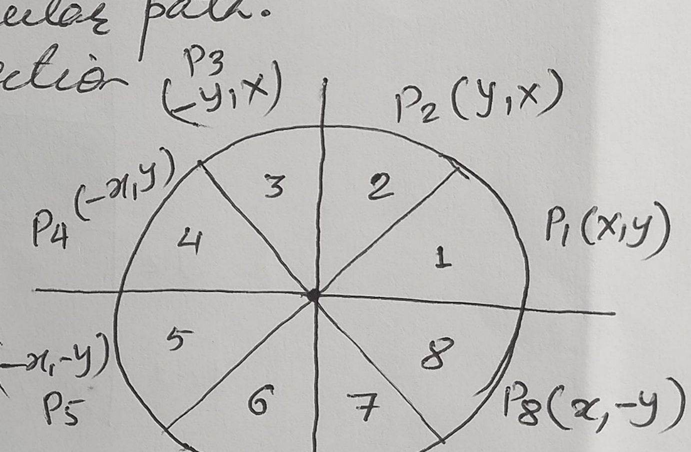
Mid-Point Algorithm of Circle

It is based on incremental calculation of decision parameter. It is used to find closest pixel to the circumference at each sampling step. The idea in this approach is to test half way position between two pixels to determine if this mid-point is inside or outside the circle boundary. We use octate symmetric property to find the points over an constant. We can take unit ~~void~~ step in positive x-direction and by using decision parameter, we find the closest y positioned in the circular path.

Consider a circular section for $x=0, y=y$ where slope of the curve varies from 0 to 1.



Calculation of circle point (x,y) in one octave gives the circle point shown for other seven octaves. To apply the mid point, we define a circle function as:

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

The relative position at any point (x,y) can be determined by checking the sign of the circle function:

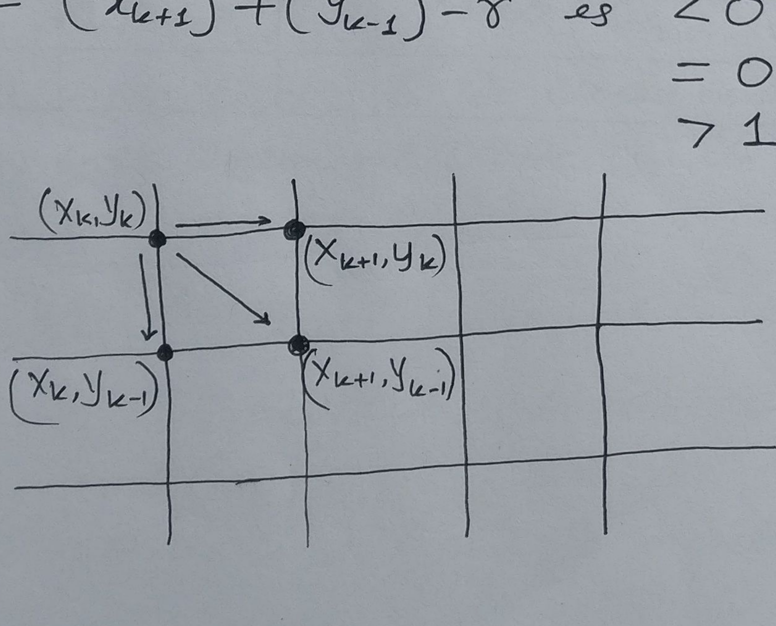
$$F_{circle}(x,y) = \begin{cases} <0 & \text{if } (x,y) \text{ is inside the circle boundary} \\ =0 & \text{if } (x,y) \text{ is on circle boundary} \\ >0 & \text{if } (x,y) \text{ is outside the circle boundary} \end{cases}$$

Thus, the circle function is the decision parameter in the midpoint algorithm and we can set up incremental calculations for this function.

In following figure we have plotted the pixel at (x_k, y_k), we next need to determine whether the pixel at position (x_{k+1}, y_k) or one at position (x_{k+1}, y_{k-1}) or at position (x_k, y_{k-1}) is closer to the circle.

First we check, if the diagonal point is inside or outside the circle i.e.

$$\Delta = (x_{k+1})^2 + (y_{k-1})^2 - r^2 \text{ is } \begin{array}{l} <0 \text{ or} \\ =0 \text{ or} \\ >1 \end{array}$$



If $\Delta < 0$, then the diagonal point is inside the circle i.e. we have to select one of the two point (x_{k+1}, y_k) or (x_{k+1}, y_{k-1}), the one which is closer to the circle boundary. For this we check for the midpoint between these two points i.e.

$$P_k = F_{circle}\left(x_k+1, y_k-\frac{1}{2}\right)$$
$$= (x_k+1)^2 + \left(y_k-\frac{1}{2}\right)^2 - r^2$$

If $P_k > 0$, this midpoint is outside the circle and the pixel on the scan line y_{k-1} is closer to the circle boundary otherwise the mid position is inside or on the circle boundary and we select the pixel on scan line y_k.

Similarly if $(\Delta > 0)$, then the diagonal point is outside the circles i.e we have to select one of the two points (x_{k+1}, y_{k-1}) and (x_k, y_{k-1}) the one which is more closer to the circle boundary. For this, once again we check for the midpoint between these two points i.e;

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

If $P_k > 0$, this midpoint is outside the circle and the pixel (x_k, y_{k-1}) is selected otherwise (inside or on the circle) and we select the pixel (x_{k+1}, y_{k-1})

If $\Delta = 0$, pixel (x_{k+1}, y_{k-1}) is selected.

The initial decision parameter at $(x_0, y_0) = (0, r)$

i.e. $P_0 = F_{circle}\left(1, r - \frac{1}{2}\right)$

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

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

$$\approx 1 - r$$

Successive decision parameters are obtained using incremental calculations. We obtain a recursive expression for the next decision parameter by evaluating the circle function at sampling position $x_{k+1} + 1 = x_k + 2$

$$P_{k+1} = f_{circle}\left(x_{k+1} + 1, y_{k+1} - \frac{1}{2}\right)$$

$$= \left[x_k + 1 + 1\right]^2 + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2$$

or $P_{k+1} = P_k + 2(x_k + 1) + \left(y_{k+1}^2 - y_k^2\right) + \left(y_{k+1} - y_k\right) + 1$

Where, y_{k+1} is either y_k or y_{k-1}, depending on the sign of P_k

If P_k is (-ve)

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

$$P_{k+1} = P_k + 2(x_k + 1) + 1$$

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

If P_k is (+ve)

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

$$\therefore P_{k+1} = P_k + 2(x_k+1) + 1 - 2y_k + 1$$

Thus, increments for obtaining P_{k+1} are either

$2x_{k+1} + 1$ (if P_k is -ve) or

$2x_{k+1} + 1 - 2y_{k+1}$ (if P_k is +ve)

and the evaluation of the terms $2x_{k+1}$ and $2y_{k+1}$ can be done as:

$$2x_{k+1} = 2x_k + 2$$

$$2y_{k+1} = 2y_k - 2$$

Mid Point Circle Algorithm

1) Input radius 'r' at circle center (x_k, y_k) and obtain the first point on the circumference of a circle centered on the origin as $(x_0\, y_0) = (0, r)$ i.e initialising starting point as $x_0 = 0$ and $y_0 = r$

2) Calculate the initial value of the decision parameter as $P_0 = 1 - r$

3) At each x_k position starting at $k=0$ perform the following :
   - If $P_k < 0$, the next point along the circle centered on $(0,0)$ is (x_{k+1}, y_k) and
     $P_{k+1} = P_k + 2(x_{k+1}) + 1$
   - Otherwise the next point along the circle is (x_k+1, y_k-1) and $P_{k+1} = P_k + 2(x_k+1) + 1 - 2(y_{k+1})$

where, $2x_{k+1} = 2x_k + 2$ and $2y_{k+1} = 2y_k - 2$.

4) Determine symmetric points in the other seven ~~quadrants~~ octants

5) Move each calculated pixel position (x, y) into circular path centered on (x_k, y_k) and plot the coordinate value $x = x + x_k$ and $y = y + y_k$

6) Repeat step 3 through 5 unit $x \geq y$.

Q Digitize a circle center at origin and radius 10 by using midpoint circle algorithm.

Solution

Center = (0,0) = (x_c, y_c)

radius $(r) = 10$

$(x_0 y_0) = (0, r) = (0, 10)$

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

$P_0 = -9$ ie < 0

K	P_k	x_{k+1}, y_{k+1}	$2x_{k+1}$	$2y_{k+1}$	P_{k+1}
0	-9	(0+1=1, 10)	2	20	-6
1	-6	(2,10)	4	20	-1
2	-1	(3,10)	6	20	6
3	6	(4,9)	8	18	-3
4	-3	(5,9)	10	18	8
5	8	(6,8)	12	16	5
6	5	(7,7)	14	14	6

Q. Digitize a circle $(x-2)^2+(y-3)^2=25$

Solution

$$(x-2)^2+(y-3)^2=25 \quad \text{[given eqn]}$$

radius $(r)=\sqrt{25}=5$

$(x_k, y_k) = (2,3) \Rightarrow$ center not at origin

center $(0, r) = (0,5)$ initializing point

$$P_0 = 1-r = 1-5 = -4 < 0$$

k.	P_k	x_{k+1}, y_{k+1}	P_{k+1}
0	-4	$(0+1=1,5)=(1,5)$	$-4+2\cdot1+1=-1$
1	-1	$(1+1=2,5)=(2,5)$	$-1+2\cdot2+1=4$
2	4	$(2+1=3,5-1=4)=(3,4)$	$4+2\cdot3+1-2\cdot4=3$
3	3	$(3+1=4,4-1=3)=(4,3)$	—

$4 \geq 3 \Rightarrow$ process stops.
$(x \geq y)$