MATLAB Programming NYCU Fall 2023 LAB#10 2024/12/5

You will write an additional app in this lab, as a third lab on GUI programming. Instead of working with images, the topic of this lab is a mathematical function viewer. The following are the list of GUI objects:

- An ***axes*** for plotting the function.
- An ***edit field*** for the user to type/edit the function to be plotted. Let **`x`** be the only variable; the function is represented by a character vector of Matlab expression. (Note: The expression should work when **`x`** is a numeric vector.)
- A ***dropdown*** control or selecting the line style for plotting the function. Initialize the list with standard Matlab line style specification strings.
- A ***radio button group*** for selecting the symbol/marker type for plotting the function. (Only one ***radio button*** can be selected at a time.) You can use the **`switch-case`** statement to identify the marker type selected.
- For line color selection: A ***pushbutton*** Color that, when pushed, call **`uisetcolor`** to bring up the color selector dialog box. The returned value is a 3-element vector for the RGB values. In the example below, the selected color is set as the **BackgroundColor** of a ***label*** for visualization.
- Two ***edit fields (numeric)*** for setting the **`x`** limits of the plot. The **`Value`** property of such ***edit field*** is automatically numeric.

The actual function evaluation and plotting:

- Use the statement: **`fp = fplot(ax, fh, [x1 x2]);`**
- Here **`ax`** is the ***axes***, **`fh`** is a function handle, and **`[x1 x2]`** is the range of **`x`**. The function handle can be obtained using **`str2func(['@(x) ' s])`**, where **`s`** is a character vector (the **`Value`** of the function ***edit field***).
- Place the actual function evaluation and plotting in a separate function. In you code, add callbacks for the **ValueChaged** event for the ***edit fields*** and the ***dropdown*** control, **SelectionChanged** for the ***radio button group***, and let the callback functions call this separate function. The same after a new color is selected.
- Use **`set(fp, attribute-name, attribute-value)`** to update the plot.

Exception handling (optional):

- Error messages provide valuable feedbacks for the app users when something goes wrong. In this app, there are at least three possible types of user errors: (1) The function ***edit field*** contains an invalid function expression; (2) Error during evaluation, such as division by zero; (3) Invalid **`x`** range.
- A simple way of exception handling is to place a **`try-catch`** statement around the function evaluation and plotting code. If an exception is caught, bring up an error message using **`uialert`**.

The following is an example of the app: