Hindawi
Journal of Mathematics
Volume 2020, Article ID 7356408, 15 pages
https://doi.org/10.1155/2020/7356408

Research Article

Iteration Methods with an Auxiliary Function for Nonlinear Equations

Faisal Ali, Waqas Aslam, Imran Khalid, and Akbar Nadeem

Centre for Advanced Studies in Pure and Applied Mathematics, Bahauddin Zakariya University, Multan 60800, Pakistan

Correspondence should be addressed to Waqas Aslam; waqas5210@yahoo.com

Received 8 June 2020; Revised 9 August 2020; Accepted 13 October 2020; Published 27 October 2020

Academic Editor: Antonio Di Crescenzo

Various iterative methods have been introduced by involving Taylor's series on the auxiliary function $g(x)$ to solve the nonlinear equation $f(x) = 0$. In this paper, we introduce the expansion of $g(x)$ with the inclusion of weights w_i such that $\sum_{i=1}^{p} w_i = 1$ and knots $\tau_i \in [0, 1]$ in order to develop a new family of iterative methods. The methods proposed in the present paper are applicable for different choices of auxiliary function $g(x)$, and some already known methods can be viewed as the special cases of these methods. We consider the diverse scientific/engineering models to demonstrate the efficiency of the proposed methods.

1. Introduction

Most of the problems in science and engineering involves nonlinear equation of the form $f(x) = 0$, where $f\colon D \subseteq R \longrightarrow R$ is a sufficiently smooth function in the neighborhood of a simple zero $\alpha \in D$. Many physical problems related to diverse areas such as biological applications in population dynamics and genetics where impulses arise naturally, motion of a particle on an inclined plane and projectile motion in physics, Van der Waals problem, and continuous stirred tank reactor equation in chemistry etc., can be modelled by nonlinear equations. Consequently, many numerical methods based on different techniques have been developed for solving nonlinear equations, see for example [1–12] and references therein. The following quadratically convergent Newton's method [13], which needs 2 function evaluations per iteration, is considered as the fundamental tool for obtaining the numerical solution of nonlinear equations:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, 3, \ldots, f'(x_n) \neq 0. \tag{1}$$

Definition 1 (see [12]).

Let p be the convergence order of the iterative method and q is the number of functional evaluations per iteration required by the method, then the efficiency index (IE) of the method is defined as

$$\mathrm{IE} = p^{(1/q)}. \tag{2}$$

Obviously, the efficiency index (IE) of Newton's method is 1.4142.

An appropriate selection of an initial guess makes Newton's method very efficient, whereas this method works ordinarily for an inappropriate initial guess. In second century BC, an ancient Chinese, Ying Buzu Shu, introduced a more useful method, which gives accurate results in some cases even where Newton's method does not work, for which two initial approximations are chosen. Using x_1 and x_2 as initial approximations, the next approximation can be obtained as

$$x_3 = \frac{x_2 f(x_1) - x_1 f(x_2)}{f(x_1) - f(x_2)}. \tag{3}$$

During the last two decades, ancient Chinese algorithms have been a scorching topic for many researchers [14–16]. In 2016, another Chinese He [17] introduced two modifications (Bubbfil Algorithms 1 and 2) of the above method with fast convergence.

Several modifications of Newton's method have also been introduced, e.g., [5, 12, 18, 19]. Weerakon and Fernando [20] constructed the following useful cubically convergent method:

$$x_{n+1} = x_n - \frac{2f(x_n)}{f'(x_n) + f'(y_n)},$$
$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{4}$$

The midpoint rule [19, 21],

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n - (f(x_n)/(2f'(x_n))))}, \tag{5}$$

with an order of convergence 3, is also a well-known technique. There exist several other third- and fourth-order methods involving diverse techniques for nonlinear equations, e.g., [5, 18, 22, 23]. Abbasbandy [1] used the Adomian decomposition method (ADM) [2] to find the simple root of nonlinear equations. But employing ADM involves higher order derivatives of Adomian polynomials which is a major weakness of ADM. But this is not the case in the decomposition technique due to Cherruault [24, 25]. The technique of Cherruault has extensively been used to develop some useful algorithms for solving nonlinear equations [3, 9, 26]. Recently, Shah et al. [27], using the same technique and an auxiliary function $g(x)$ along with the expansion of the original equation $f(x) = 0$, have introduced a family of iterative methods for nonlinear equations.

In the present paper, making use of the decomposition technique of [24, 25], together with the expansions of both $f(x)$ and $g(x)$, we construct a new family of swiftly convergent iterative methods. In order to exhibit the efficiency of proposed methods, we present numerical as well as graphical analysis by considering three mathematical models from different branches of science, i.e., physics, mathematics, and chemistry.

2. Construction of Methods

Assume the nonlinear equation:

$$f(x) = 0, \tag{6}$$

with a simple zero α and the initial guess γ sufficiently close to α. We take $g(x)$ as an auxiliary function, such that $g(x)$ is sufficiently smooth and $g(\alpha) \neq 0$,

$$f(x)g(x) = 0. \tag{7}$$

Using the quadrature formula and the fundamental law of calculus, the following coupled system can be formed:

$$f(\gamma)g(\gamma) + (x-\gamma)[f(\gamma)K(g', x, \gamma) + g(\gamma)Q(f', x, \gamma)] + h(x) = 0, \tag{8}$$

$$h(x) = f(x)g(x) - f(\gamma)g(\gamma) - (x-\gamma)[f(\gamma)K(g', x, \gamma) + g(\gamma)Q(f', x, \gamma)], \tag{9}$$

where $K(g', x, \gamma)$ and $Q(f', x, \gamma)$ represents $\sum_{i=1}^{p} w_i g'(\gamma + \tau_i(x-\gamma))$ and $\sum_{i=1}^{p} w_i f'(\gamma + \tau_i(x-\gamma))$ respectively; w_i and τ_i are weights and knots, respectively, such that $\sum_{i=1}^{p} w_i = 1$ and $\tau_i \in [0, 1]$.

We write equation (8) in terms of nonlinear operator $N(x)$ as follows:

$$x = \gamma - \frac{f(\gamma)g(\gamma) + h(x)}{f(\gamma)K(g', x, \gamma) + g(\gamma)Q(f', x, \gamma)} = c + N(x), \tag{10}$$

where

$$c = \gamma, \tag{11}$$

$$N(x) = -\frac{f(\gamma)g(\gamma) + h(x)}{f(\gamma)K(g', x, \gamma) + g(\gamma)Q(f', x, \gamma)}. \tag{12}$$

We construct a sequence of higher order iterative methods by decomposing $N(x)$ in the following way [24, 25]:

$$N(x) = N\left(\sum_{i=0}^{\infty} x_i\right) = N(x_0) + \sum_{i=1}^{\infty} \cdot \left\{N\left(\sum_{j=0}^{i} x_j\right) - N\left(\sum_{j=0}^{i-1} x_j\right)\right\}. \tag{13}$$

Our purpose is to seek the solution in the form of the following series:

$$x = \sum_{i=0}^{\infty} x_i. \tag{14}$$

Combining equations (10), (13), and (14), we get

$$\sum_{i=0}^{\infty} x_i = c + N(x_0) + \sum_{i=1}^{\infty}\left\{N\left(\sum_{j=0}^{i} x_j\right) - N\left(\sum_{j=0}^{i-1} x_j\right)\right\}. \tag{15}$$

Thus, we have the following iterative scheme:

$$\begin{aligned} x_0 &= c, \\ x_1 &= N(x_0), \\ x_2 &= N(x_0 + x_1) - N(x_0), \\ &\vdots \\ x_{n+1} &= N(x_0 + x_1 + \cdots + x_n) - N(x_0 + x_1 + \cdots + x_{n-1}), \quad n = 1, 2, \ldots . \end{aligned} \tag{16}$$

Thus,

$$\begin{aligned} x_1 + x_2 + \cdots + x_{n+1} &= N(x_0 + x_1 + \cdots + x_n), \quad n = 1, 2, 3, \ldots, \\ x &= c + \sum_{i=1}^{\infty} x_i. \end{aligned} \tag{17}$$

From equations (11), (12), and (16), we have

$$\begin{aligned} x_0 &= c = \gamma, \\ x_1 &= N(x_0) \\ &= -\frac{f(\gamma)g(\gamma) + h(x_0)}{f(\gamma)K(g', x, \gamma) + g(\gamma)Q(f', x, \gamma)}. \end{aligned} \tag{18}$$

Using equation (9), we get

$$x_1 = -\frac{f(\gamma)g(\gamma)}{f(\gamma)g'(\gamma) + g(\gamma)f'(\gamma)}. \tag{19}$$

We note that x is approximated as

$$X_m = x_0 + x_1 + \cdots + x_m, \tag{20}$$

where

$$\lim_{m \longrightarrow \infty} X_m = x. \tag{21}$$

For $m = 1$, we have

$$x \approx X_1 = x_0 + x_1 = \gamma - \frac{f(\gamma)g(\gamma)}{f(\gamma)g'(\gamma) + g(\gamma)f'(\gamma)}. \tag{22}$$

Thus, the following recurrence relation can be formed by using equation (22).

Algorithm 1 (28).

$$x_{n+1} = x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + f'(x_n)g(x_n)}, \quad n = 0, 1, 2, \ldots, \tag{23}$$

i.e., corresponding to the initial approximation x_0, $(n+1)$st approximation can be determined. It is notable that the above algorithm is actually Newton's method applied to $f(x)g(x)$.

Now, for $m = 2$, equation (20) gives

$$x \approx X_2 = x_0 + x_1 + x_2. \tag{24}$$

Using equation (12), we have

$$N(x_0 + x_1) = -\frac{f(\gamma)g(\gamma) + h(x_0 + x_1)}{f(\gamma)K(g', x_0 + x_1, \gamma) + g(\gamma)Q(f', x_0 + x_1, \gamma)}. \tag{25}$$

Using equations (9), (16), (22), and (24), we get

$$\begin{aligned} x &\approx X_2 = x_0 + x_1 + x_2, \\ &= \gamma - \frac{f(\gamma)g(\gamma)}{f(\gamma)g'(\gamma) + g(\gamma)f'(\gamma)} - \frac{f(x_0 + x_1)g(x_0 + x_1)}{f(\gamma)K(g', x_0 + x_1, \gamma) + g(\gamma)Q(f', x_0 + x_1, \gamma)}. \end{aligned} \tag{26}$$

Thus, the following recurrence relation can be formed using equation (26).

Algorithm 2

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \quad n = 0, 1, 2, \ldots, \\ x_{n+1} &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)\sum_{i=1}^{p} w_i g'(x_n + \tau_i(y_n - x_n)) + g(x_n)\sum_{i=1}^{p} w_i f'(x_n + \tau_i(y_n - x_n))}, \end{aligned} \tag{27}$$

which is a new predictor-corrector method, and corresponding to initial approximation x_0, $(n+1)$st approximation can be determined.

Now, for $m=3$, equation (20) gives

$$x \approx X_3 = x_0 + x_1 + x_2 + x_3. \tag{28}$$

Using equation (12), we have

$$N(x_0+x_1+x_2) = -\frac{f(\gamma)g(\gamma) + h(x_0+x_1+x_2)}{f(\gamma)K(g', x_0+x_1+x_2, \gamma) + g(\gamma)Q(f', x_0+x_1+x_2, \gamma)}. \tag{29}$$

Using equations (9), (16), (26), and (28), we get

$$\begin{aligned} x \approx X_3 &= x_0 + x_1 + x_2 + x_3 \\ &= \gamma - \frac{f(\gamma)g(\gamma)}{f(\gamma)g'(\gamma) + g(\gamma)f'(\gamma)} - \frac{f(x_0+x_1)g(x_0+x_1)}{f(\gamma)K(g', x_0+x_1, \gamma) + g(\gamma)Q(f', x_0+x_1, \gamma)} - \\ &\quad \cdot \frac{f(x_0+x_1+x_2)g(x_0+x_1+x_2)}{f(\gamma)K(g', x_0+x_1+x_2, \gamma) + g(\gamma)Q(f', x_0+x_1+x_2, \gamma)}. \end{aligned} \tag{30}$$

Thus, the following recurrence relation can be formed using equation (30).

Algorithm 3

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \quad n = 0, 1, 2, \ldots, \\ z_n &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)\sum_{i=1}^{p} w_i g'(x_n + \tau_i(y_n - x_n)) + g(x_n)\sum_{i=1}^{p} w_i f'(x_n + \tau_i(y_n - x_n))}, \\ x_{n+1} &= z_n - \frac{f(z_n)g(z_n)}{f(x_n)\sum_{i=1}^{p} w_i g'(x_n + \tau_i(z_n - x_n)) + g(x_n)\sum_{i=1}^{p} w_i f'(x_n + \tau_i(z_n - x_n))}, \end{aligned} \tag{31}$$

which is also a new three-step iterative technique.

2.1. Some Special Cases of Algorithm 2

Algorithm 4. Taking $p = 1, w_1 = 1$, and $\tau_1 = 0$, Algorithm 2 reduces to the following iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ x_{n+1} &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{32}$$

Shah and Noor [28] have established the above formula.

Algorithm 5. Taking $p = 1, w_1 = 1$, and $\tau_1 = 1$, Algorithm 2 reduces to the following iterative method.

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ x_{n+1} &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)g'(y_n) + g(x_n)f'(y_n)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{33}$$

Algorithm 6. Taking $p = 1, w_1 = 1$, and $\tau_1 = (1/2)$, Algorithm 2 reduces to the following iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ x_{n+1} &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)g'((y_n + x_n)/2) + g(x_n)f'((y_n + x_n)/2)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{34}$$

Algorithm 7. Taking $p = 2, w_1 = (1/4), w_2 = (3/4), \tau_1 = 0$, and $\tau_2 = (2/3)$, Algorithm 2 reduces to the following iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ x_{n+1} &= y_n - \frac{4f(y_n)g(y_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n) + 3f(x_n)g'((x_n+2y_n)/3) + 3g(x_n)f'((x_n+2y_n)/3)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{35}$$

Algorithm 8. Taking $p = 2, w_1 = (1/2), w_2 = (1/2), \tau_1 = 0$, and $\tau_2 = (1/2)$, Algorithm 2 reduces to the following iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ x_{n+1} &= y_n - \frac{2f(y_n)g(y_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n) + f(x_n)g'((x_n+y_n)/2) + g(x_n)f'((x_n+y_n)/2)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{36}$$

To the best of our knowledge, Algorithms 5–8 are new methods having convergence orders 3, 3, 4, and 4, respectively.

2.2. Some Special Cases of Algorithm 3

Algorithm 9. Taking $p = 1, w_1 = 1$, and $\tau_1 = 0$, Algorithm 3 reduces to the following iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)g(z_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{37}$$

Algorithm 10. Taking $p = 1, w_1 = 1$, and $\tau_1 = 1$, Algorithm 3 reduces to the following iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)g'(y_n) + g(x_n)f'(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)g(z_n)}{f(x_n)g'(z_n) + g(x_n)f'(z_n)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{38}$$

Algorithm 11. Taking $p = 1, w_1 = 1$, and $\tau_1 = (1/2)$, Algorithm 3 reduces to the following iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n) + g(x_n)f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)g'((y_n+x_n)/2) + g(x_n)f'((y_n+x_n)/2)}, \\ x_{n+1} &= z_n - \frac{f(z_n)g(z_n)}{f(x_n)g'((z_n+x_n)/2) + g(x_n)f'((z_n+x_n)/2)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{39}$$

Algorithm 12. Taking $p=2, w_1=(1/4), w_2=(3/4), \tau_1=0$, and $\tau_2=(2/3)$, Algorithm 3 reduces to the following iterative method:

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n)+g(x_n)f'(x_n)},\\
z_n &= y_n - \frac{4f(y_n)g(y_n)}{f(x_n)g'(x_n)+g(x_n)f'(x_n)+3f(x_n)g'((x_n+2y_n)/3)+3g(x_n)f'((x_n+2y_n)/3)},\\
x_{n+1} &= z_n - \frac{4f(z_n)g(z_n)}{f(x_n)g'(x_n)+g(x_n)f'(x_n)+3f(x_n)g'((x_n+2z_n)/3)+3g(x_n)f'((x_n+2z_n)/3)}, \quad n=0,1,2,\ldots.
\end{aligned}
\tag{40}
$$

Algorithm 13. Taking $p=2, w_1=(1/2), w_2=(1/2), \tau_1=0$, and $\tau_2=(1/2)$, Algorithm 3 reduces to the following iterative method:

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)g(x_n)}{f(x_n)g'(x_n)+g(x_n)f'(x_n)},\\
z_n &= y_n - \frac{2f(y_n)g(y_n)}{f(x_n)g'(x_n)+g(x_n)f'(x_n)+f(x_n)g'((x_n+y_n)/2)+g(x_n)f'((x_n+y_n)/2)},\\
x_{n+1} &= z_n - \frac{2f(z_n)g(z_n)}{f(x_n)g'(x_n)+g(x_n)f'(x_n)+f(x_n)g'((x_n+z_n)/2)+g(x_n)f'((x_n+z_n)/2)}, \quad n=0,1,2,\ldots.
\end{aligned}
\tag{41}
$$

To the best of our knowledge, Algorithms 10–13 are new iterative methods for solving nonlinear equation (6) with convergence orders $3, 4, 4$, and 4, respectively.

Obviously, different iterative methods can be obtained from Algorithms 2 and 3 by using distinct values of the auxiliary function $g(x)$. Suitable selection of auxiliary function $g(x)$ plays a significant role for the better performance of these methods. To demonstrate the efficiency of our proposed algorithms, we choose $g(x)=I(x)$. Thus, corresponding to this choice, algorithm $10+i$, $i=4,5,6,\ldots,13$, is the special case of algorithm i.

Algorithm 14

$$
\begin{aligned}
y_n &= x_n - \frac{x_n f(x_n)}{x_n f'(x_n)+f(x_n)},\\
x_{n+1} &= y_n - \frac{y_n f(y_n)}{x_n f'(x_n)+f(x_n)}, \quad n=0,1,2,\ldots.
\end{aligned}
\tag{42}
$$

The convergence order of the method described in Algorithm 14 is 3, and the total number of evaluations per iteration is 3. Thus, IE = 1.4422.

Algorithm 15

$$
\begin{aligned}
y_n &= x_n - \frac{x_n f(x_n)}{x_n f'(x_n)+f(x_n)},\\
x_{n+1} &= y_n - \frac{y_n f(y_n)}{x_n f'(y_n)+f(x_n)}, \quad n=0,1,2,\ldots.
\end{aligned}
\tag{43}
$$

The convergence order of the method described in Algorithm 15 is 3, and the total number of evaluations per iteration is 4. Thus, IE = 1.3161.

Algorithm 16

$$
\begin{aligned}
y_n &= x_n - \frac{x_n f(x_n)}{x_n f'(x_n)+f(x_n)},\\
x_{n+1} &= y_n - \frac{y_n f(y_n)}{x_n f'((x_n+y_n)/2)+f(x_n)}, \quad n=0,1,2,\ldots.
\end{aligned}
\tag{44}
$$

The convergence order of the method described in Algorithm 16 is 3, and the total number of evaluations per iteration is 4. Thus, IE = 1.3161.

Algorithm 17

$$y_n = x_n - \frac{x_n f(x_n)}{x_n f'(x_n) + f(x_n)},$$
$$x_{n+1} = y_n - \frac{4 y_n f(y_n)}{3 x_n f'((x_n + 2y_n)/3) + x_n f'(x_n) + 4 f(x_n)}, \quad n = 0, 1, 2, \ldots. \tag{45}$$

The convergence order of the method described in Algorithm 17 is 3, and the total number of evaluations per iteration is 4. Thus, IE = 1.3161.

Algorithm 18

$$y_n = x_n - \frac{x_n f(x_n)}{x_n f'(x_n) + f(x_n)},$$
$$x_{n+1} = y_n - \frac{2 y_n f(y_n)}{x_n f'((x_n + y_n)/2) + x_n f'(x_n) + 2 f(x_n)}, \quad n = 0, 1, 2, \ldots. \tag{46}$$

The convergence order of the method described in Algorithm 18 is 3, and the total number of evaluations per iteration is 4. Thus, IE = 1.3161.

Algorithm 19

$$y_n = x_n - \frac{x_n f(x_n)}{x_n f'(x_n) + f(x_n)},$$
$$z_n = y_n - \frac{y_n f(y_n)}{x_n f'(x_n) + f(x_n)},$$
$$x_{n+1} = z_n - \frac{z_n f(z_n)}{x_n f'(x_n) + f(x_n)}, \quad n = 0, 1, 2, \ldots. \tag{47}$$

The convergence order of the method described in Algorithm 19 is 4, and the total number of evaluations per iteration is 4. Thus, IE = 1.4142.

Algorithm 20

$$y_n = x_n - \frac{x_n f(x_n)}{x_n f'(x_n) + f(x_n)},$$
$$z_n = y_n - \frac{y_n f(y_n)}{x_n f'(y_n) + f(x_n)},$$
$$x_{n+1} = z_n - \frac{z_n f(z_n)}{x_n f'(z_n) + f(x_n)}, \quad n = 0, 1, 2, \ldots. \tag{48}$$

The convergence order of the method described in Algorithm 20 is 4, and the total number of evaluations per iteration is 6. Thus, IE = 1.2599.

Algorithm 21

$$y_n = x_n - \frac{x_n f(x_n)}{x_n f'(x_n) + f(x_n)},$$
$$z_n = y_n - \frac{y_n f(y_n)}{x_n f'((x_n + y_n)/2) + f(x_n)},$$
$$x_{n+1} = z_n - \frac{z_n f(z_n)}{x_n f'((x_n + z_n)/2) + f(x_n)}, \quad n = 0, 1, 2, \ldots. \tag{49}$$

The convergence order of the method described in Algorithm 21 is 4, and the total number of evaluations per iteration is 6. Thus, IE = 1.2599.

Algorithm 22

$$y_n = x_n - \frac{x_n f(x_n)}{x_n f'(x_n) + f(x_n)},$$
$$z_n = y_n - \frac{4 y_n f(y_n)}{3 x_n f'((x_n + 2y_n)/3) + x_n f'(x_n) + 4 f(x_n)},$$
$$x_{n+1} = z_n - \frac{4 z_n f(z_n)}{3 x_n f'((x_n + 2z_n)/3) + x_n f'(x_n) + 4 f(x_n)}, \quad n = 0, 1, 2, \ldots. \tag{50}$$

The convergence order of the method described in Algorithm 22 is 4, and the total number of evaluations per iteration is 6. Thus, IE = 1.2599.

Algorithm 23

$$y_n = x_n - \frac{x_n f(x_n)}{x_n f'(x_n) + f(x_n)},$$
$$z_n = y_n - \frac{2 y_n f(y_n)}{x_n f'((x_n + y_n)/2) + x_n f'(x_n) + 2 f(x_n)}, \tag{51}$$
$$x_{n+1} = z_n - \frac{2 z_n f(z_n)}{x_n f'((x_n + z_n)/2) + x_n f'(x_n) + 2 f(x_n)}, \quad n = 0, 1, 2, \ldots.$$

The convergence order of the method described in Algorithm 23 is 4, and the total number of evaluations per iteration is 6. Thus, IE = 1.2599.

3. Convergence Analysis

In this section, convergence criteria of proposed algorithms are studied in the form of the following theorem.

Theorem 1. *Assume that the function $f\colon I \subset \mathbb{R} \longrightarrow \mathbb{R}$ on an open interval I has a simple root $\alpha \in I$. Let $f(x)$ be sufficiently differentiable in the neighborhood of α, then the convergence orders of the methods defined by Algorithms 2 and 3 are three and four, respectively.*

Proof. Let α be a simple zero of $f(x)$. Since f is sufficiently differentiable, the Taylor series of $f(x_n)$ and $f'(x_n)$ about α are given by

$$\begin{aligned} f(x_n) &= f'(\alpha)\left[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + O\left(e_n^6\right)\right],\\ f'(x_n) &= f'(\alpha)\left[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_4^3 + 5c_5 e_n^4 + 6c_6 e_n^5 \right.\\ &\quad \left. + O\left(e_n^6\right)\right], \end{aligned} \tag{52}$$

where $e_n = x_n - \alpha$ and $c_j = (1/j!)(f^{(j)}(\alpha)/f'(\alpha))$, $j = 2, 3, \ldots$

Expanding $f(x_n)g(x_n)$, $f'(x_n)g(x_n)$ and $f(x_n)g'(x_n)$ by Taylor series, we get

$$f(x_n)g(x_n) = f'(\alpha)\left[g(\alpha)e_n + (c_2 g(\alpha) + g'(\alpha))e_n^2 + \left(\frac{1}{2}g''(\alpha) + c_2 g'(\alpha) + c_3 g(\alpha)\right)e_n^3 + O\left(e_n^4\right)\right], \tag{53}$$

$$f'(x_n)g(x_n) = f'(\alpha)\begin{bmatrix} g(\alpha) + (2c_2 g(\alpha) + g'(\alpha))e_n + \left(\frac{1}{2}g''(\alpha) + 2c_2 g'(\alpha) + 3c_3 g(\alpha)\right)e_n^2 + \\ \left(\frac{1}{6}g'''(\alpha) + c_2 g''(\alpha) + 3c_3 g'(\alpha) + 4c_4 g(\alpha)\right)e_n^3 + O\left(e_n^4\right) \end{bmatrix}, \tag{54}$$

$$f(x_n)g'(x_n) = f'(\alpha)\left[g'(\alpha)e_n + (c_2 g'(\alpha) + g''(\alpha))e_n^2 + \left(\frac{1}{2}g'''(\alpha) + c_2 g''(\alpha) + c_3 g'(\alpha)\right)e_n^3 + O\left(e_n^4\right)\right]. \tag{55}$$

From equations (53), (54), and (55), we get

$$\frac{f(x_n)g(x_n)}{f'(x_n)g(x_n) + f(x_n)g'(x_n)} = \left[e_n - \left(c_2 + \frac{g'(\alpha)}{g(\alpha)}\right)e_n^2 + \begin{pmatrix} 2c_2^2 - 2c_3 \\ + \frac{2c_2 g'(\alpha)}{g(\alpha)} - \frac{g''(\alpha)}{g(\alpha)} + \frac{2g'(\alpha)^2}{g(\alpha)^2} \end{pmatrix} e_n^3 + O\left(e_n^4\right)\right]. \tag{56}$$

Using equation (56), we have

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{f'(x_n)g(x_n) + f(x_n)g'(x_n)} \\ &= \alpha + \left(c_2 + \frac{g'(\alpha)}{g(\alpha)}\right)e_n^2 + \left(2c_3 - 2c_2^2 - \frac{2c_2 g'(\alpha)}{g(\alpha)} + \frac{g''(\alpha)}{g(\alpha)} - \frac{2g'(\alpha)^2}{g(\alpha)^2}\right)e_n^3 + O\left(e_n^4\right). \end{aligned} \tag{57}$$

The Taylor series of $f(y_n)$ and $g(y_n)$ are given as

$$\begin{aligned} f(y_n) &= f'(\alpha)\left[\left(c_2 + \frac{g'(\alpha)}{g(\alpha)}\right)e_n^2 + \left(2c_3 - 2c_2^2 - \frac{2c_2 g'(\alpha)}{g(\alpha)} + \frac{g''(\alpha)}{g(\alpha)} - \frac{2g'(\alpha)^2}{g(\alpha)^2}\right)e_n^3 + O\left(e_n^4\right)\right], \\ g(y_n) &= g(\alpha) + Ag'(\alpha) + \frac{A^2 g''(\alpha)}{2!} + \frac{A^3 g'''(\alpha)}{3!} + \cdots, \end{aligned} \tag{58}$$

where

$$A = \left(c_2 + \frac{g'(\alpha)}{g(\alpha)}\right)e_n^2 + \left(2c_3 - 2c_2^2 - \frac{2c_2 g'(\alpha)}{g(\alpha)} - \frac{2g'(\alpha)^2}{g(\alpha)^2} + \frac{g''(\alpha)}{g(\alpha)}\right)e_n^3 + O\left(e_n^4\right). \tag{59}$$

Let $v_n = x_n + \tau_i(y_n - x_n)$. Taylor's expansions of $g(v_n), g'(v_n)$ and $f'(v_n)$ are given by

$$\begin{aligned} g(v_n) &= \left[\begin{array}{c} g(\alpha) + g'(\alpha)(1-\tau_i)e_n + \left(\dfrac{g'(\alpha)^2\tau_i}{g(\alpha)} + g'(\alpha)\tau_i c_2 + \dfrac{g''(\alpha)\tau_i^2}{2} - g''(\alpha)\tau_i + \dfrac{g''(\alpha)}{2}\right)e_n^2 + \\ \left\{\begin{array}{c} 2\tau_i\left(c_3 - c_2^2\right)g'(\alpha) + \tau_i c_2(1-\tau_i)g''(\alpha) + \dfrac{g'''(\alpha)}{6}\left(1 - 3\tau_i + 3\tau_i^2 - \tau_i^3\right) + \\ \dfrac{\tau_i}{g(\alpha)}\left(2g'(\alpha)g''(\alpha) - 2c_2 g'(\alpha)^2 - \dfrac{2g'(\alpha)^3}{g(\alpha)} - \tau_i g'(\alpha)g''(\alpha)\right) \end{array}\right\}e_n^3 \end{array}\right] + O\left(e_n^4\right), \\ g'(v_n) &= \left[g'(\alpha) + g''(\alpha)(1-\tau_i)e_n + \left\{\tau_i g''(\alpha)\left(c_2 + \frac{g'(\alpha)}{g(\alpha)}\right) + \frac{g'''(\alpha)}{2}(1-\tau_i)^2\right\}e_n^2 + \right] + O\left(e_n^3\right), \\ f'(v_n) &= f'(\alpha)\left[\begin{array}{c} 1 + 2c_2(1-\tau_i)e_n + \left(2\tau_i\left(c_2 + \dfrac{g'(\alpha)}{g(\alpha)}\right)c_2 + 3c_3\left(\tau_i^2 - 2\tau_i + 1\right)\right)e_n^2 + \\ \left\{\begin{array}{c} \dfrac{2\tau_i g'(\alpha)}{g(\alpha)}\left(3c_3 - 2c_2^2 - 3\tau_i c_3 - \dfrac{2c_2 g'(\alpha)}{g(\alpha)}\right) + \\ 2c_2\tau_i\left(5c_3 - 2c_2^2 - 3\tau_i c_3 + \dfrac{g''(\alpha)}{g(\alpha)}\right) + 4c_4\left(1 - 3\tau_i + 3\tau_i^2 - \tau_i^3\right) \end{array}\right\}e_n^3 + O\left(e_n^4\right) \end{array}\right]. \end{aligned} \tag{60}$$

Now, expanding $f(y_n)g(y_n), f(x_n)g'(v_n)$ and $f'(v_n)g(x_n)$ by Taylor series, we get

$$f(y_n)g(y_n) = f'(\alpha)\begin{bmatrix} (g'(\alpha) + g(\alpha)c_2)e_n^2 + \\ \left(g''(\alpha) - 2g'(\alpha)c_2 + 2g(\alpha)c_3 - \dfrac{2g'(\alpha)^2}{g(\alpha)} - 2g(\alpha)c_2^2\right)e_n^3 + O\left(e_n^4\right)\end{bmatrix}, \tag{61}$$

$$f(x_n)g'(v_n) = f'(\alpha)\begin{bmatrix} g'(\alpha)e_n + (g''(\alpha)(1-\tau_i) + c_2 g'(\alpha))e_n^2 + \\ \left\{g'(\alpha)\left(c_3 + \dfrac{g''(\alpha)\tau_i}{g(\alpha)}\right) + c_2 g''(\alpha) + \dfrac{g'''(\alpha)}{2}(1-\tau_i)^2\right\}e_n^3 + O\left(e_n^4\right)\end{bmatrix}, \tag{62}$$

$$f'(v_n)g(x_n) = f'(\alpha)\begin{bmatrix} g(\alpha) + (g'(\alpha) - 2g(\alpha)\tau_i c_2 + 2g(\alpha)c_2)e_n + \\ \left(g(\alpha)\left(2\tau_i c_2^2 + 3\tau_i^2 c_3 - 6\tau_i c_3 + 3c_3\right) + 2c_2 g'(\alpha) + \dfrac{g''(\alpha)}{2}\right)e_n^2 + \\ \left\{\begin{matrix} g(\alpha)\left(4c_4 - 4c_2^3\tau_i - 6c_2c_3\tau_i^2 - 4c_4\tau_i^3 + 10c_2c_3\tau_i + 12c_4\tau_i^2 - 12c_4\tau_i\right) + \\ g'(\alpha)\left(3c_3 - 3c_3\tau_i^2 - 2c_2^2\tau_i\right) - \dfrac{2c_2\tau_i g'(\alpha)^2}{g(\alpha)} \\ +c_2(1+\tau_i)g''(\alpha) + \dfrac{1}{6}g'''(\alpha) \end{matrix}\right\}e_n^3 + O\left(e_n^4\right)\end{bmatrix}. \tag{63}$$

Using equations (61)–(63), we obtain

$$\frac{f(y_n)g(y_n)}{f(x_n)g'(v_n) + f'(v_n)g(x_n)} = \left[\left(\frac{g'(\alpha)}{g(\alpha)} + c_2\right)e_n^2 + \begin{pmatrix} \dfrac{2c_2 g'(\alpha)}{g(\alpha)}(\tau_i - 3) - \dfrac{4g'(\alpha)^2}{g(\alpha)^2} + \dfrac{g''(\alpha)}{g(\alpha)} + \\ 2c_2^2(\tau_i - 2) + 2c_3 \end{pmatrix}e_n^3\right] + O\left(e_n^4\right). \tag{64}$$

Using equation (64) and the fact $\sum_{i=1}^{p} w_i = 1$, the error term for Algorithm 2 is as follows:

$$\begin{aligned} z_n &= y_n - \frac{f(y_n)g(y_n)}{f(x_n)g'(v_n) + f'(v_n)g(x_n)} \\ &= \alpha + \left[\frac{2c_2 g'(\alpha)}{g(\alpha)}(2-\tau_i) + \frac{2g'(\alpha)^2}{g(\alpha)^2} + 2c_2^2(1-\tau_i)\right]e_n^3 + O\left(e_n^4\right). \end{aligned} \tag{65}$$

The last expression shows that the convergence order of Algorithm 2 is 3.

The Taylor series expansions of $f(z_n)$ and $g(z_n)$ are given by

$$f(z_n) = f'(\alpha)\left[\left(\frac{2c_2 g'(\alpha)}{g(\alpha)}(2-\tau_i) + \frac{2g'(\alpha)^2}{g(\alpha)^2} + 2c_2^2(1-\tau_i)\right)e_n^3 + O\left(e_n^4\right)\right],$$
$$g(z_n) = g(\alpha) + Bg'(\alpha) + \frac{B^2 g''(\alpha)}{2!} + \frac{B^3 g'''(\alpha)}{3!} + \cdots, \tag{66}$$

where

$$B = \left[\frac{4c_2 g'(\alpha)}{g(\alpha)} + \frac{2g'(\alpha)^2}{g(\alpha)^2} - \frac{2g'(\alpha)c_2\tau_i}{g(\alpha)} + 2c_2^2 - 2c_2^2\tau_i\right]e_n^3 + O\left(e_n^4\right). \tag{67}$$

Let $u_n = x_n + \tau_i(z_n - x_n)$. Thus, Taylor's expansions of $g(u_n)$ and $g'(u_n)$ are given by

$$g(u_n) = g(\alpha) + g'(\alpha)(1-\tau_i)e_n + \frac{g''(\alpha)}{2}(1-\tau_i)^2 e_n^2$$
$$+\left\{2c_2^2\tau_i g'(\alpha)(1-\tau_i) + \frac{2c_2\tau_i g'(\alpha)^2}{g(\alpha)}(2-\tau_i) + \frac{2\tau_i g'(\alpha)^3}{g(\alpha)^2} + \frac{g'''(\alpha)}{6}\left(1-3\tau_i+3\tau_i^2-\tau_i^3\right)\right\}e_n^3 + O\left(e_n^4\right),$$
$$g'(u_n) = \left[\begin{array}{c} g'(\alpha) + g''(\alpha)(1-\tau_i)e_n + \dfrac{g'''(\alpha)}{2}(1-\tau_i)^2 e_n^2 + \\ \left\{\begin{array}{c} \dfrac{2c_2\tau_i g'(\alpha)g''(\alpha)}{g(\alpha)}(2-\tau_i) + \dfrac{2\tau_i g'(\alpha)^2 g''(\alpha)}{g(\alpha)^2} + 2c_2^2\tau_i(1-\tau_i)g''(\alpha) + \\ + \dfrac{g''''(\alpha)}{6}\left(1-3\tau_i+3\tau_i^2-\tau_i^3\right) \end{array}\right\}e_n^3 \end{array}\right] + O\left(e_n^4\right). \tag{68}$$

Now, expanding $f(z_n)g(z_n), f(x_n)g'(u_n)$ and $f'(u_n)g(x_n)$ by Taylor series, we get

$$f(z_n)g(z_n) = f'(\alpha)\left[\left(\frac{2g'(\alpha)^2}{g(\alpha)} - 2g'(\alpha)\tau_i c_2 + 4g'(\alpha)c_2 - 2g(\alpha)\tau_i c_2^2 + 2g(\alpha)c_2^2\right)e_n^3 + O\left(e_n^4\right)\right], \tag{69}$$

$$f(x_n)g'(u_n) = f'(\alpha)\left[\begin{array}{c} g'(\alpha)e_n + \left(g''(\alpha)(1-\tau_i) + c_2 g'(\alpha)\right)e_n^2 + \\ \left\{c_3 g'(\alpha) + c_2 g''(\alpha)(1-\tau_i) + \dfrac{g'''(\alpha)}{2}(1-\tau_i)^2\right\}e_n^3 + O\left(e_n^4\right) \end{array}\right], \tag{70}$$

$$f'(u_n)g(x_n) = f'(\alpha)\left[\begin{array}{c} g(\alpha) + \left(g'(\alpha) - 2g(\alpha)\tau_i c_2 + 2g(\alpha)c_2\right)e_n + \\ \left\{3c_3\left(\tau_i^2 - 2\tau_i + 1\right)g(\alpha) + 2c_2(1-\tau_i)g'(\alpha) + \dfrac{g''(\alpha)}{2}\right\}e_n^2 \end{array}\right] + O\left(e_n^3\right). \tag{71}$$

Table 1: Numerical result for motion of a particle.

Method	n	x_n	$\lvert f(x_n)\rvert$	$\lvert x_n - x^*\rvert$
NM	6	−0.3170617745729570950	8.245568e − 22	3.926144e − 11
WF	5	−0.3170617745729570950	1.612296e − 71	1.751504e − 24
MP	5	−0.3170617745729570950	8.062909e − 78	1.477384e − 26
SH1	5	−0.3170617745729570950	4.280457e − 66	1.081215e − 22
SH2	5	−0.3170617745729570950	1.543287e − 22	1.310848e − 22
SN3	5	−0.3170617745729570950	1.543287e − 22	1.441913e − 22
NR1	4	−0.3170617745729571000	3.332577e − 19	1.042056e − 20
NR2	4	−0.3170617745729571000	3.332577e − 19	1.119981e − 13
AG1	5	−0.3170617745729570950	2.670763e − 56	1.882972e − 19
AG2	5	−0.3170617745729570950	5.041170e − 43	3.979192e − 15
AG3	4	−0.3170617745729571000	3.332577e − 19	1.782496e − 16
AG4	4	−0.3170617745729571000	3.332577e − 19	9.411615e − 12

Using equations (69), (70), and (71), we obtain

$$\frac{f(z_n)g(z_n)}{f(x_n)g'(u_n)+f'(u_n)g(x_n)} = \left[\frac{2c_2 g'(\alpha)}{g(\alpha)}(2-\tau_i) + \frac{2g'(\alpha)^2}{g(\alpha)^2} + 2c_2^2(1-\tau_i)\right]e_n^3 + O\left(e^4\right). \tag{72}$$

Thus, by using equation (72) and the fact $\sum_{i=1}^{p} w_i = 1$, the error term for Algorithm 3 is as follows:

$$\begin{aligned} x_{n+1} &= z_n - \frac{f(z_n)g(z_n)}{f(x_n)g'(u_n)+f'(u_n)g(x_n)}, \\ e_{n+1} &= \alpha + O\left(e_n^4\right). \end{aligned} \tag{73}$$

This completes the proof. □

4. Applications in Science and Engineering

In this section, we consider three scientific and engineering models related to mathematics, physics, and chemistry which include motion of a particle on an inclined plane, Van der Waal's equation of state, and continuous stirred tank rector equation. We make use of Maple software for computational work and MATLAB for graphical analysis. We apply the methods proposed in Algorithm 15 (AG1), Algorithm 16 (AG2), Algorithm 20 (AG3), and Algorithm 21 (AG4) along with Weerakon and Fernando method (WF, equation (3)), midpoint method (MP, equation (4)), Shah et al.'s [27] methods (Algorithm 11 (SH1), Algorithm 13 (SH2), and Algorithm 14 (SN3)), Noor et al.'s [9] method (Algorithm 2.13 (NR1)), Shah and Noor's [28] method (Algorithm 2.6 (NR2)), and basic Newton's method (NM, equation (1)) to show that the methods proposed in the present article work more efficiently (see the following Tables 1–3 and Figures 1–3).

We use the following stopping criteria for computer programmes:

$$\left(\lvert x_{n+1} - x_n\rvert + \lvert f(x_n)\rvert\right) < \varepsilon, \quad \text{where}\ \varepsilon = 10^{-15}. \tag{74}$$

(1) Motion of particle on an inclined plane [9].

We solve the nonlinear model formed due to the motion of a particle on an inclined plane whose angle of inclination θ changes at a constant rate $(d\theta/dt) = \omega < 0$.

$$x(t) = -\frac{g}{2\omega^2}\left(\frac{e^{\omega t} - e^{-\omega t}}{2} - \sin\omega t\right), \tag{75}$$

with

$$x_0 = -0.45. \tag{76}$$

(2) Van der Waals equation of state [29].

Now, we consider the governing nonlinear equation for the special case of well-known Van der Waals equation of state, i.e.,

$$\left(P + \frac{a_1 n^2}{V^2}\right)(V - na_2) = n\text{RT}. \tag{77}$$

Particularly, we solve the above equation in the following form:

$$x^3 - 5.22x^2 + 9.0825x - 5.2675 = 0, \tag{78}$$

with

$$x_0 = 1.65. \tag{79}$$

(3) Continuos stirred tank reactor (CSTR) [30].

We consider the nonlinear model of continuous stirred tank reactor, i.e.,

$$\begin{aligned} &x^4 + 11.50x^3 + 47.49x^2 \\ &\quad + 83.06325x + 51.23266875 = 0, \end{aligned} \tag{80}$$

Table 2: Numerical results for Van der Waals equation of state.

Method	n	x_n	$\lvert f(x_n)\rvert$	$\lvert x_n - x^*\rvert$
NM	9	1.7200000000000000000	$0.000000e+00$	$4.291037e-14$
WF	6	1.7200000000000000000	$1.436781e-34$	$3.172435e-12$
MP	6	1.7200000000000000000	$1.617053e-40$	$3.506693e-14$
SH1	6	1.7200000000000000000	$4.269024e-37$	$4.705826e-13$
SH2	6	1.7200000000000000000	$0.000000e+00$	$6.576319e-13$
SN3	6	1.7200000000000000000	$0.000000e+00$	$7.745611e-13$
NR1	6	1.7200000000000000000	$0.000000e+00$	$5.067884e-14$
NR2	6	1.7200000000000000000	$0.000000e+00$	$1.095066e-23$
AG1	5	1.7200000000000000000	$6.680166e-42$	$4.588240e-14$
AG2	6	1.7200000000000000000	$8.041535e-41$	$2.743100e-14$
AG3	4	1.7200000000000000000	$0.000000e+00$	$1.631356e-22$
AG4	5	1.7200000000000000000	$0.000000e+00$	$3.924480e-15$

Table 3: Numerical result for continuous stirred tank reactor (CSTR).

Method	n	x_n	$\lvert f(x_n)\rvert$	$\lvert x_n - x^*\rvert$
NM	8	−1.4500000000000000000	$0.000000e+00$	$4.758488e-16$
WF	6	−1.4500000000000000000	$5.336686e-58$	$2.952929e-20$
MP	5	−1.4500000000000000000	$1.329498e-30$	$4.323476e-11$
SH1	6	−1.4500000000000000000	$7.836919e-59$	$2.155427e-20$
SH2	5	−1.4500000000000000000	$0.000000e+00$	$5.727534e-24$
SN3	5	−1.4500000000000000000	$0.000000e+00$	$8.031736e-26$
NR1	5	−1.4500000000000000000	$0.000000e+00$	$1.017168e-27$
NR2	19	−2.8500000000798648000	$1.339461e-20$	$1.822556e-10$
AG1	5	−1.4500000000000000000	$7.313531e-60$	$9.512503e-21$
AG2	4	−1.4500000000000000000	$4.569485e-72$	$1.234708e-24$
AG3	4	−1.4500000000000000000	$0.000000e+00$	$1.944442e-17$
AG4	3	−1.4500000000000000000	$0.000000e+00$	$1.828241e-13$

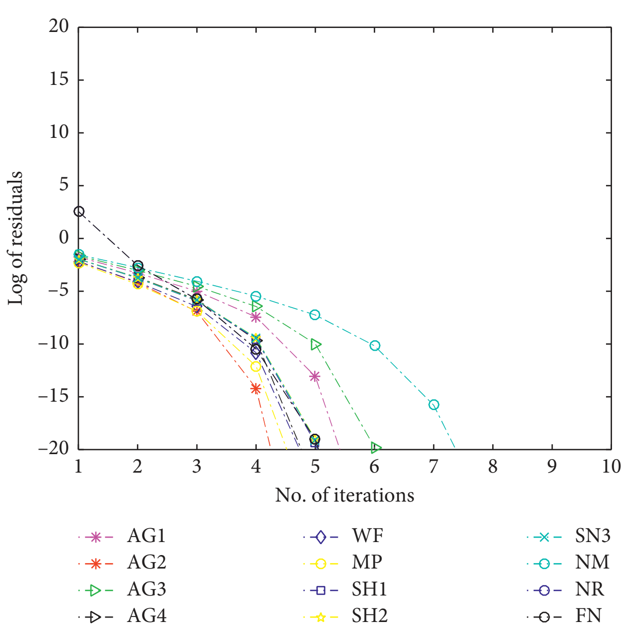

Figure 1: Log of residuals for equation (75).

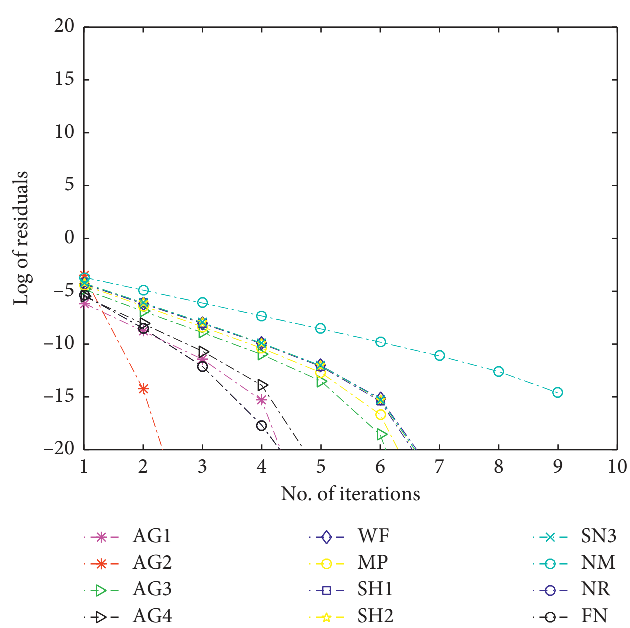

Figure 2: Log of residuals for equation (78).

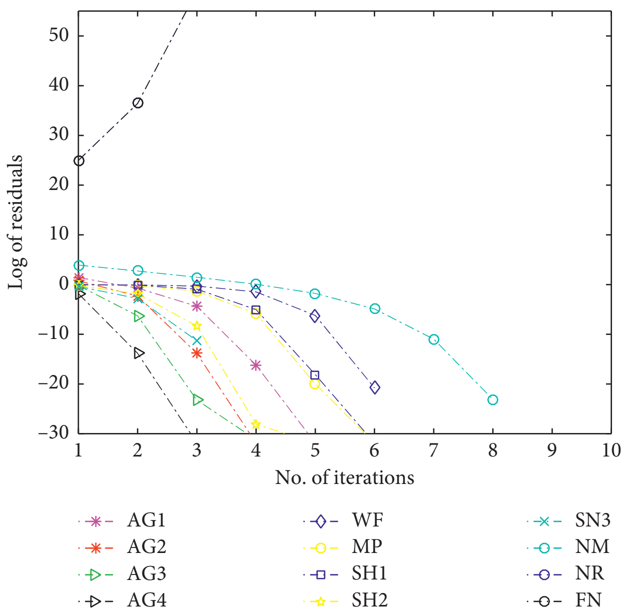

Figure 3: Log of residuals for equation (80).

with

$$x_0 = -1.75. \quad (81)$$

5. Conclusions

In this paper, introducing the weights and knots in the expansion of the auxiliary function $g(x)$, i.e., different from usual cases, we have developed a new family of iterative methods for nonlinear equations, i.e., Algorithms 15, 16, 20, and 21. We have compared our methods with some existing methods both numerically and graphically. The results obtained clearly reveal that the proposed methods are more rapidly convergent methods. These methods, besides giving some more efficient new techniques, are the generalized shape of some well-known methods.

Data Availability

All the data and supplementary material are available, if needed.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

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] G. Adomian, *Nonlinear Stochastic System and Applications to Physics*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1989.

[3] F. Ali, W. Aslam, and A. Rafiq, "Some new iterative techniques for the problems involving nonlinear equations," *International Journal of Computational Methods*, vol. 17, no. 7, Article ID 1950037, 2019.

[4] E. Babolian and J. Biazar, "Solution of nonlinear equations by modified Adomian decomposition method," *Applied Mathematics and Computation*, vol. 132, no. 1, pp. 167–172, 2002.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] A. Singh and J. P. Jaiswal, "An efficient family of optimal eighth-order iterative methods for solving nonlinear equations and its dynamics," *Journal of Mathematics*, vol. 2014, Article ID 569719, 14 pages, 2014.

[11] R. Thukral, "Introduction to higher-order iterative methods for finding multiple roots of nonlinear equations," *Journal of Mathematics*, vol. 2013, Article ID 404635, 3 pages, 2013.

[12] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall Englewood Cliffs, Upper Saddle River, NJ, USA, 1964.

[13] M. A. Ostrowski, *Solution of Equations in Euclidean and Banach Space*, Academic Press, New York, NY, USA, 3rd edition, 1973.

[14] J. H. He, "Ancient Chinese algorithm: the Ying Buzu Shu (method of surplus and deficiency) vs Newton iteration method," *Applied Mathematics and Mechanics-English Edition*, vol. 23, no. 12, pp. 1407–1412, 2002.

[15] J.-H. He, "Solution of nonlinear equations by an ancient Chinese algorithm," *Applied Mathematics and Computation*, vol. 151, no. 1, pp. 293–297, 2004.

[16] J.-H. He and Q. Yang, "Solitary wavenumber-frequency formulation using an ancient Chinese arithmetic," *International Journal of Modern Physics B*, vol. 24, no. 24, pp. 4747–4751, 2010.

[17] C. H. He, "An introduction to an ancient Chinese algorithm and its modification," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 26, no. 8, pp. 2486–2491, 2016.

[18] C. Chun, "Some fourth-order iterative methods for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 195, no. 2, pp. 454–459, 2008.

[19] 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.

[20] S. Weerakon 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.

[21] M. A. Ozban, "Some new variants of Newton's method," *Applied Mathematics Letters*, vol. 17, pp. 667–682, 2004.

[22] 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.

[23] M. Javidi, "Fourth-order and fifth-order iterative methods for nonlinear algebraic equations," *Mathematical and Computer Modelling*, vol. 50, no. 1-2, pp. 66–71, 2009.

[24] Y. Cherruault, "Convergence of Adomian's method," *Kybernetes*, vol. 8, pp. 31–38, 1988.

[25] Y. Cherruault, "Convergence of Adomian's method," *Mathematical and Computer Modelling*, vol. 14, pp. 83–86, 1990.

[26] F. Ali, W. Aslam, K. Ali, M. A. Anwar, and A. Nadeem, "New family of iterative methods for solving nonlinear models," *Discrete Dynamics in Nature and Society*, vol. 2018, Article ID 9619680, 12 pages, 2018.

[27] F. A. Shah, M. Datus, I. Faisal, and M. A. Shafiq, "Decomposition technique and family of efficient schemes for nonlinear equations," *Discrete Dynamics in Nature and Society*, vol. 2017, Article ID 3794357, 8 pages, 2017.

[28] F. A. Shah and M. A. Noor, "Some numerical methods for solving nonlinear equations by using decomposition technique," *Applied Mathematics and Computation*, vol. 251, pp. 376–386, 2015.

[29] K. J. Beers, *Numerical Methods for Chemical Engineering, Applications in MATLAB*, Cambridge University Press, Cambridge, UK, 2006.

[30] A. Constantinides and N. Mostoufi, *Numerical Methods for Chemical Engineers with MATLAB Applications*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.