Rotations of Moment of Inertia Tensor using Quaternions

Mikica B Kocic, 2012-04-22, v0.3

There is a wonderful connection between complex numbers (and quaternions, as their extension) and geometry in which, translations correspond to additions, rotations and scaling to multiplications, and reflections to conjugations. However, the beauty of describing spatial rotations with quaternions is shadowed by an expression like:

\[
\begin{pmatrix}
1 - 2(y^2 + z^2) & 2xy - 2wz & 2wy + 2xz \\
2xy + 2wz & 1 - 2(x^2 + z^2) & 2yz - 2wx \\
2xz - 2wy & 2wx + 2yz & 1 - 2(x^2 + y^2)
\end{pmatrix}^T
\]

which actually represents such simple transformation as the rotation of moment of inertia tensor using quaternions [2].

The purpose of this document is to express spatial rotations of a moment of inertia tensor in the pure quaternionic form and to highlight the algebraical meaning behind such rotations by proving some interesting theorems.

### Conventions and Definitions

Load quaternions package (see Quat.nb for the annotated source code)

```mathematica
Get[ "Quat.m", Path -> {NotebookDirectory[]} ];
```

#### Conventions

In this document, matrices, vectors and matrix representation of quaternions are shown in uppercase font (roman-bold font in text) and quaternions and other numbers are shown in lowercase font (italic font in text). Since the symbol \( \text{I} \) in Mathematica is used for imaginary unit, the selected symbol for moment of inertia is the uppercase script \( J \) i.e. \( J \).

Matrices 4×4 originated from quaternions may be geometrically partitioned into scalar (real), vector (pure imaginary) and cross-product parts [3]. Such partitions are visualized using row and column divider lines throughout this document:

```
\[
\begin{pmatrix}
\text{scalar} \\
\text{vector}
\end{pmatrix}
\begin{pmatrix}
\text{vector} \\
\text{cross product}
\end{pmatrix}
```

#### General symbols and variables used in theorems (definitions)

Hamilton's quaternions \( q, q_1 \) and \( q_2 \) with real number components (but not with complex number components as biquaternions; see definition of default $Assumptions bellow):

```mathematica
q = Q[ w, x, y, z ];
q1 = Q[ w1, x1, y1, z1 ];
q2 = Q[ w2, x2, y2, z2 ];
```

Identity matrix \( I_4 \) in \( \mathbb{R}^4 \):

```mathematica
I4 = IdentityMatrix[4];
```
Common $\mathbb{R}^4$ matrices $A$ and $B$:

$$A = \begin{pmatrix}
aww & awx &awy & awz \\
axw & axx & axy & axz \\
ayw & ayx & ayy & ayz \\
azw & azx & azy & azz
\end{pmatrix};$$

$$B = \begin{pmatrix}
bww & bwx & bwy & bwz \\
bxw & bxx & bxy & bxz \\
byw & byx & byy & byz \\
bzw & bzx & bzy & bzz \\
\end{pmatrix};$$

Symmetrical $\mathbb{R}^4$ matrix representing moment of inertia tensor $J$:

$$J = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & Ixx & Ixy & Ixz \\
0 & Ixy & Iyy & Iyz \\
0 & Ixz & Iyz & Izz
\end{pmatrix};$$

Default assumption used for all asserted statements and expressions in this notebook is that previously defined entities $q, q_1, q_2, A, B$ and $J$ are on field of $\mathbb{R}$, i.e. that their components are elements in $\mathbb{R}$:

\[
\$Assumptions = Flatten[\{ToList[\{q, q_1, q_2\}], A, B, J\} \in \text{Reals};
\]

## Different Types of Multiplications

The ordinary notation for multiplication in Mathematica is

- $A \cdot B$ : multiplication between matrices ("dot" multiplication)
- $a \times b$ : cross-product between 3D vectors
- $p ** q$ : quaternion multiplication (both in Quat.m and Mathematica Quaternions package)

Beside the ordinary notation for multiplication, this document also introduces the following notation based on the center dot (·) operator.

- $A \cdot B$ : multiplication between matrices
- $p \cdot q$ : quaternion multiplication
- $q \cdot A$ : column-wise quaternion-matrix multiplication

Note that the center dot operator in Mathematica is without built-in meaning.

### Matrix multiplication

Define center dot operator $A \cdot B$ as matrix multiplication (otherwise, Mathematica uses plain dot "." for matrix multiplications).

\[
\text{CenterDot}[m \_\_?MatrixQ] := \text{Dot}[m]
\]

### Quaternion multiplication

Mathematica uses non-commutative multiply (**) operator as symbol for quaternion multiplication. The Quat.m package and this document follow this convention. Here we define also the center dot operator as symbol for quaternion multiplication, just for convinience and esthetics.

\[
\text{CenterDot}[q \_\_?Quaternion] := \text{NonCommutativeMultiply}[q]
\]
### Column-wise (or horizontal) quaternion-matrix multiplication

Let us now define multiplication between quaternion \( q \) and matrix \( M \), where it is assumed that matrix is a row vector of quaternions in columns, i.e. \( M = ( q_1 \ q_2 \ \ldots \ q_n ) \), so that quaternion multiplication is done between \( q \) and \( q_j \) in every column \( j \).

\[
\begin{pmatrix}
q_1 & q_j & q_n \\
q_11 & q_12 & q_1n \\
q_21 & q_22 & q_2n \\
q_31 & q_32 & \ldots & q_3n \\
q_41 & q_42 & q_4n \\
\end{pmatrix}
\cdot
\begin{pmatrix}
q_1 & q_j & q_n \\
\bar{q}_11 & \bar{q}_12 & \bar{q}_1n \\
\bar{q}_21 & \bar{q}_22 & \bar{q}_2n \\
\bar{q}_31 & \bar{q}_32 & \ldots & \bar{q}_3n \\
\bar{q}_41 & \bar{q}_42 & \bar{q}_4n \\
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
\bar{q}_11 & \bar{q}_12 & \bar{q}_1n \\
\bar{q}_21 & \bar{q}_22 & \bar{q}_2n \\
\bar{q}_31 & \bar{q}_32 & \ldots & \bar{q}_3n \\
\bar{q}_41 & \bar{q}_42 & \bar{q}_4n \\
\end{pmatrix}
\]

Since the quaternion multiplication is not commutative, there are several cases to distinguish:

- **a) Left multiplication** \( q \cdot M \), where a quaternion in each column in \( M \) is multiplied by the quaternion \( q \) from the left:

  ```
  CenterDot[ q_?QQ, mat_?MatrixQ ] := Transpose@
  Table[
    ToList[ q_ ToQ@mat[[All, j]] ], { j, 1, Dimensions[mat][2] }]
  ```

- **b) Right multiplication** \( M \cdot q \), where a quaternion in each column in \( M \) is multiplied by the quaternion \( q \) from the right:

  ```
  CenterDot[ mat_?MatrixQ, q_?QQ ] := Transpose@
  Table[
    ToList[ ToQ@mat[[All, j]] \cdot q ], { j, 1, Dimensions[mat][2] }]
  ```

- **c) Compound multiplication from both sides** \( p \cdot M \cdot q \), where multiplication of each column in \( M \) by the quaternion \( p \) from the left and \( q \) from the right:

  ```
  CenterDot[ q1_?QQ, mat_?MatrixQ, q2_?QQ ] := Transpose@
  Table[
    ToList[ q1_ ToQ@mat[[All, j]] \cdot q2 ], { j, 1, Dimensions[mat][2] }]
  ```

The multiplication between a quaternion and a matrix is associative, i.e. it holds:

\[
q_1 \cdot M \cdot q_2 = q_1 \cdot (M \cdot q_2) = (q_1 \cdot M) \cdot q_2
\]

which is proved as a theorem in the following sections.

Note that for a single column matrix \( M \), the multiplication between a quaternion and a matrix falls back to an ordinary quaternion multiplication.

### Row-wise (or vertical) quaternion-matrix multiplication

Row-wise quaternion-matrix multiplication can be defined by transposing matrices and using column-wise multiplication:

\[
(q_1 \cdot M \cdot q_2)
\]

### Conj[] matrix operator

Conj[] operator conjugates only the vector part of the matrix, but not the scalar and the cross product parts:

\[
\text{Conj[ } q_?MatrixQ ] :=
\begin{pmatrix}
q_{1,11} & -q_{1,12} & -q_{1,13} & -q_{1,14} \\
-q_{2,11} & q_{2,22} & q_{2,23} & q_{2,24} \\
-q_{3,11} & q_{3,22} & q_{3,33} & q_{3,44} \\
-q_{4,11} & q_{4,22} & q_{4,33} & q_{4,44} \\
\end{pmatrix}
\]

Note that Conj[] is not a conjugate of a quaternion! The conjugate of a quaternion corresponds to the plain transpose of the matrix.
Idea behind quaternion-matrix multiplication

Basic idea behind the quaternion-matrix multiplication comes from an expression:

\[
\text{Assert} \left[ \begin{align*}
Q_1[1, 0, 0, 0] & = q \cdot Q_1[1, 0, 0, 0] \cdot q^*, \\
\text{Assumptions} \rightarrow |q|^2 & = 1
\end{align*} \right]
\]

\[\text{True (after Simplify)} \]

for which it is assumed to be also valid when transforming identity matrix \(I_4\) that is equivalent representation of the quaternion \(Q_1[1, 0, 0, 0]\) in matrix form, i.e. that there should be kind of multiplication where:

\[
\text{Assert} \left[ \begin{align*}
(1 0 0 0)^t & = q \cdot (1 0 0 0)^t \cdot q^*, \\
\text{Assumptions} \rightarrow |q|^2 & = 1
\end{align*} \right]
\]

\[\text{True (after Simplify)} \]

Theorems

- Quaternion-matrix multiplication

Quaternion multiplication associativity (sanity-check):

\[
\text{Assert} \left[ \begin{align*}
q_1 \cdot q \cdot q_2 & = q_1 \cdot (q \cdot q_2) = (q_1 \cdot q) \cdot q_2 \\
\text{True}
\end{align*} \right]
\]

Quaternion-matrix multiplication associativity theorems:

\[
\text{Assert} \left[ \begin{align*}
q_1 \cdot A \cdot q_2 & = q_1 \cdot (A \cdot q_2) = (q_1 \cdot A) \cdot q_2 \\
\text{True}
\end{align*} \right]
\]

\[
\text{Assert} \left[ \begin{align*}
q \cdot A \cdot q^* & = (q \cdot A) \cdot q^* \\
\text{True}
\end{align*} \right]
\]

\[
\text{Assert} \left[ \begin{align*}
q \cdot A \cdot q^* & = q \cdot (A \cdot q^*) \\
\text{True}
\end{align*} \right]
\]

\[
\text{Assert} \left[ \begin{align*}
(q \cdot I_4)^t & = q^* \cdot I_4 \\
\text{True}
\end{align*} \right]
\]

\[
\text{Assert} \left[ \begin{align*}
(I_4 \cdot q)^t & = I_4 \cdot q^* \\
\text{True}
\end{align*} \right]
\]

\[
\text{Assert} \left[ \begin{align*}
(q \cdot I_4) \cdot A & = q \cdot A \\
\text{True}
\end{align*} \right]
\]

\[
\text{Assert} \left[ \begin{align*}
(I_4 \cdot q) \cdot A & = A \cdot q \\
\text{True}
\end{align*} \right]
\]

\[
\text{Assert} \left[ \begin{align*}
(q_1 \cdot I_4) \cdot (I_4 \cdot q_2) & = q_1 \cdot I_4 \cdot q_2 \\
\text{True}
\end{align*} \right]
\]
\[(I_4 \cdot q_2) \cdot ((q_1 \cdot I_4) \cdot A) = (q_1 \cdot A) \cdot q_2 \] // Assert

> True (after Simplify) □

\[(I_4 \cdot q_2) \cdot (q_1 \cdot I_4) \cdot A = q_1 \cdot A \cdot q_2 \]
\[(I_4 \cdot q_2) \cdot (q_1 \cdot I_4) \cdot A = q_1 \cdot A \cdot q_2 \]
\[(q_1 \cdot I_4 \cdot q_2) \cdot A = q_1 \cdot A \cdot q_2] // Assert

> True (after Simplify) □

Conjugations of quaternion-matrix multiplications:

\[\text{Conj}[q \cdot A] = \text{Conj}[A] \cdot q^*] // Assert

> True □

\[\text{Conj}[A \cdot q] = q^* \cdot \text{Conj}[A] // Assert\]

> True □

\[\text{Conj}[q_1 \cdot A \cdot q_2] = \text{Conj}[q_1 \cdot (A \cdot q_2)] = \text{Conj}[A \cdot q_2] \cdot q_1^* = q_2^* \cdot \text{Conj}[A] \cdot q_1^*] // Assert

> True (after Simplify) □

\[\text{Conj}[q \cdot A \cdot q^*] = q \cdot \text{Conj}[A] \cdot q^*] // Assert

> True (after Simplify) □

### Left and right rotation matrices

Because the quaternion multiplication is bilinear, it can be expressed in a matrix form, and in two different ways ([1]). Multiplication on the left \(qp\) gives \(L_q \cdot p\) where \(p\) is now treated as a 4-dimensional column vector \(p\) and multiplication on the right \(pq\) gives \(p \cdot R_q\).

Let us define \(L_q\) and \(R_q\) as quaternion-multiplication with identity matrix:

\[
\text{ClearAll}\{L, R, Q\}
\]

\[
L_q := q \cdot I_4 \\
R_q := I_4 \cdot q \\
Q_q := L_q \cdot R_q^* \\
\]

Proof that the quaternion multiplication can be decomposed into L&R matrix parts and that \(L_q R_q^* p\) corresponds to \(q p q^*\):

\[
\text{Assert[}
L_q \cdot R_q^* \cdot \text{ToVector}[q_1] = \text{ToVector}[q \cdot q_1 \cdot q^*], \\
\text{Assumptions} \rightarrow |q|^2 = 1 
\]

> True (after Simplify) □

### Properties of L, R and Q matrices

\[
L_q^* = (L_q)^* \] // Assert

> True □

\[
R_q^* = (R_q)^* \] // Assert

> True □

\[
(q \cdot R_{q^*}) \cdot A = q \cdot (R_{q^*} \cdot A) = (L_q \cdot A) \cdot q^*] // Assert

> True (after Simplify) □
\[
Q_q = L_q \cdot R_q = (R_q \cdot L_q)^* = (R_q \cdot L_q)^* = R_q \cdot L_q // \text{Assert}
\]

True

\[
Q_q^* = R_q \cdot L_q^* = L_q^* \cdot R_q // \text{Assert}
\]

True

\[
Q_q \cdot Q_q^* = |q|^4 I_4 // \text{Assert}
\]

True (after Simplify)

\[
\text{Conj}[L_q \cdot A] \cdot q^* = \text{Conj}[q \cdot (L_q \cdot A)] // \text{Assert}
\]

True (after Simplify)

\[
\text{Conj}[L_q \cdot A] \cdot q^* = \text{Conj}[L_q \cdot A] \cdot q^* // \text{Assert}
\]

True

\section*{L & R matrices components}

\[
\text{Grid}\{
\begin{array}{c|ccc}
L_q & L_q^* & R_q & R_q^* \\
\hline
w & -x & -y & -z \\
x & w & -z & y \\
y & z & w & x \\
z & -y & x & w \\
\end{array}
\}
\]

\section*{Rotation matrix Q components}

\[
\text{Grid}\{
\begin{array}{c|cccc}
\text{Simplify}\{Q_q, \text{Assumptions} \rightarrow |q|^2 = 1\} // \text{QForm,}
\text{Simplify}\{Q_q^*, \text{Assumptions} \rightarrow |q|^2 = 1\} // \text{QForm}
\end{array}
\]

\section*{Shoemake's rotation matrix from quaternion}

Classical Shoemake's form of the rotation matrix from a quaternion is given as (a transformation matrix based on the Cayley-Klein parameters; see [2]):

\[
\text{RotationMatrix4}[q] // \text{QForm}
\]
**Scalar, vector and cross-product parts of L&R matrices**

Scalar part:

\[
\frac{1}{2} (L_q + L_{q'}) = \frac{1}{2} (R_q + R_{q'}) = \begin{pmatrix} w & 0 & 0 & 0 \\ 0 & w & 0 & 0 \\ 0 & 0 & w & 0 \\ 0 & 0 & 0 & w \end{pmatrix} // Assert
\]

Vector part:

\[
\frac{1}{2} (L_q - R_q) = \frac{1}{2} (R_{q'} - L_q) = \begin{pmatrix} 0 & x & y & z \\ -x & 0 & 0 & 0 \\ -y & 0 & 0 & 0 \\ -z & 0 & 0 & 0 \end{pmatrix} // Assert
\]

Cross product part:

\[
\frac{1}{2} (L_q - R_q) = \frac{1}{2} (R_{q'} - L_q) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -z & y \\ 0 & z & 0 & -x \\ 0 & -y & x & 0 \end{pmatrix} // Assert
\]

\[
\frac{1}{2} (L_q - R_q') = \frac{1}{2} (R_{q'} - L_q) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -z & y \\ 0 & z & 0 & -x \\ 0 & -y & x & 0 \end{pmatrix} // Assert
\]

\[Q \cdot Q + 2Q \cdot \text{vector part} \text{ (vector part) form.}\]

Note that the vector part is a pure imaginary part of a quaternion.

\[\text{ToQ@0} = q^2 + 2q \cdot \text{ToQ@Im[q']} - |q|^2 // Assert\]

\[\text{True} \]

\[\text{Assert} \left[ L_q = L_q \cdot L_q + 2L_q \cdot \begin{pmatrix} 0 & x & y & z \\ -x & 0 & 0 & 0 \\ -y & 0 & 0 & 0 \\ -z & 0 & 0 & 0 \end{pmatrix} \right]
\]

\[= L_q \cdot L_q + 2L_q \cdot \left( \frac{1}{2} (R_{q'} - L_q) \right) \]

\[= L_q \cdot R_{q'} \]

\[\text{True (after Simplify)} \]

\[\frac{1}{2} L_q \cdot (L_q \cdot L_q - Q_q) = \frac{1}{2} (L_q - R_{q'}) // Assert\]

\[\text{True (after Simplify)} \]
Equality of rotation matrix from quaternion and Euler-Rodrigues' formula

Euler-Rodrigues' formula for the rotation matrix corresponding to a rotation by an angle $\theta$ about a fixed axis specified by the unit vector $u$.


```
EulerRodrigues[θ_, {x_, y_, z_}] :=
Cos[θ] (1 0 0) + (1 - Cos[θ]) (xx yy zz) + Sin[θ] (0 -z y)
0 1 0 0 0 1

Proof that the rotation matrix from a quaternion (as defined in Quat.m package) is equivalent to Euler-Rodrigues' formula.
```

```
Block[{x, y, z, θ, q},
q = ToQ$AngleAxis[θ, {x, y, z}];
Assert[
RotationMatrix[q] = EulerRodrigues[θ, {x, y, z}],
Assumptions \rightarrow -\pi ≤ θ ≤ \pi \wedge x^2 + y^2 + z^2 = 1 \wedge \{x, y, z, θ\} ∈ Reals]
]

True (after Simplify) □
```

Rotation matrix theorems

```
RM = RotationMatrix4[q];
RM // QForm
```

```
\[
\begin{pmatrix}
1 & 1 - 2 (y^2 + z^2) & 2 x y - 2 w z & 2 w y + 2 x z
0 & 2 x y + 2 w z & 1 - 2 (x^2 + z^2) & -2 w x + 2 y z
0 & -2 w y + 2 x z & 2 w x + 2 y z & 1 - 2 (x^2 + y^2)
\end{pmatrix}
\]

Assert[
RM = Q_3 = L_q . R_q, \wedge
RM = Q_3 = R_q' . L_q',
Assumptions \rightarrow |q|^2 = 1
]

True (after Simplify) □
```

```
Assert[
RM . ToVector[q1] = L_q . R_q' . ToVector[q1]
= ToVector[ q . q1 . q'],
Assumptions \rightarrow |q|^2 = 1
]

True (after Simplify) □
```
Proof that both $Q_q$ and $R_M$ are orthogonal matrices:

$$\text{Assert} \left[ \begin{array}{l}
\text{RM} \cdot A \cdot \text{RM}' \\
\quad = Q_q \cdot A \cdot Q_q' \\
\quad = L_q \cdot R_q' \cdot A \cdot L_q' \cdot R_q \\
\quad = \left( q \cdot (q \cdot A \cdot q^*)' \cdot q^* \right)', \\
\text{Assumptions} \rightarrow |q|^2 = 1
\end{array} \right]$$

→ True (after Simplify) □

$$\text{Assume} \left[ \begin{array}{l}
I_4 \\
\quad = (q \cdot I_4 \cdot q^*)' \cdot I_4 \cdot (q \cdot I_4 \cdot q^*)' \\
\quad = (L_q \cdot R_q')' \cdot I_4 \cdot (L_q \cdot R_q')' \\
\quad = L_q \cdot R_q' \cdot I_4 \cdot R_q' \cdot L_q \\
\quad = L_q \cdot R_q' \cdot I_4 \cdot R_q' \cdot L_q \\
\text{Assumptions} \rightarrow |q|^2 = 1
\end{array} \right]$$

→ True (after Simplify) □

$$\text{Assume} \left[ \begin{array}{l}
R_M = (q \cdot A \cdot q^*)' \cdot I_4 \cdot (q \cdot A \cdot q^*)' \\
\quad = (L_q \cdot R_q')' \cdot A \cdot (L_q \cdot R_q')' \\
\quad = L_q \cdot R_q' \cdot A \cdot R_q' \cdot L_q \\
\quad = L_q \cdot (R_q' \cdot A \cdot R_q') \cdot L_q \\
\text{Assumptions} \rightarrow |q|^2 = 1
\end{array} \right]$$

→ True (after Simplify) □
The quaternionic meaning of $R(q) \cdot A \cdot (R(q))^\top$

The hint how quaternion multiplication affects a (moment of inertia or any other) tensor $A$ can be seen from the earlier proven theorem:

\[
\begin{align*}
\text{Assert} & \left[
Q_q \cdot Q_{q'} = Q_q \cdot Q_{q'} = \mathbf{R} \cdot \mathbf{R}' = \mathbf{I}_4,
\right.
\text{Assumptions } \rightarrow |q|^2 = 1)
\end{align*}
\]

which shows that the tensor $A$ is rotated, at first, by rotating quaternions across each column $A_c = q \cdot A \cdot q^*$, and at second, by rotating quaternions across each row $A_{rc} = \mathbf{I}_q \cdot \mathbf{A}_c \cdot \mathbf{I}_{q^*}$.

The meaning of the compound multiplications across columns and rows is extracting and affecting individual components of quaternions embedded from the previous rotations of the tensor. The moment of inertia tensor can always be reduced to its principal axes in diagonal form. In a matrix representation of a quaternion, the diagonal of the matrix contains the repeated scalar (real) part of the quaternion. (For any quaternion $q$ there exist quaternion $p$ such that $qp$ is scalar.) The principal moment of inertia, on the other hand, has on its diagonal three different scalars in general case, which gives origin to three different quaternions that are embedded and independently shuffled across-columns/rows during rotations.

References

