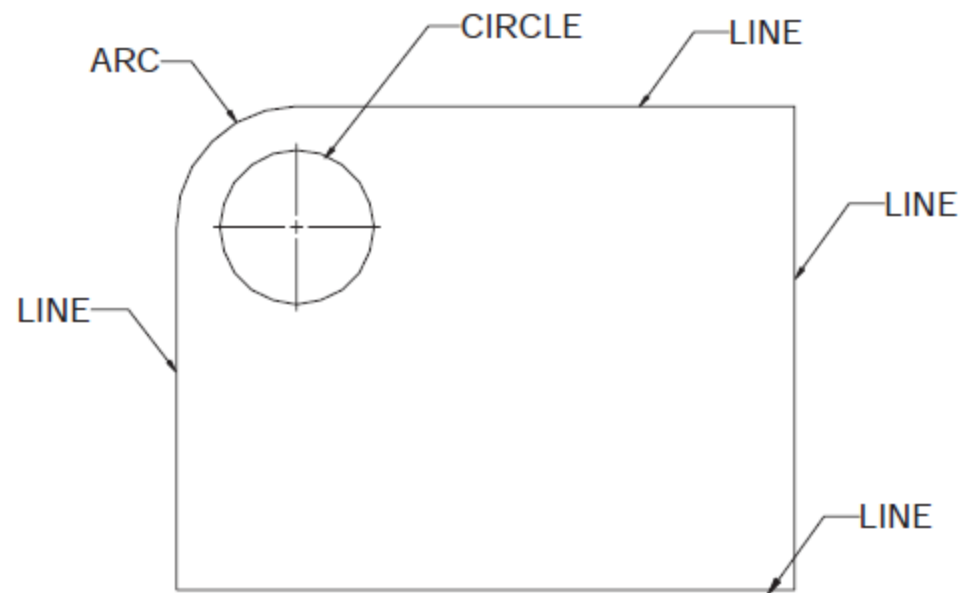


POINT PLOTTING

The frame buffer display is an example of a point plotting device. The smallest unit accepted by such displays is a single pixel. To construct a useful picture on a point plotting device, a picture must be built out of several hundreds of pixels.



DRAWING OF LINES

The following criteria have been stipulated for line drawing displays :

- i. Lines should appear straight
- ii. Lines should terminate accurately
- iii. Lines should have constant density
- iv. Line density should be independent of length and angle
- v. Line should be drawn rapidly

LINE DRAWING ALGORITHMS

The basic objective is to enable visually satisfactory images in least possible time. This is achieved by reducing the calculations to a minimum. This is by using integer arithmetic rather than floating point arithmetic. So saving of one computation per pixel will save number of computations in generating an object. This in turn minimizes the time required to generate the whole image on the screen.

DDA ALGORITHM (DIGITAL DIFFERENTIAL ANALYZER)-

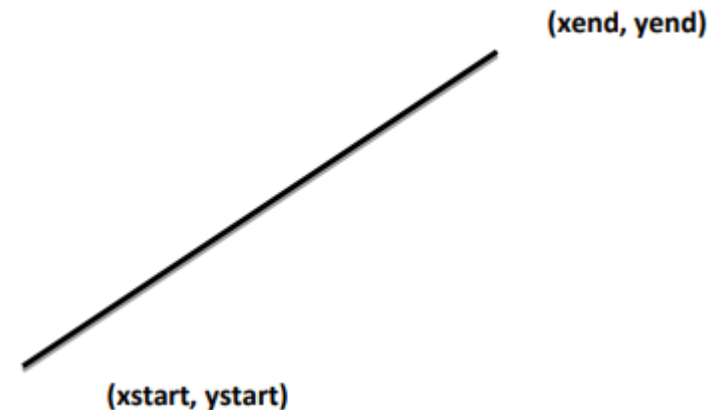
Suppose we are given the 2 end points of a line. (x_{start}, y_{start}) and (x_{end}, y_{end}) .

The general equation of a line is $y = mx + c$.

Where m is the slope of the line and c is the y - intercept.

(x, y) is any point on the line.

Slope of the above line can be expressed as:- $m = (y_{end} - y_{start}) / (x_{end} - x_{start})$



Suppose we are given 2 points on the line as (x_i, y_i) and (x_{i+1}, y_{i+1})

then also slope, $m = (y_{i+1} - y_i)/(x_{i+1} - x_i)$

We can say that, $y_{i+1} - y_i = dy$

and $x_{i+1} - x_i = dx$

Then $m = dy / dx$

That is $dy = m dx$

Thus for any given x interval dx along the line, we can compute the corresponding y interval dy.

Now suppose $x_{i+1} - x_i = 1$ or $dx = 1$

It means $x_{i+1} = x_i + 1$

Then $dy = m$

which means $y_{i+1} - y_i = m$

That is $y_{i+1} = y_i + m$

Thus a unit change in x changes y by m, which is constant for a line.

If $|m| \leq 1$, $\Delta x = 1$ if $|m| > 1$, In this case we reverse the roles of x and y by sampling at unit y intervals as,

then $x_{i+1} = x_i + 1,$

$y_{i+1} = y_i + m$

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

$x_{i+1} = x_i + 1/m,$

If $m = 1$, Then x and y coordinate tend to the Unit interval.

$$x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i + 1$$

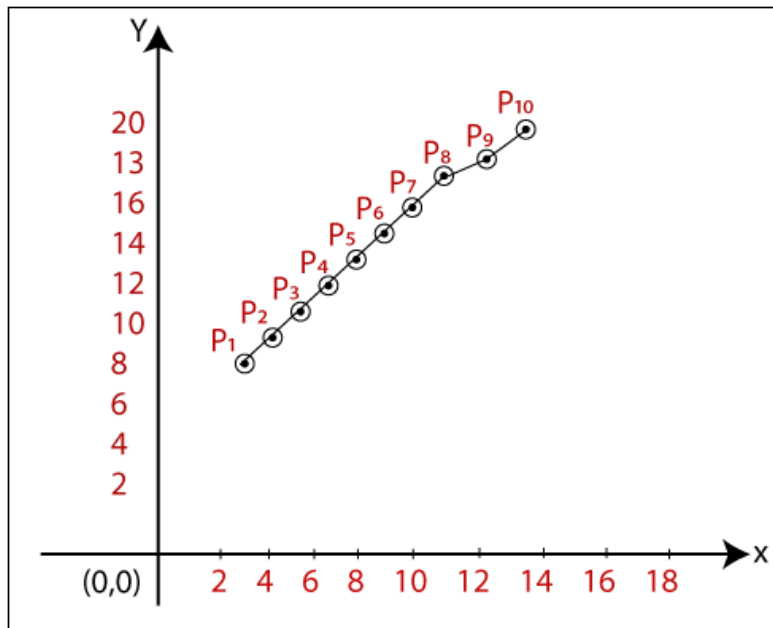
Example: A line has a starting point (1,7) and ending point (11,17).
Apply the Digital Differential Analyzer algorithm to plot a line.

$$\Delta x = x_2 - x_1 = 11 - 1 = 10$$

$$\Delta y = y_2 - y_1 = 17 - 7 = 10$$

$$m = \Delta y / \Delta x = 10 / 10 = 1$$

We get $m = 1$, Third case is satisfied



| k | x_{k+1} | y_{k+1} | (x_{k+1}, y_{k+1}) |
|----|-----------|-----------|----------------------|
| 1 | 1 | 7 | (1, 7) |
| 2 | 2 | 8 | (2, 8) |
| 3 | 3 | 9 | (3, 9) |
| 4 | 4 | 10 | (4, 10) |
| 5 | 5 | 11 | (5, 11) |
| 6 | 6 | 12 | (6, 12) |
| 7 | 7 | 13 | (7, 13) |
| 8 | 8 | 14 | (8, 14) |
| 9 | 9 | 15 | (9, 15) |
| 10 | 10 | 16 | (10, 16) |
| 11 | 11 | 17 | (11, 17) |

Ex.2 Calculate the points between the starting point (5, 6) and ending point (13, 10).

$$\Delta X = X_n - X_0 = 13 - 5 = 8$$

$$\Delta Y = Y_n - Y_0 = 10 - 6 = 4$$

$$M = \Delta Y / \Delta X = 4 / 8 = 0.50$$

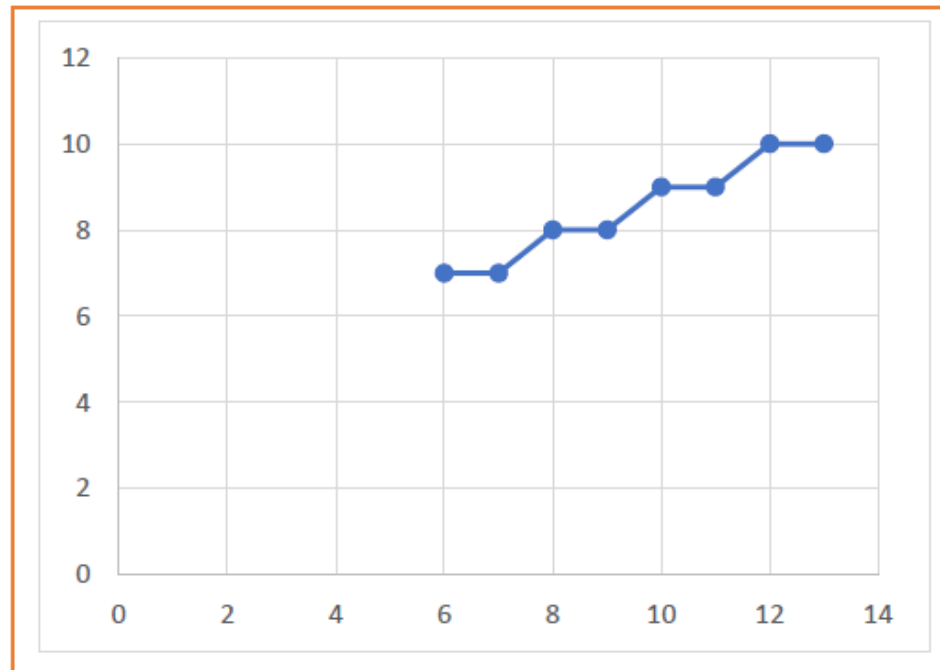
As $M < 1$, so case-01 is satisfied.

$$\Delta x = 1$$

$$x_{i+1} = x_i + 1,$$

then $y_{i+1} = y_i + m$

| i | x_p | y_p |
|---|-------|-------|
| 1 | 5 | 6 |
| 2 | 6 | 7 |
| 3 | 7 | 7 |
| 4 | 8 | 8 |
| 5 | 9 | 8 |
| 6 | 10 | 9 |
| 7 | 11 | 9 |
| 8 | 12 | 10 |
| 9 | 13 | 10 |



Advantages of DDA Algorithm- The advantages of DDA Algorithm are-

- It is a simple algorithm.
- It is easy to implement.
- It avoids using the multiplication operation which is costly in terms of time complexity.

Disadvantages of DDA Algorithm- The disadvantages of DDA Algorithm are-

- There is an extra overhead of using round off() function.
- Using round off() function increases time complexity of the algorithm.
- Resulted lines are not smooth because of round off() function.
- The points generated by this algorithm are not accurate.

Bresenham's Line Drawing algorithm- introduced by “**Jack Elton Bresenham**” in **1962**. This algorithm helps us to perform [scan conversion](#) of a line. It is a powerful, useful, and accurate method. We use incremental integer calculations to draw a line. The integer calculations include addition, subtraction, and multiplication.

let, we have initial coordinates of a line = (x_k, y_k)

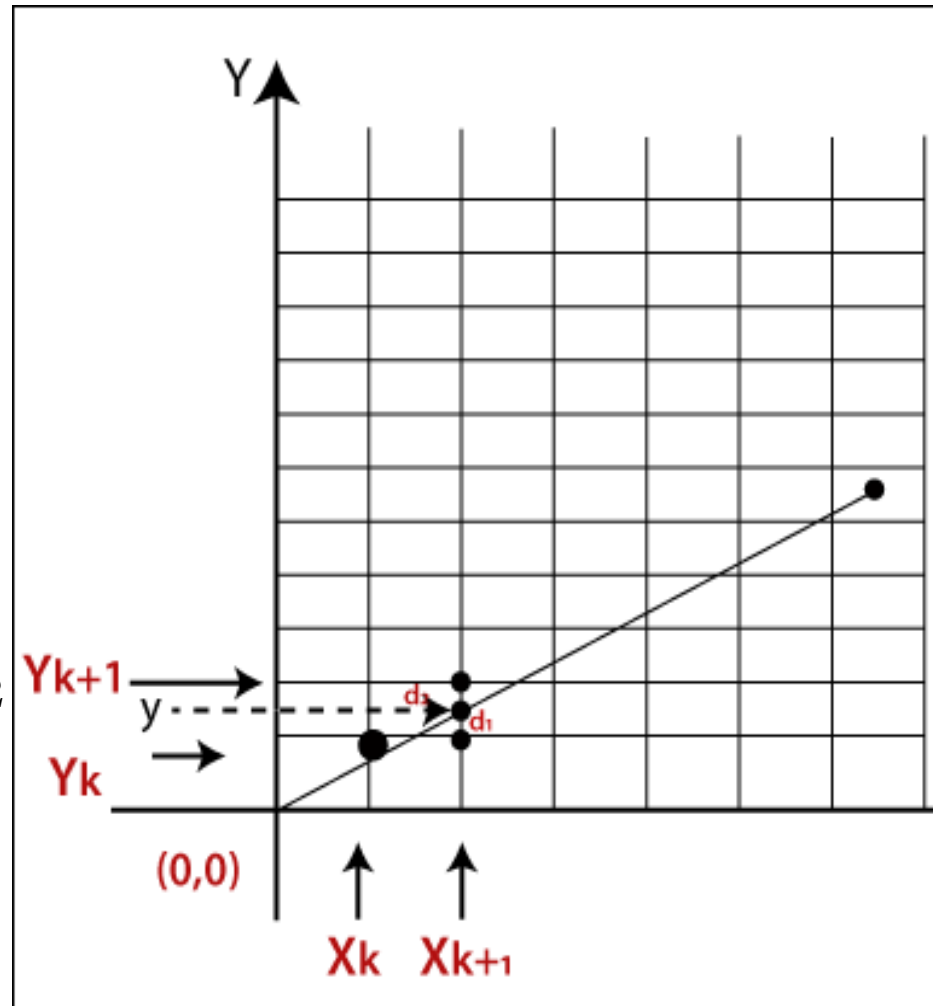
The next coordinates of a line = (x_{k+1}, y_{k+1})

The intersection point between y_k and $y_{k+1} = y$

Let we assume that the distance between y and $y_k = d_1$

The distance between y and $y_{k+1} = d_2$

Now, we have to decide which point is nearest to the intersection point.



Case (A) If $m < 1$
 then $x = x_k + 1$ { Unit Interval}
 $y = y_k + 1$ { Unit Interval}

As we know the equation of a line- $y = mx + b$

Now we put the value of x into the line equation, then

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

The value of $d_1 = y - y_k$

Now we put the value of d_1 in equation (1).

$$d_1 = m(x_k + 1) + b - y_k \dots\dots\dots (2)$$

Now, we again put the value of y in the previous equation then we got,

$$d_2 = y_{k+1} - y$$

$$= y_k + 1 - m(x_k + 1) - b \dots\dots\dots (3)$$

Now, we calculate the difference between $d_1 - d_2$

$$(d_1 - d_2) = m(x_k + 1) + b - y_k - y_k - 1 + m(x_k + 1) + b$$

now replaced the m with $\Delta y / \Delta x$

$$(d_1 - d_2) = 2m(x_{k+1}) - 2y_k + 2b - 1$$

We multiplied Δx at both side then we got,

$$\Delta x (d_1 - d_2) = \Delta x (2m(x_{k+1}) - 2y_k + 2b - 1)$$

We consider $\Delta \mathbf{x} (\mathbf{d}_1 - \mathbf{d}_2)$ as a decision parameter (\mathbf{d}_k), so

$$\mathbf{d}_k = \Delta \mathbf{x} (\mathbf{d}_1 - \mathbf{d}_2)$$

After calculation we got,

$$\mathbf{d}_k = 2 \Delta \mathbf{y} \mathbf{x}_k + 2 \Delta \mathbf{y} - 2 \Delta \mathbf{x} \mathbf{y}_k + \Delta \mathbf{x} (2\mathbf{b}-1)$$

Then, the next coordinate of \mathbf{d}_k

$$\mathbf{d}_{k+1} = 2 \Delta \mathbf{y} \mathbf{x}_{k+1} + 2 \Delta \mathbf{y} - 2 \Delta \mathbf{x} \mathbf{y}_{k+1} + \Delta \mathbf{x} (2\mathbf{b}-1)$$

Now, the difference between $\mathbf{d}_{k+1} - \mathbf{d}_k$ then,

$$\mathbf{d}_{k+1} - \mathbf{d}_k = 2 \Delta \mathbf{y} (\mathbf{x}_{k+1} - \mathbf{x}_k) - 2 \Delta \mathbf{x} (\mathbf{y}_{k+1} - \mathbf{y}_k)$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k + 2 \Delta \mathbf{y} (\mathbf{x}_{k+1} - \mathbf{x}_k) - 2 \Delta \mathbf{x} (\mathbf{y}_{k+1} - \mathbf{y}_k) \{\text{Decision parameter coordinate}\}$$

Now, we put the value of \mathbf{x}_{k+1} in above equation then we got,

$$\mathbf{d}_{k+1} = \mathbf{d}_k + 2 \Delta \mathbf{y} - 2 \Delta \mathbf{x} (\mathbf{y}_{k+1} - \mathbf{y}_k) \{\text{New decision parameter when } m < 1\}$$

Similarly, if $m > 1$, the new decision parameter for next coordinate will be

$$\mathbf{d}_{k+1} = \mathbf{d}_k + 2 \Delta \mathbf{y} - 2 \Delta \mathbf{x} (\mathbf{x}_{k+1} - \mathbf{x}_k) \{\text{New decision parameter when } m > 1\}$$

For If $m < 1$

a) If $d_k \geq 0$ {For coordinate y}

Then,

$y_{k+1} = y_k + 1$ {We will choose the nearest y_{k+1} pixel}

The next coordinate will be $(x_k + 1, y_k + 1)$

$$dk+1 = dk + 2(dy - dx)$$

b) If $d_k < 0$

Then,

$y_{k+1} = y_k$ {We will choose the nearest y_k pixel}

The next coordinate will be $(x_k + 1, y_k)$

$$dk+1 = dk + 2(dy)$$

For when $m > 1$

a) If $d_k \geq 0$ {For coordinate x}

Then,

$x_{k+1} = x_k + 1$ {We will choose the nearest x_{k+1} pixel}

The next coordinate will be $(x_k + 1, y_k + 1)$

$$dk+1 = dk + 2(dx - dy)$$

b) If $d_k < 0$

Then,

$x_{k+1} = x_k$ {We will choose the nearest x_k pixel}

The next coordinate will be $(x_k, y_k + 1)$

$$dk+1 = dk + 2(dx)$$

For If $\mathbf{m} < \mathbf{1}$

we will calculate the decision parameter \mathbf{d}_k with following formula.

$$\mathbf{d}_1 = \mathbf{d}_k = 2 \blacktriangle \mathbf{y} - \blacktriangle \mathbf{x}$$

Case 1: If

$$\mathbf{d}_k < 0$$

Then

$$\mathbf{d}_{k+1} = \mathbf{d}_k + 2 \blacktriangle \mathbf{y}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 1$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k$$

Case 2: If

$$\mathbf{d}_k \geq 0$$

Then

$$\mathbf{d}_{k+1} = \mathbf{d}_k + 2 \blacktriangle \mathbf{y} - 2 \blacktriangle \mathbf{x}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 1$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + 1$$

For If $\mathbf{m} > \mathbf{1}$

we will calculate the decision parameter \mathbf{d}_k with following formula.

$$\mathbf{d}_1 = \mathbf{d}_k = 2 \blacktriangle \mathbf{x} - \blacktriangle \mathbf{y}$$

Case 1: If

$$\mathbf{d}_k < \mathbf{0}$$

Then

$$\mathbf{d}_{k+1} = \mathbf{d}_k + 2 \blacktriangle \mathbf{x}$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{1}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k$$

Case 2: If

$$\mathbf{d}_k \geq \mathbf{0}$$

Then

$$\mathbf{d}_{k+1} = \mathbf{d}_k + 2 \blacktriangle \mathbf{x} - 2 \blacktriangle \mathbf{y}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{1}$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{1}$$

Advantages of Bresenham's Line Drawing Algorithm

- It is simple to implement because it only contains integers.
- It is quick and incremental
- It is fast to apply but not faster than the Digital Differential Analyzer (DDA) algorithm.
- The pointing accuracy is higher than the [DDA algorithm](#)

Disadvantages of Bresenham's Line Drawing Algorithm

- The *Bresenham's Line drawing algorithm* only helps to draw the basic line.
- The resulted draw line is not smooth

Example: A line has a starting point (9,18) and ending point (14,22).

Apply the Bresenham's Line Drawing algorithm to plot a line.

$$\begin{aligned}\Delta x &= x_2 - x_1 \\ &= 14 - 9 = 5\end{aligned}$$

$$\begin{aligned}\Delta y &= y_2 - y_1 \\ &= 22 - 18 = 4\end{aligned}$$

$$m = 4/5 = 0.8$$

to calculate the decision parameter (d_k)

$$d_k = 2\Delta y - \Delta x = 2 \times 4 - 5 = 3$$

we will check both the cases.

$$\text{If } d_k \geq 0$$

Case 2 is satisfied. Thus

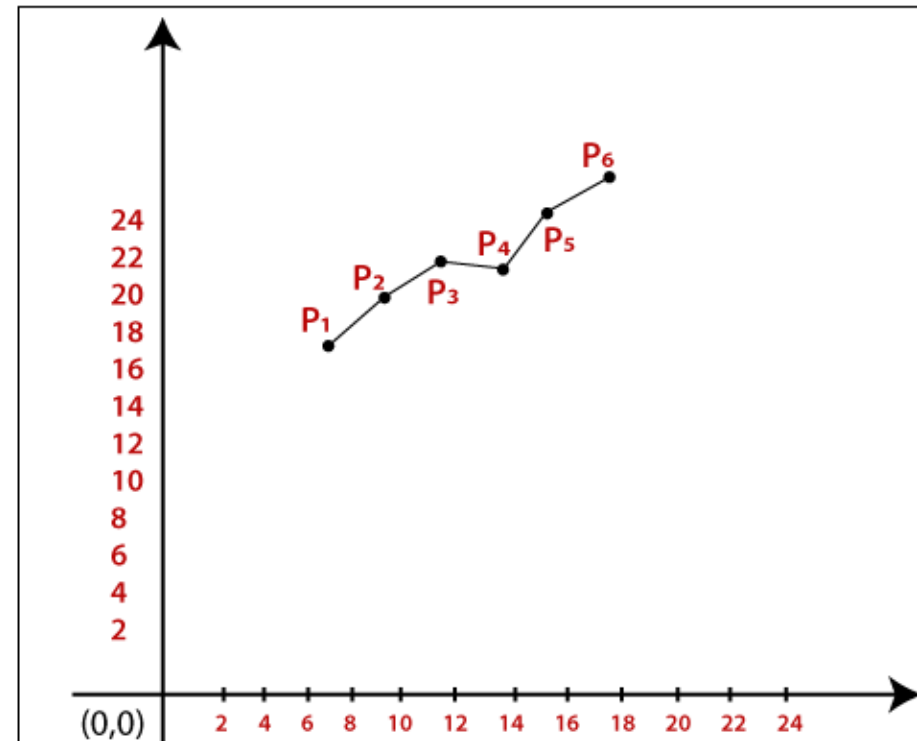
$$\begin{aligned}d_{k+1} &= d_k + 2\Delta y - 2\Delta x \\ &= 3 + (2 \times 4) - (2 \times 5) = 1\end{aligned}$$

$$x_{k+1} = x_k + 1 = 9 + 1 = 10$$

$$y_{k+1} = y_k + 1 = 18 + 1 = 19$$

Now move to next step. We will calculate the coordinates until we reach the end point of the line

| P_k | x_{k+1} | y_{k+1} |
|-------|-----------|-----------|
| | 9 | 18 |
| 3 | 10 | 19 |
| 1 | 11 | 20 |
| -1 | 12 | 20 |
| 7 | 13 | 21 |
| 5 | 14 | 22 |



Problem-01:

Calculate the points between the starting coordinates (9, 18) and ending coordinates (14, 22).

Solution-

Given-

Starting coordinates = $(X_0, Y_0) = (9, 18)$

Ending coordinates = $(X_n, Y_n) = (14, 22)$

Step-01:

Calculate ΔX and ΔY from the given input.

$$\Delta X = X_n - X_0 = 14 - 9 = 5$$

$$\Delta Y = Y_n - Y_0 = 22 - 18 = 4$$

Step-02:

Calculate the decision parameter.

$$\begin{aligned}d_k &= 2\Delta Y - \Delta X \\ &= 2 \times 4 - 5 \\ &= 3\end{aligned}$$

So, decision parameter $P_k = 3$

Step-03:

As $d_k \geq 0$, so case-02 is satisfied.

Thus,

$$d_{k+1} = d_k + 2\Delta Y - 2\Delta X = 3 + (2 \times 4) - (2 \times 5) = 1$$

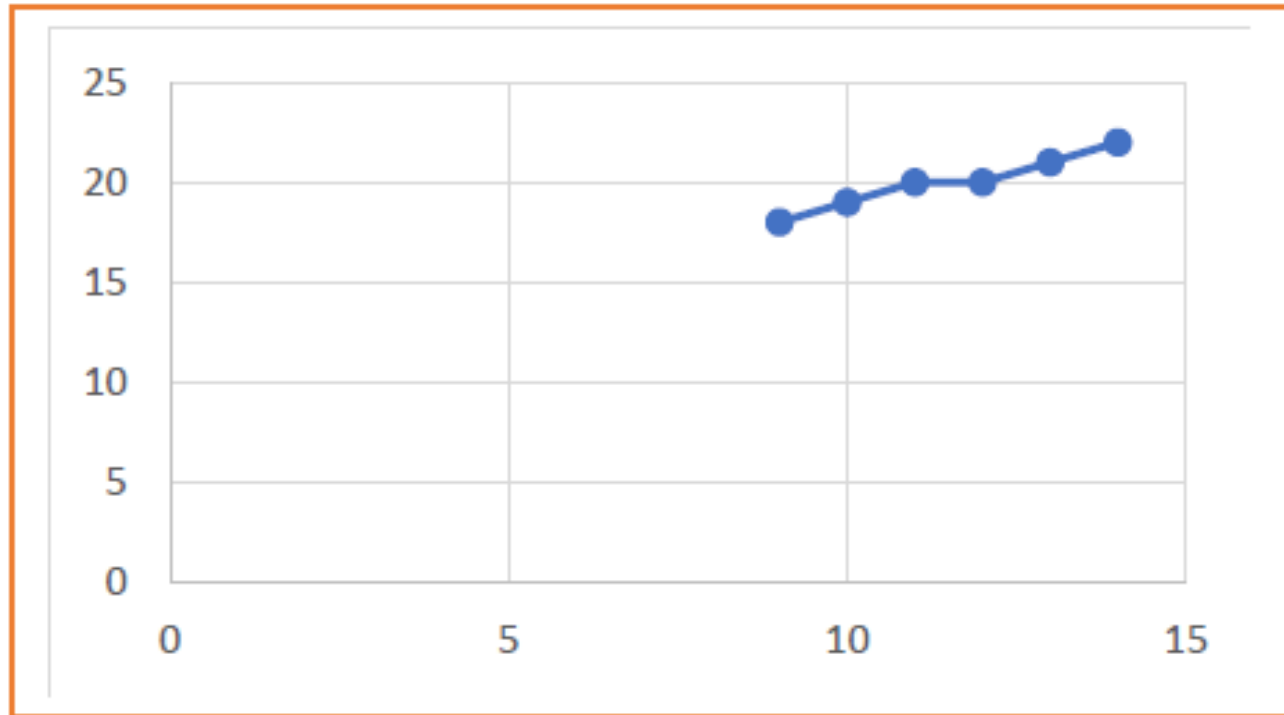
$$X_{k+1} = X_k + 1 = 9 + 1 = 10$$

$$Y_{k+1} = Y_k + 1 = 18 + 1 = 19$$

Similarly, Step-03 is executed until the end point is reached or number of iterations equals to 4 times.

$$(\text{Number of iterations} = \Delta X - 1 = 5 - 1 = 4)$$

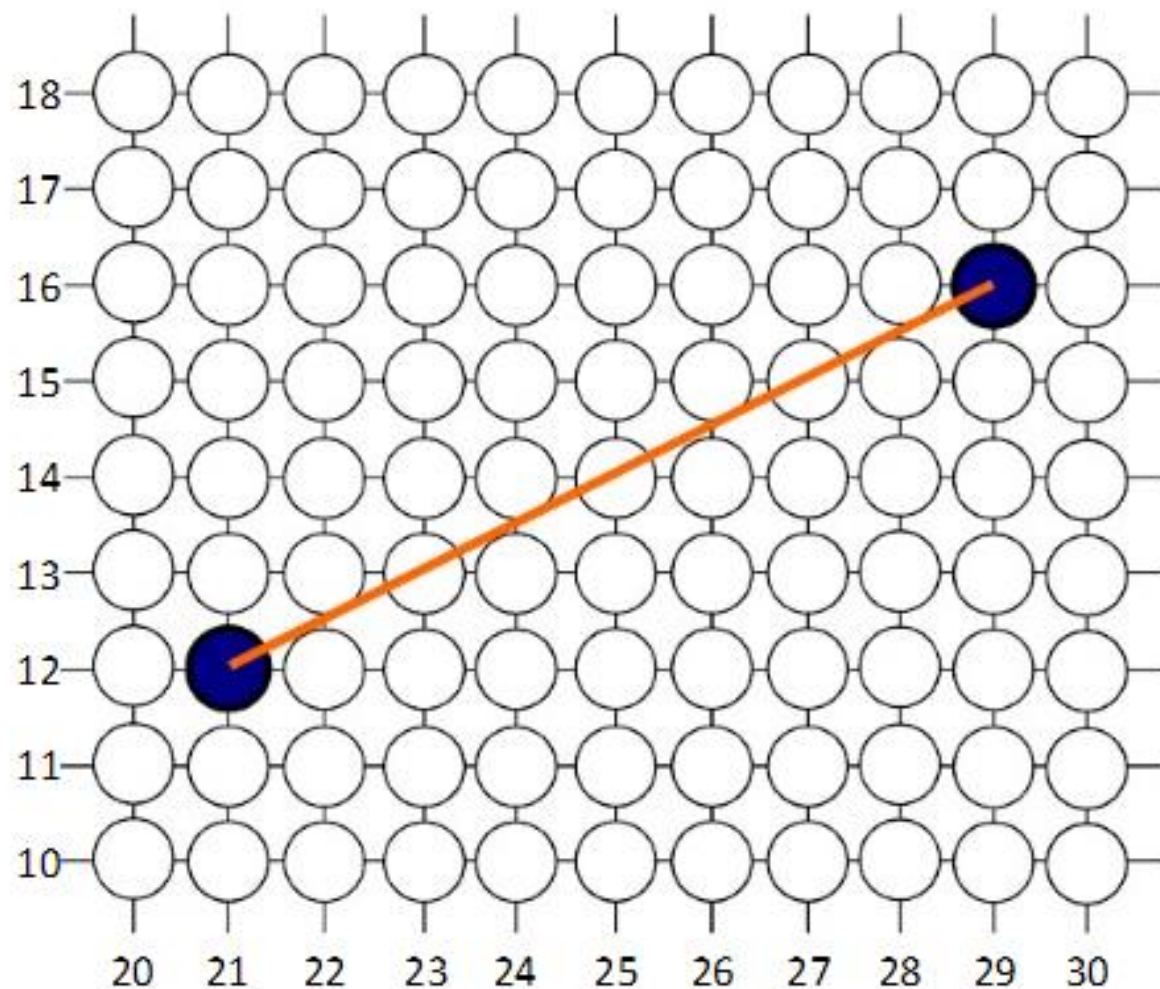
| d_k | d_{k+1} | X_{k+1} | Y_{k+1} |
|-------|-----------|-----------|-----------|
| | | 9 | 18 |
| 3 | 1 | 10 | 19 |
| 1 | -1 | 11 | 20 |
| -1 | 7 | 12 | 20 |
| 7 | 5 | 13 | 21 |
| 5 | 3 | 14 | 22 |



Ex:2. Let the given end points for the line be (30,20) and (40, 28)

- $M = \frac{dy}{dx} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{28 - 20}{40 - 30} = \frac{8}{10} = 0.8$
- $dy = 8$ and $dx = 10$
- The initial decision parameter P_0 is
- $d_0 = 2dy - dx = 2(8) - 10 = 16 - 10 = 6$
- $d_0 = 6$
- The constants $2dy$ and $2dy-2dx$ are
- $2dy = 2(8) = 16$ $2dy-2dx = 2(8) - 2(10) = 16 - 20 = -4$
- $2dy = 16$ $2dy - 2dx = -4$

Bresenham Exercise (cont...)



| k | p_k | (x_{k+1}, y_{k+1}) |
|---|-------|----------------------|
| 0 | | |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |

The starting point $(x_0, y_0)=(30,20)$ and the successive pixel positions are given in the following table

| K | d_k | (X_{k+1}, Y_{k+1}) |
|---|-------|----------------------|
| - | - | 30,20 |
| 0 | 6 | 31,21 |
| 1 | 2 | 32,22 |
| 2 | -2 | 33,22 |
| 3 | 14 | 34,23 |
| 4 | 10 | 35,24 |
| 5 | 6 | 36,25 |
| 6 | 2 | 37,26 |
| 7 | -2 | 38,26 |
| 8 | 14 | 39,27 |
| | 10 | 40,28 |

- End points (30,20) (40,28)
- $dx=x_2-x_1=40-30=10$ $dy=y_2-y_1=28-20=8$
- $d_0=2dy-dx=2 \times 8-10 = 6 > 0$ pt(31,21)
- $dk+1= dk+2dy-2dx= 6+16-20 = 2 > 0$ (32,22)
- $dk+1=2+16-20 = -2 < 0$ (33,22)
- $dk+1= dk+2dy = -2+16 = 14 > 0$ (34,23)
- $dk+1= dk+2dy-2dx=14+16-20=10 > 0$ (35,24)
- $dk+1=10+16-20=6 > 0$ (36,25)
- $dk+1= 6+16-20=2 > 0$ (37,26)
- $dk+1=2+16-20=-2 < 0$ (38,26)
- $dk+1= dk+2dy=-2+16=14 > 0$ (39,27)
- $dk+1=dk+2dy-2dx= 14+16-20=10 > 0$ (40,28)