An Iterative Method for Quaternionic Linear Equations

Bin Yu, Minghui Wang, Juntao Zhang

Abstract—By the real representation of the quaternionic matrix, an iterative method for quaternionic linear equations \(Ax = b\) is proposed. Then the convergence conditions are obtained. At last, a numerical example is given to illustrate the efficiency of this method.

Keywords—Quaternionic linear equations, Real representation, Iterative algorithm.

I. INTRODUCTION

In quaternionic quantum mechanics and some other applications of quaternions\([1], [2], [3]\), the problem of solutions of quaternionic linear equations is often encountered. Because of noncommutativity of quaternions, solving quaternionic linear equations is difficult. In papers\([4], [5], [6], [7]\), by means of a complex representation and a companion vector, the authors have studied quaternionic linear equations and presented a Cramer rule for quaternionic linear equations and an algebraic algorithm for the least squares problem, respectively, in quaternionic quantum theory. In the paper\([8]\), by using the complex representation of quaternion matrices, and the Moore-Penrose generalized inverse, the authors derive the expressions of the least squares solution with the least norm, the least squares pure imaginary solution with the least norm, and the least squares real solution with the least norm for the quaternion matrix equation \(AX = B\), respectively.

In the paper \([9]\) and \([10]\), by means of a real representation of the quaternionic matrix, we gave an iterative algorithms for the least squares problem in quaternionic quantum theory, and the relation between the positive (semi)definite solutions of quaternionic matrix equations and those of corresponding real matrix equations, respectively.

In this paper, we will pay attention to quaternionic linear equations \(Ax = b\) by means of the real representation, and propose an iterative method, which is more suitable in the large-scale systems.

Let \(\mathbf{R}\) denote the real number field, \(\mathbf{Q} = \mathbf{R} \oplus \mathbf{R}i \oplus \mathbf{R}j \oplus \mathbf{R}k\) the quaternion field, where

\[i^2 = j^2 = k^2 = -1, \quad ij = -ji = k.\]

B. Yu, M. Wang and J. Zhang are with the Department of Mathematics, Qingdao University of Science & Technology, Qingdao 266061, China, e-mail: (mhwang@yeah.net).

This work is supported by the Science and Technology Program of Shandong Universities of China (J11LA04) and the Research Award Fund for Outstanding Young Scientists of Shandong Province in China(BS2012DX009).

II. MAIN RESULTS

In this section, we will give an iterative method for

\[Ax = b,\]

where \(A = A_1 + A_2i + A_3j + A_4k \in \mathbf{Q}^{n \times n}\), \(b = b_1 + b_2i + b_3j + b_4k \in \mathbf{Q}^n\), and \(x = x_1 + x_2i + x_3j + x_4k \in \mathbf{Q}^n\). \(A\) and \(A_1\) are nonsingular. Then, we will discuss the convergence conditions for this iterative method.

The real representation equation of (2) is

\[A^R x^R = b^R,\]

that is,

\[
\begin{pmatrix}
A_1 & -A_2 & -A_3 & -A_4 \\
A_2 & A_1 & -A_3 & -A_4 \\
A_3 & A_4 & A_1 & -A_2 \\
A_4 & -A_3 & A_2 & A_1
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{pmatrix}
=
\begin{pmatrix}
x_1 - x_2 - x_3 - x_4 \\
x_2 x_1 - x_4 x_3 \\
x_3 x_4 - x_1 x_2 \\
x_4 - x_3 x_2 - x_1
\end{pmatrix},
\]

which may be written as

\[
\begin{pmatrix}
A_1 & 0 & -A_3 & -A_4 \\
0 & -A_2 & -A_3 & A_4 \\
A_3 & A_4 & 0 & -A_2 \\
A_4 & -A_3 & A_2 & 0
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{pmatrix}
=
\begin{pmatrix}
x_1 - x_2 - x_3 - x_4 \\
x_2 x_1 - x_4 x_3 \\
x_3 x_4 - x_1 x_2 \\
x_4 - x_3 x_2 - x_1
\end{pmatrix},
\]

Let \(A_l \in \mathbf{R}^{m \times n}\), \(l = 1, 2, 3, 4\). The real representation matrix is defined\([7]\) in the form

\[A^R \equiv \begin{pmatrix}
A_1 & -A_2 & -A_3 & -A_4 \\
A_2 & A_1 & -A_3 & -A_4 \\
A_3 & A_4 & A_1 & -A_2 \\
A_4 & -A_3 & A_2 & A_1
\end{pmatrix} \in \mathbf{R}^{4m \times 4n}.\]
It follows from Proposition 1 that the above formula is equivalent to
\[
\begin{align*}
A_1 x_1 &= A_2 x_2 + A_3 x_3 + A_4 x_4 + b_1 \\
A_2 x_2 &= -A_2 x_1 + A_3 x_3 - A_4 x_4 + b_2 \\
A_3 x_3 &= -A_3 x_1 - A_2 x_2 + A_4 x_4 + b_3 \\
A_4 x_4 &= -A_4 x_1 - A_2 x_2 - A_3 x_3 + b_4
\end{align*}
\]
So we can construct the iterative algorithm as follows:

**Algorithm 1.** the algorithm for \(Ax = b\)

(I). Initialization. Given arbitrary \(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}\);

(II). Iteration. For \(l = 1, 2, \ldots\)
\[
\begin{align*}
x_1^{(l+1)} &= A_1^{-1}(A_2 x_2^{(l)} + A_3 x_3^{(l)} + A_4 x_4^{(l)} + b_1) \\
x_2^{(l+1)} &= -A_1^{-1}(A_2 x_2^{(l)} + A_3 x_3^{(l)} - A_4 x_4^{(l)} + b_2) \\
x_3^{(l+1)} &= -A_1^{-1}(A_3 x_1^{(l)} - A_2 x_2^{(l)} + A_4 x_4^{(l)} + b_3) \\
x_4^{(l+1)} &= -A_1^{-1}(A_4 x_1^{(l)} + A_3 x_3^{(l)} - A_2 x_2^{(l)} + b_4)
\end{align*}
\]

(III). Check convergence.

**Theorem 2.** If \(|A_1^{-1}(A_i)| < 1, i = 1, 2, 3, 4\), then the sequence \(\{x_1^{(l)}, x_2^{(l)}, x_3^{(l)}, x_4^{(l)}\}\), generated by Algorithm 1, converges to \((x_1, x_2, x_3, x_4)\), where
\[
x = x_1^{(*)} + x_2^{(*)}i + x_3^{(*)}j + x_4^{(*)}k
\]
is the solution of (2).

**Proof.** We can easily obtain
\[
x_1^{(l+1)} - x_1^{(*)} = A_1^{-1}(A_2 x_2^{(l)} - x_2^{(*)}) + A_3 (x_3^{(l)} - x_3^{(*)}) + A_4 (x_4^{(l)} - x_4^{(*)})
\]
}\[
\sum_{i=1}^{3} f_i(A_1^{-1} A_2, A_1^{-1} A_3, A_1^{-1} A_4) (x_1^{(l)} - x_1^{(*)}),
\]
where \(f_i(x, y, z)\) is a \((l+1)\)-order homogeneous polynomial on \(x, y, z\).

If \(|A_1^{-1}(A_i)| < 1, i = 1, 2, 3, 4\), it is obvious that
\[
x_i^{(l)} \to x_i^{(*)}.
\]
Similarly, \(x_i^{(l)} \to x_i^{(*)}, i = 2, 3, 4\).

Since the iterative matrix of Algorithm 1 is
\[
B = \begin{pmatrix}
0 & -A_1^{-1} A_2 & -A_1^{-1} A_3 & -A_1^{-1} A_4 \\
A_1^{-1} A_2 & 0 & -A_1^{-1} A_4 & A_1^{-1} A_3 \\
A_1^{-1} A_3 & A_1^{-1} A_4 & 0 & -A_1^{-1} A_2 \\
A_1^{-1} A_4 & A_1^{-1} A_3 & A_1^{-1} A_2 & 0
\end{pmatrix},
\]
we can get the following result.

**Theorem 3.** If and only if \(\rho(B) < 1\), the sequence \(\{(x_1^{(l)}, x_2^{(l)}, x_3^{(l)}, x_4^{(l)})\}\), generated by Algorithm 1, converges to \((x_1, x_2, x_3, x_4)\), where \(x = x_1^{(*)} + x_2^{(*)}i + x_3^{(*)}j + x_4^{(*)}k\) is the solution of (2).

Because \(\rho(B) \leq \|B\|_\infty\), we have

**III. Numerical Example**

In this section, we present a numerical example to illustrate the efficiency of our algorithm.

Given \(A = A_1 + A_2 i + A_3 j + A_4 k, x = x_1 + x_2 i + x_3 j + x_4 k\) with
\[
A_1 = \begin{pmatrix}
9 & 12 & -37 & 6 \\
-8 & 0 & 19 & -7 \\
17 & 43 & -19 & 0 \\
78 & -98 & 0 & 12
\end{pmatrix}, \quad A_2 = \begin{pmatrix}
1 & 0 & 19 & -7 \\
0 & 1 & 0 & 0 \\
1 & 0 & 9 & 12 \\
0 & 10 & 1 & -12
\end{pmatrix}
\]
\[
x_1 = (1, 1, 3, -4)^T, x_2 = (0, -7, 8, 11)^T, x_3 = (-8, 14, 20, 3)^T, x_4 = (32, 14, 0, -17)^T, A_1 = 10A_1, b = b_1 + b_2 j + b_3 k \equiv Ax.
\]

And \(A_1\) are nonsingular, \(Ax = b\) has a unique solution \(x\).

Fig. 1. The error of the computed solutions by Algorithm 1

Fig. 1 depicts the relation of the \(k\)-step approximate
\[
x^{(l)} = x_1^{(l)} + x_2^{(l)} i + x_3^{(l)} j + x_4^{(l)} k
\]
and the true solution \(x\). From Fig. 1, we can see that Algorithm 1 is efficient for this example.

**References**

