AIMS Mathematics

AIMS Mathematics, 11(6): 16550–16569.
DOI: 10.3934/math.2026679
Received: 07 January 2026
Revised: 06 March 2026
Accepted: 17 March 2026
Published: 09 June 2026

https://www.aimspress.com/journal/Math

Research article

Predictor-corrector higher derivative-free algorithms for nonlinear equations with basin of attraction analysis

Ali Hasan Ali[1,2,*], Iman A. Abdul Samad[1], Huda J. Saeed[1], Firas Ghanim[3] and Alina Alb Lupaş[4,*]

[1] Department of Mathematics, College of Education for Pure Sciences, University of Basrah, Basrah 61001, Iraq
[2] Technical Engineering College, Al-Ayen University, Dhi Qar, 64001, Iraq
[3] Department of Mathematics, College of Sciences, University of Sharjah, Sharjah, United Arab Emirates
[4] Department of Mathematics and Computer Science, University of Oradea, Romania

* **Correspondence:** Email: ali.hasan@uobasrah.edu.iq, dalb@uoradea.ro.

Abstract: This work presents two new numerical methods for finding the zeros of nonlinear equations, based on the Adomian decomposition approach and the quadrature rule. Using a predictor–corrector scheme, the proposed iterative methods provide a high order of convergence and do not require second derivatives, thereby reducing computational time. The methods are tested on several functions, and their efficiency is demonstrated through comparisons with various established techniques. To support the numerical findings, graphical analyses are provided. Furthermore, a study of the basins of attraction is conducted to validate the stability of the proposed methods.

Keywords: quadrature rule; Adomian decomposition approach; iterative methods; basins of attraction; numerical analysis
Mathematics Subject Classification: 65B99, 65H05

---

1. Introduction

In mathematics, solving nonlinear equations is one of the most fundamental and ancient problems. We may use sequential techniques, including Newton's method and its modifications, to resolve these equations. Newton's technique is a highly efficient and esteemed iteration algorithm, recognized for its convergence to quadratic functions. There have been significant advancements in iterative methods with higher orders of convergence that seek to reduce the computation of lower-order derivatives as much as possible. In this direction, Adomian [1] introduced the Adomian decomposition method for nonlinear problems, Weerakoon and Fernando [2] proposed a third-order variant of Newton's method, Babolian and Biazar [3] developed a modified Adomian decomposition method for solving nonlinear equations, and Abdul-Hassan et al. [4] presented a fifth-order iterative method free from second derivatives. Another strategy for constructing iterative methods for solving nonlinear equations is based on the Adomian decomposition approach. Abbasbandy [5] and Kang et al. [6] improved the Newton–Raphson method using modified Adomian decomposition techniques, while Noor and Noor [7] proposed iterative schemes for nonlinear equations and Darvishi and Barati [8] introduced a third-order Newton-type method for solving systems of nonlinear equations. Over the past few years, a broad class of functional equations has been effectively solved using the Adomian decomposition method, which views the solution as an infinite series that typically converges to a correct answer. Abbaoui and Cherruault [9] established the convergence of the series solution by using the approach to solve the nonlinear equations. In order to create numerical schemes, the Adomian approach has also been modified. Additionally, several variations of the Adomian decomposition approach and Adomian polynomials have been employed by Daftardar-Gejji et al. [10] and Wazwaz [11]. However, one of the main advantages of using this method over the Adomian decomposition technique is that it is simple and does not require evaluating the derivative of the Adomian polynomial. For the purpose of solving nonlinear equations, Chun [12] employed the decomposition approach to improve Newton’s method, Noor et al. [13] proposed new iterative methods for nonlinear equations, Rafiq et al. [14] developed a new family of iterative methods using Adomian polynomials, Saqib and Iqbal [15] introduced multistep iterative methods, Ali et al. [16] presented a new family of iterative methods for nonlinear models, and Sana et al. [17] proposed multistep iterative methods based on a quadrature rule.

Obtaining efficient iterative methods for solving nonlinear equations is always an important area of research, with recent studies including many approaches to improve convergence and stability. For example, advanced fourth-order derivative-free methods have been studied for their local and semi-local convergence in Banach spaces [18]. In addition, a novel Kurchatov-type derivative-free approach with and without memory, providing accurate solutions for nonlinear systems, was studied in [19]. Moreover, Jarrat-type iterative methods such as single-parameter and fourth-order schemes were designed for solving systems of equations in [20]. Beyond standard nonlinear equations, numerical and stability analyses are also applied to complex models, including fractional glucose–insulin systems under Hyers–Ulam stability control; and this can be found in [21]. Also, the application of Laplace–Adomian–Padé methods (LAPM) and Adams–Bashforth–Moulton (ABM) methods to fractional susceptible–carrier–infectious–recovered (SCIR) models for diseases such as pneumonia can be found in [22]. Furthermore, sophisticated numerical solutions have been developed for integro-partial fractional diffusion heat equations involving tempered ψ-Caputo derivatives in [23]. After that, high-order compact schemes for nonlinear integro-differential equations with weakly singular kernels were studied in [24]. Finally, the existence results for higher-order nonlinear fractional differential equations using coincidence degree theory can be obtained from [25].

Here, we present novel predictor–corrector methods for solving an equivalent coupled system of equations by applying some numerical techniques. These methods utilize a multistage framework where a preliminary estimation (the predictor) is refined by a subsequent stage (the corrector) derived from the modified Adomian decomposition method and quadrature rules. This work also investigates the derivation of iterative methods with fourth- and eighth-order convergence for solving nonlinear equations. In addition, the basins of attraction associated with these methods are analyzed. The proposed schemes require the evaluation of three to four function values per iteration, and no higher-order derivatives of the function are required.

The remainder of the paper is structured as follows: Section 2 delineates the development and formulation of the recommended methodologies. Section 3 provides the convergence analysis supported by relevant theorems. Section 4 presents numerical experiments aimed at assessing the efficacy of the suggested methodologies relative to established procedures. Section 5 investigates the stability and robustness of the methods through the analysis of basins of attraction. Finally, Section 6 summarizes the main findings and discusses the overall effectiveness of the developed algorithms.

2. Derivation and construction of the proposed methods

Assume that γ represents the simple root of the equation

$$g(x) = 0, \tag{2.1}$$

where $g\colon I \subseteq \mathbb{R} \longrightarrow \mathbb{R}$ is a sufficiently differentiable function. Let η be an initial guess close to γ. Equation (2.1) can be stated as follows using the quadrature rule [26] and Newton's theorem:

$$g(x) = \frac{(x-\eta)}{k}\sum_{j=1}^{k} g'\left(\eta + \frac{(x-\eta)(2j-1)}{2k}\right) + g(\eta). \tag{2.2}$$

Using Eq (2.2) with $k = 2$, we get

$$g(x) = \frac{(x-\eta)}{2}\left\{g'\left(\frac{3\eta + x}{4}\right) + g'\left(\frac{\eta + 3x}{4}\right)\right\} + g(\eta). \tag{2.3}$$

As a coupled system of equations, Eq (2.3) can be shown in the following form:

$$g(x) = \hbar(x) + g(\eta) + \frac{(x-\eta)}{2}\left\{g'\left(\frac{3\eta + x}{4}\right) + g'\left(\frac{\eta + 3x}{4}\right)\right\} \tag{2.4}$$

such that

$$\hbar(x) = g(x) - \left(\frac{(x-\eta)}{2}\left\{g'\left(\frac{3\eta + x}{4}\right) + g'\left(\frac{\eta + 3x}{4}\right)\right\} + g(\eta)\right). \tag{2.5}$$

We get from Eq (2.4)

$$x = \eta - \frac{2(g(\eta) + \hbar(x))}{g'\left(\frac{3\eta + x}{4}\right) + g'\left(\frac{\eta + 3x}{4}\right)}. \tag{2.6}$$

Equation (2.6) has the following possible expressions:

$$x = \nu + \psi(x), \tag{2.7}$$

where

$$\nu = \eta, \tag{2.8}$$

and

$$\psi(x) = -\frac{2\{g(\eta) + \hbar(x)\}}{g'\left(\frac{3\eta + x}{4}\right) + g'\left(\frac{\eta + 3x}{4}\right)}. \tag{2.9}$$

In this case, the derivative of $\psi(x)$, a nonlinear operator, and a constant ν, is as follows:

$$\psi'(x) = \frac{2(g(\eta)+\hbar(x))\left(\frac{1}{4}g''\left(\frac{3\eta+x}{4}\right)+\frac{3}{4}g''\left(\frac{\eta+3x}{4}\right)\right)}{\left(g'\left(\frac{3\eta+x}{4}\right)+g'\left(\frac{\eta+3x}{4}\right)\right)^2} - \frac{2\hbar'(x)}{g'\left(\frac{3\eta+x}{4}\right)+g'\left(\frac{\eta+3x}{4}\right)}. \tag{2.10}$$

The Adomian approach, in its most basic form, uses an infinite series formulation given by

$$x = \sum_{i=0}^{\infty} x_i\,, \tag{2.11}$$

then

$$\psi(x) = \sum_{i=0}^{\infty} \mathrm{F}_i \tag{2.12}$$

such that F_i for each j is given formally by

$$\mathrm{F}_i = \frac{1}{i!}.\frac{d^i}{d\beta^i}\left[\psi\left(\sum_{j=0}^{i} \beta^j x_j\right) - \psi\left(\sum_{j=0}^{i-1} \beta^j x_j\right)\right]_{\beta=0}, \tag{2.13}$$

where F_i is the Adomian polynomial for each j, and β is a formal parameter. Such polynomials are all dependent on $x_0, x_1, \ldots$. Wazwaz [11] studied these polynomials, which are enumerated in the following order:

$$\begin{gathered}
\mathrm{F}_0 = \psi(x_0), \\
\mathrm{F}_1 = x_1\psi'(x_0), \\
\mathrm{F}_2 = x_2\psi'(x_0), \\
\vdots \\
\mathrm{F}_i = x_i\psi'(x_0), \\
\vdots
\end{gathered}$$

Equation (2.7) can be obtained by substituting Eqs (2.11) and (2.12).

$$\sum_{i=0}^{\infty} x_i = \nu + \sum_{i=0}^{\infty} \mathrm{F}_i. \tag{2.14}$$

Equations (2.8) and (2.14) lead us to the conclusion that

$$x_0 = \nu = \eta. \tag{2.15}$$

Because $\hbar(x_0) = 0$, solving Eq (2.5) is now simple. Consequently, Eqs (2.8), (2.9), and (2.14) can be used to get

$$x_1 = \psi(x_0) = -\frac{g(\eta)}{g'(\eta)}. \tag{2.16}$$

Equations (2.10), (2.14), and (2.16) are once again used to get

$$x_2 = \mathrm{F}_1 = x_1\psi'(x_0) = -\frac{g^2(\eta)g''(\eta)}{2g'^3(\eta)}. \tag{2.17}$$

Equations (2.10), (2.14), and (2.17) are additionally employed to get

$$x_3 = \mathrm{F}_2 = x_2\psi'(x_0) = -\frac{g^3(\eta)(g''(\eta))^2}{4g'^5(\eta)}. \tag{2.18}$$

Equations (2.15)–(2.18) can be substituted into Eq (2.11), giving us

$$x \approx x_3 + x_2 + x_1 + x_0 = \eta - \frac{g(\eta)}{g'(\eta)} - \frac{g^2(\eta)g''(\eta)}{2g'^3(\eta)} - \frac{g^3(\eta)(g''(\eta))^2}{4g'^5(\eta)}. \tag{2.19}$$

Thus, we can construct a new one-step iterative method based on Eq (2.19), which serves as the basis for the suggested methods. The algorithm that follows does this.

Algorithm 2.1. Using the following one-step iterative method, x_{s+1} can be computed for a given initial guess, x_0:

$$x_{s+1} = x_s - \frac{g(x_s)}{g'(x_s)} - \frac{g^2(x_s)g''(x_s)}{2g'^3(x_s)} - \frac{g^3(x_s)(g''(x_s))^2}{4g'^5(x_s)}, \quad s = 0,1,2,3,\dots .$$

Here, the predictor–corrector methods for solving nonlinear equations are presented.

Algorithm 2.2. Using the following two-step iterative method, x_{s+1} can be computed for a given initial guess, x_0:

$$x_{s+1} = y_s - \frac{g(y_s)}{g'(y_s)} - \frac{g^2(y_s)g''(y_s)}{2g'^3(y_s)} - \frac{g^3(y_s)(g''(y_s))^2}{4g'^5(y_s)},$$

$$y_s = x_s - \frac{g(x_s)}{g'(x_s)}, \quad s = 0,1,2,3,\dots .$$

Algorithm 2.3. Using the following three-step iterative method, x_{s+1} can be computed for a given initial guess, x_0:

$$x_{s+1} = z_s - \frac{g(z_s)}{g'(z_s)},$$

$$z_s = y_s - \frac{g(y_s)}{g'(y_s)} - \frac{g^2(y_s)g''(y_s)}{2g'^3(y_s)} - \frac{g^3(y_s)(g''(y_s))^2}{4g'^5(y_s)},$$

$$y_s = x_s - \frac{g(x_s)}{g'(x_s)}\ ,\ s = 0,1,2,3,\dots .$$

Additionally, by estimating the values of functions such as $g'(y_s)$, $g''(y_s)$, and $g'(z_s)$, employing the Bernstein orthogonal polynomials as a basis, we enhance the efficacy of the presented iteration strategies. This strategy circumvents the assessment of the function's second derivative, which may provide challenges.

Consider the function $\mathcal{B}(t)$, which is given as follows:

$$\mathcal{B}(t) = \mu_0 - \mu_1\left((y_s - t) + \frac{1}{2}\right) - \mu_2\left((y_s - t)^2 + (y_s - t) - \frac{1}{6}\right).$$

We estimate $g'(y_s)$ using the interpolation criteria listed below:

$$g(y_s) = \mathcal{B}(y_s),\ g(x_s) = \mathcal{B}(x_s)\ \text{ and }\ g'(x_s) = \mathcal{B}'(x_s).$$

With three unknowns, these three linear equations can be solved to provide

$$g'(y_s) = 2g[x_s, y_s] - g'(x_s) := \mathcal{B}_1(x_s, y_s).$$

Now, we examine the function

$$\mathcal{B}(t) = \mu_0 - \mu_1\left((y_s - t) + \frac{1}{2}\right) - \mu_2\left((y_s - t)^2 + (y_s - t) - \frac{1}{6}\right)$$
$$-\mu_3\left((y_s - t)^3 + \frac{3}{2}(y_s - t)^2 - \frac{1}{2}(y_s - t)\right).$$

Once more, using the following interpolation conditions, we estimate $g''(y_s)$:

$$g(x_s) = \mathcal{B}(x_s), g(y_s) = \mathcal{B}(y_s), g'(x_s) = \mathcal{B}'(x_s)\text{ and }g'(y_s) = \mathcal{B}'(y_s).$$

After determining the four unknowns and solving the four linear equations, we get

$$g''(y_s) = \frac{-2}{(x_s - y_s)}\left(2\mathcal{B}_1(x_s, y_s) + g'(x_s) - 3g[x_s, y_s]\right) := \mathcal{B}_2(x_s, y_s).$$

Now, consider the function

$$\mathcal{B}(t) = \mu_0 - \mu_1\left((z_s - t) + \frac{1}{2}\right) - \mu_2\left((z_s - t)^2 + (z_s - t) - \frac{1}{6}\right)$$

$$-\mu_3\left((z_s - t)^3 + \frac{3}{2}(z_s - t)^2 - \frac{1}{2}(z_s - t)\right).$$

The next approximation principles are used to estimate $g'(z_s)$:

$$g(z_s) = \mathcal{B}(z_s),\ g(y_s) = \mathcal{B}(y_s),\ \ g(x_s) = \mathcal{B}(x_s), \text{and}\ g'(x_s) = \mathcal{B}'(x_s).$$

After identifying the four unknowns and considering the solutions to the four linear equations, we get

$$g'(z_s) = g[y_s, z_s] + 2(g[x_s, z_s] - g[y_s, x_s]) + \frac{y_s - z_s}{y_s - x_s}(g[y_s, x_s] - g'(x_s)) := \mathcal{B}_3(x_s, y_s, z_s).$$

Thus, the following algorithms are expansions of Algorithms 2.2 and 2.3:

Algorithm 2.4. Employing the subsequent two-stage iterative approach, x_{s+1} can be computed for that specific initial evaluation (x_0) as follows:

Step 1 (Predictor Step): Calculate the intermediate value y_s using a standard Newton-like step:

$$y_s = x_s - \frac{g(x_s)}{g'(x_s)},\quad s = 0, 1, 2, 3, \ldots.$$

Step 2 (Corrector Step): Calculate the final iterate x_{s+1} by applying the correction terms derived from the Adomian polynomials and Bernstein approximations:

$$x_{s+1} = y_s - \left(\frac{g(y_s)}{\mathcal{B}_1(x_s, y_s)} + \frac{g^2(y_s)\,\mathcal{B}_2(x_s, y_s)}{2(\mathcal{B}_1(x_s, y_s)\,)^3} + \frac{g^3(y_s)\big(\,\mathcal{B}_2(x_s, y_s)\big)^2}{4(\mathcal{B}_1(x_s, y_s)\,)^5}\right).$$

Algorithm 2.5. Employing the subsequent two-stage iterative approach, x_{s+1} can be computed for that specific initial evaluation (x_0) as follows:

Step 1 (First Predictor): Compute y_s (as in the AM2 predictor).

Step 2 (Second Predictor): Compute z_s using the AM2 corrector logic to reach a higher intermediate order:

$$z_s = y_s - \left(\frac{g(y_s)}{\mathcal{B}_1(x_s, y_s)} + \frac{g^2(y_s)\,\mathcal{B}_2(x_s, y_s)}{2(\mathcal{B}_1(x_s, y_s)\,)^3} + \frac{g^3(y_s)\big(\,\mathcal{B}_2(x_s, y_s)\big)^2}{4(\mathcal{B}_1(x_s, y_s)\,)^5}\right).$$

Step 3 (Final Corrector): Compute the final iterate x_{s+1} using the $\mathcal{B}_3$ approximation:

$$x_{s+1} = z_s - \frac{g(z_s)}{\mathcal{B}_3(x_s, y_s, z_s)}.$$

3. Analysis of convergence

The convergence of the current Algorithms 2.4 and 2.5 is demonstrated in the ensuing theorems.

Theorem 3.1. Suppose that $g\colon I \subseteq \mathbb{R} \to \mathbb{R}$ with the subset I (assumed open). If the simple root $\gamma \in I$ of $g(x)$ is close enough to x_0, this indicates that the newly proposed following error equation is satisfied by Algorithm 2.4:

$$e_{s+1} = -c_2 c_3 e_s^4 + O(e_s^5).$$

Proof. Let γ denote a simple root for the equation $g(x) = 0$, (in other words, $g(\gamma) = 0$ and

$g'(\gamma) \neq 0$). Applying the extension of the Taylor series [27] may be utilized for expanding $g(x_s)$ and $g'(x_s)$ to obtain

$$g(x_s) = g'(\gamma)[e_s + c_2e_s^2 + c_3e_s^3 + c_4e_s^4 + O(e_s^5)], \tag{3.1}$$

where $c_n = (g^{(n)}(\gamma))/(n!\,g'(\gamma))$, $n = 2,\ 3,\ 4, \ldots,$ and $e_s = x_s - \gamma$.

We get from Eq (3.1) that

$$g'(x_s) = g'(\gamma)[1 + 2c_2e_s + 3c_3e_s^2 + 4c_4e_s^3 + 5c_5e_s^4 + O(e_s^5)]. \tag{3.2}$$

Moreover, we compute y_s as

$$y_s = \gamma + c_2e_s^2 + (2c_3 - 2c_2^2)e_s^3 + (3c_4 - 7c_2c_3 + 4c_2^3)e_s^4 + O(e_s^5). \tag{3.3}$$

When we expand $g(y_s)$ around γ using Eq (3.3), we obtain

$$g(y_s) = g'(\gamma)[c_2e_s^2 + (2c_3 - 2c_2^2)e_s^3 + (3c_4 - 7c_2c_3 + 5c_2^3)e_s^4 + O(e_s^5)]. \tag{3.4}$$

Thus, we attain the following:

$$\frac{g(y_s)}{\mathcal{B}_1(x_s, y_s)} = c_2e_s^2 + (-2c_2^2 + 2c_3)e_s^3 + (3c_2^3 - 6c_2c_3 + 3c_4)e_s^4 + O(e_s^5), \tag{3.5}$$

$$\frac{g^2(y_s)\,\mathcal{B}_2(x_s, y_s)}{2(\mathcal{B}_1(x_s, y_s)\,)^3} = c_2^3e_s^4 + (6c_2^2c_3 - 4c_2^4)e_s^5 + O(e_s^6), \tag{3.6}$$

$$\frac{g^3(y_s)\big(\,\mathcal{B}_2(x_s, y_s)\big)^2}{4(\mathcal{B}_1(x_s, y_s)\,)^5} = c_2^5e_s^6 + \Big(-6\big(c_2^2(c_2^2 - c_3)\big)c_2^2 + 4c_2^3(c_2c_3)\Big)\,e_s^7 + O(e_s^8). \tag{3.7}$$

Equations (3.5), (3.6), and (3.7) can be substituted to provide

$$x_{s+1} = \gamma - c_2c_3e_s^4 + O(e_s^5). \tag{3.8}$$

Thus, we have

$$e_{s+1} = -c_2c_3e_s^4 + O(e_s^5).$$

It demonstrates the fourth-order convergence of Algorithm 2.4.

Theorem 3.2. Suppose that $g: I \subseteq \mathbb{R} \to \mathbb{R}$ with the subset I (assumed open). If the simple root $\gamma \in I$ of $g(x)$ is close enough to x_0, this indicates that the newly proposed following error equation is satisfied by Algorithm 2.5:

$$e_{s+1} = c_2^2c_3(c_2c_3 - c_4)e_s^8 + O(e_s^9).$$

Proof. Let γ be a simple root of $g(x) = 0$, and define the error at each stage as $e_s = x_s - \gamma$, $e_{y,s} = y_s - \gamma$, and $e_{z,s} = z_s - \gamma$. We define the constant parameters as

$$c_n = \frac{g^{(n)}(\gamma)}{n!\,g'(\gamma)} \quad for \quad n = 2,3, \ldots$$

In Algorithm 2.5, the intermediate stage z_s is calculated using the same logic as the final iterate of the fourth-order method in Algorithm 2.4. Therefore, by applying the result established in Eq (3.8), the error at the z_s stage is

$$e_{z,s} = z_s - \gamma = -c_2 c_3 e_s^4 + (2c_2^2 c_3 - 2c_2 c_4 - 2c_3^2)e_s^5 + O(e_s^6).$$

We now expand $\mathscr{g}(z_s)$ using a Taylor series around the root γ:

$$\mathscr{g}(z_s) = \mathscr{g}'(\gamma)\left[e_{z,s} + c_2 e_{z,s}^2 + c_3 e_{z,s}^3 + O(e_{z,s}^4)\right].$$

Substituting the expression for $e_{z,s}$ into this expansion allows us to express $\mathscr{g}(z_s)$ in terms of the initial error e_s. Simultaneously, we expand the Bernstein approximation $\mathcal{B}_3(x_s, y_s, z_s)$ defined in Section 2, which estimates $\mathscr{g}'(z_s)$. The expansion around γ yields

$$\mathcal{B}_3(x_s, y_s, z_s) = \mathscr{g}'(\gamma)[1 + O(e_s^4)].$$

By performing the division of the expansion for $\mathscr{g}(z_s)$ by the expansion for $\mathcal{B}_3(x_s, y_s, z_s)$, we obtain the intermediate ratio presented as

$$\frac{\mathscr{g}(z_s)}{\mathcal{B}_3(x_s, y_s, z_s)} = -c_2 c_3 e_s^4 + (2c_2^2 c_3 - 2c_2 c_4 - 2c_3^2)e_s^5 + O(e_s^6).$$

Finally, the error for the next iteration is determined by substituting these series into the corrector step:

$$e_{s+1} = z_s - \gamma - \frac{\mathscr{g}(z_s)}{\mathcal{B}_3(x_s, y_s, z_s)}.$$

The subtraction of these two expansions results in the cancellation of all lower-order terms (from e_s^4 to e_s^7), leaving the leading error term for the eighth-order convergence as

$$e_{s+1} = c_2^2 c_3 (c_2 c_3 - c_4) e_s^8 + O(e_s^9).$$

This completes the proof, concluding eighth-order convergence for Algorithm 2.5.

3.1. The efficiency index and Kung–Traub conjecture

3.1.1. The efficiency index

The efficiency index of a method, which is explained in [28], may now be found by applying the definition $E.I. = \sqrt[M]{p}$ wherever p is the procedure's order, and M is the number of functions evaluated every iteration. Based on the use of this concept, our suggested iterative approaches have an efficiency index of $\sqrt[3]{4} \approx 1.5874$ and $\sqrt[4]{8} \approx 1.6818$, respectively.

3.1.2. Kung–Traub conjecture

According to the Kung–Traub conjecture, an iterative method without memory is considered optimal if its order of convergence reaches $p = 2^{M-1}$, where M is the number of function evaluations per iteration. Therefore, the optimality of AM2 (Algorithm 2.4) is clear, as this method has a fourth-order convergence ($p = 4$) and requires three function evaluations ($g(x_s)$, $g'(x_s)$, and $g(y_s)$) per iteration. According to the Kung–Traub conjecture, the maximum optimal order for $M = 3$ is $2^{3-1} = 4$. Thus, AM2 is an optimal fourth-order method. Moreover, the optimality of AM3 (Algorithm 2.5) is also clear, as this method has an eighth-order convergence ($p = 8$) and requires four function evaluations ($g(x_s)$, $g'(x_s)$, $g(y_s)$, and $g(z_s)$) per iteration. The maximum optimal order for $M = 4$ is $2^{4-1} = 8$. Thus, AM3 is an optimal eighth-order method.

4. Numerical applications

This section compares the performance of the suggested Algorithms 2.4 and 2.5, which we have named (AM2) and (AM3), respectively to that of the traditional Newton technique (NM) [29], the Rafiq et al. technique (A) [14], the Chun technique (CM) [12], and the Ali et al. technique (AG4) [30]. The purpose of the comparison is to demonstrate the applicability and effectiveness of the recently suggested techniques. The Maple 16 program is used to evaluate methods using floating-point computation along with 1000 digits (Digits = 1000). The calculations were carried out on a 64-bit Windows 10 Pro. The machine is equipped with a 1.8 GHz Intel(R) Core(TM) i5-8265U processor with 8.00 GB of installed RAM. The system setup provided sufficient resources to ensure that the Maple 16 algorithms employed in this investigation ran smoothly and effectively. In the experiments, the criteria for terminating the iterative process are $|x_{s+1} - x_s| < \epsilon$ and $|g(x_{s+1})| < \epsilon,$ where $\epsilon = 10^{-15}$. In addition, we provide a comprehensive numerical comparison for various test functions $g_t(x),\ t = 1, 2, 3, 4, 5$; the number of iterations (IN); the absolute error $|x_{s+1} - x_s|$; the functional residual $|g(x_{s+1})|$; and the computational order of convergence (COC), which are described in [31] as

$$\text{COC} = \frac{log|e_{s+1}/e_s|}{log|e_s/e_{s-1}|},$$

where $e_s = x_s - \gamma$ represents the error relative to the true root γ. Moreover, to provide a rigorous assessment of the computational efficiency of the proposed methods, we provide the average CPU time measured in seconds (CPU time). Noting that calculating the CPU time for only one time might be affected by background system processes and transient performance variations, we executed each algorithm 500 times for every test function and calculated the arithmetic mean. This ensures the accuracy of the timing data, providing a more representative measure of the computational effort required by each scheme. We analyze a variety of nonlinear equations using different nonlinear functions, such as polynomial, exponential, logarithmic, and trigonometric functions, to verify our proposed approaches. The equations used in our study are shown in Table 1 with their starting estimate, x_0. Tables 2–6 below compare the suggested and related strategies utilizing these equations, and Figures 1–5 show the iterations versus the residuals in logarithmic scale of these equations.

Table 1. Nonlinear equations and initial guesses employed in the investigation.

Equation	Initial guess
$\boldsymbol{g_1(x) = (x-1)^3 - 1}$	$x_0 = 3.5$
$\boldsymbol{g_2(x) = cos(x) - x}$	$x_0 = 1.7$
$\boldsymbol{g_3(x) = sin^2(x) - (x^2 - 1)}$	$x_0 = 1.0$
$\boldsymbol{g_4(x) = e^{-x} - cos\ (x)}$	$x_0 = 4.0$
$\boldsymbol{g_5(x) = 2x - 7 - ln(x)}$	$x_0 = 6.0$

Table 2. Comparison of the related and new methods for $g_1(x)$.

Method	IN	$\boldsymbol{x_{s+1}}$	$\boldsymbol{\lvert x_{s+1} - x_s \rvert}$	$\boldsymbol{\lvert g(x_{s+1}) \rvert}$	COC	CPU Time
NM	8	2.00000000000000	8.2803903e-22	2.0569459e-42	2.0000003	0.000468
CM	5	2.00000000000000	2.7423046e-24	8.4830891e-94	3.9885844	0.000656
A	6	2.00000000000000	4.1903588e-38	2.2073687e-112	2.9999963	0.000874
AG4	5	2.00000000000000	1.3348425e-18	5.7146892e-71	3.97073710	0.001250
AM2	4	2.00000000000000	2.2793394e-23	2.6992055e-91	4.0066365	0.001220
AM3	3	2.00000000000000	8.4661595e-22	8.7977235e-170	6.9001197	0.001342

Table 3. Comparison of the related and new methods for $g_2(x)$.

Method	IN	$\boldsymbol{x_{s+1}}$	$\boldsymbol{\lvert x_{s+1} - x_s \rvert}$	$\boldsymbol{\lvert g(x_{s+1}) \rvert}$	COC	CPU Time
NM	5	0.73908513321516	2.3449119e-16	2.0319711e-32	1.9999751	0.000218
CM	4	0.73908513321516	1.8674679e-53	1.0956364e-212	3.9998322	0.000500
A	4	0.73908513321516	1.1909882e-26	1.3784676e-79	2.9994961	0.000468
AG4	5	0.73908513321516	1.9043337e-15	2.9674587e-146	3.9986988	0.001092
AM2	4	0.73908513321516	1.4850772e-49	1.2057698e-197	3.9999221	0.001062
AM3	3	0.73908513321516	4.3174787e-53	2.3715233e-424	7.8544833	0.001094

Table 4. Comparison of the related and new methods for $g_3(x)$.

Method	IN	$\boldsymbol{x_{s+1}}$	$\lvert\boldsymbol{x_{s+1}-x_s}\rvert$	$\lvert\boldsymbol{g(x_{s+1})}\rvert$	COC	CPU Time
NM	7	1.40449164821534	7.3278808e-26	1.0445269e-50	2.0000000	0.000626
CM	5	1.40449164821534	1.3119542e-17	1.7691478e-67	3.9514049	0.001158
A	5	1.40449164821534	5.8667464e-21	3.0777484e-61	2.9991587	0.001158
AG4	8	1.40449164821534	7.0677645e-23	4.51492510e-88	3.9858155	0.004218
AM2	4	1.40449164821534	4.1461693e-55	5.0404850e-219	4.0001647	0.001720
AM3	3	1.40449164821534	2.9457292e-45	1.4822574e-358	7.9600164	0.001970

Table 5. Comparison of the related and new methods for $g_4(x)$.

Method	IN	$\boldsymbol{x_{s+1}}$	$\lvert\boldsymbol{x_{s+1}-x_s}\rvert$	$\lvert\boldsymbol{g(x_{s+1})}\rvert$	COC	CPU Time
NM	5	4.72129275884769	1.3519063e-18	1.6272781e-38	1.9959969	0.000250
CM	4	4.72129275884769	3.7108920e-58	6.5754249e-236	4.0323518	0.000626
A	4	4.72129275884769	1.1605193e-41	1.2281792e-127	2.9948698	0.000562
AG4	4	4.72129275884769	1.6471456e-24	2.5933843e-97	3.9638812	0.001156
AM2	4	4.72129275884769	1.3192559e-45	4.4156984e-183	4.0175123	0.001250
AM3	3	4.72129275884769	1.6693933e-48	1.1213445e-390	8.7946366	0.001374

Table 6. Comparison of the related and new methods for $g_5(x)$.

Method	IN	$\boldsymbol{x_{s+1}}$	$\lvert\boldsymbol{x_{s+1}-x_s}\rvert$	$\lvert\boldsymbol{g(x_{s+1})}\rvert$	COC	CPU Time
NM	5	4.21990648378038	1.6990551e-25	8.1054901e-52	1.9999997	0.000218
CM	3	4.21990648378038	2.0634132e-22	6.4548999e-92	3.8766924	0.000344
A	4	4.21990648378038	1.0023953e-43	4.5038777e-133	2.9999854	0.000468
AG4	4	4.21990648378038	1.1243527e-24	1.70992310e-97	3.9884335	0.000750
AM2	3	4.21990648378038	1.4979573e-20	3.5569120e-84	3.8877033	0.000750
AM3	3	4.21990648378038	9.2157809e-78	2.8521836e-626	7.8921677	0.001156

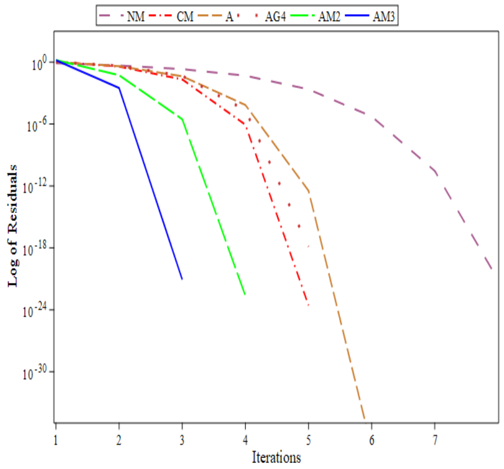


Figure 1. Residuals in logarithmic scale for $g_1(x)$.

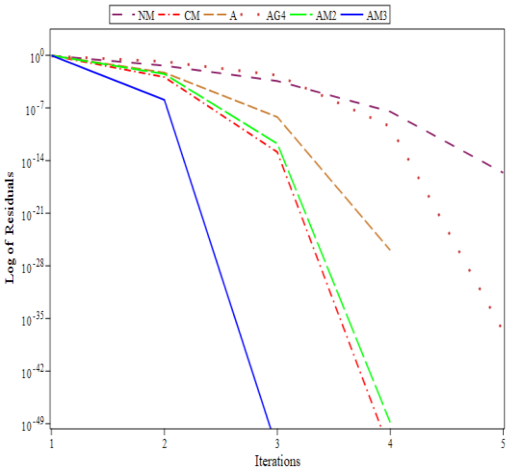


Figure 2. Residuals in logarithmic scale for $g_2(x)$.

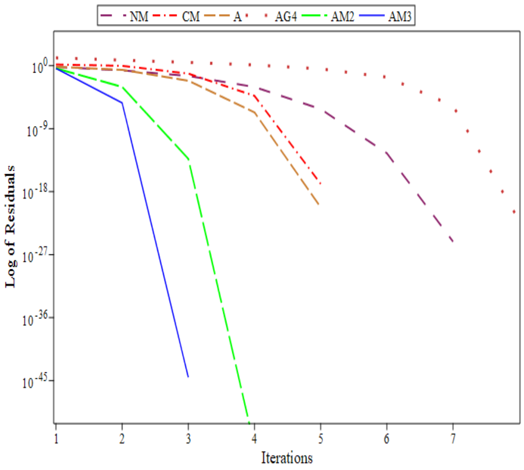


Figure 3. Residuals in logarithmic scale for $g_3(x)$.

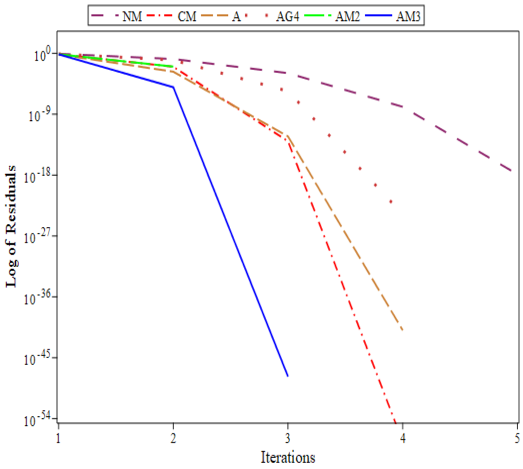


Figure 4. Residuals in logarithmic scale for $g_4(x)$.

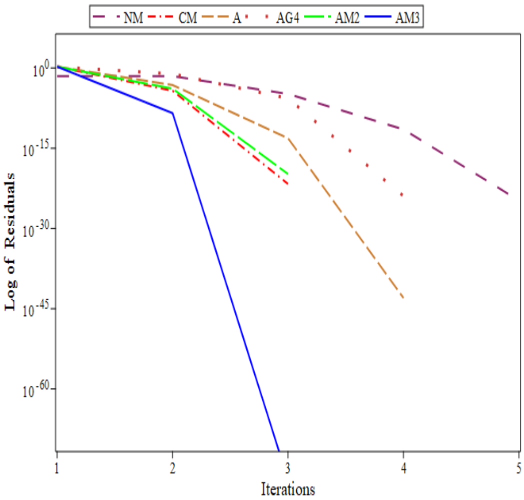


Figure 5. Residuals in logarithmic scale for $g_5(x)$.

5. Basin of attraction

The stability of iterative methods is considered a significant challenge in the research. Therefore, we study the stability of the proposed methods here through the analysis of basins of attraction. A basin of attraction for a specific root z^* of a nonlinear equation $K(z) = 0$ is formally defined as the set of all initial starting points z_0 in the complex plane that converge to z^* under the iterative process. This tool visually provides a direct comparison of the robustness of different numerical methods. In our work, we take three polynomials with distinct sets of roots to test for the analysis and the use of the iterative methods. The roots that are represented by z^* are the solutions to the equation $K(z) = 0$ and can be explained in the following way:

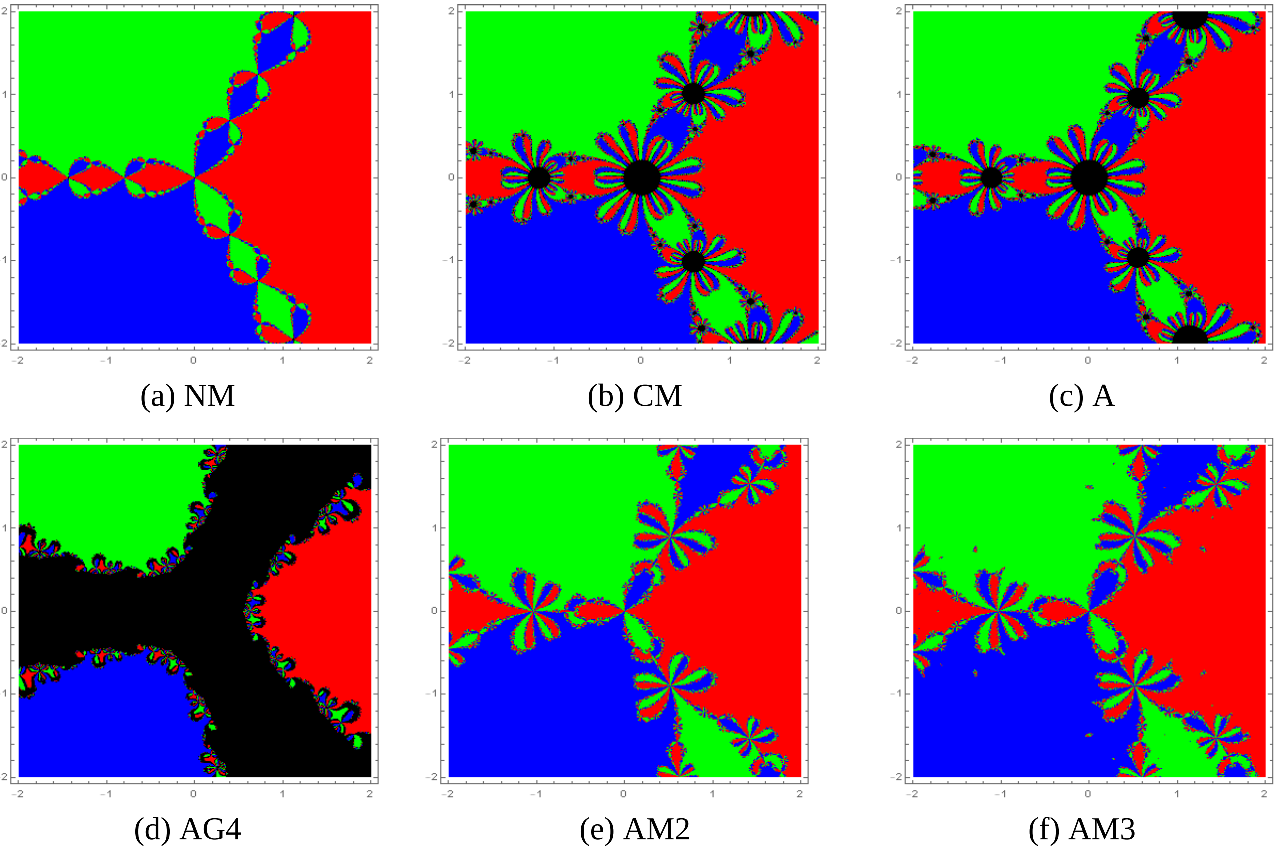
1) $K_1(z) = z^3 - 1$ with roots $z_1^* = 1,\ z_2^* = e^{2\pi i/3}$ and $z_3^* = e^{4\pi i/3}$;
2) $K_2(z) = (z^2 - 1)(z^2 + 1)$ with roots $z_1^* = 1,\ z_2^* = -1,\ z_3^* = i$ and $z_4^* = -i$;

3) $K_3(z) = z^5 - 1$ with roots $z_1^* = 1$, $z_2^* = e^{2\pi i/5}$, $z_3^* = e^{4\pi i/5}$, $z_4^* = e^{6\pi i/5}$ and $z_5^* = e^{8\pi i/5}$.

The convergence of points in the complex plane to these roots was determined using a 512×512 grid providing the region $\mathbb{R} \times \mathbb{R} = [-2,2] \times [-2,2]$. The iterative process was applied at each grid point z until the stopping criterion $|z - z^*| < 10^{-3}$ or $|g(z)| < 10^{-3}$ was satisfied, with a maximum of 25 iterations.

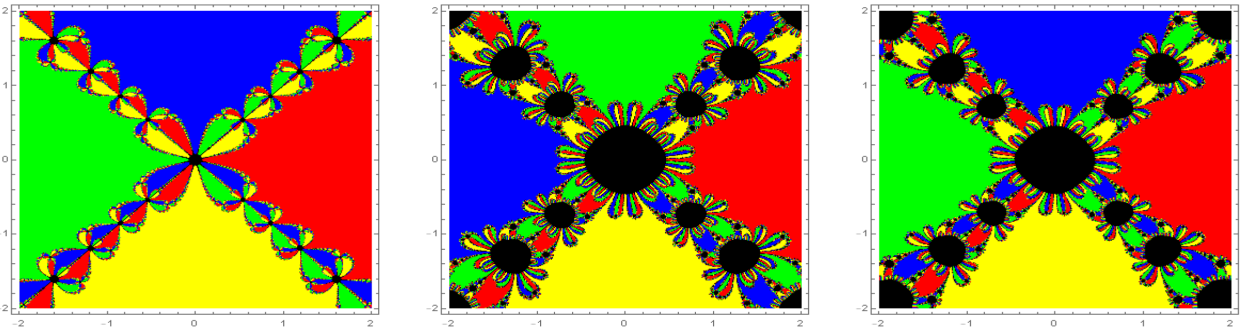
In Figures 6–8, the following six colors are employed to distinguish between the basins of attraction:

1) Red: Convergence to root z_1^*.
2) Green: Convergence to root z_2^*.
3) Blue: Convergence to root z_3^*.
4) Yellow: Convergence to root z_4^*.
5) Purple: Convergence to root z_5^* (if applicable).
6) Black: Divergent points that failed to converge within the 25-iteration limit.

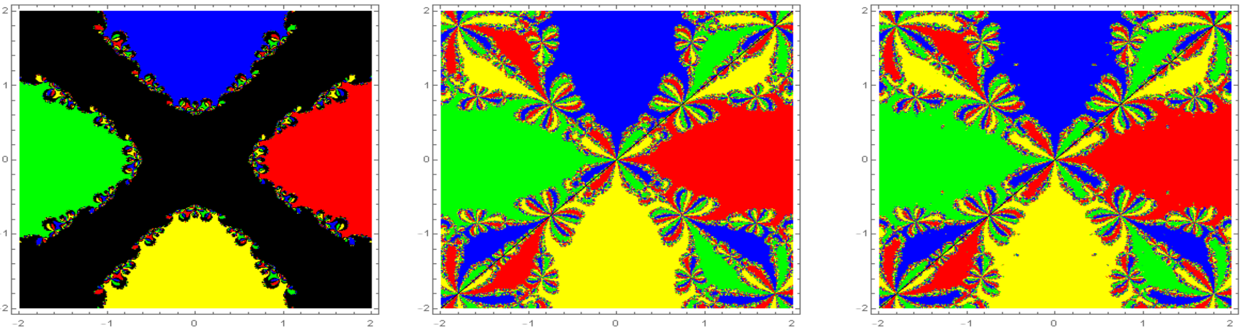


(a) NM
(b) CM
(c) A
(d) AG4
(e) AM2
(f) AM3

Figure 6. Basin of attraction of $K_1(z)$.

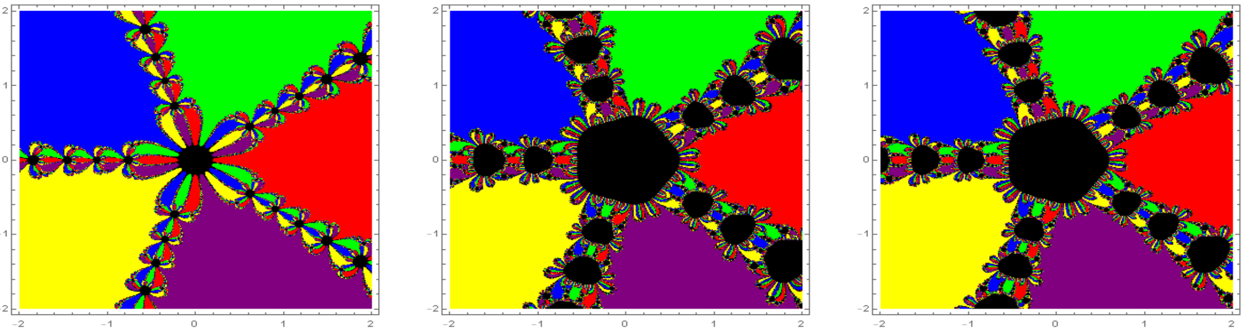


(a) NM (b) CM (c) A

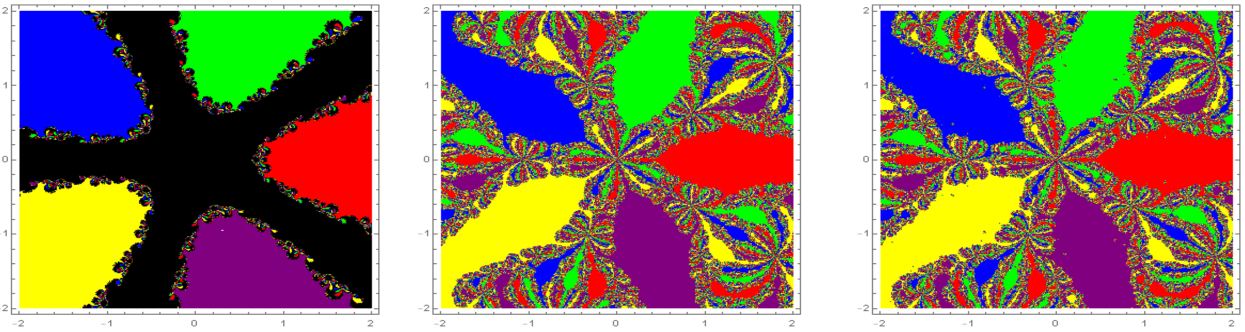


(d) AG4 (e) AM2 (f) AM3

Figure 7. Basin of attraction of $K_2(z)$.



(a) NM (b) CM (c) A



(d) AG4 (e) AM2 (f) AM3

Figure 8. Basin of attraction of $K_3(z)$.

The new proposed AM2 and AM3 techniques show faster convergence than traditional methods in terms of the number of iterations, such as Newton's method and its modifications. In addition to their higher orders of convergence (fourth and eighth, respectively), AM2 and AM3 can identify roots with less complexity. The basins of attraction generated by AM2 and AM3 are generally less sensitive to the choice of the initial guess and tend to show more intricate structures, particularly when the roots are close to each other. On the other hand, Newton's method and its modifications may quickly encounter black zones when the initial guess is far from the root or randomly chosen. Moreover, the AM2 and AM3 techniques handle such situations more effectively, converging with greater efficiency and without producing black zones. Even in critical cases, these methods are able to locate the correct root faster because they are generally more robust with respect to small variations in the initial guess.

6. Discussion and conclusions

The results of the research confirm the precision and dependability of predictor–corrector iteration techniques for identifying roots of nonlinear equations. Their high computational order of convergence (COC) demonstrates their effectiveness in solving nonlinear problems. Additionally, our methods have fewer sensitive basins of attraction; they not only converge faster than many commonly used techniques but also locate the root of nonlinear equations more accurately, as shown in the results.

In Tables 2–6, the results of the five suggested problems are gathered. The new proposed Algorithms perform better than the other methods; for example, Algorithm 2.5 is the best in terms of computing the order of convergence (COC), the error, the iteration count, and the magnitude of $\left|\mathscr{g}\left(x_{(s+1)}\right)\right|$. Also, our novel methods show high precision and high efficiency even though they only evaluate the function $\mathscr{g}(x_s)$ first derivative.

Showing fourth- and eighth-order convergence, respectively, the new modification of the Adomian decomposition approach used in this work facilitates the development of novel two-step and three-step iterative methods. Compared with traditional methods, the results demonstrate the clear advantages of the proposed methodology. Furthermore, the strong agreement between the theoretical analysis and computational results, in terms of efficiency and order of convergence, further confirms the effectiveness of the suggested techniques. We believe that this study will contribute significantly to the solution of a commonly encountered problem in mathematics and related fields. Finally, compared with other well-known methods, the proposed schemes demonstrate faster convergence, requiring fewer iterations and providing higher accuracy in approximating the roots of nonlinear equations. Both theoretical and numerical results confirm the effectiveness of the proposed methods, and they are expected to be useful to researchers and engineers in various scientific and engineering applications.

Author contributions

Ali Hasan Ali: Conceptualization, methodology, software, supervision, writing—original draft preparation. Iman A. Abdul Samad: Data curation, resources. Huda J. Saeed: Conceptualization, methodology, writing—original draft preparation. Firas Ghanim: Validation, writing—review and editing. Alina Alb Lupaș: Validation, funding acquisition, writing—review and editing. All authors have read and approved the final version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Funding

The publication of this paper was supported by the University of Oradea, Romania.

Conflict of interest

The authors declare no conflicts of interest.

References

1. G. Adomian, *Nonlinear stochastic systems theory and applications to physics*, Dordrecht: Kluwer Academic Publishers, 1989.
2. N. Y. Abdul-Hassan, A. H. Ali, C. Park, A new fifth-order iterative method free from second derivative for solving nonlinear equations, *J. Appl. Math. Comput.*, **68** (2021), 2877–2886. https://doi.org/10.1007/s12190-021-01647-1
3. S. Weerakoon, T. G. I. Fernando, A variant of Newton's method with accelerated third-order convergence, *Appl. Math. Lett.*, **13** (2000), 87–93. https://doi.org/10.1016/s0893-9659(00)00100-2
4. E. Babolian, J. Biazar, Solution of nonlinear equations by modified Adomian decomposition method, *Appl. Math. Comput.*, **132** (2002), 167–172. https://doi.org/10.1016/s0096-3003(01)00184-9
5. S. Abbasbandy, Improving Newton–Raphson method for nonlinear equations by modified Adomian decomposition method, *Appl. Math. Comput.*, **145** (2003), 887–893. https://doi.org/10.1016/s0096-3003(03)00282-0
6. S. M. Kang, W. Nazeer, M. Tanveer, Q. Mehmood, K. Rehman, Improvements in Newton–Raphson method for nonlinear equations using modified Adomian decomposition method, *Int. J. Math. Anal.*, **9** (2015), 1919–1928. https://doi.org/10.12988/ijma.2015.54124
7. M. A. Noor, K. I. Noor, Some iterative schemes for nonlinear equations, *Appl. Math. Comput.*, **183** (2006), 774–779. https://doi.org/10.1016/j.amc.2006.05.084
8. M. T. Darvishi, A. Barati, A third-order Newton-type method to solve systems of nonlinear equations, *Appl. Math. Comput.*, **187** (2007), 630–635. https://doi.org/10.1016/j.amc.2006.08.080
9. K. Abbaoui, Y. Cherruault, Convergence of Adomian's method applied to differential equations, *Comput. Math. Appl.*, **28** (1994), 103–109. https://doi.org/10.1016/0898-1221(94)00144-8
10. V. Daftardar-Gejji, H. Jafari, An iterative method for solving nonlinear functional equations, *J. Math. Anal. Appl.*, **316** (2006), 753–763. https://doi.org/10.1016/j.jmaa.2005.05.009
11. A. M. Wazwaz, A new algorithm for calculating Adomian polynomials for nonlinear operators, *Appl. Math. Comput.*, **111** (2000), 33–51. https://doi.org/10.1016/s0096-3003(99)00063-6

12. C. Chun, Iterative methods improving Newton's method by the decomposition method, *Comput. Math. Appl.*, **50** (2005), 1559–1568. https://doi.org/10.1016/j.camwa.2005.08.022
13. M. A. Noor, K. I. Noor, E. Al-Said, M. Waseem, Some new iterative methods for nonlinear equations, *Math. Probl. Eng.*, **2010** (2010), 198943. https://doi.org/10.1155/2010/198943
14. A. Rafiq, A. I. Pasha, B. S. Lee, New family of iterative methods for solving non-linear equations using new Adomian polynomials, *J. Korean Soc. Math. Educ. Ser. B*, **22** (2015), 231–243. https://doi.org/10.7468/jksmeb.2015.22.3.231
15. M. Saqib, M. Iqbal, Some multi-step iterative methods for solving nonlinear equations, *Open J. Math. Sci.*, **1** (2017), 25–33. https://doi.org/10.30538/oms2017.0003
16. F. Ali, W. Aslam, K. Ali, M. A. Anwar, A. Nadeem, New family of iterative methods for solving nonlinear models, *Discrete Dyn. Nat. Soc.*, **2018** (2018), 9619680. https://doi.org/10.1155/2018/9619680
17. G. Sana, M. A. Noor, K. I. Noor, Some multistep iterative methods for nonlinear equation using quadrature rule, *Int. J. Anal. Appl.*, **18** (2020), 920–938. https://doi.org/10.28924/2291-8639-18-2020-920
18. D. Ruan, X. Wang, Local and semilocal convergences of a fourth-order derivative-free method in Banach spaces and its applications, *Eng. Comput.*, **43** (2026), 1219–1239. https://doi.org/10.1108/ec-06-2025-0639
19. N. Shang, X. Wang, Y. Wang, Local convergence analysis of a novel Kurchatov-type derivative-free methods with and without memory for solving nonlinear systems, *Int. J. Comput. Methods*, **23** (2026), 2550033. https://doi.org/10.1142/s0219876225500331
20. J. Yu, X. Wang, A single parameter fourth-order Jarratt-type iterative method for solving nonlinear systems, *AIMS Mathematics*, **10** (2025), 7847–7863. https://doi.org/10.3934/math.2025360
21. S. Saber, B. Dridi, A. Alahmari, M. Messaoudi, Hyers–Ulam stability and control of fractional glucose–insulin systems, *Eur. J. Pure Appl. Math.*, **18** (2025), 6152. https://doi.org/10.29020/nybg.ejpam.v18i2.6152
22. M. Alhazmi, S. M. Mirgani, A. Alahmari, S. Saber, Application of LAPM and ABM methods to a fractional SCIR model of pneumonia diseases, *AIMS Mathematics*, **10** (2025), 25667–25707. https://doi.org/10.3934/math.20251137
23. S. Baroudi, M. Elomari, A. El Mfadel, A. Kassidi, Numerical solutions of the integro-partial fractional diffusion heat equation involving tempered ψ-Caputo fractional derivative, *J. Math. Sci.*, **271** (2023), 555–567. https://doi.org/10.1007/s10958-023-06640-6
24. S. Baroudi, A. El Mfadel, A. Kassidi, M. El Omari, A high-order compact scheme for nonlinear integro-differential equations with weakly singular kernels and tempered ψ-Caputo derivatives, *Math. Methods Appl. Sci.*, **48** (2025), 10774–10785. https://doi.org/10.1002/mma.10917
25. S. Baroudi, A. Kassidi, A. El Mfadel, M. Elomari, Coincidence degree theory for higher order nonlinear fractional differential equations: Existence and uniqueness results, *Commun. Nonlinear Sci. Numer. Simul.*, **147** (2025), 108847. https://doi.org/10.1016/j.cnsns.2025.108847
26. I. Aziz, Siraj-ul-Islam, W. Khan, Quadrature rules for numerical integration based on Haar wavelets and hybrid functions, *Comput. Math. Appl.*, **61** (2011), 2770–2781. https://doi.org/10.1016/j.camwa.2011.03.043
27. A. H. Ali, Z. Páles, Taylor-type expansions in terms of exponential polynomials, *Math. Inequal. Appl.*, **25** (2022), 1123–1141. https://doi.org/10.7153/mia-2022-25-69

28. H. J. Saeed, A. H. Ali, R. Menzer, A. D. Poțclean, H. Arora, New family of multi-step iterative methods based on homotopy perturbation technique for solving nonlinear equations, *Mathematics*, **11** (2023), 2603. https://doi.org/10.3390/math11122603
29. R. L. Burden, J. D. Faires, *Numerical analysis*, 10 Eds., Cengage Learning, 2011.
30. F. Ali, W. Aslam, I. Khalid, A. Nadeem, Iteration methods with an auxiliary function for nonlinear equations, *J. Math.*, **2020** (2020), 7356408. https://doi.org/10.1155/2020/7356408
31. H. J. Saeed, N. Y. Abdul-Hassan, An efficient three-step iterative methods based on Bernstein quadrature formula for solving nonlinear equations, *Basra J. Sci.*, **39** (2021), 355–383.