UNIT - 2

Curves and Surfaces

Contents

Important Repeated Questions:

1. **Explain/Derive/Write the parametric equation and list characteristics of Bezier curves.** (W25 - Q2c, 07 marks) (S23 - Q3c, 07 marks) (S24 - Q2c OR, N/A marks) (W22 - Q2c OR, 07 marks) (S25 - Q2c, 07 marks) (W24 - Q3b, 04 marks)
2. **Differentiate between/Compare Analytical (analytic) and Synthetic curves.** (S23 - Q3b, 04 marks) (W23 - Q2b, N/A marks)
3. **Explain/Describe properties of Bezier curves (endpoints, control points, curve shape).** (W25 - Q2c, 07 marks) (S25 - Q2c, 07 marks)
4. **Derive/Explain Hermite Cubic spline curve (parametric equation).** (S23 - Q3c OR, 07 marks) (S24 - Q2c, N/A marks) (S22 - Q3c, 07 marks)
5. **Explain Hermite cubic spline curve with neat sketch, its characteristics and parametric equation.** (W24 - Q2c, 07 marks)
6. **Compare Hermite Cubic spline Curve, Bezier Curve and B-Spline Curve.** (S23 - Q3a OR, 03 marks)
7. **List/Explain advantages of Parametric representation and disadvantages of Non-Parametric representation.** (S24 - Q2a, N/A marks)
8. **Write equations of lines, circle, ellipse in parametric form.** (W22 - Q2a, N/A marks)

Legends: W- Winter, S- Summer, Q- Question and 03/04/07- Marks of Question

2.1 Curve Representation

- ***Curve*** is defined as the locus of point moving with one degree freedom.
- A curve can be represented by following two methods either by storing its analytical equation or by storing an array of co-ordinates of various points
- Curves can be described mathematically by following methods:
  - (i) Non-parametric form
    - (a). Explicit form
    - (b). Implicit form
  - (ii) Parametric form

2.1.1 Non-parametric form

- In this, the object is described by its co-ordinates with respect to current reference frame in use.

2.1.1.1 *Explicit form:* *(Clearly expressed)*

- In this, the co-ordinates of y and z of a point on curve are expressed as two separate functions of x as independent variable.

$$P = [x \quad y \quad z]$$
$$= [x \quad f(x) \quad g(x)]$$

2.1.1.2 *Implicit form:* *(Not clearly expressed)*

- In this, the co-ordinates of x, y and z of a point on curve are related together by two functions.

$$F(x, y, z) = 0$$
$$G(x, y, z) = 0$$

Limitation of nonparametric representation of curves are:

1. If the slope of a curve at a point is vertical or near vertical, its value becomes infinity or very large, a difficult condition to deal with both computationally and programming-wise. Other ill-defined mathematical conditions may result.
2. Shapes of most engineering objects are intrinsically independent of any coordinate system. What determines the shape of an object is the relationship between its data points themselves and not between these points and some arbitrary coordinate system.
3. If the curve is to be displayed as a series of points or straight line segments, the computations involved could be extensive.

2.1.2 Parametric form:

- In this, a parameter is introduced and the co-ordinates of x, y and z are expressed as functions of this parameters. This parameter acts as a local co-ordinate for points on curve.

$$P(u) = [x \quad y \quad z]$$
$$= [x(u) \quad y(u) \quad z(u)]$$

2.2 Classification of Curves

Curves can be classified as shown below mainly in two categories:

1. **Analytical Curves:**

- The curves which are defined by the analytical equations are known as analytical curves.
- Examples: line, circle, ellipse, parabola, hyperbola etc.

2. **Synthetic Curves:**

- The curves which are defined by set of data points are known as synthetic curves.
- Combination of polynomial segments to represent a desired curve is called a synthetic curve.
- These curves can be conveniently twisted, shaped or bent by changing the control points.
- Examples: Cubic spline curve, B-spline curve, Bezier curve etc.
- Synthetic curves take up where the analytic curves leave — the latter are not that efficient at geometric design of mechanical parts.
- Need for synthetic curves arise in two occasions:
  - When a curve is represented by a collection of measured data points,
  - When an existing curve must change to meet new design requirements the designer needs a curve representation that is directly related to the data points and is flexible enough to bend, twist or change the shape by changing one or more data points.
- Some examples of complex geometric design are: Car bodies, Ship hulls, Airplane fuselage and wings, Propeller blades, Shoe insoles, aesthetically designed bottles etc.
- When the curve pass through all the data points, then the curve is known as Interpolant curve. It sounds more logical but can be more unstable and ringing.
- When a smooth curve is approximated through the data points, then the curve is known as approximation curve. It turns out to be convenient.

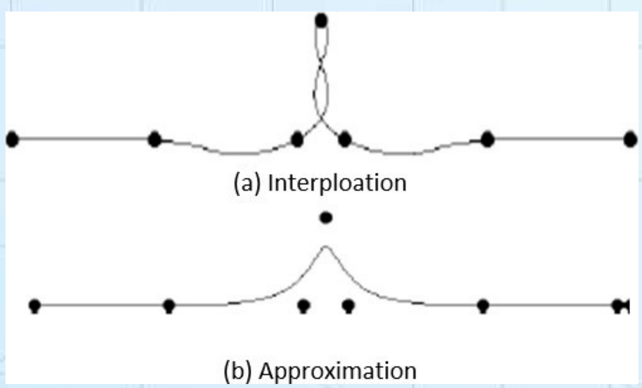

Fig.2.1 – Interpolant and Approximated Curves

- Synthetic curve pass through defined data points and can thus represented by polynomial equations.
- The parametric form for any synthetic curve is:

$$P(u) = C_3u^3 + C_2u^2 + C_1u + C_0$$

where, u = Parameter and Ci = Polynomial coefficients

2.3 Parametric representation of analytical curve

2.3.1 Lines

- ***A line*** can be represented in a parametric form by an input of two endpoints or by one point and a length and direction. Besides these, modifiers and filters can be used to represent these lines. The two important methods of establishing equations for lines are considered.

2.3.1.1 *A line connecting two endpoints in parametric form, having the values of parameter u as 0 and 1 at the endpoints:*

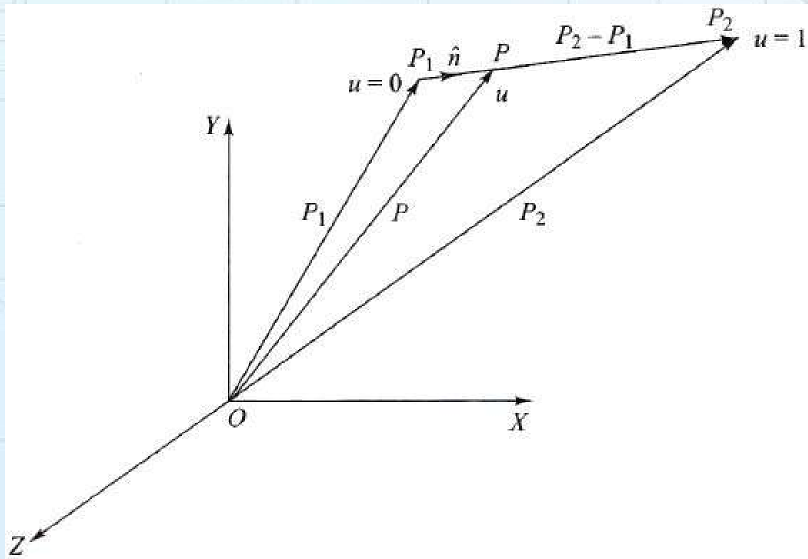

Fig.2.2 – Line between Two Endpoints

- Consider a line between two given endpoints P_1 (x_1, y_1, z_1) and $P_2(x_2, y_2, z_2)$ as shown in *Fig.2.2*. The position vectors for these two points would be P_1 and P_2. Consider any point P (x, y, z) on the line P_1P_2.

- Let a parametric system be introduced such that the point P is represented by the parameter u such that its values are 0 and 1 at endpoints P_1 and P_2 respectively.

- From ΔOPP_1, we can express the vector between endpoints P_1 and P as $P—P_1$. This vector represents the parameter u. The mapping between the Cartesian and parametric space can be done by the following relation :

$$\frac{Distance\ of\ P\ from\ P_1\ in\ cartesian\ space}{Distance\ of\ P\ from\ P_1\ in\ parametric\ space} = \frac{Endpoint\ length\ in\ cartesian\ space}{Endpoint\ length\ in\ parametric\ space}$$

$$\frac{P - P_1}{u} = \frac{P_2 - P_1}{1}$$

$$P - P_1 = u(P_2 - P_1)$$

$$P = P_1 + u(P_2 - P_1) \qquad \text{where, } 0 \le u \le 1$$

- This equation represents the vector equation of the line in parametric form. In scalar form, it can be written as:

$$x = x_1 + u(x_2 - x_1)$$

$$y = y_1 + u(y_2 - y_1)$$

$$z = z_1 + u(z_2 - z_1) \qquad \text{where, } 0 \le u \le 1$$

- Any point on the above line or its extension can be found by substituting the value of parameter u. The tangent vector of the line would be represented by :

$$P' = P_2 - P_1$$

- In scalar form these would have the coordinates expressed by the following equations:

$$x' = x_2 - x_1$$

$$y' = y_2 - y_1$$

$$z' = z_2 - z_1$$

- It can be seen that the tangent vector is independent of the parameter u, thus representing the constant slope of the line. Above equations are sufficient to represent all properties of the line for CAD/CAM representation. The length L and direction vector $\hat{n}$ can be calculated using the following equations :

$$L = |L_2 - L_1|$$

$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$\hat{n} = \frac{P_2 - P_1}{L}$$

2.3.1.2 *A line passing through a point and defined by a length and direction:*

- Consider a case, where a line is to pass through a point P_1 and has its length L and direction vector $\hat{n}$ defined as input as shown in *Fig.2.3*. The computation of the endpoint of this line needs to be done based on the above inputs. As seen earlier,

$$\hat{n} = \frac{P - P_1}{L}$$

$$P = P_1 + L(\hat{n}) \qquad \text{where, } -\infty \leq L \leq \infty$$

- Once the user inputs P_1, L and $\hat{n}$, the endpoint can be calculated as per the above equation. The two endpoints can then be expressed in parametric form with values of parameter u as 0 and 1.

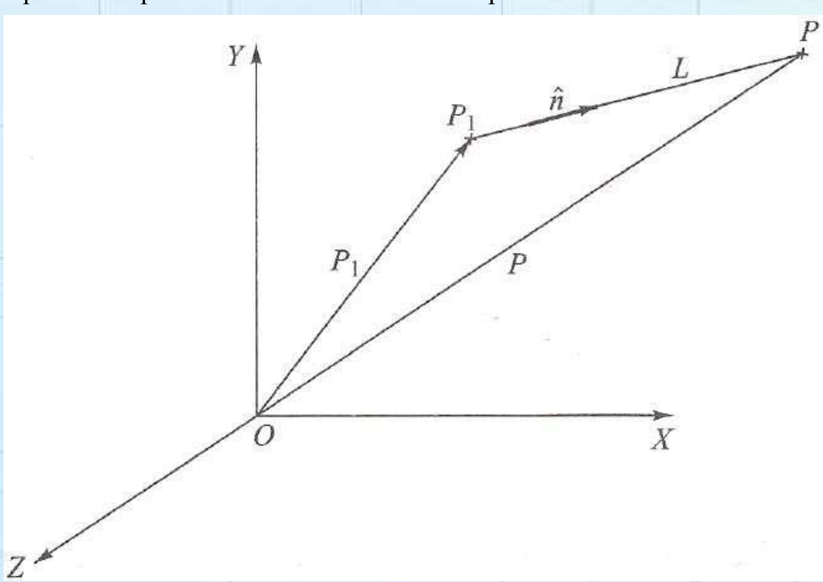

Fig.2.3 – Line passing through a point and defined by a Length and Direction

2.3.1.3 *Parallel Lines*

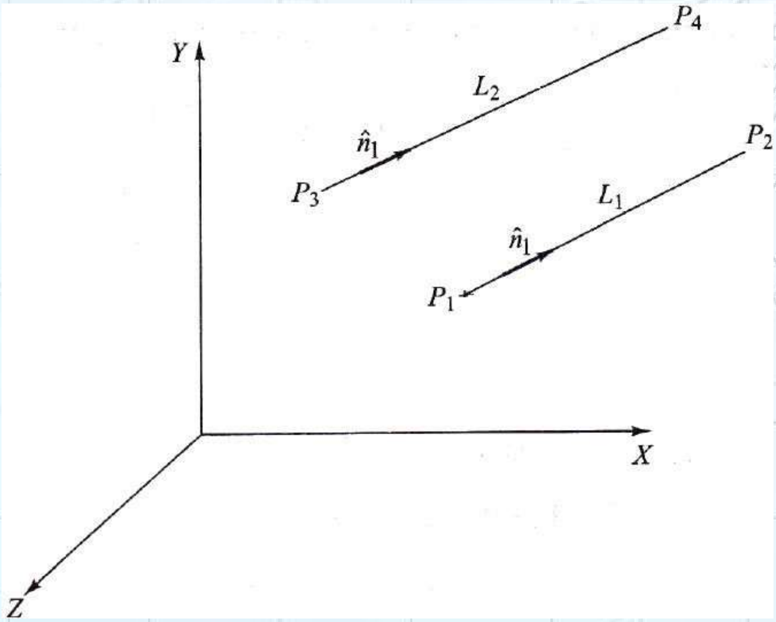

Fig.2.4

- Assume that the existing line has the two endpoints P_1 and P_2, a length L_1, and a direction defined by the unit vector $\hat{n}$. The new line has the same direction $\hat{n}$, a length L_2 and endpoints P_3 and P_4.
- The unit vector $\hat{n}$ is given by

$$\hat{n} = \frac{P_2 - P_1}{L_1}$$

- The unit vector for the parallel line will be same as unit vector for the existing line.

$$\hat{n} = \frac{P_4 - P_3}{L_2}$$

$$P_4 = P_3 + L_2(\hat{n})$$

- The equation of parallel line becomes,

$$P - P_3 = u(P_4 - P_3)$$

2.3.1.4 *Angle between two lines*

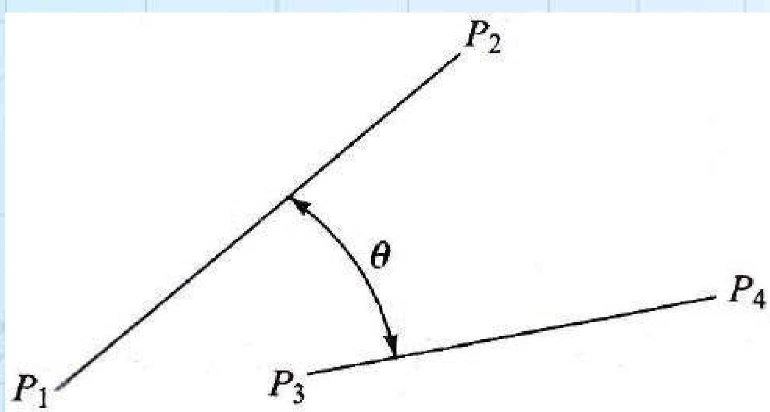

Fig.2.5

- *Fig.2.5* represents two lines L_1 and L_2 having endpoints are P_1, P_2 and P_3, P_4, the angle measurement command uses the equation.

$$\cos\theta = \frac{(P_2 - P_1).(P_4 - P_3)}{|P_2 - P_1||P_4 - P_3|}$$

$$\cos\theta = L_1.L_2$$

2.3.1.5 *Distance of a point from line*

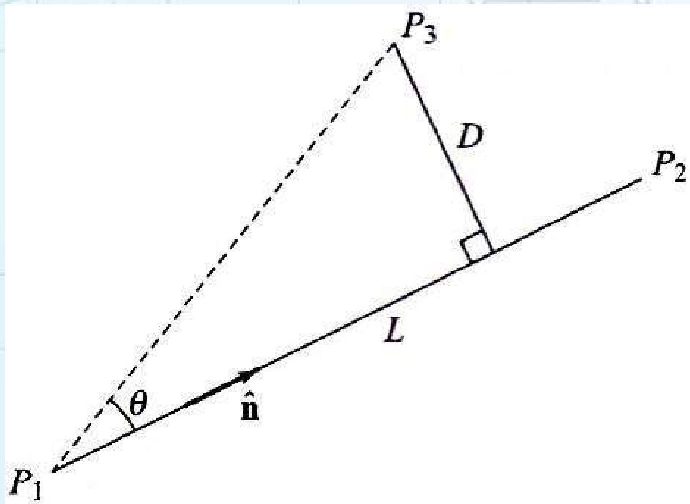

Fig.2.6

- *Fig.2.6* shows the distance D from point P_3 to the line whose endpoints are P_1 and P_2 and direction is $\hat{n}$. Its length is L. D is given by

$$D = |(P_3 - P_1) \times \hat{n}| = |(P_3 - P_1) \times \frac{(P_3 - P_1)}{|P_3 - P_1|}|$$

Ex. 2.1 ***[GTU; 7 Marks]***

For the position vectors $P_1[1\ 2]$ and $P_2[4\ 3]$, determine the parametric representation of line segment between them. Also determine the slope and tangent vector of line segment.

Solution: A parametric representation of line is

$$P = P_1 + u(P_2 - P_1)$$

$$= [1\ \ 2] + u([4\ \ 3] - [1\ \ 2])$$

$$= [\mathbf{1\ \ 2}] + \boldsymbol{u}[\mathbf{3\ \ 1}]$$

Parametric representation of x and y components are

$$x(u) = x_1 + u(x_2 - x_1)$$

$$= 1 + 3u$$

$$y(u) = y_1 + u(y_2 - y_1)$$

$$= 2 + u$$

The tangent vector is obtained by differentiating $P(u)$

$$P'(u) = [x'(u) \quad y'(u)]$$

$$= [\mathbf{3\ \ 1}]$$

The slope of line segment is

$$\frac{dy}{dx} = \frac{dy/du}{dx/du}$$

$$= \frac{y'}{x'}$$

$$= \frac{\mathbf{1}}{\mathbf{3}}$$

Ex. 2.2 ***[GTU; 7 Marks]***

The end points of line are $P_1(1,3,7)$ and $P_2(-4,5,-3)$. Determine

i. Tangent vector of the line
ii. Length of line
iii. Unit vector in the direction of line

Solution: Parametric representation of x, y and z components are

$$x(u) = x_1 + u(x_2 - x_1)$$

$$= 1 - 5u$$

$$y(u) = y_1 + u(y_2 - y_1)$$

$$= 3 + 2u$$

$$z(u) = z_1 + u(z_2 - z_1)$$

$$= 7 - 10u$$

Tangent vector,

$$P'(u) = [x'(u) \quad y'(u) \quad z'(u)]$$

$$= [\mathbf{-5} \quad \mathbf{2} \quad \mathbf{-10}]$$

$$= \mathbf{-5}\boldsymbol{i} + \mathbf{2}\boldsymbol{j} - \mathbf{10}\boldsymbol{k}$$

Length of line,

$$L = |P_2 - P_1|$$

$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$= \sqrt{(-4 - 1)^2 + (5 - 3)^2 + (3 - 7)^2}$$

$$= 11.358$$

Unit vector in the direction of line,

$$\hat{n} = \frac{P_2 - P_1}{|P_2 - P_1|} = \frac{P_2 - P_1}{L}$$

$$= \frac{1}{11.358}[-5 \quad 2 \quad -10]$$

$$= [\mathbf{-0.44} \quad \mathbf{0.176} \quad \mathbf{-0.88}]$$

$$= \mathbf{-0.44}\boldsymbol{i} + \mathbf{0.176}\boldsymbol{j} - \mathbf{0.88}\boldsymbol{k}$$

Ex. 2.3 ***[GTU; May-2017; 7 Marks]***

A line is represented by the end point $P_1(2, 4, 6)$ and $P_2(-3, 6, 9)$. If the value of parameter u at P_1 and P_2 is 0 and 1 respectively, determine the tangent vector for the line. Also determine the coordinate of a point represented by; u equal to 0, 0.25, -0.25, 1 and 1.5. Also find the length and unit vector of line between two points P_1 and P_2.

Solution: Parametric representation of x, y and z components are

$$x(u) = x_1 + u(x_2 - x_1)$$
$$= 2 - 5u$$
$$y(u) = y_1 + u(y_2 - y_1)$$
$$= 4 + 2u$$
$$z(u) = z_1 + u(z_2 - z_1)$$
$$= 6 - 3u$$

Tangent vector,

$$P'(u) = [x'(u) \quad y'(u) \quad z'(u)]$$
$$= [\mathbf{-5} \quad \mathbf{2} \quad \mathbf{-3}]$$
$$= \mathbf{-5}\boldsymbol{i} + \mathbf{2}\boldsymbol{j} - \mathbf{3}\boldsymbol{k}$$

u	**0**	**0.25**	**–0.25**	**1**	**1.5**
$x(u)$	2	0.75	3.25	–3	–5.5
$y(u)$	4	4.5	3.5	6	7
$z(u)$	6	5.25	6.75	3	1.5

Length of line,

$$L = |P_2 - P_1|$$
$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$
$$= \sqrt{(-3 - 2)^2 + (6 - 4)^2 + (9 - 6)^2}$$
$$= 6.16$$

Unit vector in the direction of line,

$$\hat{n} = \frac{P_2 - P_1}{|P_2 - P_1|} = \frac{P_2 - P_1}{L}$$
$$= \frac{1}{6.16}[-5 \quad 2 \quad -3]$$
$$= [\mathbf{-0.81} \quad \mathbf{0.324} \quad \mathbf{-0.487}]$$
$$= \mathbf{-0.81}\boldsymbol{i} + \mathbf{0.324}\boldsymbol{j} - \mathbf{0.487}\boldsymbol{k}$$

2.3.2 Circle

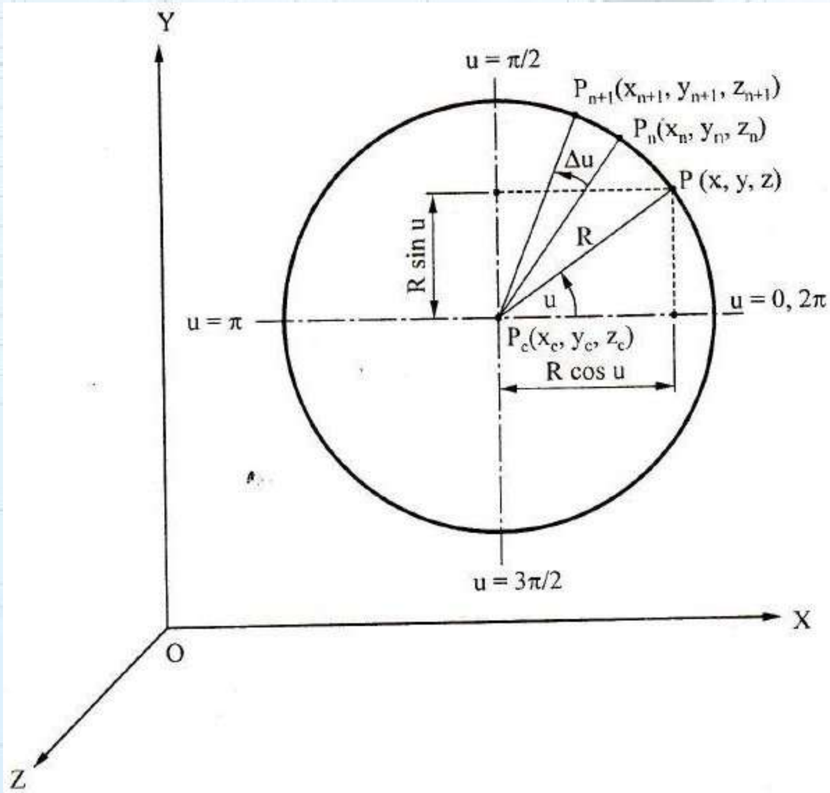

Fig.2.7 – Circle in Parametric Form

- A circle is represented in the CAD/CAM database by storing the values of its center and its radius. In a parametric form, a circle shown in *Fig.2.7*, can be represented by the following set of equations:

$$x = x_C + R\cos u$$

$$y = y_C + R\cos u$$

$$z = z_C + R\cos u$$

where, $0 \le u \le 2\pi$

- In this case the parameter u represents the angle measured from the X axis to any point P on the circumference of the circle. By incrementing the values of u from 0 to 2π, the intermediate points on the circle can be worked out.
- These points can then be connected by straight lines for the purpose of display. However, this method of computation is inefficient as it requires trigonometric evaluation of functions.
- A convenient method to determine the different points on the circle is to use the incremental equations. Consider any point P_n (x_n, y_n, z_n) on the circle.
- Let the subsequent incremental point be P_{n+1} $(x_{n+1}, y_{n+1}, z_{n+1})$. From the above equation, we get:

$$x_n = x_C + R\cos u$$

$$R\cos u = x_n - x_C \qquad \textit{Eq. (2.1)}$$

$$y_n = y_C + R\sin u$$

$$R\sin u = y_n - y_C \qquad \textit{Eq. (2.2)}$$

$$x_{n+1} = x_C + R\cos(u + \Delta u) \qquad \textit{Eq. (2.3)}$$

$$y_{n+1} = y_C + R\sin(u + \Delta u) \qquad \textit{Eq. (2.4)}$$

$$z_{n+1} = z_n \qquad \textit{Eq. (2.5)}$$

From *Eq. (2.3)*, *Eq. (2.1)* & *Eq. (2.2)*

$$x_{n+1} = x_C + R\cos u \cos \Delta u - R \sin u \sin \Delta u$$

$$x_{n+1} = x_C + (x_n - x_C)\cos \Delta u - (y_n - y_C)\sin \Delta u \qquad \textit{Eq. (2.6)}$$

From *Eq. (2.4)*, *Eq. (2.1)* & *Eq. (2.2)*

$$y_{n+1} = y_C + R\sin u \cos \Delta u - R \cos u \sin \Delta u$$

$$y_{n+1} = y_C + (y_n - y_C)\cos \Delta u - (x_n - x_C)\sin \Delta u \qquad \textit{Eq. (2.7)}$$

$$z_{n+1} = z_c$$

- Thus, the circle can start from an arbitrary point and successive points with equal spacing can be calculated. $\cos \Delta u$ & $\sin \Delta u$ have to be calculated only once, this eliminates the computations of trigonometric functions for each point.
- This algorithm is useful for hardware implementation to speed up the circle generation & display.
- If two endpoints of diameter of circle are given, then the center & radius of circle are calculated as below.
- Let $P_1(x_1, y_1, z_1)$ & $P_2(x_2, y_2, z_2)$ and center point $P_C(x_C, y_C, z_C)$

$$Radius, R = \frac{1}{2}\left(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}\right)$$

$$Center, P_C = \frac{1}{2}(P_1 + P_2)$$

$$[x_C \quad y_C \quad z_C] = \left[\frac{x_1 + x_2}{2} \quad \frac{y_1 + y_2}{2} \quad \frac{z_1 + z_2}{2}\right]$$

Ex. 2.4 ***[GTU; 3 Marks]***

The two endpoints of diameter of a circle are $P_1(13,15,7)$ and $P_2(35,40,7)$. Determine the centre and radius of circle.

Solution: The centre of a circle,

$$P_C = \frac{1}{2}(P_1 + P_2)$$

$$[x_C \quad y_C \quad z_C] = \left[\frac{x_1 + x_2}{2} \quad \frac{y_1 + y_2}{2} \quad \frac{z_1 + z_2}{2}\right]$$

$$[x_C \quad y_C \quad z_C] = \left[\frac{13 + 35}{2} \quad \frac{15 + 40}{2} \quad \frac{7 + 7}{2}\right]$$

$$[x_C \quad y_C \quad z_C] = [24 \quad 27.5 \quad 7]$$

The radius of circle

$$R = \frac{1}{2}\left(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}\right)$$

$$R = \frac{1}{2}\left(\sqrt{(35 - 13)^2 + (40 - 15)^2 + (7 - 7)^2}\right)$$

$$R = 16.65$$

2.3.3 Ellipse

- Mathematically the ellipse is a curve generated by a point moving in space such that at any position the sum of its distances from two fixed points (foci) is constant and equal to the major diameter. Each focus is located on the major axis of the ellipse at a distance from its center equal to $\sqrt{A^2 + B^2}$ (A and B are the major and minor radii). Circular holes and forms become ellipses when they are viewed obliquely relative to their planes.

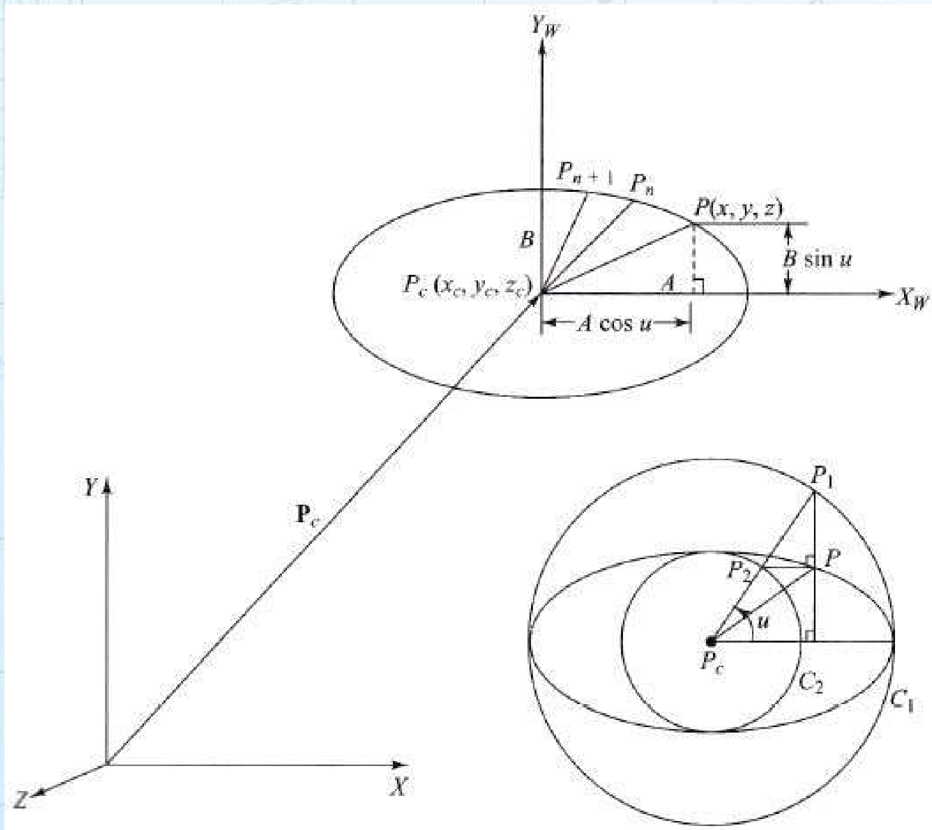

Fig.2.8 – Ellipse Defined by a Center, Major and Minor Axes

- However, four conditions (points and/or tangent vectors) are required to define the geometric shape of an ellipse.
- *Fig.2.8* shows an ellipse with point Pc as the center and the lengths of half of the major and minor axes are A and B respectively. The parametric equation of an ellipse can be written as assuming the plane of the ellipse is the XY plane.
- The parameter u is the angle as in the case of a circle. However, for a point P shown in the figure, it is not the angle between the line PPc and the major axis of the ellipse. Instead, it is defined as shown.
- To find point P on the ellipse that corresponds to an angle u, the two concentric circles C_1 and C_2 are constructed with centers at Pc and radii of A and B respectively. A radial line is constructed at the angle u to intersect both circles at points P_1 and P_2 respectively.
- If a line parallel to the minor axis is drawn from P_1 and a line parallel to the major axis is drawn from P_2, the intersection of these two lines defines the point P.

$$\left.\begin{aligned} x &= x_C + A\cos u \\ y &= y_C + B\sin u \\ z &= z_C \end{aligned}\right\} \qquad 0 \le u \le 2\pi$$

- Similar development as in the case of a circle results in the following recursive relationships which are useful for generating points on the ellipse for display purposes without excessive evaluations of trigonometric functions:

$$x_{n+1} = x_C + (x_n - x_C)\cos\Delta u - \frac{A}{B}(y_n - y_C)\sin\Delta u$$
$$y_{n+1} = y_C + (y_n - y_C)\cos\Delta u + \frac{A}{B}(x_n - x_C)\sin\Delta u \qquad \textit{Eq. (2.8)}$$
$$z_{n+1} = z_n$$

- If the ellipse major axis is inclined with an angle α relative to the x axis as shown in *Fig.2.30*, the ellipse equation becomes

$$\left.\begin{aligned} x &= x_C + A\cos u\cos\alpha - B\sin u\sin\alpha \\ y &= y_C + A\cos u\sin\alpha - B\sin u\cos\alpha \\ z &= z_C \end{aligned}\right\} \quad 0 \le u \le 2\pi \qquad \textit{Eq. (2.9)}$$

- *Eq. (2.9)* cannot be reduced to a recursive relationship similar to what is given by *Eq. (2.8)*. Instead these equations can be written as

$$x_{n+1} = x_C + A\cos(u_n + \Delta u)\cos\alpha - B\sin(u_n + \Delta u)\sin\alpha$$
$$y_{n+1} = y_C + A\cos(u_n + \Delta u)\sin\alpha + B\sin(u_n + \Delta u)\cos\alpha$$
$$z_{n+1} = z_n$$

2.3.4 Parabola

- The parabola is defined mathematically as a curve generated by a point that moves such that its distance from a fixed point (the focus P_F) is always equal to its distance to a fixed line (the directrix) as shown in *Fig.2.9*.
- The vertex P_v is the intersection point of the parabola with its axis of symmetry. It is located midway between directrix and focus. Focus lies on the axis of symmetry.
- Useful applications of the parabolic curve in engineering design include its use in parabolic sound and light reflectors, radar antennas and in bridge arches.
- Three conditions are required to define a parabola, a parabolic curve, or a parabolic arc. The default plane of a parabola is the XY plane of the current WCS at the time of construction.
- The database of a parabola usually stores the coordinates of its vertex, distances y_{HW} and y_{LW}, that define its endpoints as shown in *Fig.2.9* the distance A between the focus and the vertex (the focal distance) and the orientation angle □.
- Unlike the ellipse, the parabola is not a closed curve. Thus, the two endpoints determine the amount of the parabola to be displayed.
- Assuming the local coordinate system of the parabola as shown in *Fig.2.9*, its parametric equation can be written as

$$\left.\begin{aligned} x &= x_v + Au^2 \\ y &= y_v + 2Au \\ z &= z_v \end{aligned}\right\} \quad 0 \le u \le \infty$$

- If the range of the y coordinate is limited to y_{HW} and y_{LW} for positive and negative values respectively, the corresponding u values become

$$u_H = \frac{y_{HW}}{2A}$$
$$u_L = \frac{y_{LW}}{2A}$$

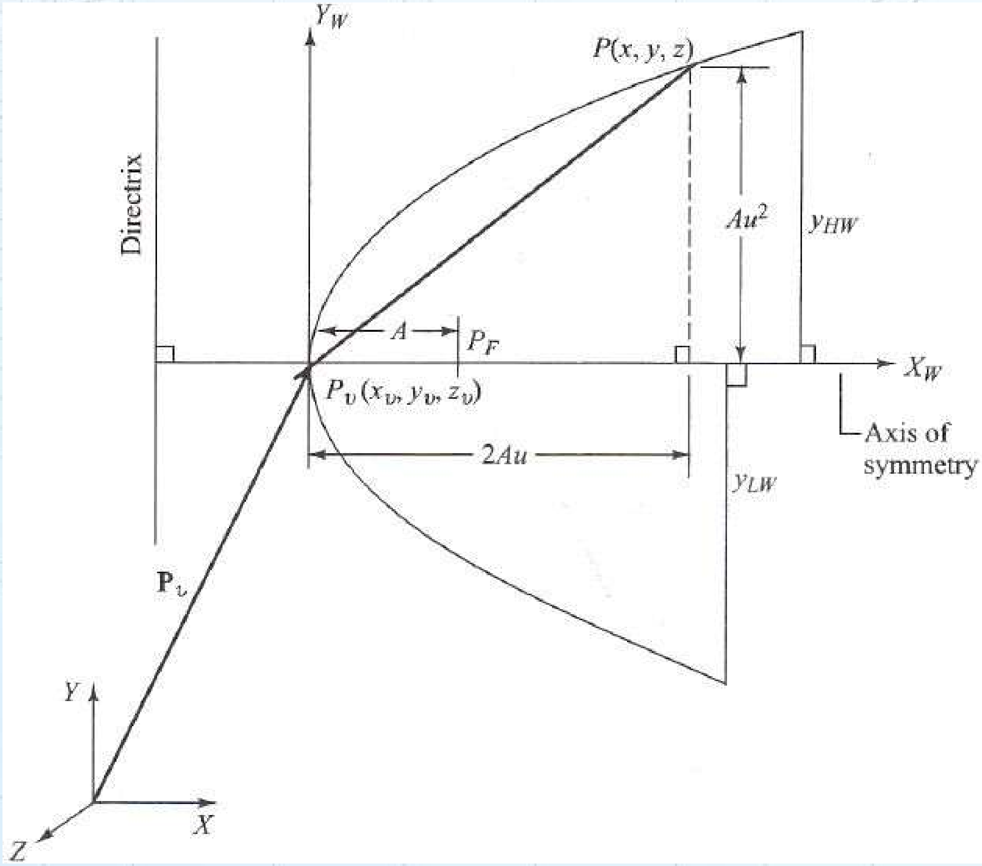

Fig.2.9 – Basic geometry of Parabola

- The recursive relationships to generate points on the parabola are obtained by substituting $u_n + \Delta u$ for points $n + 1$. This gives

$$x_{n+1} = x_n + (y_n - y_v)\Delta u + A(\Delta u)^2$$
$$y_{n+1} = y_n + 2A\Delta u$$
$$z_{n+1} = z_n$$

2.3.5 Hyperbolas

- A hyperbola is described mathematically as a curve generated by a point moving such that at any position the difference of its distances from the fixed points (foci) F and F' is a constant and equal to the transverse axis of the hyperbola.
- *Fig.2.10* shows the geometry of a hyperbola.
- The parametric equation of a hyperbola is given by

$$x = x_v + A\cosh u$$
$$y = y_v + B\sinh u$$
$$z = z_v$$

- This equation is based on the nonparametric implicit equation of the hyperbola which can be written as

$$\frac{(x - x_v)^2}{A^2} - \frac{(y - y_v)^2}{B^2} = 1$$

by utilizing the identity $\cosh^2 u - \sinh^2 u = 1$.

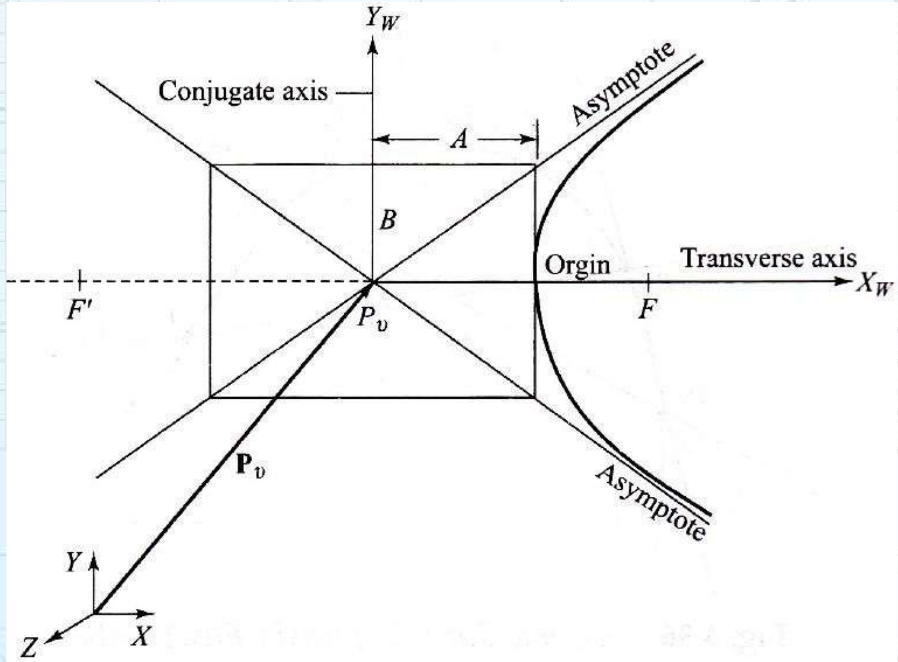

Fig.2.10 – Hyperbola Geometry

2.4 Conics

- ***Conic*** curves or conic sections form the most general form of quadratic curves. Lines, circles, ellipses, parabolas and hyperbolas are all special forms of conic curves. They all can be generated when a right circular cone of revolution is cut by planes at different angles relative to the cone axis- thus the derivation of the name conics.
- Straight lines result from intersecting a cone with a plane parallel to its axis and passing through its vertex. Circles results when the cone is sectioned by a plane perpendicular to its axis while ellipses, parabolas and hyperbolas are generated when the plane is inclined to the axis by various angles.

2.5 Parametric representation of synthetic curve

- Analytic curves, are usually not sufficient to meet geometric design requirements of mechanical parts. Products such as car bodies, ship hulls, airplane fuselage and wings, propeller blades, shoe insoles and bottles are a few examples that require free-form, or synthetic, curves and surfaces.
- The need for synthetic curves in design arises on two occasions: when a curve is represented by a collection of measured data points and when an existing curve must change to meet new design requirements.
- In the latter occasion, the designer would need a curve representation that is directly related to the data points and is flexible enough to bend, twist, or change the curve shape by changing one or more data points.
- Data points are usually called control points and the curve itself is called an interpolant if it passes through all the data points.
- Mathematically, synthetic curves represent a curve-fitting problem to construct a smooth curve that passes through given data points. Therefore, polynomials are the typical form of these curves.

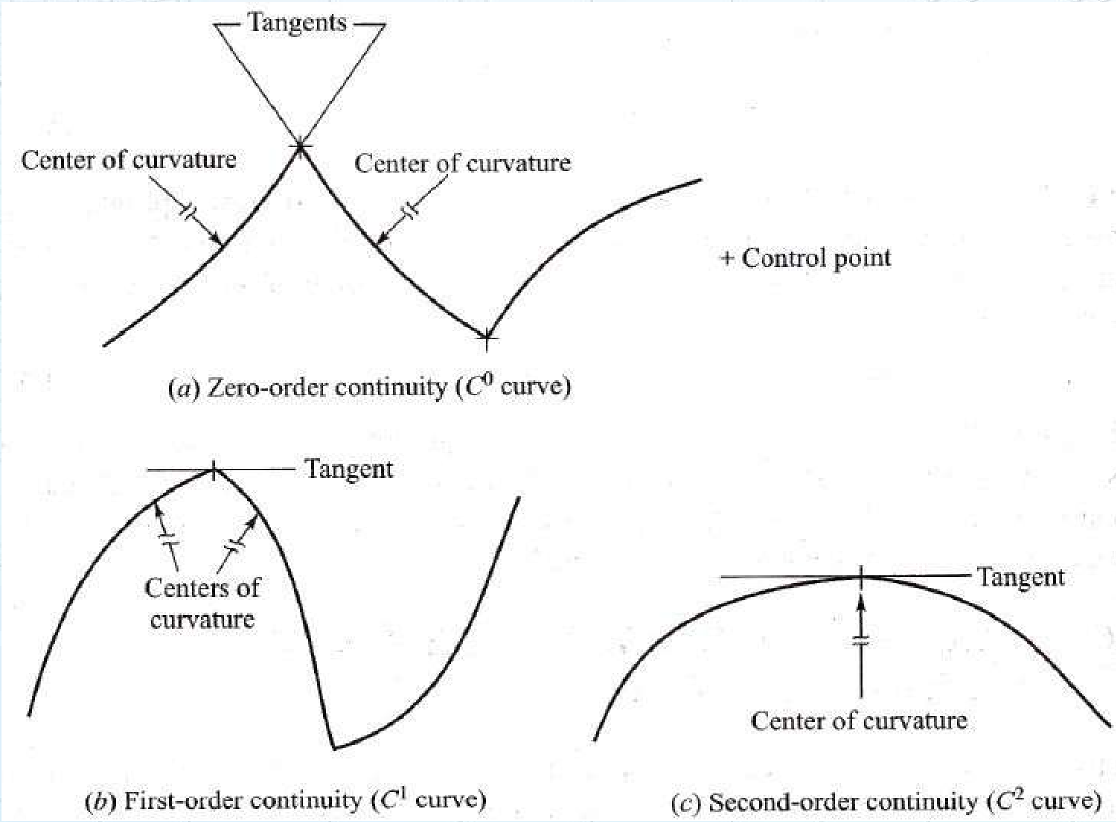

Fig.2.11 – Various Orders of Continuity of Curves

- Various continuity requirements can be specified at the data points to impose various degrees of smoothness of the resulting curve. The order of continuity becomes important when a complex curve is modeled by several curve segments pieced together end to end.
- Zero-order continuity (C^0) yields a position continuous curve. First (C^1)- and second (C^2)-order continuities imply slope and curvature continuous curves respectively.
- A C^1 curve is the minimum acceptable curve for engineering design. *Fig.2.11* shows a geometrical interpretation of these orders of continuity.
- A cubic polynomial is the minimum order polynomial that can guarantee the generation of C^0, C^1, or C^2 curves. In addition, the cubic polynomial is the lowest-degree polynomial that permits inflection within a curve segment and that allows representation of nonplanar (twisted) three dimensional curves in space.
- Higher-order polynomials are not commonly used in CAD/CAM because they tend to oscillate about control points, are computationally inconvenient and are uneconomical of storing curve and surface representations in the computer.
- The type of input data and its influence on the control of the resulting synthetic curve determine the use and effectiveness of the curve in design.
- For example, curve segments that require positions of control points and/or tangent vectors at these points are easier to deal with and gather data for than those that might require curvature information.
- Also, the designer may prefer to control the shape of the curve locally instead of globally by changing the control point(s). If changing a control point results in changing the curve locally in the vicinity of that point, local control of the curve is achieved; otherwise global control results.

- Major CAD/CAM systems provide three types of synthetic curves: Hermite cubic spline, Bezier and B-spline curves. The cubic spline curve passes through the data points and therefore is an interpolant.
- Bezier and B-spline curves in general approximate the data points, that is, they do not pass through them. Under certain conditions, the B-spline curve can be an interpolant. Both the cubic spline and Bezier curves have a first-order continuity and the B-spline curve has a second-order continuity.

2.5.1 Hermite Cubic Spline Curve

- They are used to interpolate the given data but not to design free-form curves. Splines derive their name from "French curves or splines"
- It connects two data (end) points and utilizes a cubic equation.
- Four conditions are required to determine the coefficients of the equation — two end points and the two tangent vectors at these two points.
- It passes through the control points and therefore it is an Interpolant. It has only up to C^1 continuity.

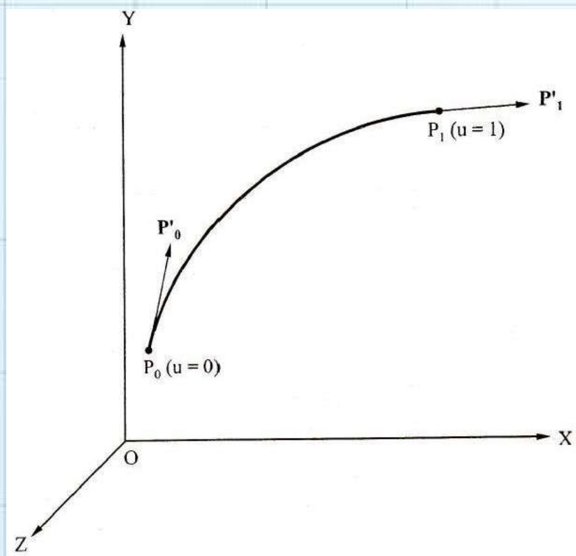

Fig.2.12 – Hermite Cubic Spline Curve

- The curve cannot be modified locally, i.e., when a data point is moved, the entire curve is affected, resulting in a global control.
- The order of the curve is always constant (cubic), regardless of the number of data points. Increase in the number of data points increases shape flexibility, however, this requires more data points, creating more splines that are joined together (only two data points and slopes are utilized for each spline).
- ***A cubic spline curve utilizes a cubic equation of the following form:***

$$P(u) = C_3u^3 + C_2u^2 + C_1u + C_0 \qquad \textit{Eq. (2.10)}$$

$$where, \quad 0 \le u \le 1$$

- The cubic spline is defined by the two endpoints P_0, P_1 and the tangent vectors at these points. The tangent vector can be found by differentiating *Eq. (2.10)*.

$$P'(u) = 3C_3u^2 + 2C_2u + C_1 \qquad \textit{Eq. (2.11)}$$

- To determine the coefficients of the curve, consider the two endpoints P_0, P_1 and the tangent vectors as shown in *Fig.2.12*.

- Applying conditions at $u = 0$, in *Eq. (2.10)* and *Eq. (2.11)*.

$$P_0 = C_0 \qquad \textit{Eq. (2.12)}$$

$$P'_0 = C_1 \qquad \textit{Eq. (2.13)}$$

- Applying conditions at $u = 1$, in *Eq. (2.10)* and *Eq. (2.11)*.

$$P_1 = C_3 + C_2 + C_1 + C_0 \qquad \textit{Eq. (2.14)}$$

$$P'_1 = 3C_3 + 2C_2 + C_1 \qquad \textit{Eq. (2.15)}$$

- Substituting *Eq. (2.12)* and *Eq. (2.13)* in *Eq. (2.14)* and *Eq. (2.15)*, we get:

$$P_1 = C_3 + C_2 + P'_0 + P_0 \qquad \textit{Eq. (2.16)}$$

$$P'_1 = 3C_3 + 2C_2 + P'_0 \qquad \textit{Eq. (2.17)}$$

- By solving simultaneous *Eq. (2.16)* and *Eq. (2.17)*, we get

$$3P_1 - P'_1 = C_2 + 2P'_0 + 3P_0$$

$$C_2 = 3P_1 - P'_1 - 2P'_0 - 3P_0 \qquad \textit{Eq. (2.18)}$$

- Similarly again by solving simultaneous *Eq. (2.16)* and *Eq. (2.17)*, we get

$$P'_1 - 2P_1 = C_3 - P'_0 - 2P_0$$

$$C_3 = P'_1 - 2P_1 + P'_0 + 2P_0 \qquad \textit{Eq. (2.19)}$$

- Substituting *Eq. (2.12)*, *Eq. (2.13)*, *Eq. (2.18)* and *Eq. (2.19)* in *Eq. (2.10)*, we get:

$$P(u) = (P'_1 - 2P_1 + P'_0 + 2P_0)u^3 + (3P_1 - P'_1 - 2P'_0 - 3P_0)u^2 + P'_0\, u + P_0$$

$$P(u) = P'_1\, u^3 - 2P_1\, u^3 + P'_0\, u^3 + 2P_0 u^3 + 3P_1\, u^2 - P'_1\, u^2 - 2P'_0\, u^2 - 3P_0\, u^2 + P'_0\, u + P_0$$

$$P(u) = 2P_0 u^3 - 3P_0\, u^2 + P_0 - 2P_1\, u^3 + 3P_1\, u^2 + P'_0\, u^3 - 2P'_0\, u^2 + P'_0\, u + P'_1\, u^3 - P'_1\, u^2$$

$$P(u) = P_0(2u^3 - 3u^2 + 1) + P_1(-2u^3 + 3u^2) + P'_0(u^3 - 2u^2 + u) + P'_1\,(u^3 - u^2) \qquad \textit{Eq. (2.20)}$$

- The expression can be written in matrix form as under:

$$P(u) = [P_0 \quad P_1 \quad P'_0 \quad P'_1] \begin{bmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{bmatrix}$$

$$P(u) = [P_0 \quad P_1 \quad P'_0 \quad P'_1] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

- On similar lines, the tangent vector equation can be determined by differentiating *Eq. (2.20)*.

$$P'(u) = P_0(6u^2 - 6u) + P_1(-6u^2 + 6u) + P'_0(3u^2 - 4u + 1) + P'_1\,(3u^2 - 2u)$$

$$P'(u) = [P_0 \quad P_1 \quad P'_0 \quad P'_1] \begin{bmatrix} 6u^2 - 6u \\ -6u^2 + 6u \\ 3u^2 - 4u + 1 \\ 3u^2 - 2u \end{bmatrix}$$

$$P'(u) = [P_0 \quad P_1 \quad P'_0 \quad P'_1] \begin{bmatrix} 0 & 6 & -6 & 0 \\ 0 & -6 & 6 & 0 \\ 0 & 3 & -4 & 1 \\ 0 & 3 & -2 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

- Changing the values of endpoints or tangent vectors would modify the shape of the curve.

Ex. 2.5 ***[GTU; 7 Marks]***

A cubic spline curve has start point $P_0(16,0)$ and end point $P_1(3,1)$. The tangent vector for end point P_0 is given by line joining P_0 and point $P_2(14,8)$. Tangent vector for end P_1 is given by line joining P_2 and point P_1.

(a). Determine the parametric equation of hermite cubic curve.

(b). Plot the hermite cubic curve.

Solution: $P_0 = [16 \;\; 0]$

$P_1 = [3 \;\; 1]$

$P'_0 = P_2 - P_0 = [14 \;\; 8] - [16 \;\; 0]$

$= [-2 \;\; 8]$

$P'_1 = P_1 - P_2 = [3 \;\; 1] - [14 \;\; 8]$

$= [-11 \;\; -7]$

Parametric equation

For any point on curve

$$P(u) = [P_0 \quad P_1 \quad P'_0 \quad P'_1]\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

X coordinate of a point on curve

$$P_x(u) = [P_{0x} \quad P_{1x} \quad P'_{0x} \quad P'_{1x}]\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_x(u) = [16 \quad 3 \quad -2 \quad -11]\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_x(u) = [13 \quad -24 \quad -2 \quad 16]\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_x(u) = 13u^3 - 24u^2 - 2u + 16$$

Y coordinate of a point on curve

$$P_y(u) = [P_{0y} \quad P_{1y} \quad P'_{0y} \quad P'_{1y}]\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_y(u) = [0 \quad 1 \quad 8 \quad -7]\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_y(u) = [-1 \quad -6 \quad 8 \quad 0]\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_y(u) = -u^3 - 6u^2 + 8u$$

Thus, parametric equation of parametric curve is

$$P_x(u) = 13u^3 - 24u^2 - 2u + 16 \qquad \textit{Eq. (2.21)}$$

$$P_y(u) = -u^3 - 6u^2 + 8u$$

Curve plot

The point on curve obtained by varying value of u from 0 to 1 in steps of 0.1 in *Eq. (2.21)* are shown in below table

Point No.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
U	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
x (u)	16	15.57	14.74	13.59	12.19	10.63	8.97	7.30	5.70	4.24	3.00
y (u)	0	0.74	1.35	1.83	2.18	2.38	2.42	2.32	2.05	1.61	1.00

The Hermite cubic curve plotted using the above coordinates is shown in *Fig.2.13*.

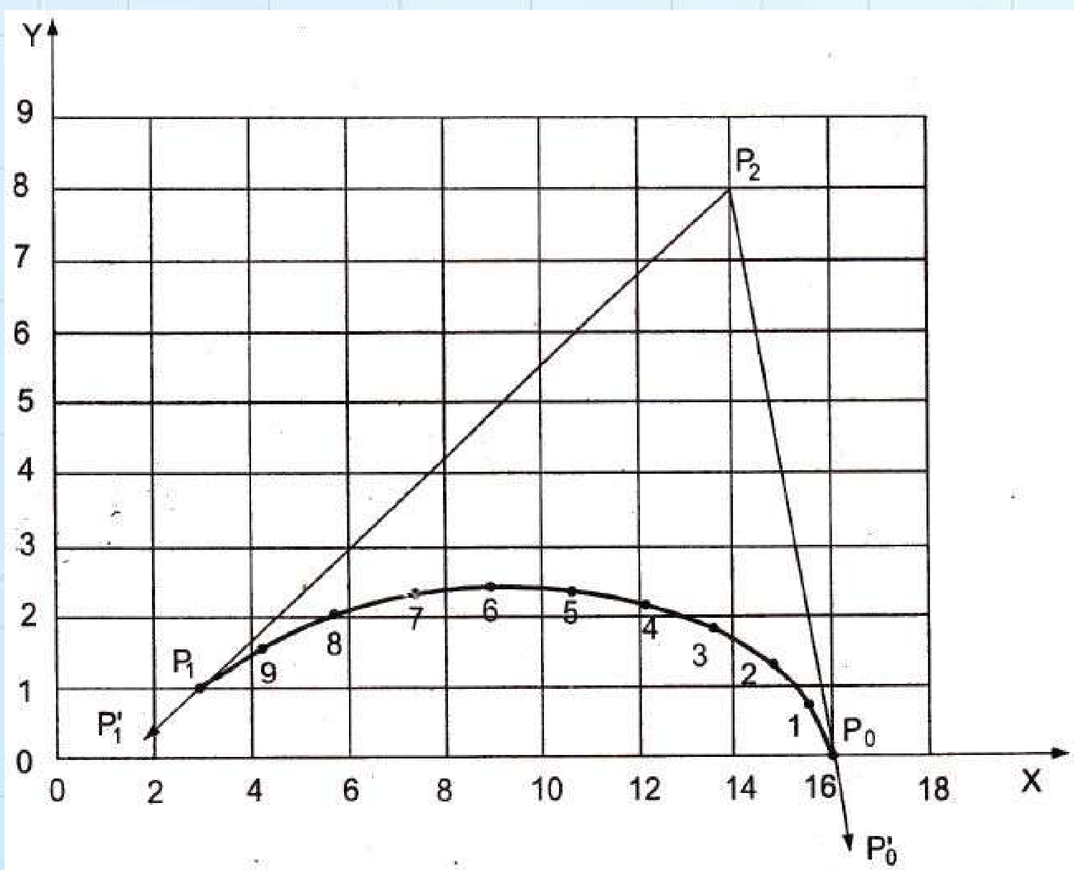

Fig.2.13

Ex. 2.6 ***[GTU; 7 Marks]***

The end point of a cubic spline curve are $P_0(1,2)$ and $P_1(7,1)$. The tangent vector for end P_0 is given by line joining P_0 and point $P_2(-2,1)$. The tangent vector for end P_0 is given by line joining $P_3(9,-2)$ and point P_1.

(a). Determine the parametric equation of hermite cubic curve.

(b). Determine the parametric equation for tangent vector.

(c). Plot the hermite cubic curve.

Solution:

$P_0 = [1 \ \ 2]$

$P_1 = [7 \ \ 1]$

$P'_0 = P_2 - P_0 = [-2 \ \ 1] - [1 \ \ 2] = [-3 \ \ -1]$

$P'_1 = P_1 - P_3 = [7 \ \ 1] - [9 \ \ -2] = [-2 \ \ 3]$

Parametric equation of curve

For any point on curve

$$P(u) = [P_0 \quad P_1 \quad P'_0 \quad P'_1] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

X coordinate of a point on curve

$$P_x(u) = [P_{0x} \quad P_{1x} \quad P'_{0x} \quad P'_{1x}] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_x(u) = [1 \quad 7 \quad -3 \quad -2] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_x(u) = [-17 \quad 26 \quad -3 \quad 1] \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_x(u) = -17u^3 + 26u^2 - 3u + 1$$

Y coordinate of a point on curve

$$P_y(u) = [P_{0y} \quad P_{1y} \quad P'_{0y} \quad P'_{1y}] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_y(u) = [2 \quad 1 \quad -1 \quad 3] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_y(u) = [4 \quad -4 \quad -1 \quad 2] \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P_y(u) = 4u^3 - 4u^2 - u + 2$$

Thus, parametric equation of parametric curve is

$$P_x(u) = -17u^3 + 26u^2 - 3u + 1 \qquad \textit{Eq. (2.22)}$$
$$P_y(u) = 4u^3 - 4u^2 - u + 2$$

Parametric equation of tangent vector

$$P'(u) = [P_0 \quad P_1 \quad P'_0 \quad P'_1] \begin{bmatrix} 0 & 6 & -6 & 0 \\ 0 & -6 & 6 & 0 \\ 0 & 3 & -4 & 1 \\ 0 & 3 & -2 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

X coordinate of tangent vector

$$P'_x(u) = [1 \quad 7 \quad -3 \quad -2] \begin{bmatrix} 0 & 6 & -6 & 0 \\ 0 & -6 & 6 & 0 \\ 0 & 3 & -4 & 1 \\ 0 & 3 & -2 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P'_x(u) = [0 \quad 51 \quad 52 \quad -3] \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P'_x(u) = 51u^2 + 52u - 3$$

Y coordinate of tangent vector

$$P'_y(u) = [2 \quad 1 \quad -1 \quad 3] \begin{bmatrix} 0 & 6 & -6 & 0 \\ 0 & -6 & 6 & 0 \\ 0 & 3 & -4 & 1 \\ 0 & 3 & -2 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P'_y(u) = [0 \quad 12 \quad -8 \quad -1] \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P'_y(u) = 12u^2 - 8u - 1$$

Thus, parametric equation of parametric curve is

$$P'_x(u) = 51u^2 + 52u - 3$$
$$P'_y(u) = 12u^2 - 8u - 1$$

Curve plot

The point on curve obtained by varying the value of u from 0 to 1 in steps of 0.1 in *Eq. (2.22)* are shown in below table

Point No.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
U	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
x (u)	1	0.94	1.30	1.98	2.87	3.88	4.89	5.81	6.54	6.97	7.00
y (u)	2	1.86	1.67	1.45	1.22	1.00	0.82	0.71	0.69	0.78	1.00

The Hermite cubic curve plotted using the above coordinates is shown in *Fig.2.14*.

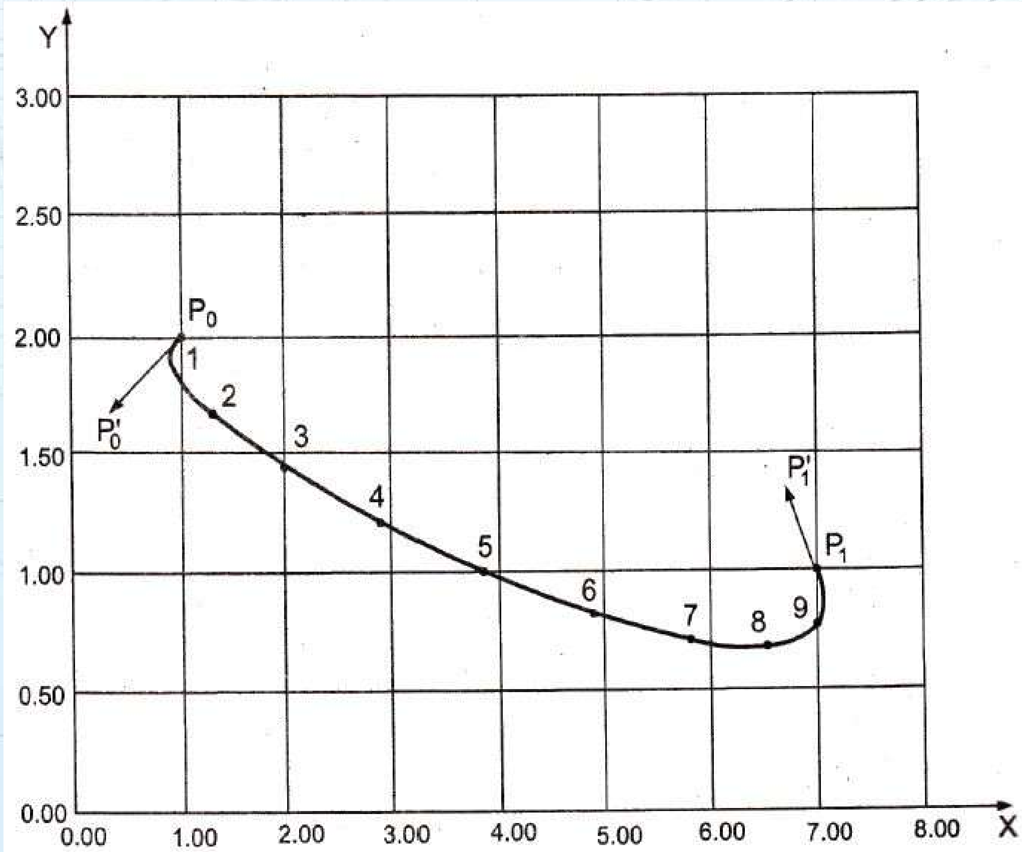

Fig.2.14

2.5.2 Bezier Curve

- Cubic splines are based on interpolation techniques. Curves resulting from these techniques pass through the given points.
- Another alternative to create curves is to use approximation techniques which produce curves that do not pass through the given data points. Instead, these points are used to control the shape of the resulting curves.
- Most often, approximation techniques are preferred over interpolation techniques in curve design due to the added flexibility and the additional intuitive feel provided by the former. Bezier and B- spline curves are examples based on approximation techniques.
- As its mathematics show shortly, the major differences between the Bezier curve and the cubic spline curve are:

1. The shape of Bezier curve is controlled by its defining points only. First derivatives are not used in the curve development as in the case of the cubic spline. This allows the designer a much better feel for the relationship between input (points) and output (curve).
2. The order or the degree of Bezier curve is variable and is related to the number of points defining it; n + 1 points define an n^{th} degree curve which permits higher-order continuity. This is not the case for cubic splines where the degree is always cubic for a spline segment.
3. The Bezier curve is smoother than the cubic spline because it has higher order derivatives.

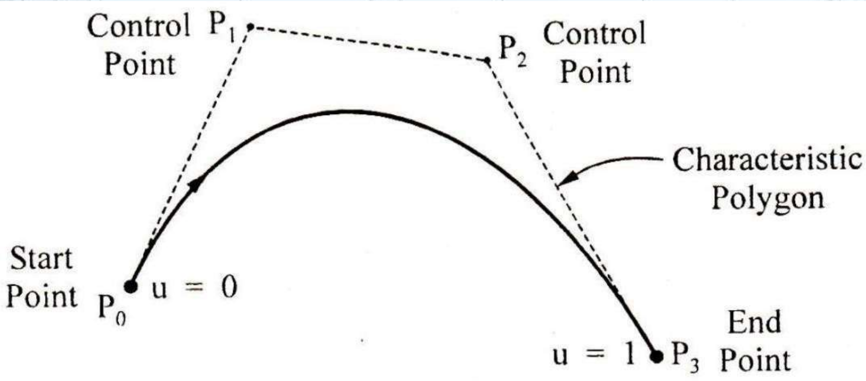

Fig.2.15 – Bezier curve

- The Bezier curve is defined in terms of (n+1) points. This points are called as control points, where n is the degree of the curve.
- If n=3 control points are 4, as shown in the *Fig.2.15*.
- These control points form the vertices of the control polygon or Bezier characteristic polygon.
- The control or Bezier characteristic polygon uniquely defines the curve.

Properties of Beizer Curve:

- The degree of polynomial defining the curve segment is one less than the no. of defining polygon points (n-1).
- The curve follows the shape of the defining polygon.
- Only the first and last control points or vertices of polygon actually lie on curve.
- The other vertices define order & shape of the curve. (see *Fig.2.16*)

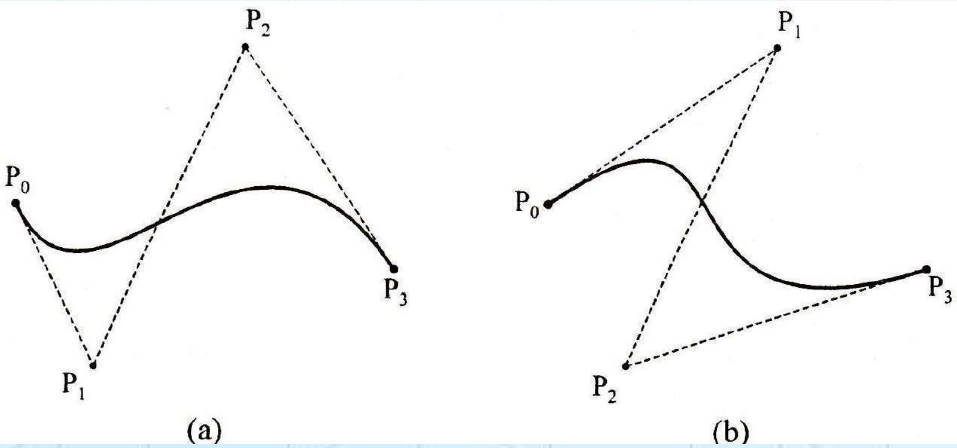

Fig.2.16 – Same data points but different orders for Bezier Curve

- The curve is also always tangent to first and last segments of polygon.
- The same Beizer curve would be generated, if the sequence of the points is changed from $P_0 - P_1 - P_2 - P_3$ to $P_3 - P_2 - P_1 - P_0$.
- The flexibility of the shape would increase with increase in number of vertices of the polygon.
- It is having global control on the shape of the curve.

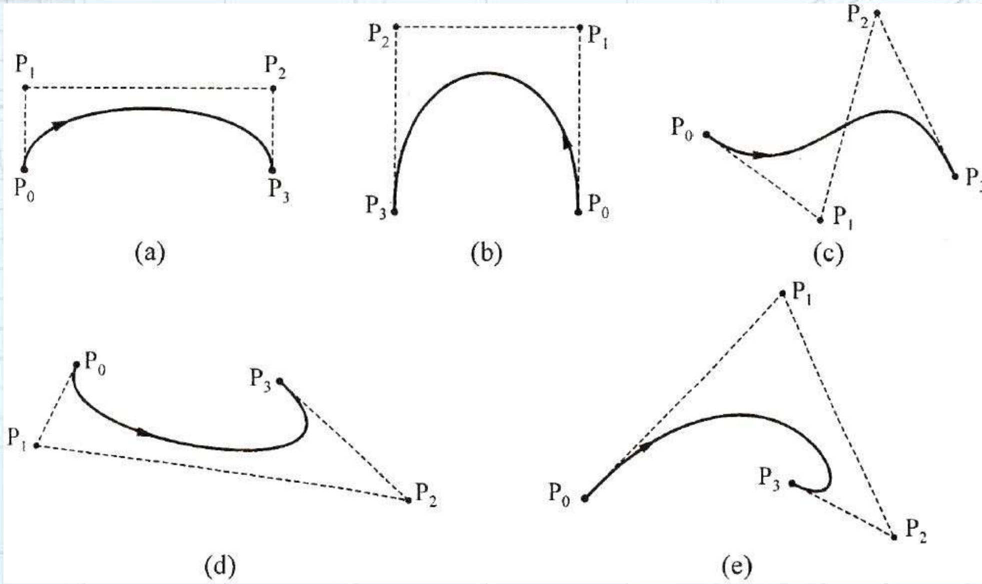

Fig.2.17 – Some Typical Examples of Bezier Curve

- Mathematically, for n + 1 control points, the Bezier curve is defined by the following polynomial of degree n:

$$P(u) = \sum_{i=0}^{n} P_i \, B_{i,n}(u), \qquad 0 \le u \le 1 \qquad \textit{Eq. (2.23)}$$

- where $P(u)$ is any point on the curve and P_i is a control point. $B_{i,n}$ are the Bernstein polynomials. Thus, the Bezier curve has a Bernstein basis. The Bernstein polynomial serves as the blending or basis function for the Bezier curve and is given by

$$B_{i,n} = C(n, i)u^i(1-u)^{n-i} \qquad \textit{Eq. (2.24)}$$

where $C(n, i)$ is the binomial coefficient

$$C(n, i) = \frac{n!}{i!\,(n-i)!} \qquad \textit{Eq. (2.25)}$$

- Utilizing *Eq. (2.24)* and *Eq. (2.25)* and observing that $C(n, 0) = C(n, n) = 1$,
- *Eq. (2.23)* can be expanded to give

$$P(u) = P_0(1-u)^n + P_1C(n, 1)u(1-u)^{n-1} + P_2C(n, 2)u^2(1-u)^{n-2}$$
$$+ \cdots + P_2C(n, n-1)u^{n-1}(1-u) + P_nu^n \qquad 0 \le u \le 1$$

- Bezier curve with 3 control points

$$P(u) = P_0(1-u)^2 + P_1C(2,1)u(1-u)^{2-1} + P_2C(2,2)u^2(1-u)^{2-2}$$
$$P(u) = (1-u)^2P_0 + 2u(1-u)P_1 + u^2P_2$$

- Bezier curve with 4 vertices

$$P(u) = P_0(1-u)^3 + P_1C(3,1)u(1-u)^{3-1} + P_2C(3,2)u^2(1-u)^{3-2} + P_3C(3,3)u^3(1-u)^{3-3}$$
$$P(u) = (1-u)^3P_0 + 3u(1-u)^2P_1 + 3u^2(1-u)P_2 + u^3P_3$$

- Bezier curve with 5 vertices

$$P(u) = P_0(1-u)^4 + P_1C(4,1)u(1-u)^{4-1} + P_2C(4,2)u^2(1-u)^{4-2} + P_3C(4,3)u^3(1-u)^{4-3}$$
$$+ P_4C(4,4)u^4(1-u)^{4-4}$$
$$P(u) = (1-u)^4P_0 + 4u(1-u)^3P_1 + 6u^2(1-u)^2P_2 + 4u^3(1-u)P_3 + u^4P_4$$

Ex. 2.7 ***[GTU; May-2016; 7 Marks]***

A Bezier curve is to be constructed using control points $P_0(35,30)$, $P_1(25,0)$, $P_2(15,25)$ and $P_3(5,10)$. The Bezier curve is anchored at P_0 and P_3. Find the equation of the Bezier curve and plot the curve for u = 0, 0.2, 0.4, 0.6, 0.8 and 1.

Solution: $P_0 = [35 \quad 30]$

$P_1 = [25 \quad 0]$

$P_2 = [15 \quad 25]$

$P_3 = [5 \quad 10]$

n = 3

- **The parametric equation of Bezier curve**

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

X-coordinate of a point on curve

$$P_x(u) = (1-u)^3P_{0x} + 3u(1-u)^2P_{1x} + 3u^2(1-u)P_{2x} + u^3P_{3x}$$

$$P_x(u) = 35(1-u)^3 + 75u(1-u)^2 + 45u^2(1-u) + 5u^3$$

$$P_x(u) = 35(1-3u+3u^2-u^3) + 75u(1-2u+u^2) + 45u^2 - 45u^3 + 5u^3$$

$$P_x(u) = 35 - 105u + 105u^2 - 35u^3 + 75u - 105u^2 + 75u^3 + 45u^2 - 45u^3 + 5u^3$$

$$P_x(u) = 35 - 30u$$

Y-coordinate of a point on curve

$$P_y(u) = (1-u)^3P_{0y} + 3u(1-u)^2P_{1y} + 3u^2(1-u)P_{2y} + u^3P_{3y}$$

$$P_y(u) = 30(1-u)^3 + 75u^2(1-u) + 10u^3$$

$$P_y(u) = 30(1-3u+3u^2-u^3) + 75u^2 - 75u^3 + 10u^3$$

$$P_y(u) = 30 - 90u + 165u^2 - 95u^3$$

The parametric equation of Bezier curve

Eq. (2.26)

$$P_x(u) = 35 - 30u$$

$$P_y(u) = 30 - 90u + 165u^2 - 95u^3$$

The points on curve obtained by varying the value of $u = 0, 0.2, 0.4, 0.6, 0.8$ and 1 in *Eq. (2.26)* are shown in below table

Point No.	**0**	**2**	**4**	**6**	**8**	**10**
u	0	0.2	0.4	0.6	0.8	1
Px (u)	35.0	29.0	23.0	17.0	11.0	5.00
Py (u)	30.0	17.84	14.32	14.88	14.96	10.00

The Bezier curve plotted using the above coordinates is shown in *Fig.2.18*.

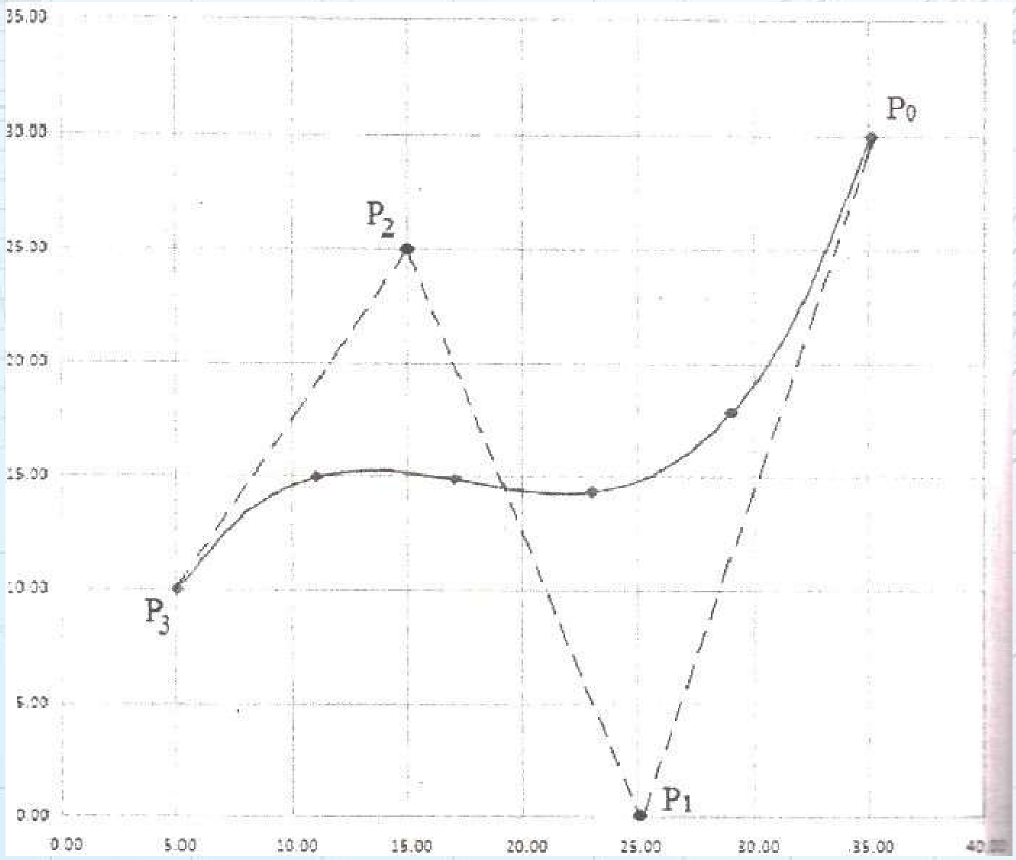

Fig.2.18

Solution: $P_0 = [20]$ $P_1 = [4\ \ 3]$

$P_2 = [52]$ $P_3 = [4\ \ -2]$

$P_4 = [5-3]$ $P_5 = [6\ \ -2]$

$n = 5$

The parametric equation for Bezier curve with 6 control points

$$P(u) = (1-u)^5P_0 + 5u(1-u)^4P_1 + 10u^2(1-u)^3P_2 + 10u^3(1-u)^2P_3 + 5u^4(1-u)P_4 + u^5P_5$$

$$P(u) = (1 - 5u + 10u^2 - 10u^3 + 5u^4 - u^5)P_0 + 5u(1 - 4u + 6u^2 - 4u^3 + u^4)P_1 + 10u^2(1 - 3u + 3u^2 - u^3)P_2 + 10u^3(1 - 2u + u^2)P_3 + 5u^4(1-u)P_4 + u^5P_5$$

$$P(u) = (1 - 5u + 10u^2 - 10u^3 + 5u^4 - u^5)P_0 + (5u - 20u^2 + 30u^3 - 20u^4 + 5u^5)P_1 + (10u^2 - 30u^3 + 30u^4 - 10u^5)P_2 + (10u^3 - 20u^4 + 10u^5)P_3 + (5u^4 - 5u^5)P_4 + u^5P_5$$

$$P(u) = P_0 + (-5P_0 + 5P_1)u + (10P_0 - 20P_1 + 10P_2)u^2 + (-10P_0 + 30P_1 - 30P_2 + 10P_3)u^3 + (5P_0 - 20P_1 + 30P_2 - 20P_3 + 5P_4)u^4 + (-P_0 + 5P_1 - 10P_2 + 10P_3 - 5P_4 + P_5)u^5$$

X-coordinate of a point on a curve

$$P_x(u) = P_{0x} + (-5P_{0x} + 5P_{1x})u + (10P_{0x} - 20P_{1x} + 10P_{2x})u^2 + (-10P_{0x} + 30P_{1x} - 30P_{2x} + 10P_{3x})u^3 + (5P_{0x} - 20P_{1x} + 30P_{2x} - 20P_{3x} + 5P_{4x})u^4 + (-P_{0x} + 5P_{1x} - 10P_{2x} + 10P_{3x} - 5P_{4x} + P_{5x})u^5$$

$$P_x(u) = 2 + (-5(2) + 5(4)u + (10(2) - 20(4) + 10(5))u^2 + (-10(2) + 30(4) - 30(5) + 10(4))u^3 + (5(2) - 20(4) + 30(5) - 20(4) + 5(5))u^4 + (-(2) + 5(4) - 10(5) + 10(4) - 5(5) + (6))u^5$$

$$P_x(u) = 2 + 10u - 10u^2 - 10u^3 \quad + 25u^4 - 11u^5$$

Y-coordinate of a point on a curve

$$P_y(u) = P_{0y} + (-5P_{0y} + 5P_{1y})u + (10P_{0y} - 20P_{1y} + 10P_{2y})u^2 + (-10P_{0y} + 30P_{1y} - 30P_{2y} + 10P_{3y})u^3 + (5P_{0y} - 20P_{1y} + 30P_{2y} - 20P_{3y} + 5P_{4y})u^4 + (-P_{0y} + 5P_{1y} - 10P_{2y} + 10P_{3y} - 5P_{4y} + P_{5y})u^5$$

$$P_y(u) = 0 + (-5(0) + 5(3))u + (10(0) - 20(3) + 10(2))u^2 + (-10(0) + 30(3) - 30(2) + 10(-2))u^3 + (5(0) - 20(3) + 30(2) - 20(-2) + 5(-3))u^4 + (-(0) + 5(3) - 10(2) + 10(2) - 5(-3) + (-2))u^5$$

$$P_y(u) = 15u - 40u^2 + 10u^3 \quad + 25u^4 - 12u^5$$

The parametric equation of Bezier curve

$$P_x(u) = 2 + 10u - 10u^2 - 10u^3 \quad + 25u^4 - 11u^5$$
$$P_y(u) = 15u - 40u^2 + 10u^3 \quad + 25u^4 - 12u^5 \qquad \textit{Eq. (2.27)}$$

The points on curve obtained by varying the value of u from 0 to 1 in steps of 0.1 in *Eq. (2.27)* are shown in below table

Point No.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
u	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
P_x (u)	2	2.89	3.56	4.01	4.29	4.47	4.62	4.82	5.12	5.52	6.00
P_y (u)	0	1.11	1.52	1.34	0.76	-0.06	-0.93	-1.68	-2.17	-2.29	-2.00

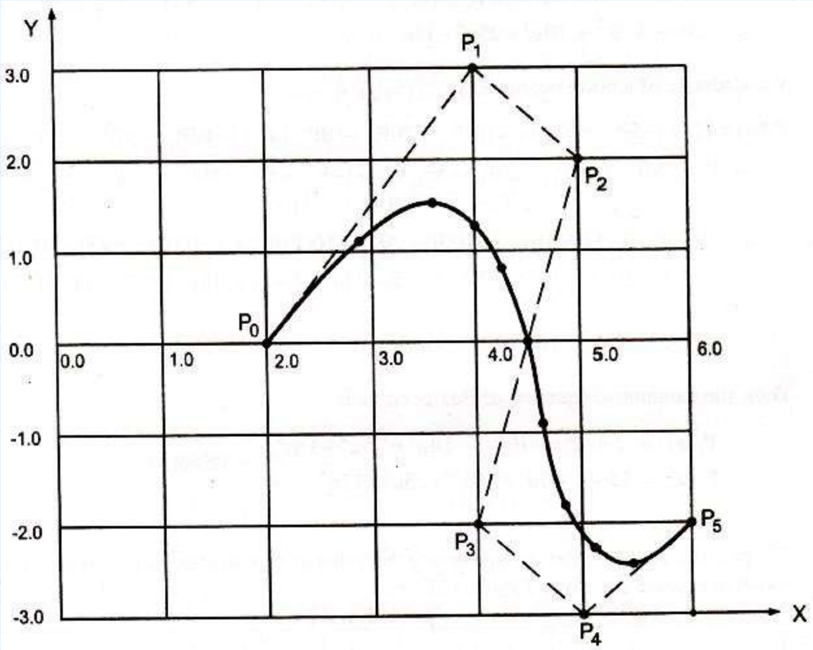

Fig.2.19

2.5.3 B-Spline Curve

- B-Spline curves are proper and powerful generalization of Bezier curve.
- They are very similar to Bezier curve and are also defined with the help of characteristic polygon.
- The major advantage of B-Spline curve is due to its ability to control the curve shape locally, as opposed to global control. (see *Fig.2.20*).

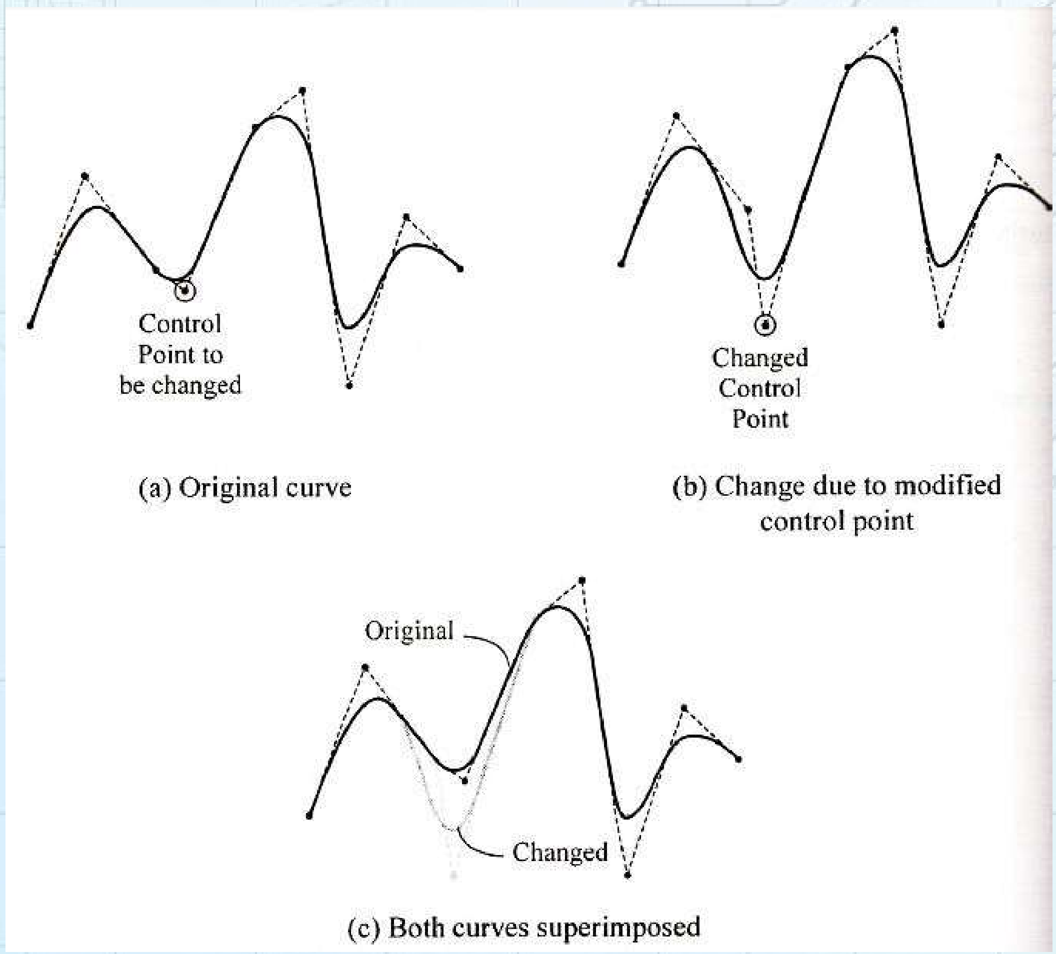

Fig.2.20 – Local Control of a B-Spline Curve

- B-Spline curves allow a local control over the shape of the spline curve.
- B-Spline curves allow us to vary the control points without changing the degree of the polynomial (i.e. number of control points can be added or subtracted).
- The degree of the curve is independent of the number of control points (i.e. four control points can generate a linear, quadratic or cubic curve).
- It gives better control over the shape of curve.
- They are more complex compared to Bezier curves.
- B-spline curve defined by n + 1 control points P_i is given by

$$P(u) = \sum_{i=0}^{n} P_i \, N_{i,k}(u), \qquad 0 \le u \le u_{max} \qquad \text{Eq. (2.28)}$$

$N_{i,k}(u)$ are the B-spline functions. Thus, the B-spline curves have a B-spline basis.

2.5.4 Rational curves

- A rational curve is defined by the algebraic ratio of two polynomials while a nonrational curve is defined by one polynomial.

- The formulation of rational curves requires the introduction of homogeneous space and the homogeneous coordinates. The homogeneous space is four-dimensional space. A point in three dimensional with coordinates (x, y, z) is represented in the homogeneous space by the coordinates (x*, y*, z*, h),where h is a scalar factor. The relationship between the two types of coordinates is

$$x = \frac{x^*}{h} \qquad y = \frac{y^*}{h} \qquad z = \frac{z^*}{h}$$

- A rational B-spline curve defined by n + 1 control points P_i is given by

$$P(u) = \sum_{i=0}^{n} P_i \, R_{i,k}(u), \qquad 0 \le u \le u_{max} \qquad \textit{Eq. (2.29)}$$

- $R_{i,k}(u)$ are the rational B-spline functions and are given by

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

  - The above equation shows that $R_{i,k}(u)$ are a generalization of the nonrational basis functions $N_{i,k}(u)$. If we substitute $h_i = 1$ in the equation, $R_{i,k}(u) = N_{i,k}(u)$. The rational basis functions $R_{i,k}(u)$ have nearly all the analytic and geometric characteristics of their nonrational B-spline counterparts.

- The main difference between rational and nonrational B-spline curves is the ability to use hi at each control point to control the behavior of the rational B-splines (or rational curves in general).

- Thus, similarly to the knot vector, one can define a homogeneous coordinate vector $H = [h_0\ h_1\ h_2\ h_3\ \ldots\ h_n]^T$ at the control points $P_0, P_1, \ldots, P_n$ of the rational B-spline curve. The choice of the H vector controls the behavior of the curve.

- A rational B-spline is considered a unified representation that can define a variety of curves and surfaces. The premise is that it can represent all wireframe, surface and solid entities. This allows unification and conversion from one modeling technique to another. Such an approach has some drawbacks, including the loss of information on simple shapes.

- For example, if a circular cylinder (hole) is represented by a B-spline, some data on the specific curve type may be lost unless it is carried along. Data including the fact that the part feature was a cylinder would be useful to manufacturing to identify it as a hole to be drilled or bored rather than a surface to be milled.

2.6 Surface Modeling

- The surface model is an extension of wire frame model which involves providing a surface representation making the object look solid to the viewer.

- Surface systems were in fact the first CAD systems to raise the possibility of integrating the whole industrial process of design and analysis through to the next stages of production and quality control, using the computer as an intermediary.

- Shape design and representation of complex objects such as car, ship, aircraft and castings cannot be achieved by a wire frame model. In such cases, surface models are preferred for representing the objects with precision and accuracy.

- These models also find widespread applications in companies manufacturing forgings, castings and molded products. Such articles are characterized by smoothly curved shapes with blended edges which are not easily represented by conventional engineering drawings.
- Another difficulty for designers with wire frame modelers is to obtain good visualization of their ideas. The surface modeler is ideal for such products as the entire surface is represented; allowing the computer to generate very realistic shaded surface pictures.
- Most surface modelers come equipped with rendering features. In this, the model once created is then provided with surface properties. For example, the surface may be given a property which may make the object appear corroded or made of brass or any such effects.
- A huge material library is generally made available to select the desired surface effect. One can also define different kinds of lights, such as spot lights, ambient lights, etc. in a 3D space and generate a photorealistic image of the object. This feature is highly used in automobile and styling industries for determining the aesthetic appeal of the object being designed.
- Advanced surface modelers go beyond representation and to some extent can be used for property calculations as well. The surface model can be used for generating NC tool paths for continuous path machining, making it an ideal mode for integrating CAD / CAM. A surface modeler can be considered as an extension of a wire frame.
- A wire frame model can be easily extracted from a surface model. It should be remembered that surface models only store the geometry of their corresponding objects and not the topology of these objects. To generate a surface model, a user typically starts with a wire frame model and then connects them with the desired surfaces.
- Once the product has been surface modeled, its geometry may be used directly in the design of the mould or die which is to be made. Essential calculations can usually be performed automatically by the computer such as for example the determination of the volume enclosed by a mould. This gives an immediate idea of the volume of raw material needed to make the product.

Kinds of Surfaces

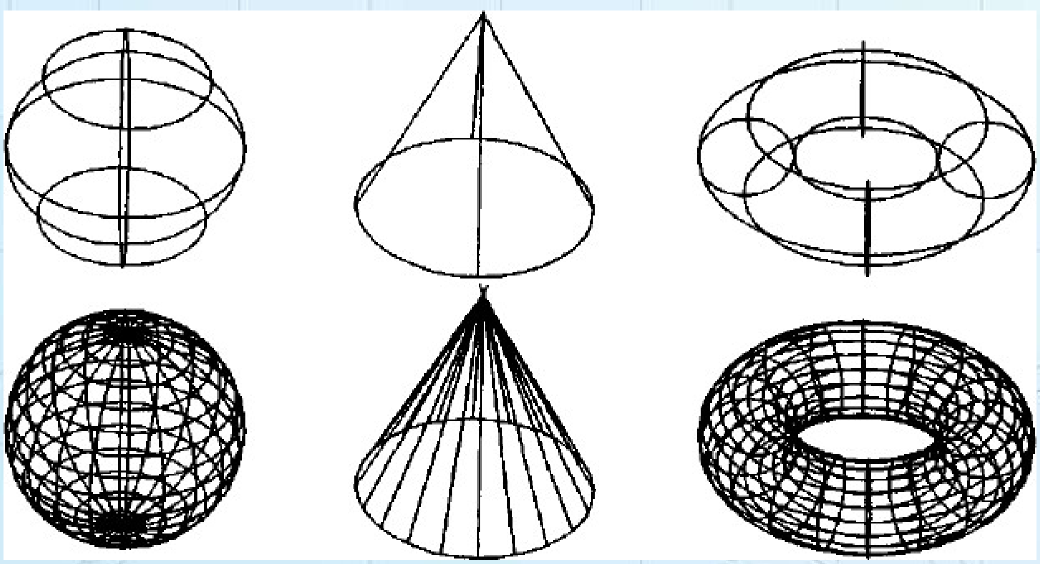

Fig.2.21 – Effect of Mesh Size on Object Visualization

- To create a surface, data related to the desired shape is required. The choice of the surface is dependent on the application. In order to visualize surfaces, a mesh of m x n in size is displayed. The mesh size is controllable and is in the form of criss-cross on the surface. The mesh size plays an important role in proper visualization as can be seen in the objects shown in *Fig.2.21*. However, it should be noted that finer the mesh size of surface entities, longer is the time required by the software to construct and modify the entities.

- A system may need two boundaries to create a ruled surface or might require one entity to create a surface of revolution. Surfaces provided by CAD / CAM software are either analytical or synthetic.

The following are descriptions of major surface entities provided by CAD/CAM systems:

1. **Plane Surface:** This is the simplest surface. It requires three noncoincident points to define an infinite plane. The plane surface can be used to generate cross-sectional views by intersecting a surface model with it, generate cross sections for mass property calculations, or other similar applications where a plane is needed. *Fig.2.22* shows a plane surface.

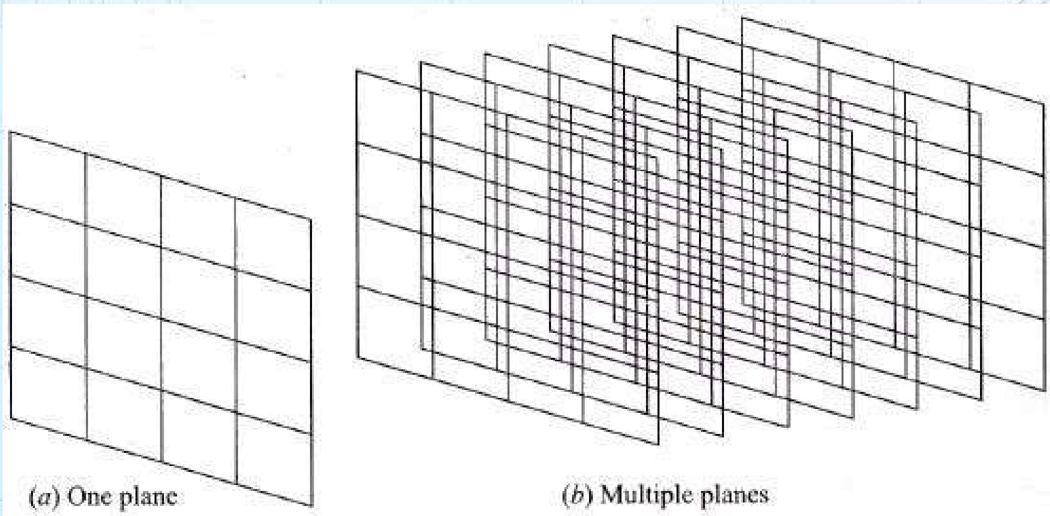

Fig.2.22 – Plane Surface

2. **Ruled (lofted) Surface:** This is a linear surface. It interpolates linearly between two boundary curves that define the surface (rails). Rails can be any wireframe entity. This entity is ideal to represent surfaces that do not have any twists or kinks. *Fig.2.23* gives some examples.

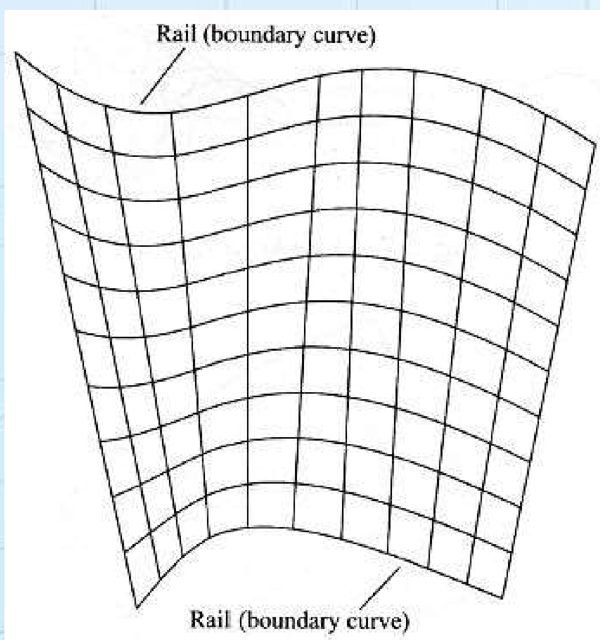

Fig.2.23 – Ruled Surface

3. **Surface of Revolution:** This is an axisymmetric surface that can model axisymmetric objects. It is generated by rotating a planar wireframe entity in space about the axis of symmetry a certain angle (*Fig.2.24*).

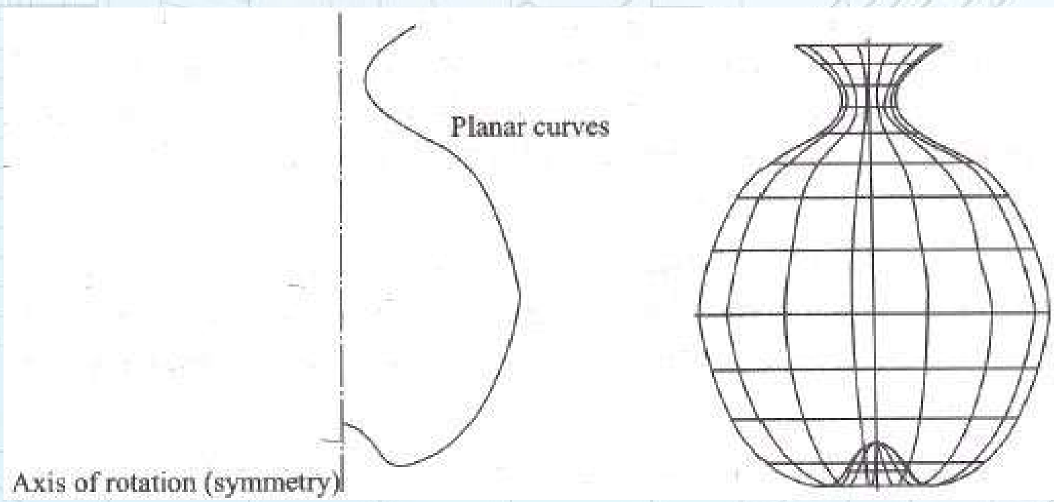

Fig.2.24 – Surface of Revolution

4. **Tabulated Cylinder:** This is a surface generated by translating a planar curve a certain distance along a specified direction (axis of the cylinder) as shown in *Fig.2.25*. The plane of the curve is perpendicular to the axis of the cylinder. It is used to generate surfaces that have identical curved cross sections. The word "tabulated" is borrowed from the APT language terminology.

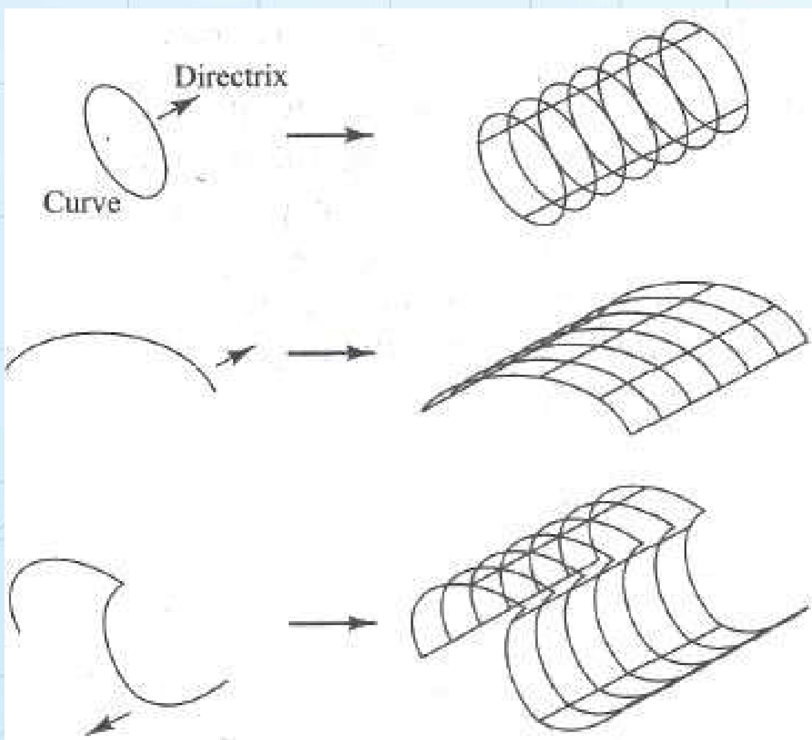

Fig.2.25 – Tabulated Cylinder

5. **Bezier surface:** This is a surface that approximates given input data. It is different from the previous surfaces in that it is a synthetic surface. Similarly to the Bezier curve, it does not pass through all given data points. It is a general surface that permits twists and kinks (*Fig.2.26*). The Bezier surface allows only global control of the surface.

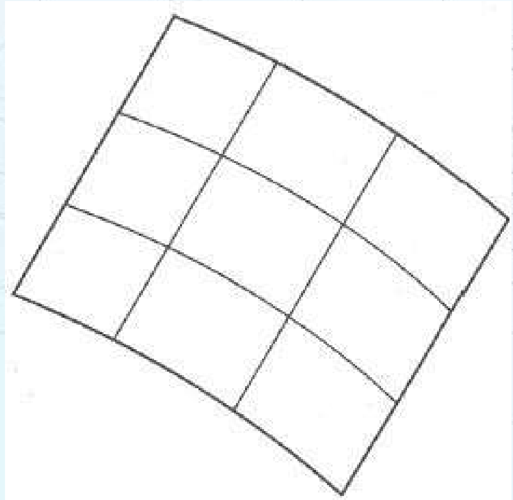

Fig.2.26 – Bezier Surface

6. **B-spline surface:** This is a surface that can approximate or interpolate given input data (*Fig.2.27*). It is a synthetic surface. It is a general surface like the Bezier surface but with the advantage of permitting local control of the surface.

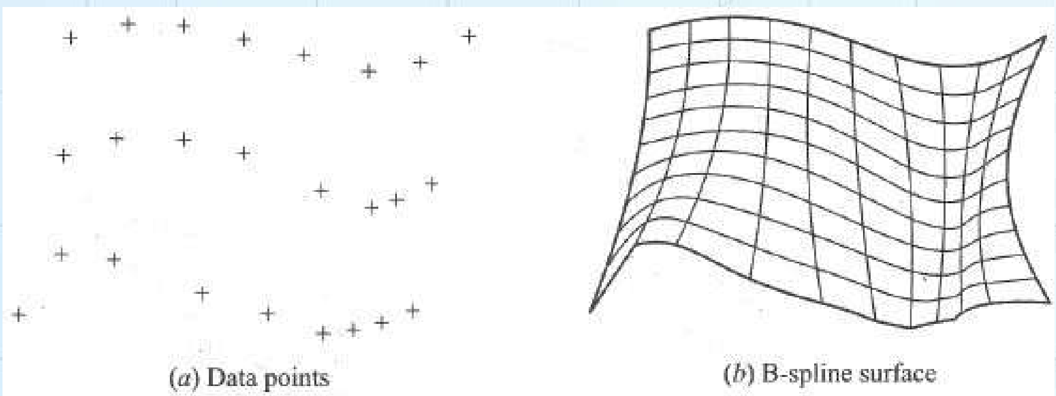

Fig.2.27 – B-spline Surface

7. **Coons Patch:** The above surfaces are used with either open boundaries or given data points. The Coons patch is used to create a surface using curves that form closed boundaries (*Fig.2.28*).

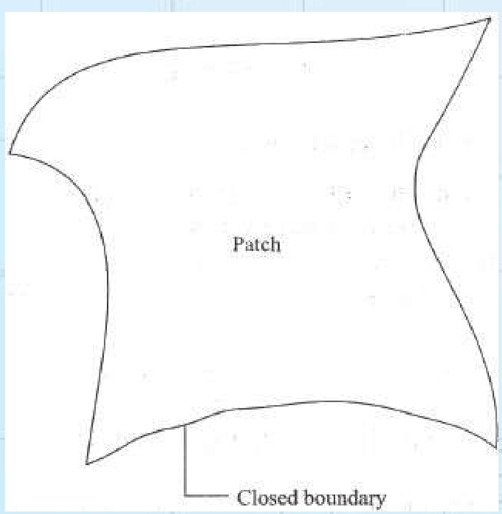

Fig.2.28 – Coons Patch

8. **Fillet surface:** This is a B-spline surface that blends two surfaces together (*Fig.2.29*). The two original surfaces may or may not be trimmed.

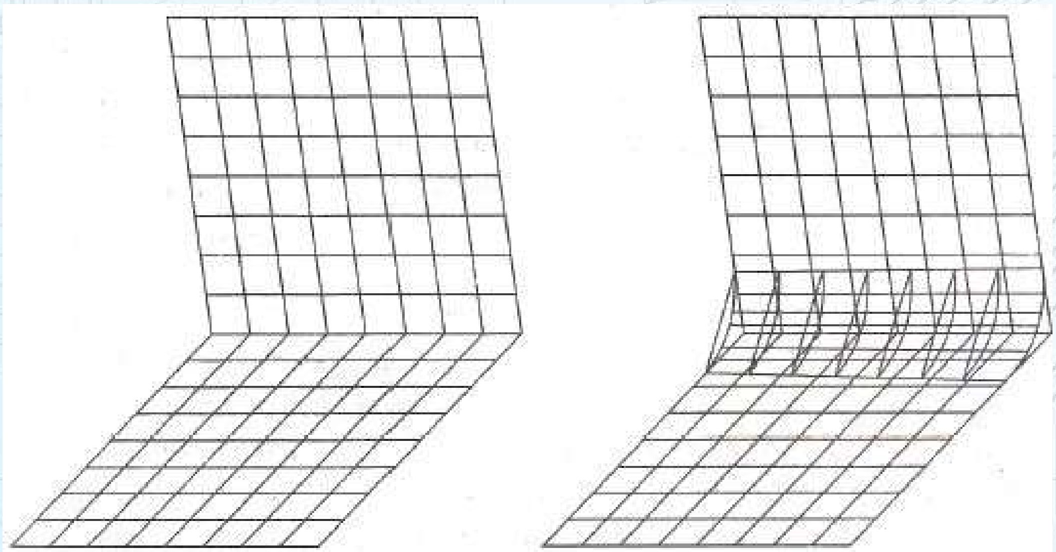

Fig.2.29 – Fillet Surface

9. **Offset surface:** Existing surfaces can be offset to create new ones identical in shape but may have different dimensions. It is a useful surface to use to speed up surface construction. For example, to create a hollow cylinder, the outer or inner cylinder can be created using a cylinder command and the other one can be created by an offset command. Offset surface command becomes very efficient to use if the original surface is a composite one. *Fig.2.30* shows an offset surface.

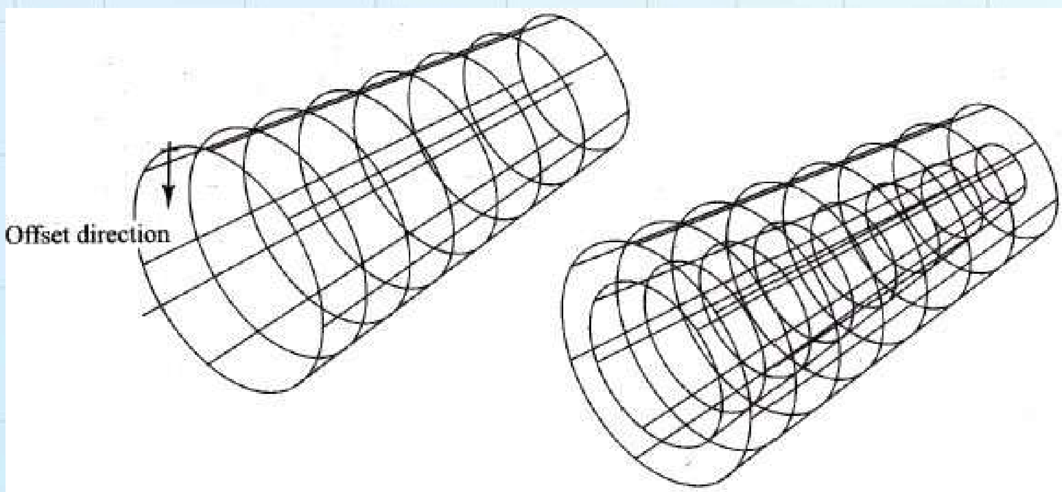

Fig.2.30 – Offset Surface

2.7 References

1) CAD/CAM Principles and Applications by P. N. Rao, Tata McGraw-Hill Publication.
2) Mastering CAD/CAM by Zeid,Ibrahim, Tata McGraw-Hill Publication.
3) CAD/CAM/CIM by Radhakrishnan,P;Subramanyan,S;Raju,V, New Age International (P) Limited.
4) CAD/CAM: Theory and Practice by Zeid,Ibrahim;Subramanian,R, Tata McGraw-Hill Publication.