Experiment No. 4
Topic: To implement midpoint Ellipse algorithm
Name: Vinith Shetty
Roll Number: 55
Date of Performance:
Date of Submission:

Experiment No. 4

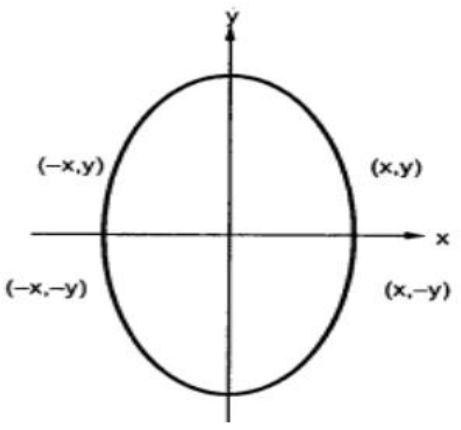
Aim-To implement midpoint Ellipse algorithm

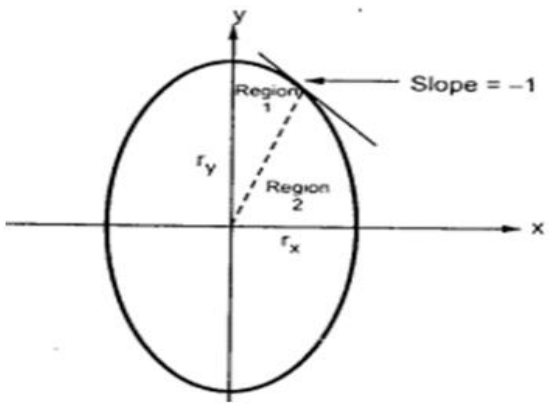
Objective:

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 rx &lt; ry. As ellipse is drawn

from 90 0 to 0 0 , 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 (xc, yc) is given as -

[(x – xc) / rx] 2 + [(y – yc) / ry] 2 = 1

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

[x / rx] 2 + [y / ry] 2 = 1

i.e. x 2 ry 2 + y 2 rx 2 = rx 2 ry 2

Let, f ellipse (x, y) = x2 ry2 + y2 rx2 - rx2 ry2

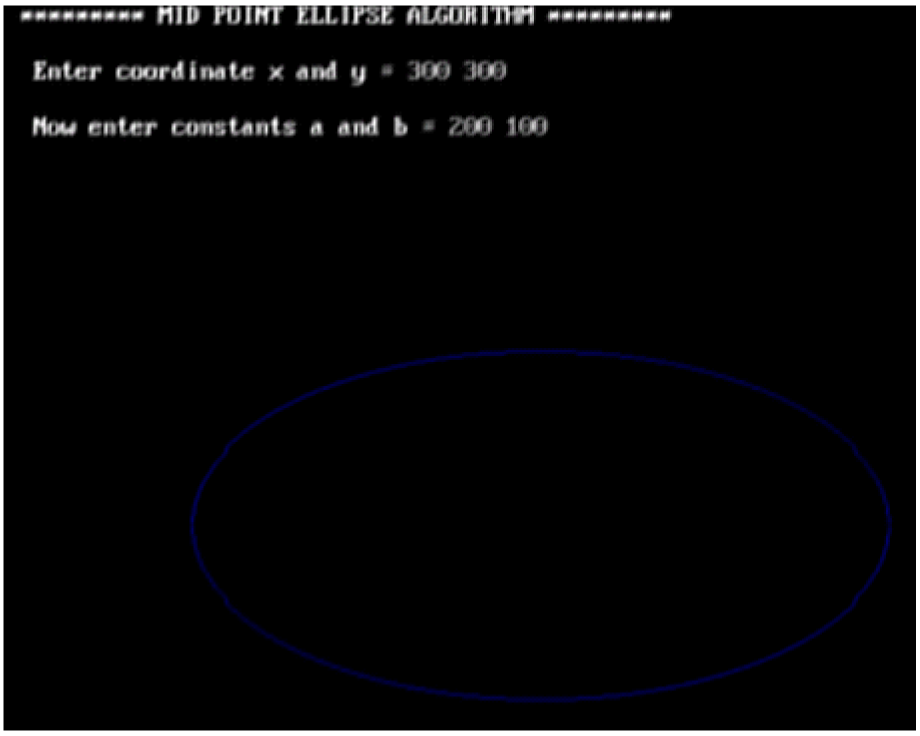
Algorithm:

Program

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void pixel(int x,int y,int xc,int yc)
{
        putpixel(x+xc,y+yc,BLUE);
        putpixel(x+xc,-y+yc,BLUE);
        putpixel(-x+xc,y+yc,BLUE);
        putpixel(-x+xc,-y+yc,BLUE);
        putpixel(y+xc,x+yc,BLUE);
        putpixel(y+xc,-x+yc,BLUE);
        putpixel(-y+xc,x+yc,BLUE);
        putpixel(-y+xc,-x+yc,BLUE);
}
main()
{
        int gd=DETECT,gm=0,r,xc,yc,x,y;
        float p;
        //detectgraph(&gd,&gm);
        initgraph(&gd,&gm," ");
        printf("\n ********* MID POINT ELLIPSE ALGORITHM ********* ")
        printf("\n Enter the radius of the circle:");
        scanf("%d",&r);
        printf("\n Enter the center of the circle:");
        scanf("%d %d",&xc,&yc);
        y=r;
        x=0;
        p=(5/4)-r;
```

```
	while(x<y)
	{
		if(p<0)
		{
			x=x+1;
			y=y;
			p=p+2*x+3;
		}
		else
		{
			x=x+1;
			y=y-1;
			p=p+2*x-2*y+5;
		}
		pixel(x,y,xc,yc);
	}
	getch();
	closegraph();
}
```

Output:



```
********* MID POINT ELLIPSE ALGORITHM *********

 Enter coordinate x and y = 300 300

 Now enter constants a and b = 200 100
```

Conclusion: Comment on

1. Slow or fast

2. Difference with circle

3. Importance of object

Speed: The Midpoint Ellipse Algorithm, like its circular counterpart for drawing circles, is also considered a relatively efficient and fast method for drawing ellipses in computer graphics. It uses integer arithmetic and incremental calculations to determine which pixels to plot on the ellipse's perimeter.

Differences with circles: The midpoint ellipse algorithm differs from the midpoint circle algorithm in terms of the decision parameters and the way points are plotted. While both algorithms utilize the concept of symmetry and incremental calculations, the parameters and equations for ellipses are different from those for circles due to the variation in the geometry of these two shapes. These algorithms use integer arithmetic and incremental calculations to efficiently plot points on the curves they are designed for.

Importance of the object: The importance of an object in Midpoint Ellipse Algorithm often lies in its context and the particular field of study or application being considered. Objects serve as building blocks, subjects, entities, or concepts that contribute to various aspects of our understanding, development, and interaction within different disciplines and industries.