Received November 16, 2020, accepted January 1, 2021, date of publication January 5, 2021, date of current version January 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049428

Some New Iterative Algorithms for Solving One-Dimensional Non-Linear Equations and Their Graphical Representation

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 Solving non-linear equation is perhaps one of the most difficult problems in all of numerical computations, especially in a diverse range of engineering applications. The convergence and performance characteristics can be highly sensitive to the initial guess of the solution for most numerical methods such as Newton's method. However, it is very difficult to select reasonable initial guess of the solution for most systems of non-linear equations. Besides, the computational efficiency is not high enough. Taking this into account, based on variational iteration technique, we develop some new iterative algorithms for solving one-dimensional non-linear equations. The convergence criteria of these iterative algorithms has also been discussed. The superiority of the proposed iterative algorithms is illustrated by solving some test examples and comparing them with other well-known existing iterative algorithms in literature. In the end, the graphical comparison of the proposed iterative algorithms with other well-known iterative algorithms have been made by means of polynomiographs of different complex polynomials which reflect the fractal behavior and dynamical aspects of the proposed iterative algorithms.

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

I. INTRODUCTION

Solving non-linear equations of the form $f(x) = 0$ is one of the most important problems in all of numerical computations, especially in a diverse range of engineering applications. Many applied problems can be reduced to solving systems of non-linear equations, which is one of the most basic problems in Mathematics. This task has applications in many scientific fields of engineering and computer sciences.

To find solution of such problems, analytical methods do not assist us and therefore, we have to find appropriate solution of such equations by numerical methods which are totally based on iterative schemes. In an iterative scheme, we start the algorithm by choosing an inial guess x_0 which is refined step by step by means of iterations until the approximated solution is achieved. Some basic iterative methods are given in literature [4], [6], [9], [29], [34], [35] and the references therein. The most famous and well-known method for finding roots of non-linear equations is of the following form:

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

Which is quadratically convergent Newton's method [35] for the solution of non-linear equations.

For acquiring better convergence, many scholars proposed a large number of iterative algorithms by means of of various types of mathematical techniques. In 1970, Househölder [13] suggested a cubically convergent method using Taylor's series expansion. Using the modified Adomian decomposition technique, Abbasbandy [1] proposed a single-step iterative method having convergence of order three in 2003. After that, in 2007, Kou [21] improved some third-order modifications of Newton's method and obtained many new methods for solving non-linear equations. Nazeer *et al.* [25], [26] in (2016), suggested some new algorithms by means of finite difference scheme, which are free from second

The associate editor coordinating the review of this manuscript and approving it for publication was Santi C. Pavone.

derivatives, having higher convergence orders. In the same year, Sharma and Arora [32] proposed a simple yet efficient family of three-point iterative methods with eighth order of convergence for solving non-linear equations. After that, in [30], the authors not only developed an optimal class of three-step eighth-order methods with higher-order weight functions employed in the second and third sub-steps, but also investigate their dynamics underlying the purely imaginary extraneous fixed points. In 2019, Alharbi *et al.* [2] suggested some higher order algorithms by decomposition technique, having convergence orders up to fifteen. Very recently, Behl and Martinez [3] constructed some new high order and efficient iterative techniques for solving a system of non-linear equations whose convergence orders varied and reached up to six. The suggested methods are the extension of the Jisheng Kou's method [22] for multidimensional case.

In this article, we proposed three new iterative algorithms using variational iteration technique by considering two auxiliary iteration functions $\phi(x)$ and $\psi(x)$. The first one, $\phi(x)$ acts as a predictor function having convergence of order q where $q \geq 1$. The predictor function is used to attain iterative methods of convergence order $q + r$, where $r \geq 1$ is the convergence order of second auxiliary function $\psi(x)$. By an iteration function, we mean a function f which is obtained by composing the function itself for a certain number of times and the convergence order is a quantity that shows how fast an iterative method approaches to the required approximate root. Mathematically, it is written as $\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^p} = \nu$, here $p \geq 1$ is called the convergence order, the constant ν is the rate of convergence or asymptotic error constant and e_n, e_{n+1} are the errors at nth and $(n+1)$th iterations respectively.

Using variational iteration technique, we develop some new higher order iterative algorithms with better performance and efficiency. The variational iteration technique was introduced by Inokuti *et al.* [15]. Using this technique, Noor [27] and Noor and Shah [28] proposed some iterative methods for the solution of non-linear equations. The purpose of this technique was to solve a variety of diverse problems [10]–[12].

Now we apply the described technique to obtain higher order iterative algorithms. These algorithms are very fast using less number of iterations to reach the required solution, free from 3rd and higher derivatives with ninth order of convergence which raises their efficiency indices. The convergence criteria of the suggested algorithms is also discussed. Various test examples have been solved to show their performance as compare to the other similar existing iterative algorithms in literature.

II. CONSTRUCTION OF SOME NEW ITERATIVE ALGORITHMS

In this section, we construct some new iterative algorithms by means of variational iteration technique. These algorithms are multi-step iterative methods which involve predictor and corrector steps. The proposed iterative algorithms have higher order of convergence than one-step algorithms. By applying variational iteration technique, we derive some new iterative algorithms of order $q + r$ where $q, r \geq 1$ are the orders of convergence of the auxiliary iteration functions $\phi(x)$ and $\psi(x)$ respectively.

Now consider the non-linear equation of the form

$$f(x) = 0 \tag{2}$$

Suppose that α is the simple root and γ is the initial guess sufficiently close to α. For better understanding and to deliver the basic idea, we suppose the approximate solution x_n of (2) such that

$$f(x_n) \neq 0$$

We consider $\phi(x)$ and $\psi(x)$ as two iteration functions of order q and r respectively. Then

$$x_{n+1} = \phi(x_n) + \mu[f(\psi(x_n))g(\psi(x_n))]^t \tag{3}$$

where $t = \frac{q}{r}$ is a recurrence relation which generates iterative algorithms of order $q + r$ and $g(x)$ is any smooth arbitrary function which later on is converted to $g(\psi(x_n))$ and μ is a parameter, called the ''Lagrange's multiplier'' and can be determined by using the optimality criteria on (3) by setting first derivative of (3) with respect to x_n equal to zero as follows:

$$\frac{dx_{n+1}}{x_n} = 0 \tag{4}$$

Which leads us to the following equality:

$$\mu = -\frac{\phi'(x_n)[f(\psi(x_n))g(\psi(x_n))]^{1-t}}{t\psi'(x_n)[f'(\psi(x_n))g(\psi(x_n)) + f(\psi(x_n))g'(\psi(x_n))]} \tag{5}$$

From (3) and (5), we get

$$x_{n+1} = \phi(x_n) - \frac{\phi'(x_n)}{t\psi'(x_n)} \frac{f(\psi(x_n))g(\psi(x_n))}{[f'(\psi(x_n))g(\psi(x_n)) + f(\psi(x_n))g'(\psi(x_n))]} \tag{6}$$

Now we are going to apply (6) for constructing a general iterative scheme for iterative methods. For this, suppose that

$$\psi(x_n) = y_n = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)f''(x_n)}{2[f'^3(x_n) - f(x_n)f'(x_n)f''(x_n)]} \tag{7}$$

Which is well-known Golbabai and Javidi's method with cubic convergence [8]. With the help of (6) and (7), we can write

$$x_{n+1} = \phi(x_n) - \frac{\phi'(x_n)f(y_n)g(y_n)}{ty'_n[f'(y_n)g(y_n) + f(y_n)g'(y_n)]} \tag{8}$$

Let

$$\phi(x_n) = z_n = y_n - \frac{f(y_n)}{f'(y_n)} \tag{9}$$

Which is two-step iterative method having convergence of order six [see eq.(23)]. Differentiate eq.(9) w.r.t ''x'', we have

$$\phi'(x_n) = z'_n = \frac{f(y_n)f''(y_n)}{f'^2(y_n)} y'_n \tag{10}$$

and from Taylor's series, we can write

$$f(z_n) = f(y_n) + (z_n - y_n)f'(y_n) + \frac{(z_n - y_n)^2}{2}f''(y_n)$$
$$= \frac{f^2(y_n)f''(y_n)}{2f'^2(y_n)} \tag{11}$$

From (10) and (11), we have

$$\phi'(x_n) = \frac{2f(z_n)}{f(y_n)}y_n' \tag{12}$$

With the help of (8), (9) and (12), we get

$$x_{n+1} = z_n - \frac{2f(z_n)g(y_n)}{t[g(y_n)f'(y_n) + g'(y_n)f(y_n)]} \tag{13}$$

Here $t = \frac{6}{3} = 2$, which is according to the above described technique. Then eq.(13) becomes:

$$x_{n+1} = z_n - \frac{f(z_n)g(y_n)}{[g(y_n)f'(y_n) + g'(y_n)f(y_n)]} \tag{14}$$

Relation(14) is the main and general iterative scheme, which we use to deduce some new iterative algorithms by considering some particular cases of the auxiliary function g.

A. CASE 1

Let $g(x_n) = e^{(-\beta x_n)}$, then $g'(x_n) = -\beta g(x_n)$. Using these values in (14), we obtain the following algorithm.

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

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)f''(x_n)}{2[f'^3(x_n) - f(x_n)f'(x_n)f''(x_n)]}$$
$$z_n = y_n - \frac{f(y_n)}{f'(y_n)}$$
$$x_{n+1} = z_n - \frac{f(z_n)}{[f'(y_n) - \beta f(y_n)]}$$

B. CASE 2

Let $g(x_n) = e^{-\beta f(x_n)}$, then $g'(x_n) = -\beta f'(x_n)g(x_n)$. Using these values in (14), we obtain the following algorithm.

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

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)f''(x_n)}{2[f'^3(x_n) - f(x_n)f'(x_n)f''(x_n)]}$$
$$z_n = y_n - \frac{f(y_n)}{f'(y_n)}$$
$$x_{n+1} = z_n - \frac{f(z_n)}{[f'(y_n) - \beta f(y_n)f'(y_n)]}$$

C. CASE 3

Let $g(x_n) = e^{-\beta x_n^2}$, then $g'(x_n) = -2\beta x_n g(x_n)$. Using these values in (14), we obtain the following algorithm.

Algorithm 3: For a given x_0, compute the approximate solution x_{n+1} by the following iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)f''(x_n)}{2[f'^3(x_n) - f(x_n)f'(x_n)f''(x_n)]}$$
$$z_n = y_n - \frac{f(y_n)}{f'(y_n)}$$
$$x_{n+1} = z_n - \frac{f(z_n)}{[f'(y_n) - 2\beta y_n f(y_n)]}$$

In all above cases, we take $n = 0, 1, 2, 3, \ldots$

To get best results in all above defined algorithms, always choose that values of β which makes the denominator non-zero and largest in magnitude.

III. CONVERGENCE ANALYSIS

In this section, we discuss the convergence criteria of the main and general iteration scheme described in relation (14).

Theorem 1: Assume that $\alpha \in I$ be the simple root of the differentiable function $f : I \subset \mathrm{R} \to \mathrm{R}$ on an open interval I. If the initial guess x_0 is sufficiently close to α, then the convergence order of the main and general iteration scheme described in relation (14) is at least nine.

Proof: To prove the convergence of the main and general iteration scheme described in relation (14), we assume that α is the simple 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 about $x = \alpha$, we have

$$f(x_n) = f'(\alpha)e_n + \frac{1}{2!}f''(\alpha)e_n^2 + \frac{1}{3!}f'''(\alpha)e_n^3$$
$$+\frac{1}{4!}f^{(iv)}(\alpha)e_n^4 + \ldots + O(e_n^{10})$$
$$f(x_n) = f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + \ldots$$
$$+O(e_n^{10})] \tag{15}$$

where

$$c_n = \frac{1}{n!}\frac{f^{(n)}(\alpha)}{f'(\alpha)}, \quad n = 2, 3, 4, \ldots$$
$$f'(x_n) = f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4$$
$$+ \ldots + O(e_n^{10})] \tag{16}$$
$$f''(x_n) = f'(\alpha)[2c_2 + 6c_3e_n + 12c_4e_n^2 + 20c_5e_n^3$$
$$+30c_6e_n^4 + \ldots + O(e_n^{10})] \tag{17}$$

With the help of (15) – (17), we get

$$y_n = \alpha - c_3e_n^3 + (c_2^3 - 3c_4)e_n^4 + \ldots + O(e_n^{10}) \tag{18}$$
$$f(y_n) = f'(\alpha)[(y_n - \alpha) + c_2(y_n - \alpha)^2 + c_3(y_n - \alpha)^3$$
$$+c_4(y_n - \alpha)^4 + \ldots + O(e_n^{10})] \tag{19}$$
$$f'(y_n) = f'(\alpha)[1 + 2c_2(y_n - \alpha) + 3c_3(y_n - \alpha)^2$$
$$+4c_4(y_n - \alpha)^3 + 5c_5(y_n - \alpha)^4$$
$$+ \ldots + O(e_n^{10})] \tag{20}$$

$$g(y_n) = g(\alpha) + (y_n - \alpha)g'(\alpha) + \frac{(y_n - \alpha)^2}{2!}g''(\alpha) + \frac{(y_n - \alpha)^3}{3!}g'''(\alpha) + \frac{(y_n - \alpha)^4}{4!}g^{(iv)}(\alpha) + \ldots + O(e_n^{10}) \tag{21}$$

$$g'(y_n) = g'(\alpha) + (y_n - \alpha)g''(\alpha) + \frac{(y_n - \alpha)^2}{2!}g'''(\alpha) + \frac{(y_n - \alpha)^3}{3!}g^{(iv)}(\alpha) + \frac{(y_n - \alpha)^4}{4!}g^{(v)}(\alpha) + \ldots + O(e_n^{10}) \tag{22}$$

With the help of (18) – (20), we have

$$z_n = \alpha + c_2 c_3^2 e_n^6 + \ldots + O(e_n^{10}) \tag{23}$$

$$f(z_n) = f'(\alpha)[(z_n - \alpha) + c_2(z_n - \alpha)^2 + c_3(z_n - \alpha)^3 + c_4(z_n - \alpha)^4 + \ldots + O(e_n^{10})] \tag{24}$$

Using equations (15) – (24) in general iteration scheme(14), we get the same result as given below

$$x_{n+1} = \alpha - 2c_2^2 c_3^3 e_n^9 + O(e_n^{10})$$

Which implies that

$$e_{n+1} = -2c_2^2 c_3^3 e_n^9 + O(e_n^{10})$$

Which shows that the main and general iteration scheme(14) is of ninth order of convergence and all iterative algorithms deduced from it have also the same order of convergence. □

IV. NUMERICAL RESULTS

In this section, we include some non-linear functions to demonstrate the performance of newly proposed iterative algorithms (for $\beta = 1$). We compare these algorithms with Newton's method (NM) [35], Golbabai and Javidi's method (JM) [8], Householder's method (HHM) [13] and modified Halley's method (MHM) [14]. For this purpose, following test examples have been solved:

$$f_1(x) = (e^{x^2} + x - 20)^{20}, \quad x_0 = 2.00,$$
$$f_2(x) = \pi - 2x\sin\frac{\pi}{x}, \quad x_0 = 2.50,$$
$$f_3(x) = x^3 + \ln x + 0.15\cos 5x, \quad x_0 = 2.20,$$
$$f_4(x) = x^3 + 4x^2 - 10, \quad x_0 = -0.30,$$
$$f_5(x) = (x-1)^3 - 1, \quad x_0 = 1.30,$$
$$f_6(x) = x^2 - e^x - 3x + 2, \quad x_0 = 2.50,$$
$$f_7(x) = x^3 - x^2 + 3x\cos x - 1, \quad x_0 = 1.00,$$
$$f_8(x) = 0.986x^3 - 5.181x^2 + 9.067x - 5.289, \quad x_0 = 3.10,$$
$$f_9(x) = \frac{x}{1-x} - \ln\left[\frac{0.4(1-x)}{0.4-0.5x}\right]^5 + 4.45977, \ x_0 = 0.78.$$

The last two numerical examples namely; f_8 and f_9 are actually the conversion of Engineering problems to the non-linear equations. The first one is the the van der Waal's equation for interpreting the real and ideal gas behavior [36] that has been converted to the non-linear form after choosing the appropriate values of the parameters. The variable x denotes the volume of the gas which is being under consideration. The second numerical example is related to the fractional transformation in a chemical reactor [31] which we converted to the non-linear problem. In this problem, x represents a fractional transformation of a particular species in the chemical reactor problem. For the negative values of x, the expression f_9 has no physical meaning and has the bounded region [0, 1] beyond which its derivative vanishes. That's why, we have to choose the initial guess carefully within the bounded region to approximate the real root of f_9. The numerical results of the above described test examples have been shown in Table 1.

Here, we take the accuracy $\varepsilon = 10^{-15}$ in the following stopping criteria

$$|x_{n+1} - x_n| < \varepsilon \tag{25}$$

Table 1 shows the numerical comparisons of our developed iterative algorithms (for $\beta = 1$), with Newton's method (NM), Golbabai and Javidi's method (JM), Householder's method (HHM) and modified Halley's method (MHM). The columns represent the number of iterations N, the magnitude $|f(x)|$ of $f(x)$ at the final estimate x_{n+1}, the approximate root x_{n+1}, the difference between two consecutive approximations $|x_{n+1} - x_n|$, the computational order of convergence (COC) which can be approximated using the following formula:

$$COC \approx \frac{ln\frac{|x_{n+1}-\alpha|}{|x_n-\alpha|}}{ln\frac{|x_n-\alpha|}{|x_{n-1}-\alpha|}}$$

which was introduced by Weerakoon and Fernando in (2000) [37] and the last column represents CPU time consumption in seconds, taken by different algorithms to approximate the required solution.

From the over all obtained results of the Table 1, we can clearly see that the suggested algorithms approximated the solutions of different test problems with the great accuracy than the estimations gained through the other four iteration schemes. These results also show that the suggested algorithms consumed less number of iteration to meet the stopping criteria (25) with high computational order of convergence as compared to the other comparable methods. It should also be noted that the presented algorithms take less CPU time in seconds to achieve the stopping criteria for approximating different test problems. In short, we can say that the suggested algorithms perform faster and are more accurate to the exact solutions in comparison with the other compared methods.

We solved all test problems and calculated the CPU time consumption in seconds with the aid of the computer program Maple 13.

Table 2 shows the comparison of the number of iterations required for different iterative methods with our developed iterative algorithms (for $\beta = 1$) to approximate the root of the given non-linear function for the stopping criteria (25) with the accuracy $\varepsilon = 10^{-100}$. The columns represent the number of iterations for different functions along with initial guess x_0.

TABLE 1. Comparison of NM, JM, HHM, MHM, Algorithm 1, Algorithm 2 & Algorithm 3.

Method	N	$\|f(x_{n+1})\|$	x_{n+1}	$\sigma = \|x_{n+1} - x_n\|$	COC	CPU Time
$f_1(x), x_0 = 2.00$						
NM	103	$4.558695e-15$	1.7078942088489236472375525333489143 3	$1.573265e-04$	2	3.826
JM	5	$1.338286e-17$	1.7026139275782093725055833178873629 3	$2.466806e-03$	3	3.857
HHM	69	$3.834935e-15$	1.7078684632857487587276982625006409 2	$2.359586e-04$	3	3.951
MHM	16	$1.090746e-15$	1.7076877032228690822027075625055696 6	$1.030725e-03$	5	3.997
Algorithm 1	4	$1.101429e-15$	1.7020503797376140619187078321016253 4	$3.493353e-03$	9	2.093
Algorithm 2	5	$1.143178e-20$	1.7032920488375961955576211516267682 8	$1.981392e-03$	9	2.171
Algorithm 3	4	$9.834412e-16$	1.7020664842647601189105553233269826 6	$3.473872e-03$	9	2.234
$f_2(x), x_0 = 2.50$						
NM	6	$2.228176e-16$	1.6574002402580060520306373082137212 8	$1.041405e-08$	2	4.133
JM	4	$6.646956e-29$	1.6574002402580061237937386723732550 9	$4.083789e-10$	3	4.208
HHM	5	$5.590553e-31$	1.6574002402580061237937386723520272 3	$6.845206e-11$	3	4.283
MHM	3	$2.339921e-31$	1.6574002402580061237937386723517718 1	$8.174677e-07$	5	4.343
Algorithm 1	2	$1.277344e-41$	1.6574002402580061237937386723518471 7	$3.534821e-05$	9	2.441
Algorithm 2	2	$1.679481e-43$	1.6574002402580061237937386723518471 7	$2.249937e-05$	9	2.519
Algorithm 3	2	$2.671200e-38$	1.6574002402580061237937386723518471 7	$7.654916e-05$	9	2.613
$f_3(x), x_0 = 2.20$						
NM	6	$6.072231e-15$	0.7467707647608674803734346483989852 6	$4.576382e-08$	2	2.976
JM	4	$4.879664e-32$	0.7467707647608657104449509917514352 3	$9.555808e-11$	3	3.038
HHM	5	$3.305553e-39$	0.7467707647608657104449509917514210 1	$8.803620e-14$	3	3.086
MHM	4	$8.986943e-53$	0.7467707647608657104449509917514210 1	$6.842616e-11$	5	3.132
Algorithm 1	3	$1.038543e-114$	0.7467707647608657104449509917514210 1	$7.899816e-13$	9	2.618
Algorithm 2	3	$2.079929e-65$	0.7467707647608657104449509917514210 1	$2.142561e-07$	9	2.696
Algorithm 3	3	$3.423949e-64$	0.7467707647608657104449509917514210 1	$3.726798e-07$	9	2.760
$f_4(x), x_0 = -0.30$						
NM	54	$8.127500e-30$	1.3652300134140968457608068289821582 5	$1.001963e-15$	2	3.343
JM	62	$6.526278e-28$	1.3652300134140968457608068289421449 7	$8.674049e-10$	3	3.375
HHM	9	$2.386831e-45$	1.3652300134140968457608068289816660 8	$7.007012e-16$	3	3.421
MHM	22	$2.439119e-36$	1.3652300134140968457608068289816660 8	$9.248288e-08$	5	3.452
Algorithm 1	3	$5.199700e-86$	1.3652300134140968457608068289816660 8	$1.044771e-09$	9	2.746
Algorithm 2	3	$2.720781e-70$	1.3652300134140968457608068289816660 8	$2.774524e-08$	9	2.810
Algorithm 3	3	$4.580612e-83$	1.3652300134140968457608068289816660 8	$1.346601e-09$	9	2.871
$f_5(x), x_0 = 1.30$						
NM	9	$1.373004e-19$	2.0000000000000000000457667933070243 7	$2.139317e-10$	2	2.928
JM	4	$1.376914e-16$	2.0000000000000000045897128898335106 22	$5.163768e-06$	3	2.959
HHM	27	$4.799079e-39$	2.0000000000000000000000000000000000 0	$9.864217e-14$	3	2.989
MHM	4	$3.402977e-44$	2.0000000000000000000000000000000000 0	$1.867040e-09$	5	3.020
Algorithm 1	3	$7.330764e-16$	1.9999999999999999755641201884010698 43	$2.730810e-02$	9	2.593
Algorithm 2	3	$2.233894e-29$	1.9999999999999999999999999999992553 69	$8.360388e-04$	9	2.656
Algorithm 3	3	$1.124811e-26$	1.9999999999999999999999999996250630 75	$1.544171e-03$	9	2.718
$f_6(x), x_0 = 2.50$						
NM	5	$6.456953e-15$	0.2575302854398590516656343320713527 6	$1.352209e-07$	2	3.532
JM	7	$2.540428e-17$	0.2575302854398607671784421200306911 8	$4.902361e-06$	3	3.579
HHM	4	$1.054587e-23$	0.2575302854398607604553645140422707 9	$4.130791e-08$	3	3.611
MHM	4	$2.646153e-74$	0.2575302854398607604553673049372417 8	$6.227672e-15$	5	3.642
Algorithm 1	2	$9.040960e-20$	0.2575302854398607604792936030837855 5	$2.202739e-02$	9	2.345
Algorithm 2	2	$3.293944e-24$	0.2575302854398607604553681766576861 2	$6.212211e-03$	9	2.376
Algorithm 3	2	$1.232057e-18$	0.2575302854398607601293116164972736 4	$3.079645e-02$	9	2.408
$f_7(x), x_0 = 1.00$						
NM	10	$5.054844e-17$	0.3953236229863151629760059168866574 8	$5.773225e-09$	2	3.145
JM	7	$1.714115e-37$	0.3953236229863151883761427884384019 9	$8.223850e-13$	3	3.208
HHM	24	$6.899623e-29$	0.3953236229863151883761427884730719 8	$2.975250e-10$	3	3.270
MHM	6	$6.314672e-50$	0.3953236229863151883761427884384019 9	$1.880569e-10$	5	3.317
Algorithm 1	4	$5.508815e-114$	0.3953236229863151883761427884384019 9	$4.177495e-13$	9	2.484
Algorithm 2	4	$1.737209e-75$	0.3953236229863151883761427884384019 9	$7.631173e-09$	9	2.515
Algorithm 3	4	$3.957250e-50$	0.3953236229863151883761427884384019 9	$5.250956e-06$	9	2.594
$f_8(x), x_0 = 3.10$						
NM	10	$1.815641e-25$	1.9298462428478622184875295285258264 9	$5.866916e-13$	2	3.353
JM	5	$1.499299e-36$	1.9298462428478622184875274278654564 7	$1.149928e-12$	3	3.392
HHM	7	$8.608929e-34$	1.9298462428478622184875274278654664 5	$5.404901e-12$	3	3.438
MHM	4	$4.590068e-47$	1.9298462428478622184875274278654564 9	$2.421897e-10$	5	3.454
Algorithm 1	3	$3.796053e-26$	1.9298462428478622184875269886697586 2	$5.481798e-04$	9	2.402
Algorithm 2	3	$2.802586e-63$	1.9298462428478622184875274278654564 9	$4.053280e-08$	9	2.465
Algorithm 3	3	$4.417216e-15$	1.9298462428478111121853965039082320 9	$1.025404e-02$	9	2.527
$f_9(x), x_0 = 0.78$						
NM	6	$2.214298e-27$	0.7573962462537538794596412979569071 8	$1.323123e-15$	2	3.065
JM	5	$1.140552e-41$	0.7573962462537538794596412979291452 9	$8.139895e-16$	3	3.128
HHM	4	$6.203623e-36$	0.7573962462537538794596412979291452 9	$6.889797e-14$	3	3.175
MHM	3	$1.203934e-41$	0.7573962462537538794596412979291452 9	$2.727415e-10$	5	3.206
Algorithm 1	2	$7.858948e-33$	0.7573962462537538794596412979291451 9	$1.303132e-05$	9	2.621
Algorithm 2	2	$7.254301e-28$	0.7573962462537538794596412979200501 7	$4.020933e-05$	9	2.668
Algorithm 3	2	$6.722228e-33$	0.7573962462537538794596412979291452 1	$1.283116e-05$	9	2.699

TABLE 2. Comparison of number of iterations required for different iterative methods for $\varepsilon = 10^{-100}$.

Method	$f_1(x)$ $x_0 = 2.00$	$f_2(x)$ $x_0 = 2.50$	$f_3(x)$ $x_0 = 2.20$	$f_4(x)$ $x_0 = -0.30$	$f_5(x)$ $x_0 = 1.30$	$f_6(x)$ $x_0 = 2.50$	$f_7(x)$ $x_0 = 1.00$	$f_8(x)$ $x_0 = 3.1$	$f_9(x)$ $x_0 = 0.78$
NM	294	07	09	56	12	08	13	13	08
JM	18	06	06	64	06	09	08	06	06
HHM	197	06	06	10	28	06	26	09	05
MHM	47	04	05	23	05	05	07	05	04
Algorithm 1	16	03	03	04	04	03	04	04	03
Algorithm 2	17	03	04	04	04	03	05	04	03
Algorithm 3	16	03	04	04	04	03	05	04	03

The numerical results of Table 2 again certified the fast and best performance of the proposed algorithms in terms of number of iterations even though the accuracy has been raised to 10^{-100}. All calculations have been carried out using the computer program Maple 13.

V. GRAPHICAL COMPARISON BY MEANS OF POLYNOMIOGRAPHS

The problems related to finding the roots of polynomials have played a vital role in engineering and mathematical sciences. It is one of the oldest and most deeply studied mathematical problems as one can study the history of Mathematics that the ancient Greeks and Sumerians considered such practical problems in 3000 B.C, that can now be stated as a root-finding problems using modern mathematical language. In the seventeenth century, Newton suggested an algorithm for approximating roots of polynomials. After that Cayley [5] studied the chaotic and strange behavior while applying Newton's method on cubic polynomial $x^3 - 1$ in complex plane. The problem arisen by Calay was explained by Julia in 1919. The Julia sets bought many new discoveries i.e., Mandelbrot set and Fractal in 1970 [24]. The term ''Polynomiography'' was first introduced in 2005 by Kalantari [16], [17]. He defined polynomiography as the combination of both science and art related to visualization of the root-finding process for a polynomial in complex plane. The individual image produced as a result of polynomiography is thus called a ''Polynomiograph''. The polynomiographs can be generated by any iterative scheme i.e., Newton's method, Halley's method etc. In the last few years, many researchers worked on polynomiography and generated new and nice looking images through different algorithms. Soleimani *et al.* [33], suggested some new iterative methods free from derivatives and then presented their fractal behaviors by means of their basins of attraction. In [20], the authors generated beautiful polynomiographs using Ishikawa and Mann iterations that were quite new and looked aesthetically pleasing comparing to the ones from standard Picard iteration. In 2016, the authors modified the Abbasbandy's method and then presented polynomiographs through the modified method [19]. Gdawiec [7] in 2017, used three different approaches, i.e., affine and s-convex combination, the use of iteration processes from fixed point theory and multi-step polynomiography and obtained new and diverse fractal patterns that have many applications in textile and ceramics patterns. In [23], the authors presented some new graphical objects obtained by the use of escape time algorithm and the derived criteria. They presented graphical examples by means of Jungck-CR iteration process with s-convexity. In 2019, Kalantari and Hans Lee [18] introduced new ways of creating mathematical art through a novel Newton-Ellipsoid method for solving polynomial equations. The nature of polynomiographs under Newton-Ellipsoid seems to be very different from the other images, which opens the possibility of generating novel artistic images.

To generate a polynomiograph via computer program, we have to select an initial rectangle R containing the roots of the polynomial. Then for each point z_0 in the region, we run an iterative method, and then color the point corresponding to z_0 is depended upon the approximate convergence of the truncated orbit to a root, or lack thereof. The resolution of the image depends on our discretization of the rectangle R. For example, discretizing R into a 2000 by 2000 grid yields a high-resolution image. Polynomiography has vast applications in many fields of science and art.

According to Fundamental Theorem of Algebra, any complex polynomial with complex coefficients $\{a_n, a_{n-1}, \ldots, a_1, a_0\}$:

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0 \tag{26}$$

or by its zeros (roots) $\{r_1, r_2, \ldots, r_{n-1}, r_n\}$:

$$p(z) = (z - r_1)(z - r_2)\ldots(z - r_n) \tag{27}$$

of degree n has n roots (zeros) which may or may not be distinct. The degree of polynomial describes the number of basins of attraction and localization of basins can be controlled by placing roots on the complex plane manually.

Usually, the polynomiographs are colored and their coloring depends upon the number of iterations needed to approximate the roots of some polynomial with a given accuracy and a chosen iterative scheme. The general and base algorithm for the generation of polynomiograph is presented in the following Algorithm 4.

In the Algorithm 4, the convergence test $(z_n + 1, z_n, \epsilon)$ returns TRUE if the applied method has converged to the root, and FALSE otherwise. The most common and widely used

Algorithm 4 Polynomiograph's Generation

Input: $p \in \mathbb{C}$ — polynomial, $A \subset \mathbb{C}$ — area, k — maximum number of iterations, I — iteration method, β — parameter for the iteration, ϵ — accuracy, colormap $[0 \ldots C-1]$ — colormap with C colors.

Output: Polynomiograph for the complex polynomial p in area A.

for $z_0 \in A$ **do**
  $i = 0$
  while $i \le k$ **do**
    $z_{n+1} = I(z_n)$
    if $|z_{n+1} - z_n| < \epsilon$ **then**
      break
    $i = i + 1$
  color z_0 by means of colormap.

convergence test has the following standard form:

$$|z_{n+1} - z_n| < \varepsilon, \tag{28}$$

where z_{n+1} and z_n are two consecutive points in an iteration process and $\varepsilon > 0$ is a given accuracy. In this article we also use the stopping criteria (28).

Using newly developed iterative algorithms (for $\beta = 1$), we present polynomiographs of the following complex polynomials and comparing them with other well-known existing iterative algorithms.

$$p_1(z) = z^3 - 1, \quad p_2(z) = (z^3 - 1)^2$$

The colormap used for the coloring of iterations in the generation of polynomiographs is presented in the following figure:

FIGURE 1. The colormap used for generating polynomiographs.

Example 1: Polynomiographs for the Polynomial $p_1(z)$ by Means of Different Iterative Methods

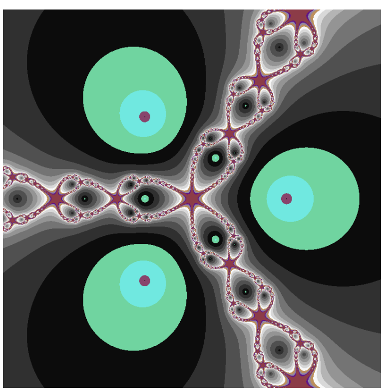

FIGURE 2. Polynomiograph for $p_1(z)$ using (NM).

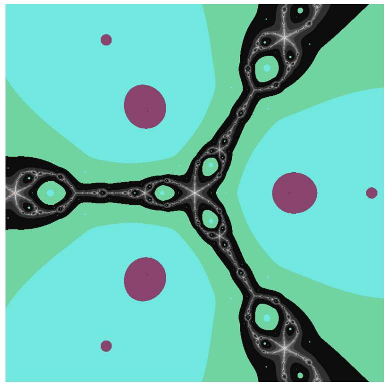

FIGURE 3. Polynomiograph for $p_1(z)$ using (JM).

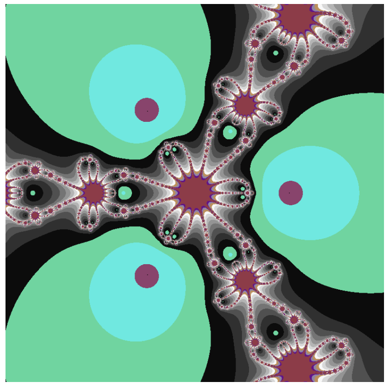

FIGURE 4. Polynomiograph for $p_1(z)$ using (HHM).

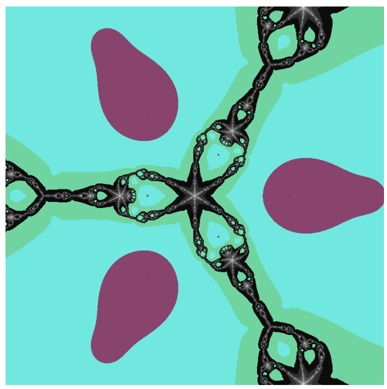

FIGURE 5. Polynomiograph for $p_1(z)$ using (MHM).

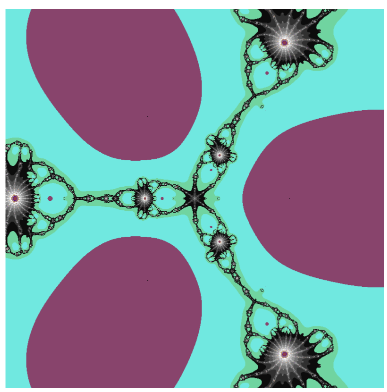

FIGURE 6. Polynomiograph for $p_1(z)$ using Algorithm 1.

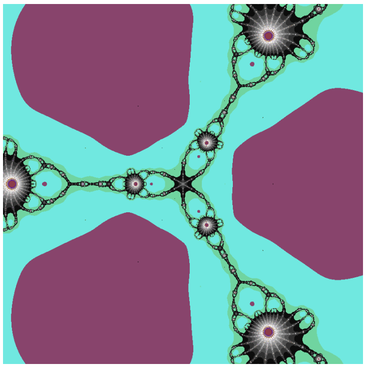

FIGURE 7. Polynomiograph for $p_1(z)$ using Algorithm 2.

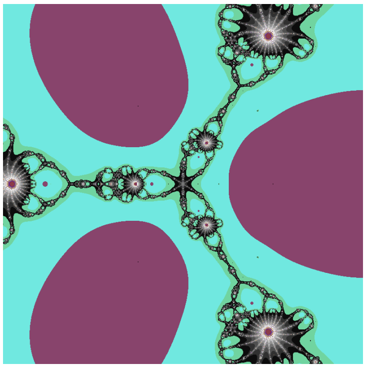

FIGURE 8. Polynomiograph for $p_1(z)$ using Algorithm 3.

Example 2: Polynomiographs for the Polynomial $p_2(z)$ by Means of Different Iterative Methods

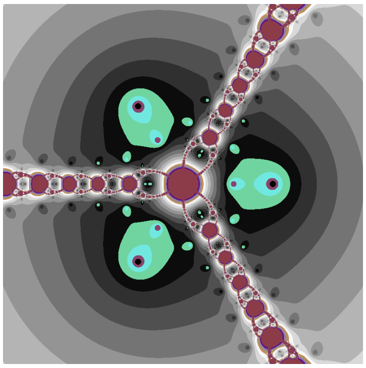

FIGURE 9. Polynomiograph for $p_2(z)$ using (NM).

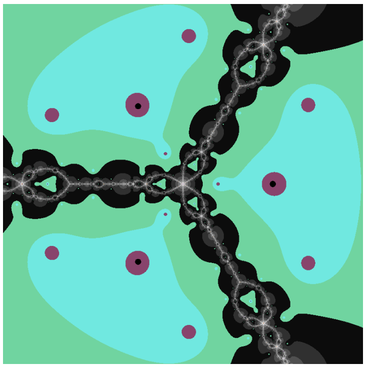

FIGURE 10. Polynomiograph for $p_2(z)$ using (JM).

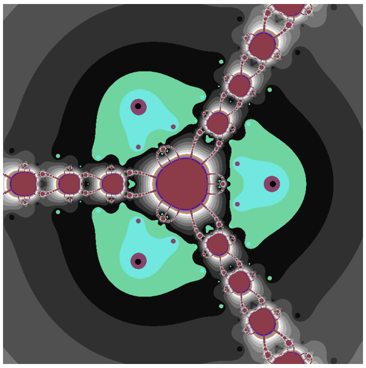

FIGURE 11. Polynomiograph for $p_2(z)$ using (HHM).

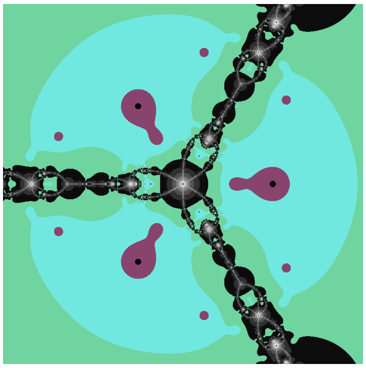

FIGURE 12. Polynomiograph for $p_2(z)$ using (MHM).

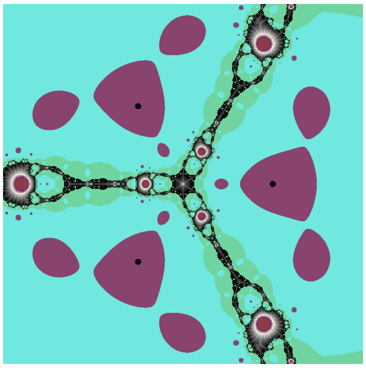

FIGURE 13. Polynomiograph for $p_2(z)$ using Algorithm 1.

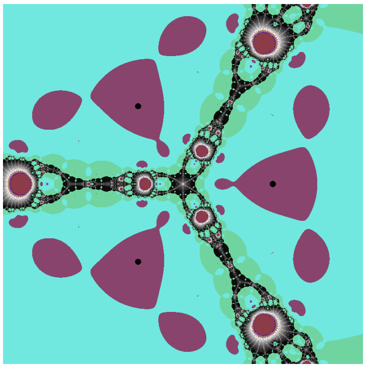

FIGURE 14. Polynomiograph for $p_2(z)$ using Algorithm 2.

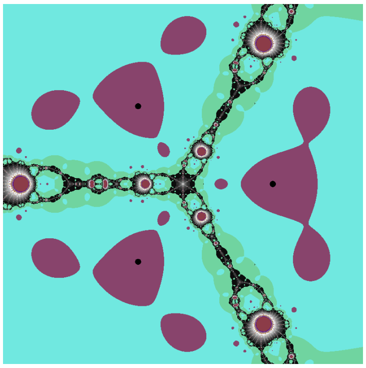

FIGURE 15. Polynomiograph for $p_2(z)$ using Algorithm 3.

In examples$(1-2)$, which include figures$(2-15)$, Polynomiographs of different complex polynomials for Newton's method (NM), Golbabai and Javidi's method (JM), Househölder's method (HHM), modified Halley's method (MHM) and our developed iterative algorithms (for $\beta = 1$) have been shown. When we look at the generated images, we can read two important characteristics. The first one is the speed of convergence of the iterative scheme, i.e., the color of each point gives us information on how many iterations were performed by the iterative scheme to reach the root. The second characteristic is the dynamics of the adopted iterative scheme. Low dynamics are in those areas where the variation of colors is small, whereas in areas with a large variation of colors the dynamics are much high.

The black color in images locates those places where the solution cannot be achieved for the given number of iterations and one can observe that the images generated through proposed iterative algorithms scarcely find such places which prove their best performance. The areas of the same colors in above figures indicate the same number of iterations required to determine the solution and they look similar to the contour lines on the map.

All these figures have been generated using the computer program Mathematica 10.0 by taking $\varepsilon = 0.01$ and $k = 15$ where ε shows the accuracy of the given root and k represents the upper bound of the number of iterations.

VI. CONCLUDING REMARKS

Based on variational iteration technique, some new iterative algorithms for the solution of one dimensional non-linear equations have been established, having ninth order of convergence. By using some test examples, the performance and efficiency of the proposed iterative algorithms have been analyzed. Tables 1–2 show the best performance of the proposed iterative algorithms as compare to other well-known existing iterative algorithms in terms of accuracy, speed, time, number of iterations and computational order of convergence. We have also presented the graphical comparison of the proposed iterative algorithms with other well-known iterative algorithms by means of polynomiographs of some complex polynomials which describe the fractal behavior and dynamical aspects of the proposed iterative algorithms. The variational iteration technique can be applied to derive a broad range of new algorithms for solving one dimensional non-linear equations.

REFERENCES

[1] S. Abbasbandy, "Improving Newton–Raphson method for nonlinear equations by modified adomian decomposition method," *Appl. Math. Comput.*, vol. 145, nos. 2–3, pp. 887–893, Dec. 2003.

[2] 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, doi: 10.1109/ACCESS.2019.2906470.

[3] R. Behl and E. Martínez, "A new high-order and efficient family of iterative techniques for nonlinear models," *Complexity*, vol. 2020, pp. 1–11, Jan. 2020, doi: 10.1155/2020/1706841.

[4] R. L. Burden and J. D. Faires, *Numerical Analysis*. Pacific Grove, CA, USA: Brooks/Cole Pub. Co., 1997.

[5] A. Cayley, "The Newton-Fourier imaginary problem," *Amer. J. Math.*, vol. 2, no. 1, p. 97, Mar. 1879.

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

[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] A. Golbabai and M. Javidi, "A third-order Newton type method for nonlinear equations based on modified homotopy perturbation method," *Appl. Math. Comput.*, vol. 191, no. 1, pp. 199–205, Aug. 2007.

[9] 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.

[10] J.-H. He, "Homotopy perturbation technique," *Comput. Math. Appl. Mech. Eng.*, vol. 178, pp. 257–262, Aug. 1999.

[11] J.-H. He, "Variational iteration method—Some recent results and new interpretations," *J. Comput. Appl. Math.*, vol. 207, no. 1, pp. 3–17, Oct. 2007.

[12] J.-H. He, "Variational iteration method–A kind of non-linear analytical technique: Some examples," *Int. J. Non-Linear Mech.*, vol. 34, no. 4, pp. 699–708, Jul. 1999.

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

[14] 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.

[15] M. Inokuti, H. Sekine, and T. Mura, "General use of the Lagrange multiplier in non-linear mathematical physics," in *Variational Methods in the Mechanics of Solids*, S. Nemat-Nasser, Ed. Oxford, U.K.: Pergamon, 1978, pp. 156–162.

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

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

[18] 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.

[19] S. M. Kang, A. Naseem, W. Nazeer, M. Munir, and C. Yong, "Polynomiography via modified Abbasbanday's method," *Int. J. Math. Anal.*, vol. 11, no. 3, pp. 133–149, 2017.

[20] W. Kotarski, K. Gdawiec, and A. Lisowska, "Polynomiography via Ishikawa and Mann iterations," in *Advances in Visual Computing* (Lecture Notes in Computer Science), vol. 7431, G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller, and M. Papka, Eds. Berlin, Germany: Springer, 2012, pp. 305–313.

[21] J. Kuo, "The improvements of modified Newton's method," *Appl. Math. Comput.*, vol. 189, pp. 602–609, Jun. 2007.

[22] J. Kou, Y. Li, and X. Wang, "A composite fourth-order iterative method for solving non-linear equations," *Appl. Math. Comput.*, vol. 184, no. 2, pp. 471–475, Jan. 2007.

[23] Y. C. Kwun, M. Tanveer, W. Nazeer, K. Gdawiec, and S. M. Kang, "Mandelbrot and Julia sets via Jungck-CR iteration with s—Convexity," *IEEE Access*, vol. 7, pp. 12167–12176, 2019, doi: 10.1109/ACCESS.2019.2892013.

[24] B. Mandelbrot, *The Fractal Geometry of Nature*. New York, NY, USA: W. H. Freeman Company, 1983.
[25] W. Nazeer, A. Naseem, S. M. Kang, and Y. C. Kwun, "Generalized Newton Raphson's method free from second derivative," *J. Non-linear Sci.*, vol. 9, no. 1, pp. 2823–2831, 2016.
[26] W. Nazeer, M. Tanveer, S. M. Kang, and A. Naseem, "A new householder's method free from second derivatives for solving non-linear equations and polynomiography," *J. Non-Linear Sci. Appl.*, vol. 9, no. 3, pp. 998–1007, 2016.
[27] M. A. Noor, "New classes of iterative methods for nonlinear equations," *Appl. Math. Comput.*, vol. 191, no. 1, pp. 128–131, Aug. 2007.
[28] M. A. Noor and F. A. Shah, "Variational iteration technique for solving nonlinear equations," *J. Appl. Math. Comput.*, vol. 31, nos. 1–2, pp. 247–254, Sep. 2009.
[29] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*. New York, NY, USA: Springer-Verlag, 2000.
[30] 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.
[31] M. Shacham, "Numerical solution of constrained non-linear algebraic equations," *Int. J. Numer. Methods Eng.*, vol. 23, no. 8, pp. 1455–1481, Aug. 1986.
[32] 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.
[33] F. Soleimani, F. Soleymani, and S. Shateyi, "Some iterative methods free from derivatives and their basins of attraction for nonlinear equations," *Discrete Dyn. Nature Soc.*, vol. 2013, pp. 1–10, Apr. 2013, doi: 10.1155/2013/301718.
[34] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 3rd ed. New York, NY, USA: Springer-Verlag, 2002.
[35] J. F. Traub, *Iterative Methods for the Solution of Equations*. Upper Saddle River, NJ, USA: Prentice-Hall, 1964.
[36] 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.
[37] 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, Nov. 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 authored or coauthored 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 Professor with the University of Management and Technology, Lahore, Pakistan. He has authored or coauthored 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 authored or coauthored 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.