

Research Article

Numerical Algorithms for Finding Zeros of Nonlinear Equations and Their Dynamical Aspects

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

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

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

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

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

Correspondence should be addressed to Thabet Abdeljawad; tabdeljawad@psu.edu.sa

Received 20 April 2020; Accepted 25 July 2020; Published 30 September 2020

Academic Editor: Francisco Balibrea

Copyright © 2020 Amir Naseem et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we developed two new numerical algorithms for finding zeros of nonlinear equations in one dimension and one of them is second derivative free which has been removed using the interpolation technique. We derive these algorithms with the help of Taylor's series expansion and Golbabai and Javidi's method. The convergence analysis of these algorithms is discussed. It is established that the newly developed algorithms have sixth order of convergence. Several numerical examples have been solved which prove the better efficiency of these algorithms as compared to other well-known iterative methods of the same kind. Finally, the comparison of polynomiographs generated by other well-known iterative methods with our developed algorithms has been made which reflects their dynamical aspects.

1. Introduction

One of the major problems in applied mathematics and engineering sciences is to solve the nonlinear equation of the form:

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

where f is a real-valued function whose domain is an open connected set.

The solution of such type of nonlinear equations cannot be found directly except in special cases. Therefore, we have to adopt iterative methods for solving such type of equations. In iterative methods, we start with an initial guess x_0 which is improved step by step by means of iterations.

We assume that α is a simple zero of equation (1) and x_0 is an initial guess sufficiently close to α .

Using Taylor's series around x_0 for equation (1), we have

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

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

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

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

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

This is so-called Newton's method [1, 2] for root-finding of nonlinear functions, which converges quadratically. From equation (2), one can evaluate

$$x_{k+1} = x_k - \frac{2f(x_k)f'(x_k)}{2f'^2(x_k) - f(x_k)f''(x_k)}. \quad (5)$$

This is so-called Halley's method [3] for root-finding of nonlinear functions, which converges cubically. Simplification of equation (2) yields another iterative method as follows:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{f^2(x_k)f''(x_k)}{2f'^3(x_k)}. \quad (6)$$

This is known as Householder's method [4] for solving nonlinear equations in one variable, which converges cubically.

In recent years, a large number of iterative methods have been developed using different techniques such as decomposition method, Taylor's series, perturbation method, quadrature formula, and variational iteration techniques [5–13] and the references therein.

For improving convergence, various modified methods have been developed in the literature. Some of them are given in [14–18] and the references therein.

In the twentieth century, Ostrowski [19] took Newton's method as a predictor step and proposed a two-step iterative method having fourth convergence order. After that, Traub [12] suggested an iterative scheme in which he took Newton's method as a predictor and corrector step and proved that the proposed method has convergence of fourth order. Noor and Noor, in 2007 [13], developed the two-step Halley's method in which Newton's method was taken as a predictor step and Halley's method as a corrector step and then proved that the suggested method has sixth order of convergence and then removing its second derivative using finite difference scheme and established a new algorithm having fifth order of convergence. In 2018, Kumar et al. [20] established a parameter-based family of algorithm which has sixth convergence order for solving nonlinear equations.

In this paper, we proposed and analyzed two new predictor-corrector type iterative methods, namely, Algorithms 1 and 2 in which we take Newton's method as a predictor step. We proved that these newly developed algorithms have sixth order of convergence and are most efficient as compared to other well-known iterative methods of the same kind. The proposed algorithms applied to solve some test examples in order to assess its validity and accuracy. In the last section, we generate the polynomiographs of complex polynomials of different degrees through our proposed algorithms and compare it with other methods of the same category. The presented polynomiographs have very interesting and aesthetic patterns which reflects different properties of the polynomials.

2. Main Results

Let $f: X \rightarrow R$, $X \subseteq R$ is a scalar function, then by using the basic idea of modified homotopy perturbation method, Golbabai and Javidi [21] derive the following method:

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{f^2(x_0)f''(x_0)}{2[f'^3(x_0) - f(x_0)f'(x_0)f''(x_0)]}. \quad (7)$$

In iterative form:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)f''(x_n)}{2[f'^3(x_n) - f(x_n)f'(x_n)f''(x_n)]}, \quad (8)$$

which is known as Golbabai and Javidi's method [21] having cubic order of convergence.

After simplification of equation (2), one can obtain

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(x - x_0)^2 f''(x_0)}{2f'(x_0)}. \quad (9)$$

Now from Golbabai and Javidi's method in equation (7),

$$x - x_0 = \frac{f(x_0)}{f'(x_0)} - \frac{f^2(x_0)f''(x_0)}{2[f'^3(x_0) - f(x_0)f'(x_0)f''(x_0)]}. \quad (10)$$

With the help of equations (9) and (10), we obtain

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{f^2(x_0)f''(x_0)[2f'^2(x_0) - f(x_0)f''(x_0)]^2}{8f'^3(x_0)[f'^2(x_0) - f(x_0)f''(x_0)]}. \quad (11)$$

Rewriting the above equation with Newton's method as a predictor gives us a new algorithm as follows.

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

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots,$$

$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{f^2(y_n)f''(y_n)[2f'^2(y_n) - f(y_n)f''(y_n)]^2}{8f'^3(y_n)[f'^2(y_n) - f(y_n)f''(y_n)]}, \quad (12)$$

which is the modification of Golbabai and Javidi's method with Newton's method as a predictor.

In order to find the solution of the given nonlinear equation, we have to calculate the first as well as the second derivative of the function $f(x)$. But in several cases, we face such a situation where the second derivative of the function does not exist and our method fails to find the solution.

To overcome this difficulty, we use interpolation technique for approximating the second derivative as follows:

Consider the function

$$D(s) = a + b(s - y_n) + c(s - y_n)^2 + d(s - y_n)^3, \quad (13)$$

where a , b , c , and d are unknowns to be found with the following interpolation conditions:

$$\begin{aligned}
 f(x_n) &= D(x_n), \\
 f(w_n) &= D(w_n), \\
 f'(x_n) &= D'(x_n), \\
 f'(w_n) &= D'(w_n), \\
 f''(w_n) &= D''(w_n).
 \end{aligned}
 \tag{14}$$

A system of four linear equations of four variables is obtained from the above conditions. Solving this system gives

$$f''(y_n) = \frac{6[f(x_n) - f(y_n)] - 2[x_n - y_n][2f'(y_n) + f'(x_n)]}{(x_n - y_n)^2} = D(x_n, y_n).
 \tag{15}$$

Using the above equation in Algorithm 1, we derive a new algorithm free from second derivative as follows.

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

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots, \\
 x_{n+1} &= y_n - \frac{f(y_n)}{f'(y_n)} \\
 &\quad - \frac{f^2(y_n)D(x_n, y_n)[2f'^2(y_n) - f(y_n)D(x_n, y_n)]^2}{8f'^3(y_n)[f'^2(y_n) - f(y_n)D(x_n, y_n)]},
 \end{aligned}
 \tag{16}$$

which is a new two-step iterative method free from second derivative, having Newton's method as a predictor step. With the help of this method, we can solve such type of nonlinear equations in which second derivative does not exist. Also this requires only two evaluations of the function and one of its derivatives which shows better efficiency of this method as compared to those which require second derivative. Several examples are given which proved best performance of this method as compared to other well-known iterative methods of the same kind.

3. Convergence Analysis

In this section, we discuss the convergence order of the purposed iterative methods.

Theorem 1. *Suppose that α is a root of the equation $f(x) = 0$. If $f(x)$ is sufficiently smooth in the neighborhood of α , then the order of convergence of Algorithm 1 is six.*

Proof. To analyze the convergence of Algorithm 1, suppose that α is a root of the equation $f(x) = 0$ and e_n be the error at

n th iteration, then $e_n = x_n - \alpha$, and by using Taylor series expansion, we have

$$\begin{aligned}
 f(x_n) &= f'(\alpha)e_n + \frac{1}{2!}f''(\alpha)e_n^2 + \frac{1}{3!}f'''(\alpha)e_n^3 + \frac{1}{4!}f^{(iv)}(\alpha)e_n^4 \\
 &\quad + \frac{1}{5!}f^{(v)}(\alpha)e_n^5 + \frac{1}{6!}f^{(vi)}(\alpha)e_n^6 + O(e_n^7), \\
 f(x_n) &= f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + O(e_n^7)],
 \end{aligned}
 \tag{17}$$

$$\begin{aligned}
 f'(x_n) &= f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 \\
 &\quad + 6c_6e_n^5 + 7c_7e_n^6 + O(e_n^7)],
 \end{aligned}
 \tag{18}$$

where $c_n = f^{(n)}(\alpha)/n!f'(\alpha)$.

With the help of equations (17) and (18), we get

$$\begin{aligned}
 y_n &= \alpha + c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + (3c_4 - 7c_3c_2 + 4c_2^3)e_n^4 \\
 &\quad + (-6c_3^3 + 20c_3c_2^2 - 10c_2c_4 + 4c_5 - 8c_2^4)e_n^5 \\
 &\quad + (-17c_3c_4 + 28c_4c_2^2 - 13c_2c_5 + 5c_6 + 33c_2^3c_3 \\
 &\quad - 52c_3c_2^3 + 16c_2^5)e_n^6 + O(e_n^7),
 \end{aligned}
 \tag{19}$$

$$\begin{aligned}
 f(y_n) &= f'(\alpha)[c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + (5c_2^3 - 7c_3c_2 + 3c_4)e_n^4 \\
 &\quad + (24c_3c_2^2 - 12c_2^4 - 10c_2c_4 + 4c_5 - 6c_2^3)e_n^5 \\
 &\quad + (-73c_3c_2^3 + 34c_4c_2^2 + 28c_2^5 + 37c_2c_3^2 - 17c_4c_3 \\
 &\quad - 13c_2c_5 + 5c_6)e_n^6 + O(e_n^7)],
 \end{aligned}
 \tag{20}$$

$$\begin{aligned}
 f'(y_n) &= f'(\alpha)[1 + 2c_2^2e_n^2 + 4(c_3c_2 - c_2^3)e_n^3 + (6c_2c_4 - 11c_3c_2^2 \\
 &\quad + 8c_2^4)e_n^4 + (28c_3c_2^3 - 20c_4c_2^2 + 8c_2c_5 - 16c_2^5)e_n^5 \\
 &\quad + (-16c_3c_2c_4 + 12c_2^3 + 60c_2^3c_4 - 26c_5c_2^2 \\
 &\quad + 10c_2c_6 + 32c_2^6)e_n^6 + O(e_n^7)],
 \end{aligned}
 \tag{21}$$

$$\begin{aligned}
f''(y_n) = f'(\alpha) & \left[2c_2 + 6c_3c_2e_n^2 + 12(c_3^2 - c_3c_2^2)e_n^3 \right. \\
& + (-42c_2c_3^2 + 18c_4c_3 + 24c_3c_2^3 + 12c_4c_2^2)e_n^4 \\
& + (-12c_3c_2c_4 + 24c_5c_3 - 36c_3^3 + 120c_2^2c_3^2 \\
& - 48c_3c_2^4 - 48c_4c_2^3)e_n^5 + (-78c_2c_5c_3 + 30c_3c_6 \\
& - 54c_4c_3^2 - 96c_3c_4c_2^2 + 198c_2c_3^3 - 312c_3^2c_2^3 + 96c_2^5c_3 \\
& \left. + 72c_2c_4^2 + 144c_4c_2^4 + 20c_5c_2^3)e_n^6 + O(e_n^7) \right]. \tag{22}
\end{aligned}$$

Using equations (19)–(22) in Algorithm 1, we get

$$x_{n+1} = \alpha + (-c_3c_2^3)e_n^6 + O(e_n^7), \tag{23}$$

which implies that

$$e_{n+1} = (-c_3c_2^3)e_n^6 + O(e_n^7). \tag{24}$$

The above equation shows that the order of convergence of Algorithm 1 is six. \square

Theorem 2. *Suppose that α is a root of the equation $f(x) = 0$. If $f(x)$ is sufficiently smooth in the neighborhood of α , then the order of convergence of Algorithm 2 is six.*

Proof. With the help of equations (17)–(21) along with the same assumptions of the previous theorem, we have

$$\begin{aligned}
D(x_n, y_n) = f'(\alpha) & \left[2c_2 + (6c_3c_2 - 2c_4)e_n^2 \right. \\
& + (12c_3^2 - 12c_3c_2^2 + 4c_2c_4 - 4c_5)e_n^3 \\
& + (2c_2c_5 + 26c_4c_3 - 42c_2c_3^2 + 24c_3c_2^3 \\
& + 2c_4c_2^2 - 6c_6)e_n^4 + (-48c_3c_2c_4 + 12c_4^2 - 24c_4c_2^3 \\
& + 28c_5c_3 + 4c_5c_2^2 + 120c_3^2c_2^2 - 48c_3c_2^4 \\
& - 8c_7 - 36c_3^3)e_n^5 + (-60c_2c_5c_3 + 28c_4c_3c_2^2 - 2c_2c_7 \\
& + 22c_5c_4 - 10c_5c_2^3 + 30c_6c_3 + 6c_6c_2^2 + 20c_2c_4^2 \\
& - 86c_4c_2^3 + 88c_4c_2^4 + 198c_2c_3^3 - 312c_3^2c_2^3 + 96c_3c_2^5 \\
& \left. - 10c_8)e_n^6 + O(e_n^7) \right]. \tag{25}
\end{aligned}$$

Using equations (20), (21), and (25) in Algorithm 2, we get

$$x_{n+1} = \alpha + (-c_3c_2^3 + c_4c_2^2)e_n^6 + O(e_n^7), \tag{26}$$

which implies that

$$e_{n+1} = (-c_3c_2^3 + c_4c_2^2)e_n^6 + O(e_n^7). \tag{27}$$

The above equation shows that the order of convergence of Algorithm 2 is six. \square

4. Numerical Examples

In this section, we include some nonlinear functions to illustrate the efficiency of our newly developed numerical

algorithms. We compare these algorithms with Noor's method one (NR1) [22], Noor's method two (NR2) [22], Ostrowski's method (OM) [19], Traub's method (TM) [12], and modified Halley's method (MHM) [13].

For this purpose, the following numerical examples have been solved:

$$f_1(x) = x^3 + 4x^2 - 10,$$

$$f_2(x) = e^x + \cos(\pi x),$$

$$f_3(x) = x^2 + \sin\left(\frac{x}{5}\right) - \frac{1}{4},$$

$$f_4(x) = xe^x - 1, \tag{28}$$

$$f_5(x) = x^3 - 10,$$

$$f_6(x) = \cos(x) - x,$$

$$f_7(x) = \tan^{-1}(x) + x.$$

Here, we take $\varepsilon = 10^{-15}$ in the following stopping criteria $|x_{n+1} - x_n| < \varepsilon$. The numerical examples have been solved using the computer program Maple 13.

Tables 1–7 show the numerical comparisons of our developed algorithms with Noor's method one (NR1), Noor's method two (NR2), Ostrowski's method (OM), Traub's method (TM), and modified Halley's method (MHM). 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} , the difference between two consecutive approximations $|x_{n+1} - x_n|$ of the equation and the computational order of convergence (COC) which can be approximated using the following formula:

$$\text{COC} \approx \frac{\ln(|x_{n+1} - x_n|/|x_n - x_{n-1}|)}{\ln(|x_n - x_{n-1}|/|x_{n-1} - x_{n-2}|)}, \tag{29}$$

which was introduced by Weerakoon and Fernando in (2000) [23].

Tables 8 shows the comparison of the number of iterations required for different iterative methods with our developed algorithms to approximate the root of the given nonlinear function for the stopping criteria $\varepsilon = 10^{-100}$. The columns represent the number of iterations for different functions along with initial guess x_0 . All calculations have been carried out using the computer program Maple 13.

5. Polynomiography

Polynomials are one of the most significant objects in many fields of mathematics. Polynomial root-finding has played a key role in the history of mathematics. It is one of the oldest and most deeply studied mathematical problems. The last interesting contribution to the polynomials root-finding history was made by Kalantari [24], who introduced the polynomiography. As a method which generates nice-

TABLE 1: Comparison of various iterative methods.

| Method | N | $ f(x_{n+1}) $ | x_{n+1} | $\sigma = x_{n+1} - x_n $ | COC |
|----------------------|-----|-------------------|------------------------|----------------------------|-----|
| $f_1(x), x_0 = -0.4$ | | | | | |
| NR1 | 27 | $3.710177e^{-28}$ | 1.36523001341409684580 | $6.769715e^{-15}$ | 2 |
| NR2 | 4 | $1.813409e^{-36}$ | | $7.403402e^{-13}$ | 3 |
| OM | 5 | $7.395989e^{-45}$ | | $8.442974e^{-12}$ | 4 |
| TM | 84 | $1.071491e^{-43}$ | | $1.531881e^{-11}$ | 4 |
| MHM | 23 | $7.392887e^{-38}$ | | $4.595974e^{-08}$ | 5 |
| Algorithm 1 | 4 | $3.518283e^{-46}$ | | $3.794720e^{-08}$ | 6 |
| Algorithm 2 | 4 | $3.518283e^{-46}$ | | $3.794720e^{-08}$ | 6 |

TABLE 2: Comparison of various iterative methods.

| Method | N | $ f(x_{n+1}) $ | x_{n+1} | $\sigma = x_{n+1} - x_n $ | COC |
|----------------------|-----|-------------------|-------------------------|----------------------------|-----|
| $f_2(x), x_0 = -1.9$ | | | | | |
| NR1 | 17 | $3.818521e^{-23}$ | -0.52248077281054548914 | $7.695727e^{-12}$ | 2 |
| NR2 | 90 | $9.891015e^{-41}$ | | $3.435165e^{-14}$ | 3 |
| OM | 88 | $7.511035e^{-25}$ | | $1.017176e^{-06}$ | 4 |
| TM | 18 | $8.259938e^{-20}$ | | $5.124588e^{-05}$ | 4 |
| MHM | 9 | $1.268356e^{-48}$ | | $4.262132e^{-11}$ | 5 |
| Algorithm 1 | 8 | $1.013204e^{-25}$ | | $1.409549e^{-04}$ | 6 |
| Algorithm 2 | 3 | $1.837741e^{-82}$ | | $5.335861e^{-14}$ | 6 |

TABLE 3: Comparison of various iterative methods.

| Method | N | $ f(x_{n+1}) $ | x_{n+1} | $\sigma = x_{n+1} - x_n $ | COC |
|-------------------|-----|-------------------|------------------------|----------------------------|-----|
| $f_3(x), x_0 = 2$ | | | | | |
| NR1 | 5 | $3.163807e^{-17}$ | 0.40999201798913713162 | $5.629386e^{-09}$ | 2 |
| NR2 | 45 | $3.382231e^{-18}$ | | $1.512669e^{-06}$ | 3 |
| OM | 4 | $6.783183e^{-56}$ | | $1.630792e^{-14}$ | 4 |
| TM | 4 | $6.063382e^{-56}$ | | $1.586231e^{-14}$ | 4 |
| MHM | 3 | $4.499472e^{-42}$ | | $1.880726e^{-08}$ | 5 |
| Algorithm 1 | 2 | $1.658501e^{-15}$ | | $1.059288e^{-02}$ | 6 |
| Algorithm 2 | 2 | $1.670194e^{-15}$ | | $1.059787e^{-02}$ | 6 |

TABLE 4: Comparison of various iterative methods.

| Method | N | $ f(x_{n+1}) $ | x_{n+1} | $\sigma = x_{n+1} - x_n $ | COC |
|-------------------|-----|-------------------|------------------------|----------------------------|-----|
| $f_4(x), x_0 = 1$ | | | | | |
| NR1 | 4 | $4.418933e^{-19}$ | 0.56714329040978387300 | $4.418705e^{-10}$ | 2 |
| NR2 | 4 | $9.674661e^{-39}$ | | $1.596435e^{-13}$ | 3 |
| OM | 3 | $8.984315e^{-40}$ | | $1.921047e^{-10}$ | 4 |
| TM | 3 | $2.130596e^{-33}$ | | $6.120512e^{-09}$ | 4 |
| MHM | 3 | $1.116440e^{-68}$ | | $2.540557e^{-14}$ | 5 |
| Algorithm 1 | 2 | $5.068654e^{-20}$ | | $6.667718e^{-04}$ | 6 |
| Algorithm 2 | 2 | $2.535508e^{-21}$ | | $4.396326e^{-04}$ | 6 |

TABLE 5: Comparison of various iterative methods.

| Method | N | $ f(x_{n+1}) $ | x_{n+1} | $\sigma = x_{n+1} - x_n $ | COC |
|---------------------|-----|-------------------|------------------------|----------------------------|-----|
| $f_5(x), x_0 = 0.5$ | | | | | |
| NR1 | 100 | $7.261452e^{-18}$ | 2.15443469003188372180 | $1.059948e^{-09}$ | 2 |
| NR2 | 11 | $4.753110e^{-35}$ | | $2.385814e^{-12}$ | 3 |
| OM | 5 | $5.537625e^{-30}$ | | $4.942045e^{-08}$ | 4 |
| TM | 5 | $4.893426e^{-16}$ | | $1.369225e^{-04}$ | 4 |
| MHM | 4 | $4.922706e^{-19}$ | | $2.732126e^{-04}$ | 5 |
| Algorithm 1 | 3 | $2.129060e^{-18}$ | | $1.663953e^{-03}$ | 6 |
| Algorithm 2 | 3 | $2.129060e^{-18}$ | | $1.663953e^{-03}$ | 6 |

TABLE 6: Comparison of various iterative methods.

| Method | N | $ f(x_{n+1}) $ | x_{n+1} | $\sigma = x_{n+1} - x_n $ | COC |
|---------------------|-----|-------------------|------------------------|----------------------------|-----|
| $f_6(x), x_0 = 2.5$ | | | | | |
| NR1 | 6 | $4.872692e^{-21}$ | 0.73908513321516064166 | $6.769715e^{-10}$ | 2 |
| NR2 | 4 | $3.229578e^{-29}$ | | $7.403402e^{-09}$ | 3 |
| OM | 3 | $2.576412e^{-26}$ | | $8.442974e^{-07}$ | 4 |
| TM | 3 | $2.617193e^{-35}$ | | $1.531881e^{-09}$ | 4 |
| MHM | 3 | $1.154792e^{-54}$ | | $4.595974e^{-11}$ | 5 |
| Algorithm 1 | 2 | $8.918884e^{-21}$ | | $3.794720e^{-03}$ | 6 |
| Algorithm 2 | 2 | $1.374491e^{-18}$ | | $3.794720e^{-03}$ | 6 |

TABLE 7: Comparison of various iterative methods.

| Method | N | $ f(x_{n+1}) $ | x_{n+1} | $\sigma = x_{n+1} - x_n $ | COC |
|-------------------|-----|--------------------|------------------------|----------------------------|-----|
| $f_7(x), x_0 = 2$ | | | | | |
| NR1 | 4 | $2.455690e^{-16}$ | 0.00000000000000000000 | $9.031605e^{-06}$ | 2 |
| NR2 | 4 | $3.424681e^{-40}$ | | $1.271327e^{-13}$ | 3 |
| OM | 3 | $2.704322e^{-41}$ | | $1.194704e^{-08}$ | 4 |
| TM | 3 | $7.907266e^{-122}$ | | $5.282400e^{-14}$ | 4 |
| MHM | 3 | $1.556811e^{-66}$ | | $6.000887e^{-10}$ | 5 |
| Algorithm 1 | 2 | $2.674407e^{-21}$ | | $8.437184e^{-03}$ | 6 |
| Algorithm 2 | 2 | $3.197382e^{-15}$ | | $3.593740e^{-02}$ | 6 |

TABLE 8: Comparison of the number of iterations required for different iterative methods for $\epsilon = 10^{-100}$.

| Methods | Functions | | | | | | |
|-------------|---------------------|---------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| | $f_1(x) x_0 = -0.4$ | $f_2(x) x_0 = -1.9$ | $f_3(x) x_0 = 2.0$ | $f_4(x) x_0 = 1.0$ | $f_5(x) x_0 = 0.5$ | $f_6(x) x_0 = 2.5$ | $f_7(x) x_0 = 2.0$ |
| NR1 | 29 | 20 | 8 | 7 | 103 | 9 | 6 |
| NR2 | 5 | 91 | 6 | 5 | 12 | 6 | 5 |
| OM | 6 | 89 | 5 | 4 | 6 | 4 | 4 |
| TM | 85 | 20 | 5 | 4 | 7 | 4 | 3 |
| MHM | 24 | 10 | 4 | 4 | 6 | 4 | 4 |
| Algorithm 1 | 5 | 9 | 4 | 3 | 4 | 3 | 3 |
| Algorithm 2 | 5 | 4 | 4 | 3 | 4 | 3 | 3 |

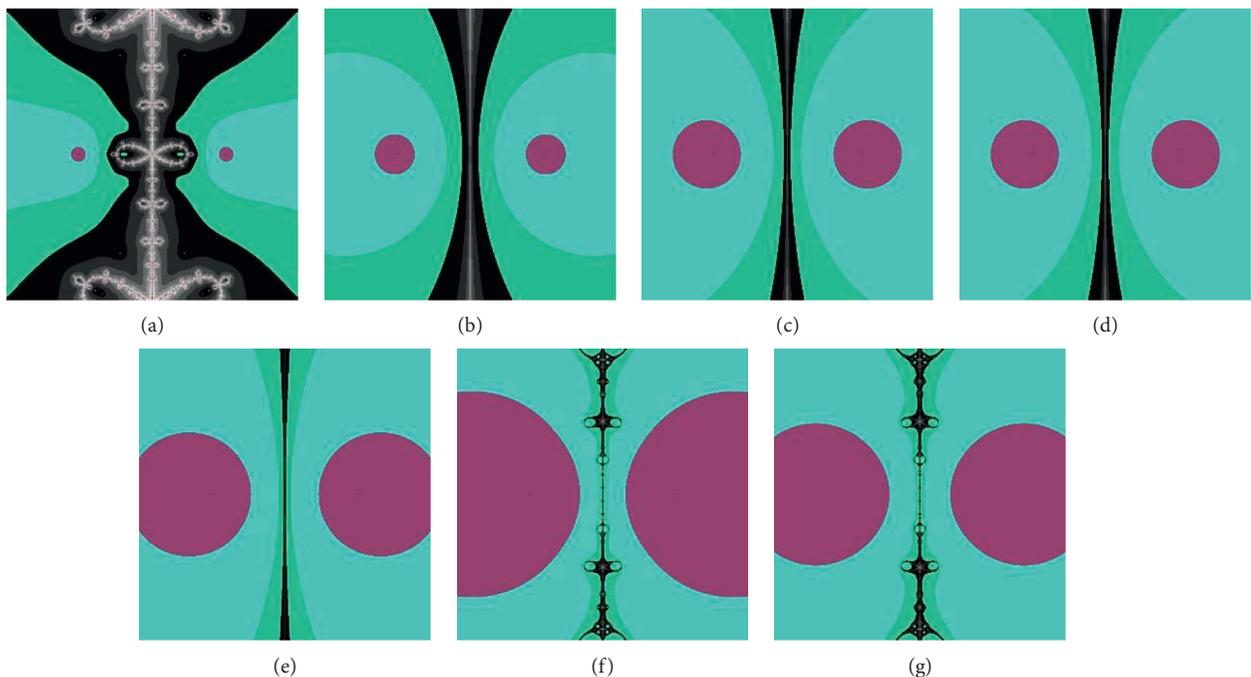


FIGURE 1: Polynomiographs for the polynomial $p_1(z)$ using various iterative methods. (a) Polynomiograph for the polynomial $p_1(z)$ using NR1. (b) Polynomiograph for the polynomial $p_1(z)$ using NR2. (c) Polynomiograph for the polynomial $p_1(z)$ using OM. (d) Polynomiograph for the polynomial $p_1(z)$ using TM. (e) Polynomiograph for the polynomial $p_1(z)$ using MHM. (f) Polynomiograph for the polynomial $p_1(z)$ using Algorithm 1. (g) Polynomiograph for the polynomial $p_1(z)$ using Algorithm 2.

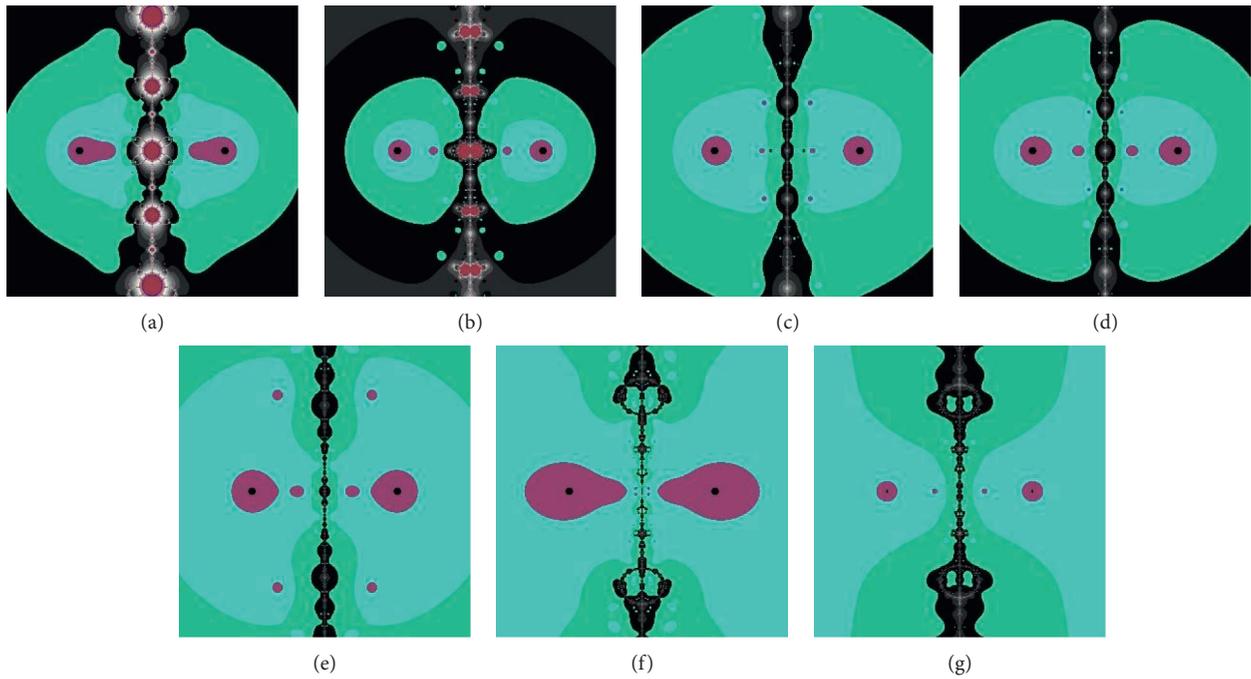


FIGURE 2: Polynomiographs for the polynomial $p_2(z)$ using various iterative methods. (a) Polynomiograph for the polynomial $p_2(z)$ using NR1. (b) Polynomiograph for the polynomial $p_2(z)$ using NR2. (c) Polynomiograph for the polynomial $p_2(z)$ using OM. (d) Polynomiograph for the polynomial $p_2(z)$ using TM. (e) Polynomiograph for the polynomial $p_2(z)$ using MHM. (f) Polynomiograph for the polynomial $p_2(z)$ using Algorithm 1. (g) Polynomiograph for the polynomial $p_2(z)$ using Algorithm 2.

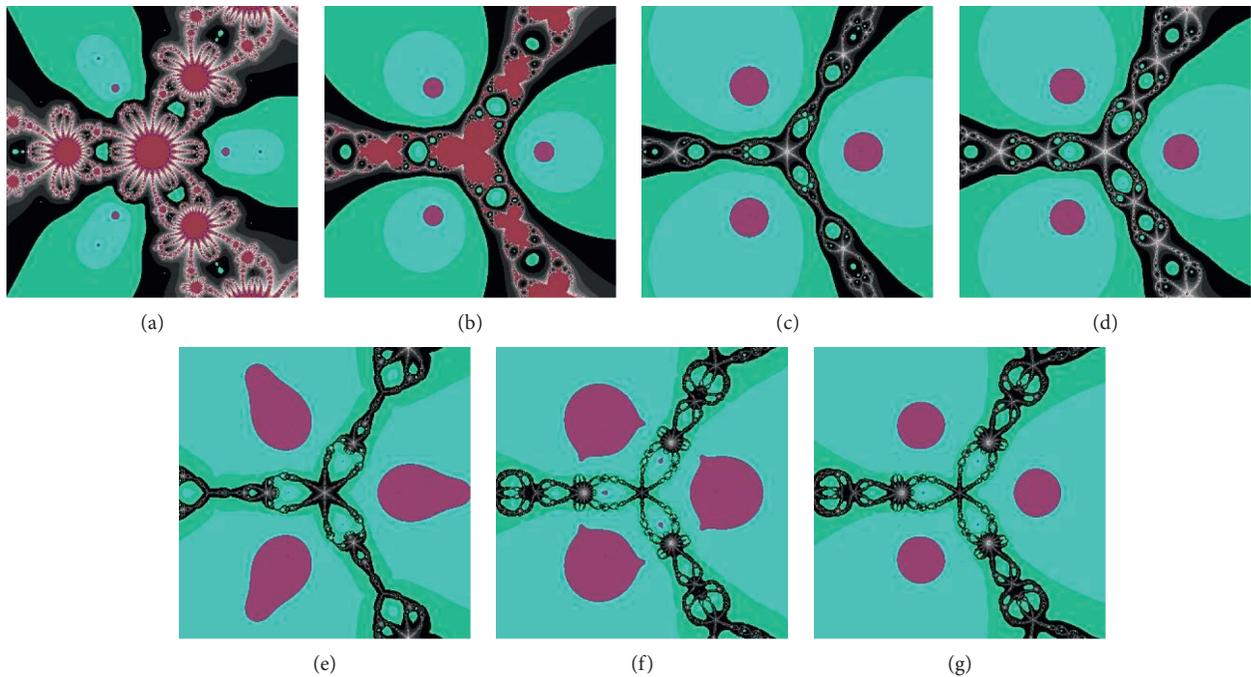


FIGURE 3: Polynomiographs for the polynomial $p_3(z)$ using various iterative methods. (a) Polynomiograph for the polynomial $p_3(z)$ using NR1. (b) Polynomiograph for the polynomial $p_3(z)$ using NR2. (c) Polynomiograph for the polynomial $p_3(z)$ using OM. (d) Polynomiograph for the polynomial $p_3(z)$ using TM. (e) Polynomiograph for the polynomial $p_3(z)$ using MHM. (f) Polynomiograph for the polynomial $p_3(z)$ using Algorithm 1. (g) Polynomiograph for the polynomial $p_3(z)$ using Algorithm 2.

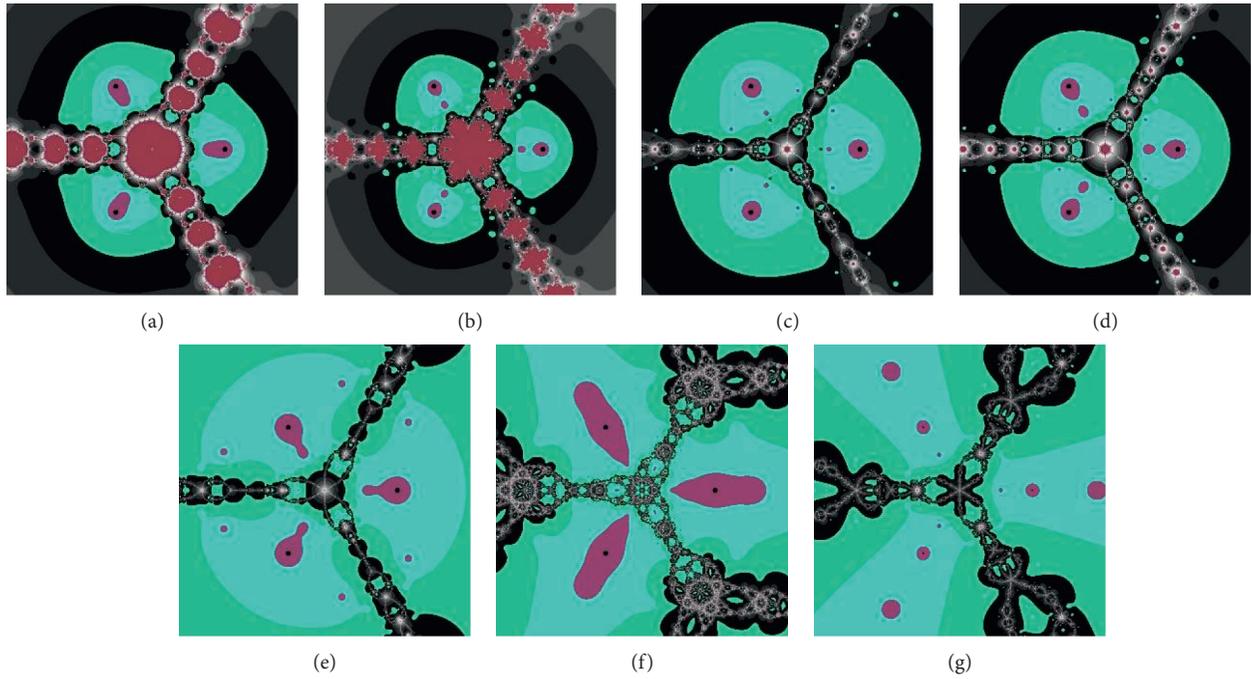


FIGURE 4: Polynomiographs for the polynomial $p_4(z)$ using various iterative methods. (a) Polynomiograph for the polynomial $p_4(z)$ using NR1. (b) Polynomiograph for the polynomial $p_4(z)$ using NR2. (c) Polynomiograph for the polynomial $p_4(z)$ using OM. (d) Polynomiograph for the polynomial $p_4(z)$ using TM. (e) Polynomiograph for the polynomial $p_4(z)$ using MHM. (f) Polynomiograph for the polynomial $p_4(z)$ using Algorithm 1. (g) Polynomiograph for the polynomial $p_4(z)$ using Algorithm 2.

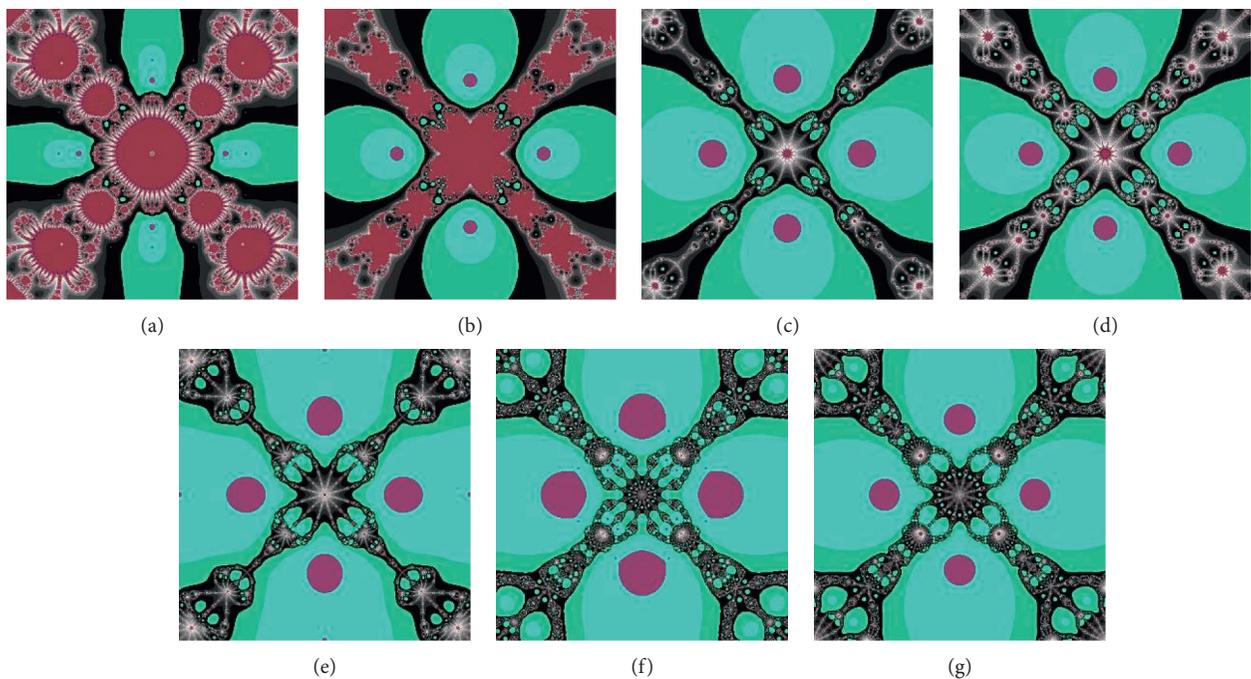


FIGURE 5: Polynomiographs for the polynomial $p_5(z)$ using various iterative methods. (a) Polynomiograph for the polynomial $p_5(z)$ using NR1. (b) Polynomiograph for the polynomial $p_5(z)$ using NR2. (c) Polynomiograph for the polynomial $p_5(z)$ using OM. (d) Polynomiograph for the polynomial $p_5(z)$ using TM. (e) Polynomiograph for the polynomial $p_5(z)$ using MHM. (f) Polynomiograph for the polynomial $p_5(z)$ using Algorithm 1. (g) Polynomiograph for the polynomial $p_5(z)$ using Algorithm 2.

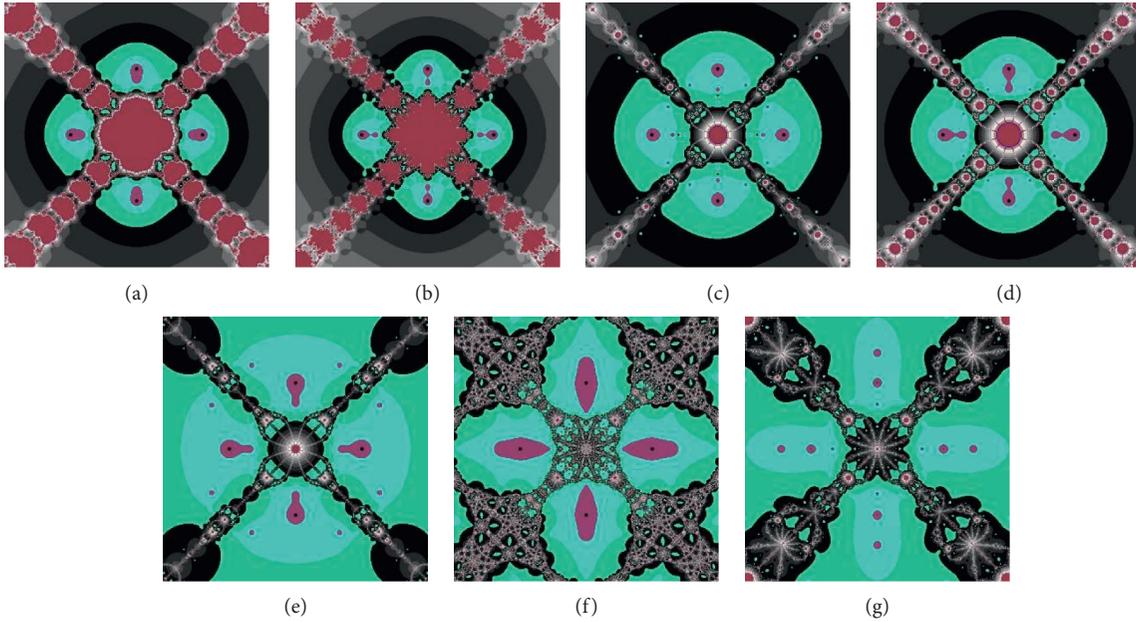


FIGURE 6: Polynomiographs for the polynomial $p_6(z)$ using various iterative methods. (a) Polynomiograph for the polynomial $p_6(z)$ using NR1. (b) Polynomiograph for the polynomial $p_6(z)$ using NR2. (c) Polynomiograph for the polynomial $p_6(z)$ using OM. (d) Polynomiograph for the polynomial $p_6(z)$ using TM. (e) Polynomiograph for the polynomial $p_6(z)$ using MHM. (f) Polynomiograph for the polynomial $p_6(z)$ using Algorithm 1. (g) Polynomiograph for the polynomial $p_6(z)$ using Algorithm 2.

looking graphics, it was patented by Kalantari in USA in 2005 [25]. Polynomiography is defined to be “the art and science of visualization in approximation of the zeros of complex polynomials, via fractal, and nonfractal images created using the mathematical convergence properties of iteration functions” [24]. An individual image is called a “polynomiograph”. Polynomiography combines both art and science aspects. Polynomiography gives a new way to solve the ancient problem by using new algorithms and computer technology. Polynomiography is based on the use of one or an infinite number of iteration methods formulated for the purpose of approximation of the roots of polynomials, e.g., Newton’s method and Halley’s method.

The word “fractal,” which partially appeared in the definition of polynomiography, was coined by the famous mathematician Benoit Mandelbrot [26]. Both fractal images and polynomiographs can be obtained via different iterative schemes. Fractals are self-similar having typical structure and independent of scale. On the other hand, polynomiographs are quite different. The “polynomiographer” can control the shape and design in a more predictable way by using different iteration methods to the infinite variety of complex polynomials. Generally, fractals and polynomiographs belong to different classes of graphical objects. Polynomiography has diverse applications in Mathematics, Science, Education, Art, and Design.

According to the fundamental theorem of algebra, any complex polynomial with complex coefficients

$$p(z) = a_0 + a_1z + a_2z^2 + \dots + a_{n-1}z^{n-1} + a_nz^n, \quad (30)$$

or by its zeros $\{r_1, r_2, r_3, \dots, r_{n-1}, r_n\}$

$$p(z) = (z - r_1)(z - r_2) \dots (z - r_{n-1})(z - r_n), \quad (31)$$

of degree n has n zeros which may or may not be distinct. The degree of polynomial describes the number of basins of attraction and localization of basins can be controlled by placing roots on the complex plane manually. Usually, polynomiographs are colored based on the number of iterations needed to obtain the approximation of some polynomial root with a given accuracy and a chosen iteration method. The description of polynomiography, its theoretical background, and artistic applications are described in [7, 8, 24–29].

6. Applications

In the numerical algorithms that are based on the iterative processes, we need a stopping criterion for the process, that is, a test that tells us that the process has converged or it is very near to the solution. This type of test is called a convergence test. Usually, in the iterative process that use a feedback, like the root-finding methods, the standard convergence test has the following form:

$$|z_{n+1} - z_n| < \varepsilon, \quad (32)$$

where z_{n+1} and z_n are two successive points in the iteration process and $\varepsilon > 0$ is a given accuracy. In this paper, we also use stopping criterion (32). The different colors of an image depend on the number of iterations to reach a root with given accuracy ε . One can obtain infinitely many nice-looking polynomiographs by changing parameter k , where k is the upper bound of the number of iterations.

Here, we present some examples of the following complex polynomials using our developed algorithms and compare it with the polynomiographs obtained by using other well-known two-step iterative methods:

$$\begin{aligned}
 p_1(z) &= z^2 - 1, \\
 p_2(z) &= (z^2 - 1)^2, \\
 p_3(z) &= z^3 - 1, \\
 p_4(z) &= (z^3 - 1)^2, \\
 p_5(z) &= z^4 - 1, \\
 p_6(z) &= (z^4 - 1)^2.
 \end{aligned} \tag{33}$$

In Figures 1–6, polynomiographs of different complex polynomials for Noor’s method one (NR1), Noor’s method two (NR2), Ostrowski’s method (OM), Traub’s method (TM), modified Halley’s method (MHM), and our developed algorithms have been shown which describe the regions of convergence of these polynomials. When we look at the generated images, we can read two important characteristics. The first one is the speed of convergence of the algorithm, i.e., the color of each point gives us information on how many iterations were performed by the algorithm to reach the root. The second characteristic is the dynamics of the algorithm. Low dynamics is in areas where the variation of colors is small, whereas in areas with a large variation of colors, the dynamics is high. The black color in images shows that places where the solution cannot be achieved for the given number of iterations. The areas of the same colors in the above figures indicate the same number of iterations required to determine the solution and they look similar to the contour lines on the map.

All these figures have been generated using the computer program Mathematica 10.0 by taking $\varepsilon = 10^{-03}$ and $k = 15$ where ε shows the accuracy of the given root and k represents the upper bound of the number of iterations.

7. Conclusions

Two new numerical algorithms for finding zeros of nonlinear equations have been developed, having convergence of order six. By using some test examples, the performance of the proposed algorithms is also discussed. The numerical results uphold the analysis of the convergence which can be seen in Tables 1–8. Both algorithms show better results in terms of the number of iterations, efficiency, and convergence order in comparison with other well-known two-step iterative methods of the same kind. Polynomiographs of complex polynomials of different degrees using two-step iterative methods and our proposed algorithms have been generated. The presented polynomiographs are rich and colorful and have very interesting and aesthetic patterns, which reflects the dynamical aspects of our proposed algorithms.

Data Availability

All data required for this paper are included within this paper.

Conflicts of Interest

The authors do not have any conflicts of interest.

Authors’ Contributions

All authors contributed equally to this paper.

References

- [1] D. E. Kincaid and E. W. Cheney, *Numerical Analysis*, Brooks/Cole Publishing Company, Pacific Grove, CA, USA, 1990.
- [2] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks/Cole Publishing Company, Pacific Grove, CA, USA, 6th edition, 1997.
- [3] 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.
- [4] A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, NY, USA, 1970.
- [5] M. Nawaz, A. Naseem, and W. Nazeer, “New twelfth order algorithms for solving nonlinear equations by using variational iteration technique,” *Journal of Prime Research in Mathematics*, vol. 14, pp. 24–36, 2018.
- [6] 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. 09, no. 5, pp. 2823–2831, 2016.
- [7] 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.
- [8] S. M. Kang, A. Naseem, W. Nazeer, and A. Jan, “Modification of abbasbandy’s method and polynomigraphy,” *International Journal of Mathematical Analysis*, vol. 10, no. 24, pp. 1197–1210, 2016.
- [9] W. Nazeer, M. Tanveer, S. M. Kang, and A. Naseem, “A new householders method free from second derivatives for solving nonlinear equations and polynomigraphy,” *Journal of Nonlinear Sciences and Applications*, vol. 09, no. 03, pp. 998–1007, 2016.
- [10] I. Muhammad, H. Ma, A. Naseem, S. Ali, and A. Nizami, “New algorithms for nonlinear equations,” *International Journal of Advanced and Applied Sciences*, vol. 5, pp. 28–32, 2018.
- [11] O. A. Taiwo and O. S. Odetunde, “Approximate solution of variational problems by an iterative decomposition method,” *Maejo International Journal of Science and Technology*, vol. 3, no. 03, pp. 426–433, 2009.
- [12] J. F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea Publishing company, New York, NY, USA, 1982.
- [13] 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, pp. 1268–1273, 2007.

- [14] J.-H. He, "Homotopy perturbation technique," *Computer Methods in Applied Mechanics and Engineering*, vol. 178, no. 3-4, pp. 257-262, 1999.
- [15] J. Kuo, "The improvements of modified Newton's method," *Applied Mathematics and Computation*, vol. 189, pp. 602-609, 2007.
- [16] 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.
- [17] J. M. Guti'erez and M. A. Hern'andez, "A family of chebyshev-halley type methods in banach spaces," *Bulletin of the Australian Mathematical Society*, vol. 55, pp. 113-130, 1997.
- [18] F. Shah, M. A. Noor, and M. Waseem, "Some second-derivative-free sixth-order convergent iterative methods for non-linear equations," *Maejo International Journal of Science and Technology*, vol. 10, no. 01, pp. 79-87, 2016.
- [19] A. M. Ostrowski, *Solution of Equations and Systems of Equations*, Academic Press, Cambridge, MA, USA, 2nd edition, 1966.
- [20] A. Kumar, P. Maroju, R. Behl, D. K. Gupta, and S. S. Motsa, "A family of higher order iterations free from second derivative for nonlinear equations in R," *Journal of Computational and Applied Mathematics*, vol. 330, pp. 215-224, 2018.
- [21] A. Golbabai and M. Javidi, "A third-order Newton type method for nonlinear equations based on modified homotopy perturbation method," *Applied Mathematics and Computation*, vol. 191, no. 1, pp. 199-205, 2007.
- [22] M. A. Noor, K. I. Noor, and K. Aftab, "Some new iterative methods for solving nonlinear equations," *World Applied Sciences Journal*, vol. 20, no. 6, pp. 870-874, 2012.
- [23] S. Weerakoon and T. G. I. Fernando, "A variant of Newton's method with accelerated third-order convergence," *Applied Mathematics Letters*, vol. 13, no. 8, pp. 87-93, 2000.
- [24] B. Kalantari, "Method of creating graphical works based on polynomials," *U.S. Patent*, vol. 6, pp. 894-705, 2005.
- [25] B. Kalantari, "Polynomiography: from the fundamental theorem of Algebra to art," *Leonardo*, vol. 38, no. 3, pp. 233-238, 2005.
- [26] B. Mandelbrot, *The Fractal Geometry of Nature*, W.H. Freeman and Company, New York, NY, USA, 1983.
- [27] B. Kalantari and E. H. Lee, "Newton-Ellipsoid polynomiography," *Journal of Mathematics and the Arts*, vol. 13, no. 4, pp. 336-352, 2019.
- [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] 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.