WIKIPEDIA
The Free Encyclopedia

Quaternions and spatial rotation

Unit quaternions, known as *versors*, provide a convenient mathematical notation for representing spatial orientations and rotations of elements in three dimensional space. Specifically, they encode information about an axis-angle rotation about an arbitrary axis. Rotation and orientation quaternions have applications in computer graphics,[1] computer vision, robotics,[2] navigation, molecular dynamics, flight dynamics,[3] orbital mechanics of satellites,[4] and crystallographic texture analysis.[5]

When used to represent rotation, unit quaternions are also called **rotation quaternions** as they represent the 3D rotation group. When used to represent an orientation (rotation relative to a reference coordinate system), they are called **orientation quaternions** or **attitude quaternions**. A spatial rotation around a fixed point of θ radians about a unit axis (X, Y, Z) that denotes the *Euler axis* is given by the quaternion $(C, X\,S, Y\,S, Z\,S)$, where $C = \cos(\theta/2)$ and $S = \sin(\theta/2)$.

Compared to rotation matrices, quaternions are more compact, efficient, and numerically stable. Compared to Euler angles, they are simpler to compose. However, they are not as intuitive and easy to understand and, due to the periodic nature of sine and cosine, rotation angles differing precisely by the natural period will be encoded into identical quaternions and recovered angles in radians will be limited to $[0, 2\pi]$.

Using quaternions as rotations

In 3-dimensional space, according to Euler's rotation theorem, any rotation or sequence of rotations of a rigid body or coordinate system about a fixed point is equivalent to a single rotation by a given angle θ about a fixed axis (called the *Euler axis*) that runs through the fixed point.[6] The Euler axis is typically represented by a unit vector $\vec{u}$ ($\hat{e}$ in the picture). Therefore, any rotation in three dimensions can be represented as a vector $\vec{u}$ and an angle θ.

Quaternions give a simple way to encode this axis–angle representation using four real numbers, and can be used to apply (calculate) the corresponding rotation to a position vector (x,y,z), representing a point relative to the origin in $\mathbf{R}^3$.[7]

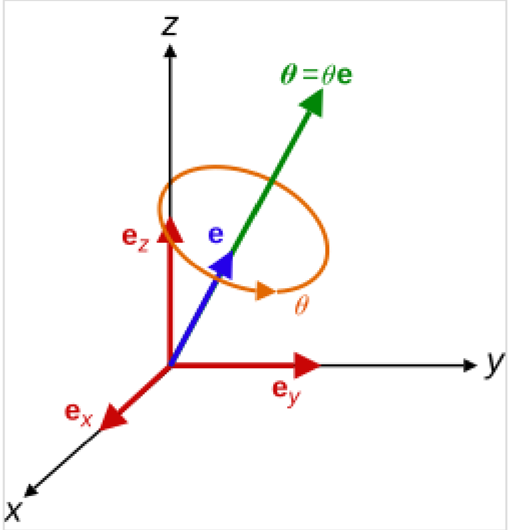
Euclidean vectors such as $(2, 3, 4)$ or (a_x, a_y, a_z) can be rewritten as $2\mathbf{i} + 3\mathbf{j} + 4\mathbf{k}$ or $a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}$, where $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are unit vectors representing the three Cartesian axes (traditionally $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$), and also obey the multiplication rules of the fundamental quaternion units by interpreting the Euclidean vector (a_x, a_y, a_z) as the vector part of the pure quaternion $(0, a_x, a_y, a_z)$.

A rotation of angle θ around the axis defined by the unit vector

$$\mathbf{u} = (u_x, u_y, u_z) = u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}$$

can be represented by conjugation by a unit quaternion $\mathbf{q}$. Since the quaternion product $(0 + u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})(0 - u_x\mathbf{i} - u_y\mathbf{j} - u_z\mathbf{k})$ gives 1, using the Taylor series of the exponential function, the extension of Euler's formula results:

$$\mathbf{q} = e^{\frac{\theta}{2}(u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})} = \cos\frac{\theta}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2} = \cos\frac{\theta}{2} + \mathbf{u}\sin\frac{\theta}{2}$$



3D visualization of a rotation about an Euler axis ($\hat{\boldsymbol{e}}$) by an angle of $\boldsymbol{\theta}$

It can be shown[8] that the desired rotation can be applied to an ordinary vector $\mathbf{p} = (p_x, p_y, p_z) = p_x\mathbf{i} + p_y\mathbf{j} + p_z\mathbf{k}$ in 3-dimensional space, considered as the vector part of the pure quaternion $\mathbf{p}'$, by evaluating the conjugation of $\mathbf{p}'$ by $\mathbf{q}$, given by:

$$L(\mathbf{p}') := \mathbf{q}\mathbf{p}'\mathbf{q}^{-1} = (0, \mathbf{r}),$$
$$\mathbf{r} = (\cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2}||\mathbf{u}||^2)\mathbf{p} + 2\sin^2\frac{\theta}{2}(\mathbf{u}\cdot\mathbf{p})\mathbf{u} + 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}(\mathbf{u}\times\mathbf{p}),$$

using the Hamilton product, where the vector part of the pure quaternion $\mathrm{L}(\mathbf{p}') = (0, r_x, r_y, r_z)$ is the new position vector of the point after the rotation. In a programmatic implementation, the conjugation is achieved by constructing a pure quaternion whose vector part is $\mathbf{p}$, and then performing the quaternion conjugation. The vector part of the resulting pure quaternion is the desired vector $\mathbf{r}$. Clearly, L provides a linear transformation of the quaternion space to itself;[9] also, since $\mathbf{q}$ is unitary, the transformation is an isometry. Also, $L(\mathbf{q}) = \mathbf{q}$ and so L leaves vectors parallel to $\mathbf{q}$ invariant. So, by decomposing $\mathbf{p}$ as a vector parallel to the vector part $(u_x, u_y, u_z)\sin\frac{\theta}{2}$ of $\mathbf{q}$ and a vector normal to the vector part of $\mathbf{q}$ and showing that the application of L to the normal component of $\mathbf{p}$ rotates it, the claim is shown. So let $\mathbf{n}$ be the component of $\mathbf{p}$ orthogonal to the vector part of $\mathbf{q}$ and let $\mathbf{n}_T = \mathbf{n} \times \mathbf{u}$. It turns out that the vector part of $L(0, \mathbf{n})$ is given by

$$\left(\cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2}\right)\mathbf{n} + 2\left(\cos\frac{\theta}{2}\sin\frac{\theta}{2}\right)\mathbf{n}_T = \cos\theta\mathbf{n} + \sin\theta\mathbf{n}_T.$$

The conjugation of $\mathbf{p}'$ by $\mathbf{q}$ can be expressed with fewer arithmetic operations as:

$$\mathbf{r} = \mathbf{p} + 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}(\mathbf{u}\times\mathbf{p}) + 2\sin^2\frac{\theta}{2}\mathbf{u}\times(\mathbf{u}\times\mathbf{p}).$$

A geometric fact independent of quaternions is the existence of a two-to-one mapping from physical rotations to rotational transformation matrices. If 0 ⩽ θ ⩽ 2π, a physical rotation about $\vec{u}$ by θ and a physical rotation about $-\vec{u}$ by $2\pi - \theta$ both achieve the same final orientation by disjoint paths through intermediate orientations. By inserting those vectors and angles into the formula for $\mathbf{q}$ above, one finds that if $\mathbf{q}$ represents the first rotation, $-\mathbf{q}$ represents the second rotation. This is a geometric proof that conjugation by $\mathbf{q}$ and by $-\mathbf{q}$ must produce the same rotational transformation matrix. That fact is confirmed algebraically by noting that the conjugation is quadratic in $\mathbf{q}$, so the sign of $\mathbf{q}$ cancels, and does not affect the result. (See 2:1 mapping of SU(2) to SO(3)) If both rotations are a half-turn $(\theta = \pi)$, both $\mathbf{q}$ and $-\mathbf{q}$ will have a real coordinate equal to zero. Otherwise, one will have a positive real part, representing a rotation by an angle less than π, and the other will have a negative real part, representing a rotation by an angle greater than π.

Mathematically, this operation carries the *set* of all "pure" quaternions $\mathbf{p}$ (those with real part equal to zero)—which constitute a 3-dimensional space among the quaternions—into itself, by the desired rotation about the axis $\boldsymbol{u}$, by the angle θ. (Each real quaternion is carried into itself by this operation. But for the purpose of rotations in 3-dimensional space, we ignore the real quaternions.)

The rotation is clockwise if our line of sight points in the same direction as $\vec{u}$.

In this instance, $\mathbf{q}$ is a unit quaternion and

$$\mathbf{q}^{-1} = e^{-\frac{\theta}{2}(u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})} = \cos\frac{\theta}{2} - (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2}.$$

It follows that conjugation by the product of two quaternions is the composition of conjugations by these quaternions: If $\mathbf{p}$ and $\mathbf{q}$ are unit quaternions, then rotation (conjugation) by $\mathbf{pq}$ is

$$\mathbf{pq}\vec{v}(\mathbf{pq})^{-1} = \mathbf{pq}\vec{v}\mathbf{q}^{-1}\mathbf{p}^{-1} = \mathbf{p}(\mathbf{q}\vec{v}\mathbf{q}^{-1})\mathbf{p}^{-1},$$

which is the same as rotating (conjugating) by $\mathbf{q}$ and then by $\mathbf{p}$. The scalar component of the result is necessarily zero.

The quaternion inverse of a rotation is the opposite rotation, since $\mathbf{q}^{-1}(\mathbf{q}\vec{v}\mathbf{q}^{-1})\mathbf{q} = \vec{v}$. The square of a quaternion rotation is a rotation by twice the angle around the same axis. More generally $\mathbf{q}^n$ is a rotation by n times the angle around the same axis as $\mathbf{q}$. This can be extended to arbitrary real n, allowing for smooth interpolation between spatial orientations; see Slerp.

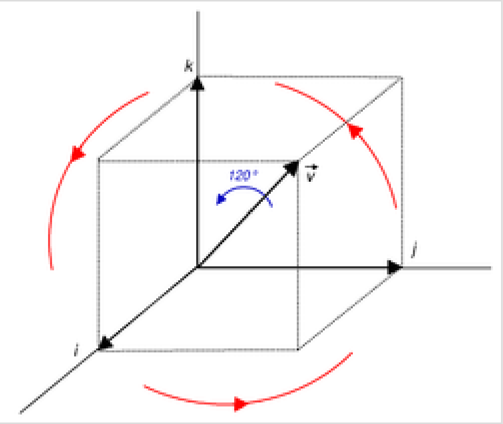
Two rotation quaternions can be combined into one equivalent quaternion by the relation:

$$\mathbf{q}' = \mathbf{q}_2\mathbf{q}_1$$

in which $\mathbf{q}'$ corresponds to the rotation $\mathbf{q}_1$ followed by the rotation $\mathbf{q}_2$. Thus, an arbitrary number of rotations can be composed together and then applied as a single rotation. (Note that quaternion multiplication is not commutative.)

Example conjugation operation

Conjugating $\mathbf{p}$ *by* $\mathbf{q}$ refers to the operation $\mathbf{p} \mapsto \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$.



A rotation of 120° around the *first diagonal* permutes $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ cyclically

Consider the rotation f around the axis $\vec{v} = \mathbf{i} + \mathbf{j} + \mathbf{k}$, with a rotation angle of 120°, or $\frac{2\pi}{3}$ radians.

$$\alpha = \frac{2\pi}{3}$$

The length of $\vec{v}$ is $\sqrt{3}$, the half angle is $\frac{\pi}{3}$ (60°) with cosine $\frac{1}{2}$, $(\cos\ 60° = 0.5)$ and sine $\frac{\sqrt{3}}{2}$, $(\sin\ 60° \approx 0.866)$. We are therefore dealing with a conjugation by the unit quaternion

$$\begin{aligned} u &= \cos\frac{\alpha}{2} + \sin\frac{\alpha}{2} \cdot \frac{1}{\|\vec{v}\|}\vec{v} \\ &= \cos\frac{\pi}{3} + \sin\frac{\pi}{3} \cdot \frac{1}{\sqrt{3}}\vec{v} \\ &= \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot \frac{1}{\sqrt{3}}\vec{v} \\ &= \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot \frac{\mathbf{i}+\mathbf{j}+\mathbf{k}}{\sqrt{3}} \\ &= \frac{1+\mathbf{i}+\mathbf{j}+\mathbf{k}}{2} \end{aligned}$$

If f is the rotation function,

$$f(a\mathbf{i} + b\mathbf{j} + c\mathbf{k}) = u(a\mathbf{i} + b\mathbf{j} + c\mathbf{k})u^{-1}$$

It can be proven that the inverse of a unit quaternion is obtained simply by changing the sign of its imaginary components. As a consequence,

$$u^{-1} = \frac{1-\mathbf{i}-\mathbf{j}-\mathbf{k}}{2}$$

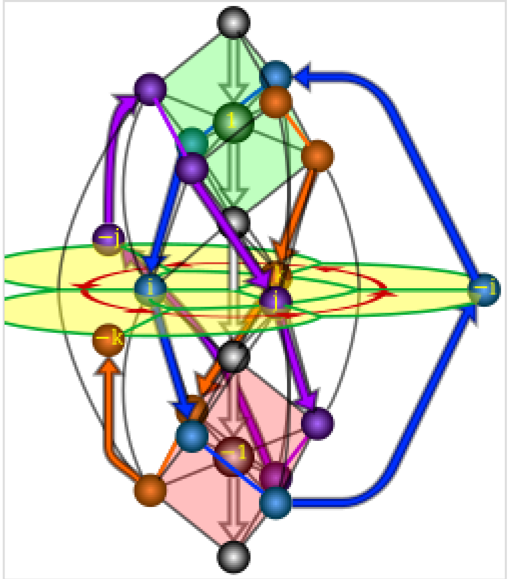
and

$$f(a\mathbf{i}+b\mathbf{j}+c\mathbf{k}) = \frac{1+\mathbf{i}+\mathbf{j}+\mathbf{k}}{2}(a\mathbf{i}+b\mathbf{j}+c\mathbf{k})\frac{1-\mathbf{i}-\mathbf{j}-\mathbf{k}}{2}$$

This can be simplified, using the ordinary rules for quaternion arithmetic, to

$$f(a\mathbf{i}+b\mathbf{j}+c\mathbf{k}) = c\mathbf{i}+a\mathbf{j}+b\mathbf{k}$$

As expected, the rotation corresponds to keeping a cube held fixed at one point, and rotating it 120° about the long diagonal through the fixed point (observe how the three axes are permuted cyclically).



$\mathbf{p} \mapsto \mathbf{q}\,\mathbf{p}$ for $\mathbf{q} = \frac{1+i+j+k}{2}$ on the unit 3-sphere. Note this one-sided (namely, left) multiplication yields a 60° rotation of quaternions

Quaternion-derived rotation matrix

A quaternion rotation $\mathbf{p'} = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$ (with $\mathbf{q} = q_r + q_i\mathbf{i} + q_j\mathbf{j} + q_k\mathbf{k}$) can be algebraically manipulated into a matrix rotation $\mathbf{p'} = \mathbf{R}\mathbf{p}$, where $\mathbf{R}$ is the rotation matrix given by:[10]

$$\mathbf{R} = \begin{bmatrix} 1-2s(q_j^2+q_k^2) & 2s(q_iq_j-q_kq_r) & 2s(q_iq_k+q_jq_r) \\ 2s(q_iq_j+q_kq_r) & 1-2s(q_i^2+q_k^2) & 2s(q_jq_k-q_iq_r) \\ 2s(q_iq_k-q_jq_r) & 2s(q_jq_k+q_iq_r) & 1-2s(q_i^2+q_j^2) \end{bmatrix}$$

Here $s = \|q\|^{-2}$ and if q is a unit quaternion, $s = 1^{-2} = 1$.

This can be obtained by using vector calculus and linear algebra if we express $\mathbf{p}$ and $\mathbf{q}$ as scalar and vector parts and use the formula for the multiplication operation in the equation $\mathbf{p'} = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$. If we write $\mathbf{p}$ as $(0,\ \mathbf{p})$, $\mathbf{p'}$ as $(0,\ \mathbf{p'})$ and $\mathbf{q}$ as $(q_r,\ \mathbf{v})$, where $\mathbf{v} = (q_i, q_j, q_k)$, our equation turns into $(0,\ \mathbf{p'}) = (q_r,\ \mathbf{v})\,(0,\ \mathbf{p})\,s\,(q_r,\ -\mathbf{v})$. By using the formula for multiplication of two quaternions that are expressed as scalar and vector parts,

$$(r_1,\ \vec{v}_1)\,(r_2,\ \vec{v}_2) = (r_1r_2 - \vec{v}_1\cdot\vec{v}_2,\ r_1\vec{v}_2 + r_2\vec{v}_1 + \vec{v}_1\times\vec{v}_2)\,,$$

this equation can be rewritten as

$$\begin{aligned}(0,\ \mathbf{p}') &= ((q_r,\ \mathbf{v})(0,\ \mathbf{p}))s(q_r,\ -\mathbf{v})\\ &= (q_r 0 - \mathbf{v}\cdot\mathbf{p},\ q_r\mathbf{p} + 0\mathbf{v} + \mathbf{v}\times\mathbf{p})s(q_r,\ -\mathbf{v})\\ &= s(-\mathbf{v}\cdot\mathbf{p},\ q_r\mathbf{p} + \mathbf{v}\times\mathbf{p})(q_r,\ -\mathbf{v})\\ &= s(-\mathbf{v}\cdot\mathbf{p}q_r - (q_r\mathbf{p} + \mathbf{v}\times\mathbf{p})\cdot(-\mathbf{v}),\ (-\mathbf{v}\cdot\mathbf{p})(-\mathbf{v}) + q_r(q_r\mathbf{p} + \mathbf{v}\times\mathbf{p}) + (q_r\mathbf{p} + \mathbf{v}\times\mathbf{p})\times(-\mathbf{v}))\\ &= s\left(-\mathbf{v}\cdot\mathbf{p}q_r + q_r\mathbf{v}\cdot\mathbf{p},\ \mathbf{v}\left(\mathbf{v}\cdot\mathbf{p}\right) + q_r^2\mathbf{p} + q_r\mathbf{v}\times\mathbf{p} + \mathbf{v}\times\left(q_r\mathbf{p} + \mathbf{v}\times\mathbf{p}\right)\right)\\ &= \left(0,\ s\left(\mathbf{v}\otimes\mathbf{v} + q_r^2\mathbf{I} + 2q_r[\mathbf{v}]_\times + [\mathbf{v}]_\times^2\right)\mathbf{p}\right),\end{aligned}$$

where $\otimes$ denotes the outer product, $\mathbf{I}$ is the identity matrix and $[\mathbf{v}]_\times$ is the transformation matrix that when multiplied from the right with a vector $\mathbf{u}$ gives the cross product $\mathbf{v}\times\mathbf{u}$.

Since $\mathbf{p}' = \mathbf{R}\mathbf{p}$, we can identify $\mathbf{R}$ as $s\left(\mathbf{v}\otimes\mathbf{v} + q_r^2\mathbf{I} + 2q_r[\mathbf{v}]_\times + [\mathbf{v}]_\times^2\right)$, which upon expansion should result in the expression written in matrix form above.

Recovering the axis-angle representation

The expression $\mathbf{q}\mathbf{p}\mathbf{q}^{-1}$ rotates any vector quaternion $\mathbf{p}$ around an axis given by the vector $\mathbf{a}$ by the angle θ, where $\mathbf{a}$ and θ depends on the quaternion $\mathbf{q} = q_r + q_i\mathbf{i} + q_j\mathbf{j} + q_k\mathbf{k}$.

$\mathbf{a}$ and θ can be found from the following equations:

$$\begin{aligned}(a_x, a_y, a_z) &= \frac{(q_i, q_j, q_k)}{\sqrt{q_i^2 + q_j^2 + q_k^2}}\\ \theta &= 2\,\operatorname{atan2}\left(\sqrt{q_i^2 + q_j^2 + q_k^2},\ q_r\right),\end{aligned}$$

where $\operatorname{atan2}$ is the two-argument arctangent. While $\theta = 2\operatorname{acos}(q_r)$ works, it is numerically unstable (inaccurate) near $q_r = \pm 1$ for numbers with finite precision.

Care should be taken when the quaternion approaches a scalar, since due to degeneracy the axis of an identity rotation is not well-defined.

The composition of spatial rotations

A benefit of the quaternion formulation of the composition of two rotations R_B and R_A is that it yields directly the rotation axis and angle of the composite rotation $R_C = R_B R_A$.

Let the quaternion associated with a spatial rotation R be constructed from its rotation axis $\mathbf{S}$ with the rotation angle φ around this axis. The associated quaternion is given by

$$S = \cos\frac{\varphi}{2} + \mathbf{S}\sin\frac{\varphi}{2}.$$

Then the composition of the rotation R_B with R_A is the rotation $R_C = R_B R_A$ with rotation axis and angle defined by the product of the quaternions

$$A = \cos\frac{\alpha}{2} + \mathbf{A}\sin\frac{\alpha}{2} \quad \text{and} \quad B = \cos\frac{\beta}{2} + \mathbf{B}\sin\frac{\beta}{2},$$

that is

$$C = \cos\frac{\gamma}{2} + \mathbf{C}\sin\frac{\gamma}{2} = \left(\cos\frac{\beta}{2} + \mathbf{B}\sin\frac{\beta}{2}\right)\left(\cos\frac{\alpha}{2} + \mathbf{A}\sin\frac{\alpha}{2}\right).$$

Expand this product to obtain

$$\cos\frac{\gamma}{2} + \mathbf{C}\sin\frac{\gamma}{2} = \left(\cos\frac{\beta}{2}\cos\frac{\alpha}{2} - \mathbf{B}\cdot\mathbf{A}\sin\frac{\beta}{2}\sin\frac{\alpha}{2}\right) + \left(\mathbf{B}\sin\frac{\beta}{2}\cos\frac{\alpha}{2} + \mathbf{A}\sin\frac{\alpha}{2}\cos\frac{\beta}{2} + \mathbf{B}\times\mathbf{A}\sin\frac{\beta}{2}\sin\frac{\alpha}{2}\right).$$

Divide both sides of this equation by the identity, which is the law of cosines on a sphere,

$$\cos\frac{\gamma}{2} = \cos\frac{\beta}{2}\cos\frac{\alpha}{2} - \mathbf{B}\cdot\mathbf{A}\sin\frac{\beta}{2}\sin\frac{\alpha}{2},$$

and compute

$$\mathbf{C}\tan\frac{\gamma}{2} = \frac{\mathbf{B}\tan\frac{\beta}{2} + \mathbf{A}\tan\frac{\alpha}{2} + \mathbf{B}\times\mathbf{A}\tan\frac{\beta}{2}\tan\frac{\alpha}{2}}{1 - \mathbf{B}\cdot\mathbf{A}\tan\frac{\beta}{2}\tan\frac{\alpha}{2}}.$$

This is Rodrigues' formula for the axis of a composite rotation defined in terms of the axes of the two rotations. He derived this formula in 1840 (see page 408).[11]

The three rotation axes $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ form a spherical triangle and the dihedral angles between the planes formed by the sides of this triangle are defined by the rotation angles. Hamilton[12] presented the component form of these equations showing that the quaternion product computes the third vertex of a spherical triangle from two given vertices and their associated arc-lengths, which is also defines an algebra for points in Elliptic geometry.

Axis–angle composition

The normalized rotation axis, removing the $\cos \frac{\gamma}{2}$ from the expanded product, leaves the vector which is the rotation axis, times some constant. Care should be taken normalizing the axis vector when γ is 0 or $k2\pi$ where the vector is near 0; which is identity, or 0 rotation around any axis.

$$\begin{aligned}\gamma &= 2\cos^{-1}\left(\cos\frac{\beta}{2}\cos\frac{\alpha}{2} - \mathbf{B}\cdot\mathbf{A}\sin\frac{\beta}{2}\sin\frac{\alpha}{2}\right)\\ \mathbf{D} &= \mathbf{B}\sin\frac{\beta}{2}\cos\frac{\alpha}{2} + \mathbf{A}\sin\frac{\alpha}{2}\cos\frac{\beta}{2} + \mathbf{B}\times\mathbf{A}\sin\frac{\beta}{2}\sin\frac{\alpha}{2}\end{aligned}$$

Or with angle addition trigonometric substitutions...

$$\begin{aligned}\gamma &= 2\cos^{-1}\left((1-\mathbf{A}\cdot\mathbf{B})\cos\frac{\beta-\alpha}{2} + (1+\mathbf{A}\cdot\mathbf{B})\cos\frac{\beta+\alpha}{2}\right)\\ \mathbf{D} &= \left(\sin\frac{\beta+\alpha}{2} + \sin\frac{\beta-\alpha}{2}\right)\mathbf{A} + \left(\sin\frac{\beta+\alpha}{2} - \sin\frac{\beta-\alpha}{2}\right)\mathbf{B} + \left(\cos\frac{\beta-\alpha}{2} - \cos\frac{\beta+\alpha}{2}\right)\mathbf{B}\times\mathbf{A}\end{aligned}$$

finally normalizing the rotation axis: $\frac{\mathbf{D}}{2\sin\frac{1}{2}\gamma}$ or $\frac{\mathbf{D}}{\|\mathbf{D}\|}$.

Differentiation with respect to the rotation quaternion

The rotated quaternion $\mathbf{p' = q\,p\,q}^{-1}$ needs to be differentiated with respect to the rotating quaternion $\mathbf{q}$, when the rotation is estimated from numerical optimization. The estimation of rotation angle is an essential procedure in 3D object registration or camera calibration. For unitary $\mathbf{q}$ and pure imaginary $\mathbf{p}$, that is for a rotation in 3D space, the derivatives of the rotated quaternion can be represented using matrix calculus notation as

$$\frac{\partial\mathbf{p}'}{\partial\mathbf{q}} \equiv \left[\frac{\partial\mathbf{p}'}{\partial q_0}, \frac{\partial\mathbf{p}'}{\partial q_x}, \frac{\partial\mathbf{p}'}{\partial q_y}, \frac{\partial\mathbf{p}'}{\partial q_z}\right] = \left[\mathbf{qp} - (\mathbf{qp})^*, (\mathbf{qpi})^* - \mathbf{qpi}, (\mathbf{qpj})^* - \mathbf{qpj}, (\mathbf{qpk})^* - \mathbf{qpk}\right].$$

A derivation can be found in.[13]

Background

Quaternions

The complex numbers can be defined by introducing an abstract symbol $\mathbf{i}$ which satisfies the usual rules of algebra and additionally the rule $\mathbf{i}^2 = -1$. This is sufficient to reproduce all of the rules of complex number arithmetic: for example:

$$(a+b\mathbf{i})(c+d\mathbf{i}) = ac + ad\mathbf{i} + b\mathbf{i}c + b\mathbf{i}d\mathbf{i} = ac + ad\mathbf{i} + bc\mathbf{i} + bd\mathbf{i}^2 = (ac-bd) + (bc+ad)\mathbf{i}.$$

In the same way the quaternions can be defined by introducing abstract symbols $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ which satisfy the rules $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i\,j\,k} = -1$ and the usual algebraic rules *except* the commutative law of multiplication (a familiar example of such a noncommutative multiplication is matrix multiplication). From this all of the rules of quaternion arithmetic follow, such as the rules on multiplication of quaternion basis elements. Using these rules, one can show that:

$$\begin{aligned}&(a+b\mathbf{i}+c\mathbf{j}+d\mathbf{k})(e+f\mathbf{i}+g\mathbf{j}+h\mathbf{k}) =\\&(ae-bf-cg-dh)+(af+be+ch-dg)\mathbf{i}+(ag-bh+ce+df)\mathbf{j}+(ah+bg-cf+de)\mathbf{k}.\end{aligned}$$

The imaginary part $b\mathbf{i}+c\mathbf{j}+d\mathbf{k}$ of a quaternion behaves like a vector $\vec{v} = (b,c,d)$ in three-dimensional vector space, and the real part a behaves like a scalar in $\mathbf{R}$. When quaternions are used in geometry, it is more convenient to define them as a scalar plus a vector:

$$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} = a + \vec{v}.$$

Some might find it strange to add a *number* to a *vector*, as they are objects of very different natures, or to *multiply* two vectors together, as this operation is usually undefined. However, if one remembers that it is a mere notation for the real and imaginary parts of a quaternion, it becomes more legitimate. In other words, the correct reasoning is the addition of two quaternions, one with zero vector/imaginary part, and another one with zero scalar/real part:

$$q_1 = s + \vec{v} = \left(s, \vec{0}\right) + (0, \vec{v})\,.$$

We can express quaternion multiplication in the modern language of vector cross and dot products (which were actually inspired by the quaternions in the first place[14]). When multiplying the vector/imaginary parts, in place of the rules $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$ we have the quaternion multiplication rule:

$$\vec{v}\vec{w} = -\vec{v}\cdot\vec{w} + \vec{v}\times\vec{w},$$

where:

- $\vec{v}\vec{w}$ is the resulting quaternion,
- $\vec{v} \times \vec{w}$ is vector cross product (a vector),
- $\vec{v} \cdot \vec{w}$ is vector scalar product (a scalar).

Quaternion multiplication is noncommutative (because of the cross product, which anti-commutes), while scalar–scalar and scalar–vector multiplications commute. From these rules it follows immediately that (see Quaternions § Quaternions and three-dimensional geometry):

$$q_1 q_2 = (s + \vec{v})\,(t + \vec{w}) = (st - \vec{v} \cdot \vec{w}) + (s\vec{w} + t\vec{v} + \vec{v} \times \vec{w})\,.$$

The (left and right) multiplicative inverse or reciprocal of a nonzero quaternion is given by the conjugate-to-norm ratio (see details):

$$q_1^{-1} = (s + \vec{v})^{-1} = \frac{(s + \vec{v})^*}{\|s + \vec{v}\|^2} = \frac{s - \vec{v}}{s^2 + \|\vec{v}\|^2}\,,$$

as can be verified by direct calculation (note the similarity to the multiplicative inverse of complex numbers).

Rotation identity

Let $\vec{u}$ be a unit vector (the rotation axis) and let $q = \cos\frac{\alpha}{2} + \vec{u}\sin\frac{\alpha}{2}$. Our goal is to show that

$$\vec{v}' = q\vec{v}q^{-1} = \left(\cos\frac{\alpha}{2} + \vec{u}\sin\frac{\alpha}{2}\right)\,\vec{v}\,\left(\cos\frac{\alpha}{2} - \vec{u}\sin\frac{\alpha}{2}\right)$$

yields the vector $\vec{v}$ rotated by an angle α around the axis $\vec{u}$. Expanding out (and bearing in mind that $\vec{u}\vec{v} = \vec{u} \times \vec{v} - \vec{u} \cdot \vec{v}$), we have

$$
\begin{aligned}
\vec{v}' &= \vec{v}\cos^2\frac{\alpha}{2} + (\vec{u}\vec{v} - \vec{v}\vec{u})\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} - \vec{u}\vec{v}\vec{u}\sin^2\frac{\alpha}{2} \\
&= \vec{v}\cos^2\frac{\alpha}{2} + 2\left(\vec{u}\times\vec{v}\right)\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} - \left(\left(\vec{u}\times\vec{v}\right) - \left(\vec{u}\cdot\vec{v}\right)\right)\vec{u}\sin^2\frac{\alpha}{2} \\
&= \vec{v}\cos^2\frac{\alpha}{2} + 2\left(\vec{u}\times\vec{v}\right)\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} - \left(\left(\vec{u}\times\vec{v}\right)\vec{u} - \left(\vec{u}\cdot\vec{v}\right)\vec{u}\right)\sin^2\frac{\alpha}{2} \\
&= \vec{v}\cos^2\frac{\alpha}{2} + 2\left(\vec{u}\times\vec{v}\right)\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} - \left(\left(\left(\vec{u}\times\vec{v}\right)\times\vec{u} - \left(\vec{u}\times\vec{v}\right)\cdot\vec{u}\right) - \left(\vec{u}\cdot\vec{v}\right)\vec{u}\right)\sin^2\frac{\alpha}{2} \\
&= \vec{v}\cos^2\frac{\alpha}{2} + 2\left(\vec{u}\times\vec{v}\right)\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} - \left(\left(\vec{v} - \left(\vec{u}\cdot\vec{v}\right)\vec{u}\right) - 0 - \left(\vec{u}\cdot\vec{v}\right)\vec{u}\right)\sin^2\frac{\alpha}{2} \\
&= \vec{v}\cos^2\frac{\alpha}{2} + 2\left(\vec{u}\times\vec{v}\right)\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} - \left(\vec{v} - 2\vec{u}\left(\vec{u}\cdot\vec{v}\right)\right)\sin^2\frac{\alpha}{2} \\
&= \vec{v}\cos^2\frac{\alpha}{2} + 2\left(\vec{u}\times\vec{v}\right)\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} + \left(2\vec{u}\left(\vec{u}\cdot\vec{v}\right) - \vec{v}\right)\sin^2\frac{\alpha}{2}
\end{aligned}
$$

If we let $\vec{v}_{\perp}$ and $\vec{v}_{\|}$ equal the components of $\vec{v}$ perpendicular and parallel to $\vec{u}$ respectively, then $\vec{v} = \vec{v}_{\perp} + \vec{v}_{\|}$ and $\vec{u}\left(\vec{u}\cdot\vec{v}\right) = \vec{v}_{\|}$, leading to

$$
\begin{aligned}
\vec{v}' &= \vec{v}\cos^2\frac{\alpha}{2} + 2\left(\vec{u}\times\vec{v}\right)\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} + \left(2\vec{u}\left(\vec{u}\cdot\vec{v}\right) - \vec{v}\right)\sin^2\frac{\alpha}{2} \\
&= \left(\vec{v}_{\|} + \vec{v}_{\perp}\right)\cos^2\frac{\alpha}{2} + 2\left(\vec{u}\times\vec{v}\right)\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} + \left(\vec{v}_{\|} - \vec{v}_{\perp}\right)\sin^2\frac{\alpha}{2} \\
&= \vec{v}_{\|}\left(\cos^2\frac{\alpha}{2} + \sin^2\frac{\alpha}{2}\right) + \left(\vec{u}\times\vec{v}\right)\left(2\sin\frac{\alpha}{2}\cos\frac{\alpha}{2}\right) + \vec{v}_{\perp}\left(\cos^2\frac{\alpha}{2} - \sin^2\frac{\alpha}{2}\right)
\end{aligned}
$$

Using the trigonometric pythagorean and double-angle identities, we then have

$$
\begin{aligned}
\vec{v}' &= \vec{v}_{\|}\left(\cos^2\frac{\alpha}{2} + \sin^2\frac{\alpha}{2}\right) + \left(\vec{u}\times\vec{v}\right)\left(2\sin\frac{\alpha}{2}\cos\frac{\alpha}{2}\right) + \vec{v}_{\perp}\left(\cos^2\frac{\alpha}{2} - \sin^2\frac{\alpha}{2}\right) \\
&= \vec{v}_{\|} + \left(\vec{u}\times\vec{v}\right)\sin\alpha + \vec{v}_{\perp}\cos\alpha
\end{aligned}
$$

This is the formula of a rotation by α around the $\vec{u}$ axis.

Quaternion rotation operations

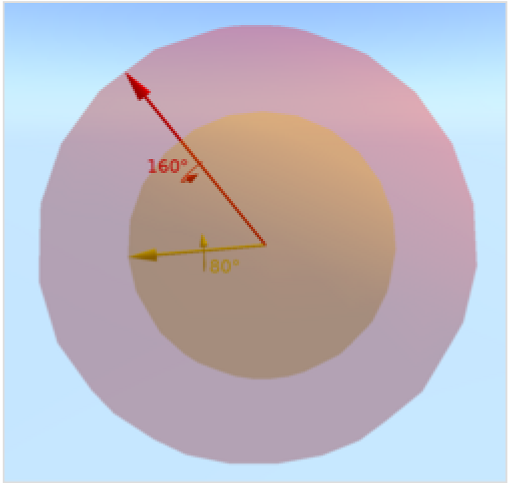
A very formal explanation of the properties used in this section is given by Altman.[15]

The hypersphere of rotations

Visualizing the space of rotations

Unit quaternions represent the group of Euclidean rotations in three dimensions in a very straightforward way. The correspondence between rotations and quaternions can be understood by first visualizing the space of rotations itself.

In order to visualize the space of rotations, it helps to consider a simpler case. Any rotation in three dimensions can be described by a rotation by some angle about some axis; for our purposes, we will use an axis *vector* to establish handedness for our angle. Consider the special case in which the axis of rotation lies in the *xy* plane. We can then specify the axis of one of these rotations by a point on a circle through which the vector crosses, and we can select the radius of the circle to denote the angle of rotation.

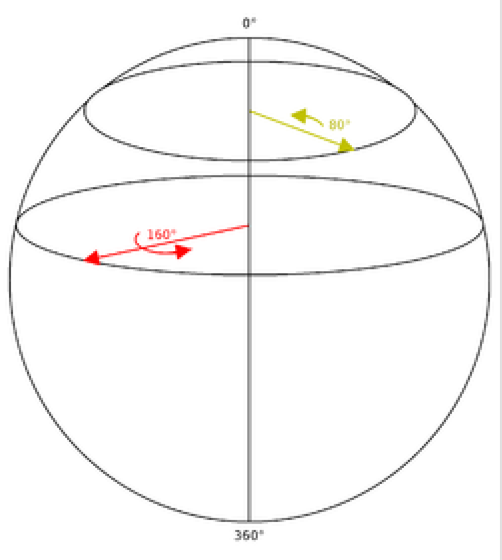


Two separate rotations, differing by both angle and axis, in the space of rotations. Here, the length of each axis vector is relative to the respective magnitude of the rotation about that axis.

Similarly, a rotation whose axis of rotation lies in the *xy* plane can be described as a point on a sphere of *fixed* radius in *three* dimensions. Beginning at the north pole of a sphere in three-dimensional space, we specify the point at the north pole to be the identity rotation (a zero angle rotation). Just as in the case of the identity rotation, no axis of rotation is defined, and the angle of rotation (zero) is irrelevant. A rotation having a very small rotation angle can be specified by a slice through the sphere parallel to the *xy* plane and very near the north pole. The circle defined by this slice will be very small, corresponding to the small angle of the rotation. As the rotation angles become larger, the slice moves in the negative *z* direction, and the circles become larger until the equator of the sphere is reached, which will correspond to a rotation angle of 180 degrees. Continuing southward, the radii of the circles now become smaller (corresponding to the absolute value of the angle of the rotation considered as a negative number). Finally, as the south pole is reached, the circles shrink once more to the identity rotation, which is also specified as the point at the south pole.

Notice that a number of characteristics of such rotations and their representations can be seen by this visualization. The space of rotations is continuous, each rotation has a neighborhood of rotations which are nearly the same, and this neighborhood becomes flat as the neighborhood shrinks. Also, each rotation is actually represented by two antipodal points on the sphere, which are at opposite ends of a line through the center of the sphere. This reflects the fact that each rotation can be represented as a rotation about some axis, or, equivalently, as a negative rotation about an axis pointing in the opposite direction (a so-called double cover). The "latitude" of a circle representing a particular rotation angle will be half of the angle represented by that rotation, since as the point is moved from the north to south pole, the latitude ranges from zero to 180 degrees, while the angle of rotation ranges from 0 to 360 degrees. (the "longitude" of a point then represents a particular axis of rotation.) Note however that this set of rotations is not closed under composition. Two successive rotations with axes in the *xy* plane will not necessarily give a rotation whose axis lies in the *xy* plane, and thus cannot be represented as a point on the sphere. This will not be the case with a general rotation in 3-space, in which rotations do form a closed set under composition.

This visualization can be extended to a general rotation in 3-dimensional space. The identity rotation is a point, and a small angle of rotation about some axis can be represented as a point on a sphere with a small radius. As the angle of rotation grows, the sphere grows, until the angle of rotation reaches 180 degrees, at which point the sphere begins to shrink, becoming a point as the angle approaches 360 degrees (or zero degrees from the negative direction). This set of expanding and contracting spheres represents a hypersphere in four dimensional space (a 3-sphere). Just as in the simpler example above, each rotation represented as a point on the hypersphere is matched by its antipodal point on that hypersphere. The "latitude" on the hypersphere will be half of the corresponding angle of rotation, and the neighborhood of any point will become "flatter" (i.e. be represented by a 3-D Euclidean space of points) as the neighborhood shrinks. This behavior is matched by the set of unit quaternions: A general quaternion represents a point in a four dimensional space, but constraining it to have unit magnitude yields a three-dimensional space equivalent to the surface of a hypersphere. The magnitude of the unit quaternion will be unity, corresponding to a hypersphere of unit radius. The vector part of a unit quaternion represents the radius of the 2-sphere corresponding to the axis of rotation, and its magnitude is the sine of half the angle of rotation. Each rotation is represented by two unit quaternions of opposite sign, and, as in the space of rotations in three dimensions, the quaternion product of two unit quaternions will yield a unit quaternion. Also, the space of unit quaternions is "flat" in any infinitesimal neighborhood of a given unit quaternion.



The sphere of rotations for the rotations that have a "horizontal" axis (in the *xy* plane).

Parameterizing the space of rotations

We can parameterize the surface of a sphere with two coordinates, such as latitude and longitude. But latitude and longitude are ill-behaved (degenerate as described by the hairy ball theorem) at the north and south poles, though the poles are not intrinsically different from any other points on the sphere. At the poles (latitudes +90° and −90°), the longitude becomes meaningless.

It can be shown that no two-parameter coordinate system can avoid such degeneracy. We can avoid such problems by embedding the sphere in three-dimensional space and parameterizing it with three Cartesian coordinates (w, x, y), placing the north pole at $(w, x, y) = (1, 0, 0)$, the south pole at $(w, x, y) = (-1, 0, 0)$, and the equator at $w = 0$, $x^2 + y^2 = 1$. Points on the sphere satisfy the constraint $w^2 + x^2 + y^2 = 1$, so we still have just two degrees of freedom though there are three coordinates. A point (w, x, y) on the sphere represents a rotation in the ordinary space around the horizontal axis directed by the vector $(x, y, 0)$ by an angle $\alpha = 2\cos^{-1} w = 2\sin^{-1} \sqrt{x^2 + y^2}$.

In the same way the hyperspherical space of 3D rotations can be parameterized by three angles (Euler angles), but any such parameterization is degenerate at some points on the hypersphere, leading to the problem of gimbal lock. We can avoid this by using four Euclidean coordinates w, x, y, z, with $w^2 + x^2 + y^2 + z^2 = 1$. The point (w, x, y, z) represents a rotation around the axis directed by the vector (x, y, z) by an angle $\alpha = 2\cos^{-1} w = 2\sin^{-1} \sqrt{x^2 + y^2 + z^2}$.

Explaining quaternions' properties with rotations

Non-commutativity

The multiplication of quaternions is non-commutative. This fact explains how the $\mathbf{p} \mapsto \mathbf{q}\,\mathbf{p}\,\mathbf{q}^{-1}$ formula can work at all, having $\mathbf{q}\,\mathbf{q}^{-1} = 1$ by definition. Since the multiplication of unit quaternions corresponds to the composition of three-dimensional rotations, this property can be made intuitive by showing that three-dimensional rotations are not commutative in general.

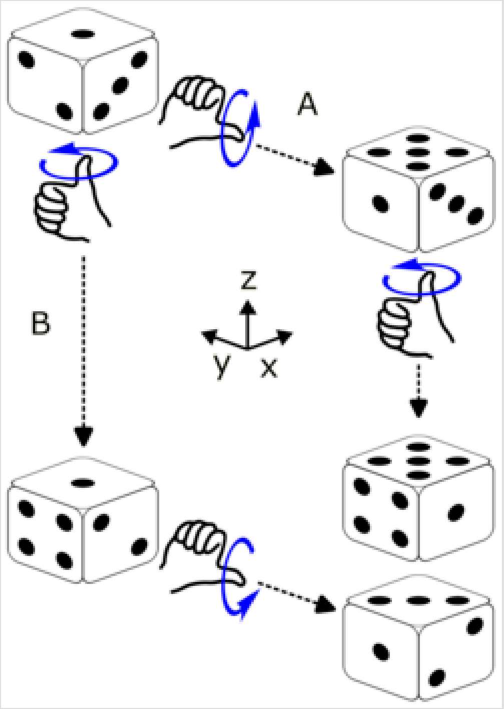
The figure to the right illustrates this with dice. Use the right hand to create a pair of 90 degree rotations. Both dice are initially configured as shown in the upper left-hand corner (with 1 dot on the top face.) Path A begins with a rotation about the $-y$ axis (using the right-hand rule.), followed by a rotation about the $+z$ axis, resulting in the configuration shown in the lower left corner (5 dots on the top face.) Path B reverses the sequence of rotations, resulting with 3 dots on top.

Alternatively, set two books next to each other. Rotate one of them 90 degrees clockwise around the z axis, then flip it 180 degrees around the x axis. Take the other book, flip it 180° around x axis first, and 90° clockwise around z later. The two books do not end up parallel. This shows that, in general, the composition of two different rotations around two distinct spatial axes will not commute.

Orientation

The vector cross product, used to define the axis–angle representation, does confer an orientation ("handedness") to space: in a three-dimensional vector space, the three vectors in the equation $\mathbf{a} \times \mathbf{b} = \mathbf{c}$ will always form a right-handed set (or a left-handed set, depending on how the cross product is defined), thus fixing an orientation in the vector space. Alternatively, the dependence on orientation is expressed in referring to such $\vec{u}$ that specifies a rotation as to *axial vectors*. In quaternionic formalism the choice of an orientation of the space corresponds to order of multiplication: $\mathbf{ij} = \mathbf{k}$ but $\mathbf{ji} = -\mathbf{k}$. If one reverses the orientation, then the formula above becomes $\mathbf{p} \mapsto \mathbf{q}^{-1}\,\mathbf{p}\,\mathbf{q}$, i.e., a unit $\mathbf{q}$ is replaced with the conjugate quaternion – the same behaviour as of axial vectors.

Alternative conventions



Non-commutative rotation of dice

It is reported[16] that the existence and continued usage of an alternative quaternion convention in the aerospace and, to a lesser extent, robotics community is incurring a *significant and ongoing cost* [*sic*]. This alternative convention is proposed by Shuster M.D. in [17] and departs from tradition by reversing the definition for multiplying quaternion basis elements such that under Shuster's convention, $\mathbf{ij} = -\mathbf{k}$ whereas Hamilton's definition is $\mathbf{ij} = \mathbf{k}$. This convention is also referred to as "JPL convention" for its use in some parts of NASA's Jet Propulsion Laboratory.

Under Shuster's convention, the formula for multiplying two quaternions is altered such that

$$(r_1,\ \vec{v}_1)\,(r_2,\ \vec{v}_2) = (r_1 r_2 - \vec{v}_1 \cdot \vec{v}_2,\ r_1\vec{v}_2 + r_2\vec{v}_1 - \vec{v}_1 \times \vec{v}_2)\,, \qquad \text{(Alternative convention, usage discouraged!)}$$

The formula for rotating a vector by a quaternion is altered to be

$$\begin{aligned}\mathbf{p}'_{\text{alt}} &= (\mathbf{v}\otimes\mathbf{v} + q_r^2\mathbf{I} - 2q_r[\mathbf{v}]_\times + [\mathbf{v}]_\times^2)\mathbf{p} \qquad \text{(Alternative convention, usage discouraged!)}\\ &= (\mathbf{I} - 2q_r[\mathbf{v}]_\times + 2[\mathbf{v}]_\times^2)\mathbf{p}\end{aligned}$$

To identify the changes under Shuster's convention, see that the sign before the cross product is flipped from plus to minus.

Finally, the formula for converting a quaternion to a rotation matrix is altered to be

$$\begin{aligned}\mathbf{R}_{alt} &= \mathbf{I} - 2q_r[\mathbf{v}]_\times + 2[\mathbf{v}]_\times^2 \qquad \text{(Alternative convention, usage discouraged!)} \\ &= \begin{bmatrix} 1-2s(q_j^2+q_k^2) & 2(q_iq_j+q_kq_r) & 2(q_iq_k-q_jq_r) \\ 2(q_iq_j-q_kq_r) & 1-2s(q_i^2+q_k^2) & 2(q_jq_k+q_iq_r) \\ 2(q_iq_k+q_jq_r) & 2(q_jq_k-q_iq_r) & 1-2s(q_i^2+q_j^2) \end{bmatrix}\end{aligned}$$

which is exactly the transpose of the rotation matrix converted under the traditional convention.

Software applications by convention used

The table below groups applications by their adherence to either quaternion convention:[16]

Hamilton multiplication convention	Shuster multiplication convention
▪ Wolfram Mathematica ▪ MATLAB Robotics System Toolbox ▪ MATLAB Aerospace Toolbox[19] ▪ ROS ▪ Eigen ▪ Boost quaternions ▪ Quaternion.js ▪ Ceres Solver ▪ SciPy spatial.transform.Rotation library ▪ SymPy symbolic mathematics library ▪ numpy-quaternion (https://github.com/moble/quaternion) library ▪ Universal Scene Description	▪ Microsoft DirectX Math Library

While use of either convention does not impact the capability or correctness of applications thus created, the authors of [16] argued that the Shuster convention should be abandoned because it departs from the much older quaternion multiplication convention by Hamilton and may never be adopted by the mathematical or theoretical physics areas.

Comparison with other representations of rotations

Advantages of quaternions

The representation of a rotation as a quaternion (4 numbers) is more compact than the representation as an orthogonal matrix (9 numbers). Furthermore, for a given axis and angle, one can easily construct the corresponding quaternion, and conversely, for a given quaternion one can easily read off the axis and the angle. Both of these are much harder with matrices or Euler angles.

In video games and other applications, one is often interested in "smooth rotations", meaning that the scene should slowly rotate and not in a single step. This can be accomplished by choosing a curve such as the spherical linear interpolation in the quaternions, with one endpoint being the identity transformation 1 (or some other initial rotation) and the other being the intended final rotation. This is more problematic with other representations of rotations.

When composing several rotations on a computer, rounding errors necessarily accumulate. A quaternion that is slightly off still represents a rotation after being normalized: a matrix that is slightly off may not be orthogonal any more and is harder to convert back to a proper orthogonal matrix.

Quaternions also avoid a phenomenon called gimbal lock which can result when, for example in pitch/yaw/roll rotational systems, the pitch is rotated 90° up or down, so that yaw and roll then correspond to the same motion, and a degree of freedom of rotation is lost. In a gimbal-based aerospace inertial navigation system, for instance, this could have disastrous results if the aircraft is in a steep dive or ascent.

Conversion to and from the matrix representation

From a quaternion to an orthogonal matrix

The orthogonal matrix corresponding to a rotation by the unit quaternion $\mathbf{z} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ (with $|\mathbf{z}| = 1$) when post-multiplying with a column vector is given by

$$R = \begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & a^2 - b^2 - c^2 + d^2 \end{pmatrix}.$$

This rotation matrix is used on vector w as $w_{\text{rotated}} = R \cdot w$. The quaternion representation of this rotation is given by:

$$\begin{bmatrix} 0 \\ w_{\text{rotated}} \end{bmatrix} = z \begin{bmatrix} 0 \\ w \end{bmatrix} z^*,$$

where z^* is the conjugate of the quaternion z, given by $\mathbf{z}^* = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}$

Also, quaternion multiplication is defined as (assuming a and b are quaternions, like z above):

$$ab = \left(a_0 b_0 - \vec{a} \cdot \vec{b}; a_0 \vec{b} + b_0 \vec{a} + \vec{a} \times \vec{b}\right)$$

where the order a, b is important since the cross product of two vectors is not commutative.

A more efficient calculation in which the quaternion does not need to be unit normalized is given by[20]

$$R = \begin{pmatrix} 1 - cc - dd & bc - ad & bd + ac \\ bc + ad & 1 - bb - dd & cd - ab \\ bd - ac & cd + ab & 1 - bb - cc \end{pmatrix},$$

where the following intermediate quantities have been defined:

$$\begin{alignedat}{3} s &= 2/(a \cdot a + b \cdot b + c \cdot c + d \cdot d), \\ bs &= b \cdot s, \quad & cs &= c \cdot s, \quad & ds &= d \cdot s, \\ ab &= a \cdot bs, \quad & ac &= a \cdot cs, \quad & ad &= a \cdot ds, \\ bb &= b \cdot bs, \quad & bc &= b \cdot cs, \quad & bd &= b \cdot ds, \\ cc &= c \cdot cs, \quad & cd &= c \cdot ds, \quad & dd &= d \cdot ds. \end{alignedat}$$

From an orthogonal matrix to a quaternion

One must be careful when converting a rotation matrix to a quaternion, as several straightforward methods tend to be unstable when the trace (sum of the diagonal elements) of the rotation matrix is zero or very small. For a stable method of converting an orthogonal matrix to a quaternion, see the Rotation matrix#Quaternion.

Fitting quaternions

The above section described how to recover a quaternion $\mathbf{q}$ from a 3×3 rotation matrix Q. Suppose, however, that we have some matrix Q that is not a pure rotation—due to round-off errors, for example—and we wish to find the quaternion $\mathbf{q}$ that most accurately represents Q. In that case we construct a symmetric 4×4 matrix

$$K = \frac{1}{3}\begin{bmatrix} Q_{xx}-Q_{yy}-Q_{zz} & Q_{yx}+Q_{xy} & Q_{zx}+Q_{xz} & Q_{zy}-Q_{yz} \\ Q_{yx}+Q_{xy} & Q_{yy}-Q_{xx}-Q_{zz} & Q_{zy}+Q_{yz} & Q_{xz}-Q_{zx} \\ Q_{zx}+Q_{xz} & Q_{zy}+Q_{yz} & Q_{zz}-Q_{xx}-Q_{yy} & Q_{yx}-Q_{xy} \\ Q_{zy}-Q_{yz} & Q_{xz}-Q_{zx} & Q_{yx}-Q_{xy} & Q_{xx}+Q_{yy}+Q_{zz} \end{bmatrix},$$

and find the eigenvector (x, y, z, w) corresponding to the largest eigenvalue (that value will be 1 if and only if Q is a pure rotation). The quaternion so obtained will correspond to the rotation closest to the original matrix Q .[21]

Performance comparisons

This section discusses the performance implications of using quaternions versus other methods (axis/angle or rotation matrices) to perform rotations in 3D.

Results

Storage requirements

Method	Storage
Rotation matrix	9 or 6 (see below)
Quaternion	4 or 3 (see below)
Angle–axis	4 or 3 (see below)

Only three of the quaternion components are independent, as a rotation is represented by a unit quaternion. For further calculation one usually needs all four elements, so all calculations would suffer additional expense from recovering the fourth component. Likewise, angle–axis can be stored in a three-component vector by multiplying the unit direction by the angle (or a function thereof), but this comes at additional computational cost when using it for calculations. Similarly, a rotation matrix requires orthogonal basis vectors, so in 3D space the third vector can unambiguously be calculated from the first two vectors with a cross product (though there is ambiguity in the sign of the third vector if improper rotations are allowed).

Performance comparison of rotation chaining operations

Method	# multiplies	# add/subtracts	total operations
Rotation matrices	27	18	45
Quaternions	16	12	28

Performance comparison of vector rotating operations[22][23]

Method		# multiplies	# add/subtracts	# sin/cos	total operations
Rotation matrix		9	6	0	15
Quaternions *	Without intermediate matrix	15	15	0	30
	With intermediate matrix	21	18	0	39
Angle–axis	Without intermediate matrix	18	13	2	31 + 2
	With intermediate matrix	21	16	2	37 + 2

* Quaternions can be implicitly converted to a rotation-like matrix (12 multiplications and 12 additions/subtractions), which levels the following vectors rotating cost with the rotation matrix method.

Used methods

There are three basic approaches to rotating a vector $\vec{v}$:

1. Compute the matrix product of a 3×3 rotation matrix R and the original 3×1 column matrix representing $\vec{v}$. This requires 3 × (3 multiplications + 2 additions) = 9 multiplications and 6 additions, the most efficient method for rotating a vector.
2. A rotation can be represented by a unit-length quaternion $\mathbf{q} = (w, \vec{r})$ with scalar (real) part w and vector (imaginary) part $\vec{r}$. The rotation can be applied to a 3D vector $\vec{v}$ via the formula $\vec{v}_{\text{new}} = \vec{v} + 2\vec{r} \times (\vec{r} \times \vec{v} + w\vec{v})$. This requires only 15 multiplications and 15 additions to evaluate (or 18 multiplications and 12 additions if the factor of 2 is done via multiplication.) This formula, originally thought to be used with axis/angle notation (Rodrigues' formula), can also be applied to quaternion notation. This yields the same result as the less efficient but more compact formula of quaternion multiplication $\vec{v}_{\text{new}} = q\vec{v}q^{-1}$.
3. Use the angle/axis formula to convert an angle/axis to a rotation matrix R then multiplying with a vector, or, similarly, use a formula to convert quaternion notation to a rotation matrix, then multiplying with a vector. Converting the angle/axis to R costs 12 multiplications, 2 function calls (sin, cos), and 10 additions/subtractions; from item 1, rotating using R adds an additional 9 multiplications and 6 additions for a total of 21 multiplications, 16 add/subtractions, and 2 function calls (sin, cos). Converting a quaternion to R costs 12 multiplications and 12 additions/subtractions; from item 1, rotating using R adds an additional 9 multiplications and 6 additions for a total of 21 multiplications and 18 additions/subtractions.

Performance comparison of n vector rotating operations

Method		# multiplies	# add/subtracts	# sin/cos	total operations
Rotation matrix		9*n*	6*n*	0	15*n*
Quaternions *	Without intermediate matrix	15*n*	15*n*	0	30*n*
	With intermediate matrix	9*n* + 12	6*n* + 12	0	15*n* + 24
Angle–axis	Without intermediate matrix	18*n*	12*n* + 1	2	30*n* + 3
	With intermediate matrix	9*n* + 12	6*n* + 10	2	15*n* + 24

Pairs of unit quaternions as rotations in 4D space

A pair of unit quaternions $\mathbf{z}_{\mathrm{L}} = a_{\mathrm{L}} + b_{\mathrm{L}}i + c_{\mathrm{L}}j + d_{\mathrm{L}}k$ and $\mathbf{z}_{\mathrm{R}} = a_{\mathrm{R}} + b_{\mathrm{R}}i + c_{\mathrm{R}}j + d_{\mathrm{R}}k$ can represent any rotation in 4D space. Given a four-dimensional vector $\vec{v}$, expressed as a quaternion $\vec{v} = w + xi + yj + zk$, we can rotate it as follows:

$$f(\vec{v}) = \mathbf{z}_{\mathrm{L}}\vec{v}\mathbf{z}_{\mathrm{R}} = M_{\mathrm{L}}M_{\mathrm{R}}\vec{v} = \begin{pmatrix} a_{\mathrm{L}} & -b_{\mathrm{L}} & -c_{\mathrm{L}} & -d_{\mathrm{L}} \\ b_{\mathrm{L}} & a_{\mathrm{L}} & -d_{\mathrm{L}} & c_{\mathrm{L}} \\ c_{\mathrm{L}} & d_{\mathrm{L}} & a_{\mathrm{L}} & -b_{\mathrm{L}} \\ d_{\mathrm{L}} & -c_{\mathrm{L}} & b_{\mathrm{L}} & a_{\mathrm{L}} \end{pmatrix} \begin{pmatrix} a_{\mathrm{R}} & -b_{\mathrm{R}} & -c_{\mathrm{R}} & -d_{\mathrm{R}} \\ b_{\mathrm{R}} & a_{\mathrm{R}} & d_{\mathrm{R}} & -c_{\mathrm{R}} \\ c_{\mathrm{R}} & -d_{\mathrm{R}} & a_{\mathrm{R}} & b_{\mathrm{R}} \\ d_{\mathrm{R}} & c_{\mathrm{R}} & -b_{\mathrm{R}} & a_{\mathrm{R}} \end{pmatrix} \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix},$$

where the matrices M_{L} and M_{R} represent left and right quaternion multiplications, respectively. Together, these matrices form an isoclinic decomposition of a rotation in $\mathbb{R}^4$. Since quaternion multiplication is associative, we have:

$$(\mathbf{z}_{\mathrm{L}}\vec{v})\mathbf{z}_{\mathrm{R}} = M_{\mathrm{R}}M_{\mathrm{L}}\vec{v} = M_{\mathrm{L}}M_{\mathrm{R}}\vec{v} = \mathbf{z}_{\mathrm{L}}(\vec{v}\mathbf{z}_{\mathrm{R}}).$$

Thus, the two matrices M_{L} and M_{R} must commute. This implies the existence of two commuting subgroups within the group of four-dimensional rotations. An arbitrary four-dimensional rotation has six degrees of freedom, with each matrix contributing three of these six degrees of freedom.

Since the generators of the four-dimensional rotations can be represented by pairs of quaternions (as follows), all four-dimensional rotations can also be represented.

$$\mathbf{z}_\mathrm{L}\vec{v}\mathbf{z}_\mathrm{R} = \begin{pmatrix} 1 & -dt_{ab} & -dt_{ac} & -dt_{ad} \\ dt_{ab} & 1 & -dt_{bc} & -dt_{bd} \\ dt_{ac} & dt_{bc} & 1 & -dt_{cd} \\ dt_{ad} & dt_{bd} & dt_{cd} & 1 \end{pmatrix} \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix}$$

$$\mathbf{z}_\mathrm{L} = 1 + \frac{dt_{ab} + dt_{cd}}{2} i + \frac{dt_{ac} - dt_{bd}}{2} j + \frac{dt_{ad} + dt_{bc}}{2} k$$

$$\mathbf{z}_\mathrm{R} = 1 + \frac{dt_{ab} - dt_{cd}}{2} i + \frac{dt_{ac} + dt_{bd}}{2} j + \frac{dt_{ad} - dt_{bc}}{2} k$$

See also

- Anti-twister mechanism
- Applications of dual quaternions to 2D geometry
- Binary polyhedral group
- Biquaternion
- Charts on SO(3)
- Clifford algebras
- Conversion between quaternions and Euler angles
- Covering space
- Dual quaternion
- Elliptic geometry
- Olinde Rodrigues
- Rotation formalisms in three dimensions
- Rotation (mathematics)
- Slerp, spherical linear interpolation
- Spin group
- Versor
- William Rowan Hamilton

References

1. Shoemake, Ken (1985). "Animating Rotation with Quaternion Curves" (https://www.cs.cmu.edu/~kiranb/animation/p245-shoemake.pdf) (PDF). *Computer Graphics*. **19** (3): 245–254. doi:10.1145/325165.325242 (https://doi.org/10.1145%2F325165.325242). Presented at SIGGRAPH '85.
2. J. M. McCarthy, 1990, *Introduction to Theoretical Kinematics* (https://books.google.com/books?id=glOqQgAACAAJ&q=introduction+to+theoretical+kinematics), MIT Press
3. Amnon Katz (1996) *Computational Rigid Vehicle Dynamics*, Krieger Publishing ISBN 978-1575240169
4. J. B. Kuipers (1999) *Quaternions and rotation Sequences: a Primer with Applications to Orbits, Aerospace, and Virtual Reality*, Princeton University Press ISBN 978-0-691-10298-6
5. Karsten Kunze, Helmut Schaeben (November 2004). "The Bingham Distribution of Quaternions and Its Spherical Radon Transform in Texture Analysis". *Mathematical Geology*. **36** (8): 917–943. doi:10.1023/B:MATG.0000048799.56445.59 (https://doi.org/10.1023%2FB%3AMATG.0000048799.56445.59). ResearchGate:226385995 (https://www.researchgate.net/publication/226385995).
6. Euclidean and non-Euclidean Geometry. Patrick J. Ryan, Cambridge University Press, Cambridge, 1987.

7. I.L. Kantor. Hypercomplex numbers, Springer-Verlag, New York, 1989.
8. Andrew J. Hanson. Visualizing Quaternions, Morgan Kaufmann Publishers, Amsterdam, 2006.
9. J.H. Conway and D.A. Smith. On Quaternions and Octonions, A.K. Peters, Natick, MA, 2003.
10. "comp.graphics.algorithms FAQ" (http://www.faqs.org/faqs/graphics/algorithms-faq/). Retrieved 2 July 2017.
11. Rodrigues, O. (1840). "Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et la variation des coordonnées provenant de ses déplacements considérés indépendamment des causes qui peuvent les produire" (http://www.numdam.org/item/JMPA_1840_1_5__380_0/). *Journal de Mathématiques Pures et Appliquées*. Série 1. **5**: 380–440.

    Translated in Friedberg, Richard (2022). "Rodrigues, Olinde: "Des lois géométriques qui régissent les déplacements d'un système solide...", translation and commentary". arXiv:2211.07787 (https://arxiv.org/abs/2211.07787) [math.HO (https://arxiv.org/archive/math.HO)].
12. William Rowan Hamilton (1844 to 1850) On quaternions or a new system of imaginaries in algebra (http://www.maths.tcd.ie/pub/HistMath/People/Hamilton/OnQuat/), Philosophical Magazine, link to David R. Wilkins collection at Trinity College, Dublin
13. Lee, Byung-Uk (1991), "Differentiation with Quaternions, Appendix B" (http://home.ewha.ac.kr/~bulee/quaternion.pdf) (PDF), *Stereo Matching of Skull Landmarks* (Ph. D. thesis), Stanford University, pp. 57–58
14. Altmann, Simon L. (1989). "Hamilton, Rodrigues, and the Quaternion Scandal". *Mathematics Magazine*. **62** (5): 306. doi:10.2307/2689481 (https://doi.org/10.2307%2F2689481). JSTOR 2689481 (https://www.jstor.org/stable/2689481).
15. Simon L. Altman (1986) *Rotations, Quaternions, and Double Groups*, Dover Publications (see especially Ch. 12).
16. Sommer, H. (2018), "Why and How to Avoid the Flipped Quaternion Multiplication", *Aerospace*, **5** (3): 72, arXiv:1801.07478 (https://arxiv.org/abs/1801.07478), Bibcode:2018Aeros...5...72S (https://ui.adsabs.harvard.edu/abs/2018Aeros...5...72S), doi:10.3390/aerospace5030072 (https://doi.org/10.3390%2Faerospace5030072), ISSN 2226-4310 (https://search.worldcat.org/issn/2226-4310)
17. Shuster, M.D (1993), "A Survey of attitude representations", *Journal of the Astronautical Sciences*, **41** (4): 439–517, Bibcode:1993JAnSc..41..439S (https://ui.adsabs.harvard.edu/abs/1993JAnSc..41..439S), ISSN 0021-9142 (https://search.worldcat.org/issn/0021-9142)
18. "MATLAB Aerospace Toolbox quatrotate" (https://www.mathworks.com/help/aerotbx/ug/quatrotate.html).
19. The MATLAB Aerospace Toolbox uses the Hamilton multiplication convention, however because it applies *passive* rather than *active* rotations, the quaternions listed are in effect active rotations using the Shuster convention.[18]
20. Alan Watt and Mark Watt (1992) *Advanced Animation and Rendering Techniques: Theory and Practice*, ACM Press ISBN 978-0201544121
21. Bar-Itzhack, Itzhack Y. (Nov–Dec 2000), "New method for extracting the quaternion from a rotation matrix", *Journal of Guidance, Control and Dynamics*, **23** (6): 1085–1087, Bibcode:2000JGCD...23.1085B (https://ui.adsabs.harvard.edu/abs/2000JGCD...23.1085B), doi:10.2514/2.4654 (https://doi.org/10.2514%2F2.4654), ISSN 0731-5090 (https://search.worldcat.org/issn/0731-5090)
22. Eberly, D., Rotation Representations and performance issues
23. "Bitbucket" (https://web.archive.org/web/20151117033447/https://bitbucket.org/eigen/eigen/src/4111270ba6e10882f8a8c571dbb660c8e9150138/Eigen/src/Geometry/Quaternion.h?at=default&fileviewer=file-view-default#Quaternion.h-469). *bitbucket.org*. Archived from the original (https://bitbucket.org/eigen/eigen/src/4111270ba6e10882f8a8c571dbb660c8e9150138/Eigen/src/Geometry/Quaternion.h?at=default&fileviewer=file-view-default#Quaternion.h-469) on 2015-11-17. Retrieved 2015-11-14.

Further reading

- Grubin, Carl (1970). "Derivation of the quaternion scheme via the Euler axis and angle". *Journal of Spacecraft and Rockets*. **7** (10): 1261–1263. Bibcode:1970JSpRo...7.1261G (https://ui.adsabs.harvard.edu/abs/1970JSpRo...7.1261G). doi:10.2514/3.30149 (https://doi.org/10.2514%2F3.30149).
- Battey-Pratt, E. P.; Racey, T. J. (1980). "Geometric Model for Fundamental Particles". *International Journal of Theoretical Physics*. **19** (6): 437–475. Bibcode:1980IJTP...19..437B (https://ui.adsabs.harvard.edu/abs/1980IJTP...19..437B). doi:10.1007/BF00671608 (https://doi.org/10.1007%2FBF00671608). S2CID 120642923 (https://api.semanticscholar.org/CorpusID:120642923).
- Arribas, M.; Elipe, A.; Palacios, M. (2006). "Quaternions and the rotations of a rigid body". *Celest. Mech. Dyn. Astron*. **96** (3–4): 239–251. Bibcode:2006CeMDA..96..239A (https://ui.adsabs.harvard.edu/abs/2006CeMDA..96..239A). doi:10.1007/s10569-006-9037-6 (https://doi.org/10.1007%2Fs10569-006-9037-6). S2CID 123591599 (https://api.semanticscholar.org/CorpusID:123591599).

External links and resources

- Shoemake, Ken. "Quaternions" (http://www.cs.ucr.edu/~vbz/resources/quatut.pdf) (PDF). Archived (https://web.archive.org/web/20200503045740/http://www.cs.ucr.edu/~vbz/resources/quatut.pdf) (PDF) from the original on 2020-05-03.
- "Simple Quaternion type and operations in over seventy-five computer languages" (http://rosettacode.org/wiki/Simple_Quaternion_type_and_operations). on Rosetta Code
- Hart, John C. "Quaternion Demonstrator" (http://graphics.stanford.edu/courses/cs348c-95-fall/software/quatdemo).
- Dam, Eik B.; Koch, Martin; Lillholm, Martin (1998). "Quaternions, Interpolation and Animation" (http://web.mit.edu/2.998/www/QuaternionReport1.pdf) (PDF).
- Leandra, Vicci (2001). "Quaternions and Rotations in 3-Space: The Algebra and its Geometric Interpretation" (http://www.cs.unc.edu/techreports/01-014.pdf) (PDF).
- Howell, Thomas; Lafon, Jean-Claude (1975). "The Complexity of the Quaternion Product, TR75-245" (http://world.std.com/~sweetser/quaternions/ps/cornellcstr75-245.pdf) (PDF). Cornell University.
- Horn, Berthold K.P. (2001). "Some Notes on Unit Quaternions and Rotation" (http://people.csail.mit.edu/bkph/articles/Quaternions.pdf) (PDF).
- Lee, Byung-Uk (1991). *Unit Quaternion Representation of Rotation - Appendix A, Differentiation with Quaternions - Appendix B* (http://home.ewha.ac.kr/~bulee/quaternion.pdf) (PDF) (Ph. D. Thesis). Stanford University.
- Vance, Rod. "Some examples of connected Lie groups" (http://www.wetsavannaanimals.net/wordpress/some-examples-of-connected-lie-groups/). Archived (https://web.archive.org/web/20181215191041/http://www.wetsavannaanimals.net/wordpress/some-examples-of-connected-lie-groups/) from the original on 2018-12-15.
- "Visual representation of quaternion rotation" (https://eater.net/quaternions).

Retrieved from "https://en.wikipedia.org/w/index.php?title=Quaternions_and_spatial_rotation&oldid=1318439832"