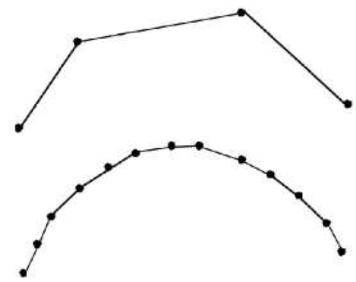
**<u>1.1 Introduction</u>**: In computer graphics, we often need to draw different types of objects onto the screen. Objects are not flat all the time and we need to draw curves many times to draw an object. **<u>"Computers can not draw curves."</u>** A curve consists of small line segments. The more points/line segments that are used, the smoother the curve.



"<u>What is the use of curves</u>"

- Representation of "irregular surfaces" , Example: Auto industry (car body design)
- Artist's representation, Clay / wood models
- Digitizing
- Surface modeling ("body in white")
- Scaling and smoothening
- Tool and die Manufacturing

**<u>1.2 Types of Curves</u>**

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**.

a) <u>Implicit Curves:</u> 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. It can represent multivalued curves for multiple y values for an x value. A common example is the circle, whose implicit representation is $x^2 + y^2 - R^2 = 0$

b) <u>Explicit Curves:</u> A mathematical function y = f(x) can be plotted as a curve. Such a function is the explicit representation of the curve. The explicit representation is not general, since it cannot represent vertical lines and is also single-valued. For each value of x, only a single value of y is normally computed by the function. A common example is the circle, whose explicit representation is $y^2 = R^2 - x^2$, Thus, y has two values either $y = + (R^2 - x^2)^{1/2}$ or $y = - (R^2 - x^2)^{1/2}$

**c)** <u>Parametric Curves:</u> Curves having parametric form are called parametric curves. The explicit and implicit curve representations can be used only when the function is known. In practice the parametric curves are used. A two-dimensional parametric curve has the following form − P(t) = f(x(t), y(t)), The functions x(t) and y(t) become the x, y coordinates of any point on the curve, and the points are obtained when the parameter 't' is varied over a certain interval [a, b], normally [0, 1]. Parametric curves are mostly used in CAD/CAM and classified as: Analytical & Synthetically.

**Analytical curves** Analytical curves are defined as those that can be described by analytical equations such as lines, circles and conics. Analytical curves provide very compact forms to represent shapes and simplify the computation of related properties such as areas and volumes. Analytical curves are not attractive to deal with interactively Analytical curves are points, lines, arcs and circles, fillets and chambers and also conics like parabola, hyperbola, ellipse, etc.

**Synthetic curves** A synthetic curve is defined as that can be described by a set of data points( control points) such as splines and bazier curves. Synthetic curves provide designers with greater flexibility and control of curve shapes by changing the positions of control points. Global and local control of a curve is possible. Synthetic curves are attractive to deal with interactively Synthetic curves include various types of splines mainly: Hermit cube or parametric cube curve or cubic spline, Bezier curve, B- spline and NURBS (Non- Uniform Rotation B- Spline)

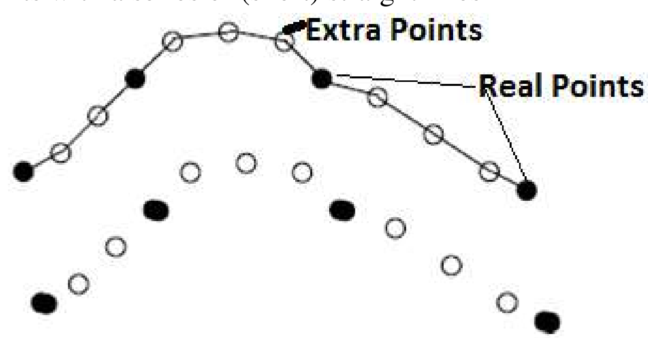
**1.3 "Why Synthetic curve?"**

a) "Problem: How to represent a curve easily and efficiently"

- •storing a curve as many small straight line segments
- •doesn't work well when scaled
- •inconvenient to have to specify so many points
- •need lots of points to make the curve look smooth
- •working out the equation that represents the curve
- •equations are difficult to derive for complex curves
- •moving an individual point requires re-calculation of the entire curve

b) Solution is "Interpolation"

- •Define a small number of points
- •Use a technique called "interpolation" to invent the extra points for us.
- •Join the points with a series of (short) straight lines




**c)** "The need for Smoothness"

- •So far, mostly polygons can approximate any geometry, but Only approximate
- •Need lots of polygons to hide discontinuities Storage problems
- •Math problems
- •Not very convenient as modeling tool
- •Gets even worse in animation

**d)** "Requirements"

- •Want mathematical smoothness
- •Some number of continuous derivatives of P
- •Local control
- •Local data changes have local effect
- •Continuous with respect to the data
- •No wiggling if data changes slightly
- •Low computational effort

**e)** "A Solution"

- Use SEVERAL polynomials

- Complete curve consists of several pieces
- All pieces are of low order
- Third order is the most common
- Pieces join smoothly
- This is the idea of spline curves
- or just "splines"

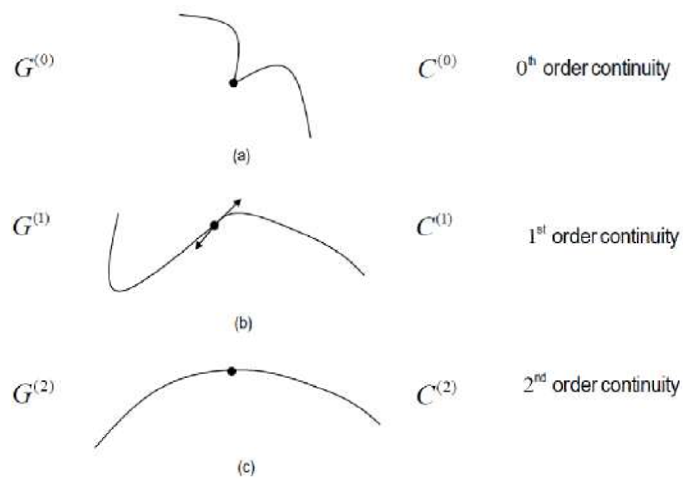
**f)** "Continuity"

Parametric continuity $C^x$:

- Only P is continuous: $C^0$ : Positional continuity
- P and first derivative dP/du are continuous: $C^1$ : Tangential continuity
- P + first + second: $C^2$ , Curvature continuity

Geometric continuity $G^x$: Only directions have to match

**g)** "Order of continuity"




**h)** "Curve Fittings"

Often designers will have to deal with information for a given object in the form of coordinate data rather than any geometric equation. In such cases it becomes necessary for the designers to use mathematical techniques of curve fitting to generate the necessary smooth curve that satisfies the requirements. Airfoil Section Curve fitted with Data Points

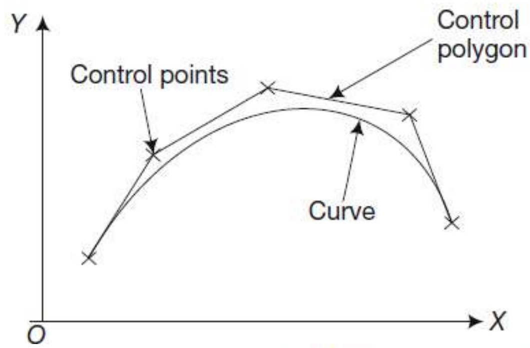
## 1.4 "Synthetic curves"

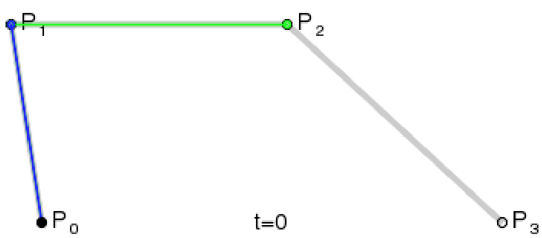
Easy to enter the data and easy to control the continuity of the curves to be designed. Requires much less computer storage for the data representing the curve. Having no computational problems and faster in computing time.

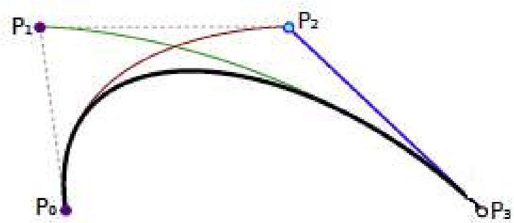
a) •Bezier curves
b) •Hermite cubic spline
c) •B-spline curves
d) •Rational B-splines (including Non-uniform rational B-splines – NURBS)

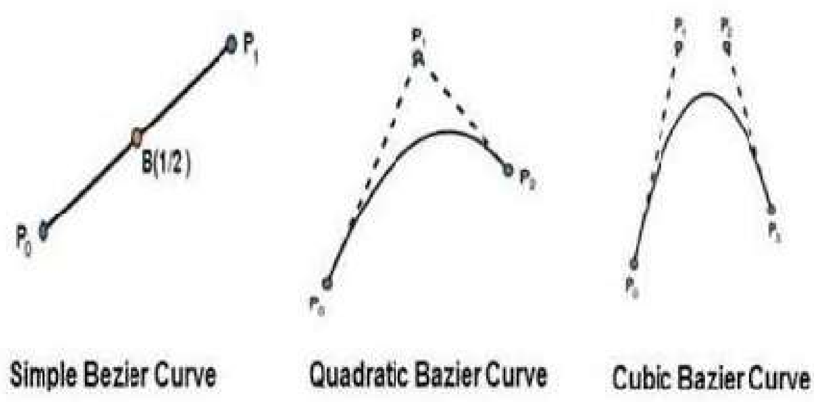
## a) Bezier curve

- Bezier curve is discovered by the French engineer Pierre Bézier. A Bezier curve is a mathematically defined curve used in two-dimensional graphic applications.
- The curve is defined by points: the initial position and the terminating position (which are called "anchors") and middle points (which are called "handles or control points").
- The shape of a Bezier curve can be altered by moving the control points.




**A linear Bézier curve B(t) = (x(t), y(t)) is a line segment joining two control points $b_0$ ($x_0$, $y_0$) and $b_1$ ($x_1$, $y_1$) and the curve can be written in**

$$B(t) = (1 - t) * b_0 + t * b_1 \text{ for } t \in [0,1]$$

*That means* $x(t) = (1-t)\, x_0 + t\, x_1$ and $y(t) = (1-t)\, y_0 + t\, y_1$

Example:

The Bézier form for the linear segment passing through points $b_0(1, 2)$ and $b_1$ (3, 4) is

$B(t) = (1 - t) * b_0 + t * b_1$ for $t \in [0,1]$

$= (1 - t)\,(1, 2) + t\,(3, 4)$
*Hence* $x(t) = (1-t) + 3t = 1 + 2t$ *and*
$y(t) = 2(1 - t) + 4t = 2 + 2t$

**Similarly Quardratic Bézier curve B(t) for three control points $b_0$, $b_1$ and $b_2$ is**

$B(t) \quad = (1 - t)^2 * b_0 + 2(1-t) * b_1 + t^2 * b_2$ *for* $t \in [0,1]$
*Hence* $x(t) = (1 - t)^2 * x_0 + 2(1-t) * x_1 + t^2 * x_2$ *and*
$y(t) = (1 - t)^2 * y_0 + 2(1-t) * y_1 + t^2 * y_2$

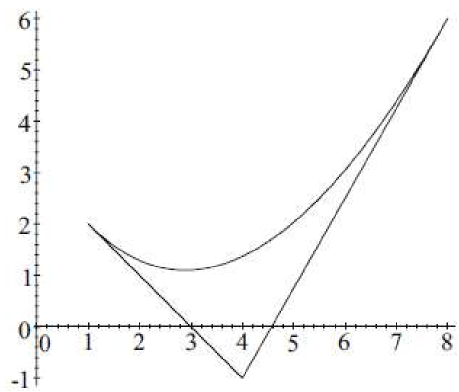
**Example**

The parametric form of the quadratic Bézier curve $\mathbf{B}(t)$ with control points $\mathrm{b}_0(1,2)$, $\mathrm{b}_1(4,-1)$, and $\mathrm{b}_2(8,6)$ is $(x(t), y(t))$ where

$$\begin{aligned} x(t) &= (1-t)^2(1) + 2(1-t)t(4) + t^2(8) = 1 + 6t + t^2\ , \text{ and} \\ y(t) &= (1-t)^2(2) + 2(1-t)t(-1) + t^2(6) = 2 - 6t + 10t^2\ . \end{aligned}$$

The point $\mathbf{B}(0.5)$ is obtained by substituting $t = 0.5$ into the equations to give $x(0.5) = 4.25$ and $y(0.5) = 1.5$, that is, $\mathbf{B}(0.5) = (4.25, 1.5)$. Alternatively, the coordinates of the point $\mathbf{B}(0.5)$ can be evaluated using the vector form of the curve

$$\begin{aligned} \mathbf{B}(t) &= (1-0.5)^2(1,2) + 2(1-0.5)(0.5)(4,-1) + (0.5)^2(8,6) \\ &= 0.25(1,2) + 0.5(4,-1) + 0.25(8,6) = (4.25, 1.5)\ . \end{aligned}$$




Quadratic Bézier curve with control points $\mathrm{b}_0(1,2)$, $\mathrm{b}_1(4,-1)$,
Suppose *four* control points $\mathrm{b}_0$, $\mathrm{b}_1$, $\mathrm{b}_2$, and $\mathrm{b}_3$ are specified, then the *cubic Bézier curve* is defined to be

$$\mathbf{B}(t) = (1-t)^3\mathrm{b}_0 + 3(1-t)^2t\mathrm{b}_1 + 3(1-t)t^2\mathrm{b}_2 + t^3\mathrm{b}_3, \quad t \in [0,1]\ . \qquad (6.2)$$

As in the quadratic case, the polygon obtained by joining the control points in the specified order is called the *control polygon*.

The parametric form of the quadratic Bézier curve $\mathbf{B}(t)$ with control points $\mathbf{b}_0(1,2)$, $\mathbf{b}_1(4,-1)$, and $\mathbf{b}_2(8,6)$ is $(x(t), y(t))$ where

$$\begin{aligned} x(t) &= (1-t)^2(1) + 2(1-t)t(4) + t^2(8) = 1 + 6t + t^2 \text{ , and} \\ y(t) &= (1-t)^2(2) + 2(1-t)t(-1) + t^2(6) = 2 - 6t + 10t^2 \text{ .} \end{aligned}$$

The point $\mathbf{B}(0.5)$ is obtained by substituting $t = 0.5$ into the equations to give $x(0.5) = 4.25$ and $y(0.5) = 1.5$, that is, $\mathbf{B}(0.5) = (4.25, 1.5)$. Alternatively, the coordinates of the point $\mathbf{B}(0.5)$ can be evaluated using the vector form of the curve

$$\begin{aligned} \mathbf{B}(t) &= (1-0.5)^2(1,2) + 2(1-0.5)(0.5)(4,-1) + (0.5)^2(8,6) \\ &= 0.25(1,2) + 0.5(4,-1) + 0.25(8,6) = (4.25, 1.5) \text{ .} \end{aligned}$$

$$\mathbf{B}(t) = (1-t)^3\mathbf{b}_0 + 3(1-t)^2 t\mathbf{b}_1 + 3(1-t)t^2\mathbf{b}_2 + t^3\mathbf{b}_3, \quad t \in [0,1] \text{ .}$$

- In some of the literature the nomenclature generally used is

  B---> p , t---> u, and $b_i$ ----> $p_i$

- Bézier chose Bernestein polynomials as the basis functions for the curves.

$$p(u) = \sum_{i=0}^{n} p_i \, B_{i,n}(u) \qquad u \in [0,1]$$

- Based on these basis functions, the equation for the Bézier curve is given by **:**

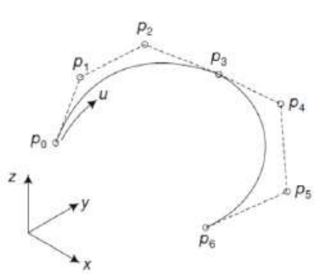
$\boldsymbol{p}(u) = (1-u)^3 \boldsymbol{p_0} + 3\,u\,(1-u)^2\, \boldsymbol{p_1} + 3\,u^2\,(1-u)\, \boldsymbol{p_2} + u^3 \boldsymbol{p_3}$

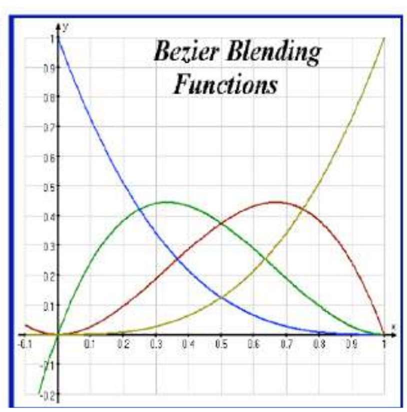
  - This can be written in matrix form as

$$p(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

  - p($u$) = {**U**} **[$\mathbf{M_B}$] [P]**

$$p(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$



Bezier curves have the following properties –

- They generally follow the shape of the control polygon, which consists of the segments joining the control points.
- They always pass through the first and last control points.
- They are contained in the convex hull of their defining control points.
- The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- A Bezier curve generally follows the shape of the defining polygon.
- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.
- They are invariant under an affine transformation.
- Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
- A given Bezier curve can be subdivided at a point $t=t_0$ into two Bezier segments which join together at the point corresponding to the parameter value $t=t_0$.

***Example***: A cubic Bezier curve is described by the four control points: (0,0), (2,1), (5,2), (6,1). Find the tangent to the curve at t = 0.5.

$$P(t) = [t^3 \quad t^2 \quad t \quad 1] \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

where, $V_0 = (0,0)$
$V_1 = (2,1)$
$V_2 = (5,2)$
$V_3 = (6,1)$

The tangent is given by the derivative of the general equation above,

$$P'(t) = [3t^2 \quad 2t \quad 1 \quad 0] \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

At t = 0.5, we get,

$$P'(t) = [3(.5)^2 \quad 2(.5) \quad 1 \quad 0] \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

$$= [6.75 \quad 1.5 \quad 0 \quad 1]$$

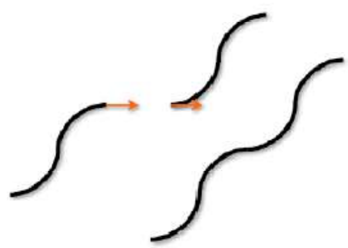
b) **Hemiot Cubic Spline**

- Hermite cubic splines are the more general form of curves that can be defined through a set of vertices (points).
- A spline is a piecewise parametric representation of the geometry of a curve with a specified level of parametric continuity.
- Each segment of a Hermite cubic spline is approximated by a parametric cubic polynomial to maintain the $C'$ continuity.

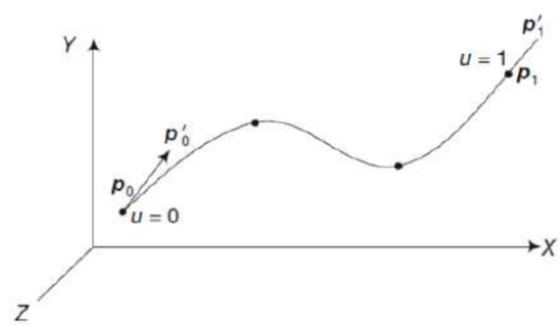
We want curves that fit together smoothly.
To accomplish this, we would like to specify a curve by providing:

- The endpoints
- The 1st derivatives at the endpoints

The result is called a *Hermite Curve.*



**Hermite Cubic Spline Curve**




- The parametric equation of a Hermite cubic spline is given by

$$p(u) = \sum_{i=0}^{3} C_i u^i \qquad u \in [0, 1]$$

- In an expanded form it can be written as

$$p(u) = \boldsymbol{C_0} + \boldsymbol{C_1}\, u^1 + \boldsymbol{C_2}\, u^2 + \boldsymbol{C_3}\, u^3$$

- Where *u* is a parameter, and $\boldsymbol{C_i}$ are the polynomial coefficients.

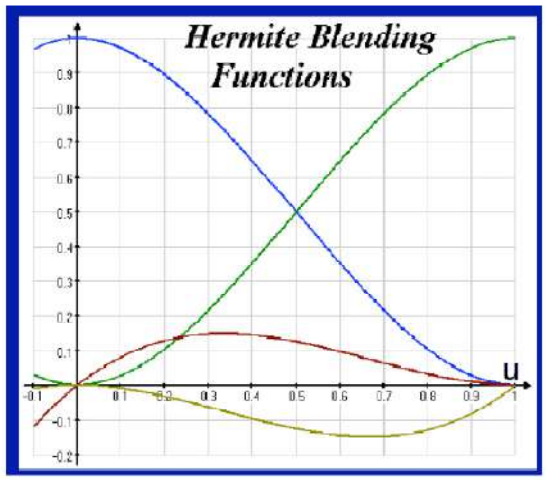
- In matrix form

$$p(u) = \begin{bmatrix} u^3 & u^2 & u^1 & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_0' \\ p_1' \end{bmatrix}$$

- p($u$) = {U} [M] [P]

$$p(u) = \begin{bmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{bmatrix}^T \begin{bmatrix} p_1 \\ p_2 \\ \nabla p_1 \\ \nabla p_2 \end{bmatrix}$$

**4 Basis Functions**




***Example 5***: A parametric cubic curve passes through the points (0,0), (2,4), (4,3), (5, -2) which are parametrized at t = 0, ¼, ¾, and 1, respectively. Determine the geometric coefficient matrix and the slope of the curve when t = 0.5.

***Solution***: The points on the curve are

(0,0) at t = 0
(2,4) at t = ¼
(4,3) at t = ¾
(5,-2) at t = 1

Substituting in equation (4.15), we get,

$$\begin{pmatrix} 0 & 0 \\ 2 & 4 \\ 4 & 3 \\ 5 & -2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0.0156 & 0.0625 & 0.25 & 1 \\ 0.4218 & 0.5625 & 0.75 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P(0) \\ P(1) \\ P'(0) \\ P'(1) \end{pmatrix}$$

$$\begin{bmatrix} P(0) \\ P(1) \\ P'(0) \\ P'(1) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 5 & -2 \\ 10.33 & 22 \\ 4.99 & -26 \end{bmatrix}$$

The slope at t = 0.5 is found by taking the first derivative of the equation (4.15), as follows,

$$P'(t) = [3t^3 \quad 2t \quad 1 \quad 0] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 5 & -2 \\ 10.33 & 22 \\ 4.99 & -26 \end{bmatrix}$$

Therefore,

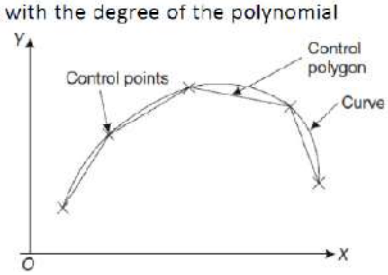
$P'(0.5) = [3.67 \quad -2.0]$, or

Slope = $\Delta x/\Delta y = -2.0/3.67 = -0.545$

## c) B-Splines

- In the case of Bezier curve, it is a single curve controlled by all the control points.
  - With an increase in the number of control points, the order of the polynomial representing the curve increases.

- B-spline generates a single piecewise parametric polynomial curve through any number of control points with the degree of the polynomial selected by the designer.




- B-spline curves have the flexibility of choosing the degree of the curve irrespective of the number of control points.

- With four control points, it is possible to get a cubic Bézier curve, while with B-spline curve one can get a linear, quadratic or cubic curve.

- B-spline also uses the basis (blending) functions and the equation is of the form

$$0 \le u \le u_{max}$$

$$p(u) = \sum_{i=0}^{n} p_i \, N_{i,k}(u)$$

- Where $N_{i,k}(u)$ are the basis functions for B-splines.

$$N_{i,1}(u) = 1 \quad \text{if } u_i \le u \le u_{i+1}$$
$$= 0 \quad \text{otherwise}$$

where $k$ controls the degree ($k$-1) of the resulting polynomial in u and also the continuity of the curve. The $u_i$ are the knot values, which relate the parametric variable u to the $\mathbf{p}_i$ control points.

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

- The plotting of B-spline curve is done by varying the parameter $u$ over the range of knot values $(u_{k-1}, u_{n+1})$.
- The knot vector adds flexibility to the curve and provides better control of its shape.
- Partition of Unity: For any knot span, $[u_i, u_{i+1}]$,
- Positivity:

$$\sum_{i=0}^{n} N_{i,k}(u) = 1 \qquad N_{i,k}(u) \geq 0 \text{ for all } i, k \text{ and } u.$$

- Local Support Property:

  $N_{i,k}(u) = 0$ if $u \notin [u_i, u_{i+k+1}]$

  This property can be deduced from the observation that $N_{i,k}(u)$ is a linear combination of $N_{i,k-1}(u)$ and $N_{i+1,k-1}(u)$.

- Continuity:
  $N_{i,k}(u)$ is $(k-2)$ times continuously differentiable, being a polynomial.

The Bezier-curve produced by the Bernstein basis function has limited flexibility.

- First, the number of specified polygon vertices fixes the order of the resulting polynomial which defines the curve.
- The second limiting characteristic is that the value of the blending function is nonzero for all parameter values over the entire curve.

The B-spline basis contains the Bernstein basis as the special case. The B-spline basis is non-global. B-spline curves have the following properties −

- The sum of the B-spline basis functions for any parameter value is 1.
- Each basis function is positive or zero for all parameter values.
- Each basis function has precisely one maximum value, except for k=1.
- The maximum order of the curve is equal to the number of vertices of defining polygon.
- The degree of B-spline polynomial is independent on the number of vertices of defining polygon.
- B-spline allows the local control over the curve surface because each vertex affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.
- The curve exhibits the variation diminishing property.
- The curve generally follows the shape of defining polygon.
- Any affine transformation can be applied to the curve by applying it to the vertices of defining polygon.
- The curve line within the convex hull of its defining polygon.

## d) NURBS: Non Uniform Rational Basis Splines

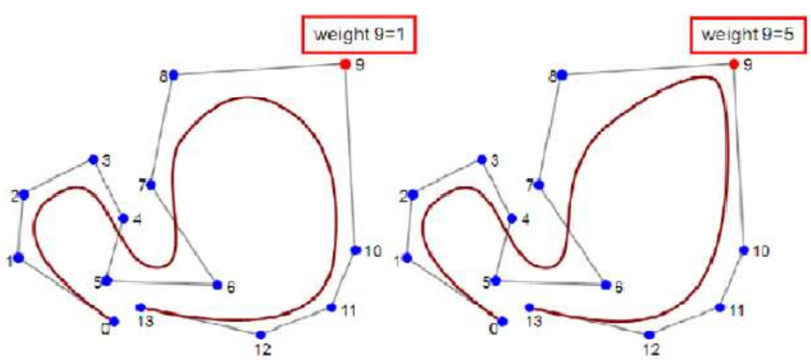
- A rational curve utilises the algebraic ratio of two polynomials.
- They are important in CAD because of their invariance when geometric transformations are applied.
- A rational curve defined by $(n+1)$ points is given by

$$\mathbf{p}(u) = \sum_{i=0}^{n} \mathbf{p}_i \, R_{i,k}(u) \qquad 0 \le u \le u_{max}$$

- Where $R_{i,k}(u)$ is the rational B-spline basis function and is given by

$$R_{i,k}(u) = \frac{h_i N_{i,k}(u)}{\sum_{i=0}^{n} h_i N_{i,k}(u)}$$

- **Non-uniform rational basis spline** (**NURBS**) is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces.
- It offers great flexibility and precision for handling both analytic (surfaces defined by common mathematical formulae) and modeled shapes.
- NURBS is flexible for designing a large variety of shapes by manipulating the control points and weights.
- Weights in the NURBS data structure determine the amount of surface deflection toward or away from its control point.
- Evaluation of NURBS is reasonably fast and numerically stable.




- Uniform cubic B-splines are the curves with the parametric intervals defined at equal lengths.
- The most common scheme used in all the CAD system is the non-uniform rational B-spline (commonly known as NURB), allowing a non-uniform knot vector.
- It includes both the Bézier and B-spline curves.

- Rational form of the B-splines can be written as

$$\mathbf{p}(u) = \frac{\sum_{i=0}^{n} w_i \ p_i \ N_{i,k}(u)}{\sum_{i=0}^{n} w_i \ N_{i,k}(u)}$$

- where $w_i$ is the weighing factor for each of the vertex.

- They have all of B-spline surface abilities. In addition they overcome the limitation of B-spline surfaces by associating each control point with a weight.
- Uniform representation for a large variety of curves and surfaces. This helps with the storage of geometric data.
- NURBS are invariant during geometric transformations as well as projections.
- NURBS is flexible for designing a large variety of shapes by manipulating the control points and weights.
- Weights in the NURBS data structure determine the amount of surface deflection toward or away from its control point.
- It makes it possible to create curves that are true conic sections.
- Surfaces based on conics, arcs or spheres can be precisely represented by a NURBS surface
- Evaluation of NURBS is reasonably fast and numerically stable.
- Number facilities available in NURBS such as knot insertion/ refinement/ removal, degree elevation, splitting, etc. makes them ideal to be used throughout the design process.
- NURBS surfaces can be incorporated into an existing solid model by "stitching" the NURBS surface to the solid model.
- Reverse engineering is heavily dependent on NURBS surfaces to capture digitized points into surfaces.

## Problems with NURBS

- Analytical curves and surfaces require additional storage.
- NURBS parameterization can often be affected by improper application of the weights, which may lead to subsequent problems in surface constructions.
- Not all geometric interrogation techniques work well with NURBS.

**Difference between Spline, B-Spline and Bezier Curves :**

| Spline | B-Spline | Bezier |
|---|---|---|
| A spline curve can be specified by giving a specified set of coordinate positions, called control points which indicate the general shape of the curve. | The B-Spline curves are specified by Bernstein basis function that has limited flexibiity. | The Bezier curves can be specified with boundary conditions, with a characterizing matrix or with blending function. |
| It follows the general shape of the curve. | These curves are a result of the use of open uniform basis function. | The curve generally follows the shape of a defining polygon. |
| Typical CAD application for spline include the design of automobile bodies, aircraft and spacecraft surfaces and ship hulls. | These curves can be used to construct blending curves. | These are found in painting and drawing packages as well as in CAD applications. |
| It possess a high degree of smoothness at the places where the polynomial pieces connect. | The B-Spline allows the order of the basis function and hence the degree of the resulting curve is independent of number of vertices. | The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. |
| 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 surfaces. | In B-Spline, there is local control over the curve surface and the shape of the curve is affected by every vertex. | It is a parametric curve used in related fields. |