Applied Numerical Mathematics 147 (2020) 101–117

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

A structure-preserving one-sided Jacobi method for computing the SVD of a quaternion matrix

Ru-Ru Ma [a,1], Zheng-Jian Bai [b,*,2]

[a] School of Mathematical Sciences, Xiamen University, Xiamen 361005, People's Republic of China

[b] School of Mathematical Sciences and Fujian Provincial Key Laboratory on Mathematical Modeling & High Performance Scientific Computing, Xiamen University, Xiamen 361005, People's Republic of China

ARTICLE INFO

Article history:
Received 31 March 2019
Received in revised form 29 May 2019
Accepted 19 August 2019
Available online 27 August 2019

Keywords:
Quaternion matrix
Singular value decomposition
One-sided cyclic Jacobi method

ABSTRACT

In this paper, we propose a structure-preserving one-sided cyclic Jacobi method for computing the singular value decomposition of a quaternion matrix. In our method, the columns of the quaternion matrix are orthogonalized in pairs by using a sequence of orthogonal JRS-symplectic Jacobi matrices to its real counterpart. We establish the quadratic convergence of our method specially. We also give some numerical examples to illustrate the effectiveness of the proposed method.

1. Introduction

The concept of quaternions was originally introduced by Hamilton [9,10]. Quaternions and quaternion matrices arise in various applications in applied science such as quaternionic quantum mechanics [1,3,6], color image processing [12,15,16, 21,28,36], and field theory [23], etc. The quaternion matrix singular value decomposition (QSVD) was studied theoretically in 1997 by Zhang [38]. Recently, the QSVD has been an important tool in many applications such as color image processing [7,26], signal processing [20,37], and electroencephalography [5], etc. In particular, in [5], a QSVD analysis was used to sleep analysis, which involved quaternion operations. In [7], a color-image-denoising algorithm was proposed by using QSVD where only complex operations were involved. In [26], a QSVD-based blind watermarking with quaternion operations was applied to color images. In [37], a convex optimization based QSVD method was proposed for fault diagnosis of rolling bearing.

Various numerical methods have been proposed for computing the QSVD. In [19,20,28], some algorithms were provided to calculate the singular value decomposition (SVD) of a quaternion matrix via utilizing its equivalent complex matrix. In [31], Sangwine and Le Bihan proposed a method for computing the QSVD based on bidiagonalization via quaternionic Householder transformations. In [22], Le Bihan and Sangwine gave an implicit Jacobi algorithm for computing the QSVD where the quaternion arithmetic was employed instead of a complex equivalent representation. In [4], Doukhnitch and Ozen presented a coordinate rotation digital computer algorithm for computing the QSVD. In [13], Jia et al. gave a Lanczos-based method for calculating some dominant SVD triplets of a large-scale quaternion matrix.

* Corresponding author.
E-mail addresses: maruru7271@126.com (R.-R. Ma), zjbai@xmu.edu.cn (Z.-J. Bai).

[1] The research of this author is partially supported by the Fundamental Research Funds for the Central Universities (No. 20720180008).

[2] The research of this author is partially supported by the National Natural Science Foundation of China (No. 11671337) and the Fundamental Research Funds for the Central Universities (No. 20720180008).

https://doi.org/10.1016/j.apnum.2019.08.017

In this paper, we propose a structure-preserving one-sided cyclic Jacobi method for computing the QSVD. This is motivated by the recent structure-preserving methods related to quaternion matrices. In particular, in [11], Jia et al. designed a real structure-preserving method for quaternion Hermitian eigenvalue problems by using the structure-preserving tridiagonalization of the real counterparts for quaternion Hermitian matrices. In [24,25], Li et al. presented a structure-preserving algorithm for computing the QSVD, which used the structure-preserving bidiagonalization of the real counterpart of quaternion matrices via Householder-based transformations. In [27], Ma et al. proposed a structure-preserving Jacobi algorithm for quaternion Hermitian eigenvalue problems. In this paper, the columns of a rectangular quaternion matrix are orthogonalized in pairs by using a sequence of orthogonal JRS-symplectic Jacobi matrices to its real counterpart. When the updated quaternion matrix has sufficiently orthogonal columns, the SVD is obtained by column scaling. Specifically, we show the quadratic convergence of the proposed structure-preserving one-sided cyclic Jacobi method. Finally, we give some numerical examples to show that our method is effective for computing the QSVD. There exist some differences between our method and the Lanczos QSVD in [13]. For an m-by-n quaternion matrix, the Lanczos QSVD only computes the k dominant SVD triplets, the total computational cost is $O(k\hat{t}n^2)$ quaternion operations, and the convergence analysis is not provided, where $\hat{t}$ means the frequency of restarting the Lanczos bidiagonalization, while our method computes the complete QSVD, the computational complexity is of $O(mn^2)$ real flops for each sweep, and the quadratic convergence is established. As argued heuristically in [2], the number of sweeps is of $O(\log(n))$.

The rest of this paper is organized as follows. In section 2 we give necessary preliminaries used in this paper. In section 3 we propose a structure-preserving one-sided cyclic Jacobi algorithm for computing the QSVD and we also derive its quadratic convergence. In section 4 we present some numerical examples to indicate the effectiveness of our algorithm. Finally, some concluding remarks are given in section 5.

2. Preliminaries

In this section, we briefly review some necessary definitions and properties of quaternions and quaternion matrices. For more information on quaternions, one may refer to [18,38] and references therein.

Throughout this paper, we need the following notation. Let $\mathbb{R}$ and $\mathbb{R}^{m\times n}$ be the set of all real numbers and the set of all $m\times n$ real matrices, respectively. Let $\mathbb{H}$ and $\mathbb{H}^{m\times n}$ be the set of all quaternions and the set of all $m\times n$ quaternion matrices, respectively. Let I_n be the identity matrix of order n. Let $A^T, \bar{A}$ and A^* stand for the transpose, conjugate and conjugate transpose of a matrix A accordingly. $\|\cdot\|_F$ means the Frobenius matrix norm.

A quaternion $a\in\mathbb{H}$ takes the form of

$$a = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k},$$

where $a_0, a_1, a_2, a_3 \in \mathbb{R}$ and the quaternion units $\mathbf{i}, \mathbf{j}, \mathbf{k}$ satisfy the following rules

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1, \quad \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \quad \mathbf{ki} = -\mathbf{ik} = \mathbf{j}, \quad \mathbf{ij} = -\mathbf{ji} = \mathbf{k}.$$

The conjugate of $a\in\mathbb{H}$ is given by $\bar{a} = a^* = a_0 - a_1\mathbf{i} - a_2\mathbf{j} - a_3\mathbf{k}$. For two quaternions $a = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} \in \mathbb{H}$ and $b = b_0 + b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k} \in \mathbb{H}$, their product (i.e., the Hamilton product) is given by

$$\begin{aligned} ab = &a_0b_0 - a_1b_1 - a_2b_2 - a_3b_3 \\ &+(a_0b_1 + a_1b_0 + a_2b_3 - a_3b_2)\mathbf{i} \\ &+(a_0b_2 - a_1b_3 + a_2b_0 + a_3b_1)\mathbf{j} \\ &+(a_0b_3 + a_1b_2 - a_2b_1 + a_3b_0)\mathbf{k}. \end{aligned}$$

The modulus $|a|$ of $a = a_0 + a_1 i + a_2 j + a_3 k \in \mathbb{H}$ is defined by

$$|a| = \sqrt{a_0^2 + a_1^2 + a_2^2 + a_3^2} = \sqrt{a\bar{a}} = \sqrt{\bar{a}a}.$$

The multiplicative inverse of any nonzero quaternion $0 \neq a \in \mathbb{H}$ is given by $a^{-1} = \bar{a}/|a|^2$. The Hamilton product is not commutative but associative and thus $\mathbb{H}$ is an associative division algebra over $\mathbb{R}$.

For an $n\times n$ quaternion matrix $A = (a_{pq})$, we define

$$\mathrm{off}(A) := \sqrt{\sum_{p=1}^{n}\sum_{\substack{q=1\\ q\neq p}}^{n} |a_{pq}|^2},$$

where $|a_{pq}|$ denotes the modulus of a_{pq}.

Suppose $A = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{H}^{m\times n}$ is a quaternion matrix, where $A_0, A_1, A_2, A_3 \in \mathbb{R}^{m\times n}$. A real counterpart of A is defined by

$$\Gamma_A = \begin{bmatrix} A_0 & A_2 & A_1 & A_3 \\ -A_2 & A_0 & A_3 & -A_1 \\ -A_1 & -A_3 & A_0 & A_2 \\ -A_3 & A_1 & -A_2 & A_0 \end{bmatrix}. \quad (2.1)$$

Next, we recall the definitions of JRS-symmetry and JRS-symplecticity [11]. Let

$$J_n = \begin{bmatrix} 0 & 0 & -I_n & 0 \\ 0 & 0 & 0 & -I_n \\ I_n & 0 & 0 & 0 \\ 0 & I_n & 0 & 0 \end{bmatrix}, \ R_n = \begin{bmatrix} 0 & -I_n & 0 & 0 \\ I_n & 0 & 0 & 0 \\ 0 & 0 & 0 & I_n \\ 0 & 0 & -I_n & 0 \end{bmatrix}, \ S_n = \begin{bmatrix} 0 & 0 & 0 & -I_n \\ 0 & 0 & I_n & 0 \\ 0 & -I_n & 0 & 0 \\ I_n & 0 & 0 & 0 \end{bmatrix}.$$

A matrix $\Omega \in \mathbb{R}^{4n\times 4n}$ is called JRS-symmetric if $J_n \Omega J_n^T = \Omega$, $R_n \Omega R_n^T = \Omega$ and $S_n \Omega S_n^T = \Omega$. A matrix $\Omega \in \mathbb{R}^{4n\times 4n}$ is called JRS-symplectic if $\Omega J_n \Omega^T = J_n$, $\Omega R_n \Omega^T = R_n$ and $\Omega S_n \Omega^T = S_n$. A matrix $\Omega \in \mathbb{R}^{4n\times 4n}$ is called orthogonal JRS-symplectic if it is orthogonal and JRS-symplectic.

In the rest of this section, we recall some basic results on the relationship between a quaternion matrix and its real counterpart. First, we have the following properties of the real counterparts of quaternion matrices [11,17,33].

Lemma 2.1. *Let $F, G \in \mathbb{H}^{m\times n}$, $H \in \mathbb{H}^{n\times s}$, $W \in \mathbb{H}^{n\times n}$, $\alpha \in \mathbb{R}$. Then*

(1) $\Gamma_{F+G} = \Gamma_F + \Gamma_G$; $\Gamma_{\alpha G} = \alpha\Gamma_G$; $\Gamma_{GH} = \Gamma_G\Gamma_H$.
(2) $\Gamma_{G^*} = \Gamma_G^T$.
(3) *Γ_W is JRS-symmetric.*
(4) *W is unitary if Γ_W is orthogonal.*
(5) *If Γ_G is orthogonal, then it is also orthogonal JRS-symplectic.*

On the SVD of a quaternion matrix, we have the following result [38, Theorem 7.2].

Lemma 2.2. *Let $A \in \mathbb{H}^{m\times n}$ be a quaternion matrix with $\operatorname{rank}(A) = r$. Then there exist unitary quaternion matrices $U \in \mathbb{H}^{m\times m}$ and $V \in \mathbb{H}^{n\times n}$ such that*

$$U^* A V = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}, \quad (2.2)$$

where $\Sigma_r = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ and $\{\sigma_w\}_{w=1}^r$ are the positive singular values of A.

Finally, we have the following result on the equivalence between the eigenvalue problem of a quaternion matrix and the eigenvalue problem of its real counterpart [11].

Lemma 2.3. *Let $A = X + Y\mathbf{j}$ be a quaternion matrix, where $X = A_0 + A_1\mathbf{i}$ and $Y = A_2 + A_3\mathbf{i}$ with $A_0, A_1, A_2, A_3 \in \mathbb{R}^{n\times n}$. Then there exists a unitary quaternion matrix*

$$Q = \frac{1}{2}\begin{bmatrix} I_n & -\mathbf{j}I_n & -\mathbf{i}I_n & -\mathbf{k}I_n \\ I_n & \mathbf{j}I_n & -\mathbf{i}I_n & \mathbf{k}I_n \\ I_n & -\mathbf{j}I_n & \mathbf{i}I_n & \mathbf{k}I_n \\ I_n & \mathbf{j}I_n & \mathbf{i}I_n & -\mathbf{k}I_n \end{bmatrix}$$

such that

$$\Gamma_A = Q^* \begin{bmatrix} X + Y\mathbf{j} & 0 & 0 & 0 \\ 0 & X - Y\mathbf{j} & 0 & 0 \\ 0 & 0 & \bar{X} + \bar{Y}\mathbf{j} & 0 \\ 0 & 0 & 0 & \bar{X} - \bar{Y}\mathbf{j} \end{bmatrix} Q. \quad (2.3)$$

3. Structure-preserving one-sided cyclic Jacobi algorithm

In this section, we present a structure-preserving one-sided cyclic Jacobi algorithm for computing the SVD of a quaternion matrix $A = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{H}^{m\times n}$, where $A_0, A_1, A_2, A_3 \in \mathbb{R}^{m\times n}$. The proposed structure-preserving one-sided cyclic Jacobi algorithm involves a sequence of orthogonal JRS-symplectic transformations $\Gamma_A \leftarrow \Gamma_A\Gamma_G$ such that the updated Γ_A is closer to a column-orthogonal matrix than its predecessor. When the updated Γ_A has sufficiently orthogonal columns, the column scaling of the updated A leads to the SVD of A.

For simplicity, we assume that $m \geq n$. The real counterpart Γ_A of A is defined by (2.1). A one-sided cyclic Jacobi algorithm includes (a) choosing an index pair (p, q) such that $1 \leq p < q \leq n$, (b) computing a cosine-sine group $(c_r, s_0, s_1, s_2, s_3)$ such that

$$G(p,q,\theta) = I_n + [\mathbf{e}_p, \mathbf{e}_q] \begin{bmatrix} c_r - 1 & s \\ -\bar{s} & c_r - 1 \end{bmatrix} \begin{bmatrix} \mathbf{e}_p^T \\ \mathbf{e}_q^T \end{bmatrix} \in \mathbb{H}^{n\times n} \tag{3.1}$$

is a unitary quaternion matrix and the p-th and q-th columns of $AG(p,q,\theta)$ are orthogonal, where $\mathbf{e}_t$ is the t-th unit vector and $s = s_0 + s_1\mathbf{i} + s_2\mathbf{j} + s_3\mathbf{k} \in \mathbb{H}$ with $c_r^2 + |s|^2 = 1$ (in fact, this corresponds zeroing the (p,q) and (q,p) entries of A^*A by using $G(p,q,\theta)^*A^*AG(p,q,\theta)$), and (c) overwriting A with $AG(p,q,\theta)$.

Our structure-preserving one-sided cyclic Jacobi algorithm aims to determine a sequence of orthogonal JRS-symplectic Jacobi matrices $\{\Gamma_{G^{(\ell)}} \in \mathbb{R}^{4n\times 4n}\}_{\ell=1}^{\eta}$ such that $\Gamma_{\widetilde{A}} = \Gamma_A \Gamma_{G^{(1)}} \Gamma_{G^{(2)}} \cdots \ \Gamma_{G^{(\eta)}}$ has sufficiently orthogonal columns, which corresponds with the off-diagonal entries of $\Gamma_{\widetilde{A}}^T \Gamma_{\widetilde{A}} = \Gamma_{G^{(\eta)}}^T \cdots \Gamma_{G^{(2)}}^T \Gamma_{G^{(1)}}^T \Gamma_A^T \Gamma_A \Gamma_{G^{(1)}} \Gamma_{G^{(2)}} \cdots \Gamma_{G^{(\eta)}}$ sufficiently close to zeros. Then, by extracting the first row partitions of $\Gamma_{\widetilde{A}}$, i.e.,

$$[\widetilde{A}_0, \widetilde{A}_2, \widetilde{A}_1, \widetilde{A}_3], \quad \widetilde{A}_w \in \mathbb{R}^{m\times n}, \quad w = 0, 1, 2, 3,$$

we get the updated quaternion matrix $\widetilde{A} = \widetilde{A}_0 + \widetilde{A}_1\mathbf{i} + \widetilde{A}_2\mathbf{j} + \widetilde{A}_3\mathbf{k} = AV$, where $V = G^{(1)}G^{(2)} \cdots \ G^{(\eta)}$ is an $n \times n$ unitary quaternion matrix. Finally, the column scaling of $\widetilde{A}$ yields the QSVD of A:

$$AV = \widetilde{A} = U\Sigma, \tag{3.2}$$

where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ with $\sigma_w \geq 0$ for $w = 1, \ldots, n$ and $U \in \mathbb{H}^{m\times n}$ satisfies $U^*U = I_n$.

The following theorem presents the orthogonalization of any two columns of an $m \times n$ quaternion matrix.

Theorem 3.1. *Let $A(p,q) = [\mathbf{a}_p, \mathbf{a}_q]$, where $\mathbf{a}_w = \mathbf{a}_{w0} + \mathbf{a}_{w1}\mathbf{i} + \mathbf{a}_{w2}\mathbf{j} + \mathbf{a}_{w3}\mathbf{k} \in \mathbb{H}^m$ is the w-th column of an $m \times n$ quaternion matrix A for $w = p, q$. If $\mathbf{a}_p^*\mathbf{a}_q \neq 0$, then there exists a 2-by-2 unitary quaternion matrix given by*

$$G(p,q;\theta) = \begin{bmatrix} c_r & s \\ -\bar{s} & c_r \end{bmatrix}$$

such that $\widetilde{A}(p,q) := A(p,q)G(p,q;\theta)$ has orthogonal columns, where $c_r = \cos(\theta) \in \mathbb{R}$ and $s = s_0 + s_1\mathbf{i} + s_2\mathbf{j} + s_3\mathbf{k} \in \mathbb{H}$ with $c_r^2 + |s|^2 = 1$.

Proof. Note that

$$\begin{aligned} \widetilde{A}(p,q) &= A(p,q)G(p,q;\theta) \\ &= [\mathbf{a}_p, \mathbf{a}_q] \begin{bmatrix} c_r & s \\ -\bar{s} & c_r \end{bmatrix} \\ &= [c_r\mathbf{a}_p - \mathbf{a}_q\bar{s}, \mathbf{a}_p s + c_r\mathbf{a}_q] \\ &:= [\widetilde{\mathbf{a}}_p, \widetilde{\mathbf{a}}_q]. \end{aligned} \tag{3.3}$$

By hypothesis $\mathbf{a}_p^*\mathbf{a}_q \neq 0$ and thus $|\mathbf{a}_p^*\mathbf{a}_q| > 0$. Define $c_r \in \mathbb{R}$ and $s = s_0 + s_1\mathbf{i} + s_2\mathbf{j} + s_3\mathbf{k} \in \mathbb{H}$ by

$$\begin{cases} s_0 = \frac{\sin(\theta)}{|\mathbf{a}_p^*\mathbf{a}_q|} a_{pq0}, \quad s_1 = \frac{\sin(\theta)}{|\mathbf{a}_p^*\mathbf{a}_q|} a_{pq1}, \quad s_2 = \frac{\sin(\theta)}{|\mathbf{a}_p^*\mathbf{a}_q|} a_{pq2}, \quad s_3 = \frac{\sin(\theta)}{|\mathbf{a}_p^*\mathbf{a}_q|} a_{pq3}, \quad \sin(\theta) = tc_r, \\ c_r = \cos(\theta) = \frac{1}{\sqrt{1+t^2}}, \quad |s| = \frac{|t|}{\sqrt{1+t^2}}, \quad t = \begin{cases} \frac{1}{\tau + \sqrt{1+\tau^2}}, & \text{if } \tau \geq 0 \\ \frac{1}{\tau - \sqrt{1+\tau^2}}, & \text{if } \tau < 0 \end{cases}, \quad \tau = \frac{\mathbf{a}_q^*\mathbf{a}_q - \mathbf{a}_p^*\mathbf{a}_p}{2|\mathbf{a}_p^*\mathbf{a}_q|}, \end{cases} \tag{3.4}$$

where $\mathbf{a}_p^*\mathbf{a}_q := a_{pq0} + a_{pq1}\mathbf{i} + a_{pq2}\mathbf{j} + a_{pq3}\mathbf{k}$ with

$$\begin{cases} a_{pq0} = \mathbf{a}_{p0}^T\mathbf{a}_{q0} + \mathbf{a}_{p1}^T\mathbf{a}_{q1} + \mathbf{a}_{p2}^T\mathbf{a}_{q2} + \mathbf{a}_{p3}^T\mathbf{a}_{q3}, \\ a_{pq1} = \mathbf{a}_{p0}^T\mathbf{a}_{q1} - \mathbf{a}_{p1}^T\mathbf{a}_{q0} - \mathbf{a}_{p2}^T\mathbf{a}_{q3} + \mathbf{a}_{p3}^T\mathbf{a}_{q2}, \\ a_{pq2} = \mathbf{a}_{p0}^T\mathbf{a}_{q2} + \mathbf{a}_{p1}^T\mathbf{a}_{q3} - \mathbf{a}_{p2}^T\mathbf{a}_{q0} - \mathbf{a}_{p3}^T\mathbf{a}_{q1}, \\ a_{pq3} = \mathbf{a}_{p0}^T\mathbf{a}_{q_3} - \mathbf{a}_{p1}^T\mathbf{a}_{q2} + \mathbf{a}_{p2}^T\mathbf{a}_{q1} - \mathbf{a}_{p3}^T\mathbf{a}_{q0}. \end{cases}$$

It is easy to see that $G(p,q;\theta)$ is unitary and

$$s = \frac{tc_r}{|\mathbf{a}_p^*\mathbf{a}_q|} \times \mathbf{a}_p^*\mathbf{a}_q \quad \text{and} \quad t^2 + 2\tau t - 1 = 0.$$

Thus,

$$
\begin{aligned}
\widetilde{\mathbf{a}}_p^*\widetilde{\mathbf{a}}_q &= (c_r\mathbf{a}_p - \mathbf{a}_q\bar{s})^*(\mathbf{a}_p s + c_r\mathbf{a}_q) \\
&= c_r\mathbf{a}_p^*\mathbf{a}_p s + c_r^2\mathbf{a}_p^*\mathbf{a}_q - s\mathbf{a}_q^*\mathbf{a}_p s - c_r\mathbf{a}_q^*\mathbf{a}_q s \\
&= c_r \times \mathbf{a}_p^*\mathbf{a}_p \times \frac{tc_r}{|\mathbf{a}_p^*\mathbf{a}_q|} \times \mathbf{a}_p^*\mathbf{a}_q + c_r^2 \times \mathbf{a}_p^*\mathbf{a}_q - c_r \times \mathbf{a}_q^*\mathbf{a}_q \times \frac{tc_r}{|\mathbf{a}_p^*\mathbf{a}_q|} \times \mathbf{a}_p^*\mathbf{a}_q \\
&\quad - \frac{tc_r}{|\mathbf{a}_p^*\mathbf{a}_q|} \times \mathbf{a}_p^*\mathbf{a}_q \times \mathbf{a}_q^*\mathbf{a}_p \times \frac{tc_r}{|\mathbf{a}_p^*\mathbf{a}_q|} \times \mathbf{a}_p^*\mathbf{a}_q \\
&= -c_r^2 \times \mathbf{a}_p^*\mathbf{a}_q \times \left(t^2 + \frac{\mathbf{a}_q^*\mathbf{a}_q - \mathbf{a}_p^*\mathbf{a}_p}{|\mathbf{a}_p^*\mathbf{a}_q|}t - 1\right) \\
&= -c_r^2 \times \mathbf{a}_p^*\mathbf{a}_q \times (t^2 + 2\tau t - 1) \\
&= 0. \quad \square
\end{aligned} \tag{3.5}
$$

In Theorem 3.1, we construct a unitary quaternion matrix $G(p,q;\theta)$ with real diagonal elements. In the proof of Theorem 3.1, we have used the commutativity of the real diagonal element c_r of $G(p,q;\theta)$ with quaternions. One may employ the generalized quaternion Givens transformation in [14] and at least one of its diagonal elements is quaternion. This needs further study.

The following corollary shows that the orthogonalization of $A(p,q)$ is the same as the diagonalization of $A^*(p,q)A(p,q)$ as in [27, Theorem 3.1].

Corollary 3.2. *If the assumptions in Theorem 3.1 hold, then there exists a* 2*-by-*2 *unitary quaternion matrix*

$$G(p,q;\theta) = \begin{bmatrix} c_r & s \\ -\bar{s} & c_r \end{bmatrix}, \qquad c_r = \cos(\theta) \in \mathbb{R} \text{ and } s = s_0 + s_1\mathbf{i} + s_2\mathbf{j} + s_3\mathbf{k} \in \mathbb{H}$$

such that

$$\Gamma_{G(p,q;\theta)}^T\Gamma_{A(p,q)}^T\Gamma_{A(p,q)}\Gamma_{G(p,q;\theta)} = \widetilde{B}(p,q;p,q) \oplus \widetilde{B}(p,q;p,q) \oplus \widetilde{B}(p,q;p,q) \oplus \widetilde{B}(p,q;p,q)$$

for some

$$\widetilde{B}(p,q;p,q) = \begin{bmatrix} b_{pp} & 0 \\ 0 & b_{qq} \end{bmatrix}.$$

Proof. By Theorem 3.1 we know that $\widetilde{A}(p,q) = A(p,q)G(p,q;\theta)$ has orthogonal columns for $G(p,q;\theta)$ with c_r and s defined as in (3.4). From (3.3) and (3.5) we have

$$\widetilde{B}(p,q;p,q) := \widetilde{A}^*(p,q)\widetilde{A}(p,q) = \begin{bmatrix} \widetilde{\mathbf{a}}_p^*\widetilde{\mathbf{a}}_p & \widetilde{\mathbf{a}}_p^*\widetilde{\mathbf{a}}_q \\ \widetilde{\mathbf{a}}_q^*\widetilde{\mathbf{a}}_p & \widetilde{\mathbf{a}}_q^*\widetilde{\mathbf{a}}_q \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{a}}_p^*\widetilde{\mathbf{a}}_p & 0 \\ 0 & \widetilde{\mathbf{a}}_q^*\widetilde{\mathbf{a}}_q \end{bmatrix}. \tag{3.6}$$

This shows that $\widetilde{B}(p,q;p,q)$ is a real diagonal matrix, where $\widetilde{\mathbf{a}}_p = c_r\mathbf{a}_p - \mathbf{a}_q\bar{s}$ and $\widetilde{\mathbf{a}}_q = \mathbf{a}_p s + c_r\mathbf{a}_q$ with c_r and s being defined in (3.4). Using Lemma 2.1 and (3.6) we find

$$
\begin{aligned}
&\Gamma_{G(p,q;\theta)}^T\Gamma_{A(p,q)}^T\Gamma_{A(p,q)}\Gamma_{G(p,q;\theta)} \\
&= \Gamma_{\widetilde{A}^*(p,q)}\Gamma_{\widetilde{A}(p,q)} = \Gamma_{\widetilde{A}^*(p,q)\widetilde{A}(p,q)} = \Gamma_{\widetilde{B}(p,q;p,q)} \\
&= \widetilde{B}(p,q;p,q) \oplus \widetilde{B}(p,q;p,q) \oplus \widetilde{B}(p,q;p,q) \oplus \widetilde{B}(p,q;p,q),
\end{aligned}
$$

where $b_{pp} = \widetilde{\mathbf{a}}_p^*\widetilde{\mathbf{a}}_p$ and $b_{qq} = \widetilde{\mathbf{a}}_q^*\widetilde{\mathbf{a}}_q$. $\square$

Remark 3.3. In fact, we choose t as defined in (3.4), which is the smaller of the two roots of $t^2 + 2\tau t - 1 = 0$. Thus the rotation angle satisfies $|\theta| \le \pi/4$ [8, p. 427]. Also, from (3.3) and (3.6) we have

$$
\begin{aligned}
\begin{bmatrix} \widetilde{\mathbf{a}}_p^*\widetilde{\mathbf{a}}_p & 0 \\ 0 & \widetilde{\mathbf{a}}_q^*\widetilde{\mathbf{a}}_q \end{bmatrix} &= \widetilde{A}^*(p,q)\widetilde{A}(p,q) \\
&= G^*(p,q;\theta)A^*(p,q)A(p,q)G(p,q;\theta) \\
&= \begin{bmatrix} c_r & s \\ -\bar{s} & c_r \end{bmatrix}^* \begin{bmatrix} \mathbf{a}_p^*\mathbf{a}_p & \mathbf{a}_p^*\mathbf{a}_q \\ \mathbf{a}_q^*\mathbf{a}_p & \mathbf{a}_q^*\mathbf{a}_q \end{bmatrix} \begin{bmatrix} c_r & s \\ -\bar{s} & c_r \end{bmatrix}.
\end{aligned}
$$

Since the Frobenius norm is unitary invariant we obtain

$$(\widetilde{\mathbf{a}}_p^*\widetilde{\mathbf{a}}_p)^2 + (\widetilde{\mathbf{a}}_q^*\widetilde{\mathbf{a}}_q)^2 = (\mathbf{a}_p^*\mathbf{a}_p)^2 + 2|\mathbf{a}_p^*\mathbf{a}_q|^2 + (\mathbf{a}_q^*\mathbf{a}_q)^2. \tag{3.7}$$

Remark 3.4. As noted in [27, p. 814], we observe from Theorem 3.1 that the matrix $\widetilde{A}^*\widetilde{A}$ agrees with A^*A except for the p-th and q-th rows and the p-th and q-th columns. Let $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ and $\widetilde{A} = [\widetilde{\mathbf{a}}_1, \widetilde{\mathbf{a}}_2, \ldots, \widetilde{\mathbf{a}}_n]$. Using (3.7) we have

$$\begin{aligned}
\mathrm{off}(\widetilde{A}^*\widetilde{A})^2 &= \|\widetilde{A}^*\widetilde{A}\|_F^2 - \sum_{t=1}^{n}(\widetilde{\mathbf{a}}_t^*\widetilde{\mathbf{a}}_t)^2 \\
&= \|A^*A\|_F^2 - \sum_{t=1}^{n}(\mathbf{a}_t^*\mathbf{a}_t)^2 + \left((\mathbf{a}_p^*\mathbf{a}_p)^2 + (\mathbf{a}_q^*\mathbf{a}_q)^2 - (\widetilde{\mathbf{a}}_p^*\widetilde{\mathbf{a}}_p)^2 - (\widetilde{\mathbf{a}}_q^*\widetilde{\mathbf{a}}_q)^2\right) \\
&= \mathrm{off}(A^*A)^2 - 2|\mathbf{a}_p^*\mathbf{a}_q|^2.
\end{aligned} \tag{3.8}$$

Since $\|\Gamma_A^T\Gamma_A\|_F^2 = 4\|A^*A\|_F^2$, we know that $\mathrm{off}(\Gamma_A^T\Gamma_A)^2 = 4\mathrm{off}(A^*A)^2$. Using (3.8), this implies that $\Gamma_A^T\Gamma_A$ is closer to a diagonal matrix with each orthogonal JRS-symplectic Jacobi rotation.

Based on Theorem 3.1 and Corollary 3.2, we present the following algorithm for generating a 2-by-2 unitary quaternion Jacobi matrix for orthogonalizing any two columns of an $m\times n$ quaternion matrix. This algorithm needs $96m+30$ operations.

Algorithm 3.5. Given $A(p,q) = \left[\mathbf{a}_p, \mathbf{a}_q\right]$, where $\mathbf{a}_w = \mathbf{a}_{w0} + \mathbf{a}_{w1}\mathbf{i} + \mathbf{a}_{w2}\mathbf{j} + \mathbf{a}_{w3}\mathbf{k} \in \mathbb{H}^m$ is the w-th column of an $m\times n$ quaternion matrix A for $w = p, q$, this algorithm computes a cosine-sine group $(c_r, s_0, s_1, s_2, s_3)$ such that $\widetilde{A}(p,q) = A(p,q)G(p,q;\theta)$ has two orthogonal columns.

function $(c_r, s_0, s_1, s_2, s_3) = \mathbf{GJSJR}(\mathbf{a}_{p0}, \mathbf{a}_{p1}, \mathbf{a}_{p2}, \mathbf{a}_{p3}, \mathbf{a}_{q0}, \mathbf{a}_{q1}, \mathbf{a}_{q2}, \mathbf{a}_{q3})$

$a_{pp0} = \mathbf{a}_{p0}^T\mathbf{a}_{p0} + \mathbf{a}_{p1}^T\mathbf{a}_{p1} + \mathbf{a}_{p2}^T\mathbf{a}_{p2} + \mathbf{a}_{p3}^T\mathbf{a}_{p3}$, $a_{pp1} = \mathbf{a}_{p0}^T\mathbf{a}_{p1} - \mathbf{a}_{p1}^T\mathbf{a}_{p0} - \mathbf{a}_{p2}^T\mathbf{a}_{p3} + \mathbf{a}_{p3}^T\mathbf{a}_{p2}$

$a_{pp2} = \mathbf{a}_{p0}^T\mathbf{a}_{p2} + \mathbf{a}_{p1}^T\mathbf{a}_{p3} - \mathbf{a}_{p2}^T\mathbf{a}_{p0} - \mathbf{a}_{p3}^T\mathbf{a}_{p1}$, $a_{pp3} = \mathbf{a}_{p0}^T\mathbf{a}_{p3} - \mathbf{a}_{p1}^T\mathbf{a}_{p2} + \mathbf{a}_{p2}^T\mathbf{a}_{p1} - \mathbf{a}_{p3}^T\mathbf{a}_{p0}$

$\beta_1 = \sqrt{a_{pp0}^2 + a_{pp1}^2 + a_{pp2}^2 + a_{pp3}^2}$

$a_{qq0} = \mathbf{a}_{q0}^T\mathbf{a}_{q0} + \mathbf{a}_{q1}^T\mathbf{a}_{q1} + \mathbf{a}_{q2}^T\mathbf{a}_{q2} + \mathbf{a}_{q3}^T\mathbf{a}_{q3}$, $a_{qq1} = \mathbf{a}_{q0}^T\mathbf{a}_{q1} - \mathbf{a}_{q1}^T\mathbf{a}_{q0} - \mathbf{a}_{q2}^T\mathbf{a}_{q3} + \mathbf{a}_{q3}^T\mathbf{a}_{q2}$

$a_{qq2} = \mathbf{a}_{q0}^T\mathbf{a}_{q2} + \mathbf{a}_{q1}^T\mathbf{a}_{q3} - \mathbf{a}_{q2}^T\mathbf{a}_{q0} - \mathbf{a}_{q3}^T\mathbf{a}_{q1}$, $a_{qq3} = \mathbf{a}_{q0}^T\mathbf{a}_{q_3} - \mathbf{a}_{q1}^T\mathbf{a}_{q2} + \mathbf{a}_{q2}^T\mathbf{a}_{q1} - \mathbf{a}_{q3}^T\mathbf{a}_{q0}$

$\beta_2 = \sqrt{a_{qq0}^2 + a_{qq1}^2 + a_{qq2}^2 + a_{qq3}^2}$

$a_{pq0} = \mathbf{a}_{p0}^T\mathbf{a}_{q0} + \mathbf{a}_{p1}^T\mathbf{a}_{q1} + \mathbf{a}_{p2}^T\mathbf{a}_{q2} + \mathbf{a}_{p3}^T\mathbf{a}_{q3}$, $a_{pq1} = \mathbf{a}_{p0}^T\mathbf{a}_{q1} - \mathbf{a}_{p1}^T\mathbf{a}_{q0} - \mathbf{a}_{p2}^T\mathbf{a}_{q3} + \mathbf{a}_{p3}^T\mathbf{a}_{q2}$

$a_{pq2} = \mathbf{a}_{p0}^T\mathbf{a}_{q2} + \mathbf{a}_{p1}^T\mathbf{a}_{q3} - \mathbf{a}_{p2}^T\mathbf{a}_{q0} - \mathbf{a}_{p3}^T\mathbf{a}_{q1}$, $a_{pq3} = \mathbf{a}_{p0}^T\mathbf{a}_{q_3} - \mathbf{a}_{p1}^T\mathbf{a}_{q2} + \mathbf{a}_{p2}^T\mathbf{a}_{q1} - \mathbf{a}_{p3}^T\mathbf{a}_{q0}$

$\beta_3 = \sqrt{a_{pq0}^2 + a_{pq1}^2 + a_{pq2}^2 + a_{pq3}^2}$

if $\beta_3 = 0$

  $c_r = 1$, $s_0 = s_1 = s_2 = s_3 = 0$

else

  $\tau = (\beta_2 - \beta_1)/(2\beta_3)$

  if $\tau \ge 0$

    $t = 1/(\tau + \sqrt{1+\tau^2})$

  else

    $t = 1/(\tau - \sqrt{1+\tau^2})$

  end

  $c_r = 1/\sqrt{1+t^2}$, $\delta = tc_r/\beta_3$, $s_0 = \delta a_{pq0}$, $s_1 = \delta a_{pq1}$, $s_2 = \delta a_{pq2}$, $s_3 = \delta a_{pq3}$

end

Algorithm 3.5 gives a scheme for orthogonalizing p-th and q-th columns of an $m\times n$ quaternion matrix $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$. One may choose p and q such that $|\mathbf{a}_p^*\mathbf{a}_q|$ is maximal as the classical Jacobi algorithm [8, Algorithm 8.4.2].

The following algorithm describes a structure-preserving one-sided classical Jacobi algorithm, which is such that a quaternion matrix has sufficiently orthogonal columns.

Algorithm 3.6. Given an $m \times n$ quaternion matrix $A = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{H}^{m\times n}$ and a tolerance $\mathtt{tol} > 0$, this algorithm overlaps the real counterpart Γ_A by $\Gamma_A \widetilde{V}$, where $\widetilde{V}$ is orthogonal and $\mathrm{off}(\widetilde{V}^T \Gamma_A^T \Gamma_A \widetilde{V}) \le \mathtt{tol} \cdot \|\Gamma_A^T\Gamma_A\|_F$.

$\widetilde{V} = \Gamma_{I_n}$, $\zeta = \mathtt{tol} \cdot \|\Gamma_A^T\Gamma_A\|_F$

while $\mathrm{off}(\Gamma_A^T\Gamma_A) > \zeta$

  Choose (p, q) so $|\mathbf{a}_p^*\mathbf{a}_q| = \max_{u\neq v} |\mathbf{a}_u^*\mathbf{a}_v|$

  $\mathbf{a}_{p0} = A_0(:, p)$, $\mathbf{a}_{p1} = A_1(:, p)$, $\mathbf{a}_{p2} = A_2(:, p)$, $\mathbf{a}_{p3} = A_3(:, p)$

  $\mathbf{a}_{q0} = A_0(:, q)$, $\mathbf{a}_{q1} = A_1(:, q)$, $\mathbf{a}_{q2} = A_2(:, q)$, $\mathbf{a}_{q3} = A_3(:, q)$

  $(c_r, s_0, s_1, s_2, s_3) = \mathbf{GJSJR}(\mathbf{a}_{p0}, \mathbf{a}_{p1}, \mathbf{a}_{p2}, \mathbf{a}_{p3}, \mathbf{a}_{q0}, \mathbf{a}_{q1}, \mathbf{a}_{q2}, \mathbf{a}_{q3})$

  $\Gamma_A = \Gamma_A \Gamma_{G(p,q,\theta)}$

  $\widetilde{V} = \widetilde{V}\Gamma_{G(p,q,\theta)}$

end

Algorithm 3.6 gives the following basic iterative scheme:

$$\Gamma_{A^{(\ell+1)}} = \Gamma_{A^{(\ell)}}\Gamma_{G^{(\ell)}}, \quad \ell = 0, 1, 2, \ldots, \tag{3.9}$$

where $A^{(0)} = A$, and $G^{(\ell)} \in \mathbb{H}^{n\times n}$ is a unitary quaternion matrix defined in (3.1) with the cosine-sine group $(c_r, s_0, s_1, s_2, s_3)$ being generated by Algorithm 3.5. Algorithm 3.6 may be seen as a structure-preserving Jacobi algorithm for solving the eigenvalue problem of an $n \times n$ quaternion Hermitian matrix $B := A^*A$ as in [27]:

$$\begin{aligned}\Gamma_{B^{(\ell+1)}} &= \Gamma^T_{A^{(\ell+1)}}\Gamma_{A^{(\ell+1)}} = \Gamma^T_{G^{(\ell)}}\Gamma^T_{A^{(\ell)}}\Gamma_{A^{(\ell)}}\Gamma_{G^{(\ell)}}\\ &= \Gamma^T_{G^{(\ell)}}\Gamma_{(A^{(\ell)})^*A^{(\ell)}}\Gamma_{G^{(\ell)}} = \Gamma^T_{G^{(\ell)}}\Gamma_{B^{(\ell)})}\Gamma_{G^{(\ell)}}, \quad \ell = 0, 1, 2, \ldots,\end{aligned}$$

where $B^{(0)} = A^*A$. Let

$$A^{(\ell)} := [\mathbf{a}_1^{(\ell)}, \ldots, \mathbf{a}_p^{(\ell)}, \ldots, \mathbf{a}_q^{(\ell)}, \ldots, \mathbf{a}_n^{(\ell)}].$$

Then

$$\begin{aligned}B^{(\ell)} &:= (A^{(\ell)})^*A^{(\ell)}\\ &= \begin{bmatrix} (\mathbf{a}_1^{(\ell)})^*\mathbf{a}_1^{(\ell)} & \cdots & (\mathbf{a}_1^{(\ell)})^*\mathbf{a}_p^{(\ell)} & \cdots & (\mathbf{a}_1^{(\ell)})^*\mathbf{a}_q^{(\ell)} & \cdots & (\mathbf{a}_1^{(\ell)})^*\mathbf{a}_n^{(\ell)}\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ (\mathbf{a}_p^{(\ell)})^*\mathbf{a}_1^{(\ell)} & \cdots & (\mathbf{a}_p^{(\ell)})^*\mathbf{a}_p^{(\ell)} & \cdots & (\mathbf{a}_p^{(\ell)})^*\mathbf{a}_q^{(\ell)} & \cdots & (\mathbf{a}_p^{(\ell)})^*\mathbf{a}_n^{(\ell)}\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ (\mathbf{a}_q^{(\ell)})^*\mathbf{a}_1^{(\ell)} & \cdots & (\mathbf{a}_q^{(\ell)})^*\mathbf{a}_p^{(\ell)} & \cdots & (\mathbf{a}_q^{(\ell)})^*\mathbf{a}_q^{(\ell)} & \cdots & (\mathbf{a}_q^{(\ell)})^*\mathbf{a}_n^{(\ell)}\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ (\mathbf{a}_n^{(\ell)})^*\mathbf{a}_1^{(\ell)} & \cdots & (\mathbf{a}_n^{(\ell)})^*\mathbf{a}_p^{(\ell)} & \cdots & (\mathbf{a}_n^{(\ell)})^*\mathbf{a}_q^{(\ell)} & \cdots & (\mathbf{a}_n^{(\ell)})^*\mathbf{a}_n^{(\ell)} \end{bmatrix}.\end{aligned} \tag{3.10}$$

From Remark 3.4 we have

$$\begin{aligned}\mathrm{off}(B^{(\ell)})^2 &= \|B^{(\ell)}\|_F^2 - \sum_{u=1}^n (b_{uu}^{(\ell)})^2\\ &= \|(A^{(\ell)})^*A^{(\ell)}\|_F^2 - \sum_{u=1}^n (b_{uu}^{(\ell)})^2\\ &= \|(G^{(\ell-1)})^*B^{(\ell-1)}G^{(\ell-1)}\|_F^2 - \Big(\sum_{u=1}^n (b_{uu}^{(\ell-1)})^2 - (b_{pp}^{(\ell-1)})^2 - (b_{qq}^{(\ell-1)})^2 + (b_{pp}^{(\ell)})^2 + (b_{qq}^{(\ell)})^2\Big)\\ &= \|B^{(\ell-1)}\|_F^2 - \sum_{u=1}^n (b_{uu}^{(\ell-1)})^2 + \big((b_{pp}^{(\ell-1)})^2 + (b_{qq}^{(\ell-1)})^2 - (b_{pp}^{(\ell)})^2 - (b_{qq}^{(\ell)})^2\big)\\ &= \mathtt{off}(B^{(\ell-1)})^2 - 2|b_{pq}^{(\ell-1)}|^2.\end{aligned}$$

We see that $\|\Gamma_{B^{(\ell)}}\|_F^2 = 4\|B^{(\ell)}\|_F^2$. Hence, $\mathtt{off}(\Gamma_{B^{(\ell)}})^2 = 4\mathtt{off}(B^{(\ell)})^2$.

We have the following result on the linear convergence of Algorithm 3.6. The proof follows from [27, Theorem 3.3] and thus we omit it here.

Theorem 3.7. *Let $\{\sigma_w\}_{w=1}^n$ be the n singular values of A and $\Gamma_{A^{(\ell)}}$ be the matrix after ℓ orthogonal JRS-symplectic Jacobi updates generated by Algorithm 3.6. Then there exists a permutation $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ of $\{1, 2, \ldots, n\}$ such that*

$$\lim_{\ell\to\infty} \Gamma_{A^{(\ell)}}^T \Gamma_{A^{(\ell)}} = \mathrm{diag}(\sigma_{\varphi_1}^2, \ldots, \sigma_{\varphi_n}^2, \sigma_{\varphi_1}^2, \ldots, \sigma_{\varphi_n}^2, \sigma_{\varphi_1}^2, \ldots, \sigma_{\varphi_n}^2, \sigma_{\varphi_1}^2, \ldots, \sigma_{\varphi_n}^2).$$

Moreover,

$$\mathrm{off}(\Gamma_{A^{(\ell)}}^T \Gamma_{A^{(\ell)}}) \le \left(1 - \frac{1}{N}\right)^\ell \mathrm{off}(\Gamma_{A^{(0)}}^T \Gamma_{A^{(0)}}), \quad N := \frac{1}{2}n(n-1).$$

However, in Algorithm 3.6, the search for the optimal columns p and q needs $O(n^2)$. To reduce the cost, one may adopt the scheme of cyclic-by-column as the cyclic Jacobi algorithm [8, Algorithm 8.4.3]. Sparked by this, we propose the following structure-preserving one-sided cyclic Jacobi algorithm for orthogonalizing the columns of an $m \times n$ quaternion matrix A.

Algorithm 3.8. Given an $m \times n$ quaternion matrix $A = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{H}^{m\times n}$ and a tolerance $\mathtt{tol} > 0$, this algorithm overlaps the real counterpart Γ_A by $\Gamma_A \widetilde{V}$, where $\widetilde{V}$ is orthogonal and $\mathrm{off}(\widetilde{V}^T \Gamma_A^T \Gamma_A \widetilde{V}) \le \mathtt{tol} \cdot \|\Gamma_A^T \Gamma_A\|_F$.

$\widetilde{V} = \Gamma_{I_n}$, $\zeta = \mathtt{tol} \cdot \|\Gamma_A^T \Gamma_A\|_F$
while $\mathrm{off}(\Gamma_A^T \Gamma_A) > \zeta$
  for $p = 1 : n-1$
    for $q = p+1 : n$
      $\mathbf{a}_{p0} = A_0(:, p)$, $\mathbf{a}_{p1} = A_1(:, p)$, $\mathbf{a}_{p2} = A_2(:, p)$, $\mathbf{a}_{p3} = A_3(:, p)$
      $\mathbf{a}_{q0} = A_0(:, q)$, $\mathbf{a}_{q1} = A_1(:, q)$, $\mathbf{a}_{q2} = A_2(:, q)$, $\mathbf{a}_{q3} = A_3(:, q)$
      $(c_r, s_0, s_1, s_2, s_3) = \mathbf{GJSJR}(\mathbf{a}_{p0}, \mathbf{a}_{p1}, \mathbf{a}_{p2}, \mathbf{a}_{p3}, \mathbf{a}_{q0}, \mathbf{a}_{q1}, \mathbf{a}_{q2}, \mathbf{a}_{q3})$
      $\Gamma_A = \Gamma_A \Gamma_{G(p,q,\theta)}$
      $\widetilde{V} = \widetilde{V} \Gamma_{G(p,q,\theta)}$
    end
  end
end

On Algorithm 3.8, we have the following remarks.

Remark 3.9. There exist some differences between Algorithm 3.8 and the classic one-sided cyclic Jacobi algorithm. Our method is applied to Γ_A, which involves real operations while the classical one was used to A in quaternion operations. For each given index pair (p, q), the classic one-sided cyclic Jacobi algorithm finds a unitary quaternion matrix $G(p, q, \theta)$ such that the p-th and q-th columns of $AG(p, q, \theta)$ are orthogonal while our method generates an orthogonal JRS-symplectic Jacobi matrices $\Gamma_{G(p,q,\theta)}$ such that the p-th and q-th columns of $AG(p, q, \theta)$ are orthogonal, where the dimension-expanding problem, caused by the real counterpart of A, is avoided since $\Gamma_A \Gamma_{G(p,q,\theta)}$ is still JRS-symmetric.

Remark 3.10. In Algorithm 3.8, we only need to store the first m rows of Γ_A, which reduce the total storage. The later numerical examples show that Algorithm 3.8 works very effectively.

We now focus on the quadratic convergence analysis of Algorithm 3.8. Algorithm 3.8 gives the following iterative scheme:

$$\Gamma_{A^{(\ell+1)}} = \Gamma_{A^{(\ell)}} \Gamma_{G^{(\ell)}}, \quad \ell = 0, 1, 2, \ldots.$$

Here, $A^{(0)} = A$, and $G^{(\ell)} \in \mathbb{H}^{n\times n}$ is a unitary quaternion matrix defined in (3.1), where the cosine-sine group $(c_r, s_0, s_1, s_2, s_3)$ is generated by Algorithm 3.5. In fact, Algorithm 3.8 can be seen as a structure-preserving cyclic Jacobi algorithm for $\Gamma_A^T \Gamma_A$.

We have the following theorem on the quadratic convergence of Algorithm 3.8. The proof can be seen as a generalization of [32,34].

Theorem 3.11. *Let $\{\sigma_w\}_{w=1}^n$ be the n singular values of A and $\Gamma_{A^{(\ell)}}$ be the matrix after ℓ orthogonal JRS-symplectic Jacobi updates generated by Algorithm 3.8. If $\mathrm{off}(\Gamma_{A^{(d)}}^T \Gamma_{A^{(d)}}) < \delta/2$ for some $d \ge 1$ where $0 < 2\delta \le \min_{\sigma_u \ne \sigma_v} |\sigma_u^2 - \sigma_v^2|$, then*

$$\mathrm{off}(\Gamma_{A^{(d+N)}}^T \Gamma_{A^{(d+N)}}) \le \sqrt{\frac{25}{72}} \cdot \frac{\mathrm{off}(\Gamma_{A^{(d)}}^T \Gamma_{A^{(d)}})^2}{\delta}.$$

Proof. Write $S^{(d)} := \Gamma^T_{A^{(d)}} \Gamma_{A^{(d)}} = D^{(d)} + E^{(d)} + (E^{(d)})^T$, where $D^{(d)}$ and $E^{(d)}$ are diagonal and strictly upper triangular, respectively. By using Theorem 3.7 and the Wielandt-Hoffman theorem ([8, Theorem 8.1.4]) we have

$$|s^{(d)}_{ww} - \sigma^2_{\varphi_w}| \le \|D^{(d)} - S^{(d)}\|_F < \frac{\delta}{2}, \quad 1 \le w \le n. \tag{3.11}$$

Thus we have for two distinct eigenvalues $\sigma^2_{\varphi_u}$ and $\sigma^2_{\varphi_v}$,

$$\begin{aligned} |s^{(d)}_{uu} - s^{(d)}_{vv}| &= |(s^{(d)}_{uu} - \sigma^2_{\varphi_u}) - (s^{(d)}_{vv} - \sigma^2_{\varphi_v}) + (\sigma^2_{\varphi_u} - \sigma^2_{\varphi_v})| \\ &\ge |\sigma^2_{\varphi_u} - \sigma^2_{\varphi_v}| - |s^{(d)}_{uu} - \sigma^2_{\varphi_u}| - |s^{(d)}_{vv} - \sigma^2_{\varphi_v}| \\ &> 2\delta - \frac{\delta}{2} - \frac{\delta}{2} = \delta. \end{aligned} \tag{3.12}$$

Since $\mathrm{off}(\Gamma^T_{A^{(\ell)}} \Gamma_{A^{(\ell)}})$ is decreasing, we know that $\mathrm{off}(\Gamma^T_{A^{(\ell)}} \Gamma_{A^{(\ell)}}) < \delta/2$ and (3.12) hold for $\ell > d$.

We show the quadratic convergence of Algorithm 3.8. We first consider the case of one multiple singular value. Assume without loss of generality that only σ_{φ_1} is a multiple singular value of $A^{(\ell)}$ ($\ell > d$) and the diagonal entries $s^{(\ell)}_{11}, s^{(\ell)}_{22}, \ldots, s^{(\ell)}_{n_1 n_1}$ of $S^{(\ell)} := \Gamma^T_{A^{(\ell)}} \Gamma_{A^{(\ell)}}$ converge to $\sigma^2_{\varphi_1}$. Then, by appropriate row and column interchanges, we get a permutation matrix P such that

$$\widehat{S}^{(\ell)} = P^T S^{(\ell)} P = \begin{bmatrix} \widehat{S}^{(\ell)}_{11} & \widehat{S}^{(\ell)}_{12} \\ \widehat{S}^{(\ell)}_{21} & \widehat{S}^{(\ell)}_{22} \end{bmatrix},$$

where all the diagonal entries of $\widehat{S}^{(\ell)}_{11} \in \mathbb{R}^{4n_1 \times 4n_1}$ converge to $\sigma^2_{\varphi_1}$.

We provide an upper bound for the quantity

$$\Phi^{(\ell)}_1 := \sqrt{\sum_{1 \le p \ne q \le 4n_1} (\hat{s}^{(\ell)}_{pq})^2}.$$

As in [35], it is easy to see that

$$\begin{aligned} T^{(\ell)} &= \begin{bmatrix} I_{4n_1} & -\widehat{S}^{(\ell)}_{12} (\widehat{S}^{(\ell)}_{22} - \sigma^2_{\varphi_1} I_{4n-4n_1})^{-1} \\ 0 & I_{4n-4n_1} \end{bmatrix} \begin{bmatrix} \widehat{S}^{(\ell)}_{11} - \sigma^2_{\varphi_1} I_{4n_1} & \widehat{S}^{(\ell)}_{12} \\ \widehat{S}^{(\ell)}_{21} & \widehat{S}^{(\ell)}_{22} - \sigma^2_{\varphi_1} I_{4n-4n_1} \end{bmatrix} \\ &= \begin{bmatrix} \widehat{S}^{(\ell)}_{11} - \sigma^2_{\varphi_1} I_{4n_1} - \widehat{S}^{(\ell)}_{12} (\widehat{S}^{(\ell)}_{22} - \sigma^2_{\varphi_1} I_{4n-4n_1})^{-1} \widehat{S}^{(\ell)}_{21} & 0 \\ \widehat{S}^{(\ell)}_{21} & \widehat{S}^{(\ell)}_{22} - \sigma^2_{\varphi_1} I_{4n-4n_1} \end{bmatrix} \end{aligned}$$

and the rank of $T^{(\ell)}$ is the same as $\widehat{S}^{(\ell)} - \sigma^2_{\varphi_1} I_{4n}$, which implies that

$$\widehat{S}^{(\ell)}_{11} - \sigma^2_{\varphi_1} I_{4n_1} = \widehat{S}^{(\ell)}_{12} (\widehat{S}^{(\ell)}_{22} - \sigma^2_{\varphi_1} I_{4n-4n_1})^{-1} \widehat{S}^{(\ell)}_{21}. \tag{3.13}$$

Let $\widetilde{\sigma}^2_{\varphi_w}$ be the eigenvalues of $\widehat{S}^{(\ell)}_{22}$. Note that

$$|\sigma^2_{\varphi_w} - \widetilde{\sigma}^2_{\varphi_w}| \le \left\| \widehat{S}^{(\ell)} - \begin{bmatrix} \Sigma^{(\ell)}_{11} & 0 \\ 0 & \widehat{S}^{(\ell)}_{22} \end{bmatrix} \right\|_F \le \mathrm{off}(\widehat{S}^{(\ell)}) = \mathrm{off}(S^{(\ell)}) \le \frac{\delta}{2},$$

where $\Sigma^{(\ell)}_{11} = \mathrm{diag}(\hat{s}^{(\ell)}_{11}, \hat{s}^{(\ell)}_{22}, \ldots, \hat{s}^{(\ell)}_{4n_1,4n_1})$, $\hat{s}^{(\ell)}_{ww}$ is the (w, w) entry of $\widehat{S}^{(\ell)}_{11}$. Thus,

$$|\sigma^2_{\varphi_1} - \widetilde{\sigma}^2_{\varphi_w}| \ge |\sigma^2_{\varphi_1} - \sigma^2_{\varphi_w}| - |\sigma^2_{\varphi_w} - \widetilde{\sigma}^2_{\varphi_w}| \ge 2\delta - \frac{\delta}{2} = \frac{3\delta}{2}.$$

This, together with (3.13), yields

$$\begin{aligned} (\Phi^{(\ell)}_1)^2 = \mathrm{off}(\widehat{S}^{(\ell)}_{11})^2 \le \|\widehat{S}^{(\ell)}_{11} - \sigma^2_{\varphi_1} I_{4n_1}\|^2_F &\le \frac{\|\widehat{S}^{(\ell)}_{12}\|^4_F}{\min_{\widetilde{\sigma}^2_{\varphi_w} \ne \sigma^2_{\varphi_1}} |\sigma^2_{\varphi_1} - \widetilde{\sigma}^2_{\varphi_w}|^2} \\ &\le \frac{2\mathrm{off}(S^{(\ell)})^2}{9\delta^2} \|\widehat{S}^{(\ell)}_{12}\|^2_F \le \frac{2\mathrm{off}(S^{(d)})^2}{9\delta^2} \|\widehat{S}^{(\ell)}_{12}\|^2_F \qquad (3.14a) \\ &\le \frac{\mathrm{off}(S^{(d)})^4}{9\delta^2} \qquad (3.14b) \end{aligned}$$

since $\|\widehat{S}^{(\ell)}_{12}\|^2_F \le 1/2\ \mathrm{off}(\widehat{S}^{(\ell)})^2 = 1/2\ \mathrm{off}(S^{(\ell)})^2 \le 1/2\mathrm{off}(S^{(d)})^2$. The estimates in (3.14) are crucial for proving the quadratic convergence of the structure-preserving one-sided cyclic Jacobi algorithm.

As in (3.4), the rotation angle θ_ℓ is chosen such that $|\theta_\ell| \le \pi/4$. Using (3.10), (3.12) and $S^{(\ell)} = \Gamma_{B^{(\ell)}}$ we have

$$|\sin\theta_\ell| \le \frac{1}{2}|\tan 2\theta_\ell| = \frac{|b_{pq}^{(\ell-1)}|}{|b_{qq}^{(\ell-1)} - b_{pp}^{(\ell-1)}|} \le \frac{|b_{pq}^{(\ell-1)}|}{\delta}. \tag{3.15}$$

This, together with $\frac{1}{2}\mathrm{off}(B^{(\ell-1)})^2 - \frac{1}{2}\mathrm{off}(B^{(\ell)})^2 = |b_{pq}^{(\ell-1)}|^2$, yields

$$\mathrm{off}(B^{(d)})^2 - 2\sum_{\ell=d+1}^{d+N} |b_{pq}^{(\ell-1)}|^2 = \mathrm{off}(B^{(d+N)})^2 \ge 0.$$

Using (3.15) we have

$$\widehat{\sum}\sin^2\theta_\ell \le \frac{\sum |b_{pq}^{(\ell-1)}|^2}{\delta^2} \le \frac{\mathrm{off}(B^{(d)})^2}{2\delta^2}, \tag{3.16}$$

where $\widehat{\sum}$ means that we include in the sum only rotations of entries outside the first n_1 rows and the first n_1 columns of $B^{(\ell)}$.

We now show the quadratic convergence of Algorithm 3.8 for the case of one multiple singular value. As in [34], for example, we take an $m \times 5$ quaternion matrix A. In this case, $B^{(0)} = A^*A \in \mathbb{H}^{5\times 5}$. In the following, we show the effect of annihilating the entries in the first row and column of $B^{(d)}$. Since we are only interested in the off diagonal entries, which is updated when these entries are affected by the current rotations, the diagonal entries are all denoted by "×".

$$B^{(d)} := \begin{bmatrix} \times & b_{12}^{(d)} & b_{13}^{(d)} & b_{14}^{(d)} & b_{15}^{(d)} \\ b_{21}^{(d)} & \times & b_{23}^{(d)} & b_{24}^{(d)} & b_{25}^{(d)} \\ b_{31}^{(d)} & b_{32}^{(d)} & \times & b_{34}^{(d)} & b_{35}^{(d)} \\ b_{41}^{(d)} & b_{42}^{(d)} & b_{43}^{(d)} & \times & b_{45}^{(d)} \\ b_{51}^{(d)} & b_{52}^{(d)} & b_{53}^{(d)} & b_{54}^{(d)} & \times \end{bmatrix} \xrightarrow{G(1,2;\theta_d)} \begin{bmatrix} \times & 0 & b_{13}^{(d+1)} & b_{14}^{(d+1)} & b_{15}^{(d+1)} \\ 0 & \times & b_{23}^{(d+1)} & b_{24}^{(d+1)} & b_{25}^{(d+1)} \\ b_{31}^{(d+1)} & b_{32}^{(d+1)} & \times & b_{34}^{(d)} & b_{35}^{(d)} \\ b_{41}^{(d+1)} & b_{42}^{(d+1)} & b_{43}^{(d)} & \times & b_{45}^{(d)} \\ b_{51}^{(d+1)} & b_{52}^{(d+1)} & b_{53}^{(d)} & b_{54}^{(d)} & \times \end{bmatrix}$$

$$\xrightarrow{G(1,3;\theta_{d+1})} \begin{bmatrix} \times & b_{12}^{(d+2)} & 0 & b_{14}^{(d+2)} & b_{15}^{(d+2)} \\ b_{21}^{(d+2)} & \times & b_{23}^{(d+2)} & b_{24}^{(d+1)} & b_{25}^{(d+1)} \\ 0 & b_{32}^{(d+2)} & \times & b_{34}^{(d+2)} & b_{35}^{(d+2)} \\ b_{41}^{(d+2)} & b_{42}^{(d+1)} & b_{43}^{(d+2)} & \times & b_{45}^{(d)} \\ b_{51}^{(d+2)} & b_{52}^{(d+1)} & b_{53}^{(d+2)} & b_{54}^{(d)} & \times \end{bmatrix}$$

$$\xrightarrow{G(1,4;\theta_{d+2})} \begin{bmatrix} \times & b_{12}^{(d+3)} & b_{13}^{(d+3)} & 0 & b_{15}^{(d+3)} \\ b_{21}^{(d+3)} & \times & b_{23}^{(d+2)} & b_{24}^{(d+3)} & b_{25}^{(d+1)} \\ b_{31}^{(d+3)} & b_{32}^{(d+2)} & \times & b_{34}^{(d+3)} & b_{35}^{(d+2)} \\ 0 & b_{42}^{(d+3)} & b_{43}^{(d+3)} & \times & b_{45}^{(d+3)} \\ b_{51}^{(d+3)} & b_{52}^{(d+1)} & b_{53}^{(d+2)} & b_{54}^{(d+3)} & \times \end{bmatrix}$$

$$\xrightarrow{G(1,5;\theta_{d+3})} \begin{bmatrix} \times & b_{12}^{(d+4)} & b_{13}^{(d+4)} & b_{14}^{(d+4)} & 0 \\ b_{21}^{(d+4)} & \times & b_{23}^{(d+2)} & b_{24}^{(d+3)} & b_{25}^{(d+4)} \\ b_{31}^{(d+4)} & b_{32}^{(d+2)} & \times & b_{34}^{(d+3)} & b_{35}^{(d+4)} \\ b_{41}^{(d+4)} & b_{42}^{(d+3)} & b_{43}^{(d+3)} & \times & b_{45}^{(d+4)} \\ 0 & b_{52}^{(d+4)} & b_{53}^{(d+4)} & b_{54}^{(d+4)} & \times \end{bmatrix}. \tag{3.17}$$

For the entries of the first row of $B^{(d+4)}$, we have the following inequalities

$$\begin{cases} |b_{14}^{(d+4)}| \le |b_{54}^{(d+3)}||\sin\theta_{d+3}|, \\ |b_{13}^{(d+4)}| \le |b_{43}^{(d+2)}||\sin\theta_{d+2}| + |b_{53}^{(d+2)}||\sin\theta_{d+3}|, \\ |b_{12}^{(d+4)}| \le |b_{32}^{(d+1)}||\sin\theta_{d+1}| + |b_{42}^{(d+1)}||\sin\theta_{d+2}| + |b_{52}^{(d+1)}||\sin\theta_{d+3}|. \end{cases} \tag{3.18}$$

Thus,

$$
\begin{aligned}
&|b_{12}^{(d+4)}|^2+|b_{13}^{(d+4)}|^2+|b_{14}^{(d+4)}|^2\\
&\le (|b_{32}^{(d+1)}|^2+|b_{42}^{(d+1)}|^2+|b_{52}^{(d+1)}|^2)(\sin^2\theta_{d+1}+\sin^2\theta_{d+2}+\sin^2\theta_{d+3})\\
&\quad +(|b_{43}^{(d+2)}|^2+|b_{53}^{(d+2)}|^2)(\sin^2\theta_{d+2}+\sin^2\theta_{d+3})+|b_{54}^{(d+3)}|^2\sin^2\theta_{d+3}\\
&\le (|b_{32}^{(d+1)}|^2+|b_{42}^{(d+1)}|^2+|b_{52}^{(d+1)}|^2+|b_{43}^{(d+2)}|^2+|b_{53}^{(d+2)}|^2+|b_{54}^{(d+3)}|^2)\\
&\quad \times(\sin^2\theta_{d+1}+\sin^2\theta_{d+2}+\sin^2\theta_{d+3}).
\end{aligned}
$$

Since each rotation affects only two entries in each of the related columns or rows while the sum of the squares of their absolute values is kept unchanged we have

$$
\begin{aligned}
|b_{15}^{(d+3)}|^2+|b_{25}^{(d+1)}|^2+|b_{35}^{(d+2)}|^2+|b_{45}^{(d+3)}|^2&=|b_{15}^{(d)}|^2+|b_{25}^{(d)}|^2+|b_{35}^{(d)}|^2+|b_{45}^{(d)}|^2,\\
|b_{14}^{(d+2)}|^2+|b_{24}^{(d+1)}|^2+|b_{34}^{(d+2)}|^2&=|b_{14}^{(d)}|^2+|b_{24}^{(d)}|^2+|b_{34}^{(d)}|^2,\\
|b_{13}^{(d+1)}|^2+|b_{23}^{(d+1)}|^2&=|b_{13}^{(d)}|^2+|b_{23}^{(d)}|^2.
\end{aligned}
$$

Thus

$$
\begin{aligned}
&|b_{12}^{(d+4)}|^2+|b_{13}^{(d+4)}|^2+|b_{14}^{(d+4)}|^2\\
&\le (|b_{13}^{(d)}|^2+|b_{23}^{(d)}|^2+|b_{14}^{(d)}|^2+|b_{24}^{(d)}|^2+|b_{34}^{(d)}|^2+|b_{15}^{(d)}|^2+|b_{25}^{(d)}|^2+|b_{35}^{(d)}|^2+|b_{45}^{(d)}|^2)\\
&\quad \times(\sin^2\theta_{d+1}+\sin^2\theta_{d+2}+\sin^2\theta_{d+3})\\
&\le \frac{1}{2}\mathrm{off}(B^{(d)})^2(\sin^2\theta_{d+1}+\sin^2\theta_{d+2}+\sin^2\theta_{d+3}).
\end{aligned} \tag{3.19}
$$

Moreover, the sum of the squares of the absolute values of these entries in the first row remains unchanged in the subsequent rotations.

Similarly, we have after successively annihilating the entries in the second row

$$
\begin{aligned}
|b_{23}^{(d+7)}|^2+|b_{24}^{(d+7)}|^2&\le \frac{1}{2}\mathrm{off}(B^{(d+4)})^2(\sin^2\theta_{d+5}+\sin^2\theta_{d+6})\\
&\le \frac{1}{2}\mathrm{off}(B^{(d)})^2(\sin^2\theta_{d+5}+\sin^2\theta_{d+6}).
\end{aligned} \tag{3.20}
$$

Furthermore, for the third row we have

$$
|b_{34}^{(d+9)}|^2\le \frac{1}{2}\mathrm{off}(B^{(d+7)})^2\sin^2\theta_{d+8}\le \frac{1}{2}\mathrm{off}(B^{(d)})^2\sin^2\theta_{d+8}. \tag{3.21}
$$

Finally, the fourth row above the diagonal is annihilated. From (3.19), (3.20) and (3.21) we obtain

$$
\mathrm{off}(B^{(d+10)})^2\le \mathrm{off}(B^{(d)})^2\sum_{t=0}^{9}\sin^2\theta_{d+t}\le \mathrm{off}(B^{(d)})^2\frac{\mathrm{off}(B^{(d)})^2}{2\delta^2}=\frac{\mathrm{off}(B^{(d)})^4}{2\delta^2}, \tag{3.22}
$$

where the second inequality follows from (3.16).

Analogously to the proof of (3.22), using the equality $\mathrm{off}(S^{(d)})^2=4\mathrm{off}(B^{(d)})^2$, (3.14b) and (3.16) we have

$$
\begin{aligned}
\mathrm{off}(S^{(d+N)})^2&=4\mathrm{off}(B^{(d+N)})^2\\
&=4\sum_{1\le p\ne q\le n_1}|b_{pq}^{(d+N)}|^2+8\sum_{p<q,q>n_1}|b_{pq}^{(d+N)}|^2\\
&\le \mathrm{off}(\widehat{S}_{11}^{(d+N)})^2+4\mathrm{off}(B^{(d)})^2\left(\widehat{\sum\sin^2\theta_\ell}\right)\\
&\le \mathrm{off}(\widehat{S}_{11}^{(d+N)})^2+4\mathrm{off}(B^{(d)})^2\frac{\mathrm{off}(B^{(d)})^2}{2\delta^2}\\
&\le \frac{\mathrm{off}(S^{(d)})^4}{9\delta^2}+\frac{\mathrm{off}(S^{(d)})^4}{8\delta^2}.
\end{aligned}
$$

This shows that Algorithm 3.8 converges quadratically when there is only one multiple singular value.

Next, we show the quadratic convergence of Algorithm 3.8 for the case of more than one multiple singular values. If there exist l multiple singular values, then we have

$$
\mathrm{off}(S^{(d+N)})^2\le \frac{9+8l}{72\delta^2}\mathrm{off}(S^{(d)})^4. \tag{3.23}
$$

In the following we decrease the factor $\frac{9+8l}{72}$. Assume that σ_{φ_w} is a multiple singular value of $A^{(\ell)}$ with multiplicity n_w for $w = 1, \ldots, l$ and $s^{(\ell)}_{11}, \ldots, s^{(\ell)}_{n_1 n_1}$, $s^{(\ell)}_{n_1+1,n_1+1}, \ldots, s^{(\ell)}_{n_1+n_2,n_1+n_2}$, …, $s^{(\ell)}_{n_1+\cdots+n_{l-1}+1,n_1+\cdots+n_{l-1}+1}, \ldots, s^{(\ell)}_{n_1+\cdots+n_l,n_1+\cdots+n_l}$ converge to $\sigma^2_{\varphi_1}$, $\sigma^2_{\varphi_2}$, …, $\sigma^2_{\varphi_l}$ accordingly. Then, by appropriate row and column interchanges, we get a permutation matrix P such that

$$\widehat{S}^{(\ell)} = P^T S^{(\ell)} P = \begin{bmatrix} \widehat{S}^{(\ell)}_{11} & \widehat{S}^{(\ell)}_{12} & \cdots & \widehat{S}^{(\ell)}_{1l} & \widehat{S}^{(\ell)}_{1,l+1} \\ \widehat{S}^{(\ell)}_{21} & \widehat{S}^{(\ell)}_{22} & \cdots & \widehat{S}^{(\ell)}_{2l} & \widehat{S}^{(\ell)}_{2,l+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \widehat{S}^{(\ell)}_{l1} & \widehat{S}^{(\ell)}_{l2} & \cdots & \widehat{S}^{(\ell)}_{ll} & \widehat{S}^{(\ell)}_{l,l+1} \\ \widehat{S}^{(\ell)}_{l+1,1} & \widehat{S}^{(\ell)}_{l+1,2} & \cdots & \widehat{S}^{(\ell)}_{l+1,l} & \widehat{S}^{(\ell)}_{l+1,l+1} \end{bmatrix},$$

where the diagonal entries of $\widehat{S}^{(\ell)}_{ww} \in \mathbb{R}^{4n_w \times 4n_w}$ converges to $\sigma^2_{\varphi_w}$ for $w = 1, \ldots, l$.

Define the quantities

$$\Phi^{(\ell)}_w := \sqrt{\sum_{\sum_{u=0}^{w-1} 4n_u + 1 \le p \ne q \le \sum_{u=0}^{w} 4n_u} \left(\hat{s}^{(\ell)}_{p,q}\right)^2}, \quad w = 1, \ldots, l,$$

where $n_0 = 0$.

Analogous to the proof of (3.14a) we have

$$(\Phi^{(\ell)}_w)^2 \le \frac{2\mathrm{off}(S^{(\ell)})^2}{9\delta^2} \sum_{u=1,u\ne w}^{l+1} \|\widehat{S}^{(\ell)}_{wu}\|_F^2.$$

We note that

$$\sum_{w=1}^{l} \Big(\sum_{u=1,u\ne w}^{l+1} \|\widehat{S}^{(\ell)}_{wu}\|_F^2 \Big) \le \sum_{1\le p\ne q\le 4n} (\hat{s}^{(\ell)}_{pq})^2 = \mathrm{off}(\widehat{S}^{(\ell)})^2 = \mathrm{off}(S^{(\ell)})^2.$$

Hence,

$$\sum_{w=1}^{l} (\Phi^{(\ell)}_w)^2 \le \frac{2\mathrm{off}(S^{(\ell)})^2}{9\delta^2} \sum_{w=1}^{l} \Big(\sum_{u=1,u\ne w}^{l+1} \|\widehat{S}^{(\ell)}_{wu}\|_F^2 \Big) \le \frac{2\mathrm{off}(S^{(\ell)})^4}{9\delta^2} \le \frac{2\mathrm{off}(S^{(d)})^4}{9\delta^2}.$$

Therefore, (3.23) is reduced to

$$\begin{aligned} \mathrm{off}(S^{(d+N)})^2 &\le \sum_{t=1}^{l} (\Phi^{(d+N)}_t)^2 + \mathrm{off}(B^{(d)})^2 \Big(\widehat{\sum \sin^2 \theta_\ell}\Big) \\ &\le \frac{2\mathrm{off}(S^{(d)})^4}{9\delta^2} + \frac{\mathrm{off}(S^{(d)})^4}{8\delta^2} \\ &\le \frac{25}{72} \cdot \frac{\mathrm{off}(S^{(d)})^4}{\delta^2}. \end{aligned} \tag{3.24}$$

That is,

$$\mathrm{off}(\Gamma^T_{A^{(d+N)}} \Gamma_{A^{(d+N)}}) \le \sqrt{\frac{25}{72}} \cdot \frac{\mathrm{off}(\Gamma^T_{A^{(d)}} \Gamma_{A^{(d)}})^2}{\delta},$$

which shows that Algorithm 3.8 is quadratically convergent when there exist more than one multiple singular values. □

From Theorem 3.11 and [32,34], we see that both Algorithm 3.8 and the classic cyclic one-side Jacobi algorithm converge quadratically. However, for Γ_A, unlike the classic cyclic one-side Jacobi algorithm, our method is structure-preserving.

Remark 3.12. In Theorem 3.11, we obtain a factor of $\sqrt{25/72}$, which is much smaller than the factor $\sqrt{17/9}$ in [32] and the factor 1 in [34].

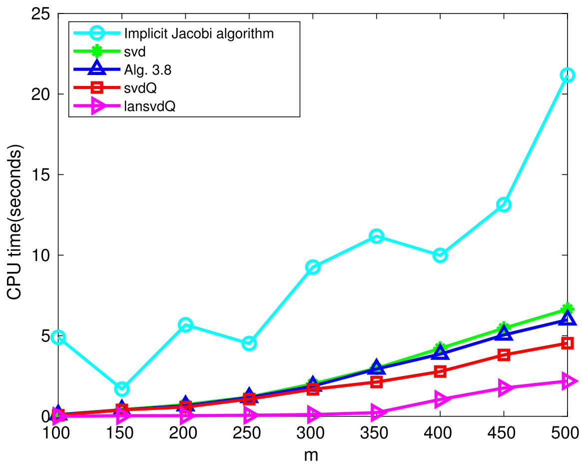

Fig. 1. Numerical results for Example 4.2.

Finally, we point out that, for an $m \times n$ quaternion matrix $A = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{H}^{m\times n}$, Algorithm 3.8 generates a matrix $\Gamma_{A^{(\Theta)}}$ after Θ orthogonal JRS-symplectic Jacobi updates such that $A^{(\Theta)} = AV^{(\Theta)}$ has sufficiently orthogonal columns (which is measured by $\mathrm{off}(\Gamma^T_{A^{(\Theta)}}\Gamma_{A^{(\Theta)}}) \le \mathtt{tol} \cdot \|\Gamma^T_A\Gamma_A\|_F$ for a prescribed tolerance $\mathtt{tol} > 0$), where $V^{(\Theta)} = G^{(0)}G^{(1)}\cdots G^{(\Theta)} \in \mathbb{H}^{n\times n}$ is a unitary matrix. Then the QSVD of A follows from column scaling of $A^{(\Theta)} = AV^{(\Theta)}$, i.e.,

$$A^{(\Theta)} = AV^{(\Theta)} = U^{(\Theta)}\Sigma^{(\Theta)}, \tag{3.25}$$

where $\Sigma^{(\Theta)} = \mathrm{diag}(\sigma^{(\Theta)}_{\varphi_1}, \sigma^{(\Theta)}_{\varphi_2}, \ldots, \sigma^{(\Theta)}_{\varphi_n})$ with $\sigma^{(\Theta)}_{\varphi_w} \ge 0$ for $w = 1, \ldots, n$ and $U^{(\Theta)} \in \mathbb{H}^{m\times n}$ is such that $(U^{(\Theta)})^*U^{(\Theta)} = I_n$. In addition, Algorithm 3.8 aims to compute all the singular values and associated left and right singular vectors of the $m \times n$ quaternion matrix A. Finally, the inherent parallelism of Algorithm 3.8 is more attractive, which need further study.

4. Numerical examples

In this section, we present some numerical examples to illustrate the effectiveness of Algorithm 3.8 for computing the SVD of a rectangular quaternion matrix and compare it with the Lanczos-based method (`lansvdQ`) [13], the structure-preserving method (`svdQ`) [24] and the implicit Jacobi algorithm in [22]. All the numerical tests were carried out in `MATLAB R2018a` running on a workstation of a Intel Xeon CPU Gold 6134 at 3.2 GHz and 32 GB of RAM.

Example 4.1. Let $A = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{H}^{8\times 5}$, where A_0, A_1, A_2, A_3 are all random real matrices, which are generated by the `MATLAB` built-in function `randn`.

We apply Algorithm 3.8 and the solver `svd` in quaternion toolbox [30] to Example 4.1. We repeat our experiments over 100 different test matrices. We observe that the computed singular values by both algorithms are the same numerically and the averaged CPU time (in seconds) taken by Algorithm 3.8 and the solver `svd` is about 0.0203 seconds and 0.0276 seconds, respectively.

Example 4.2. Let $A \in \mathbb{H}^{m\times n}$ be a random quaternion matrix with the fixed rank $\mathrm{rank}(A) = 5$, where m ranges from 100 to 500 with increment of 50 and $n = m/5$.

For Example 4.2, we compare the numerical effectiveness of the following four state-of-the-art algorithms and Algorithm 3.8.

- `Implicit Jacobi algorithm`: Cyclic classical one-sided Jacobi method in quaternion arithmetic in [22];
- `svd`: the solver in an open-source quaternion and octonion toolbox for `MATLAB` in [30];
- `svdQ`: The structure-preserving bidiagonalization method [24];
- `lansvdQ`: An iterative algorithm based on the Lanczos bidiagonalization in [13].

Figs. 1 and 2 show, respectively, the CPU time and the relative residual $\|AV^{(\Theta)} - U^{(\Theta)}\Sigma^{(\Theta)}\|_F/\|A\|_F$ at the final iterate of the corresponding algorithms for different quaternion matrix sizes.

We can see from Fig. 1 that `svd`, `svdQ`, `lansvdQ`, and Algorithm 3.8 are more efficient than `Implicit Jacobi algorithm` as the matrix size becomes larger. Also, Algorithm 3.8 is a little more effective than `svd` and `lansvdQ` is the

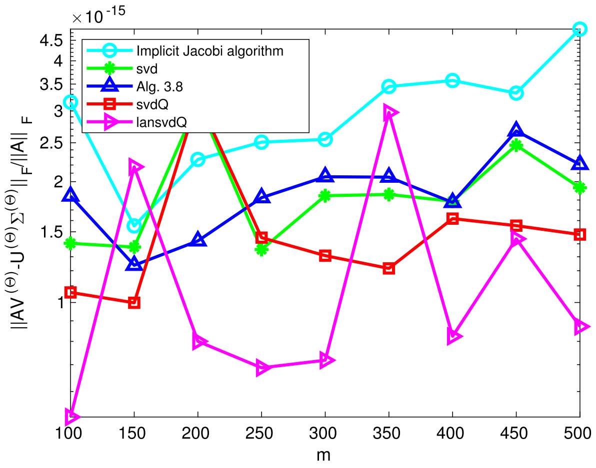

Fig. 2. Numerical results for Example 4.2.

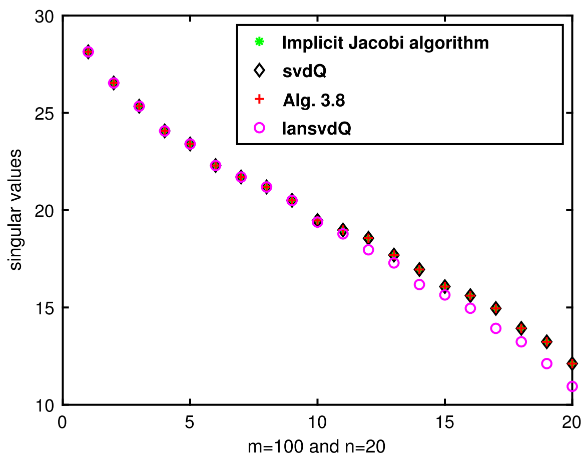

Fig. 3. Computed singular values.

most effective since `lansvdQ` only computes the r dominant SVD triplets. Fig. 2 shows that all the five algorithms find the solution with high accuracy (i.e., all the relative residuals are less than 1.0×10^{-14}).

To further illustrate the effectiveness of Algorithm 3.8, Fig. 3 depicts the computed singular values of a 100×20 full rank quaternion matrix by applying `Implicit Jacobi algorithm`, `svdQ`, Algorithm 3.8 and `lansvdQ`. We see from Fig. 3 that all algorithms obtain almost the same singular values except `lansvdQ`.

Example 4.3. An important application of the QSVD is image compression [28,29]. A color image can be represented by a pure quaternion matrix $A = [a_{ij}]_{m\times n} = R\mathbf{i} + G\mathbf{j} + B\mathbf{k}$, where R, G, B represent the red, green, blue parts of the color image. For demonstration purpose, we consider the color image compression for the color images Snowberg, Rabbit and Eiffel Tower (Eiffel) (see Figure 5(a)), whose sizes are 50×50, 50×50 and 50×100, accordingly.

We apply Algorithm 3.8 to Example 4.3. Fig. 4 shows the singular values of the original three color images Snowberg, Rabbit and Eiffel, accordingly. We can see that the singular values of these images decay very fast.

For each color image A, in Example 4.3, we use Algorithm 3.8 to compute its SVD such that $AV^{(\Theta)} = U^{(\Theta)}\Sigma^{(\Theta)}$, where $U^{(\Theta)}$, $\Sigma^{(\Theta)}$, and $V^{(\Theta)}$ are given by (3.25). Then we can compress the image by a lower-rank matrix approximation:

$$A_k = \sum_{w=1}^{k} \sigma_w^{(\Theta)} \mathbf{u}_w^{(\Theta)} (\mathbf{v}_w^{(\Theta)})^*, \tag{4.1}$$

where $\{\sigma_w^{(\Theta)}\}_{w=1}^k$ are the k largest singular values of A, $\mathbf{u}_w^{(\Theta)}$ and $\mathbf{v}_w^{(\Theta)}$ are the left and right singular vectors of A corresponding to $\sigma_w^{(\Theta)}$ for $w = 1, \dots, k$.

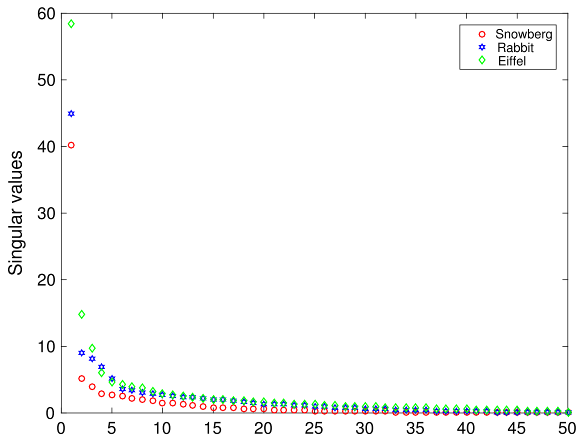

Fig. 4. Singular values of the color images Snowberg, Rabbit, and Eiffel.

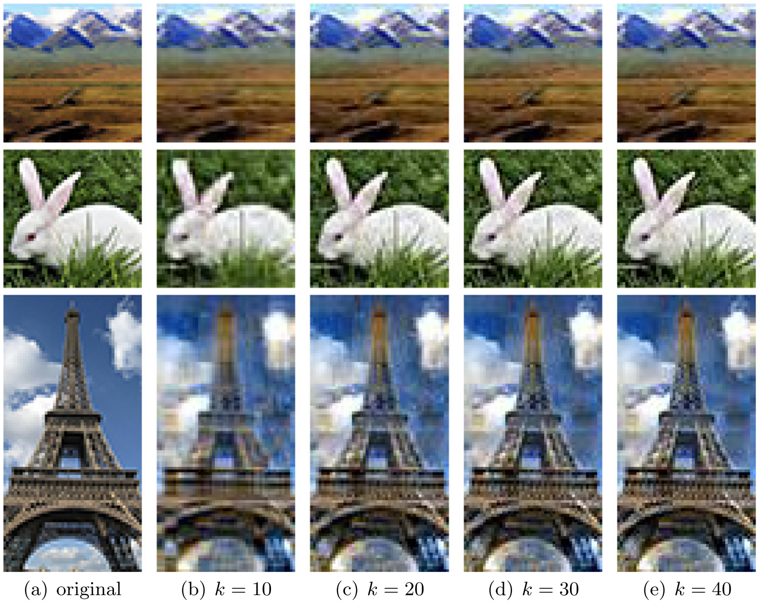

Fig. 5. Original images and compressed images for $k = 10, 20, 30, 40$.

Fig. 5 displays the original images and the compressed images for $k = 10, 20, 30$, and 40. We observe from Fig. 5 that small k already provides good estimations of the original color images. Meanwhile, the storage requirements drop from $3mn$ to $k(4m + 4n + 1)$.

To further illustrate the effectiveness of our algorithm, we check the image quality for the compressed images, which are estimated by using Algorithm 3.8, `lansvdQ`, `svd`, and `svdQ` to Example 4.3. The peak signal-to-noise ratios (PSNRs) of the compressed images are listed in Table 1 for different k. The PSNR between the original image f and a test image g, both of size $m \times n$, is defined by:

$$PSNR(f, g) = 10 \log_{10} \left(\frac{255^2}{MSE(f, g)} \right),$$

where MSE means the mean squared error defined by

$$MSE(f, g) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (f_{ij} - g_{ij})^2.$$

From Table 1, we see that Algorithm 3.8 gives almost the same PSNR as `lansvdQ`, `svd`, and `svdQ`.

Table 1
Comparative results of different algorithms.

Image	Method	$k = 10$	$k = 20$	$k = 30$	$k = 40$
Snowberg	Algorithm 3.8	35.0174	36.2627	36.4838	36.4980
	`lansvdQ`	35.0174	36.2627	36.4838	36.4980
	`svd`	35.0174	36.2627	36.4838	36.4980
	`svdQ`	35.0174	36.2627	36.4838	36.4980
Rabbit	Algorithm 3.8	30.5390	34.8253	36.9304	37.3062
	`lansvdQ`	30.5390	34.8253	36.9304	37.3062
	`svd`	30.5390	34.8253	36.9304	37.3062
	`svdQ`	30.5390	34.8253	36.9304	37.3062
Eiffel	Algorithm 3.8	29.3831	32.3879	34.0893	34.6324
	`lansvdQ`	29.3831	32.3879	34.0893	34.6357
	`svd`	29.3831	32.3879	34.0893	34.6324
	`svdQ`	29.3831	32.3879	34.0893	34.6324

5. Conclusions

In this paper, we have proposed a real structure-preserving one-sided cyclic Jacobi algorithm for computing the QSVD. This algorithm involves a sequence of column orthogonalizations in pairs via a sequence of orthogonal JRS-symplectic Jacobi rotations to the real counterpart of a quaternion matrix. The quadratic convergence is established especially. We also report some numerical results to illustrate the effectiveness of our algorithm. An interesting question is how to implement the proposed algorithm in parallel, which is the focus of our future work.

Acknowledgement

We are very grateful to the editor and the anonymous referees for their valuable comments.

References

[1] A.I. Arbab, The quaternionic quantum mechanics, Appl. Phys. Res. 3 (2011) 160–170.
[2] R. Brent, F. Luk, The solution of singular value and symmetric eigenvalue problems on multiprocessor arrays, SIAM J. Sci. Stat. Comput. 6 (1985) 69–84.
[3] A.J. Davies, B.H.J. Mckellar, Observability of quaternionic quantum-mechanics, Phys. Rev. A 46 (1992) 3671–3675.
[4] E. Doukhnitch, E. Ozen, Hardware-oriented algorithm for quaternion-valued matrix decomposition, IEEE Trans. Circuits Syst. II, Express Briefs 58 (2011) 225–229.
[5] S. Enshaeifar, S. Kouchaki, C.C. Took, S. Sanei, Quaternion singular spectrum analysis of electroencephalogram with application in sleep analysis, IEEE Trans. Neural Syst. Rehabil. Eng. 24 (2016) 57–67.
[6] D. Finkelstein, J.M. Jauch, D. Speiser, Notes on Quaternion Quantum Mechanics, CERN, Report 59-7, Logico-Algebraic Approach to Quantum Mechanics II, 1979, pp. 367–421.
[7] S. Gai, G.W. Yang, M.H. Wan, L. Wang, Denoising color images by reduced quaternion matrix singular value decomposition, Multidimens. Syst. Signal Process. 26 (2015) 307–320.
[8] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.
[9] W.R. Hamilton, On quaternions, in: Proceeding of the Royal Irish Academy, November 11, 1844.
[10] W.R. Hamilton, Elements of Quaternions, Longmans, Green and Co., London, 1866.
[11] Z.G. Jia, M.S. Wei, S.T. Ling, A new structure-preserving method for quaternion Hermitian eigenvalue problems, J. Comput. Appl. Math. 239 (2013) 12–24.
[12] Z.G. Jia, S.T. Ling, M.X. Zhao, Color two-dimensional principal component analysis for face recognition based on quaternion model, Lect. Notes Comput. Sci. 10361 (2017) 177–189.
[13] Z.G. Jia, M.K. Ng, G.J. Song, Lanczos method for large-scale quaternion singular value decomposition, Numer. Algorithms (2018) 1–19.
[14] Z.G. Jia, M.S. Wei, M.X. Zhao, Y. Chen, A new real structure-preserving quaternion QR algorithm, J. Comput. Appl. Math. 343 (2018) 26–48.
[15] Z.G. Jia, M.K. Ng, G.J. Song, Robust quaternion matrix completion with applications to image inpainting, Numer. Linear Algebra Appl. 26 (2019) e2245.
[16] Z.G. Jia, M.K. Ng, W. Wang, Color image restoration by saturation-value (SV) total variation, SIAM J. Imaging Sci. 12 (2019) 972–1000.
[17] T. Jiang, Algebraic methods for diagonalization of a quaternion matrix in quaternionic quantum theory, J. Math. Phys. 46 (2005) 052106.
[18] I.L. Kantor, A.S. Solodovnikov, Hypercomplex Numbers, An Elementary Introduction to Algebras, Springer, Berlin, 1989.
[19] N. Le Bihan, Traitement algébrique des signaux vectoriels. Application en séparation d'ondes sismiques, PhD Thesis, INPG, 2001.
[20] N. Le Bihan, J. Mars, Singular value decomposition of matrices of quaternions: a new tool for vector-sensor signal processing, Signal Process. 84 (2004) 1177–1199.
[21] N. Le Bihan, S.J. Sangwine, Quaternion principal component analysis of color images, in: Proc. Int. Conf. Image Process., ICIP 2003, vol. 1, 2003, pp. I-809-1–I-809-12.
[22] N. Le Bihan, S.J. Sangwine, Jacobi method for quaternion matrix singular value decomposition, Appl. Math. Comput. 187 (2007) 1265–1271.
[23] S.D. Leo, P. Rotelli, Quaternion scalar field, Phys. Rev. D 45 (1992) 575–579.
[24] Y. Li, M.S. Wei, F.X. Zhang, J.L. Zhao, A fast structure-preserving method for computing the singular value decomposition of quaternion matrices, Appl. Math. Comput. 235 (2014) 157–167.
[25] Y. Li, M.S. Wei, F.X. Zhang, J.L. Zhao, Real structure-preserving algorithms of Householder based transformations for quaternion matrices, J. Comput. Appl. Math. 305 (2016) 82–91.
[26] F. Liu, L.H. Ma, C. Liu, Z.M. Lu, Optimal blind watermarking for color images based on the U matrix of quaternion singular value decomposition, Multimed. Tools Appl. 77 (2018) 23483–23500.

[27] R.-R. Ma, Z.G. Jia, Z.-J. Bai, A structure-preserving Jacobi algorithm for quaternion Hermitian eigenvalue problems, Comput. Math. Appl. 75 (2018) 809–820.
[28] S.C. Pei, J.H. Cheng, J.J. Ding, Quaternion matrix singular value decomposition and its applications for color image processing, in: Proc. 2003 Int. Conf. Image Processing, ICIP 2003, vol. 1, 2003, pp. 805–808.
[29] H.S. Prasantha, H.L. Shashidhara, K.N. Balasubramanya Murthy, Image compression using SVD, in: Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, IEEE Computer Society, 2007, pp. 143–145.
[30] S.J. Sangwine, N.L. Bihan, Quaternion toolbox for Matlab, [online], Software library. Available: http://qtfm.sourceforge.net/.
[31] S.J. Sangwine, N. Le Bihan, Quaternion singular value decomposition based on bidiagonalization to a real or complex matrix using quaternion Householder transformations, Appl. Math. Comput. 182 (2006) 727–738.
[32] H.P.M. van Kempen, On the quadratic convergence of the special cyclic Jacobi method, Numer. Math. 9 (1966) 19–22.
[33] M.H. Wang, M.S. Wei, Y. Feng, An iterative algorithm for least squares problem in quaternionic quantum theory, Comput. Phys. Commun. 179 (2008) 203–207.
[34] J.H. Wilkinson, Note on the quadratic convergence of the cyclic Jacobi process, Numer. Math. 4 (1962) 296–300.
[35] J.H. Wilkinson, The QR algorithm for real symmetric matrices with multiple eigenvalues, Comput. J. 8 (1965) 85–87.
[36] Y. Xu, L. Yu, H. Xu, T. Nguyen, Vector sparse representation of color image using quaternion matrix analysis, IEEE Trans. Image Process. 24 (2015) 1315–1329.
[37] C.C. Yi, Y. Lv, Z. Dang, H. Xiao, X. Yu, Quaternion singular spectrum analysis using convex optimization and its application to fault diagnosis of rolling bearing, Measurement 103 (2017) 321–332.
[38] F.Z. Zhang, Quaternions and matrices of quaternions, Linear Algebra Appl. 251 (1997) 21–57.