Applications of Quaternions in 3D Rotation

Martin Velikov

To what extent can quaternions supersede rotation matrices constructed from Euler angles in representing 3D rotation?

Contents

1 Introduction

Historically, mathematicians have endeavoured to describe the world around us using mathematics. While mathematicians often create simplifications or models in lower dimensions, as is the case with fields of mathematics like trigonometry and complex numbers, we are ultimately inhabitants of a three dimensional world. In the modern day, demand for working in three dimensions is ever increasing, with scientific applications in aerodynamic modeling, astrophysics, weather modelling and computational chemistry, in addition to numerous uses in computer generated imagery (CGI). In all of these use cases, there is a need for a rigorous mathematical framework to represent and manipulate the 3D world.

One of the intuitively simplest, but surprisingly complex aspects of transitioning from a two dimensional space to three dimensional is the problem of rotation. Indeed, there currently exist several competing mathematical models for 3D rotation in common use which bear their own strengths and weaknesses. Of these, some of the most commonly seen constructs are 3D "rotation matrices", which are usually constructed from "Euler angles", and Quaternions.

This investigation aims to compare and contrast these methods of rotation in order to draw conclusions about the usefulness and applicability of Quaternions in 3D computer graphics and scientific simulations compared to rotation matrices constructed from Euler angles.

2 Rotation Matrices

Rotation matrices are considered one of the most versatile ways of doing 3D rotation. They are often used in computer animation and praised for their computational efficiency and versatility. Mathematically, they represent specific a 3D linear transformation which, when applied to a set of points, describes a 3D rotation.

2.1 In two dimensions

It is worthwile conceptually exploring what happens in two dimensions first, as this will introduce valuable concepts which will help with understanding how three dimensional matrices work.

Let 2D points be represented as a vector of the form $\vec{v} = \begin{bmatrix} x \\ y \end{bmatrix}$ where x and y represent the x and y coordinates of the point respective to the origin.

By representing the points as vectors, rotation of a point around the origin would be equrivalent to rotating the vector. Consider the rotation of point (1, 0) by 45 degrees anti-clockwise:

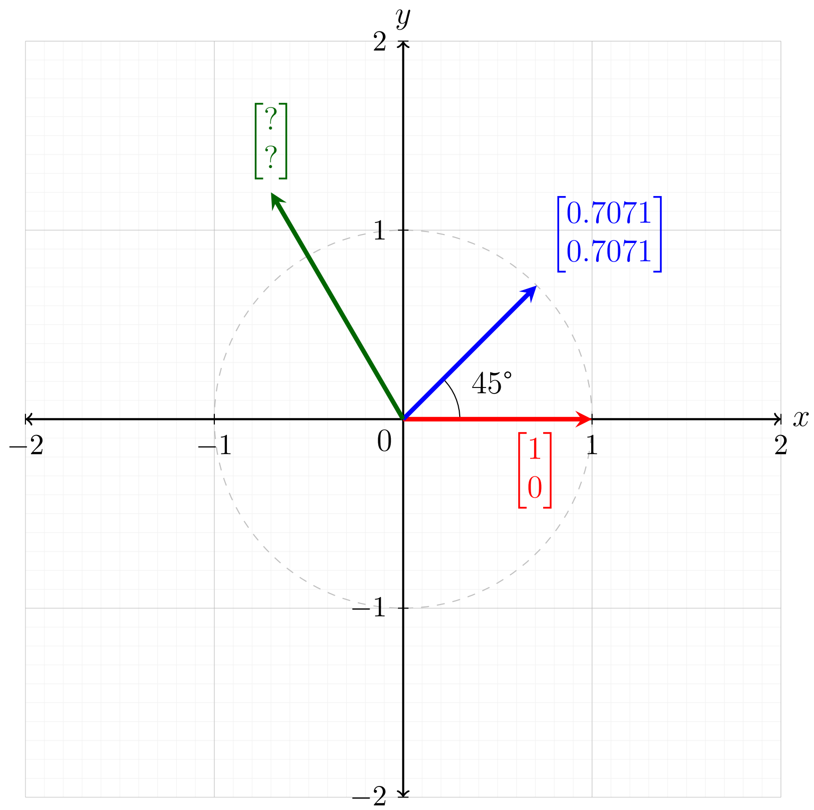

Figure 1: Apparent rotation of red vector $\begin{bmatrix}1\\0\end{bmatrix}$ by 45° anti-clockwise to blue vector

From Figure 1, it may seem like calculating this rotation is trivial by using the trigonometric functions: for any angle θ, the rotation of $\begin{bmatrix}1\\0\end{bmatrix}$ is equal to $\begin{bmatrix}\cos\theta\\\sin\theta\end{bmatrix}$.

Indeed, this is the case for the simple example above, but the applicability of this model quickly breaks down once we increase the complexity. For example, rotating the green vector about a point different to the origin by using this model is difficult to imagine. Furthermore, if we were to go beyond rotation, how could we, say, transform the red vector into the green vector?

Linear transformation using matrices elegantly deals with these problems by introducing one concept - instead of transforming each individual vector, transform the axes themselves.

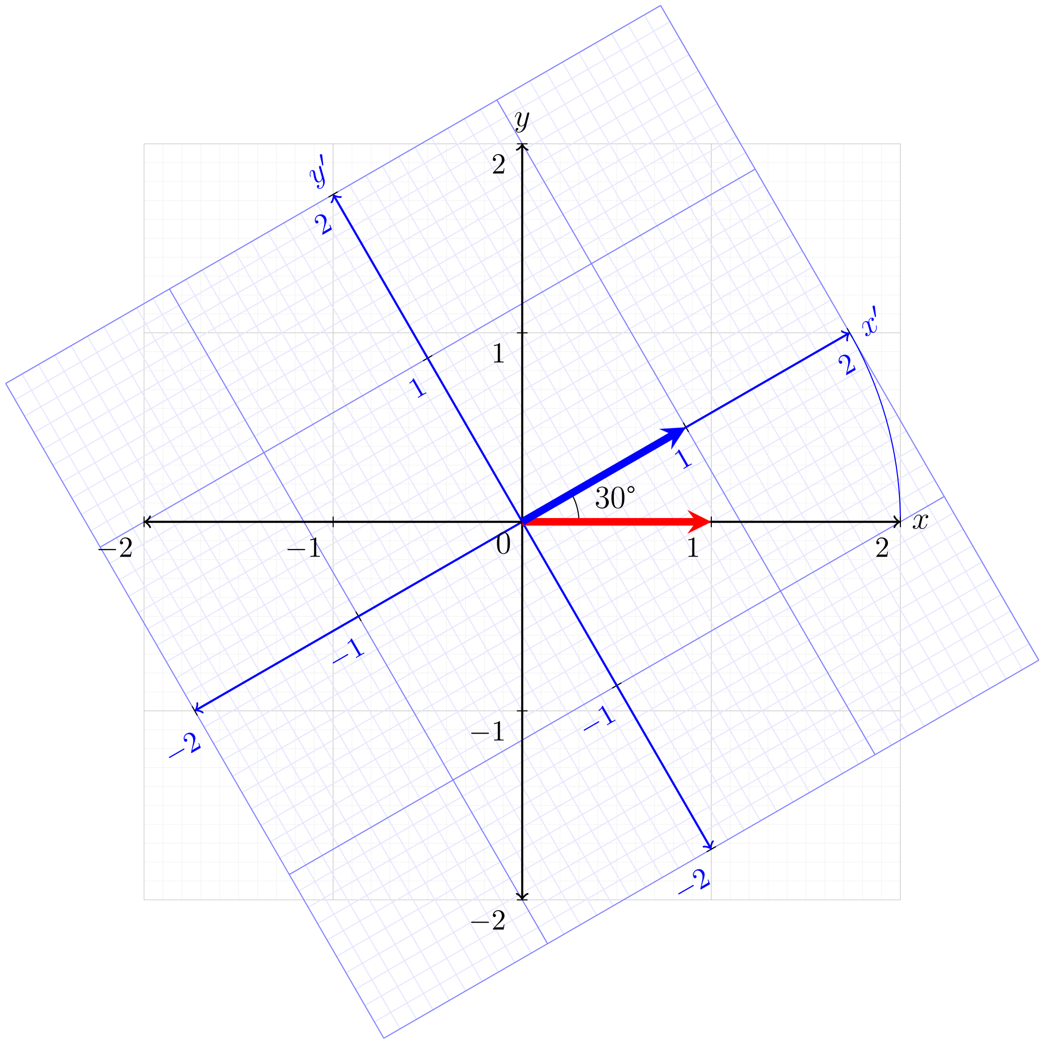

Figure 2: Overlay of new cartesian plane (blue) with axes x' and y' rotated 30° anti-clockwise relative to the original.

In Figure 2, the blue vector's coordinates on the rotated axes are identical to the red vector's coordinates on the original - both are $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$. In fact, we find that any one point on our original axis has a counterpart on the rotated axis which corresponds to the rotation of that point about the origin by 30°. If we were to examine the coordinates of the blue vector $\vec{b}$ on the original axes, we would find that, in fact, $\vec{b} = \begin{bmatrix} 0.8660\ldots \\ 0.5 \end{bmatrix} = \begin{bmatrix} \cos 30° \\ \sin 30° \end{bmatrix}$, which aligns with findings from the trigonometric model.

Another invaluable benefit of this method is that we may also choose to rotate the axes about a different point to the origin, and observe the correct off-centric rotations for vectors without any changes to how the model is evaluated. Extrapolating this to three dimensions yields immeasurable savings in complexity and extra calculations which are expensive for computers to do.

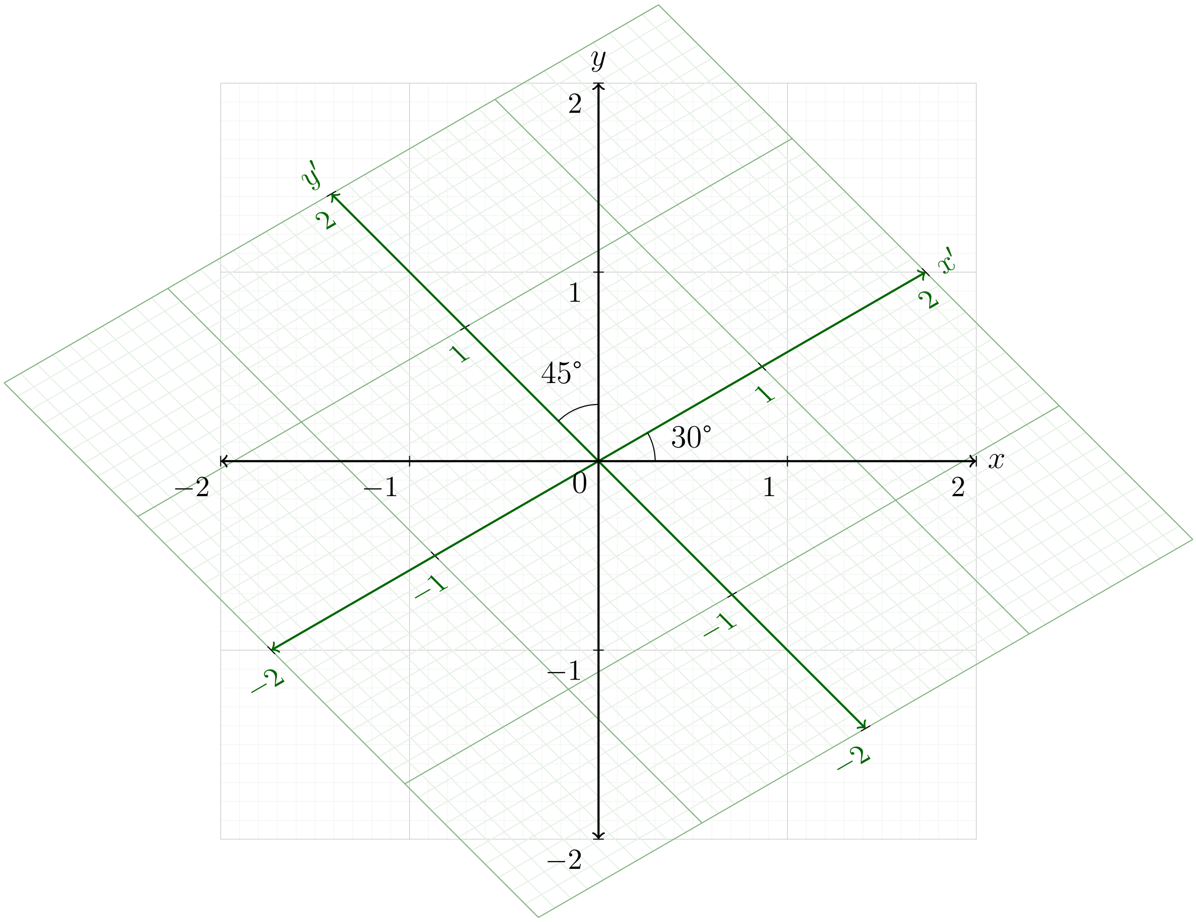

Figure 3: Overlay of new cartesian plane (green) with axes x' and y' rotated 30° anti-clockwise and 45° anti-clockwise respectively.

Figure 3 depicts what is known as a "shear". It occurs when each axis is rotated by a different angle, and results in warping of the transformed vectors, evident by the asymmetric appearance of the graph. If you were interested in going beyond rotation, there is an opportunity to translate, dilate and shear the axes to achieve various effects like the one depicted.

The elegance in this model comes from how we begin to compute one such transformation.

The first step to this is determining the direction and dilation of the transformed axes by examining what happens to the vectors $\begin{bmatrix}1\\0\end{bmatrix}$ and $\begin{bmatrix}0\\1\end{bmatrix}$ after the transformation. If the magnitude of any one of these vectors changes, that means that the respective axis has been dilated by a factor of the magnitude. If the direction changes, that means that that axis has been rotated. By knowing the final position of these two vectors after a transformation, the transformation that took place can be reconstructed and applied to more points.

The extremely useful tool that bridges the gap between this theory of axis shifting and mathematics is called a transformation matrix. 2D Transformation matrices are constructed by taking the vectors found above and storing them in a single 2×2 matrix:

Let $\vec{x}$ be the post-transform position of $\begin{bmatrix}1\\0\end{bmatrix}$, where $\vec{x} = \begin{bmatrix}\vec{x}_x\\\vec{x}_y\end{bmatrix}$, and let $\vec{y}$ be the post-transform position of $\begin{bmatrix}0\\1\end{bmatrix}$, where $\vec{y} = \begin{bmatrix}\vec{y}_x\\\vec{y}_y\end{bmatrix}$. The transformation matrix M is described by $M = \begin{bmatrix}\vec{x}_x & \vec{y}_x\\\vec{x}_y & \vec{y}_y\end{bmatrix}$.

The transformation $T_M(\vec{v})$ of $\vec{v}$ by matrix M can then be described as

$$T_M(\vec{v}) = M\vec{v} = \begin{bmatrix}\vec{x}_x & \vec{y}_x\\\vec{x}_y & \vec{y}_y\end{bmatrix}\begin{bmatrix}\vec{v}_x\\\vec{v}_y\end{bmatrix} = \begin{bmatrix}\vec{x}_x\vec{v}_x + \vec{y}_x\vec{v}_y\\\vec{x}_y\vec{v}_x + \vec{y}_y\vec{v}_y\end{bmatrix}$$

If $\vec{x}$ and $\vec{y}$ do not change after a transformation, they would be equal to $\begin{bmatrix}1\\0\end{bmatrix}$ and $\begin{bmatrix}0\\1\end{bmatrix}$ respectively. From this, a transformation matrix which does nothing (called the "identity") M_0 can be constructed in the form of $M_0 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$. In this case, $T_{M_0}(\vec{v}) = \begin{bmatrix}(1)\vec{v}_x + (0)\vec{v}_y\\(0)\vec{v}_x + (1)\vec{v}_y\end{bmatrix} = \begin{bmatrix}\vec{v}_x\\\vec{v}_y\end{bmatrix} = \vec{v}$

2.2 Object & world axes

Any object transformation can be represented by where its three axes "end up" after the transformation. The direction of each axis is represented as a 3D vector of norm one, which "points" in the desired direction. However, the meaning of the "object's axes" is ambigous here. It is important to make the distinction between the "world axes" which represent the axes of the surrounding 3D scene, and the "object axes", which describe it in its own space, as if it were the only object in a scene. The useful property of this abstraction is that the object axes do not necessary have to be oriented the same as the world axes, and this can be leveraged for transformation.

The three object X,Y,Z axes can be represented by the vectors $\vec{x}$, $\vec{y}$, $\vec{z}$:

$$\vec{x} = \begin{bmatrix}1\\0\\0\end{bmatrix} \quad \vec{y} = \begin{bmatrix}0\\1\\0\end{bmatrix} \quad \vec{z} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

This orientation of the axes is called the "identity". It describes a space where all three axes are perpendicular, and are aligned with the world axes.

2.3 Rotating points

Whenever a point $\vec{p}$ has to be rotated, its X,Y and Z components (p_x, p_y and p_z respectively) can be multiplied by their respective object axis.

$$\vec{a} = \vec{x}p_x \quad \vec{b} = \vec{y}p_y \quad \vec{c} = \vec{z}p_z$$

The resulting vectors are then added together for the final position of the transformed point $\vec{r}$.

$$\vec{r} = \vec{a} + \vec{b} + \vec{c}$$

2.3.1 3×3 Matrix form

The process of transforming $\vec{p} \rightarrow \vec{r}$ is highly reminiscent of matrix multiplication. It is useful to represent $\vec{x}$, $\vec{y}$, $\vec{z}$ as a single 3×3 matrix of their components M of the form

$$M = \begin{bmatrix} \vec{x}_x & \vec{y}_x & \vec{z}_x \\ \vec{x}_y & \vec{y}_y & \vec{z}_y \\ \vec{x}_z & \vec{y}_z & \vec{z}_z \end{bmatrix}$$

The transformation of $\vec{p}$ to $\vec{r}$ can then be described by $M\vec{p}$ because of the properties of matrix multiplication, in that

$$M\vec{p} = \vec{M}_1 p_x + \vec{M}_2 p_y + \vec{M}_3 p_z \quad \text{where } M_n \text{ represents the } n\text{th column of } M.$$

Since the columns $\vec{M}_1$, $\vec{M}_2$, $\vec{M}_3$ correspond to the object axes X, Y and Z respectively, this is equivalent to the rotation equation shown in Section 2.3.

In this form, the identity axes are represented by the matrix $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

2.4 Construction from Euler angles

2.4.1 Concept

In real life, it is practical to represent rotations with easily measurable metrics, like angles. Aircraft and watercraft make extensive use of yaw, pitch and roll angles in instrumentation to aid their pilots in judging the orientation of the craft.

Euler angles are based around this concept. A set of Euler angles is a 3D vector containing the yaw, pitch and roll of an object from a reference frame. In the example of aircraft, the reference frame is usually the surface of the Earth. In space, it is usually a prominent constellation or celestial object.

It is possible to construct rotation matrices for a rotation of θ around each axis.

2.4.2 Deriving rotation matrices for each angle

Consider rotating the identity axes by θ around the X axis. Let the rotation matrix that describes this transformation be R_x.

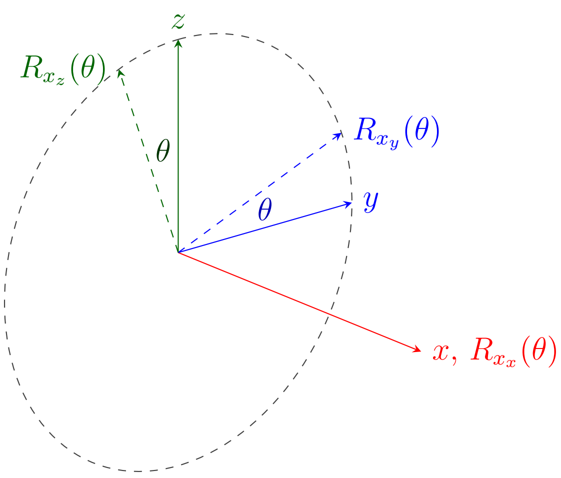

Figure 4: Rotation around the X-axis by angle θ. R_{x_x}, R_{x_y} and R_{x_z} represent the object axes after the transformation by $R_x(\theta)$ where $x \to R_{x_x}, y \to R_{x_y}$ and $z \to R_{x_z}$

From Figure 4 it becomes apparent that points that lie along the identity X-axis do not move at all. Hence, for the first column R_{x_x} representing the object X-axis post-transform,

$$R_{x_x}(\theta) = x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Points that lie along the Y axis get transformed against a circle of radius one along the YZ plane, meaning its coordinates are described by the trigonometric functions sine and cosine. Hence,

$$R_{x_y}(\theta) = \begin{bmatrix} 0 \\ \cos\theta \\ \sin\theta \end{bmatrix}$$

The Z-axis corresponds to the circle's Y-axis on a cartesian plane (along the 3D YZ plane), so it is as if the Z-axis was rotated by a right angle, then by θ. Hence,

$$R_{x_z}(\theta) = \begin{bmatrix} 0 \\ \cos\left(\theta + \frac{\pi}{2}\right) \\ \sin\left(\theta + \frac{\pi}{2}\right) \end{bmatrix} = \begin{bmatrix} 0 \\ -\sin\theta \\ \cos\theta \end{bmatrix}$$

.

Finally, combining R_{x_x}, R_{x_y} and R_{x_z},

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

A very similar process can be followed for the rotation matrices by the Y (R_y) and Z (R_z) axes, so it has been omitted for brevity. The only thing that changes in the process is that the plane of the circle follows the two axes perpendicular to the rotation axis.

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.4.3 Multiplication

For the Euler angles α, β and γ representing yaw, pitch and roll respectively, it is now possible to compute a general rotation matrix by first constructing each individual matrix, then multiplying them.

$$\begin{aligned} R(\alpha,\beta,\gamma) &= R_z(\gamma)R_y(\beta)R_x(\alpha) \\ &= \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \\ &= \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{bmatrix} \end{aligned}$$

As matrix multiplication does not commute, it is not always the case that rotating by an axis A, then rotating by B would be the same as rotating first by axis B, then by A. Hence, it is important to note that the order of multiplication of R_z, R_y and R_x produces a unique final rotation. For the rest of the investigation, the $z \to y \to x$ rotation order seen above will be used. This is alternatively called the ZYX Tait-Bryan order.

2.5 The gimball lock problem

There is one large problem with constructing rotation matrices from Euler angles. Suppose you rotated an object with the Euler angles α, β and γ, where $\beta = \frac{\pi}{2}$

$$\begin{aligned} R(\alpha,\frac{\pi}{2},\gamma) &= \begin{bmatrix} \cos\frac{\pi}{2}\cos\gamma & \sin\alpha\sin\frac{\pi}{2}\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\frac{\pi}{2}\cos\gamma + \sin\alpha\sin\gamma \\ \cos\frac{\pi}{2}\sin\gamma & \sin\alpha\sin\frac{\pi}{2}\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\frac{\pi}{2}\sin\gamma - \sin\alpha\cos\gamma \\ -\sin\frac{\pi}{2} & \sin\alpha\cos\frac{\pi}{2} & \cos\alpha\cos\frac{\pi}{2} \end{bmatrix} \\ &= \begin{bmatrix} 0 & \sin\alpha\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\gamma \\ 0 & \sin\alpha\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\gamma - \sin\alpha\cos\gamma \\ -1 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \sin(\alpha-\gamma) & \cos(\alpha-\gamma) \\ 0 & \cos(\alpha-\gamma) & -\sin(\alpha-\gamma) \\ -1 & 0 & 0 \end{bmatrix} \end{aligned}$$

Notice how in the above matrix, changing the value of both α and γ would result in a rotation around the world Z-axis as

$$R_1 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

Hence, one degree of freedom is lost here - it is not possible to rotate the object by any other axis than the Z world axis, by changing either α or γ. This problem specifically occurs whenever you rotate around an axis by 90 degrees, and "align" it with another.

This fundamental flaw with Euler angles demonstrates how despite their simplicity and intuitiveness, they are not always the best choice for representing rotations.

In the Apollo 11 mission, (talk about gimball lock)

2.6 The spherical linear interpolation problem

3 Quaternions

3.1 Axis-angle

Another interesting way to represent the rotation of an object

3.2 Standard Form

Quaternions, like complex numbers, are defined with "real" and "imaginary" parts, and are essentially an extension of the complex numbers to four dimensions. The set of all quaternions is known as $\mathbb{H}$, named after the last initial of the Irish mathematician William Rowan Hamilton. Algebraically, a quaternion q can be defined in terms of the coefficients of its terms:

$$q = a + bi + cj + dk \quad a, b, c, d \in \mathbb{R}$$

i, j and k, called "basic quaternions", do not have an explicit definition of their value but are rather defined expressly in terms of the way they interact with each other in that they must satisfy the equality

$$i^2 = j^2 = k^2 = ijk = -1$$

3.3 Basic Quaternions

3.3.1 Multiplying by Real Numbers

For any $n \in \mathbb{R}$, it is defined that $in = ni$, $jn = nj$ and $kn = nk$. Hence, for any $q \in \mathbb{H}$, $qn = nq$. Quaternion multiplication by real numbers does, in fact, commute.

3.3.2 Multiplication by other basic quaternions

Hamilton's quaternion definition can then be used to derive the multiplicative interactions between i, j and k:

$$\begin{aligned} ijk &= k^2 \\ ijk^2 &= k^2 k \\ ij(-1) &= (-1)k \\ ij &= k \end{aligned} \qquad \begin{aligned} ijk &= i^2 \\ i^2 jk &= i \cdot i^2 \\ (-1)jk &= i(-1) \\ jk &= i \end{aligned}$$

$$\begin{aligned} i &= jk \\ ji &= j^2 k \\ ji &= -k \end{aligned} \qquad \begin{aligned} k &= ij \\ kj &= ij^2 \\ kj &= -i \end{aligned}$$

$$\begin{aligned} ij &= k \\ kij &= k^2 \\ kij^2 &= k^2 j \\ ki(-1) &= (-1)j \\ ki &= j \end{aligned} \qquad \begin{aligned} k &= ij \\ ik &= i^2 j \\ ik &= -j \end{aligned}$$

An important concept becomes apparent from the above calculations - quaternion multiplication by other quaternions is not commutative, that is, it can be the case that $q_1 q_2 \neq q_2 q_1$ where $q_1, q_2 \in \mathbb{H}$. For example, it is seen above that while $ij = k$, $ji = -k$.

The multiplication table of basic quaternions is hence formed:

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

Table 1: Basic quaternion noncommutative multiplication table

3.3.3 Associativity

Quaternion multiplication is associative in that $(q_1 q_2)q_3 = q_1(q_2 q_3)$ where $q_1, q_2, q_3 \in \mathbb{H}$. The same property applies for addition, $(q_1 + q_2) + q_3 = q_1 + (q_2 + q_3)$ where $q_1, q_2, q_3 \in \mathbb{H}$.

This associativity allows for application of useful algebraic techniques like the distributive law.

3.4 Quaternion Operations

3.4.1 Multiplication of non-basic quaternions

Let $q_1 = a_1 + b_1 i + c_1 j + d_1 k$ and $q_2 = a_2 + b_2 i + c_2 j + d_2 k$ where $a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2 \in \mathbb{R}$. The multiplication $q_1 q_2$ can be computed using the distrivutive law.

$$
\begin{aligned}
q_1q_2 &= (a_1 + b_1i + c_1j + d_1k)(a_2 + b_2i + c_2j + d_2k) \\
&= a_1a_2 + a_1b_2i + a_1c_2j + a_1d_2k \\
&+ b_1a_2i + b_1b_2i^2 + b_1c_2ij + b_1d_2ik \\
&+ c_1a_2j + c_1b_2ji + c_1c_2j^2 + c_1d_2jk \\
&+ d_1a_2k + d_1b_2ki + d_1c_2kj + d_1d_2k^2 \\
&\text{Applying the basic quaternion rules then} \\
&\text{factoring out the real part and } i, j, \text{ and } k, \\
&= a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2 \\
&+ (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)i \\
&+ (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)j \\
&+ (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)k
\end{aligned}
$$

3.4.2 Conjugation

For any quaternion $q = a + bi + cj + dk \quad a, b, c, d \in \mathbb{R}$, its conjugate q^* is defined as $q^* = a - bi - cj - dk$.

3.4.3 Multiplication by Conjugate

Suppose one was to multiply a quaternion q by its conjugate q^*,

$$
\begin{aligned}
qq^* &= (a + bi + cj + dk)(a - bi - cj - dk) \\
&= a^2 - abi - acj - adk \\
&+ bia - (bi)^2 - bicj - bidk \\
&+ cja - cjbi - (cj)^2 - cjdk \\
&+ dka - dkbi - dkcj - (dk)^2 \\
&\text{Reordering real coefficients and expanding} \\
&= a^2 - abi - acj - adk \\
&+ abi - b^2i^2 - bcij - bdik \\
&+ acj - bcji - c^2j^2 - cdjk \\
&+ adk - bdki - cdkj - d^2k^2 \\
&\text{Applying basic quaternion rules} \\
&= a^2 - abi - acj - adk \\
&+ abi + b^2 - bck + bdj \\
&+ acj + bck + c^2 - cdi \\
&+ adk - bdj + cdi + d^2 \\
\\
&a, b, c, d \in \mathbb{R} \\
&\therefore a^2 + b^2 + c^2 + d^2 \in \mathbb{R}
\end{aligned}
$$

Hence, if you multiply a quaternion by its conjugate, the result will always be a real number and equal to the sum of the squares of its coefficients.

3.4.4 Inverse

The inverse q^{-1} of a quaternion $q = a + bi + cj + dk$ exists such that $qq^{-1} = 1$, effectively "undoing" any multiplication caused by q. The inverse of a quaternion can be computed using the previosuly established rules.

$$qq^{-1} = 1 \Rightarrow q^{-1} = \frac{1}{q}$$
$$\frac{1}{q} \cdot \frac{q^*}{q^*} = \frac{q^*}{qq^*} = \frac{q^*}{a^2 + b^2 + c^2 + d^2}$$

3.5 Proof that qpq^{-1} returns a pure quaternion

Whenever a pure quaternion is multiplied by a rotation quaternion, the transformation inevitably distorts it into the fourth dimension.

$$\begin{aligned}
qp = {} & (a_q a_p - b_q b_p - c_q c_p - d_q d_p) \\
& + (a_q b_p + b_q a_p + c_q d_p - d_q c_p)i \\
& + (a_q c_p - b_q d_p + c_q a_p + d_q b_p)j \\
& + (a_q d_p + b_q c_p - c_q b_p + d_q a_p)k \\
& \text{Let:} \\
a_{qp} = {} & a_q a_p - b_q b_p - c_q c_p - d_q d_p, \\
b_{qp} = {} & a_q b_p + b_q a_p + c_q d_p - d_q c_p, \\
c_{qp} = {} & a_q c_p - b_q d_p + c_q a_p + d_q b_p, \\
d_{qp} = {} & a_q d_p + b_q c_p - c_q b_p + d_q a_p
\end{aligned}$$

$$\begin{aligned}
& \text{As q is of norm one, } q^{-1} = q^* \\
& \therefore qpq^{-1} = qpq^* \\
& = a_{qp}a_q + b_{qp}b_q + c_{qp}c_q + d_{qp}d_q \\
& + (-a_{qp}b_q + b_{qp}a_q - c_{qp}d_q + d_{qp}c_q)i \\
& + (-a_{qp}c_q + b_{qp}d_q + c_{qp}a_q - d_{qp}b_q)j \\
& + (-a_{qp}d_q - b_{qp}c_q + c_{qp}b_q + d_{qp}a_q)k
\end{aligned}$$

In order for qpq^* to be pure, its real part must be equal to zero.
Let a_{qpq^*} be the real part of qpq^*.

$$
\begin{aligned}
a_{qpq^*} &= a_{qp}a_q + b_{qp}b_q + c_{qp}c_q + d_{qp}d_q \\
&= a_q(a_qa_p - b_qb_p - c_qc_p - d_qd_p) \\
&+ b_q(a_qb_p + b_qa_p + c_qd_p - d_qc_p) \\
&+ c_q(a_qc_p - b_qd_p + c_qa_p + d_qb_p) \\
&+ d_q(a_qd_p + b_qc_p - c_qb_p + d_qa_p) \\
&= a_q^2a_p - a_qb_qb_p - a_qc_qc_p - a_qd_qd_p \\
&+ b_qa_qb_p + b_q^2a_p + b_qc_qd_p - b_qd_qc_p \\
&+ c_qa_qc_p - c_qb_qd_p + c_q^2a_p + c_qd_qb_p \\
&+ d_qa_qd_p + d_qb_qc_p - d_qc_qb_p + d_q^2a_p \\
&= a_q^2a_p + b_q^2a_p + c_q^2a_p + d_q^2a_p \\
&= a_p(a_q^2 + b_q^2 + c_q^2 + d_q^2) \\
&= a_p = 0 \\
&\therefore qpq^{-1} \text{ is pure for } ||q||^2 = 1, \mathfrak{Re}(p) = 0
\end{aligned}
$$