

Understanding Quaternions

By Jeremiah van Oosten.

In this article I will attempt to explain the concept of Quaternions in an easy to understand way. I will explain how you might visualize a Quaternion as well as explain the different operations that can be applied to quaternions. I will also compare applications of matrices, euler angles, and quaternions and try to explain when you would want to use quaternions instead of Euler angles or matrices and when you would not.

DISCLAIMER: You cannot fully understand quaternions in just 45 minutes.

This article is extremely math intensive and is not intended for the weak-hearted.

Contents

1 Introduction

2 Complex Numbers

- 2.1 Adding and Subtracting Complex Numbers 2
- 2.2 Multiply a Complex Number by a Scalar 2
- 2.3 Product of Complex Numbers 2
- 2.4 Square of Complex Numbers 2
- 2.5 Complex Conjugate 3
- 2.6 Absolute Value of a Complex Number 3
- 2.7 Quotient of Two Complex Numbers 3

3 Powers of i

4 The Complex Plane

- 4.1 Rotors 4

5 Quaternions

- 5.1 Quaternions as an Ordered Pair 5
- 5.2 Adding and Subtracting Quaternions 5
- 5.3 Quaternion Products 5
- 5.4 A Real Quaternion 6
- 5.5 Multiplying a Quaternion by a Scalar 6
- 5.6 Pure Quaternions 6
- 5.7 Additive Form of a Quaternion 6
- 5.8 Unit Quaternion 7
- 5.9 Binary Form of a Quaternion 7
- 5.10 Quaternion Conjugate 7
- 5.11 Quaternion Norm 7
- 5.12 Quaternion Normalization 7
- 5.13 Quaternion Inverse 7
- 5.14 Quaternion Dot Product 8

6 Rotations

7 Quaternion Interpolation

- 7.1 SLERP 10
 - 7.1.1 Quaternion Difference 10
 - 7.1.2 Quaternion Exponentiation 11
 - 7.1.3 Fractional Difference of Quaternions 11
 - 7.1.4 Considerations 11
- 7.2 SQUAD 12

8 Conclusion

12

9 Download the Demo

12

10 Reference

13

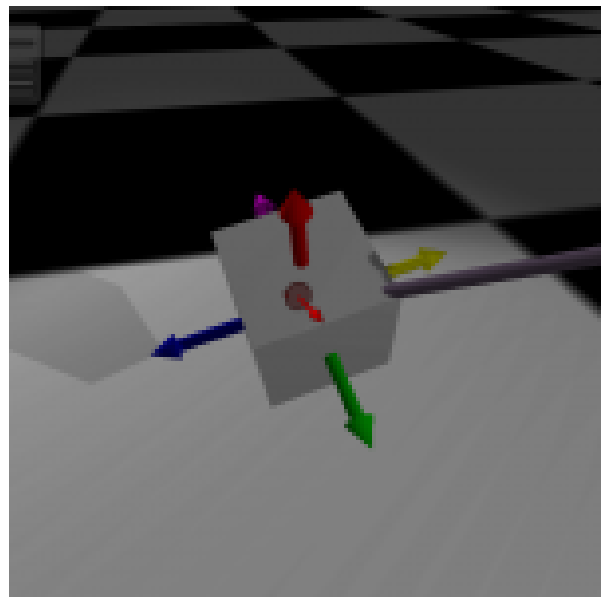


Figure 1: Understanding Quaternions [\[Image Source\]](#)

1 Introduction

In computer graphics, we use transformation matrices to express a position in space (translation) as well as its orientation in space (rotation). Optionally, a single transformation matrix can also be used to express the scale or “shear” of an object. We can think of this transformation matrix as a “basis space” where if you multiply a vector or a point (or even another matrix) by a transformation matrix you “transform” that vector, point or matrix into the space represented by that matrix.

In this article, I will not discuss the details of transformation matrices. For a detailed description of transformation matrices, you can refer to my previous article titled [Matrices](#).

In this article, I want to discuss an alternative method of describing the orientation of an object (rotation) in

space using quaternions.

The concept of quaternions was realized by the Irish mathematician [Sir William Rowan Hamilton](#) on Monday October 16th 1843 in Dublin, Ireland. Hamilton was on his way to the [Royal Irish Academy](#) with his wife and as he was passing over the [Royal Canal](#) on the [Brougham Bridge](#) he made a dramatic realization that he immediately carved into the stone of the bridge.

$$i^2 = j^2 = k^2 = ijk = -1 \quad (1)$$

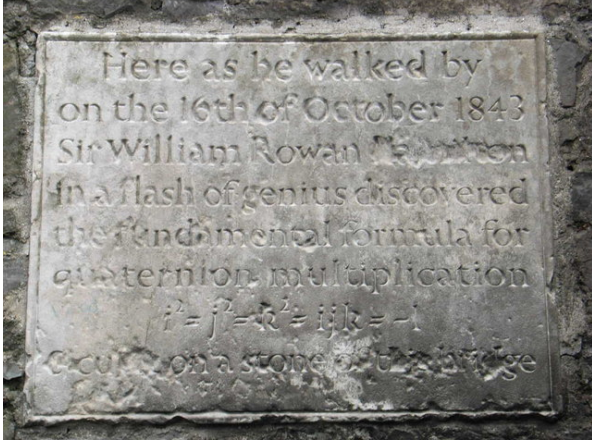


Figure 2: William Rowan Hamilton Plaque on Broome Bridge on the Royal Canal commemorating his discovery of the fundamental formula for quaternion multiplication. [\[Image Source\]](#)

2 Complex Numbers

Before we can fully understand quaternions, we must first understand where they came from. The root of quaternions is based on the concept of the complex number system.

In addition to the well-known number sets ([Natural](#), [Integer](#), [Real](#), and [Rational](#)), the Complex Number system introduces a new set of numbers called imaginary numbers. Imaginary numbers were invented to solve certain equations that had no solutions such as:

$$x^2 + 1 = 0 \quad (2)$$

To solve this expression, we must state that $x^2 = -1$ which we know is not possible because the square of any number (positive or negative) is always positive. Mathematicians generally can't accept that an expression does not have a solution so a new term was invented called the [imaginary number](#) that can be used to solve such equations.

The imaginary number has the form:

$$i^2 = -1 \quad (3)$$

Don't try to actually understand this term as there is no logical reason why it exists. We just have to accept that i is just something that squares to -1 .

The set of imaginary numbers can be represented by \mathbb{I} .

The set of complex numbers (represented by the symbol \mathbb{C}) is the sum of a real number and an imaginary number and has the form:

$$z = a + bi \quad a, b \in \mathbb{R}, \quad i^2 = -1 \quad (4)$$

It could also be stated that all Real numbers are complex numbers with $b = 0$ and all imaginary numbers are complex numbers with $a = 0$.

2.1 Adding and Subtracting Complex Numbers

Complex numbers can be added and subtracted by adding or subtracting the real, and imaginary parts. Addition:

$$(a_1 + b_1i) + (a_2 + b_2i) = (a_1 + a_2) + (b_1 + b_2)i \quad (5)$$

Subtraction:

$$(a_1 + b_1i) - (a_2 + b_2i) = (a_1 - a_2) + (b_1 - b_2)i \quad (6)$$

2.2 Multiply a Complex Number by a Scalar

A complex number is multiplied by a scalar by multiplying each term of the complex number by the scalar:

$$\lambda(a + bi) = \lambda a + \lambda bi \quad (7)$$

2.3 Product of Complex Numbers

Complex numbers can also be multiplied by applying normal algebraic rules.

$$\begin{aligned} z_1 &= (a_1 + b_1i) \\ z_2 &= (a_2 + b_2i) \\ z_1 z_2 &= (a_1 + b_1i)(a_2 + b_2i) \\ &= a_1 a_2 + a_1 b_2 i + b_1 a_2 i + b_1 b_2 i^2 \\ &= (a_1 a_2 - b_1 b_2) + (a_1 b_2 + b_1 a_2) i \end{aligned} \quad (8)$$

2.4 Square of Complex Numbers

A complex number can also be squared by multiplying by itself:

$$\begin{aligned} z &= (a + bi) \\ z^2 &= (a + bi)(a + bi) \\ &= (a^2 - b^2) + 2abi \end{aligned} \quad (9)$$

2.5 Complex Conjugate

The **conjugate** of a complex number is a complex number with the imaginary part negated and is denoted as either \bar{z} or z^* .

$$\begin{aligned} z &= (a + bi) \\ z^* &= (a - bi) \end{aligned} \quad (10)$$

The product of a complex number and its conjugate gives a special result.

$$\begin{aligned} z &= (a + bi) \\ z^* &= (a - bi) \\ zz^* &= (a + bi)(a - bi) \\ &= a^2 - abi + abi + b^2 \\ &= a^2 + b^2 \end{aligned} \quad (11)$$

2.6 Absolute Value of a Complex Number

We can use the **conjugate** of a complex number to compute the absolute value (or **norm**, or **magnitude**) of a complex number. The absolute value of a complex number is the square-root of the complex number multiplied by its **conjugate** and is denoted $|z|$:

$$\begin{aligned} z &= (a + bi) \\ |z| &= \sqrt{zz^*} \\ &= \sqrt{(a + bi)(a - bi)} \\ &= \sqrt{a^2 + b^2} \end{aligned} \quad (12)$$

2.7 Quotient of Two Complex Numbers

To compute the quotient of two complex numbers, we multiply the numerator and denominator by the complex conjugate of the denominator.

$$\begin{aligned} z_1 &= (a_1 + b_1i) \\ z_2 &= (a_2 + b_2i) \\ \frac{z_1}{z_2} &= \frac{a_1 + b_1i}{a_2 + b_2i} \\ &= \frac{(a_1 + b_1i)(a_2 - b_2i)}{(a_2 + b_2i)(a_2 - b_2i)} \\ &= \frac{a_1a_2 - a_1b_2i + b_1a_2i - b_1b_2i^2}{a_2^2 + b_2^2} \\ &= \frac{a_1a_2 + b_1b_2}{a_2^2 + b_2^2} + \frac{b_1a_2 - a_1b_2}{a_2^2 + b_2^2}i \end{aligned} \quad (13)$$

3 Powers of i

If we state that $i^2 = -1$ then it should be possible to raise i to other powers as well.

$$\begin{aligned} i^0 &= 1 \\ i^1 &= i \\ i^2 &= -1 \\ i^3 &= ii^2 = -i \\ i^4 &= i^2i^2 = 1 \\ i^5 &= ii^4 = i \\ i^6 &= ii^5 = i^2 = -1 \end{aligned} \quad (14)$$

If we keep writing this sequence, we will see a pattern emerge $(1, i, -1, -i, 1, \dots)$.

A similar pattern emerges from the increasing negative powers.

$$\begin{aligned} i^0 &= 1 \\ i^{-1} &= -i \\ i^{-2} &= -1 \\ i^{-3} &= i \\ i^{-4} &= 1 \\ i^{-5} &= -i \\ i^{-6} &= -1 \end{aligned} \quad (15)$$

You may have seen a similar pattern in mathematics before but in the form $(x, y, -x, -y, x, \dots)$ which is generated by rotating a point 90 degrees counter-clockwise on a 2D Cartesian plane and the sequence $(x, -y, -x, y, x, \dots)$ is generated by rotating a point 90 degrees clockwise on a 2D Cartesian plane.

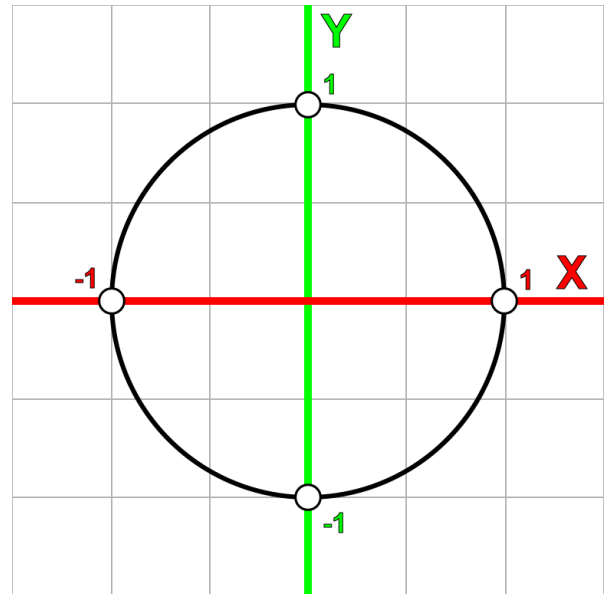


Figure 3: Cartesian Plane [\[Image Source\]](#)

4 The Complex Plane

We can also map complex numbers in a 2D grid called the **Complex Plane** by mapping the **Real** part on the horizontal axis and the **Imaginary** part on the vertical axis.

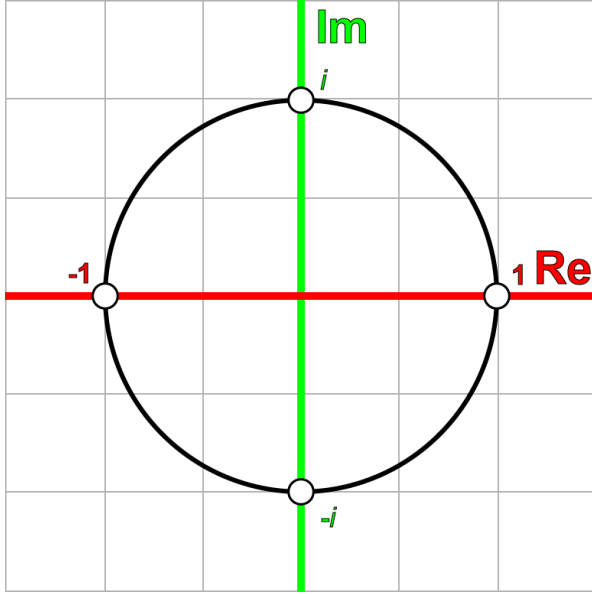


Figure 4: Complex Plane [Image Source]

As shown in the previous sequence, we can say that if we multiply a complex number by i , we can rotate the complex number through the complex plane at 90 degree increments.

Let's see if this is true. We'll take an arbitrary point \mathbf{p} in the complex plane:

$$p = 2 + i \quad (16)$$

and we multiply it by i gives \mathbf{q} :

$$\begin{aligned} p &= 2 + i \\ q &= pi \\ &= (2 + i)i \\ &= 2i + i^2 \\ &= -1 + 2i \end{aligned} \quad (17)$$

Multiplying \mathbf{q} by i gives \mathbf{r} :

$$\begin{aligned} q &= -1 + 2i \\ r &= qi \\ &= (-1 + 2i)i \\ &= -i + 2i^2 \\ &= -2 - i \end{aligned} \quad (18)$$

And multiplying \mathbf{r} by i gives \mathbf{s} :

$$\begin{aligned} r &= -2 - i \\ s &= ri \\ &= (-2 - i)i \\ &= -2i - i^2 \\ &= 1 - 2i \end{aligned} \quad (19)$$

And multiplying \mathbf{s} by i gives \mathbf{t} :

$$\begin{aligned} s &= 1 - 2i \\ t &= si \\ &= (1 - 2i)i \\ &= i - 2i^2 \\ &= 2 + i \end{aligned} \quad (20)$$

Which is exactly what we started with (\mathbf{p}). If we plot these complex numbers on the complex plane, we get the following result.

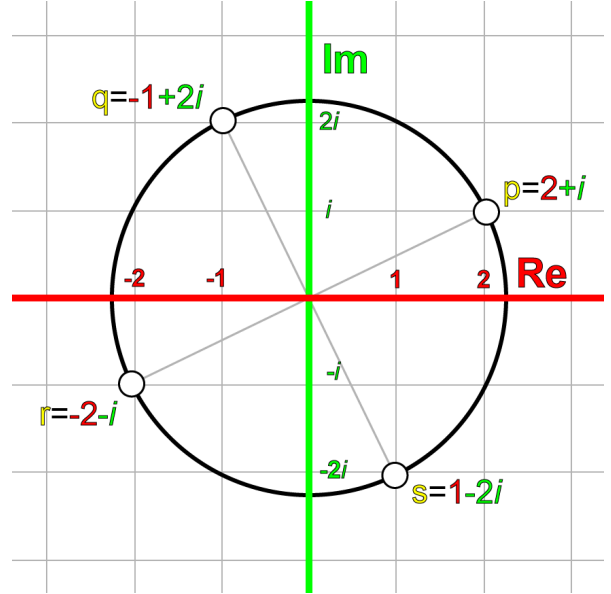


Figure 5: Complex Numbers on the Complex Plane [Image Source]

We can also rotate clock-wise in the complex plane by multiplying the complex number by $-i$.

4.1 Rotors

We can also perform arbitrary rotations in the complex plane by defining a complex number of the form:

$$q = \cos \theta + i \sin \theta \quad (21)$$

Multiplying any complex number by the rotor \mathbf{q} produces the general formula:

$$\begin{aligned} p &= a + bi \\ q &= \cos \theta + i \sin \theta \\ pq &= (a + bi)(\cos \theta + i \sin \theta) \\ a' + b'i &= a \cos \theta - b \sin \theta + (a \sin \theta + b \cos \theta)i \end{aligned} \quad (22)$$

Which can also be written in matrix form:

$$\begin{bmatrix} a' & -b' \\ b' & a' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \quad (23)$$

Which is the method to rotate an arbitrary point in the complex plane counter-clockwise about the origin.

5 Quaternions

With this knowledge of the complex number system and the complex plane, we can extend this to 3-dimensional space by adding two imaginary numbers to our number system in addition to i .

The general form to express quaternions is

$$q = s + xi + yj + zk \quad s, x, y, z \in \mathbb{R} \quad (24)$$

Were, according to Hamilton's famous expression:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (25)$$

and

$$\begin{aligned} ij &= k & jk &= i & ki &= j \\ ji &= -k & kj &= -i & ik &= -j \end{aligned} \quad (26)$$

You may have noticed that the relationship between i , j , and k are very similar to the cross product rules for the unit cartesian vectors:

$$\begin{aligned} \mathbf{x} \times \mathbf{y} &= \mathbf{z} & \mathbf{y} \times \mathbf{z} &= \mathbf{x} & \mathbf{z} \times \mathbf{x} &= \mathbf{y} \\ \mathbf{y} \times \mathbf{x} &= -\mathbf{z} & \mathbf{z} \times \mathbf{y} &= -\mathbf{x} & \mathbf{x} \times \mathbf{z} &= -\mathbf{y} \end{aligned} \quad (27)$$

Hamilton also recognized that the i , j , and k imaginary numbers could be used to represent three cartesian unit vectors \mathbf{i} , \mathbf{j} , and \mathbf{k} with the same properties of imaginary numbers, such that $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$.

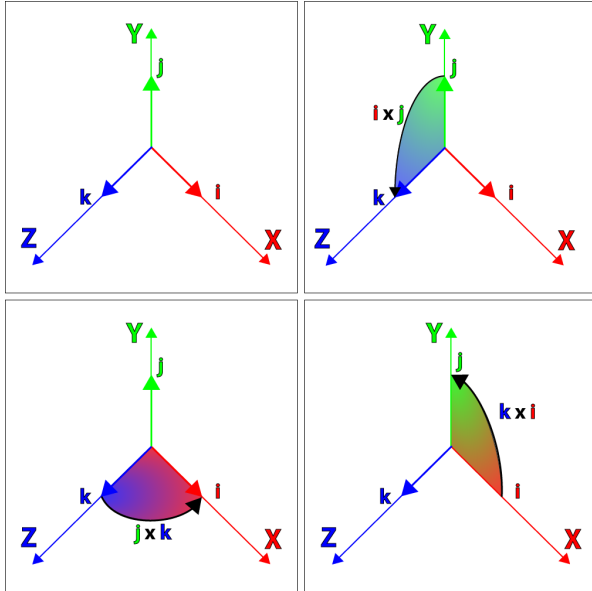


Figure 6: Visualizing the Properties of ij , jk , ki [\[Image Source\]](#)

The image above visualizes the relationship between the cartesian unit vectors represented by \mathbf{i} , \mathbf{j} , and \mathbf{k} .

5.1 Quaternions as an Ordered Pair

We can also represent quaternions as an ordered pair:

$$q = [s, \mathbf{v}] \quad s \in \mathbb{R}, \mathbf{v} \in \mathbb{R}^3 \quad (28)$$

Where \mathbf{v} can also be represented by its individual components:

$$q = [s, x\mathbf{i} + y\mathbf{j} + z\mathbf{k}] \quad s, x, y, z \in \mathbb{R} \quad (29)$$

Using this notation, we can more easily show the similarities between quaternions and complex numbers.

5.2 Adding and Subtracting Quaternions

Quaternions can be added and subtracted similar to complex numbers:

$$\begin{aligned} q_a &= [s_a, \mathbf{a}] \\ q_b &= [s_b, \mathbf{b}] \\ q_a + q_b &= [s_a + s_b, \mathbf{a} + \mathbf{b}] \\ q_a - q_b &= [s_a - s_b, \mathbf{a} - \mathbf{b}] \end{aligned} \quad (30)$$

5.3 Quaternion Products

We can also express the product of two quaternions:

$$\begin{aligned} q_a &= [s_a, \mathbf{a}] \\ q_b &= [s_b, \mathbf{b}] \\ q_a q_b &= [s_a, \mathbf{a}][s_b, \mathbf{b}] \\ &= (s_a + x_a i + y_a j + z_a k)(s_b + x_b i + y_b j + z_b k) \\ &= (s_a s_b - x_a x_b - y_a y_b - z_a z_b) \\ &\quad + (s_a x_b + s_b x_a + y_a z_b - y_b z_a) i \\ &\quad + (s_a y_b + s_b y_a + z_a x_b - z_b x_a) j \\ &\quad + (s_a z_b + s_b z_a + x_a y_b - x_b y_a) k \end{aligned} \quad (31)$$

Which results in another quaternion. If we replace the imaginary numbers i , j , and k in the previous expression by the ordered pairs (also known as the quaternion units),

$$i = [0, \mathbf{i}] \quad j = [0, \mathbf{j}] \quad k = [0, \mathbf{k}] \quad (32)$$

And substituting back to the original expression together with $[1, \mathbf{0}] = 1$ gives:

$$\begin{aligned} [s_a, \mathbf{a}][s_b, \mathbf{b}] &= (s_a s_b - x_a x_b - y_a y_b - z_a z_b)[1, \mathbf{0}] \\ &\quad + (s_a x_b + s_b x_a + y_a z_b - y_b z_a)[0, \mathbf{i}] \\ &\quad + (s_a y_b + s_b y_a + z_a x_b - z_b x_a)[0, \mathbf{j}] \\ &\quad + (s_a z_b + s_b z_a + x_a y_b - x_b y_a)[0, \mathbf{k}] \end{aligned} \quad (33)$$

And expanding this expression into a sum of ordered pairs gives:

$$\begin{aligned}
[s_a, \mathbf{a}][s_b, \mathbf{b}] &= [s_a s_b - x_a x_b - y_a y_b - z_a z_b, \mathbf{0}] \\
&\quad + [0, (s_a x_b + s_b x_a + y_a z_b - y_b z_a) \mathbf{i}] \\
&\quad + [0, (s_a y_b + s_b y_a + z_a x_b - z_b x_a) \mathbf{j}] \\
&\quad + [0, (s_a z_b + s_b z_a + x_a y_b - x_b y_a) \mathbf{k}]
\end{aligned} \tag{34}$$

If we multiply through with the quaternion unit and extract the common vector components, we can rewrite this equation in this way:

$$\begin{aligned}
[s_a, \mathbf{a}][s_b, \mathbf{b}] &= [s_a s_b - x_a x_b - y_a y_b - z_a z_b, \mathbf{0}] \\
&\quad + [0, s_a(x_b \mathbf{i} + y_b \mathbf{j} + z_b \mathbf{k}) \\
&\quad + s_b(x_a \mathbf{i} + y_a \mathbf{j} + z_a \mathbf{k}) \\
&\quad + (y_a z_b - y_b z_a) \mathbf{i} + (z_a x_b - z_b x_a) \mathbf{j} \\
&\quad + (x_a y_b - x_b y_a) \mathbf{k}]
\end{aligned} \tag{35}$$

This equation gives us the sum of two ordered pairs. The first ordered pair is a **Real** quaternion and the second is a **Pure** quaternion. These two ordered pairs can be combined into a single ordered pair:

$$\begin{aligned}
[s_a, \mathbf{a}][s_b, \mathbf{b}] &= [s_a s_b - x_a x_b - y_a y_b - z_a z_b, \\
&\quad s_a(x_b \mathbf{i} + y_b \mathbf{j} + z_b \mathbf{k}) \\
&\quad + s_b(x_a \mathbf{i} + y_a \mathbf{j} + z_a \mathbf{k}) \\
&\quad + (y_a z_b - y_b z_a) \mathbf{i} + (z_a x_b - z_b x_a) \mathbf{j} \\
&\quad + (x_a y_b - x_b y_a) \mathbf{k}]
\end{aligned} \tag{36}$$

And if we substitute,

$$\begin{aligned}
\mathbf{a} &= x_a \mathbf{i} + y_a \mathbf{j} + z_a \mathbf{k} \\
\mathbf{b} &= x_b \mathbf{i} + y_b \mathbf{j} + z_b \mathbf{k} \\
\mathbf{a} \cdot \mathbf{b} &= x_a x_b + y_a y_b + z_a z_b \\
\mathbf{a} \times \mathbf{b} &= (y_a z_b - y_b z_a) \mathbf{i} + (z_a x_b - z_b x_a) \mathbf{j} \\
&\quad + (x_a y_b - x_b y_a) \mathbf{k}
\end{aligned} \tag{37}$$

We get:

$$[s_a, \mathbf{a}][s_b, \mathbf{b}] = [s_a s_b - \mathbf{a} \cdot \mathbf{b}, s_a \mathbf{b} + s_b \mathbf{a} + \mathbf{a} \times \mathbf{b}] \tag{38}$$

Which is the general equation of a quaternion product.

5.4 A Real Quaternion

A **Real** Quaternion is a quaternion with a vector term of $\mathbf{0}$:

$$q = [s, \mathbf{0}] \tag{39}$$

And the product of two **Real** Quaternions is another **Real** Quaternion:

$$\begin{aligned}
q_a &= [s_a, \mathbf{0}] \\
q_b &= [s_b, \mathbf{0}] \\
q_a q_b &= [s_a, \mathbf{0}][s_b, \mathbf{0}] \\
&= [s_a s_b, \mathbf{0}]
\end{aligned} \tag{40}$$

Which is similar to the product of two complex numbers that contain a zero imaginary term.

$$\begin{aligned}
z_1 &= a_1 + 0i \\
z_2 &= a_2 + 0i \\
z_1 z_2 &= (a_1 + 0i)(a_2 + 0i) \\
&= a_1 a_2
\end{aligned} \tag{41}$$

5.5 Multiplying a Quaternion by a Scalar

We can also multiply a quaternion by a scalar which should obey the rule:

$$\begin{aligned}
q &= [s, \mathbf{v}] \\
\lambda q &= \lambda[s, \mathbf{v}] \\
&= [\lambda s, \lambda \mathbf{v}]
\end{aligned} \tag{42}$$

We can confirm this by using the product or **Real** Quaternions shown above to multiply a quaternion by the scalar as a **Real** Quaternion:

$$\begin{aligned}
q &= [s, \mathbf{v}] \\
\lambda &= [\lambda, \mathbf{0}] \\
\lambda q &= [\lambda, \mathbf{0}][s, \mathbf{v}] \\
&= [\lambda s, \lambda \mathbf{v}]
\end{aligned} \tag{43}$$

5.6 Pure Quaternions

Similar to **Real** Quaternions, Hamilton also defined the **Pure** Quaternion as a quaternion that has a zero scalar term:

$$q = [0, \mathbf{v}] \tag{44}$$

Or, written in its component parts:

$$q = xi + yj + zk \tag{45}$$

And we can also take the product of two **Pure** quaternions:

$$\begin{aligned}
q_a &= [0, \mathbf{a}] \\
q_b &= [0, \mathbf{b}] \\
q_a q_b &= [0, \mathbf{a}][0, \mathbf{b}] \\
&= [-\mathbf{a} \cdot \mathbf{b}, \mathbf{a} \times \mathbf{b}]
\end{aligned} \tag{46}$$

According to the quaternion product rule shown above.

5.7 Additive Form of a Quaternion

We can also express quaternions as an addition of the **Real** and **Pure** quaternion parts:

$$\begin{aligned}
q &= [s, \mathbf{v}] \\
&= [s, \mathbf{0}] + [0, \mathbf{v}]
\end{aligned} \tag{47}$$

5.8 Unit Quaternion

Given an arbitrary vector \mathbf{v} , we can express this vector in both its scalar magnitude and its direction as such:

$$\mathbf{v} = v\hat{\mathbf{v}} \text{ where } v = |\mathbf{v}|, \text{ and } |\hat{\mathbf{v}}| = 1 \quad (48)$$

And combining this definition with the definition of a pure quaternion gives:

$$\begin{aligned} q &= [0, \mathbf{v}] \\ &= [0, v\hat{\mathbf{v}}] \\ &= v[0, \hat{\mathbf{v}}] \end{aligned} \quad (49)$$

And we can also describe a unit quaternion that has a zero scalar and a unit vector as such:

$$\hat{q} = [0, \hat{\mathbf{v}}] \quad (50)$$

5.9 Binary Form of a Quaternion

We can now combine the definitions of the unit quaternion and the additive form of a quaternion, we can create a representation of quaternions which is similar to the notation used to describe complex numbers:

$$\begin{aligned} q &= [s, \mathbf{v}] \\ &= [s, \mathbf{0}] + [0, \mathbf{v}] \\ &= [s, \mathbf{0}] + v[0, \hat{\mathbf{v}}] \\ &= s + v\hat{q} \end{aligned} \quad (51)$$

This gives us a way to represent the quaternion that is very similar to complex numbers:

$$\begin{aligned} z &= a + bi \\ q &= s + v\hat{q} \end{aligned} \quad (52)$$

5.10 Quaternion Conjugate

The quaternion conjugate can be computed by negating the vector part of the quaternion:

$$\begin{aligned} q &= [s, \mathbf{v}] \\ q^* &= [s, -\mathbf{v}] \end{aligned} \quad (53)$$

And the product of a quaternion with its conjugate gives:

$$\begin{aligned} qq^* &= [s, \mathbf{v}][s, -\mathbf{v}] \\ &= [s^2 - \mathbf{v} \cdot -\mathbf{v}, -s\mathbf{v} + s\mathbf{v} + \mathbf{v} \times -\mathbf{v}] \\ &= [s^2 + \mathbf{v} \cdot \mathbf{v}, \mathbf{0}] \\ &= [s^2 + v^2, \mathbf{0}] \end{aligned} \quad (54)$$

5.11 Quaternion Norm

If you recall from the definition of the norm of a complex number:

$$\begin{aligned} |z| &= \sqrt{a^2 + b^2} \\ zz^* &= |z|^2 \end{aligned} \quad (55)$$

Similarly, the norm (or magnitude) of a quaternion is defined as:

$$\begin{aligned} q &= [s, \mathbf{v}] \\ |q| &= \sqrt{s^2 + v^2} \end{aligned} \quad (56)$$

Which allows us to express the norm of a quaternion as:

$$qq^* = |q|^2 \quad (57)$$

5.12 Quaternion Normalization

With the definition of a quaternion norm, we can use it to normalize a quaternion. A quaternion is normalized by dividing it by $|q|$:

$$q' = \frac{q}{\sqrt{s^2 + v^2}} \quad (58)$$

As an example, let's normalize the quaternion:

$$q = [1, 4\mathbf{i} + 4\mathbf{j} - 4\mathbf{k}] \quad (59)$$

First, we must compute the **norm** of the quaternion:

$$\begin{aligned} |q| &= \sqrt{1^2 + 4^2 + 4^2 + (-4)^2} \\ &= \sqrt{49} \\ &= 7 \end{aligned} \quad (60)$$

Then, we must divide the quaternion by the norm of the quaternion to compute the normalized quaternion:

$$\begin{aligned} q' &= \frac{q}{|q|} \\ &= \frac{(1 + 4\mathbf{i} + 4\mathbf{j} - 4\mathbf{k})}{7} \\ &= \frac{1}{7} + \frac{4}{7}\mathbf{i} + \frac{4}{7}\mathbf{j} - \frac{4}{7}\mathbf{k} \end{aligned} \quad (61)$$

5.13 Quaternion Inverse

The inverse of a quaternion is denoted q^{-1} . To compute the inverse of a quaternion, we take the conjugate of the quaternion and divide it by the square of the norm:

$$q^{-1} = \frac{q^*}{|q|^2} \quad (62)$$

To show this, we can take the fact that by definition of the inverse:

$$qq^{-1} = [1, \mathbf{0}] = 1 \quad (63)$$

And multiply both sides by the conjugate of the quaternion gives:

$$q^*qq^{-1} = q^* \quad (64)$$

And by substitution we get:

$$\begin{aligned} |q|^2 q^{-1} &= q^* \\ q^{-1} &= \frac{q^*}{|q|^2} \end{aligned} \quad (65)$$

And for unit-norm quaternions whose norm is 1, we can write:

$$q^{-1} = q^* \quad (66)$$

5.14 Quaternion Dot Product

Similar to vector dot-products, we can also compute the dot product between two quaternions by multiplying the corresponding scalar parts and summing the results:

$$\begin{aligned} q_1 &= [s_1, x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}] \\ q_2 &= [s_2, x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}] \\ q_1 \cdot q_2 &= s_1s_2 + x_1x_2 + y_1y_2 + z_1z_2 \end{aligned} \quad (67)$$

We can also use the quaternion dot-product to compute the angular difference between the quaternions:

$$\cos \theta = \frac{s_1s_2 + x_1x_2 + y_1y_2 + z_1z_2}{|q_1||q_2|} \quad (68)$$

And for unit-norm quaternions, we can simplify the equation:

$$\cos \theta = s_1s_2 + x_1x_2 + y_1y_2 + z_1z_2 \quad (69)$$

6 Rotations

If you recall we defined a special form of the complex number called a **Rotor** that could be used to rotate a point through the 2D complex plane as:

$$q = \cos \theta + i \sin \theta \quad (70)$$

Then by its similarities to complex numbers, it should be possible to express a quaternion that can be used to rotate a point in 3D-space as such:

$$q = [\cos \theta, \sin \theta \mathbf{v}] \quad (71)$$

Let's test if this theory holds by computing the product of the quaternion q and the vector \mathbf{p} . First, we can express \mathbf{p} as a **Pure** quaternion in the form:

$$p = [0, \mathbf{p}] \quad (72)$$

And q is a unit-norm quaternion in the form:

$$q = [s, \lambda \hat{\mathbf{v}}] \quad (73)$$

Then,

$$\begin{aligned} p' &= qp \\ &= [s, \lambda \hat{\mathbf{v}}][0, \mathbf{p}] \\ &= [-\lambda \hat{\mathbf{v}} \cdot \mathbf{p}, s\mathbf{p} + \lambda \hat{\mathbf{v}} \times \mathbf{p}] \end{aligned} \quad (74)$$

We see that the result is a general quaternion with both a scalar and a vector parts.

Let's first consider the "special" case where \mathbf{p} is perpendicular to $\hat{\mathbf{v}}$ in which case, the dot-product term $-\lambda \hat{\mathbf{v}} \cdot \mathbf{p} = 0$ and the result becomes the **Pure** quaternion:

$$p' = [0, s\mathbf{p} + \lambda \hat{\mathbf{v}} \times \mathbf{p}] \quad (75)$$

In this case, to rotate \mathbf{p} about $\hat{\mathbf{v}}$ we just substitute $s = \cos \theta$ and $\lambda = \sin \theta$.

$$p' = [0, \cos \theta \mathbf{p} + \sin \theta \hat{\mathbf{v}} \times \mathbf{p}] \quad (76)$$

As an example, let's rotate a vector \mathbf{p} 45° about the z-axis then our quaternion \mathbf{q} is:

$$\begin{aligned} q &= [\cos \theta, \sin \theta \mathbf{k}] \\ &= \left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \mathbf{k} \right] \end{aligned} \quad (77)$$

And let's take a vector \mathbf{p} that adheres to the special case that \mathbf{p} is perpendicular to \mathbf{k} :

$$p = [0, 2\mathbf{i}] \quad (78)$$

Now let's find the product of qp :

$$\begin{aligned} p' &= qp \\ &= \left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \mathbf{k} \right] [0, 2\mathbf{i}] \\ &= \left[0, 2\frac{\sqrt{2}}{2}\mathbf{i} + 2\frac{\sqrt{2}}{2}\mathbf{k} \times \mathbf{i} \right] \\ &= [0, \sqrt{2}\mathbf{i} + \sqrt{2}\mathbf{j}] \end{aligned} \quad (79)$$

Which results in a **Pure** quaternion that is rotated 45° about the \mathbf{k} axis.

We can also confirm that the magnitude of the resulting vector is maintained:

$$\begin{aligned} |\mathbf{p}'| &= \sqrt{\sqrt{2}^2 + \sqrt{2}^2} \\ &= 2 \end{aligned} \quad (80)$$

Which is exactly the result we expected!

We can visualize this by the following image:

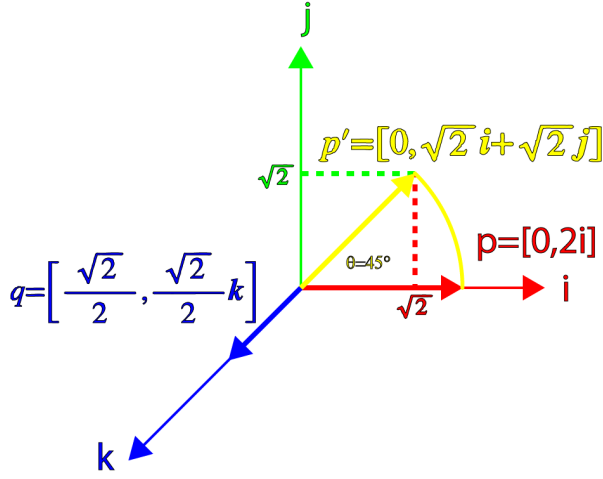


Figure 7: Quaternion Rotation (1) [Image Source]

Now let's consider a quaternion that is not orthogonal to \mathbf{p} . If we specify the vector part of our quaternion to 45° offset from \mathbf{p} we get:

$$\begin{aligned}\hat{\mathbf{v}} &= \frac{\sqrt{2}}{2}\mathbf{i} + \frac{\sqrt{2}}{2}\mathbf{k} \\ \mathbf{p} &= 2\mathbf{i} \\ q &= [\cos \theta, \sin \theta \hat{\mathbf{v}}] \\ p &= [0, \mathbf{p}]\end{aligned}\tag{81}$$

And multiplying our point \mathbf{p} by q we get:

$$\begin{aligned}p' &= qp \\ &= [\cos \theta, \sin \theta \hat{\mathbf{v}}][0, \mathbf{p}] \\ &= [-\sin \theta \hat{\mathbf{v}} \cdot \mathbf{p}, \cos \theta \mathbf{p} + \sin \theta \hat{\mathbf{v}} \times \mathbf{p}]\end{aligned}\tag{82}$$

And substituting $\hat{\mathbf{v}}$, \mathbf{p} and $\theta = 45^\circ$ gives:

$$\begin{aligned}p' &= \left[-\frac{\sqrt{2}}{2} \left(\frac{\sqrt{2}}{2}\mathbf{i} + \frac{\sqrt{2}}{2}\mathbf{k} \right) \cdot (2\mathbf{i}), \right. \\ &\quad \left. \frac{\sqrt{2}}{2}2\mathbf{i} + \frac{\sqrt{2}}{2} \left(\frac{\sqrt{2}}{2}\mathbf{i} + \frac{\sqrt{2}}{2}\mathbf{k} \right) \times 2\mathbf{i} \right] \\ &= [-1, \sqrt{2}\mathbf{i} + \mathbf{j}]\end{aligned}\tag{83}$$

Which is no longer a **pure** quaternion, and it has not been rotated 45° and the vector's norm is no longer 2 (instead it has been reduced to $\sqrt{3}$).

This result can be visualized by the image.

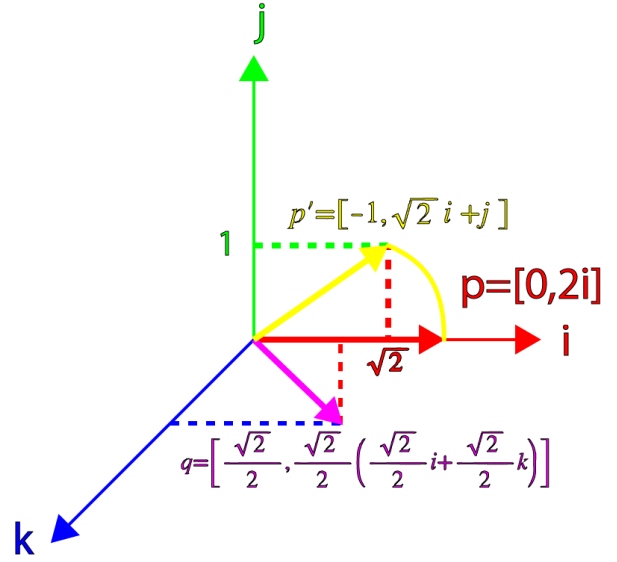


Figure 8: Quaternion Rotation (2) [Image Source]

Technically, it's incorrect to represent the quaternion \mathbf{p}' in 3D space because it's actually a 4D vector! For the sake of simplicity, I will only visualize the vector component of quaternions.

However, all is not lost. Hamilton recognized (but didn't publish) that if we post-multiply the result of qp by the inverse of q then the result is a **pure** quaternion and the norm of the vector component is maintained. Let's see if we can apply this to our example.

First, let's compute q^{-1} :

$$\begin{aligned}q &= \left[\cos \theta, \sin \theta \left(\frac{\sqrt{2}}{2}\mathbf{i} + \frac{\sqrt{2}}{2}\mathbf{k} \right) \right] \\ q^{-1} &= \left[\cos \theta, -\sin \theta \left(\frac{\sqrt{2}}{2}\mathbf{i} + \frac{\sqrt{2}}{2}\mathbf{k} \right) \right]\end{aligned}\tag{84}$$

For $\theta = 45^\circ$ gives:

$$\begin{aligned}q^{-1} &= \left[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \left(\frac{\sqrt{2}}{2}\mathbf{i} + \frac{\sqrt{2}}{2}\mathbf{k} \right) \right] \\ &= \frac{1}{2}[\sqrt{2}, -\mathbf{i} - \mathbf{k}]\end{aligned}\tag{85}$$

And combining the previous value of qp and q^{-1} gives:

$$\begin{aligned}
qp &= [-1, \sqrt{2}\mathbf{i} + \mathbf{j}] \\
qpq^{-1} &= [-1, \sqrt{2}\mathbf{i} + \mathbf{j}] \frac{1}{2} [\sqrt{2}, -\mathbf{i} - \mathbf{k}] \\
&= \frac{1}{2} [-\sqrt{2} - (\sqrt{2}\mathbf{i} + \mathbf{j}) \cdot (-\mathbf{i} - \mathbf{k}), \\
&\quad \mathbf{i} + \mathbf{k} + \sqrt{2}(\sqrt{2}\mathbf{i} + \mathbf{j}) - \mathbf{i} + \sqrt{2}\mathbf{j} + \mathbf{k}] \\
&= \frac{1}{2} [-\sqrt{2} + \sqrt{2}, \\
&\quad \mathbf{i} + \mathbf{k} + 2\mathbf{i} + \sqrt{2}\mathbf{j} - \mathbf{i} + \sqrt{2}\mathbf{j} + \mathbf{k}] \\
&= [0, \mathbf{i} + \sqrt{2}\mathbf{j} + \mathbf{k}]
\end{aligned} \tag{86}$$

Which is a **pure** quaternion and the norm of the result is:

$$\begin{aligned}
|p'| &= \sqrt{1^2 + \sqrt{2}^2 + 1^2} \\
&= \sqrt{4} \\
&= 2
\end{aligned} \tag{87}$$

which is the same as \mathbf{p} so the norm of the vector is maintained.

The image below visualizes the result of the rotation.

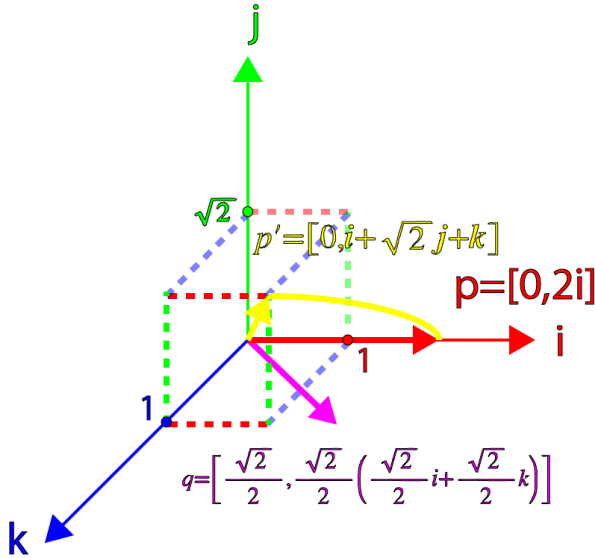


Figure 9: Quaternion Rotation (3) [\[Image Source\]](#)

So we can see that the result is a pure quaternion and that the norm of the initial vector is maintained, but the vector has been rotated 90° rather than 45° which is twice as much as desired! So in order to correctly rotate a vector \mathbf{p} by an angle θ about an arbitrary axis $\hat{\mathbf{v}}$, we must consider the half-angle and construct the following quaternion:

$$q = \left[\cos \frac{1}{2}\theta, \sin \frac{1}{2}\theta \hat{\mathbf{v}} \right] \tag{88}$$

Which is the general form of a rotation quaternion!

7 Quaternion Interpolation

One of the most important reasons for using quaternions in computer graphics is that quaternions are very good at representing rotations in space. Quaternions overcome the issues that plague other methods of rotating points in 3D space such as [Gimbal lock](#) which is an issue when you represent your rotation with euler angles.

Using quaternions, we can define several methods that represents a rotational interpolation in 3D space. The first method I will examine is called **SLERP** which is used to smoothly interpolate a point between two orientations. The second method is an extension of **SLERP** called **SQAD** which is used to interpolate through a sequence of orientations that define a path.

7.1 SLERP

SLERP stands for **Spherical Linear Int erpolation**. **SLERP** provides a method to smoothly interpolate a point about two orientations.

I will represent the first orientation as q_1 and the second orientation as q_2 . The point that is interpolated will be represented by \mathbf{p} and the interpolated point will be represented by \mathbf{p}' . The interpolation parameter t will interpolate \mathbf{p} from q_1 when $t = 0$ to q_2 when $t = 1$.

The standard linear interpolation formula is:

$$\mathbf{p}' = \mathbf{p}_1 + t(\mathbf{p}_2 - \mathbf{p}_1) \tag{89}$$

The general steps to apply this equation are:

- Compute the difference between \mathbf{p}_1 and \mathbf{p}_2 .
- Take the fractional part of that difference.
- Adjust the original value by the fractional difference between the two points.

We can use the same basic principle to interpolate between two quaternion orientations.

7.1.1 Quaternion Difference

The first step dictates that we must compute the difference between q_1 and q_2 . With regards to quaternions, this is equivalent to computing the angular difference between the two quaternions.

$$\Delta q = q_1^{-1} q_2 \tag{90}$$

7.1.2 Quaternion Exponentiation

The next step is to take the fractional part of that difference. We can compute the fractional part of a quaternion by raising it to a power whose value is in the range $0 \dots 1$.

The general formula for quaternion exponentiation is:

$$q^t = \exp(t \log q) \quad (91)$$

Where the exponential function for quaternions is given by:

$$\begin{aligned} \exp(q) &= \exp([0, \theta \hat{\mathbf{v}}]) \\ &= [\cos \theta, \sin \theta \hat{\mathbf{v}}] \end{aligned} \quad (92)$$

And the logarithm of a quaternion is given by:

$$\begin{aligned} \log q &= \log(\cos \theta + \sin \theta \hat{\mathbf{v}}) \\ &= \log(\exp(\theta \hat{\mathbf{v}})) \\ &= \theta \hat{\mathbf{v}} \\ &= [0, \theta \hat{\mathbf{v}}] \end{aligned} \quad (93)$$

For $\mathbf{t}=\mathbf{0}$, we have:

$$\begin{aligned} q^0 &= \exp(0 \log q) \\ &= \exp([\cos(0), \sin(0) \hat{\mathbf{v}}]) \\ &= \exp([1, \mathbf{0}]) \\ &= [1, \mathbf{0}] \end{aligned} \quad (94)$$

And for $\mathbf{t}=\mathbf{1}$, we have:

$$\begin{aligned} q^1 &= \exp(\log q) \\ &= q \end{aligned} \quad (95)$$

7.1.3 Fractional Difference of Quaternions

And to compute the interpolated angular rotation, we adjust the original orientation q_1 and adjust it by the fractional part of the difference between q_1 and q_2 .

$$q' = q_1 (q_1^{-1} q_2)^t \quad (96)$$

Which is the general form of spherical linear interpolation using quaternions. However, this is not the form of the **slerp** equation that is commonly used in practice.

We can apply a similar formula for performing a spherical interpolation of vectors to quaternions. The general form of a spherical interpolation for vectors is defined as:

$$\mathbf{v}_t = \frac{\sin(1-t)\theta}{\sin \theta} \mathbf{v}_1 + \frac{\sin t\theta}{\sin \theta} \mathbf{v}_2 \quad (97)$$

This is visualized in the following image.

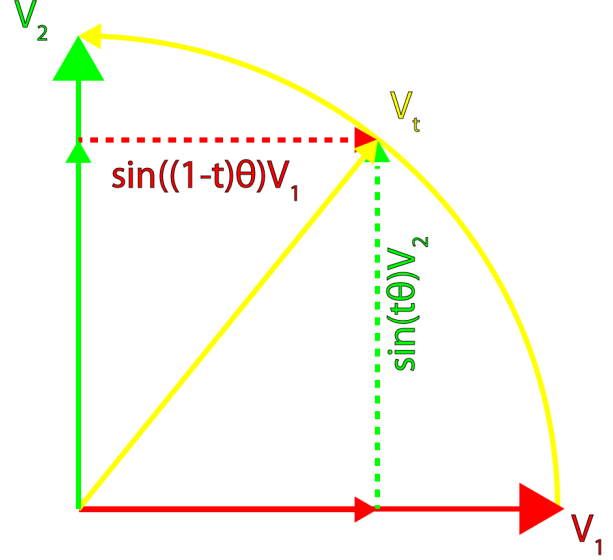


Figure 10: Quaternion Interpolation [\[Image Source\]](#)

This formula can be applied unmodified to quaternions:

$$q_t = \frac{\sin(1-t)\theta}{\sin \theta} q_1 + \frac{\sin t\theta}{\sin \theta} q_2 \quad (98)$$

And we can obtain the angle θ by computing the dot-product between q_1 and q_2 .

$$\begin{aligned} \cos \theta &= \frac{q_1 \cdot q_2}{|q_1| |q_2|} \\ &= \frac{s_1 s_2 + x_1 x_2 + y_1 y_2 + z_1 z_2}{|q_1| |q_2|} \\ \theta &= \cos^{-1} \left(\frac{s_1 s_2 + x_1 x_2 + y_1 y_2 + z_1 z_2}{|q_1| |q_2|} \right) \end{aligned} \quad (99)$$

7.1.4 Considerations

There are two issues with this implementation which must be taken into consideration during implementation.

First, if the quaternion dot-product results in a negative value, then the resulting interpolation will travel the “long-way” around the 4D sphere which is not necessarily what we want. To solve this problem, we can test the result of the dot product and if it is negative, then we can negate one of the orientations. Negating the scalar and the vector part of the quaternion does not change the orientation that it represents but by doing this we guarantee that the rotation will be applied in the “shortest” path.

The other problem arises when the angular difference between q_1 and q_2 is very small then $\sin \theta$ becomes 0. If this happens, then we will get an undefined result

when we divide by $\sin \theta$. In this case, we can fall-back to using linear interpolation between q_1 and q_2 .

7.2 SQUAD

Just as a **SLERP** can be used to compute an interpolation between two quaternions, a **SQUAD** (Spherical and **Q**uadrangle) can be used to smoothly interpolate over a path of rotations.

If we have the sequence of quaternions:

$$q_1, q_2, q_3, \dots, q_{n-2}, q_{n-1}, q_n \quad (100)$$

And we also define the “helper” quaternion (s_i) which we can consider an intermediate control point:

$$s_i = \exp \left(-\frac{\log(q_{i+1}q_i^{-1}) + \log(q_{i-1}q_i^{-1})}{4} \right) q_i \quad (101)$$

Then the orientation along the sub-curve defined by:

$$q_{i-1}, q_i, q_{i+1}, q_{i+2} \quad (102)$$

at time t is given by:

$$\begin{aligned} \text{squad}(q_i, q_{i+1}, s_i, s_{i+1}, t) = \\ \text{slerp}(\text{slerp}(q_i, q_{i+1}, t), \text{slerp}(s_i, s_{i+1}, t), 2t(1-t)) \end{aligned} \quad (103)$$

8 Conclusion

Despite being extremely difficult to understand, quaternions provide a few obvious advantages over using matrices or Euler angles for representing rotations.

- Quaternion interpolation using SLERP and SQUAD provide a way to interpolate smoothly between orientations in space.
- Rotation concatenation using quaternions is faster than combining rotations expressed in matrix form.
- For unit-norm quaternions, the inverse of the rotation is taken by subtracting the vector part of the quaternion. Computing the inverse of a rotation matrix is considerably slower if the matrix is not orthonormalized (if it is, then it’s just the transpose of the matrix).
- Converting quaternions to matrices is slightly faster than for Euler angles.
- Quaternions only require 4 numbers (3 if they are normalized. The Real part can be computed at run-time) to represent a rotation where a matrix requires at least 9 values.

However for all of the advantages in favor of using quaternions, there are also a few disadvantages.

- Quaternions can become invalid because of floating-point round-off error however this “error creep” can be resolved by re-normalizing the quaternion.

- And probably the most significant deterrent for using quaternions is that they are very hard to understand. I hope that this issue is resolved after reading this article.

There are several math libraries that implement quaternions and a few of those libraries implement quaternions correctly. In my personal experience, I find [GLM](#) (OpenGL Math Library) to be a good math library with a good implementation of quaternions. If you are interested in using quaternions in your own applications, this is the library I would recommend.

9 Download the Demo

I created a small demo that demonstrates how a quaternion is used to rotate an object in space. The demo was created with [Unity 3.5.2](#) which you can [download for free](#) and view the demo script files. The zip file also contains a Windows binary executable but Using Unity, you can also generate a Mac application (and Unity 4 introduces Linux builds as well).

[Understanding Quaternions.zip](#)

10 Reference

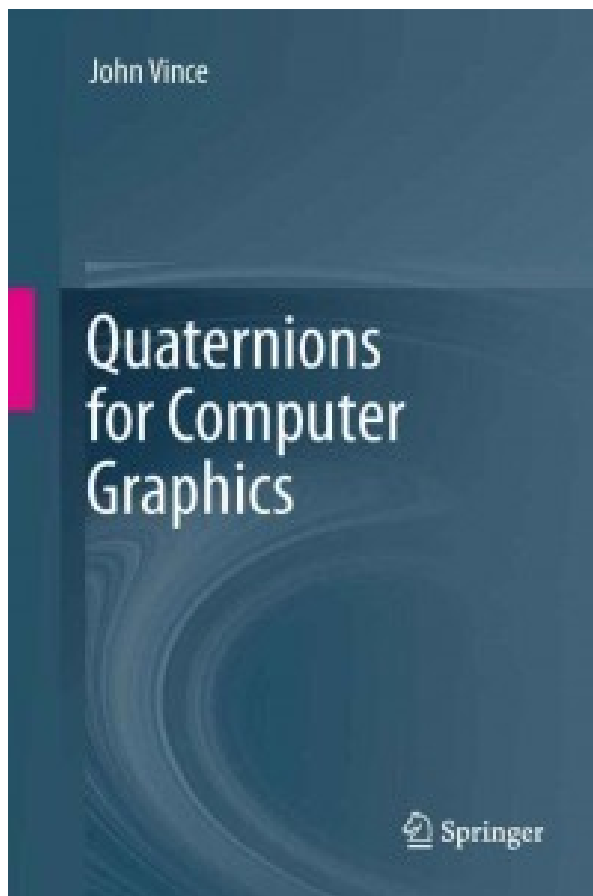


Figure 11: *Quaternions for Computer Graphics* [\[Image Source\]](#)

Vince, J (2011). *Quaternions for Computer Graphics*.

1st. ed. London: Springer.

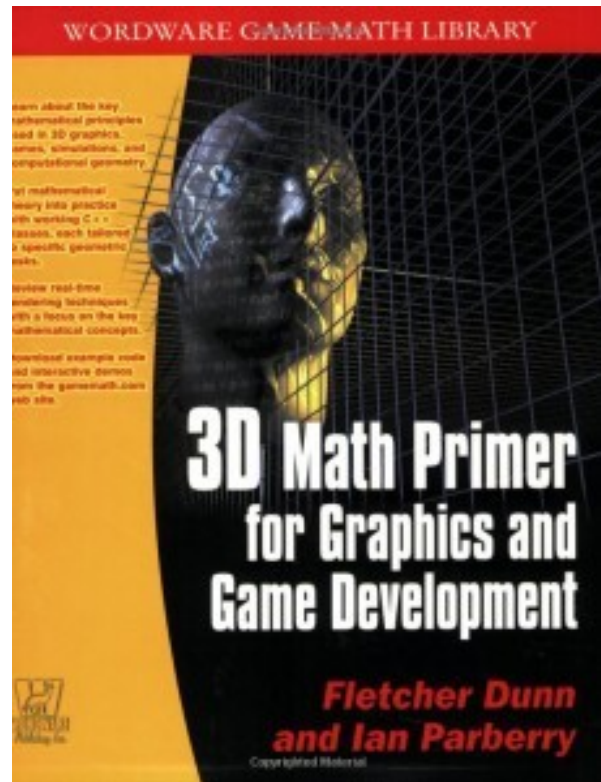


Figure 12: *3D Math Primer for Graphics and Game Development* [\[Image Source\]](#)

Dunn, F. and Parberry, I. (2002). *3D Math Primer for Graphics and Game Development*. 1st. ed. Plano, Texas: Wordware Publishing, Inc.

[Quaternions – Wikipedia](#)

[Quaternions and Spatial Rotation – Wikipedia](#)