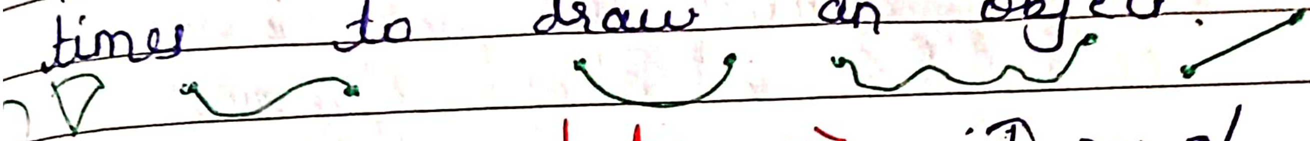
[~~A curve is~~ an [curve is a set of points or pixels].

Curve Generation ⇒ In computer graphics we often need to draw different types of objects onto the screen. Objects are not flat all time & we need to draw curves many times to draw an object.



Curve representation ⇒ 'Types of curves:'

"[The curve is an infinitely large set of points]" Each point has 2 neighbors except endpoints

(i) Explicit curve
ii) Implicit curve
iii) Parametric curve
iv) Bezier curve
v) B-spline curve

(i) **Explicit curve** → $y = fx$ →curve of ^{n}in 2D
(Single value)

where y = dependent variable
x = independent variable

for each value of x, only a single value of y is normally computed by the function

Ex $\boxed{y = f(x)}$ [y की value find करना है तो function of x के form में लिखेंगे] is an ex of exiplicit

$y = mx + c$

2) Implicit ⇒ Multi valued curves

$\boxed{f(x, y) = 0}$

Ex $x^2 + y^2 = r^2$ → eqⁿ of circle

$x^2 + y^2 - r = 0$

3) Parametric ⇒ Most of the curve follows parametric form

$$\boxed{\begin{aligned} x &= f_x(v) \\ y &= f_y(v) \end{aligned}}$$

V = parameter

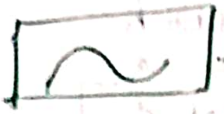
x and y as a funcⁿ [एक निश्चित variable को लेंगे]

$x = a\cos t$

$y = a\sin t$

$x^2 + y^2 = a^2$

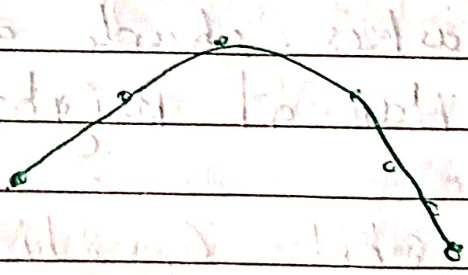
(Baba → Tree) → Piece (fine strip) of this Tree called Spline

 → ship की design करने के लिए

Spline ⇒ It provide a interface ~~facility~~ ~~and ease~~ to easily draw complex curve design and control the shape of curve also.

→ Easily draw complex curve
→ Control (modify) the shape of curve also

→ Control points ⇒ It is a co-ordinate positions for control the shape of the curve.



With the help of this control points we have to generate/design curve using 2 types of technique.

1) Interpolate or Interpolation spline
2) Approximate spline

control point ⇒ हम curve के shape को control कर सकते हैं।

Interpolation spline ⇒ In this we have to follow all control points and design our curve

[इसमें हम सारे control points को use करके curve बनाएंगे]

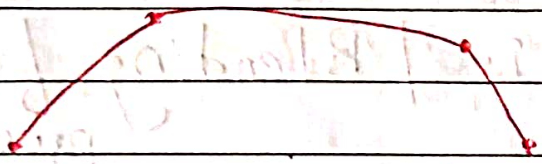


fig-: Interpolate

इसे ज्यादा use किया जाता है।

Approximate spline ⇒ ~~we~~ In this it is not necessary that ~~all sam~~ curve will pass through all (control) sample points. It is pass through near by.

n=3
control points = 4

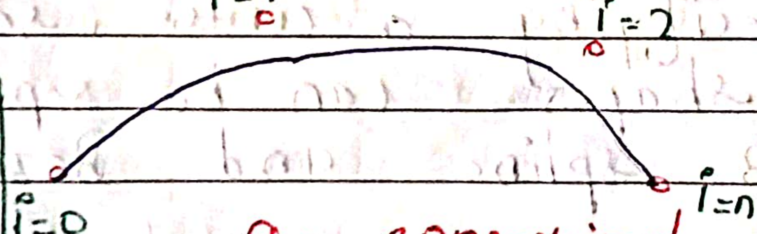
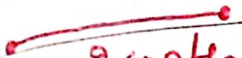


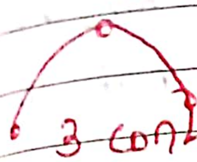
fig. approximate spline

Eqn ⇒ Cubic polynomial form में आती है

If degree is 3 the (control) sample points are 4.



2 control points $\Rightarrow$ Linear curve (means power 1)
x^1



$\Rightarrow$ Quadratic curve (degree 2)
3 control point x^2

4 control points $\Rightarrow$ cubic curve eqⁿ (degree 3)
x^3

$n \Rightarrow (n-1)$

Polynomial eqⁿ degree less than minus 1 control points.

$x^3 = 4$ C.P
$x^4 = 5$ CP
$x^5 = 6$ C.P.

Parametric and Geometric continuity $\Rightarrow$

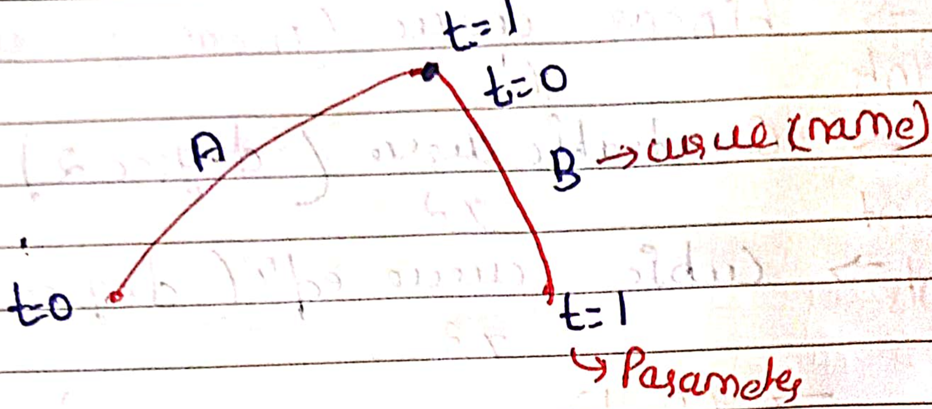
With the help of parametric and Geometric continuity we have to ~~find~~ check the smoothness of a curve after joining.

(i) Parametric continuity $\Rightarrow C^0$

(i) Zero order P.C $\Rightarrow$ In this both the curve are joined at one common point.

$$\boxed{A(t=1) = B(t=0)}$$

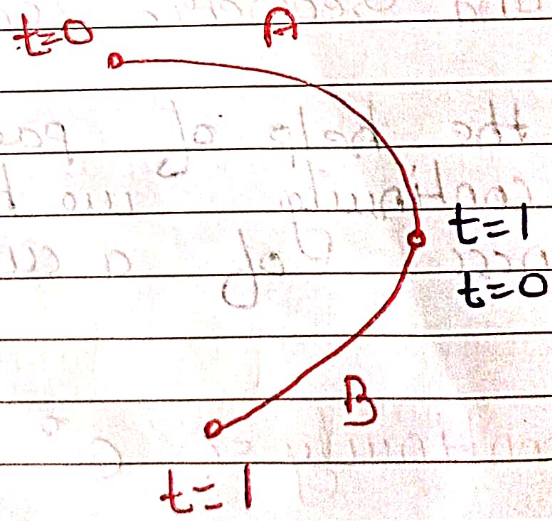
NOTES



इसमे ज्यादा smoothness nahi milegi

(ii) first order P.C ⇒ C^1

Derivative ← $\boxed{A'(t=1) = B'(t=0)}$

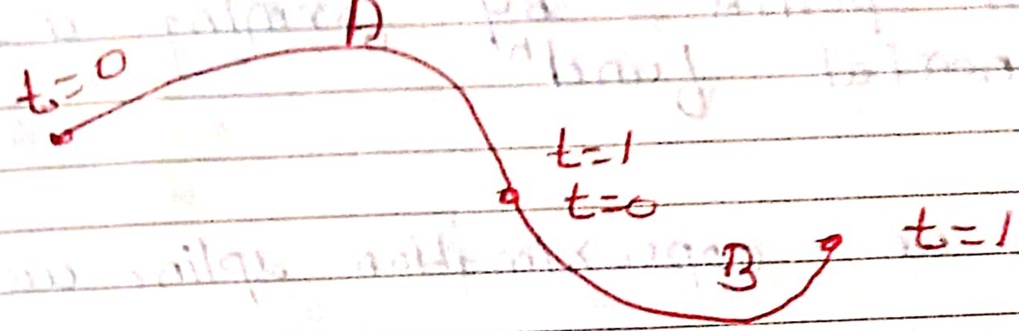


16 SUNDAY

NOTES

(iii) Second order P.C ⇒ C^2

$$\boxed{A''(t=1) = B''(t=0)}$$



Geometric continuity ⇒ G.C refers to the way that a curve on a surface looks (In this we have talk about only shape of the curve)

Bezier curve → It is a different way of specifying a curve with much more smoothness.

- A Bezier - curve is particularly a kind of spline generated from a set of control points by forming a set of polynomial funct^n.

- It is an approximation spline curve.

- ~~Be~~

- follow first control points & last control point.

- Bezier curve it will always inside convex hull of polygon boundaries

- Polynomial eq^n degree less than minus 1 control points

- Bezier Blending function are all positive & sum is always 1

NOTES

degree of polynomial

$$\boxed{\sum_{i=0}^{n} Bez_{n,i}(u) = 1}$$

Blending funct^n ⇒

$$Bez_{n,i}(u) = {}^nC_i \, u^i (1-u)^{n-i}$$

$[{}^nC_r \, p^r q^{n-r}]$

$$ {}^nC_i = \frac{\lfloor n}{\lfloor i \, \lfloor n-i} $$

For individual coordinates:

$$x(u) = \sum_{i=0}^{n} x_i \, Bez_{i,n}(u)$$

→ control points

$$y(u) = \sum_{i=0}^{n} y_i \, Bez_{i,n}(u)$$

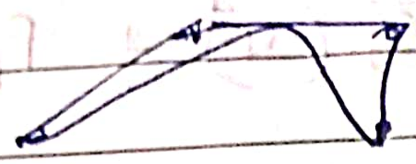
$$z(u) = \sum_{i=0}^{n} z_i \, Bez_{i,n}(u)$$

NOTES

→ derive Bezier eqⁿ

[Bezier polynomial]

→ Convex hull ⇒ Points boundary called convex hull. Our curve always lies inside this boundary.



Eqⁿ

$$Q(u) = \sum_{i=0}^{n} P_i * B_{i,n}(u) \quad -(i)$$

name of curve

$$X(u) = \sum_{i=0}^{n} x_i * B_{i,n}(u) \quad -(ii)$$

$$B_{i,n}(u) = {}^{n}C_{i} \cdot u^{i} (1-u)^{n-i}$$

where $${}^{n}C_{i} = \frac{n!}{i!\,(n-i)!}$$ ⇒ Binomial co-efficient

Bezier polynomial eq funct^n eq^n =>

$$\boxed{Q(u) = \sum_{i=0}^{n} P_i * Bez_{i,n}(u)}$$

P_i => Control points

$Bez_{i,n}$ => Blending funct^n or
Bezier funct^n or
Barstein funct^n

NOTES

Suppose,
We have 4 control points given $\Rightarrow$ degree 3 होगी.

Eq^n for 4 control points:-
$i = 0$, $n = 3$

$Q(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u)$

$+ P_2 B_{2,3}(u) + P_3 B_{3,3}(u)$

Binomial co-efficient.

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

$\uparrow$ $\uparrow$
i n

$$= \frac{1}{\cancel{0}1} \times u^0 (1-u)^3$$

$$= 1.1(1-u)^3$$

$=$

PAGE NO.:

$B_{1,3}(u) \Rightarrow P_1 \cdot 3 \cdot u(1-u)^2$

$B_{2,3}(u) \Rightarrow 3u^2(1-u)$

$B_{3,3}(u) \Rightarrow u^3$

$$P(u) \Rightarrow P_0(1-u)^3 + P_1 \cdot 3 \cdot u(1-u)^2 + P_2 \cdot 3u^2(1-u) + P_3 \cdot u^3$$

$$x(u) \Rightarrow x_0(1-u)^3 + x_1 \cdot 3 \cdot u(1-u)^2 + 3u^2(1-u)x_2 + u^3 x_3$$

$$y(u) \Rightarrow y_0(1-u)^3 + y_1 \cdot 3 \cdot u(1-u)^2 + 3u^2(1-u)y_2 + u^3 y_3$$

$$z(u) \Rightarrow z_0(1-u)^3 + z_1 \cdot 3 \cdot u(1-u)^2 + 3u^2(1-u)z_2 + u^3 z_3$$

PAGE NO:

properties

1. follow first control point $P(0)$ and last control point $P_n(1)$

$0 \leq u \leq 1$

2. Polygon boundaries by control points

3. Bezier curve it will always be inside convex hull of polygon boundaries

4. Polynomial eq^n degree less than minus 1 control points

4 control point $= x^3$

5 control points $\Rightarrow x^4$

Drawbacks -

1. Polynomial degree depends on no. of control point, so it is very complex.
2. Global control $\Rightarrow$ if we have to move control

curve
points so ~~its~~ shape will changed

n

B-spline curve →

Numericals: Er. sahil ka Gyan (Video)

Que: $B_0(1,0)$, $B_1(3,3)$, $B_2(6,3)$
$B_3(8,1)$
OR
$B_0[1\ 0]$, $B_1[3\ 3]$, $B_2[6,3]$ $B_3[8\ 1]$
Determine any 5 points lying on the curve. Also draw curve

Sol

$$Q(u) = B_0(1-u)^3 + B_1 3u(1-u)^2 + B_2 3u^2(1-u) + B_3 u^3$$

value of u lies in b/w $0 \leq u \leq 1$
we take randomly 5 values in b/w 0 to 1

Let $u = 0, 0.2, 0.5, 0.7, 1$

1) $u=0$

$$Q(0) = [1, 0](1-0)^3 + [3\ 3] \cdot 3(0)(1-0)^2 + [6\ 3] \cdot 3 \times (0)^2(1-0)^2 + [8\ 1] \times 0^3$$

$= [1\ 0]$

2) $u = 0.2$

$Q(0.2) = [2.304 \quad 1.448]$

3) $u = 0.5$

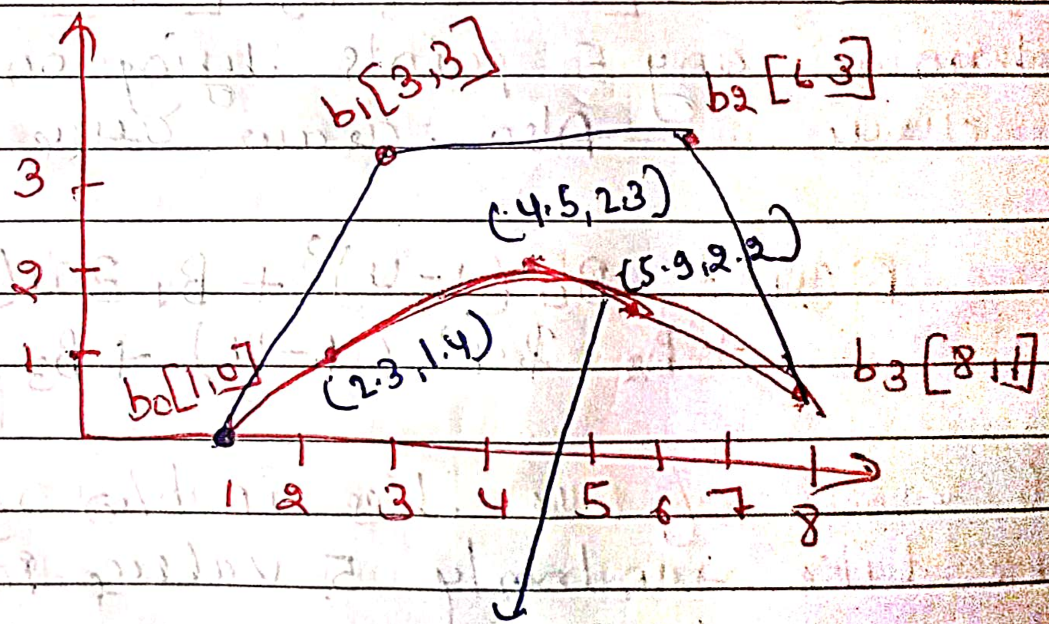
$Q(0.5) = [4.5 \quad 2.375]$

4) $u = 0.7$

$Q(0.7) = [5.984 \quad 2.233]$

5) $u = 1$

$Q(1) = [8 \quad 1]$



Curve

B-spline curve ⇒

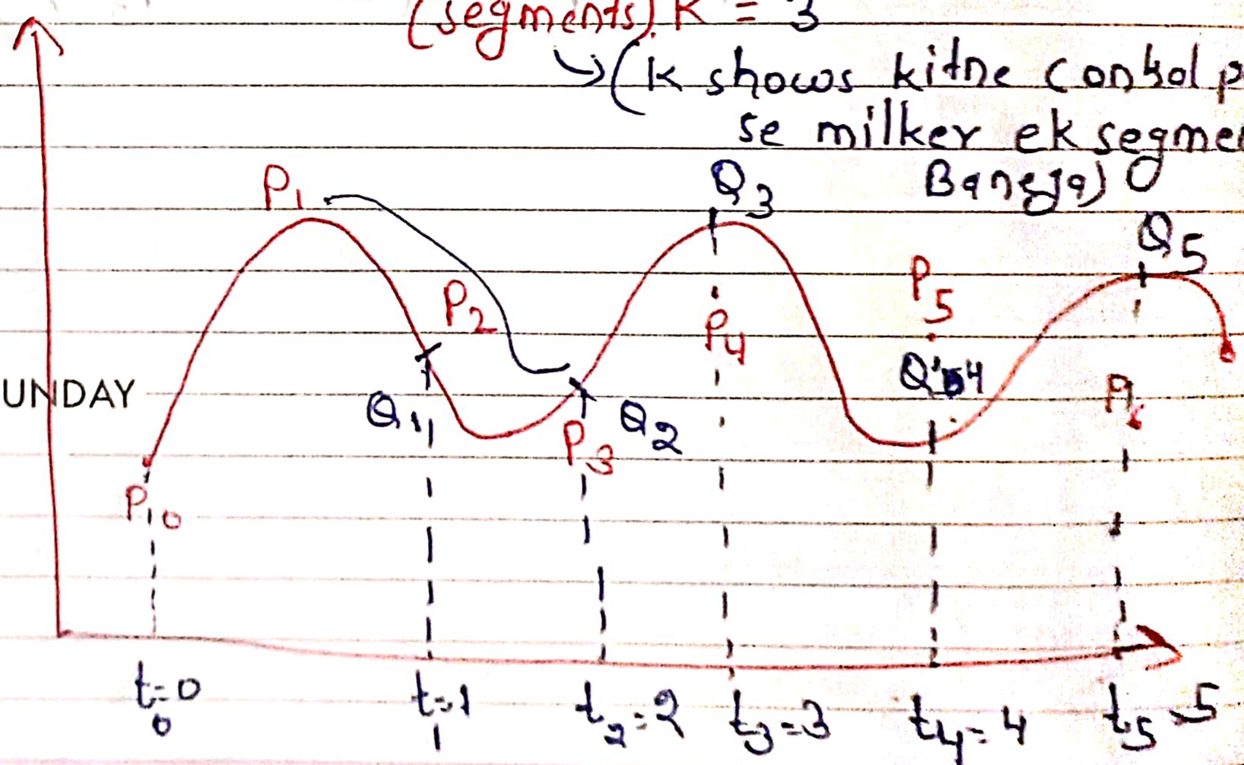
→ It is approximate spline curve.

→ It has "local control" over the control changing one control point affects only one part of the curve.

→ In B-spline curve the degree of curve depend on the no. of segments (K).

Ex

Control points = 7
degree n = 6
(segments) K = 3
↳ (K shows kitne control point se milker ek segment Banega)



NOTES

Segment	Control point	Parameter
Q_1	$P_0\ P_1\ P_2$	$t_0 = 0,\ t_1 = 1$
Q_2	$P_1\ P_2\ P_3$	$t_1 = 1,\ t_2 = 2$
Q_3	$P_2\ P_3\ P_4$	$t_2 = 2\ \ t_3 = 3$
Q_4	$P_3\ P_4\ P_5$	$t_3 = 3\ \ t_4 = 4$
Q_5	$P_4\ P_5\ P_6$	$t_4 = 4\ \ t_5 = 5$

$\Rightarrow n - K + 2$

$\Rightarrow 6 - 3 + 2$

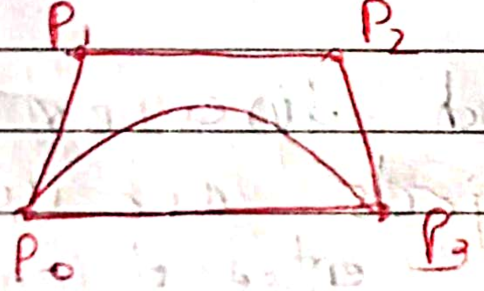
$\Rightarrow 6 - 1$

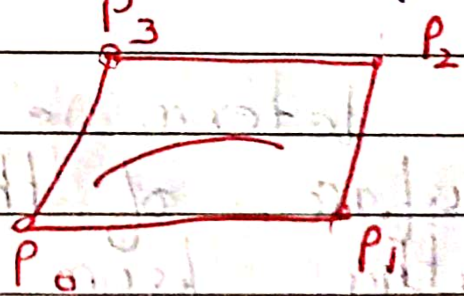
$\Rightarrow \underline{\underline{5}}$

NOTES

t_0 t_1 t_2 t_3 t_4 t_5

Bezier curve	B-Spline curve
1) It has global control over the control points. Changing one control points affects whole curve	It has local control over the control points. Changing one control points affects only one part of the curve
2) It is defined as curve lies in the convex hull of their control points	It is defined as curves lies within the convex hull of their control points





3) The curve touches at first & last control point	The curve here is not necessarily touches the first & last control points
4) The degree of curve depends on the no of control points degree $= n-1$	The degree of curve depends on the no. of segments degree $= k-1$

Hidden Surface Elimination Method ⇒

Hidden surface are object inside the viewing object that should not be seen.

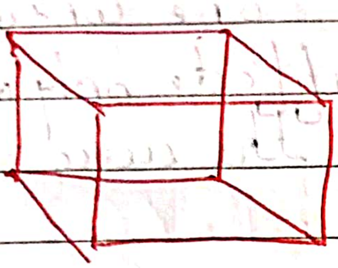


fig:- object with hidden lines

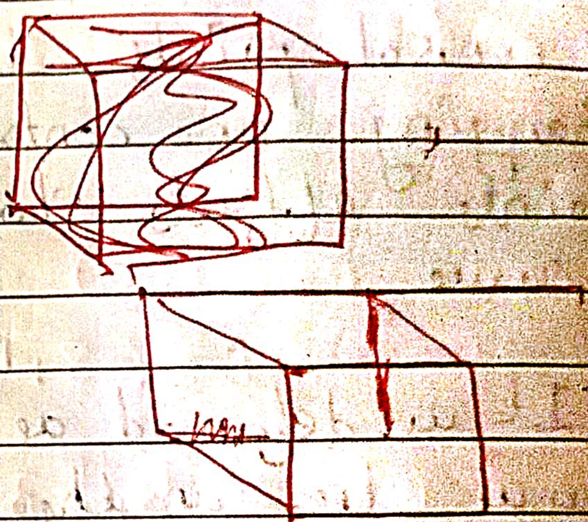


fig:- Object when hidden lines removed

To determine which lines or surface of the objects are visible either from the centre of projection or along the direct^n of projection so we can display only the visible lines or surface. This process is known as visible surface detection or hidden surface elimination.

OR

When we view a picture containing non-transparent objects and surfaces then we cannot see these objects from view which are behind from the objects closer to eye. We must remove these hidden surface to get realistic screen image.

The Identification and removal of these surface is called the hidden surface problem

Types of Hidden Surface Removal Algorithm

(i) object space method ⇒ compares object or part of object to determine visible surface (raster scan method)

(ii) Image Space method ⇒ visiblity is decided point by point : mean pixel by pixel

Hidden Surface Removal Algo ⇒

(i) Back face Removal Algo ⇒

→ It is object space type method.

→ Considering normal vector N to polygon surface with cartesian coordinates (A, B, C)

→ Suppose V is vector in viewing direction

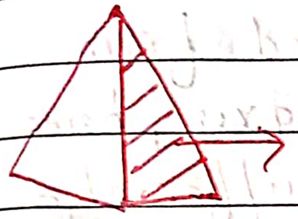
→ Polygon is back face

if $V.N > 0$

Polygon is front face

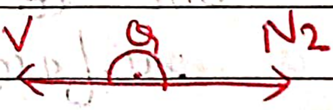
if $V.N < 0$

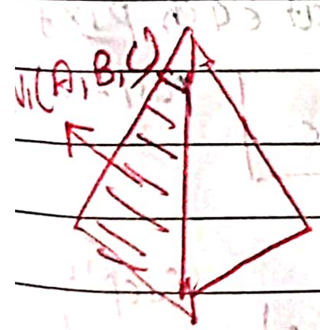
Ex



$$\boxed{V.N_2 = VN_2 \cos\theta}$$

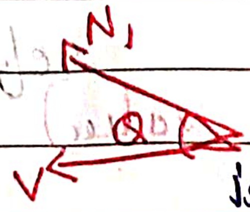
$V.N_2 < 0$





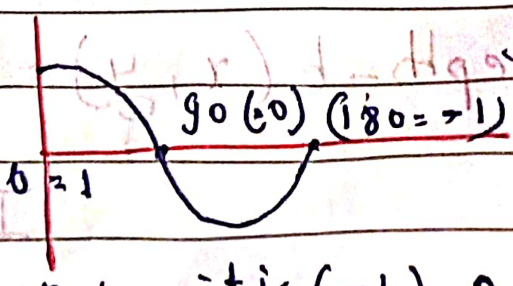
$$\boxed{V.N_1 = VN_1 \cos\theta}$$

$V.N_1 > 0$



(angle is less than 90°)

0 - 90 तक और below 90° cos θ will be greater than 0 its 1



and above 90° to 180° it is (-1) negative

Limitation ⇒ 'It work only with non overlapping (separate) objects

(ii) Depth Buffer or Z-buffer ⇒ Depth buffer is image space method compares surface depth at each pixel location on projection plane

→ Depth value is measured from view plane along z-axis, hence it is also called Z-buffer algo

→ Each surface is processed separately one point at time on that surface
OR
It compare surface depth at each pixel position

| Buffer ⇒ Temporary storage |

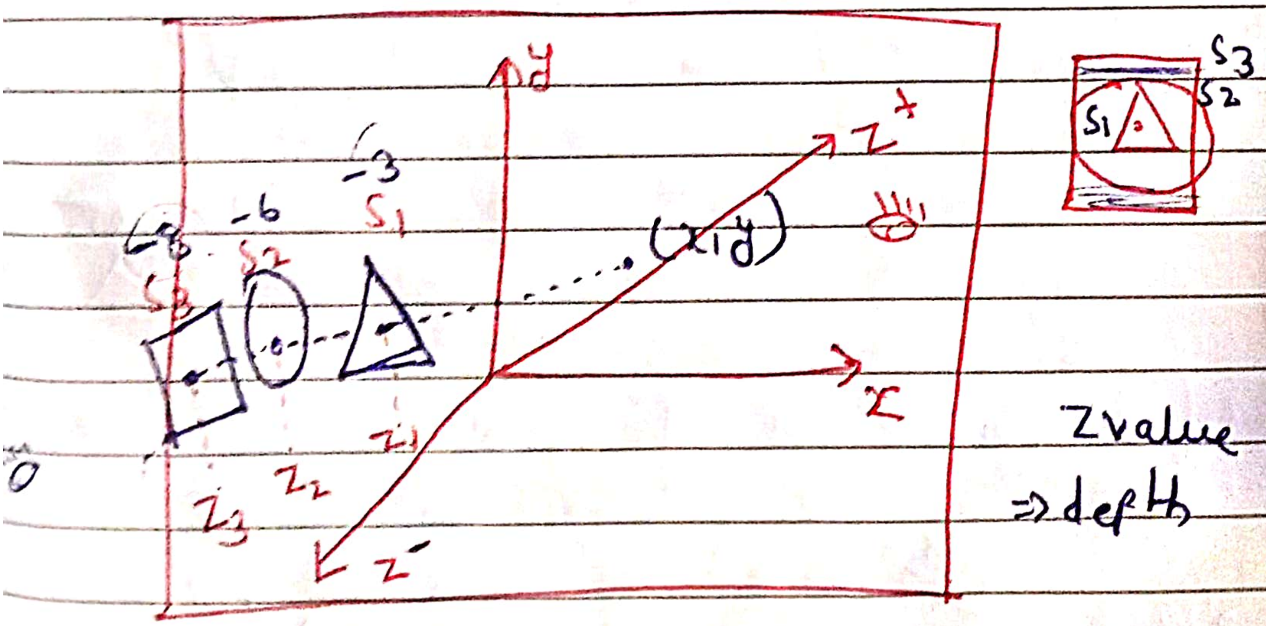
'Algo
- Initialize depth buffer (store z value (depth)) and refresh buffer (store intensity value) for all (x,y) values

depth_b (x,y) = 0, refresh_b = I_{back} → Background Intensity for each surface at each position compare depth value to determine visiblity

if.

$z > depth\_b(x,y)$ then

$depth\_b(x,y) = z$, $refresh\_b = I_{sur}(x,y)$



Zvalue ⇒ depth

$z_1 > 0$

$z_2 > z_1$

$z_3 > z_2$

PAGE NO.:

Limitation -:

(i) It can only process opaque surfaces

(ii) If there are transparent surfaces then z - buffer can't find correct intensity

(iii) Painters Algorithm ⇒