M.Eng. **René Schwarz**
Master of Engineering in Computer Science and Communication Systems (M.Eng.)
Bachelor of Engineering in Mechatronics, Industrial and Physics Technology (B.Eng.)

Memorandum № 6

Quaternions and Spatial Rotation

1 General Remarks and Notation

- A **quaternion** $q \in \mathbb{R}^4 = \mathbb{H}$ is a four-dimensional hypercomplex number. The **set of quaternions** is a four-dimensional vector space over the real numbers ($\mathbb{R}^4$) and is denoted here by $\mathbb{H}$, in honor of Sir William Rowan Hamilton, who introduced them in 1843. The set of quaternions $\mathbb{H}$ is an extension to the set of complex numbers $\mathbb{C} = \mathbb{R}^2$, which by itself is a two-dimensional vector space over the real numbers, constructed by introducing two additional imaginary parts.

- The author uses the **three notations** shown below — sometimes intermixed according to the best fitting form in context. Despite these notations one can find a myriad of other variants in literature, which will not be used by the author[1].

Hypercomplex Notation	Vector Notation	Quadruple Notation
$q = q_0 + \mathrm{i}q_1 + \mathrm{j}q_2 + \mathrm{k}q_3$	$q = [q_0, \mathbf{q}] = \begin{bmatrix} q_0, \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \end{bmatrix}$	$q = [q_0, q_1, q_2, q_3]$

In the equations shown above, $q \in \mathbb{H}$ is the quaternion, the real numbers $q_0, q_1, q_2, q_3 \in \mathbb{R}$ are the **components of the quaternion** q, and i, j and k are the three **imaginary units**, whereas[2]:

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

$$\mathrm{ij} = \mathrm{k} = -\mathrm{ji}, \quad \mathrm{jk} = \mathrm{i} = -\mathrm{kj}, \quad \mathrm{ki} = \mathrm{j} = -\mathrm{ik} \tag{2}$$

The real number $q_0 = \mathfrak{Re}(q)$ is the **real part** (also called **scalar part**) of the quaternion q, while the vector $\mathbf{q} = (q_1, q_2, q_3)^\mathrm{T} = \mathfrak{Im}(q)$ is the three-dimensional **imaginary part**[3] (also called **vector part**) of q.

- Quaternions do not satisfy the field axioms[4]; they are violating the axiom of commutativity of multiplication (all other axioms are satisfied) [4, p. 6]. Having said this, the **quaternion multiplication is non-commutative**; therefore

$$q_a q_b \neq q_b q_a, \quad q_a, q_b \in \mathbb{H}. \tag{3}$$

- Special groups of quaternions:
  - Quaternions with a real part of zero are called **pure quaternions**; the set of all pure quaternions is denoted by $\mathbb{H}_0 \subset \mathbb{H}$.
  - A quaternion with a norm[5] of 1 is called a **unit quaternion**; the set of all unit quaternions is denoted by $\mathbb{H}_1 \subset \mathbb{H}$.

---

1 In space science and engineering it is usual to use the quadruple notation, but with the real part at the end: $q = [q_1, q_2, q_3, q_0]$.

2 Warning: These are not the usual dot products — they are (non-commutative) *quaternion products*.

3 This is possible since i, j and k can form an orthonormal base in $\mathbb{R}^3$ using the correspondences $\mathrm{i} \mathrel{\widehat{=}} \mathbf{i} = (1,0,0)^\mathrm{T}$, $\mathrm{j} \mathrel{\widehat{=}} \mathbf{j} = (0,1,0)^\mathrm{T}$ and $\mathrm{k} \mathrel{\widehat{=}} \mathbf{k} = (0,0,1)^\mathrm{T}$. This way, the cross products of the unit vectors $\mathbf{e}_x = (1,0,0)^\mathrm{T}$, $\mathbf{e}_y = (0,1,0)^\mathrm{T}$, $\mathbf{e}_z = (0,0,1)^\mathrm{T}$ of the standard orthonormal base of $\mathbb{R}^3$ give $\mathbf{e}_x \times \mathbf{e}_y = \mathbf{e}_z = -\mathbf{e}_y \times \mathbf{e}_x$, $\mathbf{e}_y \times \mathbf{e}_z = \mathbf{e}_x = -\mathbf{e}_z \times \mathbf{e}_y$, $\mathbf{e}_z \times \mathbf{e}_x = \mathbf{e}_y = -\mathbf{e}_x \times \mathbf{e}_z$, which can be identified with equation set 2.

4 In abstract algebra, a field is defined as an algebraic structure with the two operations of addition and multiplication, satisfying the axioms of (1) closure under all operations, (2) associativity and commutativity of all operations, (3) existence of neutral elements and inverses for addition and subtraction and (4) distributivity of multiplication over addition. The operations of subtraction and division are implicitly defined as inverse operations of addition and multiplication, respectively. [4, pp. 5 et seq.]

5 Refer to section 2 for a definition of the quaternion norm.

- There also exist other **representation forms** for quaternions:
  - *Complex matrix representation*: $q = q_0 + \mathrm{i}q_1 + \mathrm{j}q_2 + \mathrm{k}q_3 \mapsto q_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + q_1 \begin{bmatrix} \mathrm{i}_\mathbb{C} & 0 \\ 0 & -\mathrm{i}_\mathbb{C} \end{bmatrix} + q_2 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} + q_3 \begin{bmatrix} 0 & \mathrm{i}_\mathbb{C} \\ \mathrm{i}_\mathbb{C} & 0 \end{bmatrix} = \begin{bmatrix} q_0 + \mathrm{i}_\mathbb{C} q_1 & q_2 + \mathrm{i}_\mathbb{C} q_3 \\ -q_2 + \mathrm{i}_\mathbb{C} q_3 & q_0 - \mathrm{i}_\mathbb{C} q_1 \end{bmatrix}$
  - *Real matrix representation*: $q = q_0 + \mathrm{i}q_1 + \mathrm{j}q_2 + \mathrm{k}q_3 \mapsto q_0 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + q_1 \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} + q_2 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} + q_3 \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix}$

2 Quaternion Algebra

2.1 Basic Properties

Equality	$r, s \in \mathbb{H}$	$r = s$ if and only if $r_0 = s_0, r_1 = s_1, r_2 = s_2$ and $r_3 = s_3$
Hypercomplex Conjugate	$q, \bar{q} \in \mathbb{H}$	$\bar{q} = [q_0, -\mathbf{q}]$
Norm	$q \in \mathbb{H}, \|q\| \in \mathbb{R}$	$\|q\| = \sqrt{q\bar{q}} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$
Inverse	$q, q^{-1} \in \mathbb{H}$	$q^{-1} = \frac{\bar{q}}{\|q\|^2} = \frac{\bar{q}}{q\bar{q}}$

2.2 Basic Operations

Addition and Subtraction	$r, s \in \mathbb{H}$	$r \pm s = [r_0 \pm s_0, \mathbf{r} \pm \mathbf{s}]$
Multiplication	$c \in \mathbb{R}, q \in \mathbb{H}$	$c \cdot q = [c \cdot q_0, c \cdot \mathbf{q}]$
	$r, s, t \in \mathbb{H}$	$r \cdot s = t = [r_0, \mathbf{r}] \cdot [s_0, \mathbf{s}] = [r_0 s_0 - \langle \mathbf{r}, \mathbf{s} \rangle, r_0 \mathbf{s} + s_0 \mathbf{r} + \mathbf{r} \times \mathbf{s}]$ or with help of matrix algebra: $\begin{bmatrix} t_0 \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} r_0 & -r_1 & -r_2 & -r_3 \\ r_1 & r_0 & -r_3 & r_2 \\ r_2 & r_3 & r_0 & -r_1 \\ r_3 & -r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} s_0 & -s_1 & -s_2 & -s_3 \\ s_1 & s_0 & s_3 & -s_2 \\ s_2 & -s_3 & s_0 & s_1 \\ s_3 & s_2 & -s_1 & s_0 \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix}$
Division	$c \in \mathbb{R}, q \in \mathbb{H}$	$\frac{q}{c} = \frac{1}{c} \cdot q$
	$r, s \in \mathbb{H}, s \neq 0$	$\frac{r}{s} = r s^{-1} = r \frac{\bar{s}}{\|s\|^2} = r \frac{\bar{s}}{s\bar{s}}$

Normalization	$q \in \mathbb{H}, q \neq 0,$ $\hat{q} \in \mathbb{H}_1$	$\hat{q} = \frac{q}{\|q\|}$
Cross Product	$r, s \in \mathbb{H}$	$r \times s = [0, \mathbf{r} \times \mathbf{s}]$
Dot Product	$r, s \in \mathbb{H}$	$\langle r, s \rangle = r_0 s_0 + r_1 s_1 + r_2 s_2 + r_3 s_3$

2.3 Exponential and Logarithmic Functions

In what follows, $\hat{\mathbf{v}}$ denotes a vector in $\mathbb{R}^3$ normalized to length 1, i.e., $\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ is a unit vector. The principal argument $\varphi \in (-\pi, \pi]$ is defined as $\varphi = \arccos\left(\frac{q_0}{\|q\|}\right)$; the principal value of the natural logarithm can be yield by setting $k = 0$ for $\|\mathbf{q}\| \neq 0$.

Exponential Function	$q \in \mathbb{H}$	$\mathrm{e}^q = \mathrm{e}^{q_0} \left[\cos(\|\mathbf{q}\|), \hat{\mathbf{q}} \cdot \sin(\|\mathbf{q}\|)\right]$
Natural Logarithm	$q \in \mathbb{H}, k \in \mathbb{Z}$	$\ln(q) = \begin{cases} [\ln(\|q\|), (\varphi + 2k\pi)\hat{\mathbf{q}}] & \text{for } \|\mathbf{q}\| \neq 0 \\ [\ln(q_0), 0, 0, 0] & \text{for } \|\mathbf{q}\| = 0, q_0 > 0 \\ [\ln(\lvert q_0 \rvert), \pi, 0, 0] & \text{for } \|\mathbf{q}\| = 0, q_0 < 0 \\ \text{undefined} & \text{otherwise} \end{cases}$
Logarithmic Functions	$q \in \mathbb{H}, b \in \mathbb{R}$	$\log_b(q) = \frac{\ln(q)}{\ln(b)}, \quad \lg(q) = \log_{10}(q) = \frac{\ln(q)}{\ln(10)}$
Power Functions	$q \in \mathbb{H}, p \in \mathbb{R}$	$q^p = \mathrm{e}^{p \ln(q)}$
Root Functions	$q \in \mathbb{H}, n \in \mathbb{N}^+$	$\sqrt[n]{q} = \mathrm{e}^{\frac{1}{n} \ln(q)}$

2.4 Trigonometric and Hyperbolic Functions

In what follows, $\hat{\mathbf{v}}$ denotes a vector in $\mathbb{R}^3$ normalized to length 1, i.e., $\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ is a unit vector.

Sine	$q \in \mathbb{H}$	$\sin(q) = -\hat{\mathbf{q}} \left(\mathrm{e}^{\hat{\mathbf{q}} q} - \mathrm{e}^{-\hat{\mathbf{q}} q}\right)$
Hyperbolic Sine	$q \in \mathbb{H}$	$\sinh(q) = -\hat{\mathbf{q}} \left(\mathrm{e}^{q} - \mathrm{e}^{-q}\right)$
Cosine	$q \in \mathbb{H}$	$\cos(q) = \frac{1}{2} \left(\mathrm{e}^{\hat{\mathbf{q}} q} + \mathrm{e}^{-\hat{\mathbf{q}} q}\right)$
Hyperbolic Cosine	$q \in \mathbb{H}$	$\cosh(q) = \frac{1}{2} \left(\mathrm{e}^{q} + \mathrm{e}^{-q}\right)$
Tangent	$q \in \mathbb{H}$	$\tan(q) = \frac{\sin(q)}{\cos(q)}$
Hyperbolic Tangent	$q \in \mathbb{H}$	$\tanh(q) = \frac{\sinh(q)}{\cosh(q)}$
Cotangent	$q \in \mathbb{H}$	$\cot(q) = \frac{\cos(q)}{\sin(q)}$
Hyperbolic Cotangent	$q \in \mathbb{H}$	$\coth(q) = \frac{\cosh(q)}{\sinh(q)}$

3 Spatial Rotations using Quaternions

There is a 1:1 correspondence between a vector $\mathbf{v} \in \mathbb{R}^3$ and a pure quaternion $v \in \mathbb{H}_0$:

$$\mathbf{v} \leftrightarrow v = 0 + \mathrm{i}\mathbf{v}_x + \mathrm{j}\mathbf{v}_y + \mathrm{k}\mathbf{v}_z \qquad \text{or} \qquad \mathbf{v} \leftrightarrow v = [0, \mathbf{v}] \tag{4}$$

This means, every vector in 3-dimensional space represents a pure quaternion and every pure quaternion represents a vector in 3-dimensional space. This is important, because it allows a quaternion to operate on vectors of $\mathbb{R}^3$ [4, pp. 114 et seq.]. A rotated vector $\mathbf{v}^\star$ or ${}^\star\mathbf{v}$, respectively, in $\mathbb{R}^3$ originates in a quaternion multiplication involving the rotation quaternion $q_R(\alpha, \mathbf{u})$, the quaternion v of its original vector $\mathbf{v}$ and the conjugate $\bar{q}_R(\alpha, \mathbf{u})$ of the rotation quaternion in one of two fixed sequences, called *quaternion rotation operators*:

$$v^\star = [0, \mathbf{v}^\star] = q_R(\alpha, \mathbf{u}) \cdot [0, \mathbf{v}] \cdot \overline{q_R}(\alpha, \mathbf{u}) \qquad {}^\star v = [0, {}^\star\mathbf{v}] = \overline{q_R}(\alpha, \mathbf{u}) \cdot [0, \mathbf{v}] \cdot q_R(\alpha, \mathbf{u}) \tag{5}$$

or written in the more compact form

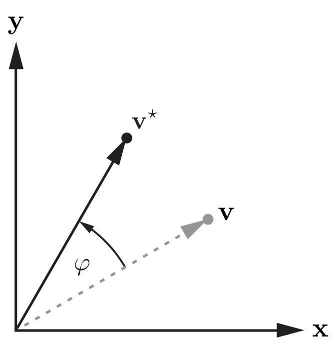
$$\mathbf{v}^\star = q_R \mathbf{v} \overline{q_R} \qquad {}^\star\mathbf{v} = \overline{q_R} \mathbf{v} q_R \tag{6}$$

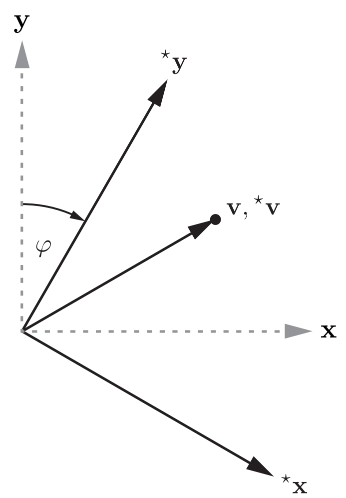
with the rotation quaternion

$$q_R(\alpha, \mathbf{u}) = \left[\cos\frac{\alpha}{2}, \sin\frac{\alpha}{2} \cdot \mathbf{u}\right], \tag{7}$$

whereas α is the rotation angle and $\mathbf{u} \in \mathbb{R}^3$ is the axis of rotation with a norm of 1 ($\|\mathbf{u}\| = 1$; $\mathbf{u}$ is therefore a unit vector) [4, pp. 118 et seq.]. Having said this, such a quaternion multiplication always results in a pure quaternion, which represents a (new) vector in 3-dimensional space. [4, pp. 119 et seq.]

Both operators $\mathbf{v}^\star$ and ${}^\star\mathbf{v}$ define the same rotation angle, but the opposite rotation direction around the chosen axis. The rotation direction depends on the perspective of the rotation as well as on the handedness of the reference frame. Assuming a right-handed coordinate system (right-hand rule applies), the operator $\mathbf{v}^\star = q_R \mathbf{v} \overline{q_R}$ causes a *rotation*, while ${}^\star\mathbf{v} = \overline{q_R} \mathbf{v} q_R$ will cause a *transformation.*

<table>
<tr><th>Rotation
$\mathbf{v}^\star = q_R \mathbf{v} \overline{q_R}$</th><th>Transformation
${}^\star\mathbf{v} = \overline{q_R} \mathbf{v} q_R$</th></tr>
<tr><td>


</td><td>


</td></tr>
<tr><td>The rotation is observed from a fixed position with respect to the coordinate frame. A point $\mathbf{v}$ will be rotated by angle φ around the z-axis. Without loss of generality, the rotated point $\mathbf{v}^\star$ has coordinates different from $\mathbf{v}$ within this frame. In a right-handed coordinate frame, this rotation is mathematical positive for a positive angle φ, i. e., it is counter-clockwise.</td><td>The rotation is observed from a fixed position with respect to the vector (point) $\mathbf{v}$ within the x-y-frame. This frame is now rotated by angle φ around the z-axis; the resulting frame is the ${}^\star\mathbf{x}$-${}^\star\mathbf{y}$-frame. The vector maintains its position within the x-y-frame, but in ${}^\star\mathbf{x}$-${}^\star\mathbf{y}$-frame the point has seemingly been rotated to a new position ${}^\star\mathbf{v}$. For the observer, the coordinate frame has been rotated mathematical negative for a positive angle φ around the z-axis, that is, clockwise.</td></tr>
</table>

Both operations can also be carried out using matrices, which is often more suitable for programming[6]:

$$\mathbf{v}^\star = 2\begin{bmatrix} q_0^2 - 0.5 + q_1^2 & q_1q_2 - q_0q_3 & q_1q_3 + q_0q_2 \\ q_1q_2 + q_0q_3 & q_0^2 - 0.5 + q_2^2 & q_2q_3 - q_0q_1 \\ q_1q_3 - q_0q_2 & q_2q_3 + q_0q_1 & q_0^2 - 0.5 + q_3^2 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad \text{and} \tag{8}$$

$$^\star\mathbf{v} = 2\begin{bmatrix} q_0^2 - 0.5 + q_1^2 & q_1q_2 + q_0q_3 & q_1q_3 - q_0q_2 \\ q_1q_2 - q_0q_3 & q_0^2 - 0.5 + q_2^2 & q_2q_3 + q_0q_1 \\ q_1q_3 + q_0q_2 & q_2q_3 - q_0q_1 & q_0^2 - 0.5 + q_3^2 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{9}$$

4 Conversions

4.1 Euler Angles to Quaternion

Inputs	**Outputs**
■ Rotation sequence (e. g. X–Y–Z to first rotate around the X-axis, then about the Y-axis and finally about the Z-axis) ■ Three angles α, β and γ applied successively to the rotation sequence (e. g. α applied to rotation about X-axis, β applied to rotation about Y-axis, γ applied to rotation about Z-axis for the aforementioned example)	■ Quaternion $q = [q_0, q_1, q_2, q_3]$ (q_0 is the scalar part)

For the sake of readability, the following shorthands are used in the table below:

$$s_\alpha \stackrel{!}{=} \sin\frac{\alpha}{2} \qquad c_\alpha \stackrel{!}{=} \cos\frac{\alpha}{2}$$
$$s_\beta \stackrel{!}{=} \sin\frac{\beta}{2} \qquad c_\beta \stackrel{!}{=} \cos\frac{\beta}{2}$$
$$s_\gamma \stackrel{!}{=} \sin\frac{\gamma}{2} \qquad c_\gamma \stackrel{!}{=} \cos\frac{\gamma}{2}$$

Rotation Sequence	**Conversion Formulae** *for rotations*
X–Y–X	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\ s_\alpha s_\beta c_\gamma - c_\alpha s_\beta s_\gamma]$
X–Y–Z	$q = [c_\alpha c_\beta c_\gamma - s_\alpha s_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha s_\beta s_\gamma,\ c_\alpha s_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\ s_\alpha s_\beta c_\gamma + c_\alpha c_\beta s_\gamma]$
X–Z–X	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\ c_\alpha s_\beta s_\gamma - s_\alpha s_\beta c_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma]$
X–Z–Y	$q = [c_\alpha c_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma - c_\alpha s_\beta s_\gamma,\ c_\alpha c_\beta s_\gamma - s_\alpha s_\beta c_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha c_\beta s_\gamma]$
Y–X–Y	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\ c_\alpha s_\beta s_\gamma - s_\alpha s_\beta c_\gamma]$
Y–X–Z	$q = [c_\alpha c_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha c_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma - c_\alpha s_\beta s_\gamma,\ c_\alpha c_\beta s_\gamma - s_\alpha s_\beta c_\gamma]$
Y–Z–X	$q = [c_\alpha c_\beta c_\gamma - s_\alpha s_\beta s_\gamma,\ s_\alpha s_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha s_\beta s_\gamma,\ c_\alpha s_\beta c_\gamma - s_\alpha c_\beta s_\gamma]$
Y–Z–Y	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\ s_\alpha s_\beta c_\gamma - c_\alpha s_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma]$
Z–X–Y	$q = [c_\alpha c_\beta c_\gamma - s_\alpha s_\beta s_\gamma,\ c_\alpha s_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\ s_\alpha s_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha s_\beta s_\gamma]$
Z–X–Z	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\ s_\alpha s_\beta c_\gamma - c_\alpha s_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma]$
Z–Y–X	$q = [c_\alpha c_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\ c_\alpha c_\beta s_\gamma - s_\alpha s_\beta c_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha c_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma - c_\alpha s_\beta s_\gamma]$
Z–Y–Z	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\ c_\alpha s_\beta s_\gamma - s_\alpha s_\beta c_\gamma,\ c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\ s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma]$

[6] For the sake of simplicity, indices and arguments of the rotation quaternion have been omitted, i. e., $q_0 = q_{R,0}(\alpha, \mathbf{u})$ is the scalar component of the rotation quaternion q_R for a rotation of angle α about the axis $\mathbf{u}$.

Rotation Sequence	**Conversion Formulae** *for transformations*
X–Y–X	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\quad c_\alpha s_\beta s_\gamma - s_\alpha s_\beta c_\gamma]$
X–Y–Z	$q = [c_\alpha c_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\quad s_\alpha c_\beta c_\gamma - c_\alpha s_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha c_\beta s_\gamma,\quad c_\alpha c_\beta s_\gamma - s_\alpha s_\beta c_\gamma]$
X–Z–X	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\quad s_\alpha s_\beta c_\gamma - c_\alpha s_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma]$
X–Z–Y	$q = [c_\alpha c_\beta c_\gamma - s_\alpha s_\beta s_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha s_\beta s_\gamma,\quad s_\alpha s_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma - s_\alpha c_\beta s_\gamma]$
Y–X–Y	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\quad s_\alpha s_\beta c_\gamma - c_\alpha s_\beta s_\gamma]$
Y–X–Z	$q = [c_\alpha c_\beta c_\gamma - s_\alpha s_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha s_\beta s_\gamma,\quad s_\alpha s_\beta c_\gamma + c_\alpha c_\beta s_\gamma]$
Y–Z–X	$q = [c_\alpha c_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\quad c_\alpha c_\beta s_\gamma - s_\alpha s_\beta c_\gamma,\quad s_\alpha c_\beta c_\gamma - c_\alpha s_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha c_\beta s_\gamma]$
Y–Z–Y	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\quad c_\alpha s_\beta s_\gamma - s_\alpha s_\beta c_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma]$
Z–X–Y	$q = [c_\alpha c_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha c_\beta s_\gamma,\quad c_\alpha c_\beta s_\gamma - s_\alpha s_\beta c_\gamma,\quad s_\alpha c_\beta c_\gamma - c_\alpha s_\beta s_\gamma]$
Z–X–Z	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\quad c_\alpha s_\beta s_\gamma - s_\alpha s_\beta c_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma]$
Z–Y–X	$q = [c_\alpha c_\beta c_\gamma - s_\alpha s_\beta s_\gamma,\quad s_\alpha s_\beta c_\gamma + c_\alpha c_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha s_\beta s_\gamma]$
Z–Y–Z	$q = [c_\alpha c_\beta c_\gamma - s_\alpha c_\beta s_\gamma,\quad s_\alpha s_\beta c_\gamma - c_\alpha s_\beta s_\gamma,\quad c_\alpha s_\beta c_\gamma + s_\alpha s_\beta s_\gamma,\quad s_\alpha c_\beta c_\gamma + c_\alpha c_\beta s_\gamma]$

4.2 Quaternion to Euler Angles

Inputs	**Outputs**
■ Quaternion $q = [q_0, q_1, q_2, q_3]$ (q_0 is the scalar part) ■ Rotation sequence (e. g. X–Y–Z to first rotate around the X-axis, then about the Y-axis and finally about the Z-axis)	■ Three angles α, β and γ applied successively to the rotation sequence (e. g. α applied to rotation about X-axis, β applied to rotation about Y-axis, γ applied to rotation about Z-axis for the example mentioned on the left)

Rotation Sequence	**Conversion Formulae** *for rotations*		
X–Y–X	$\tan\alpha = \dfrac{q_0q_3 - q_1q_2}{q_0q_2 + q_1q_3}$	$\cos\beta = 2(q_0^2 + q_1^2) - 1$	$\tan\gamma = \dfrac{q_1q_2 + q_0q_3}{q_1q_3 - q_0q_2}$
X–Y–Z	$\tan\alpha = \dfrac{q_0q_1 + q_2q_3}{0.5 - q_0^2 - q_3^2}$	$\sin\beta = 2(q_1q_3 - q_0q_2)$	$\tan\gamma = \dfrac{q_0q_3 + q_1q_2}{0.5 - q_0^2 - q_1^2}$
X–Z–X	$\tan\alpha = \dfrac{q_1q_3 + q_0q_2}{q_1q_2 - q_0q_3}$	$\cos\beta = 2(q_0^2 + q_1^2) - 1$	$\tan\gamma = \dfrac{q_0q_2 - q_1q_3}{q_0q_3 + q_1q_2}$
X–Z–Y	$\tan\alpha = \dfrac{q_0q_1 - q_2q_3}{0.5 - q_0^2 - q_2^2}$	$\sin\beta = -2(q_0q_3 + q_1q_2)$	$\tan\gamma = \dfrac{q_0q_2 - q_1q_3}{0.5 - q_0^2 - q_1^2}$
Y–X–Y	$\tan\alpha = \dfrac{q_1q_2 + q_0q_3}{q_2q_3 - q_0q_1}$	$\cos\beta = 2(q_0^2 + q_2^2) - 1$	$\tan\gamma = \dfrac{q_0q_3 - q_1q_2}{q_0q_1 + q_2q_3}$
Y–X–Z	$\tan\alpha = \dfrac{q_0q_2 - q_1q_3}{0.5 - q_0^2 - q_3^2}$	$\sin\beta = -2(q_0q_1 + q_2q_3)$	$\tan\gamma = \dfrac{q_0q_3 - q_1q_2}{0.5 - q_0^2 - q_2^2}$
Y–Z–X	$\tan\alpha = \dfrac{q_0q_2 + q_1q_3}{0.5 - q_0^2 - q_1^2}$	$\sin\beta = 2(q_1q_2 - q_0q_3)$	$\tan\gamma = \dfrac{q_0q_1 + q_2q_3}{0.5 - q_0^2 - q_2^2}$
Y–Z–Y	$\tan\alpha = \dfrac{q_0q_1 - q_2q_3}{q_1q_2 + q_0q_3}$	$\cos\beta = 2(q_0^2 + q_2^2) - 1$	$\tan\gamma = \dfrac{q_0q_1 + q_2q_3}{q_1q_2 - q_0q_3}$
Z–X–Y	$\tan\alpha = \dfrac{q_0q_3 + q_1q_2}{0.5 - q_0^2 - q_2^2}$	$\sin\beta = 2(q_2q_3 - q_0q_1)$	$\tan\gamma = \dfrac{q_0q_2 + q_1q_3}{0.5 - q_0^2 - q_3^2}$
Z–X–Z	$\tan\alpha = \dfrac{q_0q_2 - q_1q_3}{q_0q_1 + q_2q_3}$	$\cos\beta = 2(q_0^2 + q_3^2) - 1$	$\tan\gamma = \dfrac{q_0q_2 + q_1q_3}{q_2q_3 - q_0q_1}$

Z–Y–X	$\tan\alpha = \dfrac{q_0q_3 - q_1q_2}{0.5 - q_0^2 - q_1^2}$	$\sin\beta = -2(q_0q_2 + q_1q_3)$	$\tan\gamma = \dfrac{q_0q_1 - q_2q_3}{0.5 - q_0^2 - q_3^2}$
Z–Y–Z	$\tan\alpha = \dfrac{q_0q_1 + q_2q_3}{q_1q_3 - q_0q_2}$	$\cos\beta = 2(q_0^2 + q_3^2) - 1$	$\tan\gamma = \dfrac{q_0q_1 - q_2q_3}{q_0q_2 + q_1q_3}$
Rotation Sequence	**Conversion Formulae** *for transformations*		
X–Y–X	$\tan\alpha = \dfrac{q_1q_2 + q_0q_3}{q_0q_2 - q_1q_3}$	$\cos\beta = 2(q_0^2 + q_1^2) - 1$	$\tan\gamma = \dfrac{q_1q_2 - q_0q_3}{q_1q_3 + q_0q_2}$
X–Y–Z	$\tan\alpha = \dfrac{q_0q_1 - q_2q_3}{q_0^2 + q_3^2 - 0.5}$	$\sin\beta = 2(q_1q_3 + q_0q_2)$	$\tan\gamma = \dfrac{q_0q_3 - q_1q_2}{q_0^2 + q_1^2 - 0.5}$
X–Z–X	$\tan\alpha = \dfrac{q_1q_3 - q_0q_2}{q_1q_2 + q_0q_3}$	$\cos\beta = 2(q_0^2 + q_1^2) - 1$	$\tan\gamma = \dfrac{q_1q_3 + q_0q_2}{q_0q_3 - q_1q_2}$
X–Z–Y	$\tan\alpha = \dfrac{q_0q_1 + q_2q_3}{q_0^2 + q_2^2 - 0.5}$	$\sin\beta = 2(q_0q_3 - q_1q_2)$	$\tan\gamma = \dfrac{q_0q_2 + q_1q_3}{q_0^2 + q_1^2 - 0.5}$
Y–X–Y	$\tan\alpha = \dfrac{q_1q_2 - q_0q_3}{q_0q_1 + q_2q_3}$	$\cos\beta = 2(q_0^2 + q_2^2) - 1$	$\tan\gamma = \dfrac{q_1q_2 + q_0q_3}{q_0q_1 - q_2q_3}$
Y–X–Z	$\tan\alpha = \dfrac{q_0q_2 + q_1q_3}{q_0^2 + q_3^2 - 0.5}$	$\sin\beta = 2(q_0q_1 - q_2q_3)$	$\tan\gamma = \dfrac{q_1q_2 + q_0q_3}{q_0^2 + q_2^2 - 0.5}$
Y–Z–X	$\tan\alpha = \dfrac{q_0q_2 - q_1q_3}{q_0^2 + q_1^2 - 0.5}$	$\sin\beta = 2(q_1q_2 + q_0q_3)$	$\tan\gamma = \dfrac{q_0q_1 - q_2q_3}{q_0^2 + q_2^2 - 0.5}$
Y–Z–Y	$\tan\alpha = \dfrac{q_0q_1 + q_2q_3}{q_0q_3 - q_1q_2}$	$\cos\beta = 2(q_0^2 + q_2^2) - 1$	$\tan\gamma = \dfrac{q_2q_3 - q_0q_1}{q_1q_2 + q_0q_3}$
Z–X–Y	$\tan\alpha = \dfrac{q_0q_3 - q_1q_2}{q_0^2 + q_2^2 - 0.5}$	$\sin\beta = 2(q_0q_1 + q_2q_3)$	$\tan\gamma = \dfrac{q_0q_2 - q_1q_3}{q_0^2 + q_3^2 - 0.5}$
Z–X–Z	$\tan\alpha = \dfrac{q_0q_2 + q_1q_3}{q_0q_1 - q_2q_3}$	$\cos\beta = 2(q_0^2 + q_3^2) - 1$	$\tan\gamma = \dfrac{q_1q_3 - q_0q_2}{q_0q_1 + q_2q_3}$
Z–Y–X	$\tan\alpha = \dfrac{q_1q_2 + q_0q_3}{q_0^2 + q_1^2 - 0.5}$	$\sin\beta = 2(q_0q_2 - q_1q_3)$	$\tan\gamma = \dfrac{q_0q_1 + q_2q_3}{q_0^2 + q_3^2 - 0.5}$
Z–Y–Z	$\tan\alpha = \dfrac{q_2q_3 - q_0q_1}{q_0q_2 + q_1q_3}$	$\cos\beta = 2(q_0^2 + q_3^2) - 1$	$\tan\gamma = \dfrac{q_0q_1 + q_2q_3}{q_0q_2 - q_1q_3}$

4.3 Direction Cosine Matrix (DCM) to Quaternion

Inputs	Outputs
Direction cosine matrix A (orthogonal or near-orthogonal): $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$	Quaternion $q = [q_0, q_1, q_2, q_3]$ (q_0 is the scalar part)

This algorithm is also suitable for imprecise direction cosine matrices, i.e., near-orthogonal ones. The algorithm is entirely taken from [1].

1. Construct the auxiliary matrix K_3 as follows:

$$K_3 = \frac{1}{3} \begin{bmatrix} a_{11} - a_{22} - a_{33} & a_{21} + a_{12} & a_{31} + a_{13} & a_{23} - a_{32} \\ a_{21} + a_{12} & a_{22} - a_{11} - a_{33} & a_{32} + a_{23} & a_{31} - a_{13} \\ a_{31} + a_{13} & a_{32} + a_{23} & a_{33} - a_{11} - a_{22} & a_{12} - a_{21} \\ a_{23} - a_{32} & a_{31} - a_{13} & a_{12} - a_{21} & a_{11} + a_{22} + a_{33} \end{bmatrix} \tag{10}$$

2. Compute the eigenvalues of K_3.

3. Determine λ_{max}, the largest eigenvalue of K_3.
4. Calculate the eigenvector **b** of K_3 for the corresponding eigenvalue λ_{max}.
5. The quaternion q is now $q = b_4 + \mathrm{i}b_1 + \mathrm{j}b_2 + \mathrm{k}b_3$.

4.4 Quaternion to Direction Cosine Matrix (DCM)

Inputs	Outputs
■ Quaternion $q = [q_0, q_1, q_2, q_3]$ (q_0 is the scalar part)	■ Direction cosine matrix A

	Conversion Formula
for rotations	$A = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$
for transformations	$A = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$

4.5 Euler Angles to Rotation Matrix

Inputs	Outputs
■ Rotation sequence (e. g. X–Y–Z to first rotate around the X-axis, then about the Y-axis and finally about the Z-axis) ■ Three angles α, β and γ applied successively to the rotation sequence (e. g. α applied to rotation about X-axis, β applied to rotation about Y-axis, γ applied to rotation about Z-axis for the aforementioned example)	■ Rotation matrix A

Rotation Sequence	Conversion Formulae *for rotations*
X–Y–X	$A = \begin{bmatrix} \cos\beta & \sin\alpha\sin\beta & -\cos\alpha\sin\beta \\ \sin\beta\sin\gamma & \cos\alpha\cos\gamma-\sin\alpha\cos\beta\sin\gamma & \sin\alpha\cos\gamma+\cos\alpha\cos\beta\sin\gamma \\ \sin\beta\cos\gamma & -\sin\alpha\cos\beta\cos\gamma-\cos\alpha\sin\gamma & \cos\alpha\cos\beta\cos\gamma-\sin\alpha\sin\gamma \end{bmatrix}$
X–Y–Z	$A = \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma+\cos\alpha\sin\gamma & \sin\alpha\sin\gamma-\cos\alpha\sin\beta\cos\gamma \\ -\cos\beta\sin\gamma & \cos\alpha\cos\gamma-\sin\alpha\sin\beta\sin\gamma & \sin\alpha\cos\gamma+\cos\alpha\sin\beta\sin\gamma \\ \sin\beta & -\sin\alpha\cos\beta & \cos\alpha\cos\beta \end{bmatrix}$
X–Z–X	$A = \begin{bmatrix} \cos\beta & \cos\alpha\sin\beta & \sin\alpha\sin\beta \\ -\sin\beta\cos\gamma & \cos\alpha\cos\beta\cos\gamma-\sin\alpha\sin\gamma & \sin\alpha\cos\beta\cos\gamma+\cos\alpha\sin\gamma \\ \sin\beta\sin\gamma & -\sin\alpha\cos\gamma-\cos\alpha\cos\beta\sin\gamma & \cos\alpha\cos\gamma-\sin\alpha\cos\beta\sin\gamma \end{bmatrix}$

X–Z–Y	$A = \begin{bmatrix} \cos\beta\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\alpha\cos\beta & \sin\alpha\cos\beta \\ \cos\beta\sin\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma \end{bmatrix}$
Y–X–Y	$A = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma & -\sin\alpha\cos\gamma - \cos\alpha\cos\beta\sin\gamma \\ \sin\alpha\sin\beta & \cos\beta & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\beta\cos\gamma & \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma \end{bmatrix}$
Y–X–Z	$A = \begin{bmatrix} \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & \cos\beta\sin\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma \\ \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\beta\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & -\sin\beta & \cos\alpha\cos\beta \end{bmatrix}$
Y–Z–X	$A = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & -\sin\alpha\cos\beta \\ \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma \\ \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & -\cos\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma \end{bmatrix}$
Y–Z–Y	$A = \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & \sin\beta\cos\gamma & -\sin\alpha\cos\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\cos\alpha\sin\beta & \cos\beta & \sin\alpha\sin\beta \\ \sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma \end{bmatrix}$
Z–X–Y	$A = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & -\cos\beta\sin\gamma \\ -\sin\alpha\cos\beta & \cos\alpha\cos\beta & \sin\beta \\ \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma & \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & \cos\beta\cos\gamma \end{bmatrix}$
Z–X–Z	$A = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & \sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma \\ -\sin\alpha\cos\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & \sin\beta\cos\gamma \\ \sin\alpha\sin\beta & -\cos\alpha\sin\beta & \cos\beta \end{bmatrix}$
Z–Y–X	$A = \begin{bmatrix} \cos\alpha\cos\beta & \sin\alpha\cos\beta & -\sin\beta \\ \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & \cos\beta\sin\gamma \\ \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$
Z–Y–Z	$A = \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\beta\cos\gamma \\ -\sin\alpha\cos\gamma - \cos\alpha\cos\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma \\ \cos\alpha\sin\beta & \sin\alpha\sin\beta & \cos\beta \end{bmatrix}$

Rotation Sequence	**Conversion Formulae** *for transformations*
X–Y–X	$A = \begin{bmatrix} \cos\beta & \sin\alpha\sin\beta & \cos\alpha\sin\beta \\ \sin\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & -\sin\alpha\cos\gamma - \cos\alpha\cos\beta\sin\gamma \\ -\sin\beta\cos\gamma & \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma \end{bmatrix}$
X–Y–Z	$A = \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \sin\alpha\sin\gamma + \cos\alpha\sin\beta\cos\gamma \\ \cos\beta\sin\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{bmatrix}$
X–Z–X	$A = \begin{bmatrix} \cos\beta & -\cos\alpha\sin\beta & \sin\alpha\sin\beta \\ \sin\beta\cos\gamma & \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\sin\alpha\cos\beta\cos\gamma - \cos\alpha\sin\gamma \\ \sin\beta\sin\gamma & \sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma \end{bmatrix}$

X–Z–Y	$A = \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma \\ \sin\beta & \cos\alpha\cos\beta & -\sin\alpha\cos\beta \\ -\cos\beta\sin\gamma & \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma \end{bmatrix}$
Y–X–Y	$A = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma & \sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma \\ \sin\alpha\sin\beta & \cos\beta & -\cos\alpha\sin\beta \\ -\sin\alpha\cos\beta\cos\gamma - \cos\alpha\sin\gamma & \sin\beta\cos\gamma & \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma \end{bmatrix}$
Y–X–Z	$A = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\cos\beta\sin\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma \\ \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma & \cos\beta\cos\gamma & \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma \\ -\sin\alpha\cos\beta & \sin\beta & \cos\alpha\cos\beta \end{bmatrix}$
Y–Z–X	$A = \begin{bmatrix} \cos\alpha\cos\beta & -\sin\beta & \sin\alpha\cos\beta \\ \sin\alpha\sin\gamma + \cos\alpha\sin\beta\cos\gamma & \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\beta\sin\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma \end{bmatrix}$
Y–Z–Y	$A = \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\sin\beta\cos\gamma & \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma \\ \cos\alpha\sin\beta & \cos\beta & \sin\alpha\sin\beta \\ -\sin\alpha\cos\gamma - \cos\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma \end{bmatrix}$
Z–X–Y	$A = \begin{bmatrix} \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\beta\sin\gamma \\ \sin\alpha\cos\beta & \cos\alpha\cos\beta & -\sin\beta \\ \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \sin\alpha\sin\gamma + \cos\alpha\sin\beta\cos\gamma & \cos\beta\cos\gamma \end{bmatrix}$
Z–X–Z	$A = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & -\sin\alpha\cos\gamma - \cos\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma \\ \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\sin\beta\cos\gamma \\ \sin\alpha\sin\beta & \cos\alpha\sin\beta & \cos\beta \end{bmatrix}$
Z–Y–X	$A = \begin{bmatrix} \cos\alpha\cos\beta & -\sin\alpha\cos\beta & \sin\beta \\ \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma & \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\cos\beta\sin\gamma \\ \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$
Z–Y–Z	$A = \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\sin\alpha\cos\beta\cos\gamma - \cos\alpha\sin\gamma & \sin\beta\cos\gamma \\ \sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma \\ -\cos\alpha\sin\beta & \sin\alpha\sin\beta & \cos\beta \end{bmatrix}$

5 Derivative of a Rotation Quaternion

5.1 Variant $q_R(t + \Delta t) = q_R(\Delta t)q_R(t)$

In general, a time-dependent rotation quaternion $q_R(t)$ can be written as

$$q_R(t) = q_R(\alpha(t), \mathbf{u}(t)) = \left[\cos\frac{\alpha(t)}{2}, \mathbf{u}(t)\sin\frac{\alpha(t)}{2}\right], \tag{11}$$

whereas $\alpha(t)$ is the rotation angle about the given axis of rotation $\mathbf{u}(t)$ at a given time t. As known from basic differential calculus, the derivative of a function $f(x)$ can be defined via the limit of its difference quotient:

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{12}$$

Considering a time $t + \Delta t \geq t$, the derivative of the rotation quaternion $q_R(t)$ with respect to time is

$$\frac{\mathrm{d}q_R(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{q_R(t + \Delta t) - q_R(t)}{\Delta t} \tag{13}$$

and one may define the resulting rotation quaternion at time $t + \Delta t$ expressed with reference to the initial orientation $q_R(t)$ then as

$$\begin{aligned} q_R(t+\Delta t) &= q_R(\Delta t) q_R(t) && (14) \\ &= \left[\cos\frac{\alpha(\Delta t)}{2}, \mathbf{u}(t)\sin\frac{\alpha(\Delta t)}{2}\right] q_R(t). && (15) \end{aligned}$$

The rotation angle $\alpha(\Delta t)$ is the increment of the rotation angle α at time $t + \Delta t$ with respect to the rotation angle α at time t. Hence $\alpha(\Delta t)$ can be substituted by $\Delta\alpha$:

$$\left[\cos\frac{\alpha(\Delta t)}{2}, \mathbf{u}(t)\sin\frac{\alpha(\Delta t)}{2}\right] q_R(t) \xlongequal{\alpha(\Delta t) \stackrel{!}{=} \Delta\alpha} \left[\cos\frac{\Delta\alpha}{2}, \mathbf{u}(t)\sin\frac{\Delta\alpha}{2}\right] q_R(t) \tag{16}$$

Additionally, using the small-angle approximation[7] one can obtain

$$\left[\cos\frac{\Delta\alpha}{2}, \mathbf{u}(t)\sin\frac{\Delta\alpha}{2}\right] q_R(t) \xlongequal[\cos\frac{\Delta\alpha}{2}\approx 1]{\sin\frac{\Delta\alpha}{2}\approx\frac{\Delta\alpha}{2}} \left[1, \mathbf{u}(t)\frac{\Delta\alpha}{2}\right] q_R(t) \tag{17}$$

for $q_R(t + \Delta t)$. The evaluation of the difference quotient (eq. 13) yields now

$$\begin{aligned} \frac{\mathrm{d}q_R(t)}{\mathrm{d}t} &= \lim_{\Delta t\to 0} \frac{q_R(t+\Delta t) - q_R(t)}{\Delta t} \\ &= \lim_{\Delta t\to 0} \frac{\left[1, \frac{1}{2}\mathbf{u}(t)\Delta\alpha\right] q_R(t) - q_R(t)}{\Delta t} \\ &= \lim_{\Delta t\to 0} \frac{\left[1, \frac{1}{2}\mathbf{u}(t)\Delta\alpha\right] - 1}{\Delta t} q_R(t) \\ &= \lim_{\Delta t\to 0} \frac{\left[0, \frac{1}{2}\mathbf{u}(t)\Delta\alpha\right]}{\Delta t} q_R(t) \\ &= \frac{1}{2}\left[0, \mathbf{u}(t)\right] \underbrace{\lim_{\Delta t\to 0}\frac{\Delta\alpha}{\Delta t}}_{=\dot{\alpha}=\omega} q_R(t) \\ &= \frac{1}{2}\left[0, \mathbf{u}(t)\omega\right] q_R(t) && (18) \end{aligned}$$

with $\omega = \dot{\alpha}$ being the angular velocity about the rotation axis $\mathbf{u}(t)$. Finally, this equation can be rewritten in terms of a matrix multiplication:

$$\boxed{\frac{\mathrm{d}q_R(t)}{\mathrm{d}t} = \frac{1}{2}\begin{bmatrix} 0 & -u_x\omega & -u_y\omega & -u_z\omega \\ u_x\omega & 0 & -u_z\omega & u_y\omega \\ u_y\omega & u_z\omega & 0 & -u_x\omega \\ u_z\omega & -u_y\omega & u_x\omega & 0 \end{bmatrix}\begin{bmatrix} q_{R,0}(t) \\ q_{R,1}(t) \\ q_{R,2}(t) \\ q_{R,3}(t) \end{bmatrix}} \tag{19}$$

5.2 Variant $q_R(t + \Delta t) = q_R(t)q_R(\Delta t)$

In general, a time-dependent rotation quaternion $q_R(t)$ can be written as

$$q_R(t) = q_R(\alpha(t), \mathbf{u}(t)) = \left[\cos\frac{\alpha(t)}{2}, \mathbf{u}(t)\sin\frac{\alpha(t)}{2}\right], \tag{20}$$

whereas $\alpha(t)$ is the rotation angle about the given axis of rotation $\mathbf{u}(t)$ at a given time t. As known from basic differential calculus, the derivative of a function $f(x)$ can be defined via the limit of its difference quotient:

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lim_{\Delta x\to 0}\frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{21}$$

[7] Using the small-angle approximation for the trigonometric functions is possible as there will be an infinitesimal small change in the rotation angle in an infinitesimal small period in time, i. e. $\Delta\alpha \to 0$ for $\Delta t \to 0$.

Considering a time $t + \Delta t \geq t$, the derivative of the rotation quaternion $q_R(t)$ with respect to time is

$$\frac{\mathrm{d}q_R(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{q_R(t + \Delta t) - q_R(t)}{\Delta t} \tag{22}$$

and one may define the resulting rotation quaternion at time $t + \Delta t$ expressed with reference to the initial orientation $q_R(t)$ then as

$$\begin{aligned} q_R(t + \Delta t) &= q_R(t) q_R(\Delta t) && (23) \\ &= q_R(t) \left[\cos \frac{\alpha(\Delta t)}{2}, \mathbf{u}(t) \sin \frac{\alpha(\Delta t)}{2}\right]. && (24) \end{aligned}$$

The rotation angle $\alpha(\Delta t)$ is the increment of the rotation angle α at time $t + \Delta t$ with respect to the rotation angle α at time t. Hence $\alpha(\Delta t)$ can be substituted by $\Delta\alpha$:

$$q_R(t) \left[\cos \frac{\alpha(\Delta t)}{2}, \mathbf{u}(t) \sin \frac{\alpha(\Delta t)}{2}\right] \xlongequal{\alpha(\Delta t) \overset{!}{=} \Delta\alpha} q_R(t) \left[\cos \frac{\Delta\alpha}{2}, \mathbf{u}(t) \sin \frac{\Delta\alpha}{2}\right] \tag{25}$$

Additionally, using the small-angle approximation[8] one can obtain

$$q_R(t) \left[\cos \frac{\Delta\alpha}{2}, \mathbf{u}(t) \sin \frac{\Delta\alpha}{2}\right] \xlongequal[\cos \frac{\Delta\alpha}{2} \approx 1]{\sin \frac{\Delta\alpha}{2} \approx \frac{\Delta\alpha}{2}} q_R(t) \left[1, \mathbf{u}(t) \frac{\Delta\alpha}{2}\right] \tag{26}$$

for $q_R(t + \Delta t)$. The evaluation of the difference quotient (eq. 22) yields now

$$\begin{aligned} \frac{\mathrm{d}q_R(t)}{\mathrm{d}t} &= \lim_{\Delta t \to 0} \frac{q_R(t + \Delta t) - q_R(t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{q_R(t) \left[1, \frac{1}{2}\mathbf{u}(t)\Delta\alpha\right] - q_R(t)}{\Delta t} \\ &= q_R(t) \lim_{\Delta t \to 0} \frac{\left[1, \frac{1}{2}\mathbf{u}(t)\Delta\alpha\right] - 1}{\Delta t} \\ &= q_R(t) \lim_{\Delta t \to 0} \frac{\left[0, \frac{1}{2}\mathbf{u}(t)\Delta\alpha\right]}{\Delta t} \\ &= \frac{1}{2} q_R(t) \left[0, \mathbf{u}(t)\right] \underbrace{\lim_{\Delta t \to 0} \frac{\Delta\alpha}{\Delta t}}_{=\dot{\alpha}=\omega} \\ &= \frac{1}{2} q_R(t) \left[0, \mathbf{u}(t)\omega\right] && (27) \end{aligned}$$

with $\omega = \dot{\alpha}$ being the angular velocity about the rotation axis $\mathbf{u}(t)$. Finally, this equation can be rewritten in terms of a matrix multiplication:

$$\boxed{\frac{\mathrm{d}q_R(t)}{\mathrm{d}t} = \frac{1}{2} \begin{bmatrix} 0 & -u_x\omega & -u_y\omega & -u_z\omega \\ u_x\omega & 0 & u_z\omega & -u_y\omega \\ u_y\omega & -u_z\omega & 0 & u_x\omega \\ u_z\omega & u_y\omega & -u_x\omega & 0 \end{bmatrix} \begin{bmatrix} q_{R,0}(t) \\ q_{R,1}(t) \\ q_{R,2}(t) \\ q_{R,3}(t) \end{bmatrix}} \tag{28}$$

[8] Using the small-angle approximation for the trigonometric functions is possible as there will be an infinitesimal small change in the rotation angle in an infinitesimal small period in time, i. e. $\Delta\alpha \to 0$ for $\Delta t \to 0$.

6 Quaternion Interpolation Algorithms

6.1 Linear Interpolation (LERP)

Inputs	**Outputs**
■ Rotation quaternion q_s representing initial state at $s = 0$ ■ Rotation quaternion q_e representing final state at $s = 1$ ■ Interpolation position s on the path between q_s and q_e, $0 \leq s \leq 1$	■ Interpolated quaternion q_i

1. Preparations:
   a) Check if q_s and q_e are unit quaternions (otherwise throw exception):
   $$q_s, q_e \in \mathbb{H}_1 \Leftrightarrow \|q_s\| = \|q_e\| \equiv 1 \tag{29}$$
   b) Check if s is in the range of $[0, 1]$, otherwise throw exception.
2. Compute the interpolated quaternion q_i as
   $$q_i = q_s(1 - s) + q_e s. \tag{30}$$

6.2 Spherical Linear Interpolation (SLERP)

Inputs	**Outputs**
■ Rotation quaternion q_s representing initial state at $s = 0$ ■ Rotation quaternion q_e representing final state at $s = 1$ ■ Interpolation position s on the path between q_s and q_e, $0 \leq s \leq 1$	■ Interpolated quaternion q_i

6.2.1 Algorithm by Dam, Koch and Lillholm

1. Preparations:
   a) Check if q_s and q_e are unit quaternions (otherwise throw exception):
   $$q_s, q_e \in \mathbb{H}_1 \Leftrightarrow \|q_s\| = \|q_e\| \equiv 1 \tag{31}$$
   b) Check if s is in the range of $[0, 1]$, otherwise throw exception.
2. Compute the interpolated quaternion q_i as
   $$q_i = (q_e \bar{q}_s)^s q_s. \tag{32}$$

6.2.2 Algorithm by Bronstein

1. Preparations:
   a) Check if q_s and q_e are unit quaternions (otherwise throw exception):

$$q_s, q_e \in \mathbb{H}_1 \Leftrightarrow \|q_s\| = \|q_e\| \equiv 1 \tag{33}$$

   b) Check if s is in the range of $[0, 1]$, otherwise throw exception.

2. Check for special case:
   a) If $q_s = [1, 0, 0, 0]$:
      i. Extract the angle[9] φ between q_s and q_e out of q_e:

$$\cos\varphi = q_{e,0} \tag{34}$$

      ii. Extract the normalized rotation axis of q_e:

$$\hat{\mathbf{u}}_e = \frac{\mathbf{q}_e}{\sin\varphi} \cdot \frac{1}{\|q_e\|} \tag{35}$$

      iii. Compute the interpolated rotation quaternion q_i as

$$q_i = \cos(s\varphi) + \hat{\mathbf{u}}_e \sin(s\varphi). \tag{36}$$

   b) Otherwise:
      i. Determine the angle[9] φ between q_s and q_e by evaluating the dot product

$$\cos\varphi = \frac{\langle q_s, q_e \rangle}{\|q_s\| \cdot \|q_e\|}. \tag{37}$$

      ii. Compute the interpolated rotation quaternion q_i as

$$q_i = q_s \left(\frac{\sin((1-s)\varphi)}{\sin\varphi} \right) + q_e \left(\frac{\sin(s\varphi)}{\sin\varphi} \right). \tag{38}$$

References

Main theory: [4] and [2, pp. 296–313]. Quaternion to Direction Cosine Matrix (DCM) conversion: [5, pp. 43 et seq.]. Quaternion interpolation algorithms: LERP [2, section 4.4.3.1, pp. 308 et seq.], SLERP algorithm by Bronstein [2, section 4.4.3.2, pp. 309 et seq.], SLERP algorithm by Dam, Koch and Lillholm [3, section 6.1.5, pp. 42 et seqq.].

[1] Bar-Itzhack, Itzhack Y.: *New Method for Extracting the Quaternion from a Rotation Matrix*. In: *Journal of Guidance, Control, and Dynamics* **23** (6): 1085–1087. American Institute of Aeronautics and Astronautics (AIAA), 2000. ISSN: 0731-5090. DOI: 10.2514/2.4654. [→ cited on page 7]

[2] Bronstein, I. N.; Semendjajew, K. A.; Musiol, G.; Mühlig, H.: *Taschenbuch der Mathematik*. 8., vollständig überarbeitete Auflage. Wissenschaftlicher Verlag Harri Deutsch, Frankfurt am Main, 2012. ISBN: 978-3-8171-2008-6. [→ cited on page 14]

[3] Dam, Erik B.; Koch, Martin; Lillholm, Martin: *Quaternions, Interpolation and Animation*. Department of Computer Science, University of Copenhagen, Denmark, July 17, 1998. Technical Report DIKU-TR-98-5. Online available at http://www.itu.dk/people/erikdam/DOWNLOAD/98-5.pdf. [→ cited on page 14]

[4] Kuipers, Jack B.: *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. 5th printing, and first paperback printing. Princeton University Press, Princeton, New Jersey, USA, 2002. ISBN: 978-0-691-10298-6. [→ cited on pages 1, 4, and 14]

[5] Titterton, David H.; Weston, John L.: *Strapdown Inertial Navigation Technology*. Second edition. The Institution of Electrical Engineers and The American Institute of Aeronautics and Astronautics, United Kingdom and USA, 2004. ISBN: 0-86341-358-7. [→ cited on page 14]

[9] This is not the rotation angle α of the rotation quaternion in $\mathbb{R}^3$: $\alpha = 2\varphi$!

A The MATLAB Quaternion Class

A.1 How To Create an Object of the Quaternion Class

Constructor Call	**s** double[1x1]	**i** double[1x1]	**j** double[1x1]	**k** double[1x1]	**notation** char[1x...]
0 Arguments					
Quaternion() Example: Quaternion()	NaN	NaN	NaN	NaN	'math'
1 Argument					
Quaternion(double[1x4] **arg1**) Example: Quaternion([s,i,j,k])	**arg1**(1,1)	**arg1**(1,2)	**arg1**(1,3)	**arg1**(1,4)	'math'
Quaternion(double[3x1] **arg1**) Example: Quaternion([i;j;k])	0	**arg1**(1,1)	**arg1**(2,1)	**arg1**(3,1)	'math'
2 Arguments					
Quaternion('notation', char[1x...] **arg2**) Example: Quaternion('notation', 'space')	NaN	NaN	NaN	NaN	**arg2**[10]
Quaternion(double[1x1] **arg1**, double[3x1] **arg2**) Example: Quaternion(s,[i;j;k])	**arg1**	**arg2**(1,1)	**arg2**(2,1)	**arg2**(3,1)	'math'
Quaternion(double[3x1] **arg1**, double[1x1] **arg2**) Example: Quaternion([i;j;k],s)	**arg2**	**arg1**(1,1)	**arg1**(2,1)	**arg1**(3,1)	'space'
3 Arguments					
Quaternion(double[1x1] **arg1**, double[1x1] **arg2**, double[1x1] **arg3**) Example: Quaternion(i,j,k)	0	**arg1**	**arg2**	**arg3**	'math'

[10] must either be 'math' or 'space'

Quaternion(double[1x4] **arg1** 'notation', 'math') Example: Quaternion([s,i,j,k],'notation','math')	**arg1**(1,1)	**arg1**(1,2)	**arg1**(1,3)	**arg1**(1,4)	'math'
Quaternion(double[1x4] **arg1** 'notation', 'space') Example: Quaternion([i,j,k,s],'notation','space')	**arg1**(1,4)	**arg1**(1,1)	**arg1**(1,2)	**arg1**(1,3)	'space'
Quaternion(double[3x1] **arg1** 'notation', char[1x...] **arg3**) Example: Quaternion([i;j;k],'notation','math')	0	**arg1**(1,1)	**arg1**(2,1)	**arg1**(3,1)	**arg3**[10]
4 Arguments					
Quaternion(double[1x1] **arg1**, double[1x1] **arg2**, double[1x1] **arg3**, double[1x1] **arg4**) Example: Quaternion(s,i,j,k)	**arg1**	**arg2**	**arg3**	**arg4**	'math'
Quaternion(double[1x1] **arg1**, double[3x1] **arg2**, 'notation', char[1x...] **arg4**) Example: Quaternion(s,[i;j;k],'notation','math')	**arg1**	**arg2**(1,1)	**arg2**(2,1)	**arg2**(3,1)	**arg4**[10]
Quaternion(double[3x1] **arg1**, double[1x1] **arg2**, 'notation', char[1x...] **arg4**) Example: Quaternion([i;j;k],s,'notation','math')	**arg2**	**arg1**(1,1)	**arg1**(2,1)	**arg1**(3,1)	**arg4**[10]
5 Arguments					
Quaternion(double[1x1] **arg1**, double[1x1] **arg2**, double[1x1] **arg3**, 'notation', char[1x...] **arg5**) Example: Quaternion(i,j,k,'notation','math')	0	**arg1**	**arg2**	**arg3**	**arg5**[10]

6 Arguments					

`Quaternion(`
`  double[1x1] arg1,`
`  double[1x1] arg2,`
`  double[1x1] arg3,`
`  double[1x1] arg4,`
`  'notation', 'math'`
`)`
Example:
`Quaternion(s,i,j,k,'notation','math')`
	`arg1`	`arg2`	`arg3`	`arg4`	`'math'`

`Quaternion(`
`  double[1x1] arg1,`
`  double[1x1] arg2,`
`  double[1x1] arg3,`
`  double[1x1] arg4,`
`  'notation', 'space'`
`)`
Example:
`Quaternion(i,j,k,s,'notation','space')`
	`arg4`	`arg1`	`arg2`	`arg3`	`'space'`

A.2 Warning Identifiers

Exception Identifier	Explanation
`Quaternion:AngleNotNormalized`	An given or calculated angle is not normalized to be in an expected range, e. g. $[0, 2\pi)$.
`Quaternion:IsNotUnit`	A given or calculated quaternion is not a unit quaternion; hence it does not represent a proper rotation/orientation in $\mathbb{R}^3$.
`Quaternion:NotationSequenceMismatch`	The given sequence of quaternion components does not match the notation type specified.
`Quaternion:NotationTypeMismatch`	During an operation of two or more objects of class `Quaternion` at least one involved `Quaternion` has a different notation type.
`Quaternion:OperationNotDefined`	The operation requested is not defined and hence returned an Inf, -Inf, or NaN. The natural logorarithm of the quaternion $q = [0, 0, 0, 0]$, for example, is undefined.
`Quaternion:VecNotUnitLength`	A given or calculated vector does not have unit length, i. e. its 2-norm is not $\|\mathbf{v}\| = 1$.

A.3 Exception Identifiers

Exception Identifier	Explanation
`Quaternion:InputOutOfRange`	One or more inputs are not within the expected range.
`Quaternion:NotImplemented`	The feature or method is not implemented yet. This exception may only be thrown for a certain input or parameter combination.
`Quaternion:SignatureMismatch`	This exception is thrown if the function call and the function signature do not match, i. e. the number of arguments or their types are incorrect.
`Quaternion:UnexpectedException`	An unexpected exception was raised. This is due to a programming error inside the Quaternion class. Please inform the author via e-mail to mail@rene-schwarz.com including the commands leading to this exception.