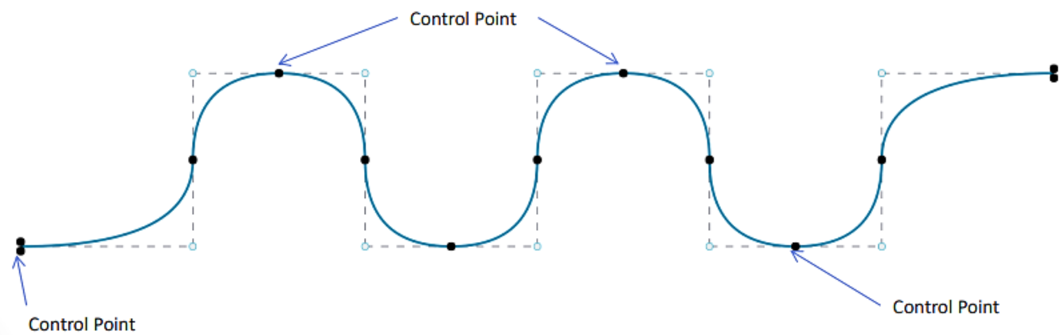
# Chapter 5 Curve Modeling

## Spline Representation:

- Curve is the set of points that are joined continuously. A spline curve is a 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 shapes.
- A Spline is a flexible strip used to produce a smooth curve through a designated set of points.
- The general approach is that the user enters a sequence of points and a curve is constructed whose shape closely follows this sequence. The point are called **control point.**
- A spline curve is defined, modified, and manipulated with operations on the control points.
- Splines are used in graphics applications to design curve and surface shapes and to specify animation path for the object or image. Typical CAD applications for splines include the design of automobile bodies, aircrafts and spacecraft surfaces, and ship hulls.




## Interpolation and Approximation Splines

- A curve is actually passes through each control point is called **interpolating curve**. These are commonly used to digitize drawing or to specify animation paths and graphs of data trends of discrete set of data points.
- A curve that passes near to the control point but not necessarily through them is called an approximating curve. These are used as design tool to structure object surfaces and for GIS(geographical Information System) applications.

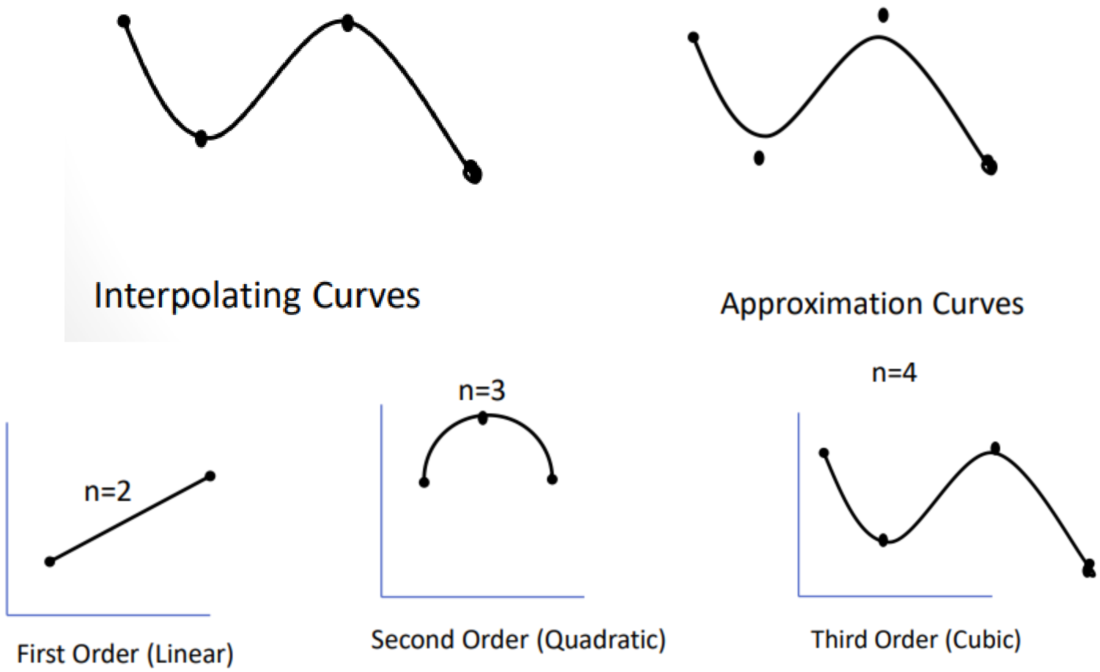



Figure Interpolating

**Convex hull:**

The convex polygon boundary that encloses a set of control points is called the convex hull. Convex hulls provide a measure for the deviation of a curve or surface from the region bounding the control points.

Some splines are bounded by the convex hull that ensures the polynomials smoothly follow the control points without erratic oscillations. Also, the polygon region inside the convex hull is useful in some algorithms as a clipping.

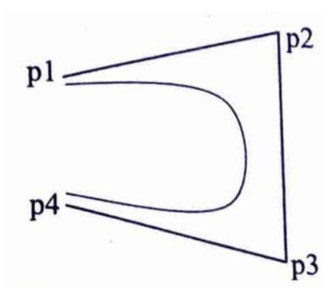



Figure Convex hull

# Chapter 5 Curve Modeling

**Three types of Curve:**

i. Open curve:

ii. Closed curve:

iii. Crossing curve:

**Representation of Curve:**

All objects are not flat but may have many bends and derivations. We have to compute all curves. We can represent curve by three mathematical function:

i. Explicit Function
ii. Implicit Function
iii. Parametric Function

**i. Explicit representation of curve:**

In this method the dependent variable is given explicitly in terms of the independent variable as;

$y = f(x)$

e.g., $y = mx + c$

$y = 5x^2 + 2x + 1$

In explicit representation, for each single value of x, only a single value of y is computed

**ii. Implicit representation of curve:**

In this method, dependent variable is not expressed in terms of some independent variables as;

$F(x,y)=0$

e.g.;

$x^2+y^2-1=0$

In implicit representation, for each single value of x, multiple values of y is computed.

If we convert implicit function to explicit function it will be more complex and will give different values.

e.g. $y=\pm\sqrt{1-x^2}$

### iii. Parametric representation of curve:

We cannot represent all curves in single equation in terms of only x and y. Instead of defining y in terms of x (i.e. y=f(x)) or x in terms of y ( i.e x=h(y)); we define both x and y in terms of a third variable in parametric form.

Curves having parametric form are called parametric curves.

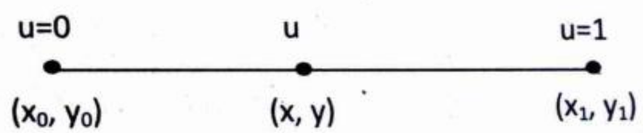
$x = f_x(u)$

$y = f_y(u)$ where u is parameter

similarly, parametric equation of line is;

$x = (1\text{-}u)\, x_0 + u\, x_1$

$y = (1\text{-}u)\, y_0 + u\, y_1$



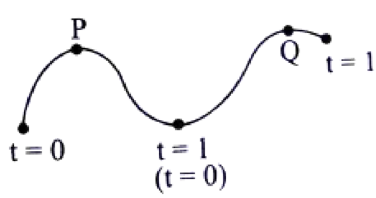

### Parametric Curve:

The parametric representation for curve is as follows:

$x = x(t)$

$y = y(t)$

$z = z(t)$



# Chapter 5 Curve Modeling

## Parametric Cubic Curve

Cubic polynomial means the polynomials which represents the curve with degree three.

$$Q(t) = [x(t) \quad y(t) \quad z(t)]$$

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

(Cubic polynomial function equation)

$$Qy(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

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

$$Q(t) = [t^3 \quad t^2 \quad t \quad 1] . \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$$

$$Q(t) = T.C.$$

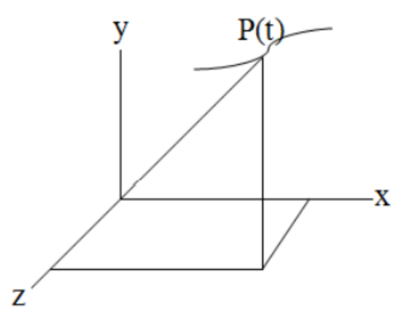
$$C = M.G.$$

Where M is 4 X 4 basis matrix and G is a four element column vector of geometric constants called geometric vector.

## Hermite spline Curve:

A parametric cubic curve is defined as

$$P(t) = \sum_{i=0}^{3} a_i t^i \qquad 0 <= t <= 1 \text{ ------ (i)}$$




Where, P($t$) is a point on the curve

a= algebraic coefficients

t= tangent Vector

Expanding equation (i) yield

$$P(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \text{ ------------------------------ (ii)}$$

This equation is separated into three components of P (t)

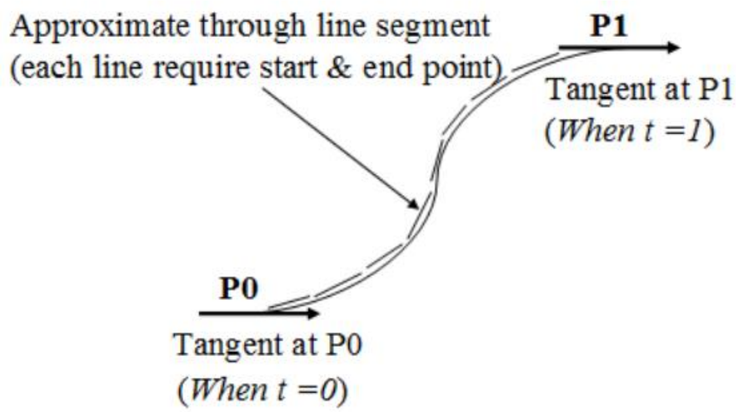
$$x(t) = a_{3x} t^3 + a_{2x} t^2 + a_{1x} t + a_{0x}$$

$$y(t) = a_{3y} t^3 + a_{2y} t^2 + a_{1y} t + a_{0y}$$

$$z(t) = a_{3z} t^3 + a_{2z} t^2 + a_{1z} t + a_{0z} \qquad \text{--------------------(iii)}$$

# Chapter 5 Curve Modeling

- To be able to solve the twelve unknown coefficients $\mathbf{a_{ij}}$ (algebraic coefficients) must be specified
- From the known end point coordinates of each segment, six of the twelve needed equations are obtained.
- The other six are found by using tangent vectors at the two ends of each segment
- The direction of the tangent vectors establishes the slopes(direction cosines) of the curve at the end point




- This procedure for defining a cubic curve using end points and tangent vector is one form of **Hermite interpolation**
- Each cubic curve segment is parameterized from 0 to 1 so that known end points correspond to the limit values of the parametric variable t, that is P(0) and P(1)
- Substituting t = 0 and t = 1 the relationship between two end point vectors and the algebraic coefficients are found

$$P(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$\mathbf{P(0) = a_0} \qquad \mathbf{P(1) = a_3 + a_2 + a_1 + a_0} \text{ ----- ---- ---- (IV)}$$

- To find the tangent vectors equation (ii) must be differentiated with respect to t

$$P(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$\mathbf{P'(t) = 3\, a_3 t^2 + 2\, a_2 t + a_1}$$

- The tangent vectors at the two end points are found by substituting t = 0 and t = 1 in this equation

$$P'(0) = a_1 \qquad\qquad P'(1) = 3a_3 + 2a_2 + a_1 \text{------------------- (V)}$$

- The algebraic coefficients '$a_i$' in equation (ii) can now be written explicitly in terms of boundary conditions – endpoints and tangent vectors are

$$a0 = P(0) \qquad\qquad a1 = P'(0)$$
$$a2 = -3P(0) - 3P(1) - 2P'(0) - P'(1)$$
$$a3 = 2P(0) - 2P(1) + P'(0) + P'(1)$$

(Note: - The value of a2 & a3 can be determined by solving the equation IV & V)

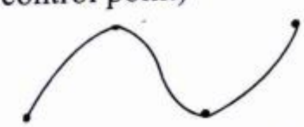
- Substituting these values of '$a_i$' in equation (ii) and rearranging the terms yields

$$P(t) = (2t^3 - 3t^2 + 1)P(0) + (-2t^3 + 3t^2)P(1) + (t^3 - 2t^2 + t)P'(0) + (t^3 - t^2)P'(1)$$

- The values of P(0), P(1), P'(0), P'(1) are called geometric coefficients and represent the known vector quantities in the above equation
- The polynomial coefficients of these vector quantities are commonly known as blending functions By varying parameter t in these blending function from 0 to 1 several points on curve segments can be found

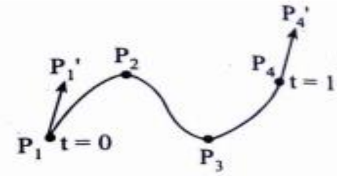
**Alternative Way:**

i. Hermite spline curve is interpolation spline curve (curve passes through control point)



ii. It uses cubic polynomial function (to make Hermite function four point is necessary)

iii. To make Hermite function it uses four point $P_1$, $P_4$, $P_1'$, $P_4'$. Where $P_1$ and $P_4$ are position vector and $P_1'$ and $P_4'$ are tangent vectors (first order derivative) which show direction of the curve.




Let Q(t) is the curve $t \in [0, 1]$

$Q(t) = [x(t) \quad y(t) \quad z(t)]$, $t \in [0, 1]$ where all points satisfy cubic parametricity.

The general curve equation is,

$p(t) = at^3 + bt^2 + ct + d$ where $0 \le t \le 1$

So,

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

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

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

$$Q(t) = [t^3 \quad t^2 \quad t \quad 1] . \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$$

Q(t) = T.C.

C = M.G.

M = basis matrix that provides blending function.

G = Geometric vector

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

$= [t^3 t^2 \quad t \quad 1]. M_H.G_H$

$Q_x(t) = P_1(t) = [0\ 0\ 0\ 1]\ M_H.G_H$

$t = 0$

$Q_x(t) = P_4(t)\ [1\ 1\ 1\ 1]\ M_H.G_H$

$t = 1$

$Q'_x(t) = R_1(t) = [3t^2\ 2t\ 1\ 0] M_H.G_H$

$t = 0$

$= [0\ 0\ 1\ 0]\ M_H.G_H$

$Q_H'(t) = R_a(t) = [3\ 2\ 1\ 0]\ M_H.G_H$

$t = 1$

$$\begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} M_H G_H$$

$$M_H.G_H = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

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

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

$$Q(t) = TM_HG_H$$

$$Q(t) = [t^3\ t^2\ t\ 1]\begin{bmatrix} 2 & -1 & 1 & 1 \\ -3 & 3 & -2 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

$$= (2t^3 - 3t^2 + 1)P_1 + (-2t^2 + 3t^2)P_2 + (t^3 - 2t^2 + t)R_1 + (t^3 - t^2)R_4$$

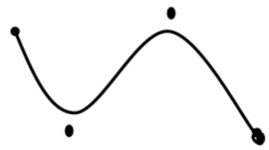
$$= P_1H_0(t) + P_4H_1(t) + R_1H_2(t) + R_4H_3(t)$$

Where $H_0(t)$, $H_1(t)$, $H_2(t)$, $H_3(t)$ are Hermite blending function.

# Chapter 5 Curve Modeling

**Bezier curves:**

**i.** Bezier curve is approximate spline curve.

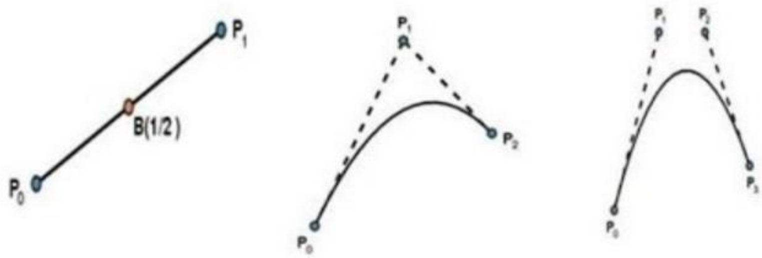


Approximation Curves

**ii.** Instead of end points and tangents, we have four control points in the case of cubic Bezier curve.

**iii.** It has Bernstein polynomial function (Its own polynomial function to provide continuity)

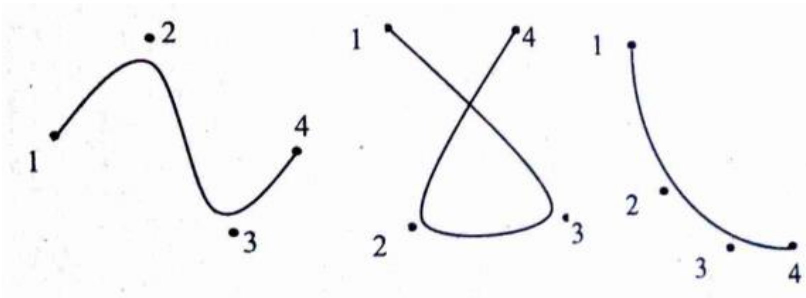
**iv.** All control point's uses to make Bezier Curve. Curve doesn't go out of polygon boundary (Convex hull)




- Bezier splines are highly useful, easy to implement and convenient for curve and surface design so are widely available in various CAD systems, graphics packages, drawing and painting packages.

The Bezier curve was developed by the French engineer Pierre Bezier for use in the design of Renault automobile bodies. A Bezier curve 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. As with the interpolation splines, a Bezier curve can be specified with boundary conditions, with a characterizing matrix, or with blending functions.

The control points are blended using Bernstein polynomials to compute a set of position vectors **P**(u) which are then joined by straight line segments to get the curve.




Suppose we are given n + 1 control-point positions: pk = ($x_k$ , $y_k$ , $z_k$ ), with k varying from 0 to n. These coordinate points can be blended to produce the following position vector P(u), which describes the path of an approximating Bezier polynomial function between $p_0$ and $p_n$ .

$$\mathbf{P}(u) = \sum_{k=0}^{n} \mathbf{p}_k \, BEZ_{k,n}(u), \qquad 0 \le u \le 1$$

The Bezier blending functions $BEZ_{k,n}(u)$ are the Bernstein polynomials:

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

Where the ***C (n, k)*** are the binomial coefficients: $C(n,k) = \frac{n!}{k!\,(n-k)!}$

This, vector equation represents a set of three parametric equations for the individual curve coordinates:

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

$$\mathbf{y}(u) = \sum_{k=0}^{n} \mathbf{y}_k \, BEZ_{k,n}(u)$$

$$\mathbf{z}(u) = \sum_{k=0}^{n} \mathbf{z}_k \, BEZ_{k,n}(u)$$

If we take n = 3, then number of control points = n +1 = 4.

Then ,

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

Then the above equations become

$$P_x(u) = x_0\, BEZ_{0,3}(u)+x_1BEZ_{1,3}(u)+x_2BEZ_{2,3}(u)+x_3BEZ_{3,3}(u)$$

$$P_y(u) = y_0\, BEZ_{0,3}(u)+y_1BEZ_{1,3}(u)+y_2BEZ_{2,3}(u)+y_3BEZ_{3,3}(u)$$

$$P_z(u) = z_0\, BEZ_{0,3}(u)+z_1BEZ_{1,3}(u)+z_2BEZ_{2,3}(u)+z_3BEZ_{3,3}(u)$$

where

$$BEZ_{0,3}(u) = C(3,0)\, u^0(1-u)^3 = (1-u)^3$$

$$BEZ_{1,3}(u) = C(3,1)\, u^1(1-u)^2 = 3u(1-u)^2$$

$$BEZ_{2,3}(u) = C(3,2)u^2(1-u) = 3u^2(1-u)$$

$$BEZ_{3,3}(u) = C(3,3)u^3(1\text{-}u)^0 = u^3$$

When u = 0 then $P(u) = P_0$ and

when u =1 then $P(u) = P_3$

$$P(u) = (1\text{-}u)^3 P_0 + 3u\,(1\text{-}u)^2 P_1 + 3u^2 (1\text{-}u)\, P_2 + u^3 P_3$$

in Matrix Form

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

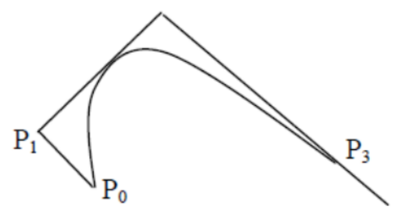
# Chapter 5 Curve Modeling

## Properties of Bezier Curve

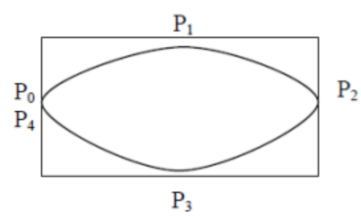
- It always passes through the first and last control points.
- It lies within the convex hull (convex polynomial boundary) of the control points. This follows from the properties of Bezier blending function: they are positive and their sum is always 1, i.e.

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

- Bezier curve lies in the convex hull of the control points which ensure that the curve smoothly follows the control Points




- The degree of the polynomial defining the curve segment is one less than the number of defining control polygon points.
  For 4 control points the degree of polynomial is 3 i.e. cubic Bezier curve.
- Closed curves can be generated by specifying the first and last control points at the same position.




- Specifying multiple control points at a single position gives more weight to that position and as a result, the curve is pulled nearer to that point.
- Collinear control points with same coordinate values produce a point.
- Control point do not have local control over the shape of the curve.
- Complicated curves are formed by piecing several sections of lower degrees together
- The slope at the beginning of the curve is along the line joining the first two control points, and the slope at the end of the curve is along the line joining the last two points

**Drawbacks:**

- The degree of Bezier curve depends on number of control points.
- Bezier curve exhibit global control property means moving a control point alters the shape of the whole curve.

**Q.** Calculate (x, y) coordinates of Bezier curve described by the following 4 control points: (0, 0), (1, 2), (3, 3), (4, 0).

Step by step solution

For four control points, n = 3.

First calculate all the blending functions,

$B_{kn}$ using the formula:

$$\boxed{B_{kn}(u) = C(n,k)\,u^k(1-u)^{n-k} = \frac{n!}{k!\cdot(n-k)!}\,u^k(1-u)^{n-k}}$$

$$B_{03}(u) = \frac{3!}{0!\cdot 3!}\,u^0(1-u)^3 = 1\cdot u^0(1-u)^3 = \mathbf{(1-u)^3}$$

$$B_{13}(u) = \frac{3!}{1!\cdot 2!}\,u^1(1-u)^2 = 3\cdot u^1(1-u)^2 = \mathbf{3u\cdot(1-u)^2}$$

$$B_{23}(u) = \frac{3!}{2!\cdot 1!}\,u^2(1-u)^1 = 3\cdot u^2(1-u)^1 = \mathbf{3u^2(1-u)}$$

$$B_{33}(u) = \frac{3!}{3!\cdot 0!}\,u^3(1-u)^0 = 1\cdot u^3(1-u)^0 = \mathbf{u^3}$$

$\underline{u = 0.0}$ $x(0) = \sum_{k=0}^{n} x_k B_{kn}(0) = x_0 B_{03}(0) + x_1 B_{13}(0) + x_2 B_{23}(0) + x_3 B_{33}(0) =$

$= 0\cdot(1-u)^3 + 1\cdot 3u\cdot(1-u)^2 + 3\cdot 3u^2(1-u) + 4\cdot u^3 =$

$= 0\cdot 1 + 1\cdot 0 + 3\cdot 0 + 4\cdot 0 =$

$= 0$

$y(0) = \sum_{k=0}^{n} y_k B_{kn}(0) = y_0 B_{03}(0) + y_1 B_{13}(0) + y_2 B_{23}(0) + y_3 B_{33}(0) =$

$= 0\cdot(1-u)^3 + 2\cdot 3u\cdot(1-u)^2 + 3\cdot 3u^2(1-u) + 0\cdot u^3 =$

$= 0$

$\underline{u = 0.2}$ $\quad x(0.2) = \sum_{k=0}^{n} x_k B_{kn}(0.2) = x_0\, B_{03}(0.2) + x_1\, B_{13}(0.2) + x_2\, B_{23}(0.2) + x_3\, B_{33}(0.2) =$

$= 0\cdot(1 - u)^3 + 1\cdot 3u\cdot(1 - u)^2 + 3\cdot 3u^2(1 - u) + 4\cdot u^3 =$
$= 0\cdot 0.512 + 1\cdot 0.384 + 3\cdot 0.096 + 4\cdot 0.008 =$
$= 0.7$

$y(0.2) = \sum_{k=0}^{n} y_k B_{kn}(0.2) = y_0\, B_{03}(0.2) + y_1\, B_{13}(0.2) + y_2\, B_{23}(0.2) + y_3\, B_{33}(0.2) =$

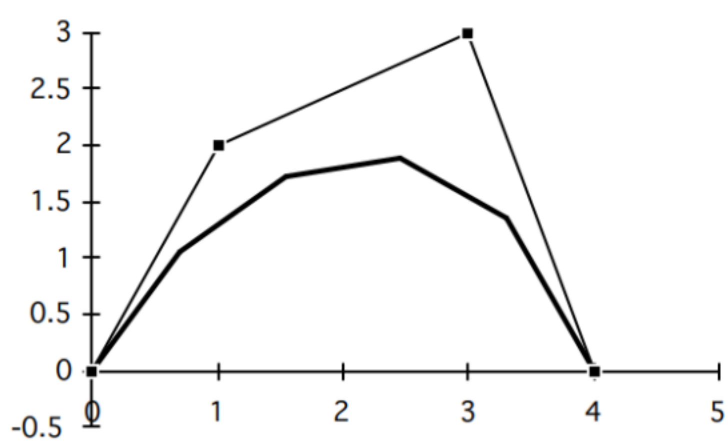
$= 0\cdot(1 - u)^3 + 2\cdot 3u\cdot(1 - u)^2 + 3\cdot 3u^2(1 - u) + 0\cdot u^3 =$
$= 0\cdot 0.512 + 2\cdot 0.384 + 3\cdot 0.096 + 0\cdot 0.008 =$
$= 1.1$

etc, giving:

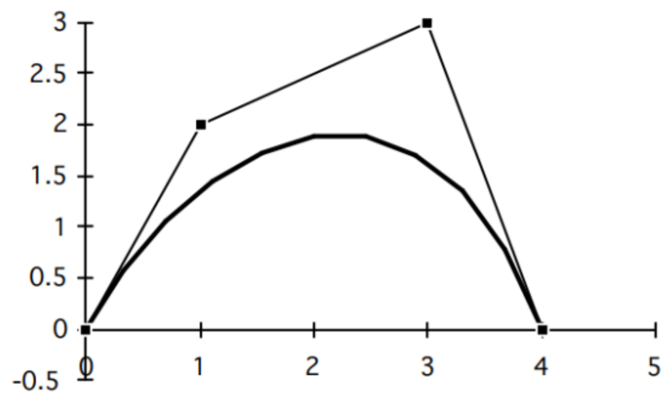
$u = 0.0$ $x(u) = 0.0$ $\quad y(u) = 0.0$
$u = 0.2$ $x(u) = 0.7$ $\quad y(u) = 1.1$
$u = 0.4$ $x(u) = 1.55$ $\quad y(u) = 1.7$
$u = 0.6$ $x(u) = 2.45$ $\quad y(u) = 1.9$
$u = 0.8$ $x(u) = 3.3$ $\quad y(u) = 1.3$
$u = 1.0$ $x(u) = 4.0$ $\quad y(u) = 0.0$

$(x(u), y(u))_{u = 0,1}$ are coordinates of the curve points.

The plot below shows control points (joined with a thin line) and a Bezier curve with 6 steps



The plot shows control points (joined with a thin line) and a Bezier curve with 11 steps; note smoother appearance of this curve in comparison to the previous one.




**Q. Find the equation of the Bezier curve which passes through (0,0) and (-4,2) and controlled through (14,10) and (4,0)**

**Solution:**

$$B(t) = \Sigma^{k}_{i=0} \; P_k \, B_{k,n}(t) \qquad \text{where, } 0<=t<=1$$

It is given that curve passes (0,0) and (-4,2), means starting point of the curve is $P_0$ = (0,0) and $P_3$ = (-4,2)

Whereas the curve is controlled by $P_1$ (14,10) and $P_2$ (4,0)

Number of control points,

$P_0$=(0,0) , $P_1$=(14,10) , $P_2$=(4,0) , $P_3$=(-4,2)

Degree of equation n =Control_point -1 = 4 -1 **= 3**

$$B(t) = P_0B_{0,3}(t)+ P_1B_{1,3}(t) + P_2B_{2,3}(t)+ P_3B_{3,3}(t)$$

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

$B(t) = P_0 (1-t)^3 + P_1 3t (1-t)^2 + P_2 3t^2 (1-t)^1 + P_3 t^3$

Now, Using this equation lets calculate equation for x control point

$P_0=(0,0)$ , $P_1=(14,10)$ , $P_2=(4,0)$ , $P_3=(-4,2)$

$= 0(1-t)^3 + (14)\ 3t (1-t)^2 + (4)\ 3t^2 (1-t) + (-4)\ t^3$

$= 42t(t^2+1-2t) + 12t^2(1-t) - 4t^3$

$= 42t^3+42t-84t^2+12t^2-12t^3-4t^3$

$\mathbf{=26t^3-72t^2+42t}$

Using this equation lets calculate equation for y control points

$= 0(1-t)^3 + (10)\ 3t (1-t)^2 + (0)\ 3t^2 (1-t) + (2)\ t^3$

$=30t\ (1+t^2-2t)+2t^3$

$=30t+30t^3-60t^2+2t^3$

$= \mathbf{32t^3 -60t^2+30t}$

| t | X(t) | Y(t) |
|---|---|---|
| 0 | | |
| 0.2 | | |
| 0.4 | | |
| 0.6 | | |
| 0.8 | | |
| 1 | | |

Q. ***Find the Bezier curve which passes through (0,0,0) and (-2, 1, 1) and is controlled by (7,5,2) and (2,0,1).***

**B-spline Curve:**

B-spline curve was developed to overcome the limitation or demerits of bezier curve. Demerits of Bezier curve are as follows:

1. Polynomial degree is decided by control points

   If C.P. = 5 then

   P.D. = 5–1 = 4

2. Blending function is non-zero for all parameter value 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 within a curve.

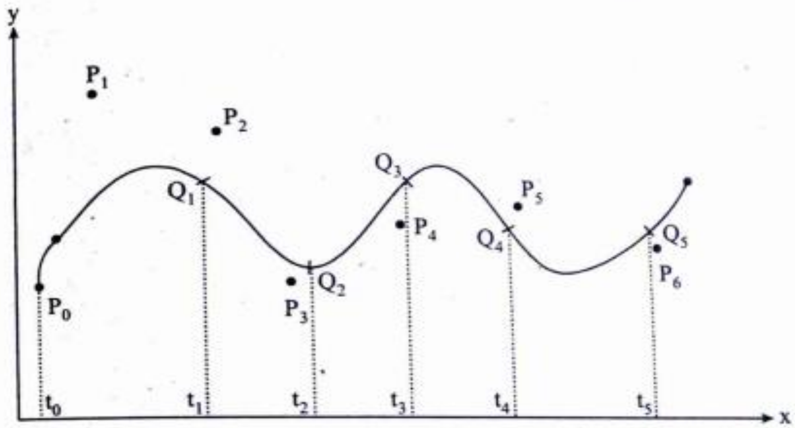
**Properties of B-spline curve:**

1. B-spline approximate spline curve with local effect. In this curve, each control point affects the shape of the curve only over range of parameter values where its associated basis function 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) function 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 without 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 within 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 |
|---|---|---|
| $Q_1$ | $P_0\ P_1\ P_2$ | $t_0 = 0$, $t_1 = 1$ |
| $Q_2$ | $P_1\ P_2\ P_3$ | $t_1 = 1$, $t_2 = 2$ |
| $Q_3$ | $P_2\ P_3\ P_4$ | $t_2 = 2$, $t_3 = 3$ |
| $Q_4$ | $P_3\ P_4\ P_5$ | $t_3 = 3$, $t_4 = 4$ |
| $Q_5$ | $P_4\ P_5\ P_6$ | $t_4 = 4$, $t_5 = 5$ |

There will be 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 function of the parameter u, a B-spline curve is given by

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

$N_{i,k}(u)$ is B-spline basis function

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

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

The parameter u various form 0 to n–k+2 along the P(u).

So there are some conditions for finding the KNOT VALUES [X]

$X_i (0 \le i \le n + k)$: Knot values

$X_i = 0$ if $i < k$

$X_i = i - k + 1$ if $k \le i \le n$

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

So as B-spline curve has Recursive Equation. So we stop at

$N_i, K(u) = 1$ if $X_i \le u\ x_{i+1}$

$= 0$ otherwise

**Example:**

$n = 5, k = 3$

then $X_i(0 \le i \le 8)$ knot values

$X_i\{0,0,0,1,2,3,4,4,4,4,\}$

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

When $i = 0, k = 3$ so $i < k$ is true

$X_0 = 0$

$i = 1, k = 3\ X_1 = 0$

$i = 2, k = 3\ X_2 = 0$

$i = 3, k = 3\ X_3 = i-k+1 = 3-3+1$

$X_3 = 1$

$i = 4, k = 3\ X_4 = i-k+1 = 4-3+1$

$X_4 = 2$

$i = 8, k = 3\ X_8 = i > n\ n - k + 2 = 5 - 3 + 2 = 4$

In this way we will calculate