Government College of Engineering and Research
Avasari, Pune

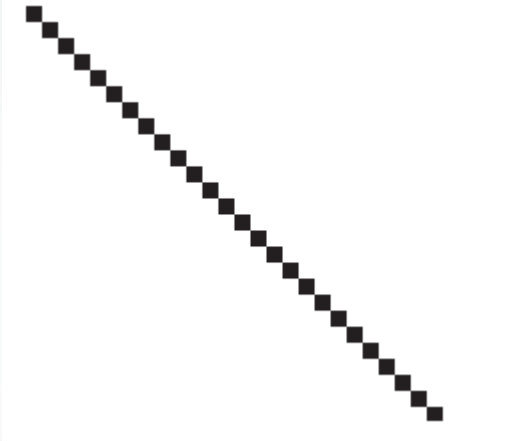
SOLID MODELING AND DRAFTING

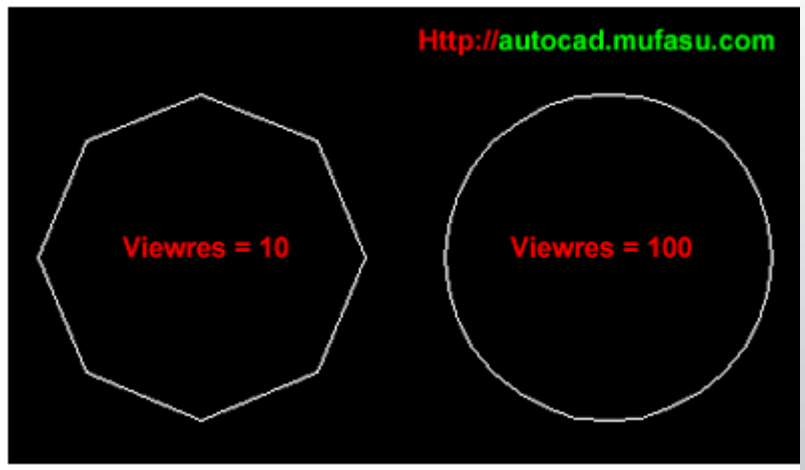
Mr. Sanjay D. Patil
Assistant Professor,
Automobile Department
sanjaypatil365@gmail.com

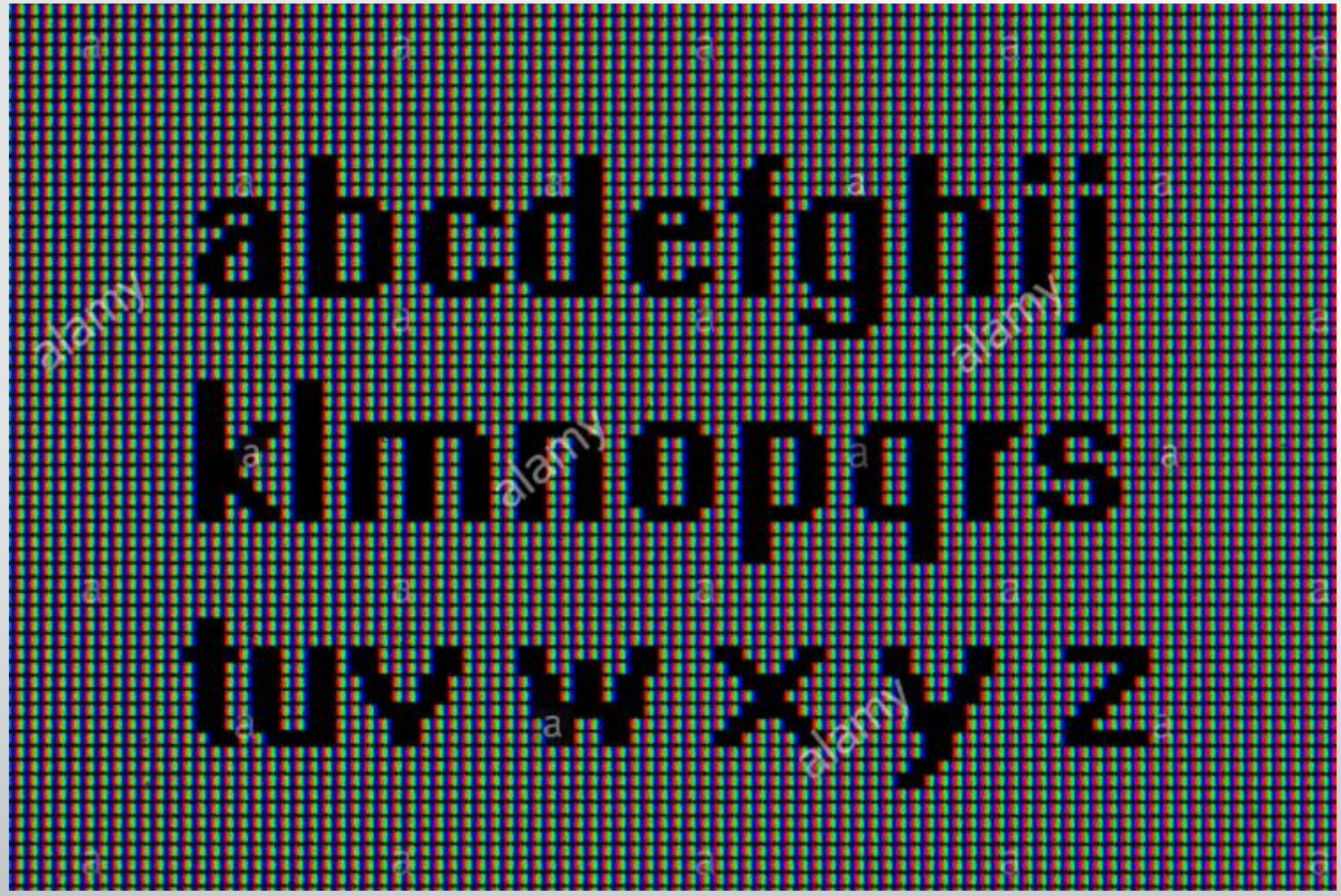
Unit 2

Curves and Surfaces

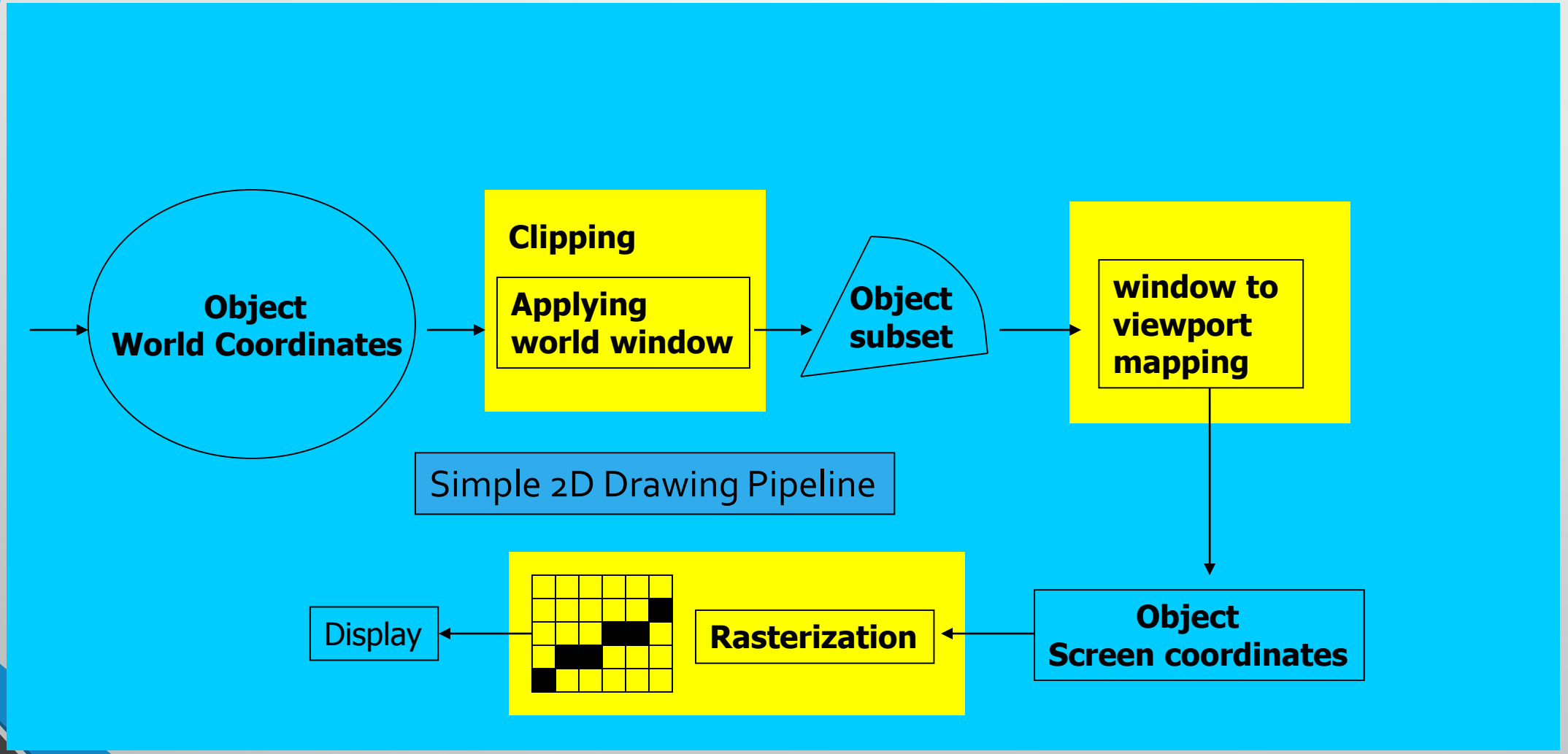
CAD Entities in ZOOM view






abcdefghij
klmnopqrs
tuvwxyz

2D Graphics Pipeline



Rasterization

- A fundamental computer graphics function
- Determine the pixels' colors, illuminations, textures, etc.
- Implemented by graphics hardware
- Rasterization algorithms
  - Lines
  - Circles
  - Triangles
  - Polygons

Line Drawing

Many computer-generated pictures are composed of straight-line segments.

A line segment is displayed by **turning on** a set of **adjacent pixels**.

In order to draw a line, it is necessary to determine which pixels lie nearest the line and provide the best approximation to the desired line.

The line drawing routine should be **accurate**, **fast**, and **easy to implement**.

The Problem of Scan Conversion

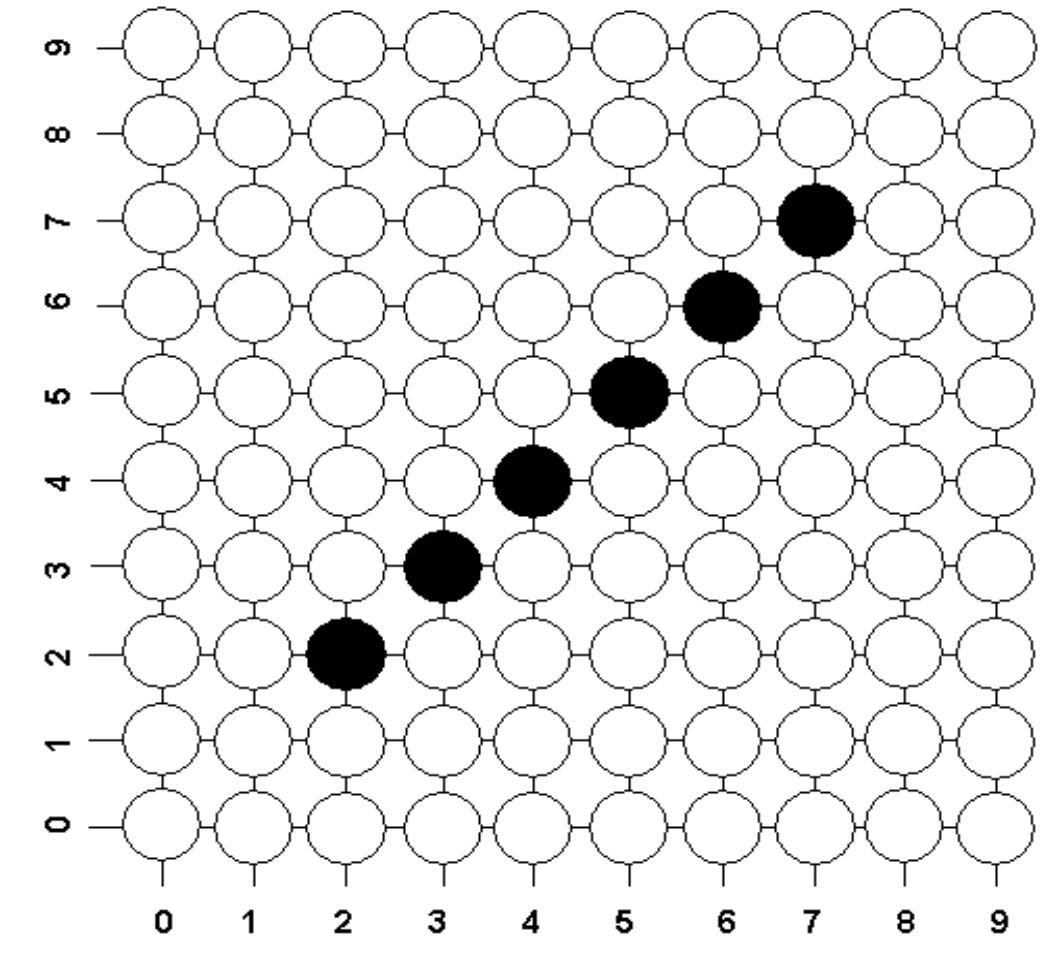
y = mx + b

where, **m** = Slope of the line

b = the y intercept of a line

A line segment is defined by the coordinate positions of the line endpoints.

We have a line with the endpoints (2,2) and (8,7) and we want to draw this line on a pixel based display.



How do we choose which pixels to turn o

Following algorithms are used for line

- Digital Differential Analyzer (simple DDA) Line Drawing Algorithm
- Bresenham 's Line Drawing Algorithm

Simple Digital Differential Analyzer (simple DDA) Line Drawing Algorithm

To illustrate the idea of DDA algorithm, we still want to draw the line segment with endpoints (Xstart, Ystart) and (Xend, Yend) having slope :

$$\mathbf{m} = \frac{\mathbf{Yend - Ystart}}{\mathbf{Xend - Xstart}}$$

Any two consecutive points (x1, y1), (x2, y2) lying on this line satisfies the equation:

$$\frac{\mathbf{y2 - y1}}{\mathbf{x2 - x1}} = \mathbf{m}$$

Simple DDA Line Drawing Algorithm

The algorithm is divided into **two cases** that depend on the absolute value of the slope of the line. **Note that:**

- We should test the line endpoints to ensure that the line is neither horizontal (**Xstart = Xend**) nor vertical (**Ystart =Yend**). If the line is horizontal use the horizontal line drawing algorithm and use the vertical line drawing algorithm when it is vertical.
- The starting and ending points of the line are plotted separately since these values are known from the given data.
- In the two cases below the computed incremental values for y2 and x2 may not be integers. The Round function must be used to obtain an integer coordinate value.

Simple DDA Line Drawing Algorithm

Case 1: For **abs(m) < 1** and **Xstart < Xend**,

we generate the line by incrementing the previous **x** value one unit until **Xend** is reached and then solve for **y**. if **Xstart > Xend,** swap the two endpoints. Thus for these consecutive points:

$$x2 = x1 + 1 \quad \text{or} \quad x2 - x1 = 1$$

Substituting this difference into equation 1 yields:

$$(y2 - y1)/1 = m \quad \text{or} \quad y2 = y1 + m$$

Simple DDA Line Drawing Algorithm

Equation 2 enables us to calculate successive values of y from the previous value by replacing the repeated multiplication with floating point addition.

This method of obtaining the current value by adding a constant to the previous value is an example of **incremental calculation**. Using knowledge of one point to compute the next is a great time saving technique.

The following code can be used to draw a line from (Xstart, Ystart) to (Xend, Yend) using simple DDA algorithm **(case 1)**:

Simple DDA Line Drawing Algorithm

```
m = (Yend-Ystart) / (Xend-Xstart)
If (abs(m)<1 and Xstart>Xend) then
        Swap endpoints  Xstart ↔ Xend and Ystart ↔ Yend
end if
Set pixel (Xstart, Ystart) with desired color
If abs(m) < 1 then
        y = Ystart
        x = Xstart + 1
Next:   y = y + m
        Set pixel (x, Round(y)) with desired color
        x = x + 1

        If x ≤ Xend-1 then go to Next
endif
Set pixel (Xend, Yend) with desired color
```

Simple DDA Line Drawing Algorithm

We will use the **simple DDA algorithm** to draw a line with starting point (2,0) and ending point (7,4) on a pixel based display. Firstly, we compute the **slope m**:

m =(Yend–Ystart)/(Xend–Xstart)=(4–0)/(7–2)=4/5 = 0.8

Δx = (Xend–Xstart)=(7-2)=5

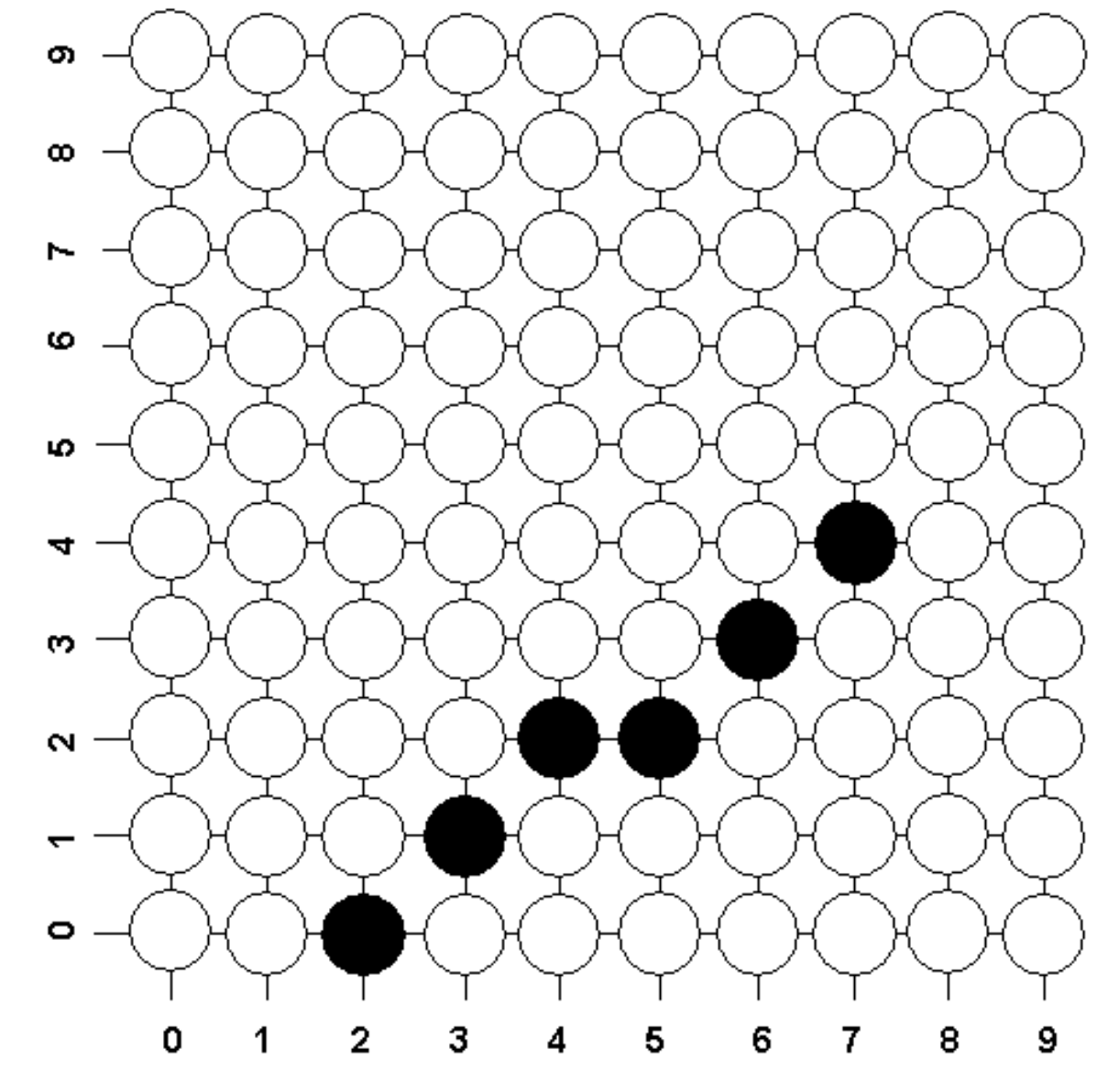
Δy =(Yend–Ystart) =(4–0) =4

Increment in X = Xstart +(Δx/5) = **Xstart** + 1

Increment in Y= Ystart + (Δy/5) = **Ystart** + 0.8

x	y	Round(y)
2	0	
3	y = y + m = 0 + 0.8=0.8	1
4	y = y + m = 0.8 + 0.8=1.6	2
5	y = y + m = 1.6 + 0.8=2.4	2
6	y = y + m = 2.4 + 0.8=3.2	3
7	4	

Simple DDA Line Drawing Algorithm



Geometrical Curve :

- All existing CAD/CAM system provide users with curve entities which can be divided into analytical and synthetic curve

Analytical curve:

i. can be defined by mathematical equation

ii. Analytic entities are points, lines, arcs circle etc

Synthetic curve

i. It is defined by set of data points

ii. Synthetics curve include B- spline curve, Bezier curve etc

Methods of defining Point

- Absolute Cartesian coordinate
- Absolute cylindrical coordinate
- Incremental Cartesian Coordinate
- Incremental Cylindrical Coordinate
- Point of Intersection
- Defining middle or break point

Methods of defining Line

- Define end points
- Parallel or Perpendicular to exiting line
- Vertical or horizontal line
- Tangent to exiting line

Methods of defining Circle

- Radius or diameter and center
- Defining three points
- Center and point on circle
- Tangent to line, pass through a given point and with radius

Methods of defining Ellipse

- Center and axis length
- Four Points
- Two conjugate diameter

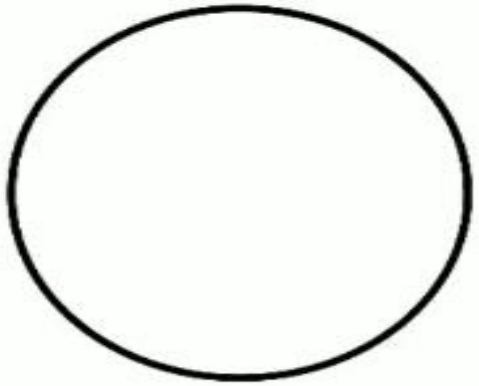
Methods of defining Parabola

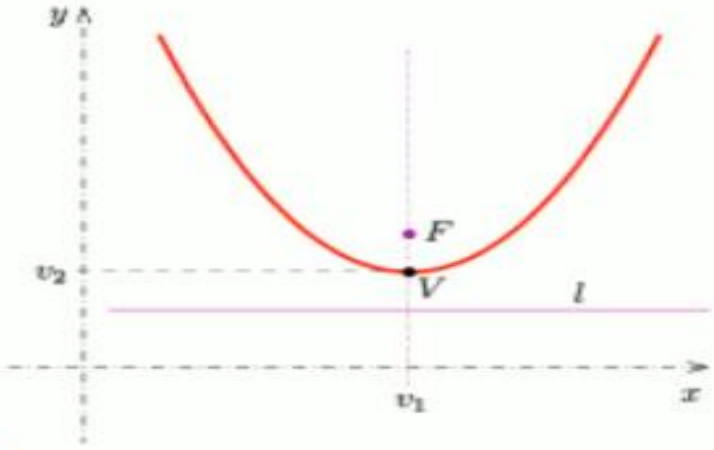
- Vertex and focus
- Three points

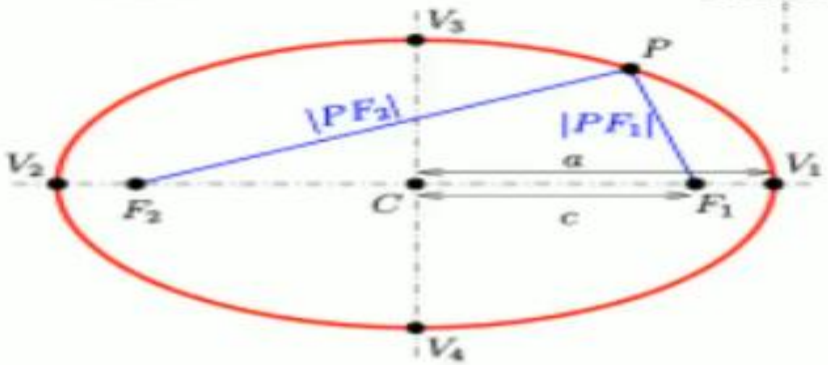
(1) Analytic Curves :

- The curves which are defined by the analytic equations are known as analytic curves.

- Ex. :
- Lines
- Circles
- Ellipse
- Parabolas
- Hyperbolas



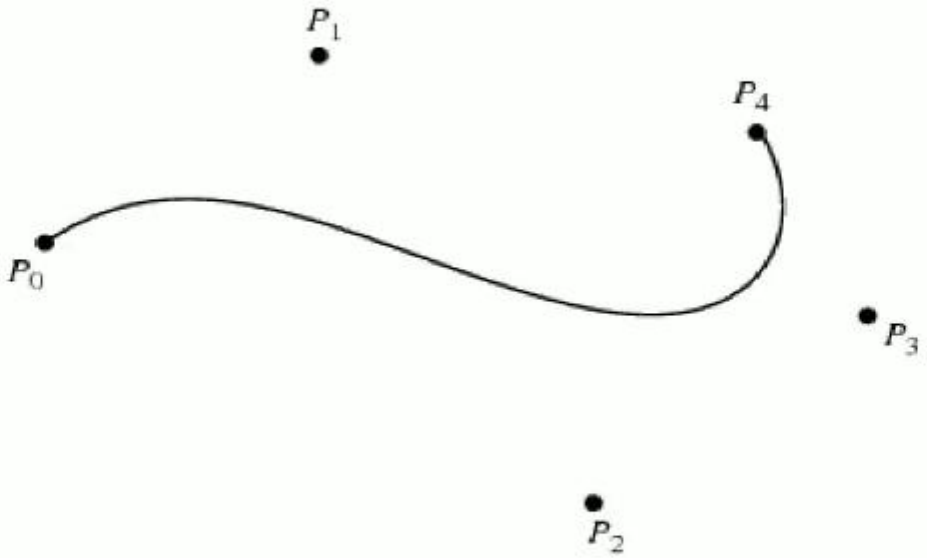




(2) Synthetic Curves :

- The curves which defined by the set of data points are known as..
- The synthetic curves needed when a curve is represented by a collection of data points.
- The synthetic curves are represented by polynomials.

Ex. Of synthetic curves :



Bezier Curves

- Cubic spline, B-Spline, Beta-Spline, Nu-Spline

Application of Synthetic Curves

- The synthetic curves are used for representing profile of :

- Car bodies
- Ship bulls
- Airplane wings
- Propeller blade
- Shoe insoles
- Bottles

Curve Representation

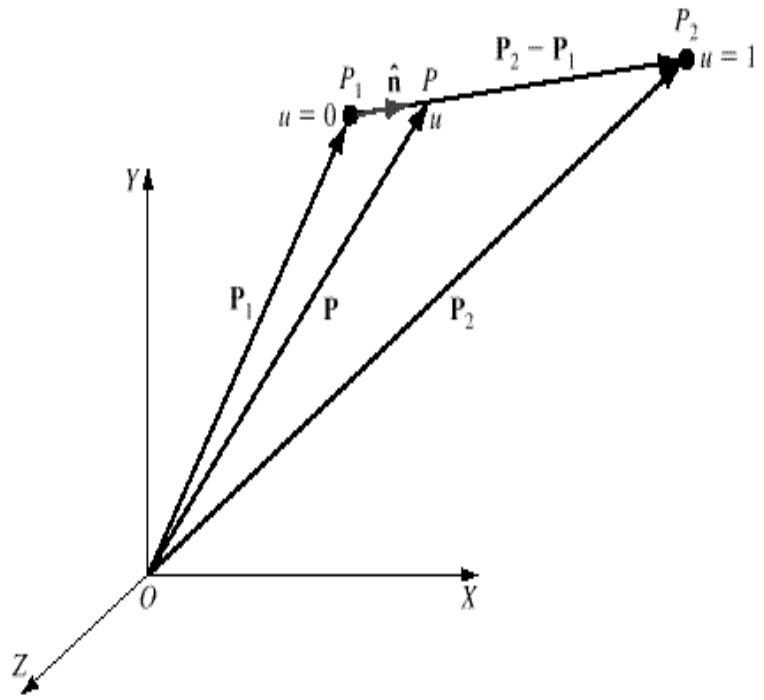
Cartesian or Non parametric

- $P = [x\ y\ z]^T = [x\ \ f(x)\ \ g(x)]^T$
- Curve represented by x, y and z coordinate or one to one relationship
- If the solve of curve is vertical or near to vertical it's value become infinite or very large so, it is difficult to deal with computationally and programme wise
- difficult to represent closed curve
- If curve displayed by as a series of points or by straight line segments, the computation involved could be extensive.

Parametric

- $P = [\, x(u) \;\; y(u) \;\; z(u) \,]^T \qquad u_{min} < u < u_{max}$
- Each point on curve is expressed as a function of parameter u. The parameter acts as a local coordinate for point on curve
- Overcomes the limitation of non parametric curve
- Closed curve can easily define
- Slope is replaced with tangent
- Well suite for computation and display
- large so, it is difficult to deal with computationally and programme wise
- difficult to represent closed curve
- If curve displayed by as a series of points or by straight line segments, the computation involved could be extensive.

Parametric Equation of Line



$$\boldsymbol{P} = \boldsymbol{P}_1 + (\boldsymbol{P} - \boldsymbol{P}_1)$$

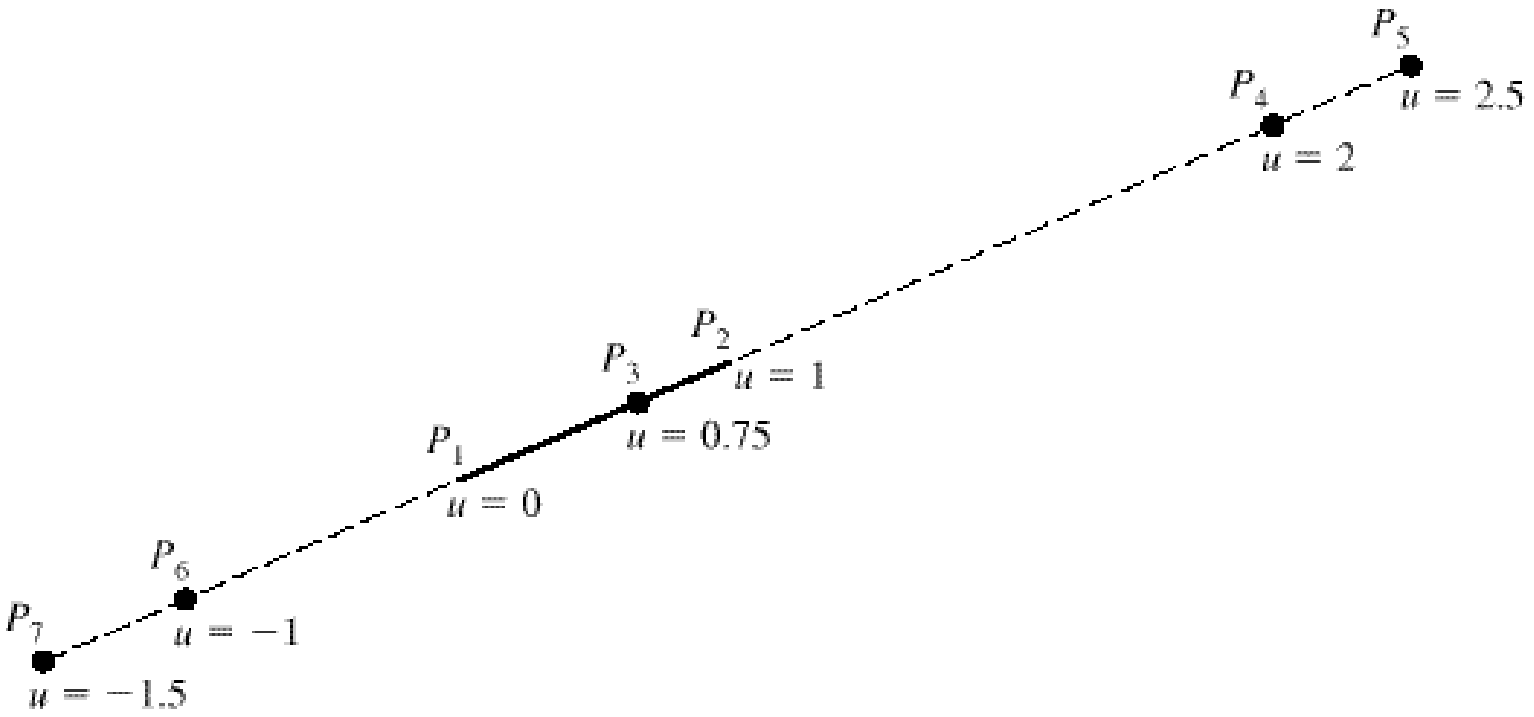
$$\boldsymbol{P} - \boldsymbol{P}_1 = u\,(\boldsymbol{P}_2 - \boldsymbol{P}_1)$$

$$\boldsymbol{P} = \boldsymbol{P}_1 + u(\boldsymbol{P}_2 - \boldsymbol{P}_1), \quad 0 \le u \le 1$$

$$\mathbf{P} = \mathbf{P}_1 + u\ \mathbf{P}_2 - \mathbf{P}_1 \qquad 0 \quad u \quad 1$$

In scalar form, this equation can be written as

$$\begin{aligned} x &= x_1 + u\ x_2 - x_1 \\ y &= y_1 + u\ y_2 - y_1 \\ z &= z_1 + u\ z_2 - z_1 \end{aligned} \qquad 0 \quad u \quad 1$$



The tangent vector of the line is given by

$$\mathbf{P} = \mathbf{P}_2 - \mathbf{P}_1 \tag{6.13}$$

or, in scalar form,

$$\begin{aligned} x &= x_2 - x_1 \\ y &= y_2 - y_1 \\ z &= z_2 - z_1 \end{aligned} \tag{6.14}$$

The independence of the tangent vector from u reflects the constant slope of the straight line. For a 2D line, the known infinite (vertical line) and zero (horizontal line) slope conditions can be generated from Eq. (6.14).

The unit vector in the direction of the line is given by

$$\hat{\mathbf{n}} = \frac{\mathbf{P}_2 - \mathbf{P}_1}{L} \tag{6.15}$$

where L is the length of the line,

$$L = |\mathbf{P}_2 - \mathbf{P}_1| = \sqrt{x_2 - x_1\ ^2 + \ y_2 - y_1\ ^2 + \ z_2 - z_1\ ^2} \tag{6.16}$$

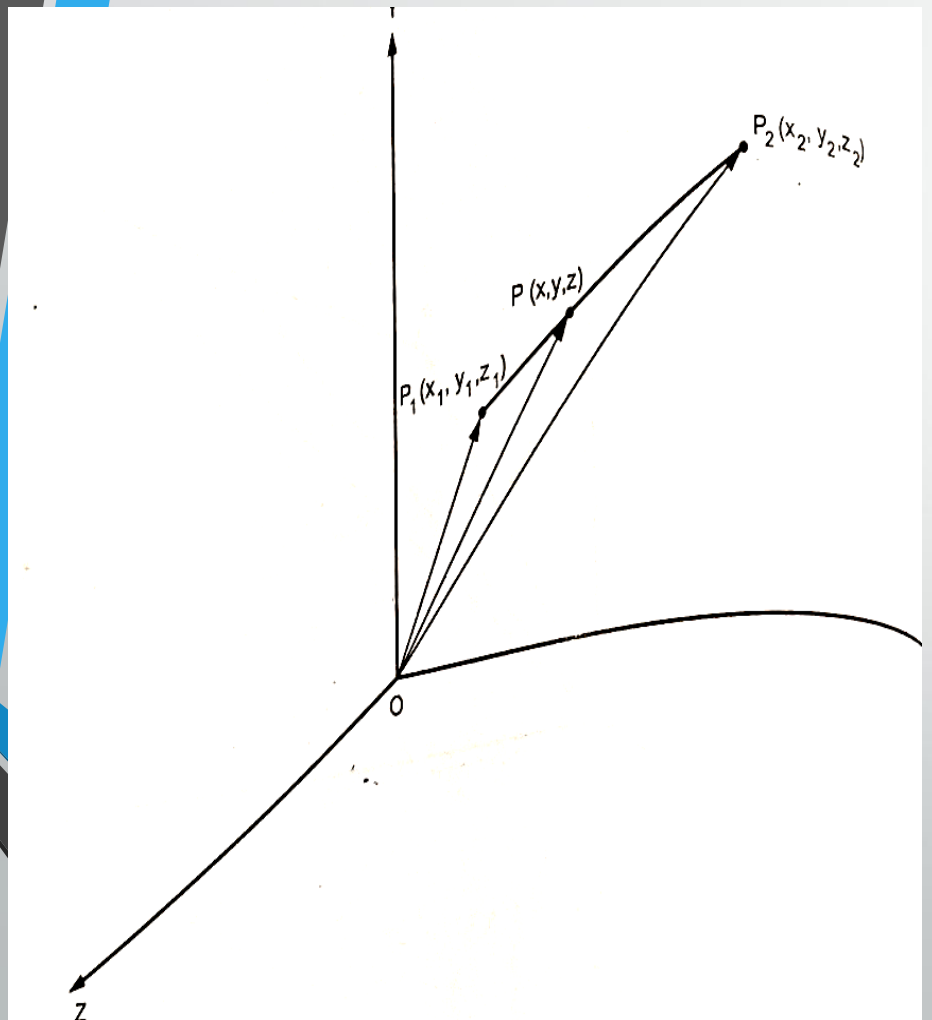
Pro.1. A Line joins two Points(3, 4, 6) & (5, 7, 1)

Find

i) The Parametric Equation of Line

ii) The tangent vector of Line

iii) The unit vector in the direction of Line



1. **Parametric equation of line :**

The parametric equation of line P_1P_2 shown in Fig. 3.1.1, is written as,

$$P = P_1 + u(P_2 - P_1) \qquad 0 \le u \le 1$$

$$= \begin{bmatrix} 3 \\ 4 \\ 6 \end{bmatrix} + u \begin{bmatrix} 5-3 \\ 7-4 \\ 1-6 \end{bmatrix} \qquad 0 \le u \le 1$$

$$\therefore \quad P = \begin{bmatrix} 3 \\ 4 \\ 6 \end{bmatrix} + u \begin{bmatrix} 2 \\ 3 \\ -5 \end{bmatrix} \qquad 0 \le u \le 1$$

$$\text{or} \quad P = [3\ 4\ 6]^T + u[2\ 3\ -5]^T \qquad 0 \le u \le 1$$

2. **Tangent vector of line :**

The tangent vector of the line is given by,

$$P' = P_2 - P_1$$

$$= [i\ j\ k] \begin{bmatrix} 5-3 \\ 7-4 \\ 1-6 \end{bmatrix}$$

$$= [i\ j\ k] \begin{bmatrix} 2 \\ 3 \\ -5 \end{bmatrix}$$

or $$\mathbf{P' = 2i + 3j - 5k}$$ ...Ans.

3. **Unit vector in direction of line :**

$$L = |P_2 - P_1|$$

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

$$= \sqrt{(5-3)^2 + (7-4)^2 + (1-6)^2}$$

or $$L = 6.164$$

The unit vector in the direction of line is given by,

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

$$= \frac{P'}{L}$$

$$= \frac{2i + 3j - 5k}{6.164}$$

or $$\hat{n} = \mathbf{0.324\,i + 0.486\,j - 0.811\,k}$$ ...Ar

Pro.2. Line L1 has end points (1, 2, 7) & (5, 6, 1), while line L2 has end points (7, 3, 4) & (3, 9, 10)

i) The Parametric Equation of Line

ii) The tangent vector of Line

iii) Are the tow lines are Parallel or Perpendicular?

Iv) Are the two lines are intersecting? If yes, find point of Intersection.

Solution :-

Given: P_1 (1, 2, 7) , P_2(5, 6, 1) , P_3 (7, 3, 4) & P_4 (3, 9, 10)

1. **Parametric equations of lines :**

- **For line L_1 :**

$$P_{L1} = P_1 + u(P_2 - P_1) \qquad 0 \le u \le 1$$

$$= \begin{bmatrix} 1 \\ 2 \\ 7 \end{bmatrix} + u \begin{bmatrix} 5-1 \\ 6-2 \\ 1-7 \end{bmatrix} \qquad 0 \le u \le 1$$

$$= \begin{bmatrix} 1 \\ 2 \\ 7 \end{bmatrix} + u \begin{bmatrix} 4 \\ 4 \\ -6 \end{bmatrix} \qquad 0 \le u \le 1$$

or $$\mathbf{P_{L1} = [1\ 2\ 7]^T + u[4\ \ 4\ \ -6]^T \quad 0 \le u \le 1}$$

- **For line L_2 :**

$$P_{L2} = P_3 + v(P_4 - P_3) \qquad 0 \le v \le 1$$

$$= \begin{bmatrix} 7 \\ 3 \\ 4 \end{bmatrix} + v \begin{bmatrix} 3-7 \\ 9-3 \\ 10-4 \end{bmatrix} \qquad 0 \le v \le 1$$

$$= \begin{bmatrix} 7 \\ 3 \\ 4 \end{bmatrix} + v \begin{bmatrix} -4 \\ 6 \\ 6 \end{bmatrix} \qquad 0 \le v \le 1$$

or $$\mathbf{P_{L2} = [7\ \ 3\ \ 4]^T + v[-4\ \ 6\ \ 6]^T \quad 0 \le v \le 1}$$

2. **Tangent vectors of lines :**

- **For line L_1 :**

$$P'_{L1} = P_2 - P_1$$

$$= [i\ \ j\ \ k] \begin{bmatrix} 5-1 \\ 6-2 \\ 1-7 \end{bmatrix}$$

$$= [i\ \ j\ \ k] \begin{bmatrix} 4 \\ 4 \\ -6 \end{bmatrix}$$

or $$\mathbf{P'_{L1} = 4i + 4j - 6k}$$ ...Ans.

- **For line L_2 :**

$$P'_{L2} = P_4 - P_3$$

$$= [i\ \ j\ \ k] \begin{bmatrix} 3-7 \\ 9-3 \\ 10-4 \end{bmatrix}$$

$$= [i\ \ j\ \ k] \begin{bmatrix} -4 \\ 6 \\ 6 \end{bmatrix}$$

or $$\mathbf{P'_{L2} = -4i + 6j + 6k}$$ ...Ans.

3. **Are two lines perpendicular ?**

If two lines L_1 and L_2 are perpendicular, the dot product of their tangent vectors should be zero.

$$P'_{L1} \cdot P'_{L2} = (4i + 4j - 6k) \cdot (-4i + 6j + 6k)$$

$$= (4 \times -4) + (4 \times 6) + (-6 \times 6)$$

$$= -16 + 24 - 36$$

or $$\mathbf{P'_{L1} \cdot P'_{L2} = -28}$$

As $P'_{L1} \cdot P'_{L2} \neq 0$, two lines L_1 and L_2 are not perpendicular.

4. **Are two lines parallel ?**

If two lines L_1 and L_2 are parallel, the cross product of their tangent vectors should be zero.

$$P'_{L1} \times P'_{L2} = \begin{vmatrix} i & j & k \\ 4 & 4 & -6 \\ -4 & 6 & 6 \end{vmatrix}$$

$$= (24 + 36)\,i - (24 - 24)\,j + (24 + 16)\,k$$

or $\mathbf{P'_{L1} \times P'_{L2} = 60i + 40k}$

As $P'_{L1} \times P'_{L2} \neq 0$, the two lines L_1 and L_2 are not parallel.

5. **Are two lines intersecting ?**

- If two lines L_1 and L_2 are intersecting then at the point of intersection,

$$P_{L1} = P_{L2}$$

- From Equations (a) and (b),

$$\begin{bmatrix} 1 \\ 2 \\ 7 \end{bmatrix} + u \begin{bmatrix} 4 \\ 4 \\ -6 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \\ 4 \end{bmatrix} + v \begin{bmatrix} -4 \\ 6 \\ 6 \end{bmatrix}$$

$$\therefore \quad 1 + 4u = 3 - 4v$$

$$2 + 4u = 7 + 6v$$

$$\text{and} \quad 7 - 6u = 4 + 6v$$

$$\therefore \quad 4u + 4v = 2$$

$$4u - 6v = 5$$

$$6u + 6v = 3$$

- If two lines L_1 and L_2 are intersecting, then Equations (c), (d) and (e) satisfy.
- From Equations (c) and (d),

$$10v = -3$$

$$\therefore \quad \mathbf{v = -0.3}$$

- From Equation (c),

$$4u - 4 \times 0.3 = 2$$

$$4u = 3.2$$

$$\therefore \quad \mathbf{u = 0.8}$$

- From Equation (e),

$$\text{L.H.S.} = 6u + 6v$$

$$= 6 \times 0.8 - 6 \times 0.3$$

$$= 4.8 - 1.8$$

$$= 3$$

$$= \text{R.H.S.}$$

- Hence, Equation (e) satisfies the values obtained from Equations (c) and (d).

Therefore two lines L_1 and L_2 are intersecting. ...Ans.

6. **Point of intersection :**

- The point of intersection can be obtained by substituting value of either $u = 0.8$ in Equation (a) or $v = -0.3$ in Equation (b).
- From Equation (a),

$$P_{int} = \begin{bmatrix} 1 \\ 2 \\ 7 \end{bmatrix} + 0.8 \begin{bmatrix} 4 \\ 4 \\ -6 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 2 \\ 7 \end{bmatrix} + \begin{bmatrix} 3.2 \\ 3.2 \\ -4.8 \end{bmatrix}$$

$$= \begin{bmatrix} 4.2 \\ 5.2 \\ 2.2 \end{bmatrix}$$

or $\mathbf{P_{int} = (4.2, 5.2, 2.2)}$ ...Ans.

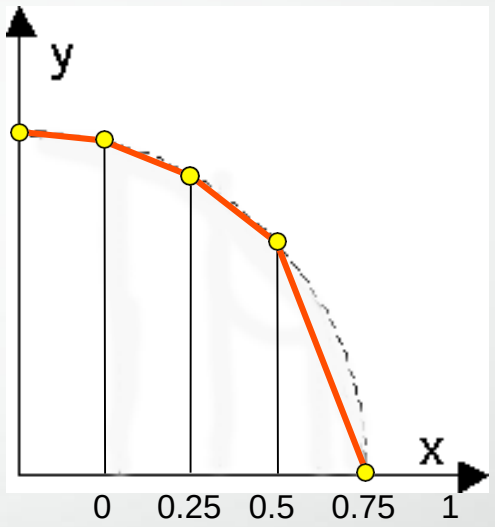
Parametric Equation of Circle

Representation 1
Non-parametric Representation

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

$$x = u \qquad (a)$$

$$y = \sqrt{1 - u^2}$$



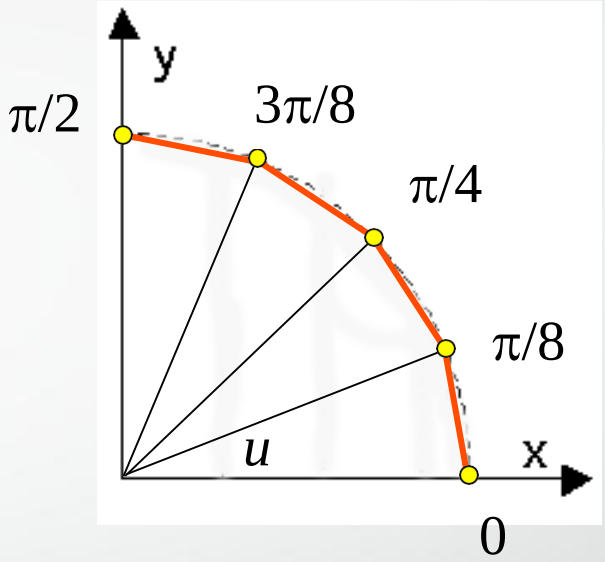
Limitations of non parametric representation

- poor and non-uniform definition
- square root complicated to compute

Representation 2 :
Parametric representation

$$x = \cos u$$
$$y = \sin u$$

(b)



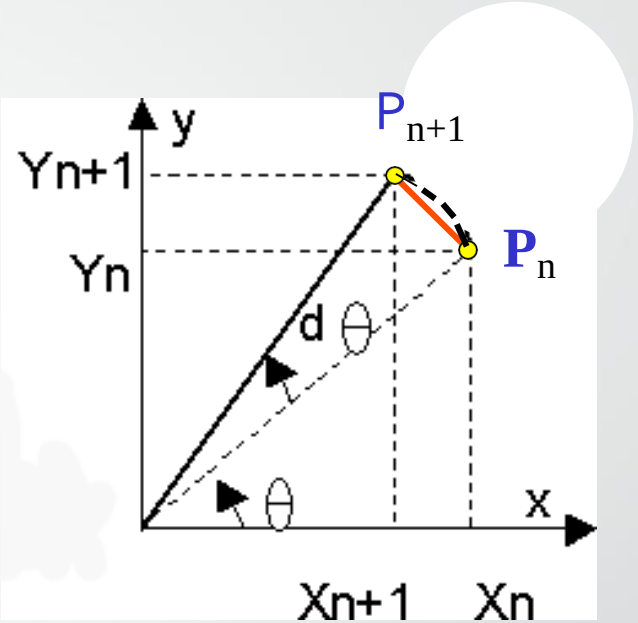
Limitations of non parametric representation

- better definition than non parametric representation
- Inefficient due to computing the trigonometric function at each point

Representation 3
Parametric Representation with recursive approach

Parametric equation for point P_n with centre of circle is origin

$$x_n = r\cos\theta$$
$$y_n = r\sin\theta$$



Now, assuming there is $d\theta$ increment between two consecutive P_n and P_{n+1}

$$x_{n+1} = r\cos(\theta + d\theta)$$
$$= r\cos\theta\cos d\theta - r\sin\theta\sin d\theta$$

$$\boxed{\begin{aligned} x_{n+1} &= x_n\cos d\theta - y_n\sin d\theta \\ y_{n+1} &= y_n\cos d\theta + x_n\sin d\theta \end{aligned}}$$

Coordinate of point P_{n+1}

Observation:

- curves are represented by a series of line-segments
- Trigonometric function cos$d\theta$ and sin$d\theta$ have to calculated once only, hence it speed up the circl generation and display

Parametric equation for point P_n with centre of circle x_c and y_c origin

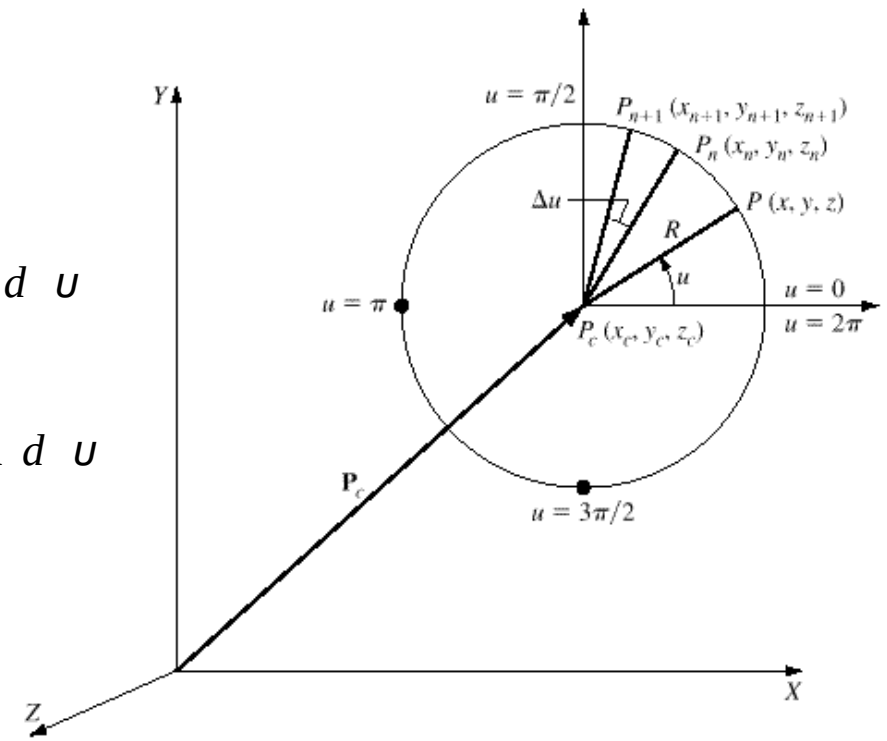
$$x_n = x_c + r\cos u$$

$$y_n = y_c + r\sin u$$

The basic parametric equation of a circle can be written as (refer to Figure 6.11):

$$x_n - x_c = r\cos u$$

$$y_n - y_c = r\sin u$$



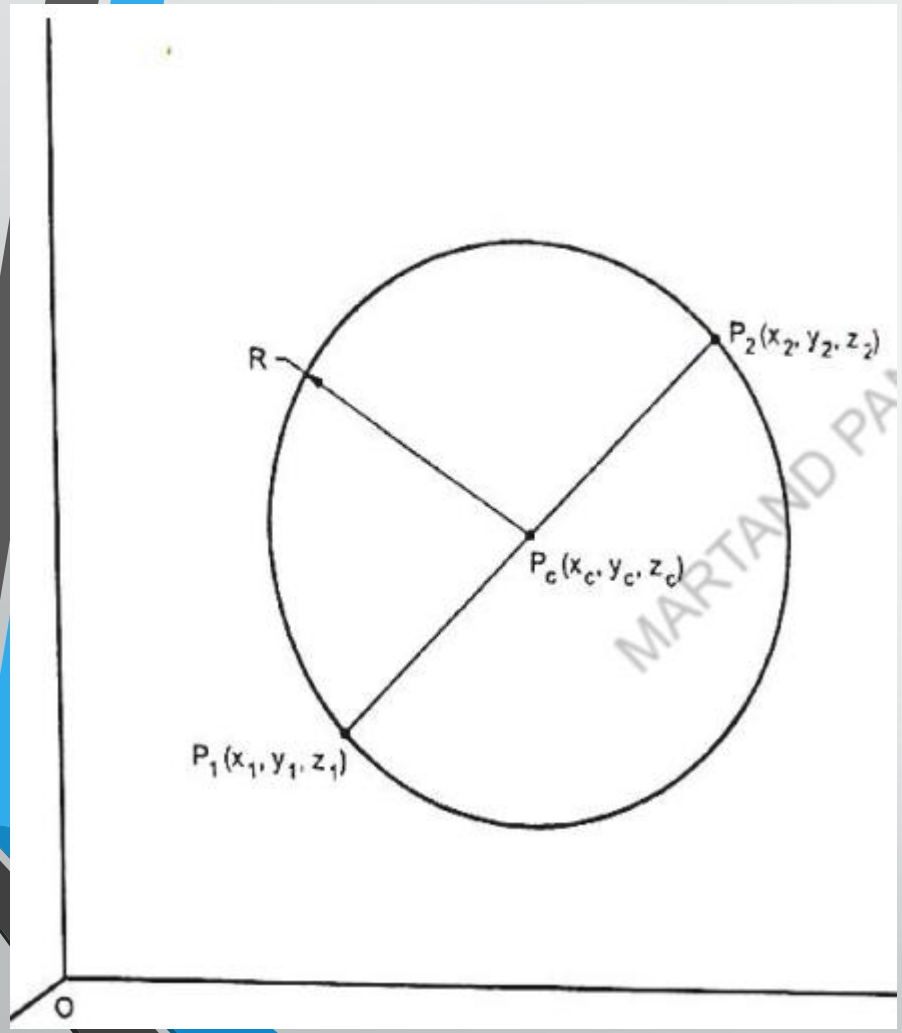
$$x_{n+1} = x_c + r\cos(u + du)$$
$$= x_c + r\cos u\cos du - r\sin u\sin du$$

$$y_{n+1} = y_c + r\sin(u + du)$$
$$= y_c + r\sin u\cos du - r\cos u\sin du$$

$$x_{n+1} = x_c + (x_n - x_c)\cos du - (y_n - y_c)\sin du$$

$$y_{n+1} = y_c + (y_n - y_c)\cos du - (x_n - y_c)\sin du$$

Determination of Center and Radius of circle from two end points of Diameter



The center 'P_c' of a circle is given by,

$$P_c = \frac{1}{2}[P_1 + P_2]$$

$$\text{or } [x_c \;\; y_c \;\; z_c]^T = \left[\frac{x_1+x_2}{2} \;\; \frac{y_1+y_2}{2} \;\; \frac{z_1+z_2}{2}\right]^T$$

The radius '**R**' of circle is given by,

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

Write a Parametric equation for a circle having center at (3, 4, 2) and passing through the point (7, 6, 2). Calculate the coordinates of points on circle.

Solution :

Given : $P_c(x_c, y_c, z_c) \equiv P_c(3, 4, 2)$;

$P_n(x_n, y_n, z_n) \equiv P_n(7, 6, 2)$.

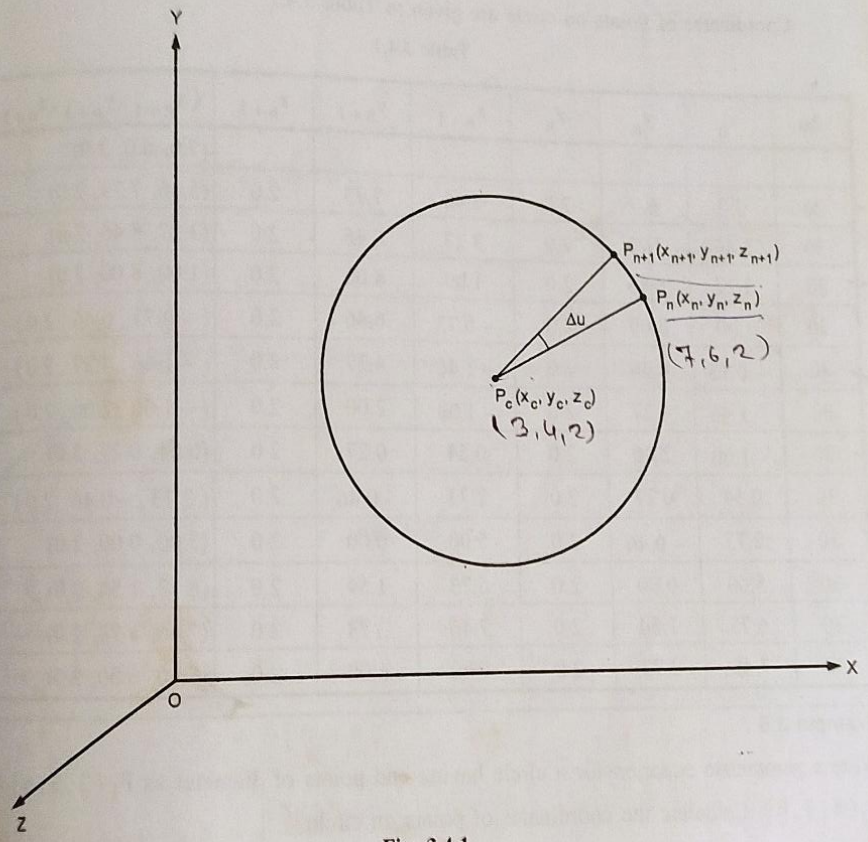


Fig. 3.4.1

1. **Parametric equation :**

From Fig. 3.4.1,

$$x_{n+1} = x_c + (x_n - x_c)\cos\Delta u - (y_n - y_c)\sin\Delta u$$

$$y_{n+1} = y_c + (y_n - y_c)\cos\Delta u + (x_n - x_c)\sin\Delta u$$

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

$$\therefore \quad \left.\begin{aligned} x_{n+1} &= 3 + (x_n - 3)\cos\Delta u - (y_n - 4)\sin\Delta u \\ y_{n+1} &= 4 + (y_n - 4)\cos\Delta u + (x_n - 3)\sin\Delta u \\ z_{n+1} &= 2 \end{aligned}\right\} \quad \text{...Ans.}$$

2. **Coordinates of points on circle :**

Coordinates of points on circle are given in Table 3.4.1

Table 3.4.1

Δu	x_n	y_n	z_n	x_{n+1}	y_{n+1}	z_{n+1}	$(x_{n+1}, y_{n+1}, z_{n+1})$
							(7.0, 6.0, 2.0)
30	7.0	6.0	2.0	5.46	7.73	2.0	(5.46, 7.73, 2.0)
30	5.46	7.73	2.0	3.27	8.46	2.0	(3.27, 8.46, 2.0)
30	3.27	8.46	2.0	1.00	8.00	2.0	(1.00, 8.00, 2.0)
30	1.00	8.00	2.0	– 0.73	6.46	2.0	(– 0.73, 6.46, 2.0)
30	– 0.73	6.46	2.0	– 1.46	4.27	2.0	(– 1.46, 4.27, 2.0)
30	– 1.46	4.27	2.0	– 1.00	2.00	2.0	(– 1.00, 2.00, 2.0)
30	– 1.00	2.00	2.0	0.54	0.27	2.0	(0.54, 0.27, 2.0)
30	0.54	0.27	2.0	2.73	– 0.46	2.0	(2.73, – 0.46, 2.0)
30	2.73	– 0.46	2.0	5.00	0.00	2.0	(5.00, 0.00, 2.0)
30	5.00	0.00	2.0	6.73	1.54	2.0	(6.13, 1.54, 2.0)
30	6.73	1.54	2.0	7.46	3.73	2.0	(7.46, 3.73, 2.0)
30	7.46	3.73	2.0	7.00	6.00	2.0	(7.00, 6.00, 2.0)

Prob. 2. P1(2, 3, 6) & P2 (8, 7, 6)

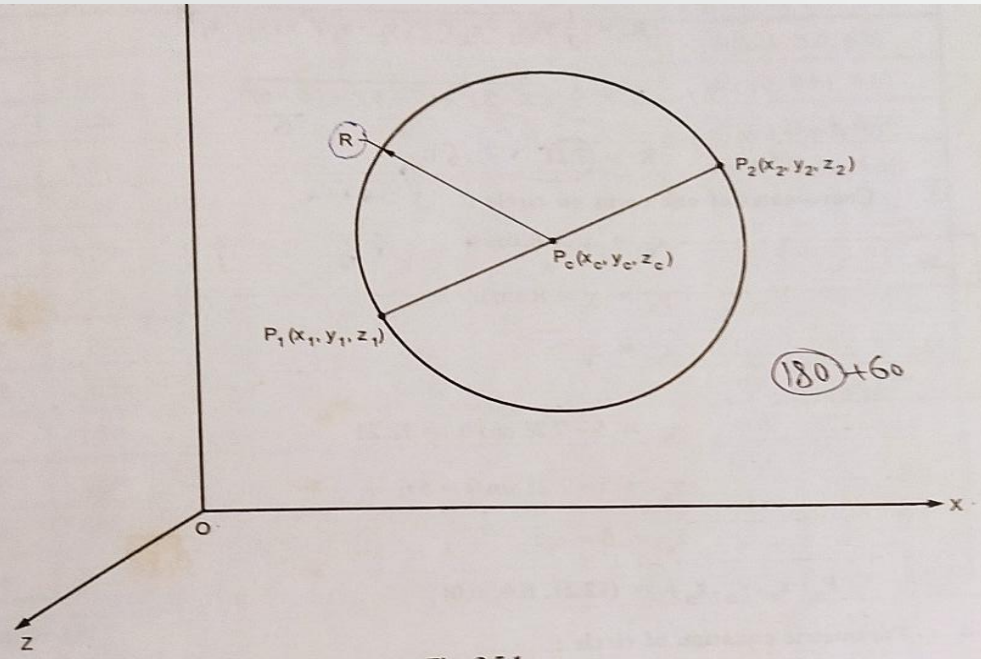


Fig. 3.5.1

Refer Fig. 3.5.1 ;

1. **Center of circle :**

The center of circle is,

$$P_c = \frac{1}{2}[P_1 + P_2]$$

$$(x_c, y_c, z_c) = \left[\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}, \frac{z_1+z_2}{2}\right]$$

$$= \left[\frac{2+8}{2}, \frac{3+7}{2}, \frac{6+6}{2}\right]$$

or $(x_c, y_c, z_c) = (5, 5, 6)$.

2. **Radius of circle :**

The radius of circle is,

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

$$= \frac{1}{2}\sqrt{(8-2)^2+(7-3)^2+(6-6)^2}$$

6 4 0

or $R = 7.21$ 3.60 $\sqrt{36+16}$

3. **Coordinates of one point on circle :**

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

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

$$z_n = z_c$$

At u = 0 ;

$$x_n = 5 + 7.21\cos 0 = 12.21$$

$$y_n = 5 + 7.21\sin 0 = 5.0$$

$$z_n = 6 = 6.0$$

$\therefore$ $P_n(x_n, y_n, z_n) = (12.21, 5.0, 6.0)$

4. **Parametric equation of circle :**

$$x_{n+1} = x_c + (x_n - x_c)\cos\Delta u - (y_n - y_c)\sin\Delta u$$

$$y_{n+1} = y_c + (y_n - y_c)\cos\Delta u + (x_n - x_c)\sin\Delta u$$

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

$$\therefore \left.\begin{aligned} x_{n+1} &= 5 + (x_n - 5)\cos\Delta u - (y_n - 5)\sin\Delta u \\ y_{n+1} &= 5 + (y_n - 5)\cos\Delta u + (x_n - 5)\sin\Delta u \\ z_{n+1} &= 6 \end{aligned}\right\} \quad \text{...Ans.}$$

5. **Coordinates of points on circle :**

The coordinates of points on circle are given in Table 3.5.1.

Table 3.5.1

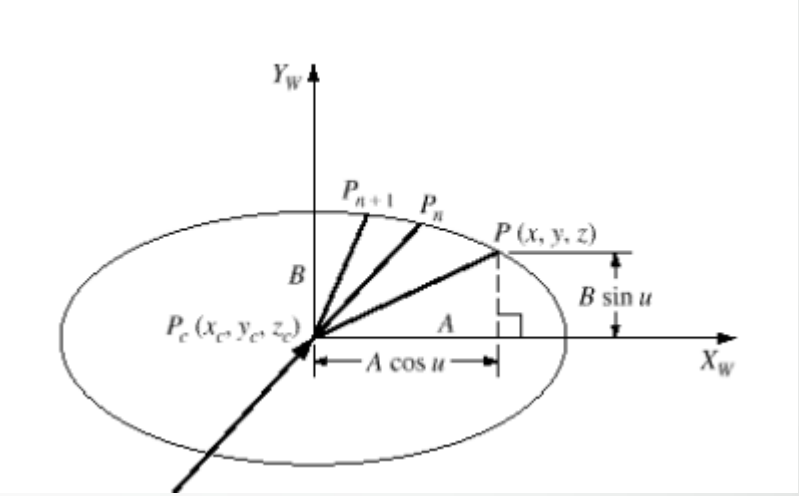
Δu	x_n	y_n	z_n	x_{n+1}	y_{n+1}	z_{n+1}	$(x_{n+1}, y_{n+1}, z_{n+1})$
30	12.21	5.00	6.0	11.24	8.61	6.0	(12.21, 5.0, 6.0)
30	11.24	8.61	6.0	8.61	11.24	6.0	(11.24, 8.61, 6.0)
30	8.61	11.24	6.0	5.00	12.21	6.0	(8.61, 11.24, 6.0)
30	5.00	12.21	6.0	1.39	11.24	6.0	(5.00, 12.21, 6.0)
30	1.39	11.24	6.0	– 1.24	8.61	6.0	(1.39, 11.24, 6.0)
30	– 1.24	8.61	6.0	– 2.21	5.00	6.0	(– 1.24, 8.61, 6.0)
30	– 2.21	5.00	6.0	– 1.24	1.39	6.0	(– 2.21, 5.00, 6.0)
30	– 1.24	1.39	6.0	1.39	– 1.24	6.0	(– 1.24, 1.39, 6.0)
30	1.39	– 1.24	6.0	5.00	– 2.21	6.0	(1.39, – 1.24. 6.0)
30	5.00	– 2.21	6.0	8.61	– 1.24	6.0	(5.00, – 2.21, 6.0)
30	8.61	– 1.24	6.0	11.24	1.39	6.0	(8.61, – 1.24, 6.0)
30	11.24	1.39	6.0	12.21	5.00	6.0	(11.24, 1.39, 6.0)
							(12.21, 5.00, 6.0)

Parametric Equation of Ellips

$$x_n = x_c + A\cos u$$

$$y_n = y_c + B\sin u \qquad 0 \le \theta \le 2\pi$$

$$z_n = z_c$$



Now uses the same method as in the Representation of circle to reduce the amount of calculation.

$$x_{n+1} = x_c + A\cos(u + du)$$
$$= x_c + A\cos u \cos du - A\sin u \sin du$$

$$y_{n+1} = y_c + B\sin(u + du)$$
$$= y_c + B\sin u \cos du + B\cos u \sin du$$

$$x_{n+1} = x_c + (x_n - x_c)\cos du - \frac{A}{B}(y_n - y_c)\sin du$$

$$y_{n+1} = y_c + (y_n - y_c)\cos du + \frac{B}{A}(x_n - y_c)\sin du$$

Parametric Equation of an inclined Ellips

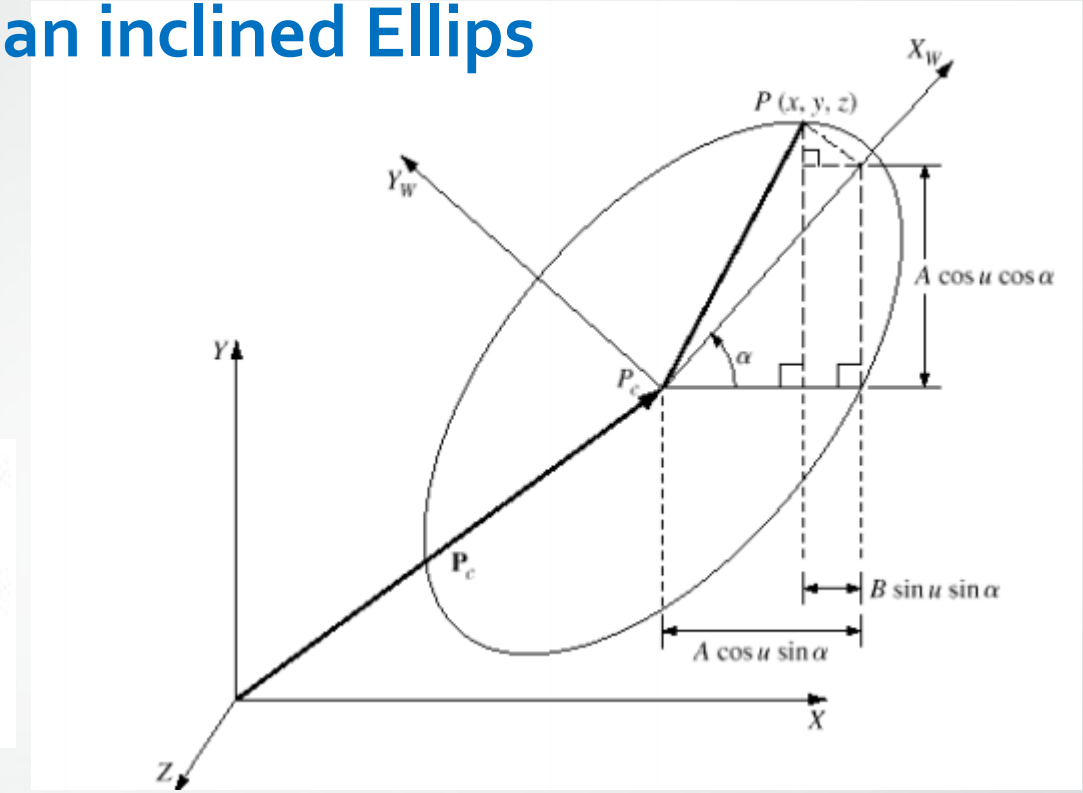
Parametric Equation of ellips inclined at angale α

$$x = x_c + A\cos u\cos\alpha - B\sin u\sin\alpha$$

$$y = y_c + A\cos u\sin\alpha + B\sin u\cos\alpha$$

$$z = z_c$$

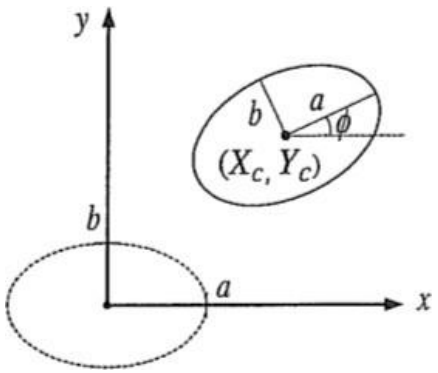
$0 \le u \le 2\pi$



An inclined ellips

Parametric Equation of Inclined Ellips

The ellipse of interest can be obtained by rotating the reference ellipse at the origin by ϕ about the z axis and translating it by X_c in the x direction and by Y_c in the y direction. If the x, y, and z coordinates of the points on the ellipse of interest are denoted x^*, y^*, and z^* and those of the reference ellipse are denoted x, y, and z, the following equation will hold:



$$\begin{bmatrix} x^* & y^* & 0 & 1 \end{bmatrix}^T = Trans(X_c, Y_c, 0) Rot(z, \phi) \begin{pmatrix} x & y & 0 & 1 \end{pmatrix}^T$$

$$= (x\cos\phi - y\sin\phi + X_c \;\; x\sin\phi + y\cos\phi + Y_c \;\; 0 \;\; 1)$$

$$x^* = x\cos\phi - y\sin\phi + X_c = a\cos\theta\cos\phi - b\sin\theta\sin\phi + X_c$$
$$y^* = x\sin\phi + y\cos\phi + Y_c = a\cos\theta\sin\phi + b\sin\theta\cos\phi + Y_c$$
$$z^* = 0 \qquad (0 \le \theta \le 2\pi)$$

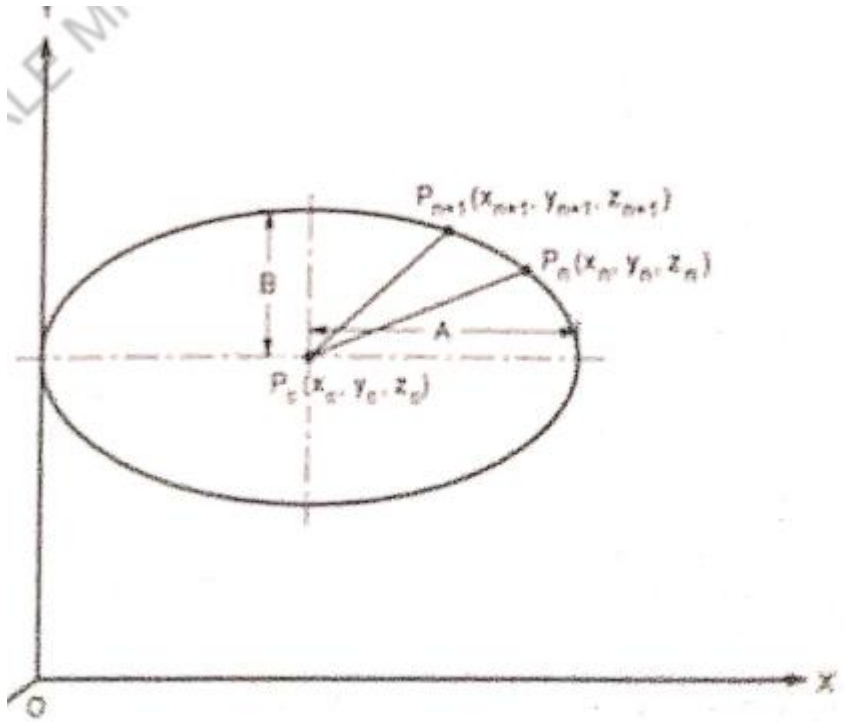
Example 1

An ellipse has major axis of 10 units & minor axis of 8 units. If the centre of ellipse is (5,6,3) write the parametric equation of an ellipse.

Given:

A= 10/2 = 5 units ; B=8/2 = 4 units

$P_c(x_c, y_c, z_c) \equiv (5, 6, 3)$.



1. Coordinates of any point on ellipse :

$$\left.\begin{aligned} x_n &= x_c + A\cos u \\ y_n &= y_c + B\sin u \\ z_n &= z_c \end{aligned}\right\} \quad \ldots(a)$$

At $u = 0$;

$$\left.\begin{aligned} x_n &= 5 + 5\cos 0 = 10 \\ y_n &= 6 + 4\sin 0 = 6 \\ z_n &= z_c = 3 \end{aligned}\right\} \quad \ldots(b)$$

2. Parametric equation of ellipse :

$$x_{n+1} = x_c + (x_n - x_c)\cos\Delta u - \frac{A}{B}(y_n - y_c)\sin\Delta u$$

$$y_{n+1} = y_c + (y_n - y_c)\cos\Delta u + \frac{A}{B}(x_n - x_c)\sin\Delta u$$

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

$$x_{n+1} = 5 + (x_n - 5)\cos\Delta u - \frac{5}{4}(y_n - 6)\sin\Delta u$$

$$y_{n+1} = 6 + (y_n - 6)\cos\Delta u + \frac{5}{4}(x_n - 5)\sin\Delta u$$

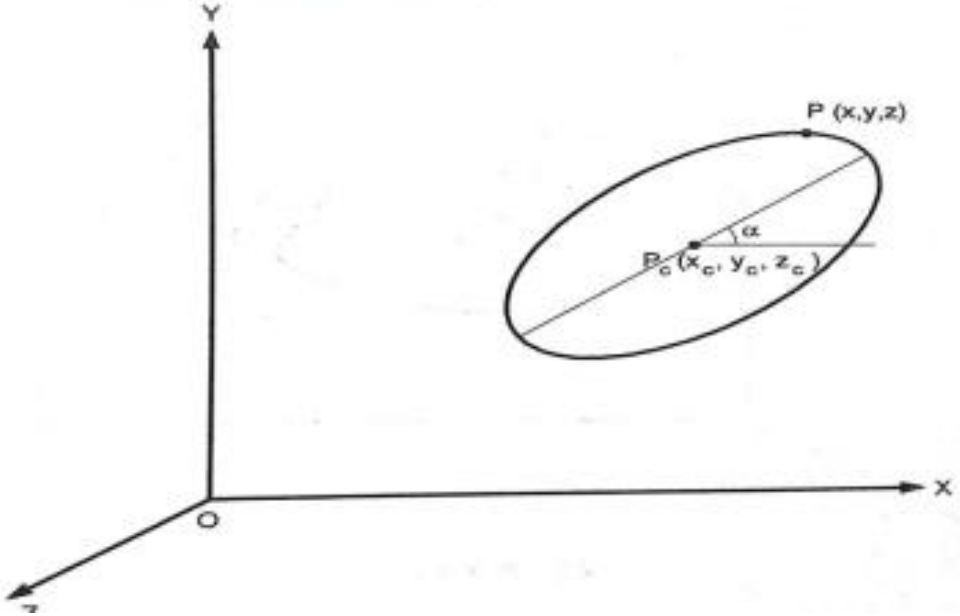
$$z_{n+1} = 3$$ ...Ans.

The ellipse has its major axis 12 units and minor axis 8 units. The inclination of major axis with X-axis is 30°. If the center of ellipse is (10, 5, 4), write its parametric equation.

Solution :

Given : $A = \frac{12}{2} = 6$ units ; $B = \frac{8}{2} = 4$ units

$P_c\ (x_c, y_c, z_c) \equiv (10, 5, 4)$; $\alpha = 30°$.



The parametric equation of ellipse, shown in Fig. can be written as,

$$\left.\begin{aligned} x &= x_c + A\cos u \cdot \cos\alpha - B\sin u \cdot \sin\alpha \\ y &= y_c + A\cos u \cdot \sin\alpha + B\sin u \cdot \cos\alpha \end{aligned}\right\} \quad 0 \le u \le 2\pi \qquad \text{...(a)}$$

$$z = z_c$$

$$\left.\begin{aligned} \therefore\ x &= 10 + 6\cos u \cdot \cos 30 - 4\sin u \cdot \sin 30 \\ y &= 5 + 6\cos u \cdot \sin 30 + 4\sin u \cdot \cos 30 \end{aligned}\right\} \quad 0 \le u \le 2\pi \text{ ...(b)}$$

$$z = 4$$

$$\left.\begin{aligned} \therefore\ x &= 10 + 5.196\cos u - 2\sin u \\ y &= 5 + 3\cos u + 3.464\sin u \\ z &= 4 \end{aligned}\right\} \quad 0 \le u \le 2\pi \qquad \text{...Ans.}$$

Order of a Synthetic Curve

- Mathematically, synthetic curves represent a curve-fitting problem to construct a smooth curve that passes through given data points.

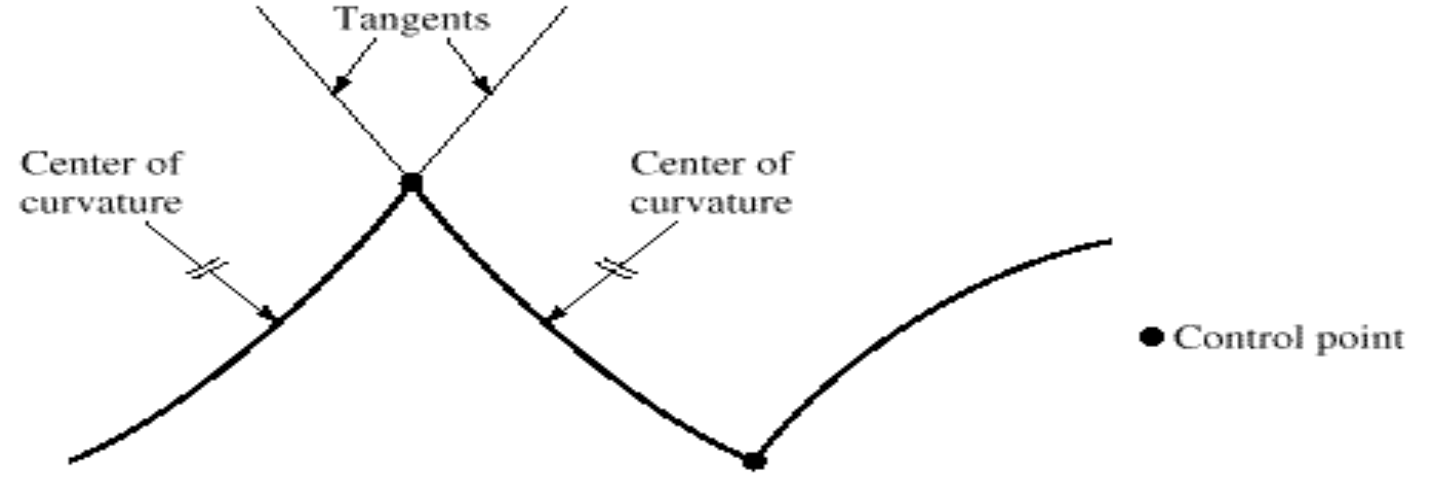
- Zero-order continuity C_0 yields a position continuous curve. First C_1-and second C_2-order continuities imply slope and curvature continuous curves respectively. A C_1 curve is the minimum acceptable curve for engineering design.

- A cubic polynomial is the minimum-order polynomial that can guarantee the generation of C_0, C_1 or C_2 curves.

- Also, the designer may prefer to control the shape of the curve locally instead of globally by changing the control points.

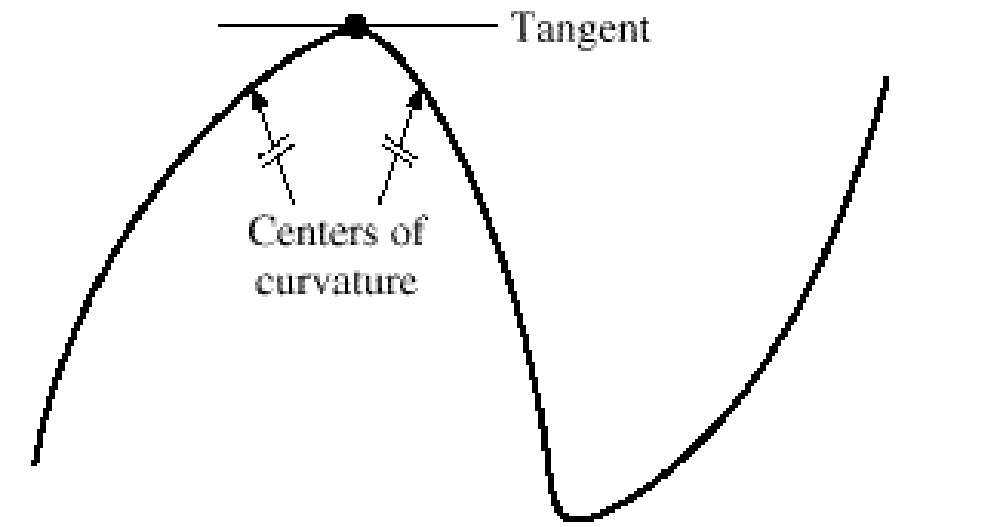
Continuity (C^0, C^1 & C^2)

C^0 continuity: Two curves sections must have the same coordinate position at the boundary point. i.e. when two curves meet at their end points but may not share same tangency curvature.



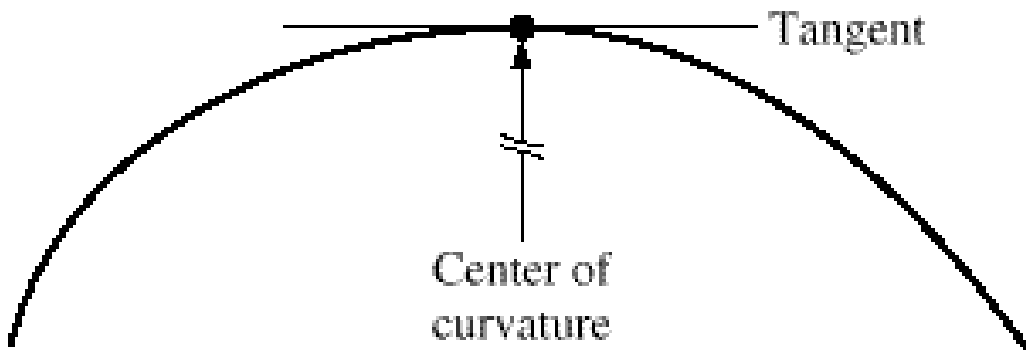
(*a*) Zero-order continuity (C^0 curve)

C^1continuity: It means that the parametric first derivatives are the same at the intersection on two successive sections. i.e. two curves have the same tangent at the matched end points.



(*b*) First-order continuity (C^1 curve)

C^2 continuity: Both the first and second parametric derivatives of the two curve sections are the same at their boundary. i.e. two curves have the same curvature at the end point and have a smooth transition from one curve to another.



(*c*) Second-order continuity (C^2 curve)

Synthetic Curve

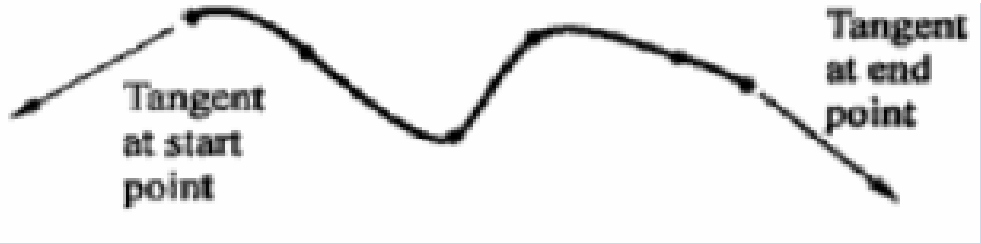
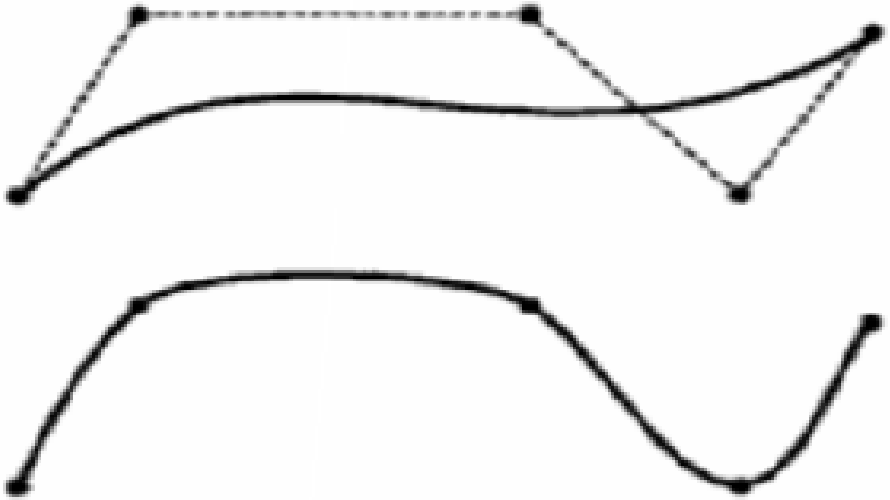
A curve which are defined by set of data points are known as synthetic curve

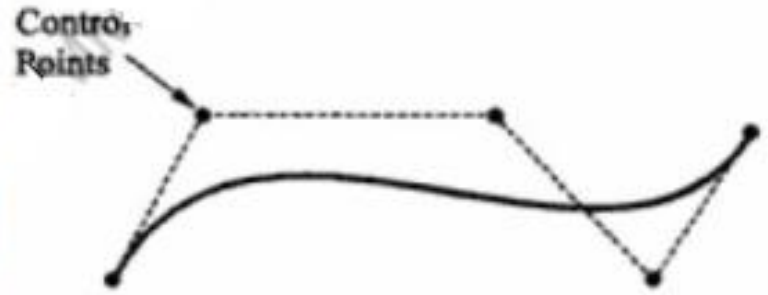
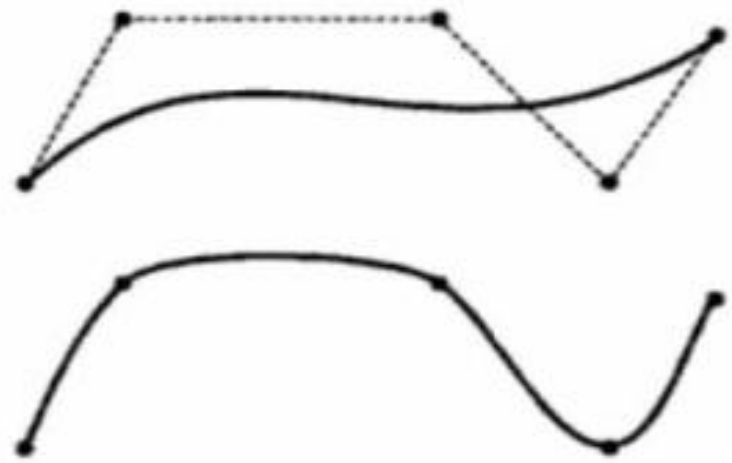
It is used for representation of profile of

- Car bodies
- Ship hulls
- Airplane wings
- Propeller blades

Few types of synthetic curve are

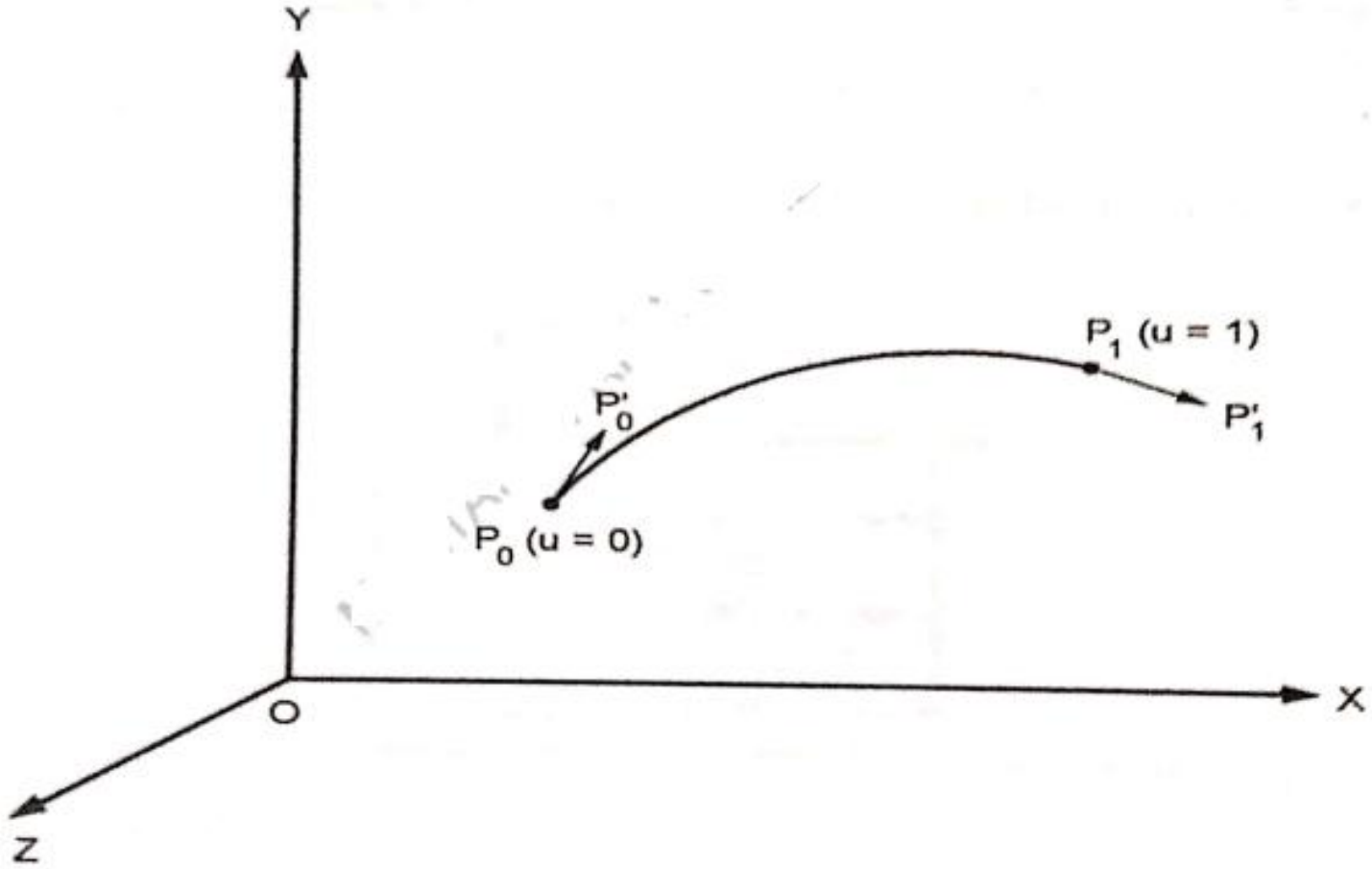
- Hermite Cubic Splines
- Bezier Curve
- B-Spline Curve

Type of Synthetic Curve	Representation
Cubic Spline curve: A curve is fitted to a set of data points. The tangents at the start and end point dictate the shape of the curve.	
Bezier Curve: These use approximations to produce a smooth curve, and do not pass through all the data points, but are influenced by the control points. The order of the curve is dependent on the number of control points.	
B-spline Curve: These are generalized form of the Bezier Curves and have the ability b control the curve shape locally as opposed to global control. They a1o provide the possibility to add control points without increasing the degree of the curve. The kinds of B-spline curves can be fined to a set of data points.	

Type of Synthetic Curve	Representation
Cubic Spline curve: A curve is fitted to a set of data points. The tangents at the start and end point dictate the shape of the curve.	
Bezier Curve: These use approximations to produce a smooth curve, and do not pass through all the data points, but are influenced by the control points. The order of the curve is dependent on the number of control points.	
B-spline Curve: These are generalized form of the Bezier Curves and have the ability b control the curve shape locally as opposed to global control. They a1o provide the possibility to add control points without increasing the degree of the curve. The kinds of B-spline curves can be fined to a set of data points.	

Hermit Cubic Splines:

Hermite cubic spline is drawn by using two end points of curve and the two tangent vectors at these points



Hermit Cubic Splines:

Parametric Equation of Hermite cubic spline

$$\left.\begin{aligned} x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y \\ z(u) &= a_z u^3 + b_z u^2 + c_z u + d_z \end{aligned}\right\} \quad 0 \le u \le 1$$

$$\text{or } P(u) = \begin{Bmatrix} x(u) \\ y(u) \\ z(u) \end{Bmatrix} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ z_z & b_z & c_z & d_z \end{bmatrix} \begin{Bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{Bmatrix} \quad 0 \le u \le 1 \ldots$$

Hermit Cubic Splines:

•General form of Parametric Equation of Hermite cubic spline in any of X, Y, Z direction.

$$P(u) = a u^3 + b u^2 + c u + d \quad 0 \le u \le 1$$

$$\text{or } P(u) = [u^3\ u^2\ u^1\ 1] \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \quad 0 \le u \le 1 \qquad \text{...(2.14.3)}$$

•Tangent Vector at any point P(u):

$$P'(u) = 3a u^2 + 2 b u + c$$

$$\text{or } P'(u) = [3 u^3\ 2u\ 1\ 0] \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \qquad \text{...(2.14.4)}$$

Hermit Cubic Splines:

- Point & Tangent Vector at u=0

$$P_0 = [0\,0\,0\,1]\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \qquad \text{...(a)}$$

$$\text{and} \quad P_0' = [0\;0\;1\;0]\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \qquad \text{...(b)}$$

- Point & Tangent Vector at u=1

$$P_1 = [1\,1\,1\,1]\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \qquad \text{...(c)}$$

$$\text{and} \quad P_1' = [3\;2\;1\;0]\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \qquad \text{...(d)}$$

Hermit Cubic Splines:

•Boundary conditions for Hermite Cubic Spline Curve:

Combing Eq. (a) , (b), (c) & (d) the boundary conditions for hermite cubic spline curve in matrix form can be written as,

$$\begin{Bmatrix} P_0 \\ P_1 \\ P'_0 \\ P'_1 \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix}$$

- **Polynomial Coefficients for Hermite Cubic Spline Curve :**

Solving matrix Equation (2.14.5), the polynomial coefficients a, b, c and d are given by,

$$\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} \begin{Bmatrix} P_0 \\ P_1 \\ P'_0 \\ P'_1 \end{Bmatrix}$$

or

$$\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} P_0 \\ P_1 \\ P'_0 \\ P'_1 \end{Bmatrix}$$

Hermit Cubic Splines:

or $$\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} 2(P_0 - P_1) + P_0' + P_1' \\ -3(P_0 - P_1) - 2P_0' - P_1' \\ P_0' \\ P_0 \end{Bmatrix}$$

3. **General parametric equation for Hermitz cubic spline curve :**

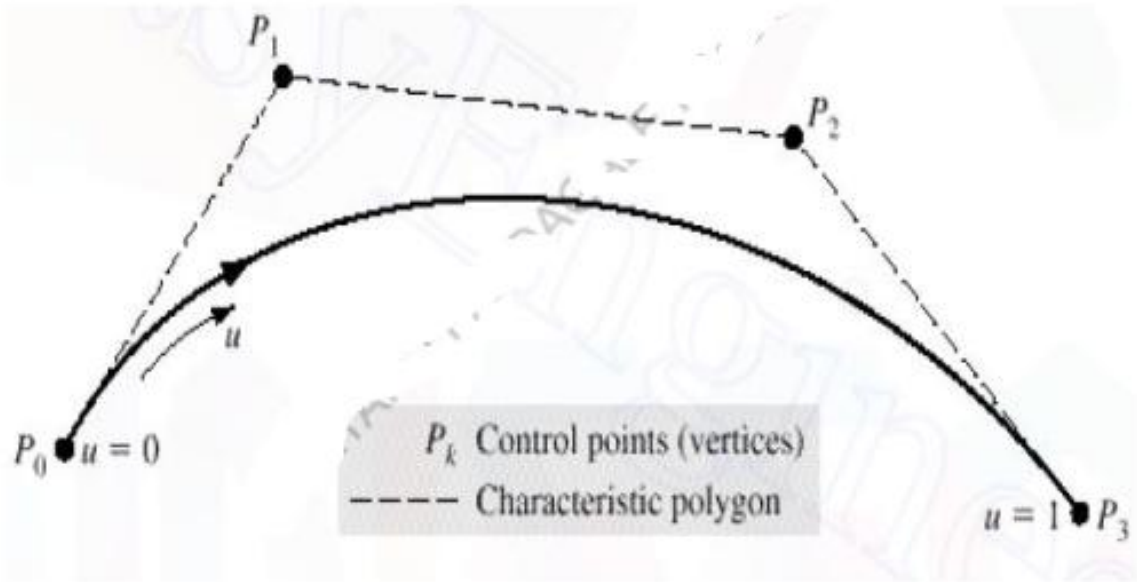
From Equation (2.14.3),

$$P(u) = [u^3\ u^2\ u\ 1] \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \quad 0 \le u \le 1$$

Substituting Equation (2.14.7) in Equation (2.14.8), the parametric equation for **Hermite cubic spline curve** can be written as,

2. Bezier Curves:

- It is not very convenient to change the shape of Hermite cubic splines curves.
- In order to provide more flexibility for changing the shape of curve.
- It is used for car surface design.



- The Curve passes through the first & last points while all other data points act as control points.

2. Bezier Curves:

- Parametric Equation for Bezier Curve:
- The parametric equation for a Bezier curve, with (n+1) data points, passing through the points Po & Pn can be written as,

$$\left.\begin{aligned} P(u) &= \sum_{i=0}^{n} P_i\, C(n,i)\, u^i (1-u)^{n-I} \qquad 0 \le u \le 1 \\ &= \sum_{i=1}^{n} C(n,i)\, P_i\, u^i (1-u)^{n-I} \qquad 0 \le u \le 1 \end{aligned}\right\}$$

$$\begin{aligned} \mathbf{P(u)} &= \mathbf{P_0 (1-u)^n + C(n,1)\, P_1\, u\, (1-u)^{n-1}} \\ &\quad \mathbf{+ C(n,2)\, P_2\, u^2 (1-u)^{n-2}} \\ &\quad \mathbf{+ \ldots + C(n, n-1)\, P_{n-1} u^{n-1} (1-u)} \\ &\quad \mathbf{+ P_n u^n, \quad 0 \le u \le 1} \qquad \ldots \end{aligned}$$

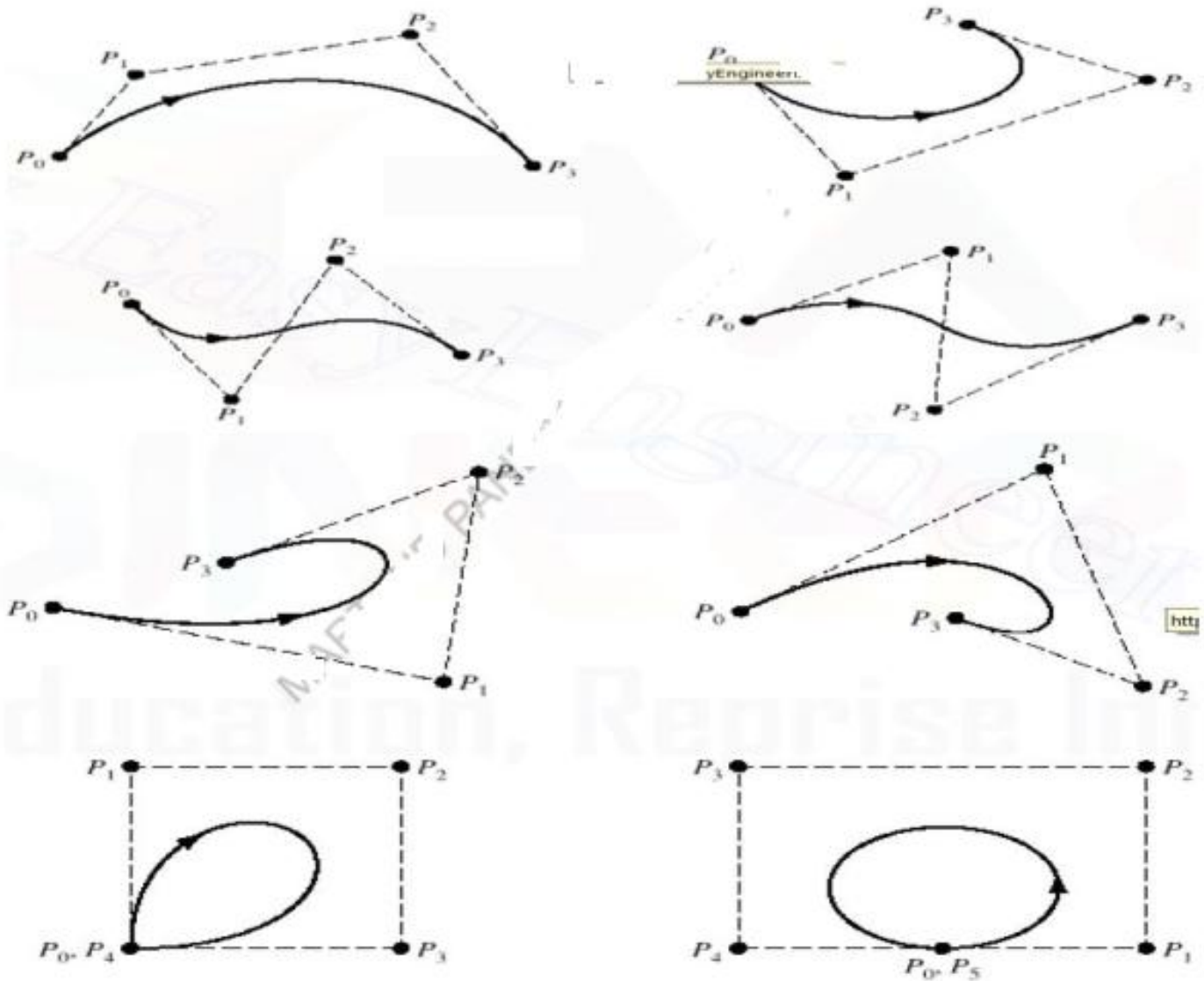
$$\mathbf{C(n,i) = \frac{n!}{i!\,(n-i)!}}$$

2. Bezier Curves:

- Characteristics of Bezier Curves:

1. It does not use tangent vectors for controlling its shape.
2. The shape of it is controlled by number of data points.
3. It gives better flexibility in controlling the shape of the curve.
4. The Bezier curve with (n+1) data points is defined by the polynomial of nth degree.
5. The Bezier curve is symmetric with respect to 'u' & '(1-u)' means the sequence of control points can be reversed without changing the shape of the curve.
6. Each control point is most influential on the curve shape.
7. The closed curve can be generated by closing its characteristics polygon i.e. choosing Po & Pn to be coincident.
8. The flexibility of it increases with increase in number of control points.

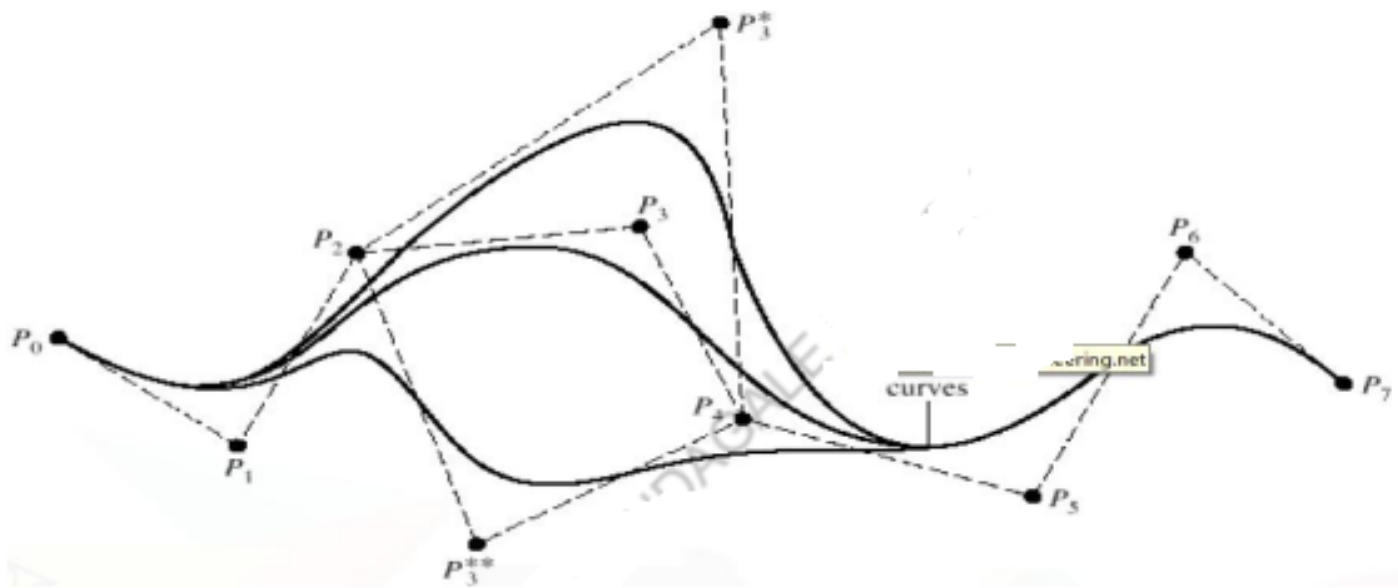
2. Bezier Curves:



B-Spline Curves

- One of the problem associated with the Bezier curve is that, with an increase in the number of control points, the degree of the polynomial representing the curve increases.
- This makes the parametric equation complicated and increases the computation. For that B-spline curve is used.
- B-Spline curve separates the degree of polynomial representing the curve from the number of given control points.
- They can generate a linear, quadratic or cubic B-spline curve.
- B-spline curve are most widely used synthetic curves.

• B- Spline Curves



1. B-spline curve separate the degree of polynomial representing the curve from the number of given control points.

2. While four control points can always generate a cubic Bezier curve, they can generate a linear, Quadratic or Cubic B-spline Curve.

• B- Spline Curves

Advantages of B-Spline Curve:

1. B-Spline curves allows local control over the shape of curve as against the global control in case of the Bezier curve.
2. In B-spline curve, the degree of polynomial representing the curve can be set independently of the number of control points.
3. B-spline Curve give better control.
4. B-spline curve permit to add or delete any number of control points without changing the degree of polynomial.

NURBS
(Non Uniform Rational B-Splines)

- NURBS is a mathematical model commonly used for generating and representing curves and surfaces in computer graphics.
- NURBS provide a single mathematical model capable of representing analytical shapes and synthetic shapes.
- It is superset of all synthetic curves studied so far.
- It is defined by the algebraic ratio of two polynmimals.

❖ The general form of NURBS curve is given by,

$$C(u) = \frac{\sum_{i=1}^{k} N_{i,n} w_i P_i}{\sum_{i=1}^{k} N_{i,n} w_i}$$

➢ **Advantages:**

- NURBS is the most versatile form of shape.
- NURBS is capable of representing all analytical curves as well as synthetic curves.
- Using NURBS, smooth curves can be created or edited with the help of few control points.

NURBS(Non uniform rational B spline curves)

- A rational curve is defined by the algebraic ratio of two polynomials while a non rational curve is defined by one polynomial. Rational curves draw their theories from projective geometry.
- 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 are commonly used in computer-aided design (CAD), manufacturing (CAM), and engineering (CAE) and are part of numerous industry wide standards, such as IGES, STEP, ACIS, and PHIGS. NURBS tools are also found in various 3D modeling and animation software packages.

- NURBS surfaces are functions of two parameters mapping to a surface in three-dimensional space.
- The shape of the surface is determined by control points.
- NURBS surfaces can represent, in a compact form, simple geometrical shapes.
- A NURBS curve is defined by its order, a set of weighted control points, and a knot vector.
- The knot vector is a sequence of parameter values that determines where and how the control points affect the NURBS curve. The number of knots is always equal to the number of control points plus curve degree plus one (i.e. number of control points plus curve order). The knot vector divides the parametric space in the intervals mentioned before, usually referred to as knot spans.
- The values of the knots control the mapping between the input parameter and the corresponding NURBS value. For example, if a NURBS describes a path through space over time, the knots control the time that the function proceeds past the control points.

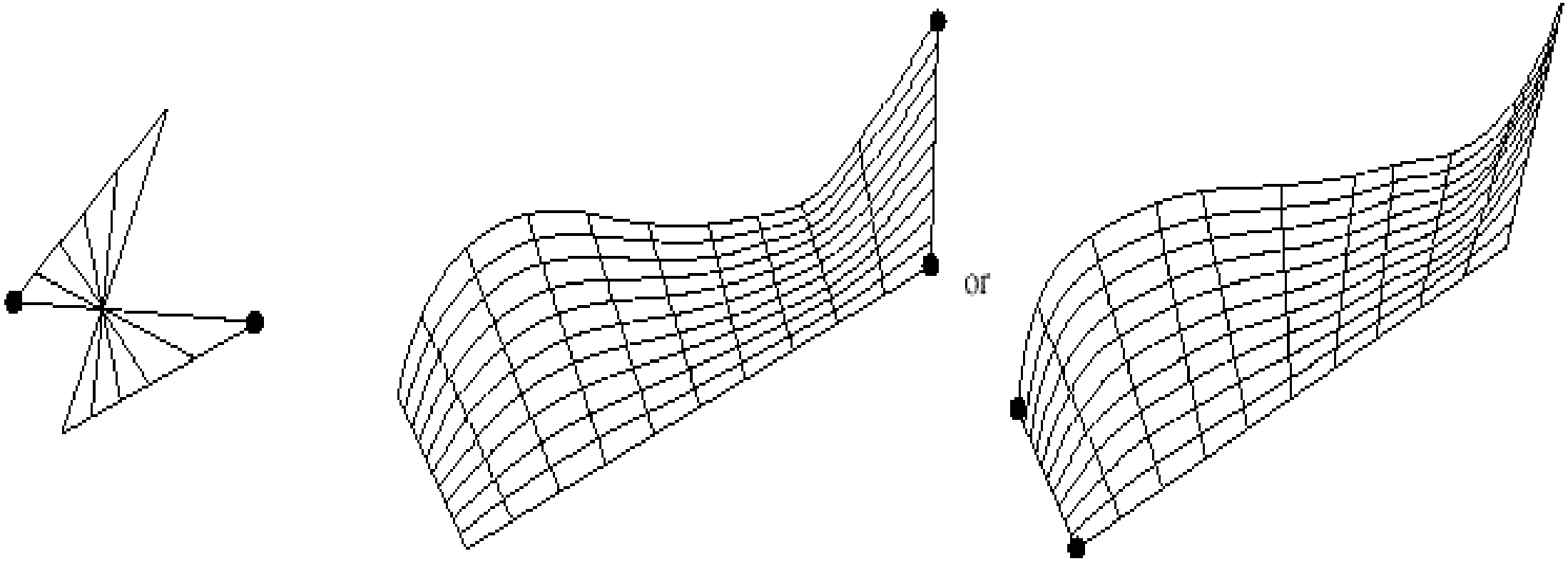
SURFACE MODELING

- Surface modeling gives you the ability to build out a visual representation of an object's exterior and its contours
- Surface modeling is a mathematical method usually provided in CAD applications for displaying solid-appearing objects. Surface modeling makes it possible for users to look at the specific object at specific angles with solid surfaces.
- Surface modeling is a popular technique for architectural designs and renderings.
- For Surface model there were no properties of mass defined and no thickness. This gives the designer the unique ability to modify the model in ways that solid models are incapable.
- Also, note that these surfaces can be represented using NURBS or polygons, depending on the application. SolidWorks typically uses NURBS in its surface modeling operators.
- Surface modeling facilitates, Make Changes to an Imported Model, Design More Complex Shapes and Create a Fillet or Draft Between Faces.

- Surface modeling is considered a more complex technique for displaying objects than wireframe modeling.
- Surface modeling has much less ambiguous display functionalities compared to wireframe modelling. but not as much or sophisticated as solid modeling. The technique often involves conversions between various three-dimensional modeling types.
- Typical processes involved in surface modeling are: Generation of a model combining the three-dimensional surfaces and solids

INTRODUCTION TO SURFACES

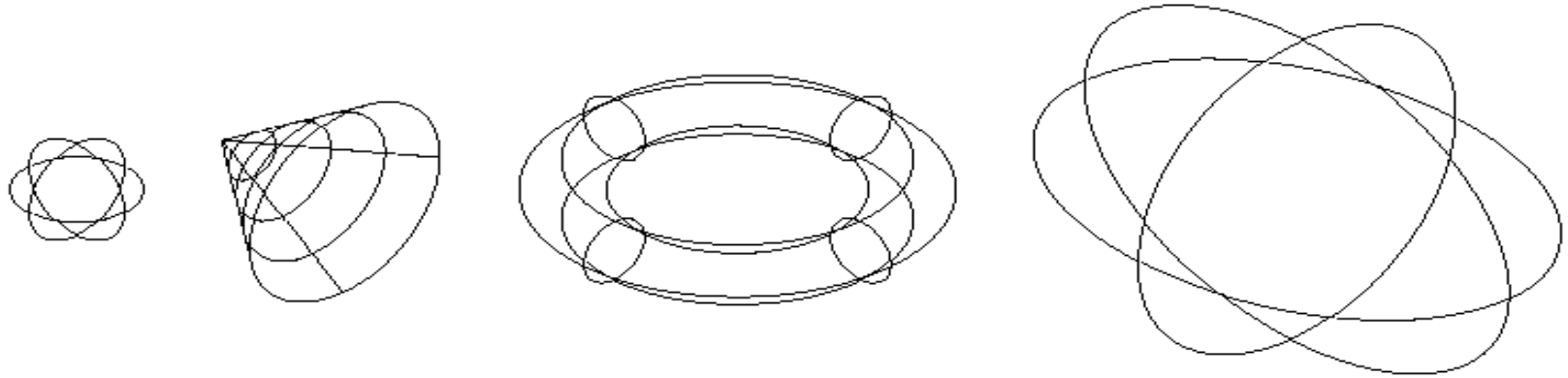
- Shape design and the representation of complex objects such as car, ship, and airplane bodies as well as castings cannot be achieved utilizing the curves covered in earlier.
- In such cases, surfaces must be utilized to describe objects precisely and accurately. We create surfaces, and then we use them to cut and trim solid features and primitives to obtain the models of the complex objects.
- Creation usually begins with data points or curves. Surface creation on CAD/CAM systems usually requires curves as a start.
- During surface creation on a CAD/CAM system, you should follow the modeling guidelines and strategies. Moreover, you should be careful when selecting curves to create surfaces. Selecting the mismatching ends of curves results in twisted surfaces. The figure shows how the wrong ruled surface is created if its defining curves are selected near the wrong ends.



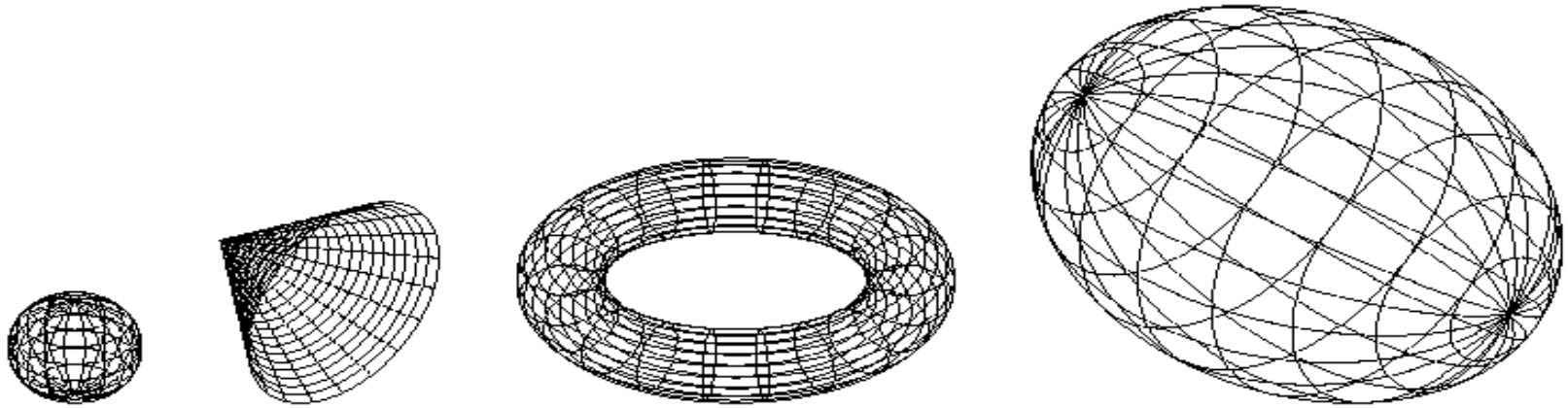
(*a*) Wrong selected end points

(*b*) Correct selected end points

- A surface might require two boundary curves, All curves covered in this Chapter can be used to generate surfaces. In order to visualize surfaces on computer screen, a mesh, say in n in size, is usually displayed. The mesh size is controllable by the user.



(*a*) 4 × 4 mesh size

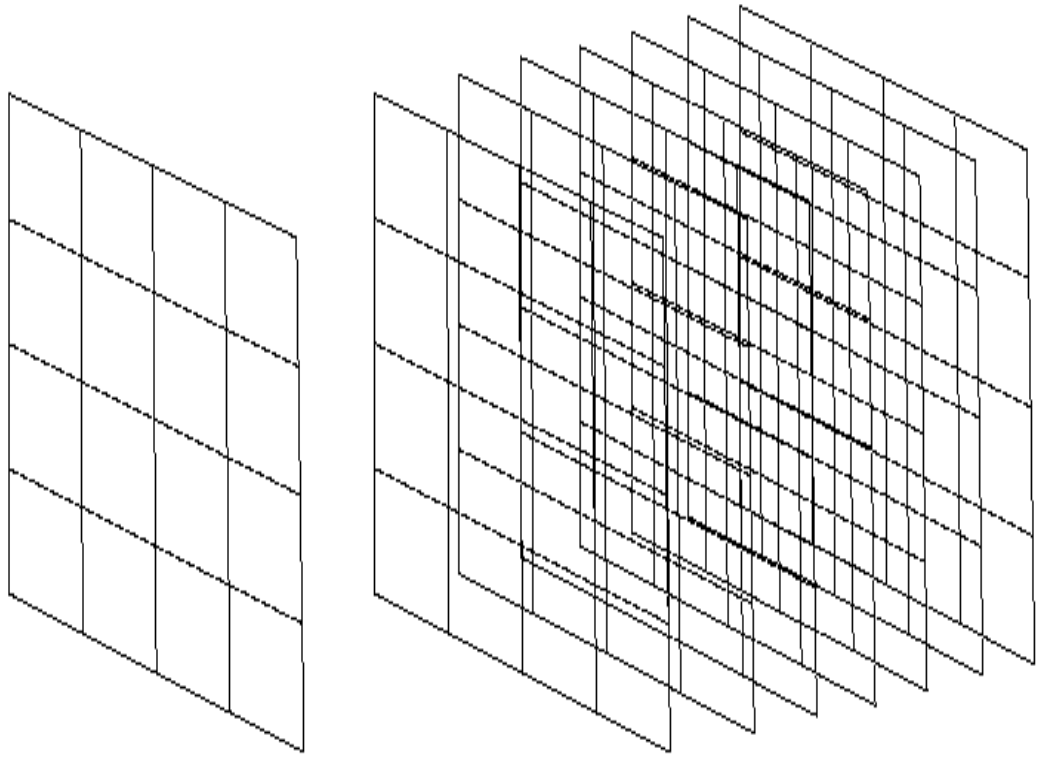


(*b*) 20 × 20 mesh size

Types of Surfaces

1. Plane surface: It is the simplest surface. It requires three non-coincident points to define an infinite plane. The plane surface can be used to generate cross sections by intersecting a solid with it.
2. Ruled (lofted) surface: It is a linear surface. It interpolates linearly between two boundary curves that define the surface (rails). Rails can be any curves, this surface is ideal for representing surfaces that do not have any twists or kinks.
3. Surface of revolution: It is an axisymmetric surface that can model axisymmetric objects. It is generated by rotating a planar curve in space about the axis of symmetry a certain angle as shown in Figure.
4. Tabulated cylinder: It is a surface generated by translating a planar curve a certain distance along a specified direction (axis of the cylinder or directrix) as shown in figure. The plane of the curve is perpendicular to the directrix. This surface is not literally a cylinder. It is used to generate extruded surfaces that have identical cross sections.
5. Bezier surface: it is a surface that approximates or interpolates given input data. It is different from the previous surfaces in that it is a synthetic surface. It extends the Bezier curve to surfaces. It is a general surface that permits twists, and kinks. Bezier surface allows only global control of the surface.

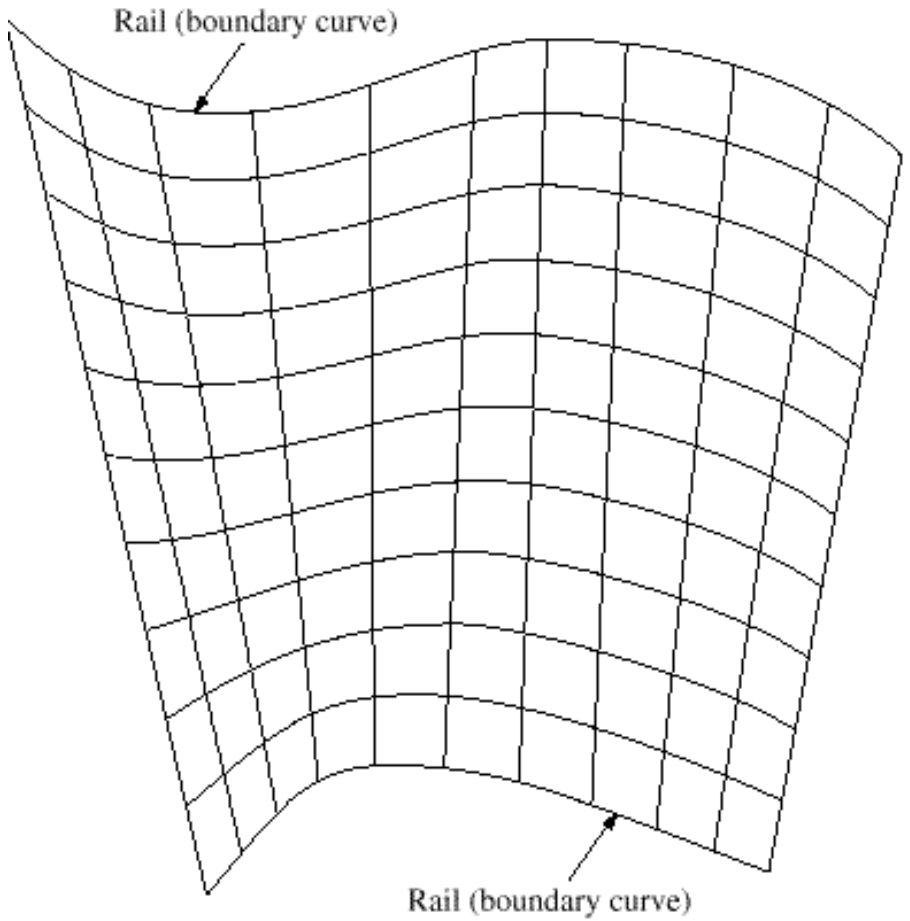
examples.



(*a*) One plane

(*b*) Multiple planes

Figure 7.3 Plane surface.



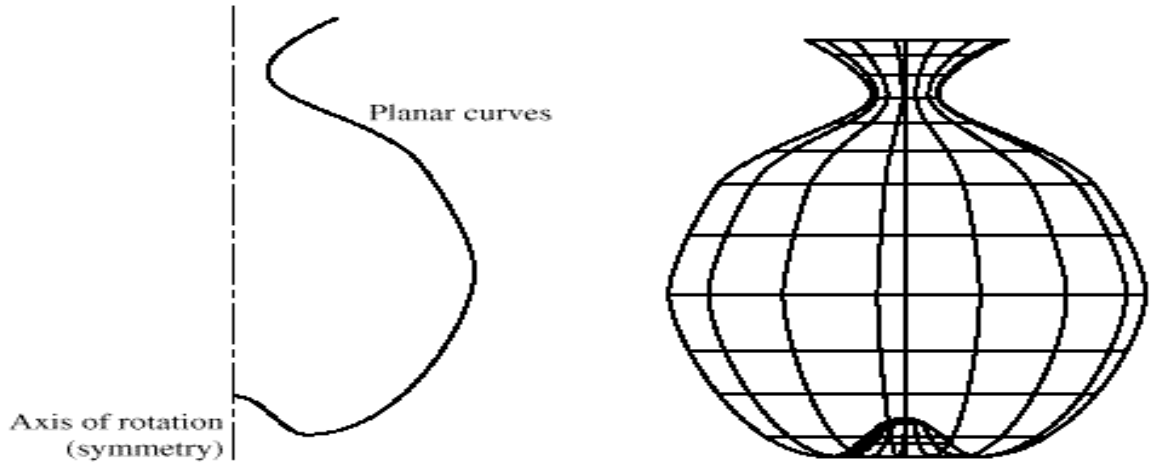


Figure 7.5 Surface of revolution.

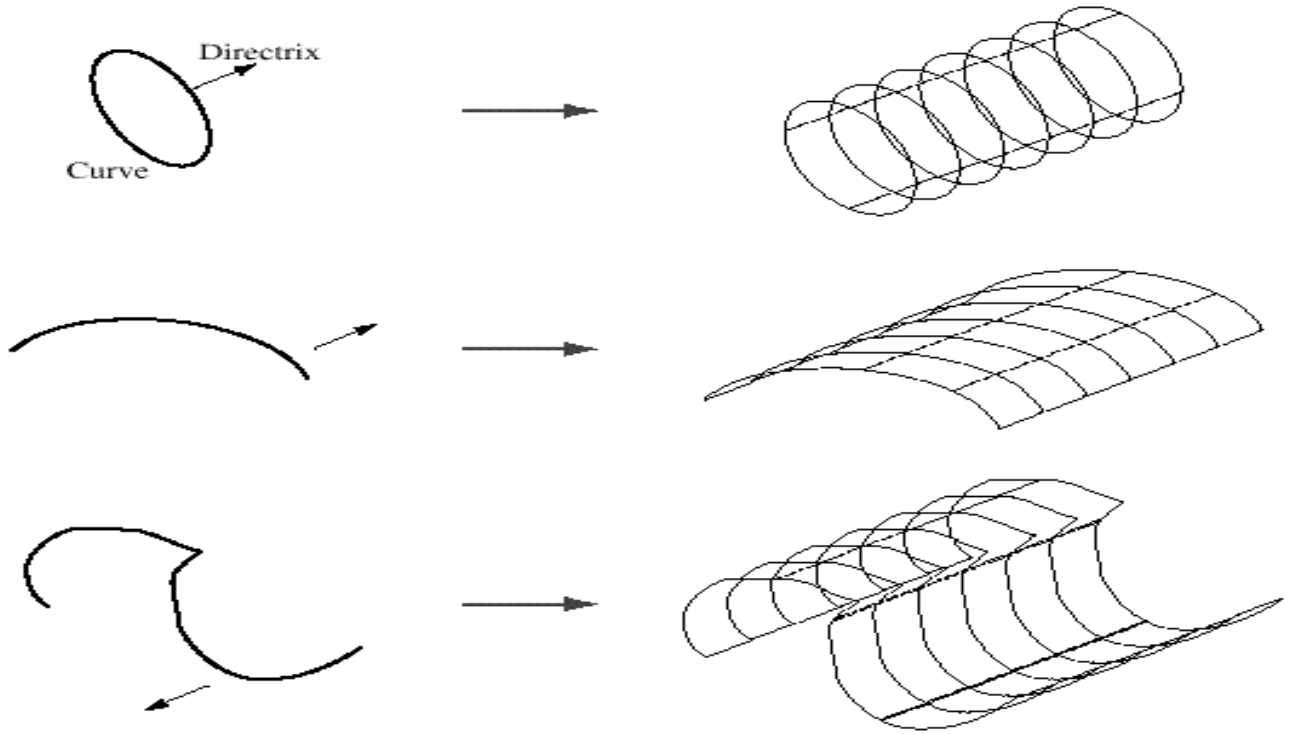


Figure 7.6 Tabulated surface.

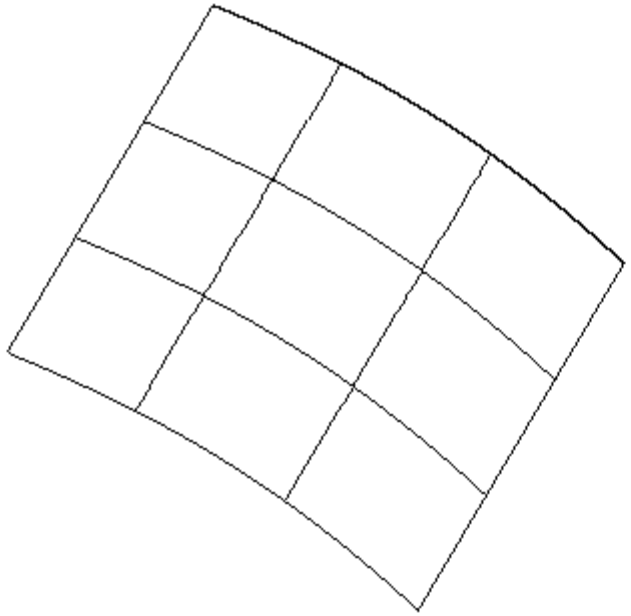
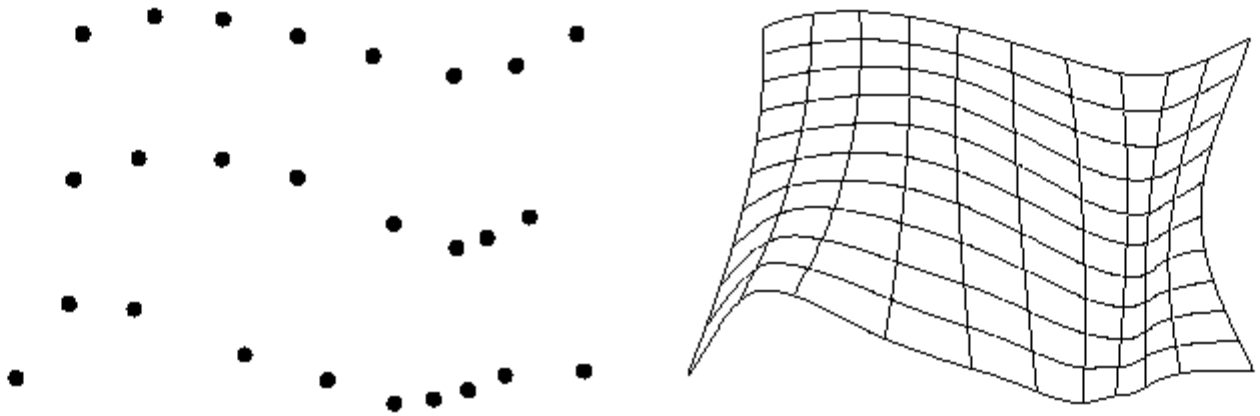


Figure 7.7 Bezier surface.



(*a*) Data points (*b*) B-spline surface

Figure 7.8 B-spline surface.

6. B-spline surface: It is a surface that can approximate or interpolate given input data. Figure shows an interpolating example. It is a synthetic surface. It is a general surface like a Bezier surface hut with the advantage of permitting local control of the surface.
7. Coon’s surface: The previously described surfaces are used with either open boundaries or given data points. A Coons patch is used to create a surface using curves that form closed boundaries .
8. Fillet surface: It is a B-spline surface that blends two surfaces. The two original surfaces may or may not be trimmed.
9. Offset surface: Existing surfaces can be offset to create new ones identical in shape hut with different dimensions. It is a useful surface to use to speed up surface creation. For example, to create a hollow cylinder, the outer or inner cylinder can he created using a cylinder command and the other one can he created by an offset command. The offset surface command becomes very efficient to use if the original surface is a composite one.
10. NURB Surface: as stated earlier, surface defined by NURB curves.
11. Coons Surface: A Coons surface or simply Coons, is a type of manifold parameterization used in computer graphics like CAD/CAM to smoothly join other surfaces together, and in computational mechanics applications, particularly in finite element method and boundary element method, to mesh problem domains into elements.

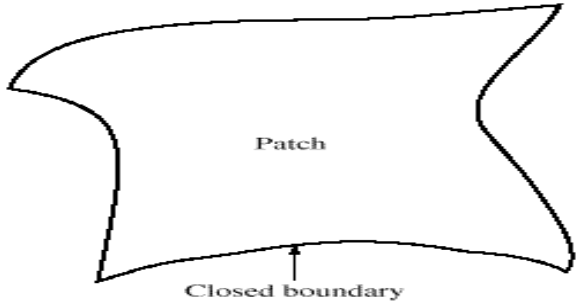


Figure 7.9 Coons surface.

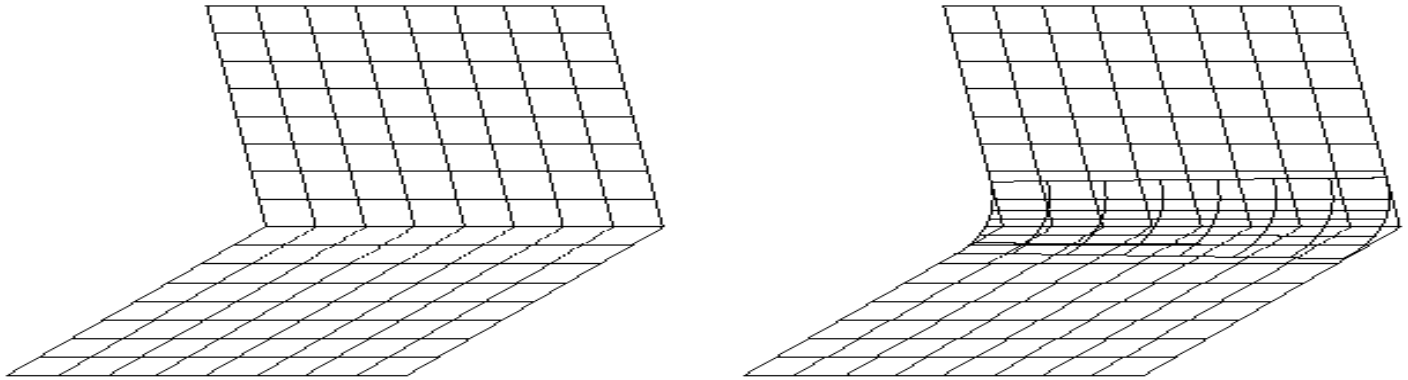


Figure 7.10 Fillet surface.

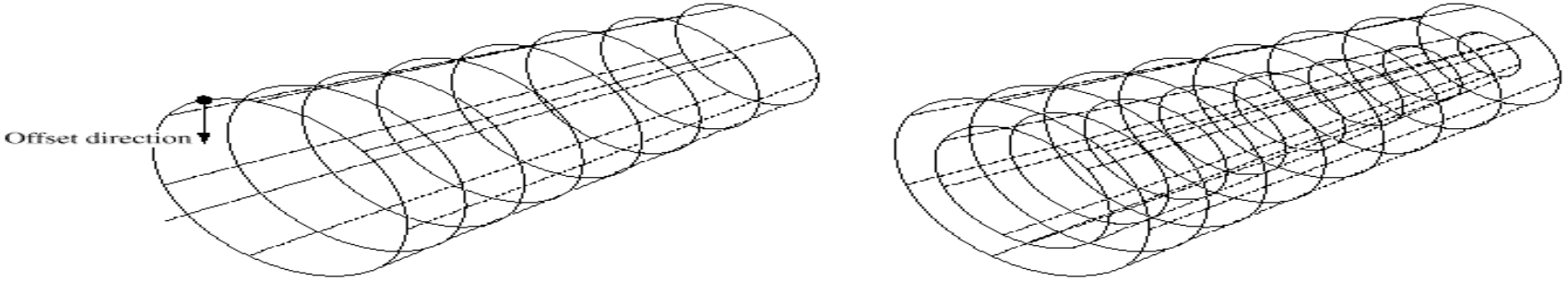
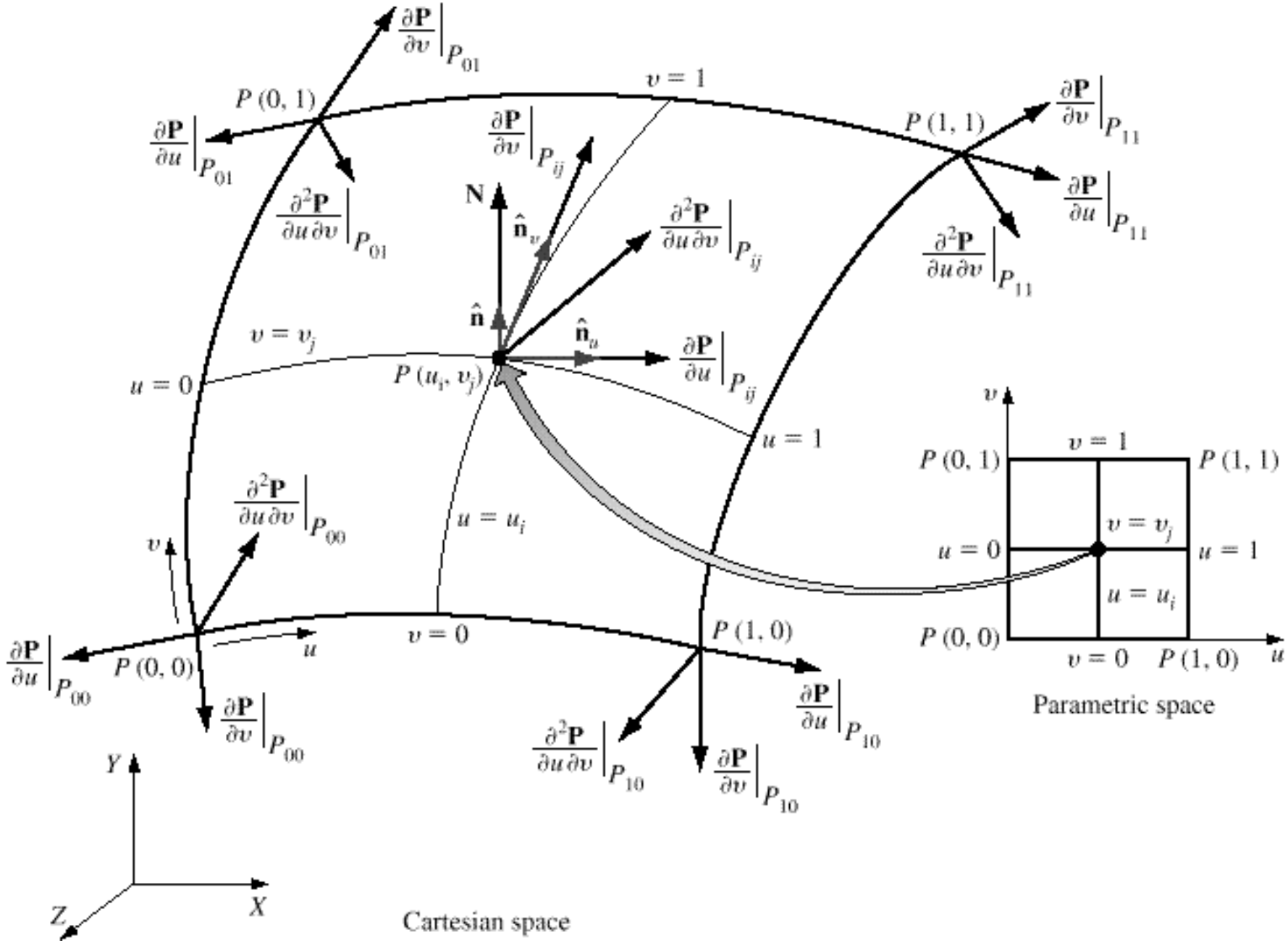


Figure 7.11 Offset surface.

Representation of Curve in Cartesian space :



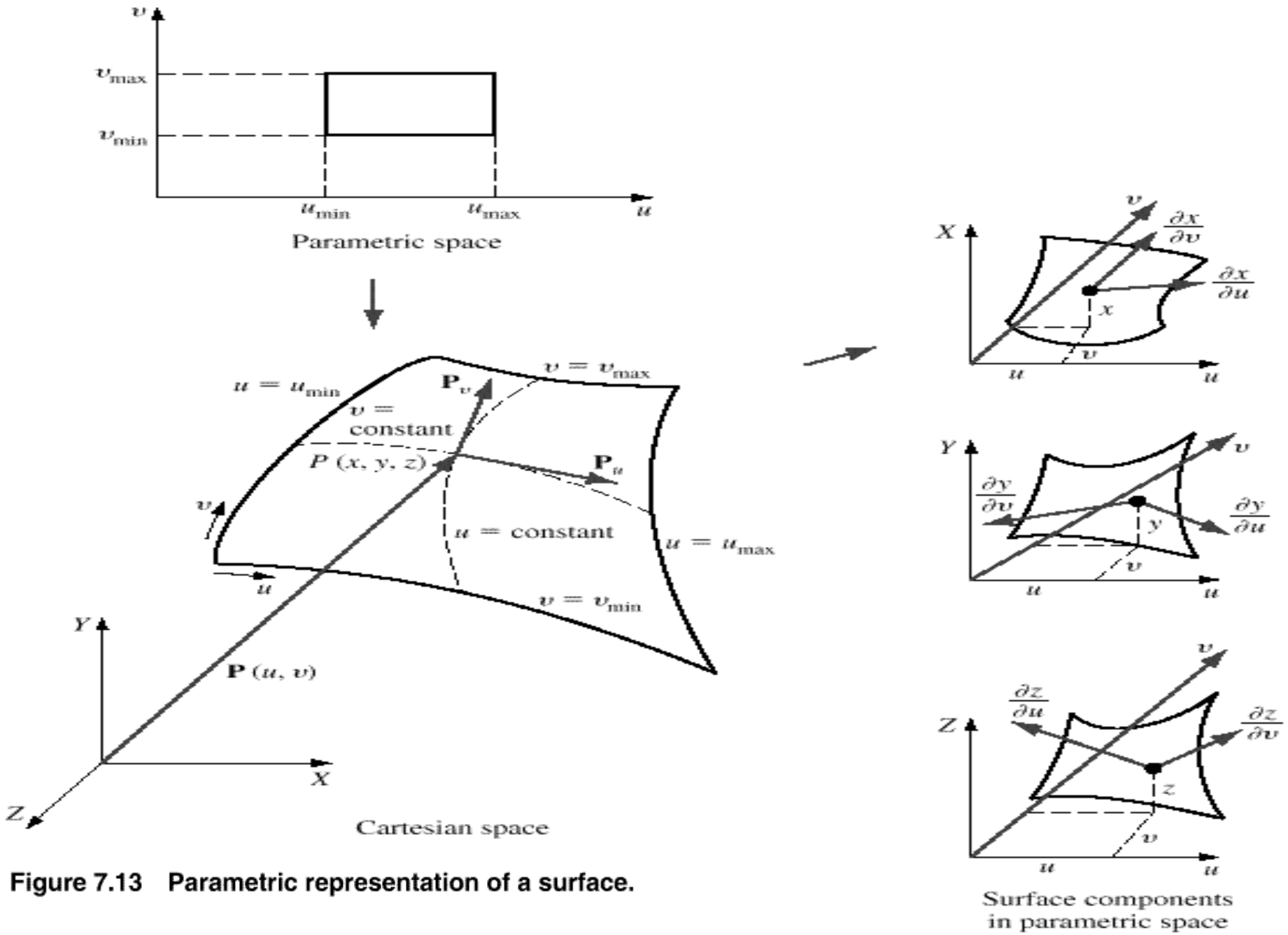


Figure 7.13 Parametric representation of a surface.

Thank You
For Your Attention