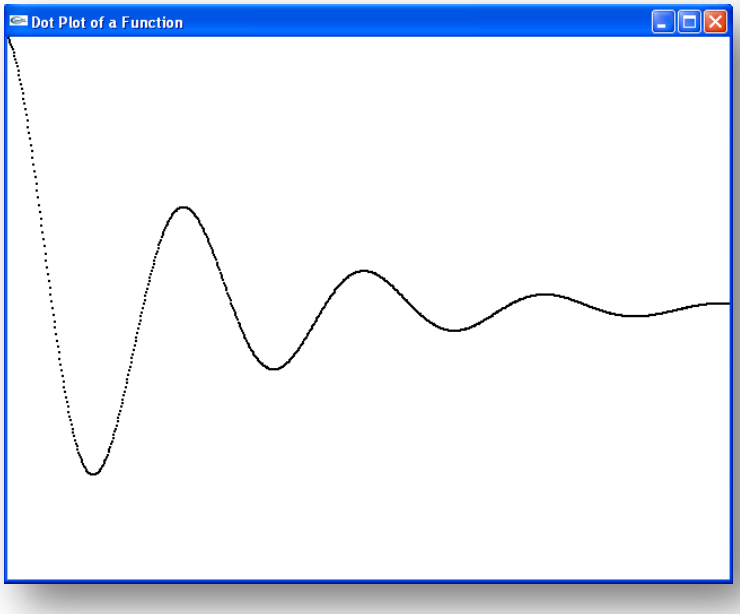
CSC 305 **Computer Graphics**

Lab Exercise

This example is to plot the mathematical function.
Suppose we wish to learn the behavior of some mathematical function f(x) as x varies, for example, $f(x) = e^{1-x1}\cos(2\pi x)$
Vary for values of x between 0 and 4? A quick plot of f(x) vs. x reveals patterns and trends of information to the analyst.
To plot this function, simply sample it as a collection of closely spaced x-values and plot a dot at each coordinate pair (x_i, $f(x_i)$). Choosing some suitable increment say 0.005, between consecutive x-values.

Sample Output



Source Code

```
// Example_2.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"


#include <windows.h> // use proper includes for your system
#include <math.h>
#include <gl/Gl.h>
#include <gl/glut.h>
const int screenWidth = 640;     // width of screen window in pixels
const int screenHeight = 480;    // height of screen window in pixels
GLdouble A, B, C, D;  // values used for scaling and shifting
//<<<<<<<<<<<<<<<<<<<<<<< myInit >>>>>>>>>>>>>>>>>>>
 void myInit(void)
 {
    glClearColor(1.0,1.0,1.0,0.0);       // background color is white
    glColor3f(0.0f, 0.0f, 0.0f);         // drawing color is black
        glPointSize(2.0);                     // a 'dot' is 2 by 2 pixels
        glMatrixMode(GL_PROJECTION);         // set "camera shape"
        glLoadIdentity();
        gluOrtho2D(0.0, (GLdouble)screenWidth, 0.0, (GLdouble)screenHeight);
             A = screenWidth / 4.0; // set values used for scaling and shifting
             B = 0.0;
             C = D = screenHeight / 2.0;
}
//<<<<<<<<<<<<<<<<<<<<<<< myDisplay >>>>>>>>>>>>>>>>>>
void myDisplay(void)
{
      glClear(GL_COLOR_BUFFER_BIT);     // clear the screen
      glBegin(GL_POINTS);
      for(GLdouble x = 0; x < 4.0 ; x += 0.005)
  {
      GLdouble   func = exp(-x) * cos(2 * 3.14159265 * x);
     glVertex2d(A * x + B, C * func + D);
            }
      glEnd();
      glFlush();           // send all output to display
}
//<<<<<<<<<<<<<<<<<<<<<<< main >>>>>>>>>>>>>>>>>>>>>>>
void main(int argc, char** argv)
{
      glutInit(&argc, argv);          // initialize the toolkit
      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB); // set display mode
      glutInitWindowSize(screenWidth, screenHeight); // set window size
      glutInitWindowPosition(0, 0); // set window position on screen
      glutCreateWindow("Dot Plot of a Function"); // open the screen window
      glutDisplayFunc(myDisplay);     // register redraw function
      myInit();
      glutMainLoop();              // go into a perpetual loop
}
```