Project Report

Baran Deniz Karahan 11/4/2019

Project Detail

The aim for this project is comparing the efficiency of different root finding methods for the function f(x) = $x^3 + 2x^2 + 10x - 20$ using MATLAB. The used methods are bisection method, false position method, modified false position method, secant method and Newton's method. To look at the results of different cases, the interval for finding the root is taken as (0,2) and (0,10) and the absolute error criteria is taken as 10^{-6}, 10^{-8} and 10^{-10} .

Note: Because in Newton's method only one point is used, instead of the lower bound, upper bound is selected. The reason for that is the fact that the upper bound changes between the cases while the lower bound stays the same.

Information About the Methods

All of the methods used in this project are root finding methods. The first method, bisection method, uses the intermediate value theorem. This theorem states that if a function is continuous, between two points a_1 and a_2, there exists a point a_n where $f(a_n)=c$, for every c between $f(a_1)$ and $f(a_2)$. From here it follows that if the two points are selected as having different signs, between these two points there exists a root of the function. Bisection method divides the interval by two, from the middle, and selects the interval that has two bounds which consists of a positive and a negative point. This process repeats until the error of the middle value is smaller than the absolute error margin.

Another method is the false position method, is a similar method to the bisection method. In this method instead of selecting the point that divides the interval from the middle of the interval, weighted average of the bound points is used. This method is again repeated until the point selected by using weighted average of two interval points are have smaller error than the absolute error margin. However, the slope of the functions differs greatly at each side of the root, this method becomes ineffective. To combat this, a further improvement to this method is possible. This method, called modified false position method keeps track of the changes to the bounds and if a bounds remain unchanged, it halves its weight in weighted average.

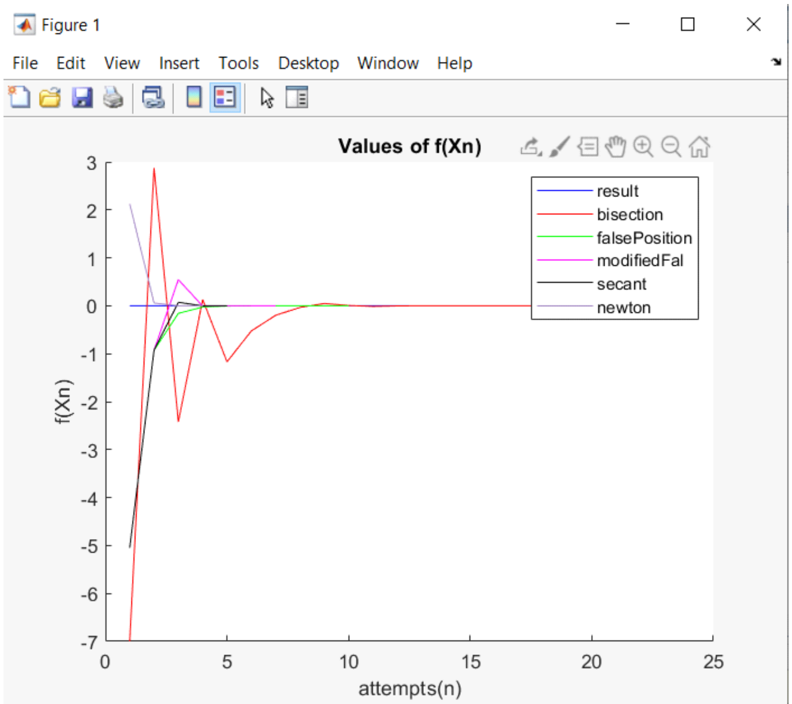
The other methods use the slope to find a root. In secant method a slope is estimated from two points, the intersection of this slope and x axis is selected as the new x value. The value of f(x) at the new x value replaces an already existing point and the process is repeated until the new value of f(x) is smaller than absolute error margin. In Newton's Method instead of two points, only one point is used. The slope is found by using the derivative of f(x).

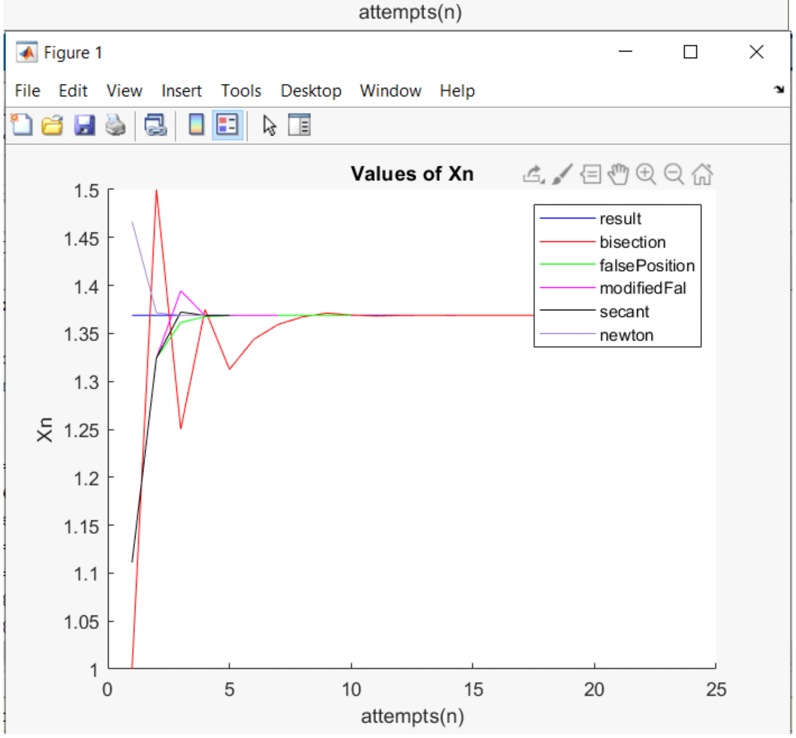
Results of the Cases: (0,2) Interval

First, the case where the interval is (0,2) and the absolute error criteria is 10^{-6} is examined. The results are as follows:

namesOfMethods	stepsTook	xValues	fValues
{'bisectionMethod' }	{[24]}	{[1.3688]}	{[8.9595e-07]}
{'falsePositionMethod' }	{[10]}	{[1.3688]}	{[-5.9302e-07]}
{'modifiedFalsePositionMethod'}	{[7]}	{[1.3688]}	{[-3.6184e-11]}
{'secantMethod' }	{[5]}	{[1.3688]}	{[-9.3768e-07]}
{'NewtonsMethod' }	{[4]}	{[1.3688]}	{[2.7313e-11]}

the name of method, steps took to reach the result, the x value of the result and the f(x) value is noted respectively for each method. For looking at each method more closely, two graphs are drawn. The first graph shows each method's $f(x_n)$ values for each step and the second graph shows each method's x_n value for each step. In both graphs the blue line(labeled as "result") shows the result that the methods converge to.

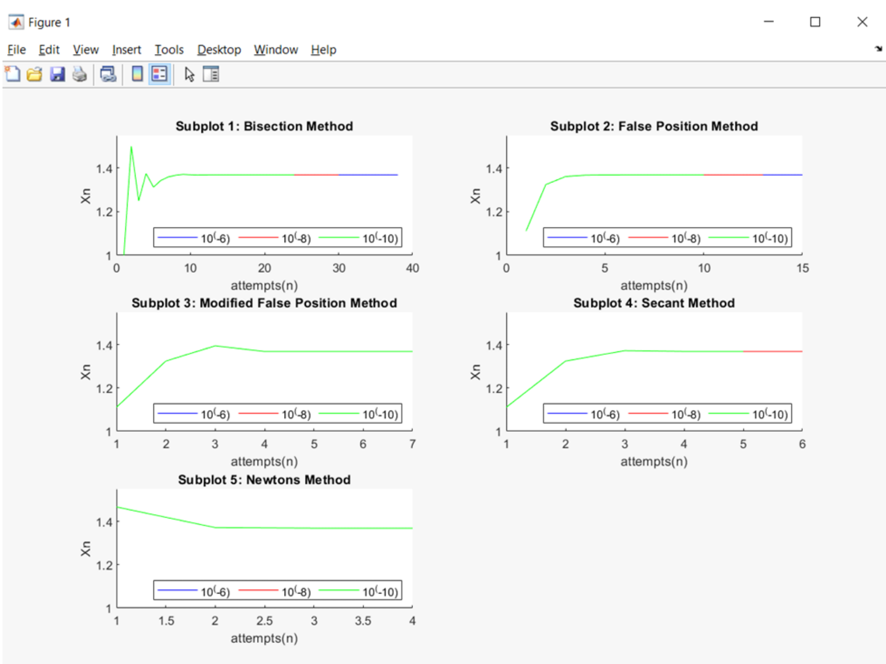


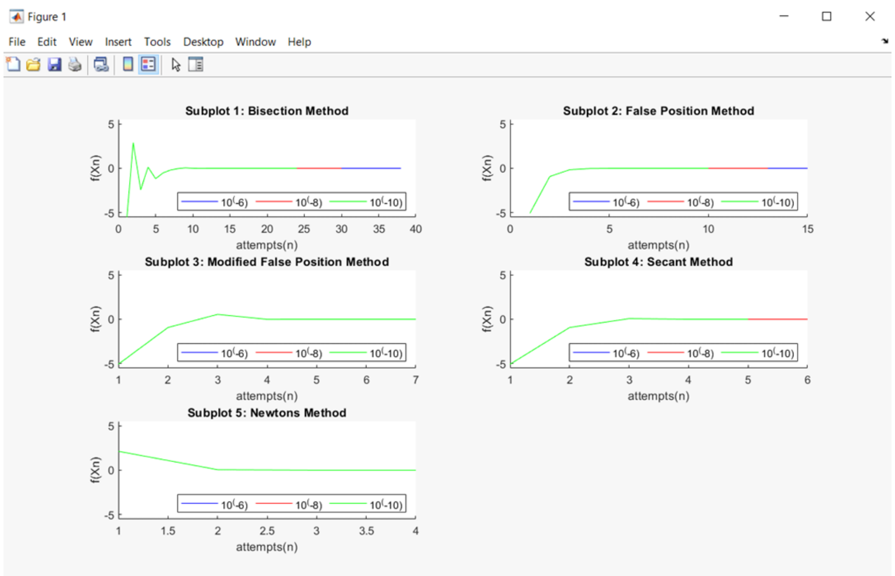


Next, the cases where absolute error criteria being changed are inspected. The results are as follows:

namesOfMethods	stepsTook	xValues	fValues
{'bisectionMethodFor10^(-6)' }	{[24]}	{[1.3688]}	{[8.9595e-07]}
{'bisectionMethodFor10^(-8)' }	{[30]}	{[1.3688]}	{[-7.8218e-09]}
{'bisectionMethodFor10^(-10)' }	{[38]}	{[1.3688]}	{[6.4482e-12]}
{'falsePositionMethodFor10^(-6)' }	{[10]}	{[1.3688]}	{[-5.9302e-07]}
{'falsePositionMethodFor10^(-8)' }	{[13]}	{[1.3688]}	{[-2.8002e-09]}
{'falsePositionMethodFor10^(-10)' }	{[15]}	{[1.3688]}	{[-7.8813e-11]}
{'modifiedFalsePositionMethodFor10^(-6)' }	{[7]}	{[1.3688]}	{[-3.6184e-11]}
{'modifiedFalsePositionMethodFor10^(-8)' }	{[7]}	{[1.3688]}	{[-3.6184e-11]}
{'modifiedFalsePositionMethodFor10^(-10)'}	{[7]}	{[1.3688]}	{[-3.6184e-11]}
{'secantMethodFor10^(-6),' }	{[5]}	{[1.3688]}	{[-9.3768e-07]}
{'secantMethodFor10^(-8),' }	{[6]}	{[1.3688]}	{[1.2108e-11]}
{'secantMethodFor10^(-10),' }	{[6]}	{[1.3688]}	{[1.2108e-11]}
{'NewtonsMethodFor10^(-6)' }	{[4]}	{[1.3688]}	{[2.7313e-11]}
{'NewtonsMethodFor10^(-8)' }	{[4]}	{[1.3688]}	{[2.7313e-11]}
{'NewtonsMethodFor10^(-10)' }	{[4]}	{[1.3688]}	{[2.7313e-11]}

the name of method with the error criteria, steps took to reach the result, the x value of the result and the f(x) value is noted respectively for each method. For inspecting the steps of each method a graph for each method is created. In each graph all 3 error criteria is plotted. From results above, it can be noted that if the error criteria is decreased the steps do not increase. This means the smallest error criteria has the most steps and the biggest has the least steps. Using this fact, the graphs are plotted, and the lack of a color represents it being equal to the results of bigger error criteria.





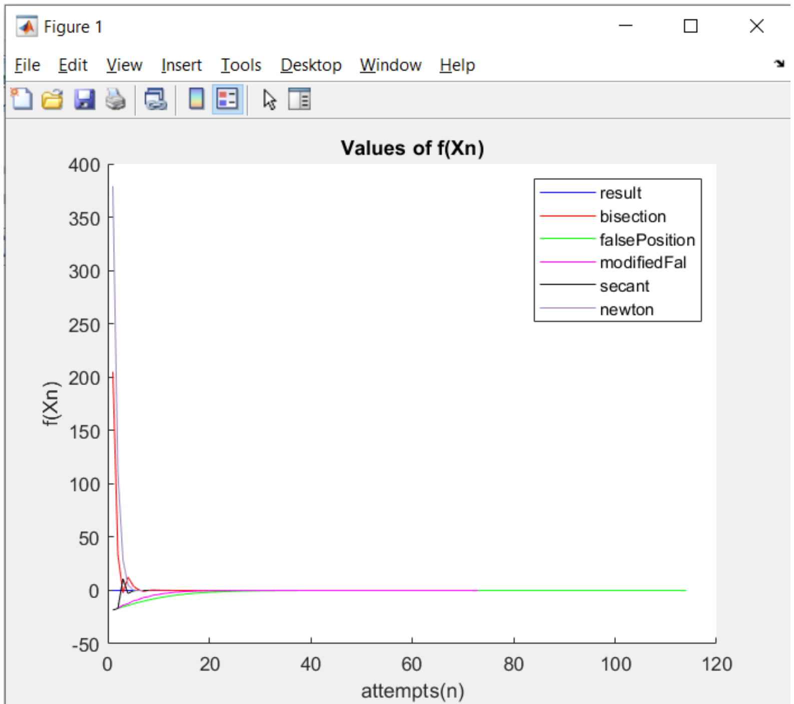
From the graphs, it can be seen that the most efficient method is the Newton's Method and the least efficient method is Bisection Method. This is expected, as Bisection Method is the simplest and uses the least amount of information from the graph while Newton's Method uses the most. The method that is most affected from the error criteria is Bisection Method and False Position Method, and the least affected is Newton's Method. This is, again, expected from the general effectiveness of the methods.

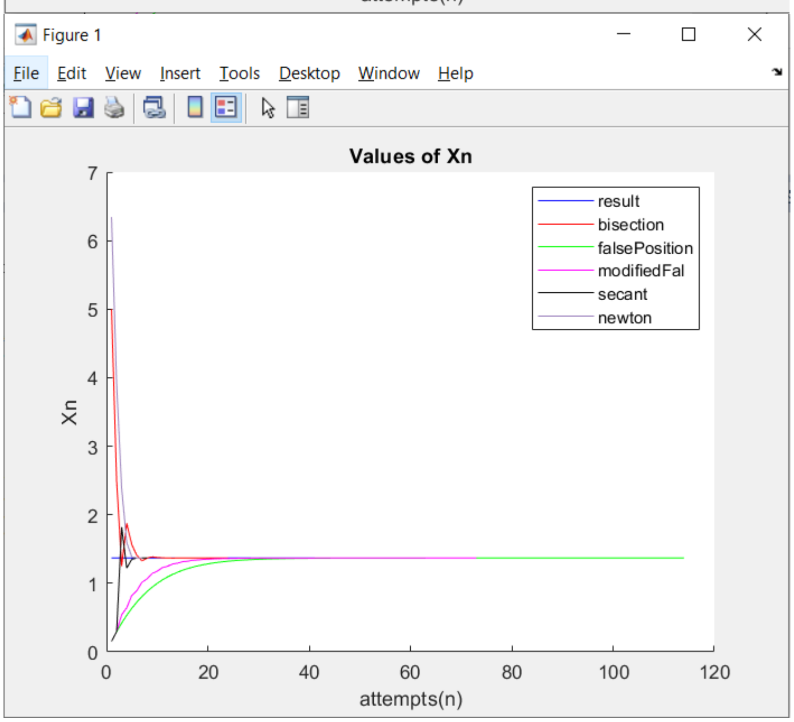
Results of the Cases: (0,10) Interval

Secondly, the case where the interval is (0,2) and the absolute error criteria is 10^{-6} is examined. The results are as follows:

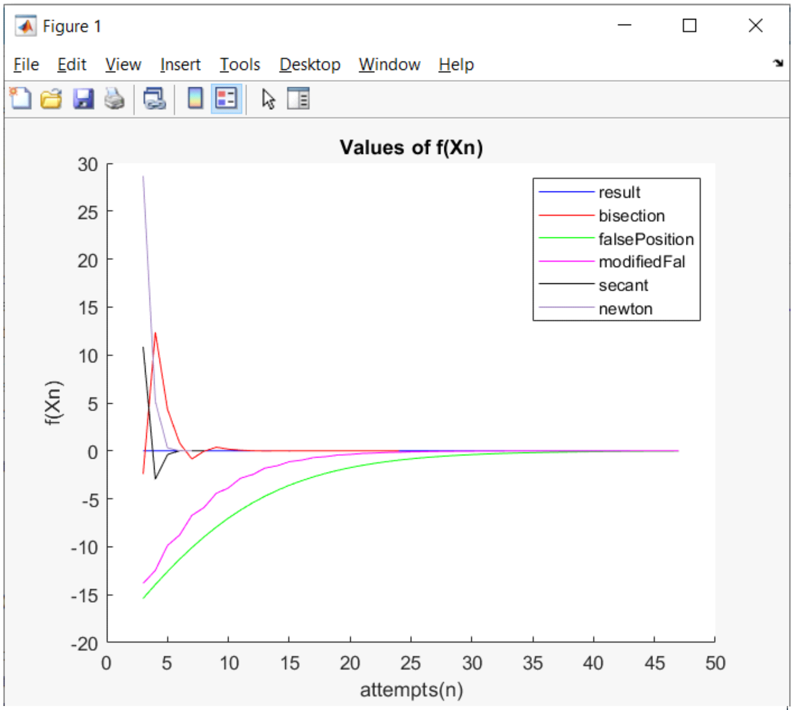
namesOfMethods	stepsTook	xValues	fValues
{'bisectionMethod' }	{[24]}	{[1.3688]}	{[8.9595e-07]}
{'falsePositionMethod' }	{[114]}	{[1.3688]}	{[-9.9406e-07]}
{'modifiedFalsePositionMethod'}	{[73]}	{[1.3688]}	{[-8.2320e-07]}
{'secantMethod' }	{[8]}	{[1.3688]}	{[-2.0431e-08]}
{'NewtonsMethod' }	{[7]}	{[1.3688]}	{[2.0794e-08]}

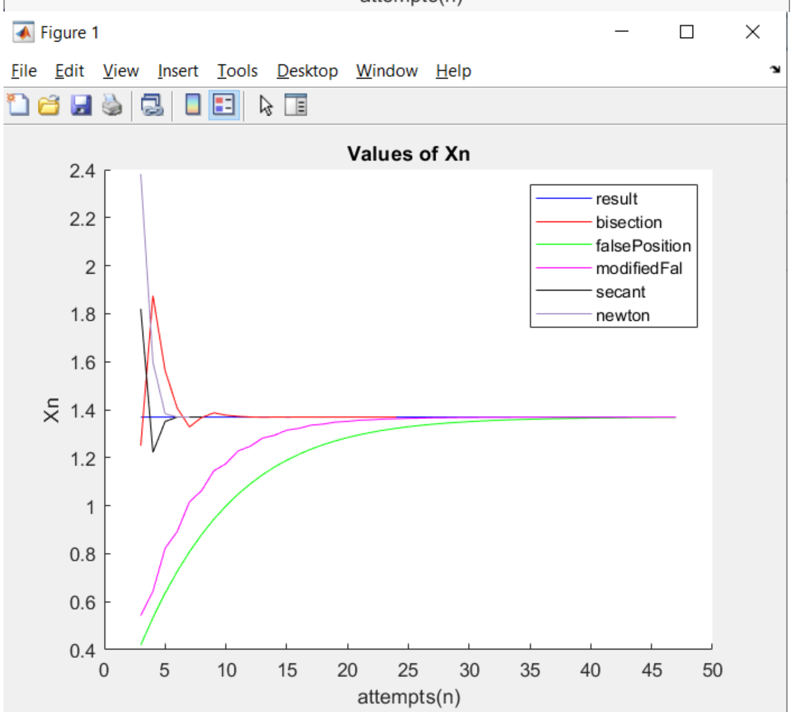
The name of method, steps took to reach the result, the x value of the result and the f(x) value is noted respectively for each method. For looking at each method more closely, two graphs are drawn. The first graph shows each method's $f(x_n)$ values for each step and the second graph shows each method's x_n value for each step. In both graphs the blue line(labeled as "result") shows the result that the methods converge to.





However, some of the methods' initial steps has too high and some method's has too much steps that has not that much of movement. To better visualize the data, the steps are bounded between (3,47)

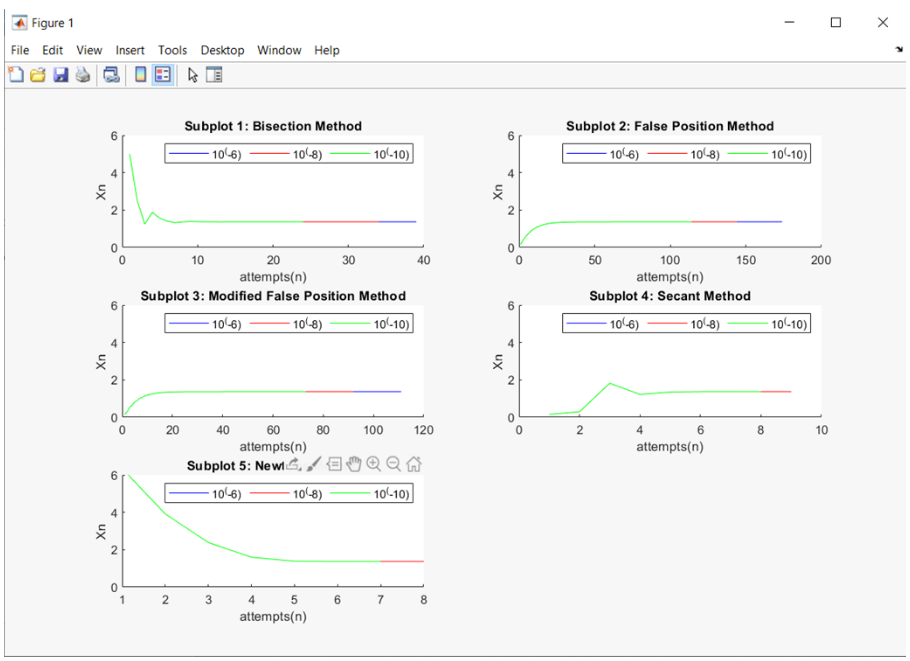


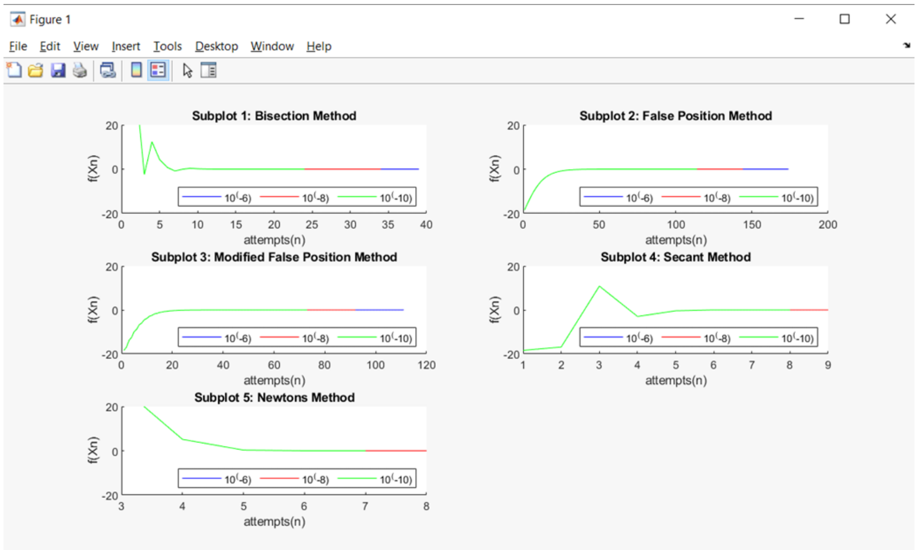


Next, the cases where absolute error criteria being changed are inspected. The results are as follows:

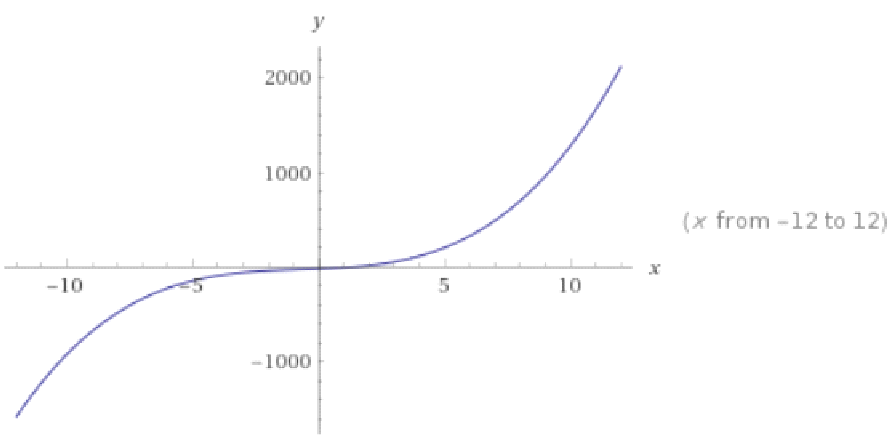
```
{'bisectionMethodFor10^(-6)'                 }    {[ 24]}    {[1.3688]}    {[ 8.9595e-07]}
{'bisectionMethodFor10^(-8)'                 }    {[ 34]}    {[1.3688]}    {[-4.5404e-10]}
{'bisectionMethodFor10^(-10)'                }    {[ 39]}    {[1.3688]}    {[-7.0301e-11]}
{'falsePositionMethodFor10^(-6)'             }    {[114]}    {[1.3688]}    {[-9.9406e-07]}
{'falsePositionMethodFor10^(-8)'             }    {[144]}    {[1.3688]}    {[-9.9581e-09]}
{'falsePositionMethodFor10^(-10)'            }    {[174]}    {[1.3688]}    {[-9.9760e-11]}
{'modifiedFalsePositionMethodFor10^(-6)'  }    {[ 73]}    {[1.3688]}    {[-8.2320e-07]}
{'modifiedFalsePositionMethodFor10^(-8)'  }    {[ 92]}    {[1.3688]}    {[-8.7205e-09]}
{'modifiedFalsePositionMethodFor10^(-10)'}    {[111]}    {[1.3688]}    {[-7.7058e-11]}
{'secantMethodFor10^(-6),'                   }    {[  8]}    {[1.3688]}    {[-2.0431e-08]}
{'secantMethodFor10^(-8),'                   }    {[  9]}    {[1.3688]}    {[ 2.4869e-14]}
{'secantMethodFor10^(-10),'                  }    {[  9]}    {[1.3688]}    {[ 2.4869e-14]}
{'NewtonsMethodFor10^(-6)'                   }    {[  7]}    {[1.3688]}    {[ 2.0794e-08]}
{'NewtonsMethodFor10^(-8)'                   }    {[  8]}    {[1.3688]}    {[          0]}
{'NewtonsMethodFor10^(-10)'                  }    {[  8]}    {[1.3688]}    {[          0]}
```

The name of method with the error criteria, steps took to reach the result, the x value of the result and the f(x) value is noted respectively for each method. For inspecting the steps of each method, the process done to (0,2) case is repeated for (0,10) case.





From the graphs, it can be seen that the most efficient method is the Newton's Method and the least efficient method is False Position Method. While Newton's Method being the most efficient is expected, False Position being least is not. The reason for this is the fact that the function's slope changes drastically between 0 and 10(visualization, done by using wolfram, is at the image below). Such occurences create inefficiency for False Position Method.

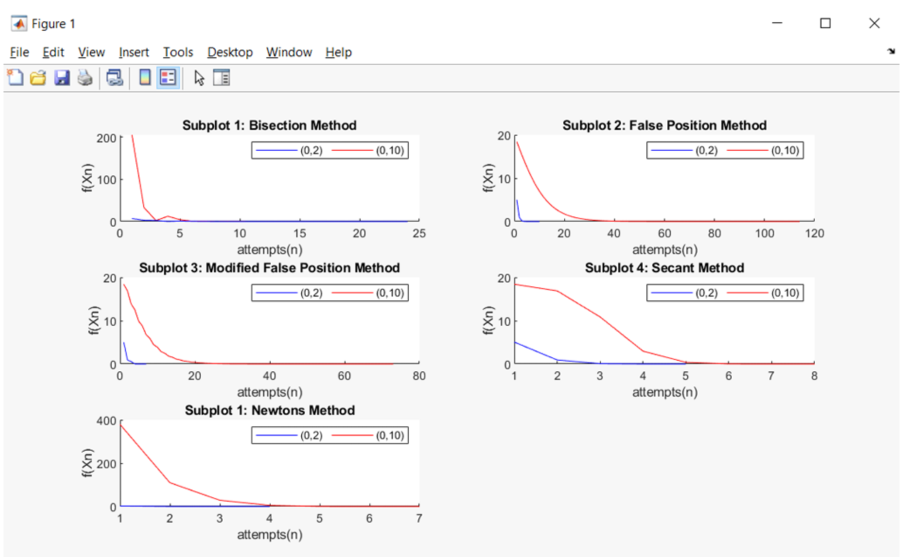


Results of the Cases: Comparing (0,2) Interval and (0,10) Interval

Lastly, how the methods efficiency differs depending on the interval is examined. The error criteria is accepted as 10^{-6}.The results are as follows:

namesOfMethods	stepsTook	xValues	fValues	AbsError
{'bisectionMethod(0,2)' }	{[24]}	{[1.3688]}	{[8.9595e-07]}	{[8.9595e-07]}
{'bisectionMethod(0,10)' }	{[24]}	{[1.3688]}	{[8.9595e-07]}	{[8.9595e-07]}
{'falsePositionMethod(0,2)' }	{[10]}	{[1.3688]}	{[-5.9302e-07]}	{[5.9302e-07]}
{'falsePositionMethod(0,10)' }	{[114]}	{[1.3688]}	{[-9.9406e-07]}	{[9.9406e-07]}
{'modifiedFalsePositionMethod(0,2)' }	{[7]}	{[1.3688]}	{[-3.6184e-11]}	{[3.6184e-11]}
{'modifiedFalsePositionMethod(0,10)'}	{[73]}	{[1.3688]}	{[-8.2320e-07]}	{[8.2320e-07]}
{'secantMethod(0,2)' }	{[5]}	{[1.3688]}	{[-9.3768e-07]}	{[9.3768e-07]}
{'secantMethod(0,10)' }	{[8]}	{[1.3688]}	{[-2.0431e-08]}	{[2.0431e-08]}
{'NewtonsMethod(0,2)' }	{[4]}	{[1.3688]}	{[2.7313e-11]}	{[2.7313e-11]}
{'NewtonsMethod(0,10)' }	{[7]}	{[1.3688]}	{[2.0794e-08]}	{[2.0794e-08]}

The name of method , steps took to reach the result, the x value of the result and the f(x) value and absolute error is noted respectively for each method. To look for the difference between steps, the below graphs are plotted. From above information it can be seen that (0,2) interval has less or equal number of steps than (0,10) interval. Using this, the graphs below are plotted with the property of blue being able to plot over red.



From those graphs, some remarks can be made.One of them is the fact that False Position Method is highly dependent on the characteristic of the function as it is the method that is affected by the interval the most. Another one is that while Modified False Position Method improves False Position Method, it does not eliminates all of its flaws. Although it is not affected as badly as False Position Method, it still has the second most increase in the number of steps. It can also be said that the Newton's Method and Secant Method are the most reliable methods, as they are the most efficient in both scenarios. Lastly, for bisection method there is little (in fact, non) changes in the required steps. The reason for this is, because bisection always halves the sections and the interval is only increased 5 times, the possible increase in required steps can be estimated as 2.2 ($\log_2 5$ = 2,3219 ≈2.2) which is not much, considering for (0,2) the number of steps was 24.