|  |
| --- |
|  |
| Project 1 |
| CS 301 – Amar Raheja |

|  |
| --- |
| Joshua Camacho  2-13-2017 |

Numerical Root Approximation

Joshua Camacho

California State Polytechnic University, Pomona

Numerical Methods

1. Overview

Frequently, the solution to a scientific problem is a number about which we have little information other than that it satisfies some equation. Since every equation can be written so that a function stands on one side and zero on the other, the desired number must be a zero of the function. Thus, if we possess an arsenal of methods for locating zeros of functions, we shall be able to solve such problems.

1. Assignment

The assignment was to write programs for the following root approximation methods

* Bisection
* Newton-Raphson
* Secant
* False-Position
* Modified Secant

Use those programs to evaluate the roots for the following functions

1. f(x) = 2x3– 11.7x2 + 17.7x -5
2. f(x) = e-x -x

Lastly, present graphs of the functions, graphs of the error evaluations, and discuss the results of each method.

1. Function Graphs

Looking at the graphs of the functions can help determine where the roots are located.

Amar Raheja

Professor of Computer Science

California State Polytechnic University, Pomona

f(x) = 2x3– 11.7x2 + 17.7x -5

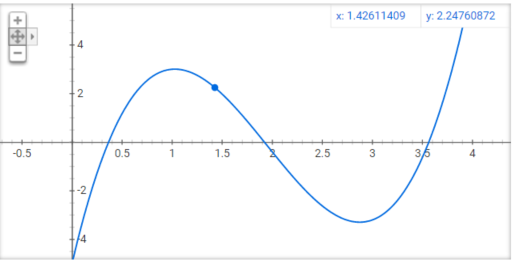


Figure : Function A

From the graph, there appears to be 3 roots. One on the interval (0,0.5), (1.5,2), and (3.5,4).

f(x) = e-x -x

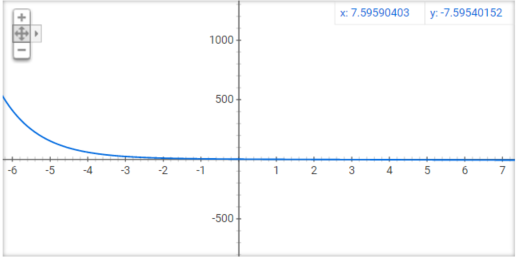


Figure : Function B

This graph was harder to see, but upon zoom there appears to be a root between (0,1). The true root for this function was also given, at x=0.56714329.

1. Root Evaluation

# Function A) f(x) = 2x3– 11.7x2 + 17.7x -5

This function had 3 roots on the intervals (0,0.5), (1.5,2), and (3.5,4)

## Bisection Method – Function A

**Bisection Method Root 1, using a=0, b=1**

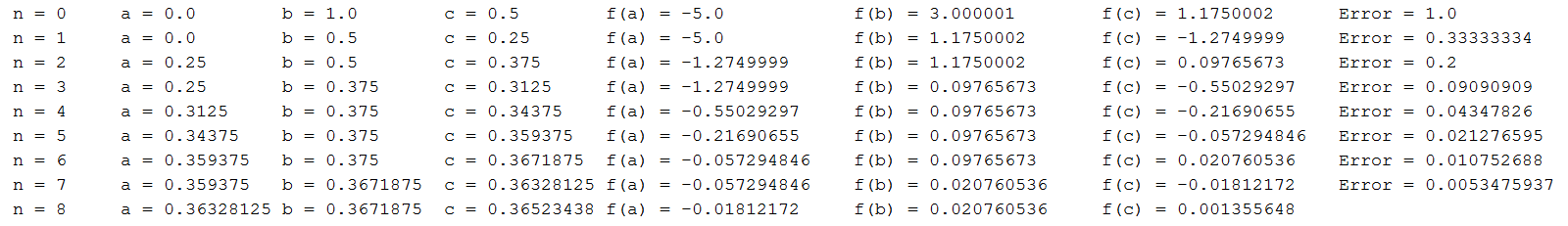


Figure : Function A, Bisection Method, Root 1

Method converges at c = 0.36523438 after 8 iterations

**Bisection Method Root 2, using a=1, b=2**

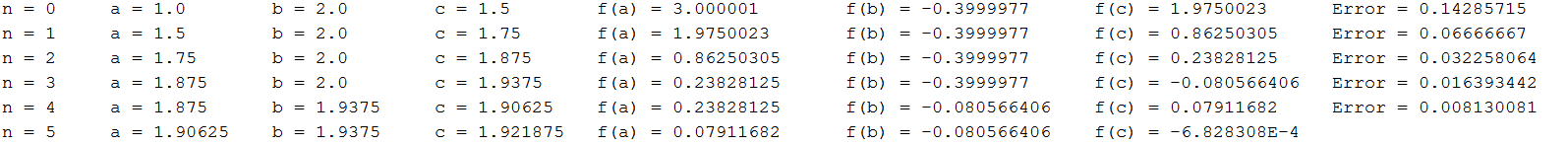


Figure : Function A, Bisection Method, Root 2

Method converges at c = 1.921875 after 5 iterations

**Root 3, using a=3, b=4**

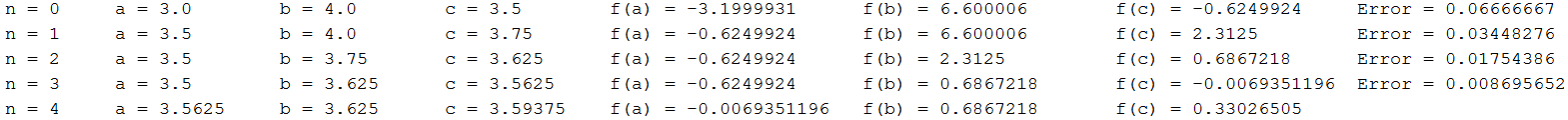


Figure : Function A, Bisection Method, Root 3

Method converges at c = 3.59375 after 4 iterations

## Newton Method – Function A

**Newton Method Root 1, using x=1**

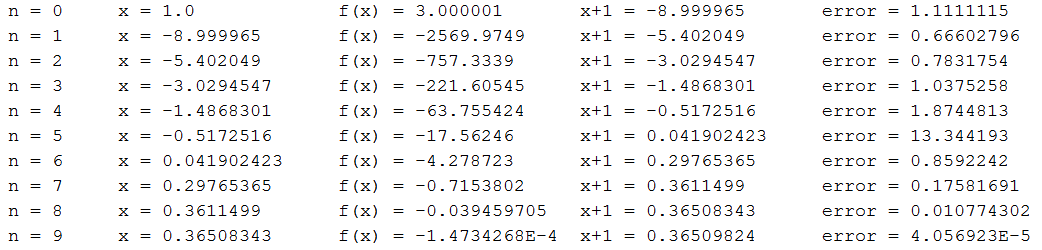


Figure : Function A, Newton Method, Root 1

Method converges at x = 0.36509824 after 10 iterations

**Newton Method Root 2, using x=2**



Figure : Function A, Newton Method, Root 2

Method converges at x = 1.9217416 after 2 iterations

**Newton Method Root 3, using x=3**

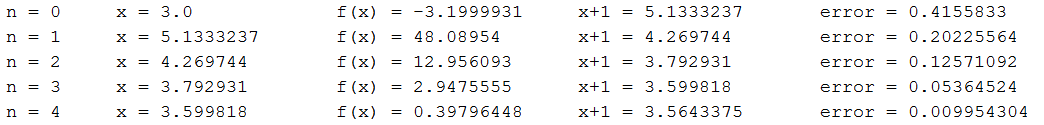
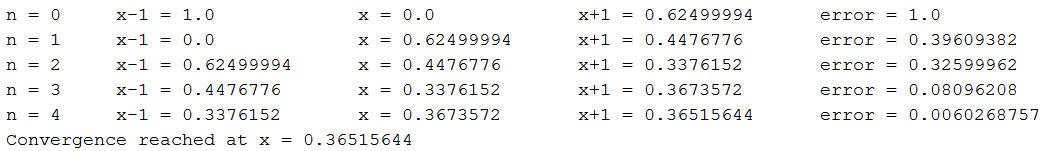


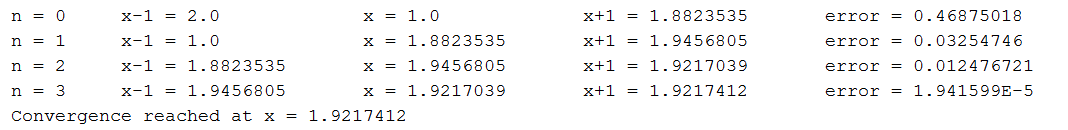
Figure : Function A, Newton Method, Root 3

Method converges at x = 3.5643375 after 5 iterations

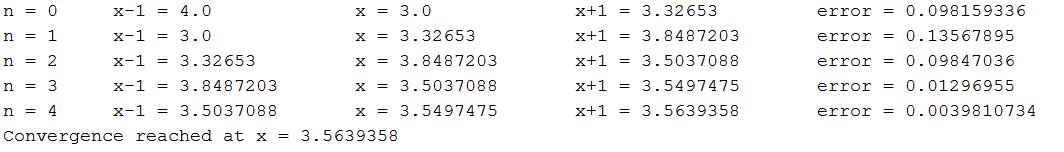
## Secant Method – Function A

**Secant Method Root 1, using x0=0, x=1**

**Secant Method Root 2, using x0=1, x=2**

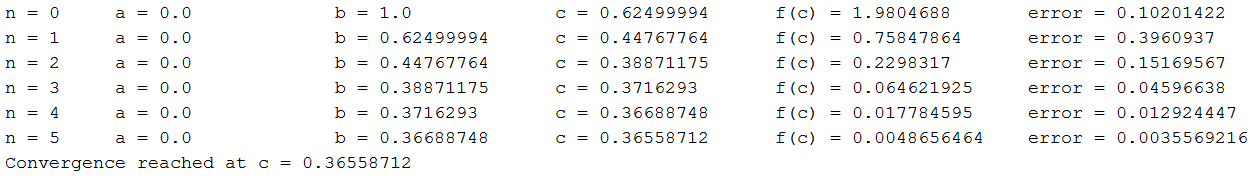


**Secant Method Root 3, using x0=3, x=4**

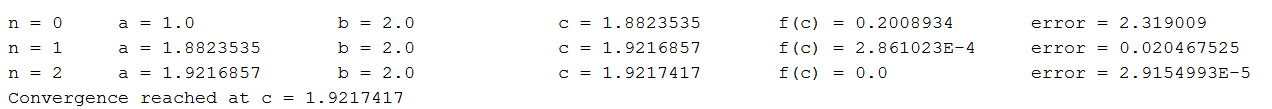


## False Position Method – Function A

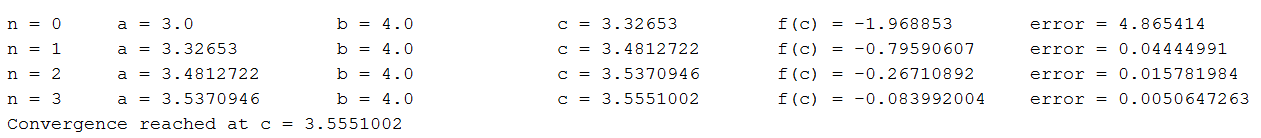
**False Position Method Root 1, using a=0, b=1**



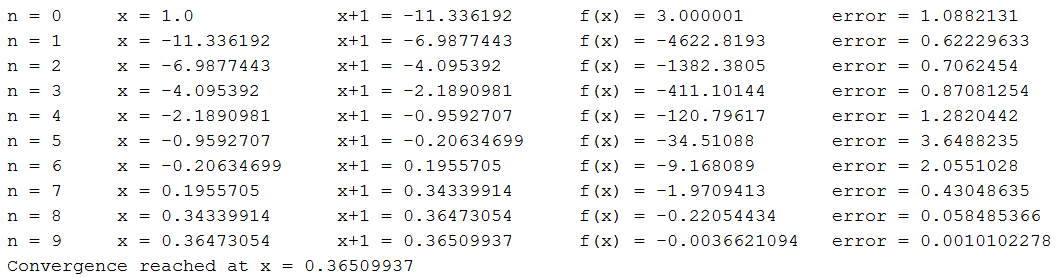
**False Position Method Root 2, using a=1, b=2**



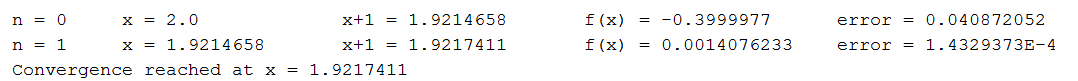
**False Position Method Root 3, using a=3, b=4**



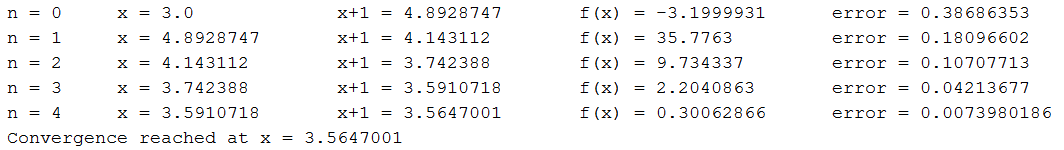
## Modified Secant Method – Function A

**Modified Secant Method Root 1, using x=1, delta=0.01**

**Modified Secant Method Root 2, using x=2, delta=0.01**



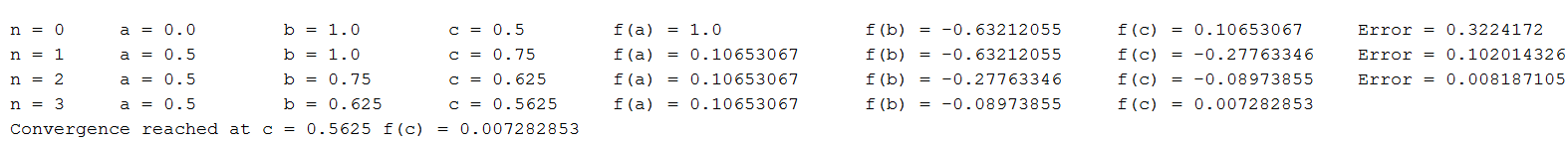
**Modified Secant Method Root 3, using x=3, delta=0.01**



# Function B) f(x) = e-x -x

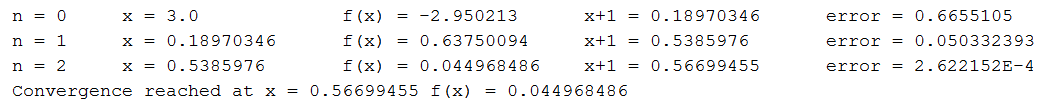
## Bisection Method – Function B

Using a=0, b=1



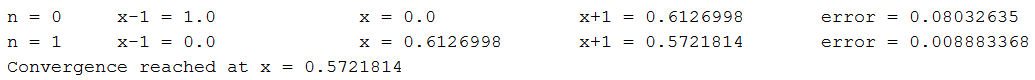
## Newton Method – Function B

Using x=3.0



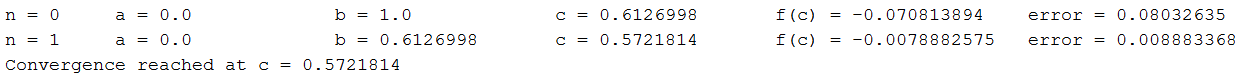
## Secant Method – Function B

Using x0=1.0, x=0



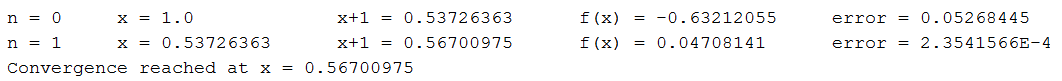
## False Position Method – Function B

Using a=0, b=1



## Modified Secant Method – Function B

Using x=1, delta=0.01



1. Error Graphs

These graphs show the true relative error for the root of function B. Endpoints 0,1 were used for the bisection, secant, and false position methods. 1 was used for newton method. X = 1 and delta = 0.01 was used for the modified secant method.

These graphs show the approximate error for root 1 of function A. Endpoints 0,1 were used for the bisection, secant, and false position methods. 1 was used for newton method. X = 1 and delta = 0.01 was used for the modified secant method.

These show the approximate % error for Function A, root 2. Endpoints for bisection, secant, and false position were 1 and 2. Newton method used 2 and modified secant used 2 with delta=0.01.

These show the approximate % error for Function A, root 2. Endpoints for bisection, secant, and false position were 3 and 4. Newton method used 3 and modified secant used 3 with delta=0.01.

VI - Discussion

# Root Summary

Function A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Bisection | Newton | Secant | False Position | Modified Secant |
| Root 1 | 0.36523438 | 0.36509824 | 0.365156 | 0.36558712 | 0.36509937 |
| Root 2 | 1.921875 | 1.9217416 | 1.927412 | 1.9217417 | 1.9217411 |
| Root 3 | 3.59375 | 3.5643375 | 3.56395 | 3.5551002 | 3.5647001 |

Function B

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Bisection | Newton | Secant | False Position | Modified Secant |
| Root | 0.56726074 | 0.566987 | 0.5721814 | 0.5721814 | 0.56700975 |

On function A- Newton method and Modified took the longest to converge for the first point. Newton method was an outlier in approximate error, at one point shooting up to a very high error % before reaching convergence.

On function B – Modified Secant, Secant, and Newton method all converged extremely fast, with bisection method being the slowest to converge.

I used the float data types for the methods, since the significant digits of error can fit into a float data type.