Int. J. Open Problems Compt. Math., Vol. 14, No. 2, June 2021
Print ISSN: 1998-6262, Online ISSN: 2079-0376

An Algorithm to Compute Real Root of Transcendental Equations Using Hyperbolic Tangent Function

Srinivasarao Thota[1] and Vivek Kumar Srivastav[2]

[1]Department of Applied Mathematics, School of Applied Natural Sciences,
Adama Science and Technology University, Post Box No. 1888, Adama, Ethiopia
[2]Department of Mathematics and Computing,
Motihari College of Engineering, Motihari, India
e-mail: [1]srinithota@ymail.com and [2]vivekapril@gmail.com

Received 2 September 2020; Accepted 12 March 2021

(Communicated by Feras M. Al-Faqih)

Abstract

This paper presents a new algorithm to find a non-zero real root of the transcendental equations using hyperbolic tangent function. Indeed, the new proposed algorithm is based on the combination of hyperbolic tangent series and Newton Raphson method, which produces better approximate root than Newton Raphson method. The implementation of the proposed algorithm is programmed in MATLAB and Maple. Certain numerical examples are presented to validate the efficiency of the proposed algorithm. This algorithm will help to implement in the commercial package for finding a real root of a given transcendental equation.

Keywords: *Algebraic equations, Transcendental equations, Hyperbolic Tangent, Newton Raphson method.*

2010 Mathematics Subject Classification: 65Hxx, 65H04.

1 Introduction

The root finding algorithms in science, engineering and computing are playing important role to compute roots of transcendental functions. A root of a function $f(x)$ is a number 'α' such that $f(\alpha) = 0$. Generally, the roots of transcendental functions cannot be expressed in closed form or cannot be computed exactly. The root-finding algorithms give us approximations to the roots, these approximations are expressed either as small isolating intervals or as floating point numbers. The solution of the equation $t(x) = s(x)$ is nothing but to find the roots of the function $f(x) = t(x) - s(x)$. It is seen that several root finding algorithms are available in literature. Most of the algorithms use iteration, producing a sequence of numbers that hopefully converge towards the root as a limit. They require one or more initial guesses of the root as starting values, then each new iteration of the algorithm produces a successively more accurate approximate root in comparison of previous iteration. The purpose of existing algorithms is to provide higher order convergence with guaranteed root. Many existing algorithms do not guarantee that they will find all the roots; in particular, if such an algorithm does not find any root, that does not mean that no root exists. There are many well known root finding algorithms available, (for example, *Bisection*, *Secant*, *Regula-Falsi*, *Newton-Raphson*, *Muller's* methods etc.) to find an approximate root of algebraic or transcendental equations. If the equation $f(x) = 0$ is an algebraic equation, then there are many algebraic formulae available to find the roots. However, if $f(x)$ is a polynomial of higher degree or an expression involving transcendental equations such as trigonometric, exponential, algorithmic etc., then there are no algebraic methods exist to express the root.

1.1 Literature Review

It is known that the Bisection method is used to find a root of the transcendental equations. It is based on the repeated application of intermediate value property [8, 21]. This method begins with two initial approximations with opposite signs of the corresponding functions. Since the *order of convergence* of this method is one, the error decreases linearly with each step by a factor of 0.5, hence the convergence is slow. To obtain desired accuracy, a large number of iterations are required, however it gives a guaranteed convergence to a root. Eiger (1984) used Bisection method for solving the system of non-linear equations. Eiger's programme was based on topological degree of mapping and a simplex-bisection scheme [5]. After that Vrahatis and Iordanidis(1986) proposed rapid generalization method of Bisection for solving system of linear equations. In the proposed method, system of linear equation are based on the non-zero value of topological degree [15]. Bisection method is not only useful for finding the roots but also several applications such as to find the maxima or limit of continuous function in a closed interval [6]. In this references, Wood (1992) has given useful generalization of Bisection method to higher dimensions [6]. Novak et al.(1995) have given hybrid Secant-Bisection method to find a root. They have stressed and analyzed adaptive stopping rule which is exponentially more powerful than non-adaptive stopping rule [4]. An improved version of Bisection method and Müllers with global and asymptotic convergence of non-linear equations was given by Xinyuan Wu (2005). This proposed algorithm is more effective than other previous methods. The modification in Muller method by incorporating Bisection property, they obtained better converges order as compared to previous [34]. Yakoubsohn (2005) used Bisection-Exclusion method for finding the zeros of univariate analytic functions [7]. The complexity of the bisection method was discussed by Gutierrez et al.(2007). They have found subexponential asymptotic upper bound for the number of triangles created on mesh obtained by iterative bisection method [1]. Bachrathy and Stepan (2012) proposed multidimensional bisection method for solving the system of non-linear equations. This method is able to find the roots where number of unknowns are larger than or equals to the number of equations [2]. Thus there are several such applications of Bisection method. Because of the slower convergence of Bisection method, researchers were interested to investigate new algorithms with guaranteed root.

Further in the literature, this method has been implemented by the researchers to obtain most approximate root having higher convergence [35]. In this context, Regula-Falsi method adopted the concept of straight line and made to speed up bisection method retaining its guaranteed convergence with order of convergence higher than Bisection method. In 1971, Dowell and Jarratt investigated the revised form of Regula-Falsi method just by replacement of *new-function* to *half of the some intermediate function*. They obtained better convergence than previous Regula-Falsi Method [3]. Later on, Saeid and LIAO (2008) studied a new modifications in false position method based on homotopy analysis method (HAM) [28]. Wu and Wu (2000) [36] discussed quadratic convergent algorithm without using the derivative function. Steffensen's method was used to obtain the final algorithm. Later on, Wu et al. (2003) [35] have given a new algorithm by employing Steffensen's method of accelerating convergence after using standard Regula-Falsi method(RFM). In their proposed algorithm, they developed a new root finding with global convergence of non-linear equation. Zhu and Wu (2003) [37] discussed derivative free method of third order. Further, researchers were interested to find the root using a single approximation. Therefore, Newton-Raphson method was proposed in this direction.

Newton-Raphson method is generally used to improve the root obtained by one of the above methods. This method used the concept of *tangent* at the initial approximation point. The next approximate root takes those values where the tangents intersect the x-axis. So this method fails where tangent becomes parallel to x-axis. Since the Newton-Raphson method converges second order accurately, therefore it converges very rapidly than other methods (Bisection, Regula-falsi, etc.). However, it may not always gives guaranteed root. Johan and Ronald (1994) discussed different sufficient conditions for the convergence of Newton-Raphson Method. They have mentioned some possible conditions for convergence of Newton-Raphson Method by converting the first derivative into some other functions [9]. In another study, Kanwar et al. (2005) given a new class of iterative technique with quadratic convergence and can be applied as the alternative of Newton-Raphson method. In their proposed algorithm a new parameter has been inserted in such away that the corresponding function and its derivative has same sign. When this new parameter equals to zero then it automatically converted into Newton-Raphson method [16]. Kanwar et al. (2005) have studied third-order iterative methods for solving non-linear equations. Using the above concept, they have given third-order multi-point methods without using second derivative [17]. Sharma and Goyal (2006) [10] presented fourth order derivative-free methods for solving non-linear equations. This algorithm was developed using Steffensen's method, which used the ideas of forward and backward difference approaches. It has been found that all the methods are classified into two category namely one-step method and two step methods [18].

Noor and Ahmad (2006) [19] presented predictor-corrector method type iterative method by using standard Regula Falsi Method (RFM). The new developed algorithm was also compared with the previous methods and found better. Later on, Noor et al.(2006) suggested and analyzed several two-step method for solving non-linear equations. After that, they have given three-step iterative methods having third order convergence [20]. Chen and Li (2006, 2007) have improved the RFM and named as improved exponential Regula-Falsi method for solving non-linear equations. In their modified method, they used an exponential iterative methods accelerating after employing classical RFM. This new modified method has asymptotic quadratic convergence [11, 12]. Jinhai Chen (2007) [13] has given quadratic convergence algorithm in which they employed two new iterative methods after using standard RFM to accelerate the rate of convergence.

Order of Newton-Raphson method for multiple roots was also discussed in the paper [9]. Sagraloff and Mehlhorn (2013) presented *Descartes method* for computing the root of the polynomial equations. By using this method, it is found good approximations of the polynomial coefficients. Their algorithm may also be applicable to implement the isolating intervals to an arbitrary small size [29]. John Abbott (2014) presented quadratic interval refinement for real roots. The new method was based on classical Bisection algorithm and Newton-Raphson Method. This new algorithm does not required to evaluate the derivatives [14]. T. Gemechu (2017) [31] derived several methods based on Taylor's expansion. There are some other algorithms available, see for example, [22–27, 30, 32, 33].

In this work, the proposed new algorithm is based on hyperbolic tangent function and Newton-Raphson methods, which provides faster roots in comparison of the previous methods. The new proposed algorithm will be useful for computing a real root of transcendental equations. The rest of the paper is as follows: Section 2 describes the proposed method, their mathematical formulation, calculation steps and flow chart; implementation of the proposed algorithm in Matlab and Maple is presented in Section 3 with sample computations; and Section 4 discuss some numerical examples to illustrate the algorithm and comparisons are made to show efficiency of the new algorithm.

2 Main Results

2.1 A Hyperbolic Tangent Algorithm

The new hyperbolic tangent iterative formula using tanh is proposed as

$$x_{n+1} = x_n \left[1 + \tanh\left(\frac{-f(x_n)}{x_n f'(x_n)}\right)\right], \qquad n = 0, 1, 2, \ldots. \tag{1}$$

By expanding this iterative formula, one can obtain the standard Newton-Raphson method as in first two terms. This is shown in the following theorem.

Theorem 2.1. *Suppose $\alpha \neq 0$ is a real exact root of $f(x)$ and θ is a sufficiently small neighbourhood of α. Let $f''(x)$ exists and $f'(x) \neq 0$ in θ. Then the iterative formula given in equation (1) produces a sequence of iterations $\{x_n : n = 0, 1, 2, \ldots\}$ with order of convergence $p \geq 2$.*

Proof. The iterative formula given in equation (1) can be expressed in the following form

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

Since

$$\lim_{x_n \to \alpha} \tanh\left(\frac{-f(x_n)}{x_n f'(x_n)}\right) = 0,$$

and hence $x_{n+1} = \alpha$.

Using the standard expansion of $\tanh(x)$ as

$$\tanh(x) = x - \frac{1}{3}x^3 + \frac{2}{15}x^5 - \frac{17}{315}x^7 + \frac{62}{2835}x^9 - \cdots \tag{2}$$

and from equations (1) and (2), we have

$$
\begin{aligned}
x_{n+1} &= x_n\left[1+\tanh\left(\frac{-f(x_n)}{x_n f'(x_n)}\right)\right]\\
&= x_n\left[1+\left(\frac{-f(x_n)}{x_n f'(x_n)}\right)-\frac{1}{3}\left(\frac{-f(x_n)}{x_n f'(x_n)}\right)^3+\frac{2}{15}\left(\frac{-f(x_n)}{x_n f'(x_n)}\right)^5-\frac{17}{315}\left(\frac{-f(x_n)}{x_n f'(x_n)}\right)^7+\cdots\right]\\
&= x_n-\frac{f(x_n)}{f'(x_n)}+\frac{1}{3x_n^2}\left(\frac{f(x_n)}{f'(x_n)}\right)^3-\frac{2}{15x_n^4}\left(\frac{f(x_n)}{f'(x_n)}\right)^5+o\left(\frac{17}{315x_n^6}\left(\frac{f(x_n)}{f'(x_n)}\right)^7\right).
\end{aligned}
$$

Since $f(x_n)\approx 0$, when we neglect higher order terms, then above equation becomes Newton-Rapson method. Indeed, we have the following formulae obtained from first two terms, three terms and four terms of the expansion respectively.

$$x_{n+1} = x_n-\frac{f(x_n)}{f'(x_n)}. \tag{3}$$

$$x_{n+1} = x_n-\frac{f(x_n)}{f'(x_n)}+\frac{1}{3x_n^2}\left(\frac{f(x_n)}{f'(x_n)}\right)^3. \tag{4}$$

$$x_{n+1} = x_n-\frac{f(x_n)}{f'(x_n)}+\frac{1}{3x_n^2}\left(\frac{f(x_n)}{f'(x_n)}\right)^3-\frac{2}{15x_n^4}\left(\frac{f(x_n)}{f'(x_n)}\right)^5. \tag{5}$$

In the above equations, we obtained Newton-Rapson method having quadratic convergence in first two terms. Therefore, the order of convergence of proposed algorithm is at least $p \geq 2$. □

2.2 Steps for Computing Root

I Select an approximation $x_n \neq 0$.

II Apply the iterative formula given in equation (1).

III Repeat Step II until we get desired approximate root, for $n = 0, 1, 2, \ldots$.

2.3 Flow Chat

Flow chat of the proposed algorithm is presented in Figure 1

3 Implementation of Proposed Algorithm

In this section, we provide the implementation of the proposed method in MATLAB and Maple.

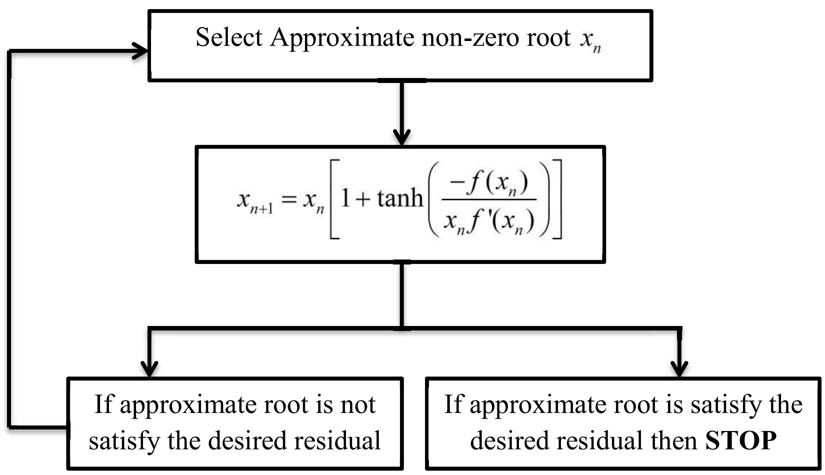

Figure 1: Flow chart for proposed algorithm

3.1 Implementation in MATLAB

The following a data type `TanhNewton(f,x0,esp,n)` gives the implementation in MATLAB, where `f` is given non-linear transcendental function, `x0` is the initial approximation of the root, `esp` is the relative error and `n` is the number of iterations required.

```
function root = TanhNewton(f,x0,esp,n)
iter = 0; ea = 0; xn = x0;
fd = inline(char(diff(formula(f))),'x');
disp(' ----------------------------');
disp(' No     Root     f(Root)     %error ');
disp(' ---------------------------');
while (1)
  xnold = xn;
  xn =x0*(1+tanh(-f(x0)/(x0*fd(x0))));
  xnew = xn;
  disp(sprintf('%4d %10.4f %10.2f %8.2f',iter+1,xn,f(xn),ea));
  iter = iter + 1;
  if xn  = 0, ea = abs((xn - xnold)/xn) * 100; end
  x0 = xn;
  if ea <= esp | iter >= n, break, end
```

```
end
disp(' -------------------------------------');
disp(['Given function f(x) = ' char(f)]);
disp(sprintf('Approximate root = %10.10f',xn));
```

3.2 Implementation in Maple

The following a data type `TanhNewton(f,x0,n)` gives the implementation in Maple, where `f` is given non-linear transcendental function, `x0` is the initial approximation of the root, and `n` is the number of iterations required.

```
 TanhNewton:=proc(f,x0,n)
 local iten, fx0;
 for iten from 1 by 1 while iten < n+1
   do
     printf("Iteration %g :  ", iten);
     x0:=(x0*(1+tanh(-subs(x=x0,f)/(x0*subs(x=x0,diff(f,x))))));
     fx0:=subs(x = x0, f);
   end do;
 return x0,fx0;
end proc:
```

Sample computations using the implementation of the proposed algorithm are presented in Section 4.

4 Numerical Examples

This section provides some numerical examples to discuss the algorithm presented in Section 2 and comparisons are taken into account to conform that the algorithm is more efficient than other existing methods.

Example 4.1. *Consider a transcendental equations of the following type. We find approximate root using formulae given in equations* (3), (4) *and* (5) *to show the convergence of the proposed algorithm. To find approximate root, we start with an initial approximation as 1.5, and we have the exact real root is 1.93.*

$$\sin x = \frac{x^2}{4}, \tag{6}$$

Example 4.2. *Consider the following transcendental equations [12]. We compare the number of iterations required to get approximation root with accuracy of* 10^{-15}. *The numerical results are provided in Table 2.*

Table 1: Comparing approximate root using formulae given in equations (3),(4),(5)

Iteration No.	Newton-Raphson method (3)	Equation No. (4)	Equation No. (5)	Proposed method
1	2.140392773	2.101485079	2.102903404	2.104126848
2	1.952008946	1.946583227	1.946782457	1.946955261
3	1.933930574	1.933841859	1.933844596	1.933847006
4	1.933753780	1.933753767	1.933753767	1.933753767
5	1.933753763	1.933753762	1.933753762	1.933753763
6	1.933753763	1.933753762	1.933753762	1.933753763

a. $f(x) = \ln(x)$, *with initial approximation* 0.5.

b. $f(x) = x - e^{\sin(x)} + 1$, *with initial approximation* 4.

c. $f(x) = 11x^{11} - 1$, *with initial approximation* 1.

d. $f(x) = xe^{-x} - 0.1$, *with initial approximation* 0.1.

Table 2: Comparing No. of iterations by different methods

Fun.	Exact Root	Regula Falsi method	Newton Raphson method	Steffen method	Proposed method
a.	1.00000	27	Divergent	Failure	4
b.	1.69681 & 0	32	Not Convergent	Failure	6
c.	0.80413	101	7	Divergent	5
d.	0.11183	15	Failure	Failure	3

The numerical results given in Table 2 shows that the proposed method is more efficient than other methods.

Example 4.3. *This example gives the sample computation using MatLab and Maple implementation as described in Section 3. Consider a transcendental equation of the form*

$$f(x) = e^{-x} - x$$

with initial approximation of the root as 0.1.

Using MatLab implementation, we have the following computations.

```
f=inline('exp(-x) - x','x');
```

```
function root = TanhNewton(f,0.1,0.00001,10)
-----------------------------------------------------
 No    Root            f(Root)         %error
-----------------------------------------------------
 1    0.1999572486    0.618809       0.00
 2    0.3870375308    0.292028       48.33647057
 3    0.5501277253    0.0267484      29.64587804
 4    0.5670852558    9.09495e-05    2.990296490
 5    0.5671432899    8e-10          0.01023270504
 6    0.5671432903    2e-10          7.052891340*10^-8
 7    0.5671432907    -5e-10         7.052891332*10^-8
 8    0.5671432904    -0             5.289668502*10^-8
-----------------------------------------------------
Given function f(x) = exp(-x) - x
Approximate root = 0.5671432904
```

Using Maple implementation, we have the following computations.

```
> f:= exp(-x) - x:
> TanhNewton(f,0.1,10);
```

Iteration 1 :	0.1999572486
Iteration 2 :	0.3870375308
Iteration 3 :	0.5501277253
Iteration 4 :	0.5670852558
Iteration 5 :	0.5671432899
Iteration 6 :	0.5671432903
Iteration 7 :	0.5671432907
Iteration 8 :	0.5671432904
Iteration 9 :	0.5671432904
Iteration 10 :	0.5671432904

One can use the implementation of the proposed algorithm to speed up the manual calculations.

5 Conclusion

In this present work, we presented a new algorithm to compute an approximate root of a given transcendental function better than previous existing methods as illustrated. The proposed new algorithm was based on hyperbolic tangent function having better convergence than previous existing methods (for example, Bisection, Regula-Falsi, Newton-Raphson, Steffen method etc.). This proposed algorithm is useful for solving the complex real life problems. Implementation of the proposed algorithm in Matlab and Maple is also discussed.

6 Open Problem

Does proposed algorithm helpful to create another new method/algorithm using arc-tangent function to compute a real root of transcendental equation?

References

[1] C. Gutierrez, F. Gutierrez, M.-C. Rivara: Complexity of the bisection method, Theoretical Computer Science Vol. 382 (2007) 131–138.

[2] D. Bachrathy, G. Stépán: Bisection method in higher dimensions and the efficiency number, Mechanical Engineering, Vol. 56 (2) (2012) 81–86.

[3] Dowell-Jarrat: A modified Regula-Falsi method for computing the real root of an equation. BIT Numerical Mathematics. Vol.11 (1971), 168–174.

[4] E. Novak, K. Ritter, And H. Wozniakowski: Mathematics of computation, Vol. 64, (212), 1995, 1517–1539.

[5] Eiger, K. Sikorski, F. Stenger: A Bisection Method for Systems of Nonlinear Equations, ACM Transactions on Mathematical Softwares, Vol. 10 (4) 1984, 367–377.

[6] G.R. Wood: The bisection method in higher dimensions, Mathematical Programming Vol. 55 (1992) 319–337.

[7] J.C.Yakoubsohn: Numerical analysis of a bisection-exclusion method to find zeros of univariate analytic functions, Journal of Complexity Vol. 21 (2005) 652–690.

[8] J.S. Chitode: Numerical Techniques, (second revised edition), Technical Publications, Pune, 2008.

[9] Johan and Ronald. The Newton-Raphson Method. International Journal of Mathematical education in Science and Technology. Vol.26 (1995), No.2, 177–193.

[10] J. R. SHARMA and R. K. GOYAL: Fourth-order derivative-free methods for solving non-linear equations, International Journal of Computer Mathematics, Vol. 83, No. 1, 2006, 101–106.

[11] J. Chen, W. Li: An exponential regula falsi method for solving nonlinear equations, Numerical Algorithms (2006) 41: 327–338.

[12] J. Chen, W. Li: An improved exponential regula falsi methods with quadratic convergence of both diameter and point for solving nonlinear equations, Applied Numerical Mathematics, 57 (2007) 80–88.

[13] Jinhai Chen: New modified regula falsi method for nonlinear equations, Applied Mathematics and Computation, 184 (2007) 965–971.

[14] John Abbott: Quadratic interval refinement for real roots. Vol.48 (2014), No.1. ACM communications in computer algebra.

[15] M.N. Vrahatis, K.I. Iordanidis: A Rapid Generalized Method of Bisection for Solving System of Non-linear Equation, Numer. Math, Vol 49, 1986, 123–138.

[16] Mamta, V. Kanwar, V.K. Kukreja, Sukhjit Singh: On a class of quadratically convergent iteration formulae, Applied Mathematics and Computation, Vol. 166 (2005) 633–637.

[17] Mamta, V. Kanwar, V.K. Kukreja, Sukhjit Singh: On some third-order iterative methods for solving nonlinear equations, Applied Mathematics and Computation, Vol. 171 (2005) 272–280.

[18] M. A. Noor, K. I. Noor, W. A. Khan, F. Ahmad: On iterative methods for nonlinear equations, Applied Mathematics and Computation, Vol. 183 (2006) 128–133.

[19] M. Aslam Noor, F. Ahmad: Numerical comparison of iterative methods for solving nonlinear equations, Applied Mathematics and Computation, Vol. 180 (2006) 167–172.

[20] M. A. Noor, K. I. Noor: Three-step iterative methods for nonlinear equations, Applied Mathematics and Computation, Vol 183 (2006) 322–327.

[21] R.M. Somesundaram, R.M. Chandrasekaran: Numerical Methods with $C++$ programming. Prentice-Hall of India, 2005.

[22] S. Hussain, V. K. Srivastav, S. Thota: Assessment of Interpolation Methods for Solving the Real Life Problem, International Journal of Mathematical Sciences and Applications, 5 (1) (2015), 91–95.

[23] S. Thota: A New Root-Finding Algorithm Using Exponential Series, Ural Mathematical Journal, 5 (1) (2019), 83–90.

[24] S. Thota, T. Gemechu: A New Algorithm for Computing a root of Transcendental Equations Using Series Expansion, Southeast Asian Journal of Sciences, 7 (2) (2019), 106–114.

[25] S. Thota, T. Gemechu, P. Shanmugasundaram: New Algorithms for Computing Non-linear Equations Using Exponential Series, Palestine Journal of Mathematics, 10 (1) (2021), 128–134.

[26] S. Thota, V. K. Srivastav: Quadratically Convergent Algorithm for Computing Real Root of Non-Linear Transcendental Equations, BMC Research Notes, (2018) 11:909.

[27] S. Thota, V. K. Srivastav: Interpolation based Hybrid Algorithm for Computing Real Root of Non-Linear Transcendental Functions, International Journal of Mathematics and Computer Research, 2 (11) (2014), 729–735.

[28] Saied and LIAO: A new modification of False-Position method based on homotopy analysis method. Applied Mathematics and Mechanics. Vol.29.No.2, 223–228.

[29] Sagraloff and Mehlhorn: Computing real roots of real polynomials. 2013.

[30] S. Thota, S. D. Kumar: A New Reduction Algorithm for Differential-Algebraic Systems with Power Series Coefficients, Information Sciences Letters, 10 (1) (2021), 59–66.

[31] T. Gemechu: Some Root Finding With Extensions to Higher Dimensions, Mathematical Theory and Modeling, 7 (4) (2017), 1–13.

[32] T. Gemechu, S. Thota: *On New Root Finding Algorithms for Solving Nonlinear Transcendental Equations*, International Journal of Chemistry, Mathematics and Physics, 4 (2) (2020), 18–24.

[33] V. K. Srivastav, S. Thota, M. Kumar: A New Trigonometrical Algorithm for Computing Real Root of Non-linear Transcendental Equations, International Journal of Applied and Computational Mathematics, (2019) 5:44.

[34] X. Wu: Improved Muller method and Bisection method with global and asymptotic superlinear convergence of both point and interval for solving nonlinear equations, Applied Mathematics and Computation, Vol. 166 (2005) 299–311.

[35] X. Wu, Z. Shen, J. Xia: An improved regula falsi method with quadratic convergence of both diameter and point for enclosing simple zeros of non-linear equations, Applied Mathematics and Computation Vol. 144 (2003) 381–388.

[36] X.Wu, H. Wu: On a class of quadratic convergence iteration formulae without derivatives, Applied Mathematics and Computation, Vol. 107 (2000) 77–80.

[37] Y.R. Zhu, X.Y. Wu: A free derivative iteration method of order three having convergence of both point and interval for non-linear equations. Appl.Math.Comput.137(2003). pp. 49–55.