Math 451 ASSIGNMENT 6 **Due at 4 PM Thursday, April 10th**

For this assignment you will write a script file that finds the solution to $\cos x = x$ using the Bisection Method, starting with an initial interval $[a_1, b_1] = [0, 1]$ in which a root is believed to lie. Here is a rough description of the algorithm:

- Test the product $f(a_1)f(b_1)$ to make sure there is a guaranteed root in the interval. If not return a message to that effect and stop.
- If the test indicates a root in the interval,
  - find the midpoint m of the interval,
  - test $f(a_1)f(m)$ to see if there is a root in $[a_1, m]$. If there is, set $a_2 = a_1$ and $b_2 = m$.
  - Otherwise, set $a_2 = m$ and $b_2 = b_1$.
- Repeat until the desired tolerance is obtained.

You should first write a script that will find the root *assuming there is one there*, so skipping the test of $f(a_1)f(b_1)$. Here are some suggestions as to how to do this, using the same idea of building your program incrementally, as demonstrated in class Monday:

1. Initialize a_1 and b_1 as sequences. Write a line to compute the first midpoint, create the function we're using, and compute $f(a_1)f(m)$. Make sure this all works.
2. Based on the result of $f(a_1)f(m)$, determine a_2 and b_2 in terms of a_1, b_1 and m. (Note that m can be a sequence or can be simply recomputed each time - it doesn't matter which you do.) Run this to make sure it works.
3. Put in a line initializing $k = 1$ and rewrite what you have now with the steps from the last suggestion done in terms of the general counter k rather than the numbers one and two. (You could start k at two, but I think you will find things go more smoothly if it starts at one.) Run again - you should get the same output as the last time.
4. Now write do the defining of a_2 and b_2 inside an `if-else` statement. Run again, still expecting the same result.
5. Create a `for` loop around the `if-else` statements to generate the first four or five $[a_k, b_k]$ intervals. Run again, of course!
6. Change your loop to a `while` loop, having it continue until two consecutive x_ns are within 10^{-4} of each other. (Include a "safety" stopping condition of 100 iterations.) You know what to do again!
7. Add comments to what you have so far. Be sure to save!

If you made it this far, you now have the "meat" of the code. Have the script output *two* vectors, one with the of successive values of a_k and the other with the successive values of b_k. We now wish to create a plot that shows the values of both the a and b sequences together on the same plot. Write a command to plot the a sequence in some color and with each point indicated by some symbol. Try `help plot` to see your options - have a little fun with this. Now add a line that says `hold on` and then add another line for plotting the b sequence with another color (please don't choose green - it is too hard for my old eyes to see!) and symbol. This should give the desired graph.

Now we want to go back and add the parts needed to test the product $f(a_1)f(b_1)$ to make sure there is a guaranteed root in the interval. If not we wish to return a message to that effect and stop. To do all of this you will need to "enclose" the code you have so far with another `if-else` statement that makes all that happen. Check to see that everything works and gives the desired output (and no more!). You can test to see whether the initial check works by changing b_1 to 0.5 and running it. It should give a message that there is no guaranteed root in the interval. (Be sure to change b_1 back to one after doing this.) Send the result to me as *yourname_assn6.m*.