Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 20, No. 2, pp. 185–211, 2024

RANDOMIZED QUATERNION MATRIX UTV DECOMPOSITION AND ITS APPLICATIONS IN QUATERNION MATRIX OPTIMIZATION*

Renjie Xu and Yimin Wei†

Abstract: Quaternion singular value decomposition (QSVD) plays a fundamental role in quaternion matrix optimization. This paper introduces a two-sided random orthogonalization decomposition named quaternion matrix UTV (QUTV) decomposition to replace the QSVD in some applications of quaternion matrix optimization. The compressed randomized QUTV (CoR-QUTV) algorithm is simple and accurate in color image reconstruction. The block-randomized QUTV (Block-QUTV) algorithm outperforms QSVD in terms of efficiency and accuracy in the problem of eigenvalues. The QUTV decompositions are theoretically reliable and the numerical examples in quaternion matrix optimization verify our claims.

Key words: *quaternion matrix UTV decomposition, randomized algorithm, quaternion matrix optimization*

Mathematics Subject Classification: *15B33, 68W20*

---

1 Introduction

Quaternions matrices have been widely used in color image reconstruction, color image classification, signal denoising, etc [4, 6, 9, 34]. The core mathematical models among most of these works can be formulated by optimization problems with quaternion matrices as decision variables. Qi *et al.* [30] provide necessary theoretical foundations on optimality analysis, in order to enrich the contents of optimization theory and to pave way for the design of efficient numerical algorithms as well. However, the disadvantages of quaternion structure will cause inefficiency in application scenarios. The singular value decomposition (SVD) [10] and the rank revealing QR (RRQR) decomposition [3, 11] are very popular and effective for computing the low-rank approximation of matrices. In this paper, we combine the accuracy of the SVD with the rank revealing of RRQR on the structure of the UTV decomposition [35, 36]. CoRQUTV decomposition and BlockUTV decomposition are proposed for dealing with large-scale color image reconstruction. The error analysis will ensure the feasibility and reliability of our quaternion UTV algorithms and numerical examples in quaternion matrix optimization will verify our claims.

Nowadays, quaternion matrices can be used to process more and more research fields, such as signal processing [9, 12], image processing [21, 17] and machine learning [27, 28]. For

*This project is supported by the Ministry of Science and Technology of China under grant G2023132005L and Shanghai Municipal Science and Technology Commission under grant 23WZ2501400.

†Corresponding author.

DOI: https://doi.org/10.61208/pjo-2023-042

all these purposes, the QSVD is the most important for processing the quaternion matrices. Due to the noncommutative multiplication of quaternion, it is still a great challenging topic to deal with the quaternion matrices. To our best knowledge, there are three kinds of methods to achieve the QSVD. The first QSVD is used in the Quaternion toolbox for Matlab (QTFM) developed by Sangwine and Bihan in 2005 [32]. The codes for QTFM are based on the quaternion arithmetic operations. They are less efficient for large-scale matrices. The second QSVD is to use the real structure-preserving QSVD method via the real counterparts of quaternion matrices [22]. The third QSVD keeps the real structure-preserving which is developed on the base of the second QSVD by Jia [16]. These two real structure-preserving decompositions maintain the structure and sacrifice simplicity. We consider the complex adjoint matrices of the quaternion matrices [21, 33]. This idea is helpful for us to keep the structure of the matrix decomposition so that a more efficient structure can be substituted into the quaternion operation. A new decomposition is considered to be the potential replacement of the QSVD.

Inspired by recent developments, this paper presents the UTV decompositions of quaternion matrices. Given a matrix $A \in \mathbb{C}^{m\times n}$, the UTV algorithm computes a decomposition $A = UTV^*$, where U and V have orthonormal columns, and T is triangular (either upper or lower triangular) [35, 36]. There are two methods that depend on the randomized algorithms to form the UTV decomposition. The first one combines the randomized algorithm with the rank-revealing QR algorithm to form the UTV decomposition. The randomized rank revealing UTV is fast and stable [7]. The second one diagonalizes the block matrix on the diagonal block. We graft this technology onto the quaternion matrices to replace the QSVD. The first one is called the CoR-QUTV algorithm and the second one is called the BlockQUTV algorithm. Then we analyze the errors of our QUTV decomposition and the numerical experiments show that the QUTV decomposition is both time-saving and accurate. The rank-revealing property of CoR-QUTV algorithm makes a great contribution to quaternion matrix optimization.

The paper is organized as follows. In Section 2, we provide some preliminary results about quaternion matrices, the quaternion SVD, the randomized SVD technology, and the UTV decomposition. The quaternion UTV decomposition will be studied in Section 3. In Section 4, the theoretical analysis is provided for the approximation errors. We test our algorithms with some numerical examples in quaternion matrix optimization and show their efficiency in Section 5.

2 Preliminaries

In this section, we will introduce the knowledge of quaternion matrices and quaternion SVD decomposition for quaternion matrix optimization. The randomized SVD and UTV decompositions will also be mentioned, and then we will reveal the relationship between them.

2.1 QSVD in Quaternion Matrix Optimization

According to the standard notation in [15, 29, 41], the quaternion field $\mathbb{Q}$ is an associative but noncommutative algebra over the real field $\mathbb{R}$. Any quaternion $q \in \mathbb{Q}$ is given by

$$q = q_0 + q_1 i + q_2 j + q_3 k,$$

where $q_0, q_1, q_2, q_3 \in \mathbb{R}$ and i,j,k symbols satisfying the multiplication table formed by $i^2 = j^2 = k^2 = i\,j\,k = -1$. The conjugate and norm of q are defined by $q^* = q_0 - q_1 i - q_2 j - q_3 k$

and $|q| = \sqrt{q^*q} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$, respectively. Similarly, for any quaternion matrices

$$\begin{cases} P = P_0 + P_1 i + P_2 j + P_3 k \in \mathbb{Q}^{m\times n}, \\ Q = Q_0 + Q_1 i + Q_2 j + Q_3 k \in \mathbb{Q}^{m\times n}, \\ S = S_0 + S_1 i + S_2 j + S_3 k \in \mathbb{Q}^{n\times l}. \end{cases}$$

Denote the conjugate of the quaternion matrix

$$P^* = P_0^\top - P_1^\top i - P_2^\top j - P_3^\top k \in \mathbb{Q}^{n\times m}$$

and the transpose of quaternion matrix P,

$$P^\top = P_0^\top + P_1^\top i + P_2^\top j + P_3^\top k \in \mathbb{Q}^{n\times m}.$$

The sum of P and Q is

$$P + Q = (P_0 + Q_0) + (P_1 + Q_1)i + (P_2 + Q_2)j + (P_3 + Q_3)k \in \mathbb{Q}^{m\times n},$$

and the multiplication of Q and S is given by

$$\begin{gathered}(Q_0S_0 - Q_1S_1 - Q_2S_2 - Q_3S_3) + (Q_0S_1 + Q_1S_0 + Q_2S_3 - Q_3S_2)i \\ (Q_0S_2 - Q_1S_3 + Q_2S_0 + Q_3S_1)j + (Q_0S_3 + Q_1S_2 - Q_2S_1 + Q_3S_0)k.\end{gathered}$$

An alternative way to define quaternions is to consider the subset of the ring $M(2,\mathbb{C})$. For $Q = Q_{c1} + Q_{c2}j \in \mathbb{Q}^{m\times n}$, where $Q_{c1} = Q_0 + Q_1 i$, $Q_{c2} = Q_2 + Q_3 i \in \mathbb{C}^{m\times n}$, we call the $2m \times 2n$ complex matrix [42]

$$\chi_Q = \begin{pmatrix} Q_{c1} & Q_{c2} \\ -\overline{Q_{c2}} & \overline{Q_{c1}} \end{pmatrix}$$

the *complex adjoint matrix* or *adjoint* of Q, symbolized χ_Q. Note that χ_Q has a special complex algebraic structure that is preserved under the following operations,

$$\chi_{k_1P+k_2Q} = k_1\chi_P + k_2\chi_Q \ (k_1, k_2 \in \mathbb{C}), \quad \chi_{Q^*} = \chi_Q^*, \quad \chi_{QS} = \chi_Q\chi_S.$$

Let A be an $n \times n$ quaternion matrix, and χ_A be the complex adjoint matrix of A. We can define the determinant of A as $\det(A) = [\det(\chi_A)]^{\frac{1}{2}}$.

Let $A \in \mathbb{Q}^{m\times n}$. $A^\dagger$ is denoted as the Moore–Penrose inverse or the pseudo-inverse of A if $X = A^\dagger$ satisfies the following four equations,

$$AXA = A, \quad XAX = X, \quad (AX)^* = AX, \quad (XA)^* = XA.$$

We will define the quaternion random Gaussian matrices [24] $\Phi \in \mathbb{Q}^{m\times n}$ as

$$\Phi = \Phi_0 + \Phi_1 i + \Phi_2 j + \Phi_3 k, \tag{2.1}$$

where the entries of $\Phi_0, \Phi_1, \Phi_2, \Phi_3$ are random and independently drawn from the $\mathcal{N}(0,1)$-normal distribution.

The quaternion SVD decomposition (QSVD) and singular values of dual quaternion matrices and their low-rank approximations can be found in [41, 23]. The algorithm of QSVD is computed by a complex adjoint matrix presented in [4].

Theorem 2.1 (QSVD [41]). *Let $A \in \mathbb{Q}^{m\times n}$ be of rank r. Then there exist unitary quaternion matrices $U \in \mathbb{Q}^{m\times m}$ and $V \in \mathbb{Q}^{n\times n}$ such that*

$$U^*AV = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \tag{2.2}$$

where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{m\times n}$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$ with σ_k denoting the k-th largest singular value of A and $r = \min\{m, n\}$.

Furthermore, through complex adjoint of QSVD: $\chi_U^* \chi_A \chi_V = \chi_\Sigma$, where χ_U and χ_V are complex orthogonal matrices and $\chi_\Sigma = \mathrm{diag}(\sigma_1, \sigma_1, \sigma_2, \sigma_2, \ldots, \sigma_r, \sigma_r)$.

Given a quaternion matrix $A \in \mathbb{Q}^{m\times n}$, where $m \geq n$, the QSVD [4] can also be defined as:

$$\begin{aligned} A &= U_A \Sigma_A V_A^* \\ &= \begin{pmatrix} U_k & U_0 \end{pmatrix} \begin{pmatrix} \Sigma_k & O \\ O & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^* \\ V_0^* \end{pmatrix} \end{aligned} \tag{2.3}$$

where $U_k \in \mathbb{Q}^{m\times k}, V_k \in \mathbb{Q}^{n\times k}$ are, respectively, sub-matrices of $U_A \in \mathbb{Q}^{m\times n}$ and $V_A \in \mathbb{Q}^{n\times n}$ by taking their first k columns, and $U_0 \in \mathbb{Q}^{m\times(n-k)}, V_0 \in \mathbb{Q}^{n\times(n-k)}$. $\Sigma_k \in \mathbb{R}^{k\times k}$ and $\Sigma_0 \in \mathbb{R}^{(n-k)\times(n-k)}$ are diagonal matrices containing the singular values, i.e., $\Sigma_k = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$ and $\Sigma_0 = \mathrm{diag}(\sigma_{k+1}, \sigma_{k+2}, \ldots, \sigma_n)$. A can be written as $A = A_k + A_0$, where $A_k = U_k \Sigma_k V_k^*$, and $A_0 = U_0 \Sigma_0 V_0^*$. The QSVD constructs the optimal rank-k approximation A_k to A.

The spectral norm (2-norm) of a quaternion vector $x = [x_i] \in \mathbb{Q}^n$ is $\|x\|_2 := \sqrt{\Sigma_i |x_i|^2}$. The 2-norm of a quaternion matrix $A = [a_{ij}] \in \mathbb{Q}^{m\times n}$ is $\|A\|_2 = \max_i \sigma_i(A)$, where $\sigma_i(A)$ is singular value of A. The Frobenius norm of A is $\|A\|_F = (\Sigma_{i,j} |a_{ij}|^2)^{\frac{1}{2}} = [\mathrm{tr}(A^*A)]^{\frac{1}{2}}$. The nuclear norm of A is $\|A\|_* = \sum_i \sigma_i$. As a result, spectral and Frobenius norms of a quaternion matrix can be represented by the ones of complex adjoint matrices as

$$\|A\|_2 = \|\chi_A\|_2, \quad \|A\|_F^2 = \frac{1}{2}\|\chi_A\|_F^2. \tag{2.4}$$

Moreover, for consistent quaternion matrices A and B, it is obvious that [10]

$$\|AB\|_F \leq \|A\|_2 \|B\|_F, \qquad \|AB\|_F \leq \|A\|_F \|B\|_2. \tag{2.5}$$

2.2 Randomized SVD and UTV Decompositions

This section will summarise the most important results [14, 25, 31] on randomized algorithms for constructing the orthogonal subspace of a given matrix. To be precise, let $A \in \mathbb{C}^{m\times n}$ and a target rank k satisfy $1 \leq k < \min\{m, n\}$, and suppose that we seek to find an $m \times k$ orthogonal matrix Q such that

$$\|A - QQ^*A\|_2 \approx \min_{\mathrm{rank}(X) \leq k} \|A - X\|_2. \tag{2.6}$$

Generally, the columns of Q should approximately be formed by the right singular subspace of A. This is ideal for the subspace iteration (see Demmel [8]), particularly when started by a Gaussian random matrix [14, 31]. The objective matrix of the projection space can be obtained through the power iterations, and we can achieve the following randomized SVD (RSVD) framework [14],

1. Draw a Gaussian random $n \times k$ matrix G ($G = randn(n, k)$).

2. Form a $m \times k$ matrix Y via $Y = (AA^*)^q AG$.

3. Construct a matrix Q whose columns form an orthonormal basis for the range of Y ($[Q, \sim] = qr(Y)$).

4. Compute the low rank approximation $\widehat{A} = Q^* A$.

5. Compute the full SVD of $\widehat{A} = \widehat{U}\widehat{S}\widehat{V}^*([\widehat{U}, \widehat{S}, \widehat{V}] = svd(Q^*A))$.

6. Achieve the approximate low rank SVD of A by $U = Q\widehat{U}(:, 1:k), S = \widehat{S}(1:k, 1:k)$ and $V = \widehat{V}(:, 1:k)$.

These are standard schemes to obtain the low-rank SVD factors by randomized algorithms. We can observe that applying A^* and A alternatively to a tall thin matrix with k columns can achieve the matrix Y in Step 2. In some cases, orthogonalization is used to reduce the computational error caused by the floating point arithmetic. It is demonstrated that taking just a few steps of power iteration ($q = 0$,1 or 2) is enough by using a Gaussian random matrix as the starting point in [14, 31].

Remark 2.2 (Oversampling)**.** To analyze the power iteration with a Gaussian random matrix, it is common to select several "extra" samples. In fact, we will choose a small integer p representing the amount of oversampling samples, actully $p = 5$ or $p = 10$, and starts with a Gaussian matrix of size $n \times (k + p)$. This constructs an orthonormal matrix Q of size $m \times (k+p)$. Then the error $\|A - QQ^*A\|_2$ is close to the minimal error in rank-k approximation in both spectral and Frobenius norms [14] with probability almost 1. The oversampling trick will reduce the error of choosing the wrong columns. We can promote the accuracy of the randomized algorithms.

Given a matrix A of size $m \times n$, we can use the random sampling technique to compute an approximate rank-k SVD which named as randomized SVD (RSVD). We denote the orthonormal matrix Q which was constructed by a randomized range finder to compute the error matrix $E = A - QQ^*A$.

The UTV decomposition [35, 36] is a compromise between the SVD and the QR with Column Pivoting (QRCP), which has the virtues of both. For the matrix $A \in \mathbb{C}^{m\times n}$, it takes the form

$$A = UTV^*, \tag{2.7}$$

where $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary matrices and $T \in \mathbb{R}^{m\times n}$ is triangular. If T is upper triangular, then the decomposition is called a rank-revealing URV decomposition [35],

$$A = U \begin{pmatrix} T_{11} & T_{12} \\ O & T_{22} \end{pmatrix} V^*, \tag{2.8}$$

where T_{11} is $k \times k$ nonsingular, T_{11} and T_{22} are upper triangular,

$$\sigma_{\min}(T_{11}) = O(\sigma_k(A)), \qquad \|[T_{12}^\top \ \ T_{22}^\top]^\top\|_2 = O(\sigma_{k+1}(A)). \tag{2.9}$$

If T is lower triangular, then the decomposition is called a rank-revealing ULV decomposition [36],

$$A = U \begin{pmatrix} T_{11} & O \\ T_{21} & T_{22} \end{pmatrix} V^*, \tag{2.10}$$

where T_{11} is $k \times k$ nonsingular, T_{11} and T_{22} are lower triangular and

$$\sigma_{\min}(T_{11}) = O(\sigma_k(A)), \qquad \|[T_{12}^\top \ \ T_{22}^\top]^\top\|_2 = O(\sigma_{k+1}(A)). \tag{2.11}$$

The URV and ULV decompositions are collectively referred to as the UTV decomposition. Generally speaking, the upper and lower triangular forms are obtained through the left and right orthogonal transformations of the matrix A. When A is approximately of rank k, the decomposition is rank revealing. The rank-revealing UTV decomposition will reveal the numerical rank in the triangular sub-matrix $T_{11} \in \mathbb{C}^{k\times k}$.

Theorem 2.3 (UTV Decomposition [35, 36])**.** *Let $A \in \mathbb{C}^{m\times n}$. There is a decomposition $A = UTV^*$, where $U \in \mathbb{C}^{m\times m}, V \in \mathbb{C}^{n\times n}$ are unitary matrices and $T \in \mathbb{C}^{m\times n}$ is triangular with sub-matrix $T_{11} \in \mathbb{C}^{k\times k}$ which is rank revealing, where $0 \le k \le \min\{m, n\}$.*

The UTV decomposition has a variety of presentation forms, such as the QR decomposition with column pivoting, the SVD decomposition, and so on. How to view the flexible decomposition depends on how to understand matrices U, T, and V. The most important characteristic of the UTV decomposition in this paper is rank revealing combining the randomized algorithms with the UTV decomposition generates many efficient algorithms. According to the above analysis, the UTV decomposition will take advantage of the principal component analysis. The UTV decompositions of the tensor and the dual matrix can be found in [5, 40].

3 Quaternion Matrix UTV Decomposition

In this section, we will develop the quaternion matrix UTV (QUTV) decomposition and present some QUTV algorithms based on different ideas.

Theorem 3.1 (QUTV Decomposition)**.** *Let $A \in \mathbb{Q}^{m\times n}$. There is a decomposition $A = UTV^*$, where $U \in \mathbb{Q}^{m\times m}, V \in \mathbb{Q}^{n\times n}$ are quaternion unitary matrices and $T \in \mathbb{Q}^{m\times n}$ is triangular with sub-matrix $T_{11} \in \mathbb{Q}^{k\times k}$, where $0 \le k \le \min\{m, n\}$. The diagonals of T are real.*

Proof. Here we only prove the upper triangular form as a description. Following Stewart [35, 36] and Bunse [1], there exist quaternion unitary matrices $U \in \mathbb{Q}^{m\times m}, V \in \mathbb{Q}^{n\times n}$ such that

$$U^*AV = \begin{pmatrix} \lambda & \alpha \\ 0 & B \end{pmatrix}, \tag{3.1}$$

where $\lambda \in \mathbb{R}, \alpha \in \mathbb{Q}^{1\times(n-1)}$ and $B \in \mathbb{Q}^{(m-1)\times(n-1)}$. Then we can get the QUTV decomposition by the mathematical induction. $\square$

Consider the scheme of the QSVD [4] and quaternion QR algorithm [1], we can use the complex adjoint matrix $\chi_A \in \mathbb{C}^{2m\times 2n}$ to compute the quaternion matrix $A \in \mathbb{Q}^{m\times n}$. Given a quaternion matrix $A \in \mathbb{Q}^{m\times n}$, we will take the upper triangular form as our QUTV decomposition:

$$A = U \begin{pmatrix} T_{11} & T_{12} \\ O & T_{22} \end{pmatrix} V^*. \tag{3.2}$$

If there is a well-defined gap in the singular value spectrum of A, i.e., $\sigma_k \gg \sigma_{k+1}$, then the QUTV decompositions are said to be rank-revealing in the sense that the numerical rank k is revealed.

The scheme of randomized algorithms is suitable for the construction of the QUTV decomposition.

3.1 Compressed Randomized Quaternion UTV Algorithm

In this section, we present a rank-revealing decomposition algorithm powered by the randomized sampling schemes named compressed randomized quaternion UTV (CoR-QUTV) decomposition, which computes a low-rank approximation of a given quaternion matrix. For the matrix $A \in \mathbb{Q}^{m\times n}$, we choose a more efficient randomized algorithm as Algorithm 1. It will be considered as a two-sided randomized QSVD (TSR-QSVD) to approximate A which can be expressed as

---

Algorithm 1: Two-Sided Randomized Quaternion SVD (TSR-QSVD)

Input: Quaternion matrix $A \in \mathbb{Q}^{m\times n}$, integers k
Output: Quaternion matrix UTV decomposition

Draw quaternion random Gaussian matrices $\Phi_1 \in \mathbb{Q}^{n\times k}$ and $\Phi_2 \in \mathbb{Q}^{m\times k}$
Compute $Y_1 = A\Phi_1$ and $Y_2 = A^*\Phi_2$ in a single pass through A
Compute quaternion QR decompositions $Y_1 = Q_1R_1, Y_2 = Q_2R_2$
Compute $B_{\text{approx}} = Q_1^* Y_1 (Q_2^*\Phi_1)^\dagger$
Compute QSVD $B_{\text{approx}} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^*$
$A \approx (Q_1\widetilde{U})\widetilde{\Sigma}(Q_2\widetilde{V})^*$

$$A \approx Q_1Q_1^*AQ_2Q_2^*. \tag{3.3}$$

In Algorithm 1, $Q_1\widetilde{U} \in \mathbb{Q}^{m\times k}$ and $Q_2\widetilde{V} \in \mathbb{Q}^{n\times k}$ are approximations to the left and right singular subspaces of A, respectively. The Q_1, Q_2 will be obtained by the orthonormal basis of $\mathbf{R}(A)$ and $\mathbf{R}(A^*)$. $\widetilde{\Sigma} \in \mathbb{R}^{k\times k}$ contains an approximation to the first k singular values of A, and B_{approx} is an approximation to $B = Q_1^*AQ_2$. TSR-QSVD is a randomized algorithm of the QSVD. Neither theoretical error analysis nor numerical test is provided in [14]. Numerical instability will also be found in subsequent numerical examples. However, the biggest advantage of TSR-QSVD is very time-saving, which can instantly compress a $m \times n$ quaternion matrix to a $k \times k$ quaternion matrix. Following the idea of rank-revealing, we will produce a suitable QUTV decomposition by combining the previous compression techniques.

Algorithm 2: Compressed Randomized Quaternion UTV (CoR-QUTV)

Input: Quaternion matrix $A \in \mathbb{Q}^{m\times n}$, integers k
Output: Quaternion matrix UTV decomposition

Draw a random quaternion Guassian matrices $\Phi_1 \in \mathbb{Q}^{n\times k}$
Compute $C_1 = A\Phi_1$
Compute $C_2 = A^*C_1$
Compute quaternion QR decompositions $C_1 = Q_1R_1, C_2 = Q_2R_2$
Compute $D = Q_1^*AQ_2$
Compute the quaternion QR decomposition with column pivoting $D = \widetilde{Q}\widetilde{R}\widetilde{P}^*$
Form the CoR-QUTVdecomposition to approximate A
$\widehat{A}_{\text{CoR}} = UTV^*; U = Q_1\widetilde{Q}, T = \widetilde{R}, V = Q_2\widetilde{P}$

Given the matrix $A \in \mathbb{Q}^{m\times n}$ and an integer $0 \le k < \min\{m, n\}$, the CoR-QUTV decomposition will be computed as follows: Draw a quaternion random Gaussian matrix

$\Phi \in \mathbb{Q}^{n\times k}$. Then we compute the quaternion product

$$C_1 = A\Phi \tag{3.4}$$

where $C_1 \in \mathbb{Q}^{m\times k}$ is, in fact, a projection onto the subspace spanned by the columns of A. With C_1, we can form $C_2 \in \mathbb{Q}^{n\times k}$ by

$$C_2 = A^* C_1 \tag{3.5}$$

where C_2 is, in fact, a projection onto the subspace spanned by the columns of A^*. With the help of quaternion QR decomposition, we factor C_1 and C_2 as

$$C_1 = Q_1 R_1, \text{ and } C_2 = Q_2 R_2 \tag{3.6}$$

where Q_1 and Q_2 are approximate basis for the range spaces $\mathbf{R}(A)$ and $\mathbf{R}(A^*)$, respectively. Now we compress A through left and right multiplications by the orthonormal basis obtained, constructing the matrix $D \in \mathbb{Q}^{k\times k}$

$$D = Q_1^* A Q_2. \tag{3.7}$$

Computing the quaternion QR decomposition with column pivoting of D

$$D = \widetilde{Q}\widetilde{R}\widetilde{P}^*. \tag{3.8}$$

Finally, we achieve the CoR-QUTV decomposition of A

$$\widehat{A}_{\mathrm{CoR}} = UTV^*, \tag{3.9}$$

where $U = Q_1\widetilde{Q} \in \mathbb{Q}^{m\times k}$ and $V = Q_2\widetilde{P} \in \mathbb{Q}^{n\times k}$ form the approximations of the k leading left and right singular vectors of A, respectively, and $T = \widetilde{R} \in \mathbb{Q}^{k\times k}$ is upper triangular with diagonals approximating the first k singular values of A. The CoR-QUTV algorithm is presented in Algorithm 2. Through the process of Algorithm 2, we can notice that after obtaining the compression quaternion matrix D which is similar to Algorithm 1.

According to the same approach idea of (2.6), we can still adopt the q steps of a power iteration to reduce errors and improve the accuracy of the QUTV decomposition in these circumstances. Given the quaternion matrix $A \in \mathbb{Q}^{m\times n}$, and integers $0 \le k \le \min(m, n)$ and q, the resulting algorithm is described in Algorithm 3.

3.2 Block Randomized Quaternion UTV Algorithm

In this section, we focus on observing the structure of T and provide a block algorithm to obtain the QUTV decomposition. Given a quaternion matrix $A \in \mathbb{Q}^{m\times n}$ and a block size k, we seek two unitary matrices $U \in \mathbb{Q}^{m\times m}$ and $V \in \mathbb{Q}^{n\times n}$, respectively, such that the matrix

$$T = U^* AV \tag{3.10}$$

has a diagonal leading $k \times k$ block like T_{11} in (3.2), and the entries beneath T_{11} are all zeroed out. For simplicity, we assume that $m \ge n$ and $n = sk$, and we work with a block size k. Each four part of the quaternion matrix A can be partitioned into s blocks of k columns each. Algorithm 5 will iterate over s steps, where the i-th block of k columns at the i-th step is driven to upper triangular form by the left and the right multiplications of quaternion unitary matrices. We will represent such a multi-step iteration through a sequence of quaternion unitary matrices.

Algorithm 3: CoR-QUTV with Power Method

Input: Quaternion matrix $A \in \mathbb{Q}^{m\times n}$, integers k and q
Output: Quaternion UTV decomposition
Draw a random quaternion Guassian matrices $C_2 \in \mathbb{Q}^{n\times k}$
for *step* $= 1 : q+1$ **do**
  Compute $C_1 = AC_2$
  Compute $C_2 = A^* C_1$
Compute quaternion QR decompositions $C_1 = Q_1R_1, C_2 = Q_2R_2$
Compute $D = Q_1^* A Q_2$ or $D_{\text{approx}} = Q_1^* C_1 (Q_2^* C_2)^\dagger$
Compute a quaternion QR decomposition with column pivoting $D = \widetilde{Q}\widetilde{R}\widetilde{P}^*$ or $D_{\text{approx}} = \widetilde{Q}\widetilde{R}\widetilde{P}^*$
Form the CoR-QUTV decomposition to approximate A
$\widehat{A}_{\text{CoR}} = UTV^*; U = Q_1\widetilde{Q}, T = \widetilde{R}, V = Q_2\widetilde{P}$

$$\begin{cases} U = U^{(1)}U^{(2)}\cdots U^{(s)}, \\ V = V^{(1)}V^{(2)}\cdots V^{(s)}, \\ A^{(0)} = A, A^{(i)} = (U^{(i)})^* A^{(i-1)} V^{(i)}, i = 1,2,\ldots,s \text{ and } T = A^{(s)}. \end{cases} \tag{3.11}$$

Algorithm 4: stepQUTV

Input: Quaternion matrix $A \in \mathbb{Q}^{m\times n}$, integers k and q
Output: Quaternion UTV decomposition
Draw a random quaternion Guassian matrices $G \in \mathbb{Q}^{m\times k}$
Compute $Y = (A^*A)^q A^* G$
Compute quaternion QR decompositions $Y = VR$
Compute $D = AV(:, 1:k)$
Compute QSVD $D = UDW^*$
Form the stepQUTV decompostion $\widetilde{A} = UTV^*; U = U, T = [D, U^* * A * V(:, k+1 : end)], V = [V(:, 1:k) * W, V(:, k+1 : end)]$

To illustrate how to perform the QUTV decomposition, we show the operation steps of the framework (3.11) by explaining the computational process of the first block in detail in Algorithm 4. First of all, let us partition U and V so that

$$U = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \text{ and } V = \begin{pmatrix} V_1 & V_2 \end{pmatrix}, \tag{3.12}$$

where U_1 and V_1 each contain k columns. Then, set $T_{ij} = U_i^* A V_j$ for $i, j = 1, 2$ so that

$$U^*AV = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}. \tag{3.13}$$

Then we obtain the identity

$$A = AVV^* = \begin{pmatrix} AV_1 & AV_2 \end{pmatrix} V^*, \tag{3.14}$$

where the partitioning $V = \begin{pmatrix} V_1 & V_2 \end{pmatrix}$ is such that V_1 holds the first k columns of V. We now perform a full QSVD on the matrix $AV_1 \in \mathbb{Q}^{m\times k}$ so that

$$AV_1 = UD\widehat{V}^*. \tag{3.15}$$

Inserting into (3.14), we have the identity

$$A = \begin{pmatrix} UD\widehat{V}^* & AV_2 \end{pmatrix} V^*. \tag{3.16}$$

Extract the factor U to the left to get

$$A = U \begin{pmatrix} D\widehat{V}^* & U^*AV_2 \end{pmatrix} V^*. \tag{3.17}$$

Finally, extract the factor $\widehat{V}$ to the right to obtain the factorization

$$A = U \begin{pmatrix} D & U^*AV_2 \end{pmatrix} \mathbb{V}^*, \text{ with } \mathbb{V} = V \begin{pmatrix} \widehat{V} & O \\ O & I_{n-b} \end{pmatrix}, \tag{3.18}$$

we seek the QUTV decomposition $A = UTV^*$ with

$$T = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} = \begin{pmatrix} D(1:k, 1:k) & U_1^*AV_2 \\ O & U_2^*AV_2 \end{pmatrix}. \tag{3.19}$$

The computation steps of the quaternion unitary matrix U are exactly analogous to the calculation steps of the quaternion unitary matrix V in (3.19). Algorithm 4 achieves the step in (3.11) with power iteration of q steps.

Algorithm 5 is obtained by applying the single-step Algorithm 4 repeatedly, to drive A to upper triangular form one block of k columns at a time. At the start of the process, we initialize the output matrices T, U, and V by setting

$$T = A, \qquad U = I_m, \qquad V = I_n. \tag{3.20}$$

In the first step of the iteration, we use Algorithm 4 to create two quaternion unitary matrices $U^{(1)}$ and $V^{(1)}$ and then update T, U, and V accordingly:

$$T \leftarrow (U^{(1)})^* T V^{(1)}, \quad U \leftarrow UU^{(1)}, \quad V \leftarrow VV^{(1)}. \tag{3.21}$$

This leads to the matrix T, whose first k columns will be driven to the upper triangular. For the second step, the quaternion matrices $U^{(2)}$ and $V^{(2)}$ form by Algorithm 4 to the remainder matrix $T((b+1):m, (b+1):n)$ and then updating T, U, and V accordingly. The process will drive T to an upper triangular form k columns by k columns.

This Block-QUTV algorithm takes advantage of a randomized algorithm (RSVD) in a small block, and the QUTV decomposition ensures that the randomized algorithm can effectively execute on the diagonal block. All singular values can be approximated with less computation. The structure of the QUTV decomposition fully plays to its strengths for reducing the computation.

4 Error Analysis for Randomized UTV Algorithms

In this section, we will provide the error analysis of the CoR-QUTV and the Block-QUTV. Our analysis is established based on the framework of [18, 26].

Algorithm 5: Block-QUTV

Input: Quaternion matrix $A \in \mathbb{Q}^{m\times n}$, integers k and q
Output: Quaternion UTV decomposition

Given $T = A$, $U = I_m$ and $V = I_n$
for $i = 1 : ceil(n/k)$ **do**
  Compute $I_1 = 1 : k(i-1)$
  Compute $I_2 = k(i-1)+1 : m$
  Compute $J_2 = k(i-1)+1 : n$
  if $length(J_2) > k$ **then**
    Compute the stepQUTV decomposition $[UU, TT, VV] = stepQUTV(T(I_2, J_2), k, q)$
  else
    Compute the QSVD $[UU, TT, VV] = QSVD(T(I_2, J_2))$
  Compute $U(:, I_2) = U(:, I_2) * UU$
  Compute $V(:, J_2) = V(:, J_2) * VV$
  Compute $T(I_2, J_2) = TT, T(I_1, J_2) = T(I_1, J_2) * VV$

4.1 Error Analysis of CoR-QUTV Algorithm

We will prove that the CoR-QUTV by Algorithm 2 is rank revealing and prove the upper bound in terms of the spectral and Frobenius norms.

4.1.1 Rank Revealing Property

In the CoR-QUTV algorithm, the T factor is formed by the QRCP of D. The QRCP is numerically stable and D is compressed by A. We can analyze the factor T by (3.8) and rewrite D as

$$D\widetilde{P} = \widetilde{Q}\widetilde{R} = \widetilde{Q}\begin{pmatrix} \widetilde{R}_{11} & \widetilde{R}_{12} \\ O & \widetilde{R}_{22} \end{pmatrix}. \tag{4.1}$$

We need some properties [38] of the singular values of D to complete the rank-revealing property.

Theorem 4.1. *Let the matrix $A \in \mathbb{Q}^{m\times n}$ have an SVD as defined in (2.3), and $D = Q_1^* A Q_2 \in \mathbb{Q}^{l\times l}$ for any $Q_1^*Q_1 = Q_2^*Q_2 = I_l$. Then for $j = 1, 2, \ldots, l$, we have*

$$\sigma_j(D) \le \sigma_j. \tag{4.2}$$

Proof. Using the complex adjoint matrix of quaternion, we can have that $\chi_D = \chi_{Q_1}^* \chi_A \chi_{Q_2}$ where $\chi_{Q_1}^*\chi_{Q_1} = \chi_{Q_2}^*\chi_{Q_2} = I_{2l}$, and any singular value of A will appear twice in those of χ_A. The proof is similar to the case in the complex field $\mathbb{C}$ in [38]. □

Thus, for $D = Q_1^* A Q_2$ obtained from CoR-QUTV algorithm, if further there exist polynomials $p_1(m,n), p_2(m,n)$ such that

$$\frac{\sigma_k}{p_1(m,n)} \le \sigma_{\min}(\widetilde{R}_{11}) \le \sigma_k(D) \le \sigma_k, \tag{4.3}$$

$$\|A - Q_1 D Q_2^*\|_\zeta \le p_2(m,n)\sigma_{k+1}, \zeta = 2, F, \tag{4.4}$$

then we say $Q_1DQ_2^*$ is a rank-revealing approximation to A. Here we will reduce to handle with the relation of $\sigma_k(D)$ and the random quaternion Gaussian matrix $\Phi \in \mathbb{Q}^{n\times l}$. Considering the oversampling operation for the randomized algorithm mentioned in remark 2.2, the following proofs are based on the oversampled randomized algorithm. First, we assume that the sample size parameter l satisfies

$$2 \leq p+k \leq l, \tag{4.5}$$

where p is called an oversampling parameter [14]. Since Φ has affection on the right singular vectors V of A (3.4), i.e., we have

$$\widetilde{\Phi} = V_A^*\Phi = \begin{pmatrix} \widetilde{\Phi}_1^* & \widetilde{\Phi}_2^* \end{pmatrix}, \tag{4.6}$$

where $\widetilde{\Phi}_1$ and $\widetilde{\Phi}_2$ have $l-p$ and $n-l+p$ rows, respectively. The following theorem using a similar technique from [19] bounds $\sigma_k(D)$.

Theorem 4.2. *Suppose that the quaternion matrix A has an SVD defined in (2.3), $2 \leq p+k \leq l$ and the matrix D is formed by Algorithm 2. Moreover, suppose that $\widetilde{\Phi}_1$ is of full row rank, then we have*

$$\sigma_k \geq \sigma_k(D) \geq \frac{\sigma_k}{\sqrt{1+\|\widetilde{\Phi}_2\|_2^2\|\widetilde{\Phi}_1^\dagger\|_2^2(\frac{\sigma_{l-p+1}}{\sigma_k})^4}} \tag{4.7}$$

and when the quaternion matrix D is formed by Algorithm 2, where $\Phi_1^\dagger$ is formed by $\chi_{\Phi_1}^\dagger$, i.e., the power method is used in Algorithm 3, we have

$$\sigma_k \geq \sigma_k(D) \geq \frac{\sigma_k}{\sqrt{1+\|\widetilde{\Phi}_2\|_2^2\|\widetilde{\Phi}_1^\dagger\|_2^2(\frac{\sigma_{l-p+1}}{\sigma_k})^{4q+4}}}. \tag{4.8}$$

Here we introduce the property of Φ in [24].

Theorem 4.3. *[24] For $t \geq 1$, the quaternion random matrix $\Phi \in \mathbb{Q}^{m\times n}$ with $n-m \geq 1$. and fixed quaternion matrix S, T. Then* $\mathbf{E}\|S\Phi T\|_2 \leq 3(\|S\|_2\|T\|_F + \|T\|_2\|S\|_F)$

$$\begin{cases} \mathbf{P}\{\|\Phi^\dagger\|_F^2 > \frac{3m}{4(n-m+1)}t\} \leq t^{-2(n-m)}, \\ \mathbf{P}\{\|\Phi^\dagger\|_2 > \frac{e\sqrt{4n+2}}{4(n-m+1)}t\} \leq \frac{\pi^{-3}}{4(n-m+1)(2n-2m+3)}t^{-4(n-m+1)}, \end{cases} \tag{4.9}$$

and

$$\begin{cases} \mathbf{E}\|\Phi^\dagger\|_F^2 = \frac{m}{4(n-m)+2}, \\ \mathbf{E}\|\Phi^\dagger\|_2 \leq \frac{e\sqrt{4n+2}}{2(n-m)+2}. \end{cases} \tag{4.10}$$

Finally, since the random quaternion matrix Φ has the entries of the standard Gaussian distribution, the lower bound of the expectation on the k-th singular value of CoR-QUTV will be given in the following theorem, using Theorem 4.3 and a similar argument from [19].

Theorem 4.4. *With the notation of Theorem 4.2, and $\gamma_k = \frac{\sigma_{l-p+1}}{\sigma_k}$, for Algorithm 2, we have the expectation*

$$\mathbf{E}(\sigma_k(D)) \geq \frac{\sigma_k}{\sqrt{1+\nu^2\gamma_k^4}},$$

and for Algorithm 3, we have the power method estimation for the expectation

$$\mathbf{E}(\sigma_k(D)) \geq \frac{\sigma_k}{\sqrt{1+\nu^2\gamma_k^{4q+4}}},$$

where $\nu = \nu_1\nu_2, \nu_1 = 3\sqrt{n-l+p} + 3\sqrt{l}$ and $\nu_2 = \frac{e\sqrt{4l+2}}{2(p+1)}$.

Now we have completed the rank-revealing property of the CoR-QUTV algorithm.

4.1.2 Approximation Error

Our CoR-QUTV algorithm obviously provides a low-rank approximation of the quaternion matrix A. Utilizing the low-rank decomposition $A = A_k + A_0$ from (2.3) and the oversampling from the randomized algorithm mentioned in remark 2.2, the errors of these approximations can be restricted by the Frobenius and spectral norms. First, we state a theorem from [24].

Theorem 4.5. *Let the quaternion matrix A have the SVD in (2.3), and $Q_1 \in \mathbb{Q}^{m\times k}$ and $Q_2 \in \mathbb{Q}^{n\times k}$ be quaternion matrices with orthonormal columns constructed by means of the CoR-QUTV, where $1 \leq k \leq l$. Let D_k be the best rank-k of $D = Q_1^* A Q_2$, and due to the CoR-QUTV algorithm, we have $\widehat{A}_{\text{CoR}} = Q_1 D Q_2^*$ by (3.9). Then, for $\zeta = 2, F$, we have*

$$\begin{aligned}\|A - \widehat{A}_{\text{CoR}}\|_\zeta \leq & \|A - Q_1 D_k Q_2^*\|_\zeta \\ \leq & \|A_0\|_\zeta + \|A_k - Q_1 Q_1^* A_k\|_F + \|A_k - A_k Q_2 Q_2^*\|_F.\end{aligned} \tag{4.11}$$

Proof. It is a similar deduction to [18]. □

Having stated the connection between the CoR-QUTV and the real CoR-UTV, we now obtain upper bounds for the CoR-QUTV-based low-rank approximation error, based on similar techniques from [19].

Theorem 4.6. *Let the quaternion matrix A have the SVD as defined in (2.3), $2 \leq p+k \leq l$, and $\widehat{A}_{\text{CoR}}$ is computed by the CoR-QUTV by Algorithm 2. Furthermore, assume that $\widetilde{\Phi}_1$ is of full row rank. Then, for $\zeta = 2, F$, we have*

$$\|A - \widehat{A}_{\text{CoR}}\|_\zeta \leq \|A_0\|_\zeta + \sqrt{\frac{\alpha^2\|\widetilde{\Phi}_2\|_2^2\|\widetilde{\Phi}_1^\dagger\|_2^2}{1+\beta^2\|\widetilde{\Phi}_2\|_2^2\|\widetilde{\Phi}_1^\dagger\|_2^2}} + \sqrt{\frac{\eta^2\|\widetilde{\Phi}_2\|_2^2\|\widetilde{\Phi}_1^\dagger\|_2^2}{1+\tau^2\|\widetilde{\Phi}_2\|_2^2\|\widetilde{\Phi}_1^\dagger\|_2^2}} \tag{4.12}$$

where $\alpha = \sqrt{2k}\frac{\sigma_{l-p+1}^2}{\sigma_k}, \beta = \frac{\sigma_{l-p+1}^2}{\sigma_1\sigma_k}, \eta = \sqrt{2k}\sigma_{l-p+1}$ and $\tau = \frac{\sigma_{l-p+1}}{\sigma_1}$. When the power iteration is used by Algorithm 3, $\alpha = \sqrt{2k}\frac{\sigma_{l-p+1}^2}{\sigma_k}(\frac{\sigma_{l-p+1}}{\sigma_k})^{2q}, \beta = \frac{\sigma_{l-p+1}^2}{\sigma_1\sigma_k}(\frac{\sigma_{l-p+1}}{\sigma_k})^{2q}$, $\eta = \frac{\sigma_k}{\sigma_{l-p+1}}\alpha$ and $\tau = \frac{1}{\sigma_{l-p+1}}\beta$.

The random quaternion Gaussian matrix Φ has the standard Gaussian distribution in each real or imaginary part, we present the average error bounds on the CoR-QUTV-based low-rank approximation by the following theorem.

Theorem 4.7. *With the notation of Theorem 4.6, and $\gamma_k = \frac{\sigma_{l-p+1}}{\sigma_k}$, for the CoR-QUTV by Algorithm 2, we have the expectation*

$$\mathbf{E}\|A - \widehat{A}_{\text{CoR}}\|_\zeta \leq \|A_0\|_\zeta + \sqrt{2k}\nu\sigma_{l-p+1}, \tag{4.13}$$

and when the power method is used by Algorithm 3, we estimate the expectation

$$\mathbf{E}\|A - \widehat{A}_{\text{CoR}}\|_\zeta \leq \|A_0\|_\zeta + \sqrt{2k}\nu\sigma_{l-p+1}\gamma_k^{2q}, \tag{4.14}$$

where ν is defined in Theorem 4.4.

The discussion on the low-rank approximation error bounds for the CoR-QUTV algorithm is completed.

$\boxed{4.2}$ Error Analysis of Block-QUTV Algorithm

In this section, we will explore the connection between the Block-QUTV and the RSVD. The stepQUTV by Algorithm 4, which is the single step of Block-QUTV by Algorithm 5, can be demonstrated as the quaternion RSVD algorithm. This means that the error analysis of the quaternion RSVD algorithm in [24] is available in the stepQUTV algorithm.

Theorem 4.8. *Let the quaternion matrix $A \in \mathbb{Q}^{m\times n}$, k satisfy $1 \le k \le \min\{m,n\}$, and $q = 0, 1$ or 2. Let the random quaternion Gaussian matrix $G \in \mathbb{Q}^{m\times k}$, and let U, T and V be the factors in the factorization $A = UTV^*$ built in Algorithm 4, partitioned as in (3.12) and (3.13). We have the following results,*

1. *Let the quaternion sampling matrix $Y = (A^*A)^q A^*G$, and let $Q \in \mathbb{Q}^{n\times k}$ be a quaternion matrix with orthonormal columns constructed by the column space of Y. Then, the error $\|A - AQQ^*\|_\zeta$ precisely equals the error caused by the quaternion RSVD with q steps of power iteration, as analyzed in [24]. For $\zeta = 2, F$, it holds that*

$$\|A - AQQ^*\|_\zeta = \|A - U_1T_{11}V_1^*\|_\zeta = \left\|\begin{pmatrix} T_{12} \\ T_{22} \end{pmatrix}\right\|_\zeta . \tag{4.15}$$

2. *Let the quaternion sampling matrix $Z = AY = (AA^*)^{q+1}G$ and let $W \in \mathbb{Q}^{m\times k}$ be a quaternion matrix with orthonormal columns constructed by the column space of Z. If the rank of A is at least k, then*

$$\|A - WW^*A\|_\zeta = \|A - U_1(T_{11}V_1^* + T_{12}V_2^*)\|_\zeta = \|T_{22}\|_\zeta, \qquad \zeta = 2, F. \tag{4.16}$$

In part (2), we can observe that the term $\|A - WW^*A\|_\zeta$ is the error resulting from quaternion RSVD with "$q + \frac{1}{2}$" steps of power iteration.

Proof. The quaternion A can be represented as:

$$A = UTV^* = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} T_{11} & T_{12} \\ O & T_{22} \end{pmatrix} \begin{pmatrix} V_1^* \\ V_2^* \end{pmatrix} = U_1T_{11}V_1^* + U_1T_{12}V_2^* + U_2T_{22}V_2^*, \tag{4.17}$$

where U_1 and V_1 have k columns each, and $T_{11} \in \mathbb{Q}^{k\times k}$. With the help of (4.17), we can find that $QQ^* = V_1V_1^*$ in the claim (1) and, $\widehat{x}_{\mathrm{TLS}} = -V_{12}V_{22}^\dagger$. in the claim (2). We can prove the conclusion as follows,

1. With the help of Section 3.2, we can find that

$$QQ^* = \mathbb{V}_1\mathbb{V}_1^* = (V_1V_{\mathrm{small}}^*)(V_1V_{\mathrm{small}}^*)^* = V_1(V_{\mathrm{small}}^*V_{\mathrm{small}})V_1^* = V_1V_1^*. \tag{4.18}$$

Then using (4.17), we have

$$AQQ^* = AV_1V_1^* = U_1T_{11}V_1^*. \tag{4.19}$$

The first identity in (4.15) follows from (4.19) immediately. The second identity holds since (4.17) implies that $A - U_1T_{11}V^* = U_1T_{12}V^* + U_2T_{22}V_2^* = U \begin{pmatrix} T_{12} \\ T_{22} \end{pmatrix} V_2^*$, with U being unitary and V_2 being orthonormal.

2. We will prove two quaternion matrices $A\mathbb{V}_1$ and Z whose sizes are $m \times k$ have the same column spaces with probability 1. We know that $\mathbb{V}_1$ is formed by the quaternion QR factorization of Y, there must be some $R \in \mathbb{Q}^{k\times k}$ which is upper triangular, such that $\mathbb{V}_1 R = Y$. The assumption that A has rank at least k implies that R is invertible with probability 1. Consequently, $A\mathbb{V}_1 = AYR^{-1} = ZR^{-1}$, since $Z = AY$. With the definition of $\mathbb{U}$ that $\mathbb{U}_1\mathbb{U}_1^* = WW^*$. Since $U_1 = \mathbb{U}_1 U_{\text{small}}$, where U_{small} is unitary quaternion matrix, and using (4.17), we have
$$WW^*A = \mathbb{U}_1\mathbb{U}_1^*A = U_1U_1^*A = U_1T_{11}V_1^* + U_1T_{12}V_2^*, \tag{4.20}$$
which completes the first identity in (4.16). The second identity holds since (4.17) implies that $A - U_1(T_{11}V_1^* + T_{12}V_2^*) = U_2T_{22}V_2^*$, with U_2 and V_2 being orthonormal.

□

5 Numerical Examples for Quaternion Matrix Optimization

In this section, we give some examples to test the features of our quaternion UTV algorithms in quaternion matrix optimization. The following numerical examples are performed via MATLAB R2020a with machine precision $u = 2.22e - 16$ in a laptop with an Intel Core i5 CPU at 1.4 GHz and memory of 16 GB.

5.1 Low-Rank Approximation

Here we consider the quaternion matrix rank-k approximation problem:
$$\min_{\text{rank}(\widehat{A})\le k} \|A - \widehat{A}\|_\zeta. \tag{5.1}$$

Since both QUTV algorithms compute a rank-k approximation of a given quaternion matrix, we should investigate how accurate this approximation is. To make a fair comparison, we construct a rank-k approximation $\widehat{A}_{\text{out}}$ to A by each algorithm, and calculate the error:
$$e_k = \|A - \widehat{A}_{\text{out}}\|_\zeta \tag{5.2}$$
where $\zeta = F$ for the Frobenius-norm error, and $\zeta = 2$ for the spectral-norm error.

We construct our test quaternion matrix $A \in \mathbb{Q}^{n\times n}$ as $A = U\Sigma V^*$, where U,V are quaternion Householder matrices taking the form $U = I - 2uu^*, V = I - 2vv^*$, u, v are quaternion unit vectors, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a real $n \times n$ diagonal matrix. Set $n = 1000$ and we consider singular values with different decaying rates as

1. $\sigma_1 = 1, \sigma_{i+1}/\sigma_i = 0.9,$ for $i = 1, 2, \ldots, n - 1$,
2. $\sigma_1 = 1, \sigma_{i+1}/\sigma_i = 0.1,$ for $i = 1, 2, \ldots, n - 1$,

where in Case 1, for the threshold singular value $\sigma_{327} > 10^{-16} > \sigma_{328}$, the numerical rank of the matrix is 327, while in Case 2, for the threshold $\theta = 10^{-15}$, the numerical rank of the matrix is 16. For all of the randomized algorithms, we run the experiment without the power method $q = 0$ and with the power method $q = 2$. For all of the randomized algorithms considered, the results presented are averaged over 5 trials. For TSR-QSVD and CoR-QUTV, we set the oversampling parameter $p = k$. For QRSVD and Block-QUTV, we set the oversampling parameter $p = 5$. For all cases in Figure 1, CoR-QUTV strongly reveals the numerical rank k, as do the QSVD, QRSVD, TSR-QSVD and Block-QUTV.

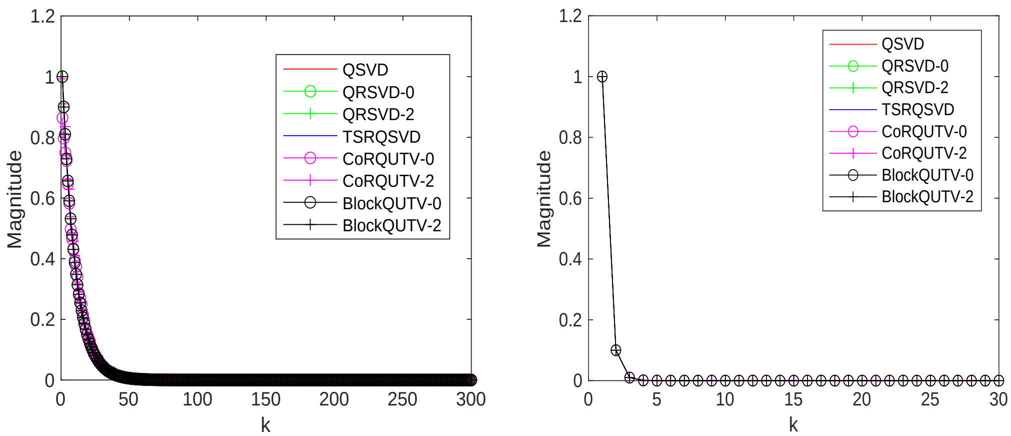

Figure 1: Comparison of singular values for quaternion SVD (QSVD), quaternion RSVD (QRSVD) with power method ($q = 0$ and 2), TSRQSVD, CoRQUTV with power method ($q = 0$ and 2), BlockQUTV with power method ($q = 0$ and 2). The left figures are the comparison in Case 1, while the right ones correspond to Case 2.

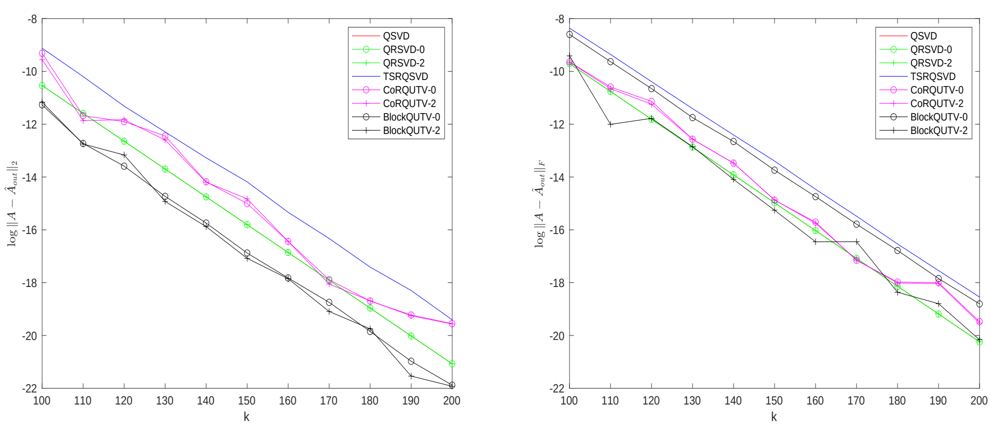

Figure 2: Comparison for quaternion SVD (QSVD), quaternion RSVD (QRSVD) without power method ($q = 0$), and $q = 2$, TSRQSVD, CoRQUTV without power method ($q = 0$), and $q = 2$, BlockQUTV with no power method ($q = 0$), and $q = 2$ in Case 1. The left figures are for the estimates of spectral-norm errors, while the right ones correspond to Frobenius-norm errors.

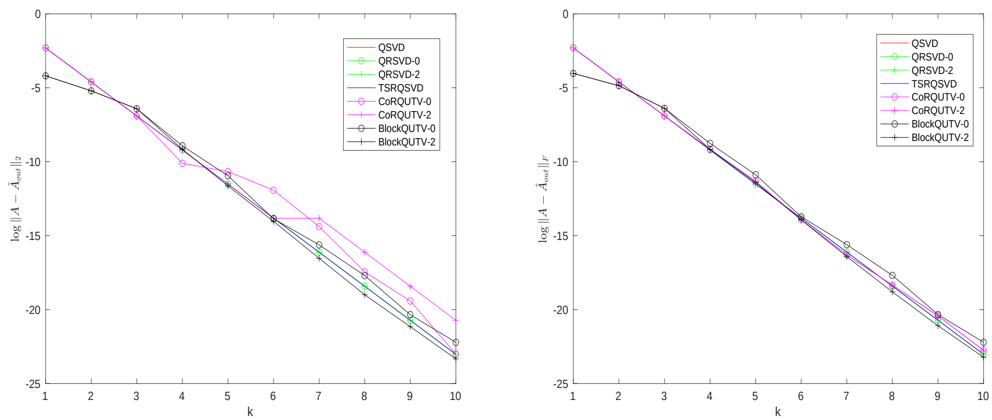

Figure 3: Comparison for quaternion SVD (QSVD), quaternion RSVD (QRSVD) without power method ($q = 0$), and $q = 2$, TSRQSVD, CoRQUTV with no power method ($q = 0$), and $q = 2$, BlockQUTV without power method ($q = 0$), and $q = 2$ in Case 2. The left figures are for the estimates of spectral-norm errors, while the right ones correspond to Frobenius-norm errors.

We choose two algorithms as baseline. A rank-k approximation for the QSVD [41], is computed as truncated svd and the rank-k QRSVD [24] is used for comparison. We choose different k for the low-rank approximation in QRSVD, TSRQSVD and CoRQUTV. The parameter k in BlockQUTV represents the size of block and we set the target rank $k = 30$ which can reduce the quantum of block and save the computing time in Case 2. In Figure 2, it represents the error in Case 1. The left figures are the the estimates of spectral-norm errors, while the right ones correspond to Frobenius-norm errors. The computational times of different factorization in Case 1 are listed in Table 1.

Table 1: Computational Times(s) of Different Factorization in Case (1)

k	100	120	140	160	180	200
QSVD	366.52	366.52	366.52	366.52	366.52	366.52
QRSVD (q=0)	16.37	16.28	16.57	17.06	17.92	23.57
QRSVD (q=2)	15.87	18.15	17.71	16.89	17.21	18.39
TSRQSVD	1.05	1.13	1.36	1.52	1.75	2.00
CoRQUTV (q=0)	1.29	1.03	1.17	1.36	1.54	1.77
CoRQUTV (q=2)	1.70	1.79	2.06	2.40	2.69	3.01
BlockQUTV (q=0)	28.62	26.90	26.84	23.13	21.33	23.94
BlockQUTV (q=2)	31.73	32.54	30.68	31.77	25.28	26.74

Figure 3, represents the error in Case 2. The left figures are the estimates of spectral-norm errors, while the right ones correspond to Frobenius-norm errors.

In Table 2, the computational times of different factorizations in Case 2 are listed. We make three observations: (1) CoRQUTV and BlockQUTV are both strongly reveal the numerical rank in two cases. (2) The performance of CoRQUTV algorithm is better than the TSRQSVD algorithm and the BlockQUTV algorithm show better performance than the QRSVD algorithm. Both QUTV algorithms are becoming close to optimal performance of

Table 2: Computational Times(s) of Different Factorization in Case (2)

k	2	4	6	8	10
QSVD	347.48	347.48	347.48	347.48	347.48
QRSVD (q=0)	15.53	14.80	16.71	14.84	17.92
QRSVD (q=2)	15.21	14.91	15.49	15.59	17.21
TSRQSVD	0.21	0.24	0.25	0.26	1.75
CoRQUTV (q=0)	0.20	0.21	0.25	0.24	1.54
CoRQUTV (q=2)	0.28	0.27	0.34	0.35	2.69
BlockQUTV (q=0)	1283.2	655.42	451.96	434.90	262.62
BlockQUTV (q=2)	1650.9	643.26	414.95	312.83	464.60

the QSVD. (3) TSRQSVD and CoRQUTV are faster than other algorithms. The costing time of BlockQUTV algorithm decreases with the growth of k and it takes about the same time as QRSVD in the proper range of k without losing accuracy.

5.2 Image Reconstruction

This section describes how to solve the robust principal component analysis (PCA) [2] problem using the proposed UTV method. Robust PCA represents an input low-rank quaternion matrix $M \in \mathbb{Q}^{m\times n}$ by solving the low-rank matrix decomposition-based completion problem which is formulated in the form of the following optimization problem,

$$\begin{cases} \text{minimize } \|L\|_* + \lambda\|S\|_1 \\ \text{subject to } L + S = M, \end{cases} \tag{5.3}$$

where L is a low rank matrix, S is a sparse matrix and $\|L\|_*$ is the nuclear norm. The efficient method to solve (5.3) is the method of augmented Lagrange multipliers, which minimizes the following augmented Lagrange function with variable L or S alternatively,

$$\mathcal{L}(L, S, Y, \mu) = \|L\|_* + \lambda\|S\|_1 + \langle Y, M - L - S\rangle + \frac{\mu}{2}\|M - L - S\|_F^2, \tag{5.4}$$

where $Y \in \mathbb{Q}^{m\times n}$ is the Lagrange multiplier matrix, and $\mu > 0$ is a penalty parameter. The robust PCA solved by the augmented Lagrange multipliers (ALM) method is given in Algorithm 6. In Algorithm 6, for any quaternion matrix A with an QSVD defined as $A = U_A\Sigma_A V_A^*$, $D_\delta(A)$ refers to a singular value thresholding operator defined as $\mathcal{D}_\delta(A) = U_A S_\delta(\Sigma_A)V_A^*$, where $\mathcal{S}_\delta(x) = \text{sgn}(x)\max(|x| - \delta, 0)$ is a shrinkage operator [13], and λ, μ, Y_0, and S_0 are initial values. The ALM method yields the optimal solution L^* and S^*, however,

Algorithm 6: Robust PCA by Alternating Directions

Input: Matrix $M, \lambda, \mu, Y_0 = S_0 = 0, j = 0$
Output: Low rank plus sparse matrix
while *the algorithm does not converge* **do**
  Compute $L_{j+1} = \mathcal{D}_{\mu^{-1}}(M - S_j + \mu^{-1}Y_j)$
  Compute $S_{j+1} = \mathcal{S}_{\lambda\mu^{-1}}(M - L_{j+1} + \mu^{-1}Y_j)$
  Compute $Y_{j+1} = Y_j + \mu(M - L_{j+1} - S_{j+1})$
Return L^* and S^*

its serious bottleneck is computing a computationally demanding QSVD at each iteration to approximate the low-rank component L of M. To address this issue, we thus, by retaining the original objective function proposed, apply the QUTV algorithm as a surrogate to the truncated QSVD to solve the robust PCA problem. We adopt the continuation technique, which increases μ in each iteration. The proposed method which is called ALM-QUTV is given in Algorithm 7. In Algorithm 7, for any matrix A having a QUTV decomposition described in Section 3, $\mathcal{C}_\delta(A)$ refers to a QUTV thresholding operator defined as,

$$\mathcal{C}_\delta(A) = U(:, 1:r)T(1:r, :)V^*, \tag{5.5}$$

where r is the number of diagonals of T greater than δ, and $\lambda, \mu_0, \bar{\mu}, \rho, Y_0$, and S_0 are initial values.

Algorithm 7: Robust PCA by ALM-QUTV

Input: Matrix $M, \lambda, \mu, Y_0 = S_0 = 0, j = 0$
Output: Low rank plus sparse matrix
while *the algorithm does not converge* **do**
  Compute $L_{j+1} = \mathcal{C}_{\mu^{-1}}(M - S_j + \mu^{-1}Y_j)$
  Compute $S_{j+1} = \mathcal{S}_{\lambda\mu^{-1}}(M - L_{j+1} + \mu^{-1}Y_j)$
  Compute $Y_{j+1} = Y_j + \mu(M - L_{j+1} - S_{j+1})$
Return L^* and S^*

We use standard test images with 512×512 pixels . This color image is characterized by a 512×512 pure quaternion matrix [34] A with entries $A_{ij} = R_{ij}i + G_{ij}j + B_{ij}k$, where R_{ij}, G_{ij}, B_{ij} represent the red, green, and blue pixel values at the location (i, j) in the image, respectively. The parameters $\mu = 0.02$ in all algorithms. We set the approximation error to be $7e - 4$ and the maximum number of iterations is 300. Figure 4 shows that the BlockQUTV algorithm a comparable visual effect effect as the QRSVD algorithm. The CoRQUTV algorithm has the best visual effect on color image reconstruction.

To demonstrate the excellent performance of our proposed randomized QUTV algorithms in image reconstruction, Table 3 provides an acceptable peak signal-to-noise ratio (PNSR) to demonstrate the algorithm's performance, where PSNR is represented by

$$PSNR(\hat{A}_{out}, A) = 20 \log_{10} \frac{3mn}{\|\hat{A}_{out} - A\|_F}. \tag{5.6}$$

where $\hat{A}_{out}$ represents the data restored by different algorithms, and $m = n = 512$ in this case. In table 3, we can see that except for the second image, the randomized QUTV algorithms have better restoration performance, with CoRQUTV generally having better restoration performance than BlockQUTV. Table 4 shows the time required for different algorithms in image reconstruction. It can be seen that although QUTV decomposition has an advantage in speed in single rank-k approximation in previous experiments, it has lost some accuracy and increased the number of iterations. However, overall, the QUTV algorithm has a significant effect as an alternative to the QSVD algorithm.

5.3 Application in Signal Denoising

While solving the signal denoising optimization, we can still find the application of the QUTV algorithm. First, we decompose the spatial signal into three dimensional vector signals.

https://sipi.usc.edu/database/database.php?volume=misc

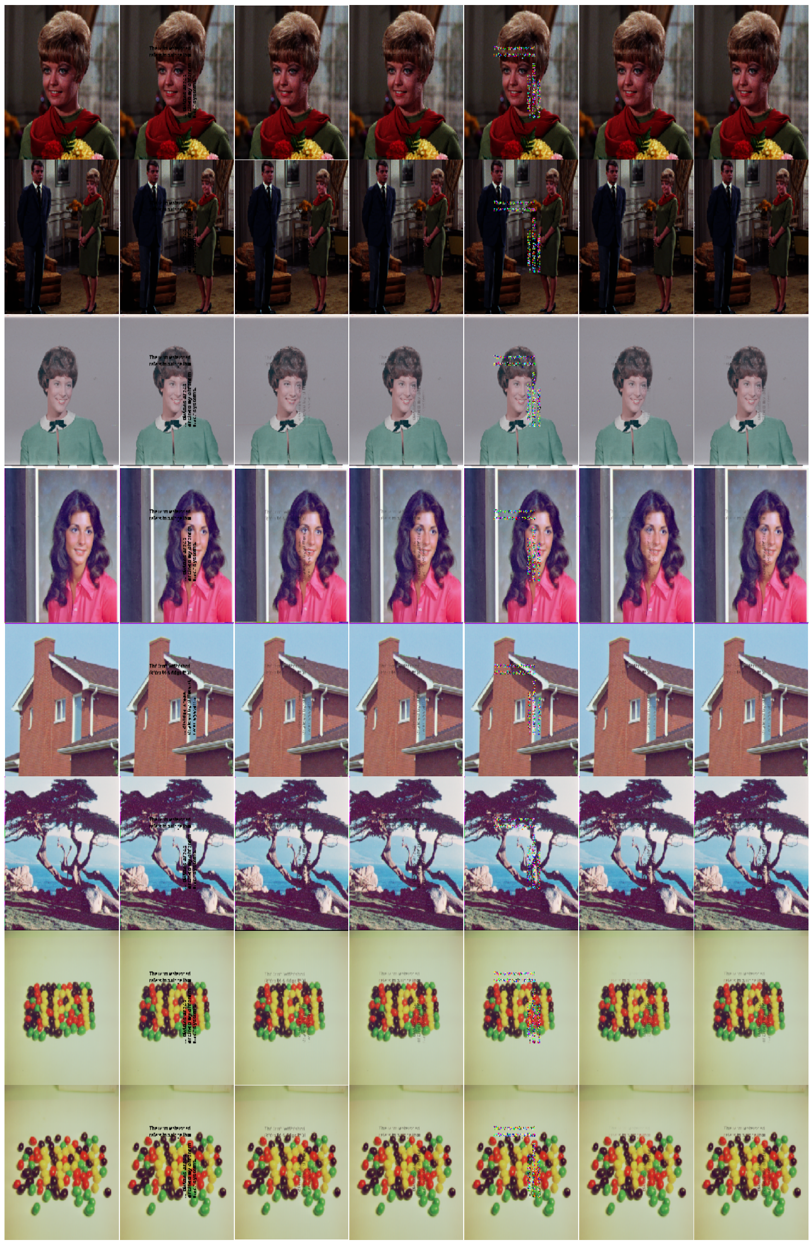

Figure 4: Comparison of low-rank image reconstruction. The first column is the original image and the second column is the observed image. The 3rd to the 8th are the complete results of QSVD, QRSVD, TSRQSVD, CoRQUTV, and BlockQUTV, respectively.

Table 3: PSNR results for low-rank image reconstruction.

QSVD	QRSVD	TSRQSVD	CoRQUTV	BlockQUTV
94.23	94.23	-4.19	**95.89**	95.48
91.75	91.74	6.45	91.42	91.70
88.51	88.49	-2.42	**96.77**	90.08
79.08	79.05	1.81	**90.72**	82.93
72.15	72.14	-1.25	**75.91**	73.83
69.61	69.50	0.50	**71.48**	70.16
75.63	75.17	1.27	**84.71**	76.05
74.41	74.38	-15.00	**83.05**	75.17

Table 4: Time comparison for low-rank image reconstruction.

QSVD	QRSVD	TSRQSVD	CoRQUTV	BlockQUTV
3279.26	72.17	**28.03**	49.83	80.03
3459.61	129.13	**46.13**	55.8	91.43
3782.48	**29.73**	36.81	52.53	43.79
3387.73	53.75	**42.85**	52.72	55.45
3569.16	**32.13**	39.83	54.71	38.77
3469.18	33.33	**32.95**	48.56	35.56
3091.78	24.75	**23.55**	49.44	27.67
3405.91	**30.16**	35.28	42.81	32.11

Then, we transform the vector signal into a Hankel matrix and construct a pure quaternion matrix by the Hankel matrices. Finally, the problem of signal denoising is transformed into a rank-k approximation problem,

$$\min_{\mathrm{rank}(\widehat{A})\leq k} \|A-\widehat{A}\|_2. \tag{5.7}$$

The Lorentz attractor [37] is a three-dimensional nonlinear system that is used in atmospheric turbulence. The model is a system of three ordinary differential equations now known as the Lorenz equations,

$$\frac{\partial x}{\partial t}=\sigma(y-x),\quad \frac{\partial y}{\partial t}=x(\rho-z)-y,\quad \frac{\partial z}{\partial t}=xy-\beta z. \tag{5.8}$$

where $\sigma,\rho,\beta>0$. For the chaotic behavior of the Lorenz attractor, we set $\sigma=10,\rho=28$, and $\beta=8/3$. In our experiment, the Lorentz system (5.8) is solved by the built-in function $ode45(f(t,[x,y,z]),[0,40],[12,4,0])$ in MATLAB, where $x,y,z\in\mathbb{R}^{2001}$. For convenience, we take part of the solutions of the Lorenz system $x_1=x(401:800), x_2=y(401:800), x_3=z(401:800)$ as three true signals. We add white Gaussian noise $y_1=awgn(x_1,snr), y_2=awgn(x_2,snr), y_3=awgn(x_3,snr)$ by the function $awgn$ in MATLAB, where the signal-to-noise ratio parameter is set to be $snr=5$. For a vector signal $x=(x(1),x(2),\dots,x(N))^\top$, we can construct a Hankel matrix as follows,

$$X=\begin{pmatrix} x(1) & x(2) & \cdots & x(s)\\ x(2) & x(3) & \cdots & x(s+1)\\ \vdots & \vdots & \ddots & \vdots\\ x(s) & x(s+1) & \cdots & x(N)\end{pmatrix}, \tag{5.9}$$

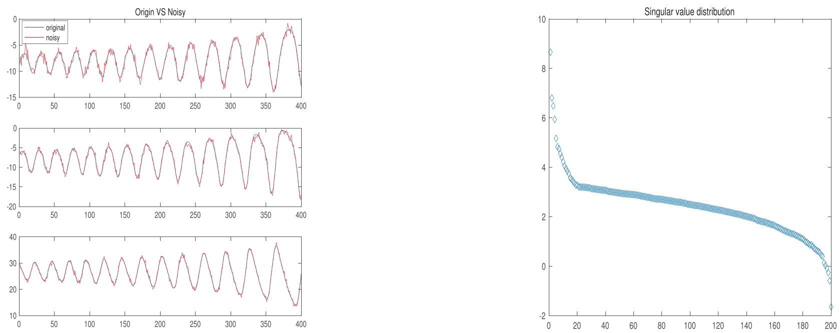

Figure 5: Original signal and noisy signal.

where $s = \lfloor N/2 \rfloor$. Similarly, we can construct three Hankel matrices X, Y, Z by vector signals x, y, and z, respectively. Then we obtain a pure quaternion noisy signal matrix [34]

$$A = Xi + Yj + Zk. \tag{5.10}$$

In Figure 5(a), we can see the original and noisy signals. We can see the singular values of the noisy signal under logarithmic scale in Figure 5(b). There are few large singular values and other singular values decrease slowly. Therefore, we choose the target rank $k = 20$ in our QUTV algorithms. We denoise the noisy signal by the CoRQUTV and the BlockQUTV in Figure 6(a) and Figure 6(b), respectively.

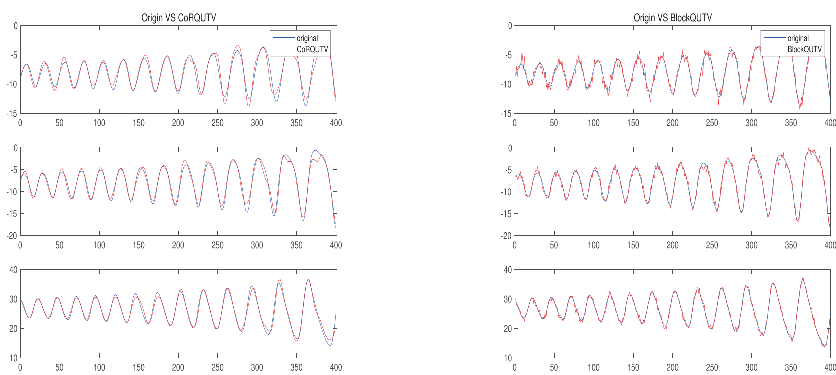

Figure 6: Recovered signal.

$\boxed{5.4}$ Total Least Squares for Signal Processing

The vector-sensor signals can be represented as quaternionic signals. The polarization waves are one kind of quaternionic signal [20]. For the complex number $z = a + bi$, the argument or phase angle is defined as $\operatorname{atan2}(b, a)$. If $z = q_0 + q_1 i + q_2 j + q_3 k$ is written it the form $z = \|z\|_2 e^{i\gamma}$, then γ is the phase (argument) of z, denoted $\arg(z) = \gamma$. Quaternions contain three complex subfields and, correspondingly, three phases components which are

$$\begin{cases} \phi = \operatorname{atan2}(n_\phi, d_\phi), \\ \theta = \operatorname{atan2}(n_\theta, d_\theta), \\ \psi = \arcsin(n_\psi), \end{cases} \tag{5.11}$$

where $n_\phi, d_\phi, n_\theta, d_\theta, n_\psi$ come from the equivalence of two homomorphic quaternion transformations and are defined as

$$\begin{cases} n_\phi = 2(q_2q_3 + q_0q_1), \\ d_\phi = q_0^2 - q_1^2 + q_2^2 - q_3^2, \\ n_\theta = 2(q_1q_3 + q_0q_2), \\ d_\theta = q_0^2 + q_1^2 - q_2^2 - q_3^2, \\ n_\psi = 2(q_1q_2 + q_0q_3). \end{cases} \tag{5.12}$$

Now we consider a set of N vector sensors, the collected vector data set S can be written as a matrix whose rows are the signals recorded on vector sensors that constitute the array $S = (s_1(m), s_2(m), \ldots, s_N(m))^\top$. The set of signals recorded on the vector-sensor array, S, is a matrix of size $N \times M$ which elements are quaternions ($S \in \mathbb{Q}^{N\times M}$). Using the quaternionic signal representation, any polarized signal is expressed as

$$s_n(m) = \rho_n(m)e^{\mu_n(m)}. \tag{5.13}$$

This is to say that its magnitude is, at time sample m, equal to $\rho(m)$, and the distribution of magnitude on the three components are carried by the eigenangle $\mu_n(m)$. Total least squares [10] (TLS) is a useful method for fitting a model to a set of data points when both the predictor (explanatory) variables and the response (dependent) variables are subject to error. This can be important in signal processing, as it is often the case that measurements of a signal are subject to some degree of uncertainty or that there are errors in the model itself. The total least squares approximation $\widehat{x}_{\text{TLS}}$ is obtained as a solution of the optimization problem,

$$\begin{cases} \widehat{x}_{\text{TLS}} = & \underset{\widehat{A},\widehat{b}}{\operatorname{argmin}} \left\| \begin{pmatrix} \widehat{A} & \widehat{b} \end{pmatrix} - \begin{pmatrix} A & b \end{pmatrix} \right\|_F \\ & \text{subject to } \widehat{A}x = \widehat{b} \end{cases} \tag{5.14}$$

where $A \in \mathbb{Q}^{m\times n}, x \in \mathbb{Q}^n, b \in \mathbb{Q}^m$ and $Ax \approx b$ is an overdetermined system ($m \geq n$). Here we take the QUTV algorithm as one of the complete orthogonal decomposition methods to deal with TLS problem. We suppose the $C = \begin{pmatrix} A & b \end{pmatrix}$ has a QUTV decomposition,

$$C = U \begin{pmatrix} T_{11} & T_{12} \\ O & T_{22} \end{pmatrix} V^*, \tag{5.15}$$

where U, V are quaternion unitary matrices, $T \in \mathbb{Q}^{m\times m}$ is nonsingular upper triangular matrix. We can rewrite V in the block form

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}, \tag{5.16}$$

where $V_{11} \in \mathbb{Q}^{p\times p}$. Then we can have the TLS solution [39] $\widehat{x}_{\text{TLS}} = -V_{12}V_{22}^\dagger$. Now we try an artificial signal example to verify our QUTV algorithm. First, we construct quaternionic signal matrix $A = USV^* \in \mathbb{Q}^{m\times n}$ by using quaternion Householder matrix U, V and $S = \operatorname{diag}(1, \sqrt{2}, \sqrt{3}, \ldots, \sqrt{n})$. Then we construct $x = (x(1), x(2), \ldots, x(n))^\top$, where $x(p) = 1/p + (1/p)i + (1/p)j + (1/p)k, p = 1, 2, \ldots, n$. We add the Gaussian noise to A and $b = Ax \in \mathbb{Q}^m$ with $\eta = 0.03$. In our experiment, we set $m = 100$ and $n = 50$. Because of the previous introduction of the total least squares method, we can only apply the BlockQUTV structure to solve this problem and set the parameter $k = 10$. In Figure 7, we can see the performance of the BLockQUTV algorithm under the total least squares method in the artificial quaternionic signal.

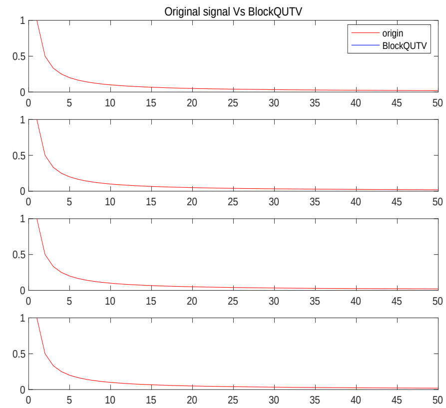

Figure 7: Original signal VS BlockQUTV recovery.

6 Conclusion Remarks

Quaternion matrix optimization problems have garnered increasing attention in the fields of color image processing and signal processing. In these applications, the QSVD plays a crucial role. This paper introduces randomized quaternion UTV decompositions as an alternative to QSVD. Experimental results demonstrate that the proposed randomized QUTV algorithms are both effective and efficient in addressing quaternion matrix optimization problems.

Acknowledgments

The authors are grateful to the handling editor and two anonymous referees for their useful comments and suggestions, which greatly improved the original presentation.

References

[1] A. Bunse-Gerstner, R. Byers and V. Mehrmann, A quaternion QR algorithm, *Numerische Mathematik* 55 (1989) 83–95.

[2] E.J. Candes, X. Li, Y. Ma and J. Wright, Robust principal component analysis, *Journal of the ACM* 58 (2011) 1–37.

[3] T.F. Chan, Rank revealing QR factorizations, *Linear Algebra and its Applications* 88 (1987) 67–82.

[4] J.-H. Chang, J.-J. Ding, et al., Quaternion matrix singular value decomposition and its applications for color image processing, in: *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, vol.1, IEEE, pages I-805.

[5] M. Che and Y. Wei, An efficient algorithm for computing the approximate t-URV and its applications, *Journal of Scientific Computing* 92 (2022): Article 93.

[6] Y. Chen, L. Qi and X. Zhang, Color image completion using a low-rank quaternion matrix approximation, *Pacific Journal of Optimization* 18 (2022) 55–75.

[7] J. Demmel, I. Dumitriu and O. Holtz, Fast linear algebra is stable, *Numerische Mathematik* 108 (2007) 59–91.

[8] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[9] T.A. Ell, N. Le Bihan and S.J. Sangwine, *Quaternion Fourier Transforms for Signal and Image Processing*, John Wiley & Sons, 2014.

[10] G.H. Golub and C.F. Van Loan, *Matrix Computations*, JHU press, 4th edition, 2013.

[11] M. Gu and S.C. Eisenstat, Efficient algorithms for computing a strong rank-revealing QR factorization, *SIAM Journal on Scientific Computing* 17 (1996) 848–869.

[12] R. Gutin, Generalizations of singular value decomposition to dual-numbered matrices, *Linear and Multilinear Algebra* 70 (2020) 5107–5114.

[13] E.T. Hale, W. Yin and Y. Zhang, Fixed-point continuation for l_1-minimization: Methodology and convergence, *SIAM Journal on Optimization* 19 (2008) 1107–1130.

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

[15] W.R. Hamilton, On quaternions; or on a new system of imaginaries in algebra, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 33 (1848) 58–60.

[16] Z. Jia, *The Eigenvalue Problem of Quaternion Matrix: Structure-Preserving Algorithms and Applications*, Science Press, Beijing, 2019.

[17] Z. Jia and M.K. Ng, Structure preserving quaternion generalized minimal residual method, *SIAM Journal on Matrix Analysis and Applications* 42 (2021) 616–634.

[18] M.F. Kaloorazi and R.C. de Lamare, Compressed randomized UTV decompositions for low-rank matrix approximations, *IEEE Journal of Selected Topics in Signal Processing* 12 (2018) 1155–1169.

[19] M.F. Kaloorazi and R.C de Lamare, Subspace-orbit randomized decomposition for low-rank matrix approximations, *IEEE Transactions on Signal Processing* 66 (2018) 4409–4424.

[20] N. Le Bihan and J. Mars, Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing, *Signal Processing* 84 (2004) 1177–1199.

[21] N. Le Bihan and S.J. Sangwine, Quaternion principal component analysis of color images, in: *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, vol.1, IEEE, 2003, pages I-809.

[22] Y. Li, M. Wei, F. Zhang and J. Zhao, Real structure-preserving algorithms of householder based transformations for quaternion matrices, *Journal of Computational and Applied Mathematics* 305 (2016) 82–91.

[23] C. Ling, H. He and L. Qi, Singular values of dual quaternion matrices and their low-rank approximations, *Numerical Functional Analysis and Optimization* 43 (2022) 1423–1458.

[24] Q. Liu, S. Ling and Z. Jia, Randomized quaternion singular value decomposition for low-rank matrix approximation, *SIAM Journal on Scientific Computing* 44 (2022), A870–A900.

[25] P.-G. Martinsson and J.A. Tropp, Randomized numerical linear algebra: Foundations and algorithms, *Acta Numerica* 29 (2020) 403–572.

[26] P.-G. Martinsson, G. Quintana-Orti and N. Heavner, Randutv: A blocked randomized algorithm for computing a rank-revealing utv factorization, *ACM Transactions on Mathematical Software* 45 (2019) 1–26.

[27] T. Minemoto, T. Isokawa, H. Nishimura and N. Matsui, Feed forward neural network with random quaternionic neurons, *Signal Processing* 136 (2017) 59–68.

[28] L. Qi, Standard dual quaternion optimization and its applications in hand-eye calibration and slam, *Communications on Applied Mathematics and Computation* 5 (2023) 1469–1483.

[29] L. Qi, C. Ling and H. Yan, Dual quaternions and dual quaternion vectors, *Communications on Applied Mathematics and Computation* 4 (2022) 1494–1508.

[30] L. Qi, Z. Luo, Q.-W. Wang and X. Zhang, Quaternion matrix optimization: motivation and analysis, *Journal of Optimization Theory and Applications* 193 (2022) 621–648.

[31] V. Rokhlin, A. Szlam and M. Tygert, A randomized algorithm for principal component analysis, *SIAM Journal on Matrix Analysis and Applications* 31 (2010) 1100–1124.

[32] S.J. Sangwine and N. Le Bihan, Quaternion singular value decomposition based on bidiagonalization to a real or complex matrix using quaternion Householder transformations, *Applied Mathematics and Computation* 182 (2006) 727–738.

[33] L. Shi, Exploration in quaternion colour, PhD thesis, School of Computing Science-Simon Fraser University, 2005.

[34] G. Song, W. Ding and M.K. Ng, Low rank pure quaternion approximation for pure quaternion matrices, *SIAM Journal on Matrix Analysis and Applications* 42 (2021) 58–82.

[35] G.W. Stewart, Updating a rank-revealing ULV decomposition, *SIAM Journal on Matrix Analysis and Applications* 14 (1993) 494–499.

[36] G.W. Stewart, An updating algorithm for subspace tracking, *IEEE Transactions on Signal Processing* 40 (1992) 1535–1541.

[37] S.H. Strogatz, *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering*, CRC Press, 2018.

[38] R.C. Thompson, Principal submatrices ix: Interlacing inequalities for singular values of submatrices, *Linear Algebra and its Applications* 5 (1972) 1–12.

[39] S. Van Huffel, *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling*, SIAM, Philadelphia, 1997.

[40] R. Xu, T. Wei, Y. Wei and H. Yan, UTV decomposition of dual matrices and its applications, *Computational and Applied Mathematics* 43 (2024): Article 41.

[41] F. Zhang, Quaternions and matrices of quaternions, *Linear Algebra and its Applications* 251 (1997) 21–57.

[42] F. Zhang and Y. Wei, Jordan canonical form of a partitioned complex matrix and its application to real quaternion matrices, *Communications in Algebra* 29 (2001) 2363–2375.

Manuscript received 10 January 2023
revised 1 June 2023
accepted for publication 9 August 2023

Renjie Xu
School of Mathematical Sciences, Fudan University
Shanghai, 200086, P.R. China.
E-mail address: rjxu17@fudan.edu.cn

Yimin Wei
School of Mathematical Sciences and
Shanghai Key Laboratory of Contemporary
Applied Mathematics, Fudan University
Shanghai, 200086, P.R. China
E-mail address: ymwei@fudan.edu.cn