University of Pennsylvania
EAS 105: Fall 2011
MATLAB PROJECT: Root Finding
Instructor: Donna Dietz

Before you begin this project, you should become familiar with the Bisection method and Newton's method. You can search online or find this in your class notes or in an old Calculus textbook. Then, go online to Canvas, in *Files* and download the zipped folder with items for this project. Download this folder and unzip it.

This project, and all projects for this course, will have a common format. Please become comfortable with this format. It is designed to help you focus your efforts and reduce wasted time.

Each project will have instructions, which is what you are reading now. For each function you are asked to write, you will be creating a text file with the name *NameOfFunction.m* where *NameOfFunction* varies according to the function. For the first line of this file, copy the appropriate *signature* (the boldface lines) found in the instructions. Also, for each function you are asked to write, you will be provided with a *pcode* file, or a precompiled version of the function. The *pcode* files will help you in several ways. First, if you wish to play with the function, you may! You may execute it just as if it were a *.m* file, but you can not read it. Second, because you can tell what your function is supposed to do, you can design your own tests if you wish. Third, if a specific function must be present in order to begin work on another function, you may temporarily use the *pcode* file in place of the *.m* file. This prevents you from getting stuck in one spot on a project where there are several dependencies. (Then, you can bring your questions to office hours!)

You will also be provided with a *Tester* file which is a *.m* file. You can use this file in one of several ways. First, place a copy of this file in the directory which you are using for your project. Run it. A report file will be generated which will advise you on your progress and recommend what you should work on next. Second, if you want to see the code, you are welcome to read this file. This will help you figure out why your code is not working correctly. (If you suspect a bug in the Tester, please report it!!!) You can adjust *your* copy of the Tester if you wish, because a fresh copy will be used to grade your project. (The goal is for everyone to get 100% on all projects!!!) Third, since you have all the *pcode* files, you may begin your project by putting all the *pcode* files and the Tester file together in a folder and slowly deleting the *pcode* files as you replace them with *.m* files. (If you do this, you might want to do this in parallel with the other suggested strategy, so you can get feedback on what you need to work on next!)

You are allowed to create additional functions if you wish, but that should not be necessary for this project. When your project passes the Tester, you should zip it all up and submit it through Canvas. If you submit it after the due date but less than a week late, you will be penalized 10% of the value. After a week, it will be penalized 20%. No projects may be submitted for credit after the final exam has been given.

For this project, create these three functions:

function [root] = BisectionRootFinder(f, a, b, tol) BisectionRootFinder implements the Bisection method. f is the handle of a function to find the root for, which is between a and b, with $a < b$. The signs of $f(a)$ and $f(b)$ must differ. The root which is returned should have an error no greater than tol. This may be implemented by looping or by recurrsion.

function [root] = NewtonRootFinder(f, fprime, c, tol) f is the handle of a function you want to find the root for, and its derivative is $fprime$, which you must supply to the function. The function begins its search at c and implements Newton's method. The root which is returned should have an error no greater than tol. This may be implemented by looping or by recurrsion.

function [newtonTime, bisectionTime] = CompareNewtonWithBisection(f, fprime, a, b, tol) This function calls BisectionRootFinder with the obvious inputs, and calls NewtonRootFinder with a starting point halfway bewteen a and b. The commands *tic* and *toc* are used to time the executions of these two functions. Actual times will vary by machine, time of day, moon phase, and general attitude of the system you happen to be on. However, the times should give you some intuitive feel for the differences and similarities.