

可视化理解四元数，愿你不再掉头发

四元数被广泛应用在计算机图形学领域，游戏引擎Unity也是用四元数在后端计算旋转。数学上，我们可以按部就班地进行演算，可是直觉上一直不知道它究竟如何运作的。今天我就带领大家通过观察四元数，更准确地说是观察四维单位超球面在三维的投影，来对它有个更深入的了解。

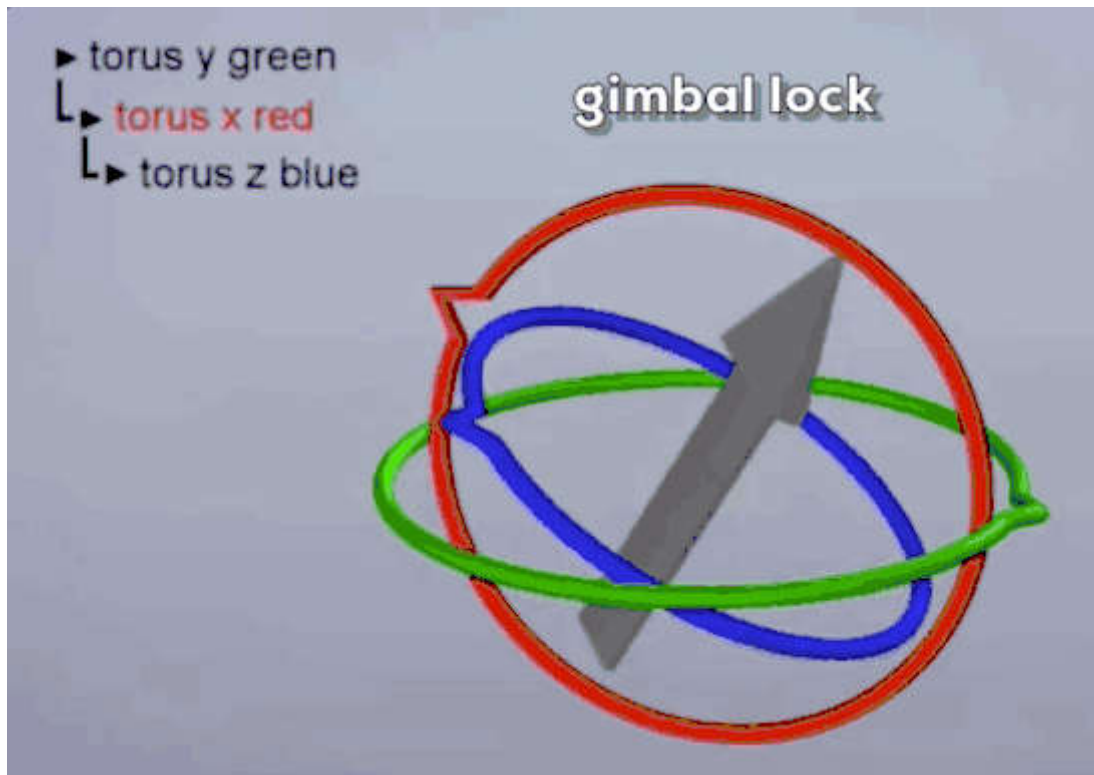
四元数的引出

四元数的一个最主要的应用就是表示旋转，它既是紧凑的，也没有奇异性。而旋转的其他表示方法各有优劣：

旋转矩阵：用九个数来表示三个自由度，矩阵中的每一列表示旋转后的单位向量方向，缺点是有冗余性，不紧凑[1]。

旋转向量：用一个旋转轴和一个旋转角来表示旋转，但是因为周期性，任何 $2n\pi$ 的旋转等价于没有旋转，具有奇异性[2,3]。

欧拉角：将旋转分解为三个分离的转角，常用在飞行器上，但因为万向锁问题（Gimbal Lock）而同样具有奇异性。

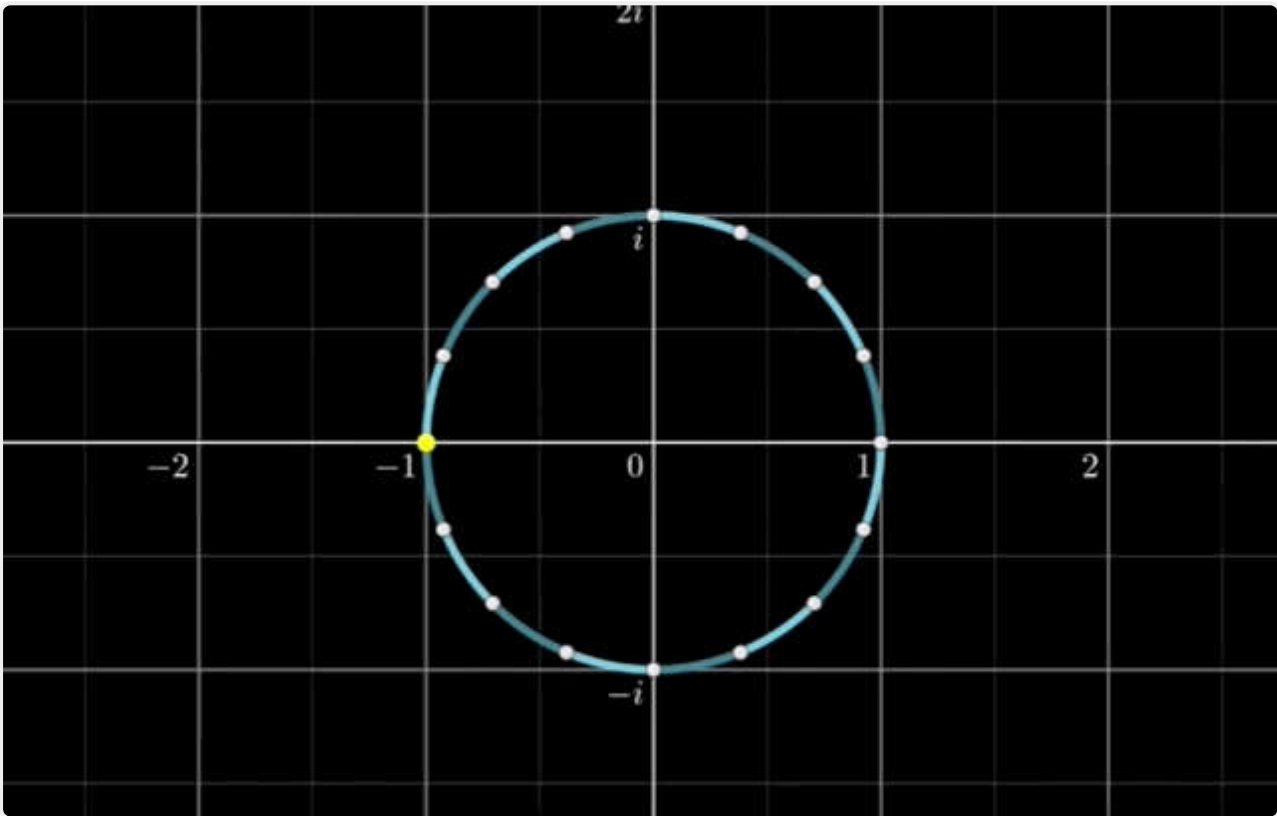


万象锁问题，一三轴重合后损失一个自由度

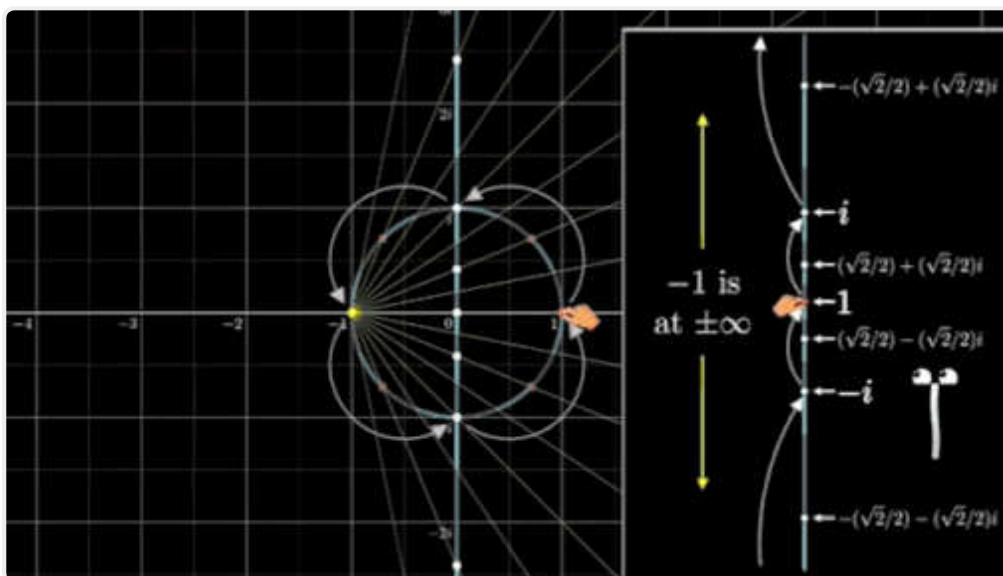
四元数之所以难以理解，是因为它是一个四维的表示。不过也不是没有办法，因为我们通常用单位四元数来表示旋转，所以我们只需要关注四维中的单位超球面 (unit hypersphere)，然后就可以较为轻松地获得它在三维的球极平面投影 (stereographic projection)。

单位圆在一维空间的投影

为了更好地理解四维单位超球面在三维空间的投影，我们先来看一看二维单位圆是怎么投影到一维空间的。



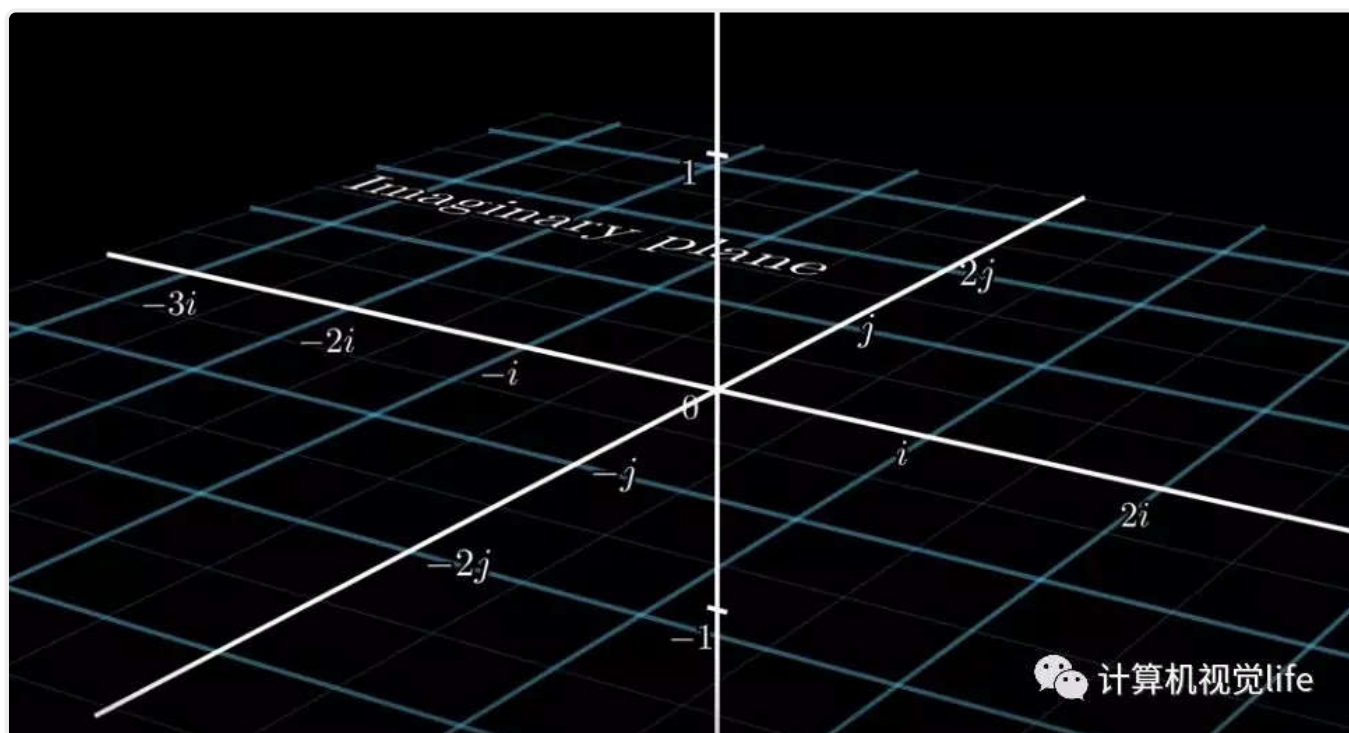
在复数平面内，对于每一个在单位圆上的点，画一条线将 -1 点与这个点相连。连线将与虚数轴交于一点，此交点就是投影点。从上图中可以看到，1 投影在一维 0 处，i 和 -i 投影后不发生变化，而 -1 投影到了正负无穷远处。这里需要注意的是，此处的投影仅仅只是二维空间中单位圆的一个投影，二维空间中的其他点是没有办法用一维来表示的。



我们可以在左边的单位圆中清楚地观察到乘以 i 对应着一个90度的旋转，与之对应的，投影在一维坐标轴上的点也在进行着移动，1 变成 i ， i 变成 -1 ， -1 变成 $-i$ ， $-i$ 变成 1 ，这与复数的乘法定义相吻合。就这样，二维空间中单位圆的纯旋转由一个维度表示清楚。

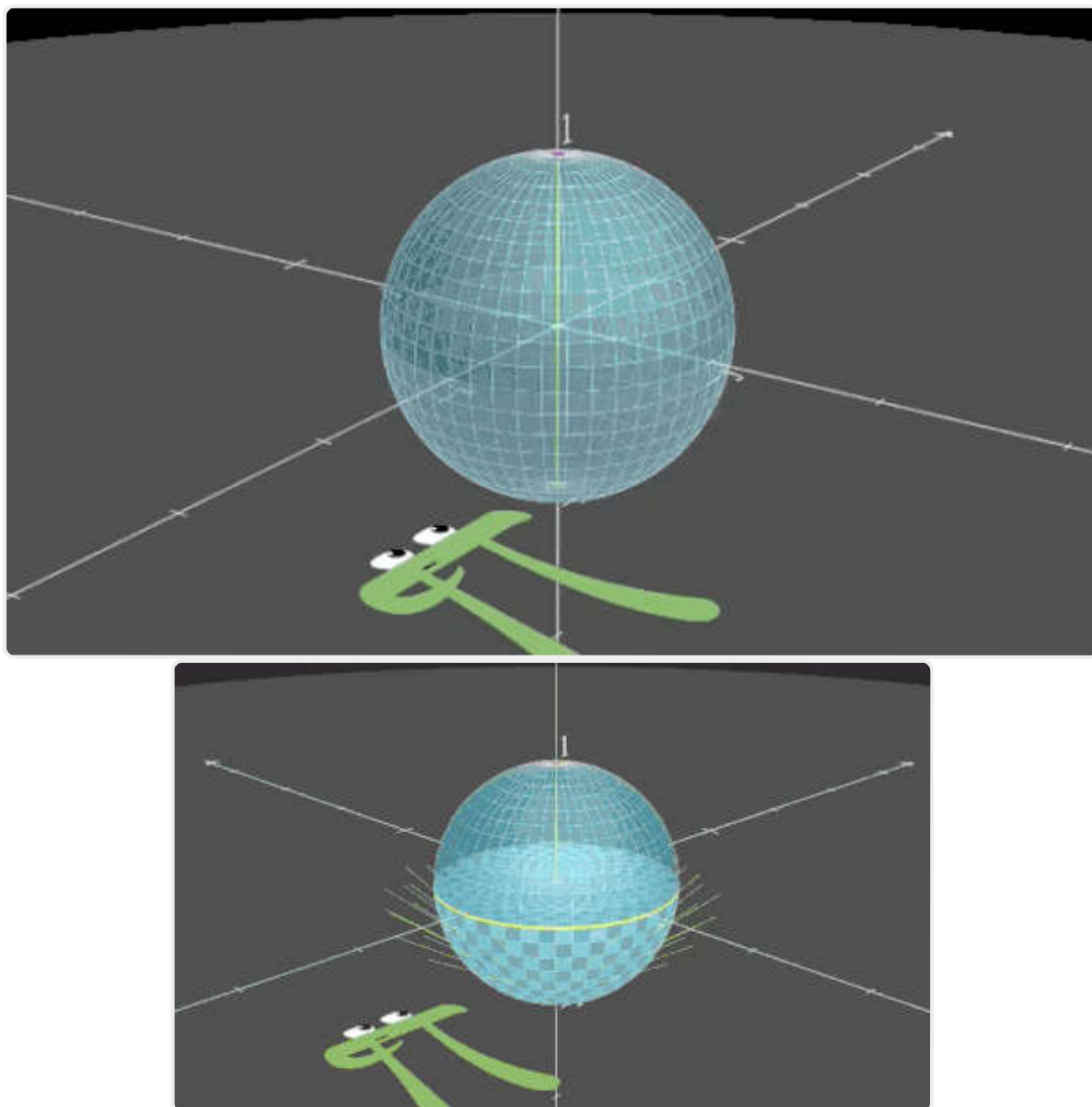
单位球面在二维空间的投影

现在想象我们如何将三维空间的纯旋转解释给二维的生物。首先我们需要构建一个新的坐标系，在这个坐标系中， i 轴和 j 轴形成一个平面，而实数轴与 z 轴对齐。

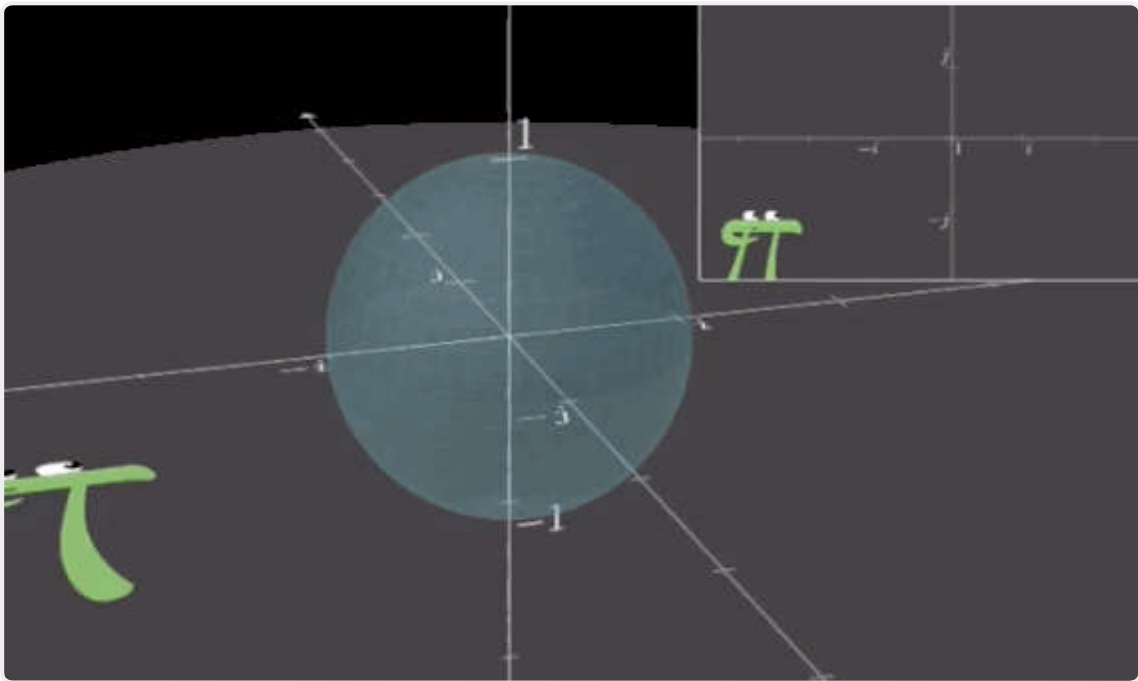


需要注意的是，这里的坐标系仅仅只是为了让概念可视化， i 和 j 并不像复数和四元数那样有良好的乘法定义。和之前的二维投影相似，我们可以用球极平面投影来描述三维的旋转。对于

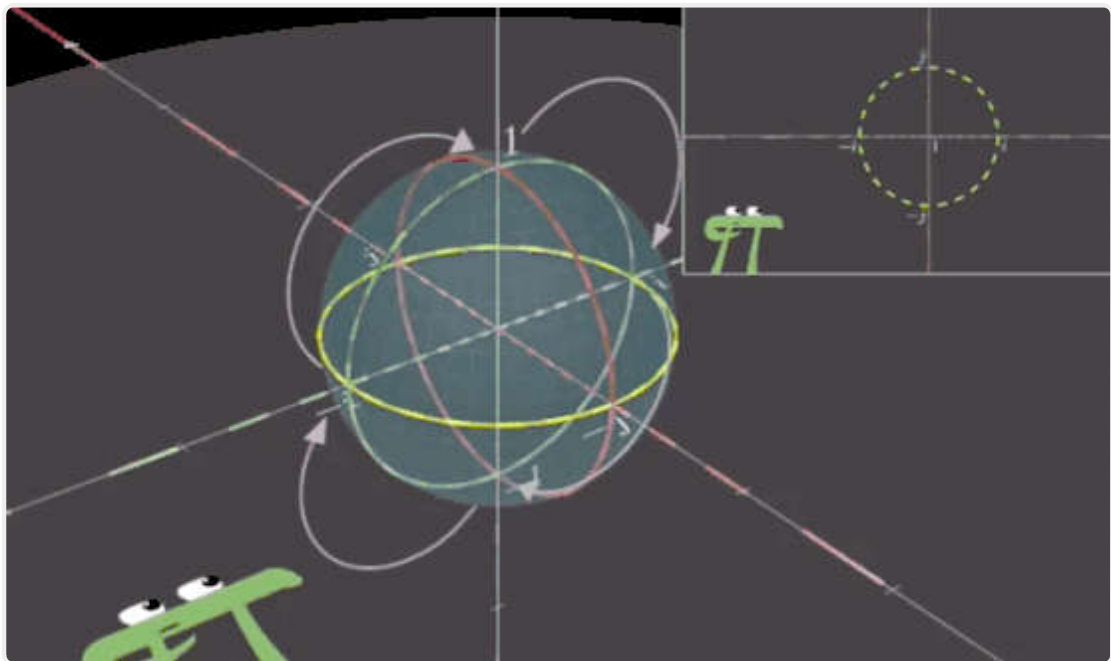
每一个单位球面上的点，我们都把它与 -1 点相连，这条线与 ij 平面的交点即是二维的投影点。



如上图所示，实数轴上的 1 会投影在平面的原点，北半球上的点会投影在 ij 平面的单位圆内，而南半球上的点会投影在单位圆之外，且任意方向的无穷远处都会是 -1 的投影。这里的单位圆是投影之后唯一不被扭曲的，位于单位球上的点。

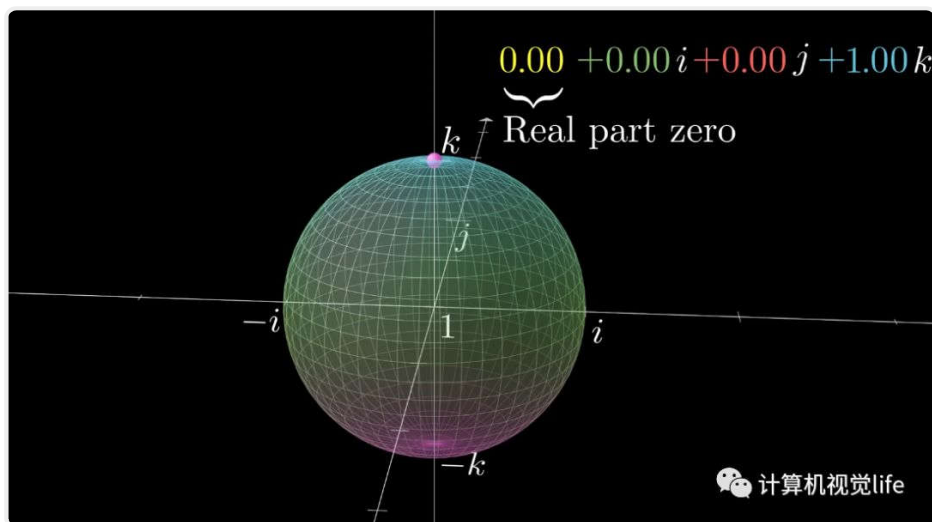


为了方便理解，我们首先关注几条参考线。经过 -1 且与 i 轴平行的圆投影在二维平面的 i 轴上，同理，与 j 轴平行的圆投影在 j 轴上。事实上，任意一个经过 -1 的圆投影到二维平面上都会是一条直线（经过 -1 等价于直线延伸到无穷远处）。当球面在三维空间中旋转时，延 i 轴或 j 轴的旋转都会使垂直方向上投影的线变成圆形，投影的圆形变成线。延实数轴的旋转投影在二维平面上仍然是一个二维的旋转。



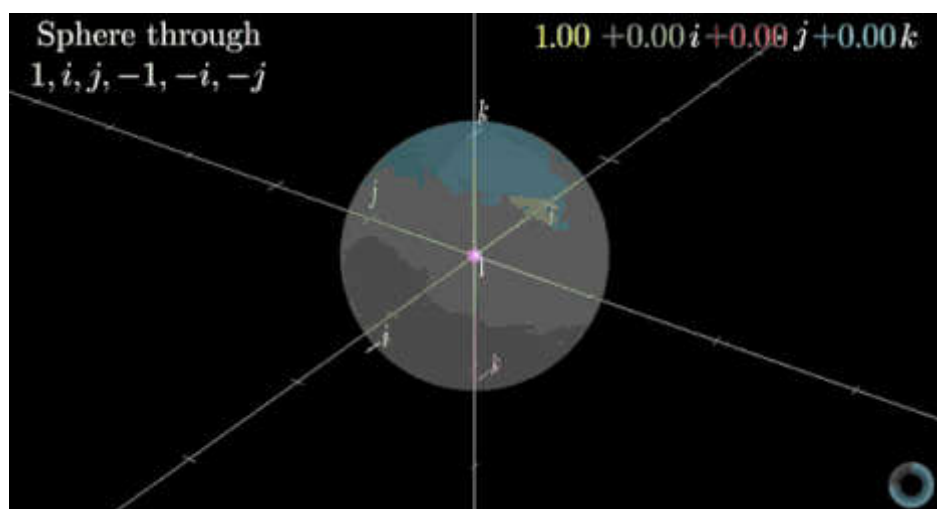
单位超球面在三维空间的投影

如同复数的定义，四元数由一个实数项和三个虚数项构成。对 q_2 左乘一个四元数 q_1 ，其作用是将 q_2 拉伸 q_1 的模长，再作用一个特殊的四维旋转。因为我们用单位四元数来表示三维空间中的旋转，所以在这里不再考虑拉伸。



对四维空间中的单位球面进行球极平面投影，实数轴的 1 投影到 ijk 坐标系的原点。如同一维中的 i, j 点，二维中的单位圆，当四维超球面投影到三维空间时，与三维空间交于一个位置不变的三维单位球面，而这个球面对应纯四元数，也就是实数部分为零。实数部分介于 0 到 1 之间的投影在了这个三维球面的里面，而实数部分小于 0 的投影在了三维球面以外，-1 投影在了各个方向的无穷远处。

正如同三维中的圆投影到二维平面中是一条线一样，四维中的球（不是超球）投影在三维是一个平面，事实上，三维投影中的平面都是四维超球中过 -1 的球面在三维的投影。

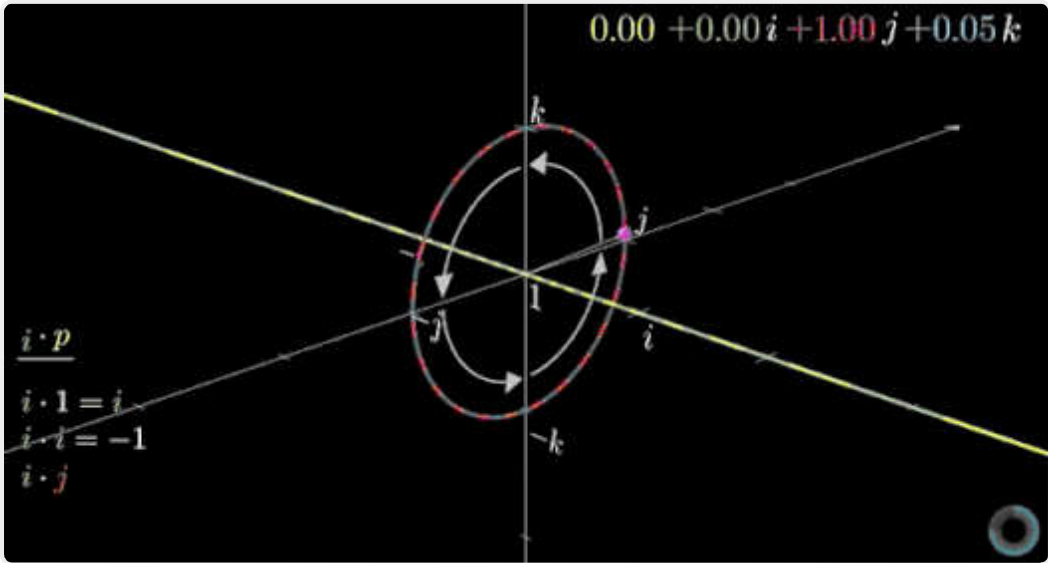


画一些参考线并观察，我们得出了整篇文章一个非常重要的结论：乘上一个单位四元数可以看成是三维空间内两个垂直且同步的二维旋转。

单位四元数乘法的可视化

讲了这么多，我们终于可以用单位四元数在三维的投影来理解四元数啦！首先是四元数的乘法。什么叫做两个垂直且同步的二维旋转呢？如下图所示，正在变化的 i 轴描述了一个二维的旋转，而 jk 平面里的圆同样描述了一个二维的旋转。对于四维空间的生物来说，他们观察到

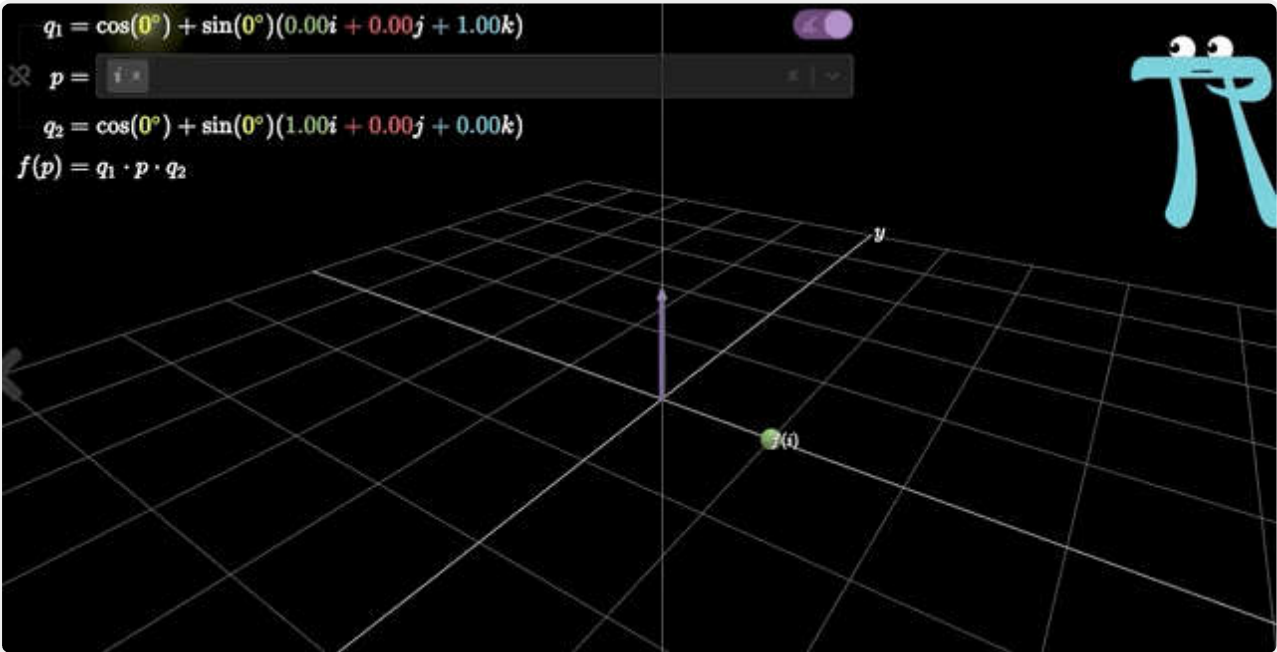
的是一个四维刚体的纯旋转，但在我们看来，只能把它分解为两个二维的旋转，它们因基坐标的定义而互相垂直，因描述同一个四维旋转而同步。



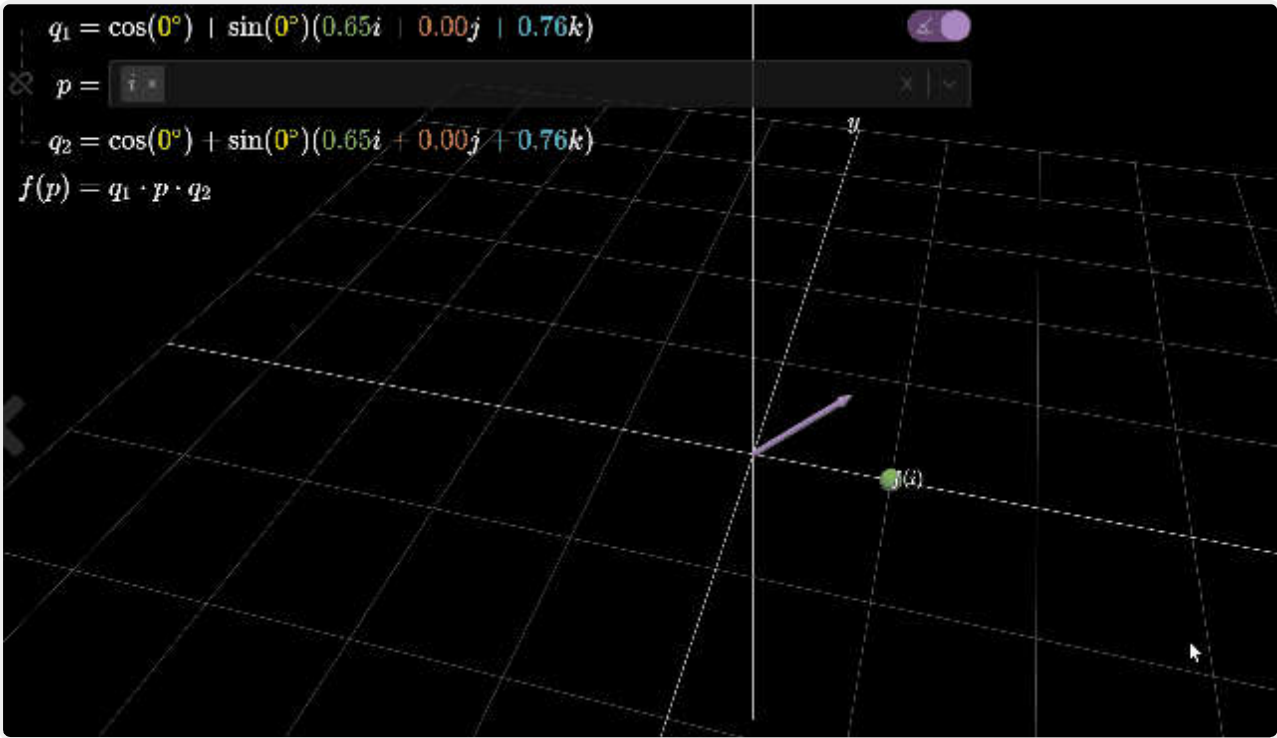
于是乎，我们可以观察到几乎所有的四元数乘法定则。左乘一个 i 相当于两个圆都转了90度，1 变成 i, i 变成 -1, j 变成 k, k 变成 -j, -j 变成 -k, -k 变成 j。

单位四元数表示三维旋转的可视化

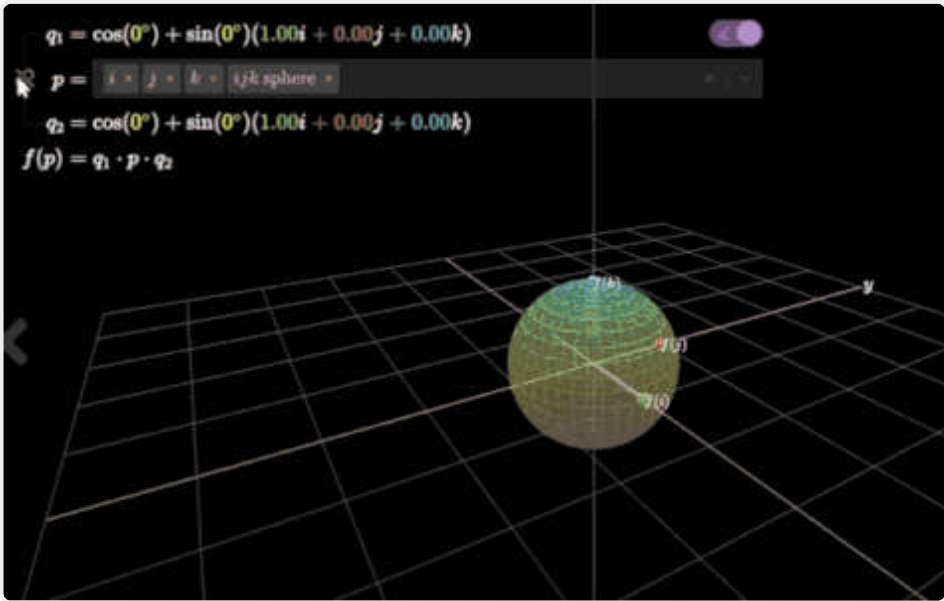
这里简单解释一下为什么我们用 qpq' 的方法表示三维旋转，其中 p 是表示三维空间向量的纯四元数。首先，如果待旋转的向量 p 与单位四元数虚部所表示的旋转轴正交时，我们只需要左乘单位四元数就可以表示此旋转。如下图所示，i 点绕 k 轴进行旋转，左乘单位四元数 q 即可。



可是对于大部分情况，向量 p 和旋转轴并不正交，单单只是左乘单位四元数 q 会在三维空间中有拉伸的效果。注意整个三维空间都是单位超球面的投影，这里的拉伸只不过是投影之后的一种效果罢了。我们用四元数并不关心四维单位超球面的旋转，而是拿它来描述三维空间的旋转，于是我们必须找到一种不带拉伸效果的四元数运算。人们发现，当右乘 q 的逆时，同方向的旋转会继续进行，而拉伸的效果互相抵消，于是就有了 qpq' 的表示方法。如果表示一个绕 u 轴 θ 度角的旋转， q 则表示为 $[\cos(\theta/2), \sin(\theta/2)u]$ 。



对于详细的 qpq' 的推导，请参考 krasjet 编写的一篇文章 [4]。最后一幅图用更多的点来描述这个道理，左乘单位四元数 q 会有拉伸效果，jk 圆逆时针旋转，右乘 q 的逆抵消了拉伸的效果，jk 圆仍然继续逆时针旋转。



结语

到这里，整个四元数的可视化就告一段落啦。在这篇文章中，我们从二维旋转在一维的投影，三维旋转在二维的投影，逐渐进阶到四维旋转在三维的投影来了解四元数。我们观察了乘法法则在三维投影中的规律，并简单解释了四元数为什么要用 qq' 的形式来表示三维旋转。整篇文章受 Youtuber 3B1B 的启发，强烈推荐大家戳链接访问这个可以互动的教学视频 (<https://eater.net/quaternions/>)

参考资料

- [1] 高翔，张涛，《视觉SLAM 十四讲》
- [2] J. Stuelpnagel, "On the Parametrization of the Three-Dimensional Rotation Group," 1964.
- [3] F. S. Grassia, "Practical Parameterization of Rotations Using the Exponential Map," 1998.
- [4] Krasjet, "四元数与三维旋转". <https://krasjet.github.io/quaternion/quaternion.pdf>
- [5] <https://www.youtube.com/watch?v=d4EgbgTm0Bg>
- [6] <https://www.youtube.com/watch?v=zjMulxRvygQ>
- [7] <https://eater.net/quaternions/>
- [8] <https://www.youtube.com/watch?v=zc8b2Jo7mno>