**Practical No 4**

**Implementation of Midpoint Ellipse Drawing Algorithm.**

**Aim: Write a program to implement a Midpoint Ellipse Drawing Algorithm.**

**Theory:**

An ellipse is defined as the set of points such that the sum of the distances from two fixed positions (foci) is the same for all points. Our approach here is similar to that used in displaying a raster circle. Given parameters rx, ry and (xc, yc), we determine points (x, y) for an ellipse in standard position cantered on the origin, and then we shift the points so the ellipse is cantered at (xc, yc). The midpoint ellipse method is applied throughout the first quadrant in two parts. Figure shows the division of the first quadrant according to the slope, of an ellipse with rx < ry. We process this quadrant by taking unit steps in the x direction where the slope of the curve has a magnitude less than 1, and taking unit steps in the y direction where the slope has a magnitude greater than 1. Regions I and 2, can be processed in various ways. We can start at position (0. r) and step clockwise along the elliptical path in the first quadrant, shifting from unit steps in x to unit steps in y when the slope becomes less than -1. We define an ellipse function fellipse (x, y) =ry2x2+rx2y2-rx2ry2. Which act as decision parameter for mid-point ellipse generation algorithm. This function helps us to decide next pixel to colour.

**Algorithm:**

1. Input rx, ry and ellipse centre (xc, yc) and obtain the first point on an ellipse centred on the origin as (x0, y0) = (0, r).
2. Calculate the initial parameter in region 1 as p10=ry2-rx2(ry-1/4).
3. At each xi position, starting at i=0,

* If p1i<0 then the next point along the ellipse centred on (0,0) is (xi+1, yi) and p1i+1=p1i+2ry2x+ry2.
* Otherwise, the point is (xi+1, yi-1) and p1i+1=p1i+2ry2x-2rx2y+ry2.
* Continue until 2ry2x>=2rx2y.

1. (x0, y0) is the last position calculated in region 1. Calculate the initial parameter in region 2 as p20=ry2(x+1/2)2+rx2(y-1)2-rx2ry2.
2. At each yi position, starting at i=0

* If p2i<0 then the next point along the ellipse centred on (0,0) is (xi, yi-1) and p2i+1=p2i -2rx2y+rx2
* Otherwise, the next point is (xi+1, yi-1) and p2i+1=p2i+2ry2x-2rx2y+rx2
* Use the same incremental calculation as in region 1. Continue until y=0

1. For both regions determine symmetry points in the other three quadrants.
2. Move each calculated pixel position (x, y) onto the elliptical path centred on (xc, yc) and x=x+xc, y=y+yc.

**Conclusion: We have implemented Midpoint Ellipse Drawing Algorithm.**

**Code:**

#include<iostream.h>

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void draw(float xc,float yc,float rx,float ry)

{

float p1,p2,x,y;

x=0;y=ry;

p1=ry\*ry-rx\*rx\*(ry-1/4.0);

while(2\*ry\*ry\*x<2\*rx\*rx\*y)

{

x++;

if(p1<0)

{

p1+=2\*ry\*ry\*(x)+ry\*ry;

}

else

{

y--;

p1+=2\*ry\*ry\*x-2\*rx\*rx\*y+ry\*ry;

}

putpixel(xc+x,yc+y,RED);

putpixel(xc-x,yc+y,RED);

putpixel(xc+x,yc-y,RED);

putpixel(xc-x,yc-y,RED);

}

p2=ry\*ry\*(x+1/2.0)\*(x+1/2.0)+rx\*rx\*(y-1.0)\*(y-1.0)-rx\*rx\*ry\*ry;

while(y!=0)

{

y--;

if(p2<0)

{

p2+=-2\*rx\*rx\*y+rx\*rx;

}

else{

x++;

p2+=2\*ry\*ry\*x-2\*rx\*rx\*y+rx\*rx;

}

putpixel(xc+x,yc+y,RED);

putpixel(xc-x,yc+y,RED);

putpixel(xc+x,yc-y,RED);

putpixel(xc-x,yc-y,RED);

}

}

float main()

{

int gd=DETECT,gm;

float rx,ry,xc,yc;

initgraph(&gd,&gm,"C:/TURBOC3/BGI");

cout<<"Midpoint Ellipse Drawing Alogorithm \n";

cout<<"Enter center axis: ";

cin>>xc>>yc;

cout<<"Enter major axis: ";

cin>>rx;

cout<<"Enter minor axis: ";

cin>>ry;

draw(xc,yc,rx,ry);

getch();

closegraph();

return 0;

}

**Output:**

