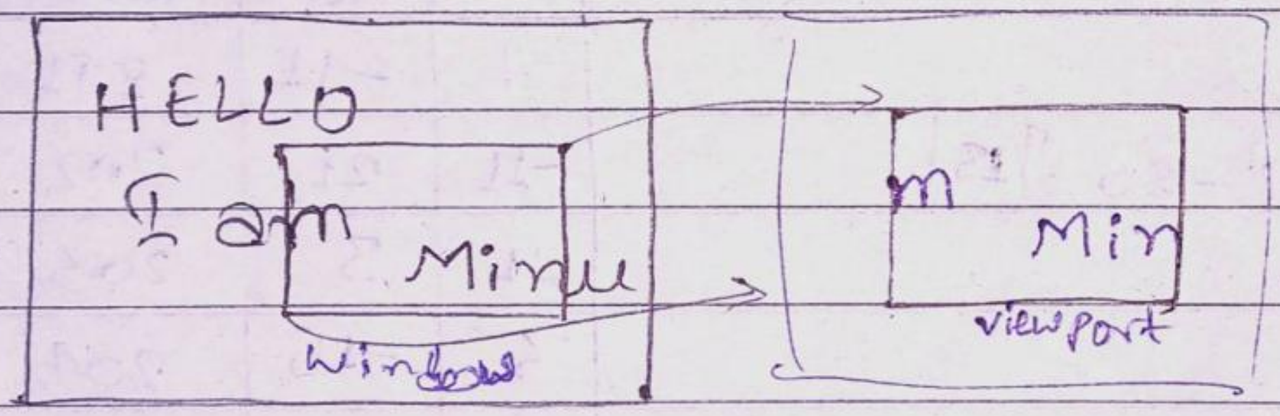
The Viewing Pipeling

PAGE NO.
DATE : / /20

- A World co-ordinate area selected for display is called "Window".
- An area on a display device to which a window is mapped a "viewport".
- Windows and viewports are rectangular in 'standard position'
- The mapping of a world coordinate scene to devices co-ordinate is refferred to as 'viewing transformation or Window-to-Viewport transformation or windows transformation.

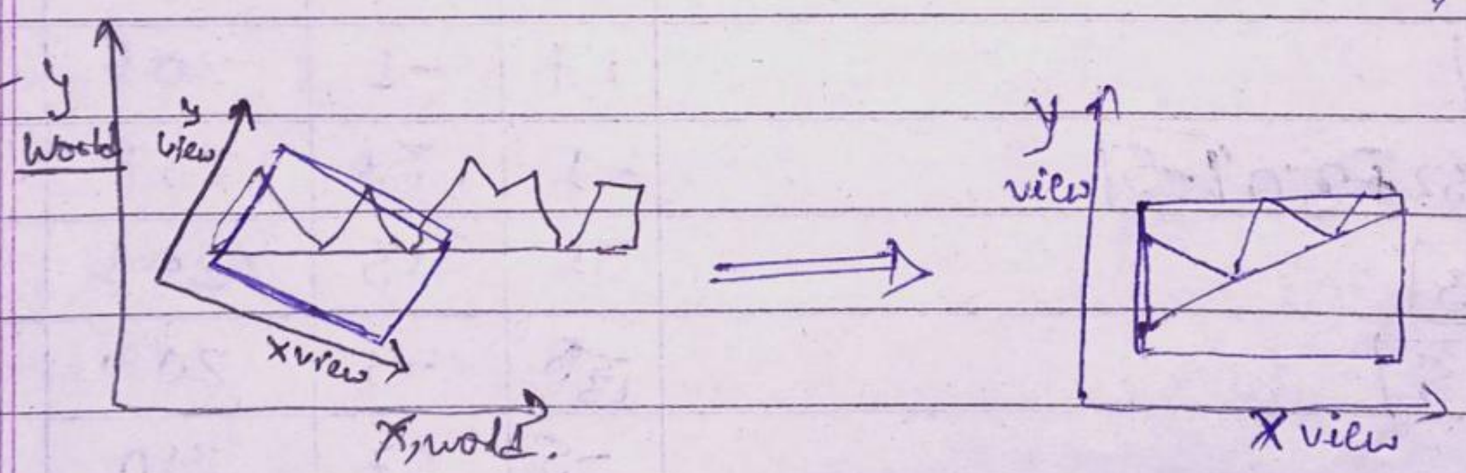
eg:-

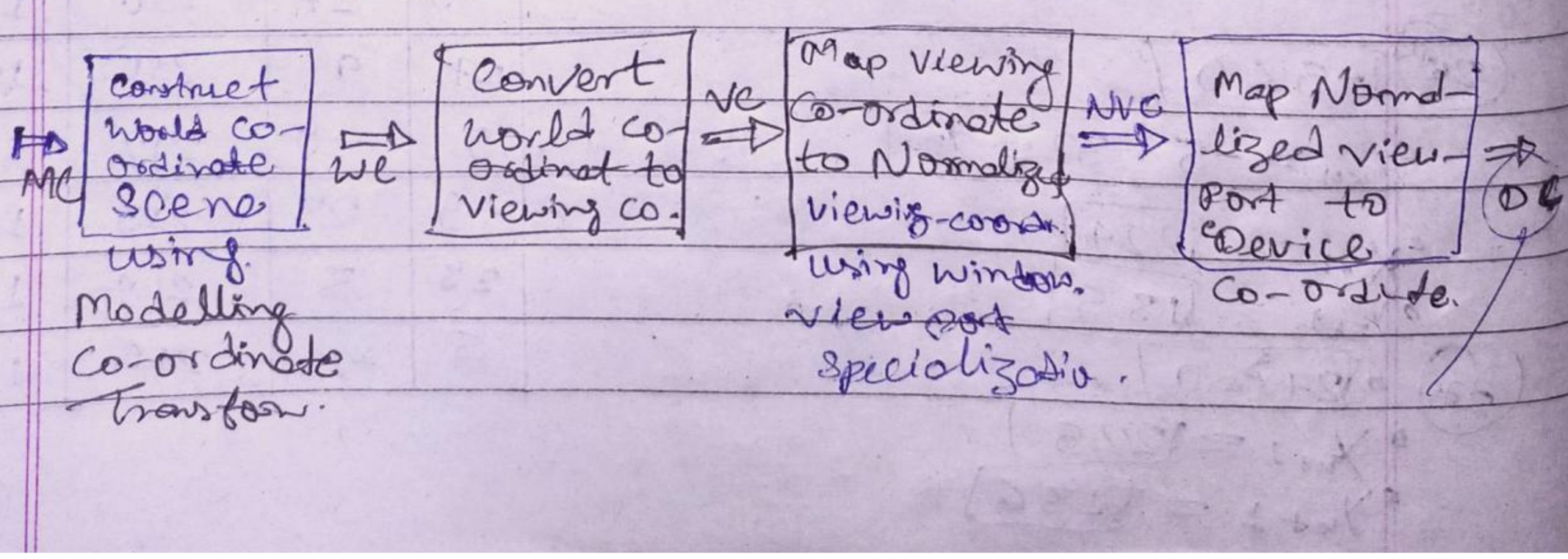


* window - to 'viewport' transformation

** $M = R \cdot T$

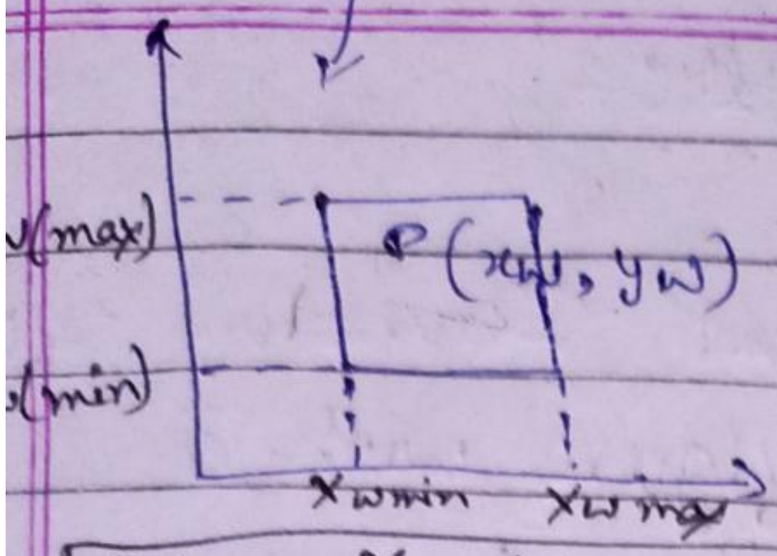
eg:-

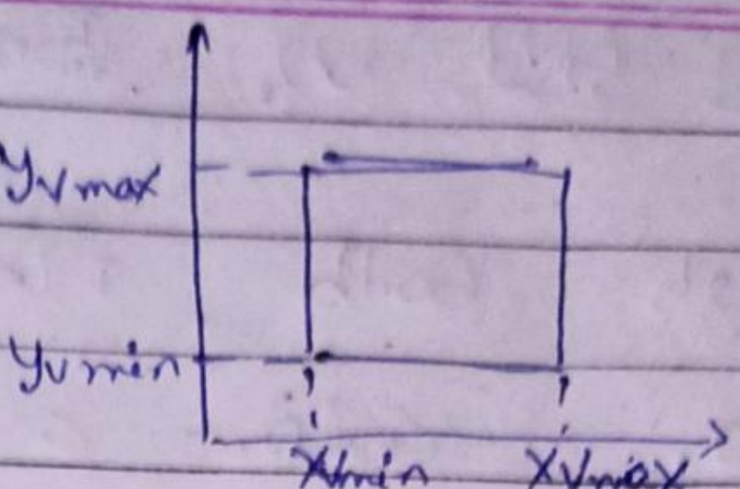




Window to viewport co-ordinate Transformation

PAGE NO.
DATE: / /20





$$\frac{x_v - x_{vmin}}{x_{vmax} - x_{vmin}} = \frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} \quad \text{---(1)}$$

* viewport co-ordinate

$$\frac{y_v - y_{vmin}}{y_{vmax} - y_{vmin}} = \frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}} \quad \text{---(ii)}$$

from eq (1)

$$x_v = \left\{ \frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} \right\} \times x_{vmax} - x_{vmin} + x_{vmin}$$

* $x_v = x_{vmin} + x_w - x_{wmin} \times \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}}$

$= [x_{vmin} + (x_w - x_{wmin})\, s_x]$ — Scaling factor.

where, $s_x = \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}}$

from eq (2)

$$y_v = [y_{vmin} + (y_w - y_{wmin})\, s_y]$$

where, $s_y = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}}$

Fill

PAGE NO.
DATE : / /20

Flood fill vs Boundary fill:-

Though both Flood fill & Boundary fill algo. color a given figure with a choosen color, they differ in one aspect

In flood :- All the connected pixels of a selected color get replaced by a fill color

In Boundary :- the program stops when a given color boundary is found.

Boundary Fill Algo

(1) It starts at a pixel inside the polygon. to be filled ~~color~~ & paints.

(2) It algo Works only if the color with which to be filled must be different from boundary color i.e $[f \neq b]$ ✓

(3) Also the boundary of polygon only contains a single color on all Boundary pixels.

(4) This Algo uses the recursive Methods.

4-way

8-way

Flood fill

PAGE NO.
DATE: / /20

It is an algo mainly used to determine a bounded area connected to a given node in a Multi-Dimensional array. It is a close resemblance to the bucket tool in plaint program.

(1) The selected Inside point is called seed point
(2) Suitable for filling multiple color Boundary's polygon.

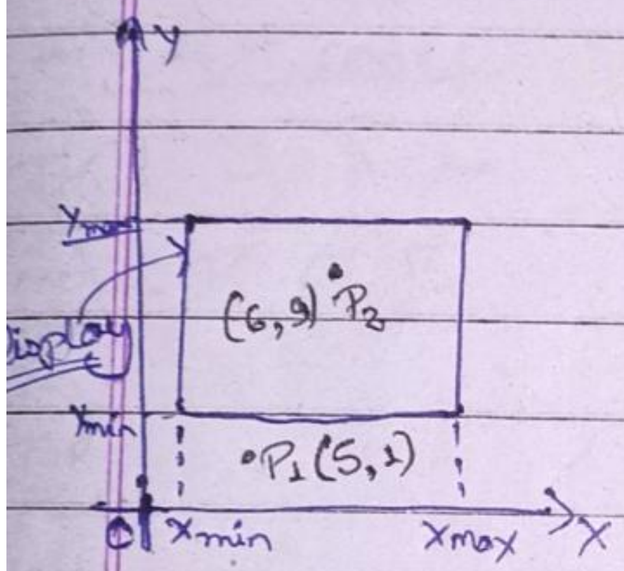
DisAdvantage :- (1) very slow Algo
(2) May be fail for large polygon.
(3) Intial pixel required more knowledge about surrounding pixel's.

C.G
=3

Point Clipping.

PAGE NO.
DATE: 01/08/2022

Point clipping:-



B → Below, R- Right

U → Upper

L → Left

Assume:- ① BL (2, 3)
Xmin Ymax

② UR (9, 11)
Xmax Ymax

* 4 - Condition Followed by :-

1) $X \geq X_{min}$
2) $X \leq X_{max}$
3) $Y \geq Y_{min}$
4) $Y \leq Y_{max}$.

} If all condition Satisfy then given point will be inside the Display.

* Check (P_1) is out or in-side

As we Assumed :- BL (2, 3)
UR (9, 11)

$P_1(5, 1)$

① $X \geq X_{min}$ → $5 \geq 2$ ✓
② $X \leq X_{max}$ → $5 \leq 9$ ✓
③ $Y \geq Y_{min}$ → $1 \geq 3$ ✗ ⟶ *It means P_1 is outside from Display
④ $Y \leq Y_{max}$ → $1 \leq 11$

$P_2(6, 9)$.

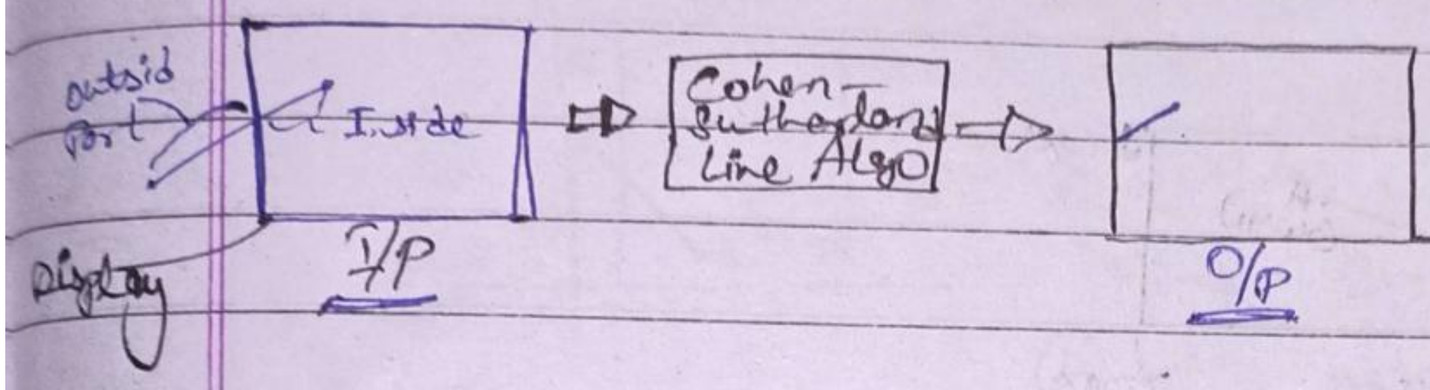
(6) $X \geq 2$ ✓
(6) ≤ 9 ✓
(9) ≥ 3 ✓
(9) ≤ 11 ✓

(P_2) is inside the Display

Line - clipping Algo

Cohen Sutherland Algo

PAGE NO.
DATE: 04/09/20



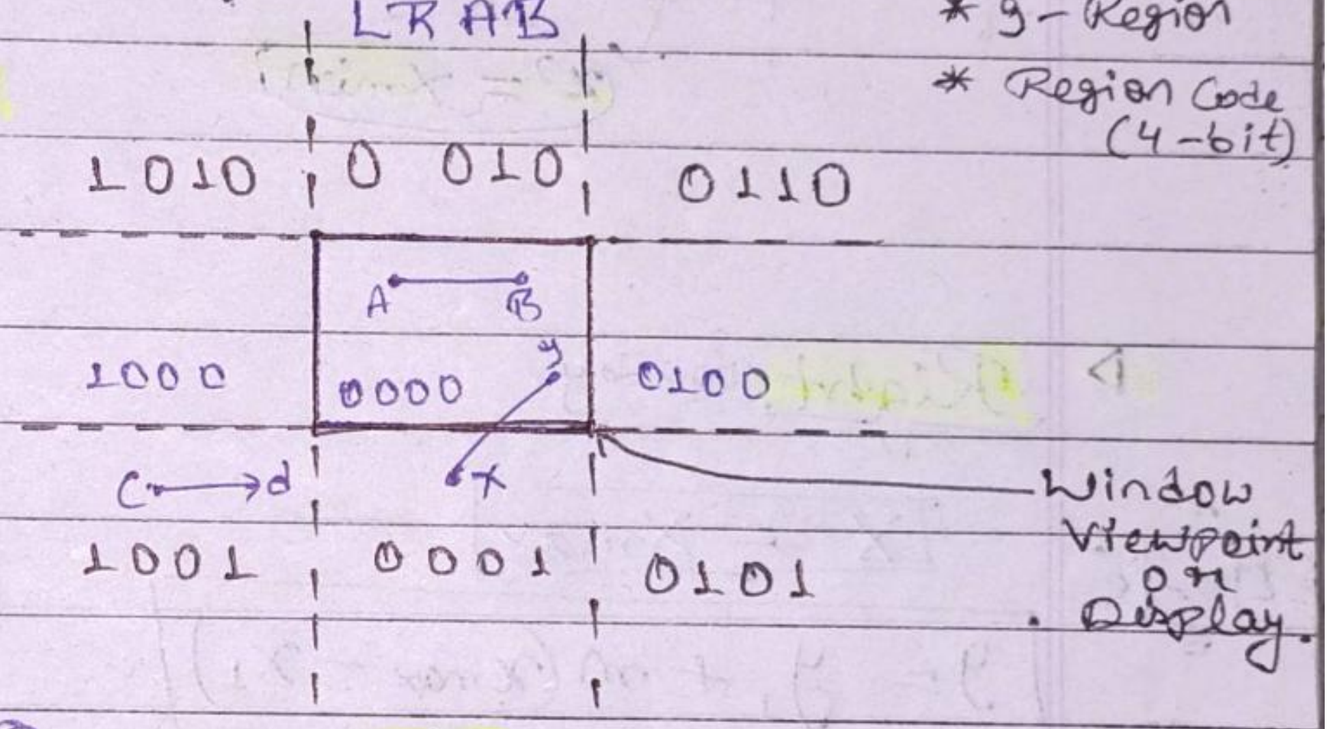
* 4-bit code

This may contain any one

T B R L
or
L R A B

T ⇒ Top
A ⇒ Abse
L ⇒ Left
R ⇒ Right
B ⇒ Below

* Here USED.



* 9-Region
* Region Code (4-bit)

(1) Line Inside :- How to Check.

↳ Check the object (placed) Region code (4-bit).

Ex:- AB have (0000) Region code

(2) Line is comple outside?

↳ take the all end point (corner) & Perform 'AND' operation, if Answer is Not Zero (0000). then outside

Ex:- C → 1001.
d → 1001 AND
1001
1001

(3) Partially Inside of the window

If 'And' operation of two points' region: And output is zero (equal to 0000) then Object is Partially Inside the window, Not completely

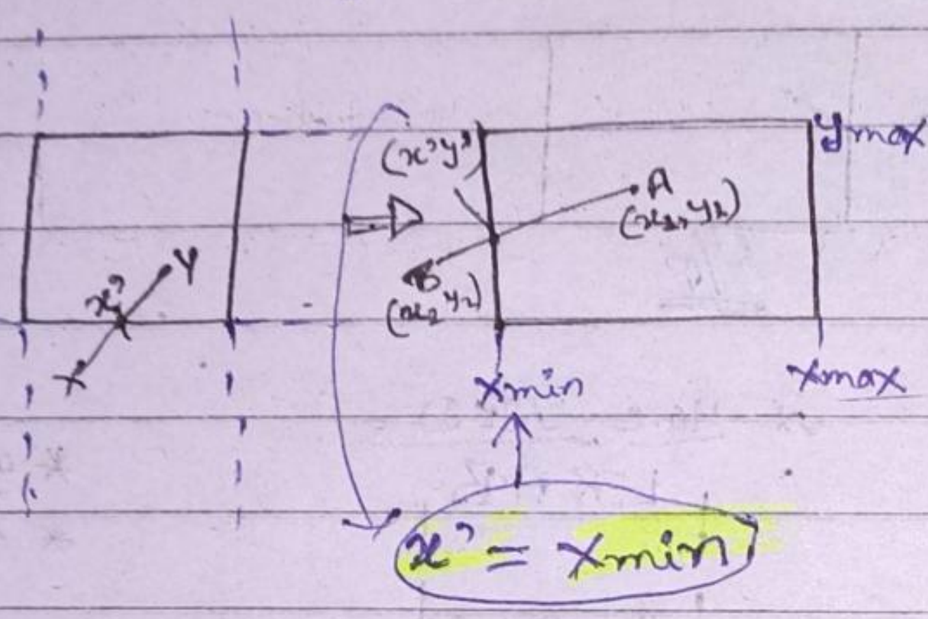
Ex:- XY : X → 0001
0000
AND 0000

Look Region

Cohen Sutherland Clipping Algo

PAGE NO.
DATE : / /20

* Partially object can be be display in window But only Partial object.



$x' = x_{min}$

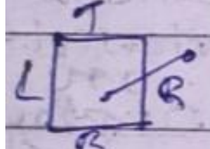
▷ Left Boundary $[x = x_{min}]$

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

$= \frac{y - y_1}{x - x_1} = \boxed{\frac{y - y_1}{x_{min} - x_1}}$

$[y = y_1 + m(x_{min} - x_1)]$

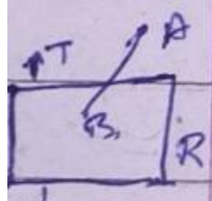
▷ Right Boundary



$[x = x_{max}]$

$[y = y_1 + m(x_{max} - x_1)]$

▷ Top/Above :-



$[y = y_{max}]$

$m = \frac{y - y_1}{x - x_1} = \frac{y_{max} - y_1}{x - x_1}$

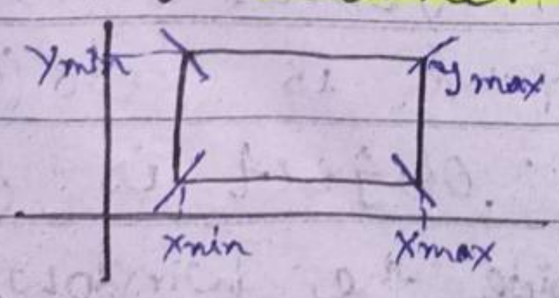
$\left[x = x_1 + \frac{(y_{max} - y_1)}{m}\right]$

▷ Bottom :-

$[y = y_{min}]$

$\left[x = x_1 + \frac{(y_{min} - y_1)}{m}\right]$

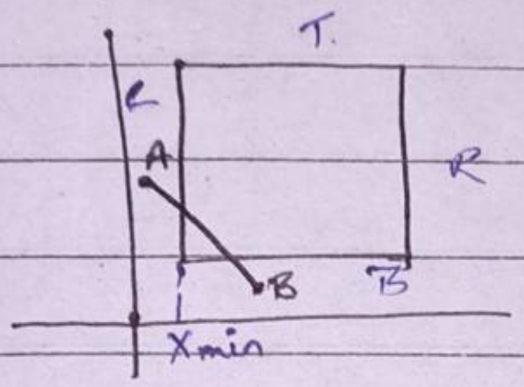
Note :- If Corner Intersected Line :-



* Here just same as Xmin or Ymin or Ymax or Xmax etc.

PAGE NO.
DATE: / /20

▷ New case :- if line is formed like ~~2 to~~ below :-

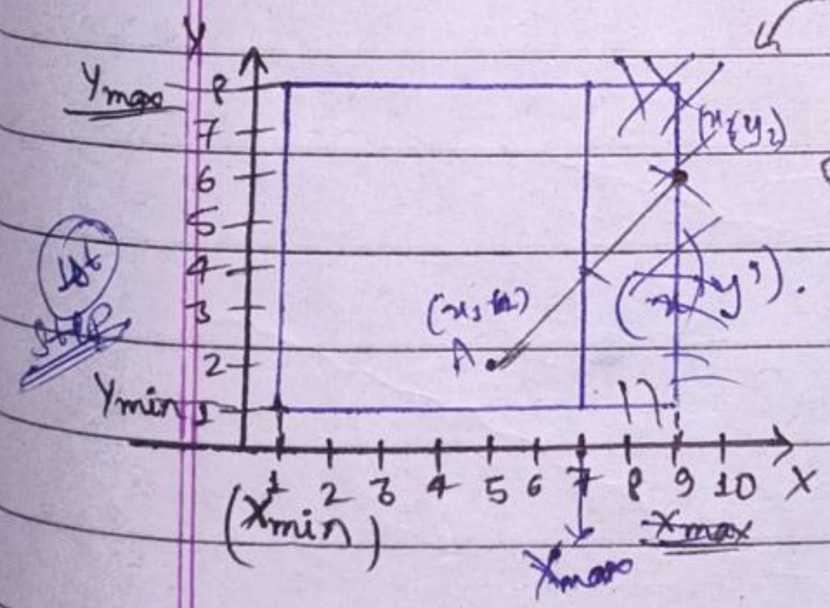


* In this case we have to Find 2- to Intersection ~~Point to Find~~
(1) with below (B')
(2) with left (A')

Numerical on Cohen S. Algo (Line clip

Q).

BL : (1, 1) → A(5, 2), B(9, 6).
UR : (7, 8)



TBRL
0 0 0 0
0 0 1 0
AND 0 0 0 0

Boundary is Right so,

▷ We need slope

$(m) = \frac{6-2}{9-5} = \frac{4}{4} = 1$

* $X = X_{main} = 7$

$y = 2 + 1(7 - 5)$

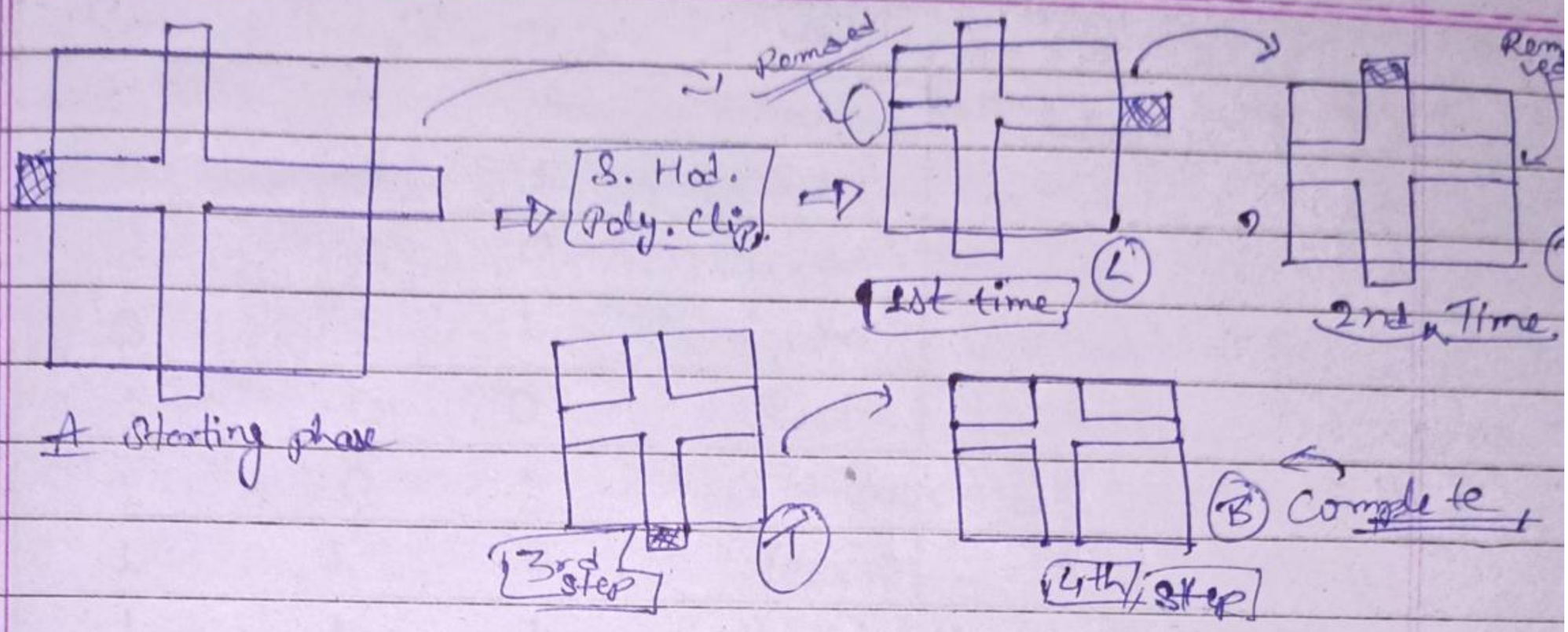
$y = 2 + 1(2) = 4$ Ans

$(x'y') = (7, 4)$ Ans

Ch
M-3

Sutherland Hodgeman Polygon clipping Algo

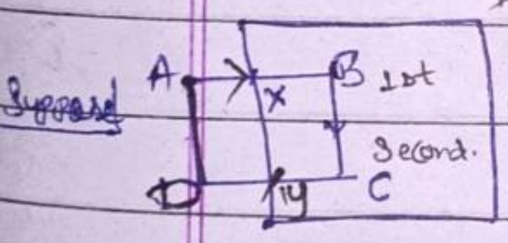
PAGE NO.
DATE: 09/08/2022



This Algo used for clipping polygons. It works by extracting each line of the convex clip polygon in turn & selecting or vertices from the Subject polygon that are on the visible side.

4 - Rules :-

(1) Left :- (a) In this case the Direction of edge is Outside → Inside (A→B)



In this, we have to Conside (X, B) (Save) only.

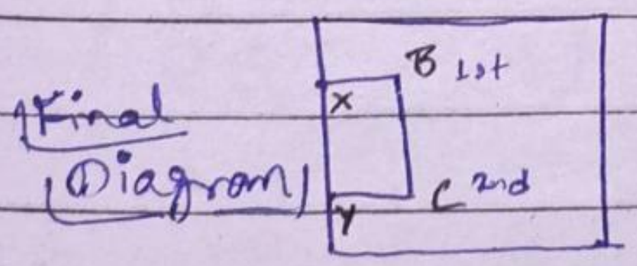
(b) Second Direction is Inside → Inside (B→C)

Here, We have to consider Second point (C) (Save)

(c) 3rd Direction is Inside to Outside

So, consider the Intersection point (Y) → (Save)

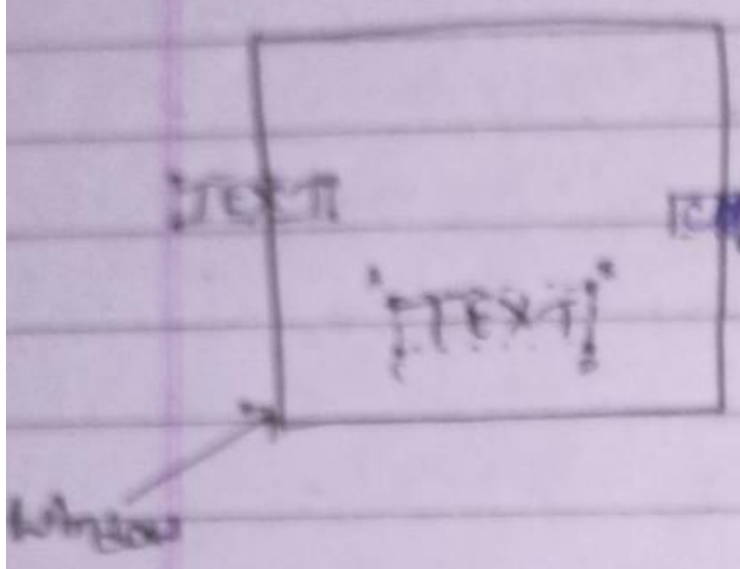
(d) Out → Out, In this Direction just Ignore. Don't Save



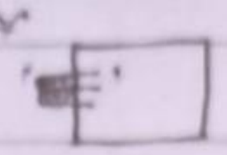
TEXT CLIPPING

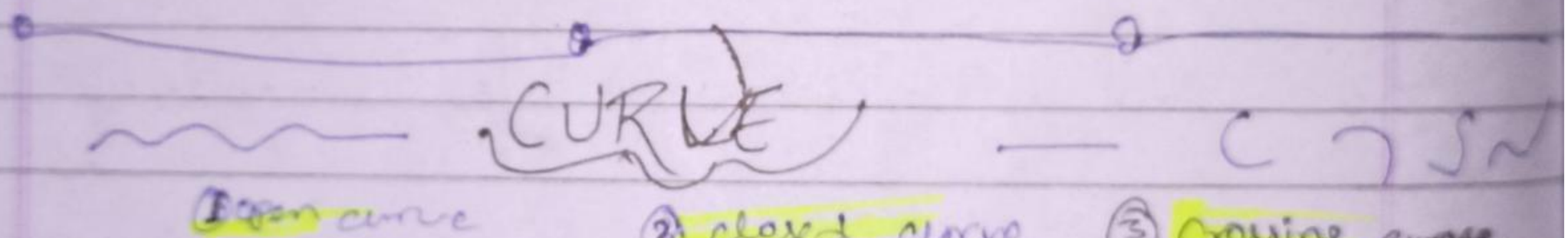
PAGE NO.
DATE: 25/ 08/ 2022

* It used to select or reject text from the window



① All string OR Null string
② All char OR none string
③ Individual char. clipping





① Open curve
↳ ∩ ∪ ⊂ ⊃ ∩

② closed curve
↳ ○ ◎ {}

③ Crossing curve
↳ ∞ &

etc

Curve Representation,

① Explicit
↳ $y = f(x)$
[dependent] In-Dependent

② Implicit :-
↳ $f(x, y) = 0$
$x^2 + y^2 - r^2 = 0$

③ Parametric :-
↳ $(x, y) = f(v)$

Hidden Surface Detection

Buffer Algorithm

End

** Depth

PAGE NO.
DATE : / /20

It also known as Depth Buffer Algo

* Simplest Image space Algorithm. * Record Intensity

* It keeps a record of the Depth of an object with in the pixel that lies closest to the observer

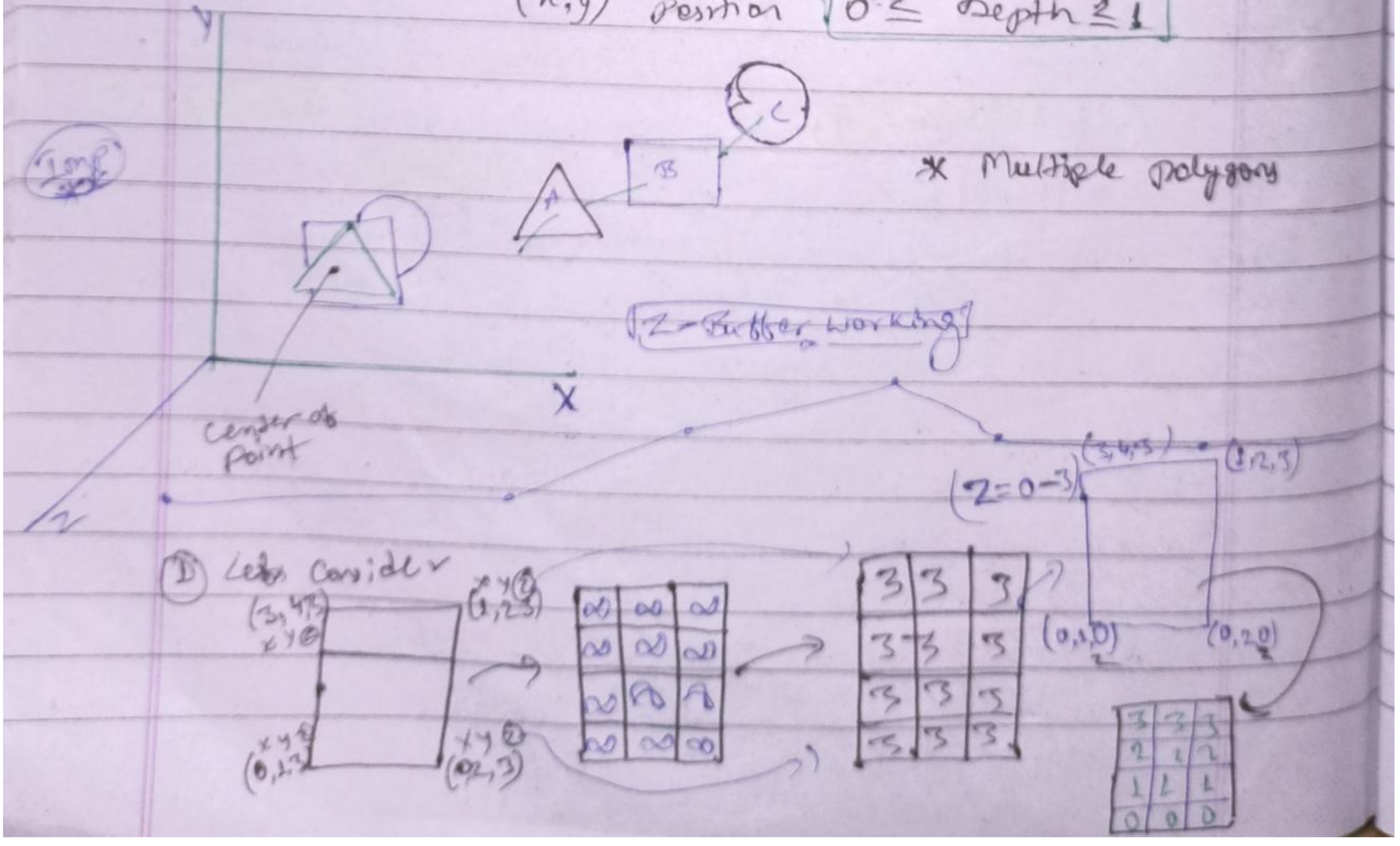
* 2-Array required → Intensity
→ Depth.

* Efficiently used in Polygon.

▷ To override the closer polygons from the far ones, two buffer:—

① Frame buffer :— used to store the Intensity value or color at each position (x,y)

② Depth buffer :- Used to store depth value for (x,y) position [0 ≤ depth ≤ 1]



Algorithm Z-Buffer

PAGE NO.
DATE: / /20

Algo

Step (1) Set the buffer values :-

1. Depth value x, y = 0';
2. Frame buffer x, y = Background color;

Step (2) Process each polygon Oneatatime :-

for each projected x, y pixel position of a polygon, calculate Depth Z.

(3) if (Z > DepthBuffer x, y)
Compute Surface color,
Set depthbuffer x, y = Z,
frame buffer x, y = Surface color x

END;

▷ ADVANTAGE :- (1) easy to Implement
(2) It reduces the Speed problem if implemented in hardware.
- It processes one object at a time.

▷ DisAdvantage :-
- It requires Large memory.
- It is time consuming process.

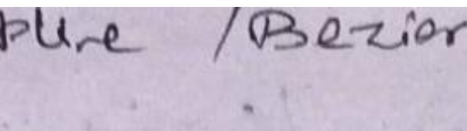
***B - Spline / Bezier Curve * DVST

**RGB Chg

Parametric Curve

PAGE NO.
DATE: / /20

- Piece wise:-
  - ↳ Linear :-
  - ↳ Polynomial :-

↳ Linear :- 

$$y = mx + C$$

↳ Polynomial :-

* $x^{n\text{-ton}}$ → x^3

- Cubic → Why?

  Degree (3)

* Parametric Cubic Curve in 3-D

$$x(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$
$$y(t) = b_3t^3 + b_2t^2 + b_1t + b_0$$
$$z(t) = c_3t^3 + c_2t^2 + c_1t + c_0$$

* Matrix :- Representation

→ $t \cdot [0-1]$

$$X(t) = T \cdot A$$

$$y(t) = T \cdot B$$

$$z(t) = T \cdot C$$

⇒ t^3 ~~t3~~ $\begin{bmatrix} t^3 & t^2 & t^1 & 1 \end{bmatrix} * \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} =$ Ans

⇒ Same with (b)

⇒ Same with (c).

Ch 1

Parametric Continuity

PAGE NO.
DATE: 05/09/2022

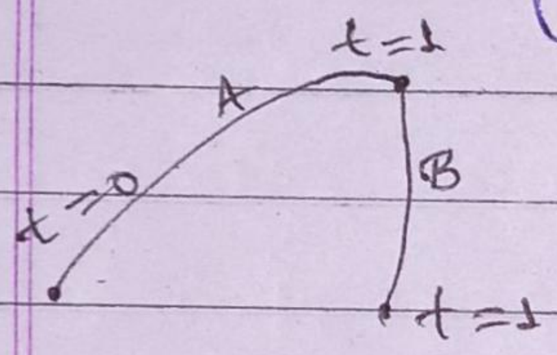
* Smoothness * continuity.

order Partial continuity.

(1) Zero OPC :-

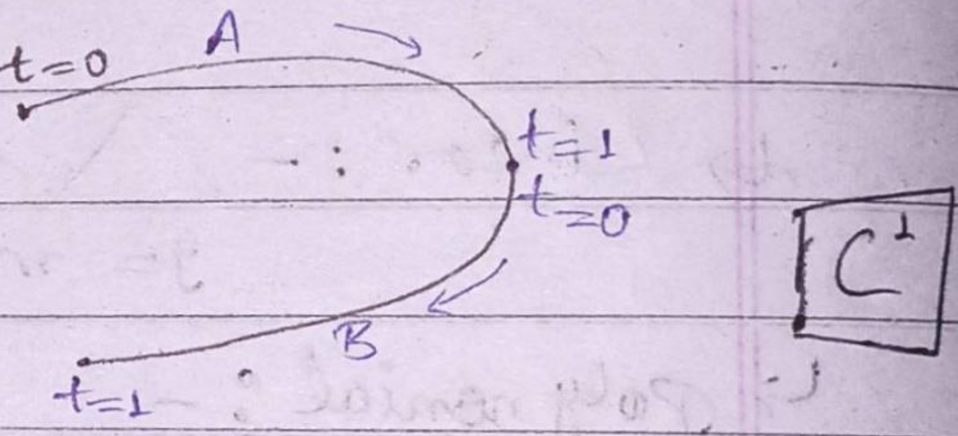
$$A(t=1) = B(t=0)$$

⇒ Represented by C^0



(2) First OPC :- 1st order

$$A'(t=1) = B'(t=0)$$



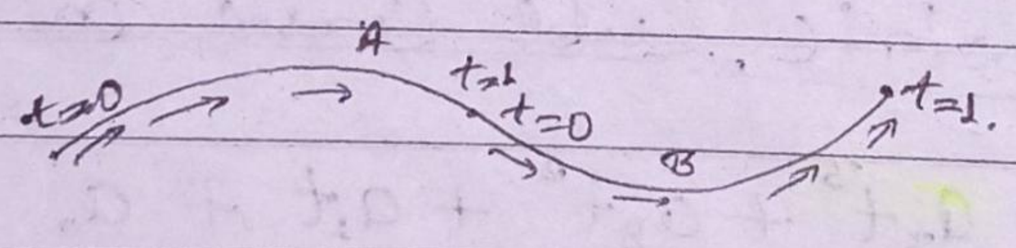
C^1

(3) Second OPC

2nd order derivative

$$A''(t=1) = B''(t=0)$$

C^2



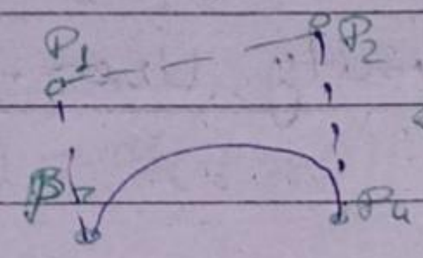
Parametric

Bezier Curve

(Important)

PAGE NO.
DATE : / /20

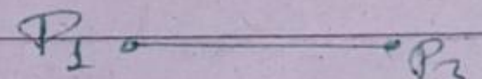
* It defined by "control Point"

Ex: 

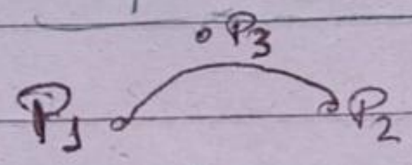
⇓

{P1, P2, P3, P4} are controlpoints.

⇒ 2-point Curve



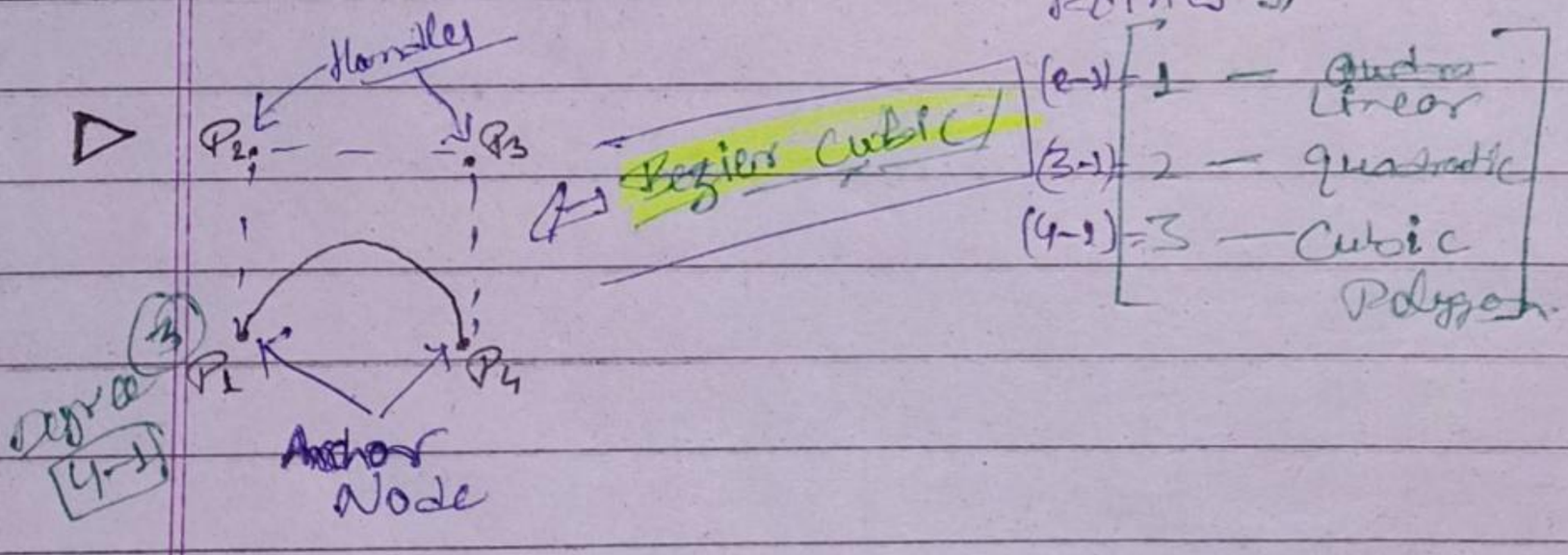
* 3 pt curve



x n-1 (2) Quadratic.

* 4-Point Curve
x Also called Cubic
n-1 = (3)

* Properties:- (1) Points are not always on curve.
(Piyoosh) (2) It always pass through first last control Point
(3) They are connected in the convex hull of their defining control ~~post~~ points.
(4) The Degree of polynomial defining the curve segment is one less then the no. of defining Polygon Points.



(5) The shape of ~~defi~~ defining Polygon is usually followed by Bezier Curve.

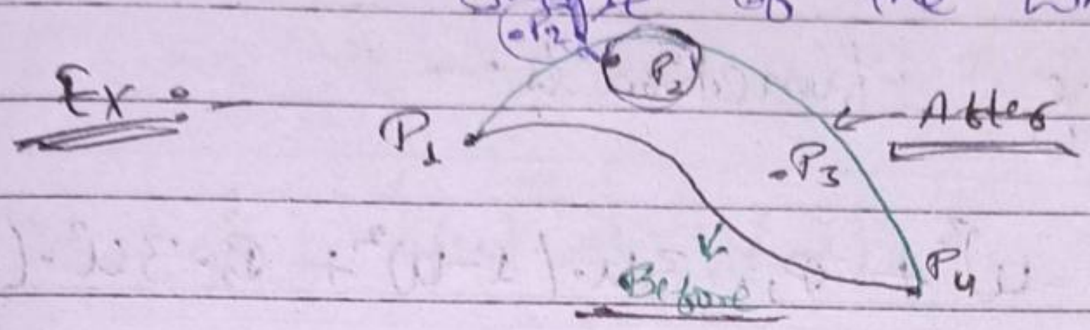
(6) Bezier Curve are tangent to their first & last Edges of control polyline.

Bezier Curve

PAGE NO.
DATE: / /20

(7) Bezier curves: Exhibit global control point means moving a control point Alters the shape of the whole curve.

Ex:-



▷ Derivation of Bezier Curve:-

(1) Blending/ basis function

Blending fun

$$P(u) = \sum_{i=0}^{n} P_i B_{i,n}(u) \quad ; \quad 0 \le u \le 1$$

n+1 ⇒ control points

eg: n = 3 then 4 control Points

Where,

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

* Cubic Bezier:-

$$P(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u) + P_2 B_{2,3}(u) + P_3 B_{3,3}(u).$$

$$\Rightarrow B_{0,3}(u) = C(3,0) \cdot u^0 \cdot (1-u)^{3-0} \qquad u = 0$$

$$= \frac{3!}{0!\,3!} (1-u)^3 \Rightarrow (1-u)^3 \quad \text{---(i)}$$

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

$$= 3u \cdot (1-u)^2 \quad \text{---(ii)}$$

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

$$= 3u^2 \cdot (1-u) \quad \text{---(iii)}$$

Bezier

PAGE NO.
DATE : / /20

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

$= u^3$ ——— (iv)

So, Final Blending function is :-

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

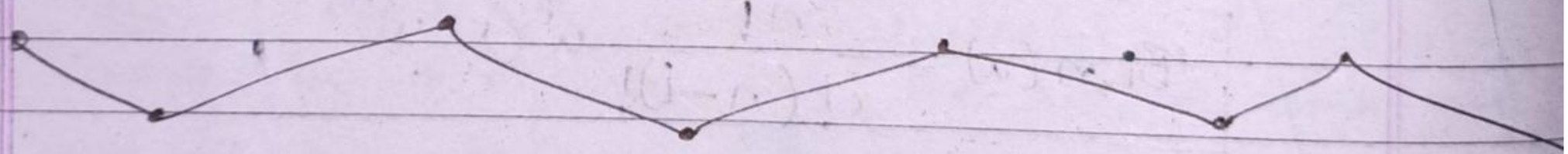
TWO-D [Do, Find x, y, z]

$P(u_x) = [P_0 x_0 (1-u)^3 + P x_1 \cdot 3u \cdot (1-u)^2 + P x_2 \cdot 3u^2 \cdot (1-u) + P x_3 \cdot u^3]$

$P(u_y) = [P y_0 (1-u)^3 + P y_1 \cdot 3u(1-u)^2 + P y_2 \cdot 3u^2 \cdot (1-u) + P y_3$

$P(u_z) = [P_{z_0}(1-u)^3 + P_{z_1} \cdot 3u \cdot (1-u)^2 + P_{z_2} \cdot 3u^2 \cdot (1-u) + P_{z_3} \cdot u^3$

3-D



Example:- Design a Bezier curve controlled by 4-points
A(1,1) B(2,3) C(4,3) d(6,4)

Soln

Given, Control point is $\Rightarrow 4$

*Degree is $n-1 = 3$

A (1,1) P_0
B (2,3) P_1
C (4,3) P_2
d (6,4) P_3

*Equation of B. Curve

$$P(u) = \sum_{i=0}^{n} P_i \cdot B_{i,n}(u)$$

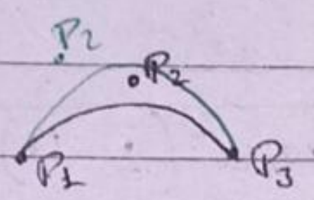
$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} \cdot u^i \cdot (1-u)^{n-i}$$

B-Spline Curve → Limited - flexibility

PAGE NO.
DATE : / /20

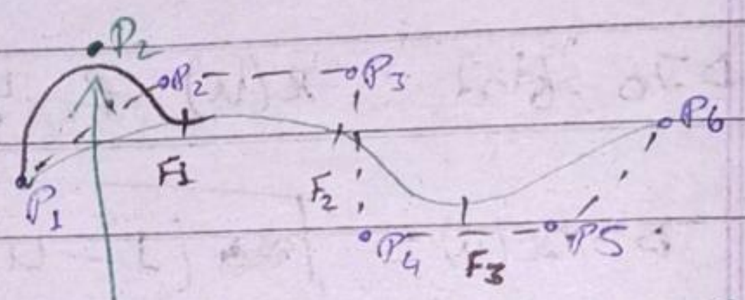
→ Bezier Curve has Global control → B-spline has Local control
→ B. Curve completely depends on control Point → Doesn't depends upon No. of Control point
→ Its Parametric Curve used in related field → It Depends upon order of Polynomial

Ex:-



Here, $= \sum_{i=0}^{n} P_i B_{i,n}(u)$

Ex:-



Here, only changed in B-spline Because its Local P.

Here, $\sum_{i=1}^{n+1} N_{i,k}(u)$

Properties :-

① The sum of B-Spline basis function at any Parameter 'u' is equal to '1'

i.e

$$\left[\sum_{i=1}^{n+1} N_{i,k}(u) = 1\right]$$

[n+1 = No. of control point]
[k = sides / order of B-spline Curve]

② The basis function is the +ve zero for all parameter value is i.e $N_{i,k}(u) \geq 0$. Except for $(k=1)$ Each basis function has one Maximum value.

③ The Max side order of the Curve is equal to the No. of vertices of defining polygon

④ The Degree of B-Spline polynomial is independent on the No. of vertices of Defining polygon

PAGE NO.
DATE: / /20

(5) B-Spline allow the control (i.e local control) over the curve surface.

(6) The curve lies with the convex hull of its defining polygon.

(7) The curve generally follow the shape of defining polygon

The B-Spline blending function, & Representation as:-

$$P(u) = \sum_{i=1}^{n+1} P_i N_{i,k}(u)$$

Where,

$$[u_{min} \leq u \leq u_{max}]$$

$$[2 \leq k \leq n+1]$$

* $$N_{i,k}(u) = \frac{(u - x_i) \cdot N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) \cdot N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

$$N_{i,k}(u) = \begin{cases} 1, & x_i \leq u \leq x_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$x_i = 0$ if $(i < k)$
$x_i = i - k + 1$ if $(k \leq i \leq n)$
$x_i = n - k + 2$ if $(i > n)$

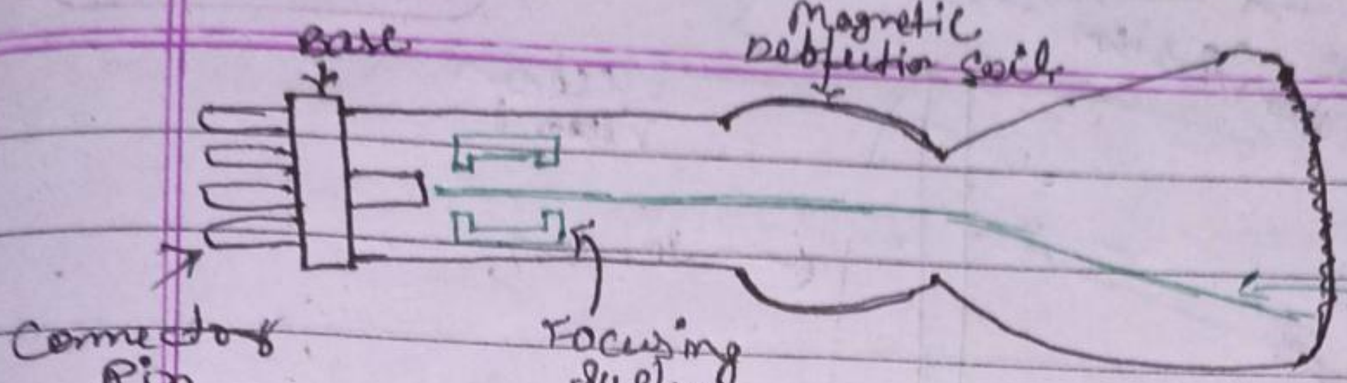
Note: When we designing the B-Spline curve we have to Evolute Knot, vector based on the no. of control points & order of curve.

⇒ Knot vector can be evaluted by using.

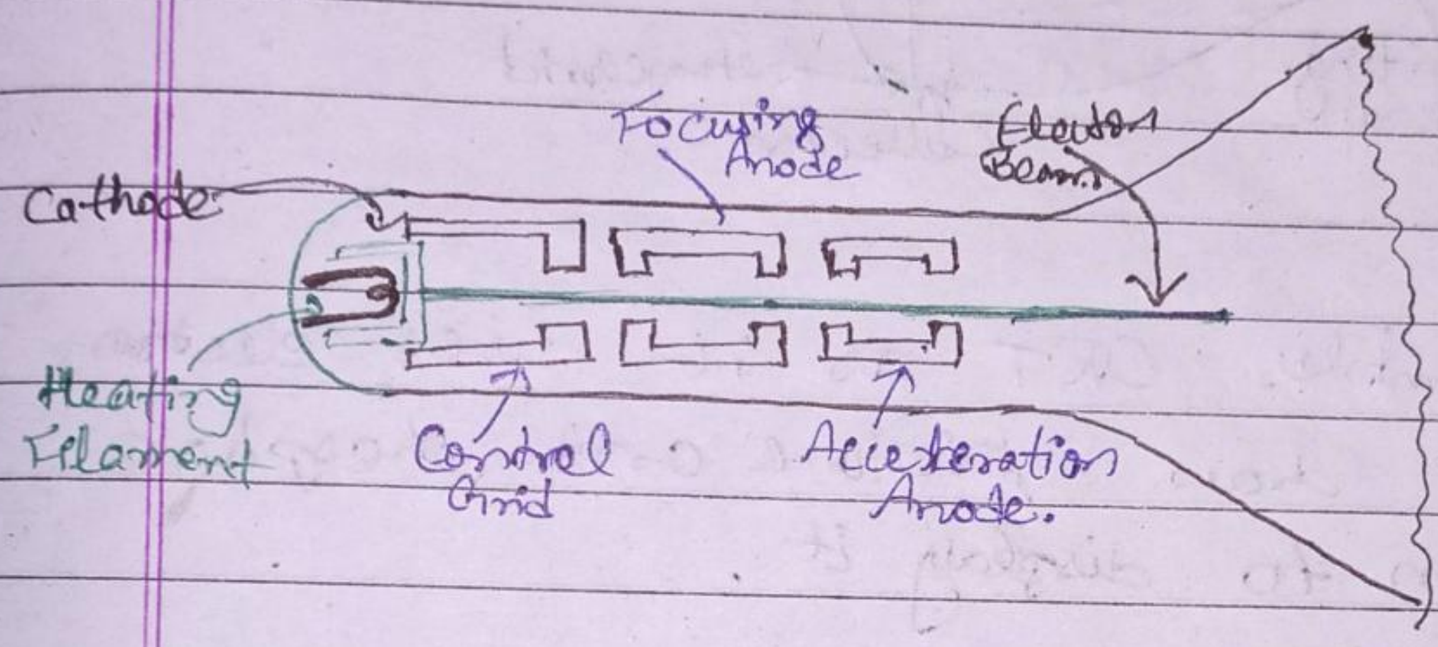
1. Uniform
2. Non-Uniform
3. open - Uniform.

CRT → Refresh Needed

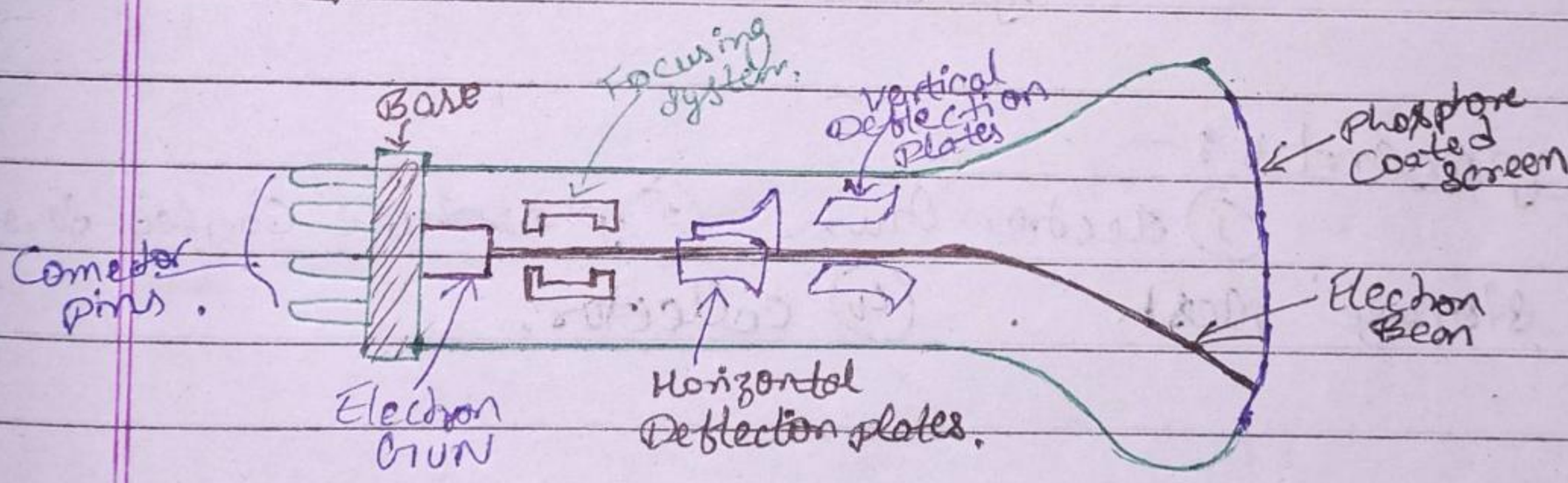
PAGE NO.
DATE : / /20



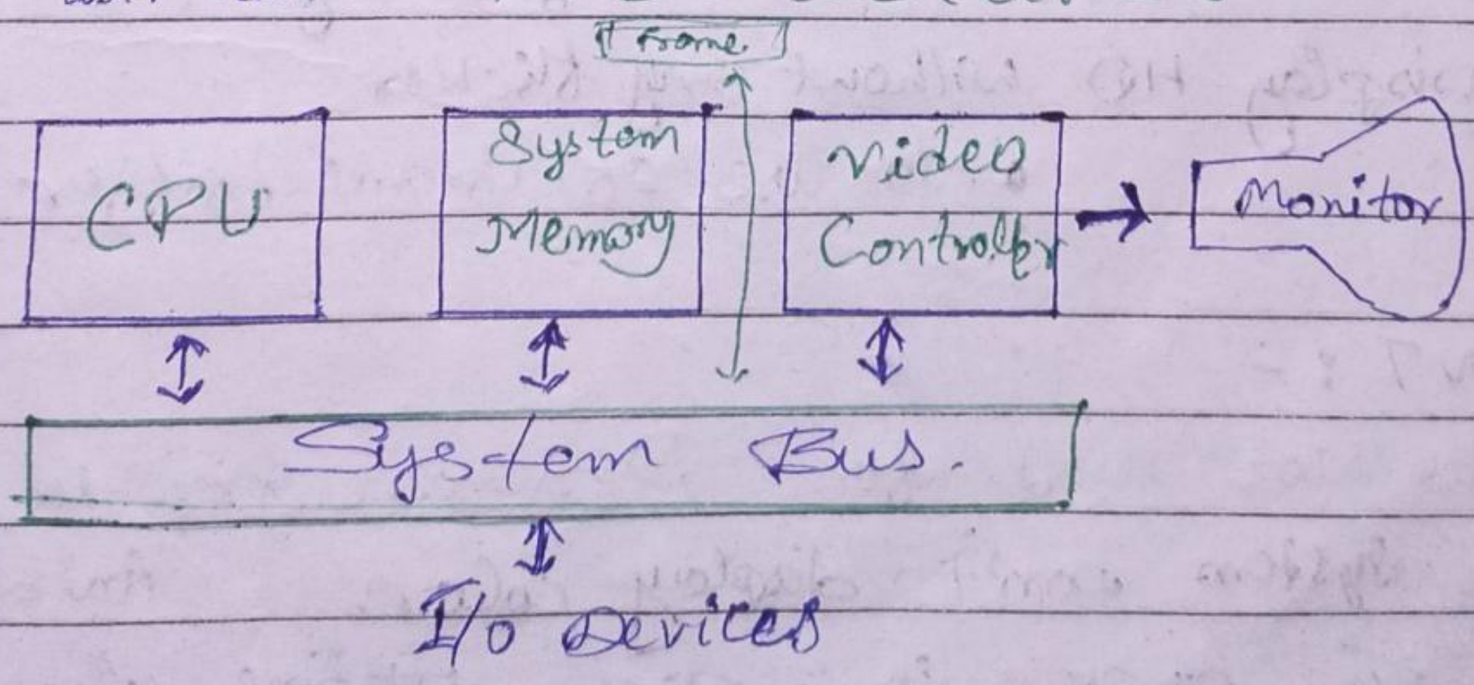
Basic



* Final :-



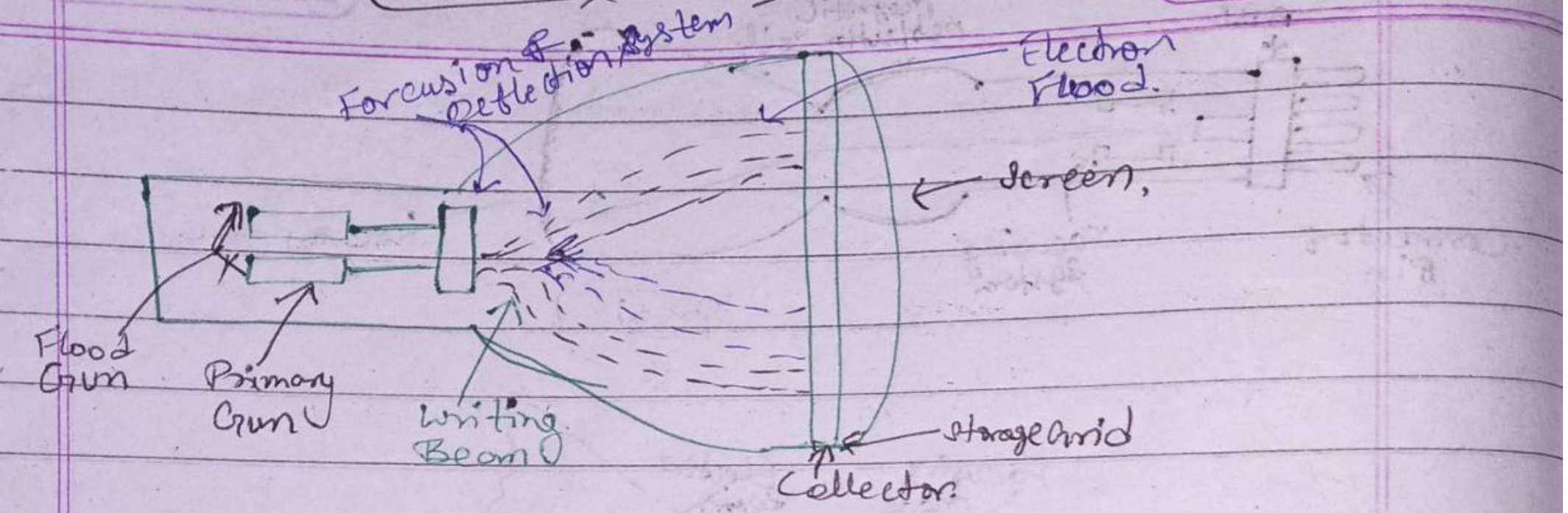
* Raster Scan Architecture :-



DVST → No need Refreshing.

Direct View Storage Tube

PAGE NO.
DATE : / /20



* DVST resembles CRT as it uses electron gun to draw picture and phosphor coated screen to display it.

↳ DVST doesn't use refresh buffer or frame buffer. to store picture defination.

* Components :-
  (1) Electron Gun. (2) Phosphore coated Screen.
  (3) Storage Mesh (4) collector.

* ADVNTG :-
  - Not require refreshing
  - Can display HD without any flicker
  - No use of frame buffer or refresh.

* DADVNT :-
  - Not used for Dynamic graphic such as Animation
  - These System don't display color.
  - To erase some part, we have clear All.

Raster System * Important

PAGE NO.
DATE : / /20

Q. Three different raster system, with resolution of 640 x 400 ; 1280 x 1024 & 2560 x 2048. What is size frame buffer (in byte) is needed for each of these system to store 12 bits per pixel

Ans Because Eight bits constitute a byte, we know

1 byte = 8 bit.

Frame - buffer sizes of the system are as follow :-

$$= \frac{640 \times 480 \times 12 \text{ bits}}{8} \text{ - we need in byte} = \boxed{450 \text{ KB}} \text{ Ans}$$

$$= \frac{1280 \times 1024 \times 12 \text{ bits}}{8} = \boxed{1920 \text{ KB}} \text{ Buffer.}$$

$$= \frac{2560 \times 2048 \times 12 \text{ bits}}{8} = \boxed{7680 \text{ KB}} \text{ Buffer need.}$$

Q).

Ans

Since 60 frames are refreshed per second & each frame consists of 640 x 480 pixels.

the access rate of such a system is (640 x 480) * 60 = $\boxed{1.8437 \times 10^7 \text{ pixel/sec}}$.

(2) Ans for 1280 x 1024 System, the access rate is

(1280 x 1024) * 60 = $\boxed{7.86432 \times 10^7 \text{ pixel/sec}}$

(3) Ans Access time is around $\boxed{54}$ nanosecond/pixel for 640 x 480,

$\boxed{12.7 \text{ ns/pixel}}$ for 1280 x 1024.