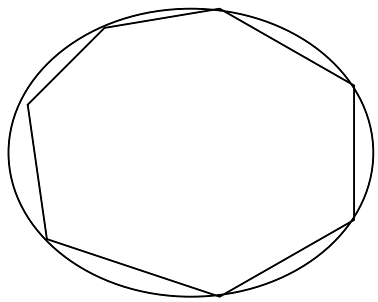
Curves

A line segment (or polyline, several connected line segments) can be used to approximate a curve.



However, when the curve to be approximated is smoother, it may take a large number of points to achieve a reasonable degree of accuracy and interactive manipulation of these points is tedious.

To achieve a more compact and manipulable representation of piecewise smooth curves, the linear functions are replaced by higher-order functions

Two ways to view a curve:

A. a very thin thread “frozen” in space

B. the path of a particle as it moves along the curve

A. Explicit Form

- Typically, in mathematics, a curve is presented as a graph of a function $Y = f(x)$
- As x is varied, $y=f(x)$ is computed by the function f and the pair of coordinates (x,y) sweeps out the curve

Problems with explicit form:

1. A function insists that there is one (and only one) value of y for each x and many curves to not fit this (e.g. circle) ◊
2. An explicit curve cannot have infinite slopes ◊
3. Any transformation such as rotation of shear may cause an explicit curve to violate the two points above.

Implicit Form

- $f(x,y) = 0$
- there may be more solutions than what we want
- need some way to specify parts of a curve

B. Parametric Form

- movement of a point through time
- called "parametric" because a parameter (frequently called t and interpreted as time) is used to distinguish points on the curve
- path fixed by two functions x(t) and y(t)
- as t varies, point (x(t), y(t)) sweep out the curve
- the curve is all points "visited" by the particle as t varies over some interval (say 0 – 1)

Example #1: line from (x1, y1) to (x2, y2)

x(t) = x1 + t(x2-x1)

y(t) = y1 + t(y2-y1)

when t=0 (x1, y1)

t=1 (x2, y2)

Example #2: circle x(t) = sin(t) y(t) = cos(t) ◊

- typically we deal with polynomial or rational functions (not trig)
- circle $x(t) = \frac{2t}{\left(1+t^2\right)}$ $y(t) = \frac{\left(1-t^2\right)}{\left(1+t^2\right)}$

- the motion of these two functions is different even if the path is the same

Consider the parametric curve:

$x(t) = 6t - 9t^2 + 4t^3$
$y(t) = 4t^3 - 3t^2$

A convenient notation is:

$$f(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} t + \begin{pmatrix} -9 \\ -3 \end{pmatrix} t^2 + \begin{pmatrix} 4 \\ 4 \end{pmatrix} t^3$$

These equations are the same. We simply save on notation by writing the "basis functions" only once. (here the basis functions are 1, t, t^2, and t^3)

Advantages of using parametric equations:

- usually have more degrees of freedom to control the curve
- parametric curves are not constrained to be single-valued along any line
- the slope of a parametric curve segment may be defined vertically

Parametric Derivatives

- the slope of a parametric curve is computed by finding the derivative vector (x'(t), y'(t)) at any point t. This vector determines the speed at which the point traces out the curve as t changes.

Example: ◊

- The parameter t moves the point (x(t), y(t)) along the path of the curve.
- The point's speed varies as t varies. The speed is higher at the ends of the curve.
- The derivative vector changes in length, reflecting the variation in the speed of the point.
- In the demonstration, the curve crosses itself, which can easily happen with parametric curves.

The derivative of our example curve would be:

$$f'(t) = \begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} = \begin{pmatrix} 6 - 18t + 12t^2 \\ 12t^2 - 6t \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} + \begin{pmatrix} -18 \\ -6 \end{pmatrix} t + \begin{pmatrix} 12 \\ 12 \end{pmatrix} t^2$$

The derivative function is itself a parametric curve of degree one less that the original curve.

The derivative curve is called the "hodograph" ◊

Parametric Cubic Curves

- curves are approximated by piecewise polynomial curves
- each curve segment is given by three functions, x, y, z which are cubic polynomials in the parameter t
- the x, y, and z coordinates are represented by a 3^{rd}-order (i.e. cubic) polynomial of some parameter t
- p(t) = (x(t), y(t), z(t))
- cubic polynomials are used because lower-degree polynomials give too little flexibility

algebraic form

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$
$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$
$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$$

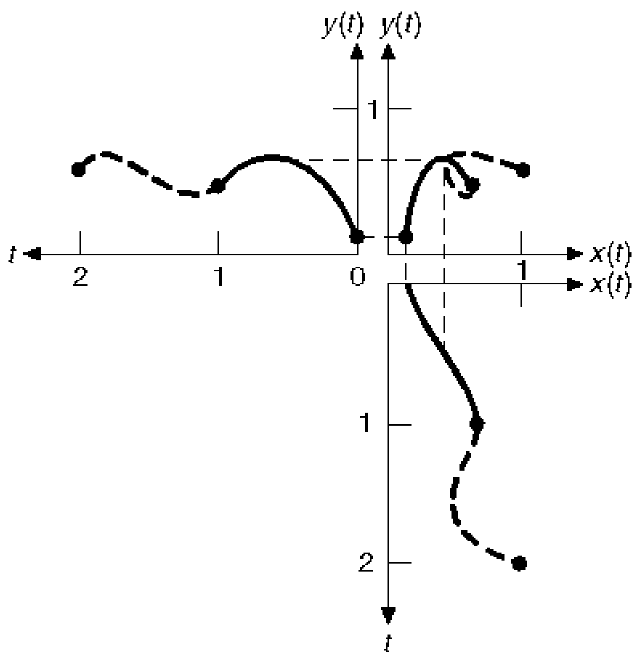
we restrict t to the [0,1] interval

$T = [t^3\ t^2\ t\ 1]^T$ and coefficient matrix: $C = \begin{bmatrix} ax & bx & cx & dx \\ ay & by & cy & dy \\ az & bz & cz & dz \end{bmatrix}$

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

Plotting parametric curves

- usually the independent variable is plotted on the x-axis and the dependent variable on the y-axis
- for parametric curves, the dependent variable t is not plotted at all
- therefore, it isn't possible to determine the tangent vector just by looking at the plot
- two curves with the same plot may have different tangent vectors



Continuity

- for explicit functions, describes when a curve does not break or tear. If it meets these conditions, it is described as C_0.
- C_0 is defined by the popular description "a curve is continuous if it can be drawn without lifting the pencil from the paper.
- If the derivative of the curve is also continuous, then the curve is first-order differentiable and is said to be C_1 continuous.
- Practically, this means that a C_1 continuous curve will not kink.
- Higher degrees of continuity imply a smoother curve. ◊

- Unfortunately, continuity does not always result in the expected smoothness when viewed parametrically. The coordinate functions (such as x(t), y(t) and z(t)) may be first-order differentiable and still kink.
- All that continuity guarantees for parametric curves is that the motion of the particle is smooth, there are no sudden jumps in velocity. It does not say that the path of the particle is smooth

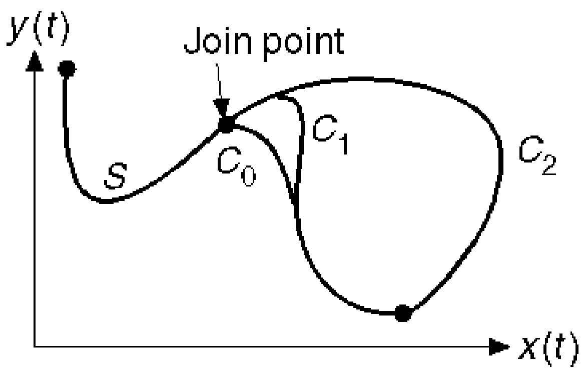
- Geometric continuity
  - is a notation that immediately tells the designer whether or not the curve is smooth
  - defined in terms of the directions of tangent vectors only
  - G_0: endpoints meet
  - G_1: direction of tangent vectors agree

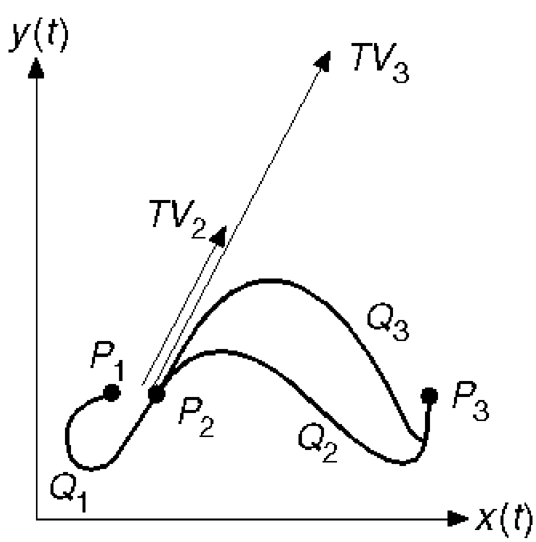
- Parametric continuity (C)
  - Defined in terms of direction and magnitude of tangent vectors
  - C_0: endpoints meet
  - C_1: tangent vectors (velocity) are equal
  - C_2: derivative of tangents (acceleration) are equal

Examples:

1. A particle travels with C_1 and G_1 continuity
   - a car makes a smooth turn on a road
2. A particle travels with C_1 but not G_1 continuity
   - a car slows to a stop sign, turns its wheels, then speeds up again in a different direction
3. A particle travels with G_1 but not C_1 continuity
   - a baton is passed from one runner to a faster runner who stay in the same lane on the track

- If a curve is C_0, it is G_0 continuous, no gaps in curve
- G_1 continuity, the directions but not necessarily the magnitudes of the tangent vectors are the same
- C_1 continuity, the tangent vectors (directions and magnitudes) are the same
- Usually C_1 implies G_1 but the converse is generally not true





Constraints on a curve

A curve segment is defined by constraints on endpoints, tangent vectors and continuity between segments. Each cubic polynomial has 4 coefficients so 4 constraints are needed.

Three major types of curves:

- Hermite: defined by 2 endpoints and 2 endpoint tangents
- Bezier: defined by 2 endpoints and 2 other points which control endpoint tangents
- B-spline (uniform, non-rational): defined by a series of control points

Each curve is defined in the same general way:

Consider the coefficient matrix C as the product of a geometry matrix G and a basis matrix M.

$$Q(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} G1 & G2 & G3 & G4 \end{bmatrix} \bullet \begin{bmatrix} m11 & m21 & m31 & m41 \\ m12 & m22 & m32 & m42 \\ m13 & m23 & m33 & m43 \\ m14 & m24 & m34 & m44 \end{bmatrix} \bullet \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

The curve is basically a weighted sum of the elements of the geometry matrix. The weights are each cubic polynomials of t and are called "blending functions". The blending function B are given by B = M • T

Hermite Curves

We impose the following constraints through the geometry matrix:

$G_H = [P_1\ P_4\ R_1\ R_4]$

Which indicates that the four columns (constraints) will be specified by two endpoints (P_1 and P_4) and two tangent vectors (R_1 and R_4).

$G_{Hx} = [P_{1x}\ P_{4x}\ R_{1x}\ R_{4x}]$

$$\mathrm{x(t)} = \mathrm{a_x t^3 + b_x t^2 + c_x t + d_x = C_x} \cdot T = G_{HX} \cdot M_H \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$\mathrm{x}(0)=\mathrm{P}_{1\mathrm{x}} = \mathrm{G}_{\mathrm{Hx}} \cdot M_H \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \mathrm{x}(1)=\mathrm{P}_{4\mathrm{x}} = \mathrm{G}_{\mathrm{Hx}} \cdot M_H \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathrm{x}'(0)=\mathrm{R}_{1\mathrm{x}} = \mathrm{G}_{\mathrm{Hx}} \cdot M_H \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \mathrm{x}'(1)=\mathrm{R}_{4\mathrm{x}} = \mathrm{G}_{\mathrm{Hx}} \cdot M_H \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

$$[P_{1x} \quad P_{4x} \quad R_{1x} \quad R_{4x}] = G_{Hx} = G_{Hx} \cdot M_H \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

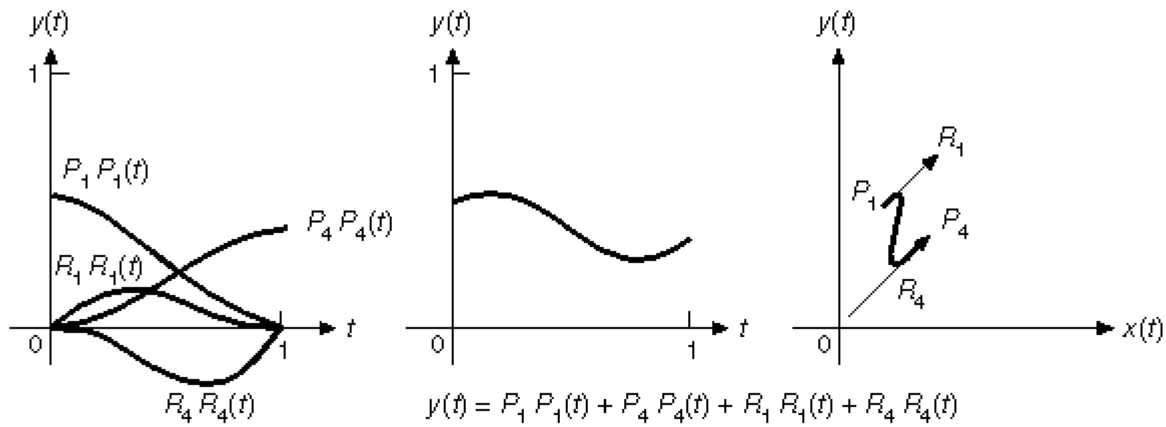
$$M_H = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

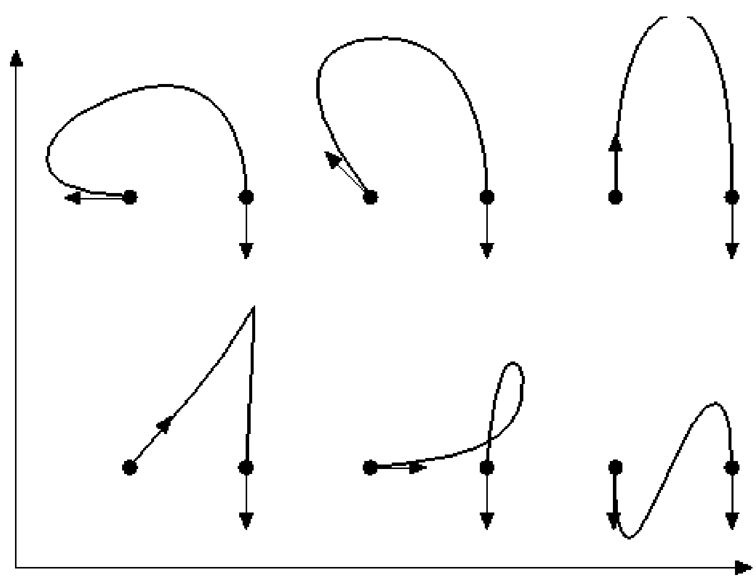
Blending functions for a Hermite curve

$Q(t) = G_H \cdot M_H \cdot T$ (and combining MH and T into the Blending function)

$= G_H\, B(t)$

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

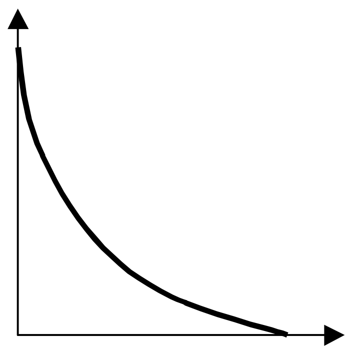




Bezier curves

Mathematical properties of the Bezier curve

Consider the parabola that passes through (0,1) and (1,0) and is tangent to the x and y axes at these points.



$f(t) = at^2 + bt + c$

where a, b and c are vector coefficients. f(t) is a vector function with two components, f(t) = (x(t), y(t)).

The above parabola can we written as:

$$f(t) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} t^2 + \begin{pmatrix} -2 \\ 0 \end{pmatrix} t + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

And can be re-written as:

$$f(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} (1-t)^2 + \begin{pmatrix} 0 \\ 0 \end{pmatrix} 2t(1-t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} t^2$$

Why?

Control points ((0,1) (0,0) (1,0) which together form the control polygon.

Bezier Curve

- Specify the tangent vectors for the endpoints indirectly with two points that are not on the curve.
- The tangent vectors are determined by the vectors P_1P_2 and P_3P_4, which are related to R_1 and R_4 in the following way:

$$R_1 = Q'(0) = 3(P_2 - P_1)$$
$$R_4 = Q'(1) = 3(P_4 - P_3)$$

The Bezier geometry matrix G_B is:

$$G_B = [P_1\ P_2\ P_3\ P_4]$$

Then, the matrix M_{HB} that defines the relation $G_H = G_B \cdot M_{HB}$ between the Hermite geometry matrix and the Bezier geometry matrix is just:

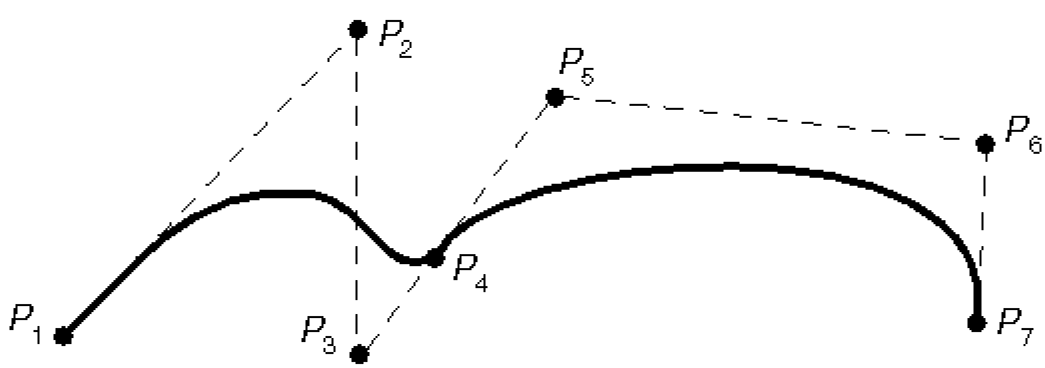
$$G_H = \begin{bmatrix} P_1 & P_4 & R_1 & R_4 \end{bmatrix} = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{bmatrix} = G_B \cdot M_{HB}$$

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

The curve is then defined by the following equation:

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

- Curves will join with G_1 continuity if $(P_3 - P_4) = k(P_4 - P_5)$, k > 0. (i.e. the three points must be distinct and collinear)
- In the more restrictive case of k=1, there is C_1 continuity in addition to G_1 continuity.



Characteristics of the Bezier Curve ◊

- Endpoint interpolation
  - f(0) = b0 and f(n) = b(n)
- Tangent conditions
  - Bezier curve is tangent to the first and last segments of the control polygon at the first and last control points
  - f'(0) = $(b_1-b_0)n$ and f'(1) = $(b_n-b_{n-1})n$ where n is a constant
- convex hull
  - the curve is contained in the convex hull of its control points for 0<=t<=1
  - the convex hull of a control polygon is the minimal convex enclosure of the control polygon
- affine invariance
  - any linear transformation (such as rotation or scaling) or translation of the control points defines a new curve that is just the transformation of translation of the original curve
- linear precision
  - if all the control points form a straight line, the curve also forms a straight line

OpenGL uses a Bezier basis for its curves and it provides function calls to efficiently evaluate the curve for drawing.