

An Effective New Proposed Method (AWN) with Fourth-Order Convergence for Solving Nonlinear Equations

Hamam Al-Hasanat ¹ and Feras M. Al-Faqih ¹

¹ Mathematics Department Al Hussein Bin Talal University Ma'an, Jordan
E-mail: Homam01@yahoo.com (H.H), falfaqih@ahu.edu.jo (F.A)

Abstract

In this paper, we develop a new proposed iterative method (AWN) for root finding of nonlinear equations. The newly proposed method was subsequently proved and the fourth-order of convergence is obtained. Computationally, Based on MATLAB simulations, We demonstrated the efficiency of the AWN method compared to many recent methods. We conclude that the AWN method will help to get a good performance in applied science compared with other methods.

Keywords: Nonlinear equation, Order of convergence, Newton-Raphson method, Secant Method, Reguli Falsi method, AWN method.

1 Introduction

In Mathematics, Economics, Science, and engineering the problem of finding an approximation root of an equation can be traced back at least as far as 1700 BC (as Babylonian mathematics)[11].

One-dimensional root-finding is the most difficult numerical approximation problem. Finding the solution of the function for $f(x) = 0$ such that $f : R \rightarrow R$ is called root or zero finding.

There are many attempts to find a method that can solve the non-linear equation, an Equations system that cannot be solved analytically due to the limitations of analytical methods has to use iteration methods, one of the

most popular iterative methods of finding roots was Newton's method which was published in 1711 by William Jones[2]. Under this topic, many methods were investigated and modified.

In a recent study, Hajer et al.,(2022) [1] used a new algorithm for solving transcendental equations, and they compared their method with other methods, and their convergence rates achieved greater than two. Moreover, methods for root finding of non-linear equations have been developed and their efficiency is improved and became more accurate than before.

Often non-linear methods do not have a closed form or can't be calculated analytically to find the root, in such cases, numerical methods are used to approximate the root. Most numerical methods use iteration to find approximation roots and look for the best convergence to the exact roots.

In iteration methods, there are two basic groups: The first is bracketing, which begins with a bounded interval that guarantees to bracket a root, and in each iteration size of the initial bracket is reduced step by step until it converges to the root with the required accuracy. the Bisection method and Reguli Falsi method are two examples that follow the bracketing method.

The second group is called the open method, it begins with an initial guess of the root, this guess point takes one or two starting values that do not have to bracket a root, There are many iteration methods that follow the open method like Newton's method [14], Householder's method [12] and Exponential Method[13].

Open methods may diverge during the computation process, but if it is converged, it usually does so much faster than bracketing.[3] The main criterion to use such methods is the rate of convergence, which represents how fast the method approaches its limit.[4]

In this paper, we will present a new proposed iterative method(AWN), its definition, theorem, and numerical examples, at the end, there is a comparison between the AWN method and some recent iterative methods, and this comparison will show the efficiency of AWN method numerically and graphically.

2 The New Proposed Iterative Method (AWN)

In this section, we will present a new iterative formula (1) for obtaining the root of a non-linear function $f(x) = 0$.

$$x_{n+1} = \left[x_n - \frac{f(x_n)}{f''(x_n)} \right] - \frac{f(x_n)}{f'(x_n)} \quad \text{where } n = 0, 1, 2, \dots, \quad (1)$$

This formula is satisfied if $f(x)$ is twice differentiable such that $f'(x)$ and $f''(x) \neq 0$.

This algorithm will be investigated in Theorem 2.1.

Definition 2.1. [?] " Let $\{x_n\}_{n=0}^{\infty}$, be a sequence in R^n that converges to $r \in R^n$ and assume that $x_n \neq r$ for each n . We say that the rate of convergence of x_n to r is of order p , with asymptotic error constant C , if

$$\lim_{x \rightarrow +\infty} \frac{\|x_{n+1} - x_n\|}{\|x_n - r\|^p} = C, \quad \text{Where } p > 1 \quad \text{and} \quad C > 0.$$

Theorem 2.1. Let U be an open interval of a real number and $f : U \rightarrow R$ be a differentiable nonlinear function and $f(x) = 0$ with a simple root $r \in U$. such that $f'(x_i), f''(x_i) \neq 0$ where $i=0,1,2,\dots,n$. If x_0 close to r , then formula (1) has order of convergence less than or equal 4. ($p \leq 4$).

Proof : Let r is the root of $f(x) = 0$, and x_n be an estimated root of r , Let the error is E , such that $E_n = r - x_n$ then by subtracting r from both sides of the equation (1), we get :

$$r - x_{n+1} = r - \left[\left(x_n - \frac{f(x_n)}{f''(x_n)} \right) - \frac{f(x_n)}{f'(x_n)} \right]$$

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

Since $E_n = r - x_n$, then $E_{n+1} = r - x_{n+1}$

$$E_{n+1} = E_n + \frac{f(x_n)}{f''(x_n)} + \frac{f(x_n)}{f'(x_n)} \quad (2)$$

Now $f(r) = 0$, Because r is the root of $f(x)$, Using Taylor expansion from (2) we have:

$$f(x_n) = f(r - E_n) = f(r) - E_n(f'(r)) + (E_n)^2 \frac{f''(r)}{2!} + \dots = -E_n \frac{f'(r)}{1} + (E_n)^2 \frac{f''(r)}{2!} + \dots$$

$$f'(x_n) = f'(r - E_n) = f'(r) - E_n(f''(r)) + (E_n)^2 \frac{f^{(3)}(r)}{2!} + \dots = f'(r) - E_n \frac{f''(r)}{1} + (E_n)^2 \frac{f^{(3)}(r)}{2!} + \dots$$

$$f''(x_n) = f''(r - E_n) = f''(r) - E_n \frac{f^{(3)}(r)}{1!} + (E_n)^2 \frac{f^{(4)}(r)}{2!} + \dots = f''(r) - E_n \frac{f^{(3)}(r)}{1} + (E_n)^2 \frac{f^{(4)}(r)}{2!}$$

The idea for E_n in $f(r - E_n)$ it's a very small value and the Taylor series around E_n equal to zero.

Now, by substitution the above terms in equation (2), we get:

$$E_{(n+1)} = E_n + \frac{-E_n \frac{f'(r)}{1} + (E_n)^2 \frac{f''(r)}{2!} + \dots}{f''(r) - E_n \frac{f^{(3)}(r)}{1} + (E_n)^2 \frac{f^{(4)}(r)}{2!} + \dots} + \frac{-E_n + (E_n)^2 \frac{f''(r)}{2!} + \dots}{f'(r) - E_n \frac{f''(r)}{1} + (E_n)^2 \frac{f'''(r)}{2!} + \dots} \quad (3)$$

Simplify the 2^{ed} term by dividing second term by $f''(r)$, also the same for 3rd term we divide by $f'(r)$ from equation (3), we get:

$$E_{(n+1)} = E_n + \frac{-E_n \frac{f'(r)}{f''(r)} + \frac{(E_n)^2}{2!} + \dots}{1 - E_n \frac{f^{(3)}(r)}{2!f''(r)} + (E_n)^2 \frac{f^{(4)}(r)}{2!f''(r)} + \dots} + \frac{-E_n + \frac{(E_n)^2}{2!} \frac{f''(r)}{f'(r)} + \dots}{1 - E_n \frac{f''(r)}{f'(r)} + (E_n)^2 \frac{f'''(r)}{2!f'(r)} + \dots} \quad (4)$$

Now, by applying Taylor series expansion for the denominator of both terms in equation (4), we get:

$$[E_{(n+1)} = E_n + (-E_n \frac{f'(r)}{f''(r)} + \frac{(E_n)^2}{2!} + \dots)(1 + E_n \frac{f^{(3)}(r)}{2!f''(r)} + (E_n)^2 \frac{f^{(4)}(r)}{2!f''(r)} + \dots) + (-E_n + \frac{(E_n)^2}{2!} \frac{f''(r)}{f'(r)} + \dots)(1 + E_n \frac{f''(r)}{f'(r)} + (E_n)^2 \frac{f^{(3)}(r)}{2!f'(r)} + \dots)]$$

By multiplying both brackets in both terms, we get:

$$E_{n+1} = E_n - E_n \frac{f'(r)}{f''(r)} - (E_n)^2 \frac{f'(r)}{f''(r)} \frac{f^{(3)}(r)}{2!f''(r)} + \frac{(E_n)^2}{2!} + (E_n)^3 \left[\frac{f^{(3)}(r)}{4f''(r)} - \frac{f'(r)}{f''(r)} \right] + (E_n)^4 \frac{f^{(4)}(r)}{4f''(r)} + \dots$$

$$\dots - E_n + (E_n)^2 \frac{f''(r)}{f'(r)} + \frac{(E_n)^2}{2} \frac{f''(r)}{f'(r)} + \frac{(E_n)^3}{2} \left[\frac{f''(r)}{f'(r)} \right]^2 - E_n^3 \frac{f^{(3)}(r)}{2f'(r)} + \frac{E_n^4}{4} \frac{f''(r)}{f'(r)} \frac{f^{(3)}(r)}{f'(r)} + \dots$$

Now, By Factoring out $\frac{f'(r)}{f''(r)}$, $\frac{f''(r)}{f'(r)}$ and $(E_n)^3$ as common factors, we get :

$$E_{n+1} = E_n - \frac{f'(r)}{f''(r)} \left[E_n + E_n^2 \frac{f^{(3)}(r)}{2!f''(r)} \right] + \frac{E_n^2}{2} + (E_n)^3 \left[\frac{f^{(3)}(r)}{4f''(r)} - \frac{f'(r)}{f''(r)} \right] + (E_n)^4 \frac{f^{(4)}(r)}{4f''(r)} + \dots$$

$$- E_n + \frac{f''(r)}{f'(r)} \left[-E_n^2 + \frac{(E_n)^2}{2} - \frac{(E_n)^3}{2} \frac{f''(r)}{f'(r)} + (E_n)^4 \frac{f^{(3)}(r)}{4f'(r)} \right] + \dots$$

By simplify E_n , we get:

$$E_{n+1} = -\frac{f'(r)}{f''(r)} \left[E_n + (E_n)^2 \frac{f^3(r)}{2f''(r)} \right] + \frac{(E_n)^2}{2} + (E_n)^3 \left[\frac{f^3(r)}{4f''(r)} - \frac{f'(r)}{f''(r)} \right] + \dots$$

$$+ \frac{f''(r)}{f'(r)} \left[-\frac{(E_n)^2}{2} - \frac{(E_n)^3}{2} \frac{f''(r)}{f'(r)} + \frac{(E_n)^4}{4} \frac{f^3(r)}{4f'(r)} \right] + \dots$$

Let $K = \frac{f'(x)}{f''(x)}$, $M = \frac{f^3(r)}{2f''(r)}$, $N = \frac{f''(r)}{f'(r)}$, $L = \frac{f^3(r)}{f'(r)}$, then

$$E_{n+1} = -K \left[E_n + M(E_n)^2 \right] + \frac{(E_n)^2}{2} + (E_n)^3 \left[\frac{M}{2} - k \right] + \dots + N \left[-\frac{(E_n)^2}{2} - \frac{(E_n)^3}{2} N + L \frac{(E_n)^4}{4} \right]$$

Now, by taking the absolute value for both sides and apply the definition(2.1), we get :

$$|E_{n+1}| \leq |K [E_n + M(E_n)^2] + \frac{(E_n)^2}{2} + (E_n)^3 \left[\frac{M}{2} - k \right] + \dots + N \left[-\frac{(E_n)^2}{2} - \frac{(E_n)^3}{2} N + L \frac{(E_n)^4}{4} \right]|$$

E_{n+1} converge to E_n^4 .

So, AWN method has an order of convergence less than or equal 4, ($p \leq 4$).

3 Implementation the New Proposed Method (AWN)

Now we employ the proposed method (AWN) to solve nonlinear equations using MATLAB, also we will show the algorithm steps and the flow chart of the proposed algorithm(AWN).

3.1 Algorithm

1. Select an initial guess point x_0 or initial interval $x_0 \neq 0$. Enter the function $f(x)$, $f'(x_0)$ and $f''(x_0) \neq 0$.
2. Apply AWN algorithm which is given in the equation (1).
3. Follow Step 2 until the approximate root is obtained. If $|x_{(n+1)} - x_n| < \text{tolerance}$, Print $x_{(n+1)}$. If $f(x_n) = 0$, then Stop. , for $n = 0, 1, 2, \dots$

3.2 Flow Chart of AWN Method

The next figure is a flow chart of this AWN algorithm.

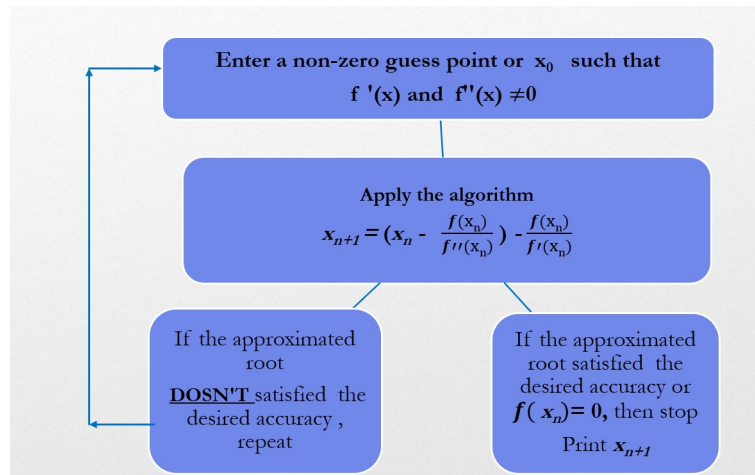


Figure 1: Flow chart of the proposed algorithm(AWN)

3.3 MATLAB Code of AWN Method

Here we show the code of the new proposed algorithm in MATLAB, where f is given a nonlinear equation, x is the initial guess of the root, i is the number of iterations needed and e is the relative error (let it equal 10^{-9}).

```

clc;clear;
syms x;
i=0;
f=input('f(x)=');
df=diff(f,x);
dff=diff(df,x);
f=inline(f);
df=inline(df);
dff=inline(dff);
format longG
disp('f(x)=');
disp(f);
x=input('x0=');
e=10^(-9);
disp(["  Root    ", ' Iteration# ']);
yn=x-f(x)/dff(x);
xn=yn-f(yn)/df(x);
tol=abs(xn-x);
while tol >e
    i=i+1;
    x=xn;
    yn=x-f(x)/dff(x);
    xn=yn-f(yn)/df(x);
    tol=abs(xn-x);
    if xn==x ; break ;
end
disp( [xn  i]);
end
disp('the root is ='); disp(xn);
disp('Iterations no =');disp(i);

```

4 Computational Examples

In this section, We will apply the newly proposed method (AWN) to several nonlinear equations then we will compare AWN method with various existing methods such as Newton Raphson method (NRM)[6], Regula Falsi(RFM)[10, 5], Householder's(HHM)[12, 9], Secant Method(SECM)[7], and the modified newton algorithm (FLM) as published paper[8].

The comparison will involve examining some examples of nonlinear equations as given below, Also it will display the approximate zero (the root x) found up to the 14th decimal place, and the tolerance error (1×10^{-9})for simplicity.

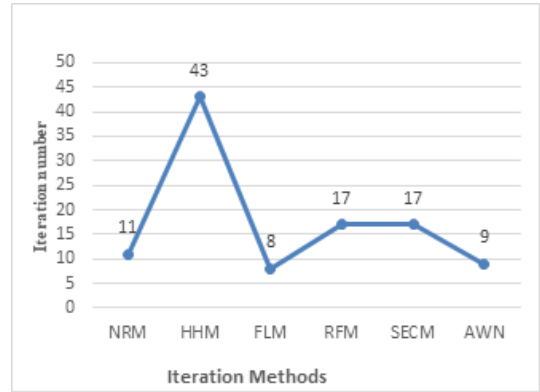
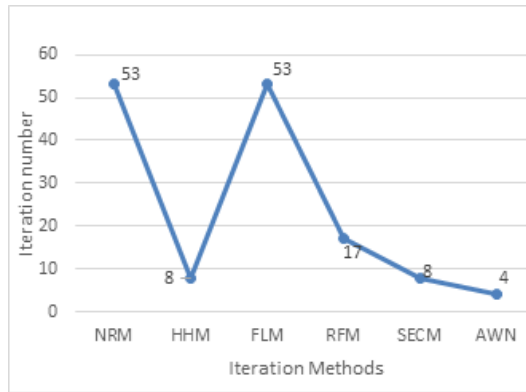
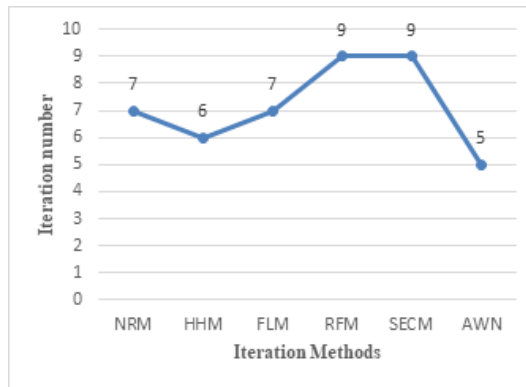
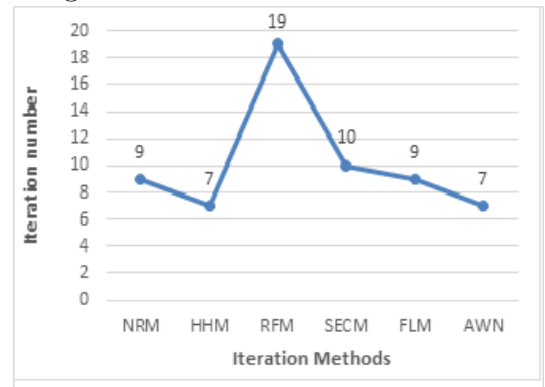
1. $f_1(x) = x^3 + 4x^2 - 10$, with initial approximation $x_0 = -0.3$.
2. $f_2(x) = (x - 2)^3 - 0.8$, with initial approximation $x_0 = 0.1$.
3. $f_3(x) = xe^{-x} - 0.1$, with initial approximation $x_0 = 1.2$.
4. $f_4(x) = x^3 - e^x$, with initial approximation $x_0 = 6$.
5. $f_5(x) = \cos(x) - x$, with initial approximation $x_0 = 0.5$.
6. $f_6(x) = \sin^2(x) - x^2 + 1$, with initial approximation $x_0 = 0.5$.
7. $f_7(x) = xe^x - \sin(x) - 5$, with initial approximation $x_0 = 1$.

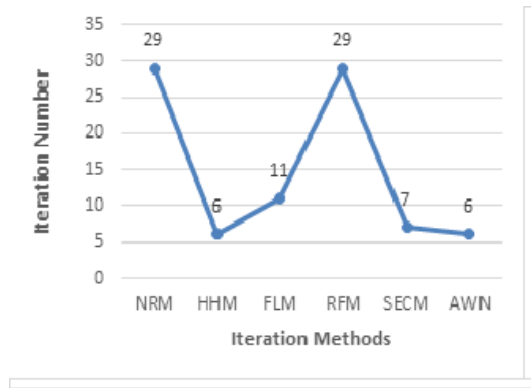
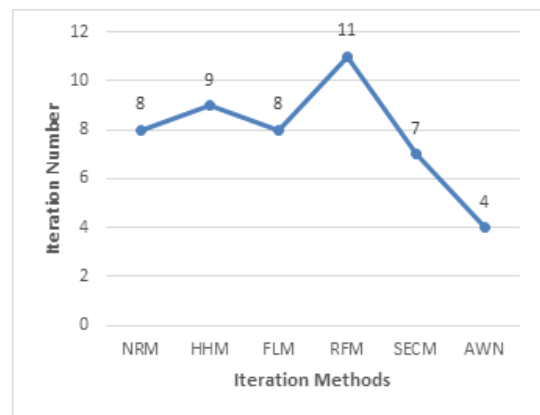
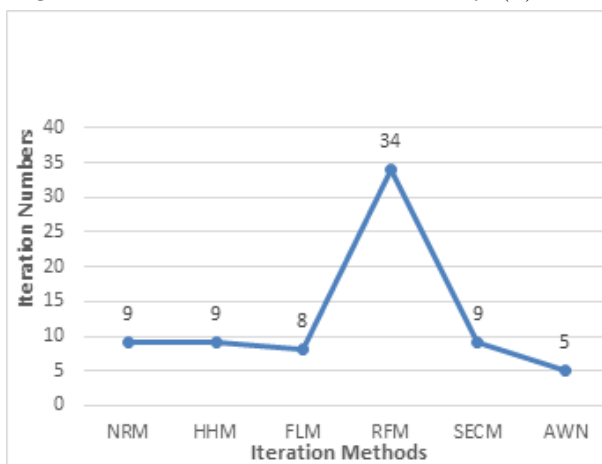
	IT	x_n	$f(x_n)$
$f_1, x_0 = -0.3, [0, 2]$			
NRM	53	1.03652300134141	0
HHM	8	1.03652300134141	0
FLM	53	1.03652300134141	0
RFM	17	1.036523001337214	0
SECM	8	1.03652300134141	0
AWN	4	1.03652300134141	0
$f_2, x_0 = 0.1, [-1, 2]$			
NRM	11	2.92831776672256	4.44089209850063e-161
HHM	43	2.92831776672256	4.44089209850063e-16
FLM	8	2.92831776672256	3.10862446895044e-15
RFM	17	2.92831776672256	0
SECM	17	2.92831776672256	4.44089209850063e-16
AWN	9	2.92831776672256	4.44089209850063e-16
$f_3, x_0 = 1.2, [0, 1]$			
NRM	7	0.111832559158963	0
HHM	6	0.111832559158963	0
FLM	7	0.111832559158963	0
RFM	9	0.111832559158963	0
SECM	9	0.111832559158963	0
AWN	5	0.111832559158963	0
$f_4, x_0 = 6, [-5, 5]$			
NRM	9	0.77288295914921	0
HHM	7	0.77288295914921	0
FLM	9	0.77288295914921	0
RFM	19	0.77288295914921	0
SECM	10	0.77288295914921	0
AWN	7	0.77288295914921	0
$f_5, x_0 = 4, [-9, 9]$			
NRM	29	0.739085133215155	1.02071205564612e-28
HHM	6	0.739085133215155	7.49910898025724e-29
FLM	29	0.739085133215155	0
RFM	11	0.739085133215155	4.44089209850063e-16
SECM	7	0.739085178846408	7.7715611723761e-15
AWN	6	0.739085133119089	0
$f_6, x_0 = 0.5, [0, 2]$			
NRM	8	1.40449164821534	3.33066907387547e-16
HHM	9	1.40449164821534	-4.44089209850063e-16
FLM	8	1.40449164821534	-1.77635683940025e-15
RFM	11	0.739085133215155	4.44089209850063e-16
SECM	7	0.739085178846408	7.7715611723761e-15
AWN	4	1.40449164821534	-4.44089209850063e-16

$f_7, x_0 = 1, [-2, 2]$			
NRM	9	1.43145283298218	0
HHM	9	1.43145283298218	0
FLM	8	1.40449164821534	-1.77635683940025e-15
RFM	34	1.43145283298218	0
SECM	9	1.43145283298218	0
AWN	5	1.43145283298218	0

Table 4.1: Comparison of iterations by different methods

As shown in table(4.1), illustrates that AWN requires the fewest iterations to approach the approximation root within a given error compared to other iteration methods. Additionally, x_n is close to zero when compared to other methods within the given error.

Figure1 :Iterations at different methods of $f_1(x)$ Figure2:Iterations at different methods of $f_2(x)$ (Figure3:Iterations at different methods of $f_3(x)$)Figure4:Iterations at different methods of $f_4(x)$

(Figure5:Iterations at different methods of $f_5(x)$)Figure6: Iterations at different method of $f_6(x)$ Figure7:Iterations at different methods of $f_7(x)$

The proposed method (AWN) requires fewer iterations than other methods, as shown in Figures 1-7.

5 Conclusion

This paper presented a newly proposed method (AWN) with fourth-order convergence for root-finding problems of nonlinear equations, AWN method is illustrated and validated with numerical examples using MATLAB. The computational results showed that the AWN method required less number of iterations compared with other methods.

Regarding the foregoing, we conclude that the performance and convergence rate of the AWN method is better than other methods, As a result, it is considered more efficient and performs better than others.

6 Open Problem

This paper outlines a new method to find the real root of a nonlinear equation. For this section, we suggest updating, upgrading, and extending this method to approximate the root of the system of nonlinear equations.

REFERENCES

- [1] Hajar Alzawaideh, Feras M. Al-Faqih, and Sahem Tarawneh. A new algorithm to compute the real root of transcendental equation. *International Journal of Scientific and Engineering Research*, 15(3):ISSN: 1998–6262, September 2022.
- [2] Hald Anders. *A history of probability and statistics and their applications before 1750*, volume 574. John Wiley & Sons, 2005.
- [3] Popovski D. B. A note on king’s method f for finding a bracketed root. *Computing*, 29(4):355–359, 1982.
- [4] Richard L Burden, J Douglas Faires, and Annette M Burden. *Numerical analysis*. Cengage learning, 2015.
- [5] J. Chen and W. Li. An improved exponential regula falsi methods with quadratic convergence of both diameter and point for solving nonlinear equations. *Applied Mathematics and Computation*, 57:80–88, 2007.
- [6] JC Ehiwario and SO Aghamie. Comparative study of bisection, newton-raphson and secant methods of root-finding problems. *IOSR Journal of Engineering*, 4(4):01–07, 2014.
- [7] King R. F. Methods without secant steps for finding a bracketed root. *Computing*, 17(1):49–57, 1976.
- [8] Liang Fang, Li Sun, and Guoping He. An efficient newton-type method with fifth-order convergence for solving nonlinear equations. *Computational & Applied Mathematics*, 27:269–274, 2008.
- [9] Alston Scott Householder. *The numerical treatment of a single nonlinear equation*. McGraw-Hill, 1970.
- [10] SA Ibrahim. An improved exponential regula falsi methods with cubic convergence for solving nonlinear equations. *Journal of applied mathematics & informatics*, 28(5-6):1467–1476, 2010.
- [11] Daniel F. Mansfield and Norman Wildberger. Plimpton 322 is babylonian exact sexagesimal trigonometry. *Historia Mathematica*, 44:395–419, 2017.
- [12] Srinivasarao Thota. A numerical algorithm to find a root of non-linear equations using householder’s method. *Int J Adv Appl Sci*, 10(2):141–148, 2021.

- [13] Srinivasarao Thota, Tekle Gemechu, and P Shanmugasundaram. New algorithms for computing a root of non-linear equations using exponential series. *Palestine Journal of Mathematics*, 10(1), 2021.
- [14] Sunethra Weerakoon and TGI Fernando. A variant of newton's method with accelerated third-order convergence. *Applied mathematics letters*, 13(8):87–93, 2000.