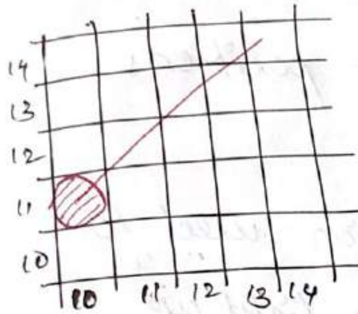
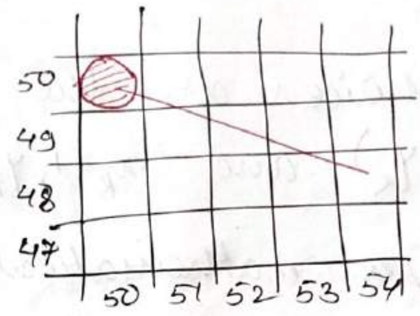
Bresenham's Line Algorithm

- An accurate and efficient raster line generating algorithm, developed by Bresenham which scan converts line using only incremental integer calculations that can be modified to display circle and other curvers.



Line with +ve slop starting with (10,11)



Line with -ve slop starting from (50,50)

- The vertical axes shows scan-line position and the horizonatal axes identify pixel column.
- sampling at unit x intervals in these examples, we need to decide which two possible pixel position is close to the line path at each sample step.
- To illustrate bresenham's approach, we first consider the scan conversion for lines with positive slop less than 1.
- pixel position along a line path are then determined by sampling at unit x interval.

- starting from left end point (x_0, y_0) of given line, we step to each successive column and plot the pixel whose scan line y values is closest to the line path.
- Assuming we have determined that the pixel at (x_k, y_k) is to be displayed, we next need to decide which pixel to plot in column $x_k + 1$
- Our choices are the pixels at positions (x_k+1, Y_k) and (x_k+1, Y_k+1)
- Let's see mathematical calculation used to decide which pixel position is light up.
- we know that equation of line is

$$Y = mx + b$$

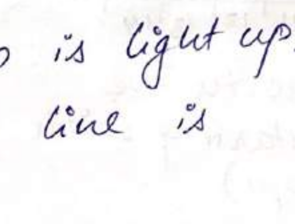
Now for position (x_k+1)

$$y = m(x_k+1) + b$$

- Now calculate distance between actual line's y value and lower pixel as d_1 and distance between actual line's y value and upper pixel as d_2.

$$d_1 = y - y_K$$

$$= m(x_k+1) + b - y_K \quad \text{— (1)}$$



$$d_2 = (y_K+1) - y$$

$$= (y_K+1) - m(x_k+1) - b \quad \text{— (2)}$$

• Now calculate $d_1 - d_2$ from eq^n (1) & (2).

$$d_1 - d_2 = \{m(x_k+1) + b - y_k\} - \{(y_k+1) - m(x_k+1) - b\}$$
$$= m(x_k+1) + b - y_k - y_k - 1 + m(x_k+1) + b$$
$$= 2m(x_k+1) - 2y_k + 2b - 1 \quad \text{——— (3)}$$

• Now we have decision parameter P_k for k^{th} step in the line algorithm is given by

$$P_k = \Delta x (d_1 - d_2)$$
$$= \Delta x \{2m(x_k+1) - 2y_k + 2b - 1\}$$
$$= \Delta x \{2\frac{\Delta y}{\Delta x}(x_k+1) - 2y_k + 2b - 1\}$$
$$= 2\Delta y x_k + 2\Delta y - 2\Delta x y_k + 2\Delta x b - \Delta x \quad \text{—— (4.1)}$$
$$P_k = 2\Delta y x_k - 2\Delta x y_k + C \quad \{\text{where } C = 2\Delta y + 2\Delta x b - \Delta x \text{ (as constant)}\}$$
(4)

• Now if P_k is negative then we plot lower pixel otherwise we plot upper pixel.

• So successive decision parameters are calculated as

$$P_{k+1} = 2\Delta y x_{k+1} - 2\Delta x y_{k+1} + C$$

• Now subtract P_k from P_{k+1}

$$P_{k+1} - P_k = 2\Delta y x_{k+1} - 2\Delta x y_{k+1} + \cancel{C} - 2\Delta y x_k + 2\Delta x y_k - \cancel{C}$$
$$= 2\Delta y (x_{k+1} - x_k) - 2\Delta x (y_{k+1} - y_k)$$

But wkt. $x_{k+1} = x_k + 1$, so $(x_{k+1} - x_k) = 1$

Hence $P_{k+1} = P_k + 2\Delta y - 2\Delta x (y_{k+1} - y_k)$ — (5)

- where the term $y_{k+1} - y_k$ is either 0 or 1 depending on sign of parameter P_k

so putting $y_{k+1} - y_k = 0$ in eq.ⁿ (5) we get

$$\boxed{P_{k+1} = P_k + 2\Delta y}$$

putting $y_{k+1} - y_k = 1$ in eq.ⁿ (5) we get

$$\boxed{P_{k+1} = P_k + 2\Delta y - 2\Delta x}$$

- The first decision parameter P_0 is calculated using eq.ⁿ (4.1) by putting (x_k, y_k) as (x_0, y_0)

$$P_k = 2\Delta y\, x_k - 2\Delta x y_k + 2\Delta y + 2\Delta x b - \Delta x$$

$$P_0 = 2\Delta y\, x_0 - 2\Delta x y_0 + 2\Delta y + 2\Delta x b - \Delta x$$

substitute $b = y_0 - m x_0$

$$P_0 = 2\Delta y\, x_0 - 2\Delta x y_0 + 2\Delta y + 2\Delta x(y_0 - m x_0) - \Delta x$$

substitute $m = \Delta y / \Delta x$

$$P_0 = 2\Delta y x_0 - 2\Delta x y_0 + 2\Delta y + 2\Delta x\left(y_0 - \frac{\Delta y}{\Delta x} x_0\right) - \Delta x$$

$$= \cancel{2\Delta y x_0} - \cancel{2\Delta x y_0} + 2\Delta y + \cancel{2\Delta x y_0} - \cancel{2\Delta y x_0} - \Delta x$$

$$\boxed{P_0 = 2\Delta y - \Delta x}$$

Algorithm :- for $|m| < 1$

1. Input the two line endpoints and store the left endpoint in (x_0, y_0)
2. Load (x_0, y_0) into the frame buffer; that is plot the first point.
3. Calculate constants Δx, Δy, $2\Delta y$ and $2\Delta y - 2\Delta x$ and obtain the starting value for the decision parameter as

$$P_0 = 2\Delta y - \Delta x$$

4. At each x_k along the line starting at $k = 0$, perform the following test:

If $P_k < 0$, then next point to plot is (x_k+1, y_k) and

$$P_{k+1} = P_k + 2\Delta y$$

otherwise, the next point to plot is (x_k+1, y_k+1) and

$$P_{k+1} = P_k + 2\Delta y - 2\Delta x$$

5. Repeat step 4 Δx times.