LINE DRAWING ALGORITHM

2021-2022

Mitocaru Ian, Orghici Cosmin

School: Colegiul National Iași, România
Teacher: Gabriela-Elena Zanoschi
Researcher and affiliation: Aurelian Claudiu Volf, Universitatea "Al. I. Cuza", Iaşi

Text of the problem

Two pixels are displayed on a computer screen. Write a drawing algorithm for the line segment that joins these pixels on the screen. You can also solve similar problems, such as drawing a circle, when you know its center and radius.

1. Line drawing algorithm

Overview

The basic ”line drawing” algorithm used in computer graphics is Bresenham's Algorithm. This algorithm was developed to draw lines on digital plotters, but has found widespread usage in computer graphics. The algorithm is fast – it can be implemented with integer calculations only – and very simple to describe.

Conventions:

- The bottom-left is $(0,0)$ such that pixel coordinates increase in the right and up directions (e.g. that the pixel at $(7,5)$ is directly above the pixel at $(7,4)$).
- The pixel centers have integer coordinates.

There are three cases that we need to take into consideration:

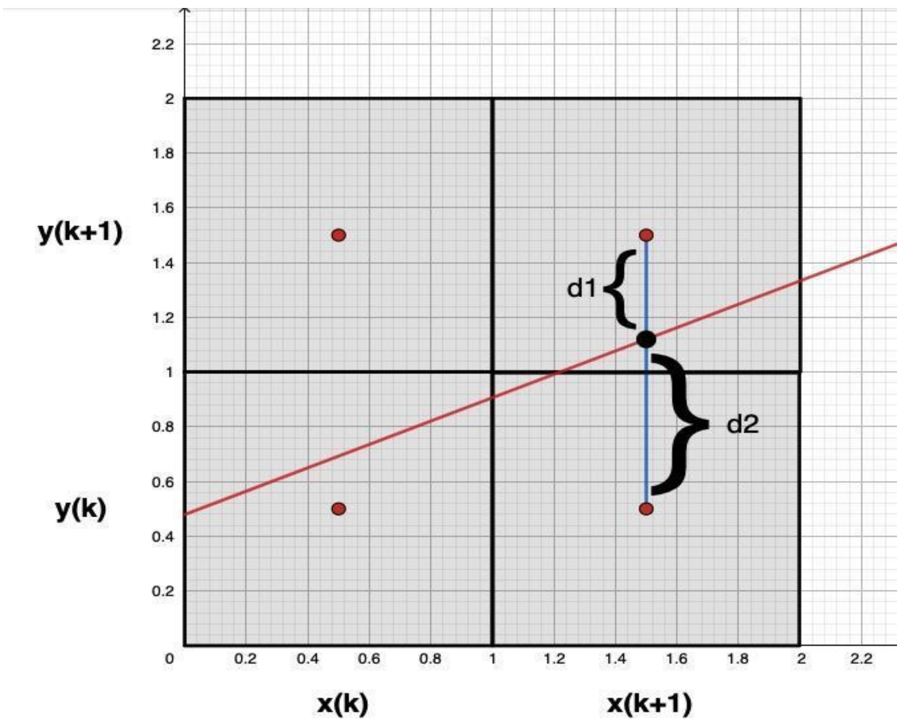
1. The angle between the line and the Ox axis is bigger than 45 degrees
2. The angle between the line and the Ox axis is 45 degrees
3. The angle between the line and the Ox axis is smaller than 45 degrees

- When the angle is more than 45 degrees, y is always incremented, and x is not always incremented.
- When the angle is 45 degrees, both x and y coordinates are incremented.
- When the angle is less than 45 degrees, x is always incremented, and y is not always incremented

We can find this out by dividing Δy by Δx, where (x_1, y_1) and (x_2, y_2) are the endpoints of the segment, x_1, x_2, y_1, y_2 are integers and $\Delta y = y_2 - y_1$, $\Delta x = x_2 - x_1$.
If $m = \Delta y/\Delta x = 1$, the slope is 45 degrees. If $m = \Delta y/\Delta x > 1$, the slope is more than 45 degrees. If $m = \Delta y/\Delta x < 1$, the slope is less than 45 degrees.

Case 1: The slope is less than 45 degrees.

In this case, the x values will be incremented at each step. $x(k)$ is incremented every time, that is $x(k+1) = x(k) + 1$. We have to decide, using the algorithm, which of $y(k+1) = y(k)$ or $y(k+1) = y(k) + 1$ corresponds to the point closest to the line, basically if the y value is incremented or not.



What we need to do now is finding out whether d_1 is bigger than d_2 or not.

The equation of the actual line is $y = m(x(k) + 1) + c$, where $c = y(1) - (\Delta y/\Delta x)\, x(1)$ (for simplicity).
Now, we replace y with $m(x(k) + 1) + c$ in the d_1 and d_2 relations.
In the figure shown above, d_1 is $y(k) + 1 - y$ and d_2 is $y - y(k)$.

$$d_1 = y(k) + 1 - m(x(k) + 1) - c \text{ and } d_2 = m(x(k) + 1) + c - y(k).$$
$$d_2 - d_1 = [m\,(x(k) + 1) + c - y(k)] - [y(k) + 1 - m\,(x(k) + 1) - c]$$
$$= 2m(x(k) + 1) - 2y(k) + 2c - 1.$$

As it is indicated in the overview above, the algorithm seeks to work only with integer values in order to be more efficient. However, $m = \Delta y/\Delta x$, which is a float value. To get integers, we are multiplying the relation with Δx. The colour red means that those values are constant.

$$\Delta x\,(d_2 - d_1) = \Delta x[2\,\Delta y/\Delta x\,(x(k) + 1) - 2y(k) + 2c - 1]$$
$$= 2\,\Delta y \cdot x(k) - 2\,\Delta x \cdot y(k) + 2\,\Delta y + 2\,\Delta x \cdot c - \Delta x.$$

Let $P(k) = \Delta x\,(d_2 - d_1) = 2\,\Delta y \cdot x(k) - 2\,\Delta x \cdot y(k) + 2\Delta y + 2\,\Delta x \cdot c - \Delta x$. Then

$$P(k+1) - P(k) = 2\,\Delta y - 2\,\Delta x\big(y(k+1) - y(k)\big) \quad (1)$$

because the constant values are cancelled in the subtraction and $x(k+1) - x(k) = 1$.

To find out which pixel to colour, we need to find the sign of $P(k)$.

1.1 If $P(k) < 0$, $d_2 < d_1$, $y(k+1) = y(k)$ and $P(k+1) = P(k) + 2\,\Delta y$.

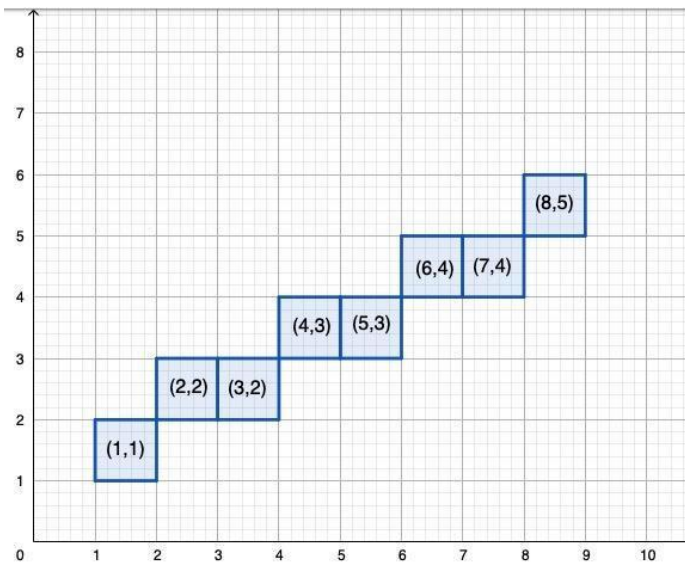
1.2 If $P(k) \geq 0$, $d_2 \geq d_1$, $y(k+1) = y(k) + 1$ and $P(k+1) = P(k) + 2\,\Delta y - 2\,\Delta x$.

Algorithm

```
Algorithm Bres (x1,x2,y1,y2)
{
x = x1;
y = y1;
dx = x2 – x1;
dy = y2 – y1;
P = 2 · dy – dx;
while(x <= x2)
{
putpixel( x, y );
x++;
if( P < 0 )
    P = P + 2 · dy;
else
{
    P = P + 2 · dy – 2 · dx;
    y++;
 }
}
}
```

Example: $x_1 = 1, y_1 = 1, x_2 = 8, y_2 = 5.$

x	y	P
1	1	1
2	2	-5
3	2	3
4	3	-3
5	3	5
6	4	-1
7	4	7
8	5	1

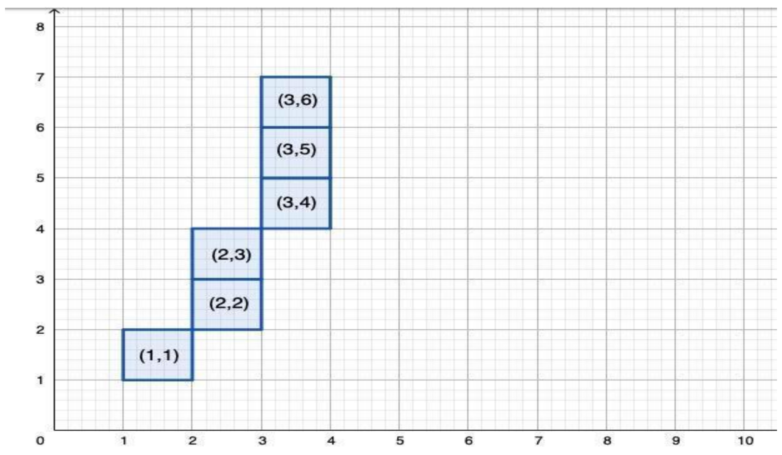


Case 2: The slope is greater than 45 degrees.

This case is identical to the first one. It is just a matter of replacing the x and y variables (if in 1st case, $\boldsymbol{P = 2\,\Delta y - \Delta x}$, in the 2nd case, $\boldsymbol{P = 2\,\Delta x - \Delta y}$)

Example: $\boldsymbol{x_1 = 1,\ y_1 = 1, x_2 = 3, y_2 = 6}$.

x	y	P
1	1	-1
2	2	3
2	3	-3
3	4	1
3	5	-5
3	6	-1

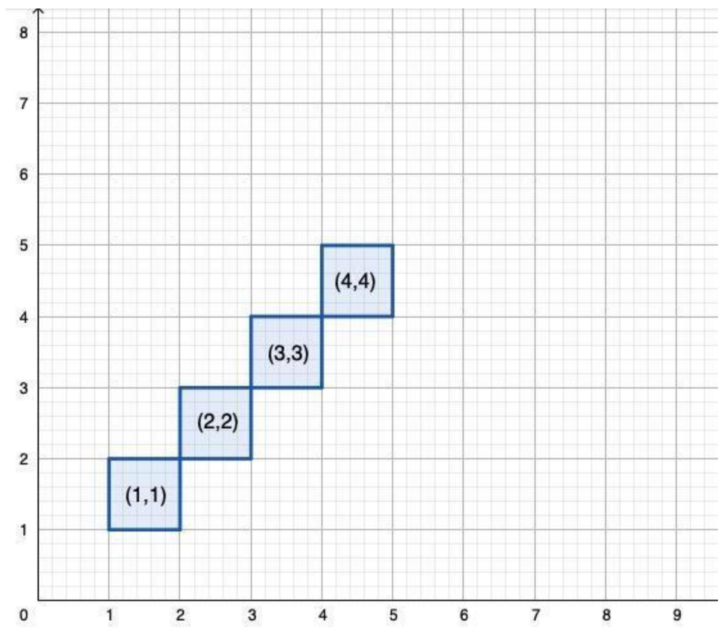


Case 3: The slope is 45 degrees.

Both $\boldsymbol{x}$ and $\boldsymbol{y}$ values are incremented every time, so we can use either of the other $\mathbf{2}$ cases, because $\boldsymbol{P}$ will always have the same value.

Example: $\boldsymbol{x_1 = 1}$, $\boldsymbol{y_1 = 1}$, $\boldsymbol{x_2 = 4}$, $\boldsymbol{y_2 = 4}$.

x	y	P
1	1	3
2	2	3
3	3	3
4	4	3



2. Circle drawing algorithm

Overview

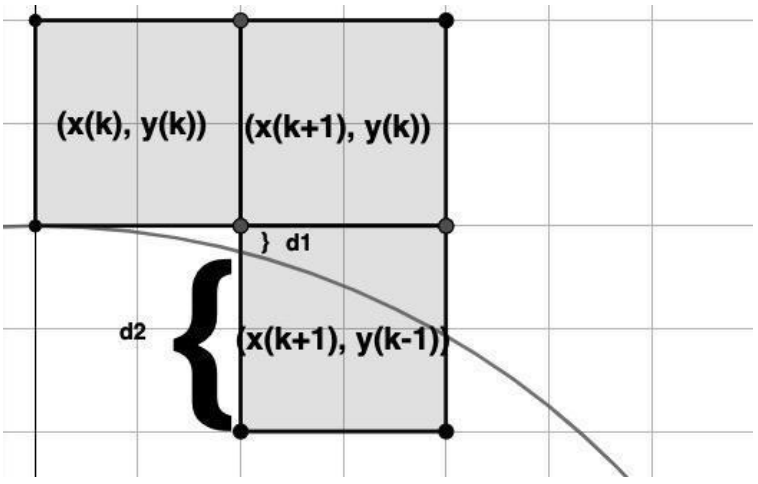
We will use Bresenham's circle drawing algorithm. We will divide the circle in four quadrants, each of 90 degrees and further in 8 octants, each of 45 degrees. The algorithm is almost the same for all of the sectors.

Conventions:

- Let initial x be $x(0)$ and initial y be $y(0)$.
- First quadrant - top right;
- Second quadrant - top left;
- Third quadrant - bottom left;
- Fourth quadrant - bottom right;
- The line is drawn clockwise.
- If a point is inside the circle, its distance from the circle will be considered negative and if it is outside, its distance from the circle will be considered positive.

Analysis

The analysis is based on an example which pictures a circle of equation $x^2 + y^2 = R^2$.
We will have a closer look at the first octant, which is the top right one, and has x>=0 and y>=x.



In this case, x will always be incremented, but we can't tell the same thing about y. So, we have to decide whether to increment y or not.

We will have to choose between the $(x+1, y)$ pixel and the $(x+1, y-1)$ pixel. We could do that by finding the distances between each of the two pixels and the circle.

Let $d(x,y) = x^2 + y^2 - R^2$. This is a parameter that we shall use to find the sign for the next incrementation (2).

So, d_1 will be the parameter used for the $(x(k)+1, y(k))$ pixel with respect to the circle and d_2 the parameter used for the $(x(k)+1, y(k)-1)$ pixel with respect to the circle, so we can write them as

$$d_1 = (x(k)+1)^2 + y(k)^2 - R^2$$

$$d_2 = (x(k)+1)^2 + (y(k)-1)^2 - R^2.$$

We now need to decide which pixel is closer to the circle and for that we will need a decision variable, which we will name $P(k)$.

$$\begin{aligned} P(k) &= d_1 + d_2 \\ &= 2(x(k)+1)^2 + y(k)^2 + (y(k)-1)^2 - 2R^2 \end{aligned}$$

$$P(k+1) = 2(x(k)+2)^2 + y(k+1)^2 + (y(k+1)-1)^2 - 2R^2$$

We need to write $P(k+1)$ in relation to $P(k)$:

$$\begin{aligned} P(k+1) = P(k) &+ 2(2\,x(k)+3) + (y(k+1)+y(k))(y(k+1)-y(k)) \\ &+ (y(k+1)-1+y(k)-1)\,(y(k+1)-1-y(k)+1). \end{aligned}$$

d_1 will be considered positive, as the $(x(k)+1, y(k))$ pixel is outside the circle and d_2 negative, because the $(x(k)+1, y(k)-1)$ pixel is inside the circle.

- If $P(k) \leq 0$, it means that $|d_2|$ is greater than d_1, and the next pixel will be $(x(k)+1, y(k))$:

  $x(k\ +\ 1) = x(k) + 1$ and $y(k+1) = y(k)$.

We now need to replace these values. That gives

$$P(k+1) = P(k) + 2(2x(k)+3) = P(k) + 4x(k) + 6.$$

- If $P(k) > 0$, it means that d_1 is greater than $|d_2|$, and the next pixel will be $(x(k)+1, y(k)-1)$:

  $x(k+1) = x(k) + 1;\ \ y(k+1) = y(k) - 1.$

We now need to replace these values. That gives

$$P(k+1) = P(k) + 4(x(k) - y(k)) + 10.$$

The first pixel has the coordinates $(0, R)$, so $P(0) = 3 - 2R$.

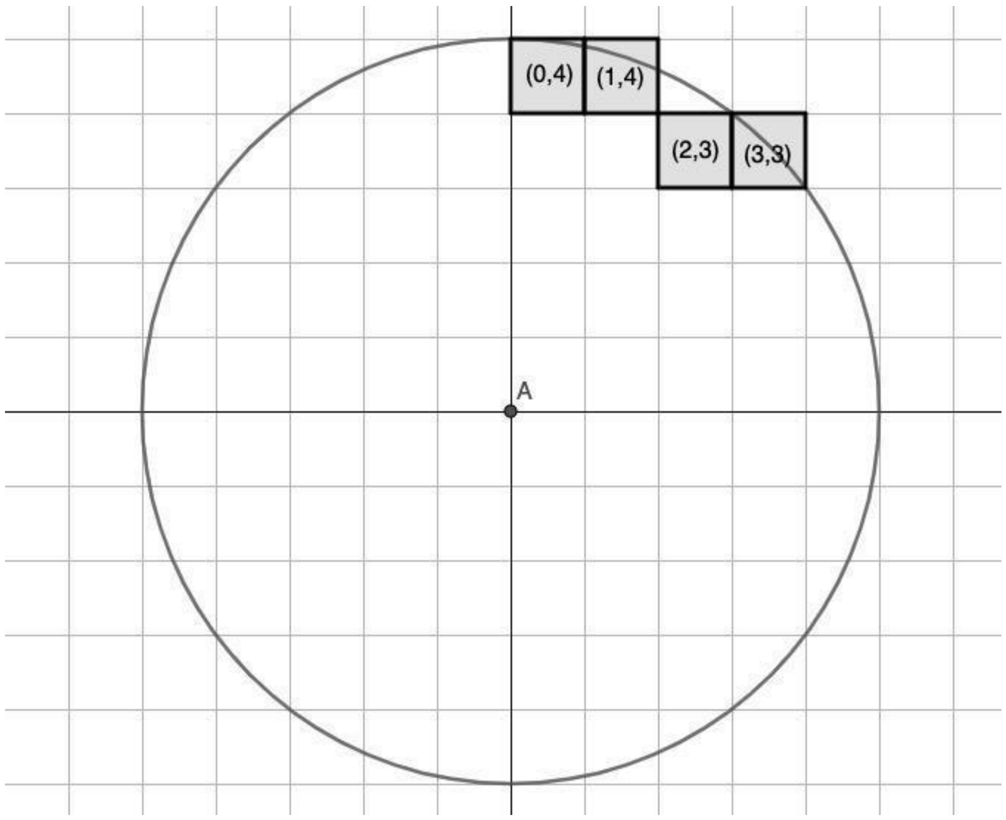
Algorithm

```
{
x = 0;
y = r;
P = 3 – 2 · r;
while(y>=x)
{
putpixel( x, y );
if (P <= 0)
        P = P + 4 · x + 6
else
{
        P = P + 4 · x – 4 · y + 10
        y--
}
x++
}
}
```

Example: Let’s take a circle with radius 4 and the centre at (0,0).

This is the table for the first octant:

$x(k)$	$y(k)$	$P(k)$
0	4	-5
1	4	1
2	3	-1
3	3	13

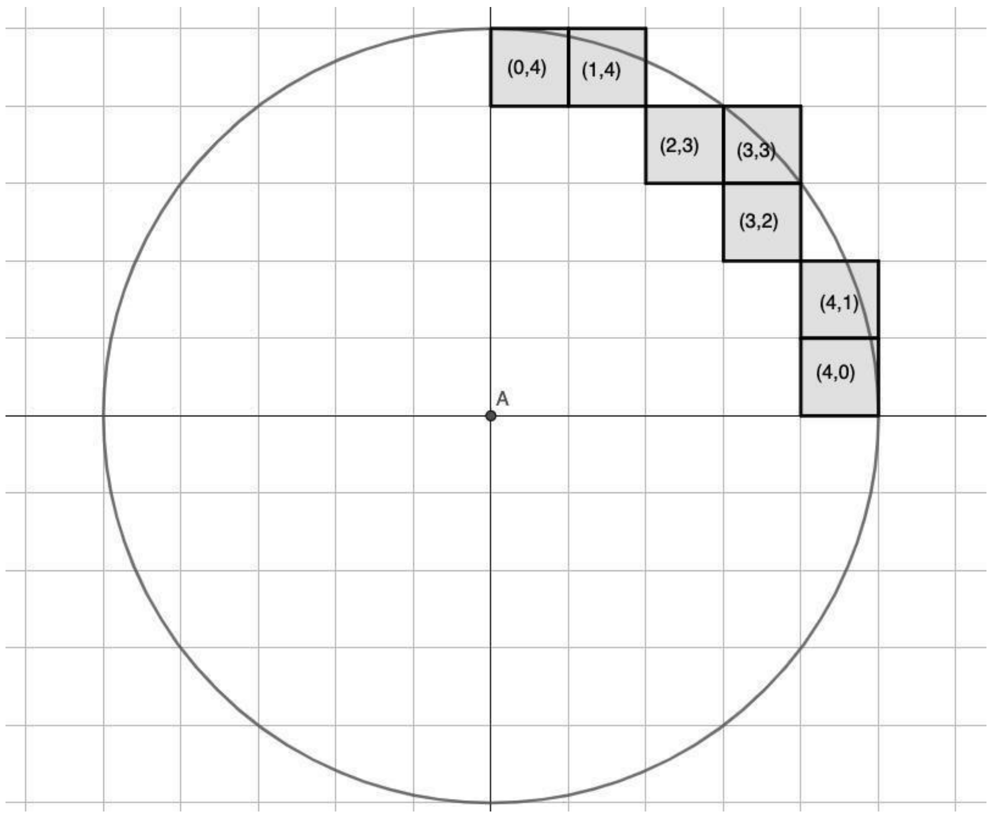


Now, we will calculate the coordinates of the second octant by swapping the x and y coordinates.

First octant	Second octant
$(0, 4)$	$(3, 3)$
$(1, 4)$	$(3, 2)$
$(2, 3)$	$(4, 1)$
$(3, 3)$	$(4, 0)$

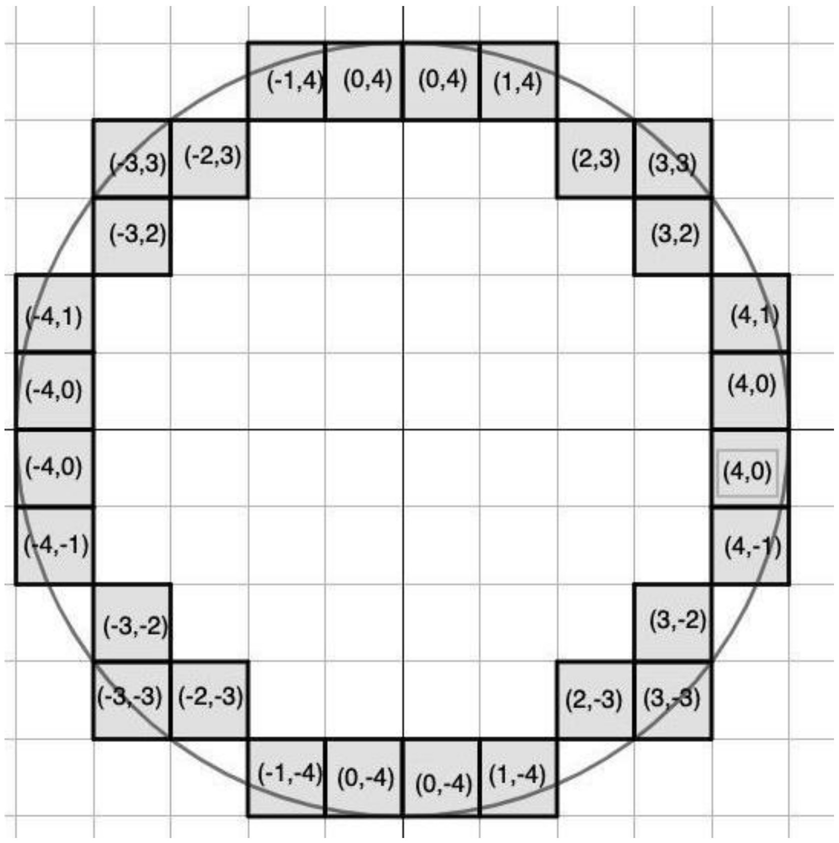
So, the first quadrant (90 degrees), will be:

First quadrant
$(0, 4)$
$(1, 4)$
$(2, 3)$
$(3, 3)$
$(3, 2)$
$(4, 1)$
$(4, 0)$



Now, we can calculate all of the circle’s points, using the first quadrant:

Quadrant 1 (x, y)	**Quadrant 2** $(-x, y)$	**Quadrant 3** $(-x, -y)$	**Quadrant 4** $(x, -y)$
$(0, 4)$	$(0, 4)$	$(0, -4)$	$(0, -4)$
$(1, 4)$	$(-1, 4)$	$(-1, -4)$	$(1, -4)$
$(2, 3)$	$(-2, 3)$	$(-2, -3)$	$(2, -3)$
(3, 3)	(−3, 3)	(−3, −3)	(3, −3)
(3, 2)	(−3, 2)	(−3, −2)	(3, −2)
(4, 1)	(−4, 1)	(−4, −1)	(4, −1)
(4, 0)	(−4, 0)	(−4, −0)	(4, 0)



Editing Notes

(1) This simple expression of $P(k+1) - P(k)$ will allow a quick calculation of $P(k)$, from one term to the next. The initial value $2\Delta y - \Delta x$ is given in the algorithm below by the line *P=2·dy–dx.*

(2) The actual distance from the point (x, y) to the circle is $\left|\sqrt{x^2+y^2} - R\right|$, but $d(x,y) = x^2 + y^2 - R^2$ can be used to compare distances without significant errors.
Note also that d_1 and d_2 in the figure above are also different from the distances to the circle and from the parameters calculated here.