ORTHOGONALIZATION VIA POLAR DECOMPOSITION

ZHENGBO ZHOU*

March 10, 2023

1 Introduction

1.1 Preliminaries

The norm, throughout this report, will be unitarily invariant norm. Namely, for any $A \in \mathbb{C}^{n\times n}$, $\|QAP\| = \|A\|$ for any two unitary matrices $Q, P \in \mathbb{C}^{n\times n}$. One important property of the unitarily invariant norm is that they are self-adjoint [7, 2013, pp. 357-358], i.e. $\|A^*\| = \|A\|$. This can be proved by considering the SVD (Theorem 2.1) of A.

We will consider two precisions: high precision, u_{high}, and low precision, u_{low}, such that $u_{\text{low}} \gg u_{\text{high}}$. For example, they can be interpreted as single precision (low) and double precision (high), where $u_{\text{single}} \approx 6 \times 10^{-8}$ and $u_{\text{high}} \approx 1.1 \times 10^{-16}$ by IEEE standard [8, 2019]. The error analysis for the eigendecomposition in arbitrary precision is readily presented by [1, Sec. 4.7.1].

Theorem 1.1. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. The computed eigendecomposition $A = \widetilde{Z}\widetilde{\Lambda}\widetilde{Z}^T$ via any LAPACK routine in precision u is nearly the exact eigendecomposition of $A + E$. Namely, $A + E = (\widetilde{Z} + \delta\widetilde{Z})\widetilde{\Lambda}(\widetilde{Z} + \delta\widetilde{Z})^T$ is a true eigendecomposition so that $\widetilde{Z} + \delta\widetilde{Z}$ is orthogonal, where $\|E\|/\|A\| \le p(n)u$ and $\|\delta\widetilde{Z}\| \le p(n)u$. Here $p(n)$ is a modestly growing function of n.*

1.2 Orthogonalization via QR

In 2022, Zhang and Bai [10, 2022] propose a mixed precision Jacobi algorithm that can be described by Algorithm 1.

Algorithm 1 A summary of the mixed precision Jacobi algorithm for symmetric eigenvalue problem proposed by Zhang and Bai [10, 2022, Alg. 4.1].

```
Require: A symmetric matrix $A \in \mathbb{R}^{n\times n}$.
Ensure: An orthogonal matrix $P \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\Delta$.
1: Compute the eigendecomposition of $A$ in low precision, $A = P_{\text{low}}\Delta_{\text{low}}P_{\text{low}}^T$.
2: Compute an QR factorization of $P_{\text{low}} = Q_P R_P$.        → Modified Gram-Schmidt
3: Precondition the matrix $A$ by $A_{\text{cond}} = Q_P^T A Q_P$.
4: Apply Jacobi algorithm on $A_{\text{cond}}$.
```

*Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (zhengbo.zhou@postgrad.manchester.ac.uk).

From the second step, the paper orthogonalizes the almost orthogonal matrix P_{low} using the modified Gram-Schmidt QR factorization. This method can compute the QR factors with a small residual. Moreover, the computed orthogonal factor Q_P has a bounded departure from orthonormality, i.e. $\|Q_P^TQ_P - I\|_2$ is bounded by some multiple of $u_{\text{high}}\kappa_2(P_{\text{low}})$ [3, 2013, Sec. 5.2.9]. The cost of the modified Gram-Schmidt algorithm is $2n^3$ for $n \times n$ matrix [3, 2013, Alg. 5.2.6]. Moreover, the input matrix for the MGS algorithm, P_{low}, is a computed eigenvector matrix in precision u_{low}. Notice that, although the MGS QR decomposition may produce a large deviation from orthogonality, but due to the excellent condition of P_{low}, the bound will be roughly a multiple of u_{high}.

Theorem 1.2. *Let P_{low} be an computed eigenvector matrix via any LAPACK routine in precision u_{low}. Then, we have the upper bound for the 2-norm of P_{low},*

$$\|P_{\text{low}}\|_2 \leq p(n)u_{\text{low}} + \sqrt{1 + 2p(n)^2u_{\text{low}}^2}$$

where $p(n)$ is a modestly increasing function of n.

Proof. By Theorem 1.1, we have

$$(P_{\text{low}} + \delta P_{\text{low}})^T(P_{\text{low}} + \delta P_{\text{low}}) = I, \qquad \|\delta P_{\text{low}}\| \leq p(n)u_{\text{low}},$$

where $p(n)$ is a modestly growing function in n. Multiply out we have

$$P_{\text{low}}^TP_{\text{low}} + P_{\text{low}}^T\delta P_{\text{low}} + \delta P_{\text{low}}^TP_{\text{low}} + \delta P_{\text{low}}^T\delta P_{\text{low}} = I.$$

Let us concentrate on the 2-norm and substitute $\|\delta P_{\text{low}}\| \leq p(n)u_{\text{low}}$,

$$\|P_{\text{low}}\|_2^2 \leq \|I\|_2 + 2p(n)u_{\text{low}}\|P_{\text{low}}\|_2 + p(n)^2u_{\text{low}}^2,$$

and the solution to this inequality is

$$\|P_{\text{low}}\|_2 \leq p(n)u_{\text{low}} + \sqrt{1 + 2p(n)^2u_{\text{low}}^2}, \tag{1.1}$$

which proves the claim. □

This theorem will be useful when we consider the convergence of the Newton–Schulz iteration for computing the unitary polar factor of P_{low} as the Newton–Schulz iteration, unlike the Newton iteration, in Section 3.2, Theorem 3.1, is only locally convergent.

2 Polar Decomposition

Before introducing the polar decomposition, the singular value decomposition (SVD) for a squared matrix is necessary to be described here.

Theorem 2.1 (singular value decomposition)**.** *Let $A \in \mathbb{C}^{n\times n}$, there exist unitary matrices $U, V \in \mathbb{C}^{n\times n}$ such that*

$$U^*AV = \Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_n) \in \mathbb{R}^{n\times n},$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. We will denote the decomposition $A = U\Sigma V^*$ as the singular value decomposition (SVD) of A, and the diagonal entries of A are called the singular values of A.

The polar decomposition can be then defined with the help of the SVD.

Theorem 2.2 (polar decomposition). *Let $A \in \mathbb{C}^{n\times n}$ have full rank. There exists a unique unitary matrix $U_A \in \mathbb{C}^{n\times n}$ and a unique Hermitian positive definite matrix $H_A \in \mathbb{C}^{n\times n}$ such that $A = U_A H_A$. The uniqueness of H_A comes from*

$$(A^*A)^{1/2} = (H_A^* U_A^* U_A H_A)^{1/2} = (H_A^2)^{1/2} = H_A,$$

and every Hermitian positive definite matrix has a unique Hermitian positive definite square root [5, 2008, Cor. 1.30]. $U_A = AH_A^{-1}$ will ensure the uniqueness of U_A.

This can be proved by using SVD. Let $A \in \mathbb{C}^{n\times n}$ have full rank with SVD $A = U\Sigma V^*$, then $\sigma_i > 0$ for all $i = 1 : n$. Then we can rewrite the SVD in the following form

$$A = U\Sigma V^* = (UV^*)(V\Sigma V^*) \coloneqq U_A H_A.$$

With this definition of U_A and H_A, all the conditions in the previous theorem can be easily checked. We will refer to U_A as the unitary polar factor and H_A as the Hermitian polar factor.

It is sufficient to only consider the square, nonsingular matrices for the polar decomposition. Let $A \in \mathbb{C}^{m\times n}$ with $m \geq n$ have the QR factorization $A = QR$ where $Q \in \mathbb{C}^{m\times n}$ and $R \in \mathbb{C}^{n\times n}$. Then find the polar decomposition $R = UH$ and $A = QU \cdot H$ becomes the polar decomposition of A. In addition, if $A \in \mathbb{C}^{m\times n}$ is singular, we can compute a complete orthogonal decomposition

$$A = P \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} Q^*,$$

where P, Q are unitary, and $R \in \mathbb{C}^{r\times r}$ is nonsingular and upper triangular. Then, we may compute the polar decomposition of R and assemble them together by [6, 1990, Sec. 2]. The unitary polar factor of A satisfies the best approximation property [5, 2008, Sec. 8.1].

Theorem 2.3 (best approximation). *Let $A \in \mathbb{C}^{n\times n}$ have the polar decomposition $A = UH$. Then $\|A - U\| = \min\{\|A - Q\| : Q^*Q = I\}$ for any unitarily invariant norm.*

Due to the best approximation property, considering the matrix P_{low} in Algorithm 1, the polar decomposition will give the unitary matrix that is closest to the input matrix. Moreover, the key benefit of using the polar decomposition over the QR decomposition is that the unitary polar factor can be computed via iteration with matrix multiplication only.

3 Computing the Unitary Polar Factor

A straightforward way of computing the unitary polar factor is using SVD.

Algorithm 2 Compute the unitary polar factor of $A \in \mathbb{R}^{n\times n}$ using the SVD.

```
1: Compute the SVD of $A$, $A = U\Sigma V^*$.
2: Compute the unitary polar factor via $U_A = UV^*$.
```

However, due to not exploiting the almost orthogonal structure, this algorithm is not competitive with the QR factorization, since the typical cost of computing the full SVD is about $26n^3$ [3, 2013, Fig. 8.6.1]. A more practical iterative method will be provided in the next section.

3.1 Newton Method

The Newton iteration for the unitary polar factor can be deduced by apply the Newton method to the equation $X^*X = I$. Let Y be an approximate solution, and then define the correction term $E = X - Y$. Substitute $X = Y + E$ into $X^*X = I$, we have $Y^*Y + Y^*E + E^*Y + E^*E = I$. Drop the second order term in E gives the Newton iteration for the unitary polar factor

$$\begin{aligned} E_k^* Y_k + Y_k^* E_k &= I - Y_k^* Y_k. \\ Y_{k+1} &= Y_k + E_k, \\ k = 0, 1, \dots, &\text{ and } Y_0 \text{ is given.} \end{aligned} \tag{3.1}$$

This form of the Newton method requires a solution for the Sylvester equation at each step, and the cost for solving the Sylvester equation is $O(n^3)$. Instead, we can point out one solution to the Sylvester equation $E_k = (Y_k - Y_k^{-*})/2$, and substitute into the Newton iteration (3.1), we have the Newton iteration

$$Y_{k+1} = \frac{1}{2}(Y_k^{-*} + Y_k), \qquad Y_0 = A \in \mathbb{C}^{n\times n}, \text{ nonsingular.} \tag{3.2}$$

Notice, we choose the starting point of our iteration to be A. This is essential for the convergence result in [4, 1986, Sec. 3.2].

3.2 Newton Schulz Iteration

The drawback of the Newton iteration is that it requires the explicit form of an inverse of a matrix during each step. To use the fact that the matrix multiplication is very fast on high-performance computers, one can transform this inversion into two matrix multiplications by using the Schulz iteration for matrix inverse. Instead of computing the all the way until we got an inverse at each step, we use the one-step Schulz iteration to approximate the inverse of Y_k^*, namely

$$Y_k^{-*} \approx X_0(2I - Y_k^* X_0), \qquad X_0 \text{ is given.}$$

According to [2, 1995], X_0 should be an approximation to Y_k^{-1}. Consider the situation we have in Algorithm 1, $Y_0 = P_{\text{low}}$ is an almost orthogonal matrix. Therefore, in case Y_k converges to an unitary matrix, Y_k should be an increasingly good approximation to Y_k^{-*}. Hence, we can let $X_0 = Y_k$ for each step, and obtain the following coupled iteration

$$\begin{aligned} Z_k &= Y_k(2I - Y_k^* Y_k), \\ Y_{k+1} &= \frac{1}{2}(Z_k + Y_k), \qquad Y_0 = A, \quad k = 0, 1, \dots, \end{aligned}$$

or more compactly, we have the Newton–Schulz iteration for the unitary polar factor of A,

$$Y_{k+1} = \frac{1}{2} Y_k(3I - Y_k^* Y_k), \qquad X_0 = A.$$

Notice that, at each step, we require 2 matrix multiplication instead of an inversion. The Newton–Schulz iteration, unfortunately, is only locally convergent, and can be characterized by the following theorem.

Theorem 3.1 (convergence of Newton–Schulz iteration)**.** *Let $A \in \mathbb{C}^{n\times n}$ have full rank with the polar decomposition $A = UH$ and $\sigma_i(A) \in (0, \sqrt{3})$. Then the Newton–Schulz iteration converges to the unitary polar factor U quadratically, and*

$$\|Y_{k+1} - U\|_2 \le \frac{1}{2}\|Y_k + 2U\|_2\|Y_k - U\|_2^2.$$

Moreover, let $R_k = I - X_k^* X_k$, then the convergence can be viewed as

$$R_{k+1} = \frac{3}{4}R_k^2 + \frac{1}{4}R_k^3.$$

Proof. See problem 8.20 in [5, 2008]. □

Since the Newton–Schulz iteration is only locally convergent, the next corollary provide a sufficient condition on P_{low} for the convergence of the iteration.

Corollary 3.2. *Let P_{low} be a computed eigenvector matrix via any LAPACK routine in precision u_{low} with $p(n)$ defined in Theorem 1.1. If $p(n)u_{\text{low}} \le \sqrt{5}-\sqrt{3}$, then $\|P_{\text{low}}\|_2 < \sqrt{3}$. Moreover, the Newton–Schulz method will converge by Theorem 3.1.*

Proof. From Theorem 1.2, we have

$$\|P_{\text{low}}\|_2 \le p(n)u_{\text{low}} + \sqrt{1 + 2p(n)^2 u_{\text{low}}^2}.$$

By Theorem 3.1, the Newton–Schulz iteration on P_{low} converges to its unitary polar factor U_P if $\sigma_i(P_{\text{low}}) \in (0, \sqrt{3})$. Since all the singular values are positive, the condition is equivalent to $\|P_{\text{low}}\|_2 = \sigma_1 < \sqrt{3}$. In order to bound the right-hand side by $\sqrt{3}$, by solving quadratic equation, we have $p(n)u_{\text{low}} < \sqrt{5} - \sqrt{3}$. □

Notice that this condition is easy to satisfy since $p(n) \approx n$ in practice. Suppose the low precision is single precision, $u_{\text{low}} \approx 6 \times 10^{-8}$, then as long as the size of the matrix does not exceed about 10^7, then we have guaranteed convergence for the Newton–Schulz iteration based on Corollary 3.2. Therefore, from now on, we should assume that $p(n)u_{\text{low}} \ll 1$. Assemble them, we have the following algorithm

Algorithm 3 Newton–Schulz iteration on P_{low}, an eigenvector matrix computed via any LAPACK routine in precision u_{low}. The algorithm will orthogonalize P_{low} to P_{high} such that $\|P_{\text{high}}^* P_{\text{high}} - I\| \le n u_{\text{high}}$ where $u_{\text{high}} \ll u_{\text{low}}$.

```
$X_0 = P_{\text{low}}$
for $i = 0, 1, 2, \cdots$, until converge do
    Compute $X_i^* X_i$                                  → 1 matrix-matrix multiplication
    if $\|X_i^* X_i - I\| \le n u_{\text{high}}$ then
        Break, $P_{\text{high}} = X_i$, Quit.
    end if
    Compute $X_{i+1} = \frac{1}{2}(3X_i - X_i(X_i^* X_i))$.   → 1 matrix-matrix multiplication
end for
```

Algorithm 3 is an iterative refinement on P_{low}, and it is hard to analyze its complexity. However, it turns out that this algorithm can converge very fast. To see this, we need an upper bound on $\|I - P_{\text{low}}^T P_{\text{low}}\|_2$.

Corollary 3.3. *Let P_{low} be an computed eigenvector matrix via any LAPACK routine in precision u_{low}. Then we have*

$$\|I - P_{\text{low}}^T P_{\text{low}}\|_2 \leq 2p(n)u_{\text{low}} + O(p(n)^2 u_{\text{low}}^2),$$

where $p(n)$ is a modestly increasing function of n.

Proof. From Theorem 1.1 and proof of Theorem 1.2, we have

$$P_{\text{low}}^T P_{\text{low}} + P_{\text{low}}^T \delta P_{\text{low}} + \delta P_{\text{low}}^T P_{\text{low}} + \delta P_{\text{low}}^T \delta P_{\text{low}} = I.$$

where $\|\delta P_{\text{low}}\|_2 \leq p(n)u_{\text{low}} =: \varepsilon$, and $p(n)$ is a modestly growing function of n. Manipulate the above equation and taking any norm, we have

$$\|I - P_{\text{low}}^T P_{\text{low}}\|_2 \leq 2\|\delta P_{\text{low}}\|_2 \|P_{\text{low}}\|_2 + \|\delta P_{\text{low}}\|_2^2.$$

Using $\|\delta P_{\text{low}}\|_2 \leq \varepsilon$, we have

$$\|I - P_{\text{low}}^T P_{\text{low}}\|_2 \leq 2\varepsilon \|P_{\text{low}}\|_2 + \varepsilon^2.$$

Using Theorem 1.2, $\|P_{\text{low}}\|_2 \leq \varepsilon + \sqrt{1 + 2\varepsilon^2}$, we have

$$\begin{aligned} \|I - P_{\text{low}}^T P_{\text{low}}\|_2 &\leq 2\varepsilon(\varepsilon + \sqrt{1 + 2\varepsilon^2}) + \varepsilon^2 \\ &\leq 2\varepsilon(\varepsilon + 1 + \sqrt{2}\varepsilon) + \varepsilon^2 \qquad \sqrt{x+y} \leq \sqrt{x} + \sqrt{y},\ \forall x, y \geq 0 \\ &= 2\varepsilon + (3 + 2\sqrt{2})\varepsilon^2 = 2\varepsilon + O(\varepsilon^2). \end{aligned}$$

Substituting back, we have proved the claim. □

Using the matrix norm property [3, 2013, Sec. 2.3.2],

$$\|A\|_F / \sqrt{n} \leq \|A\|_2, \qquad A \in \mathbb{R}^{n \times n}.$$

By Corollary 3.3, we transform the matrix 2-norm to Frobenius norm, and use the same definition $p(n)u_{\text{low}} =: \varepsilon \ll 1$, we have

$$\|I - P_{\text{low}}^T P_{\text{low}}\|_F \leq 2\sqrt{n}\varepsilon + O(\varepsilon^2).$$

Let P_{low} have the SVD $U\Sigma_P V^*$, then using the unitarily invariance and the definition of the Frobenius norm, we have

$$\|I - P_{\text{low}}^T P_{\text{low}}\|_F^2 = \|I - \Sigma_P^T \Sigma_P\|_F = (\sigma_1^2 - 1)^2 + \cdots + (\sigma_n^2 - 1)^2 \leq 2\sqrt{n}\varepsilon.$$

Since the terms in the summation are all positive, therefore $(\sigma_i^2 - 1)^2 \leq 2\sqrt{n}\varepsilon$ for $i = 1, \ldots, n$. This indicates that all the singular values of P_{low} are wiggling about 1 and this explains the matrix P_{low} is well conditioned.

Moreover, using Corollary 3.3 and Theorem 3.1, we have, after one Newton–Schulz iteration,

$$\begin{aligned} \|I - X_1^T X_1\|_2 &\leq \frac{3}{4}\|I - X_0^T X_0\|_2^2 + \frac{1}{4}\|I - X_0^T X_0\|_2^3 \\ &\leq \frac{3}{4}(2\varepsilon + O(\varepsilon^2))^2 + \frac{1}{4}(2\varepsilon + O(\varepsilon^2))^3 \\ &= 3\varepsilon^2 + O(\varepsilon^3) = 3p(n)^2 u_{\text{low}}^2 + O(p(n)^3 u_{\text{low}}^3). \end{aligned}$$

If we require the iteration converged in one iteration, then we need $3p(n)^2u_{\text{low}}^2 \leq p(n)u_{\text{high}}$, and this is equivalent to $u_{\text{high}} \geq 3p(n)u_{\text{low}}^2$. Notice that we ignore the lower order term here, since we are doing approximation here, and lower order term will not play an significant role if $p(n)^2u_{\text{low}}^3 \ll 1$. However, this condition is not possible if we consider u_{low} and u_{high} as single and double precision. Therefore, we proceed to another iteration,

$$\|I - X_2^T X_2\|_2 \leq \frac{27}{4}\varepsilon^4 + O(\varepsilon^5) = \frac{27}{4}p(n)^4u_{\text{low}}^4 + O(p(n)^5u_{\text{low}}^5). \tag{3.3}$$

Suppose we require the iteration converge in two iteration, then using the same requirement, we have $6.75p(n)^3u_{\text{low}}^4 \leq u_{\text{high}}$. Let us consider u_{low} and u_{high} as single and double precision, and assume $p(n) = n$, then to achieve this condition, we approximately need $n \leq 10^4$.

3.3 Numerical Experiments

Example 3.4. The first experiment using the almost orthogonal matrix, P_{low}, generated by applying MATLAB builtin `eig()` at single precision to symmetric matrix A with geometrically distributed singular values. The size of A is varied from 10 to 3000 and generated by using MATLAB command `round(linspace(10,3e3,1e2))`. Although from previous analysis, it is no need to control the condition number, but we set it as $\kappa(A) = 100$. We will apply two different functions: (i) the 2 step Newton–Schulz iteration; (ii) MATLAB builtin `qr`. From (3.3), using Newton–Schulz iteration, the output P_{high} should satisfy $\|P_{\text{high}}^T P_{\text{high}} - I\|_2 \leq 6.75p(n)^4u_{\text{low}}^4$. The Householder QR decomposition (used by MATLAB `qr`) produces an orthogonal matrix Q that satisfies $\|Q^*Q - I\|_2 \leq q(n)u_{\text{high}}$, where $q(n)$ is a modestly increasing function in n. The choice of the dimension ensures Newton–Schulz iteration to compute a more orthogonal result than the Householder QR factorization. This can be seen from the experiment results.

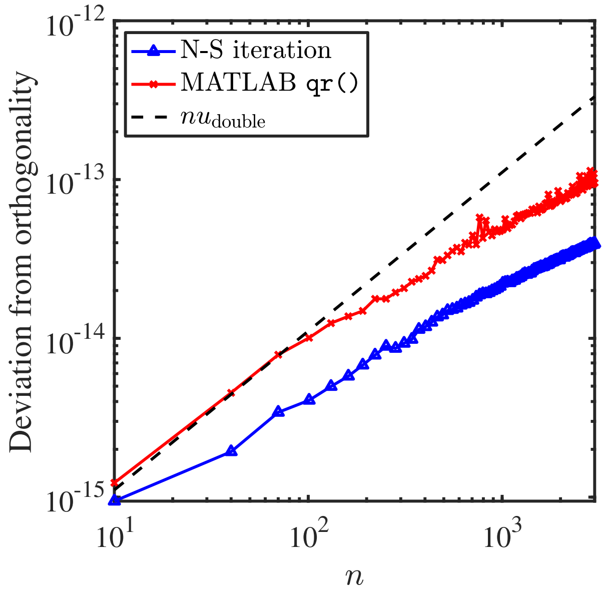

Figure 1: Deviation from orthogonality after applying two Newton–Schulz iteration on the almost orthogonal matrix P_{low} described in Algorithm 1, compared with the MATLAB builtin `qr()`.

In Figure 1, we plot the deviation from orthogonality. We see that after two iteration, the Newton–Schulz iteration will produce a uniformly less deviation from orthogonality than the QR orthogonalization process produced by `qr` function.

Example 3.5. Using the same setup as Example 3.4, we would like to further compare the deviation from orthogonality between the 2 step Newton–Schulz iteration and the modified Gram-Schmidt (MGS) QR factorization. The MATLAB program for MGS is generate from the pseudo-code in [9, 1998, Chap. 4, Alg. 1.11].

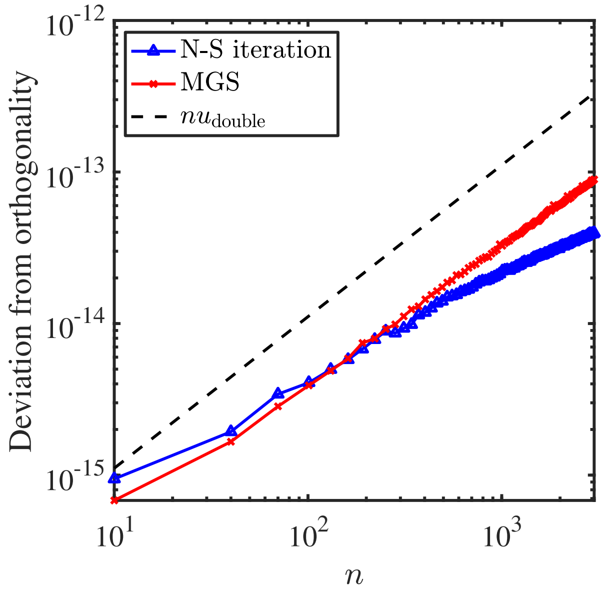

Figure 2: Deviation from orthogonality after applying two Newton–Schulz iteration on the almost orthogonal matrix P_{low} described in Algorithm 1, compared with the MGS QR factorization.

From Figure 2, Newton–Schulz iteration is competitive with the MGS, the suggested method by [10, 2022], in terms of the orthogonality. These methods both produce an orthogonal matrix in double precision. Although the gap between the trajectories is less significant than the gap shown in Figure 1. Newton–Schulz iteration produces an asymptotically smaller deviation from orthogonality.

References

[1] Edward Anderson, Zhaojun Bai, Christian Bischof, L. Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, and Danny Sorensen. *LAPACK Users' Guide*. 3rd edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999. 429 pp. ISBN 0-89871-447-8. (Cited on p. 1.)

[2] Adi Ben-Israel. An iterative method for computing the generalized inverse of an arbitrary matrix. *Mathematics of Computation*, 19(91):452–455, 1965. (Cited on p. 4.)

[3] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins studies in the mathematical sciences. 4th edition, Johns Hopkins University Press, Baltimore, MD, USA, 2013. ISBN 978-1-4214-0794-4. (Cited on pp. 2, 3, and 6.)

[4] Nicholas J. Higham. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986. (Cited on p. 4.)

[5] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. xx+425 pp. ISBN 978-0-898716-46-7. (Cited on pp. 3 and 5.)

[6] Nicholas J. Higham and Robert S. Schreiber. Fast polar decomposition of an arbitrary matrix. *SIAM Journal on Scientific and Statistical Computing*, 11(4):648–655, 1990. (Cited on p. 3.)

[7] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. 2nd edition, Cambridge University Press, Cambridge, UK, 2013. ISBN 978-0-521-83940-2 (Hardback). (Cited on p. 1.)

[8] ———. IEEE standard for floating–point arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754–2008)*, pages 1–84, 2019. (Cited on p. 1.)

[9] Gilbert W. Stewart. *Matrix Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, January 1998. xix+460 pp. Volume 1: Basic Decompositions. ISBN 978-0-898714-14-2 (v.1: pbk.). (Cited on p. 8.)

[10] Zhiyuan Zhang and Zheng-Jian Bai. A mixed precision Jacobi method for the symmetric eigenvalue problem, November 2022. (Cited on pp. 1 and 8.)