Chapter 2 (10 Marks)

b

2. Derive a decision parameter for midpoint circle algorithm assuming the start position as (+r, 0) and points are to be generated along the curve path in clockwise direction. [6]
3. Digitize the given line end points (10, 10) and (20, 5) using Bresenham's line drawing Algorithms. [4]

2. Derive the P-value for Breshenham's line drawing Algorithm for $m < 0$ and $|m| > 1$. [6]
3. Using midpoint circle algorithms, calculate the co-ordinates to plot on first and second quadrant of a circle with center (6,7) and radius = 9. [5]

b

... access time per pixel. [3+3]

2. Derive an expression for drawing on ellipse. [10]
3. Explain 2-D viewing pipeline. Obtain window to viewport transformation matrix with necessary steps and figures. Give example. [3+4+3]

[4]

2. How do you apply symmetry concept while drawing circle? Calculate the point in the circumferences of the circle having radius 8 unit and center at (+8, 10) using midpoint circle algorithm. [2+6]
3. What do you mean by homogenous coordinates? ...

2. Digitize the endpoint (20, 10) and (30, 18) using Bresenham's algorithm. How the demerits of DDA is addressed in Bresenham's algorithm. [7+3]

2. Write an algorithm for drawing a circle. Using midpoint circle drawing algorithm, calculate the coordinates on the first quadrant of a circle having radius 8 and centre (10, 10). [4+6]

2. Write the advantages of Bresenham's line drawing algorithm. Digitize the Ellipse with radius $R_x = 12$ and $R_y = 7$ and center (19, 10). [2+8]

b

2. Explain the process of drawing ellipse in a raster graphics. Determine the pixel positions of following curve in first quadrant using mid-point algorithm. [4+6]

$$\frac{x^2}{64} + \frac{y^2}{36} = 1$$

2. Differentiate between DDA and Bresenhamline drawing algorithm. Explain Bresenham line drawing algorithm and use this algorithm to draw a line with end points (25,20) and (15,10). [2+8]

b

2. How symmetry property of circle reduces omplexity to draw a complete circle. Derive decision parameter for midpoint circle algorithm assuming the start position as (-r, 0) points are to be generated along the curve path in counter clockwise direction. [3+7]

2. How decision parameters can be used to draw circle? Calculate the points to draw a circle having radious 5 and center as (10, 5). [4+6]

b

2. Digitize the line with end points A(20,10) and B(30,18) using Bresenham algorithm. [10]

b

2. Derive and write midpoint algorithm for drawing a circle. [5+5]

2. How do you apply symmetry concept while drawing circle? Calculate the point in the circumferences of the circle having radius 8 unit and center at (-5, 10) using midpoint circle algorithm. [2+8]

b

1. Derive the Bresenham's decision parameter to draw a line moving from left to right and having negative slope. State the condition to identify you are in the second region of the ellipse using mid point algorithm. [8+2]

2. Compare between DDA and Bresenham's line drawing algorithm. Derive and write mid-point algorithm to draw ellipse. [10]

b

2. What is scan conversion? Derive the Bresenham's decision parameter to draw a line with negative slope and /m/>1. [2+8]

2. Compare between DDA and Bresenham's line drawing algorithm. Derive and write mid-point algorithm to draw ellipse. [10]

2. Mention the disadvantages of DDA method. Write the complete Bresenham's line drawing algorithm and using midpoint circle drawing algorithm calculate the co-ordinate on the first quadrant of a circle having radius 6 and centre (20,10) [2+4+4]

2. Digitize the endpoint (10, 18), (15, 8) using Bresenham's algorithm. [8]

1. Write Bresenham's line algorithm (you may assume $|m| < 1$). How the demerit of DDA algorithm is corrected in Bresenham's algorithm? [7+3]

2. Calculate all pixels of a circle in the first octant, proceeding to positive X axis direction. The radius = 30 and center at (10, 20). [10]

2. Write down the Bresenham's line drawing algorithm for drawing straight line with consideration of all the slope categories. [10]

1. Devise Bresenham's decision parameters for a straight line with negative slope with $|m| < 1$, applying left to right sampling. Assume that the line is in first quadrant.

Scan conversion in computer graphics refers to the process of converting geometric data, like points, lines, and polygons, into a raster image composed of pixels. Essentially, it transforms the vector representation of graphics into a format that can be displayed on a screen or output device.

DDA (Digital Differential Analyzer) Algorithm

The DDA algorithm is a popular method used in computer graphics for drawing lines between two points. It incrementally plots the points of a line by calculating the pixel positions along the line's path. The algorithm is based on calculating either the X or Y coordinate in small increments depending on the slope of the line.

Pros of the DDA Algorithm:

1. **Simplicity**: The DDA algorithm is easy to understand and implement.
2. **Precision**: It provides a good approximation for lines, especially when the line slope is not too steep.
3. **Straightforward Calculation**: It involves simple floating-point operations, making it computationally efficient for certain hardware.

Cons of the DDA Algorithm:

1. **Floating Point Operations**: The use of floating-point arithmetic can make the DDA algorithm slower on systems where floating-point operations are expensive.
2. **Accumulated Error**: Rounding the floating-point values to the nearest integer may introduce errors that accumulate over long lines, potentially resulting in a slightly inaccurate representation.
3. **Performance**: DDA can be slower than other algorithms like Bresenham's, especially for lines with steep slopes, as it involves division operations.

~~Alog~~ : Algorithm

Step 1: Accept start and end point co-ordinates (x_0, y_0) and (x_n, y_n)

Step 2: Calculate : $dx = x_2 - x_1$

$dy = y_2 - y_1$

$m = dy / dx$

Step 3 : If abs(dx) > abs(dy)

then,

step ⟵ $k = abs(dx)$

else

$k = abs(dy)$

Step 4 : current point (x_p, y_p) and next point (x_{p+1}, y_{p+1}) find the next point by the following cases

Case I	Case II	Case III
if $m < 1$ then,	if $m > 1$ then	if $m = 1$ then
$x_{p+1} = x_p + 1$	$x_{p+1} = x_p + \frac{1}{m}$	$x_{p+1} = x_p + 1$
$y_{p+1} = y_p + m$		
	$y_{p+1} = y_p + 1$	$y_{p+1} = y_p + 1$

(1) Consider a line from (5, 6) to (8, 12). Use the sample DDA algorithm to rastrize the line

soln:

Step 1: starting point $(x_0, y_0) = (5, 6)$
end point $(x_n, y_n) = (8, 12)$

Step 2:

$dx = 8 - 5 = 3$
$dy = 12 - 6 = 6$

$m = \frac{dy}{dx} = \frac{6}{3} = 2$

Step 3:
$K = abs(dy) = 6 \rightarrow$ steps

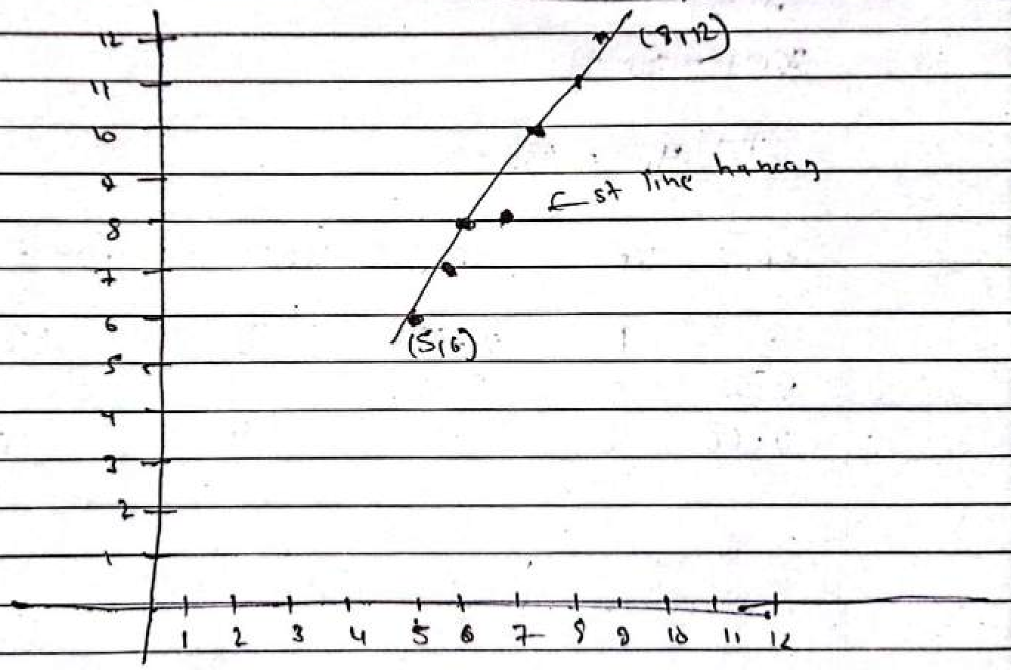
Step 4: as $m > 1$

$x_{p+1} = x_p + \frac{1}{m}$
$y_{p+1} = y_p + 1$

P.T.O

$5.5 \rightarrow 6$
$5.4 \rightarrow 5$

x_p	y_p	x_{p+1}	y_{p+1}	round off
5	6	5.5	7	(6, 7)
5.5	7	6	8	(6, 8)
6	8	6.5	9	(7, 9)
6.5	9	7	10	(7, 10)
7	10	7.5	11	(8, 11)
7.5	11	8	12	(8, 12)



$(2,1) \rightarrow (3,4)$

~~$dx = 3$~~

step 1: starting $(x_0, y_0) = (2,1)$

end $(x_n, y_n) = (3,4)$

$dx = 3-2 = 1$, $dy = 4-1 = 3$

$K = abs(dy) = 3$ $\quad$ $m = \frac{dy}{dx} = \frac{3}{1} = 3$

$m > 1$

$x_{p+1} = x_p + \frac{1}{m}$

~~x_{p+1}~~ $y_{p+1} = y_p + 1$

x_p	y_p	x_{p+1}	y_{p+1}	round off
2	1	2.33	2	(2,2)
2.33	2	2.66	3	(3,3)
2.66	3	3	4	(3,4)

Bresenham's Line Algorithm

Bresenham's Line Algorithm is an efficient algorithm used to draw lines on a raster display by determining the points that need to be plotted between two endpoints. It is an integer-based algorithm that avoids floating-point arithmetic, making it faster and more suitable for systems where integer operations are more efficient.

Pros of Bresenham's Algorithm:

1. **Efficiency**: Bresenham's algorithm is highly efficient because it only uses integer arithmetic (addition, subtraction, and bit-shifting), avoiding costly floating-point operations.
2. **Accuracy**: The algorithm accurately determines the closest pixels to the ideal line, minimizing visual artifacts.
3. **Versatility**: It can be easily adapted to draw circles, ellipses, and other curves, as well as lines with different slopes (positive, negative, shallow, or steep).
4. **Speed**: Due to its use of simple arithmetic operations, the algorithm runs faster on systems with limited computational power.

Cons of Bresenham's Algorithm:

1. **Complexity**: While Bresenham's algorithm is more efficient than the DDA algorithm, it is also more complex to implement, especially when extended to handle different line slopes and orientations.
2. **Limited to Integer Increments**: Bresenham's algorithm is tailored for raster devices where pixel positions are integer-based. For high-precision graphics applications that require sub-pixel accuracy, additional adjustments are needed.
3. **Special Cases Handling**: The algorithm needs special handling for lines with extreme slopes (near vertical or horizontal) and for lines that are purely vertical or horizontal.

Algorithm (~~[illegible]~~)

(1) Start co-ordinate (x_0, y_0)
end co-ordinate (x_n, y_n)

(2) $\Delta x = x_n - x_0$, $\Delta y = y_n - y_0$

Slope $(m) = \frac{\Delta y}{\Delta x}$

(3) Decision parameter $(P_0) = 2\Delta y - \Delta x$

(i) If $m < 1$ for this

case I

$P_k < 0$ then,
$x_{k+1} = x_k + 1$
$y_{k+1} = y_k$
$P_{k+1} = P_k + 2\Delta y$

case II

$P_k \geq 0$
$x_{k+1} = x_k + 1$
$y_{k+1} = y_k + 1$
$P_{k+1} = P_k + 2\Delta y - 2\Delta x$

(ii) if $m \geq 1 \rightarrow P_0 = 2\Delta x - \Delta y$

case I

$P_k < 0$ then,
$x_{k+1} = x_k$
$y_{k+1} = y_k + 1$
$P_{k+1} = P_k + 2\Delta x$

case II

$P_k \geq 0$
$x_{k+1} = x_k + 1$
$y_{k+1} = y_k + 1$
$P_{k+1} = P_k + 2\Delta x - 2\Delta y$

(i) (1, 1) (5, 3)

$m = \frac{\Delta y}{\Delta y} = 0.5 < 1$ $\left(\begin{matrix}\Delta x = 4 \\ \Delta y = 2\end{matrix}\right)$

$P_0 = 2\times2 - 4 = 0 \rightarrow$ case II ≥ 0

P_K	x_K	y_K	x_{K+1}	y_{K+1}	P_{K+1}	(x_{K+1}, y_{K+1})
0	1	1	2	2	-4	(2, 2)
-4	2	2	3	2	0	(3, 2)
0	3	2	4	3	-4	(4, 3)
-4	4	3	5	3	0	(5, 3)

② (0,0) to (2,3)

$m = \frac{\Delta y}{\Delta x} = \frac{3-0}{2-0} = \frac{3}{2} = 1.5 > 1$

$P_0 = 2\Delta x - \Delta y = 2\times2 - 3 = 1$ $(P > 0)$

P_K	x_K	y_K	x_{K+1}	y_{K+1}	P_{K+1}	(x_{K+1}, y_{K+1})
1	0	0	1	1	-2	(1, 1)
-2	1	1	1	2	2	(1, 2)
2	1	2	2	3	0	(2, 3)

Date ______
Page ______

(2, 4) to (8, 20)

$\Delta x = 8 - 2 = 6$, $\Delta y = 20 - 4 = 16$ | $m = \frac{\Delta y}{\Delta x} = 2.667 > 1$

$P_0 = 2\Delta x - \Delta y = 2 \times 6 - 16 = -4$

PK	xK	yK	xK+1	yK+1	PK+1	( )
-4	2	4	2	5	8	(2, 5)
8	2	5	3	6	-12	(3, 6)
-12	3	6	3	7	0	(3, 7)
0	3	7	4	8	-20	(4, 8)
-20	4	8	4	9	-8	(4, 9
						⋮
						(8, 20)

Feature	DDA (Digital Differential Analyzer)	Bresenham's Line Algorithm (BLA)
Basic Principle	Uses floating-point arithmetic to calculate intermediate points along a line.	Uses integer arithmetic to determine the points of the line.
Calculation Method	Incremental calculations using slope (m) and y-intercept (c).	Decides the closest pixel to the line by minimizing the error.
Performance	Slower due to floating-point operations.	Faster due to the use of integer arithmetic.
Accuracy	Less accurate due to rounding errors in floating-point operations.	More accurate as it uses only integer calculations.
Complexity	Simpler to understand and implement.	More complex but more efficient in execution.
Resource Usage	Requires more memory due to floating-point calculations.	Less memory-intensive due to integer calculations.
Line Appearance	Produces smoother lines with possible aliasing due to floating-point rounding.	Produces a more "stepped" line appearance, but with less aliasing.
Usage	Typically used when floating-point hardware is available and speed is not a critical factor.	Preferred in systems where performance is critical, especially in real-time applications.
	expensive	cheaper
	need special handling for each octant	handles all octant uniformly
	less accurate and efficient	more accurate and efficient
	also uses multiplication and division in it’s operation	only uses subtraction and addition in it’s operation

Data ____
Page ____

Midpoint Circle Drawing Algorithm

→ The equation of a circle centered at the origin with radius r is given by

$$f(x,y) = x^2+y^2-r^2 = 0$$

$f(x,y) = 0$ (point (x,y) lies on the circle)
$f(x,y) < 0$ (point (x,y) is inside circle)
$f(x,y) > 0$ (point (x,y) is outside the circle)

The decision parameter P_K helps determine the next pixel to plot. At each step based on P_K the algorithm chooses between moving horizontally to (x_K+1, y_K) or diagonally to (x_K+1, y_K-1)

$$\text{midpoint} = \left(\frac{x_K+1 + x_K+1}{2}, \frac{y_K + y_K-1}{2} \right)$$

$$= \left(x_K+1, y_K - \frac{1}{2} \right)$$

P_K is defined as the circle function evaluated at midpoint between these two potential points

$$P_K = x_m^2 + y_m^2 - r^2$$

$$= (x_K+1)^2 + \left(y_K-\frac{1}{2}\right)^2 - r^2 \quad \text{—— (i)}$$

$P_{k+1} = (x_{k+1} + 1)^2 + (y_{k+1} - \frac{1}{2})^2 - r^2$

Now,

$P_{k+1} - P_k = (x_{k+1} + 1)^2 - (x_k + 1)^2 + (y_{k+1} - \frac{1}{2})^2$

$- (y_k - \frac{1}{2})^2 - \cancel{r^2} + \cancel{r^2}$

We know,

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

$= (x_k + 1 + 1)^2 - (x_k + 1)^2 + (y_{k+1} - \frac{1}{2})^2$

$- (y_k - \frac{1}{2})^2$.

$= \cancel{x_k^2} + 4x_k + 4 - \cancel{x_k^2} - 2x_k - 1 + y_{k+1}^2 -$
$y_{k+1} + \cancel{\frac{1}{4}} - y_k^2 + y_k - \cancel{\frac{1}{4}}$

or, $P_{k+1} = P_k + 4x_k + 3 - 2x_k + y_{k+1}^2 - y_{k+1}$
$- y_k^2 + y_k$

$\therefore$ $P_{k+1} = P_k + 2x_k + 3 + y_{k+1}^2 - y_{k+1} - y_k^2 + y_k$

If $P_k < 0$

$y_{k+1} = y_k$

$P_{k+1} = P_k + 2x_k + 3 + \cancel{y_k^2} - \cancel{y_k} - \cancel{y_k^2} + \cancel{y_k}$
$P_{k+1} = P_k + 2x_k + 3$
$P_{k+1} = P_k + 2(x_{k+1} - 2) + 3$

$\therefore \boxed{P_{k+1} = P_k + 2x_{k+1} + 1}$

If $P_k \geq 0 \Rightarrow y_{k+1} = y_k - 1$

$P_{k+1} = P_k + 2x_k + 3 + (y_k - 1)^2 - y_k^2 + y_k$

$= P_k + 2x_k + 3 + y_k - 2y_k + 1 - y_k + y_k$

$= P_k + 2x_k$

$P_{k+1} = P_k + 2x_k + 3 + (y_k-1)^2 - (y_k-1) - y_k^2 + y_k$

$= P_k + 2x_k + 3 + \cancel{y_k^2} - 2y_k + 1 - \cancel{y_k} + 1 - \cancel{y_k^2} + \cancel{y_k}$

$= P_k + 2x_k + 5 - 2y_k$

$= P_k + 2(x_k - y_k) + 5$

$= P_k + 2(x_{k+1} - 1 - y_{k+1} - 1) + 5$

$\boxed{P_{k+1} = P_k + 2(x_{k+1} - y_{k+1}) + 1}$

$\left(\begin{array}{l} x_{k+1} = x_{k+1} \\ y_{k+1} = y_k - 1 \end{array} \right)$

Initial decision parameter (P_0)

put $x_k = 0$, $y_k = r$ in P_k

$P_0 = (0+1)^2 + \left(r - \frac{1}{2}\right)^2 - r^2$

$= 1 + \cancel{r^2} - r + \frac{1}{4} - \cancel{r^2}$

$= \frac{5}{4} - r$

$\therefore \boxed{P_0 = 1 - r}$ $\quad \frac{5}{4} \approx 1)$

Algorithm

(1) plot initial point (x_i, y_i) such that $x_i = 0$, $y_i = r$

(2) $p_0 = 1 - r$

(3) If $p_k < 0$ then

$x_{k+1} = x_k + 1$
$y_{k+1} = y_k$
$p_{k+1} = p_k + 2x_{k+1} + 1$

If $p_k \geq 0$

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

(4) Repeat steps 3, 4 untill x becomes greater than or equal to y

(5) To plot the complete cirile reflect each octant of the first octant into 7 other octant making use of 8 way symmetry

plot the 1st octant of a circle centered at origin having radius 10 units.

plot $(0, r) \rightarrow (0, 10) \rightarrow (x_0, y_0)$

$P_0 = 1 - r = 1 - 10 = -9$

P_k	x_k	y_k	x_{k+1}	y_{k+1}	P_{k+1}	(x_{k+1}, y_{k+1})
-9	0	10	1	10	-6	(1, 10)
-6	1	10	2	10	-1	(2, 10)
-1	2	10	3	10	6	(3, 10)
6	3	10	4	9	-3	(4, 9)
-3	4	9	5	9	8	(5, 9)
8	5	9	6	8	5	(6, 8)
5	6	8	7	7	6	(7, 7)

Until $x_{k+1} \geq y_{k+1}$

How do you apply symmetry and how does it reduce computational steps?

→ Euta octant plot garepachi purai circle ko points can be plot using reflecting

→ The symmetry of a circle allows calculations and operations to be performed on a fraction of it's circumference and then mirrored or rotated reducing computational steps by exploiting identical segments.

Date
Page

Digitize a circle with radius 6 centered (20, 20) in 1st quadrant

$r = 6 \quad (x_c, y_c) = (20, 20)$

$(x_0, y_0) = (0, 6)$ 1st point is circumference of 1st octant center at (0,0).

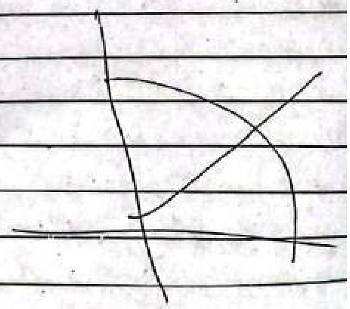
$P_0 = 1 - r = 1 - 6 = -5$

P_k	x_k	y_k	x_{k+1}	y_{k+1}	P_{k+1}	at (0,0) (x_{k+1}, y_{k+1})	at (20,20) (x_{k+1}, y_{k+1})
−5	0	6	1	6	−2	(1, 6)	(21, 26)
−2	1	6	2	6	3	(2, 6)	(22, 26)
3	2	6	3	5	0	(3, 5)	(23, 25)
0	3	5	4	4	1	(4, 4)	(24, 24)

$x_{k+1} \geq y_{k+1}$

For 2nd octant by symmetry at 45° reflection:

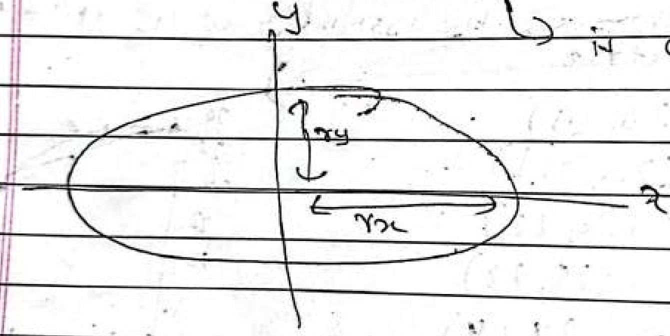
(20, 26) → (26, 20)
(21, 26) → (26, 21)
(22, 26) → (26, 22)
(23, 25) → (25, 23)
(24, 24) → (24, 24)



Midpoint Ellipse Algorithm

Midpoint Ellipse Drawing Algorithm.

is an elongated circle



Eqn of ellipse centered at (0,0)

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (a = r_x \quad b = r_y)$$

major axis $= 2a = 2r_x$
minor axis $= 2b = 2r_y$
semi major axis $= a = r_x$
semi minor axis $= b = r_y$

$$\frac{x^2b^2 + y^2a^2}{a^2b^2} = 1 \Rightarrow b^2x^2 + a^2y^2 = a^2b^2$$
$$\Rightarrow b^2x^2 + a^2y^2 - a^2b^2 = 0.$$

$\boxed{r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2 = 0}$ —(i)

If we put any point in eqn (i)

$= 0$ point lies on ellipse

< 0 inside the ellipse

> 0 outside the ellipse

4- quadrant symmetric, not 8- octant symmetric like circle

Difference between circle and ellipse

* circle has 8-way symmetry
* Ellipse has 4-way symmetry

* In circle we need to plot only 1 octant at any quadrant, but in ellipse we need to plot 2

octants i.e 1 complete quadrant to plot entire ellipse.

Midpoint ellipse drawing algorithm Derivation

We know,

$r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2 = 0$ — (*)

Differentiating w.r.t. x

$r_y^2 2x + r_x^2 \times \frac{dy^2}{dy} \times \frac{dy}{dx} - 0 = 0$

$2r_y^2 x + 2r_x^2 y \frac{dy}{dx} = 0$

$\therefore \frac{dy}{dx} = \frac{-2r_y^2 x}{2r_x^2 y}$ — (i)

At the boundary region between 1 and 2, slope = -1

$\frac{dy}{dx} = -1 = \frac{-2r_y^2 x}{2r_x^2 y}$

or $2r_y^2 x = 2r_x^2 y$

We move out of region 1 when $2r_y^2 x > 2r_x^2 y$.

We move out of region 2 when $2r_y^2 x < 2r_x^2 y$

Region 1 → $(x_k + 1, y_k)$, $(x_k + 1, y_k - 1)$

mid point $(x_k + 1, y_k - \frac{1}{2})$

put midpoint in (*)

$$r_y^2 (x_k + 1)^2 + r_x^2 \left(y_k - \frac{1}{2}\right)^2 - r_x^2 r_y^2 = P_{1k} \quad \text{--- (2)}$$

$$r_y^2 (x_{k+1} + 1) + r_x^2 \left(y_{k+1} - \frac{1}{2}\right)^2 - r_x^2 r_y^2 = P_{1k+1}$$

we know,

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

$$r_y^2 (x_k + 2)^2 + r_x^2 \left(y_{k+1} - \frac{1}{2}\right)^2 - r_x^2 r_y^2 = P_{1k+1} \quad \text{--- (2)}$$

Now,

$$P_{1k+1} - P_{1k} = r_y^2 (x_k + 2)^2 - r_y^2 (x_k + 1)^2 + r_x^2 \left(y_{k+1} - \frac{1}{2}\right)^2 - r_x^2 \left(y_k - \frac{1}{2}\right)^2$$

$$= r_y^2 \left\{ (x_k + 2)^2 - (x_k + 1)^2 \right\} + r_x^2 \left\{ \left(y_{k+1} - \frac{1}{2}\right)^2 - \left(y_k - \frac{1}{2}\right)^2 \right\}$$

$$= r_y^2 \left\{ \cancel{x_k^2} + 4x_k + 4 - \cancel{x_k^2} - 2x_k - 1 \right\} + r_x^2 \left\{ y_{k+1}^2 - y_{k+1} + \cancel{\frac{1}{4}} - y_k^2 + y_k - \cancel{\frac{1}{4}} \right\}$$

$$= r_y^2 \left\{ 2x_k + 3 \right\} + r_x^2 \left\{ y_{k+1}^2 - y_{k+1} - y_k^2 + y_k \right\}$$

When, $P_{1k} < 0 \rightarrow y_{k+1} = y_k$

$P_{1k+1} - P_{1k} + r_y^2 \{2x_{k+1} + 1\} + r_x^2$

$\{ \cancel{y_k^2} - \cancel{y_k} - \cancel{y_k^2} + \cancel{y_k} \}$

$\therefore \boxed{P_{1k+1} = P_{1k} + r_y^2 2x_{k+1} + r_y^2}$

When

$P_{1k} \geq 0 \rightarrow y_{k+1} = y_k - 1$

$P_{1k+1} - P_{1k} = r_y^2 \{2x_{k+1} + 1\} + r_x^2 \cdot \{ (y_k - 1)^2 - (y_k - 1) - y_k^2 + y_k \}$

$\therefore \boxed{P_{1k+1} = P_{1k} + 2x_{k+1} r_y^2 + r_y^2 - 2y_{k+1} r_x^2}$

Initial decision parameter

put $(0, r_y)$ in P_{1k}

$P_{1k} = r_y^2 (x_k + 1)^2 + r_x^2 (y_k - \frac{1}{2})^2 - r_x^2 r_y^2$

$P_{10} = r_y^2 (0 + 1)^2 + r_x^2 (r_y - \frac{1}{2})^2 - r_x^2 r_y$

$\therefore \boxed{P_{10} = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2}$

Region 2 : $(x_k, y_k - 1)$, $(x_{k+1}, y_k - 1)$

$\longrightarrow (x_k + \frac{1}{2}, y_k - 1)$

↑
mid point

P_{2k} → mid point to put in

$r_y^2 x^2 + r_x^2 y - r_x^2 r_y^2$

P_{2k+1} → similar

put in ($y_{k+1} = y_k - 1$)

now,

~~$P_{2k+1} - P_{2k} =$~~

~~$P_{2k+1} = P_{2k} + \ldots \{x_k^2\}$~~

~~$P_{2k+1} = P_{2k} - 2y_{k+1} r_x^2 + r_x^2$~~

~~If $P_{2k} > 0 \rightarrow x_{k+1} = x_k$~~

$P_{2k+1} = P_{2k} + r_y^2 \{x_{k+1}^2 + x_{k+1} - x_k^2 - x_k\} + r_x^2 - 2y_{k+1} r_x^2$

If $P_{2k} > 0 \rightarrow x_{k+1} = x_k$

$P_{2k+1} = P_{2k} + r_y^2 \{\cancel{x_k^2} + \cancel{x_k} - \cancel{x_k^2} - \cancel{x_k}\} - 2y_{k+1} r_x^2 + r_x^2$

$\therefore$ $\boxed{P_{2k+1} = P_{2k} - 2y_{k+1} rx^2 + rx^2}$

If $P_{2k} \leq 0 \rightarrow x_{k+1} = x_k + 1$

$\boxed{P_{2k+1} = P_{2k} + 2x_{k+1} ry^2 - 2y_{k+1} rx^2 + rx^2}$

Initial decision parameter is put (x, y) in eqn (i) of P_{2k}

$\boxed{P_{2k} = ry^2 \left(x + \frac{1}{2}\right)^2 + rx^2 (y-1)^2 - rx^2 ry^2}$

Algorithm :

→ read radii rx and ry

→ Initialize starting point at region 1 as :-
$x = 0$, $y = ry$ $(0, ry)$

calculate $P_{10} = ry^2 - rx^2 ry + \frac{1}{4} rx^2$

calculate $dx = 2ry^2 x$ $dy = 2rx^2 y$

Repeat while $(dx < dy)$

* plot (x, y)

if $(P_1 < 0)$

{

$x = x+1$, $y = y$
update $dx = 2ry^2x$, ~~dopos~~ dy = same
$P_1 = P_1 + \underbrace{2ry^2x}_{dx} + ry^2$
}

else $\rightarrow P_1 \geq 0$

{
$x = x+1$
$y = y-1$
update $dx \rightarrow 2ry^2x$
update $dy \rightarrow 2rx^2y$
$P_1 = P_1 + dx - dy + ry^2$
}

when $(dx \geq dy)$ plot region 2 as :-

$P_{20} = ry^2\left(x+\frac{1}{2}\right)^2 + rx^2(y-1)^2 - rx^2ry^2$

repeat till $(y > 0)$

plot (x, y)
if $(P_2 > 0)$

{
$x = x$
$y = y-1$
update dy
dx = ~~same~~ same

$P_2 = P_2 - dy + rx^2$

}

else $\rightarrow P_2 \le 0$

{

$x = x+1$

$y = y-1$

$dy = 2rx^2y$

$dx = 2ry^2x$

$P_2 = P_2 + dx - dy + rx^2$

}

$P_{10} = -2252$ $dx<0$ $dx \geq 0$ ~~768~~ $dy = (dx \geq dy)$ so region 1

* Given $a = 8$, $b = 6$ i.e. $rx = 8$, $ry = 6$

(x_k, y_k)	Initial Decision parameter	final (Calculate) (P_{k+1})	(x_{k+1}, y_{k+1})	dx	dy
	332	-224			
(0, 6)	~~-2252~~	10	(1, 6)	72	768
(1, 6)	-224	-44	(2, 6)	144	768
(2, 6)	-44	208	(3, 6)	216	768
(3, 6)	208	-108	(4, 5)	288	640
(4, 5)	-108	288	(5, 5)	360	640
(5, 5)	288	244	(6, 4)	432	512
(6, 4)	244		(7, 3)	504	384

$dx \geq dy$

So we switch to region 2

stop when $y \le 0$

(x_k, y_k)	P_k (initial)	(x_{k+1}, y_{k+1}) (new formula)	dx	dy	final P_k
(7,3)	-23	(8,2)	576	256	361
(8,2)	361	(8,1)	576	128	297
(8,1)	297	(8,0)			
		stop since $y \le 0$			

Ellipse centered at (20,10) $a = 4$ (semi-major) $b = 8$ (semi-minor)

$r_x = 4$, $r_y = 8$

Initial (0, 8) $(x_c, y_c) = (20, 18)$

$P_{10} = -60$

$dx = 0$ $dy = 256$

$dx < dy$

So, we begin region 1

(x, y)	Pk	(xk+1, yk+1)	dx	dy	Pk	(xc, yc)
(0, 8)	-60	(1, 8)	128	256	132	(21, 18)
(1, 8)	132	(2, 7)	256	224	—	(22, 17)
new formula at R2			dy ≥ dy (region 2)			
(2, 7)	-48	(3, 6)	~~256~~ 384	192	160	(23, 16)
(3, 6)	160	(3, 5)	384	160	16	(23, 15)
(3, 5)	16	(3, 4)	384	128	-96	(23, 14)
(3, 4)	-96	(4, 3)	512	96	336	(24, 13)
(4, 3)	336	(4, 2)	512	64	288	(24, 12)
(4, 2)	288	(4, 1)	512	32	272	(24, 11)
(4, 1)	272	(4, 0)	—	—	—	(24, 10)

y = 0 (stop)

now the points are to be plotted