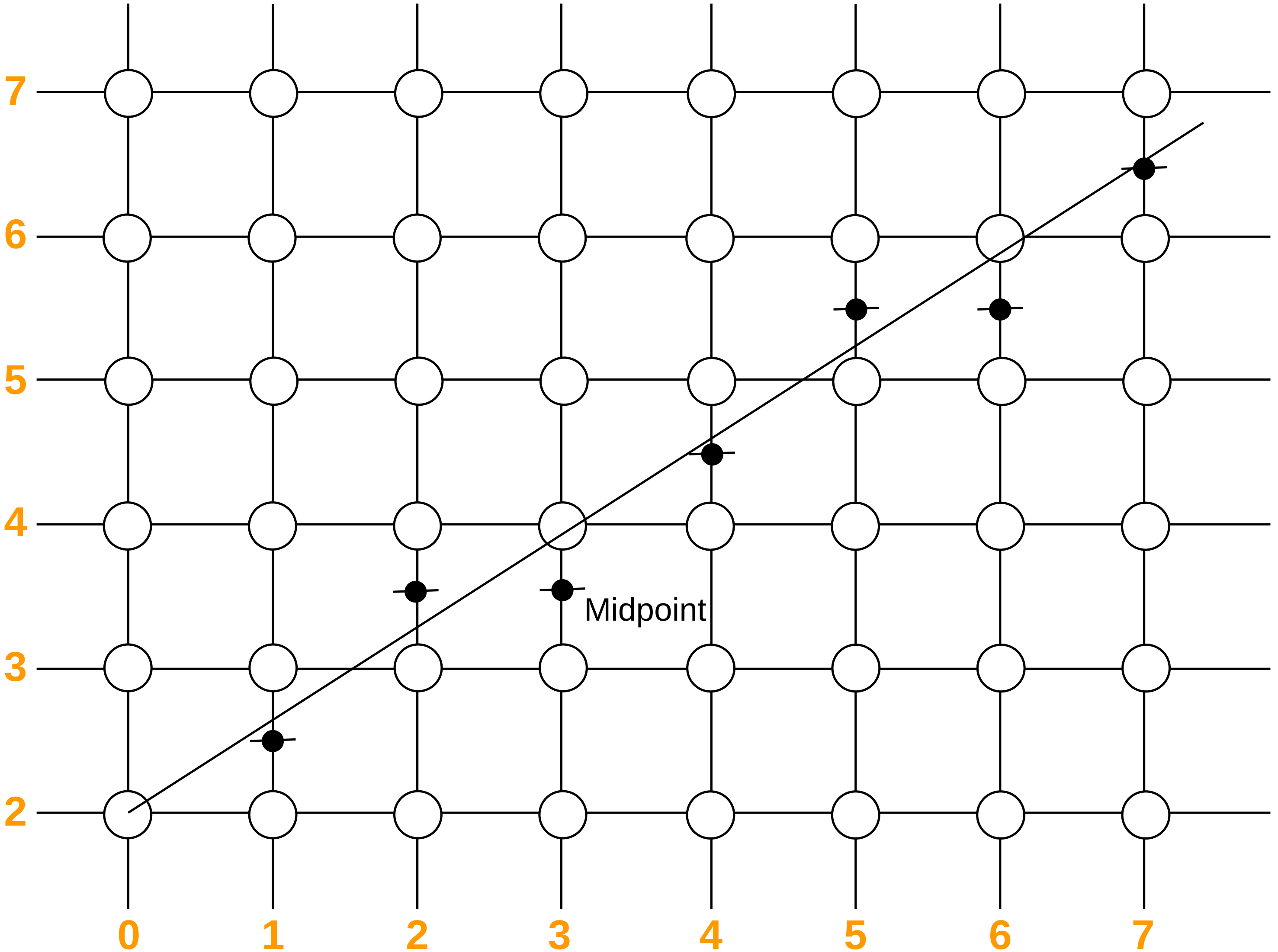
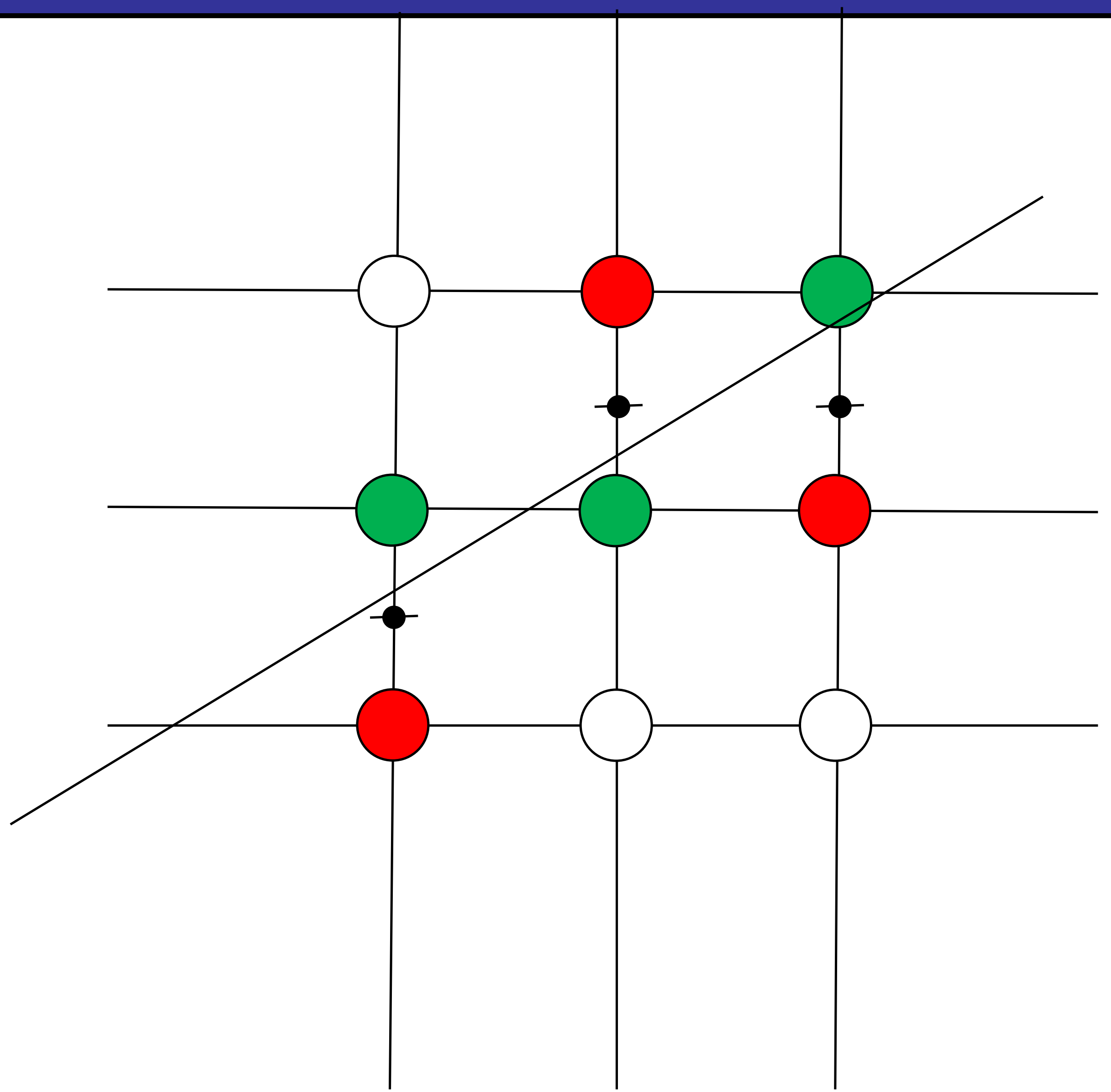
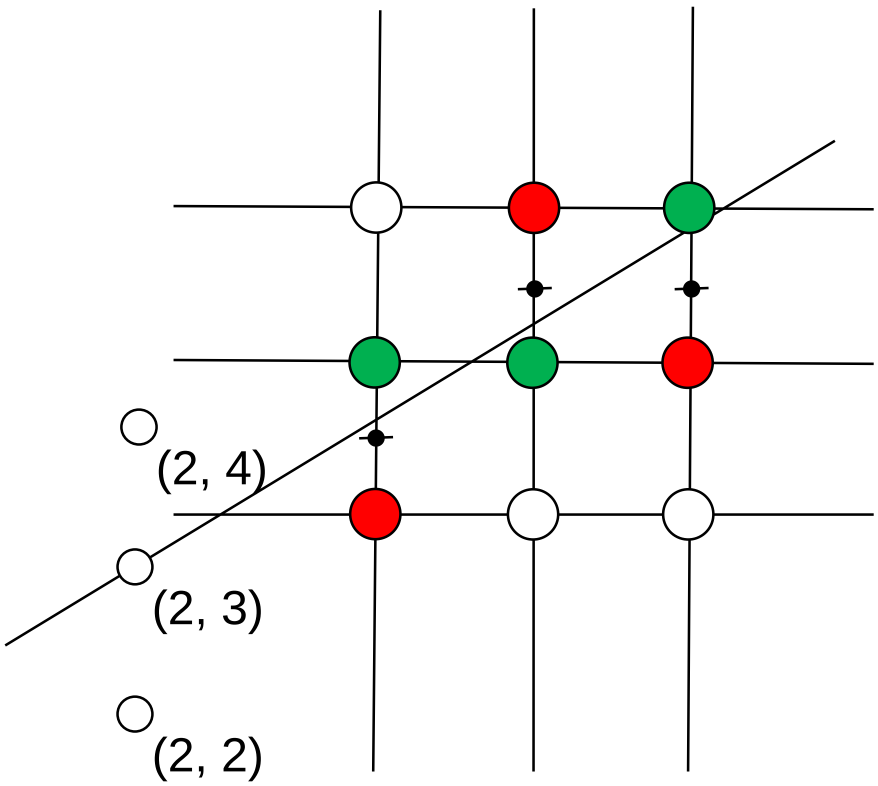
Computer Graphics:
Line Drawing Algorithms

Scan Conversion Algorithms

(Midpoint Line)


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



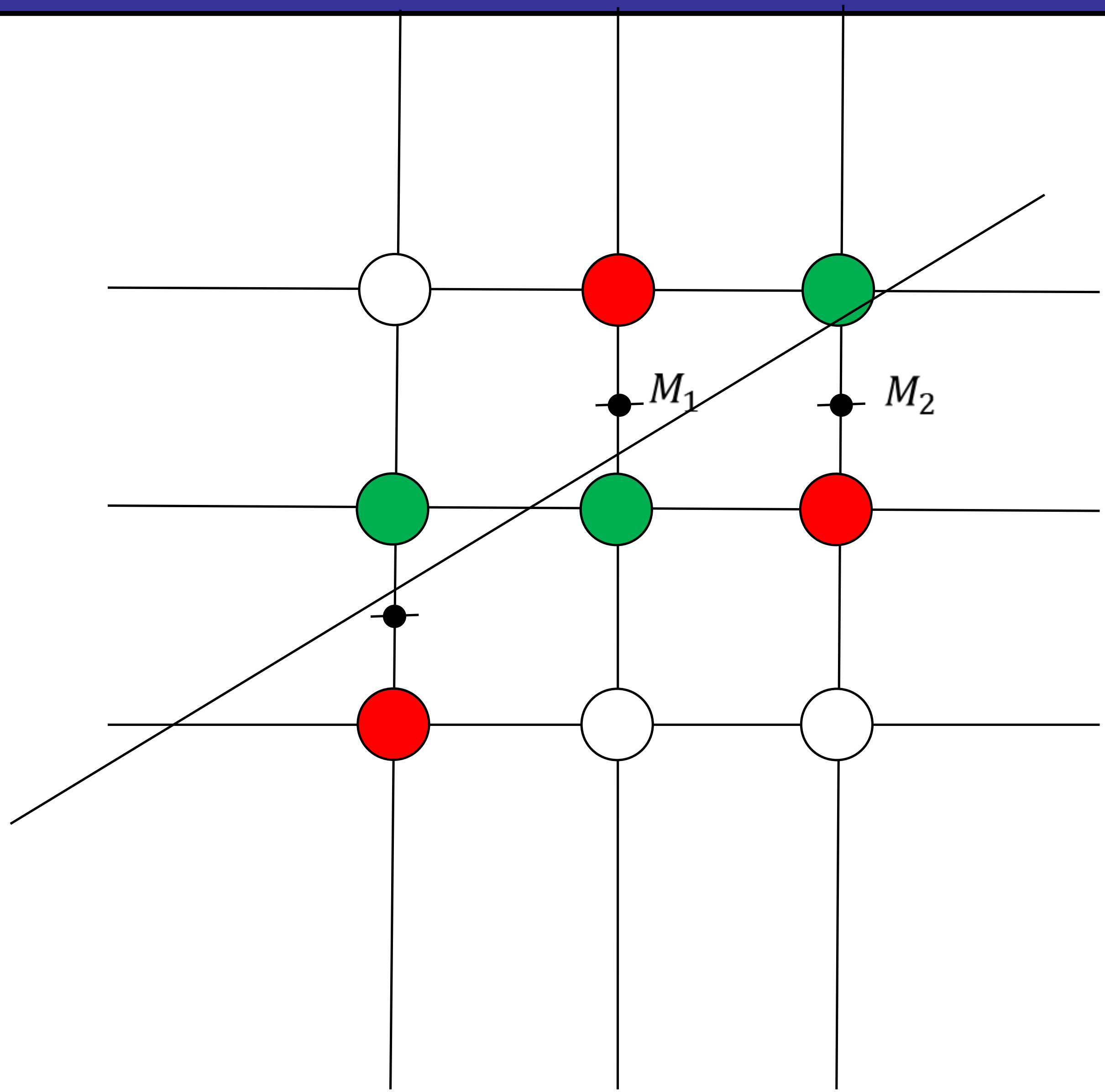


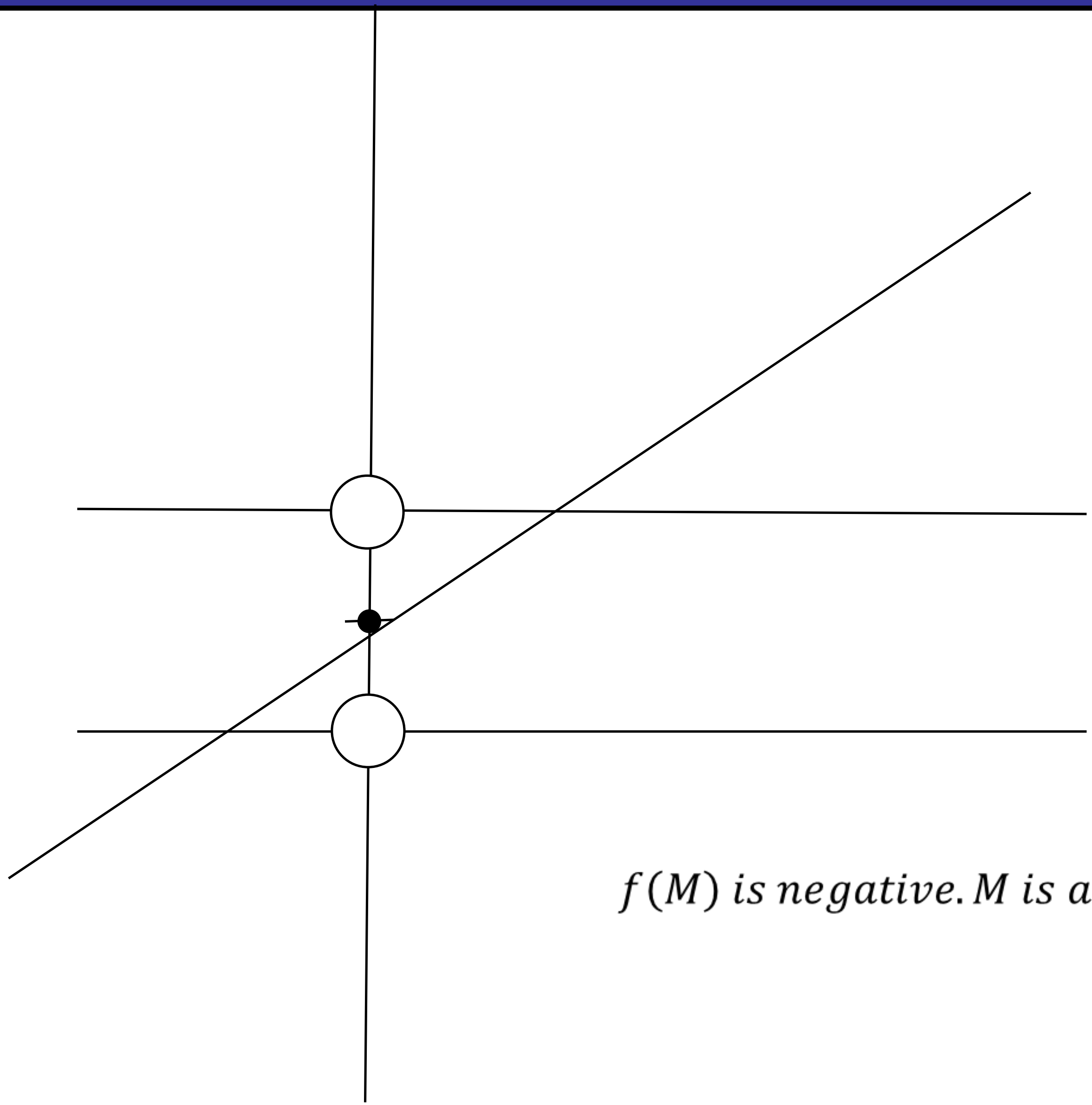
$$ax + by + c = 0$$

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

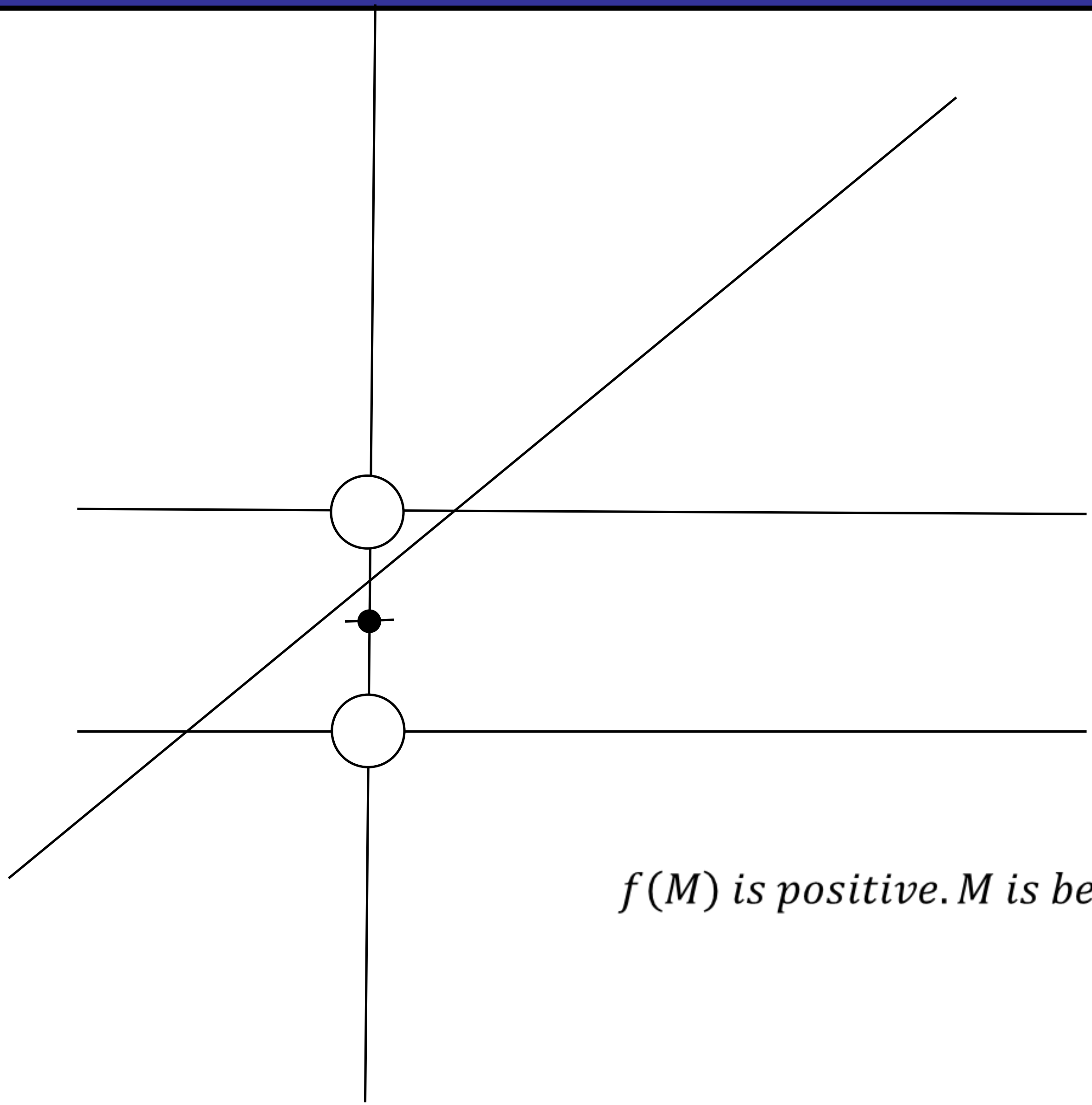
$$f(2, 4) = (-)ve$$

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


M_1
M_2

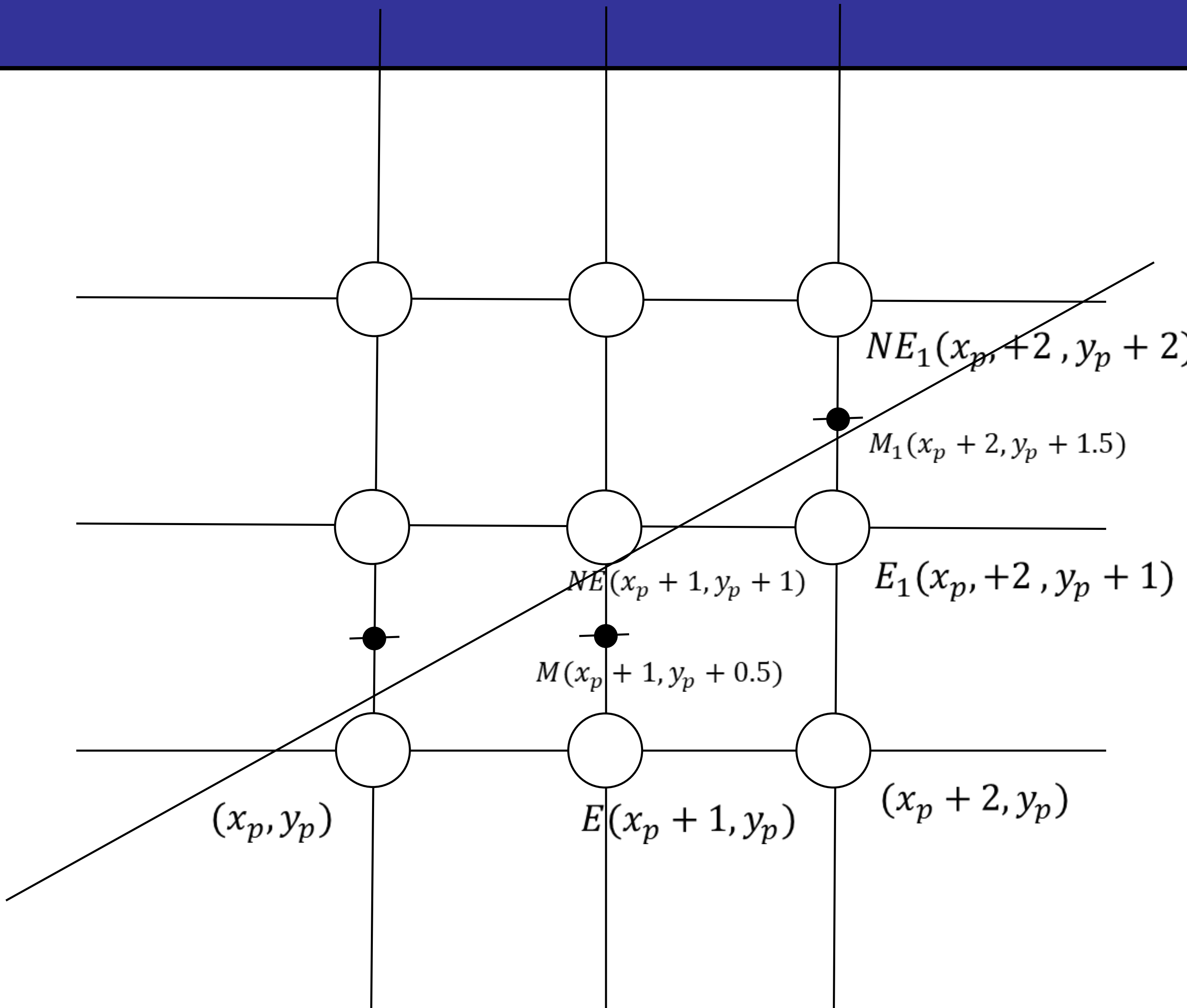


$f(M)$ is $negative.$ M is $above$ the $line$

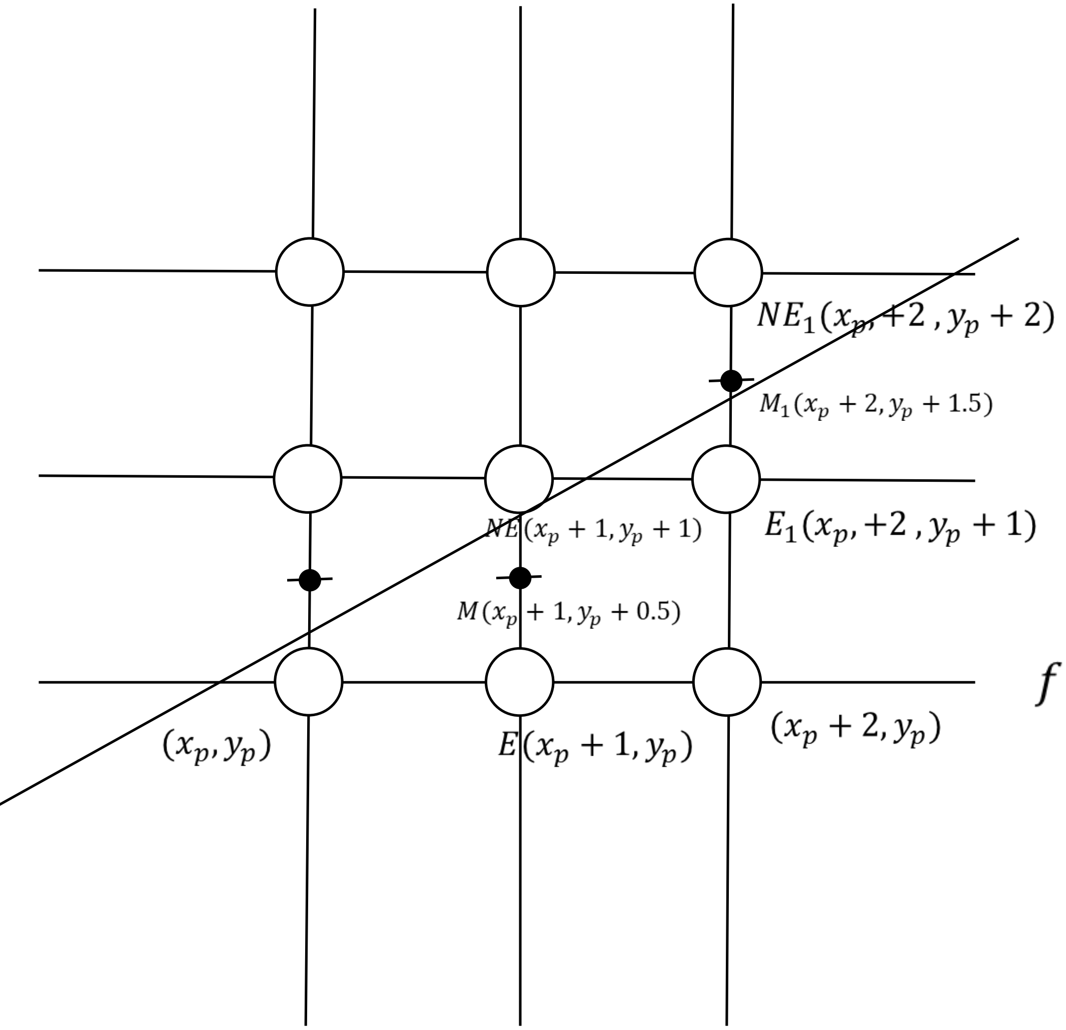


$f(M)$ *is positive.* M *is below the line*

f(M)	Pixel chosen
$f(M) > 0$	upper
$f(M) \leq 0$	lower



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

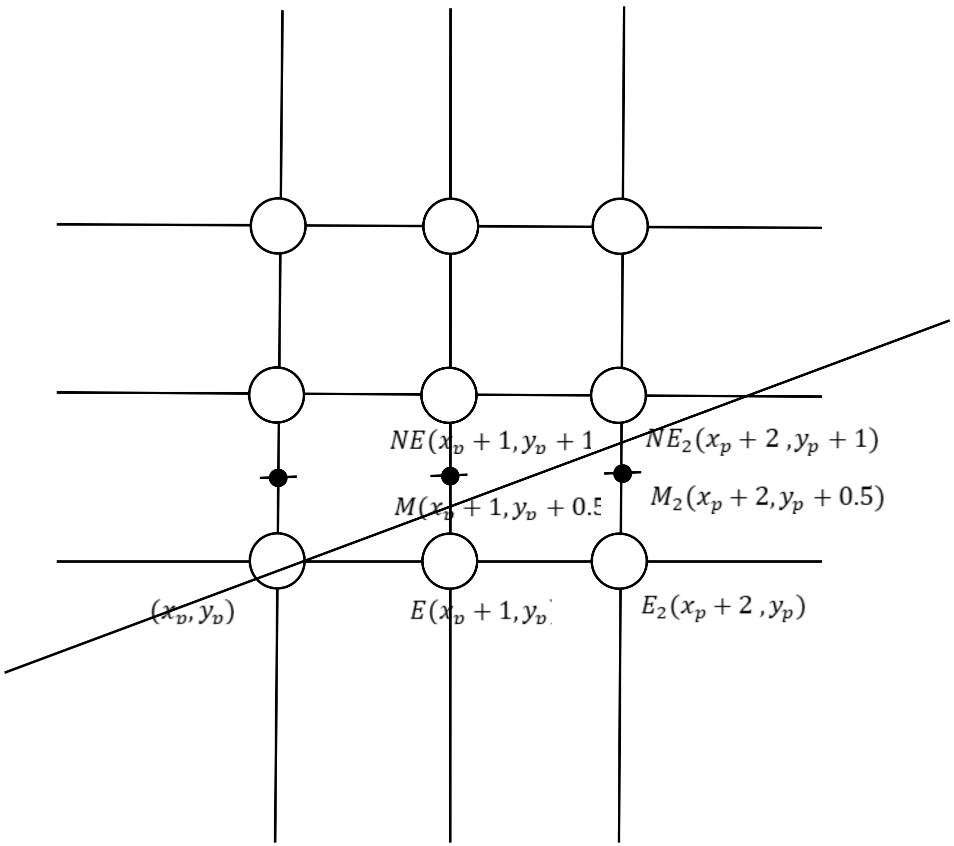


$$f(M) = f(x_p + 1, y_p + 0.5)$$
$$= a(x_p + 1) + b(y_p + 0.5) + c$$
$$d = ax_p + by_p + c + a + 0.5b$$

$$d > 0, so\ NE\ is\ chosen$$

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

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



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

$$d \le 0, so\ E\ is\ chosen$$

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

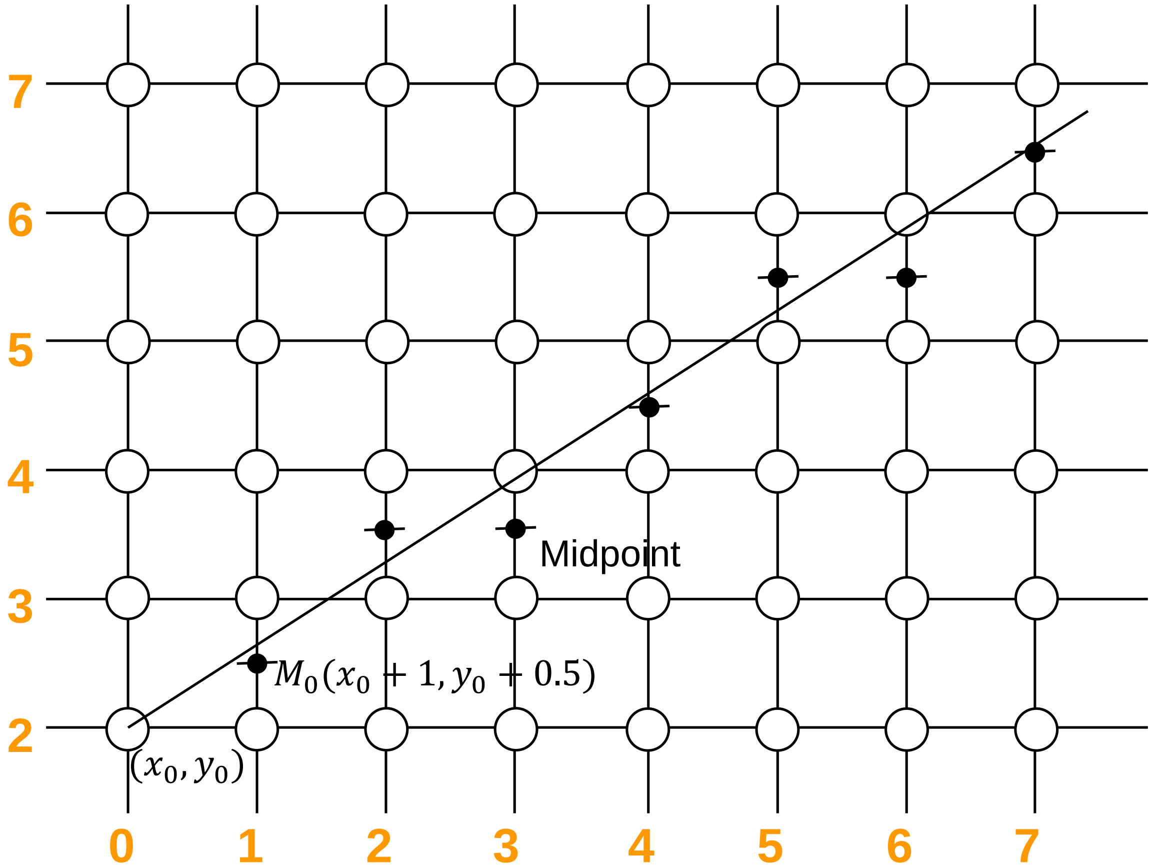
$$d_{new} \ - d = \ a$$
$$d_{new} = d + \ a$$

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.



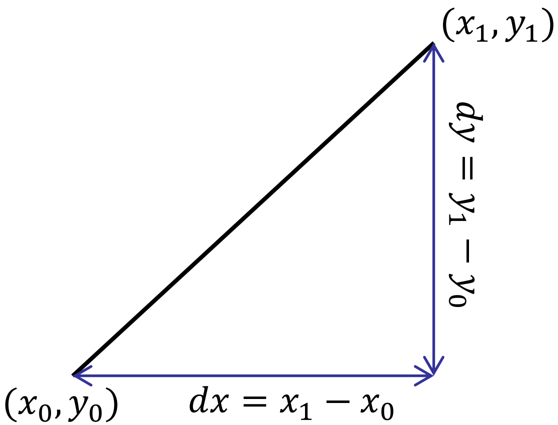
$$
\begin{aligned}
d_{init} &= f(M_0) \\
&= \mathrm{f}(\mathrm{x}_0 + 1, \mathrm{y}_0 + 0.5) \\
&= a(x_0+1) + b(y_0+0.5) + c \\
&= ax_0 + a + by_0 + 0.5b + c \\
d_{init} &= ax_0 + by_0 + c + a + 0.5b
\end{aligned}
$$

$ax + by + c = 0$

$true\ for\ all\ (x, y)\ on\ the\ line$

$So, ax_0 + by_0 + c = 0$

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



$$y = mx + B$$

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

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

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

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

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

Comparing this with,

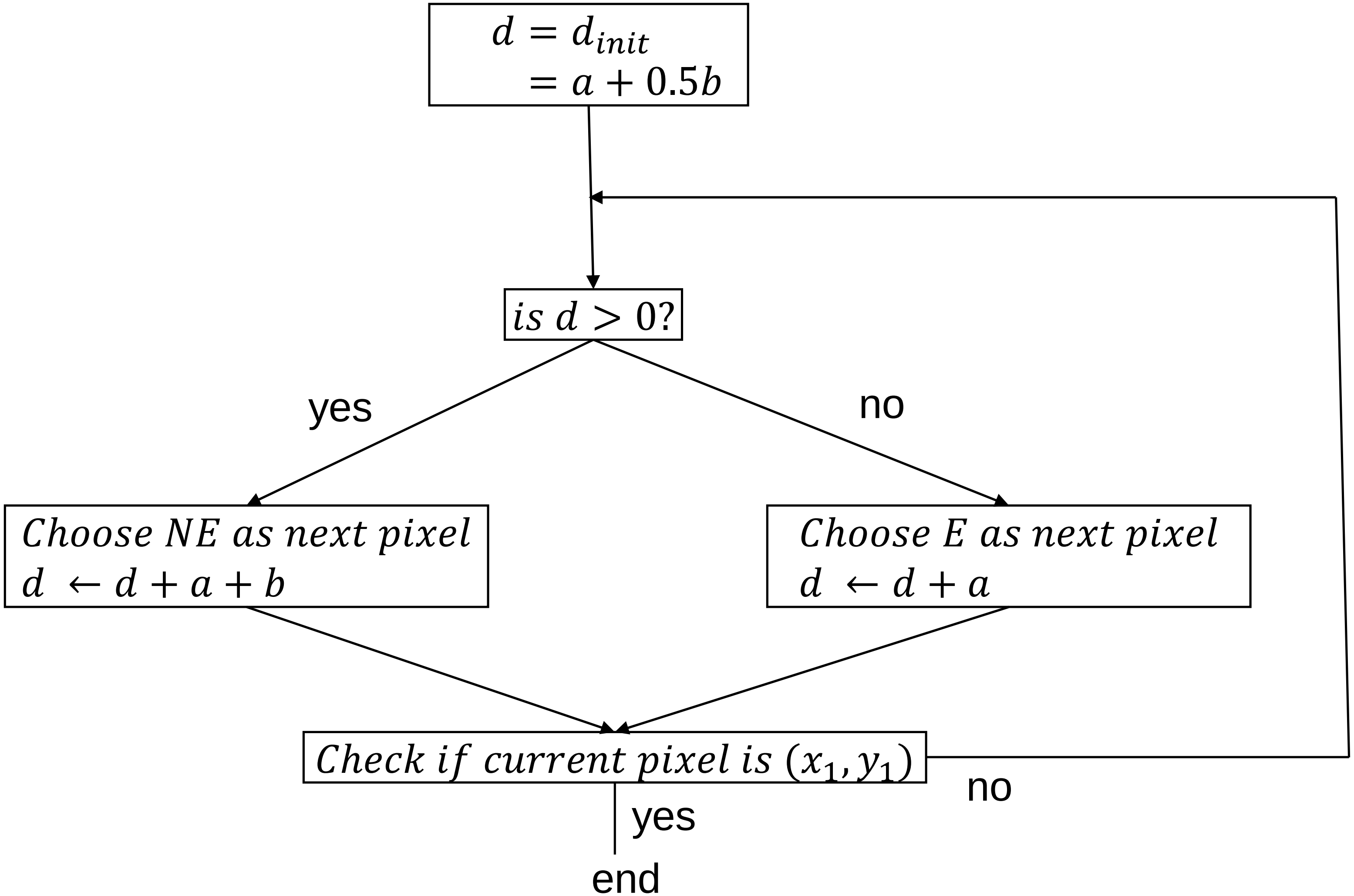
$$ax \quad + by \quad + c \quad = 0$$

We get,

$$a = dy$$

$$b = -dx$$

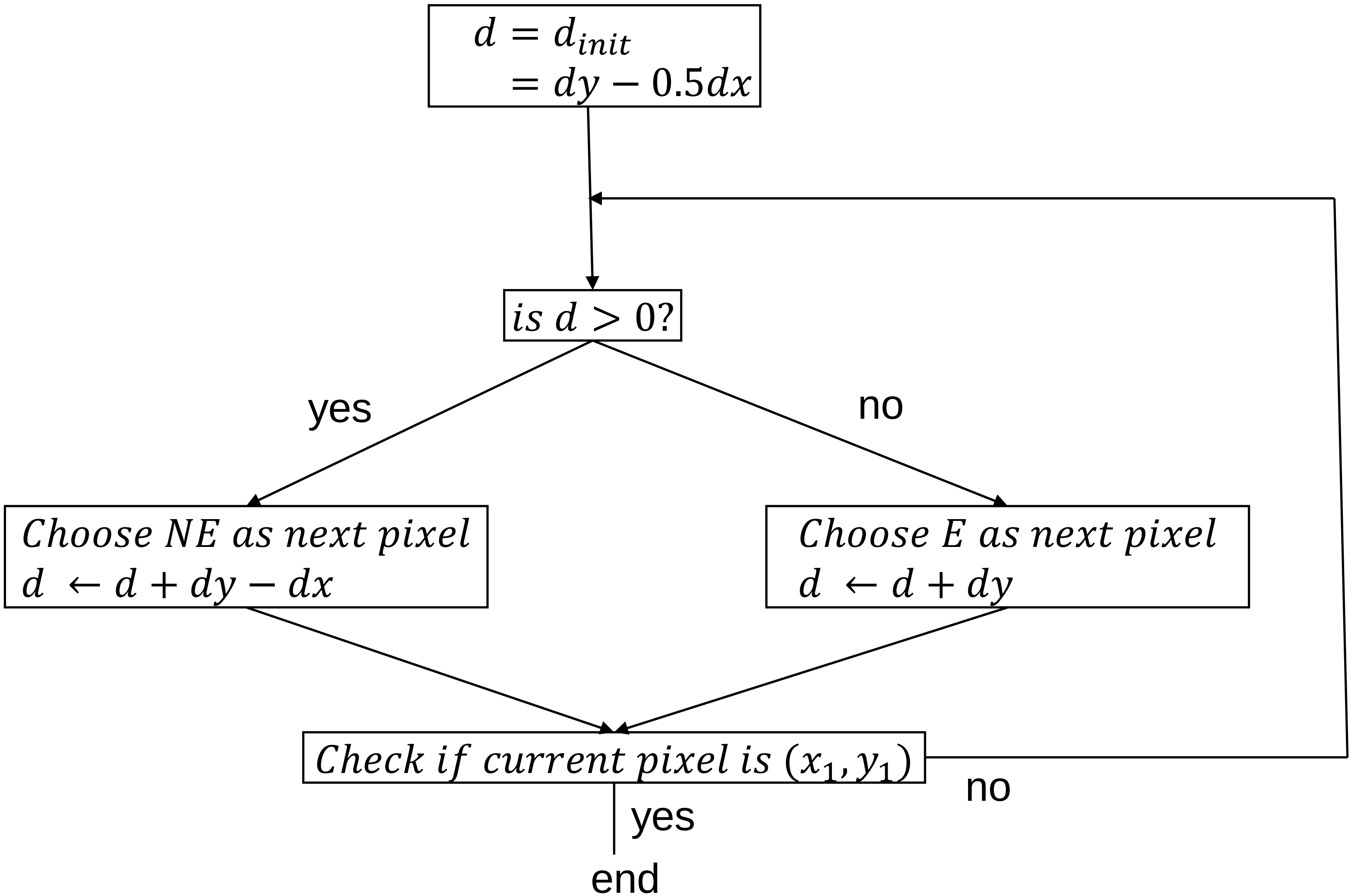
$$c = B.dx$$

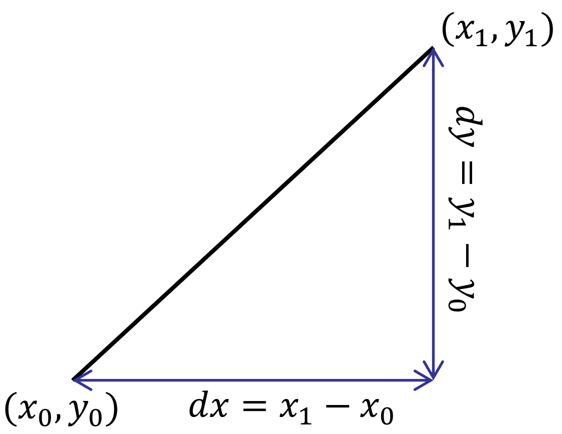

$d = d_{init}$
$= a + 0.5b$
$is\ d > 0?$
yes
no
$Choose\ NE\ as\ next\ pixel$
$d \leftarrow d + a + b$
$Choose\ E\ as\ next\ pixel$
$d \leftarrow d + a$
$Check\ if\ current\ pixel\ is\ (x_1, y_1)$
no
yes
end

Putting,

$a = dy$

$b = -dx$





$$y = mx + b$$

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

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

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

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

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

$$Comparing \; this \; with,$$

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

$$We \; get,$$

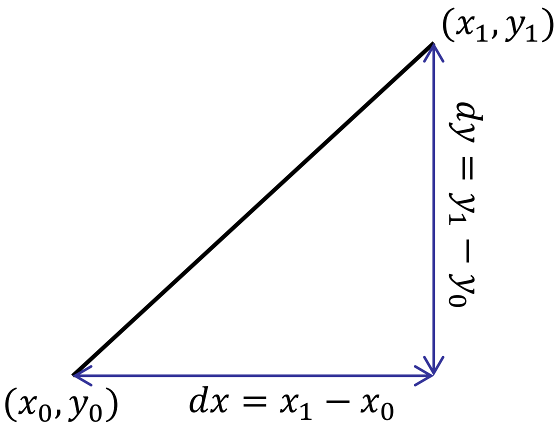
$$a = dy$$

$$b = -dx$$

$$c = b.dx$$

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

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



$$y = mx + b$$

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

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

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

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

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

Comparing this with,

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

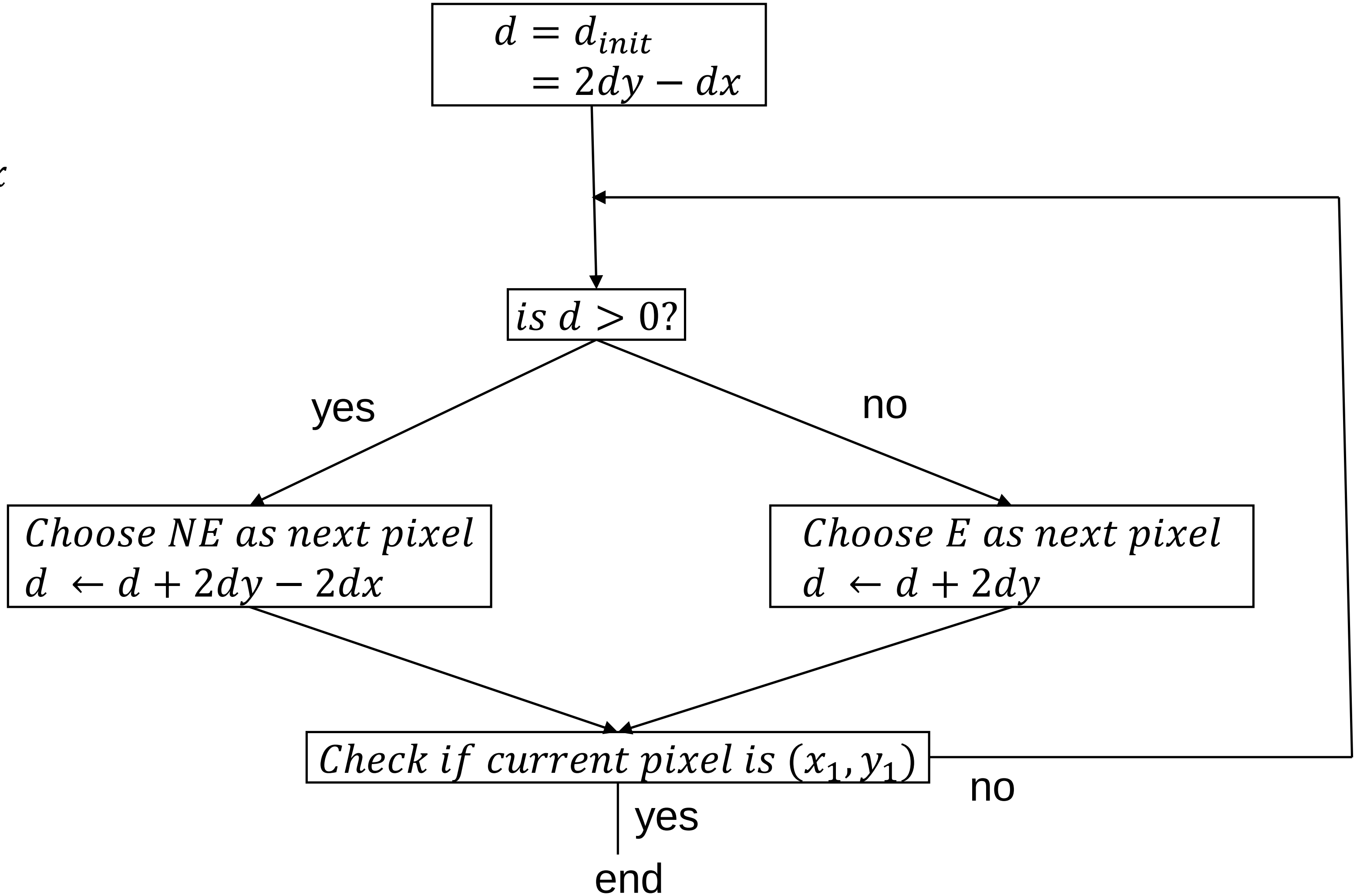
We get,

$$a = 2dy$$

$$b = -2dx$$

$$c = 2b.dx$$

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



Algorithm

```
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
        }
}
```