Chapter 5

Curve Modeling

Introduction

- A curve is an infinitely large set of points. Each point has two neighbors except endpoints. Curves can be broadly classified into three categories: **explicit, implicit,** and **parametric curves**.
- ***Implicit Curves***
  - Implicit curve representations ***define the set of points on a curve by employing a procedure*** that can test to see if a point in on the curve.
  - Usually, an implicit curve is defined by an implicit function of the form f(x, y)=0 ; in two dimensions and f(x, y, z)=0 ; in three dimensions
- ***Explicit Curves***
  - ***Explicit function:*** to express the idea that we have one dependent variable on the left-hand side of an equation, and all the independent variables and constants on the right-hand side of the equation.
  - For example, ***the equation of a line is: y=mx+b***
  - A mathematical function ***y = f(x) can be plotted as a curve***. Such a function is the explicit representation of the curve.
  - For each value of x, only a single value of y is normally computed by the function.

Introduction

- **Parametric *Curves***
  - Parametric equations ***are generators of paths through space.***
  - Curve descriptions ***provide a mapping from a free parameter to the set of points on the curve.*** The parametric form of a curve defines a function that assigns positions to values of the free parameter.

    •We write the coordinate ***pair (x, y) as a pair of functions (r cos(t), r sin(t)).***

    • ***Separate equation for each spatial variable***

    x=x(u) y=y(u) z=z(u)

    p(u)=[x(u), y(u), z(u)]T

Introduction

Why Parametric Cubic Curves?

- Parametric representations are the most common in computer graphics.
- A curve is approximated by a piecewise polynomial curve.
- ***Cubic polynomials are most often used because:***

  1) Lower-degree polynomials offer too little flexibility in controlling the shape of the curve.

  2) Higher-degree polynomials can introduce unwanted wiggles and also require more computation.

Polynomial functions

- ***Linear:*** $f(t) = at + b$

- ***Quadratic:*** $f(t) = at^2 + bt + c$

- ***Cubic:*** $f(t) = at^3 + bt^2 + ct + d$

Parametric Representation of a curve

The cubic polynomial that define a curve segment is

$$\mathbf{Q(t) = [x(t)\ \ y(t)\ \ z(t)]}$$

Where

$$\mathbf{x(t) = a_x t^3 + b_x t^2 + c_x t + d_x}$$

$$\mathbf{y(t) = a_y t^3 + b_y t^2 + c_y t + d_y}$$

$$\mathbf{z(t) = a_z t^3 + b_z t^2 + c_z t + d_z}$$

$$\mathbf{0 \le t \le 1}$$

To represent this in matrix form, we have

$$\mathbf{T = [t^3\ \ t^2\ \ t\ \ 1]}$$

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

So, we can write

$$\mathbf{Q(t) = T.C}$$

Spline Representation

- A spline is a ***flexible strip used to produce a smooth curve*** through a designated **set** of points.
- In computer graphics, the term spline curve refers to any composite curve formed with polynomial section satisfying specified continuity conditions at the boundary of the pieces.
- Splines are used in graphics applications ***to design curve and surface shapes***, to digitize drawings for computer storage, ***and to specify animation paths for the objects or the camera in a scene.***
- Typical CAD applications for splines include the design of automobile bodies, aircraft and spacecraft surfaces, and ship hulls.
- We specify a ***spline curve by giving a set of coordinate positions***, called control points, which indicates the general shape of the curve These, ***control points are then fitted with piecewise continuous parametric polynomial functions***.

Spline Representation

When polynomial sections ***are fitted so that the curve passes through each control point***, as in Figure 1**,** the resulting curve is ***said to interpolate the set of control points.***

▪ On the other hand, when the ***polynomials are fitted to the general control-point path without necessarily passing through any control point***, the resulting curve is ***said to approximate the set of control points*** (Figure 2**).**

Figure 1: A set of six control points interpolated with piecewise Continuous polynomial sections.

Figure 2: A set of six control points approximated with piecewise Continuous polynomial sections.

Parametric Continuity Conditions

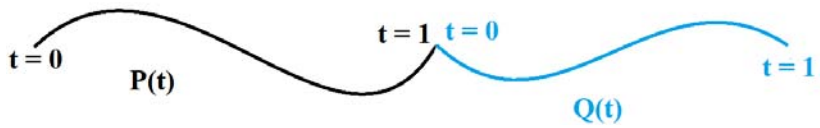
- To ensure a smooth transition from one section **of** a piecewise parametric **curve** to the next, can impose various **continuity conditions** at the connection points.
- If each ***section of a spline is described with a set of parametric coordinate*** functions of the form

$$x=x(u),\ y=y(u),\ z=z(u)\ u_1 \le u \le u_2$$

- Set parametric continuity ***by matching the parametric derivatives of adjoining curve sections*** at their common boundary.

Zero-order Parametric Continuity (C^0): If two curve ***segments join together.***

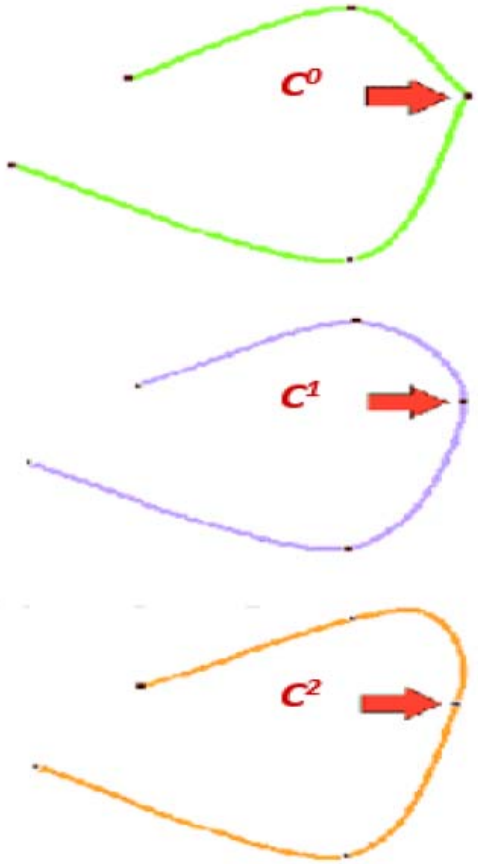
$$P(1) = Q(0)$$



First-order Parametric Continuity (C^1): First derivatives equal. If the ***directions and magnitudes*** of the two segments' tangent vectors are ***equal at a join point.*** $P'(1) = Q'(0)$

Second-order Parametric Continuity (C2): First and second derivatives are equal. The first and second parametric derivatives of the two curve sections are the same at the intersection. If t is taken to be time, this implies that the acceleration is continuous. $P''(1) = Q''(0)$

Parametric Continuity Conditions



Geometric Continuity Conditions

- An alternative methods for joining ***two successive curve sections*** is to specify conditions ***for geometric continuity:*** only require ***parametric derivatives of the two sections to be proportional to each other at their common boundary*** instead of equal to each other.
-

- **Zero order geometric (G^0):** Same as C^0

$$C_1(1) = C_2(0)$$

- **First order geometric (G^1):** The parametric first derivatives are proportional at the intersection of two successive sections.

$$C'_1(1) = \alpha C'_2(0)$$

- **Second order geometric (G^2):** Both the first and second parametric derivatives of the two curve sections are proportional at their boundary,

$$C''_1(1) = \alpha C''_2(0)$$

Spline Specification

There are three equivalent ***methods for specifying a particular spline representation:***

1) We can state the ***set of boundary conditions that are imposed on the spline***; or

2) We can ***state the matrix that characterizes the spline***; or

3) We can ***state the set of blending functions (or basis functions)*** that determine how specified geometric constraints on the curve are combined to calculate positions along the curve path.

We have the following ***parametric cubic polynomial*** representation for the ***x coordinate along the path of a spline section:***

$$x(t) = \sum_{i=0}^{n} a_i t^n \qquad ; \; 0 \leq t \leq 1$$

y(t) and z(t) are similar and each is handled independently.

i.e.

$$x(t) = a_3t^3 + a_2t^2 + a_1t + a_0 \qquad x'(t) = 3a_3t^2 + 2a_2t + a_1$$

$$y(t) = b_3t^3 + b_2t^2 + b_1t + b_0 \quad \text{AND} \quad y(t) = 3b_3t^2 + 2b_2t + b_1$$

$$z(t) = c_3t^3 + c_2t^2 + c_1t + c_0 \qquad z(t) = 3c_3t^2 + 2c_2t + c_1$$

Spline Specification

A ***compact version of the parametric equations*** can be

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

$$x(t) = T \cdot A$$

Similarly, we can write

$$y(t) = T \,.\, B$$

$$z(t) = T \,.\, C$$

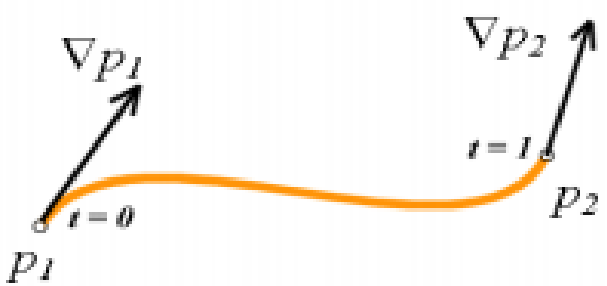
Each dimension is treated independently, so we can deal with curves in any number of dimensions.

Hermite Spline Curve

- Hermite is an interpolating ***piecewise cubic polynomial with a specified tangent at each control point.***
- If **P(u)** represents a ***parametric cubic point function*** for the curve section between control ***points P_k and P_{k+1}***, then the ***boundary conditions*** that define ***this Hermite curve*** section are

$$P(0) = P_k$$
$$P(1) = P_{k+1}$$
$$P'(0) = DP_k$$
$$P'(1) = DP_{k+1} \quad \text{.....eqn(1)}$$

with DP_k and DP_{k+1} specifying the values for the parametric derivatives (slope of the curve) a t control points P_k and P_{k+1} respectively.



Hermite Specification

- We can write the Hermite curve section as

$$P(t) = at^3 + bt^2 + ct + d; \; 0 \le t \le 1 \quad \text{.....eqn.(2)}$$

where the **x** component of P is $x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$, and similarly for the y and z components.

- The ***Matrix equivalent of eqn(2) is***

Hermite Spline Curve

- The ***Matrix equivalent of eqn(2) is***

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

- The ***derivative of the point function*** can be expressed as

$$P'(t) = \begin{bmatrix} 3t^2 & 2t & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

- Substituting ***endpoint values 0 and 1 for parameter t*** Into the previous two equations, we can express the ***Hermite boundary conditions eqn(1)***

$P_k = P(0) = d$ $\qquad$ $DP_k = P'(0) = c$

$P_{k+1} = P(1) = a + b + c + d$ $\qquad$ $DP_{k+1} = P'(1) = 3a + 2b + c$

Hermite Spline Curve

- ***In the matrix form:***

$$\begin{bmatrix} P_k \\ P_{k+1} \\ DP_k \\ DP_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

- ***And the solution is:***

$d = P_k$

$b = -3\ P_k - 2\ DP_k + 3\ P_{k+1} - DP_{k+1}$

$c = DP_k$

$a = 2\ P_k + DP_k - 2\ P_{k+1} + DP_{k+1}$

Hermite Spline Curve

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = M_H \cdot \begin{bmatrix} P_k \\ P_{k+1} \\ DP_k \\ DP_{k+1} \end{bmatrix}$$ Where M_H is Hermite matrix.

- ***The eqn(3) can be written as:*** $P(t) = [\, t^3 \;\; t^2 \; t \; 1 \,]\; M_H \cdot \begin{bmatrix} P_k \\ P_{k+1} \\ DP_k \\ DP_{k+1} \end{bmatrix}$

$$P(t) = [\, t^3 \;\; t^2 \; t \; 1 \,] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ DP_k \\ DP_{k+1} \end{bmatrix}$$

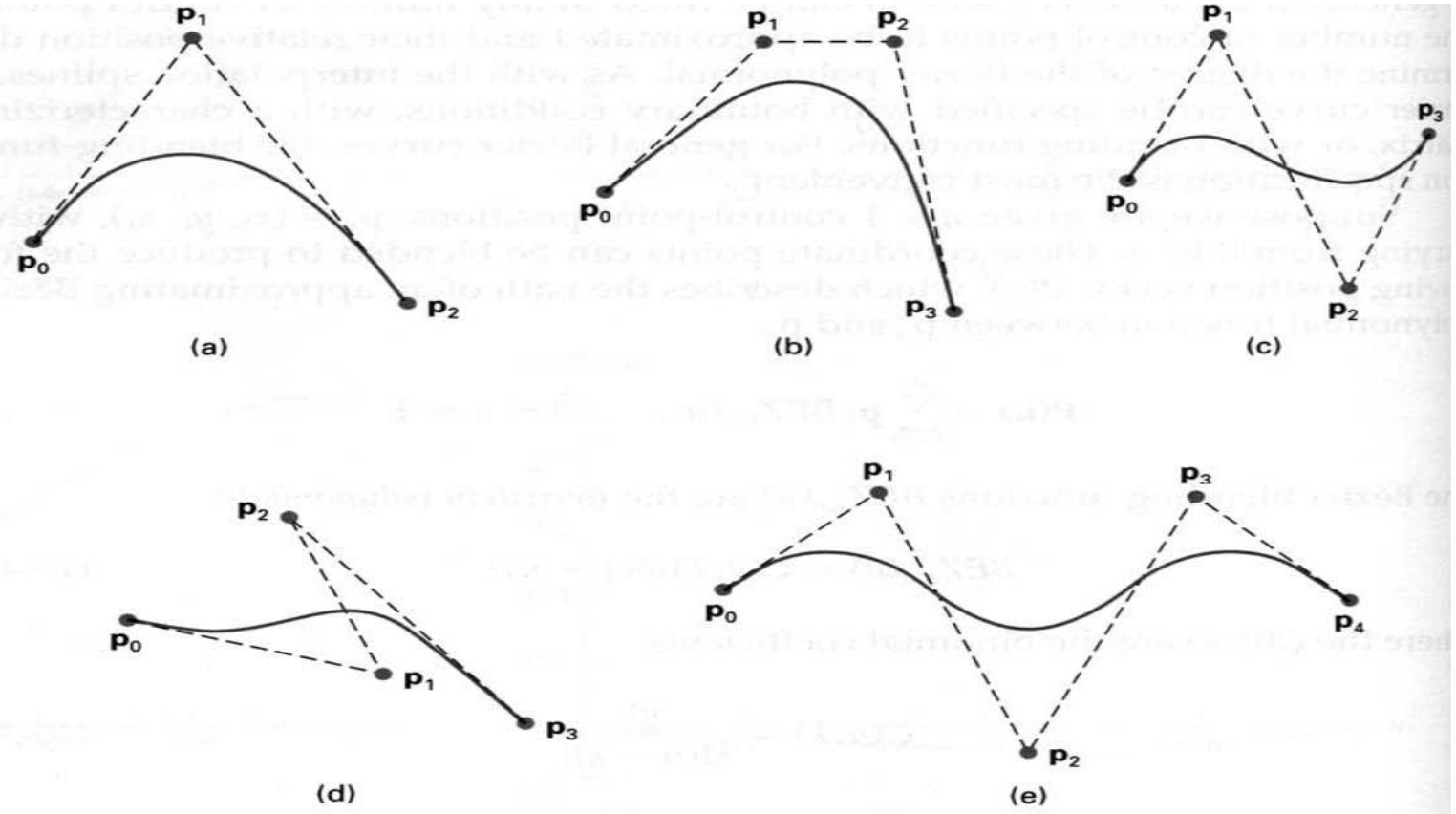
Bezier Curve

- This spline approximation method was developed by the French engineer Pierre Bezier for use in the design of Renault automobile bodies.

- Bezier splines have a number of properties that make them highly useful and convenient for curve and surface design and they are also easy to implement. For these reasons, Bezier splines are widely available in various CAD systems.
- In general, a Bezier curve section can be fitted to any number of control points.
- The number of control points to be approximated and their relative position determine the degree of the Bezier polynomial.

- ***A Bezier curve is a polynomial of degree one less than the designated number of control points.***

Bezier Curve



- ***Figure: Examples of two-dimensional Bezier curves generated from three, four, and five control points. Dashed lines connect the control-point positions.***

Bezier Curve

- ***Given n+1 control point positions:*** $\mathbf{p}_k = (x_k, y_y, z_k) \qquad 0 \le k \le n$

- These coordinate points can be blended ***to produced the following position vector p(u),*** which describes the path of an approximating Bezier polynomial function between $\mathbf{P_0}$ **and** $\mathbf{P_n}$.

$$p(u) = \sum_{k=0}^{n} \mathbf{p}_k B_{k,n}(u), \quad 0 \le u \le 1$$

- ***The Bezier blending functions are the Bernstein polynomials:***

$$B_{k,n}(u) = C(n,k)u^k(1-u)^{n-k}$$

$$C(n,k) = \frac{n!}{k!(n-k)!}$$

Bezier Curve

- Cubic Bezier curves are ***generated with four control points***. ***Bezier polynomial function between P_0 and P_3***

$$p(u) = \sum_{k=0}^{3} \mathbf{p}_k B_{k,3}(u), \quad 0 \le u \le 1$$

- **Where**

$$B_{k,3}(u) = C(3,k)u^k(1-u)^{3-k}$$

$$B_{0,3}(u) = (1-u)^3$$
$$B_{1,3}(u) = 3u(1-u)^2$$
$$B_{2,3}(u) = 3u^2(1-u)$$
$$B_{3,3}(u) = u^3$$

Bezier Curve

$$p(u) = \sum_{k=0}^{3} \mathbf{p}_k B_{k,3}(u), \quad 0 \le u \le 1$$

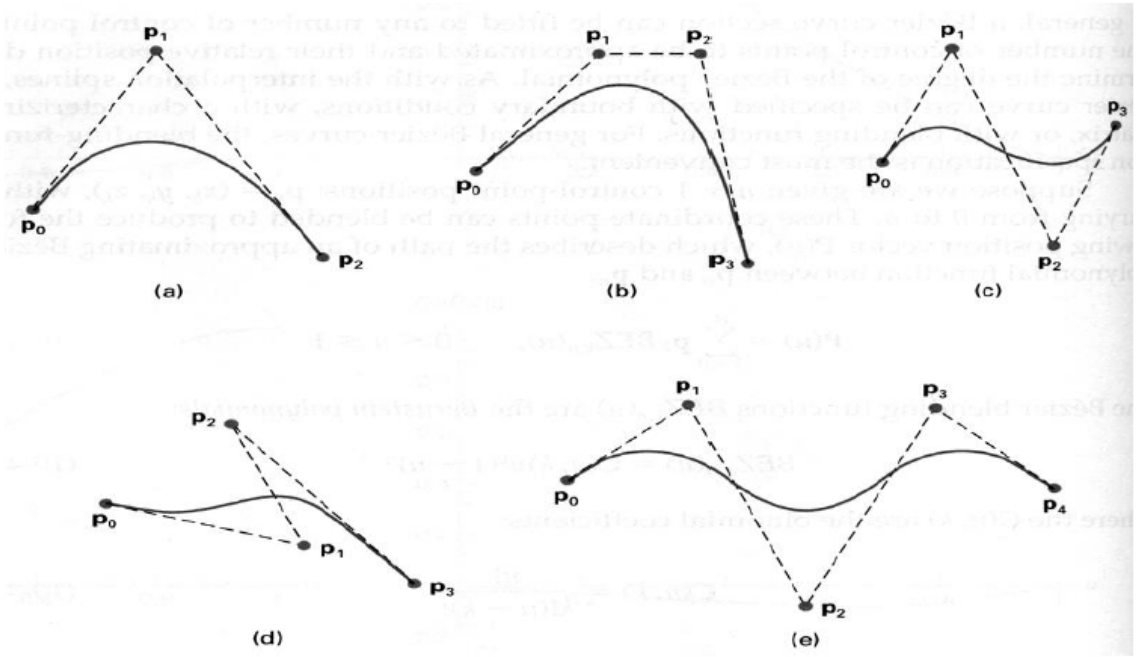
$$P(u) = (1-u)^3 P_0 + 3u(1-u)^2 P_1 + 3u^2(1-u) P_2 + u^3 P_3$$

$$p(u) = \begin{bmatrix} (1-u)^3 & 3u(1-u)^2 & 3u^2(1-u) & u^3 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$P(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

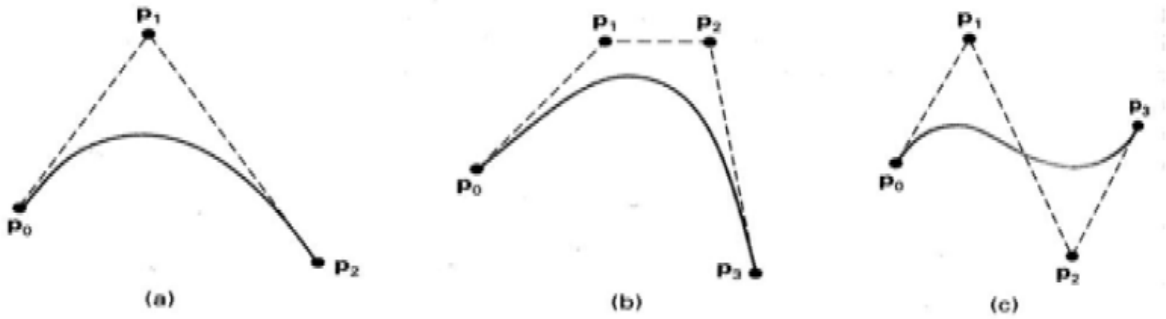
Properties Bezier Curve

1) The ***degree of a Bezier curve*** defined by ***n+1 control points is n***.
2) Bezier curve is that it always passes ***through the first*** P_0 *and* ***last control points*** P_n ; this is the so-called ***endpoint interpolation property.***



Properties Bezier Curve

3) The Bezier curve lies completely in the **convex hull** of the given control points.



4) All basis functions are positive and their sum is always 1.

$$\sum_{k=0}^{n} B_{k,n}(u) = 1$$

1) Find the coordinates at U=0.25, 0.5, and 0.75 with respect to the control points (10, 10), (15, 25), (20, 30), and (25, 5) using Bezier function. Draw your curve with given control points**.**