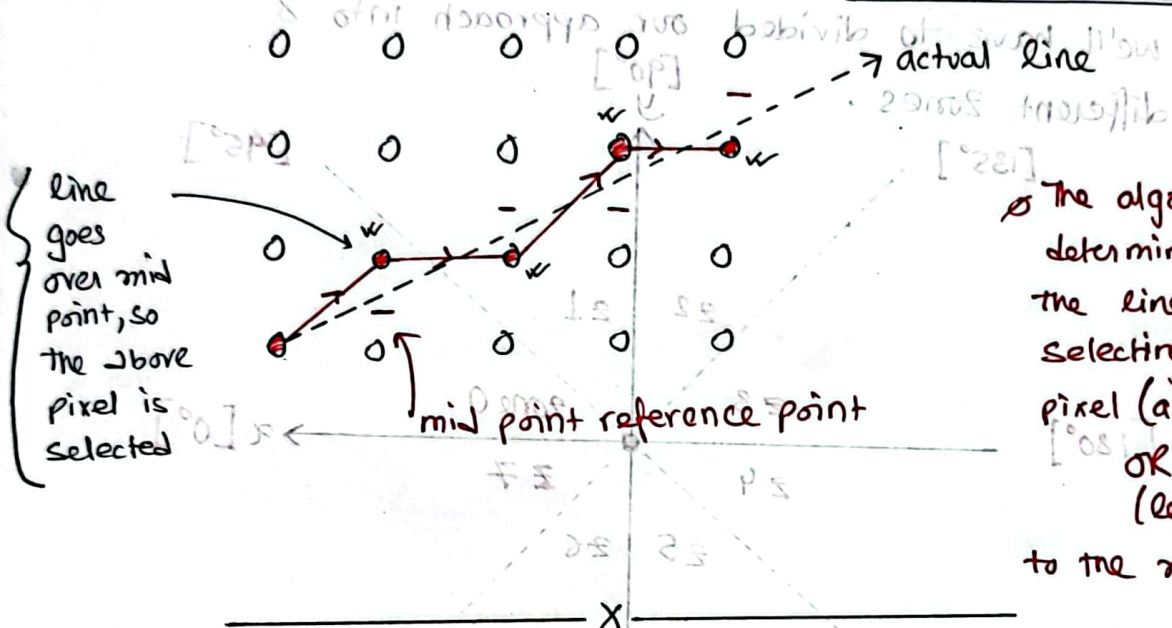


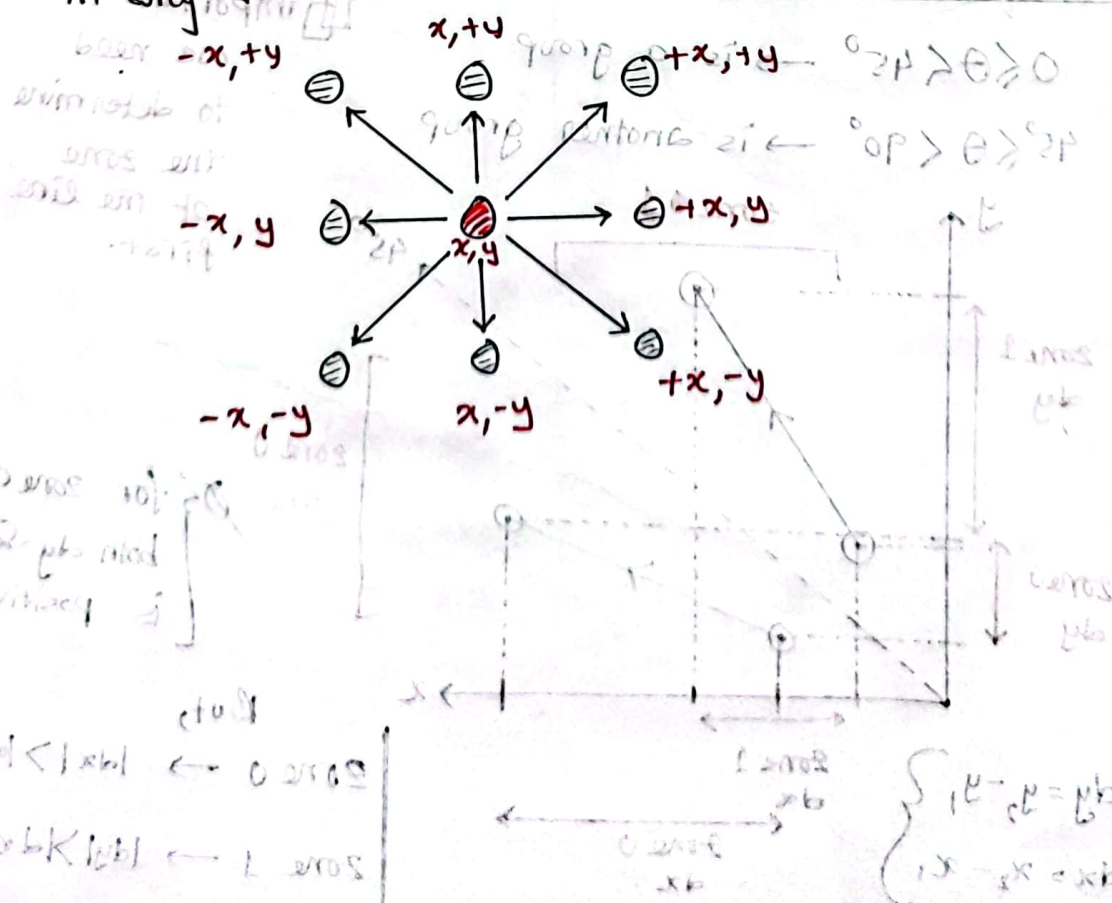
(MIDPOINT LINE DRAWING ALGORITHM) : LECTURE 3
Bresenham's Algorithm.

Bresenham's Algorithm.



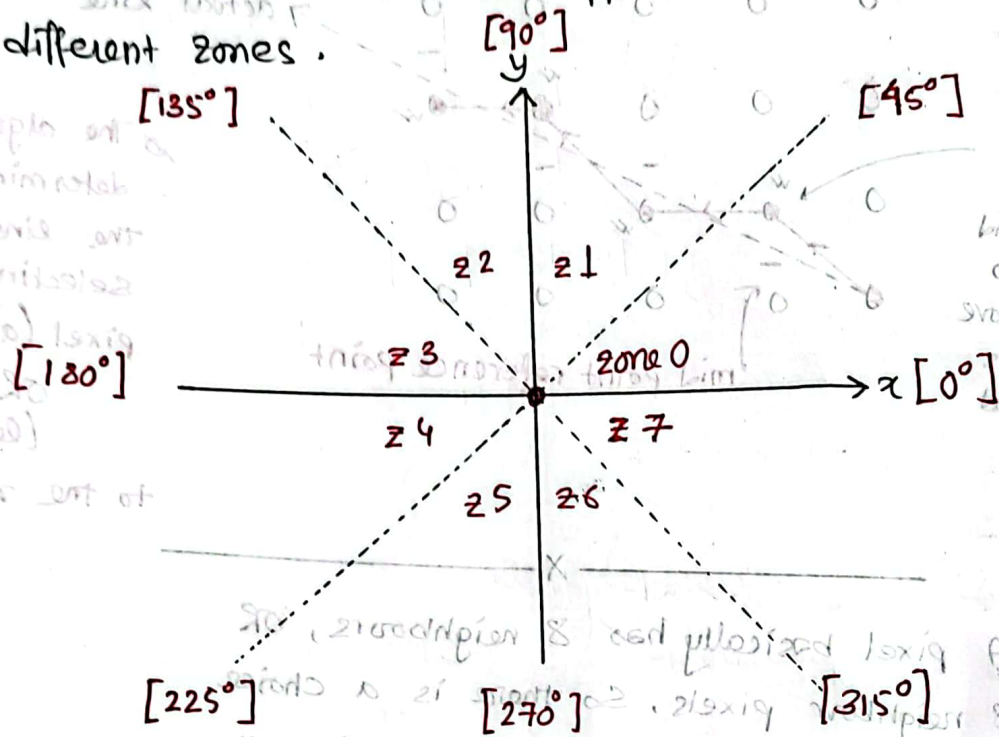
The algorithm determines if the line is selecting the pixel (above or below) OR (left or right) to the mid point.

Ø A pixel basically has 8 neighbours, OR 8 neighbour pixels. So there is a choice of 8 pixels based on the line that goes in any direction.



STEP 1: (of MPL) → Finding the zone.

Since a line can be drawn in any direction we'll have to divide our approach into 8 different zones.

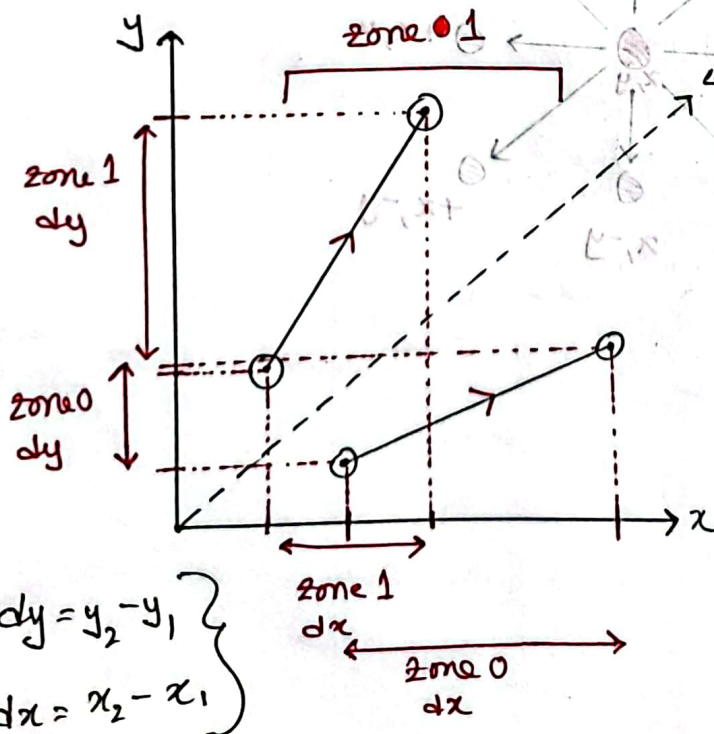


Group (0° to 90°)

$0^\circ \leq \theta < 45^\circ \rightarrow$ is a group

$45^\circ \leq \theta < 90^\circ \rightarrow$ is another group

Important: → we need to determine the zone of the line first.



$$\left. \begin{aligned} dy &= y_2 - y_1 \\ dx &= x_2 - x_1 \end{aligned} \right\}$$

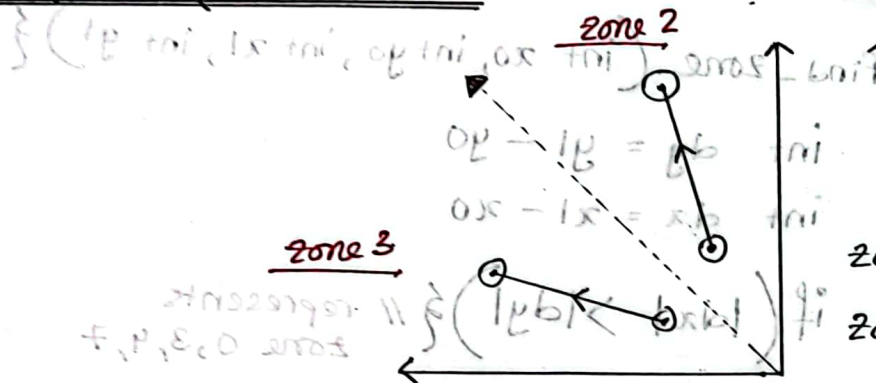
for zone 0 & 1 both dy & dx is positive

But,

zone 0 → $|dx| > |dy|$

zone 1 → $|dy| > |dx|$

Now, Group ($90^\circ - 180^\circ$)



for zone 2 & 3

$$dx = -ve$$

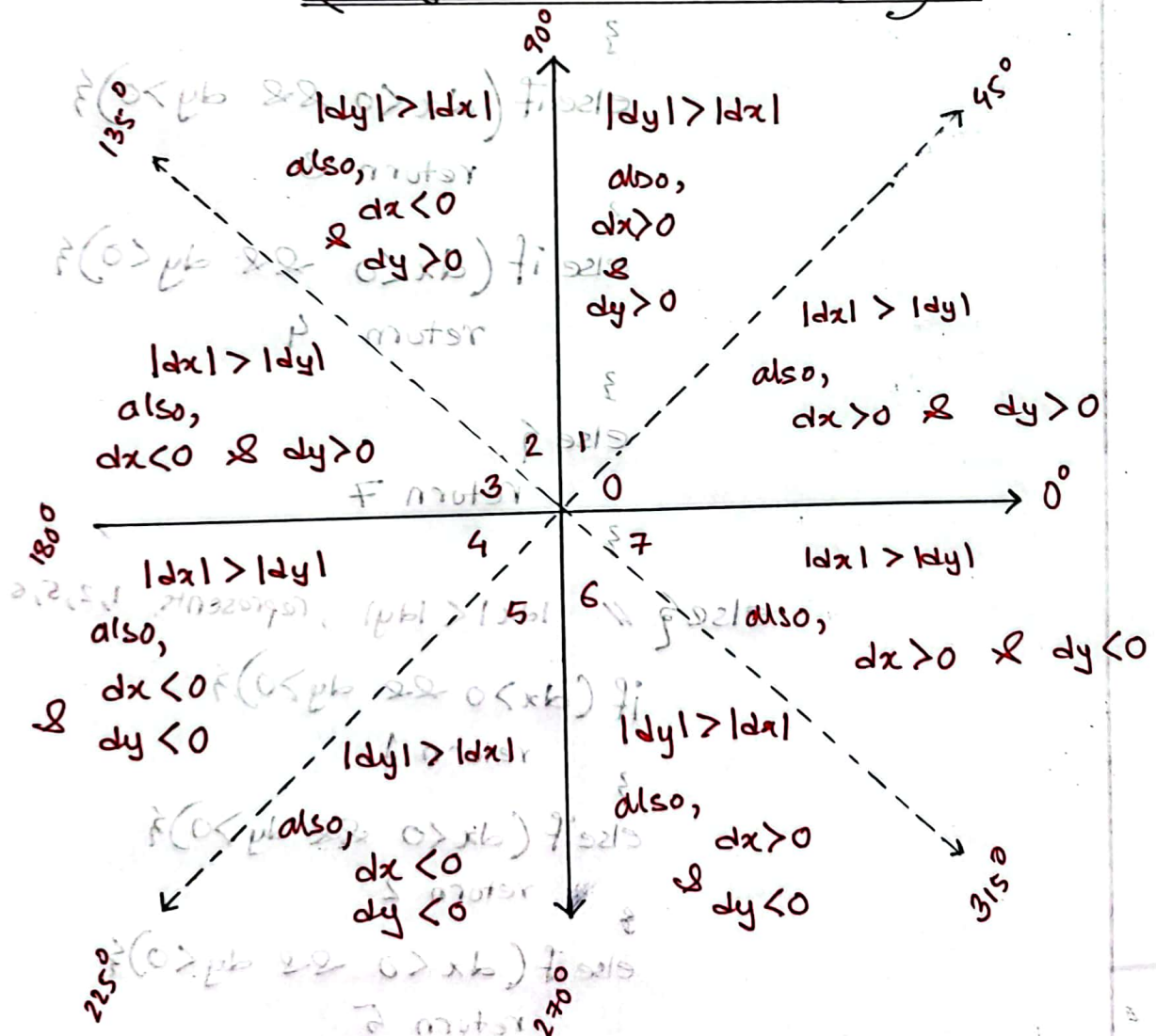
$$dy = +ve$$

But,

$$\text{zone 2} \rightarrow |dy| > |dx|$$

$$\text{zone 3} \rightarrow |dx| > |dy|$$

(Putting it all into Perspective)



Pseudo Code :

```
int find-zone (int x0, int y0, int x1, int y1) {  
    int dy = y1 - y0  
    int dx = x1 - x0  
    if (|dx| > |dy|) { // represents zone 0, 3, 4, 7  
        if (dx > 0 && dy > 0) {  
            return 0  
        }  
        else if (dx < 0 && dy > 0) {  
            return 3  
        }  
        else if (dx < 0 && dy < 0) {  
            return 4  
        }  
        else {  
            return 7  
        }  
    }  
    else { // |dx| < |dy|, represents 1, 2, 5, 6  
        if (dx > 0 && dy > 0) {  
            return 1  
        }  
        else if (dx < 0 && dy > 0) {  
            return 2  
        }  
        else if (dx < 0 && dy < 0) {  
            return 5  
        }  
        else {  
            return 6  
        }  
    }  
}
```