Kevin Smith

Math 2605

Project 5 Euler

What I did/ Implementation/ Graphing

For project 5, the goal was to implement Euler's algorithm for any function F = (f(x,y), g(x,y)) and to draw the graph based on the points returned from running the algorithm. My implementation consists of 3 classes. The Point class acts as a single object to hold an x and y value. It's used mostly to draw the graph once Euler's has been run. The Function class is a little larger, and is used to house the function F. It contains an evalF function which evaluates and returns the result when give some x and y. evalF uses an external library called MVEL. MVEL is used so that I can take in f(x,y) and g(x,y) as strings from the user, and then evaluate those strings using some defined x and y that the user decides. Originally, I planned on writing my own evaluator, but in the end decided this was a much easier and efficient way to get the job done. MVEL can be found at the link below.

http://mvel.codehaus.org/

Finally, the main class is what does all the work. The main class opens a GUI window with an empty graph and a start button below. When you hit the start button, dialog windows come up for the user to enter f and g, h, t, and the starting point. It then runs Euler's algorithm, and draws the graph 3 times based on h, each time making h smaller and smaller.

Analysis

Test 1: F(x, y) = (y, -x) with a t = 2PI

If I start with an h of .01, it's almost perfect. The first graph though does not quite aline with the other two, and doesn't quite close. Running the program at .001 works well, but does take a considerably longer amount of time to finish generating the points needed to draw the graph. The result does look like a circle, and the circle does close.

Test 2: F(x, y) = (xy(Y^2 – x), (x + 2y)(y^2 – x^2)) with a t = 10

For this one, an h of .01 is pretty much perfect and the path does tend to (4, -2)

Test 3: F(x, y) = (xy(Y^2 – x), (x + 2y)(y^2 + x^2)) with a t = .6

So this one seems to work fine at .1. The lines aren't indistinguishable, but really very close. It is worth noting that the larger values of h go farther than the smaller values of h. If you consider this not to be a problem, then .1 is fine and it is a smaller h then the stable version. However, you could make h small enough so that they all went off the screen and you definitely wouldn't be able to tell the difference then. In this case h would be smaller.