Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 2** : 1–9.
ISSN-0037-8712
doi:10.5269/bspm.80414

A Numerical Approach to Solve Nonlinear Univariate Equation Using Inverse Hyperbolic Tangent Function

K. Raghuram Bhattar[1], C. Balarama Krishna[2], B. Ravindar[3], G. Mahesh[4]

ABSTRACT: The fundamental idea behind the proposed method is to use the inverse hyperbolic tangent function, which offers a computationally efficient and mathematically efficient root-finding framework. MATLAB is used widely to implement the approach, guaranteeing a useful and repeatable computing environment. The proposed method is tested on many common benchmark problems found in mathematical modeling and engineering to check how well it works, how accurate it is, and how strong it remains under different conditions. These problems are carefully selected to test the method at different levels of nonlinearity and complexity. A detailed set of numerical experiments is carried out to evaluate how well the proposed method works. The results clearly show that the proposed method consistently performs somewhat better than several well-known root-finding techniques in both convergence speed and computational accuracy. A detailed error analysis, showing the accuracy and stability of the proposed method in different situations, further confirms its effectiveness and reliability. In addition, a detailed theoretical analysis is carried out to show that the proposed method exhibits quadratic convergence, which means the error decreases significantly with each iteration. Due to its fast convergence rate, the method is capable of providing highly accurate results using fewer iterations, thereby reducing the overall computational effort and time required for the solution.

Keywords: Inverse hyperbolic tangent function, univariate equation, quadratic convergence, root finding method.

Contents

1. Introduction

Finding the roots of nonlinear univariate transcendental equations is an essential task in many areas of science and engineering. Although numerous well established methods exist in the literature for solving such equations, researchers continue to develop new algorithms aimed at reducing the number of iterations and improving computational efficiency for faster and more accurate results. This continuous research effort has resulted in the development of several innovative techniques specifically designed to find the roots more quickly and efficiently.

Commonly used root-finding methods such as Bisection (BM), False Position (FP), Newton–Raphson (NR), Iteration (IM), and Secant (SM) methods are widely applied in practice. These techniques use iterative procedures that generate a sequence of values, each moving closer to the actual root of the equation. However, the convergence rate differs among these methods. Some achieve results rapidly, while others may need many more iterations to reach the required level of accuracy. Some methods may even fail to find the root, especially when the function is too complex or behaves irregularly.

Many researchers have worked to improve existing methods. For example, in [4], Jutaporn et al. proposed a new way to solve nonlinear equations using nonlinear regression. In [7] and [8], the authors modified the Regula Falsi method and found that these changes made it faster, often finding the root in just six steps. Neamvonk, in [8], continued this idea by further improving the Regula Falsi method and

2020 *Mathematics Subject Classification:* 65H04, 65H05.
Submitted November 27, 2025. Published March 14, 2026

studying how it crosses the x-axis in different regions. This new version was shown to work better than the original Regula Falsi method. Likewise, in [7], Naghipoor et al. presented an Improved Regula Falsi (IRF) method, which updates the classic version by using a linear interpolation technique to get a better first guess. This update made the method faster and required fewer steps to find the root. These studies show the growing effort to improve traditional methods for solving nonlinear transcendental equations and the need for more new ideas in this area.

In [1], the authors presented a group of algorithms based on two step iterative methods for solving nonlinear equations, showing that these methods work much better than the usual single step techniques. Gupta and Parida, in [9], developed a hybrid approach that mixes the Newton like method with the Bisection method to increase both convergence speed and stability when solving nonlinear equations. Zhu and Wu, in [12], proposed a derivative-free iteration method of third order, which can achieve convergence for both the solution and the interval when dealing with nonlinear equations.

In [11], the author developed a root finding method that calculates a non-zero real root of nonlinear equations by using an exponential series along with the Secant method. This approach shows quadratic convergence, making it both accurate and efficient. In [5], Mahesh et al. proposed a series based iterative method that quickly reduces errors and works very well for finding the roots of nonlinear equations. Interestingly, when the higher powers in the series are ignored, this method becomes the well-known Newton Raphson method, which also has quadratic convergence. Furthermore, in [2, 3], the authors improved the traditional Regula Falsi method by combining it with an exponential function, which made it faster and more reliable. In [6], Naghipoor et al. introduced an improved version of the classical Regula Falsi (false position) method to solve nonlinear equations. In [10], the authors discussed a class of derivative-free iterative methods with quadratic convergence for solving nonlinear equations. In [13], Mahesh et al. introduced a combination of the inverse sine series and the Newton–Raphson method to solve univariate nonlinear transcendental equations. Venkateshwarlu et al. in [14] solved nonlinear equations using an improved Newton–Raphson method and also discussed the convergence of the method. These studies clearly show the ongoing progress in iterative methods for solving nonlinear equations and the continued effort to design quicker and more effective algorithms.

2. Methodology

The iterative formula for the proposed method (PM), which uses the inverse hyperbolic tangent, is as follows:

$$t_{n+1} = t_n \left(1 + \operatorname{arctanh}\left(\frac{-f(t_n)}{t_n f'(t_n)} \Big/ \sqrt{1 - \left(\frac{f(t_n)}{t_n f'(t_n)}\right)^2}\right)\right) \tag{2.1}$$

Convergence of the PM:

Convergence Theorem: Let α denote a non-zero real exact root of $F(t)$, and let ε denote a small neighbourhood of α. Assume that $F''(t)$ exists and $F'(t) \neq 0$ within this region. Under these conditions, the formula in equation (2.1) produces a sequence of iterations that approaches the root α faster with every iteration.

Proof: The proposed iterative formula is

$$t_{n+1} = t_n \left(1 + \operatorname{arctanh}\left(\frac{-f(t_n)/(t_n f'(t_n))}{\sqrt{1-\left(\frac{f(t_n)}{t_n f'(t_n)}\right)^2}}\right)\right), \; n = 0, 1, 2, \ldots.$$

Replacing $\dfrac{-f(t_n)}{t_n f'(t_n)}$ with p, the above formula can be rewritten as

$$t_{n+1} = t_n \left(1 + \operatorname{arctanh}\left(\frac{p}{\sqrt{1-p^2}}\right)\right), \; n = 0, 1, 2, \ldots.$$

Expand using the Taylor's series expansion for $\tanh^{-1}(t) = t + \frac{t^3}{3} + \frac{t^5}{5} + \cdots$.

Thus, $t_{n+1} = t_n \left(1 + \frac{p}{\sqrt{1-p^2}} + \frac{1}{3}\left(\frac{p}{\sqrt{1-p^2}}\right)^3 + \cdots\right)$.

Since $h = \frac{f(t_n)}{f'(t_n)} = pt_n$ is a small value and $\left(\frac{f(t_n)}{f'(t_n)}\right)^2 = (pt_n)^2 = h^2 \to 0$, , we neglect higher powers of h. Rewriting the above, we get $t_{n+1} = t_n\left(1 + \frac{p}{\sqrt{1-p^2}}\right)$.

Substitute back $p = \dfrac{-f(t_n)}{t_n f'(t_n)}$, taking LCM and simplifying, Taking LCM and simplifying,

$$t_{n+1} = t_n \left(1 - \frac{\dfrac{f(t_n)}{t_n f'(t_n)}}{\sqrt{\dfrac{(t_n f'(t_n))^2 - (f(t_n))^2}{(t_n f'(t_n))^2}}}\right)$$

.

$$t_{n+1} = t_n \left(1 - \frac{\dfrac{f(t_n)}{t_n f'(t_n)}}{\dfrac{\sqrt{(t_n f'(t_n))^2 - (f(t_n))^2}}{t_n f'(t_n)}}\right)$$

$$t_{n+1} = t_n \left(1 - \frac{\dfrac{f(t_n)}{t_n f'(t_n)}}{\dfrac{t_n f'(t_n)\sqrt{1 - \frac{(f(t_n))^2}{(t_n f'(t_n))^2}}}{t_n f'(t_n)}}\right)$$

By canceling the term $t_n f'(t_n)$ from both the numerator and the denominator, and replacing $\frac{f(t_n)}{f'(t_n)}$ with h, the above equation simplifies to

$$t_{n+1} = t_n \left(1 - \frac{f(t_n)}{t_n f'(t_n)\sqrt{1 - \frac{h^2}{t_n^2}}}\right)$$

Again, neglecting h^2 (since $h^2 \to 0$), the above equation reduces to

$$t_{n+1} = t_n \left(1 - \frac{f(t_n)}{t_n f'(t_n)}\right)$$

$$t_{n+1} = t_n - \frac{f(t_n)}{f'(t_n)}$$

which shows that the proposed method reduces to the Newton–Raphson method and therefore achieves quadratic convergence.

3. Applications

The roots of nonlinear equations are very important in many areas, such as engineering, science, finance, optimization, and economics.

A. Engineering: In engineering, the roots of nonlinear equations are essential for solving complex real-world problems in many areas. They help determine equilibrium conditions in structural systems, study nonlinear behaviour in electrical circuits, and model phenomena such as vibrations, fluid flow, and heat transfer in mechanical and aerospace systems. In control engineering, root-finding methods are used to tune system parameters and design controllers that maintain system stability and ensure good performance.

B. Computer Science and Graphics: In computer science and graphics, root-finding methods are important for creating realistic visuals and accurate simulation-based models. They are used in ray tracing to compute intersections between light rays and surfaces, enabling correct lighting and shadow effects. In image processing and simulations, they help model shading, motion, and surface behaviour. In artificial intelligence and machine learning, these techniques are used in optimization tasks such as minimizing cost functions and finding gradient roots, which are essential for efficient model training.

C. Applied Sciences: In applied sciences, nonlinear equations play a key role in modeling and understanding complex natural processes. In physics, they describe nonlinear forces, equilibrium states, and wave motion. In chemistry, root-finding is used to calculate reaction rates and equilibrium concentrations. In biology, such equations help model population dynamics and nonlinear interactions among biological species.

D. Mathematics and Optimization: In mathematics and optimization, solving nonlinear equations forms the basis for many analytical and computational methods. Root-finding techniques are used to locate stationary points, minimize errors, reduce costs, and optimize designs. In scientific computing, iterative algorithms such as Newton's method and its improved variants efficiently solve large-scale nonlinear systems and boundary value problems.

E. Economics and Finance: In economics and finance, nonlinear equations are used to model complex market behaviour and financial systems. They help determine market equilibrium by solving nonlinear relationships between demand and supply. They also play an important role in investment and risk analysis, including computation of internal rates of return (IRR), pricing financial derivatives, and portfolio optimization. In game theory and economic modelling, solving nonlinear systems is essential for analyzing equilibrium conditions describing strategic interactions among market players.

F. Data Science and Machine Learning: In data science and machine learning, root-finding methods are central to optimization and model development. During model training, algorithms such as gradient descent work to find the roots of derivative functions to reduce loss and improve accuracy. Nonlinear regression depends on solving such equations to obtain best-fit parameters. In neural network optimization, finding points where the gradient becomes zero helps determine optimal weights and biases, leading to faster and more accurate learning.

4. Numerical Examples

In this section, the proposed method is tested on several numerical examples to demonstrate its effectiveness when compared with traditional root-finding methods such as the Bisection, False Position, and Newton–Raphson methods. All numerical computations are carried out in MATLAB, maintaining a high level of accuracy with an error tolerance of about 10^{-15}. The results obtained from these tests are presented in the following tables, providing a clear comparison of the performance and efficiency of the proposed method against the existing standard methods.

Example 1: $F_1(t) = \ln(t), \quad t_0 = 0.5$

Example 2: $F_2(t) = t - e^{\sin t} + 1, \quad t_0 = 2$

Example 3: $F_3(t) = 11t^{11} - 1, \quad t_0 = 1$

Example 4: $F_4(t) = te^{-t} - 0.1, \quad t_0 = 0.1$

Table 1: Numerical comparison between the proposed method (PM) and existing methods for different examples.

Example	**Exact root**	**BM**	**FPM**	**NRM**	**SM**	**PM**
1	1.00000000000000	53	23	Divergent	Failure	4
2	1.69681238680975	49	24	Not convergent	Failure	5
3	0.80413309750367	49	108	7	Divergent	7
4	0.11183255915896	42	17	Failure	Failure	4

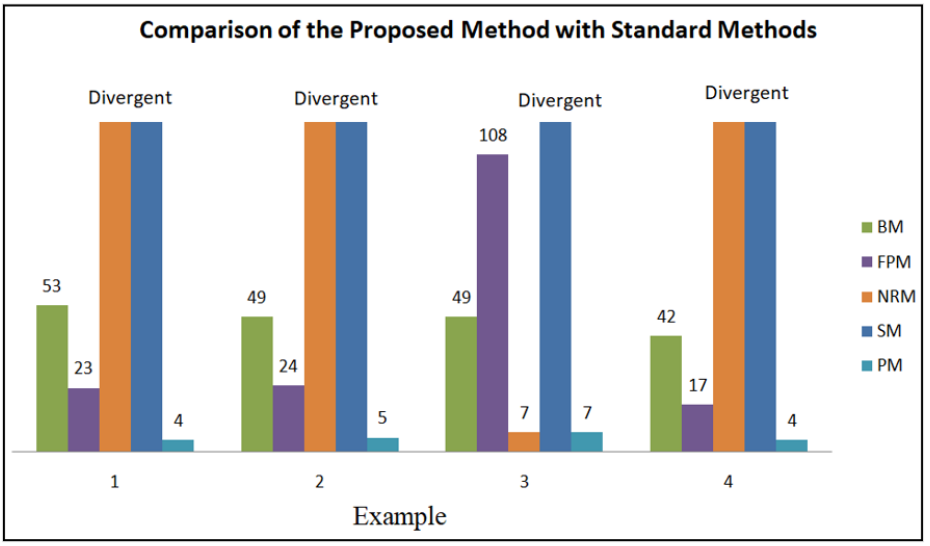


Table 2: Comparing the number of iterations by Chen & Li

Example	Exact root	t_0	Chen & Li [3]	Chen & Li [2]	PM
1	1.00000000000000	0.5	7	6	4
2	1.69681238680975	2	11	5	5
3	0.80413309750367	1	8	7	7
4	0.11183255915896	0.1	6	6	4

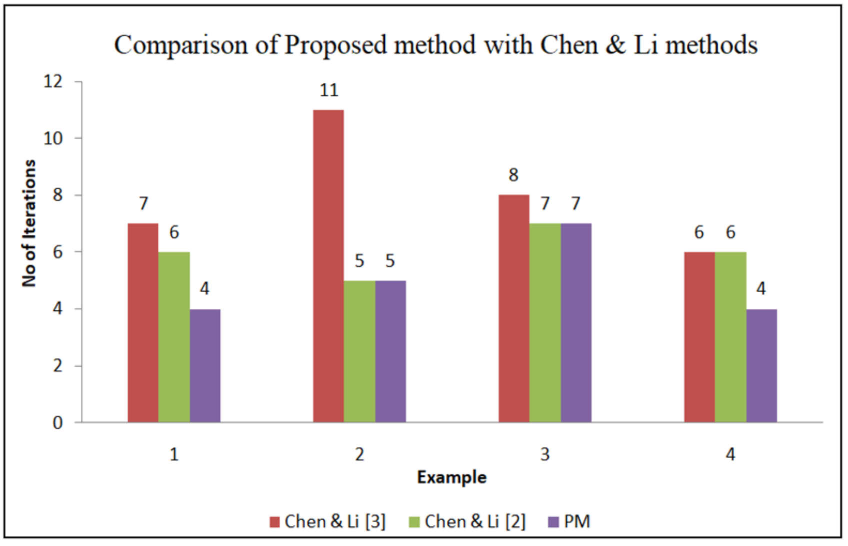
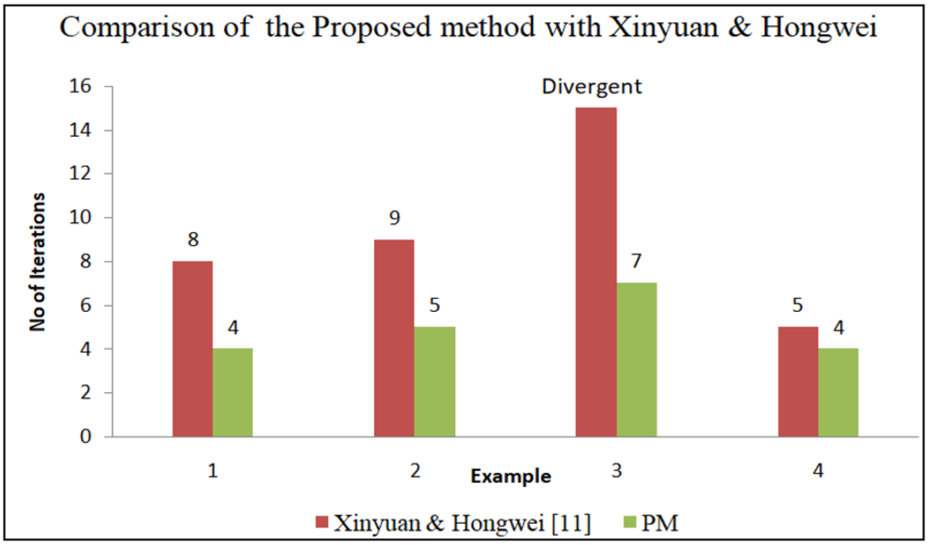


Table 3: Comparing the number of iterations by Xinyuan & Hongwei [11]

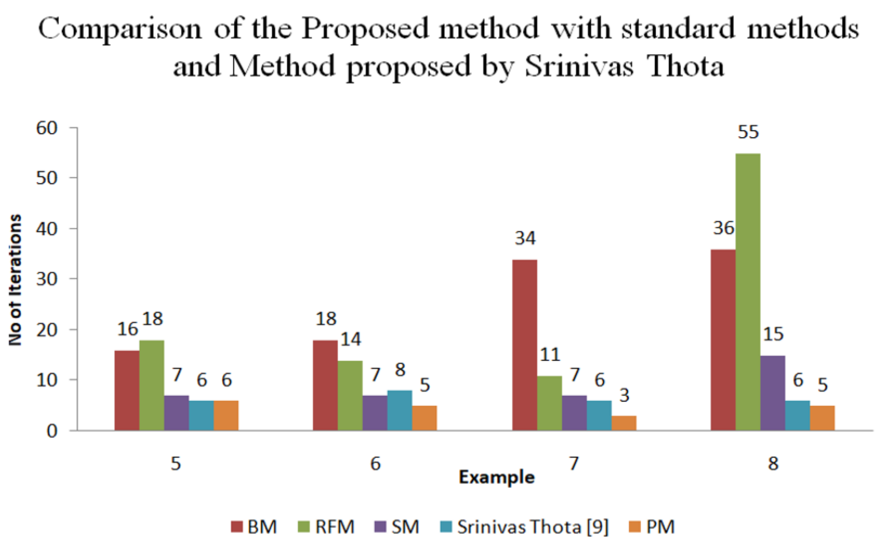
Example	Exact root	t_0	Xinyuan & Hongwei [11]	PM
1	1.00000000000000	0.5	8	4
2	1.69681238680975	2	9	5
3	0.80413309750367	1	Divergent	7
4	0.11183255915896	0.1	5	4



Example 5: $F_5(t) = t^6 - t - 1 = 0, \quad t_0 = 1,\ t_1 = 1.5$
Example 6: $F_6(t) = e^t - t - 2, \quad t_0 = 1,\ t_1 = 2$
Example 7: $F_7(t) = 8 - 4.5(t - \sin t), \quad t_0 = 2,\ t_1 = 3$
Example 8: $F_8(t) = te^{-t} - 0.1, \quad t_0 = 0,\ t_1 = 0.1$

Table 4: Comparing the number of iterations by Srinivas Thota [9]

Example	**Exact root**	**BM**	**RFM**	**SM**	**Srinivas Thota [9]**	**PM**
5	1.134724138	16	18	7	6	6
6	1.146193221	18	14	7	8	5
7	2.43046574	34	11	7	6	3
8	0.11183256	36	55	15	6	5



Example 1: $F_1(t) = \log(t)$

Table 5 below presents a comparison between the proposed method and several existing methods, using the initial guesses $t_0 = 0.5$ and $t_1 = 1.2$. Here, n represents the number of iterations, and t_n denotes the corresponding approximation of the root.

Table 5: Comparison of the number of iterations and approximated roots for Example 1

BM		**FPM**		**PM**	
It. no.	t_n	**It. no.**	t_n	**It. no.**	t_n
1	0.85	1	1.000001949490732	1	0.974880266138
2	1.025	2	1.000000543230269	2	0.999695212039
⋮	⋮	⋮	⋮	3	0.999999953571
53	1.000000000000000	23	1.000000000000000	4	1.000000000000

Example 2: $F_2(t) = t - e^{\sin t} + 1$

Table 6 below presents a comparison between the proposed method and several existing methods, using the initial guesses $t_0 = 0.5$ and $t_1 = 1.2$. Here, n represents the number of iterations, and t_n denotes the corresponding approximation of the root.

Table 6: Comparison of the number of iterations and approximated roots for Example 2

BM		**FPM**		**PM**	
It. no.	t_n	**It. no.**	t_n	**It. no.**	t_n
1	1.75	1	1.645067953924812	1	1.768578616317960
2	1.625	2	1.685074247441264	2	1.701049583555700
3	1.6875	3	1.694253896381327	3	1.696829845706660
⋮	⋮	⋮	⋮	4	1.696812387109420
49	1.696812386809751	24	1.696812386809751	5	1.696812386809751

Example 3: $F_3(t) = 11t^{11} - 1$
Table 7 below presents a comparison between the proposed method and several existing methods, using the initial guesses $t_0 = 0$ and $t_1 = 1$. Here, 'n' represents the number of iterations, and t_n denotes the corresponding approximation of the root.

BM		**FPM**		**PM**	
It. no.	t_n	It. no.	t_n	It. no.	t_n
1	0.5	1	0.090909090909091	1	0.916880790300304
2	0.75	2	0.173553719005368	2	0.852953285739974
3	0.875	3	0.248685195862868	3	0.815901267086100
4	0.8125	4	0.316986388027222	4	0.804944458359302
⋮	⋮	⋮	⋮	⋮	⋮
49	0.804133097503664	108	0.804133097503664	7	0.804133097503664

Example 4: $F_4(t) = te^{-t} - 0.1$
Table 8 below shows a comparison between the proposed method (PM) and several existing methods, using the initial guesses $t_0 = 0$ and $t_1 = 1$. Here, 'n' represents the number of iterations, and t_n denotes the corresponding approximation of the root.

BM		FPM		PM	
It. no.	t_n	It. no.	t_n	It. no.	t_n
1	0.5	1	0.271828182845905	1	0.1118210256118290
2	0.25000000000000000	2	0.131236149572146	2	0.1118325590176700
⋮	⋮	⋮	⋮	⋮	⋮
42	0.11183255915898400	17	0.11183255915896300	3	0.1118325591589630

5. Conclusion

The proposed algorithm shows that the approximate root of a given nonlinear equation is significantly more accurate compared to earlier methods such as the Bisection, False Position, Newton–Raphson, and Steffensen methods. This is shown through several representative numerical examples. Moreover, the use of the inverse hyperbolic tangent function in the proposed method is shown to act as a limiting case of the Newton–Raphson method. The convergence rate analysis confirms that the proposed method achieves quadratic convergence, highlighting its superior efficiency and reliability.

Table 2 clearly shows that the proposed method reaches the root much faster than the methods presented in [9], [11], and [12]. All numerical experiments were performed in MATLAB, ensuring high accuracy and efficient computation.

References

1. Aslam Noor. M, Ahmad. F, Javeed. Sh, "Two-step iterative methods for nonlinear equations", J. Appl. Math. Comput., 181 (2006) 1068–1075.

2. Chen. J, Li. W, "An exponential regula falsi method for solving nonlinear equations, Numerical algorithms" Vol. 41 (4) (2006) 327–338.

3. Chen. J, Li. W, "An improved exponential regula falsi methods with quadratic convergence of both diameter and point for solving nonlinear equations", Appl. Numer. math. Vol. 57 (1) (2007) 80–88.

4. Neamvonk, J., Phuenaree, B., & Neamvonk, A. (2015). "A new method for finding root of nonlinear equations by using nonlinear regression", Asian Journal of Sciences, 3(6), 818–822.

5. Mahesh. G, Swapna. G, Venkateshwarlu. K, "An iterative method for solving non-linear transcendental equations", J. Math. Comput. Sci. 10 (5) (2020), 1633–1642.

6. Naghipoor J., Ahmadian S. A. and Soheili Ali R., "An Improved Regula Falsi Method for Finding Simple Zeros of Nonlinear Equations", App. Math. Sci., vol. 2, no. 8, pp.381–386, 2008.

7. Neamvonk A., "A Modified Regula Falsi Method for Solving Root of Nonlinear Equations", Asian Journal of Applied Sciences, vol. 3, no. 4, pp. 776–778, 2015.

8. Parida P.K. and Gupta D.K., "A cubic convergent iterative method for enclosing simple roots of nonlinear equations", Appl. Math. and compu., vol. 187, pp.1544–1551, 2007.

9. Srinivasarao Thota, "A new root-finding algorithm using exponential series", Ural Mathematical Journal, Vol. 5, No. 1, 2019, pp. 83–90.

10. Wu. X, Wu. H, "On a class of quadratic convergence iteration formulae without derivatives", Appl. Math. Comput. 107 (2000), 77–80.

11. Zhu. Y, Wu. X, "A free-derivative iteration method of order three having convergence of both point and interval for nonlinear equations", Applied Mathematics and Computation, 137 (2003) 49–55.

12. K. Venkateshwarlu, V. S. Triveni, G. Mahesh, G. Swapna, "A new trigonometrical method for solving non-linear transcendental equations", Journal of Mathematics and Computer Science, Volume 25 (2), pp.176–181, 2021.

13. G. Mahesh, V. S. Triveni, G. Swapna, Venkateshwarlu. K, "A New Root-Finding Method for Univariate Non-Linear Transcendental Equations With Quadratic Convergence", Communications in Mathematics and Applications, Volume 13 (1), pp.1–8, 2022.

14. K. Venkateshwarlu, G.Mahesh , G.Swapna, "An Improved Newton-Raphson Method With Quadratic Convergence For Solving Nonlinear Transcendental Equations", Journal of Mathematical and Computational Science, Volume 12, pp.1–11, 2022.

[1,2,3]*Department of Mathematics School of CS & AI, SR University, Warangal, Telangana, India.*
E-mail address: 2406c9m006@sru.edu.in,c.balaramakrishna@sru.edu.in, b.ravindar@sru.edu.in

and

[1,4]*Department of Humanities and Sciences, Keshav Memorial Institute of Technology, Narayanaguda, Hyderabad, Telangana, India.*
E-mail address: gattumahesh790@gmail.com