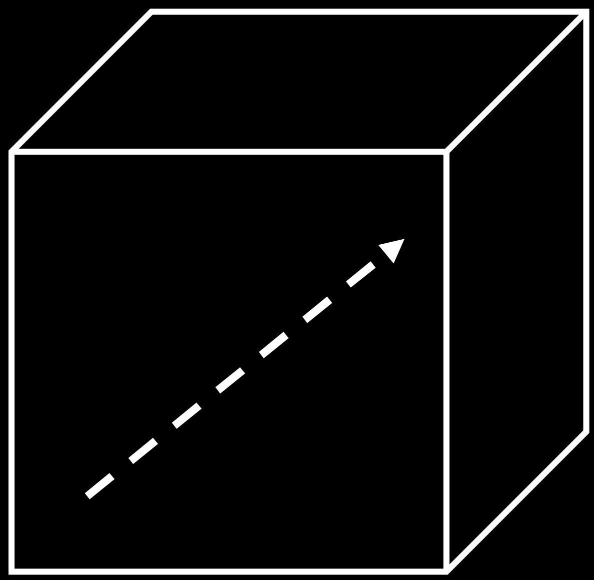
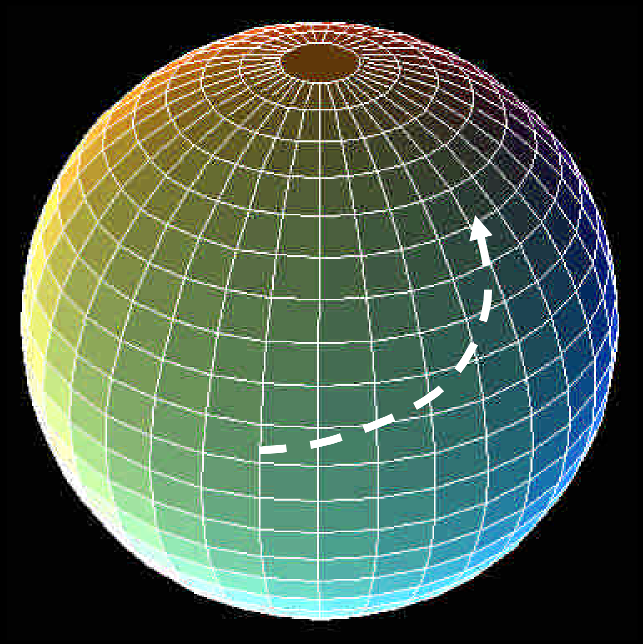
Quaternion Interpolation





3D Rotation Representations (review)

- Rotation Matrix
  - orthornormal columns/rows
  - bad for interpolation
- Fixed Angle
  - rotate about global axes
  - bad for interpolation, gimbal lock
- Euler Angle
  - rotate about local axes
  - same problem as fixed angle

3D Rotation Representations (review)

- Axis angle
  - rotate about A by θ , (A_x,A_y,A_z,θ)
  - good interpolation, no gimbal lock
  - bad for compounding rotations
- Quaternion
  - similar to axis angle but in different form
  - $q=[s,v]$
  - good for compounding rotations

Quaternion Math (review)

- Addition

$$[s_1,\ v_1]+[s_2,\ v_2]=[s_1+s_2,\ v_1+v_2]$$

- Multiplication

$$[s_1,\ v_1]\cdot[s_2,\ v_2]=[s_1 s_2 - v_1\cdot v_2,\quad s_1 v_2 + s_2 v_1 + v_1\times v_2]$$

- Multiplication is associative but not commutative

$$q_1(q_2 q_3)=(q_1 q_2)q_3 \qquad q_1 q_2 \neq q_2 q_1$$

- q and $-q$ represent the same orientation

Quaternion Rotation (review)

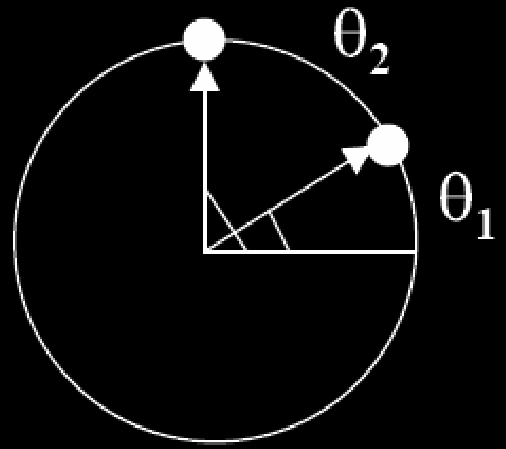
- To rotate a vector *v* using quaternion
  - Represent the vector as [0, *v*]
  - Represent the rotation as a quaternion *q*

$$v' = Rot_q(v) = q \cdot v \cdot q^{-1}$$

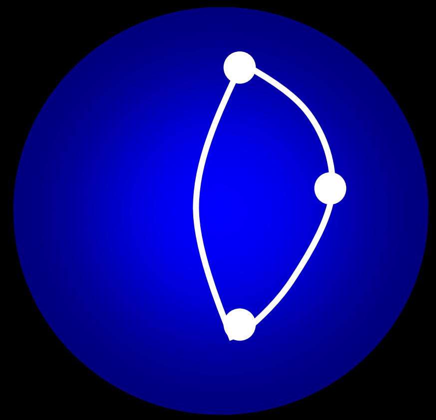
- *q* and *cq* has the same rotation effect to *v*
  - *c* is a scalar

Visualizing Rotations

- View rotations as points lying on an n-D sphere



1-angle rotation
unit circle in 2D space

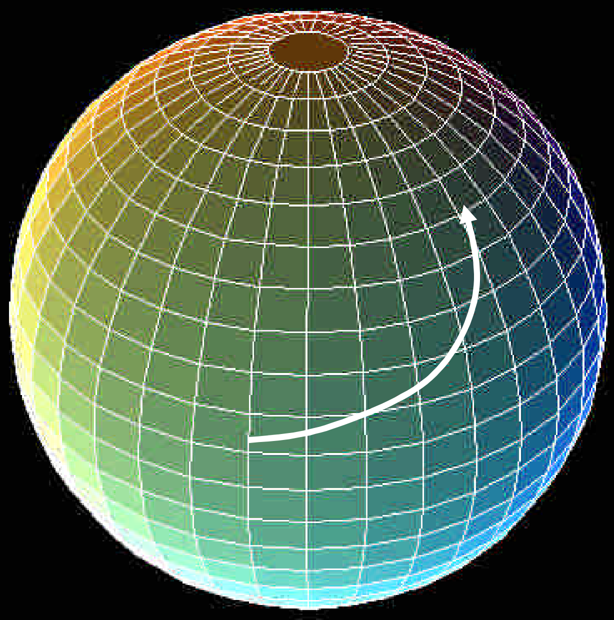


2-angle rotation
unit sphere in 3D space

- Interpolating rotation means moving on n-D sphere
- How about 3-angle rotation (quaternion)?

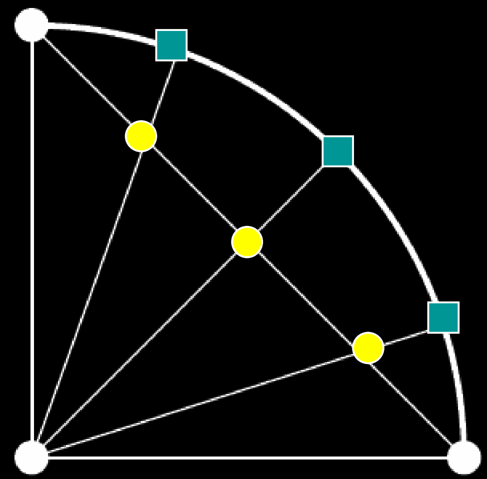
Quaternion Interpolation

- A quaternion is a point on a 4D unit sphere
- Unit quaternion: $q=(s,x,y,z)$, $||q|| = 1$
- Interpolating rotations means moving on 4D sphere



Linear Interpolation

- Linear interpolation generates unequal spacing of points after projecting to circle

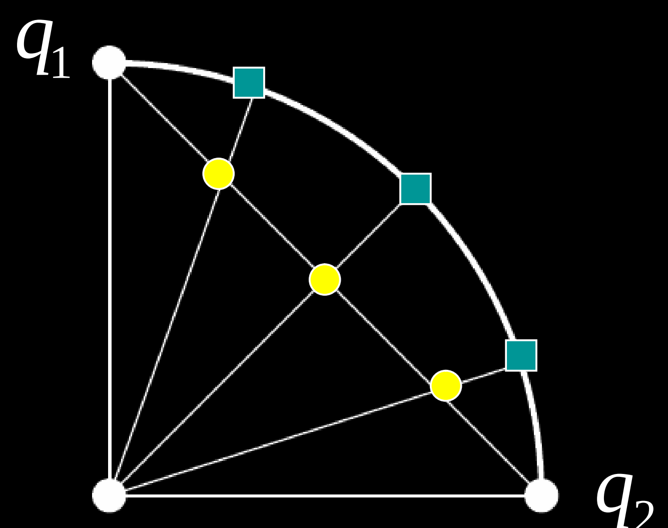


Spherical Linear Interpolation (slerp)

- Want equal increment along arc connecting two quaternions on the spherical surface

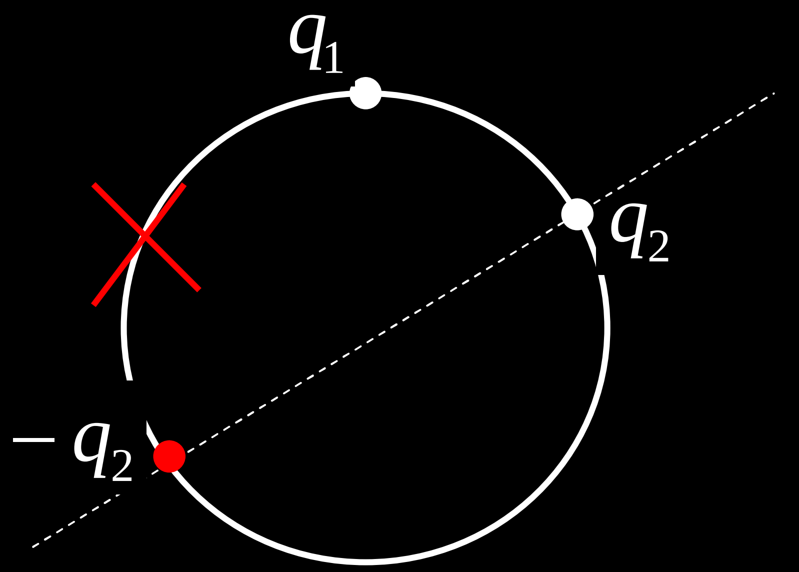
$$slerp(q_1, q_2, u) = \frac{\sin(1-u)\theta}{\sin\theta} q_1 + \frac{\sin u\theta}{\sin\theta} q_2$$

- Normalize to regain unit quaternion



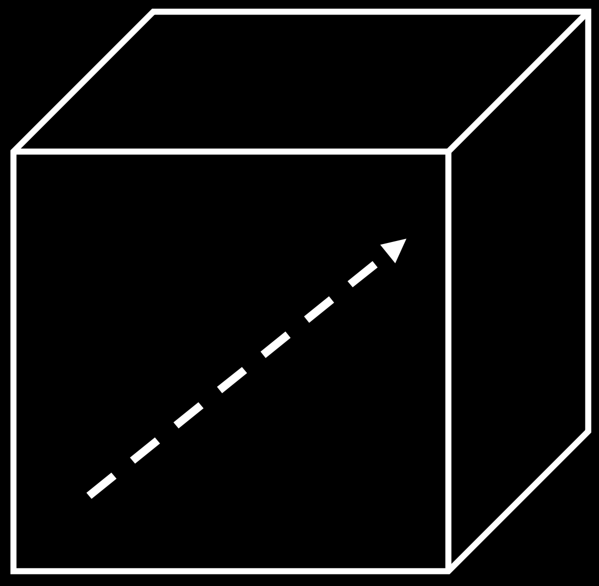
Slerp

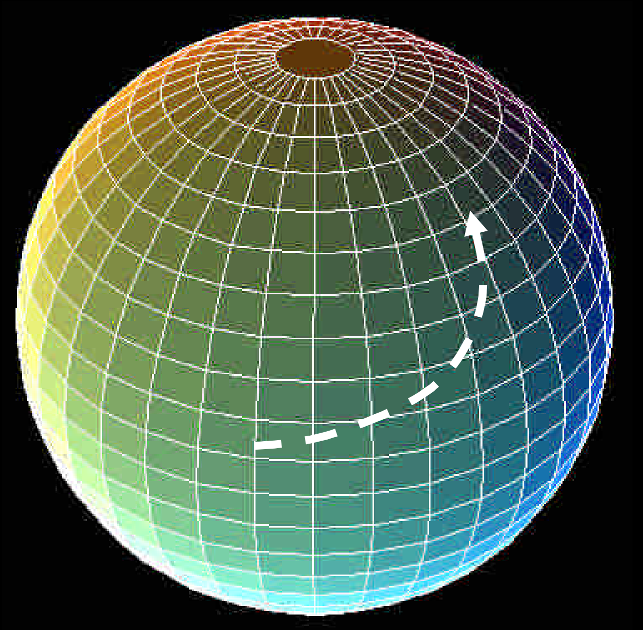
- Recall that q and $-q$ represent same rotation
- Slerp can go the LONG way!
- Have to go the short way $q_1 \cdot q_2 > 0$



Useful Analogies

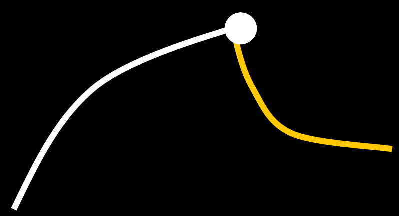
Euclidean Space	⟺	4D Spherical Space
Position		Orientation
Linear interpolation		Spherical linear interpolation





What if there are multiple segments?

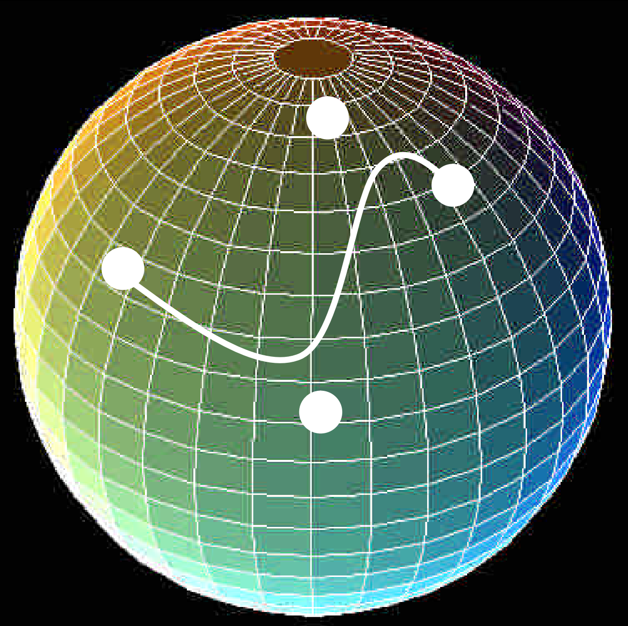
- As linear interpolation in Euclidean space, we can have first order discontinuity



- Need a cubic curve interpolation to maintain first order continuity

Bezier Interpolation on 4D Sphere

- Have to perform interpolation on 4D sphere
- Construct Bezier curve by iteratively applying slerp

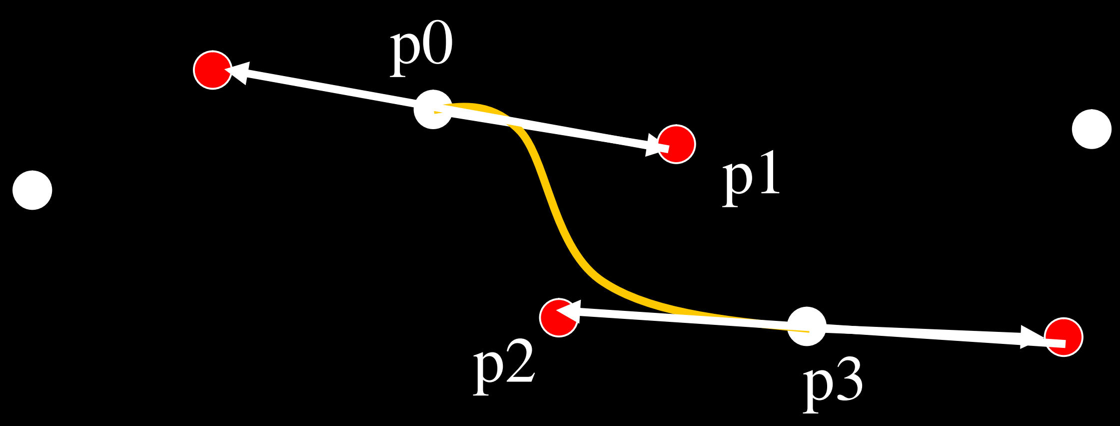


Bezier Interpolation in Euclidean Space

$$P(u) = [u3\ u2\ u\ 1] \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 3 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p0 \\ p1 \\ p2 \\ p3 \end{pmatrix}$$

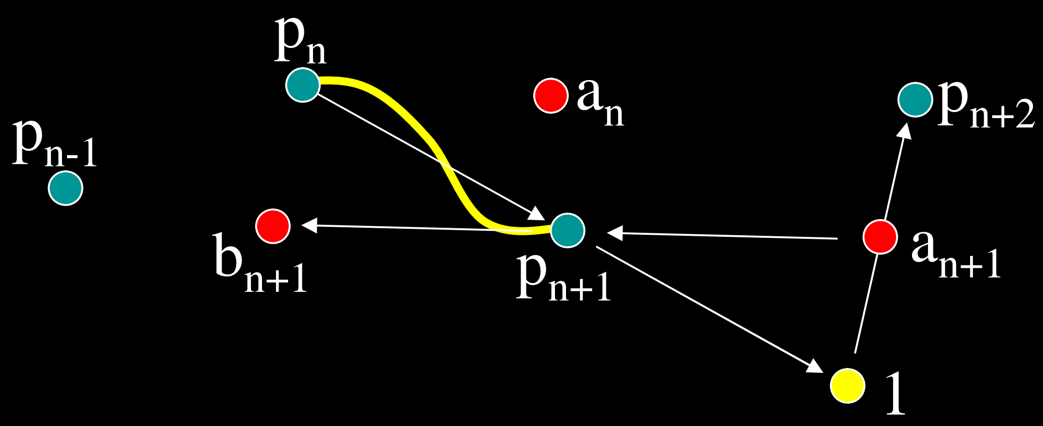
Colinearity of the control points at either side of an endpoint guarantees the 1st order continuity

$$P'(0) = 3(p1-p0),\ P'(1)=3(p3-p2)$$



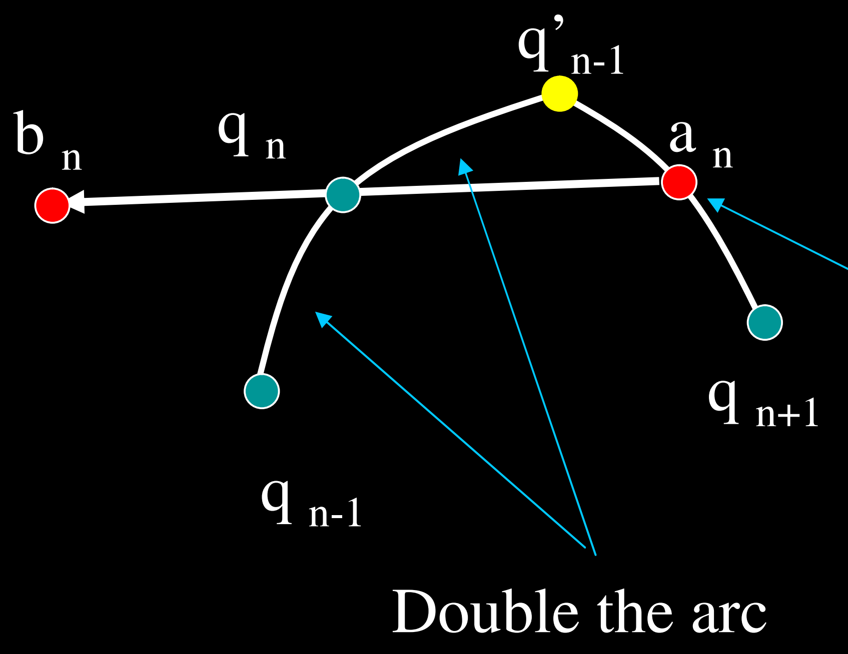
Bezier Interpolation in Euclidean Space

- Automatically generate control points



Bezier Interpolation on 4D sphere

- Automatically generating interior (spherical) control point

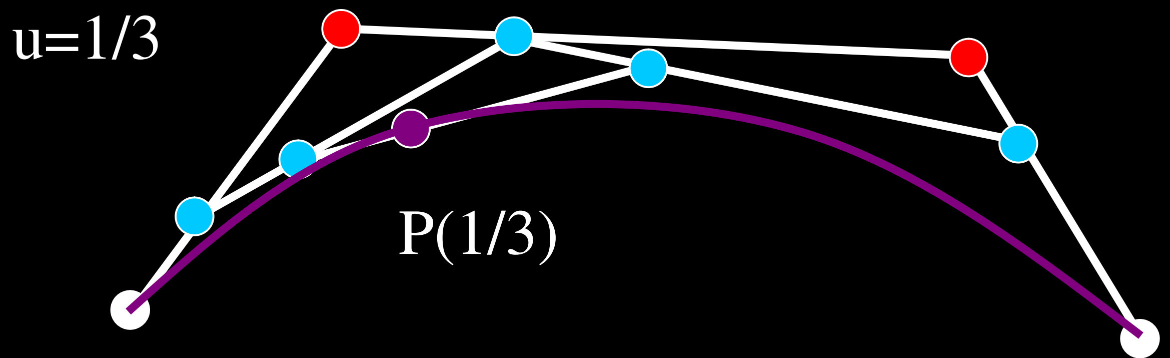


$$Bisect(q'_{n-1}, q_{n+1}) = \frac{q'_{n-1} + q_{n+1}}{\left\| q'_{n-1} + q_{n+1} \right\|}$$

$$double(q_{n-1}, q_n) = 2(q_{n-1} \cdot q_n)q_n - q_{n-1}$$

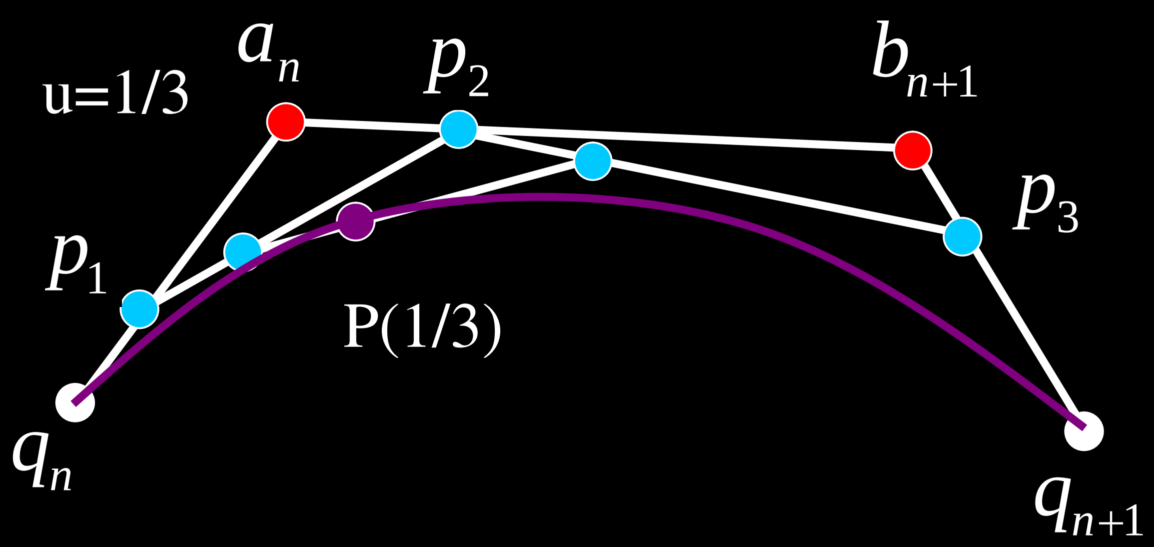
De Casteljau Construction of Bezier Curve

- Constructing Bezier curve by multiple linear interpolation



De Casteljau Construction on 4D Sphere

$$p_1 = slerp(q_n, a_n, \tfrac{1}{3})$$
$$p_2 = slerp(a_n, b_{n+1}, \tfrac{1}{3})$$
$$p_3 = slerp(b_{n+1}, q_{n+1}, \tfrac{1}{3})$$
$$p_{12} = slerp(p_1, p_2, \tfrac{1}{3})$$
$$p_{23} = slerp(p_2, p_3, \tfrac{1}{3})$$
$$p = slerp(p_{12}, p_{23}, \tfrac{1}{3})$$



Bezier Interpolation in Euclidean Space

- Automatically generate control points