Computer Graphics Assignment

2. Derive Bresenham's decision parameter to draw a line moving from left to right and having negative slope less than 1.

=

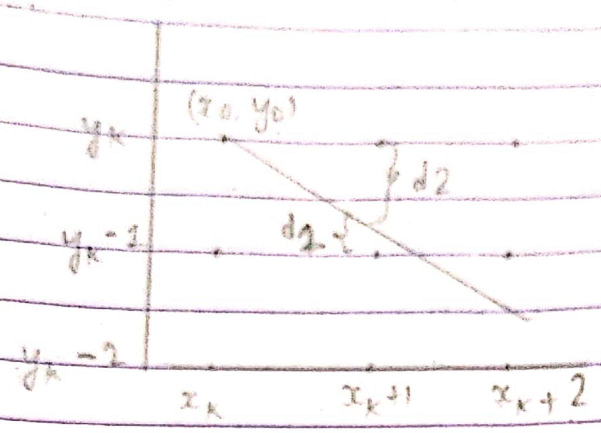


Fig :- line moving from left to right and

Let the line start from (x_0, y_0). As the line moves from left to right, the value of x co-ordinate is increased by one.

The slope of line is less than one which means the value of y co-ordinate goes on decreasing.

Assuming the pixel at (x_k, y_k) to be displayed is determined, we need to decide which pixel to plot at column $x_k + 1$. The choices we have are:

$(x_k + 1, y_k)$ and $(x_k + 1, y_k - 1)$

Now, the y co-ordinate on $(x_k + 1)$ is, $y = m(x_k + 1) + b$

while sampling in column $x_k + 1$, we calculate the separations d1 and d2 and plot the line with less separation than the mathematical line.

From figure,

$$d1 = y - (y_k - 1)$$
$$= m(x_k + 1) + b - y_k + 1 \quad \cdots (1)$$

and $d2 = y_k - y$

$= y_k - m(x_k + 1) - b \quad \cdots\cdots (II)$

Subtracting (II) from (I)

$$d1 - d2 = m(x_k+1) + b - y_k + 1 - y_k + m(x_k+1) + b$$

or, $$d1 - d2 = 2m(x_k+1) + 2b - 2y_k + 1$$

As slope $(m) = \frac{\Delta y}{\Delta x}$,

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

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

Defining decision parameter $P_k = \Delta x(d1-d2)$

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

or $$P_k = 2\Delta y\, x_k - 2\Delta x\, y_k + 2\Delta y + \Delta x(2b+1)$$

or $$P_k = 2\Delta y\, x_k - 2\Delta x\, y_k + c \quad \cdots (III)$$

where $c = 2\Delta y + \Delta x(2b+1)$

Depending on the value of P_k, we select our next pixel.

Case I: $P_k >= 0$, then $d2 < d1$, so, it means y_k is nearer than $y_k - 1$.
Hence, (x_k+1, y_k) is plotted.

II: $P_k < 0$, then $d1 < d2$ so, $y_k - 1$ is nearer, hence the new co-ordinate will be (x_k+1, y_k-1).

Now for pixel (x_k+2), the points can be either (x_k+2, y_k-1) or (x_k+2, y_k-2) by looking at sign of decision parameter P_{k+1}.

so, $P_{k+1} = 2\Delta y\, x_{k+1} - 2\Delta x\, y_{k+1} + c$ ... (IV)

subtracting (III) from (IV)

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

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

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

or $P_{k+1} - P_k = 2\Delta y - 2\Delta x(y_{k+1} - y_k)$

This means, current decision parameter can be known if previous decision parameter is known. The term $(y_{k+1} - y_k)$ has two possibilities.

case I : $P_k < 0$, $y_{k+1} = y_k - 1$ so, $(y_{k+1} - y_k) = -1$
so, pixel is plotted at (x_k+1, y_k-1) and,

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

$$\boxed{\therefore P_{k+1} = P_k + 2\Delta y + 2\Delta x} \quad \text{—— } (*)$$

case II : $P_k >= 0$, $y_{k+1} = y_k$ so, $(y_{k+1} - y_k) = 0$.
so, pixel is plotted at (x_k+1, y_k) and,

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

$$\boxed{\therefore P_{k+1} = 2\Delta y + P_k} \quad \text{—— } (**)$$

(*) and (**) are the required decision parameters.

Initial decision parameter

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

or $$p_0 = 2\Delta y x_0 - 2\Delta x y_0 + 2\Delta y + \Delta x(2b+1)$$

here,

$$b = y_0 - m x_0 = y_0 - \frac{\Delta y}{\Delta x}(x_0)$$

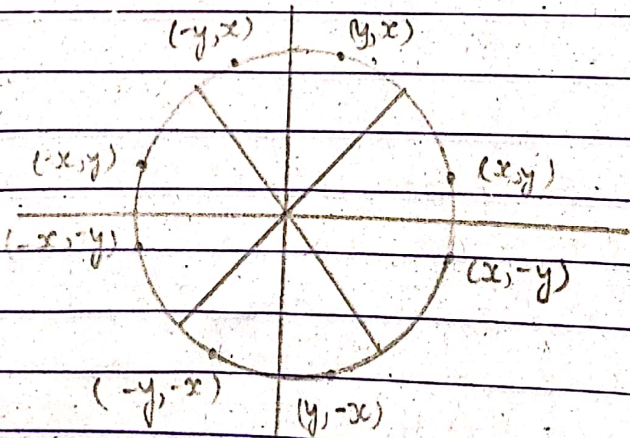
so, $$p_0 = 2\Delta y x_0 - 2\Delta x y_0 + 2\Delta y + \Delta x + 2\Delta x\left(y_0 - \frac{\Delta y}{\Delta x} x_0\right)$$

or, $$p_0 = \cancel{2\Delta y x_0} - \cancel{2\Delta x y_0} + 2\Delta y + \Delta x + \cancel{2\Delta x y_0} - \cancel{2\Delta y x_0}$$

$$\boxed{\therefore p_0 = 2\Delta y + \Delta x}$$

6. How symmetry property of circle reduces complexity to draw a circle? Derive a decision parameter for mid-point circle algorithm assuming the start position at $(-r, 0)$ and points are to be generated along the curve path is anti-clockwise direction.

: The figure showing the symmetricity of circle is given below:



Here, we can see that for a point (x,y) in a circle, seven other symmetrical points can be plotted directly in other octants as the circle is symmetric-al. Basically when one point of a circle is known, other seven points are known which reduces the computation power needed to calculate the remaining points of the circle.

We know the equation of circle is, (with center as origin)

$$x^2 + y^2 = r^2$$

so,

fuction of circle $f_{circle}(x,y) = x^2 + y^2 + -r^2$

Now for any point (x,y) in a circle

$$f_{circle}(x,y) \begin{cases} > 0 & \text{if } (x,y) \text{ is outside the circle} \\ = 0 & \text{if } (x,y) \text{ is on the boundary of circle} \\ < 0 & \text{if } (x,y) \text{ is inside the circle} \end{cases}$$

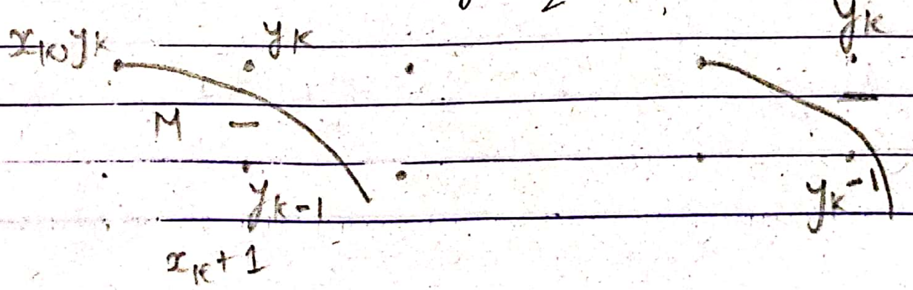
Now, for circle in anticlockwise direction, starting at $(-r,0)([x_k, y_k])$, the next point can be either,

$$(x_k+1, y_k) \text{ or } (x_k+1, y_k-1)$$

It is determined by the midpoint. So, decision parameter for next point is,

$$P_k = f_{circle}(x_k+1, y_k - 1/2)$$

or $P_k = (x_k+1)^2 = (y_k - \frac{1}{2})^2 - r^2 \; \ldots \ldots (1)$



Now, if $P_K < 0$, then midpoint is in circle, thus, y_K is closer to the boundary,

if $P_K > 0$, then midpoint is outside of circle, $y_K - 1$ is closer to the circle boundary.

Now the successive decision parameter is calculated by passing the next midpoint of $(x_{K+1}+1, y_{K+1})$ and $(x_{K+1}+1, y_{K+1}-1)$

so,

$$P_{K+1} = f(x_{K+1}+1, y_{K+1} - 1/2)$$

$$\text{or } P_{K+1} = (x_{K+1}+1)^2 + (y_{K+1} - 1/2)^2 - r^2 \quad \dots \text{(ii)}$$

Subtracting (i) from (ii)

$$P_{K+1} - P_K = (x_{K+1}+1)^2 + (y_{K+1} - 1/2)^2 - (x_K+1)^2 - (y_K - 1/2)^2$$

$$= (x_{K+1})^2 + 2x_{K+1} + \cancel{1} + (y_{K+1})^2 - y_{K+1} + \cancel{\tfrac{1}{4}} - x_K^2 - 2x_K - \cancel{1}$$

$$- y_K^2 + y_K - \cancel{1/4}$$

As, $x_{K+1} = x_K + 1$

$$= \cancel{x_K^2} + 2x_K + 1 + 2x_K + 2 + (y_{K+1})^2 - y_{K+1} - \cancel{x_K^2} - \cancel{2x_K}$$

$$- y_K^2 + y_K$$

$$P_{K+1} - P_K = 2(x_K + 1) + (y_{K+1}^2 - y_K^2) - (y_{K+1} - y_K) + 1$$

CamScanner

Now, $y_{k+1} = y_k$ or $y_k - 1$ depending on sign of p_k

If $p_k < 0$ then, $y_{k+1} = y_k$ then,

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

$$\therefore \boxed{p_{k+1} = p_k + 2(x_k+1)+1}$$

else,

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

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

or, $p_{k+1} = p_k + 2(x_k+1) - 2y_k + 1 + 1 + 1$

or $p_{k+1} = p_k + 2x_{k+1} - 2y_k + 2 + 1$

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

$$\boxed{\text{or } p_{k+1} = p_k + 2x_{k+1} - 2y_{k+1} + 1}$$

Initial decision parameter at starting point $(-r, 0)$

$$p_0 = (x_0+1)^2 + (y_0 - 1/2)^2 - r^2$$

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

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

$$\boxed{p_0 = \frac{5}{4} - 2r}$$

#

10. Digitize a line with end points A(20,10) and B(30,18) using Bresenham's Algorithm.

for $m < 1$)

$$P_{k+1} = P_k + 2\Delta y \quad , P_k < 0$$
$$P_{k+1} = P_k + 2\Delta y - 2\Delta x , P_k > 0$$

Solⁿ: Given, $(x_1, y_1) = (20, 10)$

$(x_2, y_2) = (30, 18)$

Now,

$$m = \frac{\Delta y}{\Delta x} = \frac{18-10}{30-20} = \frac{8}{10} = 0.8$$

as, slope (m) < 1, x is increased by one. Now,

K	P_k	x_{k+1}	y_{k+1}	(x_{k+1}, y_{k+1})
0	2x 8 − 10 = 6	20+1 = 21	10 + 1 = 11	(21, 11)
1	6 + 16 − 20 = 2	21+1 = 22	11+1 = 12	(22, 12)
2	2 + 16 − 20 = −2	22+1 = 23	12	(23, 12)
3	−2 + 16 = 14	23+1 = 24	12+1 = 13	(24, 13)
4	14+16−20 = 10	24+1 = 25	13+1 = 14	(25, 14)
5	10+16−20 = 6	25+1 = 26	14+1 = 15	(26, 15)
6	6+16−20 = 2	26+1 = 27	15+1 = 16	(27, 16)
7	2+16−20 = −2	27+1 = 28	16	(28, 16)
8	−2+16 = 14	28+1 = 29	16+1 = 17	(29, 17)
9	14+16−20 = 10	29+1 = 30	17+1 = 18	(30, 18)

14. What do you mean by homogeneous co-ordinates? Rotate a triangle A(5, 6), B(6,2) and C(4,1) by 45 degree anticlockwise about an arbitrary pivot point (4, 3).

ans:- The representation of a 2x2 matrix by using an extra parameter to make it 3x3 matrix for simplifying the mathematical operations for transformation is homogeneous co-ordinates.

Rotation, reflection, scaling $\leftarrow \begin{bmatrix} \boxed{\begin{matrix} a & b \\ d & e \end{matrix}} & \boxed{\begin{matrix} c \\ f \end{matrix}} \\ \boxed{\begin{matrix} g & h \end{matrix}} & \boxed{i} \end{bmatrix}$ $\begin{matrix} c \rightarrow \text{Translation} \\ i \rightarrow \text{'h'} \\ g\ h \downarrow \text{projection.} \end{matrix}$

From above question,

Composite matrix (C.M.) = $\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos45° & -\sin45° & 0 \\ \sin45° & \cos45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$

$= \cancel{\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} \cancel{0.707} & \cancel{-0.707} & \cancel{-0.707} \\ & & \\ & & \end{bmatrix}$

$= \begin{bmatrix} 0.7071 & -0.7071 & 3.292 \\ 0.7071 & 0.7071 & -1.949 \\ 0 & 0 & 1 \end{bmatrix}$

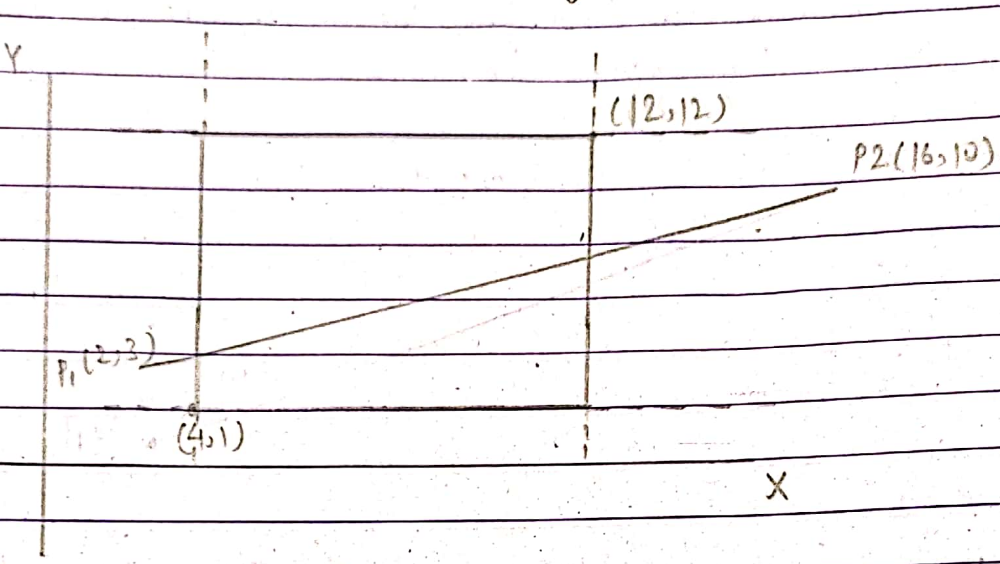
So, new co-ordinates,

$P' = C.M. \times P = \begin{bmatrix} 0.7071 & -0.7071 & 3.292 \\ 0.7071 & 0.7071 & -1.949 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 6 & 4 \\ 6 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2.58 & 6.12 & 5.41 \\ 5.82 & 3.70 & 1.58 \\ 1 & 1 & 1 \end{bmatrix}$

New co-ordinates are, A'(2,5), B'(6,3) and (5,1)

18. Clip the line P1P2 with P1(2,3) and P2(16,10) with clip window have diagonal co-ordinate (4,1) and (12,12) using Liang - Barkey line clipping method and Cohen - Sutherland line clipping algorithm.

=



Using Cohen - Sutherland Algorithm [TBRL]

The, $x_{min} = 4$. $x_{max} = 12$

$y_{min} = 1$ $y_{max} = 12$ So,

P1(2,3) and P2(16,10) are the line points. Now assigning the code for.

P1(2,3) = 0001

P2(16,10) = 0010

T	: $y > y_{max}$
B	: $y < y_{min}$
R	: $x > x_{max}$
L	: $x < x_{min}$

Logical AND gives 0000 so, clipping is needed,

Now,

$P_1' = (x_{min}, y')$

Now, $y' = y_1 + m(4-2)$

$y' = 3 + \frac{10-3}{16-2}(4-2)$

$y' = 3 + \frac{\cancel{7}}{\cancel{14}} \times \cancel{2}$

$y' = 3 + 1$

$y' = 4$

so, $P_1' = (4,4)$

Now, taking code for $P'(4,4) = 0000$

taking code for $P2(16,10) = 0010$

Logical ANDing gives 0000 so, clipping is needed

$P_2' = (x_{max}, y'')$

so,

$y'' = 3 + m(12-2)$

$y'' = 3 + \frac{10-3}{16-2} \times 10$

$y'' = 3 + \frac{\cancel{7}}{\cancel{14}\,2} \times \cancel{10}\,5$

$y'' = 3+5$

$y'' = 8$

so, $P_2' = (12,8)$ and code is 0000

Hence,

Clipped part is $P_1'(4,4)$ and $P_2'(12,8)$

Liang Barkey Algorithm

here,

$\Delta x = 16 - 2 = 14$ $\quad$ $(x_1, y_1) = (2, 3)$

$\Delta y = 10 - 3 = 7$ $\quad$ $(x_2, y_2) = (16, 10)$

Now,

K	P_k	q_k	r_k	
1	$-\Delta x$ -14	$x_1 - x_{min}$ $2 - 4 = -2$	q_k/p_k $-2/-14 = 1/7$	candidate for t_1
2	Δx 14	$x_{max} - x_1$ $12 - 2 = 10$	q_k/p_k $10/14 = 5/7$	candidate for t_2
3	$-\Delta y$ -7	$y_1 - y_{min}$ $3 - 1 = 2$	q_k/p_k $2/-7$	candidate for t_1
4	Δy 7	$y_{max} - y_1$ $12 - 3 = 9$	q_k/p_k $9/7$	candidate for t_2

So,

$t_1 = \max\left(0, \frac{1}{7}, \frac{-2}{7}\right)$ $\quad$ $t_2 = \min\left(1, \frac{5}{7}, \frac{9}{7}\right)$

$= \frac{1}{7}$ $\quad$ $= \frac{5}{7}$

Now,

$x_1' = x_1 + t_1 \Delta x$ $\quad$ $y_1' = y_1 + t_1 \Delta y$

$= 2 + \frac{1}{7} \times 14$ $\quad$ $= 3 + \frac{1}{7} \times 7$

$= 4$ $\quad$ $= 4$

$\therefore (x_1', y_1') = (4, 4)$

$x_2' = x_1 + t_2 \Delta x$

$= 2 + \frac{5}{7} \times 14$

$= 2 + 10$

$= 12$

$y_2' = y_1 + t_2 \Delta y$

$= 3 + \frac{5}{7} \times 7$

$= 8$

$(x_2', y_2') = (12, 8)$

Hence,
Required points for clipping are (4,4) and (12,8)