Midpoint Ellipse Drawing Algorithm

1 Objective

To draw an ellipse using the midpoint ellipse drawing algorithm.

2 Theory

2.1 Midpoint Ellipse Drawing Algorithm

The midpoint ellipse algorithm is an algorithm used to determine the points needed for drawing an ellipse. It is a variant of Bresenham's circle algorithm where the calculation of d is altered to accommodate for the ellipses' eccentricity. It makes use of the fact that the sum of the slopes of any point on an ellipse to its foci is always constant. The algorithm exploits this property to determine the next point to plot at each iteration. It is similar to the midpoint circle algorithm. The only difference is that the decision parameter is calculated using the midpoint ellipse formula. Since it can be used to draw circles as well as ellipses, it can be said to be a more generalized form of the circle drawing algorithm.

3 Algorithm

1. Input the center of the ellipse, radius in x and y direction.
2. Calculate the initial value of the decision parameter in region 1 as $p_1 = b^2 - a^2 b + \frac{1}{4}a^2$.
3. At each x_k position starting at k = 0, perform the following test:
   (a) If $p_k < 0$, the next point along the ellipse at x_{k+1} is chosen to be $(x_k + 1, y_k)$ and $p_{k+1} = p_k + 2b^2 x_{k+1} + b^2$.
   (b) If $p_k \geq 0$, the next point along the ellipse at x_{k+1} is chosen to be $(x_k + 1, y_k - 1)$ and $p_{k+1} = p_k + 2b^2 x_{k+1} - 2a^2 y_{k+1} + b^2$.
4. Repeat step 3 until $2b^2 x \geq 2a^2 y$.
5. At each yk position starting at k = 0, perform the following test:
   (a) If $p_k > 0$, the next point along the ellipse at y_{k+1} is chosen to be $(x_k, y_k - 1)$ and $p_{k+1} = p_k - 2a^2 y_{k+1} + a^2$.
   (b) If $p_k \leq 0$, the next point along the ellipse at y_{k+1} is chosen to be $(x_k + 1, y_k - 1)$ and $p_{k+1} = p_k + 2b^2 x_{k+1} - 2a^2 y_{k+1} + a^2$.
6. Repeat step 5 until $y_{k+1} = 0$.

4 Source Code

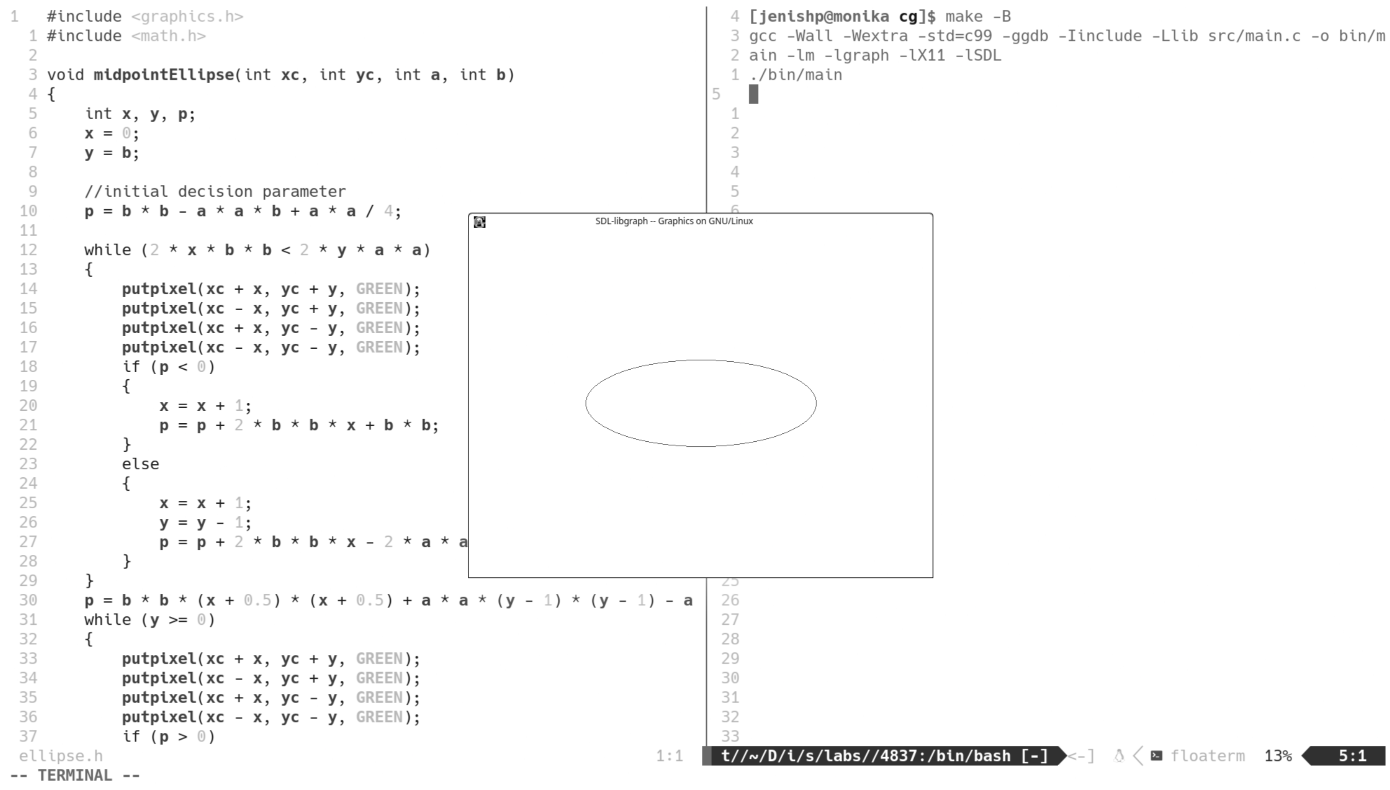
```c
#include <graphics.h>
#include <math.h>

void midpointEllipse(int xc, int yc, int a, int b)
{
    int x, y, p;
    x = 0;
    y = b;

    //initial decision parameter
    p = b * b - a * a * b + a * a / 4;

    while (2 * x * b * b < 2 * y * a * a)
    {
        putpixel(xc + x, yc + y, GREEN);
        putpixel(xc - x, yc + y, GREEN);
        putpixel(xc + x, yc - y, GREEN);
        putpixel(xc - x, yc - y, GREEN);
        if (p < 0)
        {
            x = x + 1;
            p = p + 2 * b * b * x + b * b;
        }
        else
        {
            x = x + 1;
            y = y - 1;
            p = p + 2 * b * b * x - 2 * a * a * y + b * b;
        }
    }
    p = b * b * (x + 0.5) * (x + 0.5) + a * a * (y - 1) * (y - 1) - a * a * b * b;
    while (y >= 0)
    {
        putpixel(xc + x, yc + y, GREEN);
        putpixel(xc - x, yc + y, GREEN);
        putpixel(xc + x, yc - y, GREEN);
        putpixel(xc - x, yc - y, GREEN);
        if (p > 0)
        {
            y = y - 1;
            p = p - 2 * a * a * y + a * a;
        }
        else
        {
            y = y - 1;
            x = x + 1;
            p = p + 2 * b * b * x - 2 * a * a * y + a * a;
        }
    }
}
```

5 Output



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

void midpointEllipse(int xc, int yc, int a, int b)
{
    int x, y, p;
    x = 0;
    y = b;

    //initial decision parameter
    p = b * b - a * a * b + a * a / 4;

    while (2 * x * b * b < 2 * y * a * a)
    {
        putpixel(xc + x, yc + y, GREEN);
        putpixel(xc - x, yc + y, GREEN);
        putpixel(xc + x, yc - y, GREEN);
        putpixel(xc - x, yc - y, GREEN);
        if (p < 0)
        {
            x = x + 1;
            p = p + 2 * b * b * x + b * b;
        }
        else
        {
            x = x + 1;
            y = y - 1;
            p = p + 2 * b * b * x - 2 * a * a
        }
    }
    p = b * b * (x + 0.5) * (x + 0.5) + a * a * (y - 1) * (y - 1) - a
    while (y >= 0)
    {
        putpixel(xc + x, yc + y, GREEN);
        putpixel(xc - x, yc + y, GREEN);
        putpixel(xc + x, yc - y, GREEN);
        putpixel(xc - x, yc - y, GREEN);
        if (p > 0)
ellipse.h                                                    1:1
-- TERMINAL --
```

```
[jenishp@monika cg]$ make -B
gcc -Wall -Wextra -std=c99 -ggdb -Iinclude -Llib src/main.c -o bin/m
ain -lm -lgraph -lX11 -lSDL
./bin/main

t//~/D/i/s/labs//4837:/bin/bash [-] <-]   floaterm  13%   5:1
```

Figure 1:

6 Conclusion

We have successfully drawn an ellipse using midpoint ellipse drawing algorithm. Ellipses of various eccentricities can be drawn using this algorithm. Drawing of circles is a special case of this algorithm and was achieved to demonstrate that the algorithm is a generalized version of the midpoint circle drawing algorithm.