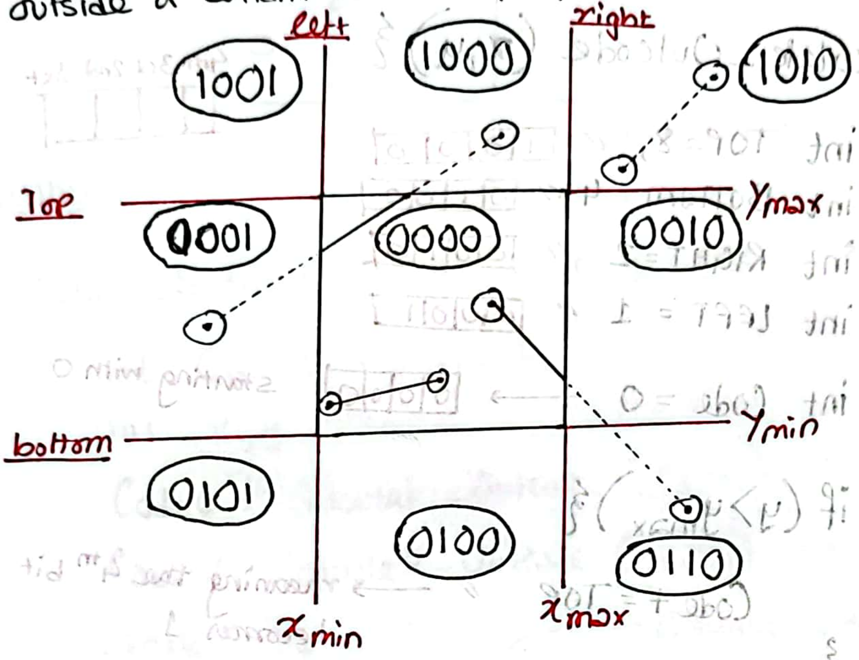
LINE CLIPPING ALGORITHMS:

COHEN-SUTHERLAND

LECTURE 7

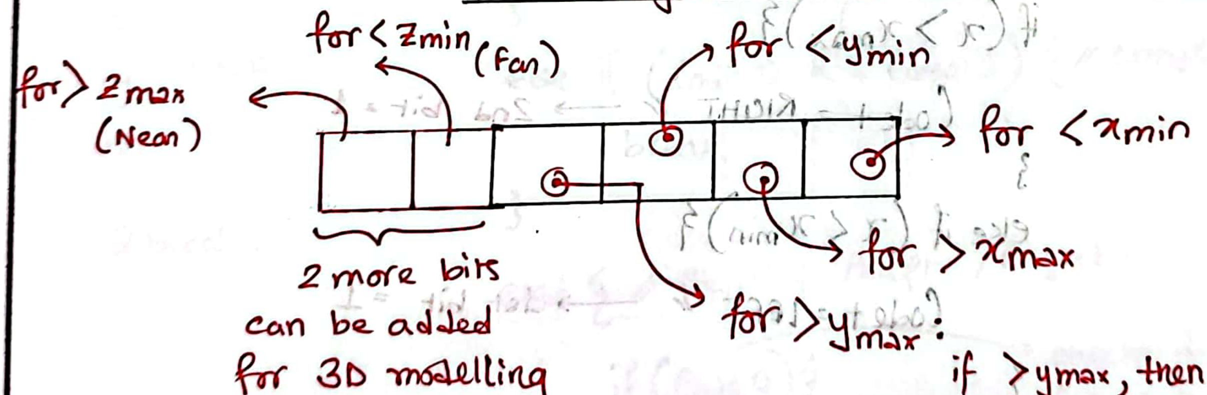
⊞ line clipping algorithms are used to clip lines that lie outside a certain window of operation.



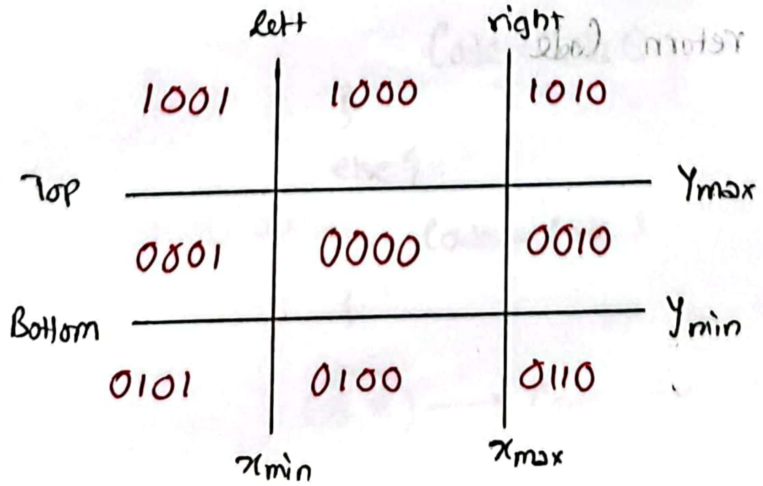
only the segment of the line inside the frame is draw, the rest clipped.

In Cohen-Sothenland we need to determine a region Outcode to determine if the line crosses the window or not.

4 bit Region Outcode: (for a 2D plane)

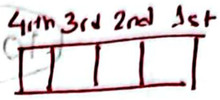


if >ymax, then → 1
else → 0



(Pseudo Code):

1) Calculating the Outcode:



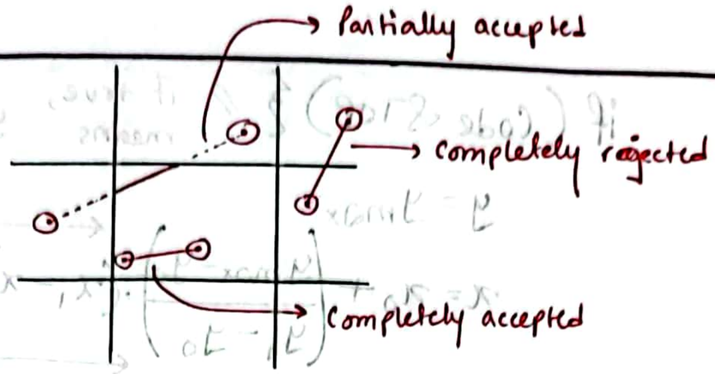
```
int Calculate_Outcode (x,y) {

    int TOP = 8,    // |1|0|0|0|
    int BOTTOM = 4  // |0|1|0|0|
    int RIGHT = 2   // |0|0|1|0|
    int LEFT = 1    // |0|0|0|1|

    int Code = 0 ⟶ |0|0|0|0|   starting with 0

    if ($y > y_{max}$){
        Code + = TOP   // ⟶ meaning the 4th bit becomes 1
    }
    else if($y < y_{min}$){
        Code + = BOTTOM // ⟶ 3rd bit becomes 1
    }
    if($x > x_{max}$){
        Code + = RIGHT  // ⟶ 2nd bit = 1
    }
    else if ($x < x_{min}$){
        Code + = LEFT   // ⟶ 1st bit = 1
    }

    return Code
}
```



Main Algorithm

```
Void Cohen_Sutherland (int x0, int y0, int x1 {, int y1)
        int Code, Code0, Code1
        int x, y
        Code0 = Calculate_Outcode (x0, y0) → starting point
        Code1 = Calculate_Outcode (x1, y1) → ending point.
        While (1) {
                                            Bitwise OR operation
            if (Code0 || Code1 == 0) { // complete Accept
                drawline (x0, y0, x1, y1)
                break
            }
                                            Bitwise AND operation
            else if (Code0 && Code1) { // complete reject
                break
            }
            else { // Partially Accept /Reject.
                if (Code0) {          → checks if Code 0 is a non-zero value,
                                        ✓ 0 = false
                                        ✓ 1 = True.
                    Code = Code0
                }
                else {
                    Code = Code1
                }
```

3 conditions to get out of this infinite loop →

1. Completely accept → drawline & break
2. Completely reject → do nothing & break
3. Partially accept → Clip line, drawline & break.

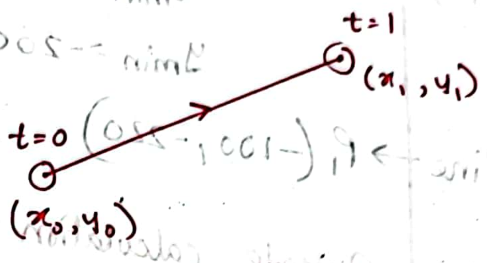
** → Continued in next Page.

```
if (Code & TOP) { // if true, means y > ymax
    $y = y_{max}$
    $x = x_0 + \left(\frac{y_{max} - y_0}{y_1 - y_0}\right) \cdot (x_1 - x_0)$      were basically using parametric equations of line.
}
else if (Code & BOTTOM) {
    $y = y_{min}$
    $x = x_0 + \left(\frac{y_{min} - y_0}{y_1 - y_0}\right) \cdot (x_1 - x_0)$
}
else if (Code & RIGHT) {
    $x = x_{max}$
    $y = y_0 + \left(\frac{x_{max} - x_0}{x_1 - x_0}\right) \cdot (y_1 - y_0)$
}
else if (Code & LEFT) {
    $x = x_{min}$
    $y = y_0 + \left(\frac{x_{min} - x_0}{x_1 - x_0}\right) \cdot (y_1 - y_0)$
}
if (Code == Code0) {
    $x_0 = x$ , $y_0 = y$      the x & y we have got is the new $x_0$ & $y_0$ after clipping
    Code0 = Calculate_Outcode ($x_0$, $y_0$)
}
else { // means we took Code 1
    $x_1 = x$ , $y_1 = y$
    Code 1 = Calculate_Outcode ($x_1$, $y_1$)
}
} } }
```

Parametric equations of a line:

∅ $x = x_0 + t\,(x_1 - x_0)$

∅ $y = y_0 + t\,(y_1 - y_0)$



~~y y~~

~~∅ y = y_0 +~~

∅ $y = y_0 + \frac{x - x_0}{x_1 - x_0} \cdot (y_1 - y_0)$

for y, $t = \frac{x - x_0}{x_1 - x_0}$ → refers to the position of x.

∅ $x = x_0 + \frac{y - y_0}{y_1 - y_0} \cdot (x_1 - x_0)$

for x, : $t = \frac{y - y_0}{y_1 - y_0}$ → refers to the position of y.

for Known value of x
we can get y.

OR

for Known value of y
we can get x.

Q1) Window →

$x_{min} = -250$, $x_{max} = 250$

$y_{min} = -200$, $y_{max} = 200$

line → $P_1(-100, -220)$ & $P_2(300, -210)$

ANS: Outcode calculation:

Outcode for P1 → 0100

Outcode for P2 → 0110

OR operation:

$$\begin{array}{r} 0100 \\ 0110 \\ \hline 0110 \end{array}$$

→ not 0, so not completely accepted.

AND operation:

$$\begin{array}{r} 0100 \\ 0110 \\ \hline 0100 \end{array}$$

→ non-zero value so, the line is completely rejected!

Q2) Clip Region $(\underset{x_{min}}{-100},\underset{y_{min}}{-120})$ to $(\underset{x_{max}}{150},\underset{y_{max}}{200})$

line given : $P_1(\underset{x_0}{-125},\underset{y_0}{260})$ to $P_2(\underset{x_1}{195},\underset{y_1}{-140})$

Ans Outcode for P1 = 1001

" " P2 = 0110

OR operation = 1111 → so, not completely accepted as non-zero value.

—x—

$$\begin{array}{r} 1001 \\ 0110 \\ \hline =0000 \end{array}$$

AND op, → so, not completely rejected as value is zero

—x—

Hence, partially accepted.

Step 1 (taking P1)

For TOP : $\begin{array}{r} 1001 \\ 1000 \\ \hline 1000 \end{array}$ → Accepted

$y = y_{max} = 200$

$x = -125 + \frac{200-260}{-140-260} \cdot (195+125)$

$= -77$

new $P_1(-77, 200)$

code of P_1 → (0000) DONE

Step 2 (taking P2)

FOR TOP : $\begin{array}{r} 0110 \\ 1000 \\ \hline 0000 \end{array}$ → Not Accepted

FOR BOTTOM : $\begin{array}{r} 0110 \\ 0100 \\ \hline 0100 \end{array}$ → Accepted

$y = y_{min} = -120$

$x = -125 + \frac{-120-260}{-140-260} \cdot (195+125)$

$= 179$

new $P_2(179, -120)$

code of P_2 → (0010) → the code is still not (0000) So, proceed to step 3

Step 3

FOR RIGHT :

$$\begin{array}{r} 0010 \\ 0010 \\ \hline 0010 \end{array} \rightarrow \text{Accepted}$$

$x = x_{max} = 150$

$y = 260 + \frac{150 + 125}{195 + 125} \cdot (-140 - 260)$

$y = -83.75$

Now $P_2 (150, -83.75)$

code for $P_2 \rightarrow (0000) \rightarrow$ DONE

Drawline $\rightarrow (-77, 200)$ to $(150, -83.75)$