Hindawi
Complexity
Volume 2021, Article ID 9234932, 9 pages
https://doi.org/10.1155/2021/9234932

Research Article

Some Real-Life Applications of a Newly Designed Algorithm for Nonlinear Equations and Its Dynamics via Computer Tools

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

[1]*Department of Mathematics, University of Management and Technology, Lahore 54770, Pakistan*
[2]*Department of Mathematics, Aden University, Khormaksar 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 August 2021; Accepted 11 October 2021; Published 22 December 2021

Academic Editor: Viet-Thanh Pham

In this article, we design a novel fourth-order and derivative free root-finding algorithm. We construct this algorithm by applying the finite difference scheme on the well-known Ostrowski's method. The convergence analysis shows that the newly designed algorithm possesses fourth-order convergence. To demonstrate the applicability of the designed algorithm, we consider five real-life engineering problems in the form of nonlinear scalar functions and then solve them via computer tools. The numerical results show that the new algorithm outperforms the other fourth-order comparable algorithms in the literature in terms of performance, applicability, and efficiency. Finally, we present the dynamics of the designed algorithm via computer tools by examining certain complex polynomials that depict the convergence and other graphical features of the designed algorithm.

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, mathematicians employed the excessive use of computers in different branches of Mathematics especially in the determination of approximated roots of the transcendental and nonlinear algebraic equations which have played an important role in different branches of computational and applied mathematics. In many engineering disciplines, a lot of problems exist which can be easily converted into nonlinear forms by employing different mathematical techniques. Analytical methods cannot find the solution needed for these problems, and therefore, we need iterative algorithms for solving out these problems. To execute an iterative algorithm, we always need a starting point (initial guess) which is refined after every iteration, and we find the approximated root up to the required accuracy after some finite iterations. The convergence rate and convergence order of an iterative algorithm are relied upon the selection of that starting point. Some of the most popular and ancient iterative algorithms are given in [1–8] and the references are cited therein. In the 15th century, Newton [1, 2] introduced a quadratic-order root-finding algorithm which has been used successfully for many years. Over the time, many experts worked on iterative algorithms and brought several modified versions of Newton's algorithm with higher-order convergence which involve predictor and corrector steps and are often referred to as multistep iterative algorithms. For more information, one can see [9–20] and the references are cited therein. In general, the convergence order of a multistep algorithm is higher because of predictor and corrector steps, but it results in a higher computational cost which is the downside of these algorithms. It is really difficult to handle the cost of computing and the convergence rate of an algorithm because these two terms are inversely proportional to each other.

Over the past few years, mathematicians have focused on the aforementioned issues and have tried to design some new iterative algorithms with higher convergence and low cost of computing by employing several mathematical methods. In [21], the authors introduced a new two-step Halley's method with sixth order convergence and then replaced its second derivative for reducing computing cost and proposed a new fifth order second derivative free algorithm. In [22], Hafiz and Al-Goria established a novel family of optimal eighth order

iterative algorithms and then studied their dynamics. In [23], the authors introduced seventh and ninth orders novel iterative algorithms with the help of the predictor-corrector technique and Simpson quadrature formulae. By employing the Newton interpolation technique along with weight functions, Salimi et al. [24] introduced a new family of eighth order optimal root-finding algorithms. In [25], the authors constructed some novel optimal iterative algorithms with higher convergence and demonstrated the applicability of the suggested methods by solving some engineering problems. Recently, Chu et al. [26] proposed a novel family iteration scheme and discussed the dynamics of the presented methods with the help of computer tools.

In the present research article, we introduce a new fourth-order and derivative free algorithm for solving engineering problems in the form of scalar nonlinear functions. The construction of this algorithm is based upon the finite difference scheme on Ostrowski's method. We also certify that the designed algorithm has fourth-order convergence. The designed algorithm is then applied to some real-world engineering problems for certifying its better performance and 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 the computer program Mathematica 12.0.

2. Main Results

Consider the nonlinear problem of the following form:

$$\psi(u) = 0, \tag{1}$$

where ψ is a real-valued function with an open interval domain.

Suppose that α is a root of (1) with u_0 as an initial guess near to the exact root α, then the implication of Taylor's series around u_0 for (1) gives us

$$\psi(u_0) - (u - u_0)\psi'(u_0) + \frac{(u - u_0)^2\psi''(u_0)}{2!} + \cdots = 0. \tag{2}$$

If $\psi'(u_0)$ is nonzero, then the above expression implies

$$u_{i+1} = u_i - \frac{\psi(u_i)}{\psi'(u_i)}, \tag{3}$$

which is Newton's root-finding algorithm [1, 2] for scalar nonlinear functions.

By taking it as a predictor, Ostrowski designed the following two-step iterative algorithm:

$$\begin{aligned} v_i &= u_i - \frac{\psi(u_i)}{\psi'(u_i)}, \\ u_{i+1} &= v_i - \frac{\psi(v_i)\psi(u_i)}{\psi'(u_i)[\psi(u_i) - 2\psi(v_i)]}, \end{aligned} \tag{4}$$

which is well-known Ostrowski's root-finding algorithm [11] for scalar nonlinear functions.

By including Newton's algorithm, the above two-step method may be converted to three-step in the following form:

$$\begin{aligned} v_i &= u_i - \frac{\psi(u_i)}{\psi'(u_i)}, \\ w_i &= v_i - \frac{\psi(v_i)}{\psi'(v_i)}, \\ u_{i+1} &= w_i - \frac{\psi(w_i)\psi(v_i)}{\psi'(v_i)[\psi(v_i) - 2\psi(w_i)]}, \end{aligned} \tag{5}$$

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

$$\psi'(u_i) = \frac{\psi(u_i + \psi(u_i))}{\psi(u_i)} = g(u_i). \tag{6}$$

To approximate $\psi'(v)$, we utilize the finite difference scheme as

$$\psi'(v_i) = \frac{\psi(v_i) - \psi(u_i)}{v_i - u_i} = h(u_i, v_i). \tag{7}$$

Using (6) and (7) in (5), we can write Algorithm 1.

Algorithm 1. For a given u_0, compute the approximate solution u_{i+1} by the following iterative schemes

$$\begin{aligned} &v_i = u_i - \psi(u_i)/g(u_i), i = 0, 1, 2, \ldots, \\ &w_i = v_i - \psi(v_i)/h(u_i, v_i), \\ &u_{i+1} = w_i - \psi(w_i)\psi(v_i)/h(u_i, v_i)[\psi(v_i) - 2\psi(w_i)] \end{aligned}$$

Algorithm 1 is a new iteration scheme for calculating the approximated roots of scalar nonlinear equations and needs only four evaluations per iteration. The main characteristic of the suggested algorithm is that it is derivative free and easily applicable to all those scalar functions whose derivatives become undefined within the domain. In this sense, the proposed algorithm's computing cost is minimal which results in a higher efficiency index.

3. Convergence Analysis

In the present section, we shall discuss the convergence criterion of the newly designed algorithm, i.e., Algorithm 1.

Theorem 1. *Suppose that α is the root of the equation $\psi(u) = 0$. If $\psi(u)$ is sufficiently smooth in the neighborhood of α, then the order of convergence of Algorithm 1 is at least four.*

Table 1: Comparison among different fourth-order algorithms.

Methods	IT	u_{i+1}	$\lvert\psi(u_{i+1})\rvert$	$\sigma = \lvert u_{i+1} - u_i\rvert$	η
OM	04	0.3426482058114499	$9.186801e^{-17}$	$2.728697e^{-05}$	4
TM	05	0.3426482058114499	$4.236557e^{-23}$	$1.848727e^{-14}$	4
ZM	05	0.3426482058114499	$7.675869e^{-25}$	$1.516654e^{-07}$	4
Algorithm 1	04	0.3426482058114499	$5.362719e^{-18}$	$1.131404e^{-06}$	4

Proof. To analyze the convergence criterion of the iteration scheme (1), we assume that α is a root of equation $\psi(u) = 0$ and e_i be the error at i^{th} iteration; then, $e_i = u_i - \alpha$, and by using Taylor's series expansion, we have

$$\begin{aligned}\psi(u_i) &= \psi'(\alpha)e_i + \frac{1}{2!}\psi''(\alpha)e_i^2 + \frac{1}{3!}\psi'''(\alpha)e_i^3 + \frac{1}{4!}\psi^{(iv)}(\alpha)e_i^4 + O\left(e_i^5\right),\\ \psi(u_i) &= \psi'(\alpha)\left[e_i + d_2e_i^2 + d_3e_i^3 + d_4e_i^4 + O\left(e_i^5\right)\right],\end{aligned} \tag{8}$$

where

$$\begin{aligned}d_i &= \frac{1}{i!}\frac{\psi^{(i)}(\alpha)}{\psi'(\alpha)},\\ g(u_i) &= \psi'(\alpha)\left[1 + 3d_2e_i + \left(7d_3 + d_2^2\right)e_i^2 + \left(6d_2d_3 + 15d_2^4\right)e_i^3 + \left(18d_2d_4 + 31d_5 + d_3d_2^2 + 5d_3^2\right)e_i^4 + O\left(e_i^5\right)\right].\end{aligned} \tag{9}$$

With the help of equations (8) and (9), we get

$$v_i = \alpha + 2d_2e_i^2 + \left(6d_3 - 5d_2^2\right)e_i^3 + \left(14d_4 - 26d_3d_2 + 13d_2^3\right)e_i^4 + O\left(e_i^5\right), \tag{10}$$

$$\psi(v_i) = \psi'(\alpha)\left[2d_2e_i^2 + \left(6d_3 - 5d_2^2\right)e_i^3 + \left(14d_4 - 26d_3d_2 + 13d_2^3\right)e_i^4 + O\left(e_i^5\right)\right], \tag{11}$$

$$h(u_i, v_i) = \psi'(\alpha)\left[1 + d_2e_i + \left(d_3 + d_2^2\right)e_i^2 + \left(8d_2d_3 - 5d_2^3 + d_4\right)e_i^3 + \left(13d_2^4 - 27d_3d_2^2 + 16d_4d_2 + +d_5 + 6d_3^2\right)e_i^4 + O\left(e_i^5\right)\right], \tag{12}$$

$$w_i = \alpha + 2d_2e_i^3 + \left(8d_2d_3 - 7d_2^3\right)e_i^4 + O\left(e_i^5\right), \tag{13}$$

$$\psi(w_i) = \psi'(\alpha)\left[2d_2e_i^3 + \left(8d_2d_3 - 7d_2^3\right)e_i^4 + O\left(e_i^5\right)\right]. \tag{14}$$

Using equations (8)–(14) in Algorithm 1 gives us the following equality:

$$u_{i+1} = \alpha - 2d_2^3e_i^4 + O\left(e^5\right), \tag{15}$$

which implies that

$$e_{i+1} = -2d_2^3e_i^4 + O\left(e^5\right). \tag{16}$$

The above equation shows that the designed algorithm is of fourth-order convergence. □

4. Real-Life Applications

In this section, we take five real-world problems in the form of scalar nonlinear functions to exhibit the applicability, validity, and efficiency of the newly designed fourth-order algorithm. We compare it with other well-known fourth-order algorithms, namely, Ostrowski's method (OM) [11], Traub's method (TM) [12], and Zhanlav method (ZM) [27].

Example 1. Fluid permeability problem:

The hydraulic permeability is actually the measurement of the flow resistance. It relates the pressure gradient to fluid

velocity and may be expressed as

$$\kappa = \frac{r_e u^3}{20(1-u)^2}, \qquad r_e u^3 - 20k(1-u)^2 = 0, \tag{17}$$

where κ denotes the specific hydraulic permeability, r_e stands for the radius, and $0 \le u \le 1$ is the porosity. For further details see [28] and the reference cited therein. By taking the values of $r_e = 100$ and $\kappa = 0.4655$ in (14), we obtain the above problem in the following nonlinear function:

$$\psi_1(u) = 100u^3 - 9.31(1-u)^2. \tag{18}$$

To solve ψ_1, the initial guess has been chosen as $u_0 = 2.0$ for starting the iteration process, and the results are given in Table 1.

Example 2. Blood rheology model.

Blood rheology is a branch of science that works to study the physical and flow properties of blood [29]. 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

$$H = 1 - \frac{16}{7}\sqrt{u} + \frac{4}{3}u - \frac{1}{21}u^4, \tag{19}$$

where flow rate reduction is computed by H. Using $H = 0.40$ in (19), we have

$$\psi_2(u) = \frac{1}{441}u^8 - \frac{8}{63}u^5 - 0.05714285714u^4 + \frac{16}{9}u^2 - 3.624489796u + 0.3. \tag{20}$$

To solve ψ_2, the initial guess has been chosen as $u_0 = 0.9$ for starting the iteration process, and the results are given in Table 2.

Example 3. Van Der Wall's equation.

The well-known equation for examining the behaviour of real and ideal gas was introduced by Van Der Wall's [30], with the following expression:

$$\left(P + \frac{C_1 n^2}{V^2}\right)(V - nC_2) = iRT. \tag{21}$$

Equation (21) may be easily transformed into the following nonlinear function by taking the particular values of the parameters:

$$\psi_3(u) = 0.986u^3 - 5.181u^2 + 9.067u - 5.289, \tag{22}$$

where u is the gas volume that may be easily determined by solving ψ_2. Because the polynomial's degree is three, it must have three roots. There is only one positive real root 1.9298462428 among these which is physically possible since the gas volume can never be negative. To solve ψ_3, the initial guess has been chosen as $u_0 = 1.0$ for starting the iteration process, and the results are given in Table 3.

Example 4. Plank's radiation law.

The energy density within the black isothermal body is calculated using Planck's radiation law [31] given as follows:

$$\varphi(\gamma) = \frac{8\pi Pc}{\gamma^5\left(e^{Pc/\gamma Tk} - 1\right)}. \tag{23}$$

Suppose we want to calculate wavelength γ_1 for the peak value of the energy density $\varphi(\gamma_1)$. To transform (23) in nonlinear form, we assume $u = Pc/\gamma Tk$ and obtain the following nonlinear expression:

$$\psi_4(u) = -1 + \frac{u}{5} + e^{-u}. \tag{24}$$

One of the estimated roots of ψ_4 is -0.0000000000000000 which represents the maximum amount of the wavelength of the radiation. To solve ψ_4, the initial point has been chosen as $u_0 = -2.0$ for starting the iteration process, and the results are given in Table 4.

Example 5. The problem of beam designing.

In Physics and Engineering sciences, the beam designing problem [32] regarding the embedment u of a sheet pile wall in the form of scalar nonlinear function is expressed as

$$\psi_5(u) = \frac{u^3 + 2.87u^2 - 4.62u - 10.28}{4.62}. \tag{25}$$

To solve ψ_5, the initial guess has been chosen as $u_0 = 3.0$ for starting the iteration process, and the results are given in Table 5.

Here, we choose the accuracy $\varepsilon = 10^{-15}$ in the following stopping criterion of the computer program:

$$\left|u_{i+1} - u_i\right| < \varepsilon. \tag{26}$$

We used the computer application Maple 13 to solve all numerical problems.

Tables 1–5 exhibit the numerical comparison of the designed fourth-order algorithm with Ostrowski's method (OM), Traub's method (TM), and Zhanlav's method (ZM). In the columns of the above tables, ITstands for the number of iterations, $|\psi(u)|$ indicates the positive value of the function $\psi(u)$, u_{i+1}indicates the estimated root, σ indicates the absolute difference of the consecutive estimations u_{i+1},

Table 2: Comparison among different fourth-order algorithms.

Methods	IT	u_{i+1}	$\lvert\psi(u_{i+1})\rvert$	$\sigma = \lvert u_{i+1} - u_i\rvert$	η
OM	03	0.0864335580522918	$2.960661e^{-18}$	$4.952936e^{-05}$	4
TM	03	0.0864335580522916	$3.697508e^{-16}$	$1.642942e^{-04}$	4
ZM	12	0.0864335580522918	$1.957052e^{-25}$	$5.571776e^{-07}$	4
Algorithm 1	03	0.0864335580522917	$1.940272e^{-17}$	$3.210544e^{-06}$	4

Table 3: Comparison among different fourth-order algorithms.

Methods	IT	u_{i+1}	$\lvert\psi(u_{i+1})\rvert$	$\sigma = \lvert u_{i+1} - u_i\rvert$	η
OM	09	1.9298462428478622	$1.626245e^{-22}$	$1.858593e^{-06}$	4
TM	10	1.9298462428478622	$4.578052e^{-27}$	$1.235520e^{-07}$	4
ZM	44	1.9298462428478622	$1.365062e^{-19}$	$6.456152e^{-06}$	4
Algorithm 1	06	1.9298462428478622	$1.335843e^{-29}$	$3.841021e^{-10}$	4

Table 4: Comparison among different fourth-order algorithms.

Methods	IT	u_{i+1}	$\lvert\psi(u_{i+1})\rvert$	$\sigma = \lvert u_{i+1} - u_i\rvert$	η
OM	04	−0.0000000000000000	$1.030994e^{-51}$	$3.261218e^{-13}$	4
TM	04	−0.0000000000000000	$1.706136e^{-36}$	$1.719178e^{-09}$	4
ZM	04	−0.0000000000000000	$1.383001e^{-20}$	$1.153483e^{-05}$	4
Algorithm 1	04	−0.0000000000000000	$2.759067e^{-45}$	$6.858899e^{-14}$	4

Table 5: Comparison of different fourth-order algorithms.

Methods	IT	u_{i+1}	$\lvert\psi(u_{i+1})\rvert$	$\sigma = \lvert u_{i+1} - u_i\rvert$	η
OM	03	2.0021187789538273	$1.247083e^{-31}$	$2.494230e^{-08}$	4
TM	03	2.0021187789538273	$1.271632e^{-29}$	$7.400791e^{-08}$	4
ZM	03	2.0021187789538273	$1.334997e^{-21}$	$5.297174e^{-06}$	4
Algorithm 1	03	2.0021187789538273	$1.760330e^{-28}$	$1.073092e^{-09}$	4

and u_i and η represent the approximated computational order of convergence given as

$$\eta \approx \frac{\mathrm{In}\left(\left|u_{i+1} - \alpha\right|/\left|u_i - \alpha\right|\right)}{\ln\left(\left|u_i - \alpha\right|/\left|u_{i-1} - \alpha\right|\right)}, \tag{27}$$

which was introduced by Weerakoon and Fernando [33].

5. Dynamical Analysis via Computer Technology

In this section, we give a detailed graphical comparison of the newly designed fourth-order algorithm with the other fourth-order algorithms via computer technology by considering some complex polynomials in the form of polynomiographs. A polynomiograph is a graphical object generated in a process known as polynomiography, introduced by Dr. Bahman Kalantri in 2005 [34]. It is defined as "the algorithmic visualization of polynomial equations by employing different iterative techniques" [35].

To draw dynamics by employing computer technology using various iterative algorithms, an initial rectangle $\mathscr{R}$ which includes the root of the investigated complex polynomial has been chosen. Then, for every point w_0 in $\mathscr{R}$, we perform the process of iteration. The image's quality is usually correlated with the discretization of$\mathscr{R}$, i.e., if the rectangle $\mathscr{R}$ has been discretized into a 2000×2000 grid, then the quality of the produced image will be better.

Typically, the colors of produced polynomiographs are fully associated with the number of iterations required to find the approximated roots with a given precision and a selected iterative algorithm. The main algorithm for the production of a polynomiograph is given in Algorithm 1.

A stopping criterion is always required for an iterative algorithm that includes the repetition of steps, since it informs us about the convergence or divergence of the investigated iterative algorithm. Such a criterion is commonly referred to as a convergence test with the following mathematical expression:

$$\left|w_{i+1} - w_i\right| < \varepsilon, \tag{28}$$

Input: $q \in \mathbb{C}$—polynomial, $\mathbb{A} \subset \mathbb{C}$—area, K—maximum No. of iterations, I—iterative algorithm, ε—accuracy, colormap $[0 \ldots C-1]$—colormap with C colors
Output: polynomiograph for the complex polynomial q in the area $\mathbb{A}$
for $w_0 \in \mathbb{A}$, **do**
  $i = 0$
  while $i \leq K$ do
    $w_{i+1} = I(w_i)$
      if $|w_{i+1} - w_i| < \epsilon$, then
        break
    $i = i + 1$
  color $\mathbf{w}_0$ via colormap

Algorithm 1: Polynomiograph's generation.

where w_{i+1} and w_i denote the successive iterations, and $\varepsilon > 0$ stands for the accuracy in the stopping criterion. The convergence test $(w_{i+1}, w_i, \varepsilon)$ is considered TRUE if the iterative algorithm under consideration is converged and FALSE if it is diverged. The abovementioned stopping criterion (28) is also used in this study. The variety of polynomiographs' colors is correlated with the performed iterations to find out the root with given precision ε. Using various iterative algorithms, a variety of aesthetically pleasant polynomiographs can be produced by altering the parameter K, where K specifies the upper limit of the number of iterations. For further information regarding polynomiography along with its applications in different fields, one can see [36–44] and the references cited therein.

For drawing polynomiographs through different iterative algorithms, we consider the following four complex polynomials:

$$\begin{aligned} q_1(w) &= w^3 - 1, \\ q_2(w) &= \left(w^3 - 1\right)^2, \\ q_3(w) &= w^4 - 1, \\ q_4(w) &= \left(w^4 - 1\right)^2. \end{aligned} \tag{29}$$

The colormap used for the coloring of iterations in the generation of polynomiographs is shown in Figure 1:

Example 6. Polynomiographs for the polynomial q_1 through different fourth-order algorithms.

In the first example, we consider a cubic-degree polynomial $q_1(w) = w^3 - 1$, having three distinct roots 1, $-1/2 + \sqrt{3}/2i$, and $-1/2 - \sqrt{3}/2i$.We used a computer program to run all the methods to get the simple roots of the under consideration polynomial q_1, and the results are shown in Figure 2.

Example 7. Polynomiographs for the polynomial q_2 through different fourth-order algorithms.

In the second example, we take a sextic-degree polynomial $q_2(w) = (w^3 - 1)^2$, which has three unique roots 1, $-1/2 + \sqrt{3}/2i$, and $-1/2 - \sqrt{3}/2i$ with multiplicity two. We perform the process of iteration for all iterative algorithms for drawing polynomiographs, and the results are shown in Figure 3.

Example 8. Polynomiographs for the polynomial q_3 through different fourth-order algorithms.

In the third example, we consider a quartic-degree polynomial $q_3(w) = w^4 - {}^1$, which has four unique roots 1, -1, i, and $-i$. We created the graphical objects by executing all iterative algorithms, and the results are shown in Figure 4.

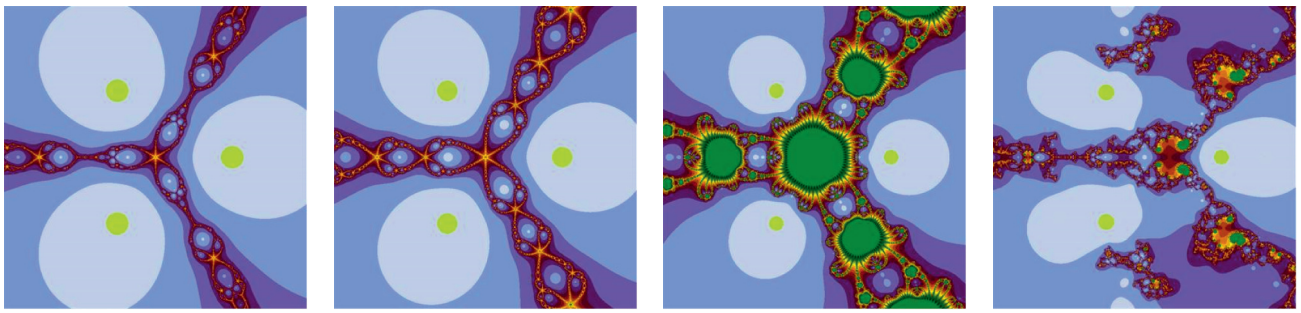
Example 9. Polynomiographs for the polynomial q_4 through different fourth-order algorithms.

In the fourth example we take an eighth-degree complex polynomial $q_4(w) = (w^4 - 1)^2$ with four unique roots 1, -1, i, and $-i$ of multiplicity two. We used a computer program to run all methods for drawing polynomiographs, and the results in the form of visually attractive pictures are shown in Figure 5.

In above examples, we compared the developed algorithm to various fourth-order iterative algorithms using a computer program by taking into account different degrees complex polynomials. Two key features may be identified from the produced graphics. The first is the iteration scheme's speed of convergence, and the second feature is the iteration scheme's dynamics. Low dynamics are seen in places with little color variation, and high dynamics are found in areas with a lot of color variety. The black coloring in the graphics denotes areas where the solution cannot be reached in the specified number of iterations. The darker zone in the above-presented pictures indicates that the iterative algorithm under consideration requires fewer iterations for finding the solution of the given problem. The same-colored regions in the graphical objects represent the same number of iterations necessary to find the required solution with the given accuracy. Note that the polynomiographs created using our proposed iterative algorithm have considerably brighter and darker regions and no black areas as compared to other similar order algorithms in the literature. Furthermore, the polynomiographs of the proposed iterative algorithm show larger convergence areas than the other comparable techniques which demonstrate the better efficiency of the suggested algorithm.

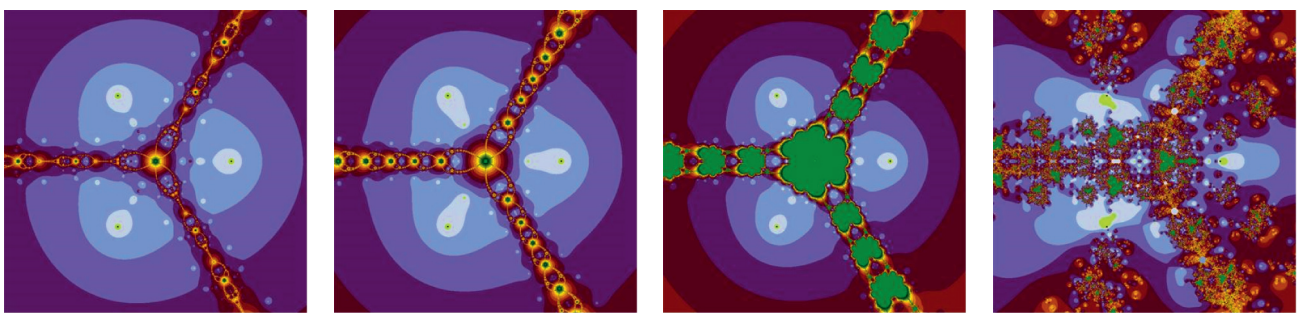
We drew all graphical objects with the computer program Mathematica 12.0 by using the values of parameters as $\varepsilon = 0.001$ and $K = 20$, where ε and K indicate the accuracy and the upper bound of the number of iterations, respectively.

Figure 1: The colormap used for generating polynomiographs.



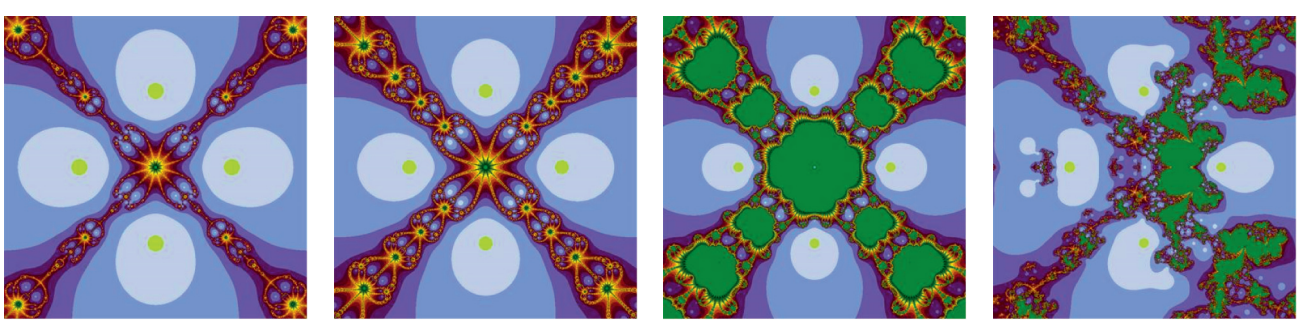
(a) (b) (c) (d)

Figure 2: Polynomiographs related to the complex polynomial q_1. (a) Ostrowski's method. (b) Traub's method. (c) Zhanlav's method. (d) Algorithm 1.



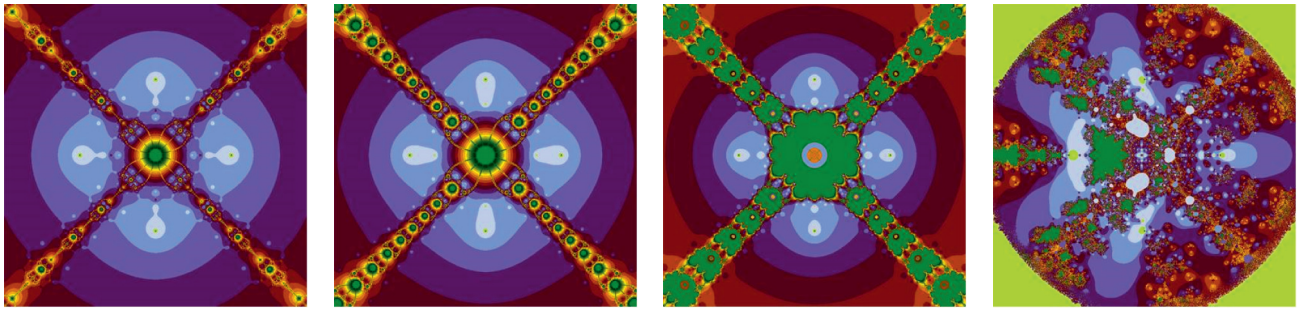
(a) (b) (c) (d)

Figure 3: Polynomiographs related to the complex polynomial q_2. (a) Ostrowski's method. (b) Traub's method. (c) Zhanlav's method. (d) Algorithm 1.



(a) (b) (c) (d)

Figure 4: Polynomiographs related to the complex polynomial q_3. (a) Ostrowski's method. (b) Traub's method. (c) Zhanlav's method. (d) Algorithm 1.



(a) (b) (c) (d)

Figure 5: Polynomiographs related to the complex polynomial q_4. (a) Ostrowski's method. (b) Traub's method. (c) Zhanlav's method. (d) Algorithm 1.

6. Conclusion

By employing the finite difference scheme on Ostrowski's method, we designed a new derivative free algorithm for calculating the approximate zeros of nonlinear scalar equations that possesses the fourth-order convergence. To analyze the applicability of the designed algorithm, we took some real-life engineering problems and solved them via computer tools. The numerical results given in Tables 1–5 proved the better performance and applicability of the designed algorithm against the other fourth-order algorithms. We have also presented the dynamics of the designed algorithm and gave a detailed comparison with the other comparable fourth-order algorithms in the literature via computer tools that revealed the convergence and other graphical characteristics of the designed algorithm. A new family of derivative free root-finding algorithms can be constructed by applying the finite difference scheme to the existing methods in the literature.

Data Availability

The data used to support this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed equally to this study.

References

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

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

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

[4] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks/Cole Publishing Company, Pacific Grove, CA, USA, 6th edition, 1997.

[5] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, NY, USA, 3rd edition, 2002.

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

[7] I. K. Argyros, "A note on the Halley method in Banach spaces," *Applied Mathematics and Computation*, vol. 58, pp. 215–224, 1993.

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

[9] 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.

[10] 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.

[11] A. M. Ostrowski, *Solution of Equations and Systems of Equations*, Academic Press, Cambridge, MA, USA, 2nd edition, 1966.

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

[13] J. Kuo, "The improvements of modified Newton's method," *Applied Mathematics and Computation*, vol. 189, no. 1, pp. 602–609, 2007.

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

[15] 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.

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

[17] F. Ahmad, E. Tohidi, M. Z. Ullah, and J. A. Carrasco, "Higher order multi-step Jarratt-like method for solving systems of nonlinear equations: application to PDEs and ODEs," *Computers & Mathematics with Applications*, vol. 70, no. 4, pp. 624–636, 2015.

[18] 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, p. 11, Article ID 5566379, 2021.

[19] F. Ahmad, E. Tohidi, and J. A. Carrasco, "A parameterized multi-step Newton method for solving systems of nonlinear equations," *Numerical Algorithms*, vol. 71, no. 3, pp. 631–653, 2016.

[20] 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, Article ID 101457, 2021.

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

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

[23] M. A. Hafiz and S. M. H. Al-Goria, "New ninth and seventh order methods for solving nonlinear equations," *European Scientific Journal*, vol. 8, no. 27, pp. 83–95, 2012.

[24] M. Salimi, N. M. A. Nik Long, S. Sharifi, and B. A. Pansera, "A multi-point iterative method for solving nonlinear equations with optimal order of convergence," *Japan Journal of Industrial and Applied Mathematics*, vol. 35, no. 2, pp. 497–509, 2018.

[25] O. S. Solaiman and I. Hashim, "Efficacy of optimal methods for nonlinear equations with chemical engineering applications," *Mathematical Problems in Engineering*, vol. 2019, Article ID 1728965, 11 pages, 2019.

[26] Y. Chu, N. Rafiq, M. Shams, S. Akram, N. A. Mir, and H. Kalsoom, "Computer methodologies for the comparison of some efficient derivative free simultaneous iterative methods for finding roots of non-linear equations," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 275–290, 2021.

[27] 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, pp. 30–35, 2010.

[28] W. M. Saltzman, *Drug Delivery: Engineering Principal for Drug Therapy*, Oxford University Press, New York, NY, USA, 2001.

[29] R. L. Fournier, *Basic Transport Phenomena in Biomedical Engineering*, pp. 1–611, Taylor & Francis, New York, NY, USA, 2007.

[30] V. D. Waals and J. Diderik, "Over de continuiteit van den gasenvloeistoftoestand (on the continuity of the gas and liquid state)," Ph.D. thesis, Leiden University, Leiden, The Netherlands, 1873.

[31] M. Planck, "The theory of heat radiation," in *Translated by Masius, M*, P. Blakiston's Son & Co., Philadelphia, PA, USA, 2nd ed. edition, 1914.

[32] M. Shams, N. Ahmad Mir, N. Rafiq, A. O. Almatroud, and S. Akram, "On dynamics of iterative techniques for nonlinear equation with applications in engineering," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5853296, 17 pages, 2020.

[33] 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.

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

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

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

[37] 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.

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

[39] A. Naseem, M. A. Rehman, and T. Abdeljawad, "Higher-order root-finding algorithms and their basins of attraction," *Journal of Mathematics*, vol. 2020, Article ID 5070363, 11 pages, 2020.

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

[41] 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.

[42] 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.

[43] 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.

[44] B. Neta, M. Scott, and C. Chun, "Basins of attraction for several methods to find simple roots of nonlinear equations," *Applied Mathematics and Computation*, vol. 218, no. 21, pp. 10548–10556, 2012.