Quaternions (Math Project)

Neil Nadella

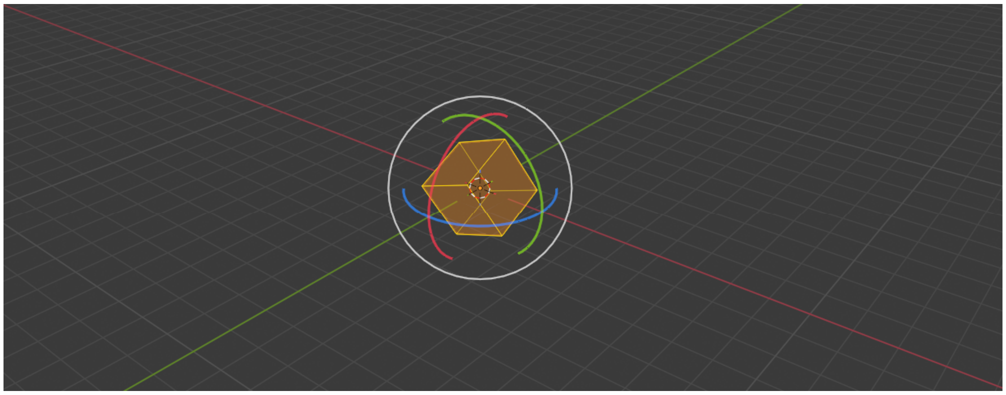
January 1, 2026

Contents

1 Introduction to rotation and Euler Angles

While using euler angles to rotate an object in 3d, we may often run into the Gimbal lock. To properly understand what the gimbal lock is, we must first understand euler rotations.

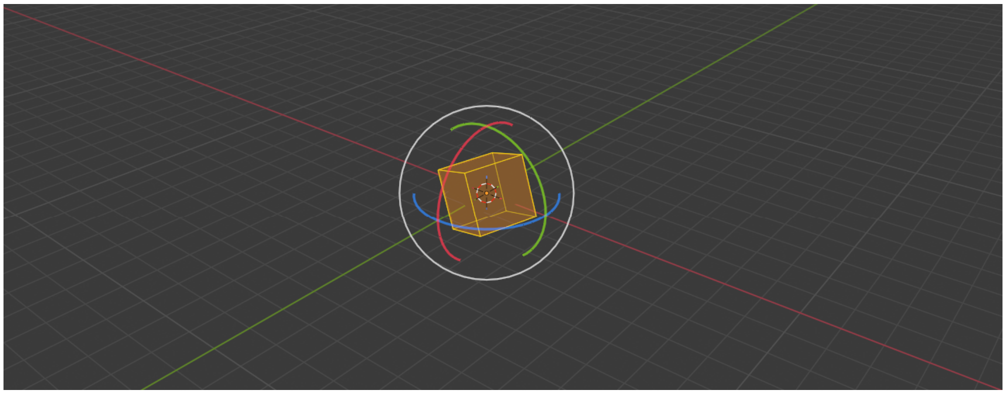
Euler rotations are pretty intuitive, to acquire a desired orientation of an object, we rotate the X, Y and Z axes separately in a stepwise sequential order to arrive at the orientation we want.



Which is 45, 60, 45 degrees in the XYZ convention (which means we rotate the axes in that order). Then i can simple rotate the X-axis by 45, Y-axis by 60, and the Z-axis by 45

(Keep in mind the frame of reference or the position and direction of the camera through which i am viewing the object remains the same even in thee next few pictures)

Now, notice what happens when I change the order of the rotation of our axes. . . .say i chose the ZYX convention. Then i rotate the Z-axis by 45, the Y-axis by 60 and the X-axis by 45.

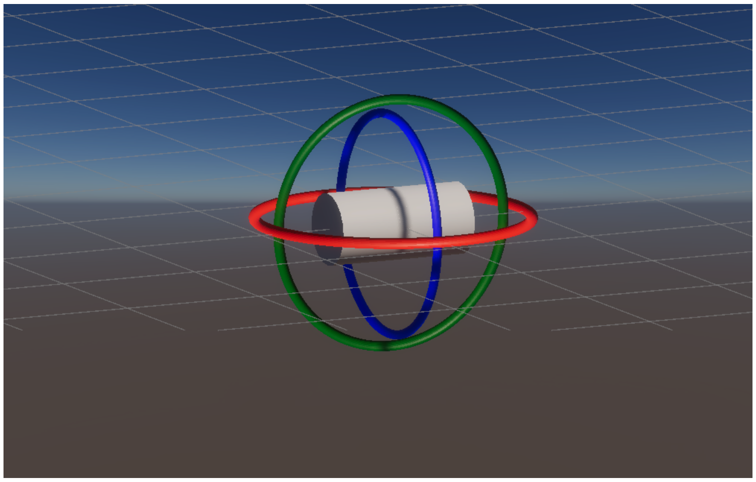


THE ORIENTATION CHANGES!!!! Which means when we are rotating using euler angles, the ORDER of rotation matters!! Also the common convention for Euler angles is the ZYX convention. This is also called yaw, pitch and roll. Where yaw is rotating about the Z-axis, pitch is rotating about the Y-axis and roll is rotating about the X-axis.

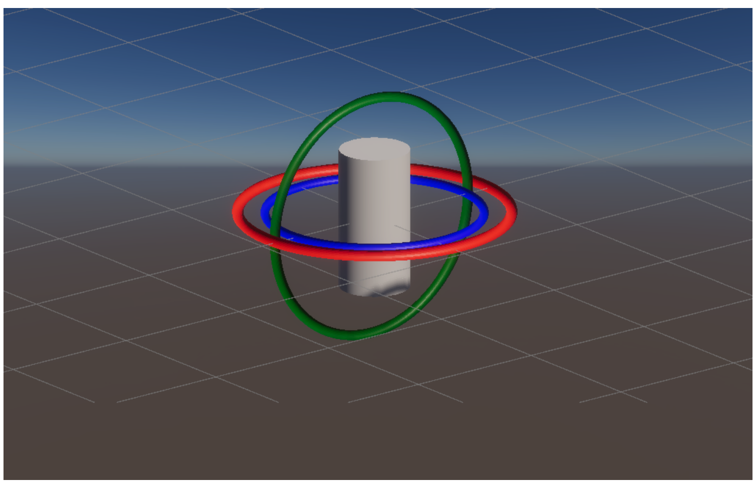
Note: Since orders matter when dealing with euler rotations, matrices are usually used to deal with them.

Now that we have all our terms defined, we can finally move on to the Gimbal Lock.

2 Understanding the Gimbal Lock



The above diagram displays a cylinder with it's axes of rotation. Notice what happens when we rotate the green axis by 90 degrees.



The blue axis and the red axis align!! This implies that we completely lose an axis of rotation as both the blue and the red axis rotate in the same way. At first glance it does not seem like a big issue, but when dealing with complex rotation it can result in messy results and orientations. This is known as the gimbal lock.

Now, why quaternions? Although quaternions are basically 4D vectors, they actually make things like rotation a piece of cake. When rotating using euler angles, we have 3 operations/steps. So in order to model complicated rotations, things can get messy. And we have to watch out for the gimbal lock as well. With quaternions on the other hand, rotation is just one operation. And it avoids the gimbal lock as well.

Now we could try to visualize quaternions and deal with them intuitively, but I think its easier to get a better understanding of quaternions by just learning the math behind them and understanding why a certain operation on a quaternion results in a pure rotation. We finally move on to quaternions.

Quaternions are essentially 4d vectors with special multiplication rules. Just like how we have our usual complex numbers which are of the form a + bi, we have quaternions in the form q = a + bi + cj + dk where a,b,c,d are real numbers. The multiplication rules will be discussed in the next section.

3 Introduction to Quaternions

3.1 Objective

to try and get a decent intuition behind what quaternions really are and learn their algebra and apply them to composing rotations in a 3D space.

3.2 Definitions and Algebra

A quaternion is

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

with $i^2 = j^2 = k^2 = -1$ and $ij = k$, $jk = i$, $ki = j$ (non-commutative multiplication). Reversing the order would instead result in the negative term.

3.3 Extended multiplication (coordinate-wise)

For $p = (a, b, c, d)$ and $q = (w, x, y, z)$,

$$\begin{aligned} pq = &(aw - bx - cy - dz) \\ &+ (ax + bw + cz - dy)\, i \\ &+ (ay - bz + cw + dx)\, j \\ &+ (az + by - cx + dw)\, k. \end{aligned}$$

3.4 Scalar-vector form

To gain a better understanding of quaternions, we introduce the scalar-vector form. Not only does this form give us a better intuition of what quaternions are, but they also ease computation by a large extent. Write $q = a + \mathbf{u}$ with $\mathbf{u} = (b, c, d) \in \mathbb{R}^3$. Then

$$(a + \mathbf{u})(w + \mathbf{v}) = (aw - \mathbf{u} \cdot \mathbf{v}) + (a\mathbf{v} + w\mathbf{u} + \mathbf{u} \times \mathbf{v}).$$

This identity can be proved easily using the quaternion multiplication rules (the proof idea is given in the next subsection). Notice that $\mathbf{u}$ is essentially just a 3D vector.

Proof of the scalar-vector identity (sketch)

Expanding $(a + \mathbf{u})(w + \mathbf{v})$ and using $\mathbf{uv} = -\mathbf{u} \cdot \mathbf{v} + \mathbf{u} \times \mathbf{v}$ yields the identity. we obtain this fact using the quaternion multiplication rules.

3.5 Conjugate, Norm and Inverse

Conjugate $q^* = a - \mathbf{u}$ (similar to complex numbers),

Now we simply chose to multiply a quaternion like its conjugate $q \cdot q* = (a + \mathbf{u})(w - \mathbf{u}) = (a^2 + \mathbf{u} \cdot \mathbf{u})$

Recall that $\mathbf{u} \cdot \mathbf{u}$ is the squared length of the vector $\mathbf{u}$.

This implies that $q \cdot q* = a^2 + \|\mathbf{u}\|^2 = a^2 + x^2 + y^2 + z^2$ which is nothing but the euclidean length of the quaternion squared!! which means $\|q\|^2 = q \cdot q*$

3.6 A few more common results

the next few formulas, identities, and properties which are extremely simple to prove. For those who have a more intricate proof, a proof sketch will be given.

Inverse: $q^{-1} = q^{*}/|q|^2$.

Quaternion conjugation: $(pq)* = q * p*$ (this is a really beautiful property of quaternions displaying anti commutativity)

Quaternion length: $\|pq\| = \|p\|\|q\|$

Unit Quaternion: $\|q\| = 1, q* = q^-1$

Scalar Quaternion: $q = a + 0, q = q*$

Vector Quaternion: $q = 0 + \mathbf{u}, q* = -q$

Quaternion Scalar Part: $a = 1/2(q + q*)$

Quaternion Vector Part: $\mathbf{u} = 1/2(q - q*)$

Quaternion Dot Product Via Conjugation: $p \cdot q = 1/2(pq * +qp*)$ (we define pq as the normal quaternion multiplication between two quaternions defined by the quaternion multiplication rules. And we define the dot product of quaternions as the dot product of 4 dimensional vectors)

Vector Quaternion Dot Product: $\mathbf{u} \cdot \mathbf{v} = 1/2(uv * +vu*) = -1/2(uv + vu)$ (where u and v are pure vector quaternions. not 3D vectors)

3D Vector cross product: $\mathbf{u} \times \mathbf{v} = 1/2(vu - uv)$

proof. we know that $uv = -(u \cdot v) + u \times v$ (given in 3.3) and we also know how to extract the vector part. so we proceed to extract the vector uv hence,

$$u \times v = 1/2(uv - (uv)* = 1/2(uv - v * u*) = 1/2(uv - vu)$$

And now finally!! we can move on to Rotations......

4 Rotations

4.1 Just a random operation for now

let q be a quaternion and let $\mathbf{u}$ be a purely vector quaternion (then $\mathbf{u}* = -\mathbf{u}$). now we define an operation on the vector quaternion $\mathbf{u}$ such that $\mathbf{u}$ is mapped to $q\mathbf{u}q*$. Now let this be equal to $\mathbf{v}$.Then,

$$\mathbf{v}* = (q\mathbf{u}q*)* = (q*) * \mathbf{u} * q* = q\mathbf{u} * q* = -q\mathbf{u}q* = -\mathbf{v}$$

which implies that $\mathbf{v}* = -\mathbf{v}$. That means that $\mathbf{v}$ is also a purely vector quaternion quaternion

now let us take a look at its length.

$$\|\mathbf{v}\| = \|q\|\|u\|\|q*\|| = \|q\|^2\|u\| = \|u\|$$

which means that the length is preserved if q is a unit quaternion! This means that this operation is not a transformation.

Now with a little more long quaternion algebra and manipulation we find out that dot products are preserved at well. This means....

$$p\mathbf{v}p \cdot q\mathbf{u}q = \mathbf{v} \cdot \mathbf{u}$$

So this operation preserves lengths and angles.. that seems like a pure rotation doesn't it? Wait so does a reflection; they preserve dot products and lengths but reverse the orientation of the space. But....with a little more manipulation, we get,

$$p\mathbf{i}p \times p\mathbf{j}p = p\mathbf{k}p$$

This means that the orientation is not reversed either. If it were to be reversed, then we would get $p{-}\mathbf{k}p = -p\mathbf{k}p$. And now we know that our operation is a pure rotation! But its not just any rotation, it is the rotation of the 3D space of vector quaternions.

4.2 Composing rotations

Say we have two unit quaternions p and q, and we want to apply the rotation by q, and then the rotation by p, to some vector v (we will soon try to intuitively understand what rotation by the unit quaternion p or q exactly means, for now you can just think of them as defining the axis of rotation). This is relatively straightforward:

$$p(q\mathbf{v}q^*)p^* = (pq)\mathbf{v}(q^*p^*) = (pq)\mathbf{v}(pq)^*$$

So, the composition of rotations is the same as the rotation by the product quaternion pq! This is where the true power of quaternions lies: when combining many rotations, we can still represent the result as a quaternion, and we know the simple and explicit formula for computing the result (i.e. quaternion multiplication).

4.3 Constructing Specific Rotations

We now know that a certain operation on a purely vector quaternion results in a rotation, but what really is this rotation, we know how to rotate by some unit quaternion p but what do we know about that unit quaternion? nothing really. Naturally, the next question would be, how can we construct a specific rotation? how can we rotate by a certain θ of our choice? and how can we chose the axis of our choice around which to rotate.

So to better understand this, we will try to construct a rotation in the XY plane **only**. To be able to do this we need to rotate around the Z or **k** axis. Which means our unit rotation quaternion p should be of the form

$$p = a + b\mathbf{k}$$

And since p is a unit quaternion, we have the condition that

$$\|a\|^2 + \|b\|^2 = 1$$

Now we want to rotate the XY plane by this quaternion p. So let's see what is does to the X (i.e to the **i** quaternion)axis first..

$$p\mathbf{i}p = (a + b\mathbf{k})\mathbf{i}(a - b\mathbf{k}) = (a\mathbf{i} + b\mathbf{ki})(a - b\mathbf{k}) = (a\mathbf{i} + b\mathbf{j})(a - b\mathbf{k}) = (a^2 - b^2)\mathbf{i} + 2ab\mathbf{j}$$