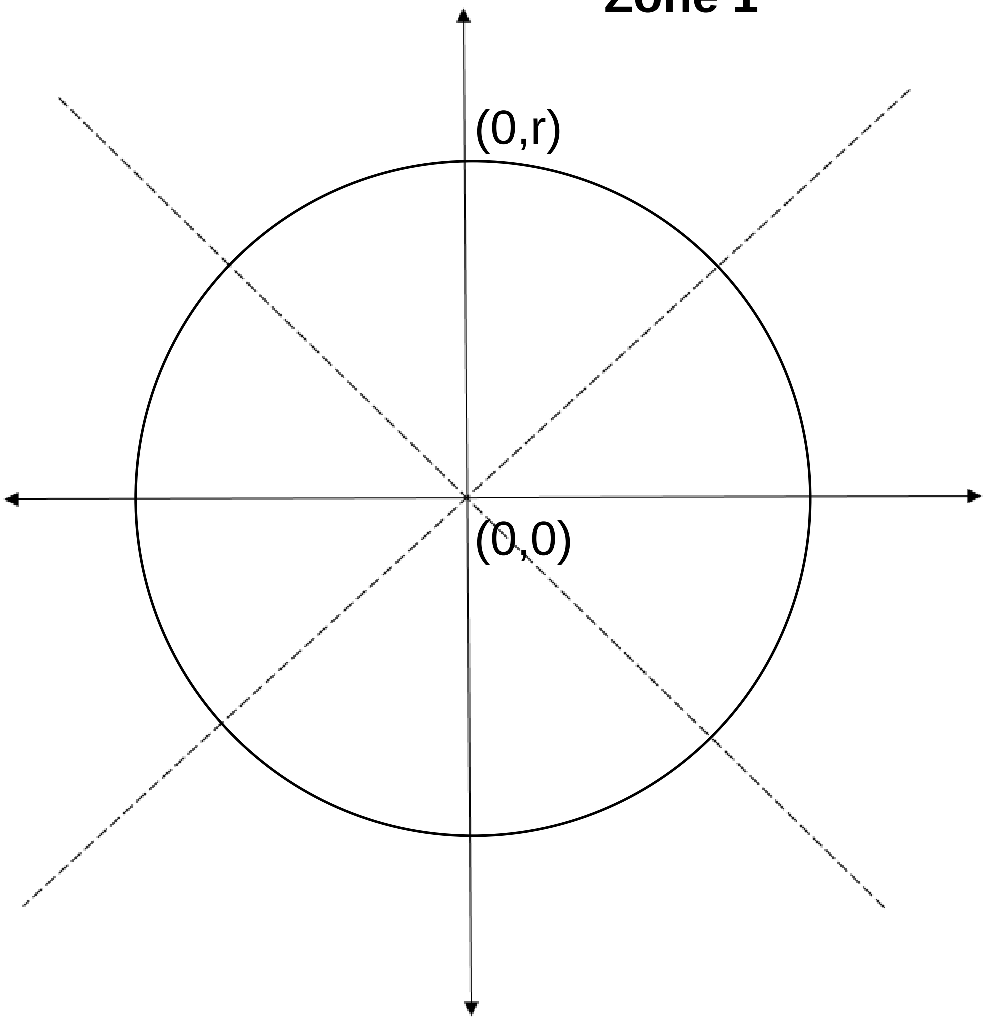
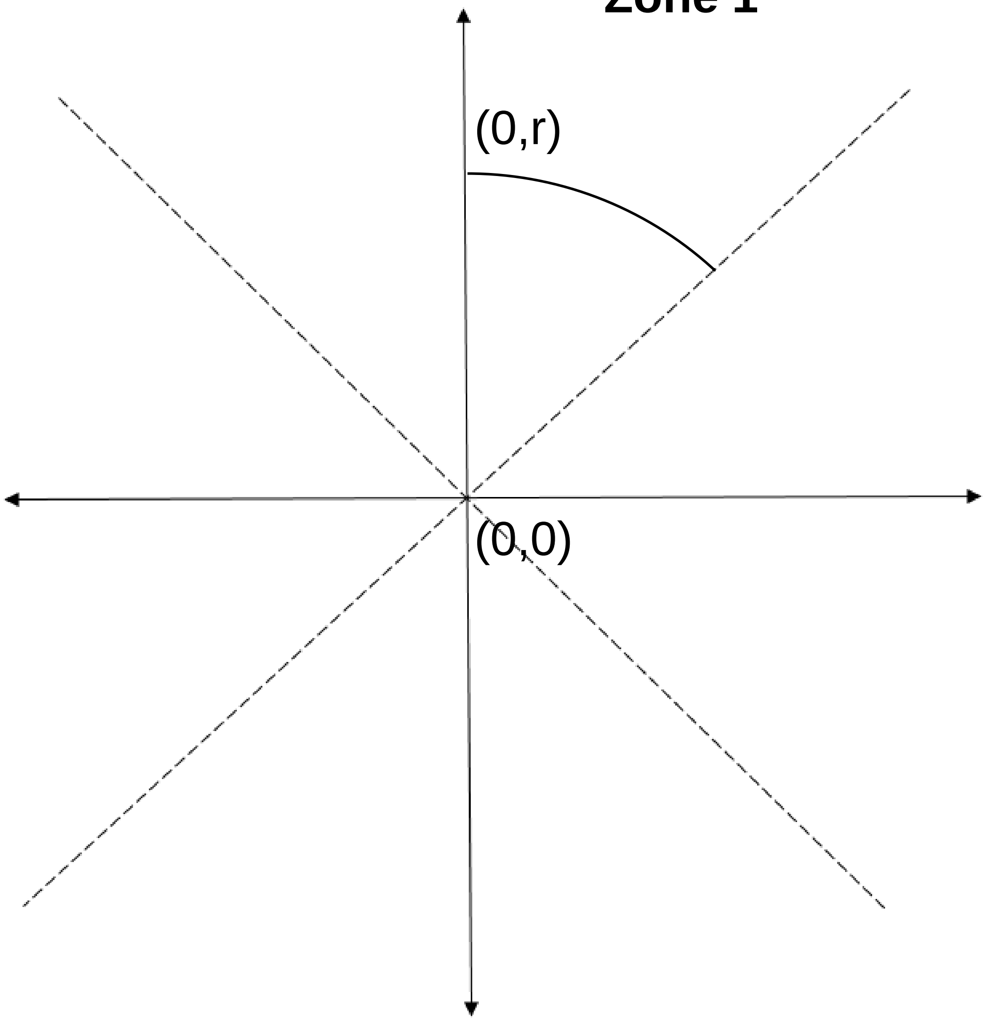
Computer Graphics: Line Drawing Algorithms

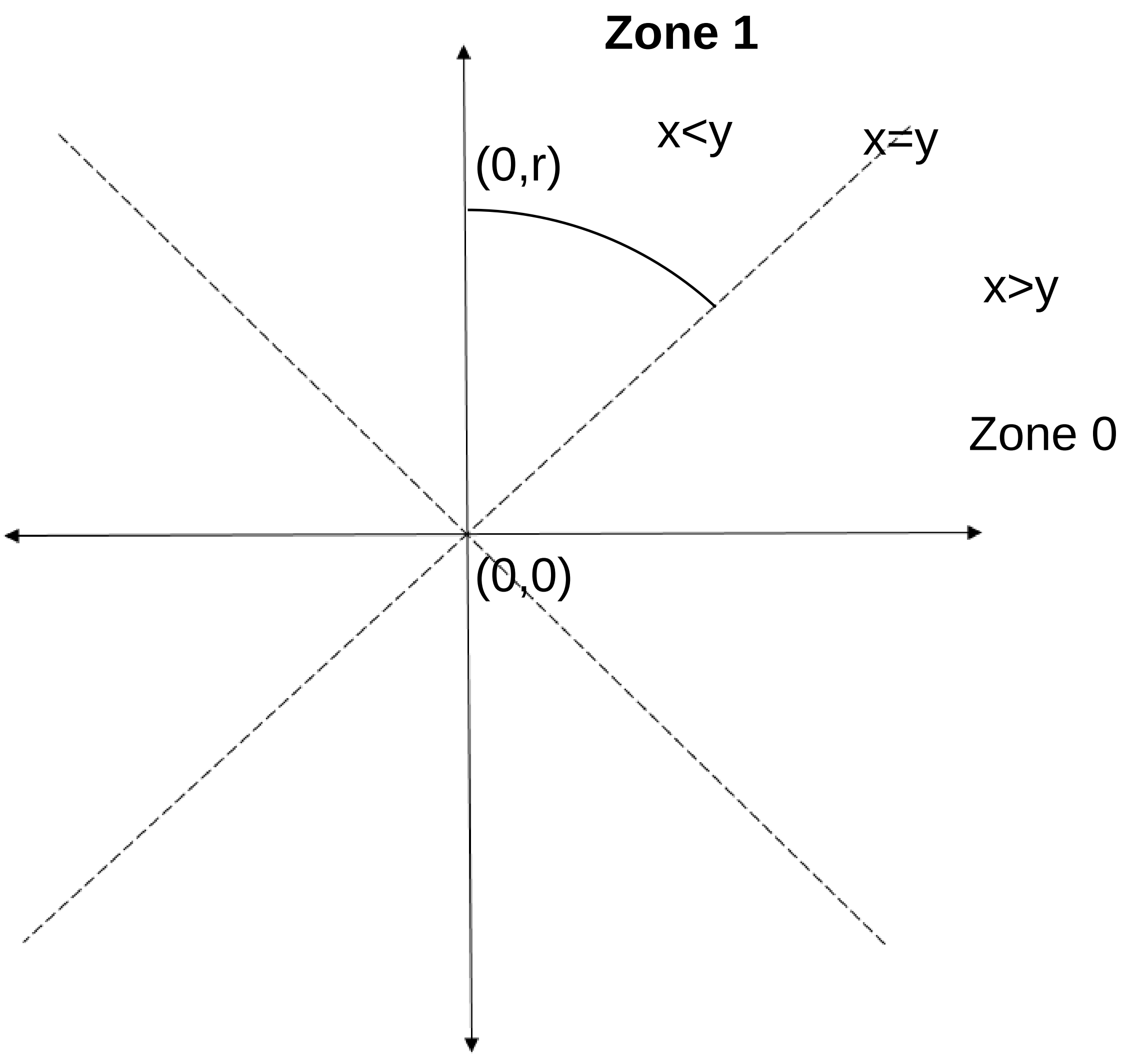
Scan Conversion Algorithms
(Midpoint Circle)

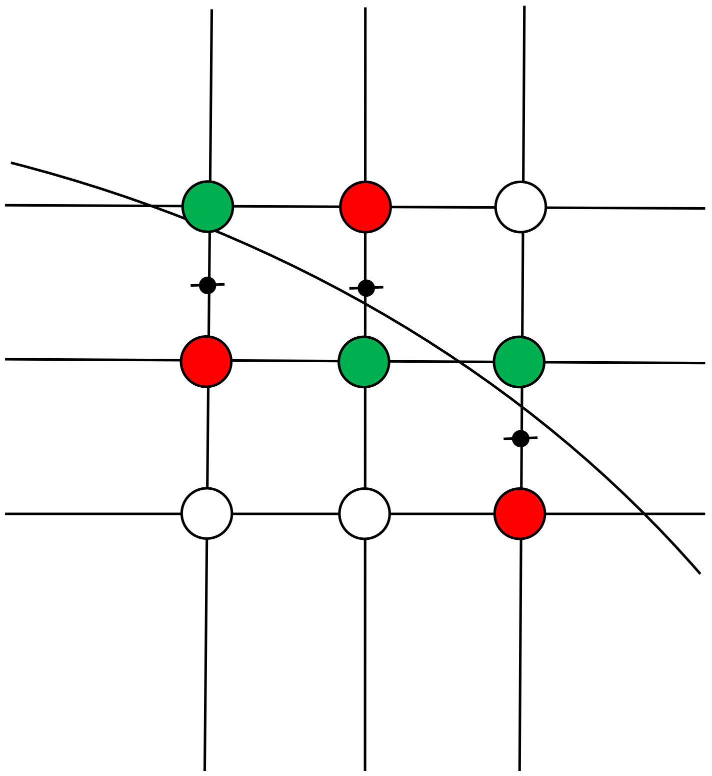
Zone 1



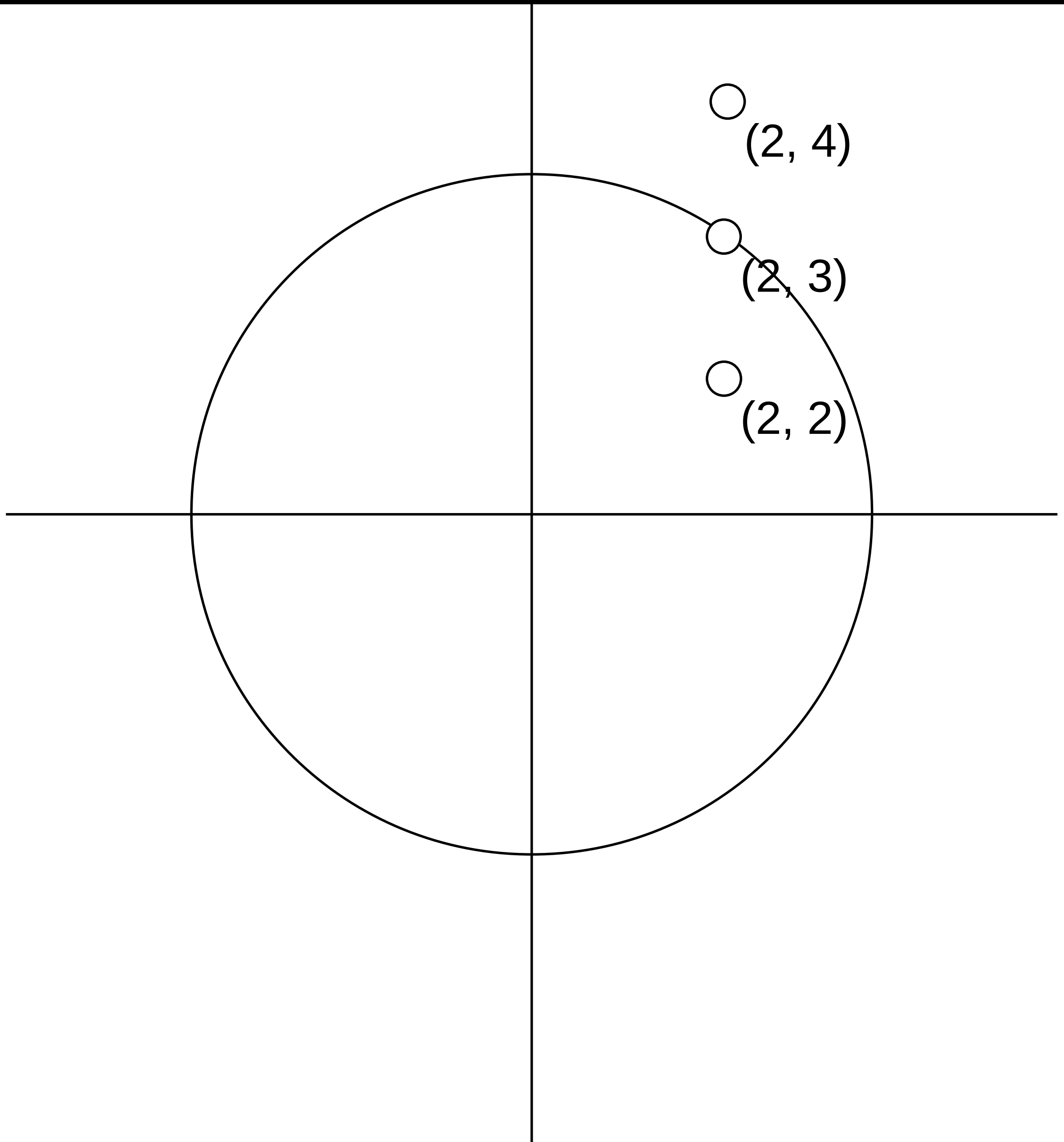
Zone 1



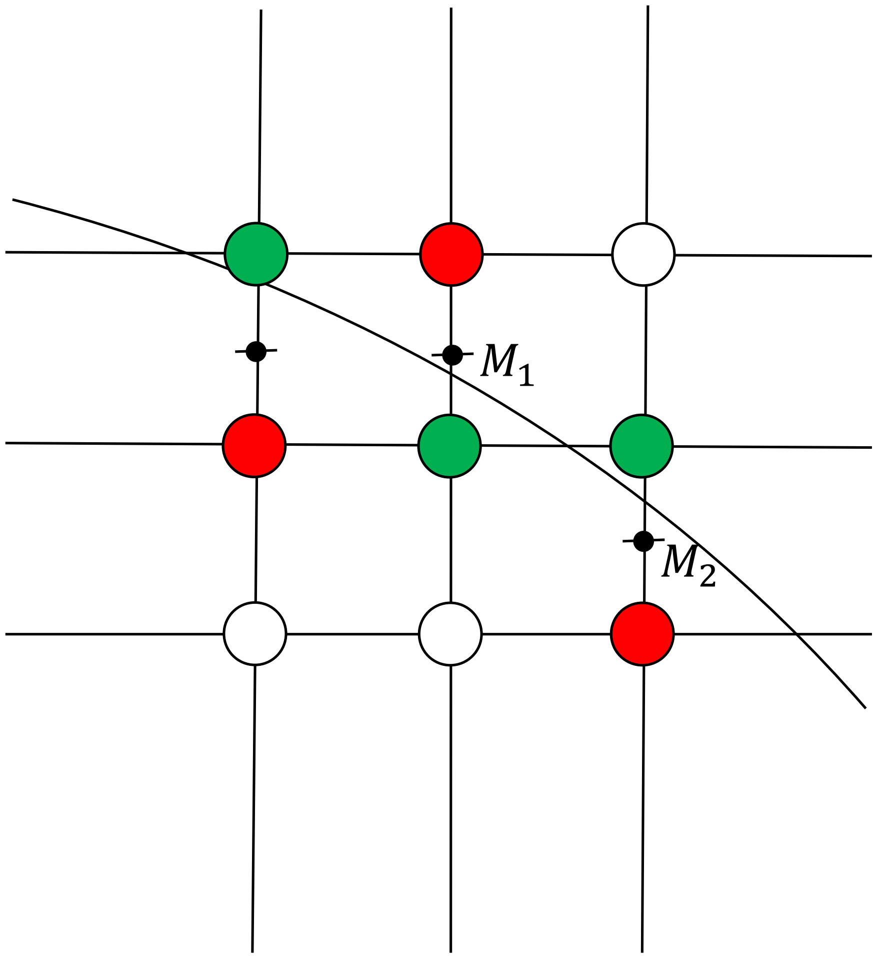

Zone 1
x<y
x=y
(0,r)
x>y
Zone 0
(0,0)

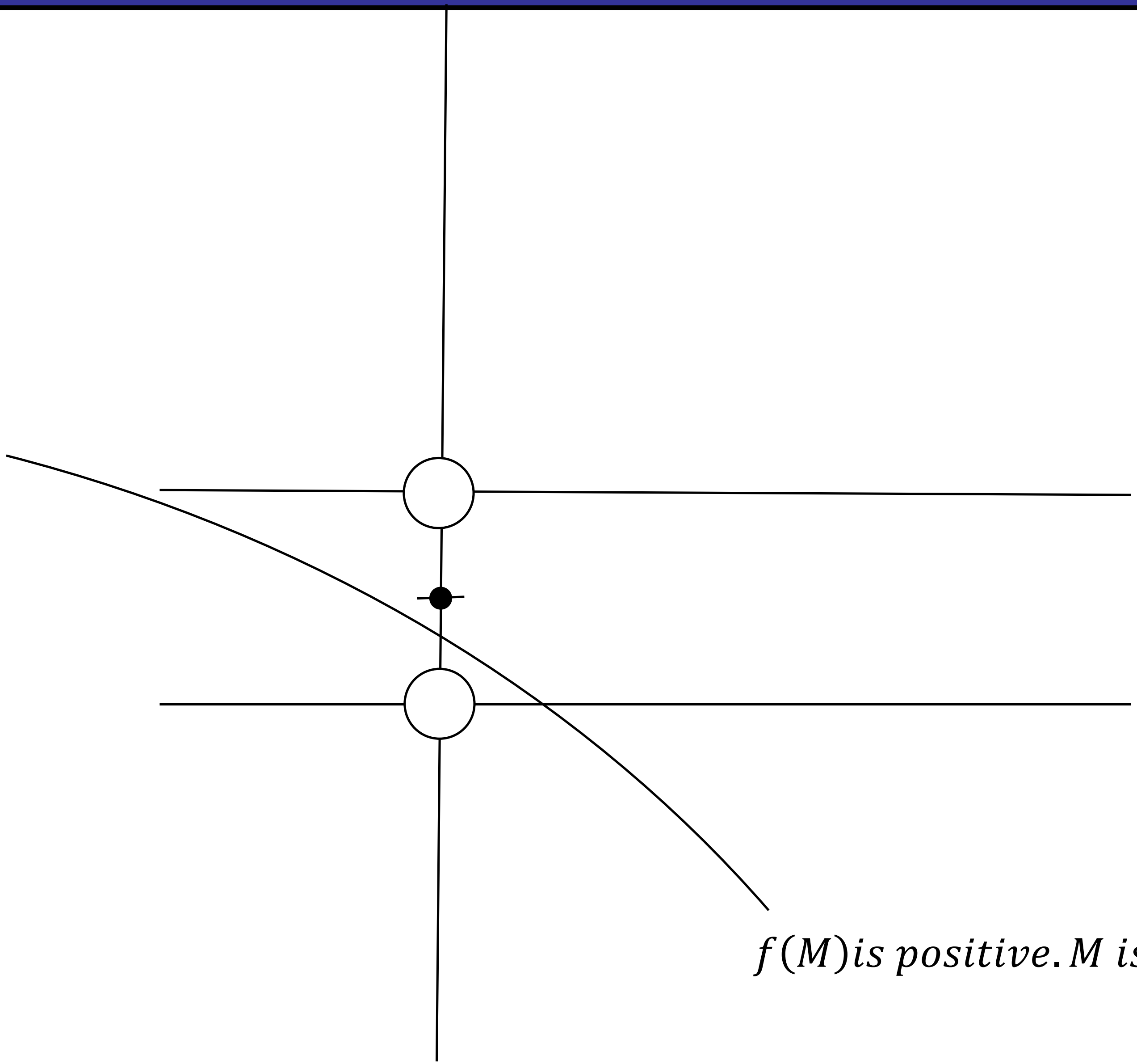


$$x^2 + y^2 = r^2$$
$$x^2 + y^2 - r^2 = 0$$
$$f(x, y) = x^2 + y^2 - r^2$$

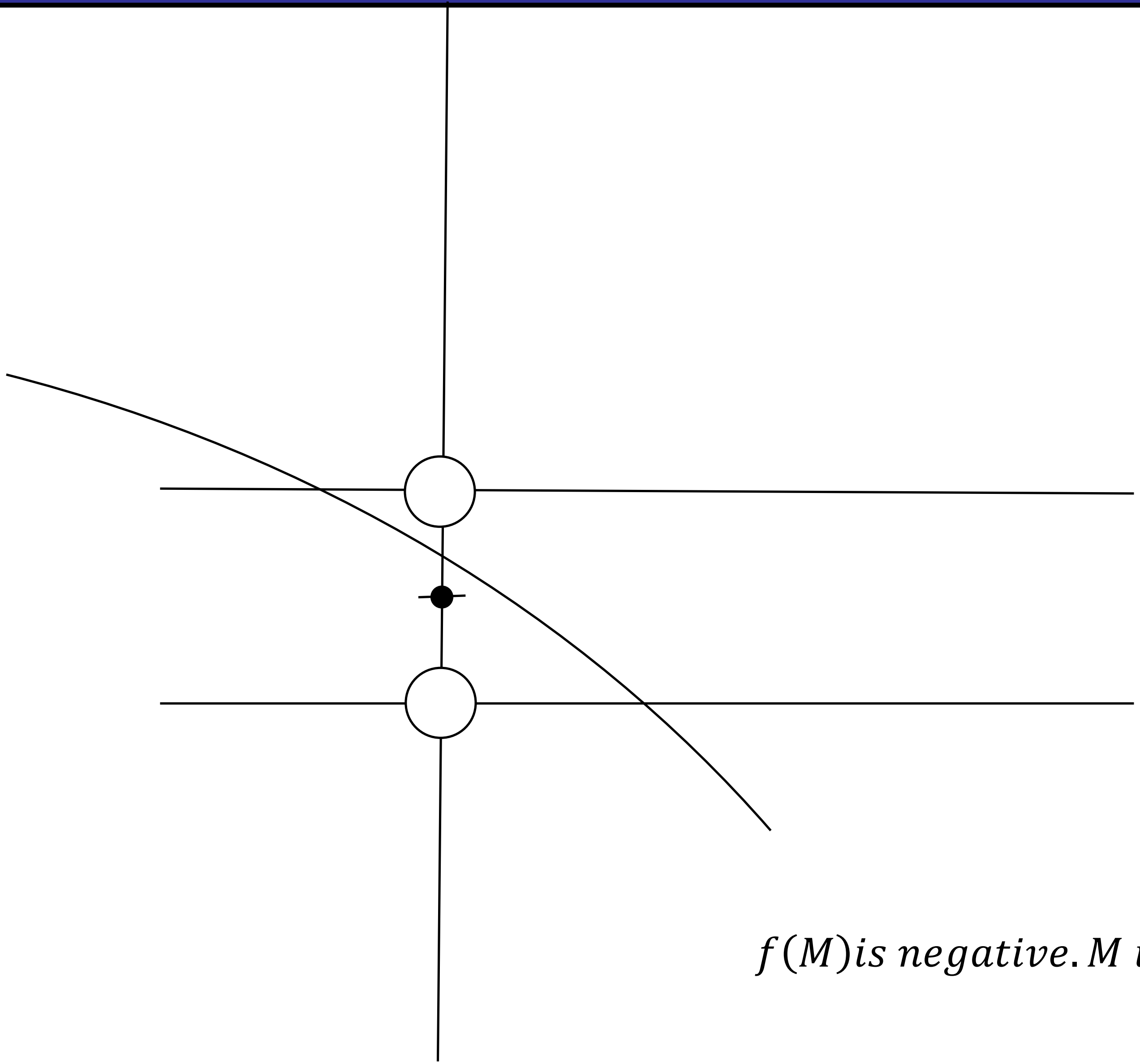


$$f(2,3) = 0$$
$$f(2,4) = (+)ve$$
$$f(2,2) = (-)ve$$


M_1
M_2

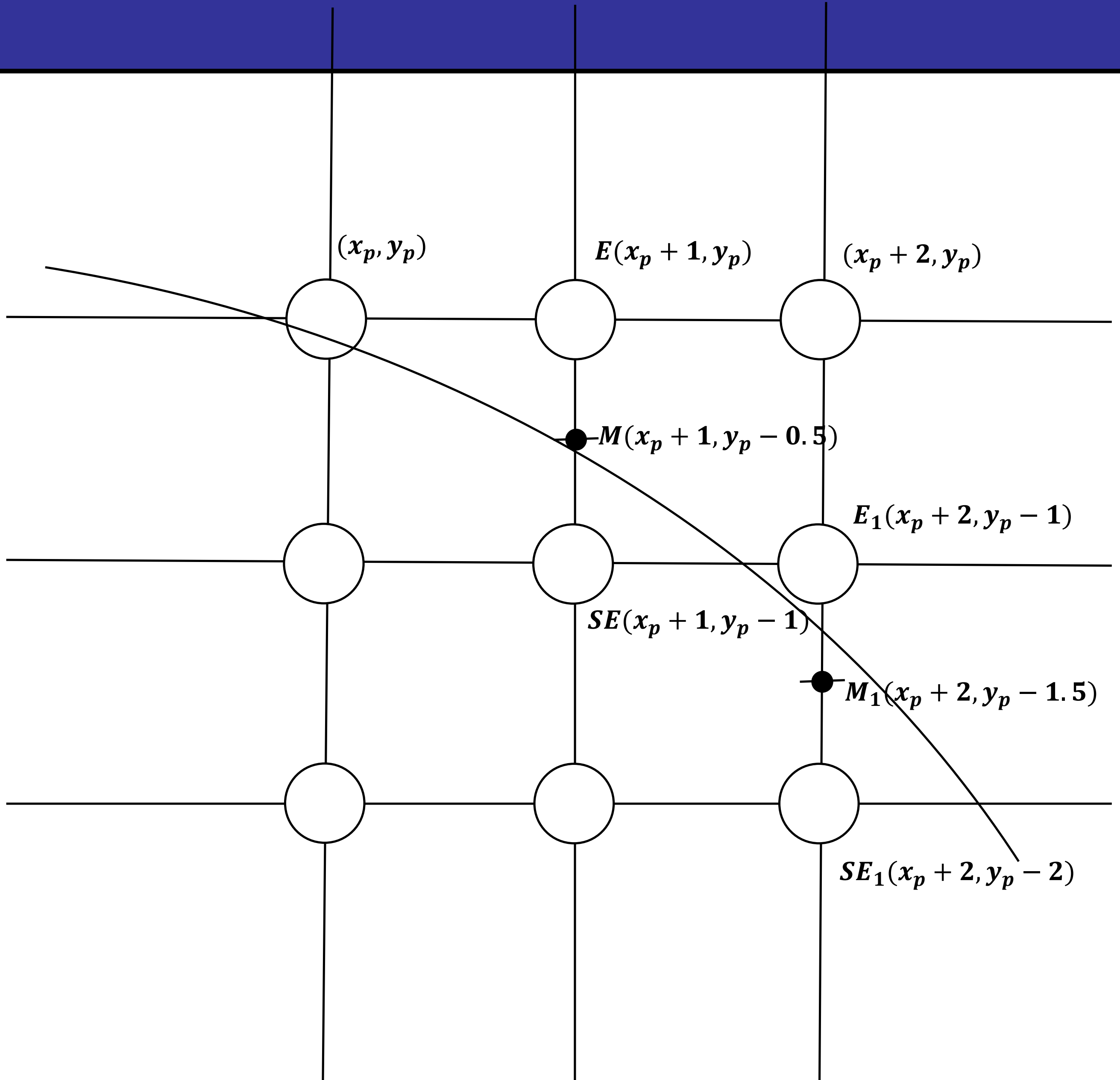


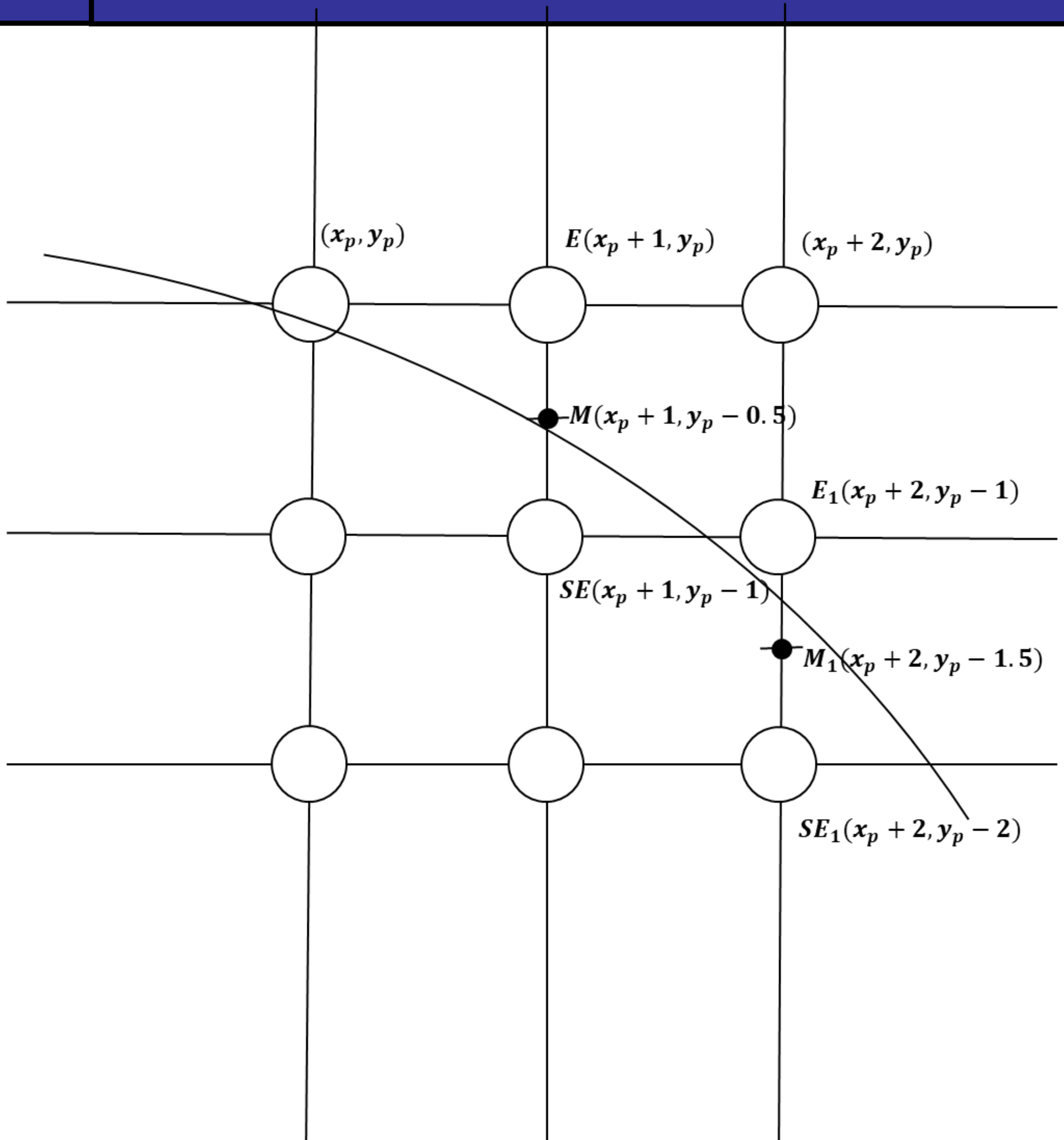
$f(M)$ is positive. M is outside the circle



$f(M) is\ negative. M\ is\ inside\ the\ circle$

f(M)	Pixel chosen
$f(M) \geq 0$	Lower
$f(M) < 0$	Upper

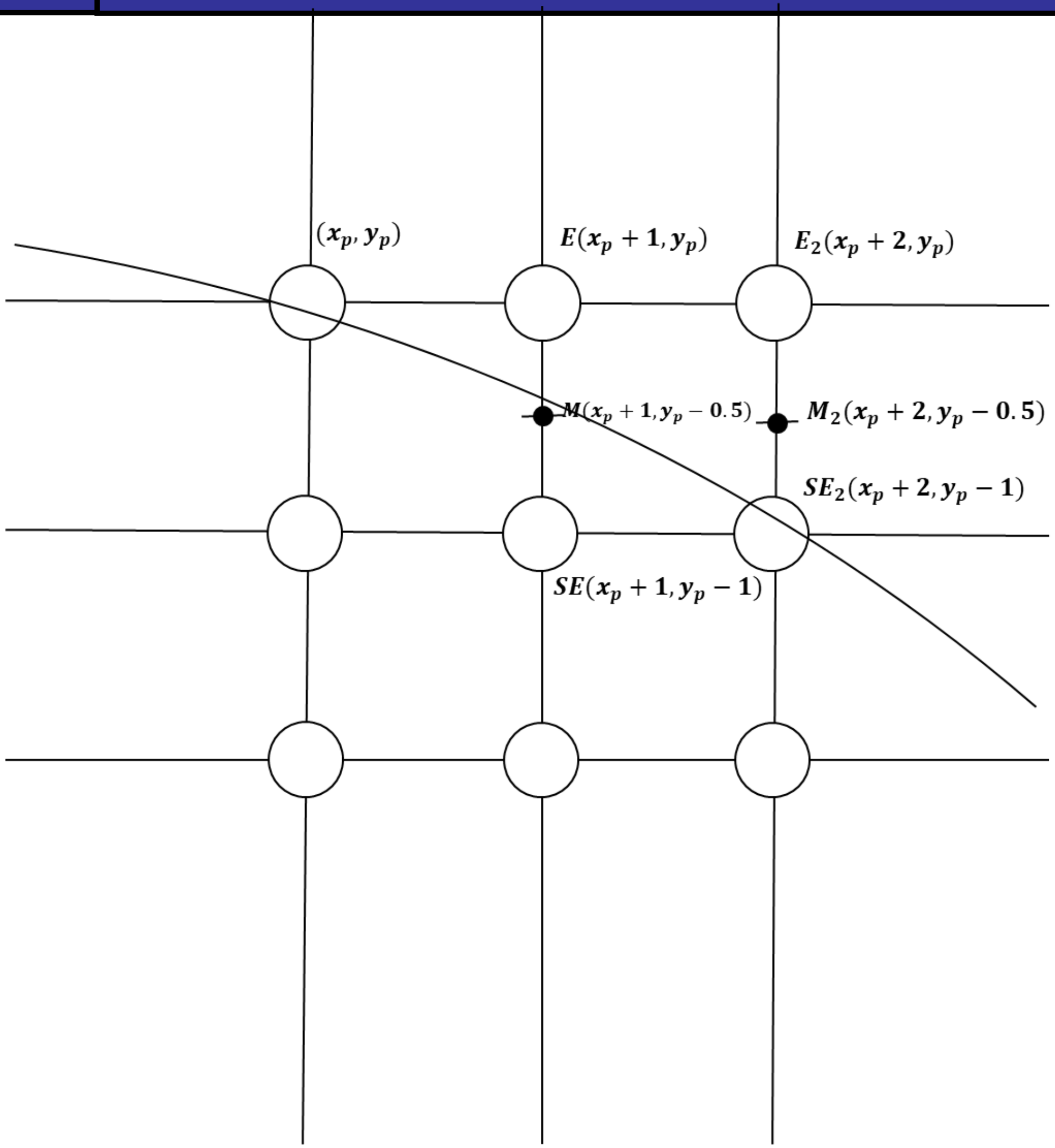

(x_p, y_p)
$E(x_p + 1, y_p)$
$(x_p + 2, y_p)$
$M(x_p + 1, y_p - 0.5)$
$E_1(x_p + 2, y_p - 1)$
$SE(x_p + 1, y_p - 1)$
$M_1(x_p + 2, y_p - 1.5)$
$SE_1(x_p + 2, y_p - 2)$



$$f(M) = f(x_p + 1, y_p - 0.5)$$
$$= (x_p+1)^2 + (y_p - 0.5)^2 - r^2$$
$$\text{d} = x_p^2 + 2x_p + 1 + y_p^2 - 2y_p.0.5 + 0.5^2 - r^2$$
$$= x_p^2 + 2x_p + 1 + y_p^2 - y_p + 0.25 - r^2$$
$$d \geq 0, so\ SE\ is\ chosen$$

$$f(M_1) = f(x_p + 2, y_p - 1.5)$$
$$= (x_p+2)^2 + (y_p - 1.5)^2 - r^2$$
$$\text{d}_{\text{new}} = x_p^2 + 4x_p + 4 + y_p^2 - 2y_p.1.5 + 1.5^2 - r^2$$
$$= x_p^2 + 4x_p + 4 + y_p^2 - 3y_p + 2.25 - r^2$$

$$d_{new} - d = 2x_p + 3 - 2y_p + 2$$
$$d_{new} = d + 2x_p - 2y_p + 5$$



$$
\begin{aligned}
f(M) &= f\left(x_p+1, y_p-0.5\right)\\
&= \left(x_p+1\right)^2 + \left(y_p-0.5\right)^2 - r^2\\
\mathrm{d} &= x_p^2 + 2x_p + 1 + y_p^2 - 2y_p.0.5 + 0.5^2 - r^2\\
&= x_p^2 + 2x_p + 1 + y_p^2 - y_p + 0.25 - r^2
\end{aligned}
$$

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

$$
\begin{aligned}
f(M_2) &= f\left(x_p+2, y_p-1.5\right)\\
&= \left(x_p+2\right)^2 + \left(y_p-0.5\right)^2 - r^2\\
\mathrm{d}_{\mathrm{new}} &= x_p^2 + 4x_p + 4 + y_p^2 - 2y_p.0.5 + 0.5^2 - r^2\\
&= x_p^2 + 4x_p + 4 + y_p^2 - y_p + 0.25 - r^2
\end{aligned}
$$

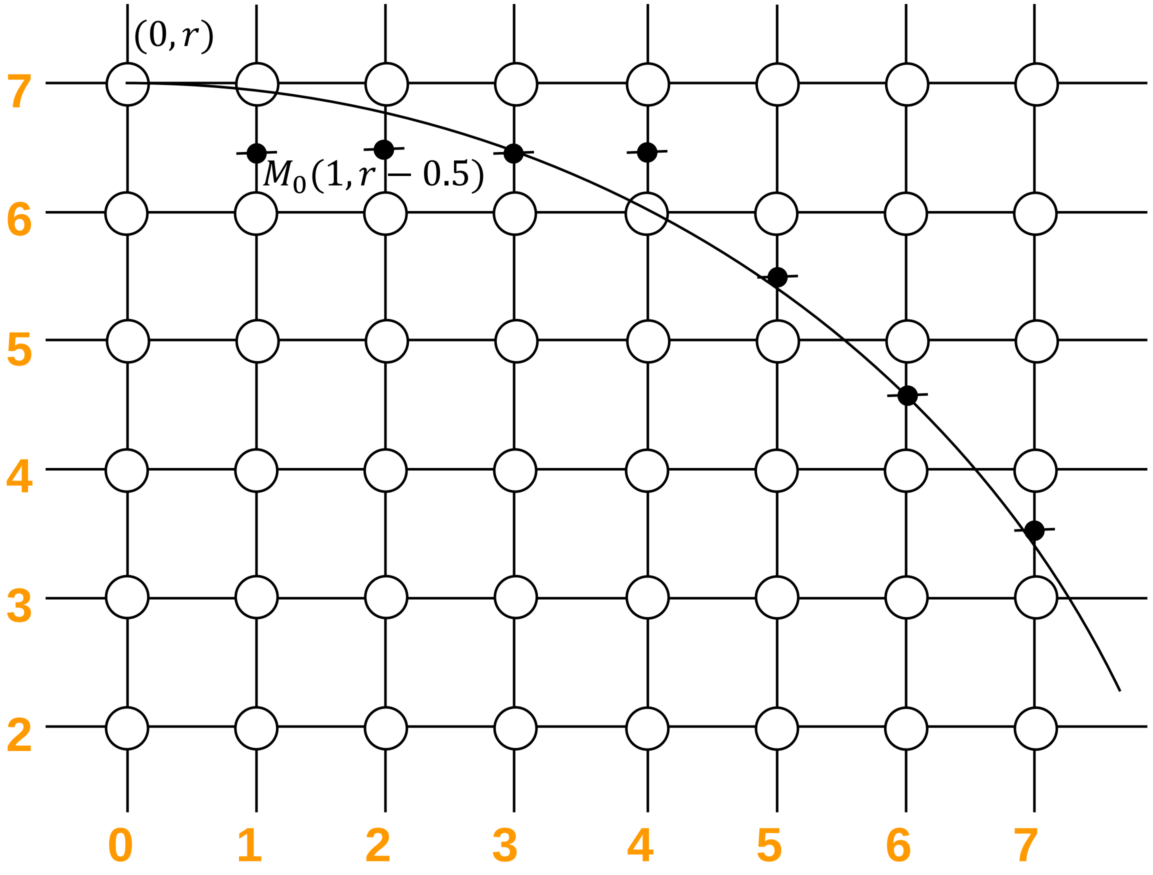
$$d_{new} - d = 2x_p + 3$$
$$d_{new} = d + 2x_p + 3$$

Calculate d for 1st column.

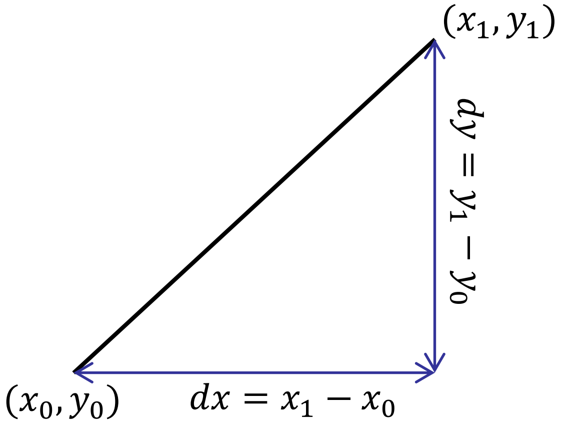
Choose E/SE.

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

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



$$
\begin{aligned}
d_{init} &= f(M_0) \\
&= \mathrm{f}(1, r - 0.5) \\
&= 1^2 + (r - 0.5)^2 - r^2 \\
&= 1 + r^2 - r + 0.25 - r^2 \\
d_{init} &= 1.25 - r
\end{aligned}
$$



$$y = mx + B$$

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

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

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

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

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

Comparing this with,

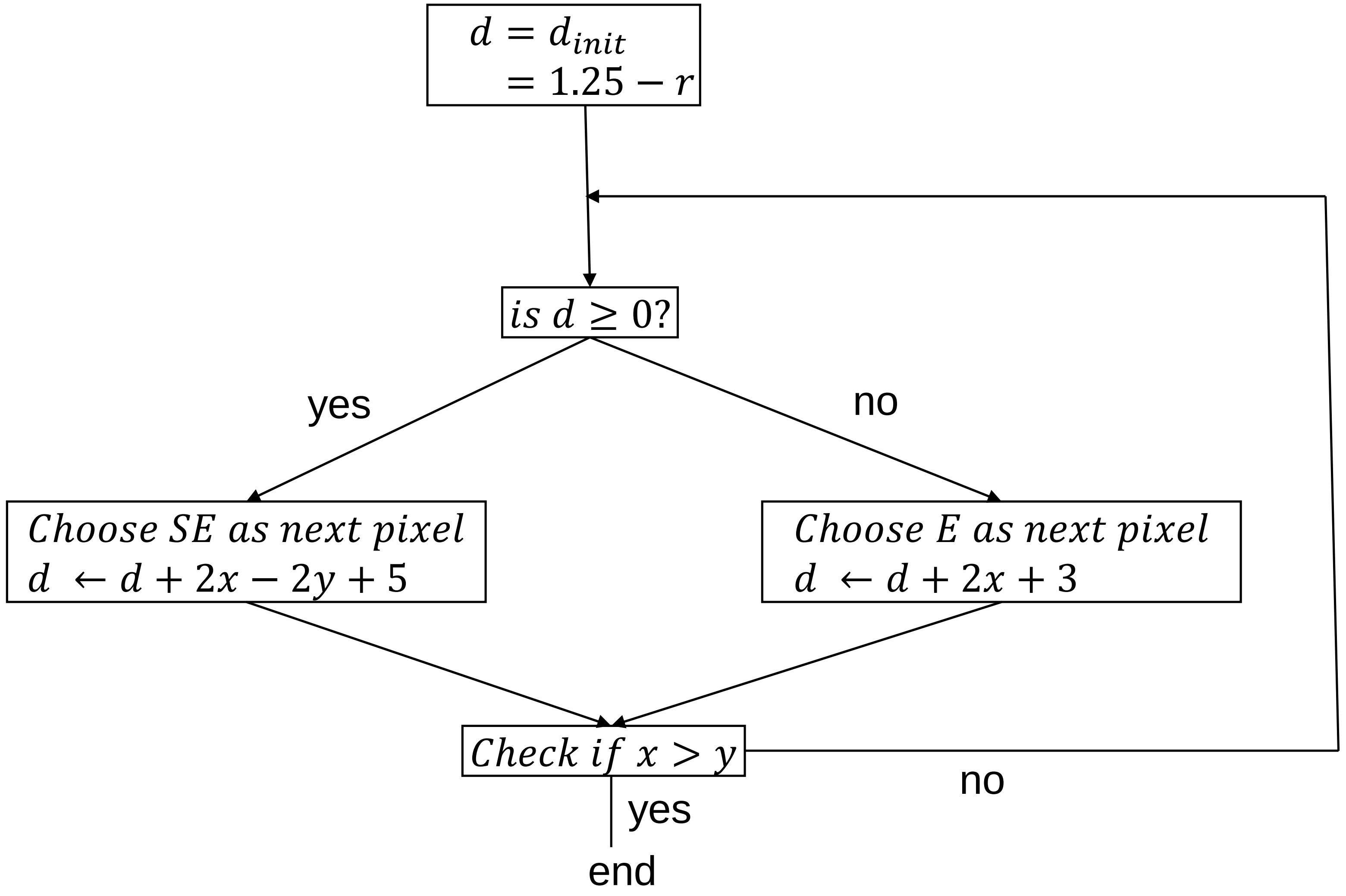
$$ax \quad + by \quad + c \quad = 0$$

We get,

$$a = dy$$

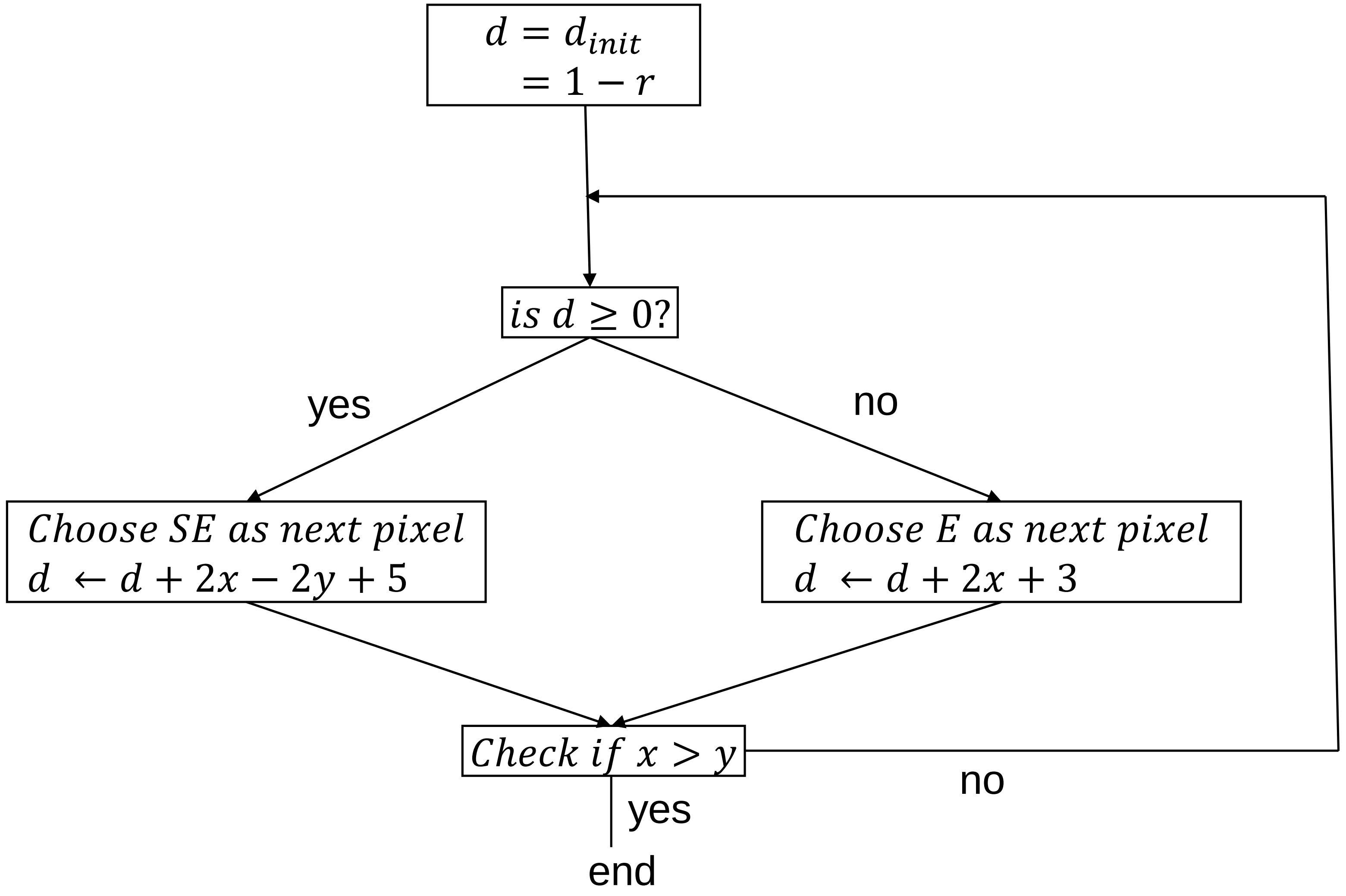
$$b = -dx$$

$$c = B.dx$$


$d = d_{init} = 1.25 - r$
$is\ d \geq 0?$
yes
no
$Choose\ SE\ as\ next\ pixel$
$d \leftarrow d + 2x - 2y + 5$
$Choose\ E\ as\ next\ pixel$
$d \leftarrow d + 2x + 3$
$Check\ if\ x > y$
no
yes
end

r	1.25-r	1-r
1	0.25, (+)ve	0, (+)ve
2	-0.75, (-)ve	-1, (-)ve
3	-1.75, (-)ve	-2, (-)ve
4	-2.75, (-)ve	-3, (-)ve
5	-3.75, (-)ve	-4, (-)ve

Using 1-r would be same as using 1.25-r as starting value of d


$d = d_{init} = 1 - r$
$is\ d \geq 0?$
yes
no
$Choose\ SE\ as\ next\ pixel$
$d \leftarrow d + 2x - 2y + 5$
$Choose\ E\ as\ next\ pixel$
$d \leftarrow d + 2x + 3$
$Check\ if\ x > y$
no
yes
end

Algorithm

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

```
void Circlepoints(int x, int y, int value){
        WritePixel (x, y, value);
        WritePixel (y, x, value);
        WritePixel (y, —x, value);
        WritePixel (x, —y, value);
        WritePixel (—x, —y, value);
        WritePixel (—y,—x, value);
        WritePixel (—y, x, value);
        WritePixel (—x, y, value);
}
```