ME 350 Lab Exercise 5 User-defined functions and root-finding
Fall 2017

To prepare for the lab exercises, download the `meanFixNaN.m`, `tanxInvxPlot.m`, and `tanxInvxPlotAnon.m` functions from the class web site.

You will also need to download `brackPlot.m`, and `bisect.m` for the root-finding exercises.

Practice with User-Defined Functions

1. Run the `meanFixNaN`, `tanxInvxPlot` and `tanxInvPlotAnon` functions and verify that you can read and describe how those codes work.

2. Create a modified version of `tanxInvxPlot` and `tanxInvPlotAnon` that can plot
$$f(x) = \tan(ax) - \frac{1}{bx}$$
Test your function with $a = 1$ and $b = 20$.

Root-finding with `brackPlot` and `bisect`

1. Write a stand-alone m-file called `myfx` to evaulate
$$f(x) = x - x^{1/3} - 2 = 0. \tag{1}$$
Your m-file should *only* have one input, x, and return one value $f(x)$. Note that both x and $f(x)$ can be vectors. Your m-file should only evaluate $f(x)$. It should not generate a vector of x values. It should not plot $y = f(x)$.

2. Test that your `myfx` function from the previous exerise is working by writing a new m-file, say `plotMyfx` that calls `myfx` to generate data for a plot of $f(x)$ on the interval $0 \leq x \leq 5$.

3. Write another (new) m-file function called `fxroot` that uses the `brackPlot` function to find the brackets for roots of the $f(x)$ defined by Equation (1) on the interval $0 \leq x \leq 5$. Store the brackets in a matrix, say `xb`.

   Since the `brackPlot` function plots $f(x)$ on the designated interval, you no longer need to separately create vectors of x and $f(x)$ values in order to plot $f(x)$ as you did with `plotMyfx`.

4. Add to `fxroot` a call to the `bisect` function to refine the find the root within the bracket interval produced by the call to `brackPlot`.

5. Continue your modification of `fxroot` by substituting the root returned by `bisect` back into the $f(x)$ function to demonstrate that you have found a true root. Use a `fprintf` statement to print both the root and $f(x)$ at the root.

Root-finding with a Nested Function to Define $f(x)$

The goal of this exercise is to write a new version of the code to find the roots of Equation (1). The new version uses a nested function instead of a separate m-file function to define $f(x)$. The final version of the code is not much different from the `fxroot` function.

1. Make a copy of the `fxroot` function that you used to complete the preceding exercise. Call the new function `fxrootNested`. In the following exercises, make one change at a time and to test your function after completing each step.

2. Add a *nested function* to the `fxrootNested` m-file to evaluate the $f(x)$ in Equation (1). In other words, the nested m-file replaces the external m-file you used with `fxroot`.

3. Modify your *call* to the `brackPlot` function so that `brackPlot` uses your newly written nested function.

   Hint: If your nested function is called `myfun`, you will need to pass the *function handle* to `brackPlot` with the following syntax.

   ```
   % -- Define the nested function
   function z = myfun(x)
      z = ...
   end

   xb = brackPlot( @myfun, ... )
   ```

   Note the `@` in `@myfun`.

4. Given the bracket found in the preceding step, use the `bisect` function to refine the root. Evaluate and print the value of $f(x)$ at the root. *Hint*: `r = bisect( @myfun, ...)`.

The result of the preceding steps is a single m-file (called `fxrootNested` that runs `brackPlot` to find the bracket containin the root, and `bisect` to refine the root.

Root-finding when $f(x)$ has multiple roots and singularities

1. Write a single m-file that finds and plots the roots of

   $$\tan(ax) = \frac{1}{x} \tag{2}$$

   for $a = 1$ in the interval $0 < x \le 5\pi$. Why do we avoid $x = 0$?

   Verify that each root returned by `bisect` is truly a root, and not a singularity.

2. Modify your solution to the preceding exercise so that it solves finds the roots in the same interval for any value of a. Find the roots in $0 < x \le 5\pi$ for $a = 0.5, 1.5, 2.0$.