

Research Article

A New Root-Finding Algorithm for Solving Real-World Problems and Its Complex Dynamics via Computer Technology

Amir Naseem ¹, M. A. Rehman ¹ and Jihad Younis ²

¹Department of Mathematics, University of Management and Technology, Lahore 54770, Pakistan

²Department of Mathematics, Aden University, P.O.Box, Aden 6014, Yemen

Correspondence should be addressed to Amir Naseem; amir.kasuri89@gmail.com and Jihad Younis; jihadalsaqqaf@gmail.com

Received 10 September 2021; Accepted 18 October 2021; Published 29 November 2021

Academic Editor: Maricel Agop

Copyright © 2021 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.

Nowadays, the use of computers is becoming very important in various fields of mathematics and engineering sciences. Many complex statistics can be sorted out easily with the help of different computer programs in seconds, especially in computational and applied Mathematics. With the help of different computer tools and languages, a variety of iterative algorithms can be operated in computers for solving different nonlinear problems. The most important factor of an iterative algorithm is its efficiency that relies upon the convergence rate and computational cost per iteration. Taking these facts into account, this article aims to design a new iterative algorithm that is derivative-free and performs better. We construct this algorithm by applying the forward- and finite-difference schemes on Golbabai–Javidi’s method which yields us an efficient and derivative-free algorithm whose computational cost is low as per iteration. We also study the convergence criterion of the designed algorithm and prove its quartic-order convergence. To analyze it numerically, we consider nine different types of numerical test examples and solve them for demonstrating its accuracy, validity, and applicability. The considered problems also involve some real-life applications of civil and chemical engineering. The obtained numerical results of the test examples show that the newly designed algorithm is working better against the other similar algorithms in the literature. For the graphical analysis, we consider some different degrees’ complex polynomials and draw the polynomiographs of the designed quartic-order algorithm and compare it with the other similar existing methods with the help of a computer program. The graphical results reveal the better convergence speed and the other graphical characteristics of the designed algorithm over the other comparable ones.

1. Introduction

The role of computers in the fields of applied mathematics cannot be denied in the modern age. Using different computer programs such as Mathematica, Matlab, and Maple, a plethora of different types of complex problems can be solved easily. In recent years, the mathematicians employed the excessive use of computers in different fields of mathematics. The most important among them is the polynomial’s root finding which has played a significant role in applied and computational mathematics and covers many other fields of modern sciences. There exist several problems in engineering which arise in the form of nonlinear equations. For solving such type of engineering problems, we need iterative algorithms because in most of the cases,

analytical methods do not work. The well-known classical iterative algorithm was given by Newton [1] in the last of fifteenth century. In the modern age, the researchers improved the order and efficiency of the existing methods and introduced multistep algorithms. For more details, see [2–11] and the references cited therein. Usually, multistep algorithms possess greater convergence with higher computational cost due to the involvement of higher derivatives which is the downside of these algorithms. It is really a tough task to balance the computational cost and the convergence order because if we increase one of them, the other would be decreased.

In the last few years, the researchers modified the existing iteration schemes by employing different mathematical techniques and suggested a new variety of multistep

iterative methods. In [12], the author introduced the predictor-corrector method given as

$$\begin{aligned} v_p &= u_p - \frac{f(u_p)}{f'(u_p)}, \quad p = 0, 1, 2, 3, \dots, \\ u_{p+1} &= u_p - \frac{f(u_p)}{f'(u_p)}, \end{aligned} \quad (1)$$

which is quartic-order algorithm suggested by Traub (TM) for root finding of scalar nonlinear equations.

Noor et al. [13], in 2012, introduced an iterative algorithm of the form:

$$\begin{aligned} v_p &= u_p - \frac{f(u_p)}{f'(u_p)}, \quad p = 0, 1, 2, 3, \dots, \\ u_{p+1} &= u_p - \frac{f(u_p)}{f'(u_p)} + \left(\frac{f(u_p)}{f'(u_p)} \right) \frac{f'(v_p)}{f'(u_p)}, \end{aligned} \quad (2)$$

which is known as Noor's method (NRM) for root finding of scalar nonlinear equations.

In 2010, Zhanlav et al. [14] introduced a new three-step iterative algorithm given as

$$\begin{aligned} v_p &= u_p - \frac{f(u_p)}{f'(u_p)}, \quad p = 0, 1, 2, 3, \dots, \\ w_p &= v_p - \frac{f(v_p)}{f'(u_p)}, \\ u_{p+1} &= v_p - \frac{f(v_p) + f(w_p)}{f'(u_p)}, \end{aligned} \quad (3)$$

which is quartic-order algorithm for root finding of scalar nonlinear equations known as Zhanlav's method (ZM).

In the present research article, we introduced a new fourth order and derivative-free algorithm for solving engineering and arbitrary problems in the form of scalar nonlinear functions. The construction of this algorithm is based upon the forward- and finite-difference schemes on Golbabai-Javidi's method. We also certified that the designed algorithm has fourth-order convergence and then applied to different real-world engineering and arbitrary problems to demonstrate its applicability among the other fourth-order algorithms in the literature. The dynamical comparison of the designed algorithm with the other comparable ones has been also presented via computer technology using the program Mathematica 12.0.

The rest of the paper is divided as follows. An efficient and derivative-free algorithm has been constructed in Section 2. The convergence criterion of the designed algorithm has been discussed in Section 3. In Section 4, nine arbitrary and engineering problems have been solved. The graphical analysis of the designed algorithm has been

presented in Section 5. Finally, the conclusion of the paper is given in Section 6.

2. Main Results

Let $f: D \rightarrow R$, $D \subset R$ be a function in one variable; then, by employing the modified homotopy perturbation method, Golbabai and Javidi [15] introduced the following root-finding algorithm:

$$u_{p+1} = u_p - \frac{f(u_p)}{f'(u_p)} - \frac{f^2(u_p)f''(u_p)}{2[f'^3(u_p) - f(u_p)f'(u_p)f''(u_p)]}, \quad (4)$$

which is known as Golbabai-Javidi's method of cubic convergence [15]. In [16], the authors modified it and suggested a two-step iteration scheme of the following form:

$$\begin{aligned} v_p &= u_p - \frac{f(u_p)}{f'(u_p)}, \\ u_{p+1} &= v_p - \frac{f(v_p)}{f'(v_p)} - \frac{f^2(v_p)f''(v_p)}{2[f'^3(v_p) - f(v_p)f'(v_p)f''(v_p)]}, \end{aligned} \quad (5)$$

which is a two-step iteration scheme for calculating zeros of nonlinear scalar equations. The major drawback of the above algorithm is its high computational cost per iteration as it requires first and second derivatives for its execution. To lower its computational cost and make it more effective, we approximate its derivatives and make it derivative-free so that it can be easily applied on those nonlinear functions whose first derivative becomes infinite or does not exist. To approximate $f''(v)$, we employ the finite-difference approximation as

$$f''(v_p) = \frac{f'(v_p) - f'(u_p)}{f(v_p) - f(u_p)}. \quad (6)$$

Now, we apply the forward-difference scheme to approximate first derivatives as follows:

$$f'(u_p) = \frac{f(u_p + f(u_p)) - f(u_p)}{f(u_p)} = g(u_p), \quad (7)$$

$$f'(v_p) = \frac{f(v_p + f(v_p)) - f(v_p)}{f(v_p)} = h(v_p). \quad (8)$$

Using (7) and (8) in (9),

$$f''(v_p) = \frac{h(v_p) - g(u_p)}{f(v_p) - f(u_p)} = j(u_p, v_p). \quad (9)$$

Using (7)–(9) in (5), we are able to write Algorithm 1.

Algorithm 1. For a given initial guess u_0 , determine the approximate solution u_{p+1} by the following iteration schemes:

$$\begin{aligned} v_p &= u_p - \frac{f(u_p)}{g(u_p)}, \\ u_{p+1} &= v_p - \frac{f(v_p)}{h(v_p)} - \frac{f^2(v_p)j(u_p, v_p)}{2[h^3(v_p) - f(v_p)h(v_p)j(u_p, v_p)]}, \end{aligned} \quad (10)$$

which is a novel derivative-free two-step iterative algorithm for calculating approximate zeros of nonlinear functions in one variable. The basic and important feature of the presented algorithm is its applicability area because it also covers such nonlinear functions in which first and second derivatives do not exist. The replacements of the first and second derivatives reduce its computational cost per iteration which yield higher efficiency index against the other similar iteration schemes. The obtained numerical results of

test examples show better performance of the proposed algorithm against the other comparable ones in literature.

3. Convergence Analysis

In the present section, we proved the convergence order of the proposed method, i.e., Algorithm 1.

Theorem 1. Suppose β be the simple zero of $f(u) = 0$, where $f(u)$ is sufficiently smooth near the exact root β ; then, Algorithm 1 possesses quartic-order convergence. Moreover, it satisfies the following error equation:

$$e_{p+1} = Ae_p^4 + O(e_p^5), \quad (11)$$

where $A = 2(f''(\beta)/2f'(\beta))^3$.

Proof. To prove the theorem, suppose e_p as the error at p th iteration; then, $e_p = u_p - \beta$ and the Taylor series about $u = \beta$ implies

$$\begin{aligned} f(u_p) &= f'(\beta)e_p + \frac{1}{2!}f''(\beta)e_p^2 + \frac{1}{3!}f'''(\beta)e_p^3 \\ &\quad + \frac{1}{4!}f^{(iv)}(\beta)e_p^4 + O(e_p^5), \end{aligned} \quad (12)$$

$$f(u_p) = f'(\beta)[e_p + c_2e_p^2 + c_3e_p^3 + c_4e_p^4 + O(e_p^5)], \quad (13)$$

$$g(u_p) = f'(\beta)[1 + 3c_2e_p + (7c_3 + c_2^2)e_p^2 + (6c_2c_3 + 15c_4)e_p^3 + (18c_2c_4 + 31c_5 + c_3c_2^2 + 5c_3^2)e_p^4 + O(e_p^5)], \quad (14)$$

where

$$c_p = \frac{1}{q!} \frac{f^{(q)}(\beta)}{f'(\beta)}. \quad (15)$$

Using (13) and (14), we obtain

$$v_p = f'(\beta)[\beta + 2c_2e_p^2 + (6c_3 - 5c_2^2)e_p^3 + (-26c_2c_3 + 13c_2^3 + 14c_4) + O(e_p^5)], \quad (16)$$

$$f(v_p) = f'(\beta)[2c_2e_p^2 + (6c_3 - 5c_2^2)e_p^3 + (17c_2^3 - 26c_2c_3 + 14c_4)e_p^4 + O(e_p^5)],$$

$$h(v_p) = f'(\beta)[1 + 6c_2^2e_p^2 + (18c_2c_3 - 15c_2^3)e_p^3 + (-50c_3c_2^2 + 43c_2^4 + 42c_2c_4)e_p^4 + O(e_p^5)], \quad (17)$$

$$\begin{aligned} j(u_p, v_p) &= f'(\beta)[3c_2 + (7c_3 - 2c_2^2)e_p + (10c_2c_3 + 15c_4 - 2c_2^3)e_p^2 \\ &\quad + (-62c_3c_2^2 + 16c_2^4 + 30c_2c_4 + 31c_5 + 40c_3^2)e_p^3 \\ &\quad + (78c_5c_2 - 258c_2c_3^2 + 188c_3c_4 + 237c_3c_2^3 - 64c_2^2c_4 + 63c_6 - 48c_2^5)e_p^4 + O(e_p^5)]. \end{aligned} \quad (18)$$

Using equations (13)–(18) in Algorithm 1, we get the following equality:

$$u_{p+1} = \beta + 2c_2^3 e_p^4 + O(e^5), \quad (19)$$

which implies that

$$e_{p+1} = 2c_2^3 e_p^4 + O(e^5). \quad (20)$$

The above equation established that Algorithm 1 is of quartic-order convergence. \square

4. Numerical Comparisons and Applications

In this section, we include five arbitrary and four engineering problems to demonstrate the validity, accuracy, and robust performance of the newly designed iteration scheme. We compare the suggested algorithm with Noor's method [13], Traub's method [12], and Zhanlav's method (ZM) [14] by considering some nonlinear problems.

Example 1. Blood rheology model.

Blood rheology is a branch of science that works to study the physical and flow properties of the blood [17]. Blood is actually a non-Newtonian fluid and treated as Caisson Fluid. The model of Caisson fluid shows that the flow of simple fluids in a tube is such a way that the center core of the fluids will move as a plug with little deformation and velocity gradient occurs near the wall. To study the plug flow of Caisson fluids flow, we consider the following function in the form of nonlinear equation as

$$H = 1 - \frac{16}{7} \sqrt{u} + \frac{4}{3} u - \frac{1}{21} u^4, \quad (21)$$

where flow-rate reduction is computed by H . Using $H = 0.40$ in (16), we have

$$f_1(u) = \frac{1}{441} u^8 - \frac{8}{63} u^5 - 0.05714285714 u^4 + \frac{16}{9} u^2 - 3.624489796 u + 0.3. \quad (22)$$

To solve f_1 , the initial guess has been chosen as $u_0 = 0.9$ for starting the iteration process, and the results are given in Table 1.

Example 2. Finding volume from van der Waal's equation.

In engineering, the famous and well-known van der Waal's equation is used to examine gases behaviors [18] which was introduced by van der Waal:

$$\left(P + \frac{K_1 n^2}{V^2} \right) (V - nK_2) = nRT. \quad (23)$$

By assuming feasible values of the appearing parameters in (23), we obtain the following nonlinear problem:

$$f_2(u) = 0.986u^3 - 5.181u^2 + 9.067u - 5.289, \quad (24)$$

where u denotes the volume and may be found easily by solving f_2 . Since f_2 is a polynomial with cubic degree, so

clearly three roots exist and the only one among them is feasible positive real root 1.9298462428 as the volume of the gas is always positive. To start the iteration process, we take $u_0 = 0.10$, and the results are inserted in Table 2.

Example 3. Open channel flow problem.

The flowing water in the open channel with the condition of uniform flow is described in the well-known Manning's equation [19]:

$$\text{Flow of Water} = G = \frac{\sqrt{s} ar^{2/3}}{N}. \quad (25)$$

The parameters r , s , and a in (25) denote hydraulic radius, slope, and area of the channel, respectively, and N is Manning's roughness coefficient. For a rectangular-shaped channel with depth u and width w , we have

$$a = wu, \text{ \& } r = \frac{wu}{w + 2u}. \quad (26)$$

Using these values in (25), we obtain

$$G = \frac{\sqrt{s} wu}{N} \left(\frac{wu}{w + 2u} \right)^{2/3}. \quad (27)$$

For determining the depth of water, the above relation may be written as

$$f_3(u) = \frac{\sqrt{s} wu}{N} \left(\frac{wu}{w + 2u} \right)^{2/3} - G. \quad (28)$$

We assign the values to the parameters as $G = 14.15 \text{ m}^3/\text{s}$, $w = 4.572 \text{ m}$, $s = 0.017$, and $N = 0.0015$. We select the starting point $u_0 = 4.50$ to initialize the iteration process, and the results are inserted in Table 3.

Example 4. Plank's radiation law.

For finding the energy density in the isothermal black body, Planck's radiation law is used which was introduced by Planck [20] in 1914 having the following mathematical expression:

$$f(\sigma) = \frac{8\pi cP}{\sigma^5 (e^{cP/\sigma kT} - 1)}. \quad (29)$$

To determine wavelength σ_1 against the peak value of energy density $f(\sigma_1)$, we convert equation (29) in nonlinear equation by considering $u = cP/\sigma kT$, given as below:

$$1 - \frac{u}{5} = e^{-u}, \quad (30)$$

which can be converted into the form of nonlinear function as follows:

$$f_4(u) = e^{-u} + \frac{u}{5} - 1. \quad (31)$$

The approximated root of f_4 denotes the highest wavelength of the radiation. We select $u_0 = 2.10$ as the initial guess in the iteration process, and the results are inserted in Table 4.

Example 5. Arbitrary problems.

TABLE 1: Comparison among different iteration schemes for f_1 .

Method	N	u_{p+1}	$ f(u_{p+1}) $	$\sigma = u_{p+1} - u_p $
$f_1(u), u_0 = 0.90$				
NRM	10	0.0864335580522917557442213197867	$4.719728e - 30$	$1.950374e - 16$
TM	3	0.0864335580522916428021146178962	$3.697508e - 16$	$1.642942e - 04$
ZM	12	0.0864335580522917542616822036793	$1.957052e - 25$	$5.571776e - 07$
Algorithm 1	3	0.0864335580522918923504632529957	$4.580893e - 16$	$9.040753e - 05$

TABLE 2: Comparison among different iteration schemes for f_2 .

Method	N	u_{p+1}	$ f(u_{p+1}) $	$\sigma = u_{p+1} - u_p $
$f_2(u), u_0 = 0.10$				
NRM	21	1.9298462428478622080651849074883	$9.008231e - 19$	$1.306816e - 09$
TM	18	1.9298462428478622184875274278655	$1.177178e - 38$	$1.564552e - 10$
ZM	—	Did not converge towards the approximated root	—	—
Algorithm 1	10	1.9298462428478622184871569605409	$3.202020e - 23$	$5.954550e - 07$

To examine the numeric behavior of the designed algorithm, we take five arbitrary problems, and the results are inserted in Table 5.

$$\begin{aligned}
f_5(u) &= u^3 + 4u^2 - 10, u_0 = -0.3, \\
f_6(u) &= u^3 + u^2 - 2, u_0 = -0.1, \\
f_7(u) &= e^u + \cos u - 1, u_0 = 1.0, \\
f_8(u) &= 2 - 2\sqrt{u}, u_0 = 2.0, \\
f_9(u) &= u^2 - 3ue^u + 2, u_0 = 3.5.
\end{aligned} \tag{32}$$

In Tables 1–5, a detailed comparison of well-known iterative algorithms with the suggested algorithm has been presented. The columns of the tables give us details about the quantity of consumed iterations, the final approximated root, the absolute value of the function at that root, and the positive difference between the two consecutive approximations.

The obtained results of the test examples which are summarized in the form of Tables 1–5 show the performance of the proposed algorithm, and from these results, and we can claim the robust performance of the designed algorithm in terms of accuracy, speed, number of iterations, computational cost and can say that it is superior to the other comparable algorithms.

5. Complex Dynamics

In this section, we study the complex dynamics of the suggested root-finding algorithm for different complex polynomials in the form of polynomiographs which are generated in the process of polynomiography. The term polynomiography was first introduced in 2005 by Bahman Kalantari [21, 22] who defined it as a process of drawing aesthetically pleasing graphical objects by employing the mathematical convergence properties of iteration functions. By an iteration function, we mean such a function that is composed onto itself again and again such as Newton's algorithm [23], Halley's algorithm [24], and Householder's

algorithm [25]. As a result of polynomiography, the generated graphical objects are known as polynomiographs. To plot the polynomiographs on the complex plane \mathbb{C} via computer program by considering different complex polynomials, we take a rectangle $R \in \mathbb{C}$ of size $[-2, 2] \times [-2, 2]$, with the accuracy $\varepsilon = 0.001$ and maximum number of iterations $L = 20$. This rectangle contains the roots of the considered polynomial, and corresponding to the starting point z_0 in R , we initialize the iteration process and assign a color to the point corresponding to z_0 . The black color is assigned to those particular points at which the algorithm fails to converge. The quality and resolution of the generated graphical objects are depended on the discretization of R , i.e., if we discretize R into the grid of 2000×2000 , the plotted polynomiographs will have better resolution and image quality.

We know that if q is an n th-degree polynomial, then it must possess n th number of zeros according to the theorem of algebra and may be expressed as

$$q(z) = d_n z^n + d_{n-1} z^{n-1} + \dots + d_1 z + d_0. \tag{33}$$

If $r_1, r_2, \dots, r_{n-1}, r_n$ are the roots (zeros) of q , then (33) may be rewritten as

$$q(z) = (z - r_1)(z - r_2) \dots (z - r_p), \tag{34}$$

where $\{d_n, d_{n-1}, \dots, d_1, d_0\}$ are the complex coefficients.

Any algorithm that involves iteration process may be applied to both above-described expressions of q for plotting graphical objects. The general algorithm for plotting polynomiographs is given in Algorithm 2.

We consider an algorithm to be converged if the convergence test $(z_{p+1}, z_p, \varepsilon)$ returns TRUE and diverged if it returns False in Algorithm 1. The standard test for studying the convergence or divergence of an algorithm is given as

$$|z_{p+1} - z_p| < \varepsilon, \tag{35}$$

where $\varepsilon > 0$ is the accuracy and z_p and z_{p+1} are the two consecutive estimations in the iteration process. In this

TABLE 3: Comparison among different iteration schemes for f_3 .

Method	N	u_{p+1}	$ f(u_{p+1}) $	$\sigma = u_{p+1} - u_p $
$f_3(u), u_0 = 4.50$				
NRM	5	1.4650912202958246423760162175260	$6.373023e - 23$	$6.845054e - 12$
TM	3	1.4650912202958246423760209097786	$5.480363e - 54$	$1.415763e - 13$
ZM	4	1.4650912202958246423760209097786	$2.756508e - 46$	$8.430688e - 12$
Algorithm 1	3	1.4650912202958246426042177582695	$3.099372e - 18$	$1.661971e - 05$

TABLE 4: Comparison among different iteration schemes for f_4 .

Method	N	u_{p+1}	$ f(u_{p+1}) $	$\sigma = u_{p+1} - u_p $
$f_4(u), u_0 = -0.75$				
NRM	6	4.9651142317442763036985768576639	$7.178359e - 17$	$1.198195e - 08$
TM	3	4.9651142317442763036987591313229	$1.855817e - 52$	$3.572333e - 12$
ZM	4	4.9651142317442763036987591313230	$2.614800e - 32$	$2.752091e - 07$
Algorithm 1	3	4.9651142317442763036987590761003	$1.065923e - 26$	$6.186151e - 06$

TABLE 5: Comparison among different iteration schemes for $f_5 - f_9$.

Method	N	$ f(u_{p+1}) $	u_{p+1}	$\sigma = u_{p+1} - u_p $
$f_5(u), u_0 = -0.3$				
NRM	58	1.3652300134140968454590769485322	$4.982586e - 18$	$7.845136e - 10$
TM	27	1.3652300134140968457608068289822	$8.127500e - 30$	$4.520819e - 08$
ZM	136	1.3652300134140968904526720345641	$7.380146e - 16$	$9.868624e - 05$
Algorithm 1	25	1.3652300134140968457608114983044	$7.710639e - 23$	$2.954643e - 07$
$f_6(u), u_0 = -0.1$				
NRM	60	0.9999999999999997639576027795525	$1.180212e - 15$	$1.717711e - 08$
TM	7	1.0000000000000000000000000000000	$7.765738e - 39$	$2.346852e - 10$
ZM	196	1.0000000000000000000000000000000	$5.211534e - 55$	$1.501988e - 14$
Algorithm 1	4	1.000000000000000000000000089529	$4.476436e - 26$	$9.784273e - 08$
$f_7(u), u_0 = 1.0$				
NRM	5	-0.9236326589551345576746795452030	$2.862038e - 28$	$5.273210e - 14$
TM	3	-0.9236326589551345576746795454425	$2.191685e - 36$	$7.319159e - 09$
ZM	3	-0.9236326589551345576746795458139	$4.437593e - 28$	$6.173604e - 07$
Algorithm 1	3	-0.9236326589551345576746795454425	$5.098187e - 41$	$3.564915e - 10$
$f_8(u), u_0 = 2.0$				
NRM	6	1.000000000000000000000000068160	$6.815985e - 27$	$1.651180e - 13$
TM	4	1.0000000000000000000000000000000	$3.098508e - 43$	$6.673185e - 11$
ZM	3	0.999999999999999999999999999988	$1.198715e - 30$	$6.617730e - 08$
Algorithm 1	2	0.999999999999999999999999970154	$2.984601e - 27$	$1.352139e - 04$
$f_9(u), u_0 = 3.5$				
NRM	7	0.4642355822914875455308056932203	$5.529675e - 19$	$3.366148e - 10$
TM	5	0.4642355822914875456220629329382	$2.303726e - 31$	$1.642476e - 08$
ZM	6	0.4642355822914875456220629329382	$2.963421e - 49$	$3.911331e - 13$
Algorithm 1	5	0.4642355822914875456220629329382	$2.044940e - 37$	$1.470701e - 10$

study, we also consider equation (20) as a stopping criterion. A variety of different color graphical objects can be plotted by changing the parameters ε and k and the iteration scheme. For further details about polynomiography and its applications, see [26–38] and the references therein.

We take the following four different complex polynomials for plotting graphical objects in the complex plane:

$$q_1(z) = z^2 - 1, q_2(z) = z^3 - 1, q_3(z) = z^4 - 1. \quad (36)$$

For coloring the iterations, we employ the colormap given in Figure 1.

Example 6. Polynomiographs for q_1 through various iteration schemes.

In the first example, we investigate and compare the dynamical results obtained through different iteration schemes with the newly designed quartic-order algorithm for the quadratic-degree polynomial $q^2 - 1$ which possesses two distinct zeros: 1 and -1 . We executed all comparable algorithms to achieve the simple roots, and the results can be visualized in Figure 2.

Example 7. Polynomiographs for q_2 through various iteration schemes.

Input: $q \in \mathbb{C}[Z]$ is a complex polynomial, $\mathfrak{A} \subset \mathbb{C}$ is area in which polynomiographs are drawn, L is upper bound of iterations, ϕ is iterative scheme, ϵ is accuracy for the stopping criterion, colormap $[0, \dots, C - 1]$ is colormap with C colors.

Output: polynomiograph in the area in \mathfrak{A} for the complex polynomial q .

for $z_0 \in \mathfrak{A}$ **do**

$p = 0$

While $p \leq L$ **do**

$z_{p+1} = \phi(z_p)$

if $|z_{p+1} - z_p| < \epsilon$ **then**

break

$p = p + 1$

 color z_0 via colormap.

ALGORITHM 2: Generation of the polynomiograph.



FIGURE 1: The colormap for drawing polynomiographs.

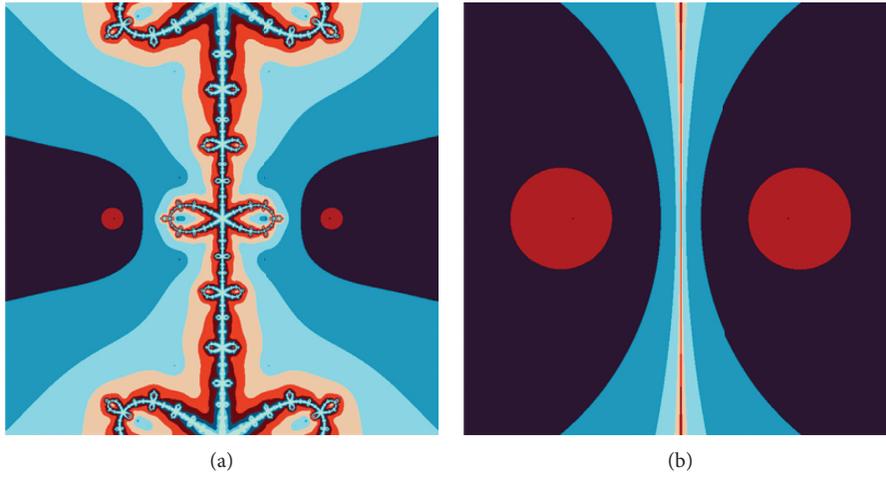


FIGURE 2: Continued.

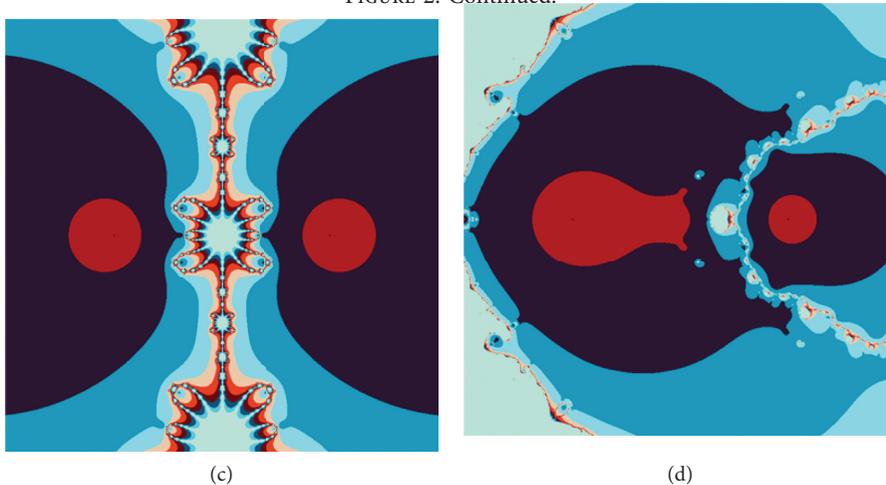


FIGURE 2: Polynomiographs for the quadratic-degree polynomial q_1 . (a) represents NRM, (b) represents TM, (c) represents ZM, and (d) represents Algorithm 1.

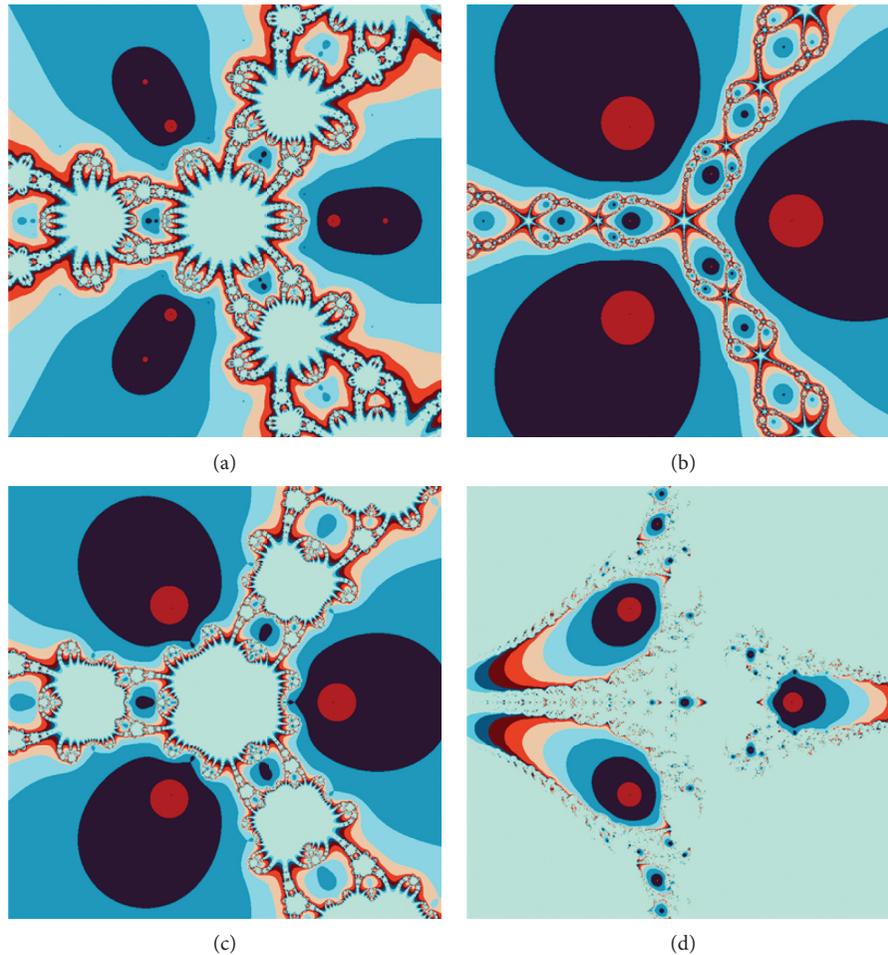


FIGURE 3: Polynomiographs for the quadratic-degree polynomial q_2 . (a) represents NRM, (b) represents TM, (c) represents ZM, and (d) represents Algorithm 1.

In the second example, we investigate and compare the dynamical results obtained through different iteration schemes with the newly designed quartic-order algorithm for the cubic-degree polynomial $q^3 - 1$ which possesses three distinct zeros: 1 , $-1/2 - \sqrt{3}/2i$, and $-1/2 + \sqrt{3}/2i$. We executed all comparable algorithms to achieve the simple roots, and the results can be visualized in Figure 3.

Example 8. Polynomiographs for q_3 through various iteration schemes.

In the third example, we consider quartic-degree polynomial q_3 , having roots 1 , -1 , i , and $-i$. We drew the aesthetically pleasing graphical objects by executing all comparable algorithms, and the results can be visualized in Figure 4.

In Examples 6–8, we ran all the comparable algorithms for drawing the aesthetically pleasing polynomiographs with the help of a computer program Mathematica 12.0. From the obtained graphical objects, we can easily examine the graphical behavior and stability of different algorithms. It should be noted that the convergence region of the newly

designed algorithm is much larger as compared to other ones. The colors' shades exhibit the performance of that algorithm by which the polynomiograph is drawn. Two main characteristics can be revealed through these graphical objects which are the convergence speed and the dynamics of the considered iteration schemes by which these graphs have been generated. The first one can be depicted by analyzing the shades of colors of the image. The richness of the color in the graphical objects shows robust convergence with less iterations. The second property can be investigated by visualizing the variation of the colors of the drawn polynomiographs. The small variation's regions indicate the low dynamic zones and the dynamics are high for the zones of large variation of colors. The black color in the presented graphical objects showing the divergence zone of the algorithm, i.e., the zone where the solution is not possible to obtain for given accuracy and number of iterations. The identical color zones represent the same number of iterations utilized by different iteration schemes for finding the solution up to given accuracy, and their graphical view to the contour lines on the map is similar.

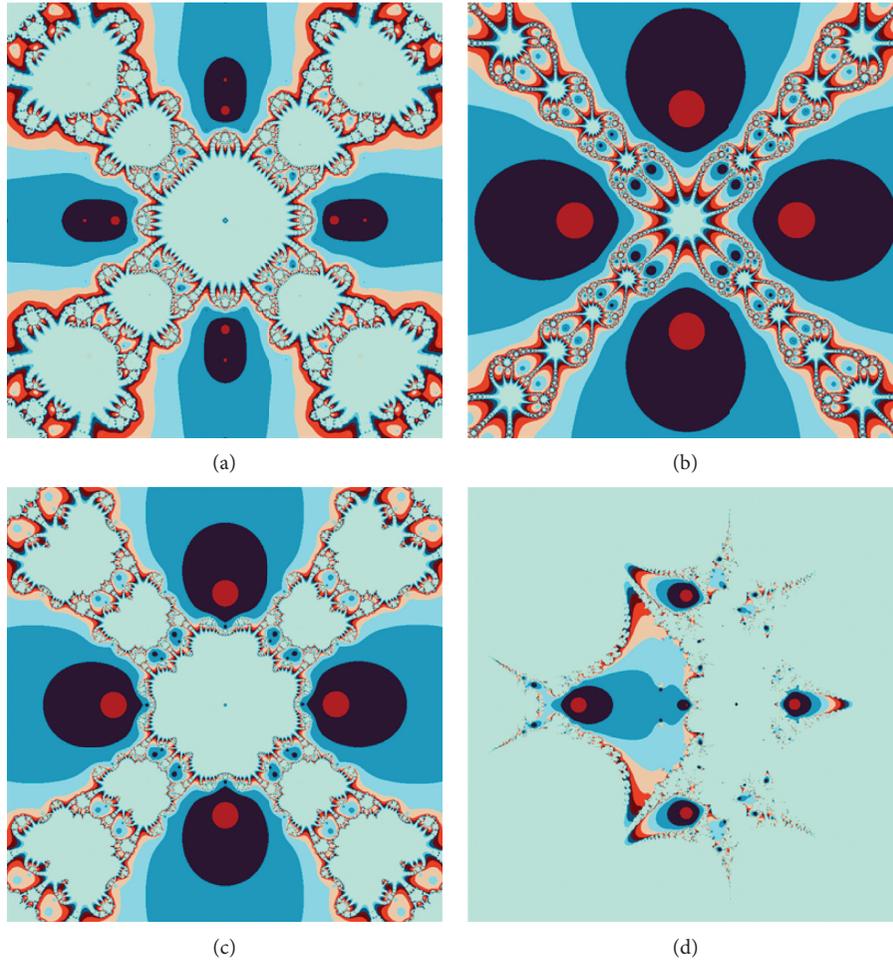


FIGURE 4: Polynomiographs for the quadratic-degree polynomial q_3 . (a) represents NRM, (b) represents TM, (c) represents ZM, and (d) represents Algorithm 1.

6. Concluding Remarks

In the present study, a new efficient and derivative-free algorithm for computing roots of nonlinear equations has been developed by employing the forward- and finite-difference schemes on Golbabai–Javidi’s. The convergence analysis of the suggested algorithm has been discussed and established that the suggested algorithm possesses the quartic-order convergence. To demonstrate the robust performance and validity of the presented algorithm, we assumed some arbitrary and engineering problems, converted them in the form of nonlinear functions, and then solved. The numerical results, contained in Tables 1–5, showed that the newly designed algorithm is superior to the other comparable algorithms with respect to different numerical terms such as convergence speed, accuracy, and computational order of convergence. The robust working of the new algorithm can also be judged by observing the accuracy of successive approximations which is much better against the other comparable ones. To study the complex dynamic nature of the suggested algorithm, polynomiographs have been drawn for some complex polynomials by utilizing the computer program Mathematica 12.0. The obtained graphical objects are novel and

aesthetically pleasing, revealing the better convergence speed and other graphical characteristics of the designed algorithm over the other comparable ones. Using the same idea of this study, one can obtain a new family of derivative-free algorithms for computing roots of nonlinear equations.

Data Availability

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

Conflicts of Interest

The authors do not have any conflicts of interest.

Authors’ Contributions

All authors contribute equally in this paper.

References

- [1] D. E. Kincaid and E. W. Cheney, *Numerical Analysis*, Brooks/Cole Publishing Company, Pacific Grove, CA, USA, 1990.

- [2] 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.
- [3] A. R. Alharbi, M. I. Faisal, F. A. Shah, M. Waseem, R. Ullah, and S. Sherbaz, "Higher order numerical approaches for nonlinear equations by decomposition technique," *IEEE Access*, vol. 7, pp. 44329-44337, 2019.
- [4] R. Behl and E. Martínez, "A new high-order and efficient family of iterative techniques for nonlinear models," *Complexity*, vol. 2020, Article ID 1706841, 11 pages, 2020.
- [5] D. Cebic, M. Paunovic, and N. M. Ralevic, "A variant of sharma-arora's optimal eighth-order family of methods for finding A simple root of nonlinear equation," in *Proceedings of the IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, September 2018.
- [6] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Higher-order root-finding algorithms and their basins of attraction," *Jurnal Matematika*, vol. 2020, Article ID 5070363, 11 pages, 2020.
- [7] M. A. Rehman, A. Naseem, and T. Abdeljawad, "Some novel sixth-order iteration schemes for computing zeros of nonlinear scalar equations and their applications in engineering," *Journal of Function Spaces*, vol. 2021, Article ID 5566379, 11 pages, 2021.
- [8] A. Naseem, M. A. Rehman, T. Abdeljawad, and Y.-M. Chu, "Novel iteration schemes for computing zeros of non-linear equations with engineering applications and their dynamics," *IEEE Access*, vol. 9, pp. 92246-92262, 2021.
- [9] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Numerical algorithms for finding zeros of nonlinear equations and their dynamical aspects," *Jurnal Matematika*, vol. 2020, Article ID 2816843, 11 pages, 2020.
- [10] 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, 2018.
- [11] 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, 2013.
- [12] J. F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea Publishing company, New York, NY, USA, 1982.
- [13] 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, 2012.
- [14] T. Zhanlav, O. Chuluunbaatar, and G. Ankhbayar, "On Newton-type methods with fourth and fifth-order convergence," *Discrete and Continuous Models and Applied Computational Science*, vol. 2, no. 2, pp. 30-35, 2010.
- [15] 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.
- [16] 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.
- [17] R. L. Fournier, *Basic Transport Phenomena in Biomedical Engineering*, Taylor & Francis, New York, NY, USA, 2007.
- [18] V. D. Waals and J. Diderik, *Over de Continuïteit van den Gasen Vloeïstoftoestand (on the continuity of the gas and liquid state)*, Ph.D. thesis, University of Illinois, Leiden, The Netherlands, 1873.
- [19] R. Manning, "On the flow of water in open channels and pipes," *Transactions of the Institution of Civil Engineers of Ireland*, vol. 20, pp. 161-207, 1891.
- [20] M. Planck, *The Theory of Heat Radiation*, P. Blakiston's Son & Co., New York, NY, USA, 2nd edition, 1914.
- [21] B. Kalantari, *Method of Creating Graphical Works Based on Polynomials*, 2005.
- [22] B. Kalantari, "Polynomiography: from the fundamental theorem of Algebra to art," *Leonardo*, vol. 38, no. 3, pp. 233-238, 2005.
- [23] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks/Cole Publishing Company, Pacific Grove, CA, USA, Sixth edition, 1997.
- [24] I. K. Argyros, "A note on the Halley method in Banach spaces," *Applied Mathematics and Computation*, vol. 58, pp. 215-224, 1993.
- [25] A. S. Householder, *The Numerical Treatment of a Single Non-linear Equation*, McGraw-Hill, New York, NY, USA, 1970.
- [26] K. Gdawiec, "Fractal patterns from the dynamics of combined polynomial root finding methods," *Nonlinear Dynamics*, vol. 90, no. 4, pp. 2457-2479, 2017.
- [27] 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.
- [28] I. Gosciniaik and K. Gdawiec, "Visual analysis of dynamics behaviour of an iterative method depending on selected parameters and modifications," *Entropy*, vol. 22, p. 734, 2020.
- [29] B. Kalantari, "An invitation to polynomiography via exponential series," 2017, <https://arxiv.org/abs/1707.09417>.
- [30] B. Kalantari and E. H. Lee, "Newton-Ellipsoid polynomiography," *Journal of Mathematics and the Arts*, vol. 13, no. 4, pp. 336-352, 2019.
- [31] W. Kotarski, K. Gdawiec, and A. Lisowska, "Polynomiography via ishikawa and mann iterations," *Advances in Visual Computing*, vol. 7431, pp. 305-313, 2012.
- [32] Y. C. Kwun, M. Tanveer, W. Nazeer, K. Gdawiec, and S. M. Kang, "Mandelbrot and julia sets via jungck-CR iteration with $\mathbb{S}\mathbb{S}$ -convexity," *IEEE Access*, vol. 7, pp. 12167-12176, 2019.
- [33] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Computational methods for non-linear equations with some real-world applications and their graphical analysis," *Intelligent Automation & Soft Computing*, vol. 30, no. 3, pp. 805-819, 2021.
- [34] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Numerical methods with engineering applications and their visual analysis via polynomiography," *IEEE Access*, vol. 9, pp. 99287-99298, 2021.
- [35] A. Naseem, M. A. Rehman, T. Abdeljawad, and Y.-M. Chu, "Some engineering applications of newly constructed algorithms for one-dimensional non-linear equations and their fractal behavior," *Journal of King Saud University Science*, vol. 33, no. 5, p. 101457, 2021.
- [36] 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.
- [37] M. Scott, B. Neta, and C. Chun, "Basin attractors for various methods," *Applied Mathematics and Computation*, vol. 218, no. 6, pp. 2584-2599, 2011.
- [38] 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.