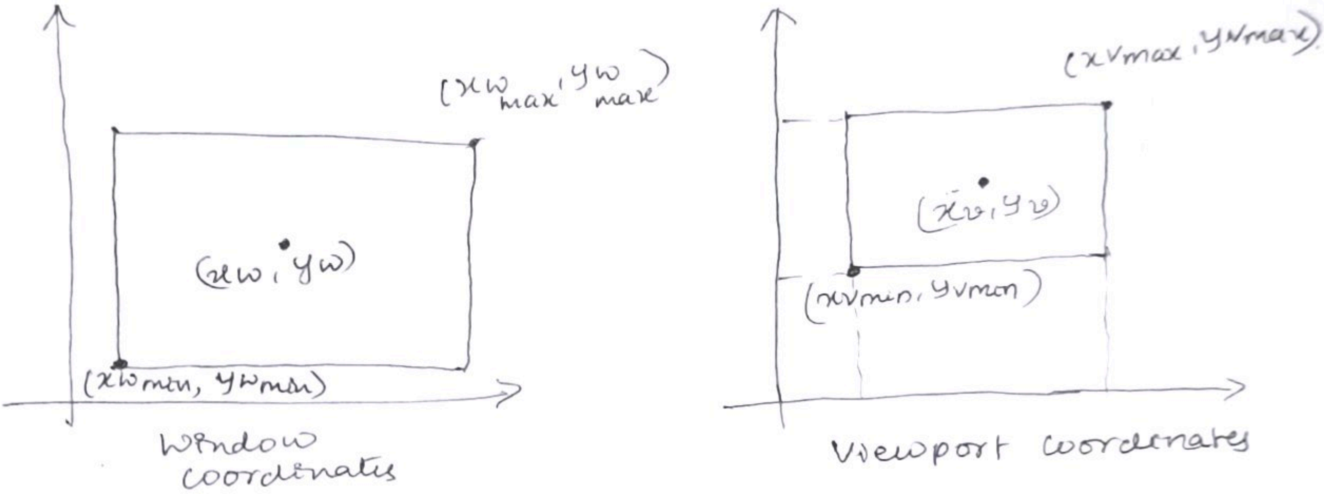
Prof. A N Ramyashree
CSE Department, RNSIT

Window to Viewport transformations

Given that output primitives are specified in world coordinates, the graphics subroutine package should be told how to map from world coordinates to screen coordinates.

Application programmer specify a rectangular region in world coordinate called world coordinate window & corresponding rectangular region in screen coordinate is called the viewport in which the world coordinate window is mapped.

Window to Viewport Transformation



A point at position (xw, yw) in a designated window is mapped to viewport coordinates (xv, yv) so that relative positions in the two areas are same.

To maintain the same relative placement in the viewport as in the window we require

$$\frac{x_v - xv_{min}}{xv_{max} - xv_{min}} = \frac{x_w - xw_{min}}{xw_{max} - xw_{min}}$$

$$\frac{y_v - yv_{min}}{yv_{max} - yv_{min}} = \frac{y_w - yw_{min}}{yw_{max} - yw_{min}}$$

Solve the equations for the viewport position (xv, yv)

$$\boxed{\begin{aligned} x_v &= xv_{min} + (xw - xw_{min})\, S_x \\ y_v &= yv_{min} + (yw - yw_{min})\, S_y \end{aligned}}$$

The scaling factor

$$S_x = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$S_y = \frac{yv_{max} - yv_{min}}{yw_{max} - xw_{min}}$$

$$\boxed{x_w = \frac{x_v - xv_{min}}{S_x} + xw_{min}}$$

$$\boxed{y_w = \frac{y_v - yv_{min}}{S_y} + yw_{min}}$$

Prof. A N Ramyashree
Department of CSE
RNSIT.

Clipping

Any procedure that eliminates those portions of a picture that are either inside or outside a specified region is referred to as clipping (clipping Algorithm)

The most common application of clipping is in the viewing pipeline, where clipping is applied to extract the designated portion of a scene (either 2D or 3D) for display on an o/p device.

The different types of clipping are:

- Point clipping
- Line clipping
- Polygon clipping (fill area clipping)
- Curve clipping
- Text clipping

Rectangular clipping window

The window is a rectangle whose sides are aligned with the coordinate axes.

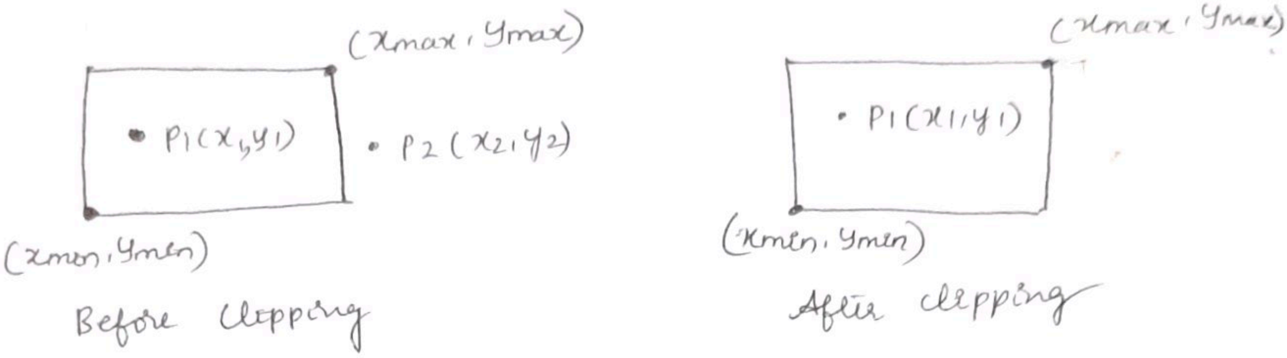
The x extent is measured from x_{min} to x_{max} and y extent is measured from y_{min} to y_{max} clipping end points

Point clipping

The point $P(x, y)$ is inside the window (visible) if all the inequalities are true i.e

$x \le x_{max}$	$x \ge x_{min}$
$y \le y_{max}$	$y \ge y_{min}$

If any of these inequalities is false then Point P is outside the window and is not displayed



Point P_1 is displayed after applying point clipping.

Line clipping

A line clipping algorithm processes each line in a scene through a series of tests and intersection calculations to determine whether the entire line or any part of it to be saved.

Cohen Sutherland Line Segment Clipping *

We divide the line clipping process into two phases.

-> Identify those lines which intersect the window and so need to be clipped.

-> Perform the clipping.

All the line segments fall into one of the following clipping categories.

1. Trivial Acceptance (Visible)

Both end points of a line segment lie within the window.

2. Trivially Rejected (Not visible)

The line segment definitely lies outside the window. This will occur, if the line segment from (x_0, y_0) to (x_1, y_1) satisfies any one of the following inequalities

$x_0, x_1 > x_{max}$	$y_0, y_1 > y_{max}$
$x_0, x_1 < x_{min}$	$y_0, y_1 < y_{min}$

3. <u>clipping candidate</u>

The line is neither category 1 or 2 then it belongs to clipping candidate category.

Cohen Sutherland Algorithm provides an efficient procedure for finding the category of line segment. The algorithm proceeds in two steps.

<u>Step 1</u>

Assign a 4 bit code to each end point of the line segment. The code is determined according to which of the following nine regions of the plane the endpoint lies in.

Top left 1001	Top 0x8 1000	Top Right 1010
0x1 Left 0001	0000 0x0	0x2 Right 0010
0101 Bottom left	0100 0x4 Bottom	0110 Bottom Right

	Top	Bottom	Right	Left
Bit	1	2	3	4

Starting from leftmost bit each bit of the code is set according to the scheme as follows:

Bit 1 = 1 if endpoint is above top edge of window ($y > y_{max}$)

Bit 2 = 1 if endpoint is below bottom edge of the window. ($y < y_{min}$)

Bit 3 = 1 if endpoint is to right of the right edge ($x > x_{max}$)

Bit 4 = 1 if endpoint is to the left of the left edge ($x < x_{min}$)

Step2 (Inside/outside) (visible/Not visible)

a. The line segment lies completely inside the clip rectangle (trivially accepted, visible) if both end point codes are 0000. i.e logical OR operation between two end point codes is false (0000)

b. Trivially rejected (Not visible) if logical AND of the code is True i.e not (0000)

c. Candidate for clipping if the logical AND of the endpoint code is 0000

Line Intersections & clipping

If bit 1 = 1 intersects with line $y = y_{max}$

If bit 2 = 1 intersects with line $y = y_{min}$

If bit 3 = 1 intersects with line $x = x_{max}$

If bit 4 = 1 intersects with line $x = x_{min}$.

The 'y' value of the intersection computed as.

$$\boxed{y = y_0 + m(x - x_0)}$$

The 'x' value of the intersection computed as

$$\boxed{x = x_0 + \frac{1}{m}(y - y_0)}$$

* Compute the outcodes of both endpoints & check for trivial acceptance & rejection.
* If neither test is successful, we find an end point that lies outside (at least one will) and then test outcode to find the edge that is crossed & to determine the corresponding intersection point.
* We can clip off the line segment from outside endpoint to intersection point by replacing outside endpoint with intersection point and compute outcode of this new point to prepare for the next iteration.

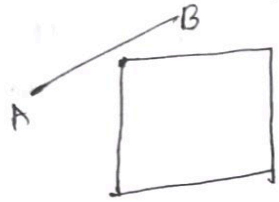
Advantage & Limitations of Cohen-Sutherland Algorithm

* It can be extended to three dimensions
* It works best when there are many line segments but few are actually displayed. In this case most of the line segments lie fully outside one or two of the extended sides of clipping rectangle and these can be eliminated on the basis of their outcodes.

Limitation

Testing & clipping are done in fixed order, sometimes algorithm performs needless clipping.

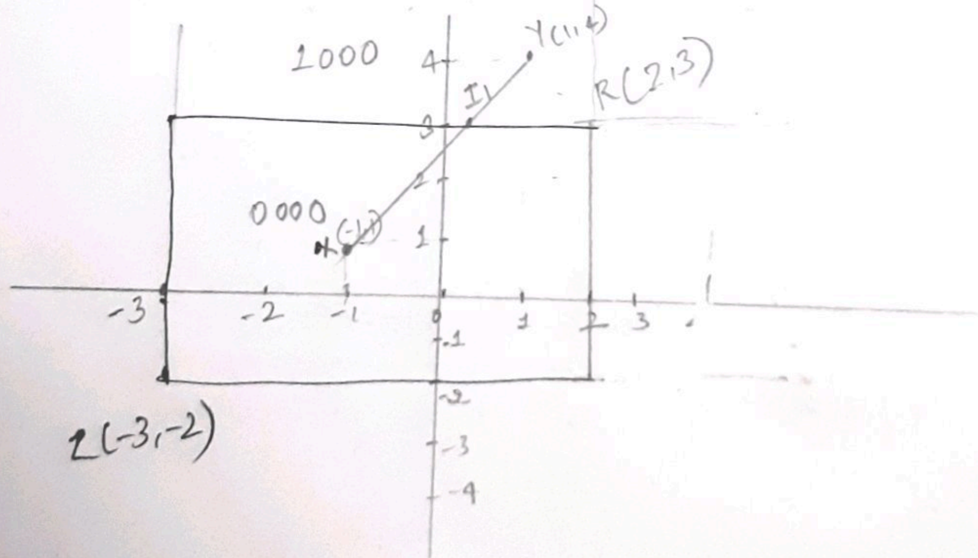
Eg:- The intersection with rectangular edge is an external intersection i.e when it does not lie on the clip rectangular boundary.



Consider line segment AB it must be clipped against both left & top sides of clipping window.

Problems

For rectangular window whose Left (-3, -2) and Right (2, 3). Find the intersection points for line XY (-1, 1) and (1, 4)



Line XY (-1, 1) and (1, 4)

find slope $M = \frac{y_2 - y_1}{x_2 - x_1}$

$$M = \frac{4-1}{1-(-1)}$$

$$m = \frac{3}{2}$$

Bit code X (-1,1): Inside the window.

0000

Bit code Y (1,4): Top part.

1000

perform Logical AND operation

0000
1000
0000

Line for clipping since logical AND of bit codes is 0000

Intersection point I_1

$$x = x_0 + 1/m\,(y - y_0)$$

$$x = -1 + (2/3)(3-1)$$

$$x = -1 + 2/3 \times 2$$

$$x = 1/3$$

Bit code Y = 1000

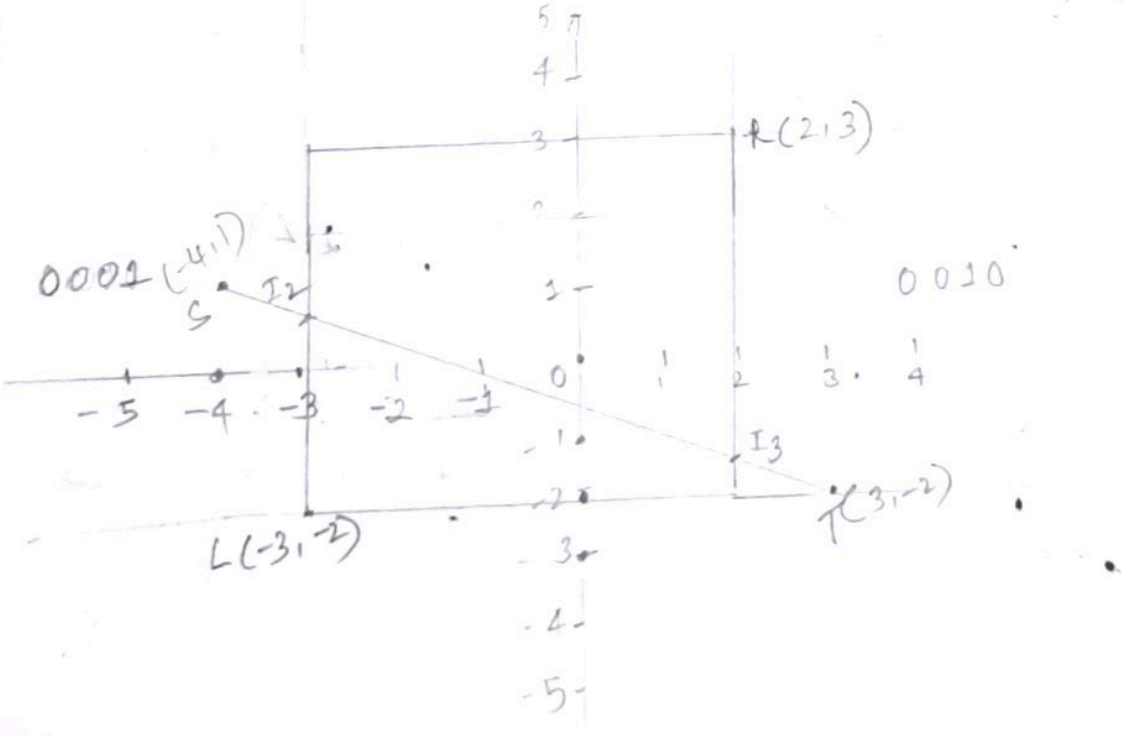
Intersects the top edge

so $y = y_{max} = 3$

The intersection point $I1 = (0.33, 3)$

Line ST (-4, 1) T(3, -2)

Window L(-3, -2)
R(2, 3)



For line ST (-4, 1) & (3, -2)

find m.

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

$$= \frac{-2 - 1}{3 - (-4)}$$

$$m = \frac{-3}{7}$$

find the opcodes. or bit codes.

(-4, 1)

lies at left side

0001

(3, -2)

lies at right side

0010

Perform Logical AND operation.

$$\begin{array}{r} 0001 \\ 0010 \\ \hline 0000 \end{array}$$ → Line for clipping.

find the two intersection points I_1 and I_3.

for I_2 $x = x_{max} = -3$

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

$= 1 + -3/7(-3 - (-4))$

$= 1 + -3/7(\cancel{-3} + 4)$

$= 1 + -3/7(1)$

$= 1 + -3/7$

$= \underline{0.56}$

$I_2 = (-3, 0.56)$

for I_3 $x = x_{max} = 2$

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

$= 1 + -3/7(2 - (-4))$

$= 1 + -3/7(2 + 4)$

$= 1 + -3/7(6)$

$= -11/7 = -1.57$

$I_3 = (2, -1.57)$

For line ST (-4, 1) & (3, -2)

find m.

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

$$= \frac{-2 - 1}{3 - (-4)}$$

$$m = \frac{-3}{7}$$

find the opcodes or bit codes.

(-4, 1)

lies at left side

0001

(3, -2)

lies at right side

0010

Perform Logical AND operation.

0001
0010
0000 → Line for clipping.

find the two intersection points I_2 and I_3.

for I_2 $x = x_{max} = -3$

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

$= 1 + -3/7(-3 - (-4))$

$= 1 + -3/7(\cancel{-3} + 4)$

$= 1 + -3/7(1)$

$= 1 + -3/7$

$= \underline{0.56}$

$I_2 = (-3, 0.56)$

for I_3 $x = x_{max} = 2$

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

$= 1 + -3/7(2 - (-4))$

$= 1 + -3/7(2 + 4)$

$= 1 + -3/7(6)$

$= -11/7 = -1.57$

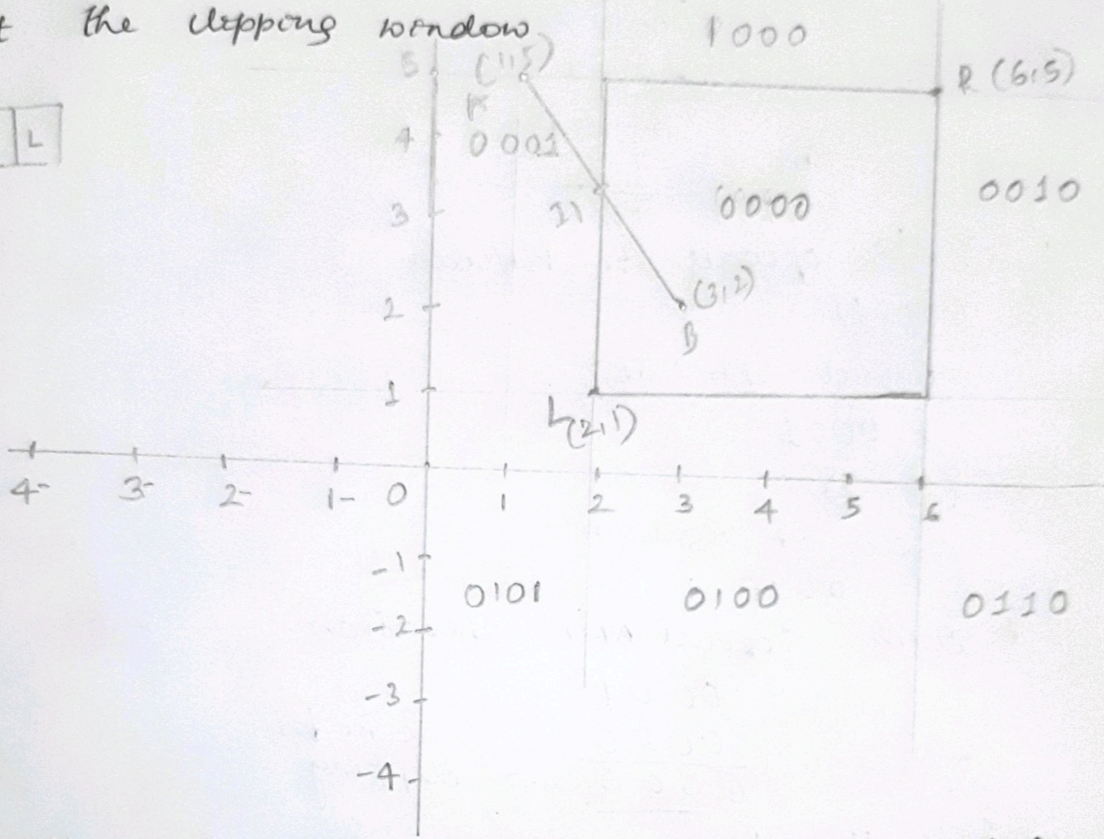
$I_3 = (2, -1.57)$

Clip the lines AB (1,5) (3,2) using Cohen Sutherland line clipping algorithm. against the clipping window L(2,1) & R(6,5)

Sol:

Plot the clipping window

T	B	R	L



Plot the line AB and observe its position relative with the opcode / or bitcode.

AB 0001 } Perform Logical AND operation.
   0000 }

0001
0000

0000 — line to be clipped.

Find slope of the line AB (1,5) (3,2)

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2-5}{3-1} = \frac{-3}{2}$$

for I_1 $x = x_{max} = 2$

$$y = y_0 + m(x - x_0)$$
$$= 5 + -3/2\,(2-1)$$
$$= 5 + -3/2\,(1)$$
$$= 7/2$$
$$= 3.5$$

$I_1 = (2, 3.5)$