From quaternionic multiplication to matrix decomposition

Kohji Suzuki[*]

kohjisuzuki@yandex.com

Abstract

We are led to certain kinds of matrix decompositions through quaternionic multiplication.

1 Introduction

It is well-known that quaternions (q's) can be expressed as matrices [1]. Presenting examples, we perform some quaternionic computations which are based mainly on matrices. We then discuss certain kinds of matrix decompositions derived from such computations.

2 Methods and examples

We denote two q's q_x and q_y by $x_0 + x_1 i + x_2 j + x_3 k$ and $y_0 + y_1 i + y_2 j + y_3 k$ with $x_n, y_n \in \mathbb{R}$ [1], respectively, and calculate their product $q_x q_y$ as follows:

[*] Protein Science Society of Japan

[1] The subscript n takes the values 0, 1, 2, 3.

$$
\begin{aligned}
q_x q_y &= (x_0 + x_1 i + x_2 j + x_3 k)(y_0 + y_1 i + y_2 j + y_3 k) \\
&= x_0(y_0 + y_1 i + y_2 j + y_3 k) + x_1 i(y_0 + y_1 i + y_2 j + y_3 k) \\
&\quad + x_2 j(y_0 + y_1 i + y_2 j + y_3 k) + x_3 k(y_0 + y_1 i + y_2 j + y_3 k) \\
&= x_0 y_0 + x_0 y_1 i + x_0 y_2 j + x_0 y_3 k + x_1 i y_0 + x_1 i y_1 i + x_1 i y_2 j + x_1 i y_3 k \\
&\quad + x_2 j y_0 + x_2 j y_1 i + x_2 j y_2 j + x_2 j y_3 k + x_3 k y_0 + x_3 k y_1 i + x_3 k y_2 j + x_3 k y_3 k \\
&= x_0 y_0 + x_0 y_1 i + x_0 y_2 j + x_0 y_3 k + x_1 y_0 i + x_1 y_1 ii + x_1 y_2 ij + x_1 y_3 ik \\
&\quad + x_2 y_0 j + x_2 y_1 ji + x_2 y_2 jj + x_2 y_3 jk + x_3 y_0 k + x_3 y_1 ki + x_3 y_2 kj + x_3 y_3 kk \\
&= {}^{2}\ x_0 y_0 + x_0 y_1 i + x_0 y_2 j + x_0 y_3 k + x_1 y_0 i - x_1 y_1 + x_1 y_2 k - x_1 y_3 j \\
&\quad + x_2 y_0 j - x_2 y_1 k - x_2 y_2 + x_2 y_3 i + x_3 y_0 k + x_3 y_1 j - x_3 y_2 i - x_3 y_3 \\
&= x_0 y_0 - x_1 y_1 - x_2 y_2 - x_3 y_3 + (x_0 y_1 + x_1 y_0 + x_2 y_3 - x_3 y_2) i \\
&\quad + (x_0 y_2 - x_1 y_3 + x_2 y_0 + x_3 y_1) j + (x_0 y_3 + x_1 y_2 - x_2 y_1 + x_3 y_0) k. \qquad (1)
\end{aligned}
$$

Then, we represent q_x and q_y by corresponding matrices M_x and M_y as shown in the following.

$$
M_x = \begin{pmatrix} x_0 & -x_1 & -x_2 & -x_3 \\ x_1 & x_0 & -x_3 & x_2 \\ x_2 & x_3 & x_0 & -x_1 \\ x_3 & -x_2 & x_1 & x_0 \end{pmatrix} [2,3], \ M_y = \begin{pmatrix} y_0 & y_1 & y_2 & y_3 \\ y_1 & -y_0 & y_3 & -y_2 \\ y_2 & -y_3 & -y_0 & y_1 \\ y_3 & y_2 & -y_1 & -y_0 \end{pmatrix}^{3}.
$$

So

$$
\begin{aligned}
M_x M_y &= \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \\
&= \begin{pmatrix} x_0 & -x_1 & -x_2 & -x_3 \\ x_1 & x_0 & -x_3 & x_2 \\ x_2 & x_3 & x_0 & -x_1 \\ x_3 & -x_2 & x_1 & x_0 \end{pmatrix} \begin{pmatrix} y_0 & y_1 & y_2 & y_3 \\ y_1 & -y_0 & y_3 & -y_2 \\ y_2 & -y_3 & -y_0 & y_1 \\ y_3 & y_2 & -y_1 & -y_0 \end{pmatrix}.
\end{aligned}
$$

Explicitly,

$$
\begin{cases}
a_{11} = x_0 y_0 - x_1 y_1 - x_2 y_2 - x_3 y_3, \\
a_{12} = x_0 y_1 + x_1 y_0 + x_2 y_3 - x_3 y_2, \\
a_{13} = x_0 y_2 - x_1 y_3 + x_2 y_0 + x_3 y_1, \\
a_{14} = x_0 y_3 + x_1 y_2 - x_2 y_1 + x_3 y_0, \qquad \text{(cont'd)}
\end{cases}
$$

[2]We have used the relations such as $i^2 = -1, ij = k$, and so forth .

[3]This matrix has a not-so-striking resemblance to the Cayley table of q's. See **4.1**.

$$\begin{cases} a_{21} = x_1y_0 + x_0y_1 - x_3y_2 + x_2y_3, \\ a_{22} = x_1y_1 - x_0y_0 + x_3y_3 + x_2y_2, \\ a_{23} = x_1y_2 + x_0y_3 + x_3y_0 - x_2y_1, \\ a_{24} = x_1y_3 - x_0y_2 - x_3y_1 - x_2y_0, \\ a_{31} = x_2y_0 + x_3y_1 + x_0y_2 - x_1y_3, \\ a_{32} = x_2y_1 - x_3y_0 - x_0y_3 - x_1y_2, \\ a_{33} = x_2y_2 + x_3y_3 - x_0y_0 + x_1y_1, \\ a_{34} = x_2y_3 - x_3y_2 + x_0y_1 + x_1y_0, \\ a_{41} = x_3y_0 - x_2y_1 + x_1y_2 + x_0y_3, \\ a_{42} = x_3y_1 + x_2y_0 - x_1y_3 + x_0y_2, \\ a_{43} = x_3y_2 - x_2y_3 - x_1y_0 - x_0y_1, \\ a_{44} = x_3y_3 + x_2y_2 + x_1y_1 - x_0y_0. \end{cases}$$

Henceforth, we will double-check our computations using OpenAxiom and wxMaxima 12.04.0. [4, 5] First, we calculate M_xM_y as follows:

```
$ open-axiom
                 OpenAxiom: The Open Scientific Computation Platform
                       Version: OpenAxiom 1.5.0-2012-02-03
                   Built on Wednesday May 16, 2012 at 12:15:25
-----------------------------------------------------------------------
   Issue )copyright to view copyright notices.
   Issue )summary for a summary of useful system commands.
   Issue )quit to leave OpenAxiom and return to shell.
-----------------------------------------------------------------------

(1) -> M_x:=[[x_0,-x_1,-x_2,-x_3],[x_1,x_0,-x_3,x_2],[x_2,x_3,x_0,-x_1],
             [x_3,-x_2,x_1,x_0]]

   (1)  [[x0,- x1,- x2,- x3],[x1,x0,- x3,x2],[x2,x3,x0,- x1],
             [x3,- x2,x1,x0]]

          Type: List List Polynomial Integer
```

[4]For some settings, see footnote 3 of this .

[5]We sometimes edit verbatim outputs by softwares we use to make them look neat.

```
(2) -> M_y:=[[y_0,y_1,y_2,y_3],[y_1,-y_0,y_3,-y_2],[y_2,-y_3,-y_0,y_1],
              [y_3,y_2,-y_1,-y_0]]

   (2)  [[y0,y1,y2,y3],[y1,- y0,y3,- y2],[y2,- y3,- y0,y1],
              [y3,y2,- y1,- y0]]

          Type: List List Polynomial Integer

(3) -> M_x*M_y

   (3)
   [
     [- x3 y3 - x2 y2 - x1 y1 + x0 y0, x2 y3 - x3 y2 + x0 y1 + x1 y0,

      - x1 y3 + x0 y2 + x3 y1 + x2 y0, x0 y3 + x1 y2 - x2 y1 + x3 y0]

     ,

     [x2 y3 - x3 y2 + x0 y1 + x1 y0, x3 y3 + x2 y2 + x1 y1 - x0 y0,

      x0 y3 + x1 y2 - x2 y1 + x3 y0, x1 y3 - x0 y2 - x3 y1 - x2 y0]

     ,

     [- x1 y3 + x0 y2 + x3 y1 + x2 y0, - x0 y3 - x1 y2 + x2 y1 - x3 y0,

      x3 y3 + x2 y2 + x1 y1 - x0 y0, x2 y3 - x3 y2 + x0 y1 + x1 y0]

     ,

     [x0 y3 + x1 y2 - x2 y1 + x3 y0, - x1 y3 + x0 y2 + x3 y1 + x2 y0,

      - x2 y3 + x3 y2 - x0 y1 - x1 y0, x3 y3 + x2 y2 + x1 y1 - x0 y0]

     ]

          Type: Matrix Polynomial Integer
```

```
$ wxmaxima
```

```
(%i1) M_x:matrix([x_0,-x_1,-x_2,-x_3],[x_1,x_0,-x_3,x_2],[x_2,x_3,x_0,-x_1],[x_3,-x_2,x_1,x_0]);
      M_y:matrix([y_0,y_1,y_2,y_3],[y_1,-y_0,y_3,-y_2],[y_2,-y_3,-y_0,y_1],[y_3,y_2,-y_1,-y_0]);
      M_x.M_y;

      [x_0  -x_1  -x_2  -x_3]
      [x_1   x_0  -x_3   x_2]
(%o1) [x_2   x_3   x_0  -x_1]
      [x_3  -x_2   x_1   x_0]

      [y_0   y_1   y_2   y_3]
      [y_1  -y_0   y_3  -y_2]
(%o2) [y_2  -y_3  -y_0   y_1]
      [y_3   y_2  -y_1  -y_0]

(%o3)
[-x_3 y_3-x_2 y_2-x_1 y_1+x_0 y_0   x_2 y_3-x_3 y_2+x_0 y_1+x_1 y_0   -x_1 y_3+x_0 y_2+x_3 y_1+x_2 y_0   x_0 y_3+x_1 y_2-x_2 y_1+x_3 y_0]
[x_2 y_3-x_3 y_2+x_0 y_1+x_1 y_0    x_3 y_3+x_2 y_2+x_1 y_1-x_0 y_0   x_0 y_3+x_1 y_2-x_2 y_1+x_3 y_0    x_1 y_3-x_0 y_2-x_3 y_1-x_2 y_0]
[-x_1 y_3+x_0 y_2+x_3 y_1+x_2 y_0   -x_0 y_3-x_1 y_2+x_2 y_1-x_3 y_0  x_3 y_3+x_2 y_2+x_1 y_1-x_0 y_0    x_2 y_3-x_3 y_2+x_0 y_1+x_1 y_0]
[x_0 y_3+x_1 y_2-x_2 y_1+x_3 y_0    -x_1 y_3+x_0 y_2+x_3 y_1+x_2 y_0  -x_2 y_3+x_3 y_2-x_0 y_1-x_1 y_0   x_3 y_3+x_2 y_2+x_1 y_1-x_0 y_0]
```

Meanwhile, we notice that (1) can be rewritten using a_{11}, a_{12}, a_{13}, and a_{14}. That is, (1) = $a_{11} + a_{12}i + a_{13}j + a_{14}k$, which prompts us to rewrite $M_x M_y$ as

$$M_z = \begin{pmatrix} \alpha & \beta & \gamma & \delta \\ \beta & -\alpha & \delta & -\gamma \\ \gamma & -\delta & -\alpha & \beta \\ \delta & \gamma & -\beta & -\alpha \end{pmatrix},$$

where

$$\begin{cases} \alpha = x_0y_0 - x_1y_1 - x_2y_2 - x_3y_3, \\ \beta = x_0y_1 + x_1y_0 + x_2y_3 - x_3y_2, \\ \gamma = x_0y_2 - x_1y_3 + x_2y_0 + x_3y_1, \\ \delta = x_0y_3 + x_1y_2 - x_2y_1 + x_3y_0. \end{cases}$$

Now we present two examples, which will be checked similarly.

Example 1

$q_1 = 1 + 3i + 4j + 6k$, $q_2 = -4 - 2i + 3j + 5k$.

$$M_1M_2 = \begin{pmatrix} 1 & -3 & -4 & -6 \\ 3 & 1 & -6 & 4 \\ 4 & 6 & 1 & -3 \\ 6 & -4 & 3 & 1 \end{pmatrix} \begin{pmatrix} -4 & -2 & 3 & 5 \\ -2 & 4 & 5 & -3 \\ 3 & -5 & 4 & -2 \\ 5 & 3 & 2 & 4 \end{pmatrix}$$

$$= \begin{pmatrix} -40 & -12 & -40 & -2 \\ -12 & 40 & -2 & 40 \\ -40 & 2 & 40 & -12 \\ -2 & -40 & 12 & 40 \end{pmatrix}. \tag{2}$$

Hence, $q_1q_2 = -40 - 12i - 40j - 2k$.

```
$ open-axiom
```
[6]

```
(1) -> M_1:=matrix[[1,-3,-4,-6],[3,1,-6,4],[4,6,1,-3],[6,-4,3,1]]

        +1   - 3   - 4   - 6+
        |                   |
        |3    1    - 6    4 |
   (1)  |                   |
        |4    6     1    - 3|
        |                   |
        +6   - 4    3     1 +

          Type: Matrix Integer

(2) -> M_2:=matrix[[-4,-2,3,5],[-2,4,5,-3],[3,-5,4,-2],[5,3,2,4]]

        +- 4   - 2   3    5 +
        |                   |
        |- 2    4    5   - 3|
   (2)  |                   |
        | 3    - 5   4   - 2|
        |                   |
        + 5     3    2    4 +

          Type: Matrix Integer
```

```
(3) -> M_1*M_2

        +- 40  - 12  - 40  - 2 +
        |                      |
        |- 12   40   - 2    40 |
   (3)  |                      |
        |- 40   2     40   - 12|
        |                      |
        +- 2   - 40   12    40 +

          Type: Matrix Integer

(4) -> qi:= quatern$Quaternion(Integer);

          Type: ((Integer,Integer,Integer,Integer) -> Quaternion Integer)

(5) -> q_1:=qi(1,3,4,6);

          Type: Quaternion Integer

(6) -> q_2:=qi(-4,-2,3,5);

          Type: Quaternion Integer

(7) -> q_1*q_2

   (7)  - 40 - 12i - 40j - 2k

          Type: Quaternion Integer
```

```
$ wxmaxima

(%i1)  M_1:matrix([1,-3,-4,-6],[3,1,-6,4],[4,6,1,-3],[6,-4,3,1]);
       M_2:matrix([-4,-2,3,5],[-2,4,5,-3],[3,-5,4,-2],[5,3,2,4]);
       M_1.M_2;
       load(atensor)$
       init_atensor(quaternion)$
       q(a,b,c,d):=a+b.v[1]+c.v[2]+d.v[1].v[2]$
       expand(atensimp(q(1,3,4,6).q(-4,-2,3,5)));
```

(%o1) $\begin{pmatrix} 1 & -3 & -4 & -6 \\ 3 & 1 & -6 & 4 \\ 4 & 6 & 1 & -3 \\ 6 & -4 & 3 & 1 \end{pmatrix}$

(%o2) $\begin{pmatrix} -4 & -2 & 3 & 5 \\ -2 & 4 & 5 & -3 \\ 3 & -5 & 4 & -2 \\ 5 & 3 & 2 & 4 \end{pmatrix}$

(%o3) $\begin{pmatrix} -40 & -12 & -40 & -2 \\ -12 & 40 & -2 & 40 \\ -40 & 2 & 40 & -12 \\ -2 & -40 & 12 & 40 \end{pmatrix}$

(%o7) $-2\,(v_1.v_2) - 40\,v_2 - 12\,v_1 - 40$

Example 2

$$q_3 = 5 + i + 2j + 3k,\ q_4 = -2 + 2i + 4j + 6k.$$

$$M_3 M_4 = \begin{pmatrix} 5 & -1 & -2 & -3 \\ 1 & 5 & -3 & 2 \\ 2 & 3 & 5 & -1 \\ 3 & -2 & 1 & 5 \end{pmatrix} \begin{pmatrix} -2 & 2 & 4 & 6 \\ 2 & 2 & 6 & -4 \\ 4 & -6 & 2 & 2 \\ 6 & 4 & -2 & 2 \end{pmatrix}$$

$$= \begin{pmatrix} -38 & 8 & 16 & 24 \\ 8 & 38 & 24 & -16 \\ 16 & -24 & 38 & 8 \\ 24 & 16 & -8 & 38 \end{pmatrix}.$$

Hence, $q_3 q_4 = -38 + 8i + 16j + 24k$.

```
$ open-axiom
```
[7]

```
(1) -> M_3:=matrix[[5,-1,-2,-3],[1,5,-3,2],[2,3,5,-1],[3,-2,1,5]]

        +5  - 1  - 2  - 3+
        |               |
        |1   5   - 3   2 |
   (1)  |               |
        |2   3    5   - 1|
        |               |
        +3  - 2   1    5 +

          Type: Matrix Integer

(2) -> M_4:=matrix[[-2,2,4,6],[2,2,6,-4],[4,-6,2,2],[6,4,-2,2]]

        +- 2   2    4    6 +
        |                  |
        | 2    2    6   - 4|
   (2)  |                  |
        | 4   - 6   2    2 |
        |                  |
        + 6    4   - 2   2 +

          Type: Matrix Integer
```

[6,7]We suppress messages following this command which we have already seen.

```
(3) -> M_3*M_4

        +- 38   8    16    24 +
        |                     |
        | 8     38   24   - 16|
   (3)  |                     |
        | 16   - 24  38    8  |
        |                     |
        + 24    16   - 8   38 +

          Type: Matrix Integer

(4) -> qi:= quatern$Quaternion(Integer);

          Type: ((Integer,Integer,Integer,Integer) -> Quaternion Integer)

(5) -> q_3:=qi(5,1,2,3);

          Type: Quaternion Integer

(6) -> q_4:=qi(-2,2,4,6);

          Type: Quaternion Integer

(7) -> q_3*q_4

   (7)  - 38 + 8i + 16j + 24k

          Type: Quaternion Integer
```

```
$ wxmaxima
```

```
(%i1)  M_3:matrix([5,-1,-2,-3],[1,5,-3,2],[2,3,5,-1],[3,-2,1,5]);
       M_4:matrix([-2,2,4,6],[2,2,6,-4],[4,-6,2,2],[6,4,-2,2]);
       M_3.M_4;
       load(atensor)$
       init_atensor(quaternion)$
       q(a,b,c,d):=a+b.v[1]+c.v[2]+d.v[1].v[2]$
       expand(atensimp(q(5,1,2,3).q(-2,2,4,6)));
```

(%o1) $\begin{pmatrix} 5 & -1 & -2 & -3 \\ 1 & 5 & -3 & 2 \\ 2 & 3 & 5 & -1 \\ 3 & -2 & 1 & 5 \end{pmatrix}$

(%o2) $\begin{pmatrix} -2 & 2 & 4 & 6 \\ 2 & 2 & 6 & -4 \\ 4 & -6 & 2 & 2 \\ 6 & 4 & -2 & 2 \end{pmatrix}$

(%o3) $\begin{pmatrix} -38 & 8 & 16 & 24 \\ 8 & 38 & 24 & -16 \\ 16 & -24 & 38 & 8 \\ 24 & 16 & -8 & 38 \end{pmatrix}$

(%o7) $24\,(v_1.v_2) + 16\,v_2 + 8\,v_1 - 38$

3 Discussion [8]

It follows from (2) that

$$\begin{pmatrix} -40 & -12 & -40 & -2 \\ -12 & 40 & -2 & 40 \\ -40 & 2 & 40 & -12 \\ -2 & -40 & 12 & 40 \end{pmatrix} = M_1 M_2. \tag{3}$$

Noticing that $q_3 q_4 = q_4 q_3$, we have

$$\begin{aligned} &\begin{pmatrix} -38 & 8 & 16 & 24 \\ 8 & 38 & 24 & -16 \\ 16 & -24 & 38 & 8 \\ 24 & 16 & -8 & 38 \end{pmatrix} \\ &= \begin{pmatrix} 5 & -1 & -2 & -3 \\ 1 & 5 & -3 & 2 \\ 2 & 3 & 5 & -1 \\ 3 & -2 & 1 & 5 \end{pmatrix} \begin{pmatrix} -2 & 2 & 4 & 6 \\ 2 & 2 & 6 & -4 \\ 4 & -6 & 2 & 2 \\ 6 & 4 & -2 & 2 \end{pmatrix} \\ &= \begin{pmatrix} -2 & -2 & -4 & -6 \\ 2 & -2 & -6 & 4 \\ 4 & 6 & -2 & -2 \\ 6 & -4 & 2 & -2 \end{pmatrix} \begin{pmatrix} 5 & 1 & 2 & 3 \\ 1 & -5 & 3 & -2 \\ 2 & -3 & -5 & 1 \\ 3 & 2 & -1 & -5 \end{pmatrix}. \end{aligned} \tag{4}$$

Thus, it follows from (3) and (4) that our quaternionic calculations have led to the cases where matrices are decomposed like $M_a = M_b M_c$ and $M_d = M_e M_f = M_g M_h$.

[8]We refrain from performing detailed computations here.

Acknowledgment. We would like to thank the developers of OpenAxiom and wxMaxima for their indirect help which enabled us to verify our computations.

References

[1] Coppel, W. A., "Number theory: An introduction to mathematics 2nd ed.," Springer-Verlag New York Inc. 2009 p49.

[2] O'Meara, K. C., Clark, J., and Vinsonhaler, C. I., "Advanced topics in linear algebra : Weaving matrix problems through the Weyr form," Oxford University Press 2011 p203.

[3] Ayala, R., Domínguez, E., and Quintero, A., "Algebraic topology: An introduction," Alpha Science 2012 p27.

4 Appendix

4.1 Obtaining a 'matrix'

The celebrated Cayley table of q's is

$\times$	1	i	j	k
1	1	i	j	k
i	i	-1	k	$-j$
j	j	$-k$	-1	i
k	k	j	$-i$	-1

.

Striking out the uppermost row and leftmost column, we get the 'matrix' below.

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

A becomes M_y, if we replace the entries $1, i, j, k$ by y_0, y_1, y_2, y_3, respectively. [9]

[9] Incidentally, we get M_z, if we replace them by $\alpha, \beta, \gamma, \delta$ in a similar manner.

4.2 On 'minimalism'

We have presented ways to get/verify $q_x q_y$. However, if we wish to be minimalistic [10] , we have only to calculate

$$\begin{pmatrix} x_0 & -x_1 & -x_2 & -x_3 \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{pmatrix} \begin{pmatrix} y_0 & y_1 & y_2 & y_3 \\ y_1 & -y_0 & y_3 & -y_2 \\ y_2 & -y_3 & -y_0 & y_1 \\ y_3 & y_2 & -y_1 & -y_0 \end{pmatrix} = \begin{pmatrix} \alpha & \beta & \gamma & \delta \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{pmatrix}$$

or

$$\begin{pmatrix} x_0 & -x_1 & -x_2 & -x_3 \\ x_1 & x_0 & -x_3 & x_2 \\ x_2 & x_3 & x_0 & -x_1 \\ x_3 & -x_2 & x_1 & x_0 \end{pmatrix} \begin{pmatrix} y_0 & - & - & - \\ y_1 & - & - & - \\ y_2 & - & - & - \\ y_3 & - & - & - \end{pmatrix} = \begin{pmatrix} \alpha & - & - & - \\ \beta & - & - & - \\ \gamma & - & - & - \\ \delta & - & - & - \end{pmatrix},$$

where dashes denote entries the minimalistic can ignore, to get $\alpha + \beta i + \gamma j + \delta k$, or $q_x q_y$. In this way, we can make $M_x M_y$-based multiplication slightly easier and faster.

[10]By *minimalistic*, we mean that we are interested only in getting the product of two q's as soon as possible.