Faculty of engineering

Software engineering Dep.

3rd Stage

Computer Graphics

Year program 2022/2023

Prepared by:
Akam H. Ahmed

Lecture content

- Line Generation Algorithm
- DDA Algorithm
- Bresenham’s Line Generation
- Mid-Point Algorithm

Lecture goals

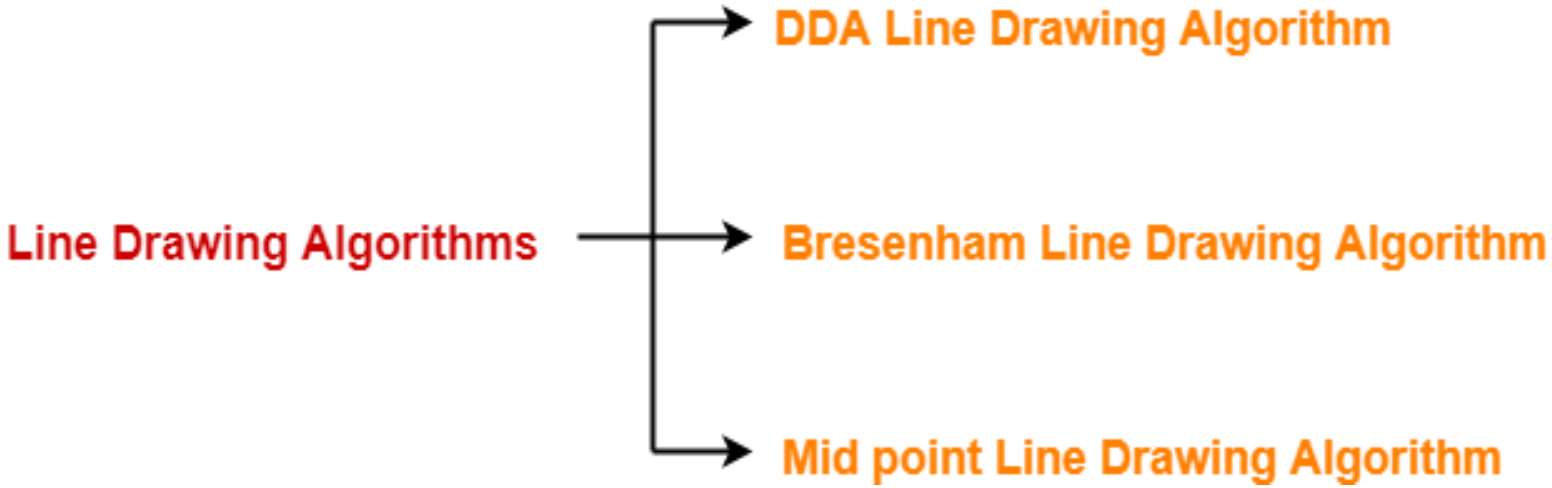
At the end of this lecture you will be able to

- Understand Drawing of line.
- Analyze Drawing Algorithms.
- Apply Drawing line mathematically.

Line Generation Algorithm

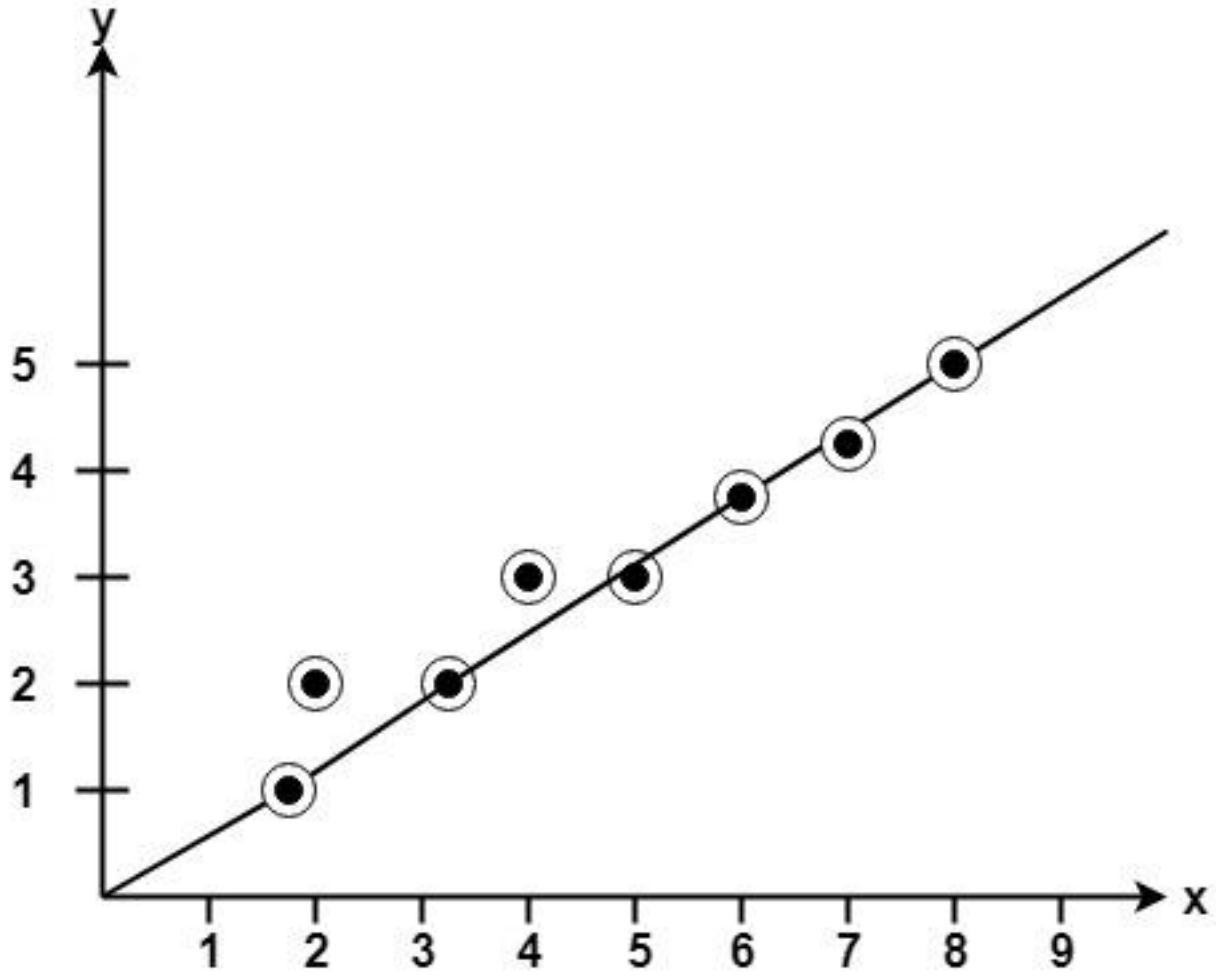
- In computer graphics, popular algorithms used to generate lines are-

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



Line Generation Algorithm

- A line connects two points.
- It is a basic element in graphics.
- To draw a line, you need two points between which you can draw a line.
- In the following three algorithms, we refer the one point of line as X0,Y0 and the second point of line as X1,Y1



DDA 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 = (X0, Y0)

Ending coordinates = (Xn, Yn)

Step 1: Calculate ΔX, ΔY and M from the given input.

These parameters are calculated as-

ΔX = Xn – X0

ΔY =Yn – Y0

M = ΔY / ΔX

Step 2: Find the number of steps or points in between the starting and ending coordinates.

if (absolute (ΔX) > absolute (ΔY))

Steps = absolute (ΔX);

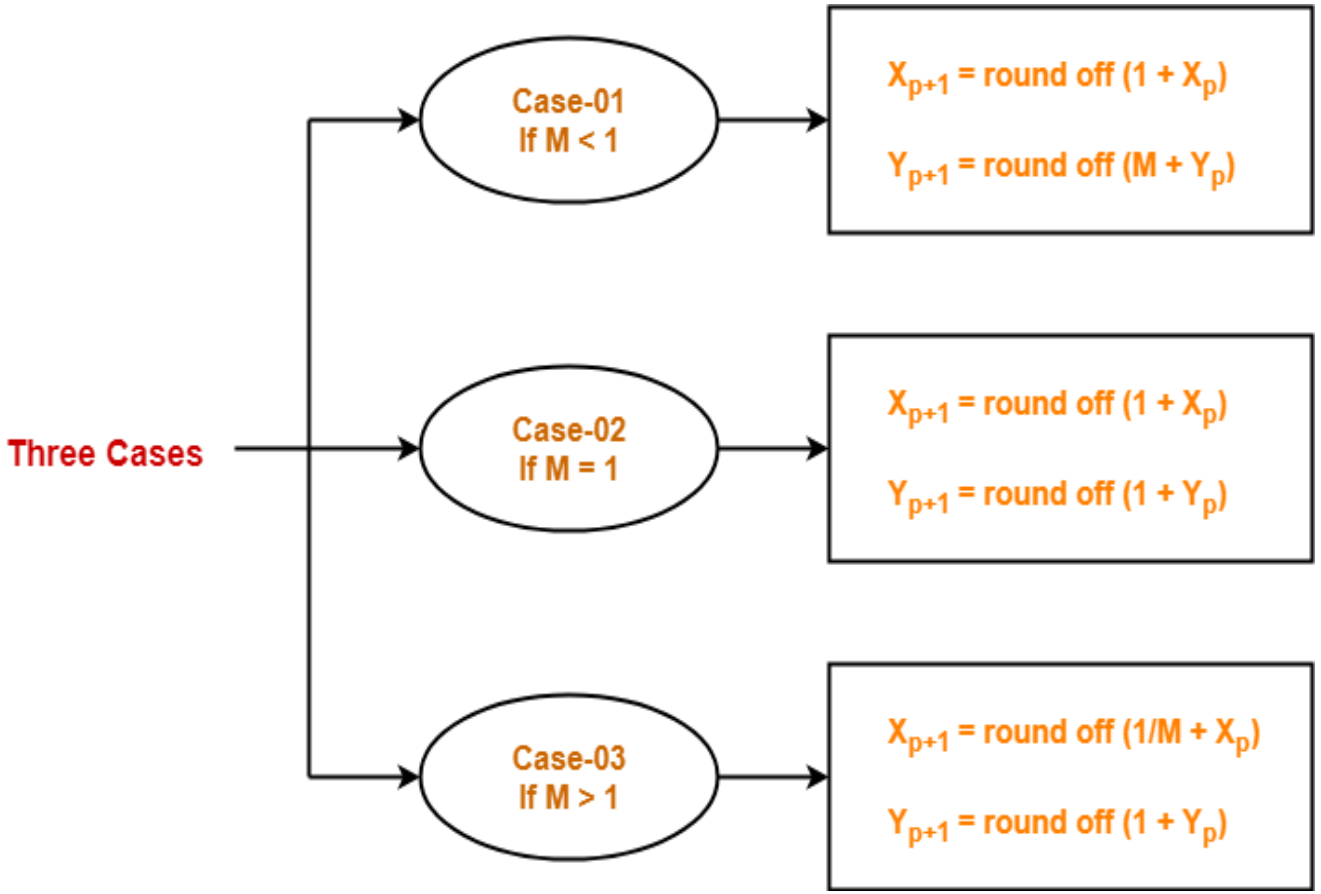
else

Steps = absolute (ΔY);

DDA Algorithm

- Step 3: Suppose the current point is (Xp, Yp) and the next point is (Xp+1, Yp+1).
- Find the next point by following the below three cases

- Step 4: 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.



DDA Algorithm

- **Example** 1:Calculate the points between the starting point (5, 6) and ending point (8, 12).
- **Solution**:

Starting coordinates = (X0, Y0) = (5, 6)

Ending coordinates = (Xn, Yn) = (8, 12)

Step-01: Calculate ΔX, ΔY and M from the given input.

dX = Xn – X0 = 8 – 5 = 3

dY =Yn – Y0 = 12 – 6 = 6

M = dY / dX = 6 / 3 = 2

Step-02:Calculate the number of steps.

abs |dX| < |dY| = 3 < 6, so number of steps = dY = 6

DDA Algorithm

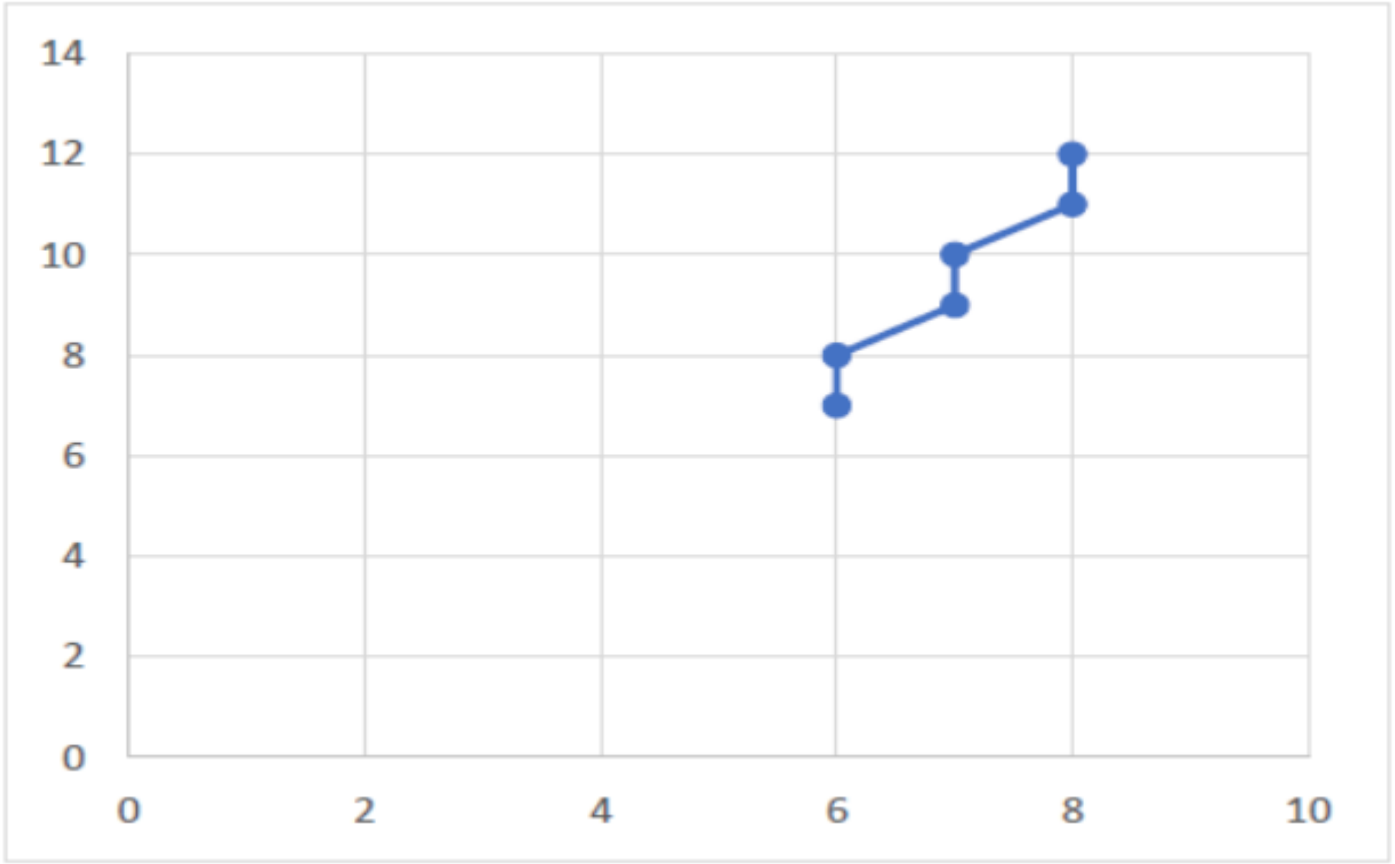
- **Step-03**:As M > 1, so case-03 is satisfied.

  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)

DDA Algorithm

- **Drawing Points:**



DDA Algorithm

- **Example 2:**Calculate the points between the starting point (5, 6) and ending point (13, 10).
- **Solution:** Given-

Starting coordinates = (X0, Y0) = (5, 6)

Ending coordinates = (Xn, Yn) = (13, 10)

Step-01: Calculate ΔX, ΔY and M from the given input.

dX = Xn – X0 = 13 – 5 = 8

dY =Yn – Y0 = 10 – 6 = 4

M = dY / dX = 4 / 8 = 0.50

Step-02: Calculate the number of steps.

As |dX| > |dY| = 8 > 4, so number of steps = dX = 8

DDA Algorithm

- **Step-03:**

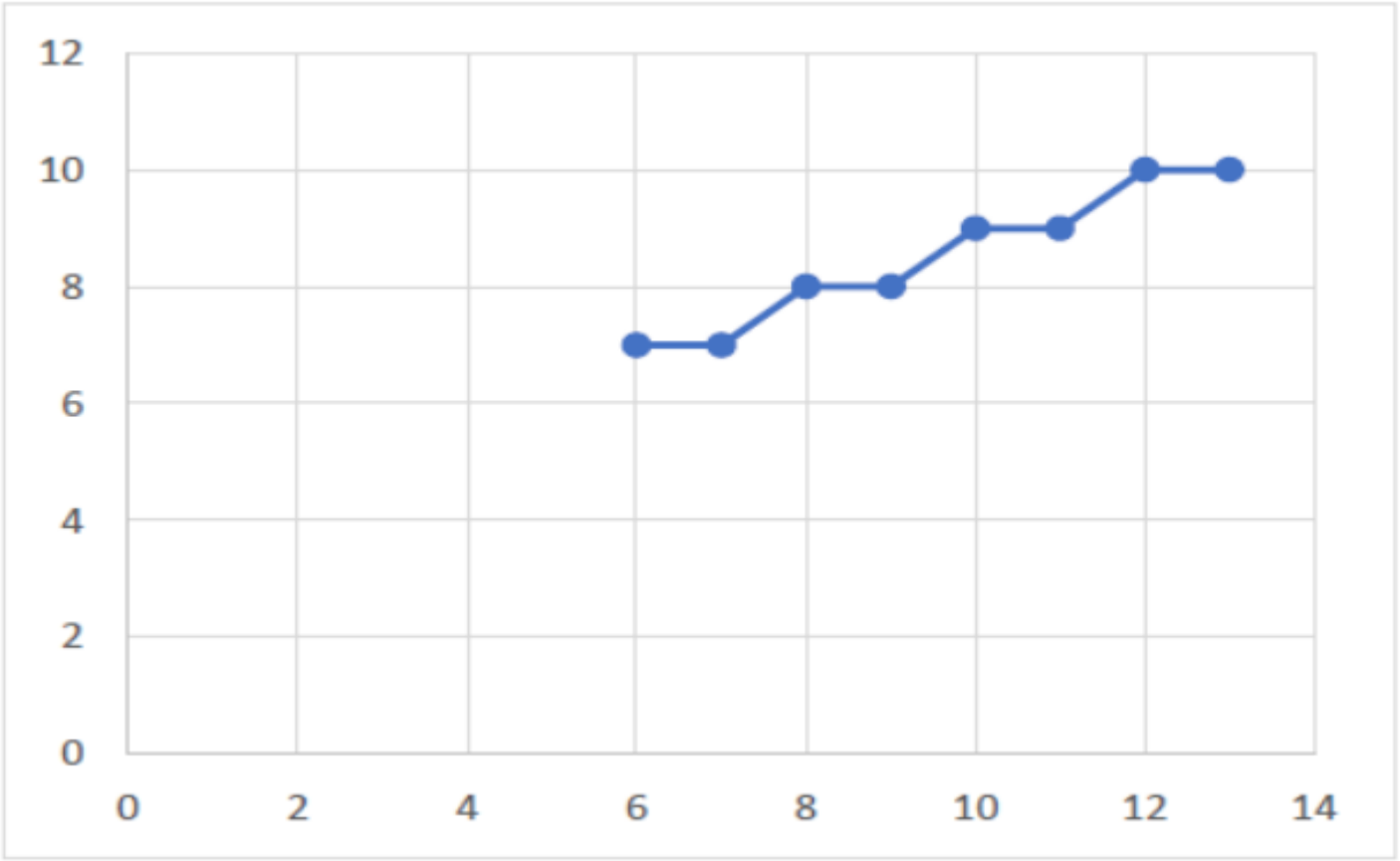
  As M < 1, so case-01 is satisfied.

  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)

DDA Algorithm

- **Plotting points.**



DDA Algorithm

- **Example 3:**Calculate the points between the starting point (1, 7) and ending point (11, 17).
- **Solution:** Given-

Starting coordinates = (X0, Y0) = (1, 7)

Ending coordinates = (Xn, Yn) = (11, 17)

Step-01: Calculate ΔX, ΔY and M from the given input.

dX = Xn – X0 = 11 – 1 = 10

dY =Yn – Y0 = 17 – 7 = 10

M = dY / dX = 10 / 10 = 1

Step-02: Calculate the number of steps.

As |dX| = |dY| = 10 = 10, so number of steps = dX = dY = 10

DDA Algorithm

- **Step-03:**

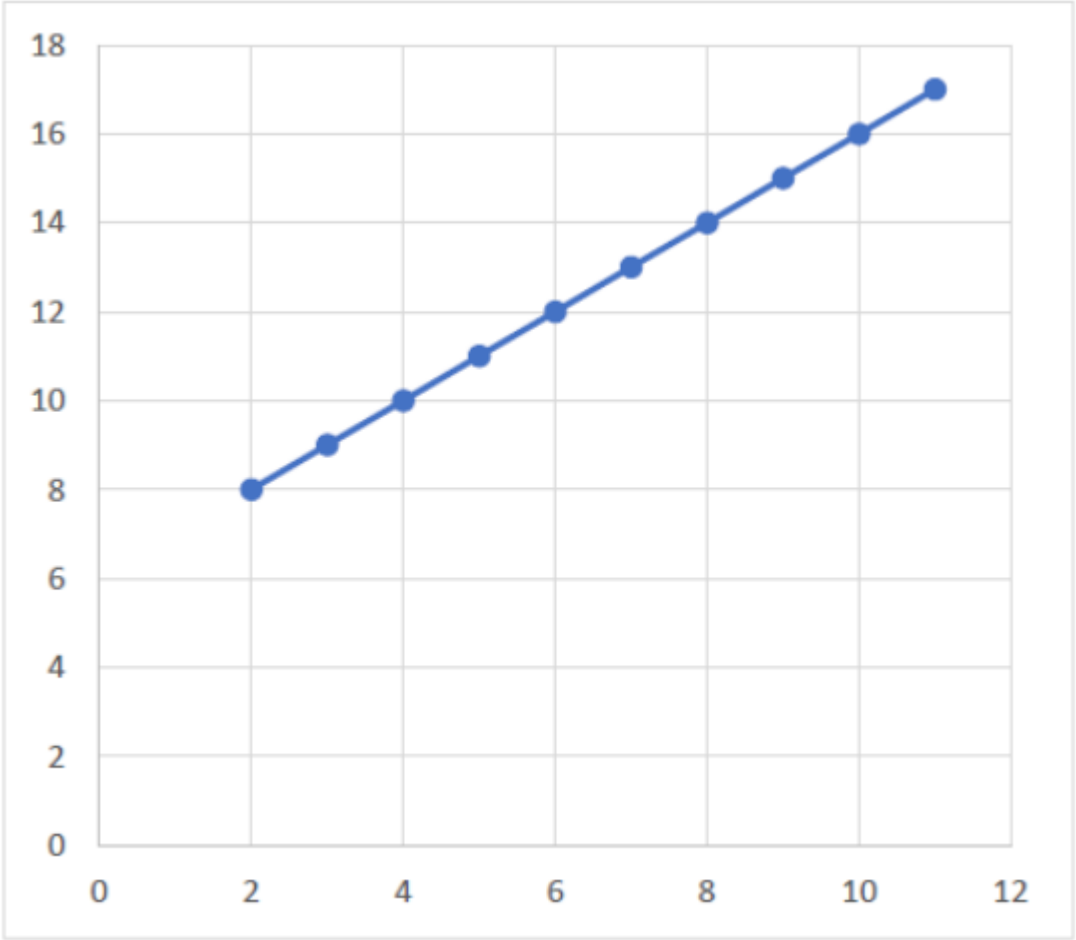
  As M = 1, so case-02 is satisfied.

  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)

DDA Algorithm

- **Plotting points.**



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.

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 Line Drawing Algorithm

Given the starting and ending coordinates of a line.

Bresenham Line Drawing Algorithm attempts to generate the points between the starting and ending coordinates.

Procedure: Given

Starting coordinates = (X0, Y0)

Ending coordinates = (Xn, Yn)

Step-01:

Calculate dX and dY from the given input.

These parameters are calculated as

dX = Xn – X0

dY =Yn – Y0

Bresenham Line Drawing Algorithm

Step-02:Calculate the decision parameter Pk.

It is calculated as-

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

Step-03:

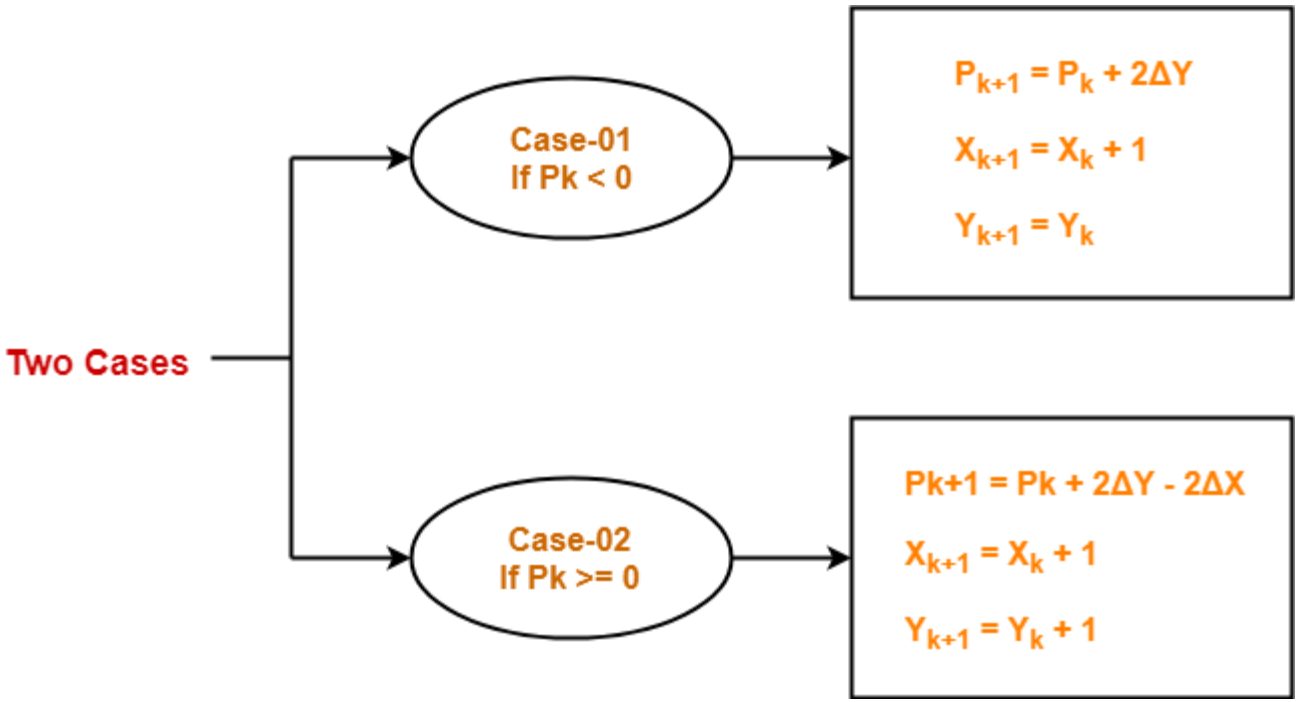
Suppose the current point is (Xk, Yk) and the next point is (Xk+1, Yk+1).

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

Follow the below two cases

Step-04:

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



Bresenham Line Drawing Algorithm

- **Example 1:**Calculate the points between the starting coordinates (9, 18) and ending coordinates (14, 22).
- **Solution:** Given-

  Starting coordinates = (X0, Y0) = (9, 18)

  Ending coordinates = (Xn, Yn) = (14, 22)

  Step-01: Calculate ΔX and ΔY from the given input.

  ΔX = Xn – X0 = 14 – 9 = 5

  ΔY =Yn – Y0 = 22 – 18 = 4

  Step-02: Calculate the decision parameter.

  Pk= 2ΔY – ΔX= 2 x 4 – 5= 3

  So, decision parameter Pk = 3

Bresenham Line Drawing Algorithm

- **Step-03:** As Pk >= 0, so case-02 is satisfied.

  Thus,

  Pk+1 = Pk + 2ΔY – 2ΔX = 3 + (2 x 4) – (2 x 5) = 1

  Xk+1 = Xk + 1 = 9 + 1 = 10

  Yk+1 = Yk + 1 = 18 + 1 = 19.

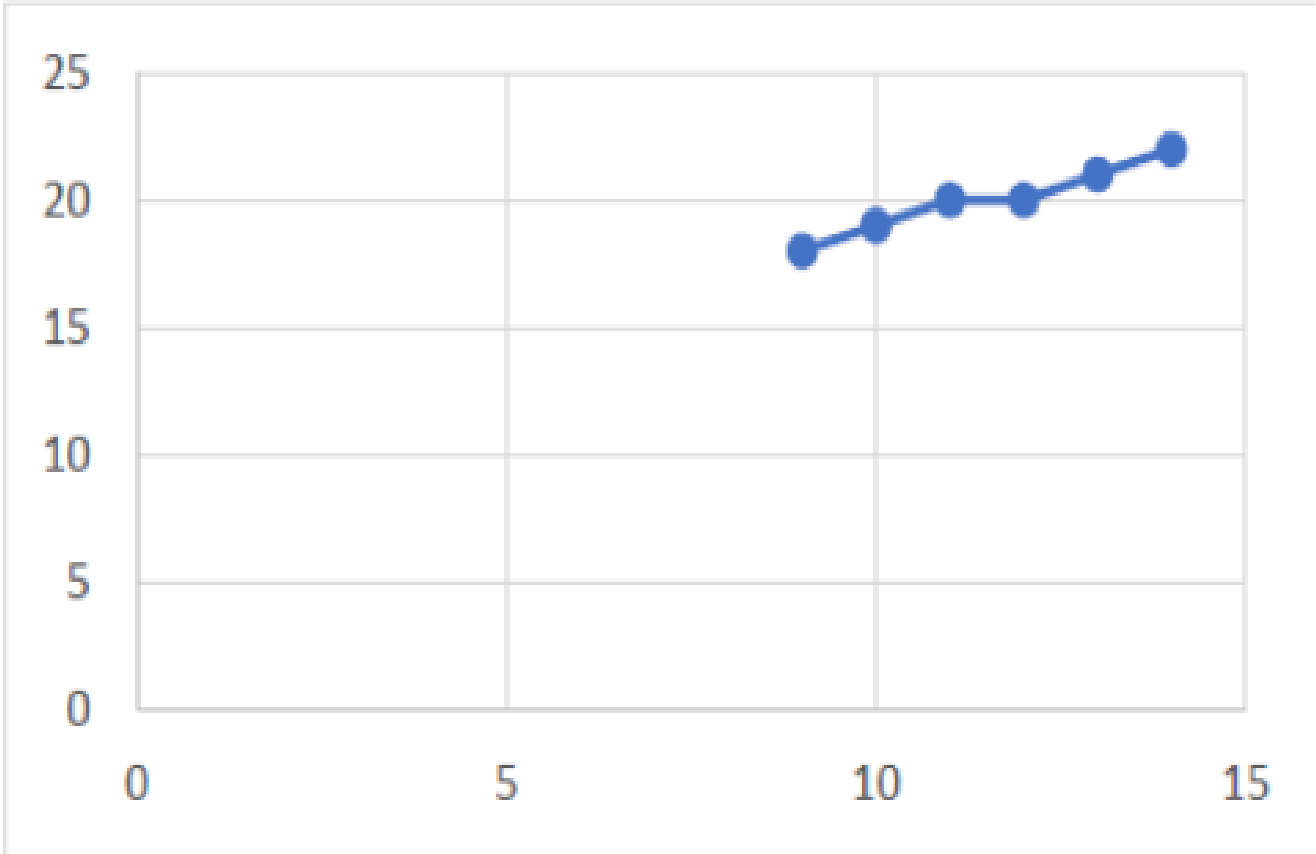
  Similarly, Step-03 is executed until the end point is reached or number of iterations equals to 4 times.

  (Number of iterations = ΔX – 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

Bresenham Line Drawing Algorithm

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



Bresenham Line Drawing Algorithm

- **Example 2:**Calculate the points between the starting coordinates (20, 10) and ending coordinates (30, 18).

  Solution: Given-

  Starting coordinates = (X0, Y0) = (20, 10)

  Ending coordinates = (Xn, Yn) = (30, 18)

- **Step-01**: Calculate ΔX and ΔY from the given input.

  ΔX = Xn – X0 = 30 – 20 = 10

  ΔY =Yn – Y0 = 18 – 10 = 8

  Step-02: Calculate the decision parameter.

  Pk= 2ΔY – ΔX= 2 x 8 – 10= 6

  So, decision parameter Pk = 6

Bresenham Line Drawing Algorithm

- **Step-03:** As Pk >= 0, so case-02 is satisfied.

  Thus,

  Pk+1 = 6 + (2 x 8) – (2 x 10) = 2

  Xk+1 = Xk + 1 20 + 1 = 21

  Yk+1 = Yk + 1 = 10 + 1 = 11

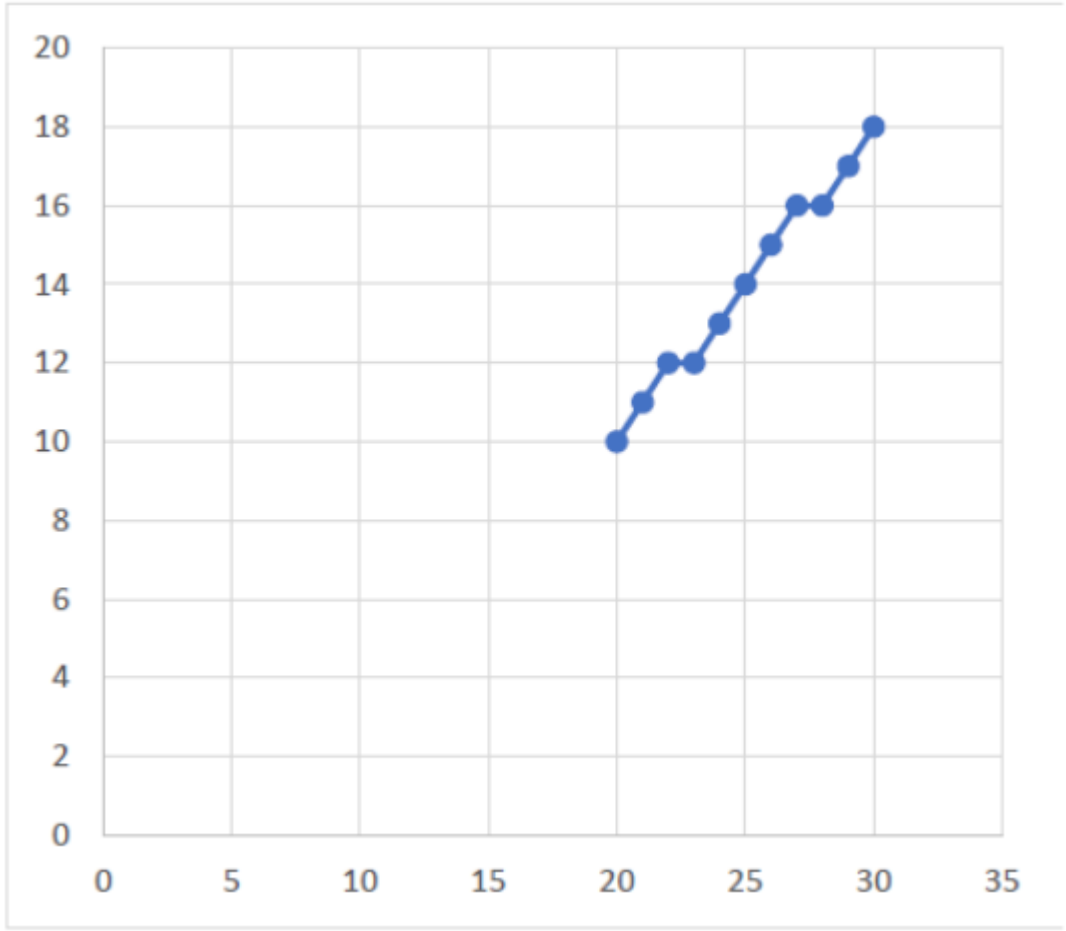
  Similarly, Step-03 is executed until the end point is reached or number of iterations equals to 9 times.

  (Number of iterations = ΔX – 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

Bresenham Line Drawing Algorithm

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



Bresenham Line Drawing Algorithm

Advantages of Bresenham Line Drawing Algorithm:

- 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:

- 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 = (X0, Y0)

Ending coordinates = (Xn, Yn)

Step-01:

Calculate ΔX and ΔY from the given input.

These parameters are calculated as-

$\Delta X = Xn - X0$

$\Delta Y = Yn - Y0$

Mid Point Line Drawing Algorithm

Step-02:Calculate the value of initial decision parameter and ΔD.

These parameters are calculated as-

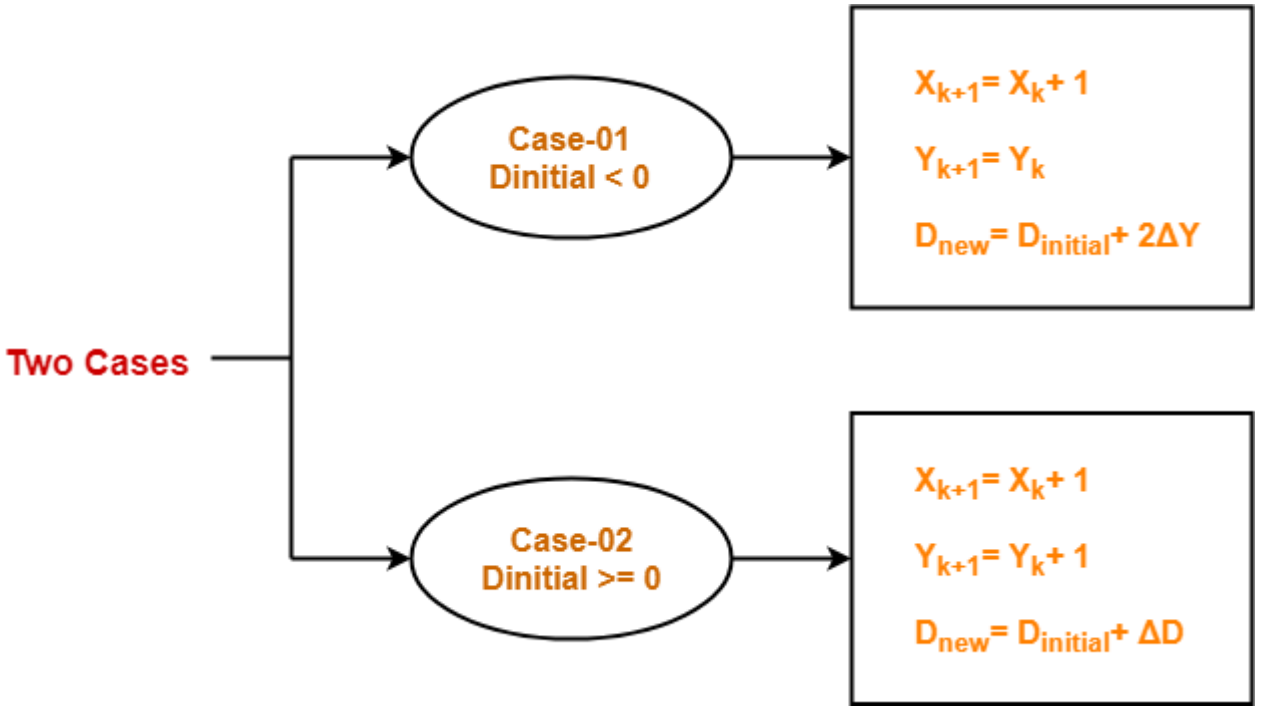
Dinitial = 2ΔY – ΔX

ΔD = 2(ΔY – ΔX)

Step-03:

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

For each Dnew value, follow the above cases to find the next coordinates.



Mid Point Line Drawing Algorithm

- **Example 1:**Calculate the points between the starting coordinates (20, 10) and ending coordinates (30, 18).

  Solution: Given-

  Starting coordinates = (X0, Y0) = (20, 10)

  Ending coordinates = (Xn, Yn) = (30, 18)

  Step-01: Calculate ΔX and ΔY from the given input.

  ΔX = Xn – X0 = 30 – 20 = 10

  ΔY =Yn – Y0 = 18 – 10 = 8

  Step-02: Calculate Dinitial and ΔD as-

  Dinitial = 2ΔY – ΔX = 2 x 8 – 10 = 6

  ΔD = 2(ΔY – ΔX) = 2 x (8 – 10) = -4

Mid Point Line Drawing Algorithm

- **Step-03:** As Dinitial >= 0, so case-02 is satisfied.

  Thus,

  Xk+1 = Xk + 1 = 20 + 1 = 21

  Yk+1 = Yk + 1 = 10 + 1 = 11

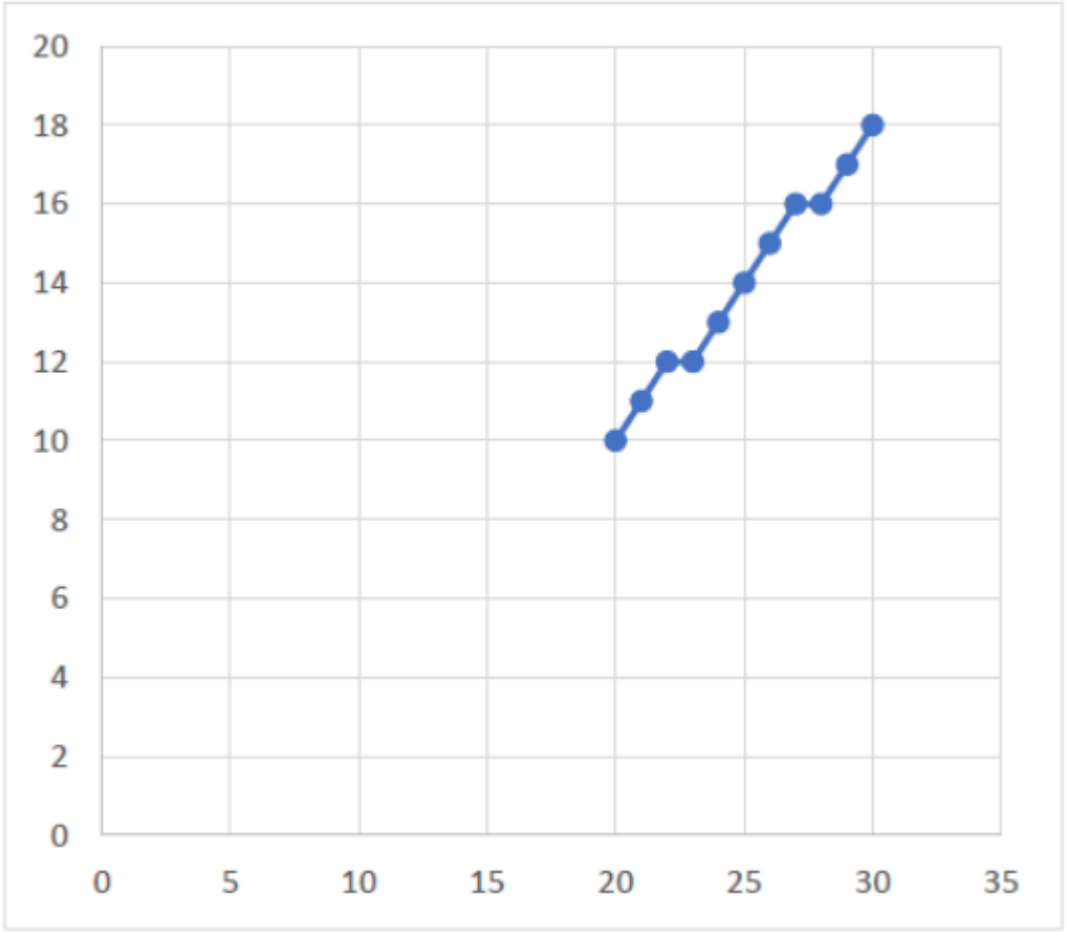
  Dnew = Dinitial + ΔD = 6 + (-4) = 2

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

Mid Point Line Drawing Algorithm

$D_{initial}$	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



Mid Point Line Drawing Algorithm

- **Example 2:**Calculate the points between the starting coordinates (5, 9) and ending coordinates (12, 16).

  Solution: Given-

  Starting coordinates = (X0, Y0) = (5, 9)

  Ending coordinates = (Xn, Yn) = (12, 16)

Step-01: Calculate ΔX and ΔY from the given input.

ΔX = Xn – X0 = 12 – 5 = 7

ΔY =Yn – Y0 = 16 – 9 = 7

Step-02: Calculate Dinitial and ΔD as-

Dinitial = 2ΔY – ΔX = 2 x 7 – 7 = 7

ΔD = 2(ΔY – ΔX) = 2 x (7 – 7) = 0

Mid Point Line Drawing Algorithm

- **Step-03:**As Dinitial >= 0, so case-02 is satisfied. Thus,

  Xk+1 = Xk + 1 = 5 + 1 = 6

  Yk+1 = Yk + 1 = 9 + 1 = 10

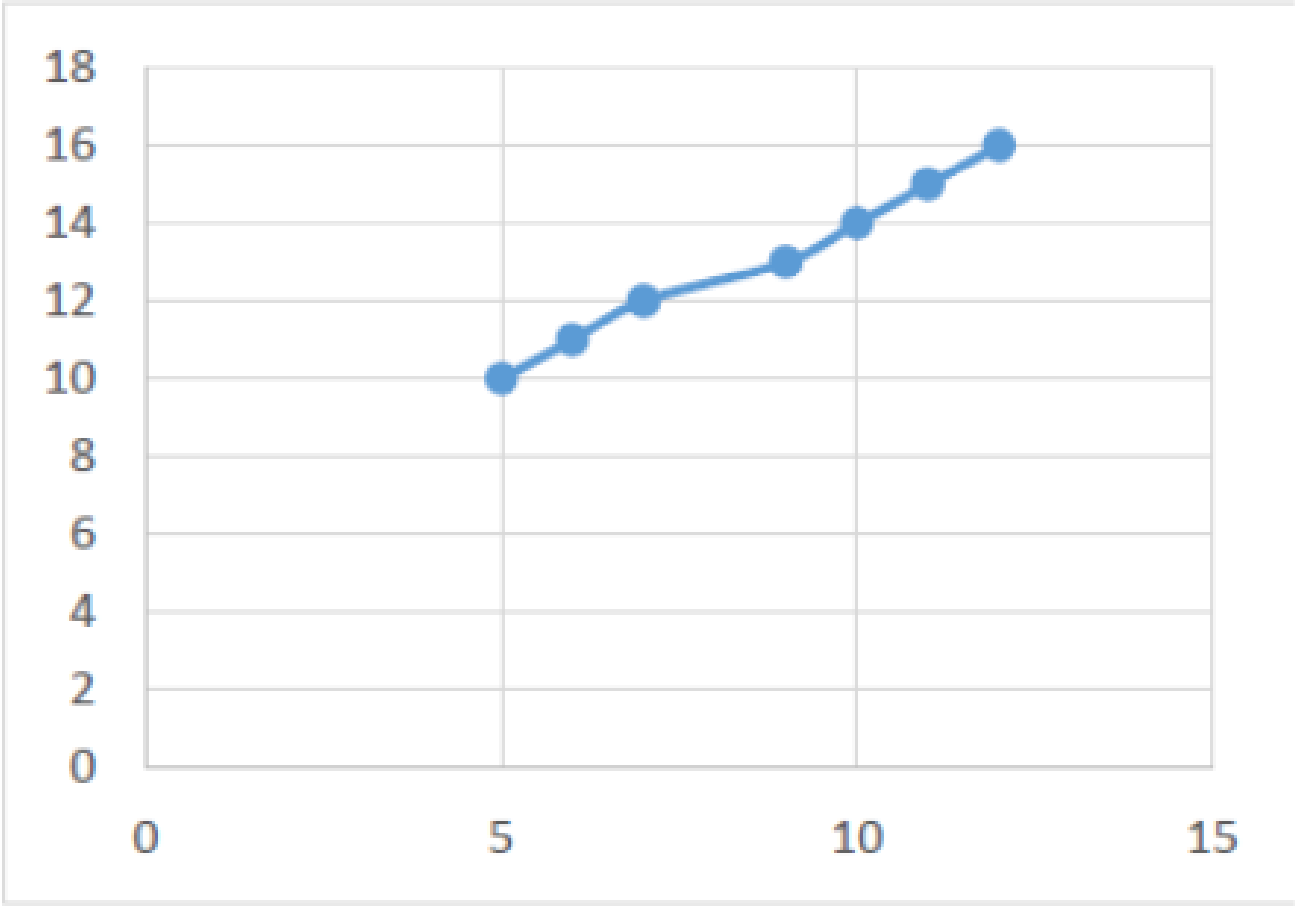
  Dnew = Dinitial + ΔD = 7 + 0 = 7

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

Mid Point Line Drawing Algorithm

$D_{initial}$	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



Mid Point Line Drawing Algorithm

Advantages of Mid Point Line Drawing Algorithm:

- 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:

- 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.

References.

- Hughes, John F., van Dam, Andries, McGuire, Morgan, Sklar, David F., Foley, James D., Feiner, Steven and Akeley, Kurt. Computer Graphics: Principles and Practice. 3 Upper Saddle River, NJ: Addison-Wesley, 2013.
- Gambetta, G. (2021). Computer Graphics From Scratch: A programmer's introduction to 3D rendering.
- “Computer Graphics.” Tutorialspoint. Accessed October 20, 2022. https://www.tutorialspoint.com/computer_graphics/index.asp.

End

Thank you