Quaternions and Exponentials

Michael Kazhdan

(601.457/657)

Recall

We saw two different methods for interpolating/approximating between rotations:

- Normalization: (SVD)
  Blend as 3×3 matrices and then map to the closest rotation.
  - ✗ Requires SVD
  - ✗ Works in a 9-dimensional space
- Parameterization: (Euler angles)
  Compute the parameter values, blend those, and then evaluate at the blended values.
  - ✗ Parameterization is not uniform
    (e.g. dense sampling near poles)

Overview

Math review

- Cross products
- Symmetric matrices
- Complex numbers
- The exponential map

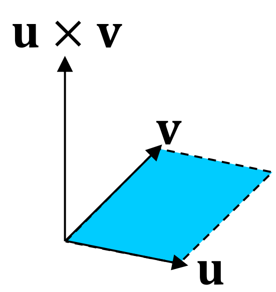
Quaternions

The exponential map

Cross Product

Given vectors $\mathbf{u} = (u_1, u_2, u_3)^\mathsf{T}$ and $\mathbf{v} = (v_1, v_2, v_3)^\mathsf{T}$ in 3D, the cross product of $\mathbf{u}$ and $\mathbf{v}$ is:

$$\mathbf{u} \times \mathbf{v} = \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix}$$



Properties:

- The cross product is orthogonal to both $\mathbf{u}$ and $\mathbf{v}$.
- The vectors $\mathbf{u}$, $\mathbf{v}$, $\mathbf{u} \times \mathbf{v}$ align with the right-hand rule.
- The length of the cross product is equal to the area of the parallelogram defined by $\mathbf{u}$ and $\mathbf{v}$.
- $\mathbf{u} \times \mathbf{v} = -\mathbf{v} \times \mathbf{u}$
- $\mathbf{u} \times (\mathbf{v} + \mathbf{w}) = \mathbf{u} \times \mathbf{v} + \mathbf{u} \times \mathbf{w}$
- $(t\mathbf{u}) \times \mathbf{v} = t(\mathbf{u} \times \mathbf{v})$

(Skew) Symmetric Matrices

A matrix $\mathbf{M}$ is <u>symmetric</u> if:

$$\mathbf{M}_{ij} = \mathbf{M}_{ji} \quad \Leftrightarrow \quad \mathbf{M} = \mathbf{M}^\top$$

A matrix $\mathbf{M}$ is <u>skew-symmetric</u> if:

$$\mathbf{M}_{ij} = -\mathbf{M}_{ji} \quad \Leftrightarrow \quad \mathbf{M} = -\mathbf{M}^\top$$

The space of (skew) symmetric matrices is closed under addition and scaling:

- If $\mathbf{A} = \mathbf{A}^\top$ and $\mathbf{B} = \mathbf{B}^\top$, then $(\mathbf{A} + \mathbf{B}) = (\mathbf{A} + \mathbf{B})^\top$.
- If $\mathbf{A} = -\mathbf{A}^\top$ and $\mathbf{B} = -\mathbf{B}^\top$, then $(\mathbf{A} + \mathbf{B}) = -(\mathbf{A} + \mathbf{B})^\top$.
- If $\mathbf{A} = \mathbf{A}^\top$ then $(\alpha\mathbf{A}) = (\alpha\mathbf{A})^\top$.
- If $\mathbf{A} = -\mathbf{A}^\top$ then $(\alpha\mathbf{A}) = -(\alpha\mathbf{A})^\top$.

Complex Numbers

Complex numbers are extensions of the real numbers, incorporating an imaginary value:

$$a + ib$$

We add complex numbers together by summing the real and imaginary components:

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

Squaring the imaginary component gives:

$$i^2 = -1$$

The product of two complex numbers is:

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

Complex Numbers

Given a complex number $c = a + ib$

The conjugate of c is:

$$\bar{c} = a - ib$$

The (squared) norm of c is the real value:

$$|c|^2 = a^2 + b^2 = c \cdot \bar{c}$$

The norm of the product is the product of the norms:

$$|c_1 \cdot c_2| = |c_1| \cdot |c_2|$$

The reciprocal of c (assuming $c \neq 0$) is defined by dividing the conjugate of c by the square norm:

$$\frac{1}{c} = \frac{1}{c} \cdot \frac{\bar{c}}{\bar{c}} = \frac{\bar{c}}{|c|^2}$$

The Exponential Map

The *exponential* is a map from real values to positive real values:

$$\exp\colon \mathbb{R} \rightarrow \mathbb{R}^{>0}$$

The inverse is the *logarithm*, taking positive real values to real values:

$$\ln\colon \mathbb{R}^{>0} \rightarrow \mathbb{R}$$

The Exponential Map

Properties:

$$\exp(0) = 1$$

$$\left.\frac{\partial \exp(t\alpha)}{\partial t}\right|_{t=0} = \alpha \exp(t\alpha)\,|_{t=0} = \alpha$$

$$\ln\big(\exp(t)\big) = t$$

The Exponential Map

<u>Taylor Expansion</u>:

Can approximate the exponential map by its Taylor Expansion around $s = 0$:

$$\exp(s) = 1 + s + \frac{1}{2!}s^2 + \cdots + \frac{1}{n!}s^n + \cdots$$

Can approximate the logarithm map by its Taylor Expansion around $s = 1$:

$$\ln(s) = (s-1) - \frac{(s-1)^2}{2} + \cdots + (-1)^{n+1}\frac{(s-1)^n}{n} + \cdots$$

Overview

Math review

Quaternions

The exponential map

Quaternions

Normalization:

- Treat rotations as living in a linear space (3x3 matrices)
- Blend rotations
- Map the blend to the closest rotation

Goal:

Find a linear space making it easy to map the blend to the closest rotation

Quaternions

Quaternions are extensions of complex numbers, with three imaginary values instead of one:

$$a + ib + jc + kd$$

Like the complex numbers, add quaternions by summing individual components:

$$\begin{array}{r} (a_1 + ib_1 + jc_1 + kd_1) \\ +(a_2 + ib_2 + jc_2 + kd_2) \\ \hline = (a_1 + a_2) + i(b_1 + b_2) + j(c_1 + c_2) + k(d_1 + d_2) \end{array}$$

Quaternions

Quaternions are extensions of complex numbers, with three imaginary values instead of one:

$$a + ib + jc + kd$$

Like the imaginary component of complex numbers, squaring the components gives:

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

The multiplication rules are:

$$\begin{array}{lll} ij = k & ik = -j & jk = i \\ ji = -k & ki = j & kj = -i \end{array}$$

<u>Note</u>:
Multiplication of quaternions is not commutative – the result is order-dependent.

Quaternions

More generally, the product of two quaternions is:

$$
\begin{array}{r}
(a_1 + ib_1 + jc_1 + kd_1) \\
\times\,(a_2 + ib_2 + jc_2 + kd_2) \\
\hline
= \;(a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) \\
+i\,(a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2) \\
+j\,(a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2) \\
+k(a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)
\end{array}
$$

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

$$
\begin{array}{lll}
ij = \;\; k & ik = -j & jk = \;\; i \\
ji = -k & ki = \;\; j & kj = -i
\end{array}
$$

Quaternions

As with complex numbers, for $q = a + ib + jc + kd$:

The conjugate is:

$$\bar{q} = a - ib - jc - kd$$

The (squared) norm is:

$$|q|^2 = a^2 + b^2 + c^2 + d^2 = q \cdot \bar{q}$$

The norm of the products is the product of the norms:

$$|q_1 \cdot q_2| = |q_1| \cdot |q_2|$$

The reciprocal is defined by dividing the conjugate by the square norm:

$$\frac{1}{q} = \frac{1}{q} \cdot \frac{\bar{q}}{\bar{q}} = \frac{\bar{q}}{|q|^2}$$

Quaternions

Can express a quaternion is as a pair consisting of a scalar (real) and a 3D vector (imaginary) :

$$q = (\alpha, \mathbf{w}) \quad \text{with } \alpha = a \text{ and } \mathbf{w} = (b, c, d)^\top$$

In this representation, multiplication becomes:

$$\begin{aligned} q_1 \cdot q_2 &\equiv (\alpha_1, \mathbf{w}_1) \cdot (\alpha_2, \mathbf{w}_2) \\ &= (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2) \end{aligned}$$

$$\boxed{\begin{aligned} q_1 \cdot q_2 = &(a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &+ i\,(a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1) \\ &+ j\,(a_1 c_2 + a_2 c_1 - b_1 d_2 + b_2 d_1) \\ &+ k(a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1) \end{aligned}}$$

Quaternions

Can express a quaternion is as a pair consisting of a scalar (real) and a 3D vector (imaginary) :

$$q = (\alpha, \mathbf{w}) \quad \text{with } \alpha = a \text{ and } \mathbf{w} = (b, c, d)^{\mathsf{T}}$$

In this representation, multiplication becomes:

$$\begin{aligned} q_1 \cdot q_2 &\equiv (\alpha_1, \mathbf{w}_1) \cdot (\alpha_2, \mathbf{w}_2) \\ &= (\boxed{\alpha_1 \cdot \alpha_2} - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2) \end{aligned}$$

$$\boxed{\begin{aligned} q_1 \cdot q_2 = (&\boxed{a_1 a_2} - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &+ i\,(a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1) \\ &+ j\,(a_1 c_2 + a_2 c_1 - b_1 d_2 + b_2 d_1) \\ &+ k(a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1) \end{aligned}}$$

Quaternions

Can express a quaternion is as a pair consisting of a scalar (real) and a 3D vector (imaginary) :

$$q = (\alpha, \mathbf{w}) \quad \text{with } \alpha = a \text{ and } \mathbf{w} = (b, c, d)^\top$$

In this representation, multiplication becomes:

$$\begin{aligned} q_1 \cdot q_2 &\equiv (\alpha_1, \mathbf{w}_1) \cdot (\alpha_2, \mathbf{w}_2) \\ &= (\alpha_1 \cdot \alpha_2 \boxed{- \langle \mathbf{w}_1, \mathbf{w}_2 \rangle}, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2) \end{aligned}$$

$$\boxed{\begin{aligned} q_1 \cdot q_2 = (a_1 a_2 &\boxed{- b_1 b_2 - c_1 c_2 - d_1 d_2)} \\ &+ i\,(a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1) \\ &+ j\,(a_1 c_2 + a_2 c_1 - b_1 d_2 + b_2 d_1) \\ &+ k(a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1) \end{aligned}}$$

Quaternions

Can express a quaternion is as a pair consisting of a scalar (real) and a 3D vector (imaginary) :

$$q = (\alpha, \mathbf{w}) \quad \text{with } \alpha = a \text{ and } \mathbf{w} = (b, c, d)^\top$$

In this representation, multiplication becomes:

$$\begin{aligned} q_1 \cdot q_2 &\equiv (\alpha_1, \mathbf{w}_1) \cdot (\alpha_2, \mathbf{w}_2) \\ &= (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \boxed{\alpha_1 \cdot \mathbf{w}_2} + \alpha_2 \cdot \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2) \end{aligned}$$

$$\begin{aligned} q_1 \cdot q_2 = &(a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &+ i\,(a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1) \\ &+ j\,(a_1 c_2 + a_2 c_1 - b_1 d_2 + b_2 d_1) \\ &+ k\,(a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1) \end{aligned}$$

Quaternions

Can express a quaternion is as a pair consisting of a scalar (real) and a 3D vector (imaginary) :

$$q = (\alpha, \mathbf{w}) \quad \text{with } \alpha = a \text{ and } \mathbf{w} = (b, c, d)^\top$$

In this representation, multiplication becomes:

$$\begin{aligned} q_1 \cdot q_2 &\equiv (\alpha_1, \mathbf{w}_1) \cdot (\alpha_2, \mathbf{w}_2) \\ &= (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \boxed{\alpha_2 \cdot \mathbf{w}_1} + \mathbf{w}_1 \times \mathbf{w}_2) \end{aligned}$$

$$\begin{aligned} q_1 \cdot q_2 = &(a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &+ i\,(a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1) \\ &+ j\,(a_1 c_2 + a_2 c_1 - b_1 d_2 + b_2 d_1) \\ &+ k(a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1) \end{aligned}$$

Quaternions

Can express a quaternion is as a pair consisting of a scalar (real) and a 3D vector (imaginary) :

$$q = (\alpha, \mathbf{w}) \quad \text{with } \alpha = a \text{ and } \mathbf{w} = (b, c, d)^{\mathsf{T}}$$

In this representation, multiplication becomes:

$$\begin{aligned} q_1 \cdot q_2 &\equiv (\alpha_1, \mathbf{w}_1) \cdot (\alpha_2, \mathbf{w}_2) \\ &= (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \boxed{\mathbf{w}_1 \times \mathbf{w}_2}) \end{aligned}$$

$$\boxed{\begin{aligned} q_1 \cdot q_2 = &(a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &+ i\,(a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1) \\ &+ j\,(a_1 c_2 + a_2 c_1 - b_1 d_2 + b_2 d_1) \\ &+ k\,(a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1) \end{aligned}}$$

Quaternions

Can express a quaternion is as a pair consisting of a scalar (real) and a 3D vector (imaginary) :

$$q = (\alpha, \mathbf{w}) \quad \text{with } \alpha = a \text{ and } \mathbf{w} = (b, c, d)^{\top}$$

In this representation, multiplication becomes:

$$\begin{aligned} q_1 \cdot q_2 &\equiv (\alpha_1, \mathbf{w}_1) \cdot (\alpha_2, \mathbf{w}_2) \\ &= (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \boxed{\mathbf{w}_1 \times \mathbf{w}_2}) \end{aligned}$$

This is the (only) part that is order-dependent.

$$\begin{aligned} q_1 \cdot q_2 = &(a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &+i\,(a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1) \\ &+j\,(a_1 c_2 + a_2 c_1 - b_1 d_2 + b_2 d_1) \\ &+k(a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1) \end{aligned}$$

Quaternions as Transformations

Can also associate points in 3D with imaginary quaternions:

$$(x, y, z) \to ix + jy + kz = (0, \mathbf{w})$$

Given a **quaternion** q and an imaginary quaternion (3D point) p, consider the map:

$$q(p) = qp\bar{q}$$

Quaternions as Transformations

$$q(p) = qp\bar{q}$$

$$q_1 \cdot q_2 = (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2)$$

Claim:

1. This takes imaginary quaternions (points in 3D) to imaginary quaternions (points in 3D).

$$\begin{aligned} q(p) &= \left(\alpha_q, \mathbf{w}_q\right) \cdot \left(0, \mathbf{w}_p\right) \cdot \left(\alpha_q, -\mathbf{w}_q\right) \\ &= \left(-\langle \mathbf{w}_q, \mathbf{w}_p \rangle, \alpha_q \mathbf{w}_p + \mathbf{w}_q \times \mathbf{w}_p\right) \cdot \left(\alpha_q, -\mathbf{w}_q\right) \\ &= \left(-\alpha_q \langle \mathbf{w}_q, \mathbf{w}_p \rangle + \alpha_q \langle \mathbf{w}_p, \mathbf{w}_q \rangle + \langle \mathbf{w}_q \times \mathbf{w}_p, \mathbf{w}_q \rangle, \dots\right) \\ &= (0, \dots) \end{aligned}$$

Quaternions as Transformations

$$q(p) = qp\bar{q}$$

$$q_1 \cdot q_2 = (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2)$$

<u>Claim</u>:

1. This takes imaginary quaternions (points in 3D) to imaginary quaternions (points in 3D).

2. The map is linear

$$\begin{aligned} q(a \cdot p_1 + b \cdot p_2) &= q(a \cdot p_1 + b \cdot p_2)\bar{q} \\ &= a \cdot qp_1\bar{q} + b \cdot qp_2\bar{q} \\ &= a \cdot q(p_1) + b \cdot q(p_2) \end{aligned}$$

Quaternions as Transformations

$$q(p) = qp\bar{q}$$

$$q_1 \cdot q_2 = (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2)$$

Claim:

1. This takes imaginary quaternions (points in 3D) to imaginary quaternions (points in 3D).
2. The map is linear
3. If $|q| = 1$, the map is norm-preserving

$$\begin{aligned} |q(p)| &= |qp\bar{q}| \\ &= |q||p||\bar{q}| \\ &= |p| \end{aligned}$$

Quaternions as Transformations

$$q(p) = qp\bar{q}$$

$$q_1 \cdot q_2 = (\alpha_1 \cdot \alpha_2 - \langle \mathbf{w}_1, \mathbf{w}_2 \rangle, \alpha_1 \cdot \mathbf{w}_2 + \alpha_2 \cdot \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2)$$

<u>Claim</u>:

1. This takes imaginary quaternions (points in 3D) to imaginary quaternions (points in 3D).

2. The map is linear

3. If $|q| = 1$, the map is norm-preserving

$$\Downarrow$$

When q is a unit quaternion, the map $p \to qp\bar{q}$ is an orthogonal transformation (specifically, a rotation).

Unit Quaternions and Rotations

If $q = a + ib + jc + kd$ is a unit quaternion ($|q| = 1$), we can associate q with the rotation:

$$\mathbf{R}(q) = \begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & a^2 - b^2 - c^2 + d^2 \end{pmatrix}$$

All of the terms are quadratic.

$\Downarrow$

The rotation associated with q is the same as the rotation associated with $-q$.

Unit Quaternions and Rotations

If $q = a + ib + jc + kd$ is a unit quaternion ($|q| = 1$), we can associate q with the rotation:

$$\mathbf{R}(q) = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{pmatrix}$$

Because q is a unit quaternion, we have:

$$|q|^2 = \|(\alpha, \mathbf{w})\|^2 = \alpha^2 + \|\mathbf{w}\|^2 = 1$$

$\Rightarrow$ Setting $\mathbf{v} = \mathbf{w}/\|\mathbf{w}\|$, there exists θ such that:

$$q = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{v}\right)$$

Unit Quaternions and Rotations

If $q = a + ib + jc + kd$ is a unit quaternion ($|q| = 1$), we can associate q with the rotation:

$$\mathbf{R}(q) = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{pmatrix}$$

Because q is a unit quaternion, we have:

$$q = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{v}\right)$$

It turns out that q corresponds to the rotation whose:

axis of rotation is $\mathbf{v}$, and

angle of rotation is θ.

Unit Quaternions and Rotations

If $q = a + ib + jc + kd$ is a unit quaternion ($|q| = 1$), we can associate q with the rotation:

$$\mathbf{R}(q) = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{pmatrix}$$

Because q is a unit quaternion, we have:

If we express rotations in the axis-angle representation, we can compute the composition by multiplying quaternions.

It turns out that q corresponds to the rotation whose:

axis of rotation is $\mathbf{v}$, and

angle of rotation is θ.

Quaternions

Instead of blending rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ and then normalizing using SVD, we can blend the quaternions and then normalize them:

For each $\mathbf{R}_i$, compute the quaternion rep. $(\alpha_i, \mathbf{w}_i)$

Quaternions

Instead of blending rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each $\mathbf{R}_i$, compute the quaternion rep. $(\alpha_i, \mathbf{w}_i)$
- Interpolate/Approximate the quaternions:

Quaternions

Instead of blending rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each $\mathbf{R}_i$, compute the quaternion rep. $(\alpha_i, \mathbf{w}_i)$
- Interpolate/Approximate the quaternions:
  - Linear Interpolation:

$$\alpha_k(t) = (1-t)\alpha_k + t\alpha_{k+1}$$
$$\mathbf{w}_k(t) = (1-t)\mathbf{w}_k + t\mathbf{w}_{k+1}$$

Quaternions

Instead of blending rotations $\{\mathbf{R}_0,\ldots,\mathbf{R}_{n-1}\}$ and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each $\mathbf{R}_i$, compute the quaternion rep. $(\alpha_i,\mathbf{w}_i)$
- Interpolate/Approximate the quaternions:
  - Linear Interpolation
  - Catmull-Rom Interpolation:

$$\alpha_k(t)=CR_0(t)\alpha_{k-1}+CR_1(t)\alpha_k+CR_2(t)\alpha_{k+1}+CR_3(t)\alpha_{k+2}$$
$$\mathbf{w}_k(t)=CR_0(t)\mathbf{w}_{k-1}+CR_1(t)\mathbf{w}_k+CR_2(t)\mathbf{w}_{k+1}+CR_3(t)\mathbf{w}_{k+2}$$

Quaternions

Instead of blending rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each $\mathbf{R}_i$, compute the quaternion rep. $(\alpha_i, \mathbf{w}_i)$
- Interpolate/Approximate the quaternions:
  - Linear Interpolation
  - Catmull-Rom Interpolation
  - Uniform Cubic B-Spline Approximation:

$$\alpha_k(t) = B_{0,3}(t)\alpha_{k-1} + B_{1,3}(t)\alpha_k + B_{2,3}(t)\alpha_{k+1} + B_{3,3}(t)\alpha_{k+2}$$
$$\mathbf{w}_k(t) = B_{0,3}(t)\mathbf{w}_{k-1} + B_{1,3}(t)\mathbf{w}_k + B_{2,3}(t)\mathbf{w}_{k+1} + B_{3,3}(t)\mathbf{w}_{k+2}$$

Quaternions

Instead of blending rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each $\mathbf{R}_i$, compute the quaternion rep. $(\alpha_i, \mathbf{w}_i)$
- Interpolate/Approximate the quaternions:
  - Linear Interpolation
  - Catmull-Rom Interpolation
  - Uniform Cubic B-Spline Approximation
- Set the value of the in-between rotation to be the normalized quaternion:

$$q_k(t) = \frac{\big(\alpha_k(t), \mathbf{w}_k(t)\big)}{\big\|\big(\alpha_k(t), \mathbf{w}_k(t)\big)\big\|}$$

Quaternions

Instead of blending rotations $\{\mathbf{R}_0,\ldots,\mathbf{R}_{n-1}\}$ and then normalizing using SVD, we can blend the quaternions and then normalize them:

For each $\mathbf{R}_i$, compute the quaternion rep. $(\alpha_i,\mathbf{w}_i)$

Interpolate/Approximate the quaternions:

Note:

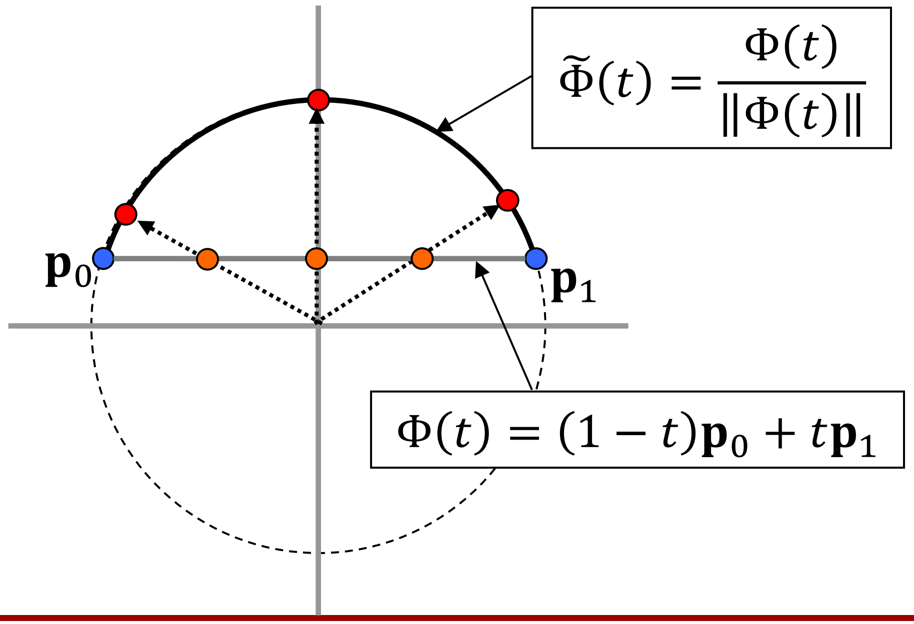
- Using SVD, we interpolated in the $(9=3\times 3)$-dimensional space of matrices and then normalized.
- With quaternions we interpolate in the 4-dimensional space of quaternions and normalize.

$$q_k(t)=\frac{\big(\alpha_k(t),\mathbf{w}_k(t)\big)}{\big\|\big(\alpha_k(t),\mathbf{w}_k(t)\big)\big\|}$$

Quaternions

As with points on the circle/sphere, this type of interpolation/approximation has the limitation:

Uniform sampling in quaternion space does not result in uniform sampling in rotation space.



Quaternions

As with points on the circle/sphere, this type of interpolation/approximation has the limitation:

Uniform sampling in quaternion space does not result in uniform sampling in rotation space.

Additionally, since $\mathbf{R}(-q) = \mathbf{R}(q)$ there are two different quaternions we can associate with a rotation, so the mapping is not unique.

Quaternions

Aside:

In animations/games, the orientation of the camera is the result of the composition of many rotations.

Due to numerical imprecision, the composition of these rotations will not, in practice, be a rotation.

To avoid distortion, need to "snap" the composition to the closest rotation.

This is easily done using quaternions to represent the camera's orientation.

Overview

Math review

Quaternions

The exponential map

The Exponential Map

Parametrization:

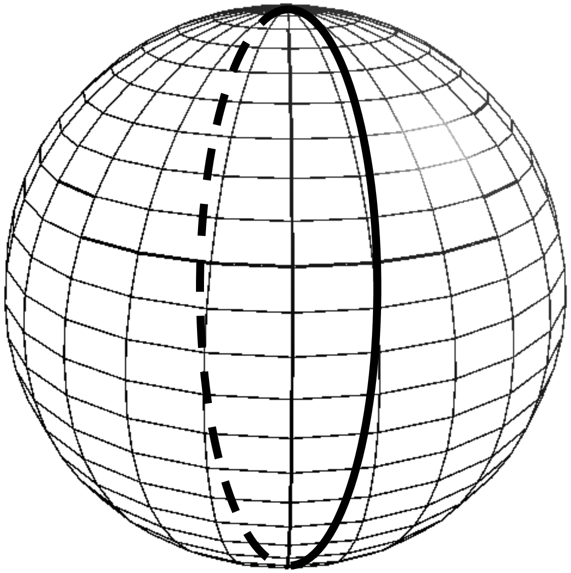
- Parameterize rotations in a linear space
- Blend parameters
- Evaluate the parameterization at the blend

Goal:

Find a canonical way to parametrize rotations so that there is little distortion

Geodesics

Given a surface $\mathcal{S}(u,v)$ a *geodesic* is a curve that is (locally) the shortest path between two points.



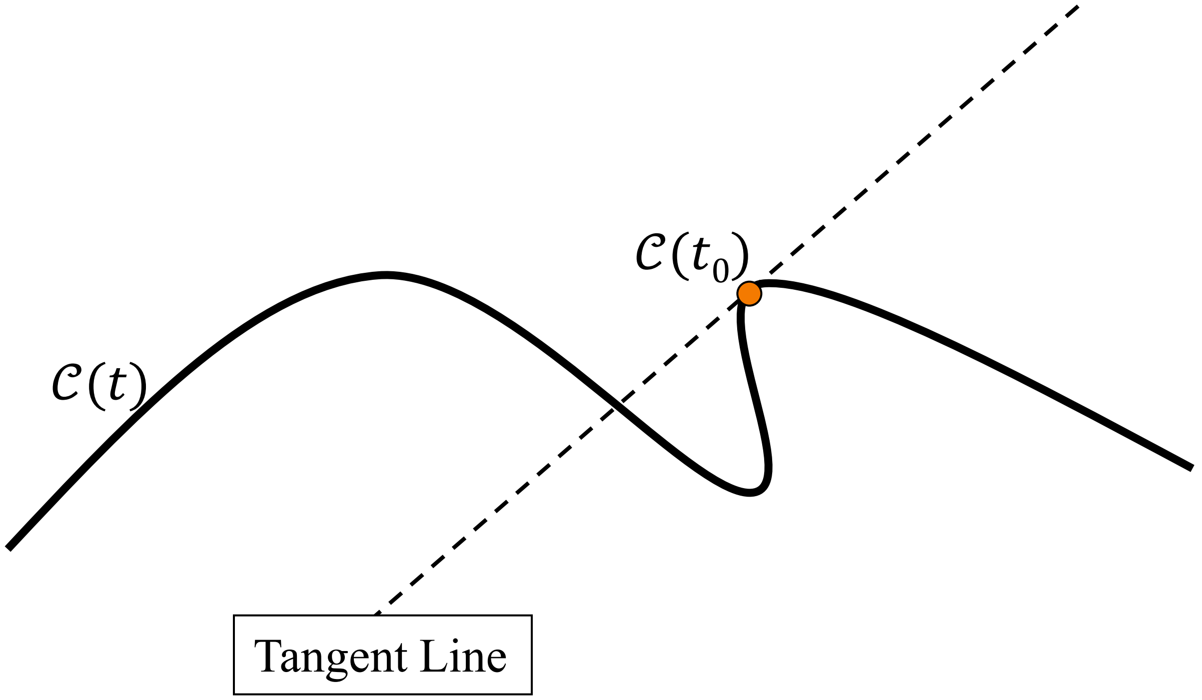
$$\mathcal{S}(u,v) = (\cos u \cos v\,, \sin u\,, \cos u \sin v)$$

Geodesics

Given a manifold (a d-dimensional surface) a *geodesic* is a curve that is (locally) the shortest path between two points.

Tangent Spaces

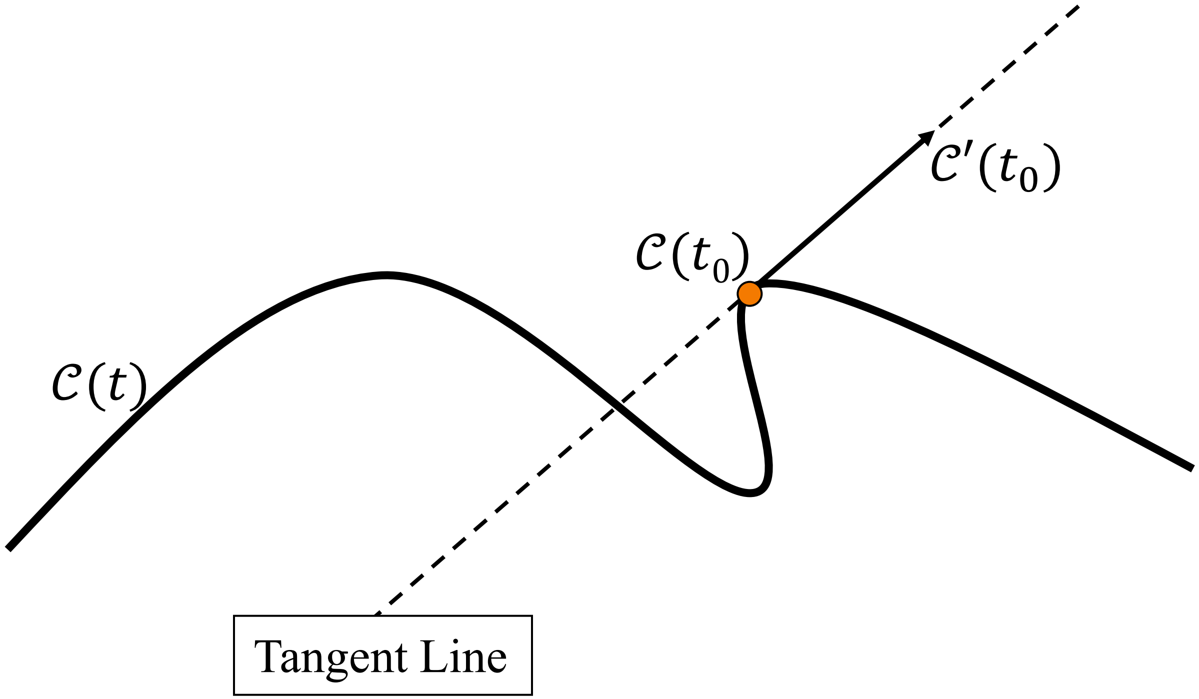
Given a curve $\mathcal{C}(t)$, the *tangent line* to the curve at a point $\mathbf{p}_0 = \mathcal{C}(t_0)$ is the line that most closely approximates the curve $\mathcal{C}(t)$ at $\mathbf{p}_0$.



Tangent Spaces

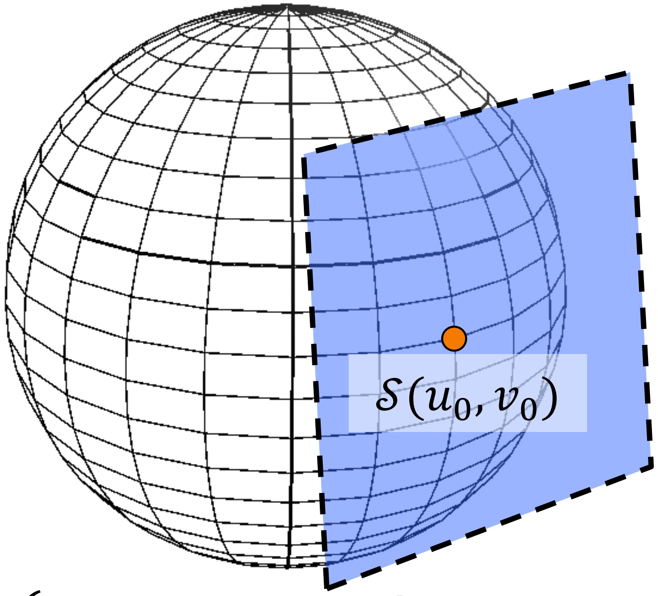
Given a curve $\mathcal{C}(t)$, the *tangent line* to the curve at a point $\mathbf{p}_0 = \mathcal{C}(t_0)$ is the line that most closely approximates the curve $\mathcal{C}(t)$ at $\mathbf{p}_0$.

This is the line through $\mathbf{p}_0$ with direction $\mathcal{C}'(t_0)$.



Tangent Spaces

Given a surface $\mathcal{S}(u, v)$ the *tangent plane* to the curve at a point $\mathbf{p}_0 = \mathcal{S}(u_0, v_0)$ is the plane that most closely approximates $\mathcal{S}(u, v)$ at $\mathbf{p}_0$.



$$\mathcal{S}(u, v) = (\cos u \cos v\,, \sin u\,, \cos u \sin v)$$

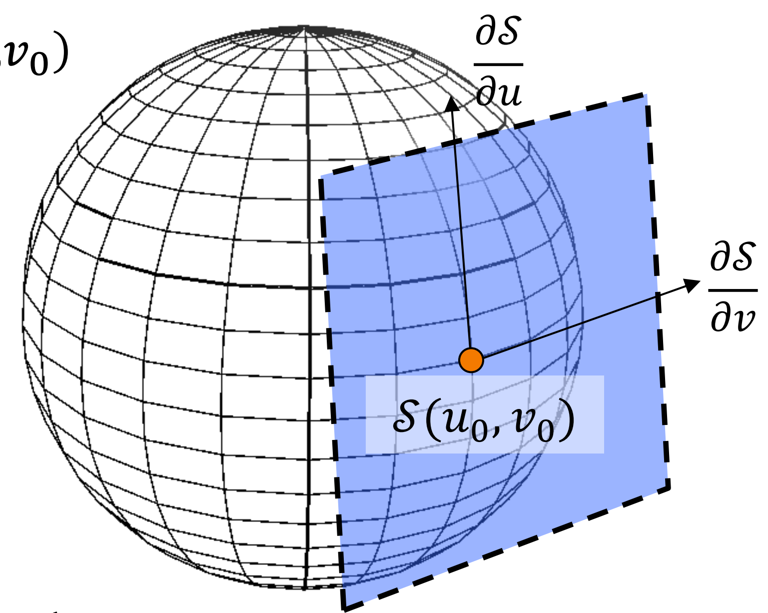
Tangent Spaces

Given a surface $\mathcal{S}(u,v)$ the *tangent plane* to the curve at a point $\mathbf{p}_0 = \mathcal{S}(u_0,v_0)$ is the plane that most closely approximates $\mathcal{S}(u,v)$ at $\mathbf{p}_0$.

This is the plane through $\mathbf{p}_0$, spanned by:

$$\left.\frac{\partial\mathcal{S}(u,v)}{\partial u}\right|_{(u_0,v_0)} \quad \text{and} \quad \left.\frac{\partial\mathcal{S}(u,v)}{\partial v}\right|_{(u_0,v_0)}$$

It describes the directions of motion along the surface from $\mathbf{p}_0$.



$$\mathcal{S}(u,v) = (\cos u \cos v, \sin u, \cos u \sin v)$$

Tangent Spaces

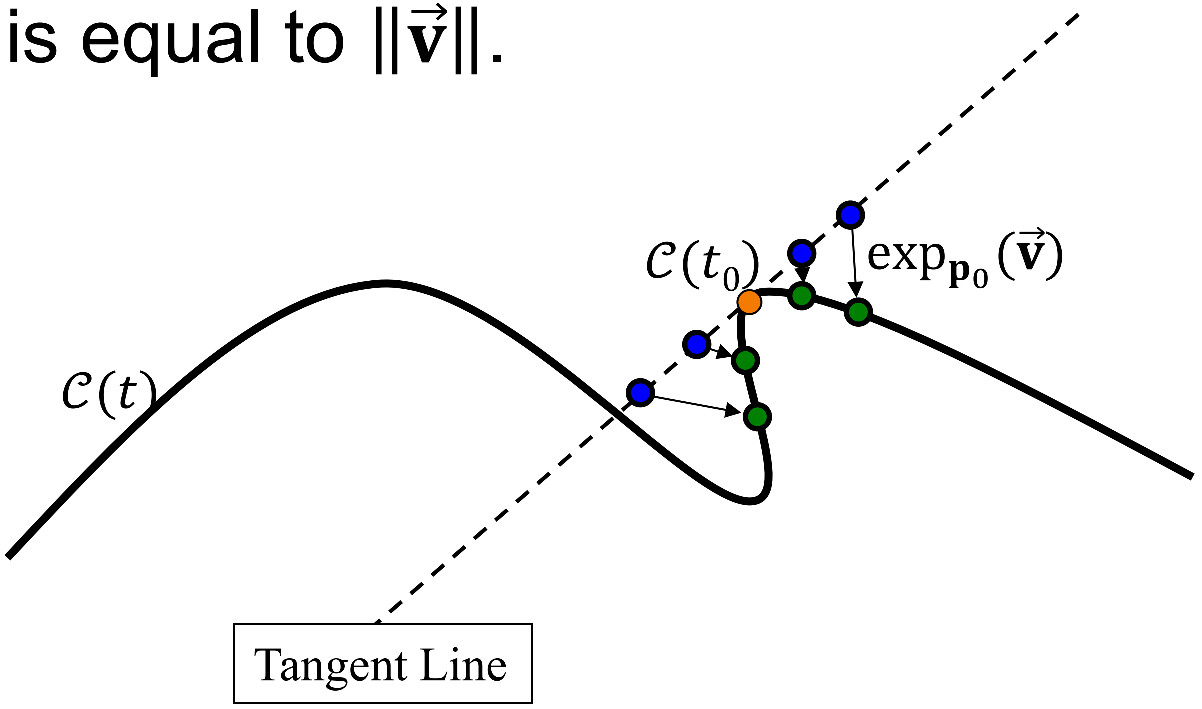
Given a manifold (a d-dimensional surface) the *tangent space* to the manifold at a point $\mathbf{p}_0$ on the manifold is the d-dimensional plane that most closely approximates the manifold at $\mathbf{p}_0$.

It describes the directions of motion along the manifold from $\mathbf{p}_0$.

The Exponential Map

Given a curve $\mathcal{C}(t)$, the *exponential* at $\mathbf{p}_0 = \mathcal{C}(t_0)$ is a map that sends points in the tangent space of $\mathbf{p}_0$ to the curve $\mathcal{C}(t)$.

The distance **along the curve** from $\mathbf{p}_0$ to point $\exp_{\mathbf{p}_0}(\vec{\mathbf{v}})$ is equal to $\|\vec{\mathbf{v}}\|$.



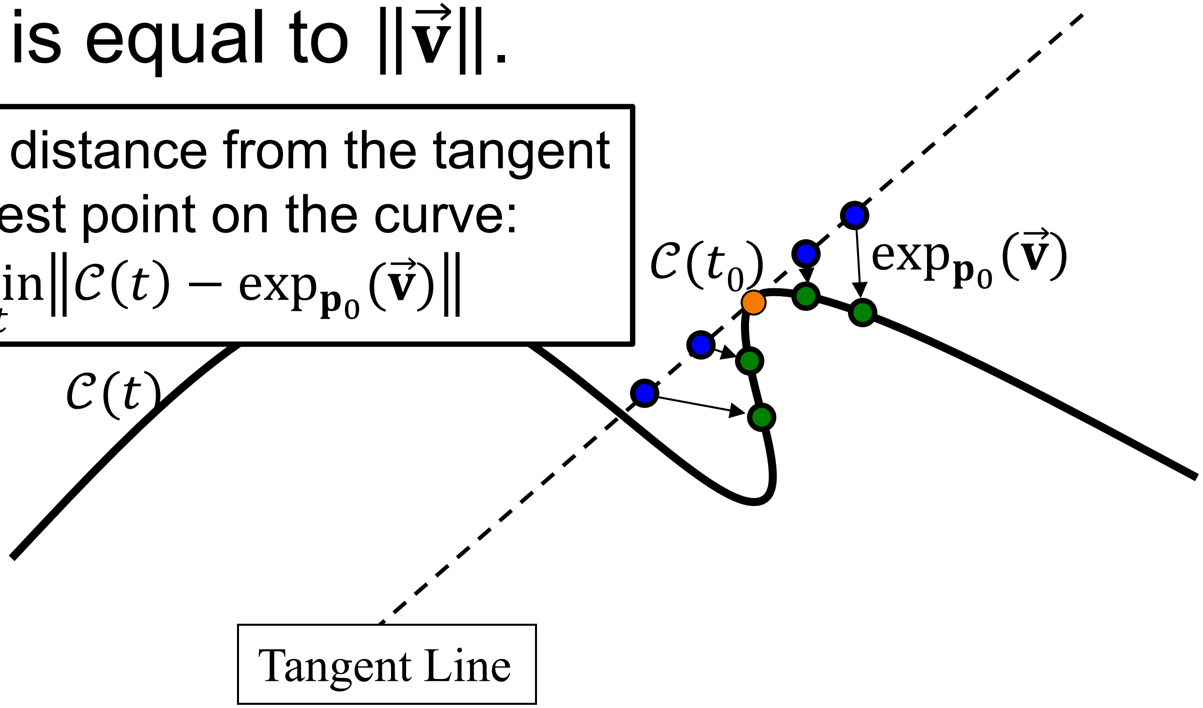
The Exponential Map

Given a curve $\mathcal{C}(t)$, the *exponential* at $\mathbf{p}_0 = \mathcal{C}(t_0)$ is a map that sends points in the tangent space of $\mathbf{p}_0$ to the curve $\mathcal{C}(t)$.

The distance **along the curve** from $\mathbf{p}_0$ to point $\exp_{\mathbf{p}_0}(\vec{\mathbf{v}})$ is equal to $\|\vec{\mathbf{v}}\|$.

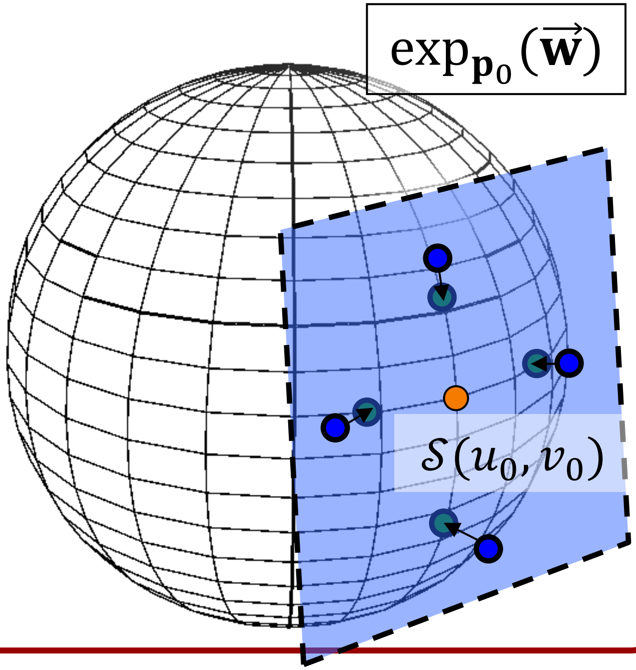
This is **not** the distance from the tangent line to the closest point on the curve:

$$\|\vec{\mathbf{v}}\| \neq \min_t \left\| \mathcal{C}(t) - \exp_{\mathbf{p}_0}(\vec{\mathbf{v}}) \right\|$$



The Exponential Map

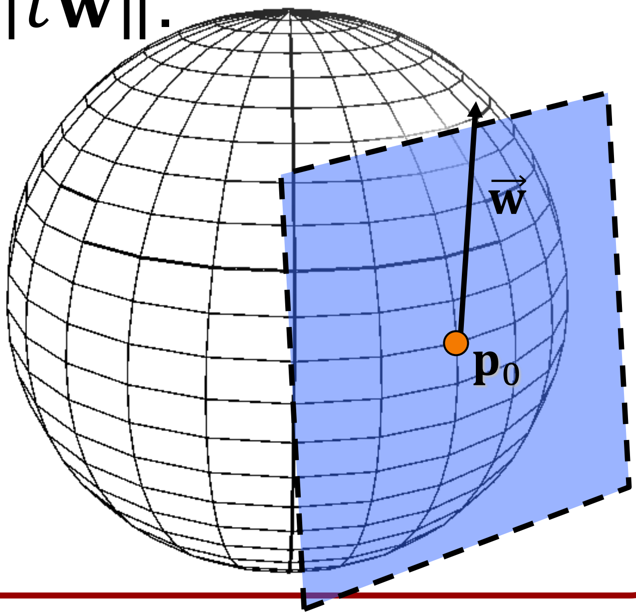
Given a surface $\mathcal{S}(u,v)$, the *exponential* at the point $\mathbf{p}_0 = \mathcal{S}(u_0,v_0)$ is a map that sends points in the tangent plane of $\mathbf{p}_0$ to the surface $\mathcal{S}(u,v)$.



The Exponential Map

Given a surface $\mathcal{S}(u,v)$, the *exponential* at the point $\mathbf{p}_0 = \mathcal{S}(u_0,v_0)$ is a map that sends points in the tangent plane of $\mathbf{p}_0$ to the surface $\mathcal{S}(u,v)$.

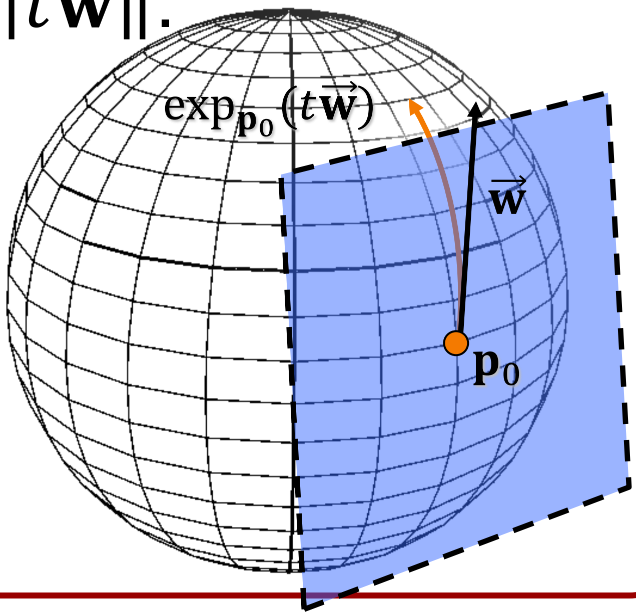
Fixing a vector $\vec{\mathbf{w}}$ in the tangent space at $\mathbf{p}_0$, the curve $\exp_{\mathbf{p}_0}(t\vec{\mathbf{w}})$ follows the geodesic leaving $\mathbf{p}_0$ in direction $\vec{\mathbf{w}}$, with length equal to $\|t\vec{\mathbf{w}}\|$.



The Exponential Map

Given a surface $\mathcal{S}(u,v)$, the *exponential* at the point $\mathbf{p}_0 = \mathcal{S}(u_0, v_0)$ is a map that sends points in the tangent plane of $\mathbf{p}_0$ to the surface $\mathcal{S}(u,v)$.

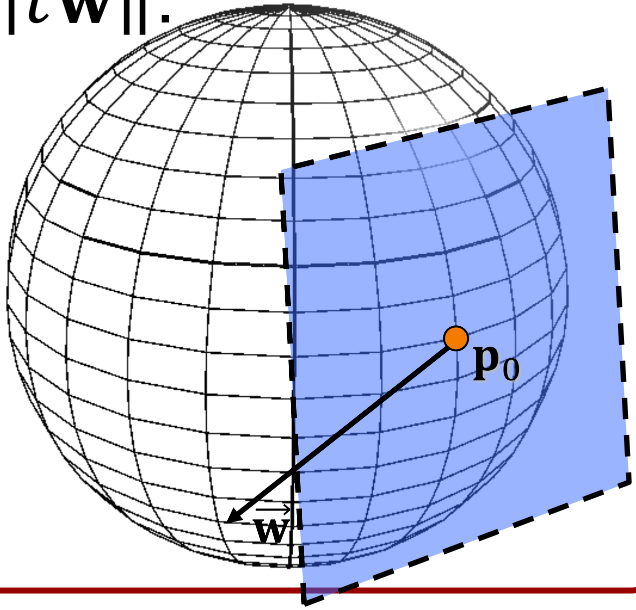
Fixing a vector $\vec{\mathbf{w}}$ in the tangent space at $\mathbf{p}_0$, the curve $\exp_{\mathbf{p}_0}(t\vec{\mathbf{w}})$ follows the geodesic leaving $\mathbf{p}_0$ in direction $\vec{\mathbf{w}}$, with length equal to $\|t\vec{\mathbf{w}}\|$.



The Exponential Map

Given a surface $\mathcal{S}(u,v)$, the *exponential* at the point $\mathbf{p}_0 = \mathcal{S}(u_0,v_0)$ is a map that sends points in the tangent plane of $\mathbf{p}_0$ to the surface $\mathcal{S}(u,v)$.

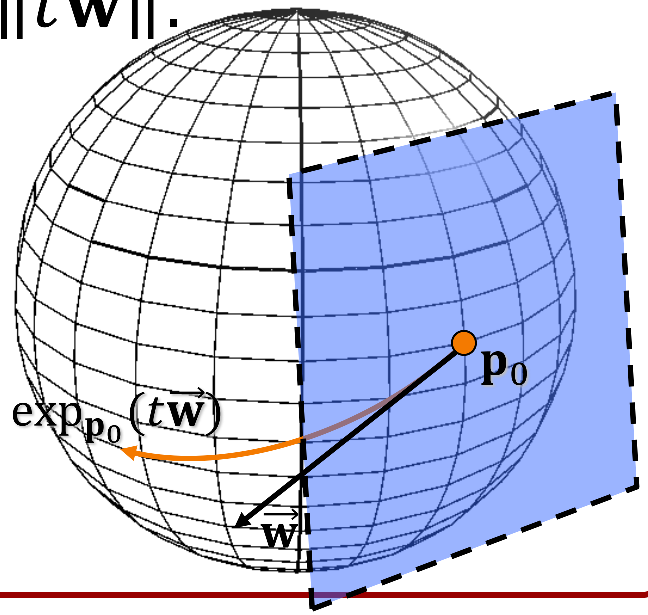
Fixing a vector $\vec{\mathbf{w}}$ in the tangent space at $\mathbf{p}_0$, the curve $\exp_{\mathbf{p}_0}(t\vec{\mathbf{w}})$ follows the geodesic leaving $\mathbf{p}_0$ in direction $\vec{\mathbf{w}}$, with length equal to $\|t\vec{\mathbf{w}}\|$.



The Exponential Map

Given a surface $\mathcal{S}(u,v)$, the *exponential* at the point $\mathbf{p}_0=\mathcal{S}(u_0,v_0)$ is a map that sends points in the tangent plane of $\mathbf{p}_0$ to the surface $\mathcal{S}(u,v)$.

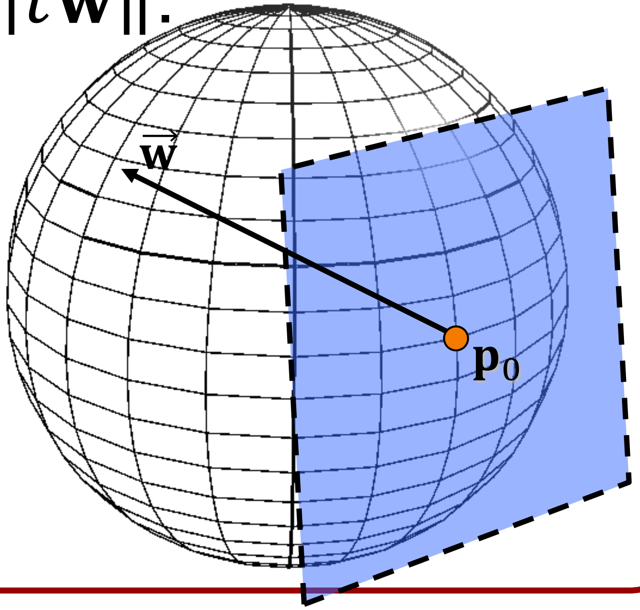
Fixing a vector $\vec{\mathbf{w}}$ in the tangent space at $\mathbf{p}_0$, the curve $\exp_{\mathbf{p}_0}(t\vec{\mathbf{w}})$ follows the geodesic leaving $\mathbf{p}_0$ in direction $\vec{\mathbf{w}}$, with length equal to $\|t\vec{\mathbf{w}}\|$.



The Exponential Map

Given a surface $\mathcal{S}(u,v)$, the *exponential* at the point $\mathbf{p}_0 = \mathcal{S}(u_0, v_0)$ is a map that sends points in the tangent plane of $\mathbf{p}_0$ to the surface $\mathcal{S}(u,v)$.

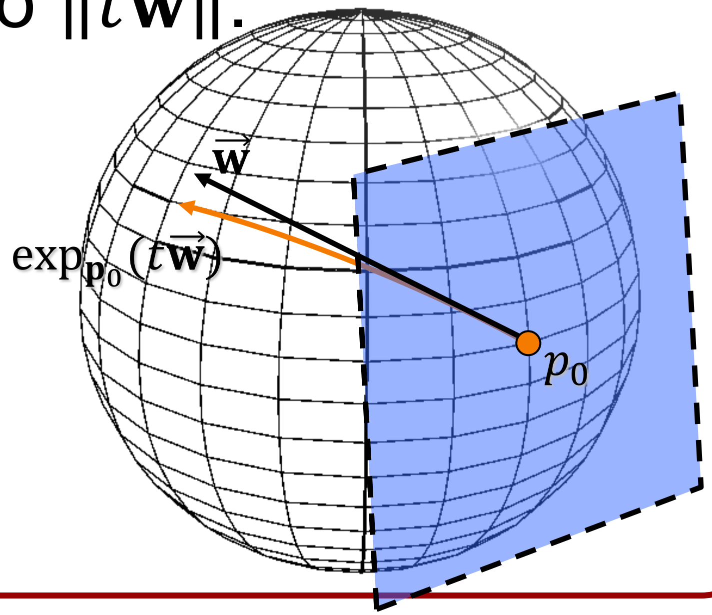
Fixing a vector $\vec{\mathbf{w}}$ in the tangent space at $\mathbf{p}_0$, the curve $\exp_{\mathbf{p}_0}(t\vec{\mathbf{w}})$ follows the geodesic leaving $\mathbf{p}_0$ in direction $\vec{\mathbf{w}}$, with length equal to $\|t\vec{\mathbf{w}}\|$.



The Exponential Map

Given a surface $\mathcal{S}(u,v)$, the *exponential* at the point $\mathbf{p}_0 = \mathcal{S}(u_0, v_0)$ is a map that sends points in the tangent plane of $\mathbf{p}_0$ to the surface $\mathcal{S}(u,v)$.

Fixing a vector $\vec{\mathbf{w}}$ in the tangent space at $\mathbf{p}_0$, the curve $\exp_{\mathbf{p}_0}(t\vec{\mathbf{w}})$ follows the geodesic leaving $\mathbf{p}_0$ in direction $\vec{\mathbf{w}}$, with length equal to $\|t\vec{\mathbf{w}}\|$.



The Exponential Map

Given a manifold (a d-dimensional surface), the *exponential* at point $\mathbf{p}_0$ on the manifold is a map that sends points in the tangent plane of $\mathbf{p}_0$ to the manifold.

Fixing a vector $\vec{\mathbf{w}}$ in the tangent space at $\mathbf{p}_0$, the curve $\exp_{\mathbf{p}_0}(t\vec{\mathbf{w}})$ follows the geodesic leaving $\mathbf{p}_0$ in direction $\vec{\mathbf{w}}$, with length equal to $\|t\vec{\mathbf{w}}\|$.

Answers the question:
Starting at a point $\mathbf{p}_0$, if we "walk" along the manifold in direction $\vec{\mathbf{w}}$ for time t, where do we end up?

The Logarithm Map

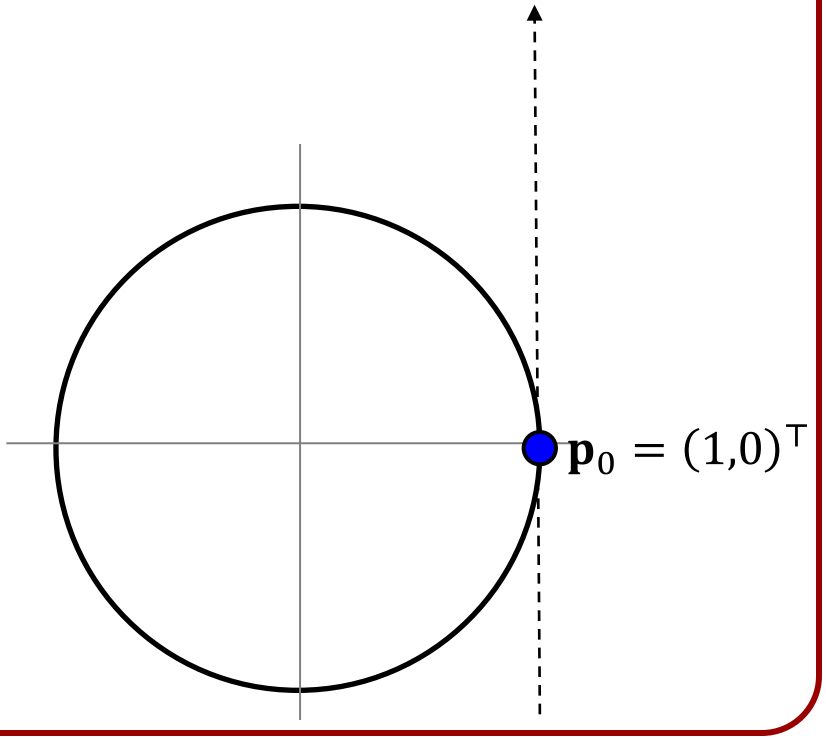
For a point $\mathbf{p}_0$ on a curve/surface/manifold, the *logarithm* is the inverse of the exponential, sending points on the curve/surface/manifold back into the tangent space of $\mathbf{p}_0$.

Answers the question:
Given a starting point $\mathbf{p}_0$, and some other point $\mathbf{p}$ on the manifold, what direction (and how long) do we need to walk from $\mathbf{p}_0$ to get to $\mathbf{p}$?

The Exponential Map

Example:

Let $\mathcal{C}$ be the unit circle, the tangent line at the point $\mathbf{p}_0 = (1,0)^\top$ is the vertical line through $\mathbf{p}_0$.



The Exponential Map

Example:

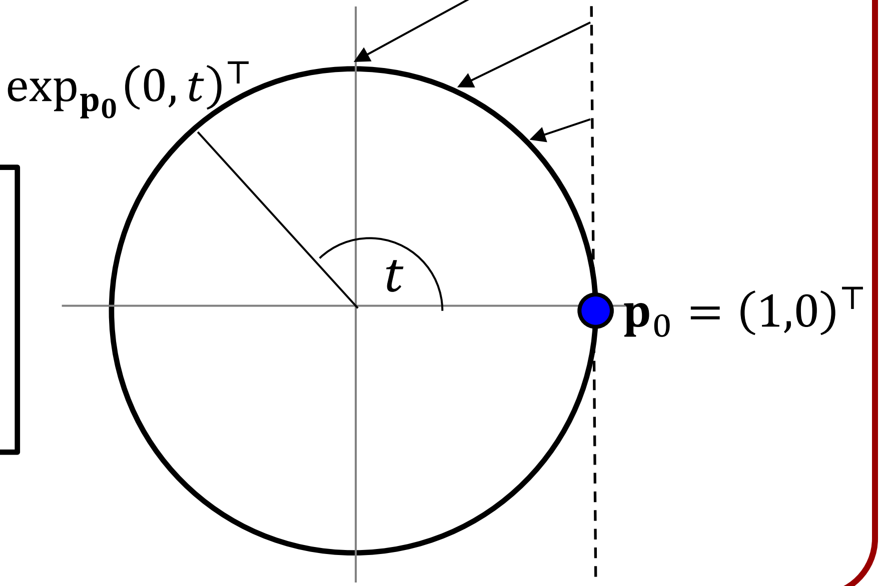
Let $\mathcal{C}$ be the unit circle, the tangent at $\mathbf{p}_0 = (1,0)^\top$ is the vertical line with direction $(0,t)^\top$.

The exponential map $\exp_{\mathbf{p}_0}$ sends the vector $(0,t)^\top$ on the tangent line to $(\cos t, \sin t)^\top$.

Note:
The exponential map is many-to-one:

$$\exp_{\mathbf{p}_0}(0,t)^\top = \exp_{\mathbf{p}_0}(0,t+2k\pi)^\top$$

so the logarithm is not unique.



The Exponential Map

Fact:

1. The tangent space to the manifold of $(n \times n)$ rotations at the identity is the is space of $(n \times n)$ skew-symmetric matrices.
2. The exponential map at the identity, $\exp_{\mathbf{id.}}$, sends skew-symmetric matrices to rotations.

The Exponential Map

How do we compute the exponential map?

The Exponential Map

How do we compute the exponential map?

Use the same Taylor series approximation:

$$\exp_{\mathbf{id.}}(\boldsymbol{S}) = \mathbf{id.} + \boldsymbol{S} + \frac{1}{2!}\boldsymbol{S}^2 + \cdots + \frac{1}{n!}\boldsymbol{S}^n + \cdots$$

Similarly for the logarithm:

$$\ln_{\mathbf{id.}}(\mathbf{R}) = (\mathbf{R} - \mathbf{id.}) - \frac{(\mathbf{R}-\mathbf{id.})^2}{2} + \cdots + (-1)^{n+1}\frac{(\mathbf{R}-\mathbf{id.})^n}{n} + \cdots$$

The Exponential Map

Properties:

$$\exp_{\mathbf{id.}}(0) = \mathbf{id}.$$

Starting at the identity and not going anywhere, keeps us at the identity.

$$\left.\frac{\partial \exp_{\mathbf{id.}}(t\boldsymbol{S})}{\partial t}\right|_{t=0} = \boldsymbol{S}$$

Following the geodesic from the identity in direction $\boldsymbol{S}$, our initial direction is $\boldsymbol{S}$.

$$\ln_{\mathbf{id.}}(\exp_{\mathbf{id.}}\boldsymbol{S}) = \boldsymbol{S}$$

The direction to travel from the identity to end up at the rotation we would get to by traveling in direction $\boldsymbol{S}$ is $\boldsymbol{S}$.

Rotation Interpolation/Approximation

Given a collection of rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ we can generate a curve passing through/near the matrices:

- For each $\mathbf{R}_i$, compute the logarithm $\mathbf{S}_i = \ln_{\mathbf{id.}}(\mathbf{R}_i)$

Rotation Interpolation/Approximation

Given a collection of rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ we can generate a curve passing through/near the matrices:

- For each $\mathbf{R}_i$, compute the logarithm $\mathbf{S}_i = \ln_{\mathbf{id.}}(\mathbf{R}_i)$
- Interpolate/Approximate the logarithms:

Rotation Interpolation/Approximation

Given a collection of rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ we can generate a curve passing through/near the matrices:

- For each $\mathbf{R}_i$, compute the logarithm $\mathbf{S}_i = \ln_{\mathbf{id.}}(\mathbf{R}_i)$
- Interpolate/Approximate the logarithms:
  - Linear Interpolation:

$$\mathbf{S}_k(t) = (1-t)\mathbf{S}_k + t\mathbf{S}_{k+1}$$

Rotation Interpolation/Approximation

Given a collection of rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ we can generate a curve passing through/near the matrices:

- For each $\mathbf{R}_i$, compute the logarithm $\mathbf{S}_i = \ln_{\mathbf{id.}}(\mathbf{R}_i)$
- Interpolate/Approximate the logarithms:
  - Linear Interpolation:
  - Catmull-Rom Interpolation:

$$\mathbf{S}_k(t) = CR_0(t)\mathbf{S}_{k-1} + CR_1(t)\mathbf{S}_k + CR_1(t)\mathbf{S}_{k+1} + CR_1(t)\mathbf{S}_{k+2}$$

Rotation Interpolation/Approximation

Given a collection of rotations $\{\mathbf{R}_0, \dots, \mathbf{R}_{n-1}\}$ we can generate a curve passing through/near the matrices:

- For each $\mathbf{R}_i$, compute the logarithm $\mathbf{S}_i = \ln_{\mathbf{id.}}(\mathbf{R}_i)$
- Interpolate/Approximate the logarithms:
  - Linear Interpolation:
  - Catmull-Rom Interpolation:
  - Uniform Cubic B-Spline Approximation:

$$\mathbf{S}_k(t) = B_{0,3}(t)\mathbf{S}_{k-1} + B_{1,3}(t)\mathbf{S}_k + B_{2,3}(t)\mathbf{S}_{k+1} + B_{3,3}(t)\mathbf{S}_{k+2}$$

Rotation Interpolation/Approximation

Given a collection of rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ we can generate a curve passing through/near the matrices:

- For each $\mathbf{R}_i$, compute the logarithm $\mathbf{S}_i = \ln_{\mathbf{id.}}(\mathbf{R}_i)$
- Interpolate/Approximate the logarithms:
  - Linear Interpolation:
  - Catmull-Rom Interpolation:
  - Uniform Cubic B-Spline Approximation:
- Set the value of the in-between rotation to be the exponent of the blended logarithms:

$$\mathbf{R}_k(t) = \exp_{\mathbf{id.}}\left(\mathbf{S}_k(t)\right)$$

Rotation Interpolation/Approximation

Given a collection of rotations $\{\mathbf{R}_0, \ldots, \mathbf{R}_{n-1}\}$ we can generate a curve passing through/near the matrices:

- For each $\mathbf{R}_i$, compute the logarithm $\mathbf{S}_i = \ln_{\mathbf{id.}}(\mathbf{R}_i)$
- Interpolate/Approximate the logarithms:

<u>Note</u>:
The logarithm of rotations is a skew-symmetric matrix and since skew-symmetric matrices are closed under addition and scaling.
⇒ The weighted average $\mathbf{S}_k(t)$ is also skew-symmetric.
⇒ Its exponent will be a rotation.

<u>Warning</u>:
Why take the exponential/logarithm with respect to the identity?
Maybe we should take it with respect to some other rotation?

Summary

To define in-between frames for an animation, we interpolate/approximate the transformations specified in the key-frames.

For translation, we can use splines

For rotations, we need to ensure that the in-between transformations are also rotations:

- Euler angles
- Exponential map

In-between transformations are guaranteed to be rotations

- SVD
- Quaternions

Normalize in-between transformations to turn them into the nearest rotations