Comparative Study on Numerical Approximation of Roots using Bisection and Newton Iterative Method

[1]Kodathala Sai Varun

Department of Electronics and Communication Engineering,

GITAM School of Technology, GITAM deemed to be University – Bengaluru

Email: kodathalasaivarun@gmail.com

Abstract- This paper presents the methods used to approximate the zeros of an equation which can be linear, non-linear, polynomial, exponential, logarithmic and trigonometric collectively known as transcendental equations. The methods utilized are bracketing method and open method which are bisection method and newton iterative method respectively. In order to analyze the results and verify the approximation the polynomial and exponential expressions are considered. The comparison of bisection and newton iterative is observed on basis of number of iterations and accuracy. And It is evident that newton iterative method is better than bisection method in terms of accuracy and number of iterations. The convergence of newton iterative and bisection method is compared, it has been evident that newton iterative method does not guarantee a root where bisection does. MATLAB package is used for results and plotting graphs.

Index Terms- bisection, Newton – Raphson, convergence, root of transcendental equations, algebraic equations

1. Introduction

In scientific oriented applications it is often required to solve the nonlinear equations of higher order where the analytical method is failed due its constructed approach. In such cases it is essential to approximate the solution considering the better accuracy as per required constraints [1]. Therefore it is observed that numerical methods are essential to be employed in order to accurate the approximation[2].

Most of the numerical methods are iterative methods where definite computations approaches to the solution, and the numerical methods are basically approximation methods where $f(x) \cong 0$. But when an analytical method is considered it determines the exact root where $f(x) = 0$. In most of real world implications like in the streams of chemistry, physics, economics, mechanics [3-8] we could not solve the equations using the analytical method because we are not defined by the boundary and the degree of the equation is high which are impractical to solve, but the same equations can be approximated by using numerical methods.

1.1. Bisection Method

The formal and simplest method to determine the root of an equation is bisection method. It could be used only if the function is continuous and the range of equation is known. Here the range of equation corresponds to the sign change in the value of function. That is if the range is given as $[a, b]$, then the function follows the following property-

$sign(f(a)) = -sign(f(b))$. If $f(a)$ is positive then $f(b)$ is negative and vice versa.

Then the approximation of first iteration is given by the mean of the maximum and minimum values, therefore the root is given for first iteration is $r = \frac{a+b}{2}$. Then the $f(r)$ is computed and corresponding range is computed. That is if the function lie in the range $\left[a, \frac{a+b}{2}\right]$ the range is modified from $[a, b]$ to $\left[a, \frac{a+b}{2}\right]$ and if the function lie in the range $\left[\frac{a+b}{2}, b\right]$ then the range is modified from $[a, b]$ to $\left[\frac{a+b}{2}, b\right]$. For the second iteration again the mean of the maximum and minimum is taken and computed for its range and the procedure is repeated until the accurate result is achieved [9].

1.1.1. Advantage of the Bisection Method

The convergence of root is certain though it is slow process because the method is implied by halving the length of bracketing interval containing the root.

1.1.2. Disadvantage of the Bisection Method

Though the convergence of this method is guaranteed, its rate of convergence is slow and it is difficult to extend the use for system of equations.

1.2. Newton Raphson Method

Newton – Raphson method is also known as fixed – point iteration method. This method requires that the function be differentiable. If the function f(x) is differentiable on the domain of function, and r_0 is the initial guess, then the first approximation r_1 is obtained by intersection of tangent at $(r_0, f(r_0))$ with the x-axis, and is defined by [10]

$$r_1 = r_0 - \frac{f(r_0)}{f'(r_0)}$$

And the successive approximations are

$$r_n = r_{n-1} - \frac{f(r_{n-1})}{f'(r_{n-1})}$$

2.1. Advantage of Newton – Raphson Method

The convergence rate is linear and this method is very fast compared to the bisection, false position and secant method.

2.2. Disadvantage of Newton – Raphson Method

The only drawback of Newton Raphson method is it fails if the derivative, $f'(x)$ is near zero at some iteration.

The error which is used as an parameter in this paper is formulated as follows

$$\in_r = \frac{(\textbf{present root} - \textbf{past root})}{\textbf{present root}} * \mathbf{100}\ \%$$

2. METHODOLOGY

The Bisection method and Newton Raphson method are explained using theorem 1 and theorem 2 respectively.

Theorem 1. *Let f(x) be a continuous function and (a,b) be a sufficiently small interval such that f(a)f(b)<0. Then the approximation of a root of f(x) can be determined using the formula given by*

$$r_n = \frac{a+b}{2}$$

If $f(r_n) < 0$ then $a = r_n$ and

If $f(r_n) > 0$ then $b = r_n$. The procedure is iterated till root is approximated to near root.

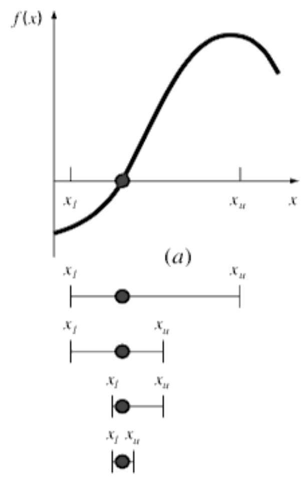


Figure 1: Bisection Method abstraction

Illustration 2.1. consider an equation

$$x^6 - x - 1 = 0$$

This equation has two real roots -0.7780895987 and 1.134724138. The following Table 1 shows the iterations with initial approximations *a = 1* and *b = 1.5*.

Table 1. Approximation using Bisection Method

Iterations	Root
1	1.25000000000000
2	1.12500000000000
3	1.18750000000000
4	1.15625000000000
5	1.14062500000000
6	1.13281250000000
7	1.13671875000000
8	1.13476562500000
9	1.13378906250000
10	1.13427734375000
11	1.13452148437500
12	1.13464355468750
13	1.13470458984375
14	1.13473510742188

One can observe from Table 1 that the bisection method gives approximate root up to 4 decimal points after 14 iterations.

Theorem 2. *Let f(x) be a continuous function and (a,b) be a sufficiently small interval such that f(a)f(b) < 0 and the f'(x) exists for the range. Then the approximation of a root of f(x) can be determined using the iterative formula given by*

$$r_0 = a$$
$$r_1 = b$$

$$r_2 = r_1 - \frac{f(r_1)}{f'(r_1)}$$

$$r_n = r_{n-1} - \frac{f(r_{n-1})}{f'(r_{n-1})}$$

Consider the same illustration given by 2.1. The following Table 2 shows the iterations.

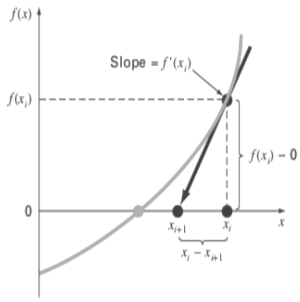


Figure 2: Newton Iterative Method abstraction

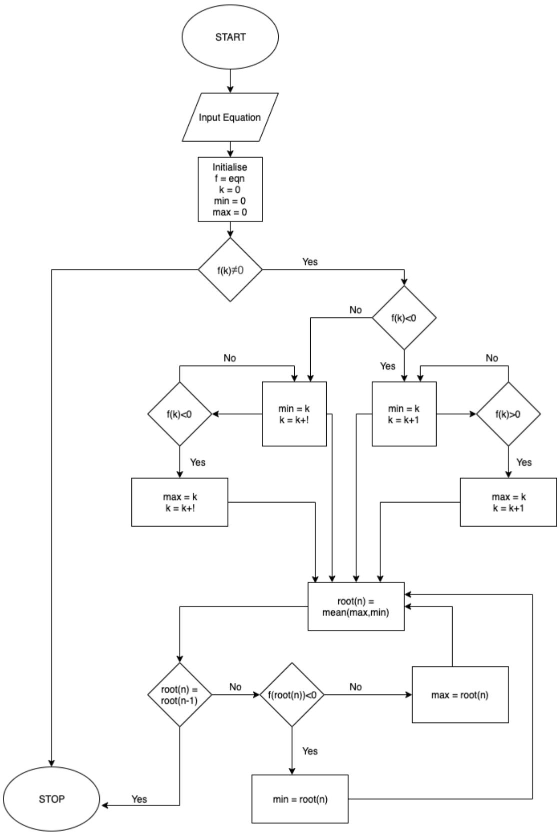
Table 2. Approximation using Newton Raphson Method

Iterations	Root
1	1.68062827225131
2	1.43073898823906
3	1.25497095610944
4	1.16153843277331
5	1.13635327417051
6	1.13473052834363
7	1.13472413850022
8	1.13472413840152
9	1.13472413840152
10	1.13472413840152
11	1.13472413840152
12	1.13472413840152
13	1.13472413840152
14	1.13472413840152

One can observe from Table 2 that the Newton Raphson method gives approximate root up to 4 decimal points after 6 iterations.

3. Implementation of Bisection Method

3.1. Flow Chart:



3.2. Algorithm:

Algorithm 1: Bisection Method

Input: f, [a,b], iterations
Output: root r
//Initialize
k= 0
repeat
//Computing the mid point
$r_k = \frac{a+b}{2}$
//Compute the function value of root
if $f(r_k) > 0$
b = r_k
endif
if $f(r_k) < 0$
a = r_k
endif
iterationCount = k
$r = r_k$
k = k+1
until k > iterations

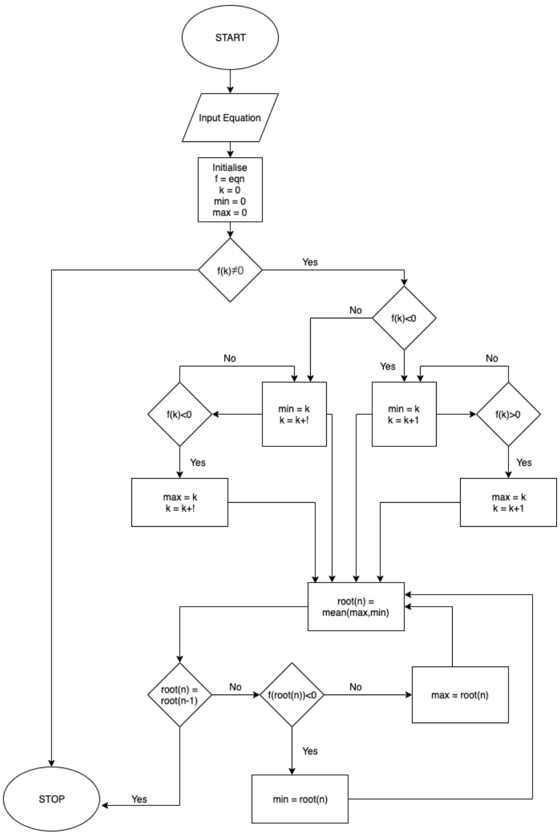
3.3 MATLAB program:

```
function [root,soln] = bisection_method(eqn, itr,k)
%BISECTION_METHOD
%
%Function used to evaluate root according to bisection method
%
%Input Parameters are Equation and Number of Iterations
%
%Equation should be in format of @(x) x.^3-x-11
%
%k can be any value, if unknown set k = 0;
f = eqn; %assignment of equation to dynamic variable
[min_range,max_range] = range_calc(eqn,k); %Returns the range of equation
for i=1:itr %Iterate the loop based on iterations
    range = [min_range,max_range]; % Conversion of scalar to vector
    root = mean(range); % Calculation of average of minimum and maximum value
    if(f(root)<0) % Test the condition for f(root) to be negative
        min_range = root; %Set the minimum range to root
    end
    if(f(root)>0)
        max_range = root; %Set the maximum range to root
    end
end
root = vpa(root); % To precise the output
```

```
soln = f(root); % To Evaluate the
solution
end
```

4. Implementation of Newton Raphson Method

4.1. Flow chart:



4.2. Algorithm:

Algorithm 2: Newton Raphson Method

Input: f, [a,b], iterations
Output: root r
//Initialize
k= 0
repeat
  //Computing the Tangent Line
  $r_k = r_{k-1} - \frac{f(r_{k-1})}{f'(r_{k-1})}$
  iterationCount = k
  $r = r_k$
  k = k+1
until k > iterations

4.3. MATLAB Program:

```
function [r, soln] =
newton_iterative(eqn,itr)
%NEWTON_ITERATIVE
%
%Function used to evaluate root
according to Newton Iterative
method
%
%Input Parameters are Equation and
Number of Iterations
%
%Equation should be in format of
@(x) x.^3-x-11
%
%k can be any value, if unknown
set k = 0;
f = eqn; %assignment of equation
to dynamic variable
[min_range,max_range] =
range_equation(eqn); %Returns the
range of equation
for n = 1:itr
    g = derivest(eqn,max_range);
%Evaluates the differentiation of
function
    root(n) = (max_range -
(f(max_range)/g)); %Computes
newton iterative equation
    max_range = root(n); %Replace
the max value with root
end
root = vpa(root(n)); % To precise
the output
soln = f(root(n)); % To Evaluate
the solution
end
```

5. Results

To evaluate the results and perform the comparison among newton iterative and bisection, several numerical examples are considered. The formulations are given in Tables

Numerical Example 5.1.

Consider the following equation to compute root.

$$x^5 - 8x^4 + 39x^3 - 62x^2 + 50x - 10 = 0$$

5.1.1 using bisection method:

Table 3. Approximation using Bisection Method

Iteration	Error	Root
1	NA	0.5000000000
2	100	0.2500000000
3	33.3333333333333	0.3750000000
4	20	0.3125000000
5	11.1111111111111	0.2812500000
6	5.26315789473684	0.2968750000
7	2.70270270270270	0.2890625000
8	1.36986301369863	0.2851562500
9	0.689655172413793	0.2832031250
10	0.346020761245675	0.2822265625
11	0.173310225303293	0.2817382812
12	0.0865800865800866	0.2819824218
13	0.0432713111207270	0.2821044921
14	0.0216309755569976	0.2821655273
15	0.0108143181572402	0.2821960449
16	0.00540686672073533	0.2822113037
17	0.00270350644786288	0.2822036743
18	0.00135177149654622	0.2821998596
19	0.000675890316519 43	0.2821979522
20	0.000337944016194 27	0.2821989059

5.1.2 using Newton Iterative Method:

Table 4. Approximation using Newton Iterative Method

Iteration	Error	Root
1	NA	0.375000000000001
2	39.1366808885	0.269519150237301
3	4.41882807284	0.28197933211688
4	0.0776242741	0.28219838698948
5	2.3447177e-05	0.28219845315508
6	2.1244522e-12	0.28219845317514
7	0	0.28219845357514
8	0	0.28219845157514
9	0	0.28219843157514
10	0	0.28219853157514
11	0	0.28219453157514
12	0	0.28218453157514
13	0	0.28298453157514
14	0	0.28219845315751
15	0	0.28219845315751
16	0	0.28219845315751
17	0	0.28219845315751
18	0	0.28219845315751
19	0	0.28219845315751
20	0	0.28219845315751

Numerical Example 5.2.

$$xe^x - \cos x = 0$$

5.2.1 using bisection method

Table 5. Approximation using Bisection Method

Iteration	Error	Root
1	NA	0.50000000000
2	33.333333333	0.75000000000
3	20	0.62500000000
4	11.11111111	0.56250000000
5	5.8823529411	0.53125000000
6	3.0303030303	0.51562500000
7	1.49253731343284	0.52343750000
8	0.75187969924812	0.51953125000
9	0.37735849056603	0.51757812500
10	0.18832391713747	0.51855468755
11	0.09425070688030	0.51806640625
12	0.04714757190004	0.51782226562
13	0.02357934449422	0.51770019531
14	0.01178828244724	0.51776123046
15	0.00589448865310	0.51773071289
16	0.00294715746662	0.51774597167
17	0.00147355701928	0.51775360107
18	0.00073677308125	0.51775741577
19	0.00036838789772	0.51775550842
20	0.000184193609	0.51775646209

5.2.2 using newton iterative method

Table 6. Approximation using Newton Iterative Method

Iteration	Error	Root
1	NA	0.653079403526
2	22.91099190	0.531343367606
3	2.593782071	0.5179099131356
4	0.0294597384	0.5177573831648
5	3.762838394e-06	0.5177573636824
6	6.43279161e-14	0.5177573636824
7	0	0.5177573636824
8	0	0.5177573636824
9	0	0.5177573636824
10	0	0.5177573636824
11	0	0.5177573636824
12	0	0.5177573636824
13	0	0.5177573636824
14	0	0.5177573636824
15	0	0.5177573636824
16	0	0.5177573636824
17	0	0.5177573636824
18	0	0.5177573636824
19	0	0.5177573636824

20	0	0.5177573636824

Numerical Example 5.3.

$$4e^{-x}\sin x - 1 = 0$$

5.3.1 using bisection method

Table 7. Approximation using Bisection Method

Iteration	Error	Root
1	NA	0.50000000000
2	100	0.250000000000
3	33.3333333333333	0.375000000000
4	20	0.312500000000
5	9.09090909090909	0.343750000000
6	4.34782608695652	0.359375000000
7	2.12765957446809	0.367187500000
8	1.05263157894737	0.371093750000
9	0.529100529100529	0.369140625000
10	0.263852242744063	0.370117187500
11	0.131752305665349	0.370605468750
12	0.065919578114700	0.370361328125
13	0.032948929159802	0.370483398437
14	0.016471750947125	0.370544433593
15	0.008235197232973	0.370574951171
16	0.004117768169652	0.370559692382
17	0.002058926475735	0.370552062988
18	0.001029452640031	0.370555877685
19	0.000514723670597	0.370557785034
20	0.000257361172949	0.370558738708

5.3.2 using Newton Iterative method

Table 8. Approximation using Newton Iterative Method

Iteration	Error	Root
1	NA	1.53757425328
2	12.3979819592523	1.36797318464
3	0.220693812475089	1.36496080061
4	0.00017340299543	1.36495843373
5	1.08309010959e-10	1.36495843373
6	0	1.36495843373
7	0	1.36495843373
8	0	1.36495843373
9	0	1.36495843373
10	0	1.36495843373
11	0	1.3649584337
12	0	1.3649584337
13	0	1.3649584337
14	0	1.3649584337
15	0	1.3649584337
16	0	1.3649584337
17	0	1.3649584337
18	0	1.3649584337
19	0	1.3649584337
20	0	1.3649584337

Numerical Example 5.4.

$$x^3 - 2x - 5 = 0$$

5.4.1 using bisection method

Table 9. Approximation using Bisection Method

Iteration	Error	Root
1	NA	2.50000000000
2	11.1111111111111	2.25000000000
3	5.88235294117647	2.12500000000
4	3.03030303030303	2.06250000000
5	1.49253731343284	2.09375000000
6	0.74074074074074	2.10937500000
7	0.37174721189591	2.101562500000
8	0.18621973929236	2.097656250000
9	0.09319664492078	2.095703125000
10	0.04662004662004	2.094726562500
11	0.02331545814875	2.094238281250
12	0.01165637020631	2.094482421875
13	0.00582784544553	2.094604492187
14	0.00291400763470	2.094543457031
15	0.00145698258905	2.09457397460938
16	0.00072849660156	2.09455871582031
17	0.00036424962755	2.09455108642578
18	0.0001821244820	2.09455490112305
19	9.106232350e-05	2.09455299377441
20	4.557134490e-05	2.09455204010010

5.4.2 using Newton Iterative method

Table 10. Approximation using Newton Iterative Method

Iteration	Error	Root
1	NA	2.36000000000
2	10.9441318270425	2.12719678015
3	1.530246373505	2.09513603693
4	0.027899197554	2.09455167382
5	9.1899812659e-06	2.09455148154
6	9.9649922659e-13	2.09455148154
7	0	2.09455148154
8	0	2.09455148154
9	0	2.09455148154233
10	0	2.09455148154233
11	0	2.09455148154233
12	0	2.09455148154233
13	0	2.09455148154233
14	0	2.09455148154233
15	0	2.09455148154233
16	0	2.09455148154233
17	0	2.09455148154233
18	0	2.09455148154233
19	0	2.09455148154233
20	0	2.09455148154233

Numerical Example 5.5.

$$x^3 - x - 1 = 0$$

5.5.1 using bisection method

Table 11. Approximation using Bisection Method

Iteration	Error	Root
1	NA	1.500000000
2	20	1.250000000
3	9.09090909090909	1.375000000
4	4.76190476190476	1.312500000
5	2.32558139534884	1.343750000
6	1.17647058823529	1.328125000
7	0.591715976331361	1.320312500
8	0.294985250737463	1.324218750
9	0.147275405007364	1.326171875
10	0.073691967575534	1.325195312
11	0.036859565057132	1.324707031
12	0.018426386585590	1.324951171
13	0.009214042200313	1.324829101
14	0.004607233356369	1.324768066
15	0.002303669745905	1.324737548
16	0.001151848140341	1.324722290
17	0.000575927387075	1.324714660
18	0.000287962864309	1.324718475
19	0.000143981639461	1.324716567
20	7.199076790390e-05	1.324717521

5.5.2 using Newton Iterative method

Table 12. Approximation using Newton Iterative Method

Iteration	Error	Root
1	NA	1.545454545454
2	13.6685488930716	1.359614915915
3	2.55042514753145	1.325801345005
4	0.08170000946206	1.324719049417
5	8.244556924e-05	1.324717957245
6	8.3908492194e-11	1.324717957244
7	0	1.324717957244
8	0	1.324717957244
9	0	1.324717957244
10	0	1.324717957244
11	0	1.324717957244
12	0	1.324717957244
13	0	1.324717957244
14	0	1.324717957244
15	0	1.324717957244
16	0	1.324717957244
17	0	1.324717957244
18	0	1.324717957244
19	0	1.324717957244
20	0	1.324717957244

To understand the convergence, consider the following equation

$$x^2 - 1 = 0$$

Table 7. Convergence of Newton Iterative method

Actual Root	Determined using Bisection Method	Determined using Newton Iterative Method
1	1	Failed

6. Discussions

The root approximations is almost the same among the two methods, as expected, yet the number of iterations used to arrive at the solution shows the advantage of newton iterative method. Similarly, through the convergence the advantage of bisection method is shown. From Table 3 – Table 12 the comparison among the two methods are inferred whereas the failure of newton iterative method is inferred from Table 13. Since it is hard to interpret the numbers in the tables of the algorithm outputs, graph plots are preferred way to visualize the algorithm behavior. The plots of the execution of the algorithms are given in Figures 3 – 7. The iterations column shows that the newton iterative method is faster than bisection method.

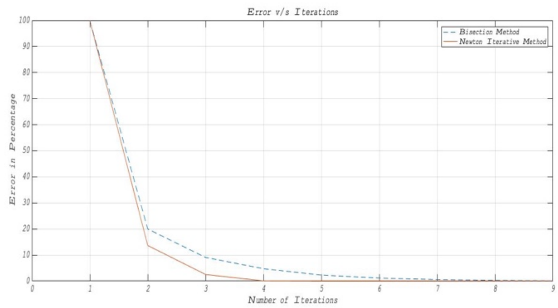


Figure 3: Plot of Error v/s Number of Iterations for Equation $x^3 - x - 1$

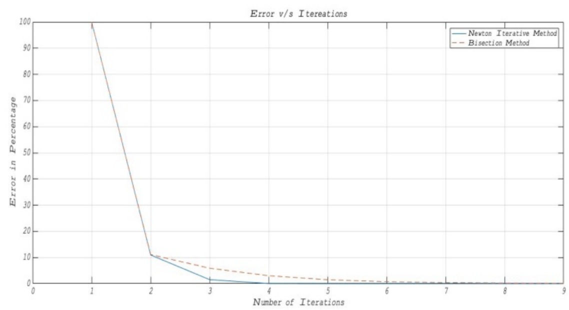


Figure 4: Plot of Error v/s Number of Iterations for Equation $x^3 - 2x - 5$

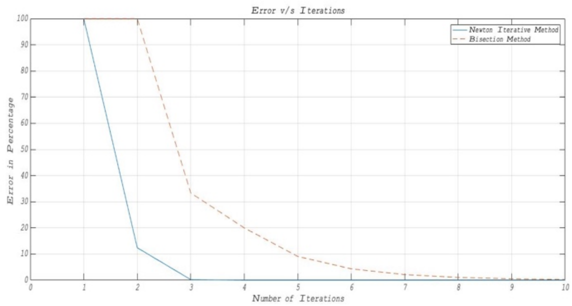


Figure 5: Plot of Error v/s Number of Iterations for Equation $4e^{-x}\sin(x) - 1$

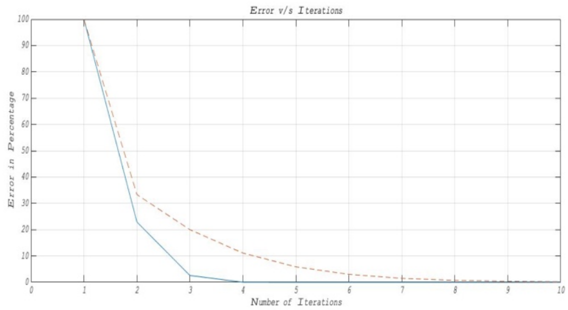


Figure 6: Plot of Error v/s Number of Iterations for Equation $xe^{-x} - \cos(x)$

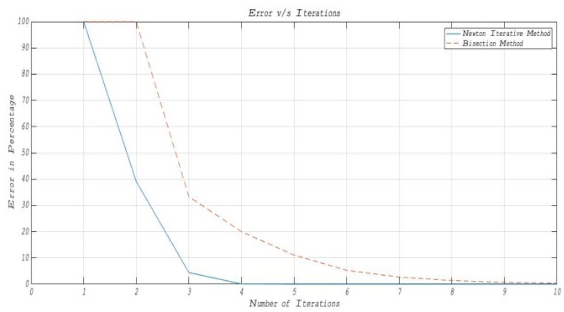


Figure 6: Plot of Error v/s Number of Iterations for Equation $x^5 - 8x^4 + 39x^3 - 62x^2 + 50x - 10$

7. Conclusions

This paper modelled bisection method and newton iterative method using MATLAB package. The count of iterations required to determine the root clearly shows the advantage of newton iterative method. However, as desired and expected, the root approximation is almost the same in both algorithms.

The algorithm was tested on all functions mentioned in illustrations. The results validate that the newton iterative method is effective both conceptually and computationally.

Additional to use of only one algorithm at once, one can make use of blend of both algorithm for effective results.

8. References:

[1] Jaime Gutierrez, Josef Schicho, Martin Weimann, " *Computer Algebra and Polynomials* ", Applications of algebra and number theory, Springer, Cham, 2015.

[2] Richard W. Hamming. 1973. "*Numerical Methods for Scientists and Engineers*". McGraw-Hill, Inc., USA.

[3] Datta, B.N. *Lecture Notes on Numerical Solution of Root Finding Problems*. 2012. Available online: **www.math.niu.edu/~dattab** (accessed on 15 January 2019).

[4] Calhoun, D. Available online: **https://math.boisestate.edu/~calhoun/teaching/matlab-tutorials/lab_16/html/lab_16.html** (accessed on 13 June 2019).

[5] Thinzar, C.; Aye, N. *Detection the storm movement by sub pixel registration approach of Newton Raphson method*. *Int. J. E Educ. E Bus. E Manag. E Learn.* **2014**, *4*, 28–31. [**Google Scholar**]

[6] Ali, A.J.M. *The application of numerical approximation methods upon digital images*. *Am. J. Signal Process.* **2017**, *7*, 39–43. [**Google Scholar**] [**CrossRef**]

[7] Bruck, H.A.; McNeill, S.R.; Sutton, M.A.; Peters, W.H. *Digital image correlation using Newton-Raphson method of partial differential correction*. *Exp. Mech.* **1989**, *29*, 261–267. [**Google Scholar**] [**CrossRef**]

[8] Cofaru, C.; Philips, W.; van Paepegem, W. *Pixel-level robust digital image correlation*. *Opt. Express* **2013**, *21*, 29979–29999. [**Google Scholar**] [**CrossRef**] [**PubMed**]

[9] Claudio Gutierreza, FlavioGutierrezb, Maria-Cecilia Rivaraa, *Complexity of the bisection method*, Theoretical Computer Science Volume 382, Issue 2, 31 August 2007, Pages 131-138

[10] Diachao Sheng, Scott W. Sloan, Andrew J. Abbo, *An Automatic Newton-Raphson Scheme*, International Journal of Geomechanics, Vol. 2, Issue 4 (October 2002)