Midpoint Circle Algorithm:

A circle is defined as a set of points that are all at a given distance *r* from a center positioned at. (x_c, y_c).
This is represented mathematically by the equation

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad \text{------------- (1)}$$

Using equation (1) we can calculate the value of y for each given value of x as

$$y = y_c \pm \sqrt{r^2 - (x_c - x)^2} \quad \text{---------------- (2)}$$

Thus one could calculate different pairs by giving step increments to x and calculating the corresponding value of y. But this approach involves considerable computation at each step and also the resulting circle has its pixels sparsely plotted for areas with higher values of the slope of the curve.
Midpoint Circle Algorithm uses an alternative approach, wherein the pixel positions along the circle are determined on the basis of incremental calculations of a decision parameter.

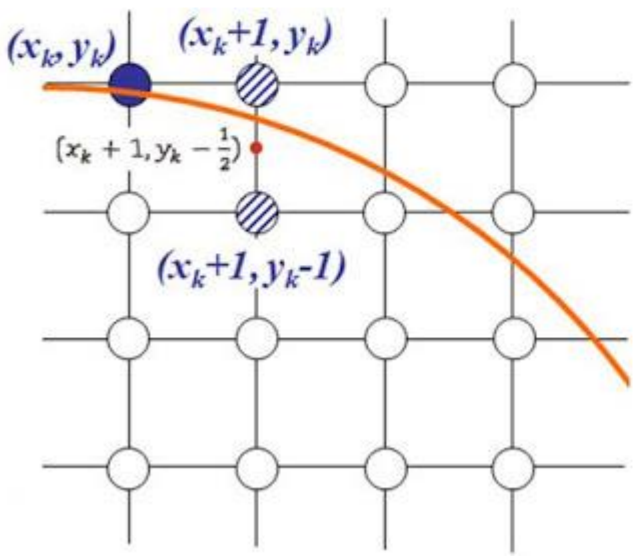
Let

$$f(x, y) = (x - x_c)^2 + (y - y_c)^2 - r^2 \quad \text{-------------- (3)}$$

Thus f(x,y)=0 represents the equation of a circle.
Further, we know from coordinate geometry, that for any point , the following holds:

1. $f(x, y) = 0 \quad \Rightarrow$ *The point lies on the circle.*
2. $f(x, y) < 0 \quad \Rightarrow$ *The point lies within the circle.*
3. $f(x, y) > 0 \quad \Rightarrow$ *The point lies outside the circle.*

In Midpoint Circle Algorithm, the decision parameter at the *k*<i></i>*th* step is the circle function evaluated using the coordinates of the midpoint of the two pixel centres which are the next possible pixel position to be plotted.
Let us assume that we are giving unit increments to x in the plotting process and determining the y position using this algorithm. Assuming we have just plotted the k^{th} pixel at (X_k, Y_k), we next need to determine whether the pixel at the position (X_{k+1}, Y_k) or the one at (X_{k+1} , Y_{k-1}) is closer to the circle.



Our decision parameter p_k at the k^{th} step is the circle function evaluated at the midpoint of these two pixels.
The coordinates of the **midpoint** of these two pixels are (X_k+1, $Y_k-1/2$).

Thus p_k

$$p_k = f\left(x_k + 1, y_k - \frac{1}{2}\right) = (x_k + 1)^2 + (y_k - \frac{1}{2})^2 - r^2$$ --------------- (4)

x
Successive decision parameters are obtained using incremental calculations, thus avoiding a lot of computation at each step. We obtain a recursive expression for the next decision parameter i.e. at the *k+1th* step, in the following manner.
Using Equ. (4), at the k*+1th* step, we have:

$$p_k = f\left(x_k + 1, y_k - \frac{1}{2}\right) = (x_k + 1)^2 + (y_k - \frac{1}{2})^2 - r^2$$

$$p_{k+1} = f\left(x_{k+1} + 1, y_{k+1} - \frac{1}{2}\right) = (x_{k+1} + 1)^2 + (y_{k+1} - \frac{1}{2})^2 - r^2$$

$$\text{Or, } p_{k+1} = (x_k + 1 + 1)^2 + (y_{k+1} - \frac{1}{2})^2 - r^2$$

$$\text{Or, } p_{k+1} = (x_k + 2)^2 + (y_{k+1} - \frac{1}{2})^2 - r^2 \text{ ---------------------------(5)}$$

(5)-(4) gives

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

$$\text{Or, } p_{k+1} = p_k + (2x_k + 3).1 + (y_{k+1} + y_k - 1)(y_{k+1} - y_k) \text{ -------(6)}$$

Now if $P_k <= 0$, then the midpoint of the two possible pixels lies within the circle, thus north pixel is nearer to the theoretical circle. Hence, $Y_{k+1} = Y_k$. Substituting this value of in Equ. (6), we have

$$p_{k+1} = p_k + (2x_k + 3) + (y_k + y_k - 1)(y_k - y_k)$$

$$\text{Or, } p_{k+1} = p_k + (2x_k + 3)$$

If $p_k > 0$ then the midpoint of the two possible pixels lies outside the circle, thus south pixel is nearer to the theoretical circle. Hence, $Y_{k+1} = Y_k - 1$. Substituting this value of in Equ. (6), we have

$$p_{k+1} = p_k + (2x_k + 3) + (y_k - 1 + y_k - 1)(y_k - y_k - 1)$$

$$\text{Or, } p_{k+1} = p_k + 2(x_k - y_k) + 5$$

For the boundary condition, we have x=0, y=r. Substituting these values in (4), we have

$$p_0 = (0 + 1)^2 + (r - \frac{1}{2})^2 - r^2 = 1 + r^2 + \frac{1}{4} - r - r^2 = \frac{5}{4} - r$$

For integer values of pixel coordinates, we can approximate $P_0 = 1 - r$,

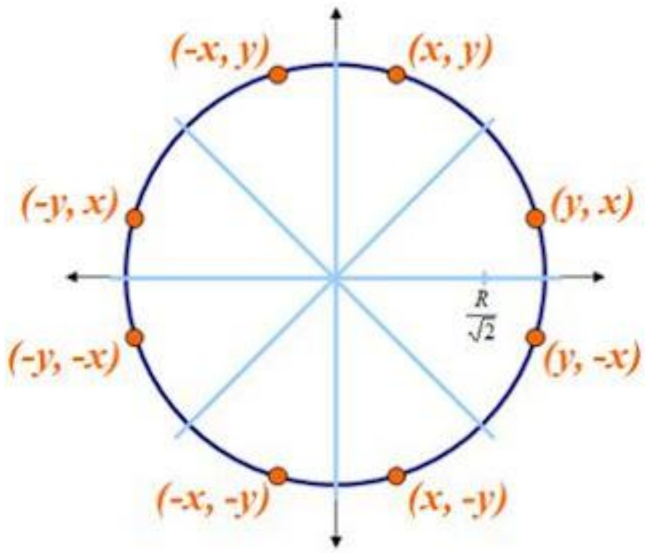
Thus we have:

If $p_k \leq 0$: $\quad y_{k+1} = y_k$ and $p_{k+1} = p_k + (2x_k + 3)$

If $p_k > 0$: $\quad y_{k+1} = y_k - 1$ and $p_{k+1} = p_k + 2(x_k - y_k) + 5$

Also, $p_0 = 1 - \mathrm{r}$

Drawing the circle:
We can reduce our calculation drastically (8th fraction) by making use of the fact that a circle has 8 way symmetry. Thus after calculating a pixel position (x,y) to be plotted, we get 7 other points on the circle corresponding to it. These are:
(x,y); (x,-y); (-x,y); (-x,-y); (y,x); (y,-x); (-y,x); (-y,-x)


(-x, y)
(x, y)
(-y, x)
(y, x)
$\frac{R}{\sqrt{2}}$
(-y, -x)
(y, -x)
(-x, -y)
(x, -y)