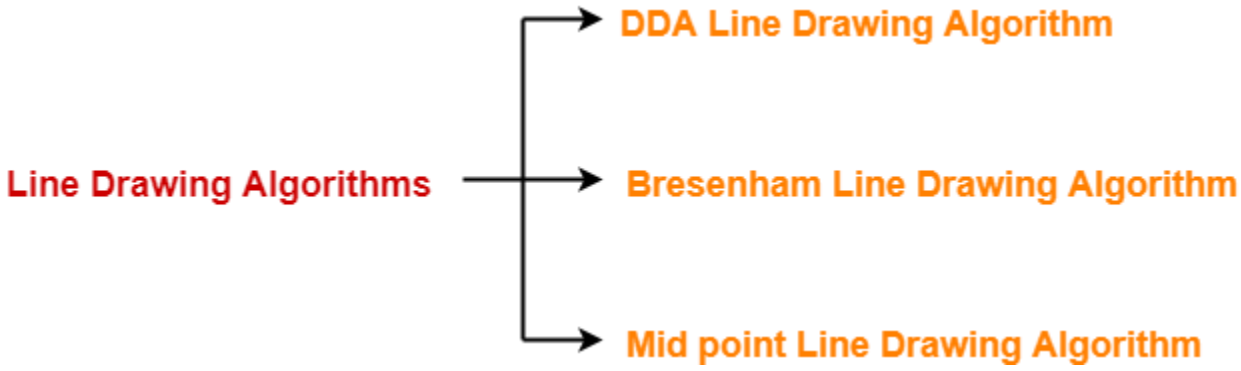
Line Drawing Algorithms-

In computer graphics, popular algorithms used to generate lines are-



1. Digital Differential Analyzer (DDA) Line Drawing Algorithm
2. Bresenham Line Drawing Algorithm
3. Mid Point Line Drawing Algorithm

In this article, we will discuss about DDA Algorithm.

DDA Algorithm-

DDA Algorithm is the simplest line drawing algorithm.

Given the starting and ending coordinates of a line, DDA Algorithm attempts to generate the points between the starting and ending coordinates.

Procedure-

Given-

- Starting coordinates = (X_0, Y_0)
- Ending coordinates = (X_n, Y_n)

The points generation using DDA Algorithm involves the following steps-

Step-01:

Calculate ΔX, ΔY and M from the given input.

These parameters are calculated as-

- $\Delta X = X_n - X_0$
- $\Delta Y = Y_n - Y_0$
- $M = \Delta Y / \Delta X$

Step-02:

Find the number of steps or points in between the starting and ending coordinates.

if (absolute (ΔX) > absolute (ΔY))

Steps = absolute (ΔX);

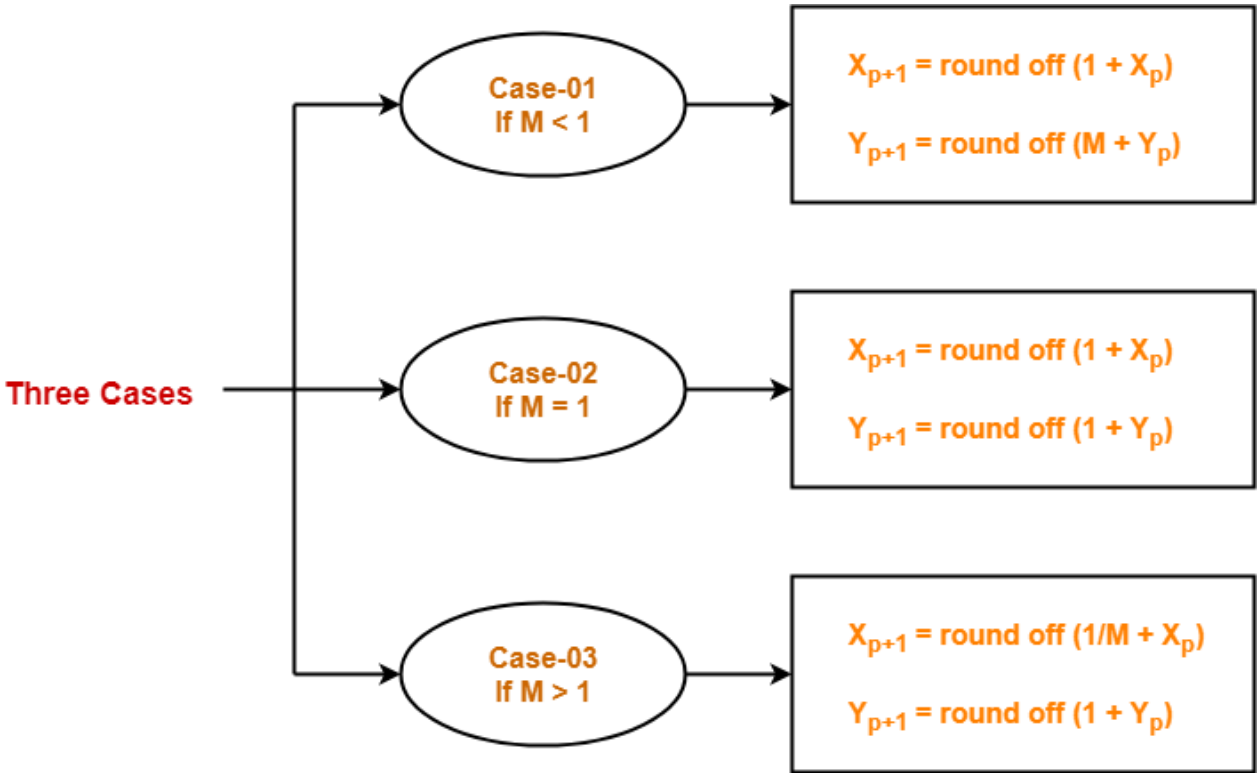
else

Steps = absolute (ΔY);

Step-03:

Suppose the current point is (X_p, Y_p) and the next point is (X_{p+1}, Y_{p+1}).

Find the next point by following the below three cases-



Step-04:

Keep repeating Step-03 until the end point is reached or the number of generated new points (including the starting and ending points) equals to the steps count.

PRACTICE PROBLEMS BASED ON DDA ALGORITHM-

Problem-01:

Calculate the points between the starting point (5, 6) and ending point (8, 12).

Solution-

Given-

- Starting coordinates = $(X_0, Y_0) = (5, 6)$
- Ending coordinates = $(X_n, Y_n) = (8, 12)$

Step-01:

Calculate ΔX, ΔY and M from the given input.

- $\Delta X = X_n - X_0 = 8 - 5 = 3$
- $\Delta Y = Y_n - Y_0 = 12 - 6 = 6$
- $M = \Delta Y / \Delta X = 6 / 3 = 2$

Step-02:

Calculate the number of steps.

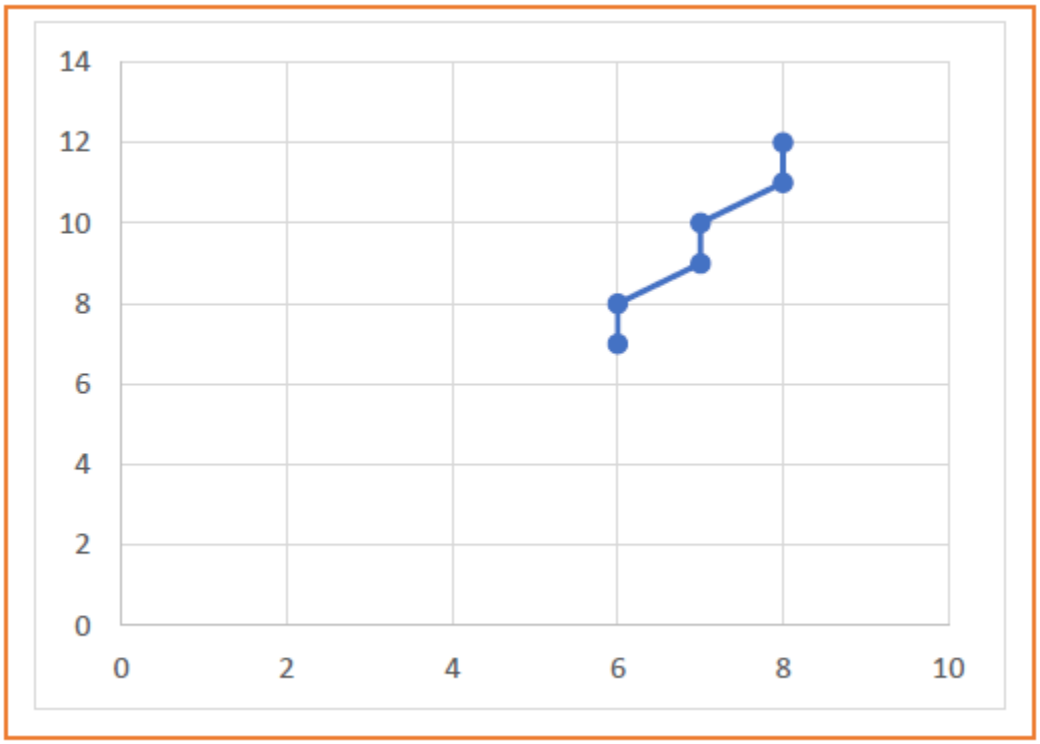
As $|\Delta X| < |\Delta Y| = 3 < 6$, so number of steps = $\Delta Y = 6$

Step-03:

As M > 1, so case-03 is satisfied.

Now, Step-03 is executed until Step-04 is satisfied.

X_p	Y_p	X_{p+1}	Y_{p+1}	**Round off (X_{p+1}, Y_{p+1})**
5	6	5.5	7	(6, 7)
		6	8	(6, 8)
		6.5	9	(7, 9)
		7	10	(7, 10)
		7.5	11	(8, 11)
		8	12	(8, 12)



Problem-02:

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

Solution-

Given-

- Starting coordinates = $(X_0, Y_0) = (5, 6)$
- Ending coordinates = $(X_n, Y_n) = (13, 10)$

Step-01:

Calculate ΔX, ΔY and M from the given input.

- $\Delta X = X_n - X_0 = 13 - 5 = 8$
- $\Delta Y = Y_n - Y_0 = 10 - 6 = 4$
- $M = \Delta Y / \Delta X = 4 / 8 = 0.50$

Step-02:

Calculate the number of steps.

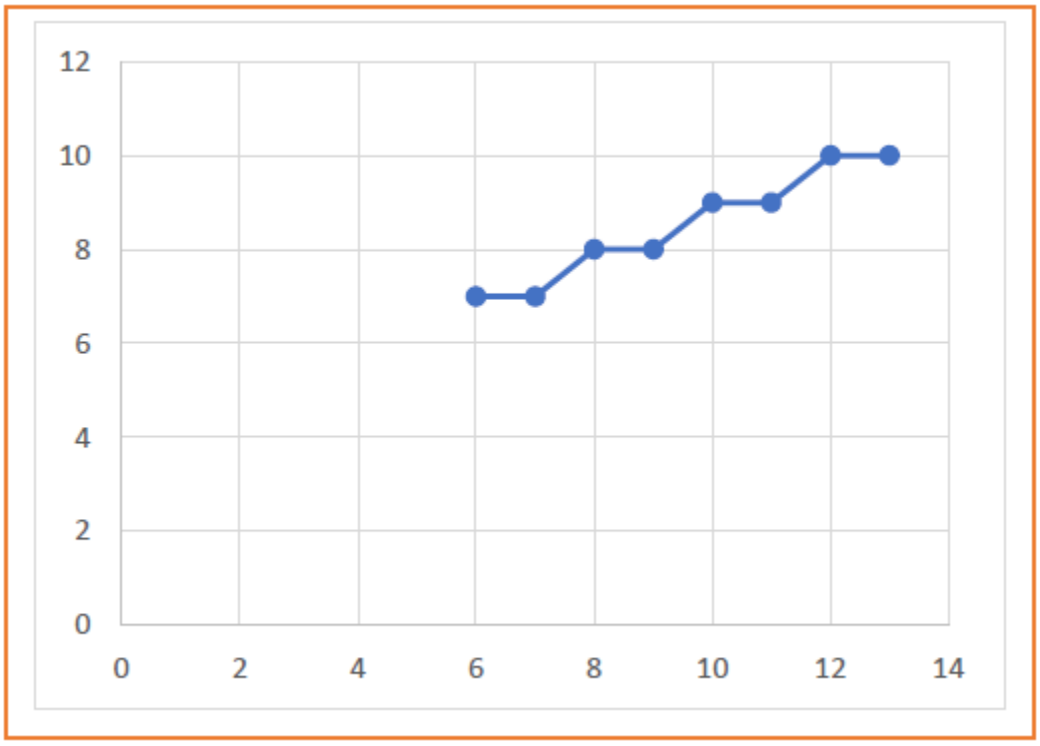
As $|\Delta X| > |\Delta Y| = 8 > 4$, so number of steps = $\Delta X = 8$

Step-03:

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

Now, Step-03 is executed until Step-04 is satisfied.

X_p	Y_p	X_{p+1}	Y_{p+1}	Round off (X_{p+1}, Y_{p+1})
5	6	6	6.5	(6, 7)
		7	7	(7, 7)
		8	7.5	(8, 8)
		9	8	(9, 8)
		10	8.5	(10, 9)
		11	9	(11, 9)
		12	9.5	(12, 10)
		13	10	(13, 10)



Problem-03:

Calculate the points between the starting point (1, 7) and ending point (11, 17).

Solution-

Given-

- Starting coordinates = $(X_0, Y_0) = (1, 7)$
- Ending coordinates = $(X_n, Y_n) = (11, 17)$

Step-01:

Calculate ΔX, ΔY and M from the given input.

- $\Delta X = X_n - X_0 = 11 - 1 = 10$
- $\Delta Y = Y_n - Y_0 = 17 - 7 = 10$
- $M = \Delta Y / \Delta X = 10 / 10 = 1$

Step-02:

Calculate the number of steps.

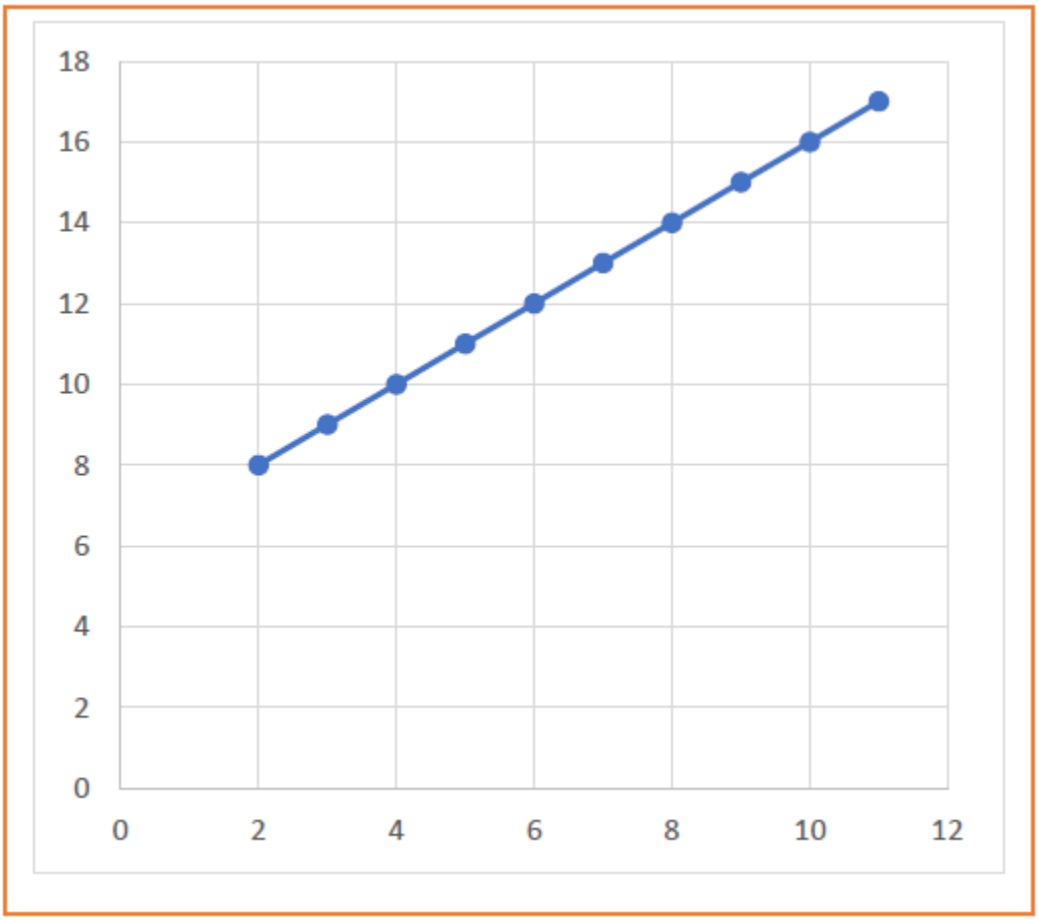
As $|\Delta X| = |\Delta Y| = 10 = 10$, so number of steps = $\Delta X = \Delta Y = 10$

Step-03:

As M = 1, so case-02 is satisfied.

Now, Step-03 is executed until Step-04 is satisfied.

X_p	Y_p	X_{p+1}	Y_{p+1}	Round off (X_{p+1}, Y_{p+1})
1	7	2	8	(2, 8)
		3	9	(3, 9)
		4	10	(4, 10)
		5	11	(5, 11)
		6	12	(6, 12)
		7	13	(7, 13)
		8	14	(8, 14)
		9	15	(9, 15)
		10	16	(10, 16)
		11	17	(11, 17)



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.

This algorithm was 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.

In **Bresenham's Line Drawing algorithm**, we have to calculate the slope (**m**) between the starting point and the ending point.

As shown in the above figure 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.

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

$$y = m(x_k + 1) + b \quad \text{..... (1)}$$

The value of $d_1 = y - y_k$

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

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

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

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

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

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

If $d_1 < d_2$

Then $y_{k+1} = y_k$ {we will choose the lower pixel as shown in figure}

If $d_1 => d_2$

Then $y_{k+1} = y_k+1$ {we will choose the upper pixel as shown in figure}

Now, we calculate the values of $d_1 - d_2$

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

We simplified the above equation and replaced the **m** with $\Delta y/\Delta x$.

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

We multiplied **x** at both side then we got,

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

We consider $\Delta x (d_1 - d_2)$ as a decision parameter (P_k), so

$p_k = \Delta x (d_1 - d_2)$

After calculation we got,

$P_k = 2\Delta y x_k + 2\Delta y - 2\Delta x y_k + \Delta x (2b-1)$

Then, the next coordinate of p_k

$p_{k+1} = 2\Delta y x_{k+1} + 2\Delta y - 2\Delta x y_{k+1} + \Delta x (2b-1)$

Now, the difference between $p_{k+1} - p_k$ then,

$p_{k+1} - p_k = 2\Delta y (x_{k+1}-x_k) - 2\Delta x (y_{k+1}-y_k)$

$p_{k+1} = p_k + 2\Delta y (x_{k+1}-x_k) - 2\Delta x (y_{k+1}-y_k)$ **{Decision parameter coordinate}**

Now, we put the value of x_{k+1} in above equation then we got,

$p_{k+1} = p_k + 2\Delta y - 2\Delta x (y_{k+1} - y_k)$ **{New decision parameter when m <1}**

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

$p_{k+1} = p_k + 2\Delta y - 2\Delta x (x_{k+1} - x_k)$ {New decision parameter when m >1}

If **$p_k >= 0$** **{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)**

If **$p_k < 0$**

Then,

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

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

Similarly,

If **$p_k >= 0$** **{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)**

If **$p_k < 0$**

Then,

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

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

Algorithm of Bresenham's Line Drawing Algorithm

Step 1: Start.

Step 2: Now, we consider Starting point as **(x_1, y_1)** and endingpoint **(x_2, y_2).**

Step 3: Now, we have to calculate **Δx** and **Δy.**

$\Delta x = x_2 - x_1$

$\Delta y = y_2 - y_1$

$m = \Delta y / \Delta x$

Step 4: Now, we will calculate the decision parameter p_k with following formula.

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

Step 5: Theinitial coordinates of the line are (x_k, y_k), and the next coordinatesare (x_{k+1}, y_{k+1}). Now, we are going to calculate two cases for decision parameter p_k

Case 1: If

$p_k < 0$

Then

$p_{k+1} = p_k + 2\Delta y$

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

$y_{k+1} = y_k$

Case 2: If

$p_k >= 0$

Then

$p_{k+1} = p_k + 2\Delta y - 2\Delta x$

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

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

Step 6: We will repeat step 5 until we found the ending point of the line and the total number of iterations =$\Delta x - 1$.

Step 7: Stop.

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.

Solution: We have two coordinates,

Starting Point = (x_1, y_1) = **(9,18)**

Ending Point = (x_2, y_2) = **(14,22)**

Step 1: First, we calculate **Δx**, **Δy.**

$\Delta x = x_2 - x_1 = 14-9 = 5$

$\Delta y = y_2 - y_1 = 22-18 = 4$

Step 2: Now, we are going to calculate the decision parameter (p_k)

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

$= 2 \times 4 - 5 = 3$

The value of $p_k = 3$

Step 3: Now, we will check both the cases.

If

$p_k >= 0$

Then

Case 2 is satisfied. Thus

$p_{k+1} = p_k + 2\Delta y-2\Delta x = 3 + (2 \times 4) - (2 \times 5) = 1$

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

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

Step 4: Now move to next step. We will calculate the coordinates until we reach the end point of the line.

$\Delta x - 1 = 5 - 1 = 4$

p_k	p_{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

Step 5: Stop.

The Coordinates of drawn lines are-

P_1 = (9, 18)

P_2 = (10, 19)

P_3 = (11, 20)

P_4 = (12, 20)

P_5 = (13, 21)

P_6 = (14, 22)

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.

Bresenham Line Drawing Algorithm:

Procedure-

Given-

- Starting coordinates = (X_0, Y_0)
- Ending coordinates = (X_n, Y_n)

The points generation using Bresenham Line Drawing Algorithm involves the following steps-

Step-01:

Calculate ΔX and ΔY from the given input.

These parameters are calculated as-

- $\Delta X = X_n - X_0$
- $\Delta Y = Y_n - Y_0$

Step-02:

Calculate the decision parameter P_k.

It is calculated as-

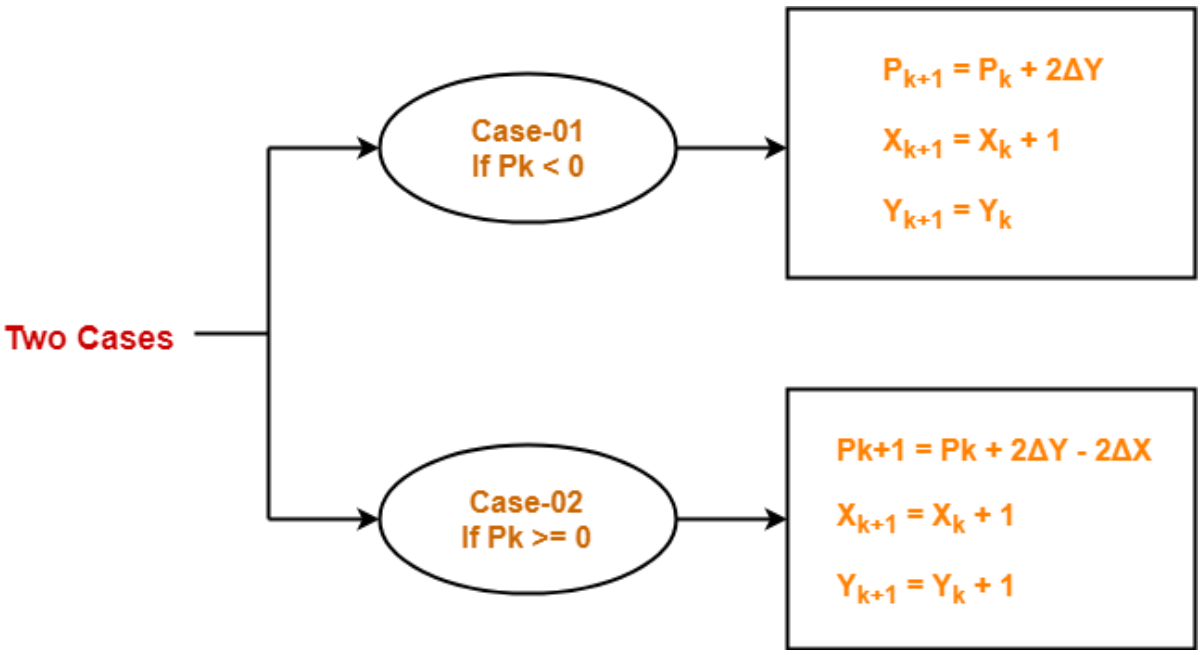
$$P_k = 2\Delta Y - \Delta X$$

Step-03:

Suppose the current point is (X_k, Y_k) and the next point is (X_{k+1}, Y_{k+1}).

Find the next point depending on the value of decision parameter P_k.

Follow the below two cases-



Step-04:

Keep repeating Step-03 until the end point is reached or number of iterations equals to $(\Delta X-1)$ times.

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.

P_k

$= 2\Delta Y - \Delta X$

$= 2 \times 4 - 5$

$= 3$

So, decision parameter $P_k = 3$

Step-03:

As $P_k >= 0$, so case-02 is satisfied.

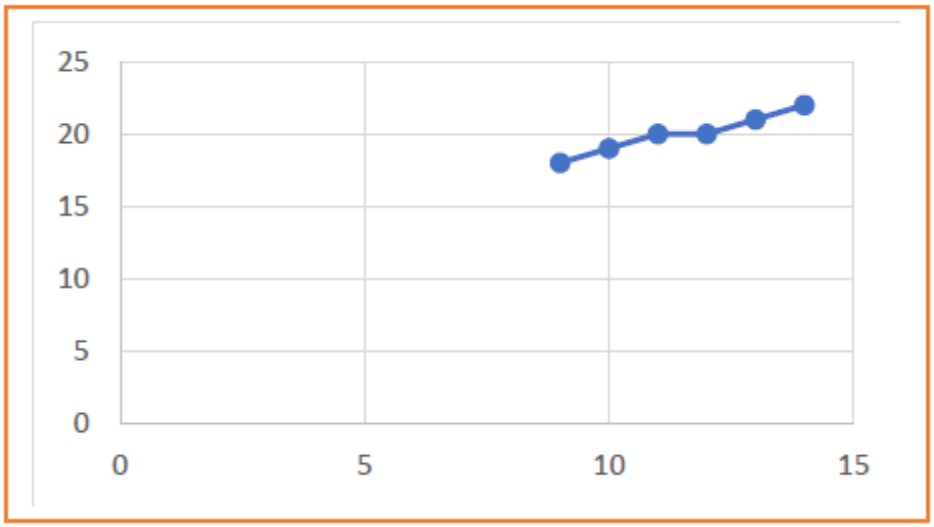
Thus,

- $P_{k+1} = P_k + 2\Delta Y - 2\Delta X = 3 + (2 \times 4) - (2 \times 5) = 1$
- $X_{k+1} = X_k + 1 = 9 + 1 = 10$
- $Y_{k+1} = Y_k + 1 = 18 + 1 = 19$

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

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

P_k	P_{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



Problem-02:

Calculate the points between the starting coordinates (20, 10) and ending coordinates (30, 18).

Solution-

Given-

- Starting coordinates = $(X_0, Y_0) = (20, 10)$
- Ending coordinates = $(X_n, Y_n) = (30, 18)$

Step-01:

Calculate ΔX and ΔY from the given input.

- $\Delta X = X_n - X_0 = 30 - 20 = 10$
- $\Delta Y = Y_n - Y_0 = 18 - 10 = 8$

Step-02:

Calculate the decision parameter.

P_k

$= 2\Delta Y - \Delta X$

$= 2 \times 8 - 10$

$= 6$

So, decision parameter $P_k = 6$

Step-03:

As $P_k >= 0$, so case-02 is satisfied.

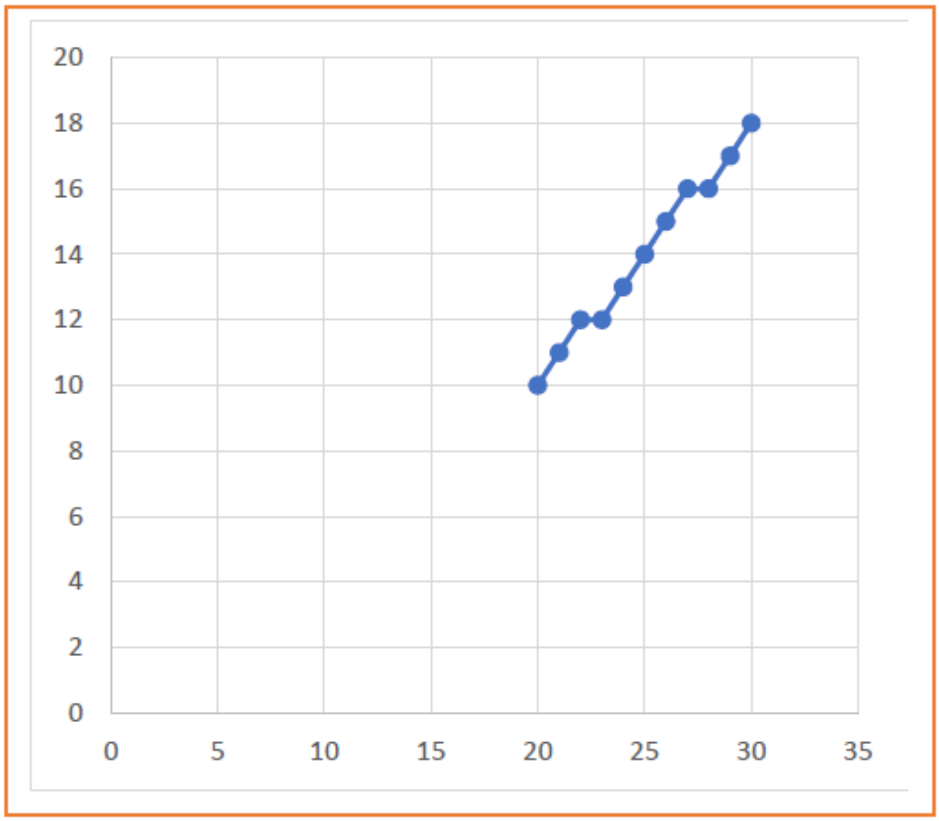
Thus,

- $P_{k+1} = P_k + 2\Delta Y - 2\Delta X = 6 + (2 \times 8) - (2 \times 10) = 2$
- $X_{k+1} = X_k + 1 = 20 + 1 = 21$
- $Y_{k+1} = Y_k + 1 = 10 + 1 = 11$

Similarly, Step-03 is executed until the end point is reached or number of iterations equals to 9 times.

(Number of iterations = $\Delta X - 1 = 10 - 1 = 9$)

P_k	P_{k+1}	X_{k+1}	Y_{k+1}
		20	10
6	2	21	11
2	-2	22	12
-2	14	23	12
14	10	24	13
10	6	25	14
6	2	26	15
2	-2	27	16
-2	14	28	16
14	10	29	17
10	6	30	18



Advantages of Bresenham Line Drawing Algorithm-

The advantages of Bresenham Line Drawing Algorithm are-

- It is easy to implement.
- It is fast and incremental.
- It executes fast but less faster than DDA Algorithm.
- The points generated by this algorithm are more accurate than DDA Algorithm.
- It uses fixed points only.

Disadvantages of Bresenham Line Drawing Algorithm-

The disadvantages of Bresenham Line Drawing Algorithm are-

- Though it improves the accuracy of generated points but still the resulted line is not smooth.
- This algorithm is for the basic line drawing.
- It can not handle diminishing jaggies.

Mid Point Line Drawing Algorithm-

Given the starting and ending coordinates of a line, Mid Point Line Drawing Algorithm attempts to generate the points between the starting and ending coordinates.

Procedure-

Given-

- Starting coordinates = (X_0, Y_0)
- Ending coordinates = (X_n, Y_n)

The points generation using Mid Point Line Drawing Algorithm involves the following steps-

Step-01:

Calculate ΔX and ΔY from the given input.

These parameters are calculated as-

- $\Delta X = X_n - X_0$
- $\Delta Y = Y_n - Y_0$

Step-02:

Calculate the value of initial decision parameter and ΔD.

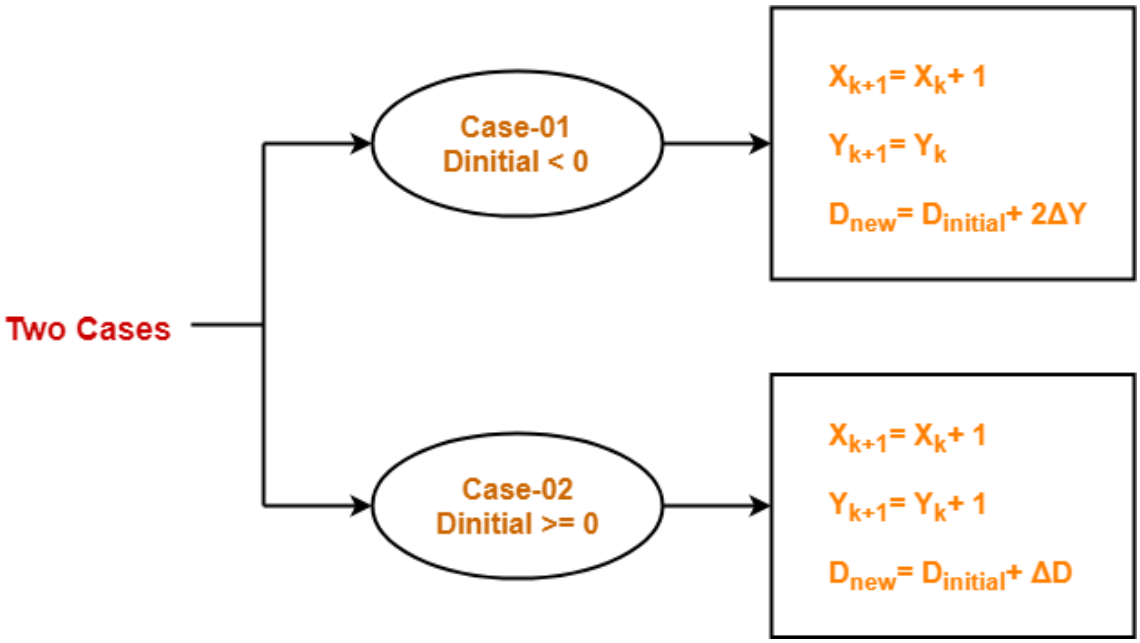
These parameters are calculated as-

- $D_{initial} = 2\Delta Y - \Delta X$
- $\Delta D = 2(\Delta Y - \Delta X)$

Step-03:

The decision whether to increment X or Y coordinate depends upon the flowing values of $D_{initial}$.

Follow the below two cases-



Step-04:

Keep repeating Step-03 until the end point is reached.

For each D_{new} value, follow the above cases to find the next coordinates.

Problem-01:

Calculate the points between the starting coordinates (20, 10) and ending coordinates (30, 18).

Solution-

Given-

- Starting coordinates = $(X_0, Y_0) = (20, 10)$
- Ending coordinates = $(X_n, Y_n) = (30, 18)$

Step-01:

Calculate ΔX and ΔY from the given input.

- $\Delta X = X_n - X_0 = 30 - 20 = 10$
- $\Delta Y = Y_n - Y_0 = 18 - 10 = 8$

Step-02:

Calculate $D_{initial}$ and ΔD as-

- $D_{initial} = 2\Delta Y - \Delta X = 2 \times 8 - 10 = 6$
- $\Delta D = 2(\Delta Y - \Delta X) = 2 \times (8 - 10) = -4$

Step-03:

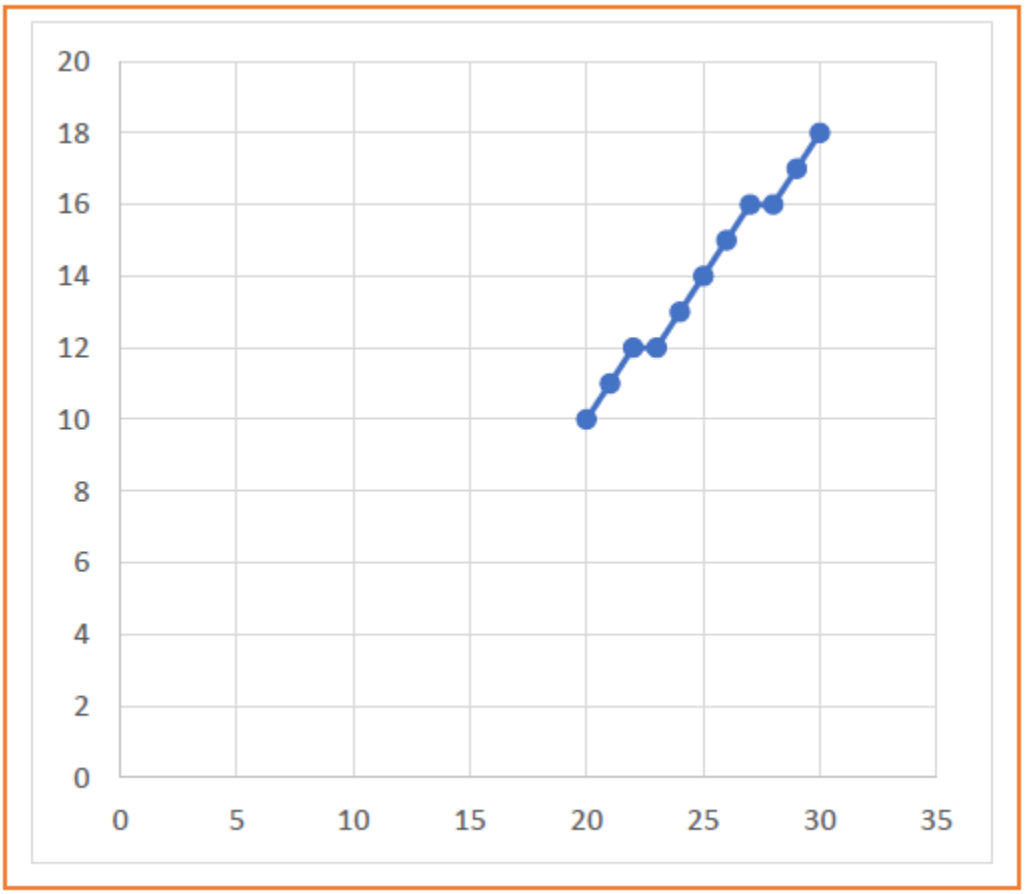
As $D_{initial} >= 0$, so case-02 is satisfied.

Thus,

- $X_{k+1} = X_k + 1 = 20 + 1 = 21$
- $Y_{k+1} = Y_k + 1 = 10 + 1 = 11$
- $D_{new} = D_{initial} + \Delta D = 6 + (-4) = 2$

Similarly, Step-03 is executed until the end point is reached.

$D_{initial}$	D_{new}	X_{k+1}	Y_{k+1}
		20	10
6	2	21	11
2	-2	22	12
-2	14	23	12
14	10	24	13
10	6	25	14
6	2	26	15
2	-2	27	16
-2	14	28	16
14	10	29	17
10		30	18



Problem-02:

Calculate the points between the starting coordinates (5, 9) and ending coordinates (12, 16).

Solution-

Given-

- Starting coordinates = $(X_0, Y_0) = (5, 9)$
- Ending coordinates = $(X_n, Y_n) = (12, 16)$

Step-01:

Calculate ΔX and ΔY from the given input.

- $\Delta X = X_n - X_0 = 12 - 5 = 7$
- $\Delta Y = Y_n - Y_0 = 16 - 9 = 7$

Step-02:

Calculate $D_{initial}$ and ΔD as-

- $D_{initial} = 2\Delta Y - \Delta X = 2 \times 7 - 7 = 7$
- $\Delta D = 2(\Delta Y - \Delta X) = 2 \times (7 - 7) = 0$

Step-03:

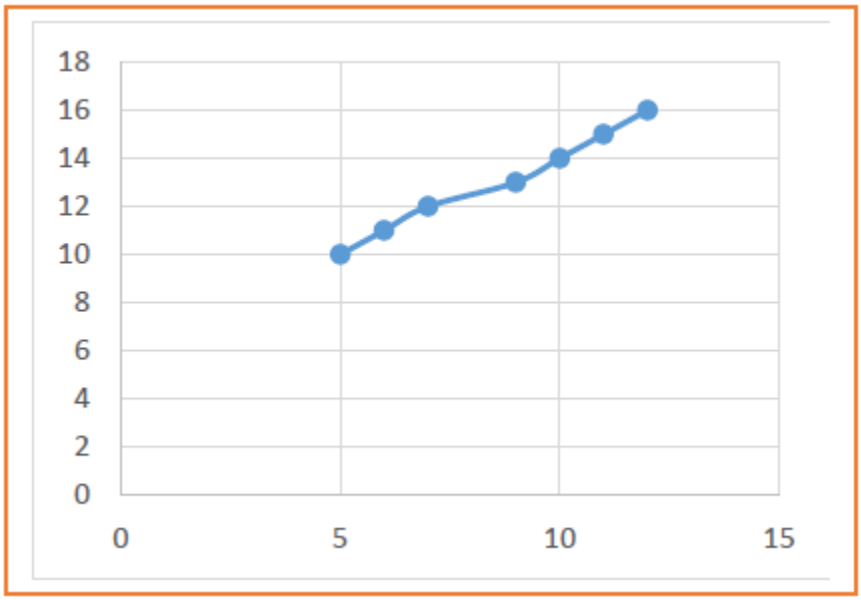
As $D_{initial}$ >= 0, so case-02 is satisfied.

Thus,

- $X_{k+1} = X_k + 1 = 5 + 1 = 6$
- $Y_{k+1} = Y_k + 1 = 9 + 1 = 10$
- $D_{new} = D_{initial} + \Delta D = 7 + 0 = 7$

Similarly, Step-03 is executed until the end point is reached.

$D_{initial}$	D_{new}	X_{k+1}	Y_{k+1}
		5	9
7	7	6	10
7	7	7	11
7	7	8	12
7	7	9	13
7	7	10	14
7	7	11	15
7		12	16



Advantages of Mid Point Line Drawing Algorithm-

The advantages of Mid Point Line Drawing Algorithm are-

- Accuracy of finding points is a key feature of this algorithm.
- It is simple to implement.
- It uses basic arithmetic operations.
- It takes less time for computation.
- The resulted line is smooth as compared to other line drawing algorithms.

Disadvantages of Mid Point Line Drawing Algorithm-

The disadvantages of Mid Point Line Drawing Algorithm are-

- This algorithm may not be an ideal choice for complex graphics and images.
- In terms of accuracy of finding points, improvement is still needed.
- There is no any remarkable improvement made by this algorithm.

The **Mid-Point line plotting algorithm** was introduced by **"Pitway** and **Van Aken."** It is an incremental line drawing algorithm. In this algorithm, we perform incremental calculations. The calculations are based on the previous step to find the value of the next point. We perform the same process for each step. By using the mid-point subdivision algorithm, we can draw a line with close approximation between two points. The mid-point subdivision line drawing algorithm subdivides the line at its midpoints continuously.

The Mid-point Subdivision algorithm helps to compute or calculate the visible areas of lines that appear in the window. This line plotting algorithm follows the bisection method to divide the line into equal partitions.

.

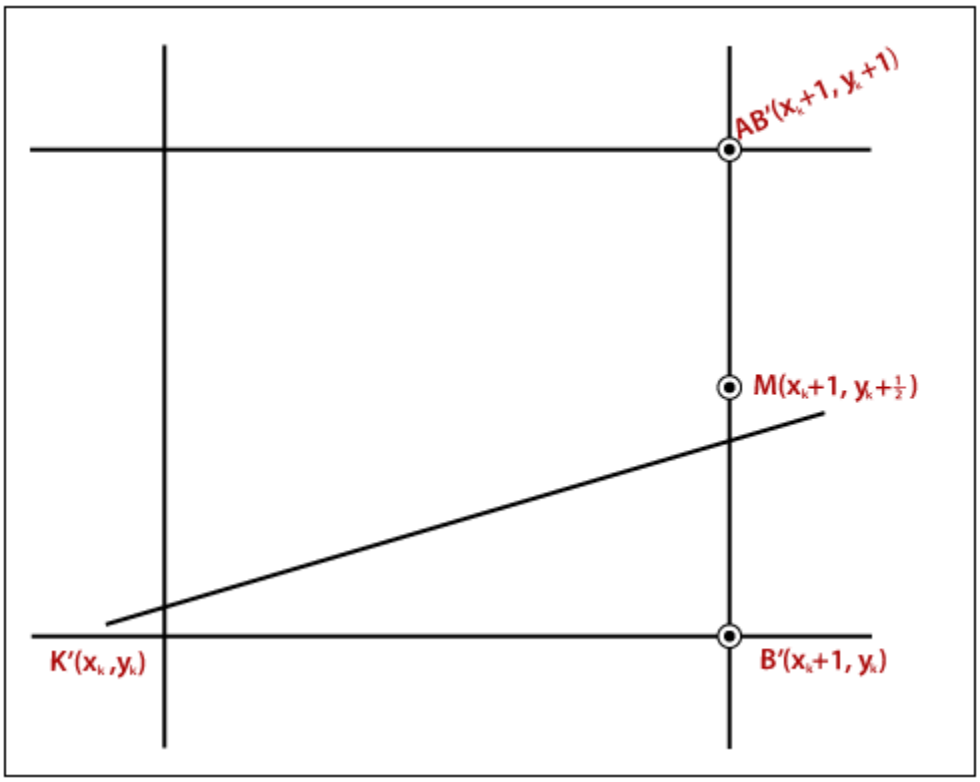


Figure:1

In **figure 1**, we have a previous point **K (x_k,y_k),** and there are two next points; the lower point **B (x_k + 1,y_k)** and the above point **AB** (x_k + 1,y_k + 1). If the midpoint **m** is below the line, then we select point **AB.**

Let us assume, we have two points of the line = **(x_1, x_2)** and **(y_1, y_2).**

Here, it is **(x_1< x_2)**

The Linear equation of a line is:

(x, y) = ax + by +c = 0 …… (1)

$d_x = x_2 - x_1$

$d_y = y_2 - y_1$

As we know, the equation of simple line is:

y = mx +B

Here, **m = dy/dx,** we can write it as **y = (dy/dx) x +B**

Put the value of **y** in above equation, we get

(x, y) = dy (x) – (dx) y +B. dx = 0…… (2)

By comparing equation **(1)** and **(2)**

a = dy

b = -dx

c = B. dx

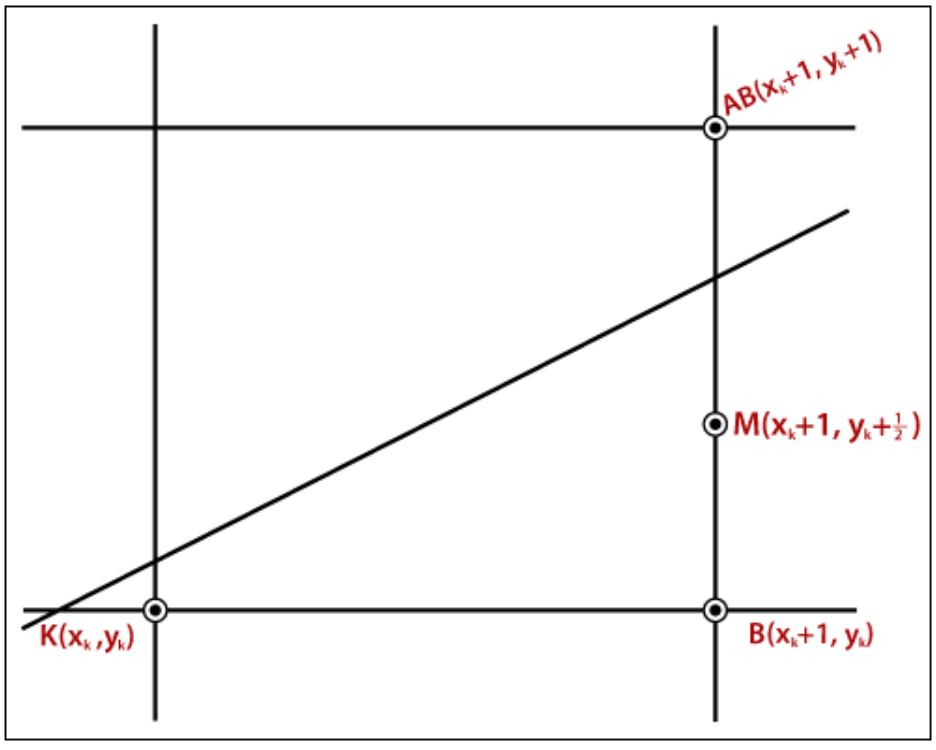


Figure: 2

There are some conditions given-

Condition 1: If all the points are on the line, then **(x, y) = 0**

Condition 2: If all the points are on the line, then **(x, y) =** Negative Number

Condition 3: If all the points are on the line, then **(x, y) =** positive Number

Now, $\mathbf{m = (x_k + 1, y_k + 1/2)}$

$\mathbf{d = m = (x_k + 1, y_k + 1/2)}$

Now, we calculate the distance value **(d).** Here, we discuss two cases:

Case 1: If

We select the lower point **B**

Then

The value of **d** is negative **(d < 0)**

We calculate the new and old value of **d-**

$\mathbf{d_n = (x_k + 2, y_k + 1/2)}$ **{The value of x coordinate is incremented by 1}**

Place the value of **(x, y)** in equation **(1)** and we got the value of $\mathbf{d_o}$

$\mathbf{d_n = a (x_k + 2) + b (y_k + 1/2) + c}$ **{New Value of d}**

$\mathbf{d_o = a (x_k + 1) + b (y_k + 1/2) + c}$ **{Old Value of d}**

Thus,

$\mathbf{\Delta d = d_n - d_o}$

$\mathbf{= a (x_k + 2) + b (y_k + 1/2) + c - a (x_k + 1) - b (y_k + 1/2) - c}$

$\mathbf{= a}$

$\mathbf{\Delta d = d_n - d_o}$

$\mathbf{\Delta d = d_o + a}$

$\mathbf{\Delta d = d_o + dy}$ **{a = dy}**

Case 2: If

We select the upper point **AB**

Then

The value of **d** is positive **(d > 0)**

We calculate the new and old value of **d-**

$d_n = (x_k + 2, y_k + 3/2)$ **{The value of x and y coordinate is incremented by 1}**

We put the value of **(x, y)** in equation **(1)** and we got the value of d_o

$d_n = a(x_k + 2) + b(y_k + 3/2) + c$ **{New Value of d}**

$d_o = a(x_k + 1) + b(y_k + 1/2) + c$ **{Old Value of d}**

Thus,

$\Delta d = d_n - d_o$

$= a(x_k + 2) + b(y_k + 3/2) + c - a(x_k + 1) - b(y_k + 1/2) - c$

$= a + b$

$\Delta d = d_n - d_o$

$\Delta d = d_o + a + b$

$\Delta d = d_o + dy - dx$ **{a + (b) = dy- dx}**

Now, we calculate the initial decision parameter (d_i)

$d_i = (x_1 + 1, y_1 + 1/2)$

Put the value of d_i in equation (1)

$d_i = a(x_1 + 1) + b(y_1 + 1/2) + c$

$= ax_1 + a + by_1 + b/2 + c$

$= (x_1, y_2) + a + b/2$

$= 0 + a + b/2$

$d_i = dy – dx/2$

Algorithm of Mid-Point Subdivision Line Drawing

Step 1: Start.

Step 2: Consider the starting point as (x_1, y_1) and ending point as (x_2, y_2).

Step 3: Now, we will calculate **Δd.**

$$\Delta d = 2 (\Delta y - \Delta x)$$

Step 4: Now, we will calculate the decision parameter d_i with the following formula.

$$d_i = 2\Delta y – \Delta x$$

Step 5: The increment of x or y coordinate depends on the following two cases-

Case 1: If

$$d_i < 0$$

Then

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

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

$$d_n = d_i + 2\Delta y$$

Case 2: If

$$d_i >= 0$$

Then

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

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

$$d_n = d_i + \Delta d$$

Step 6: We will repeat step 5 until we found the ending point of the line.

Step 7: Stop.