CIRCLE DRAWING:

Given: Center: (h,k) Radius: r Equation: (x-h)2 + (y-k)2 = r2

To simplify we'll translate origin to center Simplified Equation:

x2 + y2 = r2

Only considers circles centered at the origin with integer radii.

Can apply translations to get non-origin centered circles.

Explicit equation: y = +/- sqrt($R^2 - X^2$)

Implicit equation: F(x,y)= $X^2 + Y^2 - R^2$ =0

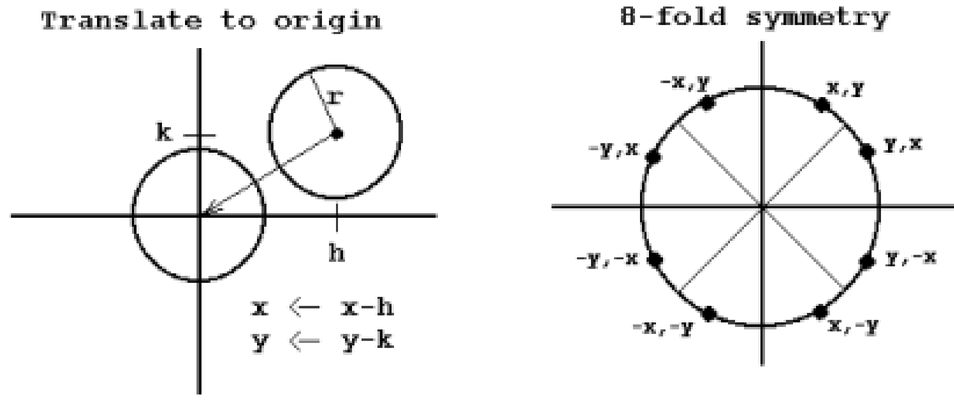
Note: Implicit equations used extensively for advanced modeling

Use of Symmetry:

Only need to calculate one octant.

One can get points in the other 7 octants.

Only need consider 450 of the circle use symmetry to find other points.



Circle has 8-fold symmetry

So only need to plot points in 1st octant

$\Delta x > \Delta y$ so step in x direction

- A circle is a symmetrical figure.
- Any circle-generating algorithm can take advantage of the circle's symmetry to plot eight points for each value that the algorithm calculates.
- Eight-way symmetry is used to reflecting each calculated point around each 450 axis.
- For example, if point 1 were calculated with a circle algorithm, seven more points could be found by reflection.

Defining a Circle

• There are two standard methods of mathematically defining a circle centered at the origin.

• Polynomial Method • Trigonometric Method

• The first method **(Polynomial Method)** defines a circle with the second-order polynomial equation: y 2 = r2 – x 2 ,

where x = the x coordinate,

y = the y coordinate and

r = the circle radius.

• With this method, each x coordinate in the sector, from 900 to 450 , is found by stepping

• x from 0 to $r/(\sqrt{2})$, and

• each y coordinate is found by evaluating $\sqrt{(r2 - x\ 2\ )}$ for each step of x.

• This is a very inefficient method, however, because for each point both x and r must be squared and subtracted from each other, then the square root of the result must be found.

The second method **(Trigonometric Method)** of defining a circle makes use of trigonometric functions: $x = r \cos \alpha$

and $y = r \sin \alpha$

where α = current angle

r = circle radius

x = x coordinate

y = y coordinate

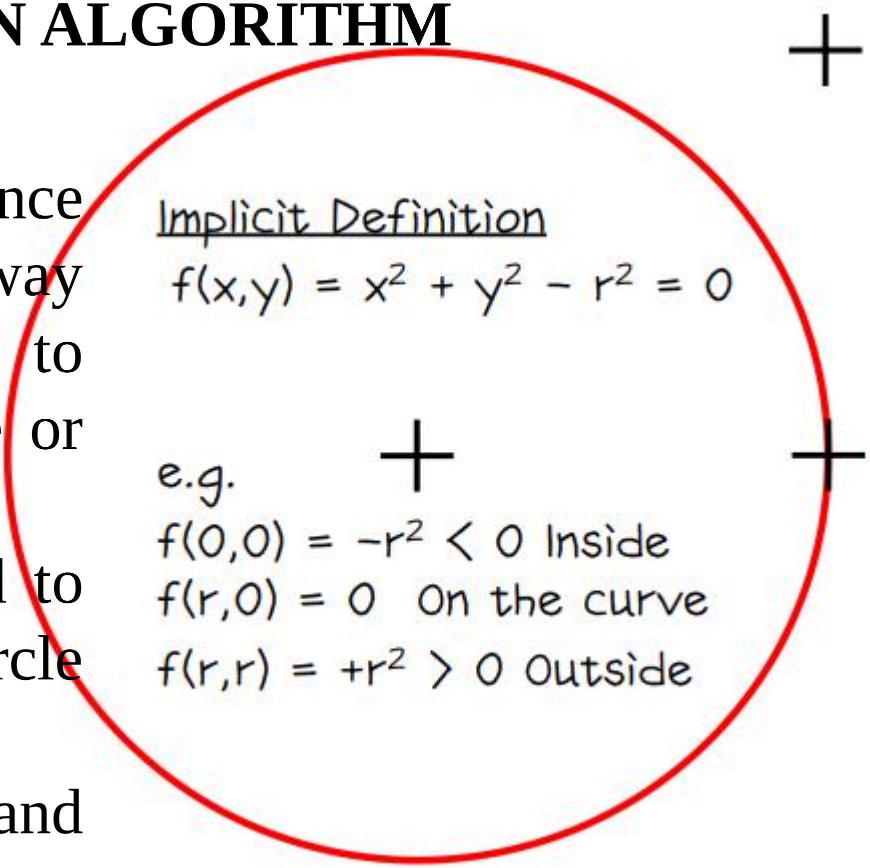
• By this method, α is stepped from 0 to $\pi/4$, and each value of x and y is calculated.

• However, computation of the values of $\sin \alpha$ and $\cos \alpha$ is even more timeconsuming than the calculations required by the first method.

MIDPOINT CIRCLE GENERATION ALGORITHM

we summarize the algorithm

- A method for direct distance comparison is to test the halfway position between two pixels to determine if this midpoint is inside or outside the circle boundary.
- This method is more easily applied to other conics, and for an integer circle radius.
- we sample at unit intervals and determine the closest pixel position to the specified circle path at each step.



- Along the circle section from $x = 0$ to $x = y$ in the first quadrant, the slope of the curve varies from 0 to - 1.
- Therefore, we can take unit steps in the positive x direction over this octant and use a decision parameter to determine which of the two possible y positions is closer to the circle path at each step.

To apply the midpoint method. we define a circle function:

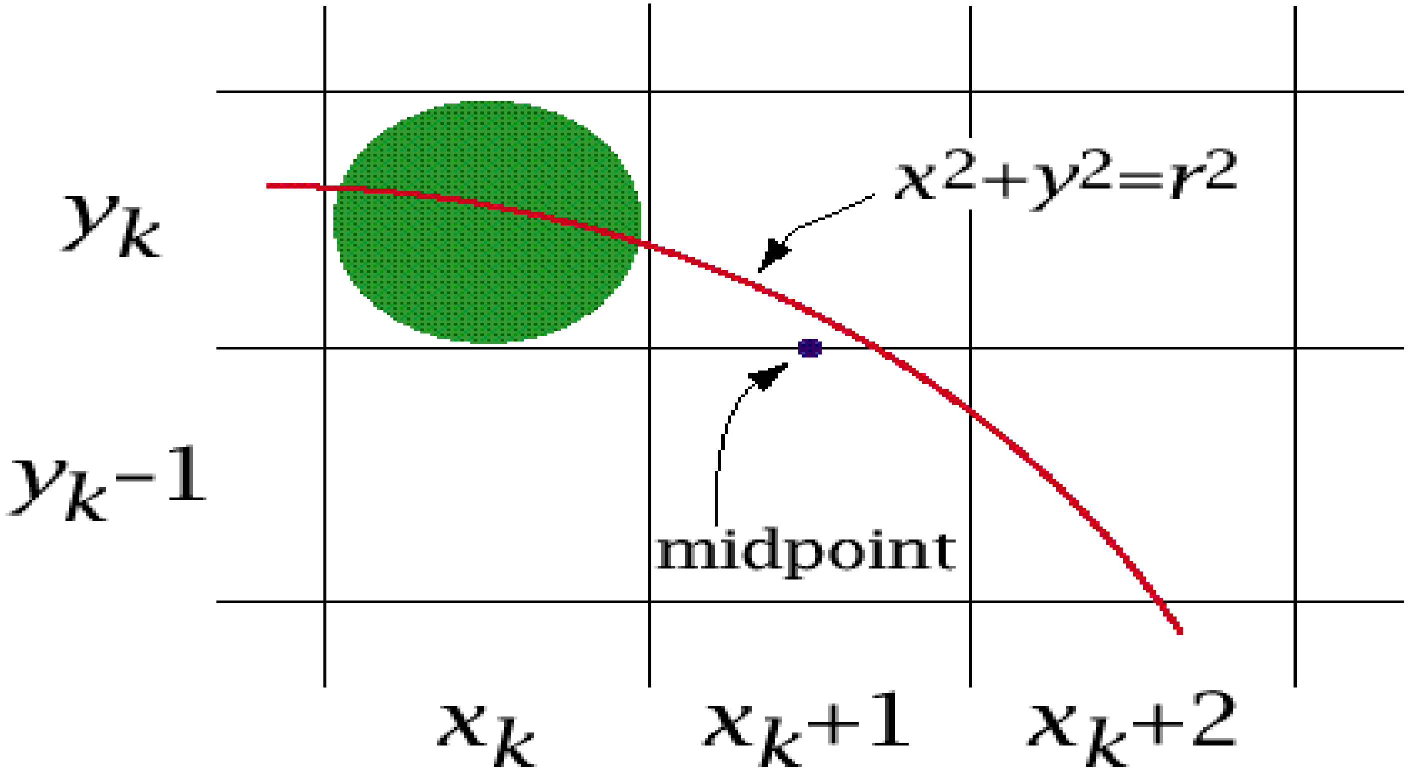
$$f_{circle}(x, y) = x^2 + y^2 - r^2$$

Any point (x, y) on the boundary of the circle with radius r satisfies the equation $f_{circle}(x, y) = 0$.

If $f_{circle}(x, y) < 0$, the point is inside the circle boundary ,

If $f_{circle}(x, y) > 0$, the point is outside the circle boundary,

If $f_{circle}(x, y) = 0$, the point is on the circle boundary.



Mid-point Circle Algorithm Calculating p_k

First, set the pixel at **(xk , yk)**, next determine whether the pixel **(xk + 1, yk)** or the pixel **(xk + 1, yk – 1)** is closer to the circle using:

$$p_k = f_{circle}(xk + 1, yk - ½)$$
$$= (xk + 1)^2 + (yk - ½)^2 - r^2$$

Now Calculating P_{k+1}

$$P_{k+1} = fcircle(xk + 1 + 1, yk + 1 - ½)$$
$$= [(xk + 1) + 1]^2 + (yk + 1 - ½)^2 - r^2$$

Or

$$Pk + 1 = pk + 2(xk + 1) + (y^2k + 1 - y^2k) - (yk + 1 - yk) + 1$$

Now Calculating p_0

$$p_0 = fcircle(1, r - ½) = 1 + (r - ½)^2 - r^2 \text{ or}$$

$$p_0 = 5/4 - r \cong 1 - r$$

2. At each x_k position, starting at **k = 0**, perform the following test:
   If $p_k < 0$, plot $(x_k + 1, y_k)$ and $p_{k+1} = p_k + 2x_{k+1} + 1$,
   Otherwise,
   plot $(x_k + 1, y_k - 1)$ and $p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$,
   where $2x_{k+1} = 2x_k + 2$ and $2y_{k+1} = 2y_k - 2$.

Mid-point Circle Algorithm – Steps

1. Input radius **r** and circle center **(x_c, y_c)**. set the first point **$(x_0, y_0) = (0, r)$.**

2. Calculate the initial value of the decision parameter as

   $p_0 = 1 – r$.

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

   a) If **$p_k < 0$**, plot **$(x_k + 1, y_k)$** and **$p_{k+1} = p_k + 2x_{k+1} + 1$**,

   $= p_k+2x_k+3$

   b) for **$p_k >= 0$** Otherwise,

   plot **$(x_k + 1, y_k – 1)$** and **$p_{k+1} = p_k + 2x_{k+1} + 1 – 2y_{k+1}$**,

   $= p_k + 2x_{k+1} +3 - 2y_k +2$

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

(Example): Given a circle radius r = 10, demonstrate the midpoint circle algorithm by determining positions along the circle octant in the first quadrant from x = 0 to x = y.

<u>Solution:</u>

$p_0 = 1 - r = -9$

Plot the initial point $(x_0, y_0) = (0,10)$

a) If $\mathbf{p_k < 0}$, plot $\mathbf{(x_k + 1, y_k)}$ and

$\mathbf{p_{k+1} = p_k + 2x_{k+1} + 1}$,

$= p_k + 2x_k + 3$

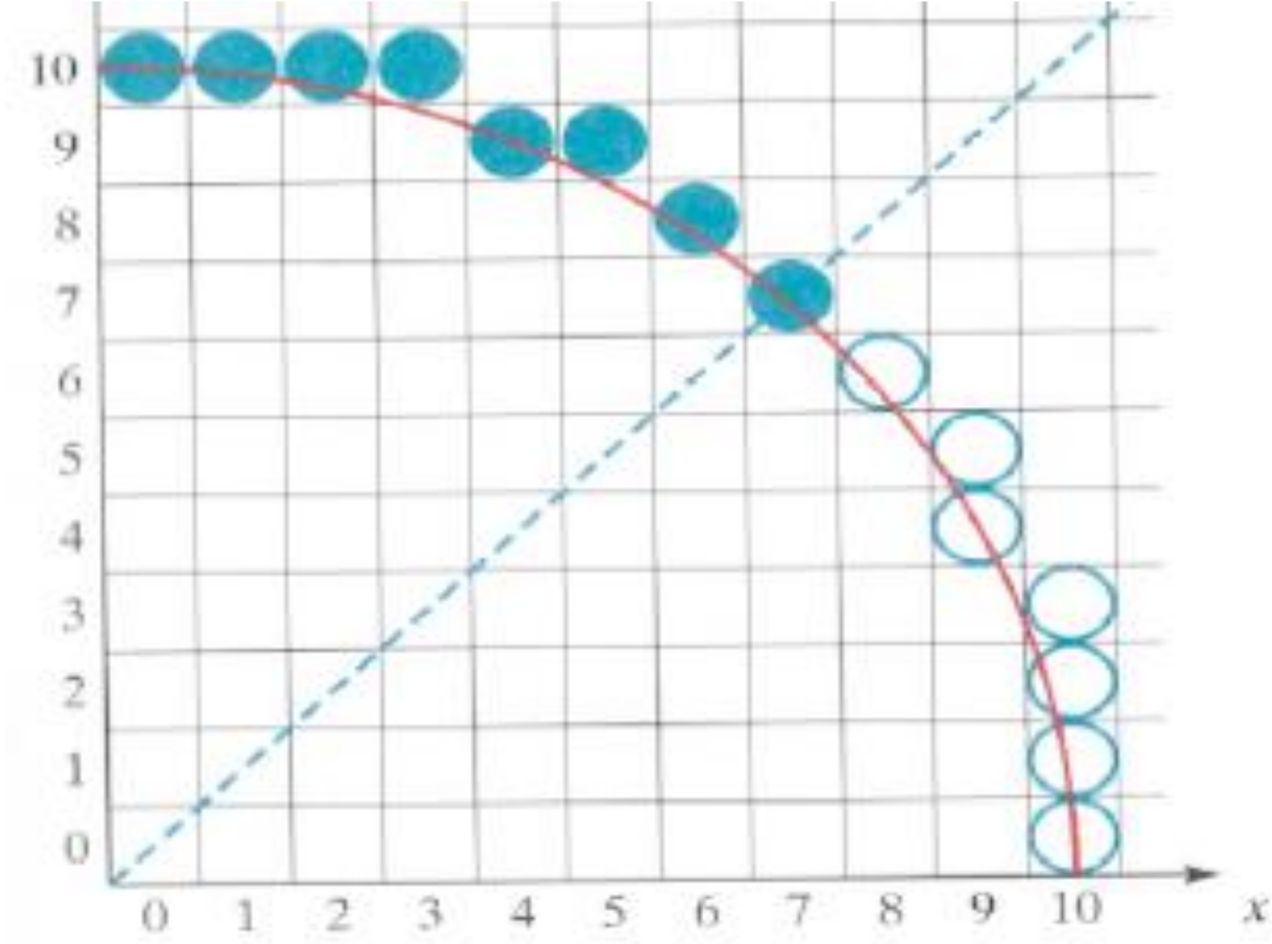
b) for $\mathbf{p_k >= 0}$ Otherwise,

plot $\mathbf{(x_k + 1, y_k - 1)}$ and $\mathbf{p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}}$,

=

$\mathbf{p_k + 2x_{k+1} + 3 - 2y_k + 2}$

K	P_k	$(x_{k+1}$	y_{k+1}
		0	10
0	– 9	1	10
1	– 6	2	10
2	– 1	3	10
3	6	4	9
4	– 3	5	9
5	8	6	8
6	5	7	7


10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10
x

Example (2) Given a circle radius r = 15, demonstrate the midpoint circle algorithm by determining positions along the circle octant in the first quadrant from x = 0 to x = y.

$$p_0 = 1 - r = -14$$

plot the initial point

$$(x_0 , y_0) = (0, 15),$$

a) If $\mathbf{p_k < 0}$, plot $\mathbf{(x_k + 1, y_k)}$ and $\mathbf{p_{k+1} = p_k + 2x_{k+1} + 1}$,

$= p_k+2x_k+3$

b) for $\mathbf{p_k >= 0}$ Otherwise,

plot $\mathbf{(x_k+ 1, y_k - 1)}$ and $\mathbf{p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}}$,

=

$\mathbf{p_k + 2x_{k+1} +3 - 2y_k +2}$

K	P_k	x_{k+1}	y_{k+1}
		0	15
0	– 14	1	15
1	– 11	2	15
2	– 6	3	15
3	1	4	14
4	– 18	5	14
5	– 7	6	14
6	6	7	13
7	– 5	8	13
8	12	9	12
9	7	10	11
10	6	11	10