Unit 4

Syllabus:----Curves and Surfaces: Quadric surfaces, Spheres, Ellipsoid, Blobby objects, Introductory concepts of Spline, Bspline and Bezier curves and surfaces.

QUADRIC SURFACES

A frequently used class of objects are the *quadric surfaces*, which are described with second-degree equations (quadratics). They include spheres, ellipsoids, tori, paraboloids, and hyperboloids. Quadric surfaces, particularly spheres and ellipsoids, are common elements of graphics scenes, and they are often available in graphics packages as primitives from which more complex objects can be constructed.

Sphere

In Cartesian coordinates, a spherical surface with radius r centered on the coordinate origin is defined as the set of points (x, y, z) that satisfy the equation

$$x^2 + y^2 + z^2 = r^2 \tag{10-7}$$

We can also describe the spherical surface in parametric form, using latitude and longitude angles (Fig. 10-8):

$$\begin{aligned} x &= r \cos\phi \cos\theta, & -\pi/2 \le \phi \le \pi/2 \\ y &= r \cos\phi \sin\theta, & -\pi \le \theta \le \pi \\ z &= r \sin\phi \end{aligned} \tag{10-8}$$

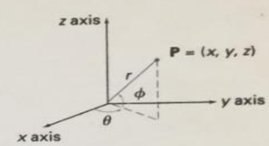


Figure 10-8
Parametric coordinate position (r, θ, ϕ) on the surface of a sphere with radius r.

The parametric representation in Eqs. 10-8 provides a symmetric range for the angular parameters θ and ϕ. Alternatively, we could write the parametric equations using standard spherical coordinates, where angle ϕ is specified as the colatitude (Fig. 10-9). Then, ϕ is defined over the range $0 \le \phi \le \pi$, and θ is often taken in the range $0 \le \theta \le 2\pi$. We could also set up the representation using parameters u and v defined over the range from 0 to 1 by substituting $\phi = \pi u$ and $\theta = 2\pi v$.

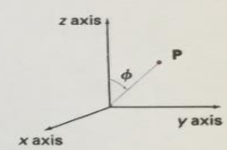


Figure 10-9
Spherical coordinate parameters (r, θ, ϕ), using colatitude for angle ϕ.

Ellipsoid

An ellipsoidal surface can be described as an extension of a spherical surface, where the radii in three mutually perpendicular directions can have different values (Fig. 10-10). The Cartesian representation for points over the surface of an ellipsoid centered on the origin is

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1 \tag{10-9}$$

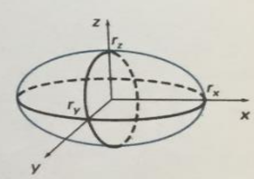
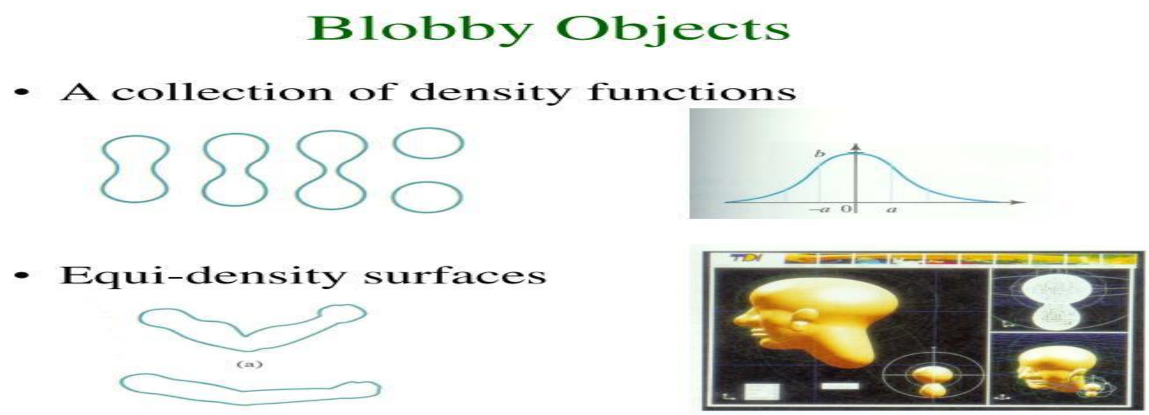


Figure 10-10
An ellipsoid with radii r_x, r_y, and r_z centered on the coordinate origin.

And a parametric representation for the ellipsoid in terms of the latitude angle ϕ and the longitude angle θ in Fig. 10-8 is

$$\begin{aligned} x &= r_x \cos\phi \cos\theta, & -\pi/2 \le \phi \le \pi/2 \\ y &= r_y \cos\phi \sin\theta, & -\pi \le \theta \le \pi \\ z &= r_z \sin\phi \end{aligned} \tag{10-10}$$

<u>Blobby objects:</u> Irregular surfaces can also be generated using some special formulating approach, to form a kind of **blobby objects** -- The shapes showing a certain degree of fluidity **.** Blobby objects (as is obvious by their name) are the objects which change their shape and size on the basis of their states. In motion or in other such states, they show a flexible or non-rigid behavior and hence, change their shape and size.



Unit :- 4

Date :
Page No. 1

Curve :-

Definition 1. When sets of points infinite or finite are joined continuous than what we get is called curve.

Definition 2. When we start from a point for drawing a geometrical figure and end at some other point without any GAP, so what we get is called CURVE.

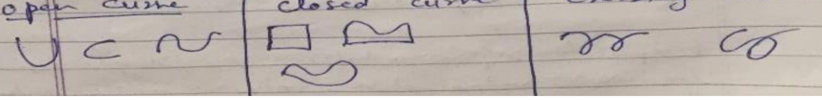
[But according to this approach a line joint end points is also a curve. mathematically.

curve

So to specify a curve, we need more information than just its endpoints. This may mean that a different display file structure should be used.

There are many types of curves like :-

open curve	closed curve	crossing curve



Representing Curves: In computer graphics we daily need to draw or design different types of objects which have bends, deviations and smoothness.

[Smoothness of a function:- is a property measured by no. of continuous derivatives] Like human face, Automobiles designs, digitize drawing, animation paths.

⇒ We can represent basically curves by 3 mathematical functions.

(1) Explicit curve function
(2) Implicit curve function
(3) Parametric curve function

1. Explicit Representation of Curves:

A mathematical function $\boxed{y = f(x)}$ can be plotted as a curve. Such function is the explicit representation of curve.
In this the dependant variable has been given explicitly in terms of the independent variable.

Explicit representation is single valued for each value of x, only a single value y is computed

2. Implicit Representation of Curve :

It defines the set of points on a curve be employing a procedure that can test to see if a point in on the curve.

In this dependent variable is not expressed in terms of some indepent variables.

$f(x, y) = 0 \quad \Rightarrow \quad x^2 + y^2 - 1 = 0$

$y^2 + x^3 + 18 = 0$

It can represent multivalued curves (multiple y values for one x value).

eg. $x^2 + y^2 + R^2 = 0$ circle

[we can change implicit to explicit but this

$\boxed{y = \pm\sqrt{1-x^2}}$ $\begin{bmatrix} (1) +ve \\ (2) -ve \end{bmatrix}$

new: explicit function becomes very complex and some times gives us 2 branches

3 Parametric Curves :-

Curves having parametric form are called parametric curves. The explicit and implicit ~~parametric~~ curves ~~are used~~. can be used only when function is known

Instead of defining y in terms of x [$y = f(x)$] or x in terms of y [$x = g(y)$].

we define both x, y in terms of a third variable called a parameter

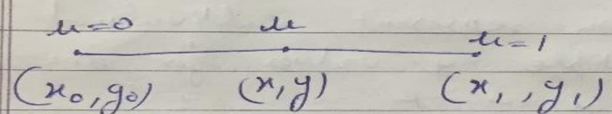
$$x = f_x(u) \qquad \left[u \text{ is a parameter} \right.$$

$$y = f_y(u)$$

Like line parametric equation is

$$x = (1-u)x_0 + u x_1$$

$$y = (1-u)y_0 + u y_1 \qquad \left[\begin{array}{l} (1-t)x_0 + t x_1 \\ 0 \le t \le 1 \end{array} \right]$$

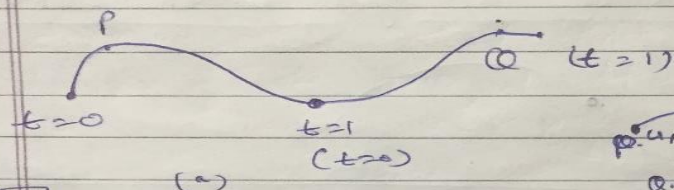


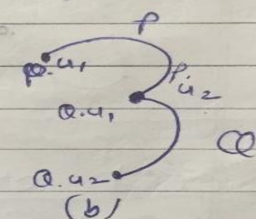
Parametric continuity conditions

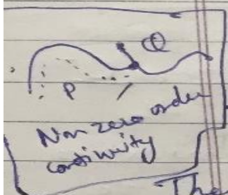
To ensure a smooth transition from one section of a piecewise parametric curve to the next, we can impose various continuity conditions at the connection points.

⇒ Parametric continuity deals in parametric equations associated to piece wise parametric polynomial curve. not the shape or appearance of the curve

(1) Zero order Parametric Continuity : C^0
means simply that the curves meet.







Zero order continuity.

There are two piece of curve P & Q

(a) If $P(t=1) = Q(t=0) \Rightarrow$ zero order Parametric continuity

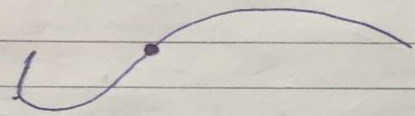
That is, the value of x, y, z evaluated at u_2 for the first curve section are equal, respectively, to the value of x, y, z evaluated at u_1 for the next curve section. (b)

~~Second-order~~

First-order parametric continuity:
referred to as C' continuity, means that the first parametric derivates (tangent lines) of the coordinate functions for two successive curve sections are equal at their joining point.

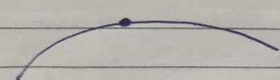
$$P'(t=1) = Q'(t=0)$$

P' & Q' are first order Derivatives.



Second order Parametric Continuity or C^2 continuity; means that both the first and second parametric derivatives of the two curve sections are the same at intersection.

$$P''(t=1) = Q''(t=0)$$



[Higher-order parametric continuity conditions are defined similarly.]

[With second-order continuity] [Smooth curve]
The rate of change of the tangent vectors for connecting sections are equal at their intersection. Thus the tangent line transitions smoothly from one section of the curve to the next.

But with first-order continuity, the rates of change of the tangent vectors for two sections can be quite different. so that the general shapes of the two adjacent sections can change abruptly.

Uses Ist order Continuity :- Digitizing drawings, some design applications.

IInd-order continuity : Setting up animation paths for camera motion & for many precision CAD requirements.

Geometric Continuity Condition :-

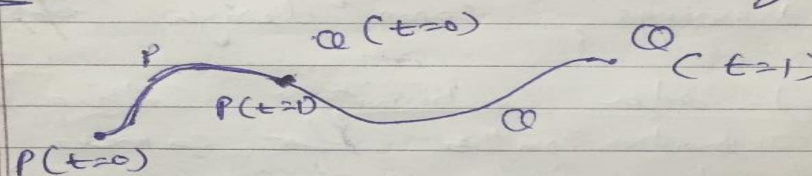
Geometric continuity refers to the way that a curve or surface looks.

In this case, we only require parametric derivates of the two sections to be proportional, to each other at their common boundary instead of equal to each other.

Zero - order geometric continuity :

described as G^0 continuity, is the same as zero-order parametric continuity.

That is, the two curves sections must have the same coordinate position at the boundary point.



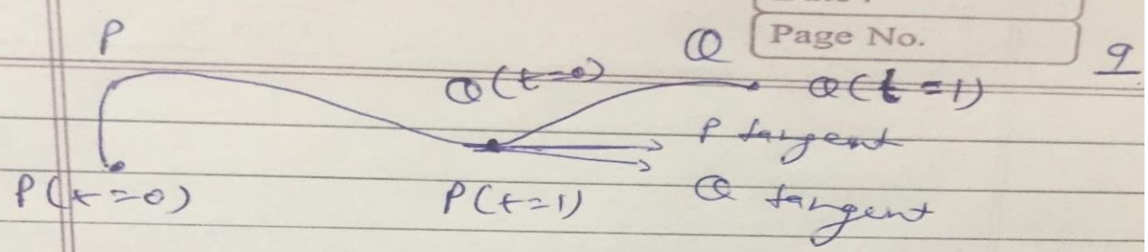
$$P(t=1) = Q(t=0)$$

where P & Q are two successive piece curve or segments of curves.

First-order Geometric continuity or G'

continuity, means that the parametric first order derivates are proportional at the intersection of two successive sections.

If we P, Q are two segment of curve, then $P'(t=1)$ & $Q'(t=0)$ must have same direction of tangent vector but not necessarily its magnitude,



$P'(t=1) = (x, y, z)$ & $Q'(t=0) = (k^*x, k^*y, k^*z)$

Second order geometric continuity or G^2

continuity means that both the first and second parametric derivatives of two ~~surface~~ curve sections are proportional at their boundary.

Under G^2 continuity, curvatures of two curve sections will match at the joining position.

[The tangent vector $Q'(t)$ is the velocity of a point on the curve w.r.t. parameter. Similarly $Q''(t)$ is the acceleration.

In short:

Order of continuity(parametric Continuity)	Geometric Continuity
The various order of parametric continuity can be described as follows: • C^0: Curves are continuous • C^1: First derivatives are continuous • C^2: First and second derivatives are continuous • C^n: First through nth derivatives are continuous	**Smoothness of curves and surfaces** A curve or surface can be described as having G^n continuity, with n being the increasing measure of smoothness. Consider the segments either side of a point on a curve: • G^0: The curves touch at the join point. • G^1: The curves also share a common tangent direction at the join point. • G^2: The curves also share a common center of curvature at the join point.

Spline Curve

→ Spline :- is a flexible strip used to produce a smooth curve through a designated set of points.
Several small weights are distributed along the length of the strip to hold it in position on the drafting table as the curve is drawn.

Spline Curve :- the spline curve originally referred to a curve drawn in this manner.

A spline curve is mathematical representation for which it is easy to build an interface that will allow a user to design and control the shape of complex curves and surfaces.

Spline curve mathematically described with a ~~piecewise~~ piecewise cubic polynomial function whose

spline curve first & second derivatives are continuous across the various curve section.

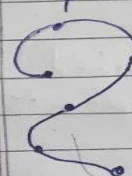
C^1 & C^2 continuity

In
whe
are
curv
All
then
be i
contr
A set of
interpol

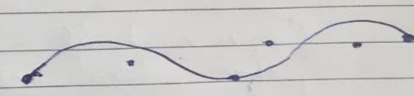
Control points :- We specify a spline curve by giving a set of coordinate positions, called control points, which indicates the general shape of the curve. These control points are then fitted with piecewise continuous parametric polynomial functions in 2 ways

Spline Curve (control points)

Interpolate or Interpolation Spline	Approximate or Approximation spline
When polynomial sections are fitted so that the curve passes through [All control points], then curve is said to be interpolate the set of contro points.	When polynomial sections are fitted to the path which is not necessarily passing through all control points, then curve is said to approximate the set of control points.



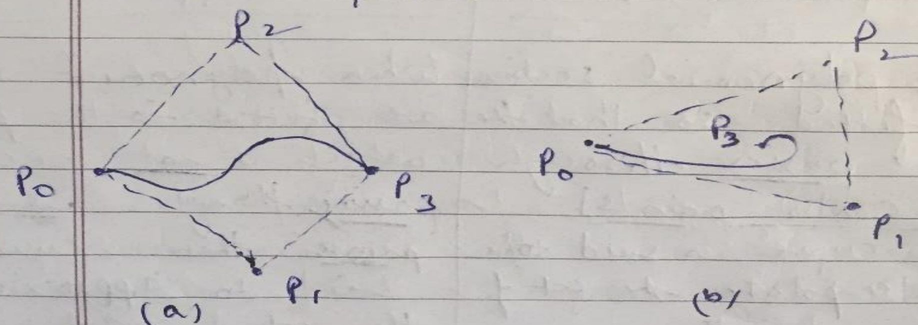
A set of six control points interpolated with piecewise continuous polygnomial sections



A set of six control points approximated with piecewise continuous polynomial sections

Interpolate	Approximate
are used to digitize drawings or to specify the animation paths.	Are used as design tools to structure object surface
eg. Hermite spline	eg. B- spline

Convex hull :- The convex polygon boundary that encloses a set of control points.



Convex-hull shapes (dashed lines) for two sets of control points.

~~Control point~~ ordering.

Control graph :- A polygon connecting the sequence of control points for the approximation spline is usually displayed to remind a designer of the control point ordering.

This set of connected line segments is referred to as the control graph. of the curve : or control polygon and characteristic polygon.

Spline specification :-

Three equivalent methods for specifying a particular spline representation

1) We can state the set of boundary conditions that are imposed on the spline.
2) We can state the matrix that characterizes the spline.
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.

To illustrate this we have parametric cubic polynomial representation for the x coordinate along the path of a spline section.

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x \quad (1)$$

$$0 \le u \le 1$$

Boundary condition for this curve is
end points, parametric first derivates (end points)
$x(0)$, $x(1)$, $x'(0)$, $x'(1)$

→ From boundary condition we can obtain matrix

$$x(u) = [u^3 \; u^2 \; u \; 1] \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad \text{--- } 2$$

$$= \underline{U \cdot C}$$

where $\underline{U}$ is the row matrix of power of parameter u,
&
$\underline{C}$ is the coefficient collumn matrix

using eq 2 we can write the boundary conditions in matrix form & solve for coefficient matrix C as

$$C = M_{spline} \cdot M_{geom} \quad \text{—(3)}$$

Mgeom is a four-element column matrix [geometric constraint values]
↓ boundary condition on spline;

Mspline is 4x4 matrix that transforms the geometric constraint values to the [polygon coefficients].

from eq (2, 3)

$$x(u) = U \cdot M_{spline} \cdot M_{geom} \quad (4)$$

where matrix Mspline characterizing a spline representation, sometimes called the [Basic matrix].

we can expand eq 4 to obtain polynomial representation for coordinate x in terms of geometric constraint parameters

$$x(u) = \sum_{k=0}^{3} g_k \cdot BF_k(u)$$

where g_k - Constraint parameter [control point coordinates / slope of curve.

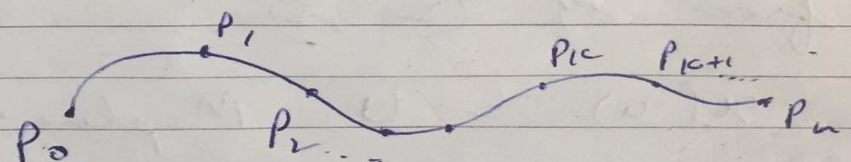
$BF_k(u)$ are polynomial Blending function.

Like wise Cubic spline interpolation for all x-axis.

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

$$(0 \le u \le 1)$$

for representing n curve section for between n+1 control points.



a piecewise continuous cubic - spline interpolation of n+1 control points.

Hermite Interpolation

A **Hermite spline** (named after the French mathematician Charles Hermite) is an interpolating piecewise cubic polynomial with a specified tangent at each control point. Unlike the natural cubic splines, Hermite splines can be adjusted locally because each curve section is only dependent on its endpoint constraints.

If $\mathbf{P}(u)$ represents a parametric cubic point function for the curve section between control points $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$, as shown in Fig. 10-27, then the boundary conditions that define this Hermite curve section are

$$\begin{aligned} \mathbf{P}(0) &= \mathbf{p}_k \\ \mathbf{P}(1) &= \mathbf{p}_{k+1} \\ \mathbf{P}'(0) &= \mathbf{Dp}_k \\ \mathbf{P}'(1) &= \mathbf{Dp}_{k+1} \end{aligned} \tag{10-27}$$

with $\mathbf{Dp}_k$ and $\mathbf{Dp}_{k+1}$ specifying the values for the parametric derivatives (slope of the curve) at control points $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$, respectively.

We can write the vector equivalent of Eqs. 10-26 for this Hermite-curve section as

$$\mathbf{P}(u) = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d}, \qquad 0 \le u \le 1 \tag{10-28}$$

where the x component of $\mathbf{P}$ is $x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$, and similarly for the y and z components. The matrix equivalent of Eq. 10-28 is

$$\mathbf{P}(u) = [u^3 \; u^2 \; u \; 1] \cdot \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} \tag{10-29}$$

and the derivative of the point function can be expressed as

$$\mathbf{P}'(u) = [3u^2 \; 2u \; 1 \; 0] \cdot \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} \tag{10-30}$$

Substituting endpoint values 0 and 1 for parameter u into the previous two equations, we can express the Hermite boundary conditions 10-27 in the matrix form:

$$\begin{bmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{Dp}_k \\ \mathbf{Dp}_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} \tag{10-31}$$

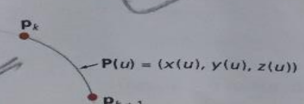


Figure 10-27
Parametric point function $\mathbf{P}(u)$ for a Hermite curve section between control points $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$.

Date :

Solving this equation for the polynomial coefficients, we have

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{Dp}_k \\ \mathbf{Dp}_{k+1} \end{bmatrix}$$

$A^{-1} = \frac{\operatorname{adj} A}{|A|}$

U . C

$$= \underbrace{\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{M_{sp}} \cdot \underbrace{\begin{bmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{Dp}_k \\ \mathbf{Dp}_{k+1} \end{bmatrix}}_{M_{geom}} \qquad (10\text{-}32)$$

$C = M_{sp} \cdot M_{geom}$

$$= \mathbf{M}_H \cdot \begin{bmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{Dp}_k \\ \mathbf{Dp}_{k+1} \end{bmatrix}$$

∴ [in terms of Hermit we write, it as M_H]

where $\mathbf{M}_H$, the Hermite matrix, is the inverse of the boundary constraint matrix. Equation 10-29 can thus be written in terms of the boundary conditions as

$$\mathbf{P}(u) = [u^3 \; u^2 \; u \; 1] \cdot \mathbf{M}_H \cdot \begin{bmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{Dp}_k \\ \mathbf{Dp}_{k+1} \end{bmatrix} \qquad (10\text{-}33)$$

[∴ $x(u) = U \cdot M_{sp} \cdot M_{geom}$

Finally, we can determine expressions for the Hermite blending functions by carrying out the matrix multiplications in Eq. 10-33 and collecting coefficients for the boundary constraints to obtain the polynomial form:

$$\begin{aligned} \mathbf{P}(u) &= \mathbf{p}_k(2u^3 - 3u^2 + 1) + \mathbf{p}_{k+1}(-2u^3 + 3u^2) + \mathbf{Dp}_k(u^3 - 2u^2 + u) \\ &\quad + \mathbf{Dp}_{k+1}(u^3 - u^2) \\ &= \mathbf{p}_k H_0(u) + \mathbf{p}_{k+1} H_1(u) + \mathbf{Dp}_k H_2(u) + \mathbf{Dp}_{k+1} H_3(u) \end{aligned} \qquad (10\text{-}34)$$

The polynomials $H_k(u)$ for $k = 0, 1, 2, 3$ are referred to as blending functions because they blend the boundary constraint values (endpoint coordinates and slopes) to obtain each coordinate position along the curve. Figure 10-28 shows the shape of the four Hermite blending functions.

Hermite polynomials can be useful for some digitizing applications where it may not be too difficult to specify or approximate the curve slopes. But for most problems in computer graphics, it is more useful to generate spline curves without requiring input values for curve slopes or other geometric information, in addition to control-point coordinates. Cardinal splines and Kochanek–Bartels splines, discussed in the following two sections, are variations on the Hermite splines that do not require input values for the curve derivatives at the control points. Procedures for these splines compute parametric derivatives from the coordinate positions of the control points.

Cardinal Splines

...cubics with

14.6. CUBIC BEZIER CURVE

Unlike the piecewise spline curve, a Bezier spline curve can be fitted to any number of control points. Without necessitating tangent vector specification at any of the control points, a set of characteristic polynomial approximating functions, called Bezier blending functions, blend the control points to produce a Bezier curve segment. However, the degree of a Bezier curve segment is determined by the number of control points to be fitted with that curve segment.

Given a set of $n + 1$ control points $P_0, P_1, \ldots, P_n$, a parametric Bezier curve segment that will fit to those points is mathematically defined by,

$$P(u) = \sum_{i=0}^{n} P_i \, B_{i,\,n}(u) \text{ where } 0 \le u \le 1 \quad \ldots (1)$$

Date :

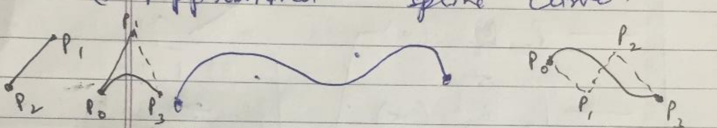
Page No.

Bezier Spline Curve

The curve method is developed by the French engineer Pierre Bezier for use in the design of Renault automobile bodies.

A Bezier curve section can be fitted to any no. of control points. No. of control pts to be approximated & their relative position determine the degree of Bezier polynomial. [n+1 - control pts , degree is n]

(i) Approximate spline curve.



(ii) Barnstein Polynomial fn

(iii) All control points use Bez. eq.

$$P(u) = \sum_{k=0}^{n} P_k \cdot B_{k,n}(u)$$

[range of u is 0, 1] [$0 \le u \le 1$]

P_k = Position vector

$B_{k,n}$ = Barnstein, Bezier fn. (Basic, continuity, smoothness)

[n+1 - control pts. i = 0-n]

$$x(u) = \sum_{k=0}^{n} x_k \cdot B_{k,n}(u)$$

$$B_{k,n} = {}^{n}C_k \cdot u^k \cdot (1-u)^{n-1}$$

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

$$\frac{n!}{k!(n-k)!}$$ binomial coeffic

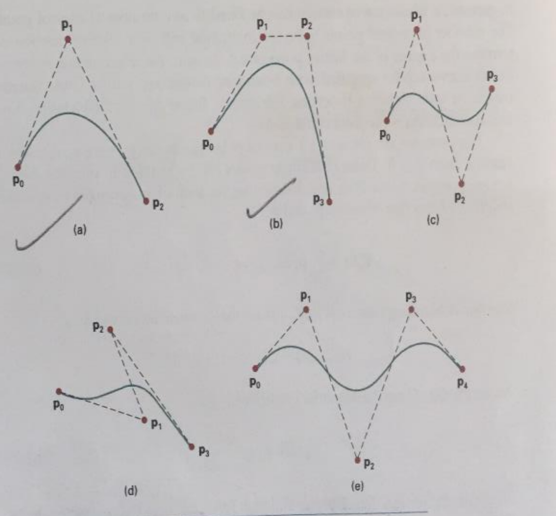


Figure 10-34
Examples of two-dimensional Bézier curves generated from three, four, and five control points. Dashed lines connect the control-point positions.

By expanding the polynomial expressions for the blending functions, we can write the cubic Bézier point function in the matrix form

$$\mathbf{P}(u) = [u^3 \; u^2 \; u \; 1] \cdot \mathbf{M}_{\text{Bez}} \cdot \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} \qquad (10\text{-}51)$$

where the **Bézier matrix** is

$$\mathbf{M}_{\text{Bez}} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad (10\text{-}52)$$

We could also introduce additional parameters to allow adjustment of curve "tension" and "bias", as we did with the interpolating splines. But the more useful B-splines, as well as β-splines, provide this capability.

[Many graphics package provide only cubic spline fn. 3 degree eq. [if we use more control pts. complexity increase] 2

if we have four control pts. given

Eq. for 4 control points :

4 pts. 0 - 3

$Q(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u) + P_2 B_{2,3}(u) + P_3 B_{3,3}(u)$

$\frac{3!}{0!3!} \times u^0 . (1-u)^3 = 1 . (1-u)^3$

$\frac{3!}{1! . 2!} u^1 (1-u)^2$

$(B_{1,3}) \Rightarrow 3 . u . (1-u^2)$

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

$B_{3,3} = u^3$

$\Rightarrow$

$Q(u) = P_0 (1-u)^3 + P_1 . 3u(1-u)^2 + P_2 . 3u^2(1-u) + P_3 u^3$ ——— eq (1)

$x(u) = x_0 (1-u)^3 + 3 . u(1-u)^2 + \cancel{x_1}\ 3 . u(1-u)^2 x_1 + 3u^2(1-u) . x_2 + u^3 x_3$.

Similarly we can calculate :-
y(u)
z(u)

$x(u) = \sum_{k=0}^{n} x_k BEZ_{k,n}(u)$

$y(u) = \sum_{k=0}^{n} y_k Ben_{k,n}(u)$

$z(u) = \sum_{k=0}^{n} z_k Ben_{k,n}(u)$

where $B_{i,n}(u)$ are the Bezier blending functions, also known as Bernstein Basis functions defined as

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

and $C(n, i)$ is the binomial coefficient.

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

Expanding the equation, we get

$$P(u) = P_0 B_{0,n}(u) + P_1 B_{1,n}(u) + ... + P_{n-1} B_{n-1,n}(u) + P_n B_{n,n}(u)$$

C(u) or

$$= P_0\{C(n, 0)\ u^0(1-u)^n\} + P_1\{C(n, 1)\ u^1(1-u)^{n-1}\} + ...$$
$$+ P_{n-1}\{C(n, n-1)\ u^{n-1}(1-u)\} + P_n\{C(n, n)\ u^n (1-u)^0\}$$

From this equation we discover two very important facts about Bezier curves, if we minutely observe the blending functions in curly brackets:

(1) n, the degree of polynomial blending functions and thus of the Bezier curve segment is one less than the number of control points used.

(2) The Bezier curve in general passes through only the first control point P_0 and the last control point P_n as because $P(u = 0) = P_0$ and $P(u = 1) = P_1$.

For Bezier curve of degree 3 ($n = 3$), we find that four ($n + 1$) control points are required to specify a cubic Bezier curve segment.

Thus, for a parametric cubic Bezier curve,

$$P(u) = \sum_{i=0}^{3} P_i B_{i,3}(u) \quad 0 \le u \le 1$$

$$\Rightarrow \quad P(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u) + P_2 B_{2,3}(u) + P_3 B_{3,3}(u)$$

where

$$B_{0,3}(u) = \left\{\frac{3!}{0!3!}\right\} u^0 (1-u)^3 = (1-u)^3$$

$$B_{1,3}(u) = \left\{\frac{3!}{1!2!}\right\} u^1 (1-u)^2 = 3u(1-u)^2$$

$$B_{2,3}(u) = \left\{\frac{3!}{2!1!}\right\} u^2 (1-u) = 3u^2 (1-u)$$

$$B_{3,3}(u) = \left\{\frac{3!}{3!0!}\right\} u^3 (1-u)^0 = u^3$$

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

$$\Rightarrow \quad P(u) = (-u^3 + 3u^2 - 3u + 1)P_0 + (3u^3 - 6u^2 + 3u) P_1 + (-3u^3 + 3u^2) P_2 + u^3 P_3.$$

This expression can be translated into the matrix form *i.e.,*

$$P(u) = [F]\ [B]$$

or

$$P(u) = [T]\ [M]\ [B]$$

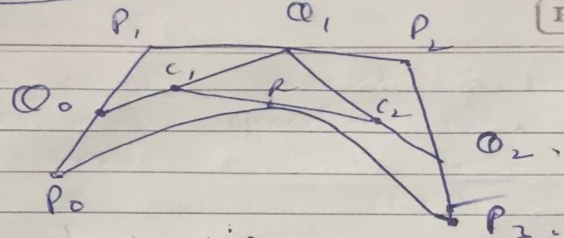
Here the basis function matrix $[F] = [B_{0,3}(u) \quad B_{1,3}(u) \quad B_{2,3}(u) \quad B_{3,3}(u)]$

The Bezier geometry matrix

$$[B] = \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

How to Derive Bezier function

Date :
Page No. 3



(by using line parametric eq)

$(P_0 - P_1)$

$Q_0 = (1-u)P_0 + u \cdot P_1$

$Q_1 = (1-u)P_1 + u \cdot P_2$

$Q_2 = (1-u)P_2 + uP_3$

$C_1 = (1-u)Q_0 + uQ_1$

$C_2 = (1-u)Q_1 + uQ_2$

$R = (1-u)C_1, uC_2$

$$\Rightarrow R = 1-u[(1-u)Q_0 + u \cdot Q_1] + u[(1-u)Q_1 + u \cdot Q_2]$$

$$= (1-u)[(1-u)[(1-u)P_0 + uP_1] + u[(1-u)P_1 + uP_2]]$$

$$+ u[(1-u)[(1-u)P_1 + u \cdot P_2]] + u[(1-u) \cdot P_2 + u \cdot P_3]]$$

after Solve we get R — eq$^{(1)}$ $Q(u)$

$B_{0,3}$, $B_{1,3}$, $B_{2,3}$, $B_{3,3}$ — Blending function.

Date :

Page No. 4

Properties of Bezier Curve :-

(1) Bezier Curve always pass through
(1.) First control P(0) & last control point $P_n(1)$

2. It make a polygon boundary by Control points.

3. It will always be inside convex hull of polygon boundary.

4. Polynomial equation degree is less than one from control point. suppose if 4 control pt. then degree is 3.

5. ⇒ It is used in CAD, CAM, corel draw software. (graphic package).

6. It is quite easy to implement.

7. They generally follow the shape of the control polygon which consists of the segments joining the control points.

Disadvantages :- 1) It's degree depends on number of control points

2) Bezier Curve exhibit global control property means moving a control point alters the shape of the whole curve. (∵ all Blending function are non-zero).

B-Spline Curve

1.B-splines are generalizations of Bernstein polynomials(Bezier) and share many of their analytic and geometric properties.

2.B-splines provide an approximation scheme where such constraints on the location of the control points are not necessary.

3.B-spline curves and surfaces meet smoothly at their joins for completely arbitrary collections of control points.

4.Thus B-splines provide a simpler, numerically more stable approach to approximating large amounts of data.

5.For these reasons B-splines have become extremely popular in large-scale industrial applications.

B-SPLINE CURVE

We have Some limitations in Bezier Curve like →

1) The Bezier Curve produced by Bernstein basis fn has limited Flexibility.

Numbers of Control points decides the degree of the Polynomial Curve. Ex:- 4 Control points results a Cubic polynomial Curve.

So only one way to reduce the degree of the Curve is to reduce the no. of Controlpoints and vice Versa.

2) The Second limitation is that the Value of the blending fn is non-zero for all parameter Values over the entire Curve.

Due to this Change in one Vertex, Changes the entire Curve and this eliminates the ability to produce a local Change with in a Curve.

So B-Spline Curve — Basis-Spline Curve is solution of this limitations of Bezier Curve.

Properties of B-Spline Curve :-

1) B-Spline basis is non-global (LOCAL) effect.
In this each Control point affects the shape of the Curve Only over range of parameter Values where its associated basis fn is non-zero.

2) B-spline Curve made up of n+1 Control point

3) B-Spline Curve let us specify the order of basis (k) fn and the degree of the resulting Curve is independent on the no. of Vertices.

4) It is possible to change the degree of the resulting Curve with out changing the no. of Control points.

5) B-Spline Can be used to define both open & close Curves.

6) Curve generally follows the shape of defining polygon
If we have order K=4 then degree will be 3 $P(K)=x^3$

7) The Curve line with in the Convex hull of its defining Polygon.

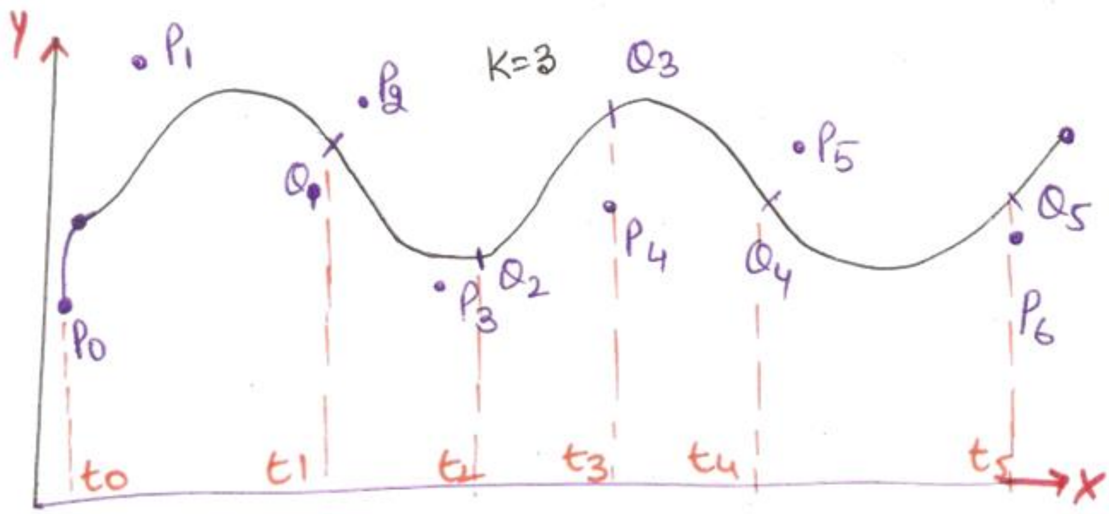
In B-Spline we Segment out the whole Curve which is decided by the order (K). by Formula '$n-K+2$'

for Example:-

If we have 7 control points and order of Curve $k=3$ then $n=6$

and this B-Spline Curve has Segments

$6-3+2=5$



five Segments Q_1, Q_2, Q_3, Q_4, Q_5

Segment	Control points	Parameter
Q_1	$P_0\ P_1\ P_2$	$t_0=0,\ t_1=1$
Q_2	$P_1\ P_2\ P_3$	$t_1=1,\ t_2=2$
Q_3	$P_2\ P_3\ P_4$	$t_2=2,\ t_3=3$
Q_4	$P_3\ P_4\ P_5$	$t_3=3\quad t_4=4$
Q_5	$P_4\ P_5\ P_6$	$t_4=4\quad t_5=5$

There willbe a Join point or knot between Q_{i-1} & Q_i for $i \geq 3$ at the parameter value t_i know as KNOT VALUE [X].

If $P(u)$ be the position vectors along the curve as a fn of the parameter u, a B-Spline curve is given by

$$P(u) = \sum_{i=0}^{n} P_i \, N_{i,k}(u) \qquad 0 \leq u \leq n-k+2$$

$N_{i,k}(u)$ is B-Spline basis fn

$$N_{i,k}(u) = \frac{(u - x_i)\, N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)\, N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

The values of x_i are the elements of a knot vector satisfying the relation $x_i \leq x_{i+1}$.

The parameter u varies from 0 to $n-k+2$ along the $P(u)$

So there are Some Conditions for finding the KNOT VALUES [X]

$x_i \, (0 \leq i \leq n+k) \rightarrow$ Knot Values

$x_i = 0$ if $i < k$

$x_i = i-k+1$ if $k \leq i \leq n$

$x_i = n-k+2$ if $i > n$

So as B-Spline Curve has Recursive Equ. so we stop at

$$N_{i,k}(u) = 1 \text{ if } x_i \leq u \, x_{i+1}$$

$$= 0 \quad \text{otherwise}$$

Example :-

$n=5, \; k=3$

then $x_i \, (0 \leq i \leq 8)$ knot Values

$x_i \left\{ \underset{x_0}{0}, \underset{x_1}{0}, \underset{x_2}{0}, \underset{x_3}{1}, 2, 3, 4, 4, \underset{x_8}{4} \right\}$

$$\underbrace{N_{0,3}(u) = (1-u)^2 \cdot N_{2,1}(u)}_{\text{After Calculation.}}$$

→ when $i=0$, $k=3$ so $i<k$ is true

$x_0 = 0$

→ $i=1$, $k=3$ $x_1 = 0$

$i=2$, $k=3$ $x_2 = 0$

$i=3$, $k=3$ $x_3 = i-k+1 = 3-3+1$

$x_3 = 1$

$i=4$, $k=3$ $x_4 = i-k+1 \Rightarrow 4-3+1$

$x_4 = 2$

$i=8$ $k=3$ $x_8 = i > n$ $n-k+2 = 5-3+2 = 4$

$x_8 = 4$

In this way we will Calculate

... the shape of at most d curve sections.

In addition, a B-spline curves lies within the convex hull of at most $d + 1$ control points, so that B-splines are tightly bound to the input positions. For any value of u in the interval from knot value u_{d-1} to u_{n+1}, the sum over all basis functions is 1:

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

Given the control-point positions and the value of parameter d, we then need to specify the knot values to obtain the blending functions.

There are three general classifications for knot vectors:

(*a*) Uniform

(*b*) Open-uniform

(*c*) Non-uniform

When the spacing between the knot values is constant, the resulting curve is called a *uniform B-spline. For example,* the integer knot vector [0 1 2 3 4 5 6 7] we have used for the 4-control point cubic B-spline.

Uniform B-splines have *periodic* blending functions. That is, for given values of n and d, all blending functions have the same shape. Each successive blending function is simply a shifted version of the previous function.

$$B_{k,d}(u) = B_{k+1,d}(u + \Delta u) = B_{k+2,d}(u + 2\Delta u)$$

where Δu is the interval between adjacent knot values.

Open, Uniform B-splines class of B-splines is a cross between uniform B-splines and non-uniform B-splines. Sometimes it is treated as a special type of uniform B-spline and sometimes it is considered to be in the non–uniform B-spline classification. For the open, uniform B-splines or simply open B-splines, the knot spacing is uniform except at the ends where knot values are repeated d times. *For example,*

[0, 0, 1, 2, 3, 3] $d = 2$ and $n = 3$

[0, 0, 0, 0, 1, 2, 2, 2, 2] $d = 4$ and $n = 4$

with nonuniform B-splines, we can choose multiple internal knot values and unequal spacing between the knot values.

For example, [0, 1, 2, 3, 3, 4]

[0, 2, 2, 3, 3, 6]

Non-unifrom B-splines provide increased flexibility in controlling a curve shape. With unequally spaced intervals in the knot vector, we obtain different shapes for the blending functions in different intervals, which can be used to adjust spline shapes. By increasing knot multiplicity, we produce subtle variations in curve shape and even introduce discontinuities. Multiple knot values also reduce the continuity by 1 for each repeat of a particular value.

14.8. BEZIER SURFACES AND B-SPLINE SURFACES

Two sets of orthogonal Bezier curves can be used to design an object surface by specifying by an input mesh of control points. The parametric vector function for the Bezier surface is formed as the Cartesian product of Bezier blending functions.

$$P(u, v) = \sum_{j=0}^{m} \sum_{k=0}^{n} P_{j,k} B_{j,m}(v) B_{k,n}(u)$$

where $P_{j,k}$ specifying the location of the $(m + 1)$ by $(n + 1)$ control points.

Bezier surfaces have the same properties as Bezier curves and they provide a convenient method for interactive design applications.

Formulation of a *B-spline surface* is similar to that for Bezier surface. We can obtain a vector point function over a B-spline surface using the Cartesian product of B-spline blending functions in the form.

$$P(u, v) = \sum_{k_1=0}^{n_1} \sum_{k_2=0}^{n_2} P_{k1,k2} B_{k1,d_1}(u) B_{k2,d_2}(v)$$

where the vector values for $P_{k1,k2}$ specify positions of the $(n_1 + 1)$ by $(n_2 + 1)$ control points.

EXAMPLE 14.4

... P (60, 60), P_4(60,

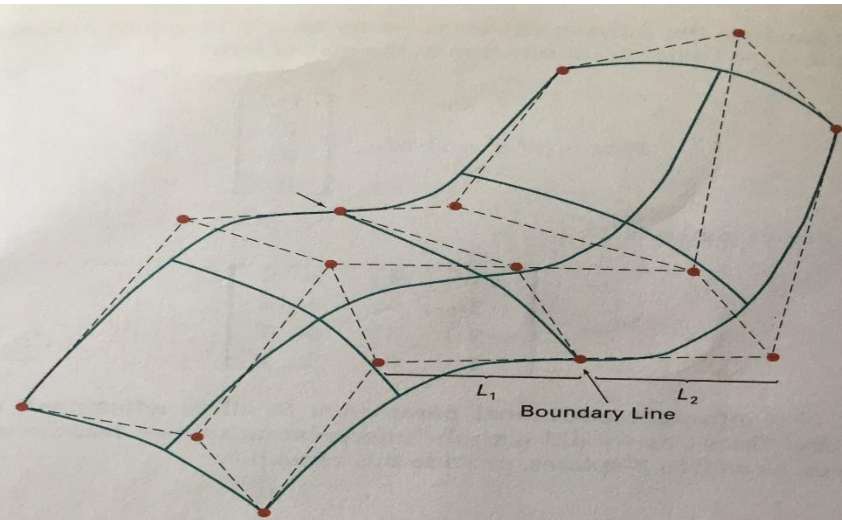


Figure 10-40
A composite Bézier surface constructed with two Bézier sections, joined at the indicated boundary line. The dashed lines connect specified control points. First-order continuity is established by making the ratio of length L_1 to length L_2 constant for each collinear line of control points across the boundary between the surface sections.