S©MR

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ
Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru
ISSN 1813-3304

Том 21, № 2, стр. 144–145 (2024) УДК 519.61
https://doi.org/10.33048/semi.2024.21.??? MSC 15B33

AN EFFICIENT COMPLEX STRUCTURE-PRESERVING ALGORITHM FOR SCHUR DECOMPOSITION OF QUATERNION MATRIX

ZHENWEI GUO

Abstract: Schur decomposition is a fundamental and powerful matrix decomposition tool in linear algebra, which plays a key role in theoretical studies and practical applications such as eigenvalue computation. In this paper, by the complex representation form of a quaternion matrix and its properties, a complex structure-preserving implicit double-shift QR algorithm is proposed for solving the Schur decomposition problem of a quaternion matrix. Numerical examples show that the proposed algorithm obtains a significant improvement in computational efficiency.

Keywords: complex structure-preserving algorithm, QR algorithm, quaternion matrix, Schur decomposition, eigenvalue problem.

Guo, Z., An efficient complex structure-preserving algorithm for Schur decomposition of quaternion matrix.

In this work, the Section 1 is supported by the Ministry of Science and Higher Education of the Russian Federation, supplementary agreement (No. 075-02-2024-1441, February 28, 2024), the Sections 2-5 are supported by the Russian Science Foundation grant (No. 23-41-00037, No. 23-71-30013) and the Chinese Government Scholarship (CSC No. 202108370087).

Поступила 15 августа 2024 г., опубликована ? ? 2024 г.

1 Introduction

In numerical linear algebra, Schur decomposition is a fundamental and important tool that is widely used in the fields of eigenvalue problems [1], matrix function computation [2], and system stability analysis [3]. For a given square matrix, Schur decomposition decomposes it into a product of a unitary matrix and an upper triangular matrix. The decomposition not only provides a stable and efficient basis for many numerical algorithms, but has also become a standard method for solving complex problems in many fields such as signal processing, control theory [4, 5].

With the increasing demand for scientific computation, the object of study is no longer limited to real or complex matrices, and the problem of matrix computation in four dimensional algebra has also received widespread attention [6, 7, 8, 9], especially the quaternion algebra, as a complex mathematical structure, has increasingly appeared in the fields of quantum computation, and image and signal processing [10, 11, 12]. The concept of quaternion was first introduced by Irish mathematician William Rowan Hamilton in 1843 [13] and is usually expressed in the following form:

$$q = q_1 + q_2\mathrm{i} + q_3\mathrm{j} + q_4\mathrm{k}, \quad \mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1, \tag{1.1}$$

where q_1, q_2, q_3, q_4 are real numbers, $\mathrm{ij} = -\mathrm{ji} = \mathrm{k}$, $rmjk = -\mathrm{kj} = \mathrm{i}$, $\mathrm{ki} = -\mathrm{ik} = \mathrm{j}$. This non-commutativity ($\mathrm{ij} \neq \mathrm{ji}$) distinguishes quaternions from complex numbers and gives them special algebraic properties. Quaternions were originally introduced to solve rotation problems in three-dimensional space. Unlike Euler angles and rotation matrices, quaternions avoid the problem of gimbal blocking and are more efficient and stable in computing rotation problems [14]. These advantages have led to the widespread use of quaternions in modern computer graphics, robotics, and aerospace engineering [15, 16, 17].

At the same time, it is due to the structural complexity of quaternions that makes traditional matrix computation methods difficult to be directly applied to quaternion matrices, making the study of quaternion matrix algorithms more difficult and challenging. So far, the problem of computing the Schur decomposition and eigenvalues of quaternion matrices mainly includes the following two methods: the quaternion QR algorithm [18] and the real (complex) representation method [19, 20, 21]. In the paper [18], Bunse-Gerstner et al. proposed a double implicit shift QR algorithm for quaternion matrices, and then Jia et al. [22] established a real structure-preserving QR algorithm taking into account the properties of the real representation structure of the quaternion matrices, which transforms the quaternion matrix computation into real matrix operations, greatly improving the computational efficiency and reducing the computational cost. The establishment of structure-preserving algorithms has received extensive attention in the field of quaternion matrix computation and applications, including Hermtian matrix eigenvalue algorithms, singular value decomposition, LU decomposition, QR decomposition algorithms, etc [23, 24, 25, 26]. Although the above algorithms have gained some improvement in computational speed compared with the direct algorithms, the optimization of the algorithms has not yet reached the same maturity as the real matrix algorithms, and there is still a lot of room for improvement. In this paper, we further focus on the Schur decomposition problem of the quaternion matrices, and propose a complex structure-preserving QR iterative algorithm, which accelerates the convergence of the matrix decomposition by transforming the quaternion matrices into the complex matrices and applying the complex structure-preserving implicit double-shift strategy. Compared with the traditional methods, the proposed algorithm can significantly improve the computational efficiency while maintaining the intrinsic structure of the quaternionic matrices and has a simpler form.

The paper is organized as follows. In Section 2, we introduce the definitions and properties of quaternions to be used in the following. In Section 3, we investigate the complex structure-preserving Schur decomposition algorithm based on the double implicit shift QR algorithm. In Section 4, we verify the effectiveness of the proposed algorithm via numerical examples.

2 Preliminaries

In this section, we will show some definitions and properties that will be needed below.

Let $\mathbf{H}$, $\mathbf{R}$ and $\mathbf{C}$ denote the skew field of quaternions, the real and complex number field, respectively. The conjugate and norm of the a quaternion $q = q_1 + q_2\mathrm{i} + q_3\mathrm{j} + q_4\mathrm{k}$ are defined as $\overline{q} = q_1 - q_2\mathrm{i} - q_3\mathrm{j} - q_4\mathrm{k}$ and $\|q\| = \sqrt{q\overline{q}} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}$, respectively. $\mathrm{Re}(q) = q_1$ denotes the real part of q. The conjugate and conjugate transpose of a quaternion matrix $A = A_1 + A_2\mathrm{i} + A_3\mathrm{j} + A_4\mathrm{k} \in \mathbf{H}^{m\times n}$ are defined as $\overline{A} = A_1 - A_2\mathrm{i} - A_3\mathrm{j} - A_4\mathrm{k}$ and $A^H = A_1^T - A_2^T\mathrm{i} - A_3^T\mathrm{j} - A_4^T\mathrm{k}$, respectively. For a quaternion matrix $A \in \mathbf{H}^{n\times n}$, then A is a unitary quaternion matrix if $AA^H = A^HA = I_n$, and A is a Hermitian quaternion matrix if $A = A^H$.

Lemma 1.1[19] Let $A \in \mathbf{H}^{n\times n}$ be a quaternion matrix. Then there exists a unitary quaternion matrix $U \in \mathbf{H}^{n\times n}$ such that U^HAU is upper triangular matrix.

Complex representation for a quaternion matrix. For any quaternion matrix $A = A_1+A_2\mathrm{i}+A_3\mathrm{j}+A_4\mathrm{k} = \mathbb{A}_1+\mathbb{A}_2\mathrm{j} \in \mathbf{H}^{m\times n}$, where $A_1, A_2, A_3, A_4 \in \mathbf{R}^{m\times n}$ and $\mathbb{A}_1 = A_1 + A_2\mathrm{i}, \mathbb{A}_2 = A_3 + A_4\mathrm{i}$, its complex representation A^C was defined as follows [27],

$$A^C = \begin{bmatrix} \mathbb{A}_1 & \mathbb{A}_2 \\ -\overline{\mathbb{A}_2} & \overline{\mathbb{A}_1} \end{bmatrix} \in \mathbf{C}^{2m\times 2n}. \tag{2.1}$$

The complex representation matrix have the following properties,

$$A^C = [A_c^C, Q_m^T\overline{A_c^C}], \quad \overline{A^C} = Q_m^TA^CQ_n, \tag{2.2}$$

where $A_c^C = \begin{bmatrix} A_1 + A_2\mathrm{i} \\ -A_3 + A_4\mathrm{i} \end{bmatrix} \in \mathbf{C}^{2m\times n}$, $Q_t = \begin{bmatrix} 0 & I_t \\ -I_t & 0 \end{bmatrix}$ is a unitary matrix, I_t is a unit matrix of order t.

Moreover, for any $A, B \in \mathbf{H}^{m\times n}, D \in \mathbf{H}^{n\times p}, a \in \mathbf{R}$, then we have

$$(A+B)^C = A^C + B^C, (AD)^C = A^C D^C, (aA)^C = aA^C, (A^H)^C = (A^C)^H. \quad (2.3)$$

From the Equation (2.3) above, it is easy to derive that $A \to A^C$ is an isomorphic mapping of $\mathbf{H}^{m\times n}$ on $\mathbf{C}^{2m\times 2n}$. For $A \in \mathbf{H}^{n\times n}$, then A is a Hermitian quaternion matrix if and only if A^C is a Hermitian matrix.

The Frobenius norm of a quaternion matrix. [28] Let $A = A_1 + A_2\mathrm{i} + A_3\mathrm{j} + A_4\mathrm{k} \in \mathbf{H}^{m\times n}$. Then

$$\|A\|_F = \|A_c^C\|_F = \sqrt{\|A_1 + A_2\mathrm{i}\|_F^2 + \| - A_3 + A_4\mathrm{i}\|_F^2}. \quad (2.4)$$

Generalized Givens matrix. Given a quaternion $q = q_1 + q_2\mathrm{i} + q_3\mathrm{j} + q_4\mathrm{k} \in \mathbf{H}$, where $q_1, q_2, q_3, q_4 \in \mathbf{R}$, if $q \neq 0$, then its generalized Givens matrix G_1 can be defined as follows

$$\begin{aligned} G_1 &= \begin{bmatrix} \cos\alpha_1 + \mathrm{i}\times\cos\alpha_2 & \cos\alpha_3 + \mathrm{i}\times\cos\alpha_4 \\ -\cos\alpha_3 + \mathrm{i}\times\cos\alpha_4 & \cos\alpha_1 - \mathrm{i}\times\cos\alpha_2 \end{bmatrix} \\ &= \begin{bmatrix} q_1 + q_2\mathrm{i} & q_3 + q_4\mathrm{i} \\ -q_3 + q_4\mathrm{i} & q_1 - q_2\mathrm{i} \end{bmatrix} /|q| = q^C/|q|, \end{aligned} \quad (2.5)$$

then $G_1^H q^C = q^C G_1^H = |q| I_2$.

Therefore, G_1 is the special case of generalized Givens matrix below,

$$G_l = \left[\begin{array}{ccc|ccc} I_{l-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cos\alpha_1 + \mathrm{i}\times\cos\alpha_2 & \mathbf{0} & \mathbf{0} & \cos\alpha_3 + \mathrm{i}\times\cos\alpha_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{n-l} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & I_{l-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\cos\alpha_3 + \mathrm{i}\times\cos\alpha_4 & \mathbf{0} & \mathbf{0} & \cos\alpha_1 - \mathrm{i}\times\cos\alpha_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I_{n-l} \end{array}\right] \in \mathbf{C}^{2n\times 2n}. \quad (2.6)$$

where $l = 1, 2, \cdots, n$, $\cos\alpha_1^2 + \cos\alpha_2^2 + \cos\alpha_3^2 + \cos\alpha_4^2 = 1$ with $\alpha_s \in [-\frac{\pi}{2}, \frac{\pi}{2}), s = 1, 2, 3, 4.$

Generalized Householder matrix is the direct sum of two identical $n\times n$ Householder matrices defined as follows,

$$H_l = \begin{bmatrix} I_n - \beta\nu\nu^T & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & I_n - \beta\nu\nu^T \end{bmatrix} \in \mathbf{R}^{2n\times 2n}. \quad (2.7)$$

where $\nu \in \mathbf{R}^{n\times 1}$ is a Householder vector and its first $l-1$ elements are equal to 0, and β is a scalar satisfying $\beta(\beta\nu^T\nu - 2) = 0$. Usually, in practical calculations, in order to reduce the storage space of the computer and increase the speed of calculations, only ν and β need to be generated instead of H_l.

The algorithm for generating the generalized Givens matrix and the generalized Householder vector is shown below.

Generalized upper Hessenberg matrix. For a complex representation matrix $\mathbb{H} \in \mathbf{C}^{2n\times 2n}$, it is called a generalized Hessenberg matrix if it has the

Algorithm 1: For two complex numbers g_1 and g_2, then a 2×2 generalized Givens matrix can be generated as follows.

Function G_1=**QC_Givens**(g_1, g_2)
  if $g_1 = g_2 = 0$, **then**
    $G_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$;
  else
    $G_1 = \begin{bmatrix} g_1 & g_2 \\ -\overline{g_2} & \overline{g_1} \end{bmatrix} / \sqrt{g_1 \times \overline{g_1} + g_2 \times \overline{g_2}}$;
  end
End

Algorithm 2: For two complex vectors $x_1, x_2 \in \mathbf{C}^{n\times 1}$, then the generalized Householder vector can be generated as follows.

Function $[u, \beta]$=**QC_House**(x_1, x_2, n)
  $u_1(1:n, 1:2) = [\overline{x_1}, x_2]$; $a = \text{norm}([x_1; x_2])$; $x = \text{norm}([x_1(1), x_2(1)])$;
  if $x == 0$, **then**
    $\alpha = a * [1, 0]$;
  else
    $\alpha = -(a/x) * ([x_1(1), x_2(1)])$;
  end
  $u_1(1, 1:2) = u_1(1, 1:2) - \alpha$;
  $u = \text{Q\_Comp}(u_1(:,1), u_1(:,2))$; $\beta = 1/(a * (a + x))$;
End

following form

$$\mathbb{H} = \begin{bmatrix} \mathbb{H}_1 & \mathbb{H}_2 \\ -\overline{\mathbb{H}_2} & \overline{\mathbb{H}_1} \end{bmatrix}, \tag{2.8}$$

where $\mathbb{H}_1 \in \mathbf{C}^{n\times n}$ is an upper Hessenberg matrix, $\mathbb{H}_2 \in \mathbf{C}^{n\times n}$ is an upper triangular matrix. Moreover, if all subdiagonal elements of $\mathbb{H}_1$ are nonzeros, $\mathbb{H}$ is called an unreduced generalized upper Hessenberg matrix.

Generalized upper triangular matrix. For a complex representation matrix $\mathbb{R} \in \mathbf{C}^{2n\times 2n}$, it is called a generalized upper triangular matrix if it has the following form

$$\mathbb{R} = \begin{bmatrix} \mathbb{R}_1 & \mathbb{R}_2 \\ -\overline{\mathbb{R}_2} & \overline{\mathbb{R}_1} \end{bmatrix}, \tag{2.9}$$

where $\mathbb{R}_1 \in \mathbf{C}^{m\times n}$ is upper triangular, $\mathbb{R}_2 \in \mathbf{C}^{m\times n}$ is strictly upper triangular. Moreover, if $\mathbb{R}_1$ is also strictly upper triangular, then $\mathbb{R}$ is called a generalized strictly upper triangular matrix.

Generalized Schur form. For a complex representation matrix $\mathbb{T} \in \mathbf{C}^{2n\times 2n}$, it is called a generalized Schur form if it has the following form

$$\mathbb{T} = \begin{bmatrix} \mathbb{T}_1 & \mathbb{T}_2 \\ -\overline{\mathbb{T}_2} & \overline{\mathbb{T}_1} \end{bmatrix}, \tag{2.10}$$

where $\mathbb{T}_1 \in \mathbf{C}^{n\times n}$ is a quasi-upper triangular matrix, $\mathbb{T}_2 \in \mathbf{C}^{n\times n}$ is upper triangular matrix.

Proposition 2.1 For a complex representation matrix $\mathbb{M} \in \mathbf{C}^{2n\times 2n}$. Then there exists a unitary complex representation matrix $\mathbb{W} \in \mathbf{C}^{2n\times 2n}$ such that $\mathbb{W}\mathbb{M}\mathbb{W}^H = \mathbb{H}$ is a generalized upper Hessenberg matrix.

Proof. We can prove the proposition by mathematical induction. For $n = 1$, the proposition clearly holds. Suppose that for the case $1 \leq n < s$, there exists unitary complex representation matrix $\widetilde{\mathbb{W}} \in \mathbf{C}^{2n\times 2n}$ such that

$$\widetilde{\mathbb{W}}\mathbb{M}\widetilde{\mathbb{W}}^H = \begin{bmatrix} \widetilde{\mathbb{H}}_1 & \widetilde{\mathbb{H}}_2 \\ -\overline{\widetilde{\mathbb{H}}_2} & \overline{\widetilde{\mathbb{H}}_1} \end{bmatrix},$$

where $\widetilde{\mathbb{H}}_1 \in \mathbf{C}^{n\times n}$ is an upper Hessenberg matrix, $\widetilde{\mathbb{H}}_2 \in \mathbf{C}^{n\times n}$ is upper triangular matrix. For $n = s$, denote

$$\mathbb{M} = \begin{bmatrix} \mathbb{M}_1 & \mathbb{M}_2 \\ -\overline{\mathbb{M}_2} & \overline{\mathbb{M}_1} \end{bmatrix}
= \left[\begin{array}{ccccc|ccccc}
\psi_{11}^{(1)} & \psi_{12}^{(1)} & \psi_{13}^{(1)} & \cdots & \psi_{1s}^{(1)} & \psi_{11}^{(2)} & \psi_{12}^{(2)} & \psi_{13}^{(2)} & \cdots & \psi_{1s}^{(2)} \\
\psi_{21}^{(1)} & \psi_{22}^{(1)} & \psi_{23}^{(1)} & \cdots & \psi_{2s}^{(1)} & \psi_{21}^{(2)} & \psi_{22}^{(2)} & \psi_{23}^{(2)} & \cdots & \psi_{2s}^{(2)} \\
\psi_{31}^{(1)} & \psi_{32}^{(1)} & \psi_{33}^{(1)} & \cdots & \psi_{3s}^{(1)} & \psi_{31}^{(2)} & \psi_{32}^{(2)} & \psi_{33}^{(2)} & \cdots & \psi_{3s}^{(2)} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\psi_{s1}^{(1)} & \psi_{s2}^{(1)} & \psi_{s3}^{(1)} & \cdots & \psi_{ss}^{(1)} & \psi_{s1}^{(2)} & \psi_{s2}^{(2)} & \psi_{s3}^{(2)} & \cdots & \psi_{ss}^{(2)} \\
\hline
-\overline{\psi_{11}^{(2)}} & -\overline{\psi_{12}^{(2)}} & -\overline{\psi_{13}^{(2)}} & \cdots & -\overline{\psi_{1s}^{(2)}} & \overline{\psi_{11}^{(1)}} & \overline{\psi_{12}^{(1)}} & \overline{\psi_{13}^{(1)}} & \cdots & \overline{\psi_{1s}^{(1)}} \\
-\overline{\psi_{21}^{(2)}} & -\overline{\psi_{22}^{(2)}} & -\overline{\psi_{23}^{(2)}} & \cdots & -\overline{\psi_{2s}^{(2)}} & \overline{\psi_{21}^{(1)}} & \overline{\psi_{22}^{(1)}} & \overline{\psi_{23}^{(1)}} & \cdots & \overline{\psi_{2s}^{(1)}} \\
-\overline{\psi_{31}^{(2)}} & -\overline{\psi_{32}^{(2)}} & -\overline{\psi_{33}^{(2)}} & \cdots & -\overline{\psi_{3s}^{(2)}} & \overline{\psi_{31}^{(1)}} & \overline{\psi_{32}^{(1)}} & \overline{\psi_{33}^{(1)}} & \cdots & \overline{\psi_{3s}^{(1)}} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
-\overline{\psi_{s1}^{(2)}} & -\overline{\psi_{s2}^{(2)}} & -\overline{\psi_{s3}^{(2)}} & \cdots & -\overline{\psi_{ss}^{(2)}} & \overline{\psi_{s1}^{(1)}} & \overline{\psi_{s2}^{(1)}} & \overline{\psi_{s3}^{(1)}} & \cdots & \overline{\psi_{ss}^{(1)}}
\end{array}\right],$$

where $\psi_{uv}^{(1)}, \psi_{uv}^{(2)} \in \mathbf{C}$ and $u, v = 1, 2, \cdots, s$. Then find a series of generalized Givens matrices $G_2, G_3, \cdots, G_s$ such that

$$\breve{\mathbb{M}} = G_s \cdots G_3 G_2 \mathbb{M} (G_s \cdots G_3 G_2)^H
= \left[\begin{array}{ccccc|ccccc}
\psi_{11}^{(1)} & \breve{\psi}_{12}^{(1)} & \breve{\psi}_{13}^{(1)} & \cdots & \breve{\psi}_{1s}^{(1)} & \psi_{11}^{(2)} & \breve{\psi}_{12}^{(2)} & \breve{\psi}_{13}^{(2)} & \cdots & \breve{\psi}_{1s}^{(2)} \\
r_{21} & \breve{\psi}_{22}^{(1)} & \breve{\psi}_{23}^{(1)} & \cdots & \breve{\psi}_{2s}^{(1)} & 0 & \breve{\psi}_{22}^{(2)} & \breve{\psi}_{23}^{(2)} & \cdots & \breve{\psi}_{2s}^{(2)} \\
r_{31} & \breve{\psi}_{32}^{(1)} & \breve{\psi}_{33}^{(1)} & \cdots & \breve{\psi}_{3s}^{(1)} & 0 & \breve{\psi}_{32}^{(2)} & \breve{\psi}_{33}^{(2)} & \cdots & \breve{\psi}_{3s}^{(2)} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
r_{s1} & \breve{\psi}_{s2}^{(1)} & \breve{\psi}_{s3}^{(1)} & \cdots & \breve{\psi}_{ss}^{(1)} & 0 & \breve{\psi}_{s2}^{(2)} & \breve{\psi}_{s3}^{(2)} & \cdots & \breve{\psi}_{ss}^{(2)} \\
\hline
-\overline{\psi_{11}^{(2)}} & -\overline{\breve{\psi}_{12}^{(2)}} & -\overline{\breve{\psi}_{13}^{(2)}} & \cdots & -\overline{\breve{\psi}_{1s}^{(2)}} & \overline{\psi_{11}^{(1)}} & \overline{\breve{\psi}_{12}^{(1)}} & \overline{\breve{\psi}_{13}^{(1)}} & \cdots & \overline{\breve{\psi}_{1s}^{(1)}} \\
0 & -\overline{\breve{\psi}_{22}^{(2)}} & -\overline{\breve{\psi}_{23}^{(2)}} & \cdots & -\overline{\breve{\psi}_{2s}^{(2)}} & r_{21} & \overline{\breve{\psi}_{22}^{(1)}} & \overline{\breve{\psi}_{23}^{(1)}} & \cdots & \overline{\breve{\psi}_{2s}^{(1)}} \\
0 & -\overline{\breve{\psi}_{32}^{(2)}} & -\overline{\breve{\psi}_{33}^{(2)}} & \cdots & -\overline{\breve{\psi}_{3s}^{(2)}} & r_{31} & \overline{\breve{\psi}_{32}^{(1)}} & \overline{\breve{\psi}_{33}^{(1)}} & \cdots & \overline{\breve{\psi}_{3s}^{(1)}} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & -\overline{\breve{\psi}_{s2}^{(2)}} & -\overline{\breve{\psi}_{s3}^{(2)}} & \cdots & -\overline{\breve{\psi}_{ss}^{(2)}} & r_{s1} & \overline{\breve{\psi}_{s2}^{(1)}} & \overline{\breve{\psi}_{s3}^{(1)}} & \cdots & \overline{\breve{\psi}_{ss}^{(1)}}
\end{array}\right]$$

where $r_{t1} = \sqrt{(\psi_{t1}^{(1)})^2 + (\psi_{t1}^{(2)})^2}, t = 2, 3, \cdots, s$. Then we can get a generalized Householder matrix $H \in \mathbf{R}^{2s\times 2s}$ such that

$$\begin{aligned}\hat{\mathbb{M}} &= H_2 \breve{\mathbb{M}} H_2^T \\ &= \left[\begin{array}{ccccc|ccccc} \psi_{11}^{(1)} & \hat{\psi}_{12}^{(1)} & \hat{\psi}_{13}^{(1)} & \cdots & \hat{\psi}_{1s}^{(1)} & \psi_{11}^{(2)} & \hat{\psi}_{12}^{(2)} & \hat{\psi}_{13}^{(2)} & \cdots & \hat{\psi}_{1s}^{(2)} \\ \hat{r}_{21} & \hat{\psi}_{22}^{(1)} & \hat{\psi}_{23}^{(1)} & \cdots & \hat{\psi}_{2s}^{(1)} & 0 & \hat{\psi}_{22}^{(2)} & \hat{\psi}_{23}^{(2)} & \cdots & \hat{\psi}_{2s}^{(2)} \\ 0 & \hat{\psi}_{32}^{(1)} & \hat{\psi}_{33}^{(1)} & \cdots & \hat{\psi}_{3s}^{(1)} & 0 & \hat{\psi}_{32}^{(2)} & \hat{\psi}_{33}^{(2)} & \cdots & \hat{\psi}_{3s}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \hat{\psi}_{s2}^{(1)} & \hat{\psi}_{s3}^{(1)} & \cdots & \hat{\psi}_{ss}^{(1)} & 0 & \hat{\psi}_{s2}^{(2)} & \hat{\psi}_{s3}^{(2)} & \cdots & \hat{\psi}_{ss}^{(2)} \\ \hline -\overline{\psi_{11}^{(2)}} & -\overline{\hat{\psi}_{12}^{(2)}} & -\overline{\hat{\psi}_{13}^{(2)}} & \cdots & -\overline{\hat{\psi}_{1s}^{(2)}} & \overline{\psi_{11}^{(1)}} & \overline{\hat{\psi}_{12}^{(1)}} & \overline{\hat{\psi}_{13}^{(1)}} & \cdots & \overline{\hat{\psi}_{1s}^{(1)}} \\ 0 & -\overline{\hat{\psi}_{22}^{(2)}} & -\overline{\hat{\psi}_{23}^{(2)}} & \cdots & -\overline{\hat{\psi}_{2s}^{(2)}} & \hat{r}_{21} & \overline{\hat{\psi}_{22}^{(1)}} & \overline{\hat{\psi}_{23}^{(1)}} & \cdots & \overline{\hat{\psi}_{2s}^{(1)}} \\ 0 & -\overline{\hat{\psi}_{32}^{(2)}} & -\overline{\hat{\psi}_{33}^{(2)}} & \cdots & -\overline{\hat{\psi}_{3s}^{(2)}} & 0 & \overline{\hat{\psi}_{32}^{(1)}} & \overline{\hat{\psi}_{33}^{(1)}} & \cdots & \overline{\hat{\psi}_{3s}^{(1)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -\overline{\hat{\psi}_{s2}^{(2)}} & -\overline{\hat{\psi}_{s3}^{(2)}} & \cdots & -\overline{\hat{\psi}_{ss}^{(2)}} & 0 & \overline{\hat{\psi}_{s2}^{(1)}} & \overline{\hat{\psi}_{s3}^{(1)}} & \cdots & \overline{\hat{\psi}_{ss}^{(1)}} \end{array}\right]\end{aligned}$$

where $\hat{r}_{21} = \sqrt{r_{21}^2 + r_{21}^2 + \cdots + r_{s1}^2}$. Then the 1st, (s+1)-th rows and columns of $\hat{\mathbb{M}}$ are deleted to obtain a $2(s-1)\times 2(s-1)$ submatrix, which satisfies the complex representation form. Therefore, the proposition is proved based on the assumptions. □

Proposition 2.2 For a complex representation matrix $\mathbb{M} \in \mathbf{C}^{2n\times 2n}$. Then there exists a unitary complex representation matrix $\mathbb{W} \in \mathbf{C}^{2n\times 2n}$ such that $\mathbb{W}^H\mathbb{M} = \mathbb{R} \in \mathbf{C}^{2n\times 2n}$ is a generalized upper triangular matrix.

The Proposition 2.2 can be proved in a similar way with Proposition 2.1.

Proposition 2.3 Suppose that $\mathbb{M} \in \mathbf{C}^{2n\times 2n}$ is a complex representation matrix. Then there exists a unitary complex representation matrix $\mathbb{W} \in \mathbf{C}^{2n\times 2n}$ such that $\mathbb{W}^H\mathbb{M}\mathbb{W} = \mathbb{T} \in \mathbf{C}^{2n\times 2n}$ is a generalized Schur form.

The Proposition 2.3 can be proved in a similar way with Proposition 2.1.

3 Complex structure-preserving QR algorithm for Schur decomposition of quaternion matrices

Based on the properties of complex representation matrices of quaternion matrices and related definitions provided in the previous section, in this section we present a QR algorithm based on the complex representation form for computing the generalized Schur decomposition.

Let $\mathbb{M} \in \mathbf{C}^{2n\times 2n}$ be complex representation matrix, then a basic complex representation QR algorithm can be written as

$$\begin{aligned} &\mathbb{H} = \mathbb{V}\mathbb{M}\mathbb{V}^H \\ &\textbf{for } s = 1, 2, \cdots \\ &\quad \mathbb{H} = \mathbb{W}\mathbb{R} \quad \text{(QR decomposition)} \\ &\quad \mathbb{H} = \mathbb{R}\mathbb{W} \\ &\textbf{end} \end{aligned} \tag{3.1}$$

where each $\mathbb{V}, \mathbb{W} \in \mathbf{C}^{2n\times 2n}$ is unitary complex representation matrix and $\mathbb{R} \in \mathbf{C}^{2n\times 2n}$ is generalized upper triangular matrix. It can be noticed that by reducing $\mathbb{M}$ to $\mathbb{H}$, then the computation of each subsequent QR iteration step can be reduced from $O(n^3)$ to $O(n^2)$. But the problem remains that as the matrix dimension increases, the convergence of this algorithm is very slow and uncertain, and may not converge to a satisfactory result after many iterations. So in order to improve the convergence speed, here we can introduce the complex conjugate Wilkinson shifts as follows,

$$\begin{aligned}
&\quad \mathbb{H} = \mathbb{V}\mathbb{M}\mathbb{V}^H \\
&\textbf{for } s = 1, 2, \cdots \\
&\quad \text{Determine the complex Wilkinson shifts } k \text{ and } \overline{k}. \\
&\quad \mathbb{H} - kI = \mathbb{W}\mathbb{R} \quad \text{(QR decomposition)} \\
&\qquad \widetilde{\mathbb{H}} = \mathbb{R}\mathbb{W} + kI \\
&\quad \widetilde{\mathbb{H}} - \overline{k}I = \widetilde{\mathbb{W}}\widetilde{\mathbb{R}} \quad \text{(QR decomposition)} \\
&\qquad \hat{\mathbb{H}} = \widetilde{\mathbb{R}}\widetilde{\mathbb{W}} + \overline{k}I \\
&\textbf{end}
\end{aligned} \tag{3.2}$$

Next, if

$$\check{\mathbb{W}}\check{\mathbb{R}} = (\mathbb{H} - kI)(\mathbb{H} - \overline{k}I)$$

is the Generalized QR decomposition of $(\mathbb{H} - kI)(\mathbb{H} - \overline{k}I)$, then we can get the following result,

$$\hat{\mathbb{H}} = (\check{\mathbb{W}})^H \mathbb{H} \check{\mathbb{W}}. \tag{3.3}$$

The strategy is called the Francis double shift strategy. Moreover, since $\mathbb{C} = (\mathbb{H} - kI)(\mathbb{H} - \overline{k}I) = \mathbb{H}^2 - 2\mathrm{Re}(k)\mathbb{H} + |k|^2 I$, the formation of $\mathbb{C}$ requires $O(n^3)$ operations. Therefore it is necessary to avoid generating the matrix $\mathbb{C}$ thus less computationally expensive by using a properties of generalized Hesseberg matrices.

Next, similar to the paper [22] it is easy to prove the uniqueness and the implicit Q theorem of generalized upper Hessenberg matrices. Then the double implicit shift strategy for simultaneously determining $\check{\mathbb{W}}$ and $\hat{\mathbb{H}}$ can be resolved into the following five steps:

1. Compute the 1 and $n+1$ columns of $\mathbb{C} \in \mathbf{C}^{2n\times 2n}$, and save them into $\mathbb{F} \in \mathbf{C}^{2n\times 2}$.

2. Compute a generalized Householder matrix $\mathbb{W}_F \in \mathbf{C}^{2n\times 2n}$ such that

$$\mathbb{W}_F^H \mathbb{F} = \sigma[e_1, e_{n+1}], \tag{3.4}$$

where each e_s is the sth column of the identity matrix I, and $\sigma \in \mathbf{C}$.

3. Let $\mathbb{H}_F = \mathbb{W}_F^H \mathbb{H} \mathbb{W}_F$.

4. Reduce $\mathbb{H}_F$ to a generalized upper Hessenberg form $\hat{\mathbb{H}}$ using the generalized Householder matrix. Denote the cumulative transformation as $\tilde{\mathbb{W}}$.

5. Let $\check{\mathbb{W}} = \mathbb{W}_F \tilde{\mathbb{W}}$.

The primary computations involve determining the 1st and n+1th columns of $\mathbb{C}$ and reducing $\mathbb{H}_F$ to a generalized upper Hessenberg form. Since $\mathbb{H}$ already in a generalized upper Hessenberg form, the first calculation can be performed in $O(1)$ operations, while the second requires $O(n^2)$ operations.

The typical approach to solve the dense nonsymmetric eigenvalue problem involves first reducing the matrix to upper Hessenberg form, and then iteratively computing the Schur form using the Francis QR step. Here we demonstrate how a generalized upper Hessenberg matrix $\mathbb{H} \in \mathbf{C}^{2n\times 2n}$ can be reduced into a generalized Schur form $\mathbb{T} = \mathbb{W}^H\mathbb{H}\mathbb{W}$ using the unitary complex representation matrix $\mathbb{W}$.

Denote that $\mathbb{H} = \begin{bmatrix} \mathbb{H}_1 & \mathbb{H}_2 \\ -\overline{\mathbb{H}}_2 & \overline{\mathbb{H}}_1 \end{bmatrix}$, $\mathbb{W} = \begin{bmatrix} \mathbb{W}_1 & \mathbb{W}_2 \\ -\overline{\mathbb{W}}_2 & \overline{\mathbb{W}}_1 \end{bmatrix}$, and $\mathbb{T} = \begin{bmatrix} \mathbb{T}_1 & \mathbb{T}_2 \\ -\overline{\mathbb{T}}_2 & \overline{\mathbb{T}}_1 \end{bmatrix}$, in which $\mathbb{H}_1, \mathbb{H}_2, \mathbb{W}_1, \mathbb{W}_2, \mathbb{T}_1, \mathbb{T}_2 \in \mathbf{C}^{n\times n}$.

Firstly, determine the maximum nonnegative integer q and the minimum nonnegative integer p such that

$$\mathbb{H}_1 = \begin{bmatrix} \mathbb{H}_{11} & \mathbb{H}_{12} & \mathbb{H}_{13} \\ \mathbf{0} & \mathbb{H}_{22} & \mathbb{H}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbb{H}_{33} \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix} \quad , \tag{3.5}$$

where $\mathbb{H}_{33}$ is upper quasi-triangular and $\mathbb{H}_{22}$ is unreduced.

Next, if $q < n$, execute a Francis QR step on the unreduced generalized upper Hessenberg matrix $\mathbb{H}_{22}$:

$$\mathbb{H}_{22} = \check{\mathbb{W}}^H \mathbb{H}_{22} \check{\mathbb{W}}. \tag{3.6}$$

Let ϵ be the computer accuracy. The generalized Schur form $\hat{\mathbb{T}}$ that is obtained exhibits the structure described in (2.10) and is unitarily similar to a complex representation matrix that approximates $\mathbb{H}$, i.e.,

$$\mathbb{W}^H(\mathbb{H} + \mathbb{E})\mathbb{W} = \hat{\mathbb{T}}, \tag{3.7}$$

where $\mathbb{W}$ is a unitary matrix, $\mathbb{E}$ is a complex representation matrix with $\|\mathbb{E}\|_2 \approx \epsilon\|\mathbb{H}\|_2$. The calculated $\tilde{\mathbb{W}}$ is almost unitary complex representation matrix in the sense that $\tilde{\mathbb{W}}^H\tilde{\mathbb{W}} - I_{2n} = \mathbb{F}$ is a complex representation matrix and $\|\mathbb{F}\|_2 \approx \epsilon$.

Remark 1 A complex representation matrix is uniquely determined by its two submatrices on the first column block, and vice versa. The structure-preserving transformation on a complex representation matrix is equivalent to corresponding transformations on two submatrices on the first column block.

Then based on the above statement and the properties of the complex representation matrix of a quaternion matrix, we can develop the complex structure-preserving implicit double shift quaternion QR algorithm for Schur decomposition of a quaternion matrix, as outlined in Algorithm 3.

From Algorithm 3, it is clear that the right eigenvalues of the original quaternion matrix A can be derived from the diagonal elements of the quasi upper-triangular quaternion matrix T. The eigenvectors of A can be obtained by computing the eigenvectors of of T, and transforming them back using the unitary quaternion transformation W. Consequently, the task of finding the eigenvectors of A is simplified to computing the eigenvectors of T.

The subfunctions to be used are shown in Algorithms 4, 5 and 6 below.

Algorithm 3: Give an arbitrary quaternion matrix $A = \mathbb{A}_1 + \mathbb{A}_2 \mathrm{j} \in \mathbf{H}^{n\times n}$, where $\mathbb{A}_1, \mathbb{A}_2 \in \mathbf{C}^{n\times n}$. This algorithm provides a method to find a unitary quaternion matrix $W = \mathbb{W}_1 + \mathbb{W}_2 \mathrm{j}$ and a upper quasi-triangular matrix $T = \mathbb{T}_1 + \mathbb{T}_2 \mathrm{j}$ such that $T = WAW^H$, where $\mathbb{W}_1, \mathbb{W}_2, \mathbb{T}_1, \mathbb{T}_2 \in \mathbf{C}^{n\times n}$.
Input: $\mathbb{A}_1$, $\mathbb{A}_2$. Output: $\mathbb{W}_1$, $\mathbb{W}_2$ and $\mathbb{T}_1$, $\mathbb{T}_2$.

```
Function [W1, W2, T1, T2]=QCSchur(A1, A2)
    1. Initialization.
        A_c^C = [A1; −conj(A2)]; m =size(A_c^C, 2) ;
        P1 = [eye(m, m); zeros(m, m)]; tol = 1e − 11 ;
    2. Generalized Hessenberg reduction;
    [P, H] =qCHessenberg(A_c^C) ;
    3. Iterative process.
    while 1 do
        for s = 2 : m do
            if norm([H(s, s − 1); H(s + m, s − 1)]) <= tol ∗ (norm([H(s, s); H
            (s + m, s)]) + norm([H(s − 1, s − 1); H(s − 1 + m, s − 1)])) then
                H([s, s + m], s − 1) = zeros(2, 1);
            end
        end
        [p, q] = find_pq(H(1 : m, 1 : m)) ;
        if q==m then
            break;
        end
        [WF, N1] =cReduction(H([p + 1 : m − q, m + p + 1 : 2 ∗ m − q], p + 1 : m − q));
        H([p + 1 : m − q, m + p + 1 : 2 ∗ m − q], p + 1 : m − q) = N1;
        d = m − p − q ;
        WR = Q_Comp(WF(1 : d, 1 : d), −conj(WF(d + 1 : 2 ∗ d, 1 : d))) ;
        P1([p + 1 : m − q, m + p + 1 : 2 ∗ m − q], :) = WR ∗ P1([p + 1 : m − q, m + p + 1 :
          2 ∗ m − q], :);
        Z = [H(1 : p, p + 1 : m − q), −conj(H(m + 1 : m + p, p + 1 : m − q))] ∗ WR′;
        H([1 : p, m+1 : m+p], p+1 : m−q) = [Z(1 : p, 1 : d); −conj(Z(1 : p, d+1 : 2∗d))];
          H([p + 1 : m − q, m + p + 1 : 2 ∗ m − q], m − q + 1 : m) = WR ∗ H([p + 1 :
          m − q, m + p + 1 : 2 ∗ m − q], m − q + 1 : m);
    end
    4. Output results.
        Q = Q_Comp(P1(1 : m, 1 : m), −conj(P1(m + 1 : 2 ∗ m, 1 : m))) ∗ P ;
        W1 = Q(1 : m, :); W2 = −conj(Q(m + 1 : 2 ∗ m, :)) ;
        T1 = H(1 : m, :); T2 = −conj(H(m + 1 : 2 ∗ m, :)) ;
End
```

4 Numerical experiments

Example 4.1 For the quaternion matrix

$$A = \begin{bmatrix} 0.8621 & 0.8967 & 0.5524 & 0.1832 & 0.4597 \\ 0.1619 & 0.2005 & 0.6417 & 0.0069 & 0.3559 \\ 0.9732 & 0.2579 & 0.9828 & 0.6645 & 0.5541 \\ 0.7322 & 0.7375 & 0.8317 & 0.5272 & 0.2285 \\ 0.6764 & 0.9324 & 0.0787 & 0.3904 & 0.4319 \end{bmatrix} + \begin{bmatrix} 0.4836 & 0.4318 & 0.8066 & 0.0253 & 0.8028 \\ 0.1207 & 0.3015 & 0.2131 & 0.0910 & 0.0497 \\ 0.6370 & 0.6584 & 0.8935 & 0.1505 & 0.5464 \\ 0.9317 & 0.6497 & 0.2957 & 0.8747 & 0.4829 \\ 0.0415 & 0.9517 & 0.3039 & 0.8246 & 0.4917 \end{bmatrix} \mathrm{i}$$

$$+ \begin{bmatrix} 0.4294 & 0.5004 & 0.0085 & 0.5511 & 0.1035 \\ 0.9474 & 0.3589 & 0.0471 & 0.3049 & 0.9541 \\ 0.1531 & 0.7213 & 0.3535 & 0.3583 & 0.9990 \\ 0.3783 & 0.5637 & 0.2810 & 0.9391 & 0.0561 \\ 0.4302 & 0.2802 & 0.5650 & 0.3715 & 0.4380 \end{bmatrix} \mathrm{j} + \begin{bmatrix} 0.0765 & 0.6257 & 0.8342 & 0.1879 & 0.1084 \\ 0.9629 & 0.0520 & 0.3362 & 0.0943 & 0.7749 \\ 0.9860 & 0.6856 & 0.1057 & 0.9640 & 0.0625 \\ 0.7642 & 0.8900 & 0.6393 & 0.5275 & 0.9438 \\ 0.5208 & 0.3602 & 0.4540 & 0.6674 & 0.2406 \end{bmatrix} \mathrm{k},$$

compute the Schur decomposition of the A.

Algorithm 4: Give an arbitrary quaternion matrix $A \in \mathbf{H}^{n\times n}$. This algorithm provides a method to find a unitary quaternion matrix $P \in \mathbf{H}^{n\times n}$ and a upper Hessenberg quaternion matrix $H \in \mathbf{H}^{n\times n}$ such that $H = PAP^H$.
Input: the first column block A_c^C of A^C. Output: the first column block P_c^C of P^C and the first column block H_c^C of H^C.

Function $[P_c^C, H_c^C]$=**qCHessenberg(**A_c^C**)**
1. Initialization.
   $B = A_c^C$; $n =$size$(B, 2)$;
   $D = [\text{eye}(n, n); \text{zeros}(n, n)]$;
2. Cyclic process;
for $t = 1 : n - 2$ **do**
   $s = t + 1$;
   if $\text{norm}([B(s : n, t); B((n + s) : (2 * n), t)]) > 0$ **then**
      $[u, \beta] = \text{QC\_House}(B(s : n, t), -\text{conj}(B(n + s : 2 * n, t)), n - s + 1)$;
      $B([s : n, s + n : 2 * n], t : n) = B([s : n, s + n : 2 * n], t : n) - (\beta * u)*$ $(u' * B([s : n, s + n : 2 * n], t : n))$;
      $D([s : n, s + n : 2 * n], 1 : n) = D([s : n, s + n : 2 * n], 1 : n) - (\beta * u)*$ $(u' * D([s : n, s + n : 2 * n], 1 : n))$;
   end
   $Z(1 : n, [s : n, s + n : 2 * n]) = [B(1 : n, s : n), -\text{conj}(B(1 + n : 2 * n, s : n))]$;
   $Z(1 : n, [s : n, s + n : 2 * n]) = Z(1 : n, [s : n, s + n : 2 * n]) - Z(1 : n, [s : n, s + n : 2 * n]) * u * (\beta * u')$;
   $B([1 : n, 1 + n : 2 * n], s : n) = [Z(1 : n, s : n); -\text{conj}(Z(1 : n, s + n : 2 * n))]$;
end
3. Output results.
   $P_c^C = D$; $H_c^C = B$;
End

It is easy to use Algorithm 3 directly to obtain $A = W^H T W$, where

$$W = \begin{bmatrix} -0.3921 - 0.0024\mathrm{i} & -0.2783 - 0.0766\mathrm{i} & -0.5304 - 0.0346\mathrm{i} & -0.5300 + 0.0476\mathrm{i} & -0.4031 + 0.0187\mathrm{i} \\ -0.2667 + 0.1236\mathrm{i} & 0.4369 - 0.1003\mathrm{i} & 0.0209 - 0.3437\mathrm{i} & 0.0306 + 0.2485\mathrm{i} & -0.0287 + 0.1186\mathrm{i} \\ -0.0952 + 0.1853\mathrm{i} & -0.5139 - 0.1436\mathrm{i} & 0.3182 - 0.1253\mathrm{i} & -0.1520 + 0.1378\mathrm{i} & 0.3772 - 0.0901\mathrm{i} \\ 0.2735 + 0.0554\mathrm{i} & -0.0071 - 0.3814\mathrm{i} & 0.1364 + 0.2204\mathrm{i} & 0.1609 + 0.0016\mathrm{i} & -0.5333 - 0.0529\mathrm{i} \\ 0.1947 - 0.4635\mathrm{i} & 0.2291 + 0.0103\mathrm{i} & 0.0362 + 0.3825\mathrm{i} & -0.2566 + 0.1769\mathrm{i} & 0.0461 - 0.1416\mathrm{i} \end{bmatrix}$$

$$+ \begin{bmatrix} -0.0138 - 0.0123\mathrm{i} & -0.0011 + 0.1496\mathrm{i} & 0.0015 + 0.0236\mathrm{i} & -0.0349 + 0.0806\mathrm{i} & 0.0009 + 0.0582\mathrm{i} \\ 0.0079 - 0.3681\mathrm{i} & 0.3245 - 0.0283\mathrm{i} & 0.2469 + 0.2842\mathrm{i} & -0.2395 - 0.0770\mathrm{i} & -0.2078 - 0.1631\mathrm{i} \\ 0.2419 - 0.1186\mathrm{i} & 0.0207 + 0.1658\mathrm{i} & 0.1085 - 0.0155\mathrm{i} & -0.2998 - 0.1719\mathrm{i} & -0.0367 + 0.3593\mathrm{i} \\ 0.1024 + 0.0896\mathrm{i} & 0.1171 + 0.1816\mathrm{i} & 0.0591 - 0.0034\mathrm{i} & -0.4692 - 0.0957\mathrm{i} & 0.2959 - 0.1036\mathrm{i} \\ 0.0397 - 0.3965\mathrm{i} & -0.1425 + 0.0974\mathrm{i} & 0.2756 - 0.1827\mathrm{i} & -0.1066 + 0.2405\mathrm{i} & -0.1593 + 0.1877\mathrm{i} \end{bmatrix} \mathrm{j},$$

and

$$T = \begin{bmatrix} 2.6657 + 2.9423\mathrm{i} & 0.3434 - 0.8901\mathrm{i} & 0.2048 + 0.2734\mathrm{i} & -0.1514 - 0.1726\mathrm{i} & -0.0536 - 0.4066\mathrm{i} \\ -0.0000 - 0.0000\mathrm{i} & -0.1391 + 0.2405\mathrm{i} & 0.2938 - 0.3686\mathrm{i} & -0.0747 - 0.1565\mathrm{i} & -0.2591 - 0.2836\mathrm{i} \\ 0.0000 + 0.0000\mathrm{i} & 0.0000 + 0.0000\mathrm{i} & -0.7233 - 0.6771\mathrm{i} & -0.0088 - 0.2895\mathrm{i} & 0.3881 - 0.3714\mathrm{i} \\ -0.0000 + 0.0000\mathrm{i} & 0.0000 - 0.0000\mathrm{i} & 0.0000 + 0.0000\mathrm{i} & 0.7659 + 0.2399\mathrm{i} & -0.2968 - 0.2463\mathrm{i} \\ -0.0000 + 0.0000\mathrm{i} & -0.0000 - 0.0000\mathrm{i} & 0.0000 + 0.0000\mathrm{i} & 0.0000 + 0.0000\mathrm{i} & 0.4351 - 0.3476\mathrm{i} \end{bmatrix}$$

$$+ \begin{bmatrix} 1.2817 + 2.4709\mathrm{i} & 0.7231 - 0.5986\mathrm{i} & 0.1772 + 0.6150\mathrm{i} & 0.0172 + 0.5724\mathrm{i} & -0.3119 - 0.7626\mathrm{i} \\ -0.0000 - 0.0000\mathrm{i} & -1.1064 - 0.6983\mathrm{i} & -0.2718 + 0.2405\mathrm{i} & -0.0094 + 0.5388\mathrm{i} & -0.3022 - 0.2913\mathrm{i} \\ -0.0000 + 0.0000\mathrm{i} & -0.0000 + 0.0000\mathrm{i} & -0.1704 - 0.6244\mathrm{i} & 0.3088 + 0.5204\mathrm{i} & 0.0510 - 0.0732\mathrm{i} \\ -0.0000 - 0.0000\mathrm{i} & -0.0000 + 0.0000\mathrm{i} & -0.0000 + 0.0000\mathrm{i} & 0.0936 + 0.0316\mathrm{i} & -0.3419 - 0.4700\mathrm{i} \\ -0.0000 - 0.0000\mathrm{i} & 0.0000 - 0.0000\mathrm{i} & -0.0000 + 0.0000\mathrm{i} & -0.0000 + 0.0000\mathrm{i} & -0.2314 - 0.0230\mathrm{i} \end{bmatrix} \mathrm{j}.$$

and the corresponding computational error is

$$\|AW^H - W^H T\|_F \approx 9.0751 \times 10^{-15}.$$

Algorithm 5: Give an arbitrary complex Hessenberg matrix $H \in \mathbf{C}^{n\times n}$. This algorithm provides a method to find the maximum nonnegative integer q and the minimum nonnegative integer p of H, where H has the form of (3.5). Input: H . Output: p and q.

```
Function [p, q]=find_pq(H)
    1. Initialization.
        n = size(H, 1); q = 0; s = n ;
    2. Cyclic process.
    while 1 do
        if s == 2 then
            q = q + 2;
            break;
        end
        if s == 1 then
            q = q + 1 ;
            break;
        end
        if H(s, s − 1) == 0 then
            q = q + 1 ;
            s = s − 1;
            continue;
        end
        if H(s − 1, s − 2) == 0 then
            q = q + 2;
            s = s − 2;
            continue;
        end
        break;
    end
    if q == n then
        p = 0;
        return;
    end
        temp=1;
    for t = n − q : −1 : 2 do
        if H(t, t − 1) == 0 then
            break;
        else
            temp=temp+1;
        end
    end
        p = n−temp−q;
    3. Output results.
        p;  q;
End
```

Example 4.2 For the quaternion matrix $A = A_1 + A_2\mathrm{i} + A_3\mathrm{j} + A_4\mathrm{k} \in \mathbf{H}^{m\times n}$ created by the function rand in Matlab, where $A_t = \mathrm{rand}(m, n), t = 1, 2, 3, 4$, when $n = 40 : 40 : 800$, perform the Schur decomposition of A by proposed Algorithm 3 and the real structure-preserving QR algorithm [22], respectively. The CPU times and computational errors by the two algorithms are shown in Fig. 1, where C_QR_Algorithm denotes proposed Algorithm 3 and R_QR_Algorithm denotes the real structure-preserving algorithm, and the computational error used is $\|AW^H - W^H T\|_F$. All these computations are performed on an Intel i7-13700K @ 3.40 GHz/32 GB computer using Matlab 2018b.

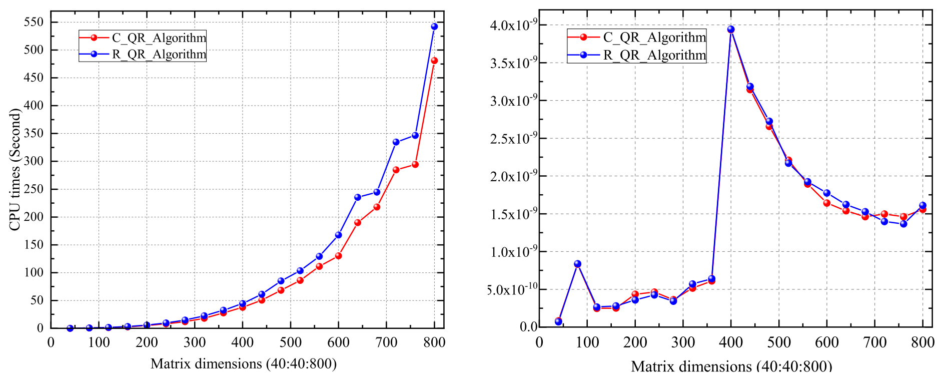

Fig.1 Time required and computational error of performing Schur decomposition for A.

It is clear from Figure 1 that our proposed new algorithm is clearly faster than the algorithm of the paper [22] in terms of computational speed, and in particular, our algorithm takes about 50 seconds less than that of the algorithm of [22] when $n = 800$. Moreover, the computational errors of the two algorithms remain almost the same, thus also proving the effectiveness of Algorithm 3.

5 Conclusions

In this paper, a complex structure-preserving QR double implicit shift iterative algorithm is proposed for Schur decomposition of quaternion matrices via the complex representation matrix of quaternion matrices, and also this is a standard method for computing the eigenvalue problem of dense quaternion matrices, which takes into account the connection between quaternion matrices and their complex representation matrices. Numerical experimental results also verify the effectiveness of the algorithm proposed in this paper. The study in this paper not only provides a new method for Schur decomposition of quaternion matrices, but also provides an important tool for solving quaternion problems in practical applications.

References

[1] A.W. Bojanczyk, G.H. Golub, P. Van Dooren, *Periodic Schur decomposition: algorithms and applications*, In Advanced Signal Processing Algorithms, Architectures, and Implementations III, **1770** (1992), 31–42.

[2] R. Mathias, *Condition estimation for matrix functions via the Schur decomposition*, SIAM journal on matrix analysis and applications, **16**:2 (2017), 565–578.

[3] S. Timme, S. Marques, K.J. Badcock, *Transonic aeroelastic stability analysis using a Kriging-based Schur complement formulation*, AIAA journal, **49**:6 (2011), 1202–1213.

[4] M. Haardt, J.A. Nossek, *Simultaneous Schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems*, IEEE Transactions on Signal Processing, **46**:1 (1998), 161–169.

[5] S. Kaynama, M. Oishi, *Schur-based decomposition for reachability analysis of linear time-invariant systems*, In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, (2009), 69–74.

[6] T. Jiang, Z. Guo, D. Zhang, V.I. Vasil'ev, *A fast algorithm for the Schrödinger equation in quaternionic quantum mechanics*, Applied Mathematics Letters, **150** (2024), 108975.

[7] Z. Guo, T. Jiang, V.I. Vasil'ev, G. Wang, *A novel algebraic approach for the Schrödinger equation in split quaternionic mechanics*, Applied Mathematics Letters, **137** (2023), 108485.

[8] T. Jiang, G. Wang, Z. Guo, D. Zhang, *Algebraic algorithms for a class of Schrödinger equations in split quaternionic mechanics*, Mathematical Methods in the Applied Sciences, **47**:7 (2024), 6205–6215.

[9] Z. Guo, T. Jiang, G. Wang, V.I. Vasil'ev, *Algebraic algorithms for eigen-problems of a reduced biquaternion matrix and applications*, Applied Mathematics and Computation, **463** (2024), 128358.

[10] S.L. Alder, *Quaternionic quantum field theory*, Communications in Mathematical Physics, **104** (1986), 611–656.

[11] S. Miron, J. Flamant, N. Le Bihan, P. Chainais, D. Brie, *Quaternions in Signal and Image Processing: A comprehensive and objective overview*, IEEE Signal Processing Magazine, **40**:6 (2023), 26–40.

[12] N. Le Bihan, J. Mars, *Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing*, Signal processing, **84**:7 (2004), 1177–1199.

[13] W.R. Hamilton, *On a new species of imaginary quantities, connected with the theory of quaternions*, Proceedings of the Royal Irish Academy, **2** (1840), 424–434.

[14] A. Alaimo, V. Artale, C. Milazzo, A. Ricciardello, *Comparison between Euler and quaternion parametrization in UAV dynamics*, In AIP Conference Proceedings, **1558**:1 (2013), 1228–1231.

[15] J. Vince, *Quaternions for computer graphics*, Springer, London, 2011.

[16] H. Liu, X. Wang, Y. Zhong, *Quaternion-based robust attitude control for uncertain robotic quadrotors*, IEEE Transactions on Industrial Informatics, **11**:2 (2015), 406–415.

[17] D. Lee, G. Vukovich, R. Lee, *Robust adaptive unscented Kalman filter for spacecraft attitude estimation using quaternion measurements*, Journal of Aerospace Engineering, **30**:4 (2017), 04017009.

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

[19] F. Zhang, *Quaternions and matrices of quaternions*, Linear algebra and its applications, **251** (1997), 21-57.

[20] T. Jiang, *Algebraic methods for diagonalization of a quaternion matrix in quaternionic quantum theory*, Journal of Mathematical Physics, **46**:5 (2005).

[21] S. De Leo, G. Scolarici, *Right eigenvalue equation in quaternionic quantum mechanics*, Journal of Physics A: Mathematical and General, **33**:15 (2000), 2971.

[22] Z. Jia, M. Wei, M. Zhao, Y. Chen, *A new real structure-preserving quaternion QR algorithm*, Journal of Computational and Applied Mathematics, **343** (2018), 26–48.

[23] Z. Guo, T. Jiang, V.I. Vasil'ev, G. Wang, *Complex structure-preserving method for Schrödinger equations in quaternionic quantum mechanics*, Numerical Algorithms, **97** (2024), 271–287.

[24] D. Zhang, T. Jiang, C. Jiang, G. Wang, *A complex structure-preserving algorithm for computing the singular value decomposition of a quaternion matrix and its applications*, Numerical Algorithms, **95**:1 (2024), 267–283.

[25] Y. Li, M. Wei, F. Zhang, J. Zhao, *A real structure-preserving method for the quaternion LU decomposition, revisited*, Calcolo, **54** (2017), 1553–1563.
[26] Y. Chen, Z. Jia, Y. Peng, Y. Peng, D. Zhang, *A new structure-preserving quaternion QR decomposition method for color image blind watermarking*, Signal Processing, **185** (2021), 108088.
[27] T. Jiang, L. Chen, *An algebraic method for Schrödinger equations in quaternionic quantum mechanics*, Computer Physics Communications, **178**:11 (2008), 795–799.
[28] G. Wang, D. Zhang, V.I. Vasiliev, T. Jiang, *A complex structure-preserving algorithm for the full rank decomposition of quaternion matrices and its applications*, Numerical Algorithms, **91**:4 (2022), 1461–1481.

Zhenwei Guo
Institute of Mathematics and Information Science,
North-Eastern Federal University,
str. Belinsky, 58,
677000, Yakutsk, Russia
Email address: guozhenweilcu@163.com

Algorithm 6: Give an unreduced upper Hessenberg quaternion matrix $A \in \mathbf{H}^{n\times n}$. This algorithm gives a unitary transformation by performing Francis QR iterations for A such that $N = WAW^H$, where $N \in \mathrm{H}^{n\times n}$ is a new reduced upper Hessenberg quaternion matrix.
Input: the first column block A_c^C of A^C. Output: the first column block W_c^C of W^C, the first column block N_c^C of N^C.

```
Function [W_c^C, N_c^C] = cReduction(A_c^C)
  1. Initialization.
  H = A_c^C; n = size(H,2); m = n − 1; W_c^C = [eye(n); zeros(n)] ;
  σ = min(eig(qComplexp(H(m : n, m : n), −conj(H(m + n : 2 ∗ n, m : n))))) ;
  s = 2 ∗ real(σ); t = (norm(σ))^2 ;
  HR = qComplexp(H(1 : n, :), −conj(H(n + 1 : 2 ∗ n, :))) ;
  F = HR([1 : 3, n + 1 : n + 3], :) ∗ HR(:, 1) − s ∗ HR([1 : 3, n + 1 :
    n + 3], 1) + t ∗ [1; zeros(5, 1)];
  2. Cyclic process.
  for k = 1 : n − 2 do
    [u, β] = QC_House(F(1 : 3), −conj(F(4 : 6)), 3);
    q = max(1, k − 1);
    H([k : k + 2, n + k : n + k + 2], q : n) = H([k : k + 2, n + k : n + k + 2], q :
      n) − β ∗ u ∗ (u′ ∗ H([k : k + 2, n + k : n + k + 2], q : n));
    W_c^C([k : k + 2, n + k : n + k + 2], 1 : n) = W_c^C([k : k + 2, n + k : n + k + 2], 1 :
      n) − β ∗ u ∗ (u′ ∗ W_c^C([k : k + 2, n + k : n + k + 2], 1 : n));
    r = min(k + 3, n);
    Z(1 : r, [k : k + 2, n + k : n + k + 2]) = [H(1 : r, k : k + 2), −conj(H(1 + n :
      r + n, k : k + 2))];
    Z(1 : r, [k : k + 2, n + k : n + k + 2]) = Z(1 : r, [k : k + 2, n + k :
      n + k + 2]) − Z(1 : r, [k : k + 2, n + k : n + k + 2]) ∗ u ∗ (β ∗ u′);
    H([1 : r, n + 1 : n + r], k : k + 2) = [Z(1 : r, k : k + 2); −conj(Z(1 : r, n + k :
      n + k + 2))];
    if k < n − 2 then
      F = H([k + 1 : k + 3, n + k + 1 : n + k + 3], k);
    end
  end
  [u1, β1] = QC_House(H(n − 1 : n, n − 2), −conj(H(2 ∗ n − 1 : 2 ∗ n, n − 2)), 2) ;
  H([n − 1 : n, 2 ∗ n − 1 : 2 ∗ n], 1 : n) = H([n − 1 : n, 2 ∗ n − 1 : 2 ∗ n], 1 :
    n) − β1 ∗ u1 ∗ (u1′ ∗ H([n − 1 : n, 2 ∗ n − 1 : 2 ∗ n], 1 : n)) ;
  ZZ(1 : n, [n − 1 : n, 2 ∗ n − 1 : 2 ∗ n]) = [H(1 : n, n − 1 : n), −conj(H(n + 1 :
    2 ∗ n, n − 1 : n))];
  ZZ(1 : n, [n − 1 : n, 2 ∗ n − 1 : 2 ∗ n]) = ZZ(1 : n, [n − 1 : n, 2 ∗ n − 1 : 2 ∗ n]) − ZZ(1 :
    n, [n − 1 : n, 2 ∗ n − 1 : 2 ∗ n]) ∗ u1 ∗ (β1 ∗ u1′) ;
  H([1 : n, n+1 : 2∗n], n−1 : n) = [ZZ(1 : n, n−1 : n); −conj(ZZ(1 : n, 2∗n−1 : 2∗n))];
  W_c^C([n − 1 : n, 2 ∗ n − 1 : 2 ∗ n], 1 : n) = W_c^C([n − 1 : n, 2 ∗ n − 1 : 2 ∗ n], 1 :
    n) − β1 ∗ u1 ∗ (u1′ ∗ W_c^C([n − 1 : n, 2 ∗ n − 1 : 2 ∗ n], 1 : n));
  [u2, β2] = QC_House(H(n, n − 1), −conj(H(2 ∗ n, n − 1)), 1);
  H([n, 2 ∗ n], 1 : n) = H([n, 2 ∗ n], 1 : n) − β2 ∗ u2 ∗ (u2′ ∗ H([n, 2 ∗ n], 1 : n));
  ZZZ(1 : n, [n, 2 ∗ n]) = [H(1 : n, n), −conj(H(n + 1 : 2 ∗ n, n))];
  ZZZ(1 : n, [n, 2 ∗ n]) = ZZZ(1 : n, [n, 2 ∗ n]) − ZZZ(1 : n, [n, 2 ∗ n]) ∗ u2 ∗ (β2 ∗ u2′);
  H([1 : n, n + 1 : 2 ∗ n], n) = [ZZZ(1 : n, n); −conj(ZZZ(1 : n, 2 ∗ n))];
  W_c^C([n, 2 ∗ n], 1 : n) = W_c^C([n, 2 ∗ n], 1 : n) − β2 ∗ u2 ∗ (u2′ ∗ W_c^C([n, 2 ∗ n], 1 : n));
  3. Output results.
    W_c^C;   N_c^C = H;
End
```