

Research Article

Some Novel Sixth-Order Iteration Schemes for Computing Zeros of Nonlinear Scalar Equations and Their Applications in Engineering

M. A. Rehman ¹, Amir Naseem ¹, and Thabet Abdeljawad ^{2,3,4}

¹Department of Mathematics, University of Management and Technology, Lahore 54770, Pakistan

²Department of Mathematics and General Sciences, Prince Sultan University, 11586 Riyadh, Saudi Arabia

³Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan

⁴Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan

Correspondence should be addressed to Amir Naseem; amir.kasuri89@gmail.com
and Thabet Abdeljawad; tabdeljawad@psu.edu.sa

Received 8 January 2021; Revised 3 March 2021; Accepted 11 March 2021; Published 13 April 2021

Academic Editor: Ioan Rasa

Copyright © 2021 M. A. Rehman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we propose two novel iteration schemes for computing zeros of nonlinear equations in one dimension. We develop these iteration schemes with the help of Taylor's series expansion, generalized Newton-Raphson's method, and interpolation technique. The convergence analysis of the proposed iteration schemes is discussed. It is established that the newly developed iteration schemes have sixth order of convergence. Several numerical examples have been solved to illustrate the applicability and validity of the suggested schemes. These problems also include some real-life applications associated with the chemical and civil engineering such as adiabatic flame temperature equation, conversion of nitrogen-hydrogen feed to ammonia, the van der Wall's equation, and the open channel flow problem whose numerical results prove the better efficiency of these methods as compared to other well-known existing iterative methods of the same kind.

1. Introduction

The solution of nonlinear scalar equations plays a vital role in many fields of applied sciences such as Engineering, Physics, and Mathematics. Analytical methods do not help us to solve such equations, and therefore, we need iterative methods for approximate the solution. In an iterative process, the first step is to choose an initial guess x_0 which is improved step by step by means of iterations till the approximate solution is achieved with the required accuracy. Some basic iterative methods are given in literature [1–8] and the references therein. In the last few years, a lot of researchers worked on iterative methods with their applications and proposed some new iterative schemes which possesses either a high convergence rate or have less number of functional evaluations per iteration, see [9–21] and the references therein. The convergence rate of an iterative method can be increased by involving predictor and corrector steps which results multi-

step iterative methods whereas the number of functional evaluations can be reduced by removing second and higher derivatives in the considered iterative method using different mathematical techniques. When we try to raise the convergence rate of an iterative scheme, we have to use more functional evaluations per iteration, and similarly, less number of functional evaluations per iterations causes low order of convergence which is the main drawback. It is much difficult to manage both terms, i.e., the convergence rate and functional evaluations per iterations as it seems that there exists an inverse relation between them. In twenty-first century, many mathematicians try to modify the existing methods with less number of functional evaluations per iterations and higher convergence order by applying different techniques such as predictor-corrector technique, finite difference scheme, interpolation technique, Taylor's series, and quadrature formula etc. In 2007, Noor et al. [22] introduced a two-step Halley's method with sextic convergence and then approximated its

second derivative by the utilization of finite difference scheme and suggested a novel second-derivative free iterative algorithm which have fifth convergence order. In 2012, Hafiz and Al-Goria [23] suggested two new algorithms with order seven and nine, respectively, which were based on the weight combination of midpoint with Simpson quadrature formulas and using the predictor-corrector technique. Nazeer et al. [24] in 2016 proposed a new second derivative free generalized Newton-Raphson's method with convergence of order five by means of finite difference scheme. In 2017, Kumar et al. [25] suggested a sixth-order parameter-based family of algorithms for solving nonlinear equations. In the same year, Salimi et al. [26] proposed an optimal class of eighth-order methods by using weight functions and Newton interpolation technique. Very recently, Naseem et al. [27] presented some new sixth-order algorithms for finding zeros of nonlinear equations and then investigated their dynamics by means of polynomio-graphy and presented some novel mathematical art through the execution of the presented algorithms.

In this paper, we suggested two novel iteration schemes in the form of predictor-corrector type numerical methods, namely, Algorithms 1 and 2, by taking Newton's iteration method as a predictor step. The derivation of the first iteration scheme is purely based on the Taylor's series expansion and generalized Newton-Raphson's method whereas in second one, we use interpolation technique for removing its second derivative which results the higher efficiency index. We examined the convergence criteria of the suggested schemes and proved that these iteration schemes bearing sextic convergence and superior to the other well-known methods of the similar nature. The efficiency indices of the presented schemes have been compared with the other similar existing two-step iteration schemes. The proposed iteration schemes have been applied to solve some real life problems along with the arbitrary transcendental and algebraic equations in order to assess its applicability, validity, and accuracy.

2. Main Results

Consider the nonlinear algebraic equation

$$f(x) = 0. \quad (1)$$

We assume that α is a simple zero of (1) and x_0 is an initial guess sufficiently close to α . Using the Taylor's series around x_0 for (1), we have

$$f(x_0) + (x - x_0)f'(x_0) + \frac{1}{2!}(x - x_0)^2f''(x_0) + \dots = 0. \quad (2)$$

If $f'(x_0) \neq 0$, we can evaluate the above expression as follows:

$$f(x_0) + (x - x_0)f'(x_0) = 0. \quad (3)$$

If we choose x_{j+1} the root of equation, then we have

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)}. \quad (4)$$

This is quadratically convergent Newton's method [2-4] for root-finding of nonlinear functions and needs two computations for its execution. From (2), one can evaluate

$$x = x_0 - \frac{f'(x_0) - \sqrt{f'^2(x_0) - 2f(x_0)f''(x_0)}}{f''(x_0)}. \quad (5)$$

In iterative form:

$$x_{j+1} = x_j - \frac{f'(x_j) - \sqrt{f'^2(x_j) - 2f(x_j)f''(x_j)}}{f''(x_j)}, \quad (6)$$

which is cubically convergent generalize Newton-Raphson's method [28] and requires three functional evaluations per iteration for the execution. After simplification of (2), one can obtain:

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(x - x_0)^2 f''(x_0)}{2f'(x_0)}. \quad (7)$$

Now from generalized Newton-Raphson's method in (5)

$$x - x_0 = - \frac{f'(x_0) - \sqrt{f'^2(x_0) - 2f(x_0)f''(x_0)}}{f''(x_0)}. \quad (8)$$

Using (8) in (7), we obtain

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{\left[f'(x_0) - \sqrt{f'^2(x_0) - 2f(x_0)f''(x_0)} \right]^2}{2f'(x_0)f''(x_0)}. \quad (9)$$

After rewriting the above obtained equality in the general form with the insertion of Newton's iteration method as a predictor, we arrive at a new algorithm of the form:

Algorithm 1. For a given x_0 , compute the approximate solution x_{j+1} by the following iterative schemes:

$$y_j = x_j - \frac{f(x_j)}{f'(x_j)}, j = 0, 1, 2, \dots,$$

$$x_{j+1} = y_j - \frac{f(y_j)}{f'(y_j)} - \frac{\left[f'(y_j) - \sqrt{f'^2(y_j) - 2f(y_j)f''(y_j)} \right]^2}{2f'(y_j)f''(y_j)}, \tag{10}$$

which is the modification of the generalized Newton-Raphson’s method for determining the approximate roots of the nonlinear algebraic equations. To find the approximate root of the given nonlinear equation by means of the above described algorithm, one has to find the first as well as the second derivative of the given function $f(x)$. But in several cases, we have to deal with such functions in which second derivative does not exist and our proposed algorithm fails to find approximate root in that situation. To resolve this issue, we apply interpolation technique for the approximation of the second derivative as follows:

Consider the function

$$\rho(u) = a_1 + a_2(u - y_j) + a_3(u - y_j)^2 + a_4(u - y_j)^3, \tag{11}$$

where the values of the unknowns $a_1, a_2, a_3,$ and a_4 can be found by applying the following interpolation conditions:

$$\begin{aligned} f(x_j) &= \rho(x_j), f(w_j) = \rho(w_j), f'(w_j) = \rho'(w_j), \\ f'(x_j) &= \rho'(x_j), f''(w_j) = \rho''(w_j). \end{aligned} \tag{12}$$

From the above conditions, we gain a system containing four linear equations with four variables, the solution of which gives the following equality:

$$\begin{aligned} f''(y_j) &= \frac{6[f(x_j) - f(y_j)] - 2[x_j - y_j][2f'(y_j) + f'(x_j)]}{(x_j - y_j)^2} \\ &= \rho(x_j, y_j). \end{aligned} \tag{13}$$

After putting the value of $f''(y_j)$ from the above equality in Algorithm 1, we gain novel second-derivative free algorithm as follows:

Algorithm 2. For a given x_0 , compute the approximate solution x_{j+1} by the following iterative schemes:

$$y_j = x_j - \frac{f(x_j)}{f'(x_j)}, j = 0, 1, 2, \dots,$$

$$x_{j+1} = y_j - \frac{f(y_j)}{f'(y_j)} - \frac{\left[f'(y_j) - \sqrt{f'^2(y_j) - 2f(y_j)\rho(x_j, y_j)} \right]^2}{2f'(y_j)\rho(x_j, y_j)}, \tag{14}$$

which is a novel second-derivative free iterative algorithm for computing the approximate solutions of the nonlinear algebraic equations. One of the main features of the suggested algorithm is that it can be applied to all those nonlinear functions in which second derivative does not exist. The removal of second derivative causes less number of functional evaluations per iteration which yields the best efficiency index as compared to those methods which require second derivative. The results of the given test examples certified its best performance in comparison with the other similar existing methods in literature.

3. Convergence Analysis

This section includes the discussion regarding the convergence criteria of the suggested iteration schemes.

Theorem 3. *Assuming α as a simple zero of the given equation $f(x) = 0$, where $f(x)$ is sufficiently smooth in the neighborhood of α , then the convergence orders of Algorithms 1 and 2 are at least six.*

Proof. To prove the convergence of Algorithms 1 and 2, we assume that α is the simple root of the equation $f(x) = 0$ and e_j be the error at n th iteration; then, $e_j = x_j - \alpha$ and by using Taylor series about $x = \alpha$, we have

$$\begin{aligned} f(x_j) &= f'(\alpha)e_j + \frac{1}{2!}f''(\alpha)e_j^2 + \frac{1}{3!}f'''(\alpha)e_j^3 + \frac{1}{4!}f^{(iv)}(\alpha)e_j^4 \\ &\quad + \frac{1}{5!}f^{(v)}(\alpha)e_j^5 + \frac{1}{6!}f^{(vi)}(\alpha)e_j^6 + O(e_j^7), \end{aligned} \tag{15}$$

$$f(x_j) = f'(\alpha) \left[e_j + c_2e_j^2 + c_3e_j^3 + c_4e_j^4 + c_5e_j^5 + c_6e_j^6 + O(e_j^7) \right], \tag{16}$$

$$\begin{aligned} f'(x_j) &= f'(\alpha) \left[1 + 2c_2e_j + 3c_3e_j^2 + 4c_4e_j^3 + c_5e_j^4 + 6c_6e_j^5 \right. \\ &\quad \left. + 7c_7e_j^6 + O(e_j^7) \right], \end{aligned} \tag{17}$$

where

$$c_n = \frac{1}{n!} \frac{f^{(n)}(\alpha)}{f'(\alpha)}. \tag{18}$$

With the help of equations (16) and (17), we get

$$\begin{aligned} y_j = f'(\alpha) & \left[\alpha + c_2 e_j^2 + (2c_3 - 2c_2^2) e_j^3 + 3c_4 - 7c_2 c_3 + 4c_2^3 \right] e_j^4 \\ & + (-6c_3^2 + 20c_3 c_2^2 - 10c_2 c_4 + 4c_5 - 8c_2^4) e_j^5 \\ & + (-17c_4 c_3 + 28c_4 c_2^2 - 13c_2 c_5 + 5c_6 \\ & + 33c_2 c_3^2 - 52c_3 c_2^3 + 16c_2^5) e_j^6 + O(e_j^7), \end{aligned} \quad (19)$$

$$\begin{aligned} f(y_j) = f'(\alpha) & \left[c_2 e_j^2 + (2c_3 - 2c_2^2) e_j^3 + (5c_2^3 - 7c_2 c_3 + 3c_4) e_j^4 \right. \\ & + (24c_3 c_2^2 - 12c_2^4 - 10c_2 c_4 + 4c_5 - 6c_2^3) e_j^5 \\ & + (-73c_3 c_2^3 + 34c_4 c_2^2 + 28c_2^5 + 37c_2 c_3^2 - 17c_4 c_3 \\ & \left. - 13c_2 c_5 + 5c_6) e_j^6 + O(e_j^7) \right], \end{aligned} \quad (20)$$

$$\begin{aligned} f'(y_j) = f'(\alpha) & \left[1 + 2c_2^2 e_j^2 + (4c_2 c_3 - 4c_2^3) e_j^3 \right. \\ & + (6c_2 c_4 - 11c_3 c_2^2 + 8c_2^4) e_j^4 + (28c_3 c_2^3 - 20c_4 c_2^2 \\ & + 8c_2 c_5 - 16c_2^5) e_j^5 + (-16c_4 c_2 c_3 - 68c_3 c_2^4 + 12c_3^3 \\ & \left. + 60c_4 c_2^3 - 26c_5 c_2^2 + 10c_2 c_6 + 32c_2^6) e_j^6 + O(e_j^7) \right], \end{aligned} \quad (21)$$

$$\begin{aligned} f''(y_j) = f'(\alpha) & \left[2c_2 + 6c_2 c_3 e_j^2 + (12c_3^2 - 12c_3 c_2^2) e_j^3 \right. \\ & + (-42c_2 c_3^2 + 18c_4 c_3 + 24c_3 c_2^3 + 12c_4 c_2^2) e_j^4 \\ & + (-12c_2 c_4 c_3 + 24c_5 c_3 - 36c_3^3 + 120c_3^2 c_2^2 \\ & - 48c_3 c_2^4 - 48c_4 c_2^3) e_j^5 + (-78c_3 c_2 c_5 + 30c_3 c_6 \\ & - 54c_4 c_2^3 - 96c_3 c_4 c_2^2 + 198c_2 c_3^3 - 312c_3^2 c_2^3 + 96c_3 c_2^5 \\ & \left. + 72c_2 c_4^2 + 144c_4 c_2^4 + 20c_5 c_2^3) e_j^6 + O(e_j^8) \right]. \end{aligned} \quad (22)$$

With the help of equations (16)–(21), we have

$$\begin{aligned} \rho(x_j, y_j) = f'(\alpha) & \left[2c_2 + (6c_2 c_3 - 2c_4) e_j^2 + (12c_3^2 - 12c_3 c_2^2 \right. \\ & + 4c_2 c_4 - 4c_3) e_j^3 + (2c_2 c_5 + 26c_3 c_4 - 42c_2 c_3^2 \\ & + 24c_3 c_2^3 + 2c_4 c_2^2 - 6c_6) e_j^4 + (-48c_4 c_2 c_3 + 12c_2^4 \\ & - 24c_4 c_2^3 + 28c_5 c_3 + 4c_5 c_2^2 + 120c_3^2 c_2^2 - 48c_3 c_2^4 \\ & - 8c_7 - 36c_3^3) e_j^5 + (-60c_5 c_2 c_3 + 28c_4 c_3 c_2^2 - 2c_2 c_7 \\ & + 22c_5 c_4 - 10c_5 c_2^3 + 30c_6 c_3 + 6c_6 c_2^2 + 20c_2 c_4^2 \\ & - 86c_4 c_2^3 + 88c_4 c_2^4 + 198c_2 c_3^3 - 312c_3^2 c_2^3 \\ & \left. + 96c_3 c_2^5 - 10c_8) e_j^6 + O(e_j^7) \right]. \end{aligned} \quad (23)$$

Using equations (19)–(23) in Algorithms 1 and 2, we get the following equalities

$$\begin{aligned} x_{j+1} & = \alpha + (-c_3 c_2^3) e_j^6 + O(e^7), \\ x_{j+1} & = \alpha + (-c_3 c_2^3 + c_4 c_2^2) e_j^6 + O(e^7), \end{aligned} \quad (24)$$

which imply that

$$e_{n+1} = (-c_3 c_2^3) e_j^6 + O(e^7), \quad (25)$$

$$e_{n+1} = (-c_3 c_2^3 + c_4 c_2^2) e_j^6 + O(e^7). \quad (26)$$

Equations (25) and (26) show that the orders of convergence of Algorithms 1 and 2 are atleast six.

4. Comparison of Efficiency Index

In numerical analysis, the efficiency index of an algorithm provides us the information about the speed and performance of the algorithm which is being under the consideration. It is actually a numerical quantity that relates to the number of computational resources needed to execute the considered algorithm. The efficiency of an algorithm can be thought of as analogous to the engineering productivity for a process that includes iterations. The term efficiency index is used to analyze the numeric behavior of different algorithms. In iterative algorithms, this quantity totally depends upon the two factors. The first one is the convergence order of the algorithm whereas the second factor is the number of computations per iteration, i.e., the number of functional and derivatives evaluations, required to execute the algorithm for the purpose of root-finding of the nonlinear functions. If the convergence order is represented by P and the number of computations per iteration by n_f , then the efficiency index can be written mathematically as:

$$\text{Efficiency Index} = P^{1/n_f}. \quad (27)$$

Since Noor's method one [11] has quadratic convergence and requires three computations per iteration for execution, so its efficiency index will be $2^{1/3} \approx 1.2599$. In the same way, the cubically convergent Noor's method two [11] requires three computations per iteration and has $3^{1/3} \approx 1.4422$ as an efficiency index. Similarly, the efficiency index of the Traub's methods [6] is $4^{1/4} \approx 1.4142$ because it possesses the convergence of order four with four computations for execution. Since the modified Halley's method [22] has fifth convergence order with four computations per iteration, so its efficiency index will be $5^{1/4} \approx 1.4953$. Now, we calculate the efficiency indices of the suggested algorithms. Both algorithms bearing the convergence of order six. The number of computations per iteration for the execution of the first algorithm is five whereas the second proposed algorithm requires only four evaluations per iteration. So, their efficiency indices will be $6^{1/5} \approx 1.4310$ and $6^{1/4} \approx 1.5651$, respectively. The efficiency indices of the different iterative methods, we have discussed above, are summarized in the following Table 1.

TABLE 1: Comparison of efficiency indices of different iterative methods.

Method	Convergence order	No. of required computations	Efficiency index
Noor's method one	2	3	1.2599
Noor's method two	3	3	1.4422
Traub's method	4	4	1.4142
Modified Halley's method	5	4	1.4953
Algorithm 1	6	5	1.4310
Algorithm 2	6	4	1.5651

Table 1 clearly shows that the presented method, namely, Algorithm 2, has better efficiency index among the other compared methods.

5. Numerical Comparisons and Applications

In this section, we include four real-life engineering problems and seven arbitrary problems in the form of transcendental and algebraic equations to illustrate the applicability and efficiency of our newly developed iterative methods. We compare these methods with the following similar existing two-step iteration schemes:

5.1. Noor's Method One (NM1). For a provided initial guess x_0 , determine the approximate root x_{j+1} with the iteration schemes given below:

$$\begin{aligned}
 x_{j+1} &= x_j - \frac{f(x_j)}{f'(x_j)}, j = 0, 1, 2, 3, \dots, \\
 x_{j+1} &= x_j - \frac{f(x_j)}{f'(x_j)} + \left[\frac{f(x_j)}{f'(x_j)} \right] \frac{f'(y_j)}{f'(x_j)},
 \end{aligned} \tag{28}$$

which is quadratically convergent Noor's method one [11] for root-finding of nonlinear equations.

5.2. Noor's Method Two (NM2). For a provided initial guess x_0 , determine the approximate root x_{j+1} with the iteration schemes given below:

$$\begin{aligned}
 y_j &= x_j - \frac{f(x_j)}{f'(x_j)}, j = 0, 1, 2, 3, \dots, \\
 x_{j+1} &= x_j - \frac{2f(x_j)}{f'(x_j) + f'(y_j)},
 \end{aligned} \tag{29}$$

which is cubically convergent Noor's method two [11] for root-finding of nonlinear equations.

5.3. Traub's Method (TM). For a provided initial guess x_0 , determine the approximate root x_{j+1} with the iteration schemes given below:

$$\begin{aligned}
 y_j &= x_j - \frac{f(x_j)}{f'(x_j)}, j = 0, 1, 2, 3, \dots, \\
 x_{j+1} &= y_j - \frac{f(y_j)}{f'(y_j)},
 \end{aligned} \tag{30}$$

which is two-step fourth order Traub's method [6] for root-finding of nonlinear equations which bearing the convergence of order four.

5.4. Modified Halley's Method (MHM). For a provided initial guess x_0 , determine the approximate root x_{j+1} with the iteration schemes given below:

$$\begin{aligned}
 y_j &= x_j - \frac{f(x_j)}{f'(x_j)}, j = 0, 1, 2, 3, \dots, \\
 x_{j+1} &= y_j - \frac{2f(x_j)f(y_j)f'(y_j)}{2f(x_j)f'2(y_j) - f'2(x_j)f(y_j) + f'(x_j)f'(y_j)f(y_j)},
 \end{aligned} \tag{31}$$

which is two-step Halley's method [22] for root-finding of nonlinear equations which has the convergence of fifth order. In order to make the numerical comparison of the above defined methods with the presented algorithms, we consider the following test Examples 1–5.

The general algorithm for finding the approximate solution of the given nonlinear functions is given as:

In Algorithm 3, we take the accuracy $\varepsilon = 10^{-15}$ in the stopping criteria $|x_{j+1} - x_j| < \varepsilon$. We did all the calculations of the numerical examples with the aid of the computer program Maple 13, and their numerical results can be seen in the following presented Tables 2–6.

Example 1. Adiabatic flame temperature equation. The adiabatic flame temperature equation is represented by the following relation:

$$f_1(x) = \Delta H + a_1(x - 298) + \frac{a_2}{2}(x^2 - 298^2) + \frac{a_3}{3}(x^3 - 298^3), \tag{32}$$

where $\Delta H = -57798$, $a_1 = 7.256$, $a_2 = 0.002298$, and $a_3 = 0.00000283$. For further details, see [29, 30] and the references therein. The above function is actually a polynomial of degree three, and by the fundamental theorem of Algebra, it must have exactly three roots. Among these roots, $\alpha = 4305.3099136661$ is a simple one which we approximated

Input: $f \in \mathbb{R}$ — non-linear function, k — maximum number of iterations, I — iteration method, ε — accuracy.
Output: Approximated root of the given non-linear function.
for $x_0 \in A$ **do**
 $i = 0$.
 while $i \leq k$ **do**
 $x_{j+1} = I(x_j)$
 if $|x_{j+1} - x_j| < \varepsilon$ **then**
 break
 $i = i + 1$
 x_{j+1} is the required solution.

ALGORITHM 3: General root's finding algorithm.

TABLE 2: Numerical comparison among different algorithms for the engineering problem f_1 .

Method	N	$ f(x_{j+1}) $	x_{j+1}	$\sigma = x_{j+1} - x_j $	COC
$f_1(x), x_0 = 2050.0$					
NR1	9	$3.688522e - 28$	4305.30991366612556300000	$3.947209e - 13$	2
NR2	4	$2.919985e - 37$	4305.30991366612556300000	$9.938805e - 11$	3
TM	3	$6.063382e - 31$	4305.30991366612556300000	$1.002459e - 05$	4
MHM	3	$1.311971e - 69$	4305.30991366612556300000	$1.526816e - 11$	5
Algorithm 1	2	$3.738643e - 18$	4305.30991366612556300000	$1.795691e - 00$	6
Algorithm 2	2	$3.738643e - 18$	4305.30991366612556300000	$1.795691e - 00$	6

TABLE 3: Numerical comparison among different algorithms for the engineering problem f_2 .

Method	N	$ f(x_{j+1}) $	x_{j+1}	$\sigma = x_{j+1} - x_j $	COC
$f_2(x), x_0 = 0.1$					
NR1	7	$9.675391e - 26$	0.27775954284172065910	$1.053628e - 13$	2
NR2	3	$1.203488e - 18$	0.27775954284172065910	$6.173552e - 07$	3
TM	3	$3.260304e - 39$	0.27775954284172065910	$1.412011e - 10$	4
MHM	2	$2.057683e - 15$	0.27775954284172065910	$1.970771e - 04$	5
Algorithm 1	2	$6.942638e - 22$	0.27775954284172065910	$2.202374e - 04$	6
Algorithm 2	2	$1.726207e - 21$	0.27775954284172065910	$2.502938e - 04$	6

TABLE 4: Numerical comparison among different algorithms for the engineering problem f_3 .

Method	N	$ f(x_{j+1}) $	x_{j+1}	$\sigma = x_{j+1} - x_j $	COC
$f_3(x), x_0 = 2.0$					
NR1	4	$1.319023e - 19$	1.92984624284786221696	$5.000588e - 10$	2
NR2	3	$3.958485e - 15$	1.92984624284786221696	$1.021691e - 05$	3
TM	3	$8.395139e - 34$	1.92984624284786221696	$2.556739e - 09$	4
MHM	2	$8.089146e - 19$	1.92984624284786221696	$1.079121e - 04$	5
Algorithm 1	2	$4.275791e - 23$	1.92984624284786221696	$7.584886e - 05$	6
Algorithm 2	2	$4.275791e - 23$	1.92984624284786221696	$7.584886e - 05$	6

through the proposed methods by choosing the initial guess $x_0 = 2050$, and the numerical results have been shown in Table2.

Example 2. Fraction conversion of nitrogen-hydrogen to ammonia. We take this example from [31], which describe the fraction conversion of nitrogen-hydrogen feed to

TABLE 5: Numerical comparison among different algorithms for the engineering problem f_4 .

Method	N	$ f(x_{j+1}) $	x_{j+1}	$\sigma = x_{j+1} - x_j $	COC
$f_4(x), x_0 = 0.4$					
NR1	6	$2.230770e - 24$	1.46509122029582464238	$1.280653e - 12$	2
NR2	3	$5.751429e - 27$	1.46509122029582464238	$4.220179e - 09$	3
TM	3	$7.765624e - 44$	1.46509122029582464238	$4.884637e - 11$	4
MHM	3	$4.020841e - 64$	1.46509122029582464238	$5.749371e - 13$	5
Algorithm 1	2	$2.394555e - 20$	1.46509122029582464238	$1.957777e - 03$	6
Algorithm 2	2	$1.709800e - 15$	1.46509122029582464238	$9.937512e - 03$	6

ammonia, usually known as fractional conversion. In this problem, the values of temperature and pressure have been taken as 500°C and 250 atm , respectively. This problem has the following nonlinear form:

$$f_2(x) = -0.186 - \frac{8x^2(x-4)^2}{9(x-2)^3}, \quad (33)$$

which can be easily reduced to the following polynomial:

$$f_2(x) = x^4 - 7.79075x^3 + 14.7445x^2 + 2.511x - 1.674. \quad (34)$$

Since the degree of the above polynomial is four, so, it must have exactly four roots. By definition, the fraction conversion lies in $(0, 1)$ interval, so only one real root exists in this interval which is 0.2777595428 . The other three roots have no physical meanings. We started the iteration process by the initial guess $x_0 = 0.1$. The numerical results through different methods have been shown in Table 3.

Example 3. Finding volume from van der Waal's equation. In Chemical Engineering, the van der Waal's equation has been used for interpreting real and ideal gas behavior [32], having the following form:

$$\left(P + \frac{A_1 n^2}{V^2}\right)(V - nA_2) = nRT. \quad (35)$$

By taking the specific values of the parameters of the above equation, we can easily convert it to the following nonlinear function:

$$f_3(x) = 0.986x^3 - 5.181x^2 + 9.067x - 5.289, \quad (36)$$

where s represents the volume that can easily be found by solving the function f_3 . Since the degree of the polynomial is three, so it must possess three roots. Among these roots, there is only one positive real root 1.9298462428 which is feasible because the volume of the gas can never be negative. We

start the iteration process with the initial guess $x_0 = 2.0$, and their results can be seen in Table 4.

Example 4. Open channel flow problem. The water flow in an open channel with uniform flow condition is given by Manning's equation [33], having the following standard form:

$$\text{Water Flow} = F = \frac{\sqrt{s}ar^{2/3}}{n}, \quad (37)$$

where s , a , and r represent the slope, area, and hydraulic radius of the corresponding channel, respectively, and n denotes Manning's roughness coefficient. For a rectangular-shaped channel, having width b and depth of water in channel x , then we may write:

$$\begin{aligned} a &= bx, \\ r &= \frac{bx}{b + 2x}. \end{aligned} \quad (38)$$

Using these values in (37), we obtain:

$$F = \frac{\sqrt{s}bx}{n} \left(\frac{bx}{b + 2x}\right)^{2/3}. \quad (39)$$

To find the depth of water in the channel for a given quantity of water, the above equation may written in the form of nonlinear function as:

$$f_4(x) = \frac{\sqrt{s}bx}{n} \left(\frac{bx}{b + 2x}\right)^{2/3} - F. \quad (40)$$

We take the values of different parameters as $F = 14.15\text{ m}^3/\text{s}$, $b = 4.572\text{ m}$, $s = 0.017$, and $n = 0.0015$. We choose the initial guess $x_0 = 0.4$ to start the iteration process, and the corresponding results through different iteration schemes are given in Table 5.

TABLE 6: Numerical comparison among different algorithms for transcendental and algebraic problems $f_5 - f_{11}$.

Method	N	$ f(x_{j+1}) $	x_{j+1}	$\sigma = x_{j+1} - x_j $	COC
$f_5(x), x_0 = -2$					
NR1	9	$1.485377e - 16$	-0.52248077281054548914	$1.517821e - 08$	2
NR2	7	$4.247308e - 18$	-0.52248077281054548914	$1.202925e - 06$	3
TM	25	$1.821056e - 17$	-0.52248077281054548914	$1.970771e - 04$	4
MHM	3	$8.137892e - 19$	-0.52248077281054548914	$3.574246e - 04$	5
Algorithm 1	2	$6.584167e - 72$	-0.52248077281054548914	$2.829449e - 12$	6
Algorithm 2	2	$3.852650e - 57$	-0.52248077281054548914	$8.860340e - 10$	6
$f_6(x), x_0 = 2$					
NR1	5	$3.163807e - 17$	0.40999201798913713162	$5.629386e - 09$	2
NR2	45	$3.382231e - 18$	0.40999201798913713162	$1.512669e - 06$	3
TM	4	$6.063382e - 56$	0.40999201798913713162	$1.586231e - 14$	4
MHM	3	$4.499472e - 42$	0.40999201798913713162	$1.880726e - 08$	5
Algorithm 1	2	$5.856963e - 25$	0.40999201798913713162	$2.788223e - 04$	6
Algorithm 2	2	$6.737587e - 25$	0.40999201798913713162	$2.852107e - 04$	6
$f_7(x), x_0 = 1.2$					
NR1	7	$3.278748e - 27$	0.56714329040978387300	$4.592634e - 14$	2
NR2	4	$1.127879e - 31$	0.56714329040978387300	$3.980665e - 11$	3
TM	3	$3.135655e - 36$	0.56714329040978387300	$1.588919e - 09$	4
MHM	3	$7.204101e - 59$	0.56714329040978387300	$2.420306e - 12$	5
Algorithm 1	2	$2.894187e - 17$	0.56714329040978387300	$2.115592e - 03$	6
Algorithm 2	2	$7.355480e - 18$	0.56714329040978387300	$1.597483e - 03$	6
$f_8(x), x_0 = 1.5$					
NR1	141	$3.196546e - 22$	2.15443469003188372180	$7.032555e - 12$	2
NR2	4	$7.066989e - 31$	2.15443469003188372180	$5.866631e - 11$	3
TM	3	$7.857615e - 27$	2.15443469003188372180	$2.740790e - 07$	4
MHM	3	$2.248861e - 57$	2.15443469003188372180	$5.868237e - 12$	5
Algorithm 1	2	$2.929220e - 17$	2.15443469003188372180	$2.575011e - 03$	6
Algorithm 2	2	$2.929220e - 17$	2.15443469003188372180	$2.575011e - 03$	6
$f_9(x), x_0 = 0.6$					
NR1	78	$2.720561e - 26$	1.00000000000000000000	$8.247062e - 14$	2
NR2	4	$3.726794e - 29$	1.00000000000000000000	$3.726794e - 29$	3
TM	3	$1.853180e - 25$	1.00000000000000000000	$5.187039e - 07$	4
MHM	3	$2.707444e - 54$	1.00000000000000000000	$1.950103e - 11$	5
Algorithm 1	2	$8.487620e - 17$	1.00000000000000000000	$2.340774e - 03$	6
Algorithm 2	2	$8.487620e - 17$	1.00000000000000000000	$2.340774e - 03$	2
$f_{10}(x), x_0 = -3.0$					
NR1	6	$7.104675e - 20$	-1.40449164821534122600	$1.911132e - 10$	2
NR2	3	$2.634964e - 20$	-1.40449164821534122600	$2.526930e - 07$	3
TM	3	$2.820318e - 22$	-1.40449164821534122600	$3.920088e - 06$	4
MHM	3	$5.741591e - 44$	-1.40449164821534122600	$3.100243e - 09$	5
Algorithm 1	2	$7.567005e - 19$	-1.40449164821534122600	$1.391725e - 03$	6
Algorithm 2	2	$5.794073e - 16$	-1.40449164821534122600	$3.524255e - 03$	2

TABLE 6: Continued.

Method	N	$ f(x_{j+1}) $	x_{j+1}	$\sigma = x_{j+1} - x_j $	COC
$f_{11}(x), x_0 = 2.0$					
NR1	4	$2.455690e - 16$	0.00000000000000000000	$9.031605e - 06$	2
NR2	4	$3.424681e - 40$	0.00000000000000000000	$1.271327e - 13$	3
TM	3	$7.907266e - 122$	0.00000000000000000000	$5.282400e - 14$	4
MHM	3	$1.556811e - 66$	0.00000000000000000000	$6.000887e - 10$	5
Algorithm 1	2	$2.328631e - 21$	0.00000000000000000000	$8.308383e - 03$	6
Algorithm 2	2	$3.198129e - 15$	0.00000000000000000000	$3.593833e - 02$	2

TABLE 7: Comparison of the iterations consumed by different algorithms for the accuracy $\epsilon = 10^{-100}$.

Function with initial guess	NM1	NM2	TM	Method MHM	Algorithm 1	Algorithm 2
$f_1(x), x_0 = 2050.0$	09	05	04	04	03	03
$f_2(x), x_0 = 0.1$	09	06	06	06	06	05
$f_3(x), x_0 = 2.0$	07	05	04	04	03	03
$f_4(x), x_0 = 0.4$	08	05	04	05	03	04
$f_5(x), x_0 = -2.0$	12	09	27	05	04	04
$f_6(x), x_0 = 2.0$	08	06	05	04	03	03
$f_7(x), x_0 = 1.2$	09	06	04	04	03	03
$f_8(x), x_0 = 1.5$	144	06	04	04	03	03
$f_9(x), x_0 = 0.6$	80	06	04	04	03	03
$f_{10}(x), x_0 = -3.0$	09	05	05	04	03	04
$f_{11}(x), x_0 = 2.0$	06	05	03	04	03	03

Example 5. Transcendental and algebraic problems. To numerically analyze the suggested algorithms, we consider the following seven transcendental and algebraic equations:

$$\begin{aligned}
 f_5(x) &= e^x + \cos(\pi x) + x, x_0 = -2.0, \\
 f_6(x) &= x^2 + \sin\left(\frac{x}{5}\right) - \frac{1}{4}, x_0 = 2.0, \\
 f_7(x) &= \ln x + x, x_0 = 1.2, \\
 f_8(x) &= x^3 - 10, x_0 = 1.5, \\
 f_9(x) &= x^3 + x^2 - 2, x_0 = 0.6, \\
 f_{10}(x) &= \sin^2 x - x^2 + 1, x_0 = -3.0, \\
 f_{11}(x) &= \tan^{-1}(x) + x, x_0 = 2.0,
 \end{aligned}
 \tag{41}$$

and their numerical results can be seen in Table 6.

Tables 2–6 exhibit the numerical comparison of the suggested algorithms with other similar-nature existing algorithms. In the columns of the above presented tables, N represents the iterations consumed by different algorithms, $|f(x)|$ denotes the absolute value of $f(x)$ at final approximation, x_{j+1} shows the final approximated root, $|x_{j+1} - x_j|$ represents the absolute distance between the two consecutive

approximations, and (COC) denotes the computational order of convergence having the following approximated formula:

$$\text{COC} \approx \frac{\ln(|x_{j+1} - \alpha|/|x_j - \alpha|)}{\ln(|x_j - \alpha|/|x_{j-1} - \alpha|)}. \tag{42}$$

The above approximation was firstly suggested in 2000 by Weerakoon and Fernando [34]. When we look at the numerical results of Tables 2–6, we come to know that the presented methods are showing best performance as compared to the other ones. For example, in second, fourth, fifth, tenth, and eleventh test examples, Algorithm 1 is the best as it took less number of iterations among the all other compared methods with great precision. In the seventh test example, Algorithm 2 showing the best performance than the other ones whereas in first, third, sixth, eighth, and ninth test examples, both proposed algorithms behave alike and looks better than all the other ones. In short, we can say that the proposed algorithms are superior in terms of accuracy, speed, number of iterations, and computational order of convergence to the other well-known existing iteration schemes.

Table 7 exhibits the comparison of the iterations consumed by different algorithms with the newly proposed

methods for the root-finding of nonlinear algebraic functions with the accuracy $\varepsilon = 10^{-100}$. Here, the columns of the table denote the iterations' number for various test functions together with the initial guess x_0 . The numerical results as shown in Table 7 again certified the fast and best performance of the presented algorithms in terms of number of iterations for the above defined stopping criteria with the given accuracy. In all test examples, the proposed algorithms consumed less number of iterations in comparison with the other iterative algorithms. We did all the calculations with the aid of the computer program Maple 13.

6. Concluding Remarks

In this work, two novel iteration schemes for computing the zeros of nonlinear functions have been established which possess the sextic convergence. The first iteration scheme is derived using the Taylor's series expansion and generalized Newton-Raphson's method whereas in second one, we apply the basic idea of interpolation technique for approximating second derivative which results higher efficiency index. A comparison table regarding the efficiency indices of different methods of the similar nature has been presented which shows that the presented method has higher efficiency index among the other compared methods. By solving some engineering and arbitrary test problems with the aid of computer program, the validity and applicability of the suggested iteration schemes have been analyzed. The numerical results of the Tables 1–7 certified the superiority of the suggested iteration schemes to the other existing two-step iteration schemes of the similar nature. Using the basic idea of interpolation technique, one can derive a broad range of new iteration schemes for computing zeros of one-dimensional nonlinear equations.

Data Availability

All data required for this paper is including within this paper.

Conflicts of Interest

The authors do not have any conflicts.

Authors' Contributions

All authors contribute equally in this paper.

References

- [1] C. Chun, "Construction of Newton-like iteration methods for solving nonlinear equations," *Numerische Mathematik*, vol. 104, no. 3, pp. 297–315, 2006.
- [2] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks/Cole Publishing Company, California, USA, 6th edition, 1997.
- [3] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, USA, 3rd edition, 2002.
- [4] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Springer-Verlag, New York, USA, 2000.
- [5] I. K. Argyros, "A note on the Halley method in Banach spaces," *Applied Mathematics and Computation*, vol. 58, pp. 215–224, 1993.
- [6] J. F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea Publishing company, New York, USA, 1982.
- [7] J. M. Gutierrez and M. A. Hernandez, "A family of Chebyshev-Halley type methods in Banach spaces," *Bulletin of the Australian Mathematical Society*, vol. 55, no. 1, pp. 113–130, 1997.
- [8] S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, USA, 1970.
- [9] A. Amiri, A. Cordero, M. T. Darvishi, and J. R. Torregrosa, "A fast algorithm to solve systems of nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 354, pp. 242–258, 2019.
- [10] W. Nazeer, S. M. Kang, M. Tanveer, and A. A. Shahid, "Fixed point results in the generation of Julia and Mandelbrot sets," *Journal of Inequalities and Applications*, vol. 2015, no. 1, 2015.
- [11] M. A. Noor, K. I. Noor, and K. Aftab, "Some new iterative methods for solving nonlinear equations," *World Applied Sciences Journal*, vol. 20, no. 6, pp. 870–874, 2012.
- [12] A. Özyapici, "Effective numerical methods for non-linear equations," *International Journal of Applied and Computational Mathematics*, vol. 6, no. 2, p. 35, 2020.
- [13] D. Kumar, J. R. Sharma, and I. K. Argyros, "Optimal one-point iterative function free from derivatives for multiple roots," *Mathematics*, vol. 8, no. 5, p. 709, 2020.
- [14] Y. C. Kwun, A. A. Shahid, W. Nazeer, M. Abbas, and S. M. Kang, "Fractal generation via CR iteration scheme with S-convexity," *IEEE Access*, vol. 7, pp. 69986–69997, 2019.
- [15] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Higher-order root-finding algorithms and their basins of attraction," *Journal of Mathematics*, vol. 2020, Article ID 5070363, 11 pages, 2020.
- [16] W. Nazeer, M. Tanveer, S. M. Kang, and A. Naseem, "A new Householder's method free from second derivatives for solving nonlinear equations and polynomiography," *Journal of Nonlinear Sciences and Applications*, vol. 9, no. 3, pp. 998–1007, 2016.
- [17] S. M. Kang, W. Nazeer, M. Tanveer, and A. A. Shahid, "New fixed point results for fractal generation in Jungck Noor orbit with ϕ -convexity," *Journal of Function Spaces*, vol. 2015, Article ID 963016, 7 pages, 2015.
- [18] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Some new iterative algorithms for solving one-dimensional non-linear equations and their graphical representation Some New Iterative Algorithms for Solving One-Dimensional Non-Linear Equations and Their Graphical Representation," *IEEE Access*, vol. 9, pp. 8615–8624, 2021.
- [19] P. B. Chand, F. I. Chicharro, N. Garrido, and P. Jain, "Design and complex dynamics of Potra-Pták-type optimal methods for solving nonlinear equations and its applications," *Mathematics*, vol. 7, no. 10, p. 942, 2019.
- [20] R. Behl, M. Salimi, M. Ferrara, S. Sharifi, and S. K. Alharbi, "Some real-life applications of a newly constructed derivative free iterative scheme," *Symmetry*, vol. 11, no. 2, p. 239, 2019.
- [21] R. Imin and A. Iminjan, "A new SPH iterative method for solving nonlinear equations," *International Journal of Computational Methods*, vol. 15, no. 3, pp. 1–9, 2018.
- [22] M. A. Noor, W. A. Khan, and A. Hussain, "A new modified Halley method without second derivatives for nonlinear equation," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1268–1273, 2007.

- [23] M. A. Hafiz and S. M. H. Al-Goria, "New ninth and seventh order methods for solving nonlinear equations," *European Scientific Journal*, vol. 8, no. 27, pp. 83–95, 2012.
- [24] W. Nazeer, A. Naseem, S. M. Kang, and Y. C. Kwun, "Generalized Newton Raphson's method free from second derivative," *Journal of Nonlinear Sciences and Applications*, vol. 9, no. 5, pp. 2823–2831, 2016.
- [25] A. Kumar, P. Maroju, R. Behl, D. K. Gupta, and S. S. Motsa, "A family of higher order iterations free from second derivative for nonlinear equations in R," *Journal of Computational and Applied Mathematics*, vol. 330, pp. 215–224, 2018.
- [26] M. Salimi, N. M. A. NikLong, S. Sharifi, and B. A. Pansera, "A multi-point iterative method for solving nonlinear equations with optimal order of convergence," *Japan Journal of Industrial and Applied Mathematics*, vol. 35, no. 2, pp. 497–509, 2018.
- [27] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Numerical algorithms for finding zeros of nonlinear equations and their dynamical aspects," *Journal of Mathematics*, vol. 2020, Article ID 2816843, 11 pages, 2020.
- [28] P. Sebah and X. Gourdon, "Newton's method and high order iterations," Technical Report, 2001, <http://numbers.computation.free.fr>.
- [29] M. Shacham, "An improved memory method for the solution of a nonlinear equation," *Chemical Engineering Science*, vol. 44, no. 7, pp. 1495–1501, 1989.
- [30] M. Shacham and E. Kehat, "Converging interval methods for the iterative solution of a non-linear equation," *Chemical Engineering Science*, vol. 28, no. 12, pp. 2187–2193, 1973.
- [31] G. V. Balaji and J. D. Seader, "Application of interval Newton's method to chemical engineering problems," *Reliable Computing*, vol. 1, no. 3, pp. 215–223, 1995.
- [32] V. D. Waals and J. Diderik, *Over de Continuïteit van den Gas-en Vloeïstoestand (On the Continuity of the Gas and Liquid State) [Ph.D. thesis]*, Doctoral dissertation, Leiden, The Netherlands, 1873.
- [33] R. Manning, "On the flow of water in open channels and pipes," *Transactions of the Institution of Civil Engineers of Ireland*, vol. 20, pp. 161–207, 1891.
- [34] S. Weerakoon and T. G. I. Fernando, "A variant of Newton's method with accelerated third-order convergence," *Applied Mathematics Letters*, vol. 13, no. 8, pp. 87–93, 2000.