Hindawi
Complexity
Volume 2022, Article ID 4331899, 17 pages
https://doi.org/10.1155/2022/4331899

Research Article

Some Recent Modifications of Fixed Point Iterative Schemes for Computing Zeros of Nonlinear Equations

Gul Sana,[1] **Muhammad Aslam Noor**,[1] **Mahmood Ul Hassan,**[1] **and Zakia Hammouch**[2,3,4]

[1]*Department of Mathematics, COMSATS University, Islamabad, Pakistan*
[2]*Division of Applied Mathematics, Thu Dau Mo University, Thu Dau Mot 75000, Vietnam*
[3]*Ecole Normale Supéerieure de Meknés, Université Moulay Ismail, Meknes 50000, Morocco*
[4]*Department of Medical Research, China Medical University Hospital, Taichung, Taiwan*

Correspondence should be addressed to Zakia Hammouch; hammouch_zakia@tdmu.edu.vn

Received 10 September 2021; Revised 14 January 2022; Accepted 18 January 2022; Published 31 March 2022

Academic Editor: Sigurdur F. Hafstein

In computational mathematics, it is a matter of deep concern to recognize which of the given iteration schemes converges quickly with lesser error to the desired solution. Fixed point iterative schemes are constructed to be used for solving equations emerging in many fields of science and engineering. These schemes reformulate a nonlinear equation $f(s) = 0$ into a fixed point equation of the form $s = g(s)$; such application determines the solution of the original equation via the support of fixed point iterative method and is subject to existence and uniqueness. In this manuscript, we introduce a new modified family of fixed point iterative schemes for solving nonlinear equations which generalize further recursive methods as particular cases. We also prove the convergence of our suggested schemes. We also consider some of the mathematical models which are categorically nonlinear in essence to authenticate the performance and effectiveness of these schemes which can be seen as an expansion and rationalization of some existing techniques.

1. Introduction

In many disciplines of engineering and mathematical sciences, a broader class of problems are studied in the framework of a nonlinear equation $f(s) = 0$, where $f\colon D\ \ R \longrightarrow R$ is a sufficiently smooth function close to simple zero $\mu \in D$. The development of various iterative methods for finding the approximate solution of a nonlinear equation $f(s) = 0$ has become an active area of research in many scientific fields. Several numerical techniques such as Taylor series, decomposition methods, quadrature formulas, modified homotopy perturbation method, and variational iterative method have been used to explore the diversity of iterative methods, for example, see [1–30]. One of the most well-known and extensively used iterative methods for solving the nonlinear equation is Newton's method which was initiated by Traub [31]. Many numerical methods have been constructed as an extension of Newton and Newton's like methods. Weerakoon and Fernando [30] have improved the convergence of the Newton method by approximating the indefinite integral in Newton's theorem by the rectangular and trapezoidal rule. Frontini and Sormani [13] have extended their results to determine another variant of the Newton method which is cubically convergent. Later on, Ozban [25] has introduced some newly improved forms of the Newton's method by using the concept of harmonic mean and midpoint rule. Abbasbandy [1], Chun [7], and Darvishi and Barati [10] have constructed and introduced different higher-order iterative methods by applying the decomposition technique of Adomian [32]. Implementation of this Adomian decomposition technique required higher-order derivatives evaluation which is a major pitfall of this method. To overcome this drawback, Daftardar-Gejji and Jafari [11] have used different modifications of the Adomian decomposition method and introduced a simple technique which does not require derivative evaluation of the Adomian polynomial. This technique is useful to rewrite the nonlinear equation as the sum of linear and nonlinear parts. Bhalekar and Daftardar-Gejji [6] have determined the convergence of the former method [11] in detail and established its equivalence with Adomian decomposition method. Numerous researchers are extensively using technique [11] and derived several higher-order iterative methods for solving nonlinear equations. Ali et al. [2] have introduced a family of iterative methods using the quadrature formula as well as the fundamental theorem of calculus and checked the validity and performance of these methods by examining two mathematical models. In [18, 20], decomposition technique [11] is implemented and merged sophistically with a coupled system of equation to investigate various order iterative methods. Alharbi et al. [4] have used the central idea of the decomposition technique along with auxiliary function and investigated generalized and comprehensive forms of higher-order iterative methods for finding solutions of nonlinear equations. Ali et al. [3] have constructed several new iterative methods by using Taylor's series expansion of the function $g(s)$. These iterative methods can be viewed as a generalization of some well-known methods such as the Newton method, Halley method, and Traub's method.

Inspired and motivated by the continuing research ventures in this direction, we consider the well-known fixed point iterative method [33], in which we rewrite the nonlinear equation $f(s) = 0$ as $s = g(s)$ that satisfies the following properties for existence and uniqueness of fixed point:

(i) If $g(s) \in C[a,b]$ and $g(s) \in [a,b]$ for all $s \in [a, b]$, then g has at least one fixed point in $[a,b]$.

(ii) If, in addition, $g'(s)$ exists on (a, (b) and a positive constant L exists, such that

(iii) $|g'(x)| \leq L < 1$, for all $s \in (a,b)$, then there is exactly one fixed point in $[a,b]$.

Then, we use Newton's theorem along with writing the functional equation $s = g(s)$ as coupled system and applying the decomposition technique presented in [11]. In the second section of this study, we introduce some new iterative methods and determine their special cases. The third section comprises of convergence analysis of the proposed iterative methods. Efficiency and performance of newly constructed family of recursive approaches are tested in the last section, by solving some test examples along with two models, i.e., motion of a particle on an inclined plane and Lenard–Jones potential applications to minimization problem. We also present the graphical analysis for these models. Numerical results of the examples reveal and validate the efficacy of our newly proposed methods.

2. New Class of Family of Iterative Schemes

Consider the nonlinear equation

$$f(s) = 0, \quad s \in R, \tag{1}$$

which can be rewritten as

$$s = g(s), \quad s \in R, \tag{2}$$

where λ is an initial guess sufficiently close to μ which is simple root of (1). Now, utilizing the technique of Noor et al. [22], approximate the function $f(s)$ using fundamental theorem of calculus and quadrature formula:

$$s = g(\lambda) + (s-\lambda)\sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i(s-\lambda)), \tag{3}$$

where τ_i represents the knots and $\tau_i \in [0,1]$ and $\widetilde{\omega}_i$ satisfy the condition:

$$\sum_{i=1}^{\kappa} \widetilde{\omega}_i = 1. \tag{4}$$

Applying the technique of He [15], and writing the nonlinear equation as an equivalent coupled system of equations,

$$s = g(\lambda) + (s-\lambda)\sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i(s-\lambda)) + H(s), \tag{5}$$

$$H(s) = g(s) - g(\lambda) - (s-\lambda)\sum_{i=1}^{\kappa} w_i g'(\lambda + \tau_i(s-\lambda)). \tag{6}$$

It can be rewritten as

$$\begin{aligned} s = {} & \frac{H(s)}{1 - \sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda + \tau_i(s-\lambda))} \\ & + \frac{g(\lambda) - \lambda\sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda + \tau_i(s-\lambda))}{1 - \sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda + \tau_i(s-\lambda))}, \end{aligned} \tag{7}$$

$$s = c + \mathscr{M}(s), \tag{8}$$

where

$$c = \lambda, \tag{9}$$

$$\begin{aligned} \mathscr{M}(s) = {} & \frac{H(s)}{1 - \sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda + \tau_i(s-\lambda))} \\ & + \frac{g(\lambda) - \lambda}{1 - \sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda + \tau_i(s-\lambda))}. \end{aligned} \tag{10}$$

It is clear that $\mathscr{M}(s)$ is nonlinear operator. Now, we construct sequence of higher-order iterative methods by employing decomposition technique initiated by Gejji and Jafari [11]. With the support of this technique, solution of (8) can be represented as in terms of the infinite series:

$$s = \sum_{r=0}^{\infty} s_r. \tag{11}$$

The nonlinear operator can be decomposed as

$$\mathscr{M}(s)=\mathscr{M}(s_0)+\sum_{r=1}^{\infty}\left\{\mathscr{M}\left(\sum_{q=0}^{r}s_q\right)-\mathscr{M}\left(\sum_{q=0}^{r-1}s_q\right)\right)\right\}. \tag{12}$$

Thus, from equations (8), (11), and (12), we have

$$\sum_{r=0}^{\infty}s_r=c+\mathscr{M}(s_o)+\sum_{r=1}^{\infty}\left\{\mathscr{M}\left(\sum_{q=0}^{r}s_q\right)-\mathscr{M}\left(\sum_{q=0}^{r-1}s_q\right)\right\}, \tag{13}$$

which generates the following iterative scheme:

$$\begin{cases} s_o=c,\\ s_1=M(s_0),\\ s_2=M(s_0+s_1)-M(s_0),\\ .\\ .\\ .\\ s_{m+1}=M\left(\sum_{q=0}^{m}s_q\right)-M\left(\sum_{q=0}^{m-1}s_q\right), \quad m=1,2,\ldots. \end{cases} \tag{14}$$

It follows that

$$s_1+s_2+\cdots s_{m+1}=\mathscr{M}(s_0+s_1+\cdots s_m), \quad m=1,2,\ldots,$$
$$s=c+\sum_{r=1}^{\infty}s_r. \tag{15}$$

Notable approximation of s is conveyed as

$$\lim_{m\longrightarrow\infty}S_m=s,$$
$$S_m=s_o+s_1+\cdots s_m, \tag{16}$$

For $m=0$, we have

$$\mathbf{s}\approx\mathbf{S}_0=\mathbf{s}_0=\mathbf{c}=\lambda. \tag{17}$$

Implementing (10),

$$s_1=\mathscr{M}(s_0)=\frac{H(s_0)}{1-\sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda+\tau_i(s_o-\lambda))}$$
$$+\frac{g(\lambda)-\lambda}{1-\sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda+\tau_i(s_o-\lambda))}. \tag{18}$$

From (6), it can easily be computed as $H(s_0)=0$ and using in (18), we get

$$=\frac{g(\lambda)-\lambda}{1-\sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda+\tau_i(s_o-\lambda))}. \tag{19}$$

For $m=1$,

$$\mathbf{s}\approx\mathbf{S}_1=\mathbf{s}_0+\mathbf{s}_1=\mathbf{s}_0+\mathscr{M}(\mathbf{s}_0),$$
$$s=s_0+s_1+\frac{g(\lambda)-\lambda}{1-\sum_{i=1}^{\kappa}\widetilde{\omega}_i g'(\lambda+\tau_i(s_o-\lambda))}. \tag{20}$$

Using (4) and (17), we have

$$s=\frac{g(\lambda)-\lambda g'(\lambda)}{1-g'(\lambda)}. \tag{21}$$

This formulation suggests Algorithm 1.

Algorithm 2 and Algorithm 3 are the main iterative schemes which generate further special cases for different values k, w, and τ.

2.1. Some Particular Manifestations of Algorithm 2.

Now, we explore the particular cases of Algorithm 2, by considering different values of κ, w, and τ. Taking $\kappa=1, \widetilde{\omega}_1=1$, and $\tau_1=0$, Algorithm 2 turns down to the subsequent iterative scheme:

2.2. Some Special Manifestations of Algorithm 3.

Picking $\kappa=1, \widetilde{\omega}_1=1$, and $\tau_1=0$, Algorithm 3 turns down to the subsequent iterative scheme:

To the best of our knowledge, Algorithm 7, Algorithm 8, Algorithm 9, Algorithm 11, and Algorithm 12 appear to be new ones.

3. Convergence Analysis

Theorem 1. *Let $f\colon I\subseteq R\longrightarrow R$ be differentiable function where I is an open interval. Let μ□ $\in I$ be a simple zero of $f(s)=0$ or $s=g(s)$ where $g\colon I\subseteq R\longrightarrow R$ is sufficiently smooth in the neighborhood of root. If s_o is an initial guess existing nearly close to μ, then the multistep methods defined by the Algorithms 2, 3, 7, 8, 9, 11, and 12 have convergence of order at least 3, 4, 3, 3, 4, 4, and 4, respectively.*

Proof. Let μ be root of nonlinear equation $f(s)=0$ or equivalently $s=g(s)$. Let the errors at nth and $(n+1)$th iterations be e_n and e_{n+1}, respectively.

Now, expanding $g(s)$ and $g'(s_n)$ by using Taylor's series about μ,

(i) For a given s_0, approximate solution s_{n+1} is computed by the following iterative scheme: $S_{n+1} = ((g(s_n) - s_n g'(s_n)) / (1 - g'(s_n)))$, $g'(s_n) \neq 1, n = 0, 1, 2$.

(ii) Kang et al. [20] developed this algorithm which has quadratic convergence. From (6) and (8), we have $H(s_0 + s_1) = g(s_0 + s_1) - g(\lambda) - (s_0 + s_1 - \lambda) \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_0 + s_1 - \lambda))$,
$\mathscr{M}(s_0 + s_1) = (H(s_0 + s_1)/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 - \lambda)))) + ((g(\lambda) - \lambda)/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 - \lambda))))$.

(iii) **For m** $= 2$, $\mathbf{s} \approx \mathbf{S}_2 = \mathbf{s}_0 + \mathbf{s}_1 + \mathbf{s}_2 = \mathbf{c} + \mathscr{M}(\mathbf{s}_0 + \mathbf{s}_1)$.

(iv) Employing (17), Algorithm 1, and simplifying, we have $s = \lambda - (\lambda/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 - \lambda)))) + (g(s_0 + s_1)/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 - \lambda))))-, (((s_0 + s_1 - \lambda) \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 - \lambda)))/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 - \lambda))))$.

(v) From (19), we take $s_0 + s_1 = t = ((g(\lambda) - \lambda g'(\lambda))/(1 - g'(\lambda)))$.

(vi) By using Algorithm 1 and computing, we get $s = (g(t)/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (t - \lambda)))) - (t \sum_{i=1}^{\kappa} \widetilde{\omega}_i g' (\lambda + \tau_i (t - \lambda)) / (1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (t - \lambda))))$.

(vii) This relation yields the following two-step algorithm.

Algorithm 1: The proposed method.

(i) For a given s_0, approximate solution s_{n+1} is computed by the following iterative scheme $t_n = (g(s_n) - s_n g'(s_n))/(1 - g'(s_n))$, $s_{n+1} = ((g(t_n) - t_n \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(s_n + \tau_i (t_n - s_n)))/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(s_n + \tau_i (t_n - s_n))))$, $n = 0, 1, 2$.

(ii) It is noted that $s_0 + s_1 + s_2 = u = (g(t)/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (t - \lambda)))) - (t \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (t - \lambda))/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (t - \lambda))))$.

(iii) For $m = 3$, $\mathbf{s} \approx \mathbf{S}_3 = \mathbf{s}_0 + \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{s}_3 = \mathbf{c} + \mathscr{M}(\mathbf{s}_0 + \mathbf{s}_1 + \mathbf{s}_2)$.

(iv) Using (6), (10), and (17), we have $s = \lambda - s = \lambda(\lambda/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 + s_2 - \lambda)))) + (g(\lambda)/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 + s_2 - \lambda)))) - (((s_0 + s_1 + s_2 - \lambda) \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 + s_2 - \lambda)))/(1 - \sum_{i=1}^{\kappa} \widetilde{\omega}_i g'(\lambda + \tau_i (s_o + s_1 + s_2 - \lambda))))$.

(v) Applying Algorithm 2, we get $s = \lambda - (\lambda/(1 - \sum_{i=1}^{k} \widetilde{\omega}_i g'(\lambda + \tau_i (u - \lambda)))) + (g(\lambda)/(1 - \sum_{i=1}^{k} \widetilde{\omega}_i g'(\lambda + \tau_i (u - \lambda)))) - (((z - \lambda) \sum_{i=1}^{k} \widetilde{\omega}_i g'(\lambda + \tau_i (u - \lambda)))/(1 - \sum_{i=1}^{k} \widetilde{\omega}_i g'(\lambda + \tau_i (u - \lambda)))), = (g(u)/(1 - \sum_{i=1}^{k} \widetilde{\omega}_i g'(\lambda + \tau_i (u - \lambda)))) - (u \sum_{i=1}^{k} \widetilde{\omega}_i g'(\lambda + \tau_i (u - \lambda))/(1 - \sum_{i=1}^{k} \widetilde{\omega}_i g'(\lambda + \tau_i (u - \lambda))))$.

(vi) This formulation yields the following three-step method for solving nonlinear (1).

Algorithm 2: Kang's method.

(i) Let s_0 be an initial guess, then one can figure out s_{n+1} (approximate solution) with the support of the subsequent recursive scheme: $t_n = ((g(s_n) - s_n g'(s_n))/(1 - g'(s_n)))$, $u_n = (g(s_n) - t_n \sum_{i=1}^{k} \widetilde{\omega}_i g'(s_n + \tau_i (t_n - s_n)))/(1 - \sum_{i=1}^{k} \widetilde{\omega}_i g'(s_n + \tau_i (t_n - s_n)))$, $s_{n+1} = ((g(u_n) - u_n \sum_{i=1}^{k} \widetilde{\omega}_i g'(s_n + \tau_i (t_n - s_n)))/(1 - \sum_{i=1}^{k} \widetilde{\omega}_i g'(s_n + \tau_i (t_n - s_n))))$, $n = 0, 1, 2$.

Algorithm 3: The proposed method.

(i) Let s_0 be an initial guess, then one can figure out the approximate solution s_{n+1} by the following recursive method: $t_n = ((g(s_n) - s_n g'(s_n))/(1 - g'(s_n)))$, $s_{n+1} = ((g(t_n) - t_n g'(s_n))/(1 - g'(s_n)))$, $n = 0, 1, 2$.

(ii) This algorithm was established by Kwun et al. [21] which has third-order convergence.

(iii) Taking $\kappa = 1, \widetilde{\omega}_1 = 1$, and $\tau_1 = 1$, Algorithm 2 minimizes to the succeeding iterative scheme.

Algorithm 4: Kwun's method.

(i) Let s_0 be an initial guess, then the approximate solution s_{n+1} by the following recursive method: $t_n = (g(s_n) - s_n g'(s_n))/(1 - g'(s_n))$, $s_{n+1} = ((g(t_n) - t_n g'(t_n))/(1 - g'(t_n)))$, $n = 0, 1, 2$.

(ii) Saqib et al. [27] have derived this algorithm which has fourth-order convergence.

(iii) Taking $\kappa = 1, \widetilde{\omega}_1 = 1$, and $\tau_1 = 1/2$, Algorithm 2 turns down to the subsequent iterative scheme.

Algorithm 5: Saqib's method.

(i) Let s_0 be an initial guess, then one can figure out the approximate solution s_{n+1} by the following recursive technique: $t_n = (g(s_n) - s_n g'(s_n))/(1 - g'(s_n)), s_{n+1} = ((g(t_n) - t_n g'((s_n + t_n)/2))/(1 - g'((s_n + t_n)/2))), n = 0, 1, 2.$
(ii) Gul et al. [29] investigated this algorithm which has third-order convergence.
(iii) Choosing $\kappa = 2, \widetilde{w}_1 = 1/4, \widetilde{w}_2 = 3/4, \tau_1 = 0$, and $\tau_2 = 2/3$, Algorithm 2 is restructured to the following recursive method.

Algorithm 6: Gul's method (1).

(i) Let s_0 be an initial guess, then one can figure out the approximate solution s_{n+1} by the following recursive scheme: $t_n = (g(s_n) - s_n g'(s_n))/(1 - g'(s_n)), s_{n+1} = (4g(t_n) - t_n\{g'(s_n) + 3g'((s_n + 2t_n)/3)\})/(4 - \{g'(s_n) + 3g'((s_n + 2t_n)/3)\}), n = 0, 1, 2.$
(ii) Taking $\kappa = 2, \widetilde{w}_1 = 1/2, \widetilde{w}_2 = 1/2, \tau_1 = 0$, and $\tau_2 = 1/2$, Algorithm 2 turns down to the subsequent iterative scheme.

Algorithm 7: A recent iterative method.

(i) Let s_0 be an initial guess, then the approximate solution s_{n+1} is computed by the following iterative scheme: $t_n = (g(s_n) - s_n g'(s_n))/(1 - g'(s_n)), s_{n+1} = ((2g(t_n) - t_n\{g'(s_n) + g'((s_n + t_n)/2)\})/(2 - \{g'(s_n) + g'((s_n + t_n)/2)\})), n = 0, 1, 2.$

Algorithm 8: A newly designed scheme.

(i) For a given s_0, compute approximate solution s_{n+1} by the following iterative scheme: $t_n = (g(s_n) - s_n g'(s_n))/(1 - g'(s_n)), u_n = ((g(t_n) - t_n g'(s_n))/(1 - g'(s_n))), s_{n+1} = ((g(u_n) - u_n g'(s_n))/(1 - g'(s_n))), n = 0, 1, 2.$
(ii) Taking $\kappa = 1, \widetilde{w}_1 = 1$, and $\tau_1 = 1/2$, Algorithm 3 reduces to the following recursive approach as follows.

Algorithm 9: The constructed method.

(i) Let s_0 be an initial guess, then the approximate solution s_{n+1} is computed by the following recursive method: $t_n = (g(s_n) - s_n g(s_n))/(1 - g'(s_n)), u_n = (g(t_n) - t_n g'((s_n + t_n)/2))/(1 - g'((s_n + t_n)/2)), s_{n+1} = ((g(u_n) - u_n g'((s_n + u_n)/2))/(1 - g'((s_n + u_n)/2))), n = 0, 1, 2.$
(ii) Gul et al. [29] suggested this algorithm which has fourth-order convergence.
(iii) Choosing $\kappa = 2, \widetilde{w}_1 = 1/4, \widetilde{w}_2 = 3/4, \tau_1 = 0$, and $\tau_2 = 2/3$, Algorithm 3 reduces to the subsequent iterative method.

Algorithm 10: Gul's method (2).

(i) Let s_0 be an initial guess, then the approximate solution s_{n+1} is computed by the following iterative scheme: $t_n = ((g(s_n) - s_n g'(s_n))/(1 - g'(s_n))), u_n = (4g(t_n) - t_n\{g'(s_n) + 3g'((s_n + 2t_n/3)\})/(4 - \{g'(s_n) + 3g'((s_n + 2t_n)/3)\}),$
$s_{n+1} = (4g(u_n) - u_n\{g'(s_n) + 3g'((s_n + 2u_n/3)\})/(4 - \{g'(s_n) + 3g'((s_n + 2u_n/3)\}), n = 0, 1, 2.$
(ii) Choosing $\kappa = 2, \widetilde{w}_1 = 1/2, \widetilde{w}_2 = 1/2, \tau_1 = 0$, and $\tau_2 = 1/2$, Algorithm 3 turns down to the subsequent iterative method.

Algorithm 11: A novel iterative method.

(i) Let s_0 be an initial guess, then one can figure out the approximate solution s_{n+1} by the following recursive scheme: $t_n = (g(s_n) - s_n g'(s_n))/(1 - g'(s_n)), u_n = (2g(t_n) - t_n\{g'(s_n) + g'(s_n + t_n/2)\})/2 - \{g'(s_n) + g'(s_n + t_n/2)\},$
$s_{n+1} = ((2g(u_n) - u_n\{g'(s_n) + g'((s_n + u_n)/2)\})/(2 - \{g'(s_n) + g'((s_n + u_n)/2)\})), n = 0, 1, 2.$

Algorithm 12: A new recursive approach.

$$g\left(s_n\right) = \mu + e_n g'\left(\mu\right) + \frac{e_n^2}{2} g''\left(\mu\right) + \frac{e_n^3}{6} g'''\left(\mu\right) + O\left(e_n^4\right), \tag{22}$$

$$g'\left(s_n\right) = g'\left(\mu\right) + e_n g''\left(\mu\right) + \frac{e_n^2}{2} g'''\left(\mu\right) + \frac{e_n^3}{6} g^{iv}\left(\mu\right) + O\left(e_n^4\right), \tag{23}$$

$$1 - g'\left(s_n\right) = 1 - g'\left(\mu\right) - e_n g''\left(\mu\right) - \frac{e_n^2}{2} g'''\left(\mu\right) - \frac{e_n^3}{6} g^{iv}\left(\mu\right) + O\left(e_n^4\right), \tag{24}$$

$$g\left(s_n\right) - s_n g\left(s_n\right) = \mu - \mu g'\left(\mu\right) - \mu g''\left(\mu\right) e_n - \frac{1}{2}\left(g''\left(\mu\right) + \mu g'''\left(\mu\right)\right) e_n^2 + \frac{1}{6}\left(2g'''\left(\mu\right) + \mu g^{iv}\left(\mu\right)\right) e_n^3 + O\left(e_n^4\right), \tag{25}$$

$$\frac{g\left(s_n\right) - s_n g\left(s_n\right)}{1 - g'\left(s_n\right)} = \mu + \frac{g''\left(\mu\right)}{2\left(-1 + g'\left(\mu\right)\right)} e_n^2 - \left\{\frac{2g'''\left(\mu\right) - 2g'''\left(\mu\right) g'\left(\mu\right) + 3g''^2\left(\mu\right)}{6\left(-1 + g'\left(\mu\right)\right)^2}\right\} e_n^3 + O\left(e_n^4\right). \tag{26}$$

Considering Algorithm 2 and following (26), where $g'\left(\mu\right) \neq 1$, we obtain

$$t_n = \mu + \frac{g''\left(\mu\right)}{2\left(-1 + g'\left(\mu\right)\right)} e_n^2 + \left\{\frac{-2g'''\left(\mu\right) + 2g'''\left(\mu\right) g'\left(\mu\right) - 3g''^2\left(\mu\right)}{6\left(-1 + g'\left(\mu\right)\right)^2}\right\} e_n^3 + O\left(e_n^4\right), \tag{27}$$

$$g\left(t_n\right) = \mu + \frac{g'\left(\mu\right) g''\left(\mu\right)}{2\left(-1 + g'\left(\mu\right)\right)} e_n^2 + \left\{\frac{g'\left(\mu\right)\left(-2g'''\left(\mu\right) + 2g'''\left(\mu\right) g'\left(\mu\right) - 3g''^2\left(\mu\right)\right)}{6\left(-1 + g'\left(\mu\right)\right)^2}\right\} e_n^3 + O\left(e_n^4\right). \tag{28}$$

Taylor's series expansion of $\sum_{i=1}^{\kappa} \widetilde{w}_i g'\left(s_n + \tau_i\left(t_n - s_n\right)\right)$ about μ,

$$\begin{aligned} &\sum_{i=1}^{\kappa} w_i g'\left(s_n + \tau_i\left(t_n - s_n\right)\right) \\ &= g'\left(\mu\right) - g''\left(\mu\right)\left\{\sum_{i=1}^{\kappa} \widetilde{w}_i\left(-1 + \tau_i\right)\right\} e_n \\ &\quad + \frac{1}{2\left(-1 + g'\left(\mu\right)\right)}\left\{\sum_{i=1}^{\kappa} w_i \begin{pmatrix} \tau_i g''^2\left(\mu\right) - g'''\left(\mu\right) + g'\left(\mu\right) g'''\left(\mu\right) + 2g'''\left(\mu\right)\tau_i \\ -2\tau_i g'''\left(\mu\right) g'\left(\mu\right) - \tau_i^2 g'''\left(\mu\right) + g'''\left(\mu\right)\tau_i^2 g'\left(\mu\right) \end{pmatrix}\right\} e_n^2 + O\left(e_n^3\right). \end{aligned} \tag{29}$$

By substituting (27)–(29) into Algorithm 2, we obtain the error term of Algorithm 2,

$$u_n = \mu - \frac{1}{2\left(-1 + g'\left(\mu\right)\right)^2}\left\{g''^2\left(\mu\right)\left(\sum_{i=1}^{\kappa} \widetilde{w}_i\left(-1 + \tau_i\right)\right)\right\} e_n^3 + O\left(e_n^4\right). \tag{30}$$

Expanding $g\left(u_n\right)$ in terms of Taylor series about μ,

$$g\left(u_n\right) = \mu - \frac{1}{2\left(-1 + g'\left(\mu\right)\right)^2}\left\{g'\left(\mu\right)g''\left(\mu\right)\left(\sum_{i=1}^{\kappa}\widetilde{\omega}_i\left(-1 + \tau_i\right)\right)\right\}e_n^3 + O\left(e_n^4\right). \tag{31}$$

Expanding $\sum_{i=1}^{\kappa}\widetilde{\omega}_i g'\left(s_n + \tau_i\left(u_n - s_n\right)\right)$ by Taylor's series,

$$\begin{aligned} &= \sum_{i-1}^{\kappa}\widetilde{\omega}_i g'\left(s_n + \tau_i\left(u_n - s_n\right)\right) = g'\left(\mu\right) - g''\left(\mu\right)\left\{\sum_{i=1}^{\kappa}\widetilde{\omega}_i\left(-1 + \tau_i\right)\right\}e_n \\ &\quad + \frac{1}{2}g'''\left(\mu\right)\left\{\sum_{i=1}^{\kappa}\left(\widetilde{\omega}_i - 2\widetilde{\omega}_i\tau_i + \widetilde{\omega}_i\tau_i^2\right)\right\}e_n^2 + O\left(e_n^3\right). \end{aligned} \tag{32}$$

Applying (30)–(32) into Algorithm 3, we get the error term of Algorithm 3,

$$\begin{aligned} s_{n+1} &= \mu + \frac{g''^3\left(\mu\right)\{(\sum_{i-1}^{\kappa}\widetilde{\omega}_i\left(-1 + \tau_i\right))\left(\sum_{i-1}^{\kappa}\widetilde{\omega}_i\left(\tau_i - 1\right)\right)\}}{2\left(-1 + g'\left(\mu\right)\right)^3}e_n^4 + O\left(e_n^5\right), \quad g'\left(\mu\right) \neq 1, \\ e_{n+1} &= \frac{g''^3\left(\mu\right)\{(\sum_{i-1}^{\kappa}\widetilde{\omega}_i\left(-1 + \tau_i\right))\left(\sum_{i-1}^{\kappa}\widetilde{\omega}_i\left(\tau_i - 1\right)\right)\}}{2\left(-1 + g'\left(\mu\right)\right)^3}e_n^4 + O\left(e_n^5\right). \end{aligned} \tag{33}$$

Now, we investigate the convergence order of special cases of Algorithms 2 and 3.

Amplifying $g'\left(s_n + 2t_n/3\right)$ in terms of Taylor series about μ,

$$g'\left(\frac{s_n + 2t_n}{3}\right) = g'\left(\mu\right) + +\frac{1}{3}g''\left(\mu\right)e_n + \frac{1}{18\left(-1 + g'\left(\mu\right)\right)}\left\{6g''^2\left(\mu\right) - g'''\left(\mu\right) + g'\left(\mu\right)g'''\left(\mu\right)\right\}e_n^2 + O\left(e_n^3\right). \tag{34}$$

Starting with (23), (27), (29), and (34), we obtain

$$\begin{aligned} &4g\left(t_n\right) - t_n\left\{g'\left(s_n\right) + 3g'\left(\frac{s_n + 2t_n}{3}\right)\right\} \\ &\quad = -4\mu\left(-1 + g'\left(\mu\right)\right) - 2\mu g''\left(\mu\right)e_n - \left\{\frac{\mu\left(-2g'''\left(\mu\right) + 2g'''\left(\mu\right)g'\left(\mu\right) + 3g''^2\left(\mu\right)\right)}{3\left(-1 + g'\left(\mu\right)\right)}\right\}e_n^2 + O\left(e_n^3\right), \end{aligned} \tag{35}$$

$$\begin{aligned} &4 - \left\{g'\left(s_n\right) + 3g'\left(\frac{s_n + 2t_n}{3}\right)\right\} \\ &\quad = 4 - 4g'\left(\mu\right) - 2g''\left(\mu\right)e_n - \left\{\frac{3g''^3\left(\mu\right) + 2g'''\left(\mu\right)g'\left(\mu\right) - 2g'''\left(\mu\right)}{3\left(-1 + g'\left(\mu\right)\right)}\right\}e_n^2 + O\left(e_n^3\right). \end{aligned} \tag{36}$$

Now, by substituting (33) and (34) into Algorithm 7 and simplifying, we obtain the error term of Algorithm 7,

$$s_{n+1} = \mu - \left\{ \frac{g''(\mu)}{4(-1+g'(\mu))^2} \right\} e_n^3 + O(e_n^4), \quad g'(\mu) \neq 1, \tag{37}$$

$$e_{n+1} = \left\{ \frac{g''(\mu)}{4(-1+g'(\mu))^2} \right\} e_n^3 + O(e_n^4). \tag{38}$$

Unfolding $g'(s_n + t_n/2)$ in terms of Taylors' series about μ,

$$g'\left(\frac{s_n+t_n}{2}\right) = g'(\mu) + \frac{1}{2} g''(\mu) e_n + \left\{ \frac{2g''(\mu)^2 - g'''(\mu) + g'''(\mu) g'(\mu)}{8(-1+g'(\mu))} \right\} e_n^2 + O(e_n^3). \tag{39}$$

Employing (23), (27), (28), and (37) and simplifying, we have

$$\begin{aligned} & 2g(t_n) - t_n\left(g'(s_n) + g'\left(\frac{s_n+t_n}{2}\right)\right) - 2 \\ & - 2\mu(-1+g'(\mu)) - \frac{3}{2}\mu(g''(\mu))e_n - \left\{ \frac{\mu\left(-5g'''(\mu) + 5g'''(\mu) g'^{(\mu)} + 2g''^2(\mu)\right)}{8(-1+g'(\mu))} \right\} e_n^2 + O(e_n^3), \end{aligned} \tag{40}$$

$$\begin{aligned} 2 - \left\{ g'(s_n) + g'\left(\frac{s_n+t_n}{2}\right)\right\} = {} & -2 - 2g'(\mu) - \frac{3}{2} g''(\mu) e_n - \frac{1}{8(-1+g'(\mu))} \\ & \left\{ \frac{-5g'''(\mu) + 5g'''(\mu) g'^{(\mu)} + 2g''^2(\mu)}{8(-1+g'(\mu))} \right\} e_n^2 + O(e_n^3). \end{aligned} \tag{41}$$

By substituting (40) and (41) into Algorithm 8, we get the error term of Algorithm 8,

$$s_{n+1} = \mu + \left\{ \frac{g''^2(\mu)}{8(-1+g'(\mu))^2} \right\} e_n^3 + O(e_n^4), \quad g'(\mu) \neq 1, \tag{42}$$

$$e_{n+1} = \left\{ \frac{g''^2(\mu)}{8(-1+g'(\mu))^2} \right\} e_n^3 + O(e_n^4). \tag{43}$$

Now, again considering the error term of Algorithm 4 investigated by Kwun et al. [21],

$$\mu + \frac{(g'(\mu))^2}{2(-1+g'(\mu))^2} u_n = e_n^3 + \left\{ \frac{g''(\mu)\left(14 g'''(\mu) g'(\mu) - 27(g''(\mu))^2 - 14 g'''(\mu)\right)}{24(-1+g'(\mu))^3} \right\} e_n^4 + O(e_n^5). \tag{44}$$

Amplifying $g(u_n)$ in terms of Taylors' series about μ,

$$g(u_n) = \mu + \frac{g'(\mu)g''^2(\mu)}{2(-1+g'(\mu))^2}e_n^3 + \frac{1}{24}\left\{\frac{g'(\mu)g''(\mu)\left(14g'''(\mu)g'(\mu) - 27(g''(\mu))^2 - 14g'''(\mu)\right)}{(-1+g'(\mu))^3}\right\}e_n^4 + O\left(e_n^5\right), \tag{45}$$

$$g(u_n) - u_n g'(s_n) = \mu - \mu g'(\mu) - \mu g''(\mu)e_n - \frac{\mu g'''(\mu)}{2}e_n^2 - \frac{\mu g^{(iv)}(\mu)}{6}e_n^3 - \frac{g''^3(\mu)}{2(-1+g'(\mu))^2}e_n^4 + O\left(e_n^5\right). \tag{46}$$

By substituting (24) and (46) into Algorithm 9, we get the error term of Algorithm 9,

$$s_{n+1} = \mu + \left\{\frac{g''^3(\mu)}{2(-1+g'(\mu))^3}\right\}e_n^4 + O\left(e_n^5\right), \quad g'(\mu) \neq 1, \tag{47}$$

$$e_{n+1} = \left\{\frac{g''^3(\mu)}{2(-1+g'(\mu))^3}\right\}e_n^4 + O\left(e_n^5\right). \tag{48}$$

Now, considering (41) and expanding $g(u_n)$ in terms of Taylor's series about μ,

$$g(u_n) = \mu + \left\{\frac{g''^2(\mu)g'(\mu)}{4(-1+g'(\mu))^2}\right\}e_n^3 + o\left(e_n^4\right). \tag{49}$$

Opening up the term $g'(s_n + 2u_n/3)$ in the form of Taylor's series about μ,

$$g'\left(\frac{s_n + 2u_n}{3}\right) = g'(\mu) + \frac{1}{3}g''(\mu)e_n + \frac{1}{18}g'''(\mu)e_n^2 + \left\{\frac{g''^3(\mu)}{6(-1+g'(\mu))^2} + \frac{g^{iv}(\mu)}{162}\right\}e_n^3 + o\left(e_n^4\right). \tag{50}$$

From (23), (48), and (49) and simplifying, we have

$$\begin{aligned} 4g(u_n) - u_n\left(g'(s_n) + 3g'\left(\frac{(s_n + 2u_n)}{3}\right)\right) &= -4(1 - \mu g'(\mu)) - 2\mu g''(\mu)e_n - \frac{2}{3}\mu g'''(\mu)e_n^2 \\ &\quad + \left\{-\frac{5}{27}\mu g^{iv}(\mu) - \frac{g''^3(\mu)}{2(-1+g'(\mu))^2}\right\}e_n^3 + O\left(e_n^4\right), \end{aligned} \tag{51}$$

$$\begin{aligned} 4 - \left(g'(s_n) + 3g'\left(\frac{(s_n + 2u_n)}{3}\right)\right) &= 4(1 - g'(\mu)) - 2g''(\mu)e_n - \frac{2}{3}g'''(\mu)e_n^2 \\ &\quad + \left\{\frac{5}{27}g^{iv}(\mu) + \frac{g''^3(\mu)}{2(-1+g'(\mu))^2}\right\}e_n^3 + O\left(e_n^4\right). \end{aligned} \tag{52}$$

By substituting (51) and (52) into Algorithm 11, we get the error term of Algorithm 11,

$$s_{n+1} = \mu + \left\{\frac{3g''^3(\mu)}{16(-1+g'(\mu))^3}\right\}e_n^4 + O\left(e_n^5\right), \quad g'(\mu) \neq 1, \tag{53}$$

$$e_{n+1} = \left\{\frac{3g''^3(\mu)}{16(-1+g'(\mu))^3}\right\}e_n^4 + O\left(e_n^5\right). \tag{54}$$

From (42) considering the error equation of Algorithm 8,

$$u_n = \mu - \left\{ \frac{g''^2(\mu)}{8(-1+g'(\mu))^2} \right\} e_n^3 + O\left(e_n^4\right). \tag{55}$$

Expanding $g(u_n)$ in terms of Taylor's series about μ,

$$g(u_n) = \mu + \left\{ \frac{g'(\mu) g'^2(\mu)}{8(-1+g'(\mu))^2} \right\} e_n^3 + \left\{ \frac{g'(\mu) g''(\mu)\left(13 g'''(\mu) g'(\mu) - 13 g'''(\mu) - 23 g''^2(\mu)\right)}{32(-1+g'(\mu))^3} \right\} e_n^4 + O\left(e_n^5\right). \tag{56}$$

Elaborating $g'(s_n + u_n/2)$ in the terminology of Taylor's series about μ,

$$\begin{aligned} g'\left(\frac{s_n+u_n}{2}\right) &= g'(\mu) + \frac{1}{2} g''(\mu) e_n + \frac{1}{8} g'''(\mu) e_n^2 \\ &\quad + \left\{ \frac{9 g''^3(\mu) + g^{iv}(\mu) - 2 g^{iv}(\mu) g'(\mu) + g^{iv}(\mu) g'^2(\mu)}{48(-1+g'(\mu))^2} \right\} e_n^3 + O\left(e_n^4\right). \end{aligned} \tag{57}$$

From (55)–(57), we have

$$\begin{aligned} & 2g(u_n) - u_n\left(g'(s_n) + g'\left(\frac{(x_n+u_n}{2}\right)\right) \\ & = -2\mu(-1+g'(\mu)) - \frac{3}{2}\mu g''(\mu) e_n \\ & \quad - \frac{5}{8}\mu g'''(\mu) e_n^2 - \left\{ \frac{3\left(\mu g^{iv}(\mu) - 2 g^{iv}(\mu) g'(\mu) + g^{iv}(\mu) g'^2(\mu) + g''^3(\mu)\right)}{16(-1+g'(\mu))^2} \right\} e_n^3 + O\left(e_n^4\right), \end{aligned} \tag{58}$$

$$\begin{aligned} & 2 - \left(g'(s_n) + g'\left(\frac{s_n+u_n}{2}\right)\right) \\ & = 2 - g'(\mu) - g''(\mu) e_n - \frac{1}{2} g'''(\mu) e_n^2 - \frac{1}{6} g^{iv}(\mu) e_n^3 + O\left(e_n^4\right). \end{aligned} \tag{59}$$

By substituting (58) and (59) into Algorithm 12, we get the error term of Algorithm 12,

$$s_{n+1} = \mu + \left\{ \frac{9 g''^3(\mu)}{32(-1+g'(\mu))^3} \right\} e_n^4 + O\left(e_n^5\right), \tag{60}$$

$$e_{n+1} = \left\{ \frac{9 g''^3(\mu)}{32(-1+g'(\mu))^3} \right\} e_n^4 + O\left(e_n^5\right). \tag{61}$$

This completes the proof. □

4. Efficiency Index

Commonly in literature, the efficiency index [31] of an algorithm supplies us with information about the numeric behavior and performance of the method under examination. It is also used to compare different iterative methods and mathematically defined as EI $= P^{1/m}$, where P represents the order of the method and m is the total number of function evaluations (the function and the derivatives involved) per iteration necessary by the method. Taking into account this fact, one can calculate the EI of different iterative methods. Since the Ullah method (UM) [5] is quadratically convergent and needs three function evaluations per iteration, thus EI for this method will be $2^{1/4} \approx 1.18921$. Similarly, EI of Farooq method FM [28] is $3^{1/5} \approx 1.24573$ because order of convergence of the method is three and requires five function evaluations per iteration. The efficiency indexes of Noor methods [22] with cubic and fourth order of convergence are $3^{1/5} \approx 1.24573$, $4^{1/6} \approx 1.2592$, and $4^1\ /8 \approx 1.18921$. Now, we compute the efficiency indexes of newly proposed algorithms. The convergence order of the methods described in Algorithms 7 and 8 is 3, see (37) and (42), and requires four function evaluations per iteration. Thus, the efficiency index for both methods is $3^{1/4} \approx 1.31607$. The convergence order of the methods described in Algorithms 9, 11, and 12 is 4, see equations (47), (53), and (60), and the total number of evaluations per iteration is 4, 6, and 6, respectively. Thus, the efficiency indexes for these methods are $\mathrm{EI} = 4^{1/4} \approx = 1.41421$, $4^{1/6} \approx 1.25992$, and $4^{1/6} \approx 1.25992$, respectively. Table 1 summarizes the efficiency indexes of different algorithms that we have computed above, and it can easily be noted that efficiency indexes of the newly established algorithms AG1, AG2, AG3, AG4, and AG5 are better than the efficiency indexes of other iterative methods.

5. Applications

In this section, we consider two well-known models related to mathematics, physics, and physical chemistry which include a nonlinear model formed due to the motion of particles on an inclined plane and Lenard–Jones potential, a renowned model that represents the interaction between neutral molecules or atoms. We also include some examples used by Chun [7] to elaborate the efficacy of the proposed algorithms. For computational work, we implement codes in MAPLE software and MATLAB for graphical analysis, and the following stopping criterion is taken into account for entire computations:

$$\begin{aligned} &|s_{n+1} - s_n| < \varepsilon, \\ &|f(s_{n+1})| < \varepsilon. \end{aligned} \tag{62}$$

We display a comparative representation of newly established methods: Algorithm 7 (AG1), Algorithm 8 (AG2), Algorithm 9 (AG3), Algorithm 11 (AG4), and Algorithm 12 (AG5), introduced in this paper, with second-order Ullah method (UM) [5], third-order Farooq method (Algorithm 13) (FM) [28], with alpha = 0.9, and Noor methods [22] {(Algorithm 2.8) (NR1), (Algorithm 2.12) (NR2), and (Algorithm 2.15) (NR3)}, to show that the proposed methods perform more efficiently, see Tables 2–4 and Figures 1–3. We obtain an estimated simple root rather than the exact based on the exactness ε of the computer and use $\varepsilon = 10^{-15}$. As for the convergence criteria is concerned it is desired that the distance δ among two consecutive estimations for zero is not more than 10^{-15}. In Tables 2–4, s_0 is the initial guess, (IT) represents the number of the iterations, s_n is the approximate root, and $f(s_n)$ is its corresponding functional value.

$$f_1(s) - f_5(s). \tag{68}$$

The computational order of convergence (COC) (see [9]) is computed to check the behavior of the proposed methods for presented examples and given by

$$\mathrm{COC} \approx \frac{ln|s_{n+1} - s_n|/ln|s_n - s_{n-1}|}{ln|s_n - s_{n-1}|/ln|s_{n-1} - s_{n-2}|}. \tag{63}$$

Example 1 (model based on population growth [33]).

Assume the mathematical modeling of the growth of population over short periods of time whose governing equation is nonlinear in nature as follows:

$$1564,000 = 1,000,000e^{\lambda} + \frac{435000}{\lambda}\left(e^{\lambda} - 1\right). \tag{64}$$

We want to determine the value of λ which demonstrates the constant birth rate of population. For computational work, we take $s_0 = 3$ as an initial estimate. The solution of this example approximated to 16 decimal digits is 0.1009979296857498. The numerical results for this problem are given in Table 2. Figure 1 shows the fall of residuals for this example. It is clear from the computational results in terms of number of iterations that new fixed point iterative methods AG1, AG2, AG3, AG4, and AG5 are more efficient in their performance as compared to the already known ordinary methods UM, FM, NR1, NR2, and NR3.

Example 2 (Lenard–Jones potential model [24]).

Consider a specific model for atom potential referred as Lenard–Jones potential which is a well-known model in atomic physics and physical chemistry.

$$V(s) = V_o\left[\left(\frac{\mu}{s}\right)^{12} - \left(\frac{\mu}{s}\right)^{6}\right], \tag{65}$$

where V_o denotes the depth of the potential and μ is the length scale representing the distance where the interparticle interaction between two atoms becomes zero. The function V(s) in (65) has minimum value at $s = 2^{1/6}\mu$. We choose $V_o = 1$ and $\mu = 2$, then the actual minimum value of function V(s) will be $s = 2^{7/6}$. Now, differentiating $V(s)$ in (65), minimization problem is transformed into the problem of finding the solution of the nonlinear equation given by

$$f(s) = \frac{\mathrm{d}}{\mathrm{d}s}\left(V(s) = V_o\left[\frac{-12\mu^{12}}{s^{13}} + \frac{6\mu^{6}}{s^{7}}\right] = 0\right). \tag{66}$$

In this example for computational evaluations, we utilize $s_o = 1.4$ as an initial estimate. The columns in Table 3 illustrate the comparison of numerical results in terms of number of iterations for this problem. Figure 2 projects the drop of residuals for this example. It is concluded from Table 3 and Figure 2 that the effectiveness and presentation of the proposed schemes are much refined than other similar standard methods. Results of this example also show that fixed point iterative methods converge more rapidly towards solution as compared to the existing ones.

Table 1: Comparison of efficiency indices of different recursive schemes.

Methods	Order of convergence	No. of required computations	EI
UM	2	4	1.18921
FM	3	5	1.24573
NR1	3	5	1.24573
NR2	4	6	1.25992
NR3	4	8	1.18921
AG1	3	4	1.31607
AG2	3	4	1.31607
AG3	4	4	1.41421
AG4	4	6	1.25992
AG5	4	6	1.25992

Table 2: Computation results for motion of particle on an inclined plane (equation (64)).

Methods	IT	s_n	$f(s_n)$	$\Delta = \lvert s_n - s_{n-1} \rvert$	COC
		$g(s) = \log(435000 + 1564000s/435000 + 1000000s),\ \ s_o = 3.0$			
UM	8	0.1009979296857498	$2.504358e-32$	$1.991639e-19$	2.00
FM	7	0.1009979296857498	$4.711728e-46$	$1.242165e-17$	3.00
NR1	5	0.1009979296857498	$3.310599e-17$	$4.808886e-08$	2.97
NR2	4	0.1009979296857498	$7.942106e-20$	$8.672936e-07$	3.99
NR3	4	0.1009979296857498	$2.520548e-18$	$2.058524e-06$	3.99
AG1	4	0.1009979296857498	$1.129996e-05$	$5.259956e-05$	2.99
AG2	5	0.1009979296857498	$6.531994e-22$	$1.775751e-10$	2.99
AG3	4	0.1009979296857498	$5.805863e-19$	$1.250969e-07$	3.98
AG4	4	0.1009979296857498	$1.873858e27$	$1.333454e-09$	3.99
AG5	4	0.1009979296857498	$3.261799e-22$	$2.223887e-08$	3.99

Table 3: Computational results, Lenard–Jones potential model (equation (66)).

Methods	IT	s_n	$f(s_n)$	$\Delta = \lvert s_n - s_{n-1} \rvert$	COC
		$g(s) = (12V_0\mu^{12}/6sV_0\mu^6)^{1/5}, \ldots, s_0 = 1.4$			
UM	11	2.2449240966187460	$5.732991e-25$	$1.852517e-13$	2.00
FM	8	2.2449240966187460	$5.417184e-15$	$3.653371e-06$	2.99
NR1	8	2.2449240966187460	$1.362441e-26$	$5.586731e-10$	2.97
NR2	6	2.2449240966187460	$1.585806e-19$	$4.564142e-06$	3.98
NR3	6	2.2449240966187460	$1.027689e-16$	$2.302915e-05$	3.99
AG1	3	2.2449240966187460	$7.477787e-34$	$4.725369e-11$	2.99
AG2	3	2.2449240966187460	$1.904114e-31$	$2.616447e-10$	2.99
AG3	3	2.2449240966187460	$5.664676e-68$	$8.183945e-17$	4.00
AG4	3	2.2449240966187460	$8.482495e-82$	$4.048693e-20$	4.00
AG5	3	2.2449240966187460	$7.407196e-74$	$3.195595e-18$	4.00

Table 4: Numerical comparison between different algorithms for test problems.

Methods	IT	x_n	$f(x_n)$	$\Delta = \lvert x_n - x_{n-1} \rvert$	COC
		$f_1(s) = s^3 - 10, \ldots, g(s) = \sqrt{10/x}, \ldots, s_0 = 1.5$			
UM	6	2.1544346900318837	$2.003246e-15$	$1.760515e-08$	2.00
FM	4	2.1544346900318837	$6.796495e-18$	$1.369664e-06$	2.99
NR1	4	2.1544346900318839	$6.359558e-34$	$5.962594e-12$	3.02
NR2	3	2.1544346900318839	$6.993369e-26$	$4.733961e-07$	4.27
NR3	3	2.1544346900318839	$1.274020e-26$	$3.092766e-07$	4.00
AG1	4	2.1544346900318837	$3.769672e-82$	$1.262120e-27$	2.99
AG2	3	2.1544346900318837	$8.598170e+60$	$2.696187e-15$	2.99
AG3	3	2.1544346900318837	$6.765235e-52$	$2.969298e-13$	3.99
AG4	3	2.1544346900318837	$1.413885e-64$	$2.839236e-16$	4.00
AG5	3	2.1544346900318837	$4.990636e-01$	$2.625771e-13$	4.00

Table 4: Continued.

Methods	IT	x_n	$f(x_n)$	$\Delta = \lvert x_n - x_{n-1}\rvert$	COC
$f_2(s) = (s-1)^3 - 1, \ldots, g(s) = 1 + \sqrt{1/s - 1}, \ldots, s_0 = 3.5$					
UM	6	2.0000000000000024	7.300615e − 15	4.933091e − 08	2.00
FM	4	2.0000000000000000	8.664196e − 17	1.424022e − 05	2.99
R1	5	2.0000000000000000	3.312983e − 40	4.797695e − 14	2.99
NR2	4	2.0000000000000000	5.783669e − 43	2.095417e − 11	3.87
NR3	4	2.0000000000000000	2.932850e − 42	3.144431e − 11	3.99
AG1	4	2.0000000000000000	2.746057e − 54	2.446636e − 18	2.99
AG2	4	2.0000000000000000	1.318575e − 48	1.673674e − 16	2.99
AG3	3	2.0000000000000000	1.301962e − 33	9.128497e − 09	3.99
AG4	3	2.0000000000000000	9.099015e − 43	6.637634e − 11	4.00
AG5	3	2.0000000000000000	3.700702e − 37	1.368643e − 09	3.99
$f_3(s) = e^{s^2+7s-30} - 1,\ g(s) = \frac{1}{7}(30 - s^2),\ s_0 = 3.5$					
UM	10	3.0000000000000000	9.951599e − 17	1.078856e − 09	2.00
FM	7	3.0000000000000000	9.507655e − 19	1.314119e − 07	2.98
NR1	8	3.0000000000000000	7.833171e − 38	5.183816e − 14	2.99
NR2	6	3.0000000000000000	3.341483e − 27	3.083060e − 08	3.99
NR3	6	3.0000000000000000	1.707093e − 23	2.606524e − 07	3.99
AG1	3	3.0000000000000000	3.811229e − 37	1.704784e − 12	2.99
AG2	3	3.0000000000000000	5.677785e − 35	7.894846e − 12	2.99
AG3	3	3.0000000000000000	4.305207e − 78	1.161329e − 19	3.99
AG4	3	3.0000000000000000	4.174513e − 90	1.629758e − 22	3.98
AG5	3	3.0000000000000000	4.769844e − 83	7.736635e − 21	3.99
$f_5(\mathrm{s}) = \sin^2(s) - s^2 + 1, \ldots, g(s) = \sin(s) + 1/(\sin(s) + s), \ldots, s_0 = -1$					
UM	6	1.4044916482153412	2.217875e − 15	3.376658e − 08	1.99
FM	5	1.4044916482153412	7.221491e − 16	2.778662e − 05	3.06
NR1	4	1.4044916482153412	1.814668e − 31	4.919474e − 11	3.03
NR2	3	1.4044916482153412	2.712791e − 24	1.227650e − 06	4.37
NR3	3	1.4044916482153412	7.318030e − 25	8.847467e − 07	4.33
AG1	4	1.4044916482153412	1.731515e − 90	1.996553e − 30	2.99
AG2	4	1.4044916482153412	1.723974e − 84	1.741614e − 28	2.99
AG3	3	1.4044916482153412	2.351397e − 61	9.774245e − 16	4.00
AG4	3	1.4044916482153412	2.878031e − 70	8.175994e − 18	3.99
AG5	3	1..4044916482153178	2.905287e − 64	2.116013e − 16	3.99
$f_6(s) = s^2 - e^s - 3s + 2, \ldots, g(s) = (s^2 - e^s + 2)/3, \ldots, s_0 = 2$					
UM	5	0.2575302854398608	1.723039e − 24	2.208907e − 12	2.00
FM	11	0.2575302854398608	2.922830e − 27	9.219828e − 10	2.99
NR1	3	0.2575302854398608	3.807427e − 17	1.048816e − 05	3.00
NR2	3	0.2575302854398608	7.862110e − 56	7.105574e − 14	3.99
NR3	3	0.2575302854398608	4.088124e − 45	3.393090e − 11	4.14
AG1	3	0.2575302854398608	4.715256e − 17	1.126308e − 05	3.50
AG2	4	0.2575302854398608	6.577461e − 46	2.368510e − 15	2.99
AG3	3	0.2575302854398608	9.720474e − 29	2.979348e − 07	4.00
AG4	3	0.2575302854398608	2.935314e − 44	5.554284e − 11	4.00
AG5	3	0.2575302854398608	1.293390e − 34	1.168400e − 08	4.00

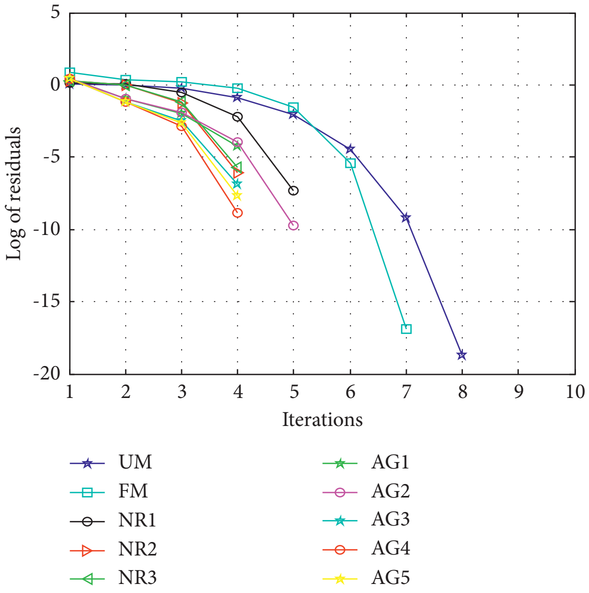

Figure 1: Log of residuals for equation (64).

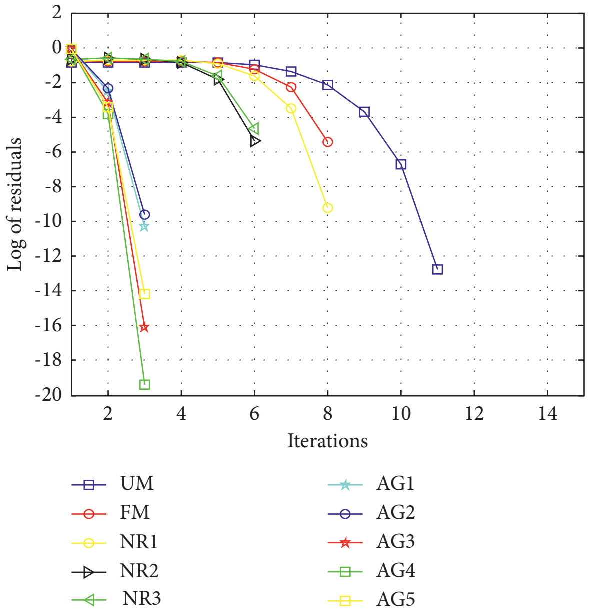

Figure 2: Log of residuals for (equation (66)).

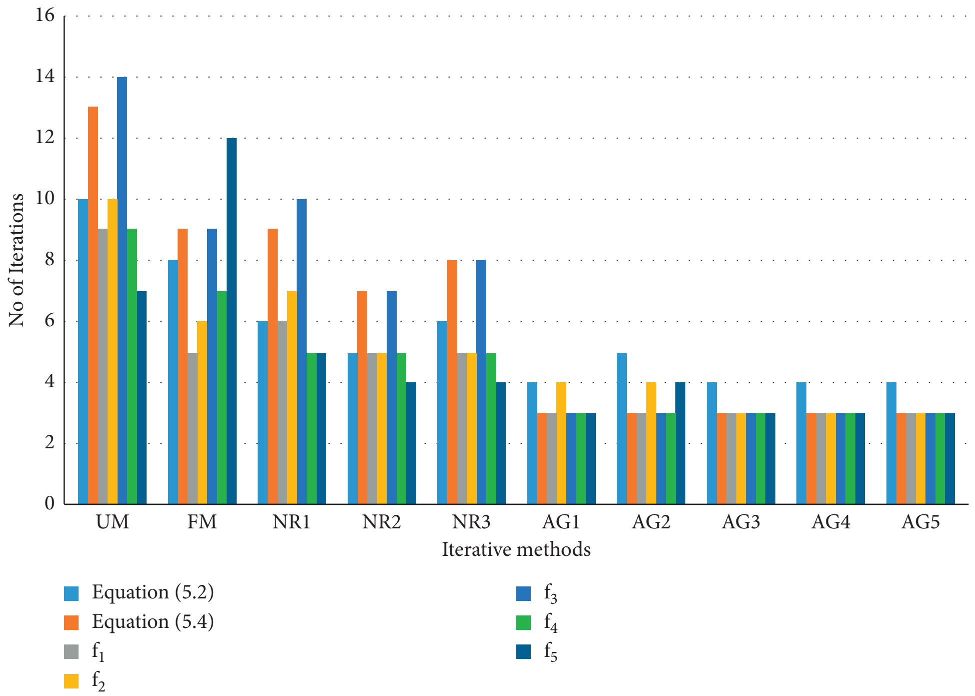

FIGURE 3: Comparison of iterative methods and number of iterations.

TABLE 5: Comparison of number of iterations of iterative method for $\varepsilon = 10^{-200}$.

Iterative Method	Equation (64) $s_o = 3$	Equation (66) $s_o = 1.4$	$f_1(s)$ $s_o = 1.5$	$f_2(s)$ $s_0 = 3.5$	$f_3(s)$ $s_o = 3.5$	$f_4(s)$ $s_o = -1$	$f_5(s)$ $s_o = 2$
UM	10	13	9	10	14	9	7
FM	8	9	5	6	9	7	12
NR1	6	9	6	7	10	5	5
NR2	5	7	5	5	7	5	4
NR3	6	8	5	5	8	5	4
AG1	4	3	3	4	3	3	3
AG2	5	3	3	4	3	3	4
AG3	4	3	3	3	3	3	3
AG4	4	3	3	3	3	3	3
AG5	4	3	3	3	3	3	3

Example 3 (transcendental and algebraic problems).

To numerically analyze the suggested algorithms, we consider the following transcendental and algebraic equations used by Chun [7]:

$$f_1(s) = s^3 - 10, g(s) = \sqrt{\frac{10}{s}}, s_0 = 1.5,$$

$$f_2(s) = (s-1)^3 - 1, g(s) = 1 + \sqrt{\frac{1}{s-1}}, s_0 = 3.5,$$

$$f_3(s) = e^{s^2+7s-30} - 1, g(s) = \frac{1}{7}\left(30 - s^2\right), s_0 = 3.5,$$

$$f_4(s) = \sin^2 s - s^2 + 1, g(s) = \text{sins} + \frac{1}{(\text{sins} + s)}, s_0 = -1, \tag{67}$$

In Table 4, we display the numerical results for examples $f_1(s), f_2(s), f_3(s), f_4(s)$, and $f_5(s)$ to validate the theoretical results. The second column in Table 4 shows the number of iterations required to reach the stopping criteria (62). It is clear from the results obtained that new methods need a smaller number of iterations when compared with other methods.

The columns in Table 5 represent the number of iterations for different nonlinear functions along with initial guess s_0. Figure 3 shows the comparison of the iterative methods with respect to number of iterations. A comparative representation of number of iterations is presented, needed for different methods with our developed methods using the stopping criteria (62) with the accuracy $\varepsilon = 10^{-200}$. It is clear from Table 5 that settling the same convergence criteria for all the methods, the number of iterations required for the new methods remains less than the number of iterations needed by the other methods of the same order.

6. Conclusion

We have introduced a new modified family of recently developed iterative methods (Algorithms 2 and 3) by using a decomposition technique for solving nonlinear equations. Various new iterative methods of different order have been constructed as special cases with the support of recently constructed family. The convergence criteria of newly proposed methods is reviewed and inspected in order to ensure convergence order. In Tables 2–5 and Figures 1–3, we furnish the comparative by taking into account both mathematically and graphically for these strategies with a few known procedures by examining two models and a few algebraic nonlinear equations. The numerical results and graphical depiction certified the swiftness and finest performance of the methods with reference to the number of iterations even though the accuracy has been raised up to $\varepsilon = 10^{-200}$.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research has been partially supported by Ministerio de Ciencia, Innovacion y Universidades, grant no. PGC2018-0971-B-100, and Fundacion Seneca de la Region de Murcia, grant no. 20783/PI/18.

References

[1] S. Abbasbandy, "Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method," *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 887–893, 2003.

[2] F. Ali, W. Aslam, and A. Rafiq, "Some new iterative techniques for the problems involving nonlinear Equations," *International Journal of Computer Mathematics*, vol. 16, pp. 1–18, 2019.

[3] F. Ali, W. Aslam, I. Khalid, and A. Nadeem, "Iteration methods with an auxiliary function for nonlinearequations," *Jurnal Matematika*, vol. 202015 pages, Article ID 7356408, 2020.

[4] A. R. Alharbi, M. I. Faisal, F. A. Shah, M. Waseem, R. Ullah, and S. Sherbaz, "Higher order numerical approaches for nonlinear equations by decomposition technique," *IEEE Acess*, vol. 7, pp. 44329–44337, 2019.

[5] M. Z. Ullah, F. Ahmad, and M. A. A. A. Jbbar, "A correction note on three-step iterative methods for nonlinear equations and generalization of method," *J. Modern. Methods. Num. Anal.*vol. 5, pp. 10–16, 2014.

[6] S. Bhalekar and V. Daftardar-Gejji, "Convergence of the new iterative method," *International Journal of Differential Equations*, vol. 2011, pp. 1–10, Article ID 989065, 2011.

[7] C. Chun, "Iterative methods improving Newton's method by the decomposition method," *Computers & Mathematics with Applications*, vol. 50, no. 10-12, pp. 1559–1568, 2005.

[8] C. Chun, "Construction of Newton-like iteration methods for solving nonlinear equations," *Numerische Mathematik*, vol. 104, no. 3, pp. 297–315, 2006.

[9] A. Cordero and J. R. Torregrosa, "Variants of Newton's Method using fifth-order quadrature formulas," *Applied Mathematics and Computation*, vol. 190, no. 1, pp. 686–698, 2007.

[10] M. T. Darvishi and A. Barati, "A third-order Newton-type method to solve systems of nonlinear equations," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 630–635, 2007.

[11] V. Daftardar-Gejji and H. Jafari, "An iterative method for solving nonlinear functional equations," *Journal of Mathematical Analysis and Applications*, vol. 316, no. 2, pp. 753–763, 2006.

[12] M. Dehghan and M. Hajarian, "New iterative method for solving non-linear equations with fourth-order convergence," *International Journal of Computer Mathematics*, vol. 87, no. 4, pp. 834–839, 2010.

[13] M. Frontini and E. Sormani, "Some variant of Newton's method with third-order convergence," *Applied Mathematics and Computation*, vol. 140, no. 2-3, pp. 419–426, 2003.

[14] J.-H. He, "Variational iteration method-Some recent results and new interpretations," *Journal of Computational and Applied Mathematics*, vol. 207, no. 1, pp. 3–17, 2007.

[15] J.-H. He, "A new iteration method for solving algebraic equations," *Applied Mathematics and Computation*, vol. 135, no. 1, pp. 81–84, 2003.

[16] H. H. H. Homeier, "On Newton-type methods with cubic convergence," *Journal of Computational and Applied Mathematics*, vol. 176, no. 2, pp. 425–432, 2005.

[17] S. Huang, A. Rafiq, and M. R. Shahzad, "New higher order iterative methods for Solving nonlinear equations," *Hacettepe Journal of Mathematics and Statistics*, vol. 47, no. 1, pp. 77–91, 2018.

[18] S. M. Kang, M. Saqib, M. Fahad, and W. Nazeer, "Two new third and fourth order algorithms for Resolution of nonlinear scalar equations based on decomposition technique," *Far East Journal of Mathematical Sciences*, vol. 101, no. 3, pp. 457–471, 2017.

[19] S. M. Kang, F. Ali, and A. Rafiq, "Iterative methods for solving scalar equations," *The Journal of Nonlinear Science and Applications*, vol. 9, no. 3, pp. 1035–1042, 2016.

[20] S. M. Kang, A. Rafiq, and Y. C. Kwun, "A new second-order iteration method for solving nonlinear Equations," *Abstract and Applied Analysis*, vol. 2013, pp. 1–4, Article ID 48706, 2013.

[21] Y. C. Kwun, M. Saqib, and M. Fahad, "A new cubically convergent iterative method for solving nonlinear equations," *International Journal of Pure and Applied Mathematics*, vol. 1, pp. 67–76, 2016.

[22] M. A. Noor, M. Waseem, K. I. Noor, and M. A. Ali, "New iterative technique for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 265, pp. 1115–1125, 2015.

[23] K. I. Noor and M. A. Noor, "Iterative methods with fourth-order convergence for nonlinear equations," *Applied Mathematics and Computation*, vol. 189, no. 1, pp. 221–227, 2007.

[24] A. Ozyapıcı, M. Riza, B. Bilgehan, and AE. Bashirov, "On multiplicative and volterra minimization methods," *Numerical Algorithms*, vol. 67, pp. 623–636, 2014.

[25] A. Y. Özban, "Some new variants of Newton's method," *Applied Mathematics Letters*, vol. 17, no. 6, pp. 677–682, 2004.

[26] A. Rafiq and M. Rafiullah, “Some multi-step iterative methods for solving nonlinear equations,” *Open Journal of Metal*, vol. 58, no. 8, 2009.

[27] M. Saqib, M. Iqbal, S. Ali, and T. Ismaeel, “New fourth and fifth-order iterative methods for solving nonlinear equations,” *Applied Mathematics*, vol. 06, no. 08, pp. 1220–1227, 2015.

[28] F. A. Shah, M. Darus, I. Faisal, and M. A. Shafiq, “Decomposition technique and a family of efficient schemes for nonlinear equations,” *Discrete Dynamics in Nature and Society*, vol. 2017, pp. 1–8, Article ID 3794357, 2017.

[29] G. Sana, M. A. Noor, and K. I. Noor, “Some multistep iterative methods for nonlinear equation using quadrature rule,” *International Journal of Analysis and Applications*, vol. 18, no. 6, pp. 920–938, 2020.

[30] 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.

[31] J. F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea publishing company, New York, USA, 1977.

[32] G. Adomian, *Nonlinear Stochastic Systems and Applications to Physics*, Kluwer Academic Publishers, Netherlands, 1989.

[33] R. L. Burden and J. D. Faires, *Numerical Analysis*, PWS Publishing Company, Boston, MA, USA, 2001.