

I sent out an email to a number of the LAs because I was pretty late to the whole getting approved party. Laurel Hilger was the first to reply and a screenshot of the email will be featured below.



Laurel Hilger, Matthew Koenig
COMP SCI I: ENG & SC CSCE155N SEC 210 Spring 2021

April 29, 2021 at 6:55pm



Hi Matthew,

I think this could be a good project. I believe a lot of this challenge will come from inputting the functions from a user interface. Just keep in mind that you must include 3 unique uicontrols, 2 callback functions, and error handling for all functions. Good luck!

Laurel Hilger

From: Matthew Koenig <notifications@instructure.com>
Sent: Thursday, April 29, 2021 6:13 PM
To: Laurel Hilger <lhilger2@huskers.unl.edu>
Subject: Matthew Koenig (COMP SCI I: ENG & SC CSCE155N SEC 210 Spring 2021) just sent you a message in Canvas.

Non-NU Email

Getting Approval for Final Project

Hi,

For my final project I'm thinking about making a program that takes a (mathematical) function from the user and gives back the zeros for it. It would also have a button to graph the function. Please let me know if this is enough or if I need to change anything.

Thanks,
MK

Design:

Going in, I knew that I would need 1 main function to hold all of the uicontrol stuff, 2 functions for callbacks from push buttons, and 1 final function to filter user input from the edit box. The main function which I named EquationSolver would house 3 text boxes, 2 push buttons, 1 edit box, and 1 plot. The first callback function called findZeros would look at what the user given function, use matlab's solve function to find solutions, and then display said solutions in one of the text boxes. The second callback function called plotExpression would look at the user given function and then plot it. The function that filters user input called inputScreener would look at user inputs and ensure that they contain a lowercase x but no unwanted characters like random letters or symbols.

Coding Process & Difficulties:

For the main function, EquationSolver it was fairly straightforward to make all of the uicontrol features and the only difficulty I found was going through each of them and making sure that they didn't overlap.

For the findZeros function I ran into a difficulty right away with trying to get the mathematical function the user put into the edit box. I ended up needing to use the get function to get what the user typed into the edit box rather than the default text in the box. I then had to translate that string into a symbolic function which matlab could then find the zeros of. I then translated those zeros into non-symbolic expressions and put those zeros into the textbox meant to hold them.

For the plotExpression function, I had to go through with the same process of getting the mathematical function from the user. I then had it plot the symbolic function on the plot.

For the inputScreener function, I first made a regular expression to check if the user input included an 'x' and return a false if not. I then made a second regular expression to check that the user used characters of a mathematical expression and return a false if they didn't. The inputScreener would then be called at the beginning of the callback functions and return if the inputScreener detects a bad input from the user.

GUIs in the future:

I would find it extremely helpful for my major if I had a GUI that could solve differential equations and or systems of equations. If there is some way to do Laplace transforms through Matlab, then I could fairly easily make a GUI to solve differential equations. Systems of linear equations would be a bit easier with Matlab's built in rref function, but systems of differential equations might get a little bit complicated. With some further research of Matlab's functions I could probably make both of these over the course of a couple weeks or so.