This project aims to find the roots to the following two functions:

1. f(x) = 2x3 – 11.7x2 + 17.7x – 5
2. f(x) = x + 10 – xcosh(50/x)

This is done using 5 methods:

1. Bisection
2. False Position
3. Secant
4. Modified Secant
5. Newton

The first function has 3 roots located at:

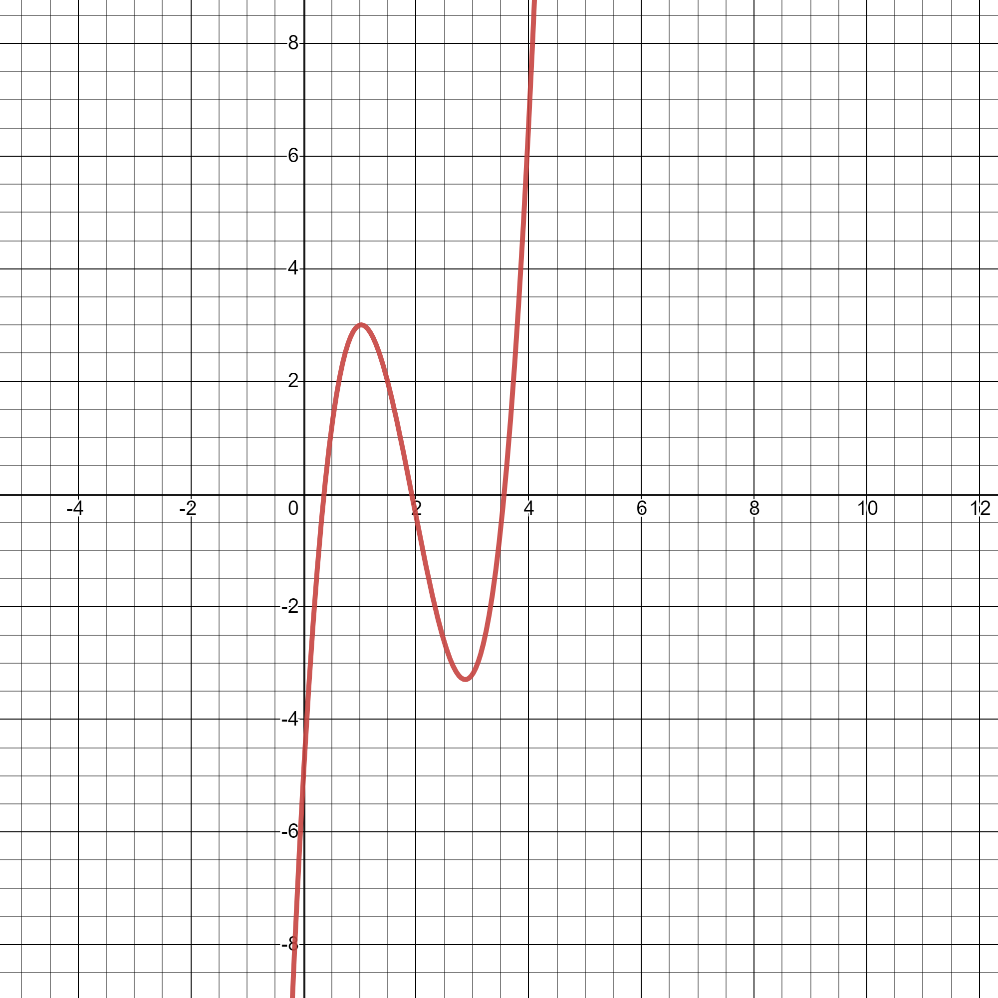
1. 0.365
2. 1.922
3. 3.563

The second function has one positive root located at

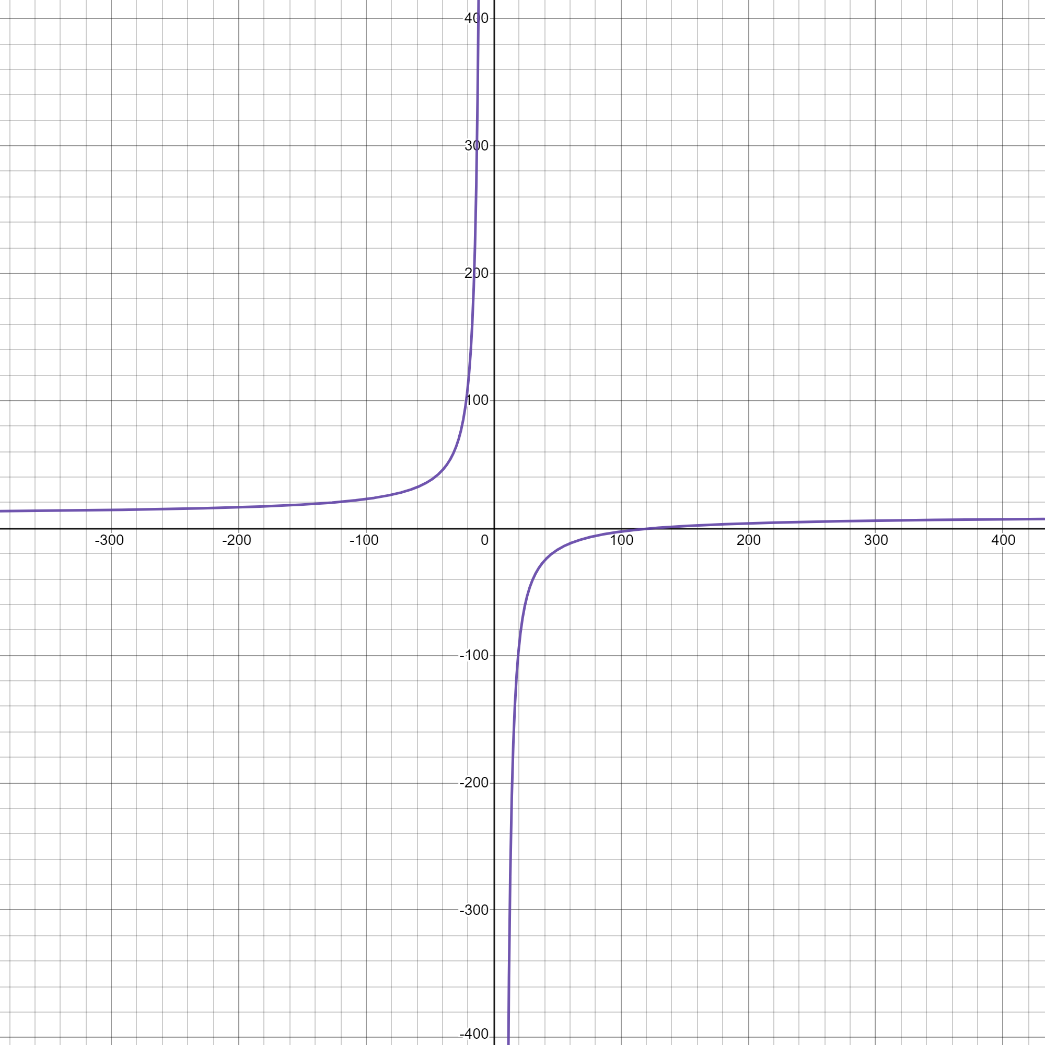
1. 126.632

These methods all approximated the value of these roots up to 0.001 accuracy. All of these methods start off with a guess to the value of the root and proceed with their unique calculations to improve the guess and therefor getting closer to the value.

**First Equation Graph:**



**Second Equation Graph:**



**Bisection**

The bisection method is a bracketing method. This means that it starts off with two values surrounding the root value. For example, for the first equation and first root I have used values A = 0 and B = 1 the functional values of these two points surround the true root = 0.365.

This method has two restrictions:

1. The functional values of the chosen points need to surround the root.
2. These two points A and B also need to be opposite signs.

The next point (Xi+1) is then calculated by taking the average of these two points = C. Then either A or B is replaced by C based on the signs. Whichever sign matches C that value is selected to be replaced. as this method runs the bracket closes more and more zoning in on the root.

The rest of the roots and the other equation are calculated the same way.

**False Position**

This is another bracketing method that is much like the Bisection. This method also has the two following restrictions:

1. The functional values of the chosen points need to surround the root.
2. These two points A and B also need to be opposite signs.

Again, taking into consideration the first equation and the first root as an example. I have chosen A=0 and B=1. This way I know that the functional values of these two points satisfy the restrictions.

In the case of the False Position the next value is calculated using the following formula:

C = a – f(a) [ b-a / f(a)-f(b) ]

Notice the part in the bracket is inverse of slope. Using the above formula, the value of C is calculated and again which ever A or B has the same sign as C is replaced with the value of C.

**Secant**

The secant method is identical to the false position method however the difference is that the values selected do not need to be on either side of the root.

The next value is again calculated using:

C = a – f(a) [ b-a / f(a)-f(b) ]

The closer A or B to the true root is then replaced with the value of C and therefor the prediction gets closer to the root. One downside to this method is that in some cases these values would approach the root at a very slow rate.

**Modified Secant**

This method is a modification to secant method that eliminated two initial guesses to the root and only requires one guess. The point is that given a small value delta in this case 0.01 we deviate the value of A and produce the B value. There are some hazards while doing this.

The formula for calculating the next value of X is as follows:

Xi+1 = Xi – f(Xi) (delta\*Xi / f(Xi + delta\*Xi) f(xi) )

**Newton**

This is an open method, also only requires one initial guess to the root. Then it uses that value to draw a tangent line on the curve at that value and estimate the next best guess to the value.

This method has proven to be quite fast yet very risky because if a wrong value is chosen for the initial guess the result could be divergent.

The formula for this method is:

X i+1 = Xi – ( f(Xi) / f ’(Xi) )

The downside to this method is that the derivative of the function needs to be available. However, in some cases the differential is very hard to attain.