Page :
Date: / /

Assignment 2 :-

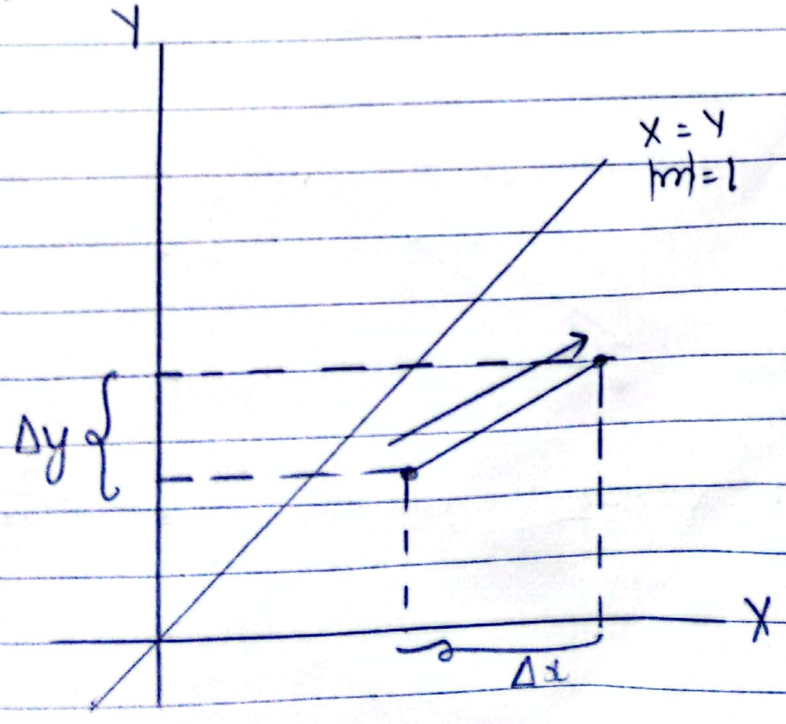
13) Write DDA Algorithm pseudo code for all eight cases.

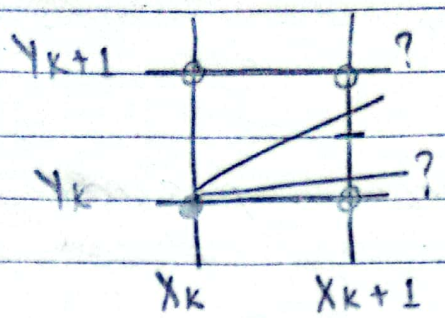
A) For lines with positive slope :-

1) for lines with slope ≤ 1 (moving from left to right)

Perform unit increment in 'X' direction, $\Delta X = 1$ as $\Delta X > \Delta Y$ i.e., $X_{k+1} = X_k + 1$ and compute successive 'y' values as
$y_{k+1} = y_k + m$

The 'y' value computed must be rounded off to the nearest whole number.



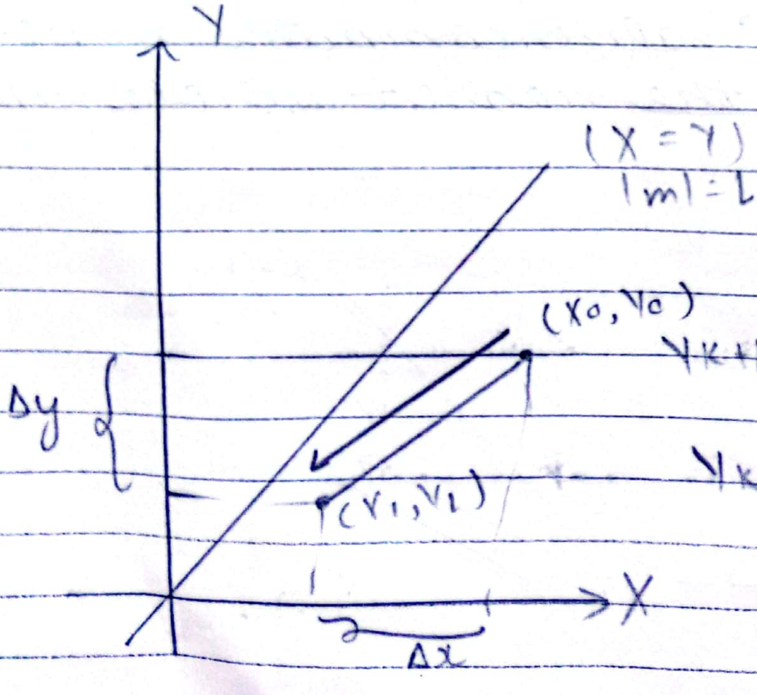


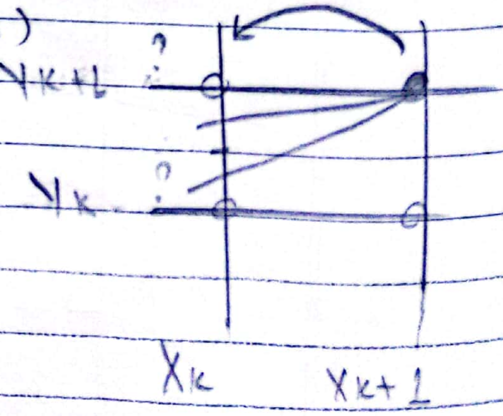
2) for lines with slope ≤ 1 (moving from right to left) :-

Perform unit decrement in 'X' direction, $\Delta X = -1$ as $\Delta X > \Delta Y$ i.e; $X_{k+1} = X_k - 1$ and compute successive 'Y' value as

$$Y_{k+1} = Y_k - m$$

The 'Y' value computed must be rounded off to the nearest whole number.



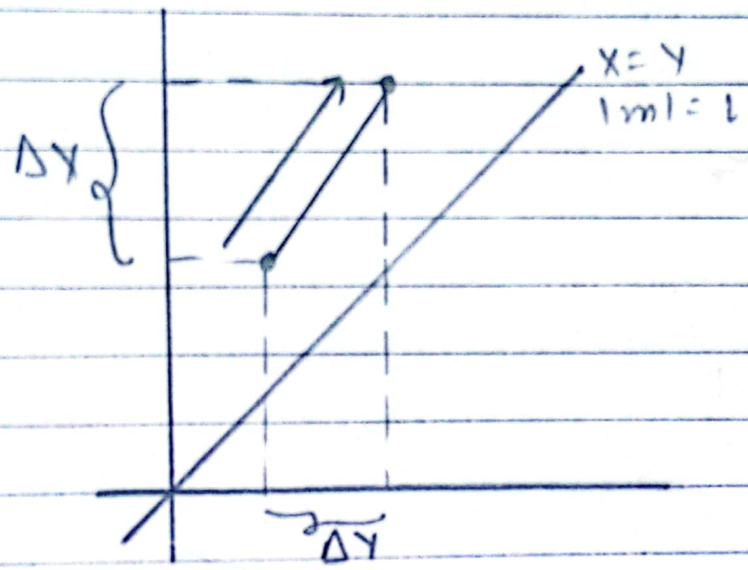


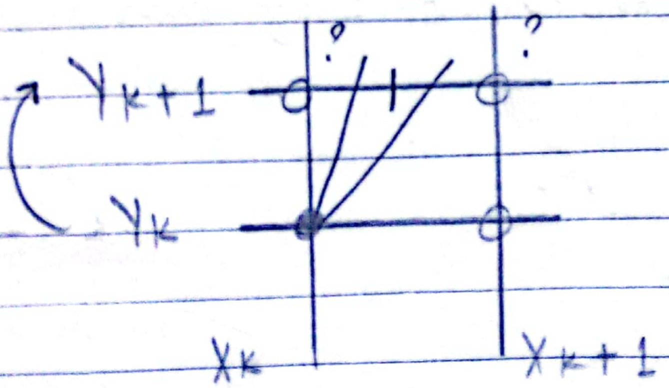
3) For lines with slope > 1 moving from left to right

Perform unit increment in 'Y' direction $\Delta Y = 1$ as $\Delta Y > \Delta X$, i.e; $Y_{k+1} = Y_k + 1$ and compute successive 'X' value as

$$X_{k+1} = X_k + \frac{1}{m}$$

The 'X' value computed must be round off to the nearest whole number.



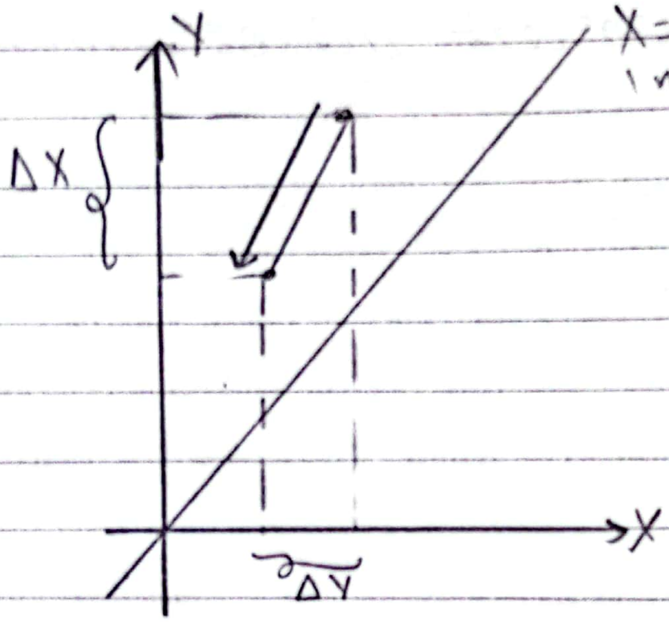


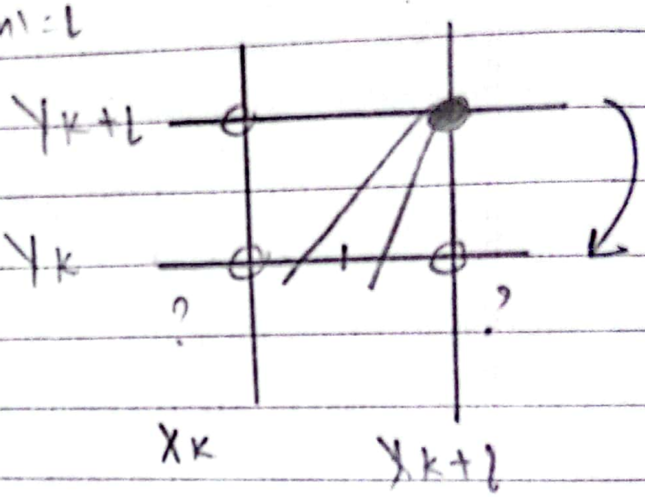
4) for lines with slope > 1 moving f
left:-

Perform unit decrement in 'Y' direction, $\Delta Y = -1$ as $\Delta Y > \Delta X$, i.e; $Y_{k+1} = Y_k - 1$ and compute successive 'X' value as

$$X_{k+1} = X_k - \frac{1}{m}$$

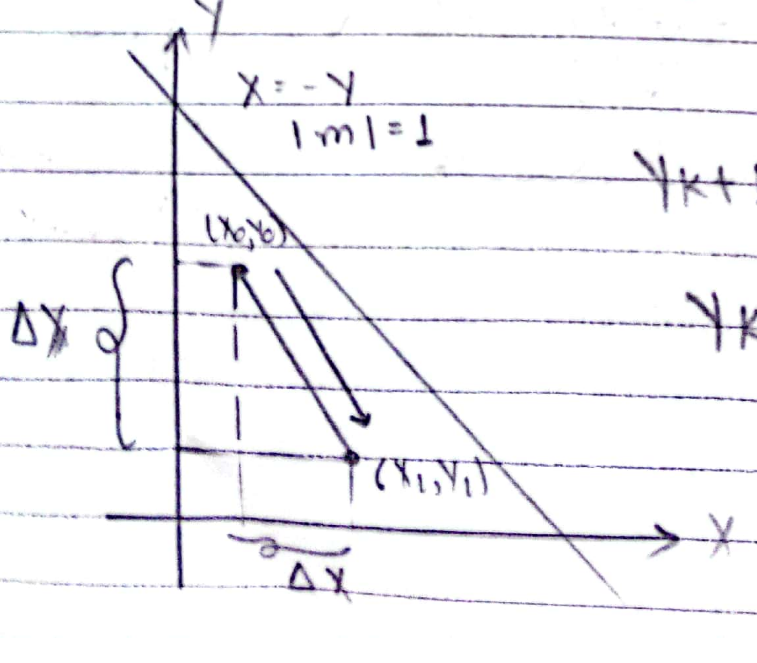
The 'X' value computed must be rounded off to the nearest whole number.

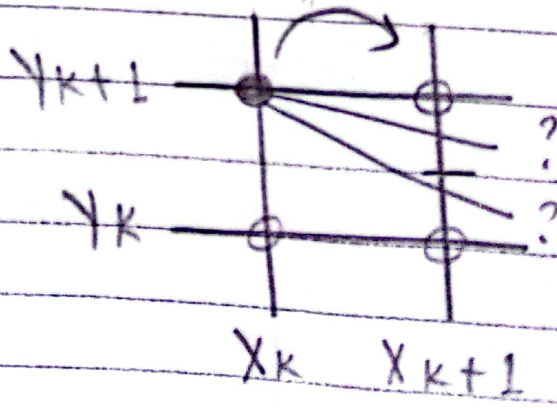




B) For lines with negative slope:-

5) for lines with $|m| \leq 1$ (from left to right)





Perform unit increment in 'X' direction, $\Delta X = 1$ as $\Delta X > \Delta Y$, i.e., $X_{k+1} = X_k + 1$ and compute successive 'Y' value as

$$Y_{k+1} = Y_k + m$$

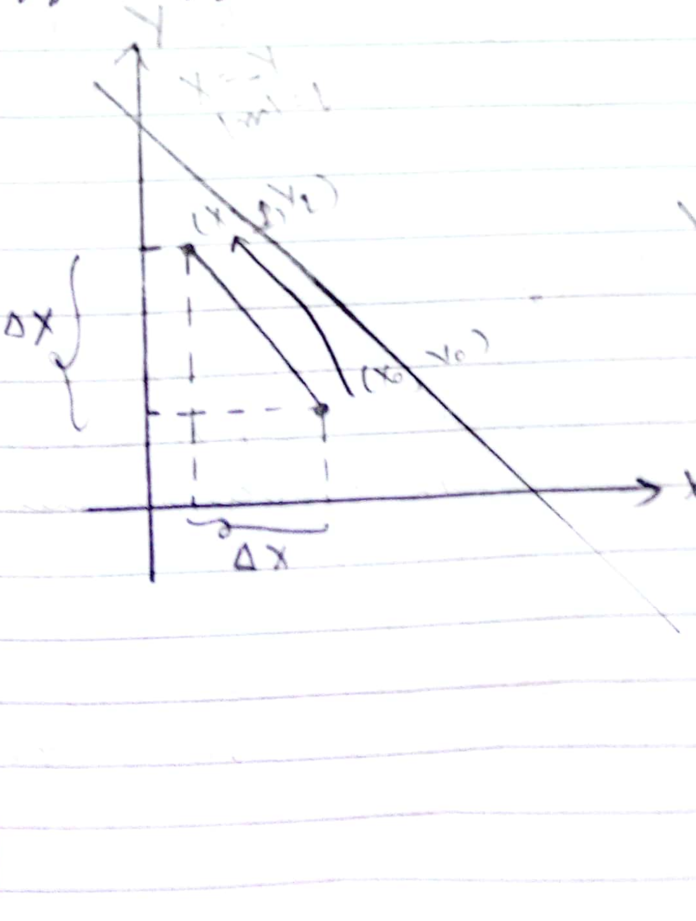
The 'Y' value computed must be rounded ~~off~~ to the nearest whole number.

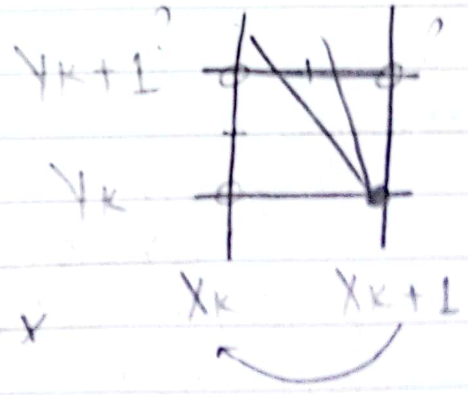
6) for lines with $|m| \le 1$ (from right to left):-

Perform unit decrement in 'X' direction, $\Delta X = -1$ as $\Delta X > \Delta Y$, i.e., $X_{k+1} = X_k - 1$ and compute successive 'Y' value as

$$Y_{k+1} = Y_k - m$$

The 'Y' value computed must be rounded ~~off~~ off to the nearest whole number.





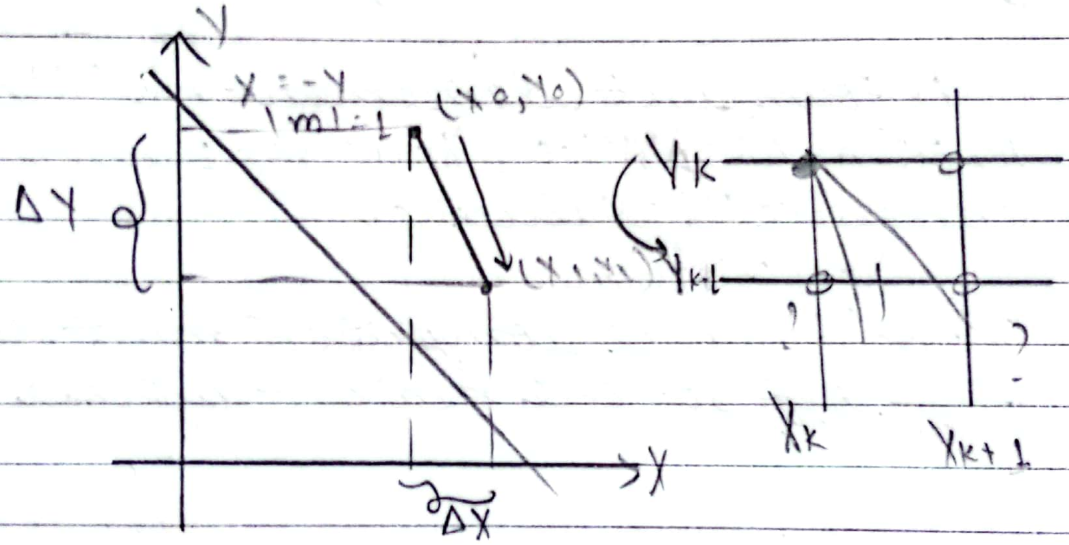
Page :
Date: / /

7) for lines with $|m| > 1$ (from left to right):-

Perform unit decrement in 'Y' direction, $\Delta Y = -1$ as $\Delta Y > \Delta X$ i.e.; $Y_{k+1} = Y_k - 1$ and compute successive 'X' value as

$$X_{k+1} = X_k - \frac{1}{m}$$

The 'X' value computed must be rounded off to the nearest whole number.

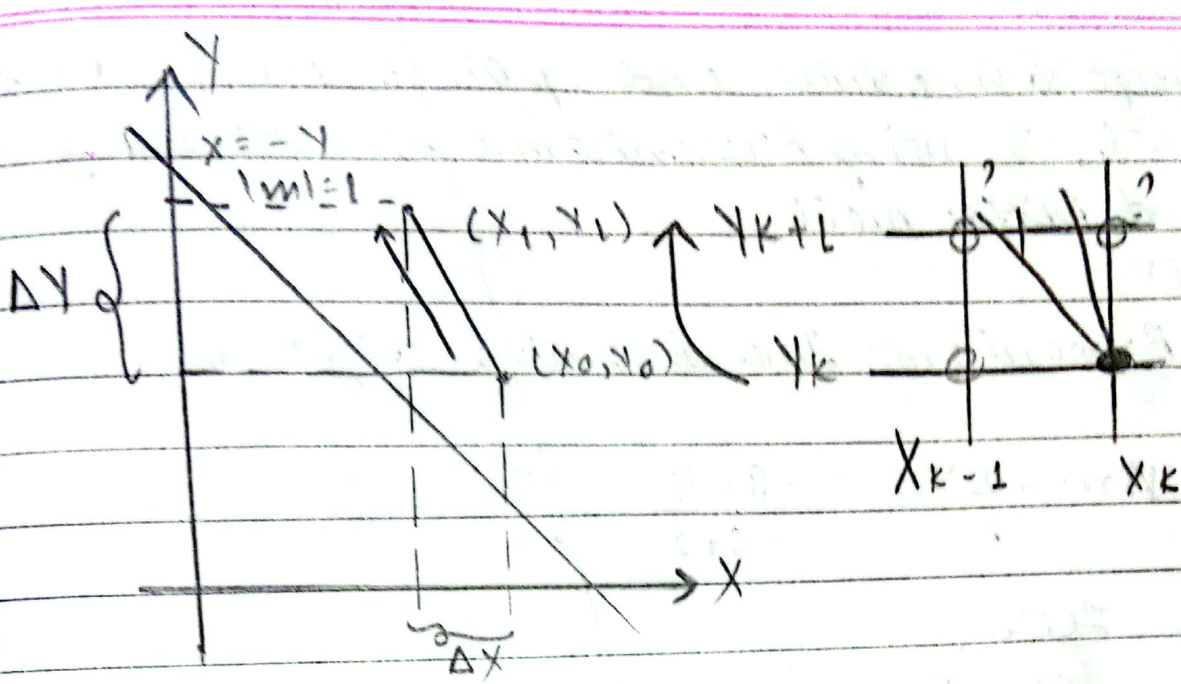


8) for lines with $|m| > 1$ (from right to left):-

Perform unit increment in 'Y' direction, $\Delta Y = 1$ as $\Delta Y > \Delta X$, i.e., $Y_{k+1} = Y_k + 1$ and compute ~~succ~~ successive 'X' value as

$$X_{k+1} = X_k + \frac{1}{m}$$

The 'X' value computed must be rounded off to the nearest whole number.


Y
X = - Y
|m| = 1
(X1, Y1)
ΔY
(X0, Y0)
X
ΔX
Yk+1
Yk
?
?
Xk-1
Xk

5) Digitize a line with end points A(-2,-4) and B(-6,-9) using Bresenham's line drawing Algorithm and DDA as well.

Soln,

(1) Bresenham's line drawing Algorithm:

$$\Delta m = \frac{\Delta y}{\Delta x} = \frac{-9+4}{-6+2} = \frac{-5}{-4} = \frac{5}{4}$$

Here,

$\Delta y = 5$

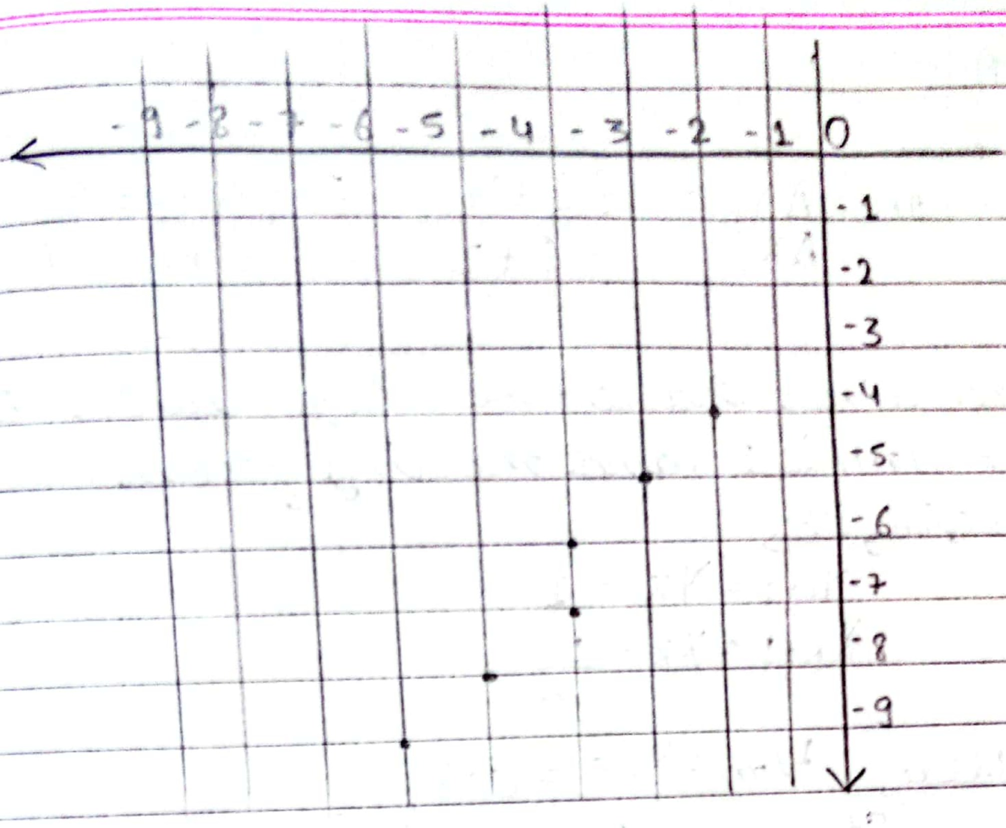
$\Delta x = 4$

Starting pixel = A(-2, -4)

$P_0 = 2\Delta X - \Delta Y = 2(4) - 5 = 8 - 5 = 3$ //

Now,

K	P_K	(X_{K+1}, Y_{K+1})
0	$P_0 = 3$	(-3, -5)
1	$P_1 = 3 + 2(4) - 2(5) = 1$	(-4, -6)
2	$P_2 = 1 + 2(4) - 2(5) = -1$	(-4, -7)
3	$P_3 = -1 + 2(4) = 7$	(-5, -8)
4	$P_4 = 7 + 2(4) - 2(5) = 5$	(-6, -9)



(ii) DDA:-

$$m = \frac{\Delta y}{\Delta x} = \frac{-9+4}{-6+2} = \frac{-5}{-4} = \frac{5}{4}$$

The line has a +ve slope with $m \geq 1$, ltoR, with $\Delta Y > \Delta X$, i.e;

$$Y_{k+1} = Y_k + 1$$

$$X_{k+1} = X_k + \frac{1}{m}$$

X_{k+1}	Round off value of X_{k+1}	Y_{k+1}
		-5
$X_{k+1} = \cancel{4.8}\ -3.2$	$\cancel{5}$ -3	-6
$X_{k+1} = \cancel{5.6}\ -2.4$	6	-7
$X_{k+1} = 6.4$	6	-8
$X_{k+1} = 7.2$	7	-9
$X_{k+1} =$		

(ii) DDA :-

$$m = \frac{\Delta Y}{\Delta X} = \frac{-9+4}{-6+2} = \frac{-5}{-4} = \frac{5}{4}$$

The line have +ve slope with $\Delta Y > \Delta X$, i.e. $|m| \geq 1$ and moving from right to left, i.e.,

$$Y_{k+1} = Y_k - 1$$

$$X_{k+1} = X_k - \frac{1}{m}$$

Here, $1/m = 4/5 = 0.8$

Starting pixel = $A(-2, -4)$

$X_{k+1} = X_k - 1/m$	Round off values of X_{k+1}	Y_{k+1}
-2.8	-3	-5
-3.6	-4	-6
-4.4	-4	-7
-5.2	-5	-8
-6	-6	-9

(ii) DDA:-

$$m = \frac{\Delta Y}{\Delta X} = \frac{-9+4}{-6+2} = \frac{-5}{-4} = \frac{5}{4}$$

The lines have +ve slope with $\Delta Y > \Delta X$, i.e. $|m| \geq 1$ and moving from right to left, i.e.,

$$Y_{k+1} = Y_k - 1$$

$$X_{k+1} = X_k - \frac{1}{m}$$

Here, $1/m = 4/5 = 0.8$

Starting pixel = $A(-2,-4)$

$X_{k+1} = X_k - 1/m$	Round off value of X_{k+1}	Y_{k+1}
-2.8	-3	-5
-3.6	-4	-6
-4.4	-4	-7
-5.2	-5	-8
-6	-6	-9

Date: / /

4. Derive necessary equations for drawing a circle using midpoint circle Algorithm? How can you use this ~~algori~~ algorithm to draw a circle if the starting ~~p~~ point is $(-r, 0)$ moving in clockwise direction?

Soln:

The equation of a circle is given by ~~X^2+y~~ $X^2+Y^2-r^2$. To apply midpoint method, assuming that the circle is centered at origin, the first pixel to plot $(0, r)$ where 'r' is the radius of pixel and ~~moving n~~ moving in clockwise direction, we define a circle function

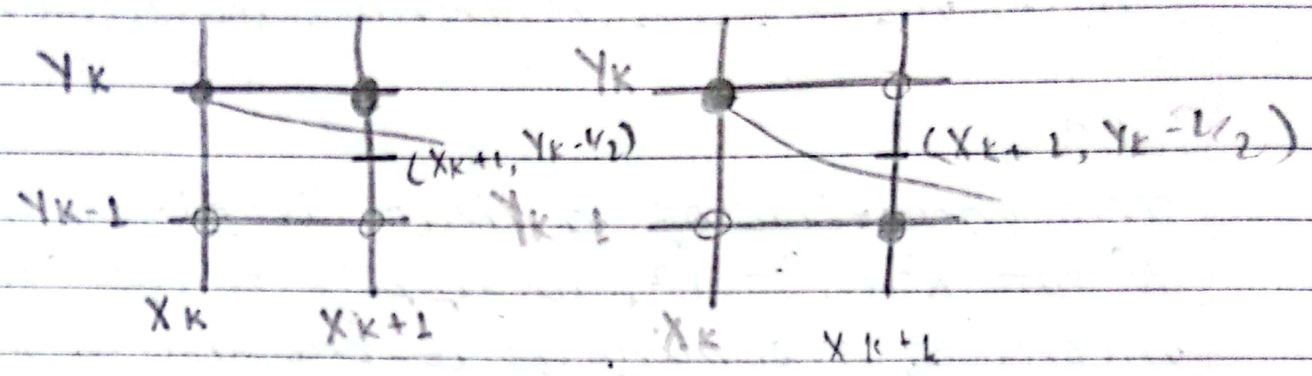
$$f_{circle}(X,Y) = X^2+Y^2-r^2$$

now,

$f_{circle}(X,Y)$ < 0 if $P(X,Y)$ is inside circle boundary
$= 0$ if $P(X,Y)$ is on circle boundary
> 0 ~~s~~ if $P(X,Y)$ is outside the circle boundary.

The circle function $f_{circle}(X,Y)$ serves as the decision parameter. Select next pixel along circle path according to the sign of the circle function evaluated at the mid-point between these two candidate pixels. Start at $(0, Y)$, take unit steps in X direction, ~~X_K~~ i.e., $X_{K+1} = X_K + 1$

~~Assumi~~ Assuming pixel at position (X_k, Y_k) has been plotted in previous step, we determine next position (X_{k+1}, Y_{k+1}) as either (X_k+1, Y_k) or (X_k+1, Y_k+1) along circle path by evaluating the



decision parameter which is the circle function ~~status~~ s evaluated ~~betwee~~ between these two candidate pixels.

$$P_k = f_{circle}(X_k+1, Y_k-1/2)$$
$$= (X_k+1)^2 + (Y_k-1/2)^2 - r^2 \quad \text{— (i)}$$

At next sampling position, the decision parameter is evaluated as

$$P_{k+1} = f_{circle}(X_{k+1}+1, Y_{k+1}-1/2)$$
$$\text{or, } P_{k+1} = [(X_{k+1})+1]^2 + (Y_{k+1}-1/2)^2 - r^2$$
$$\text{or, } P_{k+1} = (X_{k+1})^2 + 2X_{k+1} + 1 - (Y_{k+1}-1/2)^2 - r^2 \quad \text{— (ii)}$$

Subtracting (i) from (ii), we get,

$$P_{k+1} = P_k + 2X_{k+1} + (Y_{k+1}-1/2)^2 - (Y_k-1/2)^2 + 1 \quad \text{— (iii)}$$

where,

Y_{k+1} is either Y_k or Y_{k-1} depending on the ~~A sign of~~ ~~sign of P_k~~ sign of P_k.

Page:

Date: / /

Case I :-

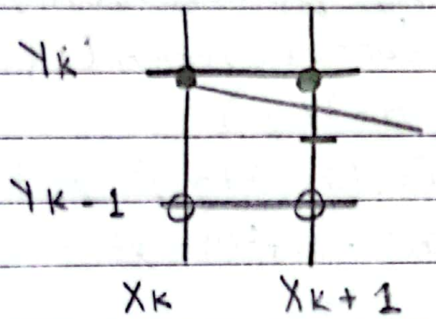
If $P_k < 0$, then mid-point is inside circle boundary and pixel on scanline 'Y_k' is closer to circle boundary. So,

$Y_{k+1} = Y_k$ and from eq (iii)

$$P_{k+1} = P_k + 2(X_k + 1) + 1 \quad \text{—— (iv)}$$

where,

$$X_{k+1} = X_k + 1$$



Case II :-

If $P_k \geq 0$, then, midpoint is on or outside the circle boundary and pixel on scanline '$Y_k - 1$' is ~~closer~~ ~~closes~~ close to circle boundary.

So, we select pixel on scanline $Y_k - 1$, i.e;

$$Y_{k+1} = Y_k - 1$$

from eqn (iii),

$$P_{k+1} = P_k + 2(X_k + 1) + (Y_k - 1 - 1/2)^2 - (Y_k - 1/2)^2 + 1$$

$$= P_k + 2(X_k + 1) + (Y_k - 3/2)^2 - (Y_k - 1/2)^2 + 1$$

$$= P_k + 2(X_k + 1) + [\cancel{Y_k^2} \cancel{- 3/2 Y_k - 3/2 Y_k} - 3Y_k + 9/4 - \cancel{Y_k^2} + Y_k - 1/4] + 1$$

$$= P_k + 2(X_k + 1) + 1 - 2Y_k + 2$$

$$= P_k + 2(X_k + 1) - 2(Y_k - 1) + 1$$

~~$= P_k + 2(X_k + 1) - 2(Y_k$~~ $P_k + 2(X_k + 1) - 2Y_{k+1} + 1$

Or, $P_{k+1} = P_k + 2X_{k+1} - 2Y_{k+1} + 1$ —— (v) where,

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

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

Initial decision parameter P_0 is obtained by evaluating circle function at the starting position $(x_0, y_0) = (0, r)$

The ~~next~~ next pixel to plot is either $(1, r)$ or $(1, r-1)$

~~Starting pos~~ Starting position $(x_0, y_0) = (0, r)$

The next pixel to plot is either $(1, r)$ or $(1, r-1)$

So, mid-point is $(1, r - 1/2)$

$$\text{or, } P_0 = f_{circle} = (1)^2 + (r - 1/2)^2 - r^2$$

$$P_0 = 1 + \cancel{r^2} - r + 1/4 - \cancel{r^2}$$

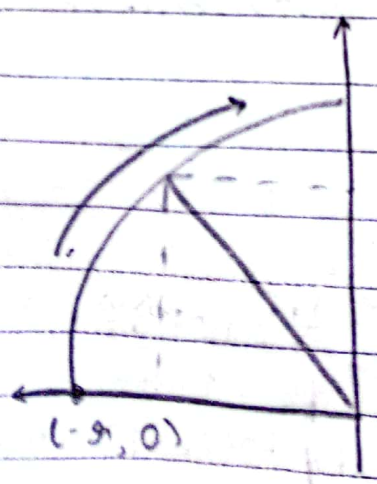
$$P_0 = 5/4 - r$$

$$\boxed{\therefore P_0 \approxeq 1 - r}$$

Again,

Starting point $(-r, 0)$

Moving in anticlockwise direction,



The eq^n of a circle is given by $X^2 + Y^2 - r^2$.
To apply midpoint method, assuming that the circle is centered at origin, the first pixel to plot $(-r, 0)$, where, 'r' is the radius of the pixel moving in clockwise direction, we define a ~~cir~~ circle function,

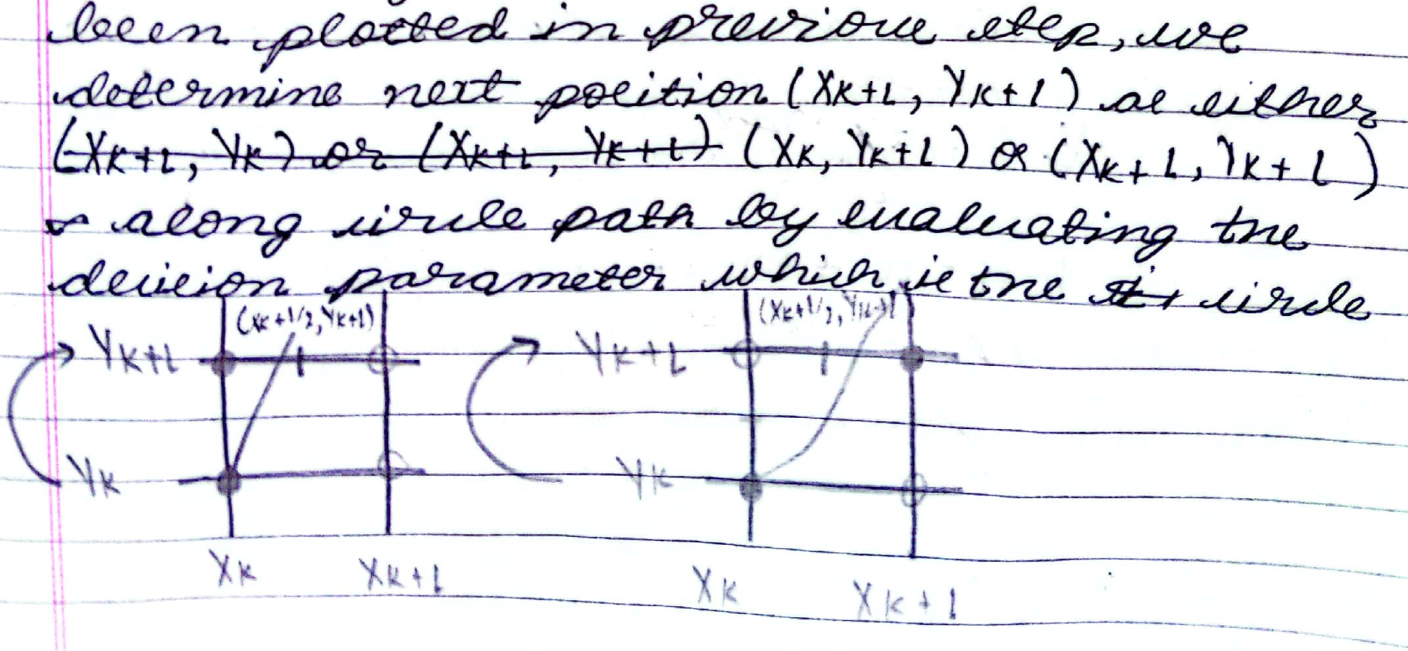
$$f_{circle}(X, Y) = X^2 + Y^2 - r^2$$

Now,

$f_{circle}(X, Y) < 0$ if $P(X, Y)$ is inside circle boundary
$= 0$ if $P(X, Y)$ is on the circle boundary
> 0 if $P(X, Y)$ is outside the circle boundary.

This circle function $f_{circle}(X, Y)$ serves as the decision parameter. Select next pixel along circle path ~~acc~~ according to the sign of the circle function evaluated at the mid-point between these two candidate pixels. Start at ~~(0, r)~~ $(-X, 0)$, take unit steps in Y ~~X~~ direction (Sample in Y direction, i.e., $Y_{k+1} = Y_k + 1$).

Assuming pixel at position (X_k, Y_k) has been plotted in previous step, we determine next position (X_{k+1}, Y_{k+1}) as either ~~(X_{k+1}, Y_k) or (X_{k+1}, Y_{k+1})~~ $(X_k, Y_k + 1)$ or $(X_k + 1, Y_k + 1)$ along circle path by evaluating the decision parameter which is the ~~st~~ circle



function evaluated between these two candidate pixels.

$$P_K = f_{circle}(X_K + 1/2,\ Y_K + 1)$$
$$= (X_K + 1/2)^2 + (Y_K + 1)^2 - r^2 \quad \text{— (i)}$$

At next ~~so~~ sampling position, the decision parameter is evaluated as,

$$P_{K+1} = f_{circle}(X_{K+1} + 1/2,\ Y_{K+1} + 1)$$
$$= f(X_{K+1} + 1/2)^2 + (Y_{K+1} + 1)^2 - r^2$$
$$\cancel{= (X_{K+1})^2 + X_{K+1}}$$
$$= (X_{K+1} + 1/2)^2 + (Y_{K+1})^2 + 2Y_{K+1} + 1 - r^2 \quad \text{— (ii)}$$

Subtracting (i) from (ii), we get,

$$P_{K+1} = P_K + (X_{K+1} + 1/2)^2 - (X_K + 1/2)^2 + 2Y_{K+1} + 1 \quad \text{— (iii)}$$

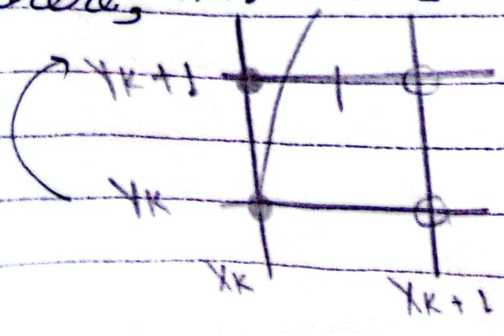
where, X_{K+1} is either X_K or X_{K+1} depending on the sign of P_K.

Case I:-

If $P_K < 0$, then, mid-point is inside circle boundary and pixel on scanline 'X_K' is closer to the circle boundary. So, ~~Xr~~ $X_{K+1} = X_K$ and from eq^n (iii).

$$P_{K+1} = P_K + 2Y_{K+1} + 1 \quad \text{— (iv)}$$

where, $Y_{K+1} = Y_K + 1$



Case II:

If $P_k \geq 0$ then, mid-point is on or outside circle boundary and pixel on scan line 'x_k' & 'x_{k+1}' 'x_k' is closer to circle boundary so we select pixel on scanline x_k+1, i.e.;

$x_{k+1} = x_k + 1$ and from eq^n (iii)

$$P_{k+1} = P_k + (x_k + 1 + 1/2)^2 - (x_k + 1/2)^2 + 2y_{k+1} + 1$$
$$= P_k + (x_k + 3/2)^2 - (x_k + 1/2)^2 + 2y_{k+1} + 1$$
$$= P_k + [\cancel{x_k^2} + 3x_k + 9/4 - \cancel{x_k^2} - x_k - 1/4] + 2y_{k+1} + 1$$
$$= P_k + [2x_k + 2] + 2y_{k+1} + 1$$
$$= P_k + 2(x_k + 1) + 2y_{k+1} + 1$$
$$= P_k + 2x_{k+1} + 2y_{k+1} + 1 \quad \text{(v)}$$

where,

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

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

Initial decision parameter P_0 is obtained by evaluating circle function at the starting position $(x_0, y_0) = (-r, 0)$.

The next pixel to plot is either $(-r+1, 1)$ or $(-r, 1)$. So, the mid-point is $(-r + 1/2, 1)$.

or, $P_0 = f_{circle} = (-r + 1/2)^2 + (1)^2 - r^2$

$P_0 = \cancel{r^2} - r + 1/4 + 1 - \cancel{r^2}$

$P_0 = 5/4 - r$

$\boxed{\therefore P_0 \approx 1 - r}$ //

Date: / /

9. Digitize a circle described by an equation $(x+5)^2+(y-3)^2=25$

Soln,

$$(x+5)^2+(y-3)^2=25$$

or, $(x-h)^2+(y-k)^2=(r)^2$

$\therefore (h,k)=(-5,3)$

$r=5$

Starting pixel $=(0,5)$

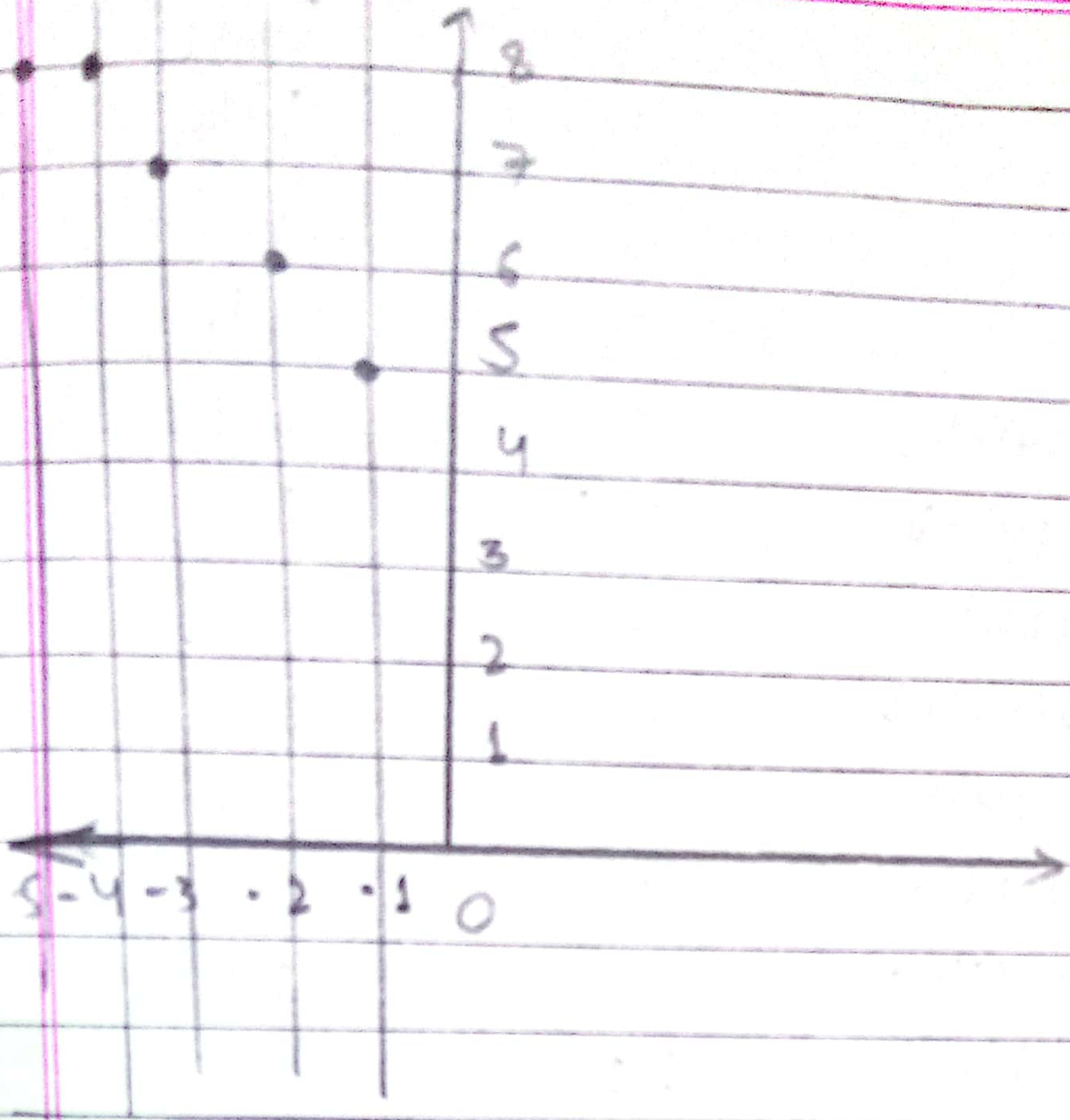
$P_0=1-r=1-5=-4$

K	P_{K+1}	X_{K+1}	Y_{K+1}
0	$P_0=-4$	1	5
1	$P_1=-4+2(1)+1$ $=-1$	2	5
2	$P_2=-1+2(2)+1$ $=4$	3	4
3	$P_3=4+2(3)-2(4)+1$ $=3$	4	3
4			

Here,

$(x_c, y_c)=(-5,3)$

$X=X+X_c$	$Y=Y+Y_c$
$1-5=-4$	$5+3=8$
$2-5=-3$	$5+3=8$
$3-5=-2$	$4+3=7$
$4-5=-1$	$3+3=6$



8
7
6
5
4
3
2
1
-5 -4 -3 -2 -1 0

Date: / /

11) Derive the criteria necessary ~~to~~ identify the ~~condit~~ condition when we leave region one and enter region two in case of Mid-point ~~elli~~ ellipse algorithm.

Soln.

The eqn of ellipse is given by $\frac{X^2}{r_X^2} + \frac{Y^2}{r_Y^2} = 1$

$\therefore f_{ellipse}(X, Y) = X^2 r_Y^2 + Y^2 r_X^2 - r_X^2 r_Y^2$

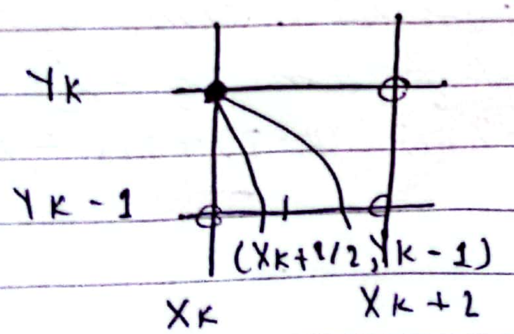
Now,

$f_{ellipse}(X, Y)$ < 0 if (X, Y) is inside ellipse ~~boundry~~ boundary

= 0 if (X, Y) is on ellipse boundary

> 0 if (X, Y) is outside ellipse boundry.

Here, we sample in 'Y' direction, the mid-point is taken between horizontal pixels at each step now. Assuming pixel at (X_K, Y_K) has been plotted, next pixel ~~to~~ to plot is (X_K+1, Y_K-1) or (X_K, Y_K-1)



So, mid-point coordinate position is $(X_K + 1/2, Y_K - 1)$ and $f_{ellipse}(X_K + 1/2, Y_K - 1)$.

or, $P2_K = r_Y^2 (X_K + 1/2)^2 + r_X^2 (Y_K - 1)^2 - r_X^2 r_Y^2$ —— (iv)

At next sampling step, the next ~~pixel~~ pixel to plot will be either $(X_{k+1}, Y_{k+1}-1)$ or $(X_{k+1}+1, Y_{k+1}-1)$,

thus,

~~f ellipse~~ f ellipse$(X_{k+1}+1/2, Y_{k+1}-1)$,

~~thus,~~

$$P_{2k+1} = r_y^2(X_{k+1}+1/2)^2 + r_x^2(Y_{k+1}-1)^2 - r_x^2 r_y^2$$

$$= r_y^2(X_{k+1}+1/2)^2 + r_x^2[(Y_{k+1}-1)]^2 - r_x^2 r_y^2 \quad \text{—— (v)}$$

Subtracting (iv) and (v), we get,

$$P_{2k+1} = P_{2k} - 2r_x^2(Y_k-1) + r_x^2 + r_y^2[(X_{k+1}+1/2)^2 - (X_k+1/2)^2] \quad \text{—— (vi)}$$

where,

X_{k+1} will either be X_k or X_{k+1} depending on the sign of P_{2k}.

Case I:-

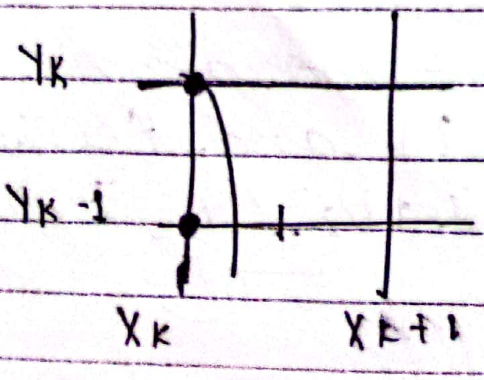
if $P_{2k} > 0$ then mid point is outside the boundary of ellipse some select ~~pix~~ pixel at (X_k, Y_k-1) and from eq^n (vi)

$$P_{2k+1} = P_{2k} - 2r_x^2(Y_k-1) + r_x^2 = P_{2k} - 2r_x^2 Y_{k+1} + r_x^2 \quad \text{—— (a)}$$

where,

$Y_{k+1} = Y_k - 1$

$X_{k+1} = X_k + 1$



Page :
Date: / /

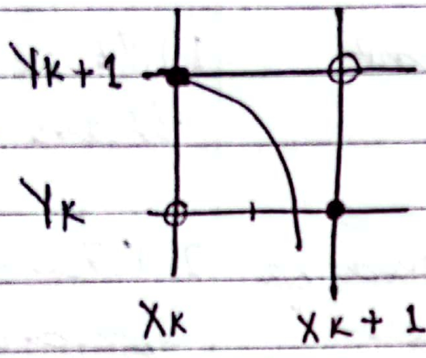
CaseII :-

if $P2_k \leq 0$ then, the mid-point is on or inside ellipse boundary. So, we select pixel at (X_k+1, Y_k-1) and from eqn (vi)

$$P2_{k+1} = P2_k - 2r_x^2 Y_{k+1} + 2r_y^2 X_{k+1} + r_x^2 \quad \text{— (b)}$$

where, $Y_{k+1} = Y_k - 1$

$X_{k+1} = X_k + 1$



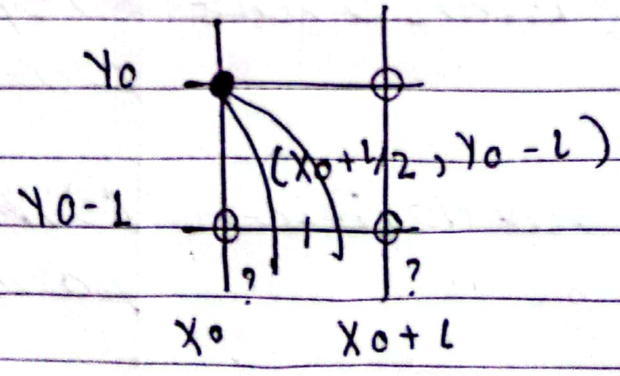
~~For reg~~

For region 2, the initial point (X_0, Y_0) is taken as the last position selected in region 1,

thus the initial decision ~~p~~ parameter in region 2 is

$$P2_0 = f_{ellipse}(X_0 + 1/2, Y_0 - 1)$$

$$= r_y^2 (X_0 + 1/2)^2 + (Y_0 - 1)^2 r_x^2 - r_x^2 r_y^2$$



12) How is DDA different from BLA?

The differences between DDA and BLA are :-

Digital Differential Analyzer	Bresenham's Line Drawing Algorithm
1) It is less efficient	1) It is more efficient than DDA algorithm.
2) The calculation speed of DDA algorithm is less than that of Bresenham's Line Drawing Algorithm.	2) The ~~th~~ calculation speed of BLA is faster than DDA algorithm.
3) It is costlier than BLA.	3) It is cheaper than DDA.
4) It has less precision or accuracy.	4) It has more precision or accuracy.
5) Optimization is not provided.	5) Optimization is provided
6) There is no decision parameter.	6) There is decision parameter.
7) Complexity of calculation is more complex	7) Complexity of calculation is simple.

3. Derive Bresenham's Line Drawing Algorithm for $|m|>1$. How can a line with end points $A(x_1,y_1)$, $B(x_2,y_2)$ and slope less than 1 can be drawn if starting ~~p~~ point is taken as $B(x_2,y_2)$ using BLA algorithm.

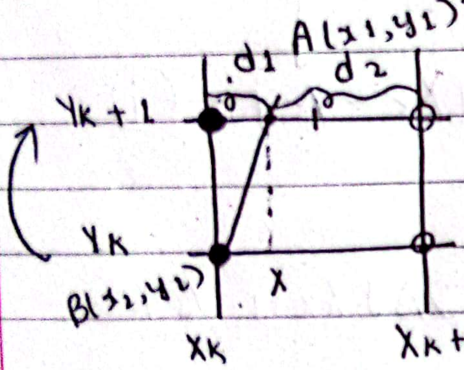
Soln,

The equation of line is given by $Y = mX + C$ — (i)

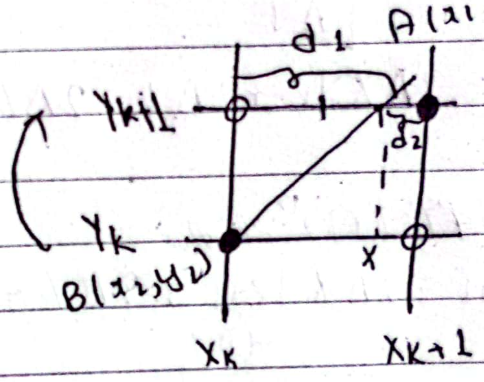
where, m is the slope.

For lines $|m| > 1$, $\Delta Y > \Delta X$, i.e., there is unit increment in the direction of 'Y'.

The decision parameters are d_1 and d_2.



Case I



Case II

The starting pixel is (X_K, Y_K). The next pixel might be (X_K, Y_K+1) or (X_K+1, Y_K+1).

$$\cancel{Y = M(X)} \quad Y = MX + C$$

$$\text{or, } X = \frac{Y - C}{M}$$

$$X = \left(\frac{Y_K + 1 - C}{M}\right) \quad \text{— (a)}$$

The distance of lower pixel from the ideal location is

$$d_1 = X - X_K = \left(\frac{Y_K + 1 - C}{M}\right) - X_K \quad \text{— (b)}$$

The distance of the upper pixel from the ideal location is :-

$$d_2 = x_k + 1 - x$$

$$d_2 = x_k + 1 - \left(\frac{y_k + 1 - c}{m} \right) \quad \text{— (c)}$$

Subtracting d_2 from d_1, we get,

$$d_1 - d_2 = \frac{2}{m} \left(y_k + 1 - c \right) - 2x_k - 1$$

At k^{th} step,

$$P_k = \Delta y (d_1 - d_2)$$

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

$$P_k = 2\Delta x y_k + 2\Delta x - 2\Delta x c - 2\Delta y x_k - \Delta y \quad \text{— (i)}$$

At $(k+1)^{th}$ step,

$$P_{k+1} = 2\Delta x y_{k+1} + 2\Delta x - 2\Delta x c - 2\Delta y x_{k+1} - \Delta y \quad \text{— (ii)}$$

Subtracting (i) from (ii), we get,

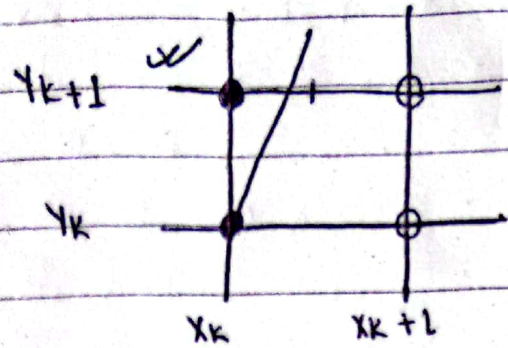
$$P_{k+1} = P_k + 2\Delta x (y_{k+1} - y_k) - 2\Delta y (x_{k+1} - x_k) \quad \text{— (iii)}$$

Case I :-

For $P_k \le 0$, then, the pixel on scan line 'x_k' is closer to the line path,

$x_{k+1} = x_k$ & eqn (iii) becomes,

$$P_{k+1} = P_k + 2\Delta y$$

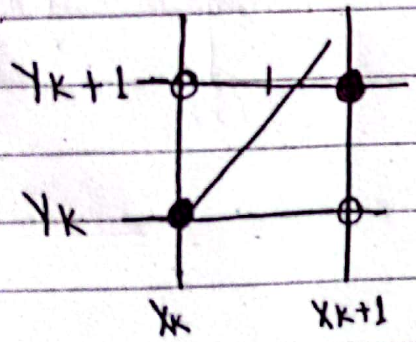


Case II :-

if $P_K \geq 0$, then, pixel on scan line 'X_{K+1}' is close to line path,

$X_{K+1} = X_K$ and eq^n (iii) becomes,

$$P_{K+1} = P_K + 2\Delta X - 2\Delta Y$$



Initial decision parameter, $P_0 = ?$

We have,

$$d_1 - d_2 = \frac{2}{m}(Y_{K+1} + c) - 2X_K - 1$$

If line passes through (X_0, Y_0), then,

$$d_1 - d_2 = \frac{2}{m}Y_0 + \frac{2}{m} + \frac{2c}{m} - 2X_0 - 1$$

$$d_1 - d_2 = \frac{2}{m} - 1 \quad \left[\frac{2}{m}Y_0 - 2X_0 + \frac{2c}{m} = 0 \right]$$

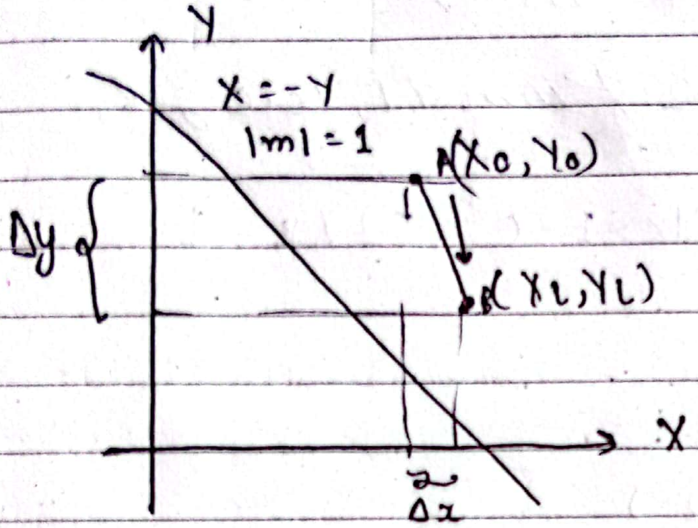
$$P_0 = \Delta Y(d1-d2)$$

$$P_0 = \Delta Y\left[\frac{2\Delta X}{\Delta Y} - 1\right]$$

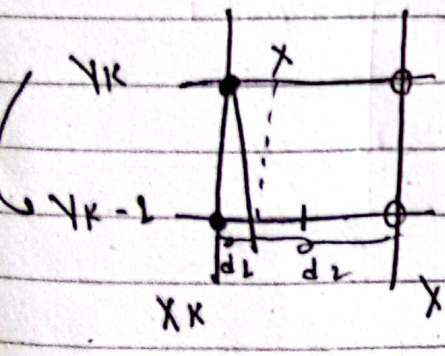
$$\therefore P_0 = 2\Delta X - \Delta Y$$ //

2) Explain the logic used to draw lines with negative slope with Bresenham's Line Drawing Algorithm.

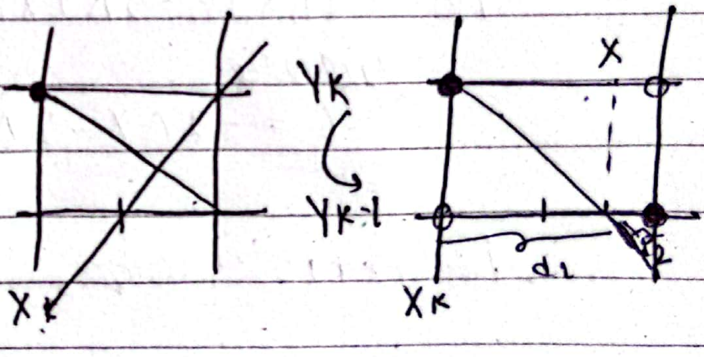
The equation of line is $Y = mX + c$ —— (1)



For lines with negative slope and lines with slope $|m| > 1$, $\Delta Y = -1$, $\Delta Y > \Delta X$, there is unit increment in the direction of 'Y'.



Case I



Case II

Let d_1 and d_2 be the decision parameters.
d_1 is the distance of lower pixel with the ideal pixel

$$d_1 = X - X_k$$

$$d_1 = \left(\frac{Y_k - 1 - c}{m}\right) - X_k$$

d_2 is the distance of the upper pixel with the ideal location,

$$d_2 = X_k + 1 - X$$

$$d_2 = X_k + 1 - \left(\frac{Y_k - 1 - C}{m}\right)$$

Subtracting d2 from d1, we get,

$$d_1 - d_2 = \frac{2}{m}\left(Y_k - 1 - C\right) - 2X_k - 1$$

At kth step,

$$P_k = \Delta Y(d_1 - d_2)$$

$$P_k = \Delta Y\left(\frac{2\Delta X}{\Delta Y}\left(Y_k - 1 - C\right) - 2X_k - 1\right)$$

$$P_k = 2\Delta X Y_k - 2\Delta X - 2\Delta X C - 2\Delta Y X_k - \Delta Y$$

$$\therefore P_k = 2\Delta X Y_k - 2\Delta Y X_k + b \quad \text{--- (i)}$$

where,

$$b = -2\Delta X - 2\Delta X C - \Delta Y$$

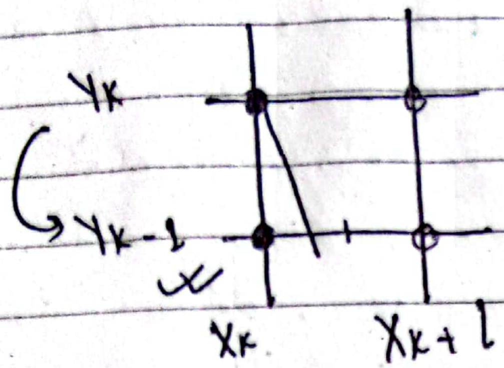
At (P_{k+1})th step,

$$P_{k+1} = 2\Delta X Y_{k+1} - 2\Delta Y X_{k+1} + b \quad \text{--- (ii)}$$

Subtracting (i) from (ii), we get,

$$P_{k+1} = P_k + 2\Delta X(Y_{k+1} - Y_k) - 2\Delta Y(X_{k+1} - X_k) \cancel{+b} \quad \text{--- (iii)}$$

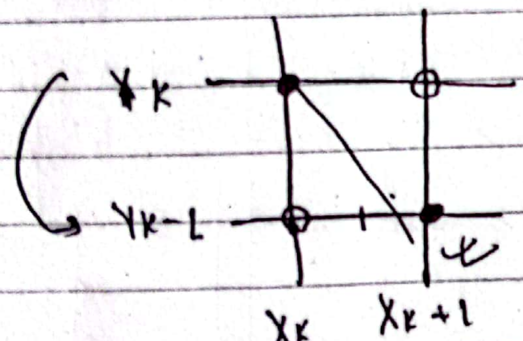
~~Case II~~ Case I :-



As $y_{k+1} = y_k - 1$, $x_{k+1} = x_k$ is selected

$\therefore P_{k+1} = P_k + 2\Delta x$

Case II :-



As $y_{k+1} = y_k - 1$, $x_{k+1} = x_k + 1$ is selected.

$\therefore P_{k+1} = P_k - 2\Delta x - 2\Delta y$

The initial decision parameter is calculated at (x_0, y_0)

$$\therefore d_1 - d_2 = \frac{2}{m}(y_0 - 1 - c) - 2x_0 - 1$$

$$= \frac{2}{m}y_0 - \frac{2}{m} - \frac{2}{m}c - 2x_0 - 1 = -\frac{2}{m} - 1$$

$$\left[\therefore \frac{2}{m}y_0 - \frac{2}{m}c - 2x_0 = 1\right]$$

$$P_0 = \Delta Y(d_1 - d_2)$$

$$P_0 = \Delta Y\left[\frac{-2}{m} - 1\right]$$

$$P_0 = \Delta Y\left[\frac{-2\Delta X}{\Delta Y} - 1\right]$$

$$\therefore P_0 = -2\Delta X - 1$$

$$\boxed{\therefore P_0 = -2\Delta X - 1}$$

1. How is the ~~de deci~~ decision parameter calculated in case of Bresenham's line Drawing Algorithm and Mid Point Circle Algorithm. What is its role?

Ans In ~~bo~~ both Bresenham's line drawing algorithm and mid-point circle algorithm, the decision parameter is used to determine the next pixel or point to be plotted in the line or circle.

1. Bresenham's Line Drawing Algorithm:-

In BLA, the decision parameter, d, is used to determine whether the next pixel should be selected along the horizontal or diagonal direction.

The algorithm uses the concept of incremental error to update the decision parameter and determine the next pixel position.

For lines with slope $|m| < 1$,

$$P_0 = d = 2\Delta Y - \Delta X$$

For lines with slope $|m| \geq 1$,

$$P_0 = d = 2\Delta X - \Delta Y$$

2. Midpoint Circle Algorithm:-

The decision parameter, i.e., d, is used to determine the next pixel or point on the circumference of the circle.

It helps in maintaining the symmetry of the circle and ensures that the approximation remains accurate.

The inital decision parameter is:-

$d = 1 - r$

6. Digitize a line with end points A(19, 9) and B(17, 11) using BLA and DDA.

BLA:-

Slope(m) $= \frac{\Delta Y}{\Delta X} = \frac{11-9}{17-19} = \frac{2}{-2} = -1$

The line is of negative slope with ~~|m| ≥ 1~~ |m| ≤
moving from ~~to left~~ ~~left to~~ right to left.

$\Delta X = -1$

$\Delta Y = 1$

Starting point = A(19, 9)

K	P_K	(X_{k+1}, Y_{k+1})
0	$P_0 = 2\Delta X - \Delta Y = -2-1 = -3$	(18, 10)
1	$P_1 = -3 \cancel{+} -2 - 2 = -6$	(17, 11)

DDA:-

Slope(m) $= \frac{\Delta Y}{\Delta X} = \frac{11-9}{17-19} = \frac{+2}{-2} = -1$

The line is of -ve slope with ~~|m| ≥ 1~~ $|m| \leq 1$
moving from ~~left to~~ right to left.

Here,

$X_{k+1} = X_k + 1$

$X_{k+1} = X_k - 1/m \quad Y_{k+1} = Y_k + m$

$1/m = -1$

X_{k+1} Round off values of X_{k+1} Y_{k+1}

X_{k+1} $Y_{k+1} = Y_k + m$ Round off values of Y_{k+1}

Starting pixel = A(15, 9)

$X_{k+1} = X_k - 1$

$Y_{k+1} = Y_k - m$

X_{k+1}	$Y_{k+1} = Y_k - m$	Round off values of Y_{k+1}
18	10	10
17	11	11

8. Digitize a circle with a radius of 14 pixels centered at $(-10,-12)$.

Soln,

Given,

Radius $(r) = 14$

$h = -10$

$k = -12$

Comparing with $(x-h)^2 + (y-k)^2 = r^2$

$(x+10)^2 + (y+12)^2 = 196$

Starting point $= (0, 14)$

$\therefore P_0 = 1 - r$

$= 1 - 14$

$= -13$ //

K	P_K	X_{K+1}	Y_{K+1}
0	$P_0 = -13$	1	14
1	$P_1 = -13 + 2 + 1 = -10$	2	14
2	$P_2 = -10 + 4 + 1 = -5$	3	14
3	$P_3 = -5 + 6 + 1 = 2$	4	13
4	$P_4 = 2 + 8 + 26 + 1 = 37$	5	12
5	$P_5 = 37 + 10 + 24 + 1 = 72$	6	11
6	$P_6 = 72 + 12 + 22 + 1 = 107$	7	10
7	$P_7 = 107 + 14 + 20 + 1 = 142$	8	9
8	$P_8 = 142 + 16 + 18 + 1 = 177$	9	8

Thus, a circle with radius of 14 pixels centered at $(-10,-12)$ is digitized.

7. ~~Derive~~ Digitize a line with end points A(11, 9) and B(29,17)

Solⁿ:

<u>Using DDA,</u>

$$m = \frac{\Delta Y}{\Delta X} = \cancel{\frac{17-9}{29-11}} \quad \frac{17-9}{29-11} = \frac{8}{18} = \frac{4}{9}$$

The lines are of +ve slope with $\Delta X > \Delta Y$, i.e; $|m| \leq 1$ ~~with~~ ~~m~~ moving from left to right.

Starting point = A(11, 9)

$X_{k+1} = X_k + 1$

$Y_{k+1} = Y_k + m$

X_{k+1}	$Y_{k+1} = Y_k + m$	Round off value of Y_{k+1}
12	9.44	9
13	9.88	10
14	10.33	10
15	10.77	11
16	11.22	11
17	11.66	12
18	12.11	12
19	12.55	13
20	13	13
21	13.44	13

22	13.88	14
23	14.33	14
24	14.77	15
25	15.22	15
26	15.66	16
27	16.11	16
28	16.55	17
29	17	17

Thus, a line with end points A(11, 9) and B(29, 17) is digitized.

Q.10 Digitize an ellipse with semi-~~mo~~ minor axis of 10 pixels and semi-major axis of 15 pixels.

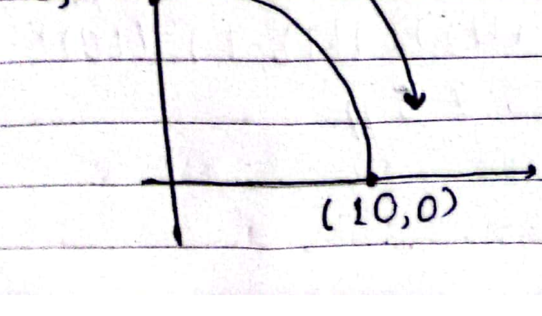
Sol^n,

Given,

$r_x = 10$

$r_y = 15$

Region 1,



Starting pixel = (0, 15)

$P_{1S} = r_x^2 + \frac{1}{4}(r_y)^2 - r_x^2 r_y$

$P_{1S} = (10)^2 + \frac{1}{4}(15)^2 - (10)^2(15)$

$\therefore P_{1S} = -1343.75$

K	P_{1K}	X_{K+1}	Y_{K+1}	$2r_y^2 X_{K+1} \geq 2r_x^2 Y_{K+1}$
0	$P_{1S} = -1343.75$	1	15	$450 \geq 3000$ (F)
1	$P_{14} = -1343.75 + 2(15)^2 + (15)^2 = \cancel{-1118.75}$ -668.75	2	15	$900 \geq 3000$ (F)
2	$P_{13} = -668.75 + 2(15)^2(2) + (15)^2 = 456.25$	3	14	$1350 \geq 2800$ (F)
3	$P_{11} = 456.25 + 2(15)^2(3) - 2(10)^2(14) + (15)^2 = -768.75$	4	14	$1800 \geq 2800$ (F)
4	$P_{11} = -768.75 + 2(15)^2(4) + (15)^2 = 1256.25$	5	13	$2250 \geq 2600$ (F)
5	$P_{20} = 1256.25 + 2(15)^2(5) - 2(10)^2(13) + (15)^2 = 1131.25$	6	12	$2700 \geq 2400$ (True)

Region 2:-

Starting point = (6, 12)

$P_{20} = r_y^2(x_0 + 1/2)^2 + (y_0 - 1)^2 r_x^2 - r_x^2 r_y^2$

$P_{20} = (15)^2(6 + 1/2)^2 + (12 - 1)^2(10)^2 - (10)^2(15)^2$

$\therefore P_{20} = -893.75$ //

k	P_{k+1}	x_{k+1}	y_{k+1}
0	$P_0 = -893.75$	7	11
1	$P_1 = \cancel{8}156.25$	7	10
2	$P_2 = \cancel{856.25} -1743.75$	8	9
3	$P_3 = 156.25$	8	8
4	$P_4 = -1343.75$	9	7
5	$P_5 = 1406.25$	9	6
6	$P_6 = 306.25$	9	5
7	$P_7 = -593.75$	10	4
8	$P_8 = 3206.25$	10	3
9	$P_9 = 2706.25$	10	2
10	$P_{10} = 2406.25$	10	1
11	$P_{11} = 2306.25$	10	0

Thus, an ellipse with ~~minor axis~~ semi-minor axis of 10 pixels and semi-major axis of 15 pixels is digitized.