

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

$$ij = k \quad \text{and} \quad ji = -k$$

with the cyclic permutation $i \rightarrow j \rightarrow k \rightarrow i$. Such was his relation, Hamilton carved these formulae on the side of the bridge and called the number:

$$q = a + bi + cj + dk$$

a 'quaternion'.

For our purposes we shall use the condensed notation

$$q = (s, v)$$

where:

$$(s, v) = s + v_x i + v_y j + v_z k$$

s is thought of as the scalar part of the quaternion and v the vector part with axes i, j and k . Using the above rules it is easy to derive the following properties. The multiplication of two quaternions:

$$q_1 = (s_1, v_1) \quad \text{and} \quad q_2 = (s_2, v_2)$$

is given by:

$$q_1 q_2 = (s_1 s_2 - v_1 \cdot v_2, s_1 v_2 + s_2 v_1 + v_1 \times v_2)$$

The multiplication of two quaternions is thus a quaternion. Mathematically, we have defined a group. Stated somewhat simplistically, a group is just a set of elements with a rule defining their multiplication such that the result of this multiplication is itself an element of that group. Groups can be constructed completely arbitrarily, though a surprising number of groups are relevant to the physical world. We shall see that a subgroup of the quaternion group is closely related to the group of rotations or, more precisely, the group of rotation matrices.

Note that except for the cross product term at the end of the previous equation, it bears a strong similarity to the law of complex multiplication:

$$(a_1 + ib_1)(a_2 + ib_2) = (a_1 a_2 - b_1 b_2) + i(a_1 b_2 + a_2 b_1)$$

The cross product term has the effect of making quaternion multiplication noncommutative.

We define the conjugate of the quaternion:

$$q = (s, v) \quad \text{to be} \quad \bar{q} = (s, -v)$$

The product of the quaternion with its conjugate defines its magnitude:

$$q\bar{q} = s^2 + |v|^2 = |q|^2$$

Finally, as promised, we come to the point of all this, which is contained in the following properties. Take a pure quaternion (one that has no scalar part):

Quaternions

The great mathematician Sir William Hamilton had been interested in complex numbers since the early 1830s. Complex numbers have the form:

$$a + ib$$

where a and b are real and the multiplication rules are:

$$i^2 = -1 \quad \text{and} \quad j^2 = -1$$

These complex numbers define a plane – the complex plane – where one axis is real and the other imaginary. For over 10 years Hamilton tried to extend this concept in order to define a complex volume by searching for a second imaginary axis. Just such a number would have three components: one real and two imaginary. This, however, he could not do. Then, on 16 October 1843, when walking past Broome Bridge in Dublin towards the Royal Irish Academy, where he was to preside over a meeting, Hamilton, in a flash of inspiration, realized that three rather than two imaginary units were needed, with the following properties:

$$p = (0, r)$$

and a unit quaternion

$$q = (s, v) \quad \text{where } q\bar{q} = 1$$

and define

$$R_q(p) = qpq^{-1}$$

Using our multiplication rule, and the fact that $q^{-1} = \bar{q}$ for q of unit magnitude, this expands to:

$$R_q(p) = (0, (s^2 - v \cdot v)r + 2v(v \cdot r) + 2sv \times r) \quad (15.6)$$

This can be simplified further since q is of unit magnitude and we can write:

$$q = (\cos \theta, \sin \theta n) \quad |n| = 1$$

Substituting into Equation (15.6) gives:

$$\begin{aligned} R_q(p) &= (0, (\cos^2 \theta - \sin^2 \theta)r + 2\sin^2 \theta n(n \cdot r) \\ &\quad + 2\cos \theta \sin \theta n \times r) \\ &= (0, \cos 2\theta r + (1 - \cos 2\theta) n(n \cdot r) \\ &\quad + \sin 2\theta n \times r) \end{aligned} \quad (15.7)$$

Now compare this with Equation (15.5). You will notice that aside from a factor of 2 appearing in the angle they are identical in form. What can we conclude from this? The act of rotating a vector r by an angular displacement (θ, n) is the same as taking this angular displacement, 'lifting' it into quaternion space, by representing it as the unit quaternion $(\cos(\theta/2), \sin(\theta/2) n)$ and performing the operation $q(\cdot)\bar{q}$ on the quaternion $(0, r)$. We could therefore parametrize orientation in terms of the four parameters:

$$\cos(\theta/2), \sin(\theta/2) n_x, \sin(\theta/2) n_y, \sin(\theta/2) n_z$$

using quaternion algebra to manipulate the components.

In practice this would seem an extremely perverse way of going about things were it not for one very important advantage afforded by the quaternion parametrization. Two quaternions multiplied together, each of unit magnitude, will result in a single quaternion of unit magnitude. If we use unit quaternions to represent rotations then this translates to two successive rotations producing a single rotation. Now a variation of Euler's theorem states that two successive rotations is equivalent to one rotation. So we can see that inherent in the algebra of the quaternion group is Euler's theorem. The single steady rotation between successive keyframes that we seek is provided for us automatically by the rules particular to the parametrization and contained in the statement:

$$R_{q''} = R_q R_{q'} \quad \text{where } q'' = qq'$$

Let us now return to our example of Figure 15.17 to see how this works in practice. The first single x -roll of π is represented by the quaternion:

$$(\cos(\pi/2), \sin(\pi/2)(1,0,0)) = (0, (1,0,0))$$

Similarly, a y -roll of π and a z -roll of π are given by $(0, (0,1,0))$ and $(0, (0,0,1))$ respectively. Now the effect of a y -roll of π followed by a z -roll of π can be represented by the single quaternion formed by multiplying these two quaternions together:

$$\begin{aligned} (0, (0,1,0)) (0, (0,0,1)) &= (0, (0,1,0) \times (0,0,1)) \\ &= (0, (1,0,0)) \end{aligned}$$

which is the single x -roll of π . From this we can see that the cross product term in (15.7) can be thought of as correcting for the interdependence of the separate axes that is ignored by Euler's angle notation.

An additional advantage afforded by using quaternions is that the gimbal lock singularity, which is a consequence of using three parameters to parametrize orientation, disappears.

Much of what now follows is based on the work of the researcher who brought quaternions to the attention of the computer graphics community. The interested reader is referred to [SHOE85] and [SHOE87] for further detail. The latter reference concerns itself more with the practical details of an implementation.

Interpolating using quaternions

Given the superiority of quaternion parametrization over Euler angle parametrization, this section covers the issue of interpolating rotation in quaternion space. Consider an animator sitting at a workstation and interactively setting up a sequence of key orientations by whatever method is appropriate. This is usually done with the principal rotation operations, but now the restrictions that were placed on the animator when using Euler angles, namely using a fixed number of principal rotations in a fixed order for each key, can be removed. In general, each key will be represented as a single rotation matrix. This sequence of matrices will then be converted into a sequence of quaternions. Interpolation between key quaternions is performed and this produces a sequence of inbetween quaternions, which are then converted back into rotation matrices. The matrices are then applied to the object. The fact that a quaternion interpolation is being used is transparent to the animator.

Moving into and out of quaternion space

The implementation of such a scheme requires us to move into and out of quaternion space, that is, to go from a general rotation matrix to a quaternion and vice versa. It can be shown that the effect of taking a unit quaternion:

$$q = (\cos(\theta/2), \sin(\theta/2) \mathbf{n})$$

and performing the operation $q(\)q^{-1}$ on a vector is the same as applying the following rotation matrix to that vector:

$$\begin{bmatrix} 1 - 2Y^2 - 2Z^2 & 2XY - 2WZ & 2XZ + 2WY & 0 \\ 2XY + 2WZ & 1 - 2X^2 - 2Z^2 & 2YZ - 2WX & 0 \\ 2XZ - 2WY & 2YZ + 2WX & 1 - 2X^2 - 2Y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the quaternion $(\cos(\theta/2), \sin(\theta/2) \mathbf{n})$ is written as $(W, (X, Y, Z))$, the notation used in Listing 15.3. By these means then, we can move from quaternion space to rotation matrices. Listing 15.3 gives the conversion from quaternion space to rotation matrix in the routine `quatmat(q, mat)`.

The inverse mapping from a rotation matrix to a quaternion is only slightly more involved. All that is required is to convert a general rotation matrix:

$$\begin{bmatrix} M_{00} & M_{01} & M_{02} & 0 \\ M_{10} & M_{11} & M_{12} & 0 \\ M_{20} & M_{21} & M_{22} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

into the matrix format directly above. The resulting quaternion is trivially $(W, (X, Y, Z))$. Given a general rotation matrix the first thing to do is to examine the sum of its diagonal components M_{ii} where $0 \leq i \leq 3$. This is called the trace of the matrix. From the above format we know:

$$\text{trace} = 1 - 2Y^2 - 2Z^2 + 1 - 2X^2 - 2Z^2 + 1 - 2X^2 - 2Y^2 + 1 = 4 - 4(X^2 + Y^2 + Z^2)$$

Since the matrix represents a rotation we know that the corresponding quaternion must be of unit magnitude, that is:

$$X^2 + Y^2 + Z^2 + W^2 = 1$$

and so the trace reduces to $4W^2$. Thus for a 4×4 homogeneous matrix we have:

$$W = (\text{trace})^{1/2}$$

The remaining components of the quaternion (X, Y, Z) which, as you will recall, is the axis of rotation scaled by half the sine of the angle of rotation, are obtained by combining diagonally opposite elements of the matrix M_{ij} and M_{ji} where $0 \leq i, j \leq 2$. We have:

$$X = \frac{M_{21} - M_{12}}{4W} \quad Y = \frac{M_{02} - M_{20}}{4W} \quad Z = \frac{M_{10} - M_{01}}{4W}$$

For zero W these equations are undefined and so other combinations of the matrix components, along with the fact that the quaternion is of unit magnitude, are used to determine the axis of rotation. Listing 15.3 gives the code in full for moving from rotation matrices to quaternions in the routine `mattoquat(mat, q)`.

Having outlined our scheme we now discuss how to interpolate in quaternion space. Since a rotation maps onto a quaternion of unit magnitude, the entire group of rotations maps onto the surface of the four-dimensional unit hypersphere in quaternion space. Curves interpolating through key orientations should therefore lie on the surface of this sphere. Consider the simplest case of interpolating between just two key quaternions. A naive, straightforward, linear interpolation between the two keys results in a motion that speeds up in the middle. This is because we are not moving along the surface of the hypersphere but cutting across it. In order to ensure a steady rotation we must employ spherical linear interpolation, where we move along an arc of the geodesic that passes through the two keys. (Figure 15.1 showing the differences between interpolating position and interpolating rotation angle is entirely analogous to this situation.) Technically, the metric of the hypersphere's surface is said to be the same as the angular metric of the rotation group.

The formula for spherical linear interpolation is easy to derive geometrically. Consider the two-dimensional case of two vectors A and B separated by angle Ω and vector P which makes an angle θ with A as shown in Figure 15.21. P is derived from spherical interpolation between A and B and we write:

$$P = \alpha A + \beta B$$

Trivially, we can solve for α and β given:

$$\begin{aligned} |P| &= 1 \\ A \cdot B &= \cos \Omega \\ A \cdot P &= \cos \theta \end{aligned}$$

to give:

$$P = A \frac{\sin(\Omega - \theta)}{\sin \Omega} + B \frac{\sin \theta}{\sin \Omega}$$

Spherical linear interpolation between two unit quaternions q_1 and q_2 , where:

$$q_1 \cdot q_2 = \cos \Omega$$

is obtained by generalizing the above to four dimensions and replacing θ by Ωu where $u \in [0,1]$. We write:

$$\text{slerp}(q_1, q_2, u) = q_1 \frac{\sin(1-u)\Omega}{\sin \Omega} + q_2 \frac{\sin \Omega u}{\sin \Omega}$$

Listing 15.3 gives a code fragment for this. `slerp(p, q, t, qt)` returns the interpolated quaternion qt , for t between p and q . The routine caters for the special cases where the keys are very close together, in which case we approximate using the more economical linear interpolation and avoid divisions by very small numbers since

$$\sin \Omega \rightarrow 0 \quad \text{as } \Omega \rightarrow 0$$

The case where p and q are diametrically opposite, or nearly so, also requires special attention.

Now, given any two key quaternions, p and q , there exist two possible arcs along which one can move, corresponding to alternative starting directions on the geodesic that connects them. One of them goes around the long way and this is the one that we wish to avoid. Naively, one might assume that this reduces to either spherically interpolating between p and q by the angle Ω , where:

$$p \cdot q = \cos \Omega$$

or interpolating in the opposite direction by the angle $2\pi - \Omega$. This, however, will not produce the desired effect. The reason is that the topology of the hypersphere of orientation is not just a straightforward extension of the three-dimensional Euclidean sphere. To appreciate this, it is sufficient to consider the fact that every rotation has two representations in quaternion space, namely q and $-q$, that is, the effect of q and $-q$ is the same. That this is so is because algebraically the operator $q(\)q^{-1}$ has exactly the same effect as $(-q)(\)(-q)^{-1}$. Thus, points diametrically opposed represent the same rotation. Because of this topological oddity care must be taken when determining the shorter arc. A strategy that works is to choose interpolating between either the quaternion pairs p and q or p and $-q$. Given two key orientations p and q find the magnitude of their difference, that is $(p - q) \cdot (p - q)$, and compare this to the magnitude of the difference when the second key is negated, that is $(p + q) \cdot (p + q)$. If the former is smaller then we are already moving along the smaller arc and

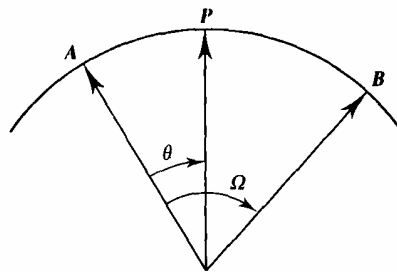


Figure 15.21 Spherical linear interpolation.

nothing needs to be done. If, however, the second is smaller, then we replace q by $-q$ and proceed. These considerations are shown schematically in Figure 15.22.

So far we have described the spherical equivalent of linear interpolation between two key orientations, and, just as was the case for linear interpolation, spherical linear interpolation between more than two key orientations will produce jerky, sharply changing motion across the keys. What is required for higher order continuity is the spherical equivalent of the cubic spline. Unfortunately, because we are now working on the surface of a four-dimensional hypersphere, the problem is far more complex than constructing splines in three-dimensional Euclidean space. [DUFF86] and [SHOE87] have tackled this problem. We shall describe the approach made in [SHOE87] since it pays greatest lip service to implementation points.

The following construction enables us to think of a cubic spline as a series of three linear interpolations. By extension [SHOE87] takes three spherical linear interpolations and defines a cubic spline on the surface of a sphere. Consider four points (S_0, S_1, S_2, S_3) at the corners of the rectangle shown in Figure 15.23. We linearly interpolate by an amount $u \in [0,1]$, along the horizontal edges to get the intermediate points S_α, S_β , where:

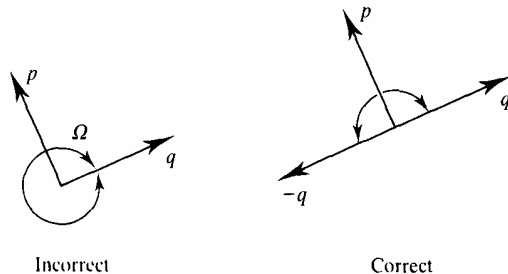


Figure 15.22 Shortest arc determination on quaternion hypersphere.

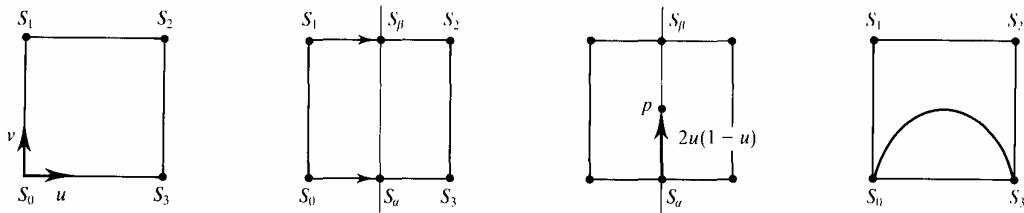


Figure 15.23 The quadrangle construction for a parabola.

$$S_\alpha = S_0(1 - u) + S_3 u$$

$$S_\beta = S_1(1 - u) + S_2 u$$

Now we perform a vertical linear interpolation by an amount

$$v = 2u(1 - u)$$

to get the point

$$p = S_\alpha(1 - v) + S_\beta v$$

As u varies from 0 to 1, the locus of p will trace out a parabola. This process of bilinear interpolation, where the second interpolation is thus restricted, is called 'parabolic blending'. Böhm [BÖHM82] shows how, given a Bézier curve segment (b_0, b_1, b_2, b_3) one can derive the quadrangle points (b_0, S_1, S_2, b_3) of the above construction. This has the geometric significance of enabling us to visualize the cubic as a parabola whose quadrangle points are not necessarily parallel or coplanar. The cubic can be thought of as a warped parabola as shown in Figure 15.24.

The mathematical significance of this construction is that it shows how to construct a cubic as a series of three linear interpolations of the quadrangle points. [SHOE87] takes this construction onto the surface of the four-dimensional hypersphere by constructing a spherical curve, using three spherical linear interpola-

tions of a quadrangle of unit quaternions. This he defines as $\text{squad}()$, where:

$$\text{squad}(b_0, S_1, S_2, b_3, ut) = \text{slerp}(\text{slerp}(b_0, b_3, ut), \text{slerp}(S_1, S_2, ut), 2u(1 - u))$$

Given a series of quaternion keys one can construct a cubic segment across keys q_i and q_{i+1} by constructing a quadrangle of quaternions $(q_i, a_i, b_{i+1}, q_{i+1})$ where a_i, b_{i+1} have to be determined. These inner quadrangle points are chosen in such a way to ensure that continuity of tangents across adjacent cubic segments is guaranteed. The derivation for the inner quadrangle points is difficult, involving as it does the calculus and exponentiation of quaternions and we will just quote the results, referring the interested reader to [SHOE87]:

$$a_i = b_i = q_i \exp \left(- \frac{\ln(q_i^{-1} q_{i+1}) + \ln(q_i^{-1} q_{i-1})}{4} \right)$$

where, for the unit quaternion:

$$q = (\cos \theta, \sin \theta v)$$

$$|v| = 1$$

$$\ln(q) = (0, \theta v)$$

and, inversely for the pure quaternion (zero scalar part):

$$q = (0, \theta v)$$

$$\exp(q) = (\cos \theta, \sin \theta v)$$

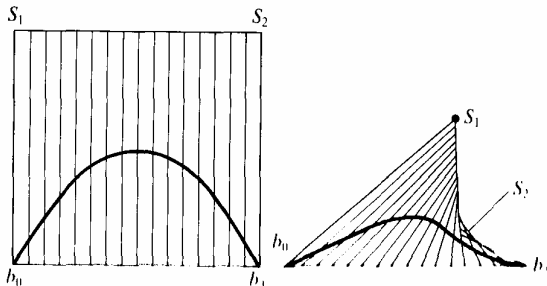


Figure 15.24 A warped parabola is a cubic.