Received June 22, 2021, accepted July 6, 2021, date of publication July 9, 2021, date of current version July 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095941

Numerical Methods With Engineering Applications and Their Visual Analysis via Polynomiography

AMIR NASEEM [1], M. A. REHMAN [1], AND THABET ABDELJAWAD [2,3,4]
[1]Department of Mathematics, University of Management and Technology, Lahore 54770, 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 41354, Taiwan

Corresponding author: Thabet Abdeljawad (tabdeljawad@psu.edu.sa)

ABSTRACT Polynomiography is a fusion of Mathematics and Art, which as a software results in a new form of abstract art. Rendered images are through algorithmic visualization of solving a polynomial equation via iteration schemes. Images are beautiful and diverse, yet unique. In short, polynomiography allows us to draw unique and complex-patterned images of polynomials which be re-colored in different ways through different iteration schemes. In the modern age, polynomiography covers a variety of applications in different fields of art and science. The aim of this paper is to present polynomiography using newly constructed root-finding algorithms for the solution of non-linear equations. The constructed algorithms are two-step predictor-corrector methods. For reducing computational cost and making the algorithm more effective, we approximate the second derivative via interpolation technique. These methods have been derived by employing Househölder's method, interpolation technique and Taylor's series expansion. The convergence criterion of the newly developed algorithms has been discussed and proved their sixth-order convergence which is higher than many existing algorithms. To analyze the accuracy, validity and applicability of the proposed methods, several arbitrary and engineering problems have been tested and the obtained numerical results certify the better efficiency of the suggested methods against the other well-known iteration schemes given in the literature. Finally, we present polynomiography through the constructed iteration schemes and give a detailed comparison with the other iteration schemes which reflects the convergence properties and graphical aspects of the constructed algorithms.

INDEX TERMS Order of convergence, non-linear equations, Newton's method, Halley's method, polynomiography.

I. INTRODUCTION

Polynomiography deals with the algorithmic visualization of complex polynomial equations via different numerical algorithms [1], [12], [19]. It is actually both, the art and science associated with the visualization of the task of root-finding for a complex polynomial equation through fractal and non-fractal images fabricated by means of mathematical convergence properties of iteration functions which were being under consideration [15]. The term "polynomiography"was introduced to science in 21st century by Kalantari [15], that was the last thought-provoking contribution to the polynomials root-finding history. He further described polynomiography as a process that creates aesthetically pleasing and beautiful graphics, it was patented by Kalantari in the USA in 2005 [9], [16]. An individual image that is generated in the process of polynomiography, is known as a "polynomiograph". There exist a lot of techniques to create polynomiographs. Among them, some are fractal while others are non-fractal. The polynomiographs generated through Newton's iteration scheme result in familiar fractals. Fractal art is one of the most widely used types of mathematical art. There exist different kinds of fractals. One of them, the canonical type occurs as a result of numerical algorithms which are used for finding roots of polynomial equations, a problem that brought an evolution in the study Science, Mathematics,

The associate editor coordinating the review of this manuscript and approving it for publication was Gokhan Apaydin.

Art and Design, etc. Another type of fractals that is used excessively in arts is the complex fractals that are the fractals generated in a complex plane through different root-finding algorithms. The obtained patterns can be used for paintings, carpet designing, sculptures and animations, etc. For further study on fractals and iterations, one can see Mandelbrot [21], Devaney [5], Beardon [3], Gleick [8], Peitgen *et al.* [30], Milnor [23] and Kalantari [16].

One of the classic and quintessential properties which are exhibited by polynomiography is the basins of attraction of the polynomials' zeros. The basins of attraction are relied on the iteration schemes that were being used. The basins of attraction impart an accurate and deep understanding about the dynamical features of the considered algorithms over a variety of examples and the sets of the parameter values. For more study, one can see [31], [35] and references are therein. It can be utilized to study the convergence properties of different numerical algorithms.

There exist many numerical algorithms that have been used to generate polynomiographs. The classical and basic Newton's iteration scheme was proposed by Isaac Newton in 1669 and later by Raphson in 1690. In 1964, J.F. Traub proved that the Newton's method converges quadratically. With the passage of time, many mathematicians tried to modify Newton's iteration scheme for the sake of better accuracy and higher convergence order. Some basic and classical iteration schemes have been introduced in the literature [1], [4], [10], [12], [19], and the references are cited therein. With the help of the latest computer technology, multi-step iterative methods become more vital because the calculation of such types of methods is much easier with the help of computer programs than before. That is why, in recent years, many researchers try to modify Newton's method and proposed multi-step iterative methods which have a higher order of convergence with less number of functional evaluations.

In [29], [37], the authors constructed quartic-order iteration schemes with the insertion of Newton's algorithms that behaves as a predictor in the given schemes. With the passage of time, Noor *et al.* [28] modified the existing Halley's iterative scheme with sextic convergence order and then made it second-derivative and introduced a new iterative scheme with quintic-convergence order. By employing the finite-difference technique, the authors in [25] constructed a novel derivative free iterative scheme with quintic-convergence order. After that in 2018, Kumar *et al.* [20], presented a novel family of iterative schemes with sixth-order convergence for computing zeros of non-linear functions. In [36], the authors employed the technique of composition along with the rational interpolation and the basic idea of Padé approximation and then constructed some novel optimal fourth- and eighth-order derivative-free modified forms of King's method by. Wang and Zhu [38], in 2020, proposed two new iterative methods by using inverse interpolation technique, having convergence of order 4.5616 and 10.1311 respectively. In 2021, Naseem *et al.* [24] by applying variational iteration technique, introduced some novel ninth-order iteration schemes and then gave a detailed graphical representation through polynomiographs using different complex polynomials.

In this article, we establish and then analyze two new root-finding iteration schemes namely; Algorithm 1 and Algorithm 2, in which we consider Newton's iterative scheme as a predictor. To derive these methods, we employ Taylor's series, Househölder's method and the interpolation technique for making the suggested method derivative-free. The convergence criterion of the suggested algorithms has been discussed and established that the proposed algorithms are of the sextic convergence. To show the superiority of the proposed algorithms among the other compared methods, we test some arbitrary and real-world problems and the numerical results certify the validity of the proposed algorithms. Finally, we give a detailed graphical comparison of the constructed algorithms with different iteration schemes of the same category by means of polynomiographs for different complex polynomials that reveals the convergence speed and the other graphical aspects of the constructed algorithms.

The rest part of the paper is divided as follows. Some basic and classical methods are discussed in Section 2. In Section 3, we proposed two new root-finding algorithms for non-linear equations. The convergence criterion of the suggested schemes is discussed in Section 4. In Section 5, several test examples along with the engineering problems have been solved to show the performance, applicability and validity of the suggested iteration schemes. In Section 6, the polynomiography through the suggested iteration schemes have been presented by considering some complex polynomials. Finally, the conclusion of the paper is given in Section 7.

II. ITERATIVE METHODS

The mechanism of getting approximate roots of non-linear functions has been involved in many scientific fields such as video and audio processing, Mathematics, Fluid-mechanics and fuzzy systems, etc. The ability to find the roots of non-linear systems of equations is the most important task that has tremendous applications in fuzzy systems.

Let us consider the non-linear algebraic equation of the form:

$$\xi(s) = 0, \tag{1}$$

where $\xi : \mathbb{D} \subset \mathbb{R} \to \mathbb{R}$ is a scalar function defined on the open interval $\mathbb{D}$.

Assuming "α" as a simple zero of equation (1) and "ω" as a starting point that should be sufficiently close to "α". By applying Taylor's series around "ω" for equation (1), one can write:

$$\xi(\omega) + (s-\omega)\xi'(\omega) + \frac{1}{2!}(s-\omega)^2\xi''(\omega) + \ldots = 0. \tag{2}$$

If $\xi'(\omega) \neq 0$, then the above expression can be approximated up to the second term as:

$$\xi(s_k) + (s - s_k)\xi'(s_k) = 0.$$

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

$$s_{k+1} = s_k - \frac{\xi(s_k)}{\xi'(s_k)}. \tag{3}$$

This is well-known, Newton's iteration scheme [4], [19], for solving non-linear functions. From (2), one can evaluate

$$s_{k+1} = s_k - \frac{2\xi(s_k)\xi'(s_k)}{2\xi'^2(s_k) - \xi(s_k)\xi''(s_k)}, \tag{4}$$

which is Halley's iteration scheme [1], [10] for approximating the roots of non-linear functions in one dimension with third-order convergence. Simplification of (2) give rise to another iterative scheme:

$$s_{k+1} = s_k - \frac{\xi(s_k)}{\xi'(s_k)} - \frac{\xi^2(s_k)\xi''(s_k)}{2\xi'^3(s_k)}, \tag{5}$$

which is cubic-order Householder's iteration method [12] for computing zeros of non-linear scalar equations.

III. CONSTRUCTION OF NEW ALGORITHMS

Let $\xi : \mathbb{D} \to \mathbb{R}$, $\mathbb{D} \subset \mathbb{R}$ is a scalar function, where the domain $\mathbb{D}$ is an open interval in $\mathbb{R}$. By choosing s_0 as a starting point sufficiently close to the exact root α and implementing Taylor's series around the starting point s_0, we have:

$$\xi(s_0) + (s - s_0)\xi'(s_0) + \frac{1}{2!}(s - s_0)^2\xi''(s_0) = 0. \tag{6}$$

After simplification, one can obtain:

$$s = s_0 - \frac{\xi(s_0)}{\xi'(s_0)} - \frac{\xi^2(s_0)\xi''(s_0)}{2\xi'^3(s_0)}. \tag{7}$$

If we rewrite the above equation in generalized form, we obtained the well-known Householder's method given in (5). Again from equation (6):

$$s = s_0 - \frac{\xi(s_0)}{\xi'(s_0)} - \frac{(s - s_0)^2\xi''(s_0)}{2\xi'(s_0)}. \tag{8}$$

Now from Householder's method in equation (7):

$$s - s_0 = -\frac{\xi(s_0)}{\xi'(s_0)} - \frac{\xi^2(s_0)\xi''(s_0)}{2\xi'^3(s_0)}. \tag{9}$$

Using equation (9) in (8), we obtain:

$$s = s_0 - \frac{\xi(s_0)}{\xi'(s_0)} - \frac{\xi^2(s_0)\xi''(s_0)[2\xi'^2(s_0) + \xi(s_0)\xi''(s_0)]^2}{8\xi'^7(s_0)}. \tag{10}$$

Now taking the Newton's iteration scheme as a predictor and rewrite the above equality in iterative form, we obtain a new root-finding algorithm of the form:

Algorithm 1 Given a starting point s_0, the approximate zero s_{n+1} can be achieved by the iterative schemes given as:

$$t_n = s_n - \frac{\xi(s_n)}{\xi'(s_n)}, n = 0, 1, 2, \ldots,$$
$$s_{n+1} = t_n - \frac{\xi(t_n)}{\xi'(t_n)} - \frac{\xi^2(t_n)\xi''(t_n)[2\xi'^2(t_n) + \xi(t_n)\xi''(t_n)]^2}{8\xi'^7(s_n)}.$$

Which is the new modified form of Householder's method in which Newton's iterative scheme has been taken as a predictor. To obtain the approximate roots of the given non-linear scalar equations, one has to find the first and second derivatives of the given function ξ. But sometimes, we have to deal with such functions where the existence of the second derivative becomes impossible that may cause the failure of the proposed algorithm for finding the solution. To resolve the above-described issue, we approximate the second derivative by utilizing the idea of interpolation.

For this purpose, we consider the following function:

$$\eta(u) = a_1 + a_2(u - t_n) + a_3(u - t_n)^2 + a_4(u - t_n)^3,$$

where a_1, a_2, a_3, and a_4 are the unknowns that can be determined by means of the interpolation conditions given as:

$$\xi(s_n) = \eta(s_n), \xi(t_n) = \eta(t_n), \xi'(s_n) = \eta'(s_n),$$
$$\xi'(t_n) = \eta'(t_n), \xi''(t_n) = \eta''(t_n),$$

where s_n and t_n are the arbitrary points in the domain of the function ξ on which it is defined.

The above conditions providing us a system containing four linear equations with four variables. Solution of this system provides us the following equality:

$$\xi''(t_n) = \frac{6[\xi(s_n) - \xi(t_n)] - 2[s_n - t_n][2\xi'(t_n) + \xi'(s_n)]}{(s_n - t_n)^2} = \eta(s_n, t_n). \tag{11}$$

After substituting the value of $\xi''(t_n)$ from equation (11) in Algorithm 1, we obtain the following new algorithm which is second-derivative free.

Algorithm 2 Given a starting point s_0, the approximate zero s_{n+1} can be achieved by the iterative schemes given as:

$$t_n = s_n - \frac{\xi(s_n)}{\xi'(s_n)}, n = 0, 1, 2, \ldots,$$
$$s_{n+1} = t_n - \frac{\xi(t_n)}{\xi'(t_n)} - \frac{\xi^2(t_n)\eta(s_n, t_n)[2\xi'^2(t_n) + \xi(t_n)\eta(s_n, t_n)]^2}{8\xi'^7(s_n)}.$$

Which is a new second-derivative free algorithm in which the Newton's iterative scheme acts as a predictor. By applying this algorithm, one can easily determine the approximate solutions of those functions where second derivative fails to exist. Also, the above-described algorithm needs just two functional evaluations and only one of its derivatives that imparts the higher efficiency index in comparison with those algorithms which need the evaluation of the second derivatives.

IV. CONVERGENCE ANALYSIS

This section includes the convergence analysis of the constructed algorithms.

Theorem 1: Assuming $\alpha \in \mathbb{D}$ as a simple zero of the differentiable function $\xi : \mathbb{D} \subset \mathbb{R} \to \mathbb{R}$ where the domain $\mathbb{D}$ is an open interval in $\mathbb{R}$. If the initial guess s_0 is the neighborhood of the exact root α, then Algorithm 1 locally converges with the convergence of order six.

Proof: To show the sixth-order convergence of Algorithm 1, we suppose "e_n" be the error at nth step of iteration, where $e_n = s_n - \alpha$ and by applying the Taylor's series about $s = \alpha$, one can obtain the following equality

$$\xi(s_n) = \xi'(\alpha)e_n + \frac{1}{2!}\xi''(\alpha)e_n^2 + \frac{1}{3!}\xi'''(\alpha)e_n^3 + \frac{1}{4!}\xi^{(iv)}(\alpha)e_n^4 + \frac{1}{5!}\xi^{(v)}(\alpha)e_n^5 + \frac{1}{6!}\xi^{(vi)}(\alpha)e_n^6 + O(e_n^7),$$

$$\xi(s_n) = \xi'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + O(e_n^7)], \quad (12)$$

$$\xi'(s_n) = \xi'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + 7c_7e_n^6 + O(e_n^7)], \quad (13)$$

where

$$c_n = \frac{1}{n!}\frac{\xi^{(n)}(\alpha)}{\xi'(\alpha)}, n = 2, 3, 4, \ldots$$

With the help of equations (12) and (13), we get

$$t_n = \xi'(\alpha)[\alpha + c_2e_n^2 + (2c_3 - 2c_2^2)e_n^3 + (3c_4 - 7c_2c_3 + 4c_2^3)e_n^4 + (-6c_3^2 + 20c_3c_2^2 - 10c_2c_4 + 4c_5 - 8c_2^4)e_n^5 + (-17c_4c_3 + 28c_4c_2^2 - 13c_2c_5 + 5c_6 + 33c_2c_3^2 - 52c_3c_2^3 + 16c_2^5)e_n^6 + O(e_n^7)], \quad (14)$$

$$\xi(t_n) = \xi'(\alpha)[c_2e_n^2 + (2c_3 - 2c_2^2)e_n^3 + (5c_2^3 - 7c_2c_3 + 3c_4)e_n^4 + (24c_3c_2^2 - 12c_2^4 - 10c_2c_4 + 4c_5 - 6c_3^2)e_n^5 + (-73c_3c_2^3 + 34c_4c_2^2 + 28c_2^5 + 37c_2c_3^2 - 17c_4c_3 - 13c_2c_5 + 5c_6)e_n^6 + O(e_n^7)], \quad (15)$$

$$\xi'(t_n) = \xi'(\alpha)[1 + 2c_2^2e_n^2 + (4c_2c_3 - 4c_2^3)e_n^3 + (6c_2c_4 - 11c_3c_2^2 + 8c_2^4)e_n^4 + 28c_3c_2^3 - 20c_4c_2^2 + 8c_2c_5 - 16c_2^5)e_n^5 + (-16c_4c_2c_3 - 68c_3c_2^4 + 12c_3^3 + 60c_4c_2^3 - 26c_5c_2^2 + 10c_2c_6 + 32c_2^6)e_n^6 + O(e_n^7)], \quad (16)$$

$$\xi''(t_n) = \xi'(\alpha)[2c_2 + 6c_2c_3e_n^2 + (12c_3^2 - 12c_3c_2^2)e_n^3 + (-42c_2c_3^2 + 18c_4c_3 + 24c_3c_2^3 + 12c_4c_2^2)e_n^4 + (-12c_2c_4c_3 + 24c_5c_3 - 36c_3^3 + 120c_3^2c_2^2 - 48c_3c_2^4 - 48c_4c_2^3)e_n^5 + (-78c_3c_2c_5 + 30c_3c_6 - 54c_4c_3^2 - 96c_3c_4c_2^2 + 198c_2c_3^3 - 312c_3^2c_2^3 + 96c_3c_2^5 + 72c_2c_4^2 + 144c_4c_2^4 + 20c_5c_2^3)e_n^6 + O(e_n^8)]. \quad (17)$$

Using equations (14) − (17) in Algorithm 1, we get

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

which implies that

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

The above equality justified the sixth-order convergence of Algorithm 1. □

Theorem 2: Assuming $\alpha \in \mathbb{D}$ as a simple zero of the differentiable function $\xi : \mathbb{D} \subset \mathbb{R} \to \mathbb{R}$ where the domain $\mathbb{D}$ is an open interval in $\mathbb{R}$. If the initial guess s_0 is the neighborhood of the exact root α, then Algorithm 2 locally converges with the convergence of order six.

Proof: From equations (12) − (16) together with the similar assumptions as in the previous theorem, we obtain:

$$D(s_n, t_n) = \xi'(\alpha)[2c_2 + (6c_2c_3 - 2c_4)e_n^2 + (12c_3^2 - 12c_3c_2^2 + 4c_2c_4 - 4c_5)e_n^3 + (2c_2c_5 + 26c_3c_4 - 42c_2c_3^2 + 24c_3c_2^3 + 2c_4c_2^2 - 6c_6)e_n^4 + (-48c_4c_2c_3 + 12c_4^2 - 24c_4c_2^3 + 28c_5c_3 + 4c_5c_2^2 + 120c_3^2c_2^2 - 48c_3c_2^4 - 8c_7 - 36c_3^3)e_n^5 + (-60c_5c_2c_3 + 28c_4c_3c_2^2 + 2c_2c_7 + 22c_5c_4 - 10c_5c_2^3 + 30c_6c_3 + 6c_6c_2^2 + 20c_2c_4^2 - 86c_4c_3^2 + 88c_4c_2^4 + 198c_2c_3^3 - 312c_3^2c_2^3 + 96c_3c_2^5 - 10c_8)e_n^6 + O(e_n^7)]. \quad (18)$$

Using equations (14)−(16) and (18) in Algorithm 2, we get

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

which implies that

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

The above equality justified the sixth-order convergence of Algorithm 2. □

V. NUMERICAL COMPARISONS AND ENGINEERING APPLICATIONS

In this section, we illustrate the accuracy, validity, applicability and performance of the constructed algorithms by considering several test functions which include some real-life problems in Engineering also. We compare the numerical results of these suggested algorithms with the following iteration schemes:

A. NOOR's ITERATION METHOD ONE (NIM1)

Given a starting point s_0, the approximate zero s_{n+1} can be achieved by the following iterative schemes:

$$t_n = s_n - \frac{\xi(s_n)}{\xi'(s_n)}, \quad n = 0, 1, 2, 3, \ldots,$$

$$s_{n+1} = t_n - \frac{\xi(s_n)}{\xi'(s_n)} + \left[\frac{\xi(s_n)}{\xi'(s_n)}\right]\frac{\xi'(t_n)}{\xi'(s_n)}.$$

Which is Noor's iteration method one [27] with convergence of second-order for computing roots of non-linear equations.

B. NOOR's ITERATION METHOD ONE (NIM2)

Given a starting point s_0, the approximate zero s_{n+1} can be achieved by the following iterative schemes:

$$t_n = s_n - \frac{\xi(s_n)}{\xi'(s_n)}, \quad n = 0, 1, 2, 3, \ldots,$$
$$s_{n+1} = s_n - \frac{2\xi(s_n)}{\xi'(s_n) + \xi'(t_n)}.$$

Which is Noor's iteration method two [27] with third-order convergence for computing roots of non-linear equations.

C. OSTROWSKI's ITERATION METHOD (OIM)

Given a starting point s_0, the approximate zero s_{n+1} can be achieved by the following iterative schemes:

$$t_n = s_n - \frac{\xi(s_n)}{\xi'(s_n)}, \quad n = 0, 1, 2, 3, \ldots,$$
$$s_{n+1} = t_n - \frac{\xi(s_n)\xi(t_n)}{\xi(s_n)\xi'(s_n) - 2\xi'(s_n)\xi(t_n)}.$$

Which is well-known Ostrowski's iteration method [29] with fourth-order convergence for computing roots of non-linear equations.

D. TRAUB's ITERATION METHOD (TIM)

Given a starting point s_0, the approximate zero s_{n+1} can be achieved by the following iterative schemes:

$$t_n = s_n - \frac{\xi(s_n)}{\xi'(s_n)}, \quad n = 0, 1, 2, 3, \ldots,$$
$$s_{n+1} = t_n - \frac{\xi(t_n)}{\xi'(t_n)}.$$

Which is known as Traub's iteration method [37] with fourth-order convergence for computing roots of non-linear equations.

E. MODIFIED HALLEY's ITERATION METHOD (MHIM)

Given a starting point s_0, the approximate zero s_{n+1} can be achieved by the following iterative schemes:

$$t_n = s_n - \frac{\xi(s_n)}{\xi'(s_n)}, \quad n = 0, 1, 2, 3, \ldots,$$
$$s_{n+1} = t_n - \frac{2\xi(s_n)\xi(t_n)\xi'(t_n)}{2\xi(s_n)\xi'^2(t_n) - \xi'^2(s_n)\xi(t_n) + \xi'(s_n)\xi'(t_n)\xi(t_n)}$$

Which is modified Halley's iteration method [28] for computing roots of non-linear equations, having the convergence of order five.

For comparing the above-described iteration schemes with the suggested methods, we consider eleven arbitrary and engineering examples and solve them.

In all examples, we choose the accuracy $\varepsilon = 10^{-15}$ for the stopping criterion of the computer programs $|s_{n+1} - s_n| < \varepsilon$.

Tables 1-2 illustrate the comparisons of the constructed algorithms with the other similar existing methods. In the columns of the given tables, the iterations consumed by different iteration schemes has been represented by the symbol N, the absolute value of the function has been denoted by $|\xi(s)|$, the approximated zero of the given function has been shown by s_{n+1}, the absolute distance of consecutive approximations through different iteration schemes has been represented by $|s_{n+1} - s_n|$ and the symbol COC stands for the computational order of convergence with the following standard form:

$$COC \approx \frac{ln\frac{|(s_{n+1}-\alpha)|}{|(s_n-\alpha)|}}{ln\frac{|(s_n-\alpha)|}{|(s_{n-1}-\alpha)|}}.$$

which was introduced in 2000 by Weerakoon *et al.* [40].

Example 1: Open Channel Flow Problem

The flow of water under the uniform flow condition is illustrated by the well-known Manning's equation [22]:

$$Water\ Flow = F = \frac{\sqrt{m}ar^{\frac{2}{3}}}{n}, \tag{19}$$

where the symbols a, m, r and n stand for the area, slope, hydraulic radius and the Manning's roughness coefficient respectively. For a channel, having rectangular shape with depth s and width b has the following equations:

$$a = bs, \quad \& \quad r = \frac{bs}{b+2s}. \tag{20}$$

With the help of (19) and (20), we can write:

$$F = \frac{\sqrt{m}bs}{n}\left(\frac{bs}{b+2s}\right)^{\frac{2}{3}}.$$

To find the depth of water in the channel for a given quantity of water, the above equation may written in the form of non-linear function as:

$$\xi_1(s) = \frac{\sqrt{m}bs}{n}\left(\frac{bs}{b+2s}\right)^{\frac{2}{3}} - F.$$

We select the specific values of different parameters appeared in the above equation as $m = 0.017$, $b = 4.572m$, $n = 0.0015$ *and* $F = 14.15\ m^3/s$. *For initialing the process of iterations, we take the starting point* $s_0 = 0.4$ *and the numerical results have been recorded in Tab. 1.*

Example 2: Adiabatic Flame Temperature Equation

Regarding the equation of the adiabatic flame temperature, we consider the following equality:

$$\xi_2(s) = \Delta H + c_1(s - 298) + \frac{c_2}{2}\left(s^2 - 298^2\right) + \frac{c_3}{3}\left(s^3 - 298^3\right),$$

where $\Delta H = -57798, c_1 = 7.256, c_2 = 0.002298, c_3 = 0.00000283$. *For more study, one can see [32], [33]. The non-linear function defined above is a cubic polynomial with three roots. One of the simple root among these three roots is* $\alpha = 4305.3099136661$ *which has been approximated by taking the starting point* $s_0 = 2050.0$ *and the numerical results have been recorded in Tab. 1.*

Example 3: Fraction Conversion of Nitrogen-Hydrogen to Ammonia

This problem has been taken from [2], that illustrates the fraction of the conversion of Nitrogen-Hydrogen feed *to Ammonia. In the present problem, we choose the values of pressure as 250 atm and that of temperature as* 500^0C*. The problem in terms of non-linear function has following form:*

$$-0.186 - \frac{8s^2\,(s-4)^2}{9\,(s-2)^3} = 0. \tag{21}$$

After simplifying (21), we can obtain the following polynomial:

$$\xi_3(s) = s^4 - 7.79075s^3 + 14.7445s^2 + 2.511s - 1.674.$$

It should be noted that polynomial given in the above equation is of degree four and in the light of fundamental theorem of Algebra, the roots of the polynomial must be four. As the numerical value of fraction conversion falls between the interval $(0, 1)$*, therefore the one and only feasible solution that falls in the* $(0, 1)$ *interval is 0.2777595428. The remanding roots of the above polynomial is meaningless physically. For initialing the process of iterations, we take the starting point* $s_0 = 0.1$ *and the numerical results have been recorded in Tab. 1.*

Example 4: Finding Volume from Van Der Wall's Equation

The well-known van der Waal's equation [39] in the Chemical Engineering and Chemistry has a significant importance and it is used for studying the behavior of gases. The standard mathematical form of this equation is:

$$(P + \frac{A_1 n^2}{V^2})\,(V - nA_2) = nRT. \tag{22}$$

By choosing the feasible values of the different parameters that appeared in (22), the following non-linear function can be obtained:

$$\xi_4(s) = 0.986s^3 - 5.181s^2 + 9.067s - 5.289,$$

where the variable s in (22) stands for the volume of the gas and can be calculated by computing the zeros of ξ_4*. As the function* ξ_4 *is a cubic degree polynomial, so there exist three roots. The positive real and simple root among the three roots is* 1.9298462428*. The remanding two roots of* ξ_4 *is meaningless physically because volume can never be negative. Therefore, for initialing the process of iterations, we take the starting point* $s_0 = 2.0$ *and the numerical results have been recorded in Tab. 1.*

Example 5: Transcendental and Algebraic Problems

To numerically analyze the suggested algorithms, we consider the following seven transcendental and algebraic equations and their numerical results can be seen in Tab. 2.

$$\begin{aligned}
\xi_5(s) &= se^{s^2} - sin^2(s) - 3\cos(s) + 5,\\
\xi_6(s) &= \cos(s) - s,\\
\xi_7(s) &= s^3 + 4s^2 - 15,\\
\xi_8(s) &= (s-1)^3 - 1,\\
\xi_9(s) &= 2 - 2\sqrt{s},\\
\xi_{10}(s) &= e^{(s^2-7s-30)} - 1,\\
\xi_{11}(s) &= s^3 - 10.
\end{aligned}$$

TABLE 1. Numerical comparison among different algorithms for the engineering problems $\xi_1 - \xi_4$.

Method	N	$\|\xi(s_{n+1})\|$	s_{n+1}	$\sigma = \|s_{n+1} - s_n\|$	COC
$\xi_1(s)$, $s_0 = 0.4$					
NIM1	6	$2.230770e-24$	1.46509122029582464237602074553413096	$1.280653e-12$	2
NIM2	3	$5.751429e-27$	1.46509122029582464237602091020202537	$4.220179e-09$	3
OIM	3	$9.825352e-31$	1.46509122029582464237602090977863844	$6.464833e-08$	4
TIM	3	$7.765624e-44$	1.46509122029582464237602090977856610	$4.884637e-11$	4
MHIM	3	$4.020841e-64$	1.46509122029582464237602090977856610	$5.749371e-13$	5
Algorithm 1	2	$2.716057e-20$	1.46509122029582464237802065566113214	$1.999354e-03$	6
Algorithm 2	2	$1.712361e-15$	1.46509122029582476845167427654564868	$9.940002e-03$	6
$\xi_2(s)$, $s_0 = 2050.0$					
NIM1	9	$3.688522e-28$	4305.30991366612556304019892944632561606	$3.947209e-13$	2
NIM2	4	$2.919985e-37$	4305.30991366612556304019892944634208621	$9.938805e-11$	3
OIM	3	$2.865067e-36$	4305.30991366612556304019892944634208621	$6.456917e-07$	4
TIM	3	$6.063382e-31$	4305.30991366612556304019892944634209814	$1.002459e-05$	4
MHIM	3	$1.311971e-69$	4305.30991366612556304019892944634208621	$1.526816e-11$	5
Algorithm 1	2	$5.757982e-21$	4305.30991366612556304019867233849842696	$6.102740e-01$	6
Algorithm 2	2	$5.757982e-21$	4305.30991366612556304019867233849842696	$6.102740e-01$	6
$\xi_3(s)$, $s_0 = 0.1$					
NIM1	7	$9.675391e-26$	0.27775954284172065909591015386800788	$1.053628e-13$	2
NIM2	3	$1.203488e-18$	0.27775954284172065896195690011981922	$6.173552e-07$	3
OIM	3	$3.158175e-34$	0.27775954284172065909591016463712051	$2.153605e-09$	4
TIM	3	$3.260304e-39$	0.27775954284172065909591016463712048	$1.412011e-10$	4
MHIM	2	$2.057683e-15$	0.27775954284172043006718791209749563	$1.970771e-04$	5
Algorithm 1	2	$1.552395e-21$	0.27775954284172065909608295267204378	$2.518612e-04$	6
Algorithm 2	2	$3.463037e-21$	0.27775954284172065909629561501923914	$2.811039e-04$	6
$\xi_4(s)$, $s_0 = 2.0$					
NIM1	4	$1.319023e-19$	1.92984624284786221696144418547353349	$5.000588e-10$	2
NIM2	3	$3.958485e-15$	1.92984624284790801736694582212524590	$1.021691e-05$	3
OIM	3	$2.230713e-36$	1.92984624284786221848752742786545651	$6.360563e-10$	4
TIM	3	$8.395139e-34$	1.92984624284786221848752742786546620	$2.556739e-09$	4
MHIM	2	$8.089146e-19$	1.92984624284786222784650752141958847	$1.079121e-04$	5
Algorithm 1	2	$1.241265e-36$	1.92984624284786221848752742786545647	$4.206257e-07$	6
Algorithm 2	2	$1.241265e-36$	1.92984624284786221848752742786545647	$4.206257e-07$	6

TABLE 2. Numerical comparison among different algorithms for transcendental and algebraic problems $\xi_5 - \xi_{11}$.

Method	N	$\lvert\xi(s_{n+1})\rvert$	s_{n+1}	$\sigma = \lvert s_{n+1} - s_n\rvert$	COC
$\xi_5(s)$, $s_0 = -2.1$					
NIM1	7	$6.729343e-16$	$-0.99267257254612149742805753089548592$	$7.489719e-09$	2
NIM2	7	$1.616064e-36$	$-0.99267257254612160321469740374314232$	$3.740481e-13$	3
OIM	5	$2.499485e-55$	$-0.99267257254612160321469740374314232$	$1.213717e-14$	4
TIM	5	$7.882515e-25$	$-0.99267257254612160321469752765787680$	$3.686855e-07$	4
MHIM	17	$7.206813e-36$	$-0.99267257254612160321469740374314232$	$3.824335e-08$	5
Algorithm 1	3	$1.688991e-19$	$-0.99267257254612160318814611645678326$	$3.393265e-04$	6
Algorithm 2	4	$8.427525e-53$	$-0.99267257254612160321469740374314232$	$1.093813e-09$	6
$\xi_6(s)$, $s_0 = 2.5$					
NIM1	6	$4.872692e-21$	$0.73908513321516064165240060537232352$	$1.148292e-10$	2
NIM2	4	$3.229578e-29$	$0.73908513321516064165531208765457635$	$1.082465e-09$	3
OIM	3	$2.576412e-26$	$0.73908513321516064165531210306819383$	$8.807980e-07$	4
TIM	3	$2.617193e-35$	$0.73908513321516064165531208767387342$	$6.173593e-09$	4
MHIM	3	$1.154792e-54$	$0.73908513321516064165531208767387340$	$4.262132e-11$	5
Algorithm 1	2	$1.193476e-25$	$0.73908513321516064165531215898526674$	$2.149671e-04$	6
Algorithm 2	2	$7.499328e-20$	$0.73908513321516064165531215898526674$	$2.518029e-03$	6
$\xi_7(s)$, $s_0 = -0.3$					
NIM1	64	$6.025940e-28$	$1.63198080556606351752210644551262427$	$8.230315e-15$	2
NIM2	55	$2.579933e-29$	$1.63198080556606351752210644554003074$	$1.822740e-10$	3
OIM	6	$1.173790e-16$	$1.63198080556606352309938541240166940$	$1.001538e-04$	4
TIM	19	$1.688878e-22$	$1.63198080556606351752211447026698691$	$3.210629e-06$	4
MHIM	16	$3.742527e-16$	$1.63198080556606353530477171002443119$	$1.068510e-03$	5
Algorithm 1	5	$8.374661e-71$	$1.63198080556606351752210644554125660$	$3.217234e-12$	6
Algorithm 2	5	$8.374661e-71$	$1.63198080556606351752210644554125660$	$3.217234e-12$	6
$\xi_8(s)$, $s_0 = 1.1$					
NIM1	36	$1.326921e-22$	$1.99999999999999999999995576931214342$	$6.650616e-12$	2
NIM2	64	$1.823488e-22$	$1.99999999999999999999993921706136734$	$3.734894e-08$	3
OIM	7	$2.232058e-35$	$2.00000000000000000000000000000000001$	$1.827759e-09$	4
TIM	7	$6.330643e-16$	$2.00000000000000002110214257980683044 6$	$1.205359e-04$	4
MHIM	6	$1.767889e-71$	$2.00000000000000000000000000000000000$	$6.520377e-15$	5
Algorithm 1	5	$2.769456e-19$	$1.99999999999999999990768479440158025$	$8.077930e-04$	6
Algorithm 2	5	$2.769456e-19$	$1.99999999999999999990768479440158025$	$8.077930e-04$	6
$\xi_9(s)$, $s_0 = 0.3$					
NIM1	4	$5.919022e-20$	$1.00000000000000000005919022143227444$	$4.865808e-10$	2
NIM2	4	$5.324651e-41$	$1.00000000000000000000000000000000000$	$7.524201e-14$	3
OIM	3	$3.278363e-36$	$1.00000000000000000000000000000000000$	$3.805917e-09$	4
TIM	3	$1.188143e-40$	$1.00000000000000000000000000000000000$	$2.952991e-10$	4
MHIM	3	$7.811149e-58$	$1.00000000000000000000000000000000000$	$9.220760e-12$	5
Algorithm 1	2	$1.889098e-20$	$1.00000000000000000001889097872707259$	$1.460200e-03$	6
Algorithm 2	2	$2.626837e-17$	$0.99999999999999997373162948618765540$	$4.550437e-03$	6
$\xi_{10}(s)$, $s_0 = 3.5$					
NIM1	8	$5.443124e-18$	$2.99999999999999999958129814273684805$	$2.523138e-10$	2
NIM2	8	$5.290016e-24$	$3.00000000000000000000000040692430863$	$1.916154e-09$	3
OIM	5	$9.011139e-16$	$3.00000000000000006931645429656286350$	$2.940997e-05$	4
TIM	6	$5.475744e-24$	$3.00000000000000000000000042121106214$	$1.961589e-07$	4
MHIM	5	$4.477869e-21$	$2.99999999999999999999965554856577832$	$1.127284e-05$	5
Algorithm 1	4	$3.491976e-19$	$2.99999999999999999997313864573539114$	$1.215950e-04$	6
Algorithm 2	4	$2.008837e-17$	$2.99999999999999999845474113487771229$	$2.693702e-04$	6
$\xi_{11}(s)$, $s_0 = 0.7$					
NIM1	13	$2.379349e-16$	$2.15443469003188370467211713039225237$	$6.067389e-09$	2
NIM2	7	$1.061994e-24$	$2.15443469003188372175929349025276116$	$6.719723e-09$	3
OIM	4	$2.348829e-19$	$2.15443469003188372177616156303777797$	$2.242804e-05$	4
TIM	5	$4.822490e-49$	$2.15443469003188372175929356651935050$	$7.671333e-13$	4
MHIM	4	$4.845049e-61$	$2.15443469003188372175929356651935050$	$1.084363e-12$	5
Algorithm 1	3	$3.460511e-17$	$2.15443469003188371927414543737272075$	$2.651841e-03$	6
Algorithm 2	3	$3.460511e-17$	$2.15443469003188371927414543737272075$	$2.651841e-03$	6

By carefully examining the results recorded in Tabs. 1–2, it is clear that the constructed iteration schemes are superior to the other comparable ones with respect to different parameters such as convergence speed, iterations' consumption and the accuracy.

Table 3 has been designed by increasing the accuracy up to 100 decimal places in the stopping criterion which gives us the detailed comparison in terms of iterations that have been consumed by different iteration schemes with the constructed algorithms. The vertical columns of Tab. 3 give us the information about the iterations' consumption by different iteration schemes for different non-linear functions together with the starting point s_0.

The results of Tab. 3, again proved that the constructed algorithms consumed less iterations when the accuracy has been increased as compared to the other iteration schemes. All the numerical computations have been carried out through the computer program Maple 15.

TABLE 3. Comparison of the iterations consumed by different algorithms for $\varepsilon = 10^{-100}$.

Function With Initial Guess	NIM1	NIM2	OIM	TIM	MHIM	Algorithm 1	Algorithm 2
$\xi_1(s)$, $s_0 = 2050.0$	09	05	04	04	03	03	03
$\xi_2(s)$, $s_0 = 0.1$	09	06	06	06	06	05	03
$\xi_3(s)$, $s_0 = 2.0$	07	05	04	04	03	03	03
$\xi_4(s)$, $s_0 = 0.4$	08	05	04	05	03	04	03
$\xi_5(s)$, $s_0 = -2.1$	10	08	06	07	18	04	05
$\xi_6(s)$, $s_0 = 2.5$	09	06	04	04	04	03	03
$\xi_7(s)$, $s_0 = -0.3$	66	57	08	21	18	06	06
$\xi_8(s)$, $s_0 = 1.1$	39	66	08	09	07	06	06
$\xi_9(s)$, $s_0 = 0.3$	07	05	04	04	04	03	03
$\xi_{10}(s)$, $s_0 = 3.5$	11	10	07	08	06	05	05
$\xi_{11}(s)$, $s_0 = 0.7$	16	09	06	06	05	04	04

VI. POLYNOMIOGRAPHY

To generate polynomiographs through the computer program for different complex polynomials, we consider an initial rectangle R that contains the zeros of the polynomial which is being under consideration. Corresponding to the starting point z_0 in the region of R, we execute the process of iteration and color the point corresponding to z_0 that relies on approximated convergence of truncated orbit to the root. The discretization of the rectangle R is directly related to the quality and resolution of the drawn image i.e., if we made the discretization of the rectangle R into the grid of 2000×2000, then the resultant image will be of high quality with the better resolution.

We know that if there exists a polynomial p with degree n then it must possess exactly n roots by the fundamental theorem of Algebra and can be written uniquely as :

$$p(z) = c_n z^n + c_{n-1} z^{n-1} + \ldots + c_1 z + c_0, \tag{23}$$

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

$$p(z) = (z - z_1)(z - z_2) \ldots (z - z_n), \tag{24}$$

where $\{c_n, c_{n-1}, \ldots, c_1, c_0\}$ are the complex coefficients.

The iterative algorithms can be easily applied to both representations of the complex polynomial p. The number of basins of attraction of the considered polynomial is depicted by its degree. The basins' location can easily be managed by changing the position of roots in the complex plane manually. The polynomiographs' colors are directly associated with the iterations' consumption to attain the approximate solution of the polynomial with the defined accuracy and the considered scheme of iteration.

The most general and the basic algorithm for drawing polynomiographs has been given in Algorithm 3.

The iteration scheme is supposed to be converged or diverged if the Convergence Test $(z_n + 1, z_n, \epsilon)$ returns TRUE or False in the above-defined Algorithm 3. The standard convergence test which is used to study the convergence or divergence of the iteration scheme has the following form:

$$|z_{n+1} - z_n| < \varepsilon, \tag{25}$$

where z_{n+1} and z_n are the two successive approximations in the process of iteration and $\varepsilon > 0$ represents the defined

Algorithm 3 Polynomiograph's Generation

```
Input: p ∈ ℂ — A complex polynomial, 𝔸 ⊂ ℂ — The
       area, k — Maximum number of iterations, I —
       Iteration scheme, ϵ — The accuracy, Colormap
       [0...C − 1] — Colormap with C colors.
Output: Polynomiograph for the complex polynomial p
        in area 𝔸.
for z0 ∈ 𝔸 do
    i = 0
    while i ≤ k do
        z_{n+1} = I(z_n)
        if |z_{n+1} − z_n| < ϵ then
            break
        i = i + 1
    color z0 through the colormap.
```

accuracy. In the present article, we consider the same stopping criterion as given in (25). Using the constructed novel algorithms and other methods of the same kind, we obtained nice-looking and interesting polynomiographs. The different coloring of polynomiographs relies on the number of iterations requires to approximate the roots of the polynomial with defined accuracy ε. A plethora of such types of images can be drawn through the computer program by the variation of k which denotes the upper bound of iterations consumed by the iteration scheme. For further study in the field of polynomiography along with the artistic applications, one can see [6], [7], [9], [11], [13], [16]–[18], [26], [34] and the references therein.

For drawing polynomiographs and comparing them with the other iteration schemes, we consider six different complex polynomials:

$$p_1(z) = z^3 - 1,\ p_2(z) = (z^3 - 1)^2,\ p_3(z) = z^4 - 1,$$
$$p_4(z) = (z^4 - 1)^2,\ p_5(z) = z^5 - 1,\ p_6(z) = (z^5 - 1)^2.$$

For drawing aesthetically pleasing images, we utilize the computer technology with the program Mathematica 12.0 for creating all the presented images by taking the accuracy $\varepsilon = 0.01$, the area $A = [-2, 2] \times [-2, 2]$, and the upper bound of the number of iterations $k = 15$. In order to color the iterations in the process of polynomiographs' generation, we utilize colormap presented in the Fig. 1.

FIGURE 1. The colormap for drawing polynomiographs.

Example 6: Polynomiographs for the Polynomial p_1 Through Various Numerical Algorithms

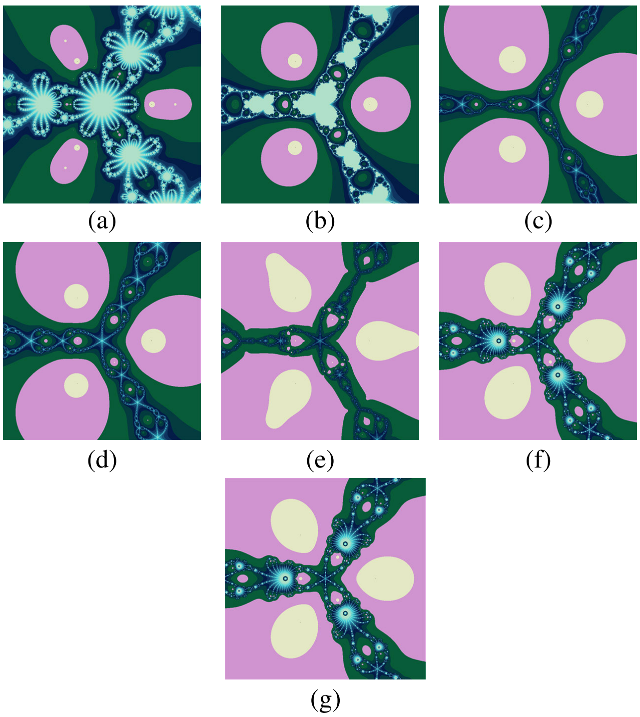

FIGURE 2. Polynomiographs associated with the polynomial p_1. (a) stands for NIM1, (b) for NIM2, (c) for OIM, (d) for TIM, (e) for MHIM, (f) for Algorithm 1 and (g) for Algorithm 2.

Example 7: Polynomiographs for the Polynomial p_2 Through Various Numerical Algorithms

Example 8: Polynomiographs for the Polynomial p_3 Through Various Numerical Algorithms

Example 9: Polynomiographs for the Polynomial p_4 Through Various Numerical Algorithms

Example 10: Polynomiographs for the Polynomial p_5 Through Various Numerical Algorithms

Example 11: Polynomiographs for the Polynomial p_6 Through Various Numerical Algorithms

In Examples 6–11, containing Figures 2–7, the convergence regions for Noor's iteration method one (NIM1), Noor's iteration method two (NIM2), Ostrowski's iteration method (OIM), Traub's iteration method (TIM), the modified Halley's iteration method (MIHM) and the newly constructed algorithms have been shown using different degrees complex polynomials.

In the first experiment, we executed all comparable iteration schemes for the purpose of obtaining the simple roots of p_1 which is actually a cubic-degree polynomial and their results can be seen in Fig. 2. In the next experiment, we take the polynomial p_2 which is a six-degree polynomial with three distinct zeros of multiplicity 2. The corresponding polynomiographs are given in Fig. 3. In the eighth and ninth examples, we again executed the comparable iteration schemes for p_3 and p_4 which are fourth- and eight-degree polynomials and their drawn images have been given in Figs. 4 and 5. Both p_3 and p_4 have four distinct roots but the latter one possesses zeros of multiplicity 2. The drawn images for p_5 and p_6 which are fifth- and tenth-degree polynomials through the comparable iteration schemes in the form of polynomiographs can be seen in Figs. 6 and 7. These polynomials possess five different zeros but p_6 possesses non-simple zeros with multiplicity 2. All zeros or roots of the considered polynomials $p_1 - p_6$ are displayed in the form of white circles in Figs. 2-7. The repeated zeros or roots of the polynomials possess identical colors and the location on the corresponding polynomiographs which can be seen in Figs. 3, 5 and 7. Moreover, the drawn images through the suggested iteration schemes have interesting and diverse fractal patterns with higher dynamics.

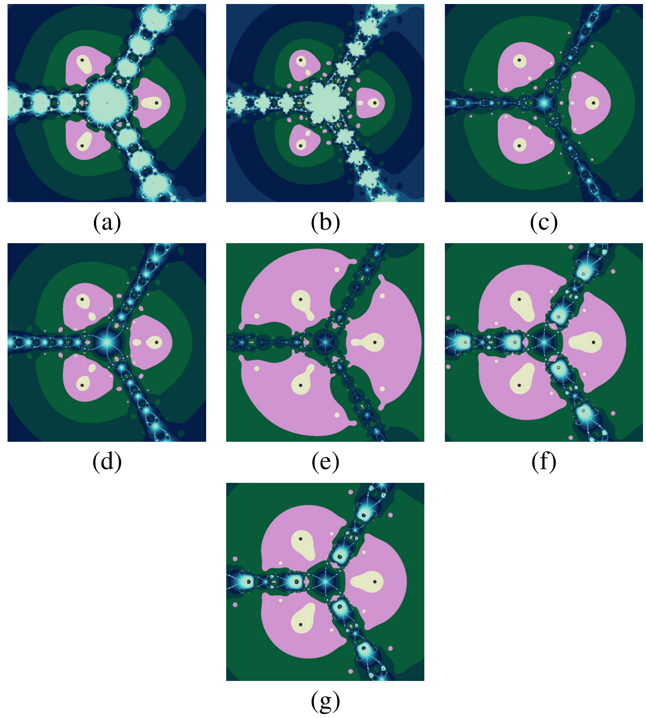

FIGURE 3. Polynomiographs associated with the polynomial p_2. (a) stands for NIM1, (b) for NIM2, (c) for OIM, (d) for TIM, (e) for MHIM, (f) for Algorithm 1 and (g) for Algorithm 2.

Two important features of the considered algorithms can be predicted by examining the generated polynomiographs carefully. The first feature among them is the rate or convergence speed of the iteration, i.e. each point's color informs us about the iterations carried out by the considered iteration scheme to approximate the zero. The second feature is the dynamics of the iteration scheme that is used to draw polynomiographs. Low dynamics of the drawn images

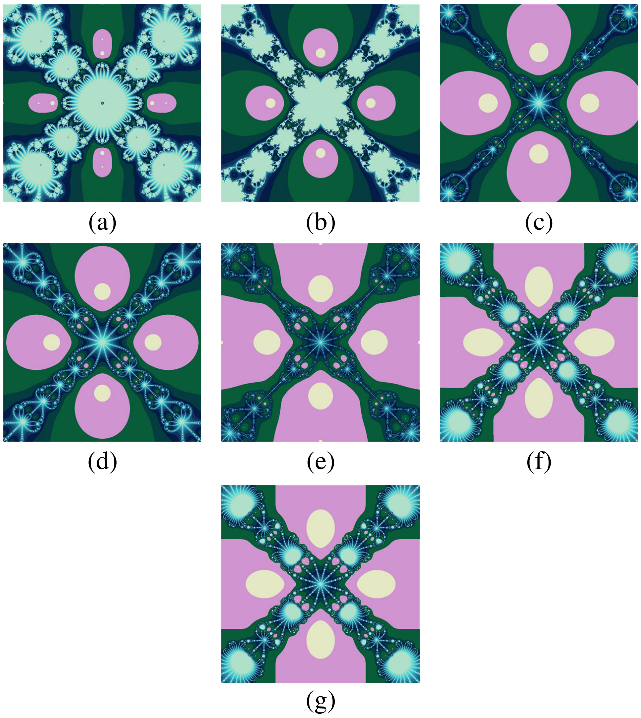

FIGURE 4. Polynomiographs associated with the polynomial p_3. (a) stands for NIM1, (b) for NIM2, (c) for OIM, (d) for TIM, (e) for MHIM, (f) for Algorithm 1 and (g) for Algorithm 2.

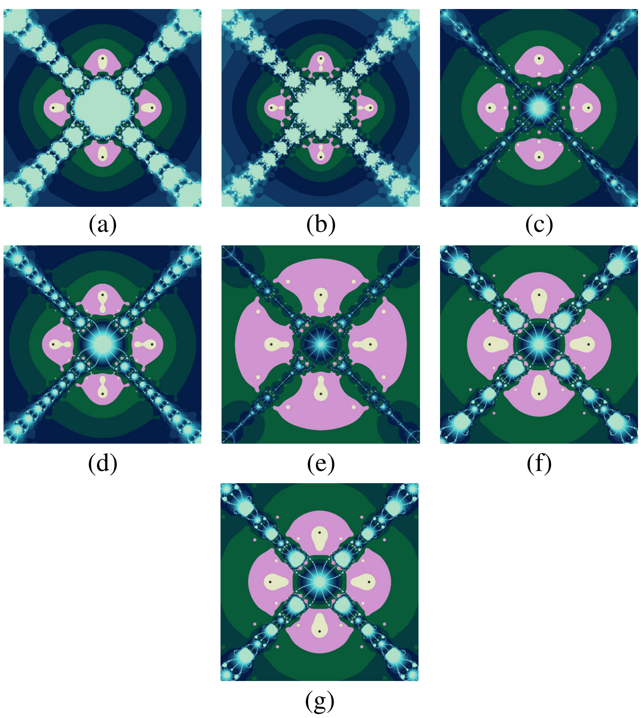

FIGURE 5. Polynomiographs associated with the polynomial p_4. (a) stands for NIM1, (b) for NIM2, (c) for OIM, (d) for TIM, (e) for MHIM, (f) for Algorithm 1 and (g) for Algorithm 2.

exist in the areas, having small colors' variation and the areas with large colors' variation, the dynamics are high. The brighter areas in figures showing the best performance of

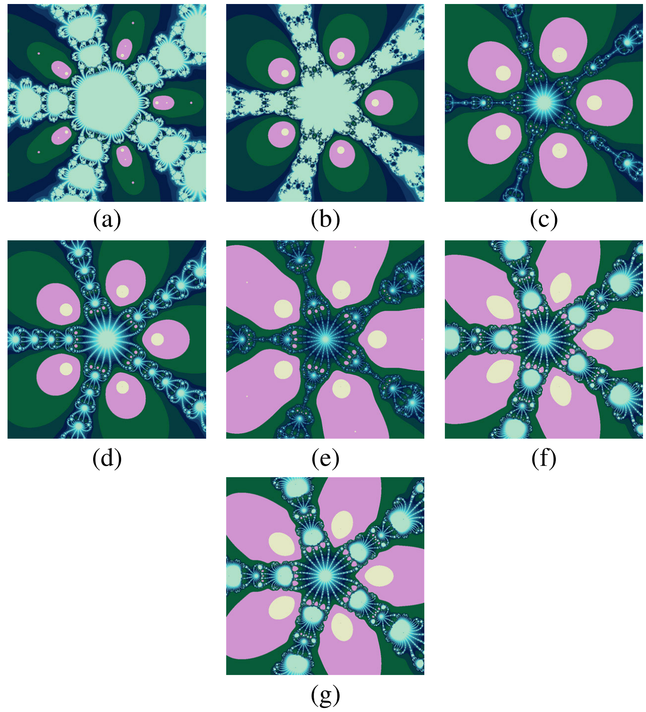

FIGURE 6. Polynomiographs associated with the polynomial p_5. (a) stands for NIM1, (b) for NIM2, (c) for OIM, (d) for TIM, (e) for MHIM, (f) for Algorithm 1 and (g) for Algorithm 2.

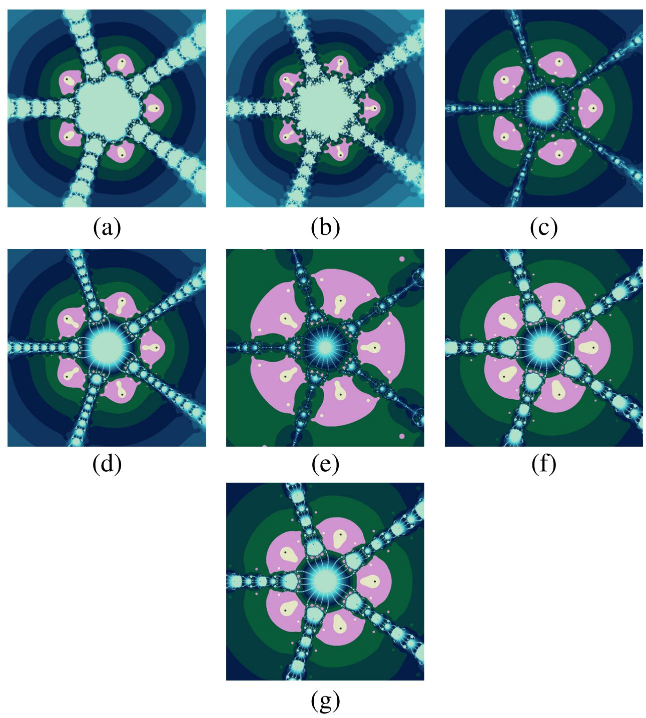

FIGURE 7. Polynomiographs associated with the polynomial p_6. (a) stands for NIM1, (b) for NIM2, (c) for OIM, (d) for TIM, (e) for MHIM, (f) for Algorithm 1 and (g) for Algorithm 2.

the considered algorithm whereas the black color in images points out those places at which the solution is impossible to achieve for the defined number of iterations. The same colors' areas in the presented images providing us the information about the same iterations' consumption for determining the approximate zeros and appear identical to the contour lines on the map. It should be noted that the drawn images through the constructed algorithms possess brighter areas and almost free from black areas against the other ones of similar type that justified the supremacy of the constructed algorithms over the other ones.

VII. CONCLUDING REMARKS

In this article, we established two new root-finding algorithms for the solution of non-linear functions which bearing the convergence of sixth-order. These algorithms have been derived by employing Househölder's method, interpolation technique and Taylor's series expansion and one of them is second-derivative free that results in a better efficiency with less computational cost. The applicability, validity and performance of the constructed algorithms have been analyzed by solving some test functions including engineering and arbitrary problems. Tables 1–3 show the numerical results of some test examples that certified the superiority of established algorithms with respect to convergence speed, efficiency, accuracy and the iterations' consumption over the other iteration schemes with similar nature. We ran the constructed algorithms with the aid of computer technology for creating some new art in the form of aesthetically pleasing images by considering some complex polynomials and compared them with other two-step algorithms. The obtained images are brighter, colorful, having complex and aesthetic patterns which reveal the convergence speed, fractal behaviors and some other graphical features of the proposed algorithms. Using the idea of this paper, one can derive a class of new iterative algorithms and create some new mathematical art through these algorithms in future work. The proven results of this paper may give rise to further research in this field.

REFERENCES

[1] D. Chen, I. K. Argyros, and Q. S. Qian, "A note on the Halley method in Banach spaces," *Appl. Math. Comput.*, vol. 58, nos. 2–3, pp. 215–224, Oct. 1993.

[2] G. V. Balaji and J. D. Seader, "Application of interval Newton's method to chemical engineering problems," *Reliab. Comput.*, vol. 3, no. 1, pp. 215–223, 1995.

[3] A. F. Beardon, "Iteration of rational functions," in *Complex Analytic Dynamical Systems*. New York, NY, USA: Springer-Verlag, 1991.

[4] R. L. Burden and J. D. Faires, *Numerical Analysis*. Boston, MA, USA: Brooks/Cole, 1997.

[5] R. L. Devaney, *A First Course in Chaotic Dynamic System Theory and Experiment*. Boca Raton, FL, USA: ABP, 1992.

[6] 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, Sep. 2019, doi: 10.3390/sym11091143.

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

[8] J. Gleick, *Making of a Science*. Harmondsworth, U.K.: Pengin Books, 1988.

[9] I. Gośćiniak and K. Gdawiec, "Visual analysis of dynamics behaviour of an iterative method depending on selected parameters and modifications," *Entropy*, vol. 22, no. 7, p. 734, Jul. 2020, doi: 10.3390/e22070734.

[10] J. M. Gutiérrez and M. A. Hernández, "An acceleration of Newton's method: Super-Halley method," *Appl. Math. Comput.*, vol. 117, nos. 2–3, pp. 223–239, Jan. 2001.

[11] D. Herceg and D. Herceg, "Eighth order family of iterative methods for nonlinear equations and their basins of attraction," *J. Comput. Appl. Math.*, vol. 343, pp. 458–480, Dec. 2018, doi: 10.1016/j.cam.2018.04.040.

[12] A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*. New York, NY, USA: McGraw-Hill, 1970.

[13] B. Kalantari, "An invitation to polynomiography via exponential series," 2017, *arXiv:1707.09417*. [Online]. Available: http://arxiv.org/abs/1707.09417

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

[15] B. Kalantari, "Method of creating graphical works based on polynomials," U.S. Patent 6 894 705, May 17, 2005.

[16] B. Kalantari, "Polynomiography: From the fundamental theorem of algebra to art," *Leonardo*, vol. 38, no. 3, pp. 233–238, Jun. 2005.

[17] B. Kalantari and E. H. Lee, "Newton–Ellipsoid polynomiography," *J. Math. Arts*, vol. 13, no. 4, pp. 336–352, Nov. 2019, doi: 10.1080/17513472.2019.1600959.

[18] S. M. Kang, A. Naseem, W. Nazeer, and A. Jan, "Modification of Abbasbandy's method and polynomigraphy," *Int. J. Math. Anal.*, vol. 10, pp. 1197–1210, 2016.

[19] D. E. Kincaid and E. W. Cheney, *Numerical Analysis*. Pacific Grove, CA, USA: Brooks/Cole, 1990.

[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," *J. Comput.*, vol. 330, pp. 676–694, Mar. 2018, doi: 10.1016/j.cam.2017.07.005.

[21] B. B. Mandelbrot, *Fractal Geometry of Nature*. New York, NY, USA: W. F. Freeman, 1993.

[22] R. Manning, "On the flow of water in open channels and pipes," *Trans. Inst. CE*, vol. 20, pp. 161–207, 1891.

[23] J. Milnor, *Dynamics in One Complex Variable: Introductory Lectures*, vol. 160. Princeton, NJ, USA: Princeton Univ. Press, 2006.

[24] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Some new iterative algorithms for solving one-dimensional non-linear equations and their graphical representation," *IEEE Access*, vol. 9, pp. 8615–8624, 2021, doi: 10.1109/ACCESS.2021.3049428.

[25] W. Nazeer, A. Naseem, S. M. Kang, and Y. C. Kwun, "Generalized Newton Raphson's method free from second derivative," *J. Nonlinear Sci. Appl.*, vol. 9, pp. 2823–2831, Jan. 2016.

[26] W. Nazeer, M. Tanveer, S. M. Kang, and A. Naseem, "A new Householder's method free from second derivatives for solving nonlinear equations and polynomiography," *J. Nonlinear Sci. Appl.*, vol. 9, pp. 998–1007, Jan. 2016.

[27] M. A. Noor, K. I. Noor, and K. Aftab, "Some new iterative methods for solving nonlinear equations," *World Appl. Sci. J.*, vol. 20, no. 6, pp. 870–874, 2012.

[28] M. A. Noor, W. A. Khan, and A. Hussain, "A new modified Halley method without second derivatives for nonlinear equation," *Appl. Math. Comput.*, vol. 189, no. 2, pp. 1268–1273, Jun. 2007.

[29] A. M. Ostrowski, *Solution of Equations and Systems of Equations*. New York, NY, USA: Academic, 1966.

[30] H. O. Peitgen, D. Saupe, H. Jurgens, and L. Yunker, *Chaos and Fractals*. New York, NY, USA: Springer-Verlag, 1992.

[31] 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," *Int. J. Comput. Math.*, vol. 95, no. 11, pp. 2174–2211, Nov. 2018, doi: 10.1080/00207160.2017.1367387.

[32] M. Shacham, "An improved memory method for the solution of a nonlinear equation," *Chem. Eng. Sci.*, vol. 44, no. 7, pp. 1495–1501, 1989.

[33] M. Shacham and E. Kehat, "Converging interval methods for the iterative solution of nonlinear equations," *Chem. Eng. Sci.*, vol. 1973, 28, pp. 2187–2193.

[34] M. Scott, B. Neta, and C. Chun, "Basin attractors for various methods," *Appl. Math. Comput.*, vol. 218, no. 6, pp. 2584–2599, Nov. 2011.

[35] J. R. Sharma and H. Arora, "A new family of optimal eighth order methods with dynamics for nonlinear equations," *Appl. Math. Comput.*, vol. 273, pp. 924–933, Jan. 2016.

[36] O. S. Solaiman, S. A. A. Karim, and I. Hashim, "Optimal fourth- and eighth-order of convergence derivative-free modifications of King's method," *J. King Saud Univ.-Sci.*, vol. 31, no. 4, pp. 1499–1504, Oct. 2019, doi: 10.1016/j.jksus.2018.12.001.

[37] J. F. Traub, *Iterative Methods for the Solution of Equations*. New York, NY, USA: Chelsea, 1982.

[38] X. Wang and M. Zhu, "Two iterative methods with memory constructed by the method of inverse interpolation and their dynamics," *Mathematics*, vol. 8, no. 7, p. 1080, Jul. 2020, doi: 10.3390/math8071080.

[39] V. D. Waals and J. Diderik, "Over de continuiteit van den gas-en vloeistoftoestand (on the continuity of the gas and liquid state)," Ph.D. dissertation, Leiden Univ., Leiden, The Netherlands, 1873.

[40] S. Weerakoon and T. G. I. Fernando, "A variant of Newton's method with accelerated third-order convergence," *Appl. Math. Lett.*, vol. 13, no. 8, pp. 87–93, 2000.

AMIR NASEEM received the M.S. degree in mathematics from Lahore Leads University, Lahore, Pakistan. He is currently pursuing the Ph.D. degree with the Department of Mathematics, University of Management and Technology, Lahore. He has published more than 25 research articles in different well known journals. His current research interests include numerical analysis, iterative methods, and polynomiography.

M. A. REHMAN received the Ph.D. degree in mathematics from Government College University, Lahore, Pakistan. He is currently a Full Professor with the University of Management and Technology, Lahore. He has published over 50 research articles in different international journals. His main research interests include modeling, graph theory, and mathematical analysis.

THABET ABDELJAWAD received the Ph.D. degree in mathematics from Middle East Technical University, in 2000. He is currently a Full Professor with Prince Sultan University. He has published more than 350 research articles in different well known journals. His main research interests include fractional calculus, discrete fractional operators, metric spaces, and fixed point theory.