

Government College of Engineering and Research
Avasari, Pune

SOLID MODELING AND DRAFTING

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



Unit 4

Geometric Transformation

Unit Outcome

On completion of the course, learner will be able to
APPLY geometric transformations to simple 2D
geometries

Contains

- Introduction, Geometric Transformations, Translation, Scaling, Rotation, Reflection/Mirror, Shear
- Homogeneous Transformation, Inverse Transformation, Concatenated Transformation (limited to 2D objects with maximum 3 points only)
- Coordinate systems - Model (MCS), Working (WCS), Screen (SCS) coordinate system, Mapping of coordinate systems
- Projections of geometric models - Orthographic and Perspective projections, Design and Engineering applications

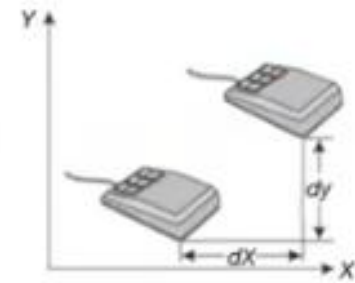
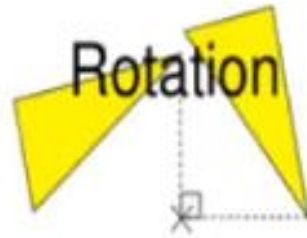
Geometric Transformation

- It can changes the orientation, size, and shape of the objects in the database as well as on the graphics image.
- This transformation actually convert geometry from one coordinate system to other.
 - 1) Translation
 - 2) Rotation
 - 3) Scaling
 - 4) Reflection

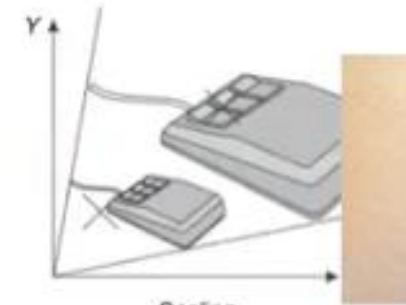
Translation



Rotation



Translation

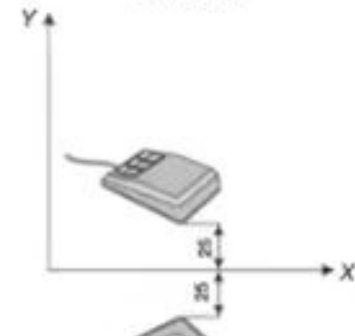
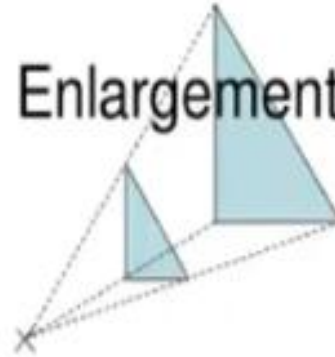


Scaling

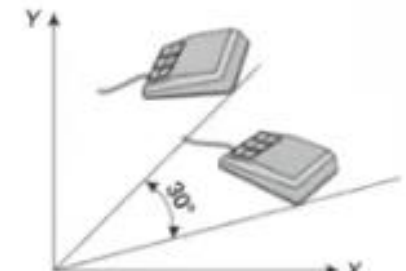
Reflection



Enlargement

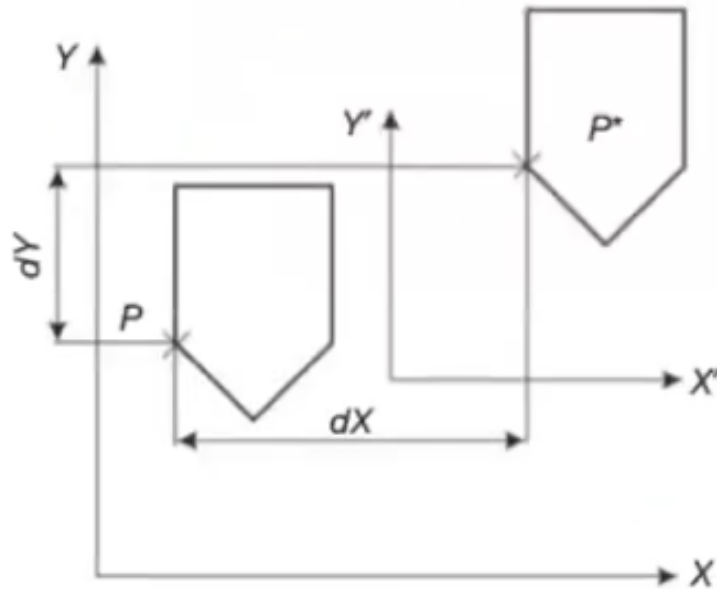


Reflection



Rotation

TRANSLATION



$$P^* = [x^*, y^*]$$

$$x^* = x + dX$$

$$y^* = y + dY$$

Putting Eqs 3.25 and 3.26 back into Eq. 3.24, we can write

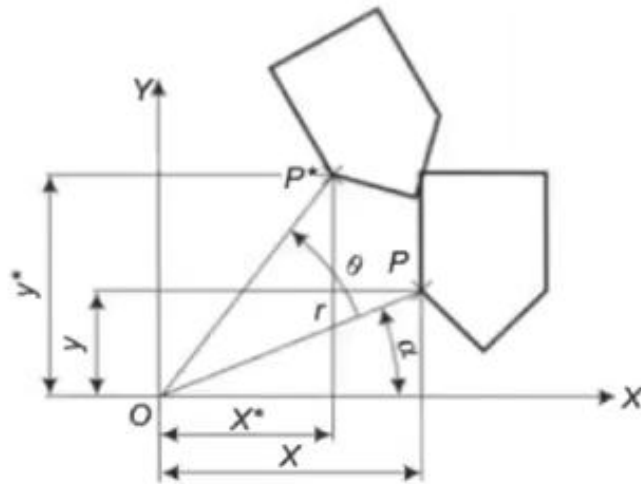
$$[P^*] = \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} x + dX \\ y + dY \end{bmatrix}$$

This can also be written in the matrix form as follows:

$$[P^*] = \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} x + dX \\ y + dY \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dX \\ dY \end{bmatrix}$$

This is normally used as **MOVE** Command in CAD Systems

ROTATIONS



The new position P^* is given by,

$$P^* = [x^*, y^*]$$

The original position is specified by

$$x = r \cos \alpha$$

$$y = r \sin \alpha$$

The new position P^* is specified by

$$\begin{aligned} x^* &= r \cos (\alpha + \theta) \\ &= r \cos \theta \cos \alpha - r \sin \theta \sin \alpha \\ &= x \cos \theta - y \sin \theta \\ y^* &= r \sin (\alpha + \theta) \\ &= r \sin \theta \cos \alpha + r \cos \theta \sin \alpha \\ &= x \sin \theta + y \cos \theta \end{aligned}$$

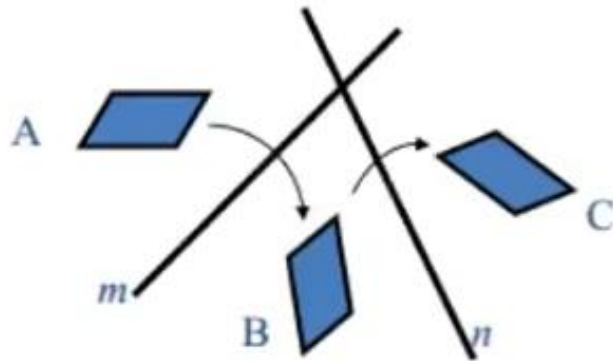
This can be given in the matrix form as

$$\begin{aligned} [P^*] &= \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ [P^*] &= [T_R] \cdot [P] \end{aligned}$$

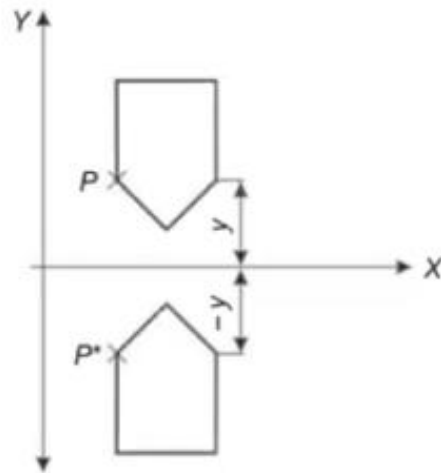
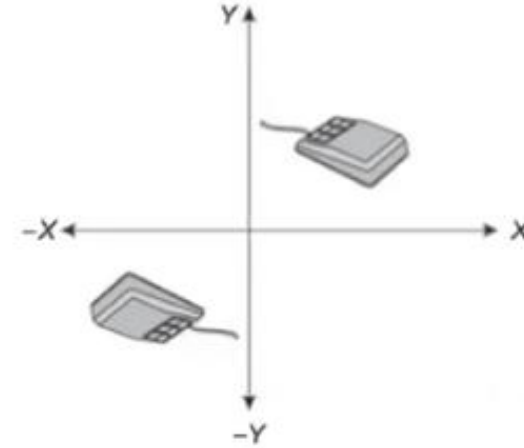
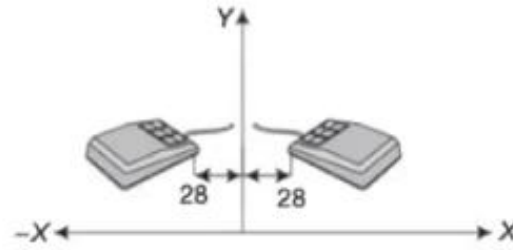
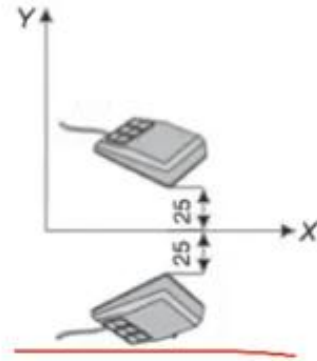
where
$$[T_R] = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

The above is the transformation matrix for rotation, which can be applied in any plane as follows:

$$\begin{aligned} \begin{bmatrix} y^* \\ z^* \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \\ \begin{bmatrix} z^* \\ x^* \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} \end{aligned}$$



REFLECTION (or) MIRROR



$$P^* = [X^*, Y^*] = [X, -Y]$$

This can be given in a matrix form as

$$[P^*] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$[P^*] = [T_m] \cdot [P]$$

where

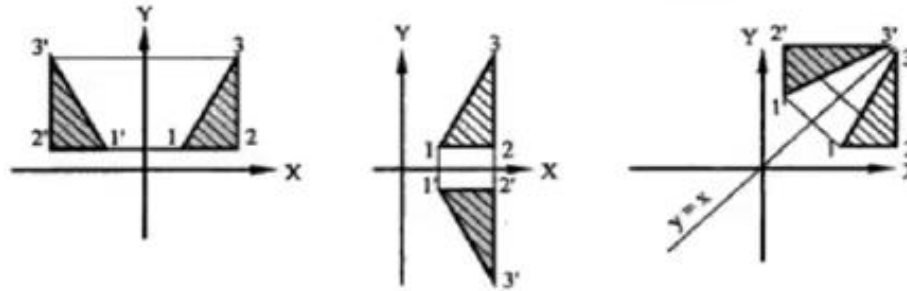
$$[T_m] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Thus, the general transformation matrix will be

$$[M] = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix}$$

Reflection / Mirror

- It is creation of same / mirror image about certain plane or line.
- If 'M' is denoted as reflection matrix then, $[P'] = [P][M]$



$$M = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

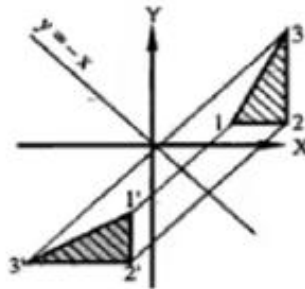
(a) Reflection about Y axis

$$M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

(b) Reflection about X axis

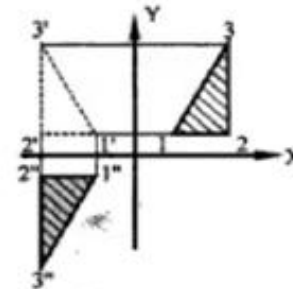
$$M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

(c) Reflection about Line $y = x$



$$M = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

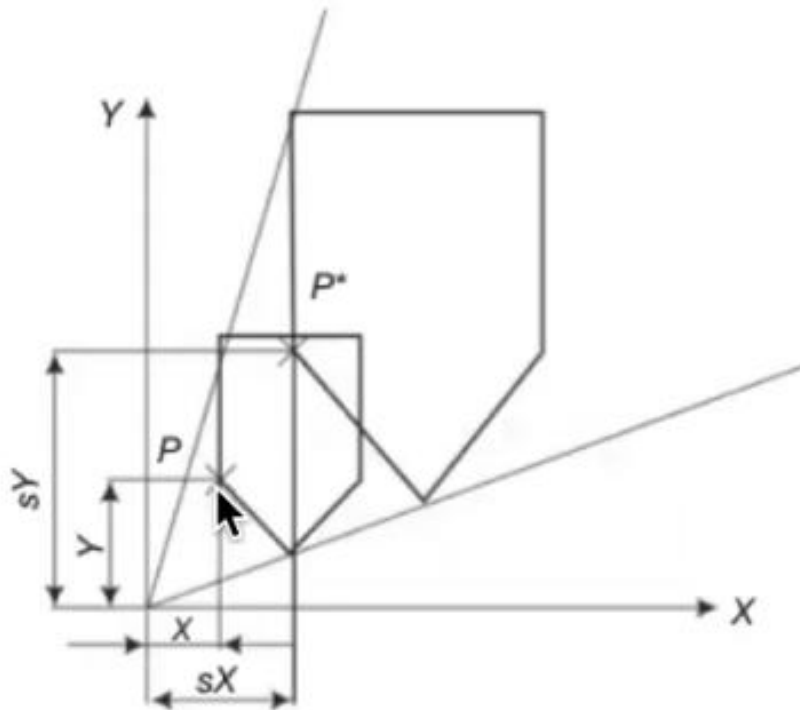
(d) Reflection about Line $y = -x$



$$M = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

(e) Reflection about both X and Y axis

SCALING



$$P^* = [X^*, Y^*] = [S_x \times X, S_y \times Y]$$

This equation can also be represented in the matrix form as follows:

$$[P^*] = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$[P^*] = [T_s] \cdot [P]$$

where

$$[T_s] = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

Shear Transformation

Shear:

A shear transformation distorts the shape of the graphics element or object.

Types of shear:

1. X-direction shear
2. Y- direction shear

Shear Transformations

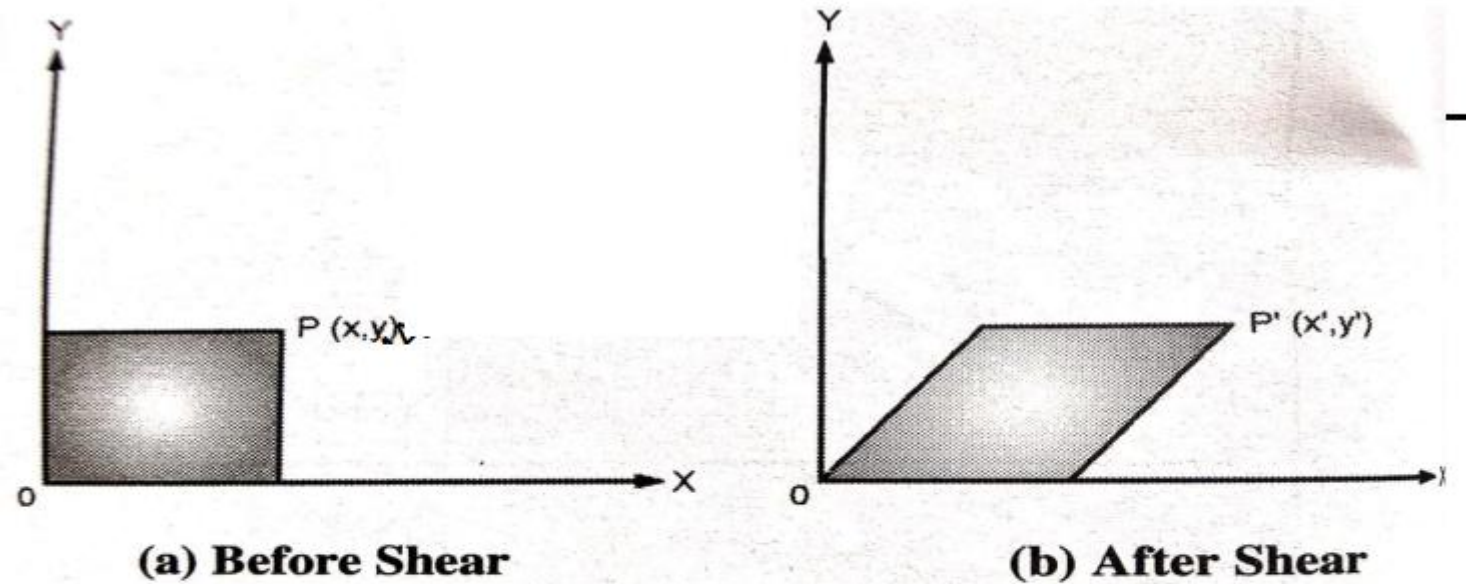
Shear:

A shear transformation distorts the shape of the graphics element or object.

Types of shear:

1. X-direction shear
2. Y- direction shear

Shear In X direction

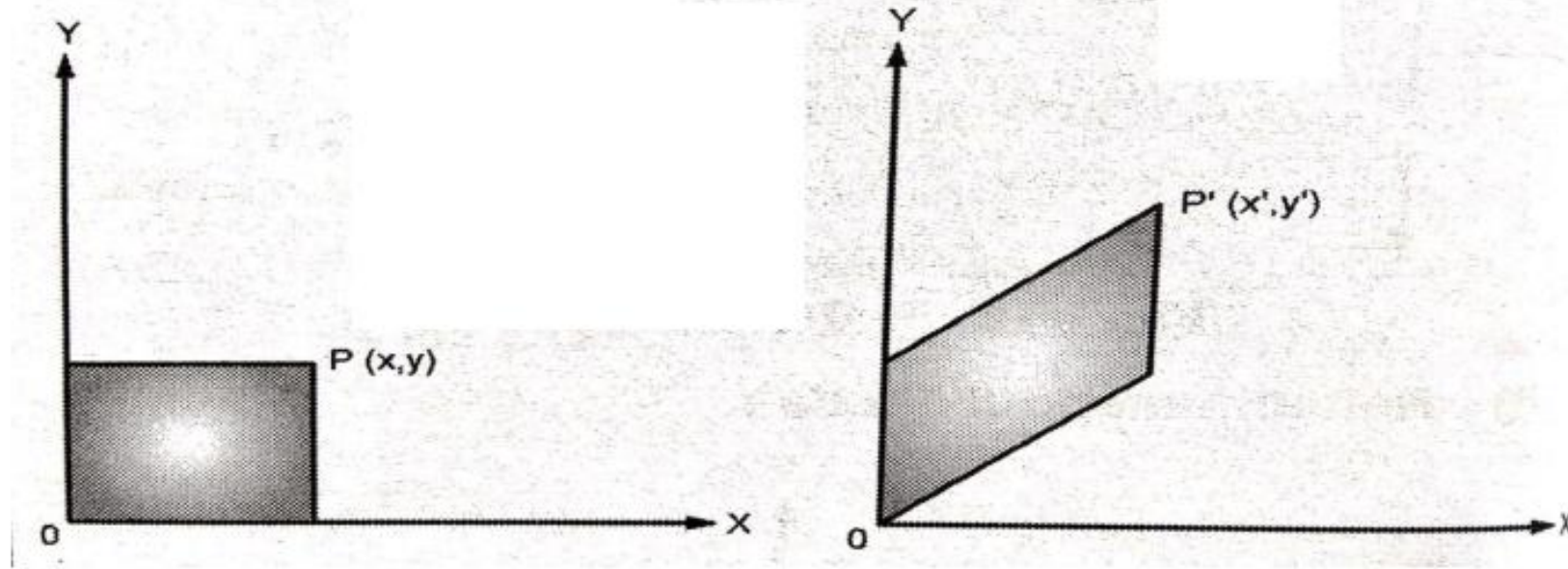


$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} 1 & Sh_x \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix}$$

$$\{ P' \} = [Sh_x] \{ P \}$$

21

(ii) Y-direction shear :



$$\therefore \begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ Sh_y & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix}$$

$$\{ P' \} = [Sh_y] \{ P \}$$

$$= \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$
$$[CT] = \begin{bmatrix} S_x \cdot \cos \theta & -S_x \cdot \sin \theta \\ S_y \cdot \sin \theta & S_y \cdot \cos \theta \end{bmatrix}$$

Translation

$$[P'] = [P] + [T]$$

$$[T] = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Rotation

$$[R] = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

For cw direction θ is (-Ve)
ccw direction θ is (+Ve)

Scaling

$$[S] = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

If $S_x = S_y$ Uniform Scaling

If $S_x \neq S_y$ Differential Scaling

Reflection

1. About X-axis: $[M_x] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

2. About Y-axis: $[M_y] = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

3. About Origin: $[M_o] = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

4. About Line Y=X: $[M_{yx}] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

5. About Line Y= - X: $[M_{y-x}] = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$

Shear

X-direction Shear Matrix $[Sh_x] = \begin{bmatrix} 1 & sh_x \\ 0 & 1 \end{bmatrix}$

Y-direction Shear Matrix $[Sh_y] = \begin{bmatrix} 1 & 0 \\ sh_y & 1 \end{bmatrix}$

Q. 1 A line PQ with end points P(2, 1) and Q (7, 6) is to be moved by
 3 units in X-direction
 and
 4 units in Y-direction
 Determine the coordinates of the end points for a translated line.

Solution:

$$P(2, 1) \quad \begin{matrix} x_1 = 2 \\ y_1 = 1 \end{matrix} \quad Q(7, 6) \quad \begin{matrix} x_2 = 7 \\ y_2 = 6 \end{matrix}$$

$$\begin{matrix} t_x = 3 \\ t_y = 4 \end{matrix} \quad \text{Translation matrix } [T] = \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$[P'] = [P] + [T]$$

$$[Q'] = [Q] + [T]$$

$$\begin{bmatrix} x_1' \\ y_1' \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

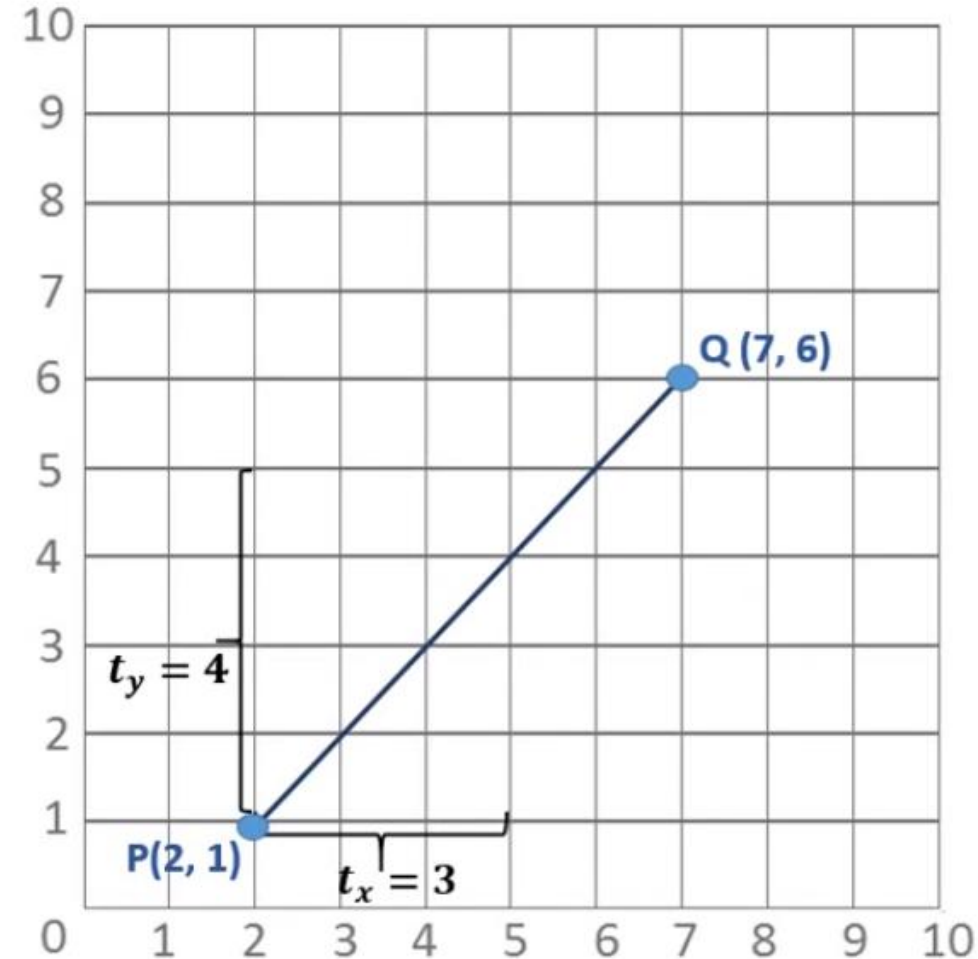
$$\begin{bmatrix} x_2' \\ y_2' \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

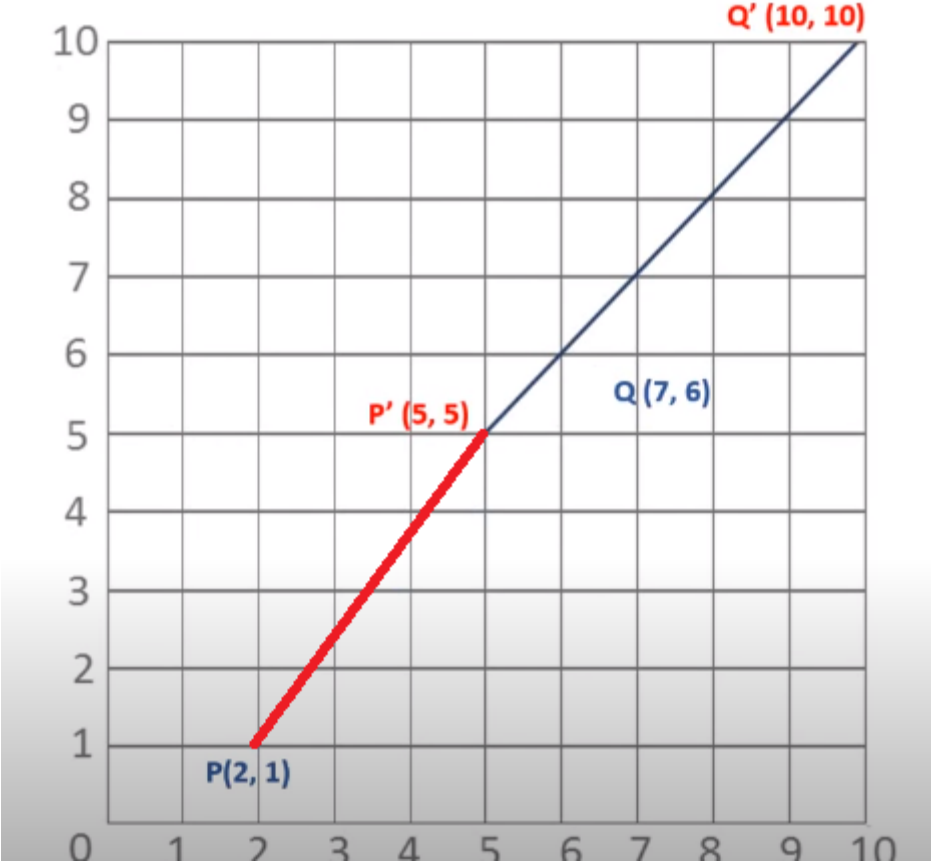
$$\begin{bmatrix} x_1' \\ y_1' \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} x_2' \\ y_2' \end{bmatrix} = \begin{bmatrix} 7 \\ 6 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} x_1' \\ y_1' \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} x_2' \\ y_2' \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$$





Q. 2 A line PQ with end points P(2, 3) and Q (7, 8) is to be rotate about origin by 30° in clockwise direction.

Determine the coordinates of the end points for a rotated line.

Solution:

$$\begin{array}{l} \mathbf{P(2, 3)} \quad x_1 = 2 \\ \quad \quad \quad y_1 = 3 \end{array} \quad \begin{array}{l} \mathbf{Q(7, 8)} \quad x_2 = 7 \\ \quad \quad \quad y_2 = 8 \end{array}$$

$$[R] = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} \cos(-30^\circ) & -\sin(-30^\circ) \\ \sin(-30^\circ) & \cos(-30^\circ) \end{bmatrix}$$

$$[R] = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}$$

$$[P'] = [R] * [P]$$

$$\begin{bmatrix} x_1' \\ y_1' \end{bmatrix} = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\begin{bmatrix} x_1' \\ y_1' \end{bmatrix} = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} x_1' \\ y_1' \end{bmatrix} = \begin{bmatrix} 3.232 \\ 1.598 \end{bmatrix}$$

$$[Q'] = [R] * [Q]$$

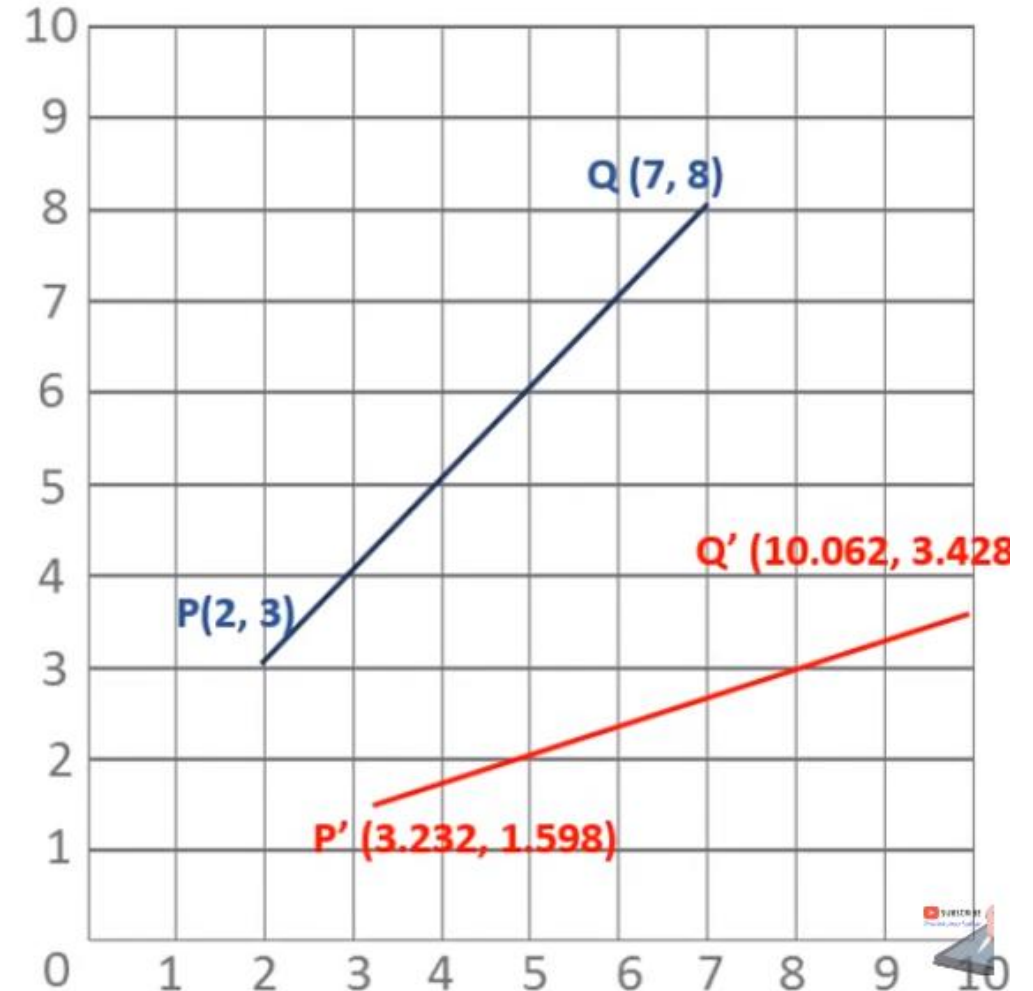
$$\begin{bmatrix} x_2' \\ y_2' \end{bmatrix} = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$\begin{bmatrix} x_2' \\ y_2' \end{bmatrix} = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

$$\begin{bmatrix} x_2' \\ y_2' \end{bmatrix} = \begin{bmatrix} 10.062 \\ 3.428 \end{bmatrix}$$

Θ is 30° clockwise

So $\Theta = -30^\circ$



Composite/ Concatenated Transformation

- Composite Transformation: It is the combination of two or more transformations such as, translation, rotation, scaling and reflection.
- It will improve the performance of graphics package.
- Composite transformation matrix:

Consider Point P (x,y) is reoriented to a new position P'(x',y') by first rotating and then scaling. Then, CT can be carried out as follows,

$$\{ \mathbf{P}' \} = [\mathbf{S}] [\mathbf{R}] \{ \mathbf{P} \}$$

or

$$\{ \mathbf{P}' \} = [\mathbf{CT}] \{ \mathbf{P} \}$$

where,

$$[\mathbf{R}] = \text{rotation matrix}$$
$$[\mathbf{S}] = \text{scaling matrix}$$

Homogenous Transformation

- The various rigid body transformations discussed so far represents translation, rotation, scaling and mirroring and shear are in the form of matrix multiplication.
- but the translation takes the form of vector addition. This makes it inconvenient to concatenate (combine) transformation involving translation.
- It is desirable to express all geometric transformations in the form of matrix multiplications only.
- Representing points by their homogeneous coordinates provides an effective way to unify the description of geometric transformations as matrix multiplications.
- Using homogeneous transformations geometric transformations are customarily embedded into graphic hardware to speed their execution.
- They are also useful for obtaining perspective views of geometric models, in projective geometry, mechanism analysis and design and Robotics for development and formulation.
- They remove many anomalous situations encountered in Cartesian geometry, such as representing points of infinity and non intersections of parallel lines and greatly simplify expressing rational parametric curves and surfaces.

In homogeneous representation, an n-dimensional space is mapped in to an (n+1) dimensional space.

Thus a 2-D point [x y] is represented by 3-D as [x y 1].

$$[P^*] = \begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Hence, the translation matrix in multiplication form can be given as

$$[MT] = \begin{bmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{bmatrix}$$

3. Scaling:

$$[S] = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \text{scaling matrix}$$

Shear In Y direction

In y-direction

$$[Sh_y] = \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \text{Y-direction shear matrix}$$

3D Transformations

- The 2D transformations can be extended to 3D by adding the Z-axis parameter.
- The transformation matrix will be 4×4 .

Translation

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dX \\ 0 & 1 & 0 & dY \\ 0 & 0 & 1 & dZ \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Reflection

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \pm 1 & 0 & 0 & 0 \\ 0 & \pm 1 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation about Z-axis (XY plane)

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation about X-axis (YZ Plane)

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation about Y-axis (ZX Plane)

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotate a triangle with vertices (10,20), (10,10), (20,10) about the origin by 30 degrees CCW and then translate it by 5 and 10 unites in x and y direction respectively.

$$\theta = 30$$

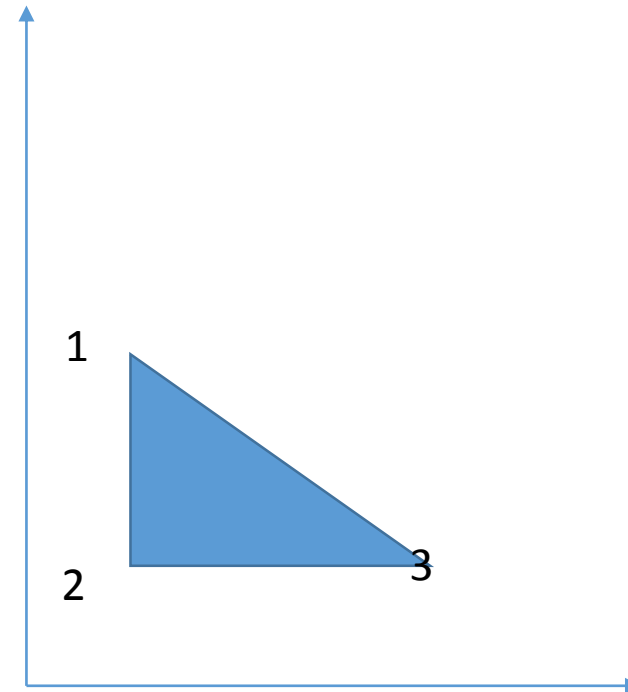
$$R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(30) & -\sin(30) & 0 \\ \sin(30) & \cos(30) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$t_x = 5, t_y = 10$$

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

$$CT = [T][R]$$

$$= \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$CT = [T][R]$$

$$CT = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.866 & -0.5 & 5 \\ 0.5 & 0.866 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

New Vertices of Triangle

$$\{P_1'\} = [CT] \{P_1\}$$

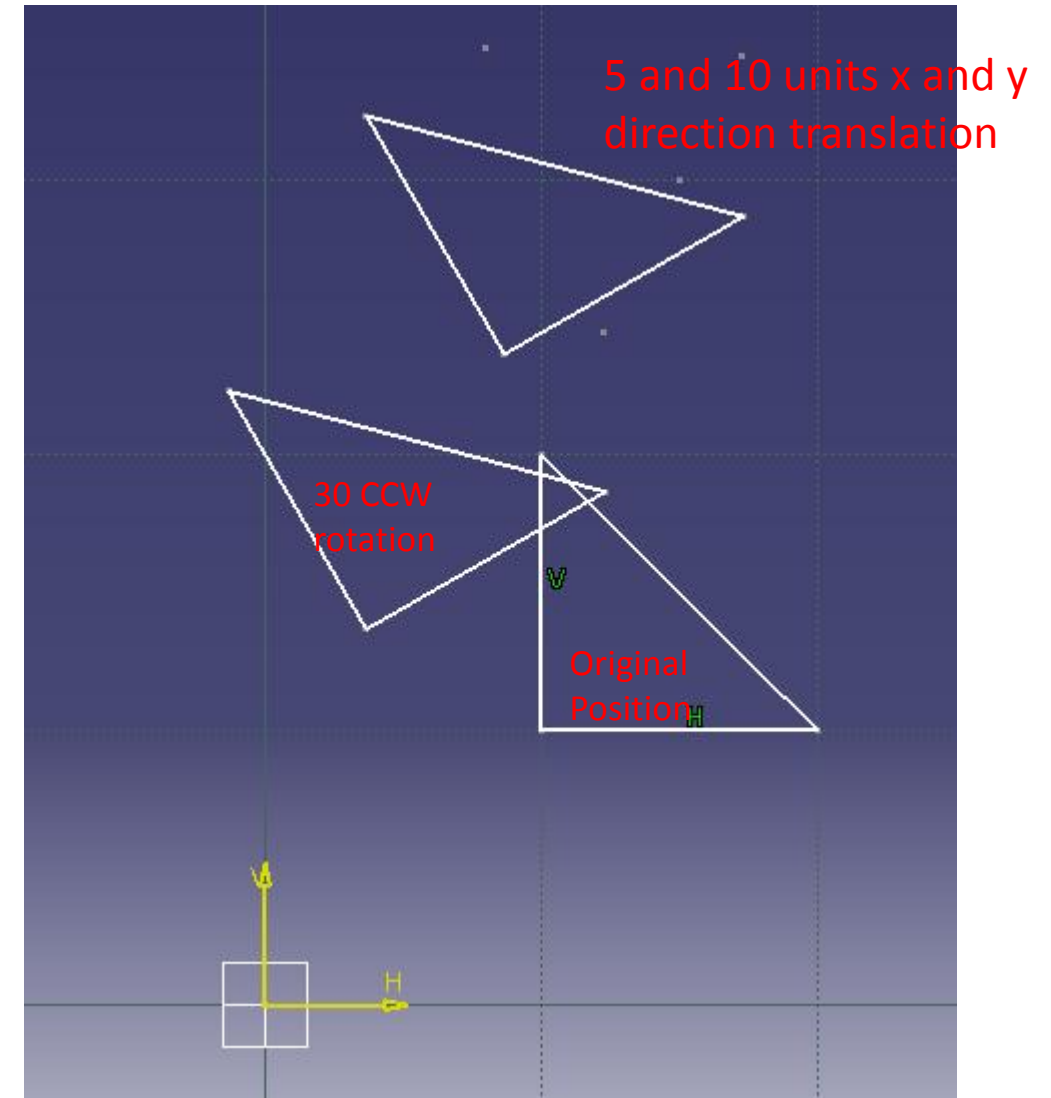
$$\begin{bmatrix} 0.866 & -0.5 & 5 \\ 0.5 & 0.866 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 10 \\ 20 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 3.66 \\ 32.32 \\ 1 \end{Bmatrix}$$

$$\{P_2'\} = [CT] \{P_2\}$$

$$\begin{bmatrix} 0.866 & -0.5 & 5 \\ 0.5 & 0.866 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 10 \\ 10 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 8.66 \\ 23.66 \\ 1 \end{Bmatrix}$$

$$\{P_3'\} = [CT] \{P_3\}$$

$$\begin{bmatrix} 0.866 & -0.5 & 5 \\ 0.5 & 0.866 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 20 \\ 10 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 17.32 \\ 28.66 \\ 1 \end{Bmatrix}$$



Inverse Transformation:

Transformations modify each point of the entity (x, y) to (x', y') . However, in many cases, we need to undo the effect of transformation, i.e. consider a case where a point (x', y') is to be reconverted to the original position (x, y) . Undoing a transformation is itself another transformation. This is determined by matrix inversion. The inverse of a matrix is another matrix such that when the two are multiplied, an identity matrix results. That is :

$$\mathbf{T} \cdot \mathbf{T}^{-1} = \mathbf{I}$$

For a transformed point represented by \mathbf{P}' , the original point represented by \mathbf{P} can be found by considering the following steps.

$$\begin{aligned} & \mathbf{P}' = \mathbf{P} \cdot \mathbf{T} \\ \therefore & \mathbf{P}' \cdot \mathbf{T}^{-1} = \mathbf{P} \cdot \mathbf{T} \cdot \mathbf{T}^{-1} \\ \therefore & \mathbf{P}' \cdot \mathbf{T}^{-1} = \mathbf{P} \cdot \mathbf{I} \\ \therefore & \mathbf{P}' \cdot \mathbf{T}^{-1} = \mathbf{P} \end{aligned}$$

Thus multiplying the inverse transformation matrix to \mathbf{P}' , gives original matrix \mathbf{P} . The inverse matrix of a homogeneous coordinate transformation matrix is :

Inverse translation matrix

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse rotation matrix

$$\mathbf{R}^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse scaling matrix

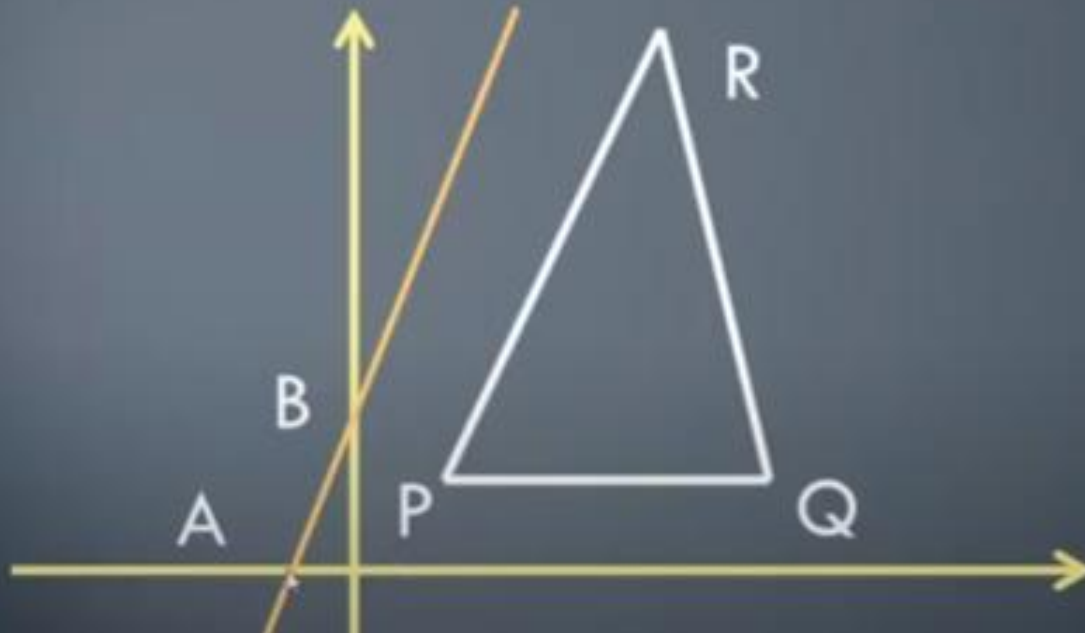
$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse reflection matrix



A triangle PQR with vertices $P(2,2)$, $Q(5,2)$ and $R(4,7)$ is to be reflected about the line $y = 0.5x + 3$. Determine :

- The concatenated transformation matrix; and
- Coordinates of vertices of reflected triangle.



A (-1.5, 0)
B (0, 3)

Solution								
		2		5		4		
	P =	2	Q =	2	R =	7		
		1		1		1		
Orbitory Line AB			-1.5			0		
	A =		0		B =	3		
			1			1		
	Tx =	-1.5	Ty =	0				
	Angle =	-63.4671	Rad	-1.1071				Reflect about X - axis

					0.45	0.89	0.67
[Tc] =	[Tt]'	[Tr]'	[Tm]		-0.89	0.45	-1.34
					0.00	0.00	1.00
				0.447	0.89	0.67	
[Tc] =	[Tt]'	[Tr]'			0.89	-0.45	1.34
					0.00	0.00	1.00
[Tc] =	[Tt]'			-0.60	0.80	-0.90	
				0.80	0.60	1.20	
				0.00	0.00	1.00	
[Tc] =				-0.60	0.80	-2.40	
				0.80	0.60	1.20	
				0.00	0.00	1.00	
	-2.00		-3.80		0.80		
P =	4.00	Q =	6.40	R =	8.60		
	1.00		1.00		1.00		

Coordinate Systems

Three types of coordinate systems are needed in order to input, store, and display model geometry and graphics. These are Model Coordinate System (MCS), Working Coordinate System (WCS), and Screen Coordinate System (SCS), respectively. Other names for MCS are database, master, or world coordinate system. Another name for SCS is device coordinate system.

Working Coordinate System

- It is a convenient user-defined system that facilitates geometric construction.
- It can be established at any position and orientation in space that the user desires.
- While the user can input data in reference to the WCS, the CAD software performs the necessary transformations to the MCS before storing the data.
- The ability to use two separate coordinate systems within the same model database in relation to one another gives the user great flexibility.
- A WCS requires three noncollinear points to define its XY plane. The first defines the origin, the first and the second define the X axis, and the third point with the first defines the V axis.
- The Z axis is determined as the cross product of the two unit vectors in the directions defined by the lines connecting the first and the second (the X axis), and the first and the third points (V axis).

Model Coordinate System

- The model coordinate system is defined as the reference space of the model with respect to which all the model geometrical data is stored.
- It is a Cartesian system which forms the default coordinate system used by a particular software program.
- The X, Y, and Z axes of the MCS can be displayed on the computer screen.
- The origin of the MCS can be arbitrarily chosen by the user while its orientation is established by the software.
- The three default sketch planes of a CAD/CAM system define the three planes of the MCS and their intersection point is the MCS origin.
- When a CAD designer begins sketching, the origin becomes a corner point of the profile being sketched. The sketch plane defines the orientation of the profile in the model 3D space.
- The MCS is the only coordinate system that the software recognizes when storing or retrieving graphical information in or from a model database.

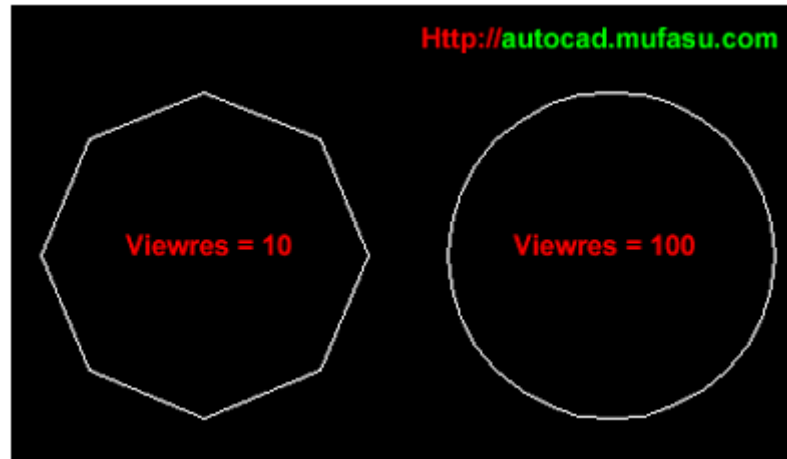
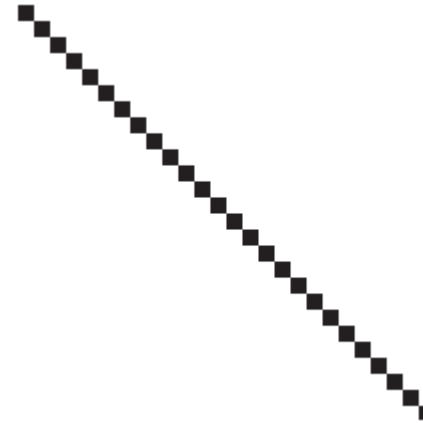
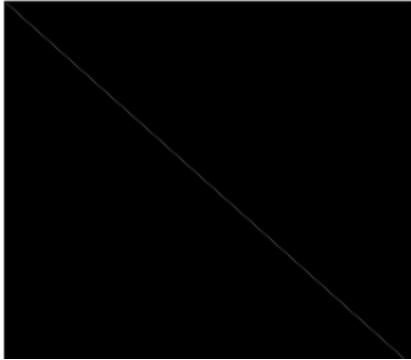
Screen Coordinate System

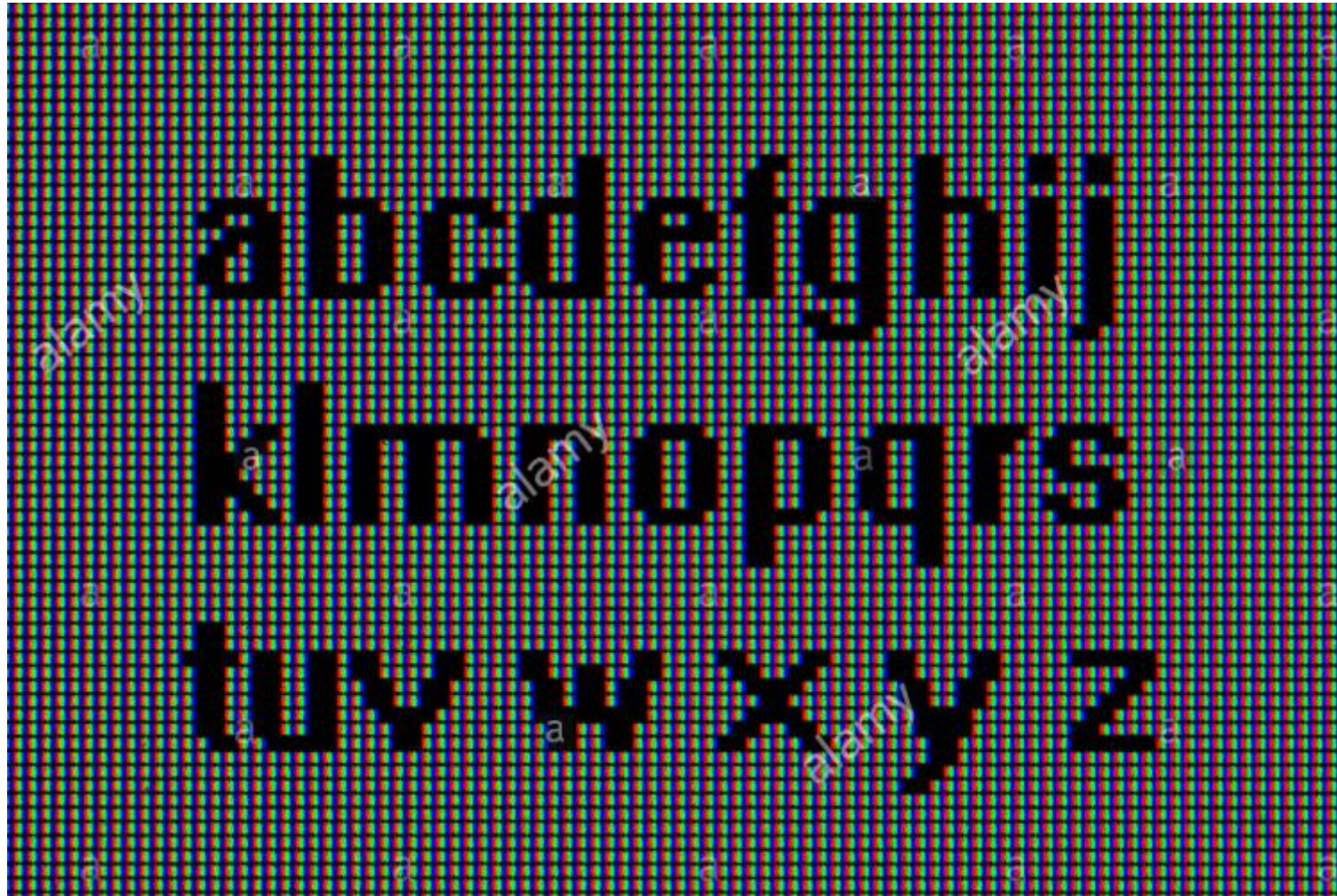
- In contrast to the MCS and WCS, the screen coordinate system (SCS) is defined as a 2D device-dependent coordinate system whose origin is usually located at the lower left corner of the graphics display.
- The physical dimensions of a device screen (aspect ratio) and the type of device (raster) determine the range of the SCS. The SCS is mostly used in view-related clicks such as definitions of view origin and window or clicking a view to select it for graphics operations.
- This SCS is used by the CAD/CAM Software to display relevant graphics by converting directly from MCS coordinates to SCS (physical device) coordinates.
- A transformation operation from MCS coordinates to SCS coordinates is performed by the software before displaying the model views and graphics. Typically, for a geometric model, there is a data structure to store its geometric data (relative to MCS), and a display file to store its display data (relative to SCS).

Mapping of Geometric Models (Coordinate Transformations):

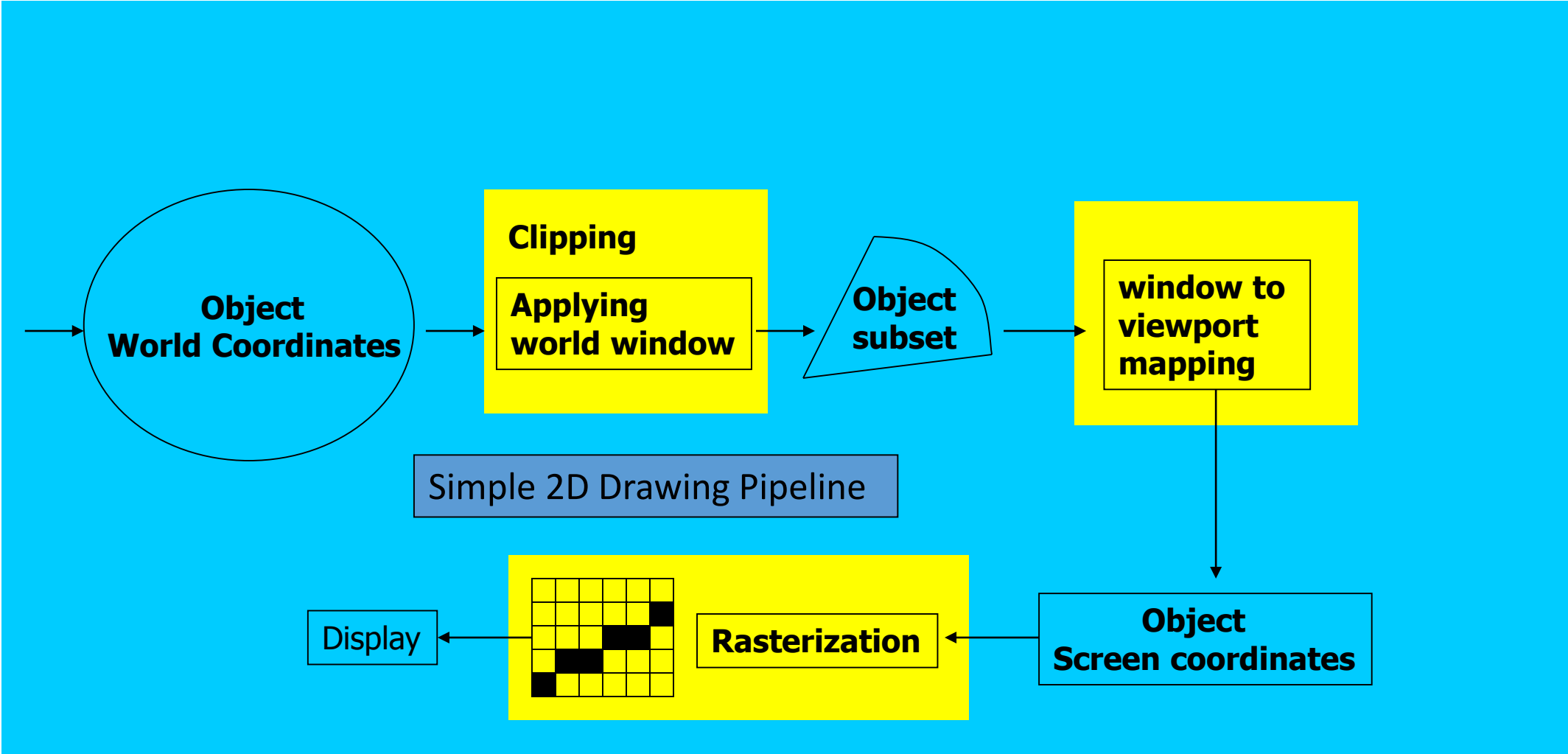
- Mapping is transformation of one coordinate system to other.
- While transformations involve one point and one coordinate system, mapping involves one point and two coordinate systems.
- Mapping the point changes its description from one coordinate system to another, but it does not change its location in the modelling space.
- We input coordinates relative to the coordinate system of the sketch plane, while CAD software maps the input to the model MCS coordinates before storing it in the model database.
- It also useful in assembly modelling. Mapping a geometric model, a collection of points, from one coordinate system to another does not change its position and orientation with respect to the origins of both systems. It only changes the description of such position and orientation.

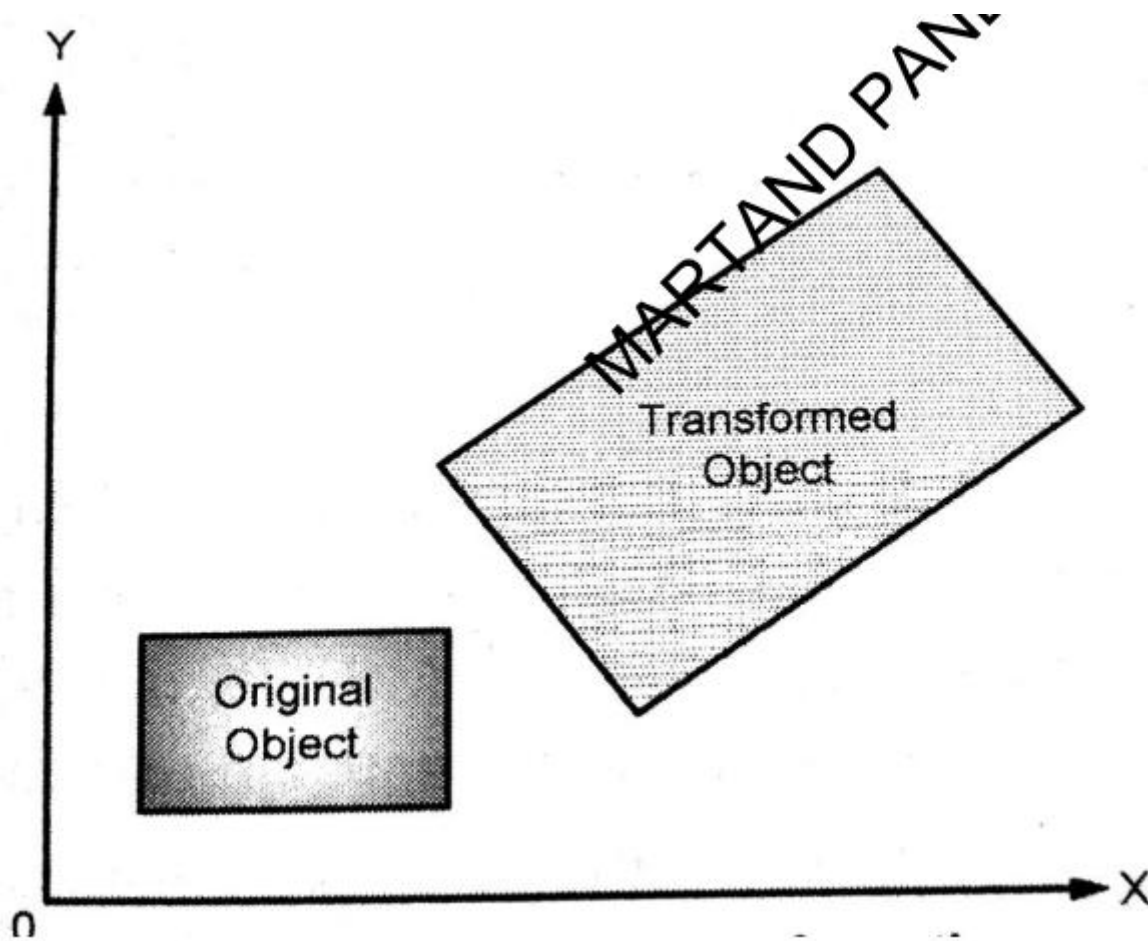
CAD Entities in ZOOM view





2D Graphics Pipeline





Geometric Transformation Changes

1. Orientation
2. Size
3. Shape

In Database as well as on the graphics image

The Problem of Scan Conversion

$$y = mx + b$$

where, m = Slope of the line

b = the y intercept of a line

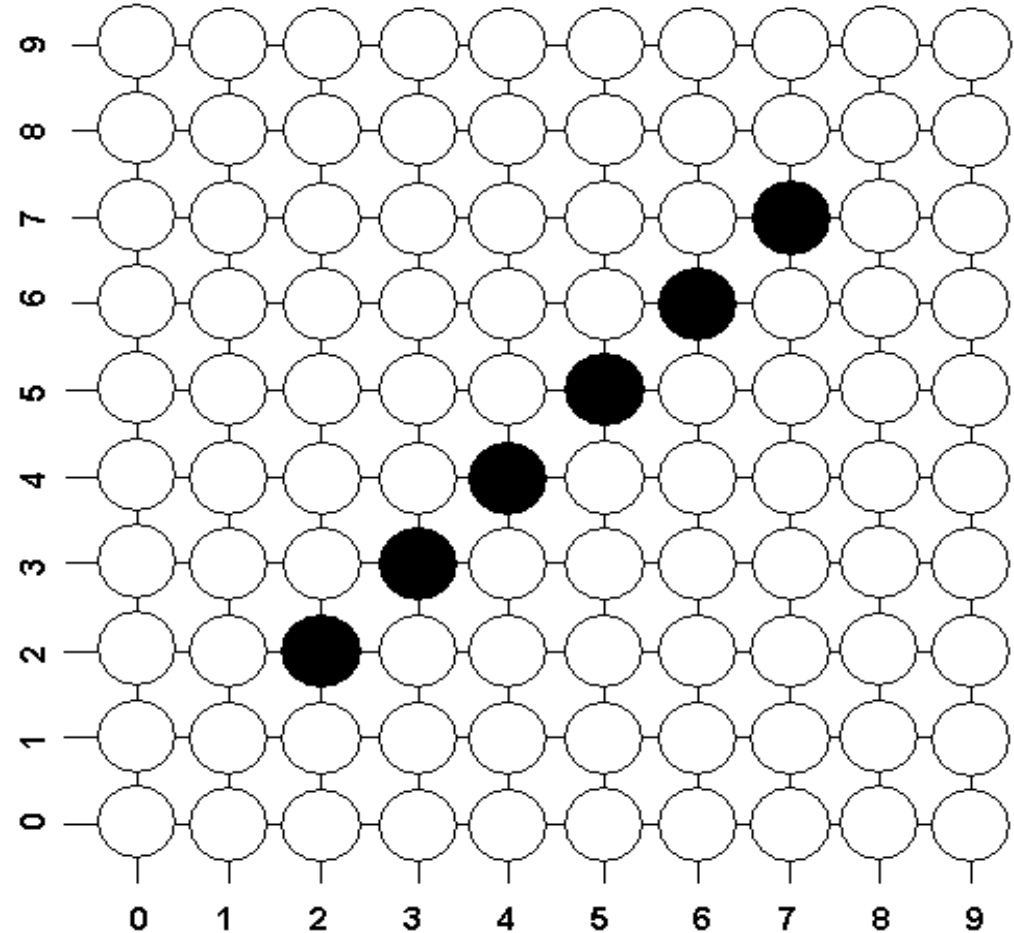
A line segment is defined by the coordinate positions of the line endpoints.

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 on

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 :

$$m = \frac{Y_{end} - Y_{start}}{X_{end} - X_{start}}$$

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

$$\frac{y_2 - y_1}{x_2 - x_1} = m$$

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 y_2 and x_2 may not be integers. The Round function must be used to obtain an integer coordinate value.

Simple DDA Line Drawing Algorithm

Case 1: For $\text{abs}(m) < 1$ and $X_{\text{start}} < X_{\text{end}}$,

we generate the line by incrementing the previous x value one unit until X_{end} is reached and then solve for y . if $X_{\text{start}} > X_{\text{end}}$, swap the two endpoints. Thus for these consecutive points:

$$x_2 = x_1 + 1 \quad \text{or} \quad x_2 - x_1 = 1$$

Substituting this difference into equation 1 yields:

$$(y_2 - y_1)/1 = m \quad \text{or} \quad y_2 = y_1 + m$$

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 (X_{start}, Y_{start}) to (X_{end}, Y_{end}) using simple DDA algorithm (**case 1**):

Simple DDA Line Drawing Algorithm

$m = (Y_{end} - Y_{start}) / (X_{end} - X_{start})$

If $(\text{abs}(m) < 1 \text{ and } X_{start} > X_{end})$ **then**

Swap endpoints $X_{start} \leftrightarrow X_{end}$ and $Y_{start} \leftrightarrow Y_{end}$

end if

Set pixel (X_{start}, Y_{start}) with desired color

If $\text{abs}(m) < 1$ **then**

$y = Y_{start}$

$x = X_{start} + 1$

Next: $y = y + m$

Set pixel $(x, \text{Round}(y))$ with desired color

$x = x + 1$

If $x \leq X_{end} - 1$ **then** go to **Next**

endif

Set pixel (X_{end}, Y_{end}) 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**:

$$m = (Y_{end} - Y_{start}) / (X_{end} - X_{start}) = (4 - 0) / (7 - 2) = 4/5 = 0.8$$

$$\Delta x = (X_{end} - X_{start}) = (7 - 2) = 5$$

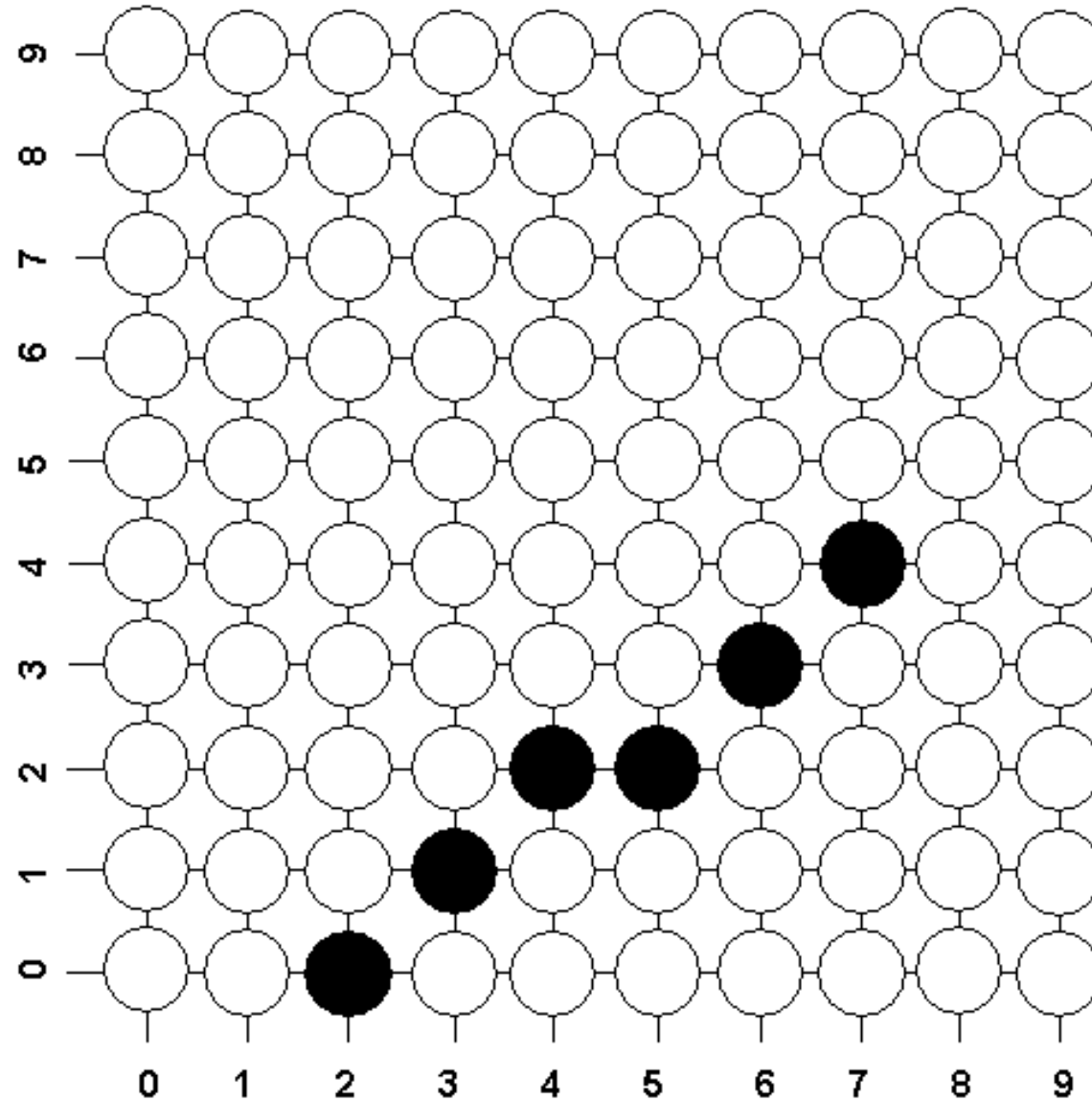
$$\Delta y = (Y_{end} - Y_{start}) = (4 - 0) = 4$$

$$\text{Increment in X} = X_{start} + (\Delta x / 5) = \mathbf{X_{start}} + 1$$

$$\text{Increment in Y} = Y_{start} + (\Delta y / 5) = \mathbf{Y_{start}} + 0.8$$

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. :

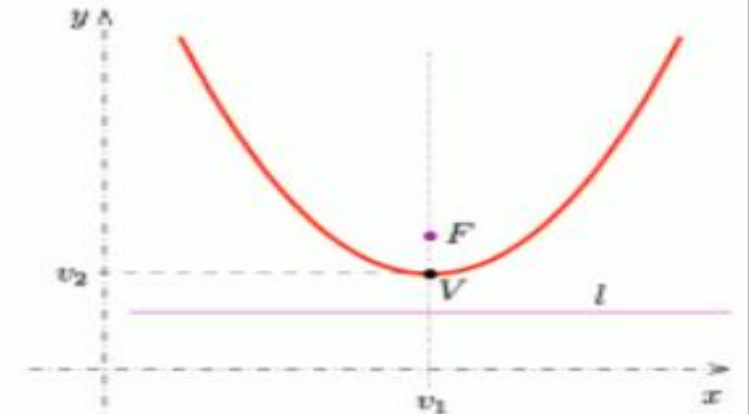
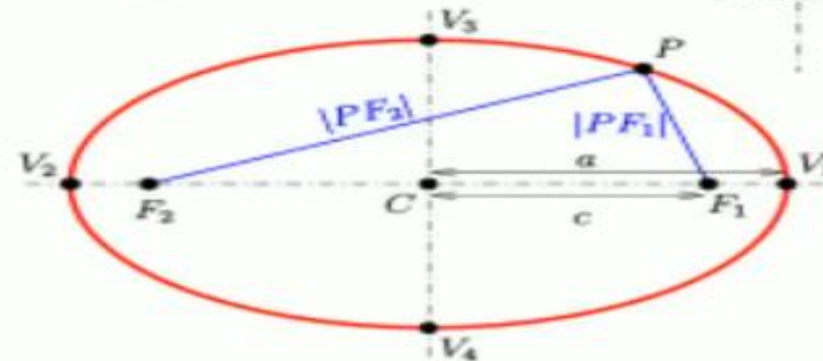
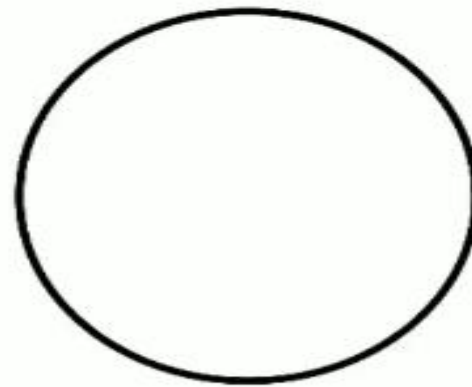
• Lines

• Circles

• Ellipse

• Parabolas

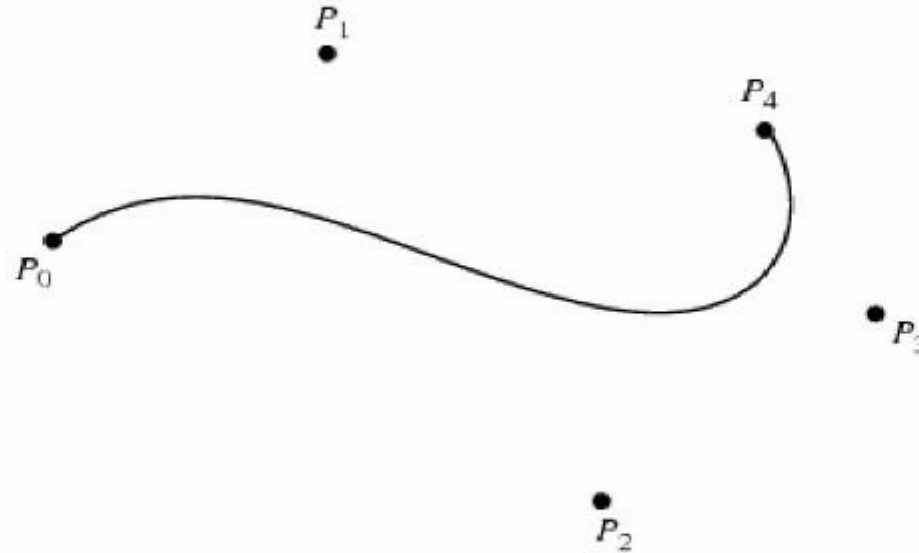
• Hyperbolas



(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 hulls**
- **Airplane wings**
- **Propeller blade**
- **Shoe insoles**
- **Bottles**

Curve Representation

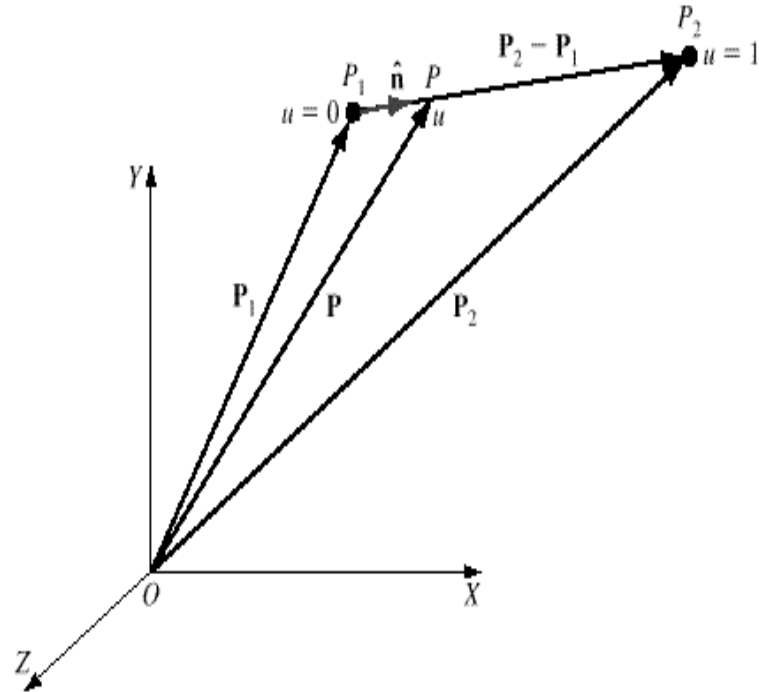
Cartesian or Non parametric

- $P = [x \ y \ z]^T = [x \ f(x) \ g(x)]^T$
- Curve represented by x, y and z coordinate or one to one relationship
- If the slope of curve is vertical or near to vertical its 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

- $P = [x(u) \ y(u) \ z(u)]^T \quad u_{\min} < u < u_{\max}$
- 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



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

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

$$P = P_1 + u(P_2 - P_1), \quad 0 \leq u \leq 1$$

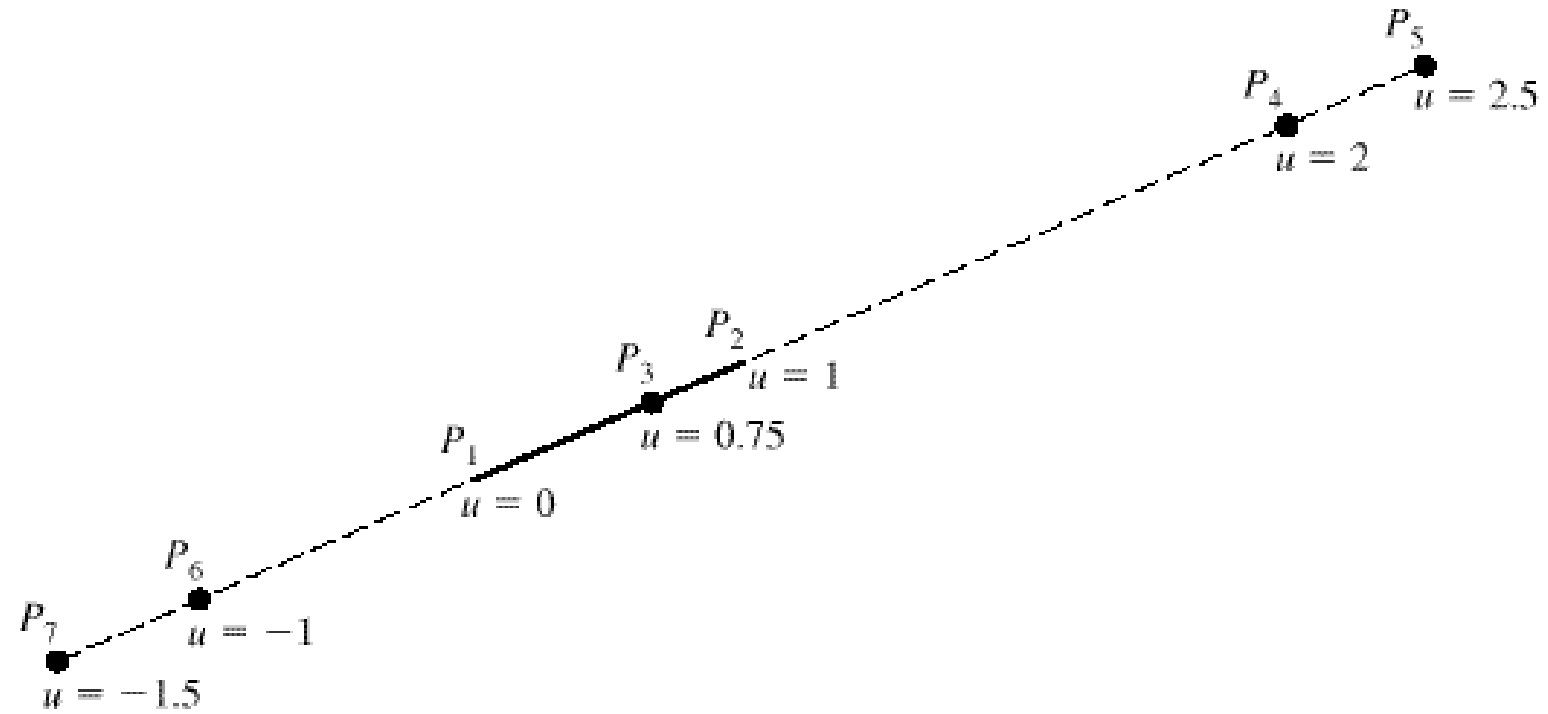
$$\mathbf{P} = \mathbf{P}_1 + u \mathbf{P}_2 - \mathbf{P}_1 \quad 0 \quad u \quad 1$$

In scalar form, this equation can be written as

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

$$y = y_1 + u (y_2 - y_1) \quad 0 \quad u \quad 1$$

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



The tangent vector of the line is given by

$$\mathbf{P} = \mathbf{P}_2 - \mathbf{P}_1 \quad (6.13)$$

or, in scalar form,

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

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

$$\hat{\mathbf{n}} = \frac{\mathbf{P}_2 - \mathbf{P}_1}{L} \quad (6.15)$$

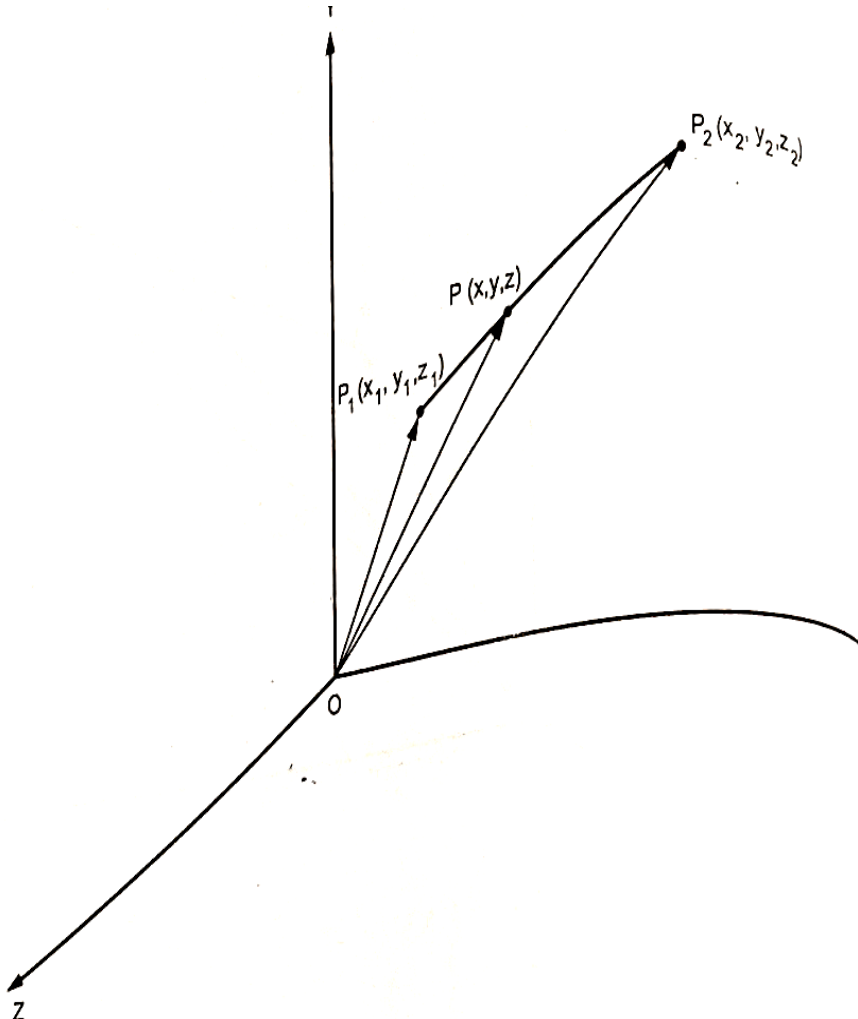
where L is the length of the line,

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

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,

$$P = P_1 + u(P_2 - P_1) \quad 0 \leq u \leq 1$$

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

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

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

2. Tangent vector of line :

The tangent vector of the line is given by,

$$\begin{aligned} \mathbf{P}' &= P_2 - P_1 \\ &= [i \ j \ k] \begin{bmatrix} 5-3 \\ 7-4 \\ 1-6 \end{bmatrix} \\ &= [i \ j \ k] \begin{bmatrix} 2 \\ 3 \\ -5 \end{bmatrix} \end{aligned}$$

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

...Ans.

3. Unit vector in direction of line :

$$\begin{aligned} L &= |P_2 - P_1| \\ &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \\ &= \sqrt{(5-3)^2 + (7-4)^2 + (1-6)^2} \end{aligned}$$

or $L = 6.164$

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

$$\begin{aligned} \hat{\mathbf{n}} &= \frac{(P_2 - P_1)}{|P_2 - P_1|} \\ &= \frac{\mathbf{P}'}{L} \\ &= \frac{2\mathbf{i} + 3\mathbf{j} - 5\mathbf{k}}{6.164} \end{aligned}$$

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

...Ans

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 two 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 :

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

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

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

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

- For line L_2 :

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

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

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

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

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 $P'_{L1} = 4i + 4j - 6k$

- 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 $P'_{L2} = -4i + 6j + 6k$

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.

$$\begin{aligned} P'_{L1} \cdot P'_{L2} &= (4i + 4j - 6k) \cdot (-4i + 6j + 6k) \\ &= (4 \times -4) + (4 \times 6) + (-6 \times 6) \\ &= -16 + 24 - 36 \end{aligned}$$

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

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 $P'_{L_1} \times P'_{L_2} = 60i + 40k$

As $P'_{L_1} \times P'_{L_2} \neq 0$, 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,

$$P_{L_1} = P_{L_2}$$

- From Equations (a) and (b),

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

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

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

and $7 - 6u = 4 + 6v$

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

$$4u - 6v = 5$$

$$6u + 6v = 3$$

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

- From Equations (c) and (d),

$$10v = -3$$

$$\therefore v = -0.3$$

- From Equation (c),

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

$$4u = 3.2$$

$$\therefore u = 0.8$$

- 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 (c) satisfies the values obtained from Equations (c) and (d).

...Ans.

Therefore two lines L_1 and L_2 are intersecting.

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),

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

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

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

or $P_{\text{int}} = (4.2, 5.2, 2.2)$

...Ans.

Parametric Equation of Circle

Representation 1

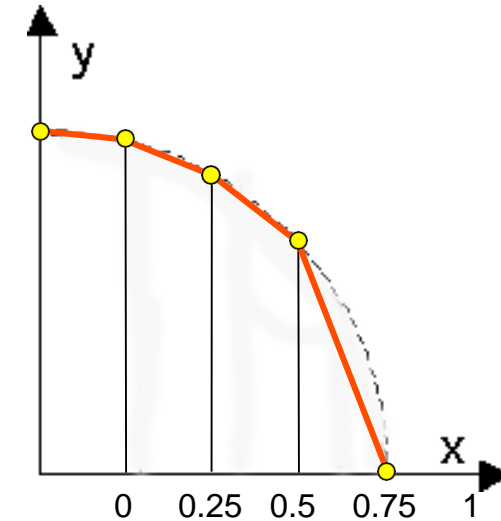
Non-parametric Representation

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

$$x = u$$

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

(a)



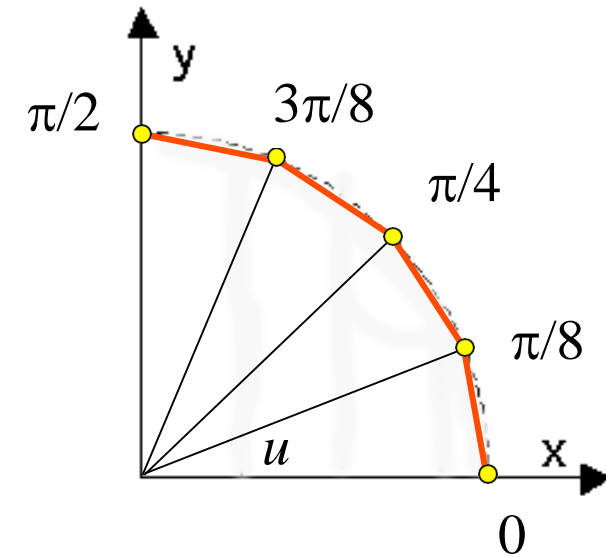
Limitations of non parametric representation

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

Representation 2 :
Parametric representation

$$\begin{aligned}x &= \cos u \\y &= \sin u\end{aligned}$$

(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

$$\begin{aligned}x_n &= r \cos \theta \\y_n &= r \sin \theta\end{aligned}$$

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

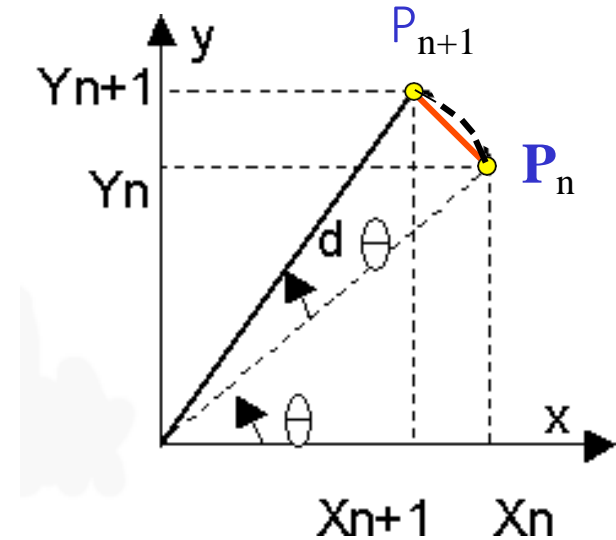
$$\begin{aligned}x_{n+1} &= r \cos(\theta + d\theta) \\&= r \cos \theta \cos d\theta - r \sin \theta \sin d\theta\end{aligned}$$

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

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 be calculated once only, hence it speeds up the circle generation and display



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

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

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

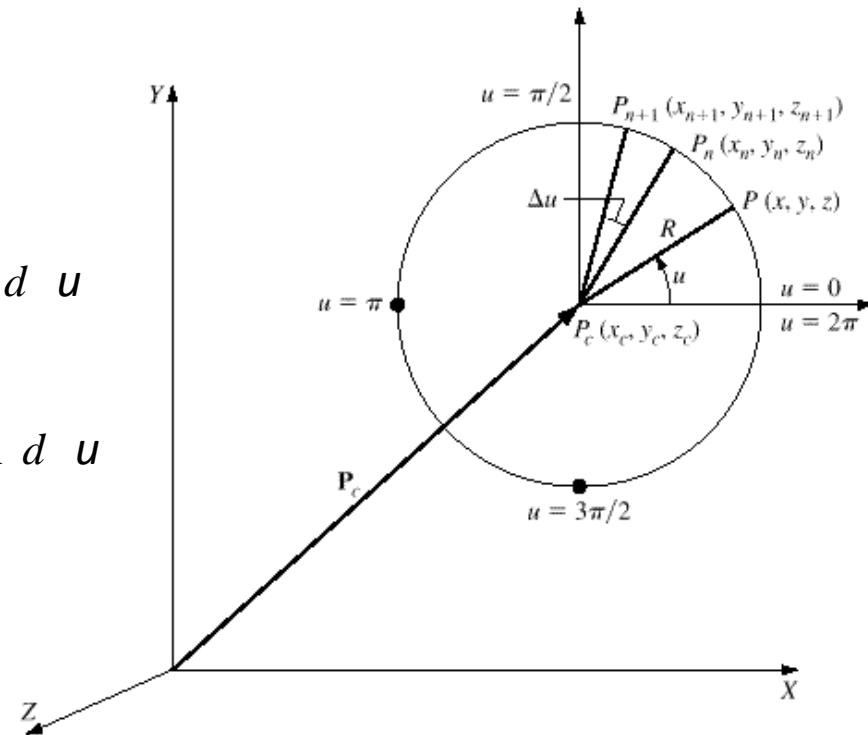
The basic parametric equation of a circle can be written as (refer to Figure 6.11):

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

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

$$\begin{aligned} x_{n+1} &= x_c + r \cos(u + d u) \\ &= x_c + r \cos u \cos d u - r \sin u \sin d u \end{aligned}$$

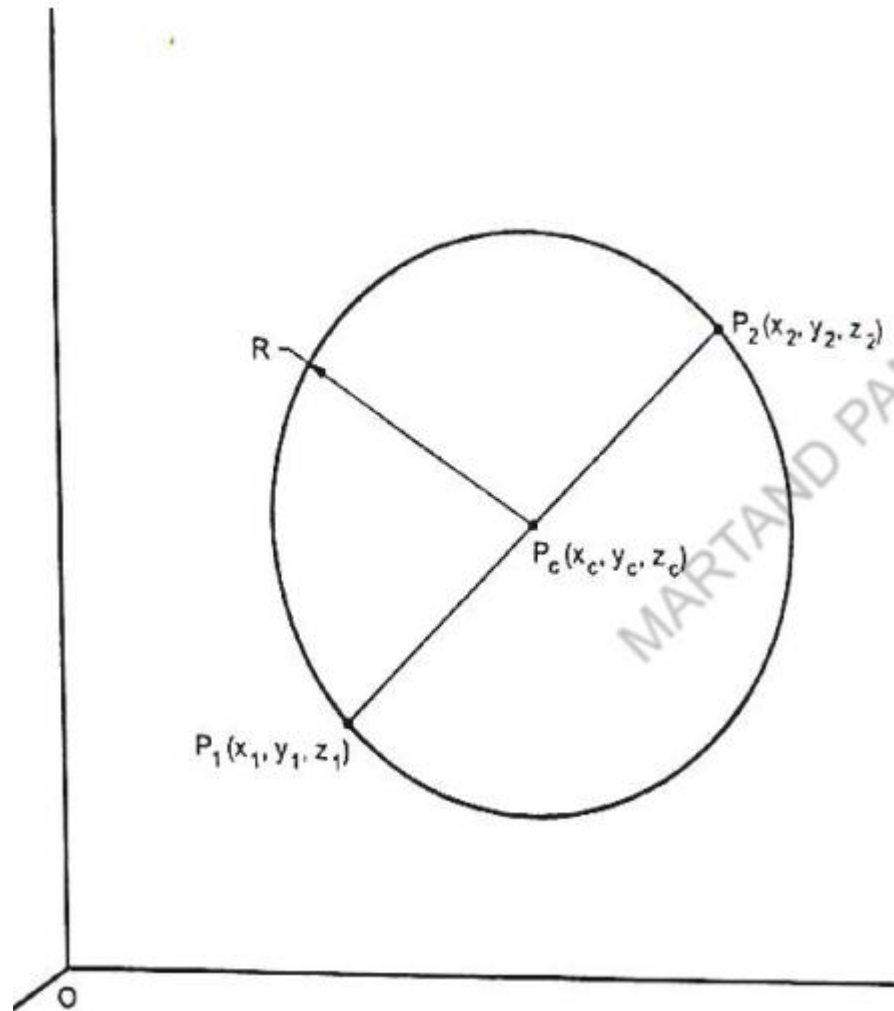
$$\begin{aligned} y_{n+1} &= y_c + r \sin(u + d u) \\ &= y_c + r \sin u \cos d u + r \cos u \sin d u \end{aligned}$$



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

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

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



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

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

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

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 : $P_c(x_c, y_c, z_c) = P_c(3, 4, 2)$;

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

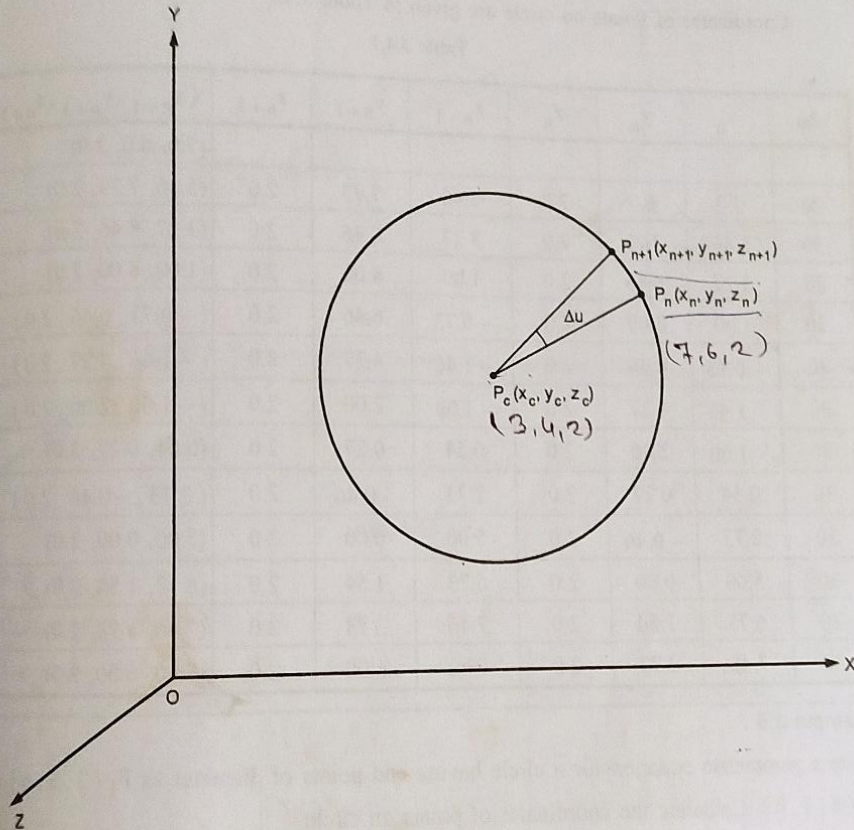


Fig. 3.4.1

1. Parametric equation :

From Fig. 3.4.1,

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

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

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

$$\therefore x_{n+1} = 3 + (x_n - 3) \cos \Delta u - (y_n - 4) \sin \Delta u$$

$$y_{n+1} = 4 + (y_n - 4) \cos \Delta u + (x_n - 3) \sin \Delta u$$

$$z_{n+1} = 2$$

...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)

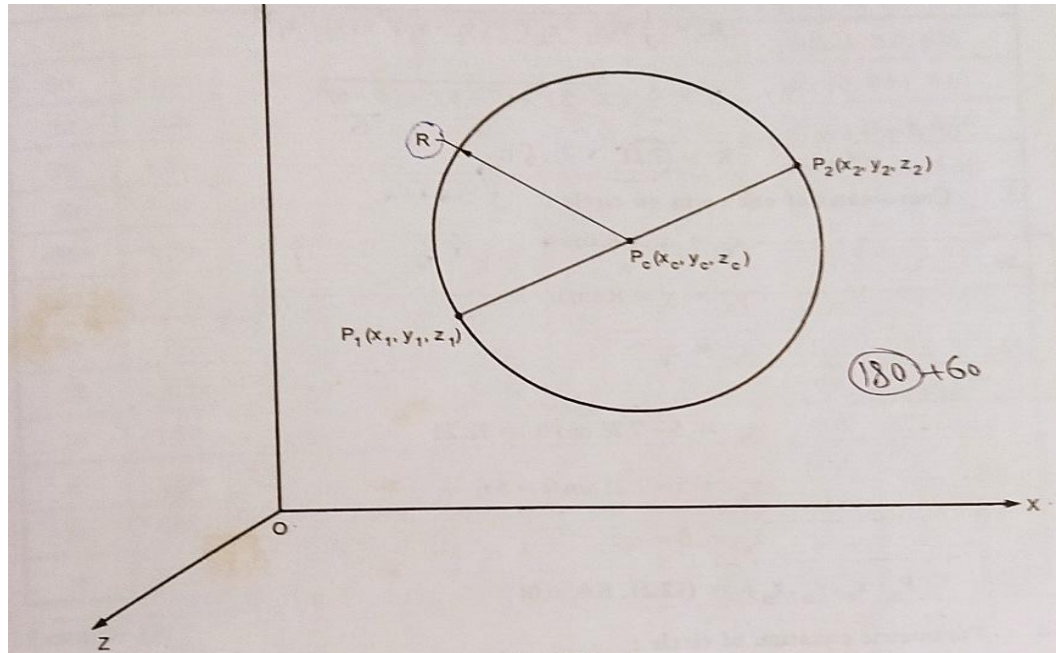


Fig. 3.5.1

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,

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

$$= \frac{1}{2} \sqrt{\frac{6}{0} + \frac{4}{0} + \frac{0}{0}}$$

or

$$R = 7.21 \approx 3.60$$

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 :

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

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

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

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

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

$$z_{n+1} = 6$$

...Ans.

5. Coordinates of points on circle :

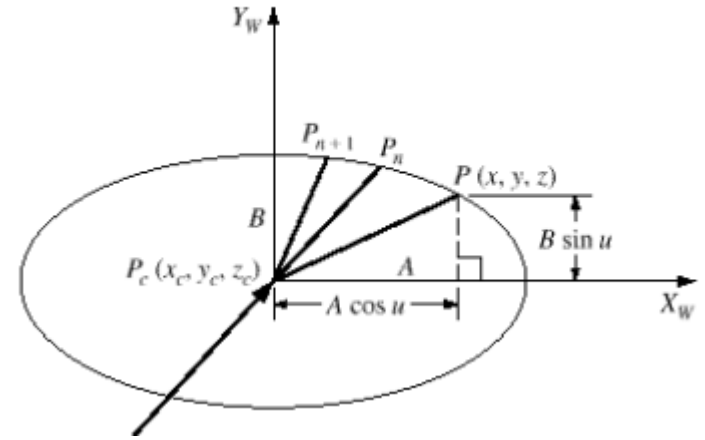
The coordinates of points on circle are given in Table 3.5.1.

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				(12.21, 5.0, 6.0)
30	11.24	8.61	6.0	11.24	8.61	6.0	(11.24, 8.61, 6.0)
30	8.61	11.24	6.0	8.61	11.24	6.0	(8.61, 11.24, 6.0)
30	5.00	12.21	6.0	5.00	12.21	6.0	(5.00, 12.21, 6.0)
30	1.39	11.24	6.0	1.39	11.24	6.0	(1.39, 11.24, 6.0)
30	-1.24	8.61	6.0	-1.24	8.61	6.0	(-1.24, 8.61, 6.0)
30	-2.21	5.00	6.0	-2.21	5.00	6.0	(-2.21, 5.00, 6.0)
30	-1.24	1.39	6.0	-1.24	1.39	6.0	(-1.24, 1.39, 6.0)
30	-1.24	1.39	6.0	1.39	-1.24	6.0	(1.39, -1.24, 6.0)
30	1.39	-1.24	6.0	5.00	-2.21	6.0	(5.00, -2.21, 6.0)
30	5.00	-2.21	6.0	8.61	-1.24	6.0	(8.61, -1.24, 6.0)
30	8.61	-1.24	6.0	11.24	1.39	6.0	(11.24, 1.39, 6.0)
30	11.24	1.39	6.0	12.21	5.00	6.0	(12.21, 5.00, 6.0)

Parametric Equation of Ellips

$$\begin{aligned}
 x_n &= x_c + A \cos u \\
 y_n &= y_c + B \sin u \\
 z_n &= z_c
 \end{aligned}
 \quad 0 \leq \theta \leq 2\pi$$



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

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

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

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

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

Parametric Equation of an inclined Ellips

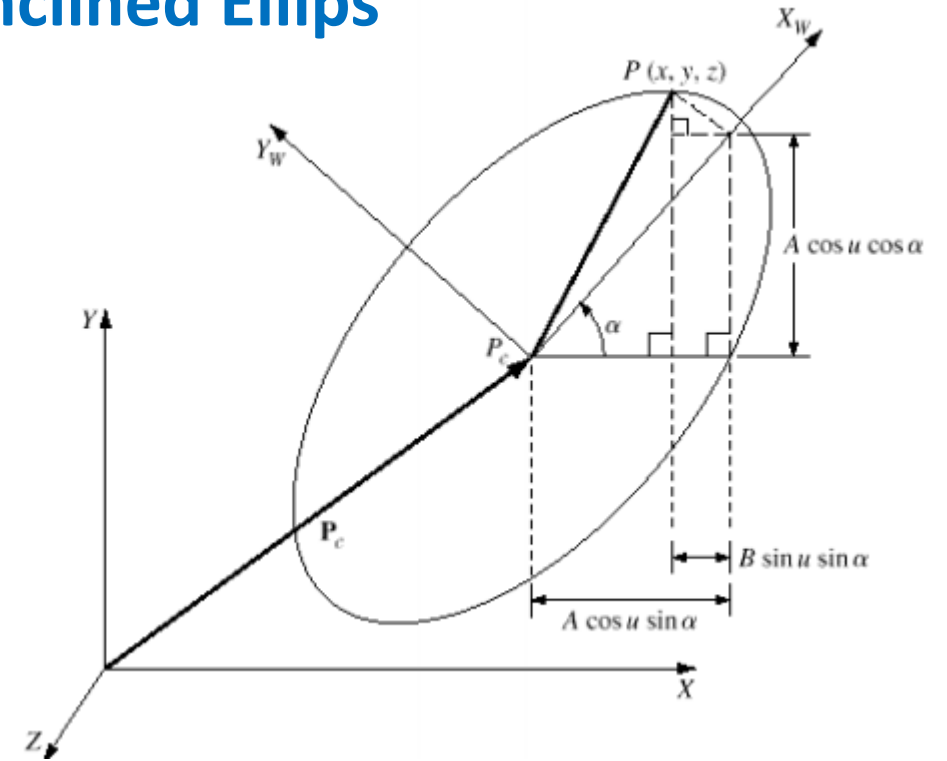
Parametric Equation of ellips inclined at angale α

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

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

$$z = z_c$$

$$0 \leq u \leq 2\pi$$



An inclined ellips

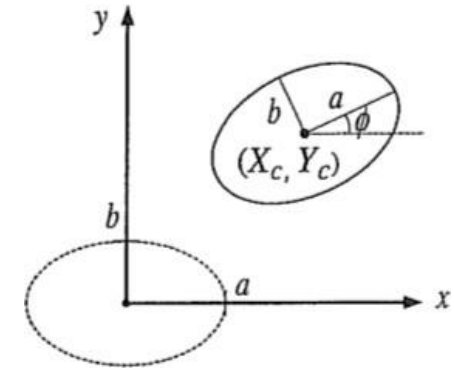
Parametric Equation of Inclined Ellips

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:

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

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

$$\begin{aligned} x^* &= x \cos \phi - y \sin \phi + X_c = a \cos \theta \cos \phi - b \sin \theta \sin \phi + X_c \\ y^* &= x \sin \phi + y \cos \phi + Y_c = a \cos \theta \sin \phi + b \sin \theta \cos \phi + Y_c \\ z^* &= 0 \quad (0 \leq \theta \leq 2\pi) \end{aligned}$$



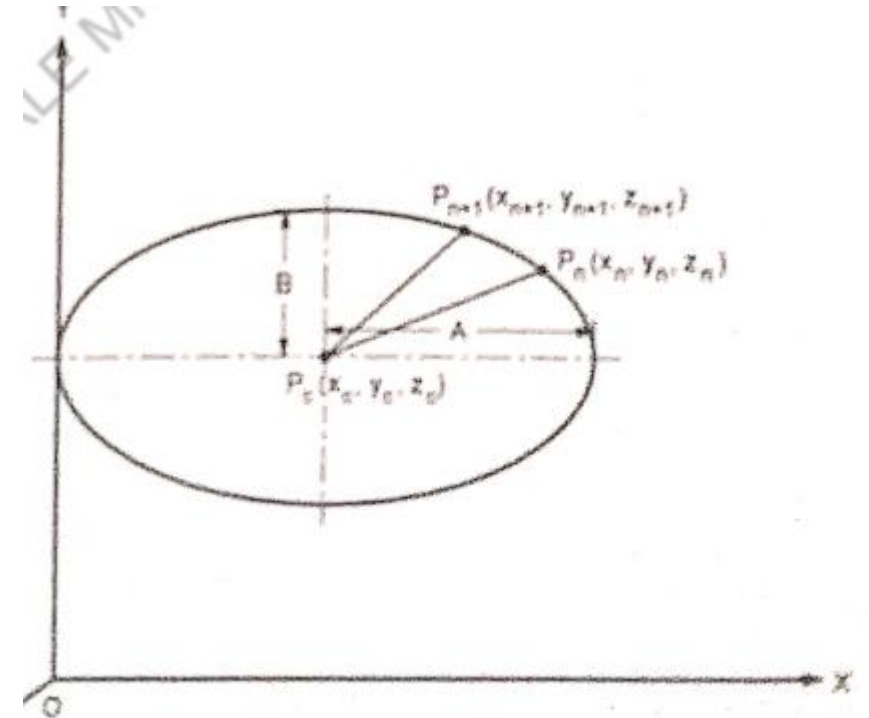
Example 1

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:

$$A = 10/2 = 5 \text{ units ; } B = 8/2 = 4 \text{ units}$$

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



1. Coordinates of any point on ellipse :

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

At $u = 0$;

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

2. Parametric equation of ellipse :

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

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

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

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

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

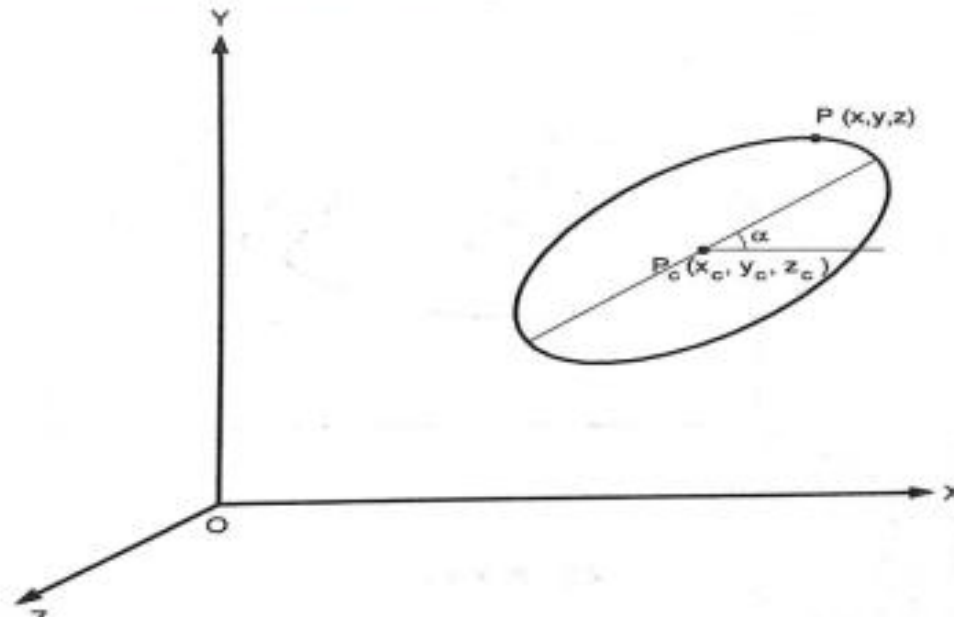
$$z_{n+1} = 3 \quad \dots \text{Ans.}$$

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

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



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

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

$$z = z_c$$

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

$$z = 4$$

$$\therefore x = 10 + 5.196 \cos u - 2 \sin u$$

$$y = 5 + 3 \cos u + 3.464 \sin u$$

$$z = 4$$

$$0 \leq u \leq 2\pi$$

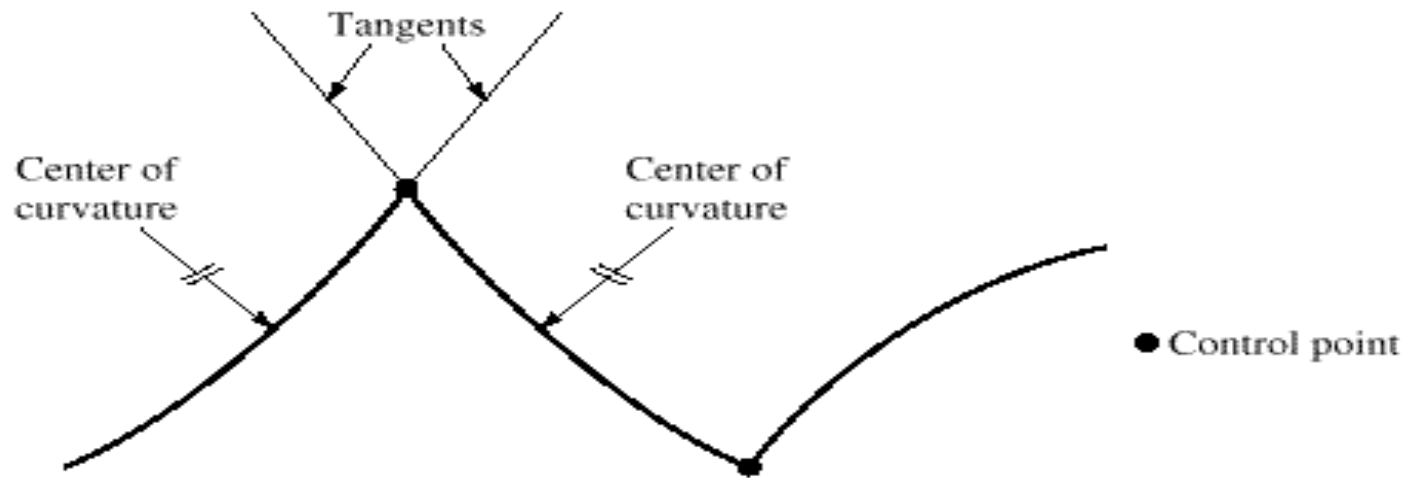
...Ans.

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.

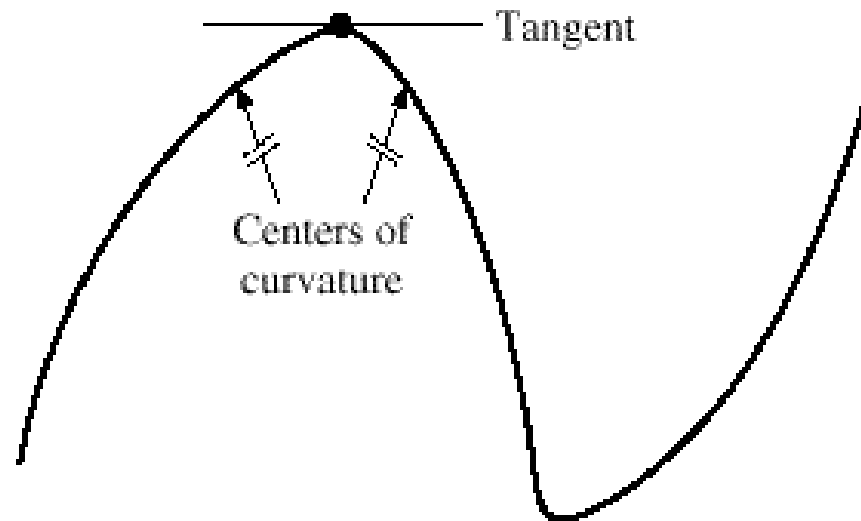
Continuity (C^0 , C^1 & C^2)

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.



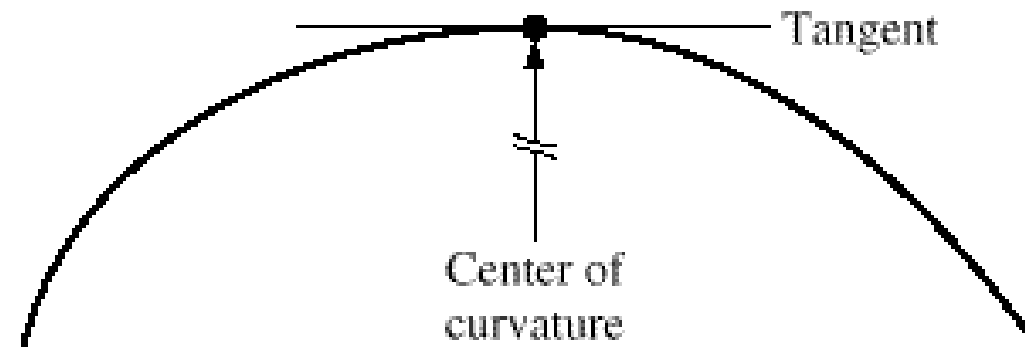
(a) Zero-order continuity (C^0 curve)

C^1 continuity: 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

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 to control the curve shape locally as opposed to global control. They also provide the possibility to add control points without increasing the degree of the curve. The kinds of B-spline curves can be fitted to a set of data points.

Representation



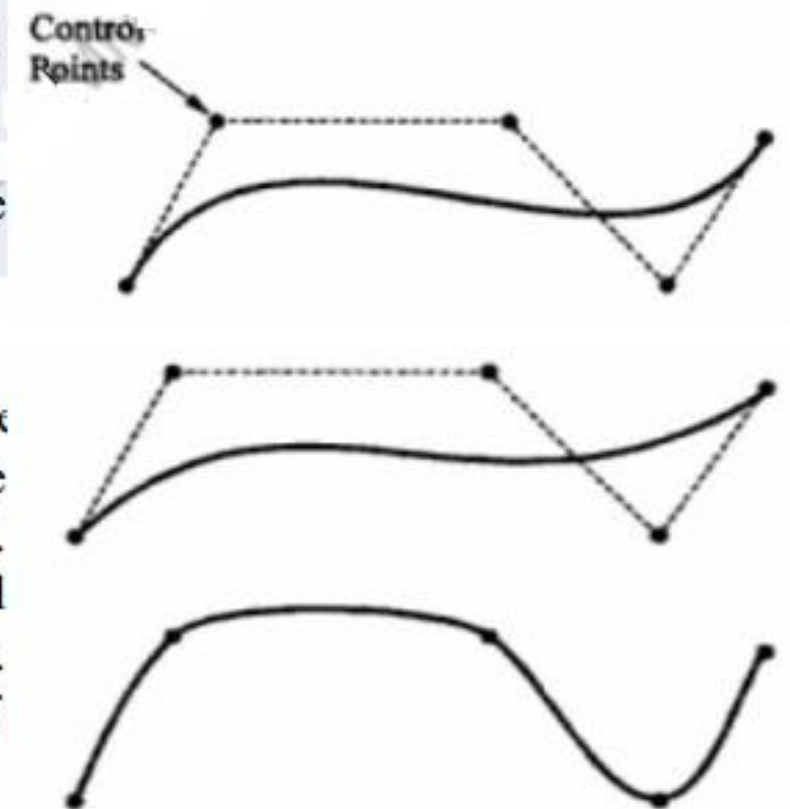
Type of Synthetic Curve

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.

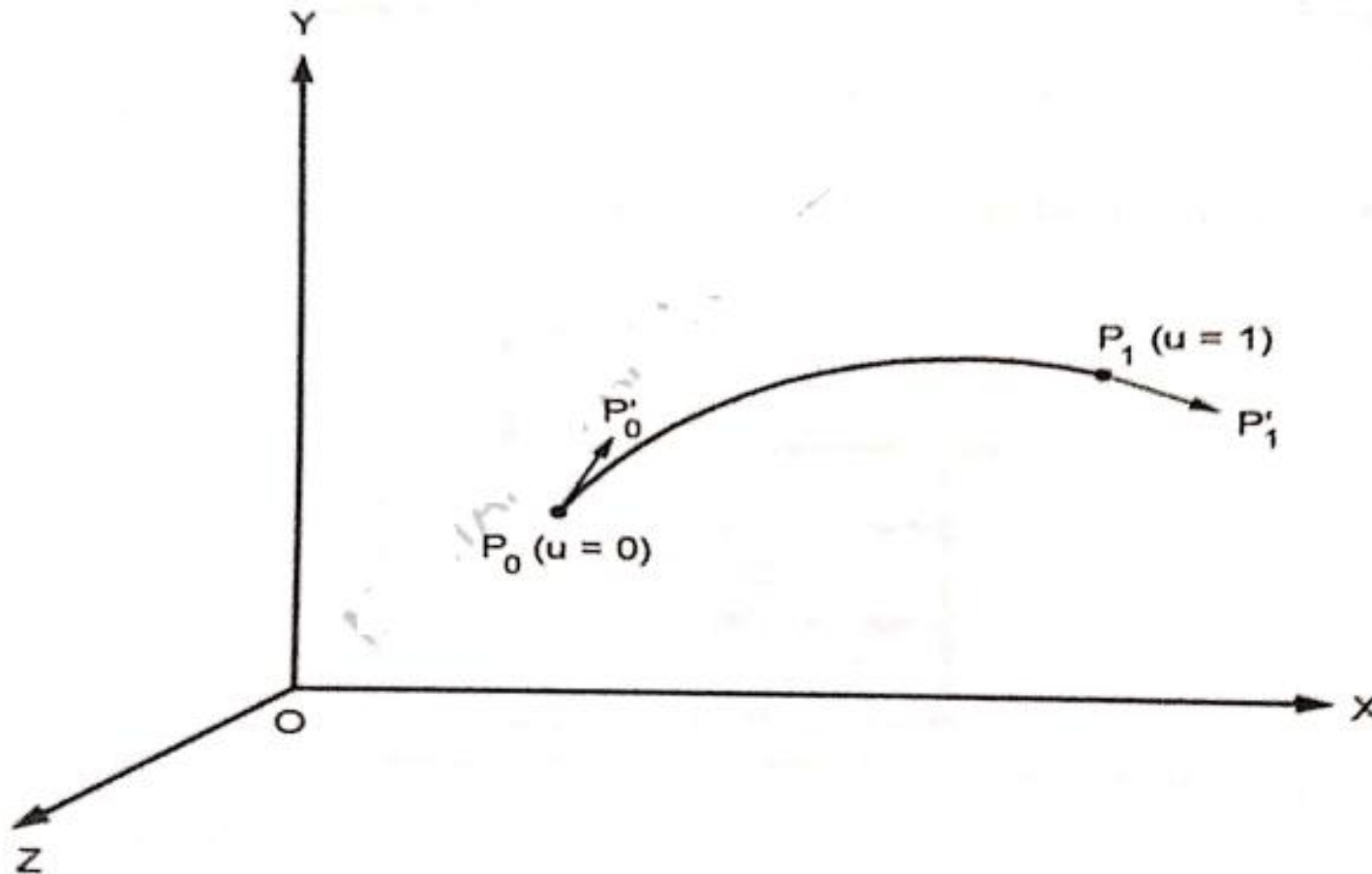
B-spline Curve: These are generalized form of the Bezier Curves and have the ability to control the curve shape locally as opposed to global control. They also provide the possibility to add control points without increasing the degree of the curve. The kinds of B-spline curves can be fitted to a set of data points.

Representation



Hermit Cubic Splines:

Hermit 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

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

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

Hermit Cubic Splines:

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

$$P(u) = au^3 + bu^2 + cu + d \quad 0 \leq u \leq 1$$

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

- Tangent Vector at any point P(u):

$$P'(u) = 3au^2 + 2bu + c$$

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

Hermit Cubic Splines:

- Point & Tangent Vector at $u=0$

$$P_0 = [0\ 0\ 0\ 1] \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \quad \dots(a)$$

$$\text{and } P'_0 = [0\ 0\ 1\ 0] \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \quad \dots(b)$$

- Point & Tangent Vector at $u=1$

$$P_1 = [1\ 1\ 1\ 1] \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \quad \dots(c)$$

$$\text{and } P'_1 = [3\ 2\ 1\ 0] \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \quad \dots(d)$$

Hermit Cubic Splines:

- Boundary conditions for Hermite Cubic Spline Curve:

Combining 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:

$$\text{or } \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} 2(P_0 - P_1) + P'_0 + P'_1 \\ -3(P_0 - P_1) - 2P'_0 - P'_1 \\ P'_0 \\ P_0 \end{Bmatrix}$$

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

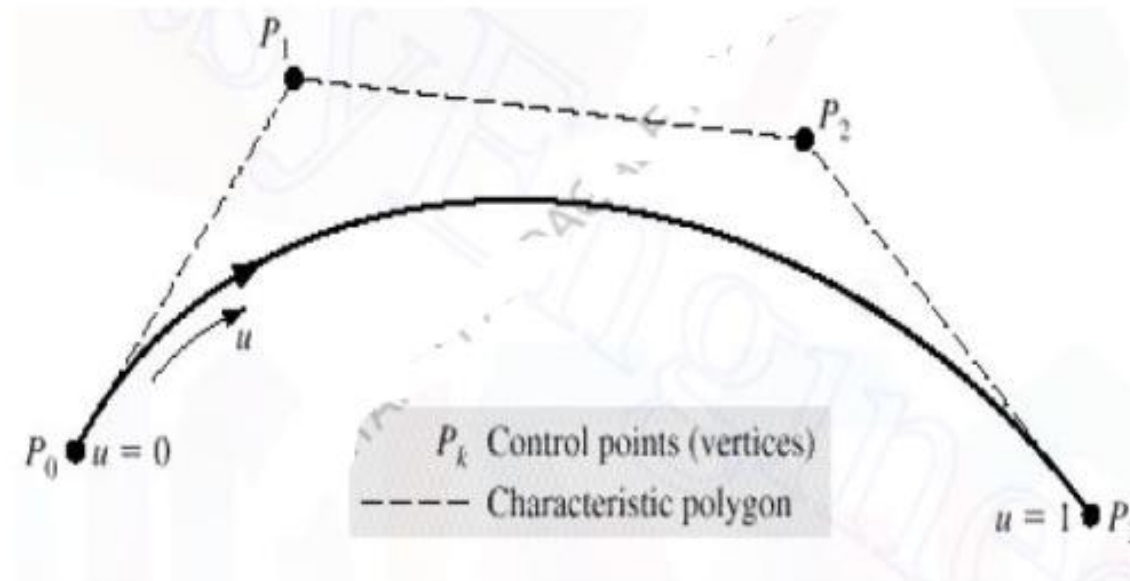
From Equation (2.14.3),

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

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 P_0 & P_n can be written as,

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

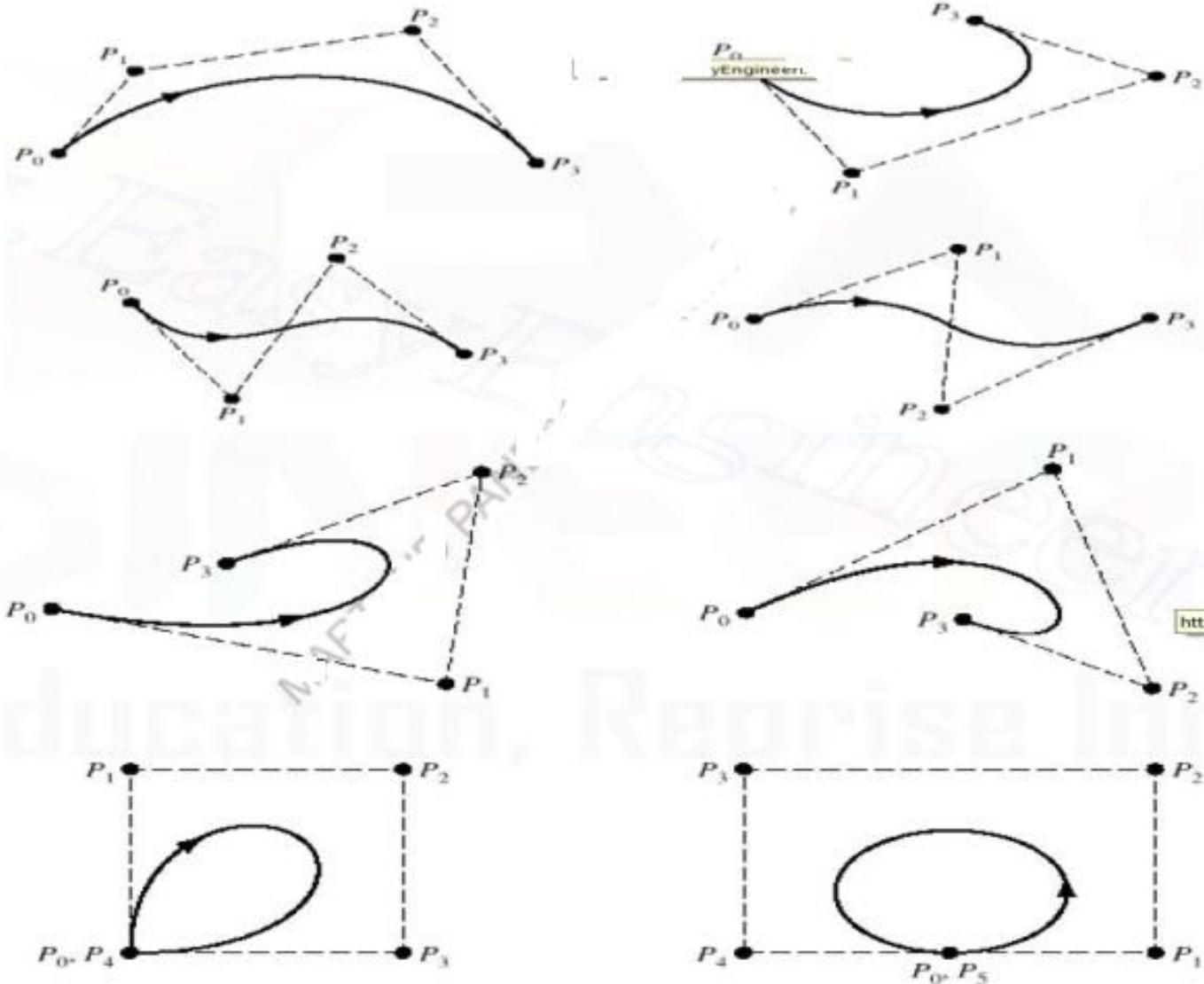
$$\begin{aligned}
 P(u) &= P_0 (1-u)^n + C(n, 1) P_1 u (1-u)^{n-1} \\
 &\quad + C(n, 2) P_2 u^2 (1-u)^{n-2} \\
 &\quad + \dots + C(n, n-1) P_{n-1} u^{n-1} (1-u) \\
 &\quad + P_n u^n, \quad 0 \leq u \leq 1
 \end{aligned}$$

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

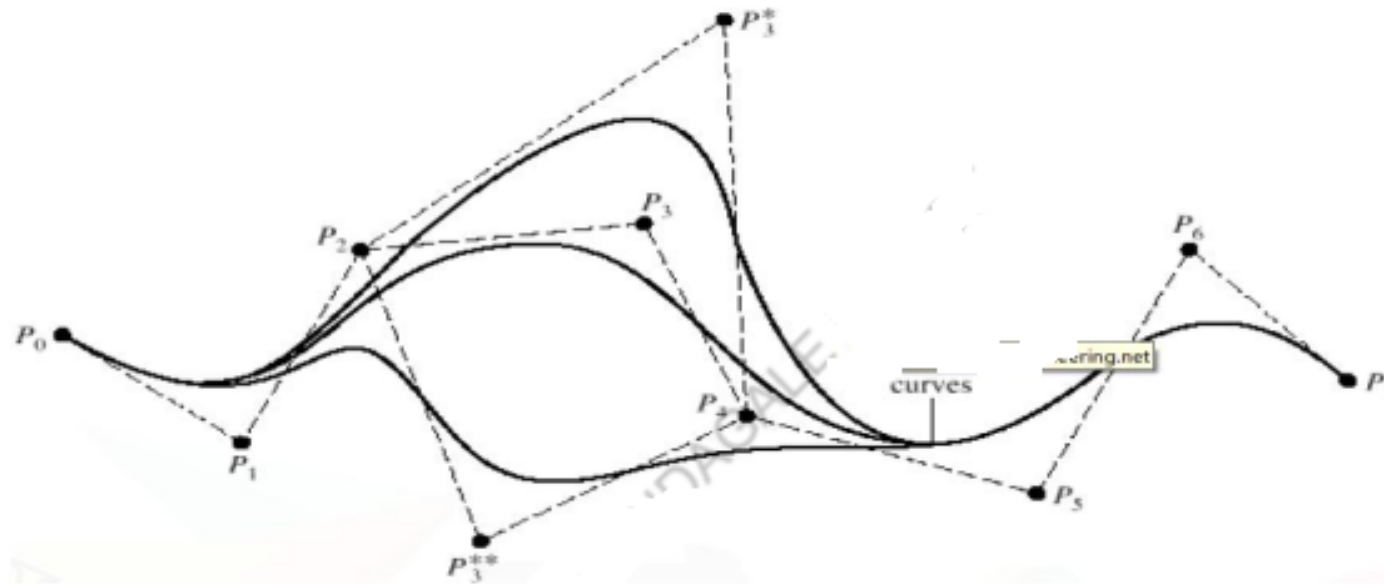
2. Bezier Curves:

- 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 n^{th} 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 P_0 & P_n to be coincident.
 8. The flexibility of it increases with increase in number of control points.

2. Bezier 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 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. E.g. NURBS
- 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.

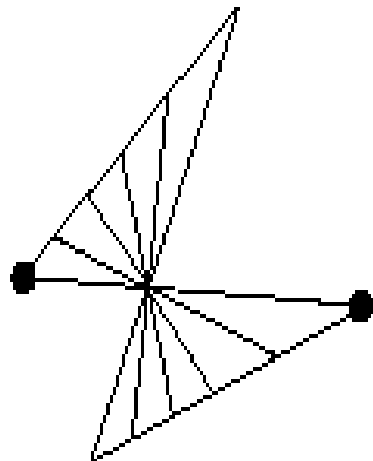
SURFACE MODELING

- 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.

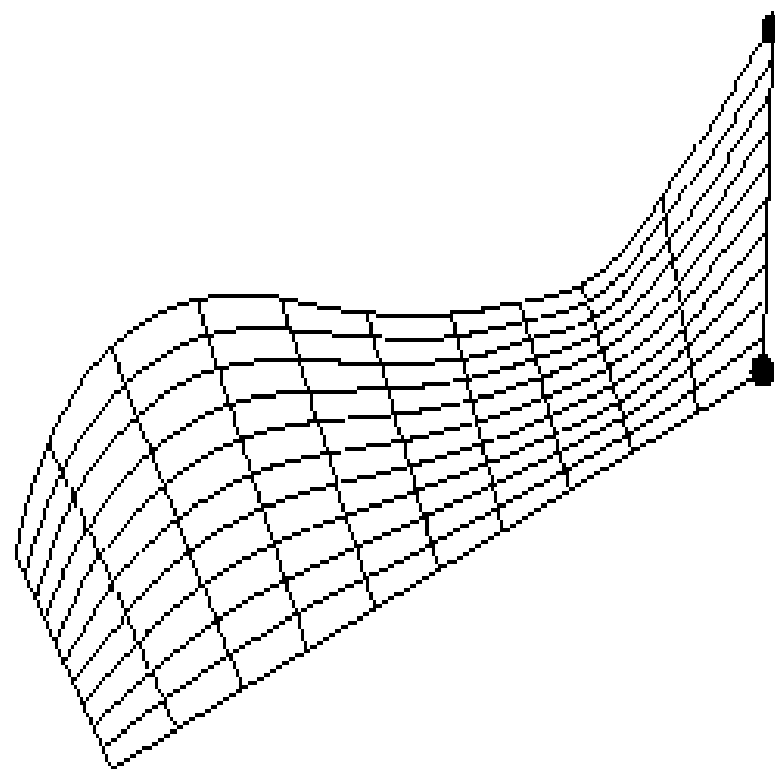
- Surface modeling is considered a more complex technique for displaying objects than wireframe modeling.
- Surface modeling has much less ambiguous display functionalities compared to wireframe modelling. but not as much or sophisticated as solid modeling. The technique often involves conversions between 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

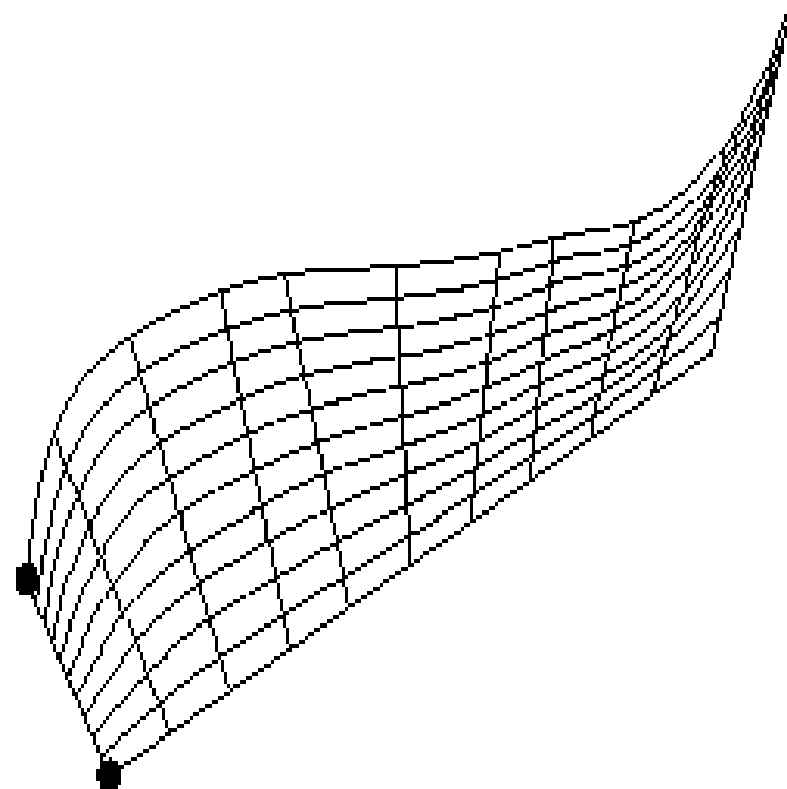
- Shape design and the representation of complex objects such as car, ship, and airplane bodies as well as castings cannot be achieved utilizing the curves covered in earlier.
- In such cases, surfaces must be utilized to describe objects precisely and accurately. We create surfaces, and then we use 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

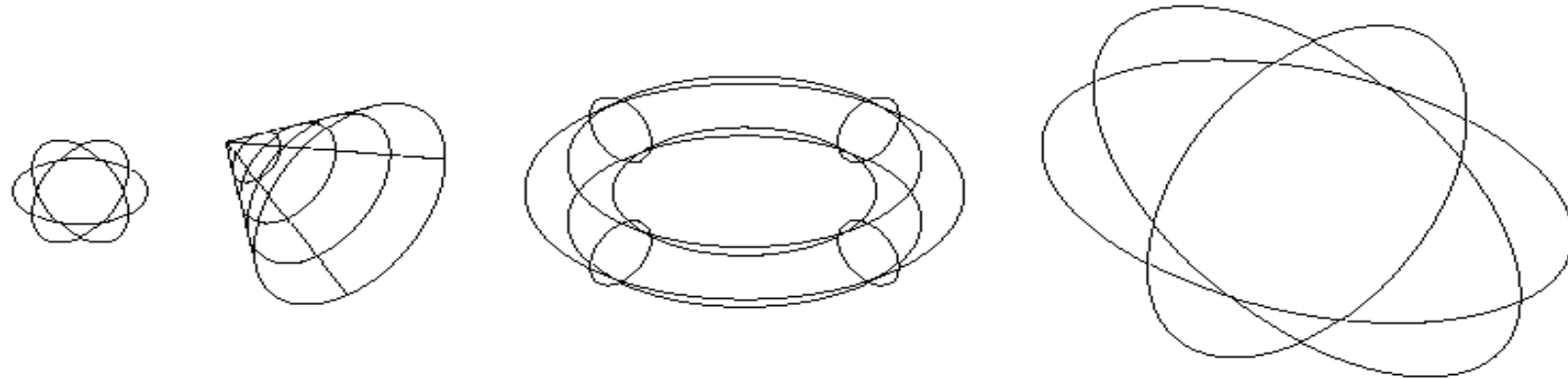


or

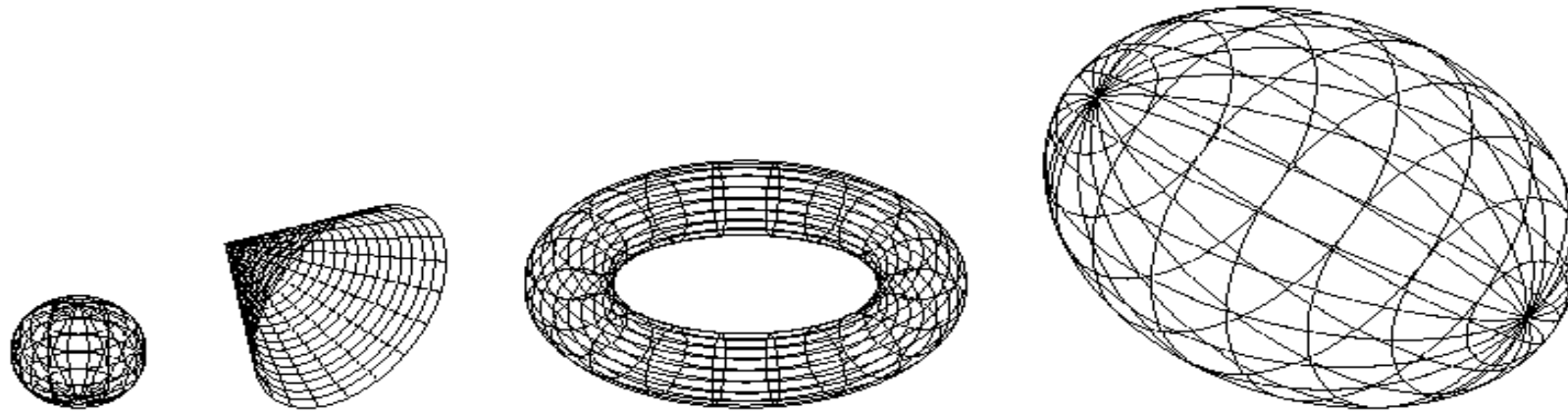


(b) Correct selected end points

- 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

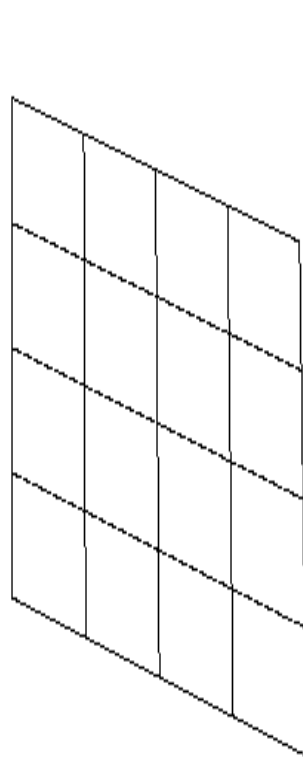


(b) 20×20 mesh size

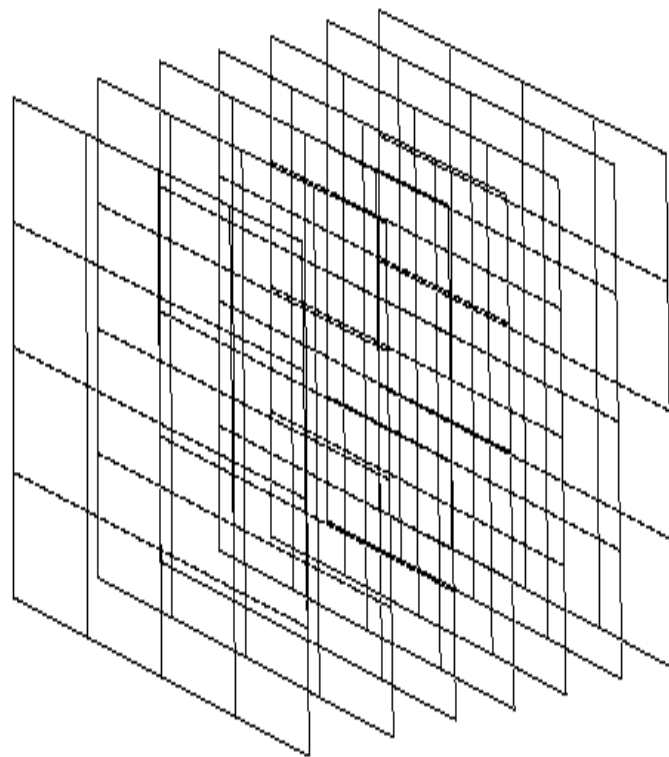
Types of Surfaces

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

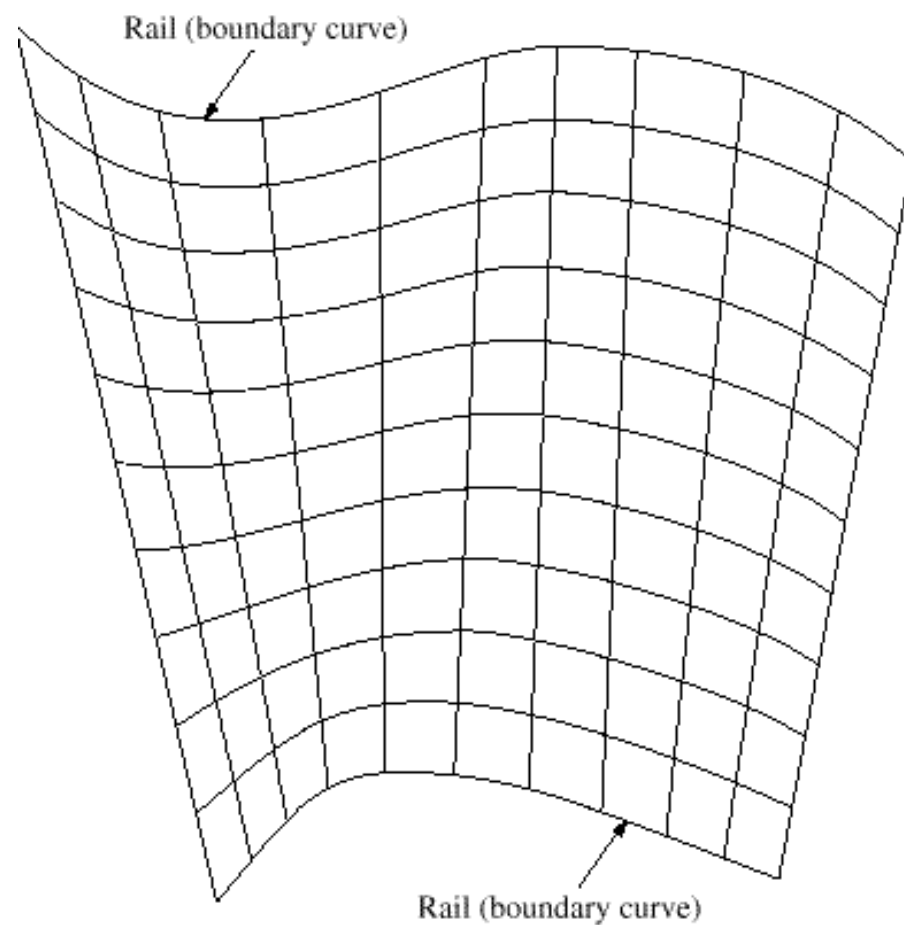


Figure 7.3 Plane surface.

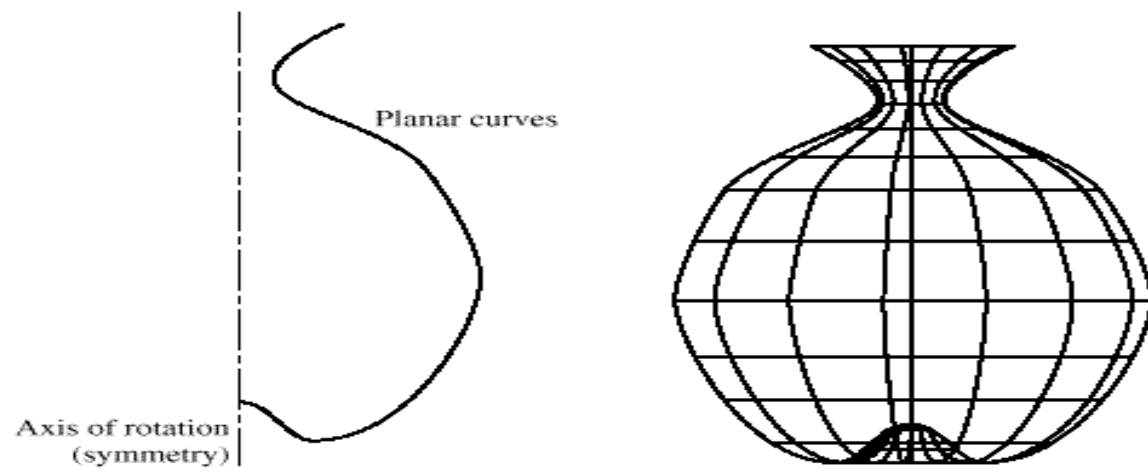


Figure 7.5 Surface of revolution.

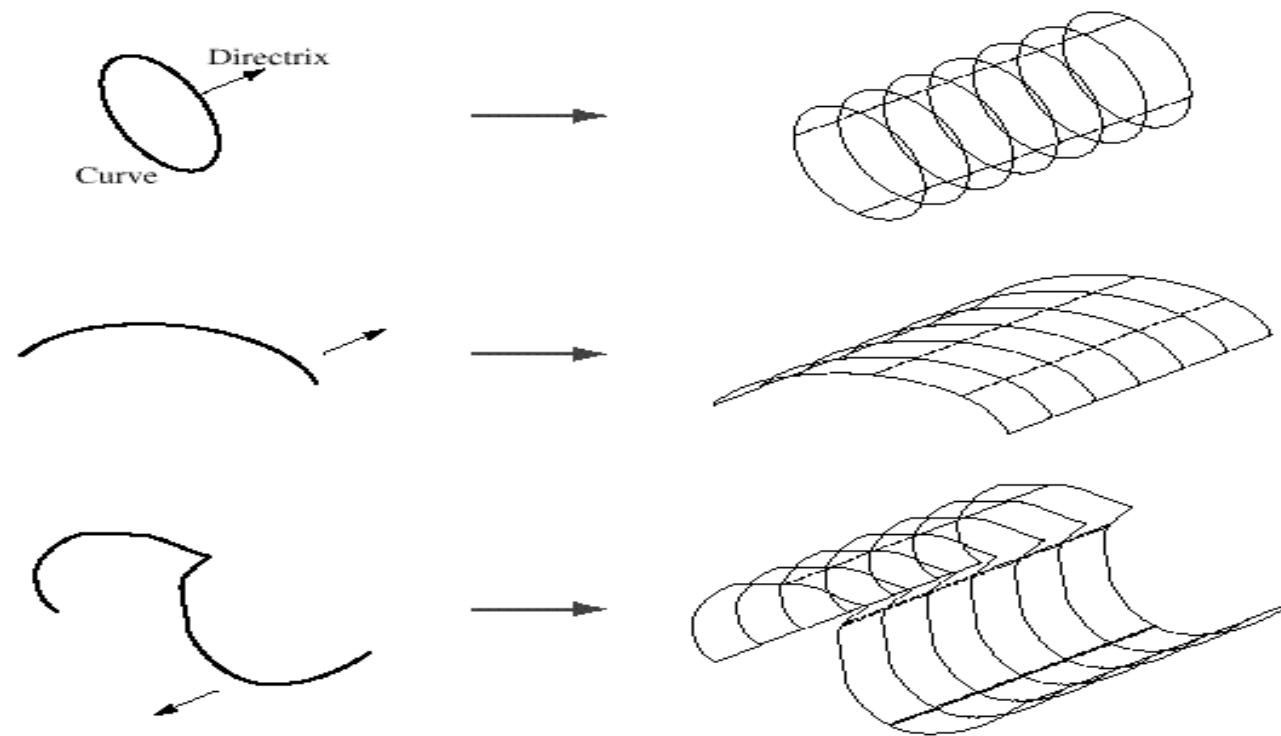


Figure 7.6 Tabulated surface.

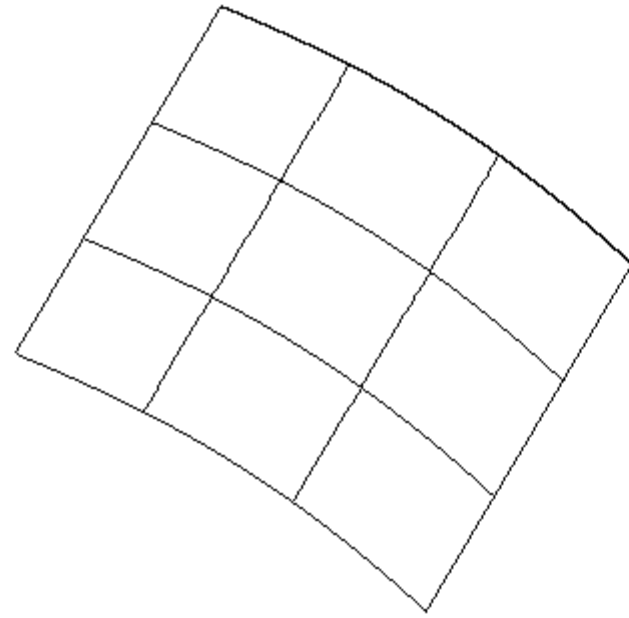
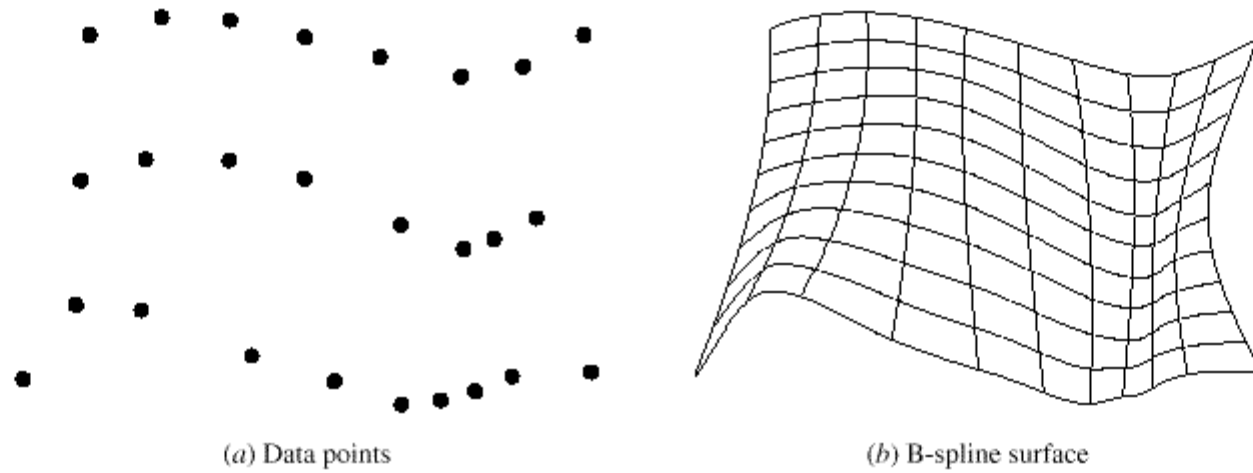


Figure 7.7 Bezier surface.



(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 but 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 but 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 be created using a cylinder command and the other one can be 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.

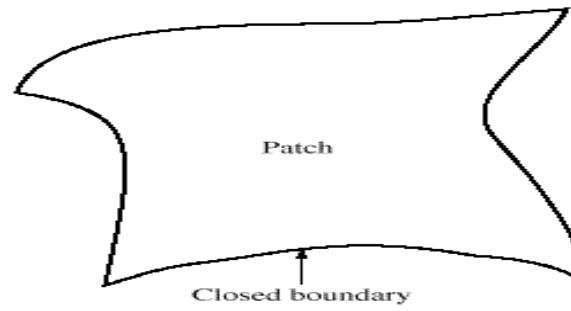


Figure 7.9 Coons surface.

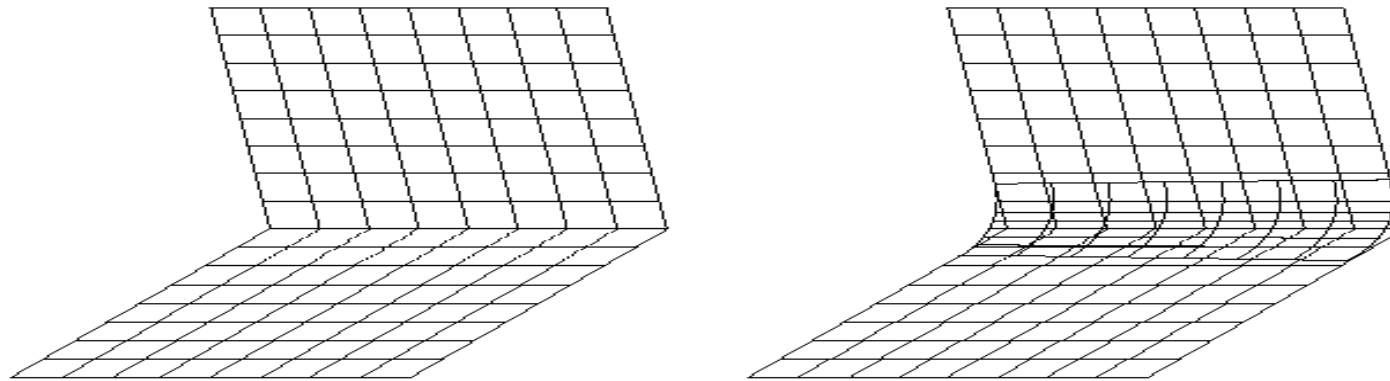


Figure 7.10 Fillet surface.

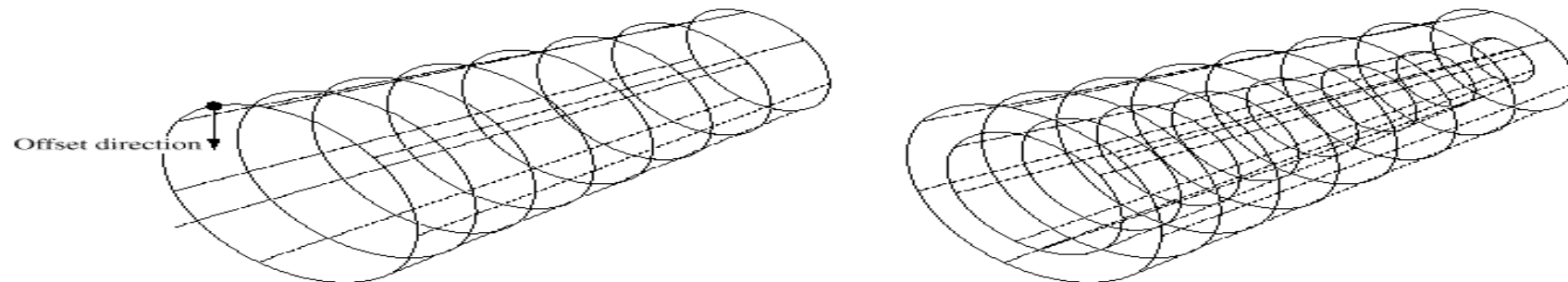
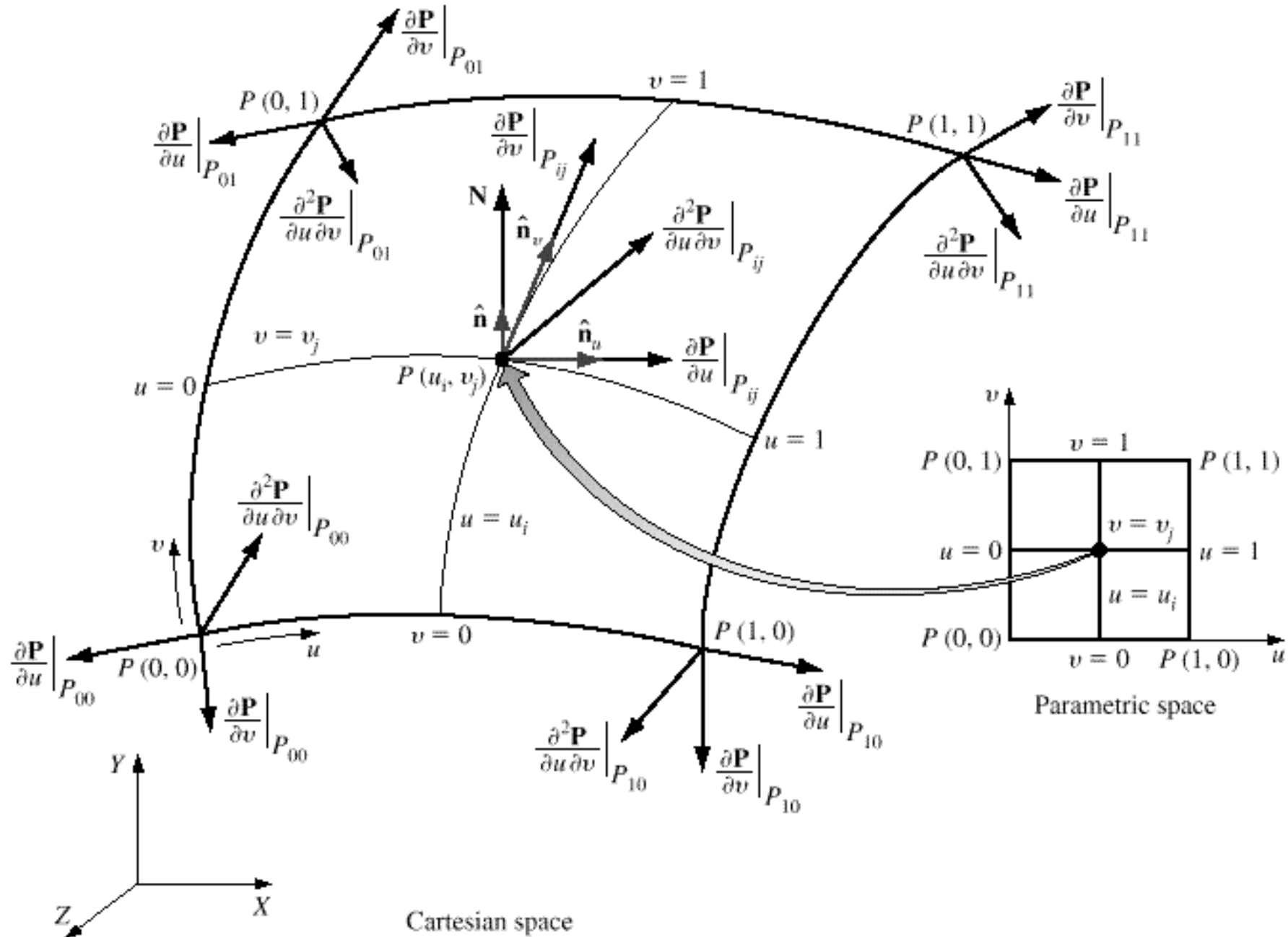


Figure 7.11 Offset surface.

Representation of Curve in Cartesian space :



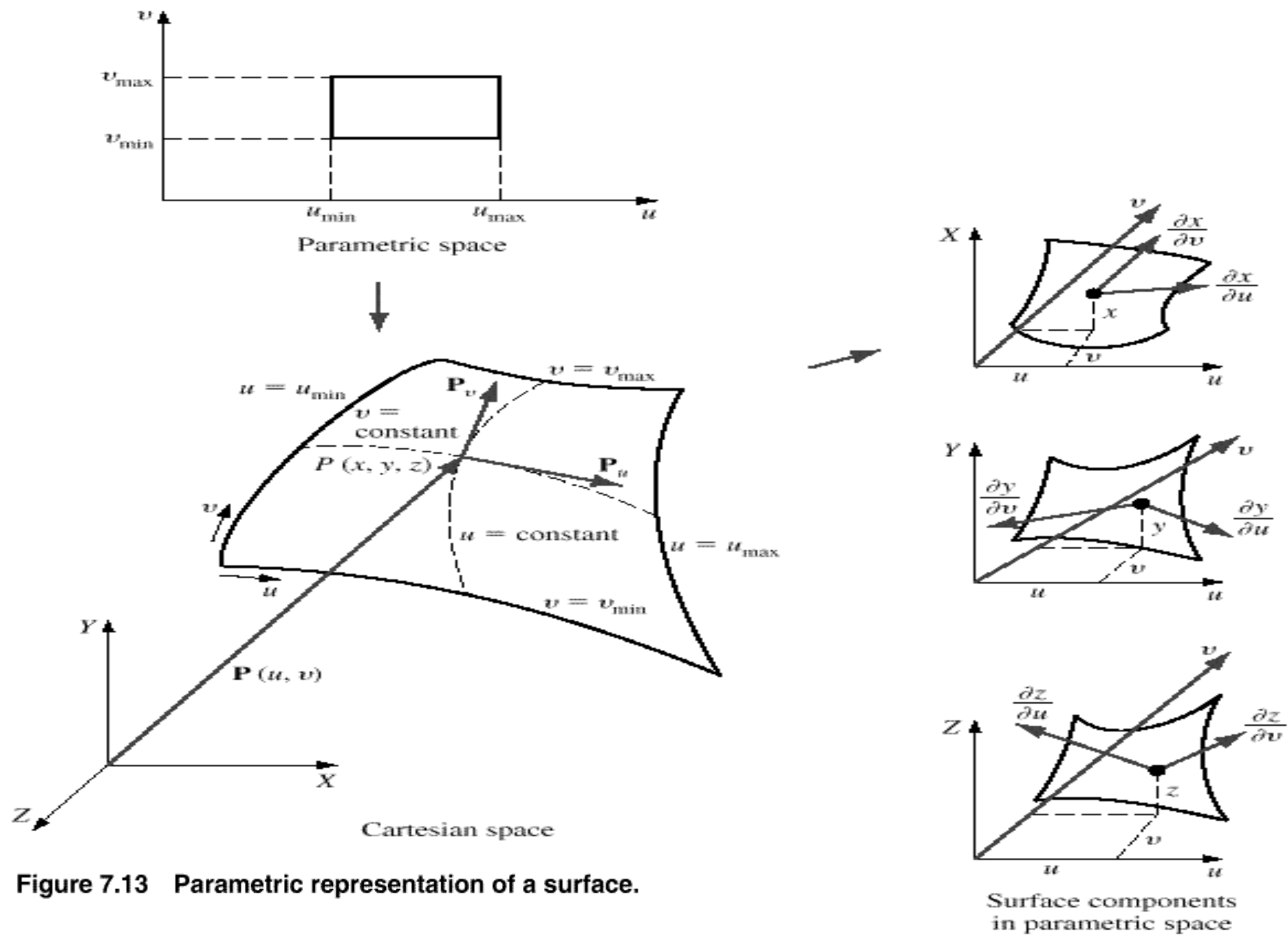


Figure 7.13 Parametric representation of a surface.

Reverse Engineering

Introduction:

In many areas of industry, it is desirable to develop products by digitizing conceptual clay models.

(create geometric models of the physical objects)

This is also referred as reverse engineering of shape/ shape synthesis from digitized point cloud.

The existence of a computer model -

Tool to improve quality and efficiency of design, manufacture and analysis.

Reverse Engineering is the process of producing design details in the form of CAD model from the physical part. The process consists of first scanning the object concerned to create a point cloud that represents the skin of the object. Then using a CAD Software such as SURFACER the skin i.e. the surface model of the object is created. The surface model thus created is exported to a Solid modeling package such as I-DEAS where the surface model is checked for water tight geometry and after modification if necessary, the solid model is generated. The solid model thus created can be then used for analysis, documentation, NC code generation or for prototyping.

APPLICATION:

Copy of part - no original design.

Existing part - analysis and modification.

Aesthetic design - wooden or clay model is first conceptualized.

Custom fits for human surfaces - helmets, shoes, space suits or prostheses.

Manufacturing.

Inspection and quality control.

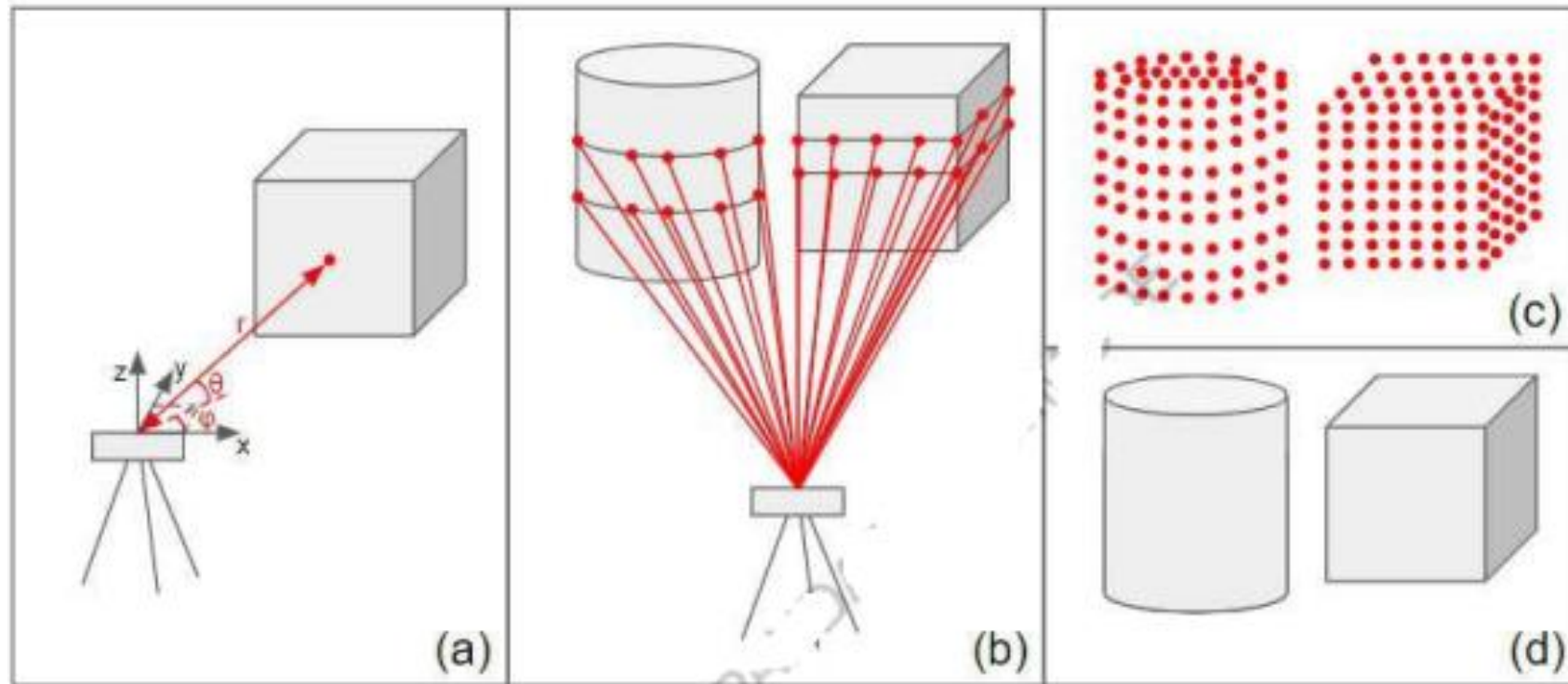
Art & Sculpting

- **Advantages:**
- Cost saving for developing new products.
- Lesser maintenance cost.
- Quality improvement.
- **RE Process:**
- Digitization of the object/ Data Capturing (using CMM, scanners etc.)
- Processing of measured data.
- Creation of CAD model
- Prototype
- **Disadvantages:**
- The large amount of surface data generated from the fast 3D digitizers is not easily modeled by current CAD/CAM systems. Inadequate data reduction methods are not only time consuming, but lose surface characteristic points.
- Consequently, the reconstructed surfaces could lose their true shape in the design cycle. The time required for reprocessing in this case would be unacceptable to the manufacturing industry and thereby be unusable for further steps.

STEPS FOR RE

- Data Acquisition
- Preprocessing
- Segmentation and Surface Fitting
- CAD Model creation

Point Cloud data



Point cloud data:

- A point cloud is a set of data points in space. Point clouds are generally produced by 3D scanners, which measure many points on the external surfaces of objects around them.
- For industrial metrology or inspection using industrial computed tomography, the point cloud of a manufactured part can be aligned to an existing model and compared to check for differences.
- Geometric dimensions and tolerances can also be extracted directly from the point cloud.
- While point clouds can be directly rendered and inspected, point clouds are often converted to polygon mesh or triangle mesh models, NURBS surface models, or CAD models through a process commonly referred to as surface reconstruction.
- Point clouds can also be used to represent volumetric data, as is sometimes done in medical imaging. Using point clouds, multi-sampling and data compression can be achieved.

- **Key characteristics of file formats that are evaluated for optimal suitability include:**

- 1) Compactness of stored data
- 2) Accessibility of the data by third-party developers
- 3) Length of record for each point, line, or face that is being stored
- 4) Speed of loading the point clouds into memory
- 5) Ability to load only a known cloud subsection into memory
- 6) Storing point values specifically, or the possibility to simplify the data to lines and planes
- 7) The ability to store large floating-point values with high precision, without using 64-bit or 128-bit storage for each bit of data associated with each point, line, or plane
- 8) Other possible characteristics

Common point cloud file formats in detail

- **OBJ:** first developed by Wave front technologies, the format has been adopted by a wide range of 3D graphics applications. These include Bentley Systems, Reality Capture and Trimble. It is a simple data format that only represents 3D geometry, normal, color and texture. It is commonly ASCII, however there are some proprietary binary versions of OBJ.
- **PLY:** known as the polygon file format or Stanford triangle format, PLY was inspired by OBJ and purpose-built to store 3D data. PLY uses lists of nominally flat polygons to represent objects. The goal was to add extensibility capabilities and the ability to store a greater number of physical elements. The result is a file format capable of representing color, transparency, surface normals, texture, coordinates and data confidence values. There are two versions of this file, one in ASCII and the other binary.

- **XYZ:** is a non-standardized set of files based on Cartesian coordinates ('x' 'y' and 'z'). XYZ is an archetypal ASCII file type, conveying data in lines of text. There are no unit standardizations for XYZ files. Although there is wide compatibility across programs for this type of file, the lack of standardization surrounding units and specifications makes it a fundamentally faulty method of data transfer unless additional information is supplied.
- **PCG, RCS, RCP:** are all file formats developed by Autodesk to specifically meet the demands of their software suite. RCS and RCP are newer. Autodesk products, however, are often able to convert some open formats, such as PTS into PCG files.
- **E57:** is a vendor-neutral file format for point cloud storage. It can also be used to store images and metadata produced by laser scanners and other 3D imaging systems. It is compact and widely used. It also utilizes binary code in tandem with ASCII, providing much of the accessibility of ASCII and the speed of binary. E57 can represent normals, colors and scalar field intensity.
- In addition to the listed file types, most of these systems are also capable of exporting common raster files such as JPEG or GeoTIFF.

- **Quality issues in Point cloud data**

- **Data loss:** When you move data within the cloud, or between the cloud and on-premise infrastructure (if you choose to do that), you run the risk of formatting problems, data loss, inaccurate timestamps and other issues that undercut data quality. For example, if you move block data from a virtual server disk into a cloud-based file-storage service, formatting differences could cause data quality problems. Or data could be damaged while being transferred over the network.
- **Data Size:** Cloud data can become very big, fast. The fact that the cloud is so scalable makes it easy to store huge volumes of information in the cloud. The more data you have, the harder it can be to maintain data quality.
- **Changes in cloud tools:** Cloud services are always changing and being updated
 - and unlike software that you set up and manage yourself on-premise, cloud-based tools may not always notify you when they are modified. Changes to your cloud-based tools can cause data quality issues if, for example, a tool modifies the way it structures data and your other tools are not configured to handle the new format.
- **Data Formats:** Every new device on the market can present data in a new form. When you are registering and processing new scans into a dataset, you now need to bring together data in different formats from different devices. This is not always an easy task.

- **Massive data:** Point clouds can now scan huge assets more cheaply. New devices also offer broader coverage from a single scan. This creates tens of thousands of data points. These are cumbersome to handle and harmonies across your entire enterprise. It can also be difficult to share massive data between project partners.
- **Interoperability:** You invest considerable amounts of time and money in selecting the right engineering software and training your team to use it. So, your new technology needs to fit with your existing software set so that makes it easier for your teams to work together. In practice, this kind of integration is not always easy.
- **Accessibility:** Everybody involved in the lifecycle of an asset can benefit from access to point cloud data. But fragmented datasets in different formats can make this difficult, as I have explained above.
- **Ownership:** Who owns point cloud data? In the past, EPCs and the contractors who capture the data become custodians of the information. But asset operators can use point cloud data throughout the asset lifecycle: for safety, training, asset visualization and even for Brownfield projects with new contractors.

Requirements for conversion of surface models into solid models

- 1) 3D model reconstruction: It refers to the process of reconstructing a 3D model of an object based on point cloud data.
 - a) Reconstruction of geometry model.

It contains the geometric information. Therefore, the reconstruction process is mainly to process the original sensing data (e.g., laser scans, images, videos, etc.) to obtain a complete point cloud.
 - b) Reconstruction of semantic model.

It not only contains geometric information, but also contains object-based semantic information. Hence, the reconstruction process also involves the recognition of objects from point cloud data and the enrichment of semantic information.
- 2) Geometry quality inspection:
 - a) dimensional quality inspection
 - b) surface quality inspection
 - c) Displacement inspection.
- 3) Point cloud data acquisition and processing:
 - a) Determination of required data quality
 - b) Determination of data acquisition parameters
 - c) Extraction and utilization of semantic information
 - d) Platforms for data visualization and processing

Applications of Point Cloud Data:

- 1) Construction industry for two major applications including 3D model reconstruction and geometry quality inspection.
- 2) Construction progress tracking: Steel structural components, earthwork, concrete structural components, fake structures simulating columns, concrete construction secondary and temporary objects.
- 3) Building performance analysis: Mechanical analysis of bridge structure, morphologic analysis, stress estimation, building accessibility diagnosis, energy performance modeling, and structural analysis of bridges such as finite element modeling.
- 4) Construction safety management: Active safety management, safety hazards identification, identification of blind spots of construction equipment, safety hazard simulation, and safety assistance for mobile crane lifting operations.
- 5) Building renovation: Re-design and renovation of the façades, interior design, energy performance retrofit, refurbishment.
- 6) Construction automation: Digital reproduction, automated earthwork excavation.

- 7) Heritage applications: Surveying, repair and maintenance, stability analysis, heritage site preservation, documentation.
- 8) Robot navigation: Navigation in indoor and outdoor environments, navigation in indoor environment.
- 9) Creating or updating geographical maps with height information of large areas scanned from airplanes or satellites.
- 10) Providing a reference to create CAD models from real-world objects.
- 11) Automatic quality inspection by comparing a computer model with the reality:
 - a) Follow the different steps during production.
 - b) Inspect a finished product for errors.
 - c) Check deformations or other wear over time.
- 12) Simulation: Point-clouds can be used to represent solid objects in a Finite Element Analysis context -- boiling down mathematically complex CAD surfaces into a relatively finite number of points. This allows engineers and scientists to simulate objects under stress, simulate deformation, etc.



Thank You
For Your Attention