Quaternions and rotating 3D objects

By Henry Hayton

Introduction

Rotating objects is a skill that has come to fruition with the increased use of computer graphics. This has some weird links to other areas of maths (complex numbers and quaternions) which people often struggle to see the use of. We will take a deep dive into how objects and points are rotated without the use of these and the unexpected elegance that is born from involving these concepts.

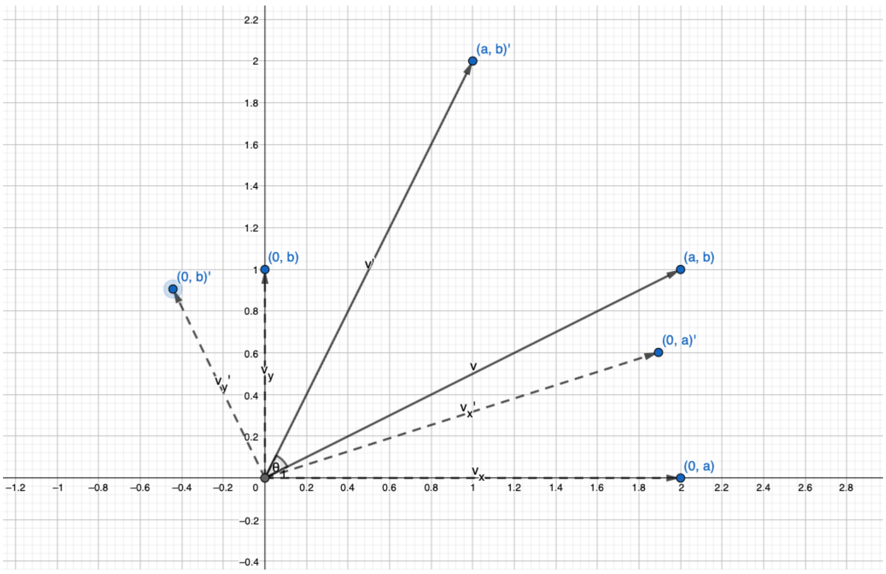
Complex numbers

There are some equations in maths that cannot be solved with conventional integers (1, 2, 3, etc.) for example 2x = 3. To solve this issue, we invented decimal numbers and then for other problems, negative numbers (for example x+1=0). Such numbers whether negative or decimal are "real" numbers although there are still equations we cannot solve, $x^2+1=0$ for example. Any number squared is positive so there is no real number that satisfies this equation. Mathematicians had the genius idea of just defining a number, i, to be √-1 (also notated as $i^2 = -1$). So how is this useful in any way? It has unexpected uses across many areas of maths and physics predominantly due to the use of the complex plane. This is a 2-dimensional coordinate system which expresses complex numbers in the form of $a + bi$ for a, b in real numbers. The a is a normal number on the number line but the bi is some multiple of i. This means that each point on the plane represents a number with the value on the y axis being the imaginary part and the value on the x axis being the real part of the complex number. Although mildly interesting this still seems redundant but for now let's take a brief detour in the land of rotating 2D objects.

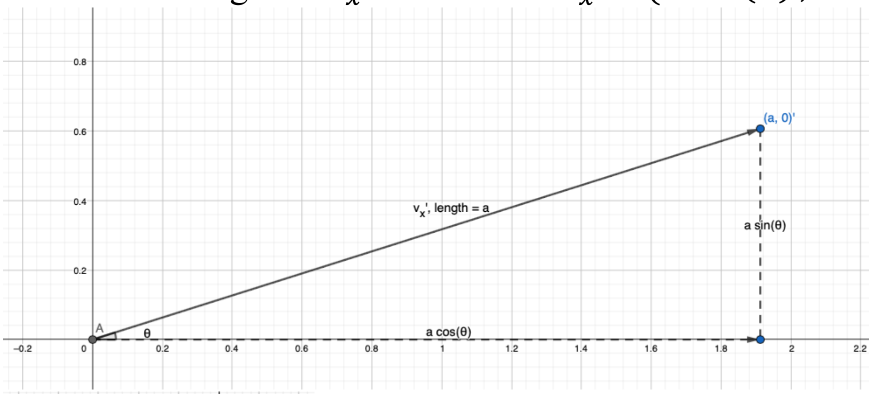
Rotating 2D objects

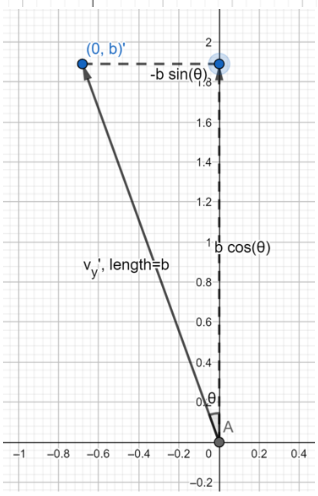
To do this we need to use vectors. Vectors are quantities that have size (like normal numbers) but also direction. We often represent them as arrows between two points in space although are often just represented as the point at which the arrow goes to.

If we consider a vector, $\vec{v} = (a, b)$ and we want to find this vector rotated θ°, called $\vec{v}'$. We can split $\vec{v}$ into two other vectors that add to get $\vec{v}$, ($\vec{v}_x = (a, 0)$, $\vec{v}_y = (0, b)$). These vectors are just the vectors position in each axis separated. We can then consider the rotated versions of these vectors, $\vec{v}_x'$ and $\vec{v}_y'$ as shown on the diagram. Hopefully you can see how $\vec{v}_x' + \vec{v}_y' = \vec{v}'$.



By considering $\vec{v}_x'$ and the vectors that make it up we can form a triangle as shown. Using trigonometry along with the fact that the length of $\vec{v}_x'$ stays the same as the length of $\vec{v}_x$ we find that $\vec{v}_x' = (a\cos(\theta)\,, a\sin(\theta))$.





We can do a similar thing for $\vec{v}_y'$. Considering the triangle made by the rotated vector and the vectors that make it up. We can use similar trigonometry to get that $\vec{v}_y' = (-b\sin(\theta)\,, b\cos(\theta))$ (note that the x co-ordinate is negative as it goes left of the y axis).

Inserting these two values into the previous formula $\vec{v}_x' + \vec{v}_y' = \vec{v}'$ we get $\vec{v}' = (a\cos(\theta), a\sin(\theta)) + (-b\sin(\theta), b\cos(\theta))$ simplifying to $\vec{v}' = (a\cos(\theta) - b\sin(\theta), a\sin(\theta) + b\cos(\theta))$. This is the formula that one can use to rotate objects in 2D. However, if you are like me then you will agree that this is not very elegant; to simplify this we need to come back to the complex world.

Let's consider the product of two complex numbers:
$(a + bi)(c + di) = a * c + a * di + bi * c + bi * di = ac + adi + bci + bi * di$. What is $bi * di$? $bi * di = bd * i * i$. By definition $i^2 = -1$ therefore $bi * di = -bd$ and $(a + bi)(c + di) = ac - bd + (ad + bc)i$. Now reconsidering the previously defined $\vec{v}' = (a\cos(\theta) - b\sin(\theta), a\sin(\theta) + b\cos(\theta))$. We see that this is very similar to the product of complex numbers with $c = \cos(\theta), d = \sin(\theta)$ this means that a rotated vector is just $(a + bi)(\cos(\theta) + \sin(\theta)i)$. However, $(a + bi)$ is just the original vector in complex form. This gives us the formula $\vec{v}' = (\cos(\theta) + i\ sin(\theta))\vec{v}$ we often write $\cos(\theta) + i\ sin(\theta)$ as $e^{i\theta}$ due to Euler's formula (often known to produce the identity $e^{i\pi} + 1 = 0$). Giving us the final form $\vec{v}' = e^{i\theta}\vec{v}$. One immediately useful ramification of this equation is for $\frac{\pi}{2}$ radians (90°); $e^{i\frac{\pi}{2}} = i$ meaning that to multiply a complex number by i is equivalent to rotating by 90° in the complex plane!

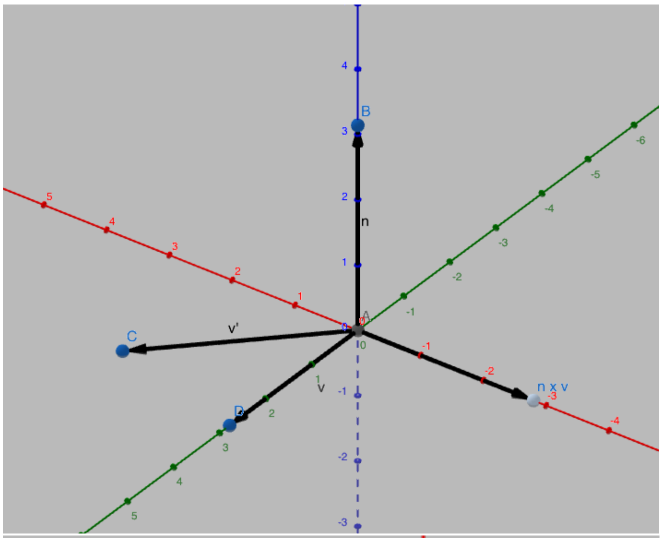
Quaternions

Complex numbers are interesting, but can we go further than complex numbers? Can we extend our number system to 3-dimensions, 4-dimensions? Enter: Quaternions. Quaternions add two more units j and k, these both along with i equal the square root of -1 (this is in a similar way to 2 square roots of 4 being 2 and -2). A form of a quaternion, similar to that of a complex number is $q = a + bi + cj + dk$. We also add the condition that $ijk = -1$, let's explore this equation. If we multiply both sides by i we get that $-jk = -i$ (using $i^2 = -1$) this gives us our first product of these new units $jk = i$. We can continue by multiplying by j; $j^2k = ji$ and using $j^2 = -1$ we find that $ji = -k$. If you continue with this process, you get the surprising result that $ij = k$. Now why is this surprising? Well, ji doesn't equal ij which is like saying that 2*3 is not the same as 3*2. This property is known as commutativity and multiplication using Quaternions doesn't follow this.

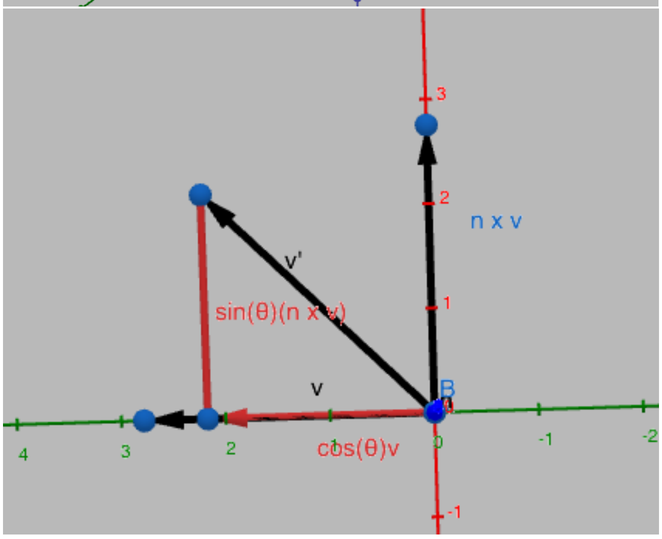
Similarly, to complex numbers this may seem like mathematicians just having fun messing around with weird numbers. Although it is at least somewhat useful when it comes to rotating 3D objects.

Rotating 3D objects

Firstly, why would we want to rotate things in 3D? Well in the modern day we use this in computer graphics to rotate things all the time. It is also used to track the orientation of your mobile phone, but how does it work?
If you think about how to do this you may realise there is an added element that you don't have in 2D, about what axis are we rotating? Unlike in 2D space where you rotate always around the centre, at what angle do we rotate in 3D space?



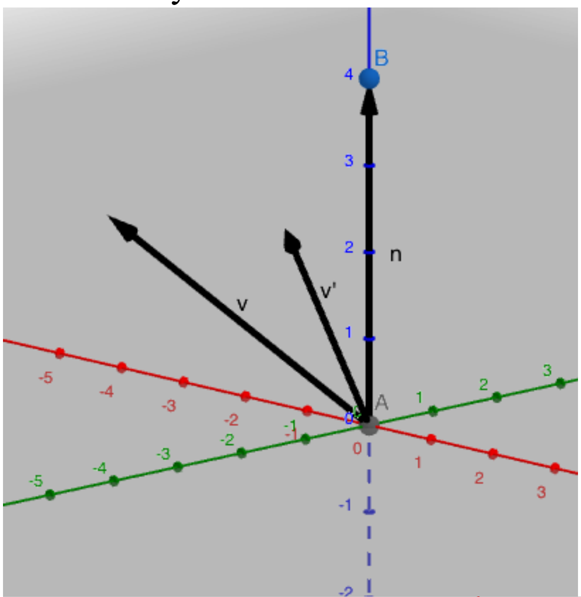
Simplifying the problem, we will assume that the axis of rotation, we will call $\hat{n}$, is perpendicular to the vectors that we are rotating as shown. Using similar naming as in 2D we will call the vector $\vec{v}$ and the rotated vector $\vec{v}'$. We will define another vector which is perpendicular to $\hat{n}$ and $\vec{v}$ called $\hat{n} \times \vec{v}$.

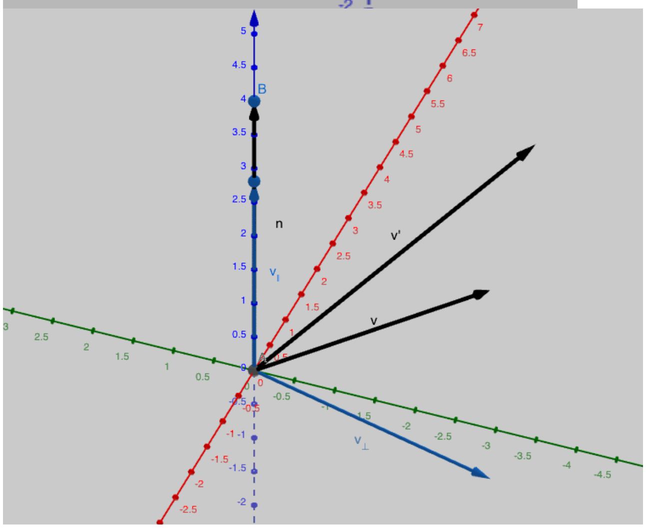


Taking a bird's eye view of this we see a 2-dimensional problem so we can use a similar approach as before. Splitting $\vec{v}'$ into the x and y direction $\vec{v}_x'$, $\vec{v}_y'$. This will add up to $\vec{v}'$ via similar logic as used before. Considering this as forming a triangle we can use trigonometry to find $\vec{v}_y' = \sin(\theta)(\hat{n} \times \vec{v})$ and $\vec{v}_x' = \cos(\theta)\vec{v}$. This is as $\vec{v}_y'$ and $\vec{v}_x'$ are parallel to $\hat{n} \times \vec{v}$ and $\vec{v}$ respectively so, they are just these vectors multiplied by some scaler length which can be easily found with trigonometry. Using $\vec{v}' = \vec{v}_x' + \vec{v}_y'$, $\vec{v}' = \cos(\theta)\vec{v} + \sin(\theta)(\hat{n} \times \vec{v}')$ while this formula is useable it could be much simpler via the use of quaternions.

Quaternions can be thought of as vectors with 4 units instead of the 2 units we are used to. We sometimes represent these vectors with 4 units (or 4D vectors) as a normal number (known as a scaler) followed by a 3D vector.

This is what we will now use, defining a quaternion called v where $v = (0, \vec{v})$. Doing this for the rotated form and the axis of rotation gives us that $v' = (0, \vec{v}')$ and $n = (0, \hat{n})$. We will continue by considering nv; $nv = (0, \vec{v})(0, \hat{n})$. The scaler of nv is 0 as when the vectors are perpendicular it is just the product of the other vectors' scaler parts which is 0. The 3D vector part is just the vector perpendicular to $\vec{v}$ and $\hat{n}$ or our previously defined $\hat{n} \times \vec{v}$. Using nv instead of $\hat{n} \times \vec{v}$ in our other formula allows us to do some fun things. This gives us $v' = \cos(\theta)v + \sin(\theta)nv$, factoring out the v gives us $v' = (\cos(\theta) + \sin(\theta)n)v$. If you evaluate n^2 you get that it is in fact equal to -1. This makes it applicable to the version of Euler's formula for complex numbers meaning we can simplify to the lovely $v' = e^{in}v$.





We are still not done as this assumes that $\hat{n}$ is perpendicular to $\vec{v}$. Now that we can use this formula for vectors that are perpendicular to the axis of rotation, we are ready to solve the completely general case. Using the same naming as before we get the diagram shown.

We can split $\vec{v}$ into a vector perpendicular to $\hat{n}$ called $\vec{v}_\perp$ (v perpendicular) and a vector parallel to $\hat{n}$ called $\vec{v}_\parallel$ (v parallel). The rotated vector is the sum of the rotation of these two vectors similarly to previous logic. $\vec{v}_\parallel' = \vec{v}_\parallel$ as it will not move. As $\vec{v}_\perp$ is perpendicular, its rotation follows the previously derived formula $\vec{v}_\perp = \cos(\theta)\vec{v}_\perp + \sin(\theta)(\hat{n} \times \vec{v}_\perp)$. The rotated vector, $\vec{v}'$, is the sum of its composite vectors rotated ($\vec{v}_\perp'$ and $\vec{v}_\parallel'$). Algebraically $\vec{v}' = \vec{v}_\perp' + \vec{v}_\parallel'$, As $\vec{v}_\parallel' = \vec{v}_\parallel$ and $\vec{v}_\perp = \cos(\theta)\vec{v}_\perp + \sin(\theta)(\hat{n} \times \vec{v}_\perp)$ we can rewrite this as $\vec{v}' = \cos(\theta)\vec{v}_\perp + \sin(\theta)(\hat{n} \times \vec{v}_\perp) + \vec{v}_\parallel$. This has a lot of variables that we can't evaluate so we should eliminate the $\vec{v}_\perp$. We can get rid one of the $\vec{v}_\perp$ terms by rearranging $\vec{v} = \vec{v}_\perp + \vec{v}_\parallel$ to get $\vec{v}_\perp = \vec{v}_\parallel - \vec{v}$. We then need to evaluate $\hat{n} \times \vec{v}_\perp$. If we consider $\hat{n} \times \vec{v} =$

$\hat{n} \times (\vec{v}_\perp + \vec{v}_\parallel) = \hat{n} \times \vec{v}_\perp + \hat{n} \times \vec{v}_\parallel$, $\hat{n} \times \vec{v}_\parallel$ is the line perpendicular to both of them but as they are parallel this is just 0. So, we are left with $\hat{n} \times \vec{v}_\perp = \hat{n} \times \vec{v}$. Plugging this along with $\vec{v}_\perp = \vec{v}_\parallel - \vec{v}$ into the formula we get $\vec{v}' = \cos(\theta)(\vec{v} - \vec{v}_\parallel) + \sin(\theta)(\hat{n} \times \vec{v}) + \vec{v}_\parallel$.The only left variable that is not defined as one of the inputs is $\vec{v}_\parallel$. To find this we need to find a vector projected on to the z axis, there is a formula for this which I will not go into here, but I strongly advise you to read into, which gives us $\vec{v}_\parallel = (\vec{v} \cdot \hat{n})\hat{n}$. The dot symbolises the dot product of the two vectors which is just each dimension of the vector multiplied with the corresponding vector. Substituting this into the equation we get $\vec{v}' = \cos(\theta)(\vec{v} - (\vec{v} \cdot \hat{n})\hat{n}) + \sin(\theta)(\hat{n} \times \vec{v}) + (\vec{v} \cdot \hat{n})\hat{n}$. Factoring out the $(\vec{v} \cdot \hat{n})\hat{n}$ giving us $\vec{v}' = (1 - \cos(\theta))((\vec{v} \cdot \hat{n})\hat{n}) + \sin(\theta)(\hat{n} \times \vec{v}) + \cos(\theta)\vec{v}$. This is now a useable formula for rotation in 3D. As we have done so many times before, we can do better.

For this we will use quaternions, instead of defining all of the quaternions we will use just note that a previously defined vector with no arrow is a quaternion (0, vector). We can get a fairly elegant solution with a $v' = v_\parallel + v_\perp'$, using the formula for perpendicular vectors with quaternions ($v' = e^{in}v$) we get that $v' = v_\parallel + e^{in}v_\perp$. This makes intuitive sense as $v_\parallel$ is unchanged due to it being parallel, while $v_\perp$ is being operated on. This formula is elegant although the $v_\parallel$ and $v_\perp$ are not inputs to the equation and have to be calculated separately. We can rewrite $v' = v_\parallel + e^{in}v_\perp$ as $v' = e^{\frac{\theta}{2}n}e^{-\frac{\theta}{2}n}v_\parallel + e^{\frac{\theta}{2}n}e^{\frac{\theta}{2}n}v_\perp$. Hopefully you can see how this is algebraically equivalent. Interestingly, $e^{\frac{\theta}{2}n}e^{-\frac{\theta}{2}n}v_\parallel$ is commutative and can be written as $e^{\frac{\theta}{2}n}v_\parallel e^{-\frac{\theta}{2}n}$ (which we will do). However, $e^{\frac{\theta}{2}n}e^{\frac{\theta}{2}n}v_\perp = e^{\frac{\theta}{2}n}v_\perp e^{-\frac{\theta}{2}n}$. This can be shown using mostly what we have learnt. Using this we get $v' = e^{\frac{\theta}{2}n}v_\parallel e^{-\frac{\theta}{2}n} + e^{\frac{\theta}{2}n}v_\perp e^{-\frac{\theta}{2}n}$ factorising this to get $v' = e^{\frac{\theta}{2}n}(v_\parallel + v_\perp)e^{-\frac{\theta}{2}n}$ (remembering the non-commutativity between quaternions). Finally, you get $v' = e^{\frac{\theta}{2}n}ve^{-\frac{\theta}{2}n}$. If you test some cases for this it makes some sense; when v is parallel to n the left-hand side of the equation is commutative, and you just get $v' = v$. When v is perpendicular to n; for it to commute the $e^{-\frac{\theta}{2}n}$ becomes $e^{\frac{\theta}{2}n}$and you get the familiar equation for perpendicular vectors; $v' = e^{in}v$. These formulae are often used to rotate 3D objects in games and animation, and I hope you understand why they are true algebraically giving an insight into some crazy areas of maths along the way!