

# Comparative Study of Numerical Methods for Solving Non-linear Equations Using Manual Computation

Isaac Azure<sup>1</sup>, Golbert Aloliga<sup>2</sup>, Louis Doabil<sup>3</sup>

<sup>1</sup>Department of Computer Science, Regentropfen College of Applied Sciences, Bolgatanga, Ghana

<sup>2</sup>Mathematics Department, St. Vincent College of Education, Yendi, Ghana

<sup>3</sup>Business School, Ghana Institute of Management & Public Administration, Accra, Ghana

## Email address:

azureike@yahoo.com (I. Azure), aloligagolbert@gmail.com (G. Aloliga), ldoabil@gimpa.edu.gh (L. Doabil)

## To cite this article:

Isaac Azure, Golbert Aloliga, Louis Doabil. Comparative Study of Numerical Methods for Solving Non-linear Equations Using Manual Computation. *Mathematics Letters*. Vol. 5, No. 4, 2019, pp. 41-46. doi: 10.11648/j.ml.20190504.11

**Received:** June 30, 2019; **Accepted:** December 25, 2019; **Published:** January 8, 2020

---

**Abstract:** This paper aims at comparing the performance in relation to the rate of convergence of five numerical methods namely, the Bisection method, Newton Raphson method, Regula Falsi method, Secant method, and Fixed Point Iteration method. A manual computational algorithm is developed for each of the methods and each one of them is employed to solve a root - finding problem manually with the help of an TI - inspire instrument. The outcome of the computations showed that all methods converged to an exact root of 1.56155, however the Bisection method converged at the 14th iteration, Fixed Point Iterative Method converged at 7th iteration, Secant method converged at the 5th iteration and Regula Falsi and Newton Raphson methods converged at the 2nd iteration, suggesting that Newton Raphson and Regula Falsi methods are more efficient in computing the roots of a nonlinear quadratic equation.

**Keywords:** Numerical Methods, Convergence, Root, Iteration, Manual Computation, Nonlinear Equations

---

## 1. Introduction

When real life problems are modelled into mathematical equations in an attempt to solve a problem, the equations are often linear or nonlinear in nature. The roots of the equations give the final result to the problem under study. This means that using the most efficient numerical method for root finding problems is very important in mathematical computations since obtaining an accurate result is important in problem solving. The Root - Finding Problem is the problem of finding a root of the equation  $f(x) = 0$ , where  $f(x)$  is a function of the single variable  $x$ . Given a function  $f(x) = 0$  and  $x = \phi$  such that  $f(\phi) = 0$ , then  $\phi$  is a root of  $f(x)$ . Root - Finding Problems arise in several fields of studies including Engineering, Chemistry, Agriculture, Biosciences and so on. This is as a result of the fact that unknown variables will always appear in problem formula involving real life problems. Relevant situations in Physics where such problems are needed to be solved include finding the equilibrium position of an object, potential surface of a field and quantized energy level of confined structure [2]. As a

matter of fact, the determination of any unknown appearing implicitly in scientific or engineering formulas, gives rise to root finding problem [3].

Some numerical methods for solving root -finding problems include; Bisection method, Newton Raphson method, Regula Falsi method, Secant method and Fixed Point Iteration. The rate of convergence could be linear, quadratic or otherwise. The higher the order, the faster the method converges [6]. Several investigations have been carried out by different authors all in an attempt to finding the right methods of solving root – finding problems. One of such researches by Ehiwario et al (2014), investigated the effectiveness of Newton Raphson, Bisection and Secant methods in solving a root finding problems [6]. Prior to Ehiwario et al (2014) investigation, Srivastava et al (2011) carried out a comparative study between Bisection, Newton Raphson and Secant methods to find out the method with the least number of iterations when applied to solve a single variable nonlinear equation [11]. Other numerical methods for solving root –finding problems such as Regula – Falsi and Fixed Point Iteration methods have applied in other researches to solve root – finding problems [5, 8]. The

purpose of the study is to compare the number of iterations needed by a given numerical method to reach a solution and the rate of convergence of the methods. In order to achieve the aim of this study, manual computational algorithms for each of the methods are used to find the root of a function.

A similar research conducted compared four iterative methods for solving non-linear equations that assist to present a better performing method [9]. MATLAB results were delivered to check the appropriateness of the best method. With the help of the approximate error graph, the Newton’s method was the most robust for solving the nonlinear equation. Besides, Newton’s method gave a lesser number of iterations compared to the others and it showed less processing time.

## 2. Numerical Methods

### 2.1. Background of Newton Raphson Method

Newton Raphson method can be derived using several procedures, notable amongst them are by using the concept of finding the slope of a function and using the Taylor series.

#### 2.1.1. Concept of Slope of a Function

Derivation of Newton Raphson method using the concept of slope of a function begins with an assumption that a tangent to a curve that cuts the x - axis gives an idea leading to the computation of likely solutions to the equation of the curve, as illustrated in the diagram below.

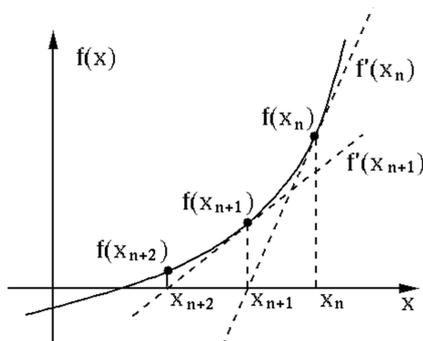


Figure 1. Diagram indicating slope of a curve (<https://images.app.goo.gl/sUAwJLCWzCj8NTN18>).

From the figure above, the slope of  $(x_n, f(x_n))$  is given by

$$f'(x_n) = \frac{f(x_n) - 0}{x_n - x_{n+1}} \tag{1}$$

Making  $x_{i+1}$  the subject in equation (1) results into the equation below, which represents Newton Raphson equation.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2}$$

The method ensures that each iterative solution is updated at every point.

#### 2.1.2. Termination Condition for Newton Raphson Method

The iterative process of Newton Raphson method is terminated at the point when approximate relative error  $\epsilon_a$  is

less than a certain threshold. Hence the relative error formula is given by

$$\epsilon_a = \frac{x_{n+1} - x_n}{x_{n+1}} \times 100 \tag{3}$$

#### 2.1.3. Using Taylor Series

Newton Raphson method can also be derived from Taylor Series. This method has proven to be more useful in computations which has to do with error analysis since Taylor Series gives an easier approach to evaluation of approximation error.

When given a function  $(x_{n+1})$ , its updated value in Taylor Series form is written as follows

$$f(x_{n+1}) \approx f(x_n) + f'(x_n)(x_{n+1} - x_n) \tag{4}$$

the point of intersection of the curve with the x-axis in figure 1, shows that the function  $f(x_{n+1}) = 0$ , hence equation (3) becomes

$$f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0 \tag{5}$$

making  $x_{i+1}$  the subject in equation (4) results into Newton Raphson equation as shown below

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{6}$$

#### 2.1.4. Algorithm for Newton Raphson Method

In order to apply Newton Raphson method to find the root of a nonlinear equation:

Step 1: Write the equation in the form  $f(x) = 0$  and find  $f'(x)$

Step 2: Test for values of  $x$  using the function  $f(x)$  to obtain the range  $(x_0, x_1)$  in which the real root lies.

Step 3: Select  $x_0$  in step 2 as the initial value and solve for  $f(x_0)$  and  $f'(x_0)$

Step 4: Use values obtained in step 3 to compute  $h_0$ , that is  $h_0 = \frac{f(x_0)}{f'(x_0)}$  and hence  $x_1 = x_0 - h_0$ .

Step 5: Use the new  $x_1$  value to repeat the process in steps 3 and 4 to obtain the value for  $x_2$ . Stop when values for  $x_{n+1}$  in two different iterations are the same. That value is the real root of the function  $f(x)$ .

#### 2.1.5. Convergence of Newton Raphson Method

Assuming that  $x_0, x_1, x_2, \dots, x_i$  is a sequence of approximations to a root  $\delta$  obtained by a numerical method, where  $\lim_{n \rightarrow \infty} |x_n - \delta| = 0$

$$\epsilon_n = x_n - \delta$$

if

$$\lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^p} = K$$

for some  $p$  and some non-zero constant  $K$ , the numerical method has an order of convergence  $p$  and  $K$  defines the asymptotic error constant. The value of  $p$  is directly proportional to the rate of convergence.

Discarding  $\epsilon_n^3$  and the higher power of  $n$ , we get

$$\epsilon_{n+1} = K\epsilon_n^2 \tag{7}$$

Where

$$K = \frac{1}{2} \frac{f''(\delta)}{f'(\delta)}$$

Since

$$\frac{|\epsilon_{n+1}|}{|\epsilon_n|^2} = \frac{|f''(\delta)|}{2|f'(\delta)|}$$

Taking limit as  $n \rightarrow \infty$  of both sides of the equation above.

In the case of Newton Raphson method, we assume that the function  $f(x)$  can be differentiated twice, and  $\delta$  is a root of the function. Substituting  $\epsilon_n + \delta = x_n$  into the function  $f(x_n)$  and expanding  $f(\delta + \epsilon_n)$  and  $f'(\delta + \epsilon_n)$  using Taylor's series about the point  $\delta$  where  $f(\delta) = 0$  we get

$$\epsilon_{n+1} = \epsilon_n - \frac{[\epsilon_n f'(\delta) + \frac{1}{2} \epsilon_n^2 f''(\delta) + \dots]}{f'(\delta) + \epsilon_n f''(\delta) + \dots}$$

$$\epsilon_{n+1} = \epsilon_n - \frac{[\epsilon_n + \frac{1}{2} \epsilon_n^2 f''(\delta) + \dots]}{f'(\delta) + \epsilon_n f''(\delta) + \dots}$$

$$\epsilon_{n+1} = \frac{1}{2} \frac{f''(\delta)}{f'(\delta)} \epsilon_n^2 + O(\epsilon_n^3)$$

$$\lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^2} = \lim_{n \rightarrow \infty} \frac{|f''(\delta)|}{2|f'(\delta)|}$$

If  $x_n \rightarrow \delta$  as  $n \rightarrow \infty$ . Hence for the root  $\delta$ , Newton Raphson method has second order - quadratic convergence and an asymptotic error constant of  $\frac{|f''(\delta)|}{2|f'(\delta)|}$ . This means that the square of the previous error  $\epsilon_n$  is proportional to the subsequent error  $\epsilon_{n+1}$ .

### 2.2. Background of Bisection Method

The concept of Bisection method is based on Bolzano's theorem on continuity. Given a function  $f(x) = 0$  such that its root lies in  $[a, b]$ . If  $f(x)$  is real and continuous and  $f(a) \cdot f(b) < 0$ , then there is at least one root between  $a$  and  $b$ . This method is classified under bracketing methods because two initial guesses for the root are required. As the name implies, these guesses must "bracket," or be on either side of, the root. The particular methods described herein employ different strategies to systematically reduce the width of the bracket and, hence, home in on the correct answer [10].

#### 2.2.1. Algorithm for Bisection Method

Step 1: Choose  $a$  and  $b$  as the initial guesses such that  $f(a)f(b) < 0$ .

Step 2: Compute the midpoint of  $a$  and  $b$  such that  $x_m = \frac{a+b}{2}$

Step 3: Find  $f(a)f(x_m)$  if  $f(a)f(x_m) < 0$ , then  $a = a$ ;  $b = x_m$

if  $f(a)f(x_m) > 0$ , then  $a = x_m$ ;  $b = b$

if  $f(a)f(x_m) = 0$  then the root of the function =  $x_m$

Step 4: Find  $x_m = \frac{a+b}{2}$

Step 5: Go back to step 3.

### 2.2.2. Convergence of Bisection Method

In the case of Bisection method, suppose that an algorithm produces iterates that converge as  $\lim_{n \rightarrow \infty} x_n = \delta$ , if there exists a sequence  $y_n$  that converges to zero and a positive constant  $K$ , such that

$$\epsilon_n = |x_n - \delta| \leq k|y_n|$$

it implies that  $x_n$  is said to converge with rate  $y_n$ . Thus in the case of Bisection method

$$|x_n - \delta| \leq |b - a| \frac{1}{2^n}$$

Hence the Bisection method has a convergence rate of  $\frac{1}{2^n}$  with  $|b - a|$  as the asymptotic convergence constant, that  $K = |b - a|$ .

### 2.3. Background of Regular Falsi Method

Regula Falsi method is based on the rational of similar triangles. Its main novelty is that it can be used to compute both zeros and extrema through a single interpolation formula generalized [7]. This method is based on the assumption that the graph of  $y = f(n)$  in the small interval  $[a_n, b_n]$  can be represented by the chord joining  $(a_n, f(a_n))$  and  $(b_n, f(b_n))$ . This implies that at the point  $x = x_n = a_n + h_n$ , at which the chord meets the  $x$ -axis, we obtain two intervals  $[a_n, x_n]$  and  $[x_n, b_n]$ , one of which must contain the root  $\alpha$ , depending upon the condition  $f(a_n)f(x_n) < 0$  and  $f(x_n)f(b_n) < 0$ .

The general Regula Falsi method recurrence relation is given by:

$$x_{n+1} = b_n - \frac{|f(a_n)|}{|f(a_n)+f(b_n)|} (b_n - a_n) \tag{8}$$

#### 2.3.1. Algorithm for Regula Falsi Method

Step 1: Find point  $a_n$  and  $b_n$  such that  $a_n < b_n$  and  $f(a_n)f(b_n) < 0$ .

Step 2: Take the interval  $[a_n, b_n]$  and determine the next value of  $x_n$ .

Step 3: If  $f(x_n) = 0$  then  $x_n$  is an exact root, else if  $f(x_n)f(b_n) < 0$  then let  $a_n = x_n$ , else if  $f(a_n)f(x_n) < 0$ , then let  $b_n = x_n$ .

Step 4: Repeat steps 2 and 3 until  $f(x_n) = 0$  or  $|f(x_n)| \leq \beta$ , where  $\beta$  is the degree of accuracy.

#### 2.3.2. Convergence of Regula Falsi Method

Given an interval  $(a_n, b_n)$  such that the function  $f(x)$  in the equation  $f(x) = 0$  has its root, then one of the points  $a_n$  or  $b_n$  is always fixed and the other point varies with  $n$  [8]. In the case where  $a_n$  is fixed, then the function  $f(x)$  is approximated by a straight line which will pass through the points  $(a_n, f(x_n))$  and  $(x_n, f(x_n))$ , for  $n = 1, 2, \dots$

From the error equation

$$\epsilon_{n+1} = K\epsilon_n\epsilon_{n-1}$$

where

$$K = \frac{|f''(\delta)|}{2|f'(\delta)|}$$

And  $\epsilon_0 = \alpha_0 - \delta$  is independent of  $K$ . Hence we can write

$$\epsilon_{n+1} = K'\epsilon_n$$

Where  $K' = K\epsilon_0$  is the asymptotic error constant. Therefore, the Regula Falsi Method has a linear rate of convergence (Hassan, 2016).

#### 2.4. Background of Secant Method

One critical challenge Newton Raphson method is faced with, which serves as a de-motivation for its usage is the fact that the derivative of the function given must always be found before proceeding to find the root of the function. In some cases, there are functions where finding their derivatives is either extremely difficult (if not impossible) or time consuming [4]. The only way to avoid such problems is to approximate the derivative by knowing the values of the function at point and the previous approximation. Hence knowing  $f(x_n)$  and  $f(x_{n-1})$ , the derivative of  $f(x_n)$  which is  $f'(x_n)$  can be approximated as:

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \quad (9)$$

Substituting equation (8) above into the general Newton Raphson equation (6), we will get

$$x_{n+1} \approx x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \quad (10)$$

Therefore the Secant method can be expressed as:

$$x_{n+1} = x_n - h_n \quad (11)$$

where

$$h_n = \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

##### 2.4.1. Algorithm for Secant Method

Step 1: Write the equation in the form  $f(x) = 0$

Step 2: Test for values of  $x$  using the function  $f(x)$  to obtain the interval  $(x_n, x_{n-1})$  in which the real root lies.

Step 3: Compute  $f(x_n)$  and  $f(x_{n-1})$

Step 4: Substitute result of step 3 into

$$h_n = \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

and find  $h_n$

Step 5: Substitute result from step 4 into

$$x_{n+1} = x_n - h_n$$

and find the approximated value of  $x_{n+1}$

Step 5: Repeat steps 3 and 4 new values of  $x_{n+1}$  until  $x_{n+1} = x_n$ .

##### 2.4.2. Convergence of Secant Method

In the case of the Secant method, lets assume that  $\delta$  is the root of the function  $f(x) = 0$ . Substituting  $x_n = \delta + \epsilon_n$  into

equation (10) above, we obtain

$$\epsilon_{n+1} = \epsilon_n - \frac{(\epsilon_n - \epsilon_{n-1})f(\delta + \epsilon_n)}{f(\delta + \epsilon_n) - f(\delta + \epsilon_{n-1})}$$

Using Taylor's series to expand  $(\delta + \epsilon_n)$  and  $f(\delta + \epsilon_{n-1})$  about the point  $\delta$ , where  $f(\delta) = 0$  we obtain

$$\epsilon_{n+1} = \epsilon_n - \frac{(\epsilon_n - \epsilon_{n-1})[\epsilon_n f'(\delta) + \frac{1}{2}\epsilon_n^2 f''(\delta) + \dots]}{(\epsilon_n - \epsilon_{n-1})f'(\delta) + \frac{1}{2}(\epsilon_n^2 - \epsilon_{n-1}^2)f''(\delta) + \dots}$$

$$\epsilon_{n+1} = \epsilon_n - \left[ \epsilon_n + \frac{1}{2}\epsilon_n^2 \frac{f''(\delta)}{f'(\delta) + \dots} \left[ 1 + \frac{1}{2}(\epsilon_{n-1} + \epsilon_n) \frac{f''(\delta)}{f'(\delta)} + \dots \right] \right]^{-1}$$

and this can be written as

$$\epsilon_{n+1} = K\epsilon_n\epsilon_{n-1} \quad (12)$$

where the higher order of  $n$  is discarded and the error constant  $K = \frac{1}{2} \frac{f''(\delta)}{f'(\delta)}$ . With the definition of convergence in mind, we want a relation which is in the form:

$$\epsilon_{n+1} = D\epsilon_n^j \quad (13)$$

where  $D$  and  $j$  are to be computed. If  $\epsilon_{n+1}$  can be written as in equation (13), then we can equally write

$$\epsilon_n = D\epsilon_{n+1}^j \quad (14)$$

and

$$\epsilon_{n-1} = D^{-1/j} \epsilon_n^{1/j} \quad (15)$$

Substituting equations (14) and (15) into equation (12), we obtain

$$\epsilon_n^j = KD^{(-1+\frac{1}{j})} \epsilon_n^{(1+\frac{1}{j})} \quad (16)$$

Comparing the powers of  $n$  in both sides of equation (16), we get

$$j = 1 + \frac{1}{j}$$

which can be expressed as

$$j^2 - j - 1 = 0 \quad (17)$$

Solving for the root of  $j$  from equation (17) using the general quadratic formula

$$j = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where  $a = 1$ ,  $b = -1$  and  $c = -1$ , we get  $j = \frac{1}{2}(1 \pm \sqrt{5})$  discarding the negative sign, we obtain the rate of convergence for the Secant Method to be  $j = 1.618$ .

#### 2.5. Background of Fixed Point Iteration Method

The Fixed Point Iteration Method is another root - finding numerical method used to approximate solutions the equation

$f(x) = 0$ . To start with, we rewrite the above function in the form  $x = \varphi(x)$ , this way any solution of the function  $f(x) = 0$  with a fixed point  $\varphi$  can also be considered as a solution of  $x = \varphi(x)$ . Hence the general recursive iterative process for Fixed Point Iteration Method is given by

$$x_{n+1} = \varphi(x), n = 0, 1, 2, \dots \tag{18}$$

**2.5.1. Algorithm for Fixed Point Iteration Method**

To find the fixed point of  $\varphi$  in an interval  $[a, b]$ , given the equation  $x = \varphi(x)$  with an initial guess  $x_0 \in [a, b]$

- Step 1: Write the equation in the form  $f(x) = 0$
- Step 2: Initialize with guess  $x_0$  at  $n = 0$
- Step 3: Set  $x_{n+1} = \varphi(x_n)$
- Step 4: If  $|x_{n+1} - x_n| > \epsilon$ , set  $n = n + 1$  and go to step 3.
- Step 5: Stop when  $x_n = x_{n+1}$

**2.5.2. Convergence of Fixed Point Iteration Method**

Given that  $\varphi \in C[a, b]$  such that  $\varphi(x) \in [a, b]$ , for all  $x \in [a, b]$ . Suppose in addition that,  $\varphi'$  exists on  $(a, b)$  and that a constant  $0 < K < 1$  exist with

$$|\varphi'(x)| \leq K$$

for all  $x \in (a, b)$  Then, for any number  $x_0$  in  $[a, b]$  the

*Table 1. Root of Equation (19) Using Newton Raphson Method.*

$n$	$x_n$	$f(x_n)$	$f'(x_n)$	$h_n = -\frac{f(x_n)}{f'(x_n)}$	$x_{n+1} = x_n + h_n$
0	1.5	-0.25	4	0.00625	1.5625
1	1.5625	0.003906	4.125	-0.000947	1.56155
2	1.56155	-0.000012	4.1231	0.000003	1.56155

Table 1, shows output results of equation (19) when Newton Raphson method is used. The method converged to an exact solution of 1.56155 after the 2<sup>nd</sup> iteration.

*Table 2. Root of Equation (19) Using Bisection Method.*

$n$	$a_n(-ve)$	$b_n(+ve)$	$x_{n+1} = \frac{a_n+b_n}{2}$	$f(x_{n+1})$
0	1.5	1.6	1.55	-0.0475
1	1.55	1.6	1.575	0.055625
2	1.55	1.575	1.5625	0.003906
3	1.55	1.5625	1.5525	-0.037244
4	1.5525	1.5625	1.5525	-0.016694
5	1.5575	1.5625	1.56	-0.0064
6	1.56	1.5625	1.56125	-0.0001248
7	1.56125	1.5625	1.56188	-0.001349
8	1.56125	1.56188	1.56157	-0.000071
9	1.56125	1.56157	1.56141	-0.000589
10	1.56141	1.56157	1.56149	-0.000259
11	1.56149	1.56157	1.56153	-0.000094
12	1.56153	1.56157	1.56155	-0.000012
13	1.56155	1.56157	1.56156	-0.00003
14	1.56155	1.56156	1.56156	-0.00003

Table 2 indicates results obtained when the Bisection method is used to solve the nonlinear equation (19). It gave a final output exact root of 1.56156 but converged after the 14<sup>th</sup> iteration.

*Table 3. Root of Equation (19) Using Regula Falsi Method.*

$n$	$a_n(+ve)$	$b_n(-ve)$	$f(a_n)$	$f(b_n)$	$h_n$	$x_{n+1}$	$f(x_{n+1})$
0	1.5	1.6	-0.25	0.16	0.060976	1.56098	-0.002361
1	1.56098	1.6	-0.002361	0.16	0.000567	1.56155	-0.000012
2	1.56155	1.6	-0.000012	0.16	0.000003	1.56155	-0.000012

Table 3 shows results after Regula Falsi method was used to find the root of equation (19). It converged to the solution 1.56155 just as Newton Raphson method. The method converged after the 2<sup>nd</sup> iteration.

sequence defined by

$$x_n = \varphi(x_{n-1})$$

converges to the unique fixed point  $x$  in  $[a, b]$ .

**3. Results and Discussion**

The solution of the equation  $x^2 + x - 4 = 0$  using Bisection method, Newton Raphson method, Regula Falsi method, Secant method and Fixed Point Iteration method is computed using the preamble which starts with writing the given equation in the form  $f(x) = 0$  as shown below

$$f(x) = x^2 + x - 4 \tag{19}$$

This implies that

$$f'(x) = 2x + 1 \tag{20}$$

Testing for the possible roots of equation (19), we have  $f(1) = -2, f(1.5) = -0.25, f(1.6) = 0.16$ , implying that the real root of equation (19) lies between 1.5 and 1.6, which are adopted as initial guesses for the rest of the computations.

**Table 4.** Root of Equation (19) Using Secant Method.

$n$	$x_{n-1}$	$x_n$	$f(x_{n-1})$	$f(x_n)$	$h_n$	$x_{n-1} = x_n - h_n$	$f(x_{n-1})$
1	1.5	1.6	-0.25	0.16	0.022222	1.57778	0.06717
2	1.6	1.57778	0.16	0.06717	0.016078	1.4951	-0.269576
3	1.57778	1.4951	0.006717	-0.269576	-0.066188	1.56129	-0.001084
4	1.4951	1.56129	-0.269576	-0.001085	0.000267	1.56156	0.00003
5	1.56129	1.56156	-0.001084	0.00003	0.000007	1.56155	-0.000012
6	1.56156	1.56155	0.000003	-0.000012	-0.000003	1.56155	-0.000012

Table 4 is results obtained after Secant method was used to compute the root of equation (19). The method converged after the 6<sup>th</sup> iteration and gave a final output of 1.56155.

**Table 5.** Root of Equation (19) Using Fixed Point Iteration Method.

$n$	$x_n$	$x_{n+1} = \phi(x_n)$
0	1.5	1.58114
1	1.58114	1.55527
2	1.55527	1.56356
3	1.56356	1.56091
4	1.56091	1.56176
5	1.56176	1.56149
6	1.56149	1.56157
7	1.56157	1.56155
8	1.56155	1.56155

Table 5 in the other hand, shows results obtained using the Fixed Point Iteration method to solve the nonlinear equation (19). It converged to an exact root of 1.56155 after the 8<sup>th</sup>

iteration.

### 4. Conclusion

From the results obtained in the computations above, it was observed that using manual computation, Newton Raphson and Regula Falsi methods will converge faster to the solution of the function compared to the other methods. This was clearly seen in the number of iterations taken by each of the methods to converge to the solution. These findings contradicts the findings of some authors who placed the Secant method ahead of Newton Raphson method in terms of efficiency [6, 11]. Whilst on the other hand, some other authors findings agree with these papers [5, 8]. Findings and conclusions of other authors who have equally researched in this area, is summarized together with conclusion of this research as shown below

**Table 6.** Summary of Conclusions on Comparison of Methods by Authors.

Author	Conclusion (Most Efficient Method)	Tool Used
This paper	$NRM = RFM > SM > FPIM > BM$	Manual with IT – inspire instrument
(Moheuddin, et al., 2019)	$NRM > SM > RFM > BM$	MATLAB
(Ebelechukwu, 2018)	$NRM > RFM > BM$	Not stated
(Abdul - Hassan, 2016)	$SM > NRM > BM$	MATLAB 2009a
(Ahmed, 2015)	$NRM > SM > RFM$	MATLAB 7.8
(Ehiwario, et al., 2014)	$SM > NRM > BM$	Mathematica 9.0
(Srivastava, et al., 2011)	$SM > NRM > BM$	C language

### References

[1] Abdul-Hassan, N. Y. (2016). New Predictor-Corrector Iterative Methods with Twelfth-Order Convergence for Solving Nonlinear Equations. *American Journal of Applied Mathematics*, 4 (4), 175-180.

[2] Adegoke, T. M., Adegoke, G. K., Yahya, A. M., & Oduwole, H. K. (2018). *Comparative Study of Some Numerical Iterations Using Zero Truncated Poisson Distribution*. 1–30.

[3] Ahmad, A. G. (2015). Comparative Study of Bisection and Newton-Rhapson Methods of Root-Finding Problems. *International Journal of Mathematics Trends and Technology*, 19 (2).

[4] Biswa, N. D. (2012). *Lecture Notes on Numerical Solution of Root – Finding Problems MATH 435*.

[5] Ebelechukwu, O. C., Johnson, B. O., Michael, A. I., & Fidelis, A. T. (2018). Comparison of Some Iterative Methods of Solving Nonlinear Equations. *International Journal of Theoretical and Applied Mathematics*, 4 (2), 22.

[6] Ehiwario, J. C., & Aghamie, S. O. (2014). Comparative Study of Bisection, Newton-Raphson and Secant Methods of Root-Finding Problems. *IOSR Journal of Engineering (IOSRJEN)*, 4 (04).

[7] Gomes, A., & Morgado, J. (2013). A generalized regula falsi method for finding zeros and extrema of real functions. *Mathematical Problems in Engineering*, 2013.

[8] Kumar, R. (2015). *Comparative Analysis of Convergence of Various Numerical Methods*. 6 (June), 290–297.

[9] Moheuddin, M. M., Uddin, M. J., & Kowsher, M. A New STUDY TO FIND OUT THE BEST COMPUTATIONAL METHOD FOR SOLVING THE NONLINEAR EQUATION.

[10] Steven, C. C and Raymond, P. C (2015). *Numerical Methods for Engineers*.

[11] Srivastava, R. B., & Srivastava, S. (2011). Comparison of Numerical Rate of Convergence of Bisection, Newton-Raphson's and Secant Methods. *Journal of Chemical, Biological and Physical Sciences (JCBPS)*, 2 (1), 472.