

Comparative Study of False Position Method and Newton Raphson Method

Purva Deepak Kolhe (Student).

Darshan Sanjay Gunjal (Student).

SYBSc (Computer Science),

Ashoka Center for Business and Computer Studies,

Nashik, India.

Abstract: This paper is predicated on comparison between iterative strategies for resolution non-linear equation in one variable. There are varied iterative strategies like Bisection, Secant, Muller, Regula-Falsi (R-F) or False Position, Newton-Raphson (N-R) strategies. During this paper we have a tendency to discuss 2 numerical strategies particularly, Regula-Falsi (R-F) technique and Newton-Raphson (N-R) technique. These 2 strategies are a lot of economical in computing the roots of a non-linear equation therefore, we decide these numeric strategies. Their rate of convergence, advantages, disadvantages, algorithm, examples are discuss in this paper. The software system SCILAB version 5.4.1 is employed to unravel non-linear equations in one variable by victimization the higher than iterative strategies and realize the worth of real root.

Keywords: - Convergence, Roots, Algorithm, Iterations, False Position (Regula-Falsi) method, Newton-Raphson (N-R) method.

I. INTRODUCTION

Root finding downside are a haul of finding a root of the equation $f(x) = 0$, wherever $f(x)$ could be a operate of one variable, x . Let $f(x)$ be a operate, we've got an interest to hunt out $x = x_i$ such $f(x_i) = 0$. The number x_i is termed the inspiration or zero of $f(x)$. $f(x)$ is conjointly algebraical, pure mathematics or transcendental operate. The idea finding downside is one of the foremost relevant machine issues. It arises throughout a really wide selection of sensible applications in Physics, Chemistry, Biosciences, Engineering, etc. Relevant things in Physics wherever such issues are required to be solved embrace finding the equilibrium position of an object, potential surface of a field and quantity energy of confined structure. The common root-finding strategies include: division, Newton-Raphson (N-R Method), Regula-Falsi technique (False position), Secant strategies, Muller's technique etc. completely different strategies converge to the premise at different rates. That is, some strategies are quicker in joining to the inspiration than others.

In arithmetic, the regula-falsi, technique of false position, or false position technique is additionally an extremely previous technique for resolution an equation with one unknown that, in changed type, continues to be in use. In easy terms, the strategy is that the trial and error technique of victimization check ("false") prices for the variable therefore adjusting the check value in step

with the result. This is usually often typically conjointly mentioned as "guess and check". Versions of the strategy predate the arrival of pure mathematics and then the utilization of equations.

Newton's technique is that the simplest notable procedure for locating the roots of an equation. It has been generalized in some ways that for the solution of alternative, harder nonlinear issues, as an example, system of nonlinear equations and nonlinear integral and differentiation equations. It is not perpetually the foremost effective technique for a given downside however its formal simplicity and its nice speed usually lead it to be the first technique that people use in making an attempt to unravel a nonlinear issues.

Further this paper is organized as a section a) through d) explains the algorithm, rate of convergence, advantages and disadvantages respectively. After that we solve 3 numerical by using scilab software in the section numerical experiments. And finally in conclusion section both methods are compared by using comparative table from section Discussion. [4]

II. METHOD OF FALSE POSITION

This is the oldest method for finding the real root open nonlinear equation $f(x) = 0$

And closely resemble the bisection method. In this method, also known as *regula – falsior method of cord*, which choose two point a & b such that $f(a)$ and $f(b)$ are of opposite signs. Hence, root must lie in between this point. Now, the equation of chord joining the two points $[a, f(a)]$ and $[b, f(b)]$ is given by

$$\frac{y - f(a)}{x - a} = \frac{f(b) - f(a)}{b - a}.$$

This method consists is replacing the part of curve between the points $[a, f(b)]$ and $[b, f(b)]$ by means of the chord joining the points, and taking the points intersection of chord with the x – axis as an approximation to the basis. The point of intersection in the present case is obtained by putting $y = 0$ thus we obtain

$$c = a - \frac{f(a)}{f(b) - f(a)}(b - a)$$

$$c = \frac{af(b) - bf(a)}{f(b) - f(a)}.$$

Which is first *approximation* to the root of $f(x) = 0$. If now $f(c)$ and $f(a)$ opposite signs, then the root lies between a and c , we replace b by c in above equation and obtain the *next* approximation. Otherwise, replace a by c and general the next approximation. The procedure is repeated till the root is obtained to desired accuracy. Figure gives a graphical representation of method. [1]

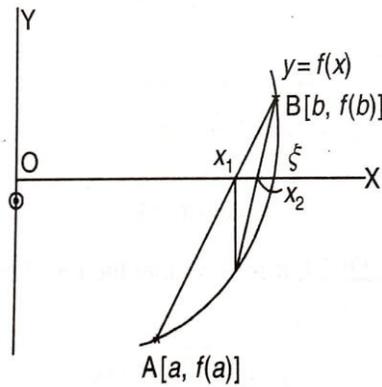


Fig.1

a) *Algorithm for False Position Method :*

Step 1: Take input data like initial guesses a and b and no. of iterations.

Step 2: Define function $y = f(x)$

Step 3: Calculate $y_1 = f(a)$ and $y_2 = f(b)$

Step 4: Check whether $f(a) \cdot f(b) < 0$ or not

Step 5: If not then enter a and b again. Again calculate $y_1 * y_2 < 0$ or not

The process is repeated till $y_1 * y_2 < 0$

Step 6: For 1st iteration i.e. $n = 1$ use RF formula for finding 1st root.

$$\text{Formula } c = \frac{[a \cdot f(b) - b \cdot f(a)]}{[f(b) - f(a)]}$$

Step 7: After finding c find $f(c) = y_3$

Step 8: Check the condition whether $y_1 * y_3 < 0$ or not

If yes then replace $b = c$

If not then $y_2 * y_3 < 0$ so replace $a = c$

Step 9: If $f(a) \cdot f(b) > 0$ then find correct interval and find the correct answer using formula.

Step 10: Repeat the process till we get correct answer.

b) Convergence of False Position Method:

Convergence of false position technique the false position formula relies on linear interpolation technique. Within the false position iteration, one in every of the place to begin is mounted whereas the opposite moves toward the answer. Assume that initial purpose breaking the answer are a and b which a move toward the answer and b is mounted as illustrated Let

$x_i = a$ and x_r be the solution.

Then,

$$e_1 = x_r - x_1$$

$$e_2 = x_r - x_2$$

That is,

$$e_i = x_r - x_i$$

It can be shown that

$$e_{i+1} = e_r \times \frac{(x_r - b)f''(R)}{f''(R)}$$

Where R is a few points within the interval within the a and b. This shows that the method of iteration converges linearly. [2]

c) Advantages of False Position Method:

1. It doesn't needed the spinoff calculation.
2. This methodology has 1st order rate convergence i.e. it's linearly focused. It forever converges.

d) Disadvantages of False Position Method:

1. Because it is trial and error methodology in some categories it's going to take massive timespan to calculate the right root and their by showing down the method.
2. It's accustomed calculate solely one unknown within the equation.

III. Newton-Raphson(N-R) method

The purpose of using this method to find approximate/more accurate root of equation. This method is derived from Taylor's series.

Let x_0 be an approximate root of $f(x) = 0$ and let $x_1 = x_0 + h$ be the correct root of $f(x) = 0$ Therefore, $f(x_1) = 0$ or $f(x_0 + h) = 0$.

Expanding $f(x_0 + h)$ by Taylor's series, we get

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \dots = 0$$

Neglecting the second and higher order derivatives (as h is small), we have

$$f(x_0) + hf'(x_0) = 0$$

$$h = -f\left(\frac{x_0}{f'(x_0)}\right)$$

Therefore, a better approximation than x_0 is given by x_1 where,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

By successive approximations, one can get the following general formula,

$$\boxed{x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}} \quad (i = 0, 1, 2, \dots, n)$$

This is called Newton-Raphson formula and it may be used to find the roots of the algebraic and transcendental equations.

The procedure of Newton-Raphson method may be explained geometrically,

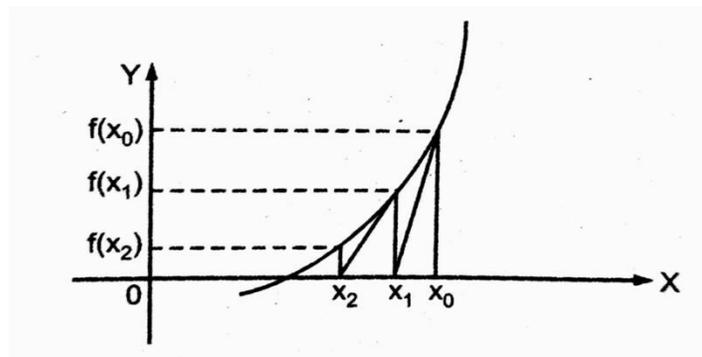


Fig.2

The tangent to the curve $f(x)$ is drawn at $x = x_0$ and the point where it cuts the X-axis call it x_1 , is the first approximation. Again, draw the tangent to the curve at $x = x_1$ and the point where it cuts X-axis, call it x_2 , this is second approximation. This procedure is repeated till $f(x_i)$ becomes nearly zero.

Geometrical interpretation of Newton-Raphson method shows that the proper choice of **initial guess** of root is very important that is if we start with the wrong initial value the Newton-Raphson iterative method gives worse approximation or leads to an endless cycle. This is explained in above fig.

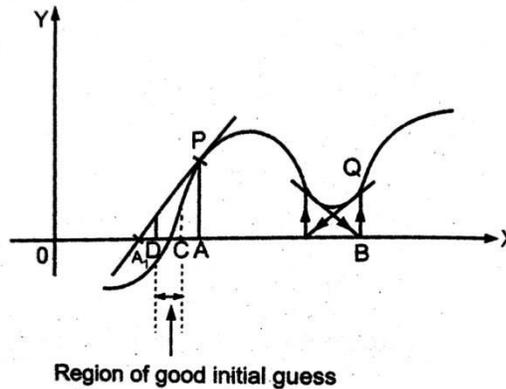


Fig.3

From the above figure, we see that if we start from the point A that is initial value is taken to be the value of point A, then the tangent at P will meet X-axis at A1, which is wrong approximation.

Again, if we start from the point B and tangent is drawn at Q which gives endless cycle. So, the correct alternative of initial curve is extremely vital. When we take any initial value between C and D, we can get the proper approximation at every stage. The region in between C and D is claimed because the region of excellent initial guess. [3]

a) Algorithm for Newton-Raphson Method:

Step 1: Define function. $y = f(x)$

Step 2: Find the derivative of given function $y = f(x)$

Step 3: Take input data i.e. values of a and b if given.

Step 4: If not, find values of a and b such that $f(a) \cdot f(b) < 0$.

Step 5: Input value n (number of iterations).

Step 6: Guess initial iterative value c or find c by using $c = \frac{(a+b)}{2}$.

Step 7: Find functional value at Point c i.e. $f(c)$.

Step 8: Find derivative value at c i.e. $f'(c)$.

Step 9: Perform first iteration by N-R formula $x_1 = c - \frac{f(c)}{f'(c)}$

Step 10: Find $f(x_1)$ also $f'(x_1)$.

Step 11: Perform next iteration by using general formula $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$.

Step 12: Repeat the process till we get correct routes up to two or three decimal places.

b) Convergence of Newton-Raphson Method:

Let x_i be an estimate of a root of the function $f(x)$. If x_i and x_{i+1} are close to each other, then using Taylor's series expansion, we can state

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(R)}{2}(x_{i+1} - x_i)^2 \quad (1)$$

Where R lies somewhere in the interval x_i to x_{i+1} and third and higher order have been dropped.

Let us assume that the exact root of $f(x)$ is x_r . Then $x_{i+1} = x_r$. Therefore $f(x_{i+1}) = 0$ and substituting these values in equation (1), we get

$$0 = f(x_i) + f'(x_i)(x_r - x_i) + \frac{f''(R)}{2}(x_r - x_i)^2 \quad (2)$$

We know that the Newton's iterative formula is given by

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Rearranging the terms, we get

$$f(x_i) = f'(x_i)(x_i - x_{i+1})$$

Substituting this for $f(x_i)$ in Eq. (2) yields

$$0 = f'(x_i)(x_r - x_{i+1}) + \frac{f''(R)}{2}(x_r - x_i)^2 \quad (3)$$

We know that the error in the estimate x_{i+1} is given by

$$e_{i+1} = x_r - x_{i+1}$$

Similarly,

$$e_i = x_r - x_i$$

Now, equation (3) can be expressed in terms of these errors as

$$0 = f'(x_i)e_{i+1} + \frac{f''(R)}{2}e_i^2$$

Rearranging the terms we get,

$$e_{i+1} = -\frac{f''(R)}{2f'(x_i)}e_i^2 \quad (4)$$

Equation (4) shows that the error is roughly proportional to the square of the error in the previous iteration. Therefore, the Newton-Raphson technique is alleged to own quadratic convergence. [2]

c) Advantages of Newton-Raphson Method:

1. The advantage of the strategy is its order of convergence is quadratic
2. Convergence rate is one in all the quickest once it will converges.
3. Simple to convert to multiple dimensions.
4. Are often accustomed “polish” a root found by different strategies.

d) Disadvantages of Newton-Raphson Method:

1. Its convergence is not secured. So, sometimes, for given equation and for given guess we won't get resolution.
2. Division by zero draw back can occur.
3. Root jumping may occur thereby not getting meant resolution.
4. Inflection purpose issue may occur.

IV. Numerical Experiments

- 1) Find a root of an equation $x^3 + x - 1 = 0$ upto 4 decimal point using False Position method (regula falsi method) and Newton Raphson (N-R) method.

Solution:-

$$\text{Function } f(x) = x^3 + x - 1 = 0$$

Enter the lower limit: 0

Enter the upper limit: 1

$$f(0).f(1) < 0$$

Therefore, the roots of equation lies between 0 and 1.

Scilab Programming Code for False Position method:

```
a=input("enter the lower limit:")
```

```
b=input("enter the upper limit:")
```

```

n=input("Enter the no. of iterations:")

function f=fun(x)

    f=(x^3)+(x)-1

endfunction

if (fun(a)*fun(b)<0) then

    for i=1:n;

        c=((a*fun(b))-(b*fun(a)))/(fun(b)-fun(a))

        if (fun(a)*fun(c)<0) then

            b=c;

        end

        if (fun(b)*fun(c)<0) then

            a=c;

        end

    end//for

end//upper if

if (fun(a)*fun(b)>0) then

    disp("please enter the correct intervals")

    break

end //if

```

Table 1.Iteration Data for False Position Method

False Position Method						
Iteration	<i>a</i>	<i>f(a)</i>	<i>b</i>	<i>f(b)</i>	<i>c</i>	<i>f(c)</i>
1	0	-1	1	1	0.5	-0.375
2	0.5	-0.375	1	1	0.6364	-0.1058
3	0.6364	-0.1058	1	1	0.6712	-0.02642
4	0.6712	-0.02642	1	1	0.6797	-0.006284
5	0.6797	-0.006284	1	1	0.6816	-0.001743

6	0.6816	-0.001743	1	1	0.6821	-0.0005459
7	0.6821	-0.0005459	1	1	0.6822	-0.0003063
8	0.6822	-0.0003063	1	1	0.6823	-0.00006664
9	0.6823	-0.00006664	1	1	0.6823	-0.00006664

So, we stop finding value of x after 9th iteration because we get same result up to 4 decimal point. So approximate root of the equation $x^3 + x - 1 = 0$ using Regula-Falsi method is 0.6823.

Scilab Programming Code for N-R method:

```

disp("Function f(x)=x^3+x-1");
a=input("Enter the lower limit:");
b=input("Enter the upper limit:");
n=input("Enter the no of iteration:");
c=(a+b)/2
for i=1:1:n
fc=((c*c*c*c)+(c*c)-80);
f_c=((4*c*c*c*c)+(2*c));
x=c-(fc/f_c)
c=x;
end

```

Table 2.Iteration Data for Newton-Raphson Method

N-R method				
Iteration	x_i	$f(x_i)$	$f'(x_i)$	x_{i+1}
1	0.5	-0.375	1.75	0.7142
2	0.7142	0.07850	2.5302	0.6832
3	0.6832	0.002092	2.4003	0.6823
4	0.6823	0.00006664	2.3966	0.6823

Here, we stop finding value of x after 4th iteration because we get same result up to 4 decimal point. So approximate root of the equation $x^3 + x - 1 = 0$ using Newton-Raphson method is 0.6823.

- 2) Find a root of an equation $\log(x) = \cos(x)$ upto 4 decimal point using False Position method (regula-falsi method) and Newton Raphson (N-R) method.

Solution:

$$\text{Function } f(x) = \log(x) - \cos(x)$$

Enter the lower limit: 1

Enter the upper limit: 2

$$f(1) \cdot f(2) < 0$$

Therefore, the roots of equation lies between 1 and 2.

Scilab Programming code for False Position method:

```
a=input("enter the lower limit:")
b=input("enter the upper limit:")
n=input("Enter the no. of iterations:")
function f=fun(x)
    f=(log(x))-cos(x)
endfunction
if (fun(a)*fun(b)<0) then
    for i=1:n;
        c=((a*fun(b))-(b*fun(a)))/(fun(b)-fun(a))
        if (fun(a)*fun(c)<0) then
            b=c;
        end
        if (fun(b)*fun(c)<0) then
            a=c;
        end
    end
end
```

```

end//for

end//upper if

if (fun(a)*fun(b)>0) then

    disp("please enter the correct intervals")

    break

end //if
    
```

Table 3.Iteration Data for False Position Method

False Position method						
Iterations	<i>a</i>	<i>f(a)</i>	<i>b</i>	<i>f(b)</i>	<i>c</i>	<i>f(c)</i>
1	1	-0.5403	2	0.7172	1.4297	0.01462
2	1	-0.5403	1.4297	0.01462	1.4184	-0.000008397
3	1.4184	-0.000008397	2	0.7172	1.4184	-0.000008397

Here, we stop finding value of x after 3rd iteration because we get same result up to 4 decimal point. So approximate root of the equation $\log(x) = \cos(x)$ using False Position method is 1.4184.

Scilab Programming code for N-R method:

```

disp("Function f(x)=log(x)-cos(x)");

a=input("Enter the lower limit:");

b=input("Enter the upper limit:");

n=input("Enter the no of iteration:");

c=(a+b)/2

for i=1:1:n

    fc=log(c)-cos(c);

    f_c=(1/c)+sin(c);

    x=c-(fc/f_c)
    
```

```
c=x;
```

```
end
```

Table 4. Iteration Data for Newton-Raphson Method

N-R method				
Iteration	x_i	$f(x_i)$	$f'(x_i)$	x_{i+1}
1	1.5	0.1054	1.6642	1.4367
2	1.4367	0.02367	1.6871	1.4227
3	1.4227	0.005558	1.6919	1.4194
4	1.4194	0.001286	1.6931	1.4186
5	1.4186	0.0002505	1.6933	1.4184
6	1.4184	0.000008397	1.6934	1.4184

Here, we stop finding value of x after 6th iteration because we get same result up to 4 decimal point. So approximate root of the equation $\log(x) = \cos(x)$ using Newton-Raphson method is 1.4184.

- 3) Find a root of an equation $x \sin(x) + \cos(x) = 0$ upto 4 decimal point using False Position method (regula-falsi method) and Newton Raphson (N-R) method.

Solution:

$$\text{Function } f(x) = x \sin(x) + \cos(x)$$

Enter the lower limit: 2

Enter the upper limit: 3

$$f(2) \cdot f(3) < 0$$

Therefore, the roots of equation lies between 2 and 3.

Scilab Programming code for False Position method:

```
a=input("enter the lower limit:")
```

```
b=input("enter the upper limit:")
```

```
n=input("Enter the no. of iterations:")
```

```
function f=fun(x)
    f=(x*sin(x))+cos(x)
endfunction
if (fun(a)*fun(b)<0) then
    for i=1:n;
        c=((a*fun(b))-(b*fun(a)))/(fun(b)-fun(a))
        if (fun(a)*fun(c)<0) then
            b=c;
        end
        if (fun(b)*fun(c)<0) then
            a=c;
        end
    end//for
end//upper if
if (fun(a)*fun(b)>0) then
    disp("please enter the correct intervals")
    break
end //if
```

Table 5. Iteration Data for False Position Method

False Position Method						
Iteration	<i>a</i>	<i>f(a)</i>	<i>b</i>	<i>f(b)</i>	<i>c</i>	<i>f(c)</i>
1	2	1.4024	3	-0.5666	2.7122	0.2199
2	2.7122	0.2199	3	-0.5666	2.7926	0.01521
3	2.7926	0.01521	3	-0.5666	2.7980	0.001017
4	2.7980	0.001017	3	-0.5666	2.7984	-0.00003677
5	2	1.4024	2.7984	-0.00003677	2.7984	-0.00003677

Here, we stop finding value of x after 5th iteration because we get same result up to 4 decimal point. So approximate root of the equation $x \sin(x) + \cos(x) = 0$ using False Position method is 2.7984.

Scilab Programming code for N-R method:

```

disp("Function f(x)=x*sin(x)+cos(x)");
a=input("Enter the lower limit:");
b=input("Enter the upper limit:");
n=input("Enter the no of iteration:");
c=(a+b)/2
for i=1:1:n
fc=c*sin(c)+cos(c);
f_c=c*cos(c);
x=c-(fc/f_c)
c=x;
end

```

Table 6. Iteration Data for Newton-Raphson Method

N-R method				
Iteration	x_i	$f(x_i)$	$f'(x_i)$	x_{i+1}
1	2.5	0.6950	-2.0028	2.8470
2	2.8470	-0.1303	-2.7243	2.7992
3	2.7992	-0.002146	-2.6367	2.7984
4	2.7984	-0.00003677	-2.6352	2.7984

Here, we stop finding value of x after 4th iteration because we get same result upto 4 decimal point. So approximate root of the equation $x \sin(x) + \cos(x) = 0$ using Newton-Raphson method is 2.7984.

V. Result and Discussion

Scilab output results are depicted in table

Table 7. Comparison Table

Iteration N	$f(x) = x^3 + x - 1$	$f(x) = \log(x) - \cos(x)$	$f(x) = x \sin(x) + \cos(x)$

	R-F method c	N-R method x_{i+1}	R-F method c	N-R method x_{i+1}	R-F method c	N-R method x_{i+1}
1	0.5	0.7142	1.4297	1.4367	2.7122	2.8470
2	0.6364	0.6832	1.4184	1.4227	2.7926	2.7992
3	0.6712	0.6823	1.4184	1.4194	2.7980	2.7984
4	0.6797	0.6823		1.4186	2.7984	2.7984
5	0.6816			1.4184	2.7984	
6	0.6821			1.4184		
7	0.6822					
8	0.6823					
9	0.6823					

Here, we've got presented the 2 iterative methods in solving non-linear equations in SCILAB environment. The SCILAB outputs of the optimal solutions are provided to test the effectiveness of the proposed method and compared the outcomes.

VI.CONCLUSION

The present paper defined two kinds of non-linear equations have tables with step of iteration. And stated variety of methods to seek out the roots of such equations and also we discussed the rate of convergence, advantages and disadvantages of iterative methods their examples with programming code. We noticed that theoretically, Table shows that among method of False Position, its first-order convergent. There are two major differences in both method, Newton's method requires two function evaluations per iterate, that of $f(x_i)$ and $f'(x_i)$, whereas the False Position method requires only one function evaluation per iterate, that of $f(c)$. Newton's method generally more expensive per iteration. On other hand, Newton's method converges more rapidly and consequently it will require fewer iterations to attain a given desired accuracy. Further, Newton-Raphson has the order of convergence 2, which is that the greater of other one. As we increase the number of iterations we get more accurate values of root. The foremost commonly used method is that the Newton-Raphson method. Hence Newton-Raphson is handiest than False Position methods. [5]

VII. REFERENCES

- [1] An Introduction to Numerical Analysis, by Kendall E. Atkinson, Second Edition 2012.
- [2] Numerical Methods, by Balagurusamy, 38th Reprint 2014.
- [3] Numerical Techniques, by M. D. Bhagat, R. S. Bhamare, N. M. Phatangare, Dr. S. G. Purane, Dr. A. S. Khairnar, First Edition: July 2020.
- [4] Ehiwario, J.C., Aghamie, S.O., Comparative Study of Bisection, Newton-Raphson and Secant Methods of Root-Finding Problems, Journal of IOSR Journal of Engineering (IOSRJEN), Vol. 04, Issue04 (April. 2014), ||V1|| PP 01-07
- [5] Kinnari C Patel, Janki M Patel, Analogical Study of Newton-Raphson Method & False Position Method International Journal of Creative Research Thoughts (IJCRT), | Volume 8, Issue 4 April 2020 | ISSN: 2320-2882.
- [6] Srivastava, R.B and Srivastava, S (2011), Comparison of Numerical Rate of Convergence of Bisection, Newton and Secant Methods. Journal of Chemical, Biological and Physical Sciences. Vol 2(1) pp 472- 479.