Computer Graphics

- Splines -

Arsène Pérard-Gayot

(Slides by Piotr Danilewski)

CURVES

Curves

Explicit	$y = f(x)$	$f: \mathbb{R} \to \mathbb{R}$	$\gamma = \{(x, f(x))\}$	$y = \sqrt{1 - x^2}$
Implicit	$F(x, y) = 0$	$F: \mathbb{R}^2 \to \mathbb{R}$	$\gamma = \{(x, y): F(x, y) = 0\}$	$x^2 + y^2 = 0$
Parametric	$f_x(t) \quad f_y(t)$ $f(t)$	$f_x, f_y: \mathbb{R} \to \mathbb{R}$ $f: \mathbb{R} \to \mathbb{R}^2$ typically: $t \in [0,1]$	$\gamma = \left\{ \begin{array}{c} (x, y): \exists t \in \mathbb{R}: \\ f_x(t) = x \\ f_y(t) = y \end{array} \right\}$	$x(t) = \cos(t)$ $y(t) = \sin(t)$ $t \in [0, 2\pi]$

Polynomial curves

- Avoids complicated functions (e.g. pow, exp, sin, sqrt)
- Use simple polynomials of low degree
- Flexible, easy to use

degree

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots$$
$$y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + \cdots$$
$$z(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + \cdots$$

monomial

$$P(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \sum_{i=0}^{n} t^i \cdot (a_i, b_i, c_i)$$

monomial basis

$$P(t) = \begin{pmatrix} t^n & t^{n-1} & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} a_n & b_n & c_n \\ a_{n-1} & b_{n-1} & c_{n-1} \\ \vdots & \vdots & \vdots \\ a_0 & b_0 & c_0 \end{pmatrix}$$

Coefficients $\in \mathbb{R}^3$

- *n* coordinates
- each coordinate is $\mathbb{R}^3$

Derivatives

- Tangent vector

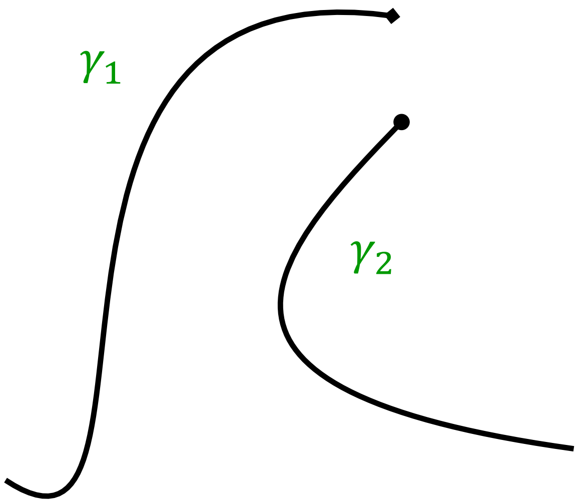
$$P(t) = \begin{pmatrix} t^n & t^{n-1} & \cdots & t & 1 \end{pmatrix} \cdot \begin{pmatrix} a_n & b_n & c_n \\ a_{n-1} & b_{n-1} & c_{n-1} \\ \vdots & \vdots & \vdots \\ a_0 & b_0 & c_0 \end{pmatrix}$$

$$P'(t) = \begin{pmatrix} nt^{n-1} & (n-1)t^{n-1} & \cdots & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_n & b_n & c_n \\ a_{n-1} & b_{n-1} & c_{n-1} \\ \vdots & \vdots & \vdots \\ a_0 & b_0 & c_0 \end{pmatrix}$$

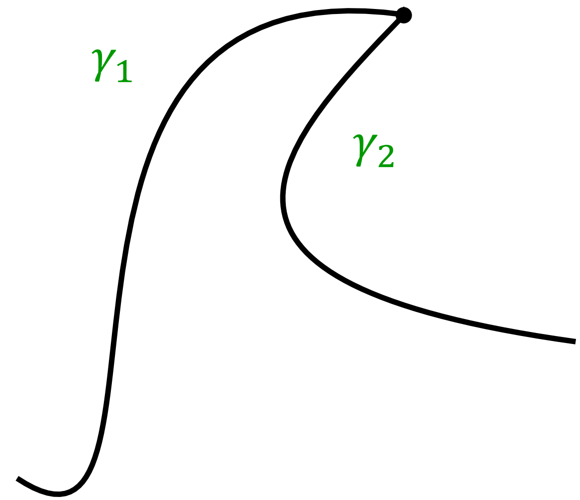
Continuity

Continuity and smoothness between parametric curves

$\gamma_1, \gamma_2 : [0,1] \to \mathbb{R}^d$



Not continuous



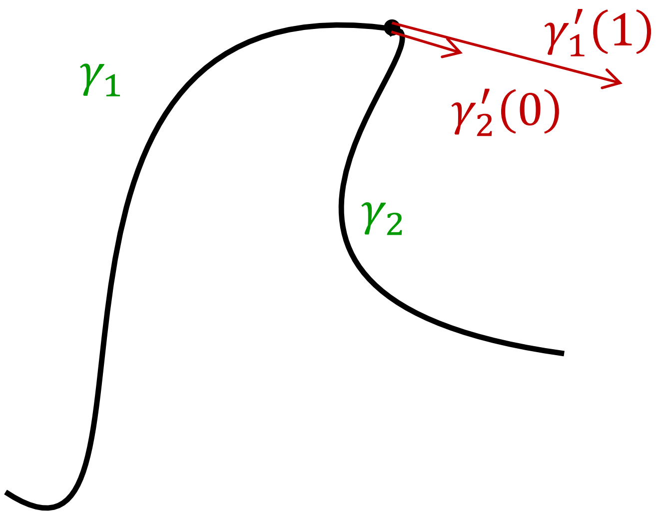
G^0- C^0-continuous

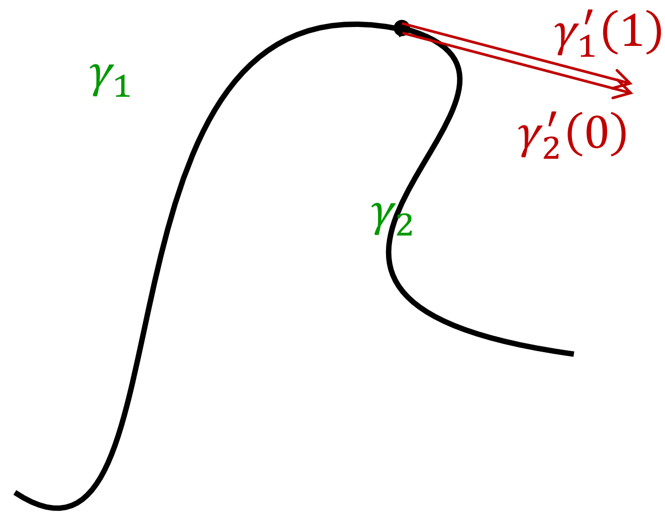
$\gamma_1(1) = \gamma_2(0)$

Continuity

Continuity and smoothness between parametric curves

$\gamma_1, \gamma_2 \colon [0,1] \to \mathbb{R}^d$





G^1-continuous

G^0 + tangent vectors parallel

$\gamma_1'(1) = p\gamma_2'(0), p \in \mathbb{R}_+$

C^1-continuous

C^0 + tangent vectors parallel

$\gamma_1'(1) = \gamma_2'(0)$

Continuity

Continuity and smoothness between parametric curves

$\gamma_1, \gamma_2 : [0,1] \to \mathbb{R}^d$

G^2-continuous

G^1 + $\gamma_1''(1) = p\gamma_2''(0), p \in \mathbb{R}_+$

⋮

C^2-continuous

C^1 + $\gamma_1''(1) = \gamma_2''(0)$

⋮

G^2 – smooth reflections

LAGRANGE INTERPOLATION

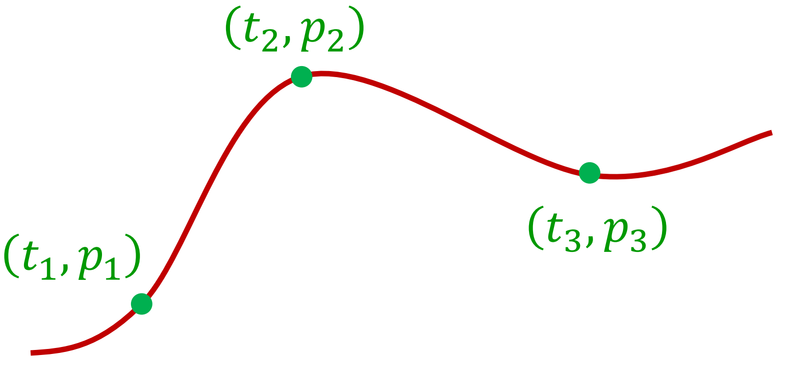
Lagrange Interpolation

Given a set of points:

$$(t_i, p_i), t \in \mathbb{R}, p_i \in \mathbb{R}^d$$

Find a polynomial P such that:

$$\forall i \; P(t_i) = p_i$$



Lagrange Interpolation

Given a set of *n* points:

$$(t_i, p_i), t \in \mathbb{R}, p_i \in \mathbb{R}^d$$

Find a polynomial P such that:

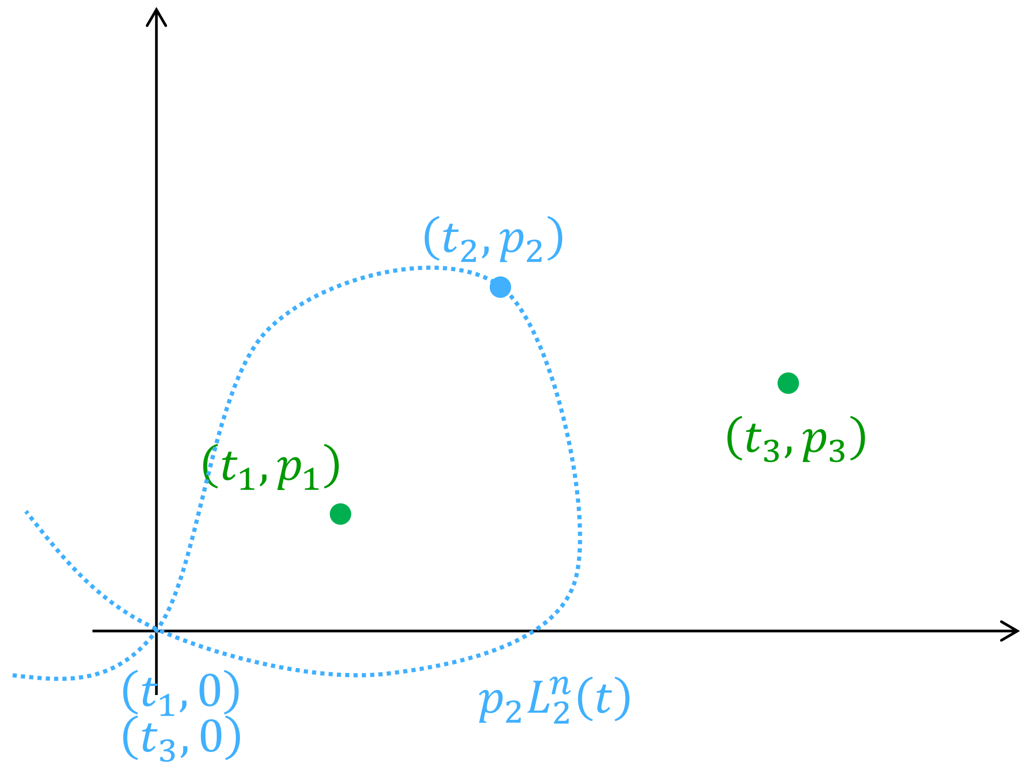
$$\forall i \; P(t_i) = p_i$$

For each point associate a Lagrange basis polynomial:

$$L_i^n(t) = \prod_{\substack{j \\ j \neq i}} \frac{t - t_j}{t_i - t_j}$$

$$(i \neq j) \quad L_i^n(t_j) = 0$$

$$L_i^n(t_i) = 1$$



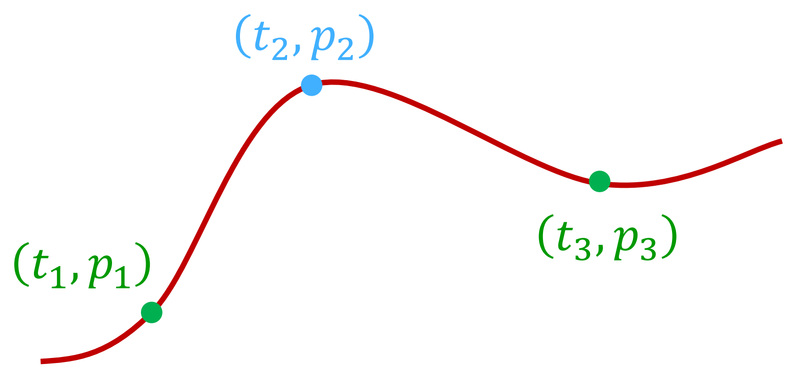
Lagrange Interpolation

Given a set of *n* points:

$$(t_i, p_i), t \in \mathbb{R}, p_i \in \mathbb{R}^d$$

Find a polynomial P such that:

$$\forall i \; P(t_i) = p_i$$



For each point associate a Lagrange basis polynomial:

$$L_i^n(t) = \prod_{\substack{j \\ j \neq i}} \frac{t - t_j}{t_i - t_j}$$

Add the Lagrange basis with points as weights:

$$P(t) = \sum_i L_i^n(t) \cdot p_i$$

Lagrange basis

$$P(t) = \begin{pmatrix} L_0^n & L_1^n & \cdots & L_{n-1}^n \end{pmatrix} \begin{pmatrix} p_{0_x} & p_{0_y} & p_{0_z} \\ p_{1_x} & p_{1_y} & p_{1_z} \\ \vdots & \vdots & \vdots \\ p_{n-1_x} & p_{n-1_y} & p_{n-1_z} \end{pmatrix}$$

Lagrange Interpolation

Given a set of *n* points:

$$(t_i, p_i), t \in \mathbb{R}, p_i \in \mathbb{R}^d$$

Find a polynomial P such that:

$$\forall i \; P(t_i) = p_i$$

For each point associate a Lagrange basis polynomial:

$$L_i^n(t) = \prod_{\substack{j \\ j \neq i}} \frac{t - t_j}{t_i - t_j}$$

Add the Lagrange basis with points as weights:

$$P(t) = \sum_i p_i L_i^n(t)$$

Given 2 points

$$L_0^2(t) = \frac{t - t_1}{t_0 - t_1}$$

$$L_1^2(t) = \frac{t - t_0}{t_1 - t_0}$$

$P(t) =$ linear interpolation

Given 3 points

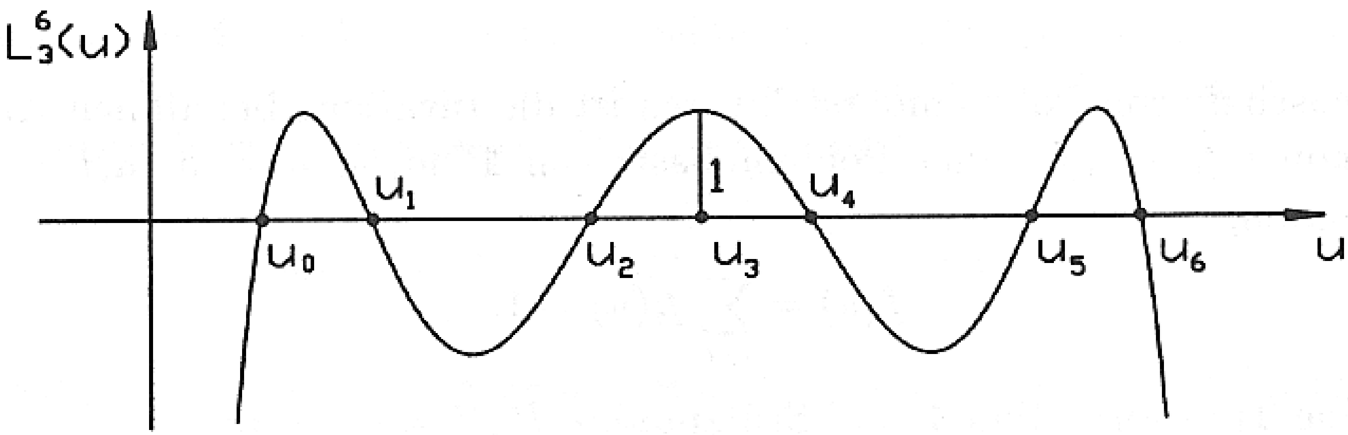
$$L_0^3(t) = \frac{t - t_1}{t_0 - t_1} \frac{t - t_2}{t_0 - t_2}$$

...

$P(t) =$ quadratic interpolation

Problems

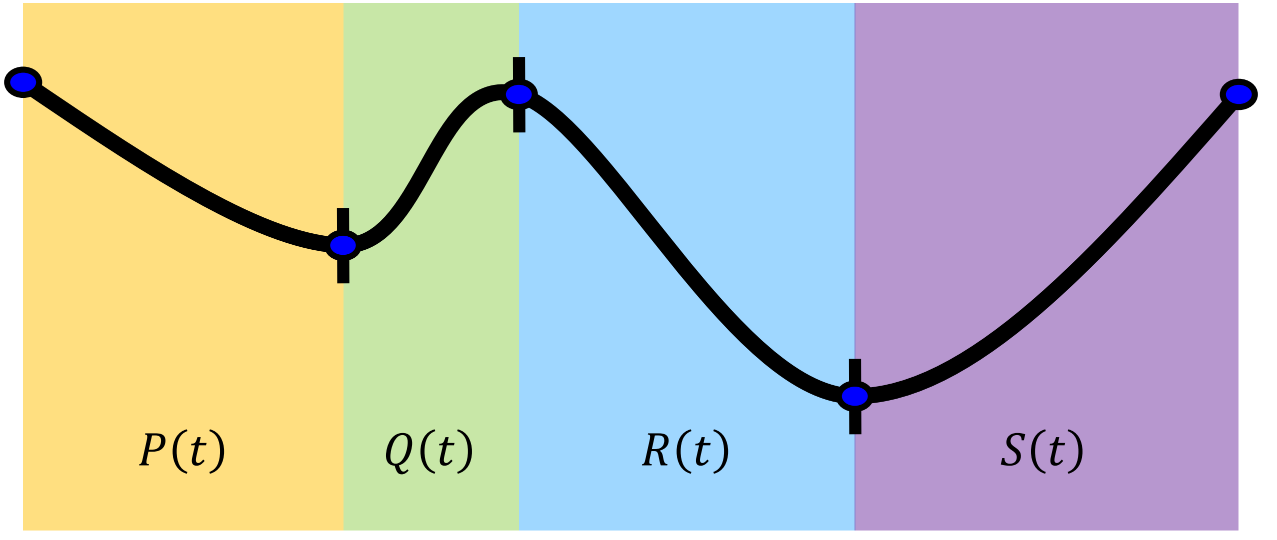
- **Problems with a single polynomial**
  - Degree depends on the number of interpolation constraints
  - Strong overshooting for high degree (n > 7)
  - Problems with smooth joints
  - Numerically unstable
  - No local changes



SPLINES

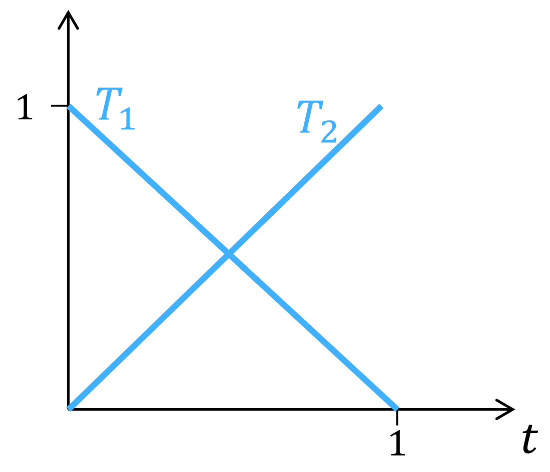
Splines

- **Functions for interpolation & approximation**
  - Standard curve and surface primitives in geometric modeling
  - Key frame and in-betweens in animations
  - Filtering and reconstruction of images
- **Historically**
  - Name for a tool in ship building
    - Flexible metal strip that tries to stay straight
  - Within computer graphics:
    - Piecewise polynomial function



Linear Interpolation

- Defined by two points: p_1, p_2
- Searching for $P(t)$ such that:
  - $P(0) = p_1$
  - $P(1) = p_2$
  - Degree of P is 1



Basis:

$T_1(t) = 1 - t$

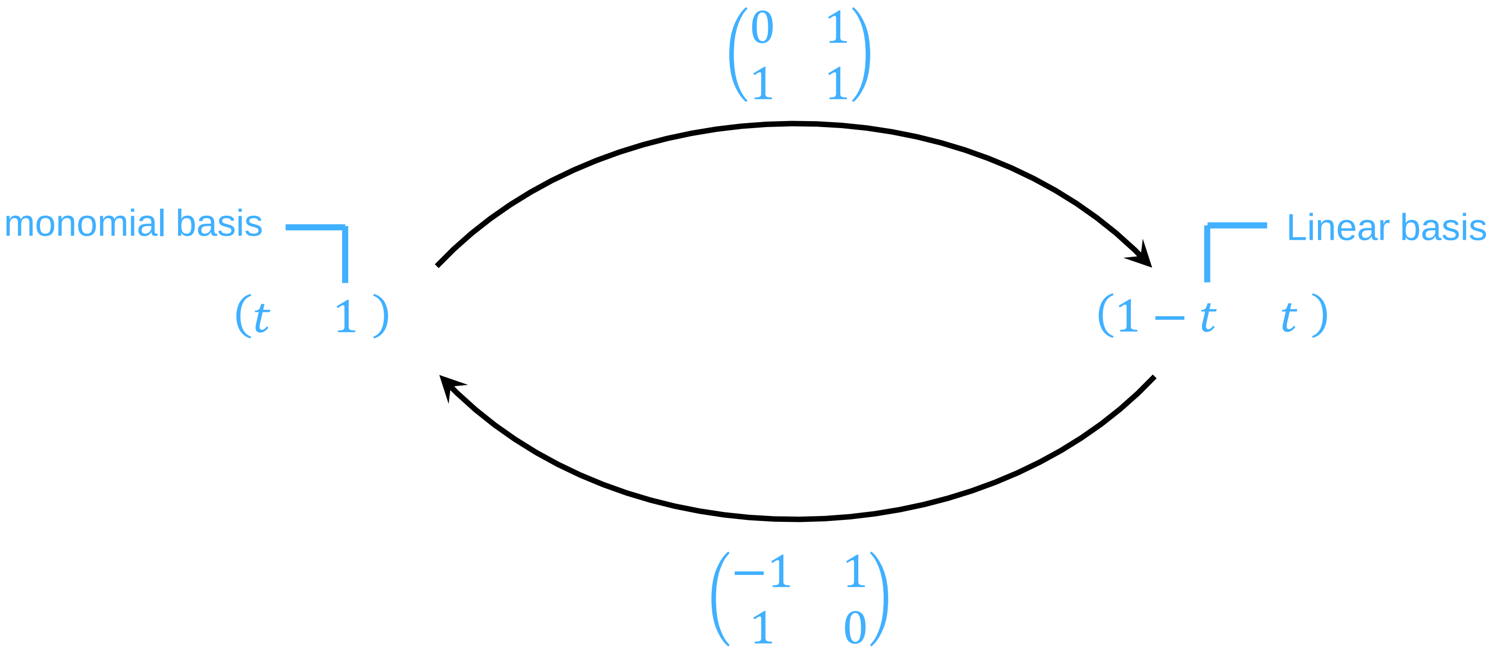
$T_2(t) = t$

$$P(t) = p_1 T_1(t) + p_2 T_2(t)$$

Linear basis

$$P(t)^T = \begin{pmatrix} 1-t & t \end{pmatrix} \begin{pmatrix} p_1^T \\ p_2^T \end{pmatrix}$$

Linear Interpolation

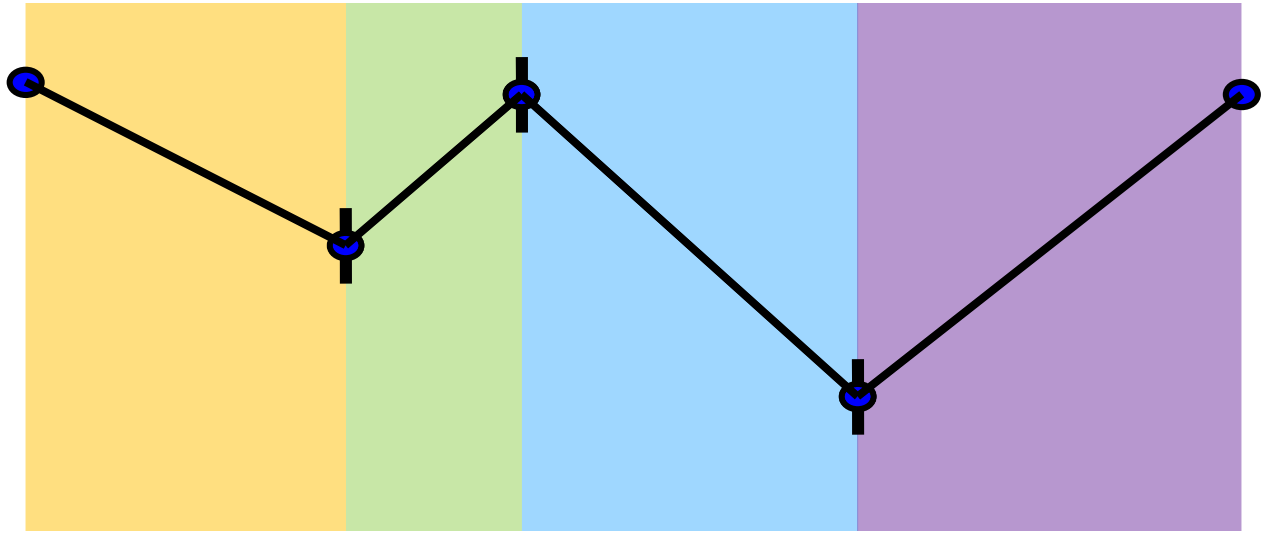


$$P(t)^T = M \cdot \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} p_1^T \\ p_2^T \end{pmatrix}$$

Linear Interpolation

$$P(t) = M \cdot \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$$

C^0-continuous



Cubic Hermite Interpolation

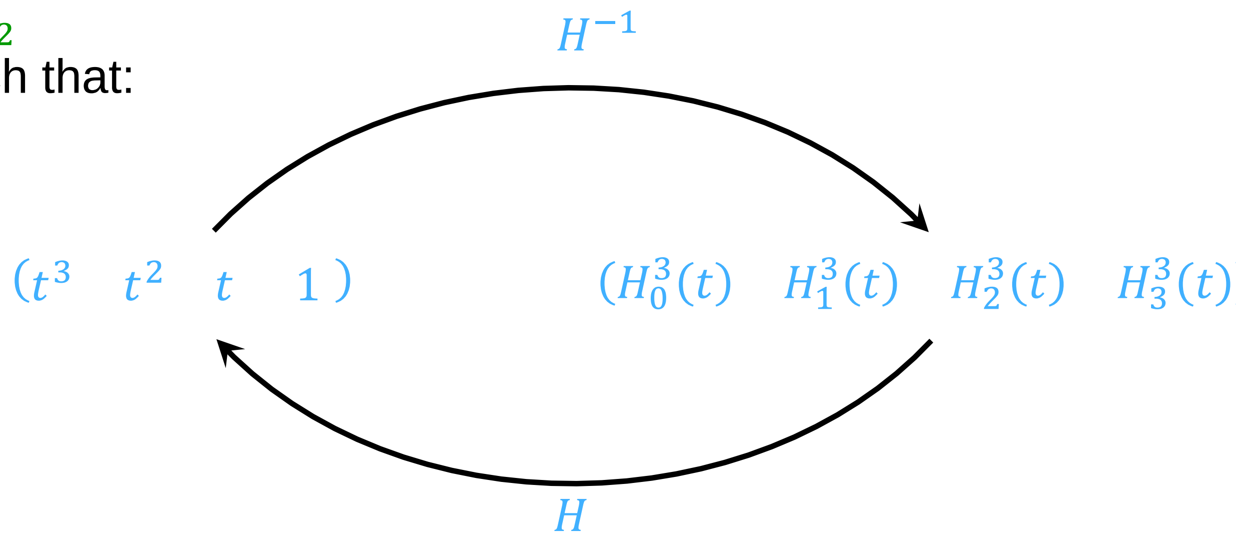
- Defined by
  - two points: p_1, p_2
  - two tangents: t_1, t_2
- Searching for $P(t)$ such that:
  - $P(0) = p_1$
  - $P'(0) = t_1$
  - $P'(1) = t_2$
  - $P(1) = p_2$
  - Degree of P is 3

Basis:

$H_0^3(t) = ?$
$H_1^3(t) = ?$
$H_2^3(t) = ?$
$H_3^3(t) = ?$

Cubic Hermite Interpolation

- Defined by
  - two points: p_1, p_2
  - two tangents: t_1, t_2
- Searching for $P(t)$ such that:
  - $P(0) = p_1$
  - $P'(0) = t_1$
  - $P'(1) = t_2$
  - $P(1) = p_2$
  - Degree of P is 3



Basis:

$H_0^3(t) = ?$
$H_1^3(t) = ?$
$H_2^3(t) = ?$
$H_3^3(t) = ?$

$$P(t)^T = M \cdot H \cdot \begin{pmatrix} p_1^T \\ t_1^T \\ t_2^T \\ p_2^T \end{pmatrix} = M \cdot H \cdot G$$

Cubic Hermite Interpolation

- Defined by
  - two points: p_1, p_2
  - two tangents: t_1, t_2
- Searching for $P(t)$ such that:
  - $P(0) = p_1$
  - $P'(0) = t_1$
  - $P'(1) = t_2$
  - $P(1) = p_2$
  - Degree of P is 3

$$P(t)^T = \begin{pmatrix} t^3 & t^2 & t^1 & 1 \end{pmatrix} \cdot H \cdot G$$

$$P'(t)^T = \begin{pmatrix} 3t^2 & 2t & 1 & 0 \end{pmatrix} \cdot H \cdot G$$

$$p_1^T = P(0)^T = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \cdot H \cdot G$$

$$t_1^T = P'(0)^T = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \cdot H \cdot G$$

$$t_2^T = P'(1)^T = \begin{pmatrix} 3 & 2 & 1 & 0 \end{pmatrix} \cdot H \cdot G$$

$$p_2^T = P(1)^T = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \cdot H \cdot G$$

$$\begin{pmatrix} p_1^T \\ t_1^T \\ t_2^T \\ p_2^T \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \cdot H \cdot \begin{pmatrix} p_1^T \\ t_1^T \\ t_2^T \\ p_2^T \end{pmatrix}$$

Cubic Hermite Interpolation

- Defined by
  - two points: p_1, p_2
  - two tangents: t_1, t_2
- Searching for $P(t)$ such that:
  - $P(0) = p_1$
  - $P'(0) = t_1$
  - $P'(1) = t_2$
  - $P(1) = p_2$
  - Degree of P is 3

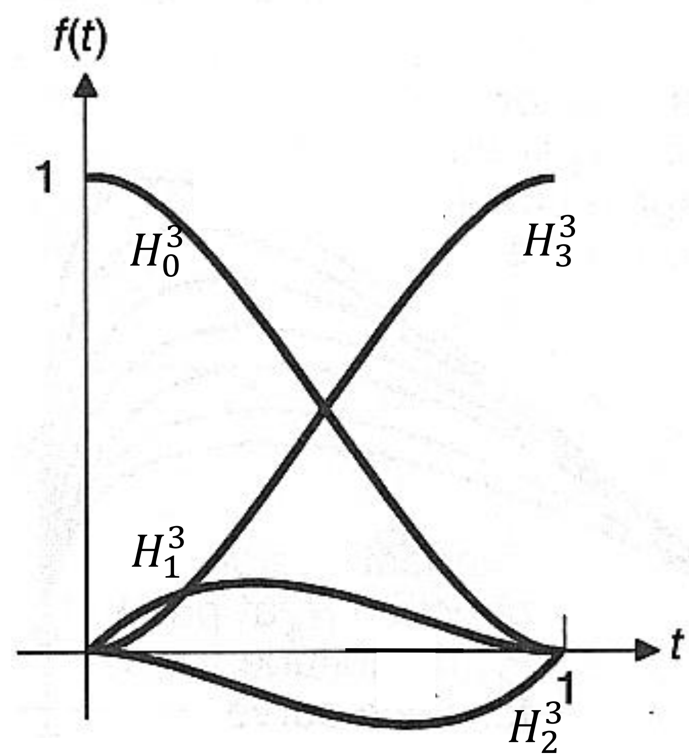
$$H = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & 1 & 1 & -2 \\ -3 & -2 & -1 & 3 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Cubic Hermite Interpolation

- Defined by
  - two points: p_1, p_2
  - two tangents: t_1, t_2
- Searching for $P(t)$ such that:
  - $P(0) = p_1$
  - $P'(0) = t_1$
  - $P'(1) = t_2$
  - $P(1) = p_2$
  - Degree of P is 3

$$H = \begin{pmatrix} 2 & 1 & 1 & -2 \\ -3 & -2 & -1 & 3 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

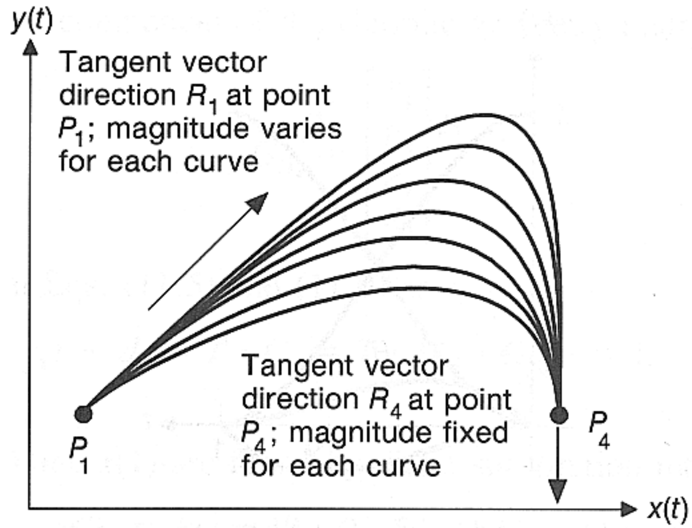
$$\left(H_0^3(t) \quad H_1^3(t) \quad H_2^3(t) \quad H_3^3(t)\right)$$

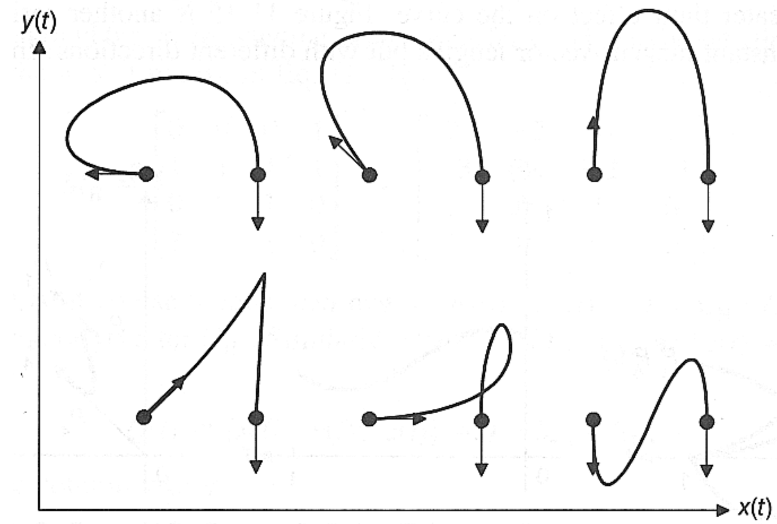


Basis:

$$H_0^3(t) = (1-t)^2(1+2t)$$
$$H_1^3(t) = t(1-t)^2$$
$$H_2^3(t) = t^2(t-1)$$
$$H_3^3(t) = (3-2t)t^2$$

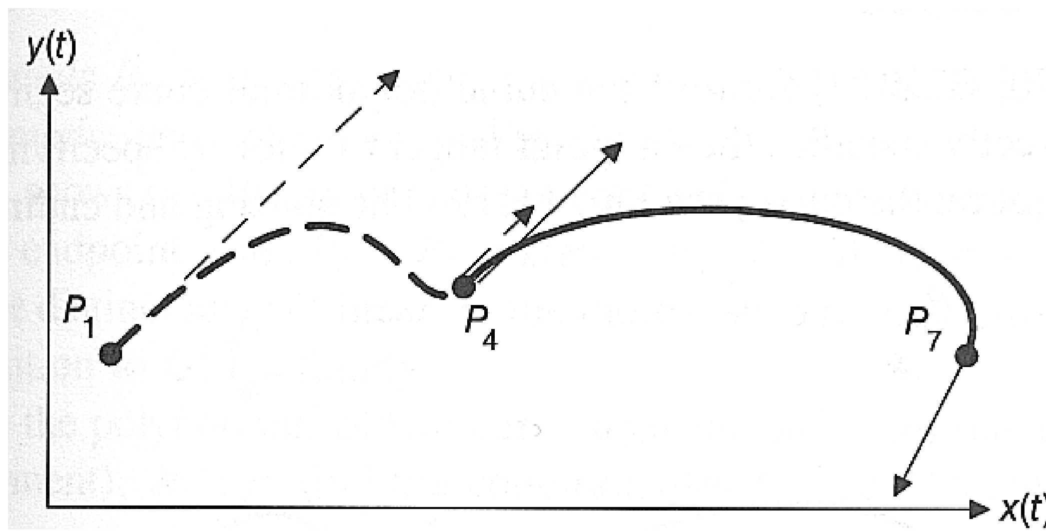
Cubic Hermite Interpolation





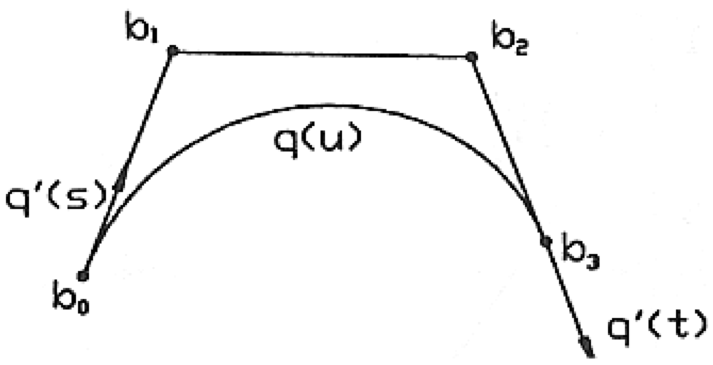
G¹-continuous

C¹-continuous



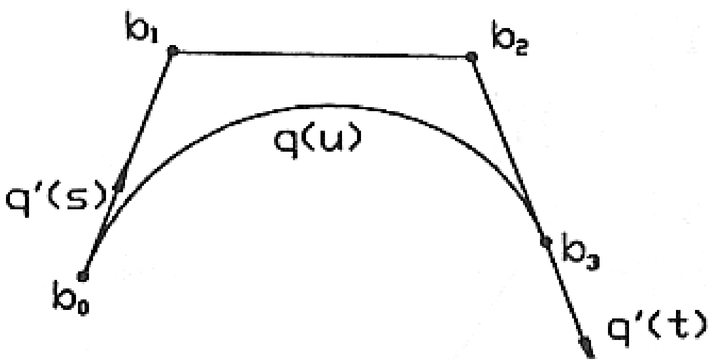
Bézier

- Defined by
  - b_0 - start point
  - b_3 - end point
  - b_1, b_2 - control points that are approximated
- Searching for $P(t)$ such that:
  - $P(0) = b_0$
  - $P'(0) = 3(b_1 - b_0)$
  - $P'(1) = 3(b_3 - b_2)$
  - $P(1) = b_3$
  - Degree of P is 3



Bézier

- Defined by
  - b_0 - start point
  - b_3 - end point
  - b_1, b_2 - control points that are approximated
- Searching for $P(t)$ such that:
  - $P(0) = b_0$
  - $P'(0) = 3(b_1 - b_0)$
  - $P'(1) = 3(b_3 - b_2)$
  - $P(1) = b_3$
  - Degree of P is 3



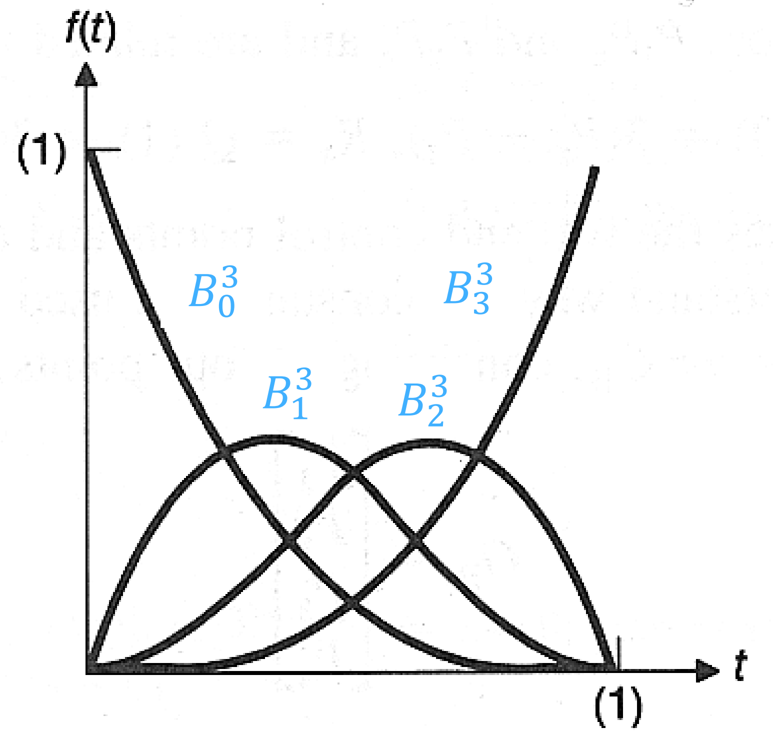
$$\begin{pmatrix} p_1^T \\ t_1^T \\ t_2^T \\ p_2^T \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_0^T \\ b_1^T \\ b_2^T \\ b_3^T \end{pmatrix}$$

$$P(t)^T = M \cdot H \cdot T_{BH} \cdot G$$

Bézier

- Defined by
  - b_0 - start point
  - b_3 - end point
  - b_1, b_2 - control points that are approximated
- Searching for $P(t)$ such that:
  - $P(0) = b_0$
  - $P'(0) = 3(b_1 - b_0)$
  - $P'(1) = 3(b_3 - b_2)$
  - $P(1) = b_3$
  - Degree of P is 3

$$B = H \cdot T_{BH} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$



Basis:

$$B_0^3(t) = (1-t)^3$$
$$B_1^3(t) = 3(1-t)^2 t$$
$$B_2^3(t) = 3(1-t)t^2$$
$$B_3^3(t) = t^3$$

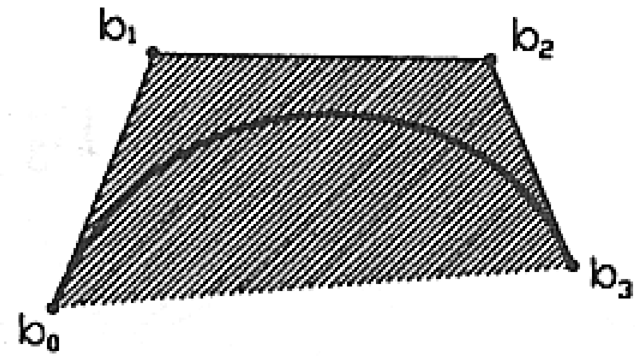
Bernstein polynomial:

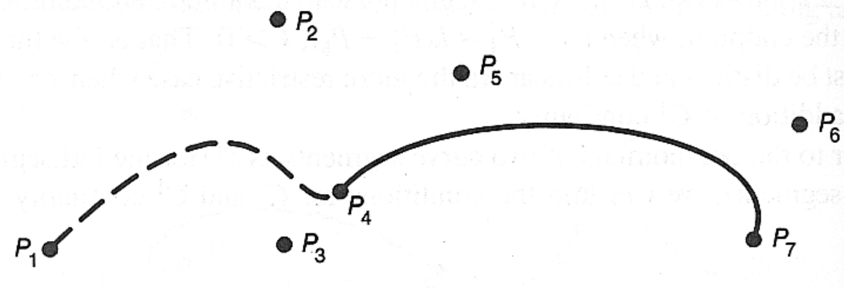
$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

$$P(t) = \sum_i b_i B_i^n(t)$$

Bézier properties

- **Advantages:**
  - End point interpolation
  - Tangents explicitly specified
  - Smooth joints are simple
    - P_3, P_4, P_5 collinear → G^1 continuous
    - P_5-P_4=P_4-P_3 → C^1 continuous
  - Geometric meaning of control points
  - Affine invariance
  - Convex hull property
    - For $0<t<1$: $B_i(t) \geq 0$
  - Symmetry: $B_i(t) = B_{n-i}(1-t)$





- **Disadvantages**
  - Smooth joints need to be maintained explicitly
    - Automatic in B-Splines (and NURBS)

DeCasteljau Algorithm

Bernstein polynomial defined recursively:

$$P(t) = \sum_i b_i B_i^n(t)$$

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} = tB_{i-1}^{n-1}(t) + (1-t)B_i^{n-1}(t)$$

Recursive point computation:

$$b_i^0(t) = b_i$$

$$b_i^{k+1}(t) = tb_{i+1}^k(t) + (1-t)b_i^k(t)$$

$$\sum_i b_i^k(t) B_i^{n-k}(t) =$$

$$\sum_i b_i^k(t) t B_{i-1}^{n-k-1}(t) + b_i^k(t)(1-t)B_i^{n-k-1}(t) =$$

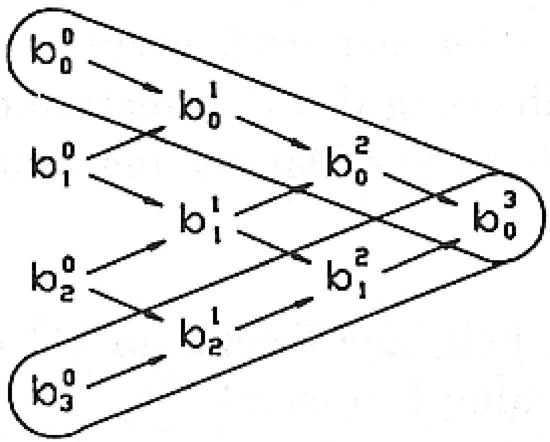
$$\sum_i b_{i+1}^k(t) t B_i^{n-k-1}(t) + b_i^k(t)(1-t)B_i^{n-k-1}(t) =$$

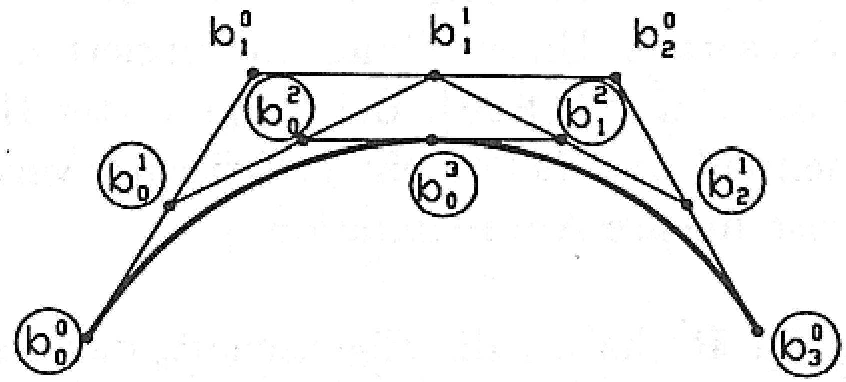
$$\sum_i b_i^{k+1}(t) B_i^{n-k-1}(t)$$

$$P(t) = \cdots = b_0^n(t)$$

DeCasteljau Algorithm

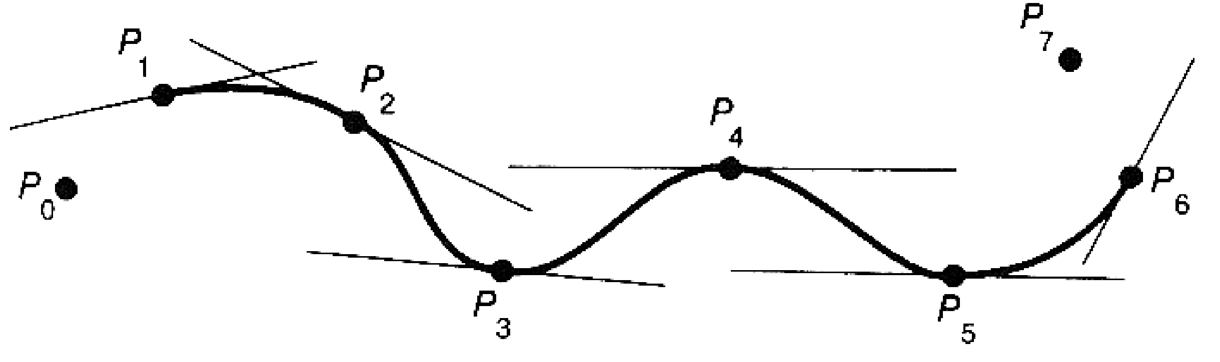
t= 0.5





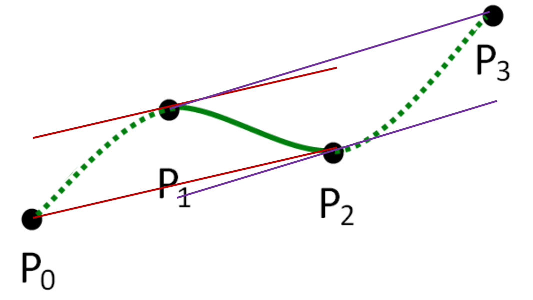
Catmull-Rom-Splines

- **Goal**
  - Smooth (C^1)-joints between (cubic) spline segments
- **Algorithm**
  - Tangents given by neighboring points P_{i-1} P_{i+1}
  - Construct (cubic) Hermite segments
- **Advantage**
  - Arbitrary number of control points
  - Interpolation without overshooting
  - Local control



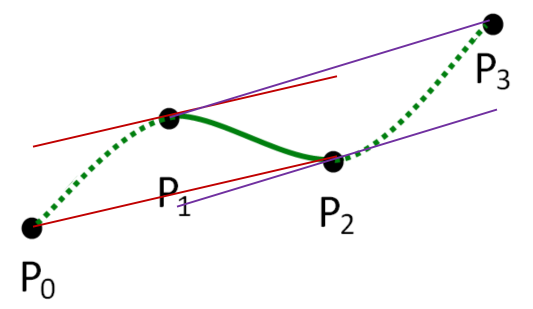
Catmull-Rom-Splines

- Each segment defined by
  - c_1 - start point
  - c_2 - end point
  - c_0, c_3 - neighbor segment points
- Searching for $P(t)$ such that:
  - $P(0) = c_1$
  - $P'(0) = \frac{1}{2}(c_2 - c_0)$
  - $P'(1) = \frac{1}{2}(c_3 - c_1)$
  - $P(1) = c_2$
  - Degree of P is 3



Catmull-Rom-Splines

- Each segment defined by
  - c_1 - start point
  - c_2 - end point
  - c_0, c_3 - neighbor segment points
- Searching for $P(t)$ such that:
  - $P(0) = c_1$
  - $P'(0) = \frac{1}{2}(c_2 - c_0)$
  - $P'(1) = \frac{1}{2}(c_3 - c_1)$
  - $P(1) = c_2$
  - Degree of P is 3

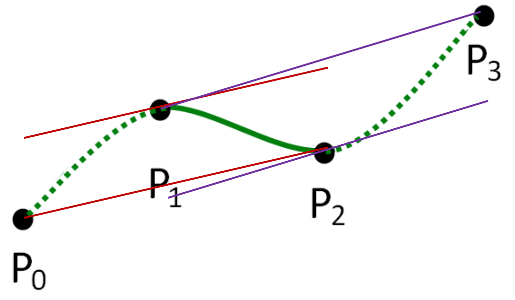


$$\begin{pmatrix} p_1^T \\ t_1^T \\ t_2^T \\ p_2^T \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} c_0^T \\ c_1^T \\ c_2^T \\ c_3^T \end{pmatrix}$$

$$P(t)^T = M \cdot H \cdot T_{CH} \cdot G$$

Catmull-Rom-Splines

- Each segment defined by
  - c_1 - start point
  - c_2 - end point
  - c_0, c_3 - neighbor segment points
- Searching for $P(t)$ such that:
  - $P(0) = c_1$
  - $P'(0) = \frac{1}{2}(c_2 - c_0)$
  - $P'(1) = \frac{1}{2}(c_3 - c_1)$
  - $P(1) = c_2$
  - Degree of P is 3



$$C = H \cdot T_{BH} = \frac{1}{2}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}$$

Basis:

$$C_0\ (t) = -\frac{1}{2}t(t-1)^2$$

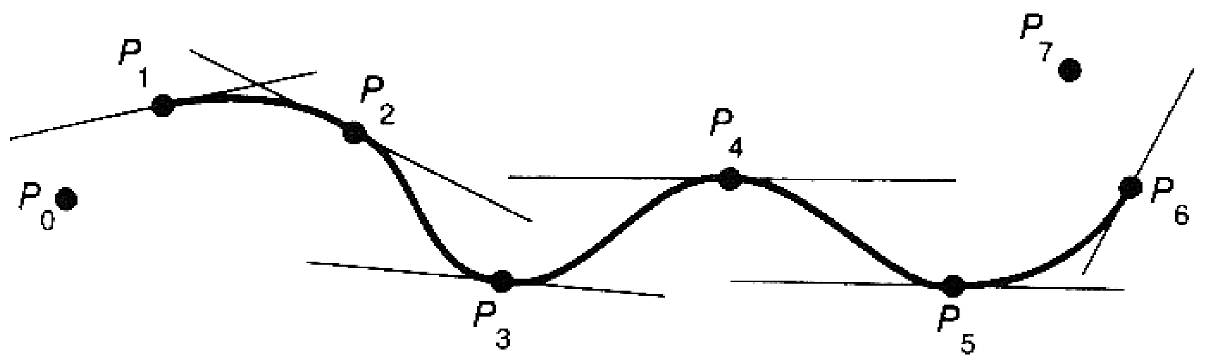
$$C_1\ (t) = \frac{1}{2}(t-1)(3t^2-2t-2)$$

$$C_2\ (t) = -\frac{1}{2}t(3t^2-4t-1)$$

$$C_3\ (t) = \frac{1}{2}t^2(t-1)$$

Catmull-Rom-Splines

- **Catmull-Rom-Spline**
  - Piecewise polynomial curve
  - Four control points per segment
  - For n control points we obtain (n-3) polynomial segments



- **Application**
  - Smooth interpolation of a given sequence of points
  - Key frame animation, camera movement, etc.
  - Control points should be equidistant in time

Choice of Parameterization

- **Problem**

–Often only the control points are given

–How to obtain a suitable parameterization t_i ?

- **Example: Chord-Length Parameterization**

$$t_0 = 0$$

$$t_i = \sum_{j=1}^{i} dist(P_i - P_{i-1})$$

–Arbitrary up to a constant factor

»**Warning**

–Distances are not affine invariant !

–Shape of curves changes under transformations !!

Choice of Parameterization

- **Chord-Length versus uniform Parameterization**
  - –Analog: Think P(t) as a moving object with mass that may overshoot