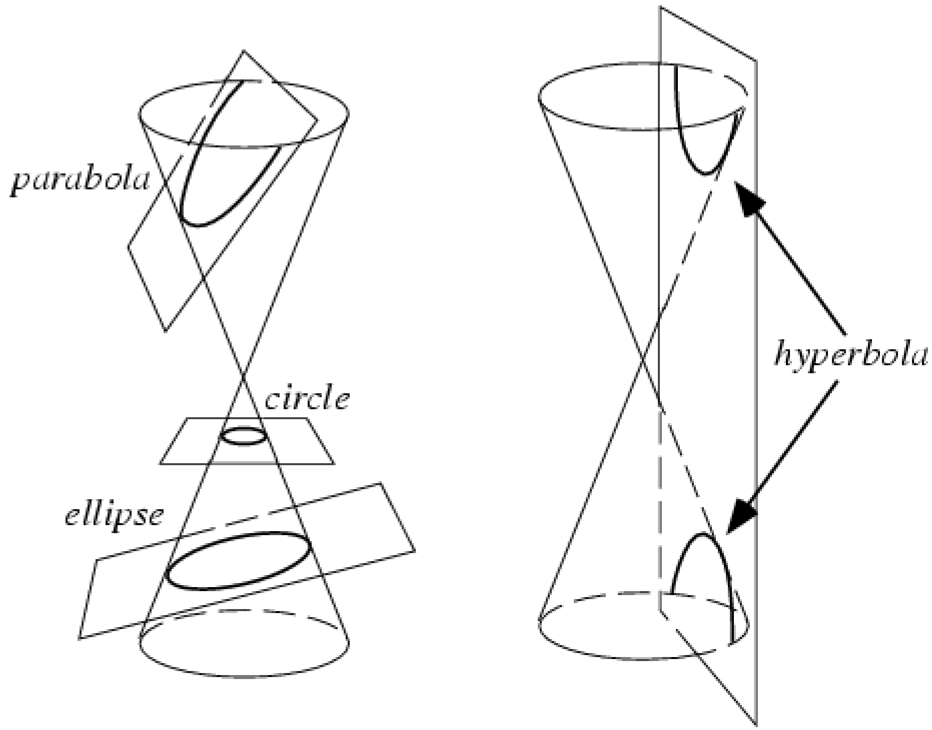
CURVES

UNIT -: 3

CONIC SECTIONS (2D & 2ND DEGREE)

The General Equation for a Conic Section:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$



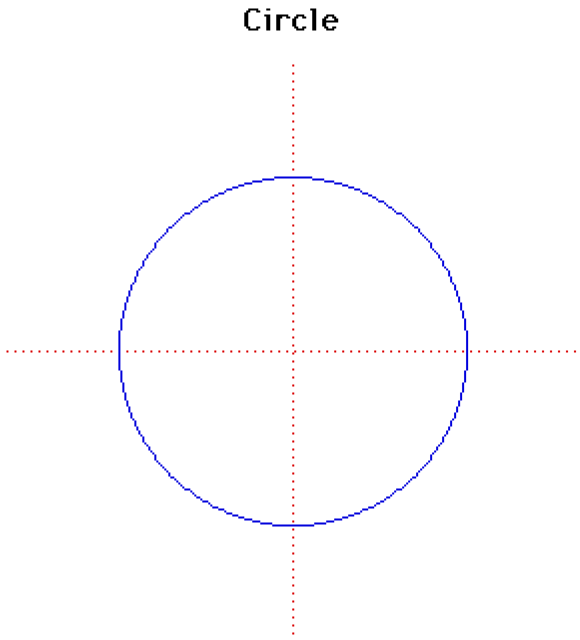
CONIC SECTIONS (2D & 2ND DEGREE)

The type of section can be found from the sign of: $B^2 - 4AC$

If $B^2 - 4AC$ is...	then the curve is a...
< 0	ellipse, circle, point or no curve.
$= 0$	parabola, 2 parallel lines, 1 line or no curve.
> 0	hyperbola or 2 intersecting lines.

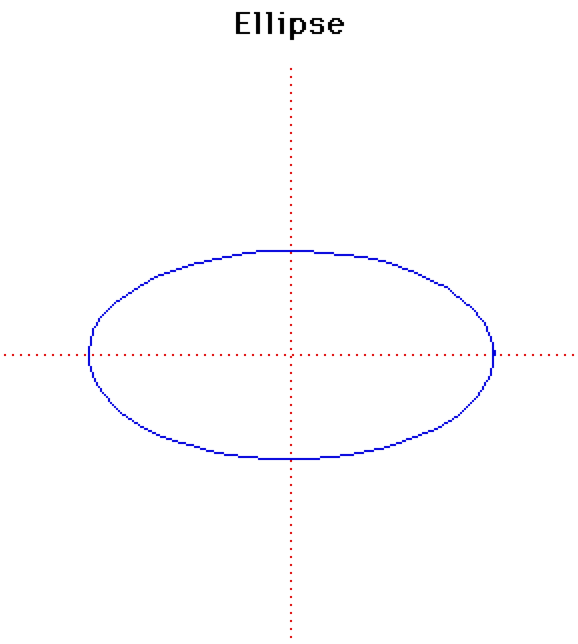
Circle

- **Cartesian equation:** $x^2 + y^2 = a^2$
- **Parametrical equation:**
  $x = a\cos(t), y = a\sin(t)$



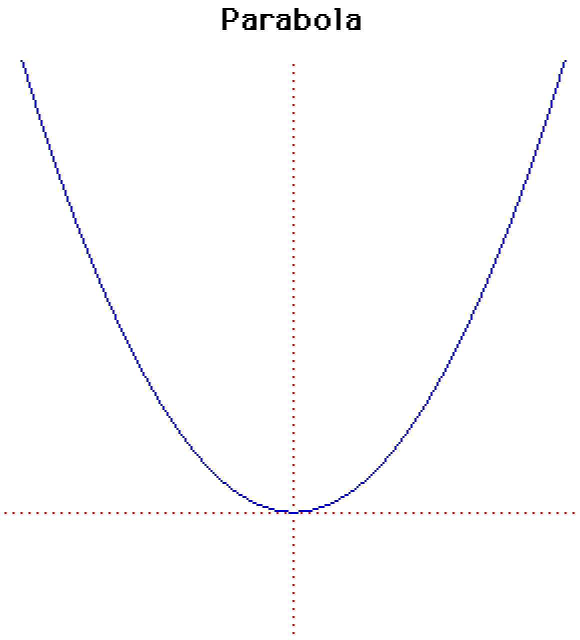
Ellipse

- **Cartesian equation:** $x^2/a^2 + y^2/b^2 = 1$
- **Parametrical equation:** $x = a\cos(t)$, $y = b\sin(t)$



Parabola

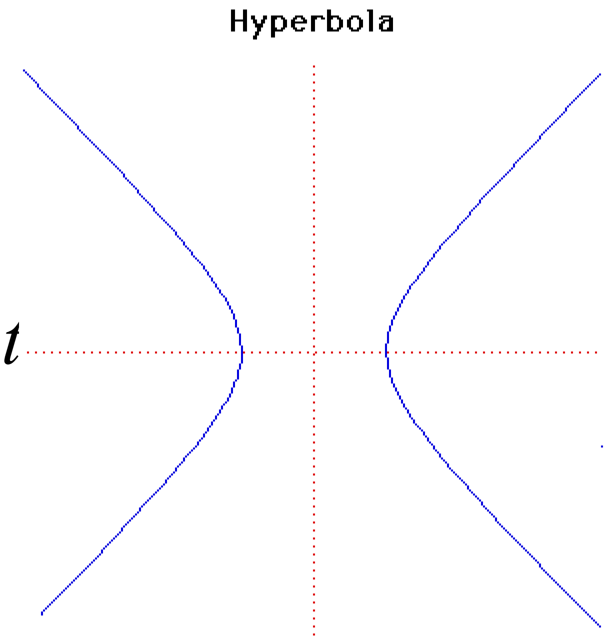
- **Cartesian equation:**
  $y = ax^2 + bx + c$



Hyperbola

- **Cartesian equation:**
  $x^2/a^2 - y^2/b^2 = 1$

- **Parametrical equation:**
  $x = a$ sec(t) $= a/\cos(t)$, $y = b \tan(t$



PARAMETRIC CURVE

- **explicit form: y = f(x) is computed by the function f, and the pair of coordinates (x, y) sweeps out the curve --**
- **A parametric curve: x(t) and y(t). As t varies, the coordinates (x(t), y(t)) sweep out the curve:**

  x(t) = sin(t), y(t) = cos(t)
- **CAGD (Computer Aided Geometric Design) deals primarily with polynomial or rational functions, not trigonometric functions:**

  x(t) = 2t/(1+t*t), y(t)=(1-t*t)/(1+t*t)
- **Both equations above yield circles, so how do they differ? It is the parameterization.**

- **The motion of the point (x(t), y(t)) is different, even if the paths (the circles) are the same.**

- **The slope is given by the tangent line at any point (x'(t), y'(t)), which determines the speed the point traces out the curve**

- ***t* moves the point (x(t), y(t)) along the path of the curve; the point's speed varies as *t* varies (derivative vector changes in length.)**

PARAMETRIC CUBIC CURVES

- **In general, a parametric polynomial is written as**

$$f(t) = a_0 + a_1 t + a_2 t^2 + \quad + a_n t^n$$

- **Parametric cubics are the lowest-degree curves that are nonplanar in 3D; generated from an input set of math functions or data points.**

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad where \quad 0 \le t \le 1$$

where the curve: $Q(t) = [x(t), y(t), z(t)]$

- ***dQ/dt = Q'* is the parametric tangent vector**

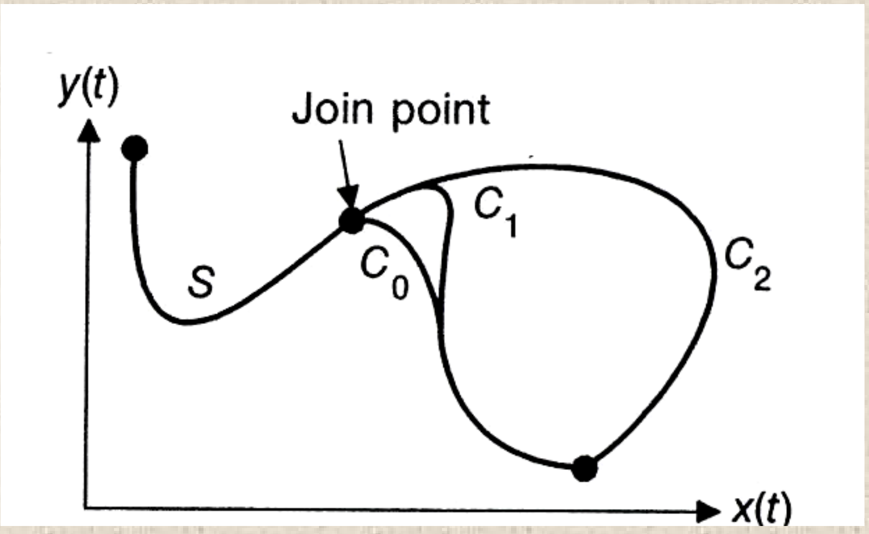
- **Continuity condition -- parametric continuity:**

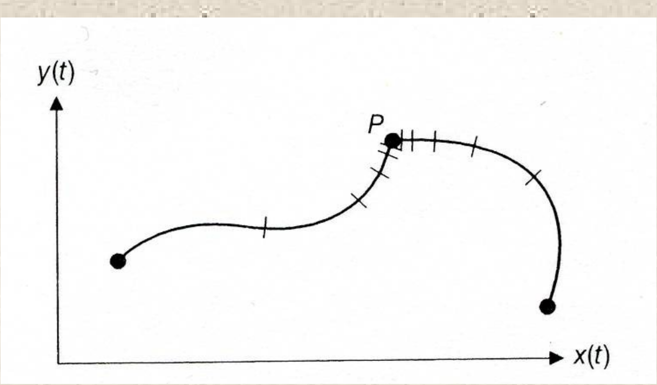
Zero-order parametric continuity, C^0, means simply that the curves meet. 1st-order parametric continuity, C^1, means that the 1st derivatives for two successive curve sections are equal at their joining point:

$$Q'_1(t=1) = Q'_2(t=0)$$

Second-order parametric continuity, C^2, means that both the first and second parametric derivatives of the two curve sections are the same at the intersection:

$$Q''_1(t=1) = Q''_2(t=0)$$





- **Problems: a particle travels in a straight line, but has distinct jumps in velocity -- not C^1, but the curve is smooth; conversely, a C^1 curve can have a kink in it when the velocity of the particle goes to zero where it changes direction and starts up again.**

- **Continuity condition -- geometric continuity:**

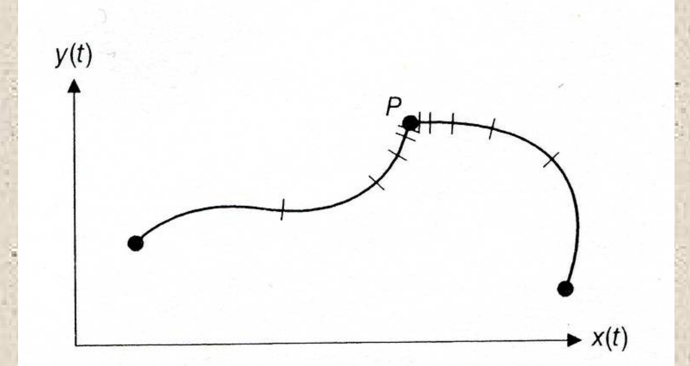
Zero-order geometric continuity, G^0, means simply that the curves meet. 1st-order geometric continuity, G^1, means that the 1st derivatives for two successive curve sections are proportional at their joining point. That is, the tangent changes continuously; the directions (but not necessarily the magnitudes) of the two segments' tangent vectors are equal at a join point:

$$Q'_1(t=1) = kQ'_2(t=0)$$

Second-order geometric continuity, G^2, means that both the first and second parametric derivatives of the two curve sections are proportional at the intersection:

$$Q''_1(t=1) = kQ''_2(t=0)$$

- **In general, C^1 implies G^1, but if a C^1 curve has a kink because its derivative goes to zero, then this curve will not be G^1, since the tangent direction changes discontinuously at the kink.**



A line segment

geometric constraints

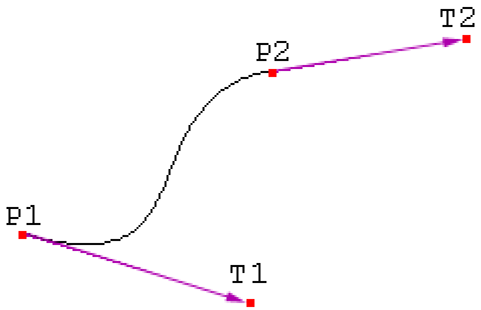
$$x(t) = g_{1x}(1 - t) + g_{2x}(t),$$
$$y(t) = g_{1y}(1 - t) + g_{2y}(t),$$
$$z(t) = g_{1z}(1 - t) + g_{2z}(t).$$

blending functions

Hermite Curves

- **Constraints on the endpoints P_1 and P_4 and tangent vectors at endpoints R_1 and R_4:**

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} M_h \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}_x$$



We can easily arrive at:

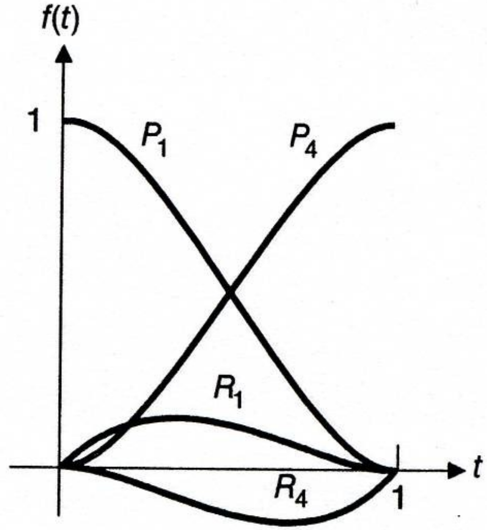
$$M_h = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Expanding the product gives the Hermite blending functions:

$$
\begin{aligned}
Q(t) &= H_0(t)P_1 + H_1(t)P_4 + H_2(t)R_1 + H_3(t)R_4 \\
&= (2t^3 - 3t^2 + 1)P_1 + (-2t^3 + 3t^2)P_4 + \\
&\qquad (t^3 - 2t^2 + t)R_1 + (t^3 - t^2)R_4
\end{aligned}
$$

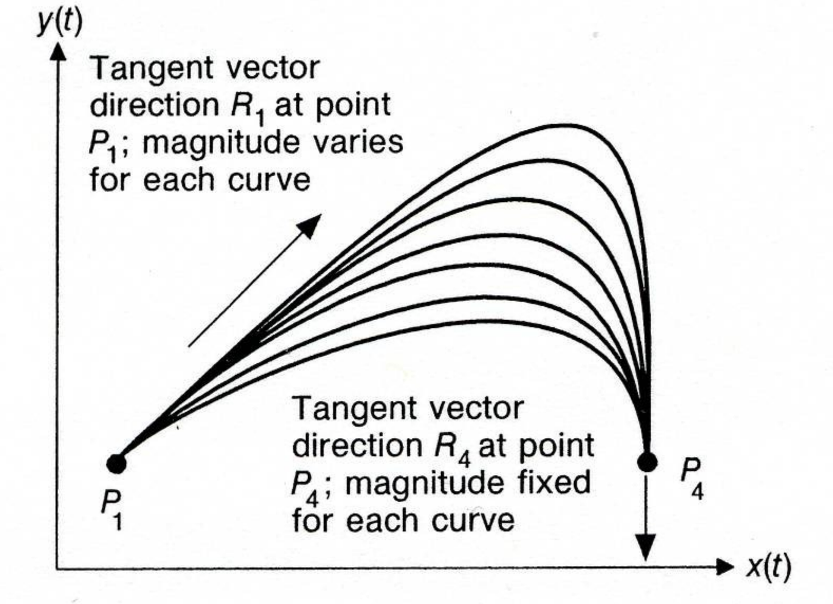
- **The polynomials $H_k(t)$ for k=0, 1, 2, 3 and referred to as blending functions because they blend the boundary constraint values to obtain each coordinate position along the curve.**

Hermite blending functions



- **At t=0, only the function H_0 is nonzero. As as t becomes greater than zero, all other blending functions begin to have an influence.**

Varying the magnitude of the tangent vector



- **If the directions of the tangent vectors are fixed, the longer the vectors, the greater their effect on the curve.**

Hermite basis matrix

$$\mathbf{p}(u) = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d}$$

$$\mathbf{p}(0) = \mathbf{d}$$

$$\mathbf{p}(1) = \mathbf{a} + \mathbf{b} + \mathbf{c} + \mathbf{d}$$

$$\mathbf{p}^u(0) = \mathbf{c}$$

$$\mathbf{p}^u(1) = 3\mathbf{a} + 2\mathbf{b} + \mathbf{c}$$

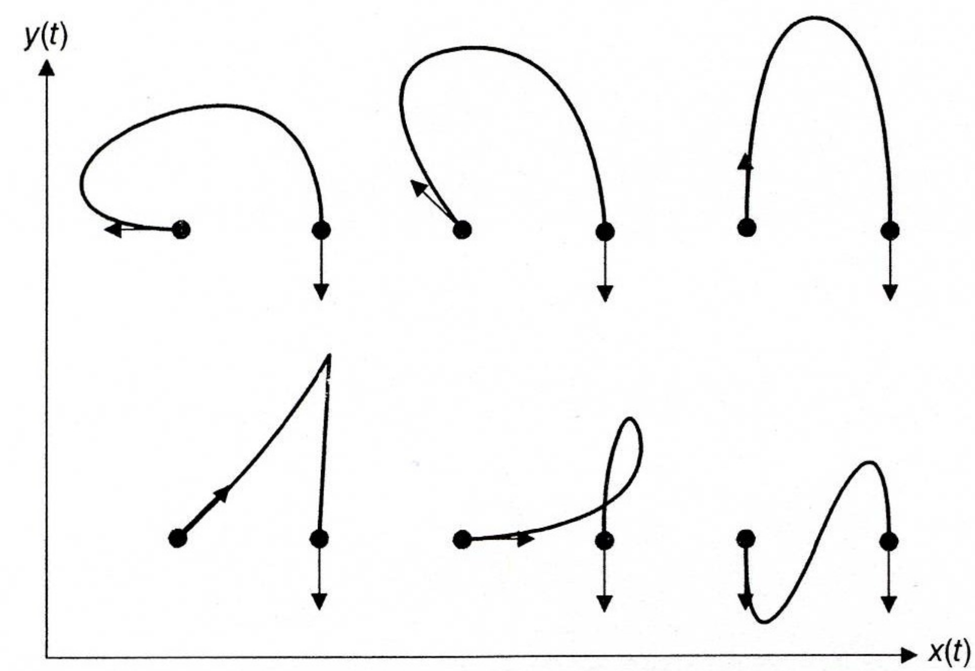
$$\mathbf{a} = 2\mathbf{p}(0) - 2\mathbf{p}(1) + \mathbf{p}^u(0) + \mathbf{p}^u(1)$$

$$\mathbf{b} = -3\mathbf{p}(0) + 3\mathbf{p}(1) - 2\mathbf{p}^u(0) - \mathbf{p}^u(1)$$

$$\mathbf{c} = \mathbf{p}^u(0)$$

$$\mathbf{d} = \mathbf{p}(0)$$

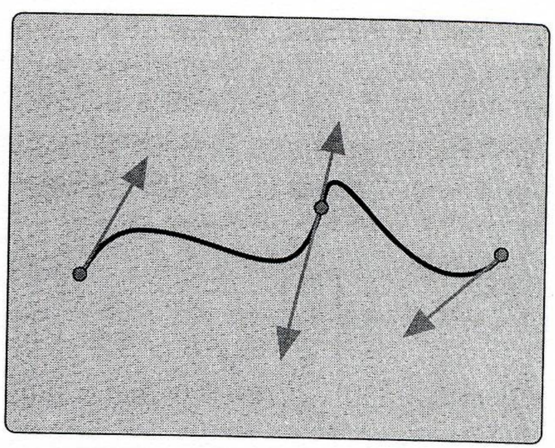
Varying the direction of the tangent vector

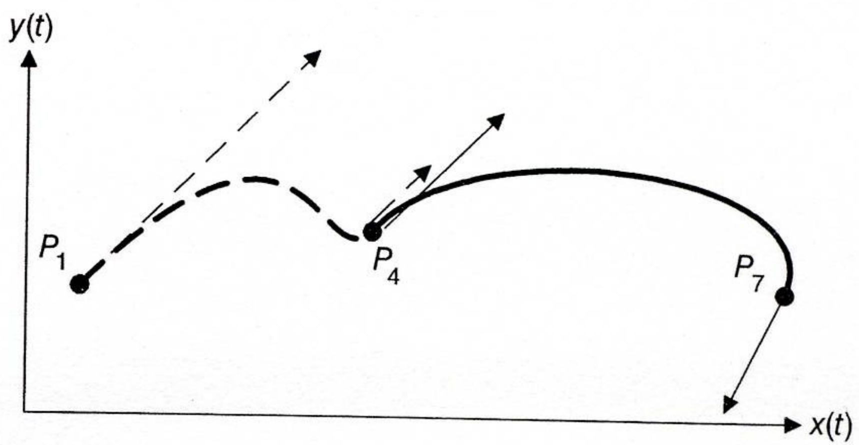


Obtaining geometric continuity G^1

$$\begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix} \text{ and } \begin{bmatrix} P_4 \\ P_7 \\ kR_4 \\ R_7 \end{bmatrix}, \text{ with } k > 0.$$

for parametric continuity C^1, $k = 1$





Hermite basis matrix

$$\mathbf{p}(u) = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d}$$

$$\mathbf{p}(0) = \mathbf{d}$$

$$\mathbf{p}(1) = \mathbf{a} + \mathbf{b} + \mathbf{c} + \mathbf{d}$$

$$\mathbf{p}^u(0) = \mathbf{c}$$

$$\mathbf{p}^u(1) = 3\mathbf{a} + 2\mathbf{b} + \mathbf{c}$$

$$\mathbf{a} = 2\mathbf{p}(0) - 2\mathbf{p}(1) + \mathbf{p}^u(0) + \mathbf{p}^u(1)$$

$$\mathbf{b} = -3\mathbf{p}(0) + 3\mathbf{p}(1) - 2\mathbf{p}^u(0) - \mathbf{p}^u(1)$$

$$\mathbf{c} = \mathbf{p}^u(0)$$

$$\mathbf{d} = \mathbf{p}(0)$$

Bezier Curve

http://www.cs.princeton.edu/~min/cs426/classes/bezier.html

Bezier Curves

- R1=3(P_2 - P_1), R4=3(P_4 - P_3) the Bezier curve interpolates the two end control points and approximates the other two. (Multiplay by 3 to have a constant velocity from P_1 to P_4)

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} M_h \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}_x = UM_h \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}_x$$

We can easily arrive at:

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

- **The curve is cotangent to the control polygon at these endpoints.**

- **Expanding the product gives the *Bernstein polynomials***

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

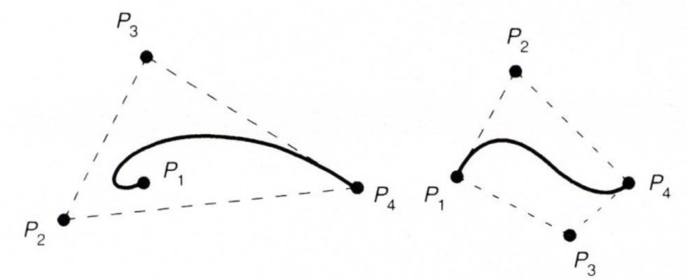
- **The sum of the 4 blending polynomials is everywhere unity and that each polynomial is everywhere nonegative:**

Q(t) is just a weighted average of the four control points -- each curve segment is completely contained in the convex hull of the 4 control points.

- **The convex-hull property holds for all cubics defined by weighted sums of control points if the blending functions are nonnegative & sum to one.**

Bézier curves
properties

- The convex hull property



- Partition of unity

$$\sum_i B_{i,n}(u) = 1$$

- Invariance under affine transformations

Bézier Curves of General Degree

- **Given n+1 control-point positions, we can blend them to produce the following:**

$$Q(t) = \sum_{k=0}^{n} P_k Bez_{k,n}(t) \quad 0 \le t \le 1,$$

Where the Bernstein functions:

$$Bez_{k,n}(t) = C(n,k) t^k (1-t)^{n-k}$$

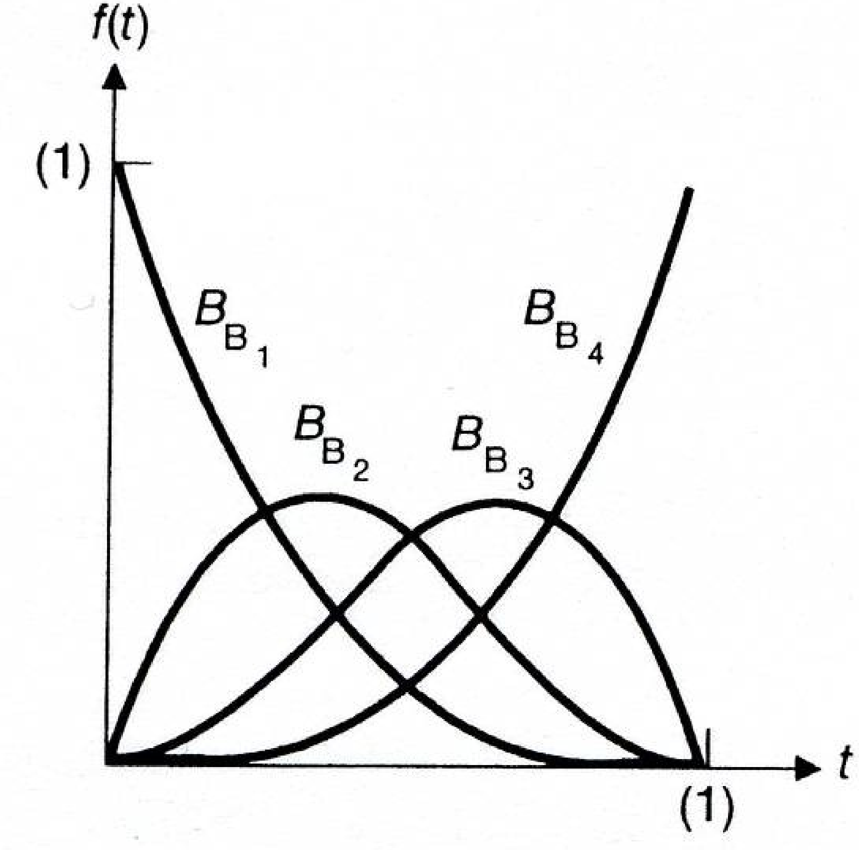
- **We know:** $(a+b)^n = \sum_{k=0}^{n} C(n,k) a^k b^{n-k}$

- **The binomial coefficient,** $C(n,k) = \dfrac{n!}{k!(n-k)!}$

is a closed form representation for an entry in Pascal's triangle:

$$C(n,k) = C(n-1,k-1) + C(n-1,k)$$

The Bernstein polynomials
n=3



The Bernstein polynomials

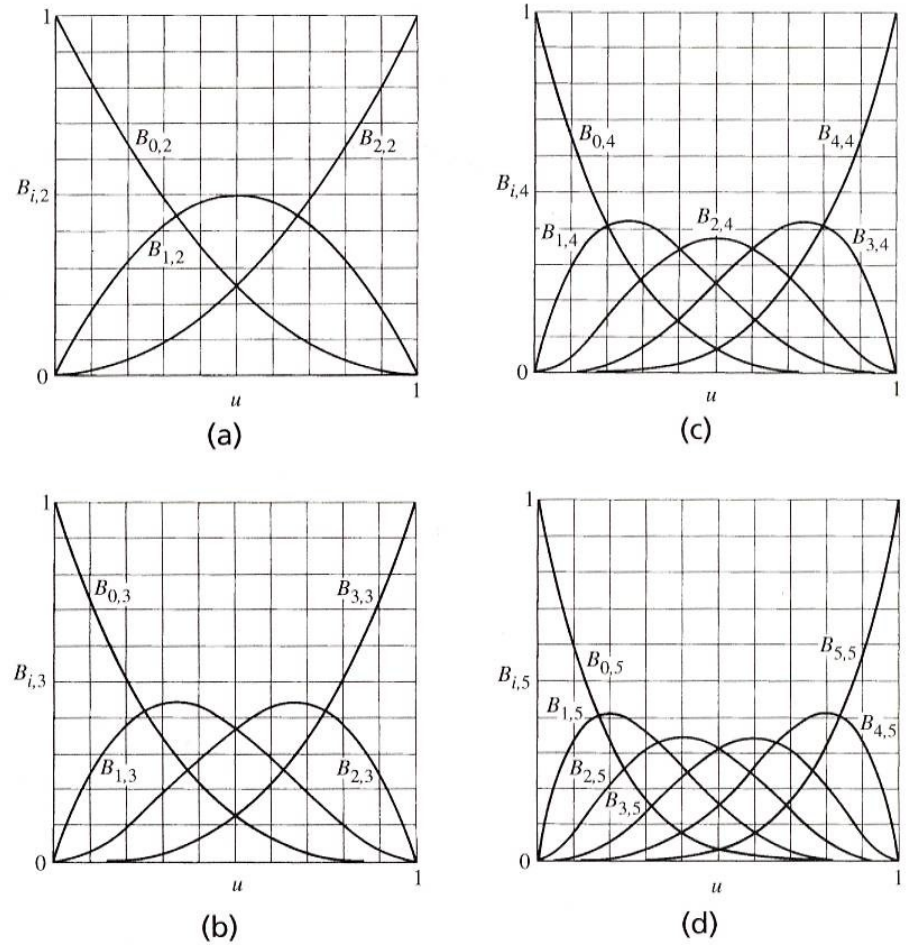
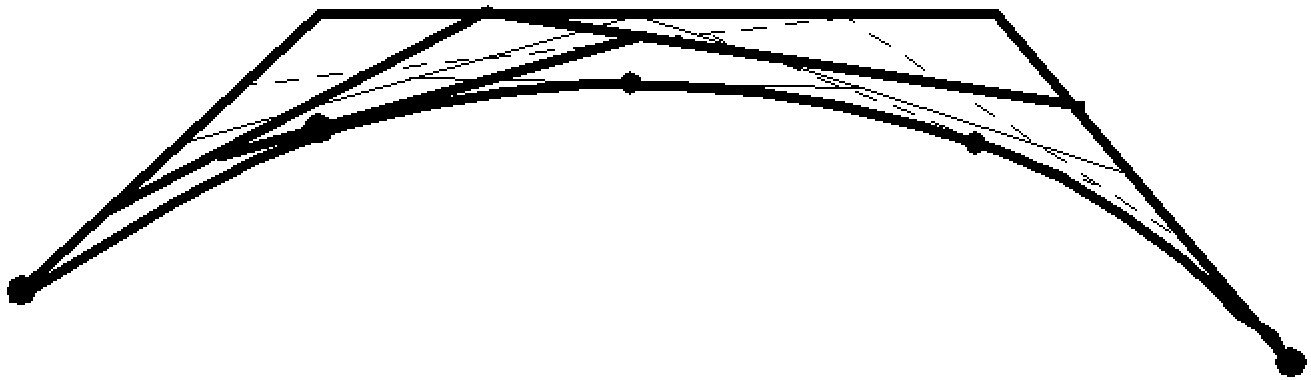


Figure 4.6 Bézier basis functions: (*a*) Three points, $n = 2$; (*b*) Four points, $n = 3$; (*c*) Five points, $n = 4$; (*d*) Six points, $n = 5$.

- **We can define recursive calculation:**

$$Bez_{k,n}(t) = (1-t)Bez_{k,n-1}(t) + tBez_{k-1,n-1}(t)$$

$$Where \;\; 1 \le k \le n$$



- **An important characteristic of the Bernstein functions is the *partition of unity*:**

$$\sum_{k=0}^{n} Bez_{k,n}(t) = 1$$

It is a key to understanding Bézier curves.

Bézier curves have a number of characteristics which define their behavior:

Endpoint Interpolation: the first and last points

Tangent Conditions: tangent to the first and last segments of the control polygon, at the first and last control points: $Q'(0) = (P_1 - P_0)n$ and $Q'(1) = (P_n - P_{n-1})n$

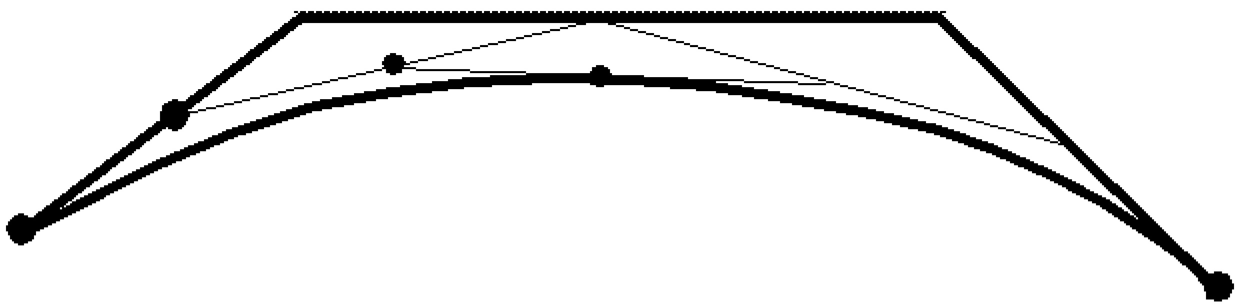
Convex Hull: contained in the convex hull of its control points for $0 \leq t \leq 1$

Affine Invariance: affinely invariant with respect to its control points. This means that any linear transformation (such as rotation or scaling) or translation of the control points defines a new curve which is just the transformation or translation of the original curve

Variation Diminishing: It does not wiggle any more than its control polygon; it may wiggle less.

Linear Precision: If all the control points form a straight line, the curve also forms a line. This follows from the convex hull property; as the convex hull becomes a line, so does the curve.

Subdivide of the Curve



The Derivative of the Bézier Curve

$$Q(t) = \sum_{k=0}^{n} P_k Bez_{k,n}(t), \qquad Q'(t) = n\sum_{k=0}^{n-1} (P_{k+1} - P_k) Bez_{k,n-1}(t)$$

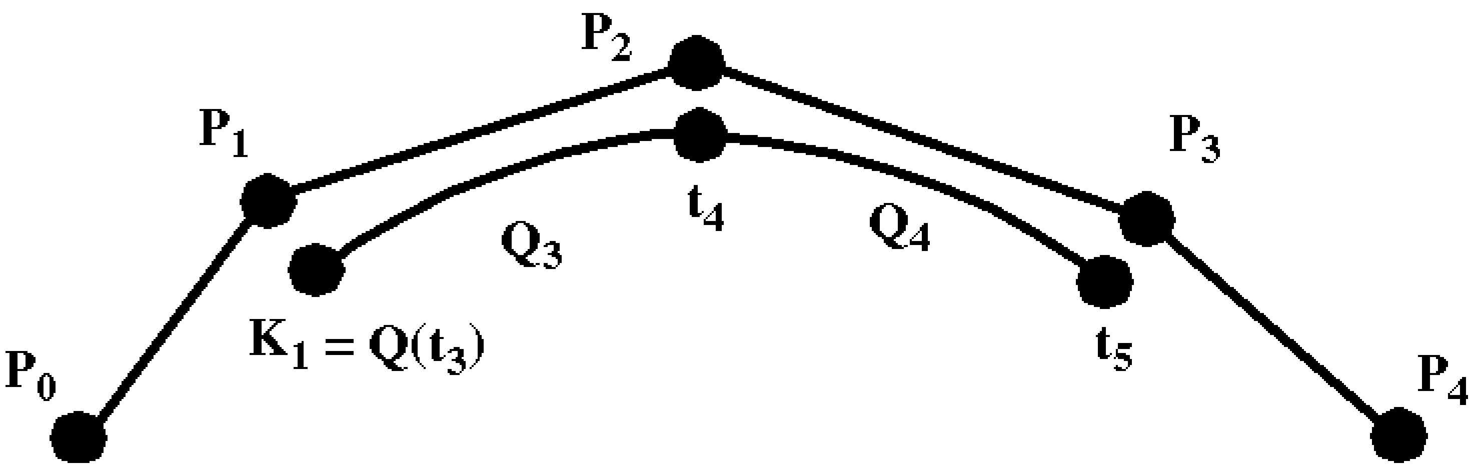
- **One less term in the derivative than in the original function; the degree of the Bernstein polynomials is one less.**
- **The control points for the derivative curve are successive differences of the original curve's control points.**

Natual Cubic Spline Curves

- Spline: a flexible strip used to produce a smooth curve through a designeated set of points.
- A natual cubic spline: cubic curve segments through the set of points
- C^2 continuity at the designeated control points
- Given n+1 control points, we have n curve sections and 4n polynomial coefficients
- We have 4n-2 equations to be satisfied by the 4n polynomial coefficients -- we need 2 more conditions: add a little assumption
- Disadvantages moving any one control point affects the entire curve; the computation time needed

Uniform Nonrational B-Splines

- Polynomial coefficients depend on just a few control points. (local control)
- Same continuity as natural splines but do not interpolate their control points.
- m+1 control points, $P_0, \ldots, P_m$, $m \geq 3$, m-2 cubic curve segments $Q_3, \ldots, Q_m$. For m=4:



- **The isotropic constraint is needed to find the curve equations**
- **Each curve is defined on its own domain $0 \le t < 1$. We can adjust the parameter ($t=t+k$) so that the parameter domains for the various curve segments are sequential: $t_i \le t < t_{i+1}$**
- **For each $i>3$, there is a joint point or knot between Q_{i-1} and Q_i at the parameter value t_i Uniform -- the knots are spaced at equal intervals of parameter *t***
- **Nonrational: not ratio of polynomials; rational: $x(t)=X(t)/W(t)$, $y(t)=Y(t)/W(t)$, $z(t)=Z(t)/W(t)$ are defined as ratio of two cubic polynomials.**
- **B - "basis." The spline is weighted sums of basis functions, in contrast to the natural splines.**

- **Each control point (except for those at the beginning and end) influences 4 curve segments**
- **The control points and knots are constrained by:** $6K_i = P_{i-1} + 4P_i + P_{i+1}$
- **A control point is used twice, then the curve is pulled closer to this point; used 3 times -- a line**

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} M_{Bs} \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}_x = T \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}_x$$

$t_i \le t < t_{i+1}, \; t_{i+1} - t_i = 1$

$3 < i \le m$

- **B-spline control points can be determined from** given knots, provided some additional info is given (tagent vectors at the 2 end knots)

Nonuniform, Nonrational B-Splines

- The parameter interval between successive knot values need not be uniform.
- The nonuniform knot-value sequence means that the blending functions are no longer the same for each curve interval
- Advantages: continuity at selected join points can be reduced from C^2 to C^1 to C^0 to none; the resulting curve can be easily reshaped
- If the continuity is reduced to C^0, then the curve interpolates a control point without the undesirable effect of uniform B-splines, where

the curve segments on either side of the interpolated control point are straight lines

- Starting & ending points can be interpolated exactly without introducing linear segments.
- Restriction: knot sequence is nondecreasing, which allows successive knot values equal: a sequence from t_0 to t_{m+4}. The curve is not defined outside the interval t_3 through t_{m+1}.
- When $t_i=t_{i+1}$ (a multiple knot), curve segment Q_i is a single point, which provides the extra flexibility of nonuniform B-splines; C^2 to C^1 for 1 extra knot, C^1 to C^0 for 2 extra knots, C^0 to no continuity for 3 extra knots (multiplicity 4)

$$Q_i(t) = P_{i-3}B_{i-3,4}(t) + P_{i-2}B_{i-2.4}(t) + P_{i-1}B_{i-1,4}(t) + P_iB_{i.4}(t)$$

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

Nonuniform, Rational Cubic Polynomial Curve Segments

- **General rational cubic curves are ratios of polynomials: x(t) = X(t)/W(t), y(t) = Y(t)/W(t), z(t) = Z(t)/W(t) whose control points are defined in homogeneous coordinates.**
- **Polynomials can be Bezier, Hermite, or any other type. When they are B-Splines, called NURBS**
- **We can also think of the curve as existing in homogeneous space: Q(t) = [X(t) Y(t) Z(t) W(t)]**
- **Useful for two reasons: 1) invariant under**

perspective transformation of the control points (nonrational curves are invariant under only rotation, scaling, and translations);

The alternative is first to generate points on the curve itself and then to apply the perspective transformation to each point, a far less efficient process

- **Useful: 2) Unlike nonrationals, they can define precisely any of the conic sections. This is useful in CAD, where general curves and surfaces as well as conics are needed. Both types of entities can be defined with NURBS.**