Introduction

Rotation is a fundamental transformation of vectors. In 2D the math to express rotations seems to be relatively simple, but in 3D the problems start. One attempt to deal with rotations is to describe them with three angles for three rotations around the three principal axes. This is rather inelegant and also impractical because trying to interpolate between rotations can result in gimbal lock. A better approach is to describe rotation as a single arbitrary axis and an angle. These are then often used to fill a 3x3 matrix with the right values. This is an improvement but the fact that the number of values needed to describe a rotation is less than 3x3 means that a matrix has many degrees of freedom, which again makes interpolation between two rotations problematic.

The solution often offered to this problem is quaternions. Quaternions describe rotation in 3D with four numbers, which prompted many to write articles, comments and make videos about just how unintuitive to understand these weird 4D-numbers are. Nothing could be further from the truth: in fact, quaternions are a very natural way to encode axis-angle rotations. The fact that there are four numbers should not be any more upsetting than the fact that a 3D axis and an angle are already four numbers as well. In this article I hope to show that quaternions are indeed not mysterious or hard to understand at all.

Complex numbers and quaternions can be understood more deeply in the framework of Geometric Algebra (GA), but the well known rotation formulas can easily be derived even without GA. We will gradually build up our understanding of rotations, complex numbers and quaternions until we have derived the apparently impenetrable half-angle sandwich product. After that I will briefly touch upon how the apparent weirdness and discrepancy between 2D and 3D that we will find can be understood in the context of GA.

We will only consider the case of rotations about the origin here. Rotating about arbitrary points and also doing translations can be done elegantly with a projective approach like dual quaternions or projective GA (PGA).

Complex numbers

Complex numbers are numbers of the form $z = a + ib$ where $i^2 := -1$. Hence the product of two complex numbers is: $zw = (a + ib)(c + id) = (ac - bd) + i(ad + bc)$. We denote the conjugate of z with $z^* := a - ib$ and the squared norm with $||z||^2 := zz^* = a^2 + b^2$.

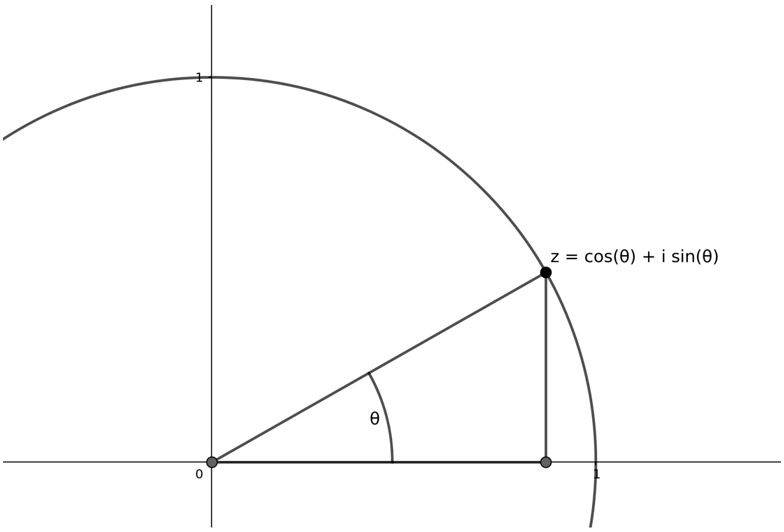
The complex numbers with $||z|| = 1$ form a unit circle with $z = \cos(\theta) + i\sin(\theta)$. A general complex number with $||z|| = r$ and angle θ can be written as $z = r(cos(\theta) + i\sin(\theta))$. The product of two complex numbers in this polar form is $r_1(\cos(\theta_1) + i\sin(\theta_1))r_2(\cos(\theta_2) + i\sin(\theta_2))$. We will show that this equals $r_1r_2(\cos(\theta_1 + \theta_2) + i\sin(\theta_1 + \theta_2))$, i.e. angles are added and magnitudes are multiplied.

Finally there is an inverse $z^{-1} := \frac{z^*}{zz^*}$ because $zz^{-1} = z^{-1}z = 1$.

Complex numbers are often plotted in a 2D-coordinate system with the scalar component in the x and the i-component in the y axis, as in the figure below.

2D Rotation

Rotation is fundamentally a 2D transformation. The components of vectors that lie in a plane are rotated in that plane, the orthogonal component is unaffected. In the 2D case there is only one plane and all vectors lie in that plane completely with no orthogonal component. This is why it is the simplest case to consider and hence why we deal with it first. It will provide the understanding we need for the 3D (or general) case.

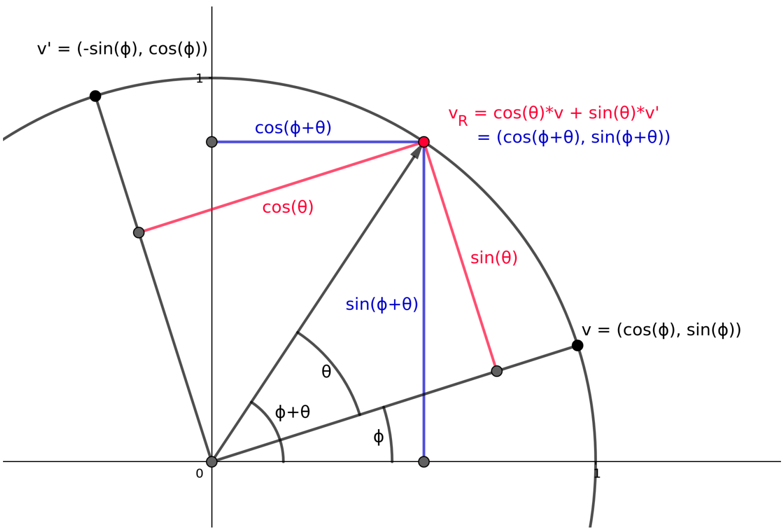


Let $\mathbf{v} = (\cos(\phi), \sin(\phi))$ be any unit vector and let $\mathbf{v_R}$ be this vector rotated by angle θ, so $\mathbf{v_R} = (\cos(\phi + \theta), \sin(\phi + \theta))$. Further assume that $\mathbf{v}$ is simply given to us and we do not know its angle, so we cannot simply add the angles and recompute the rotated vector.

To rotate $\mathbf{v}$ by θ we first rotate it by 90° to $\mathbf{v}' = (\cos(\phi + 90°), \sin(\phi + 90°)) = (-\sin(\phi), \cos(\phi))$ which effectively gives us two basis vectors ($\mathbf{v}$ and $\mathbf{v}'$) of a coordinate system rotated by ϕ. The rotated vector $\mathbf{v_R}$ should have coordinates $(\cos(\theta), \sin(\theta))$ in the local coordinate system $(\mathbf{v}, \mathbf{v}')$. So to transform $\mathbf{v_R}$ to absolute space we simply multiply the local coordinates by the basis vectors of this coordinate system, yielding

$$\begin{aligned}\mathbf{v_R} &= \cos(\theta)\mathbf{v} + \sin(\theta)\mathbf{v}' \\ &= (\cos(\theta)\cos(\phi) - \sin(\theta)\sin(\phi), \cos(\theta)\sin(\phi) + \sin(\theta)\cos(\phi)) \\ &= (\cos(\phi + \theta), \sin(\phi + \theta)).\end{aligned}$$

We have thus derived the angle addition formula.



Using complex numbers this result can be expressed very succinctly. Let $v = \cos(\phi) + i\sin(\phi)$. Note that multiplication by i is exactly the rotation by 90° that we need to calculate v'. So $v' = -\sin(\phi) + i\cos(\phi) = iv$. Finally we find

$$\begin{aligned} v_R &= \cos(\theta)v + \sin(\theta)v' \\ &= \cos(\theta)v + \sin(\theta)iv \\ &= (\cos(\theta) + i\sin(\theta))v \\ &= Rv \text{ with } R := \cos(\theta) + i\sin(\theta) \end{aligned}$$

Thus we see that multiplying unit complex numbers adds their angles. If we define a function $R(x) := \cos(x) + i\sin(x)$, we see that $R(x)R(y) = R(x+y)$ and also $\frac{d}{dx}R(x) = -\sin(x) + i\ cos(x) = iR(x)$, both of which suggest a connection with the exp-function. It can indeed be shown that $R(x) = e^{ix}$ and though we will skip a rigorous proof we will motivate why this is intuitively true.

A common definition of e^x is $\lim_{n\to\infty}(1+\frac{x}{n})^n$. So for our case $e^{ix} = \lim_{n\to\infty}(1+\frac{ix}{n})^n$. This is a complex number $1 + i\frac{x}{n}$ that is multiplied with itself n times. The angle of this inner number is $\arctan\left(\frac{x}{n}\right)$ and since multiplication adds angles, e^{ix} has the angle $\lim_{n\to\infty} n\arctan\left(\frac{x}{n}\right)$. Because $\arctan(x) \approx x$ for very small x, the n cancels out, giving angle x.

For the squared magnitude we get $||1 + i\frac{x}{n}||^2 = 1^2 + \frac{x^2}{n^2}$ and for $||e^{ix}||^2 = \lim_{n\to\infty}(1 + \frac{x^2}{n^2})^n$. Here $n^2 \to 0$ faster than the exponent can compensate, so the limit (and hence the magnitude) is 1.

Thus we have motivated $e^{ix} = \cos(x)+i\sin(x)$ and a rigorous proof of both limits is not too difficult. Note further that $e^{-ix} = \cos(-x)+i\sin(-x) = \cos(x)-i\sin(x) = e^{ix*}$. This actually gives meaning to the conjugate: it is simply a rotation in the other direction.

Note that rotation in a single plane is commutative and since complex numbers only describe a single plane, it makes sense that complex number multiplication is commutative as well.

Quaternions

Quaternions are the 3D equivalent to complex numbers. They are made up of four numbers (whence the name) but do *not* describe 4D space.

We introduce three elements i, j, k that obey the following rules:

$$\begin{aligned} i^2 = j^2 = k^2 &= -1 \\ jk = -kj &= i \\ ij = -ij &= k \\ ki = -ik &= j \end{aligned}$$

A general quaternion will be denoted as $q = q_0 + q_1 i + q_2 j + q_3 k = q_0 + \mathbf{q_v}$, where $\mathbf{q_v}$ is a quaternionic 3D vector I will call a qvector. The product of two quaternions q and p looks as follows:

$$\begin{aligned} qp &= (q_0 + \mathbf{q_v})(p_0 + \mathbf{p_v}) \\ &= q_0p_0 + q_0\mathbf{p_v} + p_0\mathbf{q_v} + \mathbf{q_v}\mathbf{p_v} \end{aligned}$$

where

$$\begin{aligned} \mathbf{q_v}\mathbf{p_v} &= -q_1p_1 - q_2p_2 - q_3p_3 + i(q_2p_3 - q_3p_2) + j(q_3p_1 - q_1p_3) + k(q_1p_2 - q_2p_1) \\ &= -\mathbf{q_v}\cdot\mathbf{p_v} + \mathbf{q_v}\times\mathbf{p_v} \end{aligned}$$

Where $\cdot$ and $\times$ are the familiar dot- and cross-product. The conjugate we will denote with $q^* := q_0 - \mathbf{q_v}$ and the squared norm with $||q||^2 := qq^* = q_0^2 + q_1^2 + q_2^2 + q_3^2$. For the conjugate of a product we find:

$$\begin{aligned}(qp)^* &= ((q_0 + \mathbf{q_v})(p_0 + \mathbf{p_v}))^* \\ &= (q_0p_0 + q_0\mathbf{p_v} + p_0\mathbf{q_v} - \mathbf{q_v} \cdot \mathbf{p_v} + \mathbf{q_v} \times \mathbf{p_v})^* \\ &= q_0p_0 - q_0\mathbf{p_v} - p_0\mathbf{q_v} - \mathbf{q_v} \cdot \mathbf{p_v} - \mathbf{q_v} \times \mathbf{p_v} \\ &= p_0q_0 - p_0\mathbf{q_v} - q_0\mathbf{p_v} - \mathbf{p_v} \cdot \mathbf{q_v} + \mathbf{p_v} \times \mathbf{q_v} \\ &= (p_0 - \mathbf{p_v})(q_0 - \mathbf{q_v}) \\ &= p^*q^*\end{aligned}$$

Note that if $\mathbf{u} \perp \mathbf{v}$ then $\mathbf{uv} = \mathbf{u} \times \mathbf{v}$ and if $\mathbf{u} \parallel \mathbf{v}$ then $\mathbf{uv} = -\mathbf{u} \cdot \mathbf{v}$. In particular $\mathbf{v}^2 = -||\mathbf{v}||^2$ so any unit qvector squares to -1. This implies $e^{\mathbf{v}x} = \cos(x) + \mathbf{v}\sin(x)$ for any unit qvector. Note also that the quaternion product is the sum of a commutative and anti-commutative part. Hence it is not generally one or the other, but can be if one of those parts is zero.

As with the complex numbers the inverse is $q^{-1} := \frac{q^*}{qq^*}$.

3D Rotation

Because there is not only a single plane of rotation in 3D (in fact there are infinite), we lose two properties that made 2D rotation very simple: rotation is no longer commutative if multiple planes of rotation are involved, and not all vectors lie in the plane of rotation.

As before we want to rotate a vector $\mathbf{v}$ by angle θ, but in 3D we also have to pick a plane of rotation. We do this by using the plane's normal as a qvector, which only works in 3D because only there are planes and vectors dual to each other but it will be acceptable for our purposes. Thus our axis of rotation will be the qvector $\mathbf{r}$.

The special case: $\mathbf{v} \perp \mathbf{r}$

Just like in the 2D case the rotated vector will be $\mathbf{v_R} = \cos(\theta)\mathbf{v} + \sin(\theta)\mathbf{v}'$ but what is $\mathbf{v}'$? As before we need $\mathbf{v}'$ to be rotated by 90° and also lie in the plane of rotation. In other words: $\mathbf{v}'$ has to be orthogonal to both $\mathbf{v}$ and $\mathbf{r}$, but this is exactly what the cross-product does, so $\mathbf{v}' = \mathbf{r} \times \mathbf{v}$. In our special case we required $\mathbf{v} \perp \mathbf{r}$ and hence $\mathbf{r} \times \mathbf{v} = \mathbf{rv}$. Finally we get $\mathbf{v_R} = \cos(\theta)\mathbf{v} + \sin(\theta)\mathbf{v}' = \cos(\theta)\mathbf{v} + \sin(\theta)\mathbf{rv} = (\cos(\theta) + \mathbf{r}\sin(\theta))\mathbf{v} = e^{\mathbf{r}\theta}\mathbf{v}$.

Note that this result looks deceptively similar to complex number rotation, however with one difference: since the cross-produce is anti-commutative we have $\mathbf{rv} = \mathbf{r} \times \mathbf{v} = -(\mathbf{v} \times \mathbf{r}) = -\mathbf{vr}$ and hence $e^{\mathbf{r}\theta}\mathbf{v} = \mathbf{v}e^{-\mathbf{r}\theta}$.

The general case

In general $\mathbf{v}$ will not be orthogonal to $\mathbf{r}$ and we need a different formula, one that only rotates the part of $\mathbf{v}$ that *is* orthogonal to $\mathbf{r}$ and leaves the part parallel to $\mathbf{r}$ unchanged. Let $\mathbf{v} = \mathbf{v}_\perp + \mathbf{v}_\parallel$ with $\mathbf{v}_\perp \perp \mathbf{r}$ and $\mathbf{v}_\parallel \parallel \mathbf{r}$. We want $\mathbf{v_R} = \mathbf{v}_\parallel + e^{\mathbf{r}\theta}\mathbf{v}_\perp$. Note that $\mathbf{v}_\parallel\mathbf{r} = \mathbf{v}_\parallel \cdot \mathbf{r} = \mathbf{r} \cdot \mathbf{v}_\parallel = \mathbf{rv}_\parallel$. Making use of commutativity and anti-commutativity we can finally derive the famous half-angle sandwich formula:

$$\begin{aligned}
\mathbf{v}_\mathbf{R} &= \mathbf{v}_\parallel + e^{\mathbf{r}\theta}\mathbf{v}_\perp \\
&= e^{\mathbf{r}\frac{\theta}{2}} e^{-\mathbf{r}\frac{\theta}{2}} \mathbf{v}_\parallel + e^{\mathbf{r}\frac{\theta}{2}} e^{\mathbf{r}\frac{\theta}{2}} \mathbf{v}_\perp \\
&= e^{\mathbf{r}\frac{\theta}{2}} \mathbf{v}_\parallel e^{-\mathbf{r}\frac{\theta}{2}} + e^{\mathbf{r}\frac{\theta}{2}} \mathbf{v}_\perp e^{-\mathbf{r}\frac{\theta}{2}} \\
&= e^{\mathbf{r}\frac{\theta}{2}} (\mathbf{v}_\parallel + \mathbf{v}_\perp) e^{-\mathbf{r}\frac{\theta}{2}} \\
&= e^{\mathbf{r}\frac{\theta}{2}} \mathbf{v} e^{-\mathbf{r}\frac{\theta}{2}}
\end{aligned}$$

As the last step we now consider the case of combining two rotations. Assume we have two quaternions q and p that represent these two rotations and remember $e^{-\mathbf{r}x} = e^{\mathbf{r}x*}$. Rotating first by q and then by p gives us:

$$\begin{aligned}
p(q\mathbf{v}q^*)p^* &= (pq)\mathbf{v}(q^*p^*) \\
&= (pq)\mathbf{v}(pq)^*
\end{aligned}$$

Where pq is just another quaternion again.

Why?

The rotation formulas for complex numbers and quaternions look rather different. There is no sandwich needed with the complex numbers (in fact it would give the wrong result) and the number of values needed to describe a rotation also seems strange: 2 for 2D but 4 for 3D? Wouldn't we expect similar expressions for any dimension? Yes we would, and the reason that they are so different is that we are deeply confused. We thought we were rotating vectors all along, but neither complex numbers nor quaternions include actual vectors.

But first of all the thing we use to rotate actually looks the same in both cases: $e^{i\theta}$ and $e^{\mathbf{r}\frac{\theta}{2}}$, the only difference being the half angle because with the sandwich product we're effectively rotating twice. We call these things rotors. In 2D they are made up of two numbers, in 3D of four numbers. Why two and four?

Remember that rotation is something that happens in a plane, yet with the complex numbers no plane was mentioned – it seemed to have been implicit – and with quaternions we used an axis $\mathbf{r}$ instead of a plane. The truth is that our "axis" $\mathbf{r}$ *was* a plane all along (in GA it is called a bivector) and the "vector" $\mathbf{v}$ we wanted to rotate was really a plane as well. Similarly the complex number i is really a plane and what we rotated was another rotor.

In 2D we confused vectors with rotors, and in 3D we confused vectors with bivectors (planes). The reasons are clear: a complex rotor is two numbers, just like a 2D-vector, and composing rotors (i.e. rotations) in the single plane that we have should result in nothing more than adding angles, just what we wanted. A quaternionic rotor however is four numbers as we have seen and we cannot identify it with a 3D-vector. The number of elementary planes however is three (xy, yz, zx), the same as the number of axes (x, y, z). This is only the case in 3D, but still rotating a plane with the sandwich product worked, so we were satisfied. Finally the dimension of a rotor is just one more than the dimension of a bivector in any dimension because we're adding a scalar to a bivector. Thus in 2D (with one plane) a rotor is two numbers, in 3D (with three elementary planes) a rotor is four numbers, in 4D (with six elementary planes) a rotor is seven numbers, and so forth.

In GA there are actual vectors that can be rotated by rotors using a sandwich product just like with quaternions. This not only works in 2D and 3D but generalizes to any dimension.