

Iterative Methods for Least Squares Problem in Split Quaternionic Mechanics

Minghui Wang* and Lingling Yue

Department of Mathematics, Qingdao University of Science and Technology, Qingdao, 266061, P.R. China
Email: mhwang@yeah.net

Abstract Split Quaternionic least squares (SQLS) problem is one method of solving overdetermined sets of quaternion linear equations $AX = E$ that is appropriate when there is error in the matrix E . In this paper, by means of real representation of split quaternion matrices, we derive an iterative method for finding the minimum-norm solution of the SQLS problems in split quaternionic mechanics.

Keywords: Split quaternion matrix, Split quaternionic mechanics, Least squares problem, LSQR, Iterative algorithm.

1 Introduction

The quaternion invented by William Rowan Hamilton(1805-1865) has been widely used in quaternionic quantum mechanics and many other fields. In 1849, James Cockle found the split quaternion, which has the following form:

$$q = q_1 + q_2i + q_3j + q_4k, i^2 = -1, j^2 = k^2 = 1, ijk = 1,$$

where q_1, q_2, q_3, q_4 are real and $ij = -ji = k, jk = -kj = -i, ki = -ik = j$. The set of all split quaternions is a ring, denoted by \mathbf{SQ} . This ring is an associative, noncommutative four-dimensional Clifford algebra and has zero divisors, nilpotent elements and nontrivial idempotents[1]. \mathbf{SQ} is different from the quaternion ring and has more complicated algebraic structure. For details, see [1] and the references therein.

In complexified classical and non-Hermitian quantum mechanics, there are surprising relations between quaternionic and split quaternionic mechanics [2]. In the literature over the past decade, the complexified mechanical systems with real energies are studied extensively, which can alternatively be viewed as certain split quaternionic extensions of the underlying real mechanical systems [3].

For dealing with some problems in the theory and numerical computations of split quaternionic mechanics, one will meet problems of approximate solutions of split quaternion linear equations $AX \approx E$, that is appropriate when there is error in the matrix E , i.e. split quaternionic least squares (SQLS) problem. The main difficulty in solving this problem is the non-commutative and non-skew-field of the split quaternion and the standard mathematical methods(see [4,5,6] and their references) of the complex number field cannot work. In [4], for the first time, the split quaternionic least squares (SQLS) problem was discussed by means of the real representation and the complex representation, which is also main methods for researching the quaternionic least squares (QLS) problem[7,8].

In this paper, by means of our real representation, we study the split quaternionic least squares (SQLS) problem, and derive iterative methods for finding solutions of the SQLS problem in split quaternionic mechanics.

Let \mathbf{R} and $\mathbf{SQ} = \mathbf{R} \oplus \mathbf{R}i \oplus \mathbf{R}j \oplus \mathbf{R}k$ denote the real number field and the split quaternion ring, respectively. For $a = a_1 + a_2i + a_3j + a_4k, b = b_1 + b_2i + b_3j + b_4k \in \mathbf{SQ}$, the conjugate of a is defined as $\bar{a} = a_1 - a_2i - a_3j - a_4k$, then $a\bar{a} = a_1^2 + a_2^2 - a_3^2 - a_4^2$. The module $|a|$ of a split quaternion a is defined as $|a| = \sqrt{|a\bar{a}|} = \sqrt{|a_1^2 + a_2^2 - a_3^2 - a_4^2|}$. a is said to be a unit split quaternion if its norm is 1.and their multiplication is defined as

$$ab = (a_1b_1 - a_2b_2 + a_3b_3 + a_4b_4) + (a_2b_1 + a_1b_2 + a_4b_3 - a_3b_4)i \\ + (a_3b_1 + a_4b_2 + a_1b_3 - a_2b_4)j + (a_4b_1 - a_3b_2 + a_2b_3 + a_1b_4)k.$$

For any quaternion matrix A , A^T and A^H denote the transpose, and conjugate transpose of A , respectively. $A(i : j, k : l)$ represents the submatrix of A containing the intersection of rows i to j and columns k to l . I_n denotes the unit matrix of order n .

In this paper, we will define a real presentation of the split quaternion matrix and study its properties. As the application of our real presentation, then we will give an alternative of the split quaternion norm, which enable us to define and study the split quaternionic least squares (SQLS) problem. Iterative methods are obtained for finding solutions of the SQLS problem in split quaternionic mechanics. At last, one numerical experiment will be provided to demonstrate the efficiency of our algorithms.

2 Real Representation of Split Quaternion Matrices

For any $A = A_1 + A_2i + A_3j + A_4k \in \mathbf{Q}^{m \times n}$, $A_l \in \mathbf{R}^{m \times n}$ ($l = 1, 2, 3, 4$), define

$$A^R \equiv \begin{pmatrix} A_1 & -A_2 & A_3 & A_4 \\ A_2 & A_1 & A_4 & -A_3 \\ A_3 & A_4 & A_1 & -A_2 \\ A_4 & -A_3 & A_2 & A_1 \end{pmatrix} \in \mathbf{R}^{4m \times 4n}. \tag{1}$$

The real matrix A^R is known as *the real representation* or *the real representation matrix* of the split quaternion matrix A . The set of all matrices shaped like (1) is denoted by $\mathbf{Rr}^{4m \times 4n}$.

Let

$$Q_t = \begin{pmatrix} 0 & -I_t & 0 & 0 \\ I_t & 0 & 0 & 0 \\ 0 & 0 & 0 & I_t \\ 0 & 0 & -I_t & 0 \end{pmatrix}, S_t = \begin{pmatrix} 0 & 0 & 0 & I_t \\ 0 & 0 & -I_t & 0 \\ 0 & -I_t & 0 & 0 \\ I_t & 0 & 0 & 0 \end{pmatrix}, R_t = \begin{pmatrix} 0 & 0 & I_t & 0 \\ 0 & 0 & 0 & I_t \\ I_t & 0 & 0 & 0 \\ 0 & I_t & 0 & 0 \end{pmatrix}.$$

Then Q_t, R_t, S_t are orthogonal matrices.

By simple computation, we can obtain the following properties.

- Theorem 2.1** *Let $A, B \in \mathbf{Q}^{m \times n}, C \in \mathbf{Q}^{n \times s}, \alpha \in \mathbf{R}$. Then*
- (1). $(A + B)^R = A^R + B^R, (\alpha A)^R = \alpha A^R, (AC)^R = A^R C^R$;
 - (2). $Q_m^2 = -I_{4m}, R_m^2 = S_m^2 = I_{4m}, Q_m^T = -Q_m, R_m^T = R_m, S_m^T = S_m$;
 - (3). $Q_m^T A^R Q_m = A^R, R_m^T A^R R_m = A^R, S_m^T A^R S_m = A^R$.

It is easy to verify that the following results are right.

Theorem 2.2 *For any $V \in \mathbf{R}^{4m \times 4n}$,*

$$V + Q_m^T V Q_m + R_m^T V R_m + S_m^T V S_m \in \mathbf{Rr}^{4m \times 4n}$$

and it is the real representation matrix of the commutative quaternion matrix

$$\tilde{V} = \frac{1}{4}(I_m, I_m i, I_m j, I_m k)(V + Q_m^T V Q_m + R_m^T V R_m + S_m^T V S_m) \begin{pmatrix} I_n \\ -I_n i \\ I_n j \\ -I_n k \end{pmatrix}.$$

Proof. Partitioning V into

$$V = \begin{pmatrix} V_{11} & V_{12} & V_{13} & V_{14} \\ V_{21} & V_{22} & V_{23} & V_{24} \\ V_{31} & V_{32} & V_{33} & V_{34} \\ V_{41} & V_{42} & V_{43} & V_{44} \end{pmatrix}$$

and taking

$$\hat{V} = V + Q_m^T V Q_m + R_m^T V R_m + S_m^T V S_m,$$

we can verify $\hat{V} \in \mathbf{Rr}^{4m \times 4n}$ with

$$\begin{aligned}\hat{V}_{11} &= V_{11} + V_{22} + V_{33} + V_{44}, \hat{V}_{21} = V_{21} - V_{12} + V_{43} - V_{34}, \\ \hat{V}_{31} &= V_{31} + V_{42} + V_{13} + V_{24}, \hat{V}_{41} = V_{41} - V_{32} + V_{23} - V_{14},\end{aligned}$$

and \hat{V} is the real representation matrix of the commutative quaternion matrix

$$\tilde{V} = \hat{V}_{11} + \hat{V}_{21}i + \hat{V}_{31}j + \hat{V}_{41}k = \frac{1}{4}(I_m, I_m i, I_m j, I_m k)\hat{V} \begin{pmatrix} I_n \\ -I_n i \\ I_n j \\ -I_n k \end{pmatrix}. \quad \square$$

Further, we can also get the following construction method.

Theorem 2.3 For any $V \in \mathbf{R}^{4m \times n}$,

$$(V, Q_m V, R_m V, S_m V) \in \mathbf{Rr}^{4m \times 4n}.$$

The following judgment rule can be obtained from Theorem 2.1 and Theorem 2.2.

Theorem 2.4 For any $V \in \mathbf{R}^{4m \times 4n}$, $V \in \mathbf{Rr}^{4m \times 4n}$ if and only if

$$V = Q_m^T V Q_m = R_m^T V R_m = S_m^T V S_m.$$

3 The Split Quaternionic Least Squares Problem

Firstly, we discuss the norm of the split quaternion matrix. By the Frobenius norm of complex matrices, we define the following Frobenius norm of the split quaternion matrix $A = A_0 + A_1i + A_2j + A_3k \in \mathbf{SQ}^{m \times n}$.

$$\|A\|_{(F)} \equiv \|A^R\|_F, \quad (2)$$

which has the following properties:

- (1) $\|A\|_{(F)} \geq 0$, and equality holds if and only if $A = 0$;
- (2) for $\alpha \in \mathbf{R}$, $\|\alpha A\|_{(F)} = |\alpha| \|A\|_{(F)}$;
- (3) $\|A + B\|_{(F)} \leq \|A\|_{(F)} + \|B\|_{(F)}$;
- (4) $\|AB\|_{(F)} \leq \|A\|_{(F)} \|B\|_{(F)}$.
- (5) $\|A\|_{(F)}^2 = 2(\|A_0\|_F^2 + \|A_1\|_F^2 + \|A_2\|_F^2 + \|A_3\|_F^2)$.

But, different from the Frobenius norm of complex matrices,

- (1) for $\alpha \in \mathbf{SQ}$, $\|\alpha A\|_{(F)} \neq |\alpha| \|A\|_{(F)}$;
- (2) $\|A\|_{(F)} \neq \sqrt{\text{trace}(A^H A)} = \sqrt{\sum |a_{ij}|^2}$.

Therefore, it is not a natural generality of Frobenius norm for complex matrices, but it is enough to measure the proximity of two split quaternion matrices.

Let $A \in \mathbf{SQ}^{m \times n}$, $E \in \mathbf{SQ}^{m \times l}$. If

$$AX = E \quad (3)$$

has the solution $X \in \mathbf{SQ}^{n \times l}$, then we have

$$A^R X^R = E^R$$

and therefore real linear equation

$$A^R Y = E^R (1 : 4m, 1 : l) \quad (4)$$

has the solution $Y \in \mathbf{R}^{4n \times l}$.

On the other hand, if (4) has a solution $Y \in \mathbf{R}^{4n \times l}$, then we have

$$\begin{aligned}A^R Q_n Y &= Q_m A^R Q_n^T Q_n Y = Q_m E^R (1 : 4m, l), \\ A^R R_n Y &= R_m A^R R_n^T R_n Y = R_m E^R (1 : 4m, l), \\ A^R S_n Y &= S_m A^R S_n^T S_n Y = S_m E^R (1 : 4m, l),\end{aligned}$$

and then

$$\begin{aligned} &A^R(Y, Q_n Y, R_n Y, S_n Y) \\ &= (E^R(1 : 4m, l), Q_m E^R(1 : 4m, l), R_m E^R(1 : 4m, l), S_m E^R(1 : 4m, l)) \\ &= C^R. \end{aligned}$$

From Theorem 2.3, we know that $(Y, Q_n Y, R_n Y, S_n Y)$ is the real representation matrix of a split quaternion matrix (marked as X), and obtain $AX = E$.

In conclusion, we have the following result.

Theorem 3.1 *Let $A \in \mathbf{SQ}^{m \times n}$, $E \in \mathbf{SQ}^{m \times l}$. Then the equation (3) has a solution in $X \in \mathbf{SQ}^{n \times l}$ if and only if the real linear equation (4) has a solution in $\mathbf{R}^{4n \times l}$. And if (4) has the solution $Y \in \mathbf{R}^{4n \times l}$, then*

$$(I_n, I_n i, I_n j, I_n k)Y$$

is the solution of (3).

If the equation (3) has no solution, on the basis of the above norm definition, we want to find a matrix $X \in \mathbf{SQ}^{n \times l}$ such that

$$\|AX - E\|_{(F)} = \min. \tag{5}$$

This problem is called the split quaternionic least squares (SQLS) problem.

On the same time, we construct the following real least squares problem

$$\|A^R Y - E^R\|_F = \min, \tag{6}$$

with unknown real matrix $Y \in \mathbf{R}^{4n \times 4l}$.

In [4], Z. Zhang studied the SQLS problem (5) through the real LS problem (6), but here Y has special structure. In fact, Z. Zhang turned the SQLS problem into a real constrained least squares problem.

Because

$$\|AX - E\|_{(F)}^2 = 4\|(A^R X^R - E^R)(1 : 4m, 1 : l)\|_F^2,$$

if we find the solution \hat{Y} of the real unconstrained least squares problem

$$\|\hat{A}Y - \hat{E}\|_F = \min, \tag{7}$$

with $\hat{A} = A^R$, $\hat{E} = E^R(1 : 4m, 1 : l)$ and unknown real matrix $Y \in \mathbf{R}^{4n \times l}$, and let $(\hat{Y}, Q_m \hat{Y}, R_m \hat{Y}, S_m \hat{Y})$ be the real representation matrix of the split quaternion matrix \hat{X} , then \hat{X} is a solution of the SQLS problem (5).

3.1 LSQR Algorithm

First, we review the LSQR algorithm proposed by Paige and Saunders[9] for solving the following least squares problem:

$$\min_{x \in \mathbf{R}^n} \|Mx - f\|_2 \tag{8}$$

with given $M \in \mathbf{R}^{m \times n}$ and $f \in \mathbf{R}^m$, whose normal equation is

$$M^T Mx = M^T f. \tag{9}$$

Algorithm LSQR

(1) Initialization.

$$\beta_1 u_1 = f, \alpha_1 v_1 = M^T u_1, h_1 = v_1, x_0 = 0, \bar{\zeta}_1 = \beta_1, \bar{\rho}_1 = \alpha_1.$$

(2) Iteration. For $i = 1, 2, \dots$

(i) bidiagonalization

- (a) $\beta_{i+1}u_{i+1} = Mv_i - \alpha_i u_i$
 (b) $\alpha_{i+1}v_{i+1} = M^T u_{i+1} - \beta_{i+1}v_i$
 (ii) construct and use Givens rotation
 $\rho_i = \sqrt{\bar{\rho}_i^2 + \beta_{i+1}^2}$
 $c_i = \bar{\rho}_i/\rho_i, s_i = \beta_{i+1}/\rho_i, \theta_{i+1} = s_i \alpha_{i+1}$
 $\bar{\rho}_{i+1} = -c_i \alpha_{i+1}, \zeta_i = c_i \zeta_i, \zeta_{i+1} = s_i \zeta_i$
 (iii) update x and h
 $x_i = x_{i-1} + (\zeta_i/\rho_i)h_i$
 $h_{i+1} = v_{i+1} - (\theta_{i+1}/\rho_i)h_i$
 (iv) check convergence.
 We can choose

$$\|M^T(f - Mx_k)\|_2 = |\alpha_{k+1}\bar{\zeta}_{k+1}c_k| < \tau$$

as convergence criteria, where $\tau > 0$ is a small tolerance.

Theorem 3.2 ([9]) *The solution generated by Algorithm LSQR is the minimum norm solution of (8).*

3.2 LSQR Algorithm for the Split Quaternionic Least Squares Problem

We consider a more general SQLS problem. Here, we are given data matrices $A \in \mathbf{SQ}^{m \times n}, B \in \mathbf{SQ}^{p \times q}$, an observation matrix $E \in \mathbf{SQ}^{m \times q}$, and are asked to find a matrix $X \in \mathbf{SQ}^{n \times p}$ such that

$$\|AXB - E\|_{(F)} = \min. \quad (10)$$

For any $X \in \mathbf{R}^{4n \times 4p}$,

$$X = \begin{pmatrix} X_1 & -X_2 & X_3 & X_4 \\ X_2 & X_1 & X_4 & -X_3 \\ X_3 & X_4 & X_1 & -X_2 \\ X_4 & -X_3 & X_2 & X_1 \end{pmatrix} \in \mathbf{R}^{4n \times 4p},$$

define

$$\text{vec}_i(X) = \text{vec}(X(:, 1:p)).$$

It is easy to know

$$\text{vec}(X) = \mathcal{F}\text{vec}_i(X),$$

where

$$\mathcal{F} = \begin{pmatrix} \text{diag}(I_{4n}, \dots, I_{4n}) \\ \text{diag}(Q_n, \dots, Q_n) \\ \text{diag}(R_n, \dots, R_n) \\ \text{diag}(S_n, \dots, S_n) \end{pmatrix} \in \mathbf{R}^{16np \times 4np}.$$

It is easy to know that \mathcal{F} is of full column rank, and

$$\mathcal{F}^T \mathcal{F} = 4I_{4np}, \mathcal{F}^T = \mathcal{F}^T \mathcal{F} \mathcal{F}^\dagger.$$

Because

$$\begin{aligned} \|AXB - E\|_{(F)}^2 &= \frac{1}{4} \|A^R X^R B^R - E^R\|_F^2 \\ &= \frac{1}{4} \|(B^{R^T} \otimes A^R) \text{vec}(X^R) - \text{vec}(E^R)\|_2^2 \\ &= \frac{1}{4} \|(B^{R^T} \otimes A^R) \mathcal{F} \text{vec}_i(X^R) - \text{vec}(E^R)\|_2^2, \end{aligned}$$

where $M \otimes N$ denote the Kronecker product of matrices M and N , the SQLS problem (10) is equivalent to

$$\min_{x \in \mathbf{R}^{4np}} \|Mx - f\|_2 \quad (11)$$

with

$$M = (B^{R^T} \otimes A^R)\mathcal{F} \in R^{16mq \times 4np}, f = \text{vec}(E^R) \in R^{16mq}.$$

Notice that the solution x of (11) corresponds to $\text{vec}_i(X^R)$, where X is the solution of (10).

Now, we will apply Algorithm LSQR to (11). The vector iteration of LSQR will be rewritten into matrix form so that the Kronecker product and the \mathcal{F} can be released. To this end, it is required to transform the matrix-vector products of Mv and $M^T u$ back to a matrix-matrix form for variant vectors $v \in \mathbf{R}^{4np}$ and $u = \text{vec}(U) \in \mathbf{R}^{16mq}$, where $U \in \mathbf{Q}_R^{m \times q}$. Notice that we do not want to construct the matrix M explicitly.

Let $\text{mat}(a)$ represent the matrix form of a vector a . For any $v \in \mathbf{R}^{4np}$ and $u = \text{vec}(U) \in \mathbf{R}^{16mq}$, where $U \in \mathbf{Q}_R^{m \times q}$. Let

$$\begin{aligned} \tilde{V} &= \text{mat}(v) = \text{vec}^{-1}(v) \in R^{4n \times p}, \\ V &= (\tilde{V}, Q_n \tilde{V}, R_n \tilde{V}, S_n \tilde{V}) \in \mathbf{R}^{4n \times 4p}. \end{aligned}$$

Then we have

$$\begin{aligned} \text{mat}(Mv) &= \text{mat}((B^{R^T} \otimes A^R)\mathcal{F}v) = \text{mat}((B^{R^T} \otimes A^R)\mathcal{F}\text{vec}(\tilde{V})) \\ &= \text{mat}((B^{R^T} \otimes A^R)\mathcal{F}\text{vec}_i(V)) = \text{mat}((B^{R^T} \otimes A^R)\text{vec}(V)) \\ &= A^R V B^R, \\ \text{mat}(M^T u) &= \text{mat}(\mathcal{F}^T (B^R \otimes A^{R^T})u) = \text{mat}(\mathcal{F}^T (B^R \otimes A^{R^T})\text{vec}(U)) \\ &= \text{mat}(\mathcal{F}^T \text{vec}(A^{R^T} U B^{R^T})) = \text{mat}(\mathcal{F}^T \mathcal{F} \mathcal{F}^\dagger \text{vec}(A^{R^T} U B^{R^T})) \\ &= \text{mat}(4I_{4np} \mathcal{F}^\dagger \text{vec}(A^{R^T} U B^{R^T})) = \text{mat}(4I_{4np} \text{vec}_i(A^{R^T} U B^{R^T})) \\ &= Z(:, 1 : p), \end{aligned}$$

where

$$Z = 4A^{R^T} U B^{R^T} \in \mathbf{R}^{4n \times 4p}.$$

Therefore, we can get the following algorithm.

Algorithm 3.1 (Algorithm LSQR-SQ for (10))

(1) Initialization.

$$\begin{aligned} X_0 &= 0 \in R^{4n \times p}, \beta_1 = \|E^R\|_F, U_1 = E^R/\beta_1, Z_1 = 4A^{R^T} U_1 B^{R^T}, \\ \tilde{V}_1 &= Z_1(:, 1 : p), \alpha_1 = \|\tilde{V}_1\|_F, \bar{V}_1 = \tilde{V}_1/\alpha_1, V_1 = (\tilde{V}_1, Q_n \tilde{V}_1, R_n \tilde{V}_1, S_n \tilde{V}_1), \\ H_1 &= \tilde{V}_1, \zeta_1 = \beta_1, \bar{\rho}_1 = \alpha_1. \end{aligned}$$

(2) Iteration. For $i = 1, 2, \dots$

(i) bidiagonalization

$$\begin{aligned} (a) \bar{U}_{i+1} &= A^R V_i B^R - \alpha_i U_i, \beta_{i+1} = \|\bar{U}_{i+1}\|_F, U_{i+1} = \bar{U}_{i+1}/\beta_{i+1}, \\ (b) Z_{i+1} &= 4A^{R^T} U_{i+1} B^{R^T}, \tilde{V}_{i+1} = Z_{i+1}(:, 1 : p) - \beta_{i+1} \tilde{V}_i, \\ \alpha_{i+1} &= \|\tilde{V}_{i+1}\|_F, \bar{V}_{i+1} = \tilde{V}_{i+1}/\alpha_{i+1}, V_{i+1} = (\tilde{V}_{i+1}, Q_n \tilde{V}_{i+1}, R_n \tilde{V}_{i+1}, S_n \tilde{V}_{i+1}). \end{aligned}$$

(ii) construct and use Givens rotation

$$\begin{aligned} \rho_i &= \sqrt{\bar{\rho}_i^2 + \beta_{i+1}^2}, c_i = \bar{\rho}_i/\rho_i, s_i = \beta_{i+1}/\rho_i, \theta_{i+1} = s_i \alpha_{i+1}, \\ \bar{\rho}_{i+1} &= -c_i \alpha_{i+1}, \zeta_i = c_i \zeta_i, \zeta_{i+1} = s_i \zeta_i. \end{aligned}$$

(iii) update X and H

$$X_i = X_{i-1} + (\zeta_i/\rho_i)H_i, H_{i+1} = \tilde{V}_{i+1} - (\theta_{i+1}/\rho_i)H_i.$$

(iv) check convergence. Output

$$X = X_i(1 : n, :) + X_i(n + 1 : 2n, :)i + X_i(2n + 1 : 3n, :)j + X_i(3n + 1 : 4n, :)k.$$

Algorithm 3.1 can find the minimum norm solution $x = \text{vec}_i(X^R)$ of (11), that is,

$$\|\text{vec}_i(X^R)\|_2 = \min.$$

Again,

$$\|X\|_{(F)}^2 = \frac{1}{4} \|X^R\|_F^2 = \|\text{vec}_i(X^R)\|_2^2,$$

so we have the following result.

Theorem 3.3 *The solution generated by Algorithm 3.1 is the minimum norm solution of the SMLS problem (10).*

Making the best of the structure of U_i and V_i , we can simplify Algorithm 3.1 to reduce operation amount and storage space greatly.

At last, we directly give the simplified algorithm for the SMLS problem (5).

Algorithm 3.2 (Algorithm LSQR-SQ for (5))

(1) Initialization.

$$\begin{aligned} X_j^{(0)} &= 0 \in R^{n \times l}, j = 1, 2, 3, 4. \beta^{(1)} = \sqrt{\sum_{j=1}^4 \|E_j\|_F^2}, \\ U_j^{(1)} &= E_j / \beta^{(1)}, j = 1, 2, 3, 4. \\ \bar{V}_1^{(1)} &= A_1^T U_1^{(1)} + A_2^T U_2^{(1)} + A_3^T U_3^{(1)} + A_4^T U_4^{(1)}, \\ \bar{V}_2^{(1)} &= -A_2^T U_1^{(1)} + A_1^T U_2^{(1)} + A_4^T U_3^{(1)} - A_3^T U_4^{(1)}, \\ \bar{V}_3^{(1)} &= A_3^T U_1^{(1)} + A_4^T U_2^{(1)} + A_1^T U_3^{(1)} + A_2^T U_4^{(1)}, \\ \bar{V}_4^{(1)} &= A_4^T U_1^{(1)} - A_3^T U_2^{(1)} - A_2^T U_3^{(1)} + A_1^T U_4^{(1)}, \\ \alpha^{(1)} &= \sqrt{\sum_{j=1}^4 \|\bar{V}_j^{(1)}\|_F^2}, V_j^{(1)} = \bar{V}_j^{(1)} / \alpha^{(1)}, H_j^{(1)} = V_j^{(1)}, j = 1, 2, 3, 4. \\ \bar{\zeta}^{(1)} &= \beta^{(1)}, \bar{\rho}^{(1)} = \alpha^{(1)}. \end{aligned}$$

(2) Iteration. For $i = 1, 2, \dots$

(i) bidiagonalization

$$\begin{aligned} (a) \bar{U}_1^{(i+1)} &= A_1 V_1^{(i)} - A_2 V_2^{(i)} + A_3 V_3^{(i)} + A_4 V_4^{(i)} - \alpha_i U_1^{(i)}, \\ \bar{U}_2^{(i+1)} &= A_2 V_1^{(i)} + A_1 V_2^{(i)} + A_4 V_3^{(i)} - A_3 V_4^{(i)} - \alpha_i U_2^{(i)}, \\ \bar{U}_3^{(i+1)} &= A_3 V_1^{(i)} + A_4 V_2^{(i)} + A_1 V_3^{(i)} - A_2 V_4^{(i)} - \alpha_i U_3^{(i)}, \\ \bar{U}_4^{(i+1)} &= A_4 V_1^{(i)} - A_3 V_2^{(i)} - A_2 V_3^{(i)} + A_1 V_4^{(i)} - \alpha_i U_4^{(i)}, \\ \beta^{i+1} &= \sqrt{\sum_{j=1}^4 \|\bar{U}_j^{(i+1)}\|_F^2}, U_j^{(i+1)} = E_j / \beta^{(i+1)}, j = 1, 2, 3, 4. \\ (b) \bar{V}_1^{(i+1)} &= A_1^T U_1^{(i+1)} + A_2^T U_2^{(i+1)} + A_3^T U_3^{(i+1)} + A_4^T U_4^{(i+1)} - \beta^{(i+1)} V_1^{(i)}, \\ \bar{V}_2^{(i+1)} &= -A_2^T U_1^{(i+1)} + A_1^T U_2^{(i+1)} + A_4^T U_3^{(i+1)} - A_3^T U_4^{(i+1)} - \beta^{(i+1)} V_2^{(i)}, \\ \bar{V}_3^{(i+1)} &= A_3^T U_1^{(i+1)} + A_4^T U_2^{(i+1)} + A_1^T U_3^{(i+1)} + A_2^T U_4^{(i+1)} - \beta^{(i+1)} V_3^{(i)}, \\ \bar{V}_4^{(i+1)} &= A_4^T U_1^{(i+1)} - A_3^T U_2^{(i+1)} - A_2^T U_3^{(i+1)} + A_1^T U_4^{(i+1)} - \beta^{(i+1)} V_4^{(i)}, \\ \alpha^{i+1} &= \sqrt{\sum_{j=1}^4 \|\bar{V}_j^{(i+1)}\|_F^2}, V_j^{(i+1)} = \bar{V}_j^{(i+1)} / \alpha^{(i+1)}, j = 1, 2, 3, 4. \end{aligned}$$

(ii) construct and use Givens rotation

$$\begin{aligned} \rho^{(i)} &= \sqrt{(\bar{\rho}^{(i)})^2 + \beta^{(i+1)^2}}, c^{(i)} = \bar{\rho}^{(i)} / \rho^{(i)}, s^{(i)} = \beta^{(i+1)} / \rho^{(i)}, \theta^{(i+1)} = s^{(i)} \alpha^{(i+1)}, \\ \bar{\rho}^{(i+1)} &= -c^{(i)} \alpha^{(i+1)}, \zeta^{(i)} = c^{(i)} \bar{\zeta}^{(i)}, \bar{\zeta}^{(i+1)} = s^{(i)} \bar{\zeta}^{(i)}. \end{aligned}$$

(iii) update X and H

$$X_j^{(i)} = X_j^{(i-1)} + (\zeta^{(i)} / \rho^{(i)}) H_j^{(i)}, H_j^{(i+1)} = V_j^{(i+1)} - (\theta^{(i+1)} / \rho^{(i)}) H_j^{(i)}, j = 1, 2, 3, 4.$$

(iv) check convergence. Output

$$X = X_1^{(i)} + X_2^{(i)} i + X_3^{(i)} j + X_4^{(i)} k.$$

Example 3.1. Let

$$A = \begin{pmatrix} 1+j & 2+k \\ -i & i+j \\ -1+i & k \end{pmatrix}, E = \begin{pmatrix} 1-k & 1+i \\ -j & j+k \\ i+j & 1+k \end{pmatrix}.$$

In [4], the authors got the solution $Y \in \mathbf{R}^{8 \times 8}$ by solving the real matrix

$$(A^R)^T A^R Y = (A^R)^T E^R,$$

and then obtained the unique solution X of (5) as follows.

$$X = \begin{pmatrix} 0.5698 - 0.3184i - 0.2179j - 0.8547k & -0.2849 - 0.0279i - 0.1229j - 0.2793k \\ 0.0978 - 0.1788i - 0.0838j - 0.0810k & 0.6006 - 0.0112i + 0.3436j + 0.0503k \end{pmatrix}.$$

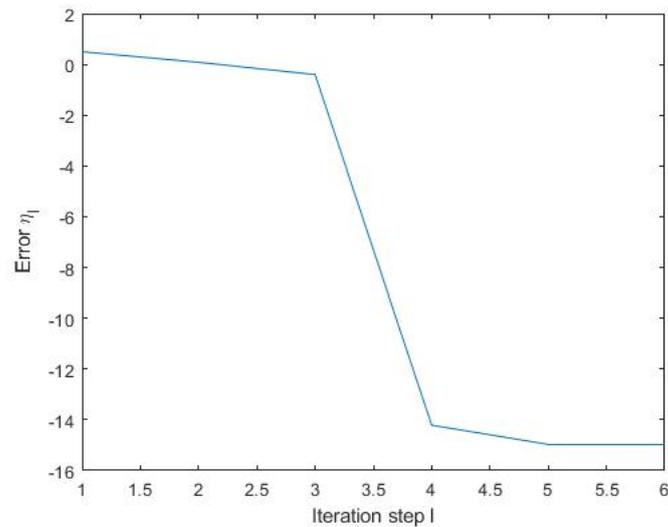


Figure 1. The relation between error and iteration step.

By our Algorithm 3.2, we also get the above result. Let $\eta_l = \log_{10}(\|M^T M x_l - M^T f\|_F)$ denotes the normal equation error after the l th iteration. Figure 1 give the relation between the normal equation error η_l and iteration step l , and show that Algorithm 3.2 is efficient.

4 Conclusion

In this paper, we define a real presentation of the split quaternion matrix and study its properties in detail, give an alternative of the split quaternion norm, which enable us to define and study the split quaternionic least squares (SQLS) problem. By the real presentation and the classical LSQR algorithm, we derive two iterative methods for finding solutions of the SQLS problem in split quaternionic mechanics. Due to our limited programming capabilities, the performance of current algorithms is not very efficient. We think that these algorithms can be further optimized to improve efficiency. This is also one of the directions we will study in the future.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (11771188).

References

1. Y. Alagoz, K.H. Oral and S. Yuce, "Split quaternion matrices," *Miskolc Math. Notes*, vol. 13, pp. 223–232, 2012.
2. D.C. Brody and E.M. Graefe, "On complexified mechanics and coquaternions," *J. Phys. A: Math. Theor.*, vol. 44, pp. 072001, 2012.
3. N. Moiseyev, *TNon-Hermitian Quantum Mechanics*. Cambridge University Press, 2011.
4. Z. Zhang, Z. Jiang and T. Jiang, "Algebraic methods for least squares problem in split quaternionic mechanics," *Applied Mathematics and Computation*, vol. 269, pp. 603–626, 2015.
5. T. Jiang, Z. Zhang and Z. Jiang, "Algebraic techniques for eigenvalues and eigenvectors of a split quaternion matrix in split quaternionic mechanics," *Computer Physics Communications*, vol. 229, pp. 1–7, 2018.
6. T. Jiang, Z. Zhang and Z. Jiang, "Algebraic techniques for Schrodinger equations in split quaternionic mechanics," *Computers and Mathematics with Applications*, vol. 75, pp. 2217–2222, 2018.
7. M.H. Wang, "An iterative algorithm for least squares problem in quaternionic quantum theory," *Comput. Phys. Comm.*, vol. 179, pp. 203–207, 2008.

8. S.T. Ling, Z.G. Jia and T.S. Jiang, “LSQR algorithm with structured preconditioner for the least squares problem in quaternionic quantum theory,” *Computers and Mathematics with Applications*, vol. 73, pp. 425–433, 2017.
9. C.C. Paige and A. Saunders, “LSQR: An algorithm for sparse linear equations and sparse least squares,” *ACM Trans. Math. Software*, vol. 8, pp. 43–71, 1982.