Structure Preserving Quaternion Generalized Minimal Residual Method *

Zhigang Jia† and Michael K. Ng‡

Abstract. The main aim of this paper is to develop the quaternion generalized minimal residual method (QGMRES) for solving quaternion linear systems. Quaternion linear systems arise from three-dimensional or color imaging filtering problems. The proposed quaternion Arnoldi procedure can preserve quaternion Hessenberg form during the iterations. The main advantage is that the storage of the proposed iterative method can be reduced by comparing with the Hessenberg form constructed by the classical GMRES iterations for the real representation of quaternion linear systems. The convergence of the proposed QGMRES is also established. Numerical examples are presented to demonstrate the effectiveness of the proposed QGMRES compared with the traditional GMRES in terms of storage and computing time.

Key words. General quaternion linear systems; Quaternion Krylov subspace; Quaternion Arnoldi method; Quaternion generalized minimal residual method; Three-dimensional signal filtering.

AMS subject classifications. 65F10, 97N30, 94A08

1. Introduction.

Let $\mathbb{Q}$ denote the quaternion skew-field and $\mathbb{Q}^{m\times n}$ the set of $m \times n$ matrices on $\mathbb{Q}$. A quaternion is expressed [1] as $\mathbf{a} = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}$ with $a_0, a_1, a_2, a_3 \in \mathbb{R}$, and three imaginary units $\mathbf{i}, \mathbf{j}, \mathbf{k}$ satisfying $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. Let $\mathbb{Q}$ denote the quaternion skew-field and $\mathbb{Q}^{m\times n}$ the set of $m \times n$ matrices on $\mathbb{Q}$. The quaternion linear systems are of the form

$$\boldsymbol{\mathcal{A}}\mathbf{x} = \mathbf{b}, \tag{1.1}$$

where $\boldsymbol{\mathcal{A}} \in \mathbb{Q}^{n\times n}$ is an invertible $n\times n$ quaternion matrix, $\mathbf{b} \in \mathbb{Q}^n$ is an n-dimensional quaternion vector, and $\mathbf{x} \in \mathbb{Q}^n$ is an unknown quaternion vector. Such linear systems arise from many scientific applications such as the decoding process of a quaternion convolutional neural network [11, 13], color image denoising [6, 8], and three-dimensional signal processing [19]. The main aim of this paper is to develop iterative methods for solving large-scale quaternion linear systems.

Existing iterative methods of solving quaternion linear systems are based on the real or complex representation, whose dimension is expanded to four or two times of the original dimension. For instance, the quaternion linear systems (1.1) can be

*This paper is supported in part by National Natural Science Foundation of China under grants 12090011 and 11771188; HKRGC Grants GRF 12200317, 12300218, 12300519, and 17201020; the Priority Academic Program Development Project (PAPD); the Top-notch Academic Programs Project PPZY2015A013 of Jiangsu Higher Education Institutions..

†1. School of Mathematics and Statistics and Research Institute of Mathematical Science, Jiangsu Normal University, Xuzhou 221116, People's Republic China. Email: zhgjia@jsnu.edu.cn

‡Department of Mathematics, Hong Kong University, Hong Kong. E-mail: mng@maths.hku.hk

equivalently rewritten to a real matrix equation

$$\mathscr{R}(\boldsymbol{\mathcal{A}})\mathscr{R}(\mathbf{x}) = \mathscr{R}(\mathbf{b}), \tag{1.2}$$

where $\mathscr{R}(\cdot)$ is a linear homeomorphic mapping from quaternion matrices (or vectors) to their real counterpart. Even though there are many different real counterparts, it is interesting to note that they are permutation equivalent; see [5, Remark 4.7]. In the following discussion, we consider the following mapping: for any $\boldsymbol{\mathcal{Q}} = Q_0 + Q_1\mathbf{i} + Q_2\mathbf{j} + Q_3\mathbf{k} \in \mathbb{Q}^{m\times n}$,

$$\mathscr{R}(\boldsymbol{\mathcal{Q}}) = \begin{bmatrix} Q_0 & -Q_1 & -Q_2 & -Q_3 \\ Q_1 & Q_0 & -Q_3 & Q_2 \\ Q_2 & Q_3 & Q_0 & -Q_1 \\ Q_3 & -Q_2 & Q_1 & Q_0 \end{bmatrix} \in \mathbb{R}^{4m\times 4n}. \tag{1.3}$$

To solve the general (real) linear systems, iterative methods can be employed, for example, BCG [10], CGS [14], BiCGSTAB [20], FOM [16], or GMRES [17]. Especially, GMRES is commonly used in the literature. The main idea of GMRES is to approximate the solution by the Krylov subspace with minimal residual. The Arnoldi iteration is used to find this subspace. When GMRES is applied to solving resulting real linear systems arising from quaternion linear systems, a real upper Hessenberg matrix appears in GRMES iterations. However, the upper Hessenberg structure cannot be preserved in quaternion representation; see Figure 1(c) in our numerical example for a demonstration. It is clear in this setting that the storage size would be increased in GMRES iterations for solving such real equivalent linear systems arising from quaternion linear systems.

The main aim of this paper is to develop a structure preserving quaternion GMRES which can inherit the algebraic symmetry of $\mathscr{R}(\boldsymbol{\mathcal{A}})$ for solving (1.1). The main advantage of the proposed method is to save computational operations and storage. This paper is organized as follows. In Section 2, we review quaternion matrices. In Section 3, we the present structure preserving quaternion Krylov subspace method and the quaternion Arnoldi method. The modified quaternion Arnoldi method is also proposed. In Section 4, we present the quaternion generalized minimal residual method (QGMRES) for solving quaternion linear systems and its convergence. In Section 5, numerical examples for three-dimensional signal filtering problems are presented to illustrate the convergence of the proposed structure preserving QGMRES method and to demonstrate that its performance is better than that of the traditional GMRES method. In Section 6, we present concluding remarks.

2. Preliminaries. In this section, we introduce quaternion matrices and the (right) Hilbert space of quaternion vectors. Notice that $\mathbb{Q}$ is an associative but multiplicatively noncommutative algebra of rank four over $\mathbb{R}$, endowed with an involutory antiautomorphism $\overline{\mathbf{q}} = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}$. The quaternion norm $|\mathbf{q}|$ is defined by $|\mathbf{q}|^2 = \overline{\mathbf{q}}\mathbf{q} = q_0^2 + q_1^2 + q_2^2 + q_3^2$. Every nonzero quaternion is invertible, and the unique inverse is given by $1/\mathbf{q} = \overline{\mathbf{q}}/|\mathbf{q}|^2$. The conjugate transpose of $\boldsymbol{\mathcal{Q}}$ is defined as $\boldsymbol{\mathcal{Q}}^* = Q_0^T - Q_1^T\mathbf{i} - Q_2^T\mathbf{j} - Q_3^T\mathbf{k}$. A quaternion matrix $\boldsymbol{\mathcal{Q}}$ is full of column rank if and only if $\boldsymbol{\mathcal{Q}}\mathbf{x} = \mathbf{0}$ has a unique solution $\mathbf{x} = \mathbf{0}$, and moreover, the columns of $\boldsymbol{\mathcal{Q}}$ are orthogonal to each other if $\boldsymbol{\mathcal{Q}}^*\boldsymbol{\mathcal{Q}} = \mathbf{I}$. We refer to [12, 21, 22] for the theory of quaternion matrices.

The right-hand- (or left-hand-) side linear combination of quaternion vectors $\mathbf{v}_1$, $\mathbf{v}_2, \cdots, \mathbf{v}_m$ is denoted by $\mathbf{v}_1\boldsymbol{\alpha}_1 + \mathbf{v}_2\boldsymbol{\alpha}_1 + \cdots + \mathbf{v}_m\boldsymbol{\alpha}_m$ (or $\boldsymbol{\alpha}_1\mathbf{v}_1 + \boldsymbol{\alpha}_1\mathbf{v}_2 + \cdots + \boldsymbol{\alpha}_m\mathbf{v}_m$), where $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_m$ are quaternion scalars. Here and after, we concentrate on the right-hand-side linear combination of quaternion vectors, since it has similar properties to the linear combination of real or complex vectors. Quaternion vectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m$ are linearly independent if $\mathbf{v}_1\boldsymbol{\alpha}_1 + \mathbf{v}_2\boldsymbol{\alpha}_1 + \cdots + \mathbf{v}_m\boldsymbol{\alpha}_m = 0$ holds if and only if $\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2 = \cdots = \boldsymbol{\alpha}_m = 0$; otherwise, they are linearly dependent.

Definition 2.1 (inner product). *The inner product of two quaternion vectors,* $\mathbf{w} = [\mathbf{w}_i],\ \mathbf{v} = [\mathbf{v}_i] \in \mathbb{Q}^n$, *is defined as*

$$\langle \mathbf{w}, \mathbf{v} \rangle := \sum_{i=1}^{n} \mathbf{v}_i^* \mathbf{w}_i. \tag{2.1}$$

The inner product of two quaternion matrices, $\boldsymbol{\mathcal{B}} = [\mathbf{b}_{ij}],\ \boldsymbol{\mathcal{C}} = [\mathbf{c}_{ij}] \in \mathbb{Q}^{m\times n}$, *is defined as*

$$\langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}} \rangle := \mathbf{Trace}(\boldsymbol{\mathcal{C}}^*\boldsymbol{\mathcal{B}}) = \sum_{j=1}^{n}\sum_{i=1}^{m} \mathbf{c}_{ji}^* \mathbf{b}_{ij}. \tag{2.2}$$

Definition 2.2 (norm). *The* p*-norm of the quaternion vector,* $\mathbf{v} = [\mathbf{v}_i] \in \mathbb{Q}^n$, *is defined as*

$$\|\mathbf{v}\|_p = \big(\sum_{i=1}^{n} |\mathbf{v}_i|^p\big)^{\frac{1}{p}},\ p \geq 1.$$

The p*-norm and* F*-norm of the quaternion matrix,* $\boldsymbol{\mathcal{Q}} = [\mathbf{q}_{ij}] \in \mathbb{Q}^{m\times n}$, *are respectively defined as*

$$\|\boldsymbol{\mathcal{Q}}\|_p = \max_{\mathbf{x}\in\mathbb{Q}^n/\{0\}} \frac{\|\boldsymbol{\mathcal{Q}}\mathbf{x}\|_p}{\|\mathbf{x}\|_p},\ p \geq 1,\ \|\boldsymbol{\mathcal{Q}}\|_F = \big(\sum_{j=1}^{n}\sum_{i=1}^{m} |\mathbf{q}_{ij}|^2\big)^{\frac{1}{2}}.$$

Let $\mathscr{S}$ stand for a linear vector space over $\mathbb{H}$ under right scalar multiplication. From [3], $\mathscr{S}$ is called a right quaternion Hilbert space if there exists a quaternionic scalar product, that is, a map $\mathscr{S} \times \mathscr{S} \ni (\mathbf{u}, \mathbf{v}) \mapsto \langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{H}$ satisfying the following four properties:

- (Right linearity) $\langle \mathbf{v}\boldsymbol{\alpha} + \mathbf{w}\boldsymbol{\beta}, \mathbf{u} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle\boldsymbol{\alpha} + \langle \mathbf{w}, \mathbf{u} \rangle\boldsymbol{\beta}$ if $\boldsymbol{\alpha},\ \boldsymbol{\beta} \in \mathbb{H}$, and $\mathbf{u},\ \mathbf{v},\ \mathbf{w} \in \mathscr{S}$.
- (Quaternionic hermiticity) $\langle \mathbf{v}, \mathbf{u} \rangle = \overline{\langle \mathbf{u}, \mathbf{v} \rangle}$ if $\mathbf{u},\ \mathbf{v} \in \mathscr{S}$.
- (Positivity) If $\mathbf{u} \in \mathscr{S}$, then $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$ and $\mathbf{u} = 0$ if $\langle \mathbf{u}, \mathbf{u} \rangle = 0$.
- $\mathscr{S}$ is complete with respect to its natural distance, defined by

$$d(\mathbf{v}, \mathbf{u}) = \sqrt{\langle \mathbf{v} - \mathbf{u}, \mathbf{v} - \mathbf{u} \rangle}$$

for any $\mathbf{u},\ \mathbf{v} \in \mathscr{S}$.

Clearly, the n-dimensional quaternion vector space $\mathbb{Q}^n$ is a right quaternionic Hilbert space with the inner product defined as (2.1). All quaternion vectors in

$$\mathscr{K} = \{\mathbf{v}_1\boldsymbol{\alpha}_1 + \mathbf{v}_2\boldsymbol{\alpha}_1 + \cdots + \mathbf{v}_m\boldsymbol{\alpha}_m \mid \mathbf{v}_j \in \mathbb{Q}^n,\ \boldsymbol{\alpha}_j \in \mathbb{Q},\ j = 1, \cdots, m\}$$

generate a subspace of $\mathbb{Q}^n$ of dimension $\mathbf{rank}([\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m])$.

3. Quaternion Arnoldi Method.

In this section, we define the quaternion Krylov subspace and present the quaternion Arnoldi method with structure preserving. We concentrate on the new theoretical results which are different from traditional ones. We solve (1.2) via the real number calculation but it is only necessary to store or overwrite four components of quaternion numbers. Theoretically, the JRS-symmetry (defined in [4, 5]) of the real counterpart is inherited. Let $\boldsymbol{\mathcal{A}} = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{Q}^{n\times n}$ with $A_0, A_1, A_2, A_3 \in \mathbb{R}^{n\times n}$ in (1.2). The real counterpart of $\boldsymbol{\mathcal{A}}$ is defined by the linear homeomorphic mapping $\mathscr{R}$,

$$\mathscr{R}(\boldsymbol{\mathcal{A}}) = \begin{bmatrix} A_0 & -A_1 & -A_2 & -A_3 \\ A_1 & A_0 & -A_3 & A_2 \\ A_2 & A_3 & A_0 & -A_1 \\ A_3 & -A_2 & A_1 & A_0 \end{bmatrix} \in \mathbb{R}^{4n\times 4n}. \tag{3.1}$$

Clearly, $\mathscr{R}(\boldsymbol{\mathcal{A}})$ is a JRS-symmetric matrix as defined in [4, 5]. Any real JRS-symmetric matrix is surely a real counterpart of a quaternion matrix. We define the inverse mapping of $\mathscr{R}$ on the real JRS-symmetric matrices by $\mathscr{R}^{-1}(\mathscr{R}(\boldsymbol{\mathcal{A}})) = \boldsymbol{\mathcal{A}}$.

Lemma 3.1. *Suppose $\boldsymbol{\mathcal{A}} = Q_0 + Q_1\mathbf{i} + Q_2\mathbf{j} + Q_3\mathbf{k}$ is an n-order square quaternion matrix and m is a positive integer not bigger than n. There exists a quaternion matrix $\boldsymbol{\mathcal{W}} \in \mathbb{Q}^{n\times m}$ with unitary columns such that*

$$\boldsymbol{\mathcal{W}}^*\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{H}} \tag{3.2}$$

is of upper Hessenberg form.

Proof. Based on the real structure-preserving transformations proposed in [4, 5], we can find a JRS-symplectic matrix with orthogonal columns

$$W := \begin{bmatrix} W_0 & -W_1 & -W_2 & -W_3 \\ W_1 & W_0 & -W_3 & W_2 \\ W_2 & W_3 & W_0 & -W_1 \\ W_3 & -W_2 & W_1 & W_0 \end{bmatrix} \in \mathbb{R}^{4n\times 4m} \tag{3.3}$$

such that

$$W^T\mathscr{R}(\boldsymbol{\mathcal{A}})W = H := \begin{bmatrix} H_0 & -H_1 & -H_2 & -H_3 \\ H_1 & H_0 & -H_3 & H_2 \\ H_2 & H_3 & H_0 & -H_1 \\ H_3 & -H_2 & H_1 & H_0 \end{bmatrix} \in \mathbb{R}^{4m\times 4m} \tag{3.4}$$

is an upper JRS-Hessenberg matrix, where $H_0 \in \mathbb{R}^{n\times n}$ is upper Hessenberg, and H_1, H_2, $H_3 \in \mathbb{R}^{n\times n}$ are upper triangular. Let $\boldsymbol{\mathcal{W}} = \mathscr{R}^{-1}(W)$ and $\boldsymbol{\mathcal{H}} = \mathscr{R}^{-1}(H)$; then (3.4) can be rewritten as

$$\mathscr{R}(\boldsymbol{\mathcal{W}})^T \mathscr{R}(\boldsymbol{\mathcal{A}}) \mathscr{R}(\boldsymbol{\mathcal{W}}) = \mathscr{R}(\boldsymbol{\mathcal{H}}). \tag{3.5}$$

By applying the inverse homeomorphic mapping $\mathscr{R}^{-1}$ on both sides of (3.5), we immediately obtain the partial upper Hessenberg form (3.2). ■

We remark that there are many different real counterparts of $\boldsymbol{\mathcal{A}}$, and all $\mathscr{R}(\boldsymbol{\mathcal{A}})$ are permutation equivalent [5, Remark 4.7]. Thus the results in Lemma 3.1 are valid for different real counterparts of $\boldsymbol{\mathcal{A}}$.

The decomposition in (3.4) is called the *structure preserving* of four real matrices A_0, A_1, A_2, $A_3 \in \mathbb{R}^{n\times n}$, since the JRS-symmetry of $\mathscr{R}(\boldsymbol{\mathcal{A}})$ is preserved. In the practical implementation, we need not generate the explicit real counterparts (the 4-by-4 real block matrices) of quaternion matrices but only generate and store their four parts (say, A_i's, W_j's and H_i's). This saves computational operations and storage. We refer to [4, 5] for the details. In contrast, the classical Arnoldi procedure [15] applied to $\mathscr{R}(\boldsymbol{\mathcal{A}})$ generates a matrix $V \in \mathbb{R}^{4n\times 4m}$ with orthogonal columns such that

$$V^T \mathscr{R}(\boldsymbol{\mathcal{A}}) V = \widehat{H} \in \mathbb{R}^{4m\times 4m},$$

where $\widehat{H} \in \mathbb{R}^{4m\times 4m}$ is of upper Hessenberg form. Here $\widehat{H}$ does not inherit the JRS-symmetry of $\mathscr{R}(\boldsymbol{\mathcal{A}})$ and $\widehat{H}$ cannot be mapped back to the quaternion upper Hessenberg matrix; see [5].

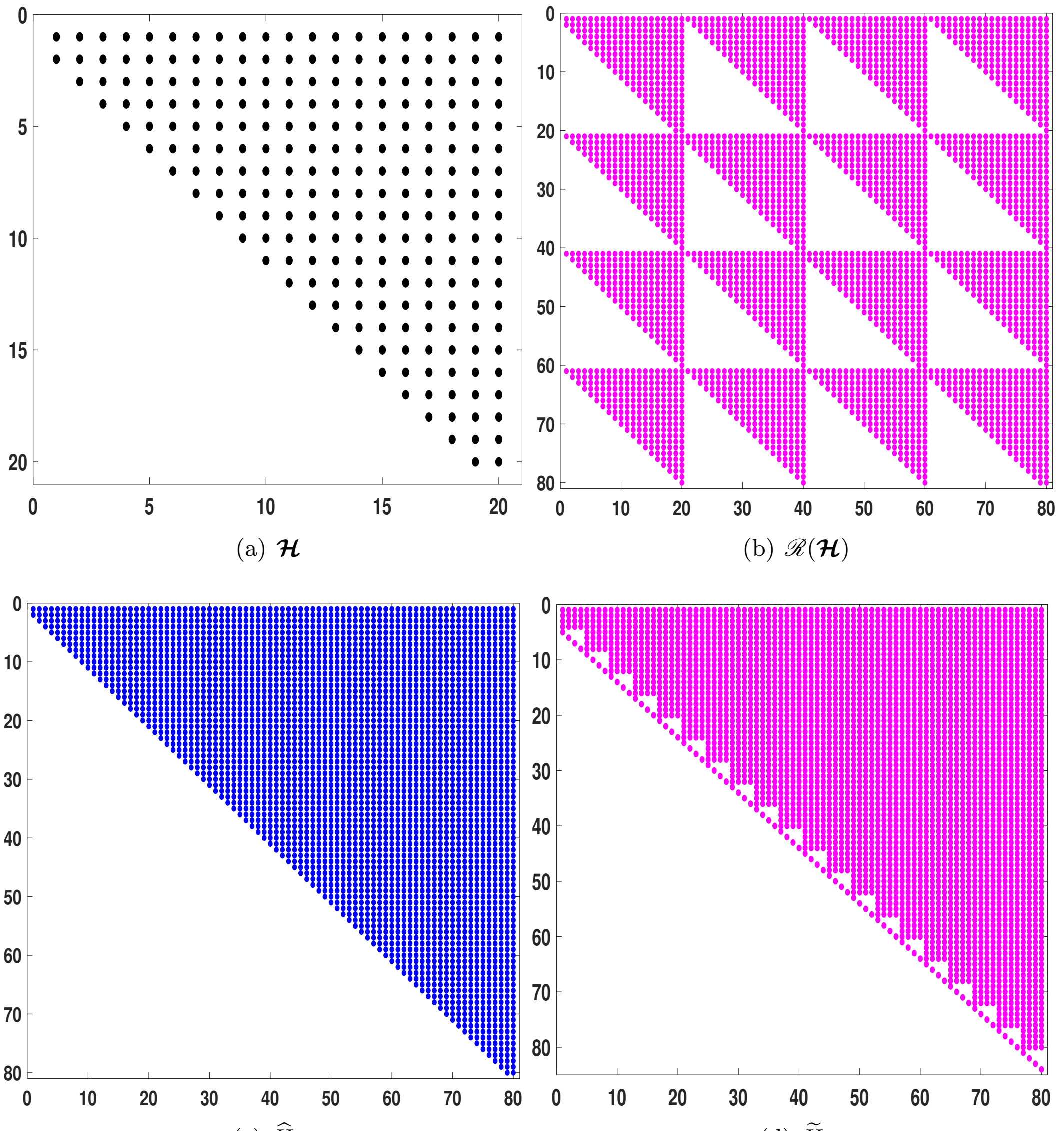
Example 3.1. *Suppose $\boldsymbol{\mathcal{A}} \in \mathbb{Q}^{n\times n}$ is a randomly generated quaternion matrix. Let $n = 1000$ and $m = 20$. By the structure preserving quaternion Arnoldi method (see Section 3.2), we compute the upper Hessenberg quaternion matrix $\boldsymbol{\mathcal{H}} = H_0 + H_1\mathbf{i} + H_1\mathbf{j} + H_3\mathbf{k} \in \mathbb{Q}^{20\times 20}$. By the classical Arnoldi method, we also compute the upper Hessenberg matrix $\widehat{H} \in \mathbb{R}^{80\times 80}$. The structures of $\boldsymbol{\mathcal{H}}$, $\mathscr{R}(\boldsymbol{\mathcal{H}})$ and $\widehat{H}$ are shown in Figure 1. We can see that $\mathscr{R}(\boldsymbol{\mathcal{H}})$ is JRS-symmetric with H_0 upper Hessenberg and H_1, H_2, H_3 upper triangular (see Figures 1(a) and 1(b)), but $\widehat{H}$ loses the JRS-symmetry of $\mathscr{R}(\boldsymbol{\mathcal{A}})$ (see Figure 1(c)).*

3.1. Quaternion Krylov Subspace.

The Krylov subspace of coefficient matrix $\boldsymbol{\mathcal{A}} \in \mathbb{F}^{n\times n}$ and a nonzero vector $\mathbf{v} \in \mathbb{F}^n$ is defined by

$$\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v}) := \mathtt{span}\{\mathbf{v}, \boldsymbol{\mathcal{A}}\mathbf{v}, \cdots, \boldsymbol{\mathcal{A}}^{m-1}\mathbf{v}\} \tag{3.6}$$

with any positive integer m. Each element of $\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})$ is a (right-hand-side) linear combination of $\mathbf{v}, \boldsymbol{\mathcal{A}}\mathbf{v}, \boldsymbol{\mathcal{A}}^2\mathbf{v}, \cdots$, and $\boldsymbol{\mathcal{A}}^{m-1}\mathbf{v}$. Note that quaternions are multiplicatively noncommutative, the above-mentioned linear combination cannot be rewritten as the multiplication of a polynomial with degree less than $m-1$ of $\boldsymbol{\mathcal{A}}$ and the vector $\mathbf{v}$. These results are different from those [15] when the field of numbers is real or complex. To develop the quaternion Arnoldi procedure, we need to directly rely on the (right-hand side) linear combination of multiplications of the power of quaternion matrix $\boldsymbol{\mathcal{A}}$ and the quaternion vector $\mathbf{v}$.



(a) $\boldsymbol{\mathcal{H}}$ (b) $\mathscr{R}(\boldsymbol{\mathcal{H}})$

(c) $\widehat{H}$ (d) $\widetilde{H}$

Figure 1. *The spy figures of $\boldsymbol{\mathcal{H}}$, $\mathscr{R}(\boldsymbol{\mathcal{H}})$, $\widehat{H}$, and $\widetilde{H}$.*

For convenience, the (right-hand-side) linear combination of $\mathbf{v}$, $\boldsymbol{\mathcal{A}}\mathbf{v}$, $\boldsymbol{\mathcal{A}}^2\mathbf{v}$, $\cdots$, and $\boldsymbol{\mathcal{A}}^{m-1}\mathbf{v}$ is denoted by

$$L_m(\boldsymbol{\mathcal{A}}, \mathbf{v}) := \mathbf{v}\boldsymbol{\alpha}_0 + \mathbf{A}\mathbf{v}\boldsymbol{\alpha}_1 + \cdots + \boldsymbol{\mathcal{A}}^{m-1}\mathbf{v}\boldsymbol{\alpha}_{m-1}, \tag{3.7}$$

where $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{m-1}$ are quaternion scalars. The dimension of $\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})$ is exactly the rank of the Krylov subspace matrix, $\boldsymbol{\mathcal{K}} := [\mathbf{v}, \boldsymbol{\mathcal{A}}\mathbf{v}, \boldsymbol{\mathcal{A}}^2\mathbf{v}, \cdots, \boldsymbol{\mathcal{A}}^{m-1}\mathbf{v}]$. It is expected to increase with m increasing, before arriving at the maximal value, called the grade of $\mathbf{v}$ (with respect to $\boldsymbol{\mathcal{A}}$).

Definition 3.2 (grade). *The* grade *of* $\mathbf{v}$ *with respect to* $\boldsymbol{\mathcal{A}}$ *is the smallest positive integer,* μ*, such that* $\mathbf{v}, \boldsymbol{\mathcal{A}}\mathbf{v}, \cdots, \boldsymbol{\mathcal{A}}^{\mu-1}\mathbf{v}$ *are linearly independent but* $\mathbf{v}, \boldsymbol{\mathcal{A}}\mathbf{v}, \cdots, \boldsymbol{\mathcal{A}}^{\mu-1}\mathbf{v}$, $\boldsymbol{\mathcal{A}}^{\mu}\mathbf{v}$ *are linearly dependent.*

Below are some important properties of quaternion Krylov subspace $\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$.

Theorem 3.3. *Let* μ *be the grade of* $\mathbf{v}$ *with respect to* $\boldsymbol{\mathcal{A}}$*. Then* $\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$ *is invariant under* $\boldsymbol{\mathcal{A}}$ *and* $\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v}) = \mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$ *for all* $m \geq \mu$*.*

Proof. We first prove that $\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$ is invariant under $\boldsymbol{\mathcal{A}}$. According to the definition (3.6),

$$\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v}) := \mathtt{span}\{\mathbf{v}, \mathbf{A}\mathbf{v}, \boldsymbol{\mathcal{A}}^2\mathbf{v}, \cdots, \boldsymbol{\mathcal{A}}^{\mu-1}\mathbf{v}\}.$$

Each vector in $\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$ can be denoted by $L_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$, as in (3.7), and then

$$\boldsymbol{\mathcal{A}}L_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v}) = \boldsymbol{\mathcal{A}}\mathbf{v}\boldsymbol{\alpha}_0 + \boldsymbol{\mathcal{A}}^2\mathbf{v}\boldsymbol{\alpha}_1 + \cdots + \boldsymbol{\mathcal{A}}^{\mu}\mathbf{v}\boldsymbol{\alpha}_{\mu-1}.$$

Since the grade of $\mathbf{v}$ is μ, the quaternion vector $\boldsymbol{\mathcal{A}}^{\mu}\mathbf{v}$ can be linearly represented by former quaternion vectors $\mathbf{v}, \boldsymbol{\mathcal{A}}\mathbf{v}, \cdots, \boldsymbol{\mathcal{A}}^{\mu-1}\mathbf{v}$. So that $\boldsymbol{\mathcal{A}}^{\mu}\mathbf{v}$ still belongs to $\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$ and thus $\boldsymbol{\mathcal{A}}\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v}) \subseteq \mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$. We have proved that $\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$ is invariant under $\boldsymbol{\mathcal{A}}$. The second part of the conclusion, $\mathscr{K}_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v}) = \mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})$, holds naturally, because any quaternion vector $L_m(\boldsymbol{\mathcal{A}}, \mathbf{v})$ can be rewritten into $L_\mu(\boldsymbol{\mathcal{A}}, \mathbf{v})$ for any positive integer m bigger than μ. ■

Theorem 3.4. $\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})$ *is of dimension* m *if and only if the grade* μ *of* $\mathbf{v}$ *with respect to* $\boldsymbol{\mathcal{A}}$ *is greater than or equal to* m*, i.e.,*

$$\dim(\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})) = m \Leftrightarrow \mu \geq m.$$

Therefore,

$$\dim(\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})) = \min\{m, \mu\}. \tag{3.8}$$

Proof. The proof follows Theorem 3.3 immediately. ■

Theorem 3.5. *Let* $\boldsymbol{\mathcal{Q}}_m$ *be any projector onto* $\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})$ *and* $\boldsymbol{\mathcal{A}}_m := \boldsymbol{\mathcal{Q}}_m\boldsymbol{\mathcal{A}}_{|\mathscr{K}_m}$ *the section of* $\boldsymbol{\mathcal{A}}$ *to* $\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})$*. Then for any linear combination* $L_i(\boldsymbol{\mathcal{A}}, \mathbf{v})$ *of the form* (3.7)*,*

$$L_i(\boldsymbol{\mathcal{A}}, \mathbf{v}) = L_i(\boldsymbol{\mathcal{A}}_m, \mathbf{v}),\ 1 \leq i \leq m, \tag{3.9}$$

and

$$\boldsymbol{\mathcal{Q}}_m L_{m+1}(\boldsymbol{\mathcal{A}}, \mathbf{v}) = L_{m+1}(\boldsymbol{\mathcal{A}}_m, \mathbf{v}). \tag{3.10}$$

Proof. First we prove that $L_i(\boldsymbol{\mathcal{A}}, \mathbf{v}) = L_i(\boldsymbol{\mathcal{A}}_m, \mathbf{v})$ holds for $1 \leq i \leq m$. It is sufficient to show the property holds for the special case $L_i(\boldsymbol{\mathcal{A}}, \mathbf{v}) = \boldsymbol{\mathcal{A}}^{i-1}\mathbf{v}$, $i = 1, \cdots, m$. The property is true for $i = 1$. Assume that it is true for i $(1 \leq i < m)$, that is,

$$L_i(\boldsymbol{\mathcal{A}}, \mathbf{v}) = L_i(\boldsymbol{\mathcal{A}}_m, \mathbf{v}). \tag{3.11}$$

Multiplying (3.11) by $\boldsymbol{\mathcal{A}}$ on both sides yields

$$L_{i+1}(\boldsymbol{\mathcal{A}}, \mathbf{v}) = \boldsymbol{\mathcal{A}}^{i+1}\mathbf{v} = \boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{A}}^{i}\mathbf{v}) = \boldsymbol{\mathcal{A}}L_i(\boldsymbol{\mathcal{A}}, \mathbf{v}) = \boldsymbol{\mathcal{A}}L_i(\boldsymbol{\mathcal{A}}_m, \mathbf{v}). \tag{3.12}$$

If $i \leq m-1$, we have $L_{i+1}(\boldsymbol{\mathcal{A}}, \mathbf{v}) \in \mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v})$, and therefore if (3.12) is multiplied on both sides by $\boldsymbol{\mathcal{Q}}_m$, then

$$L_{i+1}(\boldsymbol{\mathcal{A}}, \mathbf{v}) = \boldsymbol{\mathcal{Q}}_m\boldsymbol{\mathcal{A}}L_i(\boldsymbol{\mathcal{A}}_m, \mathbf{v}). \tag{3.13}$$

Under the assumption (3.11), $L_i(\boldsymbol{\mathcal{A}}_m, \mathbf{v}) \in \mathscr{K}_m$ holds for $i \leq m-1$. Looking at the right-hand side of (3.13), we observe that

$$\boldsymbol{\mathcal{Q}}_m\boldsymbol{\mathcal{A}}L_i(\boldsymbol{\mathcal{A}}_m, \mathbf{v}) = \boldsymbol{\mathcal{Q}}_m\boldsymbol{\mathcal{A}}_{|\mathscr{K}_m}L_i(\boldsymbol{\mathcal{A}}_m, \mathbf{v}) = L_{i+1}(\boldsymbol{\mathcal{A}}_m, \mathbf{v}). \tag{3.14}$$

Combining (3.13) and (3.14), we have $L_{i+1}(\boldsymbol{\mathcal{A}}, \mathbf{v}) = L_{i+1}(\boldsymbol{\mathcal{A}}_m, \mathbf{v})$ with $i \leq m-1$. By induction, we have proved (3.9). Now it only remains to prove (3.10). Multiplying the equation $L_m(\boldsymbol{\mathcal{A}}, \mathbf{v}) = L_m(\boldsymbol{\mathcal{A}}_m, \mathbf{v})$ by $\boldsymbol{\mathcal{Q}}_m\boldsymbol{\mathcal{A}}$ on both sides, we can immediately observe $\boldsymbol{\mathcal{Q}}_m L_{m+1}(\boldsymbol{\mathcal{A}}, \mathbf{v}) = L_{m+1}(\boldsymbol{\mathcal{A}}_m, \mathbf{v})$. ■

According to Theorem 3.5, one can see that any element represented in the power basis of $\mathscr{K}_m(A, v)$ can be represented in the power basis of $\mathscr{K}_m(A_m, v)$.

3.2. Quaternion Structure Preserving Procedure.

The quaternion Arnoldi method is an algorithm for building an orthogonal basis of the quaternion Krylov subspace; see Algorithm 3.1.

Algorithm 3.1 Quaternion Arnoldi procedure

Choose a quaternion vector $\mathbf{v}_1$, such that $\|\mathbf{v}_1\|_2 = 1$
for $j = 1, 2, \cdots, m$ **do**
    Compute $\boldsymbol{\omega}_j = \boldsymbol{\mathcal{A}}\mathbf{v}_j$
    for $i = 1, 2, \cdots, j$ **do**
        Compute $\mathbf{h}_{ij} = \langle \boldsymbol{\mathcal{A}}\mathbf{v}_j, \mathbf{v}_i \rangle$
        Compute $\boldsymbol{\omega}_j := \boldsymbol{\omega}_j - \mathbf{v}_i\mathbf{h}_{ij}$
    end for
    $\mathbf{h}_{j+1,j} = \|\boldsymbol{\omega}_j\|_2$
    If $\mathbf{h}_{j+1,j} = 0$ then stop
    $\mathbf{v}_{j+1} = \boldsymbol{\omega}_j / \mathbf{h}_{j+1,j}$
end for

In the calculation, the required quaternion matrix-vector multiplication and inner product are realized by the function (3.1) as follows,

$$\boldsymbol{\mathcal{A}}\mathbf{v}_j = \mathscr{R}^{-1}\big(\mathscr{R}(\boldsymbol{\mathcal{A}}\mathbf{v}_j)\big) = \mathscr{R}^{-1}\big(\mathscr{R}(\boldsymbol{\mathcal{A}})\mathscr{R}(\mathbf{v}_j)\big), \tag{3.15}$$

$$\langle \boldsymbol{\mathcal{A}}\mathbf{v}_j, \mathbf{v}_i \rangle = \mathscr{R}^{-1}\big(\mathscr{R}(\mathbf{v}_i^*\boldsymbol{\mathcal{A}}\mathbf{v}_j)\big) = \mathscr{R}^{-1}\big(\mathscr{R}(\mathbf{v}_i^*)\mathscr{R}(\boldsymbol{\mathcal{A}}\mathbf{v}_j)\big). \tag{3.16}$$

The core work is on the real calculation of $\mathscr{R}(\boldsymbol{\mathcal{A}})\mathscr{R}(\mathbf{v}_j)$ and $\mathscr{R}(\mathbf{v}_i^*)\mathscr{R}(\boldsymbol{\mathcal{A}}\mathbf{v}_j)$. Note that $\mathscr{R}(\boldsymbol{\mathcal{A}})\mathscr{R}(\mathbf{v}_j)$ consists of four real matrix-vector products and $\mathscr{R}(\mathbf{v}_i^*)\mathscr{R}(\boldsymbol{\mathcal{A}}\mathbf{v}_j)$

needs four inner products of real vectors. With the above realization (3.15) and (3.16), we are in fact implementing the structure preserving reduction of four real parts of $\boldsymbol{\mathcal{A}}$ as in (3.4). Assume that Algorithm 3.1 does not stop before the mth step. Then the quaternion vectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m$ form an orthonormal basis of the quaternion Krylov subspace $\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{v}_1) = \mathtt{span}\{\mathbf{v}_1, \boldsymbol{\mathcal{A}}\mathbf{v}_1, \cdots, \boldsymbol{\mathcal{A}}^{m-1}\mathbf{v}_1\}$. Let $\boldsymbol{\mathcal{V}}_m := [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m]$, $\bar{\boldsymbol{\mathcal{H}}}_m$ denote the $(m+1) \times m$ quaternion Hessenberg matrix whose nonzero entries $\mathbf{h}_{ij}$ are calculated by Algorithm 3.1, and $\boldsymbol{\mathcal{H}}_m$ be the matrix obtained from $\bar{\boldsymbol{\mathcal{H}}}_m$ by deleting its last row. Then the following relations hold:

$$\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_m\boldsymbol{\mathcal{H}}_m + \boldsymbol{\omega}_m\mathbf{e}_m^* \tag{3.17}$$

$$= \boldsymbol{\mathcal{V}}_{m+1}\bar{\boldsymbol{\mathcal{H}}}_m, \tag{3.18}$$

$$\boldsymbol{\mathcal{V}}_m^*\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{H}}_m, \tag{3.19}$$

where $\mathbf{e}_m$ is the mth column of the identity matrix. Let $\boldsymbol{\mathcal{H}}_m = H_0 + H_1\mathbf{i} + H_2\mathbf{j} + H_3\mathbf{k}$. Relation (3.19) is indeed computed by the structure preserving reduction:

$$\mathscr{R}(\boldsymbol{\mathcal{V}}_m)^T\mathscr{R}(\boldsymbol{\mathcal{A}})\mathscr{R}(\boldsymbol{\mathcal{V}}_m) = \mathscr{R}(\boldsymbol{\mathcal{H}}_m),$$

where $\mathscr{R}(\boldsymbol{\mathcal{V}}_m)$ is an orthogonal matrix with algebraic symmetry as in (3.3) and $\mathscr{R}(\boldsymbol{\mathcal{H}}_m)$ has the structure on the right-hand side of (3.4).

Note that Algorithm 3.1 may break down in case the norm of $\boldsymbol{\omega}_j$ vanishes at a certain step j. In this case, the vector $\mathbf{v}_{j+1}$ cannot be computed, and $\mathscr{K}_j(\boldsymbol{\mathcal{A}}, \mathbf{v}_1)$ is an invariant subspace of $\boldsymbol{\mathcal{A}}$.

Theorem 3.6. *The quaternion Arnoldi algorithm breaks down at the jth step (i.e., $h_{j+1,j} = 0$ in Algorithm 3.1) if and only if the grade of $\mathbf{v}_1$ with respect to $\boldsymbol{\mathcal{A}}$ is j. Moreover, in this case the quaternion subspace $\mathscr{K}_j(\boldsymbol{\mathcal{A}}, \mathbf{v}_1)$ is invariant under $\boldsymbol{\mathcal{A}}$.*

Proof. If the degree of $\mathbf{v}_1$ is j, then $\boldsymbol{\omega}_j$ must be equal to zero. Indeed, otherwise $\mathbf{v}_{j+1}$ can be defined and as a result $\mathscr{K}_{j+1}(\boldsymbol{\mathcal{A}}, \mathbf{v}_1)$ would be of dimension $j+1$. That means $\mathbf{v}_1, \boldsymbol{\mathcal{A}}\mathbf{v}_1, \cdots, \boldsymbol{\mathcal{A}}^{j-1}\mathbf{v}_1, \boldsymbol{\mathcal{A}}^j\mathbf{v}_1$ are linearly independent. Then Theorem 3.4 would imply that $\mu \geq j+1$, which is a contradiction. To prove the converse, assume that $\boldsymbol{\omega}_j = \mathbf{0}$. Then the degree μ of $\mathbf{v}_1$ must not be smaller than j. It is impossible that $\mu < j$. Otherwise, by the first part of this proof, the vector $\boldsymbol{\omega}_\mu$ would be zero and the algorithm would have stopped at the earlier step number μ. The rest of the result follows from Theorem 3.3. ∎

3.3. Modified Quaternion Arnoldi Procedure. In a practical computational process, one needs to apply the modified Gram–Schmidt, instead of the standard Gram–Schmidt, to guarantee the orthogonality of the generated $\mathbf{v}_i$'s. With the modified quaternion Gram–Schmidt alternative, we get the modified quaternion Arnoldi procedure in Algorithm 3.2. As in the traditional case, Algorithms 3.1 and 3.2 are mathematically equivalent to each other in exact arithmetic, while the later one is much more reliable in the presence of the round-off.

Again, the matrix-vector products and inner products in Algorithm 3.2 are implemented by the real calculations in (3.15) and (3.16). In lines 4 to 7 of Algorithm

Algorithm 3.2 Modified Quaternion Arnoldi procedure

```
1: Choose a quaternion vector v_1 of norm 1
2: for j = 1, 2, ··· , m do
3:    Compute ω_j := A v_j
4:    for i = 1, 2, ··· , j do
5:       Compute h_ij = ⟨ω_j, v_i⟩
6:       Compute ω_j := ω_j − v_i h_ij
7:    end for
8:    h_{j+1,j} = ‖ω_j‖_2
9:    If h_{j+1,j} = 0 then stop
10:   v_{j+1} = ω_j / h_{j+1,j}
11: end for
```

3.2, we generate a quaternion vector $\boldsymbol{\omega}_j$ orthogonal to $\mathbf{v}_1, \cdots, \mathbf{v}_j$ by realization, i.e.,

$$\boldsymbol{\omega}_j = \mathscr{R}^{-1}(\mathscr{R}(\boldsymbol{\mathcal{A}})\mathscr{R}(\mathbf{v}_j) - \mathscr{R}(\mathbf{v}_1)\mathscr{R}(\mathbf{h}_{1j}) - \mathscr{R}(\mathbf{v}_2)\mathscr{R}(\mathbf{h}_{2j}) - \cdots - \mathscr{R}(\mathbf{v}_j)\mathscr{R}(\mathbf{h}_{jj})) \tag{3.20}$$

with theoretically $\mathbf{h}_{ij} = \langle \boldsymbol{\mathcal{A}}\mathbf{v}_j - \mathbf{v}_1\mathbf{h}_{1j} - \mathbf{v}_2\mathbf{h}_{2j} - \cdots - \mathbf{v}_{i-1}\mathbf{h}_{i-1,j}, \mathbf{v}_i \rangle$ (which is equivalent to $\mathbf{h}_{ij} = \langle \boldsymbol{\mathcal{A}}\mathbf{v}_j, \mathbf{v}_i \rangle$ in exact arithmetic, but more stable in practical implementation). Immediately, one can obtain

$$\boldsymbol{\mathcal{A}}\mathbf{v}_j = \mathbf{v}_1\mathbf{h}_{1j} + \mathbf{v}_2\mathbf{h}_{2j} + \cdots + \mathbf{v}_j\mathbf{h}_{jj} + \mathbf{v}_{j+1}\mathbf{h}_{j+1,j}.$$

This finally produces the upper Hessenberg decomposition (3.17) and (3.18).

Remark. If $\boldsymbol{\mathcal{A}}$ is Hermitian, the reduced upper Hessenberg form $\boldsymbol{\mathcal{H}}_m$ will be tridiagonal. Indeed, $\boldsymbol{\mathcal{H}}_m^* = \boldsymbol{\mathcal{V}}_m^*\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_m^*\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{H}}_m$. In this case, the coefficients $\mathbf{h}_{ij}$ computed by Algorithm 3.2 are such that

$$\mathbf{h}_{ij} = 0, \text{ for } 1 \le i \le j-1, \text{ and } \mathbf{h}_{j,j+1} = \mathbf{h}_{j+1,j} \in \mathbb{R}, \ j = 1, \cdots, m.$$

Moreover, $\mathbf{h}_{jj}$ is real since $\mathbf{h}_{jj} = \langle \boldsymbol{\mathcal{A}}\mathbf{v}_j, \mathbf{v}_j \rangle = \langle \boldsymbol{\mathcal{A}}^*\mathbf{v}_j, \mathbf{v}_j \rangle = \mathbf{h}_{jj}^*$.

In the implementation, we apply the real operations on the four parts of quaternion scalars, vectors, and matrices, without storing their real counterparts. This saves the real operations and storage memory. The quaternion matrix-vector products in line 3 are theoretically computed by the real structure preserving methods as shown in (3.15). Here we generate the quaternion vector $\boldsymbol{\omega}_j$ once the first block column of $\mathscr{R}(\boldsymbol{\omega}_j)$ is obtained. That is, let $\mathbf{v}_j = v_0 + v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k}$, then

$$\boldsymbol{\omega}_j = w_0 + w_1\mathbf{i} + w_2\mathbf{j} + w_3\mathbf{k}, \tag{3.21a}$$

$$\left[\begin{array}{cccc} w_0^T & w_1^T & w_2^T & w_3^T \end{array}\right]^T = \mathscr{R}(\boldsymbol{\mathcal{A}}) \left[\begin{array}{cccc} v_0^T & v_1^T & v_2^T & v_3^T \end{array}\right]^T. \tag{3.21b}$$

Similarly, the computation in line 5 of Algorithm 3.2 is implemented by

$$\mathbf{h}_{ij} = h_0 + h_1\mathbf{i} + h_2\mathbf{j} + h_3\mathbf{k}, \tag{3.22a}$$

$$\left[\begin{array}{cccc} h_0 & h_1 & h_2 & h_3 \end{array}\right]^T = \mathscr{R}(\mathbf{v}_i)^T \left[\begin{array}{cccc} w_0^T & w_1^T & w_2^T & w_3^T \end{array}\right]^T. \tag{3.22b}$$

Applying (3.21) and (3.22) saves three-quarters of the theoretical costs. That is, the quaternion matrix-vector and inner products in our method cost only $4n(8n-1)$ and $4(8n-1)$ real operations, respectively.

3.4. Comparison with the Block Arnoldi Algorithm. Each quaternion entry of $\boldsymbol{\mathcal{A}}$ can be represented by its real counterpart, and the quaternion coefficient matrix in (1.1) can be written as follows:

$$\widetilde{A} = \begin{bmatrix} \mathscr{R}(\mathbf{a}_{11}) & \mathscr{R}(\mathbf{a}_{12}) & \cdots & \mathscr{R}(\mathbf{a}_{1n}) \\ \mathscr{R}(\mathbf{a}_{21}) & \mathscr{R}(\mathbf{a}_{22}) & \cdots & \mathscr{R}(\mathbf{a}_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathscr{R}(\mathbf{a}_{n1}) & \mathscr{R}(\mathbf{a}_{n2}) & \cdots & \mathscr{R}(\mathbf{a}_{nn}) \end{bmatrix} \in \mathbb{R}^{4n\times 4n}, \tag{3.23}$$

where the real counterpart of $\mathbf{a}_{ij} = a_{ij}^{(0)} + a_{ij}^{(1)}\mathbf{i} + a_{ij}^{(2)}\mathbf{j} + a_{ij}^{(3)}\mathbf{k}$ with $a_{ij}^{(0)}$, $a_{ij}^{(1)}$, $a_{ij}^{(2)}$, $a_{ij}^{(3)} \in \mathbb{R}$, is given by

$$\begin{bmatrix} a_{ij}^{(0)} & -a_{ij}^{(1)} & -a_{ij}^{(2)} & -a_{ij}^{(3)} \\ a_{ij}^{(1)} & a_{ij}^{(0)} & -a_{ij}^{(3)} & a_{ij}^{(2)} \\ a_{ij}^{(2)} & a_{ij}^{(3)} & a_{ij}^{(0)} & -a_{ij}^{(1)} \\ a_{ij}^{(3)} & -a_{ij}^{(2)} & a_{ij}^{(1)} & a_{ij}^{(0)} \end{bmatrix}. \tag{3.24}$$

Actually, $\widetilde{A}$ is permutation equivalent to the matrix in (3.1). Note that $\widetilde{A}$ can be viewed as a block matrix with each block of size 4-by-4. Recently, the block Arnoldi algorithm based on *-algebra for solving block linear systems was developed in [9] and [2]. For the detailed algorithm, we refer to Algorithm 1 in [9]. When this block Arnoldi method is applied to $\widetilde{A}$, we obtain the following decomposition:

$$\widetilde{A}\widetilde{V} = \widetilde{V}\widetilde{H}, \qquad \widetilde{V},\ \widetilde{H} \in \mathbb{R}^{4n\times 4n}, \tag{3.25}$$

after n iterations. Here $\widetilde{H}$ is block upper Hessenberg, i.e., $\widetilde{H}_{i,j} = 0$ if $j < i-1$. Moreover, this block Arnoldi method can preserve block structure in $\widetilde{H}$, i.e., each block is in the form of (3.24) or each block is a real counterpart of a quaternion number. As an example, we show the structure of $\widetilde{H}$ in Figure 1(d).

Indeed, $\mathbb{Q}$ can be seen as a *-algebra but it has the additional structure of not having any zero-divisors. This is one way to understand why QGMRES never suffers from linear dependence breakdowns the way block GMRES does (see Section 4.2). In the block GMRES, the Givens rotations in the more general *-algebra context are suggested to reduce the block upper Hessenberg matrix $\widetilde{H}$ in (3.25) to the upper triangular form [9, Section 3.1]. Although QGMRES and block GMRES are related, we are able to exploit more specific structural properties of quaternion linear systems to get a more efficient algorithm (Algorithm 4.1). See the numerical results in Section 5 for the comparisons.

4. Quaternion Generalized Minimal Residual Method.

In this section, we propose the QGMRES for solving large-scale quaternion linear systems as in (1.1). Let $\mathbf{x}_0$ represent an arbitrary initial guess to the solution of linear systems (1.1) and the residual vector $\mathbf{r}_0 = \mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x}_0$. Since if $\mathbf{r}_0 = 0$, then $\mathbf{x}_0$ is the solution of (1.1), we always assume that $\mathbf{r}_0 \neq 0$ in the following discussion. Similar to (3.6), the quaternion Krylov subspace is given by

$$\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{r}_0) = \mathtt{span}\{\mathbf{r}_0, \boldsymbol{\mathcal{A}}\mathbf{r}_0, \boldsymbol{\mathcal{A}}^2\mathbf{r}_0, \cdots, \boldsymbol{\mathcal{A}}^{m-1}\mathbf{r}_0\}. \tag{4.1}$$

We rewrite $\mathscr{K}_m(\boldsymbol{\mathcal{A}}, \mathbf{r}_0)$ in the short form $\mathscr{K}_m$ in the following discussion. Any quaternion vector $\mathbf{x}$ in $\mathbf{x}_0 + \mathscr{K}_m$ can be written as

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{V}_m\mathbf{y}, \tag{4.2}$$

where $\mathbf{y}$ is an m-vector. The relation (3.19) results in

$$\begin{aligned} \mathbf{b} - \mathbf{A}\mathbf{x} = \mathbf{b} - \boldsymbol{\mathcal{A}}(\mathbf{x}_0 + \mathbf{V}_m\mathbf{y}) \;&=\; \mathbf{r}_0 - \mathbf{A}\mathbf{V}_m\mathbf{y} \\ = \beta\mathbf{v}_1 - \mathbf{V}_{m+1}\bar{\mathbf{H}}_m\mathbf{y} \;&=\; \mathbf{V}_{m+1}(\beta\mathbf{e}_1 - \bar{\mathbf{H}}_m\mathbf{y}), \end{aligned} \tag{4.3}$$

where $\beta = \|\mathbf{r}_0\|_2 \neq 0,\ \mathbf{v}_1 = \mathbf{r}_0/\beta$. Defining

$$J(\mathbf{y}) := \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2, \tag{4.4}$$

one can observe from the orthogonality of the columns of $\mathbf{V}_{m+1}$ that

$$J(\mathbf{y}) = \|\beta\mathbf{e}_1 - \bar{\mathbf{H}}_m\mathbf{y}\|_2. \tag{4.5}$$

The QGMRES approximation is the unique quaternion vector from $\mathbf{x}_0 + \mathscr{K}_m$ which minimizes (4.4), and has the general form:

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m, \tag{4.6}$$

where

$$\mathbf{y}_m = \arg\ \min_{\mathbf{y}} \|\beta\mathbf{e}_1 - \bar{\mathbf{H}}_m\mathbf{y}\|_2. \tag{4.7}$$

Note that it is inexpensive to compute the minimizer $\mathbf{y}_m$ of the upper Hessenberg quaternion least-squares problem (HQLS) (4.7) when m is small. The algorithm is summarized in Algorithm 4.1.

Algorithm 4.1 QGMRES

```
1: Compute r_0 = b − 𝓐x_0, β := ‖r_0‖_2 ≠ 0, and v_1 := r_0/β
2: for j = 1, 2, ⋯ , m do
3:     Compute ω_j := 𝓐v_j
4:     for i = 1, 2, ⋯ , j do
5:         h_ij = ⟨ω_j, v_i⟩;
6:         ω_j := ω_j − h_ij v_i
7:     end for
8:     h_{j+1,j} = ‖ω_j‖_2
9:     If h_{j+1,j} = 0 set m := j and go to 12
10:    v_{j+1} = ω_j / h_{j+1,j}
11: end for
12: Define (m + 1) × m quaternion Hessenberg matrix H̄_m = [h_ij]_{1≤i≤m+1, 1≤j≤m}.
13: Compute y_m the minimizer of ‖βe_1 − H̄_m y‖_2 and x_m = x_0 + V_m y_m.
```

4.1. Solving Upper Hessenberg Quaternion Least-Squares Problem. Suppose that m steps of the QGMRES iteration have been performed and an $(m+1)\times m$ upper quaternion Hessenberg matrix $\bar{\boldsymbol{\mathcal{H}}}_m = [\mathbf{h}_{ij}]$ has been generated. Suppose that $\mathbf{r}_{ii} = \|\,[\mathbf{h}_{ii}\ \mathbf{h}_{i+1,i}]^T\,\|_2 \neq 0$ for $i = 1,\cdots,m$. If the subdiagonal element $\mathbf{h}_{i+1,i}$ of $\bar{\boldsymbol{\mathcal{H}}}_m$ is nonzero, we define the ith generalized quaternion Givens transformation by

$$\Omega_i = \begin{bmatrix} \mathbf{I}_{i-1} & & & \\ & \mathbf{g}_{11} & \mathbf{g}_{12} & \\ & \mathbf{g}_{21} & \mathbf{g}_{22} & \\ & & & \mathbf{I}_{m-i} \end{bmatrix},$$

where

$$\mathbf{g}_{11} = \frac{\mathbf{h}_{ii}}{\mathbf{r}_{ii}},\ \ \mathbf{g}_{21} = \frac{\mathbf{h}_{i+1,i}}{\mathbf{r}_{ii}}, \tag{4.8a}$$

$$\begin{cases} \mathbf{g}_{12} = |\mathbf{g}_{21}|,\ \ \mathbf{g}_{22} = -|\mathbf{g}_{21}|\mathbf{g}_{21}^{-*}\mathbf{g}_{11}^{*} & \text{if} |\mathbf{h}_{ii}| \le |\mathbf{h}_{i+1,i}|; \\ \mathbf{g}_{22} = |\mathbf{g}_{11}|,\ \ \mathbf{g}_{12} = -|\mathbf{g}_{11}|\mathbf{g}_{11}^{-*}\mathbf{g}_{21}^{*} & \text{if} |\mathbf{h}_{ii}| > |\mathbf{h}_{i+1,i}|. \end{cases} \tag{4.8b}$$

Define a unitary quaternion matrix $\boldsymbol{\mathcal{Q}}_m$ as the product of matrices Ω_i, i.e., $\boldsymbol{\mathcal{Q}}_m = \Omega_1\Omega_2\cdots\Omega_m$. Then we obtain the QR factorization of $\bar{\boldsymbol{\mathcal{H}}}_m$ and can reduce the upper Hessenberg least-squares problem (4.7) to an upper tridiagonal least squares problem

$$\mathbf{y}_m = \arg\ \min_{\mathbf{y}} \|\bar{\mathbf{g}}_m - \bar{\boldsymbol{\mathcal{R}}}_m \mathbf{y}\|_2, \tag{4.9}$$

in which

$$\bar{\boldsymbol{\mathcal{R}}}_m = \boldsymbol{\mathcal{Q}}_m^* \bar{\boldsymbol{\mathcal{H}}}_m = [\mathbf{r}_{ij}],\quad \bar{\mathbf{g}}_m = \boldsymbol{\mathcal{Q}}_m^*(\beta \mathbf{e}_1) := \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_m & \gamma_{m+1} \end{bmatrix}^T. \tag{4.10}$$

Notice that the generalized quaternion Givens rotation defined by (4.8) is quite different from the standard Givens rotations, for instance, $\mathbf{g}_{11} \neq \bar{\mathbf{g}}_{22}$ in the general case but $|\mathbf{g}_{11}| = |\mathbf{g}_{22}|$; see [5, Section 4.1] for more information. The diagonal entries of $\bar{\boldsymbol{\mathcal{R}}}_m$ are nonnegative real numbers, which makes it convenient to compute $\mathbf{y}_m$.

Now we analyze the solution of HQLS (4.7) and the residue vector under the above notation. These results generalize those on the real field in [15] to the quaternion skew-field.

Theorem 4.1. *Let $m \le n$, and let $\boldsymbol{\mathcal{R}}_m, \mathbf{g}_m$ denote the $m \times m$ upper triangular quaternion matrix and the m-dimensional quaternion vector obtained from $\bar{\boldsymbol{\mathcal{R}}}_m, \bar{\mathbf{g}}_m$ by deleting their last row and component, respectively.*

(1) *The rank of $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{V}}_m$ is equal to the rank of $\boldsymbol{\mathcal{R}}_m$; in particular, if $\mathbf{r}_{mm} = 0$, then $\boldsymbol{\mathcal{A}}$ must be singular.*

(2) *The vector $\mathbf{y}_m$ which minimizes $\|\beta\mathbf{e}_1 - \bar{\boldsymbol{\mathcal{H}}}_m\mathbf{y}\|_2$ is given by $\mathbf{y}_m = \boldsymbol{R}_m^{-1}\boldsymbol{g}_m$.*

(3) *The residual vector at step m satisfies*

$$\mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x}_m = \boldsymbol{\mathcal{V}}_{m+1}(\beta\mathbf{e}_1 - \bar{\boldsymbol{\mathcal{H}}}_m\mathbf{y}_m) = \boldsymbol{\mathcal{V}}_{m+1}\boldsymbol{\mathcal{Q}}_m(\boldsymbol{\gamma}_{m+1}\mathbf{e}_{m+1}), \tag{4.11}$$

and as a result,

$$\|\mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x}_m\|_2 = |\boldsymbol{\gamma}_{m+1}|. \tag{4.12}$$

Proof. These results can be proved in a similar way to those in [15]. ■

Without virtually additional arithmetic operations, one can obtain the residual norm in a progressive manner at each step of the QGMRES algorithm. Assume that the first m rotations have already been applied, say, start with (4.10); and the residual norm is available for $\mathbf{x}_m$ and the stopping criterion can be applied. If the residual norm, $|\boldsymbol{\gamma}_{m+1}|$, is small enough, the process must be stopped. To compute $\mathbf{y}_m$, we develop a back substitution method on the quaternion skew-field (the details are omitted here). Then the approximate solution is computed by $\mathbf{x}_m = \mathbf{x}_0 + \boldsymbol{\mathcal{V}}_m\mathbf{y}_m$. Otherwise, further steps are needed.

4.2. Breakdown of QGMRES . The only possibility of breakdown in the QGMRES algorithm (Algorithm 4.1) is when $\boldsymbol{\omega}_j = \mathbf{0}$, i.e., when $\mathbf{h}_{j+1,j} = 0$ at a given step j in the quaternion Arnoldi loop. In this situation, the algorithm stops because the next quaternion Arnoldi vector cannot be generated. The residual vector is zero in this case. That means the algorithm delivers the exact solution at this step. Fortunately, the converse is also true: If the algorithm stops at step j with $\mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x}_j = 0$, then $\mathbf{h}_{j+1,j} = 0$.

Theorem 4.2. *Let $\boldsymbol{\mathcal{A}}$ be a nonsingular quaternion matrix. Then, the QGMRES algorithm breaks down at step j, i.e., $\mathbf{h}_{j+1,j} = 0$, if and only if the approximate solution $\mathbf{x}_j$ is exact.*

Proof. If $\mathbf{h}_{j+1,j} = 0$, then $\mathbf{g}_{21} = 0$. Indeed, since $\boldsymbol{\mathcal{A}}$ is nonsingular, then $\mathbf{r}_{jj}$ is nonzero by the first part of Theorem 4.1 and (4.8) implies $\mathbf{g}_{21} = 0$. Then, the relations (4.12) and $\boldsymbol{\gamma}_{j+1} = \mathbf{g}_{12}^*\boldsymbol{\gamma}_j$ imply that $\mathbf{r}_j = 0$. Conversely, if the approximation is exact at step j and not at step $j-1$, then $\mathbf{g}_{21} = 0$. From the formula (4.8), this implies that $\mathbf{h}_{j+1,j} = 0$. ■

4.3. Convergence of QGMRES . In this section, we establish an upper bound on the convergence rate of the QGMRES iterates first and then present a global convergence result.

Let $\boldsymbol{\mathcal{A}}$ be an arbitrary square quaternion matrix and assume that $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{K}}$. Then generalizing the result in [15, Proposition 5.3] to the quaternion skew-field, a vector $\tilde{\mathbf{x}}$ is the result of an (oblique) projection method onto $\mathscr{K}$ orthogonally to $\mathscr{L}$ with the starting vector $\mathbf{x}_0$ if and only if it minimizes the 2-norm of the residual vector $\mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x}$ over $\mathbf{x} \in \mathbf{x}_0 + \mathscr{K}$, i.e., if and only if

$$R(\tilde{\mathbf{x}}) = \min_{\mathbf{x}\in\mathbf{x}_0+\mathscr{K}} R(\mathbf{x}),$$

where $R(\mathbf{x}) := \|\mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x}\|_2$. Indeed, for $\tilde{\mathbf{x}}$ to be the minimum solution, it is necessary and sufficient that $\mathbf{y}^*(\mathbf{b} - \boldsymbol{\mathcal{A}}\tilde{\mathbf{x}}) = 0$ holds for any $\mathbf{y} \in \boldsymbol{\mathcal{A}}\mathscr{K}$, which is precisely the Petrov–Galerkin condition that defines the approximate solution. Again, notice that each element of $\mathscr{K}$ in QGMRES is represented by a right-hand-side linear combination of quaternion vectors $\mathbf{r}_0, \boldsymbol{\mathcal{A}}\mathbf{r}_0, \cdots, \boldsymbol{\mathcal{A}}^{m-1}\mathbf{r}_0$, rather than a production of a polynomial of $\boldsymbol{\mathcal{A}}$ and the vector $\mathbf{r}_0$ in GMRES.

Theorem 4.3. *Let $\mathbf{x}_m$ be the approximate solution obtained from the mth step of the QGMRES algorithm, and let $\mathbf{r}_m = \mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x}_m$. Then, $\mathbf{x}_m$ is of the form*

$$\mathbf{x}_m = \mathbf{x}_0 + L_m(\boldsymbol{\mathcal{A}}, \mathbf{r}_0),$$

and $\|\mathbf{r}_m\|_2 = \|\mathbf{r}_0 - \boldsymbol{\mathcal{A}}L_m(\boldsymbol{\mathcal{A}}, \mathbf{r}_0)\|_2 = \min_{j\le m} \|\mathbf{r}_0 - \boldsymbol{\mathcal{A}}L_j(\boldsymbol{\mathcal{A}}, \mathbf{r}_0)\|_2$.

Proof. Recall the definition (3.7) that

$$L_m(\boldsymbol{\mathcal{A}}, \mathbf{r}_0) = \mathbf{r}_0\boldsymbol{\alpha}_0 + \boldsymbol{\mathcal{A}}\mathbf{r}_0\boldsymbol{\alpha}_1 + \cdots + \boldsymbol{\mathcal{A}}^{m-1}\mathbf{r}_0\boldsymbol{\alpha}_{m-1}.$$

The result follows the fact that $\mathbf{x}_m$ minimizes the 2-norm of the residual in the affine subspace $\mathbf{x}_0 + \mathscr{K}_m$ and the fact that $\mathscr{K}_m$ is the set of all vectors of the form $L_j(\boldsymbol{\mathcal{A}}, \mathbf{r}_0)$ with $j \le m$. ■

Theorem 4.4. *Assume that $\boldsymbol{\mathcal{A}}$ is a diagonalizable quaternion matrix and let $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{X}}\boldsymbol{\Lambda}\boldsymbol{\mathcal{X}}^{-1}$, where $\boldsymbol{\Lambda} = \mathrm{diag}\{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \cdots, \boldsymbol{\lambda}_n\}$ is the diagonal matrix of eigenvalues. Define*

$$\epsilon^{(m)} = \min_{\boldsymbol{\alpha}_1,\cdots,\boldsymbol{\alpha}_j\in\mathbb{Q},\ j\le m} \left(\max_{i=1,\cdots,n} (1 + |\boldsymbol{\lambda}_i||\boldsymbol{\alpha}_1| + \cdots + |\boldsymbol{\lambda}_i^j||\boldsymbol{\alpha}_j|) \right).$$

Then, the residual norm achieved by the mth step of QGMRES satisfies the inequality

$$\|\mathbf{r}_m\|_2 \le \kappa_2(\boldsymbol{\mathcal{X}})\epsilon^{(m)}\|\mathbf{r}_0\|_2,$$

where $\kappa_2(\boldsymbol{\mathcal{X}}) := \|\boldsymbol{\mathcal{X}}\|_2\|\boldsymbol{\mathcal{X}}^{-1}\|_2$.

Proof. Let $L_{j+1}(\boldsymbol{\mathcal{A}}, \mathbf{r}_0)$ satisfy the constraint $L_{j+1}(0, \mathbf{r}_0) = \mathbf{r}_0$ ($\boldsymbol{\alpha}_0 = 1$), and $\mathbf{x}$ the vector from $\mathscr{K}_m$ to which it is associated via $\mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x} = L_{j+1}(\boldsymbol{\mathcal{A}}, \mathbf{r}_0)$. Then

$$\begin{aligned}\|\mathbf{b} - \boldsymbol{\mathcal{A}}\mathbf{x}\|_2 &= \|\boldsymbol{\mathcal{X}}L_{j+1}(\boldsymbol{\Lambda}, \boldsymbol{\mathcal{X}}^{-1}\mathbf{r}_0)\|_2 \le \|\boldsymbol{\mathcal{X}}\|_2\|L_{j+1}(\boldsymbol{\Lambda}, \boldsymbol{\mathcal{X}}^{-1}\mathbf{r}_0)\|_2 \\ &\le \|\boldsymbol{\mathcal{X}}\|_2\|\boldsymbol{\mathcal{X}}^{-1}\|_2\|\mathbf{r}_0\|_2\|I + |\boldsymbol{\Lambda}||\boldsymbol{\alpha}_1| + \cdots + |\boldsymbol{\Lambda}^j||\boldsymbol{\alpha}_j|\|_2.\end{aligned}$$

Since $\mathbf{\Lambda}$ is diagonal,

$$\|I+|\mathbf{\Lambda}||\boldsymbol{\alpha}_1|+\cdots+|\mathbf{\Lambda}^j||\boldsymbol{\alpha}_j|\|_2 \le \max_{i=1,\cdots,n}(1+|\boldsymbol{\lambda}_i||\boldsymbol{\alpha}_1|+\cdots+|\boldsymbol{\lambda}_i^j||\boldsymbol{\alpha}_j|).$$

Since $\mathbf{x}_m$ minimizes the residual norm over $\mathbf{x}_0+\mathscr{K}_m$, then for any combining form $L_{j+1}(\boldsymbol{\mathcal{A}},\mathbf{r}_0)$,

$$\|\mathbf{b}-\boldsymbol{\mathcal{A}}\mathbf{x}_m\|_2 \le \|\mathbf{b}-\boldsymbol{\mathcal{A}}\mathbf{x}\|_2 \le \|\boldsymbol{\mathcal{X}}\|_2\|\boldsymbol{\mathcal{X}}^{-1}\|_2\|\mathbf{r}_0\|_2 \max_{i=1,\cdots,n}(1+|\boldsymbol{\lambda}_i||\boldsymbol{\alpha}_1|+\cdots+|\boldsymbol{\lambda}_i^j||\boldsymbol{\alpha}_j|).$$

This yields the desired result,

$$\|\mathbf{b}-\boldsymbol{\mathcal{A}}\mathbf{x}_m\|_2 \le \|\mathbf{b}-\boldsymbol{\mathcal{A}}\mathbf{x}\|_2 \le \|\boldsymbol{\mathcal{X}}\|_2\|\boldsymbol{\mathcal{X}}^{-1}\|_2\|\mathbf{r}_0\|_2\epsilon^{(m)}.$$

5. Numerical Experiments.

This section compares the proposed QGMRES with the classical GMRES and the block GMRES (blkGMRES) in [9] for quaternion linear systems. All experiments were performed by MATLAB (R2016a) on a personal computer with an Intel Core 64 × 4Core i5-3470 CPU @ 3.20 GHz/8.00 GB.

A three-dimensional signal can be denoted by a quaternion function of time, $\mathbf{x}(t)=x_r(t)\mathbf{i}+x_g(t)\mathbf{j}+x_b(t)\mathbf{k}$, where $x_r(t), x_g(t)$, and $x_b(t)$ are real functions (for example, they refer to the red, green, and blue channels, respectively). We are interested in determining quaternion filters $\{\mathbf{w}(s)\}_{s=0}^{n}$, where $\mathbf{w}(s)=w(s)_0+w(s)_r\mathbf{i}+w(s)_g\mathbf{j}+w(s)_b\mathbf{k}$ on the input signal $\mathbf{x}(t)=x_r(t)\mathbf{i}+x_g(t)\mathbf{j}+x_b(t)\mathbf{k}$ such that the filtered output can match with the target signal $\mathbf{y}(t)=y_r(t)\mathbf{i}+y_g(t)\mathbf{j}+y_b(t)\mathbf{k}$. More precisely, we have

$$\mathbf{y}(t)=\sum_{s=0}^{n}\mathbf{x}(t-s)*\mathbf{w}(s). \tag{5.1}$$

Let

$$\boldsymbol{\mathcal{X}}=\begin{bmatrix}\mathbf{x}(t) & \mathbf{x}(t-1) & \mathbf{x}(t-2) & \cdots & \mathbf{x}(t-n)\\ \mathbf{x}(t+1) & \mathbf{x}(t) & \mathbf{x}(t-1) & \cdots & \mathbf{x}(t-n+1)\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \mathbf{x}(t+m) & \mathbf{x}(t+m-1) & \mathbf{x}(t+m-2) & \cdots & \mathbf{x}(t-n+m)\end{bmatrix}, \tag{5.2a}$$

$$\mathbf{w}=\begin{bmatrix}\mathbf{w}(0) & \mathbf{w}(1) & \mathbf{w}(2) & \cdots & \mathbf{w}(n)\end{bmatrix}^T, \tag{5.2b}$$

$$\mathbf{y}=\begin{bmatrix}\mathbf{y}(t) & \mathbf{y}(t+1) & \mathbf{y}(t+2) & \cdots & \mathbf{y}(t+m)\end{bmatrix}^T. \tag{5.2c}$$

Then the process (5.1) can be rewritten as

$$\boldsymbol{\mathcal{X}}*\mathbf{w}=\mathbf{y}. \tag{5.3}$$

Let $\boldsymbol{\mathcal{X}}=X_0+X_1\mathbf{i}+X_2\mathbf{j}+X_3\mathbf{k}\in\mathbb{Q}^{N\times N}$, $\mathbf{y}=y_0+y_1\mathbf{i}+y_2\mathbf{j}+y_3\mathbf{k}\in\mathbb{Q}^N$, and $\mathbf{w}=w_0+w_1\mathbf{i}+w_2\mathbf{j}+w_3\mathbf{k}\in\mathbb{Q}^N$. QGMRES is directly applied to solve the

$N \times N$ quaternion linear systems (5.3); GMRES is utilized to solve the $4N \times 4N$ real counterpart,

$$\mathscr{R}(\boldsymbol{\mathcal{X}}) \begin{bmatrix} w_0^T & w_1^T & w_2^T & w_3^T \end{bmatrix}^T = \begin{bmatrix} y_0^T & y_1^T & y_2^T & y_3^T \end{bmatrix}^T; \tag{5.4}$$

and blkGMRES is applied to solve the following $4N \times 4N$ real linear systems with multiple right-hand sides:

$$\widetilde{X}\widetilde{\mathrm{w}} = \widetilde{\mathrm{y}}, \tag{5.5}$$

where $\widetilde{X} = [\mathscr{R}(\boldsymbol{\mathcal{X}}_{ij})] \in \mathbb{R}^{4n\times 4n}$, $\widetilde{\mathrm{w}} = [\mathscr{R}(\mathbf{w}_i)] \in \mathbb{R}^{4n\times 4}$, and $\widetilde{\mathrm{y}} = [\mathscr{R}(\mathbf{y}_i)] \in \mathbb{R}^{4n\times 4}$ are block matrices with each block of size 4-by-4, defined similarly by (3.23). For the above three methods, the relative residual error is defined by

$$Residual = \|\boldsymbol{\mathcal{X}} * \mathbf{w} - \mathbf{y}\|_2 / \|\mathbf{y}\|_2.$$

The stopping criteria of all these iterative methods are that the relative residual error is less than `tol` = 1.0e-6 and the maximum number of iterations is the number of unknowns.

Example 5.1. *The Lorenz attractor is a three-dimensional nonlinear system used originally to model atmospheric turbulence [18]. Mathematically, the Lorenz system can be expressed as a system of coupled differential equations*

$$\frac{\partial x}{\partial t} = \alpha(y - x), \ \frac{\partial y}{\partial t} = x(\rho - z) - y, \ \frac{\partial z}{\partial t} = xy - \beta z, \tag{5.6}$$

where $\alpha, \ \beta, \ \rho > 0$. *For the chaotic behavior of Lorenz attractor, the parameters were selected as* $\boldsymbol{\alpha} = 10, \ \beta = 8/3, \ \rho = 28$. *The coupled differential equations* (5.6) *are solved by the MATLAB order* $\mathtt{ODE45}(f(t, [x, y, z]), [0, T], [1, 1, 1])$, *where* $T > 0$.

In this example, we consider

$$\mathbf{y}(t) = y_r(t)\mathbf{i} + y_g(t)\mathbf{j} + y_b(t)\mathbf{k}$$

where $y_r(t)$, $y_g(t)$ *and* $y_b(t)$ *are the solutions of the Lorenz attractor. And the input*

$$\mathbf{x}(t) = y_r(t-1)\mathbf{i} + y_g(t-1)\mathbf{j} + y_b(t-1)\mathbf{k} + \mathbf{n}(t),$$

where $\mathbf{n}(t)$ *is a random noise. The quaternion linear systems is built as in* (5.3) *with* (5.2).

The numerical results of applying GMRES, blkGMRES, and QGMRES to solve (5.3) *are listed in Table* 1. *We can see from the table that QGMRES converges in many fewer iterations and costs less CPU time than GMRES, while their residual errors are comparable with each other. The convergence curves are shown in Figure* 2, *which indicates that QGMRES stops earlier than GMRES. QGMRES preserves Hessenberg structure in each quaternion component. GMRES preserves Hessenberg structure in expanded real representation. However, the storage size of QGMRES is lower than that of GMRES. To solve the HQLS problem* (4.7), *we only need*

$N-1$ generalized quaternion Givens rotations and solve an N-dimensional upper triangular quaternion linear systems. To solve the corresponding real Hessenberg least-squares problem, we need $4N-1$ real Givens rotations and to solve a $4N$-dimensional real triangular linear systems.

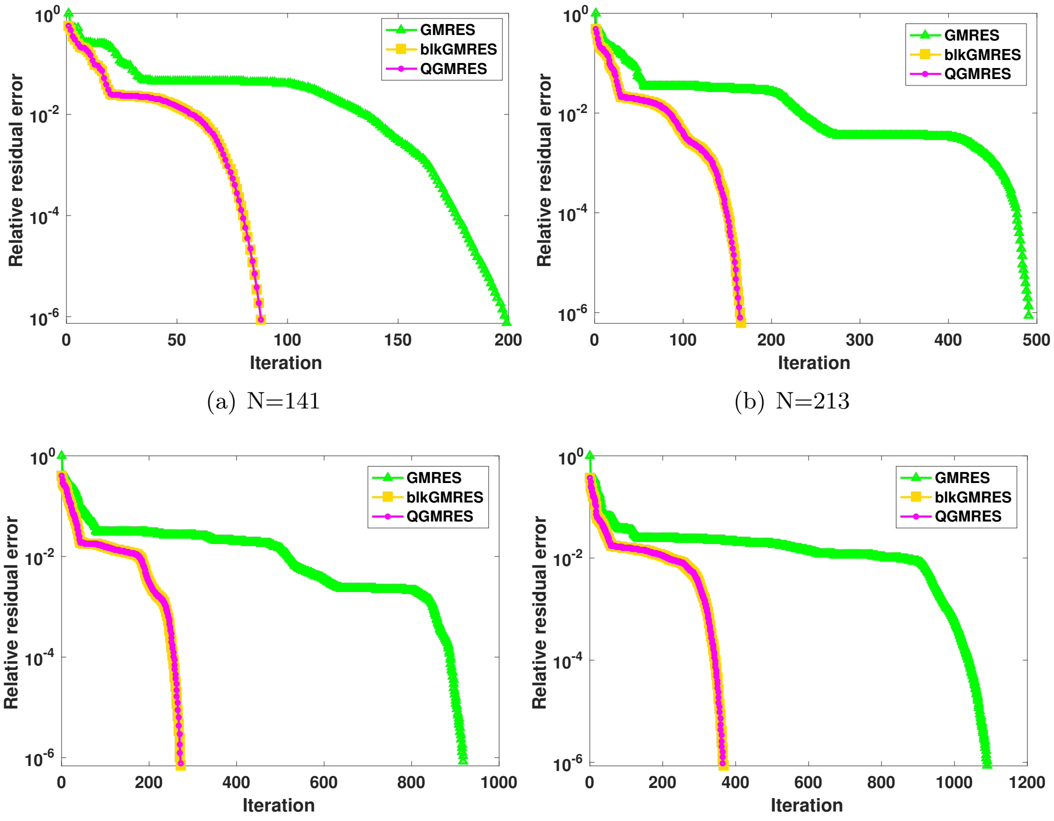
Both blkGMRES and QGMRES are structure preserving. blkGMRES computes the block upper Hessenberg matrix based on 4-by-4 block matrices via *-algebra, while QGMRES computes the upper Hessenberg matrix based on four components of quaternion numbers. We observe in Table 1 and Figures 2 and 3 their iteration numbers are almost the same, but the computational time required by QGMRES is lower than that by blkGMRES.

Table 1
Numerical results of GMRES, blkGMRES and QGMRES.

N	Method	Dimension	Iteration	CPU time	Residual
141	GMRES	564	199	6.0743	7.3936e-07
	blkGMRES	564	**88**	7.3898	8.8265e-07
	QGMRES	**141**	**88**	**3.1740**	8.9152e-07
213	GMRES	852	488	121.0523	8.7783e-07
	blkGMRES	852	165	66.2665	6.0925e-07
	QGMRES	**213**	**164**	**22.7382**	7.7894e-07
333	GMRES	1332	908	770.0953	8.7061e-07
	blkGMRES	1332	**272**	458.8207	6.8805e-07
	QGMRES	**333**	**272**	**82.9526**	7.8674e-07
425	GMRES	1700	1096	1.9022e+03	8.8433e-07
	blkGMRES	1700	366	1.4368e+03	8.4987e-07
	QGMRES	**425**	**365**	**0.2053e+03**	9.5885e-07

Example 5.2. *In this example, we consider the color image pixel prediction problem. Let $\boldsymbol{\mathcal{A}} = A_1\mathbf{i}+A_2\mathbf{j}+A_3\mathbf{k} \in \mathbb{Q}^{N_1\times N_2}$ denote a color image, where $A_1,\ A_2,\ A_3 \in \mathbb{R}^{N_1\times N_2}$ represent the red, green, and blue channels. And let $\boldsymbol{\mathcal{A}}$ be patterned into $k_1 \times k_2$ patches of size $n \times n$ say, $\{\boldsymbol{\mathcal{A}}_{ij} \in \mathbb{Q}^{n\times n} | i = 1, \cdots, k_1,\ j = 1, \cdots, k_2\}$. By concatenating the columns of each patch $\boldsymbol{\mathcal{A}}_{ij}$, we obtain an n^2-dimensional quaternion vector. Each pixel value in the formed vector is predicted by its previous pixel values with a random noise via solving the quaternion linear system in (5.1).*

The testing color image is shown in Figure 4(a) whose size is 256×256, i.e., $N_1 = N_2 = 256$. We test four cases of partitioning: $k_1 = k_2 = 8$, $k_1 = k_2 = 16$, $k_1 = 256, k_2 = 2$ and $k_1 = 2, k_2 = 256$. The numerical results are listed in Table 2, in which the notation has the following meanings: "Numblk" denotes the number of patches, "Dimension" denotes the dimension of the linear quaternion systems to predict one patch, and "Iteration," "CPU time," and "Residual" denote the average iteration, CPU time and relative residual error required by QGMRES or GMRES applied on one patch, respectively. In Figure 4, we show the visual comparison in the case that $k_1 = k_2 = 8$. Figures 4(a)-(b) are the original and observed color images, respectively. Figures 4(c)-(e) are the predicted color images by GMRES, blkGMRES, and QGMRES, respectively. We can see that the QGMRES



(a) N=141 (b) N=213

(c) N=333 (d) N=425

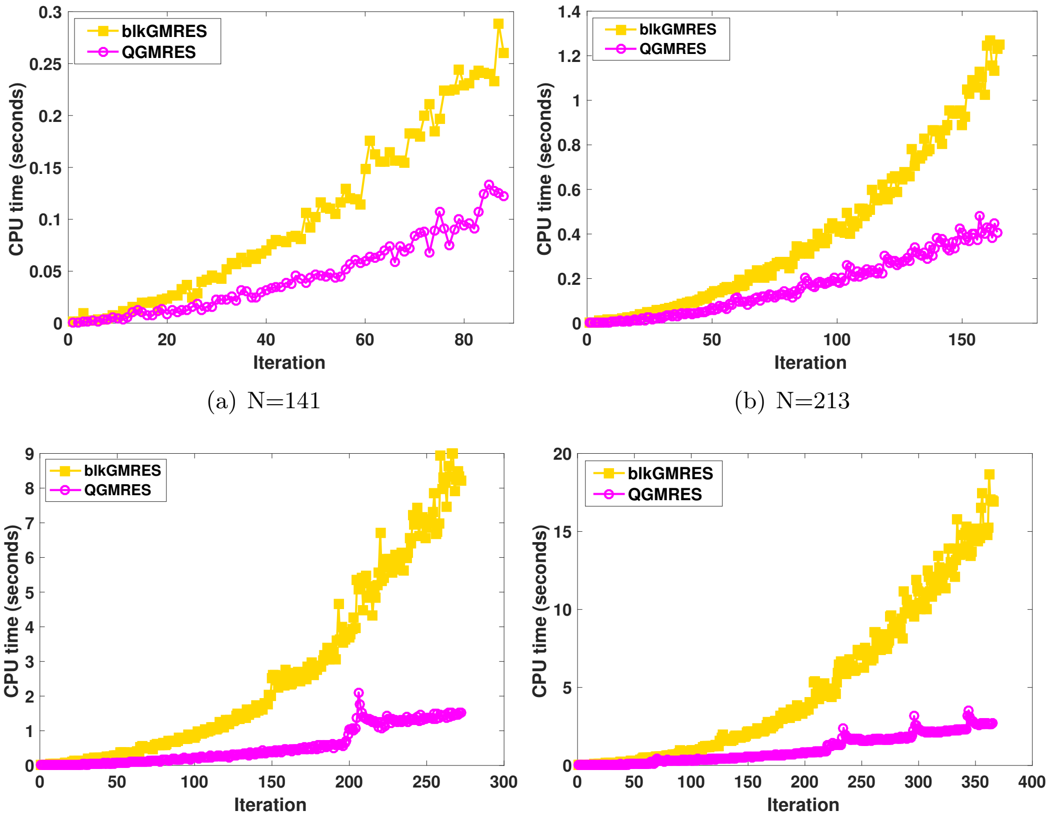
Figure 2. *The relative residual errors of GMRES, blkGMRES and QGMRES in four cases:* $N = 141,\ 213,\ 333,$ *and* 425.

method converges in fewer iterations than the GMRES method. The residual errors of QGMRES are much smaller than those of GMRES. Again, blkGMRES converges in almost the same number of iterations as QGMRES, but costs more in CPU time than QGMRES. The predicted color image by QGMRES is visually the same as the original one and is comparable with those computed by GMRES and blkGMRES.

6. Conclusion.

In this paper, we proposed the QGMRES to solve general quaternion linear systems. The main contributions are listed as follows.

- The quaternion Arnoldi method is first proposed and its basic theory is well developed with overcoming the multiplicative noncommutativity of quaternions. A new structure preserving method is presented to implement the quaternion Arnoldi procedure.
- QGMRES is newly presented for solving the general and large-scale quaternion linear systems, as well as the convergence analysis. The upper HQLS problem is solved by a new efficient solver.
- QGMRES is successfully applied to the three-dimensional signal filtering and

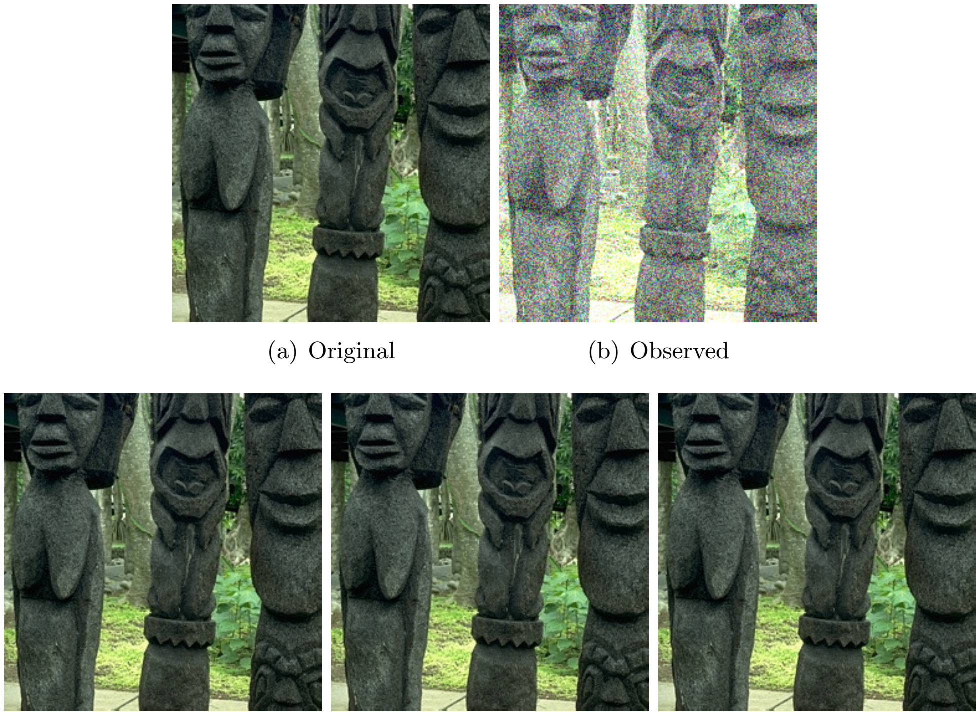


(a) N=141 (b) N=213 (c) N=333 (d) N=425

Figure 3. *The CPU times cost by the Arnoldi procedures of blkGMRES and QGMRES at the mth iteration in four cases:* $N = 141,\ 213,\ 333,$ *and* 425.

Table 2
Color image prediction: GMRES vs. QGMRES

SizeBlk (k_1, k_2)	NumBlk	Method	Dimension	Iteration	CPU time	Residual
(8,8)	1024	GMRES	256	173	3.3523	4.7726e-07
		blkGMRES	256	**63**	2.1578	5.1705e-08
		QGMRES	**64**	**63**	**1.3992**	5.8309e-08
(16,16)	256	GMRES	1024	804	561.3812	8.1980e-07
		blkGMRES	1024	256	266.8387	5.2922e-07
		QGMRES	**256**	**255**	**85.5188**	1.5324e-07
(256,2)	128	GMRES	2048	1892	1.1411e+04	7.8323e-07
		blkGMRES	2048	**511**	5.1404e+03	6.4565e-07
		QGMRES	**512**	**511**	**0.9451e+03**	1.3042e-07
(2,256)	128	GMRES	2048	1890	1.1790e+04	8.2625e-07
		blkGMRES	2048	512	5.8392e+03	5.7220e-08
		QGMRES	**512**	**511**	**0.9488e+03**	1.6005e-08



(a) Original (b) Observed

(c) GMRES (d) blkGMRES (e) QGMRES

Figure 4. *The visual comparison of color image prediction: (a)-(b) the orginal and observed color images; (c)-(e) the predicted color images by GMRES, blkGMRES and QGMRES.*

color image processing.

In the future, we will study the Householder version of the Arnoldi method, the restarted QGMRES, and the preconditioned QGMRES for other applications.

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

REFERENCES

[1] W. R. Hamilton. Elements of Quaternions, Longmans, Green, London, 1866.

[2] A. Frommer, K. Lund, and D. B. Szyld. Block Krylov subspace methods for functions of matrices, *Electron. Trans. Numer. Anal.*, 47: 100–126, 2017.

[3] R. Ghiloni, V. Moretti, and A. Perotti. Continuous slice functional calculus in quaternionic Hilbert spaces, *Rev. Math. Phys.*, 25(04): 1350006, 2013.

[4] Z. G. Jia, M. S. Wei, and S. T. Ling. A new structure-preserving method for quaternion Hermitian eigenvalue problems, *J. Comput. Appl. Math.*, 239: 12–24, 2013.

[5] Z. G. Jia, M. S. Wei, M. X. Zhao, and Y. Chen. A new real structure-preserving quaternion QR algorithm, *J. Comput. Appl. Math.*, 343: 26–48, 2018.

[6] Z. G. Jia, M. K. Ng, and G. J. Song. Robust quaternion matrix completion with applications to image inpainting, *Numer. Linear Algebra Appl.*, 26(4), e2245, 2019.

[7] Z. G. Jia, M. K. Ng, and G. J. Song. Lanczos method for large-scale quaternion singular value decomposition, *Numer. Algorithms*, 82(2): 699–717, 2019.

[8] Z. G. Jia, M. K. Ng, and W. Wang. Color image restoration by saturation-value (SV) total variation, *SIAM J. Imaging Sci.*, 12(2):972–1000, 2019.

[9] M. Kubínová and K. M. Soodhalter. Admissible and attainable convergence behavior of block Arnoldi and GMRES, *SIAM J. Matrix Anal. Appl.*, 41(2):464–486, 2020.

[10] C. Lanczos. Solution of systems of linear equations by minimized iterations, *J. Res. Nat. Bur. Stand.*, 49(1):33–53, 1952.

[11] T. Parcollet, M. Morchid, G. Linarès. A survey of quaternion neural networks, *AI Rev.*, 53, 2957–2982, 2020.

[12] L. Rodman. Topics in Quaternion Linear Algebra, Princeton University Press, Princeton, NJ, 2014.

[13] L. S. Saoud, R. Ghorbani, and F. Rahmoune, Cognitive quaternion valued neural network and some applications, *Neurocomputing*, 221: 85–93, 2017.

[14] P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 10(1): 36–52, 1989.

[15] Y. Saad. Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, 2003.

[16] Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems, *Math. Comput.*, 37(155): 105–126, 1981.

[17] Y. Saad, M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 7(3): 856–869, 1986.

[18] S. H. Strogatz. Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering, Westview Press, Boulder, Co, 2001.

[19] C. C. Took and D. P. Mandic. The quaternion LMS algorithm for adaptive filtering of hypercomplex processes, *IEEE Trans. Signal Process.*, 57(4):1316–1327, 2009.

[20] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 13(2): 631–644, 1992.

[21] M. S. Wei, Y. Li, F. X. Zhang, and J. L. Zhao. Quaternion Matrix Computations, Nova Science Publishers, Hauppauge, NY, 2018.

[22] F. Zhang. Quaternions and matrices of quaternions, *Linear Algebra Appl.*, 251: 21–57, 1997.