



Quaternions and Exponentials

Michael Kazhdan
(600.357 / 600.457)

HB A.6

FvDFH 21.1.3



Overview

- Cross Products and (Skew) Symmetric Matrices
- Quaternions
- The Exponential Map



Cross Product

Given two 3D vectors $u=(u_1, u_2, u_3)$ and $v=(v_1, v_2, v_3)$ we can define the cross product of u and v as the vector:

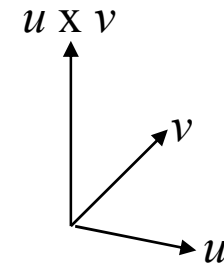
$$u \times v = \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix}$$



Cross Product

Given two 3D vectors $u=(u_1, u_2, u_3)$ and $v=(v_1, v_2, v_3)$ we can define the cross product of u and v as the vector:

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



Properties:

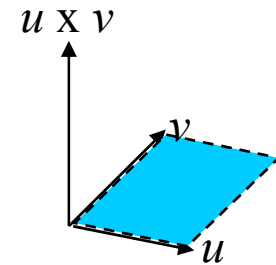
- The cross product is orthogonal to both u and v .
- The vectors u , v , $u \times v$ align with the right hand rule.



Cross Product

Given two 3D vectors $u=(u_1, u_2, u_3)$ and $v=(v_1, v_2, v_3)$ we can define the cross product of u and v as the vector:

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



Properties:

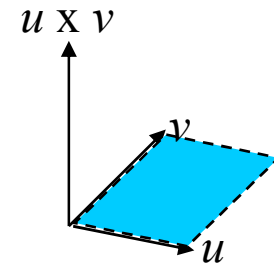
- The cross product is orthogonal to both u and v .
- The vectors u , v , $u \times v$ align with the right hand rule.
- The length of the cross product is equal to the area of the parallelogram defined by u and v .



Cross Product

Given two 3D vectors $u=(u_1, u_2, u_3)$ and $v=(v_1, v_2, v_3)$ we can define the cross product of u and v as the vector:

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



Properties:

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



(Skew) Symmetric Matrices

A matrix M is symmetric if:

$$M_{ij} = M_{ji} \iff M = M^t$$



(Skew) Symmetric Matrices

A matrix M is symmetric if:

$$M_{ij} = M_{ji} \iff M = M^t$$

A matrix M is skew-symmetric if:

$$M_{ij} = -M_{ji} \iff M = -M^t$$



(Skew) Symmetric Matrices

A matrix M is symmetric if:

$$M_{ij} = M_{ji} \iff M = M^t$$

A matrix M is skew-symmetric if:

$$M_{ij} = -M_{ji} \iff M = -M^t$$

(Skew) Symmetric matrices are closed under addition and scaling:

- If $A=A^t$ and $B=B^t$, then $(A+B)=(A+B)^t$.
- If $A=A^t$ then $(\alpha A)=(\alpha A)^t$.
- If $A=-A^t$ and $B=-B^t$, then $(A+B)=-(\A+B)^t$.
- If $A=-A^t$ then $(\alpha A)=-(\alpha A)^t$.



Overview

- Cross Products and (Skew) Symmetric Matrices
- Quaternions
- The Exponential Map



Quaternions

Quaternions are extensions of complex numbers,
with 3 imaginary values instead of 1:

$$a + ib + jc + kd$$



Quaternions

Quaternions are extensions of complex numbers, with 3 imaginary values instead of 1:

$$a + ib + jc + kd$$

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

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



Quaternions

Quaternions are extensions of complex numbers, with 3 imaginary values instead of 1:

$$a + ib + jc + kd$$

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

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



Quaternions

Quaternions are extensions of complex numbers, with 3 imaginary values instead of 1:

$$a + ib + jc + kd$$

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

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

However, the multiplication rules are more complex:

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



Quaternions

Quaternions are extensions of complex numbers, with 3 imaginary values instead of 1:

$$a + ib + jc + kd$$

Like the complex numbers, the square of a quaternion is a real number.

Note that multiplication of quaternions is not commutative:
The result of the multiplication depends on the order in which it was done

However, the multiplication rules are more complex:

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



Quaternions

More generally, the product of two quaternions is:

$$\begin{aligned} & (a_1 + ib_1 + jc_1 + kd_1) \\ & \times (a_2 + ib_2 + jc_2 + kd_2) \\ \hline & = (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) \\ & + i(a_1b_2 + a_2b_1 + c_1d_2 - c_2d_1) \\ & + j(a_1c_2 + a_2c_1 - b_1d_2 + b_2d_1) \\ & + k(a_1d_2 + a_2d_1 + b_1c_2 - b_2c_1) \end{aligned}$$

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

$$ij = k \quad ik = -j \quad jk = i$$

$$ji = -k \quad ki = j \quad kj = -i$$



Quaternions

As with complex numbers, we define the conjugate of a quaternion $q = a + ib + jc + kd$ as:

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



Quaternions

As with complex numbers, we define the conjugate of a quaternion $q=a+ib+jc+kd$ as:

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

As with complex numbers, we define the norm of a quaternion $q=a+ib+jc+kd$ as:

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



Quaternions

As with complex numbers, we define the conjugate of a quaternion $q=a+ib+jc+kd$ as:

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

As with complex numbers, we define the norm of a quaternion $q=a+ib+jc+kd$ as:

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

As with complex numbers, the reciprocal is defined by dividing the conjugate by the square norm:

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



Quaternions

One way to express a quaternion is as a pair consisting of the real value and the 3D vector consisting of the imaginary components:

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



Quaternions

One way to express a quaternion is as a pair consisting of the real value and the 3D vector consisting of the imaginary components:

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

The advantage of this representation is that it is easier to express quaternion multiplication:

$$\begin{aligned} q_1 q_2 &= (\alpha_1, w_1)(\alpha_2, w_2) \\ &= (\alpha_1 \alpha_2 - \langle w_1, w_2 \rangle, \alpha_1 w_2 + \alpha_2 w_1 + w_1 \times w_2) \end{aligned}$$



Quaternion

$$q_1 q_2 = (a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2)$$

$$+ i(a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1)$$

$$+ j(a_1 c_2 + a_2 c_1 - b_1 d_2 + b_2 d_1)$$

$$+ k(a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1)$$

One way
consisti
consisti

a pair
D vector
nts:

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

The advantage of this representation is that it is easier to express quaternion multiplication:

$$q_1 q_2 = (\alpha_1, w_1)(\alpha_2, w_2)$$

$$= (\alpha_1 \alpha_2 - \langle w_1, w_2 \rangle, \alpha_1 w_2 + \alpha_2 w_1 + w_1 \times w_2)$$



Quaternions and Rotations

If $q = a + ib + jc + kd$ is a unit quaternion ($\|q\| = 1$), q corresponds to a rotation:

$$R(q) = \begin{pmatrix} 1 - 2c^2 - 2d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & 1 - 2b^2 - 2d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & 1 - 2b^2 - 2c^2 \end{pmatrix}$$

Note that because all of the terms are quadratic, the rotation associated with q is the same as the rotation associated with $-q$.



Quaternions and Rotations

If $q = a + ib + jc + kd$ is a unit quaternion ($\|q\| = 1$), q corresponds to a rotation:

$$R(q) = \begin{pmatrix} 1 - 2c^2 - 2d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & 1 - 2b^2 - 2d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & 1 - 2b^2 - 2c^2 \end{pmatrix}$$

Because q is a unit quaternion, we can write q as:

$$q = (\cos(\theta/2), \sin(\theta/2)w) \quad \|w\| = 1$$



Quaternions and Rotations

If $q = a + ib + jc + kd$ is a unit quaternion ($\|q\| = 1$), q corresponds to a rotation:

$$R(q) = \begin{pmatrix} 1 - 2c^2 - 2d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & 1 - 2b^2 - 2d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & 1 - 2b^2 - 2c^2 \end{pmatrix}$$

Because q is a unit quaternion, we can write q as:

$$q = (\cos(\theta/2), \sin(\theta/2)w) \quad \|w\| = 1$$

It turns out that q corresponds to the rotation whose:

- axis of rotation is w , and
- angle of rotation is θ .



Quaternions

Instead of blending matrices and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each M_i , compute the quaternion rep. (α_i, w_i)



Quaternions

Instead of blending matrices and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each M_i , compute the quaternion rep. (α_i, w_i)
- Interpolate/Approximate the quaternions:



Quaternions

Instead of blending matrices and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each M_i , compute the quaternion rep. (α_i, w_i)
- Interpolate/Approximate the quaternions:

» **Linear Interpolation:**

$$\alpha_i(t) = (1-t)\alpha_i + t\alpha_{i+1}$$

$$w_i(t) = (1-t)w_i + tw_{i+1}$$



Quaternions

Instead of blending matrices and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each M_i , compute the quaternion rep. (α_i, w_i)
- Interpolate/Approximate the quaternions:
 - » Linear Interpolation
 - » Catmull-Rom Interpolation:

$$\alpha_i(t) = CR_0(t)\alpha_{k-1} + CR_1(t)\alpha_k + CR_2(t)\alpha_{k+1} + CR_3(t)\alpha_{k+2}$$

$$w_i(t) = CR_0(t)w_{k-1} + CR_1(t)w_k + CR_2(t)w_{k+1} + CR_3(t)w_{k+2}$$



Quaternions

Instead of blending matrices and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each M_i , compute the quaternion rep. (α_i, w_i)
- Interpolate/Approximate the quaternions:

» Linear Interpolation

» Catmull-Rom Interpolation

» Uniform Cubic B-Spline Approximation:

$$\alpha_j(t) = B_{0,3}(t)\alpha_{k-1} + B_{1,3}(t)\alpha_k + B_{2,3}(t)\alpha_{k+1} + B_{3,3}(t)\alpha_{k+2}$$

$$w_j(t) = B_{0,3}(t)w_{k-1} + B_{1,3}(t)w_k + B_{2,3}(t)w_{k+1} + B_{3,3}(t)w_{k+2}$$



Quaternions

Instead of blending matrices and then normalizing using SVD, we can blend the quaternions and then normalize them:

- For each M_i , compute the quaternion rep. (α_i, 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:

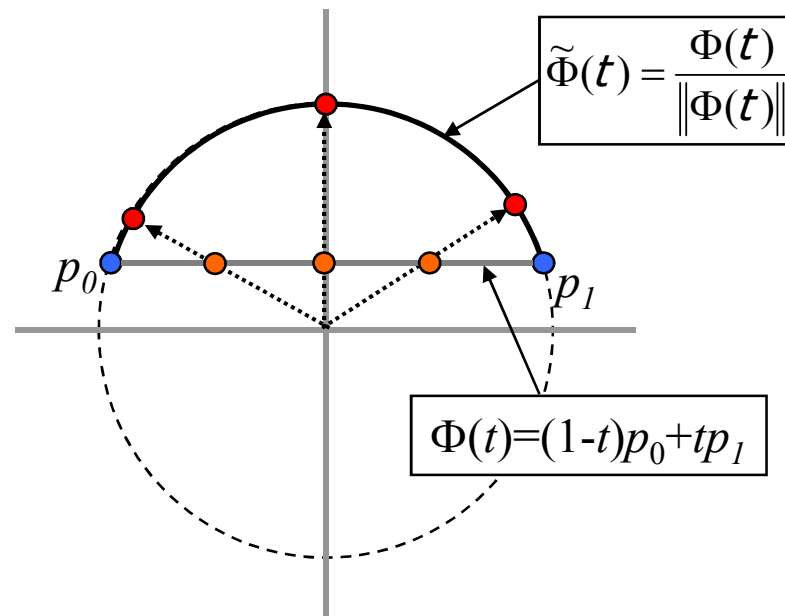
$$\Phi_i(t) = \frac{(\alpha_i(t), w_i(t))}{\|(\alpha_i(t), w_i(t))\|}$$



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.





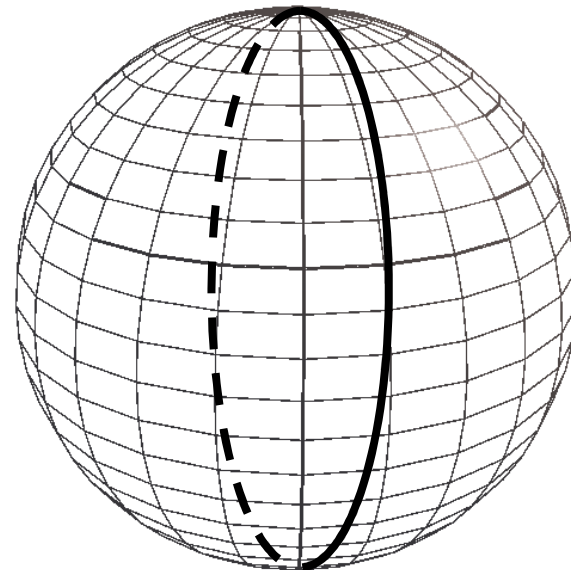
Overview

- Cross Products and (Skew) Symmetric Matrices
- Quaternions
- The Exponential Map



Geodesics

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



$$S(u,v) = (\cos(u) \cos(v), \sin(u), \cos(u) \sin(v))$$



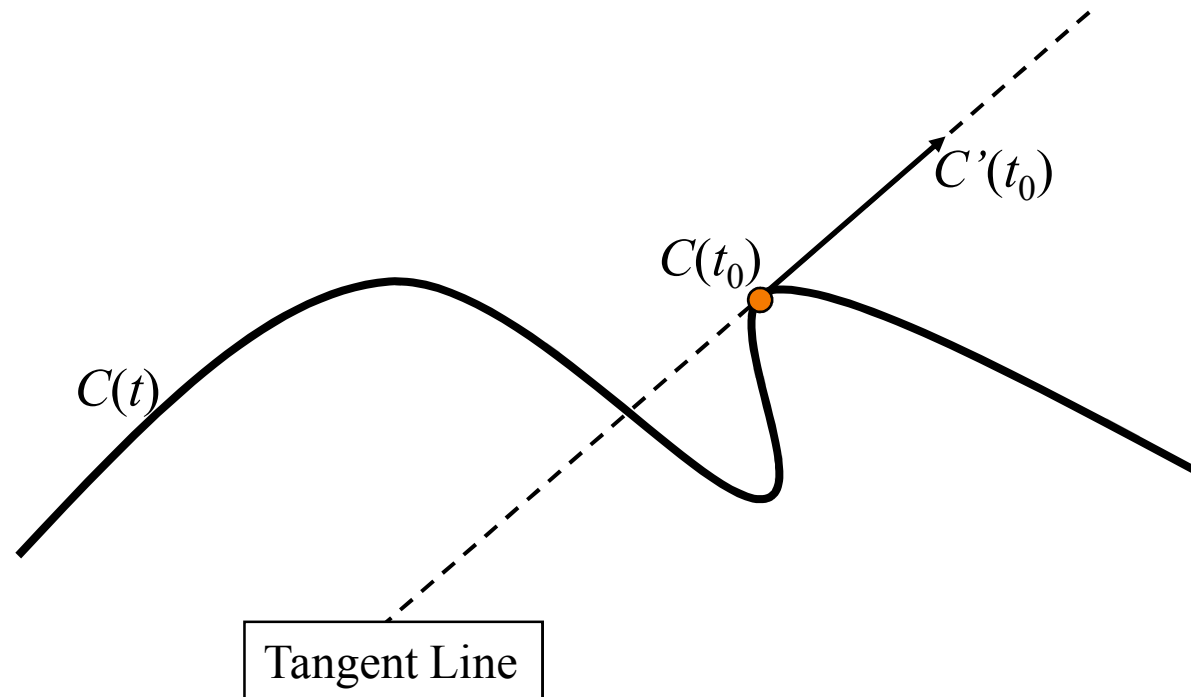
Geodesics

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 $C(t)$, the tangent line to the curve at a point $p_0 = C(t_0)$ is the line passing through p_0 with direction $C'(t_0)$.

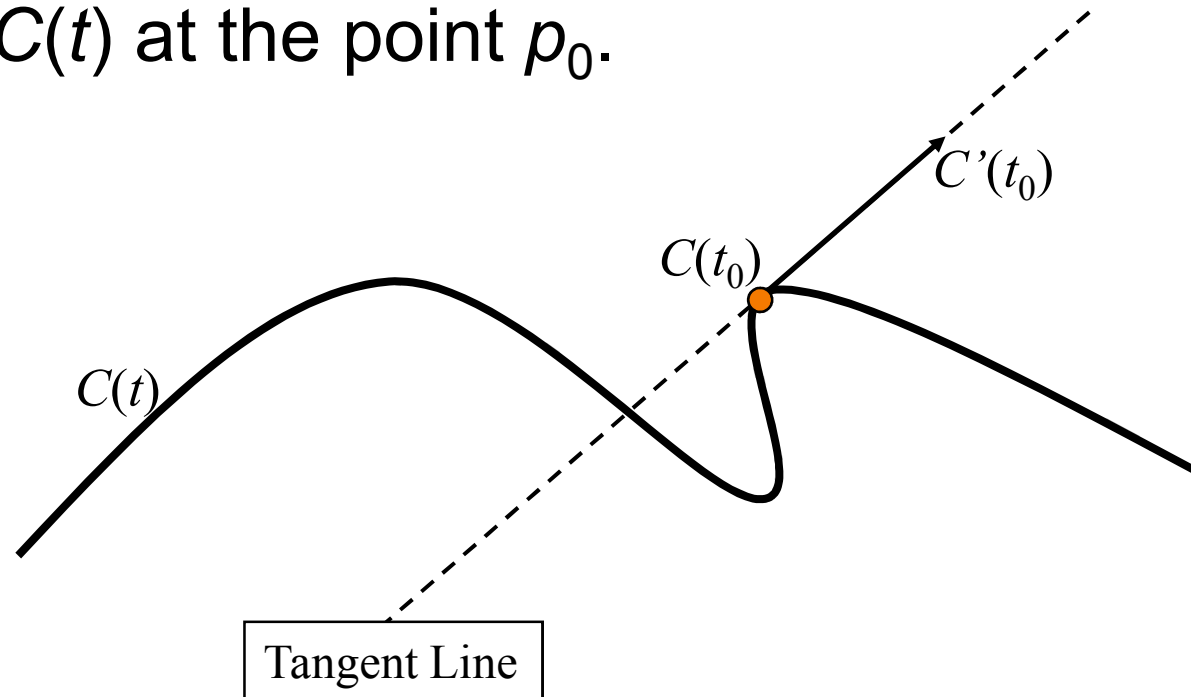




Tangent Spaces

Given a curve $C(t)$, the tangent line to the curve at a point $p_0 = C(t_0)$ is the line passing through p_0 with direction $C'(t_0)$.

This is the line that most closely approximates the curve $C(t)$ at the point p_0 .

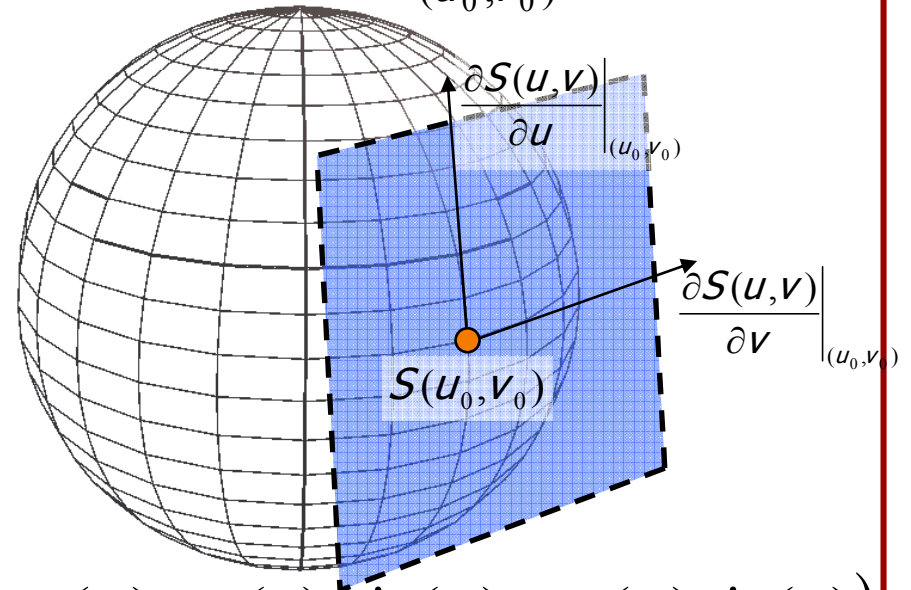




Tangent Spaces

Given a surface $S(u,v)$ the tangent plane to the curve at a point $p_0 = S(u_0, v_0)$ is the plane passing through p_0 , parallel to the plane spanned by:

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



$$S(u,v) = (\cos(u) \cos(v), \sin(u), \cos(u) \sin(v))$$

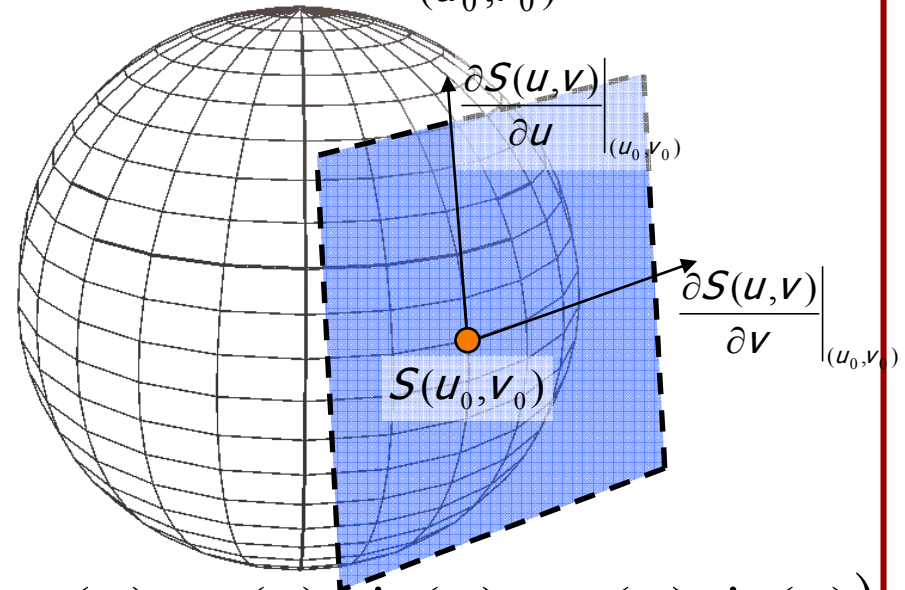


Tangent Spaces

Given a surface $S(u,v)$ the tangent plane to the curve at a point $p_0 = S(u_0, v_0)$ is the plane passing through p_0 , parallel to the plane spanned by:

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

This is the plane that most closely approximates $S(u,v)$ at the point p_0 .



$$S(u,v) = (\cos(u) \cos(v), \sin(u), \cos(u) \sin(v))$$



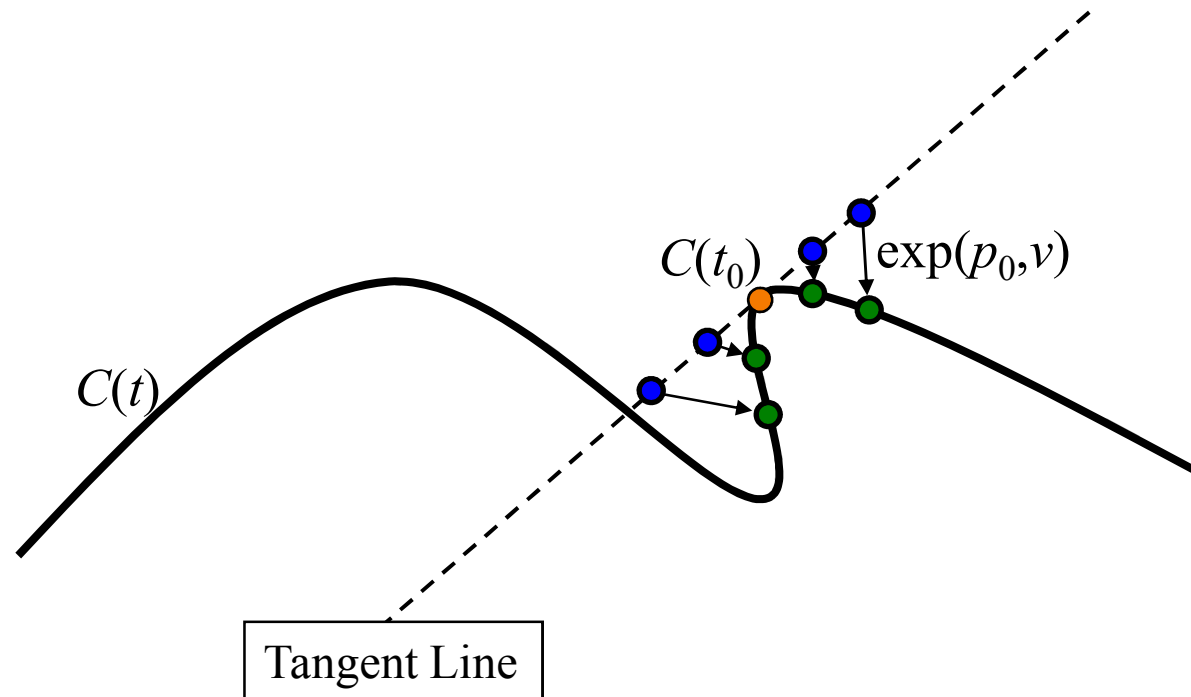
Tangent Spaces

Given a manifold (a d -dimensional surface) the tangent space to the manifold at a point p_0 on the manifold is the d -dimensional plane that most closely approximates the manifold at the point p_0 .



The Exponential Map

Given a curve $C(t)$, the exponential map at $p_0 = C(t_0)$ is a map that sends points in the tangent space of p_0 to the curve $C(t)$.

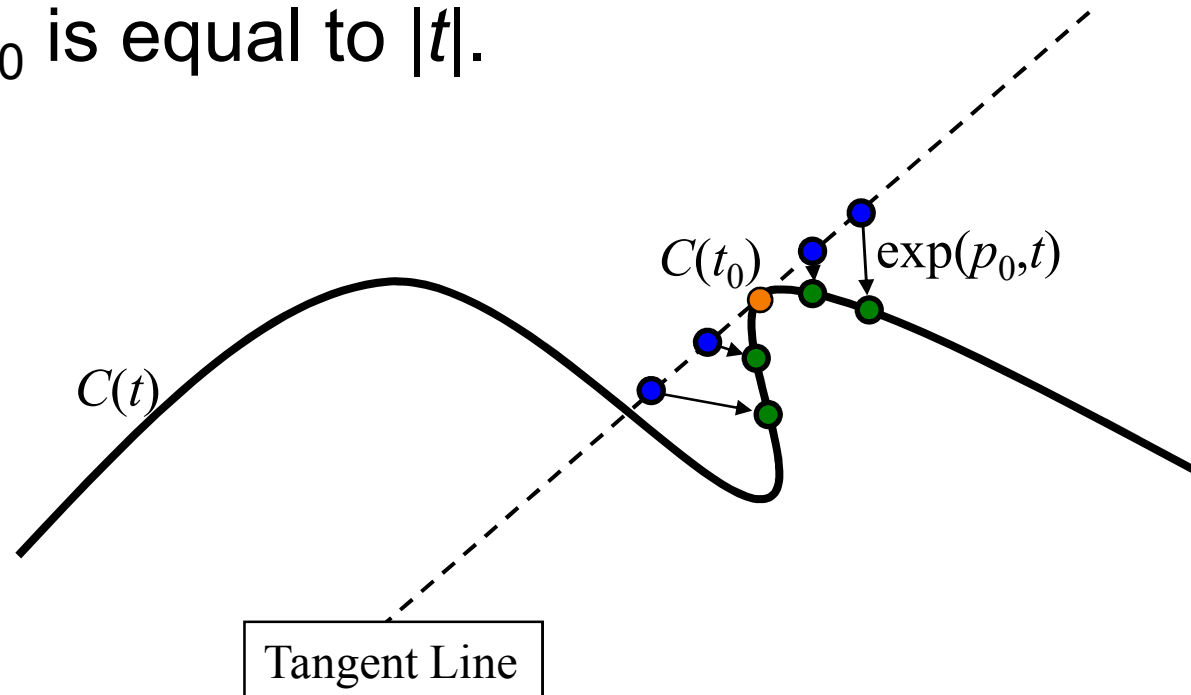




The Exponential Map

Given a curve $C(t)$, the exponential map at $p_0 = C(t_0)$ is a map that sends points in the tangent space of p_0 to the curve $C(t)$.

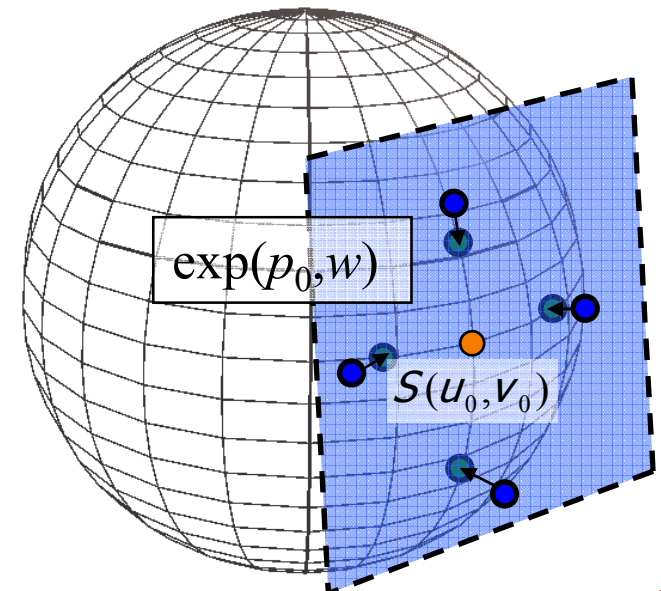
The distance along the curve of a point $\exp(p_0, t)$ from p_0 is equal to $|t|$.





The Exponential Map

Given a surface $S(u,v)$, the exponential map at $p_0 = S(u_0, v_0)$ is a map that sends points in the tangent plane of p_0 to the surface $S(u,v)$.

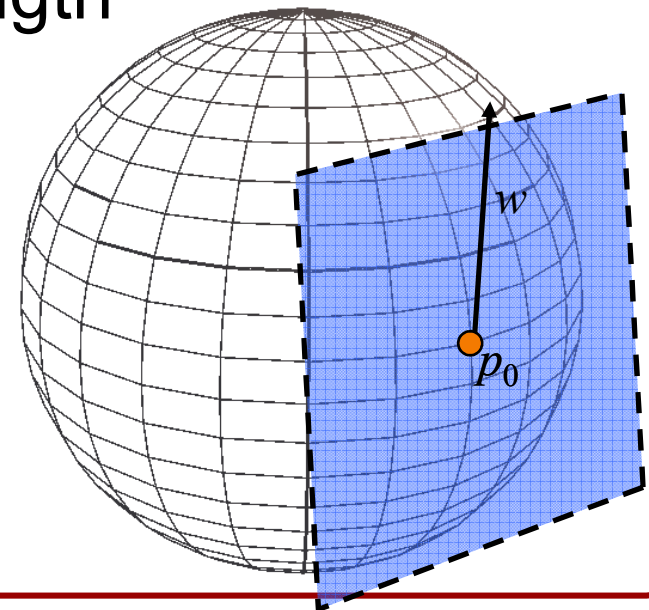




The Exponential Map

Given a surface $S(u,v)$, the exponential map at $p_0 = S(u_0, v_0)$ is a map that sends points in the tangent plane of p_0 to the surface $S(u,v)$.

If we fix a vector w in the tangent space of p_0 , then the curve $\exp(p_0, tw)$ will be a geodesic, leaving p_0 in direction w and will have length equal to $|tw|$.

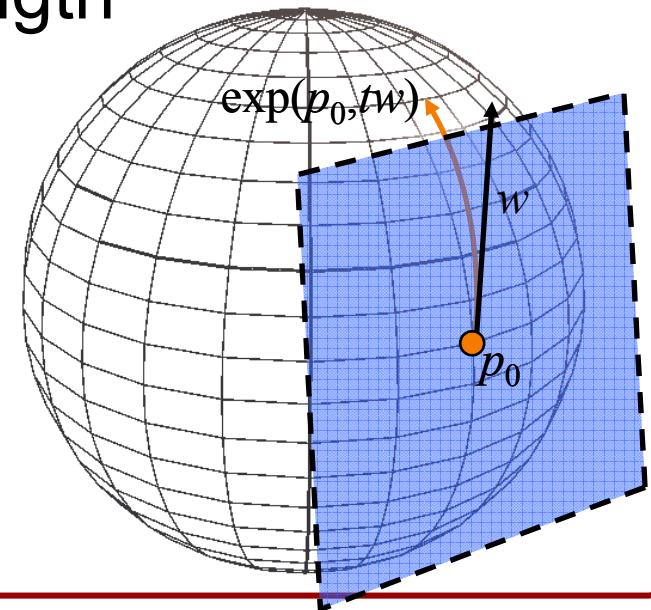




The Exponential Map

Given a surface $S(u,v)$, the exponential map at $p_0 = S(u_0, v_0)$ is a map that sends points in the tangent plane of p_0 to the surface $S(u,v)$.

If we fix a vector w in the tangent space of p_0 , then the curve $\exp(p_0, tw)$ will be a geodesic, leaving p_0 in direction w and will have length equal to $|tw|$.

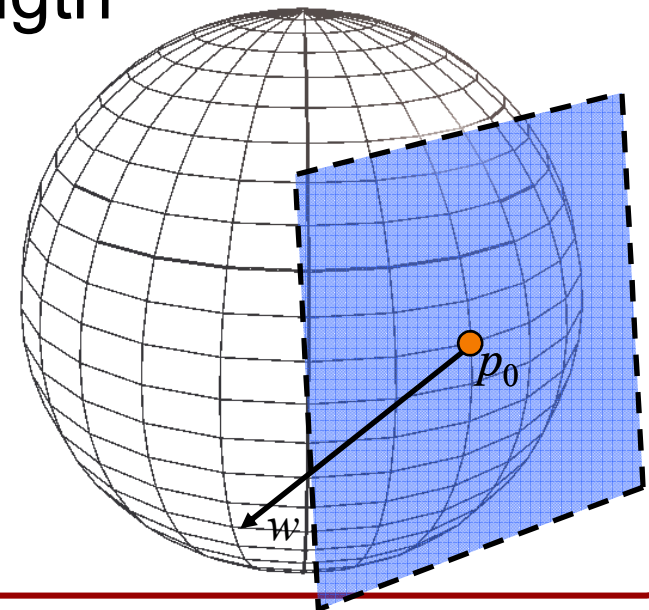




The Exponential Map

Given a surface $S(u,v)$, the exponential map at $p_0 = S(u_0, v_0)$ is a map that sends points in the tangent plane of p_0 to the surface $S(u,v)$.

If we fix a vector w in the tangent space of p_0 , then the curve $\exp(p_0, tw)$ will be a geodesic, leaving p_0 in direction w and will have length equal to $|tw|$.

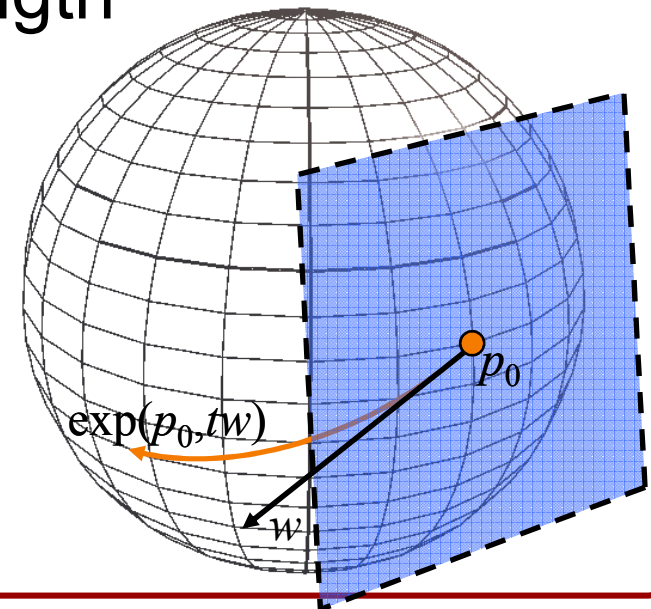




The Exponential Map

Given a surface $S(u,v)$, the exponential map at $p_0 = S(u_0, v_0)$ is a map that sends points in the tangent plane of p_0 to the surface $S(u,v)$.

If we fix a vector w in the tangent space of p_0 , then the curve $\exp(p_0, tw)$ will be a geodesic, leaving p_0 in direction w and will have length equal to $|tw|$.

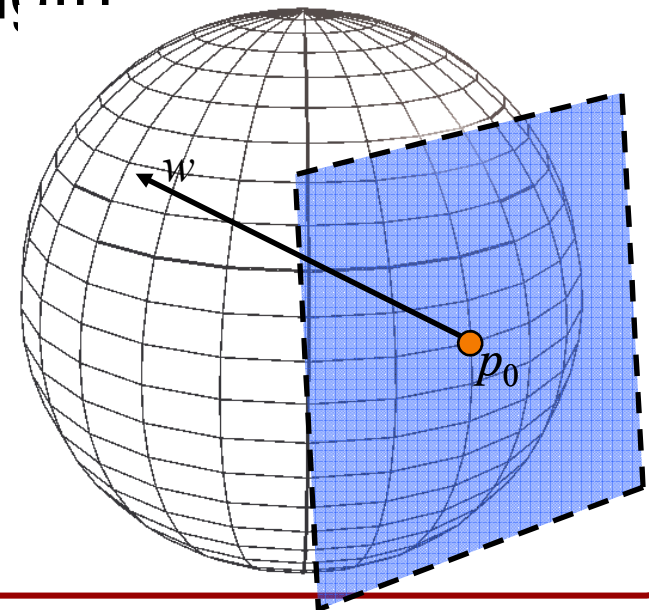




The Exponential Map

Given a surface $S(u,v)$, the exponential map at $p_0 = S(u_0, v_0)$ is a map that sends points in the tangent plane of p_0 to the surface $S(u,v)$.

If we fix a vector w in the tangent space of p_0 , then the curve $\exp(p_0, tw)$ will be a geodesic, leaving p_0 in direction w and will have length equal to $|tw|$.

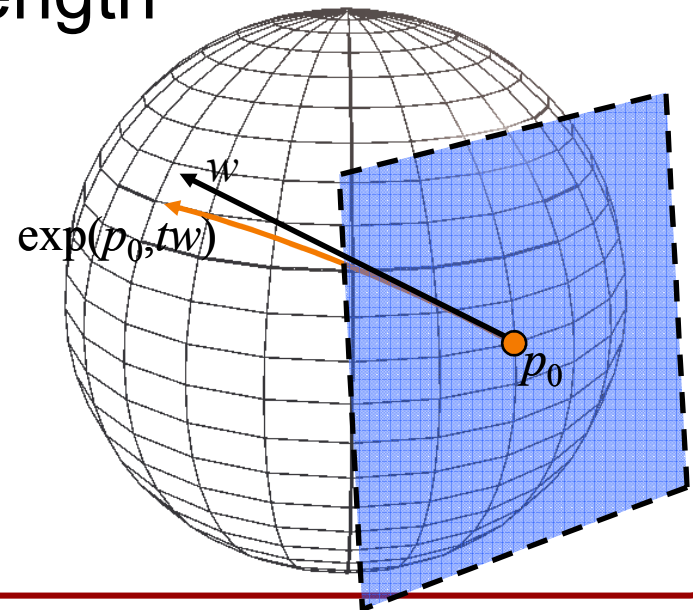




The Exponential Map

Given a surface $S(u,v)$, the exponential map at $p_0 = S(u_0, v_0)$ is a map that sends points in the tangent plane of p_0 to the surface $S(u,v)$.

If we fix a vector w in the tangent space of p_0 , then the curve $\exp(p_0, tw)$ will be a geodesic, leaving p_0 in direction w and will have length equal to $|tw|$.





The Exponential Map

Given a manifold (a d -dimensional surface), the exponential map at p_0 on the manifold is a map that sends points in the tangent plane of p_0 to the manifold.

If we fix a vector w in the tangent space of p_0 , then the curve $\exp(p_0, tw)$ will be a geodesic, leaving p_0 in direction w and will have length equal to $|tw|$.



The Logarithm Map

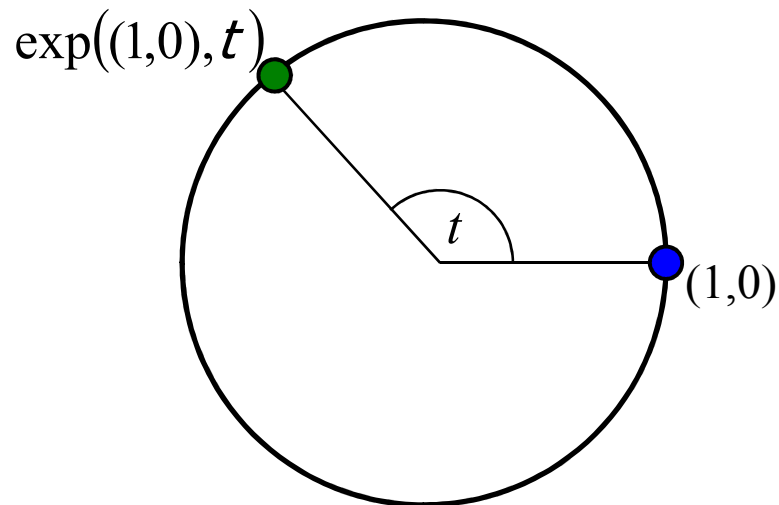
For a point p_0 on a curve/surface/manifold, the logarithm is the inverse of the exponential, sending points on the curve/surface/manifold back into the tangent space of p_0 .



The Exponential Map

Example:

Let C be the unit circle, the exponential map $\exp((1,0),t)$ is the map sending the point t to the point $(\cos(t), \sin(t))$.





The Exponential Map

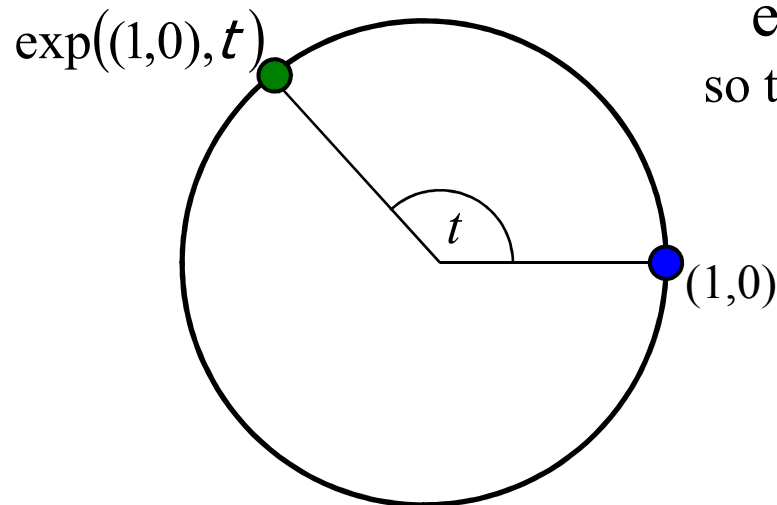
Example:

Let C be the unit circle, the exponential map $\exp((1,0), t)$ is the map sending the point t to the point $(\cos(t), \sin(t))$.

Note:

The exponential map is many-to-one:

$\exp((1,0), t) = \exp((1,0), t + 2k\pi)$
so the logarithm is not unique.





The Exponential Map

Example:

Let $GL(n)$ be the space of $n \times n$ invertible matrices, the exponential map $\exp(\text{Id}, M)$ is the map sending an arbitrary matrix M to an invertible one.



The Exponential Map

Example:

Let $GL(n)$ be the space of $n \times n$ invertible matrices, the exponential map $\exp(\text{Id}, M)$ is the map sending an arbitrary matrix M to an invertible one.

Let $SO(n)$ be the space of $n \times n$ rotation matrices, the exponential map $\exp(\text{Id}, M)$ is the map sending a skew-symmetric matrix M to a rotation.

The Exponential Map



How do we actually compute the exponential map?



The Exponential Map

How do we actually compute the exponential map?

It is difficult to find a closed form solution, but for matrices we can use a Taylor series approximation:

$$\exp(\text{Id}, A) = \exp(A) = \text{Id} + A + \frac{1}{2!} A^2 + \frac{1}{3!} A^3 + \dots + \frac{1}{n!} A^n$$



The Exponential Map

How do we actually compute the exponential map?

It is difficult to find a closed form solution, but for matrices we can use a Taylor series approximation:

$$\exp(\text{Id}, A) = \exp(A) = \text{Id} + A + \frac{1}{2!} A^2 + \frac{1}{3!} A^3 + \dots + \frac{1}{n!} A^n$$

In a similar manner, we can define the logarithm:

$$\ln(\text{Id}, A) = \ln(A) = (A - \text{Id}) - \frac{(A - \text{Id})^2}{2} + \frac{(A - \text{Id})^3}{3} + \dots + (-1)^{(n+1)} \frac{(A - \text{Id})^n}{n}$$



The Exponential Map

Properties:

- $\exp(0) = Id$
- $\left. \frac{\partial \exp(tA)}{\partial t} \right|_0 = A$
- $\exp(\ln(A)) = A$
- $\exp(\alpha A) = [\exp(A)]^\alpha$
- $\exp(A+B) = \exp(A) \exp(B)$ if and only if $AB=BA$



Rotation Interpolation/Approximation

Given a collection of rotations $\{M_0, \dots, M_{n-1}\}$ we can generate a curve passing through/near the matrices in the following manner:

- For each M_i , compute the logarithm $\ln(M_i)$



Rotation Interpolation/Approximation

Given a collection of rotations $\{M_0, \dots, M_{n-1}\}$ we can generate a curve passing through/near the matrices in the following manner:

- For each M_i , compute the logarithm $\ln(M_i)$
- Interpolate/Approximate the logarithms:



Rotation Interpolation/Approximation

Given a collection of rotations $\{M_0, \dots, M_{n-1}\}$ we can generate a curve passing through/near the matrices in the following manner:

- For each M_i , compute the logarithm $\ln(M_i)$
- Interpolate/Approximate the logarithms:

» **Linear Interpolation:**

$$LN_i(t) = (1-t)\ln(M_i) + t\ln(M_{i+1})$$



Rotation Interpolation/Approximation

Given a collection of rotations $\{M_0, \dots, M_{n-1}\}$ we can generate a curve passing through/near the matrices in the following manner:

- For each M_i , compute the logarithm $\ln(M_i)$
- Interpolate/Approximate the logarithms:
 - » Linear Interpolation:
 - » Catmull-Rom Interpolation:

$$LN_i(t) = CR_0(t) \ln(M_{k-1}) + CR_1(t) \ln(M_k) + CR_2(t) \ln(M_{k+1}) + CR_3(t) \ln(M_{k+2})$$



Rotation Interpolation/Approximation

Given a collection of rotations $\{M_0, \dots, M_{n-1}\}$ we can generate a curve passing through/near the matrices in the following manner:

- For each M_i , compute the logarithm $\ln(M_i)$
- Interpolate/Approximate the logarithms:
 - » Linear Interpolation:
 - » Catmull-Rom Interpolation:
 - » Uniform Cubic B-Spline Approximation:

$$LN_i(t) = B_{0,3}(t) \ln(M_{k-1}) + B_{1,3}(t) \ln(M_k) + B_{2,3}(t) \ln(M_{k+1}) + B_{3,3}(t) \ln(M_{k+2})$$



Rotation Interpolation/Approximation

Given a collection of rotations $\{M_0, \dots, M_{n-1}\}$ we can generate a curve passing through/near the matrices in the following manner:

- For each M_i , compute the logarithm $\ln(M_i)$
- 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:

$$\Phi_i(t) = \exp(LN_i(t))$$



Rotation Interpolation/Approximation

Given a collection of rotations $\{M_0, \dots, M_{n-1}\}$ we can generate a curve passing through/near the matrices in the following manner:

- For each M_i , compute the logarithm $\ln(M_i)$

Note that since the logarithm of rotations is a skew-symmetric matrix, and since skew-symmetric matrices are closed under addition and scaling, the weighted average $LN_i(t)$ is also skew-symmetric, so its exponent has to be a rotation.

- Set the value of the in-between rotation to be the exponent of the blended logarithms:

$$\Phi_i(t) = \exp(LN_i(t))$$



Summary

In order to define in-between frames for an animation, we need to interpolate/approximate the transformations specified in the key-frames.

- For translation, we can just 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