{y}} \|\mathbf{S}_{i-1}\mathbf{y} - \mathbf{p}_i\|$.   # macheps: $u_{fine}$
    3. Compute projection of $\mathbf{w}_i$: $\mathbf{q}'_i = \mathbf{w}_i - \mathbf{Q}_{i-1}[\mathbf{R}]_{(1:i-1,i)}$.   # macheps: $u_{crs}$
    4. Sketch $\mathbf{q}'_i$: $\mathbf{s}'_i = \mathbf{\Theta}\mathbf{q}'_i$.                               # macheps: $u_{fine}$
    5. Compute the sketched norm $r_{i,i} = \|\mathbf{s}'_i\|$.                    # macheps: $u_{fine}$
    6. Scale vector $\mathbf{s}_i = \mathbf{s}'_i / r_{i,i}$.                           # macheps: $u_{fine}$
    7. Scale vector $\mathbf{q}_i = \mathbf{q}'_i / r_{i,i}$.                           # macheps: $u_{fine}$
end for
8. (Optional) compute $\Delta_m = \|\mathbf{I}_{m\times m} - \mathbf{S}_m^{\mathrm{T}}\mathbf{S}_m\|_{\mathrm{F}}$ and $\tilde{\Delta}_m = \frac{\|\mathbf{P}_m - \mathbf{S}_m\mathbf{R}_m\|_{\mathrm{F}}}{\|\mathbf{P}_m\|_{\mathrm{F}}}$.
   Use Theorem 3.2 to certify the output.                          # macheps: $u_{fine}$
```

Algorithm 2.1 is executed with a multiprecision finite arithmetic with two unit roundoffs: a coarse one u_{crs}, and a fine one u_{fine}, $u_{fine} \leq u_{crs} \leq 0.01/m$. The roundoff u_{crs} represents the *working precision* and is used for standard operations on high-dimensional vectors and matrices in step 3, which is the most expensive computation in the algorithm. This precision is also used for storage of large matrices $\mathbf{Q}$ and $\mathbf{W}$. All other (inexpensive) operations in Algorithm 2.1 are performed and accumulated with a fine roundoff u_{fine}. We chose a multiprecision model rather than a unique precision one to show an interesting property of the algorithm: that one may guarantee stability of Algorithm 2.1 by performing standard operations on high-dimensional vectors (i.e., step 3) with unit roundoff u_{crs} independent of n (and k). This feature of the RGS process can have a particular importance for extreme-scale problems. Clearly, the results from this paper can be also used for the analysis of Algorithm 2.1 executed with unique unit roundoff u_{fine}. The stability guarantees in such a case can be derived from sections 3 and 4 simply by introducing a fictitious unit roundoff $u_{crs} = F(m, n)u_{fine}$, where $F(m, n)$ is a low-degree polynomial, and looking at Algorithm 2.1 as though it were executed with multiprecision arithmetic with unit roundoffs u_{fine} and u_{crs}.

Remark 2.10. In step 4 of Algorithm 2.1, the sketch of $\mathbf{q}'_i$ could be computed also as $\mathbf{s}'_i = \mathbf{p}_i - \mathbf{S}_{i-1}[\mathbf{R}]_{(1:i-1,i)}$ instead of $\mathbf{s}'_i = \mathbf{\Theta}\mathbf{q}'_i$. Our experiments, however, revealed that this way is less stable.

2.4. Performance analysis.

Let us now characterize the efficiency of Algorithm 2.1 executed in different computational architectures. The performance analysis is done through comparison to CGS. The CGS algorithm is the most efficient from the standard (column-oriented) algorithms for orthogonalization of a set of vectors. It requires nearly half as many flops and synchronizations between processors as CGS2, and, unlike MGS, it can be implemented using matrix-vector operations, i.e., level-2 basic linear algebra subprograms (BLAS-2).

By assuming that $m \leq k \ll n$, we shall neglect the cost of operations on sketched vectors and matrices in Algorithm 2.1. Then the computational cost is characterized by evaluation of random sketches at steps 1 and 4 and computation of the projection of $\mathbf{w}_i$ at step 3. Moreover, at steps 1 and 4 we let the sketching matrix $\mathbf{\Theta}$ be chosen depending on each particular situation to yield the most efficiency.

The RGS algorithm can be beneficial in terms of the classical metric of efficiency, which is the number of flops. If $\mathbf{\Theta}$ is taken as SRHT, then the random projections at steps 1 and 4 require in total (no more than) $4n\log(k+1)$ flops at each iteration. For sufficiently large i, this cost is much less than the cost of step 3 that is nearly $2ni$ flops. The CGS requires more than $4ni$ flops at each iteration, and therefore it is nearly twice as expensive as RGS. Furthermore, the flops at step 3 of the RGS algorithm can be done in low precision, which can make Algorithm 2.1 even more efficient.

Both CGS and RGS algorithms can be implemented by using BLAS-2 routines for high-dimensional operations. The CGS algorithm in such an implementation, however, entails (at least) two passes over the basis matrix $\mathbf{Q}_{i-1}$ at iteration i, $2 \leq i \leq m$. Algorithm 2.1, on the other hand, at each iteration requires only one pass over $\mathbf{Q}_{i-1}$ and two applications of $\mathbf{\Theta}$. The applications of $\mathbf{\Theta}$ can be performed by utilizing a seeded random number generator with negligible storage costs. Consequently, in this case RGS can be more pass-efficient than CGS. Furthermore, the matrix $\mathbf{Q}_{i-1}$ in the RGS algorithm can be maintained in lower precision and still yield similar (or better) accuracy than the CGS algorithm, which can amplify the storage reduction.

To characterize the performance of RGS in parallel/distributed computational architecture, we consider the situation when the columns of $\mathbf{W}$ are provided recursively as

$$\mathbf{w}_{i+1} = \mathbf{A}\mathbf{q}_i, \ 1 \leq i \leq m-1,$$

where $\mathbf{A}$ is an $n \times n$ matrix. This, for instance, happens in the Arnoldi algorithm for computing an orthonormal basis of a Krylov subspace, which is the core ingredient of the GMRES algorithm (see section 4 for details). We here assume that the high-dimensional matrix $\mathbf{A}$ and the vectors $\mathbf{q}_i$ are distributed among processors using block rowwise partitioning (possibly with overlaps). This is a standard situation when $\mathbf{A}$ is obtained from discretization of a PDE. It is then assumed that the computation of the matrix-vector product $\mathbf{w}_{i+1} = \mathbf{A}\mathbf{q}_i$ requires communication only between neighboring processors and has a minor impact on the overall communication cost. We also assume that along with the local matrices and vectors on each processor are also maintained copies of the sketches $\mathbf{S}_i$, $\mathbf{s}'_i$, and $\mathbf{p}_i$ and matrix $\mathbf{R}_i$.

Next we notice that the utilization of a seeded random number generator can allow efficient access to any block of $\mathbf{\Theta}$ with a minor computational cost and, in particular, with absolutely no communication. The computation of the sketch $\mathbf{p}_i = \mathbf{\Theta}\mathbf{A}\mathbf{q}_{i-1}$ in step 1 then requires only one global synchronization. The computation of the sketch $\mathbf{\Theta}\mathbf{q}'_i$ in step 4 of Algorithm 2.1 requires an additional synchronization, which implies in total two global synchronizations at each iteration of Algorithm 2.1. This communication cost is the same as of the classical implementation of CGS. In [21, section 4] is described a way to reduce the communication cost of the CGS algorithm to only one synchronization per iteration. This technique may also be applied to the RGS algorithm by incorporating a lag into steps 5–7 of Algorithm 2.1. More specifically, at iteration i, we can compute two sketches

$$\mathbf{s}'_i = \mathbf{\Theta}\mathbf{q}'_i \text{ and } \mathbf{p}'_{i+1} = \mathbf{\Theta}\mathbf{A}\mathbf{q}'_i, \ 1 \leq i \leq m-1,$$

simultaneously, by utilizing only one global synchronization, and then perform the normalizations: $r_{i,i} = \|\mathbf{s}'_i\|$, $\mathbf{s}_i = \mathbf{s}'_i/r_{i,i}$, $\mathbf{q}_i = \mathbf{q}'_i/r_{i,i}$, and $\mathbf{p}_{i+1} = \mathbf{p}'_{i+1}/r_{i,i}$. The communication cost of such implementation of the RGS algorithm then becomes only one global synchronization per iteration. We conclude that RGS and CGS should have similar numbers of required synchronizations in parallel/distributed computational architecture.

3. Stability of RGS process.

In this section we provide stability analysis of Algorithm 2.1. It is based on the following assumptions that hold with high probability if $\mathbf{\Theta}$ is an oblivious subspace embedding of sufficiently large size.

First, we assume that $\mathbf{\Theta}$ satisfies (3.1). According to Corollary 2.4, this property holds with probability at least $1-\delta$, if $\mathbf{\Theta}$ is $(1/2, \delta/n, 1)$ oblivious subspace embedding.

Furthermore, let us define rounding error vectors

$$\mathbf{\Delta q}'_i := \widehat{\mathbf{q}}'_i - \left(\widehat{\mathbf{w}}_i - \widehat{\mathbf{Q}}_{i-1}[\widehat{\mathbf{R}}]_{(1:i-1,i)}\right) \text{ and } \mathbf{\Delta q}_i := \widehat{\mathbf{q}}_i - \widehat{\mathbf{q}}'_i/\hat{r}_{i,i}$$

in steps 3 and 6 of Algorithm 2.1. Then, the standard worst-case scenario rounding analysis yields (3.2) (for derivation, see subsection 2.2). Following the arguments from subsection 2.2, we assume that $\mathbf{\Theta}$ satisfies (3.3). It follows from Corollary 2.6 and the union bound argument that this property holds under the probabilistic rounding model with probability at least $1-4\delta$ if $\mathbf{\Theta}$ is $(1/8, m^{-1}\binom{n}{d}^{-1}\delta, d)$, with $d = \mathcal{O}(\log(m/\delta))$, oblivious subspace embedding. By using the bounds (2.2) we conclude that Assumptions 3.1 hold under the probabilistic rounding model with probability at least $1-\delta$, if $\mathbf{\Theta}$ is a Rademacher matrix with $k = \mathcal{O}(\log(n)\log(m/\delta))$ rows or SRHT with $k = \mathcal{O}(\log^2(n)\log^2(m/\delta))$ rows. Note that these properties should also hold under many other (possibly deterministic) rounding models. The classical worst-case scenario model, however, entails $D = \sqrt{1+\varepsilon}\sqrt{n}$ in (3.3). The numerical stability bounds in this case can be deduced from Theorems 3.2 and 3.3 by letting $u_{crs} = u_{crs}\sqrt{n}$ and $u_{fine} = u_{fine}\sqrt{n}$.

Assumptions 3.1. It is assumed that

$$\|\mathbf{\Theta}\|_{\mathrm{F}} \leq \sqrt{1+\varepsilon}\sqrt{n} \tag{3.1}$$

with $\varepsilon \leq 1/2$. Furthermore, in Algorithm 2.1 we assume that

$$|\mathbf{\Delta q}'_i| \leq 1.02 u_{crs}(|\widehat{\mathbf{w}}_i| + i|\widehat{\mathbf{Q}}_{i-1}||[\widehat{\mathbf{R}}]_{(1:i-1,i)}|), \tag{3.2a}$$

$$|\mathbf{\Delta q}_i| \leq u_{fine}|\widehat{\mathbf{q}}'_i/\hat{r}_{i,i}|, \tag{3.2b}$$

and

$$\|\mathbf{\Theta}\mathbf{\Delta q}'_i\| \leq 1.02 u_{crs} D \||\widehat{\mathbf{w}}_i| + i|\widehat{\mathbf{Q}}_{i-1}||[\widehat{\mathbf{R}}]_{(1:i-1,i)}|\|, \tag{3.3a}$$

$$\|\mathbf{\Theta}\mathbf{\Delta q}_i\| \leq u_{fine} D\|\widehat{\mathbf{q}}'_i/\hat{r}_{i,i}\|, \tag{3.3b}$$

with $D = \sqrt{1+\varepsilon}$, $\varepsilon \leq 1/2$, $1 \leq i \leq m$.

3.1. Stability analysis.

The results in this subsection shall rely on the condition that $\mathbf{\Theta}$ satisfies the ε-embedding property for $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{W}}$. A priori analysis to satisfy this property with (high) user-specified probability of success is provided in subsection 3.2. Furthermore, in subsection 3.2 we also provide a way to efficiently certify that $\mathbf{\Theta}$ is an ε-embedding for $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{W}}$. Then the stability of Algorithm 2.1 can be characterized by coefficients

$$\Delta_m = \|\mathbf{I}_{m\times m} - \widehat{\mathbf{S}}_m^{\mathrm{T}}\widehat{\mathbf{S}}_m\|_{\mathrm{F}} \text{ and } \tilde{\Delta}_m = \|\widehat{\mathbf{P}}_m - \widehat{\mathbf{S}}_m\widehat{\mathbf{R}}_m\|_{\mathrm{F}}/\|\widehat{\mathbf{P}}_m\|_{\mathrm{F}},$$

as is shown in Theorem 3.2.

THEOREM 3.2. *Assume that*

$$100m^{1/2}n^{3/2}u_{fine} \le u_{crs} \le 0.01m^{-1} \text{ and } n \ge 100,$$

along with Assumptions 3.1. *If* $\mathbf{\Theta}$ *is an* ε*-embedding for* $\widehat{\mathbf{Q}}$ *and* $\widehat{\mathbf{W}}$ *from Algorithm* 2.1*, with* $\varepsilon \le 1/2$*, and if* $\Delta_m, \tilde{\Delta}_m \le 0.1$*, then the following inequalities hold:*

$$(1+\varepsilon)^{-1/2}(1-\Delta_m-0.1u_{crs}) \le \sigma_{min}(\widehat{\mathbf{Q}}) \le \sigma_{max}(\widehat{\mathbf{Q}}) \le (1-\varepsilon)^{-1/2}(1+\Delta_m+0.1u_{crs})$$

and

$$\|\widehat{\mathbf{W}} - \widehat{\mathbf{Q}}\widehat{\mathbf{R}}\|_{\mathrm{F}} \le 3.7u_{crs}m^{3/2}\|\widehat{\mathbf{W}}\|_{\mathrm{F}}.$$

Proof. See supplementary material (M138870SupMat.pdf [local/web 390KB]). □

According to Theorem 3.2, the numerical stability of Algorithm 2.1 can be ensured by guaranteeing that Δ_m and $\tilde{\Delta}_m$ are sufficiently small (along with the ε-embedding property of $\mathbf{\Theta}$). This can be done by employing a sufficiently accurate backward-stable solver to the least-squares problem in step 2. Below we provide theoretical bounds for Δ_m and $\tilde{\Delta}_m$ in this case.

THEOREM 3.3. *Consider Algorithm* 2.1 *utilizing QR factorization based on Householder transformation or Givens rotations for computing the solution to the least-squares problem in step* 2.

Under Assumptions 3.1*, if* $\mathbf{\Theta}$ *is an* ε*-embedding for* $\widehat{\mathbf{Q}}_{m-1}$ *and* $\widehat{\mathbf{W}}$*, with* $\varepsilon \le 1/2$*, and if*

$$u_{crs} \le 10^{-3}\mathrm{cond}(\widehat{\mathbf{W}})^{-1}m^{-2},$$
$$u_{fine} \le (100m^{1/2}n^{3/2} + 10^4m^{3/2}k)^{-1}u_{crs},$$

then Δ_m *and* $\tilde{\Delta}_m$ *are bounded by*

$$\tilde{\Delta}_m \le 4.2u_{crs}m^{3/2}\|\widehat{\mathbf{W}}\|_{\mathrm{F}}/\|\widehat{\mathbf{P}}\|_{\mathrm{F}} \le 6u_{crs}m^{3/2}, \tag{3.4}$$

$$\Delta_m \le 20u_{crs}m^2\mathrm{cond}(\widehat{\mathbf{W}}). \tag{3.5}$$

Proof. See supplementary material (M138870SupMat.pdf [local/web 390KB]). □

Remark 3.4. In general, the result of Theorem 3.3 holds for any least-squares solver in step 2 as long as the following backward-stability property is satisfied:

$$[\widehat{\mathbf{R}}]_{(1:i-1,i)} = \arg\min_{\mathbf{y}} \|(\widehat{\mathbf{S}}_{i-1} + \mathbf{\Delta S}_{i-1})\mathbf{y} - (\widehat{\mathbf{p}}_i + \mathbf{\Delta p}_i)\| \text{ with}$$
$$\|\mathbf{\Delta S}_{i-1}\|_{\mathrm{F}} \le 0.01u_{crs}\|\widehat{\mathbf{S}}_{i-1}\|, \quad \|\mathbf{\Delta p}_i\| \le 0.01u_{crs}\|\widehat{\mathbf{p}}_i\|.$$

Remark 3.5. Notice that Theorem 3.3 requires $\mathbf{\Theta}$ to be an ε-embedding for $\widehat{\mathbf{Q}}_{m-1}$ and not $\widehat{\mathbf{Q}}$. This observation will become handy for proving the ε-embedding property for $\widehat{\mathbf{Q}}$ in subsection 3.2 by using induction on m.

Theorems 3.2 and 3.3 imply a stable QR factorization for working unit roundoff u_{crs} independent of the high dimension n.

In some cases, obtaining a priori guarantees with Theorem 3.3 can be an impractical task due to the need to estimate $\mathrm{cond}(\widehat{\mathbf{W}})$. Furthermore, one may want to use Algorithm 2.1 with a higher value of u_{crs} than is assumed in Theorem 3.3, possibly with a bigger gap between the values of u_{crs} and u_{fine}. In such cases, the computed QR factorization can be efficiently certified a posteriori by computing Δ_m and $\tilde{\Delta}_m$ and using Theorem 3.2 with no operations on high-dimensional vectors and matrices, and the estimation of $\mathrm{cond}(\widehat{\mathbf{W}})$.

3.2. Epsilon embedding property. The stability analysis in subsection 3.1 holds if $\mathbf{\Theta}$ satisfies the ε-embedding property for $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{W}}$. In this section we provide a priori and a posteriori analysis of this property.

A priori analysis. Let us consider the case when $\widehat{\mathbf{W}}$ and $\mathbf{\Theta}$ are independent of each other. Then it follows directly from Definition 2.3 that if $\mathbf{\Theta}$ is (ε, δ, m) oblivious ℓ_2-subspace embedding, then it satisfies the ε-embedding property for $\widehat{\mathbf{W}}$ with high probability. Below, we provide a guarantee that in this case $\mathbf{\Theta}$ will also satisfy an ε-embedding property for $\widehat{\mathbf{Q}}$ with moderately increased value of ε.

PROPOSITION 3.6. *Consider Algorithm* 2.1 *using the Givens or Householder least-squares solver in step* 2 *and computed with unit roundoffs*

$$u_{crs} \leq 10^{-3}\mathrm{cond}(\widehat{\mathbf{W}})^{-1}m^{-2},$$
$$u_{fine} \leq (100m^{1/2}n^{3/2} + 10^4m^{3/2}k)^{-1}u_{crs}.$$

Under Assumptions 3.1, *if* $\mathbf{\Theta}$ *is an* ε*-embedding for* $\widehat{\mathbf{W}}$, *with* $\varepsilon \leq 1/4$, *then it satisfies the* ε'*-embedding property for* $\widehat{\mathbf{Q}}$ *with*

$$\varepsilon' = 2\varepsilon + 180u_{crs}m^2\mathrm{cond}(\widehat{\mathbf{W}}).$$

Proof. See supplementary material (M138870SupMat.pdf [local/web 390KB]). □

When $\widehat{\mathbf{W}}$ is generated depending on $\mathbf{\Theta}$ (as we have in the Arnoldi process in section 4), the a priori analysis for the ε-embedding property for $\widehat{\mathbf{W}}$ can be nontrivial and pessimistic. Nevertheless, $\mathbf{\Theta}$ can still be expected to be an ε-embedding because there is only a minor correlation of the rounding errors with $\mathbf{\Theta}$. When there is no a priori guarantee on the quality of $\mathbf{\Theta}$ or the guarantee is pessimistic, it can be important to be able to certify the ε-embedding property a posteriori, which is discussed next.

A posteriori certification. The quality of $\mathbf{\Theta}$ can be certified by providing an upper bound $\bar{\omega}$ for the minimum value ω of ε, for which $\mathbf{\Theta}$ satisfies the ε-embedding property for $\mathbf{V}$. The matrix $\mathbf{V}$ can be chosen as $\widehat{\mathbf{Q}}$ or $\widehat{\mathbf{W}}$. The considered a posteriori bound $\bar{\omega}$ is probabilistic. We proceed by introducing an (additional to $\mathbf{\Theta}$) sketching matrix $\mathbf{\Phi}$ used solely for the certification so that it is randomly independent of $\mathbf{V}$ and $\mathbf{\Theta}$. For efficiency, this matrix should be of size no more than the size of $\mathbf{\Theta}$. In practice, an easy and robust way is to use $\mathbf{\Phi}$ and $\mathbf{\Theta}$ of same size. Define parameters ε^* and δ^* characterizing, respectively, the accuracy of $\bar{\omega}$ (i.e., its closeness to ω) and the probability of failure for $\bar{\omega}$ to be an upper bound. Then we can use the following results from [5] (see Propositions 3.7 and 3.8).

PROPOSITION 3.7 (corollary of Proposition 5.3 in [5]). *Assume that* $\mathbf{\Phi}$ *is a* $(\varepsilon^*, \delta^*, 1)$*-oblivious subspace embedding. Let* $\mathbf{V} = \widehat{\mathbf{Q}}$ *or* $\widehat{\mathbf{W}}$, $\mathbf{V}^{\mathbf{\Theta}} = \mathbf{\Theta}\mathbf{V}$, *and* $\mathbf{V}^{\mathbf{\Phi}} = \mathbf{\Phi}\mathbf{V}$. *Let* $\mathbf{X}$ *be a matrix such that* $\mathbf{V}^{\mathbf{\Phi}}\mathbf{X}$ *is orthonormal. If*

$$\bar{\omega} = \max\{1 - (1-\varepsilon^*)\sigma_{min}^2(\mathbf{V}^{\mathbf{\Theta}}\mathbf{X}), (1+\varepsilon^*)\sigma_{max}^2(\mathbf{V}^{\mathbf{\Theta}}\mathbf{X}) - 1\} < 1,$$

then $\mathbf{\Theta}$ *is a* $\bar{\omega}$*-embedding for* $\mathbf{V}$, *with probability at least* $1 - \delta^*$.

PROPOSITION 3.8 (corollary of Proposition 5.4 in [5]). *In Proposition* 3.7, *if* $\mathbf{\Phi}$ *is a* ε'*-embedding for* $\mathbf{V}$, *then* $\bar{\omega}$ *satisfies*

$$\bar{\omega} \leq (1+\varepsilon^*)(1-\varepsilon')^{-1}(1+\omega) - 1.$$

First we notice that the coefficient $\bar{\omega}$ in Proposition 3.7 can be efficiently computed from the two sketches of $\mathbf{V}$ with no operations on high-dimensional vectors and matrices. For efficiency, (at iteration i) the sketches $\mathbf{\Phi}\widehat{\mathbf{q}}'_i$ and $\mathbf{\Phi}\widehat{\mathbf{w}}_i$ may be computed along with, respectively, $\widehat{\mathbf{s}}'_i$ in step 4 and $\widehat{\mathbf{p}}_i$ in step 1 of Algorithm 2.1. The matrix $\mathbf{X}$ can be obtained (possibly in implicit form, e.g., as inverse of upper-triangular matrix) with standard orthogonal decomposition algorithms such as the QR factorization or the singular value decomposition performed in sufficient precision. It follows that $\bar{\omega}$ is an upper bound for ω with probability at least $1-\delta^*$ if $\mathbf{\Phi}$ is a $(\varepsilon^*, \delta^*, 1)$-oblivious subspace embedding. This property of $\mathbf{\Phi}$ has to be guaranteed a priori, e.g., from the theoretical bounds (2.2) or [2, Lemma 5.1]. For instance, it is guaranteed to hold for Rademacher matrices with $k \geq 2({\varepsilon^*}^2/2 - {\varepsilon^*}^3/3)^{-1}\log(\delta^*/2)$ rows, which in particular becomes $k > 530$ for $\varepsilon^* = 1/4$ and $\delta^* = 0.1\%$ and any m and n.

In Proposition 3.8, the closeness of $\bar{\omega}$ and ω is guaranteed if $\mathbf{\Phi}$ is a ε'-embedding for $\mathbf{V}$ (for some given ε'). This condition shall be satisfied with probability at least $1-\delta'$ (for some given δ') if $\mathbf{\Phi}$ is an $(\varepsilon', \delta', m)$ oblivious ℓ_2-subspace embedding. This fact is not required to be guaranteed a priori, which allows us to choose the size for $\mathbf{\Phi}$ (and $\mathbf{\Theta}$) based on practical experience and still have a certification.

In practice, the random projections $\mathbf{V}^{\mathbf{\Theta}} = \mathbf{\Theta}\mathbf{V}$ and $\mathbf{V}^{\mathbf{\Phi}} = \mathbf{\Phi}\mathbf{V}$ can be computed only approximately due to rounding errors. In such a case, it can be important to provide a stability guarantee for the computed value of $\bar{\omega}$ (given in Proposition 3.9). In Proposition 3.9, along with Assumptions 3.1 for $\mathbf{\Theta}$, we also assume that

$$\|\mathbf{\Phi}\|_{\mathrm{F}} \leq \sqrt{1+\varepsilon}\sqrt{n}, \quad \text{and } \|\mathbf{V}^{\mathbf{\Phi}}\|_{\mathrm{F}} \geq \sqrt{1-\varepsilon}\|\mathbf{V}\|_{\mathrm{F}} \tag{3.6}$$

with $\varepsilon \leq 1/2$. These properties hold with probability at least $1-2\delta^*$ if $\mathbf{\Phi}$ is $(1/2, \delta^*/n, 1)$ oblivious subspace embedding.

PROPOSITION 3.9. *Let* $\mathbf{V} = \widehat{\mathbf{Q}}$ *or* $\widehat{\mathbf{W}}$, $\widehat{\mathbf{V}}^{\mathbf{\Theta}} = \mathrm{fl}(\mathbf{\Theta}\cdot\mathbf{V})$, *and* $\widehat{\mathbf{V}}^{\mathbf{\Phi}} = \mathrm{fl}(\mathbf{\Phi}\cdot\mathbf{V})$. *Assume that the sketches are computed with unit roundoff* u_{fine} *satisfying*

$$100n^{3/2}m^{1/2}u_{fine} \leq u_{crs} \leq \mathrm{cond}(\widehat{\mathbf{V}}^{\mathbf{\Phi}})^{-1}.$$

Assume that $\mathbf{\Phi}$ *satisfies* (3.6) *and that* $\mathbf{\Theta}$ *satisfies Assumptions* 3.1. *Let* $\widehat{\mathbf{X}}$ *be a matrix such that* $\widehat{\mathbf{V}}^{\mathbf{\Phi}}\widehat{\mathbf{X}}$ *is orthonormal. Define*

$$\hat{\bar{\omega}} = \max\{1-(1-\varepsilon^*)\sigma^2_{min}(\widehat{\mathbf{V}}^{\mathbf{\Theta}}\widehat{\mathbf{X}}), (1+\varepsilon^*)\sigma^2_{max}(\widehat{\mathbf{V}}^{\mathbf{\Theta}}\widehat{\mathbf{X}})-1\}.$$

Then we have that if $\bar{\omega} \leq 1$,

$$|\bar{\omega} - \hat{\bar{\omega}}| \leq u_{crs}\mathrm{cond}(\widehat{\mathbf{V}}^{\mathbf{\Phi}}).$$

Proof. See supplementary material (M138870SupMat.pdf [local/web 390KB]). □

It follows from Proposition 3.9 that the computed value of $\bar{\omega}$ is approximately equal to the exact one if $u_{crs}\mathrm{cond}(\widehat{\mathbf{V}}^{\mathbf{\Phi}})$ is sufficiently small. This implies the following computable certificate for the quality of $\mathbf{\Theta}$:

$$\omega \leq \bar{\omega} \leq \hat{\bar{\omega}} + u_{crs}\mathrm{cond}(\widehat{\mathbf{V}}^{\mathbf{\Phi}}),$$

which holds with probability at least $1-\mathcal{O}(\delta^*)$. Furthermore, Proposition 3.9 is consistent: $\mathrm{cond}(\widehat{\mathbf{V}}^{\mathbf{\Phi}})$ is guaranteed to be sufficiently small, namely, $\mathcal{O}(\mathrm{cond}(\mathbf{V}))$, if $\mathbf{\Phi}$ is an ε-embedding for $\mathbf{V}$.

Remark 3.10 (RGS algorithm with multiple sketches). The certification of $\mathbf{\Theta}$ can be performed at each iteration of Algorithm 2.1 (by letting $m = i$ in the above procedure). In this way, one can detect the iteration i (if there is any) without sufficient quality of $\mathbf{\Theta}$ and switch to a new sketching matrix, which is randomly independent from $\mathbf{V}$. This allows us to make sure that the used $\mathbf{\Theta}$ is an ε-embedding for $\mathbf{V}$ at each iteration. We leave the development of the RGS algorithm with multiple sketches for future research.

4. Application to Arnoldi process and GMRES.

In this section we employ the RGS process to solving high-dimensional nonsingular (possibly non-symmetric) systems of equations of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4.1}$$

with the GMRES method. Without loss of generality we here assume that (4.1) is normalized so that $\|\mathbf{b}\| = \|\mathbf{A}\| = 1$.

An order-j Krylov subspace is defined as

$$\mathcal{K}_j(\mathbf{A}, \mathbf{b}) := \operatorname{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{j-1}\mathbf{b}\}.$$

The GMRES method consists in approximation of $\mathbf{x}$ with a projection $\mathbf{x}_{m-1}$ in $\mathcal{K}_{m-1}(\mathbf{A}, \mathbf{b})$ that minimizes the residual norm

$$\|\mathbf{A}\mathbf{x}_{m-1} - \mathbf{b}\|.$$

To obtain a projection $\mathbf{x}_{m-1}$, the GMRES method first proceeds with constructing the orthonormal basis of $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ with an Arnoldi process (usually based on GS orthogonalization). Then the coordinates of $\mathbf{x}_{m-1}$ in the Arnoldi basis are found by solving a (small) transformed least-squares problem.

4.1. RGS-Arnoldi process.

The Arnoldi basis can be constructed recursively by taking the first basis vector $\mathbf{q}_1$ as a normalized right-hand-side vector $\mathbf{b}$ and each new vector $\mathbf{q}_{i+1}$ as $\mathbf{A}\mathbf{q}_i$ orthonormalized against the previously computed basis $\mathbf{q}_1, \ldots, \mathbf{q}_i$. This procedure then produces the orthonormal matrix $\mathbf{Q}_m$ satisfying the Arnoldi identity

$$\mathbf{A}\mathbf{Q}_{m-1} = \mathbf{Q}_m \mathbf{H}_m,$$

where $\mathbf{H}_m$ is the upper Hessenberg matrix. The Arnoldi algorithm can be viewed as a column-oriented QR factorization of matrix $[\mathbf{b}, \mathbf{A}\mathbf{Q}_{m-1}]$. In this case, the R factor $\mathbf{R}_m$ and the Hessenberg matrix $\mathbf{H}_m$ satisfy the relation $\mathbf{H}_m = [\mathbf{R}_m]_{(1:m,2:m)}$.

Below, we propose a randomized Arnoldi process based on the RGS algorithm from subsection 2.3 for computing the Krylov basis orthonormal with respect to the sketched product $\langle \mathbf{\Theta}\cdot, \mathbf{\Theta}\cdot \rangle$, rather than ℓ_2-inner product as in standard methods (see Algorithm 4.1). This process will serve as the core for the randomized GMRES method in subsection 4.2.

In Algorithm 4.1, the computation of the matrix-vector product in step 3 with the fine unit roundoff u_{fine} is assumed to have only a minor impact on the overall computational costs. This can be the case, for instance, when the matrix $\mathbf{A}$ is sparse or structured. Furthermore, if needed, the matrix-vector product can be computed also with a larger unit roundoff as long as the associated error satisfies $\|\widehat{\mathbf{w}}_i - \mathbf{A}\widehat{\mathbf{q}}_{i-1}\| = \mathcal{O}(u_{crs} m^{-1/2})\|\widehat{\mathbf{q}}_{i-1}\|$ required by Propositions 4.1 and 4.2.

Let us now address the accuracy of Algorithm 4.1. Clearly, if $\mathbf{\Theta}$ is an ε-embedding for $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$, then Algorithm 4.1 in infinite precision arithmetic produces a well-conditioned basis matrix $\mathbf{Q}_m$ satisfying the Arnoldi identity. In addition, we clearly

Algorithm 4.1. RGS-Arnoldi algorithm

```
Given: $n \times n$ matrix $\mathbf{A}$, $n \times 1$ vector $\mathbf{b}$, $k \times n$ matrix $\mathbf{\Theta}$ with $k \ll n$, parameter $m$.
Output: $n \times m$ factor $\mathbf{Q}_m$ and $m \times m$ upper triangular factor $\mathbf{R}_m$.
1. Set $\mathbf{w}_1 = \mathbf{b}$.
2. Perform 1st iteration of Algorithm 2.1.
for $i = 2 : m$ do
   3. Compute $\mathbf{w}_i = \mathbf{A}\mathbf{q}_{i-1}$.                    # macheps: $u_{fine}$
   4. Perform $i$th iteration of Algorithm 2.1.
end for
5. (Optional) Compute $\Delta_m$ and $\tilde{\Delta}_m$.
   Use Proposition 4.1 to certify the output.                # macheps: $u_{fine}$
```

have $\text{range}(\mathbf{Q}_m) = \text{range}(\mathbf{W}_m) = \mathcal{K}_m(\mathbf{A}, \mathbf{b})$. Since $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ and $\mathbf{\Theta}$ are independent, the matrix $\mathbf{\Theta}$ can be readily chosen as (ε, δ, m) oblivious ℓ_2-subspace embedding to have the ε-embedding property with high probability.

Numerical stability of Algorithm 4.1 in finite precision arithmetic can be derived directly from Theorems 3.2 and 3.3 characterizing the stability of the RGS algorithm.

Proposition 4.1. *Assume that*

$$100 m^{1/2} n^{3/2} u_{fine} \leq u_{crs} \leq 0.01 m^{-1} \text{ and } n \geq 100,$$

along with Assumptions 3.1. *If* $\mathbf{\Theta}$ *satisfies the* ε*-embedding property for* $\widehat{\mathbf{Q}}_m$ *and* $\widehat{\mathbf{W}}_m$ *with* $\varepsilon \leq 1/2$ *and* $\Delta_m, \tilde{\Delta}_m \leq 0.1$, *then we have*

$$(1+\varepsilon)^{-1/2}(1-\Delta_m-0.1u_{crs}) \leq \sigma_{min}(\widehat{\mathbf{Q}}_m) \leq \sigma_{max}(\widehat{\mathbf{Q}}_m) \leq (1-\varepsilon)^{-1/2}(1+\Delta_m+0.1u_{crs}).$$

We also have

$$(\mathbf{A} + \mathbf{\Delta A})\widehat{\mathbf{Q}}_{m-1} = \widehat{\mathbf{Q}}_m \widehat{\mathbf{H}}_m$$

for some matrix $\mathbf{\Delta A}$ *with* $\text{rank}(\mathbf{\Delta A}) < m$ *and* $\|\mathbf{\Delta A}\|_{\text{F}} \leq 15 u_{crs} m^2$.

Proof. See supplementary material (M138870SupMat.pdf [local/web 390KB]). □

It follows from Proposition 4.1 that the stability of the proposed RGS-Arnoldi algorithm can be guaranteed by computing or bounding a priori coefficients Δ_m and $\tilde{\Delta}_m$. Proposition 4.1 can be viewed as a backward stability characterization, since it implies that

$$\text{range}(\widehat{\mathbf{Q}}_{m-1}) = \mathcal{K}_{m-1}(\mathbf{A} + \mathbf{\Delta A}, \mathbf{b} + \mathbf{\Delta b}),$$

where $\|\mathbf{\Delta A}\|_{\text{F}} \leq 15 u_{crs} m^2$ and $\|\mathbf{\Delta b}\| = \|\hat{r}_{1,1}\widehat{\mathbf{q}}_1 - \mathbf{b}\| \leq u_{fine}$. In other words, according to Proposition 4.1, the output of Algorithm 4.1 in finite precision arithmetic is guaranteed to produce a well-conditioned basis $\widehat{\mathbf{Q}}_{m-1}$ for the Krylov space of a slightly perturbed matrix $\mathbf{A}$ and vector $\mathbf{b}$.

Let us next provide a priori bounds for Δ_m and $\tilde{\Delta}_m$. Define parameter

$$\tau(\widehat{\mathbf{Q}}_{m-1}) = \min_{\mathbf{y}_{m-1} \in \text{range}(\widehat{\mathbf{Q}}_{m-1})} \|\mathbf{A}\mathbf{y}_{m-1} - \mathbf{b}\|,$$

representing the best attainable residual error with the computed Krylov basis.

PROPOSITION 4.2. *Consider Algorithm* 4.1 *using the Givens or Householder least-squares solver in step* 2 *of Algorithm* 2.1*, and*

$$u_{crs} \leq 10^{-4}\tau(\widehat{\mathbf{Q}}_{m-1})\mathrm{cond}(\mathbf{A})^{-1}m^{-2},$$
$$u_{fine} \leq (100m^{1/2}n^{3/2} + 10^4m^{3/2}k)^{-1}u_{crs}.$$

Under Assumptions 3.1*, if* $\boldsymbol{\Theta}$ *satisfies the* ε*-embedding property for* $\widehat{\mathbf{Q}}_{m-1}$ *and* $\widehat{\mathbf{W}}_m$ *with* $\varepsilon \leq 1/2$*, then* Δ_m *and* $\tilde{\Delta}_m$ *are bounded by*

$$\tilde{\Delta}_m \leq 6u_{crs}m^{3/2},$$
$$\Delta_m \leq 160u_{crs}m^2\mathrm{cond}(\mathbf{A})\tau(\widehat{\mathbf{Q}}_{m-1})^{-1}.$$

Proof. See supplementary material (M138870SupMat.pdf [local/web 390KB]). □

Proposition 4.2 guarantees numerical stability of Algorithm 4.1 if

$$\tau(\widehat{\mathbf{Q}}_{m-1}) \geq u_{crs}P(m)\mathrm{cond}(\mathbf{A}),$$

where $P(m) = \mathcal{O}(m^2)$ is some low-degree polynomial. Clearly, if τ_0 is the desired tolerance for the GMRES solution, then one is required to use unit roundoff $u_{crs} \leq \tau_0 P(m)^{-1}\mathrm{cond}(\mathbf{A})^{-1}$.

Both Propositions 4.1 and 4.2 hold if $\boldsymbol{\Theta}$ satisfies the ε-embedding property for $\widehat{\mathbf{Q}}_m$ and $\widehat{\mathbf{W}}_m$ with $\varepsilon \leq 1/2$. This assumption comes naturally from probabilistic characteristics of rounding errors and oblivious embeddings. In particular, we can think of similar considerations as in subsection 2.2 to justify the ε-embedding property of $\boldsymbol{\Theta}$ for $\mathcal{K}_m(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}, \mathbf{b}+\boldsymbol{\Delta}\mathbf{b})$, where matrix $\boldsymbol{\Delta}\mathbf{A}$ has rows with entries that are independent centered random variables. The a priori analysis of the ε-embedding property for a perturbed Krylov space, however, is not as trivial and is left for future research. Note that the output of Algorithm 4.1 can be proven reliable a posteriori by efficient certification of the ε-embedding property with the procedure from subsection 3.2.

4.2. Randomized GMRES. The randomized GMRES method is directly derived from the randomized Arnoldi iteration. Let $\widehat{\mathbf{Q}}_m$ and $\widehat{\mathbf{H}}_m$ be the basis matrix and the Hessenberg matrix computed with Algorithm 4.1. (The randomized) GMRES method then proceeds with obtaining the solution $\mathbf{y}_{m-1}$ to the small least-squares problem

$$\mathbf{y}_{m-1} = \arg \min_{\mathbf{z}_{m-1}\in\mathbb{R}^{m-1}} \|\widehat{\mathbf{H}}_m\mathbf{z}_{m-1} - \hat{r}_{1,1}\mathbf{e}_1\|, \tag{4.2}$$

yielding an approximate solution $\mathbf{x}_{m-1} = \widehat{\mathbf{Q}}_{m-1}\mathbf{y}_{m-1}$ to (4.1). The underlined least-squares problem can be efficiently solved with a QR factorization based on Givens rotations or Householder transformation (or any other methods) in sufficient precision.

In infinite precision arithmetic, the orthogonality of $\mathbf{Q}_m$ implies that solving (4.2) is equivalent to minimizing the *sketched* norm of the residual. More specifically, we have

$$\begin{aligned}\min_{\mathbf{z}_{m-1}\in\mathbb{R}^{m-1}} \|\mathbf{H}_m\mathbf{z}_{m-1} - r_{1,1}\mathbf{e}_1\| &= \min_{\mathbf{z}_{m-1}\in\mathbb{R}^{m-1}} \|\boldsymbol{\Theta}\mathbf{Q}_m(\mathbf{H}_m\mathbf{z}_{m-1} - r_{1,1}\mathbf{e}_1)\| \\ &= \min_{\mathbf{z}_{m-1}\in\mathbb{R}^{m-1}} \|\boldsymbol{\Theta}(\mathbf{A}\mathbf{Q}_{m-1}\mathbf{z}_{m-1} - \mathbf{b})\|.\end{aligned} \tag{4.3}$$

Thus, the GMRES solution $\mathbf{x}_{m-1}$ minimizes the residual error up to a factor $\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$, provided $\boldsymbol{\Theta}$ is an ε-embedding for $\mathbf{Q}_m$.

Numerical stability of the randomized GMRES can be characterized by using Proposition 4.1 that yields the following result.

PROPOSITION 4.3. *Let Assumptions* 3.1 *hold. Assume that* $\mathbf{\Theta}$ *is an* ε*-embedding for* $\widehat{\mathbf{Q}}_m$ *and* $\widehat{\mathbf{W}}_m$*, with* $\varepsilon \leq 1/2$ *and* $\Delta_m, \tilde{\Delta}_m \leq 0.1$*; then we have*

$$\|(\mathbf{A}+\mathbf{\Delta A})\mathbf{x}_{m-1} - (\mathbf{b}+\mathbf{\Delta b})\| \leq \mathrm{cond}(\widehat{\mathbf{Q}}_m) \min_{\mathbf{v}\in\mathcal{K}_{m-1}(\mathbf{A}+\mathbf{\Delta A},\mathbf{b}+\mathbf{\Delta b})} \|(\mathbf{A}+\mathbf{\Delta A})\mathbf{v} - (\mathbf{b}+\mathbf{\Delta b})\|$$

for some matrix $\mathbf{\Delta A}$ *and vector* $\mathbf{\Delta b}$ *with* $\|\mathbf{\Delta A}\|_{\mathrm{F}} \leq 15u_{crs}m^2$ *and* $\|\mathbf{\Delta b}\| \leq u_{fine}$.

Proof. See supplementary material (M138870SupMat.pdf [local/web 390KB]). □

Notice that for sufficiently small Δ_m and $\tilde{\Delta}_m$, $\mathrm{cond}(\widehat{\mathbf{Q}}_m)$ is close to $\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$. Consequently, Proposition 4.1 guarantees that $\mathbf{x}_{m-1}$ is a quasi-optimal minimizer of the residual error over a (slightly) perturbed Krylov space.

5. Numerical experiments.

In this section the proposed methodology is verified in a series of numerical experiments and compared against classical methods. In the randomized algorithms, several sizes k and distributions for the sketching matrices $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are tested. We use in step 2 of Algorithm 2.1 the Householder least-squares solver. There was not detected any significant difference in performance (i.e., stability or accuracy of approximation for the same k) between the Rademacher (or Gaussian) distribution and (P-)SRHT, even though the theoretical bounds for (P-)SRHT are worse. Therefore, in this section we present only the results for the (P-)SRHT distribution.

For better presentation, the orthogonality of the sketch $\mathbf{S}_m$ is here measured by the condition number $\mathrm{cond}(\mathbf{S}_m)$ instead of the coefficient $\Delta_m = \|\mathbf{I}_{m\times m} - \mathbf{S}_m^{\mathrm{T}}\mathbf{S}_m\|_{\mathrm{F}}$ as in the previous sections.

5.1. Construction of an orthogonal basis for synthetic functions.

Let us first consider construction of an orthogonal basis approximating the functions

$$f_\mu(x) = \frac{\sin(10(\mu+x))}{\cos(100(\mu-x))+1.1},\ x\in[0,1],$$

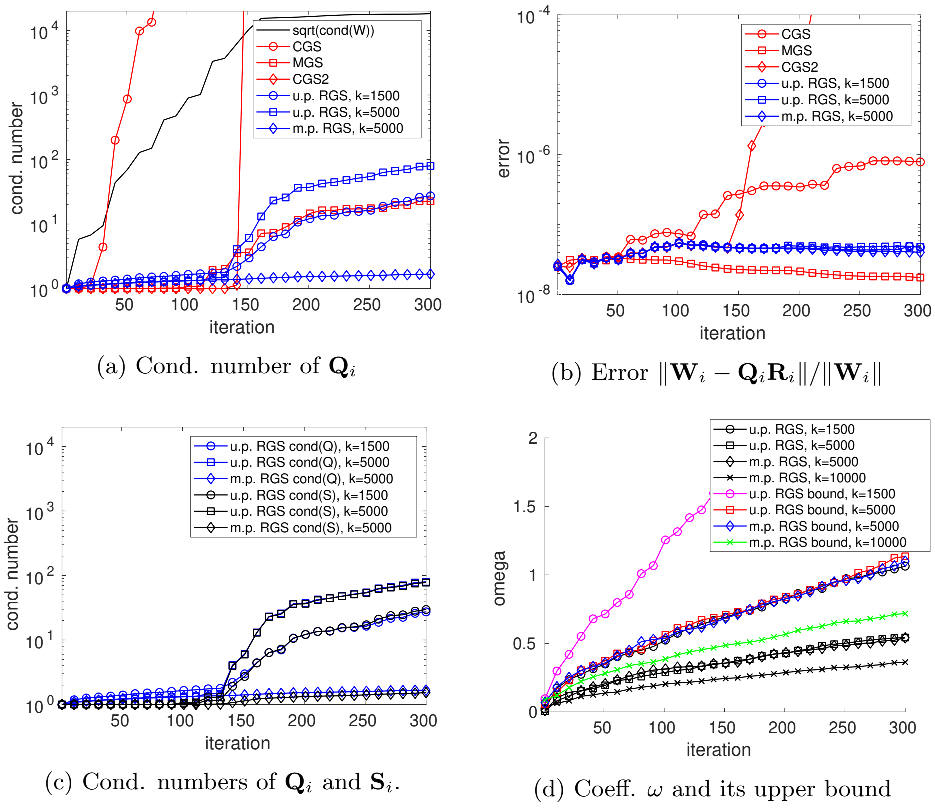
for parameter values $\mu \in [0,1]$.

The function's domain is discretized with $n = 10^6$ evenly spaced points x_j, while the parameter set is discretized with $m = 300$ evenly spaced points μ_j. Then a QR factorization of the matrix $[\mathbf{W}]_{i,j} = f_{\mu_j}(x_i)$, $1 \leq i \leq n$, $1 \leq j \leq m$, is performed with standard versions (CGS, MGS, and CGS2) of the GS process, along with the randomized version of the process, given by Algorithm 2.1. The classical algorithms are executed in float32 format with unit roundoff $\approx 10^{-8}$. Algorithm 2.1 is first executed using a unique float32 format for all the arithmetic operations, i.e., by taking $u_{crs} = u_{fine} \approx 10^{-8}$. Then, the results are compared to Algorithm 2.1 under the multiprecision model executing step 3 in float32 while executing other operations in float64, i.e., by taking $u_{crs} \approx 10^{-8}$ and $u_{fine} = 10^{-16}$. Note that the execution of Algorithm 2.1 with the unique float32 format has nearly the same computational cost as with the mixed float32/float64 formats. Furthermore, as was argued in subsection 2.4, the RGS algorithm here requires half the flops[6] and data passes than CGS

[6] We did not take into consideration the flops associated with the solutions of $k \times (i-1)$ least-squares problems in step 2 of Algorithm 2.1, which will become irrelevant for larger dimensions n. They could be reduced by solving the least-squares problems (iteratively) with normal equation.

and, respectively, four times fewer flops and data passes than CGS2. Moreover, unlike MGS, it is implemented by using BLAS-2 routines for standard high-dimensional operations.

Figure 5.1(a) presents the evolution of the condition number of the computed Q factor at each iteration of the GS process. The evolution of (square root of) the condition number of $\mathbf{W}_i$, $1 \leq i \leq m$, is also depicted. We see that for $i \geq 150$, $\mathbf{W}_i$ becomes numerically singular. For the CGS and CGS2 methods, dramatic instabilities are observed at iterations $i \geq 50$ and $i \geq 150$, respectively. The MGS method exhibits more robustness than the other two standard variants of the GS process. With this method, the condition number of $\mathbf{Q}_i$ remains close to 1 up to iteration $i = 130$ and then gradually degrades by more than an order of magnitude. The RGS algorithm executed in unique float32 format with $k = 1500$ presents a similar stability as MGS. We see from Figure 5.1(a) that even though increasing of k from 1500 to 5000 improves the quality of $\mathbf{\Theta}$ in terms of the ε-embedding property, the usage of $k = 5000$ does not improve the stability of the RGS algorithm in unique float32 format but only worsens it. This can be explained by the increased rounding errors in computations of random projections and solutions of least-squares problems in step 2. The multiprecision RGS algorithm, on the other hand, does not present this behavior. It provides a Q factor



(a) Cond. number of $\mathbf{Q}_i$

(b) Error $\|\mathbf{W}_i - \mathbf{Q}_i\mathbf{R}_i\|/\|\mathbf{W}_i\|$

(c) Cond. numbers of $\mathbf{Q}_i$ and $\mathbf{S}_i$.

(d) Coeff. ω and its upper bound

FIG. 5.1. *The construction of orthogonal basis for synthetic functions $f_\mu(x)$. In the plots, u.p. RGS and m.p. RGS, respectively, refer to the unique precision RGS and the multiprecision RGS algorithms.*

with the condition number close to $1 + \mathcal{O}(\varepsilon)$ and, particularly, an order of magnitude smaller than the condition number of the MGS Q factor.

The evolution of the approximation error $\|\mathbf{W}_i - \mathbf{Q}_i\mathbf{R}_i\|/\|\mathbf{W}_i\|$ is depicted in Figure 5.1(b). We see that for CGS the error at first is close to the machine precision, but then it gradually degrades by two orders of magnitude. For CGS2 a dramatically large error is observed at iterations $i \geq 150$. For MGS and RGS algorithms the error remains close to the machine precision at all iterations.

Figure 5.1(c) addresses a posteriori verification of the quality of the computed Q factor from its sketch. We see that indeed $\mathrm{cond}(\mathbf{Q}_i)$ can be well estimated by $\mathrm{cond}(\mathbf{S}_i)$.

Recall that the stability characterization of the RGS algorithm in subsection 3.1 relies on the ε-embedding property of $\mathbf{\Theta}$. The minimal value ω of ε for which $\mathbf{\Theta}$ satisfies the ε-embedding property for $\mathbf{Q}_i$, at each iteration, is provided in Figure 5.1(d). We also show the upper bound $\bar{\omega}$ for ω computed with Proposition 3.7 from the sketches with no operations on high-dimensional vectors or matrices. In Proposition 3.7, the matrix $\mathbf{\Phi}$ was chosen to be of same size as $\mathbf{\Theta}$. Moreover, the parameter ε^* was taken as 0.05. It is observed that for both the unique precision and the multiprecision algorithms with $k \geq 5000$, the matrix $\mathbf{\Theta}$ satisfies the ε-embedding property with (almost) $\varepsilon \leq 1/2$, which is the condition used in subsection 3.1 for deriving stability guarantees for the RGS algorithm. For $k = 1500$ at iterations $i \geq 70$, the value of ω becomes larger than 1/2. Nevertheless, it remains small enough, which suggests a sufficient stability of the RGS algorithm also for this value of k and correlates well with the experiments (see Figure 5.1(a)). The estimator $\bar{\omega}$ of ω remains an upper bound of ω at all iterations and values of k, which implies robustness of Proposition 3.7 for characterizing the ε-embedding property of $\mathbf{\Theta}$. An overestimation of ω by nearly a factor of 2 is revealed at all iterations and values of k. Moreover, this behavior of $\bar{\omega}$ is observed also in other experiments. This suggests that, in practice, the value of $\bar{\omega}$ can be divided by a factor of 2.

5.2. Orthogonalization of solution samples of a parametric PDE.

Next we consider a model order reduction problem from [5, section 6.1]. This problem describes a wave scattering with an object covered in an acoustic invisibility cloak. The cloak is multilayered. The problem is governed by a parametric PDE, where the parameters are the properties of materials composing the last 10 layers of the cloak, and the wave frequency. By discretization with second-order finite elements, the parametric PDE is further transformed into a complex-valued system of equations of the form

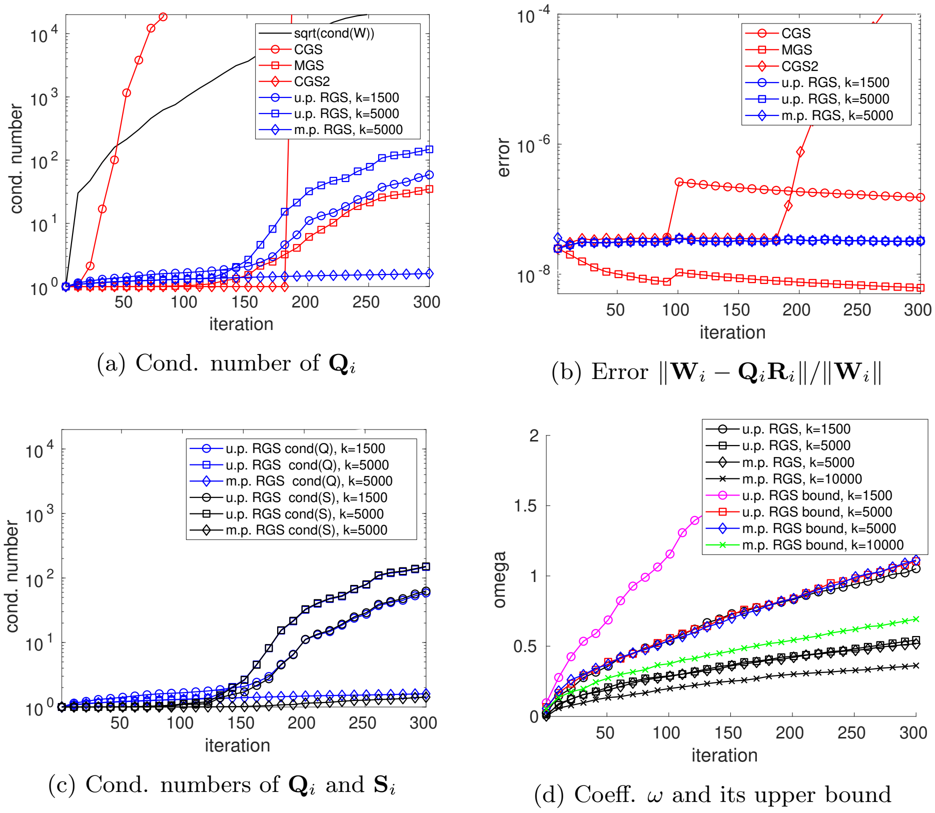
$$\mathbf{A}_\mu \mathbf{u}_\mu = \mathbf{b}_\mu, \tag{5.1}$$

where $\mathbf{A}_\mu \in \mathbb{C}^{n\times n}$ and $\mathbf{b}_\mu \in \mathbb{C}^n$ with $n \approx 400000$. The aim in [5] is to solve (5.1) for parameters μ from the parameter set of interest $\mathcal{P}$. See [5] for more detailed description of the problem.

Let us consider the construction of an orthogonal basis (so-called reduced basis) approximating the set $\{\mathbf{u}_\mu : \mu \in \mathcal{P}\}$. For this, we drew from $\mathcal{P}$ $m = 300$ uniform samples $\mu_1, \mu_2, \ldots, \mu_m$ and then performed a QR factorization of the matrix

$$\mathbf{W} = [\mathbf{u}_{\mu_1}, \mathbf{u}_{\mu_2}, \ldots, \mathbf{u}_{\mu_m}]$$

with the following versions of the GS process: CGS, CGS2, and MGS computed in float32 format, the unique precision RGS in float32, and the multiprecision RGS



(a) Cond. number of $\mathbf{Q}_i$

(b) Error $\|\mathbf{W}_i - \mathbf{Q}_i\mathbf{R}_i\|/\|\mathbf{W}_i\|$

(c) Cond. numbers of $\mathbf{Q}_i$ and $\mathbf{S}_i$

(d) Coeff. ω and its upper bound

FIG. 5.2. *The construction of orthogonal basis for the solution set of a parametric PDE. In the plots, u.p. RGS and m.p. RGS, respectively, refer to the unique precision RGS and the multiprecision RGS algorithms.*

using float32 for standard high-dimensional operations, while using float64 for other operations.[7]

The condition number of the factor $\mathbf{Q}_i$ and the approximation error $\|\mathbf{W}_i - \mathbf{Q}_i\mathbf{R}_i\|/\|\mathbf{W}_i\|$ obtained at each iteration of the algorithms are depicted in Figure 5.2(a) and (b), respectively. Furthermore, for randomized algorithms, in Figure 5.2(c) we provide a comparison of $\text{cond}(\mathbf{Q}_i)$ and $\text{cond}(\mathbf{S}_i)$. In Figure 5.2(d) we present the characterization of the ε-embedding property of $\mathbf{\Theta}$ for $\mathbf{Q}_i$ given by the value ω and its upper bound computed with Proposition 3.7. In Proposition 3.7 we chose $\mathbf{\Phi}$ to be of same size as $\mathbf{\Theta}$ with the parameter $\varepsilon^* = 0.05$.

A very similar picture is observed as in the previous numerical example. More specifically, dramatic instabilities are revealed at iterations $i \geq 50$ for CGS and $i \geq 190$ for CGS2. The MGS and the unique precision RGS with $k = 1500$ show a similar stability, which is better than the one of CGS and CGS2. For these algorithms $\text{cond}(\mathbf{Q}_i)$ remains close to 1 at iterations $i \leq 150$ but degrades by more than an order of magnitude at latter iterations. The multiprecision RGS algorithm at all iterations provides a Q factor with condition number close to 1. The approximation error $\|\mathbf{W}_i - \mathbf{Q}_i\mathbf{R}_i\|/\|\mathbf{W}_i\|$ again is close to the machine precision for MGS and RGS

[7] The extension of the theoretical analysis of the RGS process from real numbers to complex numbers is straightforward.

algorithms, while it is larger for CGS and CGS2. The condition number of the sketched Q factor, $\mathbf{S}_i$, is verified to be a good estimator of the condition number of $\mathbf{Q}_i$. For both the unique as well as the multiprecision RGS algorithms with $k \geq 5000$, the condition $\omega \leq 1/2$ used in the stability analysis is (nearly) satisfied. For $k = 1500$, the value of ω is larger than $1/2$ at iterations $i \geq 70$, though it remains sufficiently small suggesting the stability of the RGS algorithm also for this sketch size. Again, we reveal an overestimation of $\bar{\omega}$ as an upper bound of ω by nearly a factor of 2.

5.3. Solution of a linear system with GMRES. In this numerical experiment the RGS algorithm is tested in the context of the GMRES method for the solution of the linear system of equations:

$$\mathbf{A}_f \mathbf{x}_f = \mathbf{b}, \tag{5.2}$$

where the matrix $\mathbf{A}_f$ is taken as the "SiO2" matrix of dimension $n = 155331$ from the SuiteSparse matrix collection. The right-hand-side vector $\mathbf{b}$ is taken as $\mathbf{b} = \mathbf{A}\mathbf{y}/\|\mathbf{A}\mathbf{y}\|$, where $\mathbf{y} = [1, 1, \ldots, 1]^{\mathrm{T}}$. Furthermore, the system (5.2) is preconditioned from the right by the incomplete LU factorization $\mathbf{P}_f$ of $\mathbf{A}$ with zero level of fill in. With this preconditioner the final system of equations has the following form:

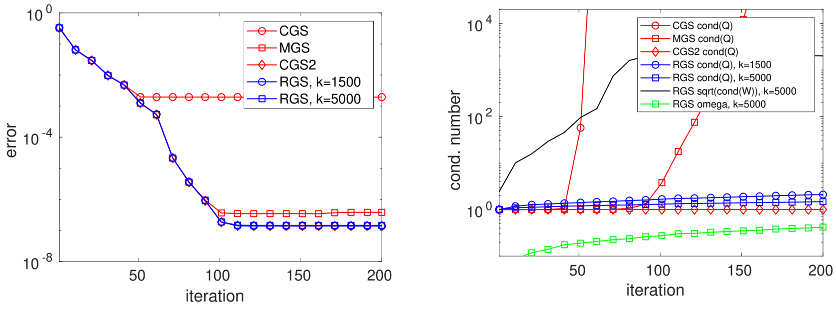
$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

where $\mathbf{A} = \mathbf{A}_f \mathbf{P}_f$ and $\mathbf{x}_f = \mathbf{P}_f \mathbf{x}$. This system is considered for the solution with the GMRES method based on different versions of the GS process. Here we test only CGS, CGS2, MGS, and the unique precision RGS. The multiprecision RGS is not considered since the unique precision RGS already provides a nearly optimal solution.

In all experiments, the products with matrix $\mathbf{A}$ are computed in float64 format. The solutions of the Hessenberg least-squares problems (4.2) are computed with Givens rotations also in float64 format. All other operations (i.e., the GS iterations) are performed in float32 format.

The convergence of the residual error is depicted in Figure 5.3(a). The condition number of the Q factor (characterizing the orthogonality of the computed Krylov basis) at each iteration i is provided in Figure 5.3(b). In Figure 5.3(b) we also provide the condition number of $\mathbf{W}_i = [\mathbf{A}\mathbf{Q}_{i-1},\ \mathbf{b}]$ and the value of ω representing the ε-embedding property of $\mathbf{\Theta}$ for $\mathbf{Q}_i$ in the RGS algorithm with $k = 5000$.

At iterations $i \geq 50$, we reveal a dramatic instability of the CGS algorithm resulting in the early stagnation of the residual error. The other versions of the



(a) Residual error $\|\mathbf{A}\widehat{\mathbf{x}}_{i-1} - \mathbf{b}\|$.

(b) Cond. number of $\mathbf{Q}_i$

FIG. 5.3. *Solution of a linear system with GMRES.*

GS process present a better stability. The CGS2 and RGS with all sketch sizes at all iterations yield almost orthogonal Q factor. For these algorithms the error has converged to machine precision. The MGS algorithm does not provide a well-conditioned Q factor at $i \geq 110$ iterations. Nevertheless, it yields the convergence of the residual error up to machine precision similar to CGS2 and RGS.

Finally, we see that for RGS with $k = 5000$, $\mathbf{\Theta}$ is verified to be an ε-embedding for $\mathbf{Q}_i$ with $\varepsilon \leq 1/2$. This implies applicability of the stability analysis from subsection 3.1.

6. Conclusion.

In this article we proposed a novel RGS process for efficient orthogonalization of a set of high-dimensional vectors. This process can be incorporated into the GMRES method or Arnoldi iteration for solving large systems of equations or eigenvalue problems. Our methodology can be adapted to practically any computational architecture.

The RGS process was introduced under a multiprecision arithmetic model that also accounts for the classical unique precision model. We proposed to perform expensive high-dimensional operations in low precision while computing the inexpensive random projections and low-dimensional operations in high precision. The numerical stability of the algorithms was shown for the low-precision unit roundoff independent of the dimension of the problem. This feature can have a major importance when solving extreme-scale problems.

The great potential of the methodology was realized with three numerical examples. In all the experiments, the multiprecision RGS algorithm provided a Q factor (i.e., the orthogonalized matrix) with condition number close to 1. It remained stable even in extreme cases, such as orthogonalization of a numerically singular matrix, where the standard CGS and CGS2 methods failed. This is in addition to the fact that the RGS algorithm can require nearly half as many flops and passes over the data than CGS.

REFERENCES

[1] N. N. Abdelmalek, *Round off error analysis for Gram–Schmidt method and solution of linear least squares problems*, BIT, 11 (1971), pp. 345–367.

[2] D. Achlioptas, *Database-friendly random projections: Johnson-Lindenstrauss with binary coins*, J. comput. Syst. Sci., 66 (2003), pp. 671–687.

[3] N. Ailon and E. Liberty, *Fast dimension reduction using Rademacher series on dual BCH codes*, Discrete Comput. Geom., 42 (2009), p. 615.

[4] O. Balabanov and A. Nouy, *Randomized linear algebra for model reduction. Part* I*: Galerkin methods and error estimation*, Adv. Comput. Math., 45 (2019), pp. 2969–3019.

[5] O. Balabanov and A. Nouy, *Randomized linear algebra for model reduction. Part* II*: Minimal residual methods and dictionary-based approximation*, Adv. Comput. Math., 47 (2021), pp. 1–54.

[6] J. L. Barlow, A. Smoktunowicz, and H. Erbay, *Improved Gram–Schmidt type downdating methods*, BIT, 45 (2005), pp. 259–285.

[7] A. Björck, *Solving linear least squares problems by Gram–Schmidt orthogonalization*, BIT, 7 (1967), pp. 1–21.

[8] E. Carson, T. Gergelits, and I. Yamazaki, *Mixed precision s-step Lanczos and conjugate gradient algorithms*, Nume. Linear Algebra Appl., in press.

[9] M. P. Connolly, N. J. Higham, and T. Mary, *Stochastic rounding and its probabilistic backward error analysis*, SIAM J. Sci. Comput., 43 (2021), pp. A566–A585.

[10] J. Drkošová, A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical stability of gmres*, BIT, 35 (1995), pp. 309–330.

[11] L. Giraud, S. Gratton, and J. Langou, *Convergence in backward error of relaxed gmres*, SIAM J. Sci. Comput., 29 (2007), pp. 710–728.

[12] L. Giraud, J. Langou, and M. Rozloznik, *The loss of orthogonality in the Gram–Schmidt orthogonalization process*, Comput. Math. Appl., 50 (2005), pp. 1069–1075.

[13] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof, *Rounding error analysis of the classical Gram–Schmidt orthogonalization process*, Numer. Math., 101 (2005), pp. 87–100.

[14] S. Gratton, E. Simon, D. Titley-Peloquin, and P. Toint, *Exploiting Variable Precision in GMRES*, preprint, arXiv:1907.10550, 2019.

[15] A. Greenbaum, M. Rozložnik, and Z. Strakoš, *Numerical behaviour of the modified Gram-Schmidt gmres implementation*, BIT, 37 (1997), pp. 706–719.

[16] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[17] N. Higham and T. Mary, *Sharper probabilistic backward error analysis for basic linear algebra kernels with random data*, SIAM J. Sci. Comput., 42 (2020), pp. A3422–A3446.

[18] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM 2nd ed., Philadelphia, 2002.

[19] N. J. Higham and T. Mary, *A new approach to probabilistic rounding error analysis*, SIAM J. Sci. Comput., 41 (2019), pp. A2815–A2835.

[20] I. C. Ipsen and H. Zhou, *Probabilistic Error Analysis for Inner Products*, preprint, arXiv:1906.10465, 2019.

[21] S. K. Kim and A. Chrortopoulos, *An efficient parallel algorithm for extreme eigenvalues of sparse nonsymmetric matrices*, Int. J. Super Comput. Appl., 6 (1992), pp. 98–111.

[22] S. J. Leon, Å. Björck, and W. Gander, *Gram-Schmidt orthogonalization:* 100 *years and more*, Numer. Linear Algebra Appl., 20 (2013), pp. 492–532.

[23] J. Malard and C. Paige, *Efficiency and scalability of two parallel QR factorization algorithms*, in Proceedings of the IEEE Scalable High Performance Computing Conference, IEEE, 1994, pp. 615–622.

[24] C. C. Paige, M. Rozlozník, and Z. Strakos, *Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284.

[25] D. Pollard, *Lecture Notes in Advanced Probability*, monoscript, Yale University, Department of Statistics and Data Science, 2017.

[26] M. Rozlozník, *Numerical Stability of the GMRES Method*, Manuscript, 1996.

[27] A. Ruhe, *Numerical aspects of Gram-Schmidt orthogonalization of vectors*, Linear Algebra appl., 52 (1983), pp. 591–601.

[28] T. Sarlos, *Improved approximation algorithms for large matrices via random projections*, in Proceddings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2006, pp. 143–152.

[29] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.

[30] A. Smoktunowicz, J. L. Barlow, and J. Langou, *A note on the error analysis of classical Gram–Schmidt*, Numer. Math., 105 (2006), pp. 299–313.

[31] K. Świrydowicz, J. Langou, S. Ananthan, U. Yang, and S. Thomas, *Low synchronization Gram–Schmidt and generalized minimal residual algorithms*, Numer. Algebra Appl., 28 (2021), p. e2343.

[32] J. Van Den Eshof and G. L. Sleijpen, *Inexact Krylov subspace methods for linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 125–153.

[33] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*, Cambridge University Press, Cambridge, UK, 2018.

[34] D. P. Woodruff, *Sketching as a tool for numerical linear algebra*, Foun. Trends Theor. Comput. Sci., 10 (2014), pp. 1–157.

[35] I. Yamazaki, S. Tomov, T. Dong, and J. Dongarra, *Mixed-precision orthogonalization scheme and adaptive step size for improving the stability and performance of CA-GMRES on GPUs*, in Proceedings of the International Conference on High Performance Computing for Computational Science, Springer, 2014, pp. 17–30.