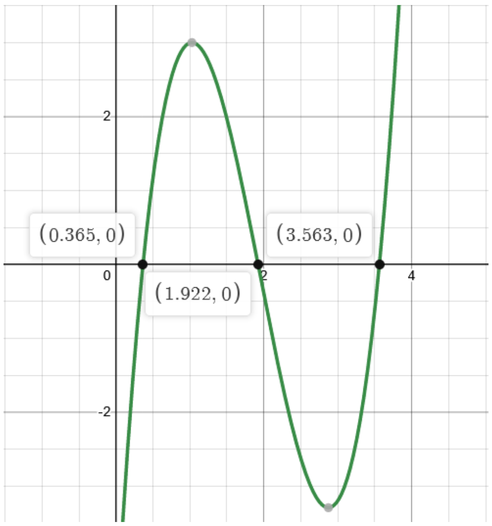
Notes

- Joseph Siwiecki, CS 3010.01, 11/19/22
- The starting points for all tests on function A for the bisection method, secant method, and false position method are all [0.0, 1.0], [3.0, 4.0], and [1.0, 3.0]. These intervals are meant to converge on root 0.365, root 1.922, and root 3.563 respectively. For the newton-raphson method for function A, the starting points are [0.0], [2.0], and [3.0]. These starting points are meant to converge on root 0.365, root 1.922, and root 3.563 respectively.
- The starting points for all tests on function B are all [125.0, 130.0], except for the newton-raphson method, which starts at [127.0]. All these intervals and starting points are meant to converge upon root 126.632.
- Most methods work fine, Newton-Raphson has a lot of problems searching for roots on function A, however, probably because there are 3 roots so it has difficulty discerning which one to converge upon. This behavior leads it to keep iterating until it reaches max iterations, unfortunately. The other methods do not have this issue, or at least not as consistently. The only other times I had this method appear was when I was using a closed-root finding method but placed multiple roots within the given interval.
- Overall, the False Position method appears to find the roots the quickest. It either tied with other methods' iteration count or was1 iteration slower than one other method, that being the Bisection Method for function B.
- It seems that function B was a lot easier for each method to calculate the root for as well, since most of the methods struggled to find roots for function A, since there were multiple roots rather than just one. Only one root in function B to solve for made things a lot simpler for the calculations, at least.
- The data types for all calculations are doubles to achieve the highest accuracy possible without employing more cumbersome methods, such as using something else like BigDecimal from Java's Math library.
- On some of the graphs, relative error is 0 or nearly 0, so the final iteration points may not seem like they appear or not, but if there is no point at a specific point of iteration of the graph, it likely means that the approximate relative error at that point is just 0 and iteration stopped there.

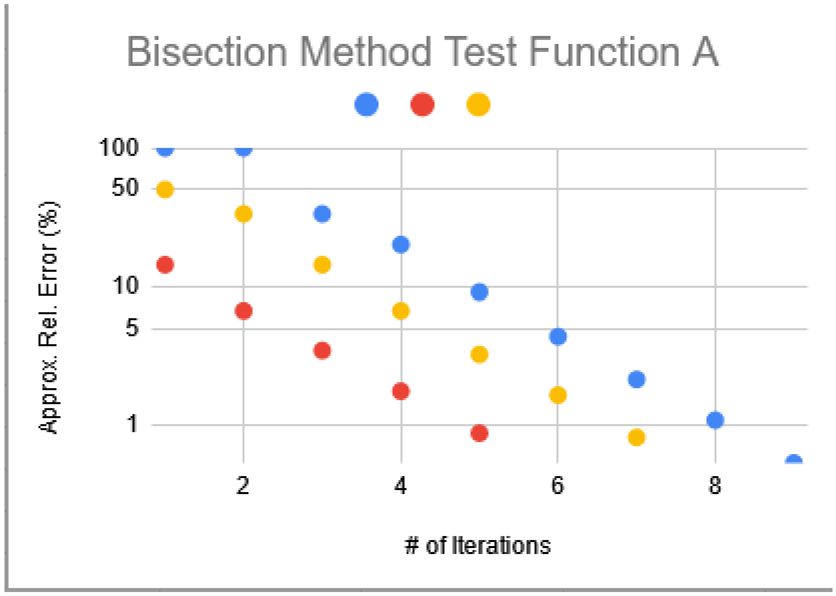
$f(x) = 2x^3 - 11.7x^2 + 17.7x - 5$

Graph



Bisection Method Tests

Approx. Rel. Error (%) vs # of Iterations Graph



Test 1

```
--- Bisection Method with inputs 0.0 and 1.0 for Function A ---

Iteration: 1
 x = 0.5000 with relative error: 100.00 with true error: 36.99

Iteration: 2
 x = 0.2500 with relative error: 100.00 with true error: 31.51

Iteration: 3
 x = 0.3750 with relative error: 33.33 with true error: 2.74

Iteration: 4
 x = 0.3125 with relative error: 20.00 with true error: 14.38

Iteration: 5
 x = 0.3438 with relative error: 9.09 with true error: 5.82

Iteration: 6
 x = 0.3594 with relative error: 4.35 with true error: 1.54

Iteration: 7
 x = 0.3672 with relative error: 2.13 with true error: 0.60

Iteration: 8
 x = 0.3633 with relative error: 1.08 with true error: 0.47

Iteration: 9
 x = 0.3652 with relative error: 0.53 with true error: 0.06

As of 9 iterations, x = 0.3652
------------------
```

Test 2

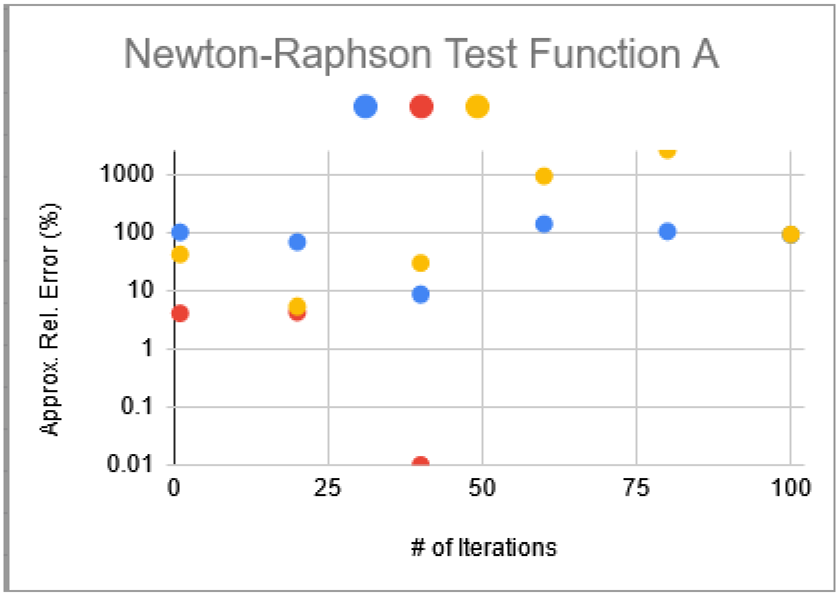
```
--- Bisection Method with inputs 3.0 and 4.0 for Function A ---

Iteration: 1
 x = 3.5000 with relative error: 14.29 with true error: 1.77

Iteration: 2
 x = 3.7500 with relative error: 6.67 with true error: 5.25

Iteration: 3
 x = 3.6250 with relative error: 3.45 with true error: 1.74

Iteration: 4
 x = 3.5625 with relative error: 1.75 with true error: 0.01

Iteration: 5
 x = 3.5938 with relative error: 0.87 with true error: 0.86

As of 5 iterations, x = 3.5938
------------------
```

Test 3

```
--- Bisection Method with inputs 1.0 and 3.0 for Function A ---

Iteration: 1
 x = 2.0000 with relative error: 50.00 with true error: 4.06

Iteration: 2
 x = 1.5000 with relative error: 33.33 with true error: 21.96

Iteration: 3
 x = 1.7500 with relative error: 14.29 with true error: 8.95

Iteration: 4
 x = 1.8750 with relative error: 6.67 with true error: 2.45

Iteration: 5
 x = 1.9375 with relative error: 3.23 with true error: 0.81

Iteration: 6
 x = 1.9063 with relative error: 1.64 with true error: 0.82

Iteration: 7
 x = 1.9219 with relative error: 0.81 with true error: 0.01

As of 7 iterations, x = 1.9219
------------------
```

Newton-Raphson Method Tests

Approx. Rel. Error (%) vs # of Iterations Graph



Test 1

Part 1

```
--- Newton-Raphson Method with input 0.0 for Function A ---

Iteration: 1
 x = 0.2825 with relative error: 100.00 with true error: 22.61

Iteration: 2
 x = 0.5650 with relative error: 50.00 with true error: 54.79

Iteration: 3
 x = 0.6418 with relative error: 11.97 with true error: 75.83

Iteration: 4
 x = 0.3875 with relative error: 65.62 with true error: 6.16

Iteration: 5
 x = -0.0140 with relative error: 2867.61 with true error: 103.84

Iteration: 6
 x = -0.0369 with relative error: 62.05 with true error: 110.11

Iteration: 7
 x = 0.2543 with relative error: 114.51 with true error: 30.32

Iteration: 8
 x = 0.5596 with relative error: 54.55 with true error: 53.31

Iteration: 9
 x = 0.6603 with relative error: 15.25 with true error: 80.90

Iteration: 10
 x = 0.4149 with relative error: 59.15 with true error: 13.67

Iteration: 11
 x = -0.0295 with relative error: 1508.03 with true error: 108.07
```

Part 2

```
Iteration: 90
 x = 1.1946 with relative error: 0.42 with true error: 227.29

Iteration: 91
 x = 2.8352 with relative error: 57.87 with true error: 676.76

Iteration: 92
 x = 4.5247 with relative error: 37.34 with true error: 1139.64

Iteration: 93
 x = -3.4151 with relative error: 232.49 with true error: 1035.64

Iteration: 94
 x = -4.0158 with relative error: 14.96 with true error: 1200.23

Iteration: 95
 x = -2.3358 with relative error: 71.93 with true error: 739.94

Iteration: 96
 x = -0.4441 with relative error: 425.99 with true error: 221.66

Iteration: 97
 x = 0.8468 with relative error: 152.44 with true error: 132.01

Iteration: 98
 x = 1.3709 with relative error: 38.23 with true error: 275.59

Iteration: 99
 x = 0.0846 with relative error: 1521.10 with true error: 76.83

Iteration: 100
 x = 0.8674 with relative error: 90.25 with true error: 137.64

As of 100 iterations, x = 0.8674
------------------
```

Test 2

```
--- Newton-Raphson Method with input 2.0 for Function A ---

Iteration: 1
 x = 1.9216 with relative error: 4.08 with true error: 0.02

Iteration: 2
 x = 1.8431 with relative error: 4.26 with true error: 4.10

Iteration: 3
 x = 1.8433 with relative error: 0.01 with true error: 4.09

As of 3 iterations, x = 1.8433
------------------
```

Test 3

Part 1

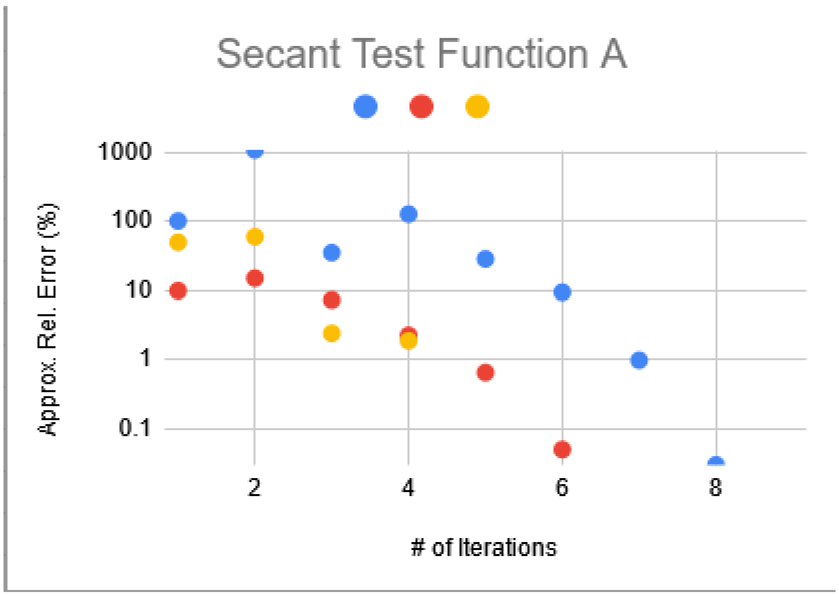
```
--- Newton-Raphson Method with input 3.0 for Function A ---

Iteration: 1
 x = 5.1333 with relative error: 41.56 with true error: 44.07

Iteration: 2
 x = 7.2667 with relative error: 29.36 with true error: 103.95

Iteration: 3
 x = 6.4031 with relative error: 13.49 with true error: 79.71

Iteration: 4
 x = 4.7420 with relative error: 35.03 with true error: 33.09

Iteration: 5
 x = 3.3922 with relative error: 39.79 with true error: 4.79

Iteration: 6
 x = 2.6936 with relative error: 25.94 with true error: 24.40

Iteration: 7
 x = 2.9002 with relative error: 7.12 with true error: 18.60

Iteration: 8
 x = 1.1612 with relative error: 149.77 with true error: 67.41

Iteration: 9
 x = 12.0344 with relative error: 90.35 with true error: 237.76

Iteration: 10
 x = 14.1399 with relative error: 14.89 with true error: 296.85
```

Part 2

```
Iteration: 91
 x = 0.0332 with relative error: 1486.16 with true error: 99.07

Iteration: 92
 x = -0.1603 with relative error: 120.69 with true error: 104.50

Iteration: 93
 x = 0.1011 with relative error: 258.48 with true error: 97.16

Iteration: 94
 x = 0.4782 with relative error: 78.85 with true error: 86.58

Iteration: 95
 x = 0.6943 with relative error: 31.13 with true error: 80.51

Iteration: 96
 x = 0.5666 with relative error: 22.55 with true error: 84.10

Iteration: 97
 x = 0.0330 with relative error: 1616.67 with true error: 99.07

Iteration: 98
 x = -0.2239 with relative error: 114.74 with true error: 106.28

Iteration: 99
 x = 0.0376 with relative error: 695.66 with true error: 98.94

Iteration: 100
 x = 0.4495 with relative error: 91.64 with true error: 87.38

As of 100 iterations, x = 0.4495
------------------
```

Secant Tests

Approx. Rel. Error (%) vs # of Iterations Graph



Test 1

```
--- Secant Method with input 0.0 and 1.0  for Function A ---

Iteration: 1
 x = 0.6250 with relative error: 100.00 with true error: 71.23

Iteration: 2
 x = -0.1034 with relative error: 1066.67 with true error: 128.34

Iteration: 3
 x = 0.4636 with relative error: 34.81 with true error: 27.02

Iteration: 4
 x = 0.3993 with relative error: 125.91 with true error: 9.39

Iteration: 5
 x = 0.3615 with relative error: 28.23 with true error: 0.95

Iteration: 6
 x = 0.3652 with relative error: 9.33 with true error: 0.06

Iteration: 7
 x = 0.3651 with relative error: 0.98 with true error: 0.03

Iteration: 8
 x = 0.3651 with relative error: 0.03 with true error: 0.03

Iteration: 9
 x = 0.3651 with relative error: 0.00 with true error: 0.03

As of 9 iterations, x = 0.3651
------------------
```

Test 2

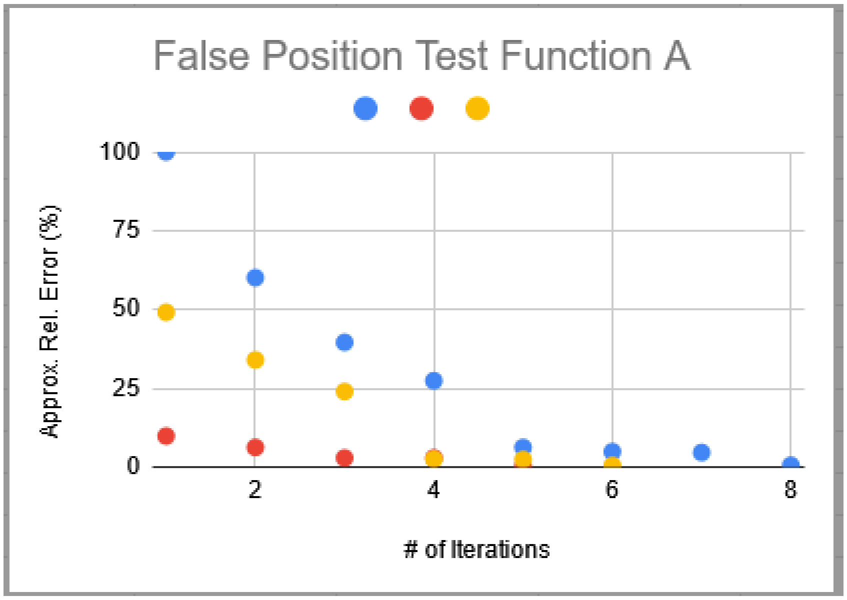
```
--- Secant Method with input 3.0 and 4.0  for Function A ---

Iteration: 1
 x = 3.3265 with relative error: 9.82 with true error: 6.64

Iteration: 2
 x = 3.4813 with relative error: 14.90 with true error: 2.29

Iteration: 3
 x = 3.5863 with relative error: 7.24 with true error: 0.65

Iteration: 4
 x = 3.5613 with relative error: 2.25 with true error: 0.05

Iteration: 5
 x = 3.5631 with relative error: 0.65 with true error: 0.00

Iteration: 6
 x = 3.5632 with relative error: 0.05 with true error: 0.00

Iteration: 7
 x = 3.5632 with relative error: 0.00 with true error: 0.00

As of 7 iterations, x = 3.5632
------------------
```

Test 3

```
--- Secant Method with input 1.0 and 3.0  for Function A ---

Iteration: 1
 x = 1.9677 with relative error: 49.18 with true error: 2.38

Iteration: 2
 x = 1.8858 with relative error: 59.08 with true error: 1.88

Iteration: 3
 x = 1.9217 with relative error: 2.40 with true error: 0.02

Iteration: 4
 x = 1.9217 with relative error: 1.87 with true error: 0.01

Iteration: 5
 x = 1.9217 with relative error: 0.00 with true error: 0.01

As of 5 iterations, x = 1.9217
------------------
```

False Position Tests

Approx. Rel. Error (%) vs # of Iterations Graph



Test 1

```
--- False Position Method with input 0.0 and 1.0  for Function A ---

Iteration: 1
 x = 0.6250 with relative error: 100.00 with true error: 71.23

Iteration: 2
 x = 0.3906 with relative error: 60.00 with true error: 7.02

Iteration: 3
 x = 0.2798 with relative error: 39.61 with true error: 23.34

Iteration: 4
 x = 0.3854 with relative error: 27.40 with true error: 5.59

Iteration: 5
 x = 0.3630 with relative error: 6.18 with true error: 0.56

Iteration: 6
 x = 0.3814 with relative error: 4.84 with true error: 4.50

Iteration: 7
 x = 0.3648 with relative error: 4.57 with true error: 0.07

Iteration: 8
 x = 0.3667 with relative error: 0.53 with true error: 0.47

As of 8 iterations, x = 0.3667
------------------
```

Test 2

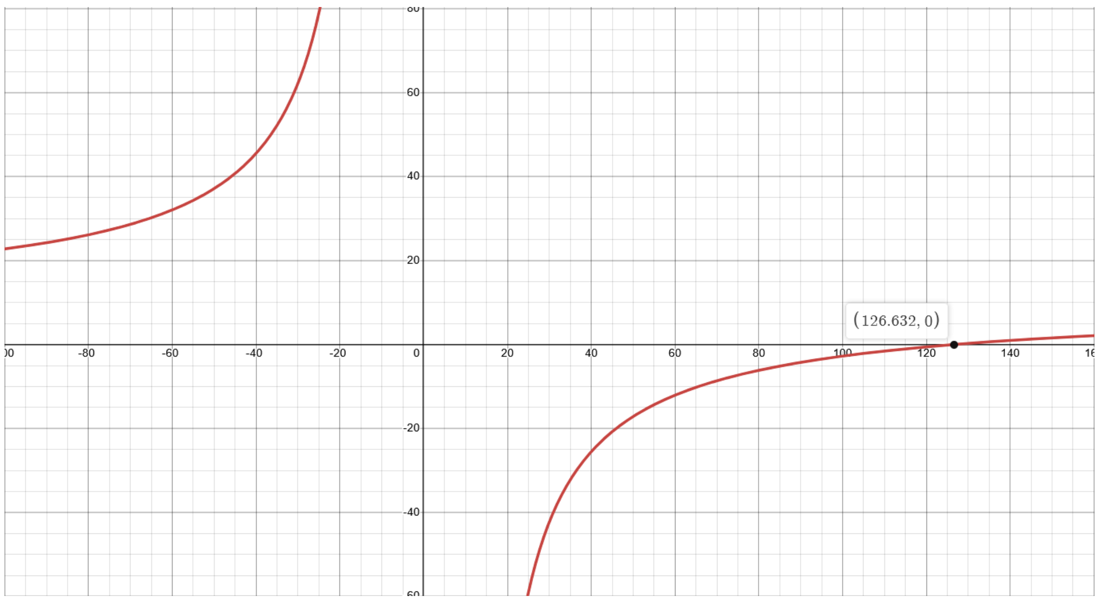
```
--- False Position Method with input 3.0 and 4.0  for Function A ---

Iteration: 1
 x = 3.3265 with relative error: 9.82 with true error: 6.64

Iteration: 2
 x = 3.5464 with relative error: 6.20 with true error: 0.46

Iteration: 3
 x = 3.6507 with relative error: 2.85 with true error: 2.46

Iteration: 4
 x = 3.5491 with relative error: 2.86 with true error: 0.39

Iteration: 5
 x = 3.5641 with relative error: 0.42 with true error: 0.03

As of 5 iterations, x = 3.5641
------------------
```

Test 3

```
--- False Position Method with input 1.0 and 3.0  for Function A ---

Iteration: 1
 x = 1.9677 with relative error: 49.18 with true error: 2.38

Iteration: 2
 x = 1.4683 with relative error: 34.02 with true error: 23.61

Iteration: 3
 x = 1.9314 with relative error: 23.98 with true error: 0.49

Iteration: 4
 x = 1.8847 with relative error: 2.48 with true error: 1.94

Iteration: 5
 x = 1.9303 with relative error: 2.37 with true error: 0.43

Iteration: 6
 x = 1.9209 with relative error: 0.49 with true error: 0.06

As of 6 iterations, x = 1.9209
------------------
```

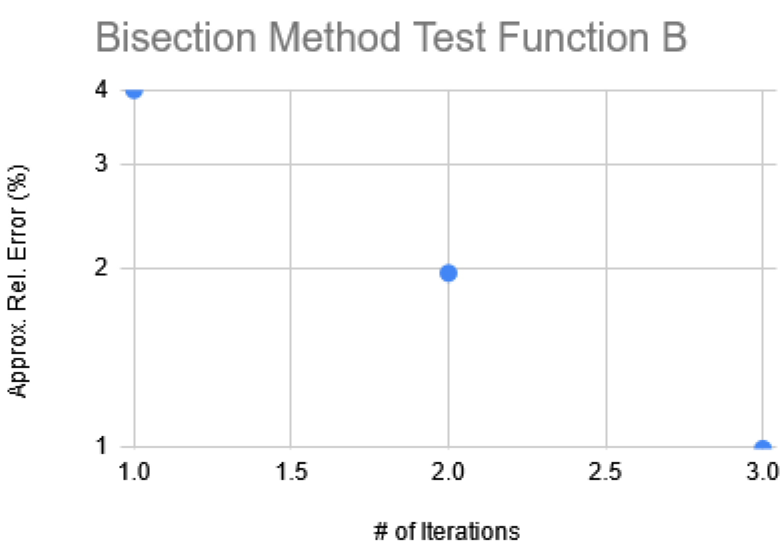
(b) f(x) = x + 10 – xcosh(50/x)

Graph



Bisection Method Test

Approx. Rel. Error (%) vs # of Iterations Graph

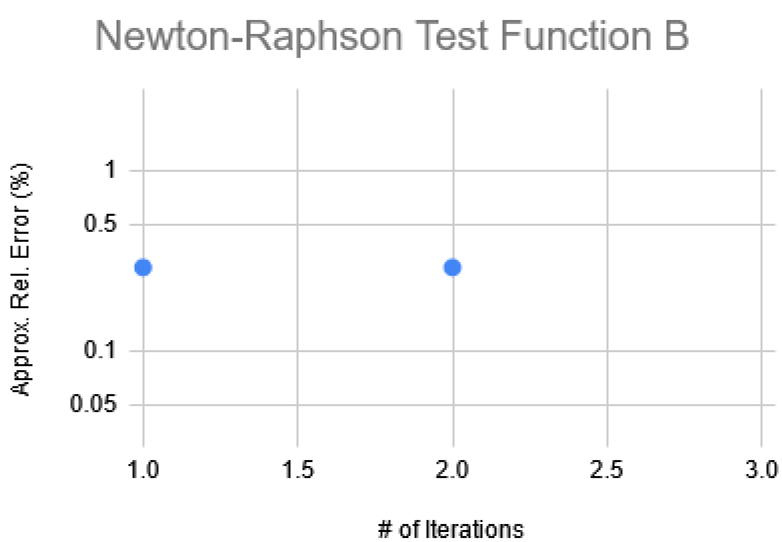


Test 1

```
--- Bisection Method with inputs 120.0 and 130.0 for Function B ---

Iteration: 1
 x = 125.0000 with relative error: 4.00 with true error: 1.29

Iteration: 2
 x = 127.5000 with relative error: 1.96 with true error: 0.69

Iteration: 3
 x = 126.2500 with relative error: 0.99 with true error: 0.30

As of 3 iterations, x = 126.2500
------------------
```

Newton-Raphson Method Test

Approx. Rel. Error (%) vs # of Iterations Graph

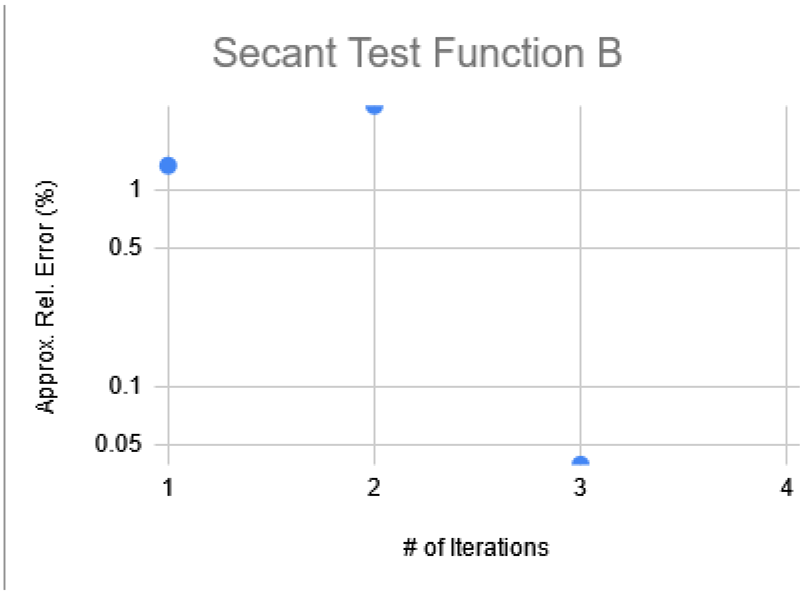


Test 1

```
--- Newton-Raphson Method with input 127.0 for Function B ---

Iteration: 1
 x = 126.6313 with relative error: 0.29 with true error: 0.00

Iteration: 2
 x = 126.2627 with relative error: 0.29 with true error: 0.29

Iteration: 3
 x = 126.2638 with relative error: 0.00 with true error: 0.29

As of 3 iterations, x = 126.2638
------------------
```

Secant Method Test

Approx. Rel. Error (%) vs # of Iterations Graph

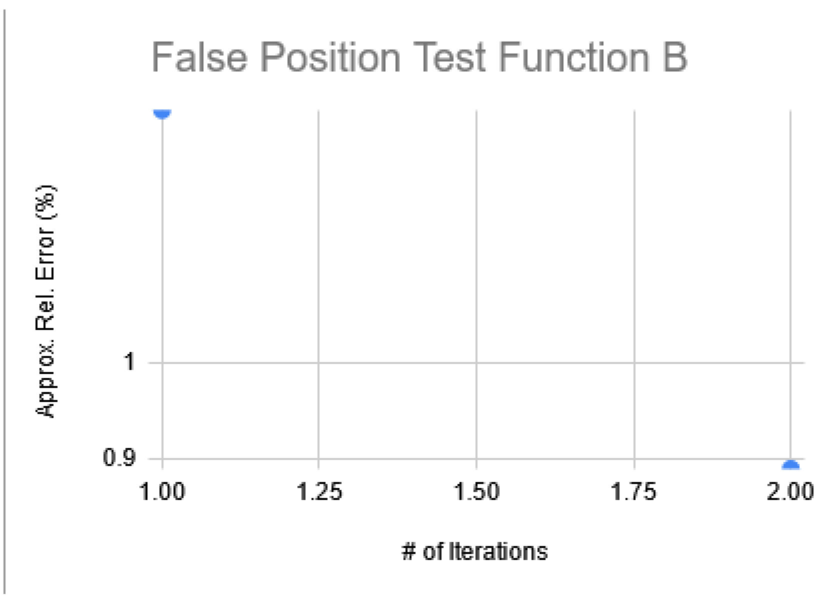


Test 1

```
--- Secant Method with input 125.0 and 130.0  for Function B ---

Iteration: 1
 x = 126.6775 with relative error: 1.32 with true error: 0.04

Iteration: 2
 x = 126.6312 with relative error: 2.66 with true error: 0.00

Iteration: 3
 x = 126.6324 with relative error: 0.04 with true error: 0.00

Iteration: 4
 x = 126.6324 with relative error: 0.00 with true error: 0.00

As of 4 iterations, x = 126.6324
------------------
```

False Position Method Test

Approx. Rel. Error (%) vs # of Iterations Graph



Test 1

```
--- False Position Method with input 125.0 and 130.0  for Function B ---

Iteration: 1
 x = 126.6775 with relative error: 1.32 with true error: 0.04

Iteration: 2
 x = 125.5628 with relative error: 0.89 with true error: 0.84

As of 2 iterations, x = 125.5628
------------------
```