Circle

The equation of a circle is given by

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

To apply the midpoint method, we define a circle function

as $F_{circle}(x,y) = x^2 + y^2 - r^2$

now $F_{circle}(x,y) < 0$ **if(x,y) is inside the circle boundary**

$= 0$ **if(x,y) is on the circle boundary**

> 0 **if(x,y) is outside the circle boundary**

This circle function $F_{circle}(x,y)$ **serves as the decision parameter**

Select next pixel along the circle path according to the sign of circle function evaluated at the midpoint between two candidate pixels.

Start at (0,y) take unit steps in 'x' direction (sample in 'x' direction $x_{k+1} = x_k + 1$**)**

Assuming position (x_k, y_k) **has been selected at 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 (circle function). The decision parameter is the circle function evaluated at the midpoint between these two pixels**

$$P_k = F_{circle}(x_k + 1, y_k - ½)$$

$$= (x_k + 1)^2 + (y_k - ½)^2 - r^2 \text{ ---------- } (i)$$

At the next sampling position $(x_{k+1} + 1 = x_k + 2)$ **, the decision parameter is evaluated as**

$$P_{k+1} = F_{circle}(x_{k+1} + 1, y_{k+1} - ½)$$

$$= [(x_k + 1) + 1]^2 + (y_{k+1} - ½)^2 - r^2 \text{ ---------- } (ii)$$

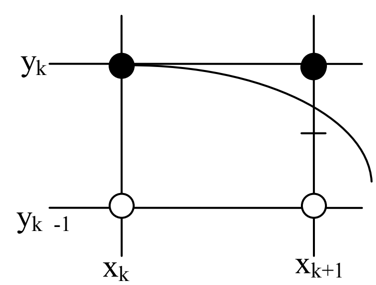
Now subtracting eq (i) and (ii),

$$P_{k+1} = P_k + 2(x_k + 1) + (y^2_{k+1} - y^2_k) - (y_{k+1} - y_k) + 1 \text{ ------- } (iii)$$

where y_{k+1} **is either** y_k **or** $y_k - 1$ **depending on the sign of** P_k**.**

<u>Case 1:</u>

If $P_k < 0$ **then the mid point is inside the circle, so pixel on scanline '**y_k**' is closer to the circle boundary and** $y_{k+1} = y_k$



From equation (iii)

or $P_{k+1} = P_k + 2x_{k+1} + 1$ **------- (a)**

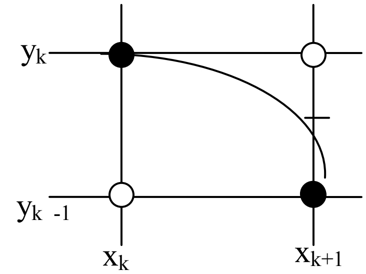
Where $x_{k+1} = x_k + 1$

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

Case 2:

If $P_k >= 0$ then the mid point is outside or on the boundary of the circle, so we select the pixel on scan line '$y_k - 1$' then $y_{k+1} = y_k - 1$

i.e. from equation (iii)



or $P_{k+1} = P_k + 2x_{k+1} - 2y_{k+1} + 1$ ---------- **(b)**

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

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

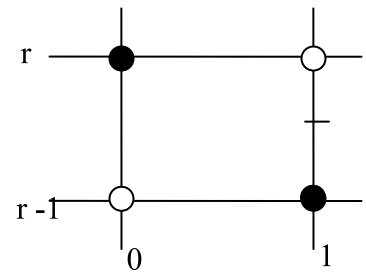
The initial decision parameter P_0 is obtained by evaluating the circle function at the start position $(x_0, y_0) = (0, r)$

Next pixel to plot is either (1,r) or (1,r - 1)

So, midpoint coordinate position is (1, r - ½)

$F_{circle}(1, r - ½) = 1 + (r - ½)^2 - r^2$

Thus,



$$P_0 = 5/4 - r$$

If the radius 'r' is specified as an integer, we can simply round P_0 to $P_0 = 1 - r$ (for 'r' an integer)

Midpoint Circle Algorithm

1. **Input radius r and circle center $(x_c\ y_c)$, and obtain the first point on the circumference of a circle centered on the origin as $(x_0, y_0) = (0, r)$**
2. **Calculate the initial value of the decision parameter as**

   $$P_0 = 5/4 - r$$

3. **At each x_k position, starting at k = 0, perform the following test:**

   If $P_k < 0$, the next point along the circle centered on (0,0) is (x_{k+1}, y_k) and $P_{k+1} = P_k + 2x_{k+1} + 1$

   Otherwise, the next point along the circle is $(x_k + 1, y_k - 1)$ and $P_{k+1} = P_k + 2x_{k+1} - 2y_{k+1} + 1$

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

4. **Determine symmetry points in the other seven octants.**
5. **Move each calculated pixel position (x, y) onto the circular path centered on (x_c, y_c) and plot the coordinate values:**

   $x = x + x_c,\ y = y + y_c$

6. **Repeat steps 3 through 5 until x >= y.**