Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 35(1): 45–68.
DOI: 10.3934/era.2027002
Received: 21 September 2026
Revised: 02 November 2026
Accepted: 10 November 2026
Published: 27 November 2026

Research article

Global bidiagonalization algorithms for quaternion linear systems with multiple right-hand sides

Jingwen Liu, Shuyang Chen and Hongxiu Zhong*

School of Mathematics and Data Science, Jiangnan University, Jiangsu 214122, China

* **Correspondence:** Email: zhonghongxiu@126.com.

Abstract: In this paper, three structure-preserving global bidiagonalization algorithms are presented for solving large-scale quaternion linear systems with multiple right-hand sides, namely the global quaternion least-squares QR algorithm (Gl-QLSQR), the global quaternion residual-reducing least-squares QR algorithm (Gl-QRRLSQR), and the global quaternion oblique projection algorithm (Gl-QLSOR). All three algorithms are based on the Golub–Kahan–Lanczos bidiagonalization theory, by which the original quaternion matrix is reduced to lower bidiagonal form, and a concise recursive formula is derived to generate the sequence of approximate solutions $\{X_k\}$. All matrix–matrix products and inner products are carried out on the first block column of the real representation, so that the full real counterpart is never formed, which theoretically reduces the arithmetic cost by about three quarters. We also present the convergence analysis of the proposed methods and show the applicability of all three methods to the quaternion Sylvester matrix equation. Numerical experiments on quaternion least-squares problems and color image deblurring problems validate the feasibility and effectiveness of the proposed algorithms.

Keywords: bidiagonalization; global quaternion LSQR method; multiple right-hand sides; quaternion linear systems; color image deblurring

1. Introduction

Quaternions, first introduced by W. R. Hamilton in the 19th century [1], have evolved into a fundamental mathematical tool with the development of quaternion matrix theory [2]. This algebraic structure has found extensive applications in computer science, statistics, signal and color image processing [3], and quantum mechanics [4], among other disciplines.

Denote by $\mathbb{R}$ the real number field, $\mathbb{H}^m$ and $\mathbb{H}^{m\times n}$ the collections of all m-dimensional vectors, and all $m \times n$ matrices over the quaternion skew-field. A quaternion $a \in \mathbb{H}$ can be written as $a = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}$ [1], where $a_0, a_1, a_2, a_3 \in \mathbb{R}$ are real coefficients and the imaginary units $\mathbf{i}, \mathbf{j}, \mathbf{k}$

satisfy the multiplication rules $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. Observe that $\mathbb{H}$ constitutes a four-dimensional associative algebra over $\mathbb{R}$ with non-commutative multiplication. The real representation of a quaternion matrix $W = W_0 + W_1\mathbf{i} + W_2\mathbf{j} + W_3\mathbf{k} \in \mathbb{H}^{m\times n}$, where $W_0, W_1, W_2, W_3 \in \mathbb{R}^{m\times n}$, determined via the linear homomorphism $\mathcal{R}(\cdot)$, is given by

$$\mathcal{R}(W) := \begin{bmatrix} W_0 & -W_1 & -W_2 & -W_3 \\ W_1 & W_0 & -W_3 & W_2 \\ W_2 & W_3 & W_0 & -W_1 \\ W_3 & -W_2 & W_1 & W_0 \end{bmatrix} \in \mathbb{R}^{4m\times 4n} \tag{1.1}$$

and the form of the first block column

$$\mathcal{R}(W)_c := \left[W_0^T\ , W_1^T\ , W_2^T\ , W_3^T\right]^T \in \mathbb{R}^{4m\times n}.$$

Throughout this paper, $\mathbb{R}$ denotes the real number field, $\mathcal{R}(\cdot)$ the real representation, and the italic letter R (with subscripts) the residual matrix.

In this paper, we consider the quaternion linear system with multiple right-hand sides as follows:

$$AX = B, \tag{1.2}$$

where the matrix $A \in \mathbb{H}^{m\times n}$, $B = [B_{:,1}, B_{:,2}, \cdots, B_{:,s}] \in \mathbb{H}^{m\times s}$, and $X = [X_{:,1}, X_{:,2}, \cdots, X_{:,s}] \in \mathbb{H}^{n\times s}$ are rectangular quaternion matrices with $s \ll m, n$. When $A \in \mathbb{H}^{n\times n}$ is invertible, Eq (1.2) admits a unique solution. Nevertheless, the system (1.2) is not always consistent in practical applications, so that no exact solution exists. In this case, it is of practical interest to seek an approximate solution $X \in \mathbb{H}^{n\times s}$ such that the residual $\|AX - B\|_F$ is minimized. This gives rise to the following quaternion matrix least-squares problem:

$$\min_{X\in\mathbb{H}^{n\times s}} \|AX - B\|_F, \tag{1.3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $A \in \mathbb{H}^{m\times n}$ is assumed to have full column rank. Equivalently, (1.3) can be decomposed column-wise into s independent subproblems $\min_{x^{(i)}\in\mathbb{H}^n} \|Ax^{(i)} - b^{(i)}\|, i = 1, 2, \cdots, s$, where $\|\cdot\|$ denotes the 2-norm.

The study of Eq (1.2) has both theoretical and applied significance in statistics [5], quantum physics [6], tensor analysis [7], color image processing [8, 9], and neural networks [10, 11]. There are two principal approaches to solving problem (1.2): performing quaternion arithmetic directly, which is hampered by the non-commutativity of quaternion multiplication, or converting the problem into a real linear system via the real matrix representation [4] and solving it within the real number field.

This paper is mainly based on the real representation of quaternions. By the linear homomorphism $\mathcal{R}(\cdot)$, Eq (1.2) can be rewritten equivalently as the real matrix equation $\mathcal{R}(A)\mathcal{R}(X) = \mathcal{R}(B) \in \mathbb{R}^{4m\times 4s}$ and can be solved by block or global Krylov subspace methods over $\mathbb{R}$, such as the block Lanczos method [12], the block generalized minimal residual method [13], the block quasi-minimal residual method [14], as well as the global full orthogonalization method (Gl-FOM) and the global generalized minimal residual method (Gl-GMRES) [15]. However, the upper Hessenberg matrix generated by these algorithms does not inherit the JRS-symmetry of $\mathcal{R}(A)$, and the arithmetic cost becomes prohibitively large as the dimension increases. To overcome these limitations, structure-preserving methods that operate directly in the quaternion domain have been developed: the quaternion

generalized minimal residual method (QGMRES) [16] and the quaternion full orthogonalization method (QFOM) [17] for the single right-hand side case ($s = 1$), and the global QFOM (Gl-QFOM) and the global QGMRES (Gl-QGMRES) methods [18] for $s > 1$, all of which preserve the JRS-symmetric structure and reduce the theoretical per-iteration arithmetic cost to about one quarter of that of their real counterparts. Nevertheless, these methods are restricted to square invertible A and are not directly applicable to the least-squares problem (1.3). For this problem, the Golub–Kahan–Lanczos bidiagonalization process provides an effective framework. Wang [19] proposed the quaternion least-squares QR algorithm (QLSQR) for $\min \|b - Ax\|$ with a single right-hand side, and Ling and Jia [20] developed the QLSQR and quaternion residual-reducing least-squares QR algorithm (QRRLSQR) for the generalized least-squares problem $\min \|ABX - E\|_F$ arising in quaternion quantum theory. In addition, Toutounian and Karimi [21] proposed the global least-squares QR algorithm (Gl-LSQR) for the real case. Unlike applying QLSQR [19] to each right-hand side separately, which requires s independent bidiagonalization processes, the proposed global methods treat all right-hand sides simultaneously within a single bidiagonalization process based on matrix–matrix products. Compared with the algorithms of Ling and Jia [20], which were designed for the specific problem $\min \|ABX - E\|_F$, our methods address the general problem (1.3) with multiple right-hand sides and further include the Gl-QLSOR algorithm. The global approach is preferable when the right-hand sides share the same coefficient matrix and all solutions are required simultaneously, as in color image deblurring.

Motivated by the above works, this paper aims to integrate the structure-preserving and global strategies of Gl-QFOM and Gl-QGMRES with the Golub–Kahan–Lanczos bidiagonalization process, and to extend the framework to the quaternion matrix least-squares problem (1.3) with multiple right-hand sides. The main contributions of this paper are summarized as follows.

1) New algorithms for rectangular matrices: Three global bidiagonalization algorithms, Gl-QLSQR, Gl-QRRLSQR and Gl-QLSOR, are proposed for problem (1.3). Unlike Gl-QFOM and Gl-QGMRES, which require a square invertible coefficient matrix, they apply to rectangular matrices of full column rank.

2) Real bidiagonal structure: The global Golub–Kahan–Lanczos process reduces A to a real bidiagonal matrix, which avoids the intricate quaternion Hessenberg reductions and yields a concise recursion for the iterates $\{X_k\}$.

3) Low arithmetic cost: Only the first block column of the real representation is used for matrix–matrix products and inner products, so the full real counterpart is never formed, and the theoretical cost is about three quarters lower than that of the real-valued global counterparts.

4) Convergence and extension: A convergence analysis is established, and all three algorithms are extended to the quaternion Sylvester equation $AX + XB = C$. Numerical experiments on least-squares problems and color image deblurring confirm their efficiency.

The remainder of this paper is organized as follows. Section 2 briefly reviews preliminary results on quaternion matrices. Section 3 develops structure-preserving global quaternion algorithms Gl-QLSQR and Gl-QRRLSQR based on the quaternion matrix Krylov subspace for solving Eq (1.3). Section 4 introduces the structure-preserving global quaternion oblique projection algorithm Gl-QLSOR, derives a computable expression of its residual, and extends all three algorithms to the

quaternion Sylvester matrix equation. Section 5 presents numerical experiments, including quaternion least-squares problems and color image deblurring, to demonstrate the efficiency of the proposed methods. Section 6 concludes the paper.

2. Preliminaries

In this section, we introduce some basic concepts and notations of quaternion matrices, as well as the real structure-preserving method for quaternion matrices. Given a quaternion $q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$, its conjugate is defined as $q^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}$. The modulus of q is given by $|q| = \sqrt{q^*q} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$. Any nonzero quaternion q possesses a unique inverse, expressed as $q^{-1} = q^*/|q|^2$. The inner product between two quaternion matrices $X, Y \in \mathbb{H}^{m\times n}$ is defined as $\langle X, Y\rangle = \mathrm{tr}(Y^*X)$, yielding a quaternion scalar, where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix. Two matrices are said to be orthogonal if their inner product is zero. The induced norm considered in this paper is the Frobenius norm. For a quaternion matrix $A = (a_{ij}) \in \mathbb{H}^{m\times n}$, the Frobenius norm is defined as

$$\|A\|_F = \sqrt{\mathrm{tr}(A^*A)} = \sqrt{\sum_{i,j} |a_{ij}|^2},$$

where A^* denotes the conjugate transpose of A.

Let $V_i \in \mathbb{H}^{m\times n}$ for $i = 1, \cdots, k$. We define the block quaternion matrix

$$\mathcal{V}_k := [V_1, V_2, \cdots, V_k] \in \mathbb{H}^{m\times nk},$$

which concatenates the matrices V_i horizontally. For a quaternion vector $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_k)^T \in \mathbb{H}^k$, we introduce the global product $\circledast$ between $\mathcal{V}_k$ and α as follows:

$$\mathcal{V}_k \circledast \alpha := \sum_{i=1}^{k} V_i\alpha_i,$$

where $V_i\alpha_i$ denotes the right scalar multiplication of the quaternion matrix V_i by the quaternion scalar α_i, i.e., each element of V_i is multiplied on the right by α_i.

Similarly, for a quaternion matrix $W \in \mathbb{H}^{k\times k}$, we define the matrix–matrix global product

$$\mathcal{V}_k \circledast W := [\mathcal{V}_k \circledast W_{:,1}, \mathcal{V}_k \circledast W_{:,2}, \cdots, \mathcal{V}_k \circledast W_{:,k}],$$

where $W_{:,j}$ denotes the j-th column of W. These definitions directly imply the following linearity properties for any $\alpha, \beta \in \mathbb{H}^k$:

$$\mathcal{V}_k \circledast (\alpha + \beta) = \mathcal{V}_k \circledast \alpha + \mathcal{V}_k \circledast \beta, \qquad (\mathcal{V}_k \circledast W) \circledast \alpha = \mathcal{V}_k \circledast (W\alpha).$$

For $\mathcal{U}_k = [U_1, U_2, \ldots, U_k]$, where $U_i \in \mathbb{H}^{m\times p}$ for $i = 1, \cdots, k$ and $R \in \mathbb{H}^{m\times p}$, their $\boxtimes$ product is defined as

$$\mathcal{U}_k^* \boxtimes R := [\langle R, U_1\rangle, \langle R, U_2\rangle, \ldots, \langle R, U_k\rangle]^T \in \mathbb{H}^k,$$

where $\langle R, U_i\rangle = \mathrm{tr}(U_i^*R)$.

Proposition 2.1. *Let $A, B \in \mathbb{H}^{m\times n}$, $C \in \mathbb{H}^{n\times p}$, and $\alpha \in \mathbb{R}$. Then*

i. $\mathcal{R}(A + B) = \mathcal{R}(A) + \mathcal{R}(B)$, $\mathcal{R}(\alpha A) = \alpha\mathcal{R}(A)$;

ii. $\mathcal{R}(A + B)_c = \mathcal{R}(A)_c + \mathcal{R}(B)_c$, $\mathcal{R}(\alpha A)_c = \alpha\mathcal{R}(A)_c$;

iii. $\mathcal{R}(AC) = \mathcal{R}(A)\mathcal{R}(C)$, $\mathcal{R}(AC)_c = \mathcal{R}(A)\mathcal{R}(C)_c$;

iv. $\mathcal{R}(A^*) = (\mathcal{R}(A))^T$, $\mathcal{R}(A^*)_c = \left((\mathcal{R}(A))^T\right)_c$;

v. $\|A\|_F = \|\mathcal{R}(A)_c\|_F$.

In the real structure-preserving method, the above proposition shows that the quaternion matrix product AC can be evaluated via the real matrix multiplication $\mathcal{R}(A)\mathcal{R}(C)_c \in \mathbb{R}^{4m\times p}$, without forming or storing the full real counterpart $\mathcal{R}(A)\mathcal{R}(C) \in \mathbb{R}^{4m\times 4p}$. This reduces the theoretical arithmetic cost per iteration to one quarter of that required for operating on the full real equivalent system. The algorithm proposed in this paper adopts the aforementioned structure-preserving method throughout its implementation.

3. Global quaternion least squares method

3.1. Global QLSQR algorithm

The LSQR algorithm for solving real least-squares problems is built upon the Golub–Kahan–Lanczos bidiagonalization procedure introduced by Golub and Kahan [22]. Wang extended the LSQR algorithm to the quaternion domain and proposed the QLSQR algorithm for quaternion least-squares problems [19]. Its core procedure reduces matrix A to a lower bidiagonal form via the following recurrences: Given an initial solution x_0 and the residual $r_0 = b - Ax_0$,

$$\begin{cases} \beta_1 u_1 = r_0, \quad \alpha_1 v_1 = A^* u_1, \\ \beta_{i+1} u_{i+1} = A v_i - \alpha_i u_i, \qquad i = 1, 2, \ldots, \\ \alpha_{i+1} v_{i+1} = A^* u_{i+1} - \beta_{i+1} v_i, \end{cases} \tag{3.1}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|u_i\|_2 = \|v_i\|_2 = 1$. Defining $U_k \equiv [u_1, \ldots, u_k]$, $V_k \equiv [v_1, \ldots, v_k]$, and the lower bidiagonal matrix $B_k \in \mathbb{R}^{(k+1)\times k}$ with entries α_i on the diagonal and β_{i+1} on the subdiagonal, the recurrences can be written compactly as $AV_k = U_{k+1}B_k$. Setting $x_k = x_0 + V_k y_k$ with $y_k \in \mathbb{H}^k$ and using the orthonormality of $\{u_i\}$, the residual norm reduces to $\|b - Ax_k\| = \|\beta_1 e_1 - B_k y_k\|$, so that the original problem is transformed into the small-scale real least-squares problem $\min \|\beta_1 e_1 - B_k y\|$, which is solved via Givens rotations applied to B_k.

We now extend this framework to multiple right-hand sides and introduce the Gl-QLSQR method for solving the quaternion matrix least-squares problem (1.3). Let $X_0 \in \mathbb{H}^{n\times s}$ be an initial solution and $R_0 = B - AX_0 \in \mathbb{H}^{m\times s}$ be the initial residual matrix. The global bidiagonalization recurrences are given as follows:

$$\begin{cases} \beta_1 U_1 = R_0, \quad \alpha_1 V_1 = A^* U_1, \\ \beta_{i+1} U_{i+1} = A V_i - \alpha_i U_i, \qquad i = 1, 2, \ldots, \\ \alpha_{i+1} V_{i+1} = A^* U_{i+1} - \beta_{i+1} V_i, \end{cases} \tag{3.2}$$

where the scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|U_i\|_F = \|V_i\|_F = 1$, with $U_i \in \mathbb{H}^{m\times s}$ and $V_i \in \mathbb{H}^{n\times s}$. Let k be the number of steps carried out in (3.2), and define the block quaternion matrices

$$\mathcal{U}_{k+1} \equiv [U_1, U_2, \ldots, U_{k+1}] \in \mathbb{H}^{m\times(k+1)s}, \qquad \mathcal{V}_k \equiv [V_1, V_2, \ldots, V_k] \in \mathbb{H}^{n\times ks}.$$

Using the $\circledast$ product notation introduced in Section 2, the first k recurrences in (3.2) can be compactly reformulated as

$$\begin{cases} \mathcal{U}_{k+1} \circledast (\beta_1 e_1) = R_0, \\ A\mathcal{V}_k = \mathcal{U}_{k+1} \circledast B_k, \\ A^*\mathcal{U}_{k+1} = \mathcal{V}_k \circledast B_k^T + [\, 0_{n\times s}, \cdots, 0_{n\times s}, \ \alpha_{k+1}V_{k+1} \,], \end{cases} \tag{3.3}$$

where e_i denotes the i-th column of the identity matrix I_{k+1} and $B_k \in \mathbb{R}^{(k+1)\times k}$ is the lower bidiagonal matrix

$$B_k \equiv \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}.$$

The orthogonality and spanning properties of the block sequences $\{V_i\}$ and $\{U_i\}$ generated by (3.2) are characterized in the following lemmas.

Lemma 3.1. *Suppose that the global quaternion bidiagonalization process* (3.2) *does not break down in the first k steps, i.e., $\alpha_i > 0$ for $i = 1, \ldots, k$ and $\beta_i > 0$ for $i = 1, \ldots, k+1$. Then the block matrix sequences $V_1, V_2, \ldots, V_k$ and $U_1, U_2, \ldots, U_{k+1}$ form F-orthonormal bases of the quaternion Krylov subspaces $\mathcal{K}_k(A^*A, V_1)$ and $\mathcal{K}_{k+1}(AA^*, U_1)$, respectively.*

Proof. For any $X, Y \in \mathbb{H}^{m\times s}$, $W \in \mathbb{H}^{n\times s}$, and $a \in \mathbb{H}$, it follows from the definition of $\langle\cdot,\cdot\rangle$ that $\langle Xa, Y\rangle = \langle X, Y\rangle a$, $\langle Y, X\rangle = \langle X, Y\rangle^*$, and $\langle AW, Y\rangle = \mathrm{tr}(Y^*AW) = \langle W, A^*Y\rangle$, where the last identity relies only on $(A^*Y)^* = Y^*A$ rather than on the cyclic property of the trace. Note also that α_i and β_i are real and therefore commute with quaternion matrices.

Setting $V_0 := 0$, the recurrences (3.2) can be rewritten as

$$AV_i = \alpha_i U_i + \beta_{i+1}U_{i+1}, \qquad A^*U_i = \alpha_i V_i + \beta_i V_{i-1}. \tag{3.4}$$

We prove the orthonormality by induction. Since $\beta_2\langle U_2, U_1\rangle = \langle V_1, A^*U_1\rangle - \alpha_1 = 0$, the matrices U_1, U_2 and V_1 are F-orthonormal. Assume that $U_1, \ldots, U_{i+1}$ and $V_1, \ldots, V_i$ are F-orthonormal. By (3.4), for $l \leq i$,

$$\alpha_{i+1}\langle V_{i+1}, V_l\rangle = \langle U_{i+1}, AV_l\rangle - \beta_{i+1}\delta_{il} = \beta_{l+1}\delta_{il} - \beta_{i+1}\delta_{il} = 0,$$

and then, for $l \leq i+1$,

$$\beta_{i+2}\langle U_{i+2}, U_l\rangle = \langle V_{i+1}, A^*U_l\rangle - \alpha_{i+1}\delta_{i+1,l} = \alpha_l\delta_{i+1,l} - \alpha_{i+1}\delta_{i+1,l} = 0.$$

Together with $\langle Y, X\rangle = \langle X, Y\rangle^*$, this shows that $U_1, \ldots, U_{i+2}$ and $V_1, \ldots, V_{i+1}$ are F-orthonormal, which completes the induction.

Next, since $A(A^*A)^j = (AA^*)^jA$ for all $j \geq 0$, it follows from (3.4) by induction that $U_{j+1} = \psi_j(AA^*)U_1$ and $V_{j+1} = \varphi_j(A^*A)V_1$, where ψ_j and φ_j are real polynomials of exact degree j, whose leading coefficients are nonzero because $\alpha_i, \beta_i > 0$. Hence $\text{span}\{V_1, \ldots, V_k\} = \mathcal{K}_k(A^*A, V_1)$ and $\text{span}\{U_1, \ldots, U_{k+1}\} = \mathcal{K}_{k+1}(AA^*, U_1)$. Since F-orthonormal matrices are linearly independent, the proof is completed. □

Lemma 3.2. *Let $\mathcal{U}_k = [U_1, U_2, \ldots, U_k]$ be the block quaternion matrix generated by the global quaternion bidiagonalization process* (3.2)*, with $U_i \in \mathbb{H}^{m\times s}$, $i = 1, \ldots, k$. Then for any $\eta = (\eta_1, \ldots, \eta_k)^T \in \mathbb{H}^k$, the following relation holds:*

$$\|\mathcal{U}_k \circledast \eta\|_F = \|\eta\|.$$

Proof. By the definition of the $\circledast$ product, the sesquilinearity of the inner product, and the orthonormality of $\{U_i\}_{i=1}^k$,

$$\|\mathcal{U}_k \circledast \eta\|_F^2 = \Big\langle \sum_{i=1}^k U_i\eta_i,\ \sum_{j=1}^k U_j\eta_j \Big\rangle = \sum_{i=1}^k \sum_{j=1}^k \eta_j^* \langle U_i, U_j\rangle \eta_i = \sum_{i=1}^k \sum_{j=1}^k \eta_j^* \delta_{ij} \eta_i = \sum_{i=1}^k |\eta_i|^2 = \|\eta\|^2,$$

from which the result follows at once. □

At the k-th iteration, we seek an approximate solution of the form

$$X_k = X_0 + \mathcal{V}_k \circledast y_k,$$

where $y_k \in \mathbb{H}^k$. The corresponding residual matrix takes the form

$$R_k = B - AX_k = R_0 - A(\mathcal{V}_k \circledast y_k) = \mathcal{U}_{k+1} \circledast (\beta_1 e_1) - \mathcal{U}_{k+1} \circledast (B_k y_k) = \mathcal{U}_{k+1} \circledast (\beta_1 e_1 - B_k y_k). \quad (3.5)$$

Analogous to the QLSQR algorithm, the Gl-QLSQR method determines y_k by minimizing the Frobenius norm of the residual matrix R_k. By Lemma 3.2 and (3.5), we have

$$\|R_k\|_F = \|\mathcal{U}_{k+1} \circledast (\beta_1 e_1 - B_k y_k)\|_F = \|\beta_1 e_1 - B_k y_k\|, \quad (3.6)$$

which reduces the minimization of $\|R_k\|_F$ to solving the following small-scale least-squares problem:

$$\min_{y_k \in \mathbb{H}^k} \|\beta_1 e_1 - B_k y_k\|.$$

By substituting the solution y_k back into $X_k = X_0 + \mathcal{V}_k \circledast y_k$, an approximate solution to problem (1.3) is obtained. The complete procedure is summarized in Algorithm 1.

Algorithm 1: Gl-QLSQR algorithm

Set $X_0 = 0_{n\times s}$, $R_0 = B, \beta_1 = \|R_0\|_F$, $U_1 = R_0/\beta_1$, $\alpha_1 = \|A^*U_1\|_F$, $V_1 = A^*U_1/\alpha_1$, $W_1 = V_1$, $\bar{\phi}_1 = \beta_1$, $\bar{\rho}_1 = \alpha_1$

For $i = 1, 2, \ldots$ **until convergence Do:**

$T_i = AV_i - \alpha_i U_i$
$\beta_{i+1} = \|T_i\|_F$
$U_{i+1} = T_i/\beta_{i+1}$
$S_i = A^*U_{i+1} - \beta_{i+1}V_i$
$\alpha_{i+1} = \|S_i\|_F$
$V_{i+1} = S_i/\alpha_{i+1}$
$\rho_i = (\bar{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$
$c_i = \bar{\rho}_i/\rho_i$
$s_i = \beta_{i+1}/\rho_i$
$\theta_{i+1} = s_i\alpha_{i+1}$
$\bar{\rho}_{i+1} = c_i\alpha_{i+1}$
$\phi_i = c_i\bar{\phi}_i$
$\bar{\phi}_{i+1} = -s_i\bar{\phi}_i$
$X_i = X_{i-1} + W_i(\phi_i/\rho_i)$
$W_{i+1} = V_{i+1} - W_i(\theta_{i+1}/\rho_i)$
If $|\bar{\phi}_{i+1}| \le \varepsilon\|R_0\|_F$ then stop

EndDo

From the bidiagonalization process (3.2), the column space of $\mathcal{V}_k$ satisfies

$$\mathcal{K}_k(A^*A,\ A^*R_0) = \text{span}\{A^*R_0,\ (A^*A)A^*R_0,\ \ldots,\ (A^*A)^{k-1}A^*R_0\},$$

and the approximate solution satisfies

$$X_k \in X_0 + \mathcal{K}_k(A^*A,\ A^*R_0).$$

More precisely, $X_k - X_0$ can be expressed as

$$X_k - X_0 = \tilde{L}_k(A^*A, A^*R_0),$$

where

$$\tilde{L}_k(A^*A, A^*R_0) = A^*R_0\, c_0 + (A^*A)A^*R_0\, c_1 + \cdots + (A^*A)^{k-1}A^*R_0\, c_{k-1},$$
$$c_i \in \mathbb{H}, \quad i = 0, 1, \ldots, k-1.$$

The convergence properties of the proposed Gl-QLSQR algorithm are established in what follows.

Lemma 3.3. *For any quaternion matrix $A \in \mathbb{H}^{m\times n}$, there exists a unitary matrix $Q \in \mathbb{H}^{n\times n}$ such that*

$$Q^*(A^*A)Q = \Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n),$$

where $\lambda_1, \lambda_2, \cdots, \lambda_n \in \mathbb{R}$. If A has full column rank, then $\lambda_i = \sigma_i^2 > 0$, where $\sigma_1 \ge \cdots \ge \sigma_n > 0$ are the singular values of A.

Proof. Since A^*A is a Hermitian matrix, the conclusion follows directly from [23, Section 6.1]. □

Theorem 3.4. *Let $A \in \mathbb{H}^{m\times n}$ have full column rank and $A^*A = Q\Lambda Q^*$ be as in Lemma 3.3. X_* be the solution of (1.3), $R_* = B - AX_*$, $\kappa = \sigma_1/\sigma_n$, and $\mathcal{P}_k$ denote the set of real polynomials p of degree at most k with $p(0) = 1$. The global convergence factor is defined as*

$$\varepsilon^{(k)} := \min_{p\in\mathcal{P}_k} \max_{1\le i\le n} |p(\lambda_i)|.$$

Then the approximation X_k generated at the k-th step of the Gl-QLSQR algorithm satisfies

(a) $X_k = X_0 + \tilde{L}_k(A^*A, A^*R_0)$ and

$$\|R_k\|_F = \min_{c_0,c_1,\dots,c_{k-1}\in\mathbb{H}} \|R_0 - A\tilde{L}_k(A^*A, A^*R_0)\|_F;$$

(b)

$$\|R_k - R_*\|_F \le \varepsilon^{(k)}\|R_0 - R_*\|_F \le 2\left(\frac{\kappa-1}{\kappa+1}\right)^k \|R_0 - R_*\|_F.$$

In particular, if $B \in \mathrm{Ran}(A)$, then $R_* = 0$ and $\|R_k\|_F \le 2\left(\frac{\kappa-1}{\kappa+1}\right)^k \|R_0\|_F$.

Proof. (a) Since the Gl-QLSQR algorithm seeks $X_k \in X_0 + \mathcal{K}_k(A^*A, A^*R_0)$ that minimizes $\|B - AX_k\|_F$, and every element of $X_0+\mathcal{K}_k(A^*A, A^*R_0)$ can be written as $X_0+\tilde{L}_k(A^*A, A^*R_0)$ for some $c_0, \dots, c_{k-1} \in \mathbb{H}$, the conclusion follows directly.

(b) Since $A^*R_* = 0$, we have $\langle AY, R_*\rangle = \mathrm{tr}((A^*R_*)^*Y) = 0$ for any $Y \in \mathbb{H}^{n\times s}$. Writing $B - AX = R_* + A(X_* - X)$, we obtain

$$\|B - AX\|_F^2 = \|R_*\|_F^2 + \|A(X_* - X)\|_F^2 \tag{3.7}$$

for any $X \in \mathbb{H}^{n\times s}$. Let $E_0 = X_*-X_0$, so that $A^*R_0 = A^*AE_0$. Restricting $c_0, \dots, c_{k-1}$ in (a) to real numbers and setting $q(t) = \sum_{j=0}^{k-1} c_j t^j$, we get $X_* - X_0 - \tilde{L}_k(A^*A, A^*R_0) = p(A^*A)E_0$ with $p(t) = 1 - t\,q(t) \in \mathcal{P}_k$. Hence, by (a) and (3.7),

$$\|R_k - R_*\|_F \le \min_{p\in\mathcal{P}_k} \|A\,p(A^*A)E_0\|_F.$$

Since real coefficients commute with quaternion matrices, $p(A^*A) = Q\,p(\Lambda)Q^*$. Denoting by z_i the i-th row of Q^*E_0, we have

$$\|A\,p(A^*A)E_0\|_F^2 = \sum_{i=1}^n \lambda_i\,p(\lambda_i)^2\|z_i\|^2 \le \max_{1\le i\le n} p(\lambda_i)^2 \sum_{i=1}^n \lambda_i\|z_i\|^2 = \max_{1\le i\le n} p(\lambda_i)^2\,\|R_0 - R_*\|_F^2,$$

which gives the first inequality in (b). Since all λ_i lie in $[\sigma_n^2, \sigma_1^2]$, the second inequality follows from the min–max property of Chebyshev polynomials [24]. □

3.2. Global QRRLSQR algorithm

The LSQR algorithm advances the iterate with the step length ϕ_i/ρ_i supplied by the QR factorization of the bidiagonal matrix, and the residual is not formed explicitly but is monitored through the Givens rotation parameters. Paige and Saunders [25] proposed the residual-reducing least-squares QR algorithm (RRLSQR), which shares the same bidiagonalization process and differs

only in the update: At each step, the step length τ_i is chosen to minimize the residual norm along the current search direction, and the residual is updated explicitly by the corresponding recursion. This modification improves the attainable accuracy for ill-conditioned problems.

In the same manner as in Section 3.1, we now extend RRLSQR to the quaternion matrix least-squares problem (1.3) with multiple right-hand sides and develop the structure-preserving Gl-QRRLSQR algorithm (Algorithm 2). All quantities are stated directly for the multiple right-hand side problem. Let the residual matrix be $R_i = B - AX_i \in \mathbb{H}^{m\times s}$, let $W_i \in \mathbb{H}^{n\times s}$ be the i-th search direction, and set $P_i = AW_i \in \mathbb{H}^{m\times s}$. The step length is

$$\tau_i = \|P_i\|_F^{-2} \langle R_{i-1}, P_i\rangle = \|P_i\|_F^{-2} \operatorname{tr}(P_i^* R_{i-1}),$$

which is the counterpart of the step length $\tau_i = \|p_i\|^{-2} p_i^* r_{i-1}$ used for a single right-hand side. The solution and residual matrices are then updated as

$$X_i = X_{i-1} + W_i\tau_i, \qquad R_i = R_{i-1} - P_i\tau_i.$$

As for a single right-hand side, the modified residual update means that the standard bidiagonalization relation no longer holds directly. The following proposition shows that $\beta_{i+1}U_{i+1}$ can nevertheless be recovered from quantities available within the algorithm, that τ_i is in fact real, and that the residual norm is given by $|\bar{\phi}_{i+1}|$, which justifies the stopping criterion used in Algorithm 2. The corresponding identity for a single right-hand side, $\beta_{i+1}u_{i+1} = p_i - (\bar{\rho}_i/\bar{\phi}_i)\, r_{i-1}$, where $\bar{\rho}_i$ and $\bar{\phi}_i$ are the Givens rotation parameters, was employed in [25] without an explicit proof. Here we state and verify its quaternion counterpart for multiple right-hand sides.

Proposition 3.5. *Suppose that Algorithm 2 does not break down at the first i steps, i.e., $\alpha_l > 0$ and $\beta_{l+1} > 0$ for $l = 1, \ldots, i$. Then*

$$\beta_{i+1}U_{i+1} = P_i - \frac{\bar{\rho}_i}{\bar{\phi}_i} R_{i-1}, \quad \tau_i = \frac{\phi_i}{\rho_i} \in \mathbb{R}, \quad \|R_i\|_F = |\bar{\phi}_{i+1}|, \tag{3.8}$$

where $P_i = AW_i$ and $R_{i-1} = B - AX_{i-1}$.

Proof. The Givens parameters are real scalars and hence commute with quaternion matrices; moreover $\bar{\phi}_{l+1} = -s_l\bar{\phi}_l \neq 0$ for $l \leq i$, since $\bar{\phi}_1 = \beta_1 > 0$ and $s_l \neq 0$. From $W_1 = V_1$, $W_{l+1} = V_{l+1} - W_l(\theta_{l+1}/\rho_l)$, and $X_l = X_{l-1} + W_l\tau_l$, we get $R_l \in \operatorname{span}\{U_1, \ldots, U_{l+1}\}$, and therefore $\langle U_{l+2}, R_l\rangle = 0$ for $l \geq 0$. We argue by induction, carrying along the auxiliary identity $\|R_{i-1}\|_F = |\bar{\phi}_i|$.

For $i = 1$, the initialization gives $R_0 = \beta_1 U_1 = \bar{\phi}_1 U_1$ and $P_1 = AV_1 = \alpha_1 U_1 + \beta_2 U_2$, so that $P_1 - (\bar{\rho}_1/\bar{\phi}_1)R_0 = \beta_2 U_2$.

Assume now that $P_i = \beta_{i+1}U_{i+1} + (\bar{\rho}_i/\bar{\phi}_i)R_{i-1}$ and $\|R_{i-1}\|_F = |\bar{\phi}_i|$. The two terms are orthogonal by the relation above, whence $\langle R_{i-1}, P_i\rangle = \bar{\rho}_i\bar{\phi}_i$ and $\|P_i\|_F^2 = \beta_{i+1}^2 + \bar{\rho}_i^2 = \rho_i^2$, so that $\tau_i = \bar{\rho}_i\bar{\phi}_i/\rho_i^2 = \phi_i/\rho_i \in \mathbb{R}$. Substituting this step length into $R_i = R_{i-1} - P_i\tau_i$ yields

$$R_i = s_i^2 R_{i-1} - s_i c_i \bar{\phi}_i U_{i+1}, \tag{3.9}$$

and taking Frobenius norms gives $\|R_i\|_F^2 = s_i^2\bar{\phi}_i^2(s_i^2 + c_i^2) = \bar{\phi}_{i+1}^2$. Finally, combining $P_{i+1} = AV_{i+1} - P_i(\theta_{i+1}/\rho_i)$ with $AV_{i+1} = \alpha_{i+1}U_{i+1} + \beta_{i+2}U_{i+2}$ and the inductive hypothesis, both P_{i+1} and $(\bar{\rho}_{i+1}/\bar{\phi}_{i+1})R_i$ contain the common part $c_i^2\alpha_{i+1}U_{i+1} - (s_i c_i \alpha_{i+1}/\bar{\phi}_i)R_{i-1}$, which cancels in the difference and leaves

$P_{i+1} - (\bar{\rho}_{i+1}/\bar{\phi}_{i+1})R_i = \beta_{i+2}U_{i+2}$. Indeed, using $\theta_{i+1} = s_i\alpha_{i+1}$, $s_i = \beta_{i+1}/\rho_i$, $c_i = \bar{\rho}_i/\rho_i$, and $c_i^2 + s_i^2 = 1$, we have

$$\begin{aligned} P_{i+1} &= \alpha_{i+1}U_{i+1} + \beta_{i+2}U_{i+2} - \frac{s_i\alpha_{i+1}}{\rho_i}\Big(\beta_{i+1}U_{i+1} + \frac{\bar{\rho}_i}{\bar{\phi}_i}R_{i-1}\Big) \\ &= \beta_{i+2}U_{i+2} + c_i^2\alpha_{i+1}U_{i+1} - \frac{s_ic_i\alpha_{i+1}}{\bar{\phi}_i}R_{i-1}, \end{aligned}$$

while $\bar{\rho}_{i+1} = c_i\alpha_{i+1}$, $\bar{\phi}_{i+1} = -s_i\bar{\phi}_i$ and (3.9) give

$$\frac{\bar{\rho}_{i+1}}{\bar{\phi}_{i+1}}R_i = -\frac{c_i\alpha_{i+1}}{s_i\bar{\phi}_i}(s_i^2R_{i-1} - s_ic_i\bar{\phi}_iU_{i+1}) = c_i^2\alpha_{i+1}U_{i+1} - \frac{s_ic_i\alpha_{i+1}}{\bar{\phi}_i}R_{i-1}.$$

Subtracting the second expression from the first yields $P_{i+1} - (\bar{\rho}_{i+1}/\bar{\phi}_{i+1})R_i = \beta_{i+2}U_{i+2}$, which completes the induction. □

Based on the above discussion, the Gl-QRRLSQR algorithm is summarized in Algorithm 2.

Algorithm 2: Gl-QRRLSQR algorithm

Set $X_0 = 0_{n\times s}$, $R_0 = B, \beta_1 = \|R_0\|_F$, $U_1 = R_0/\beta_1$, $\alpha_1 = \|A^*U_1\|_F$, $V_1 = A^*U_1/\alpha_1$, $W_1 = V_1$, $\bar{\phi}_1 = \beta_1$, $\bar{\rho}_1 = \alpha_1$

For $i = 1, 2, \ldots$ **until convergence Do:**

$P_i = AW_i$
$\tau_i = \langle R_{i-1}, P_i\rangle/\|P_i\|_F^2$
$X_i = X_{i-1} + W_i\tau_i$
$R_i = R_{i-1} - P_i\tau_i$
$T_i = P_i - R_{i-1}(\bar{\rho}_i/\bar{\phi}_i)$
$\beta_{i+1} = \|T_i\|_F$
$U_{i+1} = T_i/\beta_{i+1}$
$S_i = A^*U_{i+1} - \beta_{i+1}V_i$
$\alpha_{i+1} = \|S_i\|_F$
$V_{i+1} = S_i/\alpha_{i+1}$
$\rho_i = (\bar{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$
$c_i = \bar{\rho}_i/\rho_i$
$s_i = \beta_{i+1}/\rho_i$
$\theta_{i+1} = s_i\alpha_{i+1}$
$\bar{\rho}_{i+1} = c_i\alpha_{i+1}$
$\bar{\phi}_{i+1} = -s_i\bar{\phi}_i$
$W_{i+1} = V_{i+1} - W_i(\theta_{i+1}/\rho_i)$
If $|\bar{\phi}_{i+1}| \le \varepsilon\|R_0\|_F$ then stop

EndDo

This algorithm reduces the quaternion least-squares problem to a small-scale one (3.6) using the structure-preserving representation from Section 2, and then solves it via QR factorization of the lower bidiagonal matrix B_k. Since Gl-QLSQR and Gl-QRRLSQR operate on the same bidiagonal matrix and reduce to the same small-scale problem, the convergence analysis established for Gl-QLSQR in Section 3.1 applies equally to Gl-QRRLSQR. Moreover, analogous to the real case in [25], where

the residual reduction strategy has been shown to improve the numerical accuracy for ill-conditioned problems, our newly proposed Gl-QRRLSQR algorithm likewise enhances the numerical accuracy for ill-conditioned problems in the quaternion case. The corresponding results are reported in Example 5.2 of Section 5.

4. Global QLSOR algorithm

Both Gl-QLSQR and Gl-QRRLSQR require a QR factorization of the lower bidiagonal matrix B_k at each iteration. To avoid this cost, Paige and Saunders [25] proposed the oblique projection algorithm(LSOR) within an oblique projection framework. Let L_k denote the first k rows of B_k (i.e., the $k \times k$ lower bidiagonal matrix with diagonal entries α_i and subdiagonal entries β_{i+1}). For a single right-hand side, taking $x_k = x_0 + V_k z_k$ and imposing the oblique projection condition $U_k^* r_k = 0$ yields the lower triangular system $L_k z_k = \beta_1 e_1$, which is solved by forward substitution. Exploiting the bidiagonal structure of L_k further reduces this system to the recurrences $\zeta_k = -(\beta_k/\alpha_k)\zeta_{k-1}$ and $x_k = x_{k-1} + \zeta_k v_k$ with $\zeta_0 = -1$, which require only a scalar recursion and a vector update, so that the QR factorization is completely avoided. The residual satisfies $r_k = -\zeta_k \beta_{k+1} u_{k+1}$, providing an inexpensive stopping criterion.

We now apply this framework directly to quaternion linear systems with multiple right-hand sides, yielding the Gl-QLSOR algorithm. With the same notation, recurrence relation (3.3) can be reformulated as

$$\begin{cases} \beta_1 U_1 = R_0, \\ A\mathcal{V}_k = \mathcal{U}_k \circledast L_k + [\, 0_{m\times s}, \cdots, 0_{m\times s}, \beta_{k+1} U_{k+1} \,], \\ A^* \mathcal{U}_k = \mathcal{V}_k \circledast L_k^T, \end{cases} \tag{4.1}$$

where U_i, V_i and the block matrices $\mathcal{U}_k$, $\mathcal{V}_k$ are generated by the global quaternion Lanczos bidiagonalization process (3.2) and defined as in Section 3.1. Setting $X_k = X_0 + \mathcal{V}_k \circledast z_k$ with $z_k = [\xi_1, \xi_2, \ldots, \xi_k]^T \in \mathbb{H}^k$ and imposing the oblique projection condition $\mathcal{U}_k^* \boxtimes R_k = 0$, we obtain from (4.1)

$$\begin{aligned} 0 = \mathcal{U}_k^* \boxtimes R_k &= \mathcal{U}_k^* \boxtimes (B - AX_k) \\ &= \mathcal{U}_k^* \boxtimes (R_0 - A\mathcal{V}_k \circledast z_k) \\ &= \beta_1 e_1 - L_k z_k. \end{aligned} \tag{4.2}$$

Analogously, the residual satisfies

$$\begin{aligned} R_k &= R_0 - A\mathcal{V}_k \circledast z_k \\ &= \mathcal{U}_k \circledast (\beta_1 e_1) - \mathcal{U}_k \circledast (L_k z_k) - \xi_k \beta_{k+1} U_{k+1} \\ &= \mathcal{U}_k \circledast (\beta_1 e_1 - L_k z_k) - \xi_k \beta_{k+1} U_{k+1} \\ &= -\xi_k \beta_{k+1} U_{k+1}. \end{aligned} \tag{4.3}$$

From (4.2), the scalar ξ_k and the approximate solution X_k are updated via

$$\xi_k = -\frac{\beta_k}{\alpha_k}\xi_{k-1}, \qquad X_k = X_{k-1} + V_k \xi_k,$$

with $\xi_0 = -1$. Incorporating the structural properties of the real representation of quaternion matrices, the structure-preserving Gl-QLSOR algorithm is summarized in Algorithm 3.

Algorithm 3: Gl-QLSOR algorithm

Set $X_0 = 0_{n\times s}$, $V_0 = 0_{n\times s}$, $R_0 = B, \beta_1 = \|R_0\|_F$, $U_1 = R_0/\beta_1$, $\xi_0 = -1$

For $i = 1, 2, \ldots$ **until convergence Do:**

$\alpha_i = \|A^*U_i - \beta_i V_{i-1}\|_F$

$V_i = (A^*U_i - \beta_i V_{i-1})/\alpha_i$

$\beta_{i+1} = \|AV_i - \alpha_i U_i\|_F$

$U_{i+1} = (AV_i - \alpha_i U_i)/\beta_{i+1}$

$\xi_i = -\dfrac{\beta_i}{\alpha_i}\xi_{i-1}$

$X_i = X_{i-1} + V_i\xi_i$

If $|\xi_i|\beta_{i+1} \le \varepsilon\|R_0\|_F$ then stop

EndDo

Theorem 4.1. *The residual matrix of the approximate solution X_k generated by the Gl-QLSOR algorithm satisfies*

$$R_k = -\xi_k\beta_{k+1}U_{k+1}, \tag{4.4}$$

and consequently,

$$\|R_k\|_F = |\xi_k|\beta_{k+1}.$$

Proof. Eq (4.4) is established in (4.3). Taking the Frobenius norm on both sides and using $\|U_{k+1}\|_F = 1$, the conclusion follows. □

In what follows, we show how to apply Algorithms 1–3 to solve the quaternion Sylvester matrix equation. Consider the well-known quaternion Sylvester matrix equation

$$AX + XB = C, \tag{4.5}$$

where $A \in \mathbb{H}^{n\times n}$, $B \in \mathbb{H}^{m\times m}$, and $C \in \mathbb{H}^{n\times m}$ are known matrices, and $X \in \mathbb{H}^{n\times m}$ is required to be determined. Such matrix equations have widespread applications in system and control theory [26], among other disciplines. Rodman [2] proved that the matrix equation has a unique solution if and only if the right eigenvalues of A and B are disjoint. Define the linear operator

$$\mathcal{A}(X) := AX + XB.$$

It is straightforward to verify that the operator $\mathcal{A}$ is a linear mapping from the matrix space $\mathbb{H}^{n\times m}$ to itself. This shows that (4.5) can be rewritten as

$$\mathcal{A}(X) = C. \tag{4.6}$$

Moreover, the adjoint operator $\mathcal{A}^*\colon \mathbb{H}^{n\times m} \to \mathbb{H}^{n\times m}$ of $\mathcal{A}$ is given by

$$\mathcal{A}^*(U) = A^*U + UB^*,$$

which is required by Algorithms 1–3. Let X_0 be an initial guess to (4.5) and

$$R_0 = C - \mathcal{A}(X_0) = C - AX_0 - X_0B$$

be its corresponding residual. The quaternion matrix Krylov subspace $\mathcal{K}_j(\mathcal{A}, R_0)$ is defined as

$$\mathcal{K}_j(\mathcal{A}, R_0) := \text{span}\{R_0, \mathcal{A}(R_0), \ldots, \mathcal{A}^{j-1}(R_0)\},$$

where $\mathcal{A}^i(R_0)$ is defined recursively by $\mathcal{A}^i(R_0) = \mathcal{A}(\mathcal{A}^{i-1}(R_0))$.

5. Numerical experiments

In this section, the three proposed global quaternion algorithms are applied to three classes of experiments: the quaternion least-squares problem, the quaternion Sylvester matrix equation, and color image restoration. All numerical results were obtained using MATLAB R2021a on a personal computer equipped with an AMD Ryzen 7 processor (2.90 GHz) and 16 GB of RAM. For Examples 5.1, 5.3, and 5.5, we also employ their real-valued counterparts Gl-LSQR, Gl-RRLSQR, and Gl-LSOR for comparison. In addition, for Example 5.1, the single right-hand side quaternion algorithms QLSQR, QRRLSQR, and QLSOR are applied column by column to assess the benefit of treating all s right-hand sides simultaneously. In the experiments, "CPU" denotes the computing time in seconds, and "IT" refers to the number of iteration steps. The accuracy of the algorithms is measured by the relative error norm

$$\mathrm{RE} = \frac{\|X_k - X_{\mathrm{true}}\|_F}{\|X_{\mathrm{true}}\|_F}.$$

The stopping criterion is set as $\mathrm{RE} \leq 10^{-6}$, or the number of iterations reaches $k_{\max} = 5000$, unless otherwise stated. In all subsequent numerical experiments, the initial guess for each algorithm is taken to be the zero matrix.

Except for the random test case RANDN in Example 5.1, the real matrix A_0 in Examples 5.1 and 5.3 is chosen from the University of Florida Sparse Matrix Collection [27], which is a large and actively growing set of sparse matrices arising from real applications. In addition, a class of artificially constructed ill-conditioned matrices is employed in Example 5.2, whose detailed construction is described therein.

Table 1. Properties of the test matrices from the Florida Sparse Matrix Collection.

Matrix	Field	m	n	nnz	Symmetric
BFWB62	Electromagnetics	62	62	342	Yes
ASH85	Least-squares problem	85	85	523	Yes
BCSPWR03	Power network	118	118	476	Yes
CAN445	Structural problem	445	445	3809	Yes
BFWA782	Electromagnetics	782	782	7514	No
PDE900	2D PDE problem	900	900	4380	No

Example 5.1. Consider the quaternion linear system with multiple right-hand sides

$$AX = B,$$

where $A \in \mathbb{H}^{n\times n}$, $B \in \mathbb{H}^{n\times s}$ are given matrices and $X \in \mathbb{H}^{n\times s}$ is the matrix to be determined. The coefficient matrix is given by

$$A = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k}, \tag{5.1}$$

where A_0 is chosen from the matrices listed in Table 1, namely BFWB62, BCSPWR03, and PDE900, respectively, and the remaining quaternion components satisfy $A_1 = -A_0$, $A_2 = 2A_0$, $A_3 = 1.5A_0$. In addition, to examine the performance of the algorithms on a general quaternion matrix whose components are not proportional, we also consider a random test case (RANDN) with $n = 400$, in

which A_0, A_1, A_2, A_3 are mutually independent random matrices. The exact solution is defined as $X_{\text{true}} = X_0 + X_1\mathbf{i} + X_2\mathbf{j} + X_3\mathbf{k}$, where $X_l = \text{randn}(n, s)$, $l = 0, 1, 2, 3$ is an $n \times s$ random matrix with entries independently drawn from the standard normal distribution. The right-hand side matrix is defined as $B = AX_{\text{true}}$, so that the system is consistent.

The proposed quaternion-based algorithms Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR are applied to solve this system, together with their real counterparts Gl-LSQR, Gl-RRLSQR, and Gl-LSOR applied to the corresponding real-expanded formulation. For further comparison, the single-right-hand-side quaternion algorithms QLSQR, QRRLSQR, and QLSOR, i.e., the special cases of our proposed Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR with $s = 1$, are applied column by column to the s systems $Ax_j = b_j$, $j = 1, \ldots, s$. For these algorithms, IT denotes the average number of iterations per column, and CPU denotes the total time for all s columns.

Table 2. Numerical results of Example 5.1.

Case	Algorithm	$[m, s]$	IT	CPU	RE
BFWB62	Gl-LSQR	[248,36]	69	0.0228	6.3732e-07
	Gl-RRLSQR	[248,36]	70	0.0248	**4.5808e-07**
	Gl-LSOR	[248,36]	**66**	0.0152	9.9102e-07
	Gl-QLSQR	[62,9]	69	0.0160	5.4775e-07
	Gl-QRRLSQR	[62,9]	69	0.0201	9.7374e-07
	Gl-QLSOR	[62,9]	**66**	**0.0129**	9.8986e-07
	QLSQR	[62,9]	68	0.0421	8.2247e-07
	QRRLSQR	[62,9]	68	0.0429	8.5169e-07
	QLSOR	[62,9]	**66**	0.0370	8.1077e-07
BCSPWR03	Gl-LSQR	[472,24]	252	0.0609	5.9097e-07
	Gl-RRLSQR	[472,24]	252	0.0965	5.9685e-07
	Gl-LSOR	[472,24]	**249**	0.0607	8.7692e-07
	Gl-QLSQR	[118,6]	251	0.0503	9.1148e-07
	Gl-QRRLSQR	[118,6]	252	0.0637	**5.7044e-07**
	Gl-QLSOR	[118,6]	**249**	**0.0482**	6.5980e-07
	QLSQR	[118,6]	251	0.1085	8.2989e-07
	QRRLSQR	[118,6]	252	0.1150	7.0073e-07
	QLSOR	[118,6]	**249**	0.1028	7.1593e-07
PDE900	Gl-LSQR	[3600,28]	458	1.5184	9.8030e-07
	Gl-RRLSQR	[3600,28]	459	1.3976	**9.5544e-07**
	Gl-LSOR	[3600,28]	451	1.2005	9.5678e-07
	Gl-QLSQR	[900,7]	458	0.6123	9.7676e-07
	Gl-QRRLSQR	[900,7]	458	0.9343	9.8567e-07
	Gl-QLSOR	[900,7]	451	**0.5659**	9.6169e-07
	QLSQR	[900,7]	452	1.1738	9.7497e-07
	QRRLSQR	[900,7]	453	1.2513	9.6504e-07
	QLSOR	[900,7]	**445**	1.1509	9.6886e-07
RANDN	Gl-LSQR	[1600,36]	694	17.7678	7.8123e-07
	Gl-RRLSQR	[1600,36]	698	16.9990	**5.8236e-07**
	Gl-LSOR	[1600,36]	**692**	15.2431	9.9803e-07
	Gl-QLSQR	[400,9]	694	6.7344	6.6663e-07
	Gl-QRRLSQR	[400,9]	698	6.8383	4.1008e-07
	Gl-QLSOR	[400,9]	**692**	**6.5113**	9.1587e-07
	QLSQR	[400,9]	695	30.1146	5.3020e-07
	QRRLSQR	[400,9]	698	32.6565	6.8211e-07
	QLSOR	[400,9]	693	29.6298	6.6133e-07

As shown in Table 2, the quaternion-based algorithms attain iteration counts and relative errors comparable to those of their real counterparts across all test cases, including the random case RANDN whose quaternion components are not proportional, while requiring less CPU time as the problem size increases. In particular, the quaternion-based methods achieve an acceleration ratio ranging from 1.18 to 2.64, with the advantage generally becoming more pronounced as the computational scale of the problem grows. This is because the real-expanded formulation enlarges both dimensions of the coefficient matrix by a factor of four. Consequently, the per-iteration arithmetic cost is higher for Gl-LSQR, Gl-RRLSQR, and Gl-LSOR than for their quaternion counterparts. Moreover, compared with QLSQR, QRRLSQR, and QLSOR applied column by column, the global quaternion algorithms require a number of iterations comparable to the average iteration count per column but reduce the CPU time by a factor of 1.34 to 4.78.

Figure 1 depicts the convergence histories of all six algorithms for the BCSPWR03 test case. The relative error decreases to the prescribed tolerance 10^{-6}, and each quaternion-based algorithm exhibits a convergence trajectory virtually indistinguishable from that of its real counterpart, thereby numerically confirming their mathematical equivalence. Among the three global quaternion-based methods, Gl-QLSOR achieves the lowest CPU time, owing to the elimination of the QR factorization of the lower bidiagonal matrix B_k at each iteration.

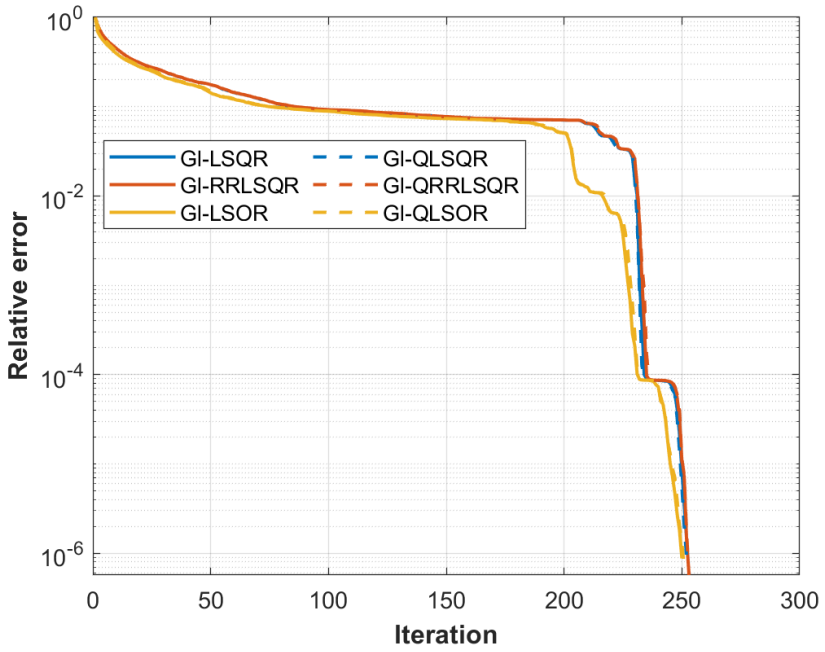


Figure 1. Convergence curves of Example 5.1 with BCSPWR03.

Example 5.2. Consider the quaternion matrix least-squares problem

$$\min_{X\in\mathbb{H}^{n\times s}} \|B - AX\|_F,$$

where $A \in \mathbb{H}^{m\times n}$ with $m > n$, $B \in \mathbb{H}^{m\times s}$ are given matrices and $X \in \mathbb{H}^{n\times s}$ is the matrix to be determined. In this example, artificially constructed ill-conditioned matrices are employed to further investigate the

numerical behavior of the proposed algorithms. Analogous to the real case in [25], where the residual reduction strategy has been shown to improve the numerical accuracy for ill-conditioned problems, our newly proposed Gl-QRRLSQR algorithm likewise enhances the numerical accuracy for ill-conditioned problems in the quaternion case. In what follows, we verify this phenomenon numerically, namely that Gl-QRRLSQR outperforms Gl-QLSQR in terms of residual accuracy when the coefficient matrix is severely ill-conditioned.

In (5.1), let A_0, A_2 be given by rand(m, n) and A_1, A_3 by ones(m, n), following Example 2 of [20]. To enhance the ill-conditioning, the last two columns of A_0 and A_2 are made nearly collinear with their predecessors:

$$A_t(:, \ell) \leftarrow A_t(:, \ell - 1) + \delta A_t(:, \ell), \qquad t \in \{0, 2\},\ \ell \in \{n - 1, n\}, \tag{5.2}$$

so that smaller values of δ yield more severely ill-conditioned coefficient matrices; we take $\delta = 10^{-14}$. The right-hand side is taken as $B = AX_{\text{ref}} + \nu\Delta$, where $X_{\text{ref}} \in \mathbb{H}^{n\times s}$ is a prescribed quaternion matrix and Δ is a random perturbation with $\|\Delta\|_F = \|AX_{\text{ref}}\|_F$. Since $m > n$ and Δ is random, $B \notin \text{Ran}(A)$ in general, so that the system $AX = B$ is inconsistent and the least-squares residual is nonzero. We measure convergence by the relative residual

$$\eta^{(k)} = \log_{10}\left(\frac{\|B - AX_k\|_F}{\|B\|_F}\right), \tag{5.3}$$

in line with [20, Eq (34)]. We set $\nu = 10^{-13}$, $s = 8$, maxit $= 2000$ and tol $= 10^{-14}$ and test problems of three sizes $(m, n) \in \{(30, 15), (40, 20), (60, 30)\}$. The condition number cond$(\mathcal{R}(A))$ and the final residuals η_1, η_2, η_3 (i.e., the values of $\eta^{(k)}$ at termination) of Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR are listed in Table 3.

Table 3. Residual results of Example 5.2.

m	n	cond$(\mathcal{R}(A))$	η_1 (Gl-QLSQR)	η_2 (Gl-QRRLSQR)	η_3 (Gl-QLSOR)
30	15	1.35×10^{17}	−12.510	**−13.125**	+7.085
40	20	3.66×10^{16}	−12.244	**−13.122**	−0.320
60	30	1.03×10^{17}	−12.024	**−13.133**	−1.492

The results show that Gl-QRRLSQR achieves a smaller residual than Gl-QLSQR in all three cases, consistent with Table 3 of [20]. In contrast, the final residuals of Gl-QLSOR are much larger than those of Gl-QLSQR and Gl-QRRLSQR, and in the case $(m, n) = (30, 15)$, the iteration diverges. This indicates that Gl-QLSOR, which is based on the oblique projection condition rather than on residual minimization, is not well suited to extremely ill-conditioned least-squares problems. This comparison also highlights the advantage of the residual-reducing strategy of Gl-QRRLSQR in such settings.

Example 5.3. We now examine the quaternion Sylvester matrix equation presented in (4.5), where the quaternion coefficient matrices A and B are of the form $A = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k}$ and $B = B_0 + B_1\mathbf{i} + B_2\mathbf{j} + B_3\mathbf{k}$, respectively. The constituent real matrices A_i and B_i are prescribed as follows [18]:

(a) $A_0 = \text{ASH85}$ $(n = 85)$, $B_1 = B_0$, $B_2 = -B_0$, $B_3 = -1.5B_0$,

(b) $A_0 = \text{CAN445}$ $(n = 445)$, $B_1 = 1.5B_0$, $B_2 = B_0$, $B_3 = -B_0$,

(c) A_0 = BFWA782 ($n = 782$), $B_1 = 2B_0$, $B_2 = -B_0$, $B_3 = 1.5B_0$,

as in (5.1) of Example 5.1, and

$$B_0 = \begin{bmatrix} 2 & 1 & & & \\ -1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & 1 \\ & & & -1 & 2 \end{bmatrix}$$

of order m. The exact solution $X_{\text{true}} = X_0 + X_1\mathbf{i} + X_2\mathbf{j} + X_3\mathbf{k}$ is generated with $X_\ell = \text{randn}(n, m)$, $\ell = 0, 1, 2, 3$, and the right-hand side matrix $\mathbf{C}$ is then computed via

$$C = AX_{\text{true}} + X_{\text{true}}B.$$

Table 4. Numerical results of Example 5.3.

Case	Algorithm	$[n, m]$	IT	CPU	RE
ASH85	Gl-LSQR	[340, 20]	340	0.137	9.0587e-07
	Gl-RRLSQR	[340, 20]	341	0.152	9.9190e-07
	Gl-LSOR	[340, 20]	**337**	0.130	9.6988e-07
	Gl-QLSQR	[85, 5]	340	0.071	9.5427e-07
	Gl-QRRLSQR	[85, 5]	341	0.080	9.3757e-07
	Gl-QLSOR	[85, 5]	338	**0.066**	**8.8860e-07**
CAN445	Gl-LSQR	[1780, 80]	1068	9.841	9.8917e-07
	Gl-RRLSQR	[1780, 80]	1069	11.478	**9.8888e-07**
	Gl-LSOR	[1780, 80]	**1028**	9.184	9.9848e-07
	Gl-QLSQR	[445, 20]	1068	4.623	9.9012e-07
	Gl-QRRLSQR	[445, 20]	1069	4.892	9.8954e-07
	Gl-QLSOR	[445, 20]	**1028**	**4.308**	9.9775e-07
BFWA782	Gl-LSQR	[3128, 100]	3189	48.202	9.9686e-07
	Gl-RRLSQR	[3128, 100]	3189	61.906	9.9743e-07
	Gl-LSOR	[3128, 100]	3120	54.129	9.9625e-07
	Gl-QLSQR	[782, 25]	3117	22.393	9.9865e-07
	Gl-QRRLSQR	[782, 25]	3119	23.440	**9.9412e-07**
	Gl-QLSOR	[782, 25]	**3062**	**20.403**	9.9766e-07

Applying Gl-LSQR, Gl-RRLSQR, Gl-LSOR, and the proposed algorithms Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR to solve (4.5), we report the numerical results in Table 4. For all test cases, the iteration counts and relative errors of the quaternion-based methods remain comparable to those of their real-arithmetic counterparts, while the CPU time is consistently reduced by approximately a factor of two, and this advantage widens with increasing problem dimension. Among the three quaternion-based methods, Gl-QLSOR consistently attains the lowest CPU time across all test cases.

To further assess the effectiveness of the proposed algorithms for solving problem (1.3) in color image restoration, the original color image is represented as a pure quaternion matrix $X_{\text{true}} = X_1\mathbf{i} + X_2\mathbf{j} + X_3\mathbf{k} \in \mathbb{H}^{n\times s}$, where $X_1, X_2, X_3 \in \mathbb{R}^{n\times s}$ denote the red, green, and blue channels, respectively, with pixel values in $[0, d]$ ($d = 255$). The blurring matrix $A \in \mathbb{H}^{m\times n}$ acts on X_{true} to produce the blurred observation $B = AX_{\text{true}} \in \mathbb{H}^{m\times s}$. A common alternative is to vectorize the columns of X_{true} and B into

a single large-scale system of dimension $ms \times ns$, whose cost grows linearly with s. The proposed methods instead solve (1.3) directly within the global quaternion matrix Krylov subspace framework, avoiding the explicit construction of the block-diagonal system and yielding substantial savings in both arithmetic operations and memory.

The reconstruction quality is measured by four metrics: computing time (CPU), peak signal-to-noise ratio (PSNR) in decibels, structural similarity index (SSIM) [28], and relative error (RE), defined as

$$\mathrm{PSNR}(X_k) = 10\log_{10}\left(\frac{3nsd^2}{\|X_{\text{true}} - X_k\|_F^2}\right), \qquad \mathrm{RE}(X_k) = \frac{\|X_{\text{true}} - X_k\|_F}{\|X_{\text{true}}\|_F}, \tag{5.4}$$

$$\mathrm{SSIM}(X_k) = \frac{(2\mu_X\mu_{X_k} + c_1)(2\sigma_{XX_k} + c_2)}{(\mu_X^2 + \mu_{X_k}^2 + c_1)(\sigma_X^2 + \sigma_{X_k}^2 + c_2)}, \tag{5.5}$$

where μ_X, μ_{X_k} and σ_X^2, $\sigma_{X_k}^2$ denote the means and variances of X_{true} and X_k, respectively, σ_{XX_k} is their covariance, and $c_1 = (0.01L)^2$, $c_2 = (0.03L)^2$ with L denoting the dynamic range of pixel intensities. The iteration stops when the relative error satisfies $\mathrm{RE} < 10^{-2}$ or the iteration count reaches 300.

Example 5.4. In this example, we consider two original color images: the "Peppers" image of size 125×120 and the "Lena" image of size 520×500. Both images are blurred by a single-channel blurring matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, whose entries are defined as follows [29, 30]:

$$a_{ij} = \begin{cases} \dfrac{1}{\sigma\sqrt{2\pi}} \exp\left(-\dfrac{(i-j)^2}{2\sigma^2}\right), & |i-j| \le r, \\ 0, & \text{otherwise}, \end{cases} \tag{5.6}$$

with parameters $r = 35$ and $\sigma = 10$, which are chosen following [18]. This blur model corresponds to image degradation caused by atmospheric turbulence.

Table 5. Numerical results of Example 5.4 with single-channel blur.

Case	Algorithm	PSNR	SSIM	CPU	RE
Peppers	Gl-QLSQR	32.4091	0.9705	1.213	6.0199e-02
	Gl-QRRLSQR	32.1892	0.9693	1.335	6.1743e-02
	Gl-QLSOR	**32.7378**	**0.9716**	**1.144**	**5.7964e-02**
Lena	Gl-QLSQR	30.2766	0.9775	27.248	5.5356e-02
	Gl-QRRLSQR	30.2765	0.9755	32.238	5.5356e-02
	Gl-QLSOR	**30.9320**	**0.9803**	**25.765**	**5.1333e-02**

We apply Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR for solving problem (1.3) and report the numerical results and the restored images in Table 5 and Figure 2. From this table, one can see that the restoration quality of all three algorithms is comparable in terms of PSNR and SSIM. Compared with Gl-QLSQR and Gl-QRRLSQR, Gl-QLSOR restores images with higher quality and requires less computing time. Moreover, Gl-QRRLSQR consistently incurs the highest computational cost among the three methods.

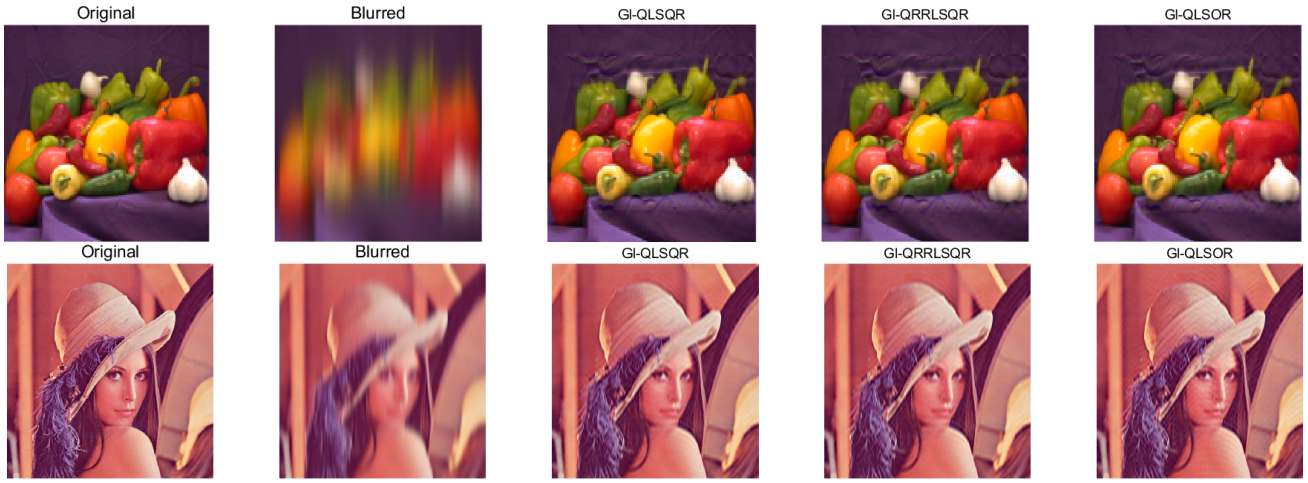


Figure 2. Left to right: The original images, blurred images, and deblurred images obtained by Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR, respectively.

Example 5.5. In this example, two original color images are considered: the "Onion" image of size 128×120 and the "House" image of size 520×512. These images are blurred by a multichannel blurring matrix of the form $A = A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{H}^{n\times n}$. Note that these images, though visually appearing grayscale, are color images with identical RGB channels.

Let $H_0 \in \mathbb{R}^{p\times p}$ be the Toeplitz matrix defined in (5.6) with $r = \sigma = 3$, and let $H_1 \in \mathbb{R}^{q\times q}$ be the Toeplitz matrix introduced in [29], whose entries are given by

$$h_{ij} = \begin{cases} \dfrac{1}{2\omega - 1}, & |i - j| \le \omega, \\ 0, & \text{otherwise}, \end{cases} \tag{5.7}$$

with $\omega = 5$. We choose p and q such that $pq = n$. Specifically, $p = 16$, $q = 8$ for the case $n = 128$, and $p = 20$, $q = 26$ for the case $n = 520$. The real components of the blurring matrix are then defined as $A_1 = H_0 \otimes H_1 \in \mathbb{R}^{n\times n}$ and $A_2 = A_3 = -0.5\,A_1$. The parameters r, σ, ω, p, and q are chosen following [18].

We apply the Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR methods, together with their real-valued counterparts Gl-LSQR, Gl-RRLSQR, and Gl-LSOR, to this problem. The numerical results and the restored color images are reported in Table 6 and Figure 3, respectively. All three quaternion algorithms effectively restore the color images blurred by the multichannel blurring matrix, and their PSNR and SSIM values are nearly identical to those of the corresponding real-valued algorithms. Meanwhile, the quaternion algorithms reduce the CPU time by a factor of about 3.2 to 3.8 compared with their real counterparts, since the latter operate on the real-expanded system whose dimensions are four times larger. Among the three quaternion methods, Gl-QLSOR achieves slightly higher PSNR and SSIM values than Gl-QLSQR and Gl-QRRLSQR, and requires the least CPU time in both test cases, confirming its computational efficiency.

Table 6. Numerical results of Example 5.5 with multichannel blur.

Case	Algorithm	PSNR	SSIM	CPU	RE
Onion	Gl-QLSQR	31.6187	0.9336	2.154	5.9121e-02
	Gl-QRRLSQR	31.6065	0.9340	2.408	5.9095e-02
	Gl-QLSOR	**31.6269**	**0.9341**	**2.104**	**5.8095e-02**
	Gl-LSQR	31.6181	0.9339	6.882	5.9122e-02
	Gl-RRLSQR	31.6181	0.9340	9.108	5.9122e-02
	Gl-LSOR	31.6181	0.9336	6.896	5.9122e-02
House	Gl-QLSQR	32.2232	0.8971	51.976	3.7098e-02
	Gl-QRRLSQR	32.2232	0.8970	55.496	3.7099e-02
	Gl-QLSOR	32.4345	0.9016	**49.827**	3.6207e-02
	Gl-LSQR	32.2174	0.8970	175.491	3.7098e-02
	Gl-RRLSQR	32.2151	0.8970	207.395	3.7108e-02
	Gl-LSOR	**32.4538**	**0.9020**	174.701	**3.6102e-02**

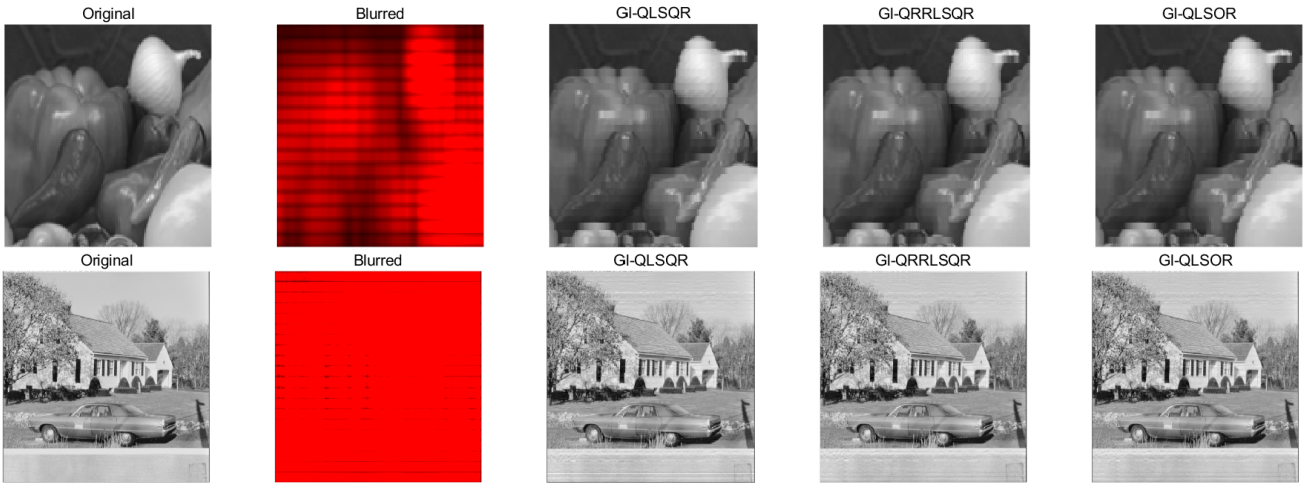


Figure 3. Left to right: The original images, blurred images, and deblurred images obtained by Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR, respectively.

6. Conclusions

In this paper, we have proposed three global structure-preserving iterative algorithms, namely Gl-QLSQR, Gl-QRRLSQR, and Gl-QLSOR, for solving quaternion linear systems with multiple right-hand sides. All three methods are built upon the Golub–Kahan–Lanczos bidiagonalization as the core algorithmic framework. By exploiting the real structure-preserving representation of quaternion matrices, they reduce the theoretical per-iteration arithmetic cost to approximately one quarter of that required by methods operating on the full real equivalent system. Gl-QRRLSQR incorporates an explicit residual update at each iteration, which enhances its numerical accuracy over Gl-QLSQR for ill-conditioned problems. Gl-QLSOR, developed within an oblique projection framework, eliminates the QR factorization of the bidiagonal matrix at each step, thereby further reducing the per-iteration cost while maintaining comparable accuracy for consistent problems; for severely ill-conditioned inconsistent problems, however, it is less reliable than the residual-minimizing methods, as shown in

Example 5.2. Convergence analysis is established for the proposed methods, and all three algorithms are extended to solve the quaternion Sylvester matrix equation. Numerical experiments on both least-squares problems and color image deblurring tasks demonstrate that the proposed algorithms achieve favorable computational efficiency and reconstruction quality. We plan to investigate effective preconditioning techniques to further enhance the robustness and applicability of the proposed methods for large-scale and severely ill-conditioned problems. Moreover, the framework developed in this paper can be extended to other quaternion matrix equations, such as the generalized Sylvester equation.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 12271216 and 52273034).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. W. R. Hamilton, *Elements of Quaternions*, Longmans, Green, and Company, 1866.
2. L. Rodman, *Topics in Quaternion Linear Algebra*, Princeton University Press, 2014.
3. Z. Jia, M. K. Ng, G. Song, Lanczos method for large-scale quaternion singular value decomposition, *Numerical Algorithms*, **82** (2019), 699–717. https://doi.org/10.1007/s11075-018-0621-0
4. F. Zhang, M. Wei, Y. Li, J. Zhao, Special least squares solutions of the quaternion matrix equation $AX = B$ with applications, *Appl. Math. Comput.*, **270** (2015), 425–433. https://doi.org/10.1016/j.amc.2015.08.046
5. P. Ginzberg, *Quaternion Matrices: Statistical Properties and Applications to Signal Processing and Wavelets*, Ph.D. thesis, Imperial College London, 2013.
6. S. L. Adler, *Quaternionic Quantum Mechanics and Quantum Fields*, Oxford University Press, 1995.
7. Y. Guan, M. T. Chu, D. Chu, SVD-based algorithms for the best rank-1 approximation of a symmetric tensor, *SIAM J. Matrix Anal. Appl.*, **39** (2018), 1095–1115. https://doi.org/10.1137/17M1136699
8. Z. Jia, M. K. Ng, G. Song, Robust quaternion matrix completion with applications to image inpainting, *Numer. Linear Algebra Appl.*, **26** (2019), e2245. https://doi.org/10.1002/nla.2245

9. G. Fan, H. Zhong, Block Golub–Kahan–Lanczos algorithms for quaternion singular value decomposition, *Numerical Algorithms*, (2026), 1–20. https://doi.org/10.1007/s11075-026-02320-z
10. L. Guo, M. Zhu, X. Ge, Reduced biquaternion canonical transform, convolution and correlation, *Signal Process.*, **91** (2011), 2147–2153. https://doi.org/10.1016/j.sigpro.2011.03.017
11. T. Isokawa, H. Nishimura, N. Matsui, Commutative quaternion and multistate Hopfield neural networks, in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, IEEE, (2010), 1–6. https://doi.org/10.1109/IJCNN.2010.5596736
12. A. El Guennouni, K. Jbilou, H. Sadok, The block Lanczos method for linear systems with multiple right-hand sides, *Appl. Numer. Math.*, **51** (2004), 243–256. https://doi.org/10.1016/j.apnum.2004.04.001
13. V. Simoncini, E. Gallopoulos, Convergence properties of block GMRES and matrix polynomials, *Linear Algebra Appl.*, **247** (1996), 97–119. https://doi.org/10.1016/0024-3795(95)00093-3
14. R. W. Freund, M. Malhotra, A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides, *Linear Algebra Appl.*, **254** (1997), 119–157. https://doi.org/10.1016/S0024-3795(96)00529-0
15. K. Jbilou, A. Messaoudi, H. Sadok, Global FOM and GMRES algorithms for matrix equations, *Appl. Numer. Math.*, **31** (1999), 49–63. https://doi.org/10.1016/S0168-9274(98)00094-4
16. Z. Jia, M. K. Ng, Structure preserving quaternion generalized minimal residual method, *SIAM J. Matrix Anal. Appl.*, **42** (2021), 616–634. https://doi.org/10.1137/20M133751X
17. T. Li, Q. Wang, Structure preserving quaternion full orthogonalization method with applications, *Numer. Linear Algebra Appl.*, **30** (2023), e2495. https://doi.org/10.1002/nla.2495
18. T. Li, Q. Wang, X. Zhang, Gl-QFOM and Gl-QGMRES: two efficient algorithms for quaternion linear systems with multiple right-hand sides, *Numer. Linear Algebra Appl.*, **32** (2025), e70008. https://doi.org/10.1002/nla.70008
19. M. Wang, Algorithm Q-LSQR for the least squares problem in quaternionic quantum theory, *Comput. Phys. Commun.*, **181** (2010), 1047–1050. https://doi.org/10.1016/j.cpc.2010.02.011
20. S. Ling, Z. Jia, Matrix iterative algorithms for least-squares problem in quaternionic quantum theory, *Int. J. Comput. Math.*, **90** (2013), 727–745. https://doi.org/10.1080/00207160.2012.739684
21. F. Toutounian, S. Karimi, Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides, *Appl. Math. Comput.*, **178** (2006), 452–460. https://doi.org/10.1016/j.amc.2005.11.065
22. G. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, *J. Soc. Ind. Appl. Math. Ser. B*, **2** (1965), 205–224. https://doi.org/10.1137/0702016
23. M. Wei, Y. Li, F. Zhang, J. Zhao, *Quaternion Matrix Computations*, Nova Science Publishers, 2018.
24. Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.
25. C. C. Paige, M. A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Software*, **8** (1982), 43–71. https://doi.org/10.1145/355984.355989

26. Z. He, Q. Wang, Y. Zhang, A system of quaternary coupled Sylvester-type real quaternion matrix equations, *Automatica*, **87** (2018), 25–31. https://doi.org/10.1016/j.automatica.2017.09.008

27. T. A. Davis, Y. Hu, The University of Florida sparse matrix collection, *ACM Trans. Math. Software*, **38** (2011), 1–25. https://doi.org/10.1145/2049662.2049663

28. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.*, **13** (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

29. A. Bouhamidi, R. Enkhbat, K. Jbilou, Conditional gradient Tikhonov method for a convex optimization problem in image restoration, *J. Comput. Appl. Math.*, **255** (2014), 580–592. https://doi.org/10.1016/j.cam.2013.06.011

30. J. Li, W. Li, R. Huang, An efficient method for solving a matrix least squares problem over a matrix inequality constraint, *Comput. Optim. Appl.*, **63** (2016), 393–423. https://doi.org/10.1007/s10589-015-9783-z



AIMS Press

© 2027 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)