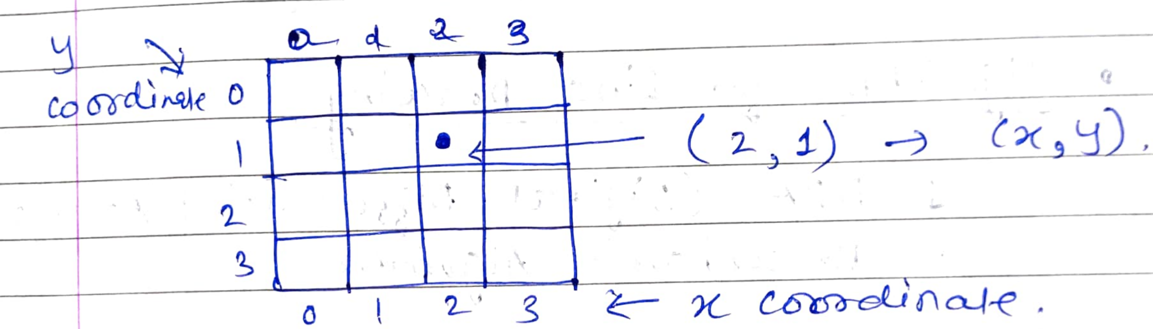
Chap 2.
Output Primitives

* **Scan conversion of point, line, circle & ellipse:**

- Each pixel on the graphics display does not represent a mathematical point. Scan converting a point involves illuminating the pixel that contains the point.

Ex:- point p(x, y) represent by the integer part of x & the integer part of y.



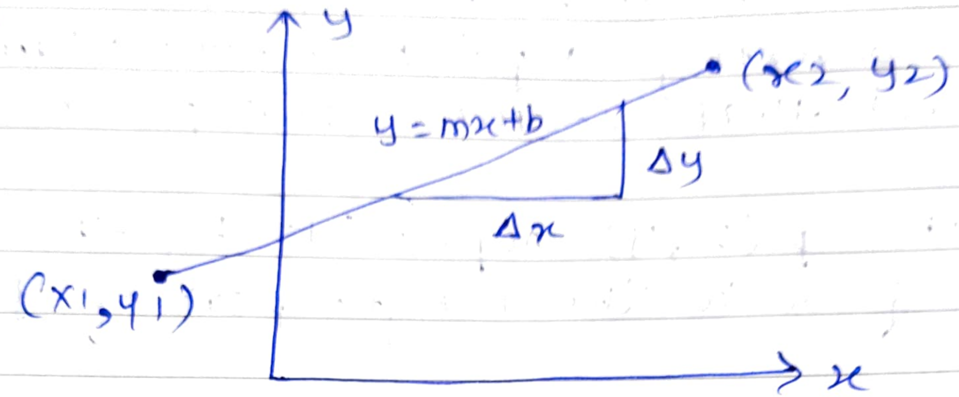
- Scan conversion is a process of representing graphics objects collection of pixels.

- The graphics objects are continuous. The pixel used are discrete. Each pixel can have either on or off state.

- The circuitry of the video display device of the computer is capable of converting binary values (0,1) into a pixel on & off information.
Using this ability graphics computer represent picture having discrete dots.

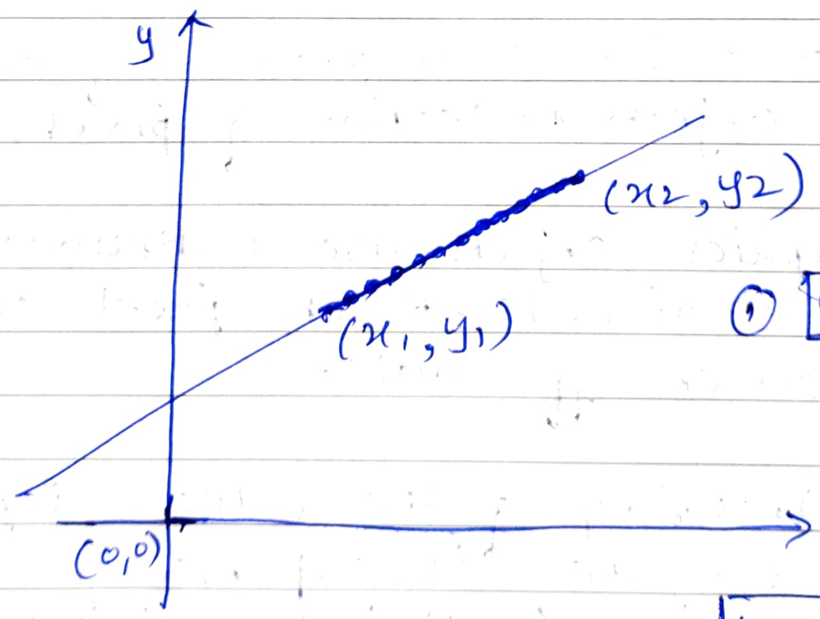
Scan Conversion of Line.

- A straight line may be defined by 2 endpoints & an equation.
  The equation of the line is used to determine the x, y coordinates of all the points that lie between these two endpoints.



Algorithm for Line Drawing

1. DDA (Digital Differential Analyzer)
2. Bresenham's Algorithm.



line eq^n

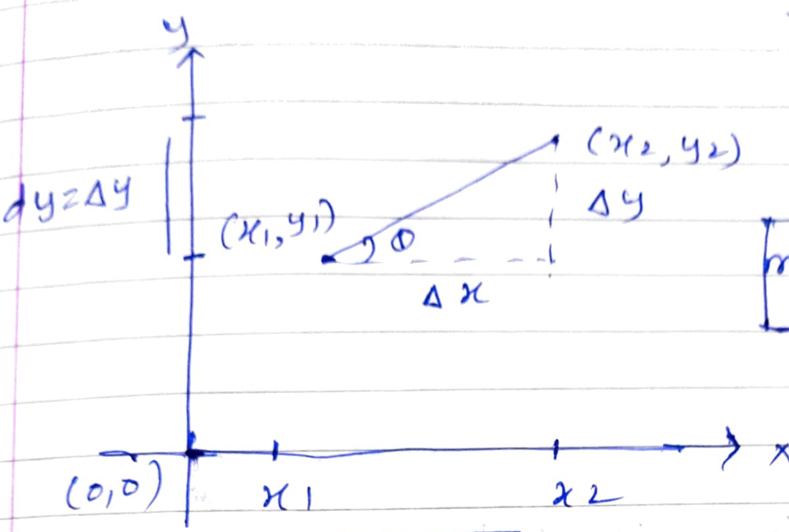
① $\boxed{y = mx + c}$ → intercept

↑ slope

② $y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1)$

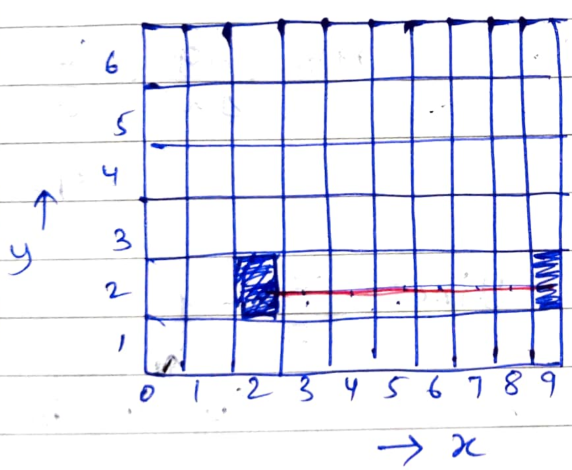
* DDA Algorithm

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



$\boxed{m = \frac{\Delta y}{\Delta x}}$ slope

$dx = \Delta x$
(distance bet^n x_1 & x_2)



(x_1, y_1) (x_2, y_2)

$(2, 2)$ $(9, 2)$

$\Delta x = 9 - 2 = 7$

$\Delta y = 2 - 2 = 0$

$m = \frac{\Delta y}{\Delta x} = \frac{0}{7}$

Steps = 7

$x_{inc} = \frac{7}{7} = 1$ $\qquad$ $y_{inc} = \frac{0}{7} = 0$

inc by 1 $\qquad$ not increa...

x	y
2	2
3	2
4	2
5	2
6	2
7	2
8	2
9	2

ex: (x_1, y_1) (x_2, y_2)

$(2, 5)$ $\qquad$ $(2, 12)$

$\Delta x = 2 - 2 = 0$ $\qquad$ $m = \frac{\Delta y}{\Delta x} = \frac{7}{0} = \infty$

$\Delta y = 12 - 5 = 7$

steps = 7

(y va...

$x_{inc} = \frac{0}{7} = 0$ $\qquad$ $y_{inc} = \frac{7}{7} = 1$

x not inc. $\qquad$ (y inc by 1)

x	y
2	5
2	6
2	7
2	8
2	9
2	10
2	11
2	12

ex:- (x_1, y_1) (x_2, y_2)

$(5, 4)$ $(12, 7)$

$\Delta x = 12 - 5 = 7$

$\Delta y = 7 - 4 = 3$

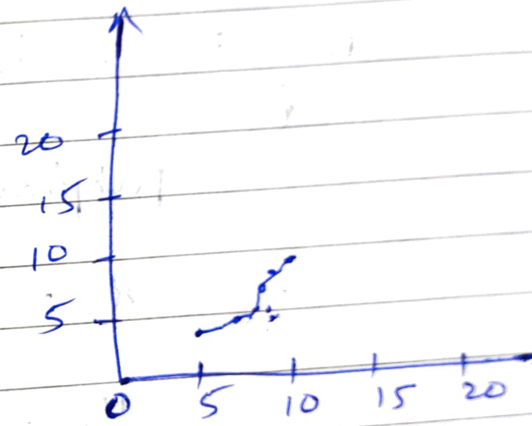
$m = \dfrac{\Delta y}{\Delta x} = \dfrac{3}{7} = 0.4$

steps 7

$x\,inc = \dfrac{7}{7} = 1$ $y\,inc = \dfrac{3}{7} = 0.4$

Here $m < 1$ (slope)

x	y	round (y)
5	4	4
6	4.4	4
7	4.8	5
8	5.2	5
9	5.6	6
10	6	6
11	6.4	6
12	6.8	7



* **Algorithm DDA** (x_1, y_1, x_2, y_2)

```
dx = x2 - x1
dy = y2 - y1

if ( abs (dx) > abs(dy))

    step = abs (dx)
else
    step = abs(dy)

xinc = dx / step

yinc = dy / step

for( i = 1; i <= step; i++)
{
    putpixel (x1, y1);

    x1 = x1 + xinc;

    y1 = y1 + yinc;
}
```

- { Algorithm takes xinc & yinc in floats calulation & floats take extra time than integer values. So Algo is slo
- We are getting points which are not accurate one so we have to round off
- Line will not smooth line.

- This is drawback of DDA Algo. So we go for Bresenham's Algorithm.

* (DDA) Algo (Practical)

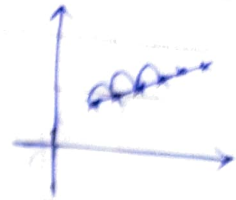
→ It is a scan conversion method for drawing line.

→ It has incremental approach

→ It analysed difference in pixel points.

→ straight line equation is,

$$y = mx + b$$



if the line has two end points $A(x_1, y_1)$ & $B(x_2, y_2)$

slope of the line is $m = \dfrac{y_2 - y_1}{x_2 - x_1}$

$$\therefore m = \frac{\Delta y}{\Delta x}$$

→ DDA based on calculation of values $\Delta x, \Delta y$.

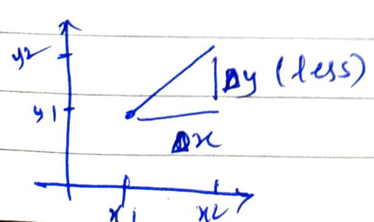
$$\therefore \Delta y = m * \Delta x \quad \text{—} \ (1)$$

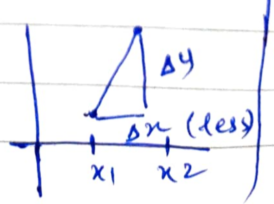
$$\Delta x = \frac{\Delta y}{m} \quad \text{—} \ (2)$$

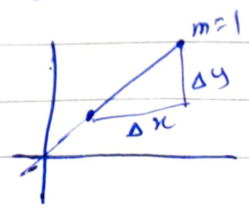
→ Given line have the slope +ve or -ve

If +ve, the Δx & Δy values are increased else decreased.

case (1) If $m < 1$	(2) if $m > 1$	(3) if $m = 1$
$x_n = x_1 + 1$	$x_n = x_1 + 1/m$	$x_n = x_1 + 1$
$y_n = y_1 + m$	$y_n \ y_1 + 1$	$y_n = y_1 + 1$







DDA Algorithm

1) ~~Steps~~ : Accept ~~start~~ & end point coordinates (x_1, y_1) & (x_2, y_2)

2) calculate $dx = x_2 - x_1$
$dy = y_2 - y_1$

3) if abs(dx) > abs(dy) // m<1
then K = abs(dx)
else
K = abs(dy)

4) calculate $\Delta x = \frac{dx}{K}$, $\Delta y = \frac{dy}{K}$

5) Initialize $x = x_1$, $y = y_1$

6) Display pixel (x, y)

7) $x = x + \Delta x$ $y = y + \Delta y$

8) Display pixel round(x), round(y)

9) Repeat step 7 & 8 'K' times.

Advantage

→ faster method when compared to direct use of line eq'n.

Q. Draw the line between (5, 5) & (10, 9) using DDA Algo.

→ $(x_1, y_1) - (5, 5)$ $(x_2, y_2) - (10, 9)$

$dx = x_2 - x_1 = 10 - 5 = 5$
$dy = y_2 - y_1 = 9 - 5 = 4$

$(dx > dy) \rightarrow (5 > 4) \rightarrow k = dx.$

$k = dx = 5$

$\Delta x = \frac{dx}{k} = \frac{5}{5} = 1$

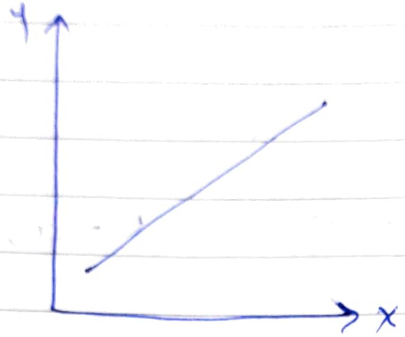
$\Delta y = \frac{dy}{k} = \frac{4}{5} = 0.8$

step. $\boxed{k = 5}$ plot(round(x), round(y))
$x + \Delta x$ $y + \Delta y$.

iteration	plot (x, y)	$x + \Delta x$	$y + \Delta y$
0	(5, 5)	5 + 1 = 6	5 + 0.8 = 5.8
1	(6, 6)	6 + 1 = 7	5.8 + 0.8 = 6.8
2.	(7, 7)	7 + 1 = 8	6.8 + 0.8 = 7.4
3	(8, 7)	8 + 1 = 9	7.4 + 0.8 = 8.2
4	(9, 8)	9 + 1 = 10	8.2 + 0.8 = 9
5	(10, 9)	10 + 1 = 11	9 + 0.8 = 9.8

* Bresenham's Line Drawing Algorithm

- Drawback of DDA algo i.e: floating point number (processing time is more th

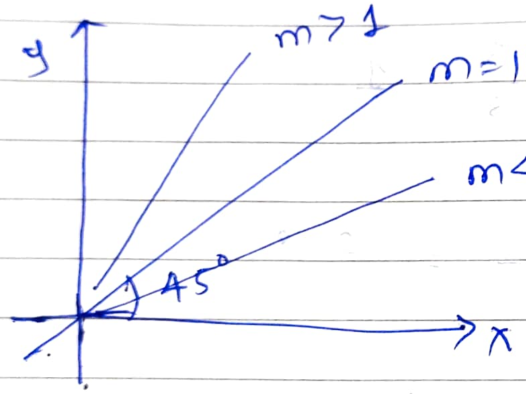


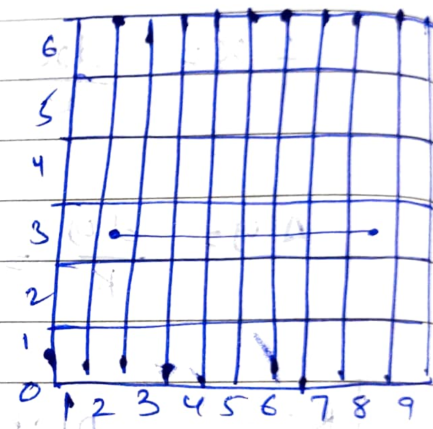
vector

$$\boxed{y = mx + c}$$

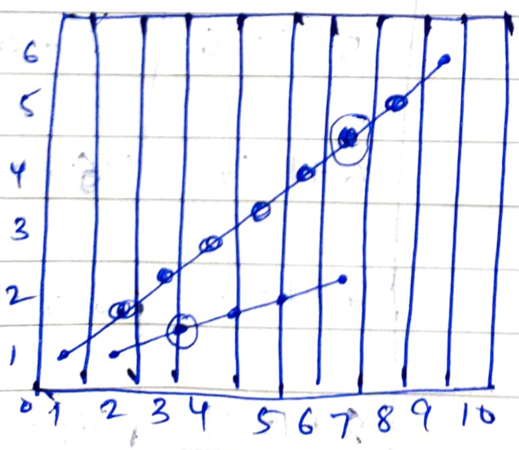
$$\frac{\Delta y}{\Delta x}$$

raster

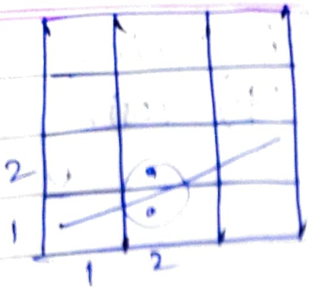


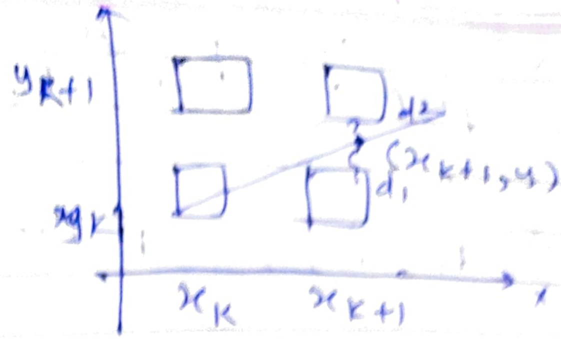


- converting $\boxed{y=mx+c}$ into screen co-ord is called Rasterization.



Line is going partially to another pixel als



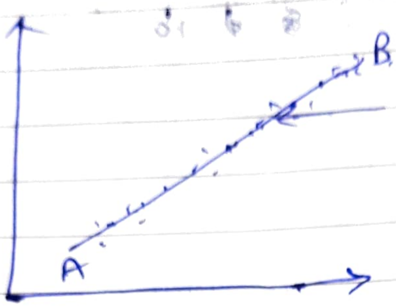


$x_{inc} = x_{k+1}$

$y_{inc} = \begin{cases} y_k \\ y_{k+1} \end{cases}$

* Bresenham's Line Drawing Algo.

- It determines the point of an n-dimensional raster that should be selected in order to form a close approximation to a straight line between two points.



It will take close approximation of pixel.

→ $A(x_1, y_1)$ $B(x_2, y_2)$

- It is efficient because it only take integer addⁿ, sub, & multiplication.

- Operations performed rapidly

• **Algorithm**

1) cal start & end co-ordinates.
(x_1, y_1) & (x_2, y_2)

2) cal Δx & Δy
$dx = x_2 - x_1$
$dy = y_2 - y_1$

3) cal decision parameter (It is used to find ~~extra~~ Extract point to draw line)

$$\boxed{P_K = 2\Delta y - \Delta x}$$

4) current point (x_k, y_k)
next point (x_{k+1}, y_{k+1})
find next point depending on value of decision parameter P_k.

case ①

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

case ②

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

5) ~~Repeat~~ Repeat step ④ until end point is reached.

Advantages

- It involve integer arithmetic, very simple.
- It avoids the generation of duplicate pts
- It can be implemented using H/w.
- It is faster than DDA

Page No.:

Date:

Example

1) $(x_1, y_1) = (9, 18)$
$(x_2, y_2) = (14, 22)$

$\rightarrow \Delta x = x_2 - x_1 \rightarrow 14 - 9 = 5$
$\Delta y = y_2 - y_1 \rightarrow 22 - 18 \quad 4$

cal decision parameter

$$P_k = 2\Delta y - \Delta x$$
$$= 2 \times 4 - 5$$
$$\boxed{P_k = 3}$$

$\therefore P_k > 0 \rightarrow$ case 2

$$P_{k+1} = P_k + 2\Delta y - 2\Delta x$$
$$= P_k + 2(\Delta y - \Delta x)$$
$$= 3 + 2(4 - 5)$$
$$= 3 + 2(-1)$$
$$= 3 - 2$$
$$= 1$$

$x_{k+1} = x_k + 1 = 9 + 1 = 10$
$y_{k+1} = y_k + 1 = 18 + 1 = 19$

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
$\cancel{3}$	$\cancel{1}$	$\cancel{15}$	$\cancel{23}$

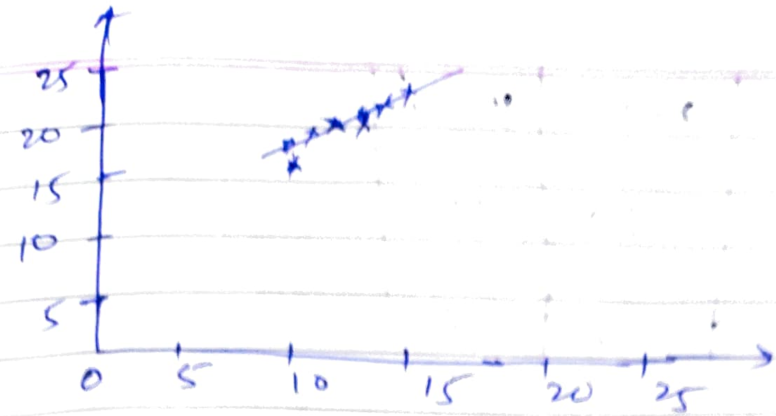
+2

$P_k < 0 \rightarrow$

$P_k < 0$

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

$= -1 + 2(4)$
$= -1 + 8 = 7$



2) Explain Bresenham line drawing Algo. ~~with~~ and identify the pixel positions of A (10,10) & B(18,16).

→ A (10,10)
B (18,16)

$\Delta x = x_2 - x_1$ 18 − 10 = 8

$\Delta y = y_2 - y_1$ 16 − 10 6

Decision parameter

$$P_K = 2\Delta y - \Delta x = 2 \times 6 - 8 = \underline{\underline{4}}$$

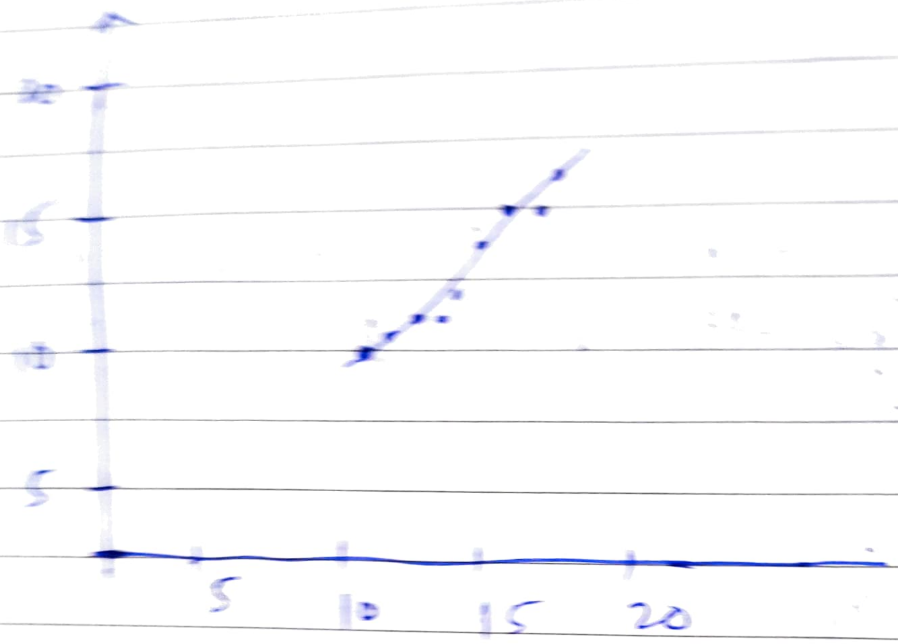
$P_K > 0$ → case ②

$P_{K+1} = P_K + 2(\Delta y - \Delta x)$
$= 4 + 2(6-8)$
$= 4 - 4 = 0$

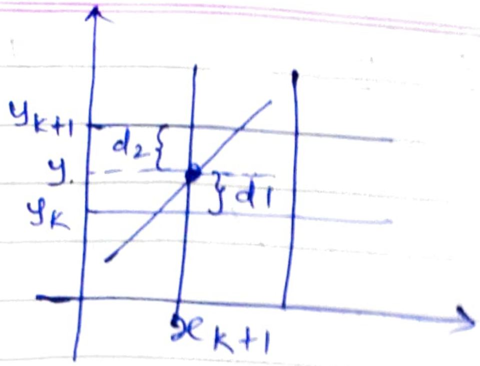
$x_{K+1} = x_K + 1$
$y_{K+1} = y_K + 1$

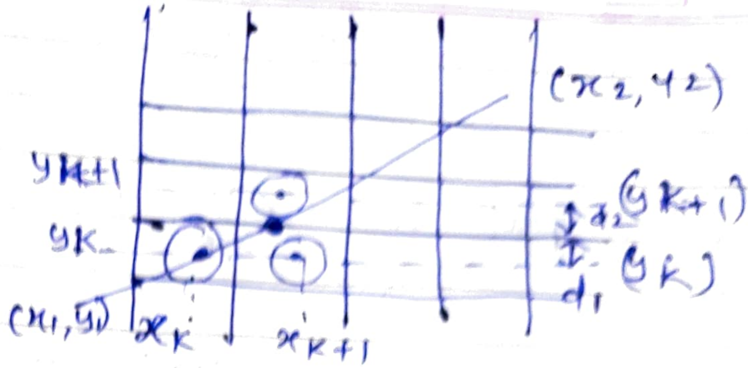
	P_K	P_{K+1}	x_{K+1}	y_{K+1}
			10	10
	4	0	11	11
$P_K \geq 0$	0	−4	12	12
$P_K < 0$	−4	8	13	12
$P_K > 0$	8	4	14	13
	4	0	15	14

p_k	p_{k+1}	x_{k+1}	y_{k+1}
6	-4	16	15
-4	18	17	15
8	4	18	16



Derivation of Bresenham's Line Drawing Algorithm.





$y = mx + c$. At sampling position x_{k+1}, corresponding y is computed as,

$$\boxed{\begin{aligned} y &= m(x_k+1)+c \\ &= m \cdot x_k + m + c \end{aligned}}$$

$$\begin{aligned} d_1 &= y - y_k \\ &= (m \cdot x_k + m + c) - y_k \end{aligned}$$

$$\begin{aligned} d_2 &= y_{k+1} - y \\ &= y_{k+1} - (m \cdot x_k + m + c) \end{aligned}$$

(for next scan line $y_{k+1} = y_k + 1$)

$$\begin{aligned} d_1 - d_2 &= (m \cdot x_k + m + c - y_k) - (y_{k+1} - m \cdot x_k - m - c) \\ &= m \cdot x_k + m + c - y_k - y_k - 1 + m \cdot x_k + m + c \\ &= \boxed{2m \cdot x_k + 2m - 2y_k + 2c - 1} \end{aligned}$$

To simplify the computation, multiply it by Δx,

Decision parameter at step k, P_k

$$\begin{aligned} P_k &= \Delta x (d_1 - d_2) \\ &= \Delta x (2m \cdot x_k + 2m - \underline{2y_k + 2c - 1}) \end{aligned}$$ constan

Derivative

$d_1 = y - y_k$

$= [m(x_k+1) + c] - y_k$

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

$= y_{k+1} - m(x_k+1) - c$

if $d_1 - d_2 < 0$ $\quad y_{next} = y_k$

$d_1 - d_2 > 0$ $\quad y_{k+1}$

$d_1 - d_2 = [m(x_k+1) + c - y_k] - [y_{k+1} - m(x_k+1) - c]$

$= 2m(x_k+1) + c - y_k - y_k - 1 + c$

$= 2m(x_k+1) - 2y_k + 2c - 1$

Here $m = \dfrac{\Delta y}{\Delta x}$

Again a float value so remove float by multiplying Δx on both side.

$\Delta x(d_1 - d_2) = \Delta x\left[\dfrac{2\Delta y}{\Delta x}(x_k+1) - 2y_k + 2c - 1\right]$

$\Delta x(d_1 - d_2) = 2\Delta y(x_k+1) - 2\Delta x y_k + 2\Delta x c - \Delta x$

$= \underbrace{2\Delta y x_k - 2\Delta x y_k}_{} + \underbrace{2\Delta y + 2\Delta x c - \Delta x}_{\text{constant}}$ --- (A)

$\Delta x(d_1 - d_2) = 2\Delta y x_k - 2\Delta x y_k$

$\Delta x(d_1 - d_2) = P_K$ — decision p

$\therefore P_K = 2\Delta y x_k - 2\Delta x y_k$

Calculate next decision parameter

$$P_{next} = 2\Delta y\, x_{next} - 2\Delta x\, y_{next}$$

$$P_{next} - P_K = [2\Delta y\, x_{next} - 2\Delta x\, y_{next}] - [2\Delta y\, x_K - 2\Delta x\, y_K]$$

$$= 2\Delta y\, x_{next} - 2\Delta x\, y_{next} - 2\Delta y\, x_K + 2\Delta x\, y_K$$

$$= 2\Delta y\,(x_{next} - x_K) - 2\Delta x\,(y_{next} - y_K)$$

Here $x_{next} = x_{K+1}$

But y_{next} has $\begin{cases} y_K \\ y_{K+1} \end{cases}$

So take decision

$$P_{next} - P_K < 0$$

then $y_{next} = y_K$, $x_{next} = x_{K+1}$

$\therefore$ $P_{next} = P_K + 2\Delta y\,(x_{K+1} - x_K) - 2\Delta x\,(y_K - y_K)$

$\therefore$ $\boxed{P_{next} = P_K + 2\Delta y}$

if $P_{next} - P_K >= 0$ then
$x_{next} = x_{K+1}$ & $y_{next} = y_{K+1}$

Page No:
Date:

$P_{next} - P_K >= 0$ then

$x_{next} = x_{K+1}$ & $y_{next} = y_{K+1}$

$P_{next} = P_K + 2\Delta y\,(x_{K+1} - x_K) - 2\Delta x\,(y_{K+1} - y_K)$

$P_{next} = P_K + 2\Delta y - 2\Delta x$

now put $P_{next} = P_{K+1}$

when $P < 0$ $\rightarrow$ $\therefore$ $\boxed{P_{K+1} = P_K + 2\Delta y}$ $P_{K+1} - P_K < 0$

when $P \geq 0$ $\rightarrow$ $\boxed{P_{K+1} = P_K + 2\Delta y - 2\Delta x}$ $P_{K+1} - P_K > 0$

Now cal initial value of decision parameter P_0 use eqn (A)

$P_K = 2\Delta y x_K - 2\Delta x y_K + 2\Delta x c - \Delta x$

put $K=1$

$P_1 = 2\Delta y x_1 - 2\Delta x y_1 + 2\Delta x c - \Delta x$

$c = y_1 - \frac{\Delta y}{\Delta x} x_1$

$P_1 = 2\Delta y x_1 - 2\Delta x y_1 + 2\Delta y + 2\Delta x \left[y_1 - \frac{\Delta y}{\Delta x} x_1\right]$

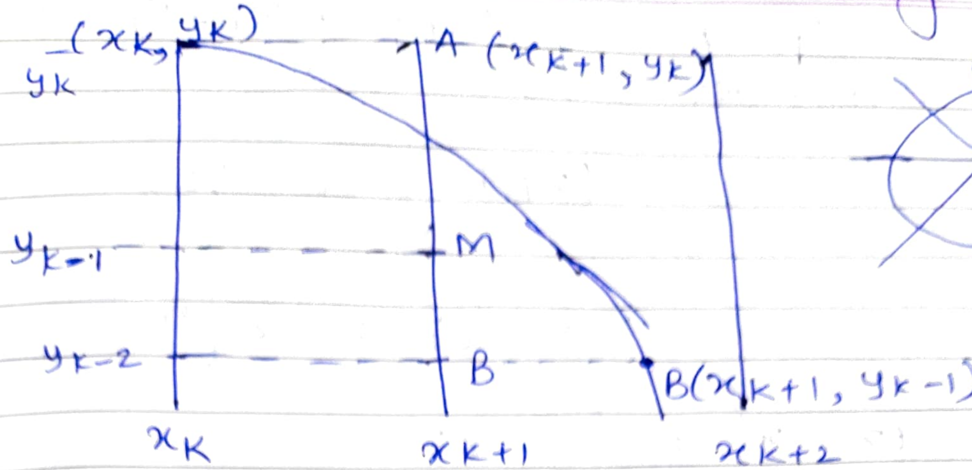
$= \cancel{2\Delta y x_1} - \cancel{2\Delta x y_1} + 2\Delta y + \cancel{2\Delta x y_1} - \cancel{2\Delta y x_1} - \Delta x$

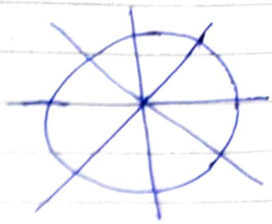
$\boxed{P_1 = 2\Delta y - \Delta x}$ — Initial value of decision parameter.

Q1. A (1, 1)
B (8, 5)

Q2 (4, 4)
(12, 9)

Mid-Point Circle Drawing Algorithm



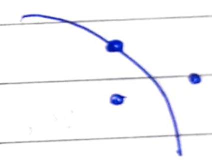


$\rightarrow \quad x^2 + y^2 = r^2$

$\rightarrow \quad x^2 + y^2 - r^2 = 0$

$(x_m, y_m) \rightarrow$ are midpoint co-ordinates

$(x_m)^2 + (y_m)^2 - r^2$

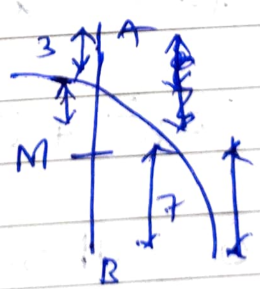


0 : point lies on circle

A → < 0 : point lies inside

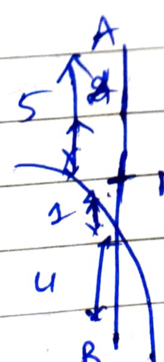
B → > 0 : point lies outside

Find minimum distance from ~~M~~ of point A & B from midpoint to select for drawing circle.



select Shortest distance.

M inside.



If m outside

Midpoint coordinates = $\left[\frac{(x_k+1 + x_k+1)}{2}, \frac{(y_k + y_{k}-1)}{2} \right]$

$= (x_k+1, y_k - 1/2)$

Put above into circle eq^n.

$x^2 + y^2 - r^2$

$$\boxed{d_k = (x_k+1)^2 + (y_k - 1/2)^2 - r^2} \quad -(1)$$

dk is decision parameter.

$$\boxed{\underline{d_{k+1}}\ \underline{d_{inc}} = (\underbrace{x_{k+1}}_{next} + 1)^2 + (\underbrace{y_{k+1}}_{next} - 1/2)^2 - r^2} \quad -(2)$$

$$\boxed{d_{k+1} - d_k}$$

x_{k+1} will same in next step

Substitute x_k+1 in eq^n (2) in place of x_{k+1}

$d_{k+1} = d_k + 2x_k + 3 + (y_k+1)^2 - y_{k+1} - y_k^2 + y_k$

If $d_k < 0$ then y coordinate only update

$y_{k+1} = y_k$

Substitute y_k in place of y_{k+1}

$$\boxed{d_{k+1} = d_k + 2x_k + 3}$$

If $d_k > 0$ then $y_{k+1} = y_k - 1$

$$\boxed{d_{k+1} = d_k + 2x_k + 2y_k + 5}$$

Now find Initial decision parameter

initial co-ordinates (x_k, y_k) substitute this with $(0, r)$ i.e. $x_k = 0$ & $y_k = r$

Now substitute $(0, r)$ in eq^n 1

$$d_0 = (0+1)^2 + (r - 1/2)^2 - r^2$$

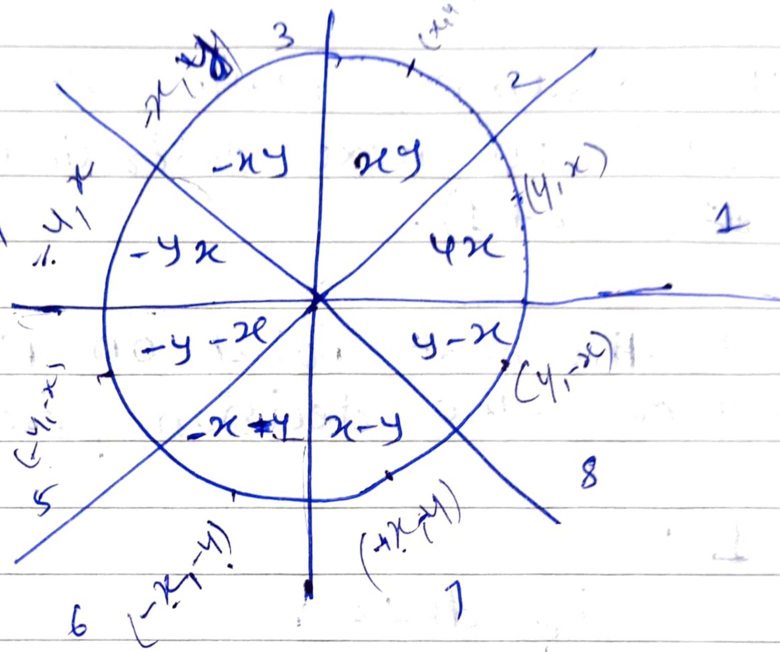
$$\boxed{d_0 = \frac{5}{4} - r} \Rightarrow \boxed{1 - r}$$

$\frac{5}{4} \cong 1$ $\because r$ is an integer

so $\boxed{d = 1 - r}$

initial value (0, r)

circle divide into 8 octants on 2D plan



Algorithm

1. Consider a center coordinates $(x1, y1)$ as

$x1 = 0;$
$y1 = r;$

2. Cal the starting decision parameter d_1

$d1 = 1 - r$

3. Let us assume starting co-ordinates as (x_k, y_k) so next coordinates are (x_{k+1}, y_{k+1}).

Find the next point on first octant based on the decision parameter (d_k)

4. Case 1 :

if $d_k < 0$
then

$x_{k+1} = x_k + 1$
$y_{k+1} = y_k$
$d_{k+1} = \cancel{d_k + 2(x_{k+1}) + 1}$
$\cancel{d_k + 2x_k + 1}$ $\boxed{d_k + 2x_{k+1} + 1}$

case 2 :

If $d_k >= 0$
then

$x_{k+1} = x_k + 1$
$y_{k+1} = y_k - 1$
$d_{k+1} = \cancel{d_k - 2(x_{k+1} + 2(x_{k+1})) + 1}$
$= \cancel{d_k + 2x + 2y}$ $\boxed{d_k - 2y_{k+1} + 2(}$

5. If the center co-ordinate pt $(x1, y1)$ is not at the origin $(0, 0)$ then finding the point as –

for x co-ordinate $= x_c + x_1$

for y co-ordinate $= y_c + y_1$

{ x_c & y_c contains current value of x & y }

6. Repeat step 4 & 5 till we get $x >= y$.

Two cases →

- $d_k < 0$ →
  $$x_{k+1} = x_{k+1}$$
  $$y_{k+1} = y_k$$
  $$d_{k+1} = d_k + 2(x_{k+1}) + 1$$

- $d_k > 0$ →
  $$x_{k+1} = x_k + 1$$
  $$y_{k+1} = y_k - 1$$
  $$d_{k+1} = d_k - 2(y_{k+1}) + 2(x_{k+1}) + 1$$

$\boxed{+1}$

Example

① Draw a circle using mid-pt circle at origin with radius $\cancel{}$ 15.

→ $(x, y) = (0, 0)$ & radius $= 15$

$x = 0$
$y = 0$
$r = 15$

$d_k = 1 - r$
$= 1 - 15$
$= -14$

$d_k < 0$. $d_{k+1} = d_k + 2x_{k+1} + 1$
$x_{k+1} = x_k + 1$
$y_{k+1} = y_k$

$d_k = \cancel{}$

As d_k is less than 0, $x_n = 1$, $y_n = 15$

$d_k = -14 + 2 \times 1 + 1 = \underline{-11}$ again less than
$\therefore x_n = 2$, $y_n = 15$

$d_k = -11 + 2 \times 2 + 1 = \underline{-6}$ again < 0
$x_n = 3$, $y_n = 15$

$d_k = -6 + 2 \times 3 + 1 = \underline{1}$ $d_k > 0$, case ②
$x_n = 4$, $y_n = 4 - 1$
$= 14$

$d_k = \cancel{1 + 2 \times 4 + 2 \times 14}$
$= \cancel{1 + 8 + 28}$
$\cancel{9}$

Page No.:
Date:

$d_k > 0$

$x_{k+1} = 4 \quad y_{k+1} = 14$

$d_{k+1} = 1 - 2 \times 14 + 2 \times 4 + 1$

$= 1 - 28 + 9$

$= -18$

$x_{k+1} = (x_{k+1}) + 1$

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

$d_{k+1} = d_k - 2y_{k+1} + 2x_{k+1} + 1$

$d_k < 0$

$x_{k+1} = 5, \quad y_{k+1} = 14$

$d_{k+1} = d_k + 2x_{k+1} + 1$

$= -18 + 2 \times 5 + 1$

$= -18 + 11 = -7$

$d_{k+1} = -7$

$d_{k+1} < 0$

$x_{k+1} = 6 \quad y_{k+1} = 14$

$d_{k+1} = -7 + 2 \times 6 + 1$

$= -7 + 13$

$= 6$

$d_{k+1} > 0$

$x_{k+1} = 7 \quad y_{k+1} = 13$

$d_{k+1} = 6 - 2 \times 13 + 2 \times 7 + 1$

$= 6 - 26 + 15$

$= -5$

$d_{k+1} < 0$

$x_{k+1} = 8 \quad y_{k+1} = 13$

$d_{k+1} = -5 + 2 \times 8 + 1$

$= -5 + 17$

$= 12$

$d_k > 0$

$x_{k+1} = 9 \quad y_{k+1} = 12$

$d_{k+1} = 12 - 2 \times 12 + 2 \times 9 + 1$

$= 12 - 24 + 19$

$= 12 - 5$

$= 7$

$d_k > 0$

$x_{k+1} = 10 \quad y_{k+1} = 11$

$- 2 \times 11 + 2 \times 10 + 1$

$= 7 \cancel{*} 22 + 2\cancel{1} \cancel{P1}$

$= 7 - 1$

$= 6$

$d_k > 0$

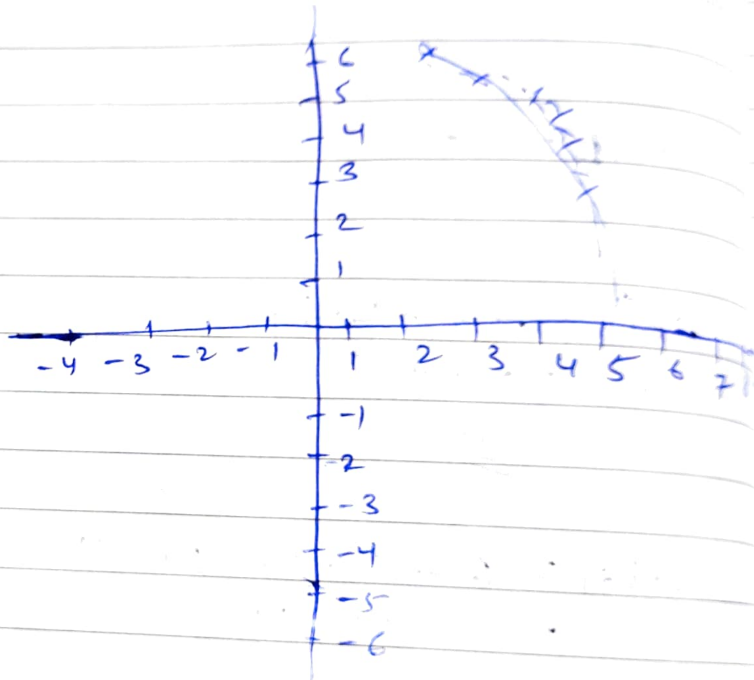
$x_{k+1} = 11 \quad y_{k+1} = 10$

Here we stop because

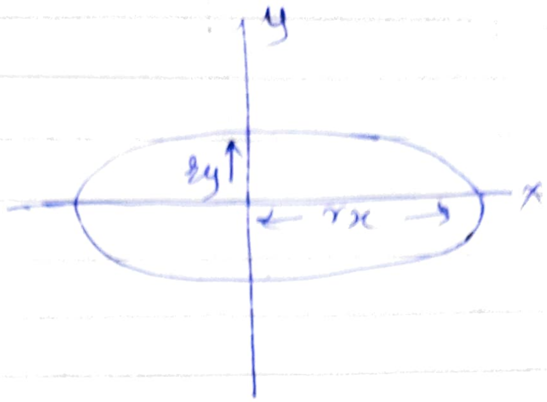
$x > y$

points are -

(1, 15)
(2, 15)
(3, 15)
(4, 14)
(5, 14)
(6, 14)
(7, 13)
(8, 13)
(9, 12)
(10, 11)
(11, 10)



Midpoint Ellipse Algorithm



ellipse is an elongated circle

eqn of ellipse is centered at (0,0)

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

were $a = rx$ & $b = ry$

Major Axis $= 2a = 2rx$ (longer axis)
Minor Axis $= 2b = 2ry$ (short axis)

Semi major Axis $= a = rx$
semi minor Axis $= b = ry$

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

$$x^2b^2 + a^2y^2 = a^2b^2$$
$$x^2b^2 + y^2a^2 - a^2b^2 = 0$$

put $a = rx$ & $b = ry$

$$\therefore x^2ry^2 + y^2rx^2 - rx^2ry^2 = 0 \quad - (1)$$

Page No.:
Date:

If we put any point in eq (1)

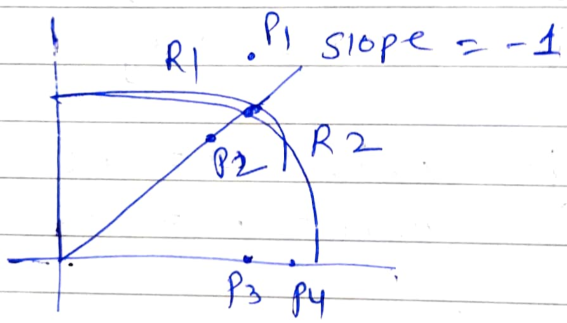
Case 1
= 0.
on boundary

Case 2
< 0
inside boundary

Case 3
> 0
outside boundary

In circle, 8 way symmetry
In ellips 4 way symmetry

In circle, we need to plot one octant of any quadrant but in ellips we need to plot 2 octants i.e. one complete quadrant.



In Region R1 we can have next point at x increment, y to take decision.

(x_{k+1}, y_k) or (x_{k+1}, y_{k-1})

But in Region R2 we are considering 2 points on major axis P3 & P4.

So here it depends on ~~the~~ whether to increment x or decrement x

In this Algo, we plot 1st quadrant which has 2 regions & have different slope in region R_1 & R_2 so we have to plot both regions by using different formulas.

Quadrant 1 → Region 1

i) start point : $(0, r_y)$
ii) slope of ellips < -1
iii) Take unit steps in positive x direction till boundary between 2 regions is reached. → $x = x+1$

Quadrant ~~1~~ → Region 2

slope of ellips > -1

Take unit steps in negative y direction ~~##~~ till the end of quadrant.

On the boundary between 2 regions, slope of the ellips is -1.

Here y is decremented by 1 so $y-1$ & will decide whether to increment x or not

Example - Midpoint Ellipse Algo

- Read radius r_x & r_y

- Initialize starting point & region 1 as
  $x = 0, \; y = r_y \rightarrow (0, r_y)$

- Calculate $\boxed{P1_0 = r_y^2 - r_x^2 \cdot r_y + \frac{1}{4} r_x^2}$

- Calculate $dx = 2 r_y^2 x, \; dy = 2 r_x^2 y$

- Repeat while $(dx < dy)$ — Region 1

* plot (x, y)
* if $(P1 < 0)$
{
$x = x + 1$
update $dx = 2 r_y^2 \cdot x \approx$ old $(dx + 2r_y^2)$

$P_1 = P_1 + 2 r_y^2 x + r_y^2$
}

* else
{
$x = x + 1$
$y = y - 1$
update $dx \rightarrow 2 r_y^2 x$ {old $dx + 2r_y^2 x$}
update $dy \rightarrow 2 r_x^2 y$ {old $dy + 2r_x^2 y$}
$P_1 = P_1 + dx - dy + r_y^2$
}

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

- Find $\boxed{P2_0 = r_y^2 (x + 1/2)^2 + r_x^2 (y-1)^2 - r_x^2 r_y^2}$

- Repeat till $(y > 0)$

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

{

$x = x$

$y = y - 1$

update $dy = 2 r_x^2 y$

$P_2 = P_2 - dy + r_x^2$

}

else

{

$x = x + 1$

$y = y - 1$

$dy = \cancel{dy} \to 2 r_x^2$

$dx = \cancel{dx} + 2 r_y^2$

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

}

Q. 1. $r_x = 8$, $r_y = 6$ given.

Iterat^n	(x_K, y_K)	P_1 or P_2	(x_{k+1}, y_{k+1})	dx: $2x_{k+1} r_y^2$	dy: $2y_{k+1} r_x^2$
0.	$(0, r_y)$ $(0, 6)$	$P_{10} = r_y^2 - r_x^2 \cdot r_y + 1/4 \cdot r_x^2$ $= 36 - 64 \times 6 + 1/4 \cdot 64$ $= 36 - 384 + 16$ $= -332$ $\therefore P_1 < 0$	$(1, 6)$ x inc, y as it is. Now $dx < dy$	$2 \times 1 + 36$ $= 72$	$2 \times 6 + 64$ $= 768$
1.	$(1, 6)$	$P_1 = P_1 + \underbrace{2 r_y^2 x}_{dx} + r_y^2$ $= -332 + 2 \times 36 + 36$			

			dx	dy
	$= -224$	(2, 6)	$= 72 \times 2$ $= 144$	768
				dx < dy
2. (2, 6)	$P_1 = -224 + 144 + 36$ $= -44$	(3, 6)	$6(36)$ $= 216$	768
				dx < dy
3. (3, 6)	$P_1 = -44 + 216 + 36$ $= 208$	(4, 5)	$8(36)$ $= 288$	10×64 $= 640$
			dx < dy	
4. (4, 5)	$P_1 = 208 + 288 + 36 - 640$ $= -108$	(5, 5) (6, 4)	$10(36)$ $= 360$	$= 640$
			dx < dy	
5. (5, 5)	$P_1 = -108 + 360 + 36$ $= 288$	(6, 4)	$= 12(36)$ $= 432$	$= 8(64)$ $= 512$
			dx < dy	
6. (6, 4)	$P_1 = 288 + 432 - 512 + 36$ $= 244$	(7, 3)	$= 14(36)$ $= 504$	$= 6(64)$ $= 384$

Now dx > dy

P2

plot Region 2

0. (7, 3)	$P_2 = r_y^2(x + 1/2)^2 + r_x^2(y-1)^2 - r_x^2 \cdot r_y^2$ $= 36(7 + 1/2)^2 + 64(3-1)^2 - 36 \times 64$ $= 1\cancel{0}2\cancel{1} + 256 - 2304$ $= -23$	(8, 2)	$16(36)$ $= 576$	4×64 $= 256$
1. (8, 2)	$P_2 = -23 + 576 - 256 + 64$			

	$P_2 = 361$	$(8,1)$	$16(36) = 576$	$2(64) = 128$
2. $(8,1)$	$P_2 = 361 - 128 + 64 = 297$	$(8,0)$ Now stop when $y=0$	~~16(36)~~ —	—
3. $(8,0)$	—	—	—	—