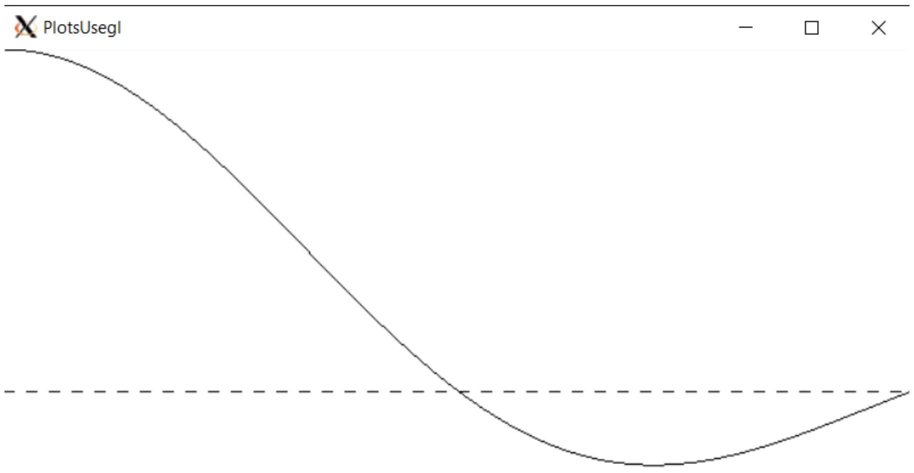
Lab 3 CSE 4200 – Spencer Wallace

Summary/Analysis:

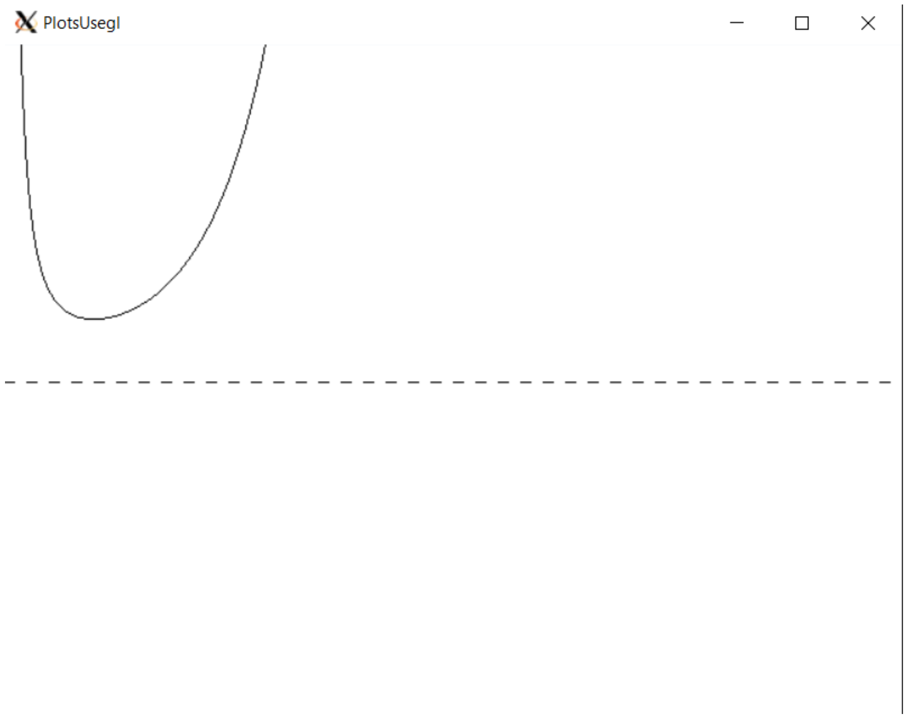
All parts were completed successfully. When comparing my results with results from a graphing calculator, they appear to be correct. Because of this, I am giving myself 20/20 points for this lab.

Part 1) f(x) = sin (x) / x , $0 < x \leq 2\pi$



For this part of the lab I started with setting up a few global variables to make using different functions easier. The variables include PI, x2, and yOffset. PI was simply used for the constant PI, x2 was used as the domain for the function as well as the window width, and yOffset was used as an offset for window height. For this problem, x2 was set to 2*PI, and yOffset was set to one, making the domain from 0 to 2*PI and the range from -1 to 1. A dotted line was also added for the x-axis. Last, the function used in "f' was changed to y = sin(x)/x.

Part 2) $f(x) = e^{x} / x^{2}$, $0 < x \le 20$



With the global variables established from the previous problem, switching to display this function was much easier. For this function x2 was set to 20, and yOffset was set to 10. This made the domain scale from 0 to 20, and the range from -10 to 10. The function used in “f” was changed to y = exp(x)/pow(x,2).

****Code****

```
#include <GL/gl.h>

#include <GL/glut.h>

#include <stdio.h>

#include <math.h>


using namespace std;


const int VWIDTH = 640;

const int VHEIGHT = 480;

const float PI = 3.1415926;

//const double x2 = 20;

const double x2 = 2.0*PI;

const int yOffset = 1;

void init(void)

{

  glClearColor(1.0,1.0,1.0,0.0);

  glMatrixMode(GL_PROJECTION);

  gluOrtho2D(0.0, x2, 0 - yOffset, yOffset);

}


double f ( double x )

{

  //double y = exp(x)/pow(x,2);

  double y = (sin(x))/x ;

  //double y = exp ( -fabs ( x ) ) * cos ( 2 * PI * x );


  return y;
```

```
}

float cx, cy;

void moveTo ( float x, float y )
{
  cx = x;
  cy = y;
}

void lineTo ( float x, float y )
{
  glBegin ( GL_LINES );
    glVertex2f ( cx, cy );
    glVertex2f ( x, y );
  glEnd();
  cx = x;
  cy = y;
}

void display()
{
  glClear(GL_COLOR_BUFFER_BIT);
  glColor3f(0, 0, 0);
  double x, y;
  x = 0.0;                              //initial position
  y = f ( 0.0 );
  moveTo ( x, y );
  for ( x = 0; x < x2; x += 0.005 ) {
    y =  f ( x );
    lineTo ( x, y );
  }
  glEnable(GL_LINE_STIPPLE);
  glLineStipple(1, 0x00FF);
  glBegin(GL_LINES); glVertex2f(0,0); glVertex2f(x2,0); glEnd();

  glFlush();
}

int main(int argc,char **argv){
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowPosition(200,300);
    glutInitWindowSize(VWIDTH, VHEIGHT);
    glutCreateWindow("PlotsUsegl");
    init();
    glutDisplayFunc( display );
    glutMainLoop();
    return 0;
}
```