05 October 2021 19:02

2D-Viewing and Clipping -

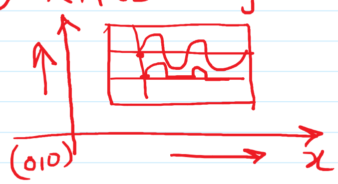
1. Basics of co-ordinate systems ⇒ 5m / 5N
2. 2D-Viewing Transformation *** 10/10m
3. Concepts of clipping [point, line, poly, text]
4. Line clipping algo → Cohen Sutherland line clipping algo; → Mid-point subdivision — " —; → Liange Barsky line clipping algo — 10/10m Numerical on any 1 algo ***
5. Polygon clipping algo → Sutherland-Hodgeman poly clip; → Weiler Artherton poly clip — 10/10m ***
6. Text clipping — 5N/5m

Introduction to coordinate system -

1. RHCS — Right Handed Coordinate System.

Cartesian coordinate sys

CRO osci



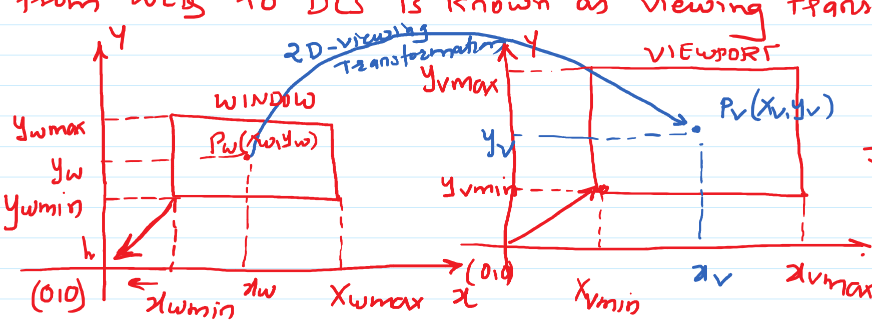
2. LHCS — Left Handed Coordinate System

ex. TV, CRT monitor, mobile screen, notebook

3. WCS — World coordinate sys.
   - WCS is RHCS
   - Phy length / True length
4. DCS — Device coordinate system
   - RHCS / LHCS
   - Logical length
5. NDCS -
6. WINDOW → WCS "what is to be displayed?"
7. VIEWPORT → DCS → "where to display?"

* 2D-Viewing Transformation - / 2D Viewing -

Transformation of an object from window to viewport OR simply from WCS to DCS is known as viewing transformation.



Inputs-

Dimension of window (x_{wmin}, y_{wmin}) & (x_{wmax}, y_{wmax})

Dimension of viewport (x_{vmin}, y_{vmin}) & (x_{vmax}, y_{vmax})

Step 1 - Translate lower left corner of window to origin.

Window pt $P_w(x_w, y_w)$

output.

Viewport pt $P_v(x_v, y_v)$

$$T1 = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_{wmin} \\ 0 & 1 & -y_{wmin} \\ 0 & 0 & 1 \end{bmatrix}$$

Step 2. – Now scale the dim. of window so that it fit to view port –

$$S = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Sx = \frac{WOV}{WOW} = \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}}$$

$$Sy = \frac{HOV}{HOW} = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}}$$

Step 3 – Now retranslate viewport to its original position

$$T2 = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & X_{vmin} \\ 0 & 1 & y_{vmin} \\ 0 & 0 & 1 \end{bmatrix}$$

Step 4. Composite matrix.

$$M_T = T2 \cdot S \cdot T1$$

$$\boxed{P_v = M_T \cdot P_w}$$

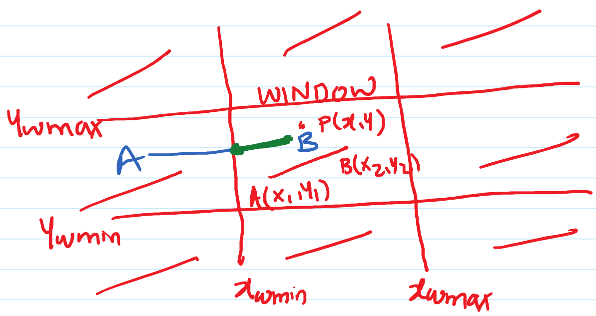
Concepts of clipping –

(a) Point clipping -

$x_{wmin} \le x \le x_{wmax}$

AND

$y_{wmin} \le y \le y_{wmax}$



(b) line clipping –

① Completely IN. –

$x_{wmin} \le x_1, x_2 \le x_{wmax}$

AND

$y_{wmin} \le y_1, y_2 \le y_{wmax}$

Display line AB & stop

② Completely OUT.

$x_1, x_2 < x_{wmin}$ OR

$x_1, x_2 > x_{wmax}$ OR

$y_1, y_2 < y_{wmin}$ OR

$y_1, y_2 > y_{wmax}$

Discard line AB & stop

③ Partially IN & partially OUT

clipping candidate

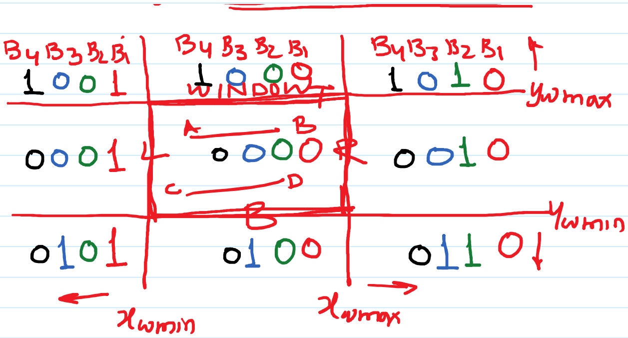
If case 1 and 2 fails then line AB is partially IN partially OUT

To clip such line we have line clipping algo

Cohen Sutherland line clipping algo →

① Accept the end point co-ordinates of line AB ie $A(X_1,Y_1)$ & $B(X_2,Y_2)$ and window boundary (x_{wmin}, y_{wmin}) & (x_{wmax}, y_{wmax})

② Assign 4 bit region-code for both end points of line AB.



If $x < x_{wmin}$ then $B_1 = 1$ else 0

If $x > x_{wmax}$ then $B_2 = 1$ else 0

If $y < y_{wmin}$ then $B_3 = 1$ else 0

If $y > y_{wmax}$ then $B_4 = 1$ else 0.

③ Check the status of line AB

ⓐ Completely IN → If the region codes for both end points are 0000 then line AB is completely IN. Display line AB and stop.

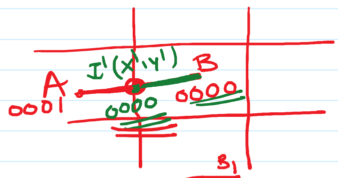
ⓑ Completely OUT → If logical AND operation betn two end point codes are NOT 0000 then the line AB is completely OUT. Discard line AB & stop.

ⓒ Partially IN partially OUT (Clipping candidate) ⇒ If case a and b fails then line AB is clipping candidate. To clip such line go to step 4.

④ Determine intersection boundary –

Check the region code for outside point.

$P = \underset{T}{B_4}\ \underset{B}{B_3}\ \underset{R}{B_2}\ \underset{L}{B_1}$



If $B_1 = 1$ ⇒ Line intersect with LEFT boundary

If $B_2 = 1$ ⇒ —"— RIGHT —"—

If $B_3 = 1$ ⇒ —"— BOTTOM —"—

If $B_4 = 1$ ⇒ —"— TOP —"—

B_1

A → 0001 &

B → 0000

0000

⇓

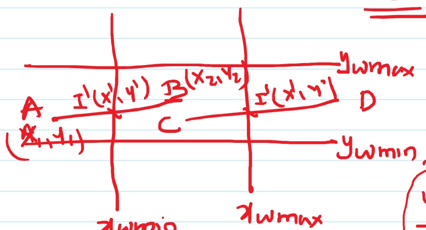
Not out

⇓

CC

⑤ Determine intersection point – $I'(x',y')$

ⓐ LEFT/RIGHT –



$x' = x_{wmin}$ (Left)
OR
x_{wmax} (Right)

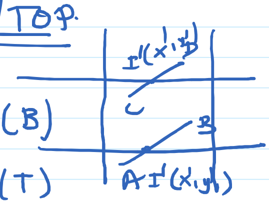
$$\frac{x'-x_1}{x_2-x_1} = \frac{y'-y_1}{y_2-y_1}$$

$$\frac{y_2-y_1}{x_2-x_1}(x'-x_1) = y'-y_1$$

$$\boxed{y' = y_1 + m(x'-x_1)}$$

$I'(x',y')$

ⓑ BOTTOM/TOP



$y' = y_{wmin}$ (B)
OR
y_{wmax} (T)

$$\frac{x'-x_1}{x_2-x_1} = \frac{y'-y_1}{y_2-y_1}$$

$$\boxed{x' = x_1 + \frac{1}{m}(y'-y_1)}$$

$I'(x',y')$

⑥ To determine the regioncode for I' go to step 2.

* – ↑Y 1000 $I' \Rightarrow$ 1000 & Consider the window

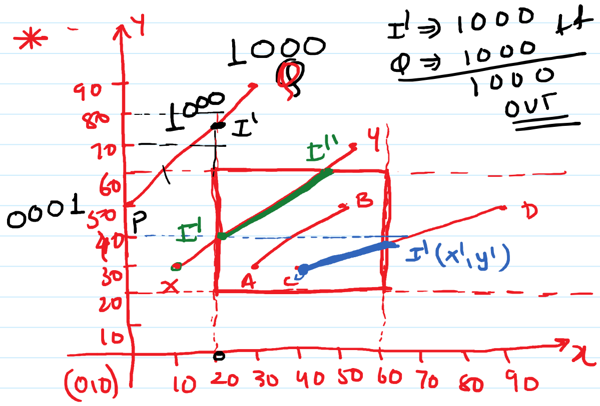
Consider the window

$(x_{wmin}, y_{wmin}) = (20, 20)$

$(x_{wmax}, y_{wmax}) = (60, 60)$

Clip the following lines wrt given window using cohen sutherland line clipping algo.

1. $AB \Rightarrow A(30,30)$ & $B(50,50)$
2. $CD \Rightarrow C(40,30)$ & $D(90,50)$
3. $XY \Rightarrow X(10,30)$ & $Y(50,70)$
4. $PQ \Rightarrow P(0,50)$ & $Q(30,90)$



① Line AB.

	B_4	B_3	B_2	B_1
	T	B	R	L
$A \Rightarrow$	0	0	0	0
$B \Rightarrow$	0	0	0	0

∵ Region code for both end pts are 0000 ⇒ line AB is completely IN

Display line AB & stop

② line CD.

$C(40,30)$ (x_1, y_1)

$D(90,50)$ (x_2, y_2)

$C \Rightarrow 0000$ ff

$D \Rightarrow 0010$

0000

∴ line CD is clipping candidate

∴ outside pt ⇒ $D = 0\,0\,1\,0$

$B_4\ B_3\ B_2\ B_1$

T B R L

∵ $B_2 = 1 \Rightarrow$ line intersect with Right boundary

$\boxed{x' = x_{wmax} = 60}$

$y' = y_1 + m(x' - x_1)$

$= 30 + \frac{50-30}{90-40}(60-40)$

$= 30 + \frac{20}{50}(20)$

$y' = 38$

x_{wmin} y_{wmin} $(20, 20)$

$(60, 60)$ x_{wmax} y_{wmax}

$\boxed{I'(x', y') = (60, 38)}$

If $x < x_{wmin} \Rightarrow B_1 = 1$ else 0

$x > x_{wmax} \Rightarrow B_2 = 1$ else 0

$y < y_{wmin} \Rightarrow B_3 = 1$ else 0

$y > y_{wmax} \Rightarrow B_4 = 1$ else 0

Region code for I'

	B_4	B_3	B_2	B_1
$I' =$	0	0	0	0
$C \Rightarrow$	0	0	0	0

Display line CI' stop

③ line XY $X(x_1, y_1) = (10, 30)$

$Y(x_2, y_2) = (50, 70)$

$X \Rightarrow 0001$ ff → Left

$Y \Rightarrow 1000$

0000 ⇒ clipping candidate → top

$I' \Rightarrow 0000$

$I'' \Rightarrow 0000$

Display line $I'I''$

④ line PQ

$P \Rightarrow 0001$ ff

$Q \Rightarrow 1000$

0000 ⇒ Clipping candidate

Outside pt ⇒ $P \Rightarrow 0\,0\,0\,1$

$B_4\ B_3\ B_2\ B_1$

T B R L

$P(0, 50)$

$Q(30, 90)$

∵ $B_1 = 1 \Rightarrow$ line intersect with Left

$\boxed{x' = x_{wmin} = 20}$

$y' = y_1 + m(x' - x_1)$

$= 50 + \frac{90-50}{30-0}(20-0)$

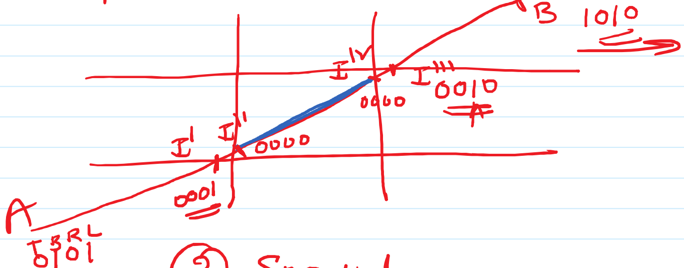
$= 50 + \frac{40}{30}(20)$

$\boxed{y' = 76.66}$

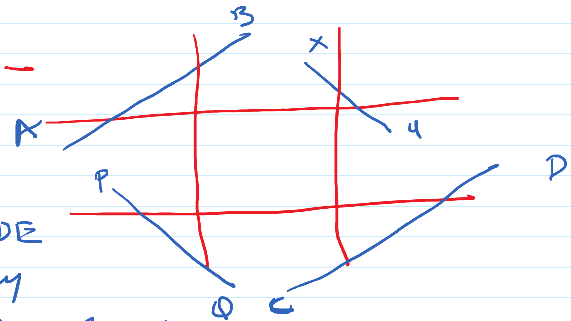
$I'(x', y') = (20, 76.66)$

Limitations of Cohen Sutherland line clipping algo

(1) Complex mathematical calculations are involved and time consuming algo if line ~~is~~ intersect with multiple window boundaries



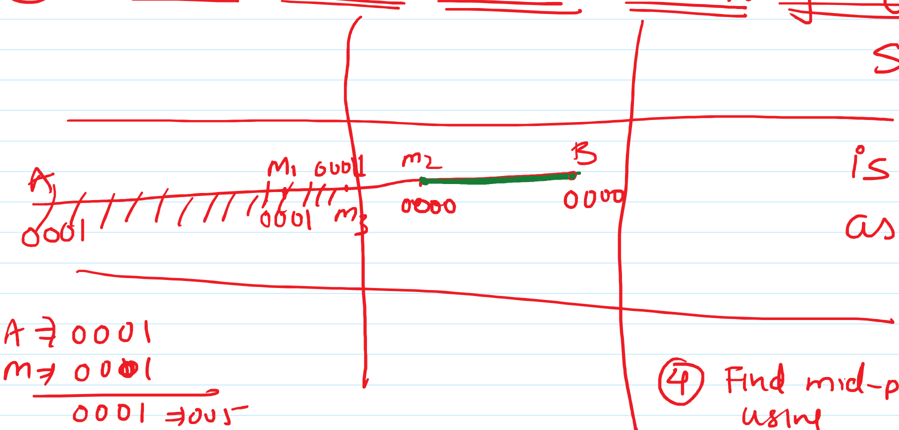
(2) Special case -



By observation, it is clearly vissible that the lines are completely OUTSIDE the window but ~~un~~necessary algo requires more than one iteration to specify that these lines are completely OUT.

(3) Drawback ⇒ This algo is only applicable to clip a line against rectangular window boundary.

(2) Mid-point subdivision line clipping algo -



Step 1 to step 3 is exactly same as Cohen Sutherland line clipping algo

A ⇒ 0001
m ⇒ 0001
0001 ⇒ OUT

(4) Find mid-point of line AB using

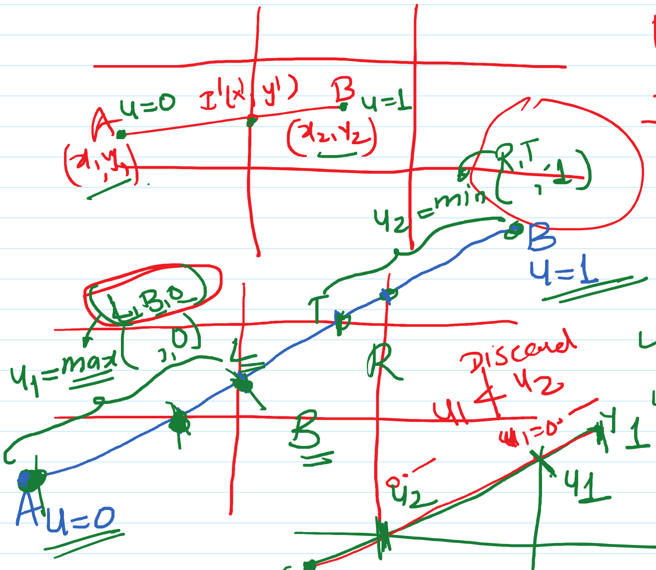
$$M = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right)$$

Midpoint M divides the line AB into two equal parts.

Process two equal parts of line AB seperately by finding region code for M,

Liange-Barsky line clipping algo →

principles ⇒ Parametric eqn of line segment.



Parametric eqn

$$\frac{x'-x_1}{x_2-x_1} = \frac{y'-y_1}{y_2-y_1} = u$$

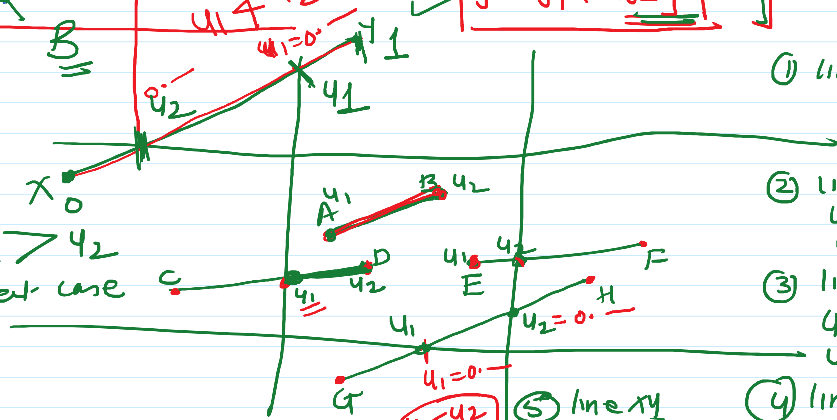
parametric const

$0 \le u \le 1$

$$\frac{x'-x_1}{x_2-x_1} = u$$

$$\boxed{x' = x_1 + u\Delta x}$$

$$\boxed{y' = y_1 + u\Delta y}$$



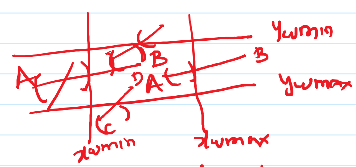
① line AB
$u_1 = 0$
$u_2 = 1$

② line CD
$u_1 = 0.$—
$u_2 = 1$

③ line EF
$u_1 = 0$
$u_2 = 0.$—

④ line GH
$u_1 < u_2$
$u_1 = 0.$—
$u_2 = 0.$—

⑤ line XY
$u_1 > u_2$
line is out



Liang-Barsky line clipping algo. ⇒

① Accept end point coordinates of line AB ie $A(x_1, y_1)$ + $B(x_2, y_2)$ and window boundary (X_{wmin}, Y_{wmin}) + (X_{wmax}, Y_{wmax})

② Find P_K and q_K for $K = 1, 2, 3, 4$

Left $P_1 = -\Delta x$ $q_1 = x_1 - x_{wmin}$

Right $P_2 = \Delta x$ $q_2 = x_{wmax} - x_1$

Bottom $P_3 = -\Delta y$ $q_3 = y_1 - y_{wmin}$

Top $P_4 = \Delta y$ $q_4 = y_{wmax} - y_1$

③ If $P_K == 0$ AND $q_K < 0$ then line AB is completely OUTSIDE
Discard line AB
STOP.

④ Calculate $r_K = \frac{q_K}{P_K}$ for $K = 1, 2, 3, 4$

$r_1 = \frac{q_1}{P_1}$ $r_3 = \frac{q_3}{P_3}$

$r_2 = \frac{q_2}{q_2}$ $r_4 = \frac{q_4}{P_4}$

⑤ Cal of u_1 and u_2

$u_1 = \max(r_K, 0)$ | $P_K < 0$
Consider only those r_K whose $P_K < 0$

$u_2 = \min(r_K, 1)$ | $P_K > 0$
consider only those r_K whose $P_K > 0$

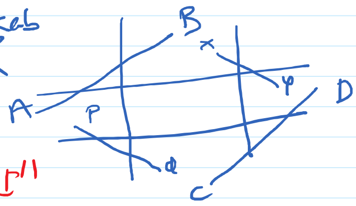
$r_2 = \frac{q_2}{p_2}$ $\quad r_4 = \frac{q_4}{p_4}$

$u_2 = \min(r_k, 1) \mid p_k > 0$

consider only those r_k whose $p_k > 0$

⑥ If $u_1 > u_2$ then line AB is completely outside
Discard line AB and stop

aaise kab hoga



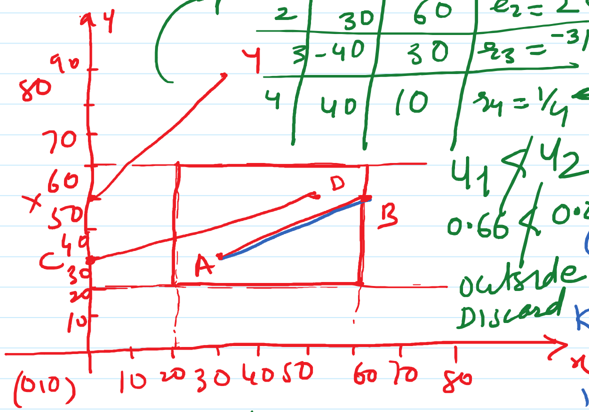
⑦ Cal I' & I''

$I' \begin{cases} x' = x_1 + u_1 \Delta x \\ y' = y_1 + u_1 \Delta y \end{cases}$

$I'' \begin{cases} x'' = x_1 + u_2 \Delta x \\ y'' = y_1 + u_2 \Delta y \end{cases}$

Display $I' I''$
Stop

*



line xy

K	p_k	q_k	$r_k = q_k/p_k$
1	−30	−20	$r_1 = 2/3$
2	30	60	$r_2 = 2$
3	−40	30	$r_3 = -3/4$
4	40	10	$r_4 = 1/4$

$u_1 = \max(r_k, 0) \mid p_k < 0$
$= \max\left(\frac{2}{3}, -\frac{3}{4}, 0\right) \Rightarrow \boxed{u_1 = \frac{2}{3} = 0.66}$

$u_2 = \min(r_k, 1) \mid p_k > 0$
$= \min(2, 1/4, 1) \Rightarrow \boxed{u_2 = 1/4 = 0.25}$

$u_1 \not< u_2$
$0.66 \not< 0.25$
outside
Discard

① line AB $\Rightarrow A(30, 30)$ $B(60, 50)$ [x_1 y_1, x_2 y_2]

② line CD $\Rightarrow C(0, 30)$ $D(50, 50)$

③ line XY $\Rightarrow X(0, 50)$, $Y(30, 90)$ [x_1 y_1, x_2 y_2]

① line AB $[u_1 = 0,\ u_2 = 1]$

K	p_k	q_k	$r_k = \frac{q_k}{p_k}$
1	$-\Delta x$ −30	$x_1 - x_{wmin}$ 10	$r_1 = -1/3$
2	Δx 30	$x_{wmax} - x_1$ 30	$r_2 = 1$
3	$-\Delta y$ −20	$y_1 - y_{wmin}$ 10	$r_3 = -1/2$
4	Δy 20	$y_{wmax} - y_1$ 30	$r_4 = 3/2$

$u_1 = \max(r_k, 0) \mid p_k < 0$
$= \max\left(\frac{-1}{3}, \frac{-1}{2}, 0\right) \Rightarrow \boxed{u_1 = 0}$

$u_2 = \min(r_k, 1) \mid p_k > 0$
$= \min(1, 3/2, 1) \Rightarrow \boxed{u_2 = 1}$

$I' \begin{cases} x' = x_1 + u_1 \Delta x = 30 \\ y' = y_1 + u_1 \Delta y = 30 \end{cases}$

$I'' \begin{cases} x'' = x_1 + u_2 \Delta x = 60 \\ y'' = y_1 + u_2 \Delta y = 50 \end{cases}$

② line CD $C(0, 30)$ $D(50, 50)$ [x_1 y_1, x_2 y_2]
$u_1 = 0$, $u_2 = 1$

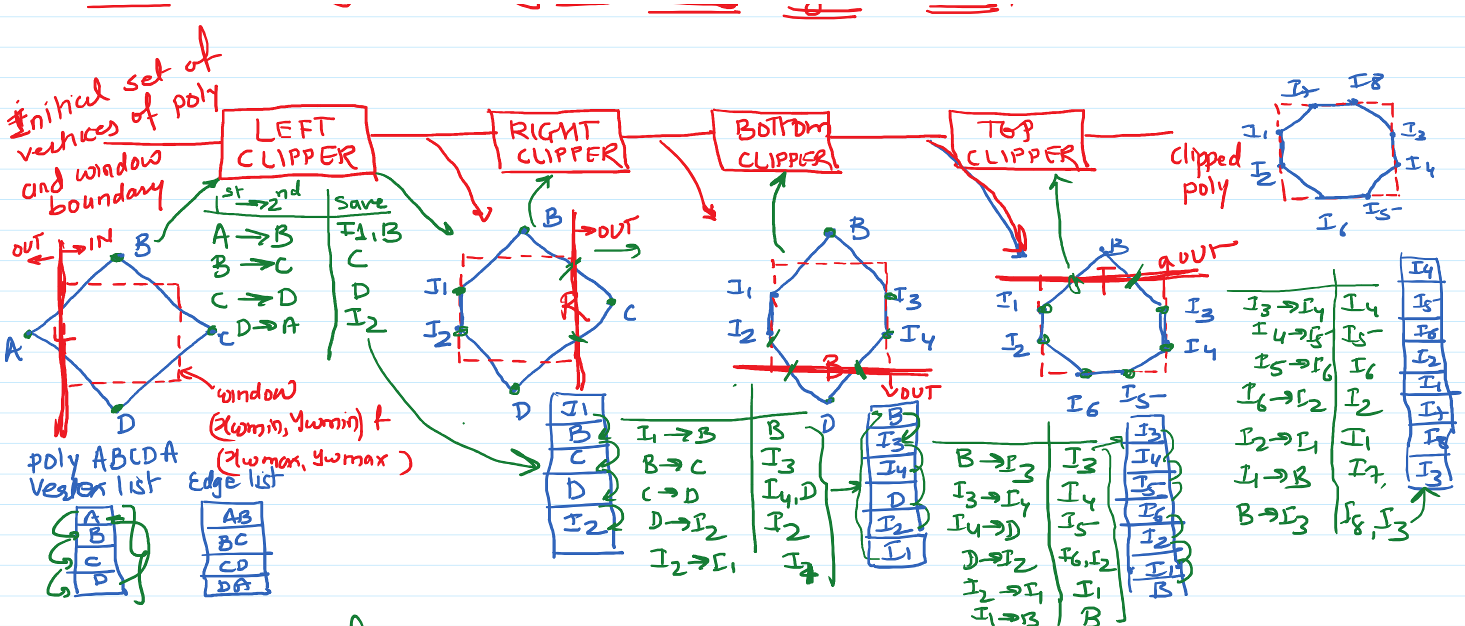
K	p_k	q_k	$r_k = \frac{q_k}{p_k}$
1	$-\Delta x$ −50	$x_1 - x_{wmin}$ −20	$r_1 = 2/5$
2	Δx 50	$x_{wmax} - x_1$ 60	$r_2 = 6/5$
3	$-\Delta y$ −20	$y_1 - y_{wmin}$ 10	$r_3 = -1/2$
4	Δy 20	$y_{wmax} - y_1$ 30	$r_4 = 3/2$

$u_1 = \max(r_k, 0) \mid p_k < 0$
$= \max\left(\frac{2}{5}, \frac{-1}{2}, 0\right) \Rightarrow \boxed{u_1 = 2/5 = 0.4}$

$u_2 = \min(r_k, 1) \mid p_k > 0$
$= \min\left(\frac{6}{5}, \frac{3}{2}, 1\right) \Rightarrow \boxed{u_2 = 1}$

* Sutherland Hodgeman polygon clipping algo → V. Imp

set of only

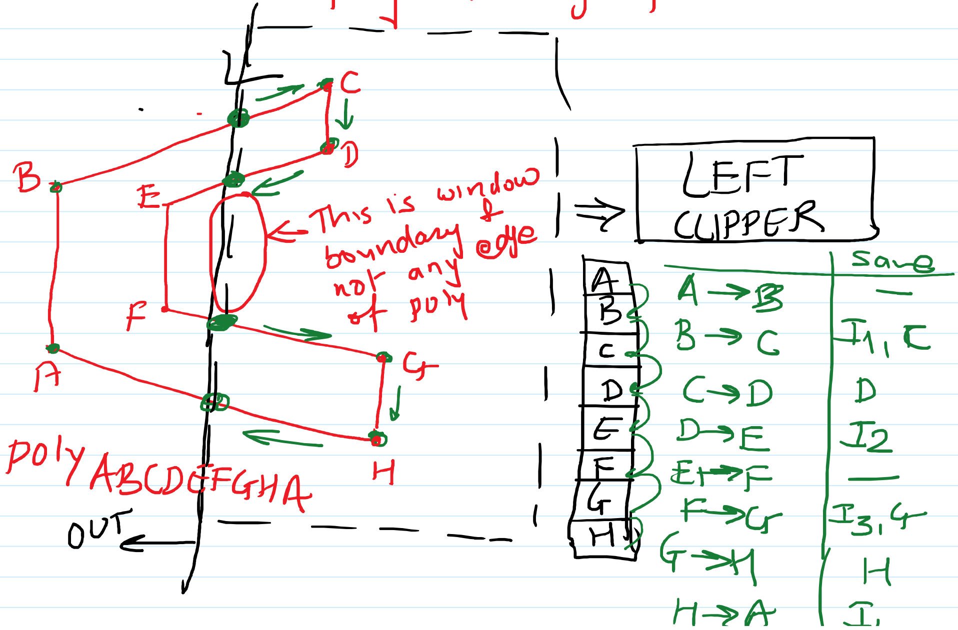


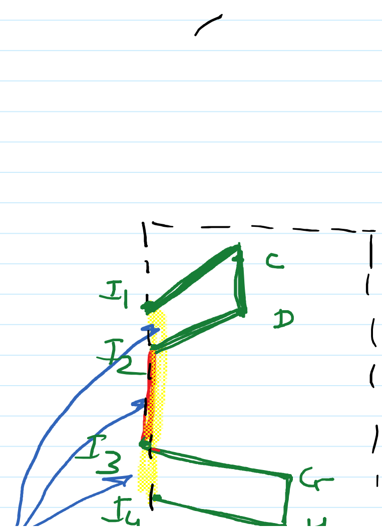
Working principles of each window clipper

1^{st} vertex	2^{nd} vertex	Save
OUT	IN	Intersection pt and 2^{nd} vertex
IN	IN	2^{nd} vertex
IN	OUT	Only intersection pt
OUT	OUT	Discard / Save nothing

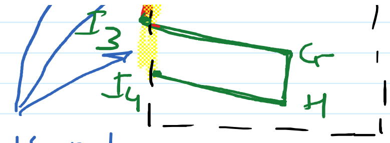
Drawback of Sutherland Hodgeman poly clipping algo -

1. This algo is only applicable for convex polygon clipping. For concave poly this algo fails





G → H | H
H → A | I_4



This is not the part of poly.

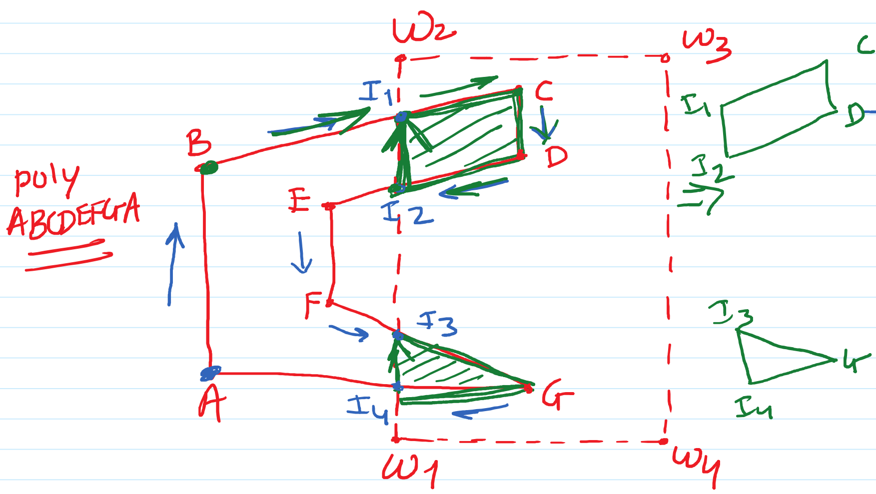
∴ CSPC fails to clip concave poly.

To clip such concave polygon we use Weiler-Atherton poly clipping algo.

Weiler-Atherton polygon clipping algo :-

Polygon vertices use the follo. rules

1. For an outside to inside pair of vertices, follow the polygon boundary
2. For an inside to outside pair of vertices follow the window boundary in clockwise direction.



Subject poly original poly	Clipping window
A	W_1
B	I_4 ✓
✓ I_1	I_3 ✓
✓ C	I_2 ✓
✓ D	I_1 ✓
✓ I_2	W_2
E	W_3
F	W_4
✓ I_3	W_1
✓ G	
✓ I_4	
A	

poly 1 ~~vertex 1~~ ⇒ $I_1 C D I_2 I_1$ vertex list of poly 1

poly 2 ⇒ $I_3 G I_4 I_3$ vertex list of poly 2