One-Dimensional Line Search Methods Assignment

Advanced Design 814

Andreas Joss

16450612

March 13, 2015

1 Introduction

This report briefly explains the three different one-dimensional line search methods which are used for this assignment. Furthermore, the algorithms are implemented in code and the results thereof are shown and discussed.

2 Theory

2.1 Newton-Raphson Method

The Newton-Raphson method can be derived from the Taylor Series expansion which is used to approximate functions. The Newton-Raphson method is described as:

$$x^* \approx x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)} \tag{1}$$

where x^{k+1} describes the next solution and x^k describes the present solution.

2.2 Newton-Raphson Method Modified

The purpose of this modification is to ensure that a solution is found for the given problem function, for cases where the normal Newton-Raphson method would diverge. This modified method might require more iterations than the unmodified method, nevertheless convergence is more likely.

The modified Newton-Raphson method is described as:

$$x^* \approx x^{k+1} = x^k - \frac{f'(x^k)}{\max(f''(x^k), \epsilon)} \tag{2}$$

For the denominator of Equation 2, the maximum value is selected being either the second derivative of the function at the current solution, or a value ϵ which is normally selected in the order of $\epsilon = 10^{-6}$. One problem arises in the unmodified Newton-Raphson method (Equation 1) when the denominator is equal to zero or very close to zero. This causes the newly calculated solution x^{k+1} to assume a value placed very far, or even infinitely far, from the previous solution ^{k}, which as a result makes it very hard for the unmodified Newton-Raphson method to converge. The modification of the Newton-Raphson method ensures that the denominator is at least as large as ϵ. The second problem of the unmodified method, is when the second derivative is found to be $f''(x^k) < 0$. This means that the function is non-convex (concave) in that region and that the next solution will move closer to a non-converging region. The modified method, would then prevent the next solution to move into the aforementioned direction, by replacing $f''(x^k)$ with a small positive value ϵ.

2.3 Golden Section Method

For this method to be useful, it is assumed that the objective function $f(x)$ has a minimum within an interval $[a, b]$ (which is selected from user experience), and that $f(x)$ is descending for $x < x^*$ and $f(x)$ is ascending for $x > x^*$. New points x_1 and x_2 are selected when certain criteria are met.

If $f(x_2 > f(x_1)$ then the new interval becomes $[a, x_2]$ with b obtaining the new boundary value, otherwise the new interval will be $[x_1, b]$ with a obtaining the new boundary value. Thus, the interval is reduced until x^* is reached within a prescribed accuracy. The ratio by which the interval is reduced, is determined by the golden ratio $r = 0.618034$. The algorithm steps are explained in full by [1:37].

3 Selecting the Appropriate Method

3.1 Newton-Raphson Methods

From Equation 1 and 2 it is clear that first and second order derivative information of the objective test function is required to use the method. Depending on the problem at hand, it is up to the user to decide how practical this method is regarding the expensiveness to obtain this information. Even if the first and second order derivatives are obtained, the method could still struggle to deliver a global minimum due to its susceptibility to noise. For the modified Newton-Raphson method, values for ϵ can be chosen experimentally for specific problems. If the method still diverges, then a larger ϵ value could be selected.

3.2 Golden Section Method

For this method, only zero order information is required from the objective test function. This is useful when the first and second order derivative information is hard and expensive to obtain, but zero order information is more accessible. This method is consequently much less susceptible to noise.

4 Example Functions

Example functions are used to evaluate the effectiveness of each algorithm and briefly compare the algorithm results with each other. Each graph displays several parameters for the particular optimization at hand as well as the solution that is found. Each test function has its own specified boundaries and also the tolerance of the accuracy of the desired solution. The following test functions are used:

(i) minimize $F(\lambda) = \lambda^2 + 2e^{-\lambda}$ over $[0, 2]$ with $\epsilon = 0.01$

(ii) maximize $F(\lambda) = \lambda \cos \lambda$ over $[0, \frac{\pi}{2}]$ with $\epsilon = 0.001$

(iii) minimize $F(\lambda) = 4(\lambda - 7)/(\lambda^2 + \lambda - 2)$ over $[-1.9, 0.9]$ with by performing no more than 10 function iterations.

(i) minimize $F(\lambda) = \lambda^4 + -20\lambda^3 + 0.1\lambda$ over $[0, 20]$ with $\epsilon = 10^{-5}$

5 Results

The algorithm are programmed in the Python language, due to future work which will also require extensive Python scripting. The Matplotlib library is used to display the results in a neat and informative manner. In each case the green dot indicates the solution that is found (if any) by the algorithm.

5.1 Newton-Raphson Method
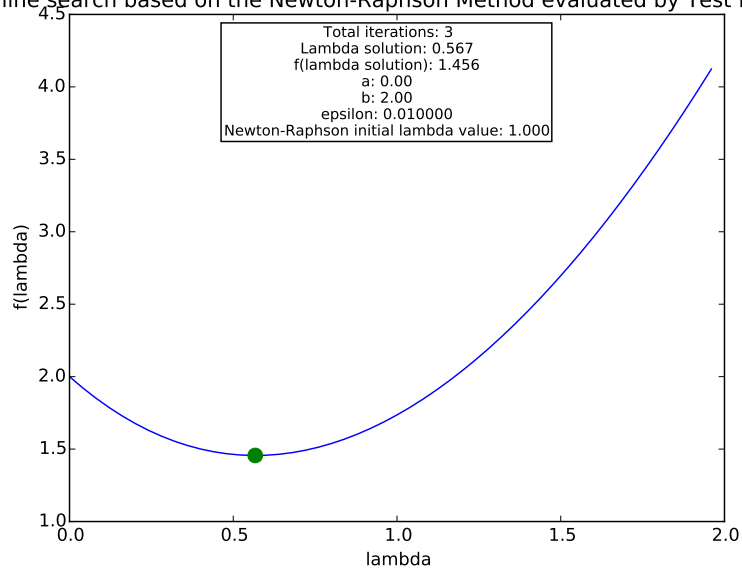

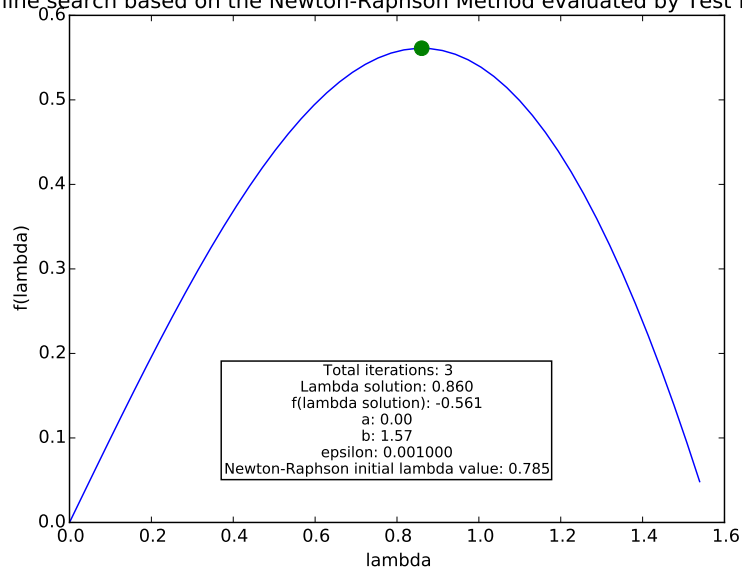Figure 1: The Newton-Raphson Method evaluated by first example function

Figure 2: The Newton-Raphson Method evaluated by second example function

Figure 1 and Figure 2 show that this algorithm manages to successfully converge near or on the optimal point. Note for both test functions, only three iterations are required.

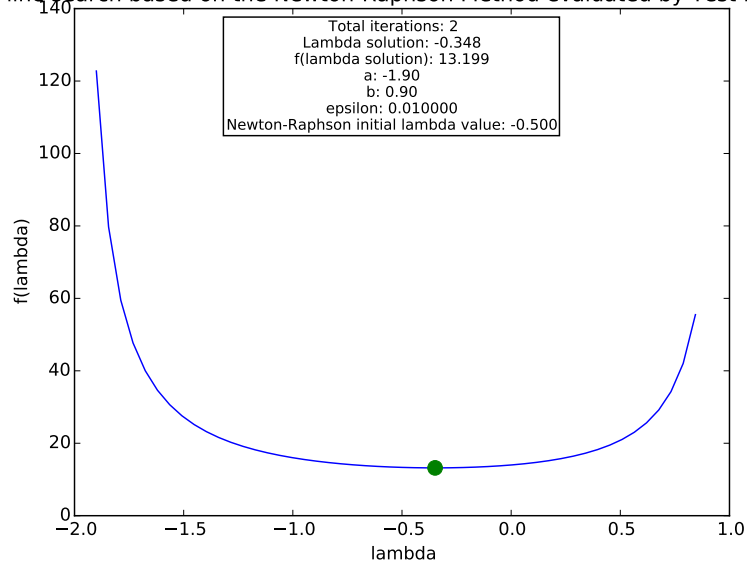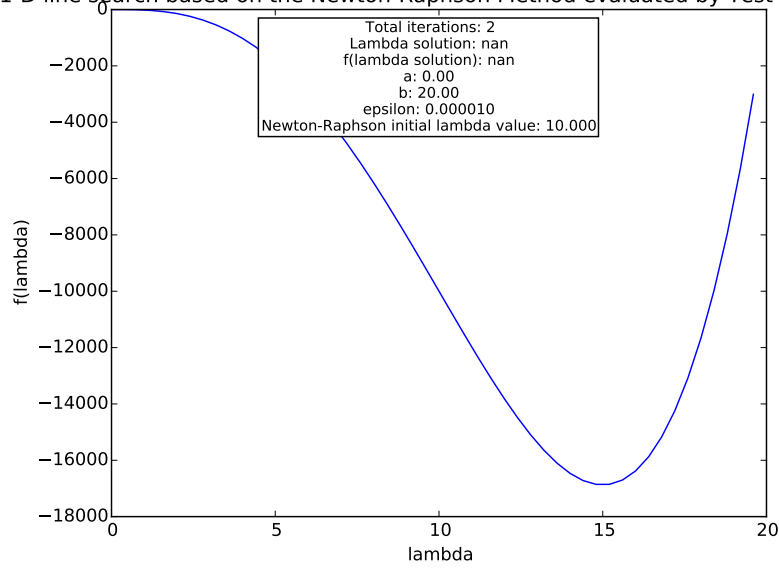Figure 3: The Newton-Raphson Method evaluated by third example function

Figure 4: The Newton-Raphson Method evaluated by fourth example function

Figure 3 indicates that this algorithm manages to successfully converge near or on the optimal point. Note that only two iterations are required to adhere to the tolerance value. From Figure 4, it is clear that the algorithm did not converge to a solution. This is due to the fact that the initial starting solution is not chosen adequately for this particular problem.

1-D line search based on the Newton-Raphson Method evaluated by Test function 4

Total iterations: 5
Lambda solution: 15.000
f(lambda solution): -16873.500
a: 0.00
b: 20.00
epsilon: 0.000010
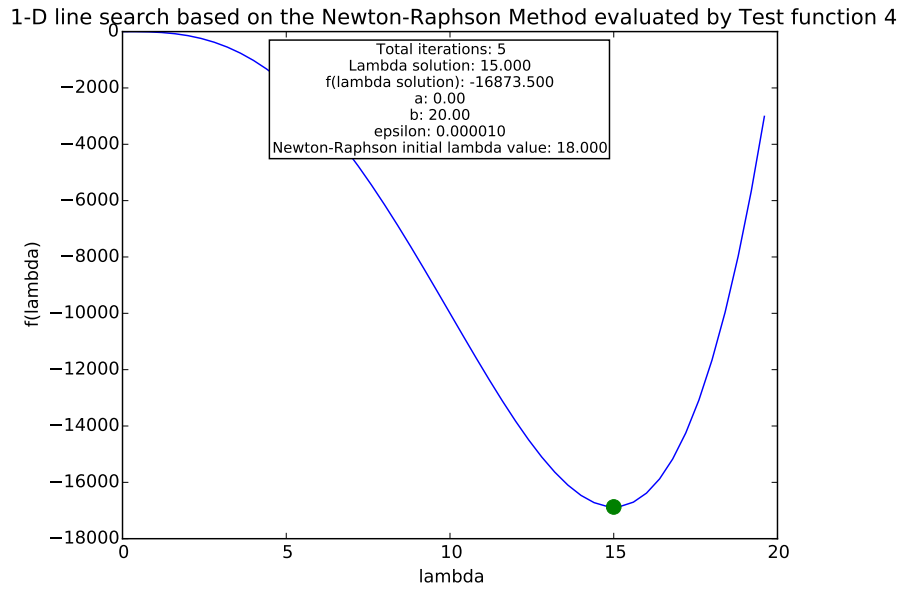Newton-Raphson initial lambda value: 18.000

Figure 5: 1-D line search based on the Newton-Raphson Method evaluated by fourth example function, with a new starting solution selected

This time, from Figure 5, it is visible that the algorithm successfully converged to a solution. This is because a different starting estimated solution is assumed compared to the estimated starting solution of Figure 4. This is indicated on the figures by "Newton-Raphson initial lambda value: 10.0" and "Newton-Raphson initial lambda value: 18.0" respectively.

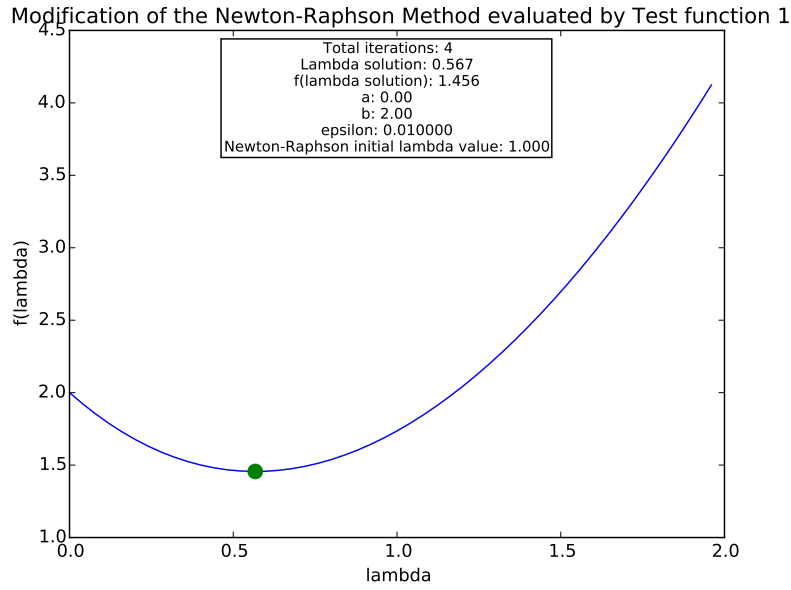## 5.2 Modification of the Newton-Raphson Method

Figure 6: Modification of the Newton-Raphson evaluated by first example function
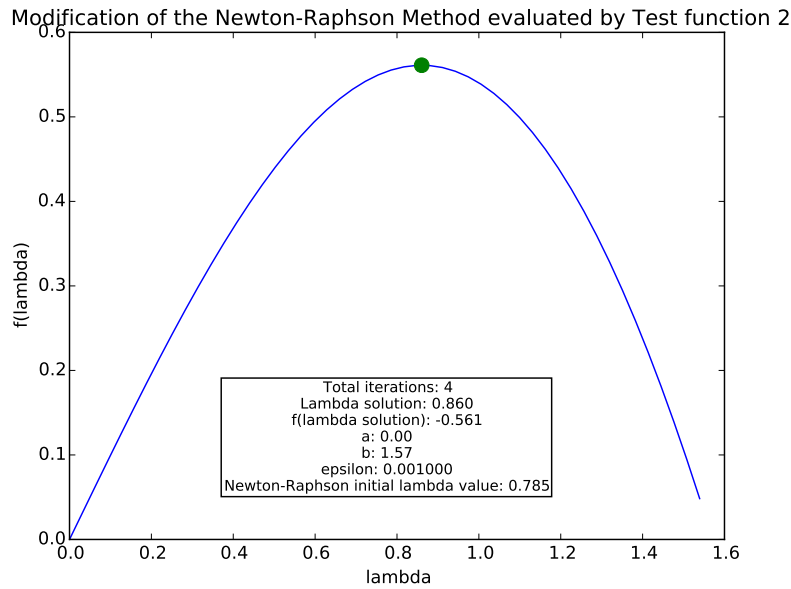

Figure 7: Modification of the Newton-Raphson evaluated by second example function

Figure 6 and Figure 7 show that this algorithm manages to successfully converge near or on the optimal point. Note that a slight increase in number of iterations are required, compared to the unmodified Newton-Raphson method's results shown by Figure 1 and Figure 2.
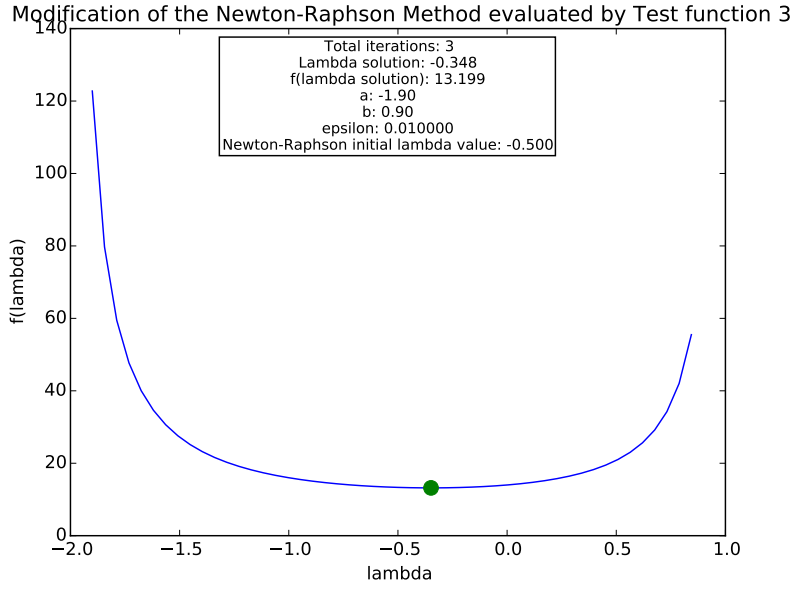
Figure 8: Modification of the Newton-Raphson evaluated by third example function
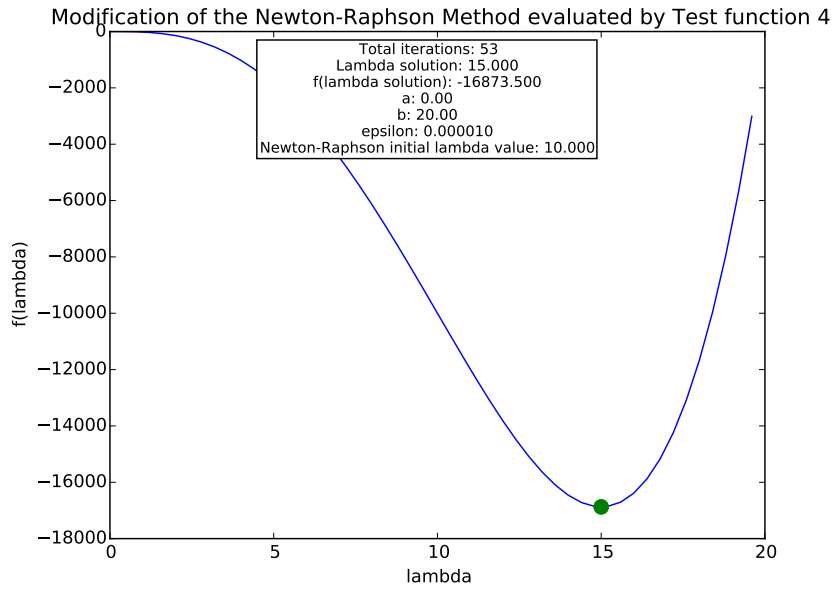

Figure 9: Modification of the Newton-Raphson evaluated by fourth example function

Figure 8 indicates that this algorithm manages to successfully converge. Notice that once again a slight increase of iterations (from 2 to 3) are needed to solve the third test function, as compared to the unmodified Newton-Raphson method shown in Figure 3. Interestingly, Figure 9 converges with the same starting solution assumed by Figure 4 ; "Newton-Raphson initial lambda value: 10.0". This is expected, as the modification of the Newton-Raphson method will more likely converge than the unmodified Newton-Raphson method, even if much more iterations are required. The number of iterations required are 53, compared to Figure 4 where the algorithm diverged outside the boundaries within two iterations. Additionally, it is also not needed to choose a different start solution as was required by the test shown in Figure 5.
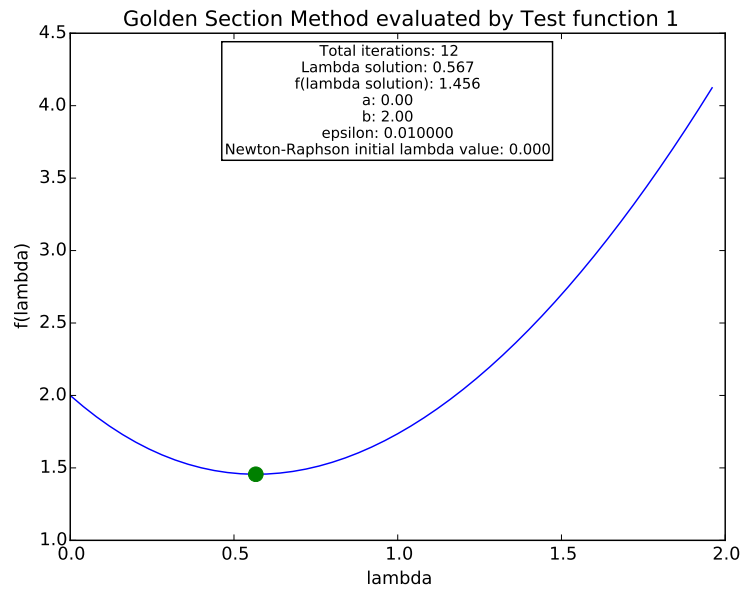
5.3 Golden Section Method

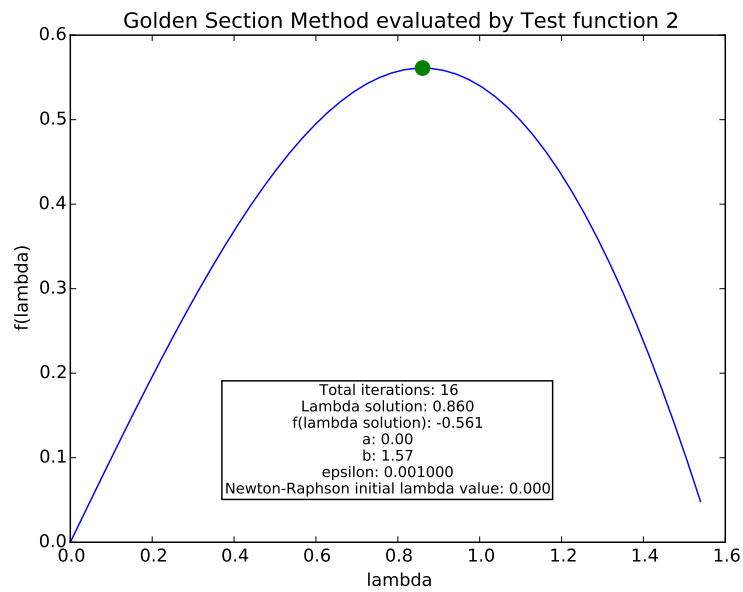Figure 10: Golden Section Method evaluated by first example function

Figure 11: Golden Section Method evaluated by second example function

Figure 10 and Figure 11 show that this algorithm manages to successfully converge near or on the optimal point. Note that much more iterations are required to achieve the equal accuracy of solutions as indicated by Figure 1 and Figure 2.
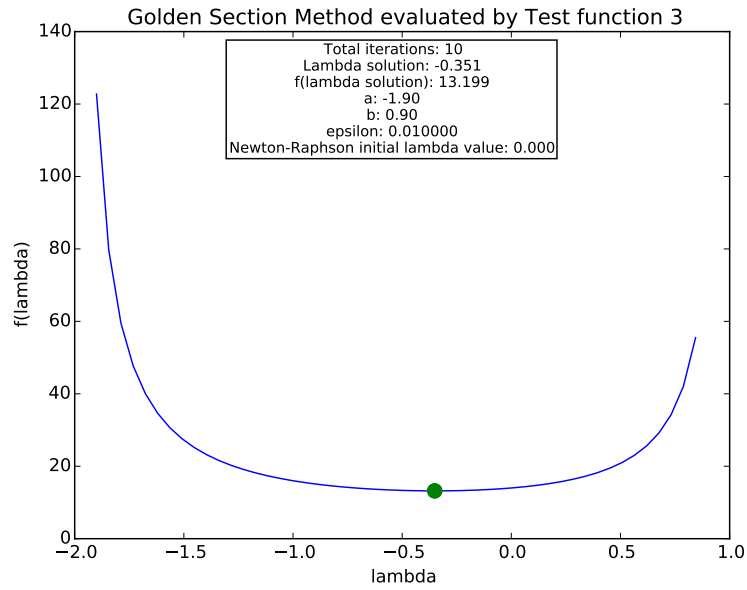
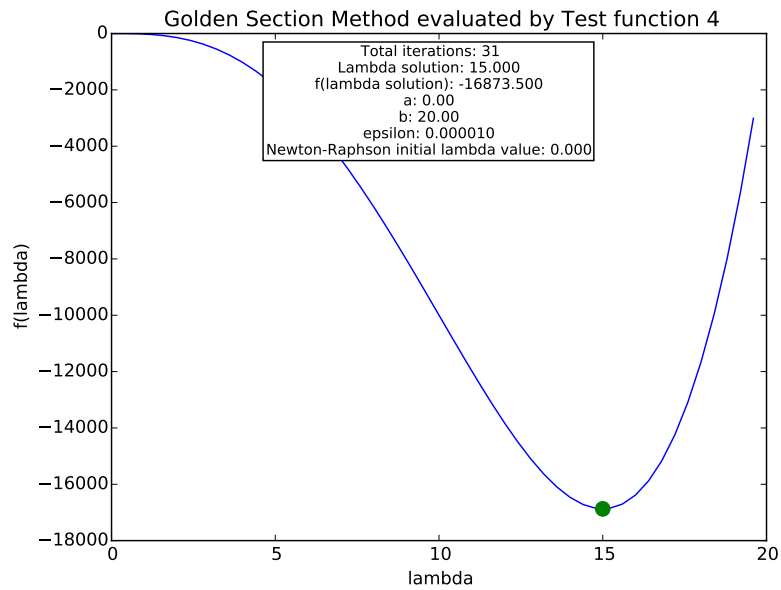Figure 12: Golden Section Method evaluated by third example function

Figure 13: Golden Section Method evaluated by fourth example function

Figure 12 indicates that this algorithm manages to successfully converge near or on the optimal point. It is also shown that the number of iterations are limited to 10 for the third test function. From Figure 13, it is clear that the algorithm did converge to a solution. Notice that 31 iterations are required to obtain the solution, as compared to Figure 5 where only 5 iterations are required (given the starting point chosen for that example).

6   Conclusion

The Newton-Raphson method proved to converge very quickly to the correct solution if the test function is convex at the location of the starting position that is chosen. If divergence occurs, it is possible to select a different starting position and achieve convergence within a relative low number of iterations.

The modification of the Newton-Raphson method shows that it takes slightly longer to converge to the correct solution as compared to the unmodified method. However, in the case where the unmodified Newton-Raphson method would diverge, the modified method would at least find the solution even if much more iterations are required (without needing to choose a different starting position).

The Golden Section method proved to be a very useful tool when first and second order derivative information is not available, and only zero order function values can be obtained. From the test functions that are used in this report, it seems that the Golden Section method is also rather rugged in the sense of converging to the global optimum. It is noted that the Golden Section method in general requires more iterations to obtain the solution, however that this does conclude for it to be a slower method than the Newton-Raphson method. The solve speed would also depend on the specific problem at hand and the time-expensiveness of obtaining zero, first and second order derivative information.

7   Bibliography

1. Snyman J. Practical Mathmematical Optimization. Technical report, Department of Mechanical and Aeronautical Engineering University of Pretoria, Pretoria, South Africa, 2005.