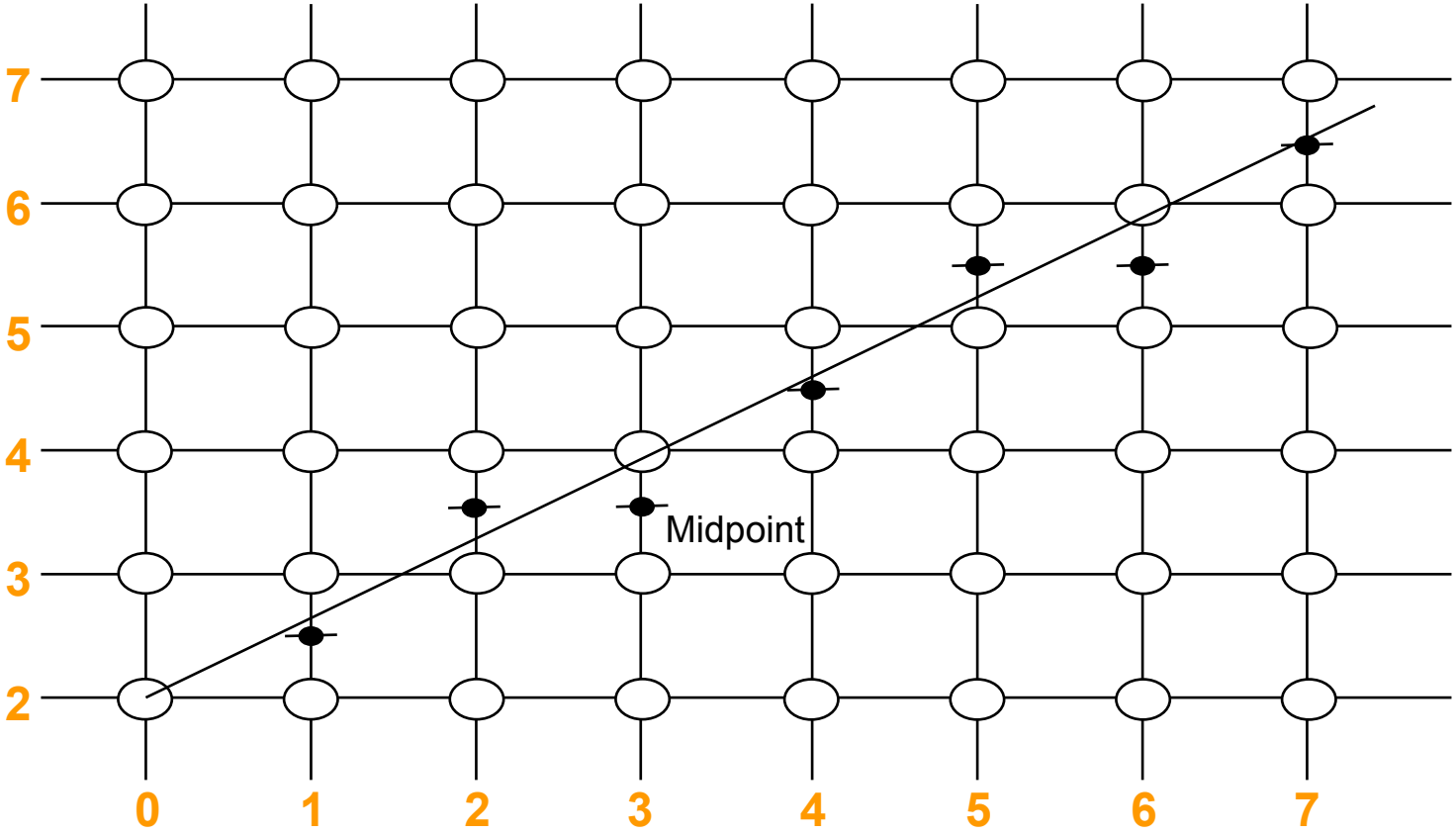
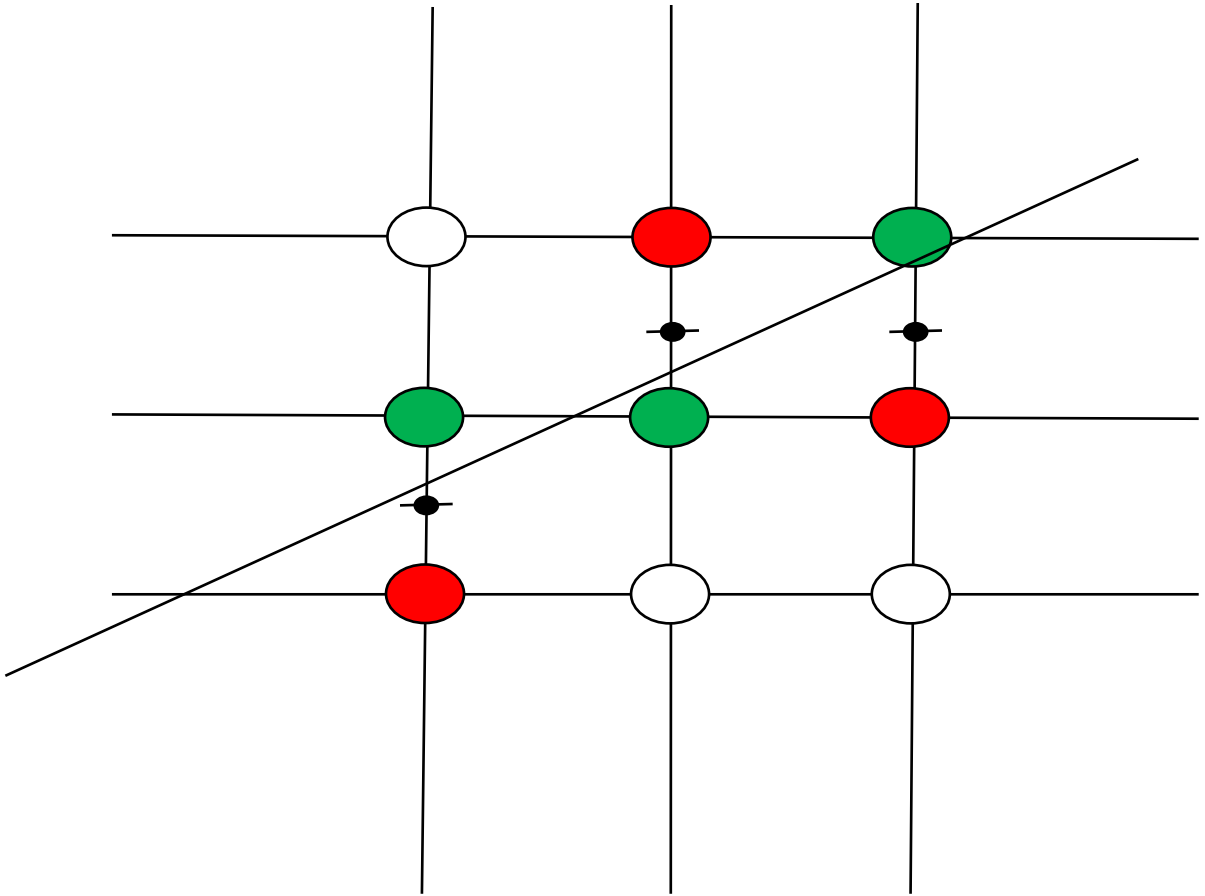
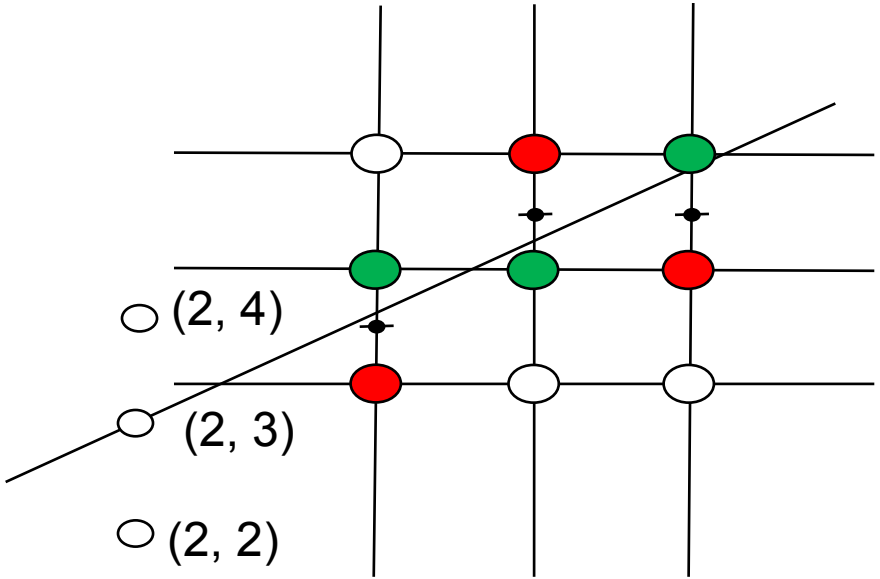
Computer Graphics: Line Drawing Algorithms

Scan Conversion Algorithms

(Midpoint Line)


7
6
5
4
3
2
0
1
2
3
4
5
6
7
Midpoint



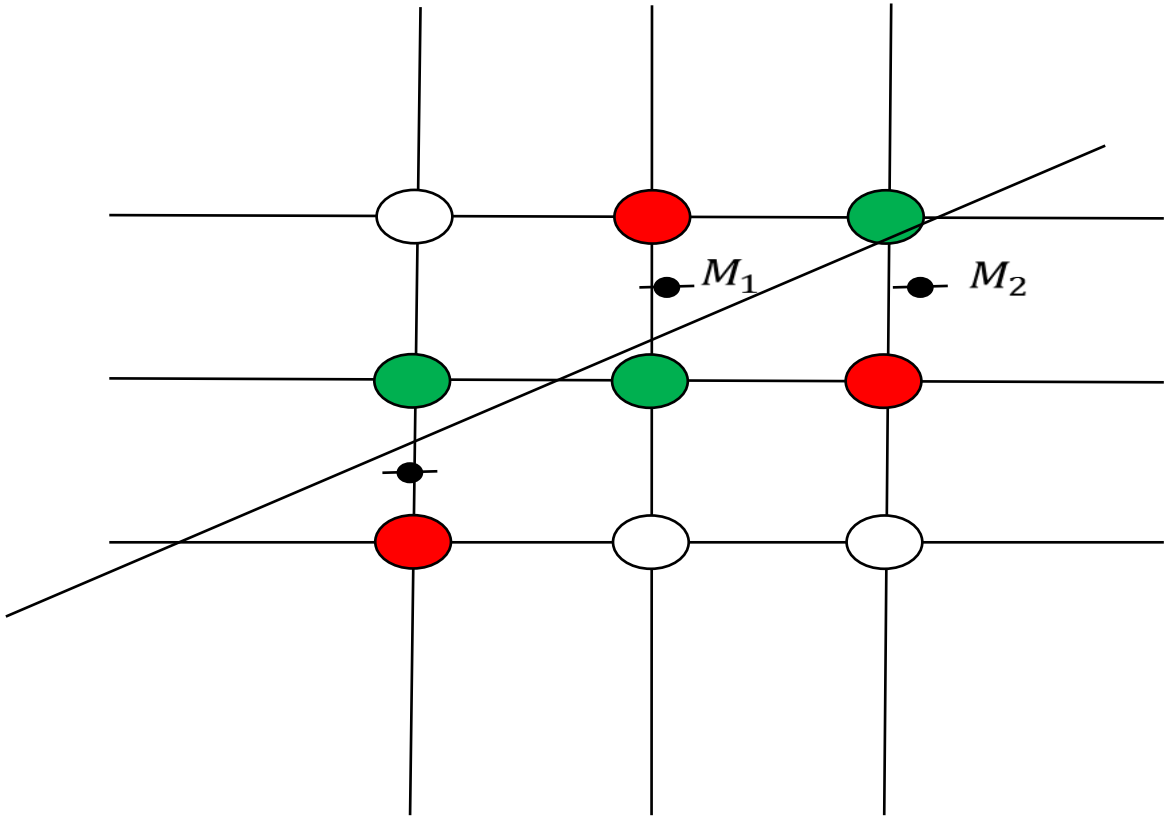


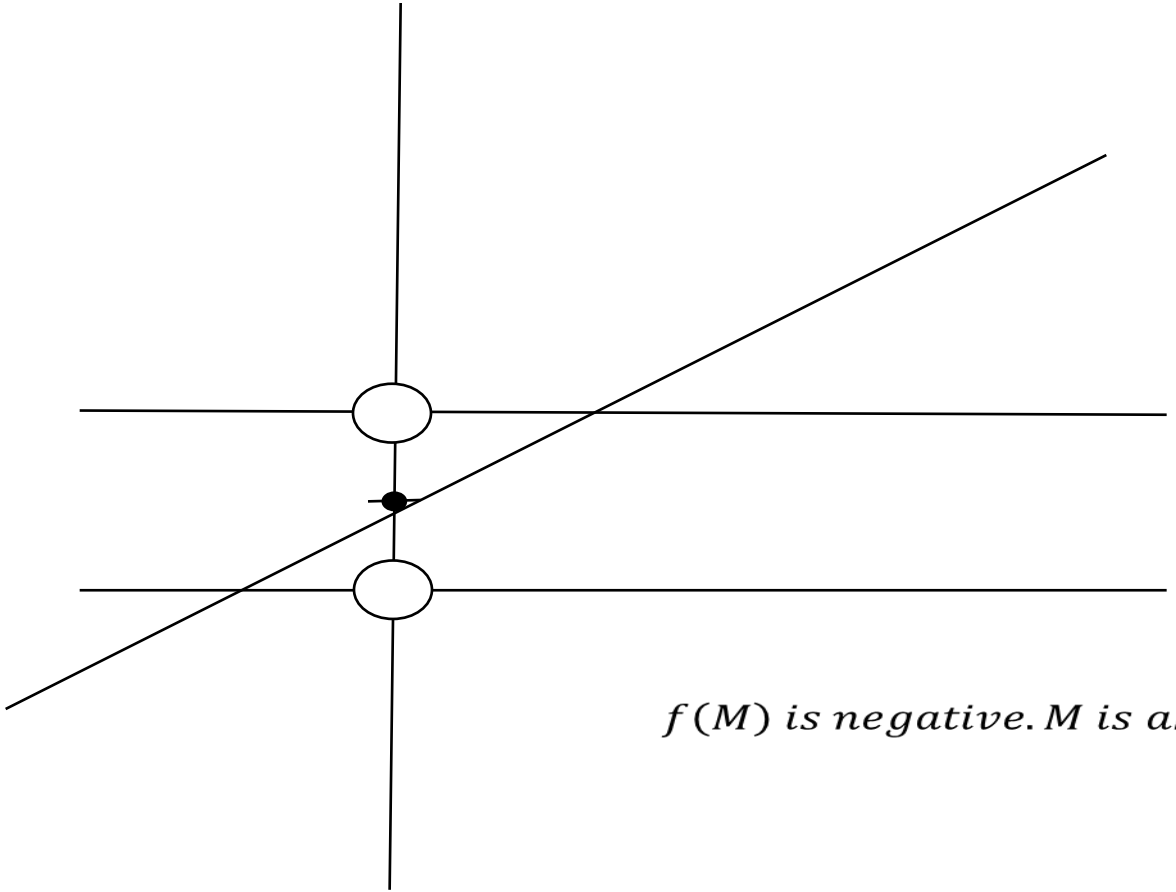
$$ax + by + c = 0$$

$$f(x, y) = ax + by + c$$

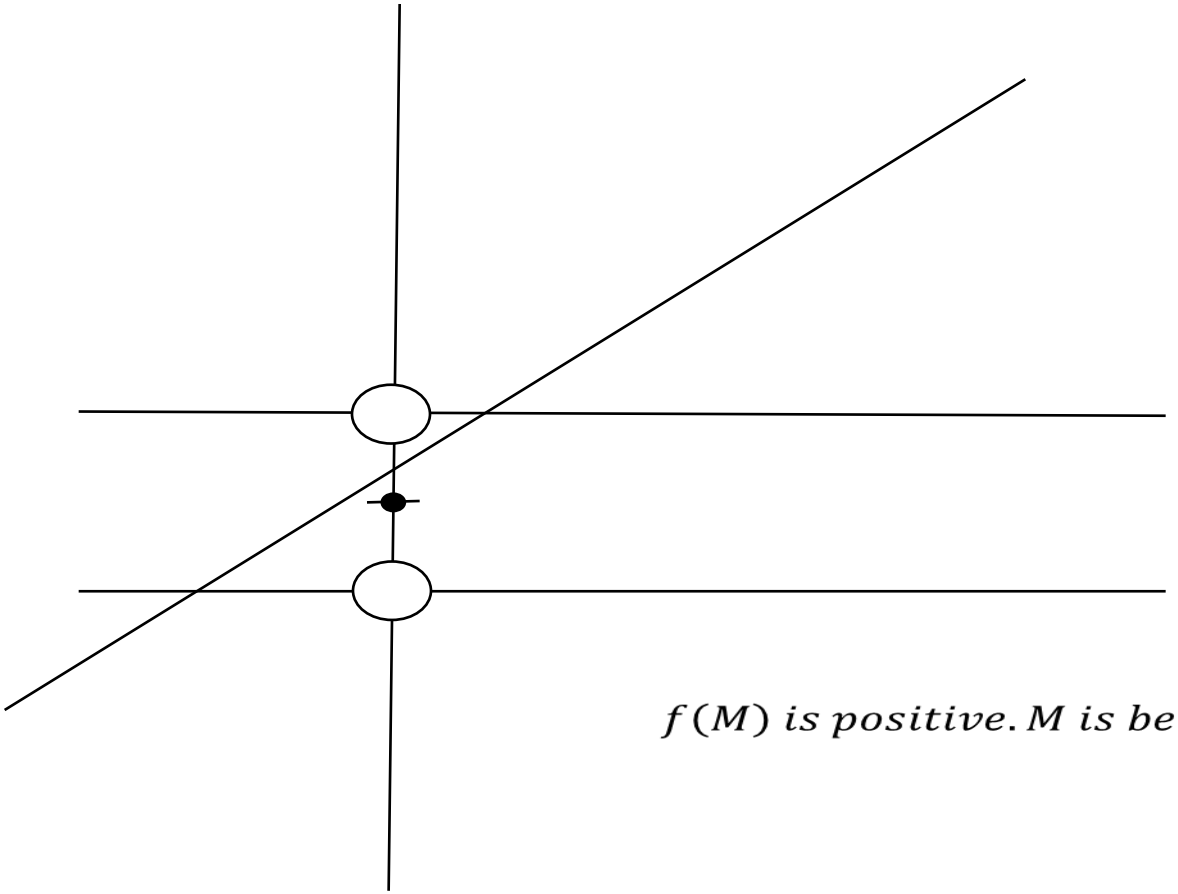
$$f(2, 4) = (-)ve$$

$$f(2, 2) = (+)ve$$


M_1
M_2

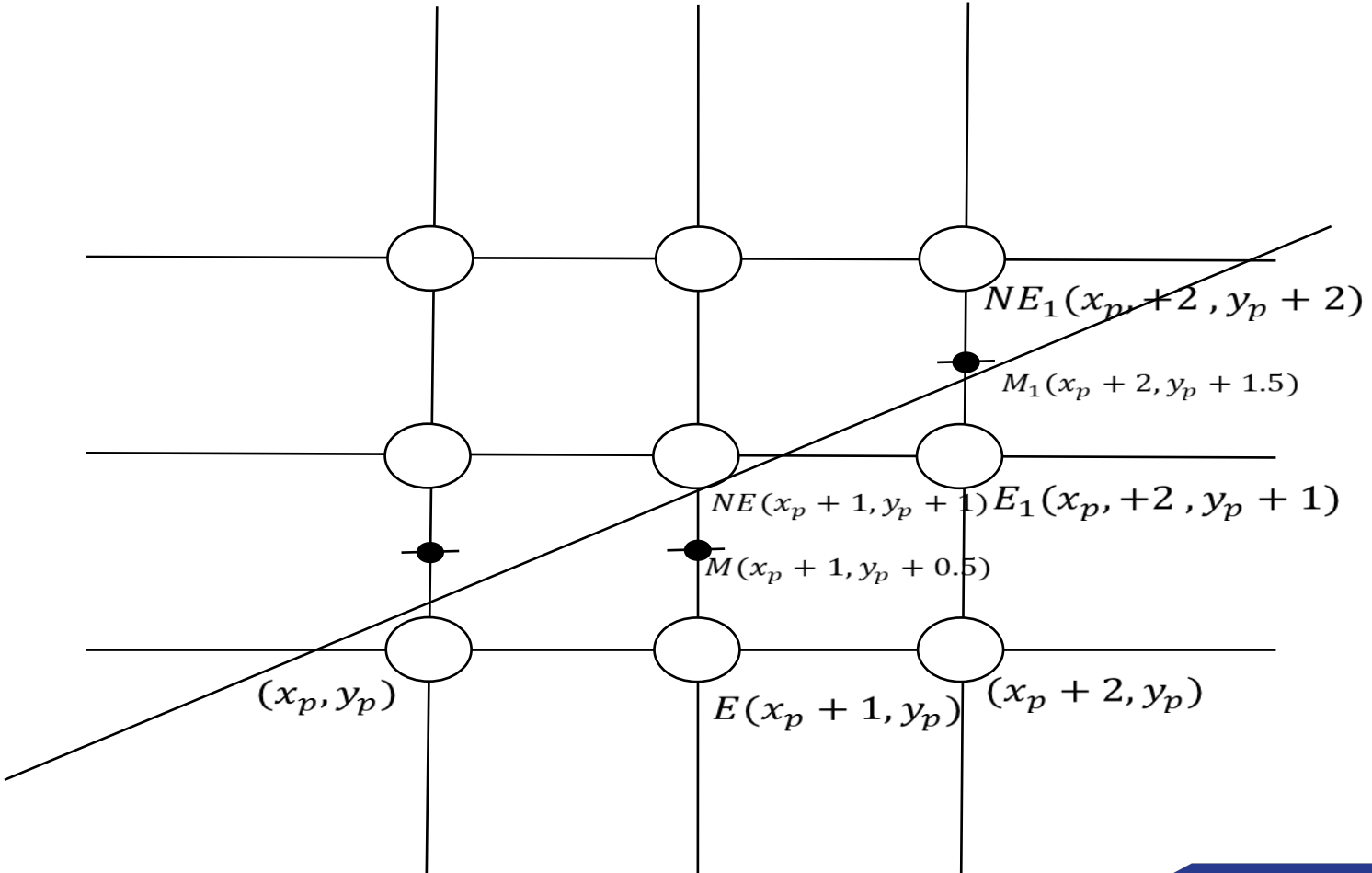


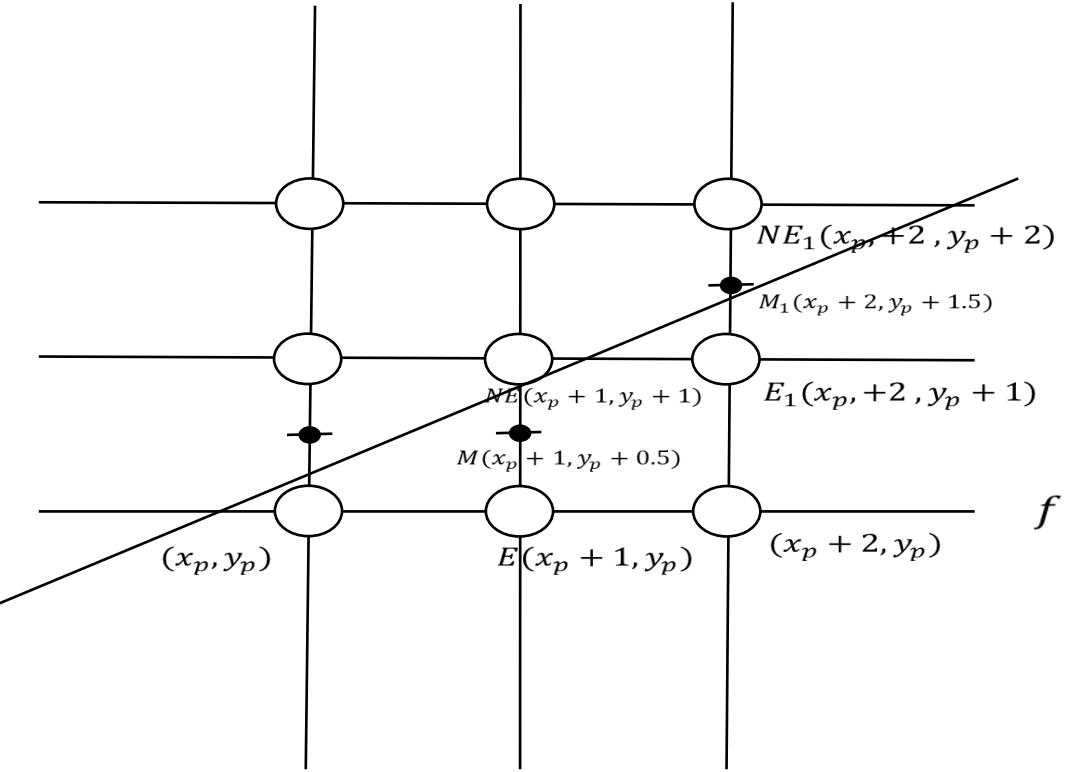
$f(M)$ *is negative.* M *is above the line*



$f(M)$ *is positive.* M *is below the line*

f(M)	Pixel chosen
f(M) > 0	upper (NE)
f(M) ≤ 0	lower (E)


$NE_1(x_p, +2, y_p + 2)$
$M_1(x_p + 2, y_p + 1.5)$
$NE(x_p + 1, y_p + 1)$
$E_1(x_p, +2, y_p + 1)$
$M(x_p + 1, y_p + 0.5)$
(x_p, y_p)
$E(x_p + 1, y_p)$
$(x_p + 2, y_p)$

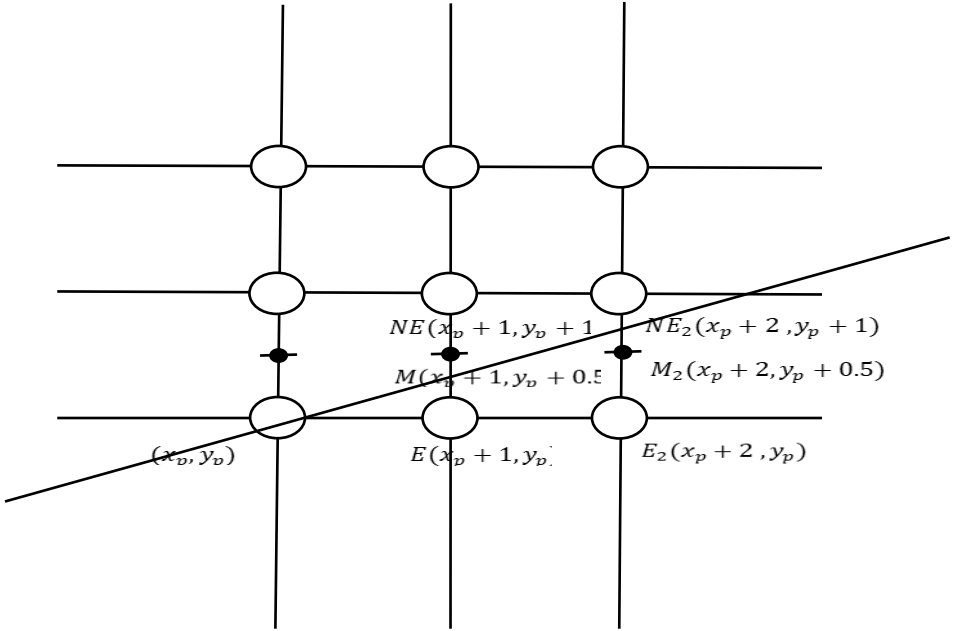


$$f(M) = f(x_p + 1, y_p + 0.5)$$
$$= a(x_p + 1) + b(y_p + 0.5) + c$$
$$d = ax_p + by_p + c + a + 0.5b$$

Suppose, $d > 0$, *so NE is chosen*

$$f(M_1) = f(x_p + 2, y_p + 1.5)$$
$$= a(x_p + 2) + b(y_p + 1.5) + c$$
$$d_{new} = ax_p + by_p + c + 2a + 1.5b$$

$$d_{new} - d = a + b$$
$$d_{new} = d + a + b$$



$$f(M) = f(x_p + 1, y_p + 0.5)$$
$$= a(x_p + 1) + b(y_p + 0.5) + c$$
$$d = ax_p + by_p + c + a + 0.5b$$

Suppose, $d \leq 0, so\ E\ is\ chosen$

$$f(M_2) = f(x_p + 2, y_p + 0.5)$$
$$= a(x_p + 2) + b(y_p + 0.5) + c$$
$$d_{new} = ax_p + by_p + c + 2a + 0.5b$$

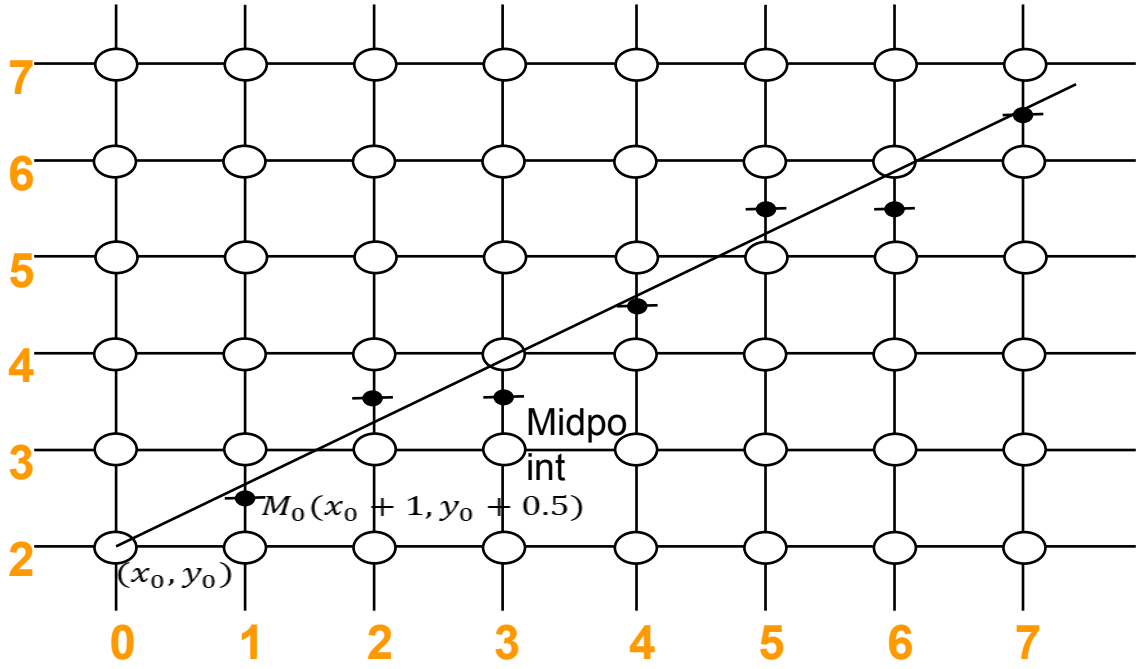
$$d_{new} - d = a$$
$$d_{new} = d + a$$

Calculate d for 1st column.

Choose E/NE.

Update d_{new} acc. to E/NE.

Use d_{new} to choose E/NE again and repeat the loop until the end.

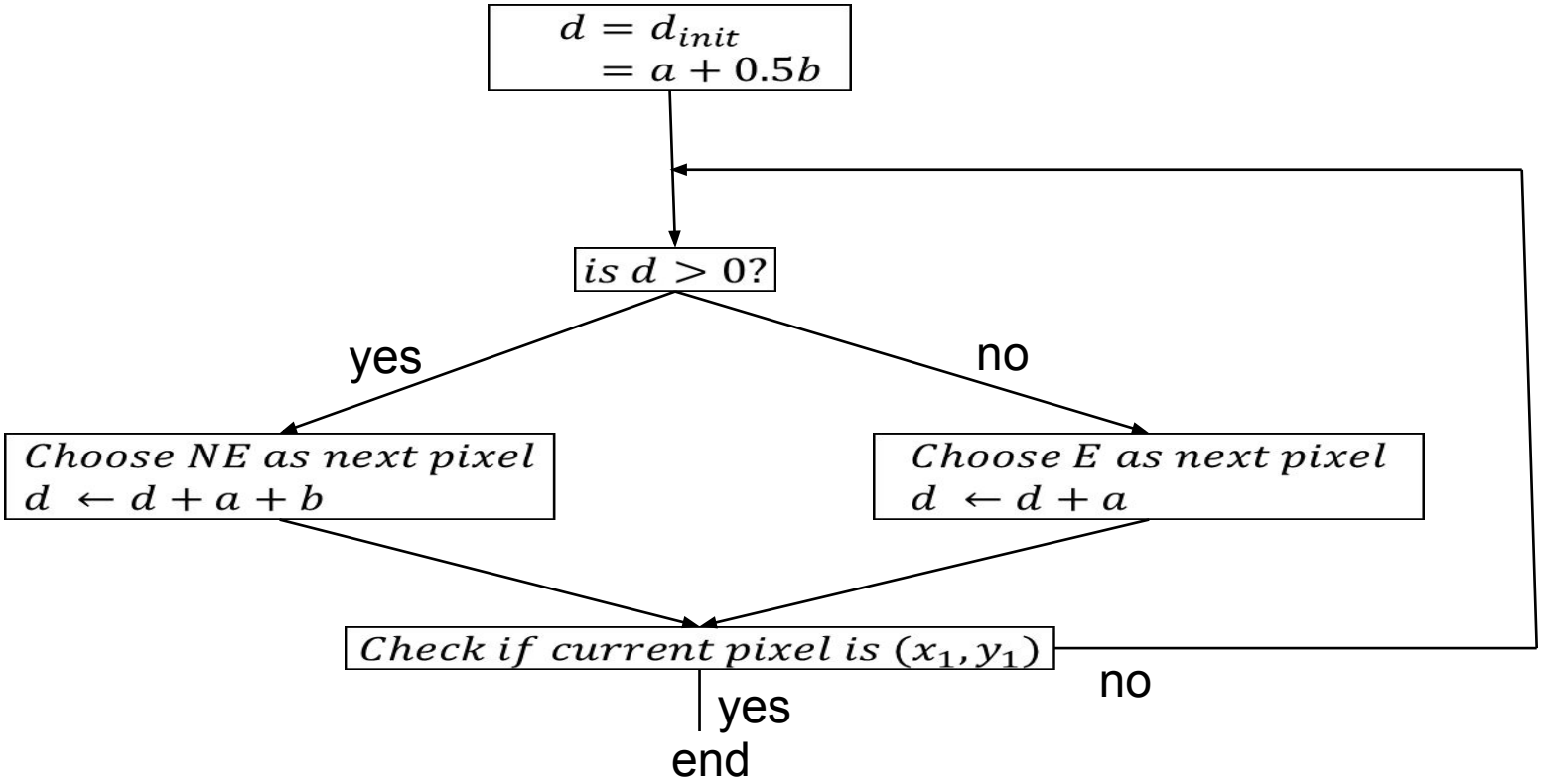


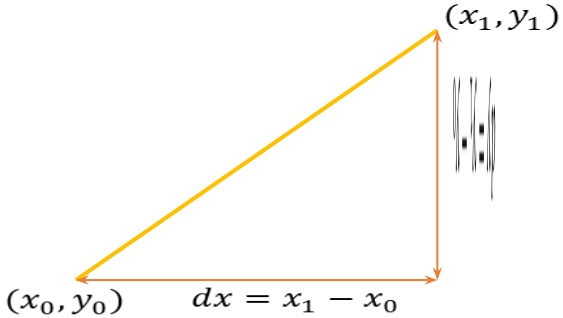
$$
\begin{aligned}
d_{init} &= f(M_0) \\
&= f(x_0 + 1, y_0 + 0.5) \\
&= a(x_0 + 1) + b(y_0 + 0.5) + c \\
&= ax_0 + a + by_0 + 0.5b + c \\
d_{init} &= ax_0 + by_0 + c + a + 0.5b
\end{aligned}
$$

$$ax + by + c = 0$$
$$true\ for\ all\ (x, y)\ on\ the\ line$$

$$So, ax_0 + by_0 + c = 0$$

$$
\begin{aligned}
d_{init} &= ax_0 + by_0 + c + a + 0.5b \\
&= 0 + a + 0.5b \\
&= a + 0.5b
\end{aligned}
$$


$d = d_{init}$
$= a + 0.5b$
$is\ d > 0?$
yes
no
$Choose\ NE\ as\ next\ pixel$
$d \leftarrow d + a + b$
$Choose\ E\ as\ next\ pixel$
$d \leftarrow d + a$
$Check\ if\ current\ pixel\ is\ (x_1, y_1)$
no
yes
end



$$y = mx + B$$

$$m = \frac{dy}{dx} \text{ where } dy = y_1 - y_0 \text{ and } dx = x_1 - x_0$$

$$y = \frac{dy}{dx}.x + B$$

$$y.dx = dy.x + B.dx$$

$$0 = dy.x - y.dx + B.dx$$

$$dy.x - dx.y + B.dx = 0$$

Comparing this with,

$$ax + by + c = 0$$

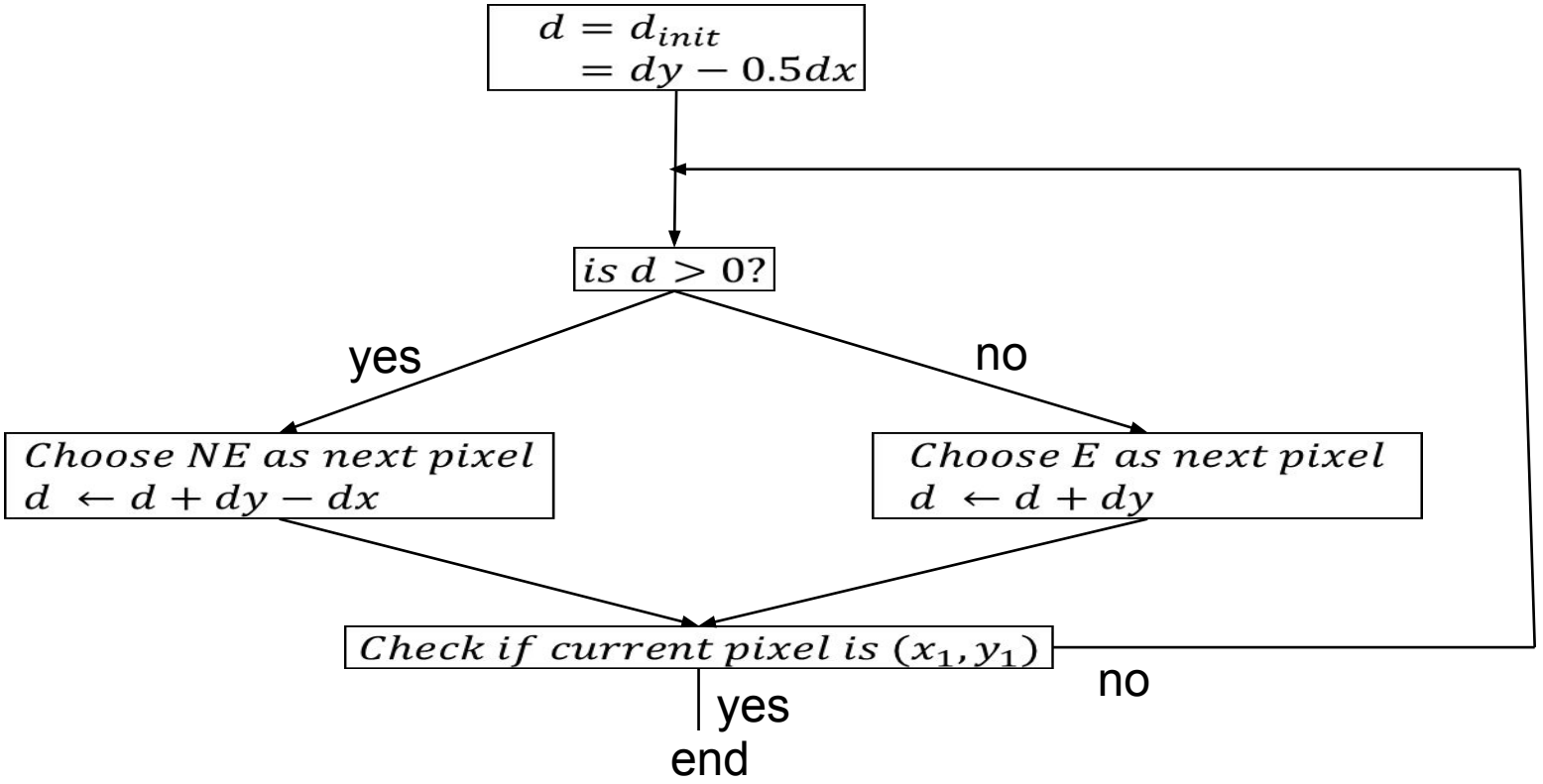
We get,

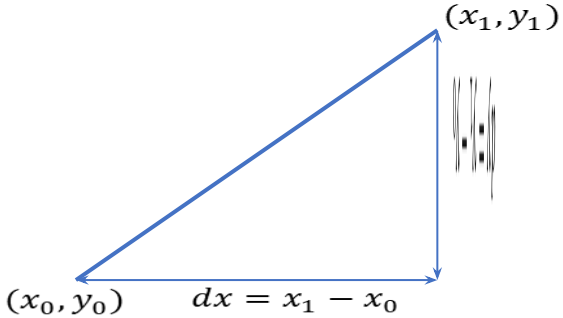
$$a = dy$$

$$b = -dx$$

$$c = B.dx$$

$Putting,$
$a = dy$
$b = -dx$





$$y = mx + b$$

$$m = \frac{dy}{dx} \text{ where } dy = y_1 - y_0 \text{ and } dx = x_1 - x_0$$

$$y = \frac{dy}{dx}.x + b$$

$$y.dx = dy.x + b.dx$$

$$0 = dy.x - y.dx + b.dx$$

$$\boldsymbol{dy.x - dx.y + b.dx = 0}$$

Comparing this with,

$$\boldsymbol{ax + by + c = 0}$$

We get,

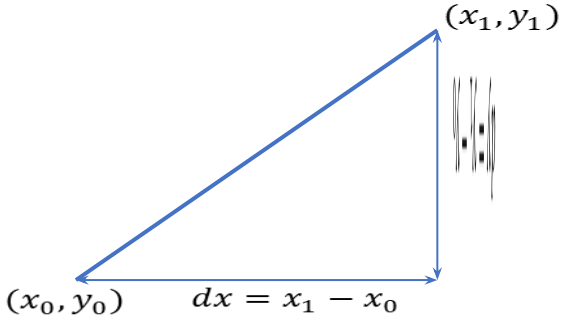
$$a = dy$$

$$b = -dx$$

$$c = b.dx$$

$$2x + 3y + 1 = 0$$

$$4x + 6y + 2 = 0$$



$$y = mx + b$$

$$m = \frac{dy}{dx} \text{ where } dy = y_1 - y_0 \text{ and } dx = x_1 - x_0$$

$$y = \frac{dy}{dx}.x + b$$

$$y.dx = dy.x + b.dx$$

$$0 = dy.x - y.dx + b.dx$$

$$\mathbf{2dy.x - 2dx.y + 2b.dx = 0}$$

Comparing this with,

$$\mathbf{ax + by + c = 0}$$

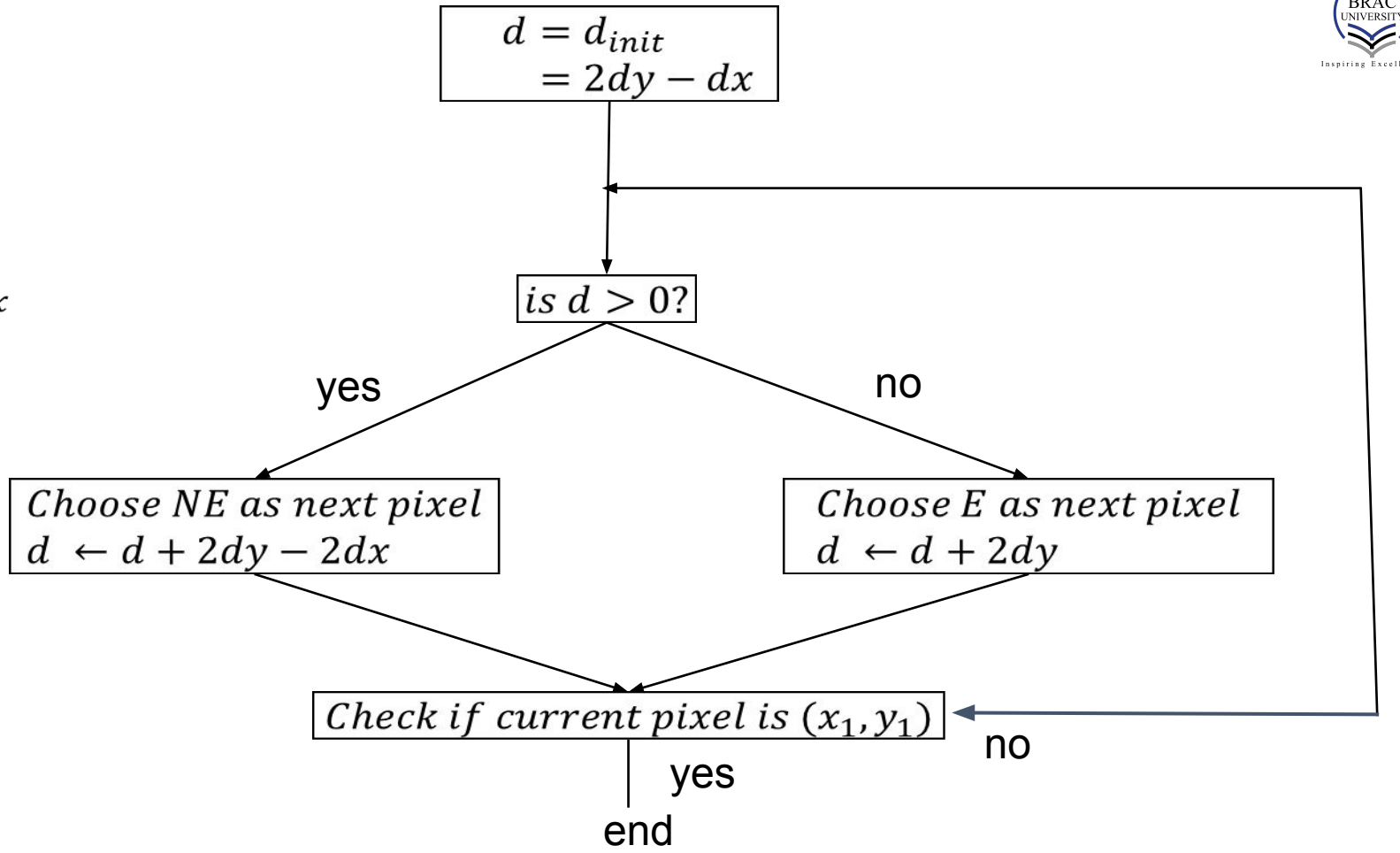
We get,

$$a = 2dy$$

$$b = -2dx$$

$$c = 2b.dx$$

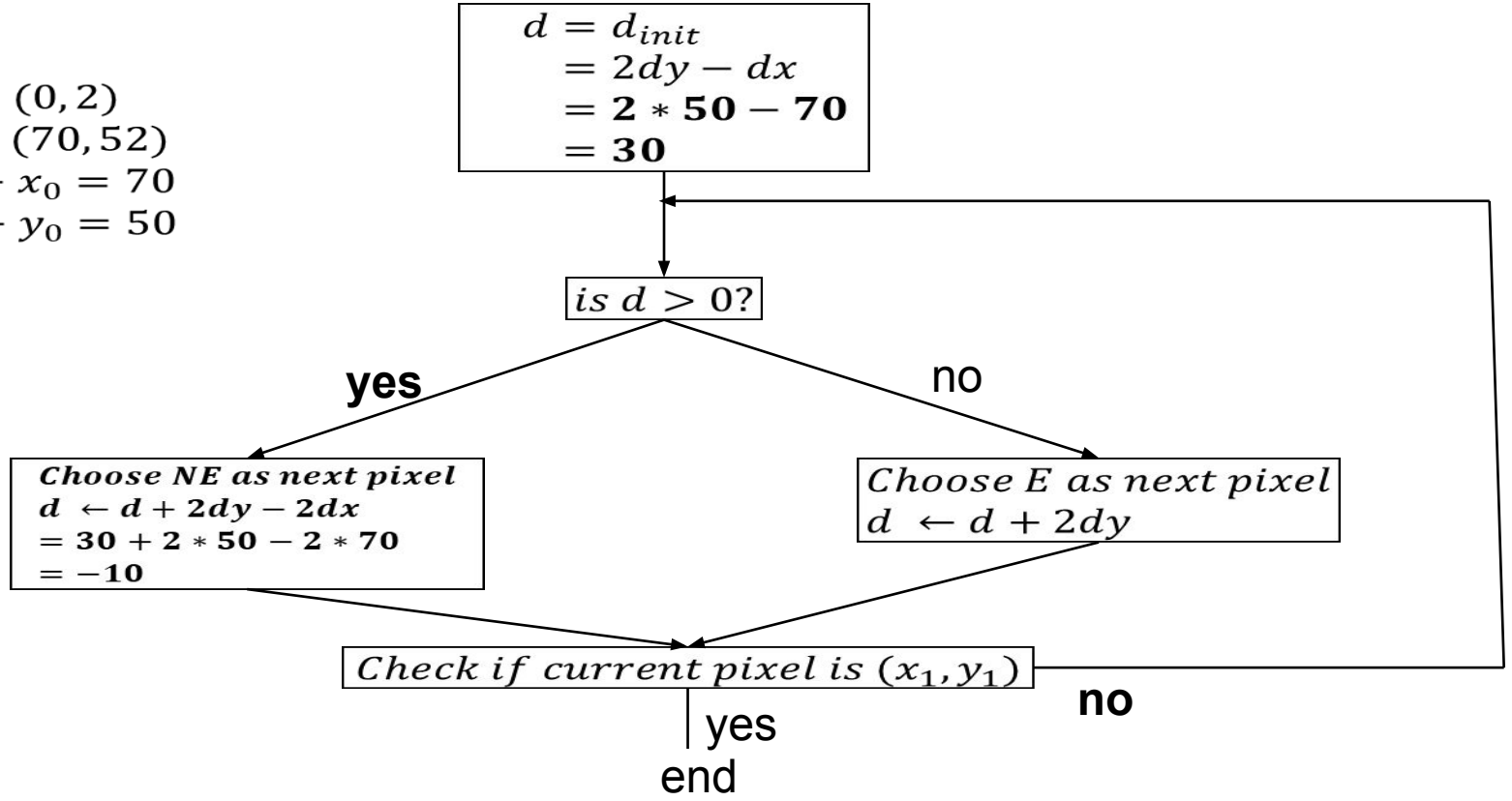
$Putting,$
$a = 2dy$
$b = -2dx$



```
func MidpointLine(int x0, int y0, int x1, int y1, int value){
        int dx, dy, incrE, incrNE, d, x, y;
        dx = x1 -x0;
        dy = y1 -y0;
        d = 2 * dy - dx;
        incrE = 2 * dy;
        incrNE = 2 * (dy - dx);
        x = x0;
        y = y0;
        WritePixel (x, y, value);
        while (x < x1) {
                if (d <= 0) {
                        //choose E
                        d = d + incrE;
                        x = x + 1;
                }
                else {
                        //choose NE
                        d = d + incrNE;
                        x = x + 1;
                        y = y + 1;
                }
                WritePixel (x,y, value) //The selected pixel closest to the line
        }
}
```

Find out the first 7 pixels of the line segment starting from (0, 2) to (70, 52) using midpoint line algorithm

$(x_0, y_0) = (0, 2)$
$(x_1, y_1) = (70, 52)$
$dx = x_1 - x_0 = 70$
$dy = y_1 - y_0 = 50$



x	y	d	NE(+1,+1)/E(+1,0)	d updating	Pixel
0	2	**30**	**NE**	**30+2x50-2x70 = -10**	**(0, 2)**
1	**3**				

$d_{init} = 2dy - dx = 2.50 - 70 = 30$
$\Delta d_{NE} = 2dy - 2dx = 2.50 - 2.70 = -40$
$\Delta d_{E} = 2dy = 2.50 = 100$

x	y	d	NE(+1,+1)/ E(+1,0)	d updating	Pixel
0	2	30	NE	30+Δd_{NE} = -10	(0, 2)
1	3	-10	E	-10+Δd_{E} = 90	(1, 3)
2	3	90	NE	90+Δd_{NE} = 50	(2, 3)
3	4	50	NE	50+Δd_{NE} = 10	(3, 4)
4	5	10	NE	10+Δd_{NE} = -30	(4, 5)
5	6	-30	E	-30+Δd_{E} = 70	(5, 6)
6	6	70	NE	70+Δd_{NE} = 30	(6, 6)
7	7				