Homework 6 Report Problem 2w

Group 334-7

HW2	Cameron Rabiyan	Maya Apotheker	Manny Gamboa
Coding	33.3%	33.3%	33.3%
Results	33.3%	33.3%	33.3%
Report	33.3%	33.3%	33.3%

Introduction to Problem 2:

In problem 2, we were instructed to create a python code that can compile, run, and plot Newton's root finding method. We were given a prewritten program that can implement this root finding method.The primary goal was creating a python code that can create, rename, and store multiple values for the rootfinder in the scenario that it is ran multiple times. When this code is ran, the code checks for the existence of rootFinder.init and if it exists, renames the original rootFinder.init to rootFinder.init.1 and stores the new data in the new rootFinder.init. After being able to implement this file manipulation code, we are tasked with using matlibplot to plot the data pulled from the code on a 2 graphs: one where the solution is related to the iterations of x, and another where the error associated with the user-defined guesses in the rootFinder.init file is related to the iterations of x.

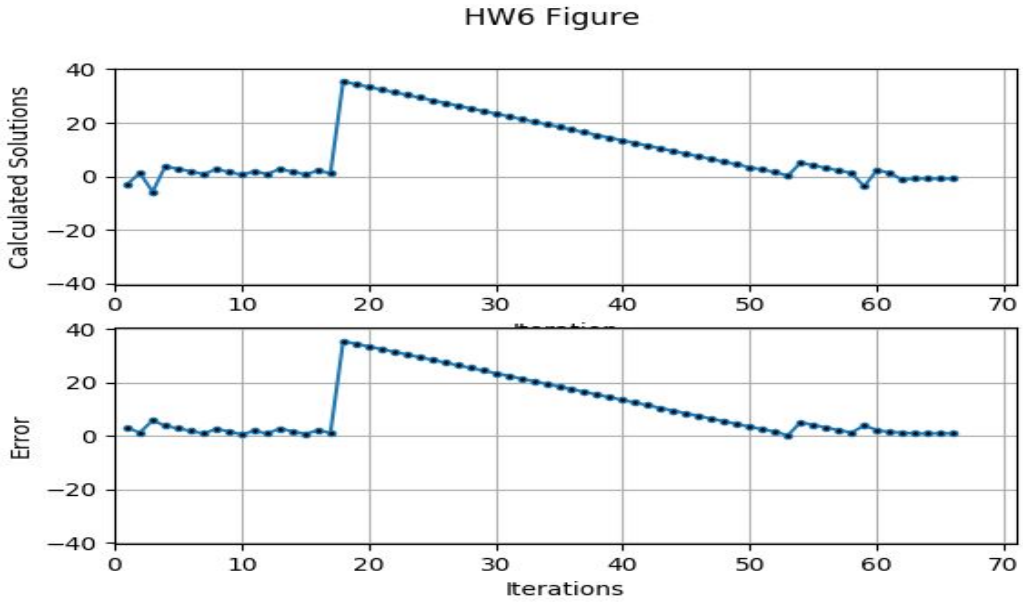
Procedures for Problem 2:

1. Remove all preexisting rootFinder.init files in newton_rootFinder.
2. Use the make_make function that checks the existence of rootFinder.init and then compiles fortran code located in a different directory, namely Hw6/newton_rootFinder/.
3. Create a runtimeParameters_init function which again, checks the existence of a rootFinder.init file. If a rootFinder.init file does not exists, this function will then generate and populate a new rootFinder.init file. If one already exists, it will increment the rootFinder.init file's name to rootFinder.init.*, where the largest * is the total number of rootFinder.init files minus one.
4. Create a def_rootfinder function, which executes the Fortran executable using the recently recreate rootFinder.init file.
5. Create a plot_data function which creates two graphs: one where the solution is related to the iterations of x, and another where the error associated with the user-defined guesses in the rootFinder.init file is related to the iterations of x. This information is plotted both on-screen and into a .png file. After closing the graph, this information will

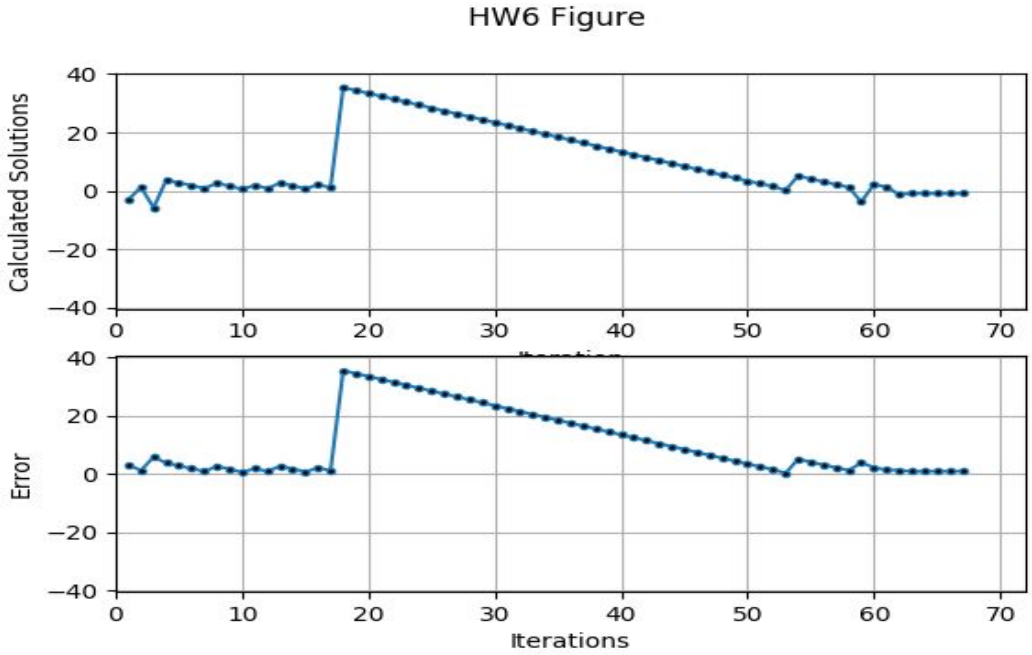
automatically populate rootFinder_newton.dat, or rootFinder_newton.dat.*, depending on the number of graphs saved.

6. Create a function that allows us to call the previously created functions using the user-defined runtime parameters and calls all the functions listed above with three different threshold values, 1.e-4. 1.e-6, 1.e-8.

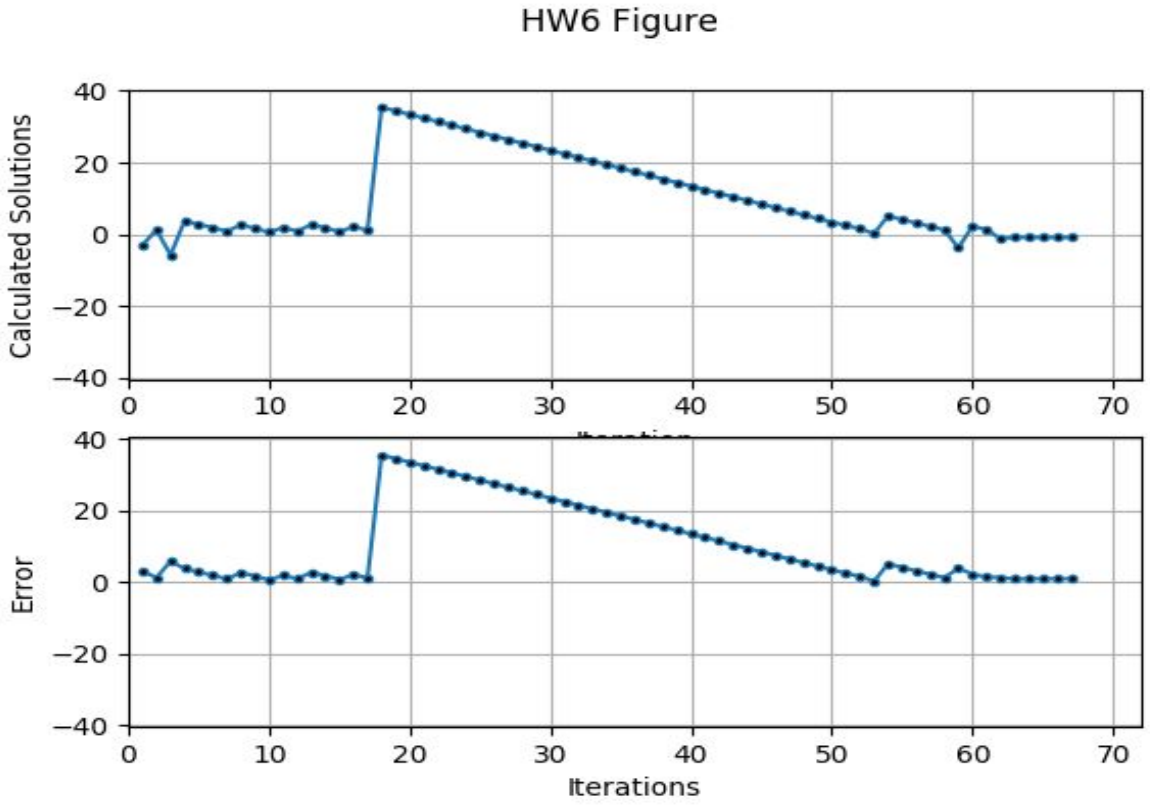
Threshold = 10^{-4}



Threshold = 10^{-6} **asd**



Threshold = 10^{-8}



Conclusion:

Based on our png file, you can see that we got 67 total iterations of our value x, meaning the 67th iteration was when the residual passed our user-defined threshold. When the code is executed, user-defined values are taken and executed in the proper fashion:

- compiles fortran code.
- writes a parameter file rootFinder.init for 3 different threshold values; 10-4, 10-6, and 10-8.
- executes a fortran code with these respective values.
- plots, displays, and stores the information the graph into a file with syntax result_1_(threshold).png.