



Modeling with curves



Motivation

- Need representations of smooth real world objects
- Art / drawings using CG need smooth curves
- Animation: camera paths



Naïve approach

- Simply use lines & polygons to approximate curves & surfaces
 - curve: piecewise linear function
 - lots of storage if accuracy desired
 - hard to interactively manipulate the shape
- Instead, we'll use higher-order functions
- Note: difference between stored model and rendered shape



Approaches

Explicit Functions: $y = f(x)$ [e.g. $y = 2x^2$]

1. Only one value of y for each x
2. Difficult to represent a slope of infinity

Implicit Equations: $f(x,y) = 0$ [e.g. $x^2 + y^2 - r^2 = 0$]

1. Need constraints to model just one part of a curve
2. Joining curves together smoothly is difficult

Parametric Equations: $x = f(t), y = g(t)$ [e.g. $x = t^2+3, y = 3t^2+2t+1$]

1. Slopes represented as parametric tangent vectors
2. Easy to join curve segments smoothly



Parametric Curves

Linear:

$$x = a_x t + b_x$$

$$0 \leq t \leq 1$$

$$y = a_y t + b_y$$

$$z = a_z t + b_z$$

Quadratic:

$$x = a_x t^2 + b_x t + c_x$$

$$y = a_y t^2 + b_y t + c_y$$

$$z = a_z t^2 + b_z t + c_z$$

Cubic:

$$x = a_x t^3 + b_x t^2 + c_x t + d_x$$

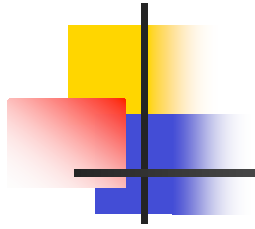
$$y = a_y t^3 + b_y t^2 + c_y t + d_y$$

$$z = a_z t^3 + b_z t^2 + c_z t + d_z$$

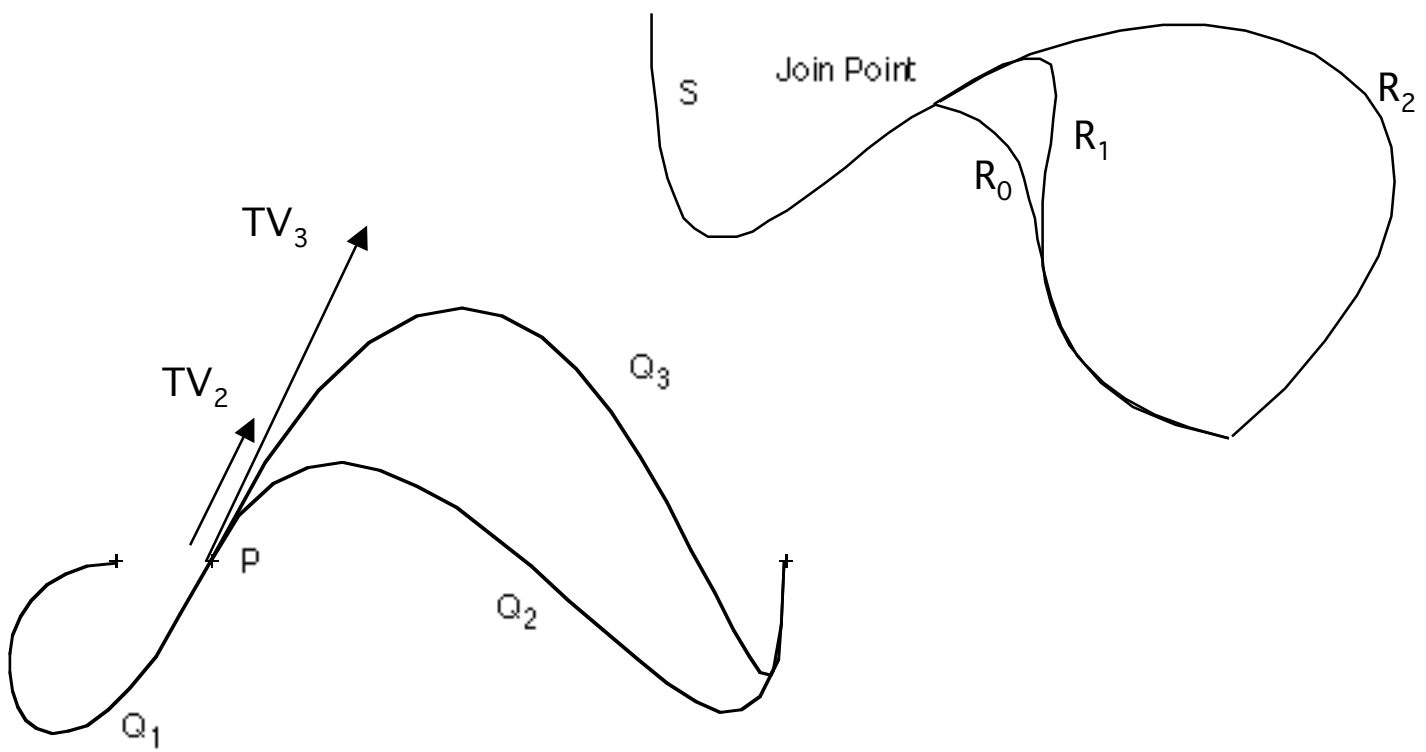


Joining Curve Segments Together

- G^0 geometric continuity: Two curve segments join together
- G^1 geometric continuity: The directions of the two segments' tangent vectors are equal at the join point
- C^1 continuity: Tangent vectors of the two segments are equal in magnitude and direction
 - ($C^1 \rightarrow G^1$ unless tangent vector = $[0,0,0]$)
- C^n continuity: Direction and magnitude through the n^{th} derivative are equal at the join point



Joining Examples





Notation

$$x = a_x t^3 + b_x t^2 + c_x t + d_x$$

$$y = a_y t^3 + b_y t^2 + c_y t + d_y$$

$$z = a_z t^3 + b_z t^2 + c_z t + d_z$$

$$Q(t) = T C$$

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

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



Alternate Notation

$$Q(t) = T M G$$

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

T Matrix

Basis Matrix

Geometry Matrix

- Idea: Different curves can be specified by changing the geometric information in the geometry matrix.
- The basis matrix contains information about the general family of curve that will be produced.



Example: Parametric Line

$$\begin{aligned}x(t) &= a_x t + b_x \\y(t) &= a_y t + b_y \\z(t) &= a_z t + b_z\end{aligned}$$

$$\begin{aligned}x'(t) &= a_x \\y'(t) &= a_y \\z'(t) &= a_z\end{aligned}$$

$$Q(t) = \begin{bmatrix} t & 1 \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{bmatrix}$$

$$Q'(t) = \begin{bmatrix} 1 & 0 \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{bmatrix}$$

$$Q(t) = \begin{bmatrix} t & 1 \\ m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} G_{1x} & G_{1y} & G_{1z} \\ G_{2x} & G_{2y} & G_{2z} \end{bmatrix}$$

$$Q'(t) = \begin{bmatrix} 1 & 0 \\ m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} G_{1x} & G_{1y} & G_{1z} \\ G_{2x} & G_{2y} & G_{2z} \end{bmatrix}_{10}$$

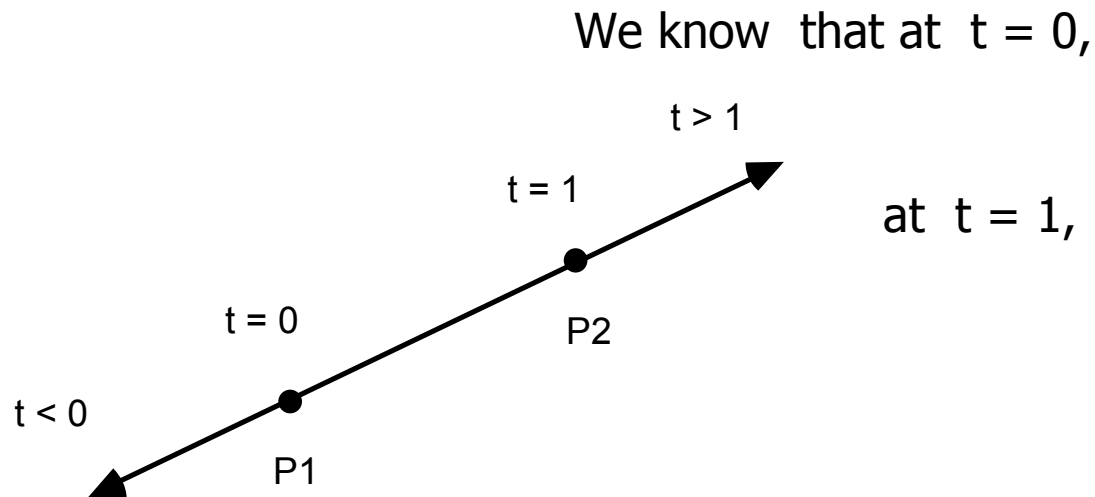


Example (cont.)

Since two points, P_1 & P_2 define a straight line, then the geometry matrix should be:

$$G_1 = P_1 \quad G_2 = P_2.$$

What is the basis matrix?



$$Q(0) = P_1$$

$$Q'(0) = P_2 - P_1$$

$$Q(1) = P_2$$

$$Q'(1) = P_2 - P_1$$



Example (cont.)

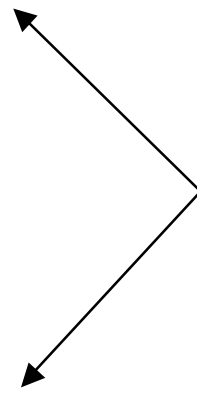
Solve these 4 simultaneous equations to find the basis matrix M_{line} :

$$Q(0) = [0, 1] M_{\text{line}} G = P_1$$

$$Q'(0) = [1, 0] M_{\text{line}} G = P_2 - P_1$$

$$Q(1) = [1, 1] M_{\text{line}} G = P_2$$

$$Q'(1) = [1, 0] M_{\text{line}} G = P_2 - P_1$$





Example (cont.)

$$m_{21}x_1 + m_{22}x_2 = x_1$$

$$m_{21}y_1 + m_{22}y_2 = y_1$$

$$m_{21}z_1 + m_{22}z_2 = z_1$$

$$(m_{11} + m_{21})x_1 + (m_{12} + m_{22})x_2 = x_2$$

$$(m_{11} + m_{21})y_1 + (m_{12} + m_{22})y_2 = y_2$$

$$(m_{11} + m_{21})z_1 + (m_{12} + m_{22})z_2 = z_2$$

$$m_{11}x_1 + m_{12}x_2 = x_2 - x_1$$

$$m_{11}y_1 + m_{12}y_2 = y_2 - y_1$$

$$m_{11}z_1 + m_{12}z_2 = z_2 - z_1$$

$$m_{11} = -1$$

$$m_{12} = 1$$

$$m_{21} = 1$$

$$m_{22} = 0$$

$$M_{line} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}$$

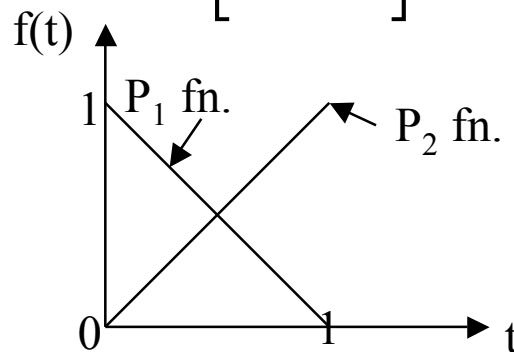
Blending functions

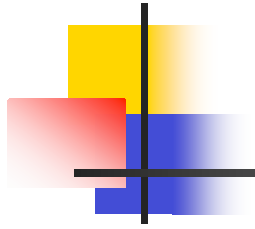
Multiplying the T and M matrices gives you a set of functions in t ; these are called *blending functions*.

There is one blending function for each of the pieces of geometric information in the G matrix.

The value of a blending function at a certain value of t determines the effect of the corresponding piece of geometric information at that point along the curve.

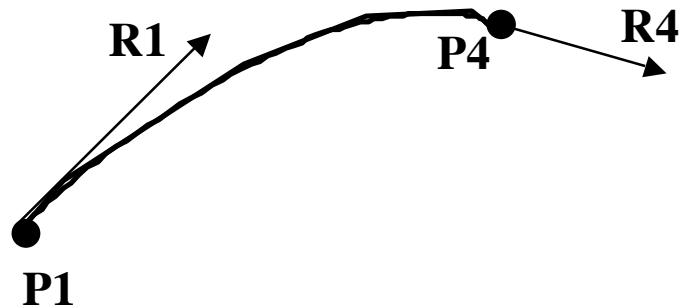
Line blending functions:
$$\begin{bmatrix} t & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1-t & t \end{bmatrix}$$





Hermite Curves

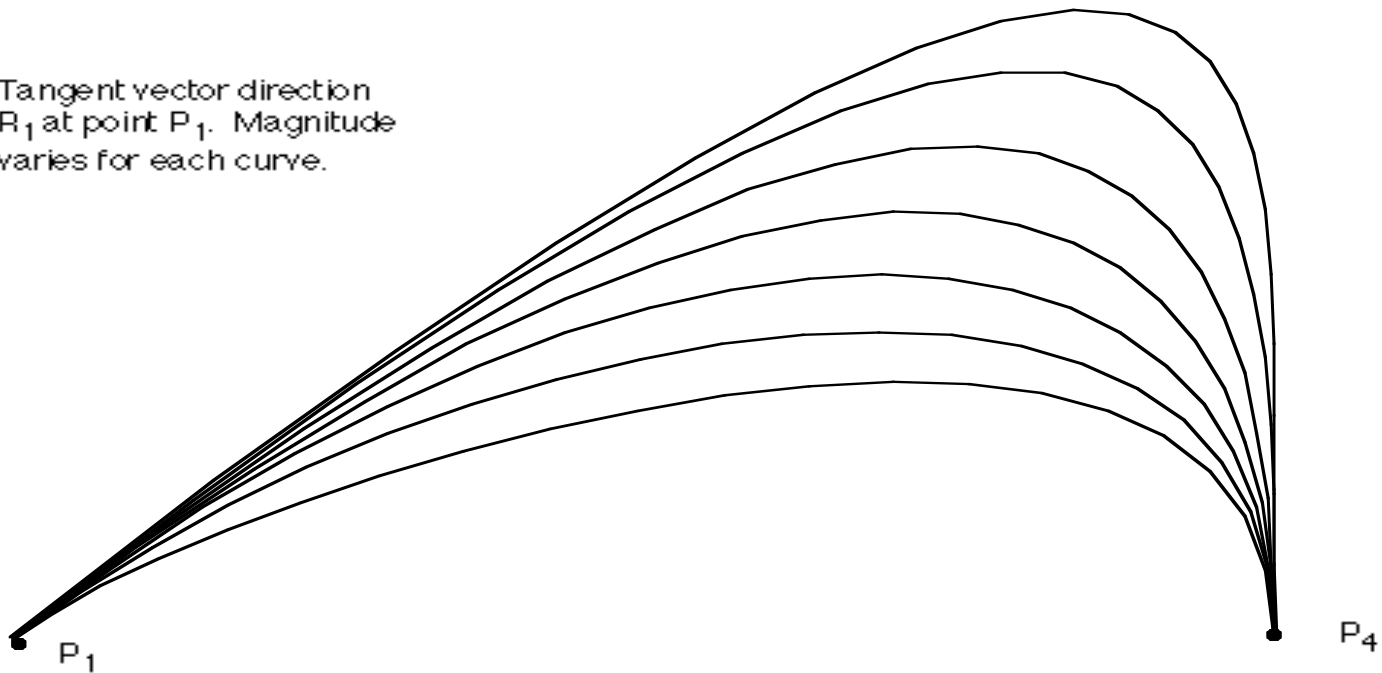
Defined by two endpoints and tangents at the endpoints.





Hermite Curve Examples

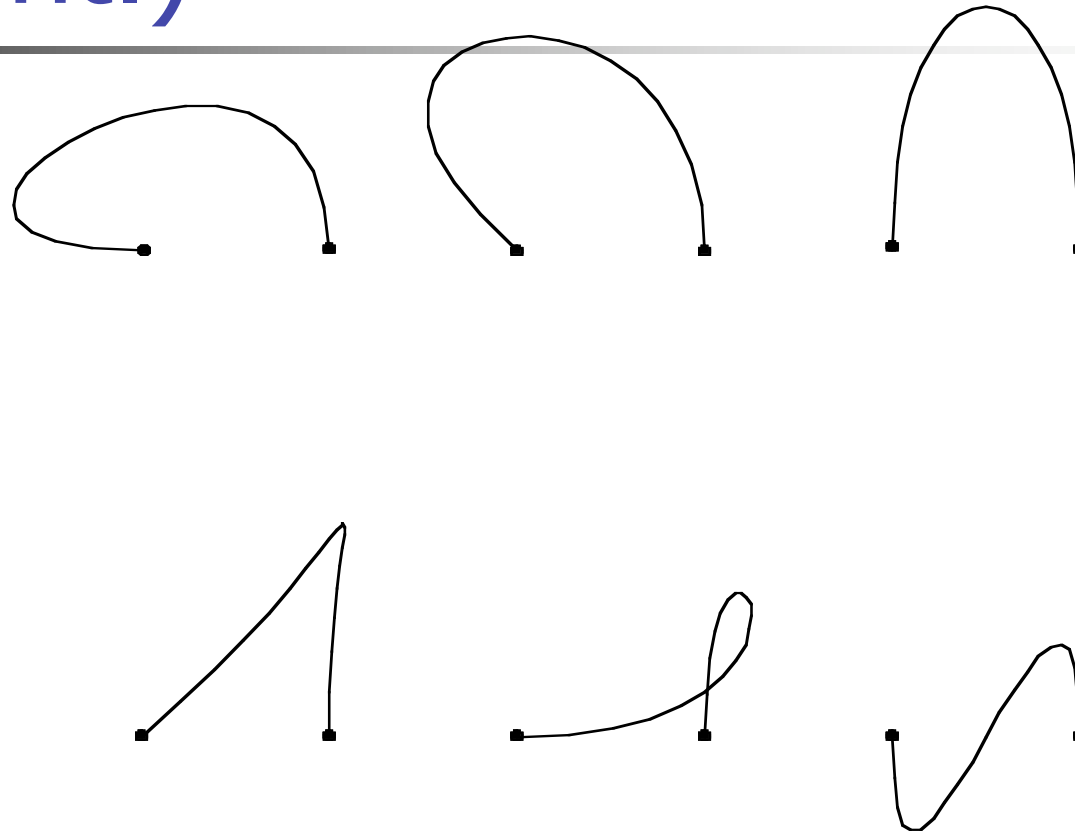
Tangent vector direction R_1 at point P_1 . Magnitude varies for each curve.



Tangent vector direction R_4 at point P_4 . Magnitude fixed for each curve.

Fig. 11.14 Family of Hermite parametric cubic curves. Only the tangent vector at P_4 varies for each curve, increasing for the higher curves.

Hermite Curve Examples (cont.)



Family of Hermite parametric cubic curves. Only the direction of the tangent vector at the left starting point varies; all tangent vectors have the same magnitude. A smaller magnitude would eliminate the loops in some of the curves.



Hermite Matrices

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

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

$$G_H = \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

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

Hermite Blending Functions

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

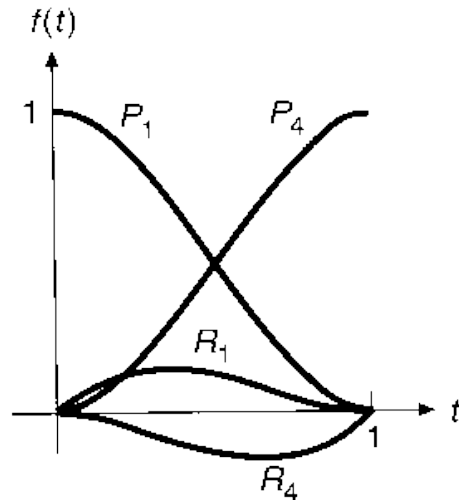


Figure 9.10

The Hermite blending functions, labeled by the elements of the geometry vector that they weight.

$$Q(t) = (2t^3 - 3t^2 + 1)P_1 + (-2t^3 + 3t^2)P_4 + (t^3 - 2t^2 + t)R_1 + (t^3 - t^2)R_4$$



Bézier Curves

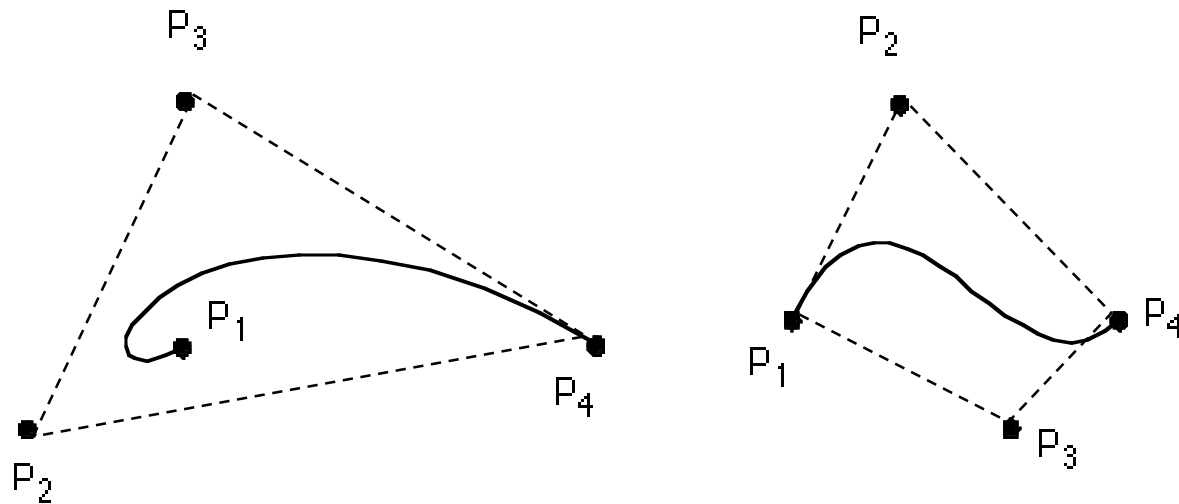


Fig. SHAPE.Bez1 Two Bézier curves and their control points. Notice that the convex hulls, shown as dashed lines, need not involve all four control points.



Bézier Matrices

$$G_B = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

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

- P_1 & P_4 : endpoints
- $R_1 = 3(P_2 - P_1)$, $R_4 = 3(P_4 - P_3)$
- constant velocity curves if control points are equally spaced

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

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

Bézier Blending Functions

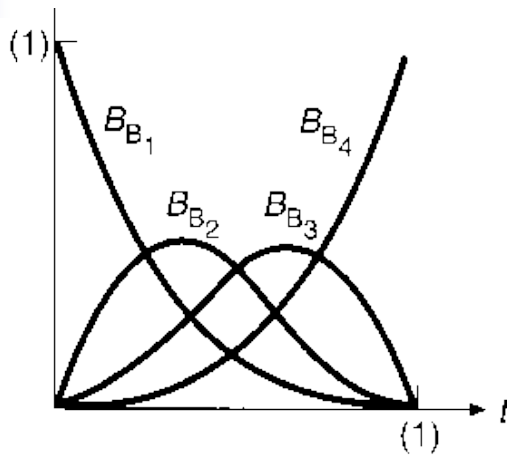


Figure 9.16

The Bernstein polynomials, which are the weighting functions for Bézier curves. At $t=0$, only B_{B_1} is nonzero, so the curve interpolates P_1 ; similarly, at $t=1$, only B_{B_4} is nonzero, and the curve interpolates P_4 .

$$Q(t) = (-t^3 + 3t^2 - 3t + 1)P_1 + (3t^3 - 6t^2 + 3t)P_2 + (-3t^3 + 3t^2)P_3 + t^3 P_4$$



General Bézier Curves

Let P_1 through P_{n+1} be points that define the curve.

Then:

$$Q_B(t) = \sum_{i=0}^n P_{i+1} B_{i,n}(t), t \in [0, 1]$$

where

$$B_{i,n}(t) = C(n,i) t^i (1-t)^{n-i} \quad \{\text{Blending functions}\}$$

and

$$C(n,i) = n!/(i!(n-i)!) \quad \{\text{Binomial coefficient}\}$$



General Bézier Curves (cont.)

For two points, $n = 1$:

$$Q_B(t) = (1-t)P_1 + tP_2$$

For three points, $n = 2$:

$$Q_B(t) = (1-t)^2P_1 + 2t(1-t)P_2 + t^2P_3$$

For four points, $n = 3$:

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



Bézier Curve Characteristics

1. The functions interpolate the first and last vertex points.
2. The tangent at P_0 is proportional to $P_1 - P_0$, and the tangent at P_n to $P_n - P_{n-1}$.
3. The blending functions are symmetric with respect to t and $(1-t)$. This means we can reverse the sequence of vertex points defining the curve without changing the shape of the curve.
4. The curve, $Q(t)$, lies within the *convex hull* defined by the control points.
5. If the first and last vertices coincide, then we produce a closed curve.
6. If complicated curves are to be generated, they can be formed by piecing together several Bézier sections.
7. By specifying multiple coincident points at a vertex, we pull the curve in closer and closer to that vertex.



Rendering curves

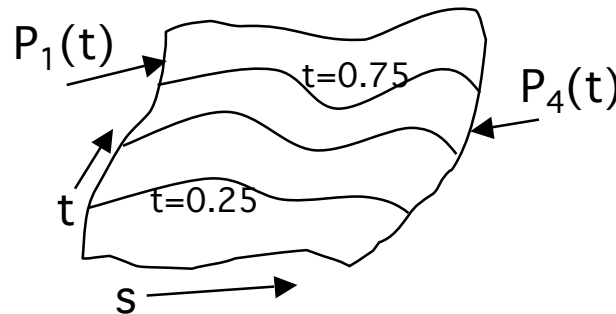
- Iterative method based on parametric formula evaluation
 - $t=0$
 - plot $x(t), y(t), z(t)$
 - increment t by a small amt. and repeat until $t \geq 1$
- This is slow, depending on step size
- Can increase performance using difference methods (as with line drawing)



Modeling with surfaces

Modeling surfaces

- Extension of parametric cubic curves called “parametric bicubic surfaces”
- Idea: infinite # of curves stacked together
 - equations now have 2 parameters
 - $Q(s,t)$





Matrix representation

- A single curve was expressed

$$Q(t) = TMG$$

- Now, the geometric information varies

$$Q(s,t) = SMG$$

$$G = \begin{bmatrix} G_1(t) \\ G_2(t) \\ G_3(t) \\ G_4(t) \end{bmatrix}$$

each $G_i(t)$ is itself
a cubic curve



Matrix representation (cont.)

- Since each G_i is a cubic curve, it can be written:

$$G_i(t) = TMG_i$$

$$G_i = \begin{bmatrix} g_{i1} \\ g_{i2} \\ g_{i3} \\ g_{i4} \end{bmatrix}$$



Matrix representation (cont.)

- Substituting, we obtain:

$$Q(s,t) = SM \begin{bmatrix} TMG_1 \\ TMG_2 \\ TMG_3 \\ TMG_4 \end{bmatrix}$$

$$= SMTM \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix}$$

This formulation does not work in terms of matrix dimensions...



Matrix representation (cont.)

So, use the transpose rule:

$$G_i(t) = G_i^T M^T T^T$$

$$Q(s,t) = S M \begin{bmatrix} G_1^T \\ G_2^T \\ G_3^T \\ G_4^T \end{bmatrix} M^T T^T$$

$$= S M \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix} M^T T^T$$



Matrix representation (cont.)

- Finally, remember that the large geometry matrix has 3 components (x , y , z) for each g_{ij} , so that we get three parametric equations:

$$x(s,t) = S M G_x M^T T^T$$

$$y(s,t) = S M G_y M^T T^T$$

$$z(s,t) = S M G_z M^T T^T$$



Hermite surfaces

- extension of Hermite curves to parametric bicubic surfaces
- four elements of the geometry matrix are now $P_1(t)$, $P_4(t)$, $R_1(t)$, $R_4(t)$
- can be thought of as interpolating the curves $Q(s,0)$ and $Q(s,1)$ or $Q(0,t)$ and $Q(1,t)$



Hermite surface matrices

$$x(s,t) = S M G_{H_x} M^T T^T$$

$$y(s,t) = S M G_{H_y} M^T T^T$$

$$z(s,t) = S M G_{H_z} M^T T^T$$

$$P_{1_x}(t) = T M G_{1_x}$$

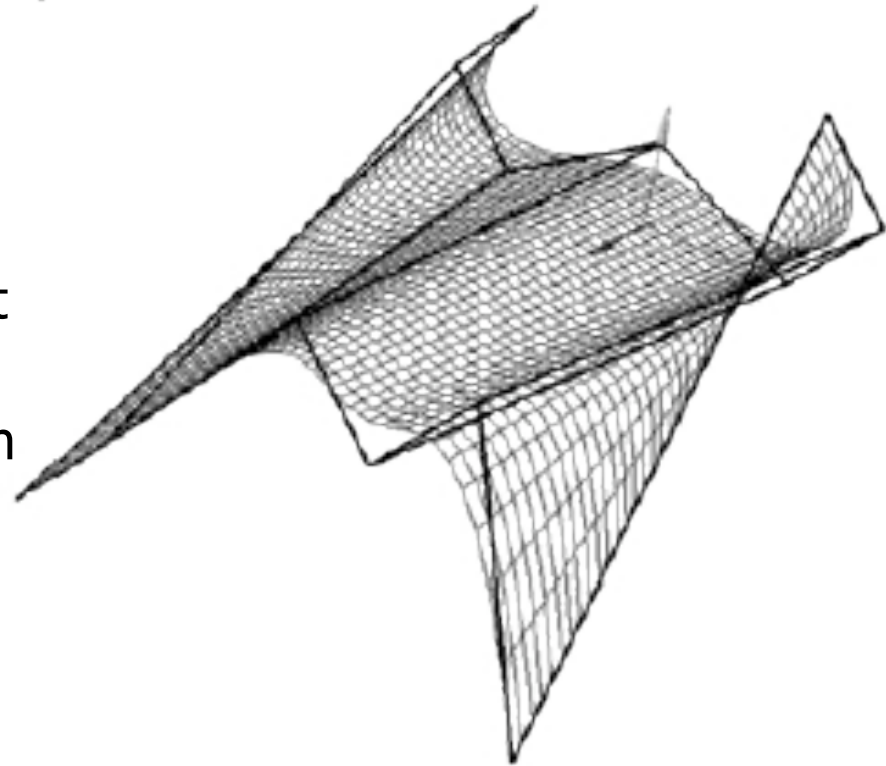
$$G_{1_x} = \begin{bmatrix} g_{11} \\ g_{12} \\ g_{13} \\ g_{14} \end{bmatrix}$$

$$G_{H_x} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix}$$

Rendering surfaces

- Can also use iterative methods in s and t
- Grid representation:
 - render curves of constant t and constant s
 - use perspective and hidden lines to indicate surface shape

a)



Rendering surfaces

- Solid representation:
 - subdivide surface into “flat” sections
 - render quadrilateral for each section
 - shape indicated by shading (need normal)

a)

