VFAST Transactions on Mathematics
http://vfast.org/journals/index.php/VTM@ 2024 ISSN(e): 2309-0022, ISSN(p): 2411-6343
Volume 12, Number 1, January-June, 2024 pp: 164-175

# An Improved Blended Numerical Root-Solver for Nonlinear Equations

**Asad Ali Chandio**[1], **Asif Ali Shaikh**[1,2], **Sania Qureshi**[1*,2,3], **Abdul Rehman Soomroo**[4]

[1*]Department of Basic Sciences and Related Studies, Mehran University, of Engineering & Technology, Jamshoro – 76062, Pakistan; [2]Department of Mathematics, Near East University, 99138, Mersin, Turkey; [3]Department of Computer Science and Mathematics, Lebanese American, University, Beirut P.O. Box 13-5053, Lebanon; [4]Department of Mathematics, Sukkur IBA University, Pakistan

***Keywords***: *Nonlinear equations; Order of convergence; Newton Raphson method; Efficiency index.*
***Subject Classification***: *Numerical analysis, Modeling for nonlinear equations.*

***Journal Info***:
*Submitted: February 27, 2024*
*Accepted: March 12, 2024*
*Published: April 20, 2024*

**Abstract** This study presents a novel three-step iterative approach for solving nonlinear equations in the domains of science and engineering. It represents a notable change from traditional methods like Halley by eliminating the need for second derivatives. The suggested method exhibits a sixth order of convergence and only requires five function evaluations, showcasing its efficiency with an index of roughly 1.430969. The suggested method effectively solves nonlinear problems involving equations with algebraic and transcendental terms. Comparative analysis against existing root-solving algorithms demonstrates their superior performance. The results not only confirm the strength and effectiveness of the three-step iterative approach but also highlight its potential for wide-ranging use in many scientific and technical situations.

***Correspondence Author Email Address:**
sania.qureshi@faculty.muet.edu.pk

## 1 Introduction

Non-linear equations are extremely important almost in every field of science and engineering. Exact solutions to nonlinear equations are often difficult, making numerical approaches the preferred approach. Being higher-order accurate and cost-efficient is crucial for a numerical approach, achieved by minimizing the number of evaluations and iterations required. The study of efficient techniques for solving non-linear equations is carried out in many studies; see for example some of the recently published research papers

[7, 15, 19–21]. Nonlinear equations frequently arise through the application of mathematical methods, and finding their solution is a crucial objective in numerical analysis. Several mathematical models are designed in terms of nonlinear equations. Root-finding algorithms are essential in the fields of science and engineering because they enable the identification of solutions to equations that are challenging or impossible to solve analytically [25]. These methods are employed to address a diverse array of problems, including determining the inherent frequencies of a mechanical system in engineering, identifying the sites of equilibrium in chemical reactions [3] , optimizing functions in operational research [8] , and modeling physical phenomena in physics [11, 14]. Nonlinear equations are commonly found in real-world problems, and as such, they play a crucial role in computational mathematics [24]. Root-finding methods are crucial tools in the progress of research and engineering [23]. They enable scientists and engineers to efficiently determine the solutions to equations, which in turn helps them comprehend and forecast the behavior of intricate systems, optimize processes, and create new technologies [26]. A nonlinear equation in one dimension can be written as:

$$f(x) = 0, \tag{1}$$

where $f : I \subset \mathbb{R} \to \mathbb{R}$ with $I$ being a neighborhood of $\alpha$. It may be noted that $\alpha$ shows the exact root for the equation in (1).

Several numerical techniques exist for finding the approximate solution of $f(x)$ =0. Among existing techniques, the Newton-Raphson (NR) method is a highly efficient technique that exhibits quadratic convergence and requires just two function evaluations per iteration. Nevertheless, the NR technique has a drawback in that it can encounter difficulties when the successive approximations are close to the singular points of the equation $f(x) = 0$. Nonetheless, this second-order classical NR method has gained significant interest among existing nonlinear numerical techniques due to its quadratic convergence and simplicity. According to the Kung-Traub conjecture [12], an optimal iterative approach has an order of convergence ($p$) as $2^{d-1} = p$, with "$d$" being the number of evaluations per iteration and $p$ is the convergence order. Based on this, the NR method is likewise an optimal technique.

The present research investigation aims to suggest a modification to the NR technique and its existing variants. The solutions that require a higher level of accuracy are also of practical significance. Given that the [12] argument is merely speculative, many researchers have focused on advancing the development of a variety of approaches, regardless of achieving the desired results [4, 13, 16]. The concept of the efficiency index ($E$) is also crucial in evaluating the performance of numerical methods. The efficiency index is determined by considering the number of function evaluations per iteration ($d$) and the rate of convergence ($p$) of a method. To achieve reliable modifications in conventional methods, scientists have developed strategies such as enhancing the demand for integration, minimizing the number of function evaluations, decreasing the requirement of resources, and reducing the occurence of higher-derivative evaluations with each iteration.

Taylor's expansion, employed in the NR strategy [12], the Homotopy methodology [18], the variational iterative technique [17], the Adomian decomposition [22], and the quadrature rules [28] are among the methods utilized to discover several nonlinear solvers. Two iterative methods with the orders $p_1$ and $p_2$ can be blended to derive a method having order $p_1 \times p_2$. Although this process includes extra function evaluations throughout each iteration, the resulting combination is always guaranteed to be faster than lower-order alternatives. Building upon this methodology, we introduce a novel sixth-order convergent method by combining the second-order convergent classical NR method with the third-order technique described in [9]. Compared to other approaches, the selected procedure suggests a strategy that saves both time and computation.

The main objective of this study is to introduce a novel and effective multi-step approach, namely a three-step methodology, for addressing approximate the solution of nonlinear problems. The proposed procedure starts with the Newton step, which is a technique that many scholars use including [1, 4, 5, 9, 10, 13, 24] and [2]. Furthermore, the suggested three-step method is proved to have sixth-order convergence by utilizing only five evaluations. The core element of the suggested technique is its ability to minimize CPU usage compared to other commonly employed strategies. The proposed technique is anticipated to surpass, and occasionally even equal, previous numerical techniques in terms of convergence rate, but for a shorter CPU time usage.

The present paper is designed as follows: Section 2 shows a brief overview of the existing numerical methods to solve $f(x) = 0$ whereas Section 3 derives the proposed numerical method in detail and its subsection explain the order of convergence of the proposed method which is proved to be six using Taylor's series approach. The performance of the proposed method is analyzed in comparison to other existing well-known methods in section 4 with the help of some nonlinear problems. At the end, the present paper is concluded with some possible future directions in Section 5.

# 2 Exiting Methods

Some existing iterative methods have been discussed in this section. The order of convergence, number of function evaluations, and efficiency indexes are also discussed here. When it comes to a root-solver then the very first method is the Newton-Raphson (NR2) method whose algorithm is shown below:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \tag{2}$$

where $i = 1, 2, 3, \ldots$. The NR method given in (2) is a one-step method with second-order convergence and its efficiency index is $2^{\frac{1}{2}} \approx 1.4142$. Recently in [1], some authors proposed a three-step sixth-order iterative technique that is a time-efficient and convergent non-linear technique. The efficiency index of the method is 1.43096, where the method contains five evaluations of functions. This technique is abbreviated as AS6.

$$\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)} \\ z_i &= y_i - \frac{f(y_i)}{f'(y_i)} \\ x_{i+1} &= y_i - \frac{f(y_i) + f(z_i)}{f'(y_i)} \qquad i = 0, 1, 2\ldots \end{aligned} \tag{3}$$

Some authors in [2] proposed a three-step iterative technique with the sixth order of convergence having six function evaluations per iteration. The efficiency index of this technique is about 1.348006. This method is denoted as SA6.

$$\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)}, \\ z_i &= y_i - \frac{f(y_i)}{f'(y_i)}, \\ x_{i+1} &= z_i - \frac{2f(z_i)}{3f'(z_i) - f'(y_i)}. \quad i = 0, 1, 2... \end{aligned} \tag{4}$$

The research work in [27] introduced a new fifth-order derivative-based iterative method, which is a four-step non-linear technique. This method has an efficiency index of 1.3797. The method is denoted by the symbol W5.

$$\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)}, \\ z_i &= y_i - \frac{f(y_i)}{f'(x_i)}, \\ w_i &= z_i - \frac{f(z_i)}{f'(x_i)}, \\ x_{i+1} &= w_i - \frac{f(w_i)}{f'(x_i)} \quad i = 0, 1, 2... \end{aligned} \tag{5}$$

# 3 Proposed Method

Getting motivation from [2, 4, 6, 9, 10, 24, 28] , we initiate a new algorithm using the Newton-Raphson method. Hence, a two-step iterative technique we consider here with third-order convergence [9] is given below:

$$\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)}, \\ x_{i+1} &= x_i - \frac{2f'(y_i)}{3f'(y_i) - f'(x_i)} \frac{f(x_i)}{f'(x_i)} \quad i = 0, 1, 2... \end{aligned} \tag{6}$$

The above two-step third-order iterative method is blended with the classical one-step second-order NR method to obtain the three-step sixth-order iterative method as shown below:

$$\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)}, \\ z_i &= y_i - \frac{f(y_i)}{f'(y_i)}, \\ x_{i+1} &= y_i - \frac{2f'(z_i)}{3f'(z_i) - f'(y_i)} \frac{f(y_i)}{f'(y_i)} \end{aligned} \tag{7}$$

The proposed method (7) has the sixth order of convergence and is denoted as Proposed6. The leading term $\phi$ of the error equation is $\frac{-3f''(\beta)^5 + f'(\beta)f''(\beta)^3 f^{(3)}(\beta)}{96f'(\beta)^5}$, the order of convergence ($p$) is 6, the efficiency index ($\rho = p^{\frac{1}{d}}$) is about 1.43, and the function evaluations ($\kappa$) are 5 per iteration.

## 3.1 Order of convergence

**Theorem 1.** Assume that $\beta$ be the root of a differentiable function $f : \mathbb{R} \subset \mathbb{R} \to \mathbb{R}$ for an open interval $I$. Then the three-step blended root-solver given in (7) exhibits sixth-order convergence, and the resulting error term is as follows:

$$e_{i+1} = \frac{-3f''(\beta)^5 + f'(\beta)f''(\beta)^3 f^{(3)}(\beta)}{96 f'(\beta)^5} e_i^6 + \mathcal{O}\left(e_i^7\right), \tag{8}$$

where $e_i = x_i$-$\beta$.

**Proof.** Let $\beta$ be the root of $f(x_i)$, $x_i$ be *ith* approximation to the root by the proposed method given in (7) and $e_i = x_i - \beta$ be the error after *ith* iteration. Using Taylor's expansion for $f(x_i)$ about $\beta$, we have

$$\begin{aligned} f(x_i) &= f(\beta) + e_i f'(\beta) + \frac{1}{2}e_i^2 f''(\beta) + \frac{1}{6}e_i^3 f^{(3)}(\beta) + \frac{1}{24}e_i^4 f^{(4)}(\beta) + \frac{1}{120}e_i^5 f^{(5)}(\beta) \\ &+ \frac{1}{720}e_i^6 f^{(6)}(\beta) + \frac{e_i^7 f^{(7)}(\beta)}{5040} + \mathcal{O}\left(e_i^8\right). \end{aligned} \tag{9}$$

Using Taylor's expansion for $\frac{1}{f'(x_i)}$ about $\beta$ , we have

$$\begin{aligned} \frac{1}{f'(x_i)} &= \frac{1}{f'(\beta)} - \frac{e_n f''(\beta)}{f'(\beta)^2} + e_n^2\left(\frac{f''(\beta)^2}{f'(\beta)^3} - \frac{f^{(3)}(\beta)}{2f'(\beta)^2}\right) + e_n^3\left(-\frac{f^{(4)}(\beta)}{6f'(\beta)^2} - \frac{f''(\beta)^3}{f'(\beta)^4} + \frac{f^{(3)}(\beta)f''(\beta)}{f'(\beta)^3}\right) \\ &+ e_n^4\left(-\frac{f^{(5)}(\beta)}{24f'(\beta)^2} + \frac{f^{(3)}(\beta)^2}{4f'(\beta)^3} + \frac{f''(\beta)^4}{f'(\beta)^5} + \frac{f^{(4)}(\beta)f''(\beta)}{3f'(\beta)^3} - \frac{3f^{(3)}(\beta)f''(\beta)^2}{2f'(\beta)^4}\right) \\ &+ e_n^5\left(-\frac{f^{(6)}(\beta)}{120f'(\beta)^2} - \frac{f''(\beta)^5}{f'(\beta)^6} + \frac{f^{(5)}(\beta)f''(\beta)}{12f'(\beta)^3} + \frac{f^{(3)}(\beta)f^{(4)}(\beta)}{6f'(\beta)^3} - \frac{f^{(4)}(\beta)f''(\beta)^2}{2f'(\beta)^4}\right. \\ &\left.+ \frac{2f^{(3)}(\beta)f''(\beta)^3}{f'(\beta)^5} - \frac{3f^{(3)}(\beta)^2 f''(\beta)}{4f'(\beta)^4}\right) + \mathcal{O}(e_i^6). \end{aligned} \tag{10}$$

Multiplying (9) and (10) and putting the result in the first step of the proposed method given in (7), we get

$$\sigma_i = x_i - \frac{f(x_i)}{f'(x_i)} = \frac{\sigma_i^2 f''(\beta)}{2f'(\beta)} + \frac{\sigma_i^3 f''(\beta)^2}{2f'(\beta)^2} + \frac{\sigma_i^4 f''(\beta)^3}{2f'(\beta)^3} - \frac{\sigma_i^5 f''(\beta)^4}{2f'(\beta)^5} + \frac{\sigma_i^6 f''(\beta)^5}{2f'(\beta)^5} + \mathcal{O}\left(\sigma_i^7\right), \tag{11}$$

where $\sigma_i = y_i - \beta$. Using Taylor's expansion for $f(y_i)$ about $\beta$, we have

$$\begin{aligned} f(y_i) &= f(\beta) + \sigma_i f'(\beta) + \frac{1}{2}\sigma_i^2 f''(\beta) + \frac{1}{6}\sigma_i^3 f^{(3)}(\beta) + \frac{1}{24}\sigma_i^4 f^{(4)}(\beta)\frac{1}{120}\sigma_i^5 f^{(5)}(\beta) \\ &+ \frac{1}{720}\sigma_i^6 f^{(6)}(\beta) + \frac{1}{5040}\sigma_i^7 f^{(6)}(\beta) + \mathcal{O}\left(\sigma_i^8\right). \end{aligned} \tag{12}$$

Using Taylor's expansion for $\frac{1}{f'(y_i)}$ about $\beta$, we have

$$\begin{aligned}\frac{1}{f'(y_i)} &= \frac{1}{f'(\beta)} - \frac{\sigma_n f''(\beta)}{f'(\beta)^2} + \sigma_n^2\left(\frac{f''(\beta)^2}{f'(\beta)^3} - \frac{f^{(3)}(\beta)}{2f'(\beta)^2}\right) + \sigma_n^3\left(-\frac{f^{(4)}(\beta)}{6f'(\beta)^2} - \frac{f''(\beta)^3}{f'(\beta)^4} + \frac{f^{(3)}(\beta)f''(\beta)}{f'(\beta)^3}\right)\\ &+ \sigma_n^4\left(-\frac{f^{(5)}(\beta)}{24f'(\beta)^2} + \frac{f^{(3)}(\beta)^2}{4f'(\beta)^3} + \frac{f''(\beta)^4}{f'(\beta)^5} + \frac{f^{(4)}(\beta)f''(\beta)}{3f'(\beta)^3} - \frac{3f^{(3)}(\beta)f''(\beta)^2}{2f'(\beta)^4}\right)\\ &+ \sigma_n^5\left(-\frac{f^{(6)}(\beta)}{120f'(\beta)^2} - \frac{f''(\beta)^5}{f'(\beta)^6} + \frac{f^{(5)}(\beta)f''(\beta)}{12f'(\beta)^3} + \frac{f^{(3)}(\beta)f^{(4)}(\beta)}{6f'(\beta)^3} - \frac{f^{(4)}(\beta)f''(\beta)^2}{2f'(\beta)^4}\right.\\ &\left.+ \frac{2f^{(3)}(\beta)f''(\beta)^3}{f'(\beta)^5} - \frac{3f^{(3)}(\beta)^2 f''(\beta)}{4f'(\beta)^4}\right) + \mathcal{O}(\sigma_i^6).\end{aligned} \tag{13}$$

Multiplying (12) and (13) and putting the result in the second step of the proposed method given in (7), we get

$$\begin{aligned}\epsilon_i = y_i - \frac{f(y_i)}{f'(y_i)} &= \frac{\sigma_i^2 f''(\beta)}{2f'(\beta)} + \frac{\sigma_i^3 f''(\beta)^2}{2f'(\beta)^2} + \frac{\sigma_i^4 f''(\beta)^3}{2f'(\beta)^3} - \frac{\sigma_i^5 f''(\beta)^4}{2f'(\beta)^5} + \frac{\sigma_i^6 f''(\beta)^5}{2f'(\beta)^5}\\ &+ \mathcal{O}\left(\sigma_i^7\right),\end{aligned} \tag{14}$$

where $\epsilon_i = z_i - \beta$. Using Taylor's expansion for $f(z_i)$ about $\beta$, we have

$$\begin{aligned}f(z_i) &= \epsilon_i f'(\beta) + \frac{1}{2}\epsilon_i^2 f''(\beta) + \frac{1}{6}\epsilon_i^3 f^{(3)}(\beta) + \frac{1}{24}\epsilon_i^4 f^{(4)}(\beta) + \frac{1}{120}\epsilon_i^5 f^{(5)}(\beta)\\ &+ \frac{1}{720}E_i^6 f^{(6)}(\beta) + \frac{\epsilon_i^7 f^{(7)}(\beta)}{5040} + \mathcal{O}\left(\epsilon_i^8\right).\end{aligned} \tag{15}$$

Differentiating (15), we have

$$\begin{aligned}f'(z_i) &= f'(\beta) + \epsilon_i f''(\beta) + \frac{1}{2}\epsilon_i^2 f^{(3)}(\beta) + \frac{1}{6}\epsilon_i^3 f^{(4)}(\beta) + \frac{1}{24}\epsilon_i^4 f^{(5)}(\beta)\\ &+ \frac{1}{120}\epsilon_i^6 f^{(5)}(\beta) + \frac{\epsilon_i^6 f^{(7)}(\beta)}{720} + \mathcal{O}\left(\epsilon_i^8\right).\end{aligned} \tag{16}$$

By putting all values in $y_i - \frac{2f'(z_i)}{3f'(z_i) - f'(y_i)}\frac{f(y_i)}{f'(y_i)}$, we finally get

$$e_{i+1} = \frac{-3f''(\beta)^5 + f'(\beta)f''(\beta)^3 f^{(3)}(\beta)}{96f'(\beta)^5}e_i^6 + \mathcal{O}\left(e_i^7\right). \tag{17}$$

The above error equation shows that the three-step proposed method given in (7) has a sixth order of convergence ($p = 6$).

# 4 Numerical Simulations

This section has addressed the topic of numerical simulations of scalar non-linear equations with the sixth-order numerical method proposed in (7). When comparing, we will consider parameters such as the number of iterations ($k$), the computational cost (COC = $kd$), the absolute error ($\varepsilon$) at the last iteration, the absolute functional value ($\chi$) at the last iteration, the approximate root, and the CPU time in seconds to run the entire code for producing the required simulation results. We used different iterative methods, which we talked about in Section 2, and compared them to the proposed sixth-order blended root-solver given in equation (7). The numerical computations are performed using MAPLE 2022 on an Intel (R) Core (TM) i7 HP laptop with 24GB of RAM, operating at a processing speed of 1.3 GHz. In regard to the numerical results, we have established a maximum precision threshold of 4,000 digits. The maximum iteration limit for achieving the desired answer is set at 50. The numerical simulations employ the following criterion to determine when to stop:

$$\varepsilon = |x_{i+1} - x_i| \leq 10^{-200}. \tag{18}$$

Given below are five nonlinear equations of scalar type whose exact solutions are given to a desired accuracy. We simulate these equations with the proposed three-step iterative method (7) including four other iterative root-solvers.

**Problem 1.** $f_1(x) = x^2 + \sin\left(\frac{x}{5}\right) - \frac{1}{4}, \quad x^* = 0.4099920179891371.$

**Table 1.** Numerical simulations for $f_1(x)$ in Problem 1 at initial guess $x_0 = 3.05$.

| Method | $k$ | COC | $\varepsilon$ | $\chi$ | root | CPU time |
|---|---|---|---|---|---|---|
| NR2 | 12 | 24 | 2.32e-292 | 5.35e-584 | 4.09e-01 | 1.41e-01 |
| AS6 | 6 | 30 | 5.80e-915 | 1.00e-4000 | 4.10e-01 | 1.40e-01 |
| SA6 | 6 | 36 | 6.68e-1109 | 0.00e+00 | 4.10e-01 | 1.56e-01 |
| W5 | 6 | 30 | 1.32e-251 | 3.00e-1254 | 4.10e-01 | 1.56e-01 |
| Proposed6 | 6 | 30 | 1.32e-1107 | 0.00e+00 | 4.10e-01 | 1.41e-01 |

Table 1 presents a comparative analysis of root-finding methods applied to $f_1(x)$. Notably, the method (Proposed6), developed in the present research study, stands out for its efficient convergence, achieving a small absolute error ($\varepsilon$) of $1.32e - 1107$ and a functional value ($\chi$) of zero in 6 iterations. Although the Newton-Raphson method (NR2) exhibits high precision with a lower COC of 24, the proposed method demonstrates competitive performance with comparable computational cost and CPU time. Other methods, including AS6 and SA6, also show effective convergence but with slightly higher absolute errors and functional values. Overall, the proposed method showcases promising efficiency and accuracy, highlighting its potential for practical applications in numerical analysis.

**Problem 2.** $f_2(x) = x^5 + x - 1000, \quad x^* \approx 3.98.$

**Table 2.** Numerical simulations for $f_2(x)$ in Problem 2 at initial guess $x_0 = 4.5$.

| Method | $k$ | COC | $\varepsilon$ | $\chi$ | root | CPU time |
|---|---|---|---|---|---|---|
| NR2 | 10 | 20 | 1.34e-327 | 1.12e-651 | 3.98e+00 | 1.60e-02 |
| AS6 | 5 | 25 | 1.62e-756 | 3.00e-3997 | 3.98e+00 | 1.60e-02 |
| SA6 | 5 | 30 | 3.08e-827 | 3.00e-3997 | 3.98e+00 | 1.60e-02 |
| W5 | 5 | 25 | 8.08e-292 | 2.20e-1453 | 3.98e+00 | 1.50e-02 |
| Proposed6 | 5 | 25 | 3.23e-859 | 3.00e-3997 | 3.98e+00 | 1.6000e-02 |

Table 2 presents a comparative analysis of several numerical methods applied to solve Problem 2, which involves finding the root of the function $f_2(x) = x^5 + x - 1000$. The methods evaluated include NR2, AS6, SA6, W5, and the proposed method (Proposed6). Each method is evaluated based on several criteria: the number of iterations ($k$), the computational cost (COC), the absolute error ($\varepsilon$), the functional value ($\chi$), the approximate root, and the CPU time (in seconds).

From the data, it is evident that all methods converge to an approximate root of $x^* \approx 3.98$. However, there are notable differences in the performance of the methods across other metrics. The proposed method (Proposed6) achieves competitive results in terms of the number of iterations, computational cost, and CPU time compared to the other methods. Additionally, it exhibits extremely low absolute error and functional value, indicating high accuracy in approximating the root of the function. Overall, the results suggest that the proposed method (Proposed6) offers a favorable balance of accuracy, efficiency, and computational cost for solving Problem 2. Its ability to converge to the desired root with minimal computational resources underscores its effectiveness as a numerical solution approach.

**Problem 3.** $f_3(x) = x^3 + \cos(x) + 2, \quad x^* \approx -1.31.$

**Table 3.** Numerical simulations for $f_3(x)$ in Problem 3 at initial guess $x_0 = 0.5$.

| Method | $k$ | COC | $\varepsilon$ | $\chi$ | root | CPU time |
|---|---|---|---|---|---|---|
| NR2 | 15 | 30 | 6.96e-210 | 1.97e-418 | -1.31e+00 | 1.88e-01 |
| AS6 | 7 | 35 | 1.18e-402 | 4.25e-2412 | -1.31e+00 | 2.04e-01 |
| SA6 | 6 | 36 | 7.58e-750 | 0.00e+00 | -1.31e+00 | 1.87e-01 |
| W5 | 20 | 100 | 5.80e+28 | 1.25e+86 | 5.01e+28 | 5.00e-01 |
| Proposed | 5 | 25 | 2.20e-410 | 7.60e-2459 | -1.31e+00 | 1.5600e-01 |

Table 3 presents a comparative analysis of several numerical methods applied to solve Problem 3, which involves finding the root of the function $f_3(x) = x^3 + \cos(x) + 2$. The methods evaluated include NR2, AS6, SA6, W5, and the proposed method (Proposed). Each method is evaluated based on several criteria: the number of iterations ($k$), the computational cost (COC), the absolute error ($\varepsilon$), the functional value ($\chi$), the approximate root, and the CPU time (in seconds).

From the data, it is evident that all methods converge to an approximate root of $x^* \approx -1.31$. However, there are significant differences in the performance of the methods across other metrics. The proposed method achieves competitive results in terms of the number of iterations, computational cost, and CPU time compared to the other methods. Additionally, it exhibits extremely low absolute error and functional value, indicating high accuracy in approximating the root of the function. Among the other methods, NR2, AS6, and SA6 also demonstrate relatively good performance, achieving low absolute errors and functional values. However, they require a slightly higher computational cost and CPU time compared to the proposed method. On the other hand, W5 method shows significantly higher computational cost and absolute error, indicating less efficiency and accuracy compared to the other methods.

Overall, the results suggest that the proposed method offers a favorable balance of accuracy, efficiency, and computational cost for solving Problem 3. Its ability to converge to the desired root with minimal computational resources underscores its effectiveness as a numerical solution approach.

**Problem 4.** $f_4(x) = \log(x) - x^3 + 2\sin(x), \quad x^* \approx 1.3.$

**Table 4.** Numerical simulations for $f_4(x)$ in Problem 4 at initial guess $x_0$ = 1.5.

| Method | $k$ | COC | $\varepsilon$ | $\chi$ | root | CPU time |
|---|---|---|---|---|---|---|
| NR2 | 10 | 20 | 6.46e-334 | 2.15e-666 | 1.30 | 1.72e-01 |
| AS6 | 5 | 25 | 2.75e-772 | 1.00e-3999 | 1.30 | 1.41e-01 |
| SA6 | 5 | 30 | 1.46e-843 | 1.00e-3999 | 1.30 | 1.72e-01 |
| W5 | 5 | 25 | 1.13e-302 | 1.97e-1508 | 1.30 | 2.66e-01 |
| Proposed | 5 | 25 | 1.90e-851 | 1.00e-3999 | 1.30 | 1.57e-01 |

Table 4 provides a comparative analysis of numerical methods applied to solve Problem 4, aiming to find the root of the function $f_4(x)$. The methods evaluated include NR2, AS6, SA6, W5, and a proposed method. While all methods converge to an approximate root of $x^* \approx 1.3$, notable differences emerge in performance metrics. The proposed method demonstrates competitive results in terms of iterations, computational cost, and CPU time, alongside minimal absolute error and functional value. SA6 also exhibits good performance but with slightly higher computational cost. NR2 is better in terms of COC but its functional value is not promosing. In contrast, W5 shows higher computational cost and absolute error. Overall, the proposed method offers a favorable balance of accuracy and efficiency for solving Problem 4. Further exploration could validate its applicability across a broader range of functions.

**Problem 5.** $f_5(x) = x^3 - 10, \quad x^* \approx 2.154434690031883.$

**Table 5.** Numerical simulations for $f_5(x)$ in Problem 5 at initial guess $x_0$ = 1.5.

| Method | $k$ | COC | $\varepsilon$ | $\chi$ | root | CPU time |
|---|---|---|---|---|---|---|
| NR2 | 10 | 20 | 4.77e-221 | 1.47e-440 | 2.15 | 1.60e-02 |
| AS6 | 5 | 25 | 4.65e-496 | 6.09e-2973 | 2.15 | 1.50e-02 |
| SA6 | 5 | 30 | 2.82e-556 | 1.50e-3334 | 2.15 | 1.60e-02 |
| W5 | 6 | 30 | 2.68e-238 | 7.18e-1188 | 2.15 | 1.60e-02 |
| Proposed | 5 | 25 | 1.98e-576 | 1.52e-3455 | 2.15 | 1.50e-02 |

In the comparative analysis of numerical methods for solving $f_5(x) = x^3 - 10$ in Table 5, the proposed sixth-order method demonstrates superior performance in key aspects of numerical approximation, especially in the metrics of convergence speed and accuracy. Despite sharing the lowest iteration count (5) with AS6 and SA6 methods, which suggests a rapid convergence towards the root, the Proposed method distinguishes itself by achieving the smallest absolute error ($\varepsilon = 1.98e{-}576$) and the smallest absolute functional value ($\chi = 1.52e - 3455$). This indicates an exceptional precision level, far surpassing the other methods in achieving closeness to the true root, which is critical for applications demanding high numerical accuracy. Furthermore, the computational cost (COC) of the proposed method is competitive, equaling that of AS6, which, combined with a comparable CPU time to the fastest methods, underscores its efficiency. Therefore, considering the balance between computational resources and the accuracy of the result, the proposed sixth-order method emerges as notably advantageous for solving scalar non-linear equations, as exemplified in this comparison.

# 5 Conclusion and Future Work

In conclusion, this study introduces an innovative three-step iterative approach that significantly advances the field of nonlinear equation solving in science and engineering. By eliminating the requirement for second derivatives, this method marks a departure from conventional techniques such as Halley's method. Demonstrating a sixth-order convergence and necessitating only five function evaluations, the proposed approach proves to be highly efficient, with an index of approximately 1.430969. Its effectiveness in addressing nonlinear problems involving equations with algebraic and transcendental terms is evident. Through comparative analysis with existing root-solving algorithms, the superior performance of the proposed method is showcased. The findings not only validate the robustness and efficacy of the three-step iterative approach but also underscore its potential for broad application across various scientific and technical domains.

For future research, there are several promising directions to consider. Firstly, the application of this approach to more complex nonlinear systems, such as those involving multiple variables or higher-dimensional spaces, warrants investigation. This could involve adapting the method to handle systems of nonlinear equations or exploring its use in solving partial differential equations commonly encountered in physics and engineering. Secondly, integrating the proposed method with other numerical techniques, such as Newton's method or the secant method, could lead to the development of hybrid approaches that combine the strengths of different algorithms, potentially offering improved convergence rates and stability for a wider range of problems. Lastly, the implementation of parallel computing strategies for the proposed method could significantly reduce computational time, making it more attractive for large-scale scientific and engineering problems. Leveraging modern parallel computing architectures and techniques could enable the efficient handling of large datasets and complex simulations, further expanding the applicability and impact of the three-step iterative approach in the scientific community.

# 6 Authors' Contributions

**Asad Ali Chandio** : Conception, Methodology, Writing-Original draft preparation **Asif Ali Shaikh**: Supervision. **Sania Qureshi** : Suggestions, Writing-Reviewing, Investigation, Supervision. **Abdul Rehman Soomroo**: Editing and Formatting.

# 7 Compliance with Ethical Standards

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

# 8 Funding Information

No funding was received for this study from any organizations that might have an interest in the outcome of the research. The authors are solely responsible for the content and writing of the paper.

# 9 Authors' Information

Asad Ali Chandio : 0009-0008-6274-5004
Asif Ali Shaikh : 0000-0002-3084-922X
Sania Qureshi: 0000-0002-7225-2309
Abdul Rehman Soomroo: 0009-0007-4583-9650

# References


[1] Abro, H. A. and Shaikh, M. M. [2019], 'A new time-efficient and convergent nonlinear solver', *Applied Mathematics and Computation* **355**, 516–536.

[2] Ali, S., Solangi, M. A. and Qureshi, S. [2022], 'A modified three-step iterative method for solving nonlinear mathematical models'.

[3] Awadalla, M., Qureshi, S., Soomro, A. and Abuasbeh, K. [2023], 'A novel three-step numerical solver for physical models under fractal behavior', *Symmetry* **15**(2), 330.

[4] Cordero, A., Hueso, J. L., Martínez, E. and Torregrosa, J. R. [2012], 'Steffensen type methods for solving nonlinear equations', *Journal of computational and applied mathematics* **236**(12), 3058–3064.

[5] Cordero, A., Torregrosa, J. R. and Triguero-Navarro, P. [2021], 'A general optimal iterative scheme with arbitrary order of convergence', *Symmetry* **13**(5), 884.

[6] Fang, L., Chen, T., Tian, L., Sun, L. and Chen, B. [2011], 'A modified newton-type method with sixth-order convergence for solving nonlinear equations', *Procedia Engineering* **15**, 3124–3128.

[7] Gdawiec, K., Argyros, I. K., Qureshi, S. and Soomro, A. [2023], 'An optimal homotopy continuation method: Convergence and visual analysis', *Journal of Computational Science* **74**, 102166.

[8] Gościniak, I. and Gdawiec, K. [2018], Pso-based newton-like method and iteration processes in the generation of artistic patterns, *in* 'Advances in Visual Computing: 13th International Symposium, ISVC 2018, Las Vegas, NV, USA, November 19–21, 2018, Proceedings 13', Springer, pp. 47–56.

[9] Jaiswal, J. P. et al. [2014], 'Some class of third-and fourth-order iterative methods for solving nonlinear equations', *Journal of Applied Mathematics* **2014**.

[10] Jamali, K., Solangi, M. A. and Qureshi, S. [2022], 'A novel hybrid iterative method for applied mathematical models with time-efficiency', *Journal of Applied Mathematics and Computational Mechanics* **21**(3).

[11] Kumari, S., Gdawiec, K., Nandal, A., Kumar, N. and Chugh, R. [2023], 'An application of viscosity approximation type iterative method in the generation of mandelbrot and julia fractals', *Aequationes mathematicae* **97**(2), 257–278.

[12] Kung, H. and Traub, J. F. [1974], 'Optimal order of one-point and multipoint iteration', *Journal of the ACM (JACM)* **21**(4), 643–651.

[13] Mastoi, V., Ali, A., Shaikh, M. M. and Shaikh, A. W. [2020], 'A new third-order derivative-based iterative method for nonlinear equations', *J. Mech. Cont. & Math. Sci* **15**(10), 110–123.

[14] Naseem, A., Rehman, M. and Abdeljawad, T. [2022], 'A novel root-finding algorithm with engineering applications and its dynamics via computer technology', *IEEE Access* **10**, 19677–19684.

[15] Naseem, A., Rehman, M., Qureshi, S. and Ide, N. A. D. [2023], 'Graphical and numerical study of a newly developed root-finding algorithm and its engineering applications', *IEEE Access* **11**, 2375–2383.

[16] Nazeer, W., Tanveer, M., Kang, S. M. and Naseem, A. [2016], 'A new householder's method free from second derivatives for solving nonlinear equations and polynomiography', *J. Nonlinear Sci. Appl* **9**(3), 998–1007.

[17] Ortega, J. M. [1990], *Numerical analysis: a second course*, SIAM.

[18] Parida, P. K. and Gupta, D. K. [2007], 'A cubic convergent iterative method for enclosing simple roots of nonlinear equations', *Applied mathematics and computation* **187**(2), 1544–1551.

[19] Qureshi, S., Argyros, I. K., Soomro, A., Gdawiec, K., Shaikh, A. A. and Hincal, E. [2024], 'A new optimal root-finding iterative algorithm: local and semilocal analysis with polynomiography', *Numerical Algorithms* **95**(4), 1715–1745.

[20] Qureshi, S., Ramos, H. and Soomro, A. K. [2021], 'A new nonlinear ninth-order root-finding method with error analysis and basins of attraction', *Mathematics* **9**(16), 1996.

[21] Qureshi, S., Soomro, A., Naseem, A., Gdawiec, K., Argyros, I. K., Alshaery, A. A. and Secer, A. [2024], 'From halley to secant: Redefining root finding with memory-based methods including convergence and stability', *Mathematical Methods in the Applied Sciences* .

[22] Shah, F. A., Noor, M. A. and Waseem, M. [2016], 'Some second-derivative-free sixth-order convergent iterative m ethods for non-linear equations', *Maejo International Journal of Science and Technology* **10**(1), 79.

[23] Soomro, A., Naseem, A., Qureshi, S., Al Din Ide, N. et al. [2022], 'Development of a new multi-step iteration scheme for solving non-linear models with complex polynomiography', *Complexity* **2022**.

[24] Tassaddiq, A., Qureshi, S., Soomro, A., Hincal, E., Baleanu, D. and Shaikh, A. A. [2021], 'A new three-step root-finding numerical method and its fractal global behavior', *Fractal and Fractional* **5**(4), 204.

[25] Thangkhenpau, G., Panday, S., Bolunduţ, L. C. and Jäntschi, L. [2023*a*], 'Efficient families of multi-point iterative methods and their self-acceleration with memory for solving nonlinear equations', *Symmetry* **15**(8), 1546.

[26] Thangkhenpau, G., Panday, S., Bolunduţ, L. C. and Jäntschi, L. [2023*b*], 'Efficient families of multi-point iterative methods and their self-acceleration with memory for solving nonlinear equations', *Symmetry* **15**(8), 1546.

[27] Waseem, M., Noor, M. A. and Noor, K. I. [2016], 'Efficient method for solving a system of nonlinear equations', *Applied Mathematics and Computation* **275**, 134–146.

[28] Zafar, F., Bibi, G. et al. [2014], 'A family of fourteenth-order convergent iterative methods for solving nonlinear equations', *Chinese Journal of Mathematics* **2014**.