Unit – 5

THREE DIMENSIONAL GRAPHICS SYSTEM

What are the issue in 3d that makes it more complex than 2d?

- **Depth perception:** 3D requires techniques like stereoscopic vision or depth mapping for realistic perception of depth and distance.
- **Rendering and visualization:** 3D rendering involves complex algorithms for lighting, shadows, reflections, and other effects, demanding more computational power.
- **Additional degrees of freedom:** In 3D, objects have an extra dimension along the Z-axis, adding complexity to object manipulation and spatial relationships.
- **Increased data complexity:** 3D objects have additional attributes like rotation and scale, increasing the complexity of data representation.
- **User interaction:** Interacting with 3D environments requires complex gestures or specialized input devices compared to simpler 2D interfaces.
- **Computational demands:** 3D graphics and simulations require more computational resources due to the complexity of rendering and physics calculations.

2D vs 3D

2D	3D
Objects are represented on a flat surface with no depth perception.	Objects have depth, allowing for realistic perception of distance and depth.
Rendering involves simpler algorithms for shapes and colors.	Rendering requires complex algorithms for lighting, shadows, reflections, and other effects.
Objects move and interact along two axes (X and Y).	Objects have an additional degree of freedom along the Z-axis, representing depth or height.
Data representation is simpler, often consisting of position and size.	Data representation is more complex, including attributes like rotation, scale, and orientation.
Interacting with objects is typically done through simple gestures or clicks.	Interacting with objects requires complex gestures or specialized input devices for control in 3D space.
User interfaces are simpler and typically involve a flat screen or surface.	User interfaces involve immersive displays or virtual reality environments to navigate and interact in 3D space.
Computational demands are relatively lower due to simpler graphics and calculations.	Computational demands are higher due to complex rendering, physics simulations, and spatial calculations.
Limited representation of the real world with less depth and realism.	Provides a more immersive and realistic representation of the real world with depth, lighting, and other effects.

3D transformation

- 3D transformations are extended from 2D transformation by including the consideration for the Z-coordinates.
- To represent the 3D object, we need 3 parameters i.e.

  X-coordinates representing the Width
  Y-coordinates representing Height and
  Z-coordinates representing Depth.
- Instead of just TWO axes, now in 3D we have THREE axes i.e. X, Y and Z.

Matrix Representation of 3D Transformations

- 2D transformations can be represented by 3×3 matrices using homogenous coordinates.
- 3D transformations can be represented by 4 × 4 matrices, providing we use homogeneous coordinate representations of points in 2 space as well.

  Thus instead of representing a point as (x, y, z), we represent if as (x, y, z, H), where two these quadruples represent the same point it one is a non zero multiple of the other the quadruple (0, 0, 0, 0) is not allowed.

- A standard representation of a point (x, y, z, H) with H not zero is given by : (x/H, y/H, z/H, 1).
- Transforming the point to this form is called homogenizing.

1. **Translation**

- A point P(X, Y, Z) is translated to P'(X', Y', Z').
- **Equation form:**

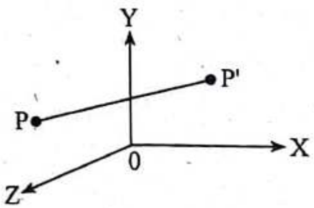
$X' = X + T_x$ (This denotes translation towards X axis)

$Y' = Y + T_y$ (This denotes translation towards Y axis)

$Z' = Z + T_z$ (This denotes translation towards Z axis)

- **In matrix form (Homogenous Co-ordinates):**

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



2. **Scaling**

- A point P(X,Y,Z) is scaled to P'(X',Y',Z').
- **Equation form:**

$X' = X \times S_x$ (This denotes scaling towards X axis)

$Y' = Y \times S_y$ (This denotes scaling towards Y axis)

$Z' = Z \times S_z$ (This denotes scaling towards Z axis)

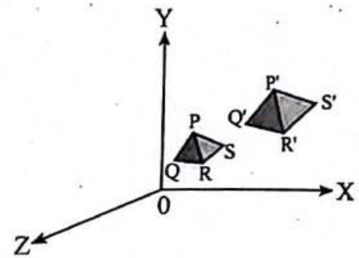
- **In matrix form (Homogenous Co-ordinates):**

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Scaling changes size of an object and repositions the object relative to the co-ordinate origin.

If transformation parameters are not all equal then figure gets distorted.

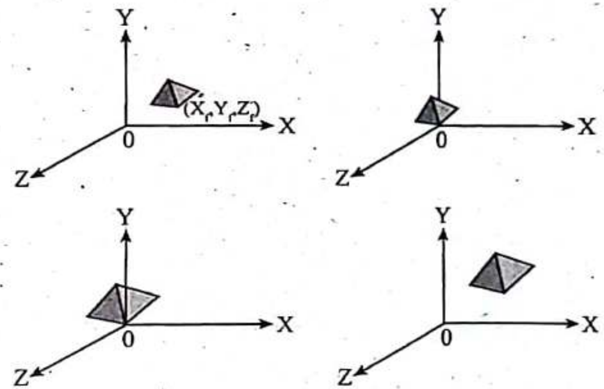
So we can preserve the original shape of an object with uniform scaling ($S_x - S_y - S_z$)



Fixed point scaling

Steps:

1. Translation of an object to that the fixed point about which an object is to be scaled coincides with origin.
2. Scaling of an object about the origin.
3. Inverse translation of an object so that the fixed point reaches to its original position.



$$C_m = T^{-1}{}_{(T_x, T_y, T_z)} S_{(S_x, S_y, S_z)} T_{(-T_x, -T_y, -T_z)}$$

$$= \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & -T_y \\ 0 & 0 & 1 & -T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & 0 & (1-S_x)T_x \\ 0 & S_y & 0 & (1-S_y)T_y \\ 0 & 0 & S_z & (1-S_z)T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, P' = C_m.P

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & (1-S_x)T_x \\ 0 & S_y & 0 & (1-S_y)T_y \\ 0 & 0 & S_z & (1-S_z)T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

3. **Rotation**

- To generate a rotation transformation for an object, we must designate an axis of rotation (about which the object is to be rotated) and the amount of angular rotation.
- Unlike 2-Dimensional application where all the transformations are carried out in the XY plane, a 3-Dimensional rotation can be specified around any line in space.
- The easiest rotation axes to handle are those that are parallel to the co-ordinate axes.

a. **Rotation about X-axis:**

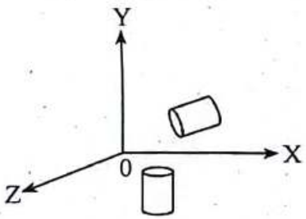
- **In equation form:**

X' = X

Y' = Ycosθ – Zsinθ

Z' = Ysinθ + Zsinθ

- **In matrix form (Homogenous Co-ordinates):**



$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{ anti-clockwise direction}$$

or, P' = Rx(θ).P

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{ clockwise direction}$$

b. **Rotation about Y-axis:**

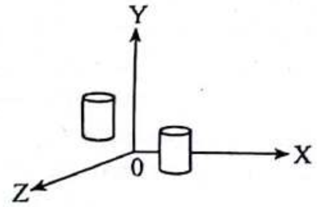
- **In equation form:**

X' = Zsinθ + Xcosθ

Y' = Y

Z' = Zcosθ – Xsinθ

- **In matrix form (Homogenous Co-ordinates):**



$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{ anti-clockwise direction}$$

or, P' = R_y(θ) . P

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{ clockwise direction}$$

c. **Rotation about Z-axis:**

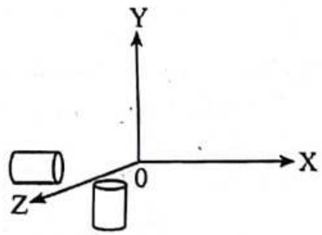
- **In equation form:**

X' = Xcosθ – Ysinθ

Y' = Xsinθ + Ycosθ

Z' = Z

- In matrix form



$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{ anti-clockwise direction}$$

or, $P' = R_z(\theta) . P$

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{ clockwise direction}$$

How it actually form

Coordinate axes Rotations:

2D z-axis rotation equations are easily extended to 3D:

$x' = x\cos\theta - y\sin\theta$

$y' = x\sin\theta + y\cos\theta$

$z' = z$ ....(i)

Cyclic permutation of the coordinate parameters x, y and z are used to get transformation equations for rotations about the other two coordinates

$x \rightarrow y \rightarrow z \rightarrow$ so,

substituting permutations in (i) for an x axis rotation we get,

$y' = y\cos\theta - z\sin\theta$

$z' = y\sin\theta + z\cos\theta$

$x' = x$

substituting permutations in (i) for a y axis rotation we get,

$z' = z\cos\theta - x\sin\theta$

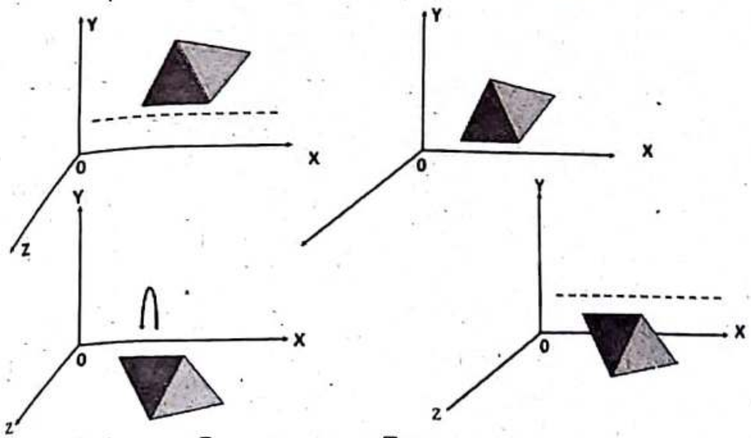
$x' = z\sin\theta + x\cos\theta$

$y' = y$

Case 1: Rotation of an object about an axis that is parallel to one of co-ordinate axes.

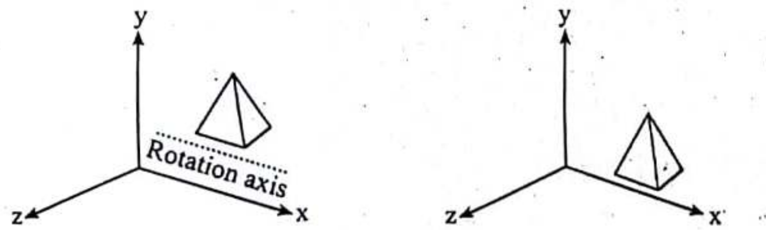
We can attain the desired rotation with the following transformation sequence.

1. Translate the object so that the rotation axis coincides with one of the co-ordinate axes.
2. Performed the specified rotation about that axis.
3. Translate the object so that the axis of rotation is moved back to its original position.



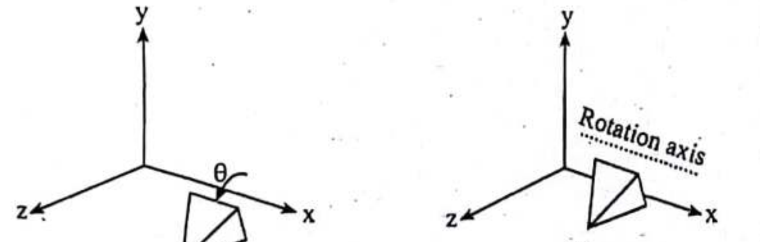
$C_m = T^{-1}(T_x, T_y, T_z) R_{(x\text{-axis(anticlockwise)})} T(-T_x, -T_y, -T_z)$

$$= \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & -T_y \\ 0 & 0 & 1 & -T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



(a) Original position or object

(b) Translate rotation axis onto x-axis



(c) Rotation object through angle θ

(d) Translate rotation axis to original position

$P' = T^{-1} . R_x(\theta) . T . P .$

Case 2: Rotation of an object about an axis that is not parallel to one of the co-ordinate axes.

We can attain the desired rotation with the following transformation sequence.

1. Translate the object so that the rotation axis passes through the origin.
2. Rotate the object so that the axis of rotation coincides with one of the co-ordinate axes.
3. Performed the specified rotation about that axis.
4. Inverse rotation to step 2 to bring the rotation axis back to its original orientation.
5. Translate the object so that the axis of rotation is moved back to its original position.

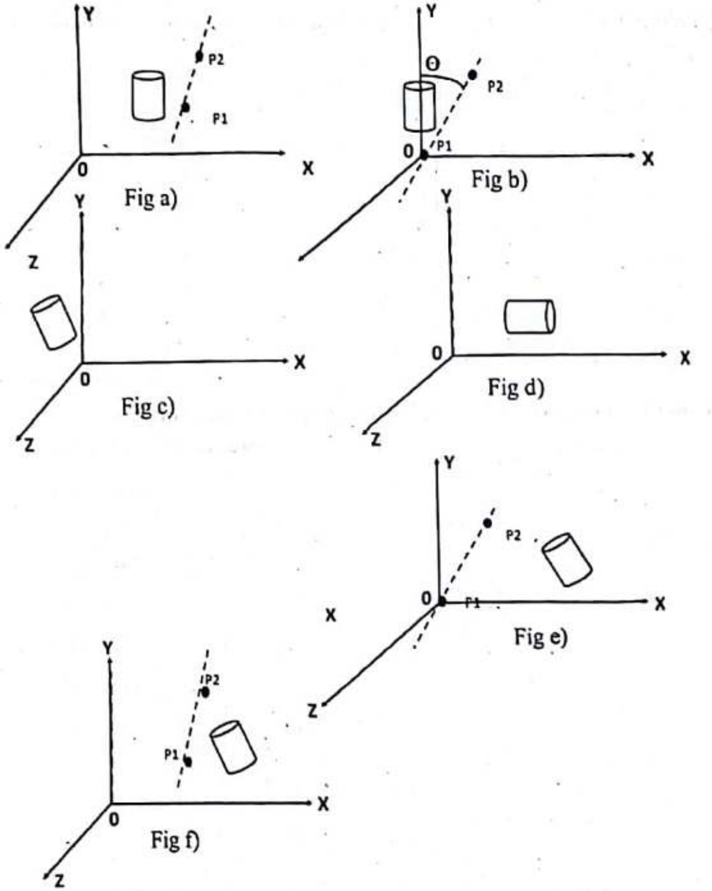


Fig a)

Fig b)

Fig c)

Fig d)

Fig e)

Fig f)

Rotation about y axis CCW

$z' = z\cos\theta - x\sin\theta$

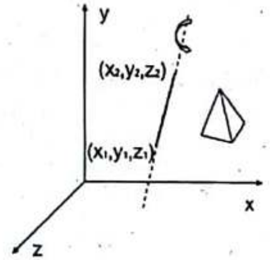
$x' = z\sin\theta + x\cos\theta$

$y' = y$

$x \rightarrow y, \rightarrow z \rightarrow x$

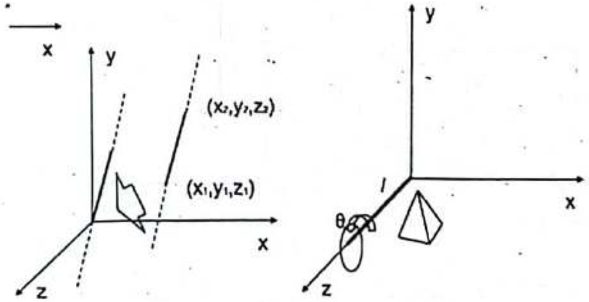
$$\begin{bmatrix} z' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

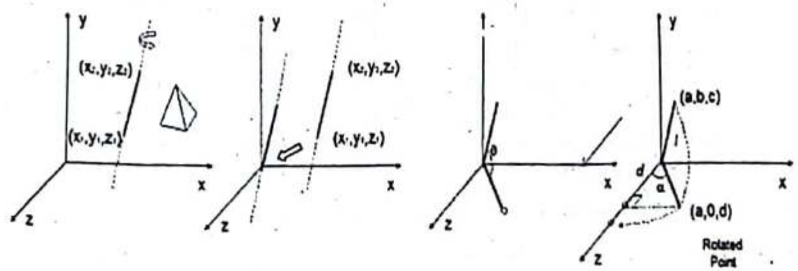


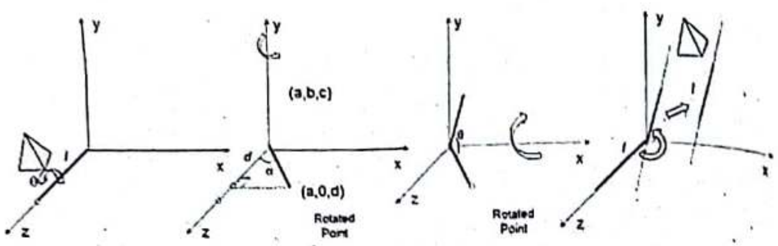
Steps

1. Translate (x_1, y_1, z_1) to the origin
2. Rotate (x'_2, y'_2, z'_2) on to the z-axis
3. Rotate the object around the z-axi
4. Rotate the axis to the original orientation
5. Translate the rotation axis to the original position



Rotation about an Arbitrary Axis



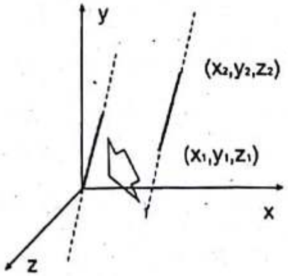


$$Cm = T'_{(x_1,y_1,z_1)} . R'_x . R'_y . R_z . R_y . R_x . T_{(-x_1,-y_1,-z_1)}$$

Explanation: Rotation about an Arbitrary Axis

Step 1

Translate (x_1, y_1, z_1) to the origin

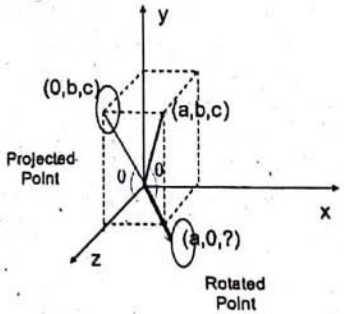


$$T = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

Step 2

Rotate the line about x axis in anti clockwise direction by θ' angle



$$\sin\theta = \frac{b}{\sqrt{b^2 + c^2}} = \frac{b}{d}$$

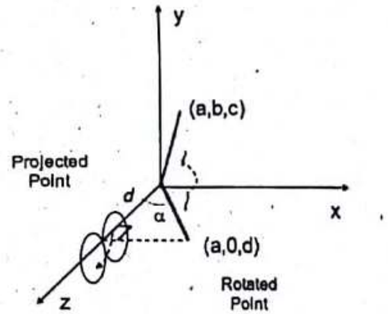
$$\cos\theta = \frac{c}{\sqrt{b^2 + c^2}} = \frac{c}{d}$$

$$[R_{x_\theta}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

Step 3

Rotate the line about y axis in clockwise direction by α' angle



$$\sin\alpha = \frac{a}{l}, \cos\alpha = \frac{d}{l}$$

$$l^2 = a^2 + b^2 + c^2 = a^2 + d^2$$

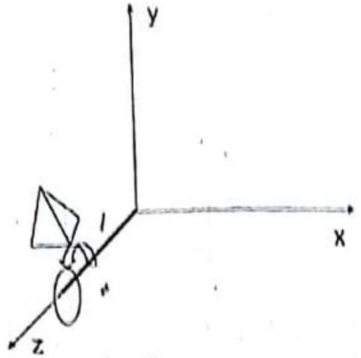
$$d = \sqrt{b^2 + c^2}$$

$$[R_{y\alpha}] = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} d/l & 0 & -a/l & 0 \\ 0 & 1 & 0 & 0 \\ a/l & 0 & d/l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

Step 4

Rotate the object about the line that has been aligned with z-axis by the desired angle 'ϕ'



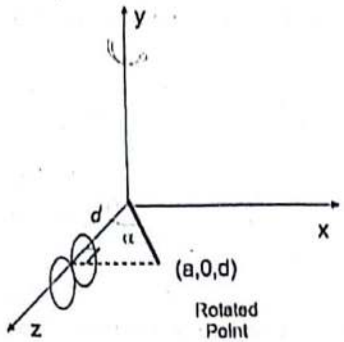
$$[R_{z\phi}] = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

Step 5

(Apply reverse transformations to place the arbitrary axis along with the reflected object back to its original position)

Rotate the line about y axis in anti-clockwise direction

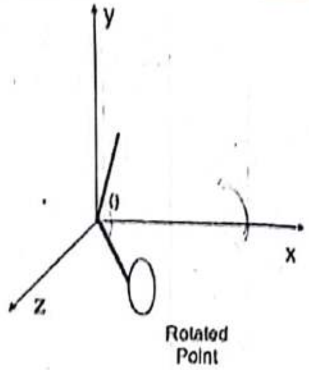


$$[R'_{y\alpha}] = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} d/l & 0 & a/l & 0 \\ 0 & 1 & 0 & 0 \\ -a/l & 0 & d/l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

Step 6

Rotate the line about x axis in clockwise direction

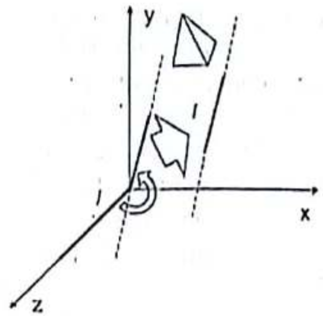


$$[R'_{x\theta}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & b/d & 0 \\ 0 & -b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

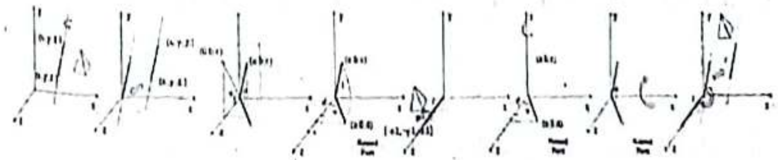
Step 7

PERFORM ANTI translate



$$T' = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis



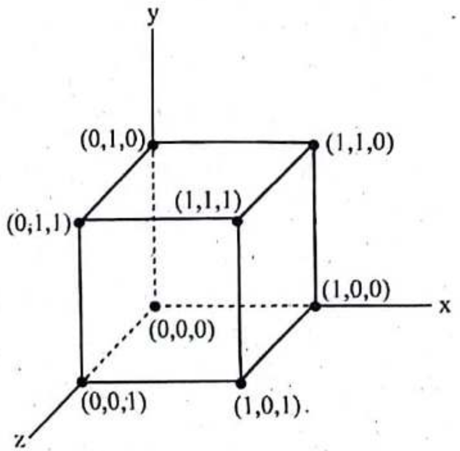
$$CM = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

How will you rotate a unit cube 90° bout an axis defined its endpoints A(2,1,0) and B(3,3,1).

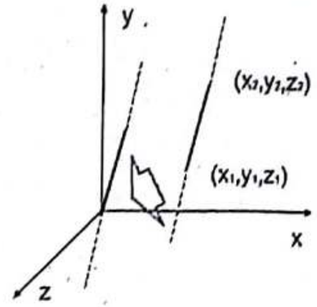


$$cm = T'_{(x_1,y_1,z_1)} \,.\, R'_x \,.\, R'_y \,.\, R_z \,.\, R_y \,.\, R_x \,.\, T_{(-x_1,-y_1,-z_1)}$$

How will you rotate a unit cube 90° about an axis defined by its endpoints A(2,1,0) and B(3,3,1).

Step 1

Translate end point A(2,1,0) to the origin

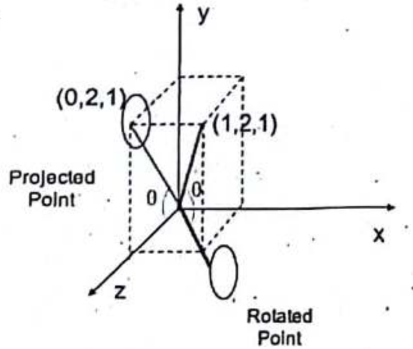


$$T = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

How will you rotate a unit cube 90° about an axis defined by its endpoints A(2,1,0) and B(3,3,1).

Step 2

Rotate the line about x axis in anti clockwise direction by θ' angle



$$\sin\theta = \frac{2}{\sqrt{2^2 + 1^2}} = \frac{2}{\sqrt{5}}$$

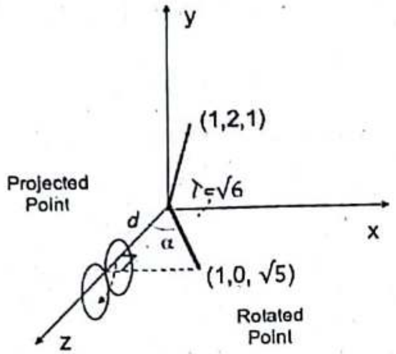
$$\cos\theta = \frac{1}{\sqrt{2^2 + 1^2}} = \frac{1}{\sqrt{5}}$$

$$[R_{x\theta}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

Step 3

Rotate the line about y axis in clockwise direction by α' angle



$l = \sqrt{6}$ $d = \sqrt{5}$

$\sin\alpha = \dfrac{1}{\sqrt{6}}, \cos\alpha = \dfrac{\sqrt{5}}{\sqrt{6}}$

$l^2 = a^2 + b^2 + c^2 = a^2 + d^2$

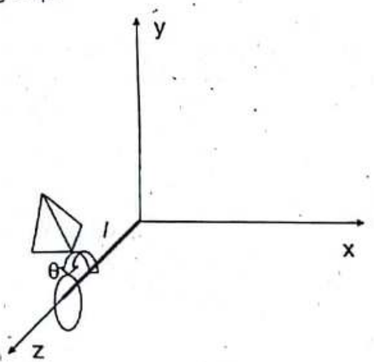
$d = \sqrt{b^2 + c^2} = \sqrt{5}$

$$[R_{y\alpha}] = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} d/l & 0 & -a/l & 0 \\ 0 & 1 & 0 & 0 \\ a/l & 0 & d/l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

Step 4

Rotate the object about the line that has been aligned with z axis by the desired angle 'ϕ'.

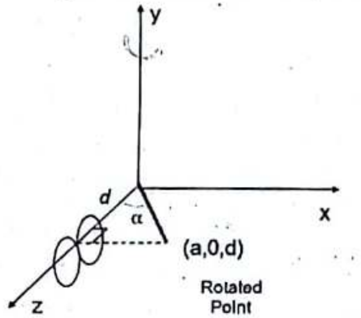


$$[R_{z\phi}] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 5

(Apply reverse transformations to place the arbitrary axis along with the reflected object back to its original position)

Rotate the line about y axis in anti clockwise direction

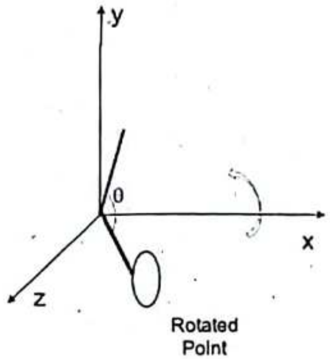


$$[R'_{y\alpha}] = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} d/l & 0 & a/l & 0 \\ 0 & 1 & 0 & 0 \\ -a/l & 0 & d/l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an Arbitrary Axis

Step 6

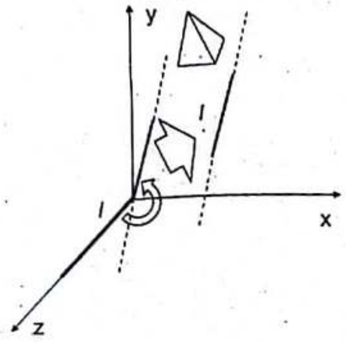
Rotate the line about x-axis in clockwise direction



Rotation about an Arbitrary Axis

Step 7

Rotation the line about y axis in clockwise direction



$$T' = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. **Shear**

I. **Shearing in XY-direction keeping Z-coordinate same (along z-axis)**

- **In equation form:**

$X' = X + Sh_x \times Z$

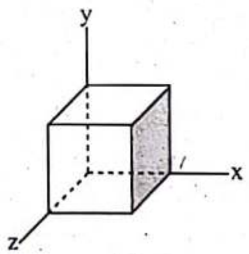
$Y' = Y + Sh_y \times Z$

$Z' = Z$

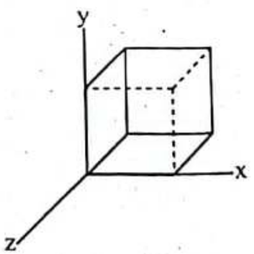
- **In matrix form (Homogenous co-ordinates):**

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & Sh_x & 0 \\ 0 & 1 & Sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

i.e. $P' = Sh_{xy} . P$



(a)



(b)

II. **Shearing in YZ-direction keeping X-coordinate same (along x-axis)**

- **In equation form:**

$X' = X$

$Y' = Y + Sh_y \times X$

$Z' = Z + Sh_z \times X$

- **In matrix form (Homogenous co-ordinates):**

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ Sh_y & 1 & 0 & 0 \\ Sh_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

i.e. $P' = Sh_{yz} . P$

III. **Shearing in XZ-direction keeping Y-coordinate same (along Y-axis)**

- **In equation form:**

$X' = X + Sh_x \times Y$

$Y' = Y$

$Z' = Z + Sh_z \times Y$

- **In matrix form (Homogenous co-ordinates):**

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & Sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

i.e. $P' = Sh_{xz} . P$

5. **Reflection:**

a. **Reflection about X-axis/YZ-plane**

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

b. **Reflection about Y-axis/XZ-plane**

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

c. **Reflection about Z-axis/XY-plane**

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Solved Numerical Examples

1. **Perform rotation of a line (10,10,10), (20,20,15) about Y-axis in clock wise direction by 90 degree.**

Solution:

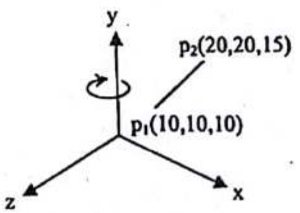
Rotation about y-axis in clock wise direction

$z' = z\cos\theta + x\sin\theta$

$x' = -z\sin\theta + x\cos\theta$

$y' = y$

$\theta = 90°$



In matrix form,

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

$$\begin{bmatrix} x_1' \\ y_1' \\ z_1' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos 90 & 0 & -\sin 90 & 0 \\ 0 & 1 & 0 & 0 \\ \sin 90 & 0 & \cos 90 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 10 \\ 10 \\ 1 \end{bmatrix}$$

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

So, $(x_1', y_1', z_1') = (-10,10,10)$

$$\begin{bmatrix} x_2' \\ y_2' \\ z_2' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 20 \\ 20 \\ 15 \\ 1 \end{bmatrix} = \begin{bmatrix} -15 \\ 20 \\ 20 \\ 1 \end{bmatrix}$$

So, $(x_2', y_2', z_2') = (-15,20,20)$

2. **Reflect the object (0,0,0), (2,3,0) and (5,0,4) about the plane y = 4.**

Solution:

Steps to reflect the object:

i. Translate the plane so that the plane coincides with the xz plane
ii. Perform the xz reflection
iii. Translate the object so that reflection plane is moved back to its original position.

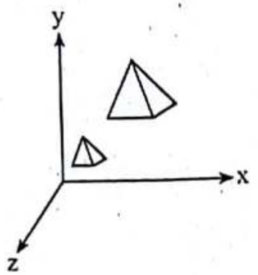
$$C.M. = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$P' = C.M. \times p$

$$\begin{bmatrix} x_1' \\ y_1' \\ z_1' \\ 1 \end{bmatrix} = C.M. \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_2' \\ y_2' \\ z_2' \\ 1 \end{bmatrix} = C.M. \times \begin{bmatrix} 2 \\ 3 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_3' \\ y_3' \\ z_3' \\ 1 \end{bmatrix} = C.M. \times \begin{bmatrix} 5 \\ 0 \\ 4 \\ 1 \end{bmatrix}$$



$P' = sh_{yz} \cdot P$ $\qquad$ $y' = y$

$x' = x + sh_x \cdot y$ $\qquad$ $z' = z + sh_z \cdot y$

Shearing in xz direction keeping y co-ordinate same (along y-axis)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$p' = sh_{xz} \cdot p$ $\qquad$ $y' = y$

$x' = x + sh_x \cdot y$ $\qquad$ $z' = z + sh_z \cdot y$

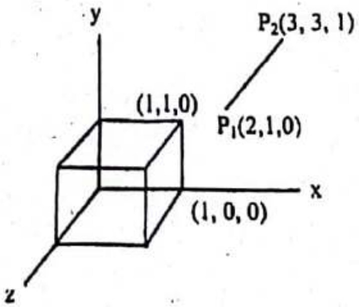
3. **Find the new co-ordinates of a unit cube 90° rotated about an axis defined by its end points A(2,1,0) and B(3,3,1).**

Solution:

$v = P_2 - P_1 = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$

$a = \frac{x_2 - x_1}{|v|} = \frac{1}{\sqrt{6}}$ $\qquad$ $b = \frac{y_2 - y_1}{|v|} = \frac{2}{\sqrt{6}}$

$c = \frac{z_2 - z_1}{|v|} = \frac{1}{\sqrt{6}}$ $\qquad$ $d = \sqrt{b^2 + c^2} = \sqrt{\frac{5}{6}}$



$R(\theta) = T^{-1} R_x^{-1}(\alpha) R_y^{-1}(\beta) . R_z(\theta) . R_y(\beta) . R_x(\alpha) . T$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & b/d & 0 \\ 0 & -b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 1 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos 90 & -\sin 90 & 0 & 0 \\ \sin 90 & \cos 90 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 1 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4472 & 0.894 & 0 \\ 0 & -0.894 & 0.4472 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.9128 & 0 & 0.408 & 0 \\ 0 & 1 & 0 & 1 \\ -0.408 & 0 & 0.9128 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.9128 & 0 & -0.408 & 0 \\ 0 & 1 & 0 & 1 \\ 0.408 & 0 & 0.9128 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4472 & -0.894 & 0 \\ 0 & 0.894 & 0.4472 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4472 & 0.894 & 0 \\ 0 & -0.894 & 0.4472 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.9128 & 0 & 0.408 & 0 \\ 0 & 1 & 0 & 1 \\ -0.408 & 0 & 0.9128 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.9128 & 0 & -0.408 & 0 \\ 0 & 1 & 0 & 1 \\ 0.408 & 0 & 0.9128 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 4472 & -0.894 & -0.4472 \\ 0 & 0.894 & 0.4472 & -0.894 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4472 & 0.894 & 0 \\ 0 & -0.894 & 0.4472 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.9128 & 0 & 0.408 & 0 \\ 0 & 1 & 0 & 1 \\ -0.408 & 0 & 0.9128 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.9128 & -0.3647 & -0.1824 & -1.46 \\ 0 & 0.4472 & -0.894 & -0.447 \\ 0.408 & 0.816 & -0.408 & -1.632 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4472 & 0.894 & 0 \\ 0 & -0.894 & 0.4472 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.9128 & 0 & 0.408 & 0 \\ 0 & 1 & 0 & 1 \\ -0.408 & 0 & 0.9128 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -0.4472 & 0.894 & -0.447 \\ 0.9128 & -0.364 & -0.1824 & -0.146 \\ 0.408 & 0.816 & -0.408 & -1.632 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.4472 & 0.894 & 0 \\ 0 & -0.894 & 0.4472 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.1664 & -0.074 & 0.983 & -0.258 \\ 0.9128 & -0.364 & -0.1824 & -1.46 \\ 0.372 & 0.927 & -0.007 & -1.672 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0.1664 & -0.074 & 0.983 & -0.258 \\ 1.278 & 0.666 & -0.075 & -2.148 \\ -0.381 & 0.74 & 0.166 & 0.558 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1664 & -0.074 & 0.983 & -0.258 \\ 1.278 & 0.666 & -0.075 & -2.148 \\ -0.381 & 0.74 & -0.166 & 0.558 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$p' = c.m.p.$

$$= \begin{bmatrix} 0.1664 & -0.074 & 0.983 & -0.258 \\ 1.278 & 0.666 & -0.075 & -2.148 \\ -0.381 & 0.74 & -0.166 & 0.558 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Let the original coordinates are A(0,1,1), B(1,1,1), C(1,0,1), D(0,0,1), E(1,1,0), F(1,0,0), G(0,1,0), H(0,0,0) then the corresponding new coordinates are A'(1.3718, 0.927, 0.37), B'(1.537, 1.6681, -0.279), C'(1.4308, 0.6381, −1.202), D'(1.2648, −0.103, −0.553), E'(0.92, 2.4721, −0.081), F'(0.813, 1.4421, −1.004), G'(0.647, 0.701, −0.355), H'(0.754, 1.731, 0.568)

4. **A unit length cube with diagonal passing through (0,0,0) and (2,2,2) is shared with respect to zx-plane with share constants = 3 in both directions. Obtain the final coordinates of the cube after shearing.**

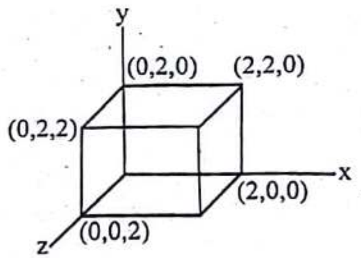
Solution:

Shearing with respect to zx-plane

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$sh_x = sh_y = 3$

A = (2, 0, 0)	B = (2, 2, 0)	C = (0, 2, 0)
D = (0, 2, 2)	E = (0, 0, 2)	F = (0, 2, 2)
G = (2, 2, 2)	H = (0, 0, 0)	



$$A' = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

A' = (2,0,0)

Similarly, we can calculate B', C', D', E', G', H'.

B' = (8,2,6)	C' = (6,2,6)	D' = (6,2,6)
E' = (0,0,0)	F' = (6,2,6)	G' = (8,2,6)
H' = (0,0,0)		

The final coordinates of the cube after shearing are A'(2,0,0), B'(8,2,6), C'(6,2,6), D'(6,2,6), E'(0,0,0), F'(6,2,6), G'(8,2,6), and H'(0,0,0).

5. **List down the steps for rotating a 3D object by 90° in counter clockwise direction about an axis joining end points (1,2,3) and (10,20,30). Also derive the final transformation matrix.**

Solution:

i. Translate (1,2,3) to origin so that the rotation axis passes through the origin.

ii. Rotate the line so that the line coincides with one of the axes, say 2 axis.

iii. Rotate the object about the co-ordinate axis by 90° in counter clockwise direction.

iv. Apply the inverse of step (ii) i.e. inverse rotation to bring the rotation axis back to its original orientation.

v. Apply the inverse of step (i) i.e. inverse translation to bring the rotation axis back to its original position.

For step (ii) i.e. for coinciding the arbitrary axis with any co-ordinate axis, the rotations are needed about other two axes.

Direction cosines of the given line is: $[v] = [(x_1 - x_0)\ (y_1 - y_0)\ (z_1 - z_0)]$

$$[c_x\ c_y\ c_z] = \left[\frac{(x_1 - x_0)\ (y_1 - y_0)\ (z_1 - z_0)}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}}\right]$$

$$c_x = \frac{x_1 - x_0}{|v|} = \frac{10 - 1}{\sqrt{9^2 + 18^2 + 27^2}} = \frac{9}{33.67}$$

$$c_y = \frac{y_1 - y_0}{|v|} = \frac{18}{33.67}$$

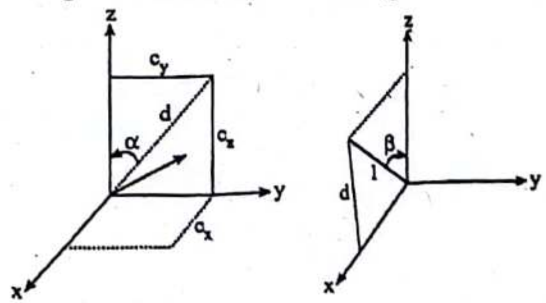
$$c_z = \frac{z_1 - z_0}{|v|} = \frac{27}{33.67}$$

To calculate the angles of rotation about x and y axes we use the direction cosines.

To put the line or rotation axis on the z axis we have to follow two steps.

a. First rotate about the x axis to transform vector u into the x z plane.

b. The swing u around to the z axis using 4 axis rotation.



$d = \sqrt{c_y^2 + c_z^2}$

$\cos\alpha = \frac{c_z}{d}$

$\sin\alpha = \frac{c_y}{d}$

$\cos\beta = d$

$\sin\beta = -c_x$

The complete the sequence of operations can be summarized as

$T = [T_r]^{-1}[R_x(\alpha)]^{-1}[R_y(\beta)]^{-1}[R_z(\theta)][R_y(\beta)][R_x(\alpha)][T_r]$

$$[T_r] = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_x(\alpha)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_y(\beta)] = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 1 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_z(\theta)] = \begin{bmatrix} \cos90^\circ & -\sin90^\circ & 0 & 0 \\ \sin90^\circ & \cos90^\circ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_y(\beta)] = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 1 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_x(\alpha)]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[T_r]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6. **Obtain perspective projection co-ordinate for the pyramid with vertices of base (15,15,10), (20,20,10), (25,15,10), (20,10,10) and apex (20,15,20) given that $z_{prp} = 20$ and $z_{vp} = 0$.**

Solution:

$z_{prp} = 20,\ z_{vp} = 0$

$x' = x - xu$

$y' = y - yu$

$z' = z - (z - z_{prp})u$

On the view plane, $z' = z_{vp}$. So,

$z_{vp} = z - (z - z_{prp})u$

$u = \frac{z_{vp} - z}{z_{prp} - z}$

For the vertex (15,15,10),

$$x_p = x - x\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$= x\left(\frac{z_{prp} - z - z_{vp} + z}{z_{prp} - z}\right)$$

$$= x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)$$

$$= 15\left(\frac{20 - 0}{20 - 10}\right)$$

$$= 30$$

$$y_p = y - y\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$= 10 - 10\left(\frac{0 - 10}{20 - 10}\right)$$

$$= 30$$

$z_p = z_{vp} = 0$

Projected points is $(x_1', y_1', z_1') = (30,30,0)$

Similarly for $P_2(X_2, Y_2, Z_2) = P_2(20,20,10)$

$$X_p = 20\left(\frac{20 - 0}{20 - 10}\right) = 40$$

$$Y_p = 20\left(\frac{20-0}{20-10}\right) = 40$$

$Z_p = 0$

Projected points is $(X'_2, Y_2', Z_2') = (40,40,0)$

Similarly, for vertex (25,15,10)

$X_p = 50$

$Y_p = 30$

$Z_p = 0$

Projected points is $P_3(X'_3, Y_3', Z_3') = (50,30,0)$

For vertex (20,10,10)

$$X_p = x - x\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$= 20 - 20\left(\frac{0-10}{20-10}\right)$$

$$Y_p = y - y\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$= 10 - 10\left(\frac{0-10}{20-10}\right)$$

$$= 20$$

$Z_p = 0$

For apex (20,15,20)

$$X_p = X\left(\frac{d_p}{z_{prp} - z}\right)$$

$$= 20\left(\frac{20}{20-20}\right)$$

$$= \infty$$

$$Y_p = Y\left(\frac{d_p}{z_{prp} - z}\right)$$

$$= 15\left(\frac{20}{20-20}\right)$$

$$= \infty$$

$Z_p = 0$

7. **A unit length cube with diagonal passing through (0,0,0) and (1,1,1) is shared with respect to yz-plane with share constants = 2 in both directions. Obtain the final coordinates of the cube after shearing.** **[2078 Bhadra]**

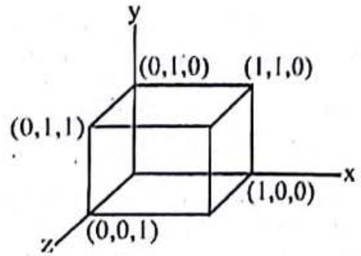
Solution:

Shearing with respect to zx-plane

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ sh_y & 1 & 0 & 0 \\ sh_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$sh_x = sh_y = 2$

A = (1,0,0)	B = (1,1,0)	C = (0,1,0)
D = (0,1,1)	E = (0,0,1)	F = (0,1,1)
G = (1,1,1)	H = (0,0,0)	



$$A' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

A' = (1,2,2)

Similarly, we can calculate B', C', D', E', F', G', H'

B' = (1,3,2)	C' = (0,1,0)	D' = (0,1,0)
E' = (0,0,0)	F' = (0,1,0)	G' = (1,3,2)
H' = (0,0,0)		

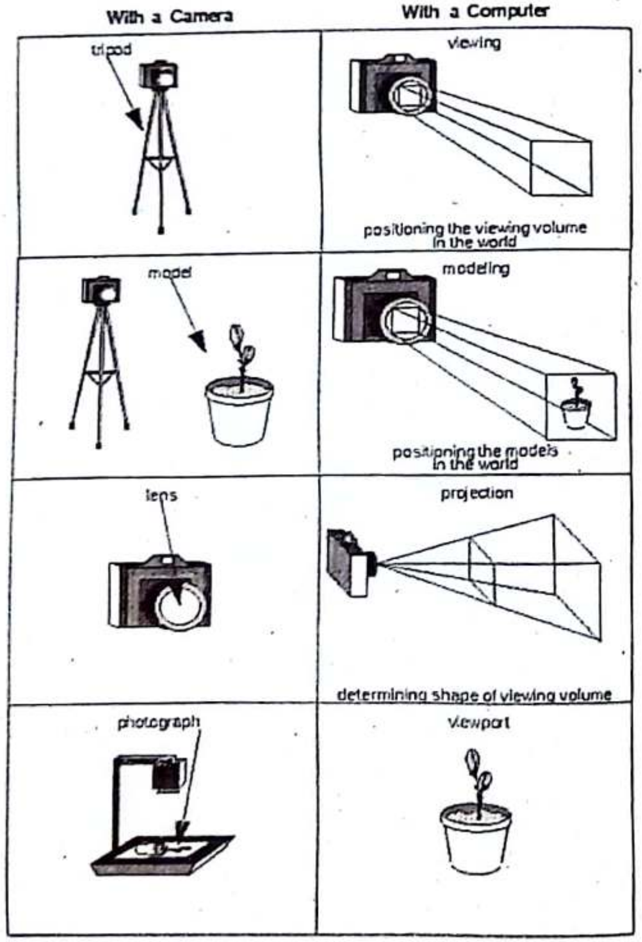
The final co-ordinates of the cube after shearing are A'(1,2,2), B'(1,3,2), C'(0,1,0), D'(0,1,0), D'(0,1,0), E'(0,0,0), F'(0,1,0), G'(1,3,2), and H'(0,0,0).

Fractal geometry method:

- Start with a basic shape, such as a line segment or triangle.
- Apply iterative transformations to the shape, using mathematical operations like scaling, rotation, translation, or other geometric transformations.
- Repeat the transformations multiple times, generating a sequence of modified shapes.

- Explore self-similarity, where smaller portions of the fractal resemble the overall shape, revealing intricate patterns at different scales.
- Adjust parameters within the transformation process to modify the appearance of the fractal, such as scaling factors, rotation angles, or probabilities assigned to different transformations.
- Iterate and refine the fractal shape by repeating the transformation process, allowing for the emergence of more complex and detailed patterns.
- Visualize and render the fractal using appropriate techniques, such as plotting points, rendering lines or curves, applying colors or textures, or utilizing advanced rendering algorithms to showcase the complexity and visual appeal of the fractal.

3D viewing



3D viewing refers to the process of perceiving or displaying images or objects in three dimensions, giving them the illusion of depth. It is commonly used in various fields such as entertainment, gaming, virtual reality, medical imaging, architecture, and more. We know that the objects available in nature are described in world coordinate system. But all graphics devices are 2D, such as screen monitor, printer etc. When we want to draw a 3D Object on a monitor, we have to convert the world co-ordinates into Screen co-ordinates. For this we have to project a 3D object on a 2D plane. So, projection is a process of representing a 3D object or scene in to 2D medium.

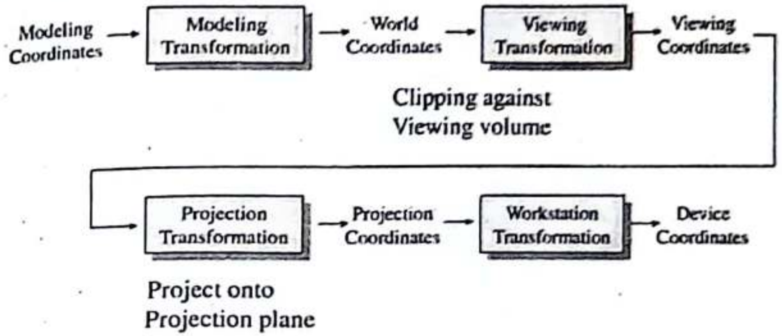


Fig.: 3D viewing pipeline

The 3D viewing pipeline refers to the sequence of steps involved in transforming and rendering a three-dimensional scene onto a two-dimensional display. It encompasses several stages, each contributing to the final visualization. Here is an overview of the typical 3D viewing pipeline:

- **Modeling:** The first step is to create a 3D model of the objects or scene that will be rendered. This involves defining the geometry, topology, and surface properties of the objects. Modeling can be done using specialized software or by scanning real-world objects using 3D scanners.
- **Scene Setup:** Once the 3D models are created, they need to be positioned, scaled, and oriented within a virtual 3D scene. The scene setup includes defining the position and properties of cameras, lights, and other elements that influence the rendering.
- **Viewing Transformation:** The viewing transformation involves transforming the 3D scene from the world coordinate system into a camera or eye coordinate system. This transformation determines the perspective and the viewing frustum, which represents the volume of space that will be rendered.

- **Projection:** In this stage, the 3D scene is projected onto a 2D plane known as the projection plane. The projection can be either perspective or orthographic. Perspective projection mimics the way human eyes perceive objects, creating a sense of depth, while orthographic projection maintains the same size of objects regardless of their distance from the camera.
- **Clipping and Culling:** Clipping is the process of removing the portions of objects that lie outside the viewing frustum. This improves performance and eliminates unnecessary calculations. Culling involves determining which objects or parts of objects are not visible to the camera and can be discarded from the rendering process.
- **Rasterization:** Rasterization converts the geometric primitives (such as points, lines, and polygons) into a pixel-based representation. It involves determining which pixels are covered by the primitives and assigning appropriate colors or textures to those pixels.
- **Shading:** Shading determines the color and appearance of each pixel based on lighting models, material properties, and surface characteristics. This includes calculations for ambient lighting, diffuse reflection, specular highlights, shadows, and other lighting effects.
- **Texturing:** Texturing is the process of mapping 2D images, called textures, onto the surfaces of 3D objects. It adds detail, patterns, and realism to the rendered scene. Texture coordinates are used to determine how textures are applied to the geometry.
- **Rendering:** The final step is the rendering of the 2D image or animation based on the transformed and shaded 3D scene. This involves converting the rasterize and shaded pixels into a format suitable for display or further processing. Various techniques, such as scanline rendering, ray tracing, or modern GPU-based rendering pipelines, can be used for this stage.

It's important to note that the specific implementation of the 3D viewing pipeline can vary depending on the rendering technology or framework being used. Advanced techniques, such as global illumination, ambient occlusion, or post-processing effects, can be integrated into the pipeline to enhance the visual quality and realism of the final output.

Projections

Projection in the context of 3D graphics refers to the process of transforming a 3D scene onto a 2D plane or display surface. This transformation is necessary because computer screens and most rendering outputs are inherently two-dimensional.

Important Terms Related to Projection

- **Center of Projection:** The point from where projection is taken. It can either be light source or eye position.
- **Projection Plane:** The plane on which projection of the object is formed.
- **Projectors:** Lines emerging from center of projection and hitting the projection plane after passing through a point in the object to be projected.

Thus, the plane geometric projection or simply projection of the objects are formed by the intersection of lines called projectors, on a plane called the projection plane. Projectors are

Types

a. **Parallel projection:** In a parallel projection, coordinate positions are transformed to the view plane along parallel lines. These are linear transforms that are useful in blueprints to produce scale drawings of three-dimensional objects.

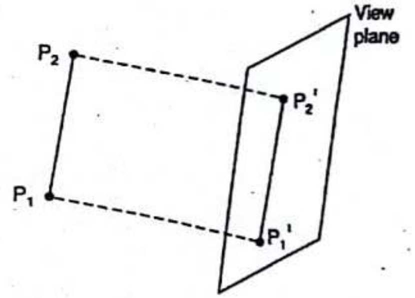


Fig.: Parallel projection of an object to the view plane

b. **Perspective projection:** For a perspective projection, object positions are transformed to the view plane along lines that converge to a point called to center of projection. The projected view of an object is determined by calculating the intersection of the projection lines with the view plane.

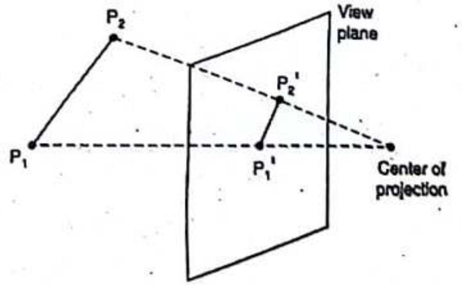


Fig.: Perspective projection of an object to the view plane

Categories of Projection

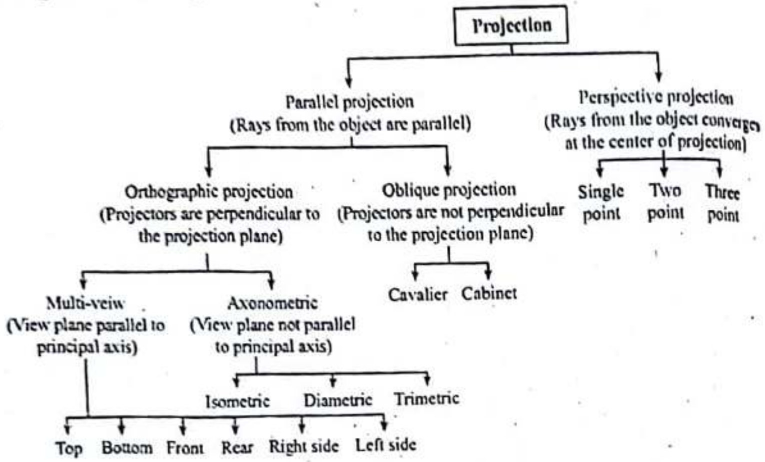
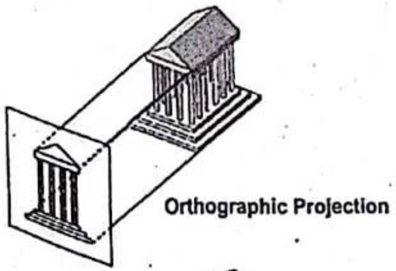
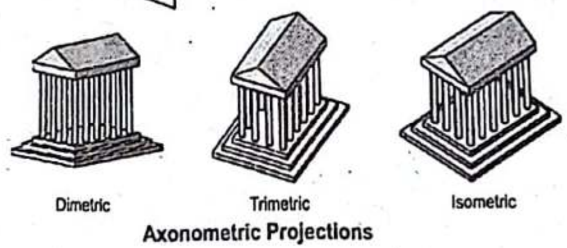
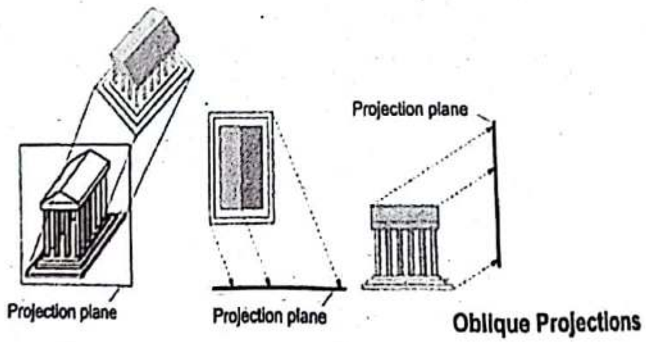


Fig.: Categories of Projection







1. Parallel Projection

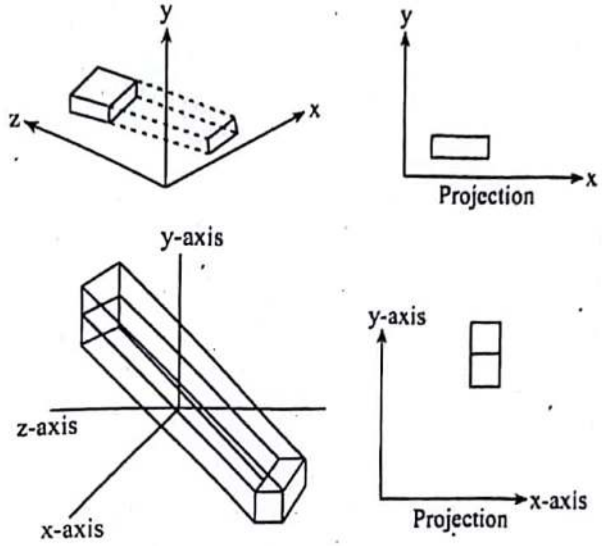


Fig.: Parallel projection

Parallel projection, also known as orthographic projection, is a type of projection used in 3D graphics where objects maintain their size and shape regardless of their distance from the viewer. In parallel projection, the projection rays from the 3D scene are parallel to each other and do not converge at a single point, as is the case with perspective projection.

In parallel projection, the 3D scene is projected onto a 2D plane without considering the depth or distance of objects. This type of projection is often used in technical and architectural drawings, engineering designs, and certain 2D games where maintaining accurate measurements and shapes is important.

Types

a. Orthographic

- Orthographic parallel projection is a technique used in computer graphics and technical drawing.
- It represents a three-dimensional object in a two-dimensional space without any perspective distortion.
- In orthographic projection, all projection lines are parallel to each other and perpendicular to the projection plane.
- The object is projected onto a plane by extending lines from each point on the object to intersect with the projection plane.

- The projection lines are perpendicular to the projection plane and do not converge at a single point like in perspective projection.
- Orthographic projection preserves the relative sizes and shapes of the object's features but eliminates depth information.
- It is commonly used in technical drawings, architectural plans, and engineering diagrams.
- Orthographic projection is useful for applications where accurate representation of object proportions is important.
- It can be performed from different viewpoints, such as front, top, side, or any desired orientation.
- In computer graphics, orthographic projection is often used in CAD software, architectural visualization, and game development.
- It is combined with other techniques to create realistic or stylized representations of three-dimensional scenes.

$$P_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } P_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

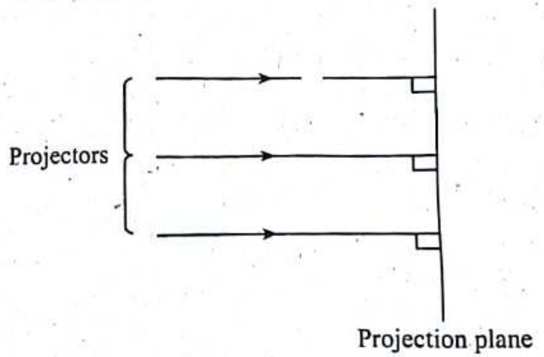


Fig.: In orthographic projection, projectors are perpendicular to projection plane

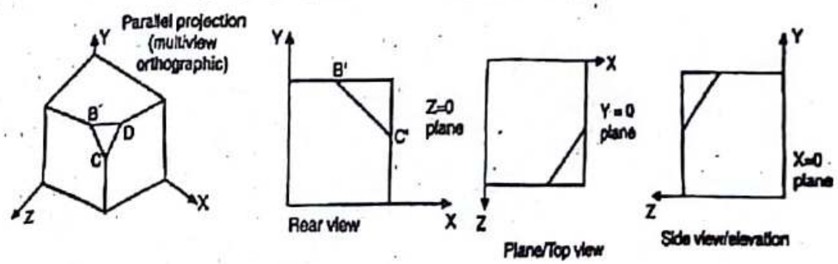
It is commonly used for engineering drawing. It is the projection on one of the coordinate planes i.e. $x = 0$, $y = 0$ or $z = 0$.

The matrix for projection onto the $x = 0$ plane is

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

In this matrix, we see that the first column, i.e., the x-column is all zero. The effect of the transformation is to set x-coordinate of the position vector to zero. Similarly, the matrices for projection onto the $y = 0$ and $z = 0$ plane are

$$P_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } P_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



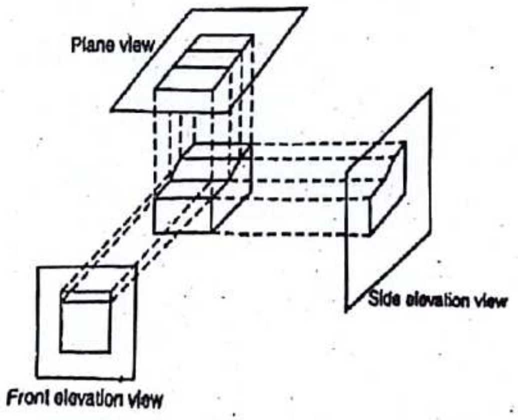


Fig.: Orthographic projections of an object

Axonometric projections

Axonometric projections are orthographic projections in which the direction of projection is not parallel to any of the three principal axis. The construction of an axonometric projection is done by using rotation and translation to manipulate the object such that atleast three adjacent faces are shown. The result is then projected at infinity onto one of the co-ordinate planes (usually $z = 0$ plane) from the centre of projection. An axonometric projection shows its true shape only when a face is parallel to the plane of projection. Here the parallel lines are equally fore shortened. The foreshortening factor is the ratio of the projected length of a line to its true length.

The sub-categories of axonometric projections are:

a. **Isometric projection:** The direction of projection makes equal angles with all three principal axis.

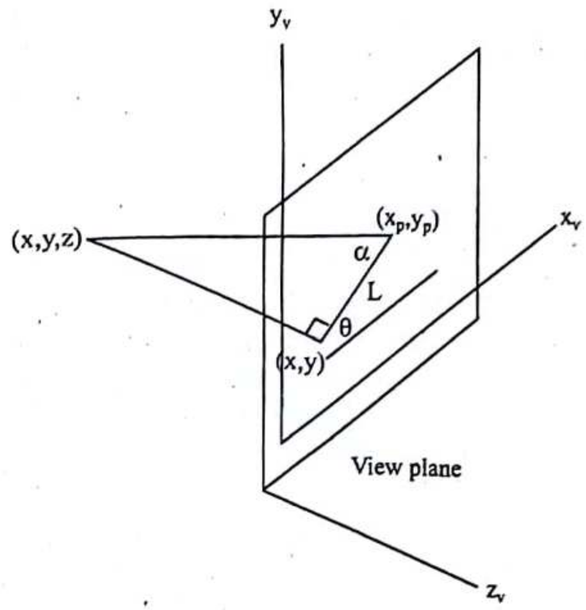
b. **Dimetric projection:** The direction of projection makes equal angles with exactly two of the principal axis.

c. **Termetric projection:** The direction of projection makes unequal angles with the three principal axis.

b. Oblique projection

- Oblique projection is a technique used in computer graphics and technical drawing to represent a three-dimensional object in a two-dimensional space.
- It is a type of projection that introduces a skew or slant to the object's appearance.
- In oblique projection, the object is projected onto the viewing plane by extending lines from each point on the object at a fixed angle.
- Unlike orthographic projection, the projection lines in oblique projection are not parallel to each other.
- The most common form of oblique projection is cavalier projection, where the object is projected with full scale along one axis and reduced scale along the other axes.
- Another form of oblique projection is called cabinet projection, where the object is projected with half scale along all axes.
- Oblique projection maintains the relative sizes and shapes of the object's features but does not accurately represent the way objects appear in the real world.
- It is often used for quick and simple representations, especially when a sense of depth is desired but not at the level of accuracy provided by perspective projection.
- Oblique projection is commonly used in technical drawings, architectural sketches, and visualizations where a simplified depiction of objects is acceptable.
- In computer graphics, oblique projection can be implemented through transformations or rendering techniques to achieve the desired skew or slant effect.

b. **Oblique parallel projection**



Obtained by projecting points along parallel lines that are not perpendicular to projection plane.

Often specified with two angles θ and α

Point (x,y,z) is projected to position (x_p,y_p) on the view plane orthographic projection coordinated on the plane are (x,y). Oblique projection line from (x,y,z) to (x_p,y_p) makes an angle with then line on the projection plane that joins (x_p,y_p) and (x,y).

This line of length L is at an angle θ with the horizontal direction in the projection plane. Expressing projection coordinates in terms of x, y, L and θ as

$x_p = x + L\cos\theta$

$y_p = y + L\sin\theta$

L depends on the angle α and z coordinate of point to be projected

$\tan\alpha = z/L$

thus, $L = z/\tan\alpha = zL_1$

L_1 is the inverse of $\tan\alpha$

so the oblique projection equations are

$x_p = x + z(L_1\cos\theta)$

$y_p = y + z(L_1\sin\theta)$

The transformation matrix for producing any parallel projection onto the $x_v y_v$ plane can be written as

$$M_{parallel} = \begin{pmatrix} 1 & 0 & L_1\cos\theta & 0 \\ 0 & 1 & L_1\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Orthographic projection is obtained when $L_1 = 0$ (occurs at projection angle α of 90*) oblique projection is obtained with none zero values for L_1.

Some common subcategories of oblique projections are:

a. Cavalier projection

b. Cabinet projection

The cavalier projection is obtained when the angle between the oblique projectors and the plane of projection is 45° and the foreshortening factors for all three principal directions are equal. In cavalier projection the resulting figure is thicker.

Cavalier projection is a form of oblique projection in which the projection lines are presumed to make a 45° vertical and a 45° horizontal angle with the plane of projection.

A Cabinet projection is used to correct the distortion that is produced by cavalier projection. An oblique projection for which the foreshortening factor for edge perpendicular to the plane of projection is one-half, is called a cabinet projection. For a cabinet projection, the angle between the projectors and the plane of projection is $\cot^{-1}(1/2) = 63.43°$.

2. **Perspective Projection**

- Perspective projection is a technique used in computer graphics and visual arts to represent a three-dimensional object in a two-dimensional space with realistic depth and foreshortening.
- It simulates the way objects appear in the real world by taking into account the principles of linear perspective.
- In perspective projection, the object is projected onto a flat plane, known as the image plane or projection plane.
- The projection lines converge at a single point, called the vanishing point, creating the illusion of depth and distance.
- Perspective projection creates a sense of depth by applying a diminishing scale factor to objects as they recede into the distance.

- Objects that are closer to the viewer appear larger, while objects that are farther away appear smaller.
- Perspective projection accurately represents the way objects appear in the real world, capturing the effects of foreshortening and spatial relationships.
- It is commonly used in realistic rendering, computer graphics, architectural visualization, and other applications where a lifelike representation of three-dimensional scenes is desired.
- Perspective projection requires defining the viewpoint or camera position, the viewing direction, and the field of view to control the extent of the scene captured.
- In computer graphics, perspective projection is achieved through mathematical transformations, such as the perspective projection matrix, applied to the object's vertices before rendering.

It transforms points along projects line that meet at projection reference point. Let the projection reference point be at z_{prp} along z_v axis and view plane be at z_{vp} the equations describing perspective coordinate points along perspective projection in parametric form is

$x' = x - x.u.$, $y' = y - y.u.$ and $z' = z - (z - z_{prp}).u.$

parameter 'u' takes values from 0 to 1 and coordinates (x',y',z') represents any point along projection line. When $u = 0$ we are at position $p(x,y,z)$ and at the other end of the line $u = 1$. On the view plane $z' = z_{vp}$.

Solving the equation for parameter u at this position $u = \frac{(z_{vp}-1)}{(z_{prp}-z)}$

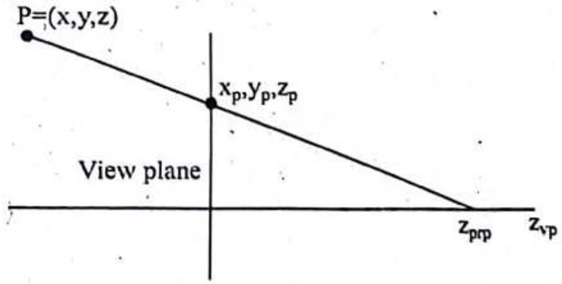
Substituting value of u in equation for x' and y'

The perspective transformation equations are

$$x_p = x.\frac{(z_{prp} - z_{vp})}{(z_{prp} - z)} = x.\left(\frac{d_p}{(z_{prp} - z)}\right)$$

$$y_p = y.\frac{(z_{prp} - z_{vp})}{(z_{prp} - z)} = y.\left(\frac{d_p}{(z_{prp} - z)}\right)$$

where $d_p = z_{prp} - z_{vp}$ the distance of the view plane form projection reference point



Using 3D homogenous coordinate representation the perspective transformation in matrix form is

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{vp}/d_p & z_{vp}(z_{prp}/d_p) \\ 0 & 0 & -1/d_p & z_{prp}/d_p \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where the homogenous factor is $h = \frac{(z_{prp} - z)}{d_p}$ and projection coordinates on the view plane are $x_p = \frac{x_h}{h}$ and $y_p = \frac{y_h}{h}$

Special cases

i. When $z_{vp} = 0$, view plane passes through origin

$$x_p = x \times \frac{(d_p)}{(z_{prp} - z)} = x \times \frac{1}{\left(1 - \frac{z}{z_{prp}}\right)}, \quad y_p = y \times \frac{(d_p)}{(z_{prp} - z)} = y \times \frac{1}{\left(1 - \frac{z}{z_{prp}}\right)}$$

ii. When $z_{prp} = 0$, reference point at origin

$$x_p = x \times \frac{z_{vp}}{z} = x \times \frac{1}{\left(\frac{z}{z_{vp}}\right)}, \quad y_p = y \times \frac{z_{vp}}{z} = y \times \frac{1}{\left(\frac{z}{z_{vp}}\right)}$$

Principal vanishing point

- Principal vanishing points are formed by the apparent intersection of lines parallel to one of the three principal x, y, z axes.
- The vanishing point for any set of lines that are parallel to one of the principal axes of an object is referred to as a principal vanishing point.
- The number of principal vanishing point is determined by number of principal axes intersected by the view plane.

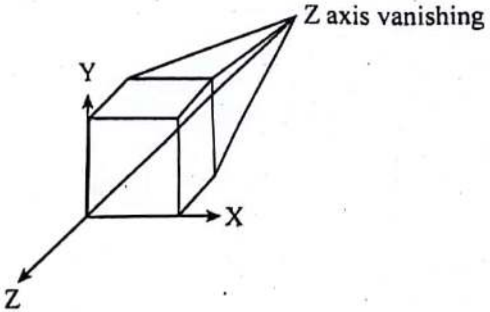
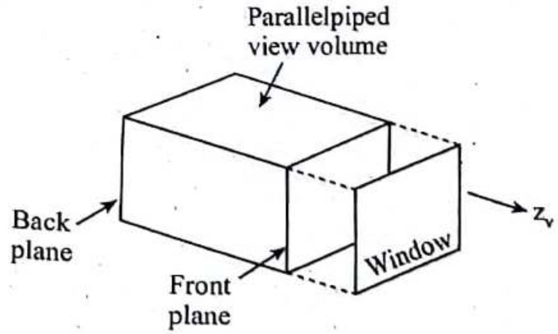
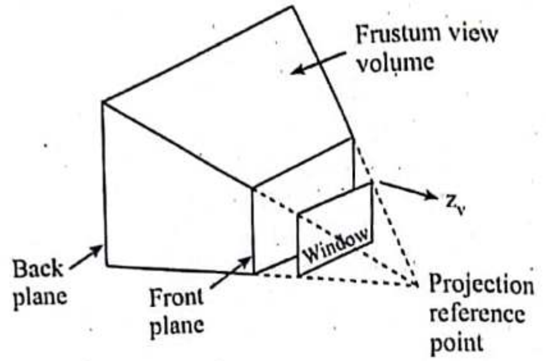


Fig.: Perspective views and z axis vanishing point

	Parallel projection	Perspective projection
Basic principle	All projection lines are parallel, and perpendicular to the plane.	Projection lines converge to a single vanishing point, creating depth.
Depth representation	Does not accurately represent depth or foreshortening.	Accurately represents depth, objects appear smaller as they recede.
Viewpoint effects	No effect on the perceived size or shape of objects.	Objects closer to the viewer appear larger, creating a sense of distance.
Application	Technical drawings, architectural plans, engineering diagrams.	Realistic rendering, computer graphics, architectural visualization, etc.
Realism	Provides a simplified, distortion free representation.	Provides a realistic representation with accurate spatial relationships.
Projection lines	Parallel and do not converge.	Converge to a vanishing point.
Field of view	Field of view does not affect the projection.	Field of view affects the perceived depth and extent of the projection.
Implementation	Relatively simpler implementation.	Requires more complex mathematical transformations and calculations.



Parallel projection



Perspective projection

3D Object Representation

Graphics scenes can contain trees, flowers, clouds rocks water, rubber, paper, bricks etc.

Polygon and quadratic surfaces provide precise descriptions for simple Euclidean objects such as polyhedrons ellipsoid.

Boundary Representation (B–rep):

Boundary representation, often referred to as B–rep, represents objects by explicitly defining their boundary surfaces. It describes the geometric properties of an object using vertices, edges, and faces. B–rep is based on the decomposition of an object's boundary into its constituent elements.

Key features of B–rep:

- **Vertices:** Points that define the corners or intersections of edges or faces.
- **Edges:** Line segments connecting pairs of vertices.
- **Faces:** Planar surfaces bounded by a closed loop of edges.

B–rep provides a detailed and explicit representation of the object's geometry, allowing for accurate visualization and manipulation. It enables operations like Boolean operations (union, intersection, subtraction) and geometric transformations (translation, rotation, scaling).

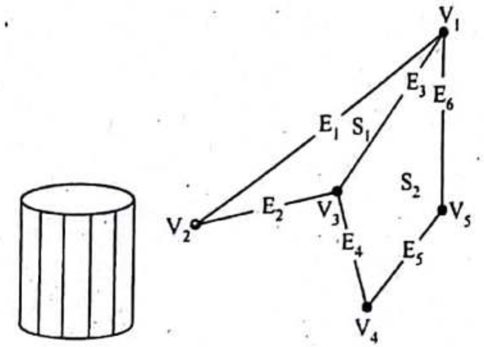
Space Partitioning Representation:

Space partitioning representation focuses on dividing 3D space into smaller regions or volumes to efficiently organize and process objects within that space. It is commonly used for spatial indexing and collision detection in computer graphics and simulations.

Polygon surfaces

- 3-D graphics object is a set of surface polygons that enclose the object interior

- A polygon mesh is a set of connected polygonally bounded planar surfaces
- Equation of plane is $A_x + B_y + C_z + D = 0$
- A polygon mesh is a collection of edges, vertices and polygons connected such that each edge is shared by at most two polygons



Vertex table	Edge table	Polygon-surface table
V_1: x_1, y_1, z_1	E_1: V_1, V_2	S_1: E_1, E_2, E_3
V_2: x_2, y_2, z_2	E_2: V_2, V_3	S_2: E_3, E_4, E_5, E_6
V_3: x_3, y_3, z_3	E_3: V_3, V_1	
V_4: x_4, y_4, z_4	E_4: V_3, V_4	
V_5: x_5, y_5, z_5	E_5: V_4, V_5	
	E_6: V_5, V_1	

Polygon tables

- **Polygon data tables can be organized into two groups:** Geometrical and attribute tables.
- Geometric data tables contain vertex coordinates and parameters to identify the spatial orientation of polygon surfaces.
- Attribute information for an object includes parameters specifying the degree of transparency of object and its surface reflectivity and texture characteristics,
- **A convenient organization for storing geometric data is to create three lists:**
  i. a vertex table
  ii. an edge table
  iii. a polygon table

I. **Vertex Table:**

- Stores information related to the vertices of a 3D object.

- Each entry represents a single vertex and includes attributes like coordinates, normals, texture coordinates, and colors.
- Provides centralized storage for vertex data, making it easy to access and manipulate vertices.

ii. Edge Table:

- Stores information about the edges of a 3D object.
- Each entry represents a single edge and includes attributes like indices of connected vertices.
- Enables efficient edge-based operations such as rendering, detection, or topological queries.

iii. Polygon Table:

- Stores information related to the polygons (faces) of a 3D object.
- Each entry represents a single polygon and includes attributes like vertex indices, normals, materials, and texture mapping information.
- Facilitates easy access to polygon data for rendering, culling, shading, and other operations.

Representing 3D objects using polygon tables:

- Define vertices with 3D coordinates.
- Create polygons by connecting the vertices.
- Store vertices in a vertex table.
- Store polygons in a polygon table with vertex indices.
- Calculate vertex normals for shading.
- Optionally, apply texture mapping to polygons.
- Render and display the 3D object.

Rules for generating error–free polygon tables:

- Use valid vertex indices.
- Avoid degenerate polygons.
- Ensure a closed object without open edges or gaps.
- Prevent intersecting edges.
- Maintain consistent polygon orientation.
- Ensure vertex normal consistency.
- Preserve texture mapping coherence.

Plane equations

The equation for a plane surface can be expressed as

$$A_x + B_y + C_z + D = 0 \text{ ....(i)}$$

where, (x,y,z) is any point on the plane-coefficients ABCD are constants describing the spatial properties of the plane

For solving ABCD consider three successive polygon vertices (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)

So equation (i) is modified to,

$$\frac{A}{D}x_k + \frac{B}{D}y_k + \frac{C}{D}z_k = -1 \text{ ....(ii) } k = 1,2,3$$

Using crammers rule,

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix}, B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}, C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix},$$

$$D = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

Expanding determinants,

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$

$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$

$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$

$$D = -x_1(y_2z_3 - y_3z_2) - x_2(y_3z_1 - y_1z_3) - x_3(y_1z_2 - y_2z_1)$$

Plane equations are used to identify the position of spatial points relative to the plane surfaces of an object. For any point (x,y,z) not on plane with parameters ABCD we have

$$A_x + B_y + C_z + D = 0$$

We can identify the point as either inside or outside the plane surface according to the sign (+ or –) of

$$A_x + B_y + C_z + D$$

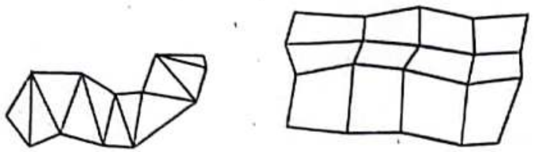
If $A_x + B_y + C_z + D < 0$ then the point (x,y,z) is inside the surface

If $A_x + B_y + C_z + D > 0$ then the point (x,y,z) is outside the surface

Polygon Meshes

Some graphics packages (PHIGS programmer's Hierarchical Interactive Graphics Standard) provide several polygon functions for modeling objects.

It produces n – 2 connected triangles, given the coordinates for n vertices.



Another similar functions the quadrilateral mesh that generates a mesh of (n−1)(m−1) quadrilaterals, given the coordinates for an n by m array of vertices.

When polygons are specified with more than 3 vertices, it is possible that the vertices may not all lie in one plane.

This can be due to numerical error or errors in selecting coordinate positions for the vertices.

Remedies:

i. Simply divide the polygons into triangles.

ii. Approximate the plane parameters ABC and project in same plane (i.e. approximate A to yz plane, B to xz plane, C to xy plane etc).

High quality graphics systems typically model objects with polygon meshes and set up a database of geometric and attribute information to facilitate processing of the polygon facets.

Curved Surfaces and Lines

Display of 3D curved lines and surfaces can be generated from an input set of mathematical functions (spheres, ellipsoid etc) defining the objects or from a set of user specified data points.

Spline representations are examples of generating curves and surfaces. These are commonly used to design new object shapes, to digitize drawings and to describe animation paths.

Curve fitting methods are also used to display graphs of data values by fitting specified curve functions to the discrete data set, using regression techniques such as the least square methods.

Curve continuity or smoothness of curve

There are two approaches which determine the smoothness of curve or curve continuity. They are as follows:

i. Parametric continuity conditions (C)

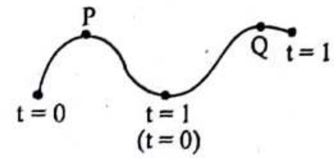
ii. Geometric continuity conditions (G)

Parametric Continuity Conditions

Parametric continuity deals in parametric equations associated to piecewise parametric polynomial curve not the shape or appearance of the curve. We set parametric continuity by matching the parametric derivatives of adjoining curve sections at their common boundary.

I. **Zero order parametric continuity (C^0):**

C^0 continuity means that two piece of curves are joined or meet at same point. The two pieces of curve P and Q are in zero order parametric continuity if $P(t = 1) = Q(t = 0)$.



II. **First order parametric continuity (C^1):**

Two successive curve sections are in first order parametric continuity if first parametric derivative of the coordinate function are equal at the joining point.

$P'(t = 1) = Q'(t = 0)$

Where P' and Q' are first order derivative.

III. **Second order parametric continuity (C^2):**

Two curves are in second order parametric continuity if both first and second derivatives of the two curve sections are same at the intersection point.

$P''(t = 1) = Q''(t = 0)$

Fig.: Piecewise construction of a curve by joining two curve segments using different order of continuity (a) zero-order continuity only (b) first-order continuity (c) second-order continuity.

In summary we can conclude that,

- **C^0:** curves are joined
- **C^1:** first derivatives equal
- **C^2:** first and second derivatives are equal and if t is taken to be time, this implies that the acceleration is continuous.
- **C^n :** n^{th} derivatives are equal

Geometric Continuity Conditions

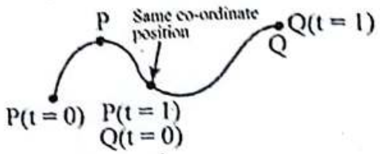
Another method for joining two successive curve sections is to specify conditions for geometric continuity. Geometric continuity refers to the way that a curve or surface looks. In this case, we only require parametric derivatives of the two sections to be proportional to each other at their common boundary instead of equal to each other.

i. Zero order geometric continuity (G^0):

Zero-order geometric continuity (G^0 continuity) is the same as zero-order parametric continuity. That is, the two curves sections must have the same coordinate position at the boundary point.

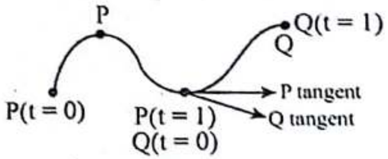
$P(t = 1) = Q(t = 0)$

P and Q are two segments of curves.



ii. First order geometric continuity (G^1):

First-order geometric continuity (G^1 continuity) means that the parametric first derivatives are proportional at the intersection of two successive sections. If P and Q are Q are two piece of curves, then P'(t) and Q'(t) must have same direction of tangent vector but not necessary to be the same magnitude.

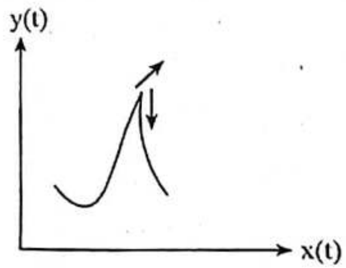


iii. Second order geometric continuity (G^2):

Second-order geometric continuity (G^2 continuity) means that both the first and second parametric derivatives of the two curve sections are proportional at their boundary. Under G^2 continuity, curvatures of two curve sections will match at the joining position.

In general C^1 continuity implies G^1 continuity but G^1 continuity doesn't imply C^1 continuity.

C^1 continuity doesn't imply G^1 continuity when segments tangent vector are [0 0 0] at join point. In this case, the tangent vectors are equal but there directions are different.



In summary we can conclude that,

- **G^0: curves are joined**
- **G^1: first derivatives are proportional at the join point and the curve tangents thus have the same direction, but not necessarily the same magnitude. i.e., $C'(1)$ = a,b,c) and $C^{2\prime}(0)$ = k×a, k×b, k×c).**
- **G^2: first and second derivatives are proportional at join point.**

As their names imply, geometric continuity requires the geometry to be continuous, while parametric continuity requires that the underlying parameterization be continuous as well.

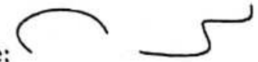
Splines

- **Splines are cubic curves which maintain C^2 continuity.**
- **Natural spline**
  - interplate all of its control points.
  - equivalent to a thin strip of metal forced to pass through control points
  - no local control

  B-spline
  - local control
  - does not interpolate control points

Three types of curve

There are three types of curve. They are:

i. Open curve: 

ii. Closed curve: 

iii. Crossing curve: 

Representation of curve

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

i. Explicit function
ii. Implicit function
iii. Parametric function

i. **Explicit representation of curve:**

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

$y = f(x)$

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

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

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

ii. **Implicit representation of curve:**

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

$F(x,y) = 0$

e.g., $x^2 + y^2 - 1 = 0$

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

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

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

iii. **Parametric representation of curve:**

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

Curves having parametric form are called parametric curves.

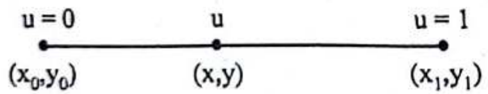
$x = f_x(u)$

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

Similarly, parametric equation of line is;

$x = (1 - u)x_0 + ux_1$

$y = (1 - u)y_0 + uy_1$



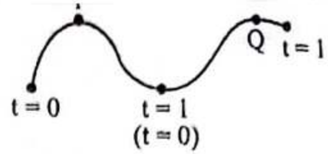
Parametric curve

The parametric representation for curve is as follows:

$x = x(t)$

$y = y(t)$

$z = z(t)$



A curve is approximated by a piecewise polynomial curve instead of piecewise linear curve. Piecewise linear curve is represented by linear equation and polyline. Piecewise polynomial curve is represented by polynomial equation.

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

The cubic polynomial that define a curve can be represented as

Bezier surfaces

To create a Bezier surface, we blend a mesh of Bezier curves using the blending function.

$$P(u,v) = \sum_{j=0}^{m} \sum_{k=0}^{n} P_{j,k} BEZ_{j,m}(v) BEZ_{k,n}(u)$$

where j and k of the knots in real space. The Bezier functions specify the weighting of a particular knot. They are the Bernstein coefficients. The definition of the Bezier functions is P_{xy}

where C(n,k) represents the binary coefficients. When u = 0, the function is one for k = 0 and zero ....

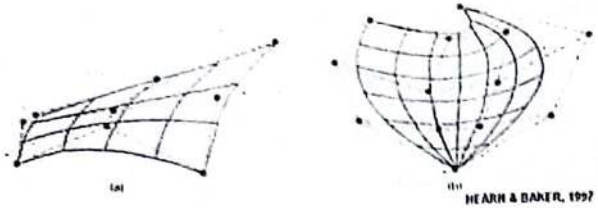
$$BEZ_{k,n}(u) = C(n,k)u^k(1 - u)^{u-k}$$

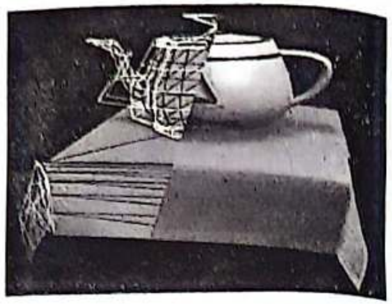
when we combine two orthogonal parameters, we find a Bezier curve along each edge of the surface, as defined by the points along that edge.

Bezier surfaces are useful for interactive design and were first applied to car body design.

The properties of Bezier surfaces are controlled by the blending functions

- The surface takes the general shape of the control points.
- The surface is contained within the convex hull of the control points.
- The corners of the surface and the corner control vertices are coincident.



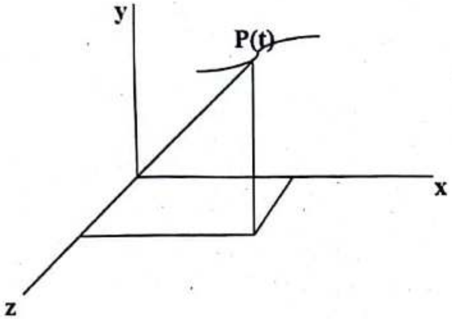


Parametric cubic curve

A parametric cubic curve is defined as $P(t) = \sum_{i=0}^{3} a_i t^i$

$0 <= t <= 1$ ....(i)

where, P(t) is a point on the curve



Expanding equation (i) yields

$P(t) = a_3t^3 + a_2t^2 + a_1t + a_0$ ....(ii)

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

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

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

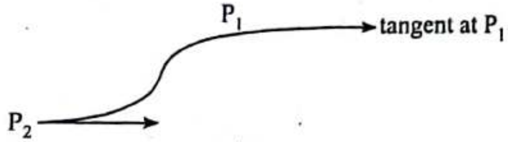
$z(t) = a_{3z}t^3 + a_{2z}t^2 + a_{1x}t + a_{0z}$ ....(iii)

To be able to solve (iii) the twelve unknown coefficients a_{ij} (algebraic coefficients) must be specified.

From the known end point coordinates of each segment, six of the twelve needed equations are obtained.

The other six are found by using tangent vectors at the two ends of each segment.

The direction of the tangent vectors establishes the slopes (direction cosines) of the curve at the end points.



This procedure for defining a cubic curve using end points and tangent vector is one form of hermitte interpolation.

Each cubic curve segment is parameterized from 0 to 1 so that known end points correspond to the limit values of the parametric variable t, that is P(0) and P(1).

Substituting t = 0 and t = 1 the relationship between two end point vectors and the algebraic coefficients are found.

$P(0) = a_0$ $\qquad$ $P(1) = a_3 + a_2 + a_1 + a_0$

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

$P'(t) = 3a_3t^2 + 2a_2t + a_1$

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

$P'(0) = a_1$ $\qquad$ $P'(1) = 3a_3 + 2a_2 + a_1$

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

$a_0 = P(0)$ $\qquad$ $a_1 = P'(0)$

$a_2 = -3P(0) + 3P(1) - 2P'(0) - P'(1)$

$a_3 = 2P(0) - 2P(1) + P'(0) + P'(1)$

substituting these values of 'a_i' in equation (ii) and rearranging the terms yields

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

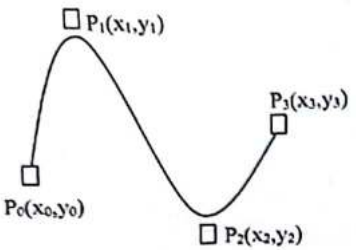
The values of P(0), P(1), P'(0), P'(1) are called geometric coefficients and represent the known vector quantities in the above equation

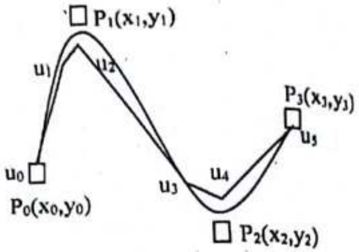
The polynomial coefficients of these vector quantities are commonly known as blending functions

By varying parameter t in these blending function from 0 to 1 several points on curve segments can be found

Spline: A spline is a flexible strip that passes thru a designated control points.

Bezier curve





The above figure shows a smooth curve comprising of a large number of very small line segments for understanding the concept to draw such a line we deal with a curve as show above which is an approximation of the curve with five line segments only.

The approach below is used to draw a curve for any number of control points.

Suppose P_0, P_1, P_2, P_3 are four control points

Number of segments in a line segment: nSeg

i = 0 to nSeg

u = i/nSeg [0,1] 0 <= u <= 1

$u_0, u_1, \ldots\ldots, u_3$

$$x(u) = \sum_{j=0}^{n} x_j\, BEZ_{j,n}(u) \qquad \text{n: number of control points}$$

$$x(u) = x_0 BEZ_{0,3}(u) + x_1 BEZ_{1,3}(u) + x_2 BEZ_{2,3}(u) + x_3 BEZ_{3,3}(u)$$

Similarly

$$y(u) = \sum_{j=0}^{n} y_j BEZ_{j,n}(u) \quad \text{n: number of control points}$$

$$y(u) = y_0 BEZ_{0,3}(u) + y_1 BEZ_{1,3}(u) + y_2 BEZ_{2,3}(u) + y_3 BEZ_{3,3}(u)$$

The Bezier blending function $BEZ_{j,n}(u)$ is defined as,

$$BEZ_{j,n}(u) = \frac{n!}{j!(n-j)!} u^j (1-u)^{n-j}$$

$$BEZ_{j,n}(u) = C_{(n-j)}\, u^j\, (1-u)^{n-j}$$

Where $C_{(n-j)}$ is the Binomial Coefficient

$$C_{(n,j)} = \frac{n!}{j!(n-j)!}$$

For each 'u' the coordinates x and y are computed and desired curve is produced when the adjacent coordinates (x,y) are connected with a straight line segment

Now,

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

Four blending functions must be found based on Bernstein Polynomials

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

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

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

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

Normalizing properties apply to blending functions that means thy all add up to one

Substituting these functions in above equation

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

When u = 0 then $Q(u) = P_0$ and when u = 1 then $Q(u) = P_3$

In Matrix Form

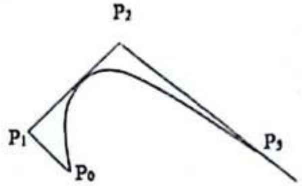
$$Q(u) = [(1-u)^3 \quad 3u(1-u)^2 \quad 3u^2(1-u) \quad u^3] \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

$$\text{or, } Q(u) = [(1-3u+3u^2-u^3) \quad (3u-6u^2+3u^3) \quad (3u^2-3u^3) \quad u^3] \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

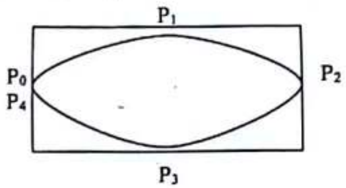
$$\text{or, } Q(u) = [u^3 \ \ u^2 \ \ u^1 \ \ 1]\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

Properties of a Bezier Curve

1. Bezier curve lies in the convex hull of the control points which ensure that the curve smoothly follows the control.



2. Four Bezier polynomials are used in the construction of curve to fit four control points.
3. It always passes thru the end points.
4. Closed curves can be generated by specifying the first and last control points at the same position



5. Specifying multiple control points at a single position gives more weight to that position.
6. Complicated curves are formed by piecing several sections of lower degrees together.
7. The tangent to the curve at an end point is along the line joining the end point to the adjacent control point.

Solved Numerical Examples

1. **Construct the Bezier curve of order 3 and with 4 polygon vertices A(1,1), B(2,3), C(4,3) and D(6,4).**

Solution:

The equation for the Bezier curve is given as

$P(u) = (1-u)^3P_1 + 3u(1-u)^2P_2 + 3u^2(1-u)P_3 + u^3P_4$

for $0 \le u \le 1$

where P(u) is the point on the curve P_1, P_2, P_3, P_4.

Let us take, $u = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$

$\therefore \ P(0) = P_1 = (1,1)$

$$\therefore \ P\left(\frac{1}{4}\right) = \left(1-\frac{1}{4}\right)^3 P_1 + 3\frac{1}{4}\left(1-\frac{1}{4}\right)^2 P_2 + 3\left(\frac{1}{4}\right)^2\left(1-\frac{1}{4}\right)P_3 + \left(\frac{1}{4}\right)^3 P_4$$

$$= \frac{27}{64}(1,1) + \frac{27}{64}(2,3) + \frac{9}{64}(4,3) + \frac{1}{64}(6,4)$$

$$= \left[\frac{27}{64}\times1 + \frac{27}{64}\times2 + \frac{9}{64}\times6, \frac{27}{64}\times1 + \frac{27}{64}\times3 + \frac{9}{64}\times3 + \frac{1}{64}\times4\right]$$

$$= \left[\frac{123}{64}, \frac{139}{64}\right] = (1.9218, 2.1718)$$

$$\therefore \ P\left(\frac{1}{2}\right) = \left(1-\frac{1}{2}\right)^3 P_1 + 3\frac{1}{2}\left(1-\frac{1}{2}\right)^2 P_2 + 3\left(\frac{1}{2}\right)^2\left(1-\frac{1}{2}\right)P_3 + \left(\frac{1}{2}\right)^3 P_4$$

$$= \frac{1}{8}(1,1) + \frac{3}{8}(2,3) + \frac{3}{8}(4,3) + \frac{1}{8}(6,4)$$

$$= \left[\frac{1}{8}\times1 + \frac{3}{8}\times2 + \frac{3}{8}\times4 + \frac{1}{8}\times6, \frac{1}{8}\times1 + \frac{3}{8}\times3 + \frac{3}{8}\times3 + \frac{1}{8}\times4\right]$$

$$= \left[\frac{25}{8}, \frac{23}{8}\right] = (3.125, 2.875)$$

$$\therefore \ P\left(\frac{3}{4}\right) = \left(1-\frac{3}{4}\right)^3 P_1 + 3\frac{3}{4}\left(1-\frac{3}{4}\right)^2 P_2 + 3\left(\frac{3}{4}\right)^2\left(1-\frac{3}{4}\right)P_3 + \left(\frac{3}{4}\right)^3 P_4$$

$$= \frac{1}{64}P_1 + \frac{9}{64}P_2 + \frac{27}{64}P_3 + \frac{27}{64}P_4$$

$$= \frac{1}{64}(1,1) + \frac{9}{64}(2,3) + \frac{27}{64}(4,3) + \frac{27}{64}(6,4)$$

$$= \left[\frac{1}{64}\times1 + \frac{9}{64}\times2 + \frac{27}{64}\times4 + \frac{27}{64}\times6, \frac{1}{64}\times1 + \frac{9}{64}\times3 + \frac{27}{64}\times3 + \frac{27}{64}\times4\right]$$

$$= \left[\frac{289}{64}, \frac{217}{64}\right] = (4.5156, 3.375)$$

$P(1) = P_3 = (6,4)$

The figure, shows the calculated points of the Bezier curve and curve passing through it.

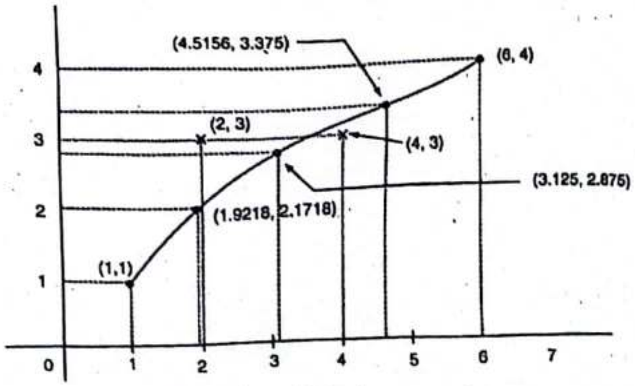


Fig.: Plotted Bezier curved

2. **Find equation of Bezier curve which passes through points (0,0) and (−2,1) and is controlled through points (7,5) and (2,0).**

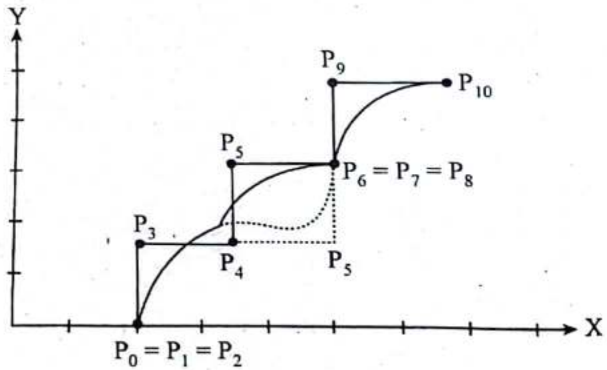
Solution:

Using four given points, as control points a cubic Bezier curve can be defined. But as the cubic Bezier passes through (0,0) and (−2,1) these two points should be considered as the end control points while the other two points (7,5) and (2,0) should be considered as control points, we obtain two different equations of cubic Bezier curve.

i. Let $P_0 = (0,0)$; $P_1 = (7,5)$; $P_2 = (2,0)$; $P_3 = (-2,1)$.

The corresponding cubic Bezier curve is given by

$$P(u) = [u^3 \;\; u^2 \;\; u \;\; 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 7 & 5 \\ 2 & 0 \\ -2 & 1 \end{bmatrix} \quad (0 \le u \le 1)$$



$$= [u^3 \;\; u^2 \;\; u \;\; 1]\begin{bmatrix} 13 & 16 \\ -36 & -30 \\ 21 & 15 \\ 0 & 0 \end{bmatrix}$$

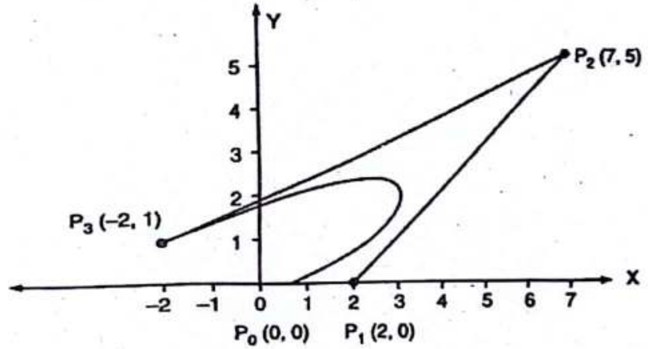
$$= (13u^3 - 36u^2 + 21u)(16u^3 - 30u^2 + 15u)$$

ii. If $P_0 = (0,0)$; $P_1 = (2,0)$; $P_2 = (7,5)$ and $P_3 = (-2,1)$ then the resulting cubic Bezier curve is given by

$$P(u) = [u^3 \;\; u^2 \;\; u \;\; 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 7 & 5 \\ -2 & 1 \end{bmatrix} \quad (0 \le u \le 1)$$

$$= [u^3 \;\; u^2 \;\; u \;\; 1]\begin{bmatrix} -17 & -14 \\ 9 & 15 \\ 6 & 0 \\ 0 & 0 \end{bmatrix}$$

$$= [-17u^3 + 9u^2 + 6u]\,[-14u^3 + 15u^2]$$



3D Clipping

In **2D Clipping** a **window is considered** as a clipping boundary **but** in **3D** a **view volume is considered**, which is a box between the two planes, the front and the back plane.

The part of the object which lies inside the view volume will be displayed and the part that lies outside will be clipped.

For a parallel projection a box or a region is a rectangular area and in case of perspective projection it is a truncated pyramidal volume called a frustum of vision.

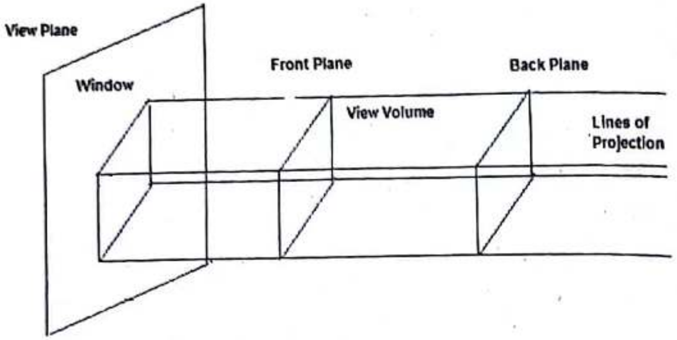
The view volume has **6 sides:** Left, Right , Bottom, Top, Near and Far.

Cohen Sutherland's region code approach **can be extended** for 3D clipping as well.

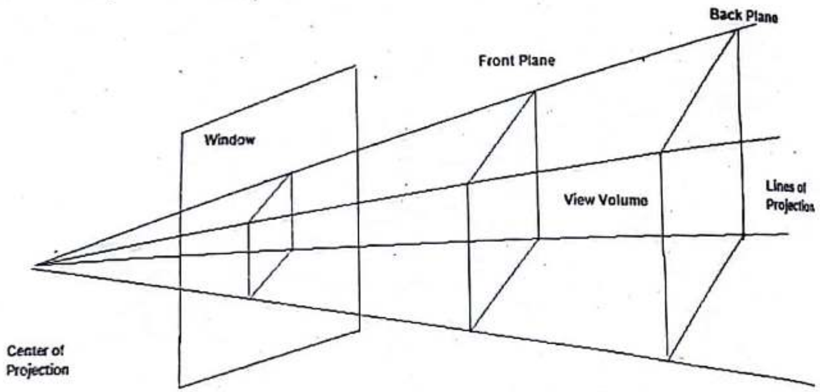
In 2D, a point is checked if it is inside the visible window/region or not but in 3D clipping a point is compared against a plane.

View Volume in case of

i. Parallel Projection



ii. Perspective Projection



The front and back planes are positioned relative to the view reference point in the direction of the view plane normal

Here six bits are used to denote the region code

The bits are set to 1 as per the following rule:

Bit 1 is set to 1 if $x < x_{vmin}$	Bit 2 is set to 1 if $x > x_{vmax}$
Bit 3 is set to 1 if $y < y_{vmin}$	Bit 4 is set to 1 if $y > y_{vmax}$
Bit 5 is set to 1 if $z < z_{vmin}$	Bit 6 is set to 1 if $z > z_{vmax}$

If both the end points have region codes 000000 then the line is **completely visible**.

If the logical AND of the two end points region codes are not 000000 i.e. the same bit position of both the end points have the value 1, then the line is **completely rejected or invisible** else it is the case of **partial visibility** so the intersections with the planes must be computed

For a line with end points $P_1 (x_1,y_1,z_1)$ and $P_2 (x_2,y_2,z_2)$, the parametric equation can be expressed as:

$x = x_1 + (x_2 - x_1)\,u \qquad y = y_1 + (y_2 - y_1)\,u \qquad z = z_1 + (z_2 - z_1)\,u$

If we are testing a line against the front plane of the viewport then $z = z_{vmin}$ and

$u = (z_{vmin} - z_1)/(z_2 - z_1)$

therefore $x_I = x_1 + (x_2 - x_1)\{(z_{vmin} - z_1)/(z_2 - z_1)\}$

$y_I = y_1 + (y_2 - y_1)\{(z_{vmin} - z_1)/(z_2 - z_1)\}$

where x_I and y_I are the intersection points with the plane

Solved Numerical Problems

1. **Find the perspective projection of a tetrahedron A(3,4,0) B(1,0,4), C(2,0,5), D(4,0,3)onto a projection plane situated at 0. The center of projection should be located at –5.**

Solution:

Center of projection on the z-axis.

$$[M_{per}] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/z_{vp} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\therefore\ P^* = [P]\,[M_{per}]$

$$\begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ = \begin{bmatrix} 3 & 1 & 2 & 4 \\ 4 & 0 & 0 & 0 \\ 0 & 4 & 5 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

2. **A triangle with vertices A(0,0,8), B(25,50,8), C(50,0,8) is required to be projected onto a plane located at (0,0,–2) using perspective projection with the vanishing point at (0,0,–10) where will the triangle be placed after projection?**

Solution:

1. For A(0,0,8) → A'(0,0,−2)

$$x' = x - x_4 = 0 - 0\left(\frac{z_{vp} - 2}{z_{prp} - 2}\right) = 0$$

$$y' = y - y_4 = 0 - 0\left(\frac{z_{vp} - 2}{z_{prp} - 2}\right) = 0$$

$z' = -2'$

2. For B(25,50,8) → B'(11.125,22.25,−2)

$$x' = 25 - 25\left(\frac{-20 - 8}{-10 - 8}\right) = 11.125$$

$$y' = 50 - 50\left(\frac{-10}{-18}\right).\ 22.25$$

$z' = (0,0,-2)$

3. C(50,0,8) → C'(22.25,0,−2)

$$x' = 50 - 50\left(\frac{-2 - 8}{-10 - 8}\right) = 22.25$$

$y' = 0 - 0\,(\,) = 0$

$z' = -2.$

3. Compute the necessary coordinates for forming a Bezier curve taking four control points $P_1(10,10)$, $P_2(40,40)$, $P_3(60,40)$, $P_4(80,10)$ and using five line segments.

Solution:

For four control points (O) = 3

Calculating the blending functions:

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

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

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

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

$BEZ_{3,3}(u) = u^3$

$$u_0 = \frac{0}{5} = 0,\ u_1 = \frac{1}{5} = 0.2,\ u_2 = \frac{2}{5} = 0.4,\ u_3 = \frac{3}{5} = 0.6,\ u_4 = \frac{4}{3} = 0.8,\ u_5 = \frac{5}{5} = 1$$

We know, $0 \le u \le 1$,

$x(u) = x_0BEZ_{0,3}(u) + x_1BEZ_{1,3}(u) + x_2BEZ_{2,3}(u) + x_3BEZ_{3,3}(u)$

$y(u) = y_0BEZ_{0,3}(u) + y_1BEZ_{1,3}(u) + y_2BEZ_{2,3}(u) + y_3BEZ_{3,3}(u)$

Here,

$x_0 = 10$ $\quad y_0 = 10$

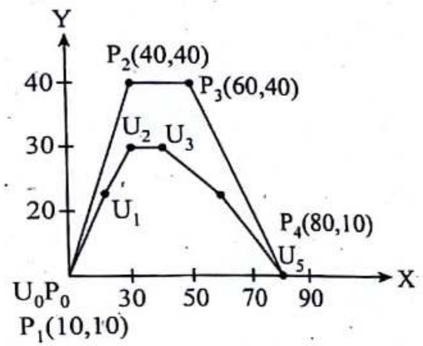
$x_1 = 40$ $\quad y_1 = 40$

$x_2 = 60$ $\quad y_2 = 40$

$x_3 = 80$ $\quad y_3 = 10$

Now,

u	x(u)	y(u)
0	10	10
0.2	26.88	24.2
0.4	41.84	31.6
0.6	55.36	31.6
0.8	67.92	24.4
1	80	10



4. A unit length cube with diagonal passing through (0,0,0) and (2,2,2) is shared with respect to zx-plane with share constants = 3 in both directions. Obtain the final coordinates of the cube after shearing.

Solution:

Shearing with respect to zx-plane

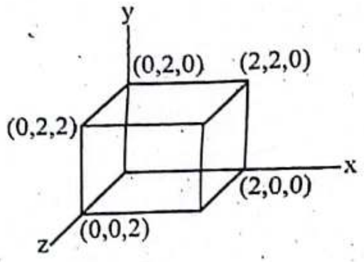
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 & 0 \\ 0 & \cdot 1 & 0 & 0 \\ 0 & sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$sh_x = sh_y = 3$

A = (2, 0, 0) $\qquad$ B = (2, 2, 0) $\qquad$ C = (0, 2, 0)

D = (0, 2, 2)   E = (0, 0, 2)   F = (0, 2, 2)

G = (2, 2, 2)   H = (0, 0, 0)



$$A' = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

A' = (2,0,0)

Similarly, we can calculate B', C', D', E', G', H'.

B' = (8,2,6)   C' = (6,2,6)   D' = (6,2,6)

E' = (0,0,0)   F' = (6,2,6)   G' = (8,2,6)

H' = (0,0,0)

The final coordinates of the cube after shearing are A'(2,0,0), B'(8,2,6), C'(6,2,6), D'(6,2,6), E'(0,0,0), F'(6,2,6), G'(8,2,6), and H'(0,0,0).

5. **List down the steps for rotating a 3D object by 90° in counter clockwise direction about an axis joining end points (1,2,3) and (10,20,30). Also derive the final transformation matrix.**

Solution:

i. Translate (1,2,3) to origin so that the rotation axis passes through the origin.

ii. Rotate the line so that the line coincides with one of the axes, say 2 axis.

iii. Rotate the object about the co-ordinate axis by 90° in counter clockwise direction.

iv. Apply the inverse of step (ii) i.e. inverse rotation to bring the rotation axis back to its original orientation.

v. Apply the inverse of step (i) i.e. inverse translation to bring the rotation axis back to its original position.

For step (ii) i.e. for coinciding the arbitrary axis with any co-ordinate axis, the rotations are needed about other two axes.

Direction cosines of the given line is

$[v] = [(x_1 - x_0)\ (y_1 - y_0)\ (z_1 - z_0)]$

$$[c_x\ c_y\ c_z] = \left[\frac{(x_1 - x_0)\ (y_1 - y_0)\ (z_1 - z_0)}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}}\right]$$

$$c_x = \frac{x_1 - x_0}{|v|} = \frac{10 - 1}{\sqrt{9^2 + 18^2 + 27^2}} = \frac{9}{33.67}$$

$$c_y = \frac{y_1 - y_0}{|v|} = \frac{18}{33.67}$$

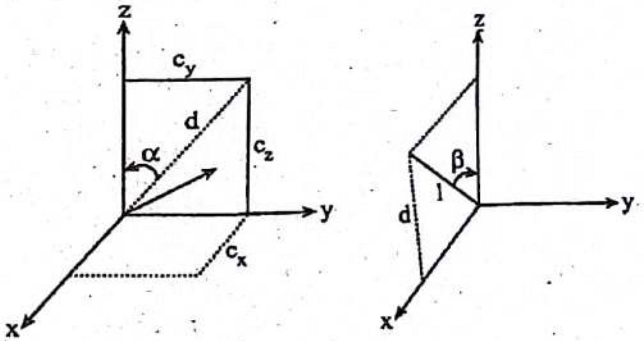
$$c_z = \frac{z_1 - z_0}{|v|} = \frac{27}{33.67}$$

To calculate the angles of rotation about x and y axes we use the direction cosines.

To put the line or rotation axis on the z axis we have to follow two steps.

a. First rotate about the x axis to transform vector u into the x z plane.

b. The swing u around to the z axis using 4 axis rotation.



$d = \sqrt{c_y^2 + c_z^2}$

$\cos\alpha = \frac{c_z}{d}$ $\sin\alpha = \frac{c_y}{d}$

$\cos\beta = d$ $\sin\beta = -c_x$

The complete the sequence of operations can be summarized as

$T = [T_r]^{-1}[R_x(\alpha)]^{-1}[R_y(\beta)]^{-1}[R_z(\theta)][R_y(\beta)][R_x(\alpha)][T_r]$

$$[T_r] = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_x(\alpha)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_y(\beta)] = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 1 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_z(\theta)] = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ & 0 & 0 \\ \sin 90^\circ & \cos 90^\circ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_y(\beta)] = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 1 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_x(\alpha)]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[T_r]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6. **Obtain perspective projection co-ordinate for the pyramid with vertices of base (15,15,10), (20,20,10), (25,15,10), (20,10,10) and apex (20,15,20) given that $z_{prp} = 20$ and $z_{vp} = 0$.**

Solution:

$z_{prp} = 20$, $z_{vp} = 0$

$x' = x - xu$

$y' = y - yu$

$z' = z - (z - z_{prp})u$

On the view plane,

$z_{vp} = z - (z - z_{prp})u$

$$u = \frac{z_{vp} - z}{z_{prp} - z}$$

For the vertex (15,15,10),

$$x_p = x - x\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$= x\left(\frac{z_{prp} - z - z_{vp} + z}{z_{prp} - z}\right)$$

$$= x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)$$

$$= 15\left(\frac{20 - 0}{20 - 10}\right)$$

$$= 30$$

$$y_p = y - y\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$= 10 - 10\left(\frac{0 - 10}{20 - 10}\right)$$

$$= 30$$

$z_p = z_{vp} = 0$

Projected points is $(x_1', y_1', z_1') = (30,30,0)$

Similarly for $P_2(X_2, Y_2, Z_2) = P_2(20,20,10)$

$$X_p = 20\left(\frac{20 - 0}{20 - 10}\right) = 40$$

$$Y_p = 20\left(\frac{20 - 0}{20 - 10}\right) = 40$$

$Z_p = 0$

Projected points is $(X'_2, Y_2', Z_2') = (40,40,0)$

Similarly, for vertex (25,15,10)

$X_p = 50$

$Y_p = 30$

$Z_p = 0$

Projected points is $P_3(X'_3, Y_3', Z_3') = (50,30,0)$

For vertex (20,10,10)

$$X_p = x - x\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$= 20 - 20\left(\frac{0 - 10}{20 - 10}\right)$$

$$Y_p = y - y\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$= 10 - 10\left(\frac{0 - 10}{20 - 10}\right) = 20$$

$Z_p = 0$

For apex (20,15,20)

$$X_p = X\left(\frac{d_p}{z_{prp} - z}\right)$$

$$= 20\left(\frac{20}{20 - 20}\right) = \infty$$

$$Y_p = Y\left(\frac{d_p}{z_{prp} - z}\right)$$

$$= 15\left(\frac{20}{20 - 20}\right) = \infty$$

$Z_p = 0$

7. **A unit length cube with diagonal passing through (0,0,0) and (1,1,1) is shared with respect to yz-plane with share constants = 2 in both directions. Obtain the final coordinates of the cube after shearing.** **[2078 Bhadra]**

Solution:

Shearing with respect to zx-plane

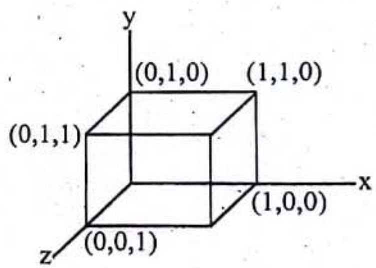
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ sh_y & 1 & 0 & 0 \\ sh_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$sh_x = sh_y = 2$

A = (1,0,0) B = (1,1,0) C = (0,1,0)

D = (0,1,1) E = (0,0,1) F = (0,1,1)

G = (1,1,1) H = (0,0,0)



$$A' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

A' = (1,2,2)

Similarly, we can calculate B', C', D', E', F', G', H'

B' = (1,3,2) C' = (0,1,0) D' = (0,1,0)

E' = (0,0,0) F' = (0,1,0) G' = (1,3,2)

H' = (0,0,0)

The final co-ordinates of the cube after shearing are A'(1,2,2), B'(1,3,2), C'(0,1,0), D'(0,1,0), D'(0,1,0), E'(0,0,0), F'(0,1,0), G'(1,3,2), and H'(0,0,0).

8. **A parametric cubic curve passes through the point (0,0), (2,4), (4,3) and (5,–2) which are parameterized at t = 0, $\frac{1}{4}$, $\frac{3}{4}$ and 1 respectively. Determine the geometric coefficient matrix and the slope of the curve when t = 0.5**

Solution:

The points on the curve are

(0,0) at t = 0 (2,4) at $t = \frac{1}{4}$

(4,3) at $t = \frac{3}{4}$ (5,–2) at t = 1

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

In matrix form the equation can be written as.

$$P(t) = [t^3 \ \ t^2 \ \ t \ \ 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p(0) \\ p(1) \\ p'(0) \\ p'(1) \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 2 & 4 \\ 4 & 3 \\ 5 & -2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ \left(\frac{1}{4}\right)^3 & \left(\frac{1}{2}\right)^2 & \left(\frac{1}{4}\right) & 1 \\ \left(\frac{3}{4}\right)^3 & \left(\frac{3}{4}\right)^2 & \left(\frac{3}{4}\right) & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p(0) \\ p(1) \\ p'(0) \\ p'(1) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.0156 & 0.0625 & 0.25 & 1 \\ 0.4218 & 0.5625 & 0.75 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p(0) \\ p(1) \\ p'(0) \\ p'(1) \end{bmatrix}$$

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

The geometric coefficients are:

p(0) = (0,0)

p(1) = (5,–2)

p'(0) = (10.33,22)

p'(1) = (4.99,–26)

The slope at t = 0.5 is found by taking the first derivative of the above equation as follows.

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

Therefore

P'(0.5) = [3.67 –2.0]

$$\text{Slope} = \frac{-2.0}{3.67} = -0.545$$

9. **Find equation of Bezier curve whose control points are $P_0(2,6)$, $P_1(6,8)$ and $P_2(9,12)$. Also find co-ordinate of point at u = 0.8.** **[2071 Chaitra]**

Solution:

Control points are $P_0(2,6)$, $P_1(6,8)$ and $P_2(9,12)$

Number of control points = 3

A Bezier curve is a polynomial of degree one less than the number of control points.

Degree of polynomial = 3 – 1 = 2

u = 0.8

$$P(u) = \sum_{k=0}^{n} p_k \quad BEZ_{k,n}(u)$$

n = 2, so,

$$P(u) = \sum_{k=0}^{2} p_k \quad BEZ_{k,2}(u)$$

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

$$C(2,k) = \frac{2!}{k!(2-k)!}$$

$$P_x(u) = \sum_{k=0}^{2} x_k BEK_{k,2}(u)$$

$$BEZ_{0,2}(u) = C(2,0)u^0(1-u)^2 = (1-0.8)^2$$

$$BEZ_{1,2}(u) = C(2,1)u^1(1'-u)^1 = 2 \times (0.8)^1 \times (1-0.8)^1$$

$$BEZ_{0,2}(u) = C(2,0)u^2(1-u)^0 = (0.8)^2 \times (1-0.8)^0$$

$$P_x(u) = x_0BEZ_{0,2}(u) + x_1BEZ_{1,2}(u) + x_2BEZ_{2,2}(u)$$

$$= 2 \times (1-0.8)^2 + 6 \times 2 \times (0.8)^1 \times (1-0.8)^1 + 9 \times (0.8)^2 \times (1-0.8)^0$$

$$= 7.76$$

$$P_y(u) = y_0BEZ_{0,2}(u) + y_1BEZ_{1,2}(u) + y_2BEZ_{2,2}(u)$$

$$= 6\times(1-0.8)^2 + 8 \times 2 \times (0.8)^1 \times (1-0.8)^1 + 12 \times (0.8)^2 \times (1-0.8)^0$$

$$= 10.48$$

The coordinate point is (7.76, 10.48).

10. **Find the Bezier curve which passes through (0,0,0) and (–2,1,1) and is controlled by (7,5,2) and (2,0,1).** **[2076 Ashwin Back]**

Solution:

In Bezier curve the starting and ending control points lie on the curve so (0,0,0) and (–2,1,1) are starting and ending control points and (7,5,2) and (2,0,1) are intermediate control points.

We know,

$$\text{Position vector } P(u) = \sum_{k=0}^{n} P_k BEZ_{k,n}(u) \text{ for } 0 \le u \le 1.$$

Where,

P_k is control point position

We have 4 control points so P_k varies k = 0 to 3

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

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

where, C(n,k) are binomial coefficient

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

So,

$P_x(u) = x_0BEZ_{0,3}(u) + x_1BEZ_{1,3}(u) + x_2BEZ_{2,3}(u) + x_3BEZ_{3,3}(u)$

$P_y(u) = y_0BEZ_{0,3}(u) + y_1BEZ_{1,3}(u) + y_2BEZ_{2,3}(u) + y_3BEZ_{3,3}(u)$

$P_z(u) = z_0BEZ_{0,3}(u) + z_1BEZ_{1,3}(u) + z_2BEZ_{2,3}(u) + z_3BEZ_{3,3}(u)$

Where,

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

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

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

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

Now,

$P_x(u) = x_0BEZ_{0,3}(u) + x_1BEZ_{1,3}(u) + x_2BEZ_{2,3}(u) + x_3BEZ_{3,3}(u)$

$= 0 \times (1-u) + 7 \times 3u(1-u)^2 + 2 \times 3u^2(1-u) + (-2) \times u^3$

$P_y(u) = y_0BEZ_{0,3}(u) + y_1BEZ_{1,3}(u) + y_2BEZ_{2,3}(u) + y_3BEZ_{3,3}(u)$

$= 0 \times (1-u) + 5 \times 3u(1-u)^2 + 0 \times 3u^2(1-u) + 1 \times u^3$

$P_z(u) = z_0BEZ_{0,3}(u) + z_1BEZ_{1,3}(u) + z_2BEZ_{2,3}(u) + z_3BEZ_{3,3}(u)$

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

11. A cubic Bezier curve is described by the four control points, (0,0), (2,1), (5,2) and (6,1) find the tangent to the curve at t = 0.5

Solution:

Here, we know the Bezier cubic polynomial equation.

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

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

$P'(t) = -3\times(1-t)^2P_0 + 6t(1-t)P_1 - 6tP_1 + 3t(1-t)^2P_1 - 3t^2P_2 + 6t(1-t)P_2 + 3t^2P_3$

$x'(t) = -3\times(1-t)^2x_0 - 6t(1-t)x_1 + 3(1-t)^2x_1 + 6t(1-t)x_2 - 3t^2x_2 + 3t^2x_3$

$= -3\times(1-0.5)^2(0) - 6(0.5)(1-0.5)(2) + 3(1-0.5)^2(2) + 6(0.5)(1-0.5)(5) - 3(0.5)^2(5) + 3(0.5)^2(6)$

$= 6.75$

$y'(t) = 3\times(1-t)^2y_0 + 6t(1-t)y_1 - 6tP_1 + 3t(1-t)^2y_1 - 3t^2y_2 + 6t(1-t)y_2 + 3t^2y_3$

$= 1.5$

Or we can solve it by alternative method,

In matrix form this equation is written as:

$$P(t) = [t^3 \; t^2 \; t \; 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

where,

$v_0 = (0,0)$ $\quad v_1 = (2,1)$

$v_3 = (5,2)$ $\quad v_4 = (6,1)$

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

$$P'(t) = [3t^2 \; 2t \; 1 \; 0]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

At t = 0.5, we get

$$P'(t) = [3(0.5)^2 \; 2(0.5) \; 1 \; 0]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$= [0.75 \; 1 \; 1 \; 0]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$= [-0.75 \; -0.75 \; 0.75 \; 0.75]\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$= [-0.75 \; -0.75 \; 0.75 \; 0.75]\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$= [6.75 \; 1.5] \quad = \frac{1.5}{6.75} = 0.222$$

Tangent = 0.222

Tangent angle = 12.53°

12. **Design a Bezier curve controlled by points A(1,1), B(2,3), C(4,3) and D(6,4).**

Solution:

Control point = 4

Degree of polynomial = n – 1 = 3

$A(1,1) = p_0$

$B(2,3) = p_1$

$C(4,3) = p_2$

$D(6,4) = p_3$

Equation of Bezier curve

$$p(u) = \sum_{k=0}^{n} p_k B_{k,n}(u)$$

where p(u) = position vector

p_k = control point

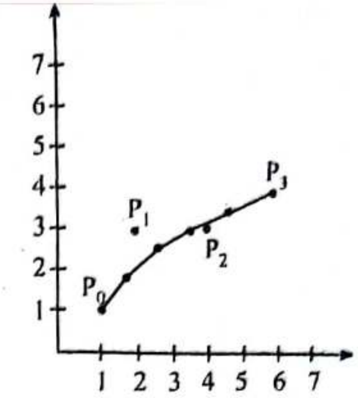
$$B_{k,n} = \frac{n!}{k!(n-k)!} u^k (1-u)^{n-k}$$

$$B_{0,3}(u) = (3,0)u^0.(1-u)^{3-0}$$
$$= \frac{3!}{0!3!}(1-u)^3$$
$$= (1-u)^3$$
$$B_{1,3}(u) = C(3,1)u^1(1-u)^{3-1}$$
$$= 3u.(1-u)^2$$
$$B_{2,3}(u) = C(3,2).u^2(1-u)^{3-2}$$
$$= 3u^2(1-u)$$
$$B_{3,3}(u) = C(3,3)u^3(1-u)^{3-3} = u^3$$
$$P(u) = p_0(1-u)^3 + p_1.3u(1-u)^2 + p_2 3u^2(1-u) + p_3 u^3$$

To find x(u) and y(u)

$$x(u) = x_0(1-u)^3 + x_1.3u(1-u)^2 + x_2 3u^2(1-u) + x_3 u^3$$
$$y(u) = y_0(1-u)^3 + y_1.3u(1-u)^2 + y_2 3u^2(1-u) + y_3 u^3$$

u	x(u)	y(u)
0	1	1
0.2	0.712	1.984
0.4	2.616	2.632
0.6	3.664	3.088
0.8	4.808	3.496
1	6	4



13. **Find the coordinates of Bezier curve at u = 0.25, 0.5 and 0.75 with respect to the control points (10,15), (15,20), (20,35), (25,10) using Bezier function.** **[2079 Baishakh]**

Solution:

Here, given,

Total number of control point = 4

Then degree of polynomial = n – 1 = 3

$A(10,15) = p_0$

$B(15,20) = p_1$

$C(20,35) = p_2$

$D(25,10) = p_3$

Equation of Bezier curve

$$p(u) = \sum_{k=0}^{n} p_k B_{k,n}(u)$$

where p(u) = position vector

p_k = control point

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

$$B_{0,3}(u) = (3,0)u^0.(1-u)^{3-0}$$
$$= \frac{3!}{0!3!}(1-u)^3$$
$$= (1-u)^3$$
$$B_{1,3}(u) = C(3,1)u^1(1-u)^{3-1}$$
$$= 3u.(1-u)^2$$
$$B_{2,3}(u) = C(3,2).u^2(1-u)^{3-2}$$
$$= 3u^2(1-u)$$

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

$P(u) = p_0(1-u)^3 + p_1.3u(1-u)^2 + p_2 3u^2(1-u) + p_3 u^3$

To find x(u) and y(u)

$x(u) = x_0(1-u)^3 + x_1.3u(1-u)^2 + x_2 3u^2(1-u) + x_3 u^3$

$y(u) = y_0(1-u)^3 + y_1.3u(1-u)^2 + y_2 3u^2(1-u) + y_3 u^3$

u	x(u)	y(u)
0	10	15
0.25	13.75	19.84375
0.5	17.5	23.75
0.75	21.25	22:03125
1	25	10

The coordinates of Bezier curve at u = 0.25, 0.5 and 0.75 are (13,75, 19.84375), (17.5, 23.75) and (21.25, 22.03125) respectively.

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

Step by step solution

For four control points, n = 3.

First calculate all the blending functions, B_{kn} for k=0,...,n using the formula

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

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

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

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

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

Numerical calculations are shown below:

u = 0.0

$$x(0) = \sum_{k=0}^{n} x_k B_{kn}(0)$$

$$= x_0B_{03}(0) + x_1B_{13}(0) + x_2B_{23}(0) + x_3B_{33}(0)$$

$$= 0.(1-u)^3 + 1.3u.(1-u)^2 + 3.3u^2(1-u) + 4.u^3$$

$$= 0.1 + 1.0 + 3.0 + 4.0$$

$$= 0$$

$$y(0) = \sum_{k=0}^{n} y_k B_{kn}(0)$$

$$= y_0B_{03}(0) + y_1B_{13}90) + y_2B_{23}(0) + y_3B_{33}(0)$$

$$= 0.(1-u)^3 + 2.3u.(1-u)^2 + 3.3u^2(1-u) + 0.u^3$$

$$= 0$$

u = 0.2

$$x(0.2) = \sum_{k=0}^{n} x_k B_{kn}(0.2)$$

$$= x_0B_{03}(0.2) + x_1B_{13}(0.2) + x_2B_{23}(0.2) + x_3B_{33}(0.2)$$

$$= 0.(1-u)^3 + 1.3u.(1-u)^2 + 3.3u^2(1-u) + 4.u^3$$

$$= 0.0.512 + 1.0.384 + 3.0.096 + 4.0.008$$

$$= 0.7$$

$$y(0.2) = \sum_{k=0}^{n} y_k B_{kn}(0.2)$$

$$= y_0B_{03}(0.2) + y_1B_{13}(0.2) + y_2B_{23}(0.2) + y_3B_{33}(0.2)$$

$$= 0.(1-u)^3 + 2.3u.(1-u)^2 + 3.3u^2(1-u) + 0.u^3$$

$$= 0.0.512 + 2.0.384 + 3.0.096 + 0.0.008$$

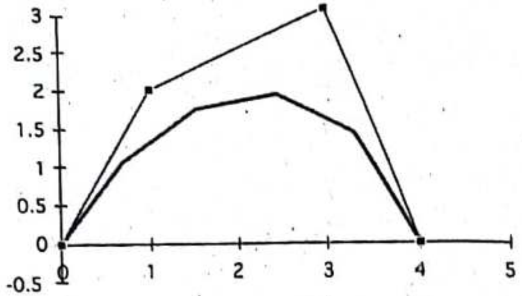
$$= 1.1$$

etc, giving:

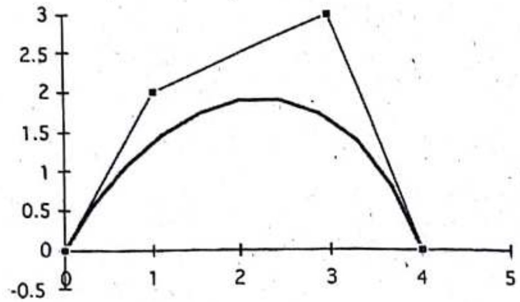
u = 0.0 x(u) = 0.0	y(u) = 0.0
u = 0.2 x(u) = 0.7	y(u) = 1.1
u = 0.4 x(u) = 1.55	y(u) = 1.7
u = 0.6 x(u) = 2.45	y(u) = 1.9
u = 0.8 x(u) = 3.3	y(u) = 1.3
u = 1.0 x(u) = 4.0	y(u) = 0.0

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

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



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



Unit – 6

VISIBLE SURFACE DETECTION

Visible surface detection

Visible surface detection, also known as visible surface determination or hidden surface removal, is a fundamental problem in computer graphics. It refers to the process of determining which surfaces or parts of surfaces in a three-dimensional (3D) scene are visible from a given viewpoint and should be displayed, while occluded or hidden surfaces are not rendered.

The goal of visible surface detection is to ensure that only the surfaces that contribute to the final image are rendered, improving the realism and efficiency of 3D graphics rendering.

- When we view a picture containing non-transparent objects and surfaces, then we cannot see those objects from view which are hidden from another the objects.
- These surfaces that are hidden or blocked from view must be removed to get a realistic view of 3D scene.
- The identification and removal of these surfaces is called **Hidden-surface problem.**
- The problem of hidden surface removal is solved by applying different algorithms.
- **These algorithms are broadly classified into two categories:**
  - Object space methods
  - Image space methods

1. **Object Space Method**
   - Implemented in the world co-ordinate system, considering the geometrical relationship between the actual objects.
   - In this method, various parts of objects are compared.
   - After comparison visible, invisible or hardly visible surface is determined.