Displays

List of display attributes:

- resolution : SD, HD (2k), Ultra HD (4k), Ultra HD (8k)
- physical size: width x height x depth : larger, thinner
- frame rate: 30 → 120 Hz
- brightness & contrast: high dynamic range (HDR)
- stereo / 3D images
- colour gamut : >3 primaries
- colour "depth": 8 → 12 bit colour

Frame rates

film : 24 Hz, but each frame is shown twice to prevent visible flicker

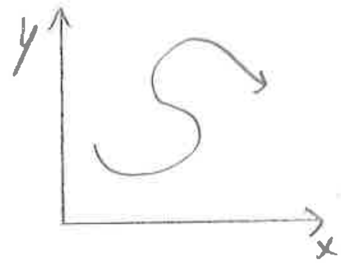
broadcast television: 30 Hz in North America (NTSC, really 29.97 Hz)

typical video game: 60+ Hz

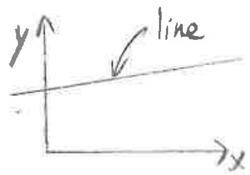
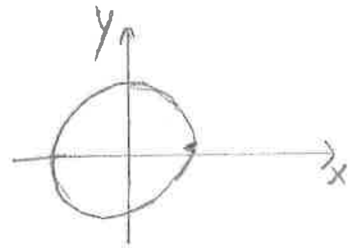
Youtube : same as what you upload.

future: adaptive framerates?

Animating a Point



Curves

			in 3D
① Explicit: $y = f(x)$	$y = mx + b$ $y = ax^2 + bx + c$ etc.	$y = \pm\sqrt{r^2 - x^2}$	$y = f(x, z)$ height field surface
② Implicit: $f(x,y) = 0$	$0 = y - mx - b$ $0 = y - ax^2 - bx - c$ etc.	$0 = r^2 - x^2 - y^2$	$f(x,y,z) = 0$ surface
③ Parametric: $x = f_x(t)$ $y = f_y(t)$	$x = a_1 t + a_0$ $y = b_1 t + b_0$	$x = r\cos t$ $y = r\sin t$ $x = a_2 t^2 + a_1 t + a_0$ $y = b_2 t^2 + b_1 t + b_0$	$\left.\begin{array}{l} x = f_x(t) \\ y = f_y(t) \\ z = f_z(t) \end{array}\right\\}$ curve $\left.\begin{array}{l} x = f_x(s,t) \\ y = f_y(s,t) \\ z = f_z(s,t) \end{array}\right\\}$ surface

Geometric modeling: most often use parametric equations to describe motion over time

Animation: parametric: t represents time

"AVARS" or "DOFs"

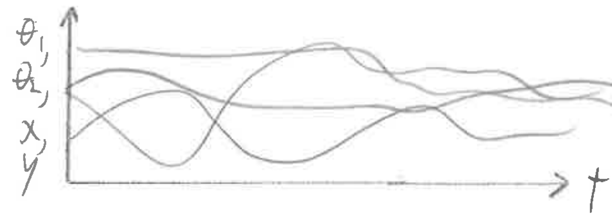
degrees of freedom

animation variables

$\theta_1 = f_1(t)$

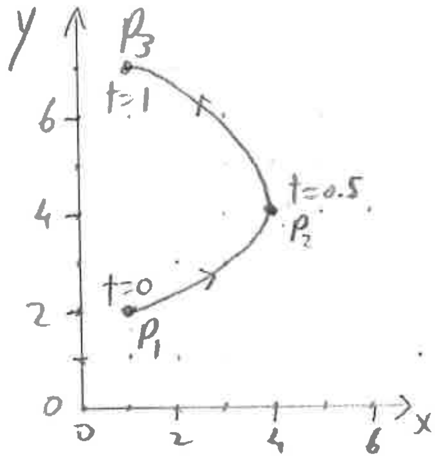
$\theta_2 = f_2(t)$

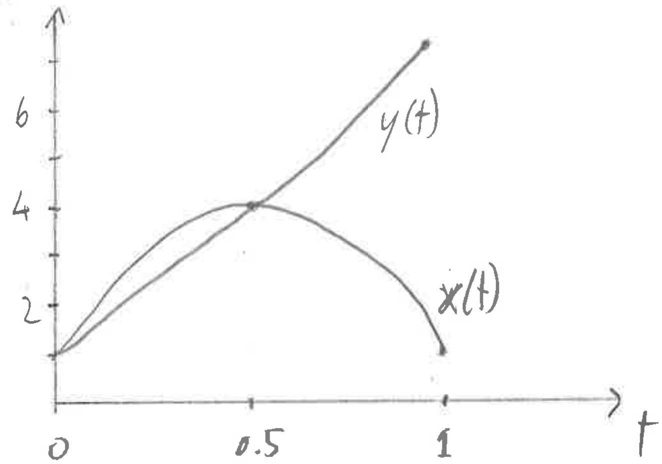
...



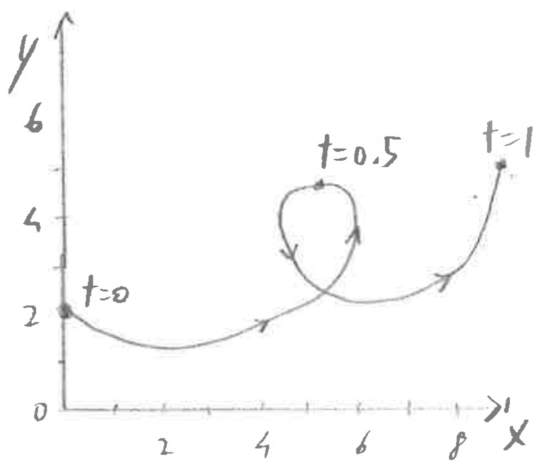
Parametric Curve Examples

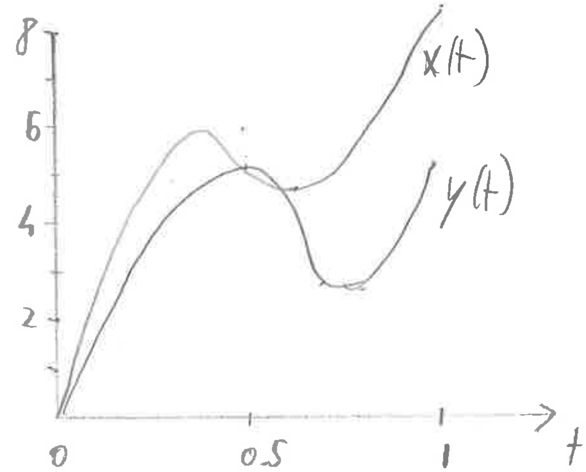
(1)





(2)



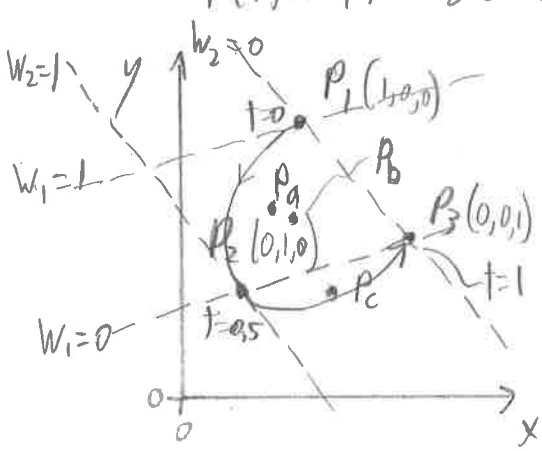


Two possible ways of thinking about curves:

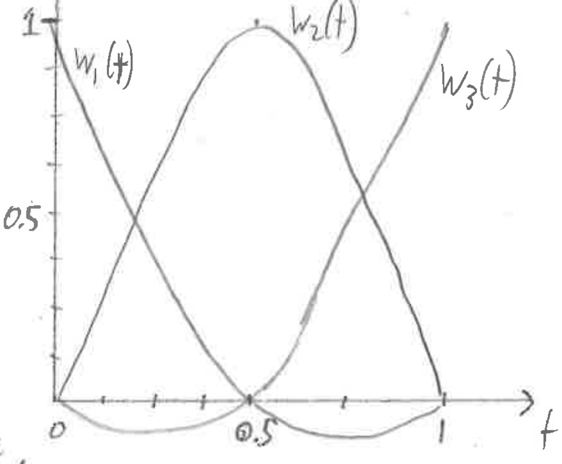
(1) Defined by <u>polynomial coefficients</u> e.g. $x(t) = \sum_{i=0}^{N-1} a_i t^i$

(2) Defined by <u>weighted combinations of points</u> e.g. $x(t) = \sum_{i=1}^{N} w_i(t)\, x_i$

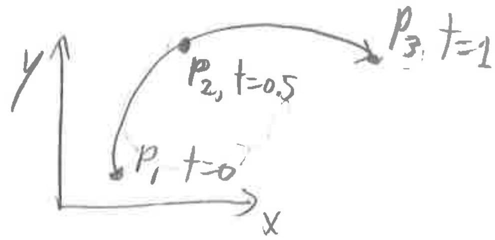
$P(t) = w_1 P_1 + w_2 P_2 + w_3 P_3 \quad \sum w_i = 1$ "barycentric coordinates"



$P(w_1, w_2, w_3)$
$P_a(0.5, 0.5, 0)$
$P_b(0.4, 0.4, 0.2)$
$P_c(-0.2, 0.6, 0.6)$
$P_d(1, 1, 0)$ — not a barycentric coordinate



Three Point Interpolating Curve



Consider example ①
Given three points on the curve $P(t)$:

Functions to model $x(t)$ and $y(t)$:

$$x(t) = a_2 t^2 + a_1 t + a_0$$
$$= [t^2 \;\; t \;\; 1] \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix}$$
$$= T \cdot A$$

Constraints:

$$X_1 = x(0) = a_2(0)^2 + a_1(0) + a_0$$
$$X_2 = x(0.5) = a_2(0.5)^2 + a_1(0.5) + a_0$$
$$X_3 = x(1) = a_2(1)^2 + a_1(1) + a_0$$

Rewrite in matrix form:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0.5^2 & 0.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix}$$
$$G_X = B \cdot A$$

Solve for A:

$A = B^{-1} G_X$
define $M = B^{-1}$.
$A = M G_X$

$$M = \begin{bmatrix} 0 & 0 & 1 \\ 0.5^2 & 0.5 & 1 \\ 1 & 1 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

Final form:

$$x(t) = T \cdot A$$
$$= T \cdot M \cdot G_X$$
$$= [t^2 \;\; t \;\; 1] \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

$$x(t) = \underbrace{T M}_{[b_1(t) \;\; b_2(t) \;\; b_3(t)]} G_X$$ "basis functions"

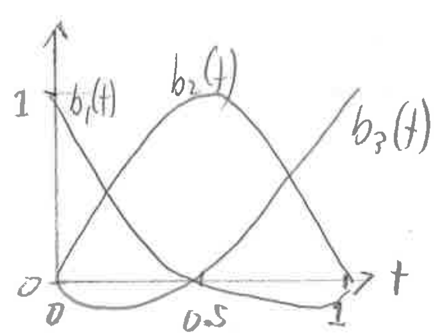
A

Basis Functions

$$T \cdot M = [\underbrace{2t^2 - 3t + 1}_{b_1(t)} \;\; \underbrace{-4t^2 + 4t}_{b_2(t)} \;\; \underbrace{2t^2 - t}_{b_3(t)}]$$

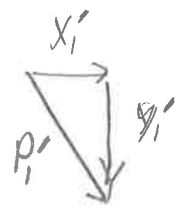
check:
$$\sum_i b_i(t) = 1$$

Sketch:

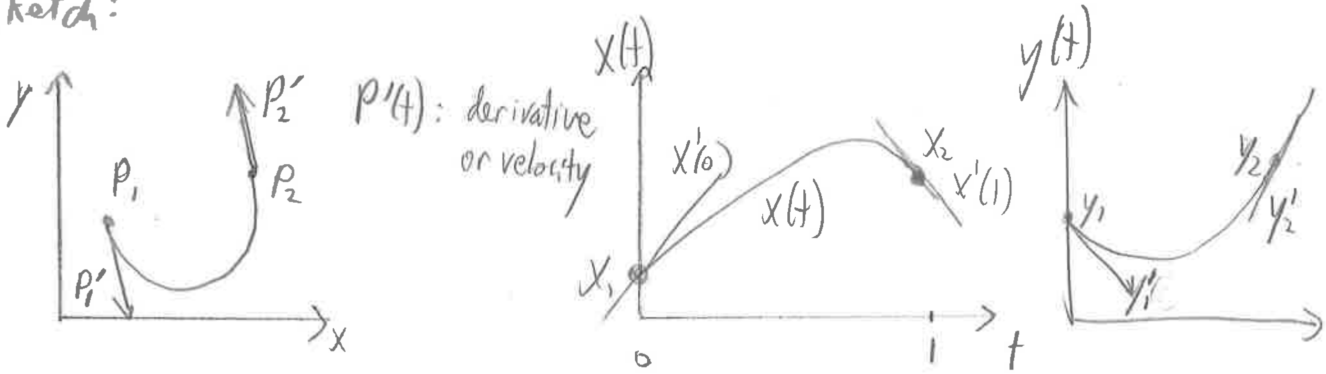


Hermite Curves

Defined by:



Sketch:



Function to model $x(t)$: $X(t) = a_3t^3 + a_2t^2 + a_1t + a_0$ $\quad X'(t) = [3t^2 \ \ 2t \ \ 1 \ \ 0] \cdot A$

$$= [t^3 \ \ t^2 \ \ t \ \ 1] \cdot \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

$$= T \cdot A$$

Constraints:

$$X_1 = X(0) = [0 \ \ 0 \ \ 0 \ \ 1] \cdot A$$
$$X_2 = X(1) = [1 \ \ 1 \ \ 1 \ \ 1] \cdot A$$
$$X_1' = X'(0) = [0 \ \ 0 \ \ 1 \ \ 0] \cdot A$$
$$X_2' = X'(1) = [3 \ \ 2 \ \ 1 \ \ 0] \cdot A$$

Matrix form

$$\begin{bmatrix} X_1 \\ X_2 \\ X_1' \\ X_2' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \qquad G_x = B \cdot A$$

Solve for A:

$$A = B^{-1} G_x$$

Final form: $X(t) = T \, M \, G_x$ ($M \to B^{-1}$)

$$\begin{array}{cccc} & T & M & G_x \\ x(t) = & [t^3 \ \ t^2 \ \ t \ \ 1] & \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} X_1 \\ X_2 \\ X_1' \\ X_2' \end{bmatrix} \end{array}$$

Basis Functions

$b_1(t) = 2t^3 - 3t^2 + 1$

$b_2(t) = -2t^3 + 3t^2$

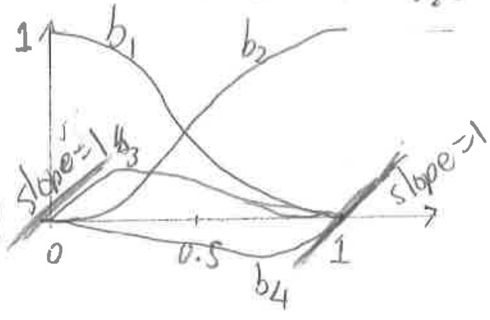
$b_3(t) = t^3 - 2t^2 + t$

$b_4(t) = t^3 - t^2$

Check: $b_1(t) + b_2(t) = 1$

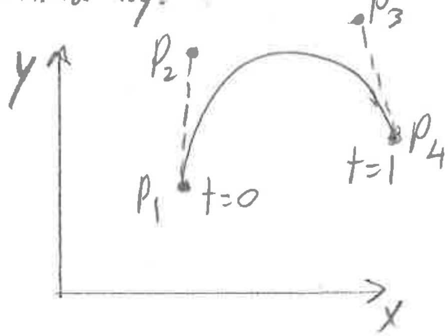
(weights for the points)

Sketch:



Bézier Curves

Defined by:



$$G_x = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

- satisfies the convex hull property i.e., $P(t)$ $t \in [0,1]$ stays within the convex hull of $\{P_1, P_2, P_3, P_4\}$

Relationship to Hermite Curves:

$$G_H = M_{Bez \to H} \, G_{Bez}$$

define $X'(0) = 3(X_2 - X_1)$

$X'(1) = 3(X_4 - X_3)$

$$\Rightarrow \begin{bmatrix} X_1 \\ X_2 \\ X_1' \\ X_2' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

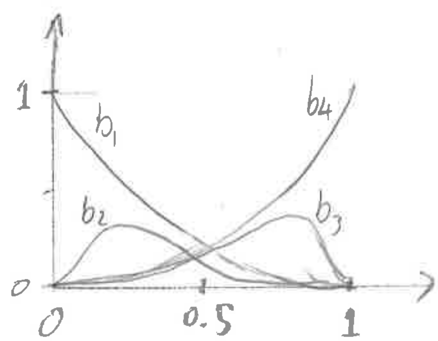
$$x(t) = T M_H G_H$$

$$= T \underbrace{M_H M_{Bez \to H}}_{M_{Bez}:\ \text{Bézier basis matrix}} G_{Bez}$$

$$= T M_{Bez} G_{Bez}$$

$$M_{Bez} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Basis Functions



$T \cdot M_{Bez} = [b_1(t) \;\; b_2(t) \;\; b_3(t) \;\; b_4(t)]$

$$\underbrace{\begin{aligned} b_1(t) &= -t^3 + 3t^2 - 3t + 1 \\ b_2(t) &= 3t^3 - 6t^2 + 3t \\ b_3(t) &= -3t^3 + 3t^2 \\ b_4(t) &= t^3 \end{aligned}}$$

Ie: $\sum_i b_i(t) = 1$

Based on Bernstein Polynomials

Basis functions are the terms obtained by expanding:

$$[(1-t) + t]^n$$

$[(1-t)+t]^1 = \underline{(1-t)} + \underline{t}$: first order Bézier basis functions

$[(1-t)+t]^2 = \underline{(1-t)^2} + \underline{2t(1-t)} + \underline{t^2}$: second order

$[(1-t)+t]^3 = \underline{(1-t)^3} + \underline{3t^2(1-t)} + \underline{3t(1-t)^2} + \underline{t^3}$: third order, i.e.,

$(a+b)^0 = 1$

$(a+b)^1 = a + b$

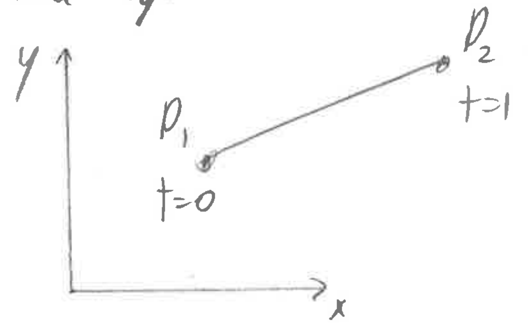
$(a+b)^2 = a^2 + 2ab + b^2$

$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$

...

Two Point Interpolating Curve

Defined by:



Function to model $x(t)$:

$$x(t) = a_1 t + a_0$$
$$= \begin{bmatrix} t & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = T \cdot A$$

Constraints:

$$x_1 = x(0) = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot A$$
$$x_2 = x(1) = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot A$$

Matrix form:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_0 \end{bmatrix}$$

$[\ ]^{-1}$

$$G_x = B \cdot A$$

Solve for A:

$$A = B^{-1} G_x$$

Final form:

$$x(t) = T \cdot A$$
$$= T \cdot B^{-1} G_x$$
$$= T \cdot M \cdot G_x$$

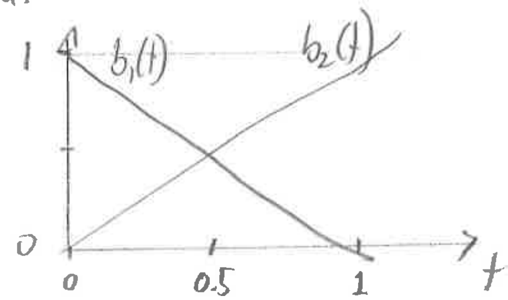
$$x(t) = \begin{array}{ccc} T & M & G \\ \begin{bmatrix} t & 1 \end{bmatrix} & \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{array}$$

Basis functions:

$$T \cdot M = \begin{bmatrix} b_1(t) & b_2(t) \end{bmatrix}$$

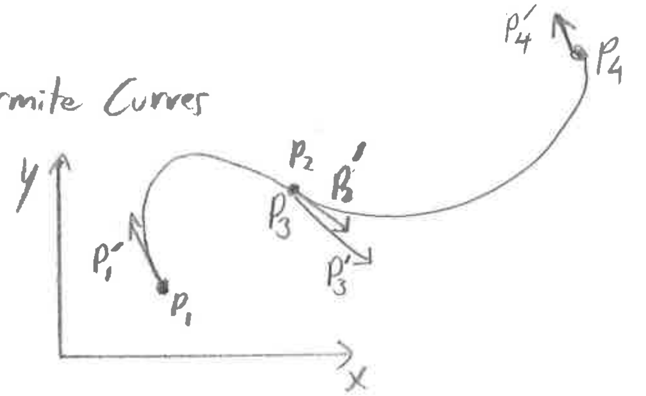
$$b_1(t) = 1 - t$$
$$b_2(t) = t$$

Sketch:



Piecewise Hermite and Bézier Curves

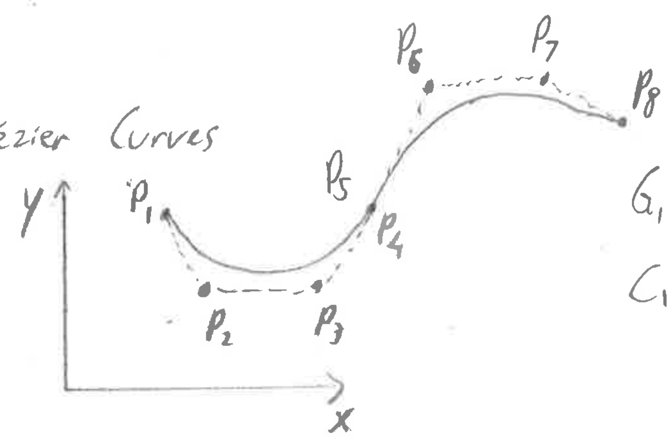
Hermite Curves



$G_1: \quad P_3 = P_2, \; P_3' = kP_2'$

$C_1: \quad P_3 = P_2, \; P_3' = P_2'$

Bézier Curves



$G_1: \quad P_6 - P_5 = k(P_4 - P_3)$

$C_1: \quad P_6 - P_5 = P_4 - P_3$

Geometric Continuity

G_0 curves are joined
G_1 first derivatives are proportional; no 'kink' or edge at join point
G_2 first and second derivatives are proportional; continuous curvature

Parametric Continuity

C_0 curves are joined
C_1 first derivatives equal
C_2 first and second derivatives equal

C_n implies G_n, but not vice-versa

possible traffic circle:

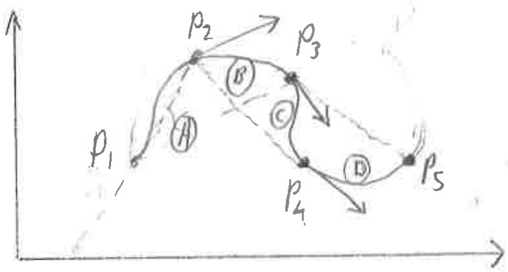
join point is <u>not G_2</u> because of the discontinuity in the curvature, i.e., the steering angle.

Catmull - Rom Curves

(powerpoint curve demo)

Defined by:

A: $\begin{bmatrix} P_1 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$, B: $\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$, C: $\begin{bmatrix} P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix}$, D: $\begin{bmatrix} P_3 \\ P_4 \\ P_5 \\ P_5 \end{bmatrix}$



- C_1 continuity
- interpolates points
- end segments: (a) repeat end point, e.g.: $\begin{bmatrix} X_1 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix}$

  (b) produce a 'reflected' virtual control point: $\begin{bmatrix} \hat{X}_1 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix}$, $\hat{X}_1 = X_1 - (X_2 - X_1)$

  (c) set $P'(0) = 0$

Construction: $P'_k = \frac{1}{2}(P_{k+1} - P_{k-1})$

$$\begin{bmatrix} X_1 \\ X_2 \\ X_1' \\ X_2' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

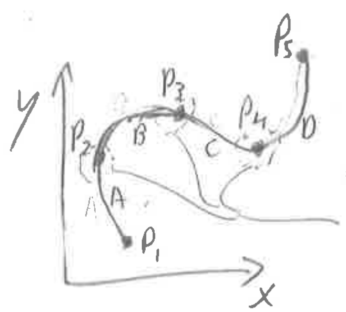
$$G_H = M_{CR\to H}\, G_{CR} \qquad x(t) = T \cdot \underbrace{M_H \cdot M_{CR\to H}}_{M_{CR}}\, G_{CR}$$

$$M_{CR} = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

Interpolating Spline

Piecewise Cubic

C_2



5 points

4 curve segments

3 join points

Issue: control is no longer local, i.e., moving any control point will cause the entire curve to change shape.

In general:

n control points

n-1 curve segments

4(n-1) parameters: n-1 curves x 4 polynomial Coeffs per curve

2(n-1) position constraints, e.g., $X_A(0) = X_1$, $X_A(1) = X_2$, etc.

n-2 first deriv. constraints, e.g., $X_A'(1) = X_B'(0)$, etc.

n-2 second deriv. constraints, e.g., $X_A''(1) = X_A''(0)$, etc.

$\boxed{\begin{matrix} 4n-6 \text{ constraints} \\ 4n-4 \text{ parameters} \end{matrix}}$ → under constrained

Choose 2 additional constraints, e.g., $X_A'(0) = 0$, $X_D''(1) = 0$

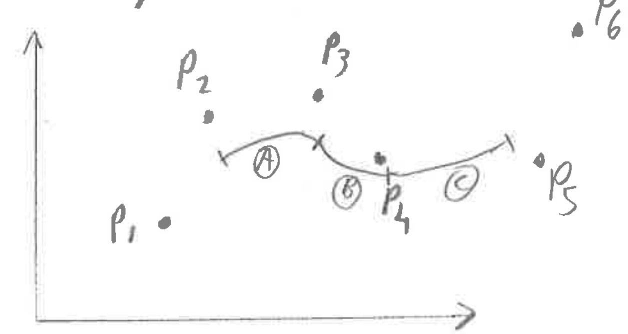
Then solve simultaneous linear system of equations.

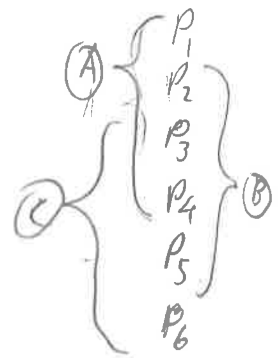
"Basis"

B-spline Curves

C_2

Defined by:



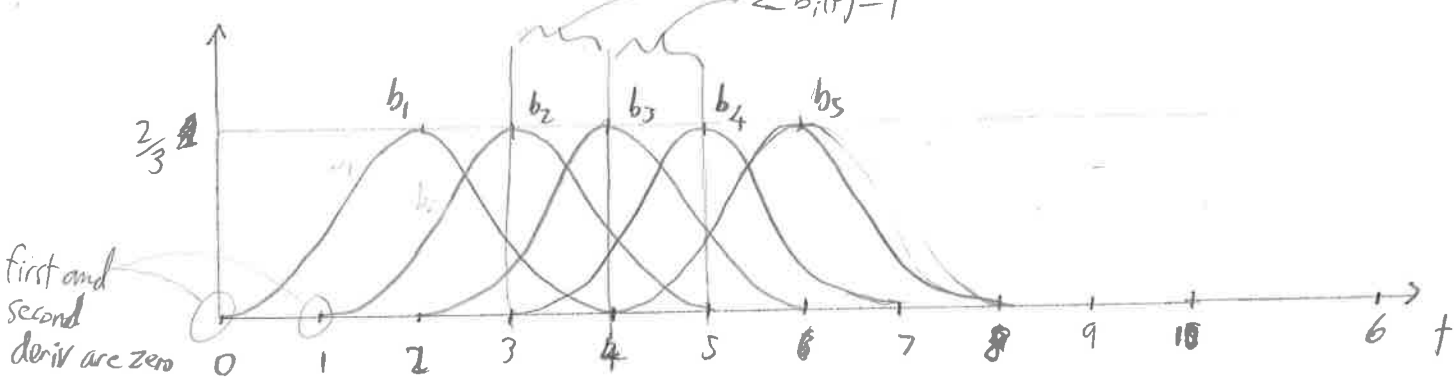


Constraints:
- continuous first derivatives at join, e.g., $x'_A(1) = x'_B(0)$
- continuous second derivatives at join, e.g., $x''_A(1) = x''_B(0)$

Final form: (derivation beyond scope of course, but not that difficult)

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Basis functions



B-spline continuity: C_2

B-spline endpoint interpolation
→ include duplicate knots, e.g.,

$P_1\ P_1\ P_1\ P_2\ P_3\ P_4\ P_5\ P_6\ P_6\ P_6$

Spline Comparison

	local control	interpolating	C_2
Catmull-Rom	✓	✓	
cubic B-spline	✓		✓
interpolating piecewise cubic C_2 spline		✓	✓

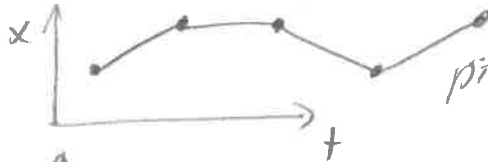
Common keyframe Types

in commercial tools, e.g. Maya

- "spline" smoothly passes through desired value, from previous keyframe to next keyframe

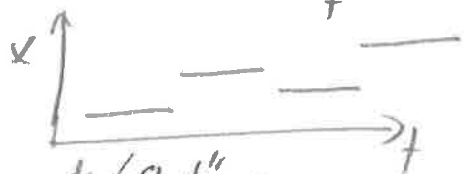
i.e, Catmull-Rom :

- "linear"



piecewise linear

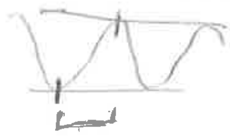
- "stepped"



piecewise constant

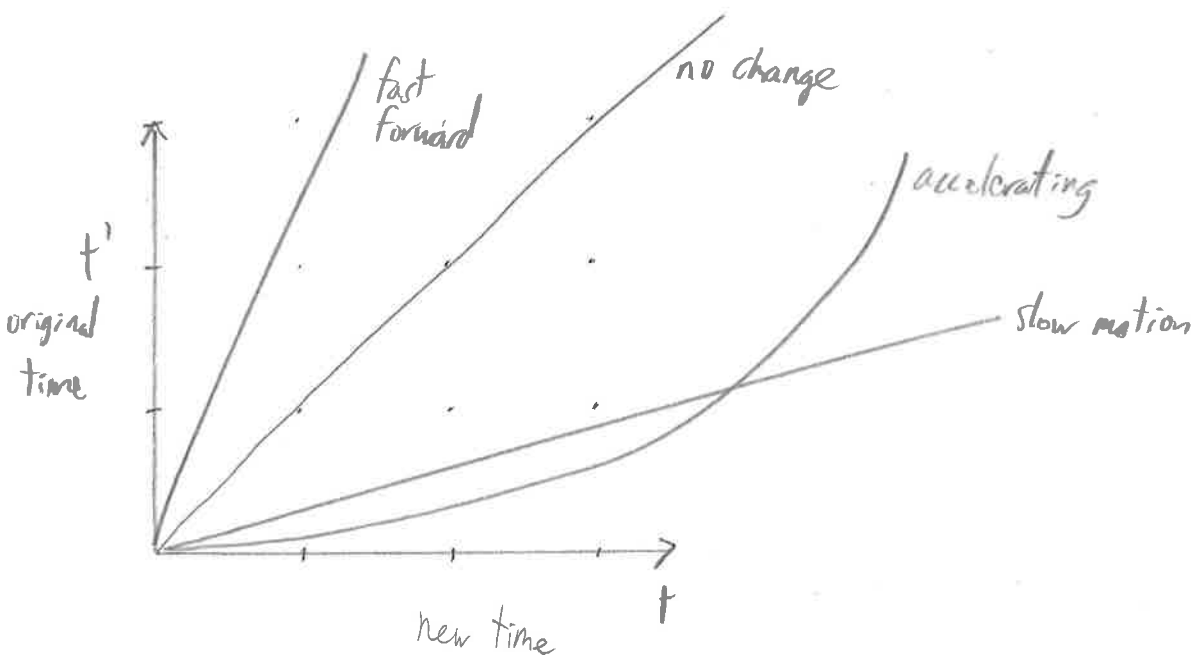
- "ease-in ease-out / flat"

use cubic spline, or portion of sinusoid



- "in tangent + out tangent"

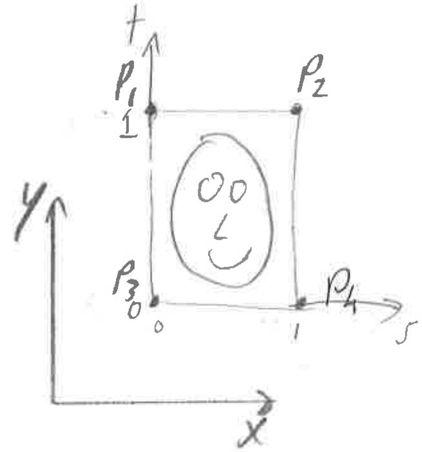
Motion Retiming

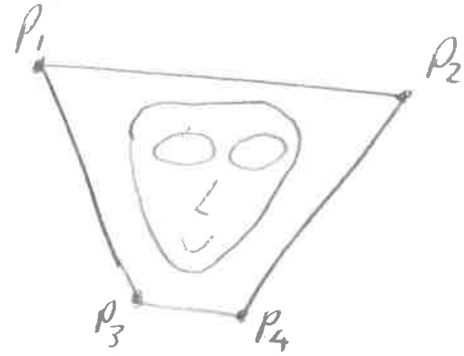


Splines for Deformation

Deform an embedding space

2D example with first-order Bézier basis functions.



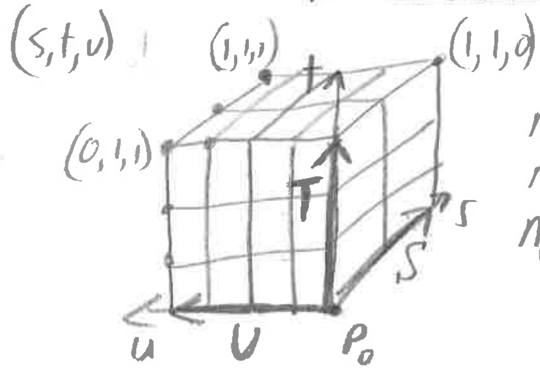


$$P = \sum_i w_i(s,t)\, P_i$$

$$= \underbrace{(1-s)t}\, P_1 + \underbrace{st}\, P_2 + \underbrace{(1-s)(1-t)}\, P_3 + \underbrace{s(1-t)}\, P_4$$

basis functions

Free Form Deformations



$n_s = 2$
$n_t = 3$
$n_u = 4$

① Define embedding grid using P_0, S, T, U

② Compute s, t, u for embedded object vertices

$$s = (P - P_0) \cdot S \cdot \frac{1}{\|S\|^2} \quad \text{ensures } s \in [0,1]$$

$$t = (P - P_0) \cdot T \cdot \frac{1}{\|T\|^2}$$

$$u = (P - P_0) \cdot U \cdot \frac{1}{\|U\|^2}$$

$(n_s + 1)(n_t + 1)(n_u + 1) = 60$ control points

③ Define initial control point locations

$$P_{ijk} = P_0 + \frac{i}{n_s} S + \frac{j}{n_t} T + \frac{k}{n_u} U$$

for $0 \le i \le n_s$
$0 \le j \le n_t$
$0 \le k \le n_u$

④ Move control points P_{ijk}

⑤ $P(s,t,u) =$

$$\sum_{i=0}^{n_s} \sum_{j=0}^{n_t} \sum_{k=0}^{n_u} \underbrace{\binom{n_s}{i}\binom{n_t}{j}\binom{n_u}{k}(1-s)^{n_s-i} s^i (1-t)^{n_t-j} t^j (1-u)^{n_u-k} u^k}_{\text{product of the Bézier basis functions in each dimension}} \, P_{ijk}$$

remember: $\left[(1-t)+t\right]^n = \sum_{i=0}^{n} \binom{n}{i}(1-t)^{n-i} t^i$ $\quad (a+b)^3 = \binom{3}{0}a^3 + \binom{3}{1}a^2b + \binom{3}{2}ab^2 + \binom{3}{3}b^3$

$$= a^3 + 3a^2b + 3ab^2 + b^3$$

Deformation Methods

"Rigging" : Preparing a 3D model for animation by adding shape deformation mechanisms. These allow an animator to control the shape and motion via a set of "handles"
i.e., control points, joint angles, "reach targets" for the hands & feet, etc.
Woody in "Toy Story": 700+ (200+ just for face)

Space Deformation Methods [handles] (also useful for image morphing)
- free form deformations : lattice embeds object [lattice points]
- "wires" : virtual armature; deforms space around a curve [curve]
- cages : control points of embedding cage; allows for flexible topology, unlike lattices
e.g. "harmonic coordinates" [cage vertices]

Optimization-based Methods
- Laplacian surface editing [anchor & manipulation vertices]
- "As rigid as possible" [same]
- several others

Physics-based Methods
- i.e., guest lecture

Skeleton-driven Methods "skinning" (to be discussed in detail later)
- drive character motion via joint angles
+ position and orientation of "root link"
- multiple possible skinning methods to produce shape deformations that result from bending and twisting at joints.

Data-driven methods
- learn deformation models from real data