NUMERICAL METHODS FOR SOLVING CUTOFF EQUATION OF COAXIAL CABLES

Wissam Razouki

CSE 3802

BRIEF INTRO/BACKGROUND

Cutoff Equation

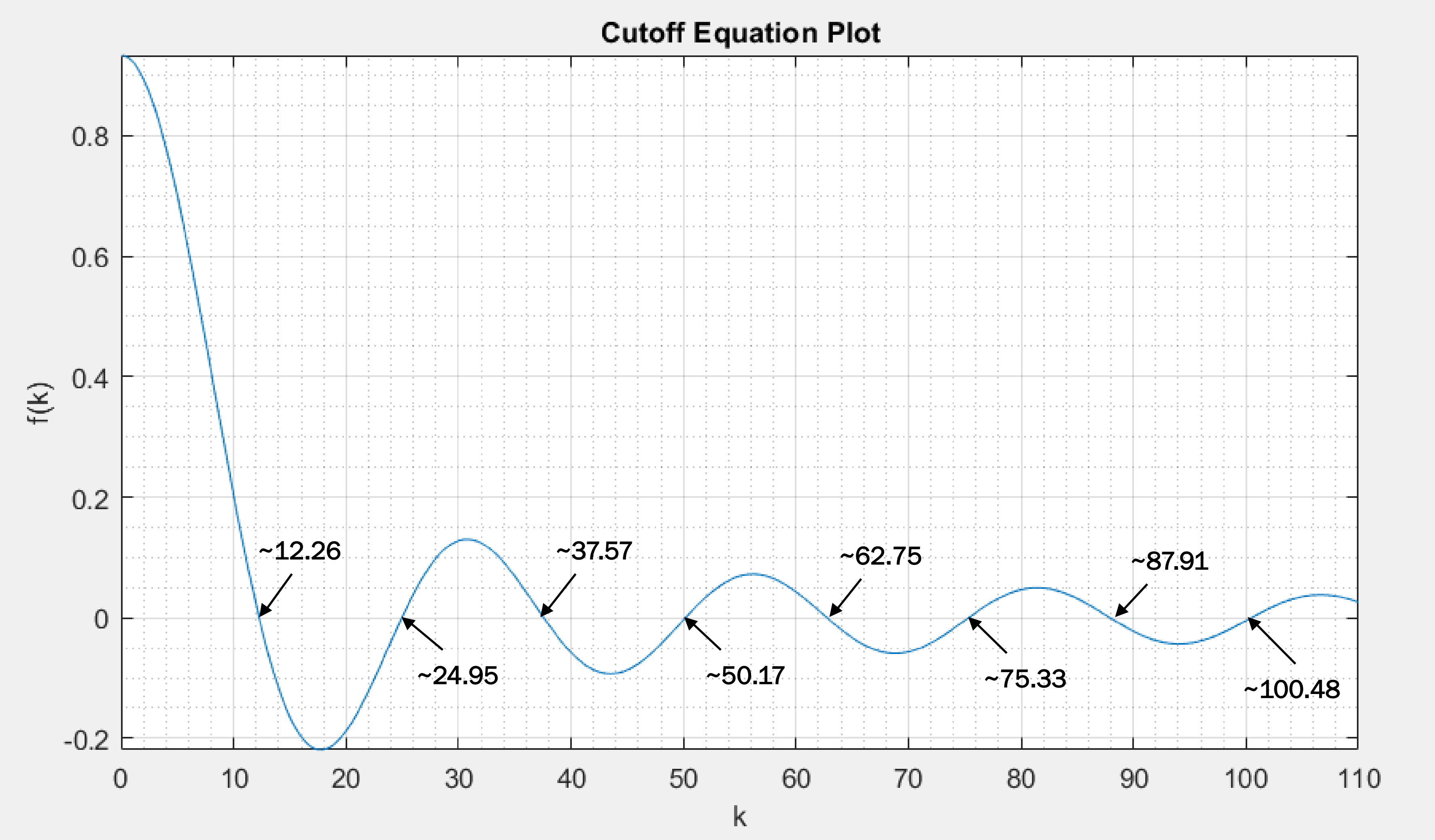
The derived equation for solving for the cutoff wavenumber k (in rad/inch) for a coaxial cable is given as:

$$J_0(ka)Y_0(kb) = J_0(kb)Y_0(ka)$$

Since most computational methods require a function in the form $f(x) = 0$, we can simply implement the above equation as:

$$J_0(ka)Y_0(kb) - J_0(kb)Y_0(ka) = 0$$

Given the physical dimensions $a = 0.075$ and $b = 0.325$ (in inches) and the Bessel functions $J_0(x)$ and $Y_0(x)$, we can solve for k using several methods in MATLAB. The goal is to find the first 4 values such that $k > 0$, and to analyze how each method handles this problem.



Plotting the function in MATLAB gives us a good idea of where the roots for k lie.
This was used to verify the code for the various numerical methods.

Methods Used

- Newton-Raphson
- Secant
- Steffensen's secant
- Bisection
- Monte Carlo bisection
- Regula falsi
- Modified regula falsi ($f(b)/2$)

TESTING

Benchmark – One Initial Guess

A benchmark is required to evaluate each method reliably

- For this benchmark, we will use a maximum of $n = 40$ iterations and a tolerance of $\delta = 0.0001$
- Each method will run 100 times for initial guesses $x_0 \in [1, 100]$ such that x_0 is a positive integer
- If the method fails to converge after 40 iterations or finds a negative solution, we mark the test as a failure
- If a positive solution is found, we mark it as a success
- Performance is based on the average number of iterations needed and the rate of success

Benchmark – Two Initial Guesses

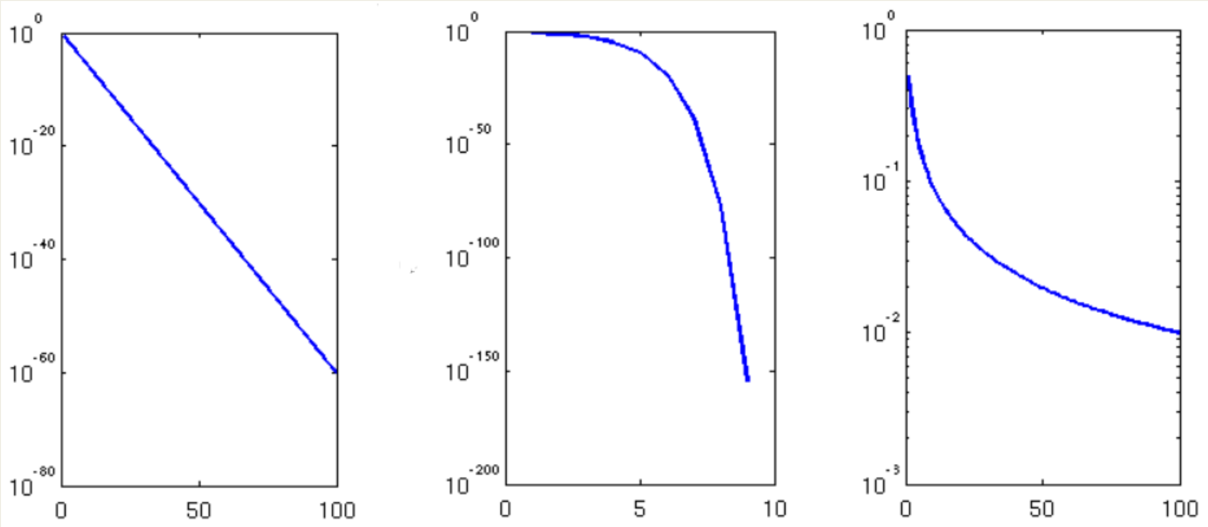
Since several methods require a second initial guess, they will be evaluated differently

- Maximum number of iterations and tolerance are the same
- Each method will run 100 times for initial guesses $x_0 \in [1, 100]$ and $x_1 \in [21, 120]$ such that x_0 and x_1 are both positive integers (i.e. Test #1: $x_0 = 1$ and $x_1 = 21$, Test #2: $x_0 = 2$ and $x_1 = 22$, etc.)
- Conditions for success and failure are the same
- Performance analysis is the same

Rate of Convergence

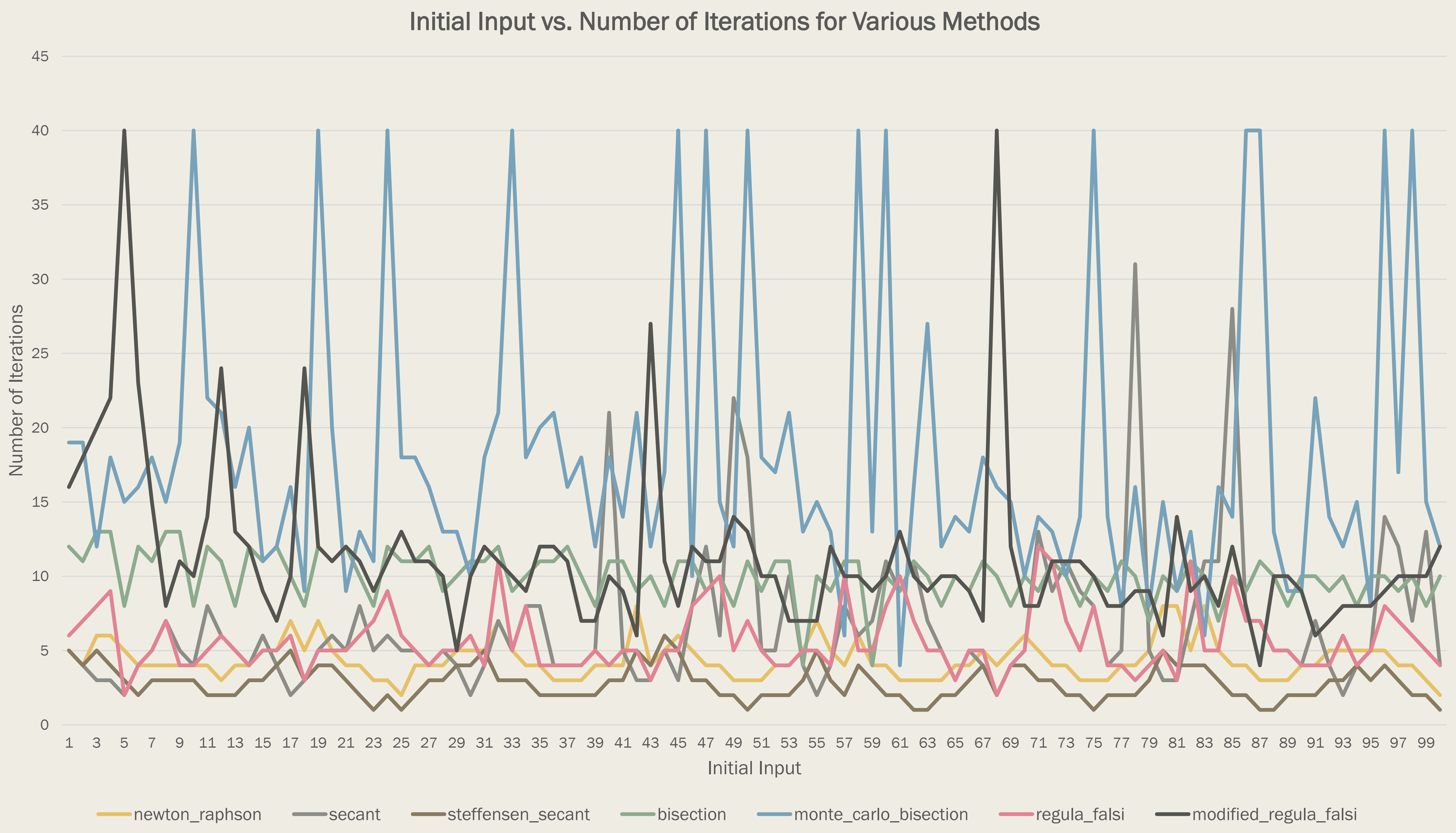
A rate of convergence for each method can be estimated based on the step-to-step error of the output for a certain input.

Plotting this error vs. the iteration in logarithmic form, we can approximate one of three types of convergence: linear, quadratic, or logarithmic.



From Wikipedia

RESULTS



Initial Input vs. Number of Iterations for Various Methods
Number of Iterations
0
5
10
15
20
25
30
35
40
45
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99
Initial Input
newton_raphson
secant
steffensen_secant
bisection
monte_carlo_bisection
regula_falsi
modified_regula_falsi

Test Data

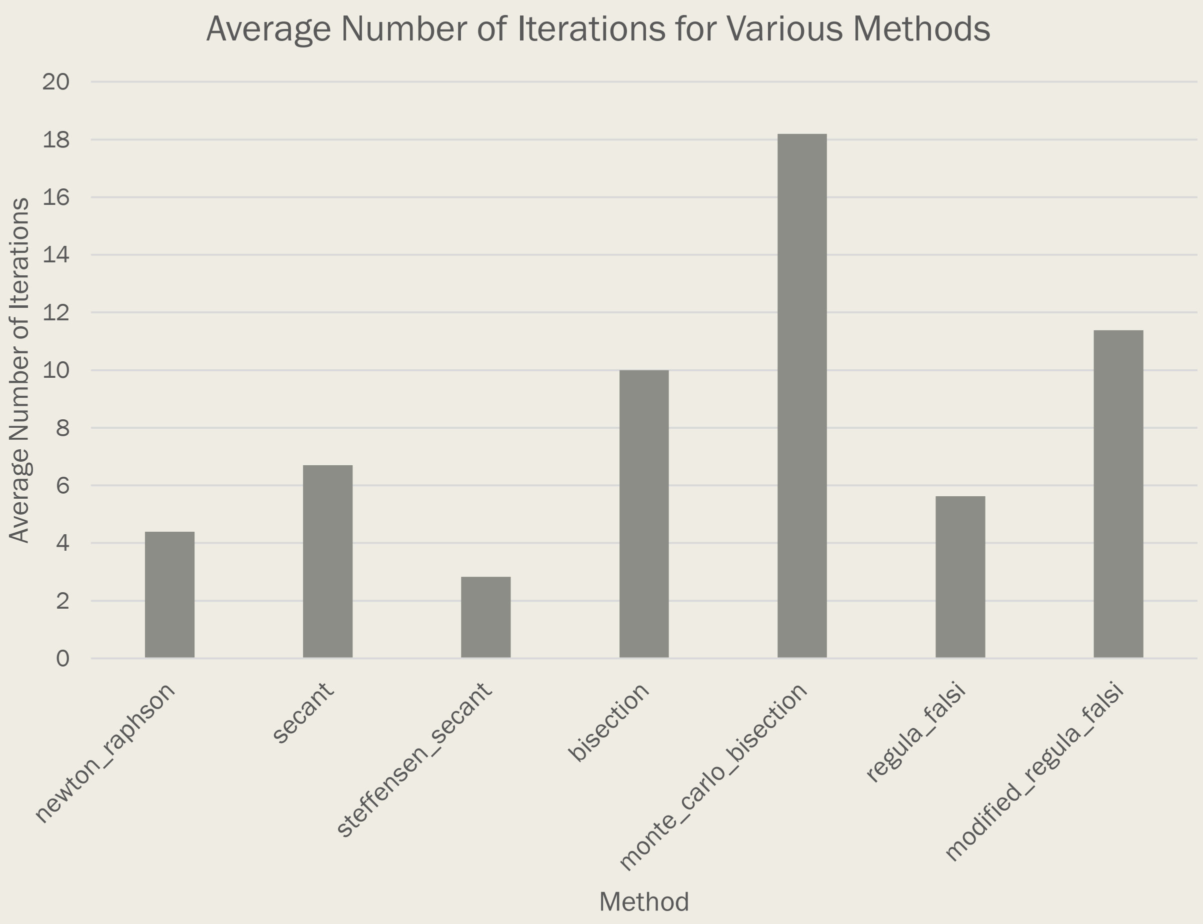
The data above shows how many iterations it took for each method to find a solution based on the (first) initial input.

Notable points:

- Monte Carlo bisection and modified regula falsi are the only methods that reach the maximum number of iterations (and thus, fail to find a solution)
  - *This is due to the unstable nature of these methods*
- Steffensen's secant and Newton-Raphson seem to be the best performing methods

Performance Data

Method	Average number of iterations	Success rate
newton_raphson	4.39	96%
secant	6.7	90%
steffensen_secant	2.83	95%
bisection	10	100%
monte_carlo_bisection	18.19	86%
regula_falsi	5.62	96%
modified_regula_falsi	11.38	95%



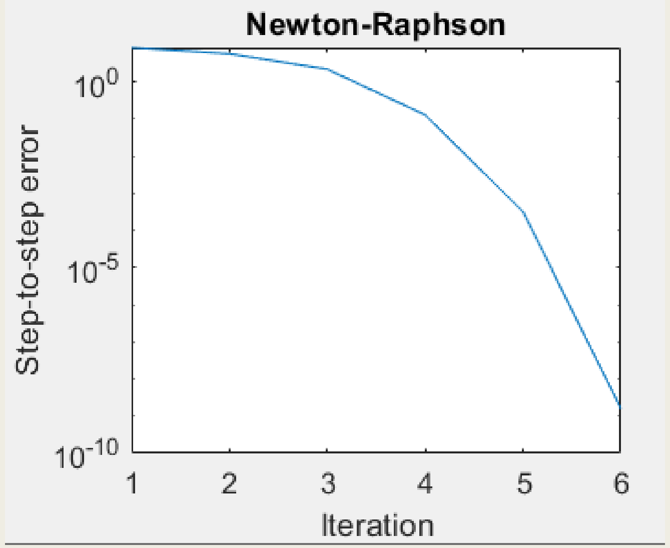
Performance Summary

The data above aggregates the number of iterations and the success rate for each method.

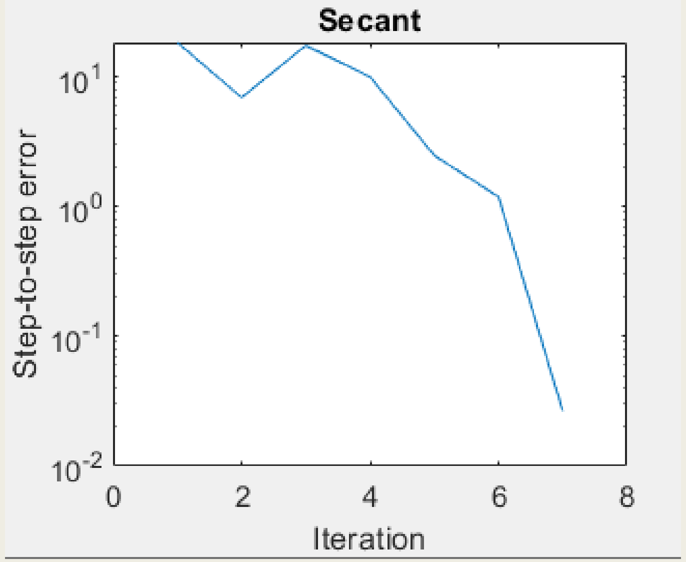
According to the data, we can conclude that:

- Steffensen's secant is the fastest method for finding roots in the least amount of iterations
- Bisection is the most reliable method as it always converges to a solution
- Monte Carlo bisection is the worst method both in terms of speed and reliability

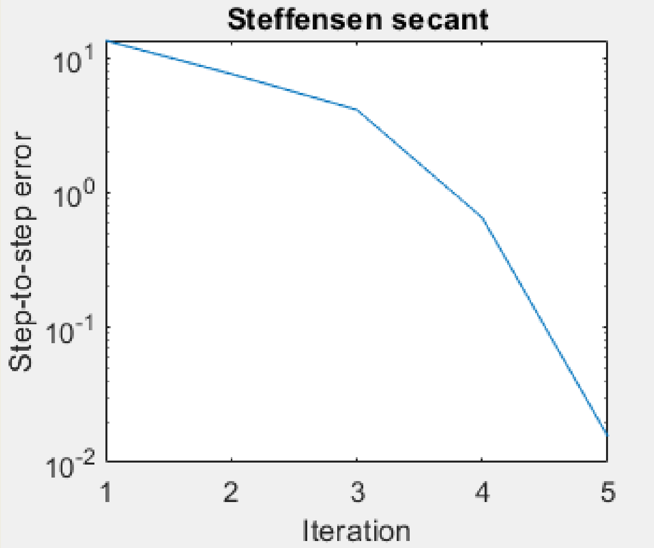
Estimated Rates of Convergence



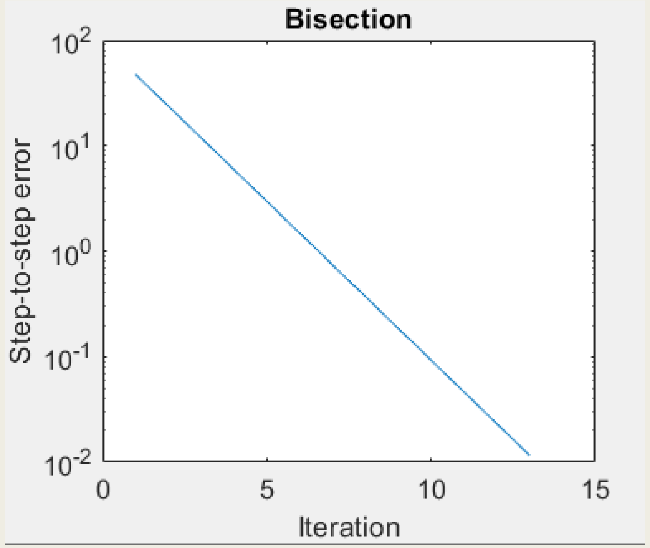
Quadratic



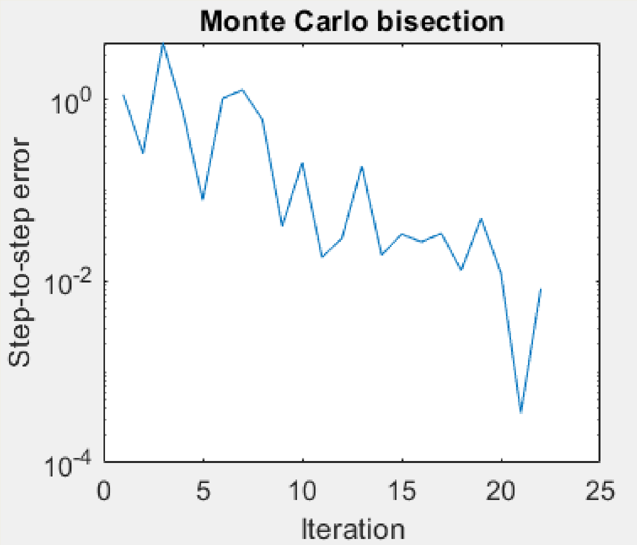
Almost Quadratic



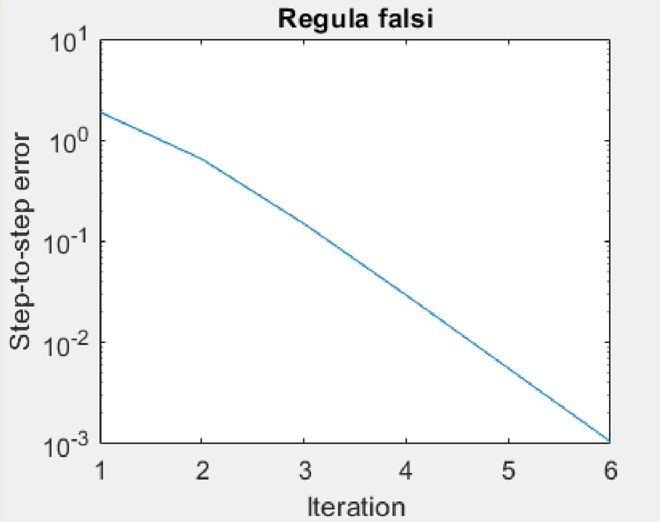
Quadratic



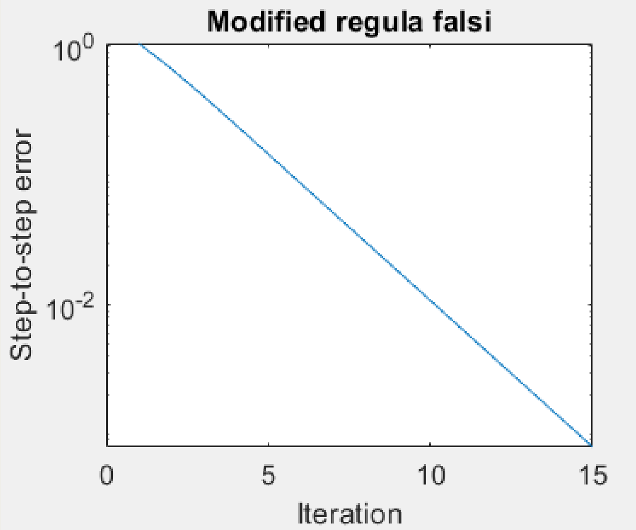
Linear



Random (pseudolinear)



Almost linear



Almost linear

Analysis and Conclusions

According to these results, we can conclude that:

- Steffensen’s secant is the fastest algorithm
- Bisection is the most reliable algorithm
  - *These methods are best suited for this problem*
- Monte Carlo bisection and modified regula falsi are unstable, slow and unreliable
  - *These methods are not suited for this problem*