Sawyer Theis

Project 2

Contents

[Introduction 1](#_Toc193384496)

[Root Calculation Methods 1](#_Toc193384497)

[Bisection Method 1](#_Toc193384498)

[Newton Method 1](#_Toc193384499)

[Secant Method 1](#_Toc193384500)

[Method Termination 2](#_Toc193384501)

[Convergence Analysis 2](#_Toc193384502)

[Results 2](#_Toc193384503)

[Function 1 2](#_Toc193384504)

[Function 2 3](#_Toc193384505)

[Function 3 3](#_Toc193384506)

[Conclusion 4](#_Toc193384507)

# Introduction

This project tested three iterative methods of finding the root of a function and compared their convergence rates and effectiveness. These methods were the bisection, newton, and secant method. Their effectiveness was compared by how many iterations they took to reach a root. The convergence rate was found to figure out how quickly each method was approaching the root. Each method was tested on three functions where they were set to each converge on the same root.

# Root Calculation Methods

## Bisection Method

The bisection method works by repeatedly dividing an interval in half and selecting the subinterval where the root lies. It requires the function f(x) to be continuous on the interval ([a, b]) and f(a)\*f(b) < 0. The bisection method converges linearly.

The steps are as follows:

1. Compute the midpoint .
2. Evaluate f(c):
   1. If f(c) = 0, c is the root.
   2. If f(a) \* f(c) < 0, set (b = c).
   3. Otherwise, set (a = c).
3. Repeat until the interval ([a, b]) is sufficiently small.

## Newton Method

The Newton method uses the derivative of the function to approximate the root. Starting with an initial guess x0, the method iteratively refines the estimate using the formula:

The method converges quadratically if the initial guess is close to the root.

## Secant Method

The Secant method is a root-finding technique similar to the Newton method but does not require the derivative of the function. Instead, it approximates the derivative using a secant line through two points. Starting with two initial guesses x0 and x1, the method iteratively refines the estimate using the formula:

The Secant method converges faster than the bisection method but slower than the Newton method. Its convergence rate is approximately the golden ratio (1.618).

# Method Termination

Each method stopped calculating new steps when the next step they calculated is the same as the previous step. This termination criterion worked well because a value that is the same as the previous step would always result in a loop for each method. Every method would eventually reach this criterion by getting stuck in a loop or by running out of precision. This criterion was chosen because it provided theoretically the most accurate possible root. A hard stop was also set at 1000 iterations, but no method ever reached that many iterations.

# Convergence Analysis

Convergence rate was found by finding the median lr (logarithmic rate) for each method for each function. Lr represents how quickly a method approaches the root. If lr >= 2 it converges quadratically, 2 > lr > 1 is superlinear, 1 = lr is linear, and 1 > lr it is sublinear. The median lr value was used to give the most accurate because there were often outlier data points that skewed the rate. Additionally, the first two steps were excluded because they were often outliers especially for the newton and secant method.

Lr was calculated by finding each e, where e was the difference between xn, step n, and r, the real root. Then the rate at which e changed was found using equation 1.

# Results

For every function, newton’s method started at -0.353 and secant method started at -0.567 for xo and -0.353 for x1. The bisection method had another function to find 2 nearby points that are shown in the program since they often changed. These points weren’t chosen for any reason other than they converged to the same roots for each function.

The results for many things in the tables below have vastly more precision but are rounded to 5 decimal places.

## Function 1

The first function was f1(x) = x2 −4 sin(x). It has a root at 0 which all the methods were aimed at finding.

Table 1. Results of each method as they approached the root of function 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | root | error | iterations |
| Bisection | 4.08432e-163 | 4.08432e-163 | 589 |
| Newton | 0.0 | 0.0 | 5 |
| Secant | 0.0 | 0.0 | 8 |

Table 2. Convergence analysis of function 1

|  |  |  |
| --- | --- | --- |
|  | Convergence | Log Rate |
| Bisection | Superlinear | 1.00111 |
| newton | Quadratic | 2.06594 |
| Secant | Superlinear | 1.65373 |
|  |  |  |

Graph 1. Steps of the bisection method for function 1

A graph with a line graph

AI-generated content may be incorrect.

Graph 2. Steps of newton’s method for function 1

A graph with a line going up

AI-generated content may be incorrect.

Graph 3. Steps of the secant method for function 1

A graph with a line going up

AI-generated content may be incorrect.

Function 2

The first function was f1(x) = x2 − 1. It has a root at -1 which all the methods were aimed at finding.

Table 3. Results of each method as they approached the root of function 2

|  |  |  |  |
| --- | --- | --- | --- |
|  | root | error | iterations |
| Bisection | -0.99999 | 2.22045e-16 | 49 |
| Newton | -1.0 | 0.0 | 6 |
| Secant | -1.0 | 0.0 | 8 |

Table 4. Convergence analysis of function 2

|  |  |  |
| --- | --- | --- |
|  | Convergence | Log Rate |
| Bisection | Superlinear | 1.00440 |
| newton | Quadratic | 2.09829 |
| Secant | Superlinear | 1.65985 |

Graph 4. Steps of the bisection method for function 2

A graph with a line going up

AI-generated content may be incorrect.

Graph 5. Steps of newton’s method for function 2

A graph with a line

AI-generated content may be incorrect.

Graph 6. Steps of the secant method for function 2

A graph with a line going up

AI-generated content may be incorrect.

Function 3

The first function was f1(x) = x3 – 3x2 + 3x -1 It has a root at 1 which all the methods were aimed at finding.

Table 5. Results of each method as they approached the root of function 3

|  |  |  |  |
| --- | --- | --- | --- |
|  | root | error | iterations |
| Bisection | 1.00003 | 2.92969e-5 | 49 |
| Newton | 1.00000 | 1.52472e-6 | 32 |
| Secant | 0.99999 | 5.37097e-6 | 45 |

Table 6. Convergence analysis of function 3

|  |  |  |
| --- | --- | --- |
|  | Convergence | Log Rate |
| Bisection | Sublinear | 0.99999 |
| newton | Superlinear | 1.06555 |
| Secant | Superlinear | 1.04433 |

Graph 7. Steps of the bisection method for function 3

A graph with a line going up

AI-generated content may be incorrect.

Graph 8. Steps of newton’s method for function 3

A graph with a dotted line

AI-generated content may be incorrect.

Graph 9. Steps of the secant method for function 3

A graph with a dotted line

AI-generated content may be incorrect.

# Conclusion

Newton’s method was by far the most efficient method. For every function tested it used the lowest iterations and had the highest convergence rate. The secant method followed closely behind with slightly more iterations for each function and a lower convergence rate. The bisection method handily took last place with far more iterations even though the numbers it started with were much closer to the roots than the other two methods. It also always held the lowest convergence rate.

The convergence rate of each method was roughly what was expected for the first two functions. In Tables 2 and 4 the log rate came out to be approximately 1 for bisection, 2 for newtons, and 1.618 for the secant method. The actual log rates were slightly above or below these numbers but that is because those convergence rates are only true under ideal conditions and there were too few data points. These results confirm that the bisection method has linear convergence, newton has quadratic convergence, and the secant method has superlinear convergence. However, in table 6 all the log rates became approximately 1 for function 3. This rate reduces every function to linear convergence. This reduction in convergence rates may have occurred because function 3 is a 3rd order polynomial, and as such it should have 3 roots. Expect, function 3 has only one root at 1. This collision of all 3 roots could have slowed down the newtons and secant methods.

The termination criterion chosen where a method would terminate if it produced an identical number worked splendidly. It provides enough steps to produce good results for the convergence analysis. However, in table 1 this termination criterion probably caused the iterations for the bisection method to shoot way up. While normally the bisection method would run out of mantissa bits then terminate, since the root was 0 it didn’t run out of mantissa bits and instead used its exponent bits to get more precise. This is a quirk of this termination criterion and could be fixed by putting a more reasonable hard limit on the number of iterations like 100 instead of 1000.

Newton’s method is shown to have the fastest convergence and use the least number of iterations. It does have a major flaw that isn’t addressed in this project. With these 3 functions the derivative was easy to find. If the derivative is difficult or impossible to calculate it may make using this method impossible. The secant method in this project was still very fast while not suffering from this flaw, which makes it much more viable for practical applications.