Zach Hallowell

CSCE 155N

Quinn Lanik

2 May 2021

Final Project Report

My final project was a gui program that solves a quadratic equation for real roots of x. The program is also able to graph the function, its’ first, and second derivatives, as well as calculate and graph the integral of the given function. I used pushbuttons, toggles, input dialog boxes, and text within my ui controls. I also had four unique callbacks. The first callback (solveEqn) was used to solve the user provided equation. I used an if statement to divide the problem into three parts. The first part determined if the root was real or not. The second determined if the root was repeated. The last solved the more general case with two real roots. The second callback (plotEqn) was not only used to plot the given function, but also calculate its first and second derivative and plot them as well. This was done by taking the user inputs and writing a general equation for power rule derivates and plotting each case. The third callback (togEqn) was used to calculate the integral of the given function, open a new figure window, and plot the integral. This was done by using a general integral formula for power functions and applying the same user inputs. The final callback (closeButton) is a simple button control that closes all currently open figures and boxes.

I encountered many issues when trying to code this program. The first was getting the callbacks to work properly. I struggled for many hours to get my program to properly reference my callbacks. When trying to run the program, I would get errors saying, ‘not enough input arguments’ or ‘variable \_\_ is not defined’. I realized that I needed to maintain the same number of inputs in my callback function as the callback itself. I also struggled to get the user inputs into a useable form. When I used breakpoints to determine the problem, I found that the user inputs were coming in as a number but were actually stored as a cell array. I had to use the cellfun() and str2num functions to get the user input into a useable number. Another problem I had was while writing the solveEqn callback. I got errors that said that the ^ operator was not supported for the type of information provided. The solution, a rather simple one, was to change the ‘^’ to a ‘.^’. I also struggled using code to calculate the derivates. At first, I would occasionally get the correct equation, but when I gave the function a coefficient for “d”, the calculation would be wildly incorrect. I eventually solved this problem by simply subtracting d from c in the first step so that I would have the equation in the form ax^2 + bx + c = 0. This made the calculation much easier to code. The last major problem I had was that the instructions text box at the very beginning of the program would open simultaneously with the first input box. I learned to set the hlpdlg box equal to a variable and run uiwait on that variable. This allowed the instructions box to open first and only open the first input box once the instructions box was closed by the user.

There are many practical applications, especially in STEM, for gui’s. For example, a gui would be very useful for solving differential equations and linear algebra problems. The format of a gui allows the creator to program an interactive display that makes solving multiple equations or problems in a short period of time very easy. They can also be used to help students visualize problems better, as they are capable of plotting and solving multiple different problems at one time. Another practical application for gui’s is plotting multiple different data sets in one place. Gui’s can store theoretically unlimited amounts of data and use controls to separate them and allow the user to access only the data set that they need. This could be useful for large projects, where searching through large amounts of data every time that a single point is needed is rather impractical.