Article ID 1672-6634(2018)01-0021-08　　**DOI** 10.19728/j.issn1672-6634.2018.01.004

Real Structure-preserving Algorithms for the Quaternion Cholesky Decomposition: Revisit ①

LI Ying[1]　WEI Mu-sheng[1,2]　ZHANG Feng-xia[1]　ZHAO Jian-li[1]

(1. School of Mathematical Sciences, Liaocheng University, Liaocheng 252059, China;
2. School of Mathematics and Science, Shanghai Normal University, Shanghai 200234, China)

Abstrat In a paper published in 2013[6], Wang and Ma proposed a structure-preserving algorithm for computing the quaternion Cholesky decomposition. In this paper, we study the quaternion Cholesky decomposition carefully and re-propose real structure-preserving algorithms for LDL^H and LL^H decompositions on quaternion Hermitian positive definite matrices, in which we make full use of high-level operations. We compared these real structure-preserving algorithms with the structure-preserving algorithm proposed by Wang and Ma[6] and Quaternion Toolbox for Matlab in terms of computational time and accuracy. Numerical experiments are provided to demonstrate that the qLDLH1 and qChol proposed in this paper are more efficient than algorithm of Wang and Ma using low-level operations and that in QTFM using quaternion arithmetics.

Key words Quaternion matrix; LDL^H; Cholesky decomposition; real representation; real structure-preserving algorithm.

CLC number O151.21　　**Document code** A

1 Introduction

In a paper published in 2013[6], Wang and Ma proposed a structure-preserving algorithm for the quaternion Cholesky decomposition. We study the structure-preserving algorithm in [6], and find that the computations are based on 'level-1' operations[1], i.e., element to element operations. In fact, we can design more efficient algorithms to solve the problem of quaternion Cholesky decomposition.

One way to quantify the volume of work associated with a computation is to count flops. Usually $fl(x \text{ op } y)$ stands for a arithmetic of $+,-,\times$, or $\div$ of x and y. For example, for $a,b\in\mathbf{R}$, there is only one real flop in $a\pm b$ and $a\times b$, while for $a,b\in\mathbf{Q}$, there are 4 real flops in $a\pm b$ and 28 real flops in $a\times b$. For $\boldsymbol{a},\boldsymbol{b}\in\mathbf{R}^n$, there is only $2n$ real flops in $\boldsymbol{a}^T\boldsymbol{b}$, while for $\boldsymbol{a},\boldsymbol{b}\in\mathbf{Q}^n$, there are $32n$ real flops in $\boldsymbol{a}^H\boldsymbol{b}$, 16 times of real flop of that in computing inner product of real vectors.

An algorithm should satisfy the following three basic requirements. First, the algorithm should be numerical stable and so is reliable. Second, computational speed should be as fast as possible. The last,

① 收稿日期:2017-11-30
基金项目:国家自然科学基金项目(11301247,11171226);山东省教育厅科技项目(J15LI10)资助
通讯作者:李莹,女,汉族,博士,教授,研究方向:数值代数,E-mail: liyingld@163.com.

it should save storage space.

In the real matrix computations, there are many stable and high accurate algorithms. Most of them can be generalized to quaternion matrix computations and they are also stable and reliable.

The computational speed of an algorithm by applying Matlab software depends on two issues: arithmetic flops and assignment number. The number of flops in a given matrix computation is usually obtained by summing the amount of arithmetic associated with the most deeply nested statements. To reduce the complexity of a algorithm, we should minimize the flops. The quaternion arithmetic is much more complicated compared with the real arithmetic. By using Theorem 1, we can convert the problem of quaternion matrix computations to that of its real representation matrix computations, and by applying Theorem 2 or Theorem 3, we can design real structure-preserving algorithms, which ensure that the numbers of real flops of the two methods are same. When it comes to design a high-performance matrix computation, it is not enough to minimize flops. Since the assignment number affects the complexity of an algorithm and so also affects computational speed. Assignment refer to call subroutines or perform matrix operations. An assignment typically requires several cycles to complete. The input scalars proceed along a computational assembly line, spending one cycle at each of all the work stations. Vector operation is a very regular sequence of scalar operation. Vector processors exploit the key idea of pipelining. With pipelining, the input vector are streamed through the operation unit. Once the pipeline is filled and steady state reached, an output component is produced every cycle. The rate of vector processing is about n time that of scalar processing, in which n is the number of cycles in a floating point operation. On the other hand, parallel algorithms using multiprocessor play a great role in improving efficiency of matrix computations, say, $\boldsymbol{B} = \boldsymbol{AX} + \boldsymbol{Y}$, we adopt the assignment $\boldsymbol{B} = \boldsymbol{A} * \boldsymbol{X} + \boldsymbol{Y}$ to utilize vector pipelining arithmetic operations rather than explicitly using triply-nested for-end loops, to speed up computations remarkably. Therefore, real arithmetic numbers as well as assignment numbers are important measures. See § 1.5 and § 1.6 of [1].

In this paper, we generalize the real $\boldsymbol{LDL}^{\mathrm{H}}$ and $\boldsymbol{LL}^{\mathrm{T}}$ decompositions and propose real structure-preserving $\boldsymbol{LDL}^{\mathrm{H}}$ and $\boldsymbol{LL}^{\mathrm{H}}$ decompositions on quaternion Hermitian positive definite matrices. We compare the running times and errors of these real structure-preserving algorithms with those of Wang and Ma, and quaternion Toolbox for Matlab (QTFM)[5]. Numerical experiments are provided to demonstrate the effectiveness of the real structure-preserving algorithms proposed in this paper.

In this paper, $\mathbf{R}$ and $\mathbf{Q}$ denote real number field and quaternion skew-field, respectively. $\mathbf{F}^{m\times n}$ denotes the set of all $m \times n$ matrices on $\mathbf{F}$. For any matrix $\boldsymbol{A} \in \mathbf{F}^{m\times n}$, $\boldsymbol{A}^{\mathrm{T}}$, $\overline{\boldsymbol{A}}$ and $\boldsymbol{A}^{\mathrm{H}}$ present the transpose, conjugate and conjugate transpose of $\boldsymbol{A}$, respectively. $\| \cdot \|$ denotes the Euclidean vector norm of a vector or the Frobenius norm of a matrix.

This paper is organized as follows. In Section 2, we recall some preliminary results used in the paper. In Section 3, we generalize the real $\boldsymbol{LDL}^{\mathrm{H}}$ and $\boldsymbol{LL}^{\mathrm{T}}$ decompositions and propose real structure-preserving $\boldsymbol{LDL}^{\mathrm{H}}$ and $\boldsymbol{LL}^{\mathrm{H}}$ decompositions on quaternion Hermitian positive definite matrices. In Section 4, numerical experiments are provided to demonstrate the effectiveness of these algorithms proposed in this paper. Finally in Section 5, we make some concluding remarks.

2 Preliminaries

In this section we recall some basic properties about quaternion and quaternion matrices. A quaternion $q \in \mathbf{Q}$ is expressed as

$$q = a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k},$$

in which a, b, c, $d \in \mathbf{R}$, and three imaginary units i, j, k satisfy

$$\mathrm{i}^2=\mathrm{j}^2=\mathrm{k}^2=\mathrm{ijk}=-1.$$

The quaternion skew-field $\mathbf{Q}$ is an associative but non-commutative algebra of rank four over $\mathbf{R}$, endowed with an involutory antiautomorphism

$$q \to \bar{q} = a - b\mathrm{i} - c\mathrm{j} - d\mathrm{k}.$$

Every non-zero quaternion is invertible, and the unique inverse is given by $q^{-1} = \frac{\bar{q}}{|q|^2}$, in which the quaternion norm $|q|$ is dened by

$$|q|^2 = \bar{q}q = a^2 + b^2 + c^2 + d^2.$$

For any quaternion matrix $\boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2\mathrm{i} + \boldsymbol{A}_3\mathrm{j} + \boldsymbol{A}_4\mathrm{k}$, in which $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{A}_4 \in \mathbf{R}^{m\times n}$, its real representation can be defined as follows,

$$\boldsymbol{A}^R \equiv \begin{bmatrix} \boldsymbol{A}_1 & -\boldsymbol{A}_2 & -\boldsymbol{A}_3 & -\boldsymbol{A}_4 \\ \boldsymbol{A}_2 & \boldsymbol{A}_1 & -\boldsymbol{A}_4 & \boldsymbol{A}_3 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 & \boldsymbol{A}_1 & -\boldsymbol{A}_2 \\ \boldsymbol{A}_4 & -\boldsymbol{A}_3 & \boldsymbol{A}_2 & \boldsymbol{A}_1 \end{bmatrix}.$$

For $\boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2\mathrm{i} + \boldsymbol{A}_3\mathrm{j} + \boldsymbol{A}_4\mathrm{k} \in \mathbf{Q}^{m\times n}$, we can get the real representation of $\boldsymbol{A}$ by the following algorithm.

Algorithm 1　A method for generating a $4m \times 4n$ real representation of matrix $\boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2\mathrm{i} + \boldsymbol{A}_3\mathrm{j} + \boldsymbol{A}_4\mathrm{k} \in \mathbf{Q}^{m\times n}$.

Function $\boldsymbol{A}^R = Realp(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{A}_4)$

$$\boldsymbol{A}^R \equiv \begin{bmatrix} \boldsymbol{A}_1 & -\boldsymbol{A}_2 & -\boldsymbol{A}_3 & -\boldsymbol{A}_4 \\ \boldsymbol{A}_2 & \boldsymbol{A}_1 & -\boldsymbol{A}_4 & \boldsymbol{A}_3 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 & \boldsymbol{A}_1 & -\boldsymbol{A}_2 \\ \boldsymbol{A}_4 & -\boldsymbol{A}_3 & \boldsymbol{A}_2 & \boldsymbol{A}_1 \end{bmatrix},$$

end

We list some properties of $\boldsymbol{A}^R$ as follows.

Theorem 1[2]　Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbf{Q}^{m\times n}$, $C \in \mathbf{Q}^{n\times s}$ and $a \in \mathbf{R}$. Then

(1) $(\boldsymbol{A}+\boldsymbol{B})^R = \boldsymbol{A}^R + \boldsymbol{B}^R$, $(\mathrm{a}\boldsymbol{A})^R = a\boldsymbol{A}^R$, $(\boldsymbol{AC})^R = \boldsymbol{A}^R\boldsymbol{C}^R$.

(2) $(\boldsymbol{A}^{\mathrm{H}})^R = (\boldsymbol{A}^R)^{\mathrm{T}}$.

(3) $\boldsymbol{A} \in \mathbf{Q}^{m\times m}$ is Hermitian positive definite if and only if $\boldsymbol{A}^R$ is symmetric positive definite.

From (1)　and Theorem 1, for the matrix $\boldsymbol{A}^R$, we only need to store the first column block of $\boldsymbol{A}^R$, denoted as $\boldsymbol{A}_c^R = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \boldsymbol{A}_3 \\ \boldsymbol{A}_4 \end{bmatrix}$, or the first row block of $\boldsymbol{A}^R$, denoted as $\boldsymbol{A}_r^R = [\boldsymbol{A}_1 \quad \boldsymbol{A}_2 \quad \boldsymbol{A}_3 \quad \boldsymbol{A}_4]$.

Theorem 2[2]　Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbf{Q}^{m\times n}$, $\boldsymbol{C} \in \mathbf{Q}^{n\times s}$, $\boldsymbol{q} \in \mathbf{Q}^m$ and $a \in \mathbf{R}$. Then

(1) $(\boldsymbol{A}+\boldsymbol{B})_c^R = \boldsymbol{A}_c^R + \boldsymbol{B}_c^R$, $(a\boldsymbol{A})_c^R = a\boldsymbol{A}_c^R$, $(\boldsymbol{AC})_c^R = \boldsymbol{A}^R\boldsymbol{C}_c^R$.

(2) $(\boldsymbol{A}^{\mathrm{H}})_c^R = [(\boldsymbol{A}^R)^{\mathrm{T}}]_c$.

(3) $\|\boldsymbol{A}\|_F = \|\boldsymbol{A}_r^R\|_F$, $\|\boldsymbol{q}\|_2 = \|\boldsymbol{q}_r^R\|_2$.

Theorem 3[2]　Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbf{Q}^{m\times n}$, $\boldsymbol{C} \in \mathbf{Q}^{n\times s}$, $\boldsymbol{q} \in \mathbf{Q}^m$ and $a \in \mathbf{R}$. Then

(1) $(\boldsymbol{A}+\boldsymbol{B})_r^R = \boldsymbol{A}_r^R + \boldsymbol{B}_r^R$, $(a\boldsymbol{A})_r^R = a\boldsymbol{A}_r^R$, $(\boldsymbol{AC})_r^R = \boldsymbol{A}_r^R\boldsymbol{C}^R$.

(2) $(\boldsymbol{A}^H)_r^R = [(\boldsymbol{A}^R)^{\mathrm{T}}]_r$.

(3) $\|\boldsymbol{A}\|_F = \|\boldsymbol{A}_r^R\|_F$, $\|\boldsymbol{q}\|_2 = \|\boldsymbol{q}_r^R\|_2$.

Therefore, if we want to perform the Cholesky decomposition of a $m \times m$ quaternion positive definite matrix $\boldsymbol{A}$, we can apply the properties of its real representation, $\boldsymbol{A}^R$, to make the computation workload, computational time and storage spaces greatly reduced.

3 Real structure-preserving algorithms for the quaternion LDL^H and Cholesky decompositions

In this section, we generalized the real $\boldsymbol{LDL}^H$ and $\boldsymbol{LL}^T$ decompositions to propose real structure-preserving $\boldsymbol{LDL}^H$ and $\boldsymbol{LL}^H$ decompositions on quaternion Hermitian positive definite matrices.

Theorem 4[6] For an Hermitian positive definite matrix $\boldsymbol{A} \in \mathbf{Q}^{m\times m}$, there exists a unique triangular matrix $\boldsymbol{L} \in \mathbf{Q}^{m\times m}$ with positive diagonal elements such that

$$\boldsymbol{A} = \boldsymbol{LL}^H.$$

The first algorithm we proposed is the real structure-preserving algorithm qLDLH1, which is based on the quaternion $\boldsymbol{LU}$ decomposition.

Because quaternion matrix $\boldsymbol{A} \in \mathbf{Q}^{m\times m}$ is Hermitian positive definition, the principal square submatrices $\boldsymbol{A}_i$, $(i = 1,\cdots,m-1)$ of $\boldsymbol{A}$ are all nonsingular, we have

$$\boldsymbol{A} = \boldsymbol{LU},$$

in which $\boldsymbol{L}$ is a unit lower triangular quaternion matrix and $\boldsymbol{U}$ is an upper triangular quaternion matrix and all the diagonal elements of $\boldsymbol{U}$ are positive. When $\boldsymbol{A} \in \mathbf{Q}^{m\times m}$ is Hermitian positive definite, it satisfies the condition that the principal square submatrices of $\boldsymbol{A}$ are all nonsingular.

Let $\boldsymbol{D}$ be a diagonal matrix whose diagonal elements are those of $\boldsymbol{U}$. Then $\boldsymbol{L}^H = \boldsymbol{D}^{-1}\boldsymbol{U}$, because $\boldsymbol{A}$ is Hermitian, and so $\boldsymbol{A} = \boldsymbol{LDL}^H$.

We take out the first column block of the real representation of $\boldsymbol{A}$, i. e. , $\boldsymbol{A}_c^R$, and implement $\boldsymbol{LDL}^H$ decomposition of $\boldsymbol{A}$ by means of executing the following real structure-preserving algorithm on $\boldsymbol{A}_c^R$.

Algorithm 2 (qLDLH1: The quaternion $\boldsymbol{LDL}^H$ decomposition) For a given Hermitian positive definite matrix $\boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2\mathrm{i} + \boldsymbol{A}_3\mathrm{j} + \boldsymbol{A}_4\mathrm{k} \in \mathbf{Q}^{m\times m}$, the input $\boldsymbol{AA}$ is the first column block of $\boldsymbol{A}$, i. e. $\boldsymbol{A}_c^R$, the output LL is the first column block of the lower triangular quaternion matrix $\boldsymbol{L}$, and $\boldsymbol{D}$ is the diagonal matrix such that $\boldsymbol{A} = \boldsymbol{LDL}^H$.

```
Function: [LL ,D] =qLDLH1(AA)
B = AA;
m = size(B,2);
for  k = 1:m - 1
     B ([k + 1:m,k + 1 + m:2 * m,k + 1 + 2 * m:3 * m,k + 1 + 3 * m:4 * m], k) =
       B([k + 1:m,k + 1 + m:2 * m,k + 1 + 2 * m:3 * m,k + 1 + 3 * m:4 * m],k) = B(k,k);
     B ([k + 1:m,k + 1 + m:2 * m,k + 1 + 2 * m:3 * m,k + 1 + 3 * m:4 * m],k + 1:m) =
       B([k + 1:m,k + 1 + m:2 * m,k + 1 + 2 * m:3 * m,k + 1 + 3 * m:4 * m],k + 1:m)
       - Realp(B(k + 1:m,k),B(k + 1 + m:2 * m,k),B(k + 1 + 2 * m:3 * m,k),B(k + 1 + 3 * m:4 * m,k))
       * B([k,k + m,k + 2 * m,k + 3 * m],k + 1:m);
end
LL = [tril(B(1:m,:), - 1) + eye(m); tril(B(m + 1:2 * m,:), - 1); tril(B(2 m + 1:3 * m,:),
    - 1); tril(B(3 * m + 1:4 * m,:), - 1)];
D = diag(diag(B(1:m,1:m)));
end
```

The second algorithm that we proposed is another $\boldsymbol{LDL}^H$ decomposition of $\boldsymbol{A}$, which is the generalization of the real $\boldsymbol{LDL}^T$ decomposition (P. 158, [1]).

Suppose that $\boldsymbol{A}=\boldsymbol{LDL}^{\mathrm{H}}$, and we know the first $j-1$ columns of $\boldsymbol{L}$, diagonal elements $d_1,\cdots,d_{j-1}$ for some j with $1\leqslant j\leqslant m$. From

$$\boldsymbol{A}(1:j,j)=L(1:j,1:j)v(1:j)$$

with

$$\boldsymbol{v}(1:j)=\begin{bmatrix} d_1\overline{\boldsymbol{L}}(j,1)\\ \vdots \\ d_{j-1}\overline{\boldsymbol{L}}(j,j-1)\\ d_j\end{bmatrix},$$

we have

$$\boldsymbol{v}(j)=d_j=\boldsymbol{A}(j,j)-\sum_{k=1}^{j-1}d_k\mid \boldsymbol{L}(j,k)\mid^2,$$

and the jth column of $\boldsymbol{L}$ satises

$$\boldsymbol{L}(j+1:m,\ j)\boldsymbol{v}(j)=\boldsymbol{A}(j+1:m,j)-\boldsymbol{L}(j+1:m,1:j-1)\boldsymbol{v}(1:j-1).$$

We implement $\boldsymbol{LDL}^{\mathrm{H}}$ decomposition of $\boldsymbol{A}$ by means of executing the following real structure-preserving algorithm, named qLDLH2, on $\boldsymbol{A}_c^R$.

Algorithm 3 (qLDLH2: The quaternion $\boldsymbol{LDL}^{\mathrm{H}}$ decomposition) For a given Hermitian positive definite matrix $\boldsymbol{A}=\boldsymbol{A}_1+\boldsymbol{A}_2\mathrm{i}+\boldsymbol{A}_3\mathrm{j}+\boldsymbol{A}_4\mathrm{k}\in\boldsymbol{Q}^{m\times m}$, the input $\boldsymbol{AA}$ is the first column block of $\boldsymbol{A}$, i. e. $\boldsymbol{A}_c^R$, the output $\boldsymbol{LL}$ is the first column block of the lower triangular quaternion matrix $\boldsymbol{L}$ and $\boldsymbol{D}$ is the diagonal matrix such that $\boldsymbol{A}=\boldsymbol{LDL}^{\mathrm{H}}$.

```
Function [LL,D] =qLDLH2( AA )
B = AA;
v = zeros(4 * m,1);
B([2:m,2 + m:2 * m,2 + 2 * m:3 * m,2 + 3 * m:4 * m],1) = …
B([2:m,2 + m:2 * m,2 + 2 * m:3 * m,2 + 3 * m:4 * m],1)/B(1,1);
for  j = 2:m
for  i = 1:j - 1
     v([i,i +m,i+2 * m,i+3 * m],1) = B(i,i) * [B(j,i);-B(j+m,i);-B(j+2 * m,i);-B(j
     + 3 * m, i)];
end
B(j,j) = B(j,j) - [B(j,1:j - 1), - B(j + m,1:j - 1), - B(j + 2 * m,1:j - 1),…
        - B(j + 3 * m,1:j - 1)] * v([1:j - 1,m + 1:j - 1 + m,2 * m + 1:j - 1 + 2 * m,
        3 * m + 1:j - 1 + 3 * m],1);
B([j + 1:m,j + 1 + m:2 * m,j + 1 + 2 * m:3 * m,j + 1 + 3 * m:4 * m],j) = …
(B([j + 1:m,j + 1 + m:2 * m,j + 1 + 2 * m:3 * m,j + 1 + 3 * m:4 * m],j) - Realp(B(j + 1:m,1:
j - 1),…
B(j + 1 + m:2 * m,1:j - 1),B(j + 1 + 2 * m:3 * m,1:j - 1),B(j + 1 + 3 * m:4 * m,1:j - 1))…
v([1:j - 1,m + 1:j - 1 + m,2 * m + 1:j - 1 + 2 * m,3 * m + 1:j - 1 + 3 * m],1))/B(j,j);
end
LL = [tril(B(1:m,:), - 1) + eye(m);tril(B(m + 1:2 * m,:), - 1);tril(B(2 * m + 1:3 * m;:), -
1);…
tril(B(3 * m + 1:4 * m;:), - 1)];
D = diag(diag(B(1:m,:)));
end
```

The third algorithm that we proposed is generalization of the real Cholesky decomposition, the Cholesky decomposition is a quaternion Cholesky decomposition. It can be derived from the partition

$$\boldsymbol{A}=\begin{bmatrix}\alpha & \boldsymbol{v}^{\mathrm{H}}\\ \boldsymbol{v} & \boldsymbol{B}\end{bmatrix}=\begin{bmatrix}\beta & 0\\ \dfrac{v}{\beta} & \boldsymbol{I}_{m-1}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \boldsymbol{B}-\dfrac{\boldsymbol{v}\boldsymbol{v}^{\mathrm{H}}}{\alpha}\end{bmatrix}\begin{bmatrix}\beta & \dfrac{\boldsymbol{v}^{\mathrm{H}}}{\beta}\\ 0 & \boldsymbol{I}_{m-1}\end{bmatrix}.$$

We can derive quaternion Cholesky decomposition by repeated application of (3.1). Now, we propose the following real structure-preserving algorithm qChol, of the quaternion Cholesky decomposition based on outer product.

Algorithm 4 (*qChol*: The quaternion Cholesky decomposition) For a given Hermitian positive definite matrix $\boldsymbol{A}=\boldsymbol{A}_1+\boldsymbol{A}_2\mathrm{i}+\boldsymbol{A}_3\mathrm{j}+\boldsymbol{A}_4\mathrm{k}\in\mathbf{Q}^{m\times m}$, the input $\boldsymbol{AA}$ is the first column block of $\boldsymbol{A}$, i. e. $\boldsymbol{A}_c^R$, the output $\boldsymbol{LL}$ is the first column block of the lower triangular quaternion matrix $\boldsymbol{L}$ such that $\boldsymbol{A}=\boldsymbol{LL}^{\mathrm{H}}$.

```
Function LL = qChol(AA)
B = AA;
for  k = 1:m
     B(k,k) = sqrt(B(k,k));
     B([k+1:m,k+1+m:2*m,k+1+2*m:3*m,k+1+3*m:4*m],k) = ...
     B([k+1:m,k+1+m:2*m,k+1+2*m:3*m,k+1+3*m:4*m],k)/B(k,k);
     C = Realp(B(k+1:m,k),B(k+1+m:2*m,k),B(k+1+2*m:3*m,k),B(k+1+3*m:
     4*m,k));
     Ct = C';
     B([k+1:m,k+1+m:2*m,k+1+2*m:3*m,k+1+3*m:4*m],k+1:m) = ...
     B([k+1:m,k+1+m:2*m,k+1+2*m:3*m,k+1+3*m:4*m],k+1:m)-C*Ct(:,
     1:m-k);
end
LL = zeros(4*m,m);
LL([2:m,2+m:2*m,2+2*m:3*m,2+3*m:4*m],1:m-1) = [tril(B(2:m,1:m-1));...
tril(B(m+2:2*m,1:m-1));tril(B(2*m+2:3*m,1:m-1));tril(B(3*m+2:4*m,1:m-
1))];
LL(1:m,:) = LL(1:m,:)+diag(diag(B(1:m,:)));
end
```

We list the real computational amounts and the assignment numbers of above algorithms for $\boldsymbol{A}\in\mathbf{Q}^{m\times m}$ in Table 1. In this Table, we can see that the real computational amount of qLDLH2 is half of qLDLH1, but the assignment number of qLDLH2 is much higher than qLDLH1 and qChol. Because in qLDLH1 and qChol, we can use high-level operations at each step of computations, while in qLDLH2, we can not compute $v(1:j,1)$ with high-level operations.

Table 1 Computation amounts and assignment numbers for qLDLH1, qLDLH2 and qChol

	realops	assignment numbers
qLDLH1	$\frac{32m^3}{3}$	$2m+2$
qLDLH2	$\frac{16m^3}{3}$	$\frac{m^2+3m}{2}$
qChol	$\frac{16m^3}{3}$	$\frac{2}{5m}$

4 Comparison of the real structure-preserving algorithms with QTFM for the quaternion Cholesky decomposition

In this section, we provide numerical examples to compare the effieiency of the above three real structure-preserving algorithms qLDLH1, qLDLH2, qChol with real structure-preserving algorithm in

[6](named RSP of [2]) and QTFM on array sizes of $10k \times 10k$ for $k = 1:50$. All these computations are performed on an Intel Core i7-2600 @ 3.40 GHz/8 GB computer. The version of Matlab used is R2013a.

Example For $a = 10; k = 1:50, m = ak$, we use the function 'rand' in Matlab to obtain four matrices whose elements are evenly distributed in interval (0,1), We use these matrices as real and three imaginary parts of a quaternion matrix $\boldsymbol{B}$, then $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}$ to guarantee $\boldsymbol{A}$ is Hermitian positive definite. We perform the quaternion Cholesky decomposition of $\boldsymbol{A}$ and compare the CPU times and errors of the real structure-preserving algorithms and QTFM for performing the quaternion Cholesky decomposition.

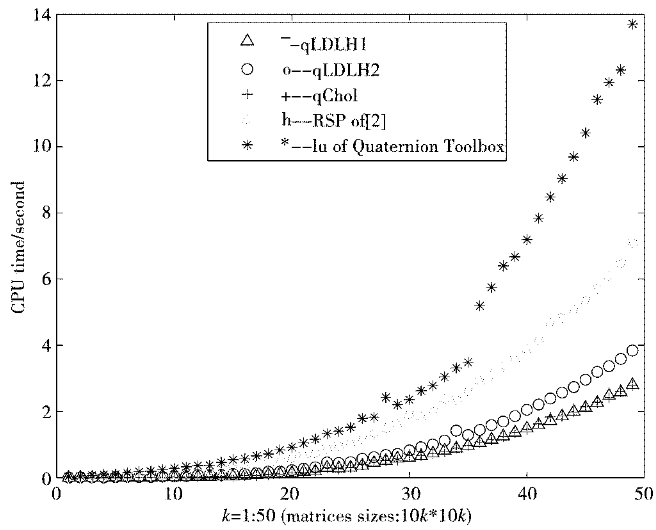

Fig. 1　CPU times for the Cholesky decomposition

From Fig. 1, we observe that, the CPU times of all the four real structure-preserving algorithms are superior to that of QTFM. When the sizes of the matrices are large, the CPU time of qLDLH1 is almost the same as that of qChol, which is about two third of that of qLDLH2, one third of that of RSP of [2] and one fifth of that of the function 'lu' of Quaternion toolbox, respectively. In Fig. 2, we observe that the Frobenius norm of $\boldsymbol{L}\boldsymbol{L}^{\mathrm{H}}$-$\boldsymbol{A}$ of qLDLH1, qChol, RSP of [2] and the function 'lu' of Quaternion toolbox are almost the same, while that of qLDLH2 is about twice of that in the other four algorithms. Therefore, both algorithms qLDLH1 and qChol are most efficient.

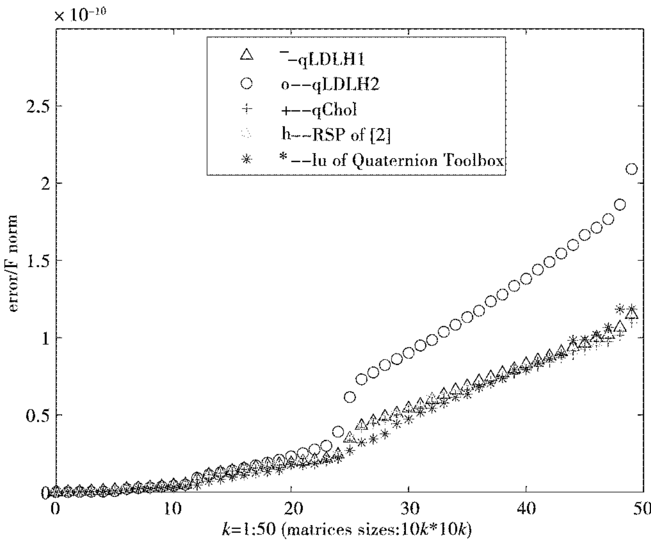

Fig. 2　errors(F norm) for the Cholesky decomposition

Remark We do not find the function of Cholesky decomposition in QTFM, the comparison in Example is performed between the real structure-preserving algorithms and the function 'lu' in QTFM.

5　Conclusions

In this paper, we generalized the real $\boldsymbol{LDL}^{\mathrm{H}}$ and $\boldsymbol{LL}^{\mathrm{T}}$ decompositions, and proposed real structure-preserving algorithms of $\boldsymbol{LDL}^{\mathrm{H}}$ and $\boldsymbol{LL}^{\mathrm{H}}$ decompositions for quaternion Hermitian positive definite matrices. We compared these real structure-preserving algorithms with the structure-preserving algorithm proposed by Wang and Ma[6] and Quaternion Toolbox for Matlab in terms of computational time and accuracy. Numerical experiments are provided to demonstrate that the qLDLH1 and qChol are most efficient.

References

[1] Golub G H, Van Loan C F. Matrix Computations[M]. 4th Edition. Baltimore: The Johns Hopkins University Press, 2013.

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

[3] Li Y, Wei M S, Zhang F X, et al. A fast structure-preserving method for computing the singular value decomposition of quaternion matrix[J]. Appl Math Comput, 2014,235:157-167.

[4] Li Y, Wei M S, Zhang F X, et al. Real structure-preserving algorithms of Householder based transformations for quaternion matrices [J]. J Comput Appl Math, 2016,305:82-91.

[5] Sangwine S J, Le Bihan N, Quaternion toolbox for matlab[EB/OL]. http://qtfm. sourceforge. net.

[6] Wang M, Ma W. A structure-preserving algorithm for the quaternion Cholesky decomposition[J]. Appl Math Comput, 2013,223: 354-361.

[7] Wang M, Ma W. A structure-preserving method for the quaternion LU decomposition in quaternionic quantum theory[J]. Comput Phys Comm, 2013,184:2182-2186.

四元数 Cholesky 分解的实保结构算法的再研究

李　莹[1]　魏木生[1,2]　张凤霞[1]　赵建立[1]

(1.聊城大学 数学科学学院,山东 聊城 252059;2.上海师范大学 数理学院,上海 200234)

摘　要　文献[6]中,作者提出了四元数 Cholesky 分解的一种实保结构算法.本文对四元数 Cholesky 分解的实保结构算法进行了细致的研究,给出了基于高效运算的四元数 Hermitian 正定矩阵的 $\boldsymbol{LDL}^{\mathrm{H}}$及 $\boldsymbol{LL}^{\mathrm{H}}$分解的实保结构算法.我们将这两种实保结构算法的运算时间及精度与文献[6]中的算法及 Matlab 中的四元数工具包 QTFM 进行了比较.数值例子表明本文所提出的算法相对于利用低效运算[6]的算法及利用四元数代数运算的 QTFM 更加有效.

关键词　四元数矩阵;$\boldsymbol{LDL}^{\mathrm{H}}$; Cholesky 分解;实表示; 实保结构算法

中图分类号　O151.21　　**文献标识码**　A

(上接第5页)

2+1 维欧拉方程的奇异解

刘希强[1]　刘　睿[2]

(1.聊城大学 数学科学学院,山东 聊城 252059;2.聊城大学 计算机学院,山东 聊城 252059)

摘　要　借助于推广的 CK 方法,获得了 2+1 维欧拉方程的等价变换和新旧解之间的关系。基于拉普拉斯方程的解,给出了构造欧拉方程某些显式解的公式,并列出了部分奇异解.利用所求出的等价变换及其求解公式,得到了 2+1 维欧拉方程一些随时间演化的新奇异解.

关键词　欧拉方程; 奇异解; 等价变换; 贝克隆变换

中图分类号　O175.2　　**文献标识码**　A