Newton Method Report
Homework # 6

Brian Vargas
AMS209 Foundations of Scientific Computing
University of California, Santa Cruz

November 21, 2016

1 Introduction

For this homework assignment, we went ahead and continued to work with Newton's method. We combined the usage of Python 2 with Fortran. We have Fortran running as the computing engine - everything we do as far as scientific computing is run through Fortran. We used Python to control the operating system and essentially feed the necessary information for the Fortran program to conduct its computations.

The Fortran files were all copied off of Professor Dongwook's code directory for the class, with the exception of the makefile altered to fully clean out the non-essential files. The students created a pyRun_rootFinder.py script in order to make this task happen.

Within the Python script we traverse directories to control the makefile, create an init file with appropriate input parameters, run the executable, and lastly plot the results.

2 Testing and Results

We tested our code on the following function:

$$f(x) = (x - 1) \log x,$$

which obviously has a root at $x = 1$.

We ran the testing for two different initial guesses all in one execution of the Python script.

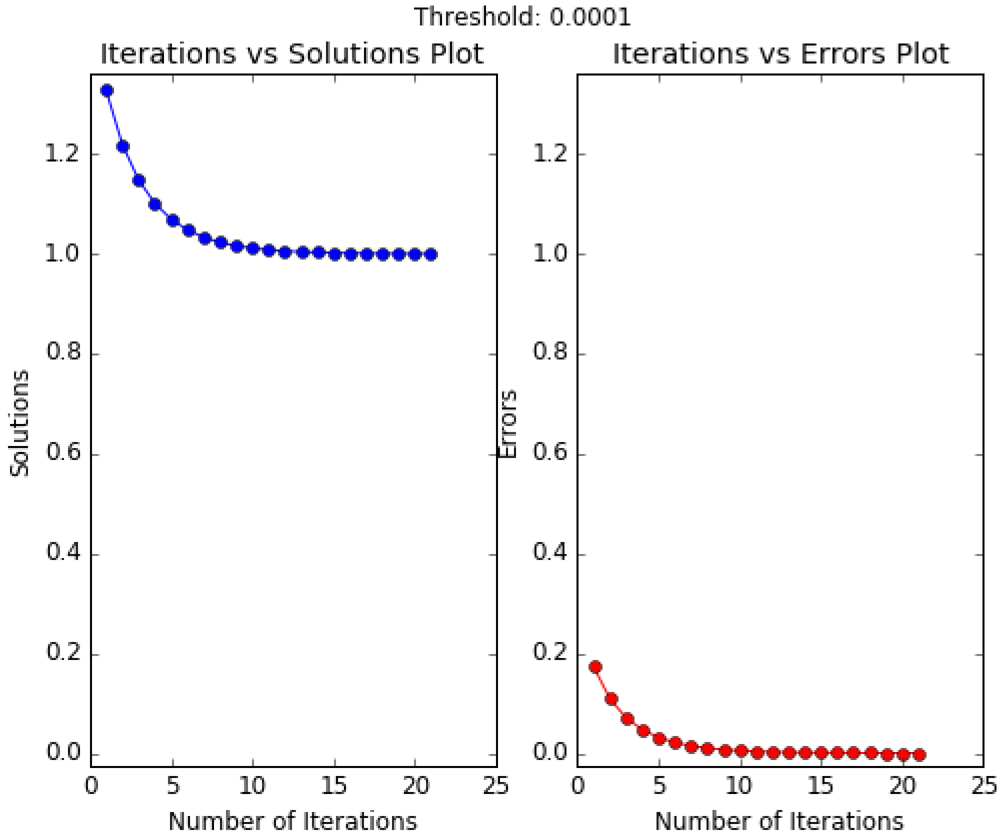
2.1 Close Initial Guess

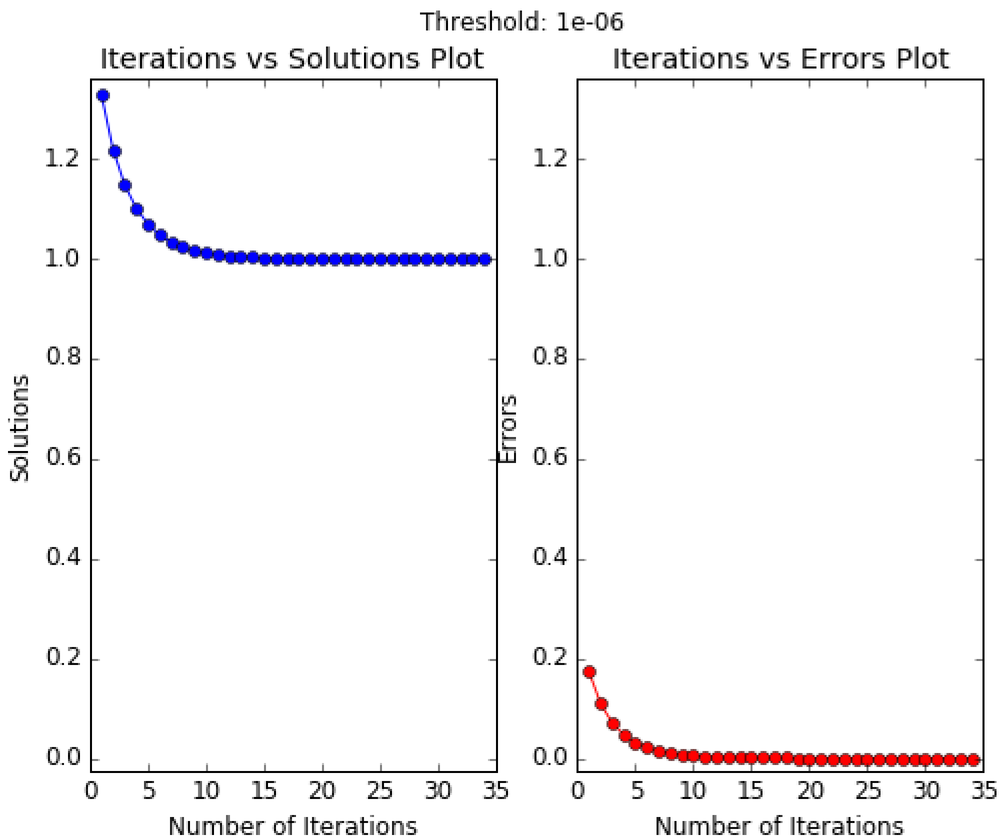
First we tested our code using an initial guess that is close to the true solution: $x_0 = 1.5$

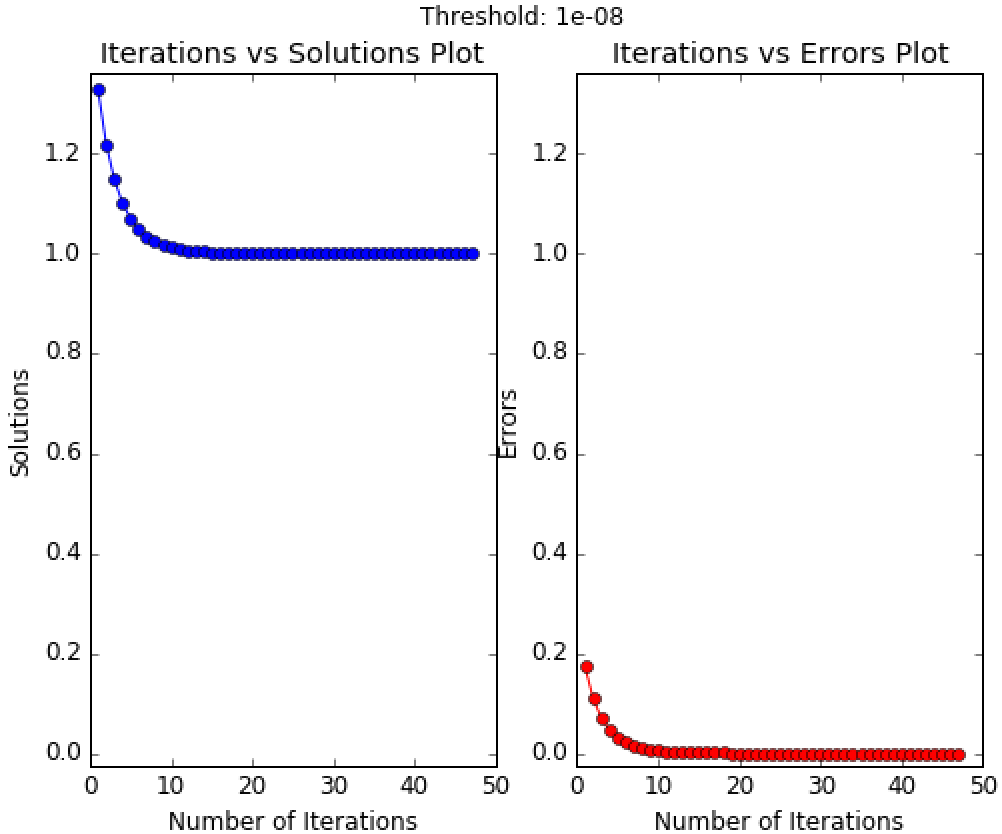
We obtained the following result summaries from the output:

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
             Solution Convergence Summary
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Your converged solution x =    1.0002087626508687
Solution converged in Nstep=          21
Threshold value =    1.0000000000000000E-004
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
             Solution Convergence Summary
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Your converged solution x =    1.0000019200371568
Solution converged in Nstep=          34
Threshold value =    9.9999999999999995E-007
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
             Solution Convergence Summary
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Your converged solution x =    1.0000000176608239
Solution converged in Nstep=          47
Threshold value =    1.0000000000000000E-008
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Obviously we were going to have more iterations to converge to a solution with a smaller threshold value. Now, let's look closely at the results plotted - number of iterations vs their respective solutions and errors.


Threshold: 0.0001
Iterations vs Solutions Plot
Solutions
Number of Iterations
Iterations vs Errors Plot
Errors
Number of Iterations
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20
25


Threshold: 1e-06
Iterations vs Solutions Plot
Solutions
Number of Iterations
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
35
Iterations vs Errors Plot
Errors
Number of Iterations


Threshold: 1e-08
Iterations vs Solutions Plot
Solutions
Number of Iterations
Iterations vs Errors Plot
Errors
Number of Iterations
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
10
20
30
40
50

As we can see from their shape, they are closely related. Since the initial guess is close enough to the true solution, it converges rapidly and goes straight to the solution.

2.2 Further Initial Guess

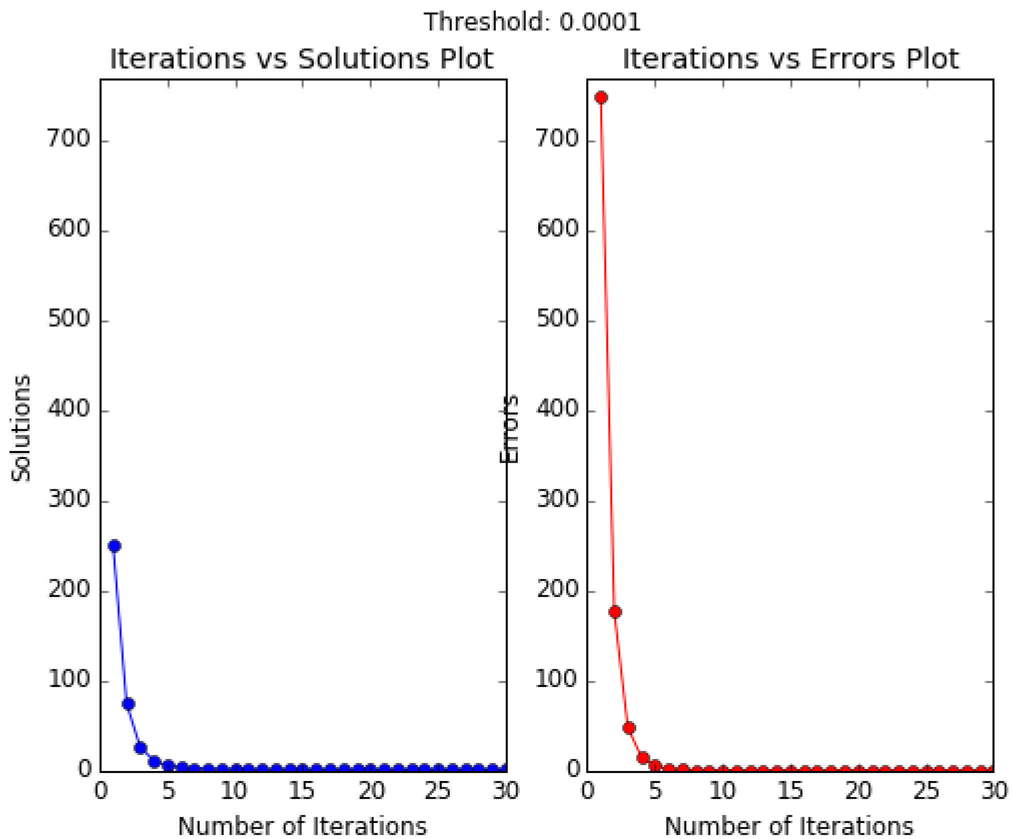
Next we tested our code using an initial guess that is much farther from the true solution, $x_0 = 1000$.

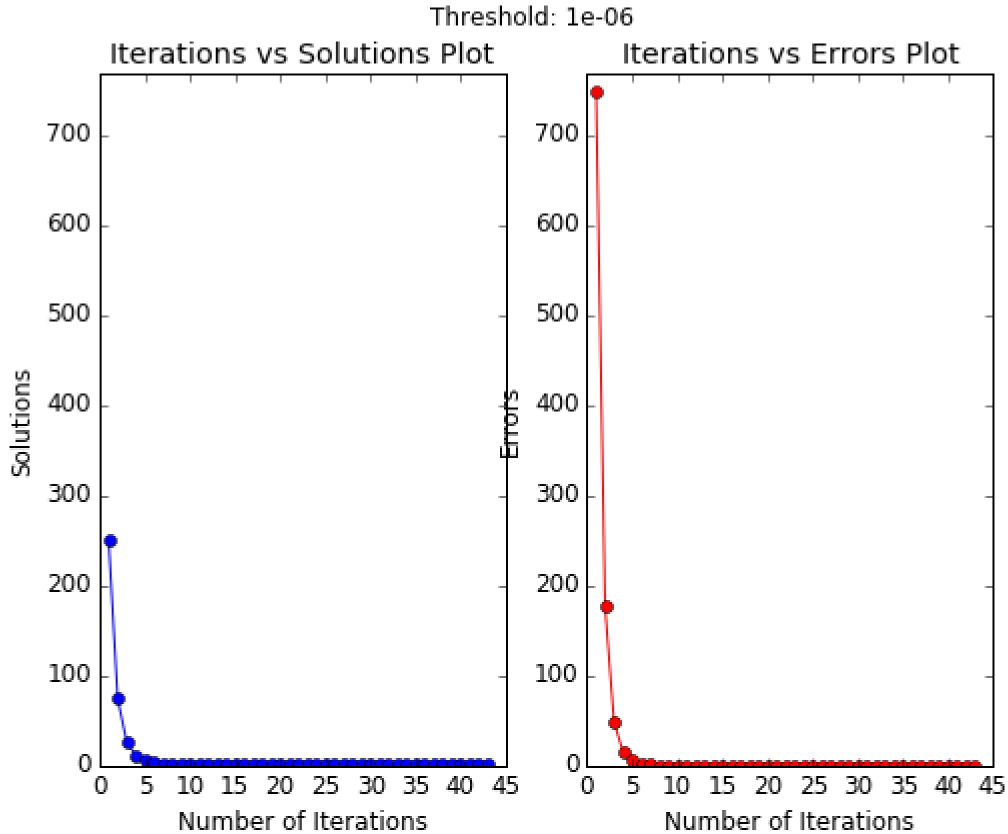
We obtained the following result summaries from the output:

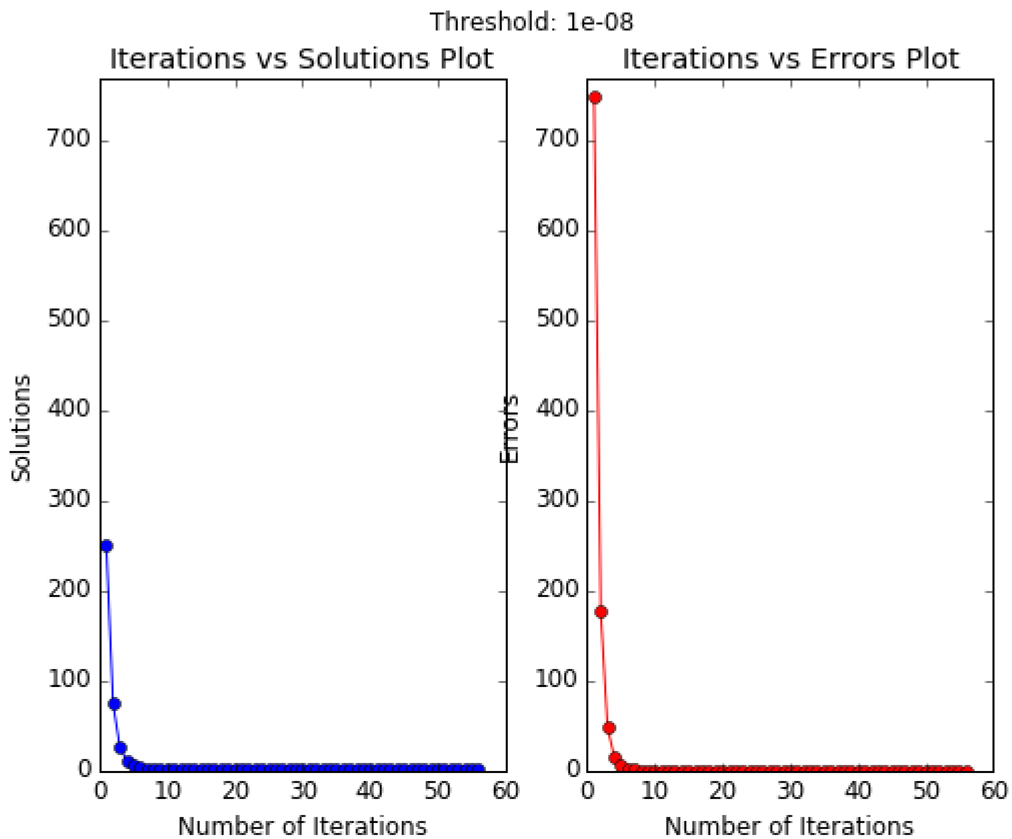
```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
             Solution Convergence Summary
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Your converged solution x =    1.0002215983631804
Solution converged in Nstep=          30
Threshold value =    1.0000000000000000E-004
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
             Solution Convergence Summary
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Your converged solution x =    1.0000020380771388
Solution converged in Nstep=          43
Threshold value =    9.9999999999999995E-007
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
             Solution Convergence Summary
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Your converged solution x =    1.0000000187465745
Solution converged in Nstep=          56
Threshold value =    1.0000000000000000E-008
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

As we can see, even though our initial guess was ridiculously large compared to the true solution, we still managed to converge. However, it did take us a few more iterations but nonetheless it wasn't an extreme number of iterations to converge within our desired threshold. This comes to show that Newton's method is a good method to use when we are looking for roots without knowing their close vicinity.

Now let's look closely at the results plotted.


Threshold: 0.0001
Iterations vs Solutions Plot
Solutions
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
Number of Iterations
Iterations vs Errors Plot
Errors
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
Number of Iterations


Threshold: 1e-06
Iterations vs Solutions Plot
Solutions
Number of Iterations
Iterations vs Errors Plot
Errors
Number of Iterations



Threshold: 1e-08
Iterations vs Solutions Plot
Solutions
Number of Iterations
Iterations vs Errors Plot
Errors
Number of Iterations

As we can see from their shape, they are closely related. Since the initial guess is close enough to the true solution, it converges rapidly and goes straight to the solution. Notice that these plots, in comparison to their closer initial guess counterparts, begin with a much larger error and rapidly converge still.

3 Conclusion

Newton's Method is a good method to use when finding roots. More often than not, the solutions will converge. Obviously being closer to the true solution will converge us faster, but if we have no idea where the true solution is near, we can rely on Newton's method to do the work for us. There are some cases where Netwon's method will diverge but there are ways around that!