CGV ASSIGNMENT

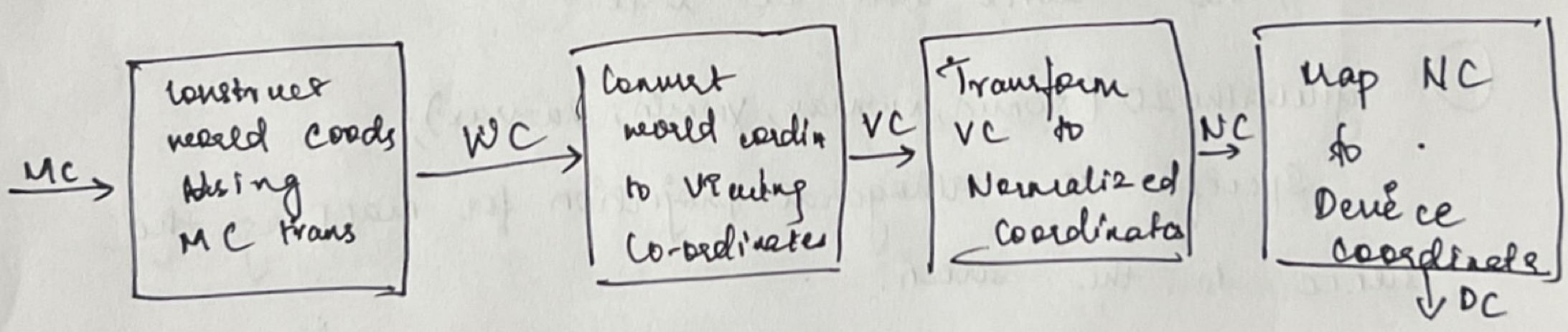
NAME - MENAKA GOWDA A
USN - 1BY20CS109
BATCH - B1

1. Build a 2D viewing transformation pipeline and also explain OpenGL 2D viewing functions.

Soln The mapping of a 2 dimensional, world-cordinate scene description to device cordinates is called 2-D viewing transformation.

This transformation is simply reffered to as the window to viewport transformation or windowing transformation.

We can describe the steps.



Once a world-coordinates scene has been constructed, we could set up a separate two dimensional, viewing co-ordinates reference frame.

At the final step of the viewing transformation, the contents of the viewport are transferred to positions within the display window.

Clipping is usually performed in normalized coordinates. This allows us to reduce computations by first concatenating.

Viewing functions.

① glMatrixMode (GL_PROJECTION)

② glLoadIdentity ();

This ensures that each time we enter the projection mode, the matrix will be reset.

③ gluOrtho2D (xwmin, xwmax, ywmin, ywmax);

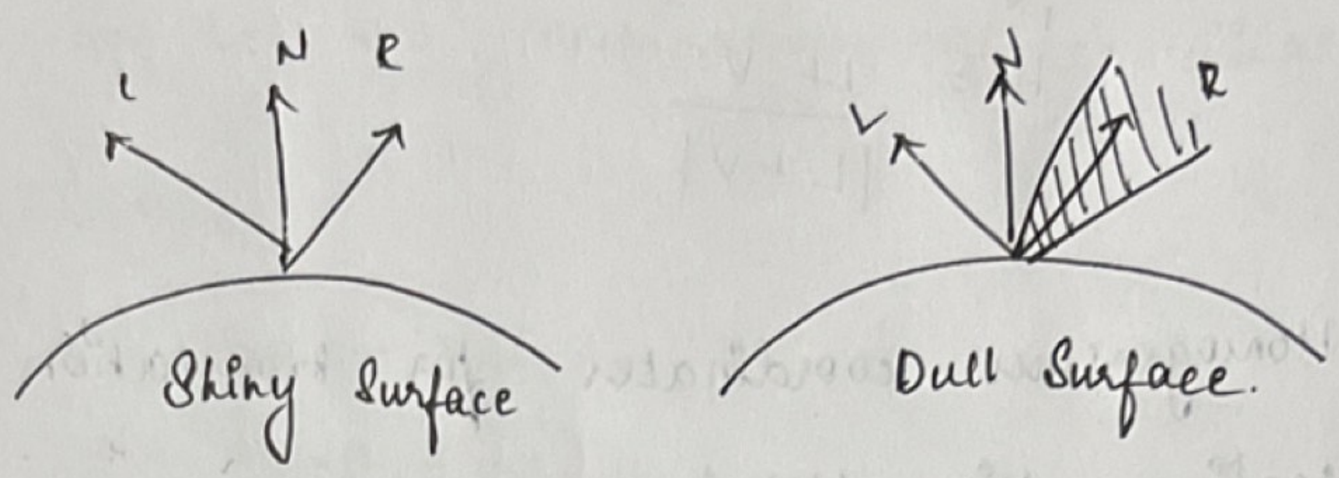
Specifies an orthogonal projection for mapping the scence to the screen.

④ glViewPort (xwmin, ywmin, width, height);

specify the position of the lowerleft corner of the viewport.

2. Build Phong Lighting Model with Equations.

An Empirical Model for calculating the specular reflection range, developed by Phong Bui Toung and called the Phong Model.



Using the spectral-reflection function $W(\theta)$ we can write the Phong specular-reflection model as

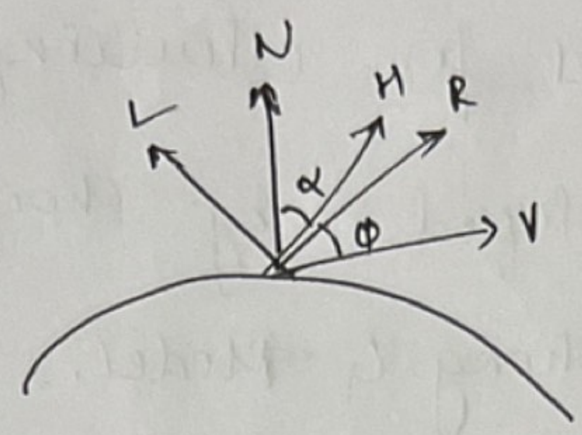
$$I_{spec} = W(\theta) I_l \cos^{n_s} \phi$$

$$I_{spec} = \begin{cases} K_s I_l (V.R)^{n_s} & \text{if } V.R > 0 \\ 0.0 & \text{if } V.R \leq 0 \end{cases}$$

The projection of L onto the direction of Normal vector has magnitude equal to dot product.

$$R + L = (2.N.L)\, N$$

if we replace V.R in the Phong Model with dot product N.H this simply replaces the $\cos \phi$



The Halway Vector is obtained as

$$H = \frac{L + V}{|L + V|}$$

3. Apply Homogeneous coordinates for translation, rotation and scaling via Matrix.

<u>Soln</u> Three dimensional Translation -

A Position $P = (x, y, z)$ in 3-dimensional space is translated to a location. ~~to a~~

$$x' = x + t_x \quad y' = y + t_y \quad z = z + t_z$$

$$P = T \cdot P$$

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

Three dimensional Rotational

Rotate an object about any axis in space

- Easiest rotation axes to handle are those that are parallel to cartesian coordinate axes.
- Also can be use combinations of coordinate axis.

Along z axis -

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

$$P' = R_z(\theta) \cdot P$$

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

Three dimensional Scaling

$$P' = S \cdot P$$

The Matrix expression for the 3-D scaling transformation of a position.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

4. Outline the differences between raster scan display and random scan displays

Raster Scan	Random Scan
* Produces jagged lines that as a discreet point sets	* Random system produces smooth lines drawing.
* Less expensive	* More Expensive
* Modification difficult	* Modification easy
* Resolution low	* Resolution easy
* Solid pattern is easy to fill	* Solid pattern is difficult to fill.

5. Demonstrate OpenGL functions for window management using GLUT

* GLUT CreateWindow - Used to Create a new window
* Glut CreateSubwindow - Used to create another window within same window
* GlutSetWindow - Used to set an particular id for window.
* Glut GetWindow - used to get the window ID
* Glut PostRedisplay - To display the window again and again.
* Glut ReshapeWindow - Used for transformation of world coordinates and displaying
* Glut PopWindow / glut pushWindow - works just like a matrix an window.
* glutHideWindow - To hide the window from being displayed on screen.
* Init ()

6. Explain OpenGL Visibility Detection Functions

(a) OpenGL polygon culling Functions

Remove backface, frontface of an object

glCullFace(mode);

glEnable(GL_CULL_FACE);

glDisable(GL_CULL_FACE);

(b) Depth-Buffer Functions

glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH)

GLUT_RGB | GLUT_DEPTH)

glClear(GL_Depth_BufferBIT)

This works as initialization function for depth buffer and refresh buffer.

glClear(GL_DEPTH_BufferBIT)

glClearDepth(MaxDepth)

glEnable(GL_DEPTH_TEST)

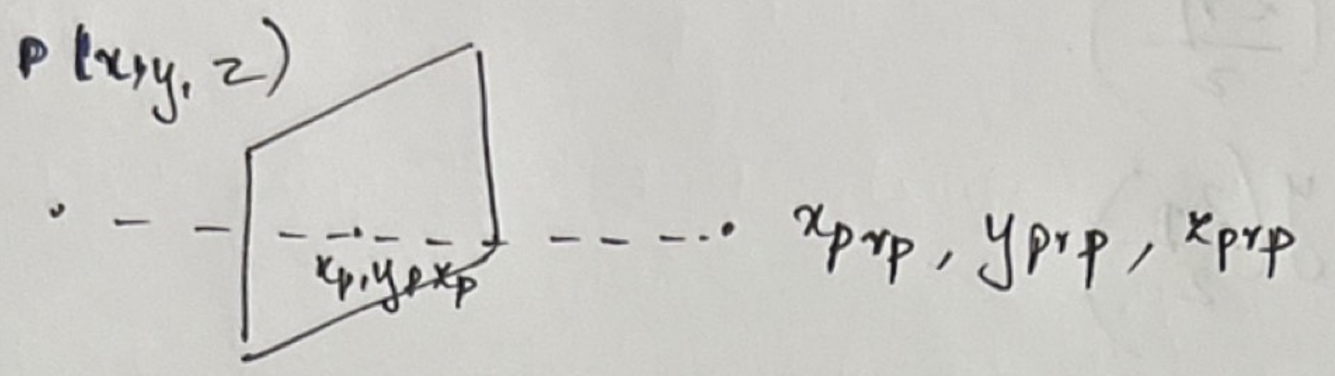
(c) OpenGL wireframe surface visibility Methods

glPolygonMode.

7. Write special cases discussed with perspective.

(5)

Projections



Consider,

$$x' = x - (x - x_{prp})\,\mu$$

$$y' = y - (y - y_{prp})\,\mu$$

$$z' = z - (z - z_{prp})\,\mu$$

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

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

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

Special cases :-

(i) $x_{prp}, y_{prp} = 0$

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

(2) $x_{prp}, y_{prp}, z_{prp} = (0,0,0)$;

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

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

(3) $z_{vp} = 0$

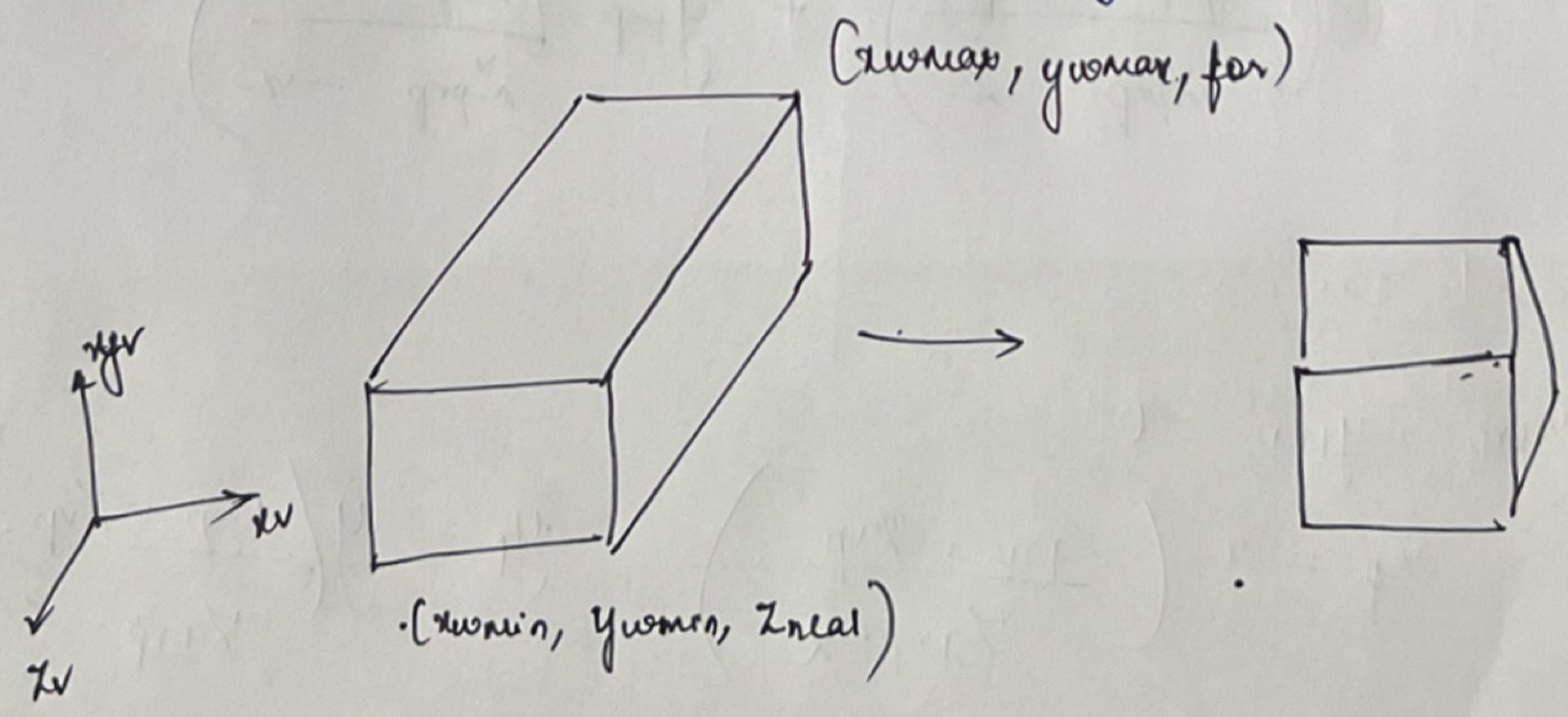
$$x_p = x\left(\frac{z_{prp}}{z_{prp} - z}\right) - x_{prp}\left(\frac{z}{z_{prp} - z}\right)$$

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

(4) $x_{prp} = 0 = y_{prp} = z_{vp}$.

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

8. Explain normalization for an orthogonal projection.



we consider a unit cube for the normalized view volume with each x, y, z coordinates normalized in the range 0 to 1

∴, we get the Normalized transformation for the ~~orgl.~~ orthogonal view volume

$$M_{ortho,norm} = \begin{bmatrix} \frac{2}{x_{wmax} - x_{wmin}} & 0 & 0 & -\frac{x_{wmax} + x_{wmin}}{x_{wmax} - x_{wmin}} \\ 0 & \frac{2}{y_{wmax} - y_{wmin}} & 0 & -\frac{y_{wmax} + y_{wmin}}{y_{wmax} - y_{wmin}} \\ 0 & 0 & \frac{-2}{z_{near} - z_{far}} & \frac{z_{near} + z_{far}}{z_{near} - z_{far}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

9. Explain Bezier Curve and its properties with equations

Bezier curves are parametric curves that are generated.

It is widely used in graphics and other related industry.

There are named after the French-engineer Pierre Bezier who discovered it.

$$\sum_{k=0}^{n} P_i B_i^n(t)$$

$B_i^n(t)$ represents Bernstein Polynomial.

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-1} t^i$$

n - polynomial degree

t = variable

i = index

n = control points number - 1

t = 0 - 1 (range)

we used the above formulae Bezier curve

$f(r) = (1-t)^{n-t+n-1}$ for each point.

10. Cohen - Sutherland line clipping algorithm.

- Cohen sutherland algorithm works on Region code
  - Region code is 4-bit code

(A B R L)

(T B R L) - top Bottom, Right, Left

1001	1000	1010
0001	0000	0010
0101	0100	0110

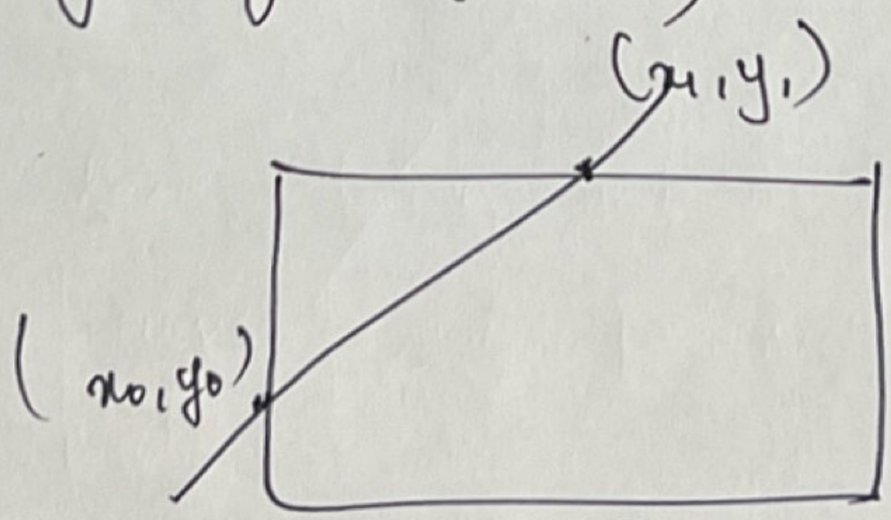
for a line - (x_0, y_0) to (x_{end}, y_{end})

$$m = (y - y_0)/(x - x_0)$$

$$m(x - x_0) = (y - y_0)$$

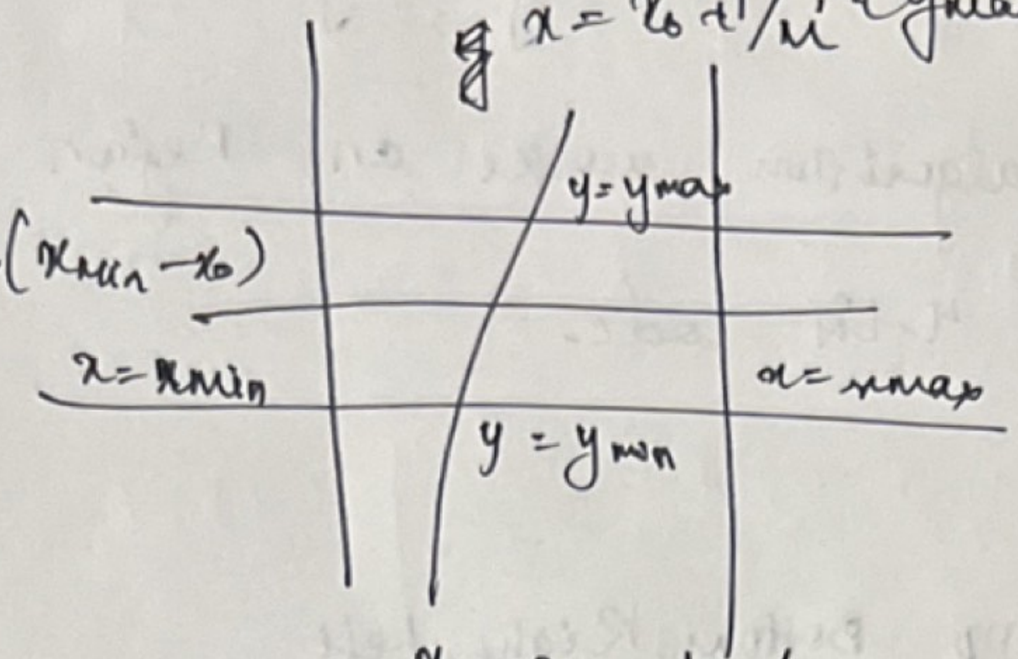
$$x = x_0 + (y - y_0)/m$$

$$y = y_0 + m(x - x_0)$$



$$x = x_0 + 1/m\,(y_{max} - y_0)$$

$$y = y_0 + m(x_{min} - x_0)$$



$$y = y_0 + m\,(x_{max} - x_0)$$

$$x = x_0 + 1/m\,(y_{min} - y_0)$$

These above formulas to be applied when a particular pipe needs to be clipped.