

## **EXPERIMENT NO 5**

~ ARYA RAUL

**COMPS 20**

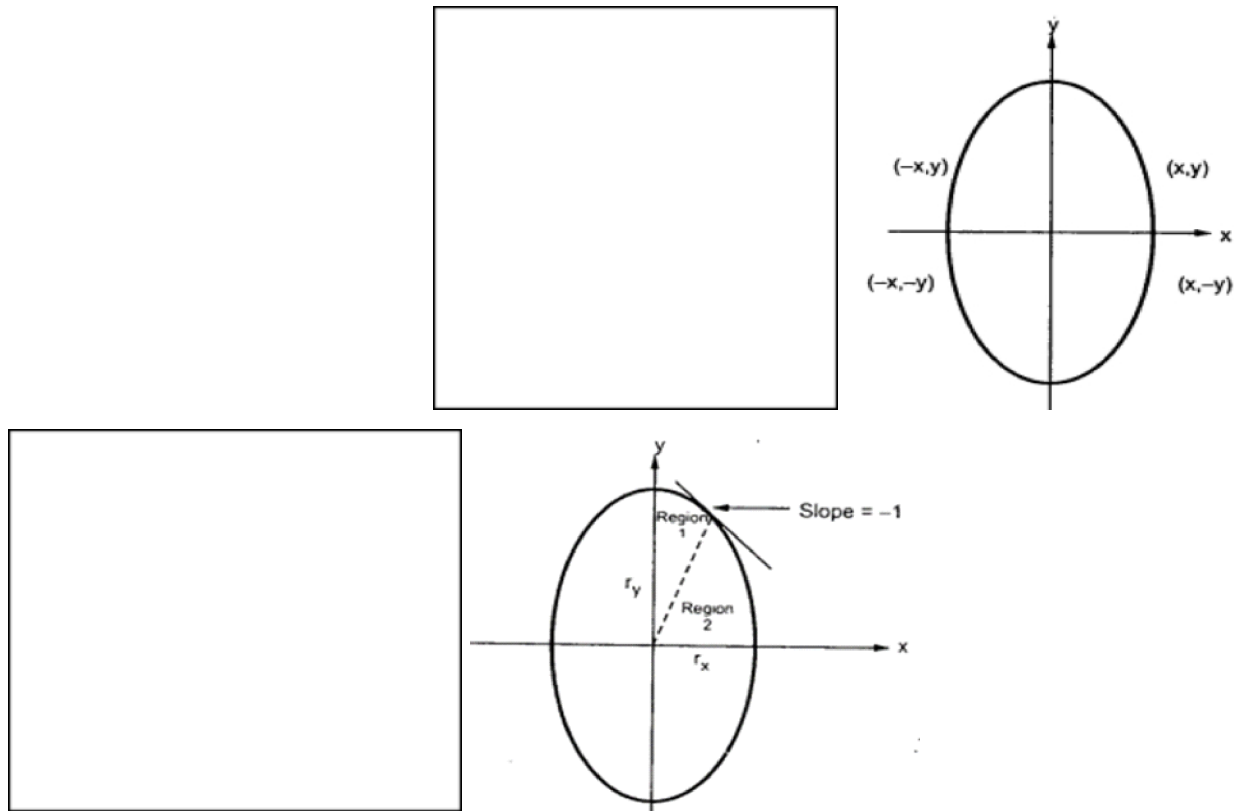
**Aim:** To implement midpoint Ellipse algorithm

### **Objective:**

Draw the ellipse using Mid-point Ellipse algorithm in computer graphics. Midpoint ellipse algorithm plots (finds) points of an ellipse on the first quadrant by dividing the quadrant into two regions.

### **Theory:**

Midpoint ellipse algorithm uses four way symmetry of the ellipse to generate it. Figure shows the 4-way symmetry of the ellipse.



Here the quadrant of the ellipse is divided into two regions as shown in the fig. Fig. shows the division of first quadrant according to the slope of an ellipse with  $r_x < r_y$ . As ellipse is drawn from  $90^\circ$  to  $0^\circ$ ,  $x$  moves in positive direction and  $y$  moves in negative direction and ellipse passes through two regions 1 and 2.

The equation of ellipse with center at  $(x_c, y_c)$  is given as -

$$\left[ \frac{(x - x_c)}{r_x} \right]^2 + \left[ \frac{(y - y_c)}{r_y} \right]^2 = 1$$

Therefor, the equation of ellipse with center at origin is given as -

$$\left[ \frac{x}{r_x} \right]^2 + \left[ \frac{y}{r_y} \right]^2 = 1$$

$$\text{i.e. } x^2 r_y^2 + y^2 r_x^2 = r_x^2 r_y^2$$

$$\text{Let, } f_{\text{ellipse}}(x, y) = x^2 r_y^2 + y^2 r_x^2 - r_x^2 r_y^2$$

### Algorithm:

1) Read  $r_x$  and  $r_y$ .

2) Initialise starting point as

$$x = 0$$

$$y = r_y$$

3) Calculate the initial value of decision parameter in region 1 as

$$d1 = ry^2 - rx^2 ry + (rx^2 / 4)$$

4) Initialise dx and dy as

$$dx = 2 ry^2 x$$

$$dy = 2 rx^2 y$$

5) do

```
{  
  plot (x, y)  
  if (d1 < 0)  
    {  
      x = x+ 1  
      y = y  
      dx = dx + 2 ry2  
      d1 = d1 + dx + ry2  
    }  
  else  
    {  
      x = x+ 1  
      y = y - 1  
      dx = dx + 2 ry2  
      dy = dy - 2 rx2  
      d1 = d1 + dx - dy + ry2
```

}

}

while (dx < dy)

6) Calculate the initial value of decision parameter in region 2 as

$$d2 = ry^2 [x + (1/2)]^2 + rx^2 (y - 1)^2 + rx^2 ry^2$$

7) do

{

plot (x, y)

if (d2 > 0)

{

x = x

y = y - 1

dy = dy + 2 rx<sup>2</sup>

d2 = d2 - dy + rx<sup>2</sup>

}

else

{

x = x+ 1

y = y - 1

dy = dy - 2 rx<sup>2</sup>

dx = dx + 2 ry<sup>2</sup>

d2 = d2 + dx - dy + rx<sup>2</sup>

}

}

while (y > 0)

8) Determine the symmetry points in other three quadrants.

9) Stop.

**Program:**

```
#include <graphics.h>
```

```
#include <conio.h>
```

```
#include <math.h>
```

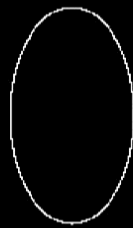
```
void drawEllipse(int xc, int yc, int rx, int ry) {  
    float x, y, angle;  
    for (angle = 0; angle <= 360; angle += 0.01) {  
        x = xc + rx * cos(angle);  
        y = yc + ry * sin(angle);  
        putpixel(x, y, WHITE);  
    }  
}
```

```
int main() {  
    int gd = DETECT, gm, xc, yc, rx, ry;  
    initgraph(&gd, &gm, "C:\\Turboc3\\BGI");  
  
    // int xc, yc, rx, ry;  
  
    // Input center and radii of the ellipse  
    printf("Enter the center of the ellipse (x y): ");
```

```
scanf("%d %d", &xc, &yc);  
printf("Enter the major and minor radii (rx ry): ");  
scanf("%d %d", &rx, &ry);  
  
// Draw the ellipse  
drawEllipse(xc, yc, rx, ry);  
getch();  
closegraph();  
return 0;
```

**Output:**

Enter the center of the ellipse (x y): 320 240  
Enter the major and minor radii (rx ry): 40 70



## Conclusion: Comment on

**Slow or fast:** This algorithm is relatively fast and is commonly used for drawing ellipses. It is an incremental algorithm that calculates points on the ellipse's boundary one at a time, making it efficient.

### Difference with circle:

- **Shape:** The primary difference is, of course, the shape being drawn. A circle is a special case of an ellipse where both the major and minor axes are equal, resulting in a perfectly round shape. Ellipses, on the other hand, can have different major and minor axes, resulting in elongated or compressed shapes.
- **Algorithm Variations:** While there are specialized algorithms for drawing circles (e.g., Bresenham's Circle Drawing Algorithm), ellipses require more complex algorithms due to their varying axes. The Midpoint and Bresenham's Circle algorithms can be seen as simplified versions of ellipse algorithms where both axes are equal.
- **Parameters:** When drawing a circle, you typically only need to specify the center coordinates and the radius. For ellipses, you need to specify the center coordinates, the lengths of the major and minor axes, and sometimes the orientation angle.
- **Complexity:** Ellipse drawing algorithms are generally more complex than circle drawing algorithms due to the need to handle varying axes. This complexity can lead to slightly slower performance compared to circle drawing, especially for highly elongated or compressed ellipses.

### Importance of object:

- **Geometric Modeling:** Objects like ellipses are essential for representing various real-world shapes and structures. Ellipses can model a wide range of objects such as planets, wheels, eyes, and more. They provide a flexible means of creating and rendering complex shapes.



- **Visual Elements:** Ellipses, as objects, are fundamental visual elements in graphic design, computer graphics, and user interfaces. They can be used to create aesthetically pleasing designs, icons, logos, and user interface components.
- **Data Visualization:** In data visualization, ellipses can be used to represent statistical information, such as confidence intervals or clusters of data points. This is particularly important in fields like statistics, data analysis, and scientific research.
- **Animation:** Objects like ellipses are used in animation to create moving and dynamic visuals. Ellipses can be used as basic building blocks for creating animated characters, objects, and effects.
- **Image Processing:** In image processing, ellipses can be used to define regions of interest, segment objects from backgrounds, and perform tasks like object tracking, object recognition, and image analysis.
- **Computer-Aided Design (CAD):** In CAD software, ellipses are essential for creating accurate and precise designs. They are used to represent curves, arcs, and other curved geometries commonly found in engineering and architectural drawings.
- **User Interaction:** Ellipses can serve as interactive elements in user interfaces. They can be used for buttons, sliders, and other UI components, allowing users to interact with software applications.
- **Mathematical Abstraction:** Ellipses are a mathematical abstraction that can be used to describe real-world objects and phenomena. They have mathematical properties that make them useful in various analytical and computational tasks.
- **Artistic Expression:** Ellipses can be used as artistic elements in digital art, illustrations, and creative design. Artists and designers often use ellipses to create unique and visually appealing compositions.