J. Appl. Math. & Informatics Vol. **34***(2016), No. 1 - 2, pp. 95 - 106*
http://dx.doi.org/10.14317/jami.2016.095

ON THE PURE IMAGINARY QUATERNIONIC LEAST SQUARES SOLUTIONS OF MATRIX EQUATION[†]

MINGHUI WANG AND JUNTAO ZHANG*

Abstract. In this paper, according to the classical LSQR algorithm for solving least squares (LS) problem, an iterative method is proposed for finding the minimum-norm pure imaginary solution of the quaternionic least squares (QLS) problem. By means of real representation of quaternion matrix, the QLS's correspongding vector algorithm is rewrited back to the matrix-form algorthm without Kronecker product and long vectors. Finally, numerical examples are reported that show the favorable numerical properties of the method.

AMS Mathematics Subject Classification : 65H05, 65F10.
Key words and phrases : Quaternion matrix, least squares problem, Algorithm LSQR, iterative method.

1. Introduction

Let $\boldsymbol{R}$, $\boldsymbol{Q}=\boldsymbol{R}+\boldsymbol{R}i+\boldsymbol{R}j+\boldsymbol{R}k$ and $\boldsymbol{IQ}^{m\times n}$ denote the real number field, the quaternion field and the set of all $m\times n$ pure imaginary quaternion matrices, respectively, where $i^2=j^2=k^2=-1$, $ij=-ji=k$. For any $x=x_1+x_2i+x_3j+x_4k\in\boldsymbol{Q}$, the conjugate of quaternion x is $\bar{x}=x_1-x_2i-x_3j-x_4k$.

Let $\boldsymbol{F}^{m\times n}$ denotes the set of $m\times n$ matrices on F. For any $A\in\boldsymbol{F}^{m\times n}$, A^T, $\bar{A}$ and A^H present the transpose, conjugate 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.

For any $A=(a_1,...,a_n)\in\boldsymbol{F}^{m\times n}$, define $\mathrm{vec}(A)=(a_1^T,...,a_n^T)^T$. The inverse mapping of $\mathrm{vec}(\cdot)$ from R^{mn} to $R^{m\times n}$ which is denoted by $\mathrm{mat}(\cdot)$ satisfies $\mathrm{mat}(\mathrm{vec}(A))=A$.

Received March 23, 2015. Revised June 26, 2015. Accepted June 29, 2015. *Corresponding author. †This work is supported by the National Natural Science Foundation of China (Grant No:11001144), the Research Award Fund for outstanding young scientists of Shandong Province in China (BS2012DX009) and the Science and Technology Program of Shandong Universities of China (J11LA04).

Quaternions and quaternion matrices have many applications in quaternionic quantum mechanics and field theory. Based on the study of [5], we also discuss the quaternion matrix equation

$$AX = B \tag{1}$$

where A and B are given matrices of suitable size defined over the quaternion field. In this paper, we will derive an operable iterative method for finding the minimum-norm pure imaginary solution of the QLS problem, which is more appropriate to large scale system.

Many people have studied the matrix equation (1) and others constrained matrix equation, see [1, 2, 12, 13, 14, etc.]. For the real, complex and quaternion matrix equations, there are many results, see [3, 4, 5, 6, 7, 8, 9, 10, etc.].

In [5], the least squares pure imaginary solution with the least norm was given of the quaternion matrix equation (1) by using the complex representation of quaternion matrix and the Moore-Penrose. For $A = A_1 + A_2 j \in Q^{s\times m}$, $B = B_1 + B_2 j \in Q^{s\times n}$, let $Q = \begin{pmatrix} iA_1 & -A_2 & iA_2 \\ -iA_2 & A_1 & iA_1 \end{pmatrix}$, $Q_1 = Re(Q)$, $Q_2 = Im(Q)$, $E_1 = \begin{pmatrix} Re(B_1) & Re(B_2) & Im(B_1) & Im(B_2) \end{pmatrix}^T$ and $R_1 = (I_{3m} - Q_1^+ Q_1)Q_2^T$,

$$\begin{aligned} H_1 &= R_1^+ + (I_{2s} - R_1^+ R_1) Z_1 Q_2 Q_1^+ {Q_1^+}^T (I_{3m} - Q_2^T R_1^+), \\ Z_1 &= (I_{2s} + (I_{2s} - R_1^+ R_1) Q_2 Q_1^+ {Q_1^+}^T Q_2^T (I_{2s} - R_1^+ R_1))^{-1}. \end{aligned}$$

And the set of solution J_L is expressed as

$$J_L = \left\{ X \;\middle|\; \begin{pmatrix} Im(X_1) \\ Re(X_2) \\ Im(X_2) \end{pmatrix} = (Q_1^+ - H_1^T Q_2 Q_1^+, H_1^T) E_1 + (I_{3m} - Q_1^+ Q_1 - R_1 R^+) Y \right\}$$

where Y is an arbitrary matrix of appropriate size. However, the method is not easy to realize in large scale system which motivated us to find an operable iterative method. Also Au-Yeung and Cheng in [6] studied the pure imaginary quaternionic solutions of the Hurwitz matrix equations.

Firstly, let us review the real least squares problem. In the LS problem, given $A \in \boldsymbol{R}^{m\times n}$ and $B \in \boldsymbol{R}^{n\times p}$ for finding a real matrix X so that

$$\min = \|AX - B\|_F, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. And the unique minimum-norm solution of the LS problem given by

$$X_{LS} = A^\dagger B,$$

where $A^\dagger$ denotes the Moore-Penrose of A.

2. Preliminary

For any $A = A_1 + A_2 i + A_3 j + A_4 k \in \boldsymbol{Q}^{m\times n}$ and $A_l \in \boldsymbol{R}^{m\times n}(l = 1, 2, 3, 4)$, define

$$A^R = \begin{pmatrix} A_1 & -A_2 & -A_3 & -A_4 \\ A_2 & A_1 & -A_4 & A_3 \\ A_3 & A_4 & A_1 & -A_2 \\ A_4 & -A_3 & A_2 & A_1 \end{pmatrix} \in \boldsymbol{R}^{4m\times 4n}. \tag{3}$$

The real matrix A^R is known as real representation of the quaternion matrix A. The set of all matrices shaped as (3) is denoted by $\boldsymbol{Q}_R^{m\times n}$. Obviously, the relation between $\boldsymbol{Q}^{m\times n}$ and $\boldsymbol{Q}_R^{m\times n}$ is one-to-one correspondence.

Let

$$P_t = \begin{pmatrix} I_t & 0 & 0 & 0 \\ 0 & -I_t & 0 & 0 \\ 0 & 0 & I_t & 0 \\ 0 & 0 & 0 & -I_t \end{pmatrix}, Q_t = \begin{pmatrix} 0 & -I_t & 0 & 0 \\ I_t & 0 & 0 & 0 \\ 0 & 0 & 0 & I_t \\ 0 & 0 & -I_t & 0 \end{pmatrix},$$

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

Then P_t, Q_t, R_t, S_t are unitary matrices, and by the definition of real representation, we can obtain the following results which given by T. Jang [4] and M. Wang [8].

Proposition 2.1. *Let $A, B \in \boldsymbol{Q}^{m\times n}$, $C \in \boldsymbol{Q}^{n\times s}$, $\alpha \in \boldsymbol{R}$. Then*

(a) $(A + B)^R = A^R + B^R,\ (\alpha A)^R = \alpha A^R,\ (AC)^R = A^R C^R$;

(b) $Q_m^2 = R_m^2 = S_m^2 = -I_{4m},\ Q_m^T = -Q_m,\ R_m^T = -R_m,\ S_m^T = -S_m$;

(c) $R_m Q_m = S_m,\ Q_m S_m = R_m,\ S_m R_m = Q_m$;

(d) $Q_m R_m = S_m^T,\ S_m Q_m = R_m^T,\ R_m S_m = Q_m^T$;

(e) $Q_m^T A^R Q_n = Q_m A^R Q_n^T = A^R,\ R_m^T A^R R_n = R_m A^R R_n^T = A^R,$
$S_m^T A^R S_n = S_m A^R S_n^T = A^R.$

Remark 2.1. Form above property (a), we know that the mapping $\boldsymbol{Q}^{m\times n} \to \boldsymbol{Q}_R^{m\times n}$ is an isomorphism.

Theorem 2.2. *For any $V \in \boldsymbol{R}^{4m\times n}$, $(V, Q_m V, R_m V, S_m V)$ is a real representation matrix of some quaternion matrix.*

Based on the definition of quaternion matrix norm in [8], which denoted by $\|\cdot\|_{(F)}$ can be proved a natural generality of Frobenius norm for complex matrices, it has the following properties:

$$(1)\quad \|A\|_{(F)} = 1/2\|A^R\|_F;$$

$$(2) \quad \|AB\|_{(F)} \leq \|A\|_{(F)} \|B\|_{(F)};$$
$$(3) \quad \|A\|_{(F)} = \sqrt{\sum |a_{ij}|^2}.$$

Then we review the LSQR algorithm proposed in [11] for solving the following LS problem:

$$\min_{x \in R^n} \|Mx - f\|_2 \tag{4}$$

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

$$M^T M x = M^T f. \tag{5}$$

The algorithm is summarized as follows.

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 $i = 1, 2, \cdots$
(i) bidiagonalization
(a) $\beta_{i+1} u_{i+1} = M v_i - \alpha_i u_i$
(b) $\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i$
(ii) construct and use Givens rotation
$\rho_i = \sqrt{\bar{\rho}_i^2 + \beta_{i+1}^2}$
$c_i = \bar{\rho}_i / \rho_i, s_i = \beta_{i+1} / \rho_i, \theta_{i+1} = s_i \alpha_{i+1}$
$\bar{\rho}_{i+1} = -c_i \alpha_{i+1}, \zeta_i = c_i \bar{\zeta}_i, \bar{\zeta}_{i+1} = s_i \bar{\zeta}_i$
(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

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

as convergence criteria, where $\tau > 0$ is a small tolerance. Obviously, there is no storage requirement for all the vector v_i and u_i.

And we can easily obtain the following theorem that if linear equation (5) has a solution $x^* \in R(M^T M) \in R(M^T)$, then x^* generated by Algorithm LSQR is the minimum norm solution of (4). So we can have the solution generated by Algorithm LSQR is the minimum-norm solution of problem (4). Specifically, it was shown in [11] that this method is numerically more reliable even if M is ill-conditioned.

3. The matrix-form LSQR method for QLS problem

In this section, we give the definition of quaternionic least squares (QLS) problem on the basis of quaternion matrix norm which is shown in section 2, for

$$\min_{X \in \boldsymbol{Q}^{n\times p}} \|AX - B\|_{(F)} \tag{6}$$

with given matrices $A \in \boldsymbol{Q}^{m\times n}$ and $B \in \boldsymbol{Q}^{m\times p}$. Then we can find problem (6) is equivalent to

$$\min_{X \in \boldsymbol{Q}_R^{n\times p}} \|A^R X - B^R\|_F \tag{7}$$

which is a constrained LS problem with given matrices $A^R \in \boldsymbol{Q}_R^{m\times n}$ and $B^R \in \boldsymbol{Q}_R^{m\times p}$.

Next, we will deduce the iterative method to find the pure imaginary quaternionic solution of the QLS problem (1). For any $X \in \boldsymbol{IQ}_R^{n\times p}$,

$$X = \begin{pmatrix} 0 & -X_2 & -X_3 & -X_4 \\ X_2 & 0 & -X_4 & X_3 \\ X_3 & X_4 & 0 & -X_2 \\ X_4 & -X_3 & X_2 & 0 \end{pmatrix} \in \boldsymbol{R}^{4n\times 4p},$$

define

$$\mathrm{vec}_i(X) = \mathrm{vec}(X(n{+}1{:}4n, 1{:}p)) = \mathrm{vec}\begin{pmatrix} X_2 \\ X_3 \\ X_4 \end{pmatrix}.$$

Obviously, there is an one to one linear mapping from the long-vector space $\mathrm{vec}(\boldsymbol{R}^{4n\times 4p})$ to the independent parameter space $\mathrm{vec}_i(\boldsymbol{R}^{3n\times p})$. Let $\mathcal{F}$ denote the pure imaginary quaternionic constrained matrix which defines linear mapping from $\mathrm{vec}_i(\boldsymbol{R}^{3n\times p})$ to $\mathrm{vec}(\boldsymbol{R}^{4n\times 4p})$, that is

$$\mathrm{vec}(X) = \mathcal{F}\mathrm{vec}_i(X), \;\; X \in \boldsymbol{R}^{4n\times 4p}.$$

Theorem 3.1. *Suppose $\mathcal{F}$ is a pure imaginary quaternionic constrained matrix, then*

$$\mathcal{F} = \mathcal{T}\begin{pmatrix} O_{n\times 3np} \\ (I_{3n}, O_{3n\times 3n(p-1)}) \\ O_{n\times 3np} \\ (O_{3n\times 3n}, I_{3n}, O_{3n\times 3n(p-2)}) \\ \cdots \\ O_{n\times 3np} \\ (O_{3n\times 3n(p-1)}, I_{3n}) \end{pmatrix} \in R^{16np\times 3np} \quad \textit{and} \quad \mathcal{F}^T\mathcal{F}\mathcal{F}^\dagger = \mathcal{F}^T,$$

where

$$\mathcal{T} = \begin{pmatrix} \mathrm{diag}(I_{4n}, ..., I_{4n}) \\ \mathrm{diag}(Q_{4n}, ..., Q_{4n}) \\ \mathrm{diag}(R_{4n}, ..., R_{4n}) \\ \mathrm{diag}(S_{4n}, ..., S_{4n}) \end{pmatrix} \in R^{16np\times 4np}.$$

Proof. First, we know

$$\operatorname{vec}(X) = \begin{pmatrix} \operatorname{vec}\begin{pmatrix} 0 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} \\ \operatorname{vec}\begin{pmatrix} -X_2 \\ 0 \\ X_4 \\ -X_3 \end{pmatrix} \\ \operatorname{vec}\begin{pmatrix} -X_3 \\ -X_4 \\ 0 \\ X_2 \end{pmatrix} \\ \operatorname{vec}\begin{pmatrix} -X_4 \\ X_3 \\ -X_2 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \operatorname{vec}\begin{pmatrix} 0 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} \\ \operatorname{vec}\left(Q_n \begin{pmatrix} 0 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix}\right) \\ \operatorname{vec}\left(R_n \begin{pmatrix} 0 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix}\right) \\ \operatorname{vec}\left(S_n \begin{pmatrix} 0 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix}\right) \end{pmatrix}$$

$$= \begin{pmatrix} I \\ I \otimes Q_n \\ I \otimes R_n \\ I \otimes S_n \end{pmatrix} \operatorname{vec}\begin{pmatrix} 0 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \mathcal{T}\operatorname{vec}\begin{pmatrix} 0 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix}$$

and

$$\operatorname{vec}\begin{pmatrix} 0 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} O_{n\times 3np} \\ (I_{3n}, O_{3n\times 3n(p-1)}) \\ O_{n\times 3np} \\ (O_{3n\times 3n}, I_{3n}, O_{3n\times 3n(p-2)}) \\ \cdots \\ O_{n\times 3np} \\ (O_{3n\times 3n(p-1)}, I_{3n}) \end{pmatrix} \operatorname{vec}_i(X).$$

Hence, we have

$$\operatorname{vec}(X) = \mathcal{T} \begin{pmatrix} O_{n\times 3np} \\ (I_{3n}, O_{3n\times 3n(p-1)}) \\ O_{n\times 3np} \\ (O_{3n\times 3n}, I_{3n}, O_{3n\times 3n(p-2)}) \\ \cdots \\ O_{n\times 3np} \\ (O_{3n\times 3n(p-1)}, I_{3n}) \end{pmatrix}_{4np\times 3np} \operatorname{vec}_i(X).$$

Therefore, let

$$\mathcal{F}=\mathcal{T}\left(\begin{array}{c} O_{n\times 3np} \\ (I_{3n}, O_{3n\times 3n(p-1)}) \\ O_{n\times 3np} \\ (O_{3n\times 3n}, I_{3n}, O_{3n\times 3n(p-2)}) \\ \cdots \\ O_{n\times 3np} \\ (O_{3n\times 3n(p-1)}, I_{3n}) \end{array}\right) \in R^{16np\times 3np},$$

and from the above we have

$$\mathrm{vec}(X)=\mathcal{F}\mathrm{vec}_i(X).$$

Then because of

$$\begin{aligned} \mathcal{F}^T\mathcal{F} &= \left(O_{3np\times n}, \left(\begin{array}{c} I_{3n} \\ O_{3n(p-1)\times 3n} \end{array}\right), O_{3np\times n}, \left(\begin{array}{c} O_{3n\times 3n} \\ I_{3n} \\ O_{3n(p-2)\times 3n} \end{array}\right), \ldots, O_{3np\times n}, \right. \\ &\left. \left(\begin{array}{c} O_{3n(p-1)\times 3n} \\ I_{3n} \end{array}\right) \right) \cdot \mathcal{T}^T\mathcal{T} \cdot \left(\begin{array}{c} O_{n\times 3np} \\ (I_{3n}, O_{3n\times 3n(p-1)}) \\ O_{n\times 3np} \\ (O_{3n\times 3n}, I_{3n}, O_{3n\times 3n(p-2)}) \\ \cdots \\ O_{n\times 3np} \\ (O_{3n\times 3n(p-1)}, I_{3n}) \end{array}\right) \\ &= 4\left[\left(\begin{array}{cc} I_{3n} & O_{3n\times 3n(p-1)} \\ O_{3n(p-1)\times 3n} & O_{3n(p-1)\times 3n(p-1)} \end{array}\right) + \left(\begin{array}{ccc} O & O & O \\ O & I_{3n} & O \\ O & O & O \end{array}\right) + \right. \\ &\left. \left(\begin{array}{cc} I_{3n} & O_{3n\times 3n(p-1)} \\ O_{3n(p-1)\times 3n} & O_{3n(p-1)\times 3n(p-1)} \end{array}\right)\right] \\ &= 4I_{3np}, \end{aligned}$$

we can know that $\mathcal{F}$ is of full column rank and

$$(\mathcal{F}^T\mathcal{F}\mathcal{F}^{\dagger})^T=(\mathcal{F}\mathcal{F}^{\dagger})^T(\mathcal{F}^T)^T=\mathcal{F}\mathcal{F}^{\dagger}\mathcal{F}=\mathcal{F},$$

that is

$$\mathcal{F}^T\mathcal{F}\mathcal{F}^{\dagger}=\mathcal{F}^T.$$

□

Because

$$\begin{aligned} \|AX-B\|_{(F)}^2 &= \frac{1}{4}\|A^RX^R-B^R\|_F^2 \\ &= \frac{1}{4}\|(I\otimes A^R)\mathrm{vec}(X^R)-\mathrm{vec}(B^R)\|_2^2 \\ &= \frac{1}{4}\|(I\otimes A^R)\mathcal{F}\mathrm{vec}_i(X^R)-\mathrm{vec}(B^R)\|_2^2, \end{aligned}$$

where $M\otimes N$ denote the Kronecker product of matrices M and N, the QLS problem (6) is equivalent to

$$\min_{x\in R^{3np}}\|Mx-f\|_2 \tag{8}$$

with

$$M = (I_{4p} \otimes A^R)\mathcal{F} \in R^{16mp\times 3np},$$
$$f = \mathrm{vec}(B^R) \in R^{16mp}. \tag{9}$$

Now, we will apply Algorithm LSQR to problem (8) and the vector iteration of it will be transformed into matrix form so that the Kronecker product and $\mathcal{F}$ can be released. Then we transform the matrix-vector product of Mv and $M^T u$ back to a matrix-matrix form so as to let vector v and u be matrix V and U respectively, which required in Algorithm LSQR.

Let $\mathrm{mat}(\alpha)$ represent the matrix form of a vector α, For any $v \in \boldsymbol{R}^{3np}$ and $u = \mathrm{vec}(U) \in \boldsymbol{R}^{16mp}$, where $U \in \boldsymbol{Q}_R^{m\times p}$. Let

$$\tilde{V} = \mathrm{mat}(v) = \mathrm{vec}^{-1}(v) \in \boldsymbol{R}^{3n\times p}, \bar{\tilde{V}} = \begin{pmatrix} O_{n\times p} \\ \tilde{V} \end{pmatrix},$$
$$V = (\bar{\tilde{V}}, Q_n\bar{\tilde{V}}, R_n\bar{\tilde{V}}, S_n\bar{\tilde{V}}) \in \boldsymbol{Q}_R^{n\times p}.$$

Then we have

$$\begin{aligned}
\mathrm{mat}(Mv) &= \mathrm{mat}((I \otimes A^R)\mathcal{F}v) \\
&= \mathrm{mat}((I \otimes A^R)\mathcal{F}\mathrm{vec}(\tilde{V})) \\
&= \mathrm{mat}((I \otimes A^R)\mathcal{F}\mathrm{vec}_i(V)) \\
&= \mathrm{mat}((I \otimes A^R)\mathrm{vec}(V)) \\
&= A^R V, \\
\mathrm{mat}(M^T u) &= \mathrm{mat}\big(\mathcal{F}^T (I \otimes A^{R^T})u\big) \\
&= \mathrm{mat}\big(\mathcal{F}^T (I \otimes A^{R^T})\mathrm{vec}(U)\big) \\
&= \mathrm{mat}\big(\mathcal{F}^T \mathcal{F}\mathcal{F}^{\dagger}\mathrm{vec}(A^{R^T}U)\big) \\
&= \mathrm{mat}\big(4I_{3np}\mathrm{vec}_i(A^{R^T}U)\big) \\
&= Z(n+1:4n, 1:p)
\end{aligned}$$

where

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

Therefore, we can get the following algorithm.

Algorithm LSQR-P.

(1) Initialization

$X_0 = O \in \boldsymbol{R}^{3n\times p},\ \beta_1 = \|B^R\|,\ U_1 = B^R/\beta_1,$

$Z_1 = 4A^{R^T}U_1,\ \bar{V}_1 = Z_1(n{+}1{:}4n, 1{:}p),$

$\alpha_1 = \|\bar{V}_1\|_F,\ \tilde{V}_1 = \bar{V}_1/\alpha_1,\ \bar{\tilde{V}}_1 = \begin{pmatrix} O_{n\times p} \\ \tilde{V}_1 \end{pmatrix},$

$V_1 = (\bar{\tilde{V}}_1,\ Q_n\bar{\tilde{V}}_1,\ R_n\bar{\tilde{V}}_1,\ S_n\bar{\tilde{V}}_1),$

$H_1 = \tilde{V}_1,\ \bar{\zeta}_1 = \beta_1,\ \bar{\rho}_1 = \alpha_1.$

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

(i) bidiagonalization

(a) $\bar{U}_{i+1} = A^R V_i - \alpha_i U_i,$

$\beta_{i+1} = \|\bar{U}_{i+1}\|_F,\ U_{i+1} = \bar{U}_{i+1}/\beta_{i+1};$

(b) $Z_{i+1} = 4{A^R}^T U_{i+1},$

$\bar{V}_{i+1} = Z_{i+1}(n{+}1{:}4n, 1{:}p) - \beta_{i+1}\bar{V}_i,$

$\alpha_{i+1} = \|\bar{V}_{i+1}\|_F,\ \tilde{V}_{i+1} = \bar{V}_{i+1}/\alpha_{i+1},\ \bar{\tilde{V}}_{i+1} = \begin{pmatrix} O_{n\times p} \\ \tilde{V}_{i+1} \end{pmatrix},$

$V_{i+1} = (\bar{\tilde{V}}_{i+1},\ Q_n\bar{\tilde{V}}_{i+1},\ R_n\bar{\tilde{V}}_{i+1},\ S_n\bar{\tilde{V}}_{i+1});$

(ii) construct and use Givens rotation

$\rho_i = \sqrt{\bar{\rho}_i^2 + \beta_{i+1}^2},$

$c_i = \bar{\rho}_i/\rho_i,\ s_i = \beta_{i+1}/\rho_i,\ \theta_{i+1} = s_i\alpha_{i+1},$

$\bar{\rho}_{i+1} = -c_i\alpha_{i+1},\ \zeta_i = c_i\bar{\zeta}_i,\ \bar{\zeta}_{i+1} = s_i\bar{\zeta}_i;$

(iii) update X and H

$X_i = X_{i-1} + (\zeta_i/\rho_i)H_i,$

$H_{i+1} = \tilde{V}_{i+1} - (\theta_{i+1}/\rho_i)H_i;$

(3) check convergence. Output

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

Algorithm LSQR-P can compute the minimum-norm solution $x = \mathrm{vec}_i(X^R)$ of (8), that is

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

Again,

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

so we have the following result.

Theorem 3.2. *The solution generated by Algorithm LSQR-P is the minimum-norm solution of problem (6).*

4. Numerical examples

In this section, we give three examples to illustrate the efficiency and investigate the performance of Algorithm LSQR-P which shown to be numerically reliable in various circumstances. All functions are defined by Matlab 7.0.

Example 4.1. Given $[m, n, p] = N$, $A = A_1 + A_2 i + A_3 j + A_4 k$, $X = X_1 + X_2 i + X_3 j + X_4 k$, $B = AX$, with A_1, A_2, A_3, A_4 defined by $rand(m, n)$ respectively. Given $X_1 = zeros(n, p)$ and X_2, X_3, X_4 defined by $rand(n, p)$ respectively. Then Fig. 4.1 plots the relation between error $\varepsilon_k = log10(\|AX - B\|_{(F)})$ and iteration number K.

Notice that in the above case, the equation $AX = B$ is consistent and has a unique solution. From Fig. 4.1 we find our algorithm is effective.

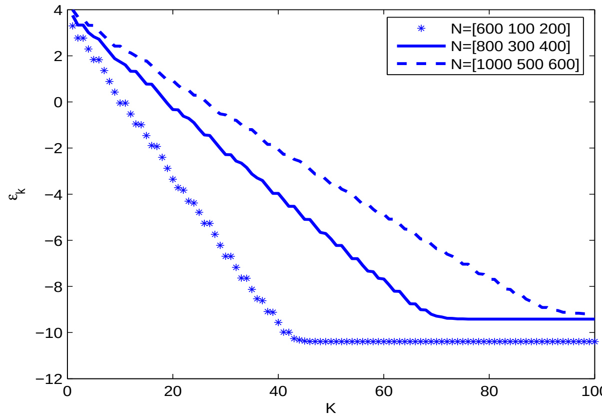

Fig. 4.1 The relation between error ε_k and iterative number K with different N

Example 4.2. Given $[m, n, p] = N$, $A = A_1 + A_2 i + A_3 j + A_4 k$, $B = B_1 + B_2 i + B_3 j + B_4 k$, with $A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4$ defined by $rand(m, n)$ respectively. Let $\eta_k = log10(\|M^T(Mx - f)\|_2)$ where M, f defined by (9). Then Fig. 4.2 plots the relation between error η_k and iteration number K.

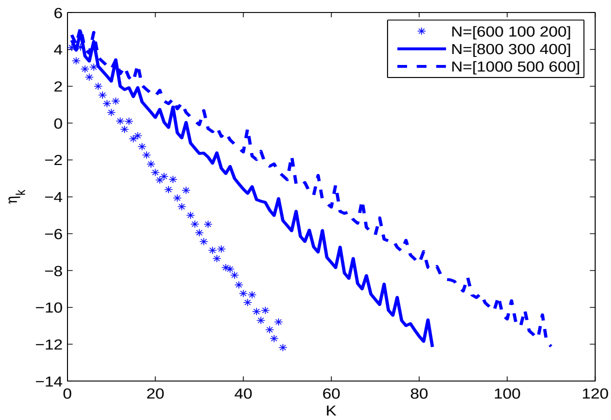

Fig. 4.2 The relation between error η_k and iterative number K with different N

Notice that in the above case, the equation $AX = B$ is not consistent and we use $\eta_k = \|M^T(f - Mx_k)\|_2 = |\alpha_{k+1}\bar{\zeta}_{k+1}c_k| < \tau = 10^{-12}$ as convergence criteria. From Fig. 4.2, we also find our algorithm work well.

Example 4.3. Given $m = n = p = 10$, $A = A_1 + A_2 i + A_3 j + A_4 k$, $X = X_1 + X_2 i + X_3 j + X_4 k$, $B = AX$, with $A_1 = hilb(m)$, $A_2 = pascal(m)$, $A_3 =$

$ones(m,n)$, $A_4 = pascal(m)$. Given $X_1 = zeros(n,p)$ and X_2, X_3, X_4 defined by $rand(n,p)$ respectively. In this case, the condition number of M is 3.9927×10^9, therefore this system is ill-conditioned. Then Fig. 4.3 plots the relation between error $\varepsilon_k = log10(\|AX - B\|_{(F)})$, $\eta_k = log10(\|X - X_k\|_F/\|X\|_F)$ and iteration number K.

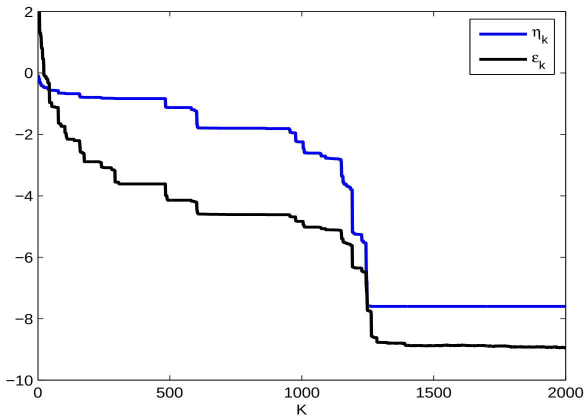

Fig. 4.3 The relation between error η_k, ε_k and iterative number K

Notice that the equation (1) is consistent and has a unique solution. The algorithm performance is not very well when the system very ill-conditioned. From Fig. 4.3 we find our algorithm is also effective.

References

1. L. Wu and B. Cain, *The re-nonnegative definite solutions to the matrix inverse problem* $AX = B$, Linear Algebra Appl. **236** (1996), 137-146.
2. C.J. Meng, X.Y. Hu and L. Zhang, *The skew symmetric orthogonal solution of the matrix equation* $AX = B$, Linear Algebra Appl. **402** (2005), 303-318.
3. Zhigang Jia, Musheng Wei and Sitao Ling, *A new structure-preserving method for quaternion Hermitian eigenvalue problems,* Journal of Comput. and Appl. Math. **239** (2013), 12-24.
4. T. Jang and L. Chen, *Algebraic algorithm for least squares problem in quaternionic quantum theory,* Comput. Phys. Comm. **176** (2007), 481-485.
5. Shifang Yuan, Qingwen Wang and Xuefeng Duan, *On solutions of the quaternion matrix equation* $AX = B$ *and their application in color image restoration,* Appl. Math. Comput. **221** (2013), 10-20.
6. Y.H. Au-Yeung and C.M. Cheng, *On the pure imaginary quaternionic solutions of the Hurwitz matrix equations,* Linear Algebra Appl. **419** (2006), 630-642.
7. Q.W. Wang and C.K. Li, *Ranks and the least-norm of the general solution to a system of quaternion matrix equation,* Linear Algebra Appl. **430** (2009), 1626-1640.
8. M.H. Wang, M.S. Wei and Y. Feng, *An iterative algorithm for least squares problem in quaternionic quantum theory,* Comput. Phys. Comm. **179** (2008), 203-207.

9. T. Jang, *Algebraic mehtods for diagonalization of a quaternion matrix in quaternionic quantum theory,* J. Math. Phys. **46** (2005), 052106.
10. M. Wang, *On positive definite solutions of quaternionic matrix equations,* World Academy of Science Engineering and Technology, **37** (2010), 535-537.
11. C.C. Paige and A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares,* Appl. Math. Comput. **8** (1982), 43-71.
12. F. Toutounian and 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.
13. Sitao Ling, *Hermitian tridiagonal solution with the least norm to quaternionic least squares problem,* Comput. Phys. Comm. **181** (2010), 481-488.
14. Z.Y. Peng, *A matrix LSQR iterative method to solve matrix equation $AXB = C$,* International Journal of Computer Mathematics **87** (2010), 1820-1830.

Minghui Wang Ph.D

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

Juntao Zhang M.D.

Department of Mathematics, Qingdao University of Science and Technology, Qingdao 266061, P.R. China.
e-mail: qustzhangjuntao@163.com