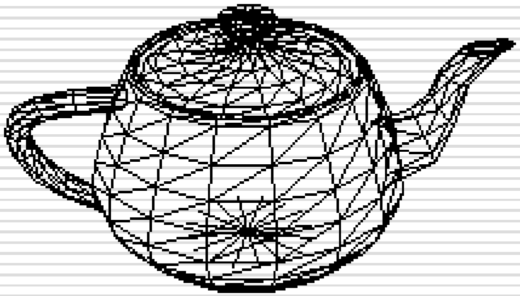
Computer Graphics

Bing-Yu Chen
National Taiwan University

Curves and Surfaces

- Mathematical Curve Representation
- Parametric Cubic Curves
- Parametric Bi-Cubic Surfaces

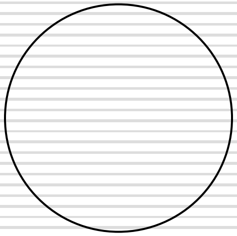
The Utah Teapot



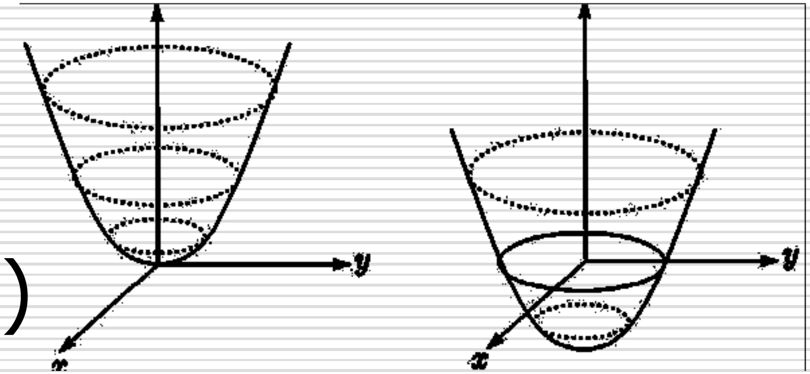
http://en.wikipedia.org/wiki/Utah_teapot
http://www.sjbaker.org/teapot/

Mathematical Curve Representation

- Explicit y=f(x)
  - what if the curve is not a function, e.g., a circle?



- Implicit g(x,y)=0



- Parametric (x(u),y(u))
  - For the circle:

Recall: Plane Equation

- $Ax + By + Cz + D = 0$
  - and (A, B, C) means the normal vector
  - so, given points P_1, P_2, and P_3 on the plane
  - $(A, B, C) = P_1P_2 \times P_1P_3$
  - what happened if $(A, B, C) = (0,0,0)$?
  - the distance from a vertex (x, y, z) to the plane is

$$d = \frac{Ax + By + Cz + D}{\sqrt{A^2 + B^2 + C^2}}$$

Parametric Polynomial Curves

- We will use parametric curves where the functions are all polynomials in the parameter.

$$x(u) = \sum_{k=0}^{n} a_k u^k$$

$$y(u) = \sum_{k=0}^{n} b_k u^k$$

- Advantages:
  - easy (and efficient) to compute
  - infinitely differentiable

Parametric Cubic Curves

- Fix $n=3$
- The cubic polynomials that define a curve segment $Q(t)=[x(t) \quad y(t) \quad z(t)]^{\mathrm{T}}$ are of the form

$$x(t)=a_x t^3+b_x t^2+c_x t+d_x,$$
$$y(t)=a_y t^3+b_y t^2+c_y t+d_y,$$
$$z(t)=a_z t^3+b_z t^2+c_z t+d_z, \quad 0\le t\le 1.$$

Parametric Cubic Curves

- The curve segment can be rewrite as

$$Q(t) = [x(t) \quad y(t) \quad z(t)]^{\mathrm{T}} = C \bullet T$$

- where $T = [t^3 \quad t^2 \quad t \quad 1]^{\mathrm{T}}$

$$C = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix}$$

Tangent Vector

$$\frac{d}{dt}Q(t) = Q'(t) = \begin{bmatrix} \frac{d}{dt}x(t) & \frac{d}{dt}y(t) & \frac{d}{dt}z(t) \end{bmatrix}^{\mathrm{T}}$$

$$= \frac{d}{dt}C \bullet T = C \bullet \begin{bmatrix} 3t^2 & 2t & 1 & 0 \end{bmatrix}^{\mathrm{T}}$$

$$= \begin{bmatrix} 3a_x t^2 + 2b_x t + c_x & 3a_y t^2 + 2b_y t + c_y & 3a_z t^2 + 2b_z t + c_z \end{bmatrix}^{\mathrm{T}}$$

Three Types of Parametric Cubic Curves

- Hermite Curves
  - defined by two **endpoints** and two endpoint **tangent vectors**
- Bézier Curves
  - defined by two **endpoints** and two **control points** which control the endpoint' **tangent vectors**
- Splines
  - defined by four **control points**

Parametric Cubic Curves

- $Q(t) = C \bullet T$
- rewrite the coefficient matrix as $C = G \bullet M$
  - where M is a 4x4 **basis matrix,** G is called the **geometry matrix**
  - so

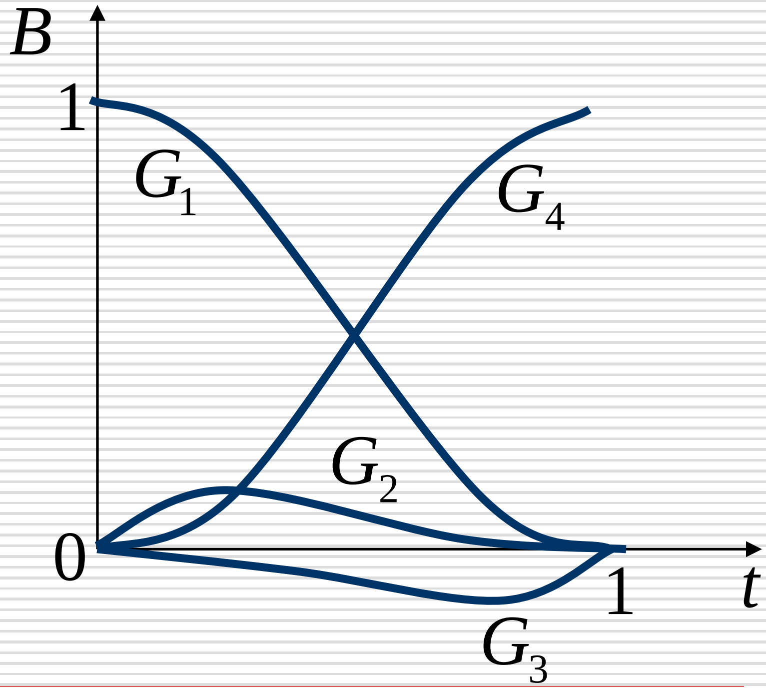
$$Q(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} G_1 & G_2 & G_3 & G_4 \end{bmatrix} \begin{bmatrix} m_{11} & m_{21} & m_{31} & m_{41} \\ m_{12} & m_{22} & m_{32} & m_{42} \\ m_{13} & m_{23} & m_{33} & m_{43} \\ m_{14} & m_{24} & m_{34} & m_{44} \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

4 endpoints or tangent vectors

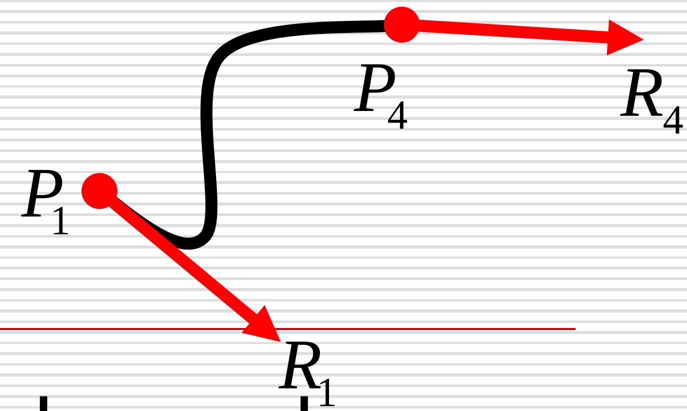
Parametric Cubic Curves

- $Q(t) = G \bullet M \bullet T = G \bullet B$

  where $B = M \bullet T$ is called the **blending functions**



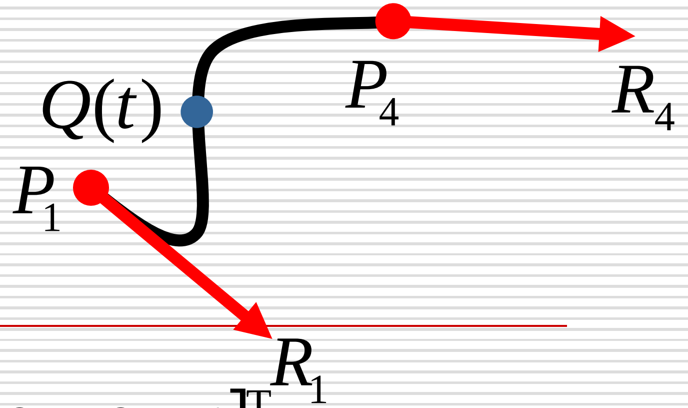
Hermite Curves



- Given the endpoints P_1 and P_4 and tangent vectors at them R_1 and R_4
- What is
  - **Hermite basis matrix** M_{H}
  - **Hermite geometry vector** G_{H}
  - **Hermite blending functions** B_{H}
- by definition

$$G_{\mathrm{H}} = \begin{bmatrix} P_1 & P_4 & R_1 & R_4 \end{bmatrix}$$

Hermite Curves



- since $Q(0) = P_1 = G_H \bullet M_H \bullet \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$

$$Q(1) = P_4 = G_H \bullet M_H \bullet \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$$

$$Q'(0) = R_1 = G_H \bullet M_H \bullet \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$Q'(1) = R_4 = G_H \bullet M_H \bullet \begin{bmatrix} 3 & 2 & 1 & 0 \end{bmatrix}^T$$

$$G_H = \begin{bmatrix} P_1 & P_4 & R_1 & R_4 \end{bmatrix} = G_H \bullet M_H \bullet \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Hermite Curves

- so

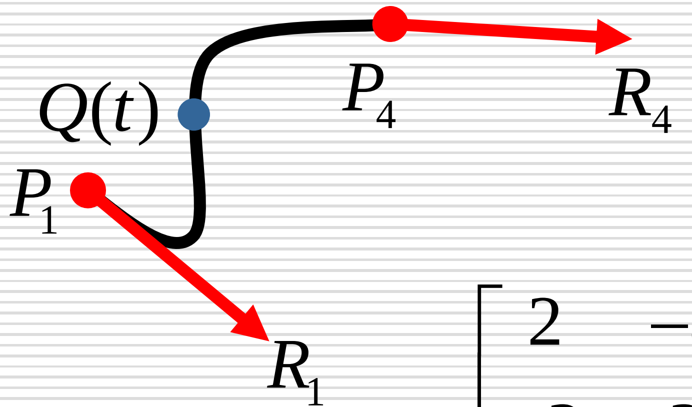
$$M_H = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

- and $Q(t) = G_H \bullet M_H \bullet T = G_H \bullet B_H$

$$B_H = \begin{bmatrix} 2t^3 - 3t^2 + 1 & -2t^3 + 3t^2 & t^3 - 2t^2 + t & t^3 - t^2 \end{bmatrix}^T$$

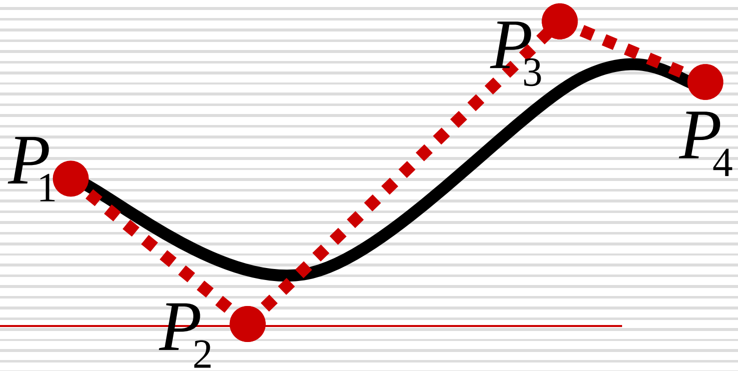
Computing a point

- Given two endpoints P_1 and P_4 and two tangent vectors at them R_1 and R_4



so

$$Q(t) = \begin{bmatrix} P_1 & P_4 & R_1 & R_4 \end{bmatrix} \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$



Bézier Curves

- Given the endpoints P_1 and P_4 and two control points P_2 and P_3 which determine the endpoints' tangent vectors, such that $R_1 = Q'(0) = 3(P_2 - P_1)$

$$R_4 = Q'(1) = 3(P_4 - P_3)$$

- What is
  - **Bézier basis matrix** M_B
  - **Bézier geometry vector** G_B
  - **Bézier blending functions** B_B

Bézier Curves

- by definition $G_B = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \end{bmatrix}$
- then $G_H = \begin{bmatrix} P_1 & P_4 & R_1 & R_4 \end{bmatrix}$

$$= \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{bmatrix} = G_B \bullet M_{HB}$$

- so $Q(t) = G_H \bullet M_H \bullet T = (G_B \bullet M_{HB}) \bullet M_H \bullet T$

$$= G_B \bullet (M_{HB} \bullet M_H) \bullet T = G_B \bullet M_B \bullet T$$

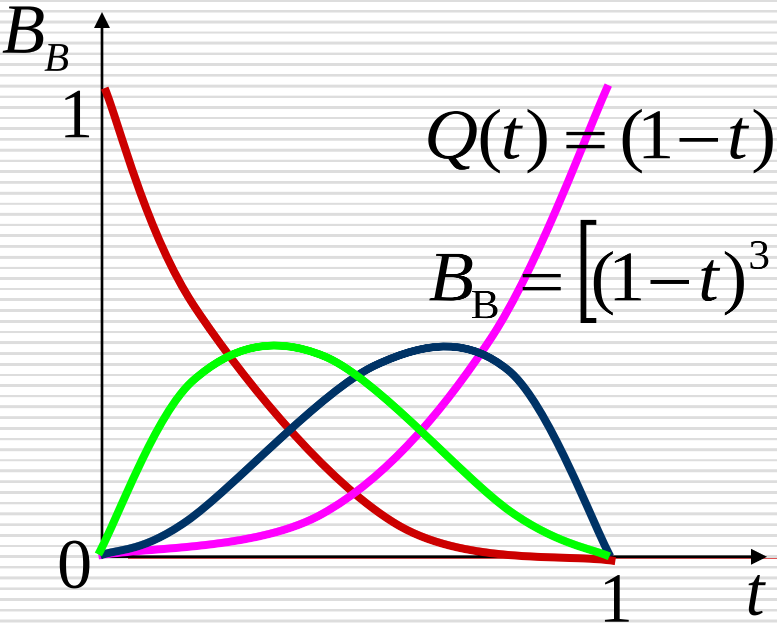
Bézier Curves

- and

$$M_B = M_{HB} \bullet M_H = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4$$

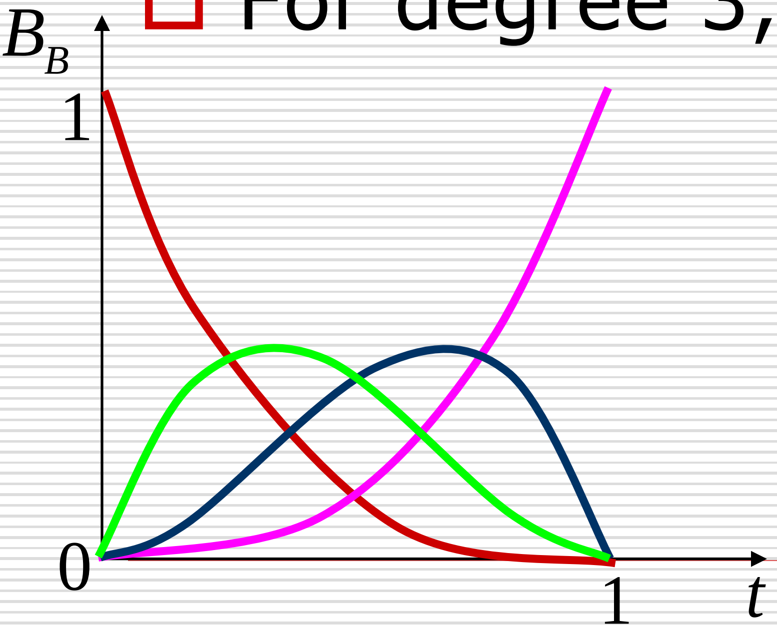
$$B_B = \left[(1-t)^3 \quad 3t(1-t)^2 \quad 3t^2(1-t) \quad t^3\right]^T$$



Bernstein polynomials

Bernstein Polynomials

- The coefficients of the control points are a set of functions called the **Bernstein polynomials**: $Q(t) = \sum_{i=0}^{n} b_i(t) P_i$
- For degree 3, we have:



$$b_0(t) = (1-t)^3$$

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

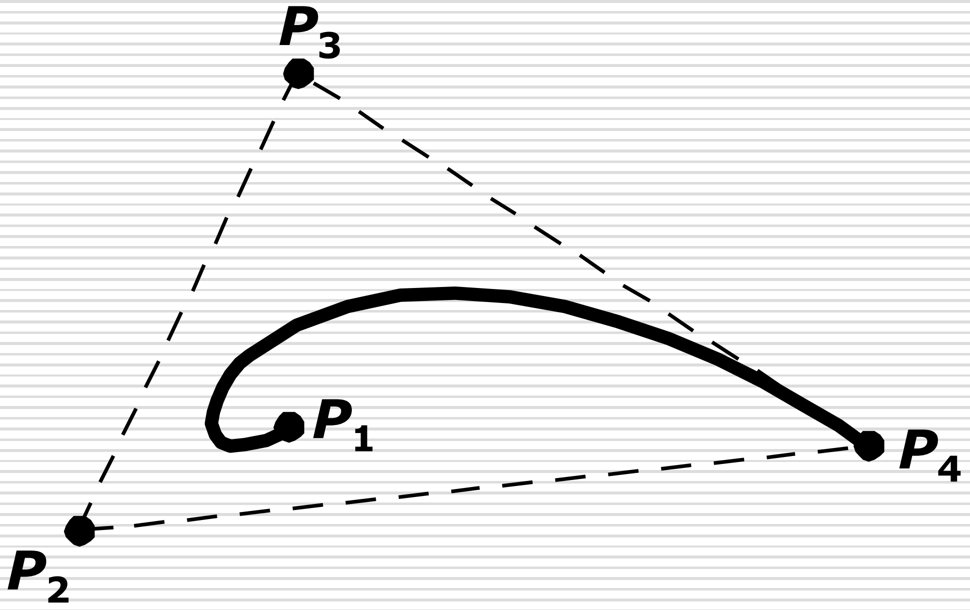
$$b_2(t) = 3t^2(1-t)$$

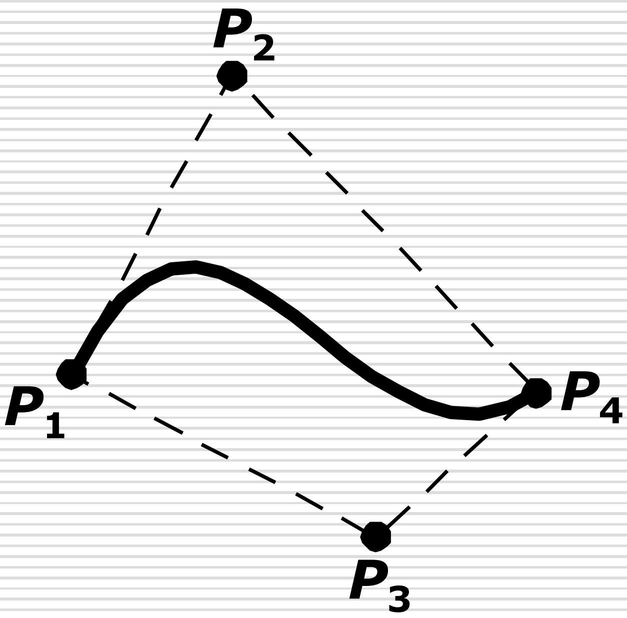
$$b_3(t) = t^3$$

Bernstein Polynomials

- Useful properties on the interval [0,1]:
  - each is between 0 and 1
  - sum of all four is exact 1
    - a.k.a., a "partition of unity"
- These together imply that the curve lines within the **convex hull** of its control points.

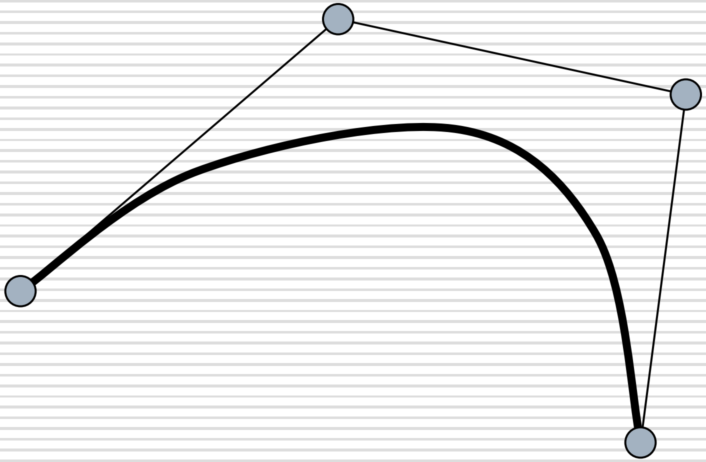
Convex Hull





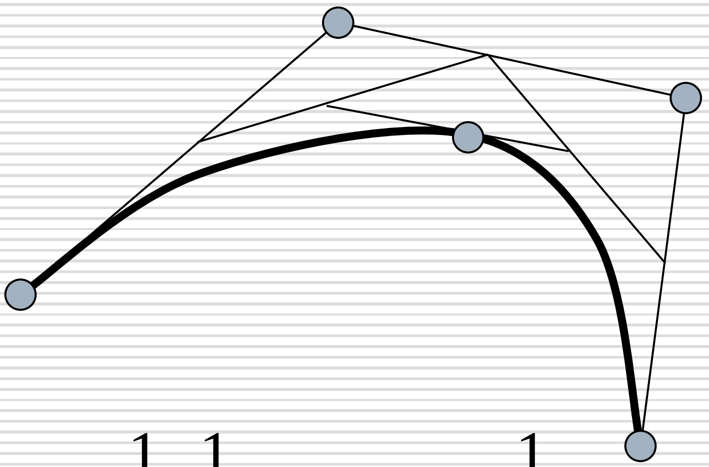
Subdividing Bézier Curves

- $Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4$
- How to draw the curve ?
- How to convert it to be line-segments ?



Subdividing Bézier Curves (de Casteljau's algorithm)

- $Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4$
- How to draw the curve ?
- How to convert it to be line-segments ?

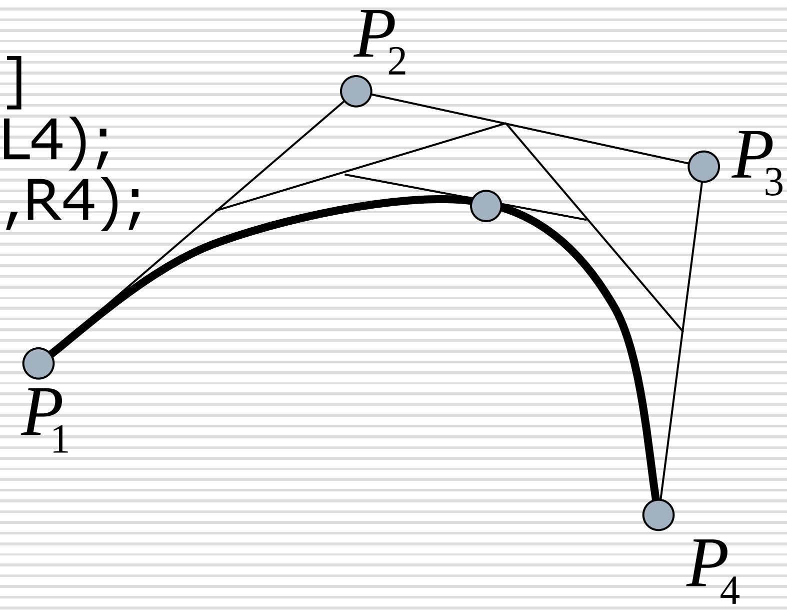


$$Q(\frac{1}{2}) = \frac{1}{8} P_1 + \frac{3}{8} P_2 + \frac{3}{8} P_3 + \frac{1}{8} P_4$$

$$= \frac{1}{2}(\frac{1}{2}(\frac{1}{2}(P_1 + P_2) + \frac{1}{2}(P_2 + P_3)) + \frac{1}{2}(\frac{1}{2}(P_3 + P_4) + \frac{1}{2}(P_2 + P_3)))$$

Display Bézier Curves

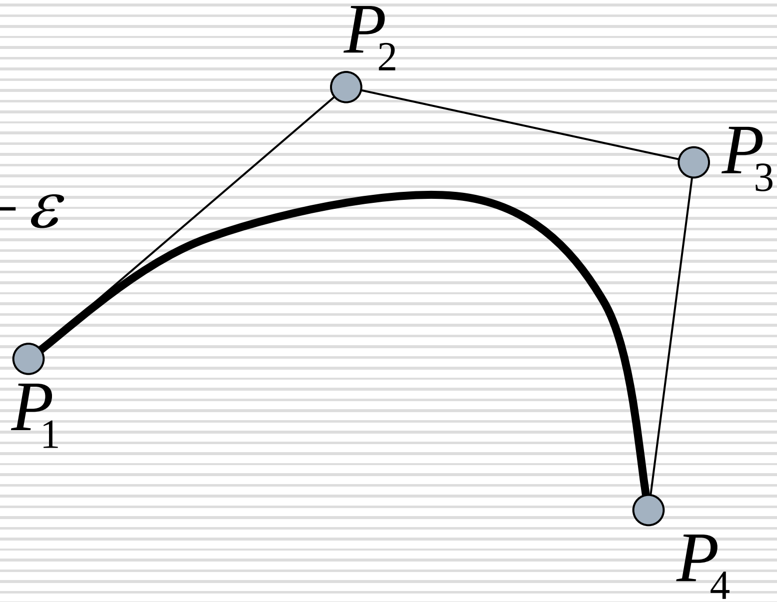
```
DisplayBezier(P1,P2,P3,P4)
begin
    if (FlatEnough(P1,P2,P3,P4))
        Line(P1,P4);
    else
        Subdivide(P[])=>L[],R[]
        DisplayBezier(L1,L2,L3,L4);
        DisplayBezier(R1,R2,R3,R4);
end;
```



Testing for Flatness

- Compare total length of control polygon to length of line connecting endpoints

$$\frac{|P_1 - P_2| + |P_2 - P_3| + |P_3 - P_4|}{|P_1 - P_4|} < 1 + \varepsilon$$

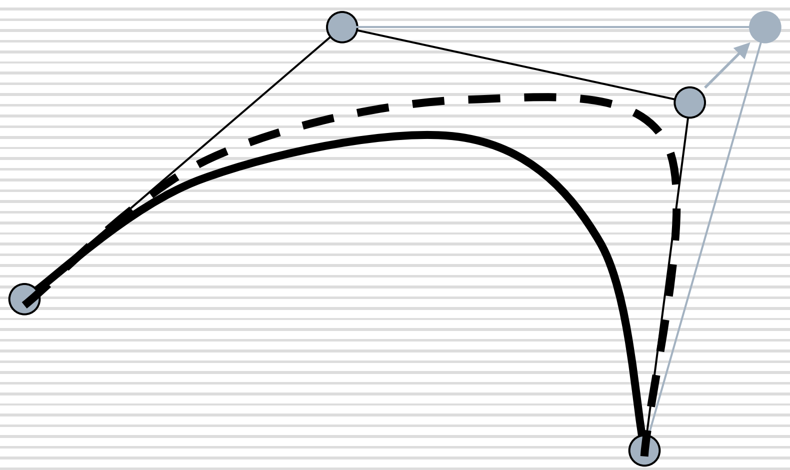


What do we want for a curve?

- Local control
- Interpolation
- Continuity

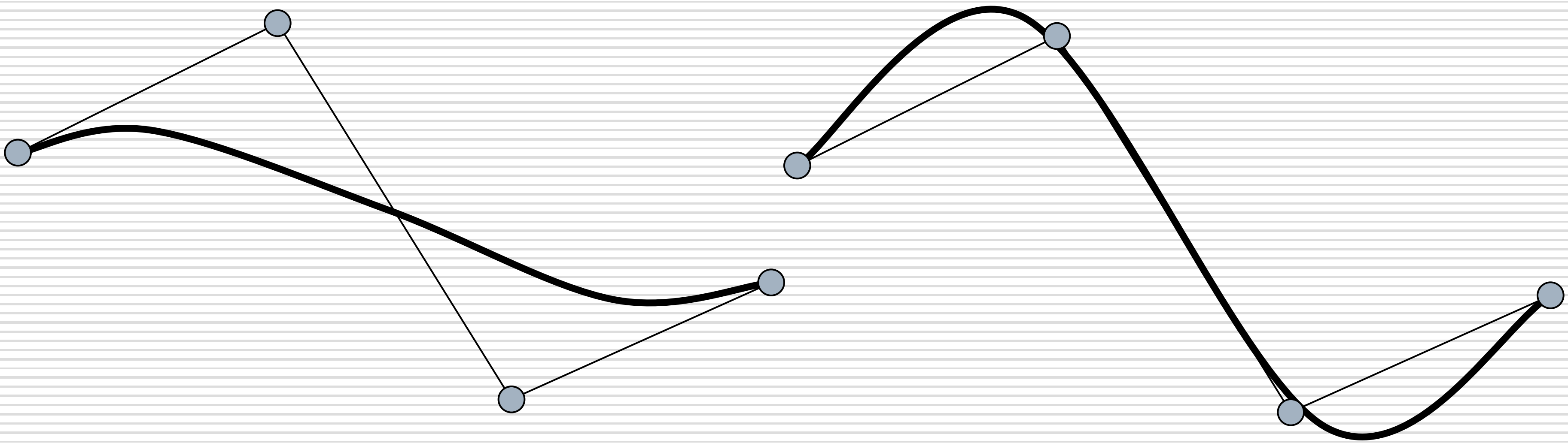
Local Control

- One problem with Bézier curve is that every control points affect every point on the curve (except for endpoints). Moving a single control point affects the whole curve.
- We'd like to have local control, that is, have each control point affect some well-defined neighborhood around that point.

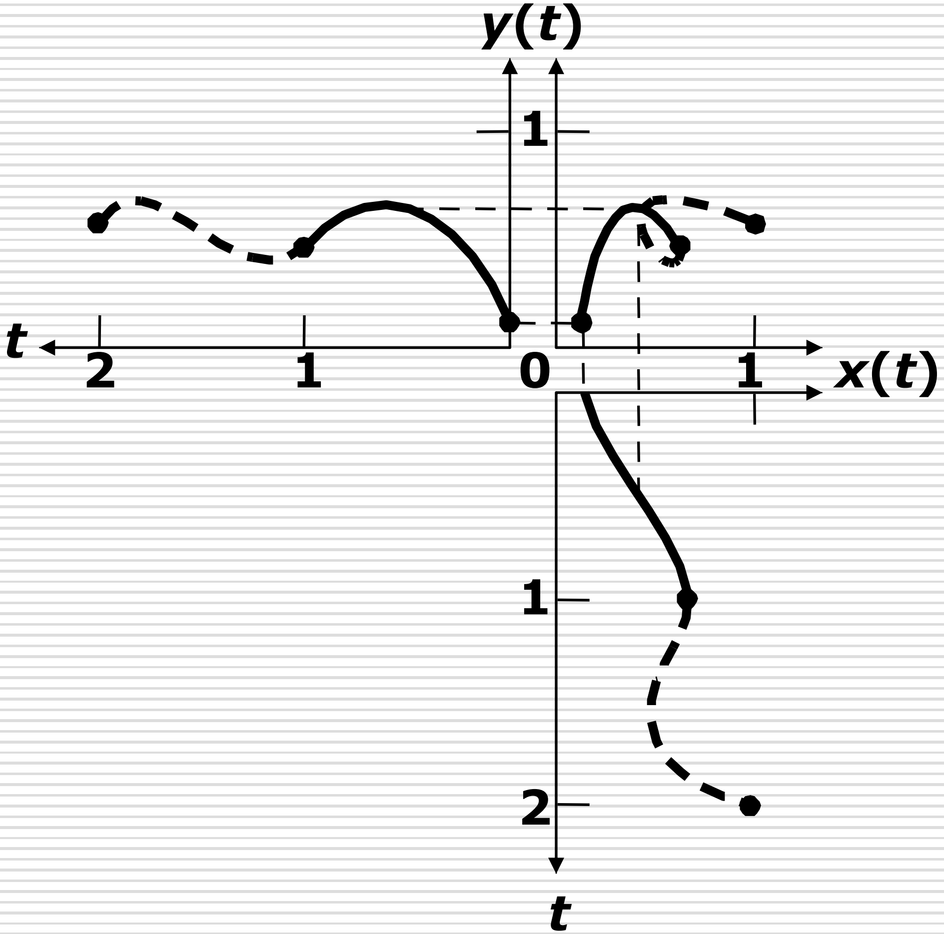


Interpolation

- Bézier curves are approximating. The curve does not necessarily pass through all the control points. We'd like to have a curve that is interpolating, that is, that always passes through every control points.



Continuity between Curve Segments



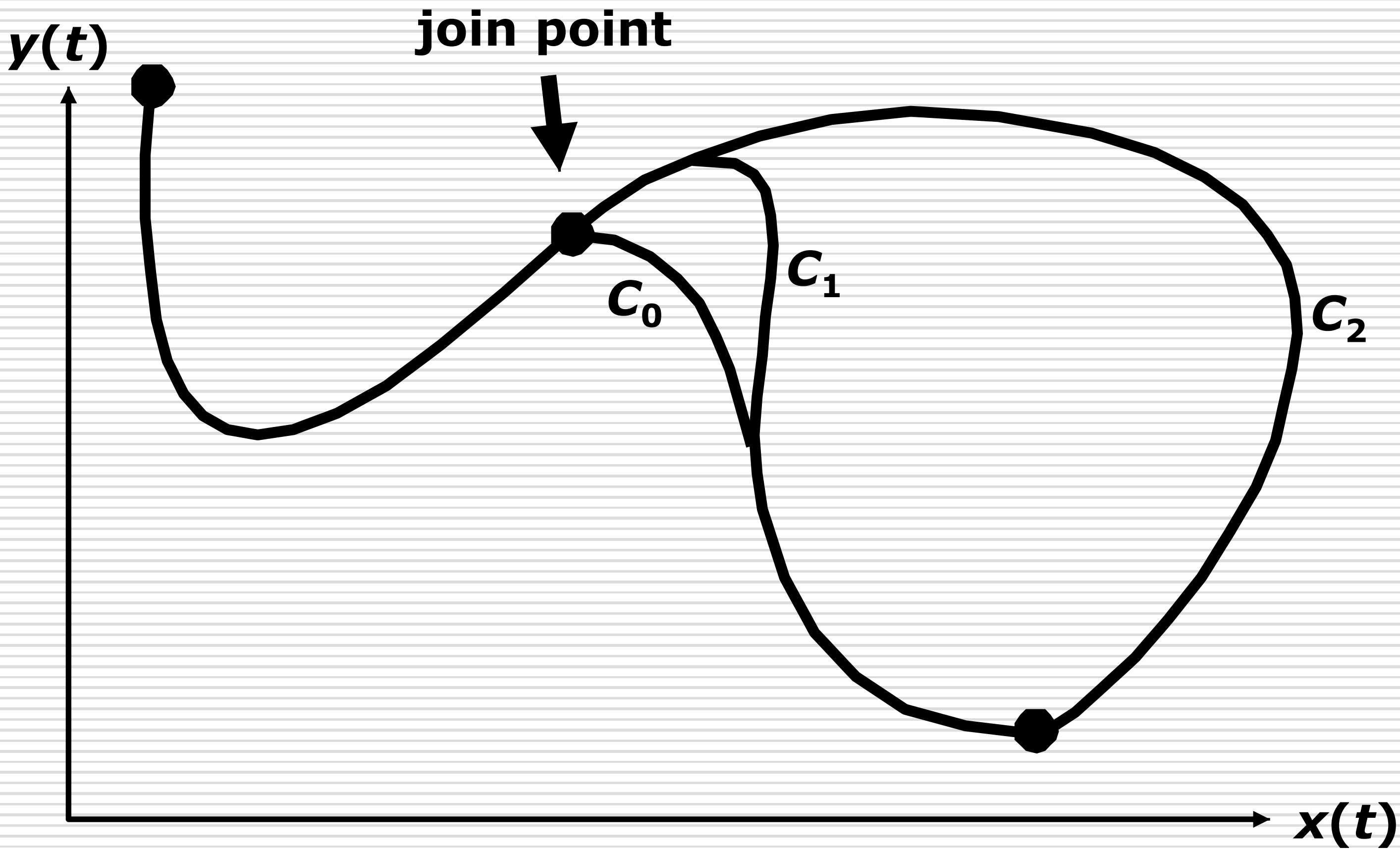
Continuity between Curve Segments

- G^0 geometric continuity
  - two curve segments join together
- G^1 geometric continuity
  - the directions *(but not necessarily the magnitudes)* of the two segments' tangent vectors are equal at a join point

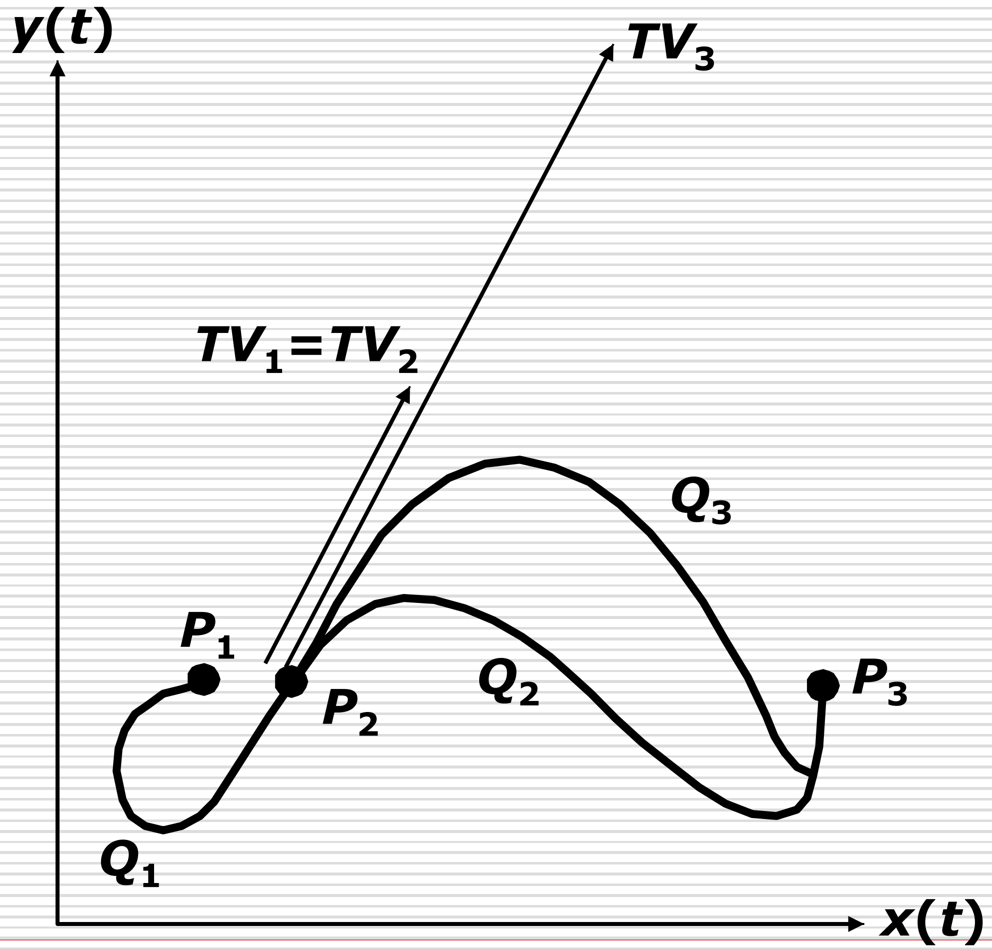
Continuity between Curve Segments

- C^1 continuous
  - the tangent vectors of the two cubic curve segments are equal *(both directions and magnitudes)* at the segments' join point
- C^n continuous
  - the direction and magnitude of $d^n / dt^n [Q(t)]$ through the *n*th derivative are equal at the join point

Continuity between Curve Segments



Continuity between Curve Segments



Bézier Curves → Splines

- Bézier curves have C-infinity continuity on their interiors, but we saw that they do not exhibit local control or interpolate their control points.
- It is possible to define points that we want to interpolate, and then solve for the Bézier control points that will do the job.
- But, you will need as many control points as interpolated points -> high order polynomials -> wiggly curves. (And you still won’t have local control.)

Bézier Curves → Splines

- We will splice together a curve from individual Bézier segments. We call these curves **splines**.
- When splicing Bézier together, we need to worry about continuity.

Ensuring C^0 continuity

- Suppose we have a cubic Bézier defined by (V_1, V_2, V_3, V_4), and we want to attach another curve (W_1, W_2, W_3, W_4) to it, so that there is C^0 continuity at the joint.

$$C^0 : Q_V(1) = Q_W(0)$$

- What constraint(s) does this place on (W_1, W_2, W_3, W_4) ?

$$Q_V(1) = Q_W(0) \Rightarrow V_4 = W_1$$

Ensuring C^1 continuity

- Suppose we have a cubic Bézier defined by (V_1, V_2, V_3, V_4), and we want to attach another curve (W_1, W_2, W_3, W_4) to it, so that there is C^1 continuity at the joint.

$$C^0 : Q_V(1) = Q_W(0)$$

$$C^1 : Q_V'(1) = Q_W'(0)$$

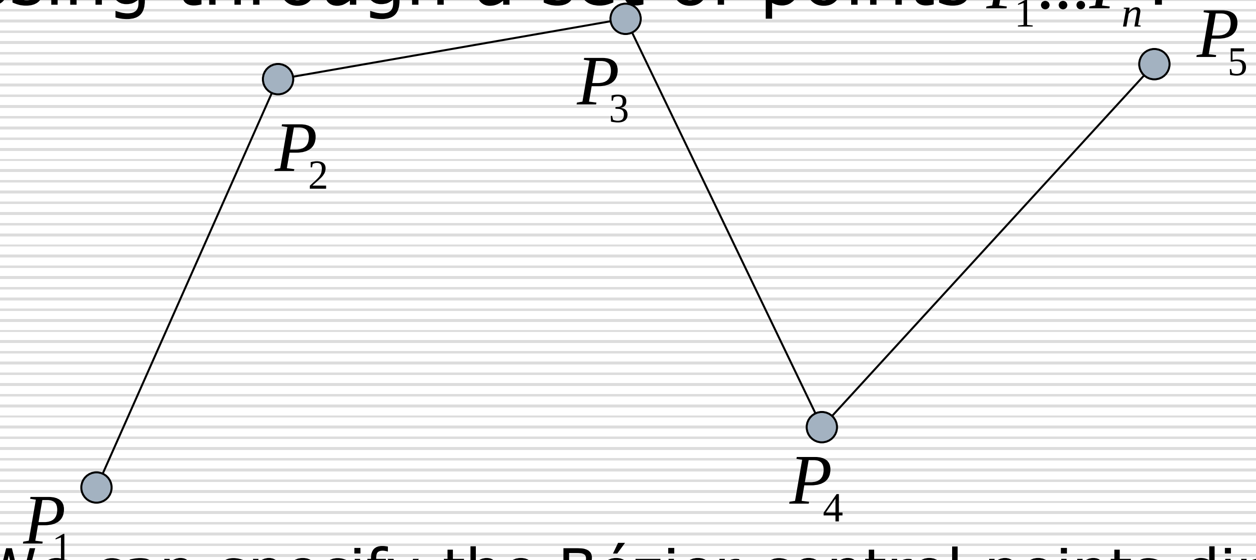
- What constraint(s) does this place on (W_1, W_2, W_3, W_4) ?

$$Q_V(1) = Q_W(0) \Rightarrow V_4 = W_1$$

$$Q_V'(1) = Q_W'(0) \Rightarrow V_4 - V_3 = W_2 - W_1$$

The C¹ Bézier Spline

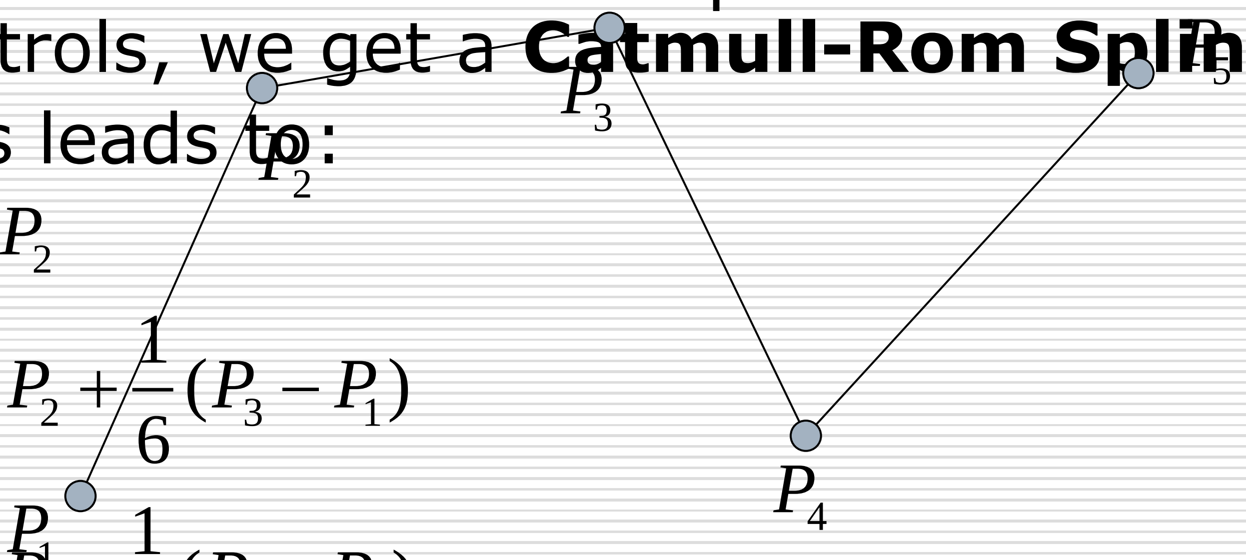
- How then could we construct a curve passing through a set of points $P_1 \ldots P_n$?



  - We can specify the Bézier control points directly, or we can devise a scheme for placing them automatically…

Catmull-Rom Spline

- If we set each derivative to be one half of the vector between the previous and next controls, we get a **Catmull-Rom Spline**.
- This leads to:



$$V_1 = P_2$$

$$V_2 = P_2 + \frac{1}{6}(P_3 - P_1)$$

$$V_3 = P_3 - \frac{1}{6}(P_4 - P_2)$$

$$V_4 = P_3$$

Catmull-Rom Basis Matrix

$$Q(t) = G_B \bullet M_B \bullet T$$

$$= G_B \bullet \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \bullet T \quad G_B = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-1}{6} & 1 & \frac{1}{6} & 0 \\ 0 & \frac{1}{6} & 1 & \frac{-1}{6} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

$$Q(t) = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \end{bmatrix} \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Ensuring C^2 continuity

- Suppose we have a cubic Bézier defined by (V_1, V_2, V_3, V_4), and we want to attach another curve (W_1, W_2, W_3, W_4) to it, so that there is C^2 continuity at the joint.

$$Q_V(1) = Q_W(0) \Rightarrow V_4 = W_1$$

$$Q'_V(1) = Q'_W(0) \Rightarrow V_4 - V_3 = W_2 - W_1$$

$$Q''_V(1) = Q''_W(0) \Rightarrow V_2 - 2V_3 + V_4 = W_1 - 2W_2 + W_3$$

$$\Downarrow$$

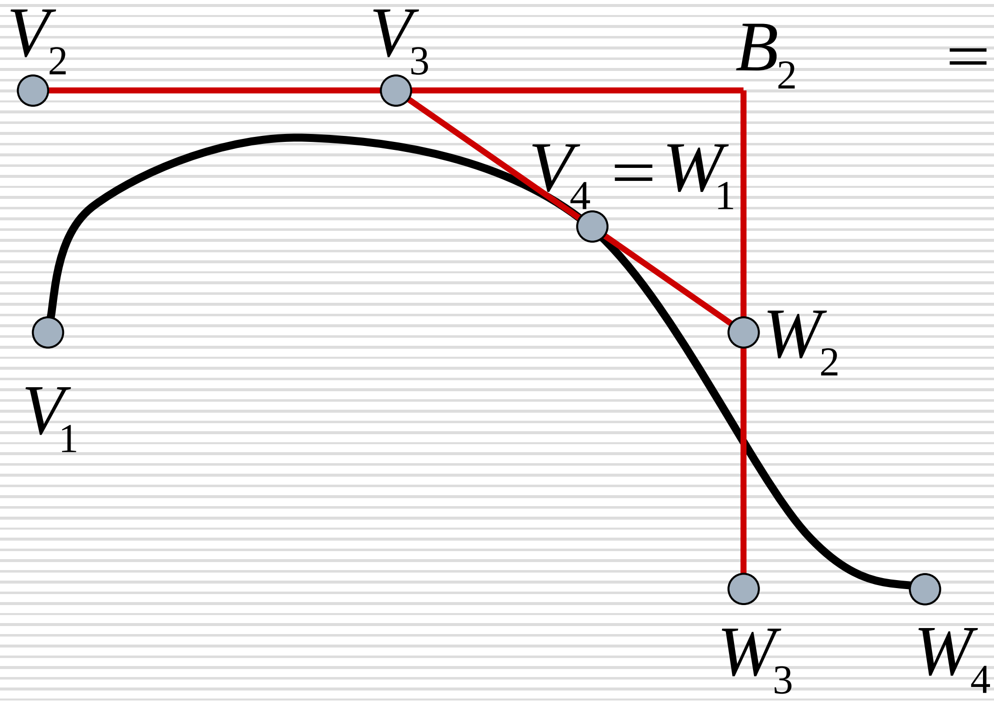
$$W_3 = V_2 - 4V_3 + 4V_4$$

B-Spline

- Instead of specifying the Bézier control points themselves, let's specify the corners of the A-frames in order to build a C^2 continuous spline.

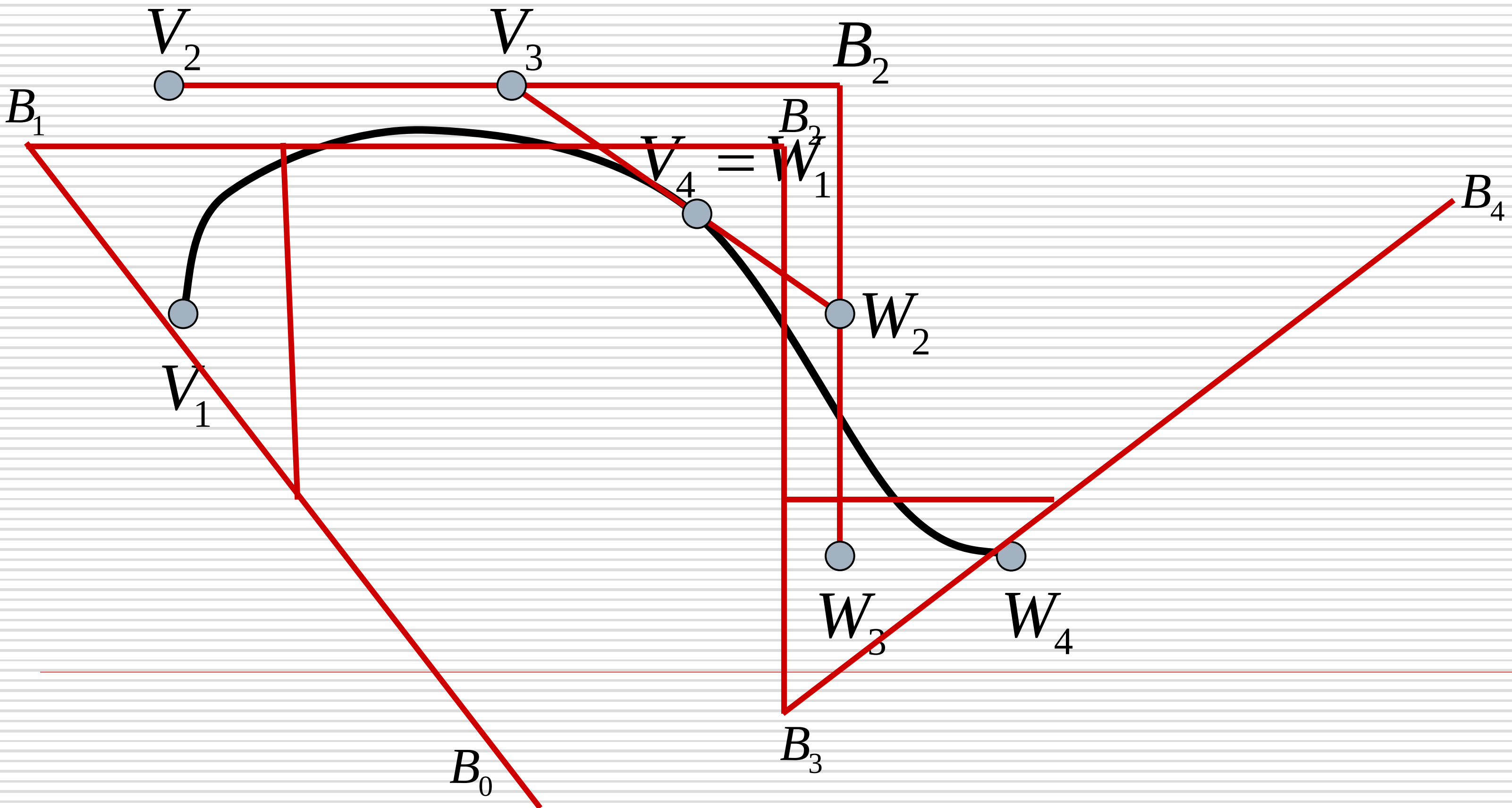
B-Spline

$$W_3 = V_2 - 4V_3 + 4V_4$$
$$= 2(2V_4 - V_3) - (2V_3 - V_2)$$
$$= 2W_2 - B_2$$



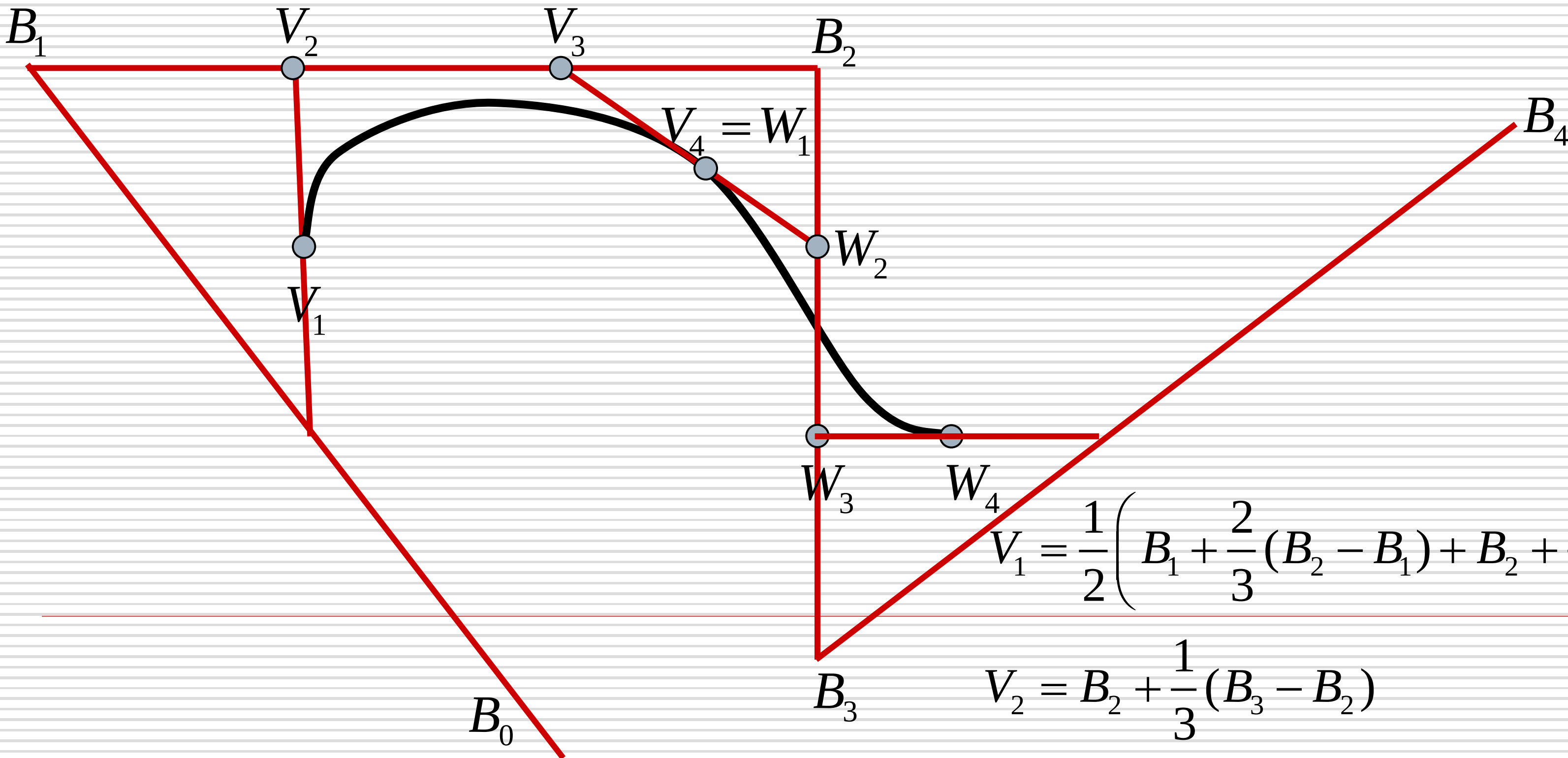
B-Spline

- Instead of specifying the Bézier control points themselves, let's specify the corners of the A-frames in order to build a C^2 continuous spline.



B-Spline

- Instead of specifying the Bézier control points themselves, let's specify the corners of the A-frames in order to build a C^2 continuous spline.

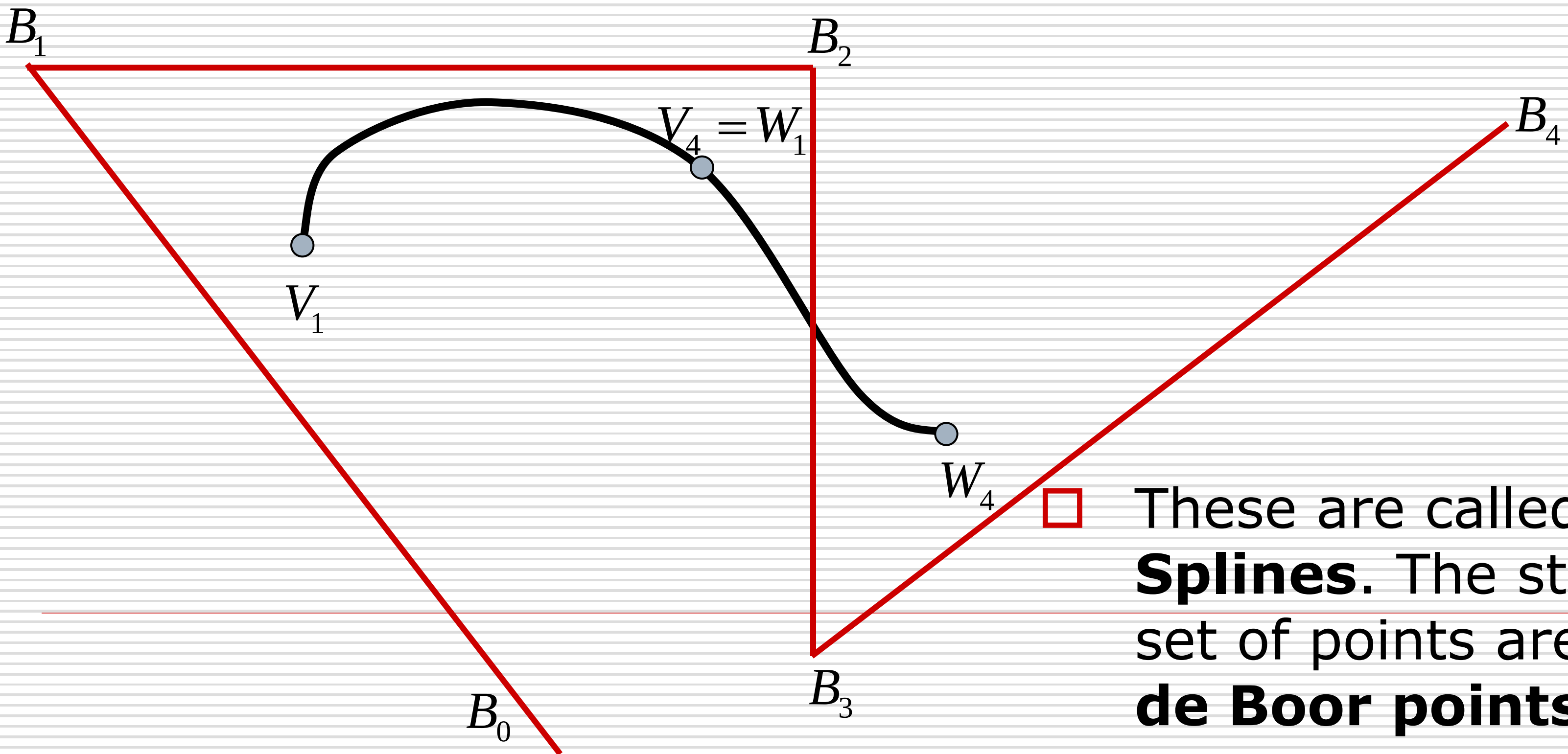


$$V_1 = \frac{1}{2}\left(B_1 + \frac{2}{3}(B_2 - B_1) + B_2 + \frac{1}{3}(B_3 - B_2) \right)$$

$$V_2 = B_2 + \frac{1}{3}(B_3 - B_2)$$

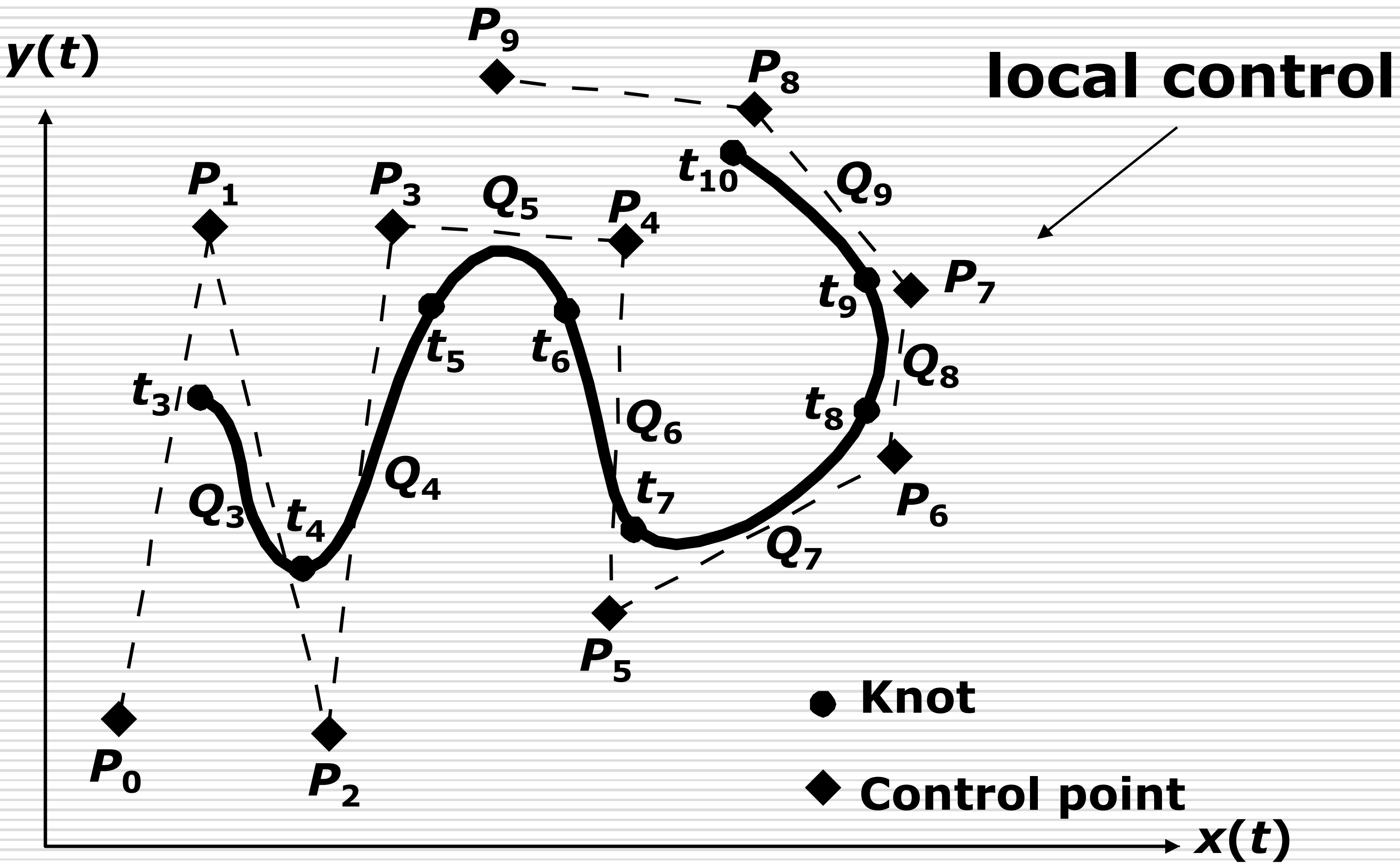
B-Spline

- Instead of specifying the Bézier control points themselves, let's specify the corners of the A-frames in order to build a C^2 continuous spline.



- These are called **B-Splines**. The starting set of points are called **de Boor points**.

B-Spline



y(t)
x(t)
P0
P1
P2
P3
P4
P5
P6
P7
P8
P9
Q3
Q4
Q5
Q6
Q7
Q8
Q9
t3
t4
t5
t6
t7
t8
t9
t10
local control
Knot
Control point

Uniform NonRational B-Splines

- cubic B-Spline
  - has $m+1$ control points $P_0, P_1, \ldots, P_m, m \geq 3$
  - has $m-2$ cubic polynomial curve segments $Q_3, Q_4, \ldots, Q_m$
- uniform
  - the knots are spaced at equal intervals of the parameter t
- non-rational
  - not rational cubic polynomial curves

Uniform NonRational B-Splines

- curve segment Q_i is defined by points $P_{i-3}, P_{i-2}, P_{i-1}, P_i$, thus
- **B-Spline geometry matrix**

$$G_{\mathrm{Bs}_i} = \begin{bmatrix} P_{i-3} & P_{i-2} & P_{i-1} & P_i \end{bmatrix}, \quad 3 \le i \le m$$

- if $T_i = \begin{bmatrix} (t-t_i)^3 & (t-t_i)^2 & (t-t_i) & 1 \end{bmatrix}^{\mathrm{T}}$
- then $Q_i(t) = G_{\mathrm{Bs}_i} \bullet M_{\mathrm{Bs}} \bullet T_i, \quad t_i \le t \le t_{i+1}$

Uniform NonRational B-Splines

- so **B-Spline basis matrix**

$$M_{\text{Bs}} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

- **B-Spline blending functions**

$$B_{\text{Bs}} = \frac{1}{6}\left[(1-\text{t})^3 \quad 3t^3 - 6t^2 + 4 \quad -3t^3 + 3t^2 + 3t + 1 \quad t^3\right]^{\text{T}}, \quad 0 \le t \le 1$$

NonUniform NonRational B-Splines

- the **knot-value sequence** is a nondecreasing sequence
- allow **multiple knot** and the number of identical parameter is the **multiplicity**
  - Ex. (0,0,0,0,1,1,2,3,4,4,5,5,5,5)
- so

$$Q_i(t) = P_{i-3} \bullet B_{i-3,4}(t) + P_{i-2} \bullet B_{i-2,4}(t) + P_{i-1} \bullet B_{i-1,4}(t) + P_i \bullet B_{i,4}(t)$$

NonUniform NonRational B-Splines

- where $B_{i,j}(t)$ is *j*th-order blending function for weighting control point P_i

$$B_{i,1}(t) = \begin{cases} 1, & t_i \le t \le t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$B_{i,2}(t) = \frac{t - t_i}{t_{i+1} - t_i} B_{i,1}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} B_{i+1,1}(t)$$

$$B_{i,3}(t) = \frac{t - t_i}{t_{i+2} - t_i} B_{i,2}(t) + \frac{t_{i+3} - t}{t_{i+3} - t_{i+1}} B_{i+1,2}(t)$$

$$B_{i,4}(t) = \frac{t - t_i}{t_{i+3} - t_i} B_{i,3}(t) + \frac{t_{i+4} - t}{t_{i+4} - t_{i+1}} B_{i+1,3}(t)$$

Knot Multiplicity & Continuity

- since $Q(t_i)$ is within the convex hull of P_{i-3}, P_{i-2}, and P_{i-1}
- if $t_i = t_{i+1}$, $Q(t_i)$ is within the convex hull of P_{i-3}, P_{i-2}, and P_{i-1} and the convex hull of P_{i-2}, P_{i-1}, and P_i, so it will lie on $\overline{P_{i-2}P_{i-1}}$
- if $t_i = t_{i+1} = t_{i+2}$, $Q(t_i)$ will lie on P_{i-1}
- if $t_i = t_{i+1} = t_{i+2} = t_{i+3}$, $Q(t_i)$ will lie on both P_{i-1} and P_i, and the curve becomes broken

Knot Multiplicity & Continuity

- multiplicity 1 : C^2 continuity
- multiplicity 2 : C^1 continuity
- multiplicity 3 : C^0 continuity
- multiplicity 4 : no continuity

NURBS: NonUniform Rational B-Splines

- rational
  - $x(t)$, $y(t)$, and $z(t)$ are defined as the ratio of two cubic polynomials
- rational cubic polynomial curve segments are ratios of polynomials

$$x(t) = \frac{X(t)}{W(t)} \quad y(t) = \frac{Y(t)}{W(t)} \quad z(t) = \frac{Z(t)}{W(t)}$$

- can be Bézier, Hermite, or B-Splines

Parametric Bi-Cubic Surfaces

- parametric cubic curves are $Q(t) = G \bullet M \bullet T$
- so, parametric bi-cubic surfaces are

  $Q(s) = G \bullet M \bullet S$
- if we allow the points in G to vary in 3D along some path, then

  $$Q(s,t) = \begin{bmatrix} G_1(t) & G_2(t) & G_3(t) & G_4(t) \end{bmatrix} \bullet M \bullet S$$
- since $G_i(t)$ are cubics

  $$G_i(t) = \boldsymbol{G}_i \bullet M \bullet T, \text{ where } \boldsymbol{G}_i = \begin{bmatrix} \boldsymbol{g}_{i1} & \boldsymbol{g}_{i2} & \boldsymbol{g}_{i3} & \boldsymbol{g}_{i4} \end{bmatrix}$$

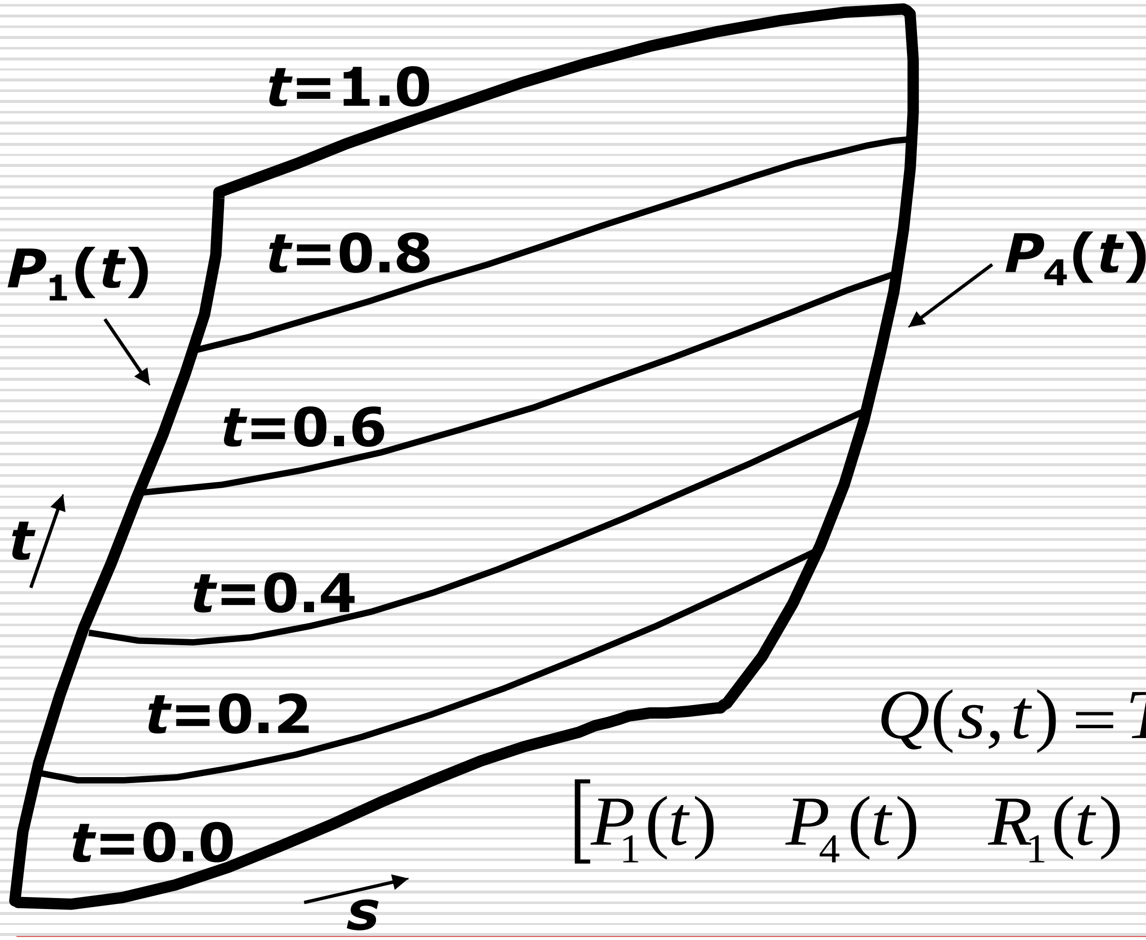
Parametric Bi-Cubic Surfaces

- so

$$Q(s,t) = T^{\mathrm{T}} \bullet M^{\mathrm{T}} \bullet \begin{bmatrix} \boldsymbol{g}_{11} & \boldsymbol{g}_{21} & \boldsymbol{g}_{31} & \boldsymbol{g}_{41} \\ \boldsymbol{g}_{12} & \boldsymbol{g}_{22} & \boldsymbol{g}_{32} & \boldsymbol{g}_{42} \\ \boldsymbol{g}_{13} & \boldsymbol{g}_{23} & \boldsymbol{g}_{33} & \boldsymbol{g}_{43} \\ \boldsymbol{g}_{14} & \boldsymbol{g}_{24} & \boldsymbol{g}_{34} & \boldsymbol{g}_{44} \end{bmatrix} \bullet M \bullet S$$

$$= T^{\mathrm{T}} \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet S, \qquad 0 \le s,t \le 1$$

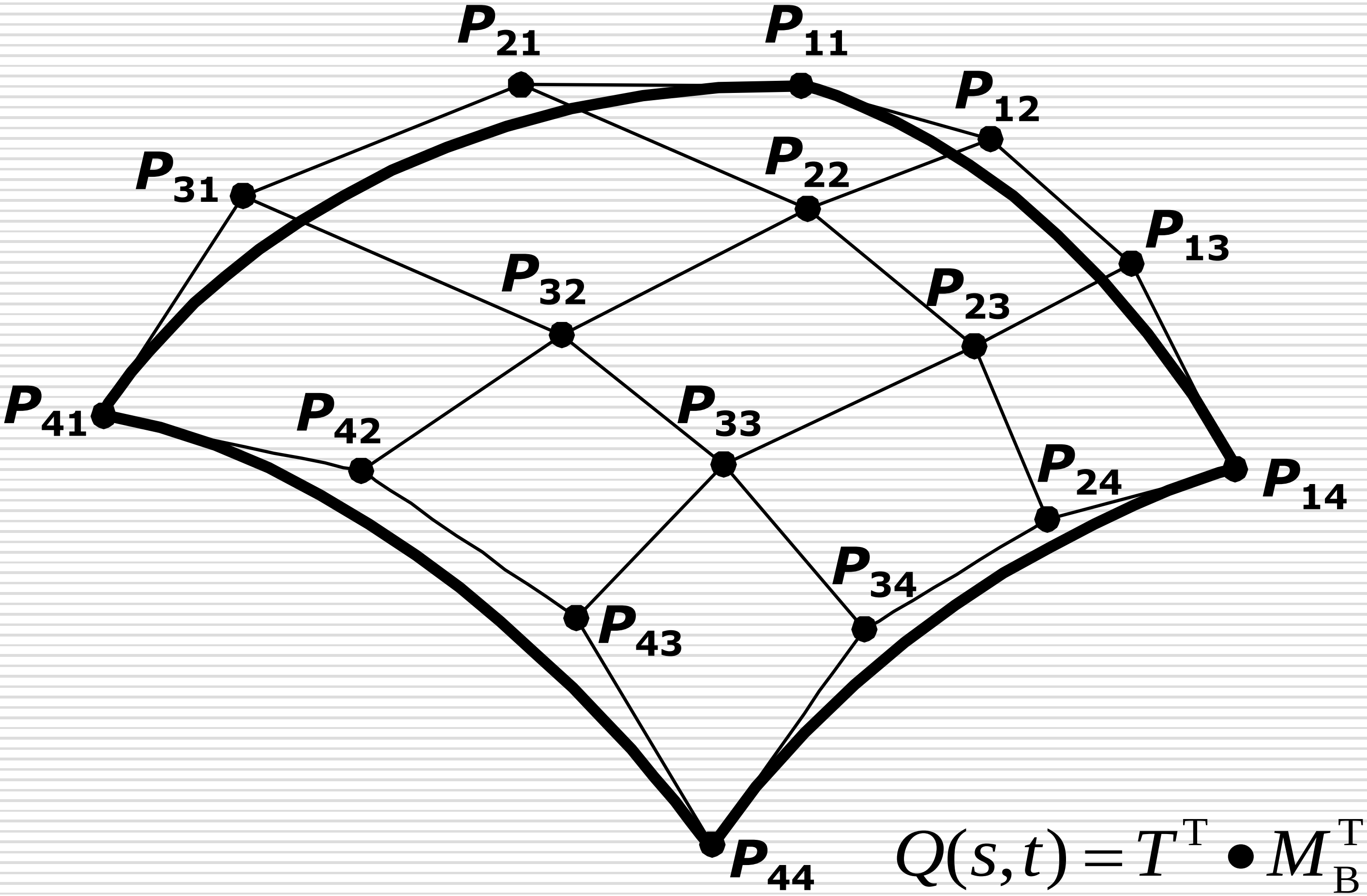
Hermite Surfaces



$$Q(s,t) = T^{\mathrm{T}} \bullet M_{\mathrm{H}}^{\mathrm{T}} \bullet \boldsymbol{G}_{\mathrm{H}} \bullet M_{\mathrm{H}} \bullet S$$

$$\begin{bmatrix} P_1(t) & P_4(t) & R_1(t) & R_4(t) \end{bmatrix} = \boldsymbol{G}_{\mathrm{H}} \bullet M_{\mathrm{H}} \bullet T$$

Bézier Surfaces



$$Q(s,t) = T^{T} \bullet M_{B}^{T} \bullet \boldsymbol{G}_{B} \bullet M_{B} \bullet S$$

Normals to Surfaces

$$\frac{\partial}{\partial s}Q(s,t) = T^{\mathrm{T}} \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet \frac{\partial}{\partial s}S$$

$$= T^{\mathrm{T}} \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet \begin{bmatrix} 3s^2 & 2s & 1 & 0 \end{bmatrix}^{\mathrm{T}}$$

$$\frac{\partial}{\partial t}Q(s,t) = \frac{\partial}{\partial t}\left(T^{\mathrm{T}}\right) \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet S$$

$$= \begin{bmatrix} 3t^2 & 2t & 1 & 0 \end{bmatrix}^{\mathrm{T}} \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet S$$

$$\frac{\partial}{\partial s}Q(s,t) \times \frac{\partial}{\partial t}Q(s,t) \longleftarrow$$ **normal vector**