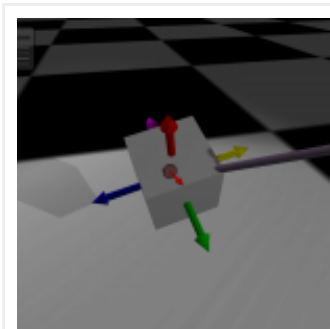


3D Game Engine Programming

Helping you build your dream game engine

Understanding Quaternions

Posted on **June 25, 2012** by **Jeremiah**



— Understanding
Quaternions

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.

Contents [\[hide\]](#)

- 1 Introduction
- 2 Complex Numbers
 - 2.1 Adding and Subtracting Complex Numbers
 - 2.2 Multiply a Complex Number by a Scalar
 - 2.3 Product of Complex Numbers
 - 2.4 Square of Complex Numbers
 - 2.5 Complex Conjugate
 - 2.6 Absolute Value of a Complex Number
 - 2.7 Quotient of Two Complex Numbers
- 3 Powers of i
- 4 The Complex Plane
 - 4.1 Rotors

5 Quaternions

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

6 Rotations

7 Quaternion Interpolation

- 7.1 SLERP
 - 7.1.1 Quaternion Difference
 - 7.1.2 Quaternion Exponentiation
 - 7.1.3 Fractional Difference of Quaternions
 - 7.1.4 Considerations

7.2 SQUAD

8 Conclusion

9 Download the Demo

10 Reference

! You cannot fully understand quaternions in just 45 minutes.
This article is extremely math intensive and is not intended for the weak-hearted.

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.

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



- William Rowan Hamilton Plaque on Broome Bridge on the Royal Canal commemorating his discovery of the fundamental formula for quaternion multiplication.

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:

$$x^2 + 1 = 0$$

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:

$$i^2 = -1$$

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:

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

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

Adding and Subtracting Complex Numbers

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

Addition:

$$(a_1 + b_1 i) + (a_2 + b_2 i) = (a_1 + a_2) + (b_1 + b_2)i$$

Subtraction:

$$(a_1 + b_1 i) - (a_2 + b_2 i) = (a_1 - a_2) + (b_1 - b_2)i$$

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:

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

Product of Complex Numbers

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

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

Square of Complex Numbers

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

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

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^* .

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

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

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

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 $|z|$:

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

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.

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

Powers of i

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

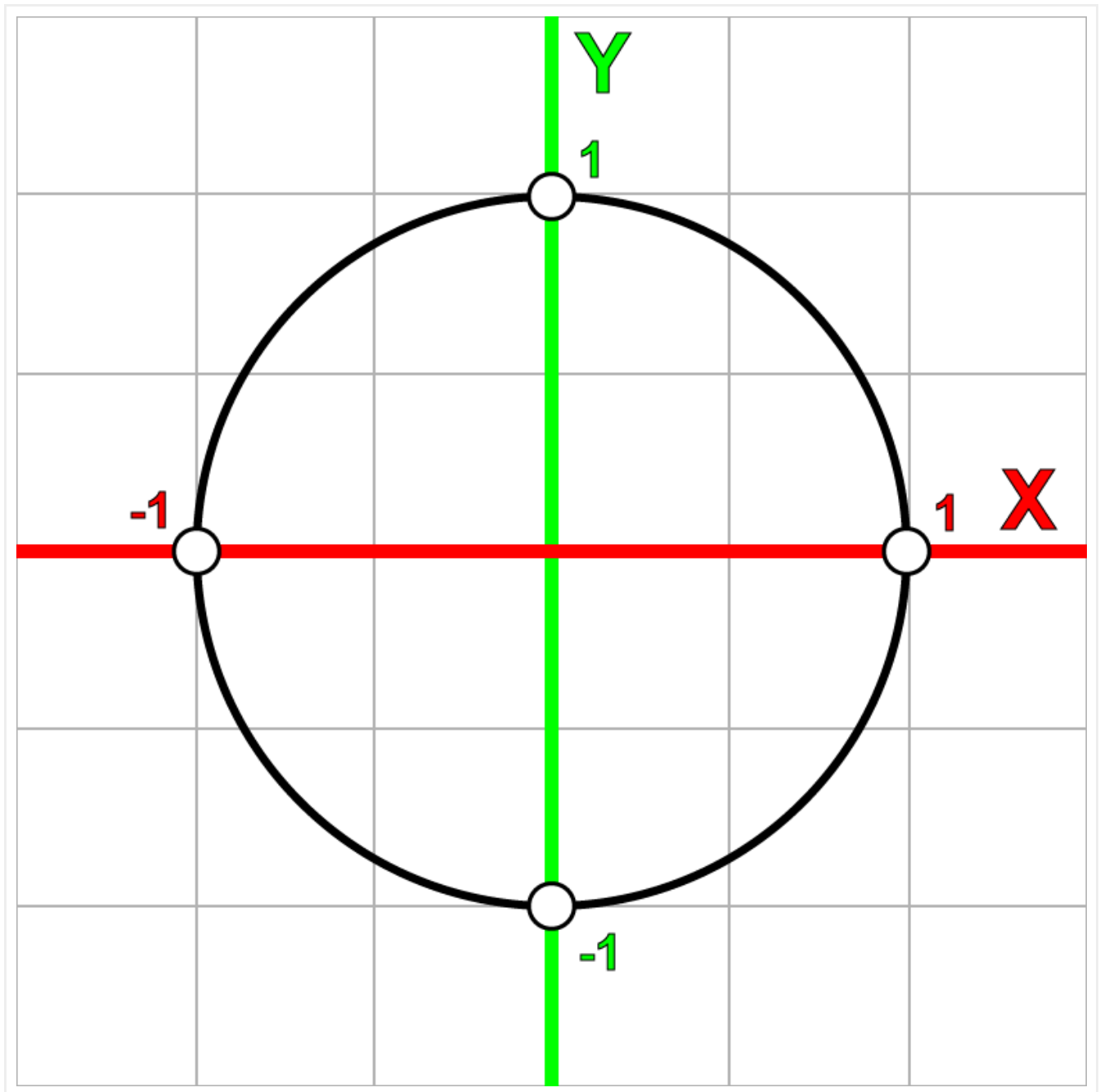
$$\begin{aligned}
 i^0 &= 1 \\
 i^1 &= i \\
 i^2 &= -1 \\
 i^3 &= ii^2 = -i \\
 i^4 &= i^2 i^2 = 1 \\
 i^5 &= ii^4 = i \\
 i^6 &= ii^5 = i^2 = -1
 \end{aligned}$$

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.

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

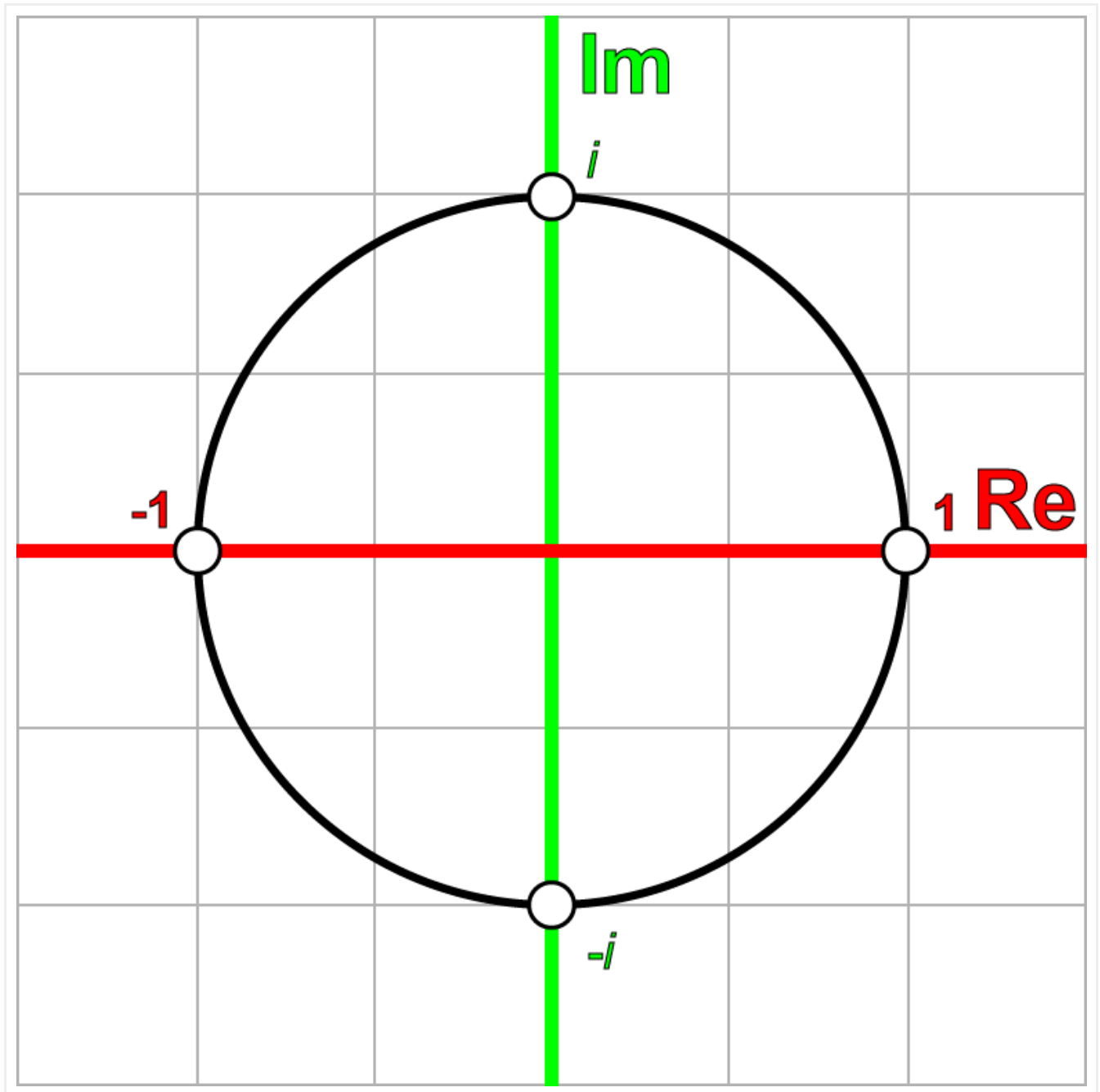
You may have seen a similar pattern in mathematics before but in the form $(x, y, -x, -y, x, \dots)$ which is generated by rotating a point 90° counter-clockwise on a 2D Cartesian plane and the sequence $(x, -y, -x, y, x, \dots)$ is generated by rotating a point 90° clockwise on a 2D Cartesian plane.



— Cartesian Plane

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.



— Complex Plane

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° increments.

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

$$p = 2 + i$$

and we multiply it by i gives q :

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

Multiplying q by i gives r :

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

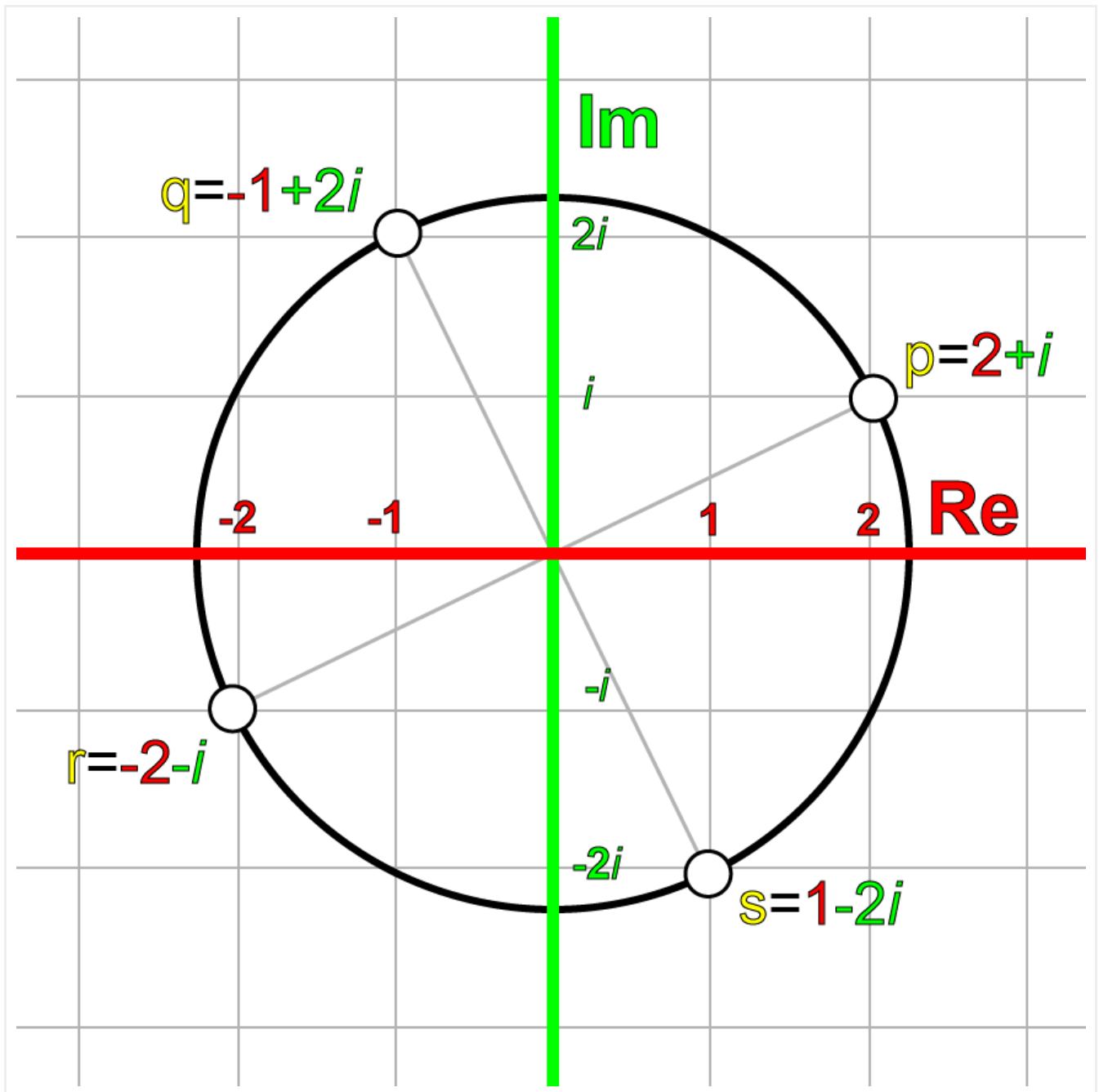
And multiplying r by i gives s :

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

And multiplying s by i gives t :

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

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



— Complex Numbers on the Complex Plane

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

Rotors

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

$$q = \cos \theta + i \sin \theta$$

Multiplying any complex number by the rotor q produces the general formula:

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

Which can also be written in matrix form:

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

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

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

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

Where, according to Hamilton's famous expression:

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

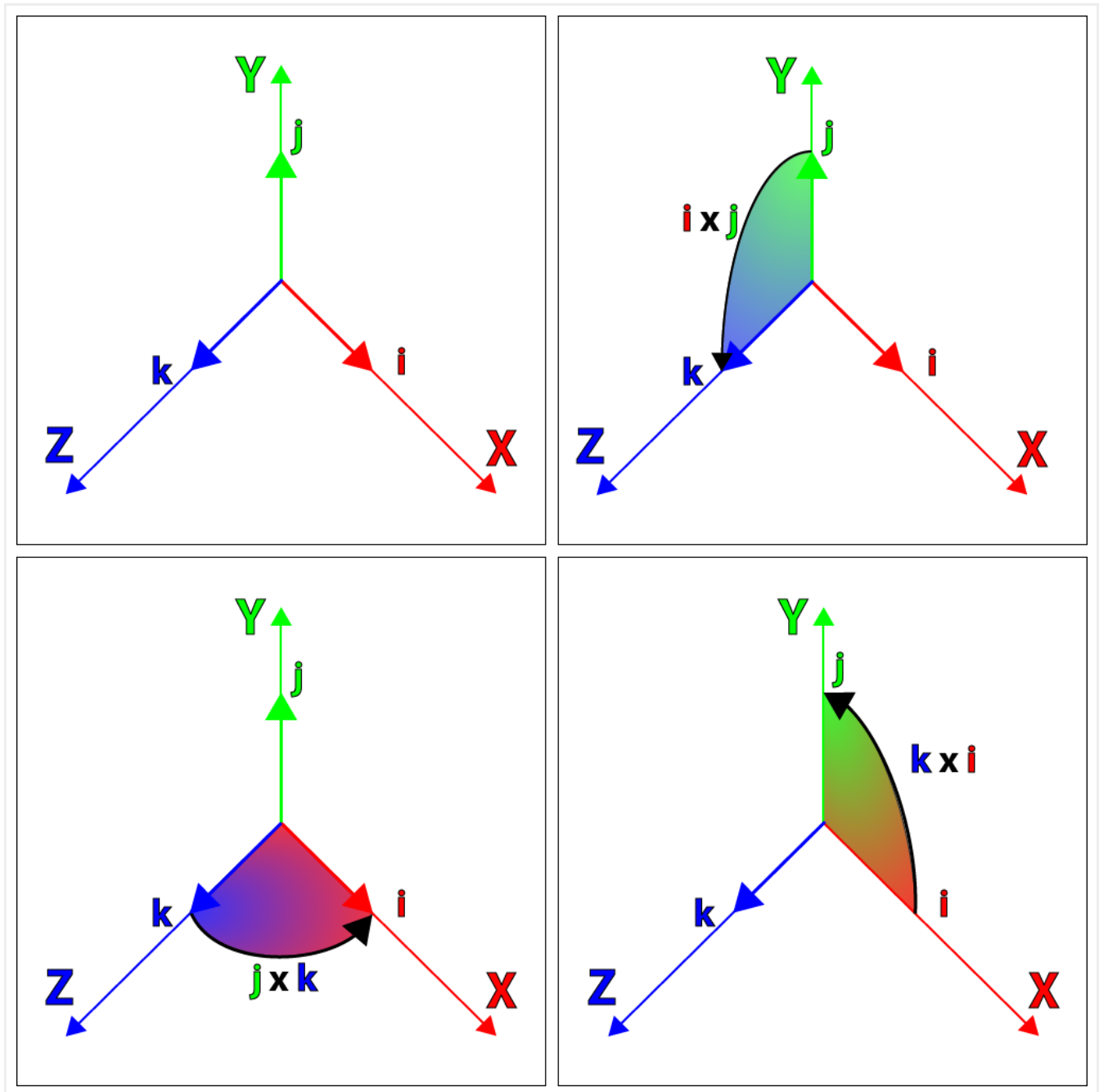
and

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

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:

$$\begin{aligned}
 \mathbf{x} \times \mathbf{y} &= \mathbf{z} & \mathbf{y} \times \mathbf{z} &= \mathbf{x} & \mathbf{z} \times \mathbf{x} &= \mathbf{y} \\
 \mathbf{y} \times \mathbf{x} &= -\mathbf{z} & \mathbf{z} \times \mathbf{y} &= -\mathbf{x} & \mathbf{x} \times \mathbf{z} &= -\mathbf{y}
 \end{aligned}$$

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



— Visualizing the Properties of \mathbf{ij} , \mathbf{jk} , \mathbf{ki}

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

Quaternions as an Ordered Pair

We can also represent quaternions as an ordered pair:

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

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

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

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

Adding and Subtracting Quaternions

Quaternions can be added and subtracted similar to complex numbers:

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

Quaternion Products

We can also express the product of two quaternions:

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

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),

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

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

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

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

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

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

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

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:

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

And if we substitute,

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

We get:

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

Which is the general equation of a quaternion product.

A Real Quaternion

A **Real** Quaternion is a quaternion with a vector term of **0**:

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

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

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

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

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

Multiplying a Quaternion by a Scalar

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

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

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

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

Pure Quaternions

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

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

Or, written in its component parts:

$$q = xi + yj + zk$$

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

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

According to the quaternion product rule shown above.

Additive Form of a Quaternion

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

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

Unit Quaternion

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

$$\mathbf{v} = v\hat{\mathbf{v}} \text{ where } v = |\mathbf{v}| \text{ and } |\hat{\mathbf{v}}| = 1$$

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

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

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

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

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:

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

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

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

Quaternion Conjugate

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

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

And the product of a quaternion with its conjugate gives:

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

Quaternion Norm

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

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

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

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

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

$$qq^* = |q|^2$$

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 $|q|$:

$$q' = \frac{q}{\sqrt{s^2 + v^2}}$$

As an example, let's normalize the quaternion:

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

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

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

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

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

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:

$$q^{-1} = \frac{q^*}{|q|^2}$$

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

$$qq^{-1} = [1, \mathbf{0}] = 1$$

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

$$q^* qq^{-1} = q^*$$

And by substitution we get:

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

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

$$q^{-1} = q^*$$

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:

$$\begin{aligned} q_1 &= [s_1, x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k}] \\ q_2 &= [s_2, x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k}] \\ q_1 \cdot q_2 &= s_1 s_2 + x_1 x_2 + y_1 y_2 + z_1 z_2 \end{aligned}$$

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

$$\cos \theta = \frac{s_1 s_2 + x_1 x_2 + y_1 y_2 + z_1 z_2}{|q_1| |q_2|}$$

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

$$\cos \theta = s_1 s_2 + x_1 x_2 + y_1 y_2 + z_1 z_2$$

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:

$$q = \cos \theta + i \sin \theta$$

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:

$$q = [\cos \theta, \sin \theta \mathbf{v}]$$

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:

$$p = [0, \mathbf{p}]$$

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

$$q = [s, \lambda \hat{\mathbf{v}}]$$

Then,

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

We see that the result is a general quaternion with both 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:

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

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

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

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

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

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

$$\mathbf{p} = [0, 2\mathbf{i}]$$

Now let's find the product of qp :

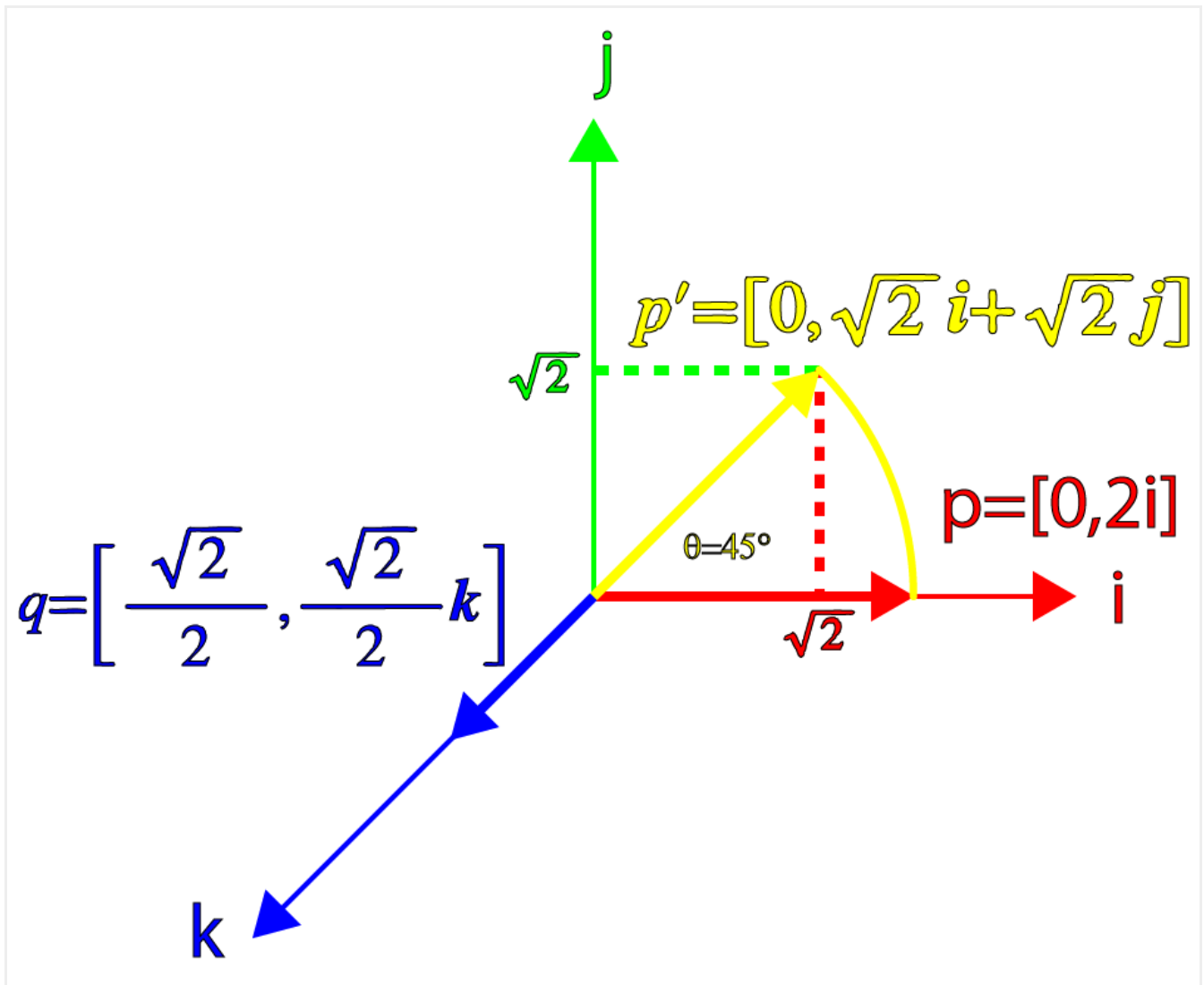
$$\begin{aligned} \mathbf{p}' &= qp \\ &= \left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \mathbf{k} \right] [0, 2\mathbf{i}] \\ &= \left[0, 2\frac{\sqrt{2}}{2}\mathbf{i} + 2\frac{\sqrt{2}}{2}\mathbf{k} \times \mathbf{i} \right] \\ &= [0, \sqrt{2}\mathbf{i} + \sqrt{2}\mathbf{j}] \end{aligned}$$

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:

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

Which is exactly the result we expected!

We can visualize this by the following image:



— Quaternion Rotation (1)

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:

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

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

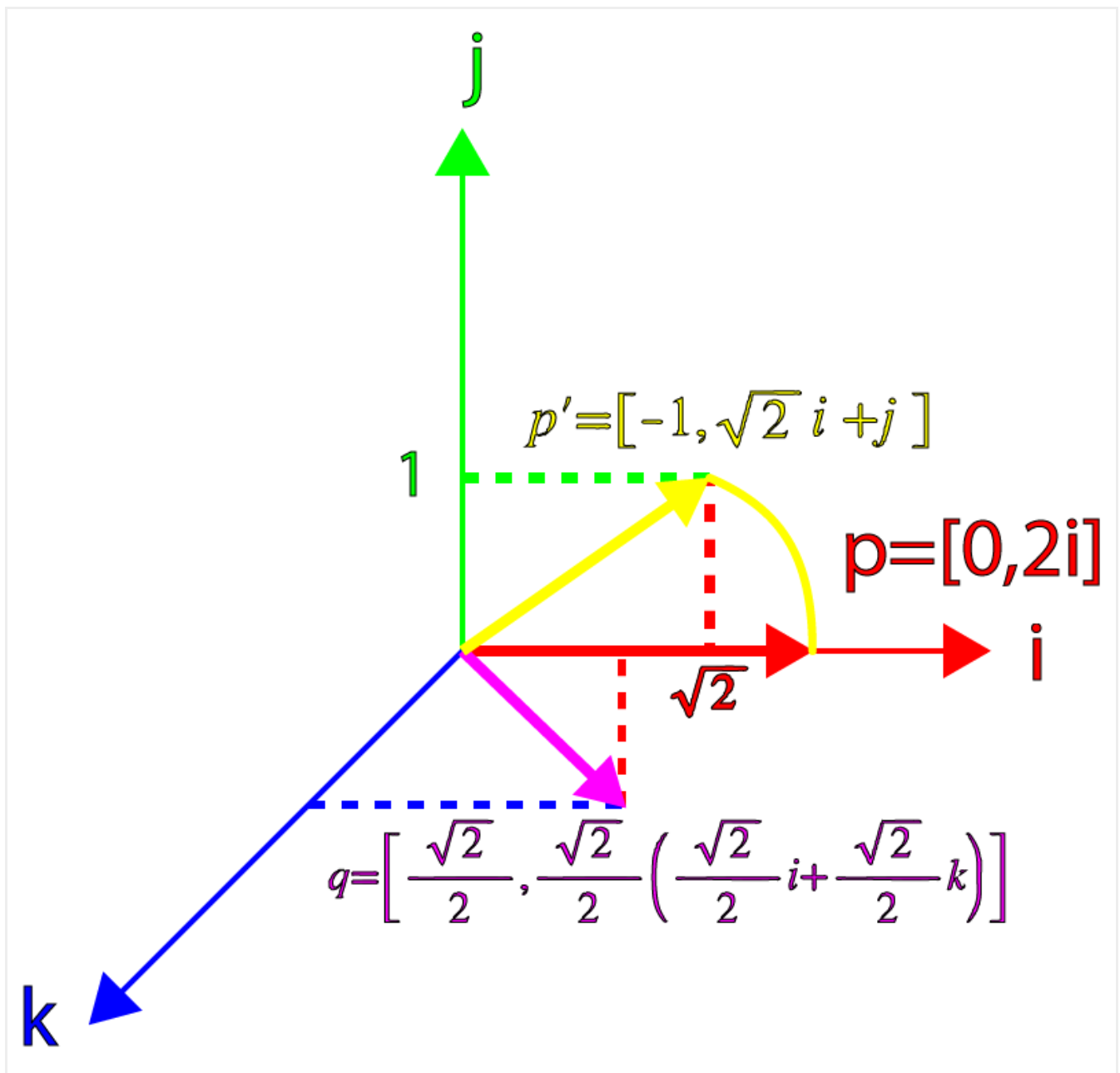
$$\begin{aligned}p' &= qp \\ &= [\cos \theta, \sin \theta \hat{\mathbf{v}}][0, \mathbf{p}] \\ &= [-\sin \theta \hat{\mathbf{v}} \cdot \mathbf{p}, \cos \theta \mathbf{p} + \sin \theta \hat{\mathbf{v}} \times \mathbf{p}]\end{aligned}$$

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

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

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.



— Quaternion Rotation (2)

i Technically, it's incorrect to represent the quaternion 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} :

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

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

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

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

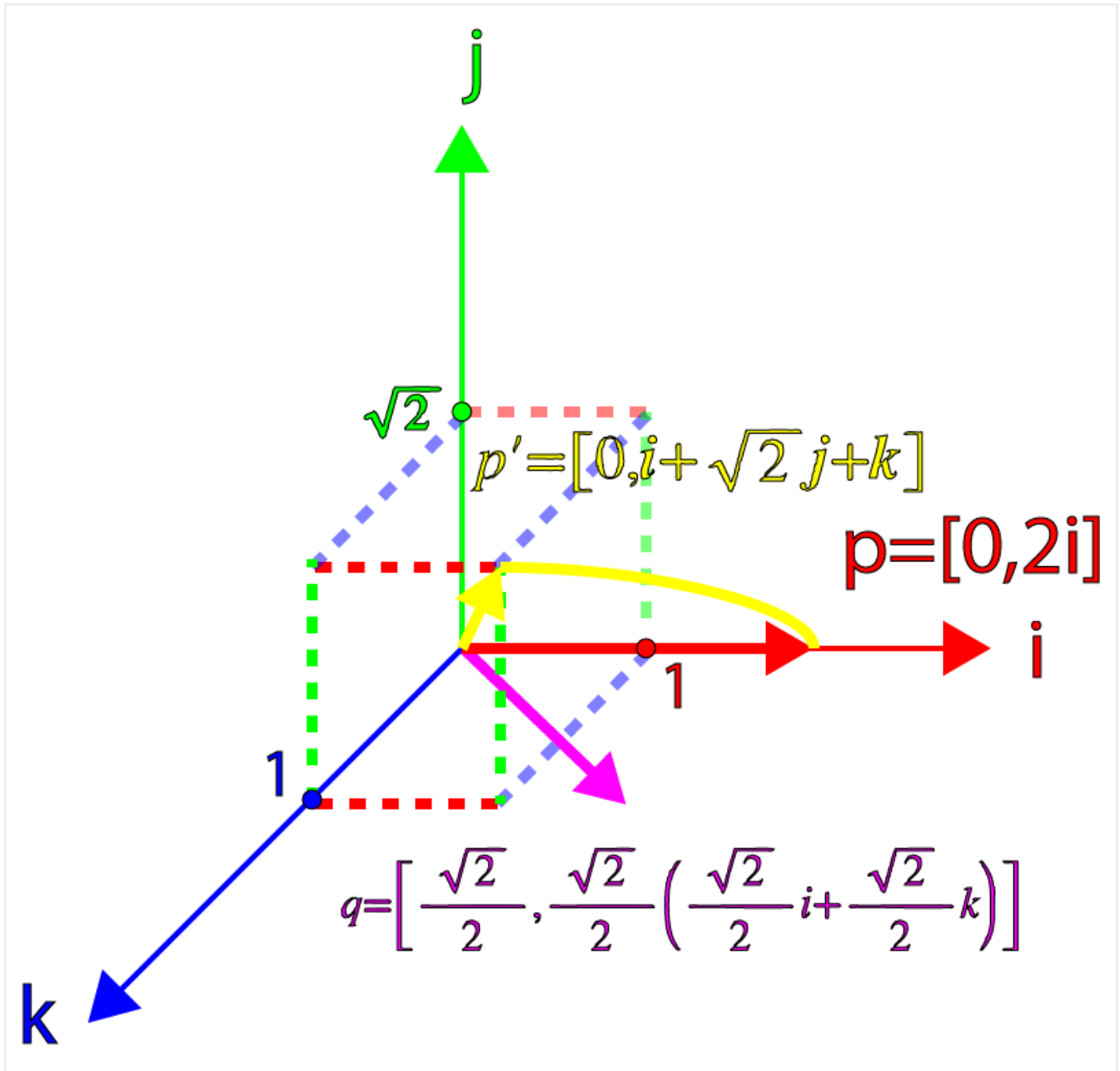
$$\begin{aligned} qp &= [-1, \sqrt{2}\mathbf{i} + \mathbf{j}] \\ qpq^{-1} &= [-1, \sqrt{2}\mathbf{i} + \mathbf{j}] \frac{1}{2} [\sqrt{2}, -\mathbf{i} - \mathbf{k}] \\ &= \frac{1}{2} [-\sqrt{2} - (\sqrt{2}\mathbf{i} + \mathbf{j}) \cdot (-\mathbf{i} - \mathbf{k}), \mathbf{i} + \mathbf{k} + \sqrt{2}(\sqrt{2}\mathbf{i}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

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

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

which is the same as p so the norm of the vector is maintained.

The image below visualizes the result of the rotation.



— Quaternion Rotation (3)

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:

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

Which is the general form of a rotation quaternion!

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.

SLERP

SLERP stands for **S**pherical **L**inear **I**nter**p**olation. **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:

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

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.

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.

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

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:

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

Where the exponential function for quaternions is given by:

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

And the logarithm of a quaternion is given by:

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

For $t = 0$, we have:

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

And for $t = 1$, we have:

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

FRACTIONAL DIFFERENCE OF QUATERNIONS

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

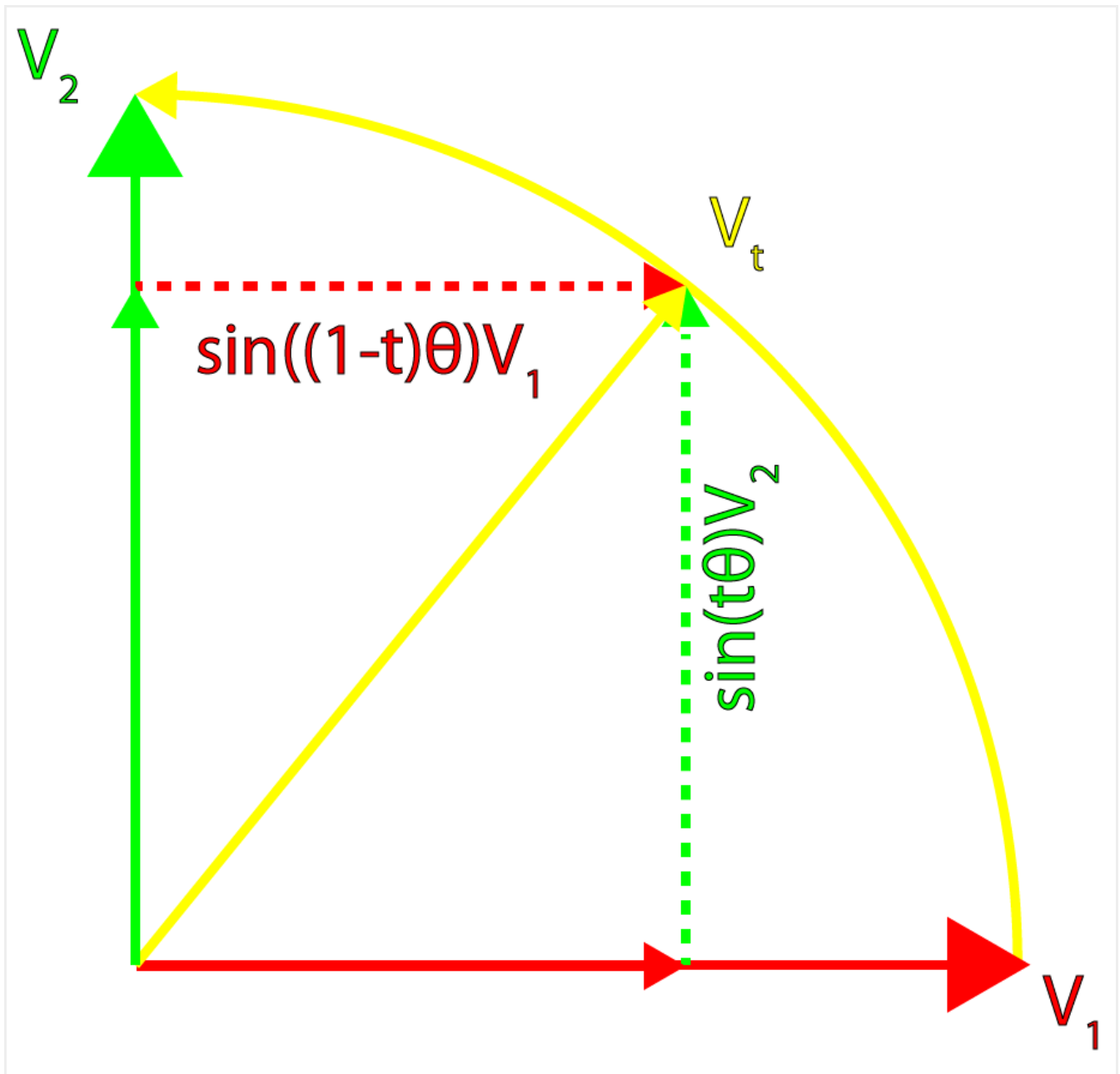
$$q' = q_1 (q_1^{-1} q_2)^t$$

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:

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

This is visualized in the following image.



— Quaternion Interpolation

This formula can be applied unmodified to quaternions:

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

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

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

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 .

SQUAD

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

If we have the sequence of quaternions:

$$q_1, q_2, q_3, \dots, q_{n-2}, q_{n-1}, q_n$$

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

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

Then the orientation along the sub-curve defined by:

$$q_{i-1}, q_i, q_{i+1}, q_{i+2}$$

at time t is given by:

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

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.

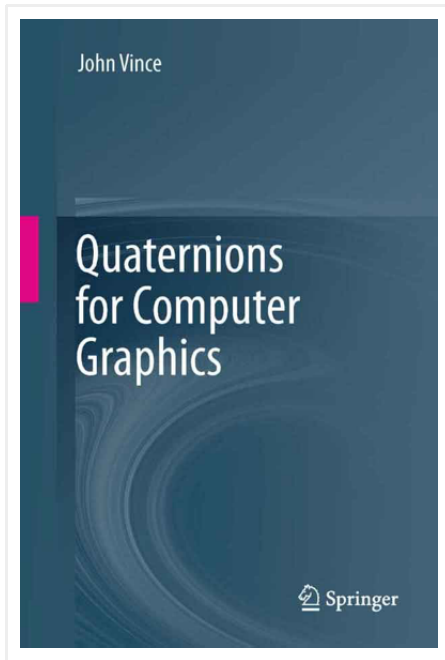
There are several math libraries that implement quaternions and a few of those libraries implement quaternions correctly. In my personal experience, I find [GLM](#) (OpenGL Math Library) to be a good math library with a good implementation of quaternions. If you are interested in using quaternions in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

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).

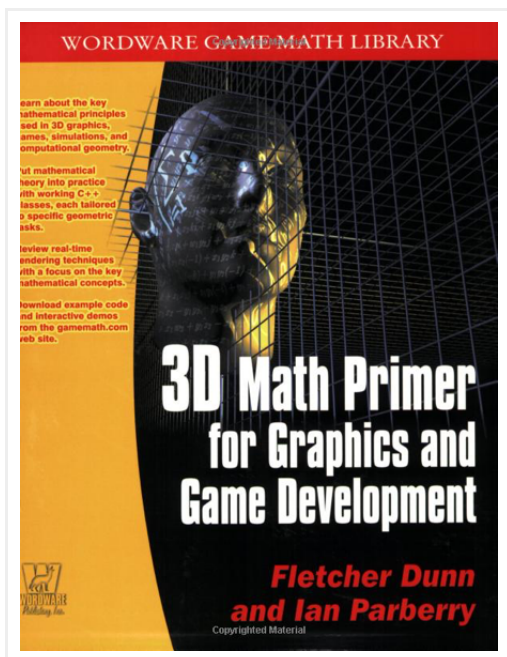
[Understanding Quaternions.zip](#)

Reference



Vince, J (2011). Quaternions for Computer Graphics. 1st. ed.
London: Springer.

— Quaternions for Computer Graphics



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

— 3D Math Primer for Graphics and Game Development

[Quaternions – Wikipedia](#)

[Quaternions and Spatial Rotation – Wikipedia](#)

This entry was posted in [Math](#) and tagged [3D](#), [complex](#), [Graphics](#), [imaginary](#), [Programming](#), [quaternion](#), [rotations](#), [rotors](#), [slerp](#), [squad](#) by [Jeremiah](#). Bookmark the [permalink](#) [<https://www.3dgep.com/understanding-quaternions/>].

97 THOUGHTS ON “UNDERSTANDING QUATERNIONS”



lawrence brooks jr.
on [June 28, 2012 at 9:31 pm](#) said:

I like this website.



[Jeremiah van Oosten](#)
on [June 29, 2012 at 9:39 am](#) said:

Thanks Lawrence, you comments are welcome.



[Dave Touretzky](#)
on [June 30, 2012 at 6:11 pm](#) said:

This is a wonderfully clear and concise introduction to quaternions! There are a few typos that could be corrected. “It’s” should be “its” throughout. (The first is a contraction of “it is”, the second is the possessive form of “it”.) Also, “gimble” should be “gimbal”. And “principal” should be “principle”: both are words but they have different meanings.

I will show this tutorial to all my robotics students.



[Jeremiah van Oosten](#)
on [July 2, 2012 at 10:00 am](#) said:

Dave,

Thanks for the corrections! Let me know if you find anything else.



Dan

on **June 5, 2016 at 9:00 pm** said:

“Besides being extremely difficult to understand”
would make more sense as:
“Despite being extremely difficult to understand”

**Jeremiah van Oosten**on **July 7, 2016 at 12:40 pm** said:

Dan,

Thanks for your feedback. I have made the change.



Gary

on **July 1, 2012 at 5:33 pm** said:

Well done! I find this website through google and found it very informative and useful. I would like to learn graphics programming for a long time, but it's quite difficult to get started, especially the mathematics, may you kindly suggest what mathematics I should know and how to learn graphics programming effectively? Thank you very much.

**Jeremiah van Oosten**on **July 2, 2012 at 10:07 am** said:

Gary,

For basic graphics programming you will want to have a good understanding of vector algebra (vector addition & subtraction, dot & cross products) as well as matrix algebra (matrix products, inverse & transpose). Also knowledge of basic lighting equations is important.

I recommend you read the books I've referenced in the math articles here: <http://3dgep.com/?cat=27> as well as read the the OpenGL articles here <http://3dgep.com/?cat=11>.

If you'd like to get into shader programming, then you should follow the Cg articles here: <http://3dgep.com/?cat=108>.

I also plan to make a set of GLSL articles, but I can't promise when they will be ready.

**Trond**on **November 28, 2012 at 1:59 pm** said:

Thank you for writing such a thorough post on quaternions, it is much appreciated!

Unfortunately though, quite a few of the equation listings don't show properly in neither Firefox nor Chrome on my Win7 computer (see <http://grab.by/hQJm> for example). Just wanted to let you know 😊

**Jeremiah van Oosten**on **November 29, 2012 at 5:40 pm** said:

Thanks for pointing this out. A few weeks after creating this tutorial, the LaTeX generator I was using went offline and i had to switch to another provider. This provider seems to have a different parser than the one I was using when I created the page on quaternions in the first place. I switched again and some of the formulas that were not showing up before are fixed now but others are broken. I really need to install my own CGI handler for LaTeX on my own server.

Thanks for the heads-up!

**Luke**on **May 18, 2013 at 8:15 pm** said:

I find this to be a great tutorial but I'm having the same problem Trond had with every single equation, but the way it looks it seems to be a great tutorial.



Jeremiah van Oosten

on **August 20, 2013 at 5:39 pm** said:

Luke,

I have moved the script that is used to generate the formulas to a new server. There have been some hiccups with this script but I hope to have them all fixed now.



Henry

on **December 3, 2012 at 3:43 pm** said:

What happen for some picture of equation.



Jeremiah van Oosten

on **December 6, 2012 at 12:44 pm** said:

Henry, when I created this article I was using a different source for the LaTeX generation. That source has since become unavailable so I had to switch to another source which apparently handles LaTeX differently. I need to go through each equation which isn't rendering correctly to find out why this is happening but this takes time... It will be fixed in the future.



paul jordan

on **March 25, 2013 at 1:44 am** said:

This is the only coherent introduction to quaternions I have found on the web. It is fairly complex and I will have to read through this several times. I am assuming you are drawing heavily from Vince, as Dunn is pretty sketchy. Thanks for pulling this together—I feel like I have found a starting point!



Sundaram RM

on **March 27, 2013 at 12:34 pm** said:

Excellent tutorial/article on quaternions, it helped me get a good insight, thanks for the effort and this website is great!



Brian Abraham

on **April 1, 2013 at 2:00 pm** said:

While familiar with the use of rotational matrices all my previous attempts to get a hold of quaternions have failed.

What worked for me with your explanation is that you explained it by analogy to rotation in the complex plane which I already understood: the next step was then easy for me.

Many thanks for your work and making it available.



Shammah

on **May 13, 2013 at 8:10 am** said:

Great tutorial, loving it so far!

I've spot one minor error I think though:

> The interpolation parameter t will interpolate P from q_1 when $t = 0$ to q_1 when $t = 1$.

Shouldn't this be:

> The interpolation parameter t will interpolate P from q_1 when $t = 0$ to q_2 when $t = 1$.



Jeremiah van Oosten

on **May 22, 2013 at 11:06 am** said:

Shammah,

Thanks for pointing this out!



Felipe Tavares

on **June 1, 2013 at 7:36 am** said:

WOW! Very nice article. I actually implemented a simple quaternion library in C++ while reading it :). Very well presented too.



mica barshap

on **July 21, 2013 at 10:36 pm** said:

Please it look like very important&impressive post, but you aware that we can't read it because lack of picture? equations?
Can you attach or sent a full pdf format to read it please?



Jeremiah van Oosten

on **August 20, 2013 at 1:59 pm** said:

Mica,

Sorry if the formulas did not show up for you. The images for the formulas are generated on another sever than where the article itself is hosted. If the server that generates the images is inaccessible, then the formulas may not show up. This is annoying but if the images don't show up, then check back in a few minutes and perhaps it is fixed at that time. I strive for

100% up-time, but some downtime is unavoidable.



David

on **April 10, 2019 at 2:34 am** said:

The images are ok, but the math markup isn't rendering. Can this be fixed? There's a lot of really useful info here, but without the equations rendering properly it's very difficult to read.



Jeremiah

on **April 12, 2019 at 1:30 pm** said:

David: The formulas on this page are rendered by your browser using a JavaScript library called [MathJax](#). If the formulas are not rendering for you, it is probably because you have disabled JavaScript in your browser.

To see if the formulas should render, try the demos on the MathJax website: <https://www.mathjax.org/#demo>.



Jannik

on **December 20, 2013 at 8:36 pm** said:

Thanks, this is actually really relevant to MA2 for first year students aswell.



Ben van Enckevort

on **March 22, 2014 at 3:22 pm** said:

This was fantastic! Thank you so much for writing it up.



Gaz

on **April 12, 2014 at 4:08 pm** said:


This is a great explanation but I got really stuck at why the product of two quaternions is minus the dot product. We have:
 $SaSb - XaXb - YaYb - ZaZb$, then we substitute in that $a.b = XaXbi^2$ etc. Does this i^2 (i squared) not resolve to -1 as per Hamilton? That would make $a.b$ equal to $-XaXb - YaYb - ZaZb$ already and it wouldn't need further negation. I'm obviously missing something in the notation, perhaps that i^2 here is the cosine of 0 and therefore equals 1.
 Confused!



Jeremiah van Oosten

on **July 9, 2014 at 4:51 pm** said:

Gaz,

You are right about the fact that  as is stated in the section titled [Quaternions](#).

Maybe if I write the product rule like this:




And we can write the dot product of the vector parts of the two quaternions as:



Which we can substitute back into the original equation:



So we are not adding any extra negatives, we are just factoring out the  from the dot-product to get the real part of the quaternion product.

Does this make it more clear?



John

on **February 10, 2016 at 5:37 pm** said:

No, Gaz is right! The squared terms of i , j , and k should be left out of the dot product equation of $a \cdot b$. Then it makes sense. As others have said, the best explanation of quaternions that I have seen.

**Jeremiah van Oosten**on **July 7, 2016 at 5:37 pm** said:

John,

Agreed. I have update the article and removed the squared terms of i , j , and k .

Thanks!



stephen

on **March 9, 2016 at 3:56 pm** said:

This explanation is still not right. The dot product of a and b is $xaxb + yayb + zazb$ not minus 1 times this result. The algebraic result of squaring i , j and k to yield -1 gives us $-a.b$ which is what needs to substitute the original algebraic expression. There is some confusion in quaternion theory over when to treat i , j and k as imaginary numbers and when to treat them as unit vectors in the Cartesian system.

**Jeremiah van Oosten**on **July 7, 2016 at 4:57 pm** said:

Stephen,

Doh, you are right! I was treating i , j , and k as imaginary numbers but in the equation

they are unit vectors! I was so blind to this for so long and I couldn't understand where the confusion was.

Thanks for pointing this out. I have updated the article.



Mark

on **March 22, 2016 at 2:16 pm** said:

This is a great tutorial, but I think Gaz was onto something as I think the dot product is incorrectly defined and is then substituted into the equations incorrectly i.e. leading to two wrongs make a right!.

In the above comment, surely after 'And we can write the dot product of the vector parts of the two quaternions as:' is incorrect. the dot product is surely $a.b = xaxb + yayb + zazb$ (by definition) and so $xaxbi^2 + yaybj^2 + zazbk^2 = -xaxb - yayb - zazb = -a.b$ and then 'Which we can substitute back into the original equation:' follows and is correct.

So in the main text, after 'And if we substitute,' it should be stated that $a.b = xaxb + yayb + zazb$ and NOT $a.b = xaxbi^2 + yaybj^2 + zazbk^2$, then the rest follows.

Thanks once again for great article.

Mark



Jeremiah van Oosten

on **July 7, 2016 at 5:06 pm** said:

Mark,

Thanks for pointing this out. After reading

stephen's comment on March 9, 2016 ([here](#)) I realized my mistake. I was treating \mathbf{i} , \mathbf{j} , and \mathbf{k} as imaginary numbers but they are actually unit vectors.

I have fixed the article with this change.



Nguyen Huy Hung

on **March 24, 2016 at 12:42 pm** said:

Ok. I understand why it is $\mathbf{S} \cdot \mathbf{a} \cdot \mathbf{b} - \mathbf{a} \cdot \mathbf{b}$. But your explain is not true. In here, we represent vector \mathbf{a} and \mathbf{b} in bases $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. Therefore $\mathbf{a} = (x_a, y_a, z_a)$, $\mathbf{b} = (x_b, y_b, z_b)$ and dot product $\mathbf{a} \cdot \mathbf{b} = x_a x_b + y_a y_b + z_a z_b$. This is the reason why we need minus before $\mathbf{a} \cdot \mathbf{b}$



Jeremiah van Oosten

on **July 7, 2016 at 3:51 pm** said:

I'm not sure what part of the explanation is not true? The original equation for the real part is:



If you factor out a \mathbf{i} from the vector part you get:



Which can be rewritten as:



Is this clear?



Oskar



on **April 13, 2014 at 5:52 am** said:

Where it says:

"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:", the following formula is wrong. It should say: $z = a + bi$; $a, b \in \mathbb{R}$; $i^2 = -1$, but it does not.



Jeremiah van Oosten

on **July 9, 2014 at 3:24 pm** said:

Oskar,

Yes, you are right. Thanks for pointing this out. After switching to a different LaTeX generator some formulas didn't render correctly anymore. I hope I have fixed all of the places (in this page) where formulas were not rendered correctly anymore but if you seen any more then please let me know!



Oskar

on **April 13, 2014 at 5:56 am** said:

... and in the following formulas, about addition, subtraction and multiplication, you should not use an implication arrow, you should use an equals sign.



Jeremiah van Oosten

on **July 9, 2014 at 3:24 pm** said:

Oskar,

Yes, again you are correct. I can't remember why I had an implication arrow there but it's fixed now.



Titus

on **April 20, 2014 at 10:24 pm** said:

Thanks, this was really useful. I finally understood quaternions. I'll use them in 3D reconstruction from multiple photos.



Henning

on **June 2, 2014 at 3:25 pm** said:

Thanks for the article. Found an error in the formulas after: "Complex numbers can also be multiplied by applying normal algebraic rules."

In row three there stand a^2a^2 where it should be a^1a^2 .

Bye



Jeremiah van Oosten

on **July 9, 2014 at 12:24 pm** said:

Henning,

Thanks for pointing this out. I have fixed the article now.



gabriel

on **June 12, 2014 at 1:32 am** said:

This article it's just beautiful. Thank you.



cubase01

on **June 23, 2014 at 7:39 am** said:

beautiful article.

I suggest you try mathjax.js for equation generation. It's client-based.
no need for host or anything.



akash

on **June 25, 2014 at 11:49 pm** said:

i have a doubt. when u calculate the norms int the example, u said,
instead of 2 its is root(3). but how is it?? $\sqrt{1 + 2 + 1} = \sqrt{4} = 2$???



Jeremiah van Oosten

on **July 9, 2014 at 12:13 pm** said:

Akash,

In the example we are rotating the vector $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ 45° by the quaternion $\begin{bmatrix} 0.7071 \\ 0.7071 \\ 0 \\ 0 \end{bmatrix}$ but in order to perform this operation, we must express $\begin{bmatrix} 0.7071 \\ 0.7071 \\ 0 \\ 0 \end{bmatrix}$ as a **pure** quaternion in the form $\begin{bmatrix} 0 \\ 0.7071 \\ 0.7071 \\ 0 \end{bmatrix}$. (notice how vectors are expressed use bold-face characters and quaternions are expressed as normal (not bold) characters).

If the quaternion $\begin{bmatrix} 0 \\ 0.7071 \\ 0.7071 \\ 0 \end{bmatrix}$ correctly rotated the vector $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ then the result should also be a **pure** quaternion (with no scalar part) and the magnitude of the **vector part** should be the same as the original vector (because a rotation should not scale the original vector) however this example shows that this is not the case.

The magnitude of the original vector is

$$\sqrt{1^2 + 0^2 + 0^2} = 1$$

But the magnitude of the **vector part** of the resulting quaternion is:

$$\sqrt{0.7071^2 + 0.7071^2} = 1$$

Which is not the same magnitude as the original vector.

If I wanted to compute the magnitude of the resulting **quaternion** then I would need to consider the quaternion's scalar part

according to the formula described in the section titled [Quaternion Norm](#). But since I'm only interested in rotating a vector by a quaternion I only want to consider the result of the vector part (and thus discard the scalar part when I compute the magnitude of the resulting vector).

I hope this answers your question.



cubase01

on **July 5, 2014 at 4:39 pm** said:

and I have a question.what's that suppose to mean?

if $a+bi$ is a point,isn't that a Matrix multiply a point equals to a point more correct?



Jeremiah van Oosten

on **July 9, 2014 at 2:42 pm** said:

cubase01,

Sorry, the matrix representation of a complex number is not explained in this article. I will try to briefly explain it here.

We can represent a complex number as the matrix $\begin{bmatrix} a & -b \\ b & a \end{bmatrix}$ which is the sum of two other matrices representing the real $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and the imaginary $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ parts (note that bold, upper-case characters represent matrices):

$$\begin{bmatrix} a & -b \\ b & a \end{bmatrix}$$


Which can also be expressed in the more familiar form of the complex number:

$$a + bi$$

where $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

The matrix equivalent of 1 is the 2 x 2 identity matrix:






And as was mentioned in the section titled [Powers of i](#) the imaginary component  can be treated as a 90° counter-clockwise rotation in the complex plane which can also be represented by a rotation matrix:



Now we can express the complex number in matrix form:



So what you see in the section titled [Rotors](#) is the matrix form of a complex number and the  and  are the real and imaginary parts of a complex number and rotating a complex number (represented in matrix form) by the 2x2 counter-clockwise rotation matrix produces another complex number (represented in matrix form).

It is also interesting to note that if we express the equation  in matrix form we get:



which verifies that the square of the imaginary number is -1.

I hope this helps.



Carlo Wood

on **July 6, 2014 at 3:43 pm** said:

I think there's a little typo in http://3dgep.com/?p=1815#Quaternion_Dot_Product
It should be q1q2, not q12.



Jeremiah van Oosten

on **July 9, 2014 at 10:55 am** said:

Carlo,

Thanks for pointing this out. It should actually be:



I recently switched LaTeX engines. The new one seems to be a bit more picky about separating parts of the equation for correct rendering.



Nicolas DAVID

on **July 27, 2014 at 6:58 pm** said:

Thank you ! THANK YOU ! this is an awesome website you made, and while the subjects are technical, they remain accessible... thanks again !



PrudliArca

on **August 21, 2014 at 6:55 pm** said:

Thank you sooo much for this explanation! I searched for good ones in the web for nearly a month and this is far the best and most detailed I read. Now I can implement Quaternion rotation without shaming me for not knowing the stuff I am using...



lexluger0

on **September 6, 2014 at 11:28 pm** said:

A very very very good explanation on quaternions. This is awesome. Well done 😊



Daniel Huber

on **February 25, 2015 at 10:57 pm** said:

Many thanks for this extraordinary introduction into quaternions.
I really appreciate this work!
Thank you.



Matt
on **March 11, 2015 at 1:10 am** said:

You don't know how many sites I went to trying to find good explanations of Quaternions before ending up here. Keep up the magnificent work. =3



Mom, I know quaternion now!
on **September 7, 2015 at 5:26 pm** said:

HANDS DOWN. The best tutorial of quaternion ever. It's step-by-step, ordered from easy to hard, with simple figures.



Stephen
on **March 5, 2016 at 3:05 pm** said:

Hello, I am a little confused about whether i , j and k are all equal to the square root of -1 or whether they are Cartesian unit vectors. To my mind I cannot see how in one step you can multiply a Cartesian unit vector and get -1 and still say their cross product is another unit vector. Are you defining i , j , and k as imaginary numbers or as unit vectors? I don't think mathematically you define the i , j and k to be both imaginary numbers AND unit vectors as surely numbers and vectors are completely different mathematical objects?

I would really appreciate a reply to this question as I otherwise very impressed with the presentation that you have laid out above.



Jeremiah van Oosten



on **February 20, 2018 at 12:52 pm** said:

I'm sorry about the confusion. When the imaginary numbers (i , j , and k) are shown using emphasis then they represent an imaginary number (along a particular imaginary axis). When they are shown in bold (**i** , **j** , and **k**) then they represent the basis vectors in the cartesian plane (which have properties similar to the imaginary numbers). This is similar to how the real numbers (x , y , and z) are used to represent the components of a 3D vector while the basis vectors are represented as **x** , **y** , and **z** using a boldface font.

I hope this answers you question.



Jens

on **March 31, 2016 at 5:57 am** said:

Why would anyone use the that formula for SLERP which divides by $\sin(\theta)$?!

$((1 - t) q_1 + t q_2) / \sqrt{1 + 2 (q_1 \cdot q_2 - 1) (1 - t) t}$ also yields spherical interpolation, and with this, division by 0 happens only when $q_1 = -q_2$ where spherical interpolation isn't well-defined, because there is no unique path from the north pole to the south pole which makes more sense than any other.



Jeremiah van Oosten

on **July 7, 2016 at 3:11 pm** said:

Jens,

Can you provide a reference to your SLERP equation that shows a proof of the equation you are providing?

The SLERP method that uses $\sin \theta$ provides a method of spherical interpolation that maintains a constant angular velocity while rotating between q_1 and q_2 . Note that it does not provide a method to maintain linear velocity over the curve. If the

quaternions are not normalized, you may notice some “ease-in” and “ease-out” effects. But this is not a problem if the quaternions are normalized.

Perhaps your formula can better achieve constant linear velocity while allowing for non-normalized quaternions?



David Morley

on **July 3, 2016 at 4:42 am** said:

As of this time and user, your latex isn't working.



Jeremiah van Oosten

on **July 7, 2016 at 10:19 am** said:

David,

Thank you for pointing this out. I recently moved 3dgep.com to a new server and the **latex** and **dvipng** binaries were not installed on the new server. I have since resolved the issue and confirmed that the equations are showing up now.

I am looking into using a different plugin (MathJax) to have the client browser generate the equations instead of relying on a server side script to generate gifs for all of the equations.

Please let me know if you find any other cases where equations are not being rendered correctly.



Simon

on **November 17, 2016 at 8:25 pm** said:

Found this great quote for visualization of quaternions. "Imagine you have a small lump of clay and a load of tooth picks. Slide a tooth pick

through the clay on each of its individual axis, X, Y and Z. Rotate the tooth picks, notice how the clay is being rotated on one of its axis at a time. This is how normal Euler axis work, and how most people understand rotation. Now, grab another tooth pick and jam it into the clay at any angle you want! Rotate the new tooth pick on its own axis, notice it isn't constrained to the x, y or z axis but is rotating on its own 'new' and unique axis. This is a quaternion, whenever you are reading up on them from now on, think of this example and hopefully it should make a little more sense."



Yang, Guanqun
on **March 9, 2017 at 2:52 pm** said:

Your article probe deep into quaternion, which remains a mystery for a long time for me. But I have two questions.

1) In the second example of the "rotations" part. It seems that the vector $p = [2, 0, 0]$ revolves 60 degrees instead of 90 since the angle between p and v is 60. Therefore, the deduction of the "rotor" in quaternion form seems to be incorrect.

2) I try to get a formula of rotors in quaternion form and I get one. But somehow I could not get the correct result with $p = [2, 0, 0]$ and $v = [\sqrt{2}/2, 0, \sqrt{2}/2]$. I calculated for more than 10 times, the formula seems to be correct.

(Here I ignore my uncertainties I talked about in 1))

The formula I get is:

$$q * p * \text{inverse}(p) = p + \sin(\theta) * \text{cross}(v, p) - \sin(\theta / 2) * \sin(\theta / 2) * \text{cross}(\text{cross}(v, p), v)$$



Yang, Guanqun
on **March 9, 2017 at 3:22 pm** said:

And it would be great if we visitors could insert mathematics formula directly into the comment area using LaTeX. Good for other visitors since using plain text to represent formulas are seriously ugly!



Jeremiah van Oosten



on **April 12, 2017 at 12:15 pm** said:

You should be able to add LaTeX formulas using the [math] shortcode. For example **[math]y=x^2[/math]** should produce



Jeremiah van Oosten

on **December 4, 2017 at 2:26 pm** said:

I've added MathJax to the site. So using $\backslash($ and $\backslash)$ for inline equations, $\backslash[$ and $\backslash]$ for display equations should work.

For example:

$\backslash\displaystyle\pi(x) = \sum_{n = 1}^{\infty} \frac{\mu(n)}{n}$

$J(\sqrt[n]{x})$

Should produce:

$$\pi(x) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} J(\sqrt[n]{x})$$



alex schwarzenberg-czerny

on **March 11, 2017 at 10:25 pm** said:

Hi,

while your formulae are correct, an explanation is missing why in $qp=(q_0p_0-q_1p_1-q_2p_2-q_3p_3, \dots)$ 3 components are subtracted and in $\text{dot}(q,p)=q_0p_0+q_1p_1+q_2p_2+q_3p_3$.

It is so in complex and/or quaternion dot product one argument is taken conjugate before multiplying so eg: $q \text{ conjg}(p) = (\text{dot}(q,p), \dots)$

Pingback: [Rotations – DubiouSoft](#)



Jeremiah van Oosten

on **July 18, 2017 at 2:15 pm** said:

DubiouSoft,

I've read your post and I just want thank you for your very kind words. I'm sincerely flattered.



Sylvia

on **October 15, 2017 at 8:52 pm** said:

Really nitpicking here, but there should also be no backslash at the end of the equation:

is very small then `<img src="https://www.3dgep.com/cgi-bin/mathtex.cgi?\sin\theta"`

Should probably read:

is very small then `<img src="https://www.3dgep.com/cgi-bin/mathtex.cgi?\sin\theta"`

Cheers



Jeremiah van Oosten

on **October 18, 2017 at 8:33 pm** said:

Sylvia,

Thanks for pointing this out. I didn't really notice anything wrong with the equation, but I fixed it anyways according to your comments.



Sylvia

on **October 15, 2017 at 8:55 pm** said:

Oh, it seems my first comment was not posted, so here goes.

Really amazing and detailed write up. Love it. Just a small correction of a missing space Δq , that lets Latex not displaying this equation properly

Cheers



Jeremiah van Oosten

on **October 18, 2017 at 8:36 pm** said:

Another great catch! I've fixed the equation now. Thanks for your keen observations Sylvia.

Also, comments don't appear until they have been approved.



Alan Q. Balkany

on **October 24, 2017 at 10:48 pm** said:

This is the best explanation of quaternions I've ever seen! I've looked at many others. I finally understand that vector rotation with quaternions is just a couple of multiplications, instead of applying a whole matrix to a vector. Thanks Jeremiah!



babala_234

on **November 28, 2017 at 10:41 am** said:

Thank you for the best explanation of quaternions! But I have one question about it: in the "rotation" part, you said the "the vector has been rotated 90° rather than 45°". As the diagram showing, the \mathbf{p} vector is $[2, 0, 0]$ and the \mathbf{p}' vector is $[1, \sqrt{2}, 1]$, so the angle between them is $\cos \theta = \frac{\mathbf{p} \cdot \mathbf{p}'}{|\mathbf{p}||\mathbf{p}'|} = \frac{1}{2}$, that is $\theta=60$ instead of 90. Am I right? Any suggestion are appreciated.



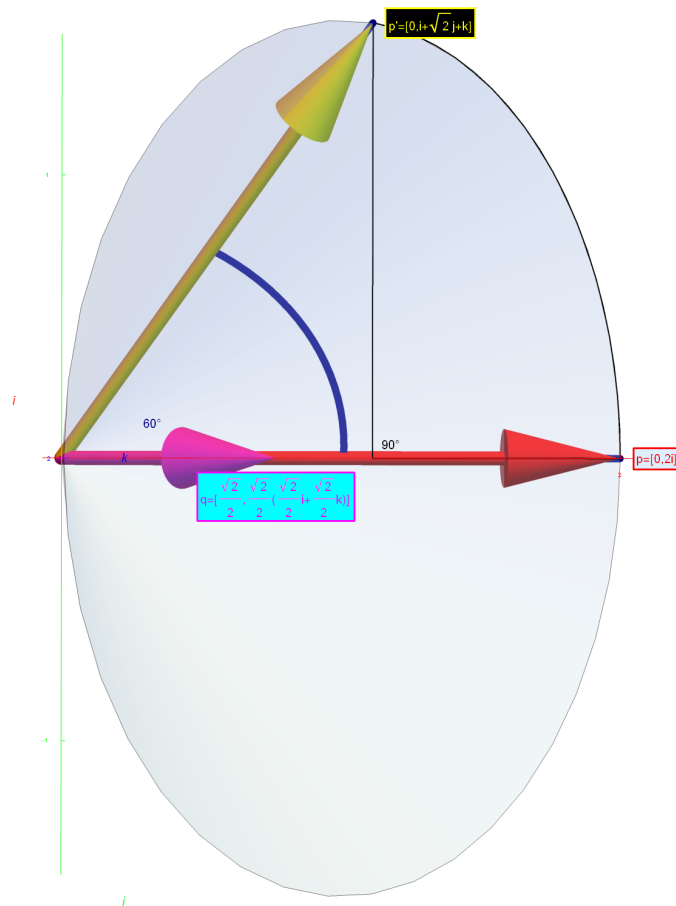
Jeremiah van Oosten



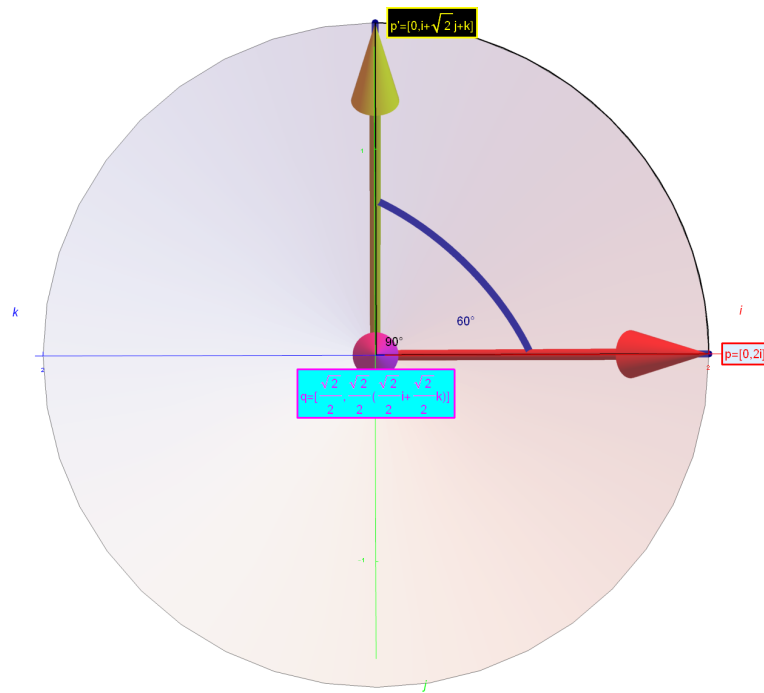
on December 2, 2017 at 11:33 pm said:

Babala,

You are right that the dot product between the two pure quaternions (\mathbf{p} and \mathbf{p}') is 60° but in the example, \mathbf{p} is rotated 90° about the circle that is formed by the quaternion \mathbf{q} . The dot product of the quaternions is similar to that of vectors, it measures the angle formed between the two quaternions. In this case, it's the linear angle between the two vectors \mathbf{p} and \mathbf{p}' but the rotation is measured about the circle that is swept out by the rotation. Perhaps an image will help illustrate this.



The image visualizes the rotation as observed from looking down the k (z) axis. In this image we see that the angle between \mathbf{p} and \mathbf{p}' is 60° but the semicircle swept out by the rotation is 90° . The next image shows it from a different angle.



This image shows the exact same rotation but now we are looking directly at the axis of rotation. Now it can easily be observed that p has been rotated exactly 90° about the quaternion axis.

So the quaternion dot product does not measure the amount of rotation that is applied, but just the angle between the vector parts of the two quaternions.

Keep in mind that if the point being rotated is very close to the axis of rotation, the circle swept by the rotation will be very small. In this case, the dot product between the original point and the rotated point may be very small (almost 0), but the amount the point was rotated could be $\pm 180^\circ$

I hope this answers your question.



Jeremiah van Oosten

on **December 4, 2017 at 2:08 pm** said:

I've added support for MathJax to the site! You can wrap your math equations in `\(` and `\)` tags for inline equations or `\[` and `\]` for display equations.

For example,

Einstein's theory of special relativity is `\(E=mc^2\)`.

And the gaussian integral is:

`\[\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \]`

Should produce:

Einstein's theory of special relativity is $E = mc^2$.

And the gaussian integral is:

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$



John Vince

on **March 20, 2018 at 6:52 pm** said:

Hello, Just to let you know that I like your tutorial and the clarity it brings to the subject.

I am just completing another book Imaginary Mathematics for Computer Science.

Best wishes.



Jeremiah van Oosten

on **March 21, 2018 at 8:48 pm** said:

Thanks John for your wonderful book on Quaternions that has inspired this article. It has help a lot of people (including myself) to better understand how quaternions can be used in graphics applications!

I will definitely be looking forward to your next book!



Joe Tan

on **April 17, 2018 at 2:05 pm** said:

Thanks for your enlightling article, Jeremiah. I benefited a lot from it. But “Negating the scalar and the vector part of the quaternion does not change the orientation that it represents” has confused me at first. Finally, I realized that what you mean might be that a quaternion or its “negative quaternion” could leads to the same rotation result despite the difference of rotation direction. However, I think it is still not appropriate to say that a quaternion and its “negative quaternion” represent the same orientation, cause they actually do have a angular difference of 180° .



Joe Tan

on **April 17, 2018 at 2:07 pm** said:

I got a typo, it is “enlightening”



Andrey

on **September 30, 2018 at 1:13 pm** said:

This is an excellent article. Thank you very much! It helped me understand some of 3D transformations I’ve been struggling with! A very good way to understand quaternions in 3D graphics when applied to simple rotations is as “a vector representing the axis of rotation plus the angle of rotation”. Most of math libraries with quaternions allow you to create a quaternion like that which is very easy to grasp.



Firat

on **October 10, 2018 at 12:04 pm** said:

I love the way you illustrate expressions as clear concepts. Excellent article. I will definitely stick on this website for a long while.



Mathias Glasdam

on **December 11, 2018 at 11:35 pm** said:

Hello, i would like to know why it was imposible for Hamilton to multiply with his 3 term quaternions?



Mike Stark

on **April 28, 2019 at 7:59 pm** said:

Great article for those brave enough to trudge thru the math. I finally got up the courage and glad that i did. Thanks

Small typo: I'm guessing 'Were, according to Hamilton's famous expression:'

should be 'Where, according to Hamilton's famous expression:'



Jeremiah

on **May 10, 2019 at 2:00 pm** said:

Mike,

Thanks for pointing this out! I've corrected the text 😊



baris

on **May 20, 2019 at 1:53 pm** said:

Hey! First of all thanks for very nice article it is really helpful.

I would like to point out an ambiguity in notations. In rotations section, q^{-1} is actually the conjugate if I am not mistaken, not the inverse. Thus, it creates an ambiguity with the definitions made before and other articles about the topic. I would humbly suggest changing the rotation formulation to qpq^* . Again ofc. if I am not mistaken:)

Best

Barış



Jeremiah

on May 28, 2019 at 11:11 pm said:

Baris,

I checked several sources for quaternion notation and this is what I found:

- A. Quaternions for Computer Graphics (Vince, 2011) (the primary reference for this article) uses q^* to denote quaternion conjugate and q^{-1} to denote quaternion inverse.
- B. Quaternion – Wikipedia (https://en.wikipedia.org/wiki/Quaternion#Conjugation,_the_norm,_and_reciprocal) uses q^* , \bar{q} , q^t , or \tilde{q} to denote quaternion conjugate and also points out that the conjugate of a quaternion is also its own inverse (conjugating an element twice returns the original element).
- C. Mathematics for 3D Game Programming & Computer Graphics (Lengyel, 2012) uses \bar{q} to denote quaternion conjugate and q^{-1} to denote quaternion inverse.

So from these references, I think it is safe to say that there is no ambiguity using q^* to denote the quaternion conjugate. Perhaps your confusion comes from the fact that the unit norm quaternion conjugate is the inverse. This is equivalent to the fact that the transpose of an orthonormalized matrix is its inverse.



Barış

on May 29, 2019 at 9:09 am said:

Thanks for the detailed reply!

Yeah you are on spot about my source of confusion:)

We are on the same page about q^* by the way. It was a bit confusing at rotation formula only, where it is qpq^{-1} but since your explanation everything is clear

now thanks!

Best.



Mohammed

on **July 14, 2019 at 9:16 pm** said:

Thanks for this very nice tutorial.



Ryan

on **December 17, 2019 at 12:20 pm** said:

I have enjoyed all your articles and this is no exception. I would like to add with this link below in response to “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 .”

I feel like it gives a good history behind complex numbers and because of it, I always replace imaginative with lateral when I read i .

<https://www.youtube.com/watch?v=T647CGsuOVU>

Best regards

Ryan



Jeremiah

on **February 21, 2020 at 12:30 pm** said:

Ryan,

Thanks for the link!



zhang

on **April 16, 2020 at 4:13 am** said:

thinks for you job; it give me deep understanding of Quaternion and Matric.



John Piliounis

on **April 27, 2020 at 12:21 am** said:

Dear Jeremiah Hello.

I'm not doing graphics but firmware for embedded ARM Cortex-Mn designs and C++/CUDA programming for large datasets processing. Looking for a solution to a problem irrelevant to my work but very relevant to some research that I'm doing I located your site and as a Physicist I was intrigued by this quaternion page and the Understanding Quaternions example that I downloaded and run. I think that this approach may answer to my query, but I don't yet know how. My query goes as following:

We are given a 2D spiral of step let's say a , that lays on the X-Z plane with its center found at $(0,0,0)$. What would be the solid structure emerging by a revolution of that spiral about the X or the Z axis from 0 to π ? I would greatly appreciate if a quaternions based example, like your example above, could be produced for seeing this in action.

Thank you very much.

John



Jeremiah

on **May 14, 2020 at 10:06 pm** said:

John,

Hmm.. This doesn't seem like something I can just whip up in a comment reply. Please consider joining the Discord server:

<https://discord.gg/gsxaxc>

There are community members that might be willing to help you with your research.

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)