



Article

An Efficient Class of Traub–Steffensen-Type Methods for Computing Multiple Zeros

Deepak Kumar ¹ , Janak Raj Sharma ¹  and Clemente Cesarano ^{2,*}

¹ Department of Mathematics, Sant Longowal Institute of Engineering and Technology, Longowal 148106, Sangrur, India; deepak.babbi@gmail.com (D.K.); jrshira@yahoo.co.in (J.R.S.)

² Section of Mathematics, International Telematic University UNINETTUNO, Corso Vittorio Emanuele II, 39, 00186 Roma, Italy

* Correspondence: c.cesarano@uninettunouniversity.net

Received: 5 May 2019; Accepted: 22 May 2019; Published: 25 May 2019



Abstract: Numerous higher-order methods with derivative evaluations are accessible in the literature for computing multiple zeros. However, higher-order methods without derivatives are very rare for multiple zeros. Encouraged by this fact, we present a family of third-order derivative-free iterative methods for multiple zeros that require only evaluations of three functions per iteration. Convergence of the proposed class is demonstrated by means of using a graphical tool, namely basins of attraction. Applicability of the methods is demonstrated through numerical experimentation on different functions that illustrates the efficient behavior. Comparison of numerical results shows that the presented iterative methods are good competitors to the existing techniques.

Keywords: nonlinear equations; multiple roots; derivative-free method; convergence

MSC: 65B99; 65D99; 65H05

1. Introduction

We consider derivative-free methods to discover a multiple root (say, α) of multiplicity m , i.e., $f^{(r)}(\alpha) = 0, r = 0, 1, 2, \dots, m - 1$ and $f^{(m)}(\alpha) \neq 0$, of a nonlinear equation $f(x) = 0$.

Many higher-order methods, either independent or based on the modified Newton's method [1] with quadratically convergent

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

have been proposed and verified in the literature; see, for example: [2–15] and references therein. Such methods require the evaluations of derivatives of either linear order or linear and second order, or both. However, higher-order derivative-free methods to handle the case of multiple roots are yet to be explored. The main reason for the non-availability of such methods is due to the difficulty in obtaining their order of convergence. The derivative-free methods are important in the conditions when derivative of the function f is difficult to compute or is expensive to obtain. One such method free from derivative is the classical Traub–Steffensen method [16] which actually replaces the derivative f' in the classical Newton method with appropriate approximation based on difference quotient,

$$f'(x_n) \simeq \frac{f(x_n + \beta f(x_n)) - f(x_n)}{\beta f(x_n)}, \quad \beta \in \mathbb{R} - \{0\},$$

or

$$f'(x_n) \simeq f[x_n, w_n],$$

where $w_n = x_n + \beta f(x_n)$ and $f[x_n, w_n] = \frac{f(w_n) - f(x_n)}{w_n - x_n}$ is a first-order divided difference.

In this way the modified Newton method (1) becomes the modified Traub–Steffensen method

$$x_{n+1} = x_n - m \frac{f(x_n)}{f[x_n, w_n]}. \quad (2)$$

The modified Traub–Steffensen method (2) is a noteworthy improvement of Newton’s method, because it keeps quadratic convergence without using derivative.

In this work, we introduce a two-step family of third-order derivative-free methods for computing multiple zeros that require three evaluations of the function f per iteration. The iterative scheme uses the Traub–Steffensen iteration (2) in the first step and Traub–Steffensen-like iteration in the second step. The procedure is based on a simple approach of using weight factors in the iterative scheme. Many special methods of the family can be generated depending on the forms of weight factors. Efficacy of these methods is tested on various numerical problems of different natures. In the comparison with existing techniques that use derivative evaluations, the new derivative-free methods are computationally more efficient.

The rest of the paper is summarized as follows. In Section 2, the scheme of third-order methods is developed and its convergence is studied. Section 3 is divided into two parts: Sections 3.1 and 3.2. In Section 3.1, an initial approach concerning the study of the dynamics of the methods with the help of basins of attraction is presented. To demonstrate applicability and efficacy of the new methods, some numerical experiments are performed in Section 3.2. A comparison of the methods with existing ones of the same order is also shown in this subsection. Concluding remarks are given in Section 4.

2. The Method

With a known multiplicity $m \geq 1$, we consider the following two-step scheme for multiple roots based on the Traub–Steffensen method (2)

$$\begin{cases} y_n = x_n - m \frac{f(x_n)}{f[x_n, w_n]}, \\ x_{n+1} = y_n - H(u) \frac{f(x_n)}{f[x_n, w_n]}, \end{cases} \quad (3)$$

where the function $H(u) : \mathbb{C} \rightarrow \mathbb{C}$ is analytic in a neighborhood of ‘0’ with $u = \left(\frac{f(y_n)}{f(x_n)} \right)^{\frac{1}{m}}$. The convergence order is obtained by the following theorem:

Theorem 1. Let $f : \mathbb{C} \rightarrow \mathbb{C}$ be an analytic function in a region enclosing a multiple zero (say, α) with multiplicity m . Assume that initial approximation x_0 is sufficiently close to α , then the iteration scheme defined by (3) has attained third order of convergence, provided that $H(0) = 0$ and $H'(0) = m$.

Proof. Let the error at n -th iteration be $e_n = x_n - \alpha$. Using Taylor’s expansion of $f(x_n)$ about α , we have that

$$f(x_n) = \frac{f^{(m)}(\alpha)}{m!} e_n^m \left(1 + C_1 e_n + C_2 e_n^2 + C_3 e_n^3 + O((e_n)^4) \right), \quad (4)$$

where $C_k = \frac{m!}{(m+k)!} \frac{f^{(m+k)}(\alpha)}{f^{(m)}(\alpha)}$ for $k \in \mathbb{N}$.

Also

$$\begin{aligned} w_n - \alpha &= x_n - \alpha + \beta f(x_n) \\ &= e_n \left(1 + \frac{\beta f^{(m)}(\alpha)}{m!} e_n^{m-1} \right) + \frac{\beta f^{(m)}(\alpha)}{m!} e_n^m \left(C_1 e_n + C_2 e_n^2 + C_3 e_n^3 + O((e_n)^4) \right). \end{aligned}$$

Then Taylor's expansion of $f(w_n)$ about α is given as

$$f(w_n) = \frac{f^{(m)}(\alpha)}{m!} (w_n - \alpha)^m \left(1 + C_1(w_n - \alpha) + C_2(w_n - \alpha) + C_3(w_n - \alpha) + O((w_n - \alpha)^4) \right), \quad (5)$$

By using (4) and (5), then the first step of the scheme (3) yields

$$y_n - \alpha = \frac{C_1 \beta}{m} e_n^2 - \frac{\beta^2}{m^2} ((m+1)C_1^2 - 2mC_2) e_n^3 + O((e_n)^4). \quad (6)$$

Expansion of $f(y_n)$ about α leads us to the expression

$$\begin{aligned} f(y_n) = \frac{f^{(m)}(\alpha)}{m!} \left(\frac{\beta C_1}{m} \right)^m e_n^{2m} & \left(1 - \frac{\beta X_1}{C_1} e_n + \frac{\beta}{2mC_1^2} (\beta(m-1)X_1^2 + 2C_1^4) e_n^2 \right. \\ & \left. - \frac{\beta X_1}{6m^2 C_1^3} (6\beta C_1^4(m+1) + \beta^2(m-1)(m-2)X_1^2) e_n^3 \right) + O((e_n^4)), \end{aligned} \quad (7)$$

where $X_1 = (m+1)C_1^2 - 2mC_2$.

Expanding $H(u)$ in the neighborhood of '0', then using (4) and (7) in the expansion of $u = \left(\frac{f(y_n)}{f(x_n)} \right)^{\frac{1}{m}}$, we have

$$H(u) = H(0) + H'(0) \left(\frac{C_1 \beta}{m} e_n - \frac{\beta}{m^2} (\beta X_1 + C_1^2) e_n^2 + \frac{\beta C_1}{2m^3} ((2\beta+1)X_1 + 2C_1^2) \right) e_n^3 + O((e_n)^4). \quad (8)$$

Using the above expressions in the last step of (3), it follows that

$$\begin{aligned} e_{n+1} = & -\frac{H(0)}{m} e_n + \frac{\beta C_1}{m^2} (m + H(0) - H'(0)) e_n^2 - \frac{1}{m^3} (\beta^2 ((1+m)C_1^2 + 2C_2) (m + H(0) + H'(0)) \\ & - 2\beta C_1^2 H'(0)) e_n^3 + O((e_n)^4). \end{aligned} \quad (9)$$

It is clear that the third order of convergence is possible only if $H(0) = 0$, $H'(0) = m$. Then, the error Equation (9) becomes

$$e_{n+1} = \frac{2\beta C_1^2}{m^2} e_n^3 + O((e_n)^4).$$

Thus, the theorem is proved. \square

Some Concrete Forms of $H(u)$

We can obtain numerous methods of the family (3) based on the form of function $H(u)$ that satisfies the conditions of Theorem 1. but we will limit choices to consider the forms of low-order polynomials or simple functions. Accordingly, the following simple forms are chosen:

- (1) $H(u) = mu$,
- (2) $H(u) = \frac{mu}{1+u}$,
- (3) $H(u) = \frac{mu}{1-u}$,
- (4) $H(u) = \frac{mu}{1+mu}$,
- (5) $H(u) = m \log(u+1)$,
- (6) $H(u) = m(e^u - 1)$.

The corresponding method for each of the above forms can be expressed as:

Method 1 (M1):

$$x_{n+1} = y_n - mu \frac{f(x_n)}{f[x_n, w_n]}.$$

Method 2 (M2):

$$x_{n+1} = y_n - \frac{mu}{1+u} \frac{f(x_n)}{f[x_n, w_n]}.$$

Method 3 (M3):

$$x_{n+1} = y_n - \frac{mu}{1-u} \frac{f(x_n)}{f[x_n, w_n]}.$$

Method 4 (M4):

$$x_{n+1} = y_n - \frac{mu}{1+mu} \frac{f(x_n)}{f[x_n, w_n]}.$$

Method 5 (M5):

$$x_{n+1} = y_n - m \log(u+1) \frac{f(x_n)}{f[x_n, w_n]}.$$

Method 6 (M6):

$$x_{n+1} = y_n - m(e^u - 1) \frac{f(x_n)}{f[x_n, w_n]}.$$

In the above each case $u = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$ and $y_n = x_n - m \frac{f(x_n)}{f[x_n, w_n]}.$

Remark 1. Computational competence of an iterative method is spanned by the efficiency index $E = s^{1/d}$ (see [17]), where s is the order of convergence and d is the computational cost measured as the number of new pieces of information needed by the method per iterative step. A “piece of information” typically is any calculation of the function f or one of its derivatives. The presented third-order methods require three function calculations per iteration, so the efficiency index of the methods is $E = \sqrt[3]{3} \approx 1.442$ which is equal to the efficiency index of existing third-order methods such as Dong [5], Halley [7], Chebyshev [7], Osada [12], and Victory and Neta [14]. Note, however, that these existing third-order methods require derivative evaluations in their computation which is not the case with the proposed methods. Also note that efficiency index of new methods is better than that of modified Newton’s method and derivative-free Traub–Steffensen’s method ($E = \sqrt{2} \approx 1.414$).

3. Numerical Tests

In this section, first we plot the basins of attraction of the zeros of some polynomials when the proposed iterative methods are applied on the polynomials. Next, we verify the theoretical results by the applications of the methods on some nonlinear functions, including those which arise in practical problems.

3.1. Basins of Attraction

Analysis of the complex dynamical behavior gives important information about convergence and stability of an iterative scheme; see e.g., [2,3,18–21]. Here, we directly describe the dynamics of iterative methods M1–M6 by means of visual display of the basins of attraction of the zeros of a polynomial $P(z) \in \mathbb{C}$. The deep dynamical study of the proposed six methods, with a detailed stability analysis of the fixed points and their behavior, requires a separate article, and so such complete analysis is not a subject of discussion in the present work.

To start with, let us take the initial point z_0 in a rectangular region $R \in \mathbb{C}$ that contains all the zeros of a polynomial $P(z)$. The iterative method starts from point z_0 in a rectangle either converges to the zero $P(z)$ or eventually diverges. The stopping criterion for convergence is considered to be 10^{-3} up to a maximum of 25 iterations. If the required tolerance is not achieved in 25 iterations, we conclude that the method starting at point z_0 does not converge to any root. The strategy adopted is as follows: A color is allocated to each initial point z_0 in the basin of attraction of a zero. If the iteration initiating at z_0 converges, then it represents the attraction basin with that particular assigned color to it, otherwise if it fails to converge in 25 iterations, then it shows the black color.

We plot the basins of attraction of the methods M1–M6 (for the choices $\beta = 10^{-2}, 10^{-4}, 10^{-6}$) on following three polynomials:

Problem 1. In the first example, we look at the polynomial $P_1(z) = (z^2 - 1)^2$ which has zeros $\{\pm 1\}$ with multiplicity 2. For making basins, we use a grid of 400×400 points in a rectangle $D \in \mathbb{C}$ of size $[-2, 2] \times [-2, 2]$ and fix the color green to each initial point in the basin of attraction of zero ‘-1’ and the color red to each point in the basin of attraction of zero ‘1’. Basins achieved for the methods M1–M6 are shown in Figures 1–3 corresponding to $\beta = 10^{-2}, 10^{-4}, 10^{-6}$. Looking at the behavior of the methods, we see that methods M2 and M4 possess fewest divergent points, followed by M3 and M5. On the contrary, the method M6 has the most divergent points, followed by M1. Notice that the basins are becoming better as parameter β assumes smaller values.

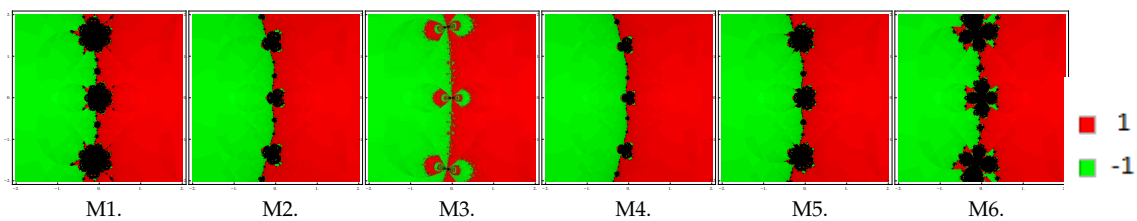


Figure 1. Basins of attraction for polynomial $P_1(z)$ for $\beta = 10^{-2}$.

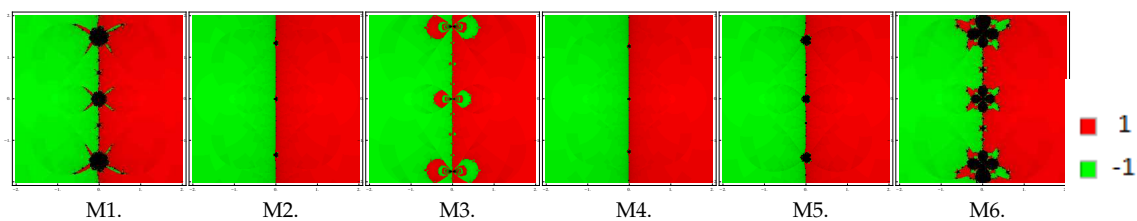


Figure 2. Basins of attraction for polynomial $P_1(z)$ for $\beta = 10^{-4}$.

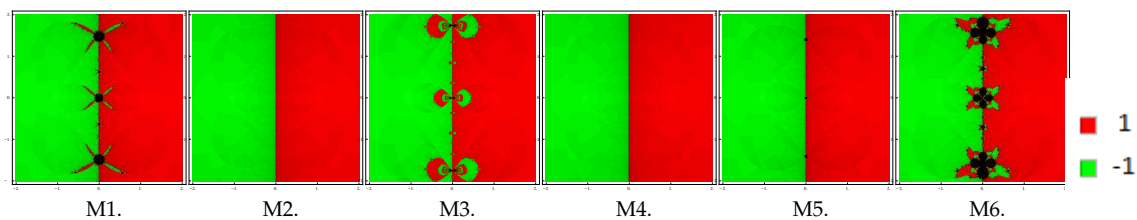


Figure 3. Basins of attraction for polynomial $P_1(z)$ for $\beta = 10^{-6}$.

Problem 2. Let us consider the polynomial $P_2(z) = (z^3 - z)^3$ with zeros $\{0, \pm 1\}$ with multiplicity 3. To examine the dynamical view, we consider a rectangle $D = [-2, 2] \times [-2, 2] \in \mathbb{C}$ with 400×400 grid points and allot the colors red, green, and blue to each point in the basin of attraction of $-1, 0$ and 1 , respectively. Basins of this problem are shown in Figures 4–6 corresponding to parameter choices $\beta = 10^{-2}, 10^{-4}, 10^{-6}$ in the proposed methods. Observing the behavior of the methods, we see that methods M3, M2, and M4 have better convergence behavior since they have the least number of divergent points. On the other hand, M6 contains large black regions followed by M1 and M5, indicating that the methods do not converge in 25 iterations starting at those points. Also, observe that the basins are improving with the smaller values of β .

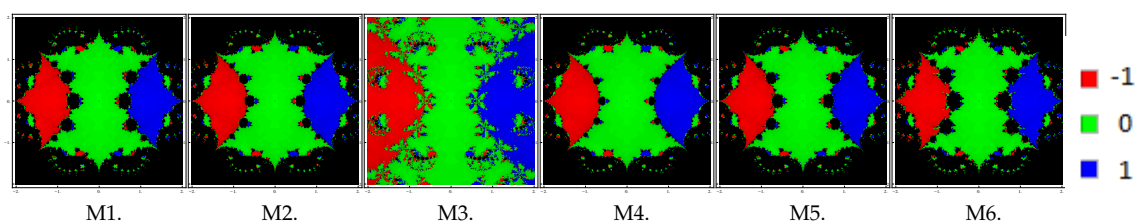


Figure 4. Basins of attraction for polynomial $P_2(z)$ for $\beta = 10^{-2}$.

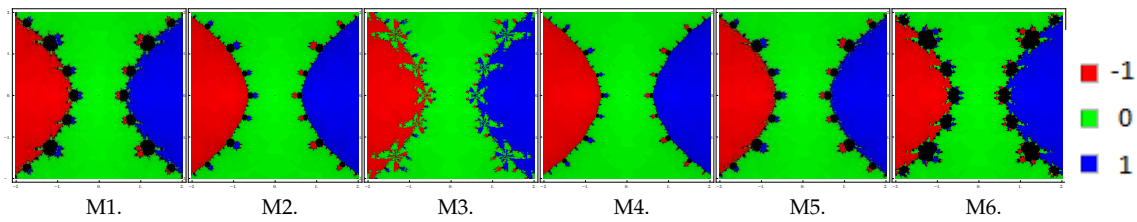


Figure 5. Basins of attraction for polynomial $P_2(z)$ for $\beta = 10^{-4}$.

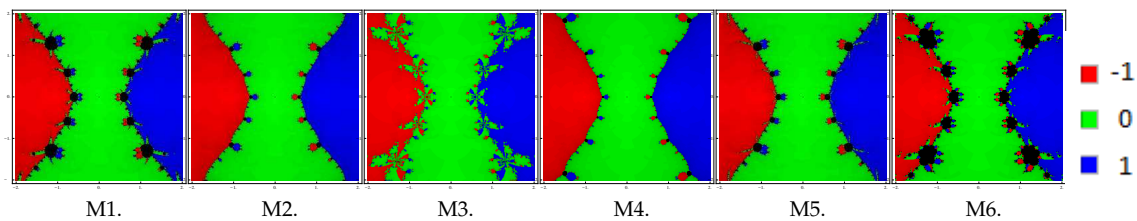


Figure 6. Basins of attraction for polynomial $P_2(z)$ for $\beta = 10^{-6}$.

Problem 3. Lastly, we consider the polynomial $P_3(z) = (z^4 - 1)^2$ with zeros $\{\pm i, \pm 1\}$ with multiplicity 2. To see the dynamical view, we consider a rectangle $D = [-2, 2] \times [-2, 2] \in \mathbb{C}$ with 400×400 grid points and allocate the colors red, blue, yellow, and green to each point in the basin of attraction of -1 , 1 , $-i$ and i , respectively. Basins for this problem are shown in Figures 7–9 corresponding to parameter choices $\beta = 10^{-2}$, 10^{-4} , 10^{-6} in the proposed methods. Observing the behavior of the methods, we see that methods M3, M2, and M4 have better convergence behavior since they have fewest divergent points. On the other hand, M6 contains large black regions followed by M1 and M5 indicating that the methods do not converge in 25 iterations starting at those points. Also, observe that the basins are becoming larger with the smaller values of β .

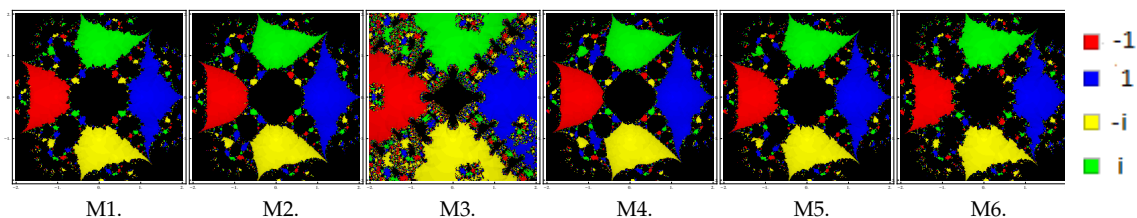


Figure 7. Basins of attraction for polynomial $P_3(z)$ for $\beta = 10^{-2}$.

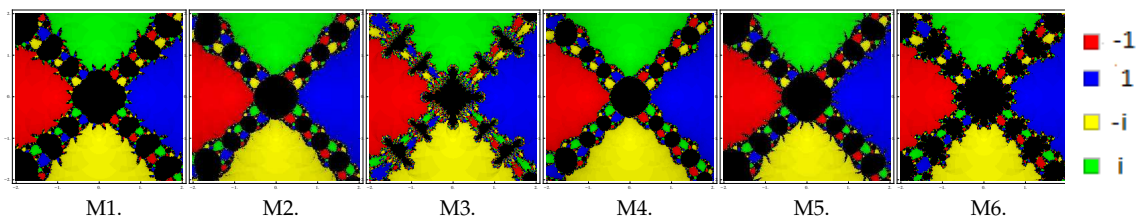


Figure 8. Basins of attraction for polynomial $P_3(z)$ for $\beta = 10^{-4}$.

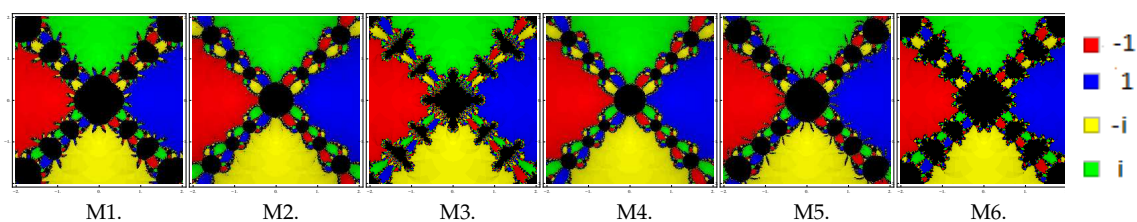


Figure 9. Basins of attraction for polynomial $P_3(z)$ for $\beta = 10^{-6}$.

From these graphics, one can easily evaluate the behavior and stability of any method. If we choose an initial point z_0 in a zone where distinct basins of attraction touch each other, it is impractical to predict which root is going to be attained by the iterative method that starts in z_0 . Hence, the choice of z_0 in such a zone is not a good one. Both the black zones and the regions with different colors are not suitable to take the initial guess z_0 when we want to acquire a unique root. The most adorable pictures appear when we have very tricky frontiers between basins of attraction, and they correspond to the cases where the method is more demanding with respect to the initial point and its dynamic behavior is more unpredictable. We conclude this section with a remark that the convergence nature of proposed methods depends upon the value of parameter β . The smaller the value of β , the better the convergence of the method.

3.2. Applications

The above six methods M1–M6 of family (3) are applied to solve a few nonlinear equations, which not only depict the methods practically but also serve to verify the validity of theoretical results we have derived. To investigate the theoretical order of convergence, we obtain the computational order of convergence (COC) using the formula (see [22])

$$\text{COC} = \frac{\ln |(x_{n+1} - \alpha)/(x_n - \alpha)|}{\ln |(x_n - \alpha)/(x_{n-1} - \alpha)|}.$$

Performance is compared with some well-known third-order methods requiring derivative evaluations such as Dong [5], Halley [7], Chebyshev [7], Osada [12], and Victory and Neta [14]. These methods are expressed as:

Dong's method (DM):

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

Halley's method (HM):

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

Chebyshev's method (CM):

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

Osada's method (OM):

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

Victory-Neta method (VNM):

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \frac{\frac{f(y_n)}{f'(y_n)} \frac{f(x_n) + Af(y_n)}{f(x_n) + Bf(y_n)}}{\frac{f(y_n)}{f'(y_n)} \frac{f(x_n) + Af(y_n)}{f(x_n) + Bf(y_n)}}, \end{cases}$$

where

$$A = \mu^{2m} - \mu^{m+1}, \quad B = -\frac{\mu^m(m-2)(m-1)+1}{(m-1)^2} \quad \text{and} \quad \mu = \frac{m}{m-1}.$$

All computations are determined in the programming package *Mathematica* using multiple-precision arithmetic. Performance of the new methods is tested by choosing value of the parameter $\beta = -0.01$. Numerical results displayed in Tables 1–4 include: (i) values of last three

consecutive errors $|x_{n+1} - x_n|$; (ii) number of iterations (n) required to converge to the solution such that $|x_{n+1} - x_n| + |f(x_n)| < 10^{-100}$; (iii) COC; and (iv) the elapsed CPU time (CPU-time) in seconds.

The following examples are chosen for numerical tests:

Example 1 (Eigenvalue problem). Finding eigenvalues of a large sparse matrix is one of the most challenging tasks in applied mathematics and engineering. Furthermore, calculating the zeros of a characteristic equation of square matrix greater than 4 is another big challenge. Therefore, we consider the following 9×9 matrix

$$M = \frac{1}{8} \begin{bmatrix} -12 & 0 & 0 & 19 & -19 & 76 & -19 & 18 & 437 \\ -64 & 24 & 0 & -24 & 24 & 64 & -8 & 32 & 376 \\ -16 & 0 & 24 & 4 & -4 & 16 & -4 & 8 & 92 \\ -40 & 0 & 0 & -10 & 50 & 40 & 2 & 20 & 242 \\ -4 & 0 & 0 & -1 & 41 & 4 & 1 & 2 & 25 \\ -40 & 0 & 0 & 18 & -18 & 104 & -18 & 20 & 462 \\ -84 & 0 & 0 & -29 & 29 & 84 & 21 & 42 & 501 \\ 16 & 0 & 0 & -4 & 4 & -16 & 4 & 16 & -92 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 \end{bmatrix}.$$

The characteristic polynomial of the matrix (M) is given as

$$f_1(x) = x^9 - 29x^8 + 349x^7 - 2261x^6 + 8455x^5 - 17663x^4 + 15927x^3 + 6993x^2 - 24732x + 12960.$$

This function has one multiple zero at $\alpha = 3$ of multiplicity 4. We find this zero with initial approximation $x_0 = 2.8$. Numerical results are shown in Table 1.

Table 1. Comparison of performance of methods for Example 1.

Methods	$ x_3 - x_2 $	$ x_4 - x_3 $	$ x_5 - x_4 $	n	COC	CPU-Time
DM	9.90×10^{-11}	1.52×10^{-31}	5.49×10^{-94}	5	3.0000	0.01942
HM	5.84×10^{-10}	4.61×10^{-29}	2.24×10^{-86}	5	3.0000	0.01950
CM	9.54×10^{-10}	2.47×10^{-28}	4.30×10^{-84}	5	3.0000	0.01885
OM	1.26×10^{-9}	6.52×10^{-28}	8.94×10^{-83}	5	3.0000	0.01955
VNM	2.50×10^{-10}	2.92×10^{-30}	4.68×10^{-90}	5	3.0000	0.02250
M1	1.51×10^{-12}	3.91×10^{-37}	0	4	3.0000	0.00775
M2	5.15×10^{-12}	2.30×10^{-35}	0	4	3.0000	0.01165
M3	2.32×10^{-13}	7.01×10^{-40}	0	4	3.0000	0.01175
M4	4.73×10^{-11}	3.59×10^{-32}	1.57×10^{-95}	5	3.0000	0.01285
M5	2.94×10^{-12}	3.57×10^{-36}	0	4	3.0000	0.01775
M6	6.71×10^{-13}	2.55×10^{-38}	0	4	3.0000	0.01115

Example 2 (Manning equation). Consider isentropic supersonic flow along a sharp expansion corner. The relationship between the Mach number before the corner (i.e., M_1) and after the corner (i.e., M_2) is given by (see [23])

$$\delta = b^{1/2} \left(\tan^{-1} \left(\frac{M_2^2 - 1}{b} \right)^{1/2} - \tan^{-1} \left(\frac{M_1^2 - 1}{b} \right)^{1/2} \right) - \left(\tan^{-1}(M_2^2 - 1)^{1/2} - \tan^{-1}(M_1^2 - 1)^{1/2} \right),$$

where $b = \frac{\gamma+1}{\gamma-1}$ and γ is the specific heat ratio of the gas.

For a special case study, we solve the equation for M_2 given that $M_1 = 1.5$, $\gamma = 1.4$ and $\delta = 10^0$. In this case, we have

$$\tan^{-1}\left(\frac{\sqrt{5}}{2}\right) - \tan^{-1}(\sqrt{x^2-1}) + \sqrt{6}\left(\tan^{-1}\left(\sqrt{\frac{x^2-1}{6}}\right) - \tan^{-1}\left(\frac{1}{2}\sqrt{\frac{5}{6}}\right)\right) - \frac{11}{63} = 0,$$

where $x = M_2$.

We consider this case for seven times and obtained the required nonlinear function

$$f_2(x) = \left[\tan^{-1}\left(\frac{\sqrt{5}}{2}\right) - \tan^{-1}(\sqrt{x^2-1}) + \sqrt{6}\left(\tan^{-1}\left(\sqrt{\frac{x^2-1}{6}}\right) - \tan^{-1}\left(\frac{1}{2}\sqrt{\frac{5}{6}}\right)\right) - \frac{11}{63} \right]^7.$$

The above function has zero at $\alpha = 1.8411027704926161 \dots$ with multiplicity 7. The required zero is determined by using initial approximation $x_0 = 1.5$. Numerical results are shown in Table 2.

Table 2. Comparison of performance of methods for Example 2.

Methods	$ x_3 - x_2 $	$ x_4 - x_3 $	$ x_5 - x_4 $	n	COC	CPU-Time
DM	6.14×10^{-9}	1.48×10^{-26}	2.06×10^{-79}	5	3.0000	0.17575
HM	5.10×10^{-8}	1.20×10^{-23}	1.54×10^{-70}	5	3.0000	0.20354
CM	5.98×10^{-8}	2.17×10^{-23}	1.04×10^{-69}	5	3.0000	0.23560
OM	6.30×10^{-8}	2.63×10^{-23}	1.91×10^{-69}	5	3.0000	0.23225
VNM	2.45×10^{-8}	1.17×10^{-24}	1.28×10^{-73}	5	3.0000	0.24875
M1	1.28×10^{-15}	4.70×10^{-47}	0	4	3.0000	0.11121
M2	2.95×10^{-15}	8.62×10^{-46}	0	4	3.0000	0.12100
M3	3.86×10^{-16}	6.45×10^{-49}	0	4	3.0000	0.10925
M4	4.96×10^{-14}	1.23×10^{-41}	0	4	3.0000	0.11725
M5	2.00×10^{-15}	2.24×10^{-46}	0	4	3.0000	0.12125
M6	7.54×10^{-16}	7.21×10^{-48}	0	4	3.0000	0.12525

Example 3. Next, we assume a standard nonlinear test function defined by

$$f_3(x) = -\frac{x^4}{12} + \frac{x^2}{2} + x + e^x(x-3) + \sin x + 3.$$

The function f_3 has multiple zero at $\alpha = 0$ of multiplicity 3. We choose the initial approximation $x_0 = 0.5$ for obtaining the zero of the function. Numerical results are displayed in Table 3.

Table 3. Comparison of performance of methods for Example 3.

Methods	$ x_3 - x_2 $	$ x_4 - x_3 $	$ x_5 - x_4 $	n	COC	CPU-Time
DM	1.02×10^{-9}	3.43×10^{-29}	1.31×10^{-87}	5	3.0000	0.12925
HM	2.58×10^{-8}	1.09×10^{-24}	8.36×10^{-74}	5	3.0000	0.14450
CM	2.85×10^{-8}	1.65×10^{-24}	3.16×10^{-73}	5	3.0000	0.13275
OM	3.13×10^{-8}	2.39×10^{-24}	1.06×10^{-72}	5	3.0000	0.12500
VNM	5.37×10^{-8}	7.00×10^{-27}	1.56×10^{-80}	5	3.0000	0.21355
M1	1.88×10^{-13}	9.27×10^{-41}	0	4	3.0000	0.05475
M2	6.24×10^{-13}	5.05×10^{-39}	0	4	3.0000	0.05852
M3	3.10×10^{-14}	2.06×10^{-43}	0	4	3.0000	0.05075
M4	3.15×10^{-12}	1.09×10^{-36}	0	4	3.0000	0.05475
M5	3.60×10^{-13}	8.07×10^{-40}	0	4	3.0000	0.05475
M6	8.56×10^{-14}	6.54×10^{-42}	0	4	3.0000	0.06253

Example 4. Lastly, we assume a standard nonlinear test function defined by

$$f_4(x) = 2(x^2 + 1)(2xe^{x^2+1} + x^3 - x) \cosh^2\left(\frac{\pi x}{2}\right).$$

The function f_4 has complex zero at $\alpha = i$ of multiplicity 4. We choose the initial approximation $x_0 = 1.25i$ for obtaining the zero of the function. Numerical results are displayed in Table 4.

Table 4. Comparison of performance of methods for Example 4.

Methods	$ x_3 - x_2 $	$ x_4 - x_3 $	$ x_5 - x_4 $	n	COC	CPU-Time
DM	7.61×10^{-9}	1.42×10^{-25}	9.14×10^{-76}	5	3.0000	1.047
HM	6.17×10^{-8}	1.12×10^{-22}	6.66×10^{-67}	5	3.0000	2.141
CM	7.82×10^{-8}	2.81×10^{-22}	1.31×10^{-65}	5	3.0000	1.985
OM	8.97×10^{-8}	4.78×10^{-22}	7.22×10^{-65}	5	3.0000	1.906
VNM	2.42×10^{-8}	5.53×10^{-24}	6.59×10^{-71}	5	3.0000	1.219
M1	7.10×10^{-12}	7.96×10^{-35}	0	4	3.0000	0.390
M2	1.88×10^{-11}	2.20×10^{-33}	3.54×10^{-99}	5	3.0000	0.516
M3	1.72×10^{-12}	5.66×10^{-37}	0	4	3.0000	0.500
M4	1.22×10^{-10}	1.22×10^{-30}	1.21×10^{-90}	5	3.0000	0.578
M5	1.20×10^{-11}	4.74×10^{-34}	0	4	3.0000	0.359
M6	3.80×10^{-12}	9.18×10^{-36}	0	4	3.0000	0.313

From the numerical results we examine that the accuracy in the values of successive approximations rises, which shows the stable nature of the methods. Also, as with the existing methods, the present methods show consistent convergence behavior. We display the value ‘0’ of $|x_{n+1} - x_n|$ in the iteration at which $|x_{n+1} - x_n| + |F(x_n)| < 10^{-100}$. From the calculation of computational order of convergence, it is also confirmed that the order of convergence of the methods is preserved. The efficient nature of proposed methods can be observed by the fact that the amount of CPU time consumed by the methods is less than that of the time taken by existing methods. In addition, the new methods are more accurate because error becomes much smaller with increasing n as compared to the error of existing techniques. The main purpose of developing the new derivative-free methods for different types of nonlinear equations is purely to illustrate the exactness of the approximate solution and the stability of the convergence to the required solution. Similarly, numerical experimentations, carried out for several problems of different type, proved the above conclusions to a large extent.

4. Conclusions

In this study, we have suggested a class of third-order derivative-free methods for solving nonlinear equations with multiple roots, with known multiplicity. Analysis of the convergence has shown the third-order convergence under basic assumptions regarding the nonlinear function whose zeros we are searching for. Some special cases of the class are presented. These are employed to solve some nonlinear equations and compared with existing techniques. Testing of the numerical results shows that the presented derivative-free methods are good competitors to the existing third-order techniques that require derivative evaluations in their algorithm.

Author Contributions: All the authors have equal contribution to this study.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schröder, E. Über unendlich viele Algorithmen zur Auflösung der Gleichungen. *Math. Ann.* **1870**, *2*, 317–365. [\[CrossRef\]](#)
2. Behl, R.; Cordero, A.; Motsa, S.S.; Torregrosa, J.R. On developing fourth-order optimal families of methods for multiple roots and their dynamics. *Appl. Math. Comput.* **2015**, *265*, 520–532. [\[CrossRef\]](#)
3. Behl, R.; Cordero, A.; Motsa, S.S.; Torregrosa, J.R.; Kanwar, V. An optimal fourth-order family of methods for multiple roots and its dynamics. *Numer. Algor.* **2016**, *71*, 775–796. [\[CrossRef\]](#)

4. Behl, R.; Zafar, F.; Alshormani, A.S.; Junjua, M.U.D.; Yasmin, N. An optimal eighth-order scheme for multiple zeros of univariate functions. *Int. J. Comput. Meth.* **2018**, *15*. [[CrossRef](#)]
5. Dong, C. A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation. *Math. Numer. Sin.* **1982**, *11*, 445–450.
6. Geum, Y.H.; Kim, Y.I.; Neta, B. A class of two-point sixth-order multiple-zero finders of modified double-Newton type and their dynamics. *Appl. Math. Comput.* **2015**, *270*, 387–400. [[CrossRef](#)]
7. Hansen, E.; Patrick, M. A family of root finding methods. *Numer. Math.* **1977**, *27*, 257–269. [[CrossRef](#)]
8. Kansal, M.; Kanwar, V.; Bhatia, S. On some optimal multiple root-finding methods and their dynamics. *Appl. Appl. Math.* **2015**, *10*, 349–367.
9. Li, S.G.; Cheng, L.Z.; Neta, B. Some fourth-order nonlinear solvers with closed formulae for multiple roots. *Comput. Math. Appl.* **2010**, *59*, 126–135. [[CrossRef](#)]
10. Li, S.; Liao, X.; Cheng, L. A new fourth-order iterative method for finding multiple roots of nonlinear equations. *Appl. Math. Comput.* **2009**, *215*, 1288–1292.
11. Neta, B. New third order nonlinear solvers for multiple roots. *Appl. Math. Comput.* **2008**, *202*, 162–170. [[CrossRef](#)]
12. Osada, N. An optimal multiple root-finding method of order three. *J. Comput. Appl. Math.* **1994**, *51*, 131–133. [[CrossRef](#)]
13. Sharma, J.R.; Sharma, R. Modified Jarratt method for computing multiple roots. *Appl. Math. Comput.* **2010**, *217*, 878–881. [[CrossRef](#)]
14. Victory, H.D.; Neta, B. A higher order method for multiple zeros of nonlinear functions. *Int. J. Comput. Math.* **1983**, *12*, 329–335. [[CrossRef](#)]
15. Zhou, X.; Chen, X.; Song, Y. Constructing higher-order methods for obtaining the multiple roots of nonlinear equations. *J. Comput. Appl. Math.* **2011**, *235*, 4199–4206. [[CrossRef](#)]
16. Traub, J.F. *Iterative Methods for the Solution of Equations*; Chelsea Publishing Company: New York, NY, USA, 1982.
17. Ostrowski, A.M. *Solution of Equations and Systems of Equations*; Academic Press: New York, NY, USA, 1966.
18. Vrscay, E.R.; Gilbert, W.J. Extraneous fixed points, basin boundaries and chaotic dynamics for Schröder and König rational iteration functions. *Numer. Math.* **1988**, *52*, 1–16. [[CrossRef](#)]
19. Varona, J.L. Graphic and numerical comparison between iterative methods. *Math. Intell.* **2002**, *24*, 37–46. [[CrossRef](#)]
20. Scott, M.; Neta, B.; Chun, C. Basin attractors for various methods. *Appl. Math. Comput.* **2011**, *218*, 2584–2599. [[CrossRef](#)]
21. Lotfi, T.; Sharifi, S.; Salimi, M.; Siegmund, S. A new class of three-point methods with optimal convergence order eight and its dynamics. *Numer. Algor.* **2015**, *68*, 261–288. [[CrossRef](#)]
22. Weerakoon, S.; Fernando, T.G.I. A variant of Newton's method with accelerated third-order convergence. *Appl. Math. Lett.* **2000**, *13*, 87–93. [[CrossRef](#)]
23. Hoffman, J.D. *Numerical Methods for Engineers and Scientists*; McGraw-Hill Book Company: New York, NY, USA, 1992.

