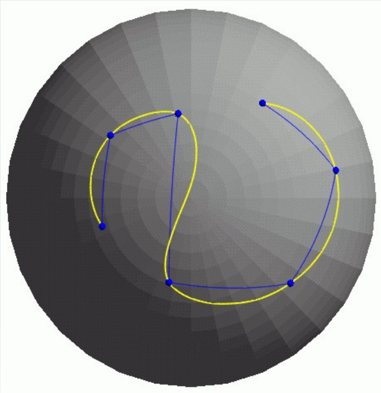
SLERP

Marc Neveu

SLERP

- Raccourci pour ”spherical linear interpolation” (Ken Shoemake)
- Contexte : interpolation de quaternions pour l'animation de rotations 3D
- Idée : mouvement à vitesse constante sur un arc de cercle unitaire, avec en données les extrémités du chemin et un paramètre d'interpolation sur [0,1].



POURQUOI?

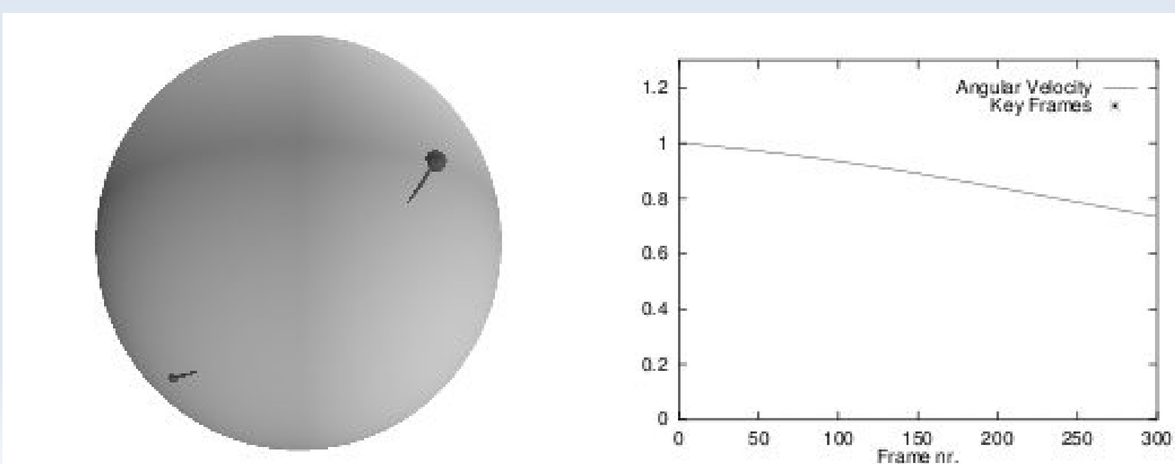
- Interpolation entre 2 rotations

Soit un ensemble M : interpolation entre $x_0 \in M$ et $x_1 \in M$ paramétrée par $t \in [0,1]$.
La courbe d'interpolation γ de MxMx[0,1] -> M doit vérifier:

(x0,x1;0) = x0
(x0,x1;1) = x1

- LinEuler

Le + simple : interpolation linéaire entre 2 tuples d'angles d'Euler
$x_0 = (\theta_0\ \phi_0\ \psi_0) \in R^3$ et $x_1 = (\theta_1\ \phi_1\ \psi_1) \in R^3$: $\text{LinEuler}(x_0,x_1; h) = x_0\,(1-h) + x_1\,h$



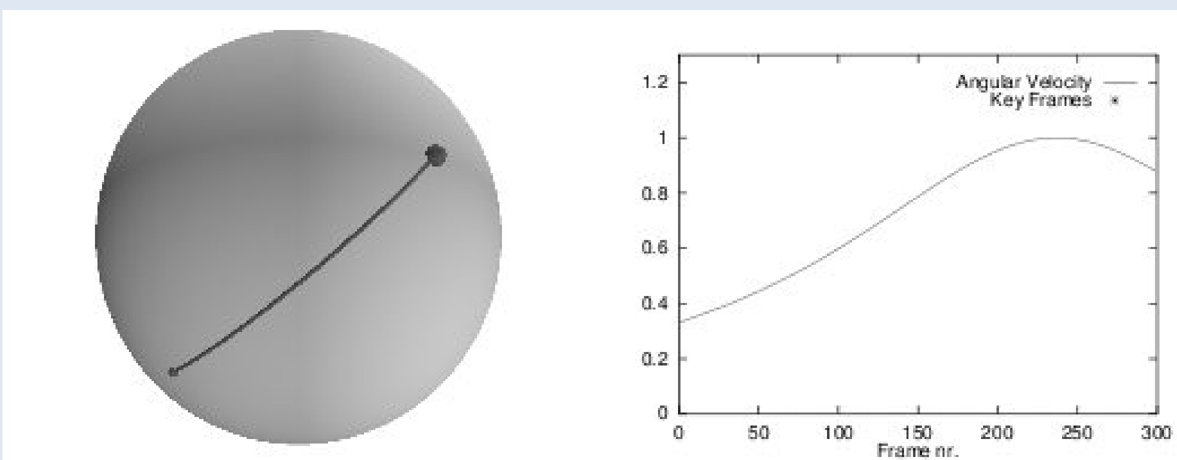
$$V(q_i) = \frac{\|q_i - q_{i-1}\|}{2} + \frac{\|q_i - q_{i+1}\|}{2}$$

Autre solution

- Interpolation Linéaire de Matrices : LinMat

interpolation linéaire entre matrices de rotation cad interpolation linéaire de chaque élément de la matrice indépendamment des autres.
avec $t \in [0\ 1]$, courbe d'interpolation entre $M_0 \in R^4xR^4$ and $M_1 \in R^4xR^4$:

$$LinMat(M_0,M_1;t) = M_0(1-t) + M_1\, t$$



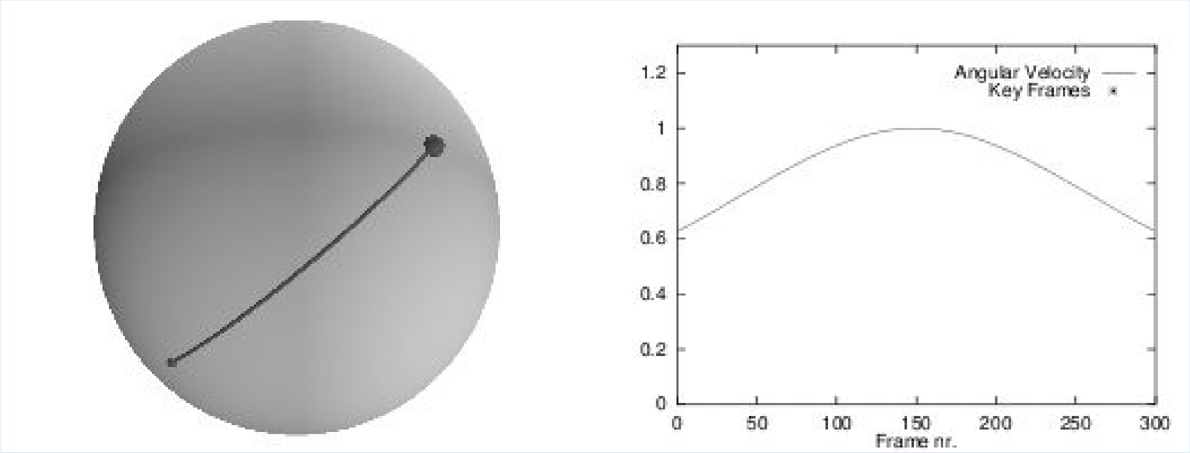
Courbe d'interpolation projetée sur la sphère unité des quaternions (partie rotation pure de l'interpolation).

Autre solution

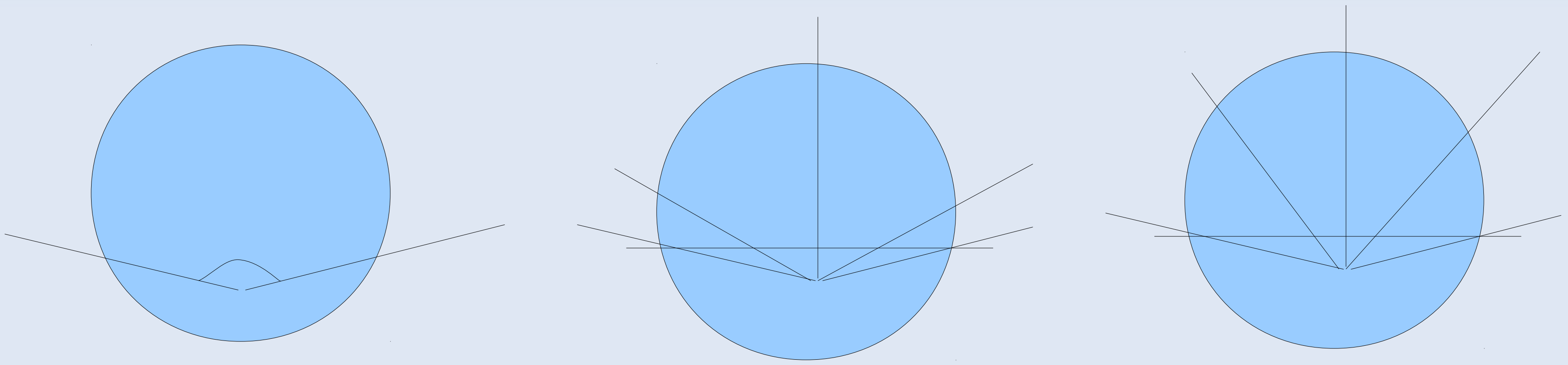
- Linear Quaternion interpolation: Lerp

Pour q_0 et $q_1 \in H$ et $t \in [0\ 1]$:

$\mathrm{Lerp}(q_0\ q_1\ t) = q_0\,(1-t) + q_1 t$ (une droite dans H)



- LERP/SLERP

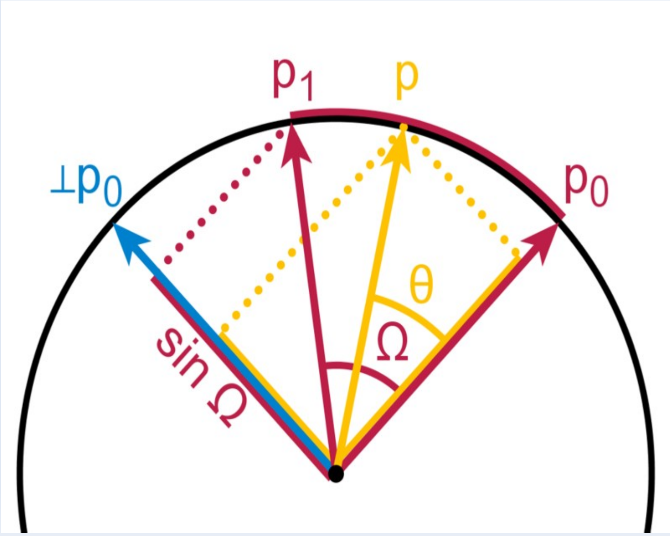


Lerp : la sécante coupée en 4 segments égaux
Slerp : l'angle est cooupé en 4 angles égaux

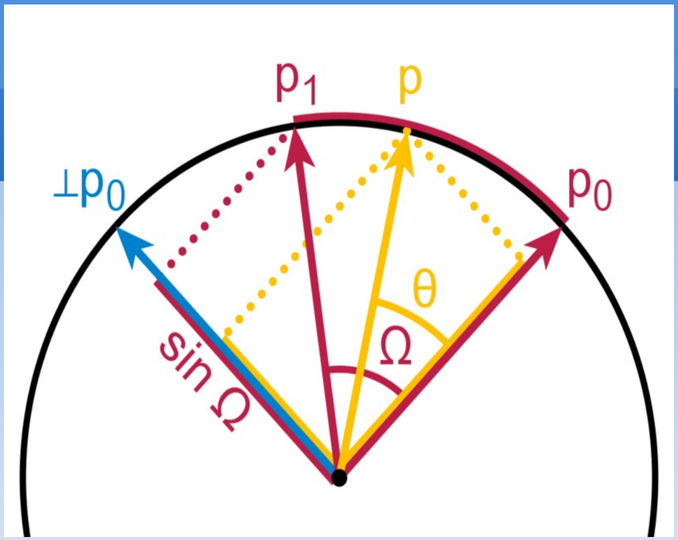
SLERP "en général"

- formule indépendante des quaternions pour un arc de courbe dans R^n
- tout point de la courbe est une combinaison linéaire des extrémités
- Soient p_0 et p_1 les extrémités de l'arc, $t \in [0,1]$

$$slerp(p_0, p_1, t) = \frac{\sin[(1-t)\Omega]}{\sin(\Omega)} p_0 + \frac{\sin[t\,\Omega]}{\sin(\Omega)} p_1$$



SLERP "en général"



$$slerp(p_0, p_1, t) = \frac{\sin[(1-t)\Omega]}{\sin(\Omega)} p_0 + \frac{\sin[t\,\Omega]}{\sin(\Omega)} p_1$$

$$\cos\Omega = p_0 \cdot p_1$$

- Pptés :
  - Slerp(p_0,p_1;t) = Slerp(p_1,p_0;1−t)
  - $\Omega \rightarrow 0$, interpolation linéaire (sin x ~ x)
  - Si $p_0 \perp p_1$: on retrouve p = p_0 cos θ + p_1 sin θ (avec θ = t π/2, et cos θ = sin (π/2−θ))
  - 1/sin Ω = normalisation, car p_1 se projette sur $\perp p_0$ avec une longueur de sin Ω

SLERP pour quaternions

- Soit q un quaternion unitaire : $\cos\Omega + \mathbf{v} \sin\Omega$
- Appliqué aux quaternions unitaires, le Slerp interpole des rotations en 3D avec un effet de rotation à vitesse angulaire uniforme autour d'un axe de rotation fixe
- Le Slerp donne le plus court chemin entre les extrémités du quaternion avec une rotation d'angle 2Ω

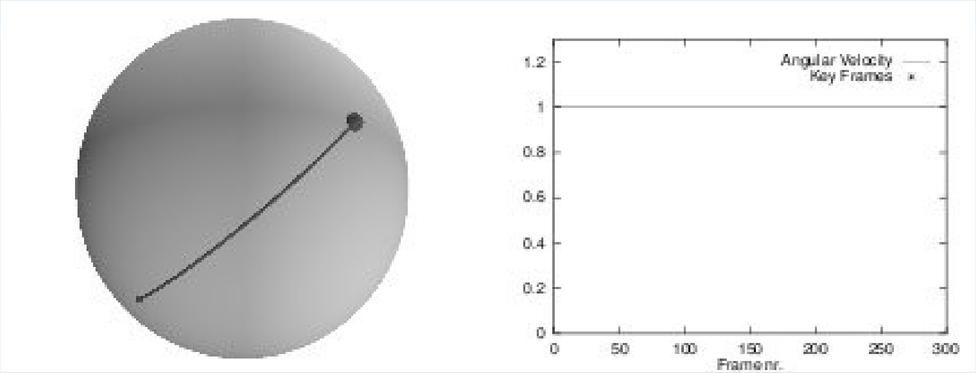
SLERP pour quaternions

- q unitaire : $q = \cos\Omega + \mathbf{v} \sin\Omega = e^{v\Omega}$ ($\mathbf{v}^2=-1$)
- $q^{\,t} = e^{vt\Omega} = \cos t\Omega + \mathbf{v} \sin t\Omega$
- $q = q_1 q_0^{-1}$, partie réelle de q : cos Ω (équivalent de $p_0.p_1$),
- expressions équivalentes du Slerp appliqué à des quaternions

$$\begin{aligned}
Slerp(q_{0,} q_1 ; t) &= q_0 (q_0^{-1} q_1)^t \\
&= q_1 (q_1^{-1} q_0)^{1-t} \\
&= (q_0 q_1^{-1})^{1-t} q_1 \\
&= (q_1 q_0^{-1})^t q_0
\end{aligned}$$

Slerp

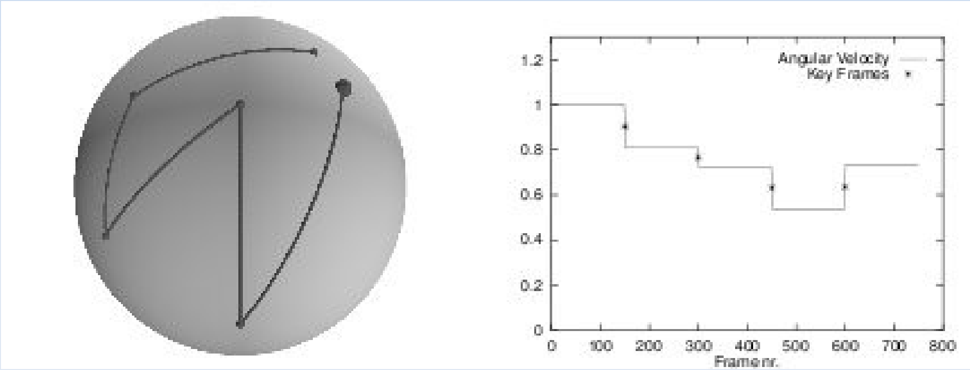
- Résultat



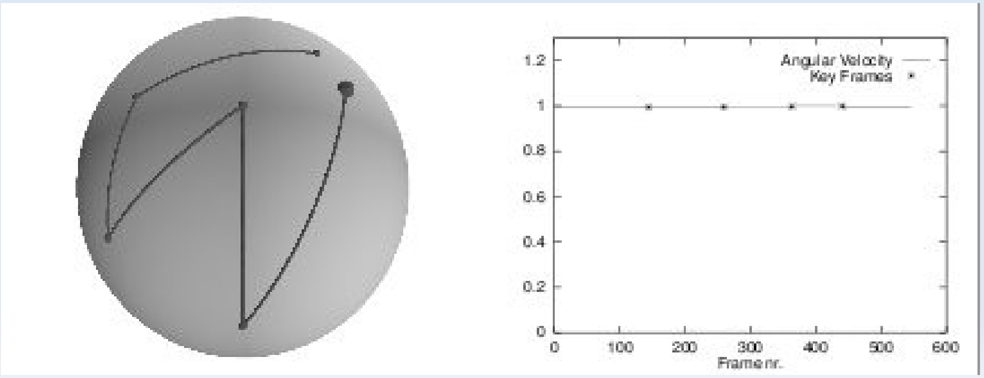
La courbe d'interpolation Slerp forme un arc sur la sphère unité des quaternions
En termes différentiels cet arc est une géodésique(= ligne droite). C'est l'arc le plus court entre les 2 quaternions. Slerp assure une vélocité constante.

Série d'interpolations

- Et si on a plusieurs rotations?

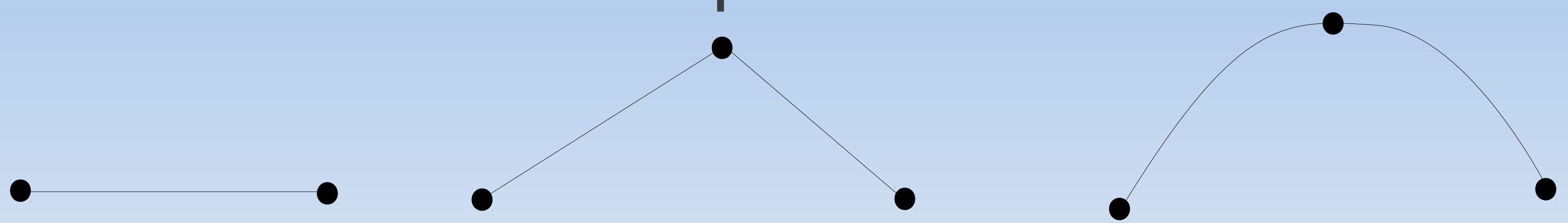


Reparamétrisation : assigner à chaque intervalle un nombre de trames dépendant de sa taille
taille = angle entre une paire de clés qi et qi+1
angle : donné par cos = qi . qi+1 (arrondi parce que nombre = entier)

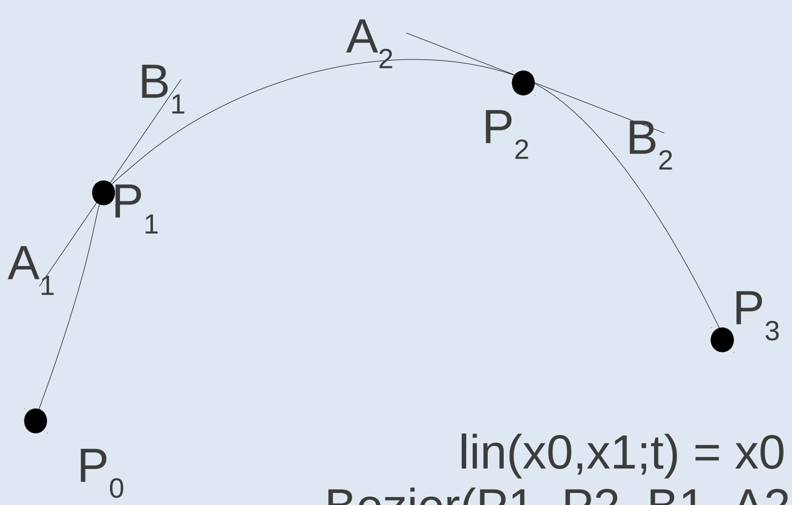


Série d'interpolations

- Faire comme avec splines



Dans le plan simple interpolation entre 2 points= ligne droite
Interpolation linéaire entre une serie de points → pas différentiable
Différentiabilité avec courbes cubiques (par exemple splines).



lin(x0,x1;t) = x0 (1- t) + x1 t
Bezier(P1, P2, B1, A2;t) = lin(lin(P1, P2; t),lin(B1, A2;t); 2t(1-t))

Squad

- Squad : Spherical & Quadrangle

$$\text{Squad}(\underbrace{q_i, q_{i+1}}_{P_1P_2}, \underbrace{s_i, s_{i+1}}_{B_1A_2}; t) = \text{Slerp}(\text{Slerp}(q_i, q_{i+1}; t), \text{Slerp}(s_i, s_{i+1}; t); 2t(1-t))$$

$$s_i = q_i \exp\left[- \frac{\log(q_i^{-1} q_{i+1}) + \log(q_i^{-1} q_{i-1})}{4} \right]$$