

## B.Tech.

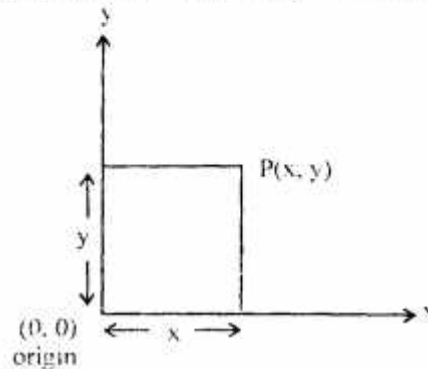
Fifth Semester Examination

### Computer Graphics (CSE-303-F)

#### Short Questions.

Q. 1. (a) What do you understand by point?

Ans. **Point** : A position in a plane is known as point and point can be represented by any ordered pair of number  $(x, y)$ , where  $x$  is horizontal distance from origin and  $y$  is vertical distance from origin.

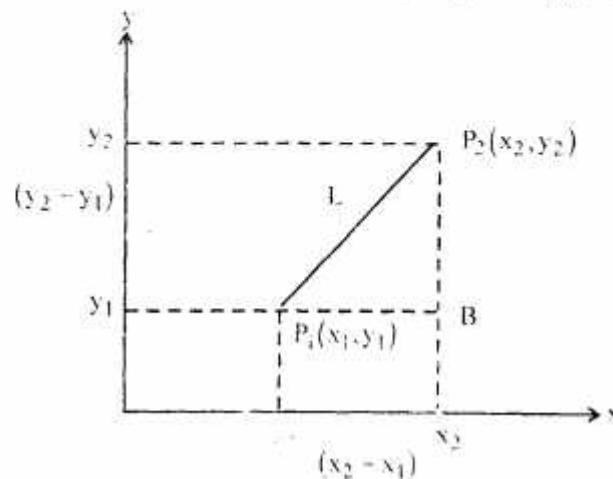


Q. 1. (b) What do you mean by line segment.

Ans. **Line Segment** : Any line or piece of line having end points is called the line segment. We can find the equation of any line segment using its end points and it can easily be checked that any point lies on the line segment or not point will be on the line segment if :

- (i) Point satisfies the equation of segment.
- (ii)  $x$  co-ordinate of point lies between  $x$  coordinate of end points.
- (iii)  $y$  co-ordinate of point lies between  $y$  coordinate of end points.

**Length of Line Segment** : There is a line segment having end points  $P_1(x_1, y_1)$  and  $P_2(x_2, y_2)$ .



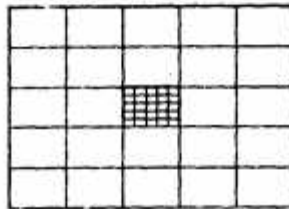
$$(P_1B)^2 + (BP_2)^2 = L^2$$

$$L^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2$$

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad \text{Ans.}$$

**Q. 1. (c) What is pixels?**

**Ans. Pixels :** Any screen is divided into small screen points and this small screen point is smallest part of the screen. is referred as a pixel or pet. Each pixel has it own intensity, name or address by which we can control it.



**Q. 1. (d) What is segment?**

**Ans. Segment :** The segment is a unit of display fill. Each segment corresponds to a component of the file. Thus, the use of segments by the programmer requires him to recognize the existence of the display file as a stored representation of the displayed image.

The segment is a logical unit, not necessary contiguous either in the display file or on the screen. It is simply a collection of display file instructions representing graphics primitives that we can manipulated as a single unit. We therefore need function to perform these manipulations and a naming scheme so that we can refer to each segment unambiguously.

Segment name	Segment start	Segment size	Visibility	Scale-X.....
0				
1				
2				
3				

*Segment Stable*

**Q. 1. (e) What do you understand by homogeneous coordinate?**

**Ans. Homogeneous Co-ordinates :** The term homogeneous co-ordinates is used in mathematics to refer to the effect of this representation on Cartesian equations. When a Cartesian point (x, y) is converted to a homogeneous representation  $(x_h, y_h, h)$ , equations containing x and y, such as  $f(x, y) = 0$ , becomes homogeneous equation in three parameters  $x_h$ ,  $y_h$  and h. This just means that if each of the three parameters is replaced by any value V times that parameter, the value V can be factored out the equations.

Expressing positions in homogeneous coordinates allow us to represent all geometric transformation equations as matrix multiplications. Coordinates are represented with three element column vectors and transformation operations are written as 3 by 3 matrices. For translation,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Which we can write in abbreviated form as :

$$P' = T(t_x, t_y) \cdot P \quad \text{Ans.}$$

**Q. 1. (f) What do you mean by mirror reflection?**

**Ans. Mirror Reflection :** Mirror reflection in three-dimensional geometry is also any analogous to that of two dimensional mirror reflection. There are following 3 cases :

Reflection with respect to xy plane

Reflection with respect to yz plane

Reflection with respect to xz plane

The point after reflection can be calculated from the following equation :

$$P'(x', y', z') = M \cdot P(x, y, z)$$

Where,  $x' = x, y' = y, z' = z$

$$P' \quad M_{xy} \quad P$$

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

Similarly reflection with respect to xz plane,

$$P' \quad M_{xy} \quad P$$

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

and reflection with respect to yz plane,

$$P' \quad M_{xy} \quad P$$

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

**Q. 1. (g) Explain echoing.**

**Ans. Echoing :** A important part of an interactive system is echoing. Echoing provides the user with formation about his actions. This allows the user to compare what he has done against what he wanted to do. For keyboard input, echoing usually takes. The form of displaying the typed character. Locator may be echoed by a screen cursor displayed at the current locator position. This allows the user to see the current locator setting and to relate its position to the objects on the display.



*Echoing*

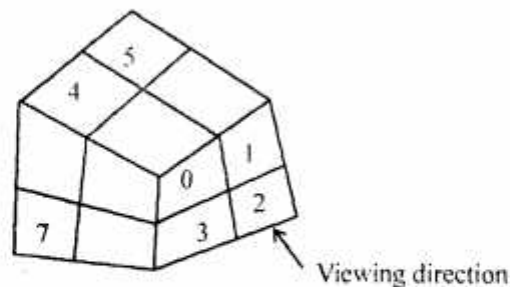
A pick may be echoed by identifying the selected objects on the display. The selected objects may be flashed or made brighter or perhaps altered in color. This will allow the user determine whether or not he has selected the intended objects.



*Echoing a Pick*

**Q. 1. (h) What is octree? Also explain BSP.**

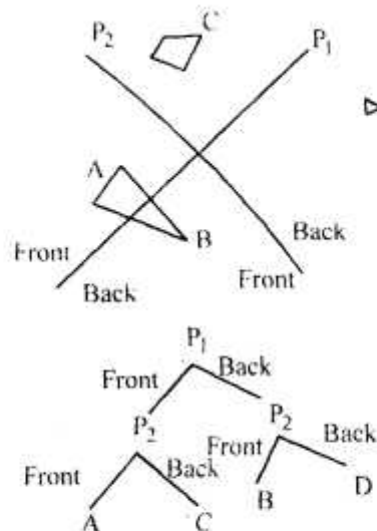
**Ans. Octree Method :** When an octree representation is used for the viewing volume, hidden surface elimination is accomplished projecting octree nodes onto the viewing surface in front to back order.



Surfaces in front of these octants are visible to viewer. Any surface to rear of the front octants or in the back octants (4, 5, 6, and 7) may be hidden by front surfaces.

**BSP Tree Method :** A binary space partitioning (BPS) tree is an efficient method for determine object visibility by painting surfaces onto the screen from back to front, as in painter's algorithm. The BSP tree is particularly useful when the view reference point changes, but the objects in a scene are at fixed positions.

Applying a BSP tree to visibility testing involves identifying surfaces that are "inside" and "outside" the partitioning plane at each step of space subdivision, relative to viewing direction.



**Q. 1. (i) What do you understand by spline representation.**

**Ans. Spline Representation :** A spline is a flexible strip used to produce a smooth curve through a designated set of points. The term spline curve referred to a curve with a piecewise cubic polynomial. Function whose first and second derivatives are continuous across the various curve sections. In computer graphics, the term. Spline curve refers to any composite curve formed with polynomial sections satisfying specified continuity condition at boundary of pieces.

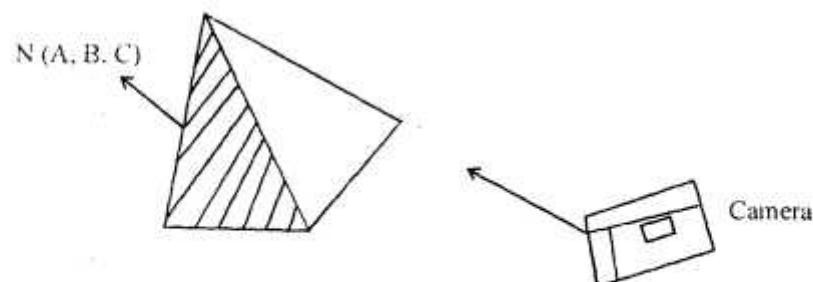
A spline surface can be describe with two sets of dorthogonal spline curves splines are used is graphics applications to design curve and surface shapes, to digital drawing for computer storage and to specify animation paths for objects or the camera in a scene.

**Q. 1. (j) What is back face removal algorithm?**

**Ans. Back Face Detection :** A fact and simple object-space method for identifying the back face of a polyhedron is based the "inside outside" test. A point (x, y, z) is "inside" a polygon surface with plane parameter A, B, C are D if,

$$[Ax + By + Cz + D < 0]$$

When an inside point is along the line of sight to the surface, the polygon must be back-face.



We can simplify this test by considering normal vector N to a polygon surface, which has Cartesian component (A, B, C). If vector V is a vector in the viewing direction from the eye position then this polygon is



a back face if,

$$[V.N > 0]$$

If object description has been converted to projection co-ordinates and our viewing direction is parallel to viewing  $Z_v$  axis then,

$$V = (0, 0, V_z) \text{ and } V.N = V_z C$$

So that we only need to consider the sign  $C$  the  $Z$  components of normal vector  $N$ .

In right handed viewing system with view in direction along the negative  $Z_v$  axis the polygon is a back face if  $C < 0$ .

#### Section-A

**Q. 2. (i) Find the distance between point  $(x_0, y_0)$  and a line  $rx + sy + t = 0$  in a plane.**

**Ans. Distance Between Point and Line :** Let a line described by the equation  $rx + sy + t = 0$  then we have to find distance from the point  $(x_0, y_0)$  to this line.

Now,  $rx + sy + t = 0$

$$y = -\frac{r}{s}x - \frac{t}{s}$$

$$m_1 = -\frac{r}{s}$$

We know that,  $m_1 m_2 = -1$

$$m_2 = \frac{s}{r}$$

Then equation of perpendicular line,

$$y = \frac{s}{r}x + c_1$$

Where  $c_1$  is any constant line passing through point  $(x_0, y_0)$  so, it will satisfy the line.

$$y_0 = \frac{s}{r}x_0 + c_1$$

$$c_1 = -\frac{s}{r}x_0 + y_0$$

$$\boxed{y = \frac{s}{r}x + \left(y_0 - \frac{s}{r}x_0\right)}$$

This is the equation of line perpendicular to the line  $rx + sy + t = 0$ , because they are intersecting to each other at point. Let  $N(x_1, y_1)$  calculate  $(x_1, y_1)$ .

$$x_1 = \frac{s^2 x_0 - rt - rsy_0}{r^2 + s^2}$$

$$y_1 = \frac{-rs^2 x_0 + rsy_0 - s^2 t}{r^2 + s^2}$$

Now the distance  $(x_0, y_0)$  and  $(x_1, y_1)$  is

$$d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$

$$d = \left[ \frac{(rx_0 + sy_0 + t)^2}{r^2 + s^2} \right]^{1/2}$$

$$d = \frac{rx_0 + sy_0 + t}{\sqrt{r^2 + s^2}}$$

Ans.

**Q. 2. (ii) What do you understand by plane? Also define the implicit form and explicit form of plane?**

**Ans. Planes :** A plane can be expressed by a linear equation, it is divided into two classes :

(i) Implicit form

(ii) Explicit form

**(i) Implicit Form :** Implicit form of the equation of a plane can be used to determine whether two points are on the same or opposite sides of the plane.

Implicit form of equation of plane,

$$Ax + Bx + Cz + D = 0$$

$$f(x, y, z) = Ax + By + Cz + D$$

Two sides of plane  $R^+$ ,  $R^-$  are determined by sign of  $f(x, y, z)$  i.e., point  $P(x_0, y_0, z_0)$  lies on the region  $R^+$  if  $f(x_0, y_0, z_0) > 0$  and region  $R^-$  if  $f(x_0, y_0, z_0) < 0$  if  $f(x_0, y_0, z_0) = 0$  the point lies on the plane.

**Explicit Form :** Explicit form of equation is given by the following :

$$Z = ax + by + c$$

**Q. 3. Implement the DDA algorithm to draw a line from (0, 0) to (6, 6). Explain vector generation of line.**

**Ans. Vector Generation :** Since intercept of line is  $y = mx + b$ .

$m$  = Slope of line

$b$  = Intercept of y axis.

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

$$b = y_1 - mx_1$$

$$y_2 - y_1 = \Delta y, \quad x_2 - x_1 = \Delta x$$

$$m = \frac{\Delta y}{\Delta x}$$

$$\Delta y = m \cdot \Delta x$$

or

$$\Delta x = \frac{\Delta y}{m}$$

Compute initial values:

$$\Delta x = x_2 - x_1$$

$$= 6 - 0 = 6$$

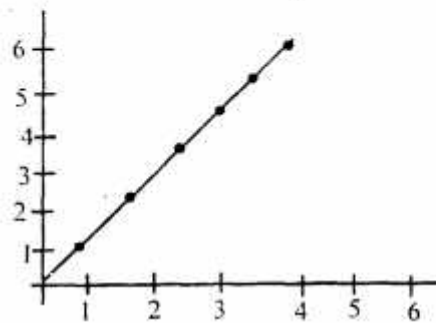
$$\Delta y = y_2 - y_1 = 6 - 0 = 6$$

$$m = \Delta y / \Delta x = 1$$

$$\Delta x = 1$$

$$y_{k+1} = y_k + m$$

Step	x	y	Pixel
	0	0	(0,0)
1	1	1	(1,1)
2	2	2	(2,2)
3	3	3	(3,3)
4	4	4	(4,4)
5	5	5	(5,5)
6	6	6	(6,6)

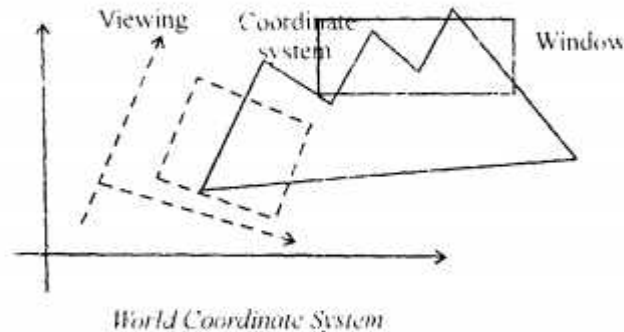




Section-B

Q. 4. (i) What do you understand by world coordinate system?

**Ans. World Coordinate System :** Objects are placed into the scene by modeling transformation to master coordinate system, referred as World Coordinate System (WCS). The WCS is the space in which the picture is defined. A rectangular window with its edges parallel to the axis of WCS is used to select the portion of the scene for which an image is to be generated.



Some time an additional coordinate system called viewing coordinate system is introduced to simulate the effect of moving and/or tilting the camera.

Q. 4. (ii) Write steps for Cohen-Sutherland line clipping algorithm.

**Ans.** When Sutherland algorithm is one of the popular line clipping algorithm. This algorithm immediately the lines which are lying totally outside the window. This algorithm divides the plane in 9 parts and assigns the outcode or binary numbers to each part.

		Above	
	1001	1000	1010
Less	0001	0000	0010
	0101	0100	0110
		Below	
			Right

End point of each line is assigned a four bit binary code which is called as outcode. The highest bit among 4 digits, i.e., 'A' will be set to 1 if end point of a line is above the window. If the end point is not above the window then it is set to 0. Similarly if 'B' bit is set, means the end point is below. If end point is inside the window then it is assigned a code as 0000.

Zero to all 4 bits indicates that the end point is not above, not below, not right and not left of the window.

Q. 5. (i) What do you understand by homogeneous coordinate.

**Ans. Homogeneous Coordinates :** The term homogeneous coordinates is used in mathematics to refer to the effect of this representation on Cartesian equations. When a Cartesian point  $(x, y)$  is converted to a homogeneous representation  $(x_n, y_n, h)$  equations containing  $x$  and  $y$  such as  $f(x, y) = 0$ , becomes homogeneous equation in three parameters  $x_n, y_n$  and  $h$ . This just means that if each of the 3 parameters is replaced by any value  $V$  times that parameter, the value can be factored out of the equations.

Expressing positions in homogeneous coordinates allow us to represent all geometric transformation equations as matrix multiplication. Coordinates are represented with three element column vectors and

transformation operation are written as 3 by 3 matrices,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\boxed{P' = T(t_x, t_y).P} \quad \text{Ans.}$$

**Q. 5. (ii)** Show that composition of two notations is additive by concatenating the matrix representations for  $R(\theta_1)$  and  $R(\theta_2)$  to obtain  $R(\theta_1).R(\theta_2) = R(\theta_1 + \theta_2)$ .

**Ans. Rotations :** Two successive rotations applied to point P produce the transformed position,

$$P' = R(\theta_2). \{R(\theta_1).P\}$$

$$P' = \{R(\theta_2).P(\theta_1)\}.P$$

$$R(\theta_1) = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 \\ \sin \theta_1 & \cos \theta_1 \end{bmatrix}$$

$$R(\theta_2) = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix}$$

$$R(\theta_2).R(\theta_1) = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix}$$

$$[R(\theta_1).R(\theta_2) = R(\theta_1 + \theta_2)]$$

$$\boxed{P' = R(\theta_1 + \theta_2).P}$$

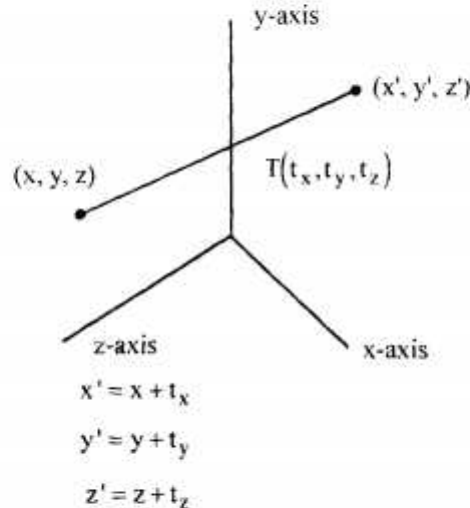
### Section-C

**Q. 6.** What are 3-D transformation? Write matrix of each transformation. Also explain mirror reflection.

**Ans. Translation :** In a three dimensional homogeneous coordinate representation, a point is translated from position  $P = (x, y, z)$  to position  $P'(x', y', z')$  with matrix operation.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$P' = T.P.$$



An object is translated in three dimensions by transforming each of the defining points of objects.

**Rotation :** Rotation in 3-D is more complex than rotation in 2-D. In 2D, a rotation is described by an angle of rotation  $\theta$  and a counter of rotation P. The 3D requires the prescription of an angle of rotation and an axis of rotation. The canonical rotations are defined when one of the positive x, y or z coordinate axis is chosen as axis of rotation.

Rotation about x-axis is,

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

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

$$z' = z$$

Rotation about y-axis ( $x \rightarrow y \rightarrow z \rightarrow x$ )

$$x' = x$$

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

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

Rotation about z-axis ( $x \rightarrow y \rightarrow z \rightarrow x$ )

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

$$y' = y$$

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

$$R_{e,k} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{o,j} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{o,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Scaling :** The motor expression for scaling transformation of position  $P = (x, y, z)$  relative to the coordinate origin can be written as,

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

$$[P = S.P.]$$

Where  $x' = x.S_x$ ,  $y' = y.S_y$ ,  $z' = z.S_z$ .

**Q. 7. (i) Explain B-splines curves. What are the properties of B-spline curves?**

**Ans. B-spline Curve :** B-spline has two advantages over bezier splines :

- (i) The degree of B spline polynomial can be set independently of number of control points.
- (ii) B spline allows local control over the shape.

The trade-off is that B-splines are more complex than bezier splines.

We can write a general expression for the calculation of coordinate positions along a B-spline curve is a blending function formulation as :

$$P(U) = \sum_{k=0}^n P_k B_{k,d}(U)$$

$$U_{min.} \leq U \leq U_{max.}, 2 \leq d \leq n+1$$

Where,  $P_k$  are an input set of  $n+1$  control points. There are several differences between this B spline formulation and that for Bezier splines. The range of parameter  $u$  now depends on how we choose are B-spline parameters. And the B-spline blending function  $B_{k,d}$  are polynomials of degree  $d-1$ .

Blending functions for B-spline curves are defined by CO<sub>x</sub>-de Boor recursion formulas :

$$B_{k,1}(U) = \begin{cases} 1 & \text{if } U_k \leq u \leq U_{k+1} \\ 0 & \text{Otherwise} \end{cases}$$

$$B_{k,d}(U) = \frac{U - U_k}{U_{k+d-1} - U_k} B_{k,d-1}(U) + \frac{U_{k+d} - U}{U_{k+d} - U_{k+1}} B_{k+1,d-1}(U)$$

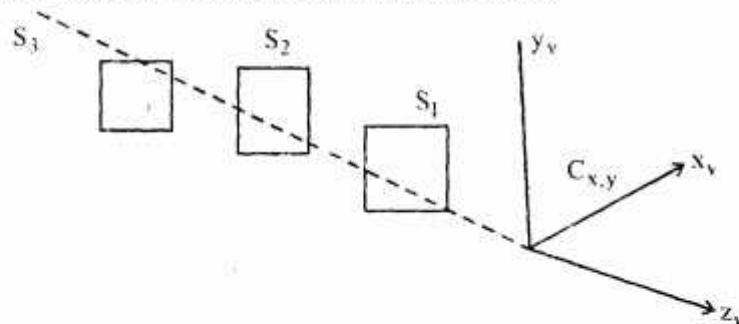
**Properties of B-Splines Curves :**

- (i) The polynomial curve has degree  $d-1$  and  $C^{d-2}$  continuity over the range.
- (ii) For  $n+1$  control points, the curve is described with  $n+1$  blending functions.
- (iii) Each blending function  $B_{k,d}$  is defined over  $d$  subintervals of the total range of  $u$ , starting at knot value  $U_k$ .
- (iv) The range of parameter  $u$  is divided into  $n+d$  subintervals of  $n+d+1$  values specified in the knot vector.
- (v) Each section of spline curve is influenced by  $d$  control point.

**Q. 7. (ii) What do you understand by z-buffer algorithm?**

**Ans. Z-buffer Algorithm :** A commonly used image space approach to detecting visible surfaces in the depth buffer method, which compare surface depth at each pixel position on the projection plane. This procedure is also referred to as z-buffer method, since object depth is measured from the view plane along the z-axis of a viewing system. Each surface of a scene is processed separately. One point at a time across the surface. The method is applied to scenes containing only polygon surfaces, because depth values can be computed very quickly.

With object description converted to projection coordinates, each  $(x, y, z)$  position on a polygon surface corresponds to the orthographic projection point  $(x, y)$  on the view plane. Therefore, for each pixel position  $(x, y)$  on view plane, object depth can be compared by comparing  $z$  values.



**Step 1 :** Initialize the depth buffer and refresh buffer so that for all buffer positions  $(x, y)$ .

$$\text{Depth}(x, y) = 0 \text{ refresh}(x, y) = I_{\text{background}}$$

**Step 2 :** For each position on each polygon surface, compare depth values to previously stored values in depth buffer to determine visibility.

- (i) Calculate the depth  $z$  for each  $(x, y)$  position on polygon.
- (ii) If  $z < \text{depth}(x, y)$  then set  $\text{depth}(x, y) = z$  refresh  $(x, y) = I_{\text{surf}}(x, y)$ .



Where  $I_{background}$  is the value for the background intensity and  $I_{surf}(x, y)$  is the projected intensity value for the surface at pixel position  $(x, y)$ . After all surface have been processed, the depth buffer contain depth values for visible surfaces and the refresh buffer contains the corresponding intensity values for those surfaces.

#### Section-D

**Q. 8. (i) What do understand by interpolation.**

**Ans. Control Points :** We specify a spline curve by giving a set of coordinate positions, called control points, which indicates the general shape of the curve. These control points are then fitted with piecewise continuous parametric polynomial functions in on of the two way :

**(i) Interpolate :** When polynomial section are fitted so that the curve passes through each control point, the resulting curve is said to interpolate the set of control points.



**(ii) Approximate :** When the polynomials are fitted to the general control point without necessary passing through any control point the resulting curve is said to approximate the set of control points.

Interpolation curves are said to digitize drawing or to specify animation paths. Approximation curves are commonly used as design tools to structure object surfaces.

**Convex null :** A convex polygon boundary that encloses a set of control points is called the convex null.

**Q. 8. (ii) Explain Bezier curves and properties of Bezier curves.**

**Ans. Bezier Curves :** A bezier curve section can be fitted to any number of control points. Bezier splines are widely available in various CAD systems. The number of control points to be approximated and their relative position determine the degree of Bezier polynomial. As with interpolation splines, a bezier curve can be specified with boundary conditions with a characterizing matrix or with blending functions. For general bezier curves, the blending function specification is most convenient.

Suppose we are given net control points position.  $P_k = (x_k, y_k, z_k)$  with  $k$  varying from 0 to  $n$ . These coordinate point can be blended to produce the following position vector  $P(u)$ , which describes the path of an approximation. Bezier polynomial function between  $P_0$  and  $P_n$ .

$$\left[ P(u) = \sum_{k=1}^n P_k \text{BEZ}_{k,n}(u) \right] \quad 0 \leq U \leq 1 \quad \dots(1)$$

The Bezier functions  $\text{BEZ}_{k,n}(u)$  are Barstein polynomials.

$$\left[ \text{BEZ}_{k,n}(u) = C(n, k) U^k (1-u)^{n-k} \right] \quad \dots(2)$$

Where  $C(n, k)$  are binomial coefficients,

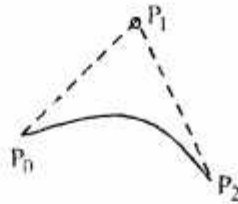
$$\left[ C(n, k) = \frac{n!}{k!(n-k)!} \right]$$

Above equation (1) represents a set of 3 parameter equations for individual coordinates.

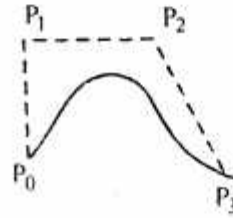
$$X(u) = \sum_{k=0}^n X_k \text{BEZ}_{k,n}(u)$$

$$Y(u) = \sum_{k=0}^n Y_k \text{BEZ}_{k,n}(u)$$

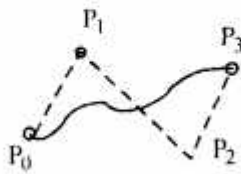
$$Z(u) = \sum_{k=0}^n Z_k \text{BEZ}_{k,n}(u)$$



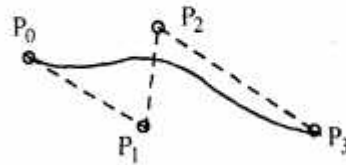
(a)



(b)



(c)



(d)

#### Examples of Bezier Curves

##### Properties of Bezier Curves :

(i) A very useful property of Bezier curve is that it always passes through the first and last control points. That is boundary condition at the two end of the curve are :

$$P(0) = P_0$$

$$P(1) = P_n$$

(ii) The sum of all bezier blending function is equal to 1.

$$\sum_{k=0}^n \text{BEZ}_{k,n}(u) = 1$$

(iii) The convex hull property for a bezier curve ensures that the polynomial only follows the control points without erratic oscillations.

(iv) Another property of Bezier curve is that it lives within the convex hull (convex polygon boundary) of the control points.

Q. 9. Construct enough points on the bezier curve whose control points are  $P_0(4, 2)$ ,  $P_1(8, 2)$  and  $P_2(16, 4)$  to draw an accurate sketch.

(i) What is the degree of curve.

(ii) What are coordinates of  $u = 0.5$ .

Ans. (i) Here  $n = 2$  i.e., order = 2

$$\text{Degree} = \text{order} - 1 = 2 - 1 = 1$$

(ii) Blending function of curve,

$$B_{0,2}(u) = {}^2C_0 u^0 (1-u)^{2-0}$$

$$= (1-u)^2$$

$$B_{1,2}(u) = {}^2C_1 u^1 (1-u)^{2-1}$$

$$= 2u(1-u)$$

$$B_{2,2}(u) = {}^2C_2 u^2 (1-u)^{2-2}$$

$$= u^2$$

Then bezier curve for given control points,

$$C(u) = P_0 B_{0,2}(u) + P_1 B_{1,2}(u) + P_2 B_{2,2}(u)$$

Dissolving the curve x and y coordinate,

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

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

Put	$x_0 = 4$	$y_0 = 2$
	$x_1 = 8$	$y_1 = 8$
	$x_2 = 16$	$y_2 = 16$

$$x(u) = u(1-u)^2 + 8 \times 2u(1-u) + 16u^2$$

$$= 4u^2 - 8u + 4$$

At  $u = 0.5$

$$x(u) = 1$$

$$y(u) = 2(1-u)^2 + 8 \times 2u(1-u) + 4u^2$$

$$= -10u^2 + 12u + 2$$

At  $u = 0.5$

$$y(u) = 5.5$$

Coordinates at  $u = 0.5 \Rightarrow (1, 5.5)$

Ans.