# UNIT 3: LINE DRAWING ALGORIHMS

## UNIT STRUCTURE


3.1 Learning objectives

3.2 Introduction

3.3 Points and lines

3.4 Line drawing algorithms

3.5 Scan conversion of lines

3.6 DDA Line drawing algorithms

3.7 Bresenham's Line drawing algorithms

3.7.1 Steps involved in Bresenham's Line drawing algorithm

3.7.2 Advantages and disadvantages of Bresenham's Line drawing algorithm

3.8 Let Us Sum Up

3.9 Further Reading

3.10 Answer to Check Your Progress

3.11 Model Questions


## 3.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- List different parameters related to graphics in a computer system.
- Explain about different issues in line drawing algorithms.
- Cite how to draw a line using DDA line drawing algorithms.
- Describe how to draw a line using Bresenham's line drawing algorithms.
- Calculate a straight line intermediate coordinates to plot pixels with examples.

## 3.2 INTRODUCTION

In the previous unit, we have discussed about graphic system. In this unit, you will be introduced to the concept of line drawing algorithm. The unit will first discuss about point and line and then we will discuss

about line drawing algorithm. Scan conversion of lines, DDA Line drawing algorithms and Bresenham's Line drawing algorithms will be discussed in the later part of the unit. In the next unit, we will discuss circle and ellipse drawing algorithm.

## 3.3 POINTS AND LINES

A straight line can be defined as the collection of dots. In coordinate geometry a straight line can be defined using the straight-line equation:

$Y = mx + c$ ---------- (1),

Where m represents the slope of the line, and c is the y-intercept. In computer graphics, line drawing can be defined as a straight line segment by plotting some discrete points between two endpoints. The line equation is used to determine the coordinates of discrete points. In computer graphics, the discrete points can be termed as PIXEL, the smallest element used to represent an object. PIXEL is the abbreviation of picture element. The computer system used binary numbers to control the display of PIXELS. For the bit value 1, the pixel becomes white, and for the value 0, the pixel turns black. By increasing the number of bits, the pixels can feel different colours. Pixel colours are determined by the number of bits used to represent it. Using two-bit binary numbers, a pixel can have three different colours: 0 0 - black, 0 1 - grey, 1 0 - grey and 1 1 - white. Number of colours can express as the power of two of the no. of bits. A system with 16 bit processor can have 216 numbers of colour combinations.

The pixel position can be represented by the two-dimensional coordinate geometry. If the coordinate of a pixel is (X, Y), then the coordinate of the next pixel in the X direction will be (X + 1, Y), and in the Y coordinate, the next position will be (X, Y + 1).

A point can be defined as a pixel in the computer graphics system. In a frame buffer system, the intensity of each pixel is calculated or measured separately. These are based on the Cartesian coordinate system. A point or a pixel is represented in terms of x and y coordinates. The total number of pixels n a display system depends on the coordinate precision and

resolution of the system. The resolution of a system or a display device can be defined as the number of pixels or dots that can be displayed in a given area of the screen. It is usually measured in dots per inch or .dpi. Greater the number of pixels infers higher the resolution. Pixels are the smallest addressable screen element. Computer graphics system images are made by controlling the intensity and colour of the pixels composing the screen. The following figure 3.1 has shown a straight line after plotting in coordinate positions. The appearance of the line is a jagged or stair-step pattern. For a system with low resolution, the appearance of straight lines or circles or any picture element will be jagged. But b increasing the number of points or resolution, the effect of stair-step can be avoided or eliminated. This is the main drawback of plotting a point in a computer graphics system. Using line drawing and circle drawing algorithms, it can be reduced.

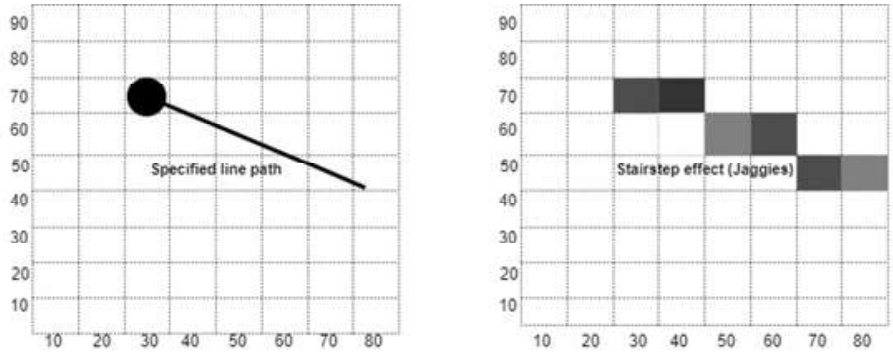
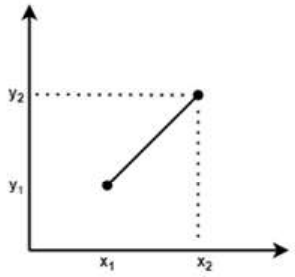



Figure 3.1 A line segment and jagged appearance of the line in graphics system

## 3.4 LINE DRAWING ALGORITHM

The equation of a line is shown in the expression (1). Where the slope (m) can be calculated as:




$$m = \frac{y_2 - y_1}{x_2 - x_1} \text{.....(2)}$$

$$c = y_1 - m.x_1 \text{.....(3)}$$

For any given x interval $x\Delta$ along a line, the corresponding y interval can be calculated as $\Delta y = m.\Delta x$ .....(4)

Similarly, the x interval $x\Delta$ along a line, corresponding to a specified $y\Delta$ as $\Delta x = \frac{\Delta y}{m}$.....(5)

These equations are the basis for determining the deflection voltages in analog devices.

## 3.5 SCAN CONVERSION OF LINES

Representation of a regular object in terms of discrete pixels is known as scan conversion. In a computer system, each graphics handler has to convert or transform the primitives into a collection of pixels. The method of scan conversion is also known as Rasterization. Each pixel in a graphics system has two states - on or off. The horizontal and vertical frequency can change for different purposes, especially in video processing, which is known as the scan converting rate.

For the scan conversion of a straight line, two endpoint coordinates are needed. In our normal life, we use scale and pencil to draw a line between two endpoints on a copy. But in a computer system, it is not possible to draw the line directly with the help of a roller. A computer can draw a line by calculating all the intermediate points in between the two endpoints. During calculation, it is possible to get fractional numbers for intermediate positions. But a computer system cannot accept a part of any pixel or a fractional number. So the fractions are rounded off to the nearest integers. Then the system fills the computed intermediate coordinate positions. A pixel never fills partially. So a straight line appearance always looks like stair-step format.

There are several algorithms used for this scan conversion. To draw a straight line - algorithms are the direct method, DDA algorithm, and Bresenham's algorithm. Similarly for drawing a circle Midpoint circle algorithm, Bresenham's Circle algorithm. To draw ellipse, midpoint ellipse drawing algorithm, etc.

**Characteristics of straight lines for different slope (m):**

- Slope direction can be positive, negative, zero or undefined.
- Positive slope - y values increases as the x values increases. The line slope upwards to the right.
- Negative slope - y values decreases as the x values increases. The line slope downwards to the right.
- Zero slope – y values does not change with the changing of x value. The line in exactly horizontal.
- Undefined slope – When the line exactly vertical, it does not have a definition.



**CHECK YOUR PROGRESS**

**Q 1:** Define slope of a straight line. Also explain the characteristics of straight lines for different slope.

**Q 2:** Explain the term scan conversion.

**Q 3:** Define the term Scan Conversion rate.

**Q 4:** Write the equation of a straight line.

## 3.6 DDA LINE DRAWING ALGORITHM

A digital differential analyser (DDA) is a scan conversion line algorithm based on calculating $\Delta y$ or $\Delta x$. A line point is sampled in unit interval along a coordinate and determines the corresponding integer value of the line path for the other coordinate.

Consider a line with a positive slope, $(m \le 1)$. For a line with a positive slope, the gradient m is always less than or equal to one. If we sample the line path on the x-axis with the unit interval $(\Delta x = 1)$, then the successive y value can be calculated using the following equation:

$$y_{k+1} = y_k + m \quad \text{.....(6)}$$

Where the *k* values will start from 1 for the first point, and increase by 1 unit until the sample points reached the endpoint of the line. The value of *m* can range In between 0 and 1. The calculated *y* value must be rounded to the nearest integer.

For a line with a positive slope greater than 1, the role x and y will be reversed. That is the line path will be a sample in unit y intervals $(\Delta y = 1)$, and calculate each successive x value as

$$x_{k+1} = x_k + \frac{1}{m} \text{.....(7)}$$

For the equation numbers (6) and (7), it is assumed that the line path starts from left endpoints to right endpoints. If the sampling is reversed, i.e. the starting endpoint starts from the right end and ended at the left, then the value of $\Delta x = -1 \text{ and } y_{k+1} - m$.....(8)

And when the slope is greater than 1, and

$$\Delta y = -1, \text{ than } x_{k+1} - \frac{1}{m} \text{.....(9)}$$

Equations (6) and (9) can be used to calculate the line path with a negative slope also. If the slope is less than 1, and the start endpoint is at left, set $\Delta x = 1$ and calculate the successive y value with equation (6). If the start endpoint is at right and the slope is positive and the slope value is less than one, set $\Delta x = -1$ and calculate the y value using the equation (8). Similarly if the absolute value of a negative slope is greater than 1, set $\Delta y = -1$ then we can use $\Delta y = -1$ in equation (9) and $\Delta y = 1$ in equation (7). The DDA algorithm is faster than equation (1) in the calculation of pixel position. The accumulation of round-off error in the calculation of successive point coordinates can drift away from the true line path from the original one. The performance of the DDA line drawing algorithm can be improved by separating the increments m and 1/m into integer and fractional parts so that all operations are restricted within the integer operation.

**DDA Algorithm:**

**Example 1:** Draw a line segment with the endpoint coordinates (0, 0) and (6, 6) using the DDA line drawing algorithm.

**Solution:** The starting point coordinate of the line is (0, 0) and The right end point coordinate of the line is (6, 6)

| Sl. No. | X | Y |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 1 | 1 |
| 3 | 2 | 2 |
| 4 | 3 | 3 |
| 5 | 4 | 4 |
| 6 | 5 | 5 |
| 7 | 6 | 6 |

So, $\Delta x = 6 - 0 = 6$ and $\Delta y = 6 - 0 = 6$, where x1=0, x2=6, y1=0 and y2=6

Thus the slope of the line m = 6/6 = 1

Since the slope of the line m = 1, the line starts at the left endpoint and is processed to the right endpoint. So the next pixel coordinate along x-axes will be: X i = X i + 1

And the corresponding y coordinate of the pixel will be calculated as yi = yi + m

The computed values for each pixel point on the line segment are shown in the table.

Example 2:Draw a line between the points (1, 1) and (8, 7) using the DDA line drawing algorithm.

Solution: The starting point is (1, 1), and the right endpoint is (8, 7)

| Sl. No. | X | Y | Pixel plotted |
|---|---|---|---|
| 1 | 2 | 2 | 2, 2 |
| 2 | 3 | 2+6/7=2.9 | 3, 3 |
| 3 | 4 | 2.9+6/7=3.8 | 4, 4 |
| 4 | 5 | 3.8+6/7=4.7 | 5, 5 |
| 5 | 6 | 4.7+6/7=5.6 | 6, 6 |
| 6 | 7 | 5.6+6/7=7.0 | 7, 7 |

The slope of the line m = (7 - 1)/ (8 - 1) = 6/7 (slope m < 1)

Here it is given that x1=1, x2=8, y1=1 and y2=7

Since the slope m < 1, the x will increment in unit intervals and y has to be determined.

So the next x coordinate position along the line path will be x1 + 1 and

And the next y coordinate of the line path will be y1 + m =y1 + 6/7 and so on. All the pixels on the line path are shown in the table:



**EXERCISE**

1. Draw a line between the points (12, 15) and (20, 23) using the DDA line drawing algorithm.
2. Draw a line between the points (32, 28) and (20, 18) using the DDA line drawing algorithm.

## 3.7 BRESENHAM'S LINE DRAWING ALGORITHM

It is the most efficient raster line-generating algorithm developed by Bresenham. In this algorithm, scan conversion is done by calculating the incremental integers which are used to display circles or any other curves. In the following Figures 4.3 and 4.4, the horizontal axes identify the pixel column and the vertical axis shows the scan line positions. While the value of x is increased by one unit, then it has to determine which of two possible pixel positions is closer to the line path at each sample step. In figure 4.3, it has to determine that, in the next pixel position whether to plot at the position (30, 30) or (30, 40). Similarly, figure3.4 shows a line with a negative slope from the left endpoint at the coordinate position (30, 70). What will be the next pixel position, whether (40, 70) or (40, 60)? In Bresenham's algorithm, it can be determined with the help of a sign of an integer parameter, which value is proportional to the difference between the separations of the two-pixel positions from the actual line path.

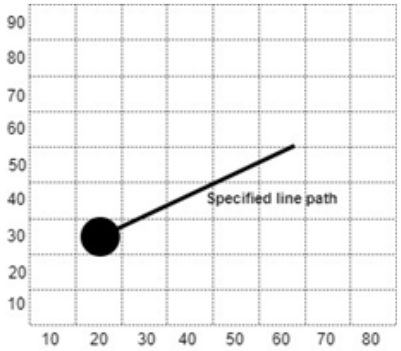



*Figure 3.3 Section of a display screen Where a positive slope line is plotted*

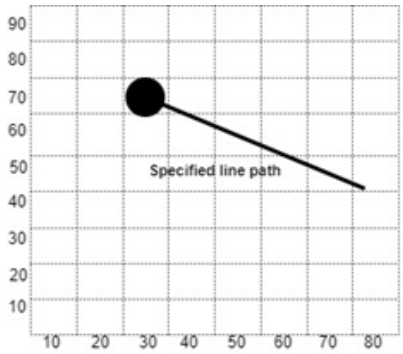



*Figure 3.4 Section of a display screen Where a negative slope line is plotted*

Consider a line with a positive slope of less than 1. Now, applying Bresenham's approach to determine the pixel positions along a line path by sampling at unit x intervals. Starting from the left endpoint ($x_0$, $y_0$) of a

given line, step to each Successive pixel towards x-axis, and calculates the corresponding y position which is closer to the line path. Figure 3.5 and 3.6 depicts the $k^{th}$ step in the algorithm. Now assume that the current pixel position is determined as $(x_k, y_k)$. Next, need to determine the y coordinate for the pixel position at $x_k + 1$. The choices for the next pixel positions are $(x_k + 1, y_k)$ and $(x_k + 1, y_k + 1)$.

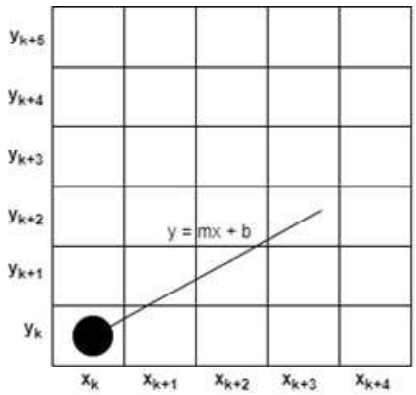



*Figure 3.5 Section of a screen grid showing a pixel at (xk, yk) Position*

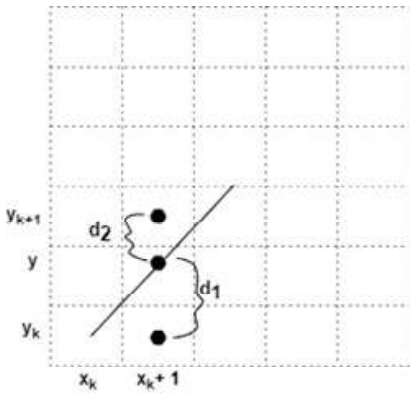



*Figure3.6Distance between pixel position and theline at xk+1 position*

In figure: 3.6, at the sampling position $x_k + 1$, the vertical separation from the mathematical or actual line path is considered as d1 and d2. Now the y coordinate on the actual line at sampling position $x_k + 1$ can be calculated as

$$y = m(x_k + 1) + b \text{.....(10)}$$

$$than \quad \begin{matrix} d1 = y - y_k \\ = m(x_k + 1) + b - y_k \end{matrix} \qquad and \qquad \begin{matrix} d2 = (y_k + 1) - y \\ = y_k + 1 - m(x_k + 1) - b \end{matrix}$$

The difference between these two separations is

$$\mathbf{d1 - d2 = 2\,m(x_k + 1) - 2\,y_k + 2\,b - 1 \text{.....(11)}}$$

A decision parameter pkfor the kth step in the line algorithm can be obtained from equation (11) by rearranging it. To remove the fractional part or to get an integer value, the m can be written as $\Delta y / \Delta x$, where the $\Delta x$ and $\Delta y$ are the vertical and horizontal separations of the endpoint positions. The pkcan be defined as:

$$d1 - d2 = 2\frac{\Delta y}{\Delta x}(x_k + 1) - 2y_k + 2b - 1$$

$$\Delta x(d1 - d2) = 2\Delta y(x_k + 1) - 2\Delta x.y_k + 2b.\Delta x - \Delta x$$

$$\therefore p_k = \Delta x(d1 - d2)$$

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

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

$$= 2\Delta y.x_k - 2\Delta x.y_k + c \quad \text{.....(12)}$$

(where c is a constant, and c=$2\Delta y + \Delta x(2b - 1)$)

The sign of the $d_1$- $d_2$ will be the same as the sign of decision parameter pk. In our example, ?x > 0, and c is constant and independent of pixel position.

If the position $y_k$ is closer to the actual line path, then the pixel at $y_k$ + 1, i.e. $d_1 < d_2$, then the sign of the decision parameter pk will be -ve, and the lower pixel have to be chosen. The coordinate changes occur in the unit step in either x or y directions. So the successive decision parameter $p_{k+1}$ can have through incremental integer calculations. At step k+1, from the equation number (12), the decision parameter can be computed as:

$$p_{k+1} = 2\Delta y.x_{k+1} - 2\Delta x.y_{k+1} + c \text{.....(13)}$$

Now subtracting the equation (12) from (13) we have

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

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

Since $x_{k+1} = x_{k+1}$, the successive decision parameter for the next pixel position will be:

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

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

$$= p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k) \text{.....(14)}$$

The value of the term $y_{k+1} - y_k$ is either 0 or 1, depending on the sign of the decision parameter $p_k$.If $y_k$ is closer than $y_{k+1}$, the sign of the decision parameter $p_k$ is negative. At this moment we have to choose the $y_k$

value for the y coordinate $y_{k+1}$, thus the value for the expression $y_{k+1} - y_k$ = 0. If $y_{k+}$ 1 is closer to the actual line path, the decision parameter will be positive and we have to choose the $y_{k+}$1 value for the coordinateposition $y_k$ $_{+1}$. Thus the value of $y_{k+1} - y_k$ = 1 for a decision parameter with a positive sign. The calculation of decision parameters has to perform recursively to get each integer x position starting from the left end of the line path. The value of decision parameter pkfor the initial or the starting coordinates ($x_0$, $y_0$) can obtain from the equation (12)

$$p_0 = \Delta x[2m(x_0+1) + 2b - 2y_0 - 1]$$
$$= \Delta x[2(mx_0 + b - y_0) + 2m - 1]$$
$$= \Delta x.0 + 2.\frac{\Delta y}{\Delta x} - 1 \quad \text{Where } mx_0 + b - y_0 = 0 \text{ and } m = \frac{\Delta y}{\Delta x}$$
$$= 2\Delta y - \Delta x$$

### 3.7.1 Steps involved in Bresenham's Line drawing algorithm

1. Required two pixels coordinates as input. Consider the left endpoint coordinate as ($x_0$, $y_0$).
2. ($x_0$, $y_0$) will be the first point in the line path
3. Compute the values for the terms:
   $\Delta y, \Delta x, 2\Delta y, 2\Delta x,$ and $2\Delta y - 2\Delta x,$ and p0the decision parameter for the initial point of the line path.
4. For each pixel coordinate $x_k$ along the line path, the following steps will perform recursively until the right endpoint. Initially k=0 or $\Delta$ x times.

4.1 if $p_k < 0$, the next point to be plot at position ($x_{k+1}$, $y_k$) and $p_{k+1} = p_k + 2\Delta y$

4.2 otherwise, the next point is to be a plot at position ($x_{k+}$ 1, $y_k$ $_{+1}$) and $p_{k+1} = p_{k+2}\Delta y - 2\Delta x$

Example 3: Draw a line with the endpoints (20, 10) and (30, 18), using Bresenham's line drawing algorithm.

Solution: Here it is given that the starting point coordinate is (20,

10), and the right endpoint is (30, 18).

So the $\Delta x = 30 - 20 = 10$ and $\Delta y = 18 - 10 = 8$

Initial decision parameter $p0 = 2\Delta y - \Delta x = 16 - 10 = 6$

And $2\Delta y = 16$, $2\Delta x = 20$

Therefore $2\Delta y - 2\Delta x = 16 - 20 = -4$

Initially plotting the pixel at (20, 10) and determining the successive pixel positions along the line path using the decision parameter is shown in the following table.

| K | $P_k$ | $(x_{k+1}, y_{k+1})$ | | K | $P_k$ | $(x_{k+1}, y_{k+1})$ |
|---|---|---|---|---|---|---|
| 0 | 6 | (21, 11) | | 5 | 6 | (26, 15) |
| 1 | 2 | (22, 12) | | 6 | 2 | (27, 16) |
| 2 | -2 | (23, 12) | | 7 | -2 | (28, 16) |
| 3 | 14 | (24, 13) | | 8 | 14 | (29, 17) |
| 4 | 10 | (25, 14) | | 9 | 10 | (30, 18) |

Example 4: Draw a line using Bresenham's Line algorithm with the endpoint coordinates (1, 1) and (8, 5).

Solution: Here it is given that the starting point coordinate is (1, 1), and the right endpoint is (8, 6).

So the $\Delta x = 8 - 1 = 7$ and $\Delta y = 5 - 1 = 4$

Initial decision parameter $p_0 = 2\Delta y - \Delta x = 8 - 7 = 1$

And $2\Delta y = 8$, $2\Delta x = 2$

Therefore $2\Delta y - 2\Delta x = 8 - 14 = -6$

Initially plotting the pixel at (1, 1) and determining the successive pixel positions along the line path using the decision parameter is shown in the following table.

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

For each value of $p_k$, if$p_k < 0$, next point is plotted at $(x_k+1, y_k)$, otherwise, at $(x_k+1, y_k+1)$



## CHECK YOUR PROGRESS

**Q 1:** Choose the correct options for the following questions:

i. What is the formula for calculating the slope 'm' of a line?

A. m = dx / dy

B. m = dy / dx

C. y = mx + c

D. None of the above

ii. Which of the following statements is not true to the Digital Differential Analyzer (DDA) algorithm?

A. It is an incremental method of scan conversion of line

B. In this method, the calculation is performed at each step but by using the results of previous steps.

C. This algorithm can also be used to draw a circle

D. None of the above

iii. Which of the following options is not correct according to the definition of Bresenham's line drawing algorithm?

A. It gives a close approximation of points of the line by determining the n-dimensional raster that should be selected.

B. It is an incremental error algorithm.

C. It gives exact line points.

D. None of the above

iv. What is the equation for a circle?

A. y = mx + c

B. $y^2 = r^2 - x^2$

C. $rx^2y^2 = rx^2ry^2 - ry^2x^2$

D. None of the above

v. Which of the following properties is followed by the ellipse?

A. 4 - symmetry property

B. 8 - symmetry property

C. 6 - symmetry property

D. All of the above

vi. Which of the following are the methods to define a circle?

A. Using the polynomial method.

B. Using polar coordinates methods.

C. Both a. and b.

D. None of the above

Example 5: Calculate the points between the starting coordinates (9, 18) and ending coordinates (14, 22) using Bresenham's line drawing algorithm.

Solution: Given that the Starting coordinates = $(X_0, Y_0) = (9, 18)$ and the Ending coordinates = $(X_n, Y_n) = (14, 22)$

Calculate $\Delta X$ and $\Delta Y$ from the given coordinates.

- $\Delta X = X_n - X0 = 14 - 9 = 5$
- $\Delta Y = Y_n - Y0 = 22 - 18 = 4$

Calculate the initial decision parameter P0

$= 2\Delta Y - \Delta X$

$= 2 \times 4 - 5$

$= 3$

So, the initial decision parameter Pk = 3

As $P_k >= 0$, Therefore,

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

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 |

Example 6:Calculate the points between the starting coordinates (20, 10) and ending coordinates (30, 18) using Bresenham's line drawing algorithm.

Solution: Given-

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

Calculate $\Delta X$ and $\Delta Y$ from the given input.

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

Calculate the initial decision parameter Pk

$= 2\Delta Y - \Delta X$

$= 2 \times 8 - 10$

$= 6$

So, initial decision parameter Pk = 6

As $P_k >= 0$, Therefore

$P_{k+1} = P_k + 2\Delta Y - 2\Delta X = 6 + (2 \times 8) - (2 \times 10) = 2$ and

- $X_{k+1} = X_k + 1 = 20 + 1 = 21$
- $Y_{k+1} = Y_k + 1 = 10 + 1 = 11$

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 |

### 3.7.2 Advantages and disadvantages of Bresenham's Line drawing algorithm

Bresenham's line drawing algorithm overcomes the drawbacks of the DDA line drawing algorithm. It is implemented using an incremental method for scan conversion lines than the DDA algorithm. It is faster and more efficient than the DDA line drawing algorithm. It uses only integer calculations such as addition, subtraction, and bit shifting operations to give an accurate result. The calculation for the terms 2?y - 2?x and 2?y are computed once for each line. This algorithm can be used to generate circles and different curves.

The main drawback of this algorithm is that it produces a jagged or stair-step line display.



**EXERCISE**

**Question3:** Draw a line segment using Bresenham's line drawing algorithm with the starting coordinate (30, 25) and the endpoint coordinate (10, 20).

**Question4:** Draw a line segment using Bresenham's line drawing line drawing algorithm with the starting coordinate (30, 25) and the endpoint coordinate (10, 20).

**Question5:** Draw a line segment using Bresenham's line drawing line drawing algorithm with the starting coordinate (10, 15) and the endpoint coordinate (10, 25).



## 3.8 LET US SUM UP

- In a graphics package, the functions which are used to describe the various picture components are known as graphics output primitives or simply primitives.
- A straight line can express using the expression y = mx + c, where m denotes the slope of the line and c is a constant.

- The slope of a line can be defined as $\Delta y / \Delta x$.
- The Digital Differential analyser (DDA) algorithm can be used to draw a straight line in computer graphics.
- DDA used an incremental approach to calculate the pixel coordinates on the line path.
- In the DDA algorithm, the round-off operation and floating-point arithmetic consume time.
- A highly accurate and efficient line drawing algorithm is Bresenham's line drawing algorithm.
- The sign of the decision parameter determines the coordinate position of the next pixel on the line path.
- Bresenham line algorithm gives a jagged or stair-steps appearance to the displayed line.
- Using the Bresenham algorithm, we can generate a circle also.



## 3.9 FURTHER READINGS

1) Computer Graphics, C Version, and Author: Donald D Hearn and M. Pauline Baker, Publisher: Pearson.
2) Introduction to computer graphics and multimedia, and AuthorAnirbanMukhopadhyay, Arup Chattopadhyay, Publisher: Vikas Computer graphics Multimedia and Animation, Author: Malay K. Pakhira, Publisher: PHI



## 3.10 ANSWERS TO CHECK YOUR PROGRESS

**Ans to Q No 1:** i.) B. m = dy / dx

**Ans to Q No 2:** ii) C. This algorithm can also be used to draw a circle.

**Ans to Q No 3:** iii) C. It gives exact line points

**Ans to Q No 4:** iv) B. y2 = r2 - x2

**Ans to Q No 5:** v) A. 4 - symmetry property

**Ans to Q No 6:** vi) C. Both a. and b.



## 3.11 MODEL QUESTIONS

**Short Questions** (Answer each question in around 150 words).

**Q 1:** Answer the following questions:

a. What do you mean by graphics output primitives?

b. Explain the DDA line algorithm.

c. Write the merits and demerits of the DDA algorithm.

d. Derive the equations for Bresenham's line algorithm.

e. What are the advantages and disadvantages of Bresenham's line algorithm?

f. Explain the midpoint circle algorithm.

g. How is the decision parameter used in the midpoint circle algorithm, explain.

h. What do you mean by the slope of a line? Explain different slope properties with a suitable diagram.

i. Define ellipse? Write any two properties of an ellipse.

j. Define a circle. Also, write atleast four properties of a circle.

k. Explain the midpoint ellipse algorithm.

*** ***** ***