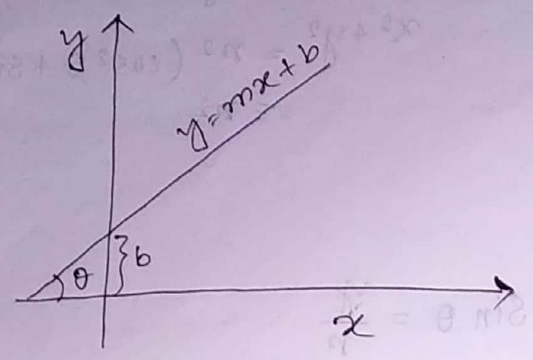
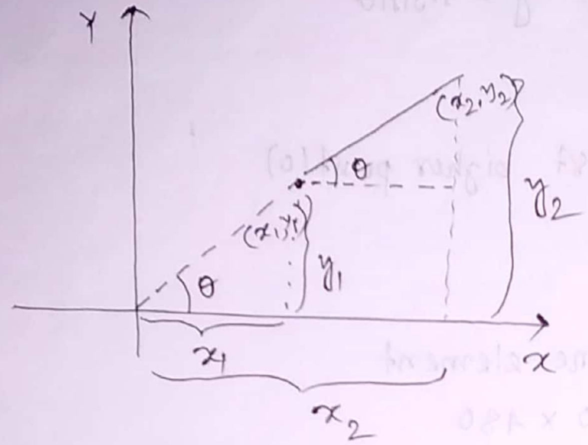
[AR]

Scan Conversion of Line

The equation of straight line in cartesian form

$$y = mx + b \quad \text{---} \; (1) \qquad m = \tan\theta$$





DDL → Digital Diffential Analyzer

$m = \tan\theta$, $m = \dfrac{y_2 - y_1}{x_2 - x_1}$

$\dfrac{dy}{dx} = K = \text{const.}$ [Diffential straight line]

$dy = m\,dx$

geometory representation → straight line (—)

DDL line drawing Algorithm

This algorithm works on the principle of obtaining the successive pixel values based on the differential of eqn of the straight line

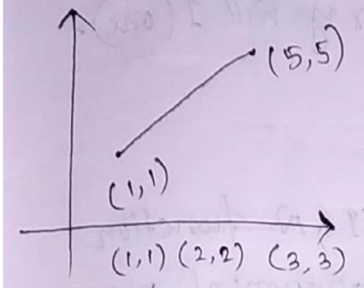
$$\frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

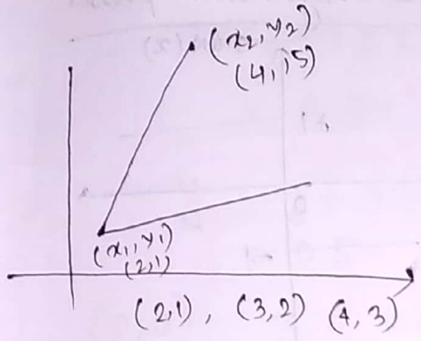
$$\Delta y = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)\Delta x$$

Hence we can obtain next value of y by using it's previous value.

$$y_{i+1} = y_i + \Delta y$$

$$= y_i + \left(\frac{y_2 - y_1}{x_2 - x_1}\right) \cdot \Delta x$$ the rester until be chosen to be either Δx or Δy depending on which is larger among the two.





20/01

$$\frac{dy}{dx} = \text{const.}$$

Let $\lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \frac{dy}{dx}$

$\Delta y = y_2 - y_1$

$\Delta x = x_2 - x_1$

DDA line Drawing Algorithm

step 1: Read the end points (x_1, y_1) and (x_2, y_2)

step 2: Approximate the length of the line

i.e if $(\text{abs}(x_2 - x_1) > \text{abs}(y_2 - y_1))$ then

length $= \text{abs}(x_2 - x_1)$

else

length $= \text{abs}(y_2 - y_1)$

Scanned by CamScanner

Step 3:

select the rester unit

$$\Delta x = \frac{x_2 - x_1}{\text{length}}$$

$$\Delta y = \frac{y_2 - y_1}{\text{length}}$$

Here, either Δx or Δy will be one.

Thus the incremental value for x or y will 1 (one).

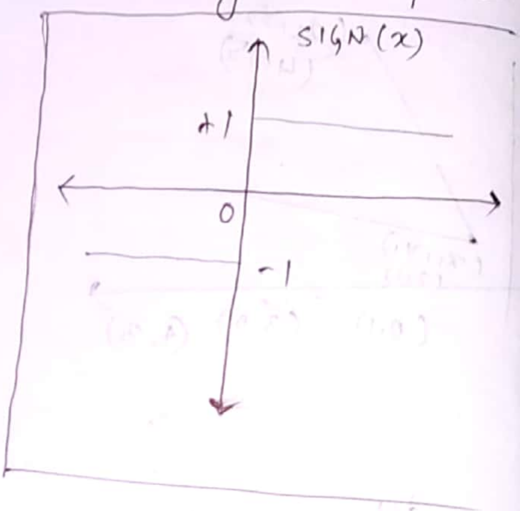
Step 4:

Round the value

Here, we use SIGN function. The SIGN function returns +1, 0, -1 depending on the argument passes through it.

$x = x_1 + 0.5 * SIGN(\Delta x)$

$y = y_1 + 0.5 * SIGN(\Delta y)$



Step 5:

Now plot the points

```
i=1;
while (i<=length)
{
    plot (integer(x), integer(y))
    x=x+Δx
    y=y+Δy
    i=i+1
}
```

Step 6:

Stop

Ex: Draw a line from (0,0) to (6,6) using DDA line drawing algorithm.

Solⁿ let us consider,

$x_1 = 0, y_1 = 0$ and $x_2 = 6, y_2 = 6$

$abs(x_2 - x_1) = 6$, $abs(y_2 - y_1) = 6$

length $= 6$.

$$\Delta x = \frac{(x_2 - x_1)}{\text{length}} = \frac{6}{6} = 1$$

$$\Delta y = \frac{(y_2 - y_1)}{\text{length}} = 1$$

$x = x + 0.5 * SIGN(x)$

$= 0 + 0.5 * 1$

$= 0.5$

$y = y + 0.5 * SIGN(y)$

$= 0.5$

(0,0) $x = x + \Delta x$ $y = y + \Delta y$

$= 0.5 + 1$ $= 0.5 + 1$

$= 1.5$ $= 1.5$

i=2

plot (1,1)

$x = x + \Delta x = 1.5 + 1 = 2.5$

$y = y + \Delta y = 2.5$

plot (2,2)

$x = x + \Delta x = 2.5 + 1 = 3.5$

$y = y + \Delta y = 3.5$

plot (3,3)

$x = x + \Delta x = 3.5 + 1 = 4.5$

$y = y + \Delta y = 4.5$

plot (4,4)

$x = x + \Delta x = 4.5 + 1 = 5.5$

$y = y + \Delta y = 5.5$

plot(5,5)

$x = x + \Delta x = 5.5 + 1 = 6.5$

$y = y + \Delta y = 6.5$

plot (6,6)

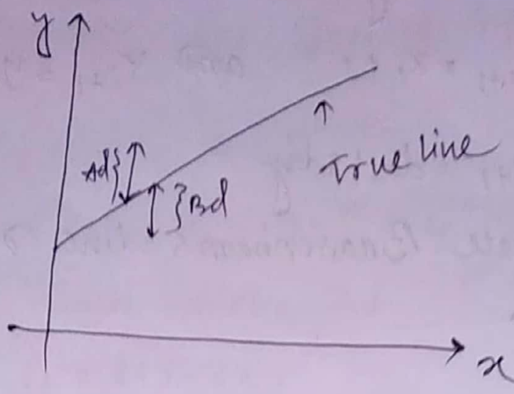
steps or

i	plot	x	y
		0.5	0.5
1	(0,0)	1.5	1.5
2	(1,1)	2.5	2.5
3	(2,2)	3.5	3.5
4	(3,3)	4.5	4.5
5	(4,4)	5.5	5.5
6	(5,5)	6.5	6.5
7			

Bresenham's line drawing algorithm — 24/01

In this algorithm the basic idea behind this algorithm is to find the decision variable or the error term. This is defined as the distance between the actual line location and the nearest pixel. The algorithm increments x or y one unit depending on the slope of the line. Ones this is done, then increment in other variable is found on the basis of error term or the decision variables.

Mathematical Description :



The line indicate the expected true line. Here we consider those pixels which are closest to the expected line.

Let, Ad = The distance of pixels which are line above the true line

Bd = The distance of pixel which are line below the true line.

Now, Let us defined a decision variable, $d_i = Bd - Ad$

Now, if $d_i < 0$ then only increment x value,
then it implies that $Ad > Bd$
i.e. Pixel below the true line is closer to the expected line.

if $d_i \geq 0$ then the pixel value above the true line is closest.

Hence, if $d_i < 0$, then we can increment only x value.
So that, we get a pixel on the line and when $d_i > 0$ increment both x and y values.

The actual compulation are explained below. We will initially said the decision variable, $d_1 = 2dy - dx$
where, $dx = x_2 - x_1$ and $dy = y_2 - y_1$

if $d_i > 0$ then x and y are incremented. So,

$x_{i+1} = x_i + 1$ and $y_{i+1} = y_i + 1$

and $d_{i+1} = d_i + 2(dy - dx)$.

if $d_i < 0$ then only increment x value

So, $x_{i+1} = x_i + 1$ and $y_{i+1} = y_i$

and, $d_{i+1} = d_i + 2dy$

So, the complete Brasenham's line drawing algorithm is as follows.

12/02
[AR]

Algorithm

step 1: Input two line end points and store the left end point in (x_0, y_0)

step 2: Load (x_0, y_0) into the frame buffer, i.e plot the first point.

step 3: Calculate constants ~~δx~~ Δx, Δy, $2\Delta y$ and $2\Delta y - 2\Delta x$ and obtain the starting value for the decision parameter as $d_0 = 2\Delta y - \Delta x$

step 4: At each x_K, along the line starting at $K=0$ perform the following if dicision ~~rate~~ parameter, $d_K < 0$ then next point to plotting (x_{K+1}, y_K) and $d_{K+1} = d_K + 2\Delta y$
otherwise the next point to plot is (x_{K+1}, y_{K+1}) and $d_{K+1} = d_K + 2\Delta y - 2\Delta x$

step 5: repeate step 4 Δx times

Problem: Draw a line with end points (10,5) and (15,9) using Bresham's line drawing Algorithm

Soln Here, $\Delta x = x_2 - x_1 = 15 - 10 = 5$

$\Delta y = y_2 - y_1 = 9 - 5 = 4$

The initial decision variable as

$$d_0 = 2\Delta y - \Delta x$$
$$= 2 \times 4 - 5$$
$$= 8 - 5$$
$$= 3 > 0$$

So plot the point (11,6)

$$d_1 = d_0 + 2(\Delta y - \Delta x)$$
$$= 3 + 2(4-5)$$
$$= 3 - 2$$
$$= 1 > 0$$

So plot the point (12,7)

$$d_2 = d_1 + 2(4-5)$$
$$= 1 - 2$$
$$= -1 < 0$$

So Plot the point (13,7)

$$d_3 = d_2 + 2(\Delta y - \Delta x)$$
$$= -1 + 8$$
$$= 7 > 0$$

So Plot the point (14,8)

$$d_4 = d_3 + 2(\Delta y - \Delta x)$$
$$= 7 + 2(4-5)$$
$$= 7 - 2$$
$$= 5 > 0$$

Plot the point (15,9)