Available online at sjuoz.uoz.edu.krd

Science Journal of University of Zakho

Vol. 13, No.4 pp589–598October-December,2025

p-ISSN: 2663-628X
e-ISSN: 2663-6298

ENHANCING NONLINEAR EQUATION SOLUTIONS THROUGH THE COMBINATION OF VARIANT NEWTON'S AND HALLEY'S METHODS

Kazhal H. Mohammed Ali[1,*] Bayda Gh. Fathi[1]

[1]Department of Mathematics, College of Science, University of Zakho, Zakho, Kurdistan Region, Iraq

*Corresponding author email: kazhal.mohammed@uoz.edu.krd

Received: 24 Apr 2025 Accepted:22Jun 2025 Published:8 Oct 2025 https://doi.org/10.25271/sjuoz.2025.13.4.1594

ABSTRACT:

This work presents a new iterative method for solving single-variable nonlinear equations. The method achieves ninth-order convergence with just three derivative evaluations per step, offering both accuracy and lower computational cost. Unlike slower bracketing methods, it builds on faster open methods, though these may sometimes fail to converge. By blending ideas from Newton's and Halley's methods, the new approach provides strong performance, as shown by a detailed convergence analysis and MATLAB tests. Compared to existing techniques, it finds solutions in fewer steps and less time, making it especially effective for difficult nonlinear problems

KEYWORDS: Newton's Method, Variant of Newton's Method, Halley's Method, Efficiency Index, Nonlinear Equations.

1. INTRODUCTION

Iterative root-finding algorithms are indispensable across engineering, physics, and applied mathematics, underpinning models from nonlinear structural analysis to parameter estimation in dynamical systems (Soomro *et al.*, 2023; Naseem *et al.*, 2022). Bracketing methods, such as the bisection algorithm, guarantee convergence, but only at a linear rate, making them impractical for high-precision requirements (Goodman *et al*., 2017) . Open methods, such as Newton's method (NM), achieve quadratic convergence but may diverge if the initial estimate is poor or if derivative evaluations are expensive (Kumar *et al*., 2013).Various mathematical models have been developed for solving differential equations, including the Successive Approximation Method (Sabali *et al*., 2021), the Adomian Decomposition Method (Azzo *et al*., 2022), and the Residual Power Series Method (Manaa *et al*., 2021).

Halley's method (HM) mitigates this by incorporating second derivatives to attain cubic convergence, but the extra derivative computation can outweigh its faster convergence in practice(Elhasadi, 2007). To reduce sensitivity to starting guesses while retaining high convergence order, variants such as the Weerakoon–Fernando third-order scheme (Weerakoon *et al.*, 2000) and sixth-order Halley-type modifications (Noor *et al*., 2007) have been proposed; however, each entails trade-offs between per-iteration cost and overall efficiency. Silalahi et *al.* (2017), introduced a method, known as NIH, that combines the Halley method, the Newton method, and the Newton inverse method.

In this paper, we propose the Variant Newton–Halley Method (VNHM), which combines a third-order Newton-type predictor with a Halley-type corrector to achieve ninth-order convergence. Moreover, we prove VNHM's convergence order and compute its efficiency index via a detailed Taylor series analysis. Furthermore, we demonstrate through MATLAB experiments on eight benchmark functions that VNHM consistently reduces the number of iterations and CPU time compared to Newton's method, Halley's method, and the Weerakoon–Fernando variant. Even though VNHM can reach very high accuracy in just a few steps when you can cheaply compute its needed derivatives, it does become overly complex and expensive if those derivatives are hard to get or noisy. Because it relies on calculating both first- and second-order derivatives every time, and its guaranteed success only applies when you start fairly close to the proper solution, it is less well suited to cases where derivative information is costly, unreliable, or when you only need a rough answer. The results were compared with those obtained from the methods in (Silalahi *et al*., 2017; Weerakoon *et al.*, 2000).

The remainder of this paper is organized as follows. Section 2 reviews NM, HM, Weerakoon–Fernando variant, and NIH before introducing VNHM. Section 3 develops the convergence analysis. Section 4 describes the test functions and their known roots. Section 5 presents numerical comparisons of iteration counts, execution times, and accuracy. Section 6 concludes and outlines directions for extending VNHM to complex-root problems.

Iterative Methods:

This section will introduce the fundamental ideas behind the NM, HM, VNM, and NIH. Furthermore, the VNHM will be presented.

* Corresponding author

Newton's Method (NM)

For the nonlinear scalar equation $f(x_i) = 0$, NM is among the most effective root-finding methods (Madhu *et al.*, 2016). The method's quadratic convergence rate makes it likely the most widely used approach for solving nonlinear equations. However, if poor initial assumptions are made, it can occasionally be weakened. However, to use it as a reference point, it needs to be calculated as a function's derivative, which is not always simple or even possible, or it cannot be expressed in terms of an elementary function (McDonough, 2007;Kusni *et al.*, 2016). Newton's method may converge more quickly than any other method, but performance comparison requires taking both convergence speed and cost into account (Azure *et al.*, 2019). The general form of the NM is:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \quad i = 0,1,2,.... \qquad (1)$$

Algorithm (NM) (Tasiu *et al*., 2020)
Given a sufficiently smooth function $f: D \subseteq \mathbb{R} \longrightarrow \mathbb{R}$ with $f'(x_i) \neq 0$ on D.
Input: Initial approximation $(x_0) \in D$, error tolerance ($Tol > 0$), and the maximum number of iterations (N).
Output: An approximation root, x_{i+1}, or a message of failure if the tolerance is not met within N iterations.
Step 1: Set $i = 0$.
Step 2: Repeat until $|f'(x_i)| < Tol$ or the maximum number of iterations is reached:
compute $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$.
Step 3: If $|x_{i+1} - x_i| <$ Tol, then return x_{i+1} as the approximate solution and stop.
Step 4: Set $i = i + 1$ and go to step 2.

A Variant of Newton's Method (VNM)

In 2000, Weerakoon and Fernando showed that the method with third-order convergence is the outcome of deriving NM, which entails an indefinite integral of the function's derivative and an approximate rectangle for the relevant area (Weerakoon *et al*., 2000) . This modification reduces the local truncation error by using a trapezoid rather than a rectangle to approximate this indefinite integral. Iterations can be performed without the need to compute the function's second or higher derivatives, which is the VNM's most significant feature. The general form of the VNM is:

$$y_{i+1} = x_i - \frac{2f(x_i)}{[f'(x_i)+f'(x_{i+1})]}. \quad i = 0,1,2,.... \qquad (2)$$

Here $x_{i+1}$is obtained using the standard Newton iteration.

Algorithm (VNM)

Given a function $f: D \subseteq \mathbb{R} \longrightarrow \mathbb{R}$, assuming $f \in C^1(D)$ With a simple root $v \in D$ of f, so $f(x_i) = 0$, $f'(x_i) \neq 0$.
Input: Initial approximation $x_0 \in D$, error tolerance $Tol > 0$, optional maximum number of iterations N.
Output: Return x_{i+1} as the root approximation or return a 'no convergence' when the tolerance criterion is not met within N Iterations.
Step 1: Set $i = 0$, calculate the first Newton iteration:
$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$.
Step 2: While $i < N$ repeat:
Step 3: Compute the predictor, which was already computed in Step 1:
$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$.
Step 4: Calculate the corrector:
$y_{i+1} = x_i - \frac{2f(x_i)}{[f'(x_i)+f'(x_{i+1})]}$.
Step 5: If $|y_{i+1} - y_i| <$ Tol, then return y_{i+1} and terminate.
Step 6: Set $i = i + 1$and go to step 3.

Halley's Iteration Method (HM)

Halley's method is a third-order root-finding algorithm closely related to Newton's method (NM). Whereas NM uses the tangent-line approximation of f to achieve quadratic convergence, Halley's method incorporates second-derivative information to accelerate convergence to cubic order (Scavo *et al.*, 1995). Given the current iteration x_i, Halley's update is:

$$x_{i+1} = x_i - \frac{2f(x_i)f'(x_i)}{2(f'(x_i))^2 - f(x_i)f''(x_i)}. \qquad i = 0,1,2,.... \qquad (3)$$

Both NM and HM belong to a wider family of explicit iterative schemes that exploit successively higher derivatives to improve convergence order(Yasir Abdul-Hassan, 2016).
Algorithm 2.3. (HM) (Thota *et al.*, 2023)
Given a sufficiently smooth function $f: D \subseteq \mathbb{R} \longrightarrow \mathbb{R}$, assuming $f \in C^2(D)$ and $v \in D$ is a simple root of f, so $f(x_i) = 0$, $f'(x_i) \neq 0$, $f''(x_i) \neq 0$.
Input: An initial guess $x_0 \in D$An error tolerance $Tol > 0$, and a maximum number of iterations N.
Output: An approximation root, x_{i+1}, or a message of failure if no convergence is achieved within N iterations.
Step 1: Set $i = 0$.
Step 2: While $i < N$ do,
Step 3: For a given x_0, calculate Halley's update:
$x_{i+1} = x_i - \frac{2f(x_i)f'(x_i)}{2(f'(x_i))^2 - f(x_i)f''(x_i)}$.
Step 4: If $|x_{i+1} - x_i| <$ Tol, then return x_{i+1} and stop.
Step 5: Set $i = i + 1$and go to step 3.

Combination of the Newton Method, the Newton Inverse Method, and the Halley Method (NIH)

This approach solves nonlinear equations by combining the Newton method, the Newton inverse method, and the Halley method, as introduced by (Silalahi *et al.*, 2017).

Algorithm. (NIH)

Given a sufficiently smooth function $f: D \subseteq \mathbb{R} \longrightarrow \mathbb{R}$, assuming $f \in C^2(D)$.
Input: An initial guess $x_0 \in D$An error tolerance $Tol > 0$, and a maximum number of iterations N.
Output: An approximation root, z_{i+1}, or a message of failure if convergence is not achieved within N iterations.
Step 1: Set $i = 0$.
Step 2: While $i < N$ do,
Step 3: For a given x_0, compute $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$
and $x_i^* = x_i - \frac{f(x_i)}{2}\left(\frac{1}{f'(x_i)} + \frac{1}{f'(x_{i+1})}\right)$.
Step 4: Evaluate $z_{i+1} = x_i^* - \frac{2f(x_i^*)f'(x_i^*)}{2(f'(x_i^*))^2 - f(x_i^*)f''(x_i^*)}$.
Step 5: If $|z_{i+1} - z_i| < Tol$ or the maximum number of iterations is reached, terminate; otherwise, return to Step 4.

Proposed Variant Newton–Halley Method (VNHM)

The Proposed Variant Newton–Halley Method (VNHM) combines the strengths of predictor–corrector techniques with the fast convergence of higher-order iterative schemes. The method is developed through a detailed Taylor series analysis. VNHM begins with a Variant Newton Method (VNM) step to ensure stability, followed by a Halley-type corrector to enhance the convergence rate without sacrificing accuracy. The main objective is to provide a method that is both efficient and reliable, while keeping the computational requirements reasonable.

$$z_{i+1} = y_{i+1} - \frac{2f(y_{i+1})f'(y_{i+1})}{2f'^2(y_{i+1}) - f(y_{i+1})f''(y_{i+1})}, i = 0,1,2, \dots \quad (4)$$

where $y_{i+1} = x_i - \frac{2f(x_i)}{[f'(x_i)+f'(x_{i+1})]}$ and $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$.

In this section, we provide all the essential steps and explanations needed for full understanding and transparency, as is standard for introducing a new algorithm in numerical analysis.

Derivation of the Method:

Suppose you have a function $f: D \subseteq \mathbb{R} \longrightarrow \mathbb{R}$ that's smooth enough, and you want to find a simple root (i.e., $f(x_*) = 0$, and $f'(x_*) \neq 0$). Start with an initial guess x_0 close to the root. The method proceeds in three clear steps:

Step 1: Newton's Predictor.

$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$.

This is the usual Newton step, giving a better estimate for the root.

Step 2: Variant Newton (VNM) Correction

$y_{i+1} = x_i - \frac{2f(x_i)}{[f'(x_i)+f'(x_{i+1})]}$.

Here, you average the derivative at x_n and y_n (an approach inspired by the Weerakoon–Fernando method) to get a more stable, higher-order update, but without needing second derivatives. This step often does a good job of improving the guess, especially if Newton's step was unstable.

Step 3: Halley's High-Order Corrector.

$z_{i+1} = y_{i+1} - \frac{2f(y_{i+1})f'(y_{i+1})}{2f'^2(y_{i+1}) - f(y_{i+1})f''(y_{i+1})}$.

Finally, Halley's formula is used at y_{i+1}. Since y_{i+1} is already a decent approximation, applying Halley's step here delivers even higher accuracy, usually more than what's possible with either Newton or VNM alone.

Algorithm (VNHM)

Given $f: D \subseteq \mathbb{R} \longrightarrow \mathbb{R}$ with f, f', f''continuous and $\nu \in D$ is a simple root of f.

Input: Initial guess $x_0 \in D$, tolerance $0 < Tol < 1$, and a maximum number of iterations N.

Output: An approximation root, z_{i+1}, or a message of failure if no convergence is achieved within N iterations.

Step 1: Set i = 0.

Step 2: For a given x_0, calculate the predictor step, which involves

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)},$$

$$y_{i+1} = x_i - \frac{2f(x_i)}{[f'(x_i)+f'(x_{i+1})]}.$$

Step 3: Evaluate the Halley correction step as follows:

$$z_{i+1} = y_{i+1} - \frac{2f(y_{i+1})f'(y_{i+1})}{2f'^2(y_{i+1}) - f(y_{i+1})f''(y_{i+1})}.$$

Step 4: If $|z_{i+1} - z_i| <$ Tol, then return z_{i+1} as the approximate solution and stop.

Step 5: Set $i = i + 1$. If $i < N$, go to step 2; otherwise, return to the algorithm failed to converge.

Remarks and Limitations:

Stability: As with all open (non-bracketing) methods, VNHM is not magic. If you start too far from the root, the method may fail to converge or may diverge entirely. Applicability: VNHM is most useful when you need high precision and have easy access to both first and second derivatives. If calculating derivatives is expensive or at risk of error, this method may not be ideal.

Convergence Analysis:

In this section, we present the convergence analysis of the new three-step iterative method (4) for solving nonlinear equations

Theorem: Let ν be a simple zero of a function that is continuously differentiable up to order eight on an open interval. If the initial guess x_0 is chosen sufficiently close to ν, then the three-step VNHM iteration in Algorithm 4 converges to ν with ninth-order accuracy.

Proof: Since $f(\nu) = 0$ and $\epsilon_i = x_i - \nu$, Taylor's theorem around the simple root ν gives

$$f(x_i) = (x_i - \nu)f'(\nu) + \frac{(x_i-\nu)^2}{2!}f^{(2)}(\nu) + \frac{(x_i-\nu)^3}{3!}f^{(3)}(\nu) + \frac{(x_i-\nu)^4}{4!}f^{(4)}(\nu) + \cdots. \quad (5)$$

This expansion will form the basis for our error-recurrence analysis. By taking the first derivative of (5) with respect to x_i, we obtain

$$f'(x_i) = f'(\nu) + (x_i - \nu)f''(\nu) + \frac{(x_i-\nu)^2}{2!}f^{(3)}(\nu) + \frac{(x_i-\nu)^3}{3!}f^{(4)}(\nu) + \cdots. \quad (6)$$

Substituting $\epsilon_i = x_i - \nu$ in (5) and (6), we have

$$f(x_i) = \epsilon_i f'(\nu) + \frac{(\epsilon_i)^2}{2!}f''(\nu) + \frac{(\epsilon_i)^3}{3!}f^{(3)}(\nu) + \frac{(\epsilon_i)^4}{4!}f^{(4)}(\nu) + O(\epsilon_i^5), \quad (7)$$

where $O(\epsilon_i^5)$ represents all terms of order 5 and higher.

$$f'(x_i) = f'(\nu) + \epsilon_i f''(\nu) + + \frac{(\epsilon_i)^2}{2!}f^{(3)}(\nu) + \frac{(\epsilon_i)^3}{3!}f^{(4)}(\nu) + O(\epsilon_i^4). \quad (8)$$

From (7) and (8), we get

$$\frac{f(x_i)}{f'(x_i)} = \frac{\epsilon_i f'(\nu) + \frac{(\epsilon_i)^2}{2!}f''(\nu) + O(\epsilon_i^3)}{f'(\nu) + \epsilon_i f''(\nu) + O(\epsilon_i^2)}. \quad (9)$$

When (9) is substituted into (1) and used $x_{i+1} = \epsilon_{i+1}^x + \nu$, the result is

$$\epsilon_{i+1}^x = \epsilon_i - \frac{\epsilon_i f'(\nu) + \frac{(\epsilon_i)^2}{2!}f''(\nu) + O(\epsilon_i^3)}{f'(\nu) + \epsilon_i f''(\nu) + O(\epsilon_i^2)}. \quad (10)$$

For small ϵ_i, approximation

$$\epsilon_{i+1}^{\mathrm{x}} = \epsilon_i - \frac{\epsilon_i + \frac{(\epsilon_i)^2}{2!}\frac{\mathrm{f}''(\nu)}{\mathrm{f}'(\nu)}}{1-\epsilon_i\frac{\mathrm{f}''(\nu)}{\mathrm{f}'(\nu)}}. \tag{11}$$

Using the binomial expansion (i.e. $\frac{1}{1+\mathrm{u}} = 1 - u + u^2 - u^3 + \cdots$), we get

$$\epsilon_{i+1}^{\mathrm{x}} = \epsilon_i - (\epsilon_i + \frac{(\epsilon_i)^2}{2!}\frac{\mathrm{f}''(\nu)}{\mathrm{f}'(\nu)})\left(1 - \epsilon_i\frac{\mathrm{f}''(\nu)}{\mathrm{f}'(\nu)} + \left(\epsilon_i\frac{\mathrm{f}''(\nu)}{\mathrm{f}'(\nu)}\right)^2 - \cdots\right). \tag{12}$$

So, the second-order binomial expansion gives

$$\epsilon_{i+1}^{\mathrm{x}} = \epsilon_i - (\epsilon_i + \frac{\epsilon_i^2}{2!}\frac{\mathrm{f}''(\nu)}{\mathrm{f}'(\nu)} - \epsilon_i^2\frac{\mathrm{f}''(\nu)}{\mathrm{f}'(\nu)} + O(\epsilon_i^3)). \tag{13}$$

Hence

$$\epsilon_{i+1}^{\mathrm{x}} = \frac{\mathrm{f}''(\nu)}{2\mathrm{f}'(\nu)}\epsilon_i^2 + O(\epsilon_i^3) = C_2\,\epsilon_i^2 + O(\epsilon_i^3), \tag{14}$$

where c_2 is defined as the constant $\frac{\mathrm{f}''(\nu)}{2\mathrm{f}'(\nu)}$. This shows that Newton's iteration, x_{i+1}, converges with order 2.

Now, we want to determine how $y_{i+1} = \mathrm{x}_i - \frac{2\mathrm{f}(\mathrm{x}_i)}{\mathrm{f}'(\mathrm{x}_i)+\mathrm{f}'(\mathrm{x}_{i+1})}$ converges to ν.

Taylor expansion of $f(\mathrm{x}_i)$ around ν is

$$\mathrm{f}(\mathrm{x}_i) = (\mathrm{x}_i - \nu)\mathrm{f}'(\nu) + \frac{(\mathrm{x}_i-\nu)^2}{2!}\mathrm{f}^{(2)}(\nu) + \frac{(\mathrm{x}_i-\nu)^3}{3!}\mathrm{f}^{(3)}(\nu) + O(\mathrm{x}_i - \nu)^4. \tag{15}$$

Since $\epsilon_i = \mathrm{x}_i - \nu$,

$$\mathrm{f}(\mathrm{x}_i) = \epsilon_i\mathrm{f}'(\nu) + \frac{(\epsilon_i)^2}{2!}\mathrm{f}''(\nu) + \frac{(\epsilon_i)^3}{3!}\mathrm{f}^{(3)}(\nu) + O(\epsilon_i^4). \tag{16}$$

$$\mathrm{f}'(\mathrm{x}_i) = \mathrm{f}'(\nu) + (\mathrm{x}_i - \nu)\mathrm{f}''(\nu) + \frac{(\mathrm{x}_i-\nu)^2}{2!}\mathrm{f}^{(3)}(\nu) + O(\mathrm{x}_i - \nu)^3. \tag{17}$$

Similarly, for the iteration $i+1$ we have

$$\mathrm{f}'(\mathrm{x}_{i+1}) = \mathrm{f}'(\nu) + (\mathrm{x}_{i+1} - \nu)\mathrm{f}''(\nu) + \frac{(\mathrm{x}_i-\nu)^2}{2!}\mathrm{f}^{(3)}(\nu) + O(\mathrm{x}_i - \nu)^3. \tag{18}$$

Substituting $\epsilon_i = \mathrm{x}_i - \nu$ and $\mathrm{x}_{i+1} - \nu = C_2\epsilon_i^2$, we have

$$\mathrm{f}'(\mathrm{x}_{i+1}) = \mathrm{f}'(\nu) + C_2\epsilon_i{}^2\mathrm{f}''(\nu) + \frac{(C_2\epsilon_i{}^2)^2}{2}\mathrm{f}^{(3)}(\nu) + O(\epsilon_i^4). \tag{19}$$

$$\mathrm{f}'(\mathrm{x}_i) + \mathrm{f}'(\mathrm{x}_{i+1}) = \left(\mathrm{f}'(\nu) + \epsilon_i\mathrm{f}''(\nu) + \frac{\epsilon_i{}^2}{2}\mathrm{f}^{(3)}(\nu) + \frac{4\epsilon_i^3}{4*6}\mathrm{f}^{(4)}(\nu) + O(\epsilon_i^5)\right) + \left(\mathrm{f}'(\nu) + C_2\epsilon_i{}^2\mathrm{f}''(\nu) + \frac{C_2^2\epsilon_i{}^4}{2}\mathrm{f}^{(3)}(\nu) + \frac{C_2{}^3\epsilon_i{}^6}{6}\mathrm{f}^{(4)}(\nu) + O(\epsilon_i^5)\right), = 2\mathrm{f}'(\nu) + \epsilon_i\mathrm{f}^{(3)}(\nu) + \left(\frac{1}{2}\mathrm{f}''(\nu) + C_2\mathrm{f}''(\nu)\right)\epsilon_i{}^2 + \frac{1}{6}\mathrm{f}^{(4)}(\nu)\epsilon_i^3 + O(\epsilon_i^4). \tag{20}$$

By substituting (18) and (20) into y_{i+1}in (2), we obtain

$$y_{i+1} = \mathrm{x}_i - \frac{2\left(\epsilon_i\mathrm{f}'(\nu)+\frac{\epsilon_i^2}{2}\mathrm{f}''(\nu)+\frac{\epsilon_i^3}{6}\mathrm{f}^{(3)}(\nu)+O(\epsilon_i^4)\right)}{2\mathrm{f}'(\nu)+\epsilon_i\mathrm{f}''(\nu)+\left(\frac{1}{2}\mathrm{f}^{(3)}(\nu)+C_2\mathrm{f}''(\nu)\right)\epsilon_i{}^2+\frac{1}{6}\mathrm{f}^{(4)}(\nu)\epsilon_i^3+\mathrm{O}(\epsilon_i{}^4)}. \tag{21}$$

Subtracting ν from both sides of (21) and let $\epsilon_{i+1}^{y} = y_{i+1} - \nu$, we obtain

$$\epsilon_{i+1}^{y} = \epsilon_i - \left[2\epsilon_i\mathrm{f}'(\nu) + \epsilon_i^2\mathrm{f}''(\nu) + \frac{\epsilon_i^3}{3}\mathrm{f}^{(3)}(\nu) + O(\epsilon_i^4)\right]\left[1/\left(2\mathrm{f}'(\nu) + \epsilon_i\mathrm{f}''(\nu) + \left(\frac{1}{2}\mathrm{f}^{(3)}(\nu) + C_2\mathrm{f}''(\nu)\right)\epsilon_i{}^2 + \mathrm{O}(\epsilon_i{}^3)\right)\right]. \tag{22}$$

Since

$$1/\left(2\mathrm{f}'(\nu) + \epsilon_i\mathrm{f}''(\nu) + \left(\frac{1}{2}\mathrm{f}^{(3)}(\nu) + C_2\mathrm{f}''(\nu)\right)\epsilon_i{}^2\right) = \frac{1}{2\mathrm{f}'(\nu)}\left[1/\left(1 + \epsilon_i\frac{\mathrm{f}''(\nu)}{2\mathrm{f}'(\nu)} + \left(\frac{\mathrm{f}^{(3)}(\nu)}{4\mathrm{f}'(\nu)} + \frac{C_2}{2\mathrm{f}'(\nu)}\mathrm{f}''(\nu)\right)\epsilon_i{}^2\right)\right], \tag{23}$$

The binomial expansion yields

$$\epsilon_{i+1}^{y} = \frac{1}{2\mathrm{f}'(\nu)}\left[1 - \frac{\epsilon_i\mathrm{f}''(\nu)}{2\mathrm{f}'(\nu)} - \left(\frac{\mathrm{f}^{(3)}(\nu)}{4\mathrm{f}'(\nu)} + \frac{C_2}{2\mathrm{f}'(\nu)}\mathrm{f}''(\nu)\right)\epsilon_i{}^2 + \left(\frac{\epsilon_i\mathrm{f}''(\nu)}{2\mathrm{f}'(\nu)} + \left(\frac{\mathrm{f}^{(3)}(\nu)}{4\mathrm{f}'(\nu)} + \frac{C_2}{2\mathrm{f}'(\nu)}\mathrm{f}''(\nu)\right)\epsilon_i{}^2\right)^2 + O(\epsilon_i^5)\right] = \epsilon_i - \left(2\epsilon_i\mathrm{f}'(\nu) + \epsilon_i^2\mathrm{f}''(\nu) + \frac{\epsilon_i^3}{3}\mathrm{f}^{(3)}(\nu) + o(\epsilon_i^4)\right)\left(\frac{1}{2\mathrm{f}'(\nu)}\left(1 - \frac{\epsilon_i\mathrm{f}''(\nu)}{2\mathrm{f}'(\nu)} - \frac{\mathrm{f}^{(3)}(\nu)}{4\mathrm{f}'(\nu)} - \frac{C_2\mathrm{f}''(\nu)}{2\mathrm{f}'(\nu)}\epsilon_i^2 + \frac{\epsilon_i^2\mathrm{f}''(\nu)^2}{4\mathrm{f}'(\nu)^2} + \frac{\epsilon_i\mathrm{f}''(\nu)}{\mathrm{f}'(\nu)}\left(\frac{\mathrm{f}^{(3)}(\nu)}{4\mathrm{f}'(\nu)} + \frac{C_2\mathrm{f}''(\nu)}{2\mathrm{f}'(\nu)}\right)\epsilon_i^2 + O(\epsilon_i^4)\right)\right). \tag{24}$$

Substituting $C_n = \frac{\mathrm{f}^{(n)}(\nu)}{n!\mathrm{f}'(\nu)}$ into (24), we yield

$$\epsilon_{i+1}^{y} = \epsilon_i - \left(2\epsilon_i\mathrm{f}'(\nu) + \epsilon_i^2\mathrm{f}''(\nu) + \frac{\epsilon_i^3}{3}\mathrm{f}^{(3)}(\nu) + O(\epsilon_i^4)\right)\left(\frac{1}{2\mathrm{f}'(\nu)}\left(1 - \epsilon_iC_2 - \epsilon_i^2\frac{3C_3}{2} - C_2^2\epsilon_i^2 + \epsilon_i^2C_2^2 + \epsilon_i^3C_2^2 + 2\epsilon_i^3C_2^4 + O(\epsilon_i^4)\right)\right). \tag{25}$$

$$= \epsilon_i - \left(2\epsilon_i\mathrm{f}'(\nu) + \epsilon_i^2\mathrm{f}''(\nu) + \frac{\epsilon_i^3}{3}\mathrm{f}^{(3)}(\nu) + o(\epsilon_i^4)\right) * \left(\frac{1}{2\mathrm{f}'(\nu)} - \frac{C_2}{2\mathrm{f}'(\nu)}\epsilon_i + \left(\frac{3C_3}{2} + C_2^2 - C_2^2\right)\frac{\epsilon_i^2}{2\mathrm{f}'(\nu)} + \frac{(C_2^2+2C_2^4)}{2\mathrm{f}'(\nu)}\epsilon_i^3 + O(\epsilon_i^4)\right). \tag{26}$$

After some algebra, we get

$$\epsilon_{i+1}^{y} = \left(C_2^2 + \frac{C_3}{2}\right)\epsilon_i^3 + O(\epsilon_i^4). \tag{27}$$

This demonstrates clearly that y_{i+1} in (2) has third-order convergence

Now, expanding $\mathrm{f}(y_{i+1})$ around ν gives

$$\mathrm{f}(y_{i+1}) = (y_{i+1} - \nu)\mathrm{f}'(\nu) + \frac{(y_{i+1}-\nu)^2}{2}\mathrm{f}''(\nu) + \frac{(y_{i+1}-\nu)^3}{3!}\mathrm{f}'''(\nu) + \frac{(y_{i+1}-\nu)^4}{4!}\mathrm{f}^{(4)}(\nu) + \frac{(y_{i+1}-\nu)^5}{5!}\mathrm{f}^{(5)}(\nu) + O((y_{i+1} - \nu)^6). \tag{28}$$

$$\mathfrak{f}'(y_{i+1}) = \mathfrak{f}'(\nu) + (y_{i+1}-\nu)\mathfrak{f}''(\nu) + \frac{(y_{i+1}-\nu)^2}{2}\mathfrak{f}'''(\nu) + \frac{(y_{i+1}-\nu)^3}{3!}\mathfrak{f}^{(4)}(\nu) + O((y_{i+1}-\nu)^4). \quad (29)$$

$$\mathfrak{f}''(y_{i+1}) = \mathfrak{f}''(\nu) + (y_{i+1}-\nu)\mathfrak{f}'''(\nu) + \frac{(y_{i+1}-\nu)^2}{2}\mathfrak{f}^{(4)}(\nu) + \frac{(y_{i+1}-\nu)^3}{3!}\mathfrak{f}^{(5)}(\nu) + \frac{(y_{i+1}-\nu)^4}{4!}\mathfrak{f}^{(6)}(\nu) + O((y_{i+1}-\nu)^5). \quad (30)$$

Replacing $y_{i+1}-\nu$ with $K\epsilon_i^3$ where $K = C_2^2 + \frac{C_3}{2}$ in (29)- (31), gives

$$\mathfrak{f}(y_{i+1}) = K\epsilon_i^3\mathfrak{f}'(\nu) + \frac{K^2\epsilon_i^6}{2}\mathfrak{f}''(\nu) + \frac{K^3\epsilon_i^9}{3!}\mathfrak{f}^{(3)}(\nu) + O(\epsilon_i^{12}), \quad (31)$$

$$\mathfrak{f}'(y_{i+1}) = \mathfrak{f}'(\nu) + K\epsilon_i^3\,\mathfrak{f}''(\nu) + \frac{(K\epsilon_i^3)^2}{2}\mathfrak{f}^{(3)}(\nu) + \frac{(K\epsilon_i^3)^3}{3!}\mathfrak{f}^{(4)}(\nu) + O\left((\epsilon_i^3)^4\right), \quad (32)$$

$$\mathfrak{f}''(y_{i+1}) = \mathfrak{f}''(\nu) + K\epsilon_i^3\mathfrak{f}^{(3)}(\nu) + \frac{(K\epsilon_i^3)^2}{2}\mathfrak{f}^{(4)}(\nu) + \frac{(K\epsilon_i^3)^3}{3!}\mathfrak{f}^{(5)}(\nu) + \frac{(K\epsilon_i^3)^4}{4!}\mathfrak{f}^{(6)}(\nu) + O\left((K\epsilon_i^3)^5\right). \quad (33)$$

$$2\mathfrak{f}(y_{i+1})\mathfrak{f}'(y_{i+1}) = \left(2K\epsilon_i^3\mathfrak{f}'(\nu) + K^2\epsilon_i^6\mathfrak{f}''(\nu) + \frac{K^3\epsilon_i^9}{3}\mathfrak{f}^{(3)}(\nu)\right)\left(\mathfrak{f}'(\nu) + K\epsilon_i^3\,\mathfrak{f}''(\nu) + \frac{K^2\epsilon_i^6}{2}\mathfrak{f}^{(3)}(\nu) + \frac{K^3\epsilon_i^9}{6}\mathfrak{f}^{(4)}(\nu)\right) =$$

$$2K\epsilon_i^3\mathfrak{f}'^2(\nu) + 3K^2\epsilon_i^6\mathfrak{f}'(\nu)\mathfrak{f}''(\nu) + \left(K^3\mathfrak{f}''^2(\nu) + \frac{4}{3}K^3\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu)\right)\epsilon_i^9 + O\left((\epsilon_i^3)^4\right). \quad (34)$$

$$2\mathfrak{f}'^2(y_{i+1}) = \left(2\mathfrak{f}'(\nu) + 2K^3\epsilon_i^3\mathfrak{f}''(\nu) + K^2\epsilon_i^6\mathfrak{f}^{(3)}(\nu) + \frac{K^3\epsilon_i^9}{3}\mathfrak{f}^{(4)}(\nu)\right)\left(\mathfrak{f}'(\nu) + K\epsilon_i^3\,\mathfrak{f}''(\nu) + \frac{K^2\epsilon_i^6}{2}\mathfrak{f}^{(3)}(\nu) + \frac{K^3\epsilon_i^9}{6}\mathfrak{f}^{(4)}(\nu)\right) =$$

$$2\mathfrak{f}'^2 4K\epsilon_i^3\mathfrak{f}'(\nu)\mathfrak{f}''(\nu) + \left(2K^2\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu) + 2K^2\mathfrak{f}''^2(\nu)\right)\epsilon_i^6 + \left(2K^3\mathfrak{f}''(\nu)\mathfrak{f}^{(3)}(\nu) + \frac{2}{3}K^3\mathfrak{f}'(\nu)\mathfrak{f}^{(4)}(\nu)\right)\epsilon_i^9 + O\left((\epsilon_i^3)^4\right). \quad (35)$$

$$\mathfrak{f}(y_{i+1})\mathfrak{f}''(y_{i+1}) = \left[K\epsilon_i{}^3\mathfrak{f}'(\nu) + \frac{K^2}{2}\epsilon_i{}^6\mathfrak{f}''(\nu) + \frac{K^3}{6}\epsilon_i{}^9\mathfrak{f}'''(\nu)\right]\begin{bmatrix}\mathfrak{f}''(\nu) + K\epsilon_i{}^3\mathfrak{f}^{(3)}(\nu)\\ +\frac{K^2}{2}\epsilon_i{}^6\mathfrak{f}^{(4)}(\nu)\end{bmatrix} = K\epsilon_i{}^3\mathfrak{f}'(\nu)\mathfrak{f}''(\nu) + \left(\frac{K^2}{2}\mathfrak{f}''^2(\nu) +$$

$$K^2\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu)\right)\epsilon_i{}^6 + \left(\left(\frac{K^3}{6} + \frac{K^3}{2}\right)\mathfrak{f}''(\nu)\mathfrak{f}^{(3)}(\nu) + \frac{K^3}{2}\mathfrak{f}'(\nu)\mathfrak{f}^{(4)}(\nu)\right)\epsilon_i{}^9. \quad (36)$$

$$2\mathfrak{f}'^2(y_{i+1}) - \mathfrak{f}(y_{i+1})\mathfrak{f}''(y_{i+1}) = \left[2\mathfrak{f}'^2(\nu) + 4K\mathfrak{f}'(\nu)\mathfrak{f}''(\nu)\epsilon_i{}^3 + \left(2K^2\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu) + 2K^3\mathfrak{f}''^2(\nu)\right)\epsilon_i{}^6 + \left(2K^3\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu) + \right.\right.$$

$$\left.\left.\frac{2}{3}K^3\mathfrak{f}'(\nu)\mathfrak{f}^{(4)}(\nu)\right)\epsilon_i{}^9 + O(\epsilon_i^{12})\right] - [K\mathfrak{f}'(\nu)\mathfrak{f}''(\nu)\epsilon_i{}^3 + \left(\frac{K^2}{2}\mathfrak{f}^{(\prime\prime)2} + K^2\mathfrak{f}'(\nu)\mathfrak{f}^3(\nu)\right)\epsilon_i{}^6 + \left(\left(\frac{K^3}{6} + \frac{K^3}{2}\right)\mathfrak{f}''(\nu)\mathfrak{f}^{(3)}(\nu) +$$

$$\frac{K^3}{2}\mathfrak{f}'(\nu)\mathfrak{f}^{(4)}(\nu)\right)\epsilon_i{}^9 + O(\epsilon_i^{12}) = 2\mathfrak{f}'^2(\nu) + 3K\mathfrak{f}'(\nu)\mathfrak{f}''(\nu)\epsilon_i{}^3 + (K^2\mathfrak{f}'(\nu)\mathfrak{f}'''(\nu) + \frac{3}{2}K^2\mathfrak{f}''^2(\nu))\epsilon_i{}^6 + (\frac{11K^3}{6}\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu) -$$

$$\frac{K^3}{2}\mathfrak{f}''(\nu)\mathfrak{f}^{(3)}(\nu) + \frac{K^3}{6}\mathfrak{f}''(\nu)\,\mathfrak{f}^{(4)}(\nu))\epsilon_i{}^9 + O(\epsilon_i^{12}). \quad (37)$$

Substituting (34)-(37) into (4), we obtain

$$z_{i+1} = y_{i+1} - \left[2K\mathfrak{f}^2(\nu)\epsilon_i^3 + 3K^2\mathfrak{f}'(\nu)\mathfrak{f}''(\nu)\epsilon_i^6 + \left(K^3\mathfrak{f}''^2(\nu) + \frac{4}{3}K^3\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu)\right)\epsilon_i^9 + O(\epsilon_i^{12})\right]\left[1/\left(2\mathfrak{f}'^2(\nu) + 3K\mathfrak{f}'(\nu)\mathfrak{f}''(\nu)\epsilon_i^3 + \right.\right.$$

$$\left.\left.\left(K^2\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu) + \frac{3}{2}K^2\mathfrak{f}''^2(\nu)\right)\epsilon_i^6 + O(\epsilon_i^{12})\right)\right], \quad (38)$$

$$= y_{i+1} - \left[2K\mathfrak{f}^2(\nu)\epsilon_i^3 + 3K^2\mathfrak{f}'(\nu)\mathfrak{f}''(\nu)\epsilon_i^6 + \left(K^2\mathfrak{f}''^2(\nu) + \frac{4}{3}K^2\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu)\right)\epsilon_i^9 + \right.$$

$$\left.O(\epsilon_i^{12})\right]\left[\frac{1}{2\mathfrak{f}'^2(\nu)}\left(1/\left(1 + \frac{(3K\mathfrak{f}'(\nu)\,\mathfrak{f}''(\nu))\,\epsilon_i^3}{2\mathfrak{f}'^2(\nu)} + \frac{\left(K^2\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu) + \frac{3}{2}K^2\mathfrak{f}''^2(\nu)\right)\epsilon_i^6}{2\mathfrak{f}'^2(\nu)}\right)\right) + O(\epsilon_i^9)\right]. \quad (39)$$

Using the binomial expansion gives

$$z_{i+1} = y_{i+1} - \left[2K\mathfrak{f}'^2(\nu)\epsilon_i^3 + 3K^2\mathfrak{f}'(\nu)\mathfrak{f}''(\nu)\epsilon_i^6 + \left(K^3\mathfrak{f}''^2(\nu) + \frac{4}{3}K^3\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu)\right)\epsilon_i^9 + O(\epsilon_i^{12})\right]$$

$$\left[\frac{1}{2\mathfrak{f}'^2(\nu)}\left(1 - \frac{(3K\mathfrak{f}'(\nu)\,\mathfrak{f}''(\nu))\,\epsilon_i^3}{2\mathfrak{f}'^2(\nu)} - \frac{\left(K^2\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu) + \frac{3}{2}K^2\mathfrak{f}''^2(\nu)\right)\epsilon_i^6}{2\mathfrak{f}'^2(\nu)} + \left(\frac{(3K\mathfrak{f}'(\nu)\,\mathfrak{f}''(\nu))\,\epsilon_i^3}{2\mathfrak{f}'^2(\nu)}\right)^2\epsilon_i^6\right)\right] + 2\left(\frac{(3K\mathfrak{f}'(\nu)\,\mathfrak{f}''(\nu))\,\epsilon_i^3}{2\mathfrak{f}'^2(\nu)}\right)\left(\frac{\left(K^2\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu) + \frac{3}{2}K^2\mathfrak{f}''^2(\nu)\right)\epsilon_i^6}{2\mathfrak{f}'^2(\nu)}\right) -$$

$$\left(\frac{(3K\mathfrak{f}'(\nu)\,\mathfrak{f}''(\nu))\,\epsilon_i^3}{2\mathfrak{f}'^2(\nu)}\right)^3 + O(\epsilon_i^{12})] \quad (40)$$

$$= y_{i+1} - \left(2K\mathfrak{f}'^2(\nu)\epsilon_i^3 + 3K^2\mathfrak{f}'(\nu)\mathfrak{f}''(\nu)\epsilon_i^6 + \left(K^3\mathfrak{f}''^2(\nu) + \frac{4}{3}K^3\mathfrak{f}'(\nu)\mathfrak{f}^{(3)}(\nu)\right)\epsilon_i^9 + O(\epsilon_i^{12})\right)\left(\frac{1}{2\mathfrak{f}'^2(\nu)}\left(1 - \frac{(3K\,\mathfrak{f}''(\nu))\,\epsilon_i^3}{2\mathfrak{f}'(\nu)} + \right.\right.$$

$$\left.\left.\frac{\left(3K^2\,\mathfrak{f}''^2(\nu)\right)\epsilon_i^6}{2\mathfrak{f}'^2(\nu)} - \frac{\left(K^2\,\mathfrak{f}^{(3)}(\nu)\right)\epsilon_i^6}{2\mathfrak{f}'(\nu)} + \frac{\left(3K^3\mathfrak{f}''(\nu)\,\mathfrak{f}^{(3)}(\nu)\right)\epsilon_i^9}{2\mathfrak{f}'^2(\nu)} - \frac{\left(9K^3\,\mathfrak{f}''^3(\nu)\right)\epsilon_i^9}{8\mathfrak{f}'^3(\nu)}\right)\right) + O(\epsilon_i^{12}). \quad (41)$$

Subtracting ν from both sides and setting ϵ_{i+1}^z, we obtain

$$\epsilon_{i+1}^z = y_{i+1} - \nu - K\epsilon_i^3 + \left(-\frac{K^3\mathfrak{f}^{(3)}(\nu)}{6\mathfrak{f}'(\nu)} + \frac{K^3\,\mathfrak{f}''^2(\nu)}{4\mathfrak{f}'^2(\nu)}\right)\epsilon_i^9 + O(\epsilon_i^{12}). \quad (42)$$

Since $y_{i+1} - \nu = K\epsilon_i^3$, it follows that

$$\epsilon_{i+1}^z = \left(\frac{K^3\,\mathfrak{f}''^2(\nu)}{4\mathfrak{f}'^2(\nu)} - \frac{K^3\mathfrak{f}^{(3)}(\nu)}{6\mathfrak{f}'(\nu)}\right)\epsilon_i^9 + O(\epsilon_i^{12}). \quad (43)$$

Repeating the same algebraic expansion shows that every $O(\epsilon_i^6)$ The contribution drops out, so the first nonzero term is simply$M\epsilon_i^9$. Thus, we have

$$\epsilon_{i+1}^z = M\epsilon_i^9 + O(\epsilon_i^{12}), \tag{44}$$

Where

$$M = K^3\left(\frac{f''^2(\nu)}{4f'^2(\nu)} - \frac{f^{(3)}(\nu)}{6f'(\nu)}\right). \tag{45}$$

Recall $K = c_2^2 + \frac{C_3}{2}$ and using some algebra, (45) becomes

$$M = \left(c_2^2 + \frac{C_3}{2}\right)^3\left(\frac{f''^2(\nu)}{4f'^2(\nu)} - \frac{f^{(3)}(\nu)}{6f'(\nu)}\right), \tag{46}$$

where $c_n = \frac{1}{n!}\frac{f^{(n)}(\nu)}{f'(\nu)}$), n=2,3,4,…

Therefore,

$$\epsilon_{i+1}^z = \left[(c_2^2 - C_3)\left(c_2^2 + \frac{C_3}{2}\right)^3\right]\epsilon_i^9 + O(\epsilon_i^{12}),$$

Which shows that the order of convergence of our new proposed method (VNHM) defined in (4) is nine. This completes the proof.

Testing Functions:

We used the same test functions as (Weerakoon *et al.*, 2000) and display the approximate zero ν found up to the 14 decimal place.

$f_1(x) = x^3 + 4x^2 - 10$, $\nu = 1.36523001341448$.
$f_2(x) = \sin^2(x) - x^2 + 1$, $\nu = 1.40449164821621$.
$f_3(x) = x^2 - e^x - 3x + 2$, $\nu = 0.257530285439771$.
$f_4(x) = \cos(x) - x$, $\nu = 0.739085133214758$.
$f_5(x) = (x-1)^3 - 1$, $\nu = 2$.
$f_6(x) = x^3 - 10$, $\nu = 2.15443469003367$.
$f_7(x) = x\exp(x^2) - \sin^2(x) + 3\cos(x) + 5$, $\nu = -1.20764782713013$.
$f_8(x) = x^2\sin^2(x) + \exp[x^2\cos(x)\sin(x)] - 28$, $\nu = 4.62210416355283$.
$f_9(x) = \exp(x^2 + 7x - 30) - 1$, $\nu = 3$.

Numerical Results:

This study was conducted using the following hardware and software: a personal computer with the specifics listed below: Intel(R) Core (TM) i7-10870H CPU @ 2.20GHz 2.21 GHz, RAM 16 GB. The following software is employed: MATLAB software and the Windows 11 Ultimate 64-bit operating system. Now, solve a few nonlinear equations using the new algorithm discovered in this paper. NM, HM, VNM, NIH, and the approach presented in this paper are also compared. Compares the number of iterations, execution time, and accuracy of the proposed method with NM, HM, VNM, and NIH at the set precision. The tolerance is Tol=10^{-14}.

Number of Iterations:

Table 1 presents the number of iterations required by each method. The results show that VNHM requires the fewest total iterations across all test cases, with only 63 iterations in total, fewer than NM, HM, VNM, or NIH. It is also important to note that the number of iterations depends on both the chosen tolerance and the initial starting point; when the initial guess is closer to the actual root, fewer iterations are generally needed.

Table 1: Comparison of the number of iterations of each method.

Function	x_0	Number of iterations for each method				
		NM	HM	VNM	NIH	VNHM
f_1	-0.5	132	74	7	7	4
	1	6	4	4	3	3
	2	6	4	4	3	3
f_2	-0.3	54	53	7	11	4
	1	7	6	5	3	3
	3	7	7	4	3	3
f_3	2	6	5	5	3	3
	3	7	5	5	3	4
f_4	1	5	4	3	3	2
	1.7	5	4	4	3	3
	-0.3	6	5	4	3	3
f_5	3.5	8	5	6	3	4
	2.5	7	5	5	3	3

f_6	1.5	7	4	5	3	3
f_7	-2	9	5	6	4	4
f_8	5	10	7	6	4	5
f_9	3.5	13	7	9	5	5
	3.25	9	6	7	4	4
Total		304	210	96	71	63

Execution Time:

Table 2 displays each method's execution time. Based on the computational results, the VNHM method has the smallest running time overall

Table 2: Execution time for each method across test functions.

Function	x_0	Execution time (s).				
		NM	HM	VNM	NIH	VNHM
f_1	-0.5	0.012174	0.006913	0.004003	0.005636	0.003387
	1	0.004597	0.001927	0.002460	0.003301	0.001864
	2	0.003197	0.003522	0.002494	0.001724	0.002106
f_2	-0.3	0.003729	0.007959	0.003999	0.010023	0.002390
	1	0.005388	0.004280	0.004652	0.012623	0.002643
	3	0.003272	0.006936	0.002767	0.003364	0.002234
f_3	2	0.002747	0.003331	0.003039	0.003406	0.002665
	3	0.005315	0.002674	0.003690	0.002218	0.002550
f_4	1	0.002221	0.002055	0.003536	0.003405	0.002452
	1.7	0.002353	0.002799	0.003601	0.002561	0.002967
f_5	-0.3	0.002911	0.003183	0.003014	0.002931	0.002227
	3.5	0.004415	0.003599	0.003602	0.002140	0.003596
	2.5	0.003077	0.002941	0.002604	0.002100	0.002663
f_6	1.5	0.003304	0.002422	0.002262	0.003735	0.002177
f_7	-2	0.004583	0.006338	0.004292	0.004053	0.004003
f_8	5	0.005394	0.007633	0.009630	0.004031	0.007925
f_9	3.5	0.002997	0.004680	0.003668	0.003145	0.004239
	3.25	0.003444	0.004051	0.005279	0.003543	0.005348
Total		0.075118	0.077243	0.068592	0.073939	0.057436

Accuracy:

Table 3 presents the computed root values obtained by each method for the selected nonlinear equations. The results show that all five methods generally converge to the expected root values across most test functions. For the benchmark function f_8, however, the root values found by NM, HM, and VNM (all approximately 3.437471) differ notably from the value obtained by both VNHM and NIH (approximately 4.622104), which is closer to the true solution reported by Weerakoon and Fernando. This comparison highlights that, while all methods perform similarly on standard cases, both VNHM and NIH demonstrate superior accuracy when solving more challenging equations. These findings confirm the robustness of VNHM, particularly for difficult problems where traditional methods may fail to reach the correct root.

Table 3: Accuracy of computed root values for each method.

Function	x_0	Root value.				
		NM	HM	VNM	NIH	VNHM
f_1	-0.5	1.36523001341410	1.36523001341410	1.36523001341410	1.36523001341410	1.36523001341410
	1	1.36523001341410	1.36523001341410	1.36523001341410	1.36523001341410	1.36523001341410
	2	1.36523001341410	1.36523001341410	1.36523001341410	1.36523001341410	1.36523001341410
f_2	-0.3	1.36523001341410	1.36523001341410	1.36523001341410	1.36523001341410	1.36523001341410
	1	1.40449164821534	1.40449164821534	1.40449164821534	1.40449164821534	1.40449164821534
	3	1.40449164821534	1.40449164821534	1.40449164821534	1.40449164821534	1.40449164821534

f_3	2	0.25753028543986	0.25753028543986	0.25753028543986	0.25753028543986	0.25753028543986
	3	0.25753028543986	0.25753028543986	0.25753028543986	0.25753028543986	0.25753028543986
f_4	1	0.73908513321516	0.73908513321516	0.73908513321516	0.73908513321516	0.73908513321516
	1.7	0.73908513321516	0.73908513321516	0.73908513321516	0.73908513321516	0.73908513321516
	-0.3	0.73908513321516	0.73908513321516	0.73908513321516	0.73908513321516	0.73908513321516
f_5	3.5	2	2	2	2	2
2	2.5	2	2	2	2	2
f_6	1.5	2.15443469003188	2.15443469003188	2.15443469003188	2.15443469003188	2.15443469003188
f_7	-2	-1.20764782713092	-1.20764782713092	-1.20764782713092	-1.20764782713092	-1.20764782713092
f_8	5	3.43747174342177	3.43747174342177	3.43747174342177	4.62210416355284	4.62210416355284
f_9	3.5	3	3	3	3	3
	3.25	3	3	3	3	3

Comparisons of Efficiency Index:

The term "efficiency index" compares the performance of different iterative methods. It depends upon the order of convergence and the number of functional evaluations of the iterative process. If " r " denotes the order of convergence and " N_f " Denote the number of functional evaluations of an iterative method, then the efficiency index E.I is defined as:

$E.I = r^{\frac{1}{N_f}}$.

On this basis, NM(Nazeer *et al.*, 2016) has an efficiency of $2^{\frac{1}{2}} \approx 1.4142$. HM (Noor *et al.*, 2007) has an order of convergence of three, and the number of functional evaluations required for this method is three, so its efficiency $3^{\frac{1}{3}} \approx 1.4422$. The VNM has an efficiency of $3^{\frac{1}{3}} \approx 1.4422$. The VNHM needs one evaluation of the function's first and second derivatives. Thus, this method has three functional evaluations. i.e.

$N_f = 6$.

Also, in the earlier section, it was proven that the order of convergence of the VNHM is nine. i.e.

r =9.

Thus, the efficiency index of VNHM is:

$$E.I = 9^{\frac{1}{6}} \approx 1.4422.$$

Table 4: Efficiency indices of the compared iterative methods.

Method	Number of function and Derivative evaluations	Efficiency index
NM, quadratic	2	$2^{\frac{1}{2}} \approx 1.41421$
HM 3rd order	3	$3^{\frac{1}{3}} \approx 1.44225$
VNM 3rd order	3	$3^{\frac{1}{3}} \approx 1.44225$
VNHM 9th order	6	$9^{\frac{1}{6}} \approx 1.44225$

Real-World Applications:

VNHM's ability to find roots with very high precision in just a few steps makes it highly suitable for real-time control problems in robotics, where fast and accurate solutions are needed (Martin, 2019). The method is also advantageous in tuning nonlinear stiffness curves in structural analysis (Engelberger, 2014) and for solving transcendental equations in optical design, such as determining resonant frequencies in photonic crystals (Reddy, 2003).In each of these cases, the combination of rapid convergence and moderate derivative evaluation cost enables VNHM to outperform traditional Newton- or Halley-based methods**.**

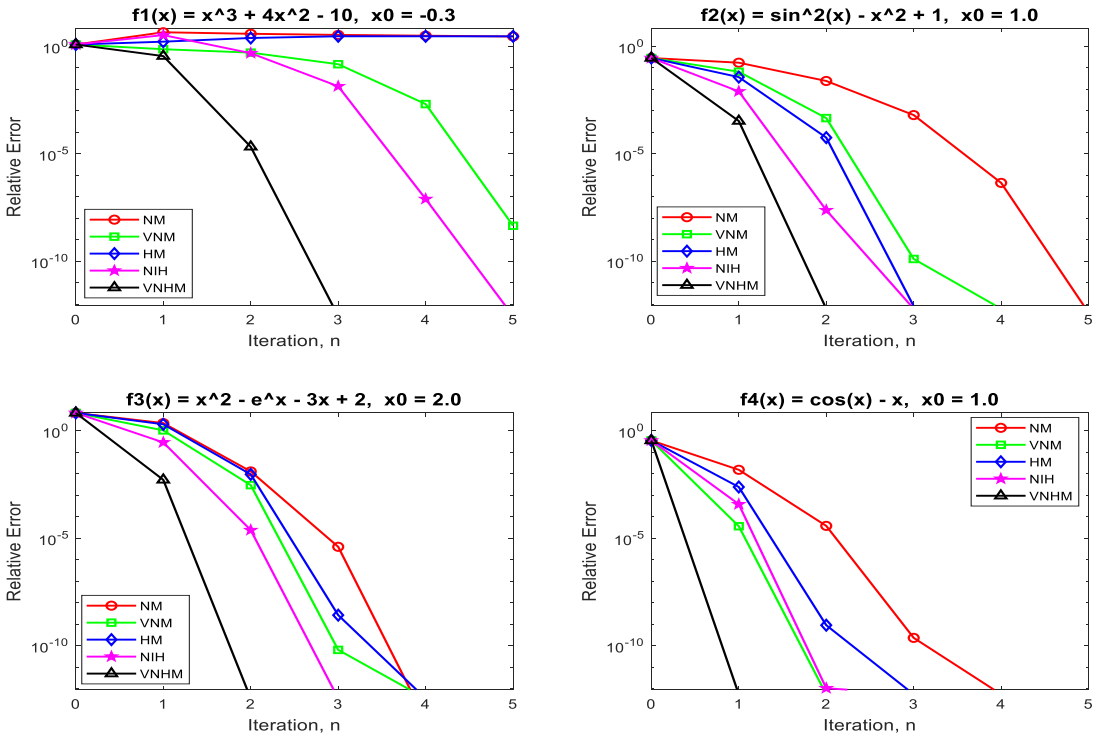

Figure 1: Reduction of the relative error graph from iteration 1 to iteration 5.

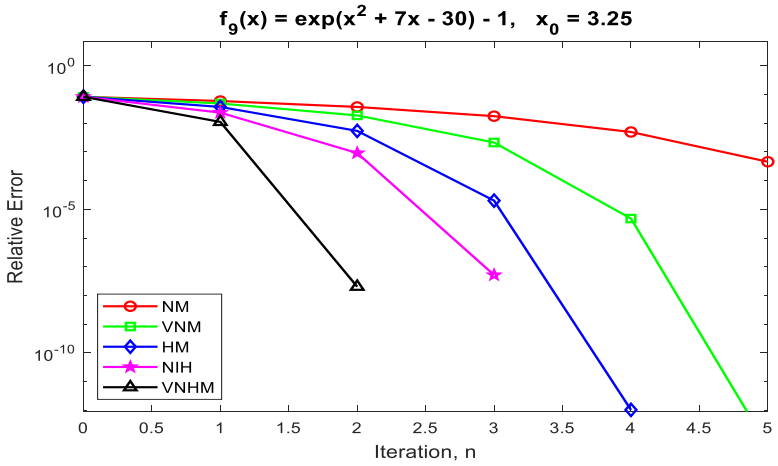

Figure 2: Reduction of relative error graph of iteration 1 to iteration 5.

Figures 1 and 2 illustrate the convergence rates of the five iterative methods for solving nonlinear equations. As shown, the VNHM consistently achieves high accuracy in the fewest steps across all test problems. In practical terms, this demonstrates that an effective combination of iterative techniques can significantly reduce computation time and enhance robustness for a wide range of equations**.**

CONCLUSIONS

In this paper, we combined the Halley method (HM) and the Variant Newton method (VNM) to construct the Variant Newton–Halley Method (VNHM) for solving nonlinear equations. This method is used to find solutions to nonlinear equations. We have shown that the proposed method has a ninth-order convergence. By using some test examples, the performance and efficiency of the VNHM have been analyzed. Tables 1, 2, 3, and 4 show the best performance of the proposed iterative algorithms as compared to other well-known existing iterative algorithms in terms of accuracy, speed, number of iterations, efficiency index, and computational order of convergence. Also, relative error reduces the fastest among other methods, as shown in Figures 1 and 2. The VNHM is effective for real-valued nonlinear equations but is currently not applicable to complex roots. Future work will focus on extending the method to complex roots.

Acknowledgment:

The authors would like to express their sincere gratitude to the Science Journal of the University of Zakho for its continued support and for providing a platform to share this research. We also thank the anonymous reviewers for their valuable comments and suggestions, which helped improve the quality of this paper.

Ethical Approval:

This study does not involve human participants or animals, and therefore, ethical approval was not required.

Declarations:

Authors' contribution: K.H.M.: Methodology, Writing original draft, Formal analysis, Validation, Software. B.Gh.F.: Resources, Acquisition, Formal Analysis, Investigation, Software Review.

Funding: This work received no external funding.

Availability of data and materials: Data sharing does not apply to this article, as no data sets were generated or analyzed during the current study.

Competing interests: The authors declare that they have neither financial nor conflict of interest.

REFERENCES

Azure, I., Aloliga, G., & Doabil, L. (2019). Comparative Study of Numerical Methods for Solving Non-linear Equations Using Manual Computation. *Mathematics Letters*, *5*(4), 41. DOI: 10.11648/j.ml.20190504.11

Azzo, S. M., & Manaa, S. A. (2022). Sumudu-Decomposition Method to Solve Generalized Hirota-Satsuma Coupled Kdv System. *Science Journal of University of Zakho*, *10*(2), 43–47. DOI: 10.25271/sjuoz.2022.10.2.879

Elhasadi, O. I. (2007). Newton's and Halley's methods for real polynomials [Master's thesis, University of Guelph]. University of 448 Guelph Atrium. https://atrium.lib.uoguelph.ca/xmlui/handle/10214/1002

Engelberger, J. (2014). Springer Handbook of Robotics Robotics & Automation *: Books for Robotics*. *June*.

Goodman, R. O. Y. H., & Obel, J. K. W. R. (2017). High-Order Bisection Method For Computing Invariant Manifolds Of Two-Dimensional Map*s*. *21*(7), 2017–2042. DOI: 10.1142/S0218127411029604

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. DOI: 10.1016/j.joems.2013.03.001

Kusni, A., & Shamsul, A. (2016). Numerical Study of Some Iterative Methods for Solving Nonlinear Equations. *5*(2), 1–10. Retrieved from www.ijesi.org

Madhu, K., & Jayaraman, J. (2016). Higher order methods for nonlinear equations and their basins of attraction. *Mathematics*, *4*(2), 1–20. DOI: 10.3390/math4020022

Manaa, S. A., Easif, F. H., & Murad, J. J. (2021). Residual Power Series Method for Solving Klein-Gordon Schrödinger Equation. *Science Journal of University of Zakho*, *9*(2), 123–127. DOI: 10.25271/sjuoz.2021.9.2.810

Martin, O. J. F. (2019). Molding the flow of light with metasurfaces. *2019 URSI Asia-Pacific Radio Science Conference, AP-RASC 2019*, 32–43. DOI: 10.23919/URSIAP-RASC.2019.8738549

McDonough, J. M. (2007). Lectures in Basic Computational Numerical Analysis. *Journal of Biomechanics*, *39*, 163. Retrieved from http://www.engr.uky.edu/~acfd/egr537-lctrs.pdf

Naseem, A., Rehman, M. A., & Abdeljawad, T. (2022). A Novel Root-Finding Algorithm with Engineering Applications and its Dynamics via Computer Technology. *IEEE Access*, *10*(1), 19677–19684. DOI: 10.1109/ACCESS.2022.3150775

Nazeer, W., & Tanveer, M. (2016). *Modified Golbabai and Javidi ' S Method (Mgjm) for Solving Modified Golbabai and Javidi ' S Method (Mgjm) for Solving Nonlinear Functions With Convergence of Order Six*. *January 2015*.

Noor, M. A., Khan, W. A., & Hussain, A. (2007). A new modified Halley method without second derivatives for nonlinear equation. *Applied Mathematics and Computation*, *189*(2), 1268–1273. DOI: 10.1016/j.amc.2006.12.011

Reddy, J. N. (2003). Mechanics of Laminated Composite Plates and Shells. *Mechanics of Laminated Composite Plates and Shells*. DOI: 10.1201/b12409

Sabali, A. J., Manaa, S. A., & Easif, F. H. (2021). New Successive Approximation Methods for Solving Strongly Nonlinear Jaulent-Miodek Equations. *Science Journal of University of Zakho*, *9*(4), 193–197. DOI: 10.25271/sjuoz.2021.9.4.869

Scavo, T. R., & Thoo, J. B. (1995). On the Geometry of Halley's Method. *The American Mathematical Monthly*, *102*(5), 417–426. DOI: 10.1080/00029890.1995.12004594

Soomro, S. A., Shaikh, A. A., Qureshi, S., & Ali, B. (2023). A Modified Hybrid Method For Solving Non-Linear Equations With Computational Efficiency. *VFAST Transactions on Mathematics*, *11*(2), 126–137. DOI: 10.21015/vtm.v11i2.1620

Silalahi, B. P., Laila, R., & Sitanggang, I. S. (2017). A combination method for solving nonlinear equations. *IOP Conference Series: Materials Science and Engineering*, *166*(1), 12011.

Tasiu, A. R., Abbas, A., Alhassan, M. N., & Umar, A. N. (2020). Comparative Study on Some Methods of Handling Nonlinear Equations. *Anale. Seria Informatică*, *XVIII*(2), 2–5.

Thota, S., Gemechu, T., & Ayoade, A. A. (2023). on New Hybrid Root-Finding Algorithms for Solving Transcendental Equations Using Exponential and Halley'S Methods. *Ural Mathematical Journal*, *9*(1), 176–186. DOI: 10.15826/umj.2023.1.016

Weerakoon, S., & Fernando, T. (2000). A variant of Newton's method with accelerated third-order convergence. *Applied Mathematics Letters*, *13*(8), 87–93.

Yasir Abdul-Hassan, N. (2016). New Predictor-Corrector Iterative Methods with Twelfth-Order Convergence for Solving Nonlinear Equations. *American Journal of Applied Mathematics*, *4*(4), 175. DOI: 10.11648/j.ajam.20160404.12