Research Article

An Iterative Solver in the Presence and Absence of Multiplicity for Nonlinear Equations

Fazlollah Soleymani,¹ Stanford Shateyi,² and Gülcan Özkum³

¹ Young Researchers and Elite Club, Zahedan Branch, Islamic Azad University, Zahedan, Iran

² Department of Mathematics, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa

³ Department of Mathematics, Science and Letter Faculty, Kocaeli University, Umuttepe Campus, Kocaeli, Turkey

Correspondence should be addressed to Stanford Shateyi; stanford.shateyi@univen.ac.za

Received 17 June 2013; Accepted 1 October 2013

Academic Editors: S. Amat, M. Inc, Z. Mukandavire, S. Ostrovska, X. Qin, and S. A. Tersian

Copyright © 2013 Fazlollah Soleymani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We develop a high-order fixed point type method to approximate a multiple root. By using three functional evaluations per full cycle, a new class of fourth-order methods for this purpose is suggested and established. The methods from the class require the knowledge of the multiplicity. We also present a method in the absence of multiplicity for nonlinear equations. In order to attest the efficiency of the obtained methods, we employ numerical comparisons alongside obtaining basins of attraction to compare them in the complex plane according to their convergence speed and chaotic behavior.

1. Introduction

There are several methods for computing a zero x^* with multiplicity m of a nonlinear equation $f(x) = 0$; see, for example, [1–4]. Newton's iterative method is only of first order unless it is modified to obtain the second order of convergence. This modification requires knowledge of the multiplicity. Traub in [5] has suggested the use of any method for $f^{(m)}(x)$ or $h(x) := f(x)/f'(x)$. Any such method will require higher derivatives than the corresponding one for simple zeros. In general, constructing iterative methods for finding multiple zeros is not an easy task and careful attention is required in their developments.

This work tries to deal with the solution of nonlinear equations when the multiplicity of the roots is available or can be approximated and also when m is very high or not available. It then will be clear that for the choice $m = 1$, the corresponding method can easily be applied for simple zeros.

In this paper, we consider first the following class of methods for simple zeros (with $c_k = f^{(k)}(x^*)/k!f'(x^*)$, ($k = 2, 3, 4$):

$$y_n = x_n - \frac{2f(x_n)}{3f'(x_n)},$$

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

$$\times \{G(t_n)\},$$

where $t_n = f'(y_n)/f'(x_n)$, $G(1) = 1$, $G'(1) = -1/4$, and $G''(1) = 7/4$ with the following error equation:

$$e_{n+1} = \frac{1}{243}$$

$$\times (-243c_2c_3 + 27c_4 + c_2^3(399 + 32G^{(3)}(1)))e_n^4 \quad (2)$$

$$+ O(e_n^5).$$

We then extend (1) for multiple zeros without lowering the convergence order or applying much more computational evaluations of the function or derivatives. Thus, the proposed

scheme will be of order four for multiple zeros with two evaluations of the first-order derivative and one evaluation of the function, which clearly implies the consistency of the general solver with the optimality conjecture of Kung-Traub [6]. Furthermore, we will discuss some qualitative issues using the attraction basins as a criterion for comparison. To this end, we refer the readers to see [7–9] on obtaining a background on this matter regarding a rational map on the Riemann sphere, Julia set, Newton's fractals, the definitions of dynamic and chaotic behaviors, and so forth.

The remaining sections of the paper are organized as follows. Section 2 will shortly report some of the existing methods for multiple roots using three functional evaluations per full step. It will be followed by Section 3 wherein the main contribution of the paper lies, as a general fourth-order solver in the presence and the absence of multiplicity for nonlinear equations. Section 4 will be devoted to the comparison of different methods based on basins of attraction to observe the chaotic behaviors of the corresponding fractals produced by each iterative method in the complex plane. Numerical reports will be furnished in Section 5 and the conclusion of the study has been presented in the last section.

2. A Brief Review

Here, we mention some of the famous schemes for multiple zeros using three functional evaluations per full cycle shortly. The old and well-known third-order method of Dong in [10] can be given as

$$\begin{aligned} y_n &= x_n - \sqrt{m} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - m \left(1 - \frac{1}{\sqrt{m}} \right)^{1-m} \frac{f(y_n)}{f'(x_n)}. \end{aligned} \quad (3)$$

Dong has also proposed the following third-order method in [10]:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - \frac{f(x_n)}{(m/(m-1))^{m+1} f'(y_n) + (m-m^2-1)/(m-1)^2 f'(x_n)}. \end{aligned} \quad (4)$$

In 2010, J. R. Sharma and R. Sharma in [11] presented the following quartically convergent method:

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \frac{m}{8} \\ &\times \left((m^3 - 4m + 8) - (m+2)^2 \left(\frac{m}{m+2} \right)^m \frac{f'(x_n)}{f'(y_n)} \right. \\ &\times \left. \left(2(m-1) - (m+2) \left(\frac{m}{m+2} \right)^m \frac{f'(x_n)}{f'(y_n)} \right) \right) \\ &\times \frac{f(x_n)}{f'(x_n)}. \end{aligned} \quad (5)$$

Recently in 2011, Zhou et al. in [12] presented the following optimal fourth-order scheme:

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n \\ &- \frac{m}{8} \left((m^3 + 6m^2 + 8m + 8) \right. \\ &+ m^3 \left(\frac{m+2}{m} \right)^{2m} \left(\frac{f'(y_n)}{f'(x_n)} \right)^2 \\ &- 2m^2 (m+3) \left(\frac{m+2}{m} \right)^m \frac{f'(y_n)}{f'(x_n)} \left. \right) \frac{f(x_n)}{f'(x_n)}. \end{aligned} \quad (6)$$

All the reported methods above require the multiplicity of the zeros and three functional evaluations per computing step to proceed. Clearly, only methods (5) and (6) are optimal in the sense of Kung-Traub for constructing multipoint methods without memory for solving nonlinear equations. Some other new developments for multiple zeros can be found in [13, 14].

3. A New General Solver for Multiple Zeros

Let us consider the following class for multiple zeros as a generalized form of (1) and by using weight function approach:

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - \frac{2f(x_n) + (y_n - x_n)(f'(y_n) + f'(x_n))}{2f'(y_n)} \\ &\times \{H(\tau_n)\}, \quad \tau_n = \frac{f'(y_n)}{f'(x_n)}. \end{aligned} \quad (7)$$

We now show that this class is of fourth order for multiple roots under certain conditions on the weight function.

Theorem 1. Let $x^* \in D$ be a multiple zero of a sufficiently differentiable function $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ for an open interval D with the multiplicity m , which includes x_0 as an initial approximation of x^* . Then, the class of methods without memory (7) is of optimal local order four, when

$$\begin{aligned} H(p^{(m-1)/m}) &= -\frac{m(2+m)p}{-2+2p+mp}, \\ H'(p^{(m-1)/m}) &= \frac{1}{4} \frac{m^2(-2m(4+m) + (2+m)^3 p)}{(-2+(2+m)p)^2}, \\ H''(p^{(m-1)/m}) &= -\frac{1}{4} \frac{m^3(4m(2+m) - 4m(8+m(5+m))p + (2+m)^4 p^2)}{p(-2+(2+m)p)^3}, \end{aligned} \quad (8)$$

and $|H^{(3)}(p^{(m-1)/m})| < \infty$, where $p = (m/(m+2))^m$.

Proof. To find the asymptotic error constant of (7) where $C_j = (m)!/(m+j)! \times f^{(m+j)}(x^*)/f^{(m)}(x^*)$, $j \geq 1$, we expand any terms of (7) around the multiple zero x^* in the n th iterate whence $e_n = x_n - x^*$. Clearly, for such symbolic computations, a Computer Algebra System such as [15] needs to be used.

A Taylor expansion around x^* yields $f(x_n) = (f^{(m)}(x^*)/m!)e_n^m(1 + \sum_{j=1}^{\infty} C_j e_n^j)$ and $f'(x_n) = (f^{(m)}(x^*)/(m-1)!)e_n^{m-1}(1 + \sum_{j=1}^{\infty} ((m+j)/m)C_j e_n^j)$, so that, using algebraic software again,

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2} \frac{f(x_n)}{f'(x_n)} = x^* + \frac{m}{m+2} e_n + \frac{2C_1}{m(m+2)} e_n^2 \\ &\quad - \frac{((m+1)C_1^2 - 2mC_2)}{m^2(m+2)} e_n^3 \\ &\quad + \frac{2((-3m^2 - 4m)C_1C_2 + (m^2 + 2m + 1)C_1^3 + 3m^2C_3)}{m^3(m+2)} \\ &\quad \times e_n^4 + O(e_n^5). \end{aligned} \quad (9)$$

Now we easily have $f'(y_n) = e_n^m((f'(\alpha)(m/(2+m))^m(2+m))/(m!e_n) + (C_1f'(\alpha)(m/(2+m))^m(-4+2m+3m^2+m^3))/(m^2m!) + (f'(\alpha)(m/(2+m))^m(-4C_1^2(-2+m) + C_2m^2(-8+4m+4m^2+m^3))e_n)/(m^4m!) + O(e_n^2))$. Furthermore, we have by Taylor expansion

$$\begin{aligned} \tau_n = \frac{f'(y_n)}{f'(x_n)} &= p^{(m-1)/m} - \frac{4p}{m^3} C_1 e_n \\ &\quad + \left(\frac{4(m^2+2)p}{m^5} C_1^2 - \frac{8p}{m^3} C_2 \right) e_n^2 \end{aligned}$$

$$\begin{aligned} &+ \left(-\frac{8(m^4 - m^3 + 5m^2 + m + 6)p}{3m^7} C_1^3 \right. \\ &\quad + \frac{8(m^2+4)p}{m^5} C_1 C_2 \\ &\quad \left. - \frac{8(m^2+6m+6)p}{m^3(m+2)} C_3 \right) e_n^3 + O(e_n^4). \end{aligned} \quad (10)$$

Thus by using (10), we obtain

$$\begin{aligned} H(\tau_n) &= H(p^{(m-1)/m}) + H'(p^{(m-1)/m})(\tau_n - p^{(m-1)/m}) \\ &\quad + \frac{H''(p^{(m-1)/m})}{2} (\tau_n - p^{(m-1)/m})^2 \\ &\quad + \frac{H^{(3)}(p^{(m-1)/m})}{6} (\tau_n - p^{(m-1)/m})^3 \\ &\quad + O(e_n^4) \\ &= -\frac{m(2+m)p}{-2+2p+mp} \\ &\quad - \frac{C_1(-2m^2p + 6p^2m^2 + p^2m^3 + 12p^2m - 8mp + 8p^2)}{m(-2+2p+mp)^2} \\ &\quad \times e_n + \dots + O(e_n^5). \end{aligned} \quad (11)$$

And also,

$$\begin{aligned} &\frac{2f(x_n) + (y_n - x_n)(f'(y_n) + f'(x_n))}{2f'(y_n)} \\ &= \left(-\frac{-2+2p+mp}{p(2+m)^2} \right) e_n \\ &\quad + \left(\frac{C_1(-2m^2 + 4m^2p + m^3p + 4mp + 8 - 4m)}{p(2+m)^3m^2} \right) e_n^2 \\ &\quad + \frac{1}{m^3(m+2)^4} \\ &\quad \times \left(-C_1^2m^5 + 2m^5C_2 + 12C_2m^4 - 4\frac{C_2m^4}{p} \right. \\ &\quad \left. - 7C_1^2m^4 + 2\frac{C_1^2m^4}{p} + 24C_2m^3 - 18C_1^2m^3 \right. \\ &\quad \left. + 10\frac{C_1^2m^3}{p} - 16\frac{C_2m^3}{p} \right. \\ &\quad \left. + 8\frac{C_1^2m^2}{p} - 20C_1^2m^2 \right) e_n^3 \end{aligned}$$

$$\begin{aligned}
& + 16C_2m^2 + 32\frac{C_2m}{p} - 16\frac{C_1^2m}{p} \\
& - 8C_1^2m - 32\frac{C_1^2}{p}\Big)e_n^3 + \dots + O(e_n^5).
\end{aligned} \tag{12}$$

By using (9), (11), and (12), class (7) satisfies the error equation:

$$\begin{aligned}
e_{n+1} &= \left(\left(-\frac{32H^{(3)}(p^{(m-1)/m})((-2+2p+mp)p^2)}{3m^9(2+m)^2} \right. \right. \\
&+ \left(\left(12-2m+2m^2+2m^3+m^4 \right. \right. \\
&+ \frac{1536}{(2+m)^4(-2+(2+m)p)^3} \\
&+ \frac{192(12+m(4+m))}{(2+m)^4(-2+(2+m)p)^2} \\
&+ \left. \left. \frac{48(24+m(20+m(8+m)))}{(2+m)^4(-2+(2+m)p)} \right) (3m^5)^{-1} \right) \Big) C_1^3 \\
&- \frac{C_1C_2}{m} + \frac{mC_3}{(m+2)^2} \Big) e_n^4 + O(e_n^5).
\end{aligned} \tag{13}$$

This shows that our class (7) reaches the local quartically convergence using three evaluations. The proof is complete. $\square$

The proposed class (7) uses only three functional evaluations per full cycle to achieve fourth order of convergence based on Theorem 1, which implied $4^{1/3} \approx 1.587$ as its classical computational efficiency index that is the same as

(5) and (6) and is higher than $3^{1/3} \approx 1.442$ of methods (3) and (4).

A simple optimal and efficient method from class (7) could be presented in what follows:

$$\begin{aligned}
y_n &= x_n - \frac{2m}{m+2} \frac{f(x_n)}{f'(x_n)}, \\
x_{n+1} &= y_n - \frac{2f(x_n) + (y_n - x_n)(f'(y_n) + f'(x_n))}{2f'(y_n)} \\
&\times \{H(\tau_n)\},
\end{aligned} \tag{14}$$

by choosing

$$\begin{aligned}
H(\tau_n) &= -\frac{m(2+m)p}{-2+2p+mp} \\
&+ \frac{-2m(4+m) + (2+m)^3p}{4(-2+(2+m)p)^2} \\
&\times \left(\frac{f'(y_n)}{f'(x_n)} - p^{(m-1)/m} \right) \\
&+ \frac{m^3(2+m)(-4m+4m(3+m)p - (2+m)^3p^2)}{8p(-2+(2+m)p)^3} \\
&\times \left(\frac{f'(y_n)}{f'(x_n)} - p^{(m-1)/m} \right)^2.
\end{aligned} \tag{15}$$

Another optimal and efficient method from class (7) with a simplified error equation could be presented in what follows:

$$\begin{aligned}
y_n &= x_n - \frac{2m}{m+2} \frac{f(x_n)}{f'(x_n)}, \\
x_{n+1} &= y_n - \frac{2f(x_n) + (y_n - x_n)(f'(y_n) + f'(x_n))}{2f'(y_n)} \\
&\times \{H(\tau_n)\},
\end{aligned} \tag{16}$$

by choosing

$$\begin{aligned}
H(\tau_n) &= -\frac{m(2+m)p}{-2+2p+mp} + \frac{-2m(4+m) + (2+m)^3p}{4(-2+(2+m)p)^2} \left(\frac{f'(y_n)}{f'(x_n)} - p^{(m-1)/m} \right) \\
&+ \frac{m^3(2+m)(-4m+4m(3+m)p - (2+m)^3p^2)}{8p(-2+(2+m)p)^3} \left(\frac{f'(y_n)}{f'(x_n)} - p^{(m-1)/m} \right)^2 \\
&+ \frac{1}{192p^2(-2+(2+m)p)^4} m^4 \\
&\times (-8m(2+m)(8+m(2+m(6+m(4+m))))
\end{aligned}$$

$$\begin{aligned}
& + 12m(2+m)(24+m(12+m(14+m(14+m(6+m))))))p \\
& - 6m(192+m(192+m(136+m(124+m(94+m(42+m(10+m)))))))p^2 \\
& + (2+m)^5(12+m(-2+m(2+m(2+m))))p^3 \left(\frac{f'(y_n)}{f'(x_n)} - p^{(m-1)/m} \right)^3.
\end{aligned} \tag{17}$$

This method has the simplified error equation:

$$e_{n+1} = \left(-\frac{C_1 C_2}{m} + \frac{m C_3}{(m+2)^2} \right) e_n^4 + O(e_n^5). \tag{18}$$

An important challenge in the iterative methods based on the known multiplicity is to find the order of multiplicity correctly. However, most algorithms to determine the order of multiplicity may lead to mutually opposite requirements; for example,

(i) Traub in [5] showed that

$$m \approx \frac{\log |f(x)|}{\log |f(x)/f'(x)|}, \tag{19}$$

when x is very close to the multiple root of f ;

(ii) Lagouanelle in [16] proposed the following approximate formula:

$$m \approx \frac{f'(x)^2}{f'(x)^2 - f(x)f''(x)}, \tag{20}$$

when once again x is very close to the multiple root of f .

The above-mentioned ways demand a very close approximation to calculate a multiplicity of high accuracy. On the other hand, to find a very close approximation to a multiple root, it is necessary to use precise multiplicity. Sometimes, both of the requirements cannot be attained at the same time.

Taking into account the opposite demands mentioned and additional calculations to find multiplicity, in those cases where we cannot provide an accurate multiplicity, it is sometimes better to apply a method which does not explicitly require the order of multiplicity, in spite of its lower computational efficiency arising from additional functional evaluations per iteration.

Hence, by applying the optimal class of methods (1) on the transformation $h(x) := f(x)/f'(x)$, we can easily extend it for dealing with multiple roots in the absence of multiplicity, when high precision alongside high order is needed. For a simple weight function in (1) which is easily constructed by the following Mathematica command:

```
FullSimplify@
DSolve[{G'''[t] == 0, G[1] == 1,
G'[1] == -1/4, G''[1] == 7/4}, G[t], t];
weight[t_] = G[t]/. %[[1]]
```

We obtain

$$\begin{aligned}
y_n &= x_n - \frac{2}{3} \frac{f(x_n)}{f'(x_n)}, \\
x_{n+1} &= y_n - \frac{2f(x_n) + (y_n - x_n)(f'(y_n) + f'(x_n))}{2f'(y_n)} \\
&\quad \times \left\{ \frac{17}{8} - 2t_n + \frac{7t_n^2}{8} \right\}, \quad t_n = \frac{f'(y_n)}{f'(x_n)},
\end{aligned} \tag{21}$$

for simple zeros and the following alternative for multiple zeros in the absence of multiplicity:

$$\begin{aligned}
y_n &= x_n + \frac{2f(x_n)f'(x_n)}{-3f'(x_n)^2 + 3f(x_n)f''(x_n)}, \\
x_{n+1} &= y_n - \left(\left(\frac{17}{8} - 2\psi_n + \frac{7\psi_n^2}{8} \right) \right. \\
&\quad \times \left(\frac{f(x_n)(2f'(x_n) + (x_n - y_n)f''(x_n))}{f'(x_n)^2} \right. \\
&\quad \left. \left. + (x_n - y_n)(-2 + \nu_n) \right) \right) \\
&\quad \times (2(1 - \nu_n))^{-1},
\end{aligned} \tag{22}$$

wherein

$$\begin{aligned}
u_n &= \frac{f(x_n)f''(x_n)}{f'(x_n)^2}, \\
v_n &= \frac{f(y_n)f''(y_n)}{f'(y_n)^2}, \\
\psi_n &= \frac{1 - \nu_n}{1 - u_n}.
\end{aligned} \tag{23}$$

Therefore, now we have an efficient method (22) of order four for finding the multiple roots too. Note that until now, we

have distinguished two kinds of methods: those which deal with a known order of multiplicity and others, such as (22), with no information on multiplicity.

4. The Dynamical Behavior of the Methods

We here investigate the comparison of iterative schemes in the complex plane using basins of attraction. It is well known that a fixed point is a point of a function that does not change under some transformation. We further recall that if we regard the evolution of a dynamical system as a series of transformations, then there may or may not be a point which remains fixed under each transformation. The final state, that a dynamical system evolves towards, corresponds to an attracting fixed point of the evolution function, but the two concepts are not equivalent because not all fixed points attract the evolution of nearby points.

The aim herein is to use the basin of attraction as another way for comparing the iteration algorithms; see, for example, [17].

In this section, we take a rectangle $D = [-3, 3] \times [-3, 3] \in \mathbb{C}$ and we assign a color to each point $z_0 \in D$ according to the multiple root at which the corresponding iterative method starting from z_0 converges, and we mark the point as black if the method does not converge. In this way, we distinguish the attraction basins by their colors for different methods.

The criteria, we have used in our Mathematica codes, are that the maximum number of iterations is 100. That is to say, if the method does not reach to the considered accuracy after 100 of its full iteration steps, we allocate the black color and also consider it as Not Convergent. The considered accuracy is the obtained residual of function to be less than 10^{-2} .

We have used methods (3), (5), (6), and (14) for some complex functions having multiple roots with known multiplicity.

For the first test, we have taken the following function.

Test Problem 1. We have

$$p_1(z) = (0.3z^5 - Iz^2 - 0.6)^5. \quad (24)$$

Equation (24) has five roots $-1.35689 + 0.836942I$, $0.121635 - 1.5236I$, $1.24874 + 0.58223I$, $0.527372 + 0.608843I$, and $-0.540858 - 0.504419I$ of the multiplicity five. Based on Figures 1 and 2 we can see that method (14), method (5), and method (3) are the good ones. Method (14) has larger basins in the areas corresponding to each root.

The second test problem is a polynomial as follows. Despite a few diverging points on the Julia set, we find that the basins of attraction for each root are largest for method (14).

Test Problem 2. We have

$$p_2(z) = (Iz^4 - 0.69z - 3)^4. \quad (25)$$

The zeros for this test are $1.28486 - 0.570612I$, $0.436728@ + 1.28774I$, $-0.577583 - 1.1469I$, and

$-1.14401 + 0.429775I$ of the multiplicity four. The results are presented in Figures 3 and 4. The performance of the methods (5) and (14) are acceptable. Methods (3), (5), (6), and (14) are quite the same though (14) has larger basins again.

The third test problem is chosen as follows.

Test Problem 3. We have

$$p_3(z) = (z^6 + z)^2. \quad (26)$$

The roots are $-0.309017 + 0.951057I$, $-0.309017 - 0.951057I$, $0.809017 + 0.587785I$, $0.809017 - 0.587785I$, and $-1, 0$, with the known multiplicity 2. The results are presented in Figures 5 and 6. The performance of the methods (5) and (14) are acceptable. The worse methods in this case are (3) and (6).

5. Numerical Reports

In this section, we exhibit numerical results showing the behavior of the methods in this paper with roots of known multiplicity m . Iterative methods (3), (4), (5), (6), and (14) for multiple roots, which also require three functional evaluations, have been chosen. The list of test nonlinear functions including multiple zeros with their multiplicity is presented in Table 1.

The results are summarized in Tables 2 and 3 after three full iterations for two different initial guesses. As they show, the novel scheme is comparable with all of the methods. All numerical instances were performed using 500 digits floating point arithmetic. We have computed the root of each test function for the initial guess x_0 .

The tests show that the considered optimal methods generate results of approximately the same accuracy. The formulas in this section have been employed on several other nonlinear equations. The experience shows that there is no clear winner among the optimal methods (5), (6), and (14) in the sense that in different situations different methods may be the winners.

Following the results here and the basins of attraction and also considering the optimal efficiency index of method (14), we could conclude that class (7) is efficient and can be considered as a good tool to find multiple zeros iteratively.

Note that the application of such iterative methods for finding generalized inverses could be considered for future researches in this trend of study; see, for example, [18, 19].

6. Concluding Remarks

We have demonstrated the performance of a new class of fourth-order methods. Convergence analysis has proved that the new methods obtained from the class preserve their order of convergence. We do not have to evaluate the second-order derivative of the functions in the presented methods when the multiplicity is known. We have also presented a quartically convergent method in the absence of multiplicity for nonlinear equations.

FIGURE 1: The methods (3) (a) and (5) (b) for Test Problem 1.

FIGURE 2: The methods (6) (a) and (14) (b) for Test Problem 1.

FIGURE 3: The methods (3) (a) and (5) (b) for Test Problem 2.

FIGURE 4: The methods (6) (a) and (14) (b) for Test Problem 2.

FIGURE 5: The methods (3) (a) and (5) (b) for Test Problem 3.

TABLE 1: The test functions considered in this paper.

Test functions	Zeros	Multiplicity
$f_1(x) = ((\sin(x))^2 + x)^5$	$x_1^* = 0$	5
$f_2(x) = (1 + x + \cos(\pi x/2) - \sqrt{1-x^2})^3$	$x_2^* \approx -0.728584046444826 \dots$	3
$f_3(x) = ((\sin(x))^2 - x^2 + 1)^4$	$x_3^* \approx 1.404491648215341 \dots$	4

TABLE 2: Results of comparisons for different methods after three full iterations.

Values of functions	Guess	(3)	(4)	(5)	(6)	(14)
$ f_1(x_3) $	0.3	$0.7e-80$	$0.8e-98$	$0.1e-164$	$0.3e-118$	$0.2e-164$
$ f_2(x_3) $	-0.6	$0.7e-72$	$0.1e-81$	$0.2e-151$	$0.4e-151$	$0.3e-151$
$ f_3(x_3) $	1.3	$0.2e-108$	$0.5e-163$	$0.2e-230$	$0.1e-228$	$0.3e-230$

TABLE 3: Results of comparisons for different methods after three full iterations.

Values of functions	Guess	(3)	(4)	(5)	(6)	(14)
$ f_1(x_3) $	0.2	$0.2e-98$	$0.8e-120$	$0.7e-206$	$0.1e-128$	$0.8e-206$
$ f_2(x_3) $	-0.8	$0.3e-72$	$0.1e-111$	$0.7e-142$	$0.3e-141$	$0.9e-142$
$ f_3(x_3) $	2	$0.2e-47$	$0.6e-68$	$0.3e-99$	$0.5e-98$	$0.4e-99$

FIGURE 6: The methods (6) (a) and (14) (b) for Test Problem 3.

Due to the fact that the basin of attraction is a method to visually comprehend how an algorithm behaves as a function of the various starting points, thus the dynamical behavior of the proposed methods alongside the comparison with the existing methods has been given in Section 4 to clearly show the efficiency of the methods.

Conflict of Interests

The authors declare that they do not have any conflict of interests.

References

- [1] E. Halley, "A new, exact and easy method of finding the roots of equations generally and that without any previous reduction," *Philosophical Transactions of the Royal Society*, vol. 18, pp. 136–148, 1694.
- [2] E. Hansen and M. Patrick, "A family of root finding methods," *Numerische Mathematik*, vol. 27, no. 3, pp. 257–269, 1976.
- [3] B. Neta and A. N. Johnson, "High-order nonlinear solver for multiple roots," *Computers and Mathematics with Applications*, vol. 55, no. 9, pp. 2012–2017, 2008.
- [4] H. D. Victory Jr. and B. Neta, "A higher order method for multiple zeros of nonlinear functions," *International Journal of Computer Mathematics*, vol. 12, no. 3–4, pp. 329–335, 1983.
- [5] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New York, NY, USA, 1964.
- [6] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *Journal of the ACM*, vol. 21, no. 4, pp. 643–651, 1974.
- [7] C. Getz and J. Helmstedt, *Graphics with Mathematica: Fractals, Julia Sets, Patterns and Natural Forms*, Elsevier, 2004.
- [8] M. McClure, "Newton's method for complex polynomials," A preprint version of a, "Mathematical graphics" column from Mathematica in Education and Research, pp. 1–15.
- [9] E. W. Weisstein, "Newton's Method," 2006, <http://mathworld.wolfram.com/NewtonsMethod.html>.
- [10] C. Dong, "A family of multipoint iterative functions for finding multiple roots of equations," *International Journal of Computer Mathematics*, vol. 21, no. 3–4, pp. 363–367, 1987.
- [11] J. R. Sharma and R. Sharma, "Modified Jarratt method for computing multiple roots," *Applied Mathematics and Computation*, vol. 217, no. 2, pp. 878–881, 2010.
- [12] X. Zhou, X. Chen, and Y. Song, "Constructing higher-order methods for obtaining the multiple roots of nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 235, no. 14, pp. 4199–4206, 2011.
- [13] M. Sharifi, D. K. R. Babajee, and F. Soleymani, "Finding the solution of nonlinear equations by a class of optimal methods," *Computers and Mathematics with Applications*, vol. 63, no. 4, pp. 764–774, 2012.
- [14] J. R. Sharma and R. Sharma, "Modified Chebyshev-Halley type method and its variants for computing multiple roots," *Numerical Algorithms*, vol. 61, pp. 567–578, 2012.
- [15] S. Wolfram, *The Mathematica Book*, Wolfram Media, 5th edition, 2003.
- [16] J. L. Lagouanelle, "Sur une mtdode de calcul de l'ordre de multiplicit des zros d'un polynme," *Comptes Rendus de l'Académie des Sciences A*, vol. 262, pp. 626–627, 1966.
- [17] F. Soleimani, F. Soleymani, and S. Shateyi, "Some iterative methods free from derivatives and their basins of attraction," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 301718, 10 pages, 2013.
- [18] F. Soleymani and P. S. Stanimirović, "A higher order iterative method for computing the Drazin inverse," *The Scientific World Journal*, vol. 2013, Article ID 708647, 11 pages, 2013.
- [19] F. Toutounian and F. Soleymani, "An iterative method for computing the approximate inverse of a square matrix and the Moore-Penrose inverse of a non-square matrix," *Applied Mathematics and Computation*, vol. 224, pp. 671–680, 2013.