# Randomized Householder QR

Laura Grigori[*], Edouard Timsit[†]

July 7th, 2023

**Abstract:** This paper introduces a randomized Householder QR factorization (RHQR). This factorization can be used to obtain a well conditioned basis of a set of vectors and thus can be employed in a variety of applications. We discuss in particular the usage of this randomized Householder factorization in the Arnoldi process. Numerical experiments show that RHQR produces a well conditioned basis and an accurate factorization. We observe that for some cases, it can be more stable than Randomized Gram-Schmidt (RGS) in single precision.

## 1 Introduction

Computing the QR factorization of a matrix $W \in \mathbb{R}^{n\times m}$, $m \ll n$, where $Q \in \mathbb{R}^{n\times m}$ has orthonormal columns and $R \in \mathbb{R}^{m\times m}$ is upper triangular, is a process that arises in a variety of applications. Since it allows to compute both an orthogonal basis of $W$ and the linear least squares problem, it lies at the heart of Krylov subspace methods used to solve linear systems of equations and eigenvalue problems. In finite precision arithmetics, the level of orthogonality achieved by the columns of a matrix $Q$ can be measured by its condition number, which is the ratio of its largest and its smallest singular values. In the case of perfect orthogonality, this ratio is 1 (*well-conditioned matrix*). As the condition number increases, the computation of coordinates in the basis $Q$ becomes less accurate (*ill-conditioned matrix*). A QR factorization is usually obtained using the Gram-Schmidt process or the Householder process. The Householder process relies on *Householder vectors* $u_1, \cdots, u_m \in \mathbb{R}^n$ and orthogonal matrices called *Householder reflectors* $H(u_1), \cdots, H(u_m) \in \mathbb{R}^{n\times n}$ such that

$$W = H(u_1)\cdots H(u_m)\begin{bmatrix} R \\ 0_{(n-m)\times m}\end{bmatrix} = \left(I_n - UTU^t\right)\begin{bmatrix} R \\ 0_{(n-m)\times m}\end{bmatrix}, \tag{1}$$

where $T \in \mathbb{R}^{m\times m}$, $U = [u_1 \mid \cdots \mid u_m] \in \mathbb{R}^{n\times m}$ is lower triangular, $R \in \mathbb{R}^{m\times m}$ is upper triangular, and $0_{(n-m)\times m}$ denotes a zero matrix of dimensions $(n-m)\times m$. J.H. Wilkinson showed in [11] that the Householder QR factorization process is normwise backward stable in finite precision. Later on, it has been shown in [7, Chapter 19.3, Thm 19.4 p360] that the Householder procedure is column-wise backward stable in finite precision. Overall, computations with Householder reflectors are well-known for their excellent numerical stability.

Applying the Woodburry-Morrison formula to $I_n - UTU^t$, it was outlined in [9] that

$$T = \left(\mathrm{sut}\left(U^tU\right) + I_m\right)^{-1}, \tag{2}$$

where sut denotes the strictly upper triangular part of a matrix. This leads to the following factored form of a sequence of Householder reflectors:

$$H(u_1)\cdots H(u_m) = I_n - U\left(\mathrm{sut}\left(U^tU\right) + I_m\right)^{-1}U^t. \tag{3}$$

[*]Laboratory for Simulation and Modelling, Paul Scherrer Institute, Switzerland; Institute of Mathematics, EPFL, Switzerland; part of the work was performed while the author was at Sorbonne Université, Inria, CNRS, Université de Paris, Laboratoire Jacques-Louis Lions, Paris, France.

[†]Sorbonne Université, Inria, CNRS, Université de Paris, Laboratoire Jacques-Louis Lions, Paris, France

An idea from Charles Sheffield and praised by Gene Golub led to the analysis in [8] revealing, among other things, that the factor $\left(\mathrm{sut}\left(U^t U\right) + I_m\right)$ captures the loss of orthogonality in finite precision Modified Gram Schmidt (MGS). Accordingly, new stable implementations of MGS were derived [4, 3] (see comparisons of different implementations in the recent review [5]).

In this work, we introduce a randomized version of Householder QR factorization that allows to reduce its computational cost. For this, we rely on a dimension reduction technique called subspace embedding. Using a particular matrix $\Omega \in \mathbb{R}^{\ell \times n}$ with $\ell \ll n$, this technique allows, for all $x, y$ in a vector subspace of interest, to approximate $\langle x, y \rangle$ ($2n$ flops) by $\langle \Omega x, \Omega y \rangle$ ($2\ell$ flops). We define randomized Householder reflectors $H^{\Omega}(u_1), \cdots H^{\Omega}(u_m)$ such that

$$H^{\Omega}(u_1) \cdots H^{\Omega}(u_m) = I_n - U\left(\mathrm{sut}\left((\Omega U)^t \Omega U\right) + I_m\right)^{-1} \mathrm{ut}\left((\Omega U)^t \Omega\right), \tag{4}$$

where ut denotes the upper triangular part of a matrix. This allows to produce the factorization $W = QR$, where $Q \in \mathbb{R}^{n \times m}$ is well conditioned and $R \in \mathbb{R}^{m \times m}$ is upper triangular. Thus $Q$ can be used as a basis for Krylov subspace methods and $R$ to approximate the solution of a linear least squares problem. More generally, it can be used to compute a well-conditioned basis of Range($W$).

The subspace embedding technique was applied in [2] to the Gram-Schmidt process, yielding the Randomized Gram-Schmidt algorithm (RGS). The condition number of the basis built by RGS is similar to that of Modified Gram-Schmidt (MGS) and sometimes even better for half the flops. Furthermore, the communication cost of RGS is the same as that of Classical Gram-Schmidt (CGS) on a parallel computer, but CGS is known to have stability problems. While RGS is shown to produce a basis with a small condition number, the authors also showed that for some difficult cases, the condition number of the basis built by both RGS and MGS increases to $10^2$ in single precision. In the same context of single precision, the basis built by RHQR has condition number $\approx 3.4$. In terms of communication cost, RHQR requires one synchronization per iteration, having thus similar communication cost to RGS and CGS. While here we focus on one precision, we note that RHQR is suitable for mixed precision computation, similarly to RGS.

This paper is organized as follows. Section 2 discusses the classical Householder QR factorization and the subspace embedding technique. Section 3 introduces the randomized Householder reflector, and shows that the application of a sequence of such reflectors to the matrix $W$ yields a triangular factor $R \in \mathbb{R}^{m \times m}$. We then introduce a factored form of such a sequence of randomized Householder reflectors and present our main algorithm, RHQR, and its block version, blockRHQR. Section 4 describes a set of numerical experiments that show that this process produces a well conditioned basis and an accurate factorization of $W$. We first show the accuracy of the factored forms involved in RHQR; we then discuss the condition number of the output $Q$ and the accuracy of the factorization; finally, we show how the Householder-Arnoldi procedure can be adapted to RHQR. A theoretical analysis of the condition number of the $Q$ factor will follow shortly in an upcoming updated version of this work. Given the connection between MGS and Householder QR, it could be possible to derive a randomized MGS procedure based on the derivations in this paper and the formulas in [3, 4].

# 2 Preliminaries

In this section, we first introduce our notations. We then outline the original Householder procedure. Finally, we introduce the subspace embedding technique.

## 2.1 Notations

The symbol $\oslash$ used between two vectors denotes the concatenation of vectors. If $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$, then the vector $x \oslash y \in \mathbb{R}^{n_1+n_2}$ is defined by

$$x \oslash y = \begin{bmatrix} x \\ y \end{bmatrix}$$

For all $2 \le q \le m$, $u_q$ is a vector of $\mathbb{R}^n$ whose first $q-1$ entries are zero. The last $n-q+1$ coordinates of $u_q$ are denoted $v_q \in \mathbb{R}^{n-q+1}$. The vector $u_1 = v_1$ is a vector of $\mathbb{R}^n$. With the concatenation notation,

$$\forall q \in \{2, \cdots m\}, \quad u_q = 0_{q-1} \oslash v_q = \begin{bmatrix} 0_{q-1} \\ v_q \end{bmatrix}, \qquad u_1 := v_1 \in \mathbb{R}^n.$$

The vectors $u_1, \cdots, u_q$ form the following matrix

$$\forall q \in \{1, \cdots m\}, \quad U_q = \begin{bmatrix} u_1 \mid u_2 \mid \cdots \mid u_q \end{bmatrix} \in \mathbb{R}^{n \times q}.$$

For all $q \in \{1, \cdots m\}$, matrices $T_q$ and $S_q$ denote $q \times q$ matrices. We make use of a matrix $\Omega \in \mathbb{R}^{\ell \times n}$ throughout this work, and denote

$$\forall q \in \{2 \dots m\}, \ \overline{\Omega}_q = \begin{bmatrix} 0_{\ell \times (q-1)} & \Omega_q \end{bmatrix} \in \mathbb{R}^{\ell \times n},$$

where $\Omega_q \in \mathbb{R}^{\ell \times (n-q+1)}$ denotes the last $n-q+1$ columns of $\Omega$ (in other words, $\overline{\Omega}_q$ is formed by erasing the first $q-1$ columns of $\Omega$). As a consequence of these notations,

$$\Omega u_q = \overline{\Omega}_q u_q = \Omega_q v_q.$$

The vectors $w_1, \cdots, w_m \in \mathbb{R}^n$ denote the $m$ columns of the input matrix $W \in \mathbb{R}^{n \times m}$ whose QR factorization is computed. We denote, for all column $w_q$ of $W$,

$$w_q = w_{q,1} \oslash w_{q,2}, \ w_{q,1} \in \mathbb{R}^{q-1}, \ w_{q,2} \in \mathbb{R}^{n-q+1},$$

that is $w_{q,1}$ denote the first $q-1$ entries of $w_q$, and $w_{q,2}$ the last $n-q+1$ entries of $w_q$. For all $1 \le i, j \le n$, $e_i^j$ denotes the $i$-th canonical vector of $\mathbb{R}^{n-j+1}$. More simply put, $e_i^j$ is the vector $e_{i+j-1}$ without its first $j-1$ zeros. If the exponent is omitted, $e_i$ denotes the $i$-th canonical vector of $\mathbb{R}^n$.

Given any matrix $A$, we denote respectively $\mathrm{ut}(A)$, $\mathrm{sut}(A)$, $\mathrm{lt}(A)$ and $\mathrm{slt}(A)$ the upper triangular, strictly upper triangular, lower triangular, strictly lower triangular parts of $A$, respectively. We denote $(A)_{i,j}$ the entry of $A$ located on the $i$-th row and the $j$-th column. We denote $A_{j_1:j_2}$ the matrix formed by the $j_1, j_1+1, \cdots j_2$-th columns of $A$. Finally, throughout this work, the letter $\epsilon$ denotes a real number of $]0,1[$.

## 2.2 Householder orthonormalization

The Householder QR factorization is an orthogonalization procedure, alternative to the Gram Schmidt process. While the Gram-Schmidt process focuses on building explicitly the orthogonal factor $Q$, Householder's procedure focuses instead on building the factor $R$ using orthogonal transformations. $Q$ and $Q^t$ are stored in a factored form. The central operation of this procedure is the Householder reflector:

$$\forall z \in \mathbb{R}^n \setminus \{0\}, \quad H(z) = I_n - \frac{2}{\|z\|^2} z\, z^t, \tag{5}$$

yielding

$$\forall z \in \mathbb{R}^n \setminus \{0\}, \ \forall x \in \mathbb{R}^n, \quad H(z) \cdot x = x - \frac{2}{\|z\|^2} \langle z, x \rangle \, z.$$

Remarking that for all $\lambda \in \mathbb{R}^*$, $H(\lambda z) = H(z)$, we usually consider that $z$ is scaled adaptively, such that their first non-zero entry is 1. Otherwise, they can also be scaled systematically such that $\|z\| = \sqrt{2}$, simplifying the expression of $H(z)$ into $I_n - zz^t$. It can be seen that the Householder reflector (5) is a symmetric matrix and that $H(z)^2 = I_n$ (involutary matrix). Those two properties combined show that $H(z)$ is an orthogonal matrix. The vector $z$ is an eigenvector associated to eigenvalue $-1$, and the orthogonal of its span is an $n-1$ dimensional eigenspace associated to eigenvalue 1, i.e the Householder reflector is a reflector with respect to the latter hyperplane. Given a vector $w \in \mathbb{R}^n$, and setting $z = w - \|w\|e_1$ and assuming that $w \neq 0$, one can straightforwardly derive

$$H(z) \cdot w = \|w\|e_1,$$

hence the Householder reflector can be used to annihilate all the coordinates of a vector, except its first entry. By induction, it can be inferred that, given a vector $w_q \in \mathbb{R}^n$, there exists $v_q \in R^{n-q+1}$, $u_q = 0_{q-1} \oslash v_q \in \mathbb{R}^n$ such that

$$H(u_q) \cdot w_q = \begin{bmatrix} I_{q-1} & \\ & H(v_q) \end{bmatrix} \cdot w_q = \begin{bmatrix} w_{q,1} \\ \|w_{q,2}\|e_1^q \end{bmatrix}, \quad w_{q,1} \in \mathbb{R}^{q-1}, \;\; w_{q,2} \in \mathbb{R}^{n-q+1}, \;\; w_q = w_{q,1} \oslash w_{q,2}, \tag{6}$$

that is the reflector $H(u_q)$ now annihilates the coordinates $q+1$ to $n$ of the vector $w$, without modifying its first $q-1$ entries. Hence there exist vectors $u_1, \cdots u_m$ such that

$$H(u_m) \cdots H(u_1)W = \begin{bmatrix} R \\ 0_{(n-m)\times m} \end{bmatrix} \iff W = H(u_1) \cdots H(u_m) \begin{bmatrix} R \\ 0_{(n-m)\times m} \end{bmatrix}$$

where $R$ is upper triangular. The formula for computing the Householder vector $z$ is not stable if $w$ is close to $\lambda e_1$ for some $\lambda \in \mathbb{R}^+$, since

$$\lambda e_1 - \|\lambda e_1\|e_1 = 0$$

hence $z = w - \|w\|e_1 \approx 0$. In this case, we prefer to set $z = w + \|w\|e_1$. It is straightforward to check that, for such $z$, $H(z) \cdot w = -\|w\|e_1$ while

$$\lambda e_1 + \|\lambda e_1\|e_1 = 2\lambda e_1$$

hence $z = w + \|w\|e_1 \approx 2w \neq 0$. Finally, straightforward induction shows that the composition $H(u_1) \cdots H(u_m)$ can be factored as

$$\begin{cases} H(u_1) \cdots H(u_m) = I_n - U_q T_q U_q^t \\ H(u_q) \cdots H(u_1) = I_n - U_q T_q^t U_q^t \end{cases} \tag{7}$$

where $T_q$ is defined by induction. If the vectors $u_1, \cdots u_q$ are scaled such that they are all of norm $\sqrt{2}$, $T_q$ is given by the following formula:

$$T_1 = [1], \quad \forall q \in \{1, \cdots m-1\}, \; T_{q+1} = \begin{bmatrix} T_q & -T_q U_q^t u_{q+1} \\ 0_{1\times q} & 1 \end{bmatrix}.$$

As a triangular matrix with a diagonal of ones, $T_q$ is non-singular. As outlined in [9], we can then apply the Woodburry Morrison formula to $I_n - U_q T_q U_q^t$ and and derive

$$T_q = \left(\mathrm{sut}(U_q^t U_q) + I_q\right)^{-1}.$$

The meaning of this matrix is then unveiled by the analysis found in [8]

## 2.3 Subspace embeddings

**Definition 2.1.** *[12, Definition 1] We say that a matrix $\Omega \in \mathbb{R}^{\ell\times n}$ is an $\epsilon$-embedding of a vector-subspace $\mathcal{W} \subset \mathbb{R}^n$ if and only if*

$$\forall x \in \mathcal{W}, \;\; (1-\epsilon)\|x\|^2 \leq \|\Omega x\| \leq (1+\epsilon)\|x\|^2. \tag{8}$$

By using polarization identities on both $x \mapsto \|\Omega x\|^2$ and $x \mapsto \|x\|^2$, and then using the parallelogram identity, one can find that (8) is equivalent to

$$\forall x, y \in \mathcal{W}, \quad |\langle \Omega x, \Omega y\rangle - \langle x, y\rangle| \le \epsilon \, \|x\| \cdot \|y\|. \tag{9}$$

We say that a vector $q \in \mathbb{R}^n$ is a sketch unit vector if and only if

$$\|\Omega q\| = 1.$$

We say that vectors $q_1, \cdots, q_m$ are sketch orthogonal if and only if

$$\forall\, i < j \in \{1, \cdots m\}, \quad \langle \Omega q_i, \Omega q_j\rangle = 0.$$

We say that they are sketch orthonormal if and only if they are sketch unit vectors and sketch orthogonal.

There exist distributions over $\mathbb{R}^{\ell\times n}$ whose realizations $\Omega \in \mathbb{R}^{\ell\times}$ are an $\epsilon$-embedding of any $m$-dimensional vector subspace $\mathcal{W}_m$ with high probability. These distributions are called oblivious subspace embeddings (OSE).

**Definition 2.2.** *[12, Definition 2] Let $\epsilon, \delta \in \,]0, 1[$. Let $m \le \ell \ll n \in \mathbb{N}^*$. We say that $\Omega \in \mathbb{R}^{\ell\times n}$ comes from oblivious subspace embedding distribution with parameters $(\epsilon, \delta, m)$ and denote $\Omega \in \mathrm{OSE}(\epsilon, \delta, m)$ if and only if for any $m$-dimensional vector subspace $\mathcal{W}_m \subset \mathbb{R}^n$, $\Omega$ is an $\epsilon$-embedding of $\mathcal{W}_m$ with probability at least $1 - \delta$.*

We choose two such distributions : Gaussian OSE and subsampled random Hadamard transform OSE (SRHT OSE). In both cases, as the sampling size $\ell$ increases, the probability of failure $\delta$ and $\epsilon$ decrease. The Gaussian OSE is the simplest OSE, and consists in drawing a matrix $G \in \mathbb{R}^{\ell\times n}$ where the coefficients are i.i.d standard Gaussian variables, and set

$$\Omega = \frac{1}{\sqrt{\ell}} G \tag{10}$$

This technique has the inconvenient of relying on a dense matrix $\Omega$, whose application to a vector $x \in \mathbb{R}^n$ costs $2\ell n$ flops. On the other hand, it has the advantage of scaling well on a parallel computer. As proven in [12], this distribution is an $(\epsilon, \delta, m)$ OSE if $\ell = \mathcal{O}\left(\epsilon^{-2}(m - \log\delta)\right)$. In the meanwhile, the SRHT OSE writes

$$\Omega = \sqrt{\frac{n}{\ell}} PHD, \tag{11}$$

where $P$ is made of $\ell$ rows of the identity matrix $I_n$ drawn uniformly at random (which corresponds to the sampling step), $H$ is the Walsh-Hadamard transform, and $D$ is a diagonal of random signs. Its application costs $O(n \log n)$ flops when implemented with standard Walsh-Hadamard Transform. Furthermore, it requires similar sampling size as Gaussian OSE, as it is a $(\epsilon, \delta, m)$ OSE if $\ell = \mathcal{O}\left(\epsilon^{-2}\left(m + \log(n - \delta)\right)\log(m - \delta)\right)$. A block version of this transform, which is suitable for distributed computing, is proposed in [1].

If $\Omega \in \mathbb{R}^{\ell\times n}$ is an $\epsilon$-embedding for $\mathrm{Range}(Q)$ where $Q \in \mathbb{R}^{n\times m}$, then in particular $Qx$ and $Qy \in \mathbb{R}^n$ are well-sketched, where $x$ and $y$ denote respectively the largest and smallest right-singular vectors of $Q$. In turn (see [12, 2]),

$$\begin{cases} \sigma_{\max}(Q) = \|Qx\| \le \dfrac{1}{\sqrt{1-\epsilon}} \|\Omega Q x\| \le \dfrac{1}{\sqrt{1-\epsilon}} \sigma_{\max}(\Omega Q) \\ \sigma_{\min}(Q) = \|Qy\| \ge \dfrac{1}{\sqrt{1+\epsilon}} \|\Omega Q y\| \ge \dfrac{1}{\sqrt{1+\epsilon}} \sigma_{\min}(\Omega Q) \end{cases}$$

yielding

$$\mathrm{Cond}(Q) \le \sqrt{\frac{1+\epsilon}{1-\epsilon}}\, \mathrm{Cond}(\Omega Q). \tag{12}$$

In particular, if $\Omega Q \in \mathbb{R}^{\ell\times m}$ has orthonormal columns, the condition number of $Q$ is simply bounded by the constant $\sqrt{1+\epsilon}/\sqrt{1-\epsilon}$. With $\epsilon = 1/2$, we get $\mathrm{Cond}(Q) \le \sqrt{3}$.

# 3 Randomized Householder process

In this section, we introduce a randomized version of the Householder reflector and the associated randomized Householder QR factorization for tall and skinny matrices. We then show how both the leftwise and rightwise compositions of these reflectors can be represented in a factored form, in the spirit of the original Householder process. Finally, we introduce the RHQR algorithm.

Let us set a matrix $\Omega \in \mathbb{R}^{\ell \times n}$ and not suppose anything about it for now, and let us denote $\Omega_q$ the last $n-q+1$ columns of $\Omega$. Let us for all $q \leq n$ set vector $u_q = 0_{q-1} \oslash v_q$, where $v_q \in \mathbb{R}^{n-q+1}$, and let us suppose that $u_q \notin \text{Ker}(\Omega)$, that is $v_q \notin \text{Ker}\left(\Omega_q\right)$. We define the randomized Householder reflector associated to vector $u_q$ and matrix $\Omega$ as

$$H^{\Omega}(u_q) := I_n - \frac{2}{\|\Omega u_q\|^2} u_q (\Omega u_q)^t \begin{bmatrix} 0_{\ell \times (q-1)} & \Omega_q \end{bmatrix} = I_n - \frac{2}{\|\overline{\Omega}_q u_q\|^2} u_q (\overline{\Omega}_q u_q)^t \overline{\Omega}_q. \tag{13}$$

It is useful to remark that, denoting for all $q \leq n$ the operator

$$A^{\Omega_q}(v_q) = I_{n-q+1} - \frac{2}{\|\Omega_q v_q\|^2} (\Omega_q v_q)^t \Omega_q, \quad A^{\Omega_q}(v_q) \in \mathbb{R}^{(n-q+1)\times(n-q+1)}, \tag{14}$$

then we can rewrite (13) as

$$H^{\Omega}(u_q) = \begin{bmatrix} I_{q-1} & \\ & A^{\Omega_q}(v_q) \end{bmatrix}. \tag{15}$$

It is straightforward to show that for all $z \in \mathbb{R}^n$ and $x \in \mathbb{R}^n$, we get $A^{\Omega}(z) \cdot A^{\Omega}(z) \cdot x = x$, hence $A^{\Omega}(z)$ is an involutary matrix, that is $A^{\Omega}(z)^2 = I_n$. By induction on $q \in \{1, \cdots m\}$, for any $v_q \in \mathbb{R}^{n-q+1}$ and $x_q \in \mathbb{R}^{n-q+1}$, the operator $A^{\Omega_q}(v_q) \in \mathbb{R}^{(n-q+1)\times(n-q+1)}$ is also an involutary matrix. In turn, for all $q \in \{1, \cdots, m\}$, for all $u_q = 0_{q-1} \oslash v_q \in \mathbb{R}^n$, the operator $H^{\Omega}(u_q) \in \mathbb{R}^{n \times n}$ is also an involutary matrix. Now let us make our first assumption on $\Omega$ and let us suppose that its columns are unit vectors of $\mathbb{R}^{\ell}$ (SRHT matrices are such matrices). In Proposition 3.1 we show that the randomized Householder reflector can be used to eliminate the coordinates of a vector below a given entry, without modifying the entries above.

**Proposition 3.1.** *Let $\Omega \in \mathbb{R}^{\ell \times n}$ be an arbitrary matrix with unit columns. Let $q \in \{1, \cdots, m\}$. Let $w \in \mathbb{R}^n$ be written as $w = x \oslash y$, where $x \in \mathbb{R}^{q-1}$ and $y \in \mathbb{R}^{n-q+1}$ denote respectively the first $q-1$ entries and the last $n-q+1$ entries of $w$. Suppose that $v_q = y - \|\Omega_q y\| e_1^q$ is not in the kernel of $\Omega_q$. Then $H^{\Omega}(u_q)$ is well defined and we get:*

$$H^{\Omega}(u_q) \cdot w = \begin{bmatrix} x \\ \|\Omega_q y\| e_1^q \end{bmatrix}.$$

*Proof.* Let $w \in \mathbb{R}^n$ and let us set $u_1 = v_1 = w - \|\Omega w\| e_1 \in \mathbb{R}^n$. Let us suppose that $u_1 \notin \text{Ker}(\Omega)$. We obtain

$$\langle \Omega u_1, \Omega w \rangle = \|\Omega w\|^2 - \|\Omega w\| \langle \Omega e_1, \Omega w \rangle.$$

We also have

$$\|\Omega u_1\|^2 = \|\Omega w\|^2 - 2\|\Omega w\| \langle \Omega w, \Omega e_1 \rangle + \|\Omega w\|^2 \|\Omega e_1\|^2.$$

As columns of $\Omega$ are supposed to be unit vectors, we get $\|\Omega e_1\| = 1$ yielding

$$\|\Omega u_1\|^2 = 2\|\Omega w\|^2 - 2\|\Omega w\| \langle \Omega w, \Omega e_1 \rangle$$

which leads to

$$\frac{2}{\|\Omega u_1\|^2} \langle \Omega u_1, \Omega w \rangle = 1.$$

Overall, we get

$$A^{\Omega}(u_1) \cdot w = \|\Omega w\| e_1.$$

The result follows from induction on $q \in \{1, \cdots m\}$. □

It is straightforward to check that, as in the deterministic Householder process, we can switch the sign in the computation of the Householder vector to avoid cancellations. We can proceed similarly to the Householder QR factorization of a matrix $W \in \mathbb{R}^{n\times m}$, where $m \ll n$, and produce the factorization:

$$H^{\Omega}(u_m) \cdot H^{\Omega}(u_{m-1}) \cdots H^{\Omega}(u_1) W = \begin{bmatrix} R \\ 0_{(n-m)\times m} \end{bmatrix},$$

where $R \in \mathbb{R}^{m\times m}$ is upper triangular. Since these operators are involutary, this is equivalent to:

$$W = H^{\Omega}(u_1) \cdot H^{\Omega}(u_2) \cdots H^{\Omega}(u_m) \begin{bmatrix} R \\ 0_{(n-m)\times m} \end{bmatrix},$$

or the thin QR factorization

$$W = Q_m R, \text{ where } Q_m = H^{\Omega}(u_1) \cdot H^{\Omega}(u_2) \cdots H^{\Omega}(u_m) I_{1:m} \in \mathbb{R}^{n\times m}.$$

We remark that the randomized Householder reflector is invariant under scaling of the randomized Householder vector. Indeed, if $\lambda \in \mathbb{R}^*$,

$$H^{\Omega}(\lambda u_q) = I_n - \frac{2}{\lambda^2 \|\overline{\Omega}_q u_q\|^2} \lambda^2 u_q (\overline{\Omega}_q u_q)^t \overline{\Omega}_q = H^{\Omega}(u_q).$$

We present in Propositions 3.2 and 3.4 formulas for the factored forms of rightwise and leftwise compositions of randomized Householder reflectors. For simplicity, these formulas assume that the randomized Householder vectors are scaled such that $\|\Omega u_q\|^2 = \|\overline{\Omega}_q u_q\|^2 = \|\Omega_q v_q\|^2 = 2$, for all $1 \le q \le m$, simplifying (13) into:

$$H^{\Omega}(u_q) = I_n - u_q (\overline{\Omega}_q u_q)^t \overline{\Omega}_q. \tag{16}$$

**Proposition 3.2.** *Assume vectors $u_1, \cdots u_m \in \mathbb{R}^n$ are scaled such that their sketched norm is $\sqrt{2}$. Define by induction the matrix $T_q$ for all $1 \le q \le m$ as*

$$\begin{aligned} &T_1 \in \mathbb{R}^{1\times 1}, \quad T_1 = [1] \\ &\forall 1 \le q \le m-1, \quad T_{q+1} = \begin{bmatrix} T_q & -T_q (\Omega U_q)^t \Omega u_{q+1} \\ 0_{1\times q} & 1 \end{bmatrix} \end{aligned} \tag{17}$$

*Then for all $1 \le q \le m$, we have the factored form:*

$$\mathcal{H}_q^{-1} := H^{\Omega}(u_1) \cdots H^{\Omega}(u_q) = I - U_q T_q \operatorname{ut}\left((\Omega U_q)^t \Omega\right). \tag{18}$$

*Proof.* We proceed by induction on $m$. For the case $m = 2$, multiplying $H^{\Omega}(u_1)$ and $H^{\Omega}(u_2)$, we observe that

$$H^{\Omega}(u_1) H^{\Omega}(u_2) = I_n - U_2 T_2 \operatorname{ut}\left((\Omega U_2)^t \Omega\right),$$

where ut denotes the upper triangular part and

$$T_2 = \begin{bmatrix} 1 & -\langle \Omega u_1, \Omega u_2 \rangle \\ 0 & 1 \end{bmatrix}.$$

Let us then suppose that

$$H^{\Omega}(u_1) \cdots H^{\Omega}(u_q) = I - U_q T_q \operatorname{ut}\left((\Omega U_q)^t \Omega\right),$$

for some matrix $T_q \in \mathbb{R}^{q\times q}$. Multiplying by $H^{\Omega}(u_{q+1})$ on the right-side, we obtain

$$\begin{aligned} &H^{\Omega}(u_1) \cdots H^{\Omega}(u_q) H^{\Omega}(u_{q+1}) = \\ &I_n - u_{q+1} (\Omega u_{q+1})^t \begin{bmatrix} 0_{\ell\times q} & \Omega_{q+1} \end{bmatrix} - U_q T_q \operatorname{ut}\left((\Omega U_q)^t \Omega\right) + U_q T_q \operatorname{ut}\left((\Omega U_q)^t \Omega\right) u_{q+1} (\Omega u_{q+1})^t \begin{bmatrix} 0_{\ell\times q} & \Omega_{q+1} \end{bmatrix}. \end{aligned}$$

We first observe that

$$(\Omega u_{q+1})^t \begin{bmatrix} 0_q & \Omega_{q+1} \end{bmatrix} = \begin{bmatrix} 0_q & (\Omega u_{q+1})^t \Omega_{q+1} \end{bmatrix}.$$

Since the first $q$ coordinates of $u_{q+1}$ are zero, we also have:

$$\operatorname{ut}\left((\Omega U_q)^t \Omega\right) u_{q+1} = (\Omega U_q)^t \Omega u_{q+1},$$

so that the total composition writes

$$I_n - u_{q+1} \begin{bmatrix} 0_{\ell\times q} & (\Omega u_{q+1})^t \Omega_{q+1} \end{bmatrix} - U_q T_q \begin{bmatrix} (\Omega u_1)^t \Omega & & & \\ 0_{1\times 1} & (\Omega u_2)^t \Omega_2 & & \\ \vdots & & & \\ 0_{1\times(q-1)} & & (\Omega u_q)^t \Omega_q & \end{bmatrix} + U_q \left(T_q (\Omega U_q)^t \Omega u_{q+1}\right) \begin{bmatrix} 0_{1\times q} & (\Omega u_{q+1})^t \Omega_{q+1} \end{bmatrix}, \tag{19}$$

which can be factored as

$$H^\Omega(u_1)\cdots H^\Omega(u_q) H^\Omega(u_{q+1}) = I - U_{q+1} \begin{bmatrix} T_q & -T_q (\Omega U_q)^t \Omega u_{q+1} \\ 0_{1\times q} & 1 \end{bmatrix} \operatorname{ut}\left((\Omega U_{q+1})^t \Omega\right).$$

The proposition follows by induction. □

Applying the Woodburry-Morrison formula to the factored form as in [9], we derive the following proposition Since $T_q$ is triangular with unit diagonal, we deduce the following formula

**Proposition 3.3.** *With the previous notations, we have*

$$T_q^{-1} + T_q^{-t} = (\Omega U_q)^t \Omega U_q \implies T_q = \left[\operatorname{sut}\left((\Omega U_q)^t \Omega U_q\right) + I_q\right]^{-1}. \tag{20}$$

By following the same reasoning, we summarize in the following proposition the factored form of the leftwise compositions of randomized Householder reflectors:

**Proposition 3.4.** *Assume vectors $u_1, \cdots u_m \in \mathbb{R}^n$ are scaled such that their sketched norm is $\sqrt{2}$. Define by induction the matrix $S_q$ for all $1 \le q \le m$ as*

$$\begin{aligned} & S_1 \in \mathbb{R}^{1\times 1}, \quad S_1 = [1] \\ & \forall 1 \le q \le m-1, \quad S_{q+1} = \begin{bmatrix} S_q & 0 \\ -(\Omega u_{q+1})^t \Omega_{q+1} U_q S_q & 1 \end{bmatrix}. \end{aligned} \tag{21}$$

*Then for all $1 \le q \le m$, we have the factored form*

$$\mathcal{H}_q = H^\Omega(u_q)\cdots H^\Omega(u_1) = I - U_q S_q \operatorname{ut}\left((\Omega U_q)^t \Omega\right). \tag{22}$$

Algorithm 1 describes the randomized Householder QR (RHQR) factorization of a matrix $W \in \mathbb{R}^{n\times m}$, with $m \ll n$, that relies on the previous derivations. We consider here that the vectors become available as in a Krylov subspace solver and we describe a left-looking version of the algorithm. In the case when the vectors are available at once, a right-looking version can be derived, that does not need to construct the factored form before the end of the factorization.

This algorithm stores $R$ in place of the upper triangular part of $W$. The matrix $U_m$ can be stored in the lower triangular part of $W$ (the diagonal of $U_m$ can be stored in a vector). The storage cost of $\Omega U_m, T_m, S_m$ is negligible compared to that of $W$. The output can be used to produce the thin factor $Q_m = \mathcal{H}_m^{-1} I_{1:m}$. Lines 3 and 7 output the result of the application of $H^\Omega(u_1)$ (resp. $H^\Omega(u_q)$) to $w_1$ (resp.

**Algorithm 1:** Randomized-Householder QR (RHQR), left-looking

**Input: Given**: Matrix $W \in \mathbb{R}^{n\times m}$, matrix $\Omega \in \mathbb{R}^{\ell\times n}$, $m < l \ll n$
1 Compute scaled $u_1$ and $\Omega u_1$ such that $H^{\Omega}(u_1)\cdot w_1 = \|\Omega w_1\| e_1$
2 Set $T_1 = S_1 = [1]\,,\ U_1 = [u_1],\ \ \Omega U_1 = [\Omega u_1]$
3 Set $(W)_{1,1} = \|\Omega w_1\|$
4 **for** $q = 2 : m$ **do**
5 $\quad w_q \leftarrow \left[I_n - U_{q-1}S_{q-1}\mathrm{ut}\left((\Omega U_{q-1})^t\Omega\right)\right]\cdot w_q$
6 $\quad$ Compute scaled $u_q$ and $\Omega u_q$ such that $H^{\Omega}(u_q)w_q = w_{q,1} \oslash (\|\Omega_q w_{q,2}\| e_1^q)$
7 $\quad$ Set $(W)_{q,q} = \|\Omega_q w_{q,2}\|$
8 $\quad$ Update $T_q, S_q$ according to (17) and (21), $\ U_q = \left[U_{q-1} \mid u_q\right],\ \ \Omega U_q = \left[\Omega U_{q-1} \mid \Omega u_q\right]$
9 $Q_m \leftarrow \left[I_n - U_m T_m \mathrm{ut}\left((\Omega U_m)^t\Omega\right)\right]\cdot I_{1:m}$ (optional)
10 Return $U_m, \Omega U_m, T_m, S_m$, $Q = Q_m$ (optional) and $R = (W_{i,j})_{1\le i\le j\le m}$

$\mathcal{H}_{q-1}\cdot w_q)$. As to Line 5, the main challenge is the application of ut $\left((\Omega U_q)^t\Omega\right)$, which can be done in several ways. For example, one can first compute $(\Omega U_q)^t\Omega x$, then substract the vector

$$\begin{bmatrix} 0 \\ \langle \Omega u_2, x_1\Omega_{1:1}\rangle \\ \langle \Omega u_3, \Omega_{1:2}x_{1:2}\rangle \\ \vdots \\ \langle \Omega u_q, \Omega_{1:q}x_{1:q}\rangle \end{bmatrix}.$$

If $\Omega$ is not stored as a dense matrix but can only be applied to a vector (e.g SRHT OSE), one can also compute explicitly along the iterations

$$\mathrm{slt}\left((\Omega U_q)^t\Omega\right) = \begin{bmatrix} 0 & 0 & 0 & \dots & & \dots & 0 \\ \langle \Omega u_2, \Omega e_1\rangle & 0 & 0 & \dots & & \dots & 0 \\ \langle \Omega u_3, \Omega e_1\rangle & \langle \Omega u_3, \Omega e_2\rangle & 0 & \dots & & & \\ \vdots & & \ddots & & & & \\ \langle \Omega u_q, \Omega e_1\rangle & \langle \Omega u_q, \Omega e_2\rangle & \dots & \langle \Omega u_q, \Omega e_{q-1}\rangle & 0 & \dots & 0 \end{bmatrix}. \tag{23}$$

Then, the computation of ut $\left((\Omega U_q)^t\Omega\right)x$ is replaced by that of $(\Omega U_q)^t\Omega x$ followed by the substraction of slt $\left((\Omega U_q)^t\Omega\right)\,x$. The same approach can be used to compute the update of $S_q$.

We finally detail in Algorithm 2 a block version of Algorithm 1. The matrix $W \in \mathbb{R}^{n\times(mb)}$ is now considered as formed by vertical slices $W_1,\cdots W_b \in \mathbb{R}^{n\times b}$. Algorithm 1 is then applied consecutively to each blocks. For each $W_j$, we denote $U_{m,j}, \Omega U_{m,j}, T_{m,j}, S_{m,j}$ the output of Algorithm 1 applied to $W_j$. Just as in Algorithm 1, the upper triangular par of $W$ is erased and replaced by the final $R$ factor, while $\left[U_{m,1} \mid \cdots \mid U_{m,b}\right]$ can be stored in place of the lower triangular part of $W$ (again, one can store the diagonal in a vector). At step $j$, the next application of Algorithm 1 must begin at the $m(j-1)+1$-th row of the block $W_j$.

# 4 Numerical experiments

In this section, we first detail the matrices of our test set. We then test the stability and accuracy of RHQR on difficult examples. Finally, we experiment an adaptation of Householder Arnoldi to the randomized Householder reflectors.

**Algorithm 2:** Block Randomized-Householder QR (block RHQR)

**Input: Given**: Matrix $W = [W_1 \cdots W_b] \in \mathbb{R}^{n\times mb}$, matrix $\Omega \in \mathbb{R}^{\ell\times n}$, $mb < l \ll n$

**Output:** $Q$ in factored form, $R$

Apply Algorithm 1 to the block $W_1$ and output $U_{m,1}, \Omega U_{m,1}, T_{m,1}, S_{m,1}$

**for** $j = 2 : b$ **do**

  Apply $\left[I_n - U_{m,j-1}S_{m,j-1}\mathrm{ut}\left((\Omega U_{m,j-1})^t\Omega\right)\right]\cdots\left[I_n - U_{m,1}T_{m,1}\mathrm{ut}\left((\Omega U_{m,1})^t\Omega\right)\right]$ to the block $W_j$

  Apply Algorithm 1 to $W_j$, starting from $m(j-1)$-th row and output $U_{m,j}, \Omega U_{m,j}, T_{m,j}, S_{m,j}$

$Q \leftarrow \left[I_n - U_{m,1}S_{m,1}\mathrm{ut}\left((\Omega U_{m,1})^t\Omega\right)\right]\cdots\left[I_n - U_{m,b}T_{m,b}\mathrm{ut}\left((\Omega U_{m,b})^t\Omega\right)\right] I_{1:mb}$ (optional)

Return $\left\{U_{m,j}, \Omega U_{m,j}, T_{m,j}, S_{m,j}\right\}_{1\le j\le b}$, $(W_{i,j})_{1\le i,j\le mb}$

The first matrix is formed by uniformly discretized parametric functions, also used in [2], that we denote $S \in \mathbb{R}^{n\times m}$. For all floating numbers $0 \le x, \mu \le 1$, the function is defined as

$$f(x,\mu) = \frac{\sin\left(10(\mu + x)\right)}{\cos\left(100(\mu - x)\right) + 1.1}$$

and the associated matrix is

$$S \in \mathbb{R}^{50000\times 600}, \quad (S)_{i,j} = f\left(\frac{i-1}{50000}, \frac{j-1}{600}\right). \tag{24}$$

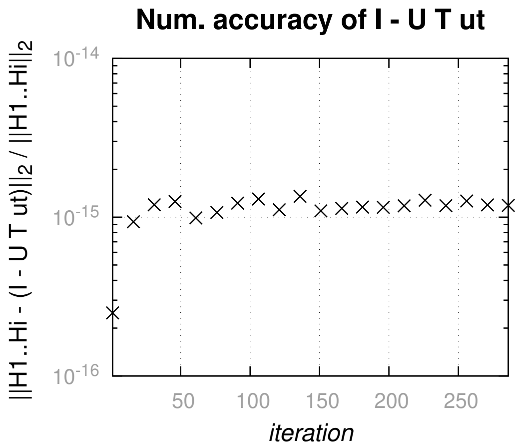
The condition number of $S \in \mathbb{R}^{50000\times 600}$ in double precision is displayed in Figure 2a and in single precision in Figure 3a. The second matrix $S_b \in \mathbb{R}^{50000\times 2000}$ is defined similarly. A third construction is used, based on a rank-one perturbation of a random matrix. We select a large positive real number $K$. We draw a random matrix and denote it $P(K) \in \mathbb{R}^{50000\times 600}$, normalize its columns and select $p$ as the first column of $P(K)$. Then, until the condition number of $P(K)$ is larger than $K$, we repeat the following steps:

$$\begin{aligned} &1.\ P(K) \quad \leftarrow \quad P(K) \quad + \quad p\cdot\begin{pmatrix}1 & 1 & \cdots & 1\end{pmatrix} \\ &2.\ \text{Normalize } \ P(K) \end{aligned} \tag{25}$$

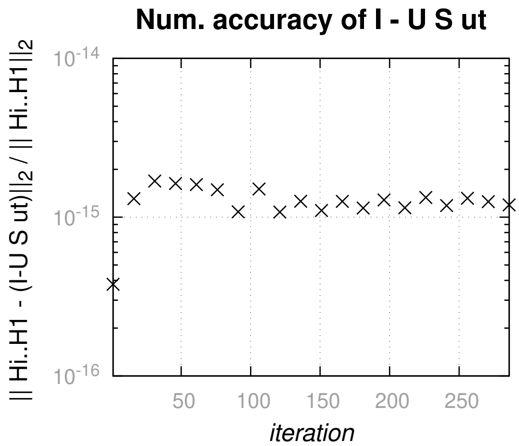
We select two matrices from SuiteSparse [6] to test the Arnoldi process based on randomized Housholder QR. The first, *Poli4*, is an unsymmetric operator arising from solving an economic problem. Its condition number is approximately 700, which qualifies it as a rather easy problem. The second problem is the matrix *SiO* from [6]. It is symmetric but not positive. Its condition number is approximately 3500. These matrices are summarized in Table 1.

| Name | Size | Cond. | Origin |
|---|---|---|---|
| $S$ | $50000 \times 600$ | $\infty$ | synthetic functions |
| $S_b$ | $50000 \times 2000$ | $\infty$ | synthetic functions |
| $P(K)$ | $50000 \times 600$ | $\ge K$ | rank 1 perturbation of random matrix |
| El3D | 18800 | $\approx 10^{26}$ | near incompressible regime 3D elasticity |

Table 1: Set of matrices used in experiments




(a) Operator $\mathcal{H}_m^{-1}, \ell = 2000$




(b) Operator $\mathcal{H}_m, \ell = 2000$

Figure 1: Relative spectral error between the applications of factored form $\mathcal{H}_m^{-1}, \mathcal{H}_m$ and the explicit compositions of all reflectors.

## 4.1 Factored form

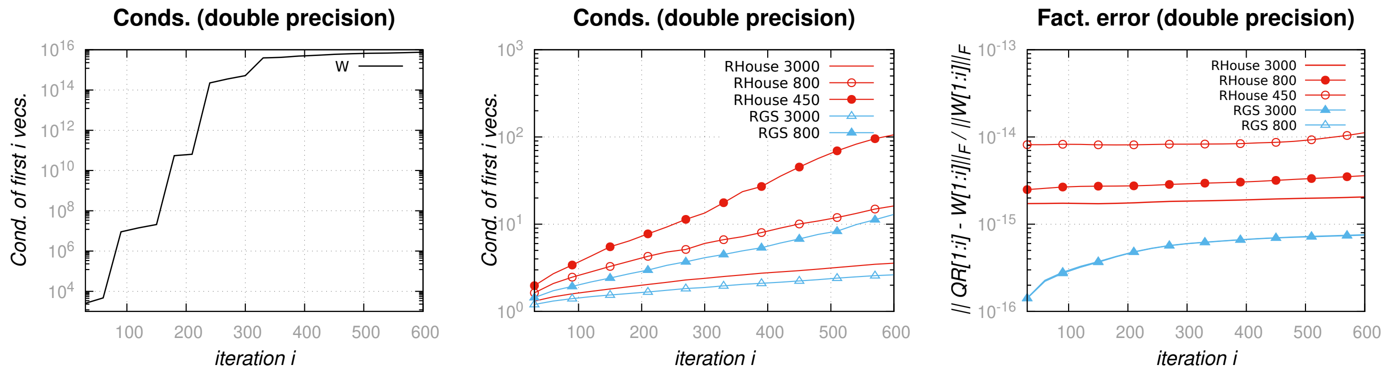
In this section we investigate the accuracy of the factored form, with respect to the expicit compositions of randomized Householder reflectors. For all factored forms, we display the spectral norm of the error, divided by that of the explicit compositions. These norms are approximated through Arnoldi iterations. In Figure 1 we illustrate the evolution of this relative spectral error as the number of compositions increase. The operators are randomly generated randomized Householder reflectors of $\mathbb{R}^{50000\times 50000}$. Both Figures 1a and 1b show that this error remains very close to $10^{-15}$.

## 4.2 Randomized Householder QR factorization

We discuss in this section the accuracy of the randomized Householder QR factorization as described in Algorithm 1. We first use SRHT OSE $\mathbf{\Omega}$, as its columns are unit vectors of $\mathbb{R}^{\ell}$. We use in these tests the set of synthetic functions described in (24). We consider 600 discretization points for $\mu$ and 50000 discretization points for $x$.

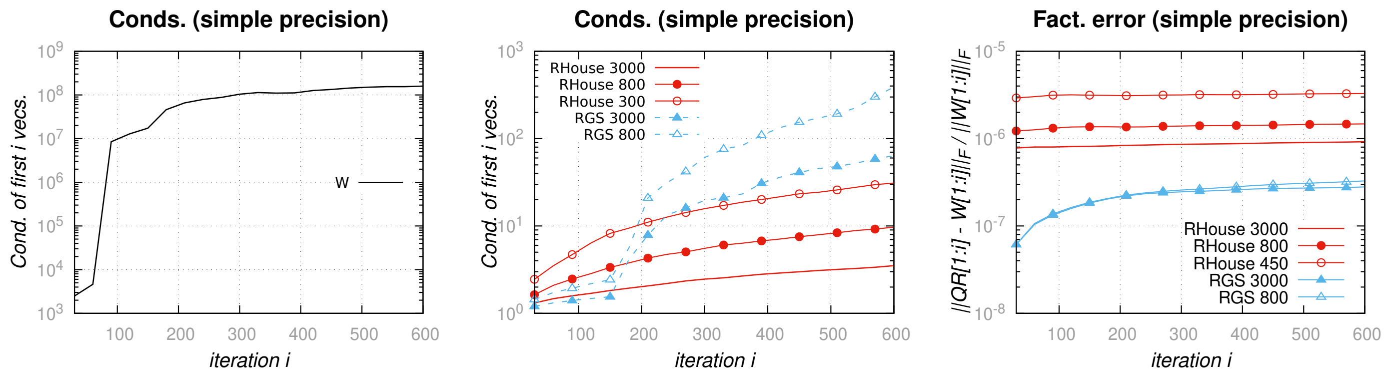
Figure 2 displays the accuracy obtained in double precision for both RHQR and RGS, in terms of condition number of the computed basis and the accuracy of the obtained factorization. Figure 2a shows the condition number of $W$ as the number of its vectors increases from 1 to 600. The condition number grows exponentially and the matrix becomes numerically singular when the 300-th vector is reached. Figure 2b displays in red the condition number of the basis obtained through RHQR from Algorithm 1, and in blue that obtained by RGS for reference. We observe that for similar sampling size, both algorithms produce a well conditioned basis, with a slight advantage for RGS. Figure 2c shows the relative Frobenius error of the QR factorization obtained by RHQR (in red) and RGS (in blue). Both algorithms produce accurate factorizations, with a slight advantage for RGS, but that becomes negligible when $m$ grows. Those results also show that Algorithm 1 produces an accurate factorization even with a sampling size inferior to the number of vectors to be sketched ($\ell = 450,\ m = 600$).

Figure 3 displays the condition number of the computed basis as well as the accuracy of the factorization produced by Algorithm 1 in single precision. Figure 3a shows the condition number of the matrix $W$, which becomes numerically singular once its 200-th vector is reached. Figure 3b shows the condition number of the Q factor produced by Algorithm 1 in red, and that of RGS in blue. We see that, while RGS produces a slightly better conditioned basis during the first iterations, its behavior changes as the matrix $W$ becomes numerically singular. The increase of the condition number of the basis obtained by RHQR from Algorithm 1 continues to grow very slowly, and produces a better conditioned basis




(a) Cond. of initial matrix (b) Cond. of basis built (c) Factorization error

Figure 2: Randomized Householder QR in double precision (RHouse in the graphs) and comparison with RGS.




(a) Cond. of initial built (b) Cond. of the Q factor, W = QR (c) Factorization error

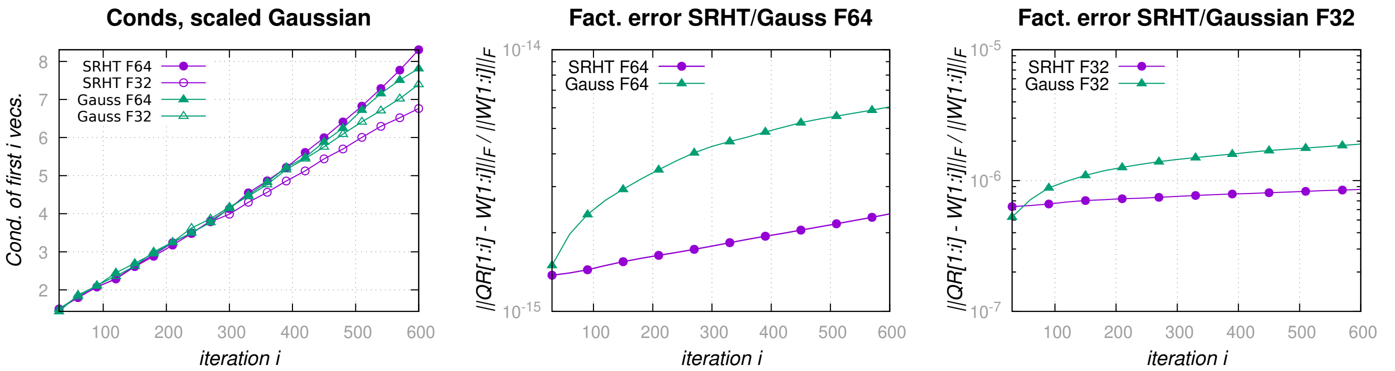
Figure 3: Randomized Householder QR in single precision (RHouse in the graphs) and comparison with RGS.

than RGS. More precisely, Algorithm 1 with a sampling size inferior to the number of columns of $W$ ($\ell = 300,\ m = 600$) outperforms RGS with a larger sampling size. In Figure 3c we show the relative Frobenius error of the factorization. We see that both algorithms perform well, with a slight advantage for RGS.

Lastly, we compare in Figure 4 the results obtained when the sketching is performed using SRHT OSE with respect to the ones obtained with a modified Gaussian OSE. The Gaussian OSE cannot be straightforwardly used in Algorithm 1, as its columns are not unit vectors of $\mathbb{R}^{\ell}$. However, our experiments show that it is possible to use Gaussian OSE if we explicitely scale its columns to be unit vectors of $\mathbb{R}^{\ell}$. Figure 4a displays the condition number of the basis produced by SRHT OSE and modified Gaussian OSE, in double (plain lines) and simple (dotted lines) precision. In double precision, the two techniques have similar behavior, with a slight advantage for modified Gaussian OSE at the end of the iterations. In simple precision, both techniques have good accuracy, but SRHT OSE seems to be slightly better for most of the iterations. Figure 4b shows the Frobenius norm of the relative error of the factorization in double precision. Both techniques lead to small errors, with a clearer advantage for SRHT OSE. Figure 4c displays the same errors in single precision, and we observe a similar behavior.

## 4.3 Block Randomized Householder QR

In this section we discuss the numerical accuracy as well as the condition number of the $Q$ factor produced by Algorithm 2. We use the same set of synthetic functions, this time on matrix $S_b$ with

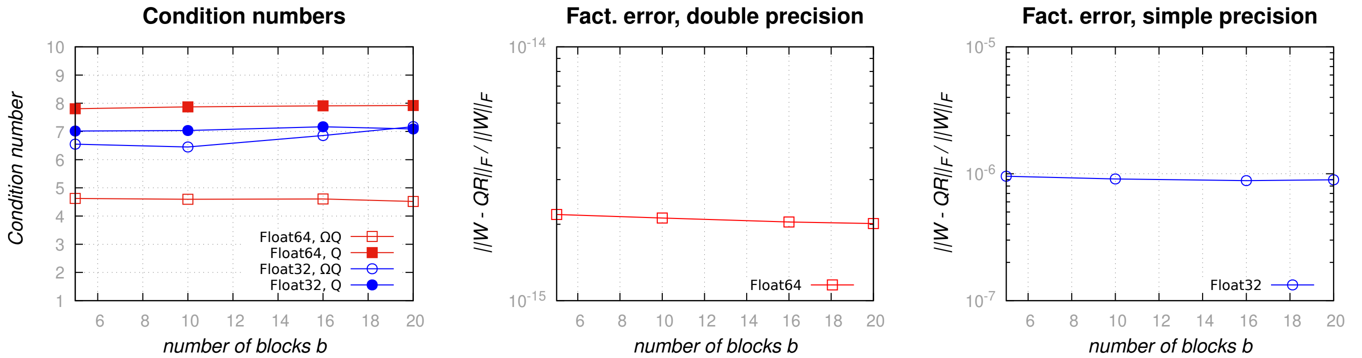



(a) Cond. of basis built

(b) Factorization error in double precision

(c) Factorization error in single precision

Figure 4: Comparison of modified Gaussian and SRHT OSE.




(a) Condition number of $Q$ as the number of blocks $b$ increases

(b) Factorization relative error $\|QR - W\|_2 \cdot \|W\|_2^{-1}$ in double precision as the number of blocks $b$ increases

(c) Same as (b) in simple precision

Figure 5: Condition number of the basis and accuracy of the factorization produced by block RHQR on matrix $S_b \in \mathbb{R}^{50000\times 2000}$, with sampling size $\ell = 4000$

$m = 2000$ columns. We consider different block sizes, $b \in \{5, 10, 16, 20\}$. The results are presented in Figure 5. Figure 5a displays the condition number of the $Q$ and sketched $\Omega Q$ factors obtained for different block sizes in both precisions. The condition numbers of factored forms don't seem to multiply. Furthermore, we see that the factorization is overall performing well with $\ell \approx 2m$ ($\mathrm{Cond}(Q) \approx 8$). In Figures 5b and 5c, we see that the factorization relative error is small in both precisions.

## 4.4 Application to Arnoldi

The Householder Arnoldi procedure [10, Algorithm 6.3, p163] can be easily adapted to use Algorithm 1, yielding Algorithm 3 in which the notations for vectors $(w_j)_{1\le j\le m}$ are also used for vectors $z_j$ (namely, $z_{j,1}$ denote the first $j-1$ coordinates of $z_j$ and $z_{j,2}$ the last $n-j+1$ coordinates of $z_j$).

**Proposition 4.1.** *Algorithm 3 produces $V_m \in \mathbb{R}^{n\times m}$, $V_{m+1} \in \mathbb{R}^{n\times(m+1)}$, $H_{m+1,m} \in \mathbb{R}^{(m+1)\times m}$ such that $AV_m = V_{m+1}H_{m+1,m}$.*

*Proof.* The proof is almost identical to that found in [10]. We recall that in this work, $\mathcal{H}_m^{-1}$ denotes the rightwise composition of reflectors $H^\Omega(u_1)\cdots H^\Omega(u_m)$. The second difference is that our operator $\mathcal{H}_m^{-1}$ is not unitary, although we have access to $\mathcal{H}_m$ in factored form. By the definition of the vector $z_{j+1}$

**Algorithm 3:** Randomized Householder Arnoldi

**Input: Given**: Matrix $A \in \mathbb{R}^{n \times n}$, $x_0, b \in \mathbb{R}^n$, $m \in \mathbb{N}^*$, matrix $\Omega \in \mathbb{R}^{\ell \times n}$, $m < l \ll n$,

1 Set $z_1 = b - Ax_0$

2 **for** $j = 1 : m + 1$ **do**

3     Compute $u_j \in \mathbb{R}^n$ such that $H^\Omega(u_j) \cdot z_j = (z_{j,1} \oslash (\|\Omega_j z_{j,2}\| e_1^j)$

4     Set $h_{j-1} = H^\Omega(u_j) z_j, \quad h_{j-1} \in \mathbb{R}^n$

5     Set $v_j = H^\Omega(u_1) \cdots H^\Omega(u_j) e_j$

6     **if** $j \leq m$ **then**

7         Set $z_{j+1} = H^\Omega(u_j) \cdots H^\Omega(u_1) A v_j$

8 Set $H_{m+1,m}$ as the first $m+1$ rows of $[h_1 \; h_2 \ldots h_m]$, discard $h_0$.

9 Return $V_{m+1}, H_{m+1,m}$

at Line 7, once the following randomized Householder reflector is applied at Line 4, we obtain:

$$h_j = H^\Omega(u_{j+1}) H^\Omega(u_j) \cdots H^\Omega(u_1) A v_j. \tag{26}$$

Since the coordinates $j+2, j+3 \cdots n$ of $h_j$ are zero, it is invariant under the remaining reflectors:

$$h_j = H^\Omega(u_m) \cdots H^\Omega(u_{j+2}) h_j = H^\Omega(u_m) \cdots H^\Omega(u_1) A v_j. \tag{27}$$

This relation being true for all $j$, we obtain the factorization:

$$H^\Omega(u_m) \cdots H^\Omega(u_1) \begin{bmatrix} r_0 & Av_1 & \ldots & Av_m \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & \cdots & h_m \end{bmatrix}.$$

Multiplying on the left by the reflectors in reverse order $H^\Omega(u_1) \cdots H^\Omega(u_m)$, we obtain:

$$\begin{bmatrix} r_0 & Av_1 & \ldots & Av_m \end{bmatrix} = H^\Omega(u_1) \cdots H^\Omega(u_m) \begin{bmatrix} h_0 & h_1 & \cdots & h_m \end{bmatrix} = \mathcal{H}_m^{-1} \begin{bmatrix} h_0 & h_1 & \cdots & h_m \end{bmatrix}.$$

From relation (27),

$$Av_j = \mathcal{H}_{j+1} h_j = \sum_{i=1}^{j+1} h_{i,j+1} \mathcal{H}_{j+1} e_i.$$

In addition, the reflectors $H^\Omega(u_{i+1}), \cdots H^\Omega(u_j)$ let $e_i$ invariant, yielding

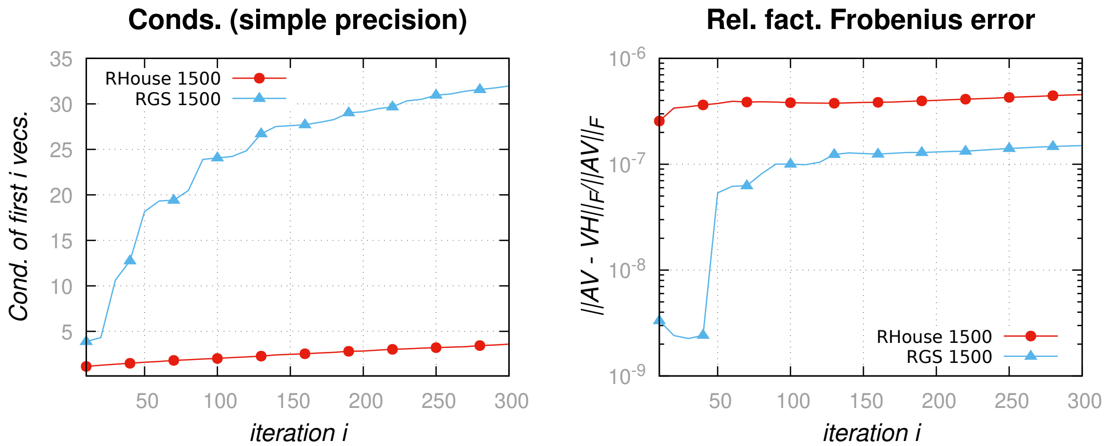
$$Av_j = \sum_{i=1}^{j+1} h_{i,j+1} v_i,$$

which concludes the proof. □

We display in Figure 6 the condition number of the Arnoldi basis built and the factorization error of Algorithm 3 on a matrix obtained from finite element methods applied to a nearly incompressible 3D elasticity PDE. It is a very difficult case, on which we can see the stability of Algorithm 3.

# 5 Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 810367).




(a) Condition number of Arnoldi basis in simple precision

(b) Relative factorization Frobenius error in simple precision

Figure 6: Performance of Algorithm 3 on El3D in simple precision

# References


[1] Oleg Balabanov, Matthias Beaupere, Laura Grigori, and Victor Lederer. “Block subsampled randomized Hadamard transform for low-rank approximation on distributed architectures”. In: *Proceedings of the 40th International Conference on Machine Learning*. https://doi.org/10.48550/arxiv.2210.1 PMLR 202. 2023.

[2] Oleg Balabanov and Laura Grigori. “Randomized Gram–Schmidt Process with Application to GMRES”. In: *SIAM Journal on Scientific Computing* 44.3 (2022), A1450–A1474.

[3] Jesse L Barlow. “Block Modified Gram–Schmidt Algorithms and Their Analysis”. In: *SIAM Journal on Matrix Analysis and Applications* 40.4 (2019), pp. 1257–1290.

[4] Åke Björck and Christopher C Paige. “Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm”. In: *SIAM journal on matrix analysis and applications* 13.1 (1992), pp. 176–190.

[5] Erin Carson, Kathryn Lund, Miroslav Rozložník, and Stephen Thomas. “Block Gram-Schmidt algorithms and their stability properties”. In: *Linear Algebra and its Applications* 638 (2022), pp. 150–195.

[6] Timothy A Davis and Yifan Hu. “The University of Florida sparse matrix collection”. In: *ACM Transactions on Mathematical Software (TOMS)* 38.1 (2011), pp. 1–25.

[7] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[8] Christopher C Paige. “A useful form of unitary matrix obtained from any sequence of unit 2-norm n-vectors”. In: *SIAM Journal on Matrix Analysis and Applications* 31.2 (2009), pp. 565–583.

[9] Chiara Puglisi. “Modification of the Householder method based on the compact WY representation”. In: *SIAM Journal on Scientific and Statistical Computing* 13.3 (1992), pp. 723–726.

[10] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[11] JH Wilkinson. “The algebraic eigenvalue problem”. In: *Handbook for Automatic Computation, Volume II, Linear Algebra*. Springer-Verlag New York, 1971.

[12] David P. Woodruff. “Sketching as a Tool for Numerical Linear Algebra”. In: *CoRR* abs/1411.4357 (2014). arXiv: `1411.4357`. URL: `http://arxiv.org/abs/1411.4357`.