TAIWANESE JOURNAL OF MATHEMATICS
Vol. 29, No. 1, pp. 21–34, February 2025
DOI: 10.11650/tjm/240902

Structure-preserving Quaternion Generalized Conjugate Residual Method

Xin-Fang Zhang, Wei Ding and Tao Li*

Abstract. Color image deblurring problem, one of the primary goals in color image processing, can be mathematically represented by quaternion linear systems. In this paper, we propose a new quaternion generalized conjugate residual method for solving the general quaternion linear systems. The main advantages are that the proposed method saves three-quarters of the theoretical costs compared to the traditional GCR iterations for the real representation of quaternion linear systems, and also converges faster and smoother than the quaternion biconjugate gradient (QBiCG) method. Finally, we provide several numerical examples to illustrate the effectiveness of the proposed method compared with some existing methods.

1. Introduction

Quaternions [7], discovered by Sir W. R. Hamilton in 1843, and quaternion matrices [8, 9, 21] play a key role in various fields, such as computer graphics [18], neural networks [16], quantum mechanics [4], especially in color image processing [11]. It is well-known that a color image, containing the red, green, and blue channels (RGB), can be expressed as a pure quaternion matrix. Actually, the matrix also maintains the coupling information between the three channels. The quaternion skew-field is associative but not commutative algebra over the real number field $\mathbb{R}$, expressed as

$$\mathbb{Q} = \big\{x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \mid \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1, x_0, x_1, x_2, x_3 \in \mathbb{R}\big\},$$

where $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are three imaginary units. Let $\mathbb{Q}^n$ and $\mathbb{Q}^{n\times n}$ denote the collections of all n-dimensional vectors and $n \times n$ matrices, respectively.

Received February 13, 2024; Accepted September 2, 2024.
Communicated by Eric Chung.
2020 *Mathematics Subject Classification.* 15B33, 65F10, 94A08.
Key words and phrases. quaternion linear systems, quaternion generalized conjugate residual method, color image deblurring.
This work was funded by the National Natural Science Foundation of China (grant numbers 12401019 and 12401493), Hainan Provincial Natural Science Foundation of China (grant numbers 122MS001 and 122QN214), and the Academic Programs project of Hainan University (grant numbers KYQD(ZR)-21119 and KYQD(ZR)-21151).
*Corresponding author.

In this paper, we mainly focus on solving the following non-Hermitian quaternion linear systems

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

where the invertible quaternion matrix $\mathbf{A} \in \mathbb{Q}^{n\times n}$ and quaternion vector $\mathbf{b} \in \mathbb{Q}^n$ are given, and the quaternion vector $\mathbf{x} \in \mathbb{Q}^n$ is unknown. Such linear systems have widespread applications in the area of scientific and engineering computing, such as quaternion quantum mechanics [5], image processing [10], and pose control [2,17], etc., and also have attracted many scholars' attention. However, the non-commutativity of quaternion multiplication poses a significant challenge for solving the quaternion linear systems. As far as we know, most existing algorithms for solving (1.1) are based on complex or real representations to overcome their non-commutativity. For example, using the linear homeomorphic mapping $\mathscr{R}(\cdot)$ from quaternion matrices (or vectors) to their real counterparts, (1.1) can be converted into a real matrix equation

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

where $\mathscr{R}(\mathbf{A}) \in \mathbb{R}^{4n\times 4n}$, $\mathscr{R}(\mathbf{x})$ and $\mathscr{R}(\mathbf{b}) \in \mathbb{R}^{4n}$. One can see that the dimension of (1.2) is expanded to four times the original dimension. This means that the storage and cost required by the traditional iterative algorithms for searching its solutions are greatly increased. Therefore, this paper aims to develop a high-performance structure-preserving algorithm for solving the original quaternion linear systems (1.1). Without expanding the dimension, the algorithm not only preserves the algebraic symmetry of the real counterparts, but also saves the computational operations and storage.

Recently, several structure-preserving quaternion Krylov subspace methods for solving (1.1) have been well-developed. These methods reduce three-quarters of the arithmetic operations compared to the traditional Krylov subspace methods over $\mathbb{R}$. In 2021, Jia and Ng [10] first established the quaternion Arnoldi procedure (QAP), and then derived the structure-preserving quaternion generalized minimal residual method (QGMRES) method for solving (1.1). After that, Li and Wang [14] proposed the quaternion full orthogonalization method (QFOM) and its preconditioned variant to solve (1.1), which converges faster than QGMRES. Very recently, Li and Wang [15] introduced the quaternion Lanczos biorthogonalization procedure, and then derived the quaternion biconjugate gradient (QBiCG) method. However, the unpredictable convergence behavior of the QBiCG method may lead to erratic oscillations in the residual norm, despite its superior performance compared to QGMRES in some cases. Herein, we will propose the quaternion generalized conjugate residual (QGCR) method, which exhibits enhanced robustness, for solving (1.1). From a projection point of view, the developed QGCR method is equivalent

to QGMRES [10] and retains the optimality of residual minimization. Compared with QBiCG, the proposed algorithm often gives smoother convergence behavior and performs much better in computing time and iterations. Specifically, the QGCR solver presented here differs in the following aspects:

- Compared to the traditional GCR over $\mathbb{R}$ for (1.1), i.e., (1.2), the QGCR solver can not only preserve the JRS-symmetry structures of the real counterparts, but also save three-quarters of the theoretical costs.

- Compared with QGMRES, the proposed method employs short-term recursion to save storage and computational costs. Moreover, from the minimum norm residual property, the QGCR solver with smooth convergence performs much better than that of QBiCG in terms of computing time.

The organization of this paper is as follows. In Section 2, we recall some fundamental results related to quaternions and quaternion matrices. In Section 3, we develop a structure-preserving quaternion generalized conjugate residual (QGCR) method to solve (1.1). In Section 4, we report numerical examples that arise in the color image deblurring problems and illustrate the effectiveness and robustness of QGCR compared to QBiCG. Finally, we conclude the whole paper by providing some remarks in Section 5.

2. Preliminaries

In this section, we present some fundamental results associated with quaternions and quaternion matrices. Given a quaternion $\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$, the conjugate of $\mathbf{q}$, denoted by $\mathbf{q}^*$, is of the form $\mathbf{q}^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}$. The norm of $\mathbf{q}$ is defined as $|\mathbf{q}|^2 = \mathbf{q}^*\mathbf{q} = q_0^2 + q_1^2 + q_2^2 + q_3^2$. The induced inverse of $\mathbf{q}$, defined as $\mathbf{q}^{-1} = \mathbf{q}^*/|\mathbf{q}|^2$, is unique. The conjugate transpose of a quaternion matrix $\mathbf{Q} = Q_0 + Q_1\mathbf{i} + Q_2\mathbf{j} + Q_3\mathbf{k} \in \mathbb{Q}^{n\times n}$, denoted by $\mathbf{Q}^*$, is of the form $Q_0^T - Q_1^T\mathbf{i} - Q_2^T\mathbf{j} - Q_3^T\mathbf{k}$. The quaternion matrix $\mathbf{Q}$ is said to be Hermitian or unitary if $\mathbf{Q} = \mathbf{Q}^*$ or $\mathbf{Q}\mathbf{Q}^* = \mathbf{I}_n$, respectively. Given a Hermitian quaternion matrix $\mathbf{Q} \in \mathbb{Q}^{n\times n}$, it is said to be positive semi-definite if $\mathbf{x}^*\mathbf{Q}\mathbf{x} \geq 0$ for any non-zero quaternion vector $\mathbf{x} \in \mathbb{Q}^n$. The real counterpart of a quaternion matrix $\mathbf{Q} \in \mathbb{Q}^{n\times n}$, defined by the linear homeomorphic mapping $\mathscr{R}(\cdot)$, is of the form

$$\mathscr{R}(\mathbf{Q}) = \begin{bmatrix} Q_0 & -Q_1 & -Q_2 & -Q_3 \\ Q_1 & Q_0 & -Q_3 & Q_2 \\ Q_2 & Q_3 & Q_0 & -Q_1 \\ Q_3 & -Q_2 & Q_1 & Q_0 \end{bmatrix} \in \mathbb{R}^{4n\times 4n}, \tag{2.1}$$

and its the first block column is denoted by

$$\mathscr{R}(\mathbf{Q})_c = \begin{bmatrix} Q_0^T & Q_1^T & Q_2^T & Q_3^T \end{bmatrix}^T \in \mathbb{R}^{4n\times n}.$$

Note that there are various real counterparts, but all of them are equivalent and surely JRS-symmetric matrices, see [13] and the references therein. Denote by $\mathbf{v}_1\boldsymbol{\alpha}_1 + \mathbf{v}_2\boldsymbol{\alpha}_2 + \cdots + \mathbf{v}_n\boldsymbol{\alpha}_n$, the right-hand side linear combination of quaternion vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, $\alpha_i \in \mathbb{Q}$, $i = 1, 2, \ldots, n$. In what follows, we only consider the right multiplication over $\mathbb{Q}$ since it has similar properties to the unitary complex vector space. From [6], one can see that $\mathbb{Q}^n$ is a right quaternionic Hilbert space. The following lemma shows that orthogonal JRS-symplectic matrix equivalence transformations can preserve the unique algebraic symmetry exhibited by their real counterparts.

Lemma 2.1. [12] *Let $W \in \mathbb{R}^{4n\times 4n}$ be an orthogonally JRS-symplectic matrix. If $\mathscr{R}(\mathbf{Q})$ defined in* (2.1) *is a JRS-symmetric matrix, then $W^T\mathscr{R}(\mathbf{Q})W \in \mathbb{R}^{4n\times 4n}$ so is.*

3. Quaternion generalized conjugate residual method

In this section, we will derive a structure-preserving quaternion generalized conjugate residual method (QGCR) for solving (1.1), which completely differs from QGMRES and QBiCG. Now we give a helpful lemma that will be used later.

Lemma 3.1. [12] *Let $\mathbf{Q} \in \mathbb{Q}^{n\times n}$ be a Hermitian quaternion matrix, and let $\mathscr{R}(\mathbf{Q})$ be its real counterpart. Then there exists an orthogonally JRS-symplectic matrix $\mathscr{R}(\mathbf{V}) \in \mathbb{R}^{4n\times 4n}$, i.e., $\mathbf{V} \in \mathbb{Q}^{n\times n}$ is unitary, such that*

$$\mathscr{R}(\mathbf{V})^T\mathscr{R}(\mathbf{Q})\mathscr{R}(\mathbf{V}) = \begin{bmatrix} D & 0 & 0 & 0 \\ 0 & D & 0 & 0 \\ 0 & 0 & D & 0 \\ 0 & 0 & 0 & D \end{bmatrix},$$

i.e., $\mathbf{V}^*\mathbf{Q}\mathbf{V} = D$, where $D \in \mathbb{R}^{n\times n}$ is a symmetric tridiagonal matrix.

The QGMRES, QBiCG, and the algorithms developed here are strongly related to, as well as defined by, the choice of a basis of the quaternion Krylov subspace

$$\mathcal{K}_m(\mathbf{A}, \mathbf{r}) := \text{span}\{\mathbf{r}, \mathbf{A}\mathbf{r}, \ldots, \mathbf{A}^{m-1}\mathbf{r}\}$$

in which $\mathbf{A} \in \mathbb{Q}^{n\times n}$ is non-Hermitian quaternion matrix, and $\mathbf{r} \in \mathbb{Q}^n$ is a certainly quaternion vector. As shown in [10], each vector of $\mathcal{K}_m(\mathbf{A}, \mathbf{r})$ can only be expressed as a linear combination of $\mathbf{r}, \mathbf{A}\mathbf{r}, \ldots, \mathbf{A}^{m-1}\mathbf{r}$, but cannot be rewritten to multiply a polynomial

with degree less than $m-1$ of $\mathbf{A}$ and the vector $\mathbf{r}$ over $\mathbb{Q}$. These results differ from those presented in [19]. The QGMRES algorithm generates an orthonormal basis, whereas the introduced QGCR algorithm produces a basis with being $\mathbf{A}^*\mathbf{A}$-orthogonal. The main results of the proposed algorithms are based on the following lemma.

Theorem 3.2. *Let $\mathbf{A} \in \mathbb{Q}^{n\times n}$ be a non-Hermitian quaternion matrix, and let $\mathbf{p}_0, \mathbf{p}_1, \ldots,$ $\mathbf{p}_{m-1}$ be a sequence of quaternion vectors, such that each set $\{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{m-1}\}$ for $m \leq n$ is a basis of the quaternion Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{r})$ which is $\mathbf{A}^*\mathbf{A}$-orthogonal, i.e., such that*

$$\langle \mathbf{A}\mathbf{p}_i, \mathbf{A}\mathbf{p}_k \rangle = 0$$

hold for $i \neq k$, $i, k = 0, \ldots, m-1$.

Proof. From Lemma 3.1, there exists a unitary quaternion matrix $\mathbf{V}$ such that

$$\mathbf{V}^*(\mathbf{A}^*\mathbf{A})\mathbf{V} = D$$

is an $n \times n$ real symmetric tridiagonal matrix. Moreover, it is also a positive semi-definite matrix, since $\mathbf{x}^*D\mathbf{x} = (\mathbf{A}\mathbf{V}\mathbf{x})^*(\mathbf{A}\mathbf{V}\mathbf{x}) \geq 0$ holds for any nonzero vector $\mathbf{x} \in \mathbb{Q}^n$. Using the standard Cholesky decomposition [19], the matrix D can be decomposed into the following form

$$D = L^T\widetilde{D}L,$$

where $L \in \mathbb{R}^{n\times n}$ is a lower triangular matrix, and $\widetilde{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix.

Define $\mathbf{P} := [\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{m-1}] = (\mathbf{V}L^{-1})$, it follows that

$$\mathbf{P}^*\mathbf{A}^*\mathbf{A}\mathbf{P} = (\mathbf{V}L^{-1})^*(\mathbf{A}^*\mathbf{A})(\mathbf{V}L^{-1}) = L^{-T}L^T\widetilde{D}LL^{-1} = \widetilde{D},$$

that is,

$$\mathscr{R}(\mathbf{P})^T\mathscr{R}(\mathbf{A}^*\mathbf{A})\mathscr{R}(\mathbf{P}) = \begin{bmatrix} \widetilde{D} & 0 & 0 & 0 \\ 0 & \widetilde{D} & 0 & 0 \\ 0 & 0 & \widetilde{D} & 0 \\ 0 & 0 & 0 & \widetilde{D} \end{bmatrix}. \tag{3.1}$$

□

This fact shows that a basis in $\mathcal{K}_m(\mathbf{A}, \mathbf{r})$ is surely $\mathbf{A}^*\mathbf{A}$-orthogonal. We say that the decomposition in (3.1) is structure-preserving since the JRS-symmetry of $\mathscr{R}(\mathbf{A}^*\mathbf{A})$ is preserved.

In what follows, we develop the quaternion generalized conjugate residual (QGCR) method, which yields a minimization property of the residual norm, for solving the original linear systems (1.1). It is also mathematically equivalent to QGMRES. The QGCR method

belonging to the quaternion Krylov subspace methods seeks an approximate solution $\mathbf{x}_m$ from an affine subspace $\mathbf{x}_0+\mathcal{K}_m(\mathbf{A},\mathbf{r}_0)$ by imposing the following orthogonality conditions

$$\mathbf{b}-\mathbf{A}\mathbf{x}_m \perp \mathbf{A}\mathcal{K}_m(\mathbf{A},\mathbf{r}_0), \tag{3.2}$$

where $\mathbf{A}\mathcal{K}_m(\mathbf{A},\mathbf{r}_0)$ is a constrained subspace. Here, $\mathbf{x}_0$ stands for an initial guess of (1.1), and the quaternion Krylov subspace $\mathcal{K}_m(\mathbf{A},\mathbf{r}_0)$ is of the form

$$\mathcal{K}_m(\mathbf{A},\mathbf{r}_0)=\mathrm{span}\{\mathbf{r}_0,\mathbf{A}\mathbf{r}_0,\ldots,\mathbf{A}^{m-1}\mathbf{r}_0\},$$

where $\mathbf{r}_0=\mathbf{b}-\mathbf{A}\mathbf{x}_0$. Observe from Theorem 3.2 that an approximate solution $\mathbf{x}_m$ can be expressed as

$$\mathbf{x}_m=\mathbf{x}_0+\sum_{i=0}^{m-1}\mathbf{p}_i\boldsymbol{\alpha}_i, \tag{3.3}$$

where $\mathbf{p}_0,\mathbf{p}_1,\ldots,\mathbf{p}_{m-1}$ is a basis of $\mathcal{K}_m(\mathbf{A},\mathbf{r}_0)$, $\boldsymbol{\alpha}_0,\ldots,\boldsymbol{\alpha}_{m-1}$ are quaternion scalars, and its associated residual vector is of the form

$$\mathbf{r}_m=\mathbf{b}-\mathbf{A}\mathbf{x}_m=\mathbf{r}_0-\sum_{i=0}^{m-1}\mathbf{A}\mathbf{p}_i\boldsymbol{\alpha}_i. \tag{3.4}$$

One can see that the core work for (3.3) is how to determine the quaternion vectors $\mathbf{p}_i$ and the parameters $\boldsymbol{\alpha}_i$. We first present the details for computing $\boldsymbol{\alpha}_i$ as follows. Recall that the orthogonality conditions (3.2) must be enforced, it follows that

$$\langle\mathbf{r}_m,\mathbf{A}\mathbf{p}_j\rangle=0,\quad 0\le j\le m-1. \tag{3.5}$$

Substitute (3.4) into the above-mentioned relation, we have

$$0=\langle\mathbf{r}_m,\mathbf{A}\mathbf{p}_j\rangle=\left\langle\mathbf{r}_0-\sum_{i=0}^{m-1}\mathbf{A}\mathbf{p}_i\boldsymbol{\alpha}_i,\mathbf{A}\mathbf{p}_j\right\rangle=\langle\mathbf{r}_0,\mathbf{A}\mathbf{p}_j\rangle-\sum_{i=0}^{m-1}\langle\mathbf{A}\mathbf{p}_i,\mathbf{A}\mathbf{p}_j\rangle\boldsymbol{\alpha}_i,$$

which implies $\boldsymbol{\alpha}_j=\langle\mathbf{A}\mathbf{p}_j,\mathbf{A}\mathbf{p}_j\rangle^{-1}\langle\mathbf{r}_0,\mathbf{A}\mathbf{p}_j\rangle$.

From the above results, it follows that $\mathbf{x}_j$ can be rewritten as

$$\mathbf{x}_j=\mathbf{x}_{j-1}+\mathbf{p}_{j-1}\boldsymbol{\alpha}_{j-1},$$

since $\boldsymbol{\alpha}_{j-1}$ and the remaining parameters $\boldsymbol{\alpha}_0,\ldots,\boldsymbol{\alpha}_{j-2}$ are independent, and the corresponding residual vector is of the form

$$\mathbf{r}_j=\mathbf{r}_{j-1}-\mathbf{A}\mathbf{p}_{j-1}\boldsymbol{\alpha}_{j-1}.$$

Again, using (3.4), it follows that

$$\langle\mathbf{r}_j,\mathbf{A}\mathbf{p}_j\rangle=\left\langle\mathbf{r}_0-\sum_{i=0}^{j-1}\mathbf{A}\mathbf{p}_i\boldsymbol{\alpha}_i,\mathbf{A}\mathbf{p}_j\right\rangle=\langle\mathbf{r}_0,\mathbf{A}\mathbf{p}_j\rangle-\sum_{i=0}^{j-1}\langle\mathbf{A}\mathbf{p}_i,\mathbf{A}\mathbf{p}_j\rangle\boldsymbol{\alpha}_i=\langle\mathbf{r}_0,\mathbf{A}\mathbf{p}_j\rangle.$$

Consequently, the quaternion scalar $\boldsymbol{\alpha}_j$ is of the form

$$\boldsymbol{\alpha}_j = \langle \mathbf{A}\mathbf{p}_j, \mathbf{A}\mathbf{p}_j \rangle^{-1} \langle \mathbf{r}_j, \mathbf{A}\mathbf{p}_j \rangle.$$

Let $\mathbf{p}_0 = \mathbf{r}_0$. Similar to the quaternion Arnoldi procedure [10], the quaternion vectors $\mathbf{A}\mathbf{p}_j$, $j = 0, \ldots, m-1$, by applying the Gram–Schmidt orthogonalization process on the set $\{\mathbf{A}\mathbf{r}_0, \mathbf{A}\mathbf{r}_1, \ldots, \mathbf{A}\mathbf{r}_{m-1}\}$, are given by

$$\mathbf{A}\mathbf{p}_j = \mathbf{A}\mathbf{r}_j - \sum_{i=0}^{j-1} \mathbf{A}\mathbf{p}_i \langle \mathbf{A}\mathbf{p}_i, \mathbf{A}\mathbf{p}_i \rangle^{-1} \langle \mathbf{A}\mathbf{r}_j, \mathbf{A}\mathbf{p}_i \rangle. \tag{3.6}$$

Multiplying the inverse of $\mathbf{A}$ on both sides of (3.6), it follows that

$$\mathbf{p}_j = \mathbf{r}_j - \sum_{i=0}^{j-1} \mathbf{p}_i \boldsymbol{\beta}_{i,j-1},$$

where $\boldsymbol{\beta}_{i,j-1} = \langle \mathbf{A}\mathbf{p}_i, \mathbf{A}\mathbf{p}_i \rangle^{-1} \langle \mathbf{A}\mathbf{r}_j, \mathbf{A}\mathbf{p}_i \rangle$, $j = 1, \ldots, m-1$.

These quaternion vectors, which is $\mathbf{A}^*\mathbf{A}$-orthogonal, form an basis of $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$. Putting these relations together gives the following algorithm.

Algorithm 3.1 Quaternion generalized conjugate residual method

1: Given an initial guess $\mathbf{x}_0 \in \mathbb{Q}^n$, we compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, and set $\mathbf{p}_0 = \mathbf{r}_0$.
2: **for** $j = 0, 1, 2, \ldots$ until convergence **do**
3: $\quad \boldsymbol{\alpha}_j = \langle \mathbf{A}\mathbf{p}_j, \mathbf{A}\mathbf{p}_j \rangle^{-1} \langle \mathbf{r}_j, \mathbf{A}\mathbf{p}_j \rangle$
4: $\quad \mathbf{x}_{j+1} = \mathbf{x}_j + \mathbf{p}_j \boldsymbol{\alpha}_j$
5: $\quad \mathbf{r}_{j+1} = \mathbf{r}_j - \mathbf{A}\mathbf{p}_j \boldsymbol{\alpha}_j$
6: $\quad$ **for** $i = 0, 1, \ldots, j$ **do**
7: $\quad\quad \boldsymbol{\beta}_{ij} = -\langle \mathbf{A}\mathbf{p}_i, \mathbf{A}\mathbf{p}_i \rangle^{-1} \langle \mathbf{A}\mathbf{r}_{j+1}, \mathbf{A}\mathbf{p}_i \rangle$
8: $\quad$ **end for**
9: $\quad \mathbf{p}_{j+1} = \mathbf{r}_{j+1} + \sum_{i=0}^{j} \mathbf{p}_i \boldsymbol{\beta}_{ij}$
10: **end for**

Clearly, the key computational work in Algorithm 3.1 is to compute $\mathbf{A}\mathbf{p}_j$, $\mathbf{A}\mathbf{r}_{j+1}$, $\langle \mathbf{A}\mathbf{p}_j, \mathbf{A}\mathbf{p}_j \rangle$ and $\langle \mathbf{r}_j, \mathbf{A}\mathbf{p}_j \rangle$. From (2.1), the above quaternion matrix-vector multiplication and inner product are implemented by

$$\mathbf{A}\mathbf{p}_j = \mathcal{R}^{-1}(\mathcal{R}(\mathbf{A})\mathcal{R}(\mathbf{p}_j)), \quad \mathbf{A}\mathbf{r}_{j+1} = \mathcal{R}^{-1}(\mathcal{R}(\mathbf{A})\mathcal{R}(\mathbf{r}_{j+1})),$$
$$\langle \mathbf{A}\mathbf{p}_j, \mathbf{A}\mathbf{p}_j \rangle = \mathcal{R}^{-1}(\mathcal{R}((\mathbf{A}\mathbf{p}_j)^*)\mathcal{R}(\mathbf{A}\mathbf{p}_j)), \quad \langle \mathbf{r}_j, \mathbf{A}\mathbf{p}_j \rangle = \mathcal{R}^{-1}(\mathcal{R}((\mathbf{A}\mathbf{p}_j)^*)\mathcal{R}(\mathbf{r}_j)).$$

Indeed, we can save three-quarters of the computational workload by exclusively computing and storing the first block columns of their real counterparts, that is,

$$\mathcal{R}(\mathbf{A}\mathbf{p}_j)_c = \mathcal{R}(\mathbf{A})\mathcal{R}(\mathbf{p}_j)_c, \quad \mathcal{R}(\mathbf{A}\mathbf{r}_{j+1})_c = \mathcal{R}(\mathbf{A})\mathcal{R}(\mathbf{r}_{j+1})_c,$$
$$\langle \mathbf{A}\mathbf{p}_j, \mathbf{A}\mathbf{p}_j \rangle = \mathcal{R}(\mathbf{A}\mathbf{p}_j)^T \mathcal{R}(\mathbf{A}\mathbf{p}_j)_c, \quad \langle \mathbf{r}_j, \mathbf{A}\mathbf{p}_j \rangle = \mathcal{R}(\mathbf{A}\mathbf{p}_j)^T \mathcal{R}(\mathbf{r}_j)_c.$$

From the above-mentioned results and Algorithm 3.1, an important proposition associated with the residual vectors $\mathbf{r}_j$ is as follows.

Proposition 3.3. *Let* $\mathbf{A}$ *be a non-Hermitian quaternion matrix. Then the residual vectors* $\mathbf{r}_m$ *and* $\mathbf{r}_j$*, (*$j < m$*,* $j = 0, 1, \ldots, m-1$*), generated by Algorithm* 3.1 *are* $\mathbf{A}$*-orthogonal, i.e.,*

$$\langle \mathbf{r}_m, \mathbf{A}\mathbf{r}_j \rangle = 0.$$

Proof. It follows from (3.6) and (3.5) that

$$\langle \mathbf{r}_m, \mathbf{A}\mathbf{r}_j \rangle = \left\langle \mathbf{r}_m, \mathbf{A}\mathbf{p}_j + \sum_{i=0}^{j-1} \mathbf{A}\mathbf{p}_i \boldsymbol{\beta}_{i,j-1} \right\rangle = 0,$$

which completes the proof. □

It is well known that much can be gained by using the modified Gram–Schmidt method with short-term recurrence instead of the standard Gram–Schmidt method with long-term recurrence in practice. One of the most important advantages is that the former performs much more reliably than the latter. From Algorithm 3.1, we can see that $\mathbf{p}_{j+1}$ is generated by $\mathbf{r}_{j+1}$ and $\mathbf{p}_i$ with long-term recurrence. This may result that the orthogonality of the computed vectors suffers from the breakdown in practical computations. Therefore, similar to the modified Gram–Schmidt method, we set $\mathbf{w}_{j+1} = \mathbf{A}\mathbf{r}_{j+1}$, and then take

$$\begin{aligned} \boldsymbol{\beta}_{ij} &= -\langle \mathbf{A}\mathbf{p}_i, \mathbf{A}\mathbf{p}_i \rangle^{-1} \langle \mathbf{w}_{j+1}, \mathbf{A}\mathbf{p}_i \rangle, \\ \mathbf{w}_{j+1} &= \mathbf{w}_{j+1} + \mathbf{A}\mathbf{p}_i \boldsymbol{\beta}_{ij}, \quad i = 0, 1, \ldots, j, \\ \mathbf{A}\mathbf{p}_{j+1} &= \mathbf{w}_{j+1} \end{aligned}$$

instead of the lines 6–9 of Algorithm 3.1. In exact arithmetic, the above formulation and the mentioned lines are mathematically equivalent, where the former is considerably more reliable than the latter in the presence of round-off errors.

The QGCR method is completely different from QBiCG [15]. For the latter, it is based on the quaternion Lanczos process and the Galerkin condition, rather than by minimizing the residual norm. This means that the QBiCG method exhibits a rather irregular convergence behavior with wild oscillations in the residual norm, which may cause breakdowns in some cases. In contrast, QGCR employs an oblique projection quaternion Krylov subspace method, focusing on minimizing the residual norm, which performs much more reliably than QBiCG.

4. Numerical experiments

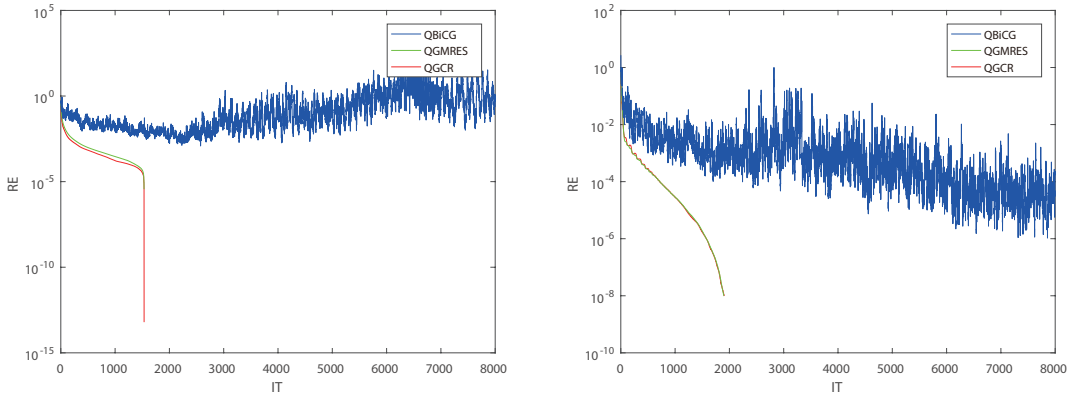
In this section, we record some results when applying QGCR solver for solving the quaternion linear systems (1.1) with some practical applications from Matrix Market and color

image deblurring problems, and then illustrate the efficiency and accuracy of the proposed method compared with QBiCG and QGMRES. All algorithms were run on a personal computer in Matlab 2019b with AMD Ryzen 7-7735H 3.20GHz and 32.00 GB memory. The stopping criterion here is set to be the relative residual error $\mathrm{RE} = \|\mathbf{b}-\mathbf{A}\mathbf{x}_j\|_2/\|\mathbf{r}_0\|_2 \leq \varepsilon$, and the initial guess is chosen to be $\mathbf{x}_0 = \mathbf{0}$. Denote by **CPU**, the elapsed computing time in seconds, and by **IT** the number of iteration steps. The symbol † represents that the relative residual can not reach the tolerance before the 8000 iterations.

Example 4.1. In this example, we will test the performance of the above methods for solving (1.1) with some practical applications. Herein, the coefficient matrix $\mathbf{A}$ and right-hand side vector $\mathbf{b}$ are given by

$$\mathbf{A} = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k},$$
$$\mathbf{b} = b_0 + b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k},$$

where $b_i = \mathrm{rand}(n, 1)$, $i = 0, 1, 2, 3$, is an $n \times 1$ random vector with entries uniformly distributed in $[0, 1]$, $A_1 = 1.5 * A_0$, $A_2 = 2 * A_0$, $A_3 = 0.5 * A_0$, and A_0 from the "MatrixMarket"[1] collection is set as **bcsstk34** ($n = 588$), **jagmesh7** ($n = 1138$), **bcspwr09** ($n = 1723$), and **add20** ($n = 2395$), respectively. The parameter ε is set to 10^{-8}. The symbol † represents that the tested methods fail to reach the desired tolerance before the maximum iterations.



(a) **bcspwr09** (b) **add20**

Figure 4.1: Convergence curves for QBiCG, QGMERS and QGCR with **bcspwr09** (a), and **add20** (b).

We report the numerical results generated by QBiCG, QGMRES, and QGCR in Table 4.1. The best results of tested algorithms are marked by boldface regarding computing

[1]The MatrixMarket is available on https://math.nist.gov/MatrixMarket/.

time and iterations. From Table 4.1, it is evident that QBiCG fails to converge when solving these problems, while QGMRES and QGCR exhibit similar fast convergence behavior. Convergence curves of the QBiCG, QGMRES, and QGCR solvers are depicted in Figure 4.1. This figure shows that the relative residual error of QGCR and QGMRES is commonly less than that of QBiCG at each iteration. Moreover, the convergence behavior of QBiCG seems considerably erratic, whereas the convergence curves of QGCR and QGMRES are much smoother than QBiCG. Indeed, both QGCR and QGMRES are quaternion Krylov subspace methods based on oblique projection with minimizing residual norms, such that their convergence behavior is similar. Additionally, the elapsed computing time corresponding to QGCR is slightly less than that of QGMRES. This shows that the QGCR solver converges faster than the latter.

Table 4.1: Numerical results of Example 4.1.

Data set	N	Algorithms	CPU	RE	IT	Application discipline
bcsstk34	588	QBiCG	†	5.94e+04	8000	Structural Problem
		QGMRES	4.50	9.09e-09	514	
		QGCR	**3.93**	8.98e-09	516	
jagmesh7	1138	QBiCG	†	2.19e-01	8000	2D/3D Problem
		QGMRES	27.43	1.00e-15	1137	
		QGCR	**23.77**	3.84e-16	1138	
bcspwr09	1723	QBiCG	†	9.02e-01	8000	Power Network Problem
		QGMRES	61.66	6.63e-14	1533	
		QGCR	**53.79**	6.18e-14	1534	
add20	2395	QBiCG	†	2.26e-04	8000	Circuit Simulation Problem
		QGMRES	164.78	9.39e-09	1901	
		QGCR	**153.04**	9.82e-09	1905	

Example 4.2. A color image of size $n \times n$, in JPG format, can be represented as a pure quaternion vector $\mathbf{x} = x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \in \mathbb{Q}^{n^2}$, where x_1, x_2 and $x_3 \in \mathbb{R}^{n^2}$ denote the red, green, and blue channels, respectively. Assume that there exists a matrix $\mathbf{A} = A_0 \in \mathbb{R}^{n^2\times n^2}$, which models single channel blurring problems, acting on color images ("**Couch**" and "**Branch**") of size 128×128, such that the observed color image is $\mathbf{b}$. Then we implement the proposed method and QBiCG to restore the contaminated color image, i.e., solving (1.1). Here, we take $A_0 = B_1 \otimes B_2$ with $B_1 = [b_{ij}^{(1)}]$ and $B_2 = [b_{ij}^{(2)}]$ being the Toeplitz matrices given by [1]

$$b_{ij}^{(1)} = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp\big(-\frac{(i-j)^2}{2\sigma^2}\big), & |i-j| \le r, \\ 0, & \text{otherwise}, \end{cases} \quad b_{ij}^{(2)} = \begin{cases} \frac{1}{2s-1}, & |i-j| \le s, \\ 0, & \text{otherwise}. \end{cases}$$

The parameters are $\sigma = 1$, $r = 4$, and $s = 7$. The numerical results with QGCR and QBiCG, including the computing time, relative residual error, peak signal-to-noise ratio (PSNR) [3] in decibels and structure similarity index measure (SSIM) [20], are recorded in Table 4.2. As seen, the proposed method restores the color images with the same quality as QBiCG solver in terms of PSNR and SSIM, whereas it requires much less CPU time and iterations than the latter. Moreover, we display the convergence curves of the two solvers in Figures 4.2 and 4.3. From the figures, we observe that the convergence behavior of QBiCG seems considerably erratic, whereas the convergence curve of QGCR performs much smoother than that of QBiCG. These show the effectiveness and robustness of our method in comparison with QBiCG.

Table 4.2: Numerical results with single channel deblurring.

Color images	**Algorithms**	**CPU**	**PSNR**	**SSIM**	**RE**	**IT**
Couch	QBiCG	5.9073e+03	36.5	0.98	1.59e-04	1000
	QGCR	**4.7183e+03**	36.5	0.98	**9.99e-06**	**765**
Branch	QBiCG	5.4071e+03	34.3	0.96	2.63e-04	1000
	QGCR	**4.0383e+03**	34.3	0.96	**9.98e-06**	**637**

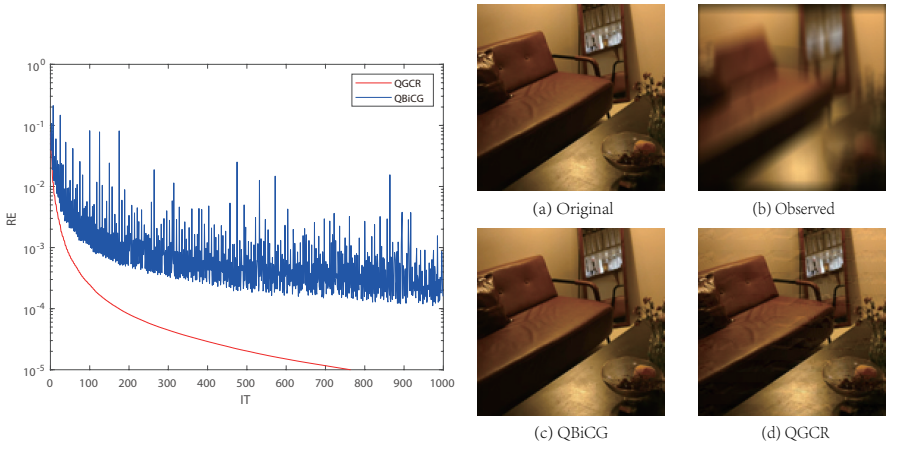


Figure 4.2: Convergence curves (left), and the deblurred color image ("**Couch**") by QBiCG and QGCR (right).

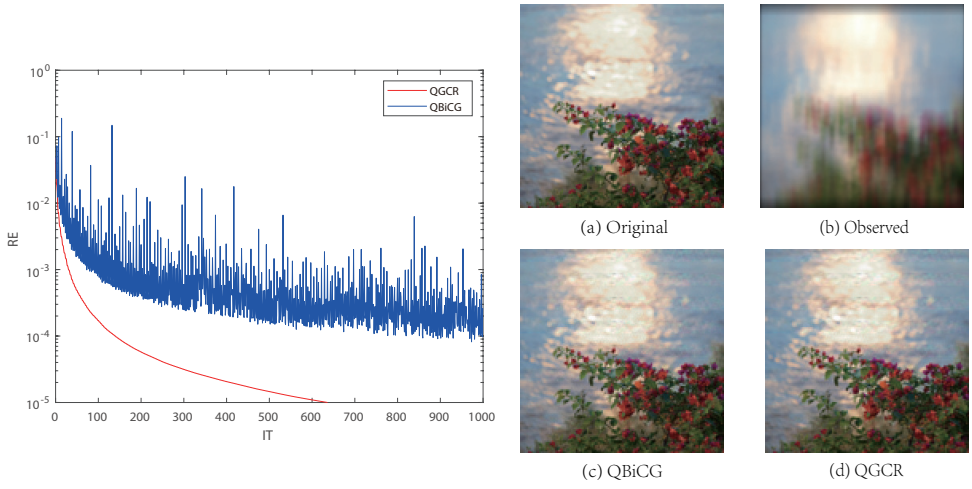


Figure 4.3: Convergence curves (left), and the deblurred color image ("**Branch**") by QBiCG and QGCR (right).

5. Conclusion

In this paper, we proposed a structure-preserving quaternion GCR (QGCR) method for solving general quaternion linear systems, which applies to some practical data and color image deblurring problems. Compared with traditional GCR iteration, our method saves about three-quarters of the theoretical costs and storage. It also converges much faster and smoother than QBiCG when solving the color image deblurring problems. Furthermore, the QGCR solver requires much less CPU time than QGMRES.

References

[1] A. Bouhamidi and K. Jbilou, *Sylvester Tikhonov-regularization methods in image restoration*, J. Comput. Appl. Math. **206** (2007), no. 1, 86–98.

[2] J. Cariño, H. Abaunza and P. Castillo, *Quadrotor quaternion control*, in: *2015 International Conference on Unmanned Aircraft Systems (Denver, Colorado)*, 825–831, IEEE, 2015.

[3] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans and A. C. Bovik, *Image quality assessment based on a degradation model*, IEEE Trans. Image Process. **9** (2000), no. 4, 636–650.

[4] S. De Leo and G. Scolarici, *Right eigenvalue equation in quaternionic quantum mechanics*, J. Phys. A **33** (2000), no. 15, 2971–2995.

[5] D. Finkelstein, J. M. Jauch, S. Schiminovich and D. Speiser, *Foundations of quaternion quantum mechanics*, J. Mathematical Phys. **3** (1962), 207–220.

[6] R. Ghiloni, V. Moretti and A. Perotti, *Continuous slice functional calculus in quaternionic Hilbert spaces*, Rev. Math. Phys. **25** (2013), no. 4, 1350006, 83 pp.

[7] W. R. Hamilton, *Elements of Quaternions*, Longmans, Green and Company, 1866.

[8] Z.-H. He, Q.-W. Wang and Y. Zhang, *A system of quaternary coupled Sylvester-type real quaternion matrix equations*, Automatica J. IFAC **87** (2018), 25–31.

[9] Z.-H. He, X.-X. Wang and Y.-F. Zhao, *Eigenvalues of quaternion tensors with applications to color video processing*, J. Sci. Comput. **94** (2023), no. 1, Paper No. 1, 15 pp.

[10] Z. Jia and M. K. Ng, *Structure preserving quaternion generalized minimal residual method*, SIAM J. Matrix Anal. Appl. **42** (2021), no. 2, 616–634.

[11] Z. Jia, M. K. Ng and W. Wang, *Color image restoration by saturation-value total variation*, SIAM J. Imaging Sci. **12** (2019), no. 2, 972–1000.

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

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

[14] T. Li and Q.-W. Wang, *Structure preserving quaternion full orthogonalization method with applications*, Numer. Linear Algebra Appl. **30** (2023), no. 5, Paper No. e2495, 15 pp.

[15] ———, *Structure preserving quaternion biconjugate gradient method*, SIAM J. Matrix Anal. Appl. **45** (2024), no. 1, 306–326.

[16] T. Parcollet, M. Morchid and G. Linarès, *A survey of quaternion neural networks*, Artif. Intell. Rev. **53** (2020), 2957–2982.

[17] A. Perez and J.-M. McCarthy, *Dual quaternion synthesis of constrained robotic systems*, J. Mech. Des. **126** (2004), no. 3, 425–435.

[18] D. Pletinckx, *Quaternion calculus as a basic tool in computer graphics*, Visual Comput. **5** (1989), 2–13.

[19] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[20] Z. Wang, A. C. Bovik and H. R. Sheikh and E. P. Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE Trans. Image Process. **13** (2004), no. 4, 600–612.

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

Xin-Fang Zhang, Wei Ding and Tao Li
School of Mathematics and Statistics, Key Laboratory of Engineering Modeling and Statistical Computation of Hainan Province, Hainan University, Haikou 570228, China
E-mail addresses: 995272@hainanu.edu.cn, d-bai@foxmail.com, tli@hainanu.edu.cn