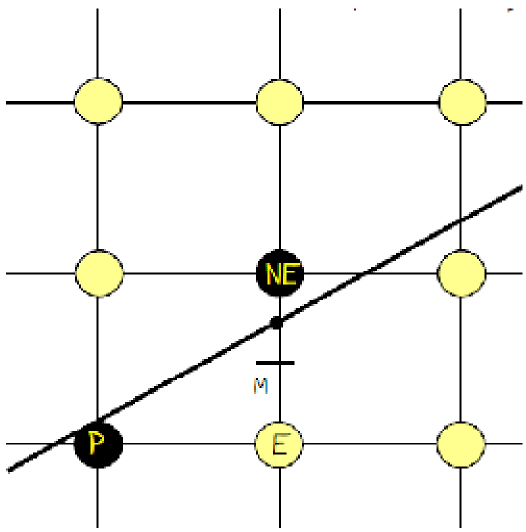
Bresenham's Algorithm

In order to remove floating point operations in line drawing algorithm, we need to remove *m* from the operations which is used in algorithms.

It's an accurate and efficient raster line algorithm. The most important decision to make is once a pixel is turned on, which is the next pixel will be turned on? I.e. if we had plotted pixel on (X_k,Y_k) then next pixel would be either (X_{k+1},Y_k) or (X_{k+1},Y_{k+1}).



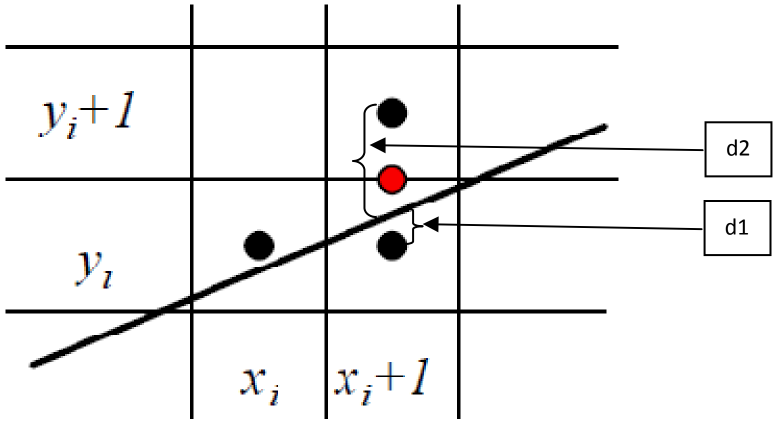
Let us try to understand the decision variables. Let us assume that we are drawing a line *y=mx+b* that passes through a point (X_0,Y_0). Here, 0<m<1. Let us also assume that the last pixel turned on is (X_k,Y_k) and the decision to be made for the next step that is for the vertical distance X_{k+1}.

Now let us assume vertical row of pixel which passes through horizontal distance X_{k+1}. There are 3 vertical points, (X_{k+1}, Y_k), (X_{k+1}, Y) and (X_{k+1}, Y_{k+1}), fall on this assumed line .In addition, the assumed distance between (X_{k+1}, Y_k) and (X_{k+1}, Y) is d1 and that between (X_{k+1}, Y) and (X_{k+1}, Y_{k+1}) is d2.

d1 < d2

OR

$P_k < 0$

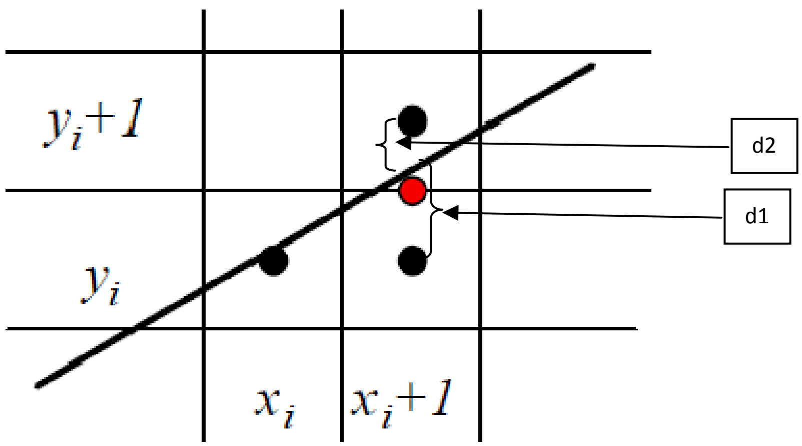


Choose $(x_i + 1, y_i)$

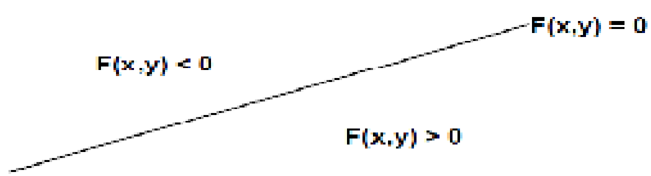
d1 > d2

OR

$P_k > 0$



Choose (x_i+1, y_{i+1})



The Y coordinate on the mathematical line at pixel column position X_{k+1} is calculated as,

$$Y = m (X_{k+1}) + b$$

Therefore,

$$d1 = Y – Y_k$$

$$= m (X_k + 1) + b – Y_k$$

$$d2 = Y_k +1 – Y$$

$$= Y_k + 1 – m (X_k + 1) – b$$

Therefore,

$$d1 – d2 = m (X_k + 1) + b – Y_k - Y_k - 1 + m (X_k + 1) + b$$

$$= 2m(X_k + 1) + 2b - 2Y_k – 1$$

Now, we want integer calculation so we will remove m by substituting m=dy/dx.

$$d1 – d2 = 2 dy/dx (X_k + 1) + 2b - 2Y_k – 1$$

$$dx(d1 – d2) = 2 dy (X_k + 1) + 2b dx - 2Y_k dx – dx$$

$$= 2 dy X_k + 2 dy + 2b dx - 2Y_k dx – dx$$

$$P_k = 2 dy X_k - 2Y_k dx + c$$

Where, c = 2 dy + 2b dx – dx and it is independent of pixel position and P_k is a decision parameter for k^{th} step.

The sign of decision parameter is same as sign of d1 – d2 as for our case 0 < m < 1, that is dx > 0

So, if P_k is positive that is dx (d1 – d2) > 0, and d1 > d2. Therefore, the upper pixel at position $Y_k + 1$ is closer to the line than the pixel Y_k, hence the pixel at position $Y_k + 1$ will be activated. In case of negative P_k, that is dx (d1- d2) < 0, d2 > d1 and therefore the pixel at position Y_k will be activated.

Now for the $(k + 1)^{st}$ step,

$$P_{k+1} = 2 dy X_{k+1} - 2Y_{k+1} dx + c$$

Therefore,

$$P_{k+1} - P_k = 2\,dy\,X_{k+1} - 2Y_{k+1}\,dx + c - (2\,dy\,X_k - 2Y_k\,dx + c)$$

$$= 2\,dy\,(X_{k+1} - X_k) - 2\,dx\,(Y_{k+1} - Y_k)$$

$$= 2\,dy - 2\,dx\,(Y_{k+1} - Y_k)\text{ , since } X_{k+1} - X_k = 1$$

$$\boxed{P_{k+1} = P_k + 2\,dy - 2\,dx\,(Y_{k+1} - Y_k)}$$

Where value of ($Y_{k+1} - Y_k$) will be 0 or 1. If $P_k < 0$ then it is 0 and 1 if $P_k > 0$.

But 1st parameter calculation P_0 is evaluated using,

$$\boxed{P_0 = 2\,dy - dx}$$

Note: - for m > 1 replace the role of X and Y and equations remains same in positive and negative slope case.