

A Comparative Study of Newton's Method, Halley's Method, Super-Halley's Method and Modified Super-Halley's Method for Solving Nonlinear Equations

Hadfina Jahizah binti Harun¹, Ruhana binti Jaafar^{2*}, Nur Solihah Khadhiah binti Abdullah³, Zanariah binti Mohd Yusof⁴, Nor Aini binti Hassanuddin⁵

^{1,2,3,4} Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM) Kuala Terengganu Branch, 21080 Kuala Terengganu, Terengganu, Malaysia

⁵ Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM) Terengganu Branch, 23000 Dungun, Terengganu, Malaysia

*Corresponding Author

DOI: <https://doi.org/10.47772/IJRISS.2026.100801053>

Received: 09 September 2026; Accepted: 14 September 2026; Published: 24 September 2026

ABSTRACT

This study evaluates the efficiency of four iterative methods for solving nonlinear equations, namely Newton's method, Halley's method, Super-Halley's method and Modified Super-Halley's method. The objective is to identify the most effective method for approximating the roots of nonlinear equations. Ten nonlinear test functions with various type such as combination of polynomial, trigonometric, logarithmic and exponential were solved under different initial guesses and two tolerance levels (10^{-10} and 10^{-15}). All methods were coded and executed in MAPLE 2016 and SigmaPlot was used to generate performance profiles based on the number of iterations and CPU time. The results show that Super-Halley's method required the fewest iterations across nearly all test functions. Meanwhile, Newton's method achieved the lowest CPU time overall. These findings indicate that a higher theoretical order of convergence such as Modified Super-Halley's method does not automatically translate into practical efficiency. This is because higher order methods are more sensitive to the choice of initial guess and carry a greater computational cost per iteration.

Keywords: Newton's Method, Halley's Method, Super-Halley's Method, Modified Super-Halley's Method, Nonlinear Equations, Performance Profile

INTRODUCTION

Numerical analysis is a branch of mathematics that focuses on the development of computational algorithms for obtaining approximate solutions to mathematical problems that cannot be solved analytically (Leader, 2022). Among these problems, nonlinear equations are particularly important because these equations arise in numerous scientific, engineering and industrial applications. Most mathematical models often lead to problems that require the use of computational rather than analytical solutions (Hildebrand, 1987). These methods have been widely applied in various fields such as engineering, physics and computer science. For example, iterative techniques have been successfully employed to extract electrical parameters for photovoltaic solar cells in engineering application (Rasheed et al., 2021).

Among the available iterative methods, Newton's method is most widely used method for solving nonlinear equations because of its simple formulation and quadratic convergence. However, its convergence is highly dependent on the selection of an appropriate initial guess and the method may diverge when the initial approximation is far from the actual root or when the derivative is small or zero (Bhatt et al., 2024). To overcome these limitations, several higher-order iterative methods have been developed. Therefore, Halley's method was developed using second-derivative information to achieve cubic convergence (Liu et al., 2026). Despite the improvement, the second derivative increase the computational cost per iteration. Super-Halley's

method further modifies Halley's correction term to improve numerical stability while maintaining third-order convergence (Herceg & Herceg, 2015). Subsequently, Kou and Li (2007) introduced the Modified Super-Halley's method which achieves fourth-order convergence without increasing the number of function or derivative evaluations per iteration, making it a promising alternative for solving nonlinear equations efficiently.

Although these iterative methods are well documented individually in literature, comparative evaluations under identical computational setting remains limited. Existing studies often focus on the convergence properties or theoretical development of a particular method rather than comparison of these methods in term of both the number of iterations and CPU time across a common set of test functions, initial guesses and tolerance levels. Furthermore, the choice of initial guess significantly influences the convergence behaviour, number of iterations, computational time and overall reliability of iterative methods. This issue is particularly evident in higher-order methods, whose convergence may become highly unstable when the initial guess is not sufficiently close to the root (Saxena, 2017).

Therefore, this study presents a comparative performance analysis of Newton's method, Halley's method, Super-Halley's method and Modified Super-Halley's method for solving nonlinear equations. The four methods are implemented using the same set of nonlinear test functions, initial guesses and tolerance level to ensure a fair and consistent evaluation. The performance of these methods is assessed based on the number of iterations required for convergence and CPU time. Performance profiles are employed to provide a comprehensive comparison of the computational efficiency of these methods. The findings of this study are expected to contribute empirical evidence that assists researchers in selecting appropriate iterative methods for solving nonlinear equations in both academic research and practical application.

LITERATURE

Newton's method, also known as the Newton-Raphson method, is one of the well-known method used for solving nonlinear equations (Herceg & Herceg, 2015). It was originally formulated by Newton in 1669 and the same idea was later applied to polynomials by Raphson in 1690 (Yang, 2020). Newton's method converges with second-order convergence with the initial guess close to a simple root and the derivative does not equal to zero (Wang & Guo, 2025). Therefore, Newton's method becomes a standard baseline method in numerical analysis because of its simple formulation and fast convergence near the solution. Despite these advantages, Newton's method is sensitive to the choice of initial guess and may fail to convergence or diverge if the initial guess is far from the root or if the derivative becomes small or zero (Herceg et al., 2025). These limitations motivate the development of higher-order and modified methods.

Halley's method was originally introduced by Edmond Halley (1656-1742), the well-known English astronomer and mathematician (Nelson, 2008). It extends Newton's method by reducing the curvature term in the correction step used to define the Newton iteration which speeds up convergence (Fernández & Verón, 2025). Unlike Newton's method, Halley's method uses second-order derivative information that allows for more accurate local approximation of the nonlinear function. Over time, Halley's method has been recognized as a core higher-order iterative technique (Suleiman, 2009). One advantage of Halley's method is its third-order convergence which makes it converge faster than Newton's method (Herceg & Herceg, 2015). However, since Halley's method requires the second derivative, its computational cost per iteration also increases. Therefore, this has led the researchers to explore the balance between convergence speed and computational cost through derivative-free and modified Halley-type methods.

Super-Halley's method was introduced to enhance Halley's method by modifying the correction factor applied to the Newton step. This enhancement aim to improve convergence behaviour while retaining the Newton-type iterative structure (Chun & Ham, 2008). This method is part of a broader family of third-order convergent iterative methods such as Chebyshev's method and Halley's method which are generated from a common iterative formulation through different parameter selections (Kanwar et al., 2012). Studies have shown that Super-Halley's method maintains third-order convergence (Herceg & Herceg, 2015). Compared with Halley's method, Super-Halley's method adjusts the correction factor by applying a stronger rational multiplier to the Newton step which can enhance efficiency per iteration and make it faster than Halley's method (Gutierrez &

Hernández, 2001). However, Super-Halley's method still depends on the second derivative which increases its computational cost per iteration and limits its practical use (Herceg et al., 2025).

To further improve the efficiency of higher-order iterative methods, Kou and Li (2007) proposed the Modified Super-Halley's method to overcome the limited order of convergence found in existing Newton-type methods. This method achieves fourth-order convergence without increasing the number of function or derivative evaluations per iteration compared to Super-Halley's method. This improvement enhances the theoretical efficiency index while maintaining a computational structure similar to its predecessor. Singh and Behera (2022) have demonstrated that the Modified Super-Halley's method is capable to achieve the desired accuracy with fewer iterations while maintaining stable convergence characteristics.

Overall, the existing literature demonstrates a clear progression in the development of Newton-type iterative methods from Newton's method through Halley's method, Super-Halley's method and Modified Super-Halley's method. These methods exhibit increasingly higher orders of convergence, which enable higher-order methods to achieve the desired accuracy with fewer iterations. However, previous studies have compared selected iterative methods using measures such as convergence order, efficiency index and the number of iterations, but a comprehensive comparison involving all four methods remains limited (Fernández & Verón, 2025). The novelty of the present study lies in providing a systematic and consistent comparative evaluation of Newton's method, Halley's method, Super-Halley's method, and Modified Super-Halley's method under the same computational framework. In particular, all four methods are evaluated using the same set of nonlinear test functions, identical initial guesses, and consistent stopping criteria based on prescribed tolerance levels of 10^{-10} and 10^{-15} . The comparison considers not only the theoretical order of convergence but also practical computational measures, including the number of iterations and computational efficiency, thereby providing a more comprehensive assessment of the trade-off between convergence order, accuracy, and computational cost. Therefore, this study conducts a systematic comparison of these four iterative methods to provide a fair evaluation of their computational efficiency and practical performance in solving nonlinear equations.

METHODOLOGY

Iterative Methods

This study compares the performance of four Newton-type iterative methods, namely Newton's method, Halley's method, Super-Halley's method and Modified Super-Halley's method for solving nonlinear equations. The four methods were selected because these algorithms represent successive developments in Newton-type algorithms with increasing orders of convergence. Newton's method achieves second-order convergence. Meanwhile, Halley's method and Super-Halley's method achieve third-order convergence. Modified Super-Halley's method further improves the convergence to fourth order.

For convenience, the following auxiliary functions are used in the iterative formulas of the higher-order methods:

$$u(x_n) = \frac{f(x_n)}{f'(x_n)},$$

$$L(x_n) = \frac{f(x_n)f''(x_n)}{[f'(x_n)]^2},$$

$$K(x_n) = \frac{f''\left(x_n - \frac{f(x_n)}{3f'(x_n)}\right)f(x_n)}{[f'(x_n)]^2}.$$

The iterative formulas and corresponding orders of convergence of the four methods are summarized in Table 1.

Table 1. Formulas of the Iterative Methods

Method	Iterative Formula	Order of Convergence
Newton's Method	$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$	2
Halley's Method	$x_{n+1} = x_n - u(x_n) \frac{2}{2 - L(x_n)}$	3
Super-Halley's Method	$x_{n+1} = x_n - u(x_n) \left(\frac{2 - L(x_n)}{2(1 - L(x_n))} \right)$	3
Modified Super-Halley's Method	$x_{n+1} = x_n - u(x_n) \left(1 + \frac{1}{2} \cdot \left(K(x_n) + \frac{[K(x_n)]^2}{1 - \alpha K(x_n)} \right) \right)$ $\alpha = 0$	4

Test Functions

A total of ten test functions were selected from previous studies such as Gutierrez and Hernández (2001), Herceg and Herceg (2015), Kou and Li (2007), Kumar et al. (2013) and Thukral (2015) to evaluate the performance of the four iterative methods. The selected functions consist of combination of polynomial, exponential, logarithmic and trigonometric equations. Each test function was evaluated using two initial guesses, both near and far from the root. The test functions and their corresponding initial guesses are presented in Table 2.

Table 2. Test functions and their corresponding initial guesses

No	Test functions	Near Initial Guess	Far Initial Guess
1	$(x-1)^3 - 2$	2	2.5
2	$x^5 + x^4 + 4x^2 - 20$	1	1.2
3	$\sin^2(x) - x^2 + 1$	1	2
4	$\ln(x) + \sqrt{x} - 5$	8.3	8.4
5	$x^3 - \sin^2(x) + 3\cos(x) + 5$	-1	-3
6	$\sin(x) - \frac{x}{2}$	2	3
7	$e^x \sin(x) - 2x - 5$	-2	-3
8	$x^2 - e^x - 3x + 2$	0.5	1
9	$\tan(x) - e^x - 1$	1.3	1.4
10	$(xe^x - \sin^2(x) + 3\cos(x) + 5)^{10}$	-1	-1.5

Experimental Setup

All iterative methods were implemented using Maple 2016 to ensure a consistent computational environment throughout the study. Each method was applied to the same set of test functions using identical initial guesses and stopping criteria to ensure a fair comparison. Two tolerance levels, $\varepsilon = 10^{-10}$ and $\varepsilon = 10^{-15}$, were considered to examine the effect of convergence accuracy on the performance of the iterative methods. For each method, the iteration was terminated when the absolute difference between two consecutive approximations satisfied $|x_{n+1} - x_n| < \varepsilon$, where ε denotes the prescribed tolerance. Thus, the stopping criterion was applied consistently using $\varepsilon = 10^{-10}$ and $\varepsilon = 10^{-15}$, respectively, for all test functions and iterative methods.

RESULTS AND DISCUSSIONS

Number of Iterations

The convergence efficiency of the four iterative methods was evaluated based on the number of iteration (NOI) required to satisfy the stopping criteria. Each method was applied to the same set of ten test functions using same initial guesses and tolerance levels to ensure a fair comparison. The results are presented in Table 3.

Table 3. Comparison of the number of iterations for the four iterative methods

Test Function	x_0	10^{-10}				10^{-15}			
		NM	HM	SHM	MSHM	NM	HM	SHM	MSHM
TF1	2	6	4	4	5	6	4	4	6
	2.5	5	4	4	5	6	4	4	5
TF2	1	7	4	4	6	7	5	5	6
	1.2	6	4	3	5	6	4	4	6
TF3	1	6	4	4	5	7	5	4	6
	2	6	4	4	5	6	5	4	6
TF4	8.3	3	3	3	4	4	3	3	5
	8.4	4	3	3	6	4	3	3	7
TF5	-1	6	4	4	6	6	4	4	6
	-3	6	4	4	8	7	5	5	8
TF6	2	5	3	3	4	5	4	4	5
	3	6	5	4	5	6	5	5	6
TF7	-2	4	3	3	5	5	4	4	6
	-3	4	3	3	8	5	4	4	9
TF8	0.5	4	3	3	5	5	4	4	5
	1	4	4	4	6	5	4	4	6
TF9	1.3	6	4	4	6	7	4	5	7
	1.4	5	3	4	5	6	4	6	5
TF10	-1	180	99	28	180	290	156	43	290
	-1.5	192	104	25	192	301	162	39	301

Table 3 shows the number of iterations required by Newton's method, Halley's method, Super-Halley's method and Modified Super-Halley's method to achieve convergence. The methods exhibited different convergence behaviours depending on the characteristic of test functions, initial guess and tolerance level. Higher-order methods generally require fewer iterations because of the higher convergence rate. However, the performance varied across different test functions indicating that convergence behaviour also influenced by choice of initial guesses.

CPU time

In addition to convergence efficiency, the computational performance of each iterative method was assessed using CPU time. The recorded execution times for all test functions are summarized in Table 4.

Table 4. Comparison of CPU time for the four iterative methods

Test Function	x_0	10^{-10}				10^{-15}			
		NM	HM	SHM	MSHM	NM	HM	SHM	MSHM
TF1	2	0.2344	0.2813	0.3125	0.3594	0.1094	0.1875	0.2031	0.3750
	2.5	0.2188	0.3125	0.3125	0.5313	0.0938	0.2344	0.1875	0.3125
TF2	1	0.0938	0.2344	0.2188	0.3750	0.0938	0.2344	0.2813	0.3281
	1.2	0.1406	0.2969	0.2344	0.3750	0.0938	0.2031	0.2344	0.3125
TF3	1	0.0781	0.2500	0.2031	0.2813	0.1094	0.1875	0.2031	0.3750

	2	0.0781	0.1875	0.2031	0.2656	0.0938	0.2344	0.1875	0.3125
TF4	8.3	0.1094	0.2500	0.2188	0.3594	0.0938	0.2031	0.1875	0.2656
	8.4	0.0781	0.2188	0.2188	0.3125	0.0781	0.2188	0.2188	0.2969
TF5	-1	0.0938	0.2500	0.2031	0.3125	0.1250	0.2344	0.2344	0.3438
	-3	0.0938	0.2344	0.2344	0.2813	0.1094	0.2344	0.2344	0.3750
TF6	2	0.0781	0.1719	0.2188	0.2656	0.0625	0.1563	0.2031	0.2813
	3	0.1094	0.2031	0.1875	0.2969	0.0938	0.2344	0.2031	0.2969
TF7	-2	0.1094	0.1094	0.2188	0.3125	0.0938	0.1094	0.2344	0.3125
	-3	0.0781	0.1094	0.2188	0.3125	0.0938	0.1094	0.1875	0.3281
TF8	0.5	0.1094	0.1094	0.1719	0.2813	0.0781	0.1563	0.2031	0.3594
	1	0.0938	0.1094	0.2500	0.3438	0.1094	0.0938	0.2188	0.3125
TF9	1.3	0.0938	0.1094	0.1250	0.1406	0.0781	0.1250	0.1250	0.1406
	1.4	0.0781	0.0625	0.2344	0.2344	0.0781	0.1094	0.2500	0.2344
TF10	-1	0.2031	0.2188	0.2969	0.7344	0.2656	0.2344	0.3438	0.8906
	-1.5	0.2344	0.2031	0.3125	0.7188	0.2500	0.2500	0.3125	0.8594

Although higher-order iterative methods generally require fewer iterations to reach convergence, additional derivative evaluations within each iteration resulting in higher computational cost. Consequently, the method with the fewest iterations does not necessarily achieve the shortest execution time. The CPU time results presented in Table 4 illustrate the trade-off between convergence efficiency and computational cost.

Performance Profile

The efficiency of the four iterative methods was evaluated using the number of iterations and CPU time as performance indicators. The number of iterations represents the total iterations required by each method to reach the stopping criteria, while CPU time measures the computational time required to obtain the approximate root. These two indicators were selected because these metrics reflect both convergence efficiency and computational performance. In addition, performance profiles were constructed based on the methodology proposed by Dolan and Moré (2002) to provide a comprehensive comparison of the relative performance of the four methods across all test functions.

To provide an overall comparison of the four methods were constructed based on the number of iterations and CPU time. Performance profile offers a graphical representation of the relative performance of each method by considering all test functions simultaneously.

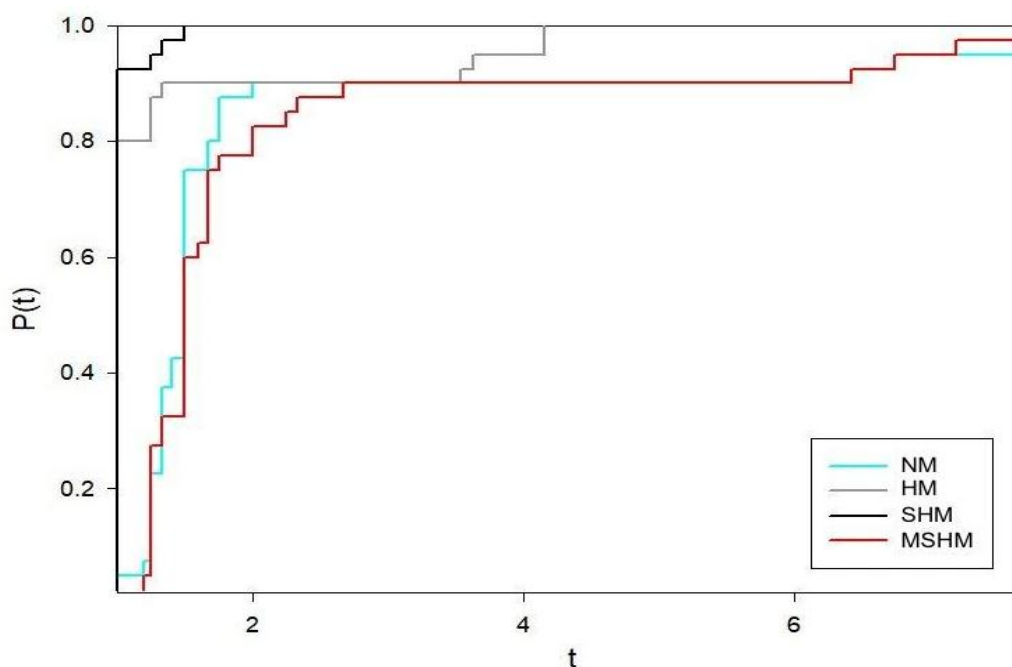


Fig 1. Performance profile based on the number of iterations

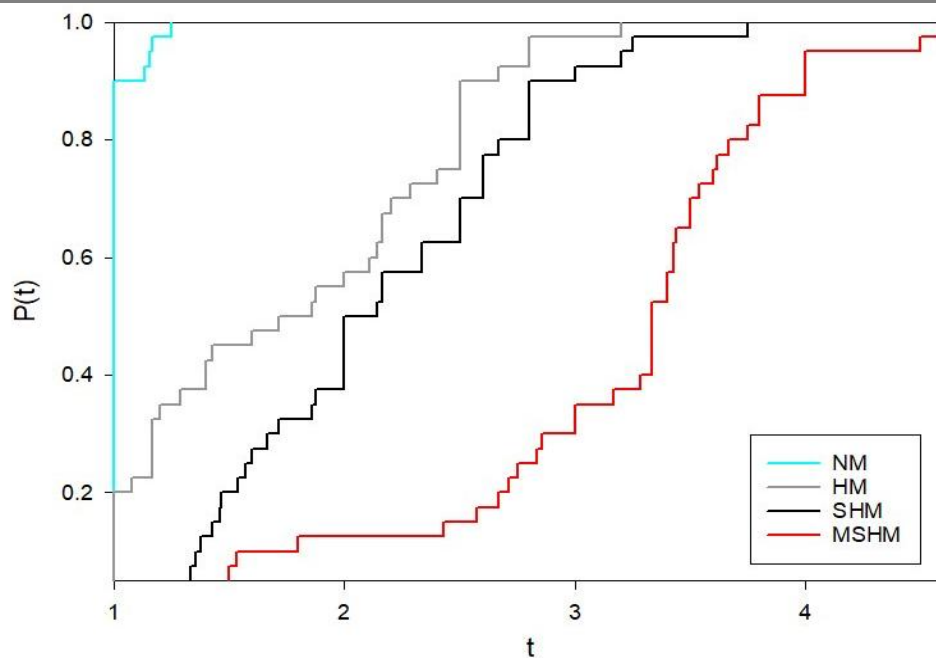


Fig 2. Performance profile based on CPU time

Figures 1 and 2 present the performance profiles of the four methods based on the number of iterations and CPU time. The performance profile provides an overall comparison of the computational performance of the methods across all nonlinear test functions. Figure 1 show that SHM achieved the highest performance in terms of the number of iterations. Meanwhile, Figure 2 shows NM recorded the best performance in terms of CPU time. This indicates that a higher convergence order does not necessarily correspond to lower computational time. The complexity of derivative evaluations also influences the overall execution time.

CONCLUSION AND RECOMMENDATIONS

This study compared the performance of Newton's method, Halley's method, Super-Halley's method and Modified Super-Halley's method for solving nonlinear equation using ten benchmark test functions under two initial guesses and two tolerance levels in Maple 2016. All four methods successfully converged to the actual root for all test functions. The computational performance was evaluated based on the number of iterations and CPU time with the overall results further illustrated using performance profiles.

Based on the number of iterations, Super-Halley's method demonstrated the best convergence performance by requiring the fewest iterations of most test functions. In terms of CPU time, Newton's method achieved the shortest execution time because each iteration requires only first-derivative evaluations. Meanwhile, Halley's method, Super-Halley's method and Modified Super-Halley's method require additional second derivative evaluations. Although, Modified Super-Halley's method possesses fourth-order convergence in theory, it demonstrated comparatively lower practical performance for the more complex test functions tests findings indicate that a higher theoretical convergence order does not necessarily guarantee a high computational efficiency. This is because the complexity of derivative evaluations and the choice of initial approximations also influence practical performance.

Future studies may consider a wider range of tolerance levels, such as 10^{-8} and 10^{-12} , to further investigate the influence of stopping criteria on convergence behaviour. In addition, derivative-free higher-order iterative methods could be explored to reduce the computational cost associates with second-derivative evaluations while maintaining rapid convergence.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Terengganu Kampus Kuala Terengganu for providing the necessary

resources and support throughout this research. Special thanks are extended to colleagues and mentors for their invaluable assistance and guidance, as well as to any funding organizations that contributed to this project

REFERENCES

1. Bhatt, M. R., Dhungana, H. L., & Dhakal, G. R. (2024). A Numerical Perspective on Solving Non-Linear Equations: Newton Vs. Bisection. *Academic Journal of Mathematics Education*, 7(1), 74-80.
2. Chun, C., & Ham, Y. (2008). Some second-derivative-free variants of super-Halley method with fourth-order convergence. *Applied Mathematics and Computation*, 195(2), 537-541.
3. Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2), 201-213.
4. Fernández, J. A. E., & Verón, M. Á. H. (2025). *Convexity in Newton's Method*. Springer.
5. Gutierrez, J. M., & Hernández, M. A. (2001). An acceleration of Newton's method: super-Halley method. *Applied Mathematics and Computation*, 117(2-3), 223-239.
6. Herceg, D., & Herceg, D. (2015). A family of methods for solving nonlinear equations. *Applied Mathematics and Computation*, 259, 882-895.
7. Herceg, D., Herceg, D., Odry, P., & Tadić, V. (2025). A New Family of Methods for Solving Systems of Nonlinear Equations. *Mathematics*, 13(24), 3945.
8. Hildebrand, F. B. (1987). *Introduction to numerical analysis*. Courier Corporation.
9. Kanwar, V., Tomar, S., Singh, S., & Kumar, S. (2012). Note on super-halley method and its variants. *Tamsui Oxford Journal of Information and Mathematical Sciences*, 28(2), 191-216.
10. Kou, J., & Li, Y. (2007). A family of modified super-Halley methods with fourth-order convergence. *Applied Mathematics and Computation*, 189(1), 366-370.
11. Kumar, M., Singh, A. K., & Srivastava, A. (2013). Various Newton-type iterative methods for solving nonlinear equations. *Journal of the Egyptian mathematical society*, 21(3), 334-339.
12. Leader, J. J. (2022). *Numerical analysis and scientific computation*. Chapman and Hall/CRC.
13. Liu, G., Pal, S., & Ponnusamy, S. (2026). On the dynamics of Halley's method: G. Liu et al. *Monatshefte für Mathematik*, 209(4), 705-733.
14. Nelson, D. (2008). *The Penguin dictionary of mathematics*. Penguin UK.
15. Rasheed, M., Alabdali, O., Shihab, S., Rashid, A., & Rashid, T. (2021). On the solution of nonlinear equation for photovoltaic cell using new iterative algorithms. *Journal of Physics: Conference Series*,
16. Saxena, D. S. (2017). Advanced Numerical Analysis Techniques: Nonlinear Equation Solving Methods. *Ascent International Journal for Research Analysis*, 2(1), 105.101–105.112. <http://www.ijcms2015.co>
17. Singh, T., & Behera, D. K. (2022). A New Approach of Iterative Methods for Solving Non-linear Equations using Boole's Quadrature. *South East Asian Journal of Mathematics & Mathematical Sciences*, 18(1).
18. Suleiman, S. T. M. (2009). *Solving System of Nonlinear Equations Using Methods in the Halley Class* [The University of Bergen].
19. Thukral, R. (2015). New Third-Order Scroder-Type Method for Finding Zeros of Nonlinear Equation Having Unknnon Multiplicity. *American Journal of Computational and Applied Mathematics*, 5(5), 147-153.
20. Wang, X., & Guo, S. (2025). Dynamic analysis of a family of iterative methods with fifth-order convergence. *Fractal and Fractional*, 9(12), 783.
21. Yang, X.-S. (2020). *Nature-inspired optimization algorithms*. Academic Press.