J. Math. Computer Sci., 28 (2023), 52–59

Online: ISSN 2008-949X

Journal of Mathematics and Computer Science

Journal Homepage: www.isr-publications.com/jmcs

New twelfth order iterative method for solving nonlinear equations and their dynamical aspects

Pairat Janngam[a], Chalermwut Comemuang[b,*]

[a]Department of Mathematics, Faculty of Education, Buriram Rajabhat University, Buriram, Thailand.

[b]Department of Mathematics, Faculty of Science, Buriram Rajabhat University, Buriram, Thailand.

Abstract

The aims of this paper are to present new twelfth order iterative methods for solving nonlinear equations and one of them is second derivative free which has been removed using the interpolation technique. Analysis of convergence finalized that the order of convergence is twelfth. Some numerical examples illustrate that the algorithm is more efficient and performs better than other methods with the same order. In the end, we present the basins of attraction using some complex polynomials of different degrees to observe the fractal behavior and dynamical aspects of the proposed algorithms.

Keywords: Iterative method, non-linear equation, convergence analysis.

2020 MSC: 65H04, 65H05, 65H20.

1. Introduction

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

$$f(x) = 0.$$

In this literature of finding a root of non-linear Newton's method (NR) [14] is one of the well known optimal methods to obtain the zero of a non-linear equation

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

After that, the iteration of the fourth order convergence was presented by Shengfeng Li [7]:

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

*Corresponding author

Email address: chalermwut.cm@bru.ac.th (Chalermwut Comemuang)

doi: 10.22436/jmcs.028.01.05

Received: 2021-09-30 *Revised:* 2022-03-11 *Accepted:* 2022-03-18

The iteration of Householder's [3] with the third order convergence is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)f''(x_n)}{2f'^3(x_n)}.$$

Recently, there are many numerical iterative methods have been developed to solve these problems. These methods are constructed by using several different techniques, such as Taylor series, quadrature formulas, homotopy perturbation technique and its variant forms, decomposition technique, variational iteration technique, and Predictor-corrector technique. For more details, see [5, 6, 9–13].

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

2. The twelfth-order method

In this section we define new twelfth order iterative methods for solving nonlinear equation. In order to construct new twelfth order methods, we use well known fourth order iterative methods, presented by Shengfeng Li [7] and Householder's [3].

Algorithm 2.1. For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

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

Algorithm 2.2. For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)f''(x_n)}{2f'^3(x_n)}.$$

This is known as Householder's method, which has cubic convergence [3].

We have suggested the following three-step method, using Algorithm 2.1 method as predictor and Algorithm 2.2 as a corrector.

Algorithm 2.3. For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad z_n = x_n - \frac{(f(x_n) - f(y_n))f(x_n)}{(f(x_n) - 2f(y_n))f'(x_n)}, \qquad x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} - \frac{f(z_n)f''(z_n)}{2f'^3(z_n)}.$$

In order to implement this method, one has to find the second derivative of this function, which may create some problems. To overcome this drawback, we use new and different technique to reduce second derivative of the function into the first derivative. This idea plays a significant role in developing some new iterative methods free from second derivatives. To be more precise, we consider

$$f''(z_n) = \frac{2}{z_n - y_n}\left(2f'(z_n) + f'(y_n) - 3\frac{f(z_n) - f(y_n)}{z_n - y_n}\right) = d.$$

We suggest the following new iterative method for solving the nonlinear equation and this is the new motivation of higher-order.

Algorithm 2.4. For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad z_n = x_n - \frac{(f(x_n)-f(y_n))f(x_n)}{(f(x_n)-2f(y_n))f'(x_n)}, \qquad x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} - \frac{f(z_n)d}{2f'^3(z_n)},$$

which is a new two-step iterative method free from second derivative.

For the method defined by Algorithm 2.4, we have the following analysis of convergence.

Theorem 2.5. *Suppose that α is a root of the equation $f(x) = 0$. If $f(x)$ is sufficiently smooth in the neighborhood of α, then the order of convergence of Algorithm 2.4 is twelve.*

Proof. To analyze the convergence of Algorithm 2.4, suppose that α is a root of the equation $f(x) = 0$ and e_n be the error at nth iteration, then $e_n = x_n - \alpha$, and by using Taylor series expansion, we have

$$f(x_n) = f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + c_7e_n^7 + \cdots], \tag{2.1}$$

$$f'(x_n) = f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + 7c_7e_n^6 + \cdots], \tag{2.2}$$

where $c_n = \dfrac{f^{(n)}(\alpha)}{n!f'(\alpha)}$. With the help of equations (2.1) and (2.2), we get

$$\begin{aligned}
\frac{f(x_n)}{f'(x_n)} =& e_n - c_2e_2^2 - (2c_3 - 2c_2^2)e_n^3 - (3c_4 - 7c_2c_3 + 4c_2^3)e_n^4 \\
&+ 2(5c_2c_4 + 4c_2^4 - 10c_2^2c_3 + 3c_3^2 - 2c_5)e_n^5 + \cdots, \\
y_n =& \alpha + c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 - (3c_4 - 7c_2c_3 + 4c_2^3)e_n^4 \\
&+ (-8c_2^4 + 20c_2^2c_3 - 10c_2c_4 - 6c_3^2 + 4c_5)e_n^5 + \cdots, \\
f(y_n) =& f'(\alpha)[c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + (5c_2^3 - 7c_2c_3 + 3c_4)e_n^4 \\
&+ (-12c_2^4 + 24c_2^2c_3 - 10c_2c_4 - 6c_3^2 + 4c_5)e_n^5 + \cdots], \\
f'(y_n) =& f'(\alpha)[1 + 2c_2^2e_n^2 + 4(c_3c_2 - c_2^3)e_n^3 + (6c_2c_4 - 11c_3c_2^2 + 8c_2^4)e_n^4 \\
&+ (-16c_2^5 + 28c_2^3c_3 - 20c_2^2c_4 + 8c_2c_5)e_n^5 + \cdots].
\end{aligned} \tag{2.3}$$

Using equations (2.1)-(2.3), we get

$$\begin{aligned}
z =& \alpha + (c_2^3 - c_2c_3)e_n^4 + (-4c_2^4 + 8c_2^2c_3 - 2c_2c_4 - 2c_3^2)e_n^5 \\
&+ (10c_2^5 - 30c_2^3c_3 + 12c_2^2c_4 + 18c_2c_3^2 - 3c_2c_5 - 7c_3c_4)e_n^6 + \cdots, \\
f(z_n) =& f'(\alpha)[(c_2^3 - c_2c_3)e_n^4 + (-4c_2^4 + 8c_2^2c_3 - 2c_2c_4 - 2c_3^2)e_n^5 \\
&+ (10c_2^5 - 30c_2^3c_3 + 12c_2^2c_4 + 18c_2c_3^2 - 3c_2c_5 - 7c_3c_4)e_n^6 \cdots], \\
f'(z_n) =& f'(\alpha)[1 + (2c_2^4 - 2c_2^2c_3)e_n^4 + (-8c_2^5 + 16c_2^3c_3 - 4c_2^2c_4 - 4c_2c_3^2)e_n^5 \\
&+ (20c_2^6 - 60c_2^4c_3 + 24c_2^3c_4 + 36c_2^2c_3^2 - 6c_2^2c_5 - 14c_2c_3c_4)e_n^6 + \cdots], \\
d =& f'(\alpha)[2c_2 + (6c_2^3c_3 - 2c_2^2c_4 - 6c_2c_3^2)e_n^4 + (-24c_2^4c_3 + 8c_2^3c_4 + 48c_2^2c_3^2 - 20c_2c_3c_4 - 12c_3^3)e_n^5 \\
&+ (60c_2^5c_3 - 20c_2^4c_4 - 180c_2^3c_3^2 - 4c_2^3c_5 + 112c_2^2c_3c_4 + 108c_2c_3^3 - 18c_2c_3c_5 \\
&- 12c_2c_4^2 - 50c_3^2c_4)e_n^6 + \ldots].
\end{aligned} \tag{2.4}$$

Using equations (2.4), we get

$$x_{n+1} = \alpha + \left(2c_2^{11} - 7c_2^9c_3 + c_2^8c_4 + 9c_2^7c_3^2 - 2c_2^6c_3c_4 - 5c_2^5c_3^3 + c_2^4c_3^2c_4 + c_2^3c_3^4\right)e_n^{12} + O(e_n^{13}),$$

which implies that

$$e_{n+1} = \left(2c_2^{11} - 7c_2^9c_3 + c_2^8c_4 + 9c_2^7c_3^2 - 2c_2^6c_3c_4 - 5c_2^5c_3^3 + c_2^4c_3^2c_4 + c_2^3c_3^4\right)e_n^{12} + O(e_n^{13}).$$

The above equation shows that the order of convergence of Algorithm 2.4 is twelve. □

3. Numerical Examples

In this section, we include some nonlinear functions to illustrate the efficiency of our newly developed numerical algorithms. We compare these algorithms with Ahmad et al. (AHM) [2], Liu and Wang (LIU) [8], and Yasir and Abdul-Hassan (YAS) [1]. For this purpose, the following numerical examples have been solved:

$$\begin{aligned}
f_1(x) &= x^2 - e^x - 3x + 2,\ x_0 = 3.6,\\
f_2(x) &= x^3 + 4x^2 - 10,\ x_0 = 1,\\
f_3(x) &= e^x \sin(x) + \ln(x^2+1),\ x_0 = 2.9,\\
f_4(x) &= x^3 - 2x^2 - 5,\ x_0 = 6.3,\\
f_5(x) &= (x+2)e^x - 1,\ x_0 = -0.9,\\
f_6(x) &= (x^2-1)^{-1} - 1,\ x_0 = 1.6,\\
f_7(x) &= e^{\sin(x)} - 1 - \frac{x}{5},\ x_0 = 1.0.
\end{aligned} \tag{3.1}$$

Here, we take $\epsilon = 10^{-200}$ in the following stopping criteria $|f(x_{n+1}| < \epsilon$ and $|x_{n+1} - x_n| < \epsilon$. All examples were performed on maple with 2000 digit decimals.

Tables 1-7 show the numerical comparisons of our developed algorithms with Khattri (KHA) [2], Liu and Wang (LIU) [8], and Yasir and Abdul-Hassan (YAS) [1]. The columns represent the number of iterations N, the approximate root x_{n+1}, the magnitude $|f(x)|$ of $f(x)$ at the final estimate x_{n+1}, and the difference between two consecutive approximations $x_{n+1} - x_n$ of the equation.

Table 1: Comparison of various iterative methods.

Method	N	x_{n+1}	$\|f(x_{n+1})\|$	$\|x_{n+1} - x_n\|$
f_1, $x_0 = 3.6$				
KHA	3	0.25753028543986076045536730493 7	7.9e − 252	1.8e − 31
LIU	3	0.257530285439860760455367304937	1.6e − 250	7.1e − 21
YAS	3	0.257530285439860760455367304937	3.6e − 310	6.9e − 26
AL2.4	3	0.257530285439860760455367304937	2.6e − 310	8.0e − 26

Table 2: Comparison of various iterative methods.

Method	N	x_{n+1}	$\|f(x_{n+1})\|$	$\|x_{n+1} - x_n\|$
f_2, $x_0 = 1$				
KHA	3	1.365230013414096845760806828981	1.6e − 370	8.4e − 47
LIU	3	1.365230013414096845760806828981	1.0e − 998	2.6e − 87
YAS	3	1.365230013414096845760806828981	1.0e − 998	2.0e − 101
AL2.4	3	1.365230013414096845760806828981	1.0e − 998	2.0e − 101

Table 3: Comparison of various iterative methods.

Method	N	x_{n+1}	$\|f(x_{n+1})\|$	$\|x_{n+1} - x_n\|$
f_3, $x_0 = 2.9$				
KHA	3	3.237562984023921313250921300445	2.7e − 237	2.1e − 30
LIU	3	3.237562984023921313250921300445	1.1e − 535	1.8e − 45
YAS	3	3.237562984023921313250921300445	3.6e − 763	2.5e − 64
AL2.4	3	3.237562984023921313250921300445	2.0e − 765	1.6e − 64

Table 4: Comparison of various iterative methods.

Method	N	x_{n+1}	$\lvert f(x_{n+1})\rvert$	$\lvert x_{n+1}-x_n\rvert$
f_4, $x_0 = 6.3$				
KHA	4	2.690647448028613750350788882676	$9.2e-650$	$1.0e-81$
LIU	4	2.690647448028613750350788882676	$1.0e-998$	$1.8e-154$
YAS	3	2.690647448028613750350788882676	$1.1e-213$	$2.7e-18$
AL2.4	3	2.690647448028613750350788882676	$1.0e-213$	$2.7e-18$

Table 5: Comparison of various iterative methods.

Method	N	x_{n+1}	$\lvert f(x_{n+1})\rvert$	$\lvert x_{n+1}-x_n\rvert$
f_5, $x_0 = -0.8$				
KHA	3	−0.442854401002388583141327999999	$1.4e-298$	$9.6e-38$
LIU	3	−0.442854401002388583141327999999	$6.2e-579$	$7.1e-49$
YAS	3	−0.442854401002388583141327999999	$4.2e-961$	$1.5e-80$
AL2.4	3	−0.442854401002388583141327999999	$4.0e-972$	$5.7e-79$

Table 6: Comparison of various iterative methods.

Method	N	x_{n+1}	$\lvert f(x_{n+1})\rvert$	$\lvert x_{n+1}-x_n\rvert$
f_6, $x_0 = 3.2$				
KHA	3	1.414213562373095048801688724209	$2.2e-424$	$1.2e-53$
LIU	3	1.414213562373095048801688724209	$1.0e-334$	$4.6e-29$
YAS	3	1.414213562373095048801688724209	$1.0e-999$	$1.6e-87$
AL2.4	3	1.414213562373095048801688724209	$1.0e-999$	$5.5e-92$

Table 7: Comparison of various iterative methods.

Method	N	x_{n+1}	$\lvert f(x_{n+1})\rvert$	$\lvert x_{n+1}-x_n\rvert$
f_7, $x_0 = 1.0$				
KHA	3	$7.080901417521118963254487e-241$	$5.7e-241$	$1.4e-30$
LIU	3	$9.117579045189126459350754e-279$	$7.3e-279$	$9.0e-24$
YAS	3	$1.202260016613233904793461e-613$	$9.6e-614$	$1.2e-51$
AL2.4	3	$3.481190908233362755514154e-767$	$2.8e-767$	$2.0e-64$

4. Polynomiography

Polynomials are one of the most significant objects in many fields of mathematics. Polynomial root-finding has played a key role in the history of mathematics. It is one of the oldest and most deeply studied mathematical problems. The last interesting contribution to the polynomials root-finding history was made by Kalantari [4], who introduced the polynomiography. Polynomiography is defined to be "the art and science of visualization in approximation of the zeros of complex polynomials, via fractal, and nonfractal images created using the mathematical convergence properties of iteration functions" [4]. An individual image is called a "polynomiograph".

In the numerical algorithms that are based on the iterative processes, we need a stopping criterion for the process, that is, a test that tells us that the process has converged or it is very near to the solution. This type of test is called a convergence test. Usually, in the iterative process that use a feedback, like the root-finding methods, the standard convergence test has the following form $|f(z_{n+1})| < \epsilon$ and $|z_{n+1} - z_n| < \epsilon$. The different colors of an image depend on the number of iterations to reach a root with given accuracy ϵ. Here, we present the basins of attractions using the following complex polynomials of different degrees:

$$p_1(z) = z^3 - 1, \qquad p_2(z) = z^4 - 1, \qquad p_3(z) = z^5 - 1,$$

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

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

In Figures 1-6, polynomiographs of different complex polynomials for Khattri (KHA) [2] Liu and Wang (LIU) [8] and Yasir and Abdul-Hassan (YAS) [1], and our developed algorithms have been shown which describe the regions of convergence of these polynomials.

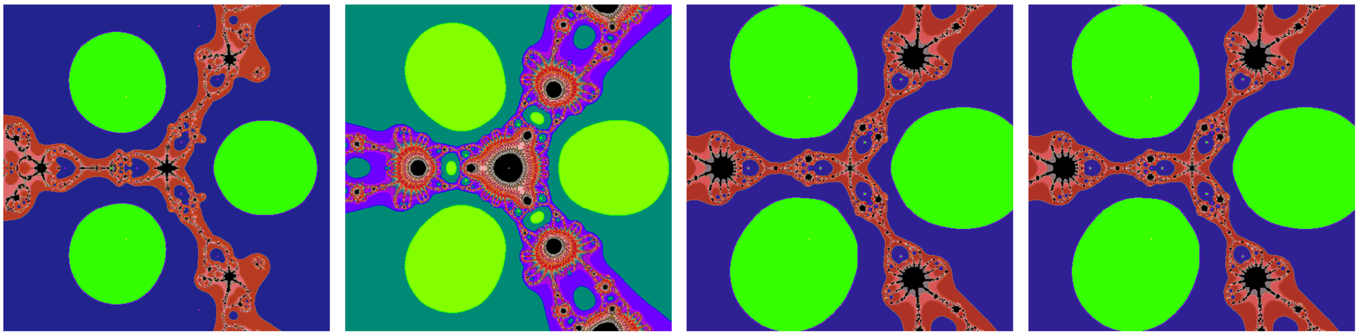

Figure 1: Polynomiographs for the polynomial of KHA, LIU, YAS, and AL2.4, respectively for $p_1(z)$.

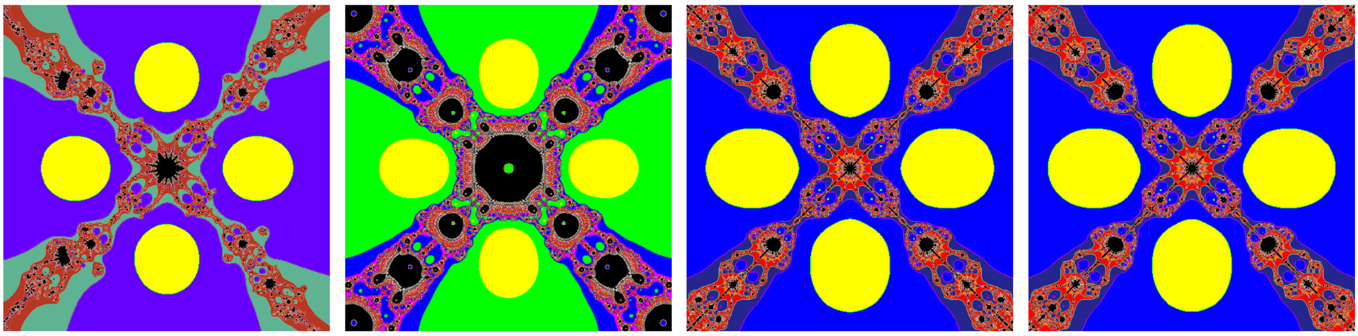

Figure 2: Polynomiographs for the polynomial of KHA, LIU, YAS, and AL2.4, respectively for $p_2(z)$.

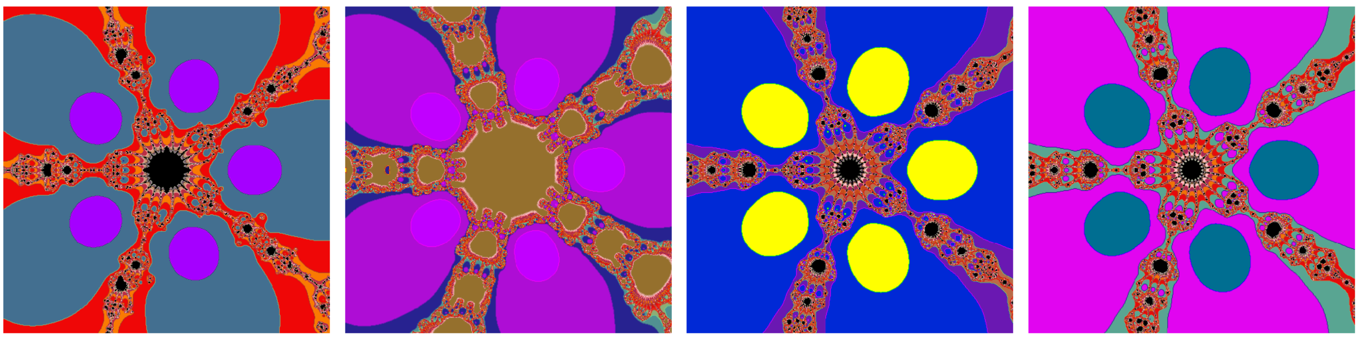

Figure 3: Polynomiographs for the polynomial of KHA, LIU, YAS, and AL2.4, respectively for $p_3(z)$.

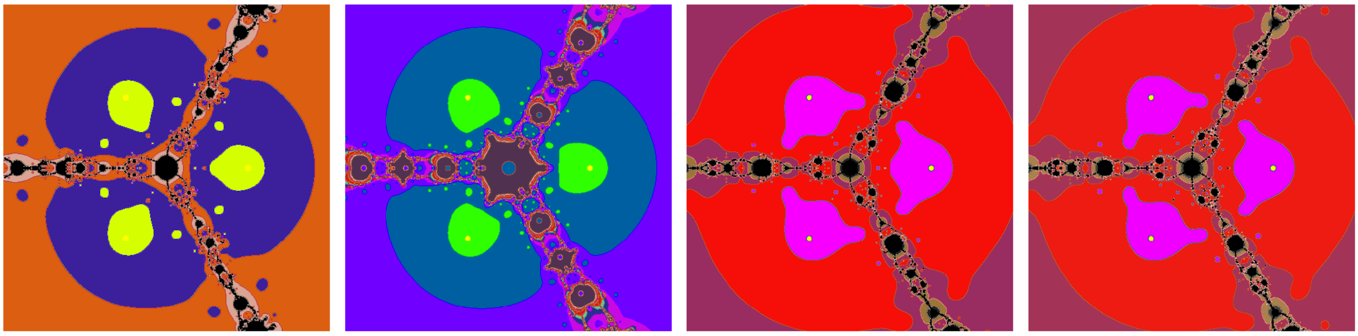

Figure 4: Polynomiographs for the polynomial of KHA, LIU, YAS, and AL2.4, respectively for $p_4(z)$.

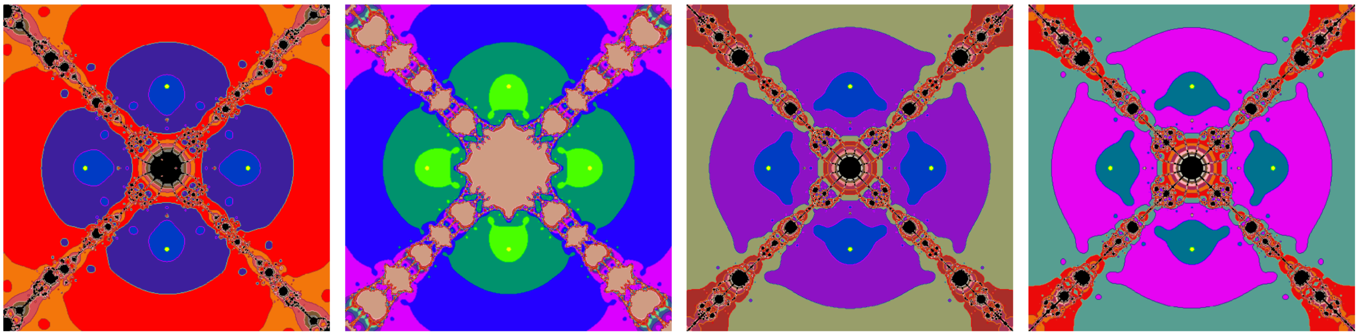

Figure 5: Polynomiographs for the polynomial of KHA, LIU, YAS, and AL2.4, respectively for $p_5(z)$.

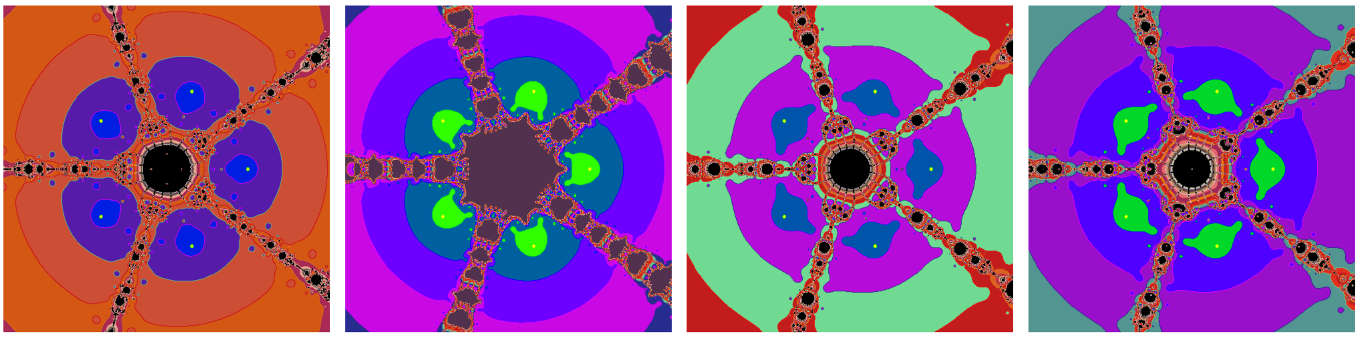

Figure 6: Polynomiographs for the polynomial of KHA, LIU, YAS, and AL2.4, respectively for $p_6(z)$.

5. Conclusions

In this method, we introduced the new twelfth order convergent iterative method. By using some test examples, the performance of the proposed algorithms is also discussed. The numerical results uphold the analysis of the convergence which can be seen in Tables 1-7. The algorithms shown is equally effective at estimating roots. But Algorithm 2.4 performed better in examples (3.1). Polynomiographs of complex polynomials of different degrees using three-step iterative methods and our proposed algorithms have been generated. The presented polynomiographs are rich and colorful and have very interesting and aesthetic patterns, which reflects the dynamical aspects of our proposed algorithms.

Acknowledgment

We appreciate reviewers for their helpful and constructive comments.

References

[1] N. Y. Abdul-Hassan, *New predictor-corrector iterative methods with twelfth-order convergence for solving nonlinear equations*, Amer. J. Appl. Math., **4** (2016), 175–180. 3, 3, 4

[2] F. Ahmad, S. Hussain, S. Hussain, A. Rafiq, *New twelfth order J-Halley method for solving nonlinear equations*, Open Sci. J. Math. Appl., **1** (2013), 1–4. 3, 3, 4

[3] A. S. Householder, *The numerical treatment of a single nonlinear equation*, McGraw-Hill, New York, USA (1970). 1, 2, 2.2

[4] B. Kalantari, *Method of creating graphical works based on polynomials*, U.S. Patent, **6** (2005), 894–705. 4

[5] S. Khattri, *Another note on some quadrature based three-step iterative methods*, Numer. Algebra, **3** (2013), 549–555. 1

[6] B. Kong-ied, *Two new eighth and twelfth order iterative methods for solving nonlinear equations*, Int. J. Math. Comput. Sci., **16** (2021), 333–344. 1

[7] S. Li, *Fourth-order iterative method without calculating the higher derivatives for nonlinear equation*, J. Algorithms Comput. Tech., **13** (2019), 8 pages. 1, 2

[8] X. Liu, X. Wang, *A family of methods for solving nonlinear equations with twelfth-order convergence*, Appl. Math., **4** (2013), 326–329. 3, 3, 4

[9] N. A. Mir, T. Zaman, *Some quadrature based three-step iterative methods for non-linear equations*, Appl. Math. Comput., **193** (2007), 366–373. 1

[10] M. S. K. Mylapalli, R. K. Palli, V. B. K. Vatti, *An iterative method with twelfth order convergence for solving non-linear equations*, Adv. Appl. Math. Sci., **20** (2021), 1633–1643.

[11] A. Naseem, M. A. Rehman, T. Abdeljawad, *Numerical algorithms for finding zeros of nonlinear equations and their dynamical aspects*, J. Math., **2020** (2020), 11 pages.

[12] M. Nawaz, A. Naseem, W. Nazeer, *New twelfth order algorithms for solving nonlinear equations by using variational iteration technique*, J. Prime Res. Math., **14** (2018), 24–36.

[13] R. Thukral, *New twelfth order iterative methods for finding multiple roots of nonlinear equations*, Amer. J. Comput. Appl. Math., **5** (2015), 33–41. 1

[14] J. F. Traub, *Iterative methods for solution of equations*, Prentice-Hall, Englewood Cliffs, NJ, (1964). 1