Hindawi
Journal of Mathematics
Volume 2020, Article ID 5070363, 11 pages
https://doi.org/10.1155/2020/5070363

Research Article

Higher-Order Root-Finding Algorithms and Their Basins of Attraction

Amir Naseem,[1] **M. A. Rehman**,[1] **and Thabet Abdeljawad**[2,3,4]

[1] *Department of Mathematics, University of Management and Technology, Lahore, Pakistan*
[2] *Department of Mathematics and General Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia*
[3] *Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan*
[4] *Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan*

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

Received 6 July 2020; Revised 26 September 2020; Accepted 26 October 2020; Published 30 November 2020

Academic Editor: Kaleem R. Kazmi

In this paper, we proposed and analyzed three new root-finding algorithms for solving nonlinear equations in one variable. We derive these algorithms with the help of variational iteration technique. We discuss the convergence criteria of these newly developed algorithms. The dominance of the proposed algorithms is illustrated by solving several test examples and comparing them with other well-known existing iterative methods in the literature. In the end, we present the basins of attraction using some complex polynomials of different degrees to observe the fractal behavior and dynamical aspects of the proposed algorithms.

1. Introduction

Most of the problems in Mathematics, Physics, and Engineering sciences are linked with the solution of nonlinear equations of the following form:

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

where $f\colon X \subset \mathbb{R} \longrightarrow \mathbb{R}$ is a scalar function defined on an open connected set X.

Mostly, the roots of such equations cannot be found directly, and therefore, we need to adopt an iterative method for approximating the roots of such type of equations. In an iterative method, we start the process by choosing an initial guess x_0 which is refined sequentially by means of iterations until the approximated solution is achieved. Some of the basic and classical methods are given in [1–9] and the references therein.

The most famous and well-known method for finding roots of nonlinear equations is of the following form:

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

which is quadratically convergent Newton's method [10] for the solution of nonlinear equations.

To improve its convergence, a large number of iteration schemes have been proposed by means of various types of techniques such as Adomian decomposition method, Taylor's series, perturbation method, quadrature formula, interpolation, and finite difference techniques for removing higher-order derivatives and variational iteration techniques, see [10–22] and the references therein.

In this paper, we proposed three new algorithms using variational iteration technique by considering two auxiliary functions $\phi(x)$ and $\psi(x)$. The first one, $\phi(x)$, behaves like a predictor function having convergence order q, where $q \geq 1$ and helps to attain iterative methods of convergence order $q + r$, where $r \geq 1$ is the order of convergence of the second auxiliary function $\psi(x)$. Using variational iteration technique, we develop some new higher order root-finding algorithms with best performance and efficiency. The variational iteration technique was introduced by Inokuti et al. [15] in 1978. Using this technique, Noor and Shah [20, 21] proposed some iterative methods for the solution of nonlinear equations. The purpose of this technique was to solve a variety of diverse problems [12–14].

Now, we apply the described technique to obtain higher-order root-finding algorithms. New algorithms are very fast using less number of iterations to reach the required root, free from 3rd and higher derivatives with ninth order of convergence which raises the efficiency index of these algorithms.

The rest of the paper is divided as follows. The new iteration schemes are described in Section 2. In Section 3, the convergence criteria of the proposed algorithms have been discussed. In Section 4, various test examples have been solved to show their performance as compared to the other similar existing methods in the literature. The basins of attraction for some complex polynomials have been presented in Section 5 which shows the dynamical and fractal behavior of the proposed algorithms. Finally, the conclusion of the paper is given in Section 6.

2. Construction of Root-Finding Algorithms Using Variational Iteration Technique

In this section, we construct some new root-finding algorithms with the help of variational iteration technique. These algorithms are multistep iterative methods which involve predictor and corrector steps. The proposed algorithms possess higher order of convergence than one-step methods. By applying variational iteration technique, we derive some new root-finding algorithms of order $q+r$ where $q, r \geq 1$ are the orders of convergence of the auxiliary iteration functions $\phi(x)$ and $\psi(x)$. Now, consider the nonlinear equation of the following form:

$$f(x) = 0. \tag{3}$$

Suppose that α is the simple root and γ is the initial guess sufficiently close to α. For better understanding and to deliver the basic idea, we suppose the approximate solution x_n of (3) such that

$$f(x_n) \neq 0. \tag{4}$$

We consider $\phi(x)$ and $\psi(x)$ as two iteration functions of order q and r, respectively. Then

$$x_{n+1} = \phi(x_n) + \mu\left[f(\psi(x_n))g(\psi(x_n))\right]^t, \tag{5}$$

where $t = q/r$, is a recurrence relation which generates iterative methods of order $q+r$ and $g(x)$ is any arbitrary function which later on is converted to $g(\psi(x_n))$ and μ is a parameter, called the "Lagrange's multiplier" and can be determined from (5) by using the optimality criteria as follows:

$$\mu = -\frac{\phi'(x_n)\left[f(\psi(x_n))g(\psi(x_n))\right]^{1-t}}{t\psi'(x_n)\left[f'(\psi(x_n))g(\psi(x_n)) + f(\psi(x_n))g'(\psi(x_n))\right]}. \tag{6}$$

From (5) and (6), we get

$$x_{n+1} = \phi(x_n) - \frac{\phi'(x_n)}{t\psi'(x_n)} \cdot \frac{f(\psi(x_n))g(\psi(x_n))}{\left[f'(\psi(x_n))g(\psi(x_n)) + f(\psi(x_n))g'(\psi(x_n))\right]}. \tag{7}$$

Now, we are going to apply (7) for constructing a general iterative scheme for iterative methods. For this, suppose that

$$\psi(x_n) = y_n = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)f''(x_n)}{2f'^3(x_n)}, \tag{8}$$

which is well-known Househölder's method with cubic convergence [7]. With the help of (7) and (8), we can write

$$x_{n+1} = \phi(x_n) - \frac{\phi'(x_n)f(y_n)g(y_n)}{ty_n'\left[f'(y_n)g(y_n) + f(y_n)g'(y_n)\right]}. \tag{9}$$

Let

$$\phi(x_n) = z_n = y_n - \frac{f(y_n)}{f'(y_n)}, \tag{10}$$

where $y_n = x_n - f(x_n)/f'(x_n) - f^2(x_n)f''(x_n)/2f'^3(x_n)$, which is the two-step iterative method having convergence of order six. Differentiate equation (10) w.r.t "x," we have

$$\phi'(x_n) = z_n' = \frac{f(y_n)f''(y_n)}{f'^2(y_n)}y_n', \tag{11}$$

and from Taylor's series, we can write

$$f(z_n) \approx f(y_n) + (z_n - y_n)f'(y_n) + \frac{(z_n - y_n)^2}{2}f''(y_n) = \frac{f^2(y_n)f''(y_n)}{2f'^2(y_n)}. \tag{12}$$

The last expression has been obtained by putting the value of $z_n - y_n$ from (10).

From (11) and (12), we have

$$\phi'(x_n) = \frac{2f(z_n)}{f(y_n)}y_n', \tag{13}$$

with the help of (9), (10), and (13), we get

$$x_{n+1} = z_n - \frac{2f(z_n)g(y_n)}{t\left[g'(y_n)f(y_n) + g(y_n)f'(y_n)\right]}, \tag{14}$$

where $t = 6/3 = 2$, which is according to the above described technique. Then, equation (14) becomes

$$x_{n+1} = z_n - \frac{f(z_n)g(y_n)}{\left[g'(y_n)f(y_n) + g(y_n)f'(y_n)\right]}. \tag{15}$$

Relation (15) is the main and general iterative scheme, which we use to deduce some new root-finding algorithms by considering some particular cases of the auxiliary function g, keeping in mind that it should not be zero or small for all iterations. If g obtains zero value, then (15) reduces to $x_{n+1} = z_n$ which is similar to (10) and we are unable to derive new algorithms.

2.1. Case 1. Let $g(x_n) = \exp(-\beta x_n)$, then $g'(x_n) = -\beta g(x_n)$. Using these values in (15), we obtain the following algorithm.

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

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n) f''(x_n)}{2f'^3(x_n)}, \quad n = 0,1,2,\ldots, \\ z_n &= y_n - \frac{f(y_n)}{f'(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{[f'(y_n) - \beta f(y_n)]}. \end{aligned} \tag{16}$$

2.2. Case 2. Let $g(x_n) = \exp(-\beta f(x_n))$, then $g'(x_n) = -\beta f'(x_n) g(x_n)$. Using these values in (15), we obtain the following algorithm.

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

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n) f''(x_n)}{2f'^3(x_n)}, \quad n = 0,1,2,\ldots, \\ z_n &= y_n - \frac{f(y_n)}{f'(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{[f'(y_n) - \beta f(y_n) f'(y_n)]}. \end{aligned} \tag{17}$$

2.3. Case 3. Let $g(x_n) = \exp(-\beta x_n^2)$, then $g'(x_n) = -2\beta x_n g(x_n)$. Using these values in (15), we obtain the following algorithm.

Algorithm 3. For a given x_0, compute the approximate solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n) f''(x_n)}{2f'^3(x_n)}, \quad n = 0,1,2,\ldots, \\ z_n &= y_n - \frac{f(y_n)}{f'(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{[f'(y_n) - 2\beta y_n f(y_n)]}. \end{aligned} \tag{18}$$

To obtain best results in all above algorithms, always choose that values of β which makes the denominator nonzero and largest in magnitude.

3. Convergence Analysis

In this section, we discuss the convergence criteria of the general iteration scheme described in relation (15).

Theorem 1. *Assume that $\alpha \in I$ be the simple root of the differentiable function $f\colon I \subset \mathbb{R} \longrightarrow \mathbb{R}$ on an open interval I. If the initial guess x_0 is sufficiently close to α, then the convergence order of the main and general iteration scheme described in relation (15) is at least nine.*

Proof. To prove the convergence of the main and general iteration scheme described in relation (15), we assume that α is the simple root of the equation $f(x) = 0$ and e_n be the error at nth iteration, then $e_n = x_n - \alpha$ and by using Taylor series about $x = \alpha$, we have

$$\begin{aligned} f(x_n) = {} & f'(\alpha) e_n + \frac{1}{2!} f'(\alpha) e_n^2 + \frac{1}{3!} f'''(\alpha) e_n^3 \\ & + \frac{1}{4!} f^{(iv)}(\alpha) e_n^4 + \frac{1}{5!} f^{(v)}(\alpha) e_n^5 + \frac{1}{6!} f^{(vi)}(\alpha) e_n^6 \\ & + \cdots + O(e_n^{10}), \end{aligned} \tag{19}$$

$$\begin{aligned} f(x_n) = {} & f'(\alpha)\left[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 \right. \\ & \left. + c_6 e_n^6 + \cdots + O(e_n^{10})\right], \end{aligned} \tag{20}$$

where

$$c_n = \frac{1}{n!} \frac{f^{(n)}(\alpha)}{f'(\alpha)}, \tag{21}$$

$$f'(x_n) = f'(\alpha)\left[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + 6c_6 e_n^5 + 7c_7 e_n^6 + \cdots + O(e_n^{10})\right], \tag{22}$$

$$f''(x_n) = f'(\alpha)\left[2c_2 + 6c_3 e_n + 12c_4 e_n^2 + 20c_5 e_n^3 + 30c_6 e_n^4 + 42c_7 e_n^5 + 56c_8 e_n^6 + \cdots + O(e_n^{10})\right]. \tag{23}$$

With the help of (20)–(23), we get

$$\begin{aligned} y_n = \alpha &+ \left(-c_3 + 2c_2^2\right)e_n^3 + \left(12c_2c_3 - 9c_2^3 - 3c_4\right)e_n^4 + \left(15c_3^2 - 63c_3c_2^2 + 30c_2^4 + 24c_2c_4 - 6c_5\right)e_n^5 \\ &+ \left(10c_6 + 40c_2c_5 + 55c_4c_3 - 112c_4c_2^2 - 136c_2c_3^2 + 251c_3c_2^3 - 88c_2^5\right)e_n^6 + \cdots + O\left(e_n^{10}\right), \end{aligned} \tag{24}$$

$$\begin{aligned} f(y_n) = f'(\alpha)\Big[&\left(-c_3 + 2c_2^2\right)e_n^3 + \left(12c_2c_3 - 9c_2^3 - 3c_4\right)e_n^4 + \left(15c_3^2 - 63c_3c_2^2 + 30c_2^4 + 24c_2c_4 - 6c_5\right)e_n^5 \\ &+ \left(-10c_6 + 40c_2c_5 + 55c_4c_3 - 112c_4c_2^2 - 135c_2c_3^2 + 247c_3c_2^3 - 84c_2^5\right)e_n^6 + \cdots + O\left(e_n^{10}\right)\Big], \end{aligned} \tag{25}$$

$$f'(y_n) = f'(\alpha)\begin{bmatrix} 1 + \left(-2c_2c_3 + 4c_2^3\right)e_n^3 + \left(24c_3c_2^2 - 18c_2^4 - 6c_2c_4\right)e_n^4 + \left(30c_2c_3^2 - 126c_3c_2^3 + 60c_2^5 + 48c_4c_2^2 - 12c_2c_5\right)e_n^5 \\ + \left(-20c_2c_6 + 80c_2^2c_5 + 110c_2c_4c_3 - 224c_4c_2^3 - 284c_2^2c_3^2 + 514c_3c_2^4 - 176c_2^6 + 3c_3^3\right)e_n^6 + \cdots + O\left(e_n^{10}\right) \end{bmatrix}, \tag{26}$$

$$\begin{aligned} g(y_n) = g(\alpha) &+ g'(\alpha)\left(-c_3 + 2c_2^2\right)e_n^3 + g'(\alpha)\left(12c_2c_3 - 9c_2^3 - 3c_4\right)e_n^4 + g'(\alpha)\left(15c_3^2 - 63c_3c_2^2 + 30c_2^4 + 24c_2c_4 - 6c_5\right)e_n^5 \\ &+ \left[g'(\alpha)\left(-10c_6 + 40c_2c_5 + 55c_4c_3 - 112c_4c_2^2 - 136c_2c_3^2 + 251c_3c_2^3 - 88c_2^5\right) + 2g''(\alpha)\left(-\frac{c_3}{2} + c_2^2\right)^2\right]e_n^6 + \cdots + O\left(e_n^{10}\right), \end{aligned} \tag{27}$$

$$\begin{aligned} g'(y_n) = g'(\alpha) &+ g''(\alpha)\left(-c_3 + 2c_2^2\right)e_n^3 + g''(\alpha)\left(12c_2c_3 - 9c_2^3 - 3c_4\right)e_n^4 + g''(\alpha)\left(15c_3^2 - 63c_3c_2^2 + 30c_2^4 + 24c_2c_4 - 6c_5\right)e_n^5 \\ &+ \left[g''(\alpha)\left(-10c_6 + 40c_2c_5 + 55c_4c_3 - 112c_4c_2^2 - 136c_2c_3^2 + 251c_3c_2^3 - 88c_2^5\right) + 2g''(\alpha)\left(-\frac{c_3}{2} + c_2^2\right)^2\right]e_n^6 + \cdots + O\left(e_n^{10}\right). \end{aligned} \tag{28}$$

With the help of (24)–(26), we have

$$z_n = \alpha + \left(4c_2^5 - 4c_3c_2^3 + c_2c_3^2\right)e_n^6 + \cdots + O\left(e_n^{10}\right), \tag{29}$$

$$f(z_n) = f'(\alpha)\left[\left(4c_2^5 - 4c_3c_2^3 + c_2c_3^2\right)e_n^6 + \cdots + O\left(e_n^{10}\right)\right]. \tag{30}$$

Using equations (20)–(38) in general iteration scheme (15), we get the same result as follows:

$$x_{n+1} = \alpha + \frac{16\left(c_2^2 - c_3/2\right)^3\left[g(\alpha)c_2 + 1/2g'(\alpha)\right]c_2}{g(\alpha)}e_n^9 + O\left(e_n^{10}\right), \tag{31}$$

which implies that

$$e_{n+1} = \frac{16\left(c_2^2 - c_3/2\right)^3\left[g(\alpha)c_2 + 1/2g'(\alpha)\right]c_2}{g(\alpha)}e_n^9 + O\left(e_n^{10}\right), \tag{32}$$

which shows that the main and general iteration scheme (15) is of ninth order of convergence and all algorithms deduced from it have also the same order of convergence. □

4. Numerical Results

In this section, we included some nonlinear functions to demonstrate the performance of newly proposed algorithms for $\beta = 1$. We compare these algorithms with the following well-known iterative methods:

4.1. Newton's Method (NM). For a given x_0, compute the approximate solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{33}$$

which is well-known Newton's method [10] for finding zeros of nonlinear functions, having quadratic order of convergence.

4.2. Halley's Method (HM). For a given x_0, compute the approximate solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2f^2(x_n) - f(x_n)f''(x_n)}. \tag{34}$$

This is so called Halley's method [2] for root-finding of nonlinear functions, which converges cubically.

4.3. Traub's Method (TM). For a given x_0, compute the approximate solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - \frac{f(y_n)}{f'(y_n)}, \end{aligned} \tag{35}$$

which is known as Traub's method [10] for finding roots of nonlinear functions and possesses fourth order of convergence.

4.4. Modified Halley's Method (MHM). For a given x_0, compute the approximate solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - \frac{2f(x_n)f(y_n)f'(y_n)}{2f(x_n)f'^2(y_n) - f'^2(x_n)f(y_n) + f'(x_n)f'(y_n)f(y_n)}, \end{aligned} \tag{36}$$

which is modified Halley's method [23] for finding the solution of nonlinear functions, having fifth order of convergence.

In order to make numerical comparison of the above described methods with the newly developed algorithms, the following test examples have been solved:

$$\begin{aligned} &f_1(x) = x^3 + 4x^2 - 10, \quad x_0 = -0.3, \\ &f_2(x) = (x-1)^3 - 2, \quad x_0 = 0.1, \\ &f_3(x) = (x+2)e^x - 1, \quad x_0 = 3.1, \\ &f_4(x) = xe^{x^2} - \sin^2 x + 3\cos x + 5, \quad x_0 = -2, \\ &f_5(x) = \sin^2 x - x^2 + 1, \quad x_0 = 5, \end{aligned} \tag{37}$$

as shown in Table 1.

Table 1 shows the numerical comparison of our developed algorithms (for $\beta = 1$) with Newton's method, Halley's method, Traub's method, and modified Halley's method. The columns represent the number of iterations N, the magnitude $|f(x)|$ of $f(x)$ at the final estimate x_{n+1}, the approximate root x_{n+1}, and the computational order of convergence (COC) that can be expressed mathematically using the following formula:

$$\text{COC} \approx \frac{\ln\left|(x_{n+1} - x_n)\right| / \left|(x_n - x_{n-1})\right|}{\ln\left|(x_n - x_{n-1})\right| / \left|(x_{n-1} - x_{n-2})\right|}, \tag{38}$$

which was suggested by Cordero and Torregrosa in (2007) [24].

When we look at the numerical results of Table 1, we come to know that our proposed algorithms are showing best performance as compared to the other ones. For example, in first test example f_1 of Table 1, Algorithm 3 is the best as it took less number of iterations among the all other compared methods with great precision. In second, third, and fifth test examples, Algorithm 1 performed better than the other ones. In the fourth one, Algorithm 3 looks superior to the other methods. In short, we can say that the proposed algorithms are best in terms of accuracy, speed, number of iterations, and computational order of convergence as compared to the other well-known iterative methods. All numerical examples have been solved using the computer program Maple 13 by taking the accuracy $\varepsilon = 10^{-15}$ for the following stopping criteria:

$$\left|x_{n+1} - x_n\right| < \varepsilon. \tag{39}$$

Table 2 shows the comparison of the number of iterations required for different iterative methods with our developed algorithms (for $\beta = 1$) to approximate the root of the given nonlinear function with the accuracy $\varepsilon = 10^{-100}$. The columns represent the number of iterations for different functions along with initial guess x_0.

The numerical results as shown in Table 2 again certified the fast and best performance of the proposed algorithms in terms of number of iterations for the above defined stopping criteria with the given accuracy. In all test examples, the proposed algorithms consumed less number of iterations as compared to the other iterative methods. All calculations have been carried out using the computer program Maple 13.

Table 3 shows the effect of parameter on the proposed algorithms by taking three different values of β. We applied the proposed algorithms on different test examples and the obtained results showing that the numeric behavior of the proposed algorithms changes with the change of the parameter β. A major change can be seen in first test example f_1 of Table 3. Here, the Algorithm 1 took six iterations for $\beta = 1$, thirty-two iterations for $\beta = 0.10$, and thirteen iterations for $\beta = 0.25$. Similar changes can be observed from the other examples of Table 3. By looking at the overall results of Table 3, we can conclude that $\beta = 1$ is the best choice of parameter for the proposed algorithms, which we have already used in Tables 1 and 2.

5. Basins of Attraction

An attractor's basin of attraction is the region of the phase space, over which iterations are defined, such that any point (any initial condition) in that region will eventually be iterated into the attractor. Equations or systems that are nonlinear can give rise to a richer variety of behavior than can linear systems. The basins of attraction describe the dynamical aspects and characteristics of an iterative method that is being under consideration over a large number of examples and sets of parameter values, see [25, 26] and references cited therein. The basin of attraction for complex Newton's method was first considered and attributed by Cayley [27]. The aim of this section is to represent the basins for the proposed algorithms by using graphical tools. In order to represent the basins of attraction via different computer programmes, we have to choose an initial rectangle R which contains the roots of the considered polynomial. Then, for every point z_0 in the region, we run an iterative method and then color the point corresponding to z_0 which is depended upon the approximate convergence of the truncated orbit to a root, or lack thereof. The resolution of the image depends upon our discretization of the rectangle R. For example, if we discretize R into a 2000 by 2000 grid, then the result would be a high-resolution image.

The basins of attraction using different algorithms have various applications in different fields such as Mathematics, Science, Education, Art, and Design. From the Fundamental Theorem of Algebra, any complex polynomial with complex coefficients $\{b_n, b_{n-1}, \ldots, b_1, b_o\}$:

Table 1: Comparison of NM, HM, TM, MHM, and Algorithms 1–3.

Method	N	$\lvert f(x_{n+1})\rvert$	x_{n+1}	COC
$f_1(x)$, $x_0 = -0.3$				
NM	54	$8.127500e-30$	1.365230013414096845760806828982158 25	2
HM	58	$6.815871e-25$	1.36523001341409684576080687025645640	3
TM	27	$8.127500e-30$	1.36523001341409684576080682898215825	4
MHM	22	$2.439119e-36$	1.36523001341409684576080682898166608	5
Algorithm 1	6	$7.091332e-18$	1.36523001341409684619023581318459998	9
Algorithm 2	5	$6.372651e-34$	1.36523001341409684576080682898166612	9
Algorithm 3	5	$4.284994e-42$	1.36523001341409684576080682898166608	9
$f_2(x)$, $x_0 = 0.1$				
NM	13	$3.733284e-27$	2.25992104989487316476721060806216888	2
HM	9	$6.698145e-33$	2.25992104989487316476721060727822976	3
TM	7	$2.322902e-54$	2.25992104989487316476721060727822835	4
MHM	7	$1.851813e-15$	2.25992104989487355362353142711262157	5
Algorithm 1	2	$2.424093e-19$	2.25992104989487316481811338069301035	9
Algorithm 2	3	$5.354563e-75$	2.25992104989487316476721060727822835	9
Algorithm 3	3	$1.555712e-62$	2.25992104989487316476721060727822835	9
$f_3(x)$, $x_0 = 3.1$				
NM	9	$4.410873e-15$	−0.44285440100238589718912232041863157	2
HM	5	$3.594315e-16$	−0.44285440100238836426950684368990422	3
TM	5	$8.240223e-30$	−0.44285440100238858314132799999431903	4
MHM	5	$1.172300e-56$	−0.44285440100238858314132799999933682	5
Algorithm 1	3	$5.050188e-47$	−0.44285440100238858314132799999933682	9
Algorithm 2	4	$2.382120e-52$	−0.44285440100238858314132799999933682	9
Algorithm 3	4	$1.515125e-87$	−0.44285440100238858314132799999933682	9
$f_4(x)$, $x_0 = -2.0$				
NM	8	$5.538944e-20$	−1.20764782713091892701214430654168756	2
HM	4	$6.330594e-16$	−1.20764782713091895818322677251719839	3
TM	4	$5.538944e-20$	−1.20764782713091892701214430654168756	4
MHM	41	$4.223498e-29$	−1.20764782713091892700941675835816388	5
Algorithm 1	3	$1.388738e-39$	−1.20764782713091892700941675835608410	9
Algorithm 2	3	$2.255420e-25$	−1.20764782713091892700941676946247375	9
Algorithm 3	2	$1.733403e-25$	−1.20764782713091892700941676689190004	9
$f_5(x)$, $x_0 = 5.0$				
NM	7	$1.502508e-22$	1.40449164821534122603514734244588648	2
HM	4	$2.649470e-38$	1.40449164821534122603508681778686808	3
TM	4	$7.125700e-45$	1.40449164821534122603508681778686808	4
MHM	3	$4.868406e-35$	1.40449164821534122603508681778686810	5
Algorithm 1	2	$2.062348e-20$	1.40449164821534122602677918211795854	9
Algorithm 2	3	$6.624520e-76$	1.40449164821534122603508681778686808	9
Algorithm 3	3	$1.970835e-56$	1.40449164821534122603508681778686808	9

Table 2: Comparison of number of iterations required for different iterative methods for $\varepsilon = 10^{-100}$.

Method	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$
	$x_0 = -0.3$	$x_0 = 0.1$	$x_0 = 3.1$	$x_0 = -2.0$	$x_0 = 5.0$
NM	56	15	12	11	10
HM	60	11	7	6	5
TM	28	8	6	6	5
MHM	23	9	6	42	4
Algorithm 1	7	3	4	4	3
Algorithm 2	6	4	5	3	4
Algorithm 3	6	4	5	3	4

$$p(z) = b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_o, \tag{40}$$

or by its zeros (roots) $\{z_1, z_2, \ldots, z_{n-1}, z_n\}$:

$$p(z) = (z - z_1)(z - z_2) \ldots (z - z_n), \tag{41}$$

of degree n has exactly n roots which may be distinct or repeated. The polynomial's degree illustrates the number of basins of attraction and placing roots on the complex plane. The localization of basins can be controlled manually. Usually, the colors of basins of attraction depend upon the considered iterative method and the total number of iterations required to attain the approximate solution of some polynomials with a given accuracy. The detailed study of basins of attraction, its theoretical background, and interesting applications are discussed in [28–36].

5.1. Applications. In a numerical algorithm that is purely based on an iterative process, we always require stopping criteria for the whole process, i.e., a test which provides us information that the process has converged or it is very close

Table 3: The effect of parameter β on the proposed algorithms.

Method	Number of iterations		
	For $\beta = 1$	For $\beta = 0.10$	For $\beta = 0.25$
$f_1(x)$, $x_0 = -0.3$			
Algorithm 1	6	32	13
Algorithm 2	5	5	5
Algorithm 3	5	4	5
$f_2(x)$, $x_0 = 0.1$			
Algorithm 1	2	3	3
Algorithm 2	3	2	3
Algorithm 3	3	2	2
$f_3(x)$, $x_0 = 3.1$			
Algorithm 1	3	4	3
Algorithm 2	4	4	4
Algorithm 3	4	3	3
$f_4(x)$, $x_0 = -2.0$			
Algorithm 1	3	3	3
Algorithm 2	3	3	3
Algorithm 3	2	3	3
$f_5(x)$, $x_0 = 5.0$			
Algorithm 1	2	3	3
Algorithm 2	3	3	3
Algorithm 3	3	3	2

to the solution. Such type of test is called a convergence test having the following standard form:

$$\left|z_{n+1} - z_n\right| < \varepsilon, \tag{42}$$

where z_{n+1} and z_n are two consecutive points in the iteration process and $\varepsilon > 0$ is a given accuracy. In this paper, we also use the stopping criteria (42). Using newly developed algorithms, we presented basins of attractions of various polynomials that were much attractive, colorful, and aesthetic ones. The different colors of an image was based on the number of iterations required to approximate the root with the defined accuracy ε. A large number of such images can be generated by giving different values to the parameter k, where k represents the upper bound of the number of iterations.

Here, we present the basins of attractions using the following complex polynomials of different degrees:

$$\begin{aligned} p_1(z) &= z^3 - 1, p_2(z) = \left(z^3 - 1\right)^2, \\ p_3(z) &= z^4 - 1, p_4(z) = \left(z^4 - 1\right)^2, \\ p_5(z) &= z^5 - 1, p_6(z) = \left(z^5 - 1\right)^2. \end{aligned} \tag{43}$$

All the figures have been generated using the computer program Mathematica 10.0 by taking $\varepsilon = 0.01$, $A = [-2, 2] \times [-2, 2]$, and $k = 15$, where ε shows the accuracy of the given root, A represents the area in which we draw the basins of attraction, and k represents the upper bound of the number of iterations.

Example 1. Basins of attraction using the complex polynomial $p_1(z)$ for proposed algorithms.

Example 2. Basins of attraction using the complex polynomial $p_2(z)$ for proposed algorithms.

Example 3. Basins of attraction using the complex polynomial $p_3(z)$ for proposed algorithms.

Example 4. Basins of attraction using the complex polynomial $p_4(z)$ for proposed algorithms.

Example 5. Basins of attraction using the complex polynomial $p_5(z)$ for proposed algorithms.

Example 6. Basins of attraction using the complex polynomial $p_6(z)$ for proposed algorithms.

In Examples 1–6, basins of attraction for the complex polynomials of different degrees through our proposed algorithms (for $\beta = 1$) have been shown.

In the first experiment, we have run all the proposed algorithms to obtain the simple zeros of the cubic polynomial $p_1(z)$. The results of the basin of attractions are given in Figure 1. For each distinct root of the considered polynomial, there exists a unique color on the corresponding basins of attraction, so there exists three unique colors, namely, brown, yellow, and red that can be easily seen from Figure 1. In the next experiment, we consider the polynomial $p_2(z)$ which has three distinct roots with multiplicity 2. The basins of attraction are presented in Figure 2. Three unique colors corresponding to the distinct roots can be seen in Figure 2. The repeated roots appeared with the same colors on the basins of attraction. In Examples 3 and 4, we ran the proposed algorithms for the complex polynomials $p_3(z)$ and $p_4(z)$. The results are given in Figures 3 and 4. Both the polynomials have four distinct roots, and the corresponding four colors can be easily seen in Figures 3 and 4, respectively. The later polynomial $p_4(z)$ has all roots with multiplicity 2. The basins of attraction for the complex polynomials $p_5(z)$ and $p_6(z)$ through our proposed algorithms have been

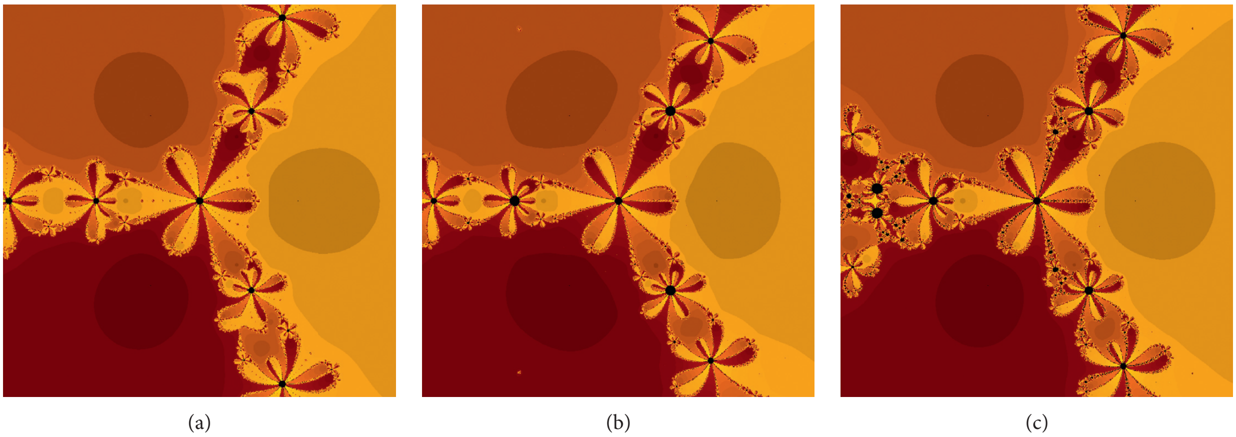
(a) (b) (c)

Figure 1: Basins of attraction associated with the polynomial $p_1(z)$ for $\beta = 1$. (a) Stands for Algorithm 1, (b) for Algorithm 2, and (c) for Algorithm 3.

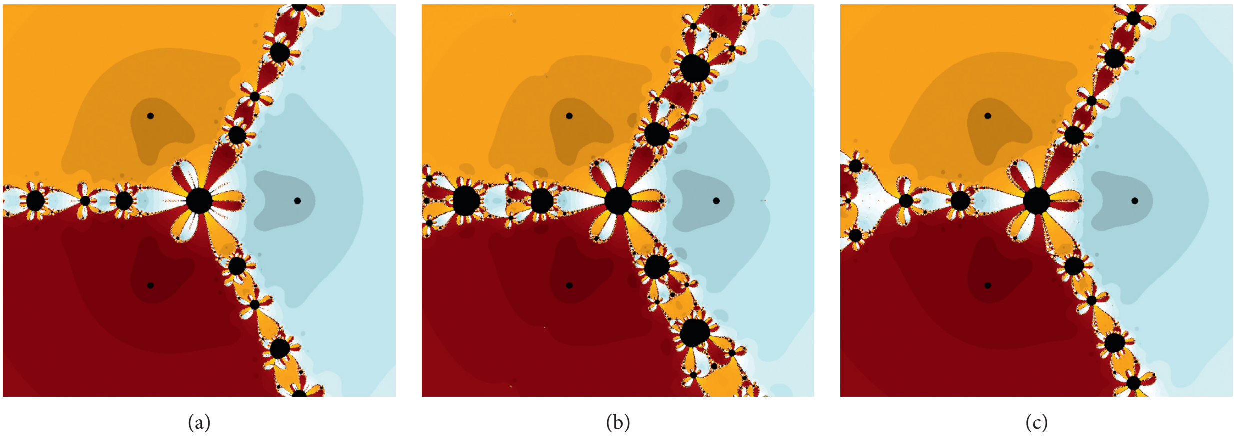
(a) (b) (c)

Figure 2: Basins of attraction associated with the polynomial $p_2(z)$ for $\beta = 1$. (a) Stands for Algorithm 1, (b) for Algorithm 2, and (c) for Algorithm 3.

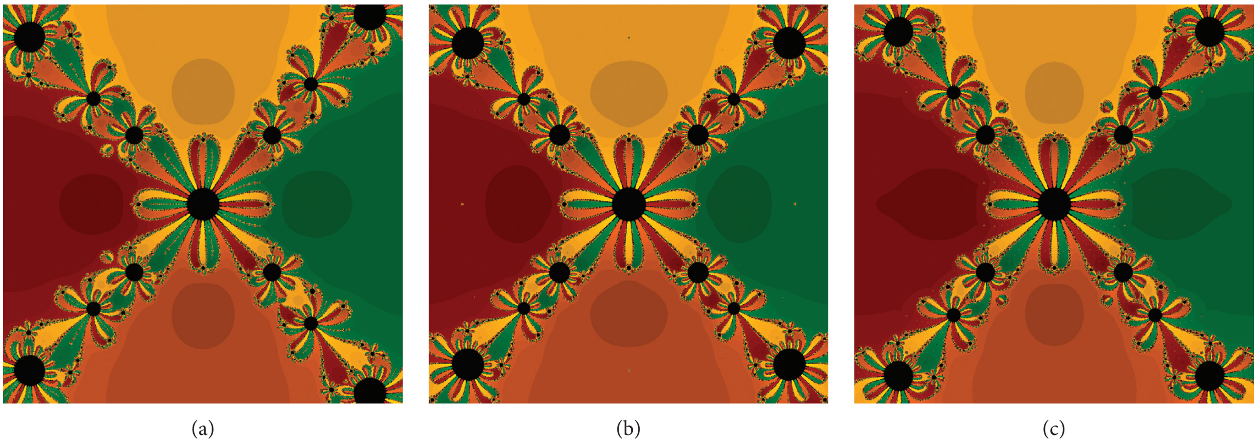
(a) (b) (c)

Figure 3: Basins of attraction associated with the polynomial $p_3(z)$ for $\beta = 1$. (a) Stands for Algorithm 1, (b) for Algorithm 2, and (c) for Algorithm 3.

presented in Figures 5 and 6, respectively. These polynomials have five distinct roots, but the zeros of the later polynomial $p_6(z)$ are not simple and have multiplicity 2. The five unique colors corresponding to these roots appeared on the basins of attraction that can be seen from Figures 5 and 6.

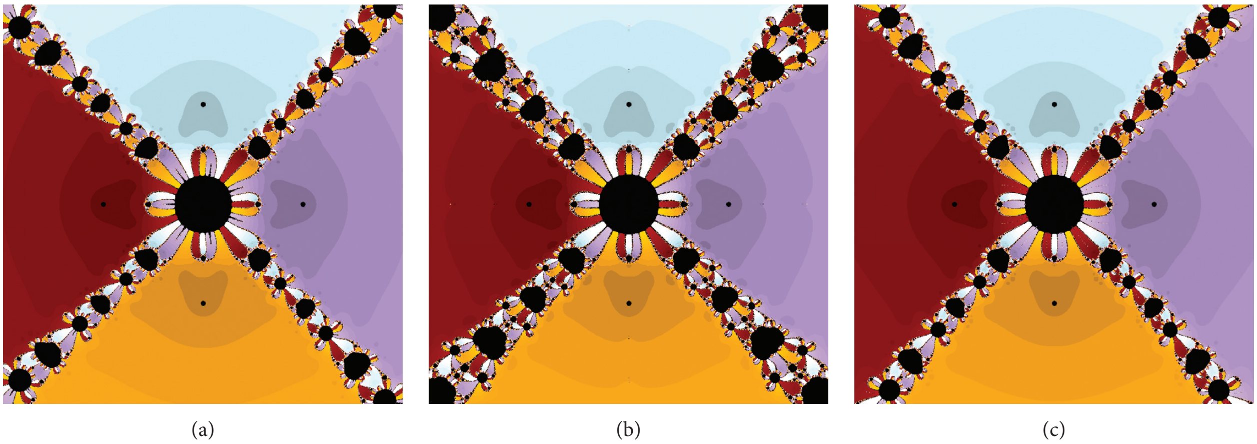

Figure 4: Basins of attraction associated with the polynomial $p_4(z)$ for $\beta = 1$. (a) Stands for Algorithm 1, (b) for Algorithm 2, and (c) for Algorithm 3.

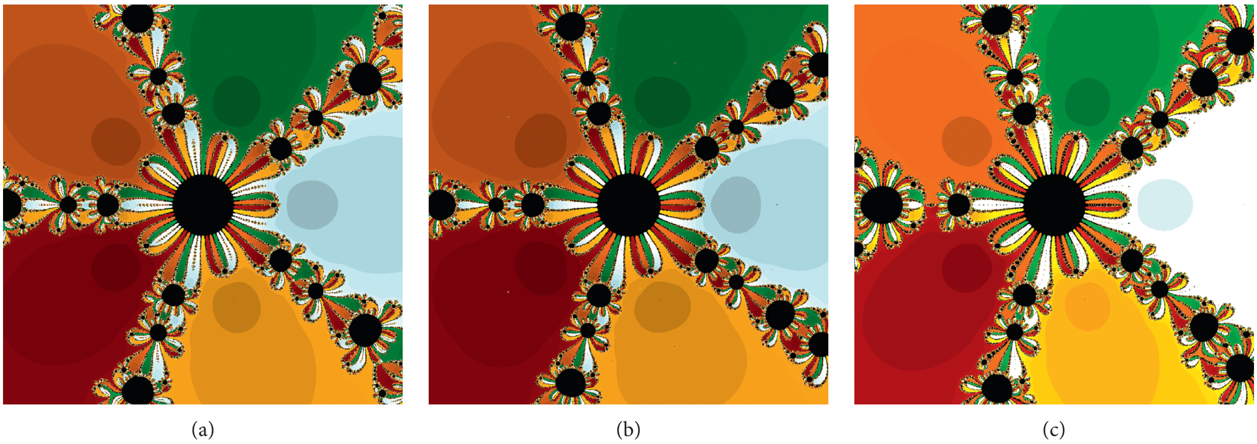

Figure 5: Basins of attraction associated with the polynomial $p_5(z)$ for $\beta = 1$. (a) Stands for Algorithm 1, (b) for Algorithm 2, and (c) for Algorithm 3.

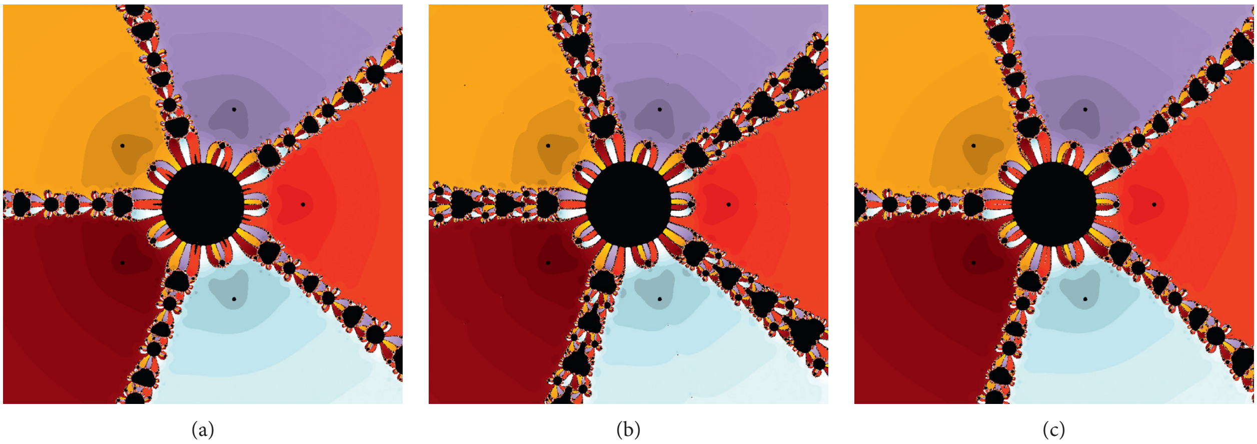

Figure 6: Basins of attraction associated with the polynomial $p_6(z)$ for $\beta = 1$. (a) Stands for Algorithm 1, (b) for Algorithm 2, and (c) for Algorithm 3.

When we look at the generated images, we can read two important characteristics. One of them is the convergence speed of the considered algorithm, which can be depicted by the shade of the color. The darkness of the colors shows less number of iterations of the considered algorithm and vice versa. The second one is the dynamical aspects of the algorithm. Low dynamics are in those specific areas which contain small variation of colors, whereas in areas having large variation of colors, the dynamics are high. The black color in images locates those places where the solution cannot be achieved for the given number of iterations. The areas with the same colors in above figures indicate the same number of iterations needed to approximate the solution and give similar look to the contour lines on the map.

6. Concluding Remarks

Using variational iteration technique, three new root-finding algorithms for the solution of nonlinear equations in one variable have been established, having ninth order of convergence. By using some test examples, the performance and efficiency of the proposed algorithms have been analyzed. Tables 1 and 2 show the best performance of the proposed algorithms in terms of accuracy, speed, number of iterations, and computational order of convergence as compared to other well-known existing iterative methods. We have also presented the basins of attraction using some complex polynomials through newly developed algorithms which describe the fractal behavior and dynamical aspects of the proposed algorithms. The variational iteration technique can be applied to derive a broad range of new algorithms for solving a system of nonlinear equations in one dimension.

Data Availability

All data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All the authors have contributed equally to this paper.

Acknowledgments

The corresponding author Thabet Abdeljawad would like to thank Prince Sultan University for funding this work through research group Nonlinear Analysis Methods in Applied Mathematics (NAMAM) (group number RG-DES-2017-01-17).

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] I. K. Argyros, "A note on the Halley method in Banach spaces," *Applied Mathematics and Computation*, vol. 58, no. 2-3, pp. 215–224, 1993.

[3] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks/Cole Publishing Company, Pacific Grove, CA, USA., Sixth edition, 1997.

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

[5] J. M. Gutiérrez and M. A. Hernández, "An acceleration of Newton's method: super-Halley method," *Applied Mathematics and Computation*, vol. 117, no. 2-3, pp. 223–239, 2001.

[6] J. M. Gutiérrez and M. A. Hernández, "A family of Chebyshev-Halley type methods in Banach spaces," *Bulletin of the Australian Mathematical Society*, vol. 55, no. 1, pp. 113–130, 1997.

[7] A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, UK, 1970.

[8] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Springer-Verlag, New York, NY, USA, 2000.

[9] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, NY, USA, Third edition, 2002.

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

[11] R. Behl and E. Martinez, "A new high-order and efficient family of iterative techniques for nonlinear models," *Complexity*, vol. 2020, Article ID 1706841, 11 pages, 2020.

[12] J.-H. He, "Homotopy perturbation technique," *Computer Methods in Applied Mechanics and Engineering*, vol. 178, no. 3-4, pp. 257–262, 1999.

[13] 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.

[14] J.-H. He, "Variational iteration method—a kind of non-linear analytical technique: some examples," *International Journal of Non-linear Mechanics*, vol. 34, no. 4, pp. 699–708, 1999.

[15] M. Inokuti, H. Sekine, and T. Mura, "General use of the lagrange multiplier in non-linear mathematical physics," in *Variational Methods in the Mechanics of Solids*, S. Nemat-Nasser, Ed., Pergamon Press, Oxford, UK, pp. 156–162, 1978.

[16] J. Kuo, "The improvements of modified Newton's method," *Applied Mathematics and Computation*, vol. 189, pp. 602–609, 2007.

[17] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Numerical algorithms for finding zeros of nonlinear equations and their dynamical aspects," *Journal of Mathematics*, vol. 2020, Article ID 2816843, 11 pages.

[18] W. Nazeer, A. Naseem, S. M. Kang, and Y. C. Kwun, "Generalized Newton Raphsons method free from second derivative," *Journal of Nonlinear Sciences and Applications*, vol. 9, no. 5, pp. 2823–2831, 2016.

[19] W. Nazeer, M. Tanveer, S. M. Kang, and A. Naseem, "A new Householders method free from second derivatives for solving nonlinear equations and polynomiography," *Journal of Nonlinear Sciences and Applications*, vol. 9, no. 3, pp. 998–1007, 2016.

[20] M. A. Noor, "New classes of iterative methods for nonlinear equations," *Applied Mathematics and Computation*, vol. 191, no. 1, pp. 128–131, 2007.

[21] M. A. Noor and F. A. Shah, "Variational iteration technique for solving nonlinear equations," *Journal of Applied Mathematics and Computing*, vol. 31, no. 1-2, pp. 247–254, 2009.

[22] F. Soleimani, F. Soleymani, and S. Shateyi, "Some iterative methods free from derivatives and their basins of attraction for nonlinear equations," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 301718, 10 pages.

[23] M. A. Noor, W. A. Khan, and A. Hussain, "A new modified Halley method without second derivatives for nonlinear equation," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1268–1273, 2007.

[24] 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.

[25] M. S. Rhee, Y. I. Kim, and B. Neta, "An optimal eighth-order class of three-step weighted Newton's methods and their dynamics behind the purely imaginary extraneous fixed points," *International Journal of Computer Mathematics*, vol. 95, no. 11, pp. 2174–2211, 2017.

[26] J. R. Sharma and H. Arora, "A new family of optimal eighth order methods with dynamics for nonlinear equations," *Applied Mathematics and Computation*, vol. 273, pp. 924–933, 2016.

[27] A. Cayley, "The Newton-Fourier imaginary problem," *American Journal of Mathematics*, vol. 2, p. 97, 1879.

[28] K. Gdawiec, "Fractal patterns from the dynamics of combined polynomial root finding methods," *Nonlinear Dynamics*, vol. 90, no. 4, pp. 2457–2479, 2017.

[29] K. Gdawiec, W. Kotarski, and A. Lisowska, "Visual analysis of the Newton's method with fractional order derivatives," *Symmetry*, vol. 11, no. 9, p. 1143, 2019.

[30] B. Kalantari, *Polynomial Root-Finding and Polynomiography*, World Scientific, Hackensack, NJ, USA, 2008.

[31] B. Kalantari and E. H. Lee, "Newton-Ellipsoid polynomiography," *Journal of Mathematics and the Arts*, vol. 13, no. 4, pp. 336–352, 2019.

[32] S. M. Kang, A. Naseem, W. Nazeer, M. Munir, and C. Y. Jung, "Polynomiography of some iterative methods," *International Journal of Mathematical Analysis*, vol. 11, no. 3, pp. 133–149, 2017.

[33] W. Kotarski, K. Gdawiec, and A. Lisowska, "Polynomiography via ishikawa and mann iterations," in *Advances in Visual Computing, Lecture Notes in Computer Science, Part I*, G. Bebis, R. Boyle, B. Parvin et al., Eds., vol. 7431, pp. 305–313, Springer, Berlin, Germany, 2012.

[34] Y. C. Kwun, Z. Majeed, A. Naseem, W. Nazeer, and S. M. Kang, "New iterative methods using variational iteration technique and their dynamical behavior," *International Journal of Pure and Applied Mathematics*, vol. 116, no. 4, pp. 1093–1113, 2017.

[35] B. B. Mandelbrot, *Fractal Geometry of Nature*, W. F. Freeman, NewYork, NY, USA, 1993.

[36] H. Susanto and N. Karjanto, "Newton's method's basins of attraction revisited," *Applied Mathematics and Computation*, vol. 215, no. 3, pp. 1084–1090, 2009.