Computer Graphics 2018

9. Splines and Curves

Hongxin Zhang
State Key Lab of CAD&CG, Zhejiang University

2018-11-28

About homework 3

- an alternative solution with WebGL

- links:
  - WebGL lessons
    http://learningwebgl.com/blog/?page_id=1217
  - My simple test
    https://github.com/hongxin/PonyGL
  - Please use google's browser: chrome

classification of curves

$y = x^2+5x+3 \longrightarrow$ **y=f(x)**

(*explicit* curve)

$(x-x_c)^2 + (y-y_c)^2 - r^2=0 \longrightarrow$ **g(x,y)=0**

(*implicit* curve)

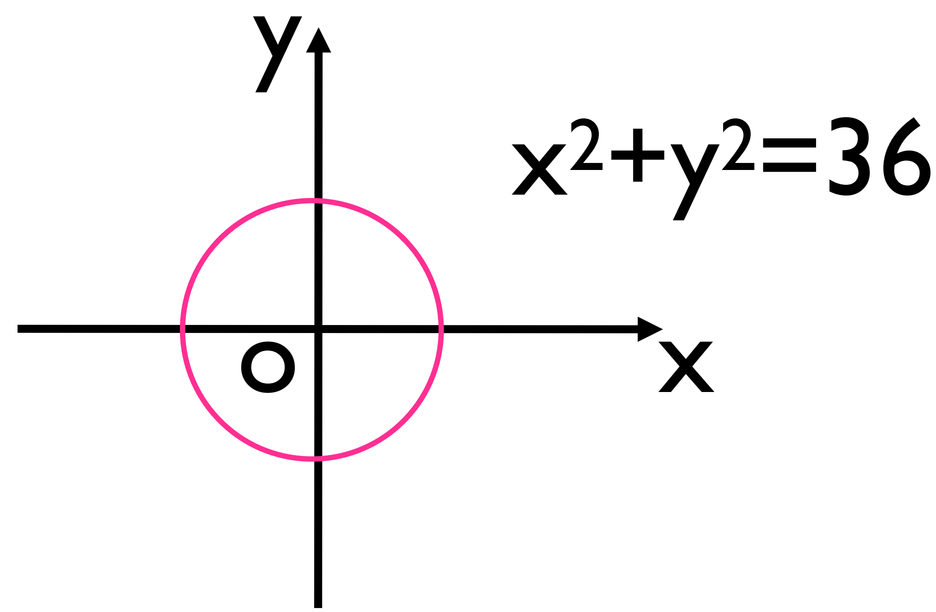
$$\begin{aligned} x &= x_c + r \cdot \cos\theta \\ y &= y_c + r \cdot \sin\theta \end{aligned} \longrightarrow \begin{cases} x = x(t) \\ y = y(t) \end{cases}$$

(*parametric* curve)

classification of curves

implicit curve

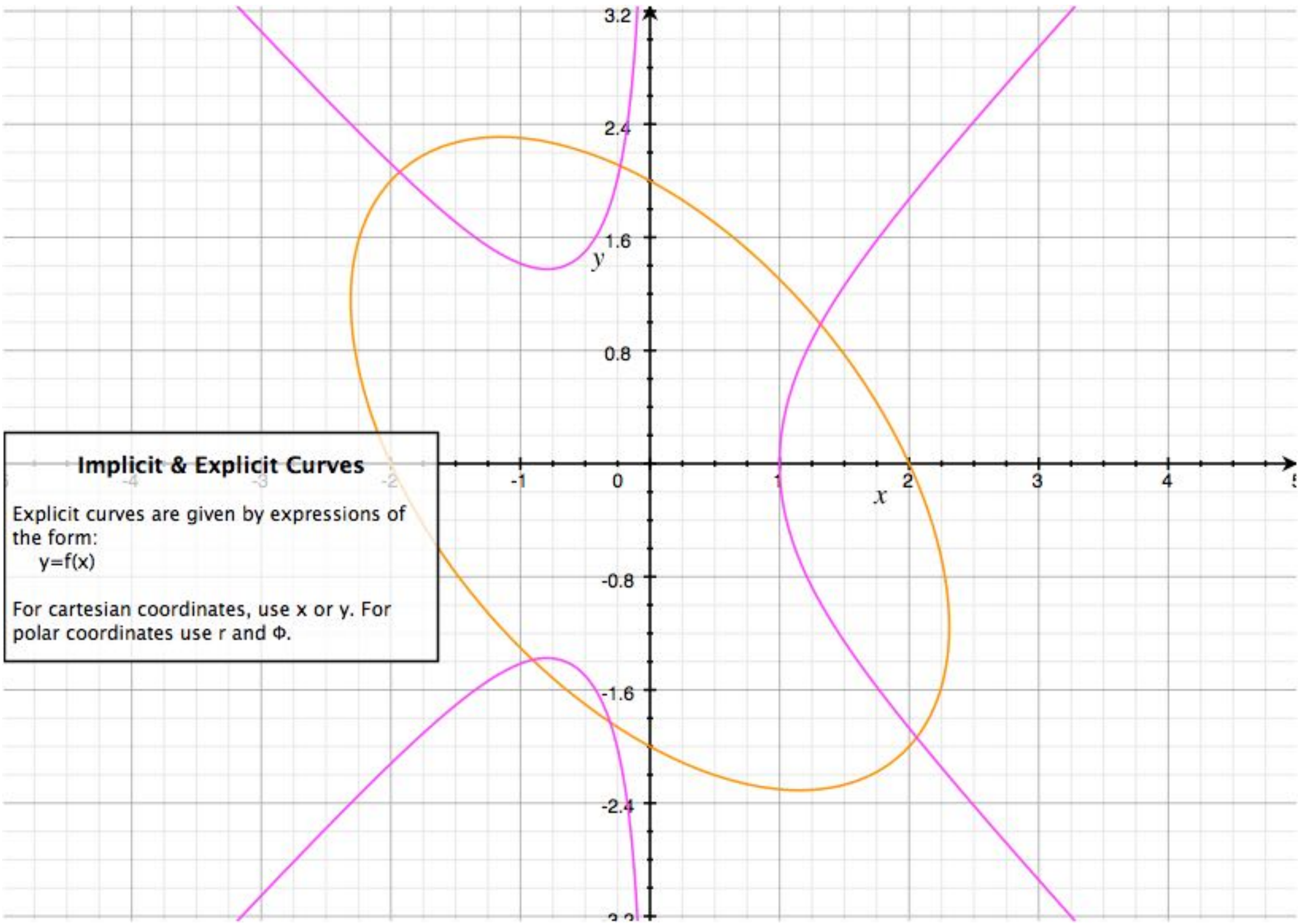
- Planar curve: $f(x,y)=0$: $x^2+y^2-36=0$



- 3D curve

$$\begin{cases} f(x,y,z)=0, \\ g(x,y,z)=0. \end{cases}$$

More examples from Grapher



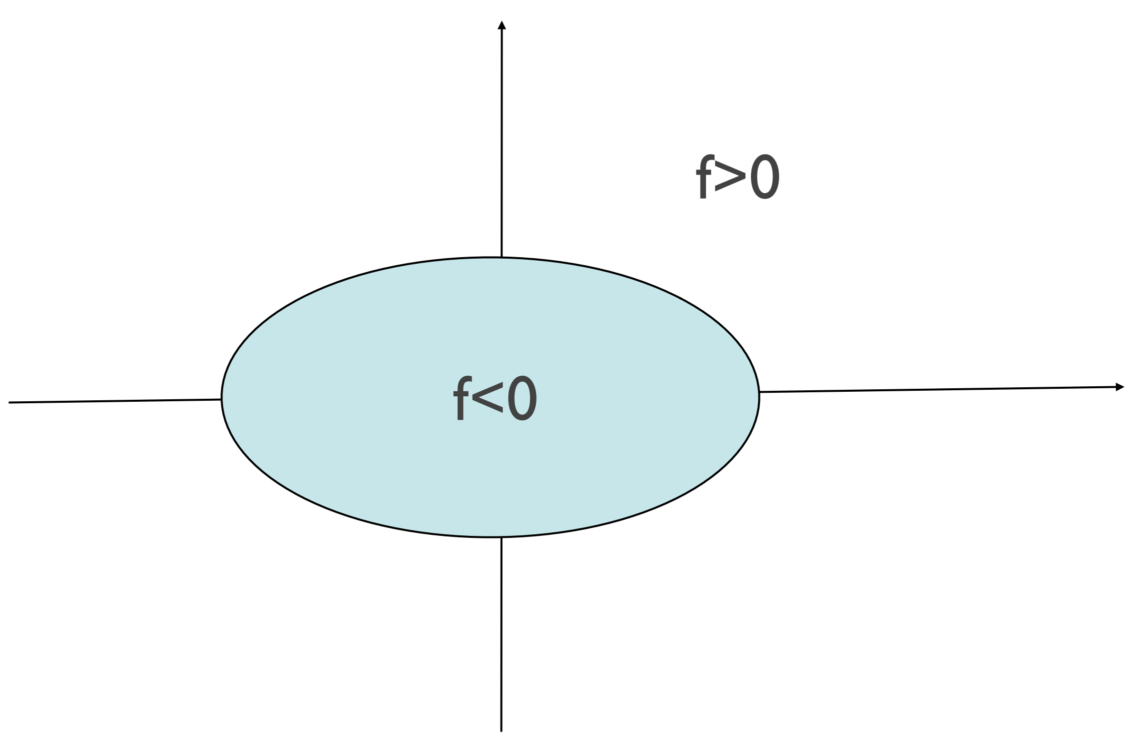
implicit curves

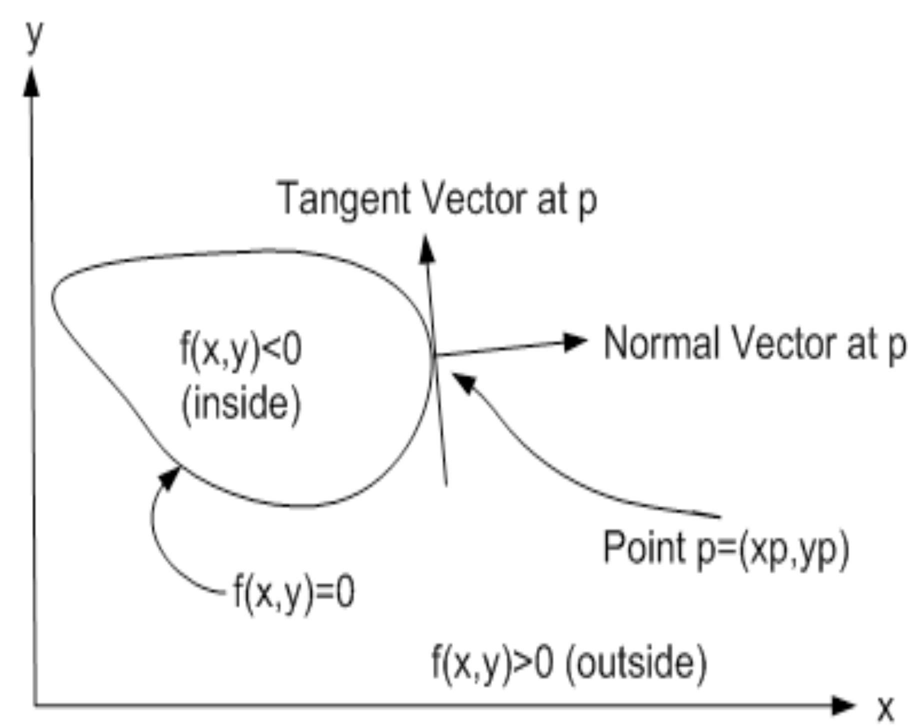
advantage of implicit curve:

To a point (x,y), it is easy to detect whether f(x,y) is >0 ,<0 or =0.

disadvantage of implicit curve:

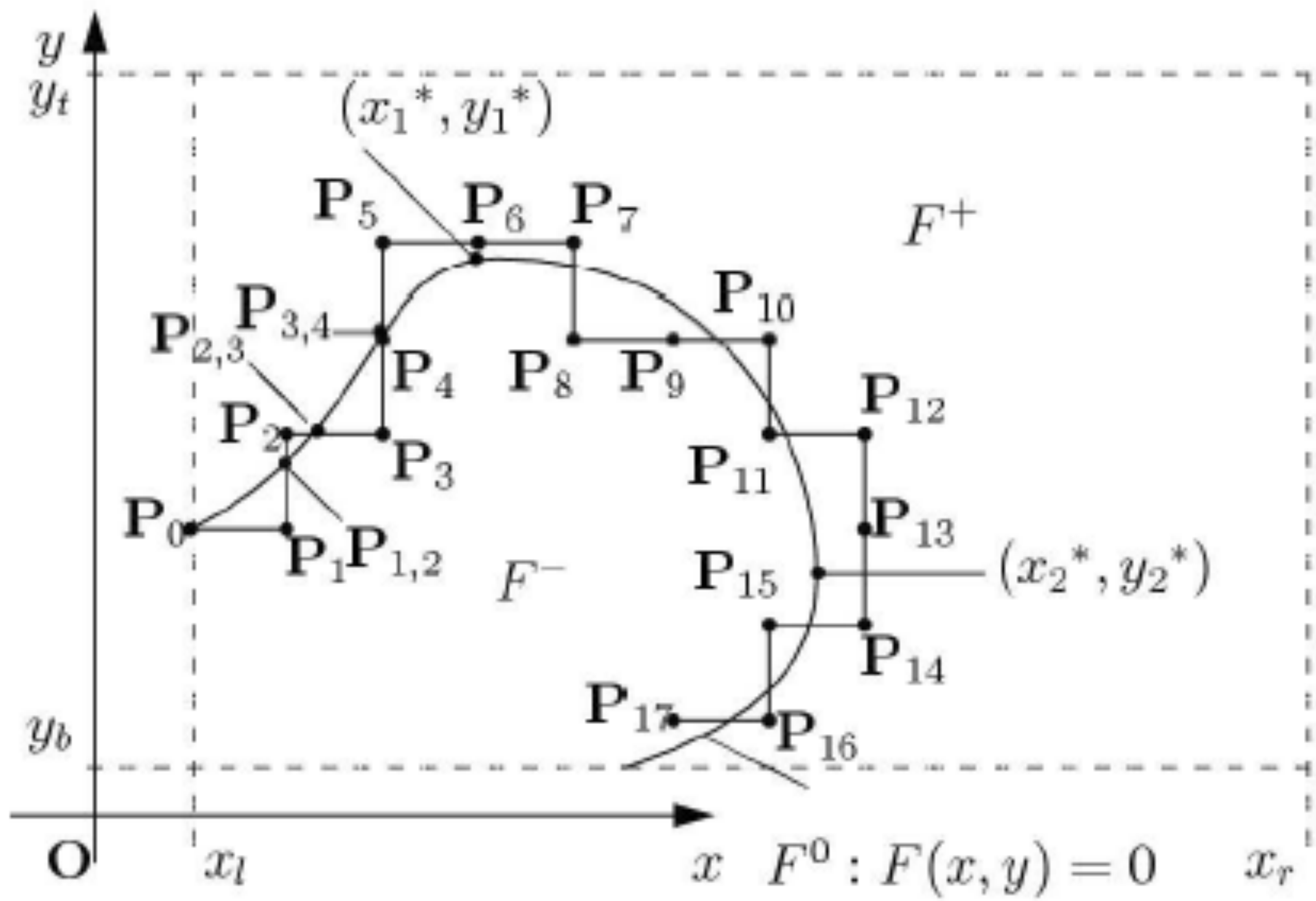
To a curve f(x,y)=0, it is difficult to find the point on it..





implicit curves

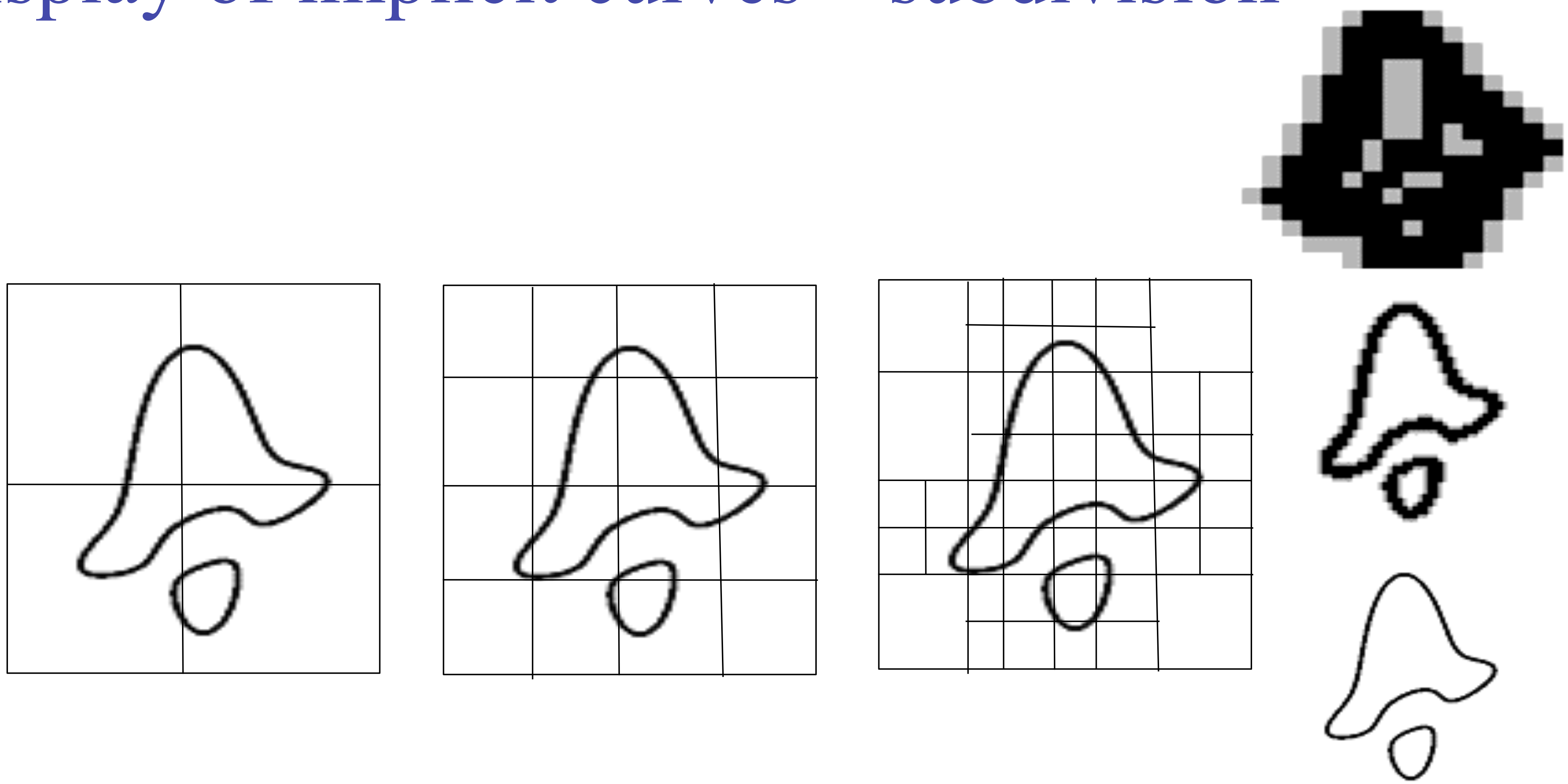
Display of implicit curves---chain coding



浙江大学计算机学院
数字媒体与网络技术

implicit curves

Display of implicit curves---subdivision

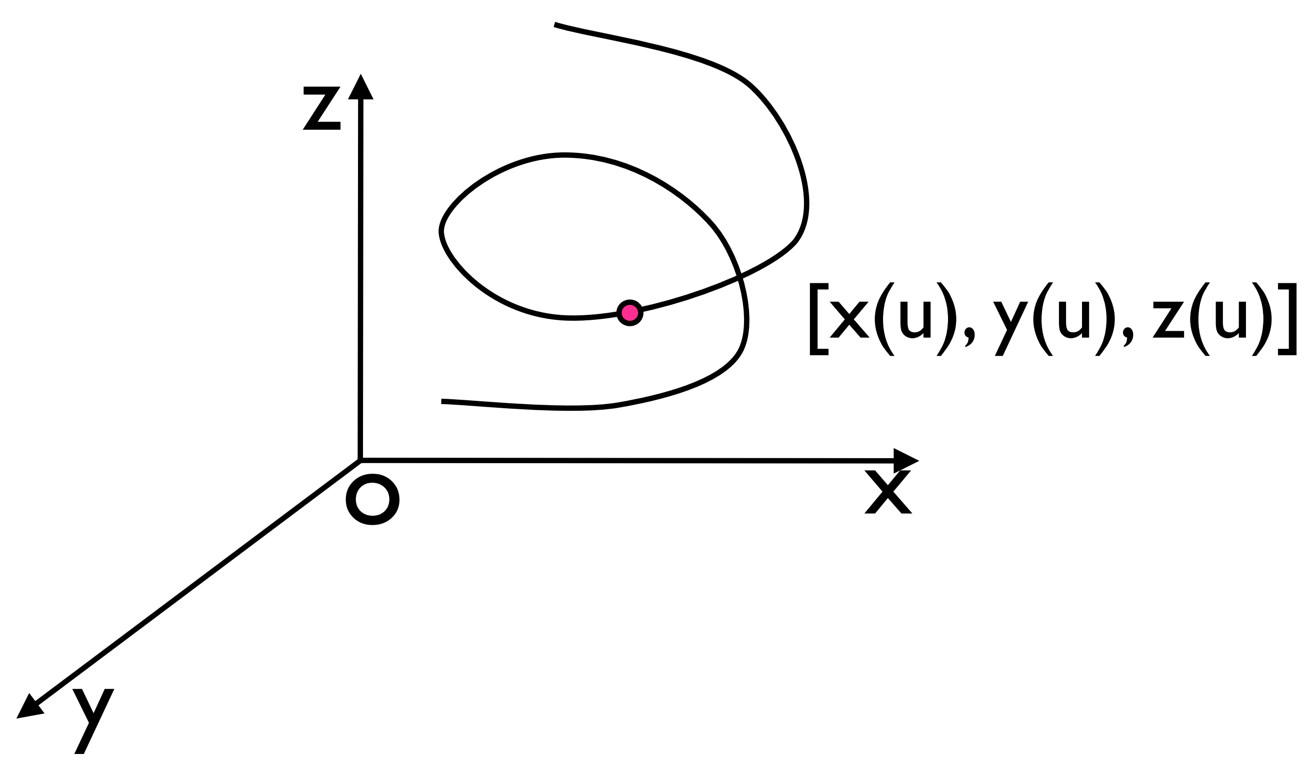


Parametric curves

- variable is a scalar, and function is a vector:

$$\boldsymbol{C}=\boldsymbol{C}(u)=[x(u), y(u), z(u)],$$

- Every element of the vector is a function of the variable(the parameter)



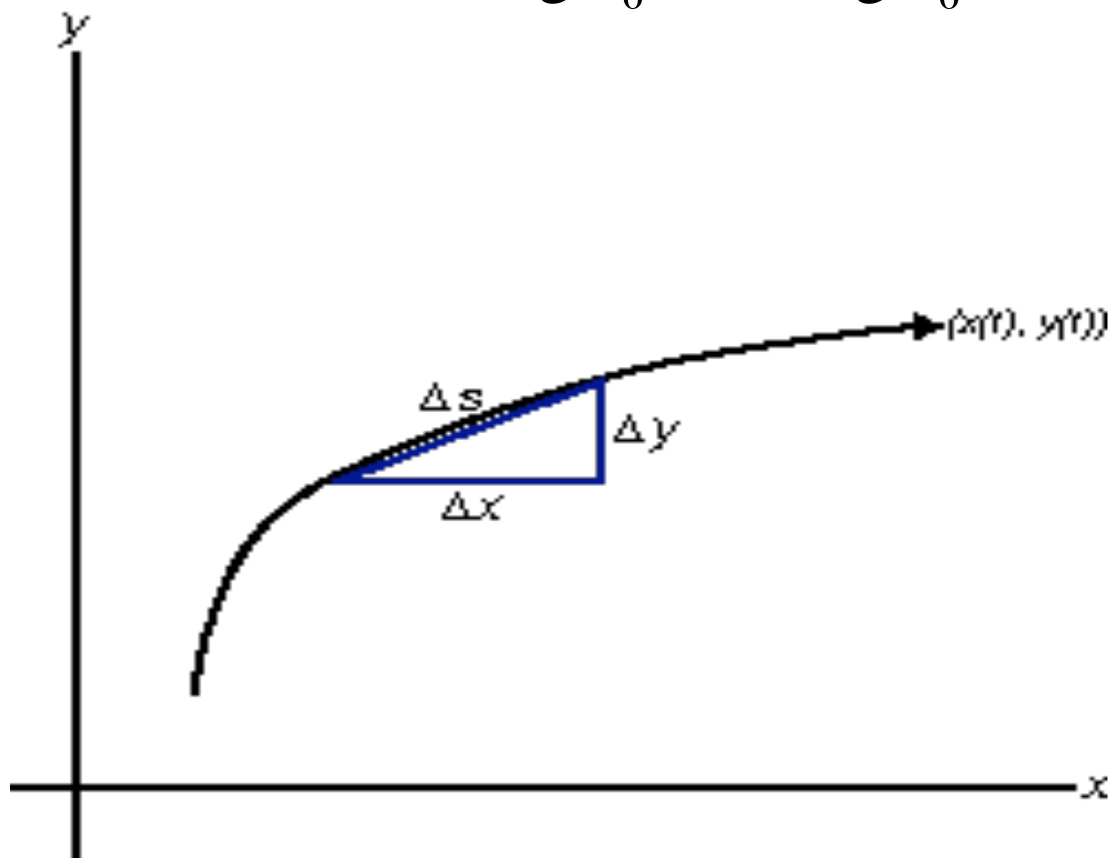
Parametric curves

given a curve ***C***(*u*), its tangent is ***T***=***C***'(*u*).

difference of arc length:

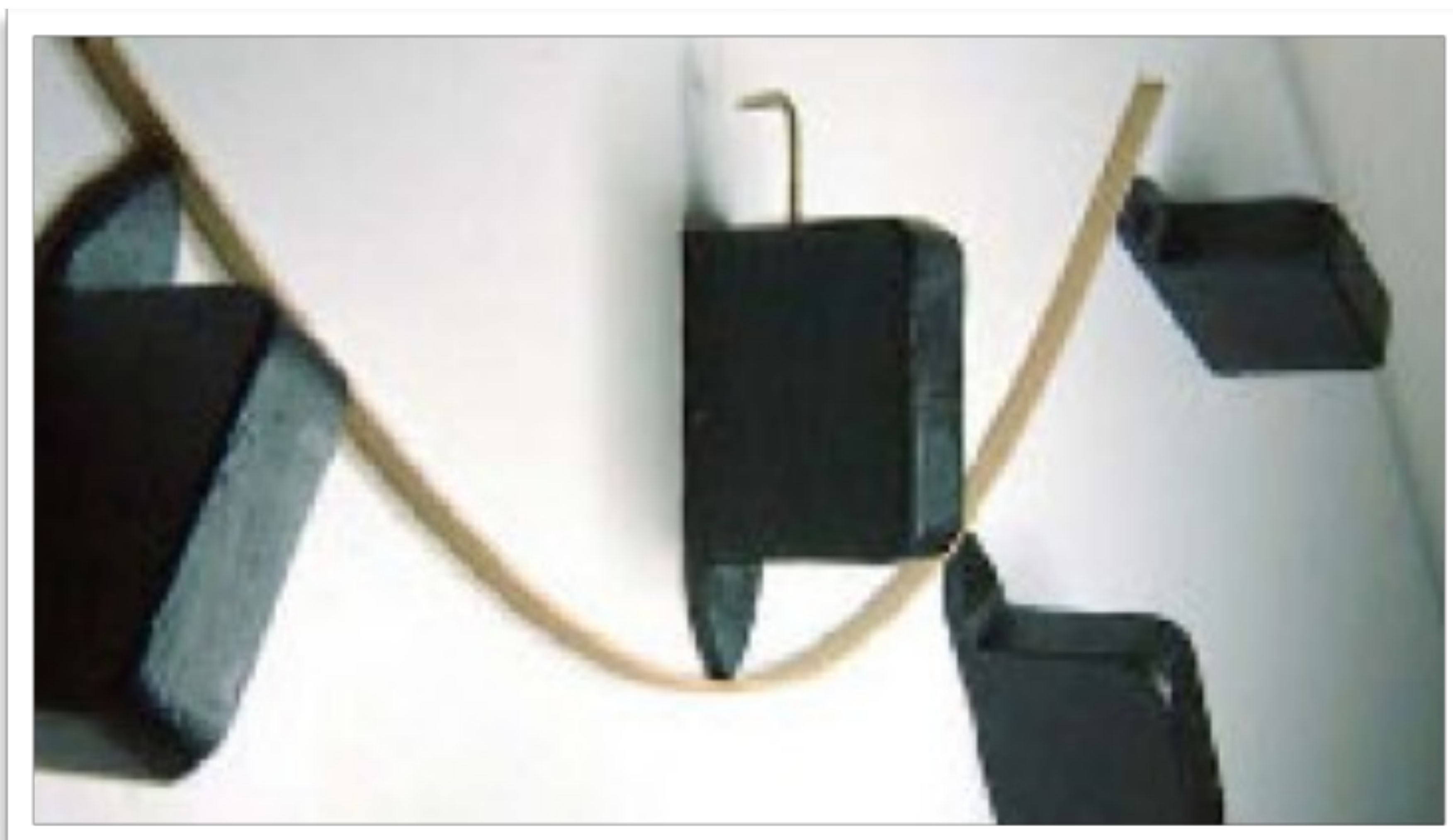
$$(ds)^2=(dx)^2+(dy)^2+(dz)^2=((x')^2+(y')^2+(z')^2)d^2u$$

- Arc length: $s=\int_{u_0}^{u} ds=\int_{u_0}^{u}\sqrt{(x')^2+(y')^2+(z')^2}\,du$

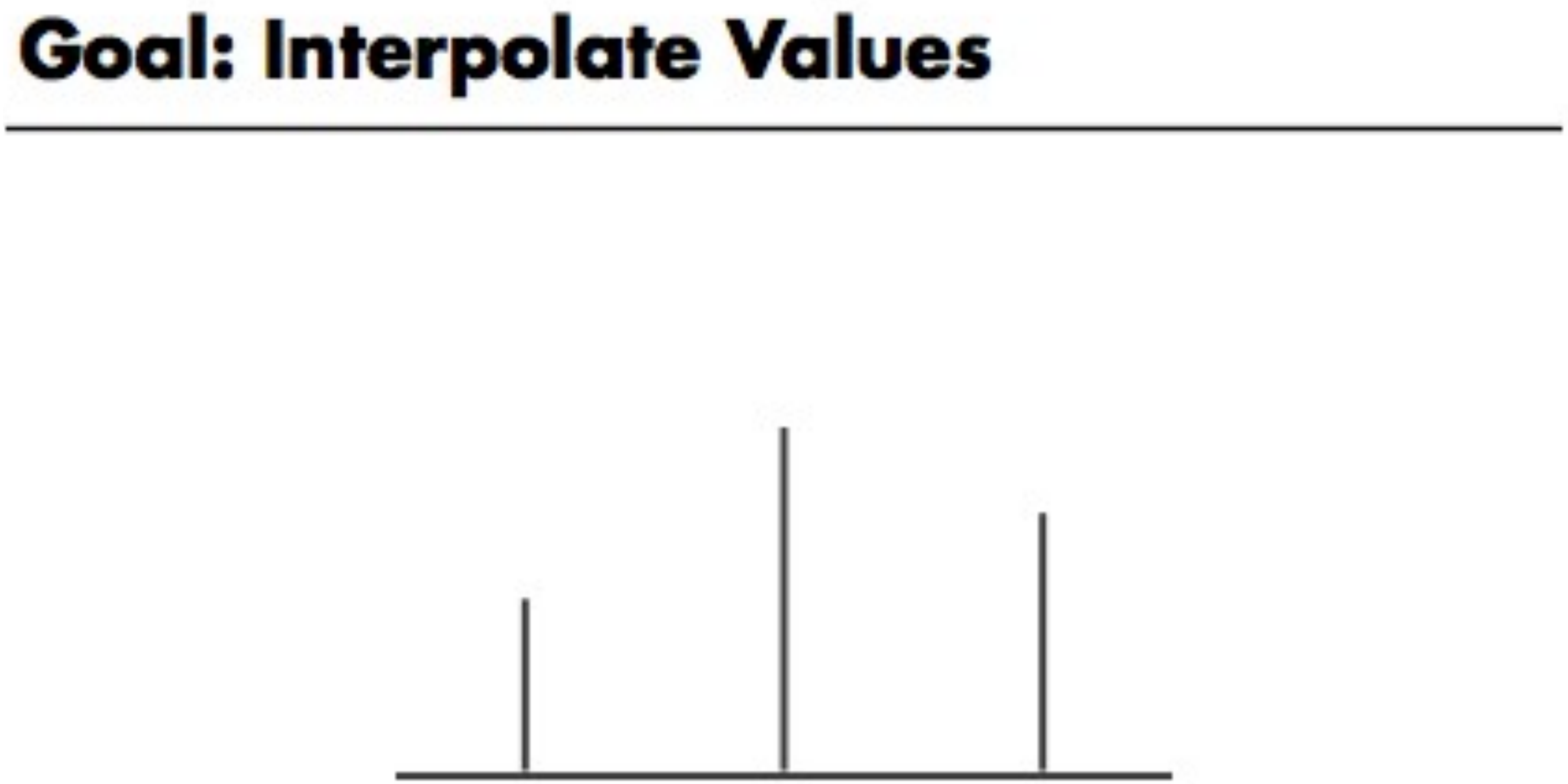


Parametric curves and splines

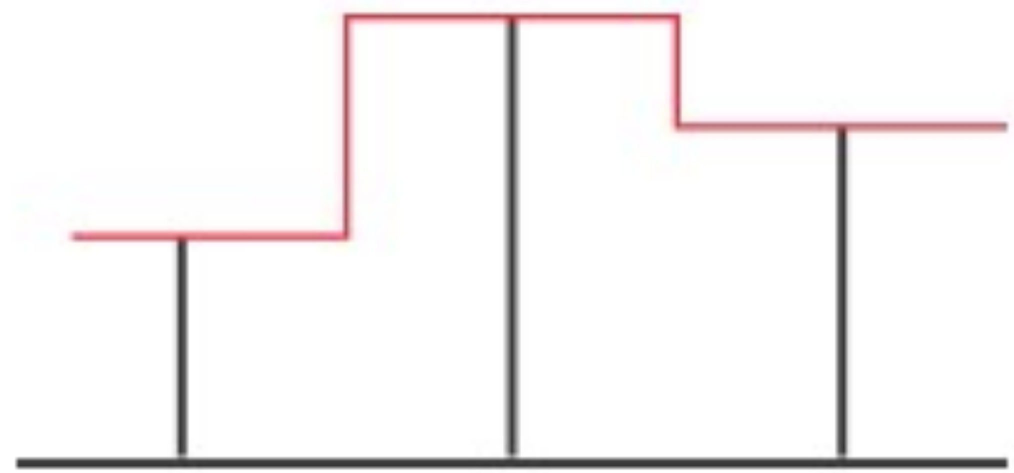
- Cubic Hermite interpolation
- Catmull-Rom interpolation
- Bezier curves



Cubic Hermite interpolation



Nearest Neighbor Interpolation

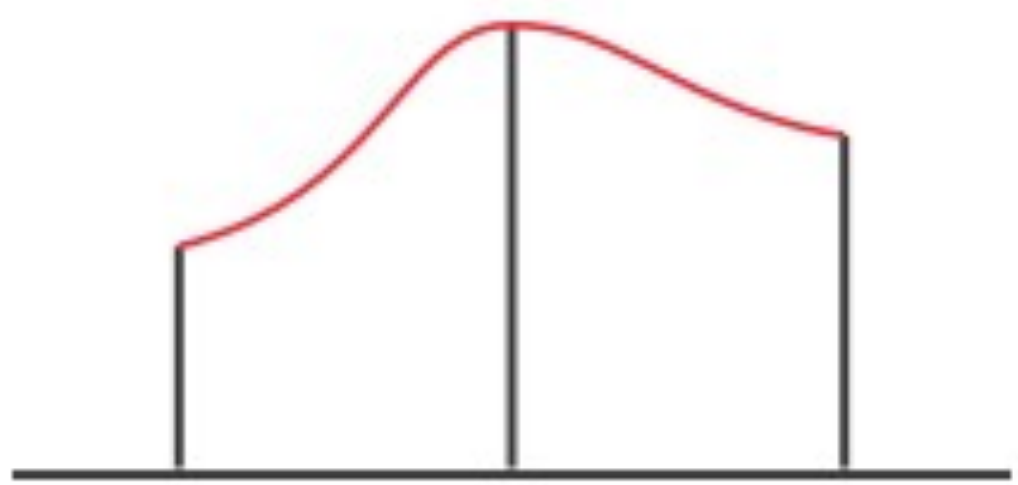


Problem: values not continuous

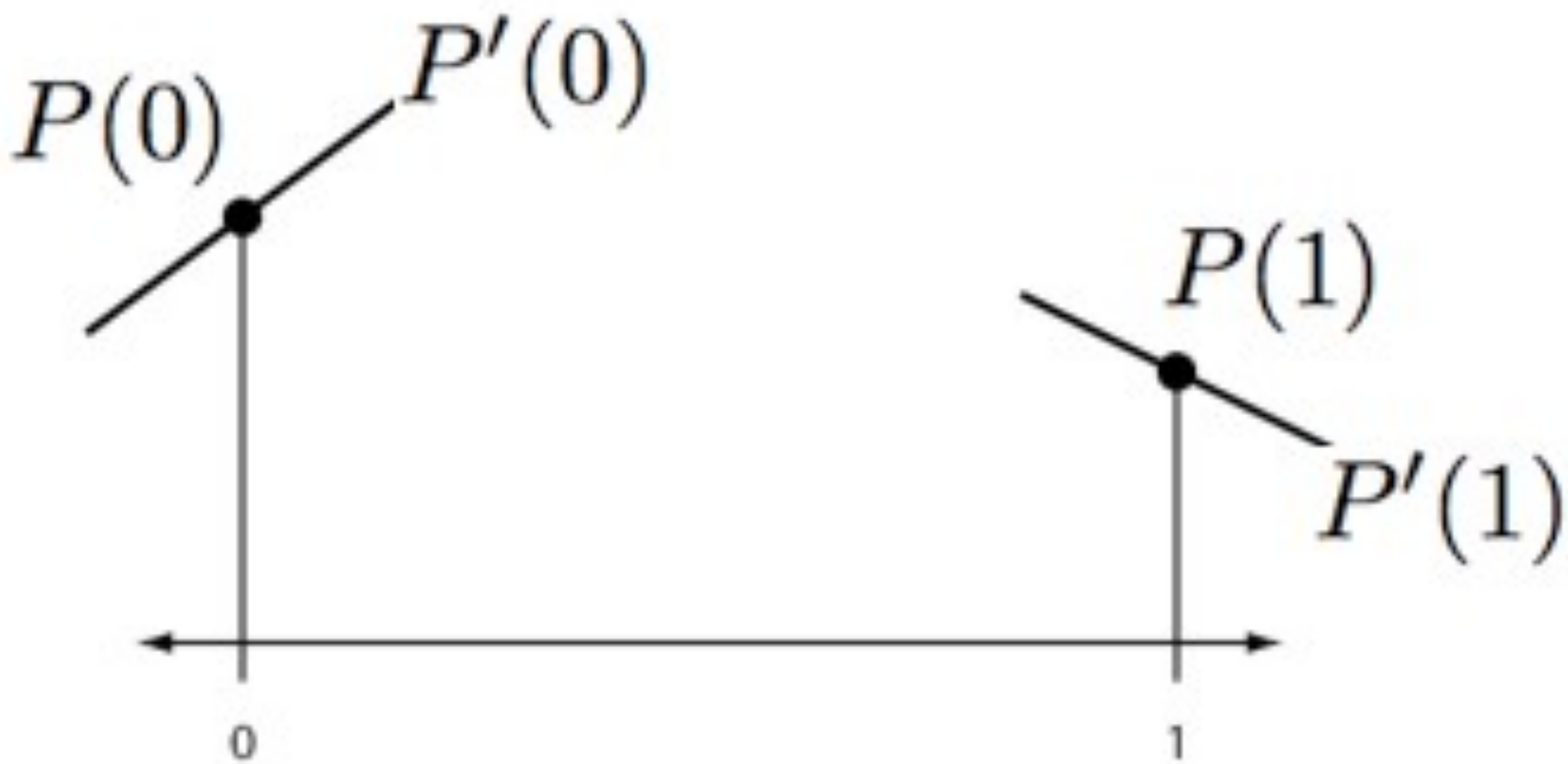
Linear Interpolation

Problem: derivatives not continuous

Smooth Interpolation?



Cubic Hermite Interpolation



Given: values and derivatives at 2 points

Cubic Polynomial Interpolation

Assume cubic polynomial

$$P(t) = a\,t^3 + b\,t^2 + c\,t + d$$

Why? 4 constraints => need 4 degrees of freedom

Cubic Hermite Interpolation

Assume cubic polynomial

$$P(t) = a\,t^3 + b\,t^2 + c\,t + d$$

$$P'(t) = 3a\,t^2 + 2b\,t + c$$

Solve for coefficients:

$$P(0) = h_0 = d$$

$$P(1) = h_1 = a + b + c + d$$

$$P'(0) = h_2 = c$$

$$P'(1) = h_3 = 3a + 2b + c$$

Matrix Representation

$$h_0 = d$$
$$h_1 = a + b + c + d$$
$$h_2 = c$$
$$h_3 = 3a + 2b + c$$

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Matrix Representation of Polynomials

$$P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Hermite Basis Functions

$$\begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}$$

$$P(t) = \sum_{i=0}^{3} h_i \, H_i(t)$$

Matrix Representation

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Solve for a, b, c, d

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix}$$

Inverse Matrix

Matrix Inverse

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

Change Basis

$$\begin{bmatrix} a & b & c & d \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}}_{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Change Basis

$$\underbrace{\begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}_{\begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix}} \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Matrix Transpose

Transpose $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$

$$\left(\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \right)^T = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Change Basis

$$\underbrace{\begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}_{\begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix}} \underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}}_{\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}}$$

Hermite Basis Functions

$$\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

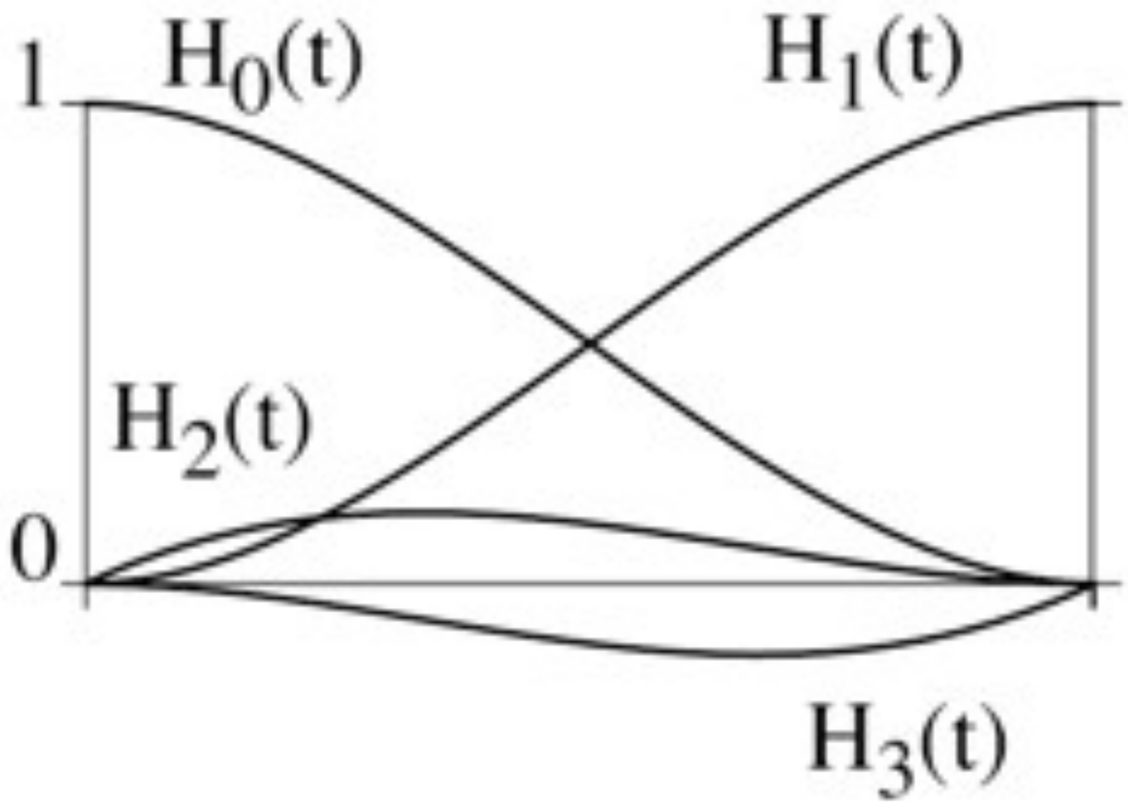
$$H_0(t) = 2t^3 - 3t^2 + 1$$

$$H_1(t) = -2t^3 + 3t^2$$

$$H_2(t) = t^3 - 2t^2 + t$$

$$H_3(t) = t^3 - t^2$$

Hermite Basis Functions



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

$$H_1(t) = -2t^3 + 3t^2$$

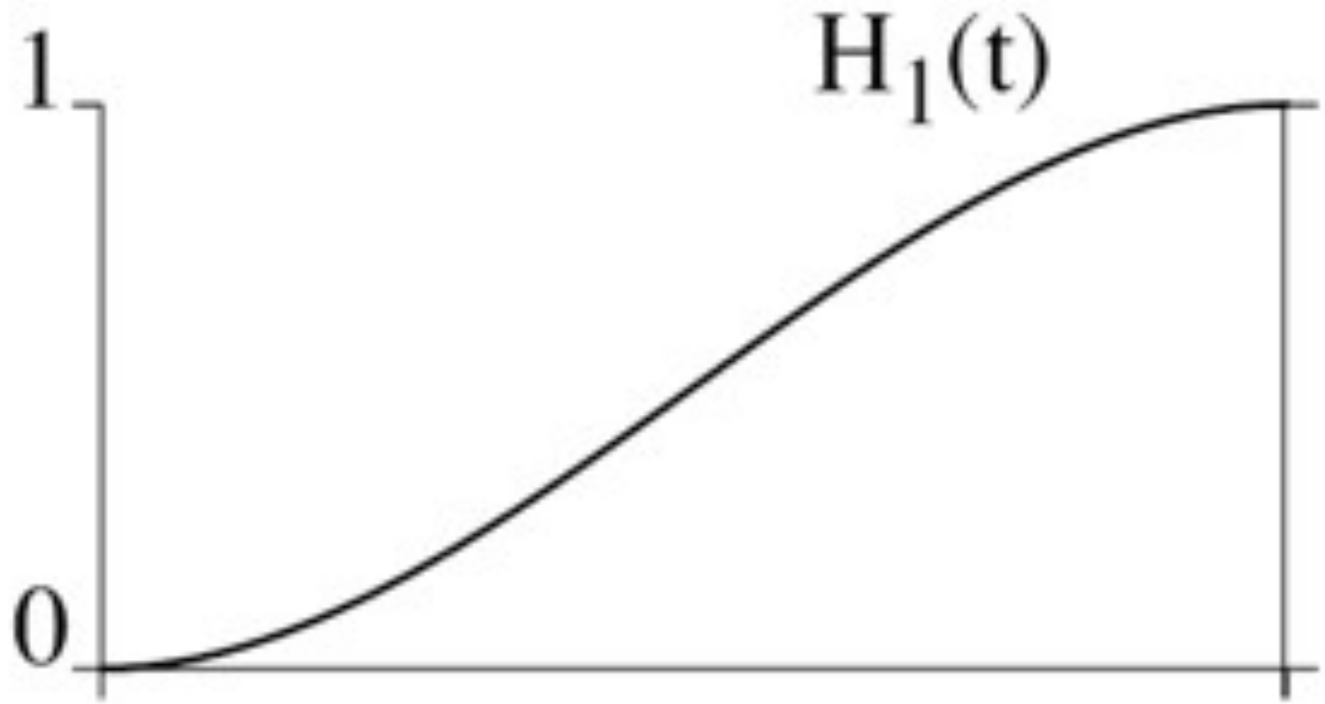
$$H_2(t) = t^3 - 2t^2 + t$$

$$H_3(t) = t^3 - t^2$$

Ease

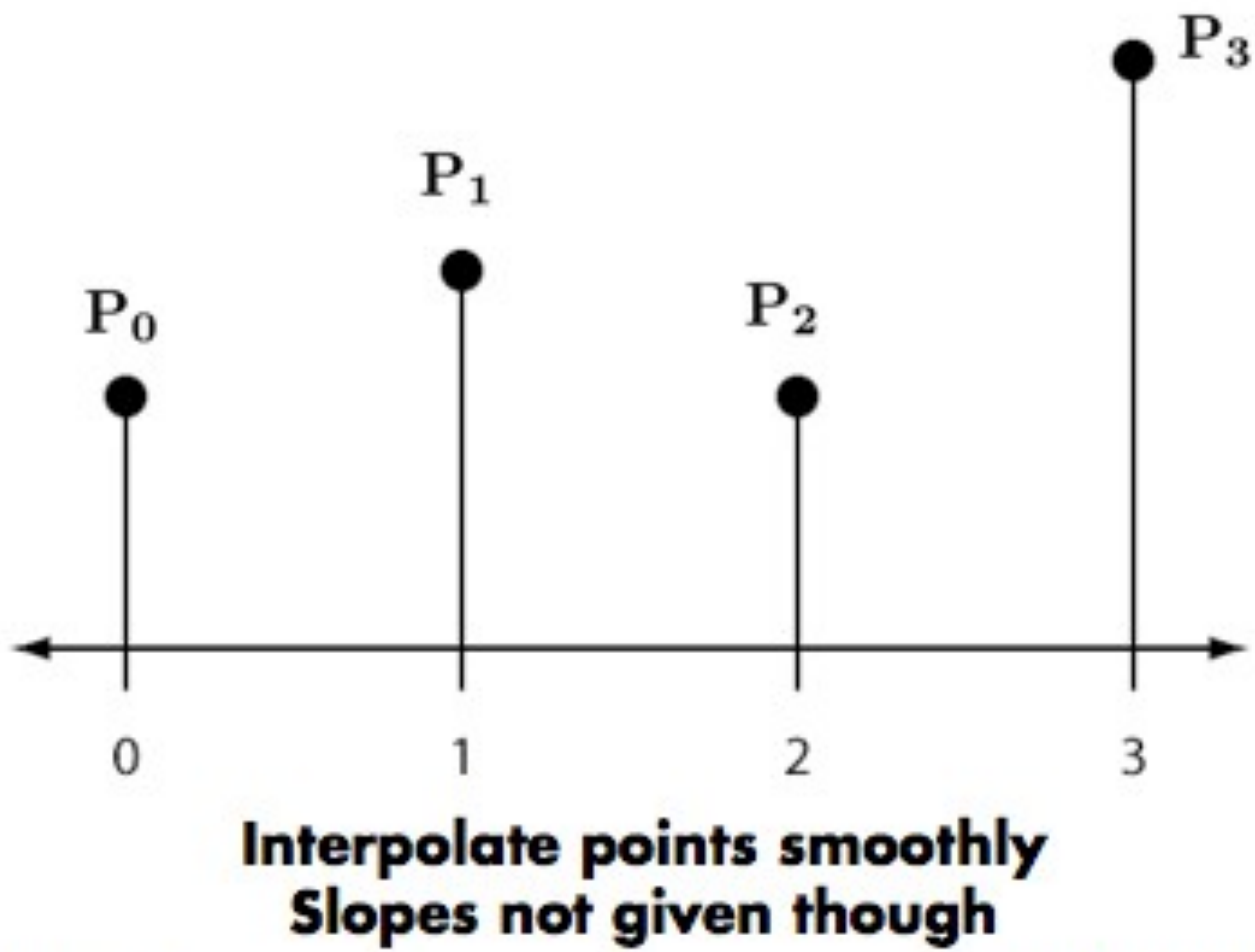
A very useful function

In animation, start and stop slowly (zero velocity)

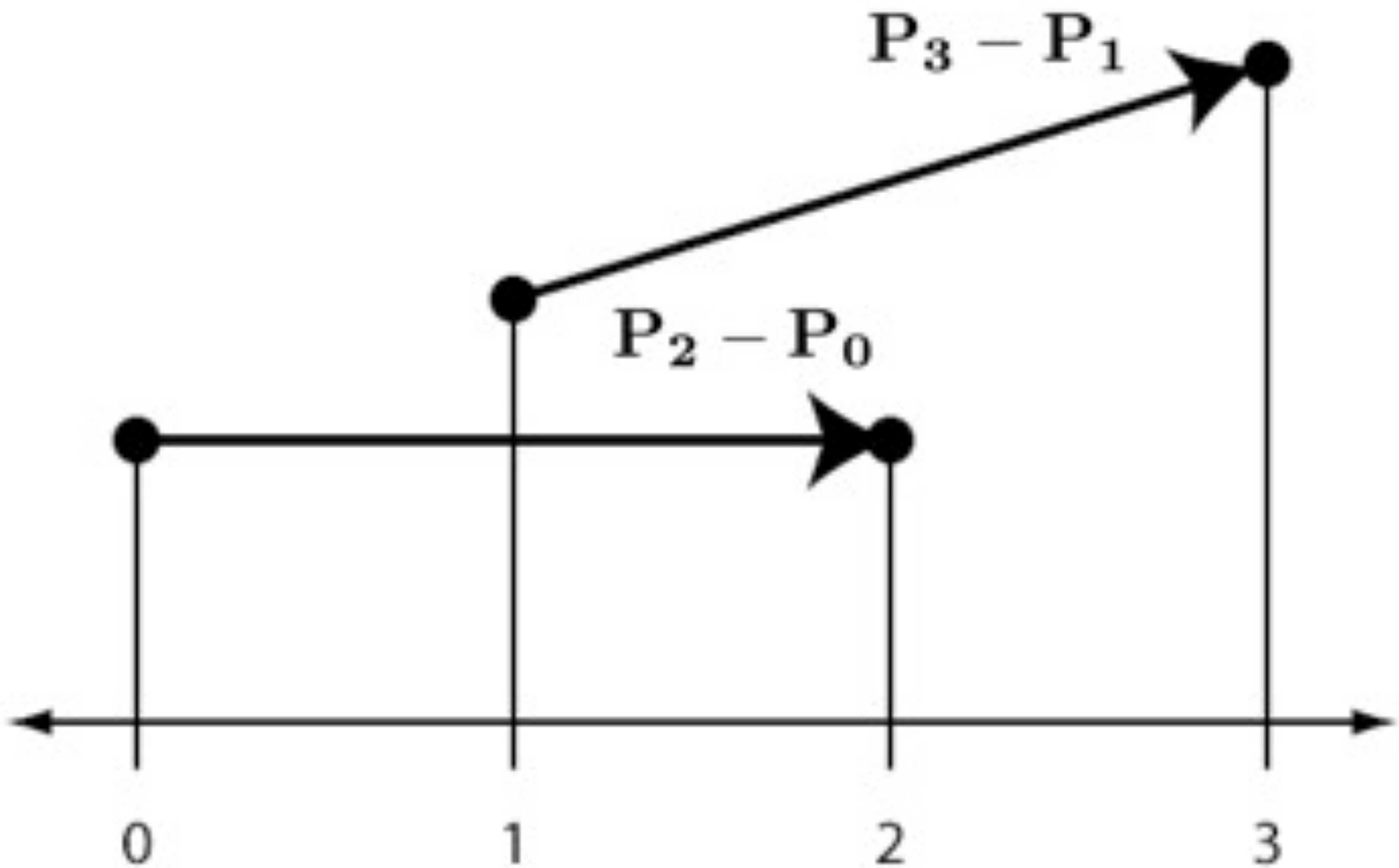


$$H_1(t) = -2t^3 + 3t^2 = t^2(3 - 2t)$$

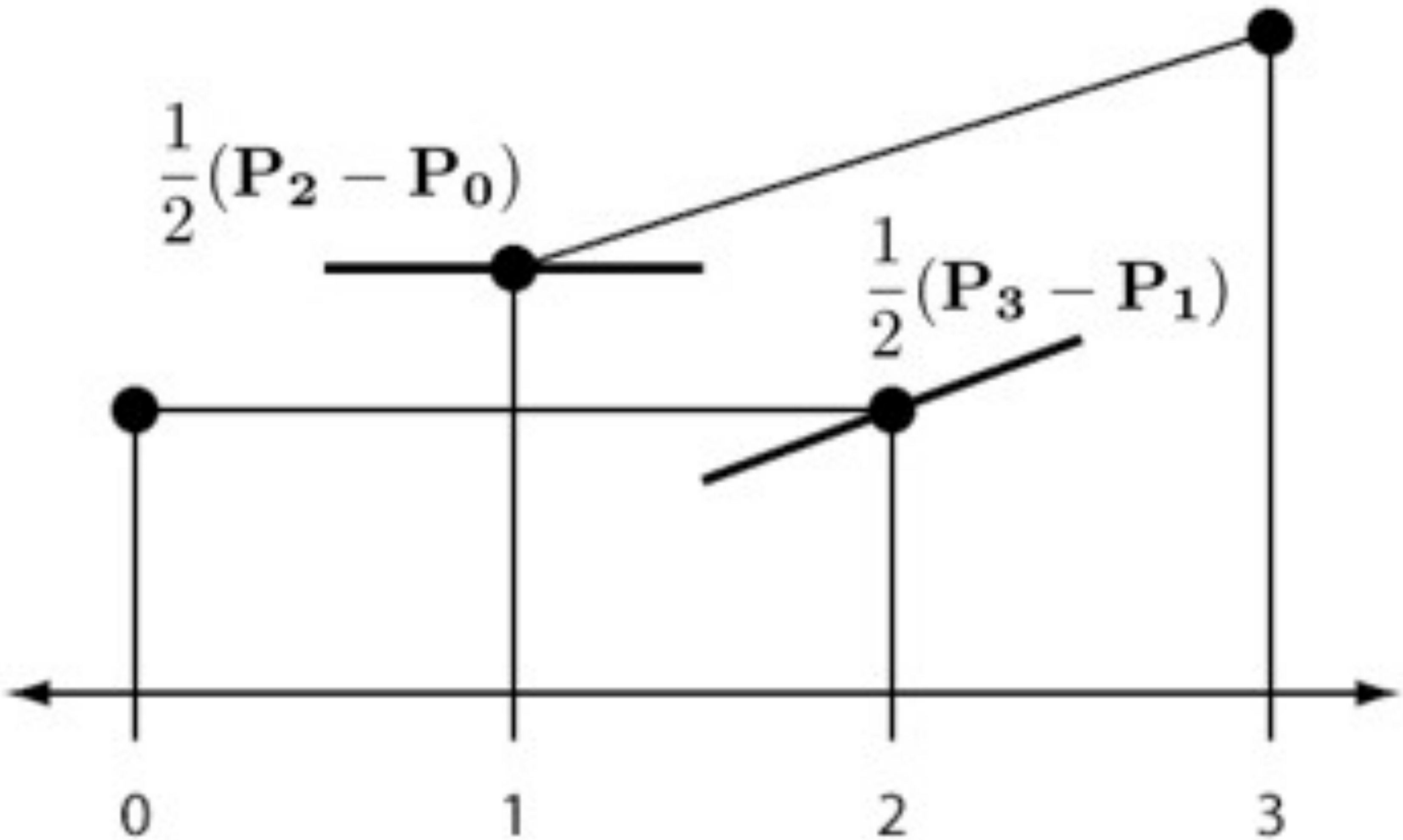
Catmull-Rom interpolation



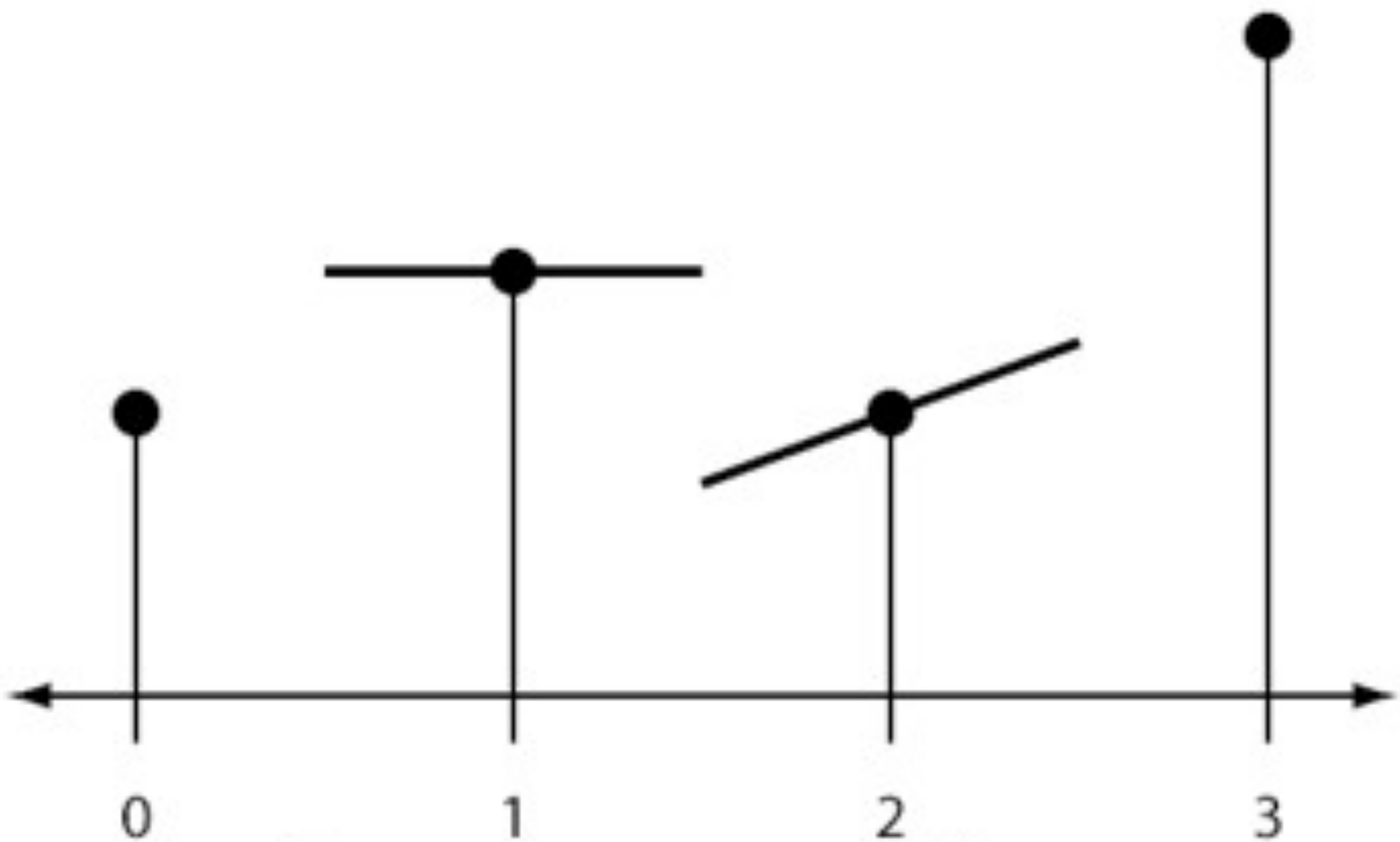
Catmull-Rom Interpolation



Catmull-Rom Interpolation

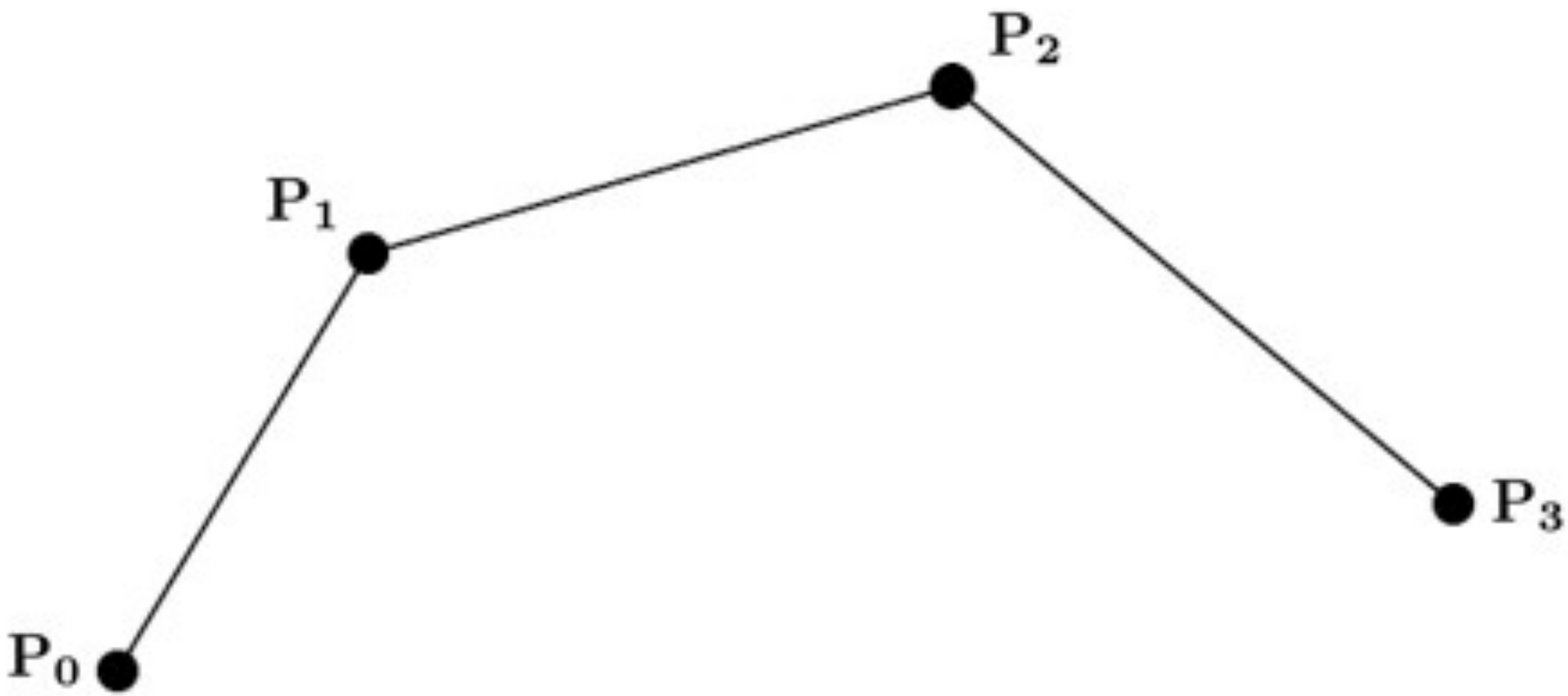


Catmull-Rom Interpolation



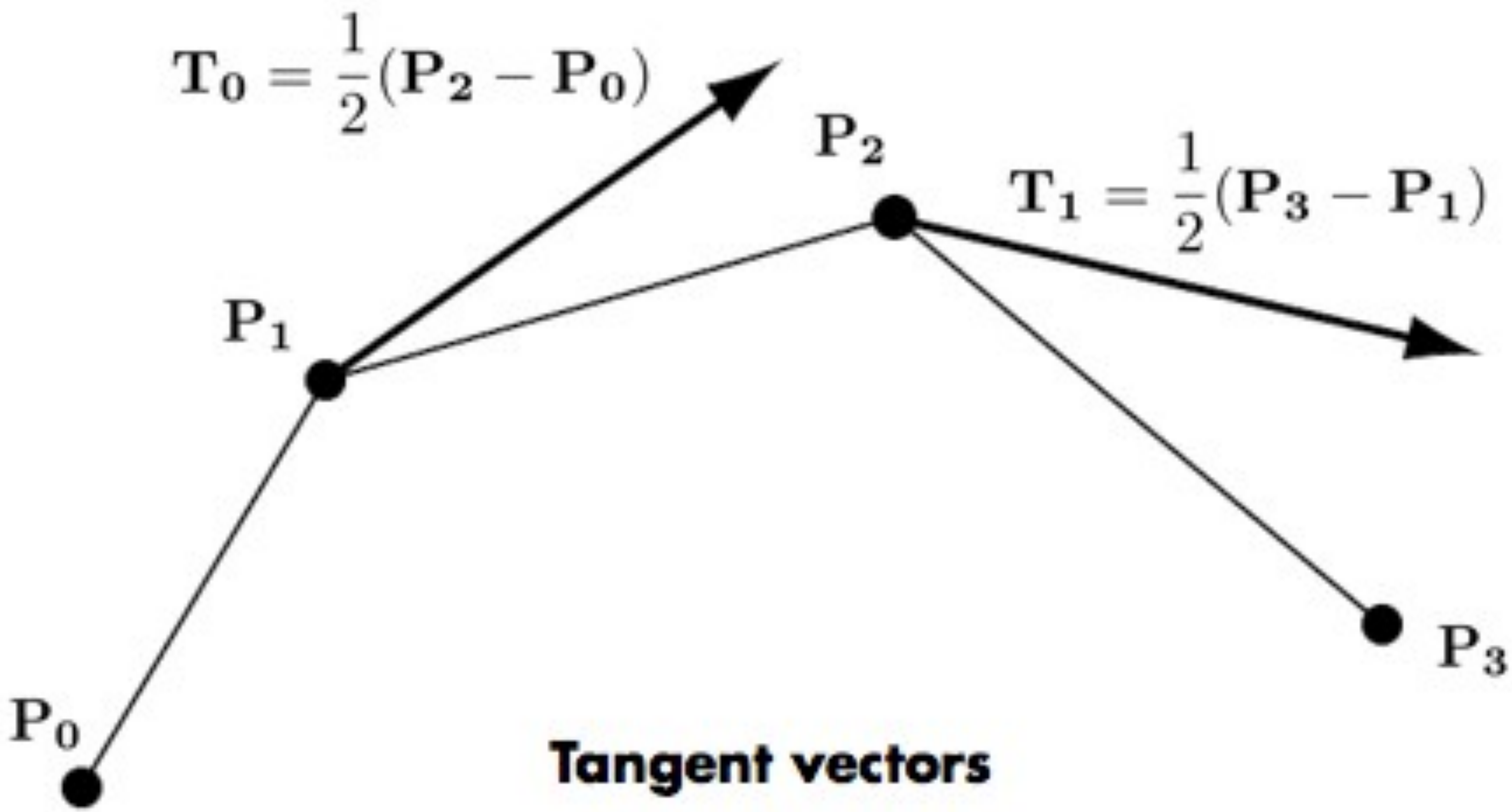
$$\mathbf{T_0} = \frac{1}{2}(\mathbf{P_2} - \mathbf{P_0}) \quad \mathbf{T_1} = \frac{1}{2}(\mathbf{P_3} - \mathbf{P_1})$$

Catmull-Rom Interpolation



We can interpolate points as easily as values

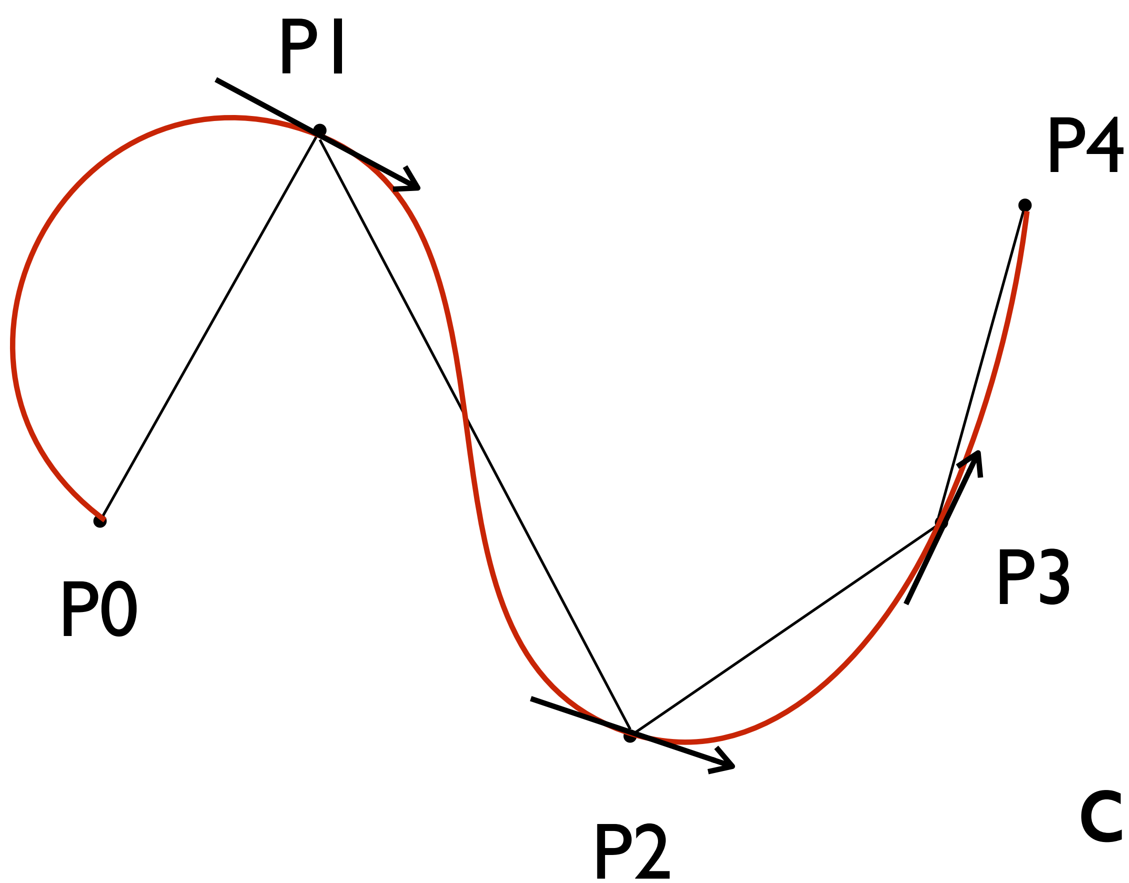
Catmull-Rom Interpolation



Tangent vectors

$$\boldsymbol{p}(t) = (2t^3 - 3t^2 + 1)\boldsymbol{p}_0 + (t^3 - 2t^2 + t)\boldsymbol{m}_0 + (-2t^3 + 3t^2)\boldsymbol{p}_1 + (t^3 - t^2)\boldsymbol{m}_1$$

How to use c-r curve?



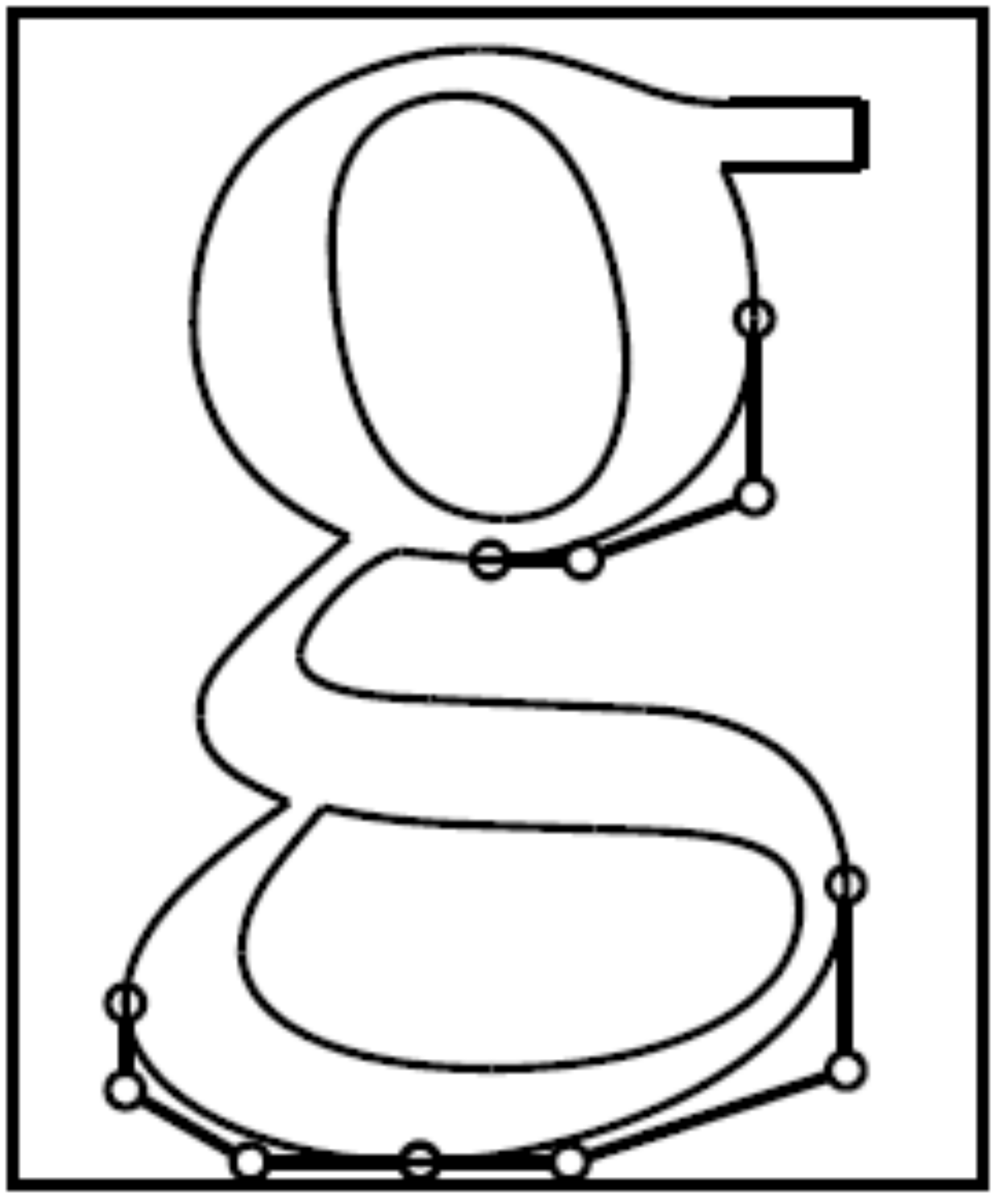
N control points
yield
N-1 curve segments

How to choose tangent
condition at two end points?

Video ^_^

- http://v.youku.com/v_show/id_XNTgyNjMwMjM2.html
- 计算机中的数学（2）－参变量函数

Bézier curve



Pierre Étienne Bézier
an engineer at Renault

Bézier curve

Bézier curve

$$C(t) = \sum_{i=0}^{n} P_i B_{i,n}(t), \quad t \in [0,1]$$

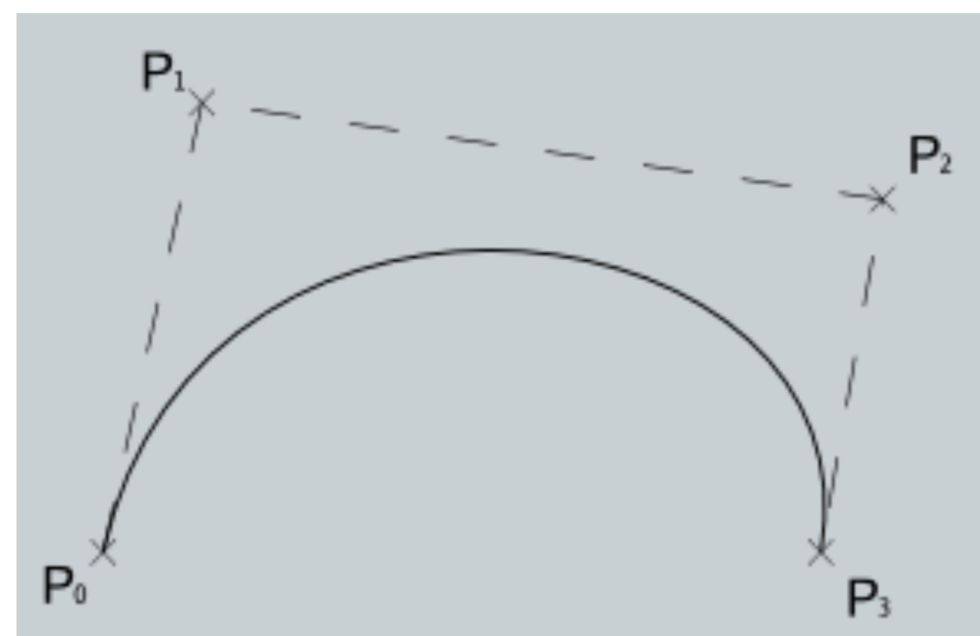
where, P_i (i=0,1,…,n) are control points.

$$\boxed{B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}, t \in [0,1]}$$

Bernstein basis

$$\begin{cases} X(t) = \sum_{i=0}^{n} x_i B_{i,t}(t) \\ Y(t) = \sum_{i=0}^{n} y_i B_{i,t}(t) \end{cases}$$

$$C(t) = \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix}, \quad P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

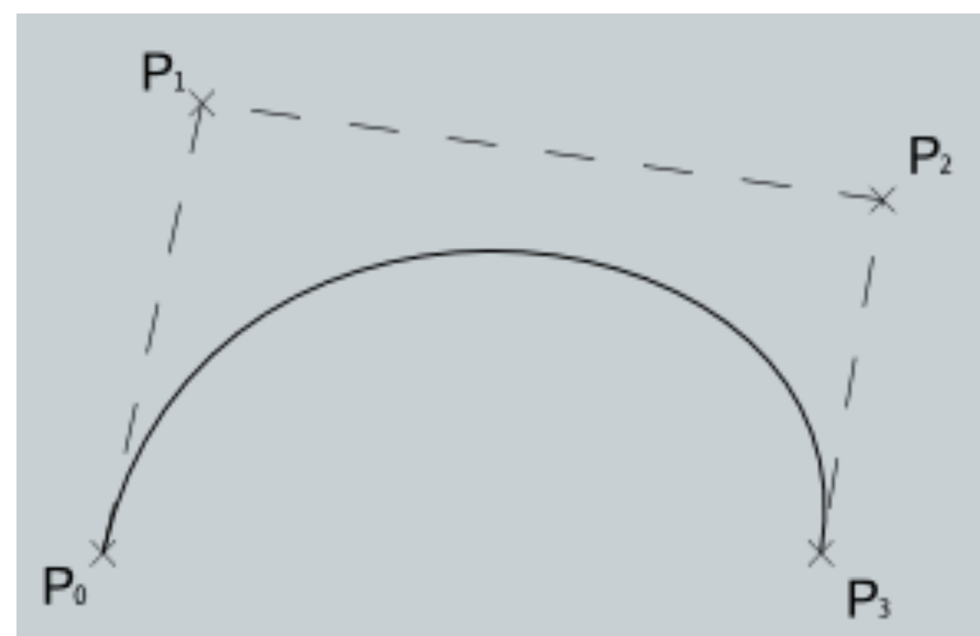


Bézier curve

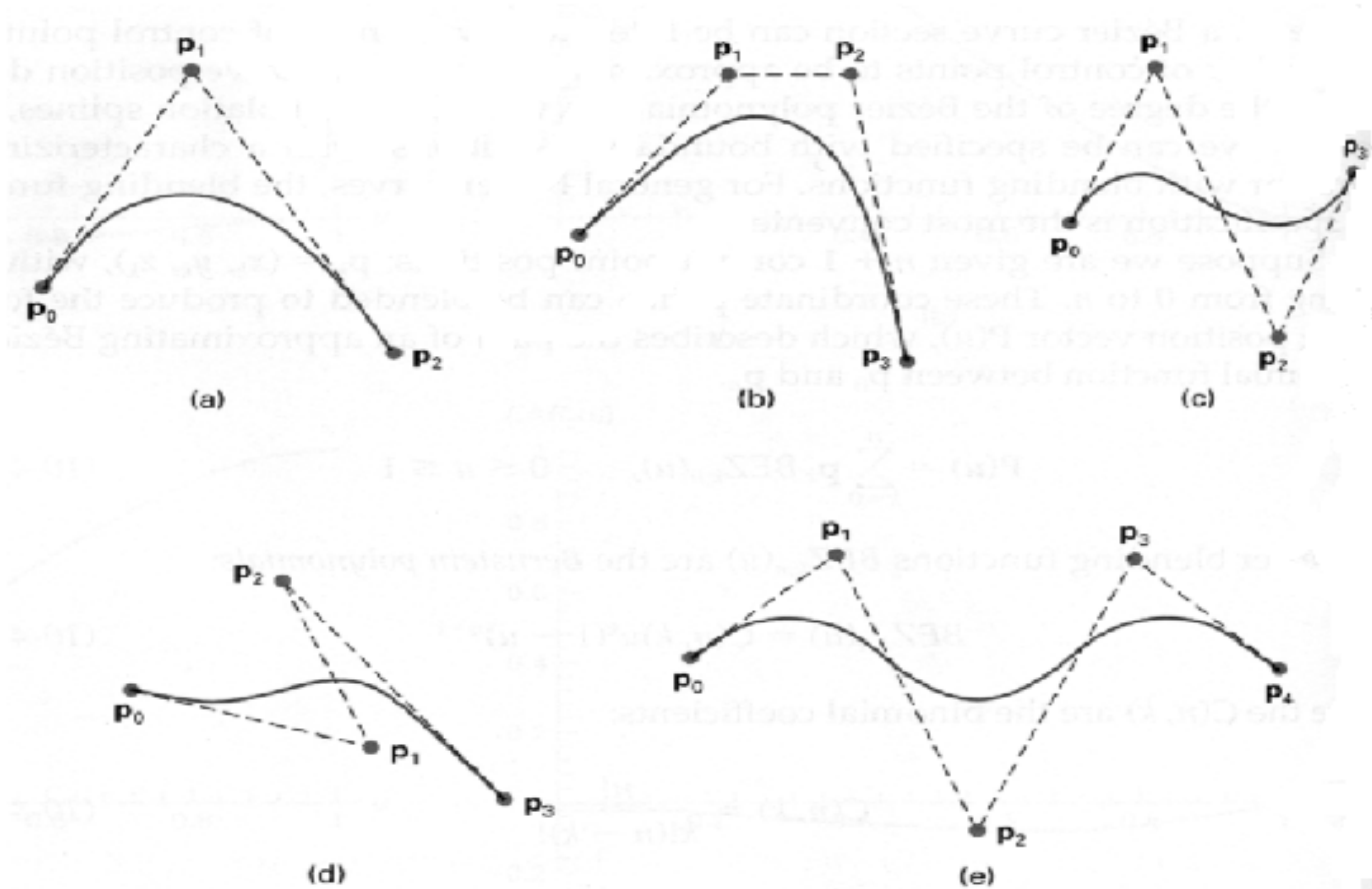
$$\begin{cases} X(t) = \sum_{i=0}^{n} x_i B_{i,t}(t) \\ Y(t) = \sum_{i=0}^{n} y_i B_{i,t}(t) \end{cases} \qquad \begin{cases} X(t) = \sum_{i=0}^{n} a_i t^i \\ Y(t) = \sum_{i=0}^{n} b_i t^i \end{cases}$$

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}, t \in [0,1]$$

$$C(t) = \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix}, \quad P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$



Bézier curve

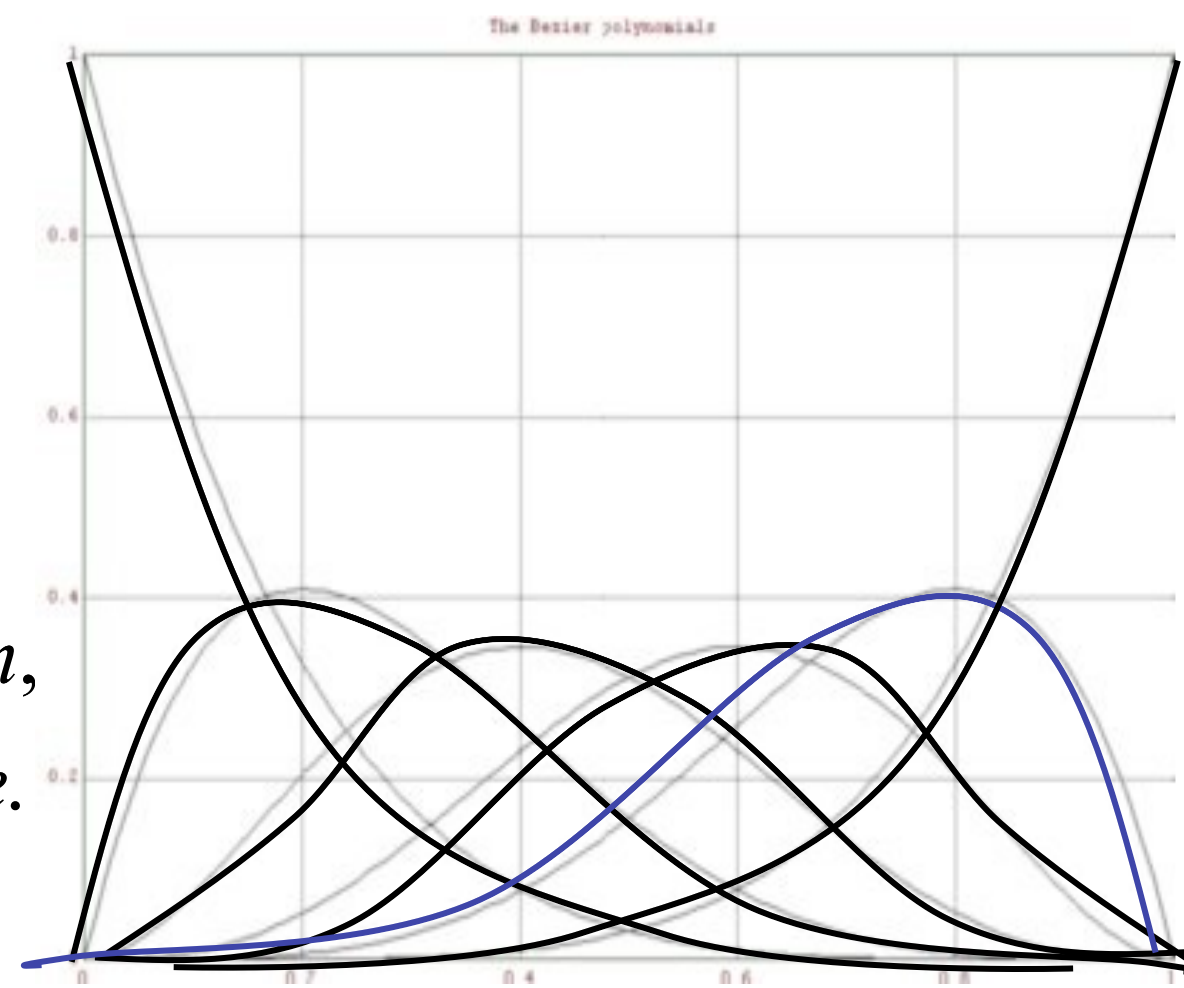


Bézier curve

Properties of Bernstein basis

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}, t \in [0,1]$$

1. $$B_{i,n}(t) \geq 0,\ i = 0,1,\cdots,n,\ t \in [0,1].$$
2. $$\sum_{i=0}^{n} B_{i,n}(t) = 1,\ t \in [0,1].$$
3. $$B_{i,n}(t) = B_{n-i,n}(1-t),$$ $$i = 0,1,\cdots,n,\ t \in [0,1].$$
4. $$B_{i,n}(0) = \begin{cases} 1, & i = 0, \\ 0, & else; \end{cases} \quad B_{i,n}(1) = \begin{cases} 1, & i = n, \\ 0, & else. \end{cases}$$



Bézier curve

Properties of Bernstein basis

5. $$B_{i,n}(t)=(1-t)B_{i,n-1}(t)+tB_{i-1,n-1}(t),\ i=0,1,\ldots,n.$$

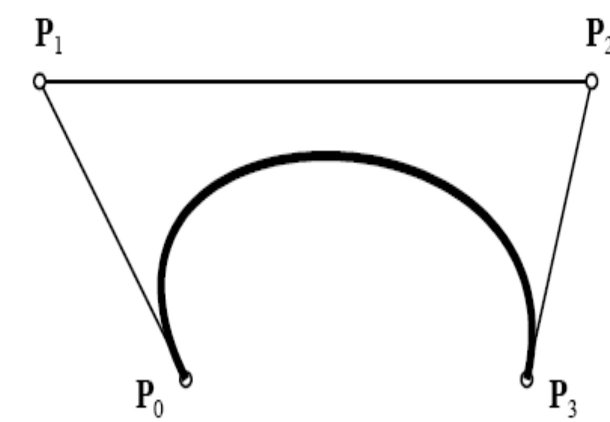
6. $$B'_{i,n}(t)=n[B_{i-1,n-1}(t)-B_{i,n-1}(t)],\ i=0,1,\ldots,n.$$

7. $$(1-t)B_{i,n}(t)=(1-\frac{i}{n+1})B_{i,n+1}(t);$$
$$tB_{i,n}(t)=\frac{i+1}{n+1}B_{i+1,n+1}(t);$$
$$B_{i,n}(t)=(1-\frac{i}{n+1})B_{i,n+1}(t)+\frac{i+1}{n+1}B_{i+1,n+1}(t).$$

Bézier curve

properties of Bézier curves

$$C(t)=\sum_{i=0}^{n} P_i B_{i,n}(t), \quad t\in[0,1]$$



1. **Endpoint Interpolation**: interpolating two end points

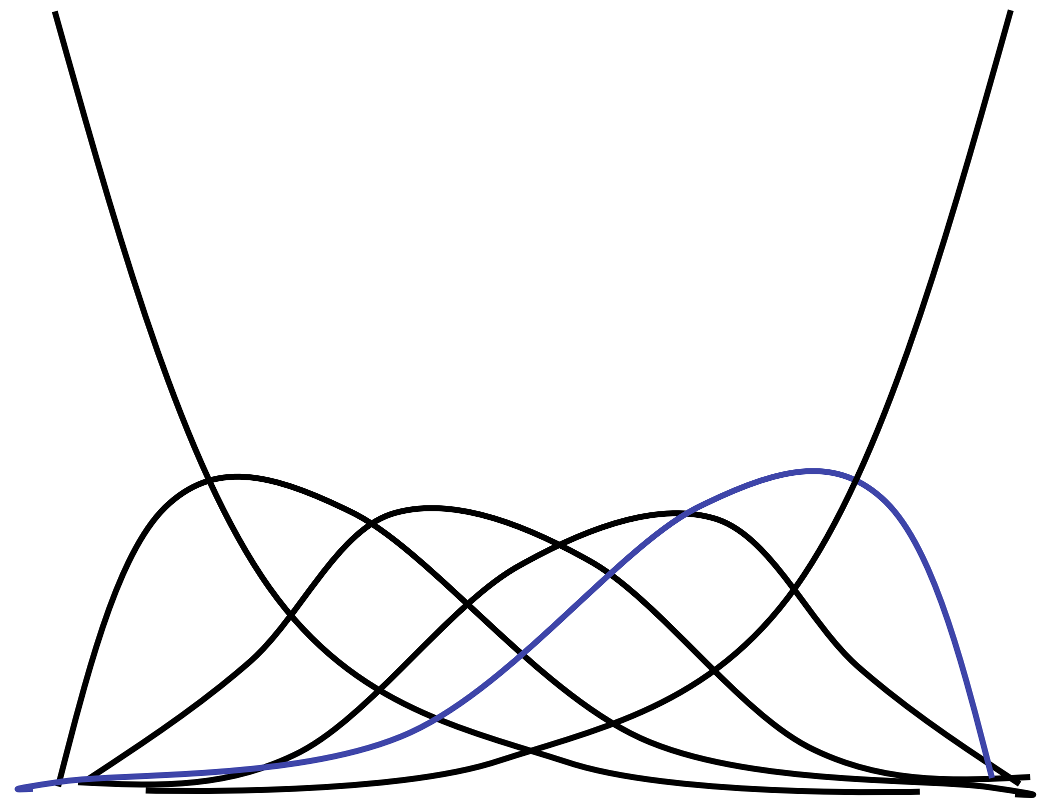
$$C(0)=P_0,\ C(1)=P_n.$$

2. **tangent direction** of P_0: P_0P_1, tangent direction of P_n : $P_{n-1}P_n$.

$$C'(t)=n\sum_{i=0}^{n-1}(P_{i+1}-P_i)B_{i,n-1}(t),\ t\in[0,1];\ \ C'(0)=n(P_1-P_0),C'(1)=n(P_n-P_{n-1}).$$

3. **Symmetry:** Let two Bezier curves be generated by ordered Bezier (control) points labelled by {p0,p1,...,pn} and {pn, pn-1,..., p0} respectively, then the curves corresponding to the two different orderings of control points look the same; they differ only in the direction in which they are traversed.

Bézier curve

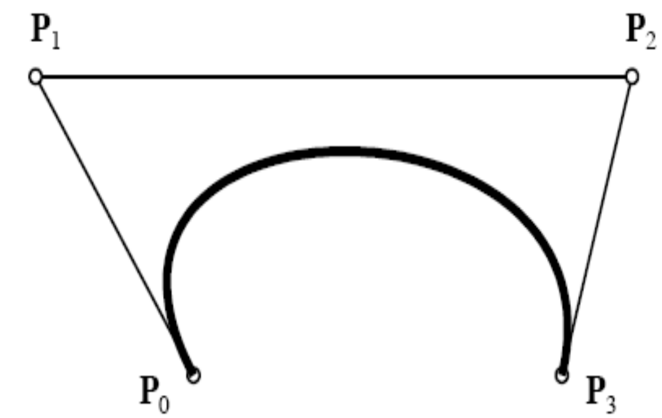


3. **Symmetry:** Let two Bezier curves be generated by ordered Bezier (control) points labelled by {p0,p1,...,pn} and {pn, pn-1,..., p0} respectively, then the curves corresponding to the two different orderings of control points look the same; they differ only in the direction in which they are traversed.

Bézier curve

properties of Bézier curves

$$C(t)=\sum_{i=0}^{n} P_i B_{i,n}(t), \quad t\in[0,1]$$



4. Affine Invariance –

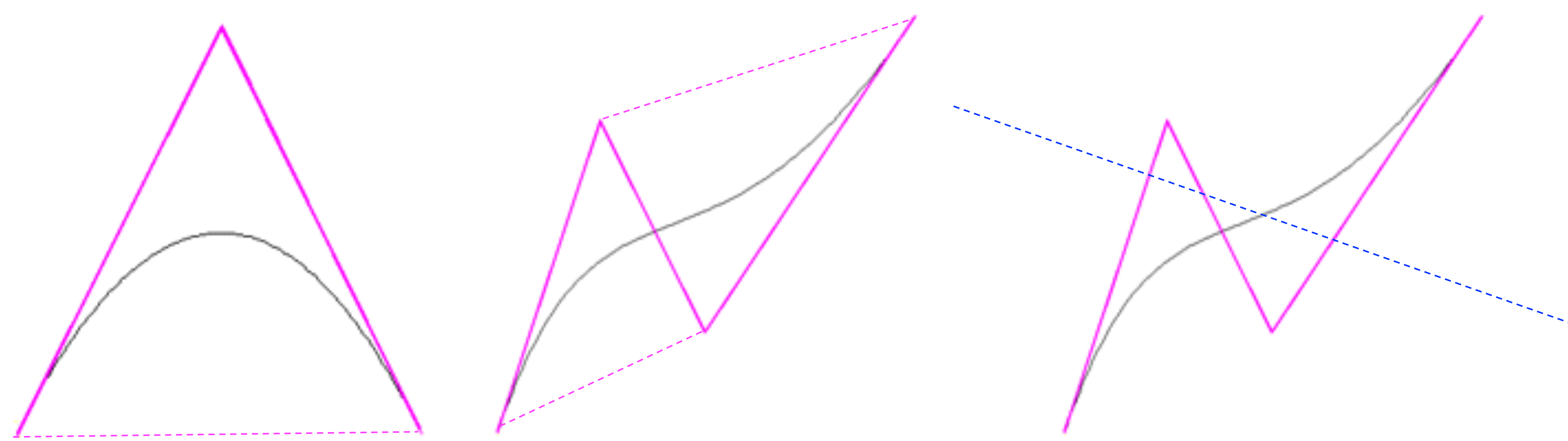
the following two procedures yield the same result:
(1) first, from starting control points {p0, p1,..., pn} compute the curve and then apply an affine map to it;
(2) first apply an affine map to the control points {p0, p1,..., pn} to obtain new control points {F(p0),...,F(pn)} and then find the curve with these new control points.

Bézier curve

properties of Bézier curves

5. **Convex Hull Property :** Bézier curve $\boldsymbol{C}(t)$ lies in the convex hull of the control points $\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$;

6. ***variation diminishing* property.** Informally this means that the Bezier curve will not "wiggle" any more than the control polygon does..

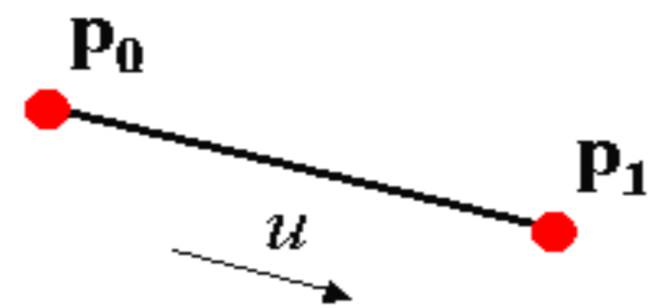


Bézier curve

Bézier curves

1. linear: $\boldsymbol{C}(t) = (1\text{-}t)\boldsymbol{P}_0 + t\boldsymbol{P}_1,\ t \in [0,1]$,

$$\boldsymbol{C}(t) = [t, 1]\begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{P}_0 \\ \boldsymbol{P}_1 \end{bmatrix}$$



2. quadratic

$$\boldsymbol{C}(t) = (1-t)^2\boldsymbol{P}_0 + 2t(1-t)\boldsymbol{P}_1 + t^2\boldsymbol{P}_2$$

Degree 2

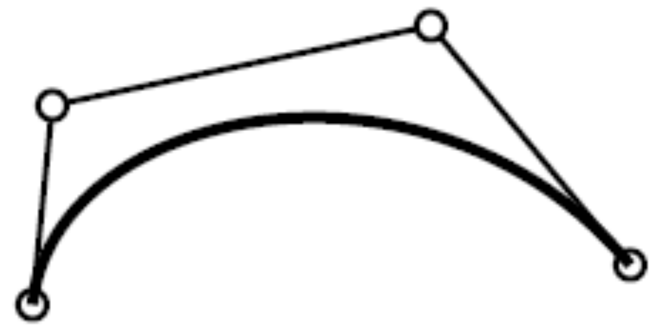
$$C(t) = [t^2 \quad t \quad 1]\begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix}$$

Bézier curve

3. cubic:

$$C(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3$$

$$C(t) = [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$



Degree 3

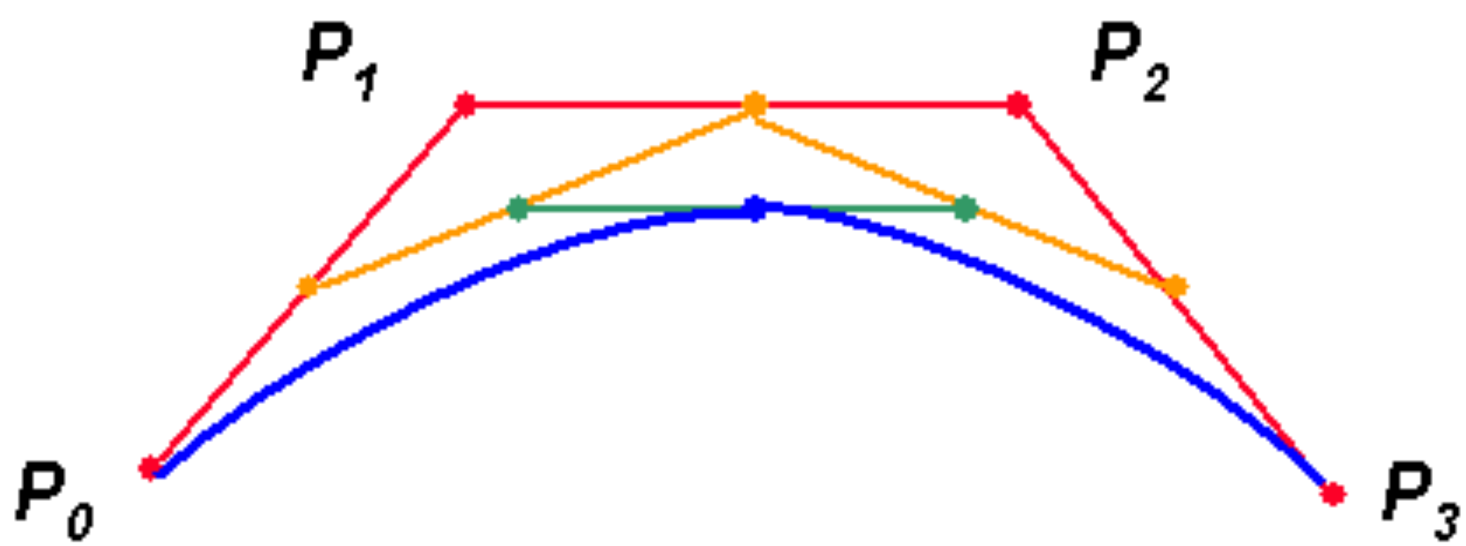
Bézier curve

De Casteljau algorithm

given the control points $\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$, and t of Bézier curve, let:

$$\boldsymbol{P}_i^r(t) = (1-t)\boldsymbol{P}_i^{r-1}(t) + t\boldsymbol{P}_{i+1}^{r-1}(t), \text{ Æä } \begin{cases} r = 1,\ldots,n;\ i = 0,\ldots,n-r \\ P_i^0(u) = P_i \end{cases}$$

then $\boldsymbol{P}_0^n(t) = \mathrm{C}(t)$.



Bézier curve

Rational Bézier Curve

$$\boldsymbol{R}(t)=\frac{\sum_{i=0}^{n} B_{i,n}(t)\omega_i \boldsymbol{P}_i}{\sum_{i=0}^{n} B_{i,n}(t)\omega_i}=\sum_{i=0}^{n} R_{i,n}(t)\boldsymbol{P}_i$$

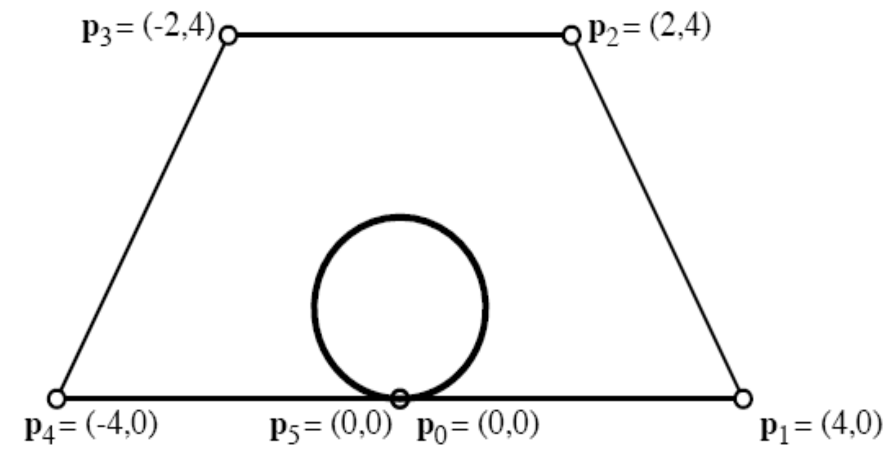


Figure 2.19: Circle as Degree 5 Rational Bézier Curve.

where $B_{i,n}(t)$ is Bernstein basis, ω_i is the weight at p_i .

It's a generalization of Bézier curve, which can express more curves, such as circle.

Bézier curve

Properties of rational Bézier curve:

1. endpoints: $\boldsymbol{R}(0) = \boldsymbol{P}_0;\ \boldsymbol{R}(1) = \boldsymbol{P}_n$

2. tangent of endpoints:

$$\boldsymbol{R}'(0) = n\frac{\omega_1}{\omega_0}(\boldsymbol{P}_1 - \boldsymbol{P}_0);\ \boldsymbol{R}'(1) = n\frac{\omega_{n-1}}{\omega_n}(\boldsymbol{P}_n - \boldsymbol{P}_{n-1})$$

3. Convex Hull Property

......

5.

6. Influence of the weights

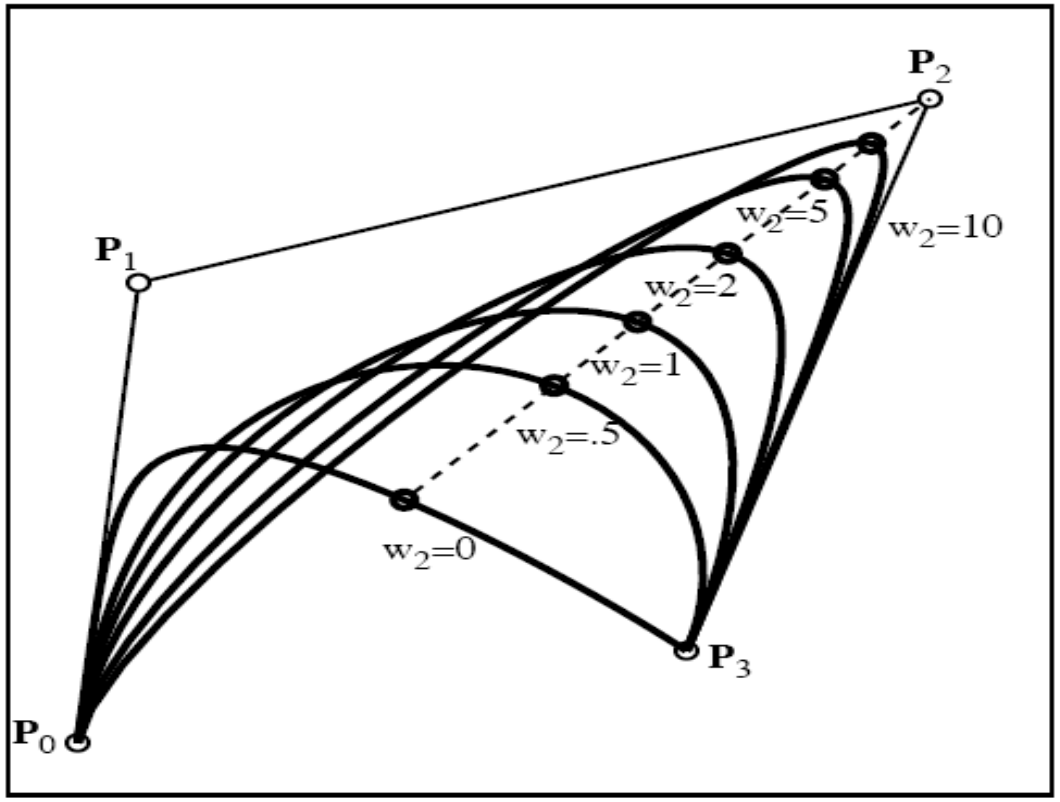


Figure 2.16: Rational Bézier curve.

Bézier surface

Bézier surface

Bézier surface:

$$\boldsymbol{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \boldsymbol{P}_{ij} B_{i,n}(u) B_{j,m}(v), \qquad 0 \le u,v \le 1$$

where $B_{i,n}(u)$和$B_{j,m}(v)$ Bernstein basis with n degree and m degree, respectively, $(n+1)\times(m+1)$ $\boldsymbol{P}_{i,j}(i=0,1,\ldots,n; j=0,1,\ldots,m)$ construct the control meshes.