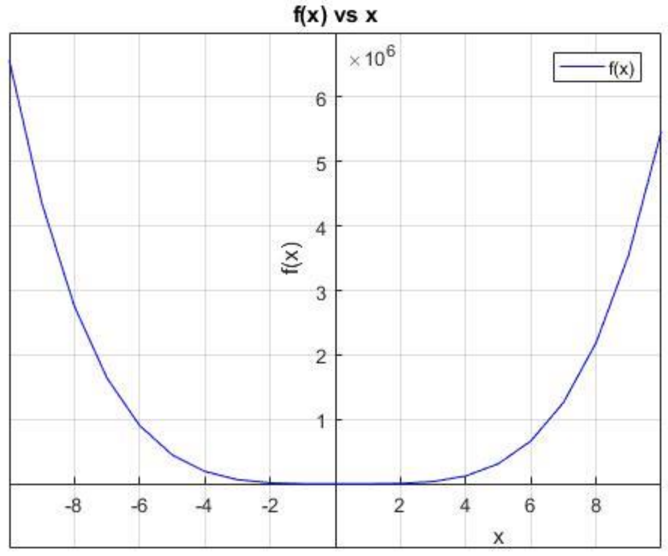
Name-Robin Singh ESO-208

Roll No.-150591 CA-01

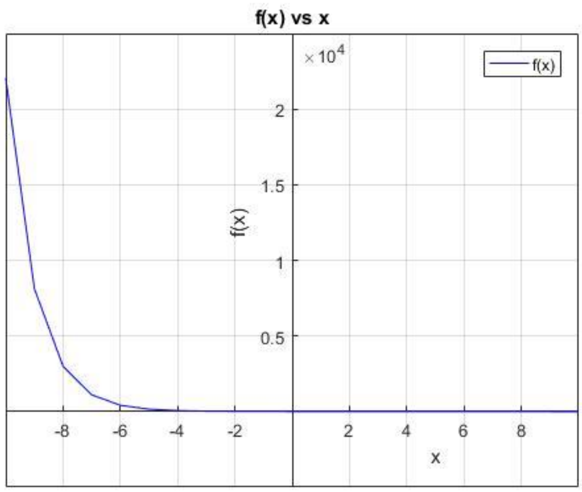
Section – G10

QUESTION-1

Test Case 1: $f(x) = 600x^4 - 550x^3 + 200x^2 - 20x$



Test Case 2: $f(x) = e^{-x} - x$

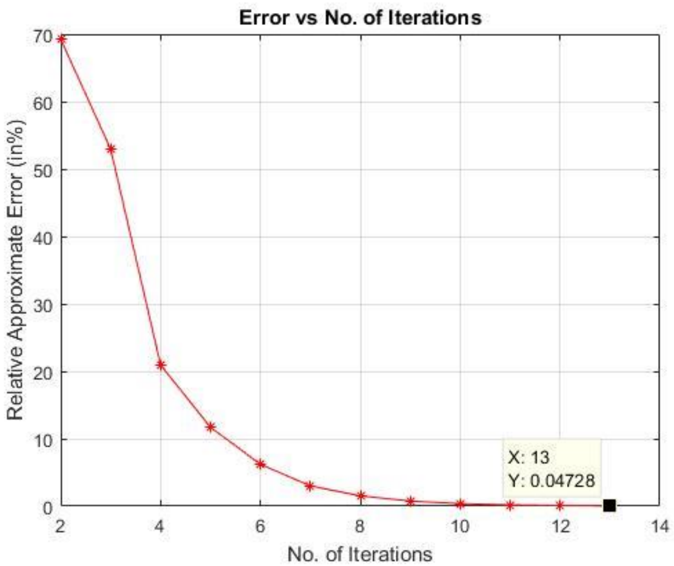


1. BISECTION METHOD

Test Case 1: $600x^4 - 550x^3 + 200x^2 - 20x - 1$

xl=0.1, $xu = 1$, Maximum Relative Approximate Error = 0.05%

Root between xl and xu is **0.232385**

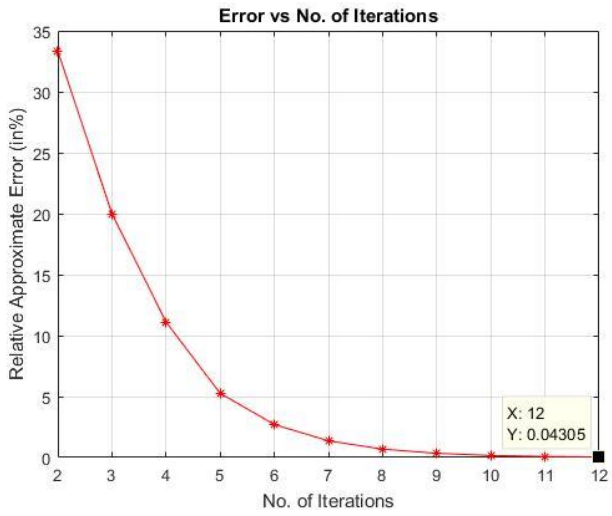


It takes **13** iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

Test Case 2: $e^{-x} - x$

xl=0.1, $xu = 1$, Maximum Relative Approximate Error = 0.05%

Root between xl and xu is **0.567139**



It takes **12** iterations for this method to converge to required root under predefined precision.

From above figure it is clear that error decreases to about half after every iteration. For ex. In 2nd test case error reduces from 20 to 10 ,10 to 5….and so on

i.e.

$$Ea_n = \frac{\Delta x}{2^n} \quad \text{where } \boldsymbol{n} \text{ is the no. of iteration.}$$

∴ Bisection Method has a Linear Convergence Rate.

Bisection Method converges with 1st order of convergence. Convergence is slow and is always stable.

It also offers guaranteed Convergence in contrast to other Methods.

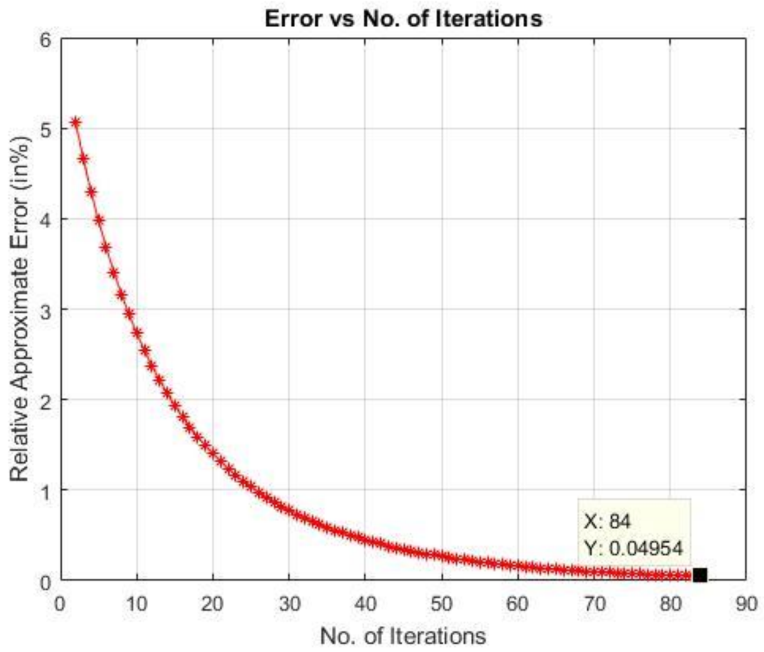
A shortcoming of the bisection method is that, in dividing the interval from xl to xu into equal halves, no account is taken of the magnitudes of $f(xl)$ and $f(xu)$. For example, if $f(xl)$ is much closer to zero than $f(xu)$, it is likely that the root is closer to xl than to xu.
For this False Position Method is devised.

2. FALSE POSITION METHOD

Test Case 1: $\boldsymbol{600x^4 - 550x^3 + 200x^2 - 20x - 1}$

xl=0.1, $xu = 1$, Maximum Relative Approximate Error = 0.05%

Root between xl and xu is **0.230035**

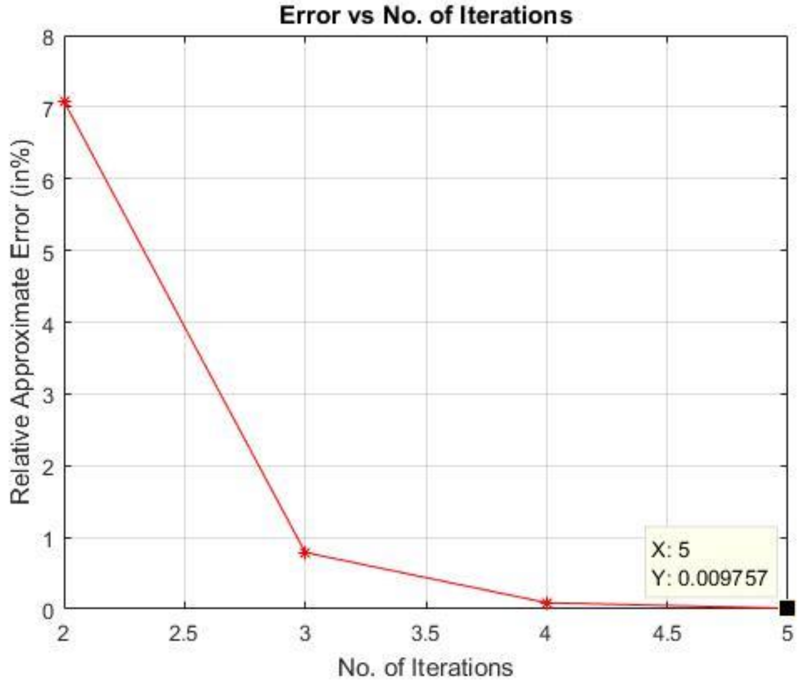


It takes **84** iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

Test Case 2: $\boldsymbol{e^{-x} - x}$

xl=0.1, $xu = 1$, Maximum Relative Approximate Error = 0.05%

Root between xl and xu is **0.567150**



It takes **5** iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

False Position Method use straight line to approximate roots hence is significantly faster than Bisection Method. As shown by test case 2 it converges only in 5 iterations as compare to bisection method's 12 iterations.

False Position Method also has first order Convergence. Convergence is slow, but works faster in some cases and is always stable.
BUT-

Test case 1 shows the one-sided nature of false position method i.e. when one of the bound x_l or x_u remains fixed during the most of the iterations and the other bound converges the root.

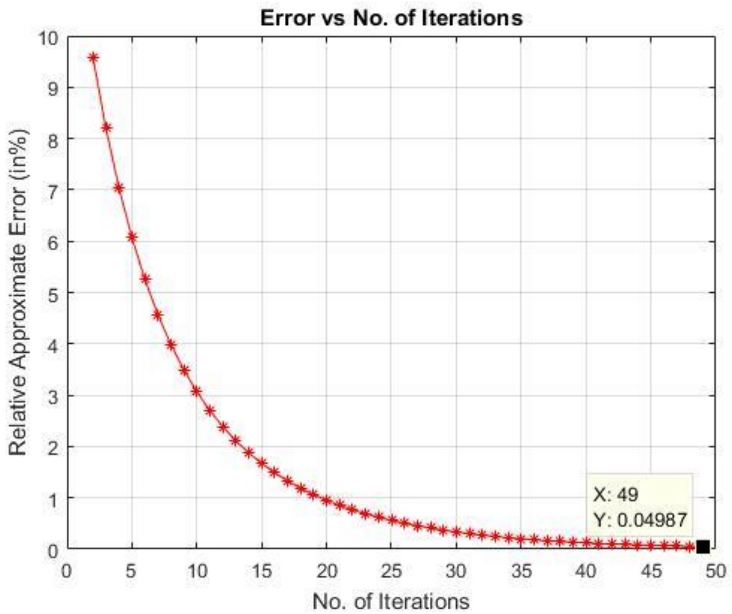
This can lead to poor convergence, particularly for functions with significant curvature.
So, we modify False Position Method's algorithm a little bit.

3. MODIFIED FALSE POSITION METHOD

Test Case 1: $\mathbf{600x^4 - 550x^3 + 200x^2 - 20x - 1}$

xl=0.1, $xu = 1$, Maximum Relative Approximate Error = 0.05%

Root between xl and xu is **0.231237**

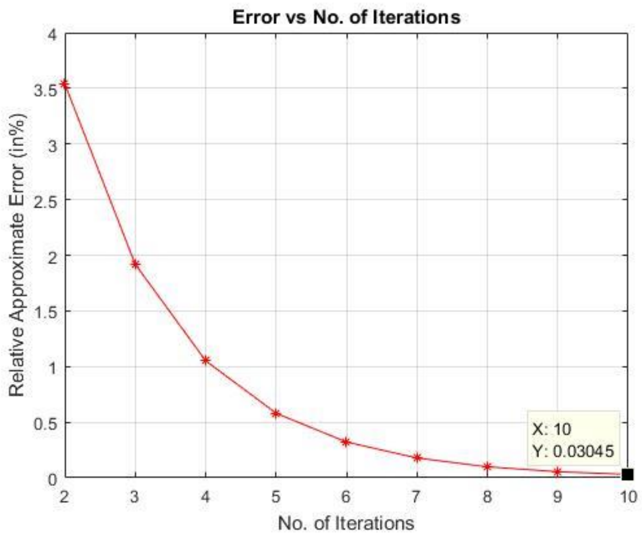


It takes **49** iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

Test Case 2: $\boldsymbol{e^{-x} - x}$

xl=0.1, $xu = 1$, Maximum Relative Approximate Error = 0.05%

Root between xl and xu is **0.567359**



It takes **10** iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

We write algorithm of Modified False Position such that it detects when one of the bounds is stuck. If this occurs, the function value at the stagnant bound can be divided in half.

Convergence is moderate and case dependent and is always stable.

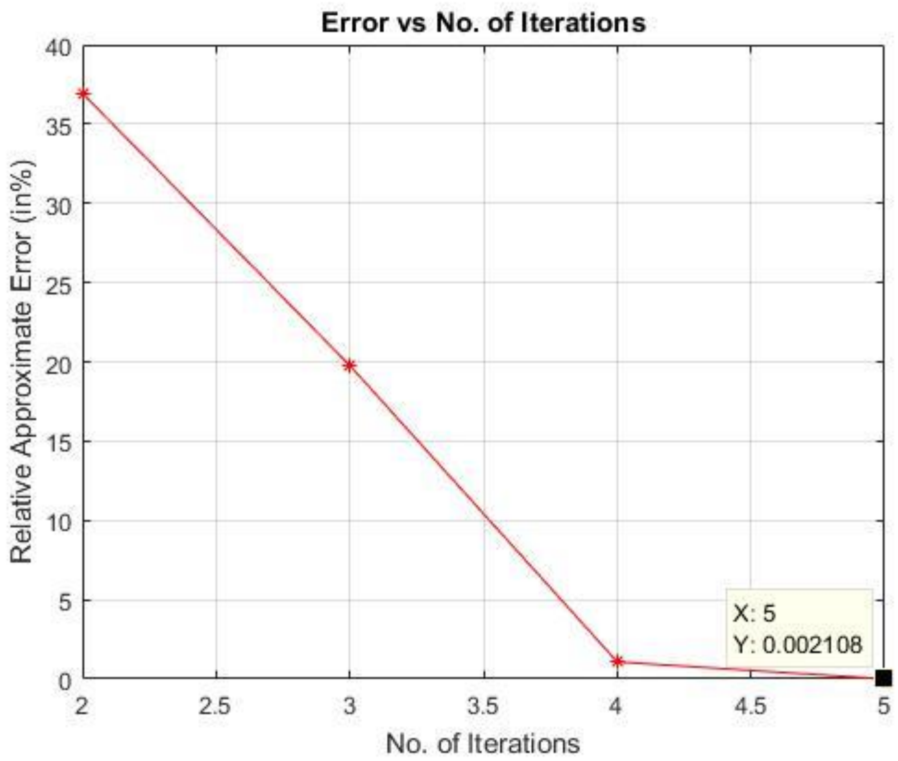
Modified False Position converges much faster than other methods as evident from above in test case 1 False Position took 84 iterations whereas Modified False takes only 49 iterations.

4. NEWTON-RAPHSON METHOD

Test Case 1: $\mathbf{600x^4 - 550x^3 + 200x^2 - 20x - 1}$

x_g=0.5, Maximum Relative Approximate Error = 0.05%

Root of the function is **0.232313**

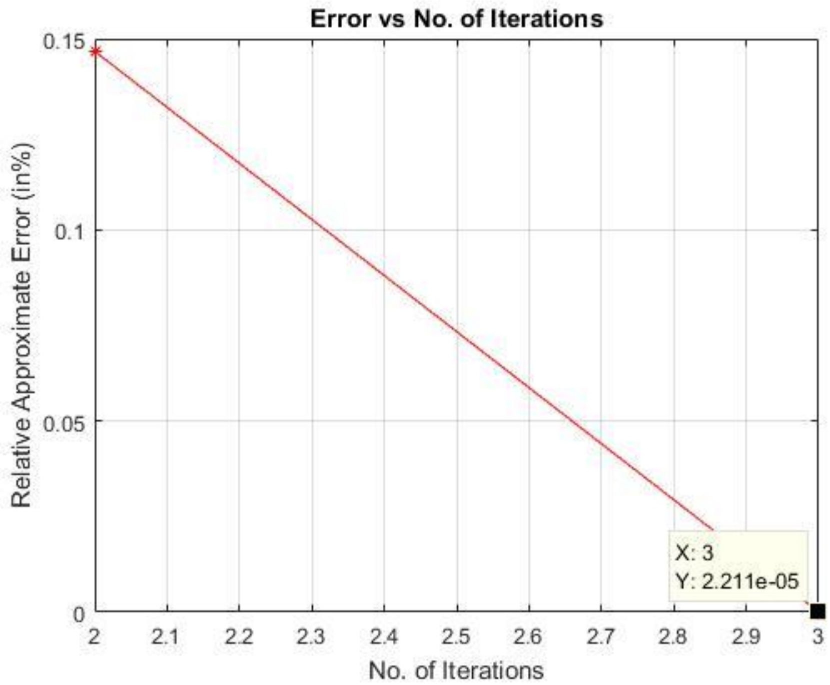


It takes **5** iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

Test Case 2: $e^{-x} - x$

x_g=0.5, Maximum Relative Approximate Error = 0.05%

Root of the function is **0.567143**



It takes 3 iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

Newton Raphson Method is one of the most widely used algorithm for computations of roots.

Here

$$\lim_{i\to\infty} \frac{e_{i+1}}{{e_i}^2} = \frac{1}{2}\left|\frac{f''(x_r)}{f'(x_r)}\right|$$

i.e. Order of Convergence is 2.

Due to second order convergence we can see from above plots that error reduces extremely faster than other methods in a single iteration.

But Newton-Raphson has some notable pitfalls -

1. Slowly converging function like $x^{10} - 1$ takes a lot of iterations before reducing to some solution.
2. It also doesn't work if there is an inflection point, maxima and minima at root.

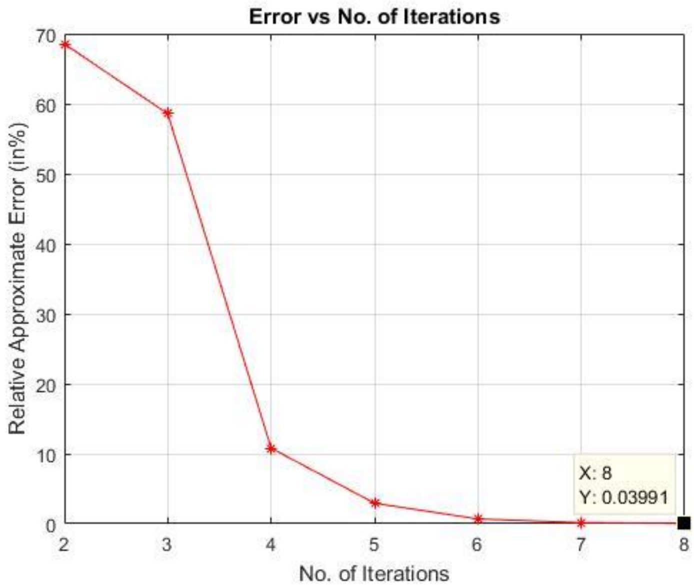
3. As derivative of the function is required in algorithm sometimes it becomes computationally heavy to calculate derivative.

5. SECANT-METHOD

Test Case 1: $\boldsymbol{600x^4 - 550x^3 + 200x^2 - 20x - 1}$

x_{-1}=0.1, $x_0 = 1$, Maximum Relative Approximate Error = 0.05%

Root of the function is **0.232371**

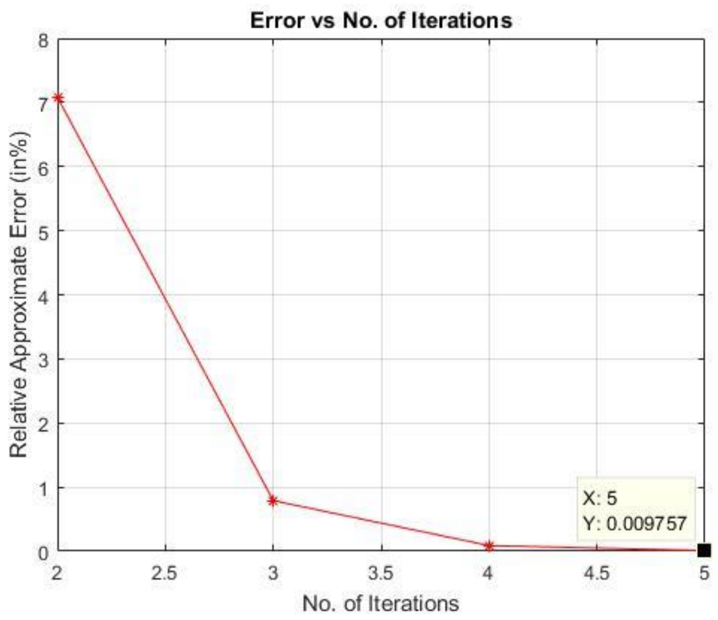


It takes 8 iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

Test Case 2: $\boldsymbol{e^{-x} - x}$

x_{-1}=0.1, $x_0 = 1$, Maximum Relative Approximate Error = 0.05%

Root of the function is **0.567150**



It takes **5** iterations for this method to converge to required root under predefined precision i.e. Relative Approximate Error of 0.05%.

In Secant Method we just approximate the derivative by simple slope equation. It helps us to get rid of that heavy derivative computation. But at the same place no. of iterations will be increased to get the desired accuracy.

This can be evident from the fact that order of convergence of Secant Method is around 1.62 called ***Golden Ratio*** which means it will take more no. of iterations than Newton-Raphson Method.